SVEUČILIŠTE U ZAGREBU
PMF – MATEMATIČKI ODJEL

Zlatko Drmač
Vjeran Hari
Miljenko Marušić
Mladen Rogina
Sanja Singer
Saša Singer

Numerička analiza

Predavanja i vježbe

Zagreb, 2003.

Predgovor

Ova elektronička knjiga nastala je zalaganjem grupe znanstvenika okupljenih na *Projektu primjene informatičke tehnologije 2001–96*, "Numerička matematika: osnovni udžbenik", Ministarstva znanosti i tehnologije Republike Hrvatske. Pisana je u 2002. i 2003. godini.

Knjiga je prvenstveno namijenjena studentima matematike i fizike te boljim studentima tehnike i ekonomije. Knjiga daje širi pregled područja numeričke matematike i predstavlja centralnu točku i osnovu za pisanje nadovezujućih knjiga iz srodnih područja: numeričke linearne algebre, numeričkog rješavanja običnih i parcijalnih diferencijalnih jednadžbi, računalne geometrije, računalne grafike, itd. Knjiga također služi kao izvor dovoljno opsežnih informacija za pisanje jednostavnijih udžbenika iz numeričke matematike, prilagođenih pojedinim fakultetima, npr. tehničkim, ekonomskim ili prirodoslovnim. Ona je zamišljena i kao začetak informacijskog portala za numeričku matematiku u Hrvatskoj, na kojem bi studenti, znanstvenici i svi zainteresirani mogli naći sadržaj većine disertacija, magistarskih i diplomskih radova iz šireg područja numeričke matematike, software, kao i pokazivače na najpoznatije svjetske web-portale iz numeričke matematike. Knjiga će se iz godine u godinu poboljšavati, dodavat će se novi primjeri, zadaci, slike, reference i pokazivači na druge sadržaje.

Knjiga je podijeljena u jedanaest poglavlja. U uvodnom dijelu dani su neki osnovni rezultati iz analize, kao npr. teoremi o srednjoj vrijednosti koji će se koristiti kasnije. U drugom poglavlju dana je opsežna i moderna analiza grešaka koje prate računanje na modernim računalima. U trećem poglavlju uvode se glavni pojmovi iz linearne algebre kao što su vektorski prostori, matrice i norme. U četvrtom poglavlju dana je teorija sustava linearnih jednadžbi i nekoliko osnovnih metoda za njihovo rješavanje. Peto poglavlje bavi se teorijom i algoritmima za nalaženje vlastitih vrijednosti matrica. Šesto poglavlje bavi se izračunavanjem vrijednosti funkcija. Sedmo, najopsežnije poglavlje, bavi se teorijom aproksimacije koja uključuje interpolaciju, numeričko deriviranje, aproksimaciju spline funkcijama i ortogonalnim polinomima i metodu najmanjih kvadrata. Osmo poglavlje bavi se numeričkim nalaženjem nultočaka funkcija, a deveto numeričkim integriranjem. U desetom poglavlju, opisane su glavne metode za numeričko rješavanje običnih diferencijalnih jednadžbi, a u kraćem jedanaestom poglavlju dan je uvod u optimizaciju.

Posao pisanja knjige bio je podijeljen tako da je svaki suradnik (ili grupa) napisao jedno ili više poglavlja. Voditelj grupe Vjeran Hari napisao je prva dva i dio (odjeljci 1., 2. i 3.) trećeg poglavlja. Zlatko Drmač napisao je četvrto, a Ivan Slapničar (uz malu pomoć voditelja) peto poglavlje. Sanja i Saša Singer napisali velik dio knjige, šesto, osmo, deveto i dio trećeg poglavlja. Sedmo poglavlje napisali su zajedničkim snagama Mladen Rogina (dio odjeljka 2. i odjeljak 4.) te Sanja i Saša Singer. Miljenko Marušić napisao je deseto i jedanaesto poglavlje. Postoje i Sanjini i

Sašini dodatni sadržaji do kojih se dođe preko pokazivača. Sanja je također preuzela težak posao uređivanja knjige, nakon što su suradnici napisali svoje dijelove.

Knjiga je napisana u LaTeX-u, čiji lijepi stil su definirali Sanja i Saša. U HTML je prevedena na FESB-u Sveučilišta u Splitu zahvaljujući grupi oko Ivana. Mladen vodi brigu oko administriranja knjige što uključuje ugrađivanje novih sadržaja. Osim DVI formata koji se čita pomoću posebnih programa (tzv. dvi-preglednika), postoje i verzije knjige u jeziku Postscript i PDF (portable document format), tako da se može čitati npr. pomoću rasprostranjenog Acrobat readera. Nalazi se na web adresi: `www.math.hr/znanost/iprojekti/numat`.

Na kraju, želim zahvaliti suradnicima na zalaganju i Ministarstvu na pomoći.

Vjeran Hari
Redoviti profesor na PMF-MO
Sveučilišta u Zagrebu

Sadržaj

1. Uvodni dio

Numerička matematika je disciplina koja proučava i numerički rješava matematičke probleme koji se javljaju u znanosti, tehnici, gospodarstvu, itd. Iako se ta disciplina najčešće povezuje s numeričkim metodama, treba znati da bez dubljeg poznavanja samog problema kojeg rješavamo, nije moguće procijeniti je li neka metoda dobra u smislu da daje zadovoljavajuće točna rješenja u dovoljno kratkom vremenskom intervalu. O problemu koji se rješava treba znati barem neka svojstva:

- postoji li barem jedno rješenje, ako da, je li rješenje jedinstveno,
- kako se rješenje (ili rješenja) ponašaju kad se polazni podaci malo promijene (teorija perturbacije).

Kad se konstruira neka metoda za dani problem otvara se mnoštvo pitanja:

- konvergencije (konvergira li niz aproksimacija prema rješenju),
- brzine konvergencije (tu se koriste termini linearna, kvadratična, kubična konvergencija),
- adaptibilnost metode za specijalna računala (paralelna, vektorska),
- složenost metode (broj računskih operacija, zauzeće memorije, dohvat operanada, prijenos podataka, ...)
- točnost, odnosno stabilnost metode (koliko značajnih znamenaka izračunatog rješenja je točno, te o čemu to ovisi).

Uz određivanje stabilnosti algoritma vezana je i analiza grešaka zaokruživanja koja pokazuje mogu li greške koje aritmetika računala generira u procesu računanja bitno narušiti točnost izlaznih podataka.

U prvom poglavlju knjige uvest ćemo oznake osnovnih pojmova koji se koriste u knjizi. Neke pojmove ćemo definirati, a za neke ćemo dati najvažnije tvrdnje.

1.1. Pomoćni rezultati iz analize

Ovdje navodimo, bez dokaza, nekoliko važnih teorema iz analize. Najprije moramo uvesti oznake za osnovne skupove koji se koriste u analizi.

S $\mathbb{N}$ označavamo skup prirodnih brojeva, dakle $\{1, 2, \ldots\}$. S $\mathbb{N}_0$ označavamo skup nenegativnih cijelih brojeva, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Sa $\mathbb{Z}$ označavamo skup svih cijelih brojeva, dakle pozitivnih, negativnih i nule. S $\mathbb{Q}$ označavamo skup racionalnih, a s $\mathbb{Q}_0$ skup nenegativnih racionalnih brojeva. S $\mathbb{R}$ ($\mathbb{C}$) označavamo skup realnih (kompleksnih) brojeva.

Ako je $\mathcal{S} \subseteq \mathbb{C}$ (ili $\mathcal{S} \subseteq \mathbb{R}$), tada je zatvarač $\mathrm{Cl}\,\mathcal{S}$ skupa $\mathcal{S}$, skup svih gomilišta od $\mathcal{S}$. S obzirom da je svaka točka od $\mathcal{S}$ ujedno i gomilište od $\mathcal{S}$, vrijedi $\mathcal{S} \subseteq \mathrm{Cl}\,\mathcal{S}$. Zatvarač $\mathrm{Cl}\,\mathcal{S}$ je najmanji zatvoren skup koji sadrži $\mathcal{S}$. S druge strane, unutrašnjost skupa $\mathcal{S}$, $\mathrm{Int}\,\mathcal{S}$ ili interior od $\mathcal{S}$ je najveći otvoren skup sadržan u $\mathcal{S}$. Jasno je da je $\mathrm{Int}\,\mathcal{S} \subseteq \mathcal{S} \subseteq \mathrm{Cl}\,\mathcal{S}$ pri čemu jednakosti vrijede npr. za cijeli $\mathbb{C}$ (ili $\mathbb{R}$). Ako je recimo, $\mathrm{Cl}\,\mathcal{S}$ zatvoreni krug, onda je $\mathrm{Int}\,\mathcal{S}$ otvoreni krug (krug bez rubne kružnice), a $\mathcal{S}$ može biti npr. otvoreni krug s bilo kojim točkama na rubu.

U numeričkoj matematici posebno su važni sljedeći teoremi o srednjim vrijednostima kojima se dokazi mogu pronaći u knjigama iz elementarne analize.

Teorem 1.1.1 (Međuvrijednost) *Neka je f realna neprekidna funkcija na konačnom segmentu $[a, b]$. Neka su*

$$m = \inf_{a \le x \le b} f(x) \quad i \quad M = \sup_{a \le x \le b} f(x)$$

infimum i supremum funkcije f. Tada za svaki realni broj β iz segmenta $[m, M]$, postoji barem jedan realni broj α iz $[a, b]$, takav da je

$$f(\alpha) = \beta.$$

Specijalno, postoje brojevi $\underline{x}$ i $\overline{x}$ iz $[a, b]$, takvi da je

$$m = f(\underline{x}), \quad M = f(\overline{x}).$$

Teorem 1.1.2 (Srednja vrijednost) *Neka je f realna funkcija neprekidna na konačnom segmentu $[a, b]$ i diferencijabilna na otvorenom intervalu (a, b). Tada postoji barem jedna točka $\xi \in (a, b)$, takva da je*

$$f(b) - f(a) = f'(\xi)(b - a).$$

Teorem 1.1.3 (Integralna srednja vrijednost) *Neka je w nenegativna i integrabilna funkcija na segmentu $[a, b]$. Neka je f neprekidna na $[a, b]$. Tada postoji točka $\xi \in [a, b]$ takva da je*

$$\int_a^b f(x)w(x)\,dx = f(\xi) \int_a^b w(x)\,dx.$$

Jedan od najvažnijih alata u numeričkoj matematici je Taylorov teorem jer daje jednostavnu metodu aproksimacije funkcije f pomoću polinoma. Kako se vrijednost polinoma određuje korištenjem Hornerovog algoritma, dobivamo način izračunavanja vrijednosti funkcije f u danoj točki x. Da bismo mogli koristiti taj pristup u aproksimaciji funkcije, ona mora biti dovoljno glatka.

Teorem 1.1.4 (Taylorov teorem) *Neka funkcija f ima neprekidne derivacije do reda $n+1$, $n+1>0$, na segmentu $[a,b]$. Ako su $x, x_0 \in [a,b]$, tada je*

$$f(x) = p_n(x) + R_{n+1}(x), \tag{1.1.1}$$

$$p_n(x) = f(x_0) + \frac{(x-x_0)}{1!} f'(x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \cdots + \frac{(x-x_0)^n}{n!} f^{(n)}(x_0), \tag{1.1.2}$$

$$R_{n+1}(x) = \frac{1}{n!} \int_{x_0}^{x} (x-t)^n f^{(n+1)}(t)\, dt = \frac{(x-x_0)^{n+1}}{(n+1)!} f^{(n+1)}(\xi), \tag{1.1.3}$$

za neki ξ između x_0 i x.

Dokaz. Izvod relacija (1.1.1)–(1.1.3) koristi n puta parcijalno integriranje, polazeći od identiteta

$$f(x) = f(x_0) + \int_{x_0}^{x} f'(t)\, dt.$$

Drugi oblik ostatka $R_{n+1}(x)$ dobije se korištenjem teorema integralne srednje vrijednosti za funkciju $w(t) = (x-t)^n$. Puni dokaz se može naći u većini knjiga iz elementarne analize. ■

Polinom p_n se naziva Taylorov razvoj funkcije f u točki x_0. Ako je f beskonačno puta derivabilna, polinom p_n prelazi u red potencija, s općim članom $(x-x_0)^n$, a zove se Taylorov red. Uz pomoć Taylorovog reda lako se dobivaju poznate formule:

$$e^x = 1 + x + \frac{x^2}{2} + \cdots + \frac{x^n}{n!} + \frac{x^{n+1}}{(n+1)!} e^{\xi_x}, \tag{1.1.4}$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + (-1)^{n+1} \frac{x^{2n+2}}{(2n+2)!} \cos(\xi_x), \tag{1.1.5}$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots + (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!} + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \sin(\xi_x), \tag{1.1.6}$$

$$(1+x)^\alpha = 1 + \binom{\alpha}{1}x + \binom{\alpha}{2}x^2 + \cdots + \binom{\alpha}{n}x^n + \binom{\alpha}{n+1} + \frac{x^{n+1}}{(1+\xi_x)^{n+1-\alpha}}, \tag{1.1.7}$$

pri čemu je

$$\binom{\alpha}{k} = \frac{\alpha(\alpha-1)\cdots(\alpha-k+1)}{k!}, \quad k = 1, 2, 3, \ldots$$

za proizvoljni realni broj α. U svim slučajevima nepoznata točka ξ_x smještena je između 0 i x.

Sjetimo se formule za n-tu parcijalnu sumu geometrijskog reda

$$1 + x + x^2 + \cdots + x^n = \frac{1 - x^{n+1}}{1 - x}. \tag{1.1.8}$$

Prikažemo li razlomak na desnoj strani kao razliku razlomaka, te dio s potencijom x^{n+1} prebacimo na drugu stranu jednakosti, dobivamo

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots + x^n + \frac{x^{n+1}}{1-x}, \quad x \neq 1, \tag{1.1.9}$$

što je drugi, jednostavniji, zapis razvoja (1.1.7) uz $\alpha = -1$ i $-x$ umjesto x. Kad pustimo da n proizvoljno raste i ako je x iz $(-1, 1)$, formula (1.1.9) prelazi u

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots + x^n + \cdots, \quad |x| < 1. \tag{1.1.10}$$

Kad isto načinimo u formulama za e^x, $\cos(x)$ i $\sin(x)$, ne trebamo ograničiti x jer pripadni redovi konvergiraju apsolutno za svaki x.

Iako su formule (1.1.1)–(1.1.3) vrlo korisne za dobivanje razvoja funkcije f oko točke x_0, često puta računanje viših derivacija predstavlja problem zbog složenosti računanja. U takvim slučajevima, možemo se koristiti već dobivenim razvojima. Sljedeći primjeri pokazuju kako se to može načiniti.

Primjer 1.1.1 *Da bismo dobili razvoj funkcije*

$$f(x) = e^{-x^2}$$

oko 0*, možemo se koristiti razvojem (1.1.4), u koji, umjesto* x*, uvrstimo* $-x^2$*,*

$$e^{-x^2} = 1 - x^2 + \frac{x^4}{2!} - \cdots + (-1)^{n+1}\frac{x^{2n+2}}{(n+1)!}\,e^{\xi_x},$$

pri čemu je $-x^2 \leq \xi_x \leq 0$*.*

Primjer 1.1.2 *Da bismo dobili razvoj funkcije*

$$f(x) = \operatorname{arctg}(x)$$

oko 0*, možemo se koristiti razvojem (1.1.9), u koji, umjesto* x*, uvrstimo* $-t^2$*,*

$$\frac{1}{1+t^2} = 1 - t^2 + t^4 - \cdots + (-1)^n t^{2n} + (-1)^{n+1}\frac{t^{2n+2}}{1+t^2}.$$

Integriranjem po t *na intervalu* $[0, x]$*, dobivamo*

$$\operatorname{arctg}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \cdots + (-1)^n \frac{x^{2n+1}}{2n+1} + (-1)^{n+1}\int_0^x \frac{t^{2n+2}}{1+t^2}\,dt.$$

Primjena teorema o integralnoj srednjoj vrijednosti daje

$$\int_0^x \frac{t^{2n+2}}{1+t^2}\,dt = \frac{x^{2n+3}}{2n+3}\cdot\frac{1}{1+\xi_x^2}, \quad \xi_x \text{ između } 0 \text{ i } x.$$

Primjer 1.1.3 *Neka je*

$$f(x) = \int_0^1 \sin(xt)\,dt.$$

Korištenjem razvoja funkcije sin *u Taylorov red oko nule (vidi relaciju (1.1.6)), dobivamo*

$$\begin{aligned} f(x) &= \int_0^1 \left[xt - \frac{x^3t^3}{3!} + \cdots + (-1)^{n-1}\frac{(xt)^{2n-1}}{(2n-1)!} + (-1)^n\frac{(xt)^{2n+1}}{(2n+1)!}\right]dt \\ &= \sum_{j=1}^{n}(-1)^{j-1}\frac{x^{2j-1}}{(2j)!} + (-1)^n\frac{x^{2n+1}}{(2n+1)!}\int_0^1 t^{2n+1}\cos(\xi_{xt})\,dt, \end{aligned}$$

pri čemu je ξ_{xt} *između* 0 *i* xt*. Integral u ostatku se jednostavno ocijeni s* $1/(2n+2)$*, ali se može dovesti i na jednostavniji oblik. Naime, može se pokazati da je* $t \mapsto \xi_{xt}$ *neprekidna funkcija, pa se može primijeniti integralni teorem srednje vrijednosti i dobiti*

$$\int_0^1 \sin(xt)\,dt = \sum_{j=1}^{n}(-1)^{j-1}\frac{x^{2j-1}}{(2j)!} + (-1)^n\frac{x^{2n+1}}{(2n+2)!}\cos(\xi_x), \quad \xi_x \text{ između } 0 \text{ i } x.$$

Primjer 1.1.4 *Neka su* x_0*,* x_1*,* x_2 *različiti realni brojevi i* f *realna funkcija definirana na intervalu koji sadrži te točke. Veličine*

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad i \quad f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \tag{1.1.11}$$

zovu se **podijeljene razlike** *prvog i drugog reda od f. Ako je f jedanput odnosno dvaput diferencijabilna na odgovarajućem intervalu, može se pokazati da je*

$$f[x_0, x_1] = f'(\xi), \quad f[x_0, x_1, x_2] = \frac{1}{2} f''(\eta), \tag{1.1.12}$$

pri čemu je ξ između x_0 i x_1, a η između minimuma i maksimuma skupa $\{x_0, x_1, x_2\}$. Prva tvrdnja lako slijedi iz teorema o srednjoj vrijednosti. Kasnije ćemo pokazati opća svojstva podijeljenih razlika proizvoljnog reda, a time i drugu tvrdnju. Također, odmah se vidi da je $f[x_0, x_1] = f[x_1, x_0]$. Pokušajte pokazati da je $f[x_0, x_1, x_2] = f[x_i, x_j, x_k]$ za bilo koju permutaciju i, j, k niza $(0, 1, 2)$.

Neka je sada f realna funkcija dviju varijabli, $f : \mathbb{R}^2 \to \mathbb{R}$. Koristeći Kartezijev koordinatni sustav Oxy, možemo svakom paru realnih brojeva (x, y) pridružiti točno jednu točku u Kartezijevoj ravnini koju označimo s (x, y) (apscisa te točke ima vrijednost x, a ordinata ima vrijednost y). Kako vrijedi i obrat, par realnih brojeva (x, y) se poistovjećuje s točkom ravnine.

Postoji generalizacija Taylorovog teorema za $f(x, y)$ u okolini točke (x_0, y_0). Jasno, i taj teorem se može dalje generalizirati za funkcije od tri i više varijabli, ali nam to za potrebe ove knjige neće trebati.

Označimo s $L(x_0, y_0; x_1, y_1)$ skup svih točaka na pravcu koji prolazi točkama (x_0, y_0) i (x_1, y_1), a nalaze se između tih točaka (kraće kažemo da je $L(x_0, y_0; x_1, y_1)$ segment između tih dviju točaka).

Teorem 1.1.5 *Neka su (x_0, y_0) i $(x_0 + \xi, y_0 + \eta)$ dvije točke u ravnini i neka je funkcija $f : \mathbb{R}^2 \to \mathbb{R}$ $n+1$ puta neprekidno diferencijabilna u nekoj okolini segmenta $L(x_0, y_0; x_0 + \xi, y_0 + \eta)$. Tada je*

$$\begin{aligned} f(x_0 + \xi, y_0 + \eta) = f(x_0, y_0) &+ \sum_{j=1}^{n} \frac{1}{j!} \left[\xi \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y} \right]^j f(x, y) \Big|_{\substack{x=x_0 \\ y=y_0}} \\ &+ \frac{1}{(n+1)!} \left[\xi \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y} \right]^{n+1} f(x, y) \Big|_{\substack{x=x_0+\theta\xi \\ y=y_0+\theta\eta}} \end{aligned} \tag{1.1.13}$$

za neki $0 \leq \theta \leq 1$. Pritom je $(x_0 + \theta\xi, y_0 + \theta\eta)$ nepoznata točka na segmentu $L(x_0, y_0; x_0 + \xi, y_0 + \eta)$.

Dokaz. Prvo se prisjetimo značenja oznake

$$\left[\xi \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y} \right]^2 f(x, y) = \xi^2 \frac{\partial^2 f(x, y)}{\partial x^2} + 2\xi\eta \frac{\partial^2 f(x, y)}{\partial x \partial y} + \eta^2 \frac{\partial^2 f(x, y)}{\partial y^2},$$

a po analogiji s razvojem od $(x + y)^j$ definira se i

$$\left[\xi \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y} \right]^j f(x, y).$$

Indeksi oblika $x = x_0$ i $y = y_0$ označavaju da se sve prisutne parcijalne derivacije izvrednjavaju u točki (x_0, y_0).

Dokaz relacije (1.1.13) baziran je na Taylorovom teoremu za funkciju jedne varijable $F(t) = f(x_0 + t\xi, y_0 + t\eta)$, $t \in [0, 1]$. Naime, F zadovoljava uvjete teorema 1.1.4, pa je

$$F(1) = f(0) + \frac{F'(0)}{1!} + \cdots + \frac{F^{(n)}(0)}{n!} + \frac{F^{(n+1)}(\theta)}{(n+1)!}$$

za neki $0 \leq \theta \leq 1$. Uočimo da je $F(0) = f(x_0, y_0)$ i $F(1) = f(x_0 + \xi, y_0 + \eta)$. Za prvu derivaciju vrijedi

$$\begin{aligned} F'(t) &= \xi \frac{\partial f(x_0 + t\xi, y_0 + t\eta)}{\partial x} + \eta \frac{\partial f(x_0 + t\xi, y_0 + t\eta)}{\partial y} \\ &= \left[\xi \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y}\right] f(x, y) \Big|_{\substack{x = x_0 + t\xi \\ y = y_0 + t\eta}}. \end{aligned}$$

Derivacije višeg reda se dobiju na sličan način. ■

Primjer 1.1.5 *Odredimo razvoj funkcije*

$$f(x, y) = \frac{x}{y}$$

oko točke $(x_0, y_0) = (2, 1)$ *do uključivo člana određenog s* $n = 1$.

Imamo $\xi = x - 2$, $\eta = y - 1$,

$$\begin{aligned} \frac{x}{y} &= f(2,1) + \xi \frac{\partial f(2,1)}{\partial x} + \eta \frac{\partial f(2,1)}{\partial y} \\ &\qquad + \frac{1}{2}\left[\xi^2 \frac{\partial^2 f(x,y)}{\partial x^2} + 2\xi\eta \frac{\partial^2 f(x,y)}{\partial x \partial y} + \eta^2 \frac{\partial^2 f(x,y)}{\partial y^2}\right]_{\substack{x=\alpha \\ y=\beta}} \\ &= 2 + (x-2)\cdot 1 + (y-1)\cdot(-2) \\ &\qquad + \frac{1}{2}\left[(x-2)^2 \cdot 0 + 2(x-2)(y-1)\left(\frac{-1}{\beta^2}\right) + (y-1)^2 \frac{2\alpha}{\beta^3}\right] \\ &= x - 2y + 2 - \frac{1}{\beta^2}(x-2)(y-1) + 2\frac{\alpha}{\beta^3}(y-1)^2, \end{aligned}$$

gdje je točka (α, β) *na segmentu* $L(2, 1; x, y)$. *Ako je* (x, y) *blizu* $(2, 1)$, *tada je segment* $L(2, 1; x, y)$ *kratak pa je i* (α, β) *blizu* $(2, 1)$ *i vrijedi*

$$\frac{x}{y} \approx x - 2y + 2.$$

Graf funkcije $z = x - 2y + 2$ *je ravnina tangencijalna na graf funkcije* $z = x/y$ *u točki* $(x, y, z) = (2, 1, 2)$.

Matematičke tvrdnje često imaju oblik implikacije: iskaz T_1 povlači iskaz T_2. Umjesto da se dokazuje ta implikacija, kadkad je zgodnije dokazati ekvivalentnu implikaciju: iskaz $\neg T_2$ povlači iskaz $\neg T_1$, gdje $\neg T$ označava suprotni iskaz od iskaza T. Ta ekvivalentnost se logički zapisuje

$$(T_1 \Longrightarrow T_2) \quad \Longleftrightarrow \quad (\neg T_2 \Longrightarrow \neg T_1) \tag{1.1.14}$$

i zove **obrat po kontrapoziciji**.

Pretpostavljamo da su čitatelji upoznati s pojmovima kao što su vektori, matrice, norme i skalarni produkti. Nešto dublje znanje iz linearne algebre koje uključuje osnovne informacije o grupama, vektorskim prostorima, linearnim operatorima, determinantama, dobro bi došlo za lakše čitanje ove knjige. Pritom je posebno važan pojam linearne nezavisnosti vektora jer se koristi u pojmovima kao što su rang i inverz matrice ili operatora, baze, itd.

Za one koji nisu upoznati s osnovama linearne algebre, uvodimo ovdje neke oznake i pojmove da bi lakše razumjeli izreke mnogih tvrdnji koje možda nemaju veze s linearnom algebrom, ali koriste oznake odatle.

S $\mathbb{R}^n$ označavamo skup svih jednostupčanih realnih matrica (koje još zovemo realni vektori)

$$\mathbb{R}^n = \left\{ \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad x_i \in \mathbb{R} \quad \text{za sve} \quad 1 \leq i \leq n \right\}.$$

Elemente tog skupa označavamo malim slovima abecede. Ako su $x, y \in \mathbb{R}^n$ vektori s komponentama x_i, y_i, $1 \leq i \leq n$, (kraće pišemo $x = [x_i]$, $y = [y_i]$) i $\alpha \in \mathbb{R}$ realni broj (skalar), tada se vektori $x + y$ i αx definiraju formulama

$$x + y = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix} \quad \text{i} \quad \alpha x = \begin{bmatrix} \alpha x_1 \\ \vdots \\ \alpha x_n \end{bmatrix}.$$

Ako je $\alpha = 0$, tada je u zadnjem izrazu αx vektor koji ima sve komponente jednake nuli. Takav vektor se zove **nul-vektor** i označava s 0. Iz konteksta ćemo razlikovati je li 0 broj (skalar) nula ili nul-vektor.

Na isti način definira se $\mathbb{C}^n$ kao skup kompleksnih vektora koji osim realnih mogu imati i kompleksne brojeve kao komponente. Jednako kao gore definira se i zbroj kompleksnih vektora kao i umnožak kompleksnog broja (skalara) i kompleksnog vektora. Uz tako definirane operacije skupovi $\mathbb{R}^n$ i $\mathbb{C}^n$ imaju cijeli niz lijepih svojstava koje ih čine vektorskim prostorima. Ako je $z = [z_i] \in \mathbb{C}$, tada se konjugirano kompleksni vektor $\bar{z} = [\bar{z}_i]$ dobije tako da mu se svaka komponenta kompleksno konjugira.

Za realni ili kompleksni vektor $a = [a_i]$, oznaka $|a| = [\,|a_i|\,]$ označava **apsolutnu vrijednost vektora po komponentama** vektora a. Dakle, $|a|$ je vektor čije komponente su nenegativni realni brojevi, pa zato vrijede sljedeća jednostavna svojstva:

(i) $|a| = 0$ ako i samo ako je $a = 0$,

(ii) $|\alpha a| = |\alpha|\,|a|$, α je skalar, a a vektor,

(iii) $|a + b| \leq |a| + |b|$, a i b su vektori.

Slična svojstva ima i funkcija **norma** koja svakom vektoru pridružuje broj. Najpoznatije norme su **euklidska** (ili 2-norma), **norma beskonačno** i **norma jedan**. Za $a \in \mathbb{C}^n$ (ili $\mathbb{R}^n$) je

$$\begin{aligned} \|a\|_2 &= \sqrt{|a_1|^2 + |a_2|^2 + \cdots + |a_n|^2}, \\ \|a\|_\infty &= \max_{1\leq i\leq n} |a_i|, \\ \|a\|_1 &= |a_1| + |a_2| + \cdots + |a_n|. \end{aligned}$$

Malo općenitije su tzv. p-norme

$$\|a\|_p = \sqrt[p]{|a_1|^p + \cdots + |a_n|^p}, \quad 1 \leq p \leq \infty.$$

Sve te norme zadovoljavaju četiri svojstva (nenegativnost, definitnost, pozitivna homogenost i nejednakost trokuta) koja funkciju $\|\cdot\|$ čine normom,

(i) $\|a\| \geq 0$ za sve vektore a,

(ii) $\|a\| = 0$ ako i samo ako je $a = 0$, pri čemu je 0 nul-vektor,

(iii) $\|\alpha a\| = |\alpha|\,\|a\|$ za sve vektore a i skalare α,

(iv) $\|a + b\| \leq \|a\| + \|b\|$ za sve vektore a i b.

Ista svojstva koja imaju skupovi $\mathbb{R}^n$ i $\mathbb{C}^n$ zajedno s operacijama zbrajanja i množenja skalarom mogu imati i drugačiji skupovi. Npr. skup

$$C[a, b] = \{f \mid f \text{ realna i neprekidna funkcija definirana na } [a, b]\}$$

s operacijom zbrajanja funkcija:

$$(f + g)(t) = f(t) + g(t), \quad t \in [a, b]$$

i množenja funkcija realnim brojem:

$$(\alpha f)(t) = \alpha f(t), \quad t \in [a, b],$$

ima ista poželjna svojstva koja ga čine vektorskim prostorom. Tipična norma na tom prostoru je

$$\|f\|_\infty = \max_{a\le x\le b} |f(t)|.$$

Lako se provjeri da ona zadovoljava svojstva (i)–(iv) za norme.

Norme postoje i za matrice. Npr. ako je

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix},$$

onda je

$$\|A\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|.$$

Norme na matricama obično zadovoljavaju dodatno svojstvo (konzistentnost) vezano uz produkt dviju matrica (za vektore x, y produkt xy nije definiran). Npr. za normu beskonačno to svojstvo se zapisuje kao

$$\|AB\|_\infty \le \|A\|_\infty \|B\|_\infty, \quad A, B \text{ reda } n.$$

Specijalno, vrijedi i

$$\|Ax\|_\infty \le \|A\|_\infty \|x\|_\infty,$$

gdje je $x \in \mathbb{R}^n$.

2. Kratki uvod u linearnu algebru

U ovom poglavlju uvodimo osnovne pojmove i rezultate iz linearne algebre koji će se koristiti u kasnijim poglavljima. Prva dva dijela su sažeti prikaz predavanja iz linearne algebre, rađeni po internoj skripti koja se nalazi na adresi `http://www.math.hr/~hari/la.pdf`. Na toj adresi se mogu naći svi dokazi koji ovdje nisu dani. Treći i četvrti odjeljak imaju glavne tvrdnje dokazane.

Prva cjelina definira pojam vektorskog prostora i njegovih elemenata koje nazivamo vektorima u širem značenju tog pojma. Definiraju se pojmovi linearno nezavisnih vektora, baze i dimenzije vektorskog prostora, norme i skalarnog produkta, te linearnog operatora.

Lijep primjer vektorskog prostora čine matrice određenog tipa. Kvadratne matrice reda n čine još bogatiju strukturu koja se zove algebra. Stoga se kratka teorija matrica opisuje u drugom odjeljku.

Jedan od najvažnijih rezultata iz teorije matrica je tzv. singularna dekompozicija matrica. Njene primjene sežu u razne numeričke probleme, npr. problem najmanjih kvadrata, određivanje ranga matrice, nalaženje generaliziranog inverza, kompresija slika u računalnoj grafici i općenito probleme aproksimacije matrice pomoću matrica manjeg ranga.

U zadnjem odjeljku prikazuju se osnovni rezultati iz teorije vektorskih i matričnih normi, posebno tzv. Hölderovih p-normi. Te norme uključuju najčešće korištene matrične norme: spektralnu (ili 2-normu), 1-normu i ∞-normu.

2.1. Vektorski prostor

Vektorski ili linearni prostor je algebarska struktura koja se sastoji od dva skupa: X čije elemente zovemo vektori i Φ čije elemente zovemo skalari. Skup skalara je najčešće skup realnih ili kompleksnih brojeva. Pritom je važno da su na skupu Φ definirane dvije operacije, koje zovemo zbrajanje i množenje skalara i da je rezultat primjene tih operacija na skalare uvijek u Φ. Za operaciju zbrajanja moraju

vrijediti svojstva asocijativnosti, postojanja nule, postojanje suprotnog elementa (za svaki element) i komutativnosti. Za produkt trebaju vrijediti barem svojstva asocijativnosti, obostrane distributivnosti prema zbrajanju te postojanje jedinice i postojanje recipročnog skalara (osim za nulu). Takva struktura $(\Phi, +, \cdot)$ se zove tijelo, a ako vrijedi i komutativnost množenja, zove se polje.

Na skupu X su definirane operacije zbrajanja vektora $\oplus$ i množenja $\otimes$ vektora skalarom iz Φ. Pritom za operaciju $\oplus$ vrijede ista svojstva kao i za $+$ u Φ: asocijativnost, postojanja nul-vektora, postojanje suprotnog vektora i komutativnosti. Za operaciju $\otimes$ moraju vrijediti sljedeća svojstva kompatibilnosti:

(a) $\alpha \otimes (x \oplus y) = \alpha \otimes x \oplus \alpha \otimes y$ za bilo koje $x, y \in X$, $\alpha \in \Phi$
(distributivnost množenja prema zbrajanju u X),

(b) $(\alpha + \beta) \otimes x = \alpha \otimes x \oplus \beta \otimes x$ za bilo koje $x \in X$, $\alpha, \beta \in \Phi$
(distributivnost množenja prema zbrajanju u Φ),

(c) $\alpha \otimes (\beta \otimes x) = (\alpha \cdot \beta) \otimes x$ za bilo koje $x \in X$, $\alpha, \beta \in \Phi$
(kompatibilnost množenja),

(d) ako je $1 \in \Phi$ neutralni element za množenje u Φ, tada je $1 \otimes x = x$ za svako $x \in X$
(netrivijalnost množenja).

Za tako definiranu strukturu kažemo da je vektorski prostor nad tijelom (ili poljem) Φ.

U daljem tekstu Φ će uvijek biti polje realnih ($\Phi = \mathbb{R}$) ili kompleksnih ($\Phi = \mathbb{C}$) brojeva. Ako je X vektorski prostor nad poljem realnih (kompleksnih) brojeva, X se naziva realni (kompleksni) vektorski prostor.

2.1.1. Osnovna svojstva vektorskog prostora

Jednostavno je pokazati da u svakom vektorskom prostoru vrijede sljedeće tvrdnje.

- Neutralni element za zbrajanje u X je jedinstven.
- Za svaki $x \in X$ inverzni element od x s obzirom na $\oplus$ je jedinstven.
- Ako je $o \in X$ neutralni element s obzirom na $\oplus$, tada za svako $\alpha \in \Phi$ vrijedi $\alpha \otimes o = o$.
- Ako je $0 \in \Phi$ nula, tj. neutralni element s obzirom na $+$, a $o \in X$ neutralni element s obzirom na $\oplus$, tada za svako $x \in X$ vrijedi $0 \otimes x = o$.

- Ako je $1 \in \Phi$ jedinica, tj. neutralni element s obzirom na $\cdot$, a -1 njegov suprotni element u Φ s obzirom na $+$, tada za svako $x \in X$ vrijedi $(-1) \otimes x = -x$, gdje je $-x$ inverzni element od x u X s obzirom na $\oplus$.

  Na vektorskom prostoru X možemo definirati razliku vektora formulom

$$x \ominus y = x \oplus (-1) \otimes y = x \oplus (-y)$$

i lako se pokazuje da vrijedi

$$\begin{aligned} (\alpha - \beta) \otimes x &= \alpha \otimes x \ominus \beta \otimes x, \quad x \in X, \quad \alpha, \beta \in \Phi \\ \alpha \otimes (x \ominus y) &= \alpha \otimes x \ominus \beta \otimes x, \quad x, y \in X, \quad \alpha \in \Phi. \end{aligned}$$

U vektorskom prostoru smijemo vektorske jednadžbe "kratiti" i sa skalarom i vektorom. Npr. ako je

- produkt $\alpha \otimes x$ skalara α i vektora x nul-vektor i jedan od faktora nije nula, onda drugi faktor mora biti nula.
- u jednadžbi $\alpha \otimes x = \beta \otimes x$, $\alpha, \beta \in \Phi$, $x \in X$, vektor $x \neq o$, tada vrijedi skalarna jednadžba $\alpha = \beta$.
- u jednadžbi $\alpha \otimes x = \alpha \otimes y$, $\alpha \in \Phi$, $x, y \in X$, skalar $\alpha \neq 0$, tada vrijedi vektorska jednadžba $x = y$.

Potprostor vektorskog prostora $(X, \oplus, \otimes)$ je svaki podskup od X koji je uz operacije $\oplus$ i $\otimes$ i sam vektorski prostor nad istim poljem. Da bi to vrijedilo dovoljno je dokazati da je zatvoren u odnosu na operacije $\oplus$, $\otimes$. Zaista, ako je $(S, \oplus, \otimes)$ potprostor od $(X, \oplus, \otimes)$ (kraće pišemo S je potprostor od X), tada je nužno (po definiciji vektorskog prostora) zatvoren u odnosu na te operacije. Obratno, ako je podskup $S \subseteq X$ zatvoren u odnosu na operacije $\oplus$, $\otimes$, tada za njega automatski vrijede (jer je podskup od X) svi uvjeti iz definicije vektorskog prostora. Dakle, vektorski potprostor je definiran upravo onim operacijama koje su nasljeđene od prostora. Trivijalni vektorski potprostori su jednočlan skup $\{o\}$ i cijeli X. Da bi neki skup $S \subseteq X$ bio potprostor nužno je i dovoljno da su ispunjeni sljedeći uvjeti:

(a) $x \oplus y \in S$ za svaka dva vektora $x, y \in S$,

(b) $\alpha \otimes x \in S$ za sve $\alpha \in \Phi$ i $x \in S$.

Naime, to su nužni i dovoljni uvjeti za zatvorenost skupa S u odnosu na operacije $\oplus$ i $\otimes$. Lako se pokaže da su uvjeti (a) i (b) ekvivalentni uvjetu

(ab) $\alpha \otimes x \oplus \beta \otimes y \in S$ za sve $x, y \in S$ i $\alpha, \beta \in \Phi$.

Neka su $a_1, a_2, \ldots, a_p$ vektori iz $(X, \oplus, \otimes)$. Linearna kombinacija vektora $a_1, a_2, \ldots, a_p$ (ili skupa vektora $\{a_1, a_2, \ldots, a_p\}$) je svaki vektor y oblika

$$y = \alpha_1 \otimes a_1 \oplus \alpha_2 \otimes a_2 \oplus \cdots \oplus \alpha_p \otimes a_p,$$

gdje su $\alpha_1, \alpha_2 \ldots, \alpha_p$ skalari iz Φ. Najmanji vektorski potprostor koji sadrži sve vektore $a_1, \ldots, a_p$ je potprostor $(L(a_1, \ldots, a_p), \oplus, \otimes)$ kojemu su elementi linearne kombinacije skupa $\{a_1, \ldots, a_p\}$,

$$L(a_1, \ldots, a_p) = \{\alpha_1 \otimes a_1 \oplus \cdots \oplus \alpha_p \otimes a_p, \quad \alpha_1, \ldots, \alpha_p \in \Phi\}.$$

Osim oznake $L(a_1, \ldots, a_p)$ još se koristi i oznaka $\text{span}\{a_1, \ldots, a_p\}$.

Ako je $Y \subseteq X$ potprostor i $z \in X$ proizvoljni vektor, skup

$$z + Y = \{x \in X \mid x = z + y,\ y \in Y\}$$

naziva se **linearna mnogostrukost** (ili višestrukost). Kad je $z \in Y$, onda je $z + Y = Y$, a kad je $Y = \{0\}$, vrijedi $z + Y = \{z\}$.

U daljem tekstu, umjesto oznaka $\oplus$, $\ominus$ i $\otimes$ koristit ćemo jednostavnije oznake $+$, $-$ i $\cdot$, pri čemu ćemo $\cdot$ izostavljati. Također, kad god su operacije $\oplus$ i $\otimes$ poznate, tj. jasne iz konteksta, umjesto $(X, \oplus, \otimes)$ pisat ćemo X. Ako je iz konteksta jasno da se radi o vektorskom prostoru ili potprostoru, atribut "vektorski" ili "linearni" katkad ćemo izostavljati.

2.1.2. Norma u vektorskom prostoru

U vektorskom prostoru se definira **norma** ili duljina vektora kao funkcija $\| \ \| : X \to \mathbb{R}$ za koju vrijede sljedeća svojstva

(i) $\|x\| \geq 0$, za svaki $x \in X$, (nenegativnost)

(ii) $\|\alpha x\| = |\alpha| \, \|x\|$, za sve $\alpha \in \Phi$, $x \in X$, (homogenost)

(iii) $\|x + y\| \leq \|x\| + \|y\|$, za sve $x, y \in X$, (nejednakost trokuta)

(iv) $\|x\| = 0$, ako i samo ako je $x = 0$ (definitnost).

Vektorski prostor na kojem je definirana norma zove se **normirani vektorski prostor**. Funkcija $\| \ \| : X \to \Phi$ koja zadovoljava samo svojstva (i), (ii) i (iii) zove se **polunorma** ili seminorma.

U normiranom vektorskom prostoru, norma sume ili razlike vektora može se ocijeniti odozdo, jer za svaku polunormu pa zato i normu vrijedi

$$\|x \pm y\| \geq | \, \|x\| - \|y\| \, |. \tag{2.1.1}$$

Primjer 2.1.1 *Neka je V_n ($\mathbb{C}_n$) skup uređenih n-torki realnih (kompleksnih) brojeva. Operaciju zbrajanja n-torki i množenja n-torki skalarom definiramo po komponentama: ako su*

$$x = (x_1, \ldots, x_n) \quad i \quad y = (y_1, \ldots, y_n)$$

dvije n-torke iz V_n (ili $\mathbb{C}_n$) i $\alpha \in \mathbb{R}$ (ili $\alpha \in \mathbb{C}$), tada su operacije zbrajanja i množenja skalarom definirane s

$$x + y = (x_1 + y_1, \ldots, x_n + y_n) \quad i \quad \alpha \cdot x = (\alpha x_1, \ldots, \alpha x_n).$$

Lako se pokaže da je $(V_n, +, \cdot)$ realni, a $(\mathbb{C}_n, +, \cdot)$ kompleksni vektorski prostor. U tim vektorskim prostorima možemo definirati sljedeće funkcije

$$\begin{aligned}
\|x\|_1 &= |x_1| + |x_2| + \cdots + |x_n|, \\
\|x\|_2 &= \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2}, \\
\|x\|_\infty &= \max\{|x_1|, |x_2|, \ldots, |x_n|\}, \\
\|x\|_p &= \sqrt[p]{|x_1|^p + |x_2|^p + \cdots + |x_n|^p}, \quad 1 \leq p \leq \infty.
\end{aligned}$$

Lako je provjeriti da su funkcije $\|x\|_1$ i $\|x\|_\infty$ norme. Za opću, tzv. Hölderovu normu $\|x\|_p$ dokaz je složeniji. Hölderova 2-norma $\|x\|_2$ se još zove euklidska norma.

Dvije norme $\| \ \|_\alpha$ i $\| \ \|_\beta$ vektorskog prostora X su ekvivalentne, ako postoje pozitivni realni brojevi c_1 i c_2 takvi da vrijedi

$$c_1\|x\|_\alpha \leq \|x\|_\beta \leq c_2\|x\|_\alpha.$$

Lako se pokaže da su norme $\| \ \|_2$, $\| \ \|_1$ i $\| \ \|_\infty$ u vektorskom prostoru V_n međusobno ekvivalentne. Pokušajte naći odgovarajuće konstante c_1 i c_2 za svaki par normi. Može se pokazati da su u vektorskim prostorima V_n i $\mathbb{C}_n$ sve norme međusobno ekvivalentne.

2.1.3. Skalarni produkt u vektorskom prostoru

Da bismo definirali ortogonalnost vektora kao i kut između vektora, potrebno je u vektorski prostor uvesti još jednu funkciju.

Neka je $(X, +, \cdot)$ vektorski prostor nad Φ, gdje je $\Phi = \mathbb{C}$. Funkcija $(\cdot \mid \cdot) : X \times X \to \Phi$ zove se skalarni produkt ako vrijedi

(i) $(x \mid x) \geq 0$ za sve $x \in X$,

(ii) $(x \mid x) = 0$ ako i samo ako $x = 0$,

(iii) $(x \mid y) = \overline{(y \mid x)}$ za sve $x, y \in X$,

(iv) $(\alpha x \mid y) = \alpha(x \mid y)$ za sve $x, y \in X$ i svaki $\alpha \in \Phi$,

(v) $(x + y \mid z) = (x \mid z) + (y \mid z)$ za sve $x, y, z \in X$.

Ako je $\Phi = \mathbb{R}$, tada u svojstvu (iii) treba ispustiti operaciju kompleksnog konjugiranja. Vektorski prostor na kojem je definiran skalarni produkt zove se **unitarni vektorski prostor**.

Svaki unitarni vektorski prostor, također je i normiran vektorski prostor, jer je formulom

$$\|x\| = \sqrt{(x \mid x)}, \tag{2.1.2}$$

definirana funkcija za koju se može pokazati da je norma. To je tzv. pridružena ili inducirana norma. U svakom unitarnom vektorskom prostoru za pridruženu normu vrijedi relacija paralelograma,

$$\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2. \tag{2.1.3}$$

Također, vrijedi i Cauchy–Schwarzova nejednakost koja povezuje skalarni produkt i induciranu normu

$$|(x \mid y)| \leq \|x\| \, \|y\|. \tag{2.1.4}$$

Postavlja se pitanje da li vrijedi i obratno, tj. možemo li svakoj normi $\| \ \|$ jednog normiranog vektorskog prostora pridružiti neki skalarni produkt $(\cdot \mid \cdot)$, tako da vrijedi

$$\|x\| = \sqrt{(x \mid x)}?$$

Odgovor je ne, osim ako norma ne zadovoljava relaciju paralelograma, a tada je skalarni produkt određen formulom

$$(x \mid y) = \frac{1}{4} \left(\|x + y\|^2 - \|x - y\|^2 \right),$$

za realni normirani prostor i formulom

$$(x \mid y) = \frac{1}{4} \left(\|x + y\|^2 - \|x - y\|^2 + i\|x + iy\|^2 - i\|x - iy\|^2 \right)$$

za kompleksni normirani prostor.

Nejednakost (2.1.4) pokazuje da vrijedi

$$-1 \leq \frac{(x \mid y)}{\|x\| \, \|y\|} \leq 1,$$

pa relacijom

$$\cos\theta = \frac{(x \mid y)}{\|x\| \, \|y\|}$$

možemo definirati kut θ između vektora x i y. Za kompleksni unitarni prostor, u brojniku moramo uzeti apsolutnu vrijednost skalarnog produkta jer je $(x \mid y)$ kompleksni broj. Stoga je kut u kompleksnom prostoru definiran samo za segment $[0, \pi/2]$.

Za dva vektora kažemo da su ortogonalni ako je $\cos\theta$, tj. ako je $(x \mid y) = 0$.

2.1.4. Dimenzija vektorskog prostora

Svaki skup vektora vektorskog prostora razapinje neki potprostor, koji može biti i cijeli vektorski prostor. Poželjno je izdvojiti iz danog skupa vektora jedan njegov podskup koji razapinje isti potprostor, ali koji je u nekom smislu minimalan. Takav podskup ima određeno svojstvo minimalnosti koje se definira na sljedeći način.

Ako je $\mathcal{S}$ podskup vektorskog prostora X, tada je $\mathcal{M}$ **minimalni razapinjući podskup** od $\mathcal{S}$, ako zadovoljava sljedeća dva svojstva:

1. $L(\mathcal{M}) = L(\mathcal{S})$,
2. Niti jedan pravi podskup skupa $\mathcal{M}$ ne razapinje $L(\mathcal{S})$.

$\mathcal{M}$ se još zove i **minimalni razapinjući skup** od $L(\mathcal{S})$.

Npr. ako je $\mathcal{S} = \{a_1, a_2, \ldots, a_p\}$, tada je $\mathcal{M} = \{u_1, u_2, \ldots, u_r\}$, $r \leq p$, gdje je svaki u_i neki od vektora iz $\mathcal{S}$. Sada se prvi uvjet zapisuje kao

$$L(u_1, \ldots, u_r) = L(a_1, \ldots, a_p).$$

Uočimo da u zadnjoj jednakosti poredak vektora $a_1, \ldots, a_p$ (odnosno $u_1, \ldots, u_r$) nije važan, jer je $L(a_1, \ldots, a_p)$ tek kraća oznaka za $L(\mathcal{S}) = L(\{a_1, \ldots, a_p\})$.

Linearno nezavisni vektori

Pokazuje se da elementi minimalnog razapinjućeg skupa imaju svojstvo linearne nezavisnosti. Neka su $x_1, \ldots, x_r$ elementi vektorskog prostora X. Ako jednadžba

$$\alpha_1 x_1 + \cdots + \alpha_r x_r = 0$$

u nepoznanicama $\alpha_1, \ldots, \alpha_r$ ima samo trivijalno rješenje $\alpha_1 = \cdots = \alpha_r = 0$, tada se vektori $x_1, \ldots, x_r$ nazivaju **linearno nezavisnim**. Vektori koji nisu linearno nezavisni nazivaju se **linearno zavisni**. Skup vektora $\{x_1, \ldots, x_r\}$ je linearno nezavisan (zavisan) ako su vektori $x_1, \ldots, x_r$ takvi.

Skup linearno nezavisnih vektora $\{u_1, \ldots, u_r\}$ ne sadrži nul-vektor. Ako je skup $\{u_1, \ldots, u_r\}$ linearno nezavisan, tada je i svaki njegov podskup linerano nezavisan. Ako je skup $\{u_1, \ldots, u_r\}$ linearno zavisan, tada je svaki njegov nadskup također linearno zavisan.

Za dani konačni skup vektora, minimalni razapinjući skup nije jedinstveno određen. Može ih biti mnogo. Ali ako pođemo ne od konačnog skupa, već od vektorskog potprostora, tada za njega uvijek imamo beskonačno mnogo minimalnih razapinjućih skupova. Ipak svi oni imaju jednu važnu zajedničku karakteristiku: broj vektora u svakom ovisi samo o potprostoru kojeg razapinju. Dakle, ako su

$\{u_1, u_2, \ldots, u_r\}$ i $\{v_1, v_2, \ldots, v_s\}$ dva minimalna razapinjuća skupa istog potprostora, onda vrijedi $r = s$. Broj vektora minimalnog razapinjućeg skupa nekog potprostora je invarijanta (konstanta) tog potprostora. Tako dolazimo do pojma dimenzije vektorskog prostora.

Ako za vektorski prostor X postoji konačni minimalni razapinjući skup vektora $\{u_1, u_2, \ldots, u_r\}$ on se naziva **konačno-dimenzionalnim vektorskim prostorom**. Pritom se broj r naziva **dimenzijom** prostora X i označava s

$$r = \dim X.$$

Dimenzija trivijalnog prostora $\{0\}$ je nula. Ako za vektorski prostor ne postoji konačni minimalni razapinjući skup vektora on se naziva **beskonačno-dimenzionalni vektorski prostor**. Dimenzija linearne višestrukosti jednaka je dimenziji vektorskog potprostora koji ju gradi.

Primjer 2.1.2 *Vektorski prostor V_n je konačno dimenzionalan, dimenzije n. To se vidi iz relacije*

$$V_n = L(e_1, e_2, \ldots, e_n),$$

gdje je

$$e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T$$

s jedinicom na i-tom mjestu, za svako i. Pritom je $\{e_1, e_2, \ldots, e_n\}$ minimalni razapinjući skup za V_n. Lako se pokaže da je skup vektora $\{e_1, \ldots, e_n\}$ linearno nezavisan.

2.1.5. Baza vektorskog prostora

Neka je X vektorski prostor. Uređeni skup vektora $\mathcal{B}$ iz X zove se **baza** vektorskog prostora X ako zadovoljava sljedeća dva uvjeta:

(i) $\mathcal{B}$ je linearno nezavisan skup,

(ii) $L(\mathcal{B}) = X$.

Iz prvog svojstva minimalnog razapinjućeg skupa $\mathcal{M}$ za X (tj. $L(\mathcal{M}) = X$) i svojstva linearne nezavisnosti od $\mathcal{M}$, zaključujemo da je svaki uređeni minimalni razapinjući skup i baza od X, a vrijedi i obrat. Zbog uređenosti skupa, bazu katkad zapisujemo kao $(u_1, \ldots, u_n)$, gdje su u_i bazni vektori.

Ako su $u_1, u_2, \ldots, u_r$ linearno nezavisni vektori u n-dimenzionalnom vektorskom prostoru X, tada je $r \leq n$, tj. sigurno ih je manje ili jednako od dimenzije prosotra X.

Svaki se niz linearno nezavisnih vektora $u_1, u_2, \ldots, u_r$ u X može nadopuniti s $n - r$ vektora do baze od X. Odatle odmah slijedi da ako je $\mathcal{X}$ potprostor n-dimenzionalnog vektorskog prostora X i ako je $\mathcal{X}$ dimenzije n, onda je $\mathcal{X} = X$.

U n-dimenzionalnom vektorskom prostoru X, sljedeće tvrdnje su ekvivalentne:

1. $(u_1, \ldots, u_n)$ je baza od X,
2. $\{u_1, \ldots, u_n\}$ je linearno nezavisan skup,
3. $L(u_1, \ldots, u_n) = X$.

Za svaki niz vektora $a_1, a_2, \ldots, a_p$, vrijedi

$$\dim L(a_1, a_2, \ldots, a_p) \leq p.$$

Važnost baze dolazi od jedinstvenosti prikaza vektora po baznim vektorima.

Propozicija 2.1.1 *Neka je X vektorski prostor i $\mathcal{B}$ baza u X. Tada se svaki vektor $x \in X$ na jedinstven način može prikazati kao linearna kombinacija vektora baze $\mathcal{B}$.*

Dokaz. Neka je $\mathcal{B} = (u_1, \ldots, u_n)$. Prema svojstvu baze $L(\mathcal{B}) = X$, svaki vektor x dopušta prikaz

$$x = \alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_n u_n.$$

Pretpostavimo da postoji još jedan prikaz

$$x = \beta_1 u_1 + \beta_2 u_2 + \cdots + \beta_n u_n.$$

Izjednačavanjem $\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_n u_n = \beta_1 u_1 + \beta_2 u_2 + \cdots + \beta_n u_n$, dobivamo

$$(\alpha_1 - \beta_1)u_1 + (\alpha_2 - \beta_2)u_2 + \cdots + (\alpha_n - \beta_n)u_n = 0.$$

Po drugom svojstvu baze, vektori u_i su linearno nezavisni. Stoga zaključujemo da vrijedi

$$\alpha_1 = \beta_1, \quad \alpha_2 = \beta_2, \quad \ldots, \quad \alpha_n = \beta_n,$$

tj. prikaz svakog vektora x po baznim vektorima je jedinstven. ■

Najzanimljivije su one baze koje se sastoje od ortonormiranih vektora. Svi vektori takvih baza su norme jedan i ortogonalni su jedan prema drugome. Tzv. Gram–Schmidtovim postupkom može se u unitarnom prostoru od svake baze dobiti ortonormirana baza.

2.1.6. Linearni operatori

Neka su X i Y vektorski prostori. Preslikavanje $\mathcal{A} : X \to Y$ je **linearni operator** ako vrijedi

(i) $\mathcal{A}(x + y) = \mathcal{A}(x) + \mathcal{A}(y)$ za sve $x, y \in X$, (aditivnost)

(ii) $\mathcal{A}(\alpha x) = \alpha \mathcal{A}(x)$ za sve $x \in X$ i sve skalare α (homogenost).

Ako je $\mathcal{A}$ bijekcija, onda ga zovemo i **izomorfizam** vektorskih prostora X i Y.

Iz svojstva (ii) vidi se da X i Y moraju biti definirani nad istim poljem. Ako su X i Y realni vektorski prostori, onda α može biti samo realan broj. Ako su X i Y kompleksni vektorski prostori, onda α može biti i realan i kompleksan broj. Kad god ćemo koristiti linearni operator, implicitno ćemo pretpostavljati da su polazni i dolazni vektorski prostori definirani nad istim poljem.

Iz svojstva (ii) odmah slijedi, uzimanjem $\alpha = 0$, $\mathcal{A}(0) = 0$. Uvjeti (i) i (ii) mogu se zamijeniti jednim uvjetom

(l) $\mathcal{A}(\alpha x + \beta y) = \alpha\mathcal{A}(x) + \beta\mathcal{A}(y)$ za sve $x, y \in X$ i sve skalare α, β,

koji zovemo **linearnost**. Pokažimo prvo da aditivnost i homogenost povlače linearnost.

$$\begin{aligned}\mathcal{A}(\alpha x + \beta y) &= \mathcal{A}(\alpha x) + \mathcal{A}(\beta y) \quad \text{(jer vrijedi (i))}\\ &= \alpha\mathcal{A}(x) + \beta\mathcal{A}(y) \quad \text{(jer vrijedi (ii)).}\end{aligned}$$

Pokažimo da linearnost preslikavanja povlači aditivnost i homogenost. Doista, uvrštavajući u (l) $\alpha = \beta = 1$ dobivamo (i). Uzimajući u (l) $\beta = 0$, dobivamo

$$\mathcal{A}(\alpha x) = \mathcal{A}(\alpha x + 0y) = \alpha\mathcal{A}(x) + 0\mathcal{A}(y) = \alpha\mathcal{A}(x) + 0 = \alpha\mathcal{A}(x),$$

za sve x i sve skalare α. Time je pokazano da (l) povlači (ii). Dakle su uvjeti (i) i (ii) ekvivalentni uvjetu (l) (tj. znače isto). Iz uvjeta (l) se vidi da je linearni operator ono preslikavanje koje linearni spoj vektora iz X prevodi u isti linearni spoj slika u Y.

Linearni operator prevodi linearnu kombinaciju proizvoljnog broja vektora u istu linearnu kombinaciju slika

$$\mathcal{A}(\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k) = \alpha_1\mathcal{A}(x_1) + \alpha_2\mathcal{A}(x_2) + \cdots + \alpha_k\mathcal{A}(x_k).$$

Dokaz je tek vježba za korištenje matematičke indukcije. Uz svaki linearni operator $\mathcal{A} : X \to Y$, vezana su dva važna potprostora,

$$\mathcal{N}(\mathcal{A}) = \{x \in X \mid \mathcal{A}(x) = 0\} \quad \text{i} \quad \mathcal{R}(\mathcal{A}) = \{\mathcal{A}(x) \mid x \in X\}.$$

$\mathcal{N}(\mathcal{A})$ i $\mathcal{R}(\mathcal{A})$ su vektorski potprostori. $\mathcal{N}(\mathcal{A})$ je **nul-potprostor** (ili **jezgra**), a $\mathcal{R}(\mathcal{A})$ je **slika** (ili **područje vrijednosti**) operatora $\mathcal{A}$. Dimenzija jezgre se zove **defekt**, a dimenzija slike **rang** operatora $\mathcal{A}$.

S $\mathcal{L}(X, Y)$ označavamo skup svih linearnih preslikavanja vektorskog prostora X u vektorski prostor Y.

Posebno važnu klasu linearnih operatora čine izomorfizmi. To su linearni operatori koji su bijekcije. Ako je $\mathcal{A} \in \mathcal{L}(X, Y)$ izomorfizam, tada postoji inverzno preslikavanje $\mathcal{A}^{-1}$ koje je također linearni operator (tj. izomorfizam). Izomorfizam

također preslikava bazu polaznog prostora u bazu dolaznog prostora, odnosno svaki skup linearno nezavisnih vektora u linearno nezavisan skup slika.

Kako razlikovati izomorfizam od općeg linearnog operatora? Koje svojstvo ga izdvaja? Sljedeća tvrdnja daje jedno takvo svojstvo.

Neka je $\dim(X) = \dim(Y)$ i $\mathcal{A} \in \mathcal{L}(X, Y)$. $\mathcal{A}$ je izomorfizam ako i samo ako je $\mathcal{N}(\mathcal{A}) = \{0\}$.

Izomorfizam je relacija ekvivalencije (ili klasifikacije) u skupu vektorskih prostora.

2.2. Matrice

Matrica je matematički objekt koji se sastoji od (realnih ili kompleksnih) brojeva koji su raspoređeni u retke i stupce. Zapisuje se u obliku pravokutne sheme, a brojeve od kojih se sastoji zovemo **elementima** matrice. Matrica A sa m redaka, n stupaca i s elementima a_{ij} zapisuje se kao

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} = [\, a_{ij} \,].$$

Takvu matricu zovemo $m \times n$ (čitaj: m puta n) matrica ili matrica tipa (reda, dimenzije) $m \times n$. Pritom je $[\, a_{ij} \,]$ tek kraći zapis za pravoktnu shemu iz gornje relacije. Ako pišemo $A = [\, a_{ij} \,]$, mislimo na cijelu shemu brojeva koji čine matricu A, dok a_{ij} (ili $[A]_{ij}$ ili $(A)_{ij}$), označava samo element koji se nalazi na presjeku $i-$tog retka i $j-$tog stupca matrice A. Pritom niz brojeva $a_{i1}, a_{i2}, a_{i3}, \ldots, a_{in}$ zovemo i-ti redak, a niz brojeva $a_{1j}, a_{2j}, \ldots, a_{mj}$ poredanih jedan ispod drugog, j-ti stupac matrice A. Ako vrijedi $m = n$, kažemo da je A **kvadratna** matrica reda n.

Matricu sa samo jednim retkom zovemo **matrica redak** ili jednoretčana matrica, a matricu sa samo jednim stupcem zovemo **matrica stupac** ili jednostupčana matrica. Jednostupčane i jednoretčane matrice kraće zovemo vektorima. Matricu, čiji su elementi realni brojevi, zovemo **realna matrica**, a matricu, čiji su elementi kako realni tako i kompleksni brojevi, zovemo **kompleksna matrica**. Matrica A je jednaka matrici B ako imaju isti broj redaka i isti broj stupaca i za njihove elemente vrijedi $a_{ij} = b_{ij}$, za sve i i j. Dakle, svaki element jedne matrice jednak je odgovarajućem elementu druge matrice.

Praktično je imati oznaku za skup svih m puta n matrica. S $\mathbb{R}^{m\times n}$ ($\mathbb{C}^{m\times n}$) označavamo skup svih realnih (kompleksnih) $m \times n$ matrica. Ukratko ćemo opisati kako se u $\mathbb{R}^{m\times n}$ uvode operacije zbrajanja i množenja realnim brojem, koje će

načiniti $(\mathbb{R}^{m\times n}, +, \cdot)$ vektorskim prostorom. Zatim ćemo definirati množenje (ulančanih) matrica. Elemente matrica A, B, C, ... označit ćemo s odgovarajućim malim slovima a_{ij}, b_{ij}, c_{ij}, ...

2.2.1. Zbrajanje matrica i množenje matrica skalarom

Neka su $A, B \in \mathbb{R}^{m\times n}$. Matricu $C \in \mathbb{R}^{m\times n}$ s elementima

$$c_{ij} = a_{ij} + b_{ij}, \quad 1 \le i \le m, \quad 1 \le j \le n$$

zovemo **zbrojem** ili sumom matrica A i B i pišemo $C = A + B$. Zbrajanje je definirano za svaki par matrica iz $\mathbb{R}^{m\times n}$ i rezulat je uvijek u $\mathbb{R}^{m\times n}$, pa je $\mathbb{R}^{m\times n}$ zatvoren u odnosu na tu operaciju. Zbrajanje u $\mathbb{R}^{m\times n}$ ima ova svojstva:

(a) $A + (B + C) = (A + B) + C$, (asocijativnost)

(b) $A + B = B + A$. (komutativnost)

(c) Postoji matrica $O \in \mathbb{R}^{m\times n}$ (tzv. neutralni element za zbrajanje) sa svojstvom da je $A + O = A$ za svaku matricu $A \in \mathbb{R}^{m\times n}$. Svi elementi matrice O jednaki su nuli.

(d) Za svaki $A \in \mathbb{R}^{m\times n}$ postoji jedna i samo jedna matrica koju označavamo s $-A$ (inverzni element), takva da vrijedi $A + (-A) = O$. Ako je $A = [\, a_{ij} \,]$, onda je $-A = [\, -a_{ij} \,]$.

Iz navedenih svojstava možemo zaključiti, da skup matrica $\mathbb{R}^{m\times n}$ zajedno s operacijom zbrajanja čini **komutativnu aditivnu grupu**, jer je ona definirana upravo tim svojstvima.

Sljedeća operacija koju možemo jednostavno definirati je množenje matrica sa skalarom. Ako je $A \in \mathbb{R}^{m\times n}$ i $c \in \mathbb{R}$, matricu $B \in \mathbb{R}^{m\times n}$ s elementima

$$b_{ij} = ca_{ij}, \quad 1 \le i \le m, \quad 1 \le j \le n$$

zovemo **umnožak** ili produkt matrice A sa skalarom c i označavamo $B = cA$. Jasno je da za svaki realni skalar c i svaku matricu $A \in \mathbb{R}^{m\times n}$, mora $B = cA$ opet biti u $\mathbb{R}^{m\times n}$. Za proizvoljne $A, B \in \mathbb{R}^{m\times n}$ i $\alpha, \beta, c \in \mathbb{R}$ vrijedi

(a) $c(A + B) = cA + cB$ (distributivnost množenja prema zbrajanju u $\mathbb{R}^{m\times n}$)

(b) $(\alpha + \beta)A = \alpha A + \beta A$ (distributivnost množenja prema zbrajanju u $\mathbb{R}$)

(c) $(\alpha\beta)A = \alpha(\beta A)$ (kompatibilnost množenja)

(d) $1A = A$ (netrivijalnost množenja).

Za svaki par prirodnih brojeva m i n dobili smo strukturu $(\mathbb{R}^{m\times n}, +, \cdot)$ za koju iz svojstava zbrajanja i množenja matrica skalarom zaključujemo da čini realni vektorski prostor.

Vektorski prostor jednostupčanih matrica $\mathbb{R}^{n\times 1}$ označava se s $\mathbb{R}^n$.

2.2.2. Množenje matrica

Ako su $a_1, \ldots, a_n \in \mathbb{R}^m$ neki vektori, tada notacija $A = [a_1, \ldots, a_n]$ znači da je matrica A tako izgrađena da su $a_1, \ldots, a_n$ njeni stupci u redoslijedu u kojem su napisani. Slično, ako su $a'_1, \ldots, a'_m \in \mathbb{R}^{1\times n}$ jednoretčane matrice (još se zovu: vektori retci), tada oznaka

$$A = \begin{bmatrix} a'_1 \\ a'_2 \\ \vdots \\ a'_m \end{bmatrix}$$

ukazuje da je $A \in \mathbb{R}^{m\times n}$ tako građena da su joj $a'_1, \ldots, a'_m$ retci, u redosljedu koji je naznačen. Oznaka $A = [a_1, \ldots, a_n]$ se još zove particija matrice po stupcima, a ona druga particija po retcima.

Neka je $A \in \mathbb{R}^{m\times n}$ i $B \in \mathbb{R}^{n\times p}$. **Umnožak** ili produkt matrica A i B je matrica $C = A \cdot B \in \mathbb{R}^{m\times p}$ kojoj su elementi određeni formulom

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \quad \text{za sve} \quad 1 \le i \le m, \quad 1 \le j \le p.$$

Produkt matrica A i B definiran je samo onda kad je broj stupaca matrice A jednak broju redaka matrice B. Za takve dvije matrice kažemo da su ulančane. Operacija množenja matrica $\cdot$ može se sagledati kao preslikavanje koje uređenom paru matrica određenih dimenzija pridružuje matricu također određenih dimenzija, $\cdot : \mathbb{R}^{m\times n} \times \mathbb{R}^{n\times p} \to \mathbb{R}^{m\times p}$. Produkt matrica obično se piše (kao i produkt skalara) bez znaka množenja između faktora, dakle AB. Za množenje matrica vrijede svojstva

(a) $A(B + C) = AB + AC$,

(b) $(B + C)A = BA + CA$,

(c) $A(BC) = (AB)C$,

(d) $\alpha(AB) = (\alpha A)B = A(\alpha B)$,

koja zovemo lijeva i desna distributivnost, asocijativnost množenja i kompatibilnost množenja matrica s množenjem skalarom. Uočimo da ne vrijedi zakon komutativnosti. Lako je naći primjere matrica kad produkt AB postoji, a BA ne postoji (ili obratno), ili kad oba produkata postoje, ali su različitih dimenzija, ili kad oba produkta postoje i istih su dimenzija, ali ne vrijedi $AB = BA$.

Jedinična matrica i dijagonalna matrica

Važnu klasu matrica čine **dijagonalne matrice**. Najpoznatiji primjer dijagonalne matrice je **jedinična** matrica

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Ona je jedinični element skupa $\mathbb{R}^{n\times n}$ s obzirom na množenje, jer vrijedi

$$AI = IA = A$$

za svako $A \in \mathbb{R}^{n\times n}$. Dakle, ima istu ulogu kod množenja matrica kao što je ima broj 1 kod množenja realnih brojeva. Malo općenitije, vrijede relacije: $IA = A$ za svako $A \in \mathbb{R}^{n\times p}$ i $AI = A$ za svako $A \in \mathbb{R}^{p\times n}$. Zbog toga što je I reda n koristi još i oznaka I_n.

Malo općenitija, dijagonalna matrica je ona kod koje su svi izvandijagonalni (to su oni koji nisu dijagonalni) elementi jednaki nuli. Dijagonalnu matricu kojoj su dijagonalni elementi $\alpha_1, \alpha_2, \ldots, \alpha_n$ označavamo s $\operatorname{diag}(\alpha_1, \alpha_2, \ldots, \alpha_n)$, tj.

$$\operatorname{diag}(\alpha_1, \alpha_2, \ldots, \alpha_n) = \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 & 0 \\ 0 & \alpha_2 & \ddots & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \ddots & \alpha_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & \alpha_n \end{bmatrix}.$$

Odmah vidimo da produkt dviju netrivijalnih dijagonalnih matrica npr.,

$$\operatorname{diag}(1, 0, 2) \cdot \operatorname{diag}(0, -1, 0) = \operatorname{diag}(0, 0, 0),$$

može biti **nul-matrica**. Nul-matrica (koja ima sve elemente nula) kraće se označava s O ili 0 i, također, pripada klasi dijagonalnih matrica.

Dijagonalne matrice čine vektorski prostor koji je zatvoren i u odnosu na matrično množenje. Neka je $D = \operatorname{diag}(\alpha_1, \ldots, \alpha_n)$ i $A \in \mathbb{R}^{n\times p}$. Tada je i-ti redak od DA jednak i-tom retku od A pomnoženom s α_i, pri čemu je $1 \leq i \leq n$. Slično, ako je $A \in \mathbb{R}^{p\times n}$, tada je i-ti stupac od AD jednak i-tom stupcu od A pomnoženom s α_i. Ako su dijagonalni elementi od D pozitivni, operacija DA (AD) naziva se **skaliranje** redaka (stupaca). Dijagonalne matrice koje dobivamo množenjem jedinične matrice skalarom (tj. one oblika αI) nazivamo **skalarne** matrice. Ako je $D = \alpha I$, onda je $DA = \alpha A$, a, također, i $AD = \alpha A$, čim su dimenzije matrica takve da su produkti definirani.

Potencije matrice, nilpotentna matrica

Potencije kvadratne matrice A definiraju se induktivno:

$$A^0 = I, \quad A^{r+1} = A\,A^r \quad \text{za} \quad r \geq 0.$$

Lako se pokaže da vrijedi

$$A^p A^q = A^q A^p = A^{p+q}$$

za sve nenegativne cijele brojeve p i q. Stoga je dobro definiran **matrični polinom**

$$p(A) = \alpha_k A^k + \alpha_{k-1} A^{k-1} + \cdots + \alpha_1 A + \alpha_0 I,$$

pri čemu su $\alpha_0, \ldots, \alpha_k$ realni brojevi.

Monoženje matrica katkad daje iznenađujuće rezultate. Npr. za

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{je} \quad A^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{pa je} \quad A^3 = O.$$

Postoje i $n \times n$ matrice za koje je prvih $n-1$ potencija različito od O, ali vrijedi $A^n = O$. Ako je $A \neq O$ i $A^k = O$ za neko $k \in \mathbb{N}$, tada se A naziva **nilpotentna** matrica.

Sljedeće iznenađujuće svojstvo matričnog množenja daje matrica

$$J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \text{jer je} \quad J^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = -I.$$

U aritmetici ne postoji realan broj čiji kvadrat je -1. U matričnoj algebri postoji realna matrica, čiji kvadrat je jednak $-I$.

Transponiranje matrica

Postoji još jedna vrlo korisna operacija na matricama. Naziva se operacijom transponiranja.

Neka je $A \in \mathbb{R}^{m\times n}$. Matrica $A^T \in \mathbb{R}^{n\times m}$ se naziva **transponirana** matrica matrici A, ako je svaki redak od A^T jednak odgovarajućem stupcu matrice A. Prema tome, transponiranu matricu dobivamo tako da stupce (retke) matrice zamijenimo njenim retcima (stupcima). Ako je

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad \text{onda je} \quad A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}.$$

Kraće to zapisujemo: ako je $A = [\, a_{ij} \,]$, onda je $A^T = [\, a_{ji} \,]$. Operacija transponiranja je unarna operacija koja matrici pridružuje njoj transponiranu matricu.

Glavna svojstva operacije transponiranja su

(a) $(A^T)^T = A$,

(b) $(A + B)^T = A^T + B^T$,

(c) $(AB)^T = B^T A^T$.

Primijetimo da su u svojstvu (c) matrice A i B zamijenile poredak.

Zadatak 2.2.1 *Neka su $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$. Dokažite sljedeće tvrdnje.*

(i) Ako je $B = [b_1, \ldots, b_p]$ stupčana particija matrice B, tada je

$$AB = [Ab_1, \ldots, Ab_p] \quad \text{stupčana particija matrice } AB.$$

(ii) Za retčane particije matrica A i AB vrijedi implikacija

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \quad \Longrightarrow \quad AB = \begin{bmatrix} a_1^T B \\ \vdots \\ a_m^T B \end{bmatrix}.$$

(iii) Ako je $A = [a_1, \ldots, a_n]$ stupčana particija matrice A i $x \in \mathbb{R}^n$, tada je

$$Ax = x_1 a_1 + \cdots + x_n a_n,$$

gdje su $x_1, \ldots, x_n$ komponente vektora x.

(iv) Ako je $y \in \mathbb{R}^n$, $y^T = [y_1, \ldots, y_n]$ i ako je

$$B = \begin{bmatrix} b_1^T \\ \vdots \\ b_n^T \end{bmatrix} \tag{2.2.1}$$

retčana particija od B, tada je

$$y^T B = y_1 b_1^T + \cdots + y_n b_n^T.$$

(v) Ako je $A = [a_1, \ldots, a_n]$ stupčana particija od A i ako je (2.2.1) retčana particija od B, tada je

$$AB = a_1 b_1^T + \ldots + a_n b_n^T.$$

Pritom je svaka matrica $a_i b_i^T \in \mathbb{R}^{m\times p}$.

Za svaki par prirodnih brojeva m i n, struktura $(\mathbb{R}^{m\times n}, +, \cdot)$ je zatvorena u odnosu na operacije $+$ i $\cdot$. Množenje matrica iz $\mathbb{R}^{m\times n}$ definirano je tek ako je $m = n$. Također je važno da je skup $\mathbb{R}^{n\times n}$ zatvoren u odnosu na matrično množenje. Označimo li privremeno operaciju množenja matrica s $\times$, možemo pisati $(\mathbb{R}^{n\times n}, +, \cdot, \times)$, pri čemu je $(\mathbb{R}^{n\times n}, +, \cdot)$ vektorski prostor, a množenje $\times$ zadovoljava svojstva (a)–(d) s početka ovog odjeljka. Takva struktura se zove **algebra**.

Kako u $\mathbb{R}^{n\times n}$ postoji i neutralni element I za množenje $\times$, kaže se da je $(\mathbb{R}^{n\times n}, +, \cdot, \times)$ algebra s jedinicom. Dakle, **skup matrica reda n čini algebru s jedinicom**. Ona općenito nije komutativna. Na njoj je definirana i unarna operacija transponiranja.

Promotrimo još skup M_R svih realnih matrica, $M_R = \bigcup_{m,n\in\mathbb{N}} \mathbb{R}^{m\times n}$. Neka $\times$ označava operaciju množenja matrica, a T operaciju transponiranja matrica. Skup M_R je sigurno zatvoren u odnosu na operaciju množenja matrice skalarom. Isto tako je zatvoren u odnosu na operaciju transponiranja. Međutim, ostale dvije operacije nisu definirane za svaki par matrica iz M_R. Ipak, postoji lijepo svojstvo skupa M_R, da je zatvoren s obzirom na te dvije operacije uvijek kad je rezultat tih dviju matrica definiran.

2.2.3. Kompleksne matrice

S $\mathbb{C}^{m\times n}$ označili smo skup kompleksnih $m \times n$ matrica. U taj skup je uključen i skup realnih matrica $\mathbb{R}^{m\times n}$, pa je skup $\mathbb{C}^{m\times n}$ veći od $\mathbb{R}^{m\times n}$.

Zbroj dviju matrica $A, B \in \mathbb{C}^{m\times n}$ je definiran na isti način kao i u $\mathbb{R}^{m\times n}$, pri čemu je operacija $+$ na nivou matričnih elemenata zbrajanja kompleksnih brojeva. Lako se pokaže da je $(\mathbb{C}^{m\times n}, +)$ komutativna aditivna grupa. Neutralni element je matrica $O \in \mathbb{C}^{m\times n}$ čiji su elementi same nule. Suprotni element od $A \in \mathbb{C}^{m\times n}$ je matrica $-A = [\,-a_{ij}\,]$. Uočimo da je O također u $\mathbb{R}^{m\times n}$. Štoviše, $(\mathbb{R}^{m\times n}, +)$ je netrivijalna podgrupa od $(\mathbb{C}^{m\times n}, +)$.

Kod množenja matrica iz $\mathbb{C}^{m\times n}$ skalarom, moramo dozvoliti množenje kompleksnim brojevima. Neka je $A \in \mathbb{C}^{m\times n}$ i $\omega \in \mathbb{C}$. Tada je $B = \omega \cdot A$ (ili kraće $B = \omega A$) ako je $b_{ij} = \omega a_{ij}$ za sve i, j. Ova operacija ima ista svojstva kao i operacija množenja realnim skalarom u $\mathbb{R}^{m\times n}$. Stoga $\mathbb{C}^{m\times n}$ uz operaciju zbrajanja i množenja skalarom iz $\mathbb{C}$ postaje vektorski prostor (nad poljem kompleksnih brojeva), koji označavamo s $(\mathbb{C}^{m\times n}, +, \cdot)$. Uočimo da $(\mathbb{R}^{m\times n}, +, \cdot)$ nije vektorski potprostor od $(\mathbb{C}^{m\times n}, +, \cdot)$ jer ti vektorski prostori nisu definirani nad istim poljem.

Elemente od $\mathbb{C}^{n\times 1}$ opet zovemo matrice stupci, jednostupčane matrice ili vektori. Oznaku $\mathbb{C}^{n\times 1}$ zamjenjujemo oznakom $\mathbb{C}^n$, a za sam vektorski prostor tih vektora koristimo oznaku $(\mathbb{C}^n, +, \cdot)$.

Operacija množenja kompleksnih matrica definirana je samo za parove ulančа-

nih matrica. Množenje je definirano na isti način kao kod realnih matrica. Neutralni element s obzirom na matrično množenje je opet identiteta, tj. jedinična matrica I (koja je i u $\mathbb{R}^{n\times n}$ i u $\mathbb{C}^{n\times n}$, ako je reda n). Množenje matrica ima ista svojstva kao i kod realnih matrica. Naime, operacija tranponiranja je definirana za svaku kompleksnu matricu na isti način kao i u slučaju realnih matrica. Time smo zapravo pokazali da je $(\mathbb{C}^{n\times n}, +, \cdot, \times)$ algebra s jedinicom, gdje smo načas matrično množenje označili s $\times$.

Glavna razlika u definicijama vezanim uz algebre $\mathbb{R}^{n\times n}$ i $\mathbb{C}^{n\times n}$ dolazi u momentu kad se želi u njima definirati skalarni produkt odnosno norma. Skalarni produkt dva realna vektora stupca x i y se zapisuje u obliku $(x \mid y) = x^T y$. Ako su vektori stupci kompleksni taj zapis prelazi u

$$(x \mid y) = x_1\bar{y}_1 + \cdots + x_n\bar{y}_n,$$

pa zato u $\mathbb{C}^n$ ne možemo pisati $(x \mid y) = x^T y$, $x, y \in \mathbb{C}^n$. Da bismo ipak $(x \mid y)$ napisali pomoću operacije matričog množenja moramo uvesti operaciju kompleksnog konjugiranja i kompleksnog transponiranja.

Ako je $z = z_1 + iz_2$ kompleksni broj, tada je $\bar{z} = z_1 - iz_2$ kompleksno konjugirani broj. Na sličan način definiramo i kompleksno konjugiranu matricu, a onda i kompleksno transponiranu matricu. Evo definicija.

Neka je $Z \in \mathbb{C}^{m\times n}$ i $Z = Z_1 + iZ_2$, pri čemu su $Z_1, Z_2 \in \mathbb{R}^{m\times n}$. Tada se $Z_1 - iZ_2$ zove **kompleksno konjugirana** matrica matrici Z i označava s $\overline{Z}$. **Kompleksno transponirana** ili **hermitski adjungirana** matrica matrici Z je matrica $Z^* = Z_1^T - iZ_2^T$.

Uočimo da su obje operacije definirane za svaku kompleksnu matricu i pritom je $\overline{Z} \in \mathbb{C}^{m\times n}$ i $Z^* \in \mathbb{C}^{n\times m}$ ako je $Z \in \mathbb{C}^{m\times n}$. Operacije kompleksnog konjugiranja i kompleksnog transponiranja imaju sljedeća svojstva.

Neka su $Z, W \in \mathbb{C}^{m\times n}$ i $\alpha \in \mathbb{C}$. Tada je

(i) $Z^* = (\overline{Z})^T = \overline{Z^T}, \qquad \overline{\overline{Z}} = Z, \qquad (Z^*)^* = Z,$

(ii) $\overline{Z+W} = \overline{Z} + \overline{W}, \qquad (Z+W)^* = Z^* + W^*,$

(iii) $\overline{\alpha Z} = \bar{\alpha}\overline{Z}, \qquad (\alpha Z)^* = \bar{\alpha} Z^*,$

(iv) $\overline{ZW} = \overline{Z}\,\overline{W}, \quad (ZW)^* = W^* Z^*.$

Sve tvrdnje izlaze iz osnovnih svojstava konjugiranja kompleksnih brojeva: $\bar{\bar{z}} = z$, $\overline{\alpha+\beta} = \bar{\alpha} + \bar{\beta}$, $\overline{\alpha\beta} = \bar{\alpha}\bar{\beta}$ i osnovnih svojstava operacije transponiranja.

Uočimo da vrijedi

$$(x \mid y) = x_1\bar{y}_1 + \cdots + x_n\bar{y}_n = \bar{y}_1 x_1 + \cdots + \bar{y}_n x_n = \bar{y}^T x = y^* x,$$

pa smo uspjeli skalarni produkt prikazati pomoću produkta matrica. Zato za normu vektora vrijedi

$$\|x\| = \sqrt{|x_1|^2 + \cdots + |x_n|^2} = \sqrt{x^* x}.$$

Uočite da tvrdnje (i)–(v) iz zadatka 2.2.1 vrijede i za kompleksne matrice.

O ostalim vektorskim i matričnim normama više ćemo reći u jednom od sljedećih odjeljaka.

2.2.4. Rang matrice

Svaku matricu $A \in \mathbb{R}^{m\times n}$ možemo promatrati kao niz njenih vektora-redaka

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix}, \quad a_i \in \mathbb{R}^n,$$

ili kao niz njenih vektora stupaca

$$A = [b_1, \ldots, b_n], \quad b_i \in \mathbb{R}^m.$$

Skup vektora-redaka (vektora-stupaca) matrice A razapinje vektorski potprostor od $\mathbb{R}^{1\times n}$ ($\mathbb{R}^m$), koji zovemo **retčani** (**stupčani**) potprostor od A. Važno svojstvo tih potprostora je da imaju istu dimenziju. Broj linearno nezavisnih vektora redaka matrice A naziva se retčani rang matrice A. Broj linearno nezavisnih vektora stupaca matrice A naziva se stupčani rang matrice A. Kako se radi o istom broju izbacuje se pridjev retčani ili stupčani, pa se broj linearno nezavisnih vektora redaka (stupaca) matrice A naziva **rang** matrice A i označava s $r(A)$ ili $\operatorname{rang}(A)$. Matrica A je punog (stupčanog ili retčanog) ranga, ako je $r(A)$ jednak broju stupaca ili redaka matrice A.

Primjer 2.2.1 *Ako su vektori stupci $x_1, x_2, \ldots, x_k$ iz $\mathbb{R}^n$ linearno nezavisni, tada je za svaki izbor od k realnih skalara $\alpha_1, \ldots, \alpha_k$ koji nisu svi jednaki nuli,*

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k \neq 0.$$

Ako iz vektora x_j načinimo matricu $X = [x_1, x_2, \ldots, x_k]$, a iz skalara α_j načinimo vektor stupac $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_k]^T$, tada linearnu kombinaciju

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k$$

možemo napisati kao produkt matrice X i vektora α. Stoga uvjet linearne nezavisnosti vektora x_j ima drugi zapis: $X\alpha \neq 0$ čim je $\alpha \neq 0$. To pokazuje da je jezgra matrice X nul-vektor, pa teorem o rangu i defektu daje $r(X) = k$, tj. X je punog ranga. U terminima linearnih operatora, uvjet $X\alpha \neq 0$ čim je $\alpha \neq 0$ kaže da je linearni operator $\alpha \mapsto X\alpha$ injekcija.

Uz svaku matricu $A \in \mathbb{R}^{m\times n}$ vezana su dva važna potprostora: **slika** i **jezgra** matrice. Skup

$$\mathcal{R}(A) = \{Ax \mid x \in \mathbb{R}^n\}$$

se zove slika matrice $A \in \mathbb{R}^{m\times n}$ u $\mathbb{R}^m$. Lako se pokaže da je $\mathcal{R}(A)$ potprostor od $\mathbb{R}^m$.

Za svako $A = [b_1, \ldots, b_n] \in \mathbb{R}^{m\times n}$ vrijedi $\mathcal{R}(A) = L(b_1, \ldots, b_n)$, pa je $\mathcal{R}(A)$ stupčani potprostor od A. Stoga je dimenzija od $\mathcal{R}(A)$ jednaka $r(A)$.

Skup rješenja jednadžbe $Ax = 0$ čini vektorski potprostor koji se zove **jezgra** ili **nul-potprostor** od A. Označimo taj potprostor s

$$\mathcal{N}(A) = \{x \in \mathbb{R}^n \mid Ax = 0\}.$$

Dimenzija od $\mathcal{N}(A)$ se naziva **defekt** matrice A i označava s $d(A)$. Postavlja se pitanje kolika je dimenzija jezgre? Teorem o rangu i defektu kaže da je da $\mathcal{N}(A) \subseteq \mathbb{R}^n$ ima dimenziju $n - r(A)$, tj. da je $r(A) + d(A) = n$.

2.2.5. Sustav linearnih jednadžbi i inverz matrice

Jedan od najvažnijih problema linearne algebre jest rješavanje sustava linearnih jednadžbi

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\quad \vdots \qquad\qquad \vdots \qquad\quad &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n. \end{aligned} \tag{2.2.2}$$

Uvođenjem matrice sustava $A \in \mathbb{R}^{m\times n}$, vektora rješenja $x \in \mathbb{R}^n$ i vektora desne strane sustava $b \in \mathbb{R}^m$,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix},$$

sustav (2.2.2) prelazi u matrični problem

$$Ax = b.$$

Proširena matrica sustava (2.2.2) dobije se tako da matricu napišemo pomoću stupčane blok-particije i dodamo vektor b kao $(n+1)$-vi stupac,

$$\widetilde{A} = [A, b] = [a_1, \ldots, a_n, b] \quad \text{gdje su } a_1, \ldots, a_n \text{ stupci od } A.$$

Nužne i dovoljne uvjete za rješivost sustava $Ax = b$ daje

Teorem 2.2.1 (Kronecker–Cappeli) *Neka je $A \in \mathbb{R}^{m\times n}$ i $b \in \mathbb{R}^m$. Sustav $Ax = b$ ima rješenje ako i samo ako vrijedi $r(A) = r([A, b])$. U tom slučaju skup svih rješenja sustava $Ax = b$ čini linearnu mnogostrukost u $\mathbb{R}^n$ dimenzije $n - r(A)$, koja je oblika $x^{(P)}+\mathcal{N}(A)$ gdje je $\mathcal{N}(A)$ nul-potprostor matrice, a $x^{(P)}$ je bilo koje rješenje sustava.*

Dokaz. Ako postoji rješenje $x = [x_1, \ldots, x_n]^T$ sustava $Ax = b$, tada produkt Ax možemo zapisati pomoću stupaca a_j, $1 \leq j \leq n$ matrice A, (vidi zadatak 2.2.1(iii)), pa vrijedi: $x_1a_1 + \cdots + x_na_n = b$. Stoga je b linearna kombinacija vektora a_j, pa je $b \in L(a_1, \ldots, a_n)$. To znači da je $L(a_1, \ldots, a_n) = L(a_1, \ldots, a_n, b)$. Zadnja jednakost se može zapisati kao $\mathcal{R}(A) = \mathcal{R}([A, b])$. Uzimanjem dimenzija potprostora na obje strane, odmah se dobiva jednakost ranga polazne i proširene matrice.

Drugi smjer polazi od uvjeta $r(A) = r([A, b])$, koji povlači $L(a_1, \ldots, a_n) = L(a_1, \ldots, a_n, b)$, pa je $b \in L(a_1, \ldots, a_n)$. Stoga je b neka linearna kombinacija stupaca, pa je $b = x_1b_1 + \cdots + x_nb_n$ za neke skalare $x_1, \ldots, x_n$. Ako tim skalarima definiramo vektor $x = [x_1, \ldots, x_n]^T$, tada vrijedi $Ax = b$, pa imamo rješenje sustava. ■

2.2.6. Lijevi i desni inverz, regularne i singularne matrice

Prvo poopćenje Kronecker–Cappelijeva teorema kaže da matrična jednadžba $AX = B$ ima rješenje ako i samo ako je $r(A) = r([A, B])$, gdje smo proširenu matricu definirali pomoću svih stupaca od B. U specijalnom slučaju, kad je B jedinična matrica, imamo problem $AX = I$. Svako rješenje te jednadžbe zove se desni inverz od A. Slično, svako rješenje matrične jednadžbe $XA = I$ zove se lijevi inverz od A. Može se pokazati da matrica $A \in \mathbb{R}^{m\times n}$ ima barem jedan desni inverz ako i samo vrijedi $r(A) = m \leq n$, odnosno barem jedan lijevi inverz ako i samo ako vrijedi $r(A) = n \leq m$. Štoviše, može se pokazati da su sljedeće tvrdnje ekvivalentne:

(i) $m = n$ i $r(A) = n$

(ii) A ima barem jedan lijevi i barem jedan desni inverz, te

(iii) A ima jedinstveni lijevi inverz Y, jedinstveni desni inverz X i vrijedi $X = Y$.

Posebno, ako je $A \in \mathbb{R}^{n\times n}$ i $r(A) = n$, zaključujemo da postoji samo jedan lijevi inverz i samo jedan desni inverz i da su oni jednaki. Prirodno je tu matricu zvati jednostavno **inverz** matrice A i označiti je s A^{-1}. Matrice za koje postoji inverz obično se nazivaju **invertibilne**, **regularne** ili **nesingularne**. Kvadratne matrice koje nisu regularne nazivaju se singularne.

Matrica $A \in \mathbb{R}^{n\times n}$ je singularna ako i samo ako postoji netrivijalan vektor $x \in \mathbb{R}^n$ takav da je $Ax = 0$. Doista, $A \in \mathbb{R}^{n\times n}$ je singularna, ako i samo ako je $r(A) < n$ odnosno ako i samo ako je defekt$(A) = n - r(A) \geq 1$. Dakle, ako i

samo ako je dimenzija nul-potprostora $\mathcal{N}(A)$ barem jedan, tj. ako i samo ako postoji $x \in \mathcal{N}(A)$, $x \neq 0$, što je i trebalo dokazati.

Obrat po kontrapoziciji implicira da je matrica A nesingularna ako i samo ako za svaki $x \neq 0$ vrijedi $Ax \neq 0$. Drugim načinom to vidimo ovako: $r(A) = n$ ako i samo ako je $\mathcal{N}(A) = \{0\}$, tj. ako i samo ako $x \neq 0$ povlači $Ax \neq 0$.

Dakle, za regularne matrice je preslikavanje $x \mapsto Ax$ injekcija, a zbog

$$\dim(\mathcal{R}(A)) = r(A) = n - d(A) = n,$$

ono je i bijekcija skupa $\mathbb{R}^n$ na sebe. Skup regularnih matrica ima posebnu važnost u primjenama, jer uz operaciju matričnog množenja on postaje grupa. Ako su $A_1, A_2, \ldots, A_k$ regularne matrice, korištenjem matričnog množenja lako pokazujemo da je

$$(A_1 A_2 \cdots A_k)^{-1} = A_k^{-1} \cdots A_2^{-1} A_1^{-1}.$$

Ako su S i T regularne matrice i vrijedi $B = SAT$, kažemo da su matrice A i B **ekvivalentne**. Ako je veza između A i B, $B = S^{-1}AS$, kažemo da su A i B **slične** matrice. Konačno, ako vrijedi $B = S^T AS$ (za kompleksne matrice $B = S^* AS$) kažemo da su A i B **kongruentne**.

2.2.7. Specijalne klase matrica

U ovom dijelu, ako nije izričito drugačije naznačeno, podrazumijevamo da se radi o kvadratnoj matrici iz skupa $\mathbb{R}^{n\times n}$.

Simetrične i antisimetrične matrice

Matrica A je **simetrična** ako vrijedi $A^T = A$. Matrica A je **antisimetrična** ako vrijedi $A^T = -A$. Za proizvoljnu kvadratnu matricu X, matrica

$$X_s = \frac{1}{2}(X + X^T)$$

naziva se **simetrični dio** matrice X, a matrica

$$X_a = \frac{1}{2}(X - X^T)$$

antisimetrični dio matrice X. Vrlo je jednostavno pokazati, da je za svaku matricu X, X_s uvijek simetrična, a X_a uvijek antisimetrična matrica i vrijedi $X = X_s + X_a$. Kod simetričnih (antisimetričnih) matrica je antisimetrični (simetrični) dio jednak nul-matrici. Čak štoviše, ovakav rastav matrice X na simetrični i antisimetrični dio je jedinstven, tj. ako vrijedi $X = U + V$, $U^T = U$, $V^T = -V$, tada je

$$U = \frac{1}{2}(X + X^T) \quad \text{i} \quad V = \frac{1}{2}(X - X^T),$$

tj. $U = X_s$, $V = X_a$. Ako je A simetrična i B je kongruentna s A, tada je i B simetrična.

Trag matrice i kvadratna forma

Trag matrice je jednostavna skalarna funkcija na vektorskom prostoru kvadratnih matrica reda n, a ima korisna svojstva. Trag matrice A je suma njezinih dijagonalnih elemenata:

$$\operatorname{tr}(A) = \sum_{i=1}^{n} a_{ii}.$$

Funkcija tr ima sljedeća svojstva:

(i) $\operatorname{tr}(A + B) = \operatorname{tr}(A) + \operatorname{tr}(B)$,

(ii) $\operatorname{tr}(cA) = c \operatorname{tr}(A)$,

(iii) $\operatorname{tr}(A^T) = \operatorname{tr}(A)$,

(iv) $\operatorname{tr}(AB) = \operatorname{tr}(BA)$,

(v) $\operatorname{tr}(ABC) = \operatorname{tr}(BCA) = \operatorname{tr}(CAB)$.

Prva dva svojstva pokazuju da je trag linearna funkcija pa se još naziva linearni funkcional. Neosjetljivost traga na operaciju transponiranja je jasna jer se dijagonala matrice ne mijenja transponiranjem matrice. Invarijantnost traga u odnosu na komutaciju u produktu matrica je važno svojstvo koje trag čini privlačnim kako u teoretskim razmatranjima tako i u praktičnim numeričkim primjenama.

Na vektorskom prostoru $\mathbb{R}^n$ možemo definirati i kvadratični funkcional $q_A : \mathbb{R}^n \to \mathbb{R}$ formulom: ako je $A \in \mathbb{R}^{n\times n}$ simetrična matrica, onda funkciju q_A,

$$q_A(x) = x^T A x, \quad x \in \mathbb{R}^n,$$

nazivamo **kvadratna forma**. Iako je $q_A(x)$ dobro definiran za proizvoljnu kvadratnu matricu A, on ima lijepa svojstva ako je A simetrična matrica.

Pozitivno definitne matrice

Za simetričnu matricu A kažemo da je **pozitivno semidefinitna**, ako vrijedi

$$x^T A x \geq 0 \quad \text{za svaki } x \in \mathbb{R}^n.$$

Ako vrijedi

$$x^T A x > 0 \quad \text{za svaki } x \in \mathbb{R}^n, \quad x \neq 0,$$

onda kažemo da je A **pozitivno definitna** matrica. Dakle, za simetričnu pozitivno semidefinitnu (pozitivno definitnu) matricu A vrijedi $q_A(x) \geq 0$ za svako x ($q_A(x) >$

0 za svako $x \neq 0$). Može se pokazati da je svaka pozitivno definitna simetrična matrica punog ranga, tj. da je regularna matrica.

Za $B \in \mathbb{R}^{m\times n}$ može se pokazati,

(i) matrice $B^TB \in \mathbb{R}^{n\times n}$ i $BB^T \in \mathbb{R}^{m\times m}$ su pozitivno semidefinitne,

(ii) ako je B punog stupčanog ranga, onda je $B^TB \in \mathbb{R}^{n\times n}$ pozitivno definitna matrica,

(iii) ako je B punog retčanog ranga, onda je $BB^T \in \mathbb{R}^{m\times m}$ pozitivno definitna matrica.

Može se pokazati, da za svaku pozitivno semidefinitnu matricu A postoji kvadratna matrica B, takva da je $A = B^TB$. Pritom je B regularna ako i samo ako je takva A. Matrica B općenito nije jedinstvena, ali uz neke uvjete jest. Matrica B se može odabrati gornje ili donje trokutastom i u tom slučaju je B jedinstveno određena uvjetom regularnosti matrice A i pozitivnošću dijagonalnih elemenata od B.

Ortogonalne matrice

Promotrimo sada matrice Q koje istovremeno zadovoljavaju jednadžbe

$$Q^TQ = I, \quad QQ^T = I.$$

Iz prve jednadžbe slijedi da je Q^T lijevi inverz od Q, a iz druge da je Q^T desni inverz od Q. Stoga zaključujemo da je Q nužno kvadratna i regularna. Pritom je lijevi inverz isto što i desni inverz i to je baš inverz matrice. Dakle je Q^T inverz od Q.

Kvadratna matrica Q koja ima svojstvo $Q^TQ = I$ zove se **ortogonalna matrica**.

Ova definicija je dovoljna da se zaključi kako vrijedi i relacija $QQ^T = I$. Doista, $Q^TQ = I$ kaže da Q ima lijevi inverz pa je $r(Q) = n$, a kako je kvadratna vrijedi $m = n = r(A)$, pa je regularna, tj. lijevi inverz je inverz. Stoga vrijedi i $QQ^T = I$. Na sličan način se pokazuje da $QQ^T = I$ i uvjet $m = n$ daje $Q^TQ = I$, pa je za ortogonalnost, uz kvadratičnost, dovoljno tražiti uvjet $Q^TQ = I$ ili $Q^TQ = I$.

Za ortogoinalne matrice Q vrijedi $\|Qx\| = \|x\|$, tj. **ortogonalne matrice čuvaju euklidsku duljinu vektora**. Ortogonalne matrice su jedine kvadratne matrice koje posjeduju to svojstvo.

Prisjetimo se definicije kuta između dvaju vektora iz dijela o vektorskim prostorima:

$$\cos\theta = \frac{(x \mid y)}{\|x\|\,\|y\|}, \quad x \neq 0, \quad y \neq 0, \quad 0 \leq \theta \leq \pi.$$

Ako je Q ortogonalna matrica, onda je kut između vektora Qx i Qy jednak kutu između vektora x i y, tj. **ortogonalne matrice čuvaju kut među vektorima**. Doista, vrijedi

$$\frac{(Qx \mid Qy)}{\|Qx\| \, \|Qy\|} = \frac{x^T Q^T Q y}{\|x\| \, \|y\|} = \frac{x^T y}{\|x\| \, \|y\|} = \frac{(x \mid y)}{\|x\| \, \|y\|}.$$

Budući da se kut između dvaju vektora uvijek nalazi u intervalu $[0, \pi]$, tvrdnja je dokazana.

Posebno su značajne ortogonalne matrice tipa

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

kojima su stupci rotirani kanonski vektori e_1 i e_2 rotirani za kut θ (u pozitivnom smjeru) oko ishodišta, u ravnini određenoj vektorima e_1 i e_2. Zato se matrice tog oblika zovu matrice rotacije u $\mathbb{R}^2$ za kut θ.

Neka su

$$Q = [\, q_1 \quad q_2 \quad \cdots \quad q_n \,] \quad \text{i} \quad Q = \begin{bmatrix} \tilde{q}_1^T \\ \tilde{q}_2^T \\ \vdots \\ \tilde{q}_n^T \end{bmatrix}$$

stupčana i retčana particija kvadratne matrice Q. Tada je Q ortogonalna ako i samo vrijedi

$$(q_i \mid q_j) = q_i^T q_j = \begin{cases} 1 & i = j, \\ 0 & i \neq j \end{cases} \quad \text{i} \quad (\tilde{q}_i \mid \tilde{q}_j) = \tilde{q}_i^T \tilde{q}_j = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases}$$

tj. ako i samo ako ima ortonormirane retke i stupce.

Doista, $Q^T Q = I$ je ekvivalentno s $[\, Q^T Q \,]_{ij} = \delta_{ij}$, gdje je δ_{ij} Kroneckerova delta. Jer je

$$[\, Q^T Q \,]_{ij} = e_i^T Q^T Q e_j = (Qe_i)^T (Qe_j) = q_i^T q_j, \quad i, j \in \{1, \ldots, n\},$$

$Q^T Q = I$ je ekvivalentno s $q_i^T q_j = \delta_{ij}$. Drugim riječima $Q^T Q = I$ je ekvivalentno s tvrdnjom da Q ima ortonormirane stupce.

Na sličan način se pokazuje da je $QQ^T = I$ ekvivalentno s $\tilde{q}_i^T \tilde{q}_j = \delta_{ij}$, odnosno s tvrdnjom da Q ima ortonormirane retke.

Zaključujemo da Q zadovoljava obje relacije $Q^T Q = I$ i $QQ^T = I$ ako i samo ako ima ortonormirane stupce i retke.

Matrice permutacije

Sljedeća klasa ortogonalnih matrica je u bliskoj vezi s permutacijama skupa $S_n = \{1, 2, \ldots, n\}$. Skup svih permutacija na S_n označimo s Π_n. Ako, kao binarnu operaciju na Π_n, definiramo kompoziciju permutacija $\circ$, tada $(\Pi_n, \circ)$ postaje multiplikativna grupa.

Matrica permutacije je matrica kojoj su elementi nule ili jedinice i koja u svakom retku i u svakom stupcu ima točno jednu jedinicu.

Iz definicije odmah slijedi da je svaka matrica permutacije kvadratna. Skup svih matrica permutacija reda n označit ćemo s $\mathbf{P}_n$. Kako svaka matrica permutacije ima ortonormirane stupce i retke, $\mathbf{P}_n$ je podskup skupa ortogonalnih matrica. Specijalno, matrice iz $\mathbf{P}_n$ su regularne.

Svakoj permutaciji $p \in \Pi_n$ možemo pridružiti matricu permutacije P formulom

$$Pe_i = e_{p(i)}, \quad 1 \leq i \leq n,$$

gdje su e_i stupci jedinične matrice reda n. Tom relacijom definirano je preslikavanje $\mathcal{J} : \Pi_n \to \mathbf{P}_n$, takvo da je $\mathcal{J}(p) = P$.

Npr., permutaciju koja niz brojeva $1, 2, 3, 4, 5$ preslikava u niz $5, 3, 1, 2, 4$ označimo s

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 1 & 2 & 4 \end{pmatrix}.$$

Njena inverzna permutacija je

$$p^{-1} = \begin{pmatrix} 5 & 3 & 1 & 2 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix},$$

a zapisana na uobičajeni način (elementi koje permutiramo u rastućem poretku)

$$p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 2 & 5 & 1 \end{pmatrix},$$

pa je

$$\mathcal{J}(p) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{J}(p^{-1}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Skup $\mathbf{P}_n$ zajedno s operacijom množenja matrica čini grupu. Preslikavanje $\mathcal{J}$ je tzv. **izomorfizam grupa** $(\Pi_n, \circ)$ i $(\mathbf{P}_n, \cdot)$, jer vrijedi

$$\mathcal{J}(p \circ q) = \mathcal{J}(p)\mathcal{J}(q) \quad \text{za sve } p, q \in \Pi_n.$$

Koju ulogu u matričnom računu imaju matrice permutacije? Neka za matrice $P, Q \in \mathbf{P}_n$ vrijedi $P = \mathcal{J}(p)$ i $Q = \mathcal{J}(q)$. Promotrimo elemente matrice $A' = P^T AQ$. Za element a'_{ij} matrice A' vrijedi

$$\begin{aligned} a'_{ij} &= e_i^T A' e_j = e_i^T (P^T AQ) e_j = (e_i^T P^T) A (Q e_j) = (P e_i)^T A (Q e_j) \\ &= e_{p(i)}^T A e_{q(j)} = a_{p(i)q(j)}. \end{aligned}$$

Specijalno, ako je $Q = I$, i-ti redak od $P^T A$ je $p(i)$-ti redak od A. Dakle, retci od $P^T A$ su (permutacijom p) ispermutirani retci od A. Ako je $P = I$, j-ti stupac od AQ je $q(j)$-ti stupac od A. Dakle, stupci od AQ su (permutacijom q) ispermutirani stupci od A. Ako je $P = Q$, tada vrijedi $a'_{ij} = a_{p(i)p(j)}$, pa je dijagonala od A', gledana kao niz brojeva, ispermutirana (permutacijom p) dijagonala od A.

Napomenimo još da simetrične, antisimetrične i ortogonalne matrice pripadaju široj klasi tzv. **normalnih** matrica koje su karakterizirane uvjetom $AA^T = A^T A$.

Hermitske, antihermitske i unitarne matrice

U skupu kompleksnih matrica, uloge simetrične, antisimetrične i ortogonalne matrice imaju **hermitske**, **antihermitske** i **unitarne** matrice. Njihove su definicije: $A^* = A$ za hermitsku, $A^* = -A$ za antihermitsku i $U^*U = I$ za unitarnu matricu. Većina lijepih svojstava simetričnih, antisimetričnih i ortogonalnih matrica prenosi se na hermitske, antihermitske i unitarne matrice. Dijagonalni elementi hermitske matrice su realni, dok su kod antihermitske imaginarni (a ne nule kao kod antisimetrične matrice). Trag je na kompleksnim matricama definiran na isti način kao i na realnim matricama. Međutim, kvadratna forma ima oblik x^*Ax, pa je pomoću nje definirana pozitivno definitna i semidefinitna matrica. Unitarne matrice čuvaju euklidsku normu vektora i kuteve među vektorima. Kompleksne normalne matrice definirane su uvjetom $A^*A = AA^*$ i one uključuju klase hermitskih, antihermitskih i unitarnih matrica.

Determinanta

U definiciji determinante se pojavljuje broj inverzija u permutaciji skupa S_n. **Inverzija** u permutaciji p je svaki par $(p(i), p(j))$ za koji vrijedi $p(i) > p(j)$ kad je $i < j$. $I(p)$ je ukupni **broj inverzija** u permutaciji p. Permutacija p je **parna** (**neparna**) ako je $I(p)$ paran (neparan) broj. **Parnost** je funkcija $\pi : \Pi_n \to \{-1, 1\}$ definirana s $\pi(p) = (-1)^{I(p)}$. Parnost se može definirati i pomoću algebarskog izraza. Za funkciju parnosti vrijedi

$$\pi(p) = \prod_{1 \le i < j \le n} \frac{p(j) - p(i)}{j - i}, \quad n \ge 2.$$

Determinanta matrice $A = [\, a_{ij} \,] \in \mathbb{R}^{n\times n}$ ili $\mathbb{C}^{n\times n}$ je skalar

$$\det(A) = \sum_{p\in\Pi_n} \pi(p) a_{1p(1)} a_{2p(2)} a_{3p(3)} \cdots a_{np(n)}.$$

Determinanta $\det(A)$ je reda n ako je A reda n. Umjesto $\det(A)$ još se koristi oznaka $\det A$, $|A|$ ili

$$\begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \cdots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix}.$$

Slično kao kod matrice, govorit ćemo o elementima, retcima, stupcima i dijagonali determinante. Uočimo da se u svakom članu sume nalazi po jedan element iz svakog stupca i po jedan element iz svakog retka matrice. Stoga, u slučaju jedinične matrice, od $n!$ članova sume, samo jedan je različit od nule, pa je

$$\det(I_n) = [\, I_n \,]_{11} [\, I_n \,]_{22} \cdots [\, I_n \,]_{nn} = 1 \cdots 1 = 1.$$

Definicija determinante je matematički jasna, ali je za računanje determinante reda većeg od tri, praktično neupotrebljiva. Naime, broj članova u sumi, pa zato i broj računskih operacija, strahovito brzo raste s n.

Za determinantu vrijede sljedeća svojstva.

- $\det(A^T) = \det(A)$.
- Ako u determinanti zamijenimo dva retka (stupca), ona mijenja predznak. Stoga determinanta s dva jednaka retka ili stupca, ima vrijednost nula.
- Ako se jedan redak (stupac) determinante pomnoži skalarom α, cijela determinanta se množi s α. Stoga, ako determinanta ima jedan nul-redak ili nul-stupac, njena vrijednost je nula. Specijalno, za $A \in \mathbb{R}^{n\times n}$ vrijedi

  $$\det(\alpha A) = \alpha^n \det(A).$$

- Ako su

  $$\begin{aligned} A &= [a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n], \\ B &= [a_1, \ldots, a_{i-1}, b_i, a_{i+1}, \ldots, a_n], \\ C &= [a_1, \ldots, a_{i-1}, a_i + b_i, a_{i+1}, \ldots, a_n] \end{aligned}$$

  stupčane particije kvadratnih matrica A, B i C, respektivno, tada vrijedi

  $$\det(C) = \det(A) + \det(B).$$

  Slična tvrdnja vrijedi i za retke.
- Ako jednom retku (stupcu) determinante dodamo neki drugi redak (stupac) pomnožen proizvoljnim skalarom, determinanta matrice se ne mijenja.

- Kvadratna matrica A je regularna ako i samo ako je $\det(A) \neq 0$.
- Za kvadratne matrice A i B vrijedi

$$\det(AB) = \det(A)\det(B)$$

(to je poznati Binet–Cauchyjev teorem). Odatle odmah slijedi da je produkt dviju kvadratnih matrica singularan ako i samo ako je bar jedna od matrica u produktu singularna.

Razvoj determinante po retku i stupcu

Da bi se izračunala vrijednost determinante može se iskoristiti tzv. **razvoj determinante po retku ili stupcu**, koji se još naziva i **Laplaceov razvoj determinante**.

Neka je $A \in \mathbb{R}^{n\times n}$. S $A^c_{ij} \in \mathbb{R}^{(n-1)\times(n-1)}$ označit ćemo podmatricu od A koja nastaje izbacivanjem njenog i-tog retka i j-tog stupca. Npr. ako je

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1,j-1} & a_{1j} & a_{1,j+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{i-1,1} & \cdots & a_{i-1,j-1} & a_{i-1,j} & a_{i-1,j+1} & \cdots & a_{i-1,n} \\ a_{i,1} & \cdots & a_{i,j-1} & a_{ij} & a_{i,j+1} & \cdots & a_{i,n} \\ a_{i+1,1} & \cdots & a_{i+1,j-1} & a_{i+1,j} & a_{i+1,j+1} & \cdots & a_{i+1,n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{n,j-1} & a_{nj} & a_{n,j+1} & \cdots & a_{nn} \end{bmatrix},$$

onda je

$$A^c_{ij} = \begin{bmatrix} a_{11} & \cdots & a_{1,j-1} & a_{1,j+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{i-1,1} & \cdots & a_{i-1,j-1} & a_{i-1,j+1} & \cdots & a_{i-1,n} \\ a_{i+1,1} & \cdots & a_{i+1,j-1} & a_{i+1,j+1} & \cdots & a_{i+1,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{n,j-1} & a_{n,j+1} & \cdots & a_{nn} \end{bmatrix}.$$

Minora elementa a_{ij} matrice $A \in \mathbb{R}^{n\times n}$ je determinanta matrice A^c_{ij}, koja nastaje iz A izbacivanjem i-tog retka i j-tog stupca. **Algebarski komplement** ili **kofaktor** elementa a_{ij} je skalar $(-1)^{i+j}\det(A^c_{ij})$.

Dakle algebarski komplement se razlikuje od odgovarajuće minore tek u faktoru $(-1)^{i+j}$, tj. eventualno do na predznak. Inače, sam pojam minore je općenitiji i označava determinantu proizvoljne kvadratne podmatrice zadane matrice.

Za $A = [\, a_{ij} \,] \in \mathbb{R}^{n\times n}$ vrijedi

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A^c_{ij}), \quad 1 \leq i \leq n.$$

Ovu formulu nazivamo **razvoj determinante po i-tom retku**. Isto tako vrijedi formula

$$\det(A) = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} \det(A_{ij}^c), \quad 1 \leq j \leq n$$

koju nazivamo **razvoj determinante po j-tom stupcu**.

Razvojem determinante po stupcu ili retku, lako se pokaže da je determinanta trokutaste matrice jednaka produktu njenih dijagonalnih elemenata.

Pomoću svojstva determinante, mogu se izvesti neke korisne formule za vrijednost specijalnih determinanata.

Primjer 2.2.2 *Neka je*

$$V(x_1, x_2, x_3, x_4) = \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_1^3 & x_2^3 & x_3^3 & x_4^3 \end{vmatrix}$$

tzv. Vandermondeova determinanta reda 4. *Da bismo odredili formulu za njeno računanje, pomnožimo ju s lijeva s pogodnom matricom poznate determinante i iskoristimo Binet–Cauchyjev teorem:*

$$V(x_1, x_2, x_3, x_4) = \begin{vmatrix} 1 & & & \\ -x_1 & 1 & & \\ & -x_1 & 1 & \\ & & -x_1 & 1 \end{vmatrix} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_1^3 & x_2^3 & x_3^3 & x_4^3 \end{vmatrix}$$

$$= \begin{vmatrix} 1 & 1 & 1 & 1 \\ 0 & x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ 0 & x_2(x_2 - x_1) & x_3(x_3 - x_1) & x_4(x_4 - x_1) \\ 0 & x_2^2(x_2 - x_1) & x_3^2(x_3 - x_1) & x_4^2(x_4 - x_1) \end{vmatrix}.$$

Razvojem determinante po prvom stupcu dobivamo determinantu reda tri. U toj determinanti možemo izlučiti ispred determinante faktore $(x_2 - x_1)$, $(x_3 - x_1)$ i $(x_4 - x_1)$ koji množe 1., 2. i 3. stupac. Tako dobijemo

$$V(x_1, x_2, x_3, x_4) = (x_2 - x_1)(x_3 - x_1)(x_4 - x_1) \begin{vmatrix} 1 & 1 & 1 \\ x_2 & x_3 & x_4 \\ x_2^2 & x_3^2 & x_4^2 \end{vmatrix}$$

$$= (x_2 - x_1)(x_3 - x_1)(x_4 - x_1) V(x_2, x_3, x_4).$$

Na isti način se općenito pokazuje da vrijedi

$$V(x_1, x_2, \ldots, x_n) = (x_n - x_1)(x_{n-1} - x_1) \cdots (x_2 - x_1) V(x_2, \ldots, x_n).$$

Posebno, vrijedi

$$V(x_2, x_3, x_4) = (x_3 - x_2)(x_4 - x_2)V(x_3, x_4),$$
$$V(x_3, x_4) = (x_4 - x_3)V(x_4) = (x_4 - x_3) \cdot 1 = x_4 - x_3,$$

pa je

$$V(x_1, x_2, x_3, x_4) = (x_2 - x_1)(x_3 - x_1)(x_4 - x_1) \cdot (x_3 - x_2)(x_4 - x_2) \cdot (x_4 - x_3).$$

Općenitije, lako se pokaže da vrijedi

$$V(x_1, x_2, \ldots, x_n) = \prod_{i<j}(x_j - x_i).$$

Ovaj rezultat kaže da su vektori čije su komponente (iste) potencije međusobno različitih brojeva $x_1, \ldots, x_n$, linearno nezavisni.

Adjunkta i Cramerovo pravilo

Adjunkta kvadratne matrice A reda n, je matrica

$$\operatorname{adj}(A) = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1n} \\ \vdots & \cdots & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nn} \end{bmatrix},$$

pri čemu su

$$\alpha_{ij} = (-1)^{j+i} \det(A^c_{ji})$$

algebarski komplementi elemenata a_{ji} matrice A.

Adjunkta ima vrlo specijalno svojstvo:

$$A \operatorname{adj}(A) = \operatorname{adj}(A)A = \det(A)\, I_n.$$

Stoga, ako je A nesingularna, vrijedi

$$A^{-1} = \frac{\operatorname{adj}(A)}{\det(A)}.$$

Promotrimo sada sustav $Ax = b$ gdje je $A \in \mathbb{R}^{n\times n}$ nesingularna matrica. Definirajmo matrice

$$A_i = [a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n], \quad 1 \le i \le n,$$

pri čemu je $A = [a_1, \ldots, a_n]$ stupčana particija od A. Tada vrijedi tzv. Cramerovo pravilo: ako je $A \in \mathbb{R}^{n\times n}$ nesingularna, $b \in \mathbb{R}^n$ i ako je $x = [x_1, \ldots, x_n]^T$) rješenje sustava $Ax = b$, tada je

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad 1 \le i \le n.$$

Provjerimo formule kad je $n = 2$. Ako linearne jednadžbe glase

$$ax_1 + bx_2 = e$$
$$cx_1 + dx_2 = f$$

tada množenjem prve jednadžbe s d i druge jednadžbe s $-b$ i sumiranjem dobivenih jednadžbi (eliminacija nepoznanice x_2), dolazimo do

$$x_1 = \frac{ed - bf}{ad - bc} = \frac{\begin{vmatrix} e & b \\ f & d \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} = \frac{\det(A_1)}{\det(A)}.$$

Slično, množenjem prve jednadžbe s $-c$ i druge jednadžbe s a i sumiranjem dobivenih jednadžbi (eliminacija nepoznanice x_1), dolazimo do

$$x_2 = \frac{af - ce}{ad - bc} = \frac{\begin{vmatrix} a & e \\ c & f \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} = \frac{\det(A_2)}{\det(A)}.$$

2.2.8. Vlastite vrijednosti i vektori

Iako se vlastite (ili svojstvene) vrijednosti i vektori mogu definirati na nivou linearnih operatora, u praksi se najčešće koriste na nivou matrica. Naime, svakom linearnom operatoru u bilo kojem paru baza ("polaznog" i "dolaznog" vektorskog prostora) pripada matrica, pa se problem vlastitih vrijednosti operatora može reducirati na problem vlastitih vrijednosti matrica.

Neka je $A \in \mathbb{C}^{n\times n}$. Vektor $x \in \mathbb{C}^n$, $x \neq 0$ je **vlastiti vektor** od A ako postoji skalar λ takav da je

$$Ax = \lambda x.$$

U tom slučaju, λ je **vlastita vrijednost**, a par (x, λ) je **vlastiti par** matrice A. Ako je $A \in \mathbb{R}^{n\times n}$, vlastiti vektor od A je netrivijalni vektor iz $\mathbb{R}^n$, a vlastita vrijednost skalar iz $\mathbb{R}$.

Vidimo da se zahtijeva netrivijalnost vlastitog vektora, pa se zapravo radi o smjerovima koji se ne mijenjaju djelovanjem matrice (na vektore tih smjerova). Za neke matrice se lako odrede pripadni vlastiti parovi. Npr. ako je A nul-matrica odnosno jedinična matrica, tada je svaki netrivijalni vektor vlastiti vektor koji pripada vlastitoj vrijednosti nula odnosno jedan. S druge strane, neke realne matrice nemaju vlastite vrijednosti pa ni vlastite vektore. Takva je npr. rotacija u $\mathbb{R}^2$ koja mijenja sve netrivijalne vektore (smjerove), osim ako je kut višekratnik od π.

Neka je λ vlastita vrijednost matrice A. Po definiciji, postoji $x \neq 0$, takav da je $Ax = \lambda x$. Ima li osim x i drugih vlastitih vektora za λ? Odgovor je povrdan jer se lako provjeri da je skup

$$X_\lambda = \{x \mid Ax = \lambda x\}$$

vektorski potprostor od $\mathbb{C}^n$ (ili $\mathbb{R}^n$ ako je A realna).

Ako su u i v vlastiti vektori koji pripadaju međusobno različitim vlastitim vrijednostima λ i μ, respektivno, onda $u + v$ nije vlastiti vektor, a nije to niti bilo koja linearna kombinacija $\alpha u + \beta v$ koja zadovoljava uvjet $\alpha \neq 0$ i $\beta \neq 0$. To osigurava sljedeći rezultat.

Neka je A kvadratna matrica reda n. Neka su $u_1, \ldots, u_r$ vlastiti vektori od A i $\lambda_1, \ldots, \lambda_r$ pripadne vlastite vrijednosti. Ako vrijedi

$$\lambda_i \neq \lambda_j \quad \text{čim je } i \neq j,$$

onda je skup vektora $\{u_1, \ldots, u_r\}$ linearno nezavisan.

Ako je $r = n$, tvrdnja se može interpretirati na sljedeći način: ako matrica A reda n ima n međusobno različitih vlastitih vrijednosti, tada postoji baza prostora ($\mathbb{R}^n$ ili $\mathbb{C}^n$) koja se sastoji od vlastitih vektora matrice A.

Karakteristični polinom

Neka je λ vlastita vrijednost matrice A reda n. Tada postoji (vlastiti) vektor $x \neq 0$, takav da je $Ax = \lambda x$. Zadnja jednadžba se može zapisati u obliku $Ax - \lambda x = 0$ ili $(A - \lambda I)x = 0$. Dakle matrica $A - \lambda I$ prebacuje netrivijalan vektor u nulu, pa je singularna. Stoga je njegova determinanta nula, $\det(A - \lambda I) = 0$. Budući da je $A - \lambda I$ singularna, imamo

$$\det(\lambda I - A) = (-1)^n \det(A - \lambda I) = \begin{vmatrix} \lambda - a_{11} & \cdots & -a_{nn} \\ \vdots & \ddots & \vdots \\ -a_{n1} & \cdots & \lambda - a_{nn} \end{vmatrix} = 0. \qquad (2.2.3)$$

Determinanta matrice $zI - A$ je polinom n-tog stupnja u z,

$$\det(zI - A) = z^n - \sigma_{n-1} z^{n-1} - \cdots - \sigma_1 z - \sigma_0$$

koji se zove **karakteristični polinom** matrice A, a (2.2.3) je **karakteristična jednadžba** matrice A. S obzirom da $B = S^{-1}AS$ povlači $B - zI = S^{-1}(A - zI)S$, vidimo da slične matrice imaju isti karakteristični polinom. To znači da koeficijenti σ_i ovise o cijeloj klasi sličnih matrica matrici A.

Vrijednost λ je bila proizvoljna vlastita vrijednost, pa smo pokazali da je svaka vlastita vrijednost nultočka karakterističnog polinoma. Pokažimo da vrijedi i obrat. Neka je μ nultočka karakterističnog polinoma. Dakle, $\det(\mu I - A) = 0$. Stoga je

matrica $\mu I - A$ singularna, pa joj jezgra nije samo nul-vektor. Dakle, postoji netrivijalan vektor x, takav da je $(\mu I - A)x = 0$ ili $Ax = \mu x$, $x \neq 0$. Zaključujemo da je svaka nultočka karakterističnog polinoma vlastita vrijednost matrice A. Time smo našli jednu karakterizaciju vlastitih vrijednosti i dokazali

Teorem 2.2.2 *Neka je A kvadratna matrica. Skalar λ je vlastita vrijednost matrice A onda i samo onda ako je nultočka karakterističnog polinoma.*

Koeficijenti karakterističnog polinoma su određene sume produkata matričnih elemenata, pa neprekidno ovise o matrici. Kako su nultočke polinoma neprekidne funkcije koeficijenata polinoma, zaključujemo da su vlastite vrijednosti neprekidne funkcije matrice.

Nultočke polinoma mogu biti višestruke, npr.

$$p(z) = z^3(z-2)^2$$

ima dvije nultočke: 0 kratnosti tri i 2 kratnosti dva. Pritom je zbroj kratnosti (ili višestrukosti) svih nultočaka jednak stupnju polinoma.

Algebarska višestrukost vlastite vrijednosti λ kvadratne matrice A je njena višestrukost kao nultočke karakterističnog polinoma. **Geometrijska višestrukost** vlastite vrijednosti λ je defekt matrice $\lambda I - A$.

Koristeći fundamentalni teorem algebre, koji kaže da polinom stupnja n nad (zatvorenim poljem, a takvo je) $\mathbb{C}$ ima n nultočaka, brojeći ih s višestrukostima, zaključujemo da svaka matrica iz $\mathbb{C}^{n\times n}$ ima točno n vlastitih vrijednosti. Ako pritom sve vlastite vrijednosti imaju algebarsku kratnost jedan, tada znamo da postoji i baza vektorskog prostora $\mathbb{C}^n$ koja se sastoji od vlastitih vektora matrice A. Ako postoje vlastite vrijednosti kratnosti veće od jedan, tada može, ali i ne mora postojati baza vlastitih vektora.

Ako je matrica realna, tada ćemo u razvoju determinante $\det(zI - A)$ imati uz potencije od z uvijek realne brojeve, tj. koeficijenti karakterističnog polinoma će biti realni. Polinom stupnja n nad poljem realnih brojeva (koje nije zatvoreno) ne mora imati n nultočaka. Npr. polinom $p(t) = t^2 + 1$ uopće nema realnih korijena. Stoga matrice nad realnim poljem ne moraju imati vlastite vrijednosti, odnosno mogu ih imati manje od n.

S druge strane, svaka realna matrica A se može promatrati kao da je kompleksna, jer su realni brojevi također kompleksni brojevi. Sada je $A \in \mathbb{C}^{n\times n}$, pa karakteristični polinom ima prema osnovnom teoremu algebre n nultočaka koji su općenito kompleksni brojevi. Peciznije, ako karakteristični polinom ima realne koeficijente, njegove nultočke mogu biti ili realni brojevi ili kompleksni brojevi koji se javljaju kao parovi konjugirano kompleksnih brojeva. Stoga, kod računanja vlastitih vrijednosti realne matrice A, u pravilu se računaju sve vlastite vrijednosti od A

gledajući na A kao da je iz $\mathbb{C}^{n\times n}$. Nakon toga se odbace sve vlastite vrijednosti (i pripadni vlastiti vektori) koje nisu realni.

Postavlja se pitanje kada kvadratna matrica ima bazu koja se sastoji od vlastitih vektora, a kad nema takvu bazu.

Teorem 2.2.3 *Neka je $A \in \mathbb{C}^{n\times n}$. Matrica A ima n linearno nezavisnih vlastitih vektora ako i samo ako postoji regularna matrica G, takva da je*

$$A = G\Lambda G^{-1} \tag{2.2.4}$$

pri čemu je Λ dijagonalna matrica.

Matrice koje dopuštaju rastav (2.2.4) zovu se **dijagonalizibilne**. Ne mogu se sve matrice dijagonalizirati pomoću transformacije sličnosti. Mogu se dijagonalizirati samo one za koje postoji pun sistem vlastitih vektora. Važnu klasu dijagonalizibilnih matrica čine tzv. normalne matrice za koje vrijedi $A^*A = AA^*$. Svaka normalna matrica iz $\mathbb{C}^{n\times n}$ se može dijagonalizirati transformacijom sličnosti s unitarnom matricom G. U praksi su najvažnije simetrične realne matrice, pa ćemo im posvetiti dužnu pažnju.

Matrični polinomi

Pretpostavimo da se matrica A dade dijagonalizirati, dakle da postoji regularna matrica S, takva da je

$$S^{-1}AS = \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n).$$

Matrica Λ je dijagonalna, pa vrijedi

$$\Lambda^k = \operatorname{diag}(\lambda_1^k, \ldots, \lambda_n^k).$$

Kako je linearna kombinacija dijagonalnih matrica opet dijagonalna matrica, vrijedit će i

$$p(\Lambda) = \begin{bmatrix} p(\lambda_1) & & \\ & \ddots & \\ & & p(\lambda_n) \end{bmatrix},$$

gdje je $p(\lambda)$ bilo koji polinom.

Ako je $A = S\Lambda S^{-1}$, imamo

$$\begin{aligned} A^2 &= (S\Lambda S^{-1})(S\Lambda S^{-1}) = S\Lambda\Lambda S^{-1} = S\Lambda^2 S^{-1} \\ A^3 &= A^2 A = (S\Lambda^2 S^{-1})(S\Lambda S^{-1}) = S\Lambda^3 S^{-1} \\ &\vdots \\ A^k &= A^{k-1}A = (S\Lambda^{k-1} S^{-1})(S\Lambda S^{-1}) = S\Lambda^k S^{-1}, \end{aligned}$$

pa odmah slijedi

$$p(A) = Sp(\Lambda)S^{-1} = S \begin{bmatrix} p(\lambda_1) & & \\ & \ddots & \\ & & p(\lambda_n) \end{bmatrix} S^{-1}.$$

Na osnovu formule za polinom, funkcija matrice se definira na sličan način. Npr. ako je $f : \mathbb{R} \to \mathbb{R}$ beskonačno puta diferencijabilna funkcija i ako A ima realne vlastite vrijednosti, $f(A)$ se može ovako definirati

$$f(A) = Sf(\Lambda)S^{-1} = S \begin{bmatrix} f(\lambda_1) & & \\ & \ddots & \\ & & f(\lambda_n) \end{bmatrix} S^{-1}.$$

Neka je $A \in \mathbb{C}^{n\times n}$. Tada se mogu izračunati potencije

$$A^0 = I,\ A^1 = A,\ A^2 = A \cdot A,\ \ldots,\ A^k = A^{k-1} \cdot A,\ \ldots$$

Matrice $I, A, A^2, \ldots$ su elementi vektorskog prostora $\mathbb{C}^{n\times n}$ dimenzije n^2, pa u nizu $I, A, A^2, \ldots$ nema više od n^2 linearno nezavisnih matrica. Stoga, pođemo li od $I, A, A^2, \ldots$ možemo naći prvu potenciju A^r, koja se može prikazati kao linearna kombinacija prethodnih potencija. Dakle, možemo pisati

$$A^r = \mu_{r-1}A^{r-1} + \mu_{r-2}A^{r-2} + \cdots + \mu_1 A + \mu_0,$$

gdje su μ_i skalari. Ako definiramo polinom

$$\mu(z) = z^r - \mu_{r-1}z^{r-1} - \mu_{r-2}z^{r-2} - \cdots - \mu_1 z - \mu_0,$$

on ima svojstvo da je polinom najmanjeg stupnja koji poništava matricu A, tj. vrijedi $\mu(A) = 0$, koeficijent uz najvišu potenciju mu je jedan. Ovi uvjeti najmanjeg stupnja i normiranosti vodećeg koeficijenta, osiguravaju jedinstvenost takvog polinoma, pa se on zove **minimalni polinom** matrice A. On ima važno svojstvo da dijeli svaki polinom koji poništava matricu A. Sljedeći teorem kaže da stupanj minimalnog polinoma nije veći od n.

Teorem 2.2.4 (Hamilton–Cayley) *Neka je $A \in \mathbb{C}^{n\times n}$ i neka je $\chi(z)$ karakteristični polinom od A. Tada je $\chi(A) = 0$, tj. matrica poništava svoj karakteristični polinom.*

Kako karakteristični polinom poništava matricu, on je djeljiv (to znači djeljiv bez ostatka) s minimalnim polinomom. Dakle, minimalni polinom ima stupanj koji je manji ili jednak stupnju karakterističnog polinoma.

Iz Hamilton-Cayleyjeva teorema zaključili smo da minimalni polinom dijeli karakteristični polinom. To znači da nultočke minimalnog polinoma leže u skupu

vlastitih vrijednosti matrice. Pomnija analiza pokazuje da je svaka vlastita vrijednost matrice nultočka minimalnog polinoma. To znači da možemo pisati

$$\mu(z) = (z-\lambda_1)^{m_1}(z-\lambda_2)^{m_2}\cdots(z-\lambda_p)^{m_p},$$
$$\chi(z) = (z-\lambda_1)^{n_1}(z-\lambda_2)^{n_2}\cdots(z-\lambda_p)^{n_p}$$

pri čemu je p broj međusobno različitih vlastitih vrijednosti matrice. Pritom uvijek vrijedi $m_i \leq n_i$ za sve $1 \leq i \leq p$, a za mnoge matrice vrijedi jednakost $m_i = n_i$ za sve $1 \leq i \leq p$.

Jordanova forma matrice

Da bismo dobili uvid u doseg transformacija sličnosti kao alata za rješavnje problema vlastitih vrijednosti navest ćemo opći teorem o redukciji kvadratne matrice na oblik koji je onoliko blizak dijagonalnom obliku, koliko matrica dopušta.

Matrica oblika

$$J_k(\nu) = \begin{bmatrix} \nu & 1 & 0 & \cdots & 0 \\ 0 & \nu & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \nu & 1 \\ 0 & \cdots & \cdots & 0 & \nu \end{bmatrix} \in \mathbb{C}^{k\times k},$$

zove se **elementarna Jordanova klijetka** (ili blok) dimenzije k. Pomoću elementarnih Jordanovih klijetki građena je **Jordanova klijetka** (ili blok)

$$J(\nu) = \begin{bmatrix} J_{k_1}(\nu) & & & \\ & J_{k_2}(\nu) & & \\ & & \ddots & \\ & & & J_{k_r}(\nu) \end{bmatrix} \in \mathbb{C}^{l\times l}, \tag{2.2.5}$$

pri čemu je $k_1 \geq k_2 \geq \cdots \geq k_r \geq 1$ i $k_1 + k_2 + \cdots + k_r = l \leq n$.

Teorem 2.2.5 (Jordan) *Neka je $A \in \mathbb{C}^{n\times n}$ i neka je $\lambda_1, \ldots, \lambda_p$ zadani uređaj međusobno različitih vlastitih vrijednosti od A, s algebarskim kratnostima $n_1, \ldots, n_p$, respektivno. Tada postoji regularna matrica S, takva da je*

$$S^{-1}AS = \begin{bmatrix} J(\lambda_1) & & & \\ & J(\lambda_2) & & \\ & & \ddots & \\ & & & J(\lambda_p) \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_1 \\ \vdots \\ \}n_p \end{matrix}. \tag{2.2.6}$$

Pritom je $n_1 + \cdots + n_p = n$.

Teorem 2.2.5 kaže da se svaka matrica pomoću transformacije sličnosti može svesti na oblik koji je blizak dijagonalnom. Naime, svaki Jordanov blok $J(\lambda_k)$ je oblika (2.2.5), pa za svako $1 \leq k \leq p$, postoji rastav $n_k = m_{k,1} + \cdots + m_{k,r_k}$, pri čemu je $m_{k,j}$ dimenzija j-te po redu elementarne Jordanove klijetke u $J(\lambda_k)$. U teoremu može biti $p = 1$, $r_1 = 1$ i tada je Jordanov oblik od A jedna elementarna Jordanova klijetka reda n. U drugoj krajnosti kad je $p = n$, mora za svako k biti $n_k = 1$, pa je $r_k = 1$ i $m_{k,1} = 1$, što znači da je Jordanova klijetka $J(\lambda_k)$ jedna elementarna Jordanova klijetka. Dakle, sve jordanove klijetke u Jordanovoj formi matrice su reda 1, pa je matrica dijagonalizibilna. U ostalim slučajevima će općenito vrijediti

$$\begin{aligned} n &= n_1 + \cdots + n_p \\ n_1 &= m_{1,1} + \cdots + m_{1,r_1} \\ &\vdots \\ n_p &= m_{p,1} + \cdots + m_{p,r_p}. \end{aligned}$$

Za broj r_k elementarnih Jordanovih klijetki unutar Jordanove klijetke $J(\lambda_k)$ vrijedi ograničenje $1 \leq r_k \leq n$, a za dimenzije elementarnih Jordanovih klijetki $m_{k,j}$ vrijedi $0 \leq m_{k,j} \leq n_k \leq n$.

Zbog

$$\det(zI - S^{-1}AS) = \det(S^{-1}(zI - A)S) = \det(zI - A)$$

zaključujemo da je n_i algebarska kratnost od λ_i. Kako elementarne Jordanove klijetke imaju samo jedan vlastiti vektor e_1, zaključujemo da je geometrijska kratnost od λ_i broj r_i. Iz teorema 2.2.5 također slijedi da je minimalni polinom od A,

$$\mu(z) = (z - \lambda_1)^{m_{1,1}}(z - \lambda_1)^{m_{2,1}} \cdots (z - \lambda_1)^{m_{p,1}},$$

gdje je za svako k, $m_{k,1}$ najveći red elementarne klijetke unutar Jordanove klijetke $J(\lambda_k)$.

Dijagonalizacija simetrične matrice

Vidjeli smo da se transformacijom sličnosti matrica može svesti na Jordanovu formu, oblik koji je dosta blizak dijagonalnoj fromi. Međutim ako zahtijevamo da sličnost bude načinjena pomoću unitarne matrice, tada je doseg dijagonalizacije slabiji. Pomoću unitarne transformacije sličnosti matrica se može svesti na trokutasti oblik.

Teorem 2.2.6 (Schur) *Za proizvoljnu matricu A postoji unitarna matrica U, takva da je $T = U^*AU$ gornjetrokutasta matrica. Matrica U može biti izabrana tako da se na dijagonali matrice T pojave vlastite vrijednosti od A u bilo kojem zadanom poretku.*

Schurov teorem ima neke vrlo važne posljedice, kao što je naredni teorem koji kaže da je Schurova dekompozicija normalnih matrica dijagonalna. Sjetimo se, matrica A je normalna ako je $A^*A = AA^*$. Pokažimo da je unitarno slična matrica normalnoj matrici opet normalna. Neka je A normalna i $B = U^*AU$ za neku unitarnu matricu U. Tada je

$$B^*B = U^*A^*UU^*AU = U^*A^*AU = U^*AA^*U = U^*AUU^*A^*U = BB^*,$$

pa je B normalna. Sljedeći teorem pokazuje da se normalne matrice mogu dijagonalizirati pomoću unitarne transformacije sličnosti.

Teorem 2.2.7 *Ako je A normalna matrica i $T = U^*AU$ Schurova dekompozicija od A, tada je T dijagonalna matrica.*

Jedna od posljedica ovog teorema jest tvrdnja da normalna matrica reda n ima sustav od n ortonormiranih (ortogonalnih i normiranih tj. euklidske norme jedan) vlastitih vektora koji razapinju $\mathbb{C}^n$.

Spomenuli smo dvije najznačajnije klase normalnih matrica: unitarne i hermitske matrice. Sljedeće tvrdnje proizlaze iz teorema 2.2.7.

Unitarna matrica je normalna matrica kojoj vlastite vrijednosti imaju modul jedan. Hermitska matrica je normalna matrica koja ima realne vlastite vrijednosti.

Dakle, hermitska matrica H može se prikazati u obliku

$$H = U\Lambda U^*, \tag{2.2.7}$$

gdje je $U = [u_1, \ldots, u_n]$ unitarna matrica kojoj su stupci vlastiti vektori od H, a Λ je dijagonalna matrica s realnim vlastitim vrijednostima od H na dijagonali. Taj oblik zovemo **spektralna dekompozicija** od H. Ponekad ćemo koristiti i zapis

$$H = \sum_i \lambda_i u_i u_i^*, \tag{2.2.8}$$

koji je ekvivalentan s (2.2.7). To odmah slijedi iz zadatka 2.2.1(v) u kojem uzmite $A = U\Lambda$, $B = U^*$. Pomoću formule (2.2.8) možemo proširiti pojam skalarne funkcije na hermitske matrice. To se čini na slijedeći način

$$\varphi(H) = \sum_i \varphi(\lambda_i) u_i u_i^*. \tag{2.2.9}$$

Uočimo da formula (2.2.9) vrijedi za polinome, a vrijedi i za neprekidne funkcije realnog argumenta. Jedna od najčešće korištenih funkcija je drugi korijen. Ako je H pozitivno definitna matrica, onda se drugi korijen matrice H dobiva formulom

$$H^{\frac{1}{2}} = \sum_i \lambda_i^{\frac{1}{2}} u_i u_i^*.$$

Ako je Hermitska matrica realna, zove se simetrična. Ona se može dijagonalizirati pomoću transformacije sličnosti s ortogonalnom matricom, pa se njena spektralna dekompozicija zapisuje u obliku

$$A = Q\Lambda Q^T,$$

gdje je Λ dijagonalna matrica vlastitih vrijednosti od A, a Q je ortogonalna matrica čiji su stupci vlastiti vektori od A.

2.3. Singularna dekompozicija matrice

Singularna dekompozicija matrice jedna je od najkorištenijih dekompozicija u numeričkoj linearnoj algebri. Stoga ćemo dokazati glavne rezultate vezane uz tu dekompoziciju.

2.3.1. Definicija i osnovni teoremi

Ako je H hermitska pozitivno definitna (semidefinitna) matrica onda su njene vlastite vrijednosti pozitivne (nenegativne). Ako je $C \in \mathbb{C}^{m\times n}$ onda su obje matrice C^*C i CC^* hermitske i pozitivno semidefinitne. Ako je $m = n$, vlastite vrijednosti matrica C^*C i CC^* se podudaraju. Ako je $m \neq n$, matrica C^*C (CC^*) ima n (m) vlastitih vrijednosti. Pritom su netrivijalne vlastite vrijednosti ovih matrica jednake. Uskoro ćemo pokazati vezu vlastitih vrijednosti ovih matrica sa singularnim vrijednostima od C.

Sljedeći teorem o egzistenciji singularne dekompozicije služi i kao definicija singularnih vrijednosti i vektora matrice. Pomoću njega će se jednostavno dokazati spomenuta tvrdnja o vlastitim vrijednostima matrica C^*C i CC^*.

Teorem 2.3.1 (Singularna dekompozicija matrice) *Ako je $C \in \mathbb{C}^{m\times n}$, tada postoje unitarne matrice $U \in \mathbb{C}^{m\times m}$ i $V \in \mathbb{C}^{n\times n}$, takve da je*

$$U^*CV = \Sigma, \quad \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min\{m,n\}}),$$

pri čemu vrijedi

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}} \geq 0.$$

Brojeve $\sigma_1, \sigma_2, \ldots, \sigma_{\min\{m,n\}}$ zovemo **singularne vrijednosti** *matrice C. Stupce matrice U zovemo lijevi, a stupce matrice V desni singularni vektori matrice C.*

Dokaz. Pošto je jedinična sfera u $\mathbb{C}^n$ ograničen i zatvoren skup, on je kompaktan, pa svaka neprekidna funkcija na njemu dostiže minimum i maksimum. Funkcija $f(x) = \|Cx\|_2$ je neprekidna, pa postoji jedinični vektor $v \in \mathbb{C}^n$, takav da je

$$\|Cv\|_2 = \max\{\|Cx\|_2 \mid \ \|x\|_2 = 1,\ x \in \mathbb{C}^n\}.$$

Ako je $\|Cv\|_2 = 0$, onda je $C = 0$ i faktorizacija u iskazu teorema je trivijalna uz $\Sigma = 0$ i s proizvoljnim unitarnim matricama U i V reda m i n, respektivno.

Ako je $\|Cv\|_2 > 0$, stavimo $\sigma_1 = \|Cv\|_2$ i formirajmo jedinični vektor

$$u_1 = \frac{Cv}{\sigma_1} \in \mathbb{C}^m.$$

Nadopunimo u_1 s $m-1$ vektora do baze u $\mathbb{C}^m$ i onda primijenimo npr. Gramm–Schmidtov proces ortogonalizacije, tako da dobijemo ortonormiranu bazu $u_1, \ldots, u_m$ za $\mathbb{C}^m$. Drugim riječima, dobili smo unitarnu matricu $U_1 = [u_1, u_2, \ldots, u_m]$. Slično, za $v_1 = v$ postoji $n-1$ ortonormiranih vektora $v_2, v_3, \ldots, v_n \in \mathbb{C}^n$, takvih je da matrica $V_1 = [v_1, v_2, \ldots, v_n]$ unitarna. Tada je

$$C_1 = U_1^* C V_1 = \begin{bmatrix} u_1^* \\ u_2^* \\ \vdots \\ u_m \end{bmatrix} [\, Cv_1 \quad Cv_2 \quad \cdots \quad Cv_n \,] = \begin{bmatrix} u_1^* \\ u_2^* \\ \vdots \\ u_m \end{bmatrix} [\, \sigma_1 u_1 \quad Cv_2 \quad \cdots \quad Cv_n \,]$$

$$= \begin{bmatrix} \sigma_1 & u_1^* C v_2 & \cdots & u_1^* C v_n \\ 0 & u_2^* C v_2 & \cdots & u_2^* C v_n \\ \vdots & \vdots & & \cdots \\ 0 & u_m^* C v_2 & \cdots & u_m^* C v_n \end{bmatrix} = \begin{bmatrix} \sigma_1 & z^* \\ 0 & C_2 \end{bmatrix},$$

gdje je $z \in \mathbb{C}^{n-1}$, $C_2 \in \mathbb{C}^{(m-1)\times(n-1)}$. Za jedinični vektor

$$y = \frac{1}{\sqrt{\sigma_1^2 + z^* z}} \begin{bmatrix} \sigma_1 \\ z \end{bmatrix}, \tag{2.3.1}$$

zbog unitarne invarjantnosti euklidske norme, vrijedi

$$\|C(V_1 y)\|_2^2 = \|(U_1^* C V_1) y\|_2^2 = \|C_1 y\|_2^2 = \frac{(\sigma_1^2 + z^* z)^2 + \|C_2 z\|_2^2}{\sigma_1 + z^* z} \geq \sigma_1^2 + z^* z,$$

a ovo je striktno veće od σ_1^2 ako je $z \neq 0$. Pošto je to u suprotnosti sa maksimalnošću od σ_1, zaključujemo da je $z = 0$. Stoga je

$$C_1 = U_1^* C V_1 = \begin{bmatrix} \sigma_1 & 0 \\ 0 & C_2 \end{bmatrix}. \tag{2.3.2}$$

Sada ponavljamo isti argument za matricu $C_2 \in \mathbb{C}^{(m-1)\times(n-1)}$. Na taj taj način dobivamo unitarne matrice U i V kao produkt unitarnih matrica koje su dobijene nakon svakog koraka. Ako je $m \geq n$ taj postupak vodi do dijagonalne matrice Σ.

Ako je $m \leq n$, u zadnjem koraku radimo s matricom $C_m \in \mathbb{C}^{1\times(n-m+1)}$. Može se pokazati da za C_m postoji unitarna matrica, takva da je

$$C_m V_m = [\, \|C_m\|_2, \quad 0, \quad \cdots, \quad 0 \,],$$

(to tzv. LQ faktorizazija od C_m), pa će lijeve i desne komponentne unitarne matrice u zadnjem koraku biti $U_m = I_1$ i V_m, respektivno.

Ako čitatelju nije jasan pojam LQ faktorizacije, zadnji dio dokaza teorema možemo premostiti na sljedeći način. Ako je $m < n$, primijenimo postupak opisan u dokazu teorema na matricu C^*, za koju je broj redaka n veći od broja stupaca m. Nakon dobivene dekompozicije $C^* = U\Sigma V^*$, kompleksno transponirajmo obje matrice u toj jednakosti. Dobijemo

$$C = \widetilde{V}\widetilde{\Sigma}^T\widetilde{U}^*,$$

što je tražena singularna dekompozicija od C, pri čemu moramo još preimenovati $\widetilde{V}$ u U, $\widetilde{U}$ u V i $\widetilde{\Sigma}^T$ u Σ. ■

Primjer 2.3.1 *Evo primjera singularne dekompozicije matrice reda dva:*

$$C = \begin{bmatrix} 0.96 & 1.72 \\ 2.28 & 0.96 \end{bmatrix} = U\Sigma V^T = \begin{bmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{bmatrix}^T.$$

Singularne vrijednosti su unitarno invarijantne, jer ako je $C = U\Sigma V^*$ singularna dekompozicija od C, tada je za proizvoljne unitarne matrice $W_1 \in \mathbb{C}^{m\times m}$ i $W_2 \in \mathbb{C}^{n\times n}$, $(W_1U)\Sigma(W_2^*V)^*$ singularna dekompozicija od W_1CW_2.

Iz teorema 2.3.1 odmah slijede važne relacije

$$Cv_i = \sigma_i u_i, \quad C^*u_i = \sigma_i v_i,$$

koje otkrivaju vezu ortonormiranih baza $u_i, \ldots, u_m$ i $v_1, \ldots, v_n$ s matricom C. Iz relacije

$$Cx = U\Sigma V^*x = U(\Sigma(V^*x)),$$

vidimo da se djelovanje matrice na vektor može opisati kao niz od tri jednostavnije transformacije: rotacije, produljivanje/skraćivanje komponenata vektora i opet jedne rotacije. Naime, unitarna matrica ne mijenja normu vektora, pa je njen geometrijski smisao rotacija.

Polazeći od definicije singularnih vrijednosti iz teorema 2.3.1, pokažimo da su σ_i, $1 \le i \le \min\{m,n\}$ kvadratni korijeni vlastitih vrijednosti matrica C^*C i CC^*. Neka je $U^*CV = \Sigma$, $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min\{m,n\}})$ singularna dekompozicija matrice C. Tada je $\Sigma^* = V^*C^*U$, pa je

$$\Sigma^*\Sigma = V^*C^*\underbrace{UU^*}_{I}CV = V^*C^*CV \in \mathbb{C}^{n\times n}$$
$$\Sigma\Sigma^* = U^*C\underbrace{VV^*}_{I}C^*U = U^*CC^*U \in \mathbb{C}^{m\times m}.$$

Dakle, $\sigma_1^2, \sigma_2^2, \ldots, \sigma_{\min\{m,n\}}^2$ su (neke) vlastite vrijednosti od C^*C i CC^*. Ako je $m > n$ ($n > m$) onda CC^* (C^*C) ima još dodatnih $m - n$ ($n - m$) vlastitih

vrijednosti koje su jednake nuli. U tom smislu kažemo da su singularne vrijednosti matrice C korijeni vlastitih vrijednosti od C^*C i CC^*.

Veza između singularnih vrijednosti od C i vlastitih vrijednosti od C^*C pokazuje da su singularne vrijednosti jedinstvene, tj. da je matrica Σ određena na jedinstven način matricom C. Sljedeći teorem govori o tome koliko su U i V jedinstvene. U njemu oznaka ortogonalne sume matrica $U_1 \oplus U_2 \oplus \cdots \oplus U_k \oplus U_0$ označava blok-dijagonalnu matricu $\operatorname{diag}(U_1, \ldots, U_k, U_0)$.

Teorem 2.3.2 *Neka je C definirana kao u teoremu 2.3.1. Pretpostavimo da za netrivijalne singularne vrijednosti matrice C i za pripadne višekratnosti, vrijedi*

$$\sigma_1 > \sigma_2 > \cdots > \sigma_k > 0 \quad i \quad \mu_1 + \mu_2 + \cdots + \mu_k = r,$$

respektivno. Neka je

$$C = U\Sigma V^*, \quad \Sigma = \operatorname{diag}(\sigma_1 I_{\mu_1}, \sigma_2 I_{\mu_2}, \ldots, \sigma_k I_{\mu_k}, 0_{\min\{m,n\}-r})$$

singularna dekompozicija od C i neka su $\widetilde{U} \in \mathbb{C}^{m\times m}$ i $\widetilde{V} \in \mathbb{C}^{n\times n}$ unitarne matrice. Tada je

$$C = \widetilde{U}\Sigma\widetilde{V}^*$$

onda i samo onda ako postoje unitarne matrice $U_0 \in \mathbb{C}^{(m-r)\times(m-r)}$, $V_0 \in \mathbb{C}^{(n-r)\times(n-r)}$ i $U_i \in \mathbb{C}^{\mu_i\times\mu_i}$, $1 \le i \le k$, takve da je

$$\widetilde{U} = U(U_1 \oplus U_2 \oplus \cdots \oplus U_k \oplus U_0)$$
$$\widetilde{V} = V(V_1 \oplus V_2 \oplus \cdots \oplus V_k \oplus V_0).$$

Dokaz. Iz singularne dekompozicije $C = U\Sigma V^*$ slijedi

$$CC^* = U\Sigma\Sigma^*U^* = U[\sigma_1^2 I_{\mu_1} \oplus \sigma_2^2 I_{\mu_2} \oplus \cdots \oplus \sigma_k^2 I_{\mu_k} \oplus 0_{m-r}]U^* \equiv US_1U^*,$$
$$C^*C = V\Sigma^*\Sigma V^* = V[\sigma_1^2 I_{\mu_1} \oplus \sigma_2^2 I_{\mu_2} \oplus \cdots \oplus \sigma_k^2 I_{\mu_k} \oplus 0_{n-r}]V^* \equiv VS_2V^*,$$

gdje su $I_{\mu_j} \in \mathbb{C}^{\mu_j\times\mu_j}$, $1 \le j \le k$ jedinične matrice, a $0_{m-r} \in \mathbb{C}^{(m-r)\times(m-r)}$ i $0_{n-r} \in \mathbb{C}^{(n-r)\times(n-r)}$ nul-matrice. Pritom su S_1 i S_2 dijagonalne. Ako su $\widetilde{U} \in \mathbb{C}^{m\times m}$ i $\widetilde{V} \in \mathbb{C}^{n\times n}$ unitarne matrice za koje je $C = \widetilde{U}\Sigma\widetilde{V}^*$, tada vrijedi

$$CC^* = \widetilde{U}S_1\widetilde{U}^*,$$

pa je

$$US_1U^* = \widetilde{U}S_1\widetilde{U}^*.$$

Odavde slijedi

$$S_1(U^*\widetilde{U}) = (U^*\widetilde{U})S_1.$$

Na sličan način, relacija $C = \widetilde{U}\Sigma\widetilde{V}^*$ povlači

$$C^*C = \widetilde{V}S_2\widetilde{V}^*,$$

odnosno

$$VS_2V^* = \widetilde{V}S_2\widetilde{V}^*,$$

pa je

$$S_2(V^*\widetilde{V}) = (V^*\widetilde{V})S_2.$$

Dakle $U^*\widetilde{U}$ i $V^*\widetilde{V}$ komutiraju s dijagonalnim matricama S_1 i S_2, respektivno. To povlači da su one unitarne blok dijagonalne s dimenzijama dijagonalnih blokova $\mu_1, \mu_2, \ldots, \mu_k, m-r$ i $\mu_1, \mu_2, \ldots, \mu_k, n-r$, respektivno. Stoga vrijedi

$$\widetilde{U} = U[U_1 \oplus U_2 \oplus \cdots \oplus U_k \oplus U_0],$$
$$\widetilde{V} = V[V_1 \oplus V_2 \oplus \cdots \oplus V_k \oplus V_0],$$

gdje su $U_0 \in \mathbb{C}^{(m-r)\times(m-r)}$, $V_0 \in \mathbb{C}^{(n-r)\times(n-r)}$ i $U_i, V_i \in \mathbb{C}^{\mu_i\times\mu_i}$, $1 \leq i \leq k$, unitarne. Sada iz relacije

$$\widetilde{U}\Sigma\widetilde{V}^* = U\Sigma V^*$$

slijedi

$$U_i\sigma_iV_i^* = \sigma_iI_{\mu_i}, \quad \text{odnosno} \quad \sigma_iU_i = \sigma_iV_i \quad 1 \leq i \leq k.$$

Kako su $\sigma_i > 0$, slijedi $U_i = V_i$, $1 \leq i \leq k$. ■

Ako je polazna matrica realna, singularna dekompozicija se zapisuje kao $C = U\Sigma V^T$, pri čemu je Σ kao i dosad, a $U \in \mathbb{R}^{m\times m}$ i $V \in \mathbb{R}^{n\times n}$ su ortogonalne. Pažljivo čitajući dokaze prethodnih teorema, ta tvrdnja se lako dokaže. Na svim mjestima gdje se koristi znak kompleksnog transponiranja treba staviti znak transponiranja, a riječ unitarna zamijeniti s ortogonalna. Nadalje, treba koristiti prostore $\mathbb{R}^m$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$ umjesto $\mathbb{C}^m$, $\mathbb{C}^n$, $\mathbb{C}^{m\times n}$, respektivno.

2.3.2. Izravne posljedice singularne dekompozicije

Zanimljiv je odnos između vlastitih i singularnih vrijednosti normalne matrice. Neka je $N \in \mathbb{C}^{n\times n}$ normalna matrica i neka je $N = V\Lambda V^*$ njena spektralna dekompozicija. Pritom je V unitarna matrica, a Λ dijagonalna,

$$\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|.$$

Definirajmo

$$\Sigma = \operatorname{diag}(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_n|)$$

i

$$\Phi = \operatorname{diag}(e^{i\phi_1}, e^{i\phi_2}, \ldots, e^{i\phi_n}),$$

tako da je $\Lambda = \Phi\Sigma$. Tada je

$$N = V\Lambda V^* = V\Phi\Sigma V^* = \underbrace{V\Phi}_{U}\Sigma V^* = U\Sigma V^*,$$

singularna dekompozicija matrice N. Zaključujemo da su singularne vrijednosti normalne matrice apsolutne vrijednosti njenih vlastitih vrijednosti.

Sljedeći korolar pokazuje neke od pogodnosti koje pruža singularna dekompozicija matrice. Pritom ćemo koristiti oznake: $\sigma_i(C)$ je i-ta najveća singularna vrijednost matrice C, $\sigma_{\max}(C) = \sigma_1(C)$ je najveća, a $\sigma_{\min}(C) = \sigma_{\min\{m,n\}}(C)$ najmanja singularna vrijednost matrice C. Ako znamo da se radi o matrici C, umjesto $\sigma_i(C)$ koristit ćemo kraću oznaku σ_i. To znači da pretpostavljamo nerastući uređaj singularnih vrijednosti, kako je to u teoremu 2.3.1 i definirano.

Korolar 2.3.1 *Neka je $C = U\Sigma V^*$ singularna dekompozicija matrice $C \in \mathbb{C}^{m\times n}$ za koju vrijedi*

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_{\min} = 0.$$

Neka su

$$\begin{aligned}\Sigma_r &= \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{C}^{r\times r},\\ U_r &= [u_1, u_2, \ldots, u_r] \in C^{m\times r},\\ V_r &= [v_1, v_2, \ldots, v_r] \in \mathbb{C}^{n\times r}\end{aligned}$$

gdje su u_i i v_i stupci od U i V, respektivno. Tada je $\operatorname{rang}(C) = r$ *i*

(i) $\mathcal{N}(C) = \operatorname{span}\{v_{r+1}, \ldots, v_n\}$,

(ii) $\mathcal{N}(C^*) = \operatorname{span}\{u_{r+1}, \ldots, u_m\}$

(iii) $\mathcal{R}(C) = \operatorname{span}\{u_1, u_2, \ldots, u_r\}$

(iv) $\mathcal{R}(C^*) = \operatorname{span}\{v_1, v_2, \ldots, v_r\}$

(v) $\mathcal{R}(C^*) = \mathcal{N}(C)^\perp$

(vi) $\mathcal{N}(C^*) = \mathcal{R}(C)^\perp$

(vii) $\mathcal{R}(C^*C) = \mathcal{R}(C^*)$, $\mathcal{R}(CC^*) = \mathcal{R}(C)$

(viii) $C = \sum_{i=1}^r \sigma_i u_i v_i^* = U_r \Sigma_r V_r^*$

(ix) $\|C\|_F^2 = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_{\min}^2$

(x) $\|C\|_2 = \sigma_1$.

Dokaz. Rang matrice se ne mijenja ako pomnožimo matricu slijeva ili zdesna nesingularnom matricom. Stoga je $\operatorname{rang}(C) = \operatorname{rang}(\Sigma)$. Kako je rang broj linearno nezavisnih stupaca (ili redaka) matrice, očito je $\operatorname{rang}(\Sigma) = r$.

(i) Pošto je $Cv_i = 0$, $r+1 \leq i \leq n$, zaključujemo da je

$$\operatorname{span}\{v_{r+1}, \ldots, v_n\} \subseteq \mathcal{N}(C).$$

Iz $\operatorname{defekt}(C) = n - \operatorname{rang}(C) = n - r$ izlazi da je dimenzija potprostora $\operatorname{span}\{v_{r+1}, \ldots, v_n\}$ ista kao i dimenzija od $\mathcal{N}(C)$.

(ii) Singularna dekompozicija od C^* je $C^* = V\Sigma U^*$, a C i C^* imaju isti rang, pa je tvrdnja (ii) je tvrdnja (i) za C^*.

(iii) Znamo da je $\mathcal{R}(C) = \text{span}(Cv_1, \ldots, Cv_n)$ jer je $v_1, \ldots, v_n$ baza. Stoga je $\text{span}(Cv_1, \ldots, Cv_r) \subseteq \mathcal{R}(C)$. Međutim, za $1 \leq i \leq r$ vrijedi

$$Cv_i = U\Sigma V^* v_i = U\Sigma e_i = \sigma_i U e_i = \sigma_i u_i \quad \text{i} \quad \sigma_i > 0.$$

Vektori u_i su linearno nezavisni, pa

$$\text{span}(\sigma_1 u_1, \ldots, \sigma_r u_r) = \text{span}(u_1, \ldots, u_r)$$

ima dimenziju r kao i $\mathcal{R}(C)$, te je $\mathcal{R}(C) = \text{span}(u_1, \ldots, u_r)$.

(iv) Dokaz ove tvrdnje koristi isti argument kao i dokaz tvrdnje (ii).

(v) Korištenjem tvrdnji (iv) i (i), imamo

$$\mathcal{R}(C^*) = \text{span}\{v_1, \ldots, v_r\} = \mathcal{N}(C)^\perp.$$

(vi) Korištenjem tvrdnji (ii) i (iii), izlazi

$$\mathcal{N}(C^*) = \text{span}\{u_1, \ldots, u_r\} = \mathcal{R}(C)^\perp.$$

(vii) Zbog toga što su

$$\mathcal{R}(C^*C) = \mathcal{R}(V\Sigma^*\Sigma V^*) \quad \text{i} \quad \mathcal{R}(C^*C) = \mathcal{R}(U\Sigma\Sigma^* U^*)$$

singularne dekompozicije matrica C^*C i CC^*, respektivno, tvrdnja (vii) slijedi iz tvrdnji (iv) i (iii).

(viii) Tvrdnja slijedi izravno pomnožimo li odgovarajuće matrice:

$$\begin{aligned} C &= \begin{bmatrix} U_r, & U_{m-r} \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r, & V_{n-r} \end{bmatrix}^* = \begin{bmatrix} U_r\Sigma_r, & 0 \end{bmatrix} \begin{bmatrix} V_r^* \\ V_{n-r}^* \end{bmatrix} \\ &= U_r\Sigma_r V_r^* = \sum_{i=1}^{r} \sigma_i u_i v_i^*. \end{aligned}$$

(ix) Tvrdnja slijede iz činjenice da je euklidska norma unitarno invarijantna

$$\|C\|_F^2 = \|U\Sigma V^*\|_F^2 = \|\Sigma\|_F^2 = \sum_{i=1}^{r} \sigma_i^2.$$

(x) Slično kao u (ix), 2-norma je unitarno invarijantna, pa je

$$\|C\|_2 = \|U\Sigma V^*\|_2 = \|\Sigma\|_2 = \sigma_{\max}(C).$$

■

Dekompoziciju opisanu u tvrdnji (viii) korolara 2.3.1 zvat ćemo **skraćena singularna dekompozicija** matrice C.

Jedna od važnijih posljedica singularne dekompozicije matrice je i polarna dekompozicija. Ona se dobije izravno od singularne dekompozicije matrice.

Teorem 2.3.3 *Svaka kvadratna matrica C dopušta prikaz*

$$C = QH_1 = H_2Q,$$

gdje je Q unitarna, a H_1 i H_2 su hermitske pozitivno semidefinitne matrice kojima su vlastite vrijednosti upravo singularne vrijednosti od C.

Dokaz. Iz singularne dekompozicije matrice C odmah slijedi

$$\begin{aligned} C &= U\Sigma V^* = UV^*V\Sigma V^* = (UV^*)(V\Sigma V^*) = QH_1 \\ &= U\Sigma V^* = U\Sigma U^*UV^* = (U\Sigma U^*)(UV^*) = H_2Q. \end{aligned}$$

■

2.3.3. Aproksimacija matrice matricom manjeg ranga

Jedna od važnijih upotreba singularne dekompozicije matrice je određivanje ranga matrice. Brojni teoremi u linearnoj algebri imaju oblik : "ako je ta i ta matrica punog ranga, onda vrijedi to i to svojstvo". Zbog pogrešaka zaokruživanja i eventualnih pogrešaka ulaznih podataka, određivanje ranga postaje netrivijalan zadatak. Za singularnu dekompoziciju matrice postoje pouzdane i prilično točne metode. Stoga se, iako postoje brže metode, rang matrice najsigurnije računa iz singularne dekompozicije.

Sljedeći teorem nam govori o aproksimaciji matrice pomoću matrica manjeg ranga.

Teorem 2.3.4 (Ekhard, Young, Mirsky) *Neka je $C = U\Sigma V^*$ singularna dekompozicija matrice $C \in \mathbb{C}^{m\times n}$ ranga r. Neka je $k < r$ i*

$$C_k = \sum_{i=1}^{k} \sigma_i u_i v_i^*.$$

Tada je

$$\min_{rang(K)=k} \|C - K\|_2 = \|C - C_k\|_2 = \sigma_{k+1}.$$

Dokaz. Pošto je

$$U^*C_kV = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0),$$

lako se dobije

$$\|C - C_k\|_2 = \|U^*(C - C_k)V\|_2 = \|\operatorname{diag}(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_{\min\{m,n\}})\|_2 = \sigma_{k+1}.$$

Neka je $K \in \mathbb{C}^{m\times n}$ proizvoljna matrica ranga k. To znači da možemo naći ortonormirane vektore $x_1, \ldots, x_{n-k}$ takve da je

$$\mathcal{N}(K) = \operatorname{span}\{x_1, \ldots, x_{n-k}\}.$$

Zbog

$$\operatorname{span}\{x_1, \ldots, x_{n-k}\} \bigcap \operatorname{span}\{v_1, \ldots, v_{k+1}\} \neq \{0\},$$

možemo pronaći jedinični vektor z koji pripada ovom presjeku. Pošto je $Kz = 0$ i

$$Cz = \sum_{i=1}^{k+1} \sigma_i (v_i^* z) u_i,$$

imamo

$$\|C - K\|_2^2 \geq \|(C - K)z\|_2^2 = \|Cz\|_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 |v_i^* z|^2 \geq \sigma_{k+1}^2.$$

Zadnja nejednakost vrijedi jer se radi o konveksnoj sumi brojeva σ_i, $1 \leq i \leq k+1$, a konveksnost sume slijedi iz činjenice da je $0 \leq |v_i^* z|^2 \leq 1$ i

$$\sum_{i=1}^{k+1} |v_i^* z|^2 = \|z\|_2^2 = 1.$$

■

Zadnji teorem nam specijalno kaže da je najmanja singularna vrijednost matrice jednaka udaljenosti (u spektralnoj normi) između matrice i skupa singularnih matrica. Sljedeći primjeri ilustriraju tvrdnju teorema.

Primjer 2.3.2 *Neka se*

$$C = \begin{bmatrix} 6.4 & 1.8 & 2.4 \\ -4.8 & 2.4 & 3.2 \\ 0.0 & -2.4 & 1.8 \end{bmatrix}, \quad K = \begin{bmatrix} 6.4 & 0.0 & 0.0 \\ -4.8 & 0.0 & 0.0 \\ 0.0 & -2.4 & 1.8 \end{bmatrix}.$$

Iz singularne dekompozicije od C

$$\begin{bmatrix} 6.4 & 1.8 & 2.4 \\ -4.8 & 2.4 & 3.2 \\ 0.0 & -2.4 & 1.8 \end{bmatrix} = \begin{bmatrix} -0.8 & 0.6 & 0.0 \\ 0.6 & 0.8 & 0.0 \\ 0.0 & 0.0 & -1.0 \end{bmatrix} \begin{bmatrix} 8 & & \\ & 5 & \\ & & 3 \end{bmatrix} \begin{bmatrix} -1.0 & 0.0 & 0.0 \\ 0.0 & 0.6 & 0.8 \\ 0.0 & 0.8 & -0.8 \end{bmatrix},$$

zaključujemo da je $\sigma_1(C) = 8$, $\sigma_2(C) = 5$, $\sigma_3(C) = 3$, *pa je* $\operatorname{rang}(C) = 3$. *Uočimo da je* $\operatorname{rang}(K) = 2$. *Spektralnu normu od* $C - K$ *dobijemo opet preko njene singularne dekompozicije*

$$\begin{bmatrix} 0.0 & 1.8 & 2.4 \\ 0.0 & 2.4 & 3.2 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} = \begin{bmatrix} 0.6 & -0.8 & 0.0 \\ 0.8 & 0.6 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 5 & & \\ & 0 & \\ & & 0 \end{bmatrix} \begin{bmatrix} 0.0 & 0.6 & 0.8 \\ 0.0 & -0.8 & 0.6 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}.$$

Dakle imamo $\|C - K\|_2 = 5 > 3 = \sigma_3(C)$.

Primjer 2.3.3 *Neka su*

$$C = \begin{bmatrix} 2.40 & 0.00 & 2.40 & 0.00 \\ 1.92 & 0.96 & 1.08 & 1.28 \\ 2.56 & 0.72 & 1.44 & 0.96 \\ 0.00 & 0.80 & 0.00 & 0.60 \end{bmatrix}, \quad K = \begin{bmatrix} 2.40 & 0.00 & 2.40 & 0.00 \\ 1.92 & 0.96 & 1.08 & 1.28 \\ 2.56 & 0.72 & 1.44 & 0.96 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}.$$

Slično kao u prethodnom primjeru, singularnom dekompozicijom matrice C dobivamo $\sigma_1(C) = 4$, $\sigma_2(C) = 3$, $\sigma_3(C) = 2$, $\sigma_4(C) = 1$. Dakle, vrijedi $\operatorname{rang}(C) = 4$. *Očito je* $\operatorname{rang}(K) = 3$, *a spektralna norma od $C - K$ je*

$$\|C - K\|_2 = \left\| \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.0 & 0.6 \end{bmatrix} \right\|_2 = 1.$$

Dakle, imamo $\|C - K\|_2 = \sigma_4(C)$, pa se udaljenost matrice C do skupa singularnih matrica dostiže na matrici K.

2.3.4. Wielandtova matrica

Sljedeći teorem je jedan od značajnijih rezultata vezanih uz singularnu dekompoziciju matrice. On nam daje svezu imeđu singularnih vrijednosti (i vektora) matrice i vlastitih vrijednosti (i vektora) jedne hermitske matrice koja je usko povezana uz danu matricu.

Teorem 2.3.5 (Jordan–Wielandt) *Neka je*

$$C = U\Sigma V^*, \quad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n),$$

singularna dekompozicija matrice $C \in \mathbb{C}^{m\times n}$. Neka je

$$A = \begin{bmatrix} 0 & C^* \\ C & 0 \end{bmatrix} \in \mathbb{C}^{(m+n)\times(m+n)}.$$

Tada postoji unitarna matrica Q, takva da vrijedi

$$Q^*AQ = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min}, -\sigma_1, \ldots, -\sigma_{\min}, \underbrace{0, \ldots, 0}_{|m-n|}).$$

Dokaz. Možemo pretpostaviti da je $m \geq n$. U protivnom radimo s matricom C^*. Neka je

$$U^*CV = \begin{bmatrix} \Sigma \\ 0 \end{bmatrix}$$

singularna dekompozicija matrice C. Konjugiranjem i transponiranjem dobivamo

$$V^*C^*U = [\,\Sigma, \quad 0\,].$$

Množenjem lijevih i desnih strana posljednje dvije jednakosti slijedi

$$V^*(C^*C)V = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_n^2) \in \mathbb{C}^{n\times n},$$

odnosno

$$U^*(CC^*)U = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_n^2, \underbrace{0, \ldots, 0}_{m-n}) \in \mathbb{C}^{m\times m}.$$

Označimo matricu prvih n stupaca matrice U s U_1, a matricu preostalih $m-n$ stupaca s U_2. Dakle $U = [\,U_1, \quad U_2\,]$. Neka je

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} V & V & 0 \\ U_1 & -U_1 & \sqrt{2}U_2 \end{bmatrix}.$$

Q je unitarna matrica jer vrijedi

$$\begin{aligned} Q^*Q &= \frac{1}{2} \begin{bmatrix} V^* & U_1^* \\ V^* & -U_1^* \\ 0 & \sqrt{2}U_2^* \end{bmatrix} \begin{bmatrix} V & V & 0 \\ U_1 & -U_1 & \sqrt{2}U_2 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} V^*V + U_1^*U_1 & V^*V - U_1^*U_1 & \sqrt{2}U_1^*U_2 \\ V^*V - U_1^*U_1 & V^*V + U_1^*U_1 & -\sqrt{2}U_1^*U_2 \\ \sqrt{2}U_2^*U_1 & -\sqrt{2}U_2^*U_1 & 2U_2^*U_2 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 2I_n & 0 & 0 \\ 0 & 2I_n & 0 \\ 0 & 0 & 2I_{m-n} \end{bmatrix} = I. \end{aligned}$$

Izračunajmo još

$$\begin{aligned} Q^*AQ &= \frac{1}{2} \begin{bmatrix} V^* & U_1^* \\ V^* & -U_1^* \\ 0 & \sqrt{2}U_2^* \end{bmatrix} \begin{bmatrix} 0 & C^* \\ C & 0 \end{bmatrix} \begin{bmatrix} V & V & 0 \\ U_1 & -U_1 & \sqrt{2}U_2 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} U_1^*C & V^*C^* \\ -U_1^*C & V^*C^* \\ \sqrt{2}U_2^*C & 0 \end{bmatrix} \begin{bmatrix} V & V & 0 \\ U_1 & -U_1 & \sqrt{2}U_2 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} U_1^*CV + V^*C^*U_1 & U_1^*CV - V^*C^*U_1 & \sqrt{2}V^*C^*U_2 \\ -U_1^*CV + V^*C^*U_1 & -U_1^*CV - V^*C^*U_1 & \sqrt{2}V^*C^*U_2 \\ \sqrt{2}U_2^*CV & \sqrt{2}U_2^*CV & 0 \end{bmatrix}. \end{aligned}$$

Produkt $V^*C^*U_2$ je nul-matrica jer je

$$V^*C^*U_2 = V^*C^* \underbrace{UU^*}_{I} U_2 = [\,\Sigma, \quad 0\,] \begin{bmatrix} 0 \\ I_{m-n} \end{bmatrix} = 0.$$

Zbog toga vrijedi $U_2^*CV = (V^*C^*U_2)^* = 0$. Pošto su $U_1^*CV = \Sigma$ i $V^*C^*U_1 = \Sigma^*$ kvadratne realne i dijagonalne, one su i jednake. Stoga dobivamo

$$\begin{aligned} Q^*AQ &= \frac{1}{2}\begin{bmatrix} \Sigma+\Sigma & 0 & 0 \\ 0 & -\Sigma-\Sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \Sigma & 0 & 0 \\ 0 & -\Sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &= \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n, -\sigma_1, -\sigma_2, \ldots, -\sigma_n, \underbrace{0, \ldots, 0}_{m-n}). \end{aligned}$$

■

Dakle, netrivijalne vlastite vrijednosti matrice A su pozitivne i negativne netrivijalne singularne vrijednosti matrice C, dok se iz matrice Q lako odrede lijevi i desni singularni vektori od C i obratno.

Prethodni teorem omogućava da mnoge rezultate o vlastitim vrijednostima hermitske matrice formuliramo, uz prirodne modifikacije, za singularne vrijednosti proizvoljne matrice.

2.3.5. Neke nejednakosti sa singularnim vrijednostima

U ovom odjeljku prikazat ćemo dva teorema. Rezultat prvog teorema je jedna od mnogobrojnih posljedica Fischerove karakterizacije vlastitih vrijednosti hermitskih matrica. Prvo ćemo formulirati Weylov teorem, a njegov dokaz se može pronaći u poglavlju o vlastitim vrijednostima.

Teorem 2.3.6 *Neka su H i M hermitske matrice i neka je $\widetilde{H} = H + M$. Neka su redom vlastite vrijednosti matrica H, M, i $\widetilde{H}$ $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ i $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_n$. Tada vrijedi*

$$\lambda_i + \mu_n \leq \tilde{\lambda}_i \leq \lambda_i + \mu_1, \quad i = 1, \ldots, n.$$

Dokaz. Vidi npr. [8, teorem 3.14 (str. 25–26)]. ■

Teorem 2.3.6 ima mnogo važnih posljedica. Ovdje ćemo dokazati tek dvije.

Teorem 2.3.7 *Neka matrica $C \in \mathbb{C}^{m\times n}$ ima particiju*

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \begin{matrix} \}k \\ \}m-k \end{matrix}.$$

Tada vrijedi

$$\begin{aligned} &\sigma_i(C_1) \leq \sigma_i(C), \quad 1 \leq i \leq \min\{n, k\} \\ &\sigma_j(C_2) \leq \sigma_j(C), \quad 1 \leq j \leq \min\{n, m-k\}. \end{aligned}$$

Dokaz. Kvadrati singularnih vrijednosti od C jednaki su vlastitim vrijednostima od C^*C, a kvadrati singularnih vrijednosti od C_1 (C_2) su vlastite vrijednosti od $C_1^*C_1$ ($C_2^*C_2$). Zamijetimo da je

$$C^*C = C_1^*C_1 + C_2^*C_2.$$

Matrica $C_2^*C_2$ ($C_1^*C_1$) je pozitivno semidefinitna. Prema teoremu 2.3.6 slijedi da su vlastite vrijednosti od C^*C, uzete u nepadajućem poretku, veće ili jednake od odgovarajućih vlastitih vrijednosti matrice $C_1^*C_1$ ($C_2^*C_2$). ■

Zanimljivo je vidjeti koji je odnos između singularnih vrijednosti produkta dviju matrica i singularnih vrijednosti samih matrica. O tome govori sljedeći teorem.

Teorem 2.3.8 *Neka su $A \in \mathbb{C}^{m\times k}$, $B \in \mathbb{C}^{k\times n}$ i $C = AB$. Tada vrijedi*

$$\sigma_i(AB) \leq \sigma_1(A)\sigma_i(B), \quad \sigma_i(AB) \leq \sigma_i(A)\sigma_1(B), \quad 1 \leq i \leq \min\{m, n, k\}.$$

Dokaz. Druga tvrdnja slijedi iz prve zbog

$$\sigma_i(C) = \sigma_i(C^*) = \sigma_i(B^*A^*) \leq \sigma_1(B^*)\sigma_i(A^*) = \sigma_1(B)\sigma_i(A), \quad 1 \leq i \leq \min\{m, n, k\}.$$

Prvu tvrdnju dokazat ćemo uspoređivanjem vlastitih vrijednosti matrica C^*C s vlastitim vrijednostima od $\|A\|_2^2 B^*B$. Neka je $A^*A = Q\Lambda^2Q^*$ spektralna dekompozicija od A^*A. Tada je

$$M = Q(\|A\|_2^2 I - \Lambda^2)Q^*$$

hermitska pozitivno semidefinitna matrica za koju je

$$A^*A + M = \|A\|_2^2 I.$$

Vrijedi

$$\|A\|_2^2 B^*B = B^*(A^*A + M)B = C^*C + B^*MB.$$

Uočimo da je

$$x^*B^*MBx = (Bx)^*M(Bx) \geq 0,$$

pa je B^*MB, također, pozitivno semidefinitna. Na osnovu teorema 2.3.6 zaključujemo da su vlastite vrijednosti od $\|A\|^2 B^*B = \sigma_1(A)^2 B^*B$ veće ili jednake od odgovarajućih vlastitih vrijednosti od C^*C. ■

2.3.6. Generalizirani inverz

Singularna dekompozicija ima direktnu primjenu u računanju generaliziranog inverza opće matrice. Generalizirani inverz koristan je alat u raznim područjima linearne algebre, npr. linearnom problemu najmanjih kvadrata. Na ovom mjestu uvest

ćemo pojam generaliziranog inverza $A^\dagger$ matrice A i navesti neka njegova osnovna svojstva.

Dobro je poznato, da za regularnu matricu $A \in \mathbb{C}^{n\times n}$ postoji jedinstvena matrica X koja zadovoljava

$$AX = XA = I. \tag{2.3.3}$$

Matrica X je inverz matrice matrice A i označava se s A^{-1}.

Prirodno je pokušati proširiti pojam inverza i na matrice koje nisu regularne ili čak nisu niti kvadratne. To je moguće postići zahtijevajući da matrica X zadovoljava ponešto drugačije (oslabljene) uvjete nego što je (2.3.3). Najpoznatiji generalizirani inverz je tzv. Moore–Penroseov inverz koji je određen sa sljedeća četiri uvjeta (tzv. **Penroseovi uvjeti** [8]):

1. $AXA = A$,
2. $XAX = X$,
3. $(AX)^* = AX$,
4. $(XA)^* = XA$.

Pokazat ćemo da prethodna četiri uvjeta na jedinstven način određuju matricu X koju onda označavamo s $A^\dagger$. U slijedećem teoremu određujemo eksplicitnu formulu za generalizirani inverz $A^\dagger$.

Teorem 2.3.9 *Neka je $A \in \mathbb{C}^{m\times n}$. Tada postoji jedinstvena matrica $X \in \mathbb{C}^{n\times m}$, koja zadovoljava Penroseove uvjete 1–4. Ta matrica ima oblik*

$$A^\dagger = V \begin{bmatrix} \Sigma_+^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^*,$$

pri čemu je

$$A = U \begin{bmatrix} \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix} V^*$$

singularna dekompozicija matrice A.

Dokaz. Za matricu X koja zadovoljava, npr. i-ti i j-ti Penroseov uvjet koristit ćemo oznaku $A^{(i,j)}$ i zvati ga (i,j)-inverz. Tako će, na primjer, (1)-inverz $A^{(1)}$ zadovoljavati samo prvi Penroseov uvjet, a $(1,2,4)$-inverz, $A^{(1,2,4)}$, zadovoljavat će prvi, drugi i četvrti uvjet. Dakle $A^{(1,2,3,4)}$ će zadovoljavati sva četiri uvjeta, pa će ta matrica biti generalizirani inverz matrice A.

Budući da je $A^{(1)} \in \mathbb{C}^{n\times m}$ matricu $A^{(1)}$ možemo zapisati u obliku

$$A^{(1)} = V \begin{bmatrix} T & K \\ L & M \end{bmatrix} U^*.$$

Uvrštavanjem ovog izraza za $A^{(1)}$ u prvi Penroseov uvjet, dobivamo

$$AA^{(1)}A = U \begin{bmatrix} \Sigma_+ T \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix} V^* = A = U \begin{bmatrix} \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix} V^*.$$

Dakle je $T = \Sigma_+^{-1}$, tj. svaki (1)-inverz ima oblik

$$A^{(1)} = V \begin{bmatrix} \Sigma_+^{-1} & K \\ L & M \end{bmatrix} U^*,$$

gdje su K, L i M proizvoljne. Izračunajmo sada $(1,2)$-inverz. On je i (1)-inverz pa polazimo od te matrice. Iz drugog Penroseovog uvjeta slijedi

$$A^{(1,2)} = V \begin{bmatrix} \Sigma_+^{-1} & K \\ L & M \end{bmatrix} U^* = A^{(1,2)}AA^{(1,2)} = V \begin{bmatrix} \Sigma_+^{-1} & K \\ L & L\Sigma_+ K \end{bmatrix} U^*,$$

pa zaključujemo da je $M = L\Sigma_+ K$, dok su K i L još uvijek proizvoljne matrice.

Na isti način odredit ćemo i $(1,2,3)$-inverz. Polazimo od

$$A^{(1,2,3)} = V \begin{bmatrix} \Sigma_+^{-1} & K \\ L & L\Sigma_+ K \end{bmatrix} U^*.$$

Treći Penroseov uvjet se zapisuje u obliku

$$(AA^{(1,2,3)})^* = U \begin{bmatrix} I & 0 \\ K^*\Sigma_+ & 0 \end{bmatrix} U^* = A^{(1,2,3)}A = U \begin{bmatrix} I & \Sigma_+ K \\ 0 & 0 \end{bmatrix} U^*,$$

odakle zaključujemo da svaki $(1,2,3)$-inverz ima oblik

$$A^{(1,2,3)} = V \begin{bmatrix} \Sigma_+^{-1} & 0 \\ L & 0 \end{bmatrix} U^*.$$

Pritom je L proizvoljna podmatrica.

Na isti način, iz četvrtog Penroseovog uvjeta zaključujemo da svaki $(1,2,4)$-inverz ima oblik

$$A^{(1,2,4)} = V \begin{bmatrix} \Sigma_+^{-1} & K \\ 0 & 0 \end{bmatrix} U^*,$$

gdje je K proizvoljna podmatrica.

Konačno, iz općeg oblika $(1,2,3)$-inverza i $(1,2,4)$-inverza zaključujemo da je

$$A^\dagger = A^{(1,2,3,4)} = V \begin{bmatrix} \Sigma_+^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^*. \tag{2.3.4}$$

Dakle, ako $A^\dagger$ postoji nužno ima oblik kao u relaciji (2.3.4). Iz dokaza međutim slijedi da $A^{(1,2,3,4)}$ zadovoljava sve sve Penroseove uvjete pa je $A^\dagger = A^{(1,2,3,4)}$. Kako u dokazu ništa nije proizvoljno, slijedi da je $A^\dagger$ relacijom (2.3.4) na jedinstven način

određena iz singularne dekompozicije matrice A. Sama singularna dekompozicija je prema teoremu 2.3.2 određena do na množenje matrica U i V s desna s blok-dijagonalnim unitarnim matricama. Međutim, kao što A ne ovisi o toj slobodi u matricama U i V, tako ne ovisi ni $A^\dagger$. ■

Sada ćemo navesti neka osnovna svojstva generaliziranog inverza koja se lako dokazuju ili direktno iz Penroseovih uvjeta ili iz relacije (2.3.4).

Teorem 2.3.10 *Za proizvoljnu matricu A vrijedi:*

1. $(A^\dagger)^\dagger = A$,
2. $(\bar{A})^\dagger = \overline{(A^\dagger)}$,
3. $(A^T)^\dagger = (A^\dagger)^T$,
4. $\operatorname{rang}(A) = \operatorname{rang}(A^\dagger) = \operatorname{rang}(AA^\dagger) = \operatorname{rang}(A^\dagger A)$,
5. $(AA^*)^\dagger = (A^*)^\dagger A^\dagger$, $(A^*A)^\dagger = A^\dagger (A^*)^\dagger$,
6. $(AA^*)^\dagger AA^* = AA^\dagger$, $(A^*A)^\dagger A^*A = A^\dagger A$.
7. *Ako matrica $A \in \mathbb{C}^{m\times n}$ ima rang n, tada je $A^\dagger = (A^*A)^{-1}A^*$ i $A^\dagger A = I_n$.*
8. *Ako matrica $A \in \mathbb{C}^{m\times n}$ ima rang m, tada je $A^\dagger = A^*(AA^*)^{-1}$ i $AA^\dagger = I_m$.*
9. *Ako je $A = FG^*$ i $\operatorname{rang}(A) = \operatorname{rang}(F) = \operatorname{rang}(G)$, tada je*
$$A^\dagger = G(F^*AG)^{-1}F^* \quad i \quad A^\dagger = (G^\dagger)^* F^\dagger.$$
10. *Ako su U i V unitarne matrice, tada je*
$$(UAV)^\dagger = V^* A^\dagger U^*.$$

2.4. Vektorske i matrične norme

2.4.1. Vektorske norme

Vektorske i matrične norme osnovno su sredstvo koje koristimo kod ocjene grešaka vezanih uz numeričke metode, posebno u numeričkoj linearnoj algebri.

Definicija 2.4.1 (Vektorska norma) *Vektorska norma je svaka funkcija $\|\cdot\| : \mathbb{C}^n \to \mathbb{R}$ koja zadovoljava sljedeća svojstva:*

1. $\|x\| \geq 0$, $\forall x \in \mathbb{C}^n$, *a jednakost vrijedi ako i samo ako je $x = 0$,*
2. $\|\alpha x\| = |\alpha|\,\|x\|$, $\forall \alpha \in \mathbb{R}$, $\forall x \in \mathbb{C}^n$,

3. $\|x+y\| \leq \|x\| + \|y\|$, $\forall x, y \in \mathbb{C}^n$. *Ova je nejednakost poznatija pod imenom nejednakost trokuta (zbroj duljina bilo koje dvije stranice trokuta veći je od duljine treće stranice).*

Analogno se definira vektorska norma na bilo kojem vektorskom prostoru V nad poljem $F = \mathbb{R}$ ili $\mathbb{C}$.

Neka je x vektor iz $\mathbb{C}^n$ s komponentama x_i, $i = 1, \ldots, n$, u oznaci $x = (x_1, \ldots, x_n)^T$, ili, skraćeno $x = [x_i]$. U numeričkoj linearnoj algebri najčešće se koriste sljedeće tri norme:

1. 1-norma ili ℓ_1 norma, u engleskom govornom području poznatija kao "Manhattan" ili "taxi-cab" norma
$$\|x\|_1 = \sum_{i=1}^{n} |x_i|,$$
2. 2-norma ili ℓ_2 norma ili euklidska norma
$$\|x\|_2 = (x^*x)^{1/2} = \sqrt{\sum_{i=1}^{n} |x_i|^2},$$
3. ∞-norma ili ℓ_∞ norma
$$\|x\|_\infty = \max_{i=1,\ldots,n} |x_i|.$$

Primijetite da je samo 2-norma izvedena iz skalarnog produkta, dok ostale dvije to nisu.

2-norma ima dva bitna svojstva koje je čine posebno korisnom. Ona je invarijantna na unitarne transformacije vektora x, tj. ako je Q unitarna matrica ($Q^*Q = QQ^* = I$), onda je
$$\|Qx\|_2 = (x^*Q^*Qx)^{1/2} = (x^*x)^{1/2} = \|x\|_2.$$
Također ona je diferencijabilna za sve $x \neq 0$, s gradijentom
$$\nabla\|x\|_2 = \frac{x}{\|x\|_2}.$$

Sve ove tri norme specijalni su slučaj Hölderove p-norme (ℓ_p norme) definirane s:
$$\|x\|_p = \Big(\sum_{i=1}^{n} |x_i|^p\Big)^{1/p}, \quad p \geq 1.$$
Za Hölderove p-normu vrijedi i poznata Hölderova nejednakost
$$|x^*y| \leq \|x\|_p \|y\|_q, \quad \frac{1}{p} + \frac{1}{q} = 1.$$

Posebni slučaj Hölderove nejednakosti za $p = q = 2$ je Cauchy-Schwarzova nejednakost

$$|x^* y| \leq \|x\|_2 \|y\|_2.$$

Koliko se dvije p-norme međusobno razlikuju, pokazuje sljedeća nejednakost, koja se može dostići. Neka su α i β dvije p norme takve da je $\alpha \leq \beta$. Tada vrijedi

$$\|x\|_\beta \leq \|x\|_\alpha \leq n^{(1/\alpha - 1/\beta)} \|x\|_\beta.$$

Ova se nejednakost često proširuje i zapisuje tablicom $\|x\|_\alpha \leq C_M \|x\|_\beta$, gdje su C_M-ovi

$\alpha \backslash \beta$	1	2	∞
1	1	$\sqrt{n}$	n
2	1	1	$\sqrt{n}$
∞	1	1	1

Primijetite da sve p-norme ovise samo o apsolutnoj vrijednosti komponenti x_i, pa je p-norma rastuća funkcija apsolutnih vrijednosti komponenti x_i. Označimo s $|x|$ vektor apsolutnih vrijednosti komponenti vektora x, tj. $|x| = [|x_i|]$. Za vektore apsolutnih vrijednosti (u $\mathbb{R}^n$) možemo uvesti parcijalni uređaj relacijom

$$|x| \leq |y| \iff |x_i| \leq |y_i|, \quad \forall i = 1, \ldots, n.$$

Definicija 2.4.2 (Monotona i apsolutna norma) *Norma na $\mathbb{C}^n$ je monotona ako vrijedi*

$$|x| \leq |y| \implies \|x\| \leq \|y\|, \quad \forall x, y \in \mathbb{C}^n.$$

Norma na $\mathbb{C}^n$ je apsolutna ako vrijedi

$$\| \, |x| \, \| = \|x\|, \quad \forall x \in \mathbb{C}^n.$$

Bauer, Stoer i Witzgall dokazali su netrivijalni teorem koji pokazuje da su ta dva svojstva ekvivelentna.

Teorem 2.4.1 *Norma na $\mathbb{C}^n$ je monotona ako i samo ako je apsolutna.*

Definicija vektorskih normi u sebi ne sadrži zahtjev da je vektorski prostor iz kojeg su vektori konačno dimenzionalan. Na primjer, norme definirane na vektorskom prostoru neprekidnih funkcija na $[a, b]$ (u oznaci $C[a, b]$) definiraju se slično normama na C^n:

1. L_1 norma

$$\|f\|_1 = \int_a^b |f(t)| \, dt,$$

2. L_2 norma
$$\|f\|_2 = \Big(\int_a^b |f(t)|^2 \, dt \Big)^{1/2},$$

3. L_∞ norma
$$\|f\|_\infty = \max\{|f(x)| \mid x \in [a,b]\},$$

4. L_p norma
$$\|f\|_p = \Big(\int_a^b |f(t)|^p \, dt \Big)^{1/p}, \quad p \geq 1.$$

2.4.2. Matrične norme

Zamijenimo li u definiciji 2.4.1 vektor $x \in \mathbb{C}^n$ matricom $A \in \mathbb{C}^{m\times n}$, dobivamo matričnu normu.

Definicija 2.4.3 (Matrična norma) *Matrična norma je svaka funkcija* $\|\cdot\| : \mathbb{C}^{m\times n} \to \mathbb{R}$ *koja zadovoljava sljedeća svojstva:*

1. $\|A\| \geq 0$, $\forall A \in \mathbb{C}^{m\times n}$, *a jednakost vrijedi ako i samo ako je* $A = 0$,
2. $\|\alpha A\| = |\alpha| \, \|A\|$, $\forall \alpha \in \mathbb{R}$, $\forall A \in \mathbb{C}^{m\times n}$,
3. $\|A + B\| \leq \|A\| + \|B\|$, $\forall A, B \in \mathbb{C}^{m\times n}$.

Za matričnu normu ćemo reći da je konzistentna ako vrijedi

4. $\|AB\| \leq \|A\| \, \|B\|$

kad god je matrični produkt AB definiran. Oprez, norme od A, B i AB ne moraju biti definirane na istom prostoru (dimenzije)!

Neki autori smatraju da je i ovo posljednje svojstvo sastavni dio definicije matrične norme (tada to svojstvo obično zovu submultiplikativnost). Ako su ispunjena samo prva tri svojstva, onda to zovu generalizirana matrična norma.

Matrične norme mogu nastati na dva različita načina. Ako matricu A promatramo kao vektor s $m \times n$ elemenata, onda, direktna primjena vektorskih normi (uz oznaku a_{ij} matričnog elementa u i-tom retku i j-tom stupcu) daje sljedeće definicije:

1. ℓ_1 norma
$$|||A|||_1 := \|A\|_S = \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|,$$

2. ℓ_2 norma (euklidska, Frobeniusova, Hilbert–Schmidtova, Schurova)

$$|||A|||_2 := \|A\|_F = (\mathrm{tr}(A^*A))^{1/2} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2},$$

3. ℓ_∞ norma

$$|||A|||_\infty := \|A\|_M = \max_{\substack{i=1,\dots,m\\ j=1,\dots,n}} |a_{ij}|.$$

tr je oznaka za trag matrice – zbroj dijagonalnih elemenata matrice.

Pokažimo da ℓ_1 i ℓ_2 norma zadovoljavaju svojstvo konzistentnosti, a ℓ_∞ norma ga ne zadovoljava. Vrijedi

$$\begin{aligned}
\|AB\|_S &= \sum_{i=1}^{m}\sum_{j=1}^{s}\sum_{k=1}^{n}|a_{ik}b_{kj}| \le \sum_{i=1}^{m}\sum_{j=1}^{s}\sum_{k=1}^{n}\sum_{\ell=1}^{n}|a_{ik}b_{\ell j}| \\
&\le \sum_{i=1}^{m}\sum_{k=1}^{n}|a_{ik}|\sum_{\ell=1}^{n}\sum_{j=1}^{s}|b_{\ell j}| = \|A\|_S\,\|B\|_S, \\
\|AB\|_F^2 &= \sum_{i=1}^{m}\sum_{j=1}^{s}\left|\sum_{k=1}^{n}a_{ik}b_{kj}\right|^2 \le \sum_{i=1}^{m}\sum_{j=1}^{s}\Big(\sum_{k=1}^{n}|a_{ik}|^2\Big)\Big(\sum_{\ell=1}^{n}|b_{\ell j}|^2\Big) \\
&= \Big(\sum_{i=1}^{m}\sum_{k=1}^{n}|a_{ik}|^2\Big)\Big(\sum_{\ell=1}^{n}\sum_{j=1}^{s}|b_{\ell j}|^2\Big) = \|A\|_F^2\,\|B\|_F^2.
\end{aligned}$$

Primijetite da se u dokazu da je Frobeniusova norma konzistentna koristila Cauchy-Schwarzova nejednakost.

Pokažimo na jednom primjeru da ℓ_∞ norma ne zadovoljava svojstvo konzistentnosti. Za matrice

$$A = B = \begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} \quad \text{je} \quad AB = \begin{bmatrix}2 & 2\\ 2 & 2\end{bmatrix},$$

pa je

$$\|AB\|_M = 2, \quad \|A\|_M\,\|B\|_M = 1.$$

Ipak i od $\|\cdot\|_M$ se može napraviti konzistentna norma. Definiramo li

$$|||A||| = m\,\|A\|_M,$$

vrijedi

$$\begin{aligned}
|||AB||| &= m \max_{\substack{i=1,\dots,m\\ j=1,\dots,s}}\left|\sum_{k=1}^{n}a_{ik}b_{kj}\right| \le m \max_{\substack{i=1,\dots,m\\ j=1,\dots,s}}\sum_{k=1}^{n}|a_{ik}b_{kj}| \\
&\le m \max_{\substack{i=1,\dots,m\\ j=1,\dots,s}}\sum_{k=1}^{n}\|A\|_M\,\|B\|_M = (m\,\|A\|_M)\,(n\,\|B\|_M) = |||A|||\,|||B|||.
\end{aligned}$$

S druge strane, matrične norme možemo dobiti kao **operatorske norme** iz odgovarajućih vektorskih korištenjem definicije

$$\|A\| = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|} (\text{ili } = \max_{\|x\|=1} \|Ax\|). \tag{2.4.1}$$

Kad se uvrste odgovarajuće vektorske norme u (2.4.1), dobivamo

1. matrična 1-norma, "maksimalna stupčana norma"

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}|,$$

2. matrična 2-norma, spektralna norma

$$\|A\|_2 = (\rho(A^*A))^{1/2} = \sigma_{\max}(A),$$

3. matrična ∞-norma, "maksimalna retčana norma"

$$\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^{n} |a_{ij}|,$$

pri čemu je ρ oznaka za spektralni radijus kvadratne matrice (maksimalna po apsolutnoj vrijednosti svojstvena vrijednost)

$$\rho(B) = \max\{|\lambda| \mid \det(B - \lambda I) = 0\}, \quad (B \text{ kvadratna!}), \tag{2.4.2}$$

a σ je standardna oznaka za tzv. singularnu vrijednost matrice. Detaljnu definiciju što je to singularna vrijednost, dobit ćete u poglavlju koje će se baviti dekompozicijom singularnih vrijednosti.

Matrična 2-norma se teško računa, (trebalo bi naći po apsolutnoj vrijednosti najveću svojstvenu vrijednost), pa je uobičajeno da se ona procjenjuje korištenjem ostalih normi.

Tablica ovisnosti koja vrijedi među matričnim normama je: $\|A\|_\alpha \leq C_M \|A\|_\beta$, gdje su C_M-ovi

$\alpha\backslash\beta$	1	2	∞	F	M	S
1	1	$\sqrt{m}$	m	$\sqrt{m}$	m	1
2	$\sqrt{n}$	1	$\sqrt{m}$	1	$\sqrt{mn}$	1
∞	n	$\sqrt{n}$	1	$\sqrt{n}$	n	1
F	$\sqrt{n}$	$\sqrt{\text{rang}(A)}$	$\sqrt{m}$	1	$\sqrt{mn}$	1
M	1	1	1	1	1	1
S	n	$\sqrt{mn\,\text{rang}(A)}$	m	$\sqrt{mn}$	mn	1

Posebno su važne **unitarno invrijantne norme**, tj. one za koje vrijedi

$$\|UAV\| = \|A\|, \tag{2.4.3}$$

za sve unitarne matrice U i V.

Dvije najpoznatije unitarno invarijantne norme su Frobeniusova i spektralna norma. Pokažimo to. Kvadrat Frobeniusove norme matrice A možemo promatrati kao zbroj kvadrata normi stupaca a_j:

$$\|A\|_F^2 = \sum_{j=1}^{n} \|a_j\|^2.$$

S druge strane, za svaku unitarnu matricu U vrijedi

$$\|Ua_j\|_2^2 = a_j^* U^* U a_j = a_j^* a_j = \|a_j\|_2^2.$$

Objedinimo li te relacije, dobivamo

$$\|UA\|_F^2 = \sum_{j=1}^{n} \|Ua_j\|^2 = \sum_{j=1}^{n} \|a_j\|^2 = \|A\|_F^2.$$

Konačno, vrijedi

$$\|UAV\|_F^2 = \|AV\|_F^2 = \|V^*A^*\|_F^2 = \|A^*\|_F^2 = \|A\|_F^2.$$

Da bismo dokazali da je matrična 2-norma unitarno ekvivalentna, potrebno je pokazati da transformacije sličnosti čuvaju svojstvene vrijednosti matrice. Ako je S nesingularna (kvadratna) matrica, a B kvadratna, onda je matrica $S^{-1}BS$ slična matrici B. Ako je spektralna faktorizacija matrice $S^{-1}BS$

$$S^{-1}BSX = X\Lambda,$$

pri čemu je X matrica svojstvenih vektora, a Λ svojstvenih vrijednosti. Množenjem sa S slijeva, dobivamo

$$B(SX) = (SX)\Lambda,$$

tj. matrica svojstvenih vektora je SX, dok su svojstvene vrijednosti ostale nepromijenjene. Primijetite da za unitarne matrice vrijedi $V^* = V^{-1}$.

Za matričnu 2-normu, onda vrijedi

$$\|UAV\|_2 = (\rho(V^*A^*U^*UAV))^{1/2} = (\rho(V^*A^*AV))^{1/2}.$$

Budući da je V unitarna, A^*A i V^*A^*AV su unitarno ekvivalentne, pa je

$$\|UAV\|_2 = (\rho(A^*A))^{1/2} = \|A\|_2.$$

3. Greške u numeričkom računanju

Da bismo mogli ocjenjivati izračunava li neki algoritam, implementiran na računalu (kao program), traženo rješenje problema s dovoljnom točnošću, potrebno je upoznati pojmove apsolutne i relativne greške, greške unaprijed i unazad, stabilnosti algoritma, uvjetovanosti problema itd. Svi ti pojmovi nastajali su vremenski postupno, a uglavnom su vezani uz računanje na računalima.

3.1. Tipovi grešaka

U praktičnom računanju postoje dvije osnovne vrste grešaka: greške zbog polaznih aproksimacija i greške zaokruživanja.

3.1.1. Greške zbog polaznih aproksimacija

Taj tip grešaka često se javlja kod rješavanja praktičnih problema. Te greške možemo podijeliti u sljedeće klase: greške modela, greške metode i greške u polaznim podacima.

Greške modela

Najčešće nastaju zamjenom složenih sustava (u skali veličina, od svemirskih sustava do atomskih struktura, u skali složenosti, od reakcija u atomskim centralama do npr. kemijskih i bioloških fenomena u stanicama raka, u svakidašnjem životu: od sustava koji određuje prognozu vremena do sustava koji daje kretanje cijena na burzama) jednostavnijima koje možemo opisati matematičkim zapisom. Često su stvarni fenomeni takvi da ih ni približno ne možemo opisati današnjim matematičkim teorijama, ili su previše složeni za današnji stupanj matematike. Stoga se vrše razna pojednostavljenja, s jedne strane da bismo uopće opisali fenomene matematičkim zapisom, a s druge strane da bismo i njih pojednostavili kako bismo

ih uopće mogli riješiti. Npr. kod gibanja u zemaljskim uvjetima često se zanemaruje utjecaj otpora zraka. Često se dobri modeli zamjenjuju slabijim da bi se mogle primijeniti numeričke metode (na primjer, sustavi nelinearnih parcijalnih diferencijalnih jednadžbi se linearizira). Pogreške u modelu mogu nastati i kod upotrebe modela u graničnim slučajevima. Na primjer, kod matematičkog njihala se $\sin x$ aproksimira s x, što vrijedi samo za male kuteve, a upotrebljava se, recimo i za veće kutove npr. za 15°. Pogreške modela su neuklonjive, i na korisnicima je da procijene dobivaju li primjenom danog modela očekivane rezultate.

Primjer 3.1.1 *Među prvim primjenama trenutno jednog od najbržih računala na svijetu bilo je određivanje trodimenzionalne strukture i elektronskog stanja ugljik-*36 *fulerena (engl. carbon-*36 *fullerene) — jednog od najmanjih, ali i najstabilnijih članova iz redova jedne vrste spojeva (engl. buckminsterfullerene). Primjena tog spoja može biti višestruka, od supravodljivosti na visokim temperaturama do preciznog doziranja lijekova u stanice raka.*

Prijašnja istraživanja kvantnih kemičara dala su dvije moguće strukture tog spoja. Eksperimentalna mjerenja pokazivala su da bi jedna struktura trebala biti stabilnija, a teoretičari su tvrdili da bi to trebala biti druga struktura. Naravno, te dvije strukture imaju različita kemijska svojstva. Prijašnja računanja, zbog pojednostavljivanja i interpolacije, kao odgovor davala su prednost "teoretskoj" strukturi. Definitivan odgovor, koji je proveden računanjem bez pojednostavljivanja pokazao je da je "eksperimentalna" struktura stabilnija.

Greške metode

Te greške nastaju kad se beskonačni procesi zamjenjuju konačnim. Također, nastaju kod računanja veličina koje su definirane limesom, poput derivacija i integrala ili su definirane limesima konvergentnih nizova ili produkata. Velik broj numeričkih metoda za aproksimaciju funkcija i rješavanje jednadžbi upravo je tog oblika.

Greške koja nastaju zamjenom beskonačnog nečim konačnim, obično dijelimo u dvije kategorije.

Greške diskretizacije nastaju zamjenom kontinuuma konačnim diskretnim skupom točaka, ili kad se "beskonačno" mala veličina (najčešće u oznaci h ili ε) zamijeni nekim konkretnim malim brojem. Također, te greške nastaju kad se derivacija zamijeni podijeljenom razlikom, diferencijalna jednadžba diferencijskom jednadžbom, integral nekom kvadraturnom formulom. Još jednostavniji, tipični primjer greške diskretizacije je aproksimacija funkcije f na intervalu (segmentu) $[a, b]$, vrijednostima te funkcije na konačnom skupu točaka (tzv. mreži)

$$\{x_1, \ldots, x_n\} \subset [a, b].$$

Greške odbacivanja nastaju zamjenom beskonačnog niza, reda, produkta konačnim nizom, sumom, produktom (tada kažemo da odbacujemo ostatak niza, reda, produkta).

Grubo rečeno, diskretizacija je vezana za kontinuum (npr. skupovi $\mathbb{R}$ i $\mathbb{C}$), a odbacivanje za diskretnu beskonačnost ($\mathbb{N}$, $\mathbb{Z}$). Objekti koji nedostaju pri tim zamjenama tvore tip grešaka koji se zovu **greške metode**.

Greške u polaznim podacima

Te greške imaju izvor u mjerenjima fizičkih veličina, te smještanju podataka u računalo, a također i u prethodnim računanjima. Naime, često su izlazne vrijednosti iz nekog prethodnog računanja, ulazni podaci za novo računanje. Greške mjerenja ili smještanja u računalo daleko je jednostavnije ocijeniti od grešaka koje nastaju uslijed brojnih zaokruživanja u toku računanja.

3.1.2. Greške zaokruživanja

Te greške nastaju u računalima, jer ona koriste **konačnu aritmetiku** ili preciznije binarnu **aritmetiku s pomičnom točkom**, kod koje je unaprijed rezerviran određen broj binarnih mjesta za eksponent i za mantisu. Stoga se svaka računska operacija u kojoj sudjeluju dva broja izračunava s nekom malom greškom, koja može biti i nula. Ako je nula, tada je rezultat te operacije na danim operandima egzaktano izračunat. Ako nije nula, tada je izračunat s nekom malom greškom koju možemo precizno ocijeniti i koju zovemo **greška zaokruživanja**.

Ako je algoritam složeniji, bit će mnogo računskih operacija. Kod gotovo svake računske operacije postojat će greška zaokruživanja, pa se postavlja pitanje s kojom točnošću ćemo dobiti traženo rješenje? Da bi se dobio odgovor na to pitanje potrebno je istražiti i sam matematički problem i algoritam koji je odabran za računanje rješenja. Pri analizi grešaka koje nastaju u toku računanja koristimo se **teorijom grešaka zaokruživanja**, a osjetljivošću rješenja matematičkog problema koji rješavamo na pomake u polaznim podacima bavi se **teorija perturbacije** za dani problem. Njihovom usklađenom uporabom često je moguće procijeniti točnost algoritma, koji je korišten na računalu. Ako možemo reći da je točnost izračunatih podataka približno jednaka točnosti polaznih podataka (koje dolaze npr. od mjernih uređaja ili od smještanja u računalo), tada govorimo o **stabilnom algoritmu**.

Kroz polustoljetni razvoj, teorija grešaka zaokruživanja i teorija stabilnosti numeričkih algoritama ušle su u zrelu fazu, pa je potrebno upoznati osnovne pojmove i rezultate u toj grani numeričke matematike. Cilj ovog poglavlja je svladati tehniku ocjenjivanja grešaka zaokruživanja, ukazati na opasnosti i neželjene efekte koji mogu nastupiti kod računanja na računalima i kalkulatorima, te pokazati kako se dokazuje

stabilnost algoritama. Greške zaokruživanja su neizbježne pri svakom zahtjevnijem računanju, jer računala koriste **konačnu aritmetiku**.

Već samo uskladištavanje podataka u računalo dovodi do grešaka jer se svaki broj mora reprezentirati konačnim brojem binarnih znamenaka. Zna se da npr. 1/10 ima u binarnom sustavu prikaz s beskonačno nula i jedinica: $(1/10)_2 = 0.000110011001100110\ldots$ (vidi primjer 3.2.1). Dakle, ne samo iracionalni, već i mnogi racionalni brojevi, čak i oni umjerene veličine i s malo dekadskih znamenaka, ne moraju biti točno reprezentirani u računalu. Kad se jednom polazni brojevi smjeste u računalo, to uglavnom nisu njihove točne, nego približne binarne reprezentacije. Ako je x realni broj, njegova reprezentacija u računalu se obično označava sa $f\ell(x)$.

Pretpostavimo sada da su x i y brojevi koji su smješteni u računalo (to matematički zapisujemo $x = f\ell(x)$ i $y = f\ell(y)$). Tada će svaka aritmetička operacija u računalu između ta dva broja rezultirati brojem koji je najčešće tek aproksimacija točnog rezultata. Tipični slučaj je množenje koje u pravilu daje mantisu rezultata čija je duljina približno zbroj duljina mantisa faktora. Ako smještaj u računalu dopušta p binarnih mjesta za mantisu, tada će svaki od faktora imati mantisu duljine p, a umnožak će imati mantisu duljine $2p$ ili $2p-1$. Kako se rezultat opet sprema u memoriju, zadržat će se samo prvih p znamenaka, a ostale će se odbaciti (s korektnim zaokruživanjem na posljednjem značajnom mjestu). Kod zbrajanja i oduzimanja brojeva koji nisu istog reda veličine, u procesu izvršavanja operacija, mantisa jednog od njih (onog s manjom apsolutnom vrijednosti) bit će pomaknuta za jedno ili više mjesta, pa će rezultat opet imati mantisu duljine koja je veća od p. Konačno, kod dijeljenja kao najkompliciranije operacije, algoritam za dijeljenje u računalu također daje rezultat koji nije točan. Detaljno proučavanje (vidjeti npr. [9, 4]) pokazuje da je kod svake operacije u računalu prisutna greška. Ako je greška nula rezultat je točan. Taj zaključak se zapisuje u obliku

$$f\ell(x \circ y) = (x \circ y)(1+\varepsilon), \quad |\varepsilon| \leq u \quad \circ \in \{+, -, *, /\} \tag{3.1.1}$$

pri čemu su $+, -, *, /$ operacije u računalu, a u je tzv. **preciznost računanja** ili **strojni** u. Pritom greška ε ovisi o operandima x, y i operaciji $\circ$, dok u ovisi o računalu (ili točnije o IEEE standardu ako ga računalo podržava). Iz zadnje relacije vidi se da je u uniformna gornja ograda za sve greške ε na danom računalu. Kasnije ćemo obrazložiti da za računala koja koriste binarnu aritmetiku sa p binarnih znamenaka u mantisi, vrijedi $u = 2^{-p+1}$ ili $u = 2^{-p}$ ovisno o načinu zaokruživanja koje računalo koristi.

Glavna zadaća osobe koja se bavi numeričkim računanjima, je određivanje što bolje aproksimacije rješenja u što kraćem vremenu. Da bi se to ostvarilo, treba imati pod kontrolom sve tipove grešaka koje smo nabrojali, a to znači da treba poznavati sve fenomene koji mogu naići prije ili u tijeku računanja, a vezani su uz netočnosti polaznih podataka, međurezultata te konačnog rezultata.

3.1.3. Apsolutna i relativna greška, značajne znamenke

Neka je $\hat{x}$ neka aproksimacija realnog broja x. Najkorisnije mjere za točnost broja $\hat{x}$ kao aproksimacije od x su **apsolutna greška**

$$G_{\text{aps}}(\hat{x}) = |x - \hat{x}|$$

i **relativna greška**

$$G_{\text{rel}}(\hat{x}) = \frac{|x - \hat{x}|}{|x|},$$

koja nije definirana za $x = 0$. Drugi način zapisivanja relativne greške je

$$G_{\text{rel}}(\hat{x}) = |\rho|, \quad \hat{x} = x(1 + \rho). \tag{3.1.2}$$

Neki autori izostavljaju u definiciji apsolutne greške znak apsolutne vrijednosti. Time se daje (ili zahtijeva) dodatna informacija o predznaku greške, no tada je pridjev "apsolutna" suvišan. Međutim, često točna vrijednost x nije poznata, već se tek zna neka ograda za $|x - \hat{x}|$.

Ako je x poznat ili se zna da je "reda veličine jedan", apsolutna greška je dobra mjera za udaljenost aproksimacije od točne vrijednosti. U znanosti međutim, x često varira između vrlo malih (npr. veličine vezane uz atomsku ili molekularnu strukturu) i velikih (npr. veličine vezane uz svemir i njegove objekte) vrijednosti. Tada je relativna greška često primjerenija. Ona ima dodatno lijepo svojstvo da je nezavisna od skaliranja,

$$\frac{|x - \hat{x}|}{|x|} = \frac{|\alpha x - \alpha \hat{x}|}{|\alpha x|}, \quad \alpha \in \mathbb{R}.$$

Relativna greška povezana je s "brojem točnih značajnih znamenaka". Značajne znamenke u broju su prve netrivijalne (tj. različite od nule) znamenke i one koje slijede iza njih u zapisu. Npr. u broju 6.9990 imamo pet značajnih znamenaka, dok ih u broju 0.0832 imamo tri. Što znači broj "točnih" ili "korektnih" značajnih znamenaka? Npr. ako je

$$\begin{aligned} x &= 1.00000, \quad \hat{x} = 1.00499, \quad G_{\text{rel}} = 4.99 \cdot 10^{-3}, \\ x &= 9.00000, \quad \hat{x} = 8.99899, \quad G_{\text{rel}} = 1.12 \cdot 10^{-4}, \end{aligned}$$

vidimo da se $\hat{x}$ slaže s odgovarajućim x na tri značajne znamenke, a ipak je relativna greška za ta dva slučaja različita čak za faktor 44.

Evo jedne moguće definicije: $\hat{x}$ kao aproksimacija od x ima p korektnih značajnih znamenaka ako se $\hat{x}$ i x zaokružuju na isti broj od p značajnih znamenaka. Zaokružiti broj na p značajnih znamenaka znači zamijeniti ga s najbližim brojem koji ima p značajnih znamenaka. Međutim, prema toj definiciji, brojevi $x = 0.9949$

i $\hat{x} = 0.9951$, se ne slažu u dvije značajne znamenke, ali se slažu u jednoj i u tri značajne znamenke. Dakle, definicija nije dobra.

Pokušajmo zato s ovakvom definicijom: $\hat{x}$ kao aproksimacija od x ima p korektnih značajnih znamenaka ako je $|x-\hat{x}|$ manja od jedne polovine jedinice u p-toj značajnoj znamenci od x. Ova definicija implicira da se 0.123 i 0.127 slažu u dvije značajne znamenke, iako će mnogi tvrditi da se slažu u tri.

Vidimo da je relativna greška preciznija mjera od "slaganja u p značajnih znamenaka", pa bi je trebalo preferirati.

Kad su u pitanju vektori, apsolutna greška vektora $\hat{x}$ kao aproksimacije vektora x, definira se pomoću prikladne norme (npr. euklidske)

$$G_{\text{aps}}(\hat{x}) = \|x - \hat{x}\|_2.$$

Relativna greška je

$$G_{\text{rel}}(\hat{x}) = \frac{\|x - \hat{x}\|}{\|x\|},$$

a relacija

$$\frac{\|x - \hat{x}\|}{\|x\|} < \frac{1}{2} \cdot 10^{-p}$$

implicira da komponente $\hat{x}_i$ za koje vrijedi $|x_i| \approx \|x\|$ imaju približno p točnih značajnih znamenaka.

Ako želimo sve komponente vektora x staviti u isti plan, tada koristimo relativne greške po komponentama, a veličinu

$$\max_i \frac{|x_i - \hat{x}_i|}{|x_i|}$$

zovemo (maksimalna) **relativna greška po komponentama**.

Preciznost i točnost

Ove dvije riječi se često zamijenjuju, pa kad ih već imamo, možemo načiniti sljedeću distinkciju. **Točnost** se odnosi na apsolutnu ili relativnu grešku kojom se aproksimira neka veličina. **Preciznost** je točnost kojom se izvršavaju osnovne računske operacije. Kod računanja u aritmetici pomične točke, preciznost mjerimo pomoću preciznosti računanja u. Kako je preciznost određena brojem bitova u reprezentaciji mantise, ista riječ će se koristiti i za broj bitova u mantisi.

Napomenimo da točnost općenito nije limitirana preciznošću. Naime, pomoću aritmetike dane preciznosti može se simulirati računanje u (proizvoljno) većoj preciznosti. Međutim takve simulacije su skupe (u računskom vremenu) pa nisu od praktične važnosti. Stoga pretpostavljamo da se dana konačna aritmetika ne koristi za simulaciju aritmetike veće preciznosti.

3.2. Aritmetika s pomičnom točkom

Na kalkulatorima se često može izabrati tzv. "znanstvena notacija" brojeva, koja npr. broj -27.77 prikazuje kao $-2.777 \cdot 10^1$ pri čemu je $-$ **predznak** broja (ili mantise), . je **decimalna točka**, 2.777 je **mantisa**, koja se još zove signifikantni ili razlomljeni dio broja, 10 je **baza**, a 1 je **eksponent**. Zapis $x = -27.77 = -2.777 \cdot 10^1$ je kraća oznaka za prikaz broja

$$\begin{aligned} x &= -(2 \cdot 10^1 + 7 \cdot 10^0 + 7 \cdot 10^{-1} + 7 \cdot 10^{-2}) \\ &= -(2 \cdot 10^0 + 7 \cdot 10^{-1} + 7 \cdot 10^{-2} + 7 \cdot 10^{-3}) \cdot 10^1, \end{aligned}$$

a kako vrijedi

$$x = [(-1) \cdot (2 \cdot 10^0 + 7 \cdot 10^{-1} + 7 \cdot 10^{-2} + 7 \cdot 10^{-3})] \cdot 10^1,$$

predznak broja je ujedno i predznak mantise.

Općenito, svaki se realni broj može na jednoznačan način zapisati u obliku $\pm m \times 10^e$ gdje je $1 \leq m < 10$.

Računala za spremanje realnih brojeva koriste sličnu reprezentaciju broja koja se zove **pomična točka**, ali se ne uzima baza 10 već baza 2 (s iznimkom baze 16 kod računala IBM 370 i baze 10 kod većine kalkulatora). Tako je npr.

$$\begin{aligned} y = (11.1011)_2 &= 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} \\ &= (1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5}) \cdot 2^1 \\ &= (1.11011)_2 \cdot 2^1. \end{aligned}$$

Ovdje je $(11.1011)_2$ binarna reprezentacija broja y, a njen oblik u prikazu s pomičnom točkom $(1.11011)_2 \cdot 2^1$. Lako se izračuna da y ima decimalnu reprezentaciju $(3.625)_{10}$, što se uobičajeno kaže da je y broj 3.625.

3.2.1. Pretvaranje decimalne u binarnu reprezentaciju

Ne samo iracionalni, već i mnogi racionalni brojevi imaju beskonačno mnogo znamenaka u decimalnom brojnom sustavu. Npr. $1/3$ ima prikaz $0.\dot{3}$, dok $5/17$ ima prikaz $0.\dot{2}941176470588235\dot{}$, pri čemu se znamenke koje su označene između točaka periodički ponavljaju. Uočimo da se i svaki konačni decimalni prikaz broja može napisati s beskonačno znamenaka, npr. $x = 32.75 = 32.74\dot{9}$. Slična je situacija i s brojevima zapisanim u binarnom sustavu. Međutim, zanimljivo je da mnogi racionalni brojevi imaju u dekadskom sustavu konačni prikaz, dok u binarnom sustavu imaju beskonačni prikaz. Vrijedi li i obrat?

Da bismo se u to uvjerili, izvedimo prvo algoritam za pretvaranje decimalnog oblika u binarni oblik.

Neka x ima konačni binarni prikaz. Tada x možemo prikazati kao sumu njegovog cijelog i razlomljenog dijela,

$$\begin{aligned} x &= (a_k a_{k-1} \cdots a_1 a_0 . b_1 b_2 \cdots b_{l-1} b_l)_2 = x_c + x_r, \\ x_c &= (a_k a_{k-1} \cdots a_1 a_0)_2 \\ &= a_k \cdot 2^k + a_{k-1} \cdot 2^{k-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0, \qquad (3.2.1) \\ x_r &= (.b_1 b_2 \cdots b_{l-1} b_l)_2 \\ &= b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \cdots + b_{l-1} \cdot 2^{-(l-1)} + b_l \cdot 2^{-l}. \qquad (3.2.2) \end{aligned}$$

Ako je zadan cijeli broj x_c u decimalnom obliku, kako odrediti njegove binarne znamenke $a_k, \ldots, a_0$, koristeći nama razumljivu decimalnu aritmetiku?

Iz relacije (3.2.1) vidimo da cjelobrojno dijeljenje broja x_c s 2 daje novi cijeli broj

$$x_c' = a_k \cdot 2^{k-1} + a_{k-1} \cdot 2^{k-2} + \cdots + a_1 \cdot 2^0$$

i ostatak a_0. Ako postupak ponovimo na broju x_c', dobivamo sljedeću znamenku, a_1. Ponavljanjem tog postupka (u našem slučaju k puta) dobivamo sve binarne znameke broja x_c u redosljedu od a_0 do a_k.

S obzirom da je x_c cijeli broj, on će i u binarnom sustavu imati konačni prikaz. Pritom će binarni prikaz imati $[\log_2(x_c)] + 1$ binarnih znamenaka (bitova). Oznaka $[z]$ za realni broj z označava najveći cijeli broj koji nije veći od z. Ako x_c ima p decimalnih znamenaka (decimala), tada vrijedi $10^{p-1} \leq x_c < 10^p$. Primjenom funkcije $\log_2$ i korištenjem njene monotonosti, zadnje nejednakosti daju

$$\log_2(10^{p-1}) \leq \log_2(x_c) < \log_2(10^p),$$

odnosno

$$(p-1)\log_2(10) \leq \log_2(x_c) < p\log_2(10).$$

Kako je $\log_2(10) \approx 3.3219$, binarna reprezentacija zahtijeva oko $3.3p$ znamenka, dakle oko 3.3 puta više nego decimalna reprezentacija.

Neka je sada x_r kao u relaciji (3.2.2). Tada množenjem broja x_r s dva i uzimanjem cijelog dijela rezultata dobivamo b_1. Ponavljajući isti postupak s ostatkom

$$x_r' = b_2 \cdot 2^{-1} + \cdots + b_{l-1} \cdot 2^{-(l-2)} + b_l \cdot 2^{-(l-1)},$$

dobivamo binarnu znamenku b_2. Nastavljanjem postupka (l puta) lako dobivamo sve binarne znamenke broja x_r, u redosljedu od b_1 do b_l. Zbog jednostavnosti postupka, odmah zaključujemo da postupak vrijedi i u slučaju kad x_r nema konačni binarni prikaz, ali tada imamo beskonačno koraka.

U sljedećem algoritmu za pretvaranje decimalne reprezentacije brojeva u binarnu, pretpostavljamo korištenje decimalne aritmetike.

Algoritam 3.2.1 *Neka je zadan racionalan broj $x = x_c + x_r$ pri čemu je x_c cijeli dio broja x, a x_r razlomljeni dio takav da je $0 \leq x_r < 1$. Algoritam računa binarnu reprezentaciju decimalnih brojeva x_c i x_r. Algoritam koristi operaciju* **div** *za cjelobrojno dijeljenje,* **mod** *za dobivanje ostatka kod cjelobrojnog dijeljenja i* **int** *za zaokruživanje realnog broja do cijelog broja u smjeru nule. Dakle,* **int** *je funkcija koja odgovara matematičkoj funkciji $x \mapsto [x]$ koju smo već spomenuli.*

$y := x_c$;
$k := -1$;
repeat
  $k := k + 1$;
  $a(k) := \mathbf{mod}(y, 2)$;
  $y := \mathbf{div}(y, 2)$;
until $y = 0$;

$z := x_r$;
$l := 0$;
repeat
  $l := l + 1$;
  $zz := 2 * z$;
  $b(k) := \mathbf{int}(zz)$;
  $z := zz - b(k)$;
until $z = 0$ **or** $l > lmax$

Pritom lmax označava najveći dopušteni broj binarnih znamenaka u binarnoj mantisi (sjetimo se da $1/10$ *ima beskonačno nula i jedinica).*

Primjer 3.2.1 *Korištenjem algoritma 3.2.1 pretvorit ćemo decimalni broj* 111.1 *u binarni. Kad radimo "ručno" zgodno je nacrtati tablicu, tako da kod pretvaranja broja* 111 *za svaki korak pišemo s desne strane rezultat cijelobrojnog dijeljenja, a ispod djeljenika ostatak pri dijeljenju. Točka na kraju označava kraj.*

111	55	27	13	6	3	1	0
1	1	1	1	0	1	1	.

što daje

$$111 = (110111)_2.$$

Slično, kod pretvaranja decimalnog dijela broja, 0.1 *u binarni broj, s desne strane pišemo rezultat množenja s dva, a s ispod crte rezultat zaokruživanja do cijelog broja u smjeru nule. Ako je rezultat dijeljenja veći ili jednak jedan, u sljedećem množenju, prije samog množenja automatski oduzimamo jedinicu. Binarna točka prethodi računanju i pišemo je ispod polaznog decimalnog broja*

0.1	0.2	0.4	0.8	1.6	1.2	0.4	0.8	1.6	1.2
.	0	0	0	1	1	0	0	1	1

.

Primijetite da se parovi znamenki 0011 *ponavljaju, pa dobivamo*

$$0.1 = (0.0001100110011001100\ldots)_2 = (0.0\dot{0}01\dot{1})_2,$$

odnosno

$$(111.1)_{10} = (1101111.0001100110011001100\ldots)_2 = (1101111.0\dot{0}01\dot{1})_2.$$

3.2.2. Reprezentacija brojeva u računalu

U programskim jezicima postoji nekoliko vrsta aritmetika koje koriste posve određene tipove podataka. Cjelobrojna aritmetika koristi **cjelobrojni tip** podataka koji čine konačni interval u skupu cijelih brojeva, realna aritmetika koristi takozvani **realni tip** podataka kojem pripada tek konačni podskup racionalnih brojeva. Nešto veći skup racionalnih brojeva čini **tip brojeva u dvostrukoj preciznosti** kojima su mantise više od dvostruko dulje od mantisa realnog tipa. I pripadna aritmetika je drukčija od realne aritmetike (jer je više nego dvostruko preciznija). Konačno, konačni podskup skupa kompleksnih brojeva reprezentiran je tipom kompleksnih brojeva (koji nije prisutan kod svih programskih jezika) nad kojim je definirana pripadna kompleksna aritmetika. Svaki od tih glavnih tipova obično ima svoje podtipove (ili ekstenzije) koje obično određuje broj bajtova. Jedan bajt (engl. byte) čini 8 bitova, koji su dostatni za reprezentaciju jednog znaka. Današnja računala imaju ćelije od 32 bita pa su tome prilagođeni građa procesora, implementacije aritmetika kao i cijeli operacioni sustav. Danas su dostupna i računala bazirana na 64 bitnim ćelijama.

Reprezentacija cijelih brojeva u računalu

Pozitivni cijeli brojevi reprezentiraju se u 32 bitnoj ćeliji kao desno pozicionirani binarni brojevi. Npr. broj 111 bit će smješten ovako

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1

Na taj način možemo smjestiti sve brojeve od 0 (koja je reprezentirana s 32 nule) do $2^{32}-1$ (koji je reprezentiran s 32 jedinice). Broj 2^{32} je prevelik. Njegova binarna reprezentacija zahtijeva jednu jedinicu i 32 nule. Neki programski jezici imaju tip podataka **cijeli broj bez predznaka** i spomenuta reprezentacija je odgovarajuća za njih.

Međutim, ako unutar 32 bita moramo spremati i pozitivne i negativne brojeve, prvo moramo vidjeti kako spremiti negativne cijele brojeve. Najočitija mogućanost je uzeti jedan bit za predznak, na primjer prvi, tako da je taj bit 0 ako je broj pozitivan i 1 ako je negativan. S obzirom da na raspolaganju imamo još 31 bit,

raspon tako reprezentiranih brojeva je od od $-2^{31}+1$ do $2^{31}-1$. Međutim, gotovo sva današnja računala koriste pametniji način reprezentacije negativnih brojeva koji se zove **drugi komplement**[1] i piše 2. komplement.

U sustavu koji koristi drugi komplement, nenegativni cijeli broj x, $0 \leq x \leq 2^{31}-1$ smješta se kao binarna reprezentacija tog broja, dok se $-x$, $1 \leq x \leq 2^{31}-1$ smješta kao binarna reprezentacija broja $2^{32}-x$. 1. komplement broja dobiva se jednostavnim komplementiranjem znamenki, pa je za 111 1. komplement

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

.

Dodamo li 1 toj reprezentaciji, dobili smo dvojni komplement

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	1

,

čime je reprezentiran broj -111.

Primjer 3.2.2 *Zbrojimo u binarnoj aritmetici brojeve* 111 *i* -111 *korištenjem* 32*-bitne reprezentacije i drugog komplementa. Imamo*

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1
+	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Ovdje je vodeća jedinica u rezultatu tzv. prekobrojni bit, jer je na 33*. mjestu, nema se gdje spremiti, pa se odbacuje. Preostaje*

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

što je reprezentacija broja nula.

Zadatak 3.2.1 *Koliko se različitih cijelih brojeva može reprezentirati ako se koristi sustav*

(*a*) *predznak i modul,* (*b*) 2*. komplement?*

Koristite potencije od 2*. Za koji od ovih sustava je reprezentacija nule jedinstvena? Promotrite slučajeve kad za reprezentaciju cijelih brojeva imate na raspolaganju*

(*i*) 32 *bita, tj.* 4 *bytea,* (*ii*) 16 *bita, tj.* 2 *bytea,*
(*iii*) 8 *bita, tj.* 1 *byte,* (*iv*) 64 *bita, tj.* 16 *bytea.*

[1] Još se koristi naziv **dvojni komplement**. Ime dolazi odatle što se je između 1960. i 1980. na nekim superračunalima koristio tzv. 1. komplement kod kojeg se negativni broj $-x$ smjestio kao binarna reprezentacija od $2^{32}-x-1$.

Zadatak 3.2.2 *Pokažite da je u sustavu reprezentacije cijelih brojeva pomoću 2. komplementa (npr. u 32 bitnom formatu), najljeviji bit* 1 *ako i samo ako je* x *negativan.*

Zadatak 3.2.3 *Da bi u sustavu reprezentacije cijelih brojeva pomoću 2. komplementa s 32 bita, od već uskladištenog broja* x *dobili* $-x$ *treba poduzeti dva koraka. Prvi je promijeniti svaku* 0 *u* 1 *i svaki* 1 *u* 0*. Koji je drugi korak?*

Sva računala imaju ugrađene ("hardverske") instrukcije za zbrajanje cijelih brojeva. Ako se zbroje dva pozitivna cijela broja ili dva negativna cijela broja, (stvarni) rezultat može biti broj koji je veći od maksimalnog prikazivog broja. Zanimljivo je da tada računalo **neće javiti nikakvu grešku**, jer je matematički gledano, aritmetika cijelih brojeva u računalu modularna aritmetika u prstenu ostataka modulo 2^{32}, samo je sistem ostataka simetričan oko 0, tj.

$$-2^{31}, \ldots, -1, 0, 1, \ldots, 2^{31} - 1.$$

Brojeve izvan tog raspona uopće ne možemo spremiti u računalo.

Primjer 3.2.3 *U cjelobrojnoj aritmetici izračunajmo* $n!$ *za sve* $n \leq 34$.

Napišimo jednostavni program koji računa $n!$*, s tim da su sve varijable u programu prvo cijeli brojevi, a zatim, kontrole radi stavimo varijablu* $fact$ *da pripada relnim brojevima (u barem dvostrukoj točnosti).*

$fact := 1;$
for $i := 1$ **to** 34 **do**;
  begin
  $fact := fact * i;$
  (**ispis** $i! = fact;$)
  end;

Rezultati programa bit će (lijevo rezultati u cjelobrojnoj aritmetici, u zagradi u realnoj):

$1! =$	1	$(1.00000000000000e{+}00)$
$2! =$	2	$(2.00000000000000e{+}00)$
$3! =$	6	$(6.00000000000000e{+}00)$
$4! =$	24	$(2.40000000000000e{+}01)$
$5! =$	120	$(1.20000000000000e{+}02)$
$6! =$	720	$(7.20000000000000e{+}02)$
$7! =$	5040	$(5.04000000000000e{+}03)$
$8! =$	40320	$(4.03200000000000e{+}04)$
$9! =$	362880	$(3.62880000000000e{+}05)$

$$
\begin{aligned}
10! &= 3628800 && (3.62880000000000e{+}06)\\
11! &= 39916800 && (3.99168000000000e{+}07)\\
12! &= 479001600 && (4.79001600000000e{+}08)\\
13! &= 1932053504 && (6.22702080000000e{+}09)\\
14! &= 1278945280 && (8.71782912000000e{+}10)\\
15! &= 2004310016 && (1.30767436800000e{+}12)\\
16! &= 2004189184 && (2.09227898880000e{+}13)\\
17! &= -288522240 && (3.55687428096000e{+}14)\\
18! &= -898433024 && (6.40237370572800e{+}15)\\
19! &= 109641728 && (1.21645100408832e{+}17)\\
20! &= -2102132736 && (2.43290200817664e{+}18)\\
21! &= -1195114496 && (5.10909421717094e{+}19)\\
22! &= -522715136 && (1.12400072777761e{+}21)\\
23! &= 862453760 && (2.58520167388850e{+}22)\\
24! &= -775946240 && (6.20448401733239e{+}23)\\
25! &= 2076180480 && (1.55112100433310e{+}25)\\
26! &= -1853882368 && (4.03291461126606e{+}26)\\
27! &= 1484783616 && (1.08888694504184e{+}28)\\
28! &= -1375731712 && (3.04888344611714e{+}29)\\
29! &= -1241513984 && (8.84176199373970e{+}30)\\
30! &= 1409286144 && (2.65252859812191e{+}32)\\
31! &= 738197504 && (8.22283865417792e{+}33)\\
32! &= -2147483648 && (2.63130836933694e{+}35)\\
33! &= -2147483648 && (8.68331761881189e{+}36)\\
34! &= 0 && (2.95232799039604e{+}38).
\end{aligned}
$$

Primijetite da razlike počinju kod $13!$*, kad u cjelobrojnoj aritmetici dobijemo da je* $13! < 12!$*. Nadalje, primijetite da* $n!$ *u cjelobrojnoj aritmetici može biti negativan broj ili* 0*.*

Zadatak 3.2.4 *Pokažite, da bez obzira kolika je duljina ćelija u koju spremamo cijeli broj, postoji* k *takav da je za svaki* $n > k$ *uvijek* $n! = 0$*.*

Promotrimo operaciju $x + (-y)$, gdje su $0 \leq x \leq 2^{31} - 1$ i $0 \leq y \leq 2^{31}$. S obzirom da je broj $-y$ u reprezentaciji 2. komplementa ima binarni prikaz od $2^{32} - y$, zbroj ćemo zapisati kao

$$2^{32} + x - y = 2^{32} - (y - x).$$

Ako je $x \geq y$ najljeviji bit u reprezentaciji od $2^{32} + (x - y)$ bit će jedinica (na poziciji potencije 2^{32}) i ona će se odbaciti, pa će ostati točan rezultat $x - y$. Ako je $x < y$, rezultat $2^{32} - (y - x)$ stane unutar 32-bitne reprezentacije i reprezentira broj $-(y - x)$. Ovo razmatranje pokazuje jedno važno svojstvo sustava reprezentacije s 2. komplementom: nije potrebna nikakva posebna (hardverska) operacija za cjelobrojno oduzimanje. Jer, kad se jednom broj $-y$ reprezentira, dovoljno je koristiti samo operaciju hardverskog zbrajanja.

Zadatak 3.2.5 *Ako cijele brojeve prikazujemo* 8*-bitnom reprezentacijom, pokažite detalje kako nastaju cjelobrojne sume:* $50 + (-200)$*,* $200 + (-50)$ *i* $200 - 200$*.*

Pored zbrajanja, postoje još dvije standardne (hardverske) operacije za cjelobrojne operande: cjelobrojno množenje i cjelobrojno dijeljenje.

Reprezentacija realnih brojeva u računalu

U računalu se realni brojevi reprezentiraju pomoću binarne reprezentacije u "znanstvenoj notaciji",

$$x = \pm m \times 2^e, \quad \text{gdje je} \quad 1 \leq m < 2. \tag{3.2.3}$$

Stoga je

$$m = (b_0.b_1b_2b_3 \ldots b_{p-1})_2 \quad \text{pri čemu je} \quad b_0 = 1. \tag{3.2.4}$$

Na primjer, broj $111.5 = (1101111.1)_2$ se može napisati kao $(1.1011111)_2 \times 2^6$. Vidimo da je u znanstvenom prikazu binarna točka pomaknuta za 6 mjesta ulijevo, a pritom se eksponent povećao za 6. Kako se zahtijeva da je $b_0 = 1$, možemo pisati

$$m = (1.b_1b_2b_3 \ldots b_{p-1})_2.$$

U tom prikazu, binarne znamenke desno od binarne točke čine razlomljeni dio mantise, a relacije (3.2.3) i (3.2.4) predstavljaju **normalizirani** oblik ili reprezentaciju broja x. Sam proces dobivanja tog oblika se zove normalizacija.

Da bismo spremili normalizirane brojeve u računalo, podijelimo memorijsku riječ (sadržaj jedne ćelije) u tri dijela koja zovemo polja. Kod 32-bitnih računala, riječ uobičajeno ima 32 bita pa se obično dijeli na sljedeći način (tip `single`): 1 bit za predznak, 8 bita za eksponent e i 23 bita za mantisu. Bit za predznak je 0 (1) ako je broj pozitivan (negativan). Polje za eksponent ima osam bitova pa može reprezentirati ekponent e koji je između granica -128 i 127 (npr. pomoću reprezentacije s drugim komplementom). Preostala 23 bita za smještaj mantise koriste se za smještaj razlomljenog dijela mantise, jer je uvijek $b_0 = 1$, pa ga ne treba spremati. Zato se b_0 obično naziva skriveni bit. Realni broj x nazivamo **egzaktno reprezentabilnim** u računalu ili **brojem s pomičnom točkom** ako se na opisani način može bez greške smjestiti u računalo. Ako broj nije egzaktno reprezentabilan u računalu, on se mora prije smještanja u računalu zaokružiti.

Primjer 3.2.4 *U primjeru 3.2.1 pokazano je da je*

$$(111.1)_{10} = (1101111.0\dot{0}01\dot{1})_2,$$

pa $x = 111.1$ *nije egzaktno reprezentabilan u računalu. Normalizirajmo taj broj*

$$x = (1.1011110\dot{0}01\dot{1})_2 \cdot 2^6.$$

Očito je da se u računalo može pohraniti samo konačan broj znamenki iza binarne točke. U slučaju mantise duljine 23 *bita, broj se zaokružuje na* 23 *znamenke iza decimalne točke i sprema u prostor za mantisu. U polje eksponenta sprema se broj* 6 *u binarnom zapisu* $(110)_2$*, što ćemo kraće označavati* eksp(6)*. Dakle, sprema se*

predznak: | 0 |

eksponent: | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | = eksp(6)

mantisa: | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |.

S druge strane broj $y = -2^{22} = -4194304$*, će imati egzaktnu reprezentaciju*

predznak: | 1 |

eksponent: eksp(22)

mantisa: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |.

Zadatak 3.2.6 *Koji su najveći i najmanji pozitivni brojevi koji se mogu reprezentirati kao realni brojevi u računalu (brojevi zapisani s pomičnom točkom, gdje je* 1*. bit za predznak,* 8 *bitova za eksponent i* 23 *bita za mantisu)? Ne zaboravite na skriveni bit i na ograničen interval* $[-128, 127]$ *za eksponent. Koji je najmanji cijeli pozitivni broj koji nije egzaktno reprezentabilan?*

Zadatak 3.2.7 *Pretpostavimo da umjesto oblika (3.2.4) u relaciji (3.2.3) zahtijevamo*

$$S = (0.b_1b_2b_3 \ldots b_{p-1})_2,$$

pri čemu je $b_1 = 1$*. Pretpostavimo da se u polje za mantisu smještaju binarne znamenke* $b_2b_3 \ldots b_{24}$ *i da je, kao i prije, raspon za eksponent* $-128 \leq e \leq 127$*. Koji je najveći i najmanji pozitivni broj u tom formatu? Koji je najmanji pozitivni broj koji nije egzaktno reprezentabilan?*

Primijetite da u realnom formatu postoje brojevi koji su "preveliki" da bi se mogli smjestiti u odgovarajući format. Na primjer, množenjem dva velika broja rezultat može biti takav da se ne može spremiti. U tom slučaju dolazi do tzv. **preljeva** ili **prekoračenja** (engl. **overflow**). Što računalo u takvom slučaju načini ovisi o **prevodiocu** (engl. **compiler**) za odgovarajući programski jezik. Najčešće se prekida računanje uz adekvatnu poruku.

Strojni epsilon, ulp i preciznost

Preciznost p možemo u danom brojevnom sustavu definirati brojem bitova u mantisi pri čemu se računa i skriveni bit. U opisanom brojevnom sustavu je $p = 24$. U brojevnom sustavu s preciznošću p, normalizirani broj s pomičnom točkom ima oblik

$$x = \pm(1.b_1b_2 \dots b_{p-2}b_{p-1}) \times 2^e. \tag{3.2.5}$$

Najmanji takav x koji je veći od 1 je

$$(1.00 \dots 01)_2 = 1 + 2^{-(p-1)}.$$

Razmak između ta dva broja se zove **strojni epsilon** (engl. **machine epsilon**) i zapisuje

$$\epsilon_M = 2^{-(p-1)}.$$

Općenitije, za x kao u relaciji (3.2.5) definira se

$$\text{ulp}(x) = (0.00 \dots 01)_2 \times 2^e = 2^{-(p-1)} \times 2^e = \epsilon_M \times 2^e. \tag{3.2.6}$$

Ulp je kratica engleskih riječi **unit in the last place**. Ako je $x > 0$ ($x < 0$), $\text{ulp}(x)$ je razmak između x i sljedećeg većeg (manjeg) reprezentabilnog broja.

Zadatak 3.2.8 *Neka je $p = 24$, tako da je $\epsilon_M = 2^{-23}$. Odredite* $\text{ulp}(x)$ *redom za* $x = 0.5, 0.125, 3, 8, 10, 100, 125$.

Nadalje, postavlja se pitanje kako spremiti nulu u računalo? To se postiže korištenjem specijalnog niza bitova u polju za eksponent. Tomu ćemo posvetiti više pažnje kod opisa IEEE standarda.

BSP ili brojevni sustav za prikazivanje

Da bismo shvatili koje točke na brojevnom pravcu odgovaraju reprezentabilnim brojevima, definirat ćemo pojednostavljeni **brojevni sustav za prikazivanje** ili kraće BSP, koji se sastoji od brojeva oblika

$$\pm(b_0.b_1b_2)_2 \times 2^e, \quad b_0, b_1, b_2 \in \{0, 1\}, \quad e \in \{-1, 0, 1\}.$$

Preciznost je $p = 3$, najveći prikazivi broj je

$$(1.11)_2 \times 2^1 = (3.5)_{10},$$

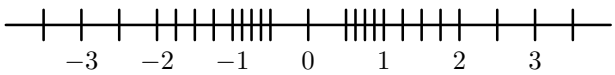
a najmanji

$$(1.00)_2 \times 2^{-1} = (0.5)_{10}.$$

Kako je desni susjed od 1 u BSP-u 1.25, strojni epsilon je $\epsilon_M = 0.25$. Ako promatramo sve brojeve u BSP-u za koje je $e = 0$, vidjet ćemo da su to brojevi 1, 1.25,

1.5 i 1.75. Između svih njih je isti razmak $\text{ulp}(x) = \epsilon_M$. Brojevi u BSP-u za koje je $e = 1$ dobiju se množenjem s dva onih brojeva za koje je $e = 0$, pa su to brojevi 2, 2.5, 3, 3.5 i za njih je $\text{ulp}(x) = 2\epsilon_M$. Slično, brojevi iz BSP-a za koje je $e = -1$ su 0.5, 0.625, 0.75 i 0.825 i za njih je $\text{ulp}(x) = \epsilon_M/2$. Vidimo da je razmak između broja $x \in$ BSP i njegovog desnog susjeda u BSP-u jednak

$$\text{ulp}(x) = \epsilon_M \times 2^e.$$



Slika 3.2.1 BSP brojevni sustav

Uočimo da je razmak između nule i ± 0.5 mnogo veći od razmaka brojeva između ± 0.5 i ± 1. Uskoro ćemo vidjeti da se razmak između nule i ± 0.5 može ispuniti tzv. subnormalnim brojevima.

3.3. IEEE Aritmetika

U kasnim 70-tim i ranim 80-tim godinama, došlo je do izuzetne suradnje eksperata iz industrije i sa sveučilišta. Predvođeni W. Kahanom sa kalifornijskog sveučilišta u Berkeleyu, stručnjaci iz računarstva i dizajneri mikročipova uspjeli su se dogovoriti oko standarda smještanja brojeva u računalo i standarda aritmetike. To je bilo vrijeme naglog razvoja osobnih računala s Intelom i Motorolom kao glavnim proizvođačima procesora. IEEE standard za binarnu aritmetiku je publiciran 1985. i spominje se kao IEEE 754. Razvoj kalkulatora i malih, tzv. handheld računala koja koriste decimalnu aritmetiku utjecao je na razvoj standarda IEEE 854 koji vrijedi za svaku bazu i konzistentan je s prethodnim standardom za binarnu aritmetiku. Kad se spominje "IEEE aritmetika" podrazumijeva se da ta aritmetika udovoljava spomenute standarde.

Bitni zahtjevi IEEE standarda su:

(i) konzistentna reprezentacija brojeva s pomičnom točkom na svim računalima koja prihvaćaju standard,

(ii) korektno zaokružene računske operacije u svim načinima rada i

(iii) konzistentno tretiranje izvanrednih situacija kao što je npr. dijeljenje s nulom.

U tom standardu je vodeća jedinica normalizirarog broja skrivena, pa je zato potrebna specijalna reprezentacija broja nula. Uvodi se i specijalna reprezentacija brojeva $\pm\infty$, brojeva ± 0 (to je za razliku od matematički gledano različitih $+\infty$ i $-\infty$, isti

broj), te specijalnih izmišljenih brojeva "NaN" ("Not a number", koji je oznaka za, npr. kvocijent 0/0). IEEE format specificira dva osnovna formata: jednostruki i dvostruki.

3.3.1. Jednostruki format

Broj u jednostrukom formatu sprema se u ćeliju koja ima tri polja: polje za predznak (1 bit, $\pm$), polje za eksponent (8 bitova, bitovi $a_1a_2\ldots a_8$) i polje za mantisu (23 bita, bitovi $b_1b_2\ldots b_{23}$) i prikazat ćemo ih na sljedeći način:

$\pm$	$a_1a_2\ldots a_8$	$b_1b_2\ldots b_{23}$

.

Ovaj format opisan je tablicom 3.3.1. Pritom znak $\pm$ u nizu bitova znači 0 ako je

ako je niz $a_1a_2\ldots a_8$	onda je numerička vrijednost
$(00000000)_2=(0)_{10}$	$\pm(0.b_1b_2\ldots b_{23})_2\times 2^{-126}$
$(00000001)_2=(1)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2\times 2^{-126}$
$(00000010)_2=(2)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2\times 2^{-125}$
$(00000011)_2=(3)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2\times 2^{-124}$
$\vdots$	$\vdots$
$(01111111)_2=(127)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2\times 2^{0}$
$(10000000)_2=(128)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2\times 2^{1}$
$\vdots$	$\vdots$
$(11111100)_2=(252)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2\times 2^{125}$
$(11111101)_2=(253)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2\times 2^{126}$
$(11111110)_2=(254)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2\times 2^{127}$
$(11111111)_2=(255)_{10}$	$\pm\infty$ ako su svi $b_i=0$, inače NaN

Tablica 3.3.1 IEEE jednostruki format

predznak broja + i 1 ako je −. Iz prvog reda vidimo da je nula reprezentirana s

$\pm$	00000000	00000000000000000000000

.

Svi retci u tablici 2, osim prvog i zadnjeg, reprezentiraju normalizirane brojeve. Vidimo da eksponent nije prikazan, ni kao drugi komplement, niti u obliku predznak plus modul, već kao $127+e$. Broj $127+e$ zvat ćemo **karakteristika**. Npr. broj 1 reprezentira se kao

0	01111111	00000000000000000000000

,

dok se broj

$$0.1 = (1.1\dot{0}01\dot{1})_2 \cdot 2^{-4},$$

ako se "višak" znamenki odbacuje, reprezentira kao

0	01111011	10011001100110011001100

a ako se "višak" znamenki korektno zaokružuje (tako to rade sva moderna računala), jer su prve dvije odbačene znamenke 11,

0	01111011	10011001100110011001101

Najmanji i najveći eksponent od 2 u tom formatu su $e_{\min} = -126$ i $e_{\max} = 127$. Najmanji normalizirani broj

$$N_{\min} = (1.000\ldots0)_2 \times 2^{-126} = 2^{-126} \approx 1.1755 \times 10^{-38}$$

reprezentira se s

0	00000001	00000000000000000000000

dok se najveći broj

$$N_{\max} = (1.111\ldots1)_2 \times 2^{127} = (2 - 2^{-23}) \times 2^{127} \approx 2^{128} \approx 3.4028 \times 10^{38}$$

reprezentira s

0	11111110	11111111111111111111111

Odmah vidimo da je

$$\frac{1}{N_{\min}} < N_{\max},$$

pa inverz najmanjeg normaliziranog broja ne dovodi do prekoračenja gornje granice (overflow).

Iz zadnjeg retka tablice vidimo da niz bitova 1111111 u eksponencijalnom dijelu reprezentacije vodi ili do $\pm\infty$ (samo ako su svi b_i jednaki 0) ili do NaN.

Konačno, pogledajmo još jednom prvi redak tablice 3.3.1. U njemu je eksponencijalni dio uvijek 0000000, dok razlomljeni dio može imati bilo koji izbor jedinica i nula. Ako su svi b_i nula, imat ćemo $+0$, ili -0 ovisno o prvom bitu. Ako nisu svi b_i nula, dobit ćemo tzv. **subnormalne** ili **denormalizirane** brojeve koji ekvidistantno ispunjavaju razmak između $-N_{\min}$ i $+N_{\min}$. Najmanji pozitivni subnormalni broj reprezentiran je kao

0	00000001	00000000000000000000001

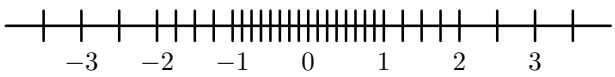
a vrijednost mu je

$$2^{-149} = (0.000\ldots01) \times 2^{-126}.$$

To je najmanji broj koji se može reprezentirati pomoću jednostrukog formata. Najveći subnormalni broj je

$$(0.111\ldots11)_2 \times 2^{-126}$$

i on je za 2^{-149} manji od $N_{\min}$. Sve reprezentabilne brojeve u brojevnom sustavu za prikazivanje, uključujući i subnormalne brojeve sada možemo prikazati slikom 3.3.1.



Slika 3.3.1 BSP brojevni sustav sa subnormalnim brojevima

Subnormalni brojevi su manje točni od normaliziranih. Npr.

$$(1/10) \times 2^{-136} = (0.\dot{1}10\dot{0})_2 \times 2^{-139}$$

ima reprezentaciju

0	00000000	00000000000001100110011

.

Zadatak 3.3.1 *U jednostrukom IEEE formatu odredite reprezentacije sljedećih brojeva s pomičnom točkom:* $3, 2000, 11.5, 11.5 \times 2^{100}$ *i* 0.1×2^{-142}.

Zadatak 3.3.2 *Napišite algoritam koji određuje koji je od dva broja x, y zapisanih u jednostrukom IEEE formatu veći od drugoga. Treba uspoređivati bitove slijeva nadesno i donijeti zaključak na prvom mjestu na kojem se razlikuju. Činjenica da se to može učiniti tako jednostavno utjecala je na to da se eksponent smjesti u polje pomoću karakteristike.*

3.3.2. Dvostruki format

Kod zahtjevnijih računanja, jednostruki format nije adekvatan, kako zbog neizbježnih grešaka zaokruživanja, tako zbog premalog raspona brojeva u tom formatu. Zato IEEE standard specificira i drugi tzv. **dvostruki format** koji koristi 64-bitnu riječ,

$\pm$	$a_1a_2\ldots a_{11}$	$b_1b_2\ldots b_{52}$

.

Detalji su vidljivi u tablici 3.3.2. Ideje su iste kao kod jednostrukog formata, samo su polja za mantisu i eksponent veća: 11 bitova za eksponent i 52 bita za razlomljeni dio mantise, pa su zato $e_{\min} = -1022$ i $e_{\max} = 1023$, te

$$N_{\min} = 2^{-1022} =\approx 2.225 \times 10^{-308}$$

ako je niz $a_1a_2\ldots a_{11}$	onda je numerička vrijednost
$(00000000000)_2 = (0)_{10}$	$\pm(0.b_1b_2\ldots b_{52})_2 \times 2^{-1022}$
$(00000000001)_2 = (1)_{10}$	$\pm(1.b_1b_2\ldots b_{52})_2 \times 2^{-1022}$
$(00000000010)_2 = (2)_{10}$	$\pm(1.b_1b_2\ldots b_{52})_2 \times 2^{-1021}$
$(00000000011)_2 = (3)_{10}$	$\pm(1.b_1b_2\ldots b_{52})_2 \times 2^{-1020}$
$\vdots$	$\vdots$
$(01111111111)_2 = (1023)_{10}$	$\pm(1.b_1b_2\ldots b_{52})_2 \times 2^{0}$
$(10000000000)_2 = (1024)_{10}$	$\pm(1.b_1b_2\ldots b_{52})_2 \times 2^{1}$
$\vdots$	$\vdots$
$(11111111100)_2 = (2044)_{10}$	$\pm(1.b_1b_2\ldots b_{52})_2 \times 2^{1021}$
$(11111111101)_2 = (2045)_{10}$	$\pm(1.b_1b_2\ldots b_{52})_2 \times 2^{1022}$
$(11111111110)_2 = (2046)_{10}$	$\pm(1.b_1b_2\ldots b_{52})_2 \times 2^{1023}$
$(11111111111)_2 = (2047)_{10}$	$\pm\infty$ ako su svi $b_i = 0$, inače NaN

Tablica 3.3.2 IEEE dvostruki format

i

$$N_{\max} = (2 - 2^{-52}) \times 2^{1023} \approx 1.797693 \times 10^{308}.$$

IEEE standard zahtijeva da računalo omogućava jednostruki format. Dvostruki format se zahtijeva tek kao mogućnost, iako ga gotovo sva računala koja podržavaju standard imaju. Podrška na zahtjeve standarda može biti programska ("softverska") ili elektronička ("hardverska"), iako zbog brzine rada računala, proizvođači računala najčešće daju hardversku podršku standardu.

U sljedećoj tablici prikazani su karakteristični podaci za jednostruki i dvostruki format.

format	$e_{\min}$	$e_{\max}$	$N_{\min}$	$N_{\max}$
jednostruki	-126	127	$2^{-126} \approx 1.2 \times 10^{-38}$	$\approx 2^{128} \approx 3.4 \times 10^{38}$
dvostruki	-1022	1023	$2^{-1022} \approx 2.2 \times 10^{-308}$	$\approx 2^{1024} \approx 1.8 \times 10^{308}$

Tablica 3.3.3 Raspon brojeva s pomičnom točkom u IEEE formatima

Standard također snažno preporuča i podršku za tzv. **proširени** (engl. extended) format koji bi trebao imati barem 15 bitova za eksponent i barem 63 bita za

mantisu. Intelovi mikroprocesori implementiraju aritmetiku s proširenim formatom u hardware-u, korištenjem 80-bitnih regisatra, od čega se 15 bitova koristi za eksponent, a 64 bita za mantisu, pri čemu, za razliku od jednostrukog i dvostrukog formata, vodeći bit (jedinica) nije skriven. Drugi strojevi (npr. Sun i Sparc) implementiraju prošireni format sa 128 bitova, ali softverski, pa je aritmetika ovdje sporija. Kod Intelovog proširenog formata, prvi desni susjed od 1 bio bi $1 + 2^{-64}$, ali kako se on ne može reprezentirati jer nema skrivenog bita, prvi desni susjed je $1 + 2^{-63}$. S obzirom da je $\log_{10}(2^{23}) \approx 6.9236$, $\log_{10}(2^{24}) \approx 7.2247$, preciznost $p = 24$ odgovara približno 7 značajnih decimalnih znamenki. Slično, preciznost $p = 53$ odgovara približno 16, a $p = 64$ odgovara približno 19 značajnih decimalnih znamenki.

format	preciznost (p)	strojni epsilon (ϵ_M)
jednostruki	24	$2^{-23} \approx 1.2 \times 10^{-7}$
dvostruki	53	$2^{-52} \approx 2.2 \times 10^{-16}$
prošireni (Intel)	64	$2^{-63} \approx 1.1 \times 10^{-19}$

Tablica 3.3.4 Preciznost kod IEEE formata

Moderna računala adresiraju memoriju po byte-ovima, pa 32-bitnu riječ adresiraju sa 4 byte-a, nazovimo ih $B_1, \ldots, B_4$, pri čemu je $B_4 = B_1 + 3$. U jednostrukom formatu, najvažniji je byte u kojem su smješteni $\sigma, a_1, \ldots, a_7$ (σ je predznak). Ako je taj byte adresiran s B_1 (B_4), tada se takav adresni sustav zove **Big** (**Small**) **Endian**. Npr. IBM i Sun koriste BIG, dok Intel koristi Small Endian (neki procesori kao DEC Alpha mogu raditi s oba sustava). To znači da kod transfera podataka s jednog računala na drugo traba biti oprezan.

3.3.3. BSPT i zaokruživanje u BSPT

Brojevni sustav s pomičnom točkom kraće ćemo označavati s BSPT, a skup brojeva s pomičnom točkom s BPT. Te oznake se odnose na razmatrani sustav/skup brojeva za koji ćemo smatrati, ako drugačije ne naznačimo, da zadovoljavaju IEEE standard.

Za realni broj x reći ćemo da leži u **intervalu normaliziranih brojeva** sustava s pomičnom točkom, ako vrijedi

$$N_{\min} \leq |x| \leq N_{\max}.$$

Dakle, brojevi ± 0, $\pm\infty$ i subnormalni brojevi nisu u tom intervalu, iako pripadaju spomenutom brojevnom sustavu.

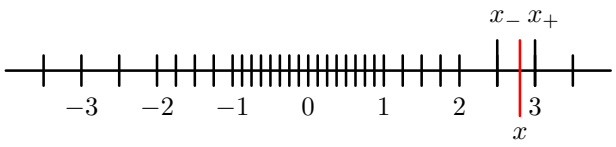
Neka je x realni broj koji nije reprezentabilan u sustavu brojeva s pomičnom točkom. Tada je barem jedna od sljedećih tvrdnji istinita:

- x leži izvan intervala normaliziranih brojeva (npr. $x = 2^{129}$ u jednostrukom formatu),
- binarna reprezentacija od x zahtijeva više od p bitova za egzaktnu reprezentaciju (npr. $x = 0.1$).

U oba slučaja porebno je X zamijeniti brojem iz sustava brojeva s pomičnom točkom. Neka je

$$x_- \leq x \leq x_+,$$

pri čemu su brojevi s pomičnom točkom x_- i x_+ najbliži broju x (vidi sliku).



Slika 3.3.2 Zaokruživanje u BSPT brojevnom sustavu

Ako pretpostavimo da je

$$x = (1.b_1b_2 \ldots b_{p-1}b_pb_{p+1} \ldots)_2 \times 2^e, \tag{3.3.1}$$

onda su

$$x_- = (1.b_1b_2 \ldots b_{p-1})_2 \times 2^e$$
$$x_+ = [(1.b_1b_2 \ldots b_{p-1})_2 + (0.00 \ldots 01)_2] \times 2^e,$$

i razmak između x_- i x_+ je $2^{p-1} \times 2^e = \text{ulp}(x_-)$.

Ako je $x > N_{\max}$, tada je $x_- = N_{\max}$ i $x_+ = \infty$. Ako je $0 < x < N_{\min}$, tada je x_- ili nula ili subnormalni broj, a x_+ je subnormalni ili $N_{\min}$. Ako je x negativan, tada je situacija analogna (zrcalna slika u odnosu na ishodište). IEEE standard definira **korektno zaokruženu vrijednost** od x, koja se označava s round(x) na sljedeći način. Ako je x broj s pomičnom točkom, tada je round(x) = x. Ako nije, onda vrijednost od round(x) ovisi o načinu (**modu**) zaokruživanja koji je aktivan. Postoje četiri načina:

- zaokruživanje prema dolje (još se kaže prema $-\infty$): round(x) = x_-,
- zaokruživanje prema gore (još se kaže prema ∞): round(x) = x_+,
- zaokruživanje prema nuli:

$$\text{round}(x) = \begin{cases} x_- & \text{ako je } x > 0, \\ x_+ & \text{ako je } x < 0, \end{cases}$$

- zaokruživanje prema najbližem:

$$\text{round}(x) = \begin{cases} x_- & \text{ako je } |x - x_-| < |x - x_+|, \\ x_+ & \text{ako je } |x - x_-| > |x - x_+|. \end{cases}$$

Ako je $|x-x_-| = |x-x_+|$ uzima se x_- ili x_+, već prema tome je li u x_- ili u x_+ najmanje značajni bit b_{p-1} nula. Ako je $x > N_{\max}$ uzima se $\text{round}(x) = +\infty$, a ako je $x < N_{\min}$ uzima se $\text{round}(x) = -\infty$.

U praksi se gotovo uvijek koristi zadnji način zaokruživanja, prema najbližem susjedu u sustavu brojeva s pomičnom točkom i njega ćemo još malo pojasniti. Neka je x opet kao u (3.3.1). Ako je prvi bit koji se ne može reprezentirati, b_p jednak 0, tada je $\text{round}(x) = x_-$. Ako je $b_p = 1$ i još barem jedan od sljedećih bitova nije nula, tada je $\text{round}(x) = x_+$. Ako je $b_p = 1$ i svi preostali bitovi od x su 0, tada se uzima x_-, ili x_+ ovisno o tome koji od njih ima najmanje značajan bit 0. Uočimo da se x_- i x_+ razlikuju samo u zadnjem bitu, pa će jedan sigurno biti izabran. IEEE standard zahtijeva da se kao prvi (uobičajen, **default**) način uzima zaokruživanje prema najbližem, pa će se nadalje, ako se ne spomene drugačije, $\text{round}(x)$ koristiti u tom smislu. Ako je $x > N_{\max}$ i način zaokruživanja je prema najbližem, tada je $\text{round}(x) = \infty$ iako je x naravno bliže $N_{\max}$ nego ∞.

Zadatak 3.3.3 *Napišite u jednostrukom IEEE formatu reprezentacije zaokruženih vrijednosti (u načinu do najbližeg) brojeva:* $1/10, 2 \cdot 2^{24}$ *i* 2^{131}.

Zadatak 3.3.4 *Konstruirajte x za koji su x_- i x_+ jednako udaljeni od x i nađite pripadnu reprezentaciju broja x.*

Zadatak 3.3.5 *Neka x, $0 < x < N_{\min}$ nije subnormalni broj. Tada je*

$$x = (0.b_1 b_2 \dots b_{p-1} b_p b_{p+1} \dots)_2 \times 2^{e_{\min}},$$

pri čemu je bar jedan od brojeva $b_p b_{p+1} \dots$ jednak 1. Što je x_-? Uzmi kao primjere brojeve 2^{-130} i 2^{-150}. (Pretpostavka je jednostruki format i $e_{\min} = -126$).

Ako broj x,

$$x = (1.b_1 \dots b_{p-1} b_p b_{p+1} \dots)_2 \times 2^e$$

nije reprezentabilan jasno je da je $x_- \le x \le x_+$, bez obzira na način zaokruživanja. Stoga je

$$|\,\text{round}(x) - x| < 2^{-(p-1)} \times 2^e,$$

pa kažemo da je apsolutna greška kod zaokruživanja manja od jednog ulp-a. Pritom se misli na $\text{ulp}(x_-)$ ako je $x > 0$ i $\text{ulp}(x_+)$ ako je $x < 0$. Kad je prisutno zaokruživanje do najbližeg, onda vrijedi jača ocjena

$$|\,\text{round}(x) - x| \le 2^{-p} \times 2^e,$$

pa kažemo da je apsolutna greška zaokruživanja pola ulp-a.

Zadatak 3.3.6 *Odredite apsolutnu grešku prikaza* $1/10$ *u jednostrukom IEEE formatu. Koristite svaki od četiri načina zaokruživanja.*

Zadatak 3.3.7 *Vrijede li gornje ocjene za apsolutnu grešku zaokruživanja ako je* $|x| < N_{\min}$*? Objasnite.*

Zadatak 3.3.8 *Kolika je apsolutna greška za svaki od načina zaokruživanja ako je* $|x| < N_{\min}$*. Pazite na definiciju* $\operatorname{round}(x)$*.*

S obzirom da apsolutna greška kod zaokruživanja raste kad $|x|$ raste, promotrimo relativnu grešku od $\operatorname{round}(x)$ kao aproksimaciju od x. Iz relacije (3.3.1) znamo da je $|x| \geq 2^e$. Stoga za sve načine zaokruživanja vrijedi

$$\frac{|\operatorname{round}(x) - x|}{|x|} < \frac{2^{-(p-1)} \times 2^e}{2^e} = 2^{-(p-1)} \equiv u.$$

Kod zaokruživanja do najbližeg vrijedi malo bolja ocjena

$$\frac{|\operatorname{round}(x) - x|}{|x|} \leq \frac{2^{-p} \times 2^e}{2^e} = 2^{-p} \equiv u.$$

Korištenjem oznaka iz relacija (3.1.2) i (3.1.1) možemo za realne brojeve koji leže u normaliziranom intervalu napisati

$$\operatorname{round}(x) = x(1+\delta), \quad |\delta| \leq u, \tag{3.3.2}$$

gdje je $u = 2^{-p}$ ako se koristi zaokruživanje do najbližeg i $u = 2^{-p+1}$ ako se koriste ostali načini zaokruživanja. Pritom je p preciznost računala. Uočimo da $\delta = \delta(x)$ ovisi o x i načinu zaokruživanja. S obzirom da računala u normalnom radu koriste zaokruživanje do najbližeg, ako neće biti drugačije naznačeno, $u = 2^{-p}$, pri čemu je $p = 24$ za jednostruki, $p = 53$ za dvostruki i $p = 64$ za prošireni format.

Zadatak 3.3.9 *Nađite* x *u normaliziranom intervalu BPT za koji je* $\delta = 2^{-p}$ *ako se koristi uobičajeni način zaokruživanja do najbližeg.*

Zadatak 3.3.10 *Vrijedi li ocjena (3.3.2) za* $|x| < N_{\min}$*?*

S obzirom da je $-\log_2(\delta(x)) > p$ (odnosno $p-1$ ako nije uobičajeni način zaokruživanja), vidimo da mjera $-\log_2(\delta(x))$ daje broj binarnih znamenaka do kojeg se x i $\operatorname{round}(x)$ podudaraju. Slično će $-\log_{10}(\delta(x)) > -\log_{10}(u)$ mjeriti broj decimalnih mjesta u kojima se x i $\operatorname{round}(x)$ podudaraju.

S obzirom da je

$$\frac{|\operatorname{round}(x) - x|}{|\operatorname{round}(x)|} < \frac{2^{-p+1} \times 2^e}{2^e} = 2^{-p+1} \equiv u,$$

odnosno,

$$\frac{|\operatorname{round}(x) - x|}{|\operatorname{round}(x)|} \leq \frac{2^{-p} \times 2^e}{2^e} = 2^{-p} \equiv u,$$

ako je prisutno zaokruživanje do najbližeg, možemo pored relacije (3.3.2) koristiti i relaciju

$$\operatorname{round}(x) = \frac{x}{1+\delta}, \quad |\delta| \leq u. \tag{3.3.3}$$

Pritom δ iz relacije (3.3.3) ne mora biti jednak onom iz relacije (3.3.2). Relacije (3.3.2) i (3.3.3) su osnova za analizu grešaka zaokruživanja koju ćemo upoznati kasnije.

3.3.4. Korektno zaokružene osnovne računske operacije

Jedna od najznačajnijih značajki IEEE standarda je da zahtijeva vrlo poželjnu preciznost osnovnih računskih operacija: rezultat mora biti takav kao da je izračunat točno i tek onda zaokružen.

Označimo sa $\oplus$, $\ominus$, $\otimes$ i $\oslash$ operacije $+$, $-$, $\times$, i $/$ kako su stvarno implementirane u računalu. Ako s $\circ$ i $\odot$ označimo odgovarajuće operacije u algebri i u računalu, tada IEEE standard zahtijeva da vrijedi

$$x \odot y = fl(x \circ y) = \operatorname{round}(x \circ y) = (x \circ y)(1+\delta), \quad |\delta| \leq u, \tag{3.3.4}$$

gdje su $\circ \in \{+, -, \times, /\}$, $\odot \in \{\oplus, \ominus, \otimes, \oslash\}$, a x i y su reprezentabilni brojevi. Osim za četiri osnovne računske operacije, standard zahtijeva da (3.3.4) vrijedi i za unarnu operaciju drugog korijena. Konačna aritmetika koja zadovoljava (3.3.4) katkad se naziva **aritmetika s korektim zaokruživanjem**.

Lijeva strana u (3.3.4) je broj u skupu BPT, pa vrijedi: $1 \otimes x = x$, $x \oslash x = 1$, $0.5 \otimes x = x \oslash 2$ kao i važno svojstvo: ako su x i y brojevi u skupu BPT i vrijedi $x \ominus y = 0$, tada je $x = y$.

Korištenjem izvoda i same relacije (3.3.3), odmah vidimo da vrijedi i ocjena

$$x \odot y = fl(x \circ y) = \frac{x \circ y}{1+\delta}, \quad |\delta| \leq u, \tag{3.3.5}$$

pri čemu zadnji δ nije općenito jednak δ iz relacije (3.3.4).

Napomena 3.3.1 *U numeričkoj matematici uvriježila se notacija $fl(x+y)$, $fl(x-y)$, $fl(x \times y)$, $fl(x/y)$ za operacije $x \oplus y$, $x \ominus y$, $x \otimes y$ i $x \oslash y$. Također, $fl(izraz)$ označava izračunatu vrijednost izraza pomoću operacija $\oplus$, $\ominus$, $\otimes$ i $\oslash$. Slično, $fl(f(x))$ označava izračunatu vrijednost funkcije f u točki x.*

Zadatak 3.3.11 *Koji je najveći broj x iz skupa BPT za koji je $1 \oplus x = 1$. Pretpostavke su: IEEE jednostruki format i zaokruživanje do najbližeg. Uz iste uvjete, koliko je $1 \oplus \text{round}(10^{-5})$, $1 \oplus \text{round}(10^{-10})$, $1 \oplus \text{round}(10^{-15})$?*

Iz same definicije operacija $\oplus$, $\ominus$ i $\otimes$ slijedi $x \oplus y = y \oplus x$, $x \ominus y = -(y \ominus x)$ i $x \otimes y = y \otimes x$. Međutim, već za malo složenije izraze, ono što vrijedi u aritmetici, ne vrijedi u konačnoj algebri s IEEE standardom.

Primjer 3.3.1 *Neka je $x = 1$, $y = 2^{-25}$, $z = 1$. Ti brojevi su u skupu BPT. Kako je $y = 1.0 \times 2^{-25}$, suma*

$$x + y = 1.00000000000000000000000001$$

se ne može egzaktno reprezemtirati u IEEE jednostrukom formatu, pa će uz zaokruživanje prema najbližem biti

$$x \oplus y = 1.$$

Stoga će vrijediti

$$(x \oplus y) \ominus z = 1 \ominus z = 0.$$

S druge strane, egzaktni je rezultat

$$(x + y) - z = 2^{-25}$$

reprezentabilan u sustavu brojeva BPT, pa vrijedi

$$\text{round}((x + y) - z) = 2^{-25}.$$

Pokazali smo da je $fl((x + y) - z) = 0$, iako je $\text{round}((x + y) - z) = 2^{-25}$, pa za taj izraz ne vrijedi pravilo koje je aksiom IEEE standarda za osnovne računske operacije.

Slično, asocijativnost zbrajanja i distributivnost množenja prema zbrajanju, neće u IEEE aritmetici uvijek vrijediti.

Zadatak 3.3.12 *Neka su $x = 1$, $y = 2^{-25}$, $z = 1.1 \times 2^{-25}$. Uz pretpostavku IEEE jednostrukog formata i zaokruživanja do najbližeg, izračunajte*

$$a \equiv (x \oplus y) \oplus z,$$
$$b \equiv x \oplus (y \oplus z).$$

Pokažite da a i b nisu jednaki!

Rješenje

$$x \oplus y = 1,$$
$$(x \oplus y) \oplus z = 1 \oplus z = 1,$$
$$y \oplus z = 2.1 \times 2^{-25},$$
$$x \oplus (y \oplus z) = 1 \oplus 2.1 \times 2^{-25} = 1 + 2^{-23}.$$

■

3.3.5. Implementacija operacija na računalu

Počnimo od zbrajanja i oduzimanja, koje nećemo posebno razlikovati, jer je

$$x - y = x + (-y).$$

Ako operandi nemaju isti eksponent, onda se (po modulu) manji operand napiše u obliku nenormaliziranog broja s eksponentom većeg. To znači da mu se mantisa pomakne udesno za onoliko binarnih mjesta kolika je na početku bila razlika u eksponentima operanada. Zatim se rezultat svede na normalizirani oblik, pri čemu se prvo mantisa pomakne ulijevo ili udesno ako je potrebno, zaokruži i, ako je nakon zaokruživanja potrebno, opet se pomakne. Svako pomicanje mantise rezultira istovremenim pomakom eksponenta.

Sigurnosni i zalijepljeni bitovi

Da bi se implementirala IEEE aritmetika potrebni su i tzv. **sigurnosni** (**dodatni, zaštitni, rezervni**) bitovi. To znači da aritmetika u računalu koristi registre koji sadrže riječ s više bitova nego što zahtijeva standard za reprezentaciju brojeva.

Promotrimo računanje razlike $x - y$ u jednostrukom formatu, pri čemu je

$$x = (1.0)_2 \times 2^0 \quad \text{i} \quad y = (1.111\ldots1)_2 \times 2^{-1}.$$

Izjednačavanjem eksponenata, dobije se

$$\begin{aligned} &(1.00000000000000000000000|\ )_2 \times 2^0 \\ -\ &(0.11111111111111111111111|1)_2 \times 2^0 \\ =\ &(0.00000000000000000000000|1)_2 \times 2^0 \end{aligned}$$

normalizacija:

$$= (1.00000000000000000000000|0)_2 \times 2^{-24}.$$

Ovdje smo koristili jedan dodatni bit (iza okomite crte | koja prikazuje granicu). Ovo je primjer **kraćenja**, jer se skoro svi bitovi rezultata pokrate.

Sljedeći primjer pokazuje da katkad nije dovoljan jedan bit. Promotrimo računanje $x - y$ u jednostrukom formatu uz zaokruživanje do najbližeg, pri čemu su

$$x = (1.0)_2 \times 2^0 \quad \text{i} \quad y = (1.000\ldots01)_2 \times 2^{-24}$$

brojevi iz BSPT.

Korištenjem (čak!) 25 dodatnih bitova, dobivamo

$$\begin{aligned} &(1.00000000000000000000000|\qquad\qquad\qquad\qquad\qquad\qquad)_2 \times 2^0 \\ -\ &(0.00000000000000000000000|01000000000000000000000001)_2 \times 2^0 \\ =\ &(0.11111111111111111111111|10111111111111111111111111)_2 \times 2^{-1} \end{aligned}$$

normalizacija:

$$= (1.11111111111111111111111|01111111111111111111110)_2 \times 2^{-24}$$

zaokruživanje:

$$= (1.11111111111111111111111| \qquad\qquad\qquad\qquad)_2 \times 2^{-24}.$$

Da smo koristili samo dva dodatna bita, rezultat bi (nakon dodatne renormalizacije) bio $(1.00000000000000000000000)_2 \times 2^0$. Isti (pogrešan) rezultat dobivamo korištenjem $3, 4, \ldots, 24$ dodatna bita. Na sreću, umjesto 25 (ili u sličnim namještenim slučajevima po volji veliki broj) dodatnih bitova, dovoljno je koristiti tek tri dodatna bita, točnije, dva zaštitna i jedan **zalijepljeni** bit. Zovemo ga zalijepljeni jer kad ga aktiviramo, on se više ne mijenja. On se aktivira onda kad je potrebno pomaknuti mantisu za više od dva bita i ta jedinica se stavlja iza drugog zaštitnog bita. Dakle, u zadnjem primjeru, prije oduzimanja treba dodati zalijepljenu jedinicu iza drugog zaštitnog bita:

$$\begin{aligned} & (1.00000000000000000000000| \quad)_2 \times 2^0 \\ - & (0.00000000000000000000000|011)_2 \times 2^0 \\ = & (0.11111111111111111111111|101)_2 \times 2^{-1} \end{aligned}$$

normalizacija:

$$= (1.11111111111111111111111|01 \ )_2 \times 2^{-24}$$

zaokruživanje:

$$= (1.11111111111111111111111| \quad)_2 \times 2^{-24}.$$

Množenje i dijeljenje

Množenje i dijeljenje u BSPT ne zahtijeva poravnavanje eksponenata. Ako je

$$x = m_x \times 2^{e_x} \quad \text{i} \quad y = m_y \times 2^{e_y},$$

tada je

$$z = x \cdot y = (m_x \cdot m_y) \times 2^{e_x + e_y},$$

pa pri implementaciji množenja u BSPT postoje tri koraka:

- množenje mantisa,
- zbrajanje eksponenata,
- ako je potrebno normalizacija produkta mantisa i pravilno zaokruživanje rezultata.

Kod dijeljenja isto ćemo tako računati

- kvocijent mantisa,
- razliku eksponenata,
- i normalizaciju sa zaokruživanjem.

Današnji dizajneri čipova uspjeli su, uz dovoljnu rezervu memorije, toliko ubrzati množenje da je gotovo jednako brzo kao i zbrajanje, odnosno oduzimanje. Jedino treba voditi računa da je dijeljenje još uvijek nekoliko puta sporije od množenja. Dijeljenje s nulom razmotrit ćemo kasnije.

Uočimo da nejednakosti $1 \leq m_x < 2$ i $1 \leq m_y < 2$ povlače

$$1 \leq m_z = m_x \cdot m_y < 4.$$

Dakle, lijevo od binarne točke u binarnoj reprezentaciji od m_z mogu biti samo kombinacije bitova $11, 10$ ili 01 tj 1. Stoga se mogući izbor pomaka mantise svodi najviše za jedno mjesto ulijevo uz istodobno povećanje eksponenta za 1.

Kod dijeljenja će vrijediti

$$\frac{1}{2} < \frac{m_x}{m_y} < 2,$$

pa će pomicanje kod normalizacije biti najviše jedan bit ulijevo ili udesno.

Zadatak 3.3.13 *Odgovorite na sljedeća pitanja u slučaju:*

- *brojevnog sustava za prikazivanje ($p = 3$, $\epsilon_M = 0.25$, $-1 \leq e \leq 1$) i*
- *BSPT u jednostrukom IEEE formatu.*

(i) Koliko brojeva x s pomičnom točkom zadovoljava $1 \leq x < 2$? Koliko od njih zadovoljava $1 \leq x < 3/2$, a koliko $3/2 \leq x < 2$?

(ii) Koliko brojeva y s pomičnom točkom zadovoljava $1/2 < y \leq 1$? Koliko od njih aproksimativno zadovoljava $1/2 < x \leq 2/3$, a koliko $2/3 < x \leq 1$?

(iii) Moraju li postojati dva različita broja s pomičnom točkom između 1 i 2 za koje su izračunate recipročne vrijednosti $1 \oslash x_1$ i $1 \oslash x_2$ jednake (zaokružene u istom formatu)? Odnosi li se to na x_1 i x_2 između 1 i $3/2$ ili između $3/2$ i 2? Vrijedi li to bez obzira na način zaokruživanja?

(iv) Postoje li brojevi s pomičnom točkom za koje $(1 \oslash x) \otimes x$ nije jednako 1? Vrijedi li to za svaki način zaokruživanja? Postoje li brojevi s pomičnom točkom za koje $1 \oslash (1 \oslash x)$ nije jednako x?

3.3.6. Drugi korijen, ostatak pri dijeljenju i konverzija formata

Pored zahtjeva da su osnovne aritmetičke operacije korektno zaokružene, IEEE standard zahtijeva da su korektno zaokružene i operacija drugog korijena i ostatak pri dijeljenju. Drugi korijen je unarna operacija (jer je funkcija) definirana za

nenegativni operand pri čemu u implementaciji (koja je najčešće hardverska) vrijedi $\sqrt{-0} = -0$. Ostatak pri dijeljenju realnih brojeva x i y je realni broj $x - y \times n$, gdje je n cijeli broj najbliži kvocijentu x/y.

Brojevi ulaze u računalo na nekoliko načina. Kod računanja na računalu najčešće se koriste viši programski jezici koji procesiraju naredbe kao prevoditelji ili interpreteri. Ako npr. broj 0.4 sudjeluje u računu, njega možemo u programu uvesti kao konstantu ili pridružiti tu vrijednost nekoj varijabli. Pridruživanje varijabli može biti u naredbi oblika $x = 0.4$, a x može dobiti vrijednost i učitavanjem broja 0.4 s neke vanjske memorije, npr. iz neke tekst datoteke na disku. Prevodilac ili interpreter tada poziva standardne rutine odnosno potprograme za ulaz podataka. Te rutine generiraju strojne instrukcije koje prevode decimalni broj, tj. niz znakova (koji su: decimalne znamenke, točka i možda slova kao e, E, d, D koje označavaju početak eksponencijalnog dijela broja) u binarni format i pretvore ga kao korektno zaokruženi broj u varijablu memorije ili registra. Isto tako, moguće je 0.4 unijeti kao razlomak 4/10, dakle unijeti (cijele ili realne) brojeve 4 i 10, a zatim unutar programa koristiti operaciju dijeljenja da bismo izračunali 0.4. Ako se 4 i 10 unose kao cijeli (realni) brojevi, prevodilac ili interpreter će generirati strojne instrukcije koje će nizove znakova '4' i '10' pretvoriti u njihove cjelobrojne (ili realne IEEE) binarne reprezentacije. U svakom slučaju, prije nego što će se pozvati operacija dijeljenja, operandi moraju biti u binarnoj reprezentaciji pomične točke jer će rezultat biti u tom formatu. Kod izlaza podataka iz računala, npr. kod ispisa ili spremanja rezultata u tekst datoteku, prevodilac ili interpreter (kraće kažemo – računalo) generira instrukcije za konverziju binarnih formata u decimalne brojeve, koje smo navikli prepoznavati.

IEEE standard zahtijeva podršku za korektno prevođenje između raznih formata. Tu spadaju sljedeće pretvorbe ili konverzije.

- Konverzije između formata brojeva s pomičnom točkom. Konverzija iz kraćeg u dulji format (npr. iz jednostrukog u dvostruki) mora biti egzaktna. Konverzije iz dužeg u kraći zahtijeva zaokruživanje.
- Konverzija iz BSPT u cjelobrojni format zahtijeva zaokruživanje do najbližeg cijelog broja uz trenutno aktivni način zaokruživanja. Ako je broj u BSPT cijeli broj, konverzija mora dati isti cijeli broj, osim ako se taj cijeli broj ne može reprezentirati u danom cjelobrojnom formatu. Obrnuta konverzija, iz cjelobrojnog formata u format s pomičnom točkom možda će zahtijevati zaokruživanje. Također, zahtijeva se zaokruživanje broja iz skupa BSPT u cijeli broj reprezentiran istim formatom.
- Konverzija iz decimalnog u binarni sustav i obrnuto. Pritom se koristi način zaokruživanja koji je aktivan. Nakon 1985. kad je uveden IEEE 754 standard, pronađeni su novi i/ili efikasniji algoritmi za korektnu konverziju između svih tipova formata i oni često imaju bolja svojstva nego što to standard zahtijeva.

3.3.7. Izuzeci

U matematici su realni ili cijeli brojevi uvijek konačni. Oznake ∞ ili $-\infty$ tek služe za naznačavanje da vrijednost neke veličine (npr. opći član niza brojeva) teži prema sve većim ili manjim brojevima. U ovom dijelu teksta, misleći na brojeve s kojima računalo radi, spominjemo ∞ i $-\infty$ kao brojeve. Isto tako $+0$ i -0, jer imaju različitu reprezentaciju, spominjemo kao dva različita broja, iako imaju istu vrijednost 0.

Najjednostavniji primjer jedne izuzetne situacije je dijeljenje s nulom. Prije IEEE standarda neki su proizvođači računala rezultat dijeljenja pozitivnog broja s nulom definirali najvećim prikazivim brojem. Obrazlagali su to tvrdnjom da kada korisnik primijeti vrlo velike brojeve, znat će da je nešto krenulo loše. Tada bi rezultat operacije kao što je $1/0-1/0$ bio 0, pa to korisnik vjerojatno ne bi primijetio, pogotovo ako se takva stvar dogodila usred složenijeg računanja. Stoga su nakon 1960. proizvođači računala usvojili pravilo da se izvršavanje programa prekida ako se broj dijeli s nulom, uz poruku oblika: "fatalna greška – dijeljenje s nulom". Na taj način je bilo otežano programiranje, jer se smatralo da je odgovornost programera da ne dođe do djeljenja s nulom. Kako je to razriješio IEEE standard?

Matematički je jasno da vrijedi: $a \times 0 = 0$ za svaki konačni broj a, $a/0 = \infty$ za svaki pozitivni broj a, $a \times \infty = \infty$ za svaki pozitivni a. S druge strane izrazi poput $0 \times \infty$, $0/0$, ∞/∞ nemaju smisla, pa je najbolje da računalo to tretira kao nepravilnu operaciju. Stoga IEEE standard zahtijeva da se svakoj takvoj operaciji pridruži vrijednost NaN (Not a Number - nije broj). Ako se NaN pojavi u nekom izrazu, tada cijeli izraz dobiva vrijednost NaN. Na kraju će, možda sve, varijable s izlaznim vrijednostima imati sadržaj NaN. Na taj način doći će poruka programeru da je nešto krenulo loše.

Zbrajanja u kojima se pojavljuje ∞ uglavnom imaju smisla: $a + \infty = \infty$, $a - \infty = -\infty$ za svaki konačni broj a, pa je i $\infty + \infty = \infty$, $-\infty + (-\infty) = -\infty$. To se lako pokaže pomoću konvergirajućih (prema a) i divergirajućih (prema ∞) nizova. Međutim, $\infty - \infty$ nema smisla pa je $\infty - \infty = \text{NaN}$. Slično, $\infty/0 = \infty$, $0/\infty = 0$, dok $0/0$ i ∞/∞ nemaju smisla jer mogu u interpretaciji nizova biti bilo koji brojevi. Dakle je $0/0 = \text{NaN}$ i $\infty/\infty = \text{NaN}$.

S obzirom da standard uvodi predznake ispred posebnih brojeva 0 i ∞: $+0$ je isto što i 0 pa koristimo 0 i -0; $+\infty$ je isto što i ∞ pa koristimo ∞ i $-\infty$. Pogledajmo kako ih iskoristiti. Ako je a pozitivni konačni broj, stavljamo $a/0 = \infty$, $a/-0 = -\infty$, $-a/-0 = \infty$, $-a/0 = -\infty$ kao i $a/\infty = 0$, $a/-\infty = -0$, $-a/\infty = -0$, $-a/-\infty = 0$. Predikat $0 = -0$ je istinit, dok je predikat $\infty = -\infty$ lažan. Na taj način smo došli do toga da implikacija

$$a = b \quad \Longrightarrow \quad \frac{1}{a} = \frac{1}{b}$$

ne vrijedi ako je $a = 0$ i $b = -0$.

Upotreba NaN-ova prikladna je ako se zahtijeva vrijednost funkcije za argument koji nije u domeni funkcije, npr. rezultat operacije $\sqrt{-3.2}$ će biti NaN. Također, zgodno je polazne vrijednosti varijabli koje nemaju polazne vrijednosti definirati s NaN, jer će nam to otkriti grešku njihova preranog korištenja (prije no što dobiju neku vrijednost) u programu. NaN nije u uređaju ni s jednim brojem, a rezultat operacije $a \odot \mathrm{NaN}$ je NaN za svaki reprezentabilni broj i svaku aritmetičku operaciju $\odot$.

Zadatak 3.3.14 *Ima li još slučajeva kad implikacija*

$$a = b \quad \Longrightarrow \quad \frac{1}{a} = \frac{1}{b}$$

ne vrijedi? Koje su vrijednosti izraza: $0/-0$, $\infty/-\infty$ *i* $-\infty/-0$*?*

3.3.8. Prekoračenje, potkoračenje i postupno potkoračenje

O **prekoračenju** (engl. overflow) govori se kad je egzaktni rezultat neke operacije konačan broj, ali po modulu je veći od najvećeg prikazivog broja u BSPT kojeg označavamo s $N_{\max}$. Uočimo da $N_{\max}$ ovisi o binarnom formatu. Prema IEEE standardu, takav broj treba zaokružiti prema načinu zaokruživanja koji se koristi. Ako je broj koji prekoračuje pozitivan, tada će zaokruživanje prema gore zaokružiti broj na ∞, dok će zaokruživanje prema dolje i prema nuli broj zaokružiti na $N_{\max}$. Zaokruživanje prema najbližem, odudara od ove matematičke logike jer će broj zaokružiti na ∞. Razlog je praktične naravi, radi se o osnovnom modu zaokruživanja koji se gotovo uvijek koristi, a kad bi se broj zaokružio na (matematički najbliži BPT) $N_{\max}$, računanje bi moglo dati rezultat koji zavarava – izgleda kao da je sve u redu, a nije.

Potkoračenje (engl. underflow) nastaje kada je egzaktni rezultat neke operacije broj različit od nule koji je po modulu manji od najmanjeg pozitivnog broja prikazivog u BSPT, kojeg označavamo s $N_{\min}$. Prije IEEE standarda tipični odgovor računala je bio stavljanje broja–rezultata na nulu. U IEEE aritmetici, standardni odgovor je pravilno zaokruženi broj (prema aktivnom načinu u BSPT) koji je možda subnormalni broj. Taj događaj se naziva **postupno potkoračenje** (engl. gradual underflow) i on je i danas najkontraverzniji dio IEEE standarda.

Naime, postupno potkoračenje daje rezultate koji imaju manju (to manju što je potkoračenje veće) relativnu točnost. Međutim, i to je u većini slučajeva bolje nego definirati rezultat nulom i tako potpuno izgubiti relativnu točnost. Osim toga postupno potkoračenje osigurava da za brojeve u BSPT vrijedi implikacija

$$x \ominus y = 0 \quad \Longrightarrow \quad x = y.$$

Primjer 3.3.2 *Neka je* $y = 0.1 \times 2^{-123}$, $x = 25 \times 2^{-131}$ *i* $z = y \ominus x$.

Uočimo da je $25 = (11001)_2$. *Računaje broja* z *u jednostrukom IEEE formatu daje*

$$\begin{aligned} &(1.10011001100110011001101)_2 \times 2^{-127} \\ -&(1.10010000000000000000000)_2 \times 2^{-127} \\ =&(0.00001001100110011001101)_2 \times 2^{-127} \end{aligned}$$

normalizacija:

$$= (1.10011001100110011010000)_2 \times 2^{-132}.$$

Pritom je zadnja jedinica u binarnom prikazu broja x *nastala zbog zaokruživanja. Vidimo da je rezultat subnormalni broj, koji je, što se relativne preciznosti tiče, izgubio zadnja četiri bita što odgovara približno jednoj decimalnoj znamenki. Bez postupnog potkoračenja, rezultat bi bio nula.*

Zadatak 3.3.15 *Promotrimo operaciju* $(y \ominus x) \oplus x$, *pri čemu rezultat prve operacije potkoračuje. Koji je rezultat, ako postoji, ili ako ne postoji postupno potkoračenje? Koristite prethodni primjer.*

Zadatak 3.3.16 *Neka su* x *i* y *brojevi u BSPT za koje vrijedi*

$$\frac{1}{2} \leq \frac{x}{y} \leq 2.$$

Pokažite da je tada $x - y$ *broj u BSPT, tj. da je tada oduzimanje egzaktno.*

IEEE standard ukupno definira pet vrsta izuzetnih situacija koje su ukratko opisane u sljedećoj tablici.

pogrešna operacija	rezutat je NaN
dijeljenje s nulom	rezultat je $\pm\infty$
prekoračenje	rezultat je $\pm\infty$ ili $\pm N_{max}$
potkoračenje	rezultat je ± 0 ili $\pm N_{min}$ ili subnormalni broj
neegzaktni rezultat	rezultat je korektno zaokruženi broj

Tablica 3.3.5 IEEE odgovor na izuzetne situacije

IEEE standard specificira da se svaki izuzetni događaj mora signalizirati pomoću tzv. statusnog signala (engl. status flag) kojeg bi programer mogao dohvatiti tj. iskoristiti za pravljenje programa (vidi sljedeći primjer) ili pustiti računalo da daje standardne ispise kao u zadnjoj tablici, za vrijeme izvršavanja programa.

Primjer 3.3.3 *Računanje izraza $\sqrt{x^2+y^2}$ daje jednostavan primjer kako se korištenjem IEEE standarda za izuzetne situacije mogu načiniti brzi i pouzdani programi. Pravilo je: pokušaj prvo na jednostavan i brz način, a ako dođe do izuzetne situacije, računaj ponovo na siguran, ali sporiji način. U ovom primjeru, ako je $|x|$ ili $|y|$ veći od $\sqrt{N_{\max}}$, doći će do prekoračenja. Tada se $z = \sqrt{x^2+y^2}$ računa pomoću algoritma:*

$u := \max(|x|, |y|);$
$v := \min(|x|, |y|);$
if $u = 0$ **then**
  $z := 0$
else
  $z := u\sqrt{1+(v/u)^2};$

Ovaj algoritam zahtijeva kao i onaj direktni dva množenja, jedno zbrajanje i jedan korijen, međutim dodatno zahtijeva izvršavanje nekoliko funkcija: $|\cdot|$, $\max$ i $\min$ i jedno dijeljenje, pa je nekoliko puta sporiji i nešto netočniji (vidi primjer 3.5.1).

S IEEE aritmetikom, prvo se izračuna $z = \sqrt{x^2+y^2}$ direktno. Ako se pojavi izuzetna situacija s prekoračenjem (što je vrlo rijedak slučaj), tek tada treba uključiti sofisticiraniji algoritam. Pokažite da je sporiji algoritam prikladan i za slučaj $|x| < \sqrt{N_{\min}}$ i $y < \sqrt{N_{\min}}$.

3.4. Stabilnost numeričkog računanja

U ovom dijelu bavit ćemo se stabilnošću numeričkih algoritama, pojmom usko vezanim uz pozdanost dobivenih rješenja. Susrest ćemo niz pojmova koji određuju ili opisuju točnost kojom algoritam računa izlazne podatke, u prisustvu grešaka zaokruživanja koje su neizbježne kad se koristi konačna aritmetika. Kroz primjere upoznat ćemo neke neugodne fenomene koji se mogu pojaviti kod korištenja konačne odnosno aritmetike računala. Većina primjera je uzeta iz knjige [5].

3.4.1. Greške unazad i unaprijed

Neka je f realna funkcija realne varijable. Pretpostavimo da se u aritmetici preciznosti u vrijednost $y = f(x)$ izračuna kao $\hat{y}$. Kako možemo mjeriti kvalitetu $\hat{y}$ kao aproksimacije od y?

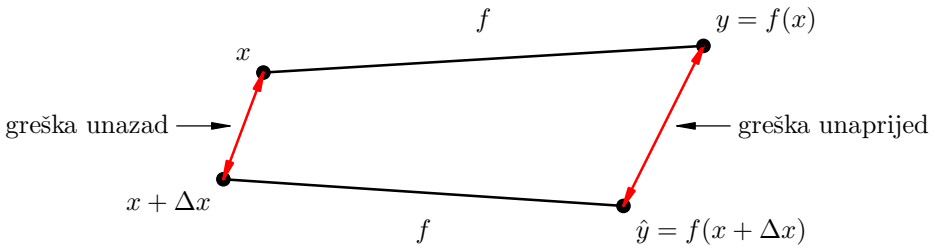
U većini slučajeva bit ćemo sretni ako postignemo malu relativnu grešku u rezultatu, tj. $E_{\text{rel}} \approx \varepsilon$, ali to se neće moći uvijek postići. Umjesto toga možemo se zapitati za koji skup podataka smo zapravo riješili problem? Dakle, za koji Δx

imamo

$$\hat{y} = f(x + \Delta x)?$$

Općenito, bit će više takvih Δx pa će nas zanimati najveći. Vrijednost $|\Delta x|$ (ili $\max|\Delta x|$) možda podijeljena sa $|x|$ zove se **greška unazad** (engl. backward error) ili **povratna greška**. Apsolutna i relativna greška od $\hat{y}$ zovu se **greške unaprijed** ili kraće **greške**. Proces omeđivanja (tj. traženja ograda za) povratne greške izračunatog rješenja zove se analiza povratne greške ili **povratna analiza greške** (engl. backward error analysis), a motivacija za taj postupak je dvostruka.

Prvo, ona interpretira greške zaokruživanja kao greške u podacima. Podaci često kriju netočnosti nastale zbog prethodnih izračunavanja, zbog spremanja u računalo ili kao rezultat mjerenja. Ako povratna greška nije veća od tih polaznih netočnosti, tada se izračunato rješenje ne može mnogo kritizirati jer je to rješenje koje tražimo do na "ulaznu" netočnost. Druga privlačnost povratne analize grešaka je što ona reducira problem omeđivanja greške unaprijed na primjenu teorije perturbacije za dani problem. Teorija perturbacije je dobro poznata za većinu problema i važno je da ona ovisi o problemu, a ne o pojedinoj metodi za dani problem. Kada dobijemo ocjenu za povratnu grešku rješenja kod primjene određene metode, tada dodatnom primjenom opće teorije perturbacije za dani problem lako dolazimo do ocjene za grešku unaprijed.



Slika 3.4.1 Greške unaprijed i unazad

Metoda za računanje vrijednosti $y = f(x)$ je **povratno stabilna** ili **stabilna unazad** (engl. backward stable), ako ona za svako x producira izračunati $\hat{y}$ s malom povratnom greškom, tj. vrijedi $\hat{y} = f(x + \Delta x)$ za malo Δx. Oznaka **mala** ovisi o kontekstu. U načelu, za dani problem može postojati više metoda od kojih će neke biti povratno stabilne, a neke neće.

Npr. sve osnovne računske operacije u računalu zadovoljavaju relaciju (3.1.1) pa daju rezultat koji je točan za malo pomaknute polazne podatke: $x \to x(1 + \delta)$ i $y \to y(1 + \delta)$ uz $|\delta| \leq u$. Dakle, sve su osnovne operacije u računalu povratno stabilne.

Međutim, većina metoda za računanje funkcije $\cos(x)$ ne zadovoljava relaciju $\hat{y} = \cos(x + \Delta x)$ za malo Δx, već samo slabiji rezultat:

$$\hat{y} + \Delta y = \cos(x + \Delta x)$$

uz male Δx i Δy. Greška u rezultatu koji se zapisuje

$$\hat{y} + \Delta y = f(x + \Delta x), \quad |\Delta y| \leq \eta |y|, \quad |\Delta x| \leq \xi |x| \tag{3.4.1}$$

naziva se **miješana naprijed-nazad** greška. Ako su ξ i η u (3.4.1) mali, može se reći: izračunato rješenje $\hat{y}$ se jedva razlikuje od vrijednosti $\hat{y} + \Delta y$ koje se dobije **egzaktnim računom** na ulaznoj vrijednosti $x + \Delta x$ koja se jedva razlikuje od stvarnog ulaznog podatka x.

Algoritam je **numerički stabilan** ako je stabilan u smislu relacije (3.4.1) s malim ξ i η. Ova definicija uglavnom se odnosi na izračunavanja u kojima su greške zaokruživanja (osnovnih aritmetičkih operacija) dominantni oblici grešaka. Inače, pojam stabilnosti ima različita značenja u drugim područjima numeričke matematike.

3.4.2. Uvjetovanost

Odnos između greške unaprijed i greške unazad za dani problem u velikoj mjeri određen je uvjetovanošću problema, tj. osjetljivošću rješenja problema na ulazne podatke.

Pretpostavimo da je dano približno rješenje $\hat{y}$ problema $y = f(x)$ koje zadovoljava $\hat{y} = f(x + \Delta x)$. Ako pretpostavimo da je f dvaput neprekidno derivabilna, razvoj u Taylorov red daje

$$\hat{y} - y = f(x + \Delta x) - f(x) = f'(x)\Delta x + \frac{f''(x + \Theta \Delta x)}{2!}(\Delta x)^2, \quad \Theta \in (0, 1)$$

i možemo mu ocijeniti desnu stranu. Zbog

$$\frac{\hat{y} - y}{y} = \frac{f'(x)}{f(x)}\Delta x + \frac{f''(x + \Theta \Delta x)}{2f(x)}(\Delta x)^2,$$

imamo

$$\frac{\hat{y} - y}{y} = \frac{x f'(x)}{f(x)}\frac{\Delta x}{x} + \mathcal{O}((\Delta x)^2),$$

pa veličina

$$c(x) = \left| \frac{x f'(x)}{f(x)} \right|$$

mjeri relativnu promjenu y za malu relativnu promjenu x. Zato $c(x)$ možemo zvati (relativni) **broj uvjetovanosti** od f, ili kraće **uvjetovanost** od f. Ako su f ili x vektori, tada se broj uvjetovanosti definira na sličan način korištenjem norme. Uvjetovanost služi za mjerenje najveće relativne promjene koja se dostiže za neku vrijednost broja ili vektora x. Npr. za $f(x) = \ln(x)$ imamo $c(x) = 1/|\ln(x)|$, pa

je uvjetovanost velika za $x \approx 1$. To znači da mala relativna promjena u x uvijek izazove malu apsolutnu promjenu promjenu u $f(x) = \ln(x)$, jer je

$$f(x + \Delta x) - f(x) \approx f'(x)\Delta x = \frac{\Delta x}{x},$$

ali i veliku relativnu promjenu za neke x.

Kad se greške unaprijed, unazad, te uvjetovanost za dani problem definiraju na konzistentan način, vrijedi jednostavno pravilo:

$$\text{greška unaprijed} \lessapprox \text{uvjetovanost} \times \text{greška unazad}.$$

Dakle, izračunato rješenje loše uvjetovanog problema može imati veliku grešku unaprijed. Čak i kad izračunato rješenje ima malu grešku unazad, prijelazom na grešku unaprijed, ona se može povećati za faktor velik kao broj uvjetovanosti. Zato se uvodi sljedeća definicija.

Definicija 3.4.1 *Ako metoda daje rješenja s greškama unaprijed, koja su sličnog reda veličine kao ona koja se dobiju primjenom povratno stabilne metode, tada se za metodu kaže da je stabilna unaprijed.*

Dakle, sama metoda ne treba biti povatno stabilna da bi bila stabilna unaprijed. Povratna stabilnost implicira stabilnost unaprijed, dok obrat ne vrijedi (primjer je npr. Cramerovo pravilo za 2×2 linearne sustave).

3.4.3. Akumulacija grešaka zaokruživanja

Dosta je rasprostranjeno mišljenje da velike brzine modernih računala koje omogućavaju u svakoj sekundi izvršavanje nekoliko milijardi računskih operacija imaju pri zahtjevnim proračunima za posljedicu potencijalno zastrašujuće velike greške u rezultatu. Na sreću, ta tvrdnja uglavnom nije istinita, a u rijetkim slučajevima kada dolazi do većih grešaka u rezultatu, kriva je jedna, ili tek nekoliko, podmuklih grešaka zaokruživanja.

Primjer 3.4.1 (Slučaj složenih kamata.) *Ako se a novčanih jedinica investira na godinu dana po godišnjoj kamatnoj stopi x (npr. $x = 0.05$), uz n ukamaćivanja (npr. za kvartalno ukamaćivanje je $n = 4$), tada je buduća vrijednost uloženog novca nakon jedne godine, dana formulom*

$$C_n(x, a) = aC_n(x), \quad \textit{gdje je} \quad C_n(x) = \left(1 + \frac{x}{n}\right)^n.$$

To je tzv. formula složenog ukamaćivanja. Poznato je da kad broj ukamaćivanja tijekom godine n raste, $C_n(x)$ monotono raste prema vrijednosti e^x. U graničnom

slučaju, kad $n \to \infty$, govorimo o neprekidnom ukamaćivanju. Taj slučaj se u praksi manje susreće u bankarstvu, a više u biološkim i medicinskim područjima. Tada se govori o prirastu šume, razmnožavanju kunića u nekoj populaciji ili širenju zaraze u medicini. Nas će ovdje zanimati ulaganje novca pa ćemo pretpostaviti da je n konačan. Proučit ćemo stabilnost nekoliko algoritama za računanje $C_n(x)$. No, prvo ćemo odrediti uvjetovanost za $C_n(x)$. Imamo

$$\kappa(C_n(x)) = \left|\frac{xC_n'(x)}{C_n(x)}\right| = \left|\frac{x}{1+\frac{x}{n}}\right| = \frac{|x|}{|1+\frac{x}{n}|},$$

pa uvjetovanost konvergira prema $|x|$ (provjerite da je to uvjetovanost od e^x) kad n raste. Dakle, velika je tek kad je $|x|$ veliko. Promotrimo sljedeće algoritme za računanje $C_n(x)$.

Algoritam 3.4.1

$z := 1.0 + x/n;$
$w := 1.0;$
for $i := 1$ **to** n
  $w := w * z;$
$C := w;$

Algoritam 3.4.2

$z := 1.0 + x/n;$
$C := \mathrm{pow}(z, n);$

*Kako n može biti velik, pametno je iskoristiti operaciju (ili funkciju) potenciranja, ako postoji. U FORTRANU se x^n piše x**n, dok se u jeziku C poziva funkcija* $\mathrm{pow}(x, n)$. *U Matlabu je oblika x^n. Prednost je u tome što algoritam koji stoji iza operacije ili funkcije potenciranja zahtijeva oko* $\log_2(n)$ *množenja, umjesto n množenja kao u prvom algoritmu (npr. x^{17} se računa kao $x \cdot (((x^2)^2)^2)^2$, pa koristi 5 množenja). Što je manje množenja, bit će i manje grešaka zaokruživanja.*

Sljedeća je mogućnost iskoristiti identitet $z^n = e^{n\ln(z)}$. Pritom možemo koristiti sljedeća dva algoritma

Algoritam 3.4.3

$z := 1.0 + x/n;$
$w := \log(z);$
$C := \exp(n * w);$

Algoritam 3.4.4

$C := \exp(n * \log(1.0 + x/n));$

U tablici su prikazani rezultati koje daju sva četiri algoritma za $x = 0.05$. Programi su napisani u jeziku FORTRAN 77 i korišten je Digital Visual Fortran 6.0 prevodilac za PC računala. Korišten je jednostruki realni format (REAL), a da bismo dobili referentne (gotovo točne) podatke korišten je i potprogram za računanje složenih kamata u dvostrukom realnom formatu (DOUBLE PRECISION). U sljedećoj tablici

dani su izlazni rezultati.

	Algoritam			
n	3.4.1	3.4.2	3.4.3	3.4.4
4	1.050946	1.050946	1.050946	1.050945
12	1.051163	1.051163	1.051163	1.051162
365	1.051262	1.051262	1.051262	1.051268
1000	1.051215	1.051216	1.051216	1.051270
10000	1.051331	1.051342	1.051342	1.051271
100000	1.047684	1.048839	1.048839	1.051271
1000000	1.000000	1.000000	1.000000	1.051271

Usporedbom s rezultatima izračunatim u dvostrukoj preciznosti, zaključili smo da zadnji stupac daje točne rezultate u danom jednostrukom formatu. To je zato jer zadnji algoritam koristi naredbu $C := \exp(n * \log(1.0 + x/n))$ *koja se zapravo izvršava u registrima aritmetičke jedinice. Kako su na Intel-ovim čipovima registri duljine* 80 *bitova, rezultat je izračunat u proširenoj točnosti, a zaokružen je tek kod spremanja u varijablu* C*.*

Do netočnosti u ostalim algoritmima dolazi u prvom redu zbog greške u varijabli z*. Iako je greška tek u zadnjoj decimali, potenciranjem na visoku potenciju, prisutna greška se može bitno povećati. Npr. za* $n = 100000$*,* $\epsilon = 2^{-24}$ *imamo*

$$(1+\epsilon)^n \approx 1 + n\epsilon = 1.005960464477539.$$

S druge strane, za $n = 1000000$ *imamo*

$$z = 1 + 0.05/10^6 = 1 + 5 \times 10^{-8},$$

pa je $fl(z) = 1$ *što se vidi iz zadnjeg retka tablice. Dakle, do greške dolazi jer funkcija* $\mathrm{pot} : z \mapsto z^n$ *nije dobro uvjetovana za veliko* n*. Provjerimo njezinu uvjetovanost:*

$$\kappa(\mathrm{pot}) = \frac{|z \cdot nz^{n-1}|}{|z^n|} = n.$$

Još je jedan fenomen zanimljiv. U prvom stupcu je rezultat za $n = 10000$ *točniji nego u drugom stupcu, dok je za* $n = 100000$ *manje točan. To dolazi od toga što su greške zaokruživanja i pozitivne i negativne, pa u rezultatu može biti stvarna greška mnogo manja od očekivane.*

Primjer 3.4.2 *Izračunajmo aproksimaciju broja e korištenjem relacije*

$$e = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n,$$

tako da se za n uzmu potencije broja 10. *Sljedeća tablica izračunata je korištenjem jezika FORTRAN 77 u jednostrukoj preciznosti ($u = 2^{-24}$):*

n	e_n	$\lvert e_n - e\rvert$
10^1	2.593743	$1.24539e-01$
10^2	2.704811	$1.34705e-02$
10^3	2.717051	$1.23104e-03$
10^4	2.718597	$3.15107e-04$
10^5	2.721962	$3.68039e-03$
10^6	2.595227	$1.23055e-01$
10^7	3.293968	$5.75686e-01$

,

pri čemu je

$$e_n = \left(1 + \frac{1}{n}\right)^n.$$

Kako dolazi do greške? Uočimo da svaka negativna potencija od 10 *ima grešku jer njen binarni prikaz ima beskonačno mnogo znamenki. Kad se formira* $1+1/n$ *za veće* n *(npr. za* $n = 10^6$ *ili* 10^7*), samo nekoliko značajnih binarnih znamenaka od* $1/n$ *ostaje u konačnom prikazu od* $1 + 1/n$*. Nadalje, potenciranje broja* $1 + 1/n$*, makar izračunato posve točno, mora bitno povećati polaznu grešku. Lako se provjerava da računanje potencija od* $1 + 1/n$ *u dvostrukoj preciznosti ($u = 2^{-53}$), nakon što je broj* $1 + 1/n$ *prvo smješten u jednostruki format, a zatim pretvoren iz jednostrukog u dvostruki format, daje iste brojeve u tablici. Zaključujemo da za netočan rezultat nije kriv veliki broj računskih operacija već jedna podmukla greška.*

Ovaj problem, kao i prethodni, može se riješiti mnogo točnije korištenjem formule

$$\left(1 + \frac{1}{n}\right)^n = \exp\left(n \cdot \ln\left(1 + \frac{1}{n}\right)\right),$$

pri čemu se $\ln(1 + 1/n)$ *računa tako da se za*

$$x = \frac{1}{n}$$

prvo izračuna $y = 1 + x$*, a onda vrijednost funkcije*

$$f(x) = \begin{cases} x, & y = 1, \\ \dfrac{x \ln(x)}{y - 1}, & y \neq 1. \end{cases}$$

Detaljna analiza grešaka zaokruživanja pokazuje da će $f\ell(f(x))$ *biti vrlo točna aproksimacija za* $\ln(1 + 1/n)$*.*

Primjer 3.4.3 *Za $x \geq 0$ sljedeći algoritam ne bi smio promijeniti x:*

za $i = 1$ do 60 računaj
  $x := \sqrt{x}$;
za $i = 1$ do 60 računaj
  $x := x^2$;

Ovaj algoritam ne uključuje oduzimanje i svi međurezultati leže između 1 i x, pa možemo očekivati da je na kraju računanja, izračunati $\hat{x}$ dobra aproksimacija od x. Na kalkulatoru HP48G, polazeći od $x = 100$, algoritam daje $\hat{x} = 1.0$. Zapravo, za svaki x, kalkulator izračuna umjesto $f(x) = x$, funkciju

$$\hat{f}(x) = \begin{cases} 0, & 0 \leq x < 1, \\ 1, & x \geq 1. \end{cases}$$

Dakle, kalkulator daje posve pogrešan rezultat na bazi samo 120 operacija, od kojih svaka ponaosob ima malu relativnu grešku. Zašto je to tako?

Reprezentabilni brojevi na kalkulatoru HP48G zadovoljavaju nejednakost

$$10^{-499} \leq x \leq 9.99999999999 \cdot 10^{499}.$$

Definirajmo funkciju

$$r(x) = x^{\frac{1}{2^{60}}} \quad za \quad x \geq 1,$$

koja odgovara međurezultatu nakon prve petlje. Vrijedi

$$1 \leq r(x) < r(10^{500}) = 10^{\frac{500}{2^{60}}} \approx e^{500 \cdot 2^{-60} \ln(10)} < e^{10^{-15}} = 1 + 10^{-15} + \frac{1}{2}\,10^{-30} + \cdots$$

pa se u kalkulatoru koji radi na 12 decimala $r(x)$ zaokruži na 1. U drugoj petlji 60 kvadriranja jedinice opet daje jedinicu.

Za $0 < x < 1$ imamo $x \leq 0.999999999999$ jer je x reprezentabilan. Stoga za $\sqrt{x}$ vrijedi

$$\sqrt{x} \leq \sqrt{1 - 10^{-12}} = 1 - \frac{1}{2}\,10^{-12} - \frac{1}{8}\,10^{-24} - \cdots = 0.999999999999499999999999987\ldots$$

Ova gornja granica se zaokružuje na broj 0.999999999999. I nakon 60 vađenja drugog korijena u kalkulatoru će biti broj koji nije veći od 0.999999999999. Promotrimo sada kvadriranja. Neka je

$$s(x) = x^{2^{60}}.$$

Imamo

$$\begin{aligned} s(x) &\leq s(0.999999999999) = (1 - 10^{-12})^{2^{60}} = 10^{2^{60} \log(1 - 10^{-12})} \\ &= 10^{2^{60} \ln(1 - 10^{-12}) \log(e)} \approx 10^{-2^{60} \cdot 10^{-12} \log(e)} \approx 3.568 \cdot 10^{-500708}. \end{aligned}$$

Gornja granica je manja od najmanjeg reprezentabilnog broja na kalkulatoru. Isto vrijedi i za svaki $0 < x \leq 1 - 10^{-12}$. *Zato dolazi do zamjene broja s nulom (tzv.* **underflow***).*

*Zaključak je jasan: ništa nije loše u kalkulatoru. Ovo na prvi pogled nedužno računanje iscrpljuje preciznost i rang brojeva u računalu (*12 *značajnih dekadskih znamenki i eksponent s tri dekadske znamenke).*

Primjer 3.4.4 *Poznato je da vrijedi*

$$1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \cdots = \sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} \approx 1.644934066848.$$

Pretpostavimo da ne znamo taj indentitet i da želimo numerički izračunati tu sumu. Najjednostavnija strategija je izračunati parcijalne sume

$$s_n = \sum_{k=1}^{n} \frac{1}{k^2}$$

sve dok se vrijednost od s_n *ne ustali. U FORTRANU 77, u jednostrukoj preciznosti dobiva se* 1.64472532 *za* $n = 4096$. *Ta vrijednost se slaže sa* $\pi^2/6$ *tek u prve* 4 *značajne znamenke. Objašnjenje tako netočnog rezultata leži u činjenici da se zbrajanje vrši od većih prema manjim članovima reda. Za* $n = 4096$ *čini se sljedeći doprinos parcijalnoj sumi:*

$$s_n = s_{n-1} + \frac{1}{n^2} = s_{n-1} + 2^{-24},$$

pri čemu je $s_{n-1} \approx 1.6$. *U jednostrukoj preciznosti računalo radi s mantisama od* 24 *bita pa član koji se dodaje ne daje nikakav doprinos, isto kao i svi sljedeći članovi.*

Najjednostavniji način za popravku te netočnosti je zbrajanje u obrnutom redoslijedu. Nažalost, taj pristup zahtijeva znanje koliko članova zbrojiti. Korištenjem 10^9 *članova dobije se suma* 1.66493406 *koja je korektna na* 8 *značajnih znamenaka.*

Detaljnija analiza o tome kako što točnije računati sumu brojeva može se naći u [5, 4. pogl.].

3.4.4. Kraćenje

Kraćenje nastaje kad se oduzimaju dva približno jednaka broja. To najčešće, iako ne uvijek, ima kao posljedicu netočan rezultat. Promotrimo npr. funkciju

$$f(x) = \frac{1 - \cos(x)}{x^2}.$$

Za $x = 1.2 \cdot 10^{-5}$, vrijednost $\cos(x)$, zaokružena na 10 značajnih znamenki iznosi

$$c = 0.9999999999,$$

tako da je

$$1 - c = 0.0000000001.$$

Dakle, aproksimacija za funkciju f u $x = 1.2 \cdot 10^{-5}$ je

$$\frac{1-c}{x^2} = \frac{10^{-10}}{1.44 \times 10^{-10}} \approx 0.6944,$$

što je očito loše, jer je $0 \leq f(x) \leq 1/2$ za sve $x \neq 0$. Dakle, desetoroznamenkasta aproksimacija za $\cos(x)$ nije dovoljno točna da bi izračunata vrijednost funkcije imala barem jednu točnu znamenku. Problem je u tome što $1 - c$ ima samo jednu značajnu znamenku. Oduzimanje $1 - c$ je **egzaktno**, ali to oduzimanje proizvodi rezulat koji je veličine kao i greška u c. Drugim riječima **oduzimanje podiže značaj prethodne greške**. U ovom primjeru $f(x)$ se može napisati tako da se izbjegne kraćenje, jer uvrštavanjem

$$\cos(x) = 1 - 2\sin^2\left(\frac{x}{2}\right),$$

dobivamo

$$f(x) = \frac{1}{2}\left(\frac{\sin(x/2)}{x/2}\right)^2.$$

Izračunavanje $f(1.2 \cdot 10^{-5})$ pomoću ove formule, korištenjem desetoroznamenkaste aproksimacije za $\sin(x/2)$, daje vrijednost 0.5, koja je točan rezultat na deset značajnih znamenki.

Da bismo dobili dodatni uvid u fenomen kraćenja promotrimo oduzimanje (u egzaktoj aritmetici)

$$\hat{x} = \hat{a} - \hat{b},$$

gdje su

$$\hat{a} = a(1 + \Delta_a), \quad \hat{b} = b(1 + \Delta_b).$$

Članovi Δ_a i Δ_b su relativne greške ili netočnosti u podacima, koje, recimo, dolaze od prethodnih računanja. Izraz za relativnu grešku daje

$$\left|\frac{x - \hat{x}}{x}\right| = \left|\frac{-a\Delta_a + b\Delta_b}{a - b}\right| \leq \max\{|\Delta_a|, |\Delta_b|\}\,\frac{|a| + |b|}{|a - b|}.$$

Ograda za relativnu grešku od $\hat{x}$ je velika ako je

$$|a - b| \ll |a| + |b|,$$

a to je istina ako postoji bitno kraćenje u oduzimanju. Ova analiza pokazuje da se zbog kraćenja (u oduzimanju), postojeće greške ili netočnosti u $\hat{a}$ i $\hat{b}$ povećaju. Drugim riječima, **kraćenje dovodi prethodne greške na vidjelo**.

Važno je znati da kraćenje nije **uvijek** loša stvar. Ima nekoliko razloga. Prvo, brojevi koji se oduzimaju mogu biti bez prethodnih grešaka, dakle točni ulazni podaci. Npr. računanje podijeljenih razlika uključuje mnogo oduzimanja, ali pola tih oduzimanja uključuje polazne podatke, pa su bezopasna za pogodni uređaj točaka. Drugi je razlog što kraćenje može biti simptom loše uvjetovanosti problema pa zato mora biti prisutno. Treće, efekt kraćenja ovisi o ulozi koji taj rezultat (zapravo međurezultat) igra u preostalom računanju. Npr. ako je $x \gg y \approx z > 0$, tada je kraćenje u izrazu $x + (y - z)$ bezopasno.

Promotrimo jedan važan primjer kraćenja koji je karakterističan za područje numeričkog rješavanja diferencijalnih jednadžbi.

Primjer 3.4.5 *Kako odrediti približnu vrijednost derivacije neke realne funkcije realnog argumenta. Ako funkciju označimo s f, tada je*

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

U xy koordinatnom sustavu, derivacija je koeficijent smjera tangente na graf funkcije f u točki $(x, f(x))$ tj. tangens kuta kojeg taj pravac zatvara s apscisom. U ovoj definiciji h može biti i pozitivan i negativan (osim ako se promatra tzv. derivacija slijeva ili zdesna za rubne vrijednosti od x).

Što je h po modulu manji to će kvocijent $(f(x+h) - f(x))/h$ biti točnija aproksimacija $f'(x)$. Za funkcije koje nisu zadane analitički (svojom formulom), nego nekim možda kompliciranim algoritmom ili tabličnim vrijednostima, kvocijent

$$\frac{f(x+h) - f(x)}{h}$$

koji se još zove **podijeljena razlika** *ili* **podijeljena diferencija** *moći će se izračunati barem za neke male vrijednosti h, pa ćemo imati neku informaciju o derivaciji. Što je korak ili pomak h manji, dobit ćemo točniju aproksimaciju za $f'(x)$.*

Pretpostavimo sada da je f dva puta diferencijabilna. Korištenjem Taylorovog razvoja funkcije f oko točke x, dobivamo

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\bar{x}),$$

gdje je $\bar{x} \in [x, x+h]$ ako je $h \geq 0$ i $\bar{x} \in [x+h, x]$ ako je $h < 0$. To nam daje ocjenu

$$\frac{f(x+h) - f(x)}{h} - f'(x) = \frac{h}{2} f''(\bar{x}), \tag{3.4.2}$$

pa možemo očekivati da je aproksimacija to bolja što je $|h|$ manje. Greška (3.4.2) se zove **greška diskretizacije**.

Što se međutim događa ako koristimo konačnu aritmetiku?

Primjer 3.4.6 *Odaberimo funkciju koja je eksplicitno zadana i kojoj želimo numerički izračunati derivaciju. Neka je to npr.*

$$\mathrm{ch}(x) = \frac{e^x + e^{-x}}{2}.$$

Derivacija te funkcija je

$$\mathrm{sh}(x) = \frac{e^x - e^{-x}}{2}.$$

Promotrimo aproksimacije derivacije u točki $x = 1$. Računat ćemo na računalu, koristeći dvostruki IEEE format. Kako je

$$\mathrm{ch}(1) = \frac{e^2 + 1}{2e} \approx 1.5431$$

i

$$\mathrm{ch}'(1) = \mathrm{sh}(1) = \frac{e^2 - 1}{2e} \approx 1.1752,$$

znamo vrijednosti funkcije i derivacije u točki 1. Za h ćemo uzeti padajući niz vrijednosti $0.1, 0.01, 0.001, \ldots, 10^{-20}$. Za računaje izraza $(f(x+h)-f(x))/h$ koristit ćemo sljedeći niz naredbi jezika FORTRAN

$x = 1.0;$
$fx = f(x);$
$der = (\exp(x) - \exp(-x))/2.0;$
do $i = 1, 17$
  $h = 10{**}(-i);$
  $xh = x + h;$
  $fh = f(xh);$
  $dif = fh - fx;$
  $g = dif/h;$
  $apsgr = g - der;$
  $relgr = (g - der)/der$
enddo

gdje je $f(x) = (\exp(x) + \exp(-x))/2$. Vrijednosti dobivene korištenjem prevodioca

DV Fortran 6.0, dane su u sljedećoj tablici:

h	*der*	g	*apsgr*	*relgr*
$0.10e+00$	$0.11752012e+01$	$0.12543792e+01$	$0.79e-01$	$0.67e-01$
$0.10e-01$	$0.11752012e+01$	$0.11829362e+01$	$0.77e-02$	$0.66e-02$
$0.10e-02$	$0.11752012e+01$	$0.11759729e+01$	$0.77e-03$	$0.66e-03$
$0.10e-03$	$0.11752012e+01$	$0.11752783e+01$	$0.77e-04$	$0.66e-04$
$0.10e-04$	$0.11752012e+01$	$0.11752089e+01$	$0.77e-05$	$0.66e-05$
$0.10e-05$	$0.11752012e+01$	$0.11752020e+01$	$0.77e-06$	$0.66e-06$
$0.10e-06$	$0.11752012e+01$	$0.11752013e+01$	$0.78e-07$	$0.67e-07$
$0.10e-07$	$0.11752012e+01$	$0.11752012e+01$	$0.94e-08$	$0.80e-08$
$0.10e-08$	$0.11752012e+01$	$0.11752013e+01$	$0.76e-07$	$0.65e-07$
$0.10e-09$	$0.11752012e+01$	$0.11752022e+01$	$0.96e-06$	$0.82e-06$
$0.10e-10$	$0.11752012e+01$	$0.11752155e+01$	$0.14e-04$	$0.12e-04$
$0.10e-11$	$0.11752012e+01$	$0.11752821e+01$	$0.81e-04$	$0.69e-04$
$0.10e-12$	$0.11752012e+01$	$0.11746160e+01$	$-0.59e-03$	$-0.50e-03$
$0.10e-13$	$0.11752012e+01$	$0.11768364e+01$	$0.16e-02$	$0.14e-02$
$0.10e-14$	$0.11752012e+01$	$0.13322676e+01$	$0.16e+00$	$0.13e+00$
$0.10e-15$	$0.11752012e+01$	$0.00000000e+00$	$-0.12e+01$	$-0.10e+01$
$0.10e-16$	$0.11752012e+01$	$0.00000000e+00$	$-0.12e+01$	$-0.10e+01$

Iz tablice vidimo da su aproksimacije sve bolje do određene vrijednosti h, koja je približno jednaka $\sqrt{\epsilon_M}$. Apsolutna i relativna greška smanjuju se za faktor 10 *što je sukladno formuli (3.4.2). Međutim, kad h postaje sve manji, vrijednosti $f(x+h)$ i $f(x)$ postaju sve bliskije, pa dolazi do jakog kraćenja. Npr. za $h = 10^{-10}$ imamo*

$$f(x+h) - f(x) \approx 1.543080634932764 - 1.543080634815244 = 1.1752 \cdot 10^{-10},$$

pa je izgubljeno 10 *decimalnih znamenka, što znači da relativna greška aproksimacije ne može biti bitno bolja od* 10^{-6}*. Svakim smanjivanjem koraka* 10 *puta, pojačava se kraćenje za jednu decimalnu znamenku, što rezultira relativnom greškom koja je približno* 10 *puta veća. Apsolutna greška je istog reda veličine jer se dobije iz relativne greške množenjem s* $|f'(x)| \approx 1.1752$*.*

Dokle god je f dovoljno glatka, postoji mogućnost dobivanja bolje aproksimacije za $f'(x)$ no što je to korištenjem obične podijeljene razlike. Jedno takvo poboljšanje je **simetrična** ili **centralna podijeljena razlika**

$$\frac{f(x+h) - f(x-h)}{2h}.$$

Razvojem u Taylorov red $f(x+h)$ i $f(x-h)$ oko točke x, pokažite da vrijedi

$$\frac{f(x+h) - f(x-h)}{2h} - f'(x) = \frac{h^2}{12}\left(f'''(\bar{x}_1) + f'''(\bar{x}_2)\right), \tag{3.4.3}$$

gdje su $\bar{x}_1$ i $\bar{x}_2$ u intervalima $[x, x+h]$ i $[x-h, x]$. Dakle, diskretizacijska greška pada s faktorom h^2. Uočite da to vrijedi samo za ekvidistantne mreže čvorova, tj. kad je h isti za cijeli promatrani interval.

Zadatak 3.4.1 *Načinite program sličan prethodnom, koji ispisuje apsolutnu i relativnu grešku za simetričnu podijeljenu razliku kao aproksimaciju derivacije za*

$$f(x) = \mathrm{ch}(x)$$

u točki $x = 1$.

Rezultat bi trebala biti tablica oblika

h	*der*	g	*apsgr*	*relgr*
$0.10e+00$	$0.11752012e+01$	$0.11771608e+01$	$0.20e-02$	$0.17e-02$
$0.10e-01$	$0.11752012e+01$	$0.11752208e+01$	$0.20e-04$	$0.17e-04$
$0.10e-02$	$0.11752012e+01$	$0.11752014e+01$	$0.20e-06$	$0.17e-06$
$0.10e-03$	$0.11752012e+01$	$0.11752012e+01$	$0.20e-08$	$0.17e-08$
$0.10e-04$	$0.11752012e+01$	$0.11752012e+01$	$0.16e-10$	$0.13e-10$
$0.10e-05$	$0.11752012e+01$	$0.11752012e+01$	$-0.62e-10$	$-0.53e-10$
$0.10e-06$	$0.11752012e+01$	$0.11752012e+01$	$-0.62e-09$	$-0.53e-09$
$0.10e-07$	$0.11752012e+01$	$0.11752012e+01$	$-0.17e-08$	$-0.15e-08$
$0.10e-08$	$0.11752012e+01$	$0.11752012e+01$	$-0.35e-07$	$-0.30e-07$
$0.10e-09$	$0.11752012e+01$	$0.11752010e+01$	$-0.15e-06$	$-0.12e-06$
$0.10e-10$	$0.11752012e+01$	$0.11752044e+01$	$0.32e-05$	$0.27e-05$
$0.10e-11$	$0.11752012e+01$	$0.11751711e+01$	$-0.30e-04$	$-0.26e-04$
$0.10e-12$	$0.11752012e+01$	$0.11746160e+01$	$-0.59e-03$	$-0.50e-03$
$0.10e-13$	$0.11752012e+01$	$0.11768364e+01$	$0.16e-02$	$0.14e-02$
$0.10e-14$	$0.11752012e+01$	$0.12212453e+01$	$0.46e-01$	$0.39e-01$
$0.10e-15$	$0.11752012e+01$	$0.00000000e+00$	$-0.12e+01$	$-0.10e+01$
$0.10e-16$	$0.11752012e+01$	$0.00000000e+00$	$-0.12e+01$	$-0.10e+01$

Uvjerite se da greške padaju s faktorom 100, a ne s 10 kako je to bilo prije. Nakon vrijednosti $h = 0.10e-05$ greške rastu, kao i prije, s faktorom oko 10. Za to vrijedi isti argument kao u prethodnm primjeru. Kako objašnjavate činjenicu da je minimalna greška manja nego u prethodnom slučaju?

3.4.5. Kraćenje grešaka zaokruživanja

Nije neobično da se u stabilnim algoritmima greške zaokruživanja krate na taj način da konačni rezultat bude mnogo točniji od međurezultata (tj. veličina

koje se koriste u algoritmu). Ovaj fenomen nije toliko poznat jer se međurezultati obično gledaju tek kad se otkrije da krajnji rezultat nije dovoljno točan. Evo jednog primjera.

Promotrimo računanje funkcije

$$f(x) = \frac{e^x - 1}{x} = 1 + \frac{x}{2!} + \frac{x^2}{3!} + \frac{x^3}{4!} + \cdots = \sum_{i=0}^{\infty} \frac{x^i}{(i+1)!}.$$

Uočimo da $f(x) \to 1$ kad $x \to 0$, što se vidi i iz redova na desnoj strani. Pogledajmo sljedeća dva algoritma koji računaju tu funkciju.

Algoritam 3.4.5

$y = \exp(x)$
if (x .eq. 0.0) **then**
  $f = 1.0$
else
  $f = (y - 1.0)/x$
endif

Algoritam 3.4.6

$y = \exp(x)$
$z = \text{alog}(y)$
if (y .eq. 1.0) **then**
  $f = 1.0$
else
  $f = (y - 1.0)/z$
endif

Loša strana prvog algoritma je što dolazi do jakog kraćenja za male vrijednosti $|x|$. Drugi algoritam, izgleda čudno jer izračunava dvije funkcije exp i ln (funkcija alog) umjesto samo jedne exp, kao što to čini prvi algoritam.

Međutim ako se pogledaju rezulati za $x = 10^{-k}$, $k = 0, 1, \ldots, 15$ u Tablici 3.4.1, vidi se da Algoritam 2 daje za sve x vrlo točne rezultate dok Algoritam 1 daje za rastuće k sve netočnije podatke.

Da bismo dobili uvid što se to događa s Algoritmom 3.4.6, promotrimo računanje za $x = 9 \cdot 10^{-8}$ uz $u = 2^{-24} \approx 6 \cdot 10^{-8}$. Točan rezultat je (do na broj prikazanih znamenki) 1.000000005. Algoritam 3.4.5 daje posve netočan rezultat, kao što se i očekuje

$$f\ell\left(\frac{e^x - 1}{x}\right) \equiv f\ell\left(\frac{1.19209290 \cdot 10^{-7}}{9.0000000 \cdot 10^{-8}}\right) = 1.32454766.$$

Algoritam 3.4.6 daje rezultat koji je točan u svim osim u zadnjoj znamenci (što se i očekuje jer se radi o 9-toj značajnoj znamenki, a aritmetika radi s nepunih 8 znamenki)

$$f\ell\left(\frac{e^x - 1}{\ln(e^x)}\right) \equiv f\ell\left(\frac{1.19209290 \cdot 10^{-7}}{1.19209282 \cdot 10^{-7}}\right) = 1.00000006.$$

Evo sada veličina koje bismo dobili Algoritmom 3.4.6 u egzaktnoj aritmetici (korektno na onoliko decimala koliko se koristi u prikazu)

$$\frac{e^x - 1}{\ln(e^x)} = \frac{9.00000041 \cdot 10^{-8}}{9.00000001 \cdot 10^{-8}} = 1.00000005.$$

x	Algoritam		Egzaktno
	3.4.5	3.4.6	
$0.1000000e+00$	$0.1051710e+01$	$0.1051709e+01$	$0.1051709e+01$
$0.1000000e-01$	$0.1005018e+01$	$0.1005017e+01$	$0.1005017e+01$
$0.1000000e-02$	$0.1000524e+01$	$0.1000500e+01$	$0.1000500e+01$
$0.1000000e-03$	$0.1000166e+01$	$0.1000050e+01$	$0.1000050e+01$
$0.1000000e-04$	$0.1001358e+01$	$0.1000005e+01$	$0.1000005e+01$
$0.1000000e-05$	$0.9536742e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-06$	$0.1192093e+01$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-07$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-08$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-09$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-10$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-11$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-12$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-13$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-14$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$
$0.1000000e-15$	$0.0000000e+00$	$0.1000000e+01$	$0.1000000e+01$

Tablica 3.4.1 Primjena algoritama za $x = 10^{-k}$, $k = 0, 1, \ldots, 15$

Vidimo da Algoritam 3.4.6 izračunava vrlo netočno vrijednosti $e^x - 1$ i $\ln(e^x)$, ali je kvocijent tih vrijednosti vrlo točan rezultat. Zaključak: dijeljenjem u Algoritmu 3.4.6 greške se pokrate (dokinu). Koristeći analizu grešaka zaokruživanja može se objasniti to začuđujuće kraćenje grešaka (vidjeti [5]).

3.4.6. Rješavanje kvadratne jednadžbe

Rješavanje kvadratne jednadžbe

$$ax^2 + bx + c = 0$$

je matematički trivijalan problem (ako je jednadžba zaista kvadratna, tj. ako je $a \neq 0$): postoje dva rješenja

$$x_{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{3.4.4}$$

Ako je $a = 0$, postoji samo jedno rješenje $x = -c/b$ jer je jednadžba linearna.

Numerički, problem je izazovniji, jer niti izračunavanje izraza (3.4.4), niti točnost izračunatog rješenja nije garantirana.

Najjednostavniji aspekt rješavanja kvadratne jednadžbe je izbor formule za računanje rješenja. Ako je $b^2 \gg |4ac|$ tada je $\sqrt{b^2-4ac} \approx |b|$, pa se za jedan izbor predznaka u formuli (3.4.4) događa kraćenje. To je opasno kraćenje jer je jedan od argumenata, $f\ell(\sqrt{b^2-4ac})$ netočan zbog korištenja konačne aritmetike, pa kraćenje pojačava grešku u $f\ell(\sqrt{b^2-4ac})$ (također i u $f\ell(b)$ ako i on nosi grešku u odnosu na b). Kako izbjeći grešku je dobro poznato: treba prvo izračunati veće (po modulu) rješenje, nazovimo ga s x_1, pomoću formule

$$x_1 = \frac{-(b + \text{sign}(b)\sqrt{b^2-4ac})}{2a}. \tag{3.4.5}$$

Drugo rješenje dobije iz Vièteove formule

$$x_1 \cdot x_2 = \frac{c}{a},$$

uz prethodno izračunato rješenje x_1

$$x_2 = \frac{c}{a \cdot x_1} = \frac{-2c}{b + \text{sign}(b)\sqrt{b^2-4ac}}, \tag{3.4.6}$$

pri čemu se obje formule u (3.4.6) mogu koristiti. Formula $c/(a \cdot x_1)$ je vrlo točna, tj. ako se x_1 izračuna s relativnom greškom ε_{x_1}, tada se x_2 izračuna s relativnom greškom ne većom od $\varepsilon_{x_1} + 2u$. Druga formula za x_2 u (3.4.6) je direktnija jer koristi samo koeficijente kvadratne jednadžbe, ali zato ima nešto više operacija od prve. Ocjena greške za x_2 po toj formuli bit će jednaka ocjeni greške za x_1 po formuli (3.4.5). Stoga se možemo posvetiti proučavanju točnosti formule (3.4.5) za x_1.

Nažalost, postoji mnogo opasniji izvor kraćenja, onaj u izrazu b^2-4ac. Točnost se gubi ako je $b^2 \approx 4ac$ (slučaj bliskih korijena) i nikakva algebarska transformacija neće izbjeći kraćenje. Jedini način da se garantira točno izračunata diskriminanta je korištenje povećane preciznosti (ili trikova u postojećoj aritmetici koji su ekvivalentni u rezultatu, korištenju dvostruke preciznosti) u izračunavanju izraza $b^2 - 4ac$.

Zbog važnosti koju kvadratna jednadžba ima u numeričkom računanju, cjelovitu analizu grešaka koje nastaju kod traženja korijena načinit ćemo kad se upoznamo s osnovama analize grešaka zaokruživanja.

Umjesto zaključka, pokušat ćemo ukratko dati upute kako dizajnirati stabilne (tj. točne u okruženju aritmetike računala) algoritme, te pokušati svratiti pažnju na često prisutna kriva uvjerenja vezana uz računanje na računalima. Neke od tvrdnji koje ćemo sada susresti obrazložit ćemo u sljedećim poglavljima jer se tiču specijalnih područja numeričke matematike.

3.4.7. Kako dizajnirati stabilne algoritme

Nema jednostavnog recepta za dizajniranje stabilnih algoritama, ali najbolji je savjet spoznaja da je numerička stabilnost važnija od drugih karakteristika algoritma, kao što su npr. broj računskih operacija, te dobra vektorizacija ili paralelizacija (prilagodljivost algoritma vektorskim ili višeprocesorskim računalima). Evo nekih **općih uputa** u redosljedu kako se spominju u [5]:

1. Pokušajte izbjeći oduzmanje veličina bliskih vrijednosti koje nose greške, iako je to katkad nemoguće.
2. Minimizirajte veličinu međurezultata u odnosu na konačni rezultat. To je važno da konačni rezultat ne bi bio dobiven opasnim oduzimanjem velikih vrijednosti koje nose u sebi greške.
3. Potražite drugačije formulacije istog problema ili druge formule za isti račun.
4. Koristite prednosti jednostavnih formula za ažuriranje tipa

   $$\text{nova vrijednost} = \text{stara vrijednost} + \text{mala korekcija},$$

   ako se mala korekcija može izračunati na dovoljan broj značajnih znamenki (Eulerova metoda, Newtonova metoda, ...).
5. Koristite samo dobro uvjetovane transformacije za dobivanje rješenja. Kod matrica to znači koristiti ortogonalne matrice kad god je to moguće, jer neortogonalne matrice mogu biti loše uvjetovane.
6. Poduzmite mjere opreza da biste spriječili moguća prekoračenja granice konačne aritmetike (overflow i underflow).
7. Kod nekih računala centralna aritmetička jedinica koristi precizniju aritmetiku za operande u registrima, a zaokruživanje nastupa tek kod spremanja podatka u memoriju. To znači da nije povoljno cijepati složenije formule u više programskih linija korištenjem pomoćnih varijabla.

U tijeku računanja uvijek je dobro imati određeno praćenje nekih, po stabilnost važnih, veličina. Ako se u tijeku računanja pamti i ažurira najveći po modulu broj on može biti indikacija što se događa s algoritmom.

S vremenom se nakupilo podosta **krivih uvjerenja** u vezi računanja na računalima. Nabrojimo neka.

1. Kraćenje pri oduzimanju je uvijek loša stvar.
2. Greške zaokruživanja mogu "nadvladati" rezultat samo ako ih ima veliki broj.
3. Kratki račun bez kraćenja i bez prekoračenja granica (underflow i overflow) uvijek mora dati točan rezultat (vidjeti [5]).

4. Povećanje preciznosti aritmetike koja se koristi uvijek povećava točnost rezultata (vidjeti [5]).
5. Izlazni rezultat kod nekog algoritma ne može biti točniji od bilo kojeg međurezultata, tj. greške se ne mogu pokratiti.
6. Greške zaokruživanja mogu samo pogoršati, ali ne i pomoći u uspjehu računanja. Ovdje je tipični primjer metoda inverznih iteracija za računanje vlastitih vektora matrice (vidjeti [9, 5]).

3.5. Osnove analize grešaka zaokruživanja

U ovom dijelu, pozabavit ćemo se osnovnim alatom za analiziranje stabilnosti numeričkih algoritama. Analiza grešaka zaokruživanja je zajedno sa perturbacijskom analizom problema koji se rješava, moćan alat za analiziranje pa zato i dizajniranje numeričkih algoritama.

Označimpo sa $\mathtt{P}$ skup negativnih potencija broja 2 koje ulaze u definiciju preciznosti aritmetike računala po IEEE standardu:

$$\mathtt{P} = \{2^{-23}, 2^{-24}, 2^{-52}, 2^{-53}, 2^{-63}\}. \tag{3.5.1}$$

Ako je n cijeli broj koji ima ulogu dimenzije vektora ili matrice, stupnja polinoma, broja sumanada u konačnoj sumi, ili broja faktora u konačnom produktu i sl., tada ćemo pretpostaviti da vrijedi

$$u \in \mathtt{P} \quad \Longrightarrow \quad nu \leq 2^{-6}. \tag{3.5.2}$$

To znači, ako radimo u jednostrukoj preciznosti i koristimo bilo koji način zaokruživanja, maksimalna vrijednost od n će biti

$$2^{17} = 131072.$$

Ako koristimo standardni način zaokruživanja do najbližeg, dozvoljavamo

$$n \leq 262144.$$

Ako želimo raditi s još većim dimenzijama, pretpostavka je da ćemo koristiti dvostruku preciznost (IEEE dvostruki format). U tom slučaju, za n dozvoljavamo gornju granicu $2^{46} \approx 7.037 \cdot 10^{13}$, a ako još koristimo standardni način zaokruživanja, n najviše može biti $1.40737488355328 \cdot 10^{14}$. Ako želimo još veći n, moramo osigurati korištenje računanja u proširenoj točnosti. Konačno, ako u razmatranje želimo uključiti i kalkulatore, tada u skup $\mathtt{P}$ možemo ubaciti i neke negativne potencije od 10 (npr. 10^{-12}), jer kalkulatori koriste decimalnu aritmetiku.

Sljedeća "tehnička" lema često će se koristiti u ocjenjivanju grešaka zaokruživanja.

Lema 3.5.1 *Neka $u \in \mathtt{P}$ i n zadovoljavaju uvjet (3.5.2). Ako je $|\varepsilon| \leq u$, tada vrijedi*

(i) $(1+\varepsilon)^2 = 1+\varepsilon_2$, $|\varepsilon_2| \leq 2.00000012u$,

(ii) $(1+\varepsilon)^3 = 1+\varepsilon_3$, $|\varepsilon_3| \leq 3.00000036u$,

(iii) $(1+\varepsilon)^{-1} = 1+\varepsilon_1'$, $|\varepsilon_1'| \leq 1.00000012u$,

(iv) $(1+\varepsilon)^{-2} = 1+\varepsilon_2'$, $|\varepsilon_2'| \leq 2.00000036u$,

(v) $(1+\varepsilon)^n = 1+\varepsilon_n$, $|\varepsilon_n| \leq 1.008nu$,

(vi) $(1+\varepsilon)^{-n} = 1+\varepsilon_n'$, $|\varepsilon_n'| \leq 1.008nu$,

(vii) $(1+\varepsilon)^{\frac{1}{2}} = 1+\varepsilon_{\sqrt{\ }}$, $|\varepsilon_{\sqrt{\ }}| \leq 0.500000015\,|\varepsilon| \leq 0.500000015\,u$.

Dokaz.

(i) Zbog $(1+\varepsilon)^2 = 1+\varepsilon(2+\varepsilon)$ je

$$|\varepsilon_2| = |2+\varepsilon| \cdot |\varepsilon| \leq (2+2^{-23})u \leq 2.00000012u.$$

(ii) Slično, $(1+\varepsilon)^3 = 1+\varepsilon(3+3\varepsilon+\varepsilon^2)$, pa je

$$|\varepsilon_3| \leq (3+3\cdot 2^{-23}+2^{-46})u \leq 3.00000036u.$$

(iii) Zbog $|\varepsilon| < 1$, imamo

$$\left|\frac{1}{1+\varepsilon}\right| \leq \frac{1}{1-|\varepsilon|} = 1+\frac{|\varepsilon|}{1-|\varepsilon|} \leq 1+(1-2^{-23})^{-1}u \leq 1.00000012u.$$

(iv) Korištenjem Taylorovog razvoja

$$(1+t)^{-2} = 1+2t+3t^2+\cdots kt^{k-1}+\cdots,$$

odmah slijedi

$$(1+\varepsilon)^{-2} = 1+\varepsilon(2+3\varepsilon+4\varepsilon^2+\cdots).$$

Još treba iskoristiti da je

$$2+3|\varepsilon|+4|\varepsilon|^2+5|\varepsilon|^3+\cdots \leq 2.00000035762793.$$

(v) Razvoj binoma na n-tu potenciju daje

$$\begin{aligned}
|\varepsilon_n| &= n\left|\varepsilon\left(1+\frac{n-1}{2}\,\varepsilon+\frac{(n-1)(n-2)}{2\cdot 3}\,\varepsilon^2+\cdots+\varepsilon^{n-2}+\frac{1}{n}\,\varepsilon^{n-1}\right)\right| \\
&\leq nu\left(1+\frac{2^{-6}}{2}\left(1+\frac{n-2}{3}\,u+\left[\frac{n-2}{3}\,u\right]^2+\left[\frac{n-2}{3}\,u\right]^3+\cdots\right)\right) \\
&\leq nu\left(1+\frac{2^{-7}}{1-\frac{(n-2)u}{3}}\right) \leq nu\left(1+\frac{2^{-7}}{1-\frac{2^{-6}}{3}}\right) = \frac{385}{382}\,nu \leq 1.008nu.
\end{aligned}$$

(vi) Dokaz je posve analogan prethodnom. Kako je, zbog (iii),

$$(1+\varepsilon)^{-n} = (1+\varepsilon')^n = 1+\varepsilon'_n,$$

imamo

$$\begin{aligned}|\varepsilon'_n| &= n\left|\varepsilon'\left(1+\frac{n-1}{2}\,\varepsilon'+\frac{(n-1)(n-2)}{2\cdot 3}\,\varepsilon'^2+\cdots+\varepsilon'^{n-2}+\frac{1}{n}\varepsilon'^{n-1}\right)\right| \\ &\le n\alpha u\left(1+\frac{2^{-6}\alpha}{2}\left(1+\frac{n-2}{3}\,\alpha u+\left[\frac{n-2}{3}\,\alpha u\right]^2+\cdots\right)\right) \\ &\le n\alpha u\left(1+\frac{2^{-7}\alpha}{1-\frac{(n-2)\,\alpha u}{3}}\right),\end{aligned}$$

gdje je $\alpha = 1.00000012$. Kako je

$$\alpha\left(1+\frac{2^{-7}\alpha}{1-\frac{2^{-6}\alpha}{3}}\right) < 1.007854,$$

dokaz je gotov.

(vii) Ako su η i $\eta_{\sqrt{\ }}$ povezani relacijom

$$\sqrt{1+\eta} = 1+\eta_{\sqrt{\ }},$$

tada je

$$\eta_{\sqrt{\ }} = \sqrt{1+\eta}-1 = \frac{\eta}{\sqrt{1+\eta}+1},$$

pa je

$$\begin{aligned}|\eta_{\sqrt{\ }}| &\le \frac{|\eta|}{\sqrt{1-|\eta|}+1} \le \frac{|\eta|}{\sqrt{1-u}+1} \le \frac{|\eta|}{\sqrt{1-2^{-23}}+1} \\ &\le 0.500000015|\eta| \le 0.500000015u. \qquad (3.5.3)\end{aligned}$$

Time je dokaz lemme 3.5.1 završen. ■

Za prvi dojam o korisnosti leme 3.5.1, riješimo sljedeći problem.

Primjer 3.5.1 *Neka su x i y reprezentabilni u računalu, tako da vrijedi $x = fl(x)$ i $y = fl(y)$. S kolikom relativnom greškom će računalo koje koristi IEEE standard izračunati $z = \sqrt{x^2+y^2}$?*

Da bismo riješili problem, pretpostavimo prvo da je

$$fl(x^2)+fl(y^2) < N_{\max} \quad i \quad \min\{|x|,|y|\} \ge \sqrt{N_{\min}}.$$

Tada, možemo pisati

$$x_2 = x \otimes x = fl(x^2) = x^2(1+\varepsilon_1), \quad |\varepsilon_1| \le u$$
$$y_2 = y \otimes y = fl(y^2) = y^2(1+\varepsilon_2), \quad |\varepsilon_2| \le u.$$

Umjesto

$$fl(fl(x^2) + fl(y^2)) = fl(x^2) \oplus fl(y^2),$$

kraće pišemo $fl(x^2+y^2)$. *Sada imamo*

$$z_2 = fl(x_2 + y_2) = (x_2 + y_2)(1+\varepsilon_3), \quad |\varepsilon_3| \le u$$
$$z = fl(\sqrt{z_2}) = \sqrt{z_2}(1+\varepsilon_4), \quad |\varepsilon_4| \le u.$$

Povežimo sve te jednadžbe:

$$\begin{aligned} z &= (1+\varepsilon_4)\sqrt{z_2} = (1+\varepsilon_4)\sqrt{(x_2+y_2)(1+\varepsilon_3)} \\ &= (1+\varepsilon_4)\sqrt{1+\varepsilon_3}\sqrt{x^2(1+\varepsilon_1)+y^2(1+\varepsilon_2)} \\ &= (1+\varepsilon_4)\sqrt{1+\varepsilon_3}\sqrt{x^2+y^2}\sqrt{1+\frac{x^2\varepsilon_1+y^2\varepsilon_2}{x^2+y^2}} \\ &= (1+\varepsilon_4)\sqrt{1+\varepsilon_3}\sqrt{1+\varepsilon_5}\sqrt{x^2+y^2}, \\ &= \sqrt{x^2+y^2}(1+\varepsilon_z), \end{aligned}$$

pri čemu je

$$\varepsilon_5 = \frac{x^2\varepsilon_1 + y^2\varepsilon_2}{x^2+y^2}, \quad 1+\varepsilon_z = (1+\varepsilon_4)\sqrt{(1+\varepsilon_3)(1+\varepsilon_5)}.$$

Ocijenimo prvo ε_5. *Kako je*

$$\frac{x^2\varepsilon_1 + y^2\varepsilon_2}{x^2+y^2} = \frac{x^2}{x^2+y^2}\,\varepsilon_1 + \frac{y^2}{x^2+y^2}\,\varepsilon_2,$$

ε_5 *je konveksna suma*[1] *brojeva* ε_1 *i* ε_2, *pa se nalazi u zatvorenom intervalu* $[\varepsilon_1, \varepsilon_2]$ *(ili* $[\varepsilon_2, \varepsilon_1]$ *ako je* $\varepsilon_2 < \varepsilon_1$*). To opet znači da vrijedi*

$$|\varepsilon_5| \le \max\{|\varepsilon_1|, |\varepsilon_2|\} \le u.$$

Sada imamo,

$$\begin{aligned} |\varepsilon_z| &= |(1+\varepsilon_4)\sqrt{(1+\varepsilon_3)(1+\varepsilon_5)} - 1| \le (1+u)\sqrt{(1+u)(1+u)} - 1 \\ &\le (1+u)(1+u) - 1 \le (2+u)u \le 2.00000012u. \end{aligned}$$

[1] suma $\alpha_1x_1 + \alpha_2x_2 + \cdots + \alpha_kx_k$ je konveksna ako vrijedi $\sum_i \alpha_i = 1$, te za svako i, $0 \le \alpha_i \le 1$.

Kod korištenja IEEE aritmetike, slučaj prekoračenja će biti dojavljen[2]*. Kod najnovijih prevodioca (npr. za programski jezik FORTRAN 90), postoji mogućnost da programer ugradi u kôd programa grananje koje u slučaju prekoračenja nastavlja s računanjem po formuli (vidi primjer 3.3.3)*

$$\mu = \max\{|x|, |y|\}, \quad \nu = \min\{|x|, |y|\}, \quad z = f\ell\left(\mu\sqrt{1+\left(\frac{\nu}{\mu}\right)^2}\right).$$

Ova formula je sigurna jer nam podaci $x = f\ell(x)$ i $y = f\ell(y)$ ukazuju da su μ i ν reprezentabilni brojevi. U ovom slučaju je prekoračenje moguće tek ako je

$$\mu \leq N_{\max} < \mu\sqrt{1+\left(\frac{\nu}{\mu}\right)^2} \leq \sqrt{2}\mu.$$

Pretpostavimo da to nije slučaj kao i da neće doći do potkoračenja međurezultata (o tome ćemo kasnije). Načinimo uz te pretpostavke analizu grešaka zaokruživanja za taj drugi algoritam.

Da bi analiza bila jasnija uvest ćemo niz pomoćnih varijabli z_i čije vrijednosti su reprezentabilni brojevi.

$$\begin{aligned}
z_1 &= f\ell\left(\frac{\nu}{\mu}\right) = \left(\frac{\nu}{\mu}\right)(1+\varepsilon_1), \quad |\varepsilon_1| \leq u,\\
z_2 &= f\ell(z_1^2) = (z_1^2)(1+\varepsilon_2), \quad |\varepsilon_2| \leq u,\\
z_3 &= f\ell(1+z_2) = (1+z_2)(1+\varepsilon_3), \quad |\varepsilon_3| \leq u,\\
z_4 &= f\ell(\sqrt{z_3}) = \sqrt{z_3}(1+\varepsilon_4), \quad |\varepsilon_4| \leq u,\\
z_5 &= f\ell(\mu \cdot z_4) = (\mu \cdot z_4)(1+\varepsilon_5), \quad |\varepsilon_5| \leq u.
\end{aligned}$$

Postupak možemo nastaviti kao i prije počevši "od kraja":

$$z_5 = (\mu \cdot z_4)(1+\varepsilon_5) = \mu \cdot \sqrt{(1+z_2)(1+\varepsilon_3)}(1+\varepsilon_4)(1+\varepsilon_5) = \cdots$$

Pokušajte završiti taj postupak.

Da bismo pokazali druge mogućnosti, ocijenimo, krenuvši "od početka", relativne greške koje sadrže svi međurezultati. Zamislimo da se koristila egzaktna (još kažemo i beskonačna) aritmetika. Tada bismo umjesto vrijednosti z_i dobili neke druge vrijednosti, nazovimo ih sa w_i, $1 \leq i \leq 5$. Ako pišemo

$$z_i = w_i(1+\eta_i), \quad 1 \leq i \leq 5,$$

tada trebamo ocijeniti η_5. Do ocjene za $|\eta_5|$ možemo doći ocjenjujući greške η_1, η_2, η_3 i η_4 u redosljedu kako su napisane. S obzirom da je $\eta_1 = \varepsilon_1$, imamo $|\eta_1| \leq u$. Nadalje, imamo

$$w_2(1+\eta_2) = [w_1(1+\eta_1)]^2(1+\varepsilon_2) = w_1^2(1+\eta_1)^2(1+\varepsilon_2),$$

[2]Najčešće tako da se program prekida nakon javljanja greške.

a kako je $w_2 = w_1^2$, *vrijedi*

$$\begin{aligned}|\eta_2| &= |(1+\eta_1)^2(1+\varepsilon_2) - 1| = |\varepsilon_2 + 2\eta_1 + \eta_1^2 + \varepsilon_2(2\eta_1 + \eta_1^2)| \\ &\le u + 2u + u^2 + u(2u + u^2) = 3u + 3u^2 + u^3 = (3 + 3u + u^2)u \\ &\le 3.00000036u.\end{aligned}$$

Sljedeća jednadžba

$$z_3 = (1+z_2)(1+\varepsilon_3) = [1 + w_2(1+\eta_2)](1+\varepsilon_3) = (1+w_2)\Big(1 + \frac{w_2}{1+w_2}\,\eta_2\Big)(1+\varepsilon_3),$$

zbog $w_2 \ge 0$, *daje*

$$\begin{aligned}|\eta_3| &= \Big|\frac{w_2}{1+w_2}\,\eta_2 + \varepsilon_3 + \frac{w_2}{1+w_2}\,\eta_2\varepsilon_3\Big| \le |\varepsilon_3| + |\eta_2| + |\varepsilon_3\eta_2| \\ &\le [1 + 3.00000036(1+u)]u \le 4.0000012u.\end{aligned}$$

Pišući

$$z_4 = \sqrt{z_3}(1+\varepsilon_4) = \sqrt{w_3(1+\eta_3)}(1+\varepsilon_4) = w_4\sqrt{1+\eta_3}(1+\varepsilon_4),$$

i koristeći relaciju (3.5.3), dobivamo

$$1 + \eta_4 = \sqrt{1+\eta_3}(1-\varepsilon_4),$$

pa je

$$\begin{aligned}|\eta_4| &\le |\sqrt{1+\eta_3}(1+\varepsilon_4) - 1| \\ &\le \Big(1 + \frac{4.00000072u}{1 + \sqrt{1 - 4.00000072 \cdot 2^{-23}}}\Big)(1+u) - 1 \le 3.000000837u.\end{aligned}$$

Konačno,

$$1 + \eta_5 = (1+\eta_4)(1+\varepsilon_5),$$

pa je

$$|\eta_5| \le |\eta_4| + |\varepsilon_5| + |\eta_4\varepsilon_5| \le 4.0000012u.$$

Korištenjem sofisticiranije formule, koja zahtijeva više instrukcija, dobivamo i rezultat koji je generalno nešto netočniji (iako se još uvijek radi o grešci u zadnjoj sigurnoj decimali kad se egzaktni rezultat smjesti u računalo). Međutim, dobitak je u proširenju domene funkcije koja paru reprezentabilnih brojeva (x, y) *pridružuje* $f\ell(\sqrt{x^2+y^2})$. *To je dosta važno kako bi se izbjeglo prekoračenje i prekid rada računala ili moguća relativna netočnost ako je rezultat mali broj.*

Ostalo je još promotriti što se događa ako su x *ili* y *takvi da ili* x^2 *ili* y^2 *potkoračuje, ako* z *postaje nula, ili ako* $\sqrt{x^2+y^2}$ *padne u područje subnormalnih brojeva. Ako samo* x^2 (y^2) *potkoračuje u nulu ili subnormalni broj, rezultat će imati*

malu relativnu grešku, ne veću od nekoliko u. Ako izračunati $f\ell(x^2+y^2)$ padne u područje subnormalnih brojeva, tada će $f\ell(x^2+y^2)$ možda nositi veliku relativnu grešku. Vađenje korijena će tu grešku približno prepoloviti, ali će ona i dalje ostati velika. Potkoračenje obaju brojeva u nulu će dati maksimalnu relativnu grešku -1. Dakle, kad prijeti potkoračenje ili postupno potkoračenje, direktna formula neće dati točan rezultat. Što će dati druga kompliciranija formula?

Ako je $|\mu| \geq N_{\min}$, tada će se $f\ell(\mu\sqrt{1+(\nu/\mu)^2})$ izračunati s malom relativnom greškom (kako je gore izračunato) bez obzira na to je li ν subnormalan broj ili ne. Ako je pak μ subnormalan broj, tada su oba polazna broja, također, subnormalna i kod smještanja brojeva u računalo došlo je do gubljenja relativne točnosti. Tada će i egzaktni kvocijent ν/μ i izračunati kvocijent $f\ell(\nu/\mu)$ imati veću (ili veliku) relativnu grešku pa će isto vrijediti i za z. Jedini lijek je da se ulazni podaci x i y, prije učitavanja, skaliraju potencijom od 2 tako da μ ne bude subnormalni broj. Nakon računanja, z treba skalirati inverznom (negativnom) potencijom broja 2.

U zaključku možemo reći da kompliciraniju formulu treba koristiti ako u programskom jeziku nije implementirano signaliziranje da je došlo do izuzetne situacije. Ako je signaliziranje implementirano, tada je najbolje koristiti jednostavniji algoritam, uz skretnicu koja u slučaju izuzetne situacije vodi kontrolu na kompliciraniji algoritam.

3.5.1. Propagiranje grešaka zaokruživanja

Promotrimo prvo kako jedna računska operacija na računalu povećava postojeće greške u podacima. Pretpostavit ćemo da su

$$\hat{x} = x(1+\varepsilon_x) \quad \text{i} \quad \hat{y} = y(1+\varepsilon_y)$$

podaci spremljeni u računalo. Oni aproksimiraju točne podatke x i y s relativnim greškama ε_x i ε_y, respektivno.

Za operaciju množenja vrijedi

$$\begin{aligned} f\ell(\hat{x}\cdot\hat{y}) &= (\hat{x}\cdot\hat{y})(1+\varepsilon_\times) = xy(1+\varepsilon_x)(1+\varepsilon_y)(1+\varepsilon_\times) \\ &= xy(1+\varepsilon_x+\varepsilon_y+\varepsilon_x\varepsilon_y+\varepsilon_\times+\alpha). \end{aligned}$$

Pritom je $|\varepsilon_\times| \leq u$ i

$$|\alpha| = |(\varepsilon_x+\varepsilon_y+\varepsilon_x\varepsilon_y)\varepsilon_\times| \approx |(\varepsilon_x+\varepsilon_y)\varepsilon_\times| \leq u(|\varepsilon_x|+|\varepsilon_x|)$$

po pretpostavci manje od u pa nas dalje ne zanima. Ako su $|\varepsilon_x|$ i $|\varepsilon_y|$ tako mali, da je i $|\varepsilon_x\varepsilon_y|$ manje od u i taj član možemo zanemariti. Zaključujemo da se relativna greška kod množenja propagira tako da se **zbroju relativnih grešaka faktora**

pridoda greška nastala množenjem. Tek ako su ε_x i ε_y približno istih modula i suprotnih predznaka, može $\varepsilon_x\varepsilon_y$ utjecati na ukupnu grešku.

Za dijeljenje imamo

$$f\ell\left(\frac{\hat{x}}{\hat{y}}\right) = \frac{\hat{x}}{\hat{y}}(1+\varepsilon_/) = \frac{x}{y}\,\frac{1+\varepsilon_x}{1+\varepsilon_y}(1+\varepsilon_/) = \frac{x}{y}\,(1+\varepsilon_x)\Big(1-\varepsilon_y+\frac{\varepsilon_y^2}{1+\varepsilon_y}\Big)(1+\varepsilon_/)$$
$$= \frac{x}{y}\left(1+\varepsilon_x-\varepsilon_y-\varepsilon_x\varepsilon_y+\frac{\varepsilon_y^2}{1+\varepsilon_y}+\varepsilon_/+\beta\right),$$

pritom je $|\varepsilon_/| \le u$. Veličinu

$$\beta = \Big(\varepsilon_x-\varepsilon_y-\varepsilon_x\varepsilon_y+(1+\varepsilon_x)\frac{\varepsilon_y^2}{1+\varepsilon_y}\Big)\varepsilon_/ + \frac{\varepsilon_x\varepsilon_y^2}{1+\varepsilon_y}$$

ćemo zanemariti, smatrajući da je manja od osnovne greške zaokruživanja u. Vidimo da je relativna greška dominirana članom $\varepsilon_x-\varepsilon_y-\varepsilon_x\varepsilon_y+\dfrac{\varepsilon_y^2}{1+\varepsilon_y}$, pri čemu, ako ε_x i ε_y nisu skoro jednaki, možemo zanemariti članove višeg reda $\varepsilon_x\varepsilon_y$ i $\dfrac{\varepsilon_y^2}{1+\varepsilon_y}$. Kao i kod množenja, nova greška $\varepsilon_/$ ima utjecaj tek na zadnju decimalu binarnog (pogotovo decimalnog) prikaza kvocijenta.

Kod aditivnih operacija imamo

$$f\ell(\hat{x}\pm\hat{y}) = (\hat{x}\pm\hat{y})(1+\varepsilon_\pm) = (x(1+\varepsilon_x)\pm y(1+\varepsilon_y))(1+\varepsilon_\pm)$$
$$= x\pm y + x\varepsilon_x \pm y\varepsilon_y + x\varepsilon_\pm \pm y\varepsilon_\pm + x\varepsilon_x\varepsilon_\pm \pm y\varepsilon_y\varepsilon_\pm$$
$$= (x\pm y)\Big(1+\frac{x}{x\pm y}\,(\varepsilon_x+\varepsilon_\pm+\varepsilon_x\varepsilon_\pm)\pm\frac{y}{x\pm y}\,(\varepsilon_y+\varepsilon_\pm+\varepsilon_y\varepsilon_\pm)\Big).$$

Ako pišemo

$$f\ell(x\pm y) = (x\pm y)(1+\varepsilon_s),$$

tada je ε_s relativna greška u $f\ell(x\pm y)$ kao aproksimaciji za $x\pm y$.

Neka su x i y takvi da je

$$x\pm y = \operatorname{sign}(x)(|x|+|y|).$$

To će vrijediti ako je

$$\pm = + \quad \text{i} \quad \operatorname{sign}(x) = \operatorname{sign}(y),$$

ili ako je

$$\pm = - \quad \text{i} \quad \operatorname{sign}(x) = -\operatorname{sign}(y).$$

Tada je

$$\begin{aligned}\varepsilon_s &= \frac{x}{x \pm y}(\varepsilon_x + \varepsilon_\pm + \varepsilon_x\varepsilon_\pm) \pm \frac{y}{x \pm y}(\varepsilon_y + \varepsilon_\pm + \varepsilon_y\varepsilon_\pm) \\ &= \frac{|x|}{|x| + |y|}(\varepsilon_x + \varepsilon_\pm + \varepsilon_x\varepsilon_\pm) + \frac{|y|}{|x| + |y|}(\varepsilon_y + \varepsilon_\pm + \varepsilon_y\varepsilon_\pm) \qquad (3.5.4)\end{aligned}$$

konveksna suma realnih brojeva $\varepsilon_x + \varepsilon_\pm + \varepsilon_x\varepsilon_\pm$ i $\varepsilon_y + \varepsilon_\pm + \varepsilon_y\varepsilon_\pm$, pa je ε_s između tih brojeva. Stoga je

$$\begin{aligned}|\varepsilon_s| &\le \max\{|\varepsilon_x + \varepsilon_\pm + \varepsilon_x\varepsilon_\pm|, |\varepsilon_y + \varepsilon_\pm + \varepsilon_y\varepsilon_\pm|\} \\ &\le \max\{|\varepsilon_x|, |\varepsilon_y|\} + |\varepsilon_\pm|(1 + \max\{|\varepsilon_x|, |\varepsilon_y|\}) \\ &\approx \max\{|\varepsilon_x|, |\varepsilon_y|\} + u, \qquad (3.5.5)\end{aligned}$$

pa na relativnu grešku rezultata utječu obje "donesene" greške ε_x i ε_y. Finije razmatranje koristit će relaciju (3.5.4) koja pokazuje da se ε_x množi s $|x|/(|x| + |y|)$ dok se ε_y množi s $|y|/(|x| + |y|)$. Ako je npr. $|x| \gg |y|$ i $|\varepsilon_x| \ll |\varepsilon_y|$, tada će $|\varepsilon_s|$ biti blizu manje greške $|\varepsilon_x|$.

Kod svih dosadašnjih slučajeva relativna greška u rezultatu nije bitno veća od relativnih grešaka u polaznim podacima. Ako polazimo od točnih polaznih podataka, potreban je ogroman broj operacija tih vrsta da bi greška u "zadnjem međurezultatu" bitnije narasla. Međutim još nismo razmatrali preostali slučaj,

$$x \pm y = \operatorname{sign}(x)(|x| - |y|),$$

koji će vrijediti ako je

$$\pm = + \quad \text{i} \quad \operatorname{sign}(x) = -\operatorname{sign}(y),$$

ili ako je

$$\pm = - \quad \text{i} \quad \operatorname{sign}(x) = \operatorname{sign}(y).$$

Umjesto relacije (3.5.4), imat ćemo

$$\varepsilon_s = \frac{|x|}{|x| - |y|}(\varepsilon_x + \varepsilon_\pm + \varepsilon_x\varepsilon_\pm) - \frac{|y|}{|x| - |y|}(\varepsilon_y + \varepsilon_\pm + \varepsilon_y\varepsilon_\pm). \qquad (3.5.6)$$

Ako je

$$|x| \approx |y|$$

greške $\varepsilon_x + \varepsilon_\pm + \varepsilon_x\varepsilon_\pm$ i $\varepsilon_y + \varepsilon_\pm + \varepsilon_y\varepsilon_\pm$ množit će se s potencijalno vrlo velikim brojevima

$$b_1 = \frac{|x|}{|x| - |y|} \quad \text{odnosno} \quad b_2 = -\frac{|y|}{|x| - |y|}.$$

Taj fenomen smo već upoznali kao opasno/katastrofalno "kraćenje". Ako su b_1 i b_2 brojevi reda veličine jedan, onda neće doći do opasnog kraćenja.

U zaključku možemo reći da su operacije množenja i dijeljenja, te zbrajanja sumanada istog predznaka "dobre" operacije koje neće bitnije pogoršati relativnu pogrešku operanada, dok za operaciju oduzimanja brojeva istog predznaka to vrijedi tek ako nije došlo do kraćenja. Ako je došlo do kraćenja, tada vrijedi pravilo: **što jače kraćenje u operandima, to je veća relativna greška u rezultatu**.

Koji puta je kraćenje neizbježno zbog naravi problema (loše uvjetovani problemi), pa možemo tek birati između algoriatma koji će dati katastrofalno kraćenje u jednoj ili nekoliko operacija, ili algoritama koji će postupno, jedva primjetno, kratiti međurezultate kroz mnogo aditivnih operacija.

Promotrimo sada **akumulaciju** grešaka kod računanja produkta i sume n brojeva, kao i kod osnovnih vektorskih i matričnih operacija.

3.5.2. Stabilnost produkta od n brojeva

Za produkt brojeva vrijedi

Lema 3.5.2 *Neka je* $x_i = fl(x_i)$, $1 \le i \le n$ *i neka je*

$$|\, fl(x_1 \cdot x_2 \cdots x_i)| \le N_{\max}, \quad 1 \le i \le n.$$

Tada je

$$fl(x_1 \cdots x_n) = (x_1 \cdots x_n)(1+\varepsilon), \quad |\varepsilon| \le 1.008(n-1)u.$$

Dokaz. Ako definiramo pomoćne varijable z_i, sljedeći iteracije algoritma za množenje brojeva $x_1 x_2 \cdots x_n$ u redoslijedu kako je napisano, korištenjem relacije (3.3.4), imat ćemo

$$\begin{aligned}
z_1 &= x_1 \\
z_2 &= fl(z_1 \cdot x_2) = (z_1 \cdot x_2)(1+\varepsilon_2) \\
z_3 &= fl(z_2 \cdot x_3) = (z_2 \cdot x_3)(1+\varepsilon_3) \\
&\vdots \qquad\qquad \vdots \\
z_{n-1} &= fl(z_{n-2} x_{n-1}) = (z_{n-2} x_{n-1})(1+\varepsilon_{n-1}) \\
z_n &= fl(z_{n-1} x_n) = (z_{n-1} x_n)(1+\varepsilon_n).
\end{aligned}$$

Množeći lijeve i desne strane svih jednakosti, dobit ćemo, nakon kraćenja pomoćnih varijabli,

$$z_n = x_1 \cdots x_n (1+\varepsilon_2)(1+\varepsilon_3)\cdots(1+\varepsilon_n).$$

Definiramo li

$$1+\varepsilon = (1+\varepsilon_2)(1+\varepsilon_3)\cdots(1+\varepsilon_n), \tag{3.5.7}$$

dobivamo

$$(1-|\varepsilon_2|)(1-|\varepsilon_3|)\cdots(1-|\varepsilon_n|) - 1 \le \varepsilon \le (1+|\varepsilon_2|)(1+|\varepsilon_3|)\cdots(1+|\varepsilon_n|) - 1.$$

Kako je

$$1-(1-|\varepsilon_2|)(1-|\varepsilon_3|)\cdots(1-|\varepsilon_n|)\leq(1+|\varepsilon_2|)(1+|\varepsilon_3|)\cdots(1+|\varepsilon_n|)-1,$$

zaključujemo da je

$$|\varepsilon|\leq(1+u)^{n-1}-1\leq 1.008(n-1)u.$$

Zadnja nejednakost slijedi iz leme 3.5.1(v) uzimanjem $\varepsilon=u$. ■

Iz dokaza slijedi da ocjena vrijedi za proizvoljan poredak faktora (svaki algoritam koji u nekom poretku množi faktore) ako svi parcijalni produkti ne prekoračuju (tj. za pomoćne varijable vrijedi $z_k\leq N_{\max}$).

Ovaj rezultat pokazuje da množenje više faktora sporo akumulira relativnu grešku u rezultatu. Ako se koristi način zaokruživanja prema najbližem, greške ε_i iz dokaza će biti i pozitivne i negativne, pa će ukupna greška

$$(1+\varepsilon_2)(1+\varepsilon_3)\cdots(1+\varepsilon_n)-1\approx\varepsilon_2+\varepsilon_3+\cdots+\varepsilon_n$$

biti još manja. Npr. umnožak milijun brojeva računan u dvostrukoj točnosti, imat će barem 10 točnih značajnih znamenaka, a uz zaokruživanje prema najbližem može se očekivati barem 13 točnih značajnih znamenaka (to će biti jasnije nakon odjeljka o sumiranju).

Formalno, lema 3.5.2 pokazuje da je svaki algoritam koji uzastopce množi brojeve stabilan unaprijed. Iz dokaza leme vidimo da rezultat možemo zapisati u obliku

$$f\ell(x_1\cdots x_n)=x_1[x_2(1+\varepsilon_2)][x_3(1+\varepsilon_3)]\cdots[x_n(1+\varepsilon_n)],\quad|\varepsilon_i|\leq u,\quad 2\leq i\leq n.$$

Ako interpretiramo faktore $x_i(1+\varepsilon_i)$ kao malo perturbirane polazne podatke x_i, vidimo da je algoritam množenja n brojeva savršeno stabilan unazad. Izračunati produkt je jednako točan kao kad polazne brojeve koji nisu BPZ učitamo u računalo, a zatim ih egzaktno pomnožimo!

Iz relacije (3.5.7) slijedi

$$\begin{aligned}\varepsilon&=(1+\varepsilon_2)(1+\varepsilon_3)\cdots(1+\varepsilon_n)-1\\&=\varepsilon_2+\varepsilon_3+\cdots+\varepsilon_n+\sum_{2\leq i<k\leq n}\varepsilon_i\varepsilon_k+\cdots+\varepsilon_2\varepsilon_3\cdots\varepsilon_n\\&=\varepsilon_2+\varepsilon_3+\cdots+\varepsilon_n+\mathcal{O}(u^2),\end{aligned}$$

pa se iskaz leme još zapisuje u obliku

$$f\ell(x_1\cdots x_n)=(x_1\cdots x_n)(1+\varepsilon),\quad|\varepsilon|\leq(n-1)u+\mathcal{O}(u^2).\tag{3.5.8}$$

Prednost takvog zapisa je u tome što daje najmanju moguću konstantu uz linearni član u. Međutim, ako se taj rezultat koristi kao dio analize nekog složenijeg izraza ili

algoritma, postoje dvije neugodnosti. Prvo, ako u nejednakosti postoji član oblika $\mathcal{O}(u^2)$, više nemamo čistu ocjenu već tek ocjenu "do na veličinu reda u^2". Drugo, ako se u kasnijoj analizi član $\mathcal{O}(u^2)$ množi s potencijalno velikim brojevima, više nismo sigurni je li postao $\mathcal{O}(u)$ ili možda čak $\mathcal{O}(1)$, pa je svrha analize narušena. Stoga se oblik (3.5.8) uglavnom koristi kod jednostavnijih izraza u svrhu dobivanja što povoljnije konstante uz u.

Jedan koristan pomoćni rezultat

U formuli (3.5.7) javlja se produkt faktora koji su oblika $1+\varepsilon_k$. Za ocjenjivanje izraza koji sadrže samo množenja i dijeljenja možemo koristiti sljedeću lemu.

Lema 3.5.3 *Neka je $u > 0$ realni broj i n cijeli pozitivni broj takav da vrijedi $nu < 1$. Neka su za $i = 1, \ldots, n$, δ_i realni brojevi, i $p_i \in \{-1, 1\}$. Ako su svi $|\delta_i| \leq u$, onda vrijedi*

$$\prod_{i=1}^{n}(1+\delta_i)^{p_i} = 1 + \theta_n, \tag{3.5.9}$$

uz ocjenu

$$|\theta_n| \leq \gamma_n := \frac{nu}{1-nu}. \tag{3.5.10}$$

Dokaz. Dokaz se provodi indukcijom po n. Za $n = 1$ vrijedi $u < 1$. Ako je $p_1 = 1$, onda je $\theta_1 = \delta_1$, pa je

$$|\theta_1| \leq u \leq \frac{u}{1-u}.$$

Ako je $p_1 = -1$, onda je

$$1 + \theta_1 = \frac{1}{1+\delta_1},$$

pa je

$$\theta_1 = \frac{1}{1+\delta_1} - 1 = -\frac{\delta_1}{1+\delta_1}.$$

S obzirom da je $1 + \delta_1 \geq 1 - u > 0$, dobivamo

$$|\theta_1| = \frac{|\delta_1|}{1+\delta_1} \leq \frac{u}{1-u},$$

čime je lema dokazana za $n = 1$.

Pretpostavimo da tvrdnja leme vrijedi za neko $n \geq 1$. To znači da je $nu < 1$, pa je $1 + \delta_i > 0$ za sve $1 \leq i \leq n$. Prema tome, produkt na lijevoj strani relacije (3.5.9) je pozitivan, pa je θ_n dobro definiran i zadovoljava ocjenu (3.5.10).

Pokažimo da tvrdnja leme vrijedi za $n + 1$. Iz pretpostavke

$$(n+1)u < 1$$

i $|\delta_i| \leq u$, $1 \leq i \leq n+1$ slijedi $nu < 1$ i $|\delta_i| \leq u$, $1 \leq i \leq n$, pa možemo koristiti relaciju (3.5.9) i ocjenu (3.5.10). Također, broj θ_{n+1} je dobro definiran relacijom

$$1 + \theta_{n+1} = \prod_{i=1}^{n+1} (1 + \delta_i)^{p_i},$$

pa je

$$1 + \theta_{n+1} = \prod_{i=1}^{n} (1 + \delta_i)^{p_i} \cdot (1 + \delta_{n+1})^{p_{n+1}} = (1 + \theta_n)(1 + \delta_{n+1})^{p_{n+1}}.$$

Za $p_{n+1} = 1$ dobivamo

$$\theta_{n+1} = \theta_n + \delta_{n+1} + \theta_n \delta_{n+1},$$

pa je

$$|\theta_{n+1}| \leq \frac{nu}{1 - nu} + u + \frac{nu^2}{1 - nu} = \frac{(n+1)u}{1 - nu} < \frac{(n+1)u}{1 - (n+1)u}.$$

Za $p_{n+1} = -1$ dobivamo

$$\theta_{n+1} = \frac{1 + \theta_n}{1 + \delta_{n+1}} - 1 = \frac{\theta_n - \delta_{n+1}}{1 + \delta_{n+1}},$$

pa je

$$|\theta_{n+1}| \leq \frac{|\theta_n| + |\delta_{n+1}|}{1 + \delta_{n+1}}.$$

Kako je $1 + \delta_{n+1} \geq 1 - u > 0$, lako slijedi

$$|\theta_{n+1}| \leq \frac{\dfrac{nu}{1 - nu} + u}{1 - u} = \frac{(n+1)u - nu^2}{1 - (n+1)u + nu^2} < \frac{(n+1)u}{1 - (n+1)u}.$$

Time je dokaz indukcijom završen. ■

Brojevi γ_n, $n \geq 1$ imaju vrlo praktično svojstvo,

$$\gamma_m + \gamma_n + \gamma_m \gamma_n \leq \gamma_{m+n}. \tag{3.5.11}$$

Zadatak 3.5.1

(i) Dokažite nejednakost (3.5.11).

(ii) Primijenite lemu 3.5.3 na relaciju (3.5.7). Je li dobivena konstanta uz u veća ili manja od 1.008?

Nakon analize stabilnosti produkta brojeva, proučimo stabilnost sume brojeva.

3.5.3. Stabilnost sume

Neka je

$$s_n = x_1 + x_2 + \cdots + x_n,$$

pri čemu za brojeve x_i vrijedi, kao i prije, $x_i = fl(x_i)$, $1 \leq i \leq n$. Suma s_n može se računati na razne načine. Promotrimo prvo najjednostavniji **rekurzivni algoritam**:

$s_1 := x_1$;
for $i := 2$ **to** n **do**
  $s_i := s_{i-1} + x_i$;

Označimo s $\hat{s}_i$ izračunatu vrijednost s_i. Tada, prema relaciji (3.3.4), vrijedi

$$\hat{s}_2 = fl(x_1 + x_2) = (x_1 + x_2)(1 + \varepsilon_1) = x_1(1 + \varepsilon_1) + x_2(1 + \varepsilon_1), \quad |\varepsilon_1| \leq u.$$

Slično,

$$\begin{aligned}\hat{s}_3 &= fl(\hat{s}_2 + x_3) = (\hat{s}_2 + x_3)(1 + \varepsilon_2) \\ &= x_1(1 + \varepsilon_1)(1 + \varepsilon_2) + x_2(1 + \varepsilon_1)(1 + \varepsilon_2) + x_3(1 + \varepsilon_2).\end{aligned}$$

Nastavljajući na taj način, dobivamo

$$\begin{aligned}\hat{s}_n &= fl(\hat{s}_{n-1} + x_n) = (\hat{s}_{n-1} + x_n)(1 + \varepsilon_{n-1}) \\ &= x_1(1 + \varepsilon_1)(1 + \varepsilon_2)\cdots(1 + \varepsilon_{n-1}) + x_2(1 + \varepsilon_1)(1 + \varepsilon_2)\cdots(1 + \varepsilon_{n-1}) \\ &\quad + x_3(1 + \varepsilon_2)\cdots(1 + \varepsilon_{n-1}) + \cdots + x_{n-1}(1 + \varepsilon_{n-2})(1 + \varepsilon_{n-1}) + x_n(1 + \varepsilon_{n-1}),\end{aligned}$$

gdje su svi $|\varepsilon_i| \leq u$. Zadnju relaciju možemo pojednostaviti uvođenjem novih varijabli

$$\begin{aligned}1 + \eta_1 &= (1 + \varepsilon_1)(1 + \varepsilon_2)\cdots(1 + \varepsilon_{n-1}) \\ 1 + \eta_2 &= (1 + \varepsilon_1)(1 + \varepsilon_2)\cdots(1 + \varepsilon_{n-1}) \\ 1 + \eta_3 &= (1 + \varepsilon_2)\cdots(1 + \varepsilon_{n-1}) \\ &\vdots \\ 1 + \eta_{n-1} &= (1 + \varepsilon_{n-2})(1 + \varepsilon_{n-1}) \\ 1 + \eta_n &= 1 + \varepsilon_{n-1}.\end{aligned}$$

Tada prethodnu relaciju možemo zapisati u obliku

$$\hat{s}_n = x_1(1 + \eta_1) + x_2(1 + \eta_2) + \cdots + x_n(1 + \eta_n), \tag{3.5.12}$$

pri čemu η_i možemo ocijeniti pomoću leme 3.5.3 ili još oštrije pomoću leme 3.5.1(v),

$$|\eta_i| \leq 1.008 \cdot \begin{cases}(n-1)u, & i = 1, \\ (n-i+1)u, & 2 \leq i \leq n.\end{cases} \tag{3.5.13}$$

Relacije (3.5.12) i (3.5.13) pokazuju da je rekurzivni algoritam za sumiranje brojeva povratno stabilan. Ocjene za relativnu grešku svakog perturbiranog polaznog podatka $x_i(1+\eta_i)$ su manje od nu. Je li algoritam, također, stabilan unaprijed?

Da bismo odgovorili na to pitanje oduzmimo od $\hat{s}_n$ egzaktnu sumu s_n,

$$\hat{s}_n - s_n = x_1\eta_1 + x_2\eta_2 + \cdots + x_n\eta_n$$

i ocjenjujemo apsolutnu vrijednost razlike

$$\begin{aligned} |\hat{s}_n - s_n| &\leq |x_1|\,|\eta_1| + |x_2|\,|\eta_2| + \cdots + |x_n|\,|\eta_n| \\ &\leq \max_i\{|\eta_i|\}(|x_1| + |x_2| + \cdots + |x_n|) \\ &\leq (|x_1| + |x_2| + \cdots + |x_n|)\cdot 1.008(n-1)u. \end{aligned} \tag{3.5.14}$$

Stoga je relativna greška od $\hat{s}_n$ kao aproksimacije za s_n, ocijenjena s

$$\frac{|\hat{s}_n - s_n|}{|s_n|} \leq \mathrm{cond}(s_n)1.008(n-1)u, \tag{3.5.15}$$

gdje je

$$\mathrm{cond}(s_n) = \frac{|x_1| + |x_2| + \cdots + |x_n|}{|x_1 + x_2 + \cdots + x_n|}. \tag{3.5.16}$$

Ako na trenutak pretpostavimo da je $\eta_i x_i$ proizvoljna perturbacija od x_i za svako i, onda vidimo da formula (3.5.14) daje oštru ocjenu jer se jednakost dostiže npr. za pozitivne x_i i $\eta_1 = \eta_2 = \cdots = \eta_n$. Nejednakost (3.5.15) pokazuje kako se ponaša relativna greška funkcije s_n pri relativnim greškama njenih argumenata x_i. Stoga je $\mathrm{cond}(s_n)$ broj uvjetovanosti za s_n i on, kao što znamo iz odjeljka o uvjetovanosti, povezuje grešku unaprijed s greškom unazad. Kako $\mathrm{cond}(s_n)$ može biti proizvoljno velik (uzmimo npr. brojeve $1, -1$ i t, pa pustite da $t \to 0$), algoritam nije općenito stabilan unaprijed. Ipak, **ako svi** x_i **imaju isti predznak**, onda je $\mathrm{cond}(s_n) = 1$, pa je i greška unaprijed mala, **a to znači da je algoritam stabilan unaprijed**. Isti zaključak vrijedi kadgod $\mathrm{cond}(s_n)$ nije veliki broj.

Umjesto da sumiranje vršimo redom od x_1 do x_n, možemo koristiti bilo koji (sekvencijalni) redoslijed sumiranja. Ocjene će nakon odgovarajuće analize grešaka zaokruživanja biti posve analogne. Pritom će vrijediti relacija (3.5.12) s tim da umjesto x_1 i x_2 stoje oni sumandi koji se prvi zbrajaju, umjesto x_3 onaj sumand koi se sljedeći dodaje sumi itd.

Katkad su svi članovi sume istog predznaka i pritom još uređeni u monotoni niz. Tada se postavlja pitanje u kojem redoslijedu treba sumirati da bi relativna greška u $\hat{s}_n$ bila što manja? Prvo uočimo da je traženje najmanje relativne greške za taj problem ekvivalentno traženju najmanje apsolutne greške (jer nju dijelimo sa s_n, a s_n ne ovisi o redoslijedu sumiranja). Dakle, tražimo redoslijed indeksa $j_1, j_2, \ldots, j_n$ za koji je

$$|x_{j_1}\eta_1 + x_{j_2}\eta_2 + \cdots + x_{j_n}\eta_n|$$

najmanje. Problem je težak jer greške ε_k koje definiraju svaki η_i mogu biti i pozitivne i negativne, pa $|\eta_1|$ može biti manji od bilo kojeg η_j, $3 \leq j \leq n$. Stoga moramo zahtjev malo oslabiti pitanjem **koji redoslijed indeksa** $j_1, j_2, \ldots, j_n$ **daje najbolju ocjenu** za $|x_{j_1}\eta_1 + \cdots + x_{j_n}\eta_n|$? Tako dolazimo do jednostavnog pravila.

Ako je niz $|x_1|, |x_2|, \ldots, |x_n|$ **monoton, sumiramo u smjeru od apsolutno najmanjeg do apsolutno najvećeg člana**.

Kaskadno sumiranje

Osim danog jednostavnog algoritma za sumiranje, spomenimo još jedan koji daje najmanju ocjenu greške, ali na računalu traži cijelo polje dodatnih lokacija u memoriji. Zovemo ga **sumiranje po parovima** ili **kaskadno sumiranje**.

Algoritam ima približno $\log_2(n)$ glavnih koraka, a približno tolika će biti i ograda za **svaki** $|\eta_i|$ iz relacije (3.5.12). Ako je $n = 2^k$, $k \geq 2$ algoritam ima k glavnih koraka. U prvom zbrajamo u parovima $z_i = x_{2i-1} + x_{2i}$, $1 \leq i \leq [n/2]$,

$$z_1 = x_1 + x_2, \quad z_2 = x_3 + x_4, \quad \ldots, \quad z_{m_1} = x_{n-1} + x_n, \quad m_1 = n/2 = 2^{k-1}.$$

U drugom ponavljamo istu operaciju nad $z_1, \ldots, z_{m_1}$,

$$w_1 = z_1 + z_2, \quad w_2 = z_3 + z_4, \quad \ldots, \quad w_{m_2} = z_{m_1-1} + z_{m_1}, \quad m_2 = m_1/2 = 2^{k-2}.$$

Nastavljajući na taj način, nakon $k-1$ glavnih koraka imamo $2^{(k-(k-1))} = 2$ člana, pa se u k-tom koraku oni zbroje dajući s_n.

Zadatak 3.5.2 *Ako je* $n = 2^k$*, dokažite da svaki* x_i *sudjeluje u točno* k *zbrajanja, pa je u relaciji (3.5.12) svaki* $|\eta_i|$ *omeđen sa* $1.008ku$*. Pokušajte napisati algoritam za opći slučaj kad* n *nije potencija od* 2*. Koliko prolaza ima glavna petlja kad je* n *jednako* $8, 16, 9, 10, 11, 12, 13, 14, 15$*? Generalizirajte zaključak za* n *između* 2^k *i* 2^{k+1}*. Koliko zbrajanja prolazi svaki* x_i*? Koliki je ukupni broj računskih operacija? Koliko dodatne memorije (dodatnih ćelija za pomoćne varijable) bi koristili za realizaciju algoritma, ako želite sačuvati vrijednosti polaznog niza u polaznom polju?*

Jedna generalnija analiza sumacijskih metoda i pripadnih grešaka zaokruživanja (vidi [5]) daje sljedeću generalnu uputu.

Kod odabiranja sumacijske metode s ciljem dobivanja što točnije sume s_n**, treba izabrati onu koja daje što manje apsolutne vrijednosti međurezultata**.

Zadatak 3.5.3 *Prije primjene zadnjeg algoritma možemo sortirati elemente niza* (x_i)*. Koju strategiju biste izabrali? Npr.*

$$y_1 = x_{\min} + x_{\max}$$

i kako dalje, ili

$$y_1 = x_{\min_1} + x_{\min_2},$$

gdje su $x_{\min_1}$ *i* $x_{\min_2}$ *dva najmanja broja, i kako dalje?*

Napomena 3.5.1 *Prednost standardnog algoritma prema "binarnom" je i u kraćem trajanju na računalu. Ne radi se o većem broju zbrajanja, nego o načinu dohvaćanje elemenata polja u računalo kod izvođenja aritmetičkih operacija s elementima polja. Prije zbrajanja operanada moderna računala zahvate cijeli blok podataka (onih u uzastopnim ćelijama) oko tekućih operanada u brzu cache memoriju. Ako je elemenata* x_i *previše da bi svi stali u brzu memoriju, dohvaćaju se po blokovima. Ako se binarni algoritam ne programira vrlo precizno, uz jasno poznavanje kako koristi brzu memoriju, mogao bi zahtijevati daleko veći broj dohvaćanja članova niza* x_i *nego standardni algoritam.*

Isplati li se investirati dodatnu memoriju i računsko vrijeme da bismo imali osigurane bitno manje ograde za svaki η_i, nego kod standardnog algoritma? Kod većine aplikacija ne. Usprkos mnogo većim ogradama stvarne greške η_i kod standardnog algoritma su bitno manje.

Malo više svjetla na tu zagonetku baca i statistička analiza osnovnih grešaka zaokruživanja za sumu $\varepsilon_1 + \cdots + \varepsilon_n$. Analiza koja pretpostavlja da su osnovne greške nezavisne slučajne varijable s očekivanjem nula i konačnom standardnom devijacijom, pokazuje da je očekivana vrijednost za njihovu sumu omeđena s $\sqrt{n}u$ što je daleko povoljnije od teoretske granice nu (npr. očekivana vrijednost za η_1 je oko $\sqrt{n-1}u$, što je mnogo povoljnije od $(n-1)u$).

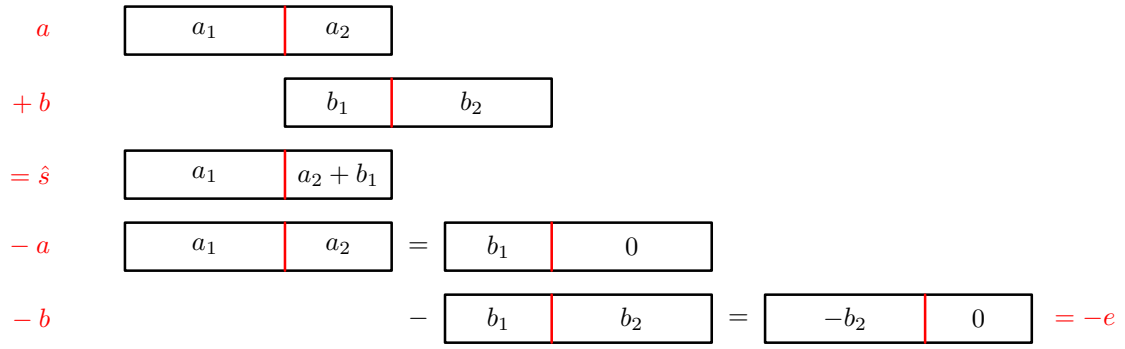
3.5.4. Kompenzirano sumiranje

Kompenzirano sumiranje je posebno atraktivno ako već koristimo najpreciznija aritmetiku u računalu (npr. dvostruku preciznost podataka i aritmetike). Radi se o rekurzivnom algoritmu s korektivnim sumandom koji je tako određen da gotovo poništi greške zaokruživanja.

Neka su $a = f\ell(a)$ i $b = f\ell(b)$ takvi da je $|a| \geq |b|$. Neka je

$$\hat{s} = f\ell(a + b).$$

Promotrimo sljedeću ilustraciju, koja koristi izdužene pravokutnike za prikaz mantise. Pri zbrajanju u akumulatoru (aritmetičkoj jedinici procesora) mantisa manjeg po modulu broja se prvo pomakne u desno za broj bitova koji odgovara razlici eksponenata brojeva a i b. Zatim se mantise zbroje, zaokruže na t bitova (koliko mantisa dopušta), a ako je pritom došlo do prekoračenja vrijednosti, mantisa se normalizira, a eksponent podesi.



Slika 3.5.1 Izvlačenje greške zaokruživanja

Iz slike 3.5.1 možemo zaključiti da je greška e,

$$e = -[((a+b)-a)-b] = (a-\hat{s})+b,$$

ako se izračuna na računalu u redoslijedu naznačenom zagradama

$$\hat{e} = (((a \oplus b) \ominus a) \ominus b),$$

dobra aproksimacija greške $(a+b)-\hat{s}$. Zapravo, za standardni način zaokruživanja u IEEE aritmetici može se pokazati da je

$$a + b = \hat{s} + \hat{e},$$

pa $\hat{e}$ reprezentira pravu grešku.

Sljedeći Kahanov algoritam koristi korekciju e u svakom koraku standardnog rekurzivnog sumacijskog algoritma. Nakon što je izračunata parcijalna suma s_{i-1}, odmah se računa njena korekcija, koja se u sljedećem koraku dodaje članu x_i prije njegova dodavanja parcijalnoj sumi s_{i-1}.

Gotovo uvijek je parcijalna suma po modulu veća od elementa koji joj se dodaje, pa korekcija e, koja je mala, može mijenjati element x_i, ali ne može mijenjati s_{i-1} koja je već najbolja strojna aproksimacija od $s_{i-2} + x_{i-1}$.

Evo algoritma.

```
s := 0;
e := 0;
for i := 1 to n do
   begin
   temp := s;
   z := x_i + e;
   s := temp + z;
   e := (temp − s) + z     {važan je redoslijed izvršavanja}
end
```

Ova metoda ipak ima dva manja nedostatka. Prvo, $\hat{e}$ nije nužno točna korekcija, jer uvjet $|s_{i-1}| \geq |x_i|$ (u algoritmu: $|temp| \geq |y|$) neće uvijek biti zadovoljen. Drugo, u računalu se računa $z := x_i \oplus e$ a ne $z := x_i + e$. Ipak, može se pokazati da η_i iz formule (3.5.12) za kompenziranu sumaciju zadovoljava

$$|\eta_i| \leq 2u + \mathcal{O}(nu^2),$$

što je gotovo nedostižan rezultat.

3.5.5. Stabilnost skalarnog produkta i osnovnih matričnih operacija

Neka su $x, y \in \mathbb{R}^n$, $x = [x_1, \ldots, x_n]^T$, $y = [y_1, \ldots, y_n]^T$. Da bismo analizirali skalarni produkt vektora x i y, označimo

$$s = s_n = x^T y = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n,$$

te za svako i parcijalne sume

$$s_i = x_1 y_1 + \cdots + x_i y_i.$$

Kao i kod sumacije, možemo definirati više algoritama za skalarni produkt, ali ćemo zbog jednostavnosti analizirati samo iterativni algoritam

$s_1 := x_1 \cdot y_1;$
for $i := 2$ **to** n **do**
  $s_i := s_{i-1} + x_i \cdot y_i;$

Zbog sličnosti s prijašnjim algoritmom za sumiranje n brojeva, možemo preuzeti neke dijelove izvoda ocjena grešaka zaokruživanja. Kao i prije, sa $\hat{s}_i$ označimo izračunate vrijednosti. Tada imamo,

$$\begin{aligned}
\hat{s}_1 &= fl(x_1 y_1) = x_1 y_1 (1 + \delta_1) \\
\hat{s}_2 &= fl(\hat{s}_1 + x_2 y_2) = [\hat{s}_1 + x_2 y_2 (1 + \delta_2)](1 + \varepsilon_1) \\
&= x_1 y_1 (1 + \delta_1)(1 + \varepsilon_1) + x_2 y_2 (1 + \delta_2)(1 + \varepsilon_1) \\
\hat{s}_3 &= x_1 y_1 (1 + \delta_1)(1 + \varepsilon_1)(1 + \varepsilon_2) + x_2 y_2 (1 + \delta_2)(1 + \varepsilon_1)(1 + \varepsilon_2) \\
&\quad + x_3 y_3 (1 + \delta_3)(1 + \varepsilon_2) \\
&\quad \vdots \\
\hat{s}_n &= fl(\hat{s}_{n-1} + x_n y_n) = [\hat{s}_{n-1} + x_n y_n (1 + \delta_n)](1 + \varepsilon_{n-1}) \\
&= x_1 (1 + \delta_1)(1 + \varepsilon_1)(1 + \varepsilon_2) \cdots (1 + \varepsilon_{n-1}) \\
&\quad + x_2 (1 + \delta_2)(1 + \varepsilon_1)(1 + \varepsilon_2) \cdots (1 + \varepsilon_{n-1}) \\
&\quad + x_3 (1 + \delta_3)(1 + \varepsilon_2) \cdots (1 + \varepsilon_{n-1}) \\
&\quad + \cdots + x_{n-1} (1 + \delta_{n-1})(1 + \varepsilon_{n-2})(1 + \varepsilon_{n-1}) \\
&\quad + x_n (1 + \delta_n)(1 + \varepsilon_{n-1}),
\end{aligned}$$

gdje je $|\delta_1| \leq u$, pa je

$$\hat{s}_n = x_1y_1(1+\eta_1) + x_2y_2(1+\eta_2) + \cdots + x_ny_n(1+\eta_n), \tag{3.5.17}$$

pri čemu η_i možemo ocijeniti pomoću leme 3.5.3, ili još oštrije pomoću leme 3.5.1(v)

$$|\eta_i| \leq 1.008 \cdot \begin{cases} nu, & i = 1, \\ (n-i+2)u, & 2 \leq i \leq n. \end{cases} \tag{3.5.18}$$

Relacija (3.5.17) je rezultat analize povratne greške koji se može ovako interpretirati.

Izračunati skalarni produkt $f\ell(x^Ty)$ je egzaktan skalarni produkt malo perturbiranih vektora x i y. Algoritam je povratno stabilan.

Kao perturbirane vektore možemo uzeti npr.

$$x + \delta x = [x_1(1+\eta_1), \ldots, x_n(1+\eta_n)] \quad \text{i} \quad y,$$

također,

$$x \quad \text{i} \quad y + \delta y = [y_1(1+\eta_1), \ldots, y_n(1+\eta_n)],$$

ali i

$$x + \delta x = [x_1\sqrt{1+\eta_1}, \ldots, x_n\sqrt{1+\eta_n}] \quad \text{i} \quad y + \delta y = [y_1\sqrt{1+\eta_1}, \ldots, y_n\sqrt{1+\eta_n}].$$

Da bismo dobili ocjenu za grešku unaprijed, koristimo analogne ocjene kao u relaciji (3.5.14),

$$\begin{aligned} \frac{|\hat{s}_n - s_n|}{|s_n|} &\leq \frac{|x_1y_1| + |x_2y_2| + \cdots + |x_ny_n|}{|x_1y_1 + x_2y_2 + \cdots + x_ny_n|} \max_i\{|\eta_i|\} \\ &\leq \mathrm{cond}(s_n) 1.008nu. \end{aligned} \tag{3.5.19}$$

Vidimo da broj uvjetovanosti $\mathrm{cond}(s_n)$ za skalarni produkt može biti po volji velik. To se događa kad je $|x^Ty| \ll |x|^T|y|$. Zaključak je isti kao i kod sumiranja: greška unaprijed može biti velika, pa algoritam nije stabilan unaprijed. Ipak on je stabilan unaprijed kad su svi produkti x_iy_i istog predznaka (tada je $\mathrm{cond}(s_n) = 1$).

Računanje 2-norme,

$$\|x\| = \sqrt{x_1^2 + \cdots + x_n^2}$$

zaslužuje razmatranje. S obzirom da je za taj slučaj broj uvjetovanosti 1, vrijedi

$$f\ell(x_1^2 + \cdots + x_n^2) = (x_1^2 + \cdots + x_n^2)(1+\varepsilon), \quad |\varepsilon| \leq 1.008nu,$$

pa se jednostavno dobije

$$f\ell(\|x\|) = \|x\|(1+\varepsilon_{\|\cdot\|}), \quad |\varepsilon_{\|\cdot\|}| \leq \sqrt{1+1.008nu}(1+u) - 1 \leq (1+0.504n)u$$

što je manje od $0.504(n+1)u$ za $n \geq 2$. Dakle, računanje norme je stabilno (unaprijed i unazad).

Zadatak 3.5.4 *Je li računanje ostalih normi, npr.* $\|x\|_1$, $\|x\|_\infty$, $\|x\|_p$, *stabilno?*

Za razliku od skalarnog (unutrašnjeg) produkta, vanjski produkt vektora

$$A = xy^T$$

nije povratno stabilan. Ovdje čak x i y ne trebaju biti iste dimenzije. Ako elemente matrice A označimo s a_{ij}, onda vrijedi $a_{ij} = x_i x_j$, pa je

$$\hat{a}_{ij} = f\ell(a_{ij}) = a_{ij}(1 + \varepsilon_{ij}), \quad |\varepsilon_{ij}| \leq u$$

za sve i, j. Stoga je

$$\hat{A} = f\ell(xy^T) = A + E, \quad E = [\,\varepsilon_{ij}\,],$$

pa vrijedi

$$|E| \leq u|xy^T| = u|A|,$$

odnosno,

$$|\hat{A} - A| \leq u|A|,$$

pa je računanje vanjskog produkta operacija stabilna unaprijed. Razlika između tipova stabilnosti unutrašnjeg i vanjskog produkta izražava opći princip: **numerički proces će prije biti stabilan unazad (unaprijed) ako ima više (manje) ulaznih nego izlaznih podataka**.

4. Sustavi linearnih jednadžbi

Jedan od osnovnih problema numeričke matematike je rješavanje linearnih sustava jednadžbi. U ovom poglavlju istraživat ćemo metode za rješavanje kvadratnih $n \times n$ sustava, tj. sustava s n jednadžbi i n nepoznanica,

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j + \cdots + a_{2n}x_n &= b_2 \\
\vdots \qquad \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad \vdots & \\
a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j + \cdots + a_{in}x_n &= b_i \\
\vdots \qquad \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad \vdots & \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nj}x_j + \cdots + a_{nn}x_n &= b_n
\end{aligned}$$

Matrica $A = [\, a_{ij} \,]_{i,j=1}^{n} \in \mathbb{R}^{n \times n}$ je **matrica sustava**, a njeni elementi su **koeficijenti sustava**. Vektor $b = [\, b_i \,]_{i=1}^{n} \in \mathbb{R}^n$ je **vektor desne strane** sustava. Treba odrediti **vektor nepoznanica** $x = [\, x_i \,]_{i=1}^{n} \in \mathbb{R}^n$ tako da vrijedi $Ax = b$.

Kako znamo iz linearne algebre, za teorijsku matematiku je rješavanje sustava $Ax = b$ gotovo trivijalan problem, posebno u slučaju kada je matrica sustava kvadratna i regularna. Rješenje x je dano formulom $x = A^{-1}b$ u kojoj je A^{-1} inverzna matrica od A ($AA^{-1} = A^{-1}A = I$). Pri tome postoje eksplicitne formule i za elemente matrice A^{-1} i za samo rješenje x.

Osim toga, svima dobro poznata Gaussova metoda eliminacija dolazi do rješenja u $O(n^3)$ elementarnih operacija[3]. Dakle, situacija je potpuno jasna: rješenje $x = A^{-1}b$ postoji, i to samo jedno, i znamo jednostavan algoritam koji to rješenje eksplicitno računa koristeći samo jednostavne aritmetičke operacije.

U primijenjenoj matematici, posebno u numeričkoj linearnoj algebri (grana numeričke matematike koja se bavi problemima linearne algebre) je situacija puno kompliciranija. Zašto? U numeričkoj matematici danas riješiti problem znači biti u stanju u konkretnoj situaciji sa konkretnim podacima, koristeći računalo, **brzo** doći do **dovoljno točne numeričke aproksimacije** rješenja. Na primjer, ako su $n \times n$ matrica A i vektor b zapisani u nekim datotekama na disku, ili su dane

[3]Ovdje elementarna operacija označava zbrajanje, oduzimanje, množenje ili dijeljenje.

procedure (potprogrami) koji generiraju A i b, onda je zadatak izračunati numeričke vrijednosti x_i, $i = 1, \ldots, n$.

Ono po čemu su računala poznata je brzina – dobro jednoprocesorsko računalo može napraviti npr. 10^9 operacija u sekundi. Međutim, ono što je osnovna značajka moderne numeričke matematike je da joj u primjenama dolaze problemi sve većih dimenzija. Kako je broj operacija u Gaussovim eliminacijama $O(n^3)$, to znači da je npr. za $n = 10^5$ broj operacija reda veličine 10^{15} pa brzinom od 10^9 operacija u sekundi dobivamo vrijeme izvršavanja[4] oko 10^6 sekundi, što je više od deset dana.

U ozbiljnim primjenama treba u procesu projektiranja puno puta rješavati takve sustave. Tada je brzina računala samo jedan faktor u razumno (ili dovoljno) brzom rješavanju linearnog sustava. Dakle, problem koji je matematički posve jednostavan u praksi može biti puno izazovniji i netrivijalniji.

U divljenju brzini kojom računalo zbraja, oduzima ili množi brojeve često zaboravljamo da su rezultati tih operacija uglavnom **netočni**. Sjetimo se, računalo reprezentira brojeve i izvršava računske operacije koristeći fiksni broj znamenki – to znači da se rezultati operacija zaokružuju. Dakle, moguće je da je svaka od $O(n^3)$ operacija u Gaussovom algoritmu izvršena s greškom zaokruživanja. Koliko onda možemo vjerovati izračunatom rješenju?

4.1. Vodič kroz ovo poglavlje

Materijal u ovom poglavlju je prije svega namijenjen naprednom studiranju numeričke matematike i usvajanju nekih osnovnih principa. Ipak, organiziran je u više nivoa tako da ga mogu čitati i početnici i napredniji čitatelji. Sljedeći pregled bi trebao pomoći čitatelju pri planiranju proučavanja ponuđenog materijala.

1. nivo : Proučiti i shvatiti barem jedan primjer iz odjeljka 4.2. o primjeni linearnih sustava jednadžbi. Materijal iz odjeljka 4.3. čitati do iskaza teorema. Pažljivo, uz papir i olovku, obraditi primjere i opis Gaussovih eliminacija na primjeru male dimenzije. Čitatelj na ovom nivou treba naučiti kako funkcioniraju Gaussove eliminacije, uočiti vezu između procesa eliminacija i faktorizacije matrice koeficijenata sustava, svladati algoritme za rješavanje trokutastih sustava, te biti u stanju na ruke riješiti sustave manje dimenzije.

2. nivo : Materijal iz odjeljka 4.3. svladati u potpunosti.

3. nivo : Sekcije 4.3. i 4.4. svladati u potpunosti. Po potrebi se služiti materijalom iz dodataka ili druge literature. Analizirati i shvatiti sve primjere. Shvatiti važnost pivotiranja za numeričku stabilnost Gaussovih eliminacija.

[4]Ovdje namjerno pojednostavljujemo ocjenu vremena izvršavanja. Za precizniju procjenu je potrebno uračunati i vrijeme pristupa memoriji i dohvaćanje podataka, veličinu tzv. cache memorije, itd. U ovom trenutku važno je dobiti osjećaj za red veličine.

4. nivo : Na ovom nivou čitatelj bez poteškoća čita sav materijal i služi se programima za rješavanje sustava na računalu.

4.2. Primjeri: Kako nastaje linearni sustav jednadžbi

U ovom odjeljku dajemo niz primjera problema iz primijenjene matematike čije rješavanje je bazirano na sustavima linearnih jednadžbi.

Primjer 4.2.1 *Zadani su parovi točaka (x_i, y_i), $i = 0, \ldots, n$, gdje su $y_i = f(x_i)$ izmjerene (ili na neki drugi način dobivene) vrijednosti funkcije $f(x)$ koju želimo aproksimirati polinomom $p(x)$ stupnja n. Pretpostavljamo i da je $x_i \neq x_j$ za $i \neq j$. Kriterij za odabir polinoma $p(x)$ je da u točkama x_i ima vrijednost $p(x_i) = y_i = f(x_i)$. (Govorimo o* **interpolacijskom** *polinomu.) Ako $p(x)$ prikažemo u kanonskom obliku*

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = \sum_{j=0}^{n} a_j x^j$$

onda treba odrediti koeficijente $a_0, \ldots, a_n$ tako da vrijede uvjeti interpolacije $p(x_i) = y_i$, $i = 0, \ldots, n$, tj.

$$\begin{aligned}
a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_{n-1} x_0^{n-1} + a_n x_0^n &= y_0 \\
a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_{n-1} x_1^{n-1} + a_n x_1^n &= y_1 \\
\vdots \quad \vdots \quad \vdots \quad \cdots \quad \vdots \quad \vdots \quad \vdots & \\
a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_{n-1} x_i^{n-1} + a_n x_i^n &= y_i \\
a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_{n-1} x_n^{n-1} + a_n x_n^n &= y_n.
\end{aligned}$$

Vidimo da svaki uvjet interpolacije daje jednu jednadžbu u kojoj se nepoznati koeficijenti pojavljuju linearno. Sve jednadžbe čine sustav linearnih jednadžbi kojeg možemo matrično zapisati u obliku $Va = y$, tj.

$$\underbrace{\begin{bmatrix}
1 & x_0 & x_0^2 & x_0^3 & \cdots & x_0^{n-1} & x_0^n \\
1 & x_1 & x_1^2 & x_1^3 & \cdots & x_1^{n-1} & x_1^n \\
 & & \cdots & & \cdots & & \\
1 & x_i & x_i^2 & x_i^3 & \cdots & x_i^{n-1} & x_i^n \\
 & & \cdots & & \cdots & & \\
1 & x_n & x_n^2 & x_n^3 & \cdots & x_n^{n-1} & x_n^n
\end{bmatrix}}_{V}
\underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix}}_{a}
= \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}}_{y}. \tag{4.2.1}$$

Matrica V zove se Vandermondeova matrica. Zahvaljujući njenom specijalnom obliku, moguće je efikasno odrediti $a = V^{-1}y$.

Primjer 4.2.2 *Promotrimo sljedeći rubni problem:*

$$-\frac{d^2}{dx^2}u(x) = f(x), \quad 0 < x < 1, \tag{4.2.2}$$

$$u(0) = u(1) = 0. \tag{4.2.3}$$

Rješenje u problema (4.2.2, 4.2.3) aproksimirat ćemo na skupu od konačno mnogo točaka iz segmenta $[0,1]$*. Odaberimo prirodan broj* n *i definirajmo*

$$h = \frac{1}{n+1}, \quad x_i = ih, \quad i = 0, \ldots, n+1. \tag{4.2.4}$$

Sada promatramo vrijednosti $u_i = u(x_i)$*,* $i = 0, \ldots, n+1$*. Iz uvjeta (4.2.3) odmah vidimo da je* $u_0 = u_{n+1} = 0$*. Iz Taylorovog teorema je*

$$u(x_i + h) = u(x_i) + u'(x_i)h + \frac{u''(x_i)}{2}h^2 + \frac{u'''(x_i)}{6}h^3 + \frac{u^{(4)}(x_i + \alpha_i)}{24}h^4, \tag{4.2.5}$$

$$u(x_i - h) = u(x_i) - u'(x_i)h + \frac{u''(x_i)}{2}h^2 - \frac{u'''(x_i)}{6}h^3 + \frac{u^{(4)}(x_i + \zeta_i)}{24}h^4, \tag{4.2.6}$$

gdje su $\alpha_i \in (x_i, x_i + h)$*,* $\zeta_i \in (x_i - h, x_i)$*. Zbrajanjem jednadžbi (4.2.5) i (4.2.6) za* $i = 1, \ldots, n$ *dobivamo*

$$u_{i+1} + u_{i-1} = 2u_i + u''(x_i)h^2 + (u^{(4)}(x_i + \alpha_i) + u^{(4)}(x_i + \zeta_i))\frac{h^4}{24}, \tag{4.2.7}$$

tj.

$$-u''(x_i) = \frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} - \mathbf{e}_i, \tag{4.2.8}$$

gdje je

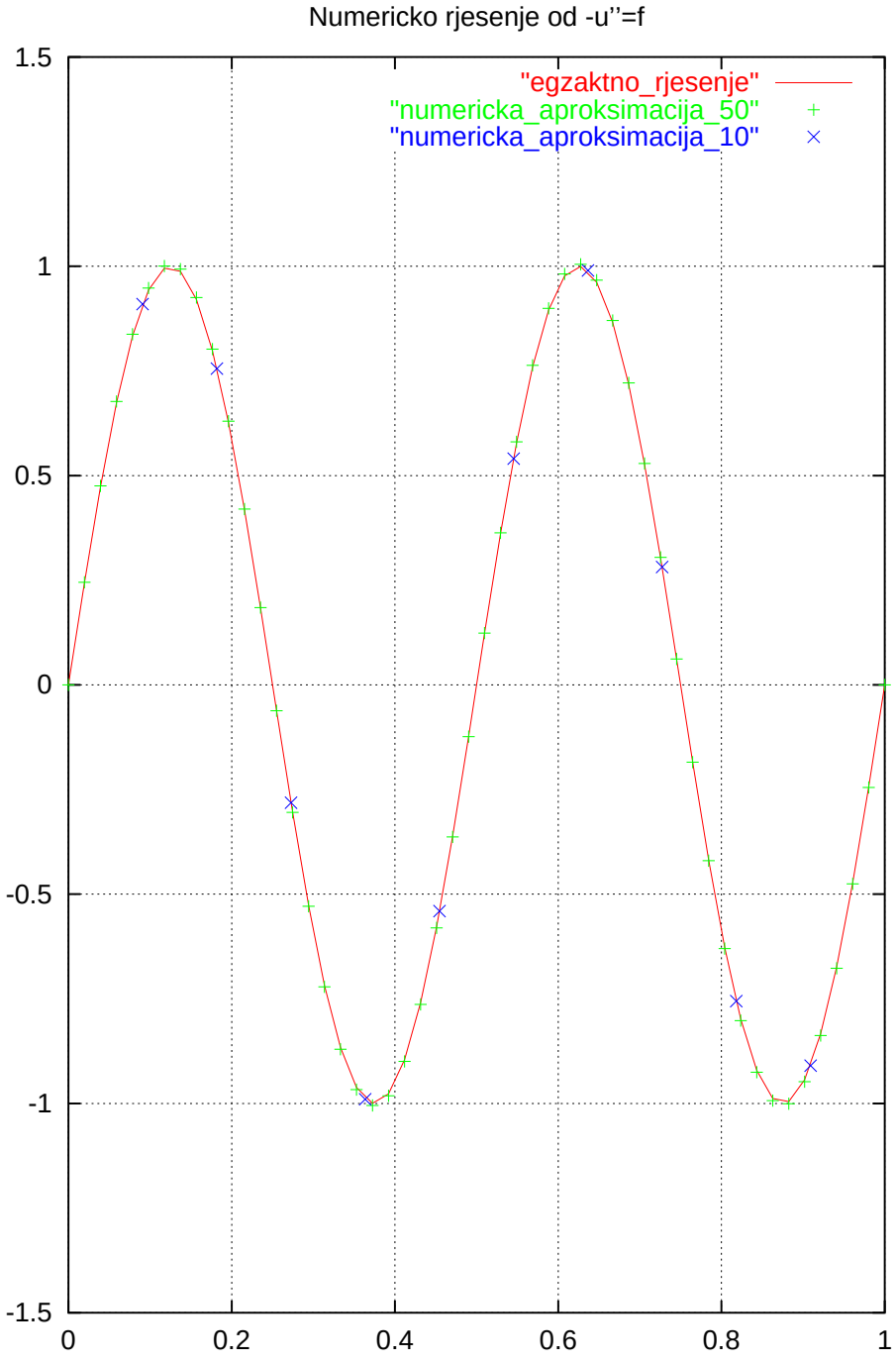
$$\mathbf{e}_i = -(u^{(4)}(x_i + \alpha_i) + u^{(4)}(x_i + \zeta_i))\frac{h^2}{24}.$$

Matrično to možemo zapisati kao

$$\underbrace{\begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix}}_{T_n} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-2} \\ u_{n-1} \\ u_n \end{bmatrix}}_{u} = h^2 \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-2} \\ f_{n-1} \\ f_n \end{bmatrix}}_{f} + h^2 \underbrace{\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \\ \vdots \\ \mathbf{e}_{n-2} \\ \mathbf{e}_{n-1} \\ \mathbf{e}_n \end{bmatrix}}_{\mathbf{e}}. \tag{4.2.9}$$

U jednadžbi $T_n u = h^2 f + h^2 \mathbf{e}$ *član* $\mathbf{e}$ *ne znamo, pa ga zanemarujemo, tj. pokušat ćemo riješiti* $T_n \hat{u} = h^2 f$*. Primijetimo da pod određenim uvjetima možemo očekivati da je* $\|\mathbf{e}\|_2$ *mali broj,* $\|\mathbf{e}\|_2 = O(h^2)$*.*

Tek da ilustriramo, uzmimo npr. $f(x) = 16\pi^2 \sin 4\pi x$*, za kojeg znamo točno rješenje* $u(x) = \sin 4\pi x$*. Uzet ćemo* $n = 10$ *i* $n = 50$ *i vidjeti koliko su dobre dobivene aproksimacije. Rezultati su dani na sljedećoj slici.*



Egzaktno rješenje i numerička rješenja dobivena s $n = 10$ *i* $n = 50$ *čvorova.*

Za iskusnije čitatelje odmah ćemo malo prodiskutirati dobiveno rješenje. (Ostali mogu preskočiti sljedeću diskusiju.) Lako se pokaže da je matrica T_n regularna. Neka je $\hat{u} = h^2 T_n^{-1} f$. Imamo

$$u - \hat{u} = T_n^{-1}(h^2 f + h^2 \mathbf{e}) - h^2 T_n^{-1} f = h^2 T_n^{-1} \mathbf{e}, \tag{4.2.10}$$

pa je

$$\frac{\|u - \hat{u}\|_2}{\|\hat{u}\|_2} = \frac{\|T_n^{-1}\mathbf{e}\|_2}{\|T_n^{-1} f\|_2} \leq \|T_n\|_2 \|T_n^{-1}\|_2 \frac{\|\mathbf{e}\|_2}{\|f\|_2}. \tag{4.2.11}$$

Pri tome smo koristili sljedeće nejednakosti:

$$\|T_n^{-1}\mathbf{e}\|_2 \leq \|T_n^{-1}\|_2 \|\mathbf{e}\|_2 \quad \textit{(jer je } \|T_n^{-1}\|_2 = \max_{x \neq 0} \frac{\|T_n^{-1} x\|_2}{\|x\|_2} \textit{)},$$

$$\|T_n^{-1} f\|_2 \geq \frac{\|f\|_2}{\|T_n\|_2} \quad \textit{(jer je } \|T_n\|_2 = \max_{x \neq 0} \frac{\|T_n x\|_2}{\|x\|_2} = \max_{y \neq 0} \frac{\|y\|_2}{\|T_n^{-1} y\|_2} \textit{)}.$$

Vidimo da nejednakost u relaciji (4.2.11) može za neke izbore vektora $\mathbf{e}$ *i* f *prijeći u jednakost, što znači da je relacija (4.2.11) realistična ocjena greške diskretizacije* $\mathbf{e}$ *na aproksimaciju rješenja polaznog rubnog problema (4.2.2)–(4.2.3).*

Mi ne možemo točno odrediti niti $\hat{u}$ jer računamo u aritmetici s konačnom preciznosti. Neka je $\tilde{u}$ izračunata aproksimacija vektora $\hat{u}$. Pitanje je koliku točnost aproksimacije treba imati $\tilde{u}$. Iz relacije (4.2.11) slijedi da je zadovoljavajuća točnost postignuta ako je $\|\hat{u} - \tilde{u}\|_2 / \|\tilde{u}\|_2$ najviše reda veličine $\kappa_2(T_n)\|\mathbf{e}\|_2 / \|f\|_2$ (sjetimo se da mi zapravo želimo aproksimirati u).

4.3. Gaussove eliminacije i trokutaste faktorizacije

Metoda Gaussovih eliminacija je svakako najstariji, najjednostavniji i najpoznatiji algoritam za rješavanje sustava linearnih jednadžbi $Ax = b$. Ideja je jednostavna. Da bismo riješili sustav

$$\begin{aligned} 2x_1 - \ \ x_2 &= 1 \\ -x_1 + 2x_2 &= 1 \end{aligned}$$

dovoljno je primijetiti da zbog prve jednadžbe vrijedi $x_1 = \dfrac{1}{2}(1 + x_2)$, pa je druga jednadžba

$$-\underbrace{\frac{1}{2}(1 + x_2)}_{x_1} + 2x_2 = 1, \quad \text{tj. } \frac{3}{2} x_2 = \frac{3}{2}, \quad \text{tj.} \quad x_2 = 1,$$

odakle je $x_1 = 1$. Kažemo da smo x_1 **eliminirali** iz druge jednadžbe.

Ovu ideju možemo lako generalizirati na dimenziju $n > 1$, gdje sustavno eliminiramo neke nepoznanice iz nekih jednadžbi. Pokazuje se da takav algoritam ima dosta zanimljivu strukturu i da ga se može ekvivalentno zapisati u terminima matričnih operacija. Kvalitativno novi moment u analizi metode eliminacija nastaje kada sam proces eliminacija interpretiramo kao faktorizaciju matrice sustava A na produkt trokutastih matrica.[1]

4.3.1. Matrični zapis metode eliminacija

Primjer 4.3.1 *Riješimo sljedeći sustav jednadžbi:*

$$\begin{aligned} 5x_1 + x_2 + 4x_3 &= 19 \\ 10x_1 + 4x_2 + 7x_3 &= 39 \\ -15x_1 + 5x_2 - 9x_3 &= -32 \end{aligned} \quad \equiv \quad \underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ -15 & 5 & -9 \end{bmatrix}}_{A = [\,a_{ij}\,]_{i,j=1}^{3}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 19 \\ 39 \\ -32 \end{bmatrix}}_{b = [\,b_i\,]_{i=1}^{3}}. \tag{4.3.1}$$

Koristimo metodu supstitucija, odnosno eliminacija. Prvo iz prve jednadžbe izrazimo x_1 pomoću x_2 i x_3, te to uvrstimo u zadnje dvije jednadžbe, koje postaju dvije jednadžbe s dvije nepoznanice (x_2 i x_3). Dobivamo

$$x_1 = \frac{1}{5}\,(19 - x_2 - 4x_3),$$

pa druga jednadžba sada glasi

$$\frac{10}{5}\,(19 - x_2 - 4x_3) + 4x_2 + 7x_3 = 39,$$

tj.

$$-\frac{10}{5}\,(x_2 + 4x_3) + 4x_2 + 7x_3 = 39 + \Big(-\frac{10}{5}\,19\Big).$$

Dakle, efekt ove transformacije je ekvivalentno prikazan kao rezultat množenja prve jednadžbe s

$$-\frac{a_{21}}{a_{11}} = -\frac{10}{5} = -2$$

i zatim njenim dodavanjem (pribrajanjem) drugoj jednadžbi. Druga jednadžba sada glasi

$$2x_2 - x_3 = 1.$$

Ako ovu transformaciju sustava zapišemo matrično, imamo

$$\underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ -15 & 5 & -9 \end{bmatrix}}_{A} \quad \mapsto \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{L^{(2,1)}} \underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ -15 & 5 & -9 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ -15 & 5 & -9 \end{bmatrix}}_{A^{(1)} = \left[\,a_{ij}^{(1)}\,\right]_{i,j=1}^{3}}.$$

[1] Takav koncept je prvi uveo Alan Turing.

Nepoznanicu x_1 eliminiramo iz zadnje jednadžbe ako prvu pomnožimo s

$$-\frac{a_{31}^{(1)}}{a_{11}^{(1)}} = -\frac{15}{5} = -3$$

i onda je pribrojimo zadnjoj. To znači sljedeću promjenu matrice koeficijenata:

$$\underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ -15 & 5 & -9 \end{bmatrix}}_{A^{(1)}} \mapsto \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}}_{L^{(3,1)}} \underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ -15 & 5 & -9 \end{bmatrix}}_{A^{(1)}} = \underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 8 & 3 \end{bmatrix}}_{A^{(2)} = \left[a_{ij}^{(2)} \right]_{i,j=1}^{3}}.$$

Vektor desne strane je u ove dvije transformacije promijenjen u

$$\underbrace{\begin{bmatrix} 19 \\ 39 \\ -32 \end{bmatrix}}_{b} \mapsto \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{L^{(2,1)}} \begin{bmatrix} 19 \\ 39 \\ -32 \end{bmatrix} = \underbrace{\begin{bmatrix} 19 \\ 1 \\ -32 \end{bmatrix}}_{b^{(1)}}$$

$$\mapsto \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}}_{L^{(3,1)}} \begin{bmatrix} 19 \\ 1 \\ -32 \end{bmatrix} = \underbrace{\begin{bmatrix} 19 \\ 1 \\ 25 \end{bmatrix}}_{b^{(2)}}.$$

Novi, ekvivalentni, sustav je $A^{(2)}x = b^{(2)}$, tj.

$$\begin{aligned} 5x_1 + \ x_2 + 4x_3 &= 19 \\ 2x_2 - \ x_3 &= \ 1 \\ 8x_2 - 3x_3 &= 25, \end{aligned} \tag{4.3.2}$$

u kojem su druga i treća jednadžba sustav od dvije jednadžbe s dvije nepoznanice. Očito je da rješenje $x = (x_1, x_2, x_3)^T$ sustava (4.3.1) zadovoljava i sustav (4.3.2). Obratno, ako trojka x_1, x_2, x_3 zadovoljava (4.3.2), onda množenjem prve jednadžbe u (4.3.2) s 2 i zatim pribrajanjem drugoj jednadžbi, dobijemo drugu jednadžbu sustava (4.3.1). Na sličan način iz prve i treće jednadžbe sustava (4.3.2) rekonstruiramo treću jednadžbu polaznog sustava (4.3.1). U tom smislu kažemo da sustavi (4.3.1) i (4.3.2) ekvivalentni: imaju isto rješenje.

Nadalje, primijetimo da smo proces eliminacija (tj. izražavanja nepoznanice x_1 pomoću x_2 i x_3 i eliminiranjem x_1 iz zadnje dvije jednadžbe) jednostavno opisali matričnim operacijama. Eliminaciju nepoznanice x_1 smo prikazali kao rezultat množenja matrice koeficijenata i vektora desne strane s lijeva jednostavnim matricama $L^{(2,1)}$ i $L^{(3,1)}$.

Jasno je da je sustav (4.3.2) jednostavniji od polaznog. Zato sada nastavljamo s primjenom iste strategije: iz treće jednadžbe eliminiramo x_2 tako što drugu jednadžbu pomnožimo s

$$-\frac{a_{32}^{(2)}}{a_{22}^{(2)}} = -4$$

i pribrojimo je trećoj. Tako treća jednadžba postaje $7x_3 = 21$, a cijeli sustav ima oblik

$$\begin{aligned} 5x_1 + \ x_2 + 4x_3 &= 19 \\ 2x_2 - \ x_3 &= \ 1 \\ 7x_3 &= 21. \end{aligned} \tag{4.3.3}$$

Transformaciju eliminacije x_2 iz treće jednadžbe možemo matrično zapisati kao transformaciju matrice koeficijenata

$$\underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 8 & 3 \end{bmatrix}}_{A^{(2)}} \quad \mapsto \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix}}_{L^{(3,2)}} \underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 8 & 3 \end{bmatrix}}_{A^{(2)}} = \underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 0 & 7 \end{bmatrix}}_{A^{(3)}} \tag{4.3.4}$$

i transformaciju vektora desne strane

$$\underbrace{\begin{bmatrix} 19 \\ 1 \\ 25 \end{bmatrix}}_{b^{(2)}} \quad \mapsto \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix}}_{L^{(3,2)}} \underbrace{\begin{bmatrix} 19 \\ 1 \\ 25 \end{bmatrix}}_{b^{(2)}} = \underbrace{\begin{bmatrix} 19 \\ 1 \\ 21 \end{bmatrix}}_{b^{(3)} = (b_i^{(3)})_{i=1}^3}. \tag{4.3.5}$$

Sustav (4.3.3), koji je ekvivalentan polaznom, lako riješimo.

1. *Iz treće jednadžbe je $x_3 = \dfrac{21}{7} = 3$.*
2. *Iz druge jednadžbe je $x_2 = \dfrac{1}{2}(1 + x_3) = 2$.*
3. *Iz prve jednadžbe je $x_1 = \dfrac{1}{5}(19 - x_2 - 4x_3) = 1$.*

Jednostavna provjera potvrđuje da su x_1, x_2, x_3 rješenja polaznog sustava (4.3.1).

Analizirajmo postupak rješavanja u prethodnom primjeru. Pažljivo pogledajmo oblik matrica u realaciji

$$A^{(3)} = L^{(3,2)} L^{(3,1)} L^{(2,1)} A.$$

Matrica $A^{(3)}$ je gornjetrokutasta, a produkt $L^{(3,2)} L^{(3,1)} L^{(2,1)}$ je donjetrokutasta matrica. Dakle, polaznu matricu A smo množenjem slijeva donjetrokutastom matricom načinili gornjetrokutastom. To možemo pročitati i ovako:

$$A = LA^{(3)}, \quad L = (L^{(2,1)})^{-1} (L^{(3,1)})^{-1} (L^{(3,2)})^{-1},$$

gdje je L donjetrokutasta matrica. Lako se provjerava da je

$$L = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{(L^{(2,1)})^{-1}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}}_{(L^{(3,1)})^{-1}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 1 \end{bmatrix}}_{(L^{(3,2)})^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 4 & 1 \end{bmatrix}.$$

Dakle, matricu A smo napisali kao produkt donjetrokutaste i gornjetrokutaste matrice, $A = LA^{(3)}$. Gornjetrokutastu matricu u ovom kontekstu obično označavamo s $U = A^{(3)}$, pa je A rastavljena na produkt $A = LU$. Govorimo o **LU faktorizaciji** matrice A (neki tu faktorizaciju zovu i LR faktorizacija matrice). Uočimo da je računanje produkta koji definira matricu L jednostavno. Inverze matrica $L^{(2,1)}$, $L^{(3,1)}$ i $L^{(3,2)}$ dobijemo samo promjenom predznaka netrivijalnih elemenata u donjem trokutu, a cijeli produkt jednostavno je stavljanje tih elemenata na odgovarajuće pozicije u matrici L. Sada još primijetimo da je relacija (4.3.3) zapravo linearni sustav $Ux = b^{(3)}$, gdje je $b^{(3)} = L^{(3,2)}L^{(3,1)}L^{(2,1)}b = L^{-1}b$. Jasno,

$$x = A^{-1}b = (LU)^{-1}b = U^{-1}L^{-1}b.$$

Dakle, u terminima matrice A i vektora b, linearni sustav u primjeru 4.3.1 riješen je metodom koja se sastoji od tri glavna koraka.

1. Matricu sustava A treba faktorizirati u obliku $A = LU$, gdje je L donjetrokutasta, a U gornjetrokutasta matrica.
2. Rješavanjem donjetrokutastog sustava $Ly = b$ treba odrediti vektor $y = L^{-1}b$.
3. Rješavanjem gornjetrokutastog sustava $Ux = y$ treba odrediti vektor $x = U^{-1}y = U^{-1}(L^{-1}b)$.

Ovakav zapis metode opisane u primjeru 4.3.1 ima niz prednosti:

- Operacije su iskazane u terminima matrice A i desne strane b, a ne u terminima izražavanja neke nepoznanice pomoću ostalih. Umjesto "x_1 izrazimo pomoću $x_2, x_3, \ldots$" i sl., operacije izražavamo operacijama s matricama i vektorima. To omogućuje jednostavnu i sustavnu primjenu opisane metode na sustav s proizvoljnim brojem nepoznanica. Sam linearni sustav je u računalu pohranjen kao matrica koeficijenata A i vektor desne strane b. Dakle, ovakav zapis metode eliminacija je prirodan.
- Ponekad u primjenama rješavamo nekoliko linearnih sustava s istom matricom A, ali s nizom različitih desnih strana b. Vidimo da je u tom slučaju transformacije na matrici A dovoljno napraviti jednom (prvi korak u gornjem zapisu metode), a zatim za različite desne strane provesti samo zadnja dva koraka.

4.3.2. Trokutasti sustavi: rješavanje supstitucijama unaprijed i unazad

Trokutasti sustavi jednadžbi lako se rješavaju. Pogledajmo, na primjer, donjetrokutasti sustav $Lx = b$ dimenzije $n = 4$:

$$\begin{bmatrix} \ell_{11} & 0 & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} & 0 \\ \ell_{41} & \ell_{42} & \ell_{43} & \ell_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

Neka je matrica L regularna. To znači da su $\ell_{ii} \neq 0$ za $i = 1, 2, 3, 4$. Očito je

$$\begin{aligned} x_1 &= \frac{b_1}{\ell_{11}} \\ x_2 &= \frac{1}{\ell_{22}} (b_2 - \ell_{21} x_1) \\ x_3 &= \frac{1}{\ell_{33}} (b_3 - \ell_{31} x_1 - \ell_{32} x_2) \\ x_4 &= \frac{1}{\ell_{44}} (b_4 - \ell_{41} x_1 - \ell_{42} x_2 - \ell_{43} x_3). \end{aligned}$$

Vidimo da x_1 možemo odmah izračunati, a za $i > 1$ formula za x_i je funkcija od b_i, i-tog retka matrice L i nepoznanica $x_1, \ldots, x_{i-1}$ koje su prethodno već izračunate. Dakle, prvo izračunamo x_1, pa tu vrijednost uvrstimo u izraz koji daje x_2; zatim x_1 i x_2 uvrstimo u izraz za x_3, itd. Ovakav postupak zovemo **supstitucija unaprijed**.

Algoritam 4.3.1 *Rješavanje linearnog sustava jednadžbi $Lx = b$ s regularnom donjetrokutastom matricom $L \in \mathbb{R}^{n \times n}$.*

/* Supstitucija unaprijed za $Lx = b$ */

$x_1 = \dfrac{b_1}{\ell_{11}}$;

za $i = 2, \ldots, n$ {

$x_i = \left(b_i - \displaystyle\sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Big/ \ell_{ii}$; }

Prebrojimo operacije u gornjem algoritmu:

- dijeljenja: n ;
- množenja: $1 + 2 + \cdots + (n-1) = \dfrac{1}{2} n(n-1)$;
- zbrajanja i oduzimanja: $1 + 2 + \cdots + (n-1) = \dfrac{1}{2} n(n-1)$.

Dakle, ukupna složenost je $O(n^2)$.

Gornjetrokutaste sustave rješavamo na sličan način. Ako je sustav $Ux = b$ oblika

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}, \quad \prod_{i=1}^{4} u_{ii} \neq 0,$$

onda, polazeći od zadnje jednadžbe unazad, imamo

$$\begin{aligned} x_4 &= \frac{b_4}{u_{44}} \\ x_3 &= \frac{1}{u_{33}} (b_3 - u_{34}x_4) \\ x_2 &= \frac{1}{u_{22}} (b_2 - u_{23}x_3 - u_{24}x_4) \\ x_1 &= \frac{1}{u_{11}} (b_1 - u_{12}x_2 - u_{13}x_3 - u_{14}x_4). \end{aligned}$$

Ovakav postupak zovemo **supstitucija unazad**.

Algoritam 4.3.2 *Rješavanje linearnog sustava jednadžbi $Ux = b$ s regularnom gornjetrokutastom matricom $U \in \mathbb{R}^{n\times n}$.*

/∗ Supstitucija unazad za $Ux = b$ ∗/

$x_n = \dfrac{b_n}{u_{nn}}$;

za $i = n-1, \ldots, 1$ {

$x_i = \left(b_i - \displaystyle\sum_{j=i+1}^{n} u_{ij}x_j\right) \Big/ u_{ii}$; }

Kao i kod supstitucija naprijed, složenost ovog algoritma je $O(n^2)$.

4.3.3. LU faktorizacija

Sada kad smo uočili da se rješavanje linearnog sustava $Ax = b$ faktoriziranjem matrice A svodi na trokutaste sustave, ostaje nam posebno proučiti faktorizaciju matrice $A \in \mathbb{R}^{n\times n}$ na produkt donje i gornjetrokutaste matrice. Zanima nas proizvoljna dimenzija n, ali ćemo zbog jednostavnosti razmatranja na početku sve

ideje ilustrirati na primjeru $n = 5$. Neka je

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}.$$

Sjetimo se, eliminacija prve nepoznanice manifestira se poništavanjem koeficijenata na pozicijama $(2,1),(3,1),\ldots,(n,1)$. To možemo napraviti u jednom potezu[2]. Ako definiramo matricu

$$L^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\dfrac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ -\dfrac{a_{31}}{a_{11}} & 0 & 1 & 0 & 0 \\ -\dfrac{a_{41}}{a_{11}} & 0 & 0 & 1 & 0 \\ -\dfrac{a_{51}}{a_{11}} & 0 & 0 & 0 & 1 \end{bmatrix},$$

onda je x_1 eliminiran iz svih jednadžbi osim prve, tj.

$$A^{(1)} \equiv L^{(1)}A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} & a_{35}^{(1)} \\ 0 & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} & a_{45}^{(1)} \\ 0 & a_{52}^{(1)} & a_{53}^{(1)} & a_{54}^{(1)} & a_{55}^{(1)} \end{bmatrix}.$$

Objasnimo oznake koje koristimo za elemente matrice $A^{(1)}$. Općenito, elementi $A^{(1)}$ označeni su s $a_{ij}^{(1)}$, $1 \le i,\ j \le n$. Međutim, elementi prvog retka u $A^{(1)}$ jednaki su prvom retku u A, $a_{1j}^{(1)} = a_{1j}$, $1 \le j \le n$, pa smo to eksplicitno naznačili u zapisu matrice $A^{(1)}$.

Primijetimo da je transformaciju $A \mapsto A^{(1)}$ moguće izvesti samo ako je

$$a_{11} \neq 0. \tag{4.3.6}$$

[2]U primjeru 4.3.1 smo zbog jednostavnosti poništavali koeficijente jedan po jedan.

Također, lako se uvjerimo da je

$$(L^{(1)})^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & 0 & 1 & 0 & 0 \\ \frac{a_{41}}{a_{11}} & 0 & 0 & 1 & 0 \\ \frac{a_{51}}{a_{11}} & 0 & 0 & 0 & 1 \end{bmatrix},$$

te da iz $A = (L^{(1)})^{-1}A^{(1)}$ slijedi

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{a_{21}}{a_{11}} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22}^{(1)} \end{bmatrix}.$$

Jednostavno, dobili smo faktorizaciju vodeće 2×2 podmatrice od A. Uvjet za izvod ove faktorizacije bio je (4.3.6). Stavimo

$$\alpha_2 \equiv \det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22}^{(1)}.$$

Ako je $\alpha_2 \neq 0$, onda je i $a_{22}^{(1)} \neq 0$ pa je dobro definirana matrica

$$L^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ 0 & -\frac{a_{42}^{(1)}}{a_{22}^{(1)}} & 0 & 1 & 0 \\ 0 & -\frac{a_{52}^{(1)}}{a_{22}^{(1)}} & 0 & 0 & 1 \end{bmatrix} \quad \text{i njen inverz} \quad (L^{(2)})^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ 0 & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & 0 & 1 & 0 \\ 0 & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & 0 & 0 & 1 \end{bmatrix}.$$

Vrijedi

$$A^{(2)} \equiv L^{(2)}A^{(1)} = L^{(2)}L^{(1)}A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \end{bmatrix}. \tag{4.3.7}$$

Uočimo da oznake u relaciji (4.3.7) naglašavaju da je u matrici $A^{(2)} = \left[a_{ij}^{(2)}\right]_{i,j=1}^{n}$ prvi redak jednak prvom retku matrice A, a drugi redak jednak drugom retku matrice $A^{(1)}$. Ako sada u relaciji $A = (L^{(1)})^{-1}(L^{(2)})^{-1}A^{(2)}$ izračunamo produkt $(L^{(1)})^{-1}(L^{(2)})^{-1}$ dobivamo

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \dfrac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \dfrac{a_{31}}{a_{11}} & \dfrac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \dfrac{a_{41}}{a_{11}} & \dfrac{a_{42}^{(1)}}{a_{22}^{(1)}} & 0 & 1 & 0 \\ \dfrac{a_{51}}{a_{11}} & \dfrac{a_{52}^{(1)}}{a_{22}^{(1)}} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \end{bmatrix}, \tag{4.3.8}$$

odakle zaključujemo da vrijedi

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \dfrac{a_{21}}{a_{11}} & 1 & 0 \\ \dfrac{a_{31}}{a_{11}} & \dfrac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} \\ 0 & 0 & a_{33}^{(2)} \end{bmatrix}.$$

Dakle, ako je $a_{11} \neq 0$ i $\alpha_2 \neq 0$, onda smo dobili trokutastu faktorizaciju vodeće 3×3 podmatrice od A. Stavimo

$$\alpha_3 \equiv \det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}a_{22}^{(1)}a_{33}^{(2)}.$$

Ako je $\alpha_3 \neq 0$ onda je i $a_{33}^{(2)} \neq 0$ pa su dobro definirane matrice

$$L^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ 0 & 0 & -\dfrac{a_{53}^{(2)}}{a_{33}^{(2)}} & 0 & 1 \end{bmatrix}, \quad (L^{(3)})^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ 0 & 0 & \dfrac{a_{53}^{(2)}}{a_{33}^{(2)}} & 0 & 1 \end{bmatrix},$$

i vrijedi

$$A^{(3)} \equiv L^{(3)}A^{(2)} = L^{(3)}L^{(2)}L^{(1)}A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & a_{54}^{(3)} & a_{55}^{(3)} \end{bmatrix}.$$

Ako izračunamo produkt $(L^{(1)})^{-1}(L^{(2)})^{-1}(L^{(3)})^{-1}$, onda vidimo da vrijedi

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ \frac{a_{51}}{a_{11}} & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & \frac{a_{53}^{(2)}}{a_{33}^{(2)}} & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & a_{54}^{(3)} & a_{55}^{(3)} \end{bmatrix},$$

te da je

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} \end{bmatrix}.$$

Ponovo zaključujemo na isti način: definiramo

$$\alpha_4 \equiv \det \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = a_{11}a_{22}^{(1)}a_{33}^{(2)}a_{44}^{(3)}.$$

Ako je $\alpha_4 \neq 0$, onda je i $a_{44}^{(3)} \neq 0$, pa su dobro definirane matrice

$$L^{(4)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\dfrac{a_{54}^{(3)}}{a_{44}^{(3)}} & 1 \end{bmatrix}, \quad (L^{(4)})^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dfrac{a_{54}^{(3)}}{a_{44}^{(3)}} & 1 \end{bmatrix}. \tag{4.3.9}$$

Lako provjerimo da vrijedi

$$A^{(4)} \equiv L^{(4)}A^{(3)} = L^{(4)}L^{(3)}L^{(2)}L^{(1)}A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & 0 & a_{55}^{(4)} \end{bmatrix}.$$

te da je, nakon računanja produkta $(L^{(1)})^{-1}(L^{(2)})^{-1}(L^{(3)})^{-1}(L^{(4)})^{-1}$,

$$A = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \dfrac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \dfrac{a_{31}}{a_{11}} & \dfrac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \dfrac{a_{41}}{a_{11}} & \dfrac{a_{42}^{(1)}}{a_{22}^{(1)}} & \dfrac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ \dfrac{a_{51}}{a_{11}} & \dfrac{a_{52}^{(1)}}{a_{22}^{(1)}} & \dfrac{a_{53}^{(2)}}{a_{33}^{(2)}} & \dfrac{a_{54}^{(3)}}{a_{44}^{(3)}} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & 0 & a_{55}^{(4)} \end{bmatrix}}_{U}. \tag{4.3.10}$$

Vidimo da je izvedivost operacija koje su dovele do faktorizacije $A = LU$ ovisila o uvjetima

$$a_{11} \neq 0, \quad a_{22}^{(1)} \neq 0, \quad a_{33}^{(2)} \neq 0, \quad a_{44}^{(3)} \neq 0.$$

Također, uočili smo da su ti uvjeti osigurani ako su u matrici A determinante glavnih podmatrica dimenzija $1, 2, \ldots, n-1$ različite od nule. To je u našem primjeru značilo uvjete

$$\alpha_1 \equiv a_{11} \neq 0, \quad \alpha_2 \neq 0, \quad \alpha_3 \neq 0, \quad \alpha_4 \neq 0.$$

Brojeve a_{11}, $a_{22}^{(1)}$, $a_{33}^{(2)}$, $a_{44}^{(3)}$ zovemo **pivotni elementi** ili kratko **pivoti**. Brojevi α_1, α_2, α_3, α_4 su **glavne minore** matrice A.

Dakle, možemo zaključiti sljedeće:

- *Ako je prvih $n-1$ minora matrice A različito od nule, onda su i svi pivotni elementi različiti od nule i Gaussove eliminacije daju LU faktorizaciju matrice A.*

U tom slučaju sljedeći algoritam računa faktorizaciju $A = LU$.

Algoritam 4.3.3 *Računanje LU faktorizacije matrice A.*

$L = I$;
za $k = 1, \ldots, n-1$ {
  za $j = k+1, \ldots, n$ {
    $\ell_{jk} = \dfrac{a_{jk}^{(k-1)}}{a_{kk}^{(k-1)}}$;
    $a_{jk}^{(k)} = 0$; }
  za $j = k+1, \ldots, n$ {
    za $i = k+1, \ldots, n$ {
      $a_{ij}^{(k)} = a_{ij}^{(k-1)} - \ell_{ik} a_{kj}^{(k-1)}$; } } }
$U = A^{(n-1)} = \left[a_{ij}^{(n-1)} \right]$.

Sljedeći teorem i formalno dokazuje egzistenciju i jedinstvenost LU faktorizacije.

Teorem 4.3.1 *Neka je $A \in \mathbb{R}^{n\times n}$ i neka su determinante glavnih podmatrica $A(1:k, 1:k)$ različite od nule za $k = 1, 2, \ldots, n-1$. Tada postoji donjetrokutasta matrica L s jedinicama na dijagonali i gornjetrokutasta matrica U, tako da vrijedi $A = LU$. Ako faktorizacija $A = LU$ postoji i ako je još matrica A regularna, onda je faktorizacija jedinstvena: postoji točno jedna matrica L i točno jedna matrica U s ovim svojstvima. Tada je i*

$$\det(A) = \prod_{i=1}^{n} u_{ii}.$$

Dokaz. Dokažimo prvo jedinstvenost LU faktorizacije. Neka postoje dvije takve faktorizacije,

$$A = LU = L'U'.$$

Ako je A regularna onda su i L, U, L', U', također, regularne matrice pa vrijedi

$$L^{-1}L' = U(U')^{-1}.$$

U gornjoj relaciji imamo jednakost donjetrokutaste i gornjetrokutaste matrice, a one mogu biti jednake samo ako su obje dijagonalne matrice. Nadalje, L i L' po pretpostavci imaju jedinice na dijagonali, a zbog činjenice da se na dijagonali produkta donjetrokutastih matrica nalaze produkti dijagonalnih elemenata matrica koje se

množe, na dijagonali produkta $L^{-1}L'$ su jedinice. Dakle, $L^{-1}L' = I$, tj. $L = L'$. Tada je i $U = U'$.

Dokažimo sada egzistenciju LU faktorizacije. Induktivni dokaz je zapravo već skiciran u opisu računanja faktorizacije 5×5 matrice. Pogledajmo kako uvjeti teorema omogućuju prijelaz s $A^{(k)}$ na $A^{(k+1)}$, gdje je

$$A^{(k)} = L^{(k)} \cdots L^{(1)} A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & \vdots & a_{2,k+1}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \ddots & a_{33}^{(2)} & & \vdots & a_{3,k+1}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & & \ddots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & a_{kk}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ 0 & \cdots & & \cdots & 0 & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ \vdots & & & & \vdots & \vdots & & \vdots \\ 0 & \cdots & & \cdots & 0 & a_{n,k+1}^{(k)} & \cdots & a_{nn}^{(k)} \end{bmatrix}.$$

Kako je produkt $(L^{(k)} \cdots L^{(1)})^{-1}$ donjetrokutasta matrica s jedinicama na dijagonali, zaključujemo da je

$$\det A(1:k+1, 1:k+1) = a_{11} a_{22}^{(1)} a_{33}^{(2)} \cdots a_{kk}^{(k-1)} a_{k+1,k+1}^{(k)} \neq 0.$$

Odavde je i $a_{k+1,k+1}^{(k)} \neq 0$ pa možemo definirati matricu $L^{(k+1)}$ koja će poništiti elemente ispod dijagonale u $(k+1)$-om stupcu i dati $A^{(k+1)} = L^{(k+1)} A^{(k)}$. Jasno je da nakon konačnog broja koraka dobijemo matricu $A^{(n-1)}$ koja je gornjetrokutasta. ■

Napomena 4.3.1 *Primijetimo, ako je A regularna i ako ima LU faktorizaciju, onda su nužno i sve glavne podmatrice $A(1:k, 1:k)$ regularne. To slijedi iz činjenice da je*

$$\det A(1:k, 1:k) = \prod_{i=1}^{k} u_{ii}, \quad k = 1, \ldots, n.$$

4.3.4. LU faktorizacija s pivotiranjem

Jedan, očit problem, s LU faktorizacijom koju smo opisali u prethodnom odjeljku je da za njeno računanje, prema opisanom algoritmu, matrica A mora imati specijalnu strukturu: sve njene glavne podmatrice do uključivo reda $n-1$ moraju biti regularne. Sljedeći primjer ilustrira taj problem.

Primjer 4.3.2 *Neka je matrica sustava $Ax = b$ dana s*

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Ova matrica je regularna, $\det A = -1$*, pa sustav uvijek ima rješenje, ali A očito nema LU faktorizaciju, jer*

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \ell_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$$

povlači da je

$$\begin{aligned} 1 \cdot u_{11} &= 0 \\ 1 \cdot u_{12} &= 1 \\ \ell_{21} \cdot u_{11} &= 1 \\ \ell_{21} \cdot u_{12} + u_{22} &= 1. \end{aligned}$$

Iz prve jednadžbe odmah vidimo da mora biti $u_{11} = 0$, a iz treće slijedi da $\ell_{21} \cdot 0 = 1$, što je nemoguće.

S druge strane, matrica A reprezentira linearni sustav

$$\begin{aligned} 0x_1 + x_2 &= b_1 \\ x_1 + x_2 &= b_2 \end{aligned}$$

koji uvijek ima rješenje $x_1 = b_2 - b_1$, $x_2 = b_1$, i kojeg možemo ekvivalentno zapisati kao[3]

$$\begin{aligned} x_1 + x_2 &= b_2 \\ 0x_1 + x_2 &= b_1. \end{aligned}$$

Matrica ovog sustava je

$$A' = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

i očito ima jednostavnu LU faktorizaciju s $L = I$, $U = A'$. Vezu između A i A' zapisujemo matrično:

$$A' = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}}_{A}.$$

Matricu P zovemo **matrica permutacije** *ili jednostavno permutacija. Njeno djelovanje na matricu A je jednostavno permutiranje redaka.*

[3] Zamjena redoslijeda jednadžbi ne mijenja rješenje sustava.

Da bismo ilustrirali kako zamjenama redaka uvijek možemo dobiti LU faktorizaciju, vratimo se našem 5×5 primjeru i pogledajmo npr. relacije (4.3.7), (4.3.8):

$$A^{(2)} \equiv L^{(2)}A^{(1)} = L^{(2)}L^{(1)}A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \end{bmatrix},$$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \dfrac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \dfrac{a_{31}}{a_{11}} & \dfrac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \dfrac{a_{41}}{a_{11}} & \dfrac{a_{42}^{(1)}}{a_{22}^{(1)}} & 0 & 1 & 0 \\ \dfrac{a_{51}}{a_{11}} & \dfrac{a_{52}^{(1)}}{a_{22}^{(1)}} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \end{bmatrix}.$$

Neka je $a_{33}^{(2)} = 0$. Dakle, više ne možemo kao ranije definirati $L^{(3)}$. Pogledajmo elemente $a_{43}^{(2)}$ i $a_{53}^{(2)}$. Ako su oba jednaka nuli, onda možemo staviti $L^{(3)} = I$ i nastaviti dalje, jer je cilj transformacije $L^{(3)}$ poništiti $a_{43}^{(2)}$ i $a_{53}^{(2)}$. Ako su oni već jednaki nuli onda u ovom koraku ne treba ništa raditi pa je transformacija jednaka jediničnoj matrici. Neka je sada npr. $a_{53}^{(2)} \neq 0$. Ako definiramo matricu

$$P^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \text{onda je} \quad P^{(3)}A^{(2)} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \end{bmatrix}.$$

Sada možemo definirati matrice

$$L^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{a_{43}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \\ 0 & 0 & -\dfrac{a_{33}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \end{bmatrix}, \quad (L^{(3)})^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{a_{43}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \\ 0 & 0 & \dfrac{a_{33}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \end{bmatrix}$$

i postići

$$A^{(3)} \equiv L^{(3)}P^{(3)}A^{(2)} = L^{(3)}P^{(3)}L^{(2)}L^{(1)}A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & a_{54}^{(3)} & a_{55}^{(3)} \end{bmatrix}.$$

Primijetimo da je treći redak matrice $A^{(3)}$ jednak petom retku matrice $A^{(2)}$. Za sljedeći korak eliminacija provjeravamo vrijednost $a_{44}^{(3)}$. Ako je $a_{44}^{(3)} \neq 0$, postupamo kao i ranije, tj. definiramo matricu $L^{(4)}$ kao u relaciji (4.3.9). Ako je $a_{44}^{(3)} = a_{54}^{(3)} = 0$, onda možemo staviti $L^{(4)} = I$, jer je u tom slučaju $A^{(3)}$ već gornjetrokutasta. Neka je $a_{44}^{(3)} = 0$, ali $a_{54}^{(3)} \neq 0$, tako da $L^{(4)}$ nije definirana. Lako provjerimo da permutacijska matrica

$$P^{(4)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{daje} \quad P^{(4)}A^{(3)} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \\ 0 & 0 & 0 & a_{54}^{(3)} & a_{55}^{(3)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \end{bmatrix}.$$

Kako je po pretpostavci $a_{44}^{(3)} = 0$, možemo staviti $L^{(4)} = I$ i matrica $U = L^{(4)}P^{(4)}A^{(3)}$ je gornjetrokutasta. Sve zajedno, vrijedi relacija

$$U = L^{(4)}P^{(4)}L^{(3)}P^{(3)}L^{(2)}L^{(1)}A.$$

Vidjeli smo ranije da je množenje inverza trokutastih matrica $L^{(k)}$ jednostavno. Međutim, mi sada imamo permutacijske matrice između, pa ostaje istražiti kako

one djeluju na strukturu produkta. Primijetimo,

$$P^{(4)}L^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{a_{43}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \\ 0 & 0 & -\frac{a_{33}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{a_{33}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \\ 0 & 0 & -\frac{a_{43}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{a_{33}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \\ 0 & 0 & -\frac{a_{43}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \end{bmatrix}}_{\widetilde{L}^{(3)}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \widetilde{L}^{(3)}P^{(4)}.$$

Dakle, $P^{(4)}$ možemo **prebaciti s lijeve na desnu stranu od** $L^{(3)}$, ako u $L^{(3)}$ permutiramo elemente ispod dijagonale u trećem stupcu. Tako dobivena matrica $\widetilde{L}^{(3)}$ ima istu strukturu kao i $L^{(3)}$. Na isti način je $P^{(3)}L^{(2)}L^{(1)} = \widetilde{L}^{(2)}\widetilde{L}^{(1)}P^{(3)}$ i $P^{(4)}\widetilde{L}^{(2)}\widetilde{L}^{(1)} = \widetilde{\widetilde{L}}^{(2)}\widetilde{\widetilde{L}}^{(1)}P^{(4)}$ pa je

$$U = L^{(4)}P^{(4)}L^{(3)}P^{(3)}L^{(2)}L^{(1)}A = L^{(4)}\widetilde{L}^{(3)}\widetilde{\widetilde{L}}^{(2)}\widetilde{\widetilde{L}}^{(1)}P^{(4)}P^{(3)}A,$$

tj.

$$\underbrace{P^{(4)}P^{(3)}}_{P}A = \underbrace{(L^{(4)})^{-1}(\widetilde{L}^{(3)})^{-1}(\widetilde{\widetilde{L}}^{(2)})^{-1}(\widetilde{\widetilde{L}}^{(1)})^{-1}}_{L}U.$$

Produkt koji definira matricu L je iste strukture kao i ranije, dakle, imamo jednostavno slaganje odgovarajućih elemenata. Nadalje matrica

$$P = P^{(4)}P^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

je opet matrica permutacije.

Jasno je kako bi ovaj postupak izgledao općenito. Na kraju eliminacija bi vrijedilo

$$U = A^{(n-1)} = L^{(n-1)}P^{(n-1)}(\cdots(L^{(3)}P^{(3)}\underbrace{(L^{(2)}P^{(2)}\underbrace{(L^{(1)}P^{(1)}A)}_{A^{(1)}})}_{A^{(2)}})\cdots), \qquad (4.3.11)$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{A^{(3)}}$$

i $P = P^{(n-1)}P^{(n-2)}\cdots P^{(2)}P^{(1)}$, gdje neke od permutacija $P^{(k)}$ mogu biti jednake identitetama (jediničnim matricama).

Ilustrirajmo opisanu proceduru jednim numeričkim primjerom.

Primjer 4.3.3 *Neka je*

$$A = \begin{bmatrix} 1 & 1 & 4 & 1 \\ 2 & 1 & 1 & 6 \\ 5 & 1 & 1 & 0 \\ 1 & 4 & 1 & 3 \end{bmatrix}.$$

Najveći element u prvom stupcu od A je na poziciji $(3,1)$ – to znači da prvi pivot maksimiziramo ako zamijenimo prvi i treći redak od A. Tu zamjenu realizira permutacija $P^{(1)}$, gdje je

$$P^{(1)} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P^{(1)}A = \begin{bmatrix} 5 & 1 & 1 & 0 \\ 2 & 1 & 1 & 6 \\ 1 & 1 & 4 & 1 \\ 1 & 4 & 1 & 3 \end{bmatrix}.$$

Sada definiramo

$$L^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{2}{5} & 1 & 0 & 0 \\ -\frac{1}{5} & 0 & 1 & 0 \\ -\frac{1}{5} & 0 & 0 & 1 \end{bmatrix}, \quad \textit{pa je} \quad A^{(1)} = L^{(1)}P^{(1)}A = \begin{bmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{3}{5} & \frac{3}{5} & 6 \\ 0 & \frac{4}{5} & \frac{19}{5} & 1 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \end{bmatrix}.$$

Sljedeći pivot je maksimiziran permutacijom $P^{(2)}$, gdje je

$$P^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad P^{(2)}A^{(1)} = \begin{bmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \\ 0 & \frac{4}{5} & \frac{19}{5} & 1 \\ 0 & \frac{3}{5} & \frac{3}{5} & 6 \end{bmatrix}.$$

Sljedeći korak eliminacija glasi

$$L^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\frac{4}{19} & 1 & 0 \\ 0 & -\frac{3}{19} & 0 & 1 \end{bmatrix}, \quad A^{(2)} = L^{(2)}P^{(2)}A^{(1)} = \begin{bmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \\ 0 & 0 & \frac{69}{19} & \frac{7}{19} \\ 0 & 0 & \frac{9}{19} & \frac{105}{19} \end{bmatrix}.$$

Sljedeća permutacija je identiteta, $P^{(3)} = I$, pa u zadnjem koraku imamo

$$L^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{9}{69} & 1 \end{bmatrix}, \quad A^{(3)} = L^{(3)}P^{(3)}A^{(2)} = \begin{bmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \\ 0 & 0 & \frac{69}{19} & \frac{7}{19} \\ 0 & 0 & 0 & \frac{7182}{1311} \end{bmatrix}.$$

Sada primijetimo da je $A^{(3)} = L^{(3)}IL^{(2)}P^{(2)}L^{(1)}P^{(1)}A$, gdje je

$$P^{(2)}L^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{5} & 0 & 0 & 1 \\ -\frac{1}{5} & 0 & 1 & 0 \\ -\frac{2}{5} & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{5} & 1 & 0 & 0 \\ -\frac{1}{5} & 0 & 1 & 0 \\ -\frac{2}{5} & 0 & 0 & 1 \end{bmatrix} P^{(2)} = \widetilde{L}^{(1)}P^{(2)}.$$

Dakle, $U \equiv A^{(3)} = L^{(3)}L^{(2)}\widetilde{L}^{(1)}P^{(2)}P^{(1)}A$. Ako stavimo $P = P^{(2)}P^{(1)}$, onda vrijedi

$$PA = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 4 & 1 \\ 2 & 1 & 1 & 6 \\ 5 & 1 & 1 & 0 \\ 1 & 4 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 5 & 1 & 1 & 0 \\ 1 & 4 & 1 & 3 \\ 1 & 1 & 4 & 1 \\ 2 & 1 & 1 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{5} & 1 & 0 & 0 \\ \frac{1}{5} & 0 & 1 & 0 \\ \frac{2}{5} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{4}{19} & 1 & 0 \\ 0 & \frac{3}{19} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{9}{69} & 1 \end{bmatrix} U$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{5} & 1 & 0 & 0 \\ \frac{1}{5} & \frac{4}{19} & 1 & 0 \\ \frac{2}{5} & \frac{3}{19} & \frac{9}{69} & 1 \end{bmatrix} \begin{bmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \\ 0 & 0 & \frac{69}{19} & \frac{7}{19} \\ 0 & 0 & 0 & \frac{7182}{1311} \end{bmatrix}.$$

Dakle, možemo zaključiti sljedeće:

- *Za proizvoljnu $n \times n$ matricu A postoji permutacija P tako da Gaussove eliminacije daju LU faktorizaciju od PA, tj. $PA = LU$, gdje je L donjetrokutasta matrica s jedinicama na dijagonali, a U je gornjetrokutasta matrica. Permutaciju P možemo odabrati tako da su svi elementi matrice L po apsolutnoj vrijednosti najviše jednaki jedinici.*

  Preciznije, vrijedi sljedeći teorem.

Teorem 4.3.2 *Neka je $A \in \mathbb{R}^{n\times n}$ proizvoljna matrica. Tada postoji permutacija P takva da Gaussove eliminacije daju LU faktorizaciju $PA = LU$ matrice PA. Matrica $L = [\,\ell_{ij}\,]$ je donjetrokutasta s jedinicama na dijagonali, a U je gornjetrokutasta. Pri tome, ako je P produkt od p inverzija, vrijedi da je*

$$\det A = (-1)^p \prod_{i=1}^{n} u_{ii}.$$

Ako su matrice $P^{(k)}$ odabrane tako da vrijedi

$$|(P^{(k)}A^{(k-1)})_{kk}| = \max_{k\le j\le n} |(P^{(k)}A^{(k-1)})_{jk}|$$

onda je

$$\max_{1\le k\le n} \max_{1\le i,j\le n} |(L^{(k)})_{ij}| = \max_{1\le i,j\le n} |\ell_{ij}| = 1.$$

U tom slučaju faktorizaciju $PA = LU$ zovemo LU faktorizacijom s (standardnim) pivotiranjem redaka.

Dokaz. Na početku, primijetimo da za matricu

$$L^{(k)} = \begin{bmatrix} I_k & 0 \\ 0 \quad v & I_{n-k} \end{bmatrix}, \quad v = \begin{bmatrix} \ell^{(k)}_{k+1,k} \\ \vdots \\ \ell^{(k)}_{nk} \end{bmatrix}$$

i permutaciju $\Pi \in \mathcal{S}_n$ oblika

$$\Pi = \begin{bmatrix} I_k & 0 \\ 0 & \widehat{\Pi}_{n-k} \end{bmatrix}, \quad \widehat{\Pi} \in \mathcal{S}_{n-k}$$

vrijedi

$$\Pi L^{(k)} = \begin{bmatrix} I_k & 0 \\ 0 & \widehat{\Pi}_{n-k} v \quad \widehat{\Pi}_{n-k} \end{bmatrix} = \begin{bmatrix} I_k & 0 \\ 0 & \widehat{\Pi}_{n-k} v \quad I_{n-k} \end{bmatrix} \Pi = \widetilde{L}^{(k)} \Pi.$$

Nadalje, svaka permutacija Π oblika

$$\Pi = \begin{bmatrix} I_m & 0 \\ 0 & \widehat{\Pi}_{n-m} \end{bmatrix}, \quad m > k, \quad \widehat{\Pi} \in \mathcal{S}_{n-m}$$

je trivijalno oblika i

$$\Pi = \begin{bmatrix} I_k & 0 \\ 0 & \widetilde{\Pi}_{n-k} \end{bmatrix}, \quad \widetilde{\Pi}_{n-k} = \begin{bmatrix} I_{m-k} & 0 \\ 0 & \widehat{\Pi}_{n-m} \end{bmatrix} \in \mathcal{S}_{n-k},$$

pa je množenje analogno slučaju $m = k$. Kratko kažemo da "Π prolazi kroz $L^{(k)}$".

Nadalje, jasno je da u svakom koraku možemo odrediti permutaciju $P^{(k)}$ tako da postoji donjetrokutasta transformacija $L^{(k)}$ s jedinicama na dijagonali za koju $L^{(k)} P^{(k)} A^{(k-1)}$ ima sve nule ispod dijagonale u k-tom stupcu.

Dakle, kao u relaciji (4.3.11), možemo postići da je $U = A^{(n-1)}$ gornjetrokutasta matrica. U produktu

$$U = L^{(n-1)} P^{(n-1)} L^{(n-2)} P^{(n-2)} L^{(n-3)} P^{(n-3)} L^{(n-4)} P^{(n-4)} \cdots L^{(2)} P^{(2)} L^{(1)} P^{(1)} A$$

je $P^{(k+1)}$ oblika

$$P^{(k+1)} = \begin{bmatrix} I_k & 0 \\ 0 & \widehat{\Pi}_{n-k} \end{bmatrix}, \quad \widehat{\Pi} \in \mathcal{S}_{n-k},$$

što znači da $P^{(n-1)}$ prolazi kroz $L^{(n-2)}$, produkt $P^{(n-1)} P^{(n-2)}$ prolazi kroz $L^{(n-3)}$, produkt $P^{(n-1)} P^{(n-2)} P^{(n-3)}$ prolazi kroz $L^{(n-4)}$, itd.

Ako stavimo $P = P^{(n-1)} P^{(n-2)} \cdots P^{(2)} P^{(1)}$, onda je

$$U = \widetilde{L}^{(n-1)} \widetilde{L}^{(n-2)} \cdots \widetilde{L}^{(2)} \widetilde{L}^{(1)} P A,$$

odakle kao i ranije dobijemo $PA = LU$. Jasno je da strategija odabira permutacija iz iskaza teorema osigurava da su svi elementi od L po apsolutnoj vrijednosti najviše jednaki jedinici. ■

4.4. Numerička svojstva Gaussovih eliminacija

U prethodnim odjeljcima Gaussovim smo se eliminacijama bavili u okvirima linearne algebre. Preciznije, nismo razmatrali praktične detalje realizacije izvedenih algoritama na računalu. Zapravo, termin **praktični detalji** bi trebalo čitati kao **problemi**. Zašto?

Računalo je ograničen, konačan stroj. Imamo ograničenu količinu memorijskog prostora u kojem možemo držati polazne podatke, međurezultate i rezultate računanja[1]. Umjesto skupa realnih brojeva $\mathbb{R}$ imamo njegovu aproksimaciju pomoću konačno mnogo prikazivih brojeva (realni brojevi koje računalo koristi su zapravo konačan skup razlomaka) što znači da računske operacije ne možemo izvršavati niti točno niti rezultat možemo po volji dobro aproksimirati.

Za one čitatelje koji nisu svladali osnove numeričkih operacija linearne algebre na računalu, kao i za one koji taj materijal žele ponoviti, osnovne činjenice su dane u dodatku u odjeljku 4.4.4.. Preporučamo da čitatelj svakako "baci pogled' na taj odjeljak prije nastavka čitanja ovog materijala.

Praktično je odvojeno analizirati LU faktorizaciju i rješenje trokutastog sustava. Počinjemo s LU faktorizacijom, gdje nas očekuje niz zanimljivih zaključaka.

4.4.1. Analiza LU faktorizacije. Važnost pivotiranja.

Prije nego prijeđemo na numeričku analizu algoritma, pogledajmo kako ga možemo implementirati na računalu s minimalnim korištenjem dodatnog memorijskog prostora. Prisjetimo se našeg 5×5 primjera i relacije (4.3.10):

$$A = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ \frac{a_{51}}{a_{11}} & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & \frac{a_{53}^{(2)}}{a_{33}^{(2)}} & \frac{a_{54}^{(3)}}{a_{44}^{(3)}} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & 0 & a_{55}^{(4)} \end{bmatrix}}_{U}.$$

Vidimo da je za spremanje svih elemenata matrica L i U dovoljno n^2 varijabli (lokacija u memoriji), dakle onoliko koliko zauzima originalna matrica A. Ako pažljivo pogledamo proces računanja LU faktorizacije, uočavamo da ga možemo izvesti tako da matrica U ostane zapisana u gornjem trokutu matrice A, a strogo donji trokut matrice L bude napisan na mjestu elemenata strogo donjeg trokuta polazne matrice A. Kako matrica L po definiciji ima jedinice na dijagonali, te elemente

[1] Svaka operacija zahtijeva izvjesno vrijeme izvršavanja pa je ukupno trajanje algoritma također važan faktor. U ovom odjeljku prvenstveno ćemo analizirati problem točnosti.

ne treba nigdje posebno zapisivati. Na taj način se elementi polazne matrice gube, a računanje možemo shvatiti kao promjenu sadržaja polja A koje sadrži matricu A:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} \quad \mapsto \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \frac{a_{21}}{a_{11}} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & a_{44}^{(3)} & a_{45}^{(3)} \\ \frac{a_{51}}{a_{11}} & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & \frac{a_{53}^{(2)}}{a_{33}^{(2)}} & \frac{a_{54}^{(3)}}{a_{44}^{(3)}} & a_{55}^{(4)} \end{bmatrix}.$$

Sve matrice $A^{(k)}$, $k = 1, 2, \ldots, n-1$ su pohranjene u istom $n \times n$ polju koje na početku sadrži matricu $A \equiv A^{(0)}$. Na ovaj način zapis algoritma 4.3.3 postaje još jednostavniji i elegantniji.

Algoritam 4.4.1 *Računanje LU faktorizacije matrice A bez dodatne memorije.*

za $k = 1, \ldots, n-1$ {
  za $j = k+1, \ldots, n$ {
    $A(j,k) = \dfrac{A(j,k)}{A(k,k)}$; }
  za $j = k+1, \ldots, n$ {
    za $i = k+1, \ldots, n$ {
      $A(i,j) = A(i,j) - A(i,k)A(k,j)$; } } }

Primijetimo da smo koristili oznake uobičajene u programskim jezicima – element matrice (dvodimenzionalnog polja) označili smo s $A(i,j)$. Isto tako, vidimo da konkretna realizacija algoritma na računalu uključuje dodatne trikove i modifikacije kako bi se što racionalnije koristili resursi računala (npr. memorija). Dodatnu pažnju zahtijeva izvođenje aritmetičkih operacija pri čemu ne možemo izbjeći greške zaokruživanja.

Analiza grešaka zaokruživanja je, ponekad, tehnički kompliciranа. Važno je uočiti da cilj takve analize nije jednostavno tehničko prebrojavanje svih grešaka zaokruživanja nego izvođenje složenijih i dubljih zaključaka o numeričkoj stabilnosti algoritma i o pouzdanosti korištenja dobivenih rezultata.

Da bismo dobili ideju o kvaliteti izračunate faktorizacije, analizirat ćemo primjer faktorizacije 4×4 matrice

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}.$$

Izračunate aproksimacije matrica $L = [\,\ell_{ij}\,]$ i $U = [\,u_{ij}\,]$ označiti ćemo s $\widetilde{L} = [\,\tilde{\ell}_{ij}\,]$ i $\widetilde{U} = [\,\tilde{u}_{ij}\,]$. Kao u opisu algoritam za računanje LU faktorizacije u odjeljku 4.3.3., koristit ćemo matrice $L^{(i)}$ i transformacije oblika $A^{(i)} = L^{(i)}A^{(i-1)}$, $i = 1, \ldots, n-1$. Izračunate aproksimacije označavamo s $\widetilde{L}^{(i)}$ i $\widetilde{A}^{(i)}$.

Primijetimo da je prvi redak matrice $\widetilde{U}$ jednak prvom retku polazne matrice A,

$$\widetilde{U} = \begin{bmatrix} \tilde{u}_{11} & \tilde{u}_{12} & \tilde{u}_{13} & \tilde{u}_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & 0 & \tilde{u}_{33} & \tilde{u}_{34} \\ 0 & 0 & 0 & \tilde{u}_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & 0 & \tilde{u}_{33} & \tilde{u}_{34} \\ 0 & 0 & 0 & \tilde{u}_{44} \end{bmatrix}, \quad \tilde{u}_{1j} = a_{1j}, \quad j = 1, \ldots, 4.$$

Sada umjesto matrica $L^{(1)}$ i $A^{(1)} = L^{(1)}A$ imamo izračunate matrice

$$\widetilde{L}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\tilde{\ell}_{21} & 1 & 0 & 0 \\ -\tilde{\ell}_{31} & 0 & 1 & 0 \\ -\tilde{\ell}_{41} & 0 & 0 & 1 \end{bmatrix}$$

$$\widetilde{A}^{(1)} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} \ominus \tilde{\ell}_{21} \odot \tilde{u}_{12} & a_{23} \ominus \tilde{\ell}_{21} \odot \tilde{u}_{13} & a_{24} \ominus \tilde{\ell}_{21} \odot \tilde{u}_{14} \\ 0 & a_{32} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{12} & a_{33} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{13} & a_{34} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{14} \\ 0 & a_{42} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{12} & a_{43} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{13} & a_{44} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{14} \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{u}_{11} & \tilde{u}_{12} & \tilde{u}_{13} & \tilde{u}_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & \star & \star & \star \\ 0 & \star & \star & \star \end{bmatrix}, \quad \tilde{u}_{2j} = a_{2j} \ominus \tilde{\ell}_{21} \odot \tilde{u}_{1j}, \quad j = 2, 3, 4.$$

Ovdje smo sa $\star$ označili one elemente koje ćemo mijenjati u sljedećem koraku. Primijetimo da su u prva dva retka matrice $\widetilde{A}^{(1)}$ već izračunata prva dva retka matrice $\widetilde{U}$. U sljedećem koraku računamo

$$\widetilde{L}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\tilde{\ell}_{32} & 1 & 0 \\ 0 & -\tilde{\ell}_{42} & 0 & 1 \end{bmatrix}$$

$$\widetilde{A}^{(2)} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & 0 & (a_{33} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{13}) \ominus \tilde{\ell}_{32} \odot \tilde{u}_{23} & (a_{34} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{14}) \ominus \tilde{\ell}_{32} \odot \tilde{u}_{24} \\ 0 & 0 & (a_{43} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{14}) \ominus \tilde{\ell}_{42} \odot \tilde{u}_{23} & (a_{44} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{14}) \ominus \tilde{\ell}_{42} \odot \tilde{u}_{24} \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{u}_{11} & \tilde{u}_{12} & \tilde{u}_{13} & \tilde{u}_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & 0 & \tilde{u}_{33} & \tilde{u}_{34} \\ 0 & 0 & \star & \star \end{bmatrix}, \quad \tilde{u}_{3j} = (a_{3j} \ominus \tilde{\ell}_{31}\tilde{u}_{1j}) \ominus \tilde{\ell}_{32} \odot \tilde{u}_{2j}, \quad j = 3, 4.$$

I, u zadnjem koraku je ostala transformacija

$$\tilde{L}^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\tilde{\ell}_{43} & 1 \end{bmatrix},$$

koja primjenom na $\widetilde{A}^{(2)}$, daje i preostali element matrice $\widetilde{U}$,

$$\tilde{u}_{44} = ((a_{44} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{14}) \ominus \tilde{\ell}_{42} \odot \tilde{u}_{24}) \ominus \tilde{\ell}_{43} \odot \tilde{u}_{34}.$$

Uočavamo da se elementi u_{ij} računaju prema formuli

$$u_{ij} = a_{ij} - \sum_{m=1}^{i-1} \ell_{im} u_{mj}, \quad 2 \le i \le n, \quad i \le j \le n,$$

pri čemu je $u_{1j} = a_{1j}$, $1 \le j \le n$. Ovu formulu je lako provjeriti raspisivanjem produkta $A = LU$ po elementima. U našem algoritmu, zbog grešaka zaokruživanja, vrijedi

$$\tilde{u}_{ij} = (\cdots((a_{ij} \ominus \tilde{\ell}_{i1} \odot \tilde{u}_{1j}) \ominus \tilde{\ell}_{i2} \odot \tilde{u}_{2j}) \ominus \cdots) \ominus \tilde{\ell}_{i,i-1} \odot \tilde{u}_{i-1,j}. \tag{4.4.1}$$

Formula (4.4.1) je samo specijalan slučaj računanja općenitog izraza oblika

$$s = v_1 w_1 \pm v_2 w_2 \pm v_3 w_3 \pm \cdots \pm v_p w_p,$$

pri čemu se koristi algoritam

$\tilde{u}_{ij} = a_{ij}$;
za $m = 1, \ldots, i-1$ {
  $\tilde{u}_{ij} = \tilde{u}_{ij} \ominus \tilde{\ell}_{im} \odot \tilde{u}_{mj}$; }

Korištenjem propozicije 4.4.3, zaključujemo da postoje ξ_{ij}, ζ_{ijm} takvi da je u (4.4.1)

$$\tilde{u}_{ij} = a_{ij}(1 + \xi_{ij}) - \sum_{m=1}^{i-1} \tilde{\ell}_{im}\tilde{u}_{mj}(1 + \zeta_{ijm}). \tag{4.4.2}$$

Pri tome je za sve i, j, m

$$|\xi_{ij}|, |\zeta_{ijm}| \leq \frac{n\varepsilon}{1-n\varepsilon}.$$

Relaciju (4.4.2) možemo pročitati i kao

$$a_{ij} = \sum_{m=1}^{i} \tilde{\ell}_{im}\tilde{u}_{mj} + \delta a_{ij}, \quad \delta a_{ij} = \sum_{m=1}^{i} \tilde{\ell}_{im}\tilde{u}_{mj}\zeta_{ijm} - \xi_{ij}a_{ij}, \tag{4.4.3}$$

gdje smo $\tilde{u}_{ij}$ napisali kao $\tilde{\ell}_{ii}\tilde{u}_{ij}(1+\zeta_{iji})$, uz $\tilde{\ell}_{ii} = 1$ i $\zeta_{iji} = 0$.

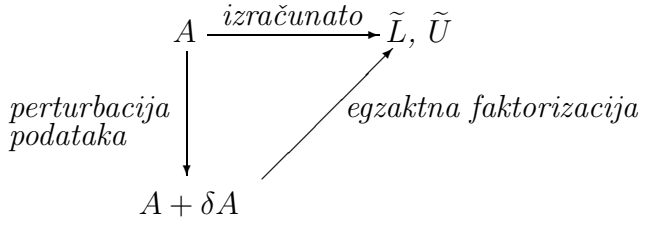
Time smo dokazali sljedeći teorem.

Teorem 4.4.1 *Neka je algoritam 4.4.1 primijenjen na matricu $A \in \mathbb{R}^{n\times n}$ i neka su uspješno izvršene sve njegove operacije. Ako su $\widetilde{L}$ i $\widetilde{U}$ izračunati trokutasti faktori, onda je*

$$\widetilde{L}\widetilde{U} = A + \delta A, \quad |\delta A| \leq \frac{n\varepsilon}{1-n\varepsilon}(|A| + |\widetilde{L}|\,|\widetilde{U}|) \leq \frac{2n\varepsilon}{1-2n\varepsilon}\,|\widetilde{L}|\,|\widetilde{U}|,$$

gdje prva nejednakost vrijedi za $n\varepsilon < 1$, a druga za $2n\varepsilon < 1$.

Napomena 4.4.1 *Rezultat teorema zaslužuje poseban komentar. Naša analiza nije dala odgovor na pitanje koliko su $\widetilde{L}$ i $\widetilde{U}$ daleko od točnih matrica L i U. Umjesto toga, dobili smo zaključak da $\widetilde{L}$ i $\widetilde{U}$ čine egzaktnu LU faktorizaciju matrice $A + \delta A$. Drugim riječima, ako bismo A promijenili u $A + \delta A$ i zatim uzeli egzaktnu faktorizaciju, dobili bismo upravo $\widetilde{L}$ i $\widetilde{U}$. Ovu situaciju možemo ilustrirati komutativnim dijagramom na slici 4.4.1. Dobiveni rezultat je u praksi od izuzetne važnosti, jer često je u primjenama nemoguće raditi s egzaktnim podacima – matrica A može biti rezultat mjerenja ili nekih prethodnih proračuna, dakle netočna. Ako je egzaktna (nepoznata) matrica $\widehat{A}$ i $A = \widehat{A} + \delta\widehat{A}$, onda je $\widetilde{L}\widetilde{U} = \widehat{A} + \delta\widehat{A} + \delta A$ i $LU = \widehat{A} + \delta\widehat{A}$. Ako su δA i $\delta\widehat{A}$ usporedivi po veličini, onda možemo u mnogim primjenama $\widetilde{L}$ i $\widetilde{U}$ smatrati jednako dobrim kao i L i U.*



Slika 4.4.1 Komutativni dijagram LU faktorizacije u aritmetici konačne preciznosti. Izračunati rezultat je ekvivalentan egzaktnom računu s promijenjenim polaznim podacima.

Iz prethodne analize jasno je da je δA mala ako produkt $|\widetilde{L}|\,|\widetilde{U}|$ nije prevelik u usporedbi s $|A|$. To na žalost nije osigurano u LU faktorizaciji. Sljedeći primjer pokazuje numeričku nestabilnost algoritma.

Primjer 4.4.1 *Neka je α mali parametar, $|\alpha| \ll 1$, i neka je matrica A definirana s*

$$A = \begin{bmatrix} \alpha & 1 \\ 1 & 1 \end{bmatrix}.$$

U egzaktnom računanju imamo

$$L^{(2,1)} = \begin{bmatrix} 1 & 0 \\ -\dfrac{1}{\alpha} & 1 \end{bmatrix}, \quad L^{(2,1)}A = \begin{bmatrix} \alpha & 1 \\ 0 & 1 - \dfrac{1}{\alpha} \end{bmatrix},$$

pa je LU faktorizacija matrice A dana s

$$\underbrace{\begin{bmatrix} \alpha & 1 \\ 1 & 1 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 \\ \dfrac{1}{\alpha} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} \alpha & 1 \\ 0 & 1 - \dfrac{1}{\alpha} \end{bmatrix}}_{U}.$$

Pretpostavimo sada da ovaj račun provodimo na računalu u aritmetici s 8 decimalnih znamenki, tj. točnosti $\varepsilon \approx 10^{-8}$. Neka je $|\alpha| < \varepsilon$, npr. neka je $\alpha = 10^{-10}$. Kako je problem jednostavan, vrijedi

$$\begin{aligned} \tilde{\ell}_{21} &= \ell_{21}(1+\epsilon_1), \quad |\epsilon_1| \le \varepsilon, \\ \tilde{u}_{11} &= u_{11}, \\ \tilde{u}_{12} &= u_{12}, \\ \tilde{u}_{22} &= 1 \ominus 1 \oslash \alpha = -1 \oslash \alpha = -\frac{1}{\alpha}\,(1+\epsilon_1). \end{aligned}$$

Primijetimo da je

$$\left|\frac{\tilde{u}_{22} - u_{22}}{u_{22}}\right| \le \frac{2\varepsilon}{1-\varepsilon}.$$

Dakle svi elementi matrica $\widetilde{L}$ i $\widetilde{U}$ izračunati su s malom relativnom pogreškom. Sjetimo se da je ovaj primjer najavljen kao primjer numeričke nestabilnosti procesa eliminacija, odnosno LU faktorizacije. Gdje je tu nestabilnost ako su svi izračunati elementi matrica $\widetilde{L}$ i $\widetilde{U}$ gotovo jednaki točnim vrijednostima? Odstupanje (relativna greška) je najviše reda veličine dvije greške zaokruživanja – gdje je onda problem?

Izračunajmo (egzaktno) $\widetilde{L}\widetilde{U}$:

$$\widetilde{L}\widetilde{U} = \begin{bmatrix} 1 & 0 \\ 1 \oslash \alpha & 1 \end{bmatrix} \begin{bmatrix} \alpha & 1 \\ 0 & -1 \oslash \alpha \end{bmatrix} = \begin{bmatrix} \alpha & 1 \\ 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha & 1 \\ 1 & 1 \end{bmatrix}}_{A} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}}_{\delta A}.$$

Primijetimo da δA ne možemo smatrati malom perturbacijom polazne matrice A – jedan od najvećih elemenata u matrici A, $a_{22} = 1$, je promijenjen u nulu. Ako bismo koristeći $\widetilde{L}$ i $\widetilde{U}$ pokušali riješiti linearni sustav $Ax = b$, zapravo bismo radili na sustavu $(A + \delta A)x = b$. Tek da dobijemo osjećaj kako katastrofalno loš rezultat možemo dobiti, pogledajmo linearne sustave

$$\begin{bmatrix} \alpha & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \begin{bmatrix} \alpha & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Njihova rješenja su

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \dfrac{-1}{\alpha - 1} \\ \dfrac{2\alpha - 1}{\alpha - 1} \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 - 2\alpha \end{bmatrix}.$$

Vidimo da se x_1 i $\tilde{x}_1$ potpuno razlikuju. Zaključujemo da Gaussove eliminacije mogu biti numerički nestabilne – dovoljna je jedna greška zaokruživanja "u krivo vrijeme na krivom mjestu" pa da dobiveni rezultat bude potpuno netočan.

Napomena 4.4.2 *I ovaj primjer zaslužuje komentar. Vidimo da katastrofalno velika greška nije uzrokovana akumuliranjem velikog broja grešaka zaokruživanja. Cijeli problem je u samo jednoj aritmetičkoj operaciji (pri računanju $\tilde{u}_{22}$) koja je zapravo izvedena jako točno, s malom greškom zaokruživanja.*

Cilj numeričke analize algoritma je da otkrije moguće uzroke nestabilnosti, objasni fenomene vezane za numeričku nestabilnost i ponudi rješenja za njihovo uklanjanje.

Nestabilnost ilustrirana primjerom u skladu je s teoremom 4.4.1. Naime, ako izračunamo $|\widetilde{L}|\,|\widetilde{U}|$ dobijemo

$$|\widetilde{L}|\,|\widetilde{U}| = \begin{bmatrix} |\alpha| & 1 \\ 1 + \epsilon & 2|1 \oslash \alpha| \end{bmatrix},$$

gdje je $1 + \epsilon = \alpha(1 \oslash \alpha)$, $|\epsilon| \leq \varepsilon$. Kako je na poziciji $(2, 2)$ u matrici $|\widetilde{L}|\,|\widetilde{U}|$ element koji je reda veličine $1/|\alpha| > 1/\varepsilon$, vidimo da nam teorem ne može garantirati mali δA.

Jasno nam je da je, zbog nenegativnosti matrica $|\widetilde{L}|$ i $|\widetilde{U}|$, mali produkt $|\widetilde{L}|\,|\widetilde{U}|$ moguć samo ako su elementi od $\widetilde{L}$ i $\widetilde{U}$ mali po apsolutnoj vrijednosti. Pogledajmo nastavak primjera 4.4.1.

Primjer 4.4.2 *Neka je A matrica iz primjera 4.4.1. Zamijenimo joj poredak redaka,*

$$A' = PA = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \alpha & 1 \end{bmatrix}.$$

LU faktorizacija matrice $A' = LU$ je

$$\begin{bmatrix} 1 & 1 \\ \alpha & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1-\alpha \end{bmatrix}.$$

Ako je $|\alpha| < \varepsilon$, onda su izračunate matrice

$$\widetilde{L} = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix}, \quad \widetilde{U} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

i vrijedi

$$\widetilde{L}\widetilde{U} = \begin{bmatrix} 1 & 1 \\ \alpha & 1+\alpha \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ \alpha & 1 \end{bmatrix}}_{A'} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & \alpha \end{bmatrix}}_{\delta A'}, \quad |\delta A'| \leq \varepsilon |A|.$$

Primijetimo i da je produkt

$$|\widetilde{L}|\,|\widetilde{U}| = \begin{bmatrix} 1 & 1 \\ |\alpha| & 1+\alpha \end{bmatrix}$$

po elementima istog reda veličine kao i $|A'|$. Dakle, u ovom primjeru je bilo dovoljno zamijeniti poredak redaka u A (redoslijed jednadžbi) pa da imamo garantirano dobru faktorizaciju u smislu da je $\widetilde{L}\widetilde{U} = A' + \delta A'$ s malom perturbacijom $\delta A'$.

Iz prethodnih primjera i diskusija jasno je da standardno pivotiranje redaka, koje osigurava da su u matrici L svi elementi po apsolutnoj vrijednosti najviše jednaki jedinici[2], doprinosi numeričkoj stabilnosti. Naime, lako se vidi da su tada i svi elementi matrice $|\widetilde{L}|$ manji ili jednaki od jedan. U tom slučaju veličina produkta $|\widetilde{L}|\,|\widetilde{U}|$ bitno ovisi o elementima matrice $\widetilde{U}$. S druge strane, elementi matrice $\widetilde{U}$ su dobiveni iz matrica $\widetilde{A}^{(k)}$, $k = 0, 1, \ldots, n-1$, pa je broj

$$\rho = \frac{\max\limits_{i,j,k} \tilde{a}_{ij}^{(k)}}{\max\limits_{ij} a_{ij}} \tag{4.4.4}$$

dobra mjera za relativni rast (u odnosu na A) elemenata u produktu $|\widetilde{L}|\,|\widetilde{U}|$. Broj ρ zovemo faktor rasta elemenata u LU faktorizaciji i definiran je bez obzira da li koristimo pivotiranje redaka.

Primijetimo da u analizi grešaka zaokruživanja pivotiranje ne predstavlja dodatnu tehničku poteškoću, pa odmah možemo iskazati sljedeći teorem.

Teorem 4.4.2 *Neka je LU faktorizacija $n \times n$ matrice A izračunata s pivotiranjem redaka u aritmetici s relativnom točnosti ε i neka su $\widetilde{L}$ i $\widetilde{U}$ dobivene aproksimacije za L i U. Ako je pri tome korištena permutacija P, onda je*

$$\widetilde{L}\widetilde{U} = P(A + \delta A), \quad |\delta A| \leq \frac{2n\varepsilon}{1-2n\varepsilon} P^T |\widetilde{L}|\,|\widetilde{U}|.$$

[2]Vidi teorem 4.3.2.

Specijalno je, bez obzira na pivotiranje,

$$\|\delta A\|_F \leq O(n^3)\varepsilon\rho\|A\|_F.$$

Dokaz. Nakon ponovnog čitanja dokaza teorema 4.3.2 bi trebalo biti jasno da permutacije prolaze kroz elementarne transformacije $\widetilde{L}^{(i)}$ neovisno o točnosti računanja (egzaktno ili do na greške zaokruživanja). Dakle, možemo zaključiti da čak i računanje faktorizacije s pivotiranjem na računalu odgovara računu bez pivotiranja ali s polaznom matricom $A' = PA$. Sada primjenom teorema 4.4.1 dobijemo da vrijedi

$$\widetilde{L}\widetilde{U} = A' + \delta A', \quad |\delta A'| \leq \frac{2n\varepsilon}{1-2n\varepsilon}\,|\widetilde{L}|\,|\widetilde{U}|.$$

Kako je $A' + \delta A' = P(A + P^T\delta A')$, stavljanjem $\delta A = P^T\delta A'$ dobivamo tvrdnju teorema. Primijetimo i da je, bez obzira da li pivotiramo retke ili ne,

$$\|\delta A\|_F \leq \frac{2n\varepsilon}{1-2n\varepsilon}\sqrt{\frac{n(n+1)}{2}}\sqrt{\frac{n(n+1)}{2}}\,\rho\|A\|_F.$$

■

Razlika u numeričkoj stabilnosti koju donosi pivotiranje redaka je bolje ponašanje parmetra ρ, tj. pivotiranjem možemo osigurati umjeren rast elemenata u toku LU faktorizacije. U primjeru 4.4.1 smo vidjeli da u LU faktorizaciji bez pivotiranja rast elemenata tokom faktorizacije može biti po volji velik. Sljedeća propozicija pokazuje da u slučaju pivotiranja redaka faktor ρ ima gornju ogradu koja je funkcija samo dimenzije problema.

Propozicija 4.4.1 *Ako LU faktorizaciju računamo s pivotiranjem redaka u aritmetici s maksimalnom greškom zaokruživanja ε, onda je*

$$\rho \leq 2^{n-1}(1+\varepsilon)^{2(n-1)}.$$

Specijalno, u slučaju egzaktnog računanja je $\rho \leq 2^{n-1}$.

Dokaz. Dokaz ostavljamo čitatelju za vježbu. ■

Primijetimo da je gornja ograda za ρ reda veličine 2^n, što brzo raste kao funkcija od n. Postoje primjeri na kojima se ta gornja ograda i dostiže. Ipak, iskustvo iz prakse govori da su takvi primjeri rijetki i da je LU faktorizacija s pivotiranjem redaka dobar algoritam za rješavanje sustava linearnih jednadžbi. Možemo zaključiti i preporučiti sljedeće:

- *Gaussove eliminacije, odnosno LU faktorizaciju, valja u praksi uvijek raditi s pivotiranjem redaka.*

4.4.2. Analiza numeričkog rješenja trokutastog sustava

Kako smo vidjeli u odjeljku 4.3.2., trokutaste sustave rješavamo jednostavnim i elegantnim supstitucijama unaprijed ili unazad. Ta jednostavnost se odražava i na dobra numerička svojstva supstitucija, kada ih provedemo na računalu. Sljedeća propozicija opisuje kvalitetu numerički izračunatog rješenja trokutastog sustava jednadžbi.

Propozicija 4.4.2 *Neka je T donjetrokutasta (gornjetrokutasta) matrica reda n i neka je sustav $Tv = d$ riješen supstitucijama unaprijed (unazad) kako je opisano u odjeljku 4.3.2.. Ako je $\tilde{v}$ rješenje dobiveno primjenom aritmetike računala preciznosti ε, onda postoji donjetrokutasta (gornjetrokutasta) matrica δT takva da vrijedi*

$$(T + \delta T)\tilde{v} = d, \quad |\delta T| \leq \eta_{\triangleright}|T|, \quad 0 \leq \eta_{\triangleright} \leq \frac{n\varepsilon}{1 - n\varepsilon}.$$

Dokaz. Dokaz zbog jednostavnosti provodimo samo za donjetrokutastu matricu T. Pretpostavljamo da se i-ta komponenta rješenja za $i > 1$ računa na sljedeći način:

$\tilde{v}_i = T_{i1} \odot \tilde{v}_1$;
za $j = 2, \ldots, i-1$ {
  $\tilde{v}_i = \tilde{v}_i \oplus T_{ij} \odot \tilde{v}_j$; }
$\tilde{v}_i = (d_i \ominus \tilde{v}_i) \oslash T_{ii}$.

Primjenom pravila aritmetike računala dobijemo[3]

$$\tilde{v}_1 = \frac{d_1}{T_{11}}(1 + \epsilon_1), \quad |\epsilon_1| \leq \varepsilon,$$

te za $i = 2, \ldots, n$

$$\tilde{v}_i = \frac{d_i - \sum_{j=1}^{i-1} T_{ij}(1 + \zeta_j)\tilde{v}_j}{\frac{T_{ii}}{(1 + \epsilon_{1,i})(1 + \epsilon_{2,i})}}, \quad |\zeta_j| \leq \frac{(i-1)\varepsilon}{1 - (i-1)\varepsilon}, \quad |\epsilon_{1,i}| \leq \varepsilon, \quad |\epsilon_{2,i}| \leq \varepsilon.$$

■

Napomena 4.4.3 *Koliko god da je prethodni rezultat tehnički jednostavan, valja naglasiti da je zaključak o točnosti rješenja trokutastog sustava važan: izračunato rješenje zadovoljava trokutasti sustav sa matricom koeficijenata koja se po elementima malo razlikuje od zadane. Pojednostavljeno govoreći, ako radimo s $\varepsilon \approx 10^{-8}$ i ako je $n = 1000$, onda izračunati vektor $\tilde{v}$ zadovoljava $\widetilde{T}\tilde{v} = d$, gdje se elementi od $\widetilde{T}$ i T poklapaju u barem 5 decimalnih znamenki (od 8 na koliko je zadana matrica T).*

[3]Vidi odjeljak 4.4.4..

4.4.3. Točnost izračunatog rješenja sustava

Sada nam ostaje napraviti kompoziciju dobivenih rezultata i ocijeniti koliko točno možemo na računalu riješiti linearni sustav $Ax = b$ u kojem smo izračuali LU faktorizaciju $PA = LU$ i supstitucijama naprijed i unazad izračunali

$$x = U^{-1}(L^{-1}(Pb)).$$

Kako smo vidjeli u prethodnim odjeljcima, permutacija P se može (za potrebe analize) odmah primjeniti na polazne podatke i numeričku analizu možemo provesti bez pivotiranja. Kako to pojednostavljuje oznake, mi ćemo pretpostaviti da su na polazne podatke A i b već primijenjene zamjene redaka, tako da su formule jednostavno $A = LU$ i $x = U^{-1}(L^{-1}b))$.

Neka su $\widetilde{L}$ i $\widetilde{U}$ izračunate trokutaste matrice, gdje je $\widetilde{L}\widetilde{U} = A + \delta A$, kao u teoremu 4.4.1. Izračunato rješenje $\tilde{y}$ sustava $\widetilde{L}y = b$ zadovoljava (prema propoziciji 4.4.2)

$$(\widetilde{L} + \delta\widetilde{L})\tilde{y} = b, \quad |\delta\widetilde{L}| \leq \frac{n\varepsilon}{1 - n\varepsilon}\,|\widetilde{L}|.$$

Na isti način rješenje $\tilde{x}$ sustava $\widetilde{U}x = \tilde{y}$ zadovoljava

$$(\widetilde{U} + \delta\widetilde{U})\tilde{x} = \tilde{y}, \quad |\delta\widetilde{U}| \leq \frac{n\varepsilon}{1 - n\varepsilon}\,|\widetilde{U}|.$$

Dakle, $(\widetilde{L} + \delta\widetilde{L})(\widetilde{U} + \delta\widetilde{U})\tilde{x} = b$, tj.

$$\begin{aligned}(A + \delta A + E)\tilde{x} = b, \quad E &= \widetilde{L}\delta\widetilde{U} + \delta\widetilde{L}\widetilde{U} + \delta\widetilde{L}\delta\widetilde{U},\\ |E| &\leq |\widetilde{L}|\,|\delta\widetilde{U}| + |\delta\widetilde{L}|\,|\widetilde{U}| + |\delta\widetilde{L}|\,|\delta\widetilde{U}|.\end{aligned}$$

Time smo dokazali sljedeći teorem.

Teorem 4.4.3 *Neka je $\tilde{x}$ rješenje regularnog $n \times n$ sustava jednadžbi $Ax = b$, dobiveno Gaussovim eliminacijama s pivotiranjem redaka. Tada postoji perturbacija ΔA za koju vrijedi*

$$(A + \Delta A)\tilde{x} = b, \quad |\Delta A| \leq \frac{5n\varepsilon}{1 - 2n\varepsilon} P^T\,|\widetilde{L}|\,|\widetilde{U}|.$$

Ovdje je P permutacija koja realizira zamjenu redaka. Također pretpostavljamo da je $2n\varepsilon < 1$.

Na kraju ovog odjeljka, pokažimo kako cijeli algoritam na računalu možemo implementirati bez dodatne memorije. Kako smo prije vidjeli, LU faktorizaciju možemo napraviti tako da L i U smjestimo u matricu A. Sada još primijetimo da sustave $Ly = b$ i $Ux = y$ možemo riješiti tako da y i x u memoriju zapisujemo na mjesto vektora b. Tako dobijemo sljedeću implementaciju Gaussovih eliminacija:

Algoritam 4.4.2 *Rješavanje trokutastog sustava jednadžbi $Ax = b$ Gaussovim eliminacijama bez dodatne memorije.*

/* LU faktorizacija, $A = LU$ */
za $k = 1, \ldots, n-1$ {
  za $j = k+1, \ldots, n$ {
    $A(j,k) = \dfrac{A(j,k)}{A(k,k)}$; }
  za $j = k+1, \ldots, n$ {
    za $i = k+1, \ldots, n$ {
      $A(i,j) = A(i,j) - A(i,k)A(k,j)$; } } }
/* Rješavanje sustava $Ly = b$, y napisan na mjesto b. */
za $i = 2, \ldots, n$ {
  za $j = 1, \ldots, i-1$ {
    $b(i) = b(i) - A(i,j)b(j)$; } }
/* Rješavanje sustava $Ux = y$, x napisan na mjesto b. */
$b(n) = \dfrac{b(n)}{A(n,n)}$;
za $i = n-1, \ldots, 1$ {
  za $j = i+1, \ldots, n$ {
    $b(i) = b(i) - A(i,j)b(j)$; }
  $b(i) = b(i)/A(i,i)$; }

4.4.4. Dodatak: Osnove matričnog računa na računalu

Na računalu općenito ne možemo egzaktno izvršavati aritmetičke operacije. Rezultat zbrajanja, oduzimanja, množenja ili dijeljenja dva broja x i y prikaziva u računalu je po definiciji broj u računalu koji je najbliži egzaktnom zbroju, razlici, umnošku, odnosno kvocijentu x i y. Pri tome je relativna greška tako izvedenih operacija manja ili jednaka polovini najvećeg relativnog razmaka dva susjedna broja u računalu. Na primjer, u standardnoj jednostrukoj preciznosti (32-bitna reprezentacija) je relativni razmak susjednih brojeva omeđen s 2^{-23} pa je relativna greška aritmetičkih operacija najviše $\varepsilon \approx 10^{-8}$.

Navedena pravila za izvršavanje elementarnih aritmetičkih operacija lako za-

pišemo na sljedeći način:

$$\begin{aligned}
\text{zbrajanje:} \quad & x \oplus y = (x+y)(1+\epsilon_1), \quad |\epsilon_1| \le \varepsilon \\
\text{oduzimanje:} \quad & x \ominus y = (x-y)(1+\epsilon_2), \quad |\epsilon_2| \le \varepsilon \\
\text{množenje:} \quad & x \odot y = xy(1+\epsilon_3), \quad |\epsilon_3| \le \varepsilon \\
\text{dijeljenje:} \quad & x \oslash y = \frac{x}{y}(1+\epsilon_4), \quad |\epsilon_4| \le \varepsilon, \quad y \neq 0.
\end{aligned}$$

Ove relacije vrijede ako su rezultati navedenih operacija po apsolutnoj vrijednosti u intervalu (μ, M) gdje je npr. u 32-bitnoj reprezentaciji $\mu = 2^{-126} \approx 10^{-38}$ najmanji a $M = (1 + 2^{-1} + \cdots + 2^{-23})2^{127} \approx 10^{38}$ najveći normalizirani strojni broj. (U dvostrukoj preciznosti (64-bitna reprezentacija brojeva) je $\mu \approx 10^{-308}$, $M \approx 10^{308}$.) Analiza za rezultate izvan intervala (μ, M) je nešto složenija pa je nećemo raditi.

Kako na računalu izgledaju osnovne operacije linearne algebre? Lako se uvjerimo da je većina operacija (skalarni produkt, norma, linearne kombinacije, matrične operacije) bazirana na računanju

$$s = \sum_{i=1}^{m} x_i y_i,$$

gdje su x_i, y_i skalari (realni ili kompleksni brojevi ili njihove aproksimacije na računalu). Ako s računamo na standardan način, u računalu dobijemo, npr. s $m = 4$, izraz oblika

$$\tilde{s} = ((((x_1 \odot y_1) \oplus x_2 \odot y_2) \oplus x_3 \odot y_3) \oplus x_4 \odot y_4).$$

Sustavnom primjenom osnovnih svojstava aritmetike na stroju, lako se provjeri da je

$$\begin{aligned}
\tilde{s} &= (((x_1y_1(1+\epsilon_1) + x_2y_2(1+\epsilon_2))(1+\xi_2) + x_3y_3(1+\epsilon_3))(1+\xi_3) \\
&\qquad + x_4y_4(1+\epsilon_4))(1+\xi_4) \\
&= x_1y_1 \underbrace{(1+\epsilon_1)(1+\xi_2)(1+\xi_3)(1+\xi_4)}_{1+\zeta_1} + x_2y_2 \underbrace{(1+\epsilon_2)(1+\xi_2)(1+\xi_3)(1+\xi_4)}_{1+\zeta_2} \\
&\qquad + x_3y_3 \underbrace{(1+\epsilon_3)(1+\xi_3)(1+\xi_4)}_{1+\zeta_3} + x_4y_4 \underbrace{(1+\epsilon_4)(1+\xi_4)}_{1+\zeta_4} \\
&= \sum_{i=1}^{m=4} x_i y_i (1+\zeta_i),
\end{aligned}$$

gdje su sve vrijednosti ϵ_i, ξ_i po modulu manje od ε. Sada je jasno kako bi izgledala formula za proizvoljan broj od m sumanada. Primijetimo da $1+\zeta_k$ možemo ocijeniti s

$$1 - m\varepsilon \le 1 + \zeta_k \le \frac{1}{1-m\varepsilon}, \quad \text{tj. vrijedi} \quad |\zeta_k| \le \frac{m\varepsilon}{1-m\varepsilon}, \quad k = 1, 2, \ldots, m.$$

Propozicija 4.4.3 *Neka su $x_1, \ldots, x_m$, $y_1, \ldots, y_m$ brojevi u računalu, $m \geq 1$. Ako vrijednost*

$$s = \sum_{i=1}^{m} x_i y_i$$

računamo kao

$\tilde{s} = x_1 \odot y_1;$
za $i = 2, \ldots, m$ {
$\quad \tilde{s} = \tilde{s} \oplus x_i \odot y_i;$ }

onda postoje brojevi ζ_i, $i = 1, \ldots, m$, takvi da vrijedi

$$\tilde{s} = \sum_{i=1}^{m} x_i y_i (1 + \zeta_i), \quad |\zeta_i| \leq \frac{m\varepsilon}{1 - m\varepsilon}, \quad i = 1, 2, \ldots, m.$$

Dokaz. Dokaz smo već skicirali na primjeru $m = 4$. Očito je formalni dokaz najlakše izvesti matematičkom indukcijom po m. Dovoljno je primijetiti da je u koraku indukcije

$$\begin{aligned} \tilde{s} \oplus x_{m+1} \odot y_{m+1} &= (\tilde{s} + x_{m+1} y_{m+1}(1 + \omega_1))(1 + \omega_2) \\ &= \tilde{s}(1 + \omega_2) + x_{m+1} y_{m+1}(1 + \omega_1)(1 + \omega_2), \quad |\omega_1| \leq \varepsilon, \quad |\omega_2| \leq \varepsilon, \end{aligned}$$

te da je $1 - (m+1)\varepsilon \leq (1 - \varepsilon)(1 - m\varepsilon)$ i

$$\frac{1 + \varepsilon}{1 - m\varepsilon} \leq 1 + \frac{(m+1)\varepsilon}{1 - (m+1)\varepsilon}.$$

■

4.5. Numeričko rješavanje simetričnog sustava jednadžbi

U mnogim važnim primjenama, posebno u inženjerskim znanostima, linearni sustav jednadžbi je **simetričan**. To znači da je u sustavu $Ax = b$ matrica $A = [\, a_{ij} \,]_{i,j=1}^{n}$ simetrična, $A = A^T$, tj. za sve i, j je $a_{ij} = a_{ji}$.

Naravno, ako pametno iskoristimo simetriju, $A = A^T$, onda Gaussove eliminacije možemo provesti puno efikasnije. Pokazuje se da u slučaju simetrične matrice LU faktorizacija ima općeniti oblik

$$A = R^T J R,$$

gdje je R gornjetrokutasta matrica, a $J = \mathrm{diag}(\pm 1)$. Da bismo dobili ideju zašto je to tako, pogledajmo LU faktorizaciju $A = LU$ simetrične regularne matrice A. Iz

$A = LU$ slijedi da je $U^{-\tau}AU^{-1} = U^{-\tau}L$ istovremeno simetrična i gornjetrokutasta matrica. Dakle, $D = U^{-\tau}L$ je dijagonalna matrica, pa je $L = U^T D$, tj. $A = U^T DU$. Ako je

$$D = \operatorname{diag}(d_i)_{i=1}^n,$$

onda definiramo

$$|D|^{1/2} = \operatorname{diag}(|d_i|^{1/2}), \quad J = \operatorname{diag}(\operatorname{sign}(d_i)) \quad \text{i} \quad R = |D|^{1/2}U.$$

Slijedi da je $A = R^T JR$.

Postojanje LU faktorizacije je uvijek osigurano ako pivotiramo. Kako nam je cilj sačuvati simetriju, kod simetrične matrice ćemo istovremeno permutirati retke i stupce, tj. radit ćemo s PAP^T, gdje je P matrica permutacije.

4.5.1. Pozitivno definitni sustavi. Faktorizacija Choleskog

Kažemo da je simetrična $n \times n$ matrica A **pozitivno definitna** ako za sve $x \in \mathbb{R}^n$, $x \neq 0$, vrijedi

$$x^T Ax > 0.$$

Ako npr. uzmemo $x = e_i$, i-ti stupac jedinične matrice, onda je $a_{ii} = e_i^T Ae_i > 0$, tj. dijagonalni elementi pozitivno definitne matrice su uvijek pozitivni. Nadalje, ako je S bilo koja regularna matrica i $x \neq 0$, onda je i $y = Sx \neq 0$ i vrijedi

$$x^T(S^T AS)x = (Sx)^T A(Sx) = y^T Sy > 0,$$

pa zaključujemo da je i $S^T AS$ pozitivno definitna matrica.

Pozitivna definitnost osigurava egzistenciju LU faktorizacije bez pivotiranja. Pogledajmo kako. Ako A particioniramo tako da je

$$A = \begin{bmatrix} a_{11} & a^T \\ a & \widehat{A} \end{bmatrix}, \quad \widehat{A} \in \mathbb{R}^{(n-1)\times(n-1)},$$

onda je $a_{11} > 0$ i prvi korak eliminacija je

$$\begin{bmatrix} 1 & 0 \\ -\dfrac{a}{a_{11}} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{11} & a^T \\ a & \widehat{A} \end{bmatrix} = \begin{bmatrix} a_{11} & a^T \\ 0 & \widehat{A} - \dfrac{aa^T}{a_{11}} \end{bmatrix}.$$

Sada primijetimo da vrijedi i

$$\begin{bmatrix} 1 & 0 \\ -\dfrac{a}{a_{11}} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{11} & a^T \\ a & \widehat{A} \end{bmatrix} \begin{bmatrix} 1 & -\dfrac{a^T}{a_{11}} \\ 0 & I_{n-1} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 \\ 0 & \widehat{A} - \dfrac{aa^T}{a_{11}} \end{bmatrix},$$

pri čemu je dobivena matrica također pozitivno definitna. Sada se lako provjeri da je i matrica $\widehat{A} - \dfrac{aa^T}{a_{11}}$ pozitivno definitna, dakle, njen prvi dijagonalni element je strogo veći od nule pa se postupak eliminacija može nastaviti. Time je pokazana egzistencija faktorizacije $A = R^T R$ u kojoj je R gornjetrokutasta matrica. Faktorizaciju $A = R^T R$ zovemo **faktorizacija Choleskog** ili **trokutasta faktorizacija** simetrične pozitivno definitne matrice.

Elemente matrice R možemo izračunati jednostavnim nizom formula. Raspisivanjem relacije

$$A = R^T R = \begin{bmatrix} r_{11} & 0 & \cdots & 0 & 0 \\ r_{12} & r_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & r_{n-1,n-1} & 0 \\ r_{1n} & r_{2n} & \cdots & r_{n-1,n} & r_{nn} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & \cdots & r_{2n} \\ \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & \vdots & \ddots & r_{n-1,n-1} & r_{n-1,n} \\ 0 & 0 & \cdots & 0 & r_{nn} \end{bmatrix}$$

po komponentama, za $i \leq j$, dobijemo

$$a_{ij} = \sum_{k=1}^{i} r_{ki} r_{kj}$$

odakle direktno slijedi sljedeći algoritam za računanje matrice R.

Algoritam 4.5.1 *Računanje faktorizacije Choleskog simetrične pozitivno definitne matrice* $A \in \mathbb{R}^{n \times n}$.

za $i = 1, \ldots, n$ {

$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}$;

/* za $i = 1$, $r_{ii} = \sqrt{a_{ii}}$ */

za $j = i+1, \ldots, n$ {

$r_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj} \right) \Big/ r_{ii}$; } }

Promotrimo sada linearni sustav jednadžbi u kojem je $A \in \mathbb{R}^{n \times n}$ simetrična, pozitivno definitna matrica. Ako je $A = R^T R$ trokutasta faktorizacija, onda rješenje

$$x = A^{-1} b = R^{-1} R^{-\tau} b$$

možemo dobiti tako da prvo nađemo rješenje y sustava $R^T y = b$, a zatim riješimo sustav $Rx = y$. Kako je R gornjetrokutasta matrica, cijeli postupak je vrlo jednostavan i možemo ga zapisati na sljedeći način:

Algoritam 4.5.2 *Rješavanje linearnog sustava jednadžbi $Ax = b$ s pozitivno definitnom matricom $A \in \mathbb{R}^{n\times n}$.*

/* Trokutasta faktorizacija $A = R^T R$ */
za $i = 1, \ldots, n$ {
$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}$;
/* za $i = 1$, $r_{ii} = \sqrt{a_{ii}}$ */
za $j = i+1, \ldots, n$ {
$r_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}\right) \Big/ r_{ii}$; } }
/* Supstitucije naprijed za $R^T y = b$ */
$y_1 = \dfrac{b_1}{r_{11}}$;
za $i = 2, \ldots, n$ {
$y_i = \left(b_i - \sum_{j=1}^{i-1} r_{ji} y_j\right) \Big/ r_{ii}$; }
/* Supstitucije unazad za $Rx = y$ */
$x_n = \dfrac{y_n}{r_{nn}}$;
za $i = n-1, \ldots, 1$ {
$x_i = \left(y_i - \sum_{j=i+1}^{n} r_{ij} x_j\right) \Big/ r_{ii}$; }

Naš je cilj ispitati numerička svojstva algoritma 4.5.1, ako njegove operacije izvedemo na računalu u aritmetici konačne preciznosti ε.

Propozicija 4.5.1 *Neka je za zadanu $n \times n$ pozitivno definitnu matricu A algoritam 4.5.1 uspješno izvršio sve operacije u konačnoj aritmetici s greškom zaokruživanja ε. Ako je $\widetilde{R}$ izračunata aproksimacija matrice R, onda je $\widetilde{R}^T \widetilde{R} = A + \delta A$, gdje je $\delta A = [\,\delta a_{ij}\,]$ simetrična matrica i za sve $1 \le i, j \le n$ vrijedi*

$$|\delta a_{ij}| \le \eta_C \sqrt{a_{ii} a_{jj}}, \quad \eta_C = \frac{c(n)\varepsilon}{1 - 2c(n)\varepsilon}, \quad c(n) = \max\{3, n\}. \tag{4.5.1}$$

Dokaz. U i-tom koraku, $2 \le i \le n$, u algoritmu 4.5.1 vrijedi[1]

$$\tilde{r}_{ii} = (1+\varepsilon_2)\sqrt{(1+\varepsilon_1)\Big(a_{ii} - \sum_{k=1}^{i-1} \tilde{r}_{ki}^2 (1+\zeta_k)\Big)}, \tag{4.5.2}$$

[1]Ovdje koristimo pretpostavku da je algoritam uspješno završio sve operacije, tj. da je uspješno izračunao $\tilde{r}_{ii} > 0$ za sve i.

pa kvadriranjem i uz oznaku $1+\eta=(1+\varepsilon_1)(1+\varepsilon_2)^2$ dobijemo

$$\sum_{k=1}^{i}\tilde{r}_{ki}^2 = a_{ii} + \frac{\eta}{1+\eta}\,\tilde{r}_{ii}^2 - \sum_{k=1}^{i-1}\tilde{r}_{ki}^2\zeta_k = a_{ii}+\delta a_{ii}. \tag{4.5.3}$$

Ovu relaciju možemo zapisati i u obliku

$$\tilde{r}_{ii} = \sqrt{a_{ii}+\delta a_{ii} - \sum_{k=1}^{i-1}\tilde{r}_{ki}^2}. \tag{4.5.4}$$

Za $i=1$ očito je

$$\tilde{r}_{11} = \sqrt{(1+\varepsilon_2)^2 a_{11}} = \sqrt{a_{11}+\delta a_{11}}.$$

Definirajmo

$$\eta_i = \max\left\{\frac{|\eta|}{1+\eta}, \max_{1\le k\le i-1}|\zeta_k|\right\}.$$

Uz prethodnu definiciju, lako provjerimo da u relaciji (4.5.4) vrijedi

$$|\delta a_{ii}| \le \eta_i \sum_{k=1}^{i}\tilde{r}_{ki}^2 \le \frac{\eta_i}{1-\eta_i}\,a_{ii}, \quad \eta_i \le \frac{c(n)\varepsilon}{1-c(n)\varepsilon}.$$

Na isti način analiziramo računanje vrijednosti $\tilde{r}_{ij}$, $j>i$. Imamo

$$\tilde{r}_{ij} = (1+\epsilon_1)(1+\epsilon_2)\,\frac{a_{ij} - \sum_{k=1}^{i-1}\tilde{r}_{kj}\tilde{r}_{ki}\,(1+\zeta_k')}{\tilde{r}_{ii}}, \quad |\zeta_k'| \le \frac{(i-1)\varepsilon}{1-(i-1)\varepsilon},$$

pa stavljanjem $1+\tau=(1+\epsilon_1)(1+\epsilon_2)$ lako dolazimo do relacije

$$\sum_{k=1}^{i}\tilde{r}_{kj}\tilde{r}_{ki} = a_{ij} + \frac{\tau}{1+\tau}\,\tilde{r}_{ij}\tilde{r}_{ii} - \sum_{k=1}^{i-1}\tilde{r}_{kj}\tilde{r}_{ki}\zeta_k' = a_{ij}+\delta a_{ij} = a_{ji}+\delta a_{ji}. \tag{4.5.5}$$

Ovu relaciju možemo pročitati i na drugi način kao

$$\tilde{r}_{ij} = \frac{a_{ij}+\delta a_{ij} - \sum_{k=1}^{i-1}\tilde{r}_{kj}\tilde{r}_{ki}}{\tilde{r}_{ii}}. \tag{4.5.6}$$

Ako sada definiramo

$$\tau_i = \max\left\{\frac{|\tau|}{1+\tau}, \max_{1\le k\le i-1}|\zeta_k'|\right\},$$

možemo pisati da je

$$\begin{aligned}|\delta a_{ij}| &\le \tau_i \sum_{k=1}^{i}|\tilde{r}_{kj}|\,|\tilde{r}_{ki}| \le \tau_i \sqrt{\sum_{k=1}^{i}\tilde{r}_{kj}^2}\,\sqrt{\sum_{k=1}^{i}\tilde{r}_{ki}^2}\\ &\le \frac{\tau_i}{\sqrt{(1-\eta_i)(1-\eta_j)}}\,\sqrt{a_{ii}a_{jj}}, \quad \tau_i \le \frac{c(n)\varepsilon}{1-c(n)\varepsilon}.\end{aligned}$$

Konačno, primijetimo da relacije (4.5.3) i (4.5.5) pokazuju da je $\widetilde{R}^T\widetilde{R} = A + \delta A$, gdje je $\delta A = [\,\delta a_{ij}\,]_{i,j=1}^n$. Time je tvrdnja propozicije dokazana. ■

Sad nas zanimaju numerička svojstva algoritma 4.5.2. Ako je $\tilde{x}$ izračunata aproksimacija točnog rješenja $x = A^{-1}b$, što možemo reći o $\tilde{x}$? Iz propozicije 4.5.1 znamo da izračunata matrica $\widetilde{R}$ zadovoljava

$$\widetilde{R}^T\widetilde{R} = A + \delta A, \quad \max_{i,j} \frac{|\delta a_{ij}|}{\sqrt{a_{ii}a_{jj}}} \leq \eta_C.$$

U sljedećem koraku rješavamo dva trokutasta sustava, $\widetilde{R}^T y = b$ i $\widetilde{R}x = y$. Neka su $\tilde{y} \approx y$ i $\tilde{x} \approx x$, redom izračunati vektori rješenja. Prema propoziciji 4.4.2, postoje gornjetrokutaste matrice $\delta_1\widetilde{R}$ i $\delta_2\widetilde{R}$ tako da vrijedi

$$(\widetilde{R} + \delta_1\widetilde{R})^T\tilde{y} = b, \quad (\widetilde{R} + \delta_2\widetilde{R})\tilde{x} = \tilde{y}.$$

Pri tome je $|\delta_1\widetilde{R}| \leq \eta_\triangleright|\widetilde{R}|$ i $|\delta_2\widetilde{R}| \leq \eta_\triangleright|\widetilde{R}|$, gdje je

$$\eta_\triangleright = \frac{n\varepsilon}{1 - n\varepsilon}.$$

Nadalje, slijedi da izračunati $\tilde{x}$ zadovoljava

$$(\widetilde{R} + \delta_1\widetilde{R})^T(\widetilde{R} + \delta_2\widetilde{R})\tilde{x} = b, \quad \text{tj.} \quad (\widetilde{R}^T\widetilde{R} + \widetilde{R}^T\delta_2\widetilde{R} + (\delta_1\widetilde{R})^T\widetilde{R} + (\delta_1\widetilde{R})^T\delta_2\widetilde{R})\tilde{x} = b.$$

Definiramo li

$$E = \widetilde{R}^T\delta_2\widetilde{R} + (\delta_1\widetilde{R})^T\widetilde{R} + (\delta_1\widetilde{R})^T\delta_2\widetilde{R}$$

dobivamo da je

$$|E| \leq (2\eta_\triangleright + \eta_\triangleright^2)|\widetilde{R}|^T|\widetilde{R}|,$$

pa zaključujemo da $\tilde{x}$ zadovoljava sustav $(\widetilde{R}^T\widetilde{R} + E)\tilde{x} = b$, u kojem je E po elementima mala perturbacija matrice sustava $\widetilde{A} = \widetilde{R}^T\widetilde{R}$. Primijetimo i da je

$$|E_{ij}| \leq (2\eta_\triangleright + \eta_\triangleright^2)\sqrt{\tilde{a}_{ii}\tilde{a}_{jj}} \leq (2\eta_\triangleright + \eta_\triangleright^2)(1 + \eta_C)\sqrt{a_{ii}a_{jj}}, \quad 1 \leq i, j \leq n.$$

Kako je $\widetilde{A} = A + \delta A$, dobivamo vezu između polaznog sustava i izračunatog rješenja $\tilde{x}$:

$$(A + F)\tilde{x} = b, \quad \text{gdje je} \quad F = \delta A + E.$$

Pri tome su i E i δA ocijenjeni po elementima, na isti način i s približno istom gornjom ogradom. Možemo reći da smo riješili sustav s matricom $A + F$ koja je blizu zadane matrice A u smislu da je

$$\max_{i,j} \frac{|F_{ij}|}{\sqrt{a_{ii}a_{jj}}} \leq \xi, \quad \xi = \eta_C + (2\eta_\triangleright + \eta_\triangleright^2)(1 + \eta_C).$$

Da smo, na primjer, nakon faktorizacije $A + \delta A = \widetilde{R}^T\widetilde{R}$ trokutaste sustave po y i x riješili egzaktno, imali bismo $F = \delta A$ i $\xi = \eta_C$.

Ipak, nismo sasvim zadovoljni zaključkom. Zašto? Pažljivi čitatelj je sigurno već uočio da općenito E nije simetrična, što povlači da F nije simetrična, pa niti $A+F$ nije simetrična. Mi jesmo riješili sustav blizak zadanom, ali smo izgubili važnu strukturu polaznog sustava – simetriju. Simetrija matrice A sustava je posljedica strukture problema kojeg opisujemo sustavom $Ax = b$, pa nam je važno znati da $\tilde{x}$ odgovara rješenju bliskog problema, s istom strukturom, tj. simetrijom. To nas vodi do sljedećeg problema:

- *Ako je $(A + F)\tilde{x} = b$, postoji li simetrična perturbacija ΔA tako da je $(A + \Delta A)\tilde{x} = b$ i tako da za veličinu od ΔA postoje ocjene analogne onima za F?*

Sljedeća propozicija daje potvrdan odgovor na to pitanje.

Propozicija 4.5.2 *Neka je $(A+F)\tilde{x} = b$, gdje je A simetrična i pozitivno definitna i neka vrijedi*

$$\max_{i,j} \frac{|F_{ij}|}{\sqrt{a_{ii}a_{jj}}} \leq \xi.$$

Tada postoji simetrična perturbacija ΔA takva da je $(A + \Delta A)\tilde{x} = b$. Pri tome je

$$\max_{i\neq j} \frac{|\Delta a_{ij}|}{\sqrt{a_{ii}a_{jj}}} \leq \xi, \quad \max_{i} \frac{|\Delta a_{ii}|}{a_{ii}} \leq (2n-1)\xi.$$

Dokaz. Primijetimo da ΔA mora zadovoljavati jednadžbu $\Delta A\tilde{x} = F\tilde{x}$, koja daje n uvjeta za $n(n+1)/2$ stupnjeva slobode u ΔA. Stavimo

$$D = \operatorname{diag}(\sqrt{a_{ii}})_{i=1}^{n}$$

i promotrimo skalirani sustav $D^{-1}(A + F)D^{-1}D\tilde{x} = D^{-1}b$, tj.

$$(A_s + F_s)z = D^{-1}b, \quad A_s = D^{-1}AD^{-1}, \quad F_s = D^{-1}FD^{-1}.$$

Neka je P permutacija takva da vektor $\tilde{z} = P^T z$ zadovoljava

$$|\tilde{z}_1| \leq |\tilde{z}_2| \leq \cdots \leq |\tilde{z}_n|.$$

Gornji sustav zapisat ćemo u ekvivalentnom obliku

$$P^T(A_s + F_s)P\tilde{z} = P^T D^{-1}b, \quad \text{tj.} \quad (A_{s,p} + F_{s,p})\tilde{z} = P^T D^{-1}b,$$

gdje je $A_{s,p} = P^T A_s P$, $F_{s,p} = P^T F_s P$. Konstruirat ćemo simetričnu matricu $M = [\, m_{ij} \,]$ za koju je $M\tilde{z} = F_{s,p}\tilde{z}$. Definirajmo

$$m_{ij} = \begin{cases} (F_{s,p})_{ij} & \text{za } i < j, \\ (F_{s,p})_{ji} & \text{za } j < i \end{cases}$$

i odredimo dijagonalne elemente m_{ii} tako da je

$$m_{ii}\tilde{z}_i + \sum_{j\neq i} m_{ij}\tilde{z}_j = (F_{s,p})_{ii}\tilde{z}_i + \sum_{j\neq i}(F_{s,p})_{ij}\tilde{z}_j.$$

Iskoristimo li definiciju izvandijagonalnih elemenata matrice M, dobivamo relaciju

$$m_{ii}\tilde{z}_i = (F_{s,p})_{ii}\tilde{z}_i + \sum_{j=1}^{i-1}((F_{s,p})_{ij} - (F_{s,p})_{ji})\tilde{z}_j.$$

Ako je $\tilde{z}_i = 0$, stavimo $m_{ii} = 0$. Inače, definiramo

$$m_{11} = (F_{s,p})_{11}, \quad m_{ii} = (F_{s,p})_{ii} + \sum_{j=1}^{i-1}((F_{s,p})_{ij} - (F_{s,p})_{ji})\,\frac{\tilde{z}_j}{\tilde{z}_i}, \quad i = 2,\ldots,n.$$

Očito je $|m_{ii}| \leq (2i-1)\xi$, za sve i, te $\max\limits_{i\neq j}|m_{ij}| \leq \xi$. Po konstrukciji matrice A_s vrijedi

$$(A_{s,p} + M)\tilde{z} = P^T D^{-1} b \quad \text{ili, ekvivalentno,} \quad (A_s + PMP^T)z = D^{-1}b. \tag{4.5.7}$$

Pri tome je

$$\max_{i\neq j}|(PMP^T)_{ij}| \leq \xi$$

i

$$\max_{i}|(PMP^T)_{ii}| \leq (2n-1)\xi.$$

Skaliranjem sustava (4.5.7) dobijemo

$$(A + \Delta A)\tilde{x} = b, \quad \Delta A = D(PMP^T)D,$$

čime je dokaz završen. ■

Napomena 4.5.1 *Rezultat ovog odjeljka možemo sažeti u jednostavan zaključak. Pozitivno definitne sustave na računalu možemo riješiti s pogreškom koja je ekvivalentna malim promjenama koeficijenata u matrici sustava.*

4.6. Teorija perturbacija za linearne sustave

Iz prethodnih razmatranja jasno je da u primjenama rijetko možemo izračunati egzaktno rješenje sustava $Ax = b$, jer, i formiranje samog sustava (računanje koeficijenata sustava i desne strane) i njegovo rješavanje na računalu, uzrokuju greške. Analizom tih grešaka dolazimo do zaključka da izračunata aproksimacija rješenja $\tilde{x} = x + \delta x$ zadovoljava tzv. perturbirani sustav, $(A + \delta A)(x + \delta x) = b + \delta b$. Sada se

postavlja pitanje kako ocijeniti veličinu greške $\delta x = \tilde{x} - x$, ako je poznata informacija o veličini grešaka δA i δb.

U primjeru 4.4.1 vidjeli smo da čak i mala perturbacija δA može potpuno promijeniti rješenje x. Kako mi u realnoj primjeni ne znamo točno rješenje, cilj je otkriti kako možemo iz matrice A i vektora b dobiti ne samo (što je moguće bolju) aproksimaciju $\tilde{x}$, nego i procjenu koliko je ta aproksimacija dobra.

Na početku teorijske analize, promotrimo jednostavniji slučaj u kojem je $\delta b = 0$. Dakle, jedina perturbacija je ona koja A promijeni u $A+\delta A$. Zbog jednostavnosti promatrat ćemo samo (inače, važan) slučaj u kojem je matrica koeficijenata $A+\delta A$ i dalje regularna, pa je $x + \delta x$ jedinstveno određen.

Iz jednakosti $A + \delta A = A(I + A^{-1}\delta A)$ vidimo da je regularnost matrice $A + \delta A$ osigurana ako je $I + A^{-1}\delta A$ regularna. Uvjet pod kojim možemo garantirati regularnost matrice $I + A^{-1}\delta A$ daje sljedeća propozicija.

Propozicija 4.6.1 *Neka je X $n \times n$ matrica i $\|\cdot\|$ proizvoljna matrična norma. Ako je $\|X\| < 1$ onda je $I - X$ regularna matrica i*

$$(I - X)^{-1} = I + X + X^2 + \cdots = \sum_{k=0}^{\infty} X^k.$$

Dokaz. Primijetimo da za svaki prirodan broj m vrijedi

$$(I-X)(I+X+\cdots+X^m) = I+X+\cdots+X^m-X-\cdots-X^m-X^{m+1} = I-X^{m+1}.$$

Ako označimo $S_m = I + X + \cdots + X^m$, onda možemo pisati

$$(I - X)S_m = S_m(I - X) = I - X^{m+1},$$

tj.

$$S_m = (I - X)^{-1} - (I - X)^{-1}X^{m+1}.$$

Kako je, zbog $\|X\| < 1$,

$$\|(I - X)^{-1}X^{m+1}\| \leq \|(I - X)^{-1}\| \, \|X\|^{m+1} \longrightarrow 0, \quad \text{kada} \quad m \to \infty,$$

zaključujemo da je za dovoljno veliki indeks m matrica S_m po volji blizu matrici $(I - X)^{-1}$. ■

Korištenjem ove propozicije, regularnost matrice $A + \delta A$ obično osiguravamo tako da zahtijevamo da je $\|A^{-1}\delta A\| < 1$. Izbor matrične norme $\|\cdot\|$ ovisi o konkretnoj situaciji, npr. o tipu informacije o δA ili o teorijskim rezultatima koje koristimo u analizi. Neka je izabrana matrična norma jednaka Frobeniusovoj normi, $\|\cdot\| = \|\cdot\|_F$,

$$\|X\|_F = \sqrt{\sum_{i,j=1}^{n} |X_{ij}|^2} = \sqrt{\mathrm{tr}(X^T X)},$$

pri čemu je tr oznaka za trag matrice. Informacija o perturbaciji δA je važan faktor u razvoju analize. Neka je, na primjer, zadano (poznato) da je

$$\epsilon \equiv \frac{\|\delta A\|_F}{\|A\|_F} \ll 1$$

mali broj, tj. da je perturbacija **mala po normi**. Regularnost matrice $A + \delta A$ je osigurana ako je npr.

$$\|A^{-1}\|_F \|\delta A\|_F = \epsilon(\|A\|_F \|A^{-1}\|_F) < 1, \quad \text{tj.} \quad \epsilon < \frac{1}{\|A\|_F \|A^{-1}\|_F}.$$

Tada je $\|A^{-1}\delta A\|_F < 1$ i

$$(A + \delta A)^{-1} = (I + A^{-1}\delta A)^{-1} A^{-1},$$

pa $\tilde{x} = (A + \delta A)^{-1} b$ možemo pisati kao

$$\tilde{x} = (I + A^{-1}\delta A)^{-1} A^{-1} b = (I + A^{-1}\delta A)^{-1} x, \quad \text{tj.} \quad (I + A^{-1}\delta A)\tilde{x} = x.$$

Znači, $x - \tilde{x} = A^{-1}\delta A \tilde{x}$, pa je

$$\|x - \tilde{x}\|_2 \leq \|A^{-1}\delta A\|_F \|\tilde{x}\|_2.$$

Kako je $\|A^{-1}\delta A\|_F \leq \|A^{-1}\|_F \|\delta A\|_F$, dobivamo

$$\frac{\|x - \tilde{x}\|_2}{\|\tilde{x}\|_2} \leq \|A^{-1}\|_F \|A\|_F \frac{\|\delta A\|_F}{\|A\|_F} = \epsilon \|A^{-1}\|_F \|A\|_F. \tag{4.6.1}$$

Relacija (4.6.1) pokazuje da relativna greška u izračunatom rješenju $\tilde{x}$ može biti uvećana najviše s faktorom $\kappa_F(A) = \|A^{-1}\|_F \|A\|_F$ u odnosu na relativnu promjenu $\epsilon = \|\delta A\|_F / \|A\|_F$ u polaznoj matrici A.

4.6.1. Perturbacije male po normi

Sljedeći teorem daje potpuni opis greške ako je perturbacija dana po normi. Općenito ćemo promatrati i δA i δb, a mjerenja perturbacija će biti u proizvoljnoj vektorskoj normi $\|\cdot\|$ i pripadnoj matričnoj normi $\|\cdot\|$.

Teorem 4.6.1 *Neka je $Ax = b$, $(A + \delta A)(x + \delta x) = b + \delta b$, gdje je $\|\delta A\| \leq \epsilon \|A\|$, $\|\delta b\| \leq \epsilon \|b\|$. Ako je $\epsilon \|A^{-1}\| \, \|A\| < 1$, onda je*

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\epsilon}{1 - \epsilon \|A^{-1}\| \, \|A\|} \left(\frac{\|A^{-1}\| \, \|b\|}{\|x\|} + \|A^{-1}\| \, \|A\| \right) \leq 2 \frac{\epsilon \|A^{-1}\| \, \|A\|}{1 - \epsilon \|A^{-1}\| \, \|A\|}.$$

Pri tome postoje perturbacije δA i δb za koje je gornja nejednakost skoro dostignuta. Preciznije, postoje δA i δb takvi da je $\|\delta A\| = \epsilon \|A\|$, $\|\delta b\| = \epsilon \|b\|$, te

$$\frac{\|\delta x\|}{\|x\|} \geq \frac{\epsilon}{1 + \epsilon \|A^{-1}\| \, \|A\|} \left(\frac{\|A^{-1}\| \, \|b\|}{\|x\|} + \|A^{-1}\| \, \|A\| \right).$$

Dokaz. Iz pretpostavki teorema je

$$\delta x = A^{-1}\delta b - A^{-1}\delta A x - A^{-1}\delta A \delta x, \tag{4.6.2}$$

pa uzimanjem norme dobijemo

$$\|\delta x\| \leq \epsilon \|A^{-1}\| \, \|b\| + \epsilon \|A^{-1}\| \, \|A\| \, \|x\| + \epsilon \|A^{-1}\| \, \|A\| \, \|\delta x\|,$$

odakle, rješavanjem nejednakosti po $\|\delta x\|$ slijedi tvrdnja.

Da bismo konstruirali perturbacije za koje dobivena nejednakost skoro postaje jednakost, pogledajmo desnu stranu jednakosti (4.6.2). Vrijedi

$$\|\delta x\| \geq \|A^{-1}\delta b - A^{-1}\delta A x\| - \|A^{-1}\delta A \delta x\|.$$

Pokušajmo odrediti perturbacije δA, δb tako da vrijedi

$$\|A^{-1}\delta b - A^{-1}\delta A x\| = \epsilon \|A^{-1}\| \, \|b\| + \epsilon \|A^{-1}\| \, \|A\| \, \|x\|.$$

Dakle, δA i δb treba odabrati tako da je $\|\delta A\| \leq \epsilon \|A\|$, $\|\delta b\| \leq \epsilon \|b\|$,

$$\|A^{-1}\delta b\| = \epsilon \|A^{-1}\| \, \|b\|, \quad \|A^{-1}\delta A x\| = \epsilon \|A^{-1}\| \, \|A\| \, \|x\|,$$

te da je norma razlike $A^{-1}\delta b - A^{-1}\delta A x$ jednaka sumi normi vektora. Ovaj zadnji uvjet znači da $A^{-1}\delta b$ i $-A^{-1}\delta A x$ moraju biti kolinearni.

Ako je u jedinični vektor za kojeg je $\|A^{-1}u\| = \|A^{-1}\|$, onda $\delta b = \epsilon \|b\| u$ zadovoljava $\|\delta b\| = \epsilon \|b\|$ i $\|A^{-1}\delta b\| = \epsilon \|A^{-1}\| \, \|b\|$. Sada stavimo $\delta A = \epsilon \|A\| u v^T$, gdje je $v \in \mathbb{R}^n$ vektor kojeg ćemo odrediti da postignemo željene relacije:

(i) $\|\delta A\| = \epsilon \|A\| \max\limits_{z \neq 0} \dfrac{\|uv^T z\|}{\|z\|} = \epsilon \|A\| \max\limits_{z \neq 0} \dfrac{|v^T z|}{\|z\|}$ treba postati $\|\delta A\| = \epsilon \|A\|$;

(ii) $\|A^{-1}\delta A x\| = \epsilon \|A\| \, \|A^{-1}\| \, |v^T x|$ treba postati $\|A^{-1}\delta A x\| = \epsilon \|A\| \, \|A^{-1}\| \, \|x\|$.

Dakle, treba nam vektor v sa svojstvom da je, za sve z, $|v^T x z| \leq \|z\|$, te da je $|v^T x| = \|x\|$. Postojanje takvog vektora je rezultat Hahn–Banachovog teorema: takav vektor v postoji. Dakle, konstruirali smo δA i δb za koje je

$$\|\delta x\| \geq \epsilon \|A^{-1}\| \, \|b\| + \epsilon \|A^{-1}\| \, \|A\| \, \|x\| - \epsilon \|A^{-1}\| \, \|A\| \, \|\delta x\|,$$

čime je dokaz druge tvrdnje teorema završen. ■

Vidimo da teorem 4.6.1 iz zadane informacije o veličini perturbacija po normi ($\|\delta A\| \leq \epsilon \|A\|$, $\|\delta b\| \leq \epsilon \|b\|$) izvodi optimalnu[1] ocjenu iz koje se jasno vidi da je broj

$$\kappa(A) = \|A^{-1}\| \, \|A\| \tag{4.6.3}$$

odlučujući faktor u donošenju suda o numeričkoj kvaliteti izračunate aproksimacije $\tilde{x} = x + \delta x$ sustava $Ax = b$. Pravilo je jednostavno:

[1] Ovdje pod optimalnosti podrazumijevamo činjenicu da je gornju ogradu za $\|\delta x\|/\|x\|$ nemoguće bitno poboljšati.

- *Ako je relativna greška (po normi) u podacima najviše ϵ, onda se relativna greška u rješenju ponaša kao $\kappa(A)\epsilon$.*

4.6.2. Rezidualni vektor i stabilnost

Postoji još jedan jednostavan i koristan način kako prosuditi kvalitetu aproksimacije $\tilde{x}$ rješenja sustava $Ax = b$. Radi se o **rezidualnom vektoru**

$$r = b - A\tilde{x}. \tag{4.6.4}$$

Kako je $b - Ax = 0$, jasno je da bi za dobru aproksimaciju $\tilde{x}$ pripadni rezidual trebao biti mali po normi. Ako relaciju (4.6.4) pročitamo kako

$$A\tilde{x} = b - r, \quad \text{tj. kao} \quad A\tilde{x} = b + \delta b, \quad \text{gdje je} \quad \delta b = -r,$$

onda vidimo da je $\tilde{x}$ egzaktno rješenje sustava koji je dobiven iz originalnog sustava promjenom desne strane u $b + \delta b$. Ako je

$$\epsilon \equiv \frac{\|r\|}{\|b\|}$$

dovoljno mali broj, onda možemo $\tilde{x}$ prihvatiti kao zadovoljavajuću aproksimaciju. Zašto? Recimo da su naša polazna matrica A i vektor b rezultati mjerenja ili nekih prethodnih proračuna.

Teorem 4.6.2 *Neka je $\tilde{x}$ aproksimacija rješenja sustava $Ax = b$ i neka je*

$$\beta(\tilde{x}) = \min\{\epsilon \,:\, (A + \delta A)\tilde{x} = b + \delta b, \quad \|\delta A\| \le \epsilon\|A\|, \quad \|\delta b\| \le \epsilon\|b\|\}.$$

Tada je

$$\beta(\tilde{x}) = \frac{\|b - A\tilde{x}\|}{\|A\|\,\|\tilde{x}\| + \|b\|}.$$

Dokaz. Neka je $r = b - A\tilde{x}$ rezidualni vektor. Ako je $\epsilon \ge 0$ takav da postoje δA, δb takvi da je $\|\delta A\| \le \epsilon\|A\|$, $\|\delta b\| \le \epsilon\|b\|$, $(A + \delta A)\tilde{x} = b + \delta b$, onda vrijedi

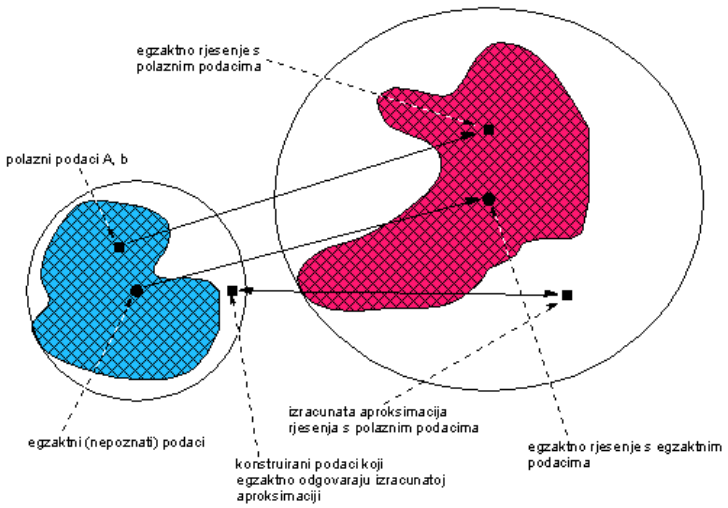
$$r = \delta A\tilde{x} - \delta b, \quad \text{pa je} \quad \|r\| \le \epsilon(\|A\|\,\|\tilde{x}\| + \|b\|), \quad \text{tj.} \quad \epsilon \ge \underline{\epsilon} \equiv \frac{\|b - A\tilde{x}\|}{\|A\|\,\|\tilde{x}\| + \|b\|}.$$

Stavimo sada

$$\delta b = -\frac{\|b\|}{\|A\|\,\|\tilde{x}\| + \|b\|} r.$$

Očito je $\|\delta b\| = \underline{\epsilon}\|b\|$. Odredimo δA tako da je $\|\delta A\| = \underline{\epsilon}\|A\|$ i

$$(A + \delta A)\tilde{x} = b - \frac{\|b\|}{\|A\|\,\|\tilde{x}\| + \|b\|} r, \quad \text{tj.} \quad \delta A\tilde{x} = \frac{\|A\|\,\|\tilde{x}\|}{\|A\|\,\|\tilde{x}\| + \|b\|}\, r.$$



Slika 4.6.1 Interpretacija izračunatog rješenja $\tilde{x}$ kao egzaktnog rješenja promijenjenog sustava jednadžbi.

Definirajmo

$$\delta A = \frac{\|A\|}{\|A\|\,\|\tilde{x}\| + \|b\|} r v^T,$$

gdje je v vektor sa svojstvima

$$v^T \tilde{x} = \|\tilde{x}\| \quad \text{i za sve } z \text{ vrijedi} \quad |v^T z| \leq \|z\|.$$

Takav vektor v postoji po Hahn–Banachovom teoremu i lako provjerimo da δA ima sva tražena svojstva. ■

4.6.3. Perturbacije po elementima

Najpreciznija ocjena perturbacije u matrici A je kada imamo informaciju o perturbaciji svakog njenog elementa, tj. svakog koeficijenta u sustavu jednadžbi. Ako je $A = [\, a_{ij} \,]_{i,j=1}^n$ i $A + \delta A = [\, a_{ij} + \delta a_{ij} \,]_{i,j=1}^n$, onda je takva ocjena dana relacijama

$$|\delta a_{ij}| \leq \varepsilon |a_{ij}|, \quad i, j = 1, 2, \ldots, n,$$

gdje je $0 \leq \varepsilon \ll 1$. Ove nejednakosti jednostavno zapisujemo kao $|\delta A| \leq \varepsilon |A|$, tj. apsolutne vrijednosti matrica i nejednakost među matricama shvaćamo po elementima. Na isti način pišemo $|\delta b| \leq \varepsilon |b|$. Primijetimo da ovakve perturbacije ($|\delta A| \leq \varepsilon |A|$, $|\delta b| \leq \varepsilon |b|$) čuvaju strukturu u smislu da nule u matrici A i vektoru b ostaju nepromijenjene. Nadalje, ove perturbacije su neizbježne pri pohranjivanju podataka u računalo.

Teorem 4.6.3 *Neka je* $Ax = b$ *i* $(A + \delta A)(x + \delta x) = b + \delta b$, *gdje je*

$$|\delta A| \leq \varepsilon |A|, \quad |\delta b| \leq \varepsilon |b|.$$

Uzmimo proizvoljnu apsolutnu vektorsku normu $\|\cdot\|$ *i njenu induciranu matričnu normu, također označenu s* $\|\cdot\|$. *Neka je* $\varepsilon \| |A^{-1}||A| \| < 1$. *Tada vrijedi*

$$\frac{\|\delta x\|}{\|x\|} \leq \varepsilon \frac{\| \, |A^{-1}| \, |A| \, |x| + |A^{-1}| \, |b| \, \|}{(1 - \varepsilon \| \, |A^{-1}| \, |A| \, \|)\|x\|}. \tag{4.6.5}$$

Nadalje, postoje perturbacije δA *i* δb *takve da je* $|\delta A| = \varepsilon |A|$, $|\delta b| = \varepsilon |b|$ *i da za rješenje* $x + \delta x = (A + \delta A)^{-1}(b + \delta b)$ *vrijedi*

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty} \geq \varepsilon \frac{\| \, |A^{-1}| \, |A| \, |x| + |A^{-1}| \, |b| \, \|_\infty}{(1 + \varepsilon \| \, |A^{-1}| \, |A| \, \|_\infty)\|x\|_\infty}. \tag{4.6.6}$$

Dokaz. Prije svega, uvjet $\varepsilon \| \, |A^{-1}| \, |A| \, \| < 1$ osigurava da je $A + \delta A$ regularna matrica, pa je $x + \delta x$ jedinstveno određen. Sada lako provjerimo da vrijedi

$$\delta x = -A^{-1}\delta A(x + \delta x) + A^{-1}\delta b, \tag{4.6.7}$$

pa primjenom nejednakosti trokuta (mnogokuta) dobijemo da je

$$\begin{aligned} |\delta x| &\leq |A^{-1}| \, |\delta A| \, |x| + |A^{-1}| \, |\delta A| \, |\delta x| + |A^{-1}| \, |\delta b| \\ &\leq \varepsilon |A^{-1}| \, |A| \, |x| + \varepsilon |A^{-1}| \, |A| \, |\delta x| + \varepsilon |A^{-1}| \, |b| \end{aligned}$$

pa je

$$\|\delta x\| \leq \varepsilon (\| \, |A^{-1}| \, |A| \, |x| + |A^{-1}| \, |b| \, \|) + \varepsilon \| \, |A^{-1}| \, |A| \, \| \, \|\delta x\|.$$

Neka je $m \in \{1, 2, \ldots, n\}$ odabran tako da je

$$(|A^{-1}| \, |A| \, |x| + |A^{-1}| \, |b|)_m = \| \, |A^{-1}| \, |A| \, |x| + |A^{-1}| \, |b| \, \|_\infty.$$

Definirajmo dijagonalne matrice

$$D_1 = \operatorname{diag}(\operatorname{sign}((A^{-1})_{mi}))_{i=1}^n, \quad D_2 = \operatorname{diag}(\operatorname{sign}(x_i))_{i=1}^n,$$

i perturbacije $\delta A = \varepsilon D_1 |A| D_2$, $\delta b = -\varepsilon D_1 |b|$. Sada lako računamo

$$\begin{aligned} (A^{-1}\delta A x - A^{-1}\delta b)_m &= \sum_{j=1}^n \sum_{i=1}^n (A^{-1})_{mj} (\delta A)_{ji} x_i - \sum_{j=1}^n (A^{-1})_{mj} \delta b_j \\ &= \varepsilon (|A^{-1}| \, |A| \, |x| + |A^{-1}| \, |b|)_m \\ &= \varepsilon \| \, |A^{-1}| \, |A| \, |x| + |A^{-1}| \, |b| \, \|_\infty. \end{aligned}$$

S druge strane, iz relacije (4.6.7) lako izvedemo da je

$$(A^{-1}\delta A x - A^{-1}\delta b)_m = -(\delta x + A^{-1}\delta A \delta x)_m,$$

pa je

$$\varepsilon \| \, |A^{-1}| \, |A| \, |x| + |A^{-1}| \, |b| \, \|_\infty \leq \|\delta x\|_\infty + \varepsilon \| \, |A^{-1}| \, |A| \, \|_\infty \|\delta x\|_\infty.$$

■

Korolar 4.6.1 *Relativni faktor osjetljivosti je dan relacijom*

$$\limsup_{\varepsilon\to 0}\left\{\frac{\dfrac{\|\delta x\|_\infty}{\|x\|_\infty}}{\varepsilon} \;:\; (A+\delta A)(x+\delta x)=b+\delta b,\quad |\delta A|\le\varepsilon|A|,\quad |\delta b|\le\varepsilon|b|\right\}$$
$$=\frac{\|\,|A^{-1}|\,|A|\,|x|+|A^{-1}|\,|b|\,\|_\infty}{\|x\|_\infty}.$$

Primijetimo da je

$$\|\,|A^{-1}|\,|A|\,|x|\,\|_\infty\le\|\,|A^{-1}|\,|A|\,|x|+|A^{-1}|\,|b|\,\|_\infty\le 2\|\,|A^{-1}|\,|A|\,|x|\,\|_\infty.$$

Zato kao koeficijent osjetljivosti možemo koristiti veličinu

$$\kappa_\infty(A,x)=\frac{\|\,|A^{-1}|\,|A|\,|x|\,\|_\infty}{\|x\|_\infty}.$$

Teorem 4.6.4 *Neka je $\tilde{x}$ izračunata aproksimacija rješenja sustava $Ax=b$ i neka je $r=b-A\tilde{x}$ izračunati rezidual. Vrijedi*

$$\min\{\varepsilon\ge 0\;:\;(A+\delta A)\tilde{x}=b+\delta b,\quad |\delta A|\le\varepsilon|A|,\quad |\delta b|\le\varepsilon|b|\}=\max_i\frac{|r_i|}{(|A|\,|\tilde{x}|+|b|)_i}.$$

Optimalna perturbacija polaznih podataka koja reproducira izračunato rješenje dana je s

$$\delta A=D_1|A|D_2,\quad \delta b=-D_1|b|,$$

gdje je

$$D_1=\operatorname{diag}\left(\frac{|r_i|}{(|A|\,|\tilde{x}|+|b|)_i}\right)_{i=1}^n,\quad D_2=\operatorname{diag}(\operatorname{sign}(\tilde{x}_i))_{i=1}^n.$$

Primjer 4.6.1 *U ovom primjeru istražujemo stabilnost Gaussovih eliminacija po elementima matrice. Neka su $\alpha\neq 0$ i $\beta\neq 0$ zadani i neka je*

$$A=\begin{bmatrix}\alpha & \beta\\ \alpha & 0\end{bmatrix},\quad b=\begin{bmatrix}1\\0\end{bmatrix},\quad x=A^{-1}b=\begin{bmatrix}0\\ \dfrac{1}{\beta}\end{bmatrix}.$$

Trokutasta LU faktorizacija matrice A je

$$\underbrace{\begin{bmatrix}\alpha & \beta\\ \alpha & 0\end{bmatrix}}_{A}=\underbrace{\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}}_{L}\underbrace{\begin{bmatrix}\alpha & \beta\\ 0 & -\beta\end{bmatrix}}_{U}.$$

Vektor $y = L^{-1}b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ *je izračunat bez greške. Ako pogledamo proces supstitucija unazad, egzaktne formule* $x_2 = 1/\beta$, $x_1 = 0$, *prelaze u*

$$\tilde{x}_2 = \frac{1}{\beta}(1+\xi_1),$$
$$\tilde{x}_1 = \frac{1}{\alpha}(1-\beta\tilde{x}_2(1+\xi_2))(1+\xi_3)(1+\xi_4) = \frac{1}{\alpha}(-\xi_1-\xi_2-\xi_1\xi_2)(1+\xi_3)(1+\xi_4),$$

gdje su ξ_1, ξ_2, ξ_3, ξ_4 *male greške reda veličine točnosti računala. Primijetimo da općenito ne možemo garantirati* $\tilde{x}_1 = 0$. *Ako je* β *broj u računalu takav da je* $\beta \odot (1 \oslash \beta) \neq 1$, *bit će* $\tilde{x}_1 \neq 0$.

Na primjer, u IEEE aritmetici je takav broj, npr.

$$\beta = 4.057062130620955e{-}001,$$

pri čemu je $\beta \odot (1 \oslash \beta) = 9.999999999999999e{-}001$. *(Čitatelju za vježbu ostavljamo da pokuša naći još takvih brojeva.)*

Pokušajmo sada izračunato rješenje $\tilde{x}_1$, $\tilde{x}_2$ *interpretirati kao egzaktno rješenje sustava* $(A+\delta A)\tilde{x} = b+\delta b$, *gdje su elementi matrice* δA *oblika* $a_{ij}(1+\epsilon_{ij})$, *a elementi od* δb *su* $b_i(1+\varepsilon_i)$. *Drugim riječima, treba odrediti* ϵ_{ij}, ε_i, $i, j = 1, \ldots, n$, *tako da vrijedi*

$$\begin{bmatrix} \alpha(1+\epsilon_{11}) & \beta(1+\epsilon_{12}) \\ \alpha(1+\epsilon_{21}) & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} 1+\varepsilon_1 \\ 0 \end{bmatrix}. \tag{4.6.8}$$

Ako je $\tilde{x}_1 \neq 0$, *jasno je da je za zadovoljavanje druge jednadžbe u gornjem sustavu nužno uzeti* $\epsilon_{21} = -1$, *što znači da element* a_{21} *treba promijeniti u nulu. Znači da imamo veliku promjenu elementa,* $\delta a_{21} = -a_{21}$, *tj.* $(\delta A)_{22} = -\alpha$, *pa je i*

$$\frac{\|\delta A\|_F}{\|A\|_F} \geq \frac{|\alpha|}{\sqrt{3}|\alpha|} > \frac{1}{2}.$$

Iz prethodnog primjera zaključujemo da bez obzira na točnost koju koristimo u računanju na računalu, općenito ne možemo garantirati da će izračunato rješenje $\tilde{x}$ biti točno rješenje sustava $\tilde{A}\tilde{x} = \tilde{b}$ u kojem su $\tilde{A}$ i $\tilde{b}$ nastali malim relativnim perturbacijama koeficijenata u A i b.

4.6.4. Dodatak: Udaljenost matrice do skupa singularnih matrica

Neka je A $n \times n$ regularna matrica. Pokušajmo joj naći najbližu singularnu matricu, pri čemu mjerimo u nekoj matričnoj normi $\|\cdot\|$. Neka je $A+\Delta A$ singularna. Iz $A + \Delta A = A(I + A^{-1}\Delta A)$ slijedi da je $I + A^{-1}\Delta A$ singularna. (I više, $A + \Delta A$

i $I + A^{-1}\Delta A$ su istog ranga.) Tada je nužno $\|A^{-1}\Delta A\| \geq 1$. Naime, $\|A^{-1}\Delta A\| < 1$ povlači (zbog propozicije 4.6.1) da je $I + A^{-1}\Delta A$ regularna. Dakle, $1 \leq \|A^{-1}\Delta A\| \leq \|A^{-1}\|\|\Delta A\|$, pa je

$$\|\Delta A\| \geq \frac{1}{\|A^{-1}\|}.$$

Time smo dokazali:

Propozicija 4.6.2 *Neka je A regularna matrica i $\|\cdot\|$ proizvoljna matrična norma. Tada je*

$$\inf_{\det(X)=0} \|A - X\| \geq \frac{1}{\|A^{-1}\|}.$$

Ako odaberemo $\|\cdot\| = \|\cdot\|_2$, onda se korištenjem SVD dekompozicije matrice A, $A = \sum_{i=1}^{n} \sigma_i u_i v_i^T$ može pokazati da je nejednakost u gornjoj propoziciji zapravo jednakost koja se postiže za $X = \sum_{i=1}^{n-1} \sigma_i u_i v_i^T$.

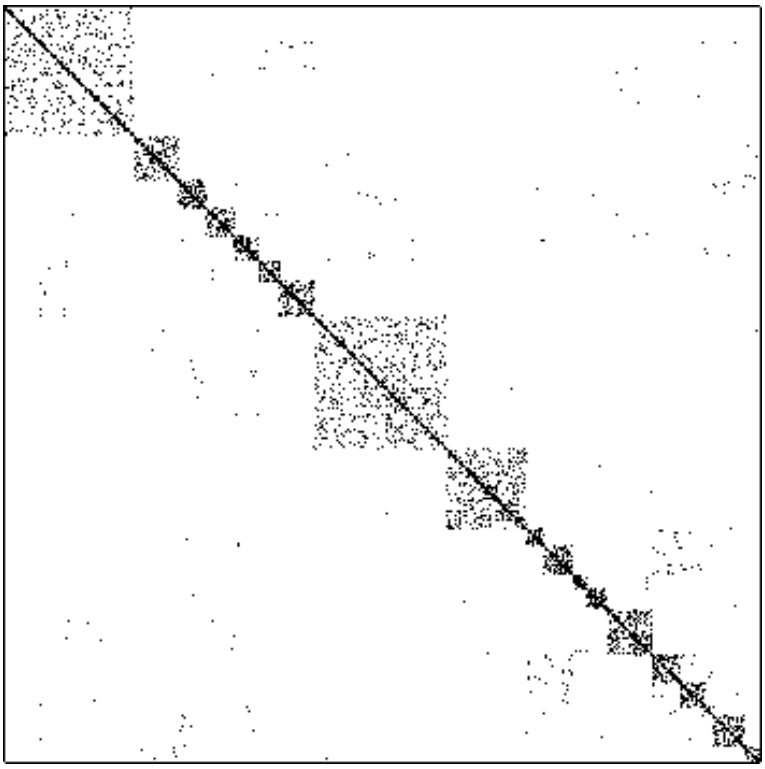
4.7. Iterativne metode

U prethodnim odjeljcima vidjeli smo da rješenje linearnog sustava $Ax = b$ općenito ne možemo izračunati apsolutno točno. Također, u nekim primjenama niti točno rješenje $x = A^{-1}b$ nije puno bolje od neke dovoljno dobre aproksimacije $\tilde{x}$, gdje obično $\tilde{x}$ zadovoljava sustav $(A + \delta A)\tilde{x} = b$, blizak polaznom. Dakle, Gaussove eliminacije, koje su jednostavno konačan niz formula koje vode rješenju sustava, ne garantiraju idealnu točnost.

Nadalje, u praksi moramo biti svjesni da je računalo omeđen ne samo u pitanju numeričke točnosti nego i u još dva važna aspekta: raspoloživom memorijskom prostoru i vremenu izvršavanja. Moderne primjene matematike zahtijevaju rješenja sustava velikih dimenzija, npr. $n > 10^5$. Lako se uvjeriti da je u takvim primjerima proces Gaussovih eliminacija često praktično neupotrebljiv. Jer, matrica dimenzije $n = 10^5$ zahtijeva $n^2 = 10^{10}$ lokacija u memoriji, svaka barem 4 bajta (veličina reprezentacije realnog broja u jednostrukoj preciznosti). Dakle, moguće je da samo spremanje matrice koeficijenata u memoriju računala predstavalja poteškoću – ponekad matricu držimo na vanjskoj, sporoj memoriji (datoteka na disku) i onda možemo dijelove matrice učitavati dio po dio u radnu memoriju. Kako u takvim uvjetima implementirati Gaussove eliminacije?

U puno važnih primjena matrica sustava A je velike dimenzije, ali je **rijetko popunjena**. To znači da je velika većina elemenata od A jednaka nuli, a elementi koji nisu nula obično su pravilno raspoređni po matrici ili čak imaju i pravilno raspoređene numeričke vrijednosti. Na primjer, matrica iz primjera 4.2.2 ima broj 2

na dijagonali i -1 na dvije sporedne dijagonale, a ostali elementi su nule. To znači da za takvu matricu A računanje produkta Av, gdje je $v \in \mathbb{R}^n$, zahtijeva $3n$ množenja i $2n$ zbrajanja – ukupno $5n$ operacija. (Ako A nema strukturu, onda općenito računanje Av zahtijeva $2n^2 - n$ operacija.) Još jedan primjer rijetko popunjene matrice je dan na slici 4.7.1. Ponovo uočavamo da je u svakom retku broj elemenata koji nisu nula unaprijed poznat (kao i pozicije gdje se ti elementi nalaze) i da je broj takvih elemenata puno manji od dimenzije n. To znači da je računanje produkta Av složenosti puno manje od $2n^2 - n$.



Slika 4.7.1 Rijetko popunjena matrica. Točkice pokazuju pozicije elemenata u matrici koji su različiti od nule.

Ponekad je matrica A velike dimenzije i gusto popunjena (svi ili velika većina elemenata je različita od nule), tako da jednostavno ne može stati u memoriju računala. Najbolje što možemo je učitavanje dijelova matrice iz vanjske u radnu memoriju. Ponekad je poznat način kako su generirani elementi matrice (npr. poznato da je $a_{ij} = f_{ij}(\cdots)$, gdje su $f_{ij}(\cdots)$ poznate funkcije nekih parametara) pa uvijek možemo generirati dijelove matrice. Moguće je i da je u konkretnoj aplikaciji jedino zadano kako A djeluje kao linearni operator – postoji potprogram koji za zadani v računa Av. Ako je to jedini način kako doći do matrice A, kako onda riješiti $Ax = b$?

Prethodna diskusija nas motivira da potražimo i drugačije pristupe za rješavanje linearnog sustava $Ax = b$. Primijetimo da ne moramo nužno težiti pronalaženju egzaktnog rješenja – umjesto toga želimo **dovoljno dobru** aproksimaciju $\tilde{x}$. Zato ima smisla pokušati konstruirati niz $x^{(0)}, x^{(1)}, \ldots, x^{(k)}, \ldots$ vektora iz $\mathbb{R}^n$ sa sljedećim svojstvima:

(i) za svaki k formula za računanje $x^{(k)}$ je jednostavna;

(ii) $x^{(k)}$ teži prema $x = A^{-1}b$ i za neki k (obično $k \ll n$) je $x^{(k)}$ prihvatljiva aproksimacija za x.

Nabrojana svojstva su za sada namjerno dana u nepreciznoj, ali lako razumljivoj formi. Detalji, koji ovise o konkretnom problemu i o konkretnom načinu konstruiranja niza $(x^{(k)})$, bit će dani malo kasnije.

4.7.1. Jacobijeva i Gauss–Seidelova metoda

Jacobijeva i Gauss–Seidelova metoda pripadaju klasičnim i najjednostavnijim iterativnim metodama za rješavanje linearnih sustava. Ideju Jacobijeve metode ilustrirat ćemo na primjeru 2×2 sustava

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \end{aligned}\right\}, \quad a_{11} \neq 0,\ a_{22} \neq 0.$$

Uočimo da rješenje $x = [\, x_1, x_2 \,]^T$ zadovoljava

$$\begin{aligned} x_1 &= \frac{1}{a_{11}} (b_1 - a_{12}x_2) \\ x_2 &= \frac{1}{a_{22}} (b_2 - a_{21}x_1). \end{aligned}$$

Te relacije motiviraju da neku približnu vrijednost rješenja $x^{(0)} = \left[\, x_1^{(0)}, x_2^{(0)} \,\right]^T$ korigiramo pomoću formula

$$\begin{aligned} x_1^{(1)} &= \frac{1}{a_{11}} (b_1 - a_{12}x_2^{(0)}) \\ x_2^{(1)} &= \frac{1}{a_{22}} (b_2 - a_{21}x_1^{(0)}). \end{aligned} \tag{4.7.1}$$

Nadamo se da je $x^{(1)} = \left[\, x_1^{(1)}, x_2^{(1)} \,\right]^T$ bolja aproksimacija nego što je to $x^{(0)}$. Na isti način možemo iskoristiti $x^{(1)}$ da dobijemo sljedeću aproksimaciju, $x^{(2)}$, zatim $x^{(3)}$, itd. Pitanje je, naravno, pod kojim uvjetima te iteracije teže prema rješenju x.

Prije same analize, zgodno je uočiti strukturu računanja vektora $x^{(k+1)}$ pomoću $x^{(k)}$. Primijetimo da vrijedi

$$\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}} & 0 \\ 0 & \frac{1}{a_{22}} \end{bmatrix} \left(\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} 0 & -a_{12} \\ -a_{21} & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \end{bmatrix} \right).$$

Dakle, ako stavimo

$$A = D - N, \quad D = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}, \quad N = \begin{bmatrix} 0 & -a_{12} \\ -a_{21} & 0 \end{bmatrix}, \tag{4.7.2}$$

onda možemo jednostavno pisati

$$x^{(k+1)} = D^{-1}(b + Nx^{(k)}) = D^{-1}Nx^{(k)} + D^{-1}b. \tag{4.7.3}$$

Relacijom (4.7.3) definirana je **Jacobijeva iterativna metoda**.

Pokušajmo ovaj proces ilustrirati na jednom primjeru. Zbog jednostavnosti primjer je dimenzije 2×2 tako da svatko može lako slijediti račun.

Primjer 4.7.1 *Neka je*

$$A = \begin{bmatrix} 2 & 0.1 \\ -0.1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 19.9 \\ -3 \end{bmatrix}, \quad x = A^{-1}b = \begin{bmatrix} 10 \\ -1 \end{bmatrix}.$$

Matricu A napišimo kao u relaciji (4.7.2). Za početnu iteraciju uzmimo vektor

$$x^{(0)} = D^{-1}b = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 19.9 \\ -3 \end{bmatrix} = \begin{bmatrix} 9.949999999999999 \\ -1.5 \end{bmatrix}.$$

Primijetimo da u početnoj iteraciji pokušavamo "pogoditi" rješenje. Ponekad je dovoljno uzeti slučajno odabran vektor. Ipak, poželjno je da je polazna iteracija što je moguće bliže cilju. Naš izbor je bio rezultat jednostavne ideje: matricu A aproksimiramo s D (jer su dijagonalni elementi veći od izvandijagonalnih), pa $A^{-1}b$ aproksimiramo s $D^{-1}b$. Naravno da je ovo gruba aproksimacija, ali ipak ima smisla. Sada iteriramo kao u relaciji (4.7.3) i dobijemo

$$x^{(1)} = \begin{bmatrix} 1.002500000000000e{+}001 \\ -1.002500000000000e{+}000 \end{bmatrix},$$

$$x^{(2)} = \begin{bmatrix} 1.000012500000000e{+}001 \\ -9.987500000000000e{-}001 \end{bmatrix},$$

$$x^{(3)} = \begin{bmatrix} 9.999937500000000e{+}000 \\ -9.999937500000000e{-}001 \end{bmatrix},$$

$$x^{(4)} = \begin{bmatrix} 9.999999687499999e{+}000 \\ -1.000003125000000e{+}000 \end{bmatrix},$$

$$x^{(5)} = \begin{bmatrix} 1.000000015625000e{+}001 \\ -1.000000015625000e{+}000 \end{bmatrix}.$$

Ako izračunamo relativne greške $\epsilon_k = \|x - x^{(k)}\|_\infty / \|x\|_\infty$*, onda je*

$$\begin{aligned}
\epsilon_0 &= 5.000000000000000e{-}001\\
\epsilon_1 &= 2.499999999999858e{-}002\\
\epsilon_2 &= 1.249999999999973e{-}003\\
\epsilon_3 &= 6.250000000029843e{-}005\\
\epsilon_4 &= 3.125000000103739e{-}006\\
\epsilon_5 &= 1.562499996055067e{-}007.
\end{aligned}$$

Korištenjem (4.7.3) i relacije

$$b = Ax = (D - N)x,$$

lako izračunamo ponašanje pogreške $e^{(k)} = x^{(k)} - x$. Vrijedi

$$e^{(k+1)} = x^{(k+1)} - x = D^{-1}N(x^{(k)} - x) = D^{-1}Ne^{(k)}. \tag{4.7.4}$$

Primijetimo da izvedene relacije (4.7.2), (4.7.3), (4.7.4) vrijede za proizvoljni $n \geq 2$, gdje je

$$A = D - N, \quad D = \operatorname{diag}(a_{11}, \ldots, a_{nn}), \quad \prod_{i=1}^{n} a_{ii} \neq 0.$$

Sada iz (4.7.4) slijedi

$$e^{(k)} = D^{-1}Ne^{(k-1)} = (D^{-1}N)^2 e^{(k-2)} = (D^{-1}N)^k e^{(0)}, \tag{4.7.5}$$

gdje je $e^{(0)} = x^{(0)} - x$ pogreška prve iteracije $x^{(0)}$. Uzimanjem proizvoljne vektorske norme $\|\cdot\|$ i odgovarajuće matrične norme, dobivamo

$$\|e^{(k)}\| \leq \|(D^{-1}N)^k\| \|e^{(0)}\| \leq \|D^{-1}N\|^k \|e^{(0)}\|. \tag{4.7.6}$$

Iz relacije (4.7.6) zaključujemo da će $e^{(k)}$ težiti nuli za svaki početni $x^{(0)}$ ako matrice $(D^{-1}N)^k$ teže nuli za $k \to \infty$. Na primjer, ako je $\|D^{-1}N\| < 1$, onda svakako $\|D^{-1}N\|^k \to 0$ za $k \to \infty$. Vidimo i da je nakon k-tog koraka greška $\|e^{(k)}\|$ barem $\|D^{-1}N\|^k$ puta manja od polazne $\|e^{(0)}\|$. Zapišimo ove zaključke u obliku propozicije.

Propozicija 4.7.1 *Ako je u rastavu* $A = D - N$ *u nekoj matričnoj normi ispunjeno*

$$\|D^{-1}N\| < 1,$$

onda za svaku početnu iteraciju $x^{(0)}$ *niz*

$$x^{(k+1)} = D^{-1}(b + Nx^{(k)}), \quad k = 0, 1, 2, \ldots \tag{4.7.7}$$

konvergira rješenju x *sustava* $Ax = b$.

Propozicija 4.7.2 *Ako je matrica A dijagonalno dominantna u smislu da je*

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad i = 1, \ldots, n,$$

onda niz generiran Jacobijevom metodom s proizvoljnim $x^{(0)}$ konvergira prema rješenju sustava $Ax = b$.

Dokaz. Lako je pokazati da je $\|D^{-1}N\|_\infty < 1$. ■

Primijetimo da smo u relacijama (4.7.1) $x_1^{(1)}$ i $x_2^{(1)}$ računali neovisno, pomoću $x_1^{(0)}$ i $x_2^{(0)}$. Ako pažljivije promotrimo formule (4.7.1), vidimo da bi imalo smisla u formuli za $x_2^{(1)}$ umjesto $x_1^{(0)}$ koristiti novu, upravo izračunatu (i vjerojatno bolju) vrijednost $x_1^{(1)}$. Općenito, formulu za $x^{(k+1)}$ modificiramo tako da u svakoj komponenti koristimo najsvježije izračunate vrijednosti. Na primjer, u slučaju $n = 4$ imali bismo

$$\begin{aligned}
x_1^{(k+1)} &= \frac{1}{a_{11}}\,(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)}),\\
x_2^{(k+1)} &= \frac{1}{a_{22}}\,(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - a_{24}x_4^{(k)}),\\
x_3^{(k+1)} &= \frac{1}{a_{33}}\,(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - a_{34}x_4^{(k)}),\\
x_4^{(k+1)} &= \frac{1}{a_{44}}\,(b_4 - a_{41}x_1^{(k+1)} - a_{42}x_2^{(k+1)} - a_{43}x_3^{(k+1)}).
\end{aligned} \tag{4.7.8}$$

Jasno je da je u općenitom slučaju

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, \ldots, n. \tag{4.7.9}$$

Gornje formule definiraju **Gauss–Seidelovu iterativnu metodu**.

Da bismo bolje shvatili strukturu izvedenih formula, vratimo se primjeru $n = 4$. Uočimo da je u relaciji (4.7.8)

$$-\begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} \\ 0 & 0 & a_{23} & a_{24} \\ 0 & 0 & 0 & a_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \\ x_4^{(k)} \end{bmatrix} = \begin{bmatrix} -a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} \\ -a_{23}x_3^{(k)} - a_{24}x_4^{(k)} \\ -a_{34}x_4^{(k)} \\ 0 \end{bmatrix},$$

te da je vektor $x^{(k+1)}$ zapravo rješenje donjetrokutastog sustava

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \\ x_4^{(k+1)} \end{bmatrix} = \begin{bmatrix} -a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} \\ -a_{23}x_3^{(k)} - a_{24}x_4^{(k)} \\ -a_{34}x_4^{(k)} \\ 0 \end{bmatrix}. \tag{4.7.10}$$

Stavimo

$$L = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}, \quad U = - \begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} \\ 0 & 0 & a_{23} & a_{24} \\ 0 & 0 & 0 & a_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.7.11}$$

Vrijedi $A = L - U$ i, uz uvjet regularnosti matrice L, Gauss–Seidelovu metodu možemo zapisati kao

$$x^{(k+1)} = L^{-1}(b + Ux^{(k)}), \quad k = 1, 2, \ldots \tag{4.7.12}$$

Kao i u analizi Jacobijeve metode, dobivamo da je

$$e^{(k)} = (L^{-1}U)^k e^{(0)}, \quad k = 1, 2, \ldots \tag{4.7.13}$$

Propozicija 4.7.3 *Ako je A simetrična i pozitivno definitna matrica, onda niz generiran Gauss–Seidelovom metodom s proizvoljnim početnim $x^{(0)}$ konvergira prema rješenju sustava $Ax = b$.*

I Jacobijeva i Gauss–Seidelova metoda generirane su istom shemom: Matrica A je napisana u obliku $A = M - S$, gdje je M regularna matrica, a iteracije su dane s

$$x^{(k+1)} = M^{-1}(b + Sx^{(k)}), \quad k = 1, 2, \ldots \tag{4.7.14}$$

Pri tome je matrica M odabrana tako da ju je lako invertirati, tj. da je lako riješiti sustav

$$Mx^{(k+1)} = b + Sx^{(k)}.$$

U Jacobijevoj metodi je $M = D$ dijagonalna, a u Gauss–Seidelovoj $M = L$ je donjetrokutasta matrica. Konvergencija prema rješenju sustava $Ax = b$ je osigurana ako je $\|M^{-1}N\| < 1$ za neku matričnu normu $\|\cdot\|$.

4.8. Matematički software za problem $Ax = b$

U ovom poglavlju opisujemo BLAS i LAPACK, dvije trenutno najpopularnije biblioteke programa za rješavanje problema numeričke linearne algebre. Biblioteka BLAS (Basic Linear Algebra Subroutines, osnovni potprogrami za linearnu algebru) sadrži potprograme za elementarne operacije s vektorima i matricama, dok je LAPACK opsežna biblioteka sa gotovim rješavačima za probleme kao što su linearni sustavi jednadžbi, problemi najmanjih kvadrata, problemi svojstvenih i singularnih vrijednosti za matrice i matrične parove. U LAPACK-u su sve elementarne operacije nad matricama i vektorima izvedene pomoću poziva biblioteke BLAS.

Obje biblioteke dostupne preko Interneta. Na adresi `http://www.netlib.org` mogu se naći implementacije u programskim jezicima FORTRAN i C.

Cilj ovog odjeljka nije detaljan opis tih biblioteka. Umjesto toga želimo čitatelju dati osnovnu informaciju i uputiti ga na izvore informacija. Za potrebe ove knjige smo skupili neke bilbioteke programa, tako da ih čitatelj može lako kopirati na svoje računalo i koristiti.

4.8.1. Pregled biblioteke BLAS

Biblioteka BLAS je nastala iz potrebe da se elementarne operacije u linearnoj algebri na neki način standardiziraju. Korist od takvog standardiziranja je višestruka. Kao prvo, programiranje kompliciranih algoritama je pojednostavljeno: pri proračunu konstrukcije tankera ne treba trošiti vrijeme oko detalja kao što je npr. duljina vektora ili produkt dvije matrice. Nadalje, i najsloženiji proračuni su sastavljeni od takvih elementarnih operacija, pa je poželjno da su elementarne operacije implementirane na najefikasniji mogući način.

Optimalna implementacija algoritma kao što je na primjer množenje dvije matrice nije uvijek jednostavan posao. Za optimalni rezultat je potrebno poznavati detalje građe i funkcioniranja pojedinog računala. Sve to zahtijeva dodatne napore što onda poskupljuje izradu programa. Osim toga, prelaskom na drugo računalo se mijenjaju parametri optimalne implementacije i postupak treba ponoviti. Time je i održavanje dobrog, efikasnog programa skupo.

Činilo se razumnim odabrati jedan, ne preveliki, skup operacija koji je pak dovoljno velik da se iz njega može izvesti većina ostalih operacija. Za odabrane operacije se specificiraju procedure i to

(*i*) imenom;

(*ii*) listom ulaznih parametara,

(*iii*) listom izlaznih vrijednosti.

Takodjer, zahtijeva se da su operacije implementirane za sve tipove brojeva koje koristi računalo. Kako je BLAS povijesno vezan za programski jezik FORTRAN, ti tipovi su REAL, DOUBLE PRECISION, COMPLEX i DOUBLE COMPLEX. U nekim operacijama, kao, na primjer, pri traženju komponente vektora s najvećom apsolutnom vrijednošću, rezultat je cjelobrojan (INTEGER). Zbog preglednosti, uzeto je da je prvo slovo imena procedure oznaka tipa. Tako imena koja počinju sa I označavaju cjelobrojni tip (INTEGER), S označava jednostruku preciznost (SINGLE), D označava dvostruku preciznost (DOUBLE), C označava kompleksni tip (COMPLEX), a Z označava kompleksni tip u dvostrukoj preciznosti. (Implementacija BLAS-a u jeziku C ima sličnu strukturu, gdje je hijerarhija tipova ona iz C-a.)

Na ovaj način je pojednostavljeno programiranje, a optimizacija implementacije je svedena na mali skup standardiziranih procedura. Implementacija odabranih

procedura se onda može prepustiti stručnjacima za *software*. Tržišna utakmica između proizvođača računala je dovela do toga da proizvođači nude tvornički optimirane verzije BLAS biblioteke za svoja računala[1]. Takve implementacije imaju vrijeme izvršavanja znatno kraće od "naivne" implementacije.

Operacije s vektorima: BLAS 1

Zbog jednostavnosti, u ovoj točki je s (n, X, kx) zadan vektor x tipa REAL. To znači da su elementi vektora x na pozicijama $X(1 + (i-1)kx)$, $i = 1, \ldots, n$. Parametar kx je **korak** (engleski termin je *stride*) i njegovo korištenje je vezano za način spremanja vektora u memoriji računala.

Slično je s (n, DX, kx) zadan vektor tipa DOUBLE PRECISION, s (n, CX, kx) vektor tipa COMPLEX i s (n, ZX, kx) vektor tipa DOUBLE COMPLEX.

Dat ćemo kratak pregled osnovnih operacija u paketu BLAS 1. Kako su i izvorni kod i dokumentacija dostupni preko Interneta, nećemo ulaziti u sve detalje, niti ćemo dati pregled cijelog paketa. U skupini operacija sa vektorima izdvajamo:

- SCOPY kopira vektor x u vektor y.
  Deklarirana je sa
  SUBROUTINE SCOPY (n, X, kx, Y, ky).
  Varijante su:
  DCOPY (n, DX, kx, DY, ky),
  CCOPY (n, CX, kx, CY, ky),
  ZCOPY (n, ZX, kx, ZY, ky).
- SSWAP razmjenjuje sadžaj vektora x i y.
  Deklarirana je sa
  SUBROUTINE SSWAP (n, X, kx, Y, ky).
  Varijante su:
  DSWAP (n, DX, kx, DY, ky),
  CSWAP (n, CX, kx, CY, ky),
  ZSWAP (n, ZX, kx, ZY, ky).
- ISAMAX računa najmanji indeks i za koji je $|x_i| = \max\limits_{1 \le j \le n} |x_j|$.
  Deklarirana je s
  INTEGER FUNCTION ISAMAX (n, X, kx).
  Varijante su:
  INTEGER FUNCTION IDAMAX (n, DX, kx),
  INTEGER FUNCTION ICAMAX (n, CX, kx),
  INTEGER FUNCTION IZAMAX (n, ZX, kx).

[1]Naravno, ne besplatno. Obično se takve biblioteke kupuju zajedno sa ostalom programskom podrškom.

- SNRM2 računa euklidsku duljinu $\|x\|_2$ vektora $x = (n, X, kx)$ tipa REAL.
  Deklarirana je s
  REAL FUNCTION SNRM2 (n, X, kx).
  Varijante su:
  DOUBLE PRECISION FUNCTION DNRM2 (n, DX, kx),
  REAL FUNCTION SCNRM2 (n, CX, kx),
  DOUBLE PRECISION FUNCTION DZNRM2 (n, ZX, kx).
- SASUM računa ℓ_1 normu $\|x\|_1 = |x_1| + \cdots + |x_n|$.
  Deklarirana je s
  REAL FUNCTION SASUM (n, X, kx).
  Varijante su:
  DOUBLE PRECISION FUNCTION DASUM (n, DX, kx),
  REAL FUNCTION SCASUM (n, CX, kx),
  DOUBLE PRECISION FUNCTION DZASUM (n, ZX, kx).
- SSCAL računa $a \cdot x$, gdje je a skalar istog tipa kao i vektor x. Rezultat je u vektoru x.
  Deklarirana je sa
  SUBROUTINE SSCAL (n, SA, SX, kx).
  Varijante su:
  DSCAL (n, DA, DX, kx),
  CSCAL (n, CA, CX, kx),
  ZSCAL (n, ZA, ZX, kx).
- SDOT računa skalarni produkt $(x, y) = y^* x$ vektora x i y.
  Deklarirana je s
  REAL FUNCTION SDOT (n, X, kx, Y, ky).
  Varijante su:
  DOUBLE PRECISION FUNCTION DDOT (n, DX, kx, DY, ky),
  COMPLEX FUNCTION CDOTC (n, CX, kx, CY, ky) (računa $\sum_i \overline{x}_i y_i$),
  COMPLEX FUNCTION CDOTU (n, CX, kx, CY, ky) (računa $\sum_i x_i y_i$),
  DOUBLE COMPLEX FUNCTION ZDOTC (n, ZX, kx, ZY, ky) i
  DOUBLE COMPLEX FUNCTION ZDOTU (n, ZX, kx, ZY, ky).
- SROT primijenjuje ravninsku rotaciju na vektore x i y. Preciznije, matrica $[x\ y]$ je zamijenjena s $[c \cdot x + s \cdot y, c \cdot y - s \cdot x]$.
  Procedura je deklarirana sa
  SUBROUTINE SROT (n, X, kx, Y, ky, C, S).
  Varijante su:
  DROT $(n, DX, kx, DY, ky, DC, DS)$,
  CSROT $(n, CX, kx, CY, ky, C, S)$,
  ZDROT $(n, ZX, kx, ZY, ky, DC, DS)$.

- SROTG računa Givensovu ravninsku rotaciju.
  Deklarirana je sa
  SUBROUTINE SROTG (A, B, C, S).
  Izlazni parametri C i S su izračunati tako da je
  $$\begin{bmatrix} C & S \\ -S & C \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sqrt{A^2+B^2} \\ 0 \end{bmatrix}.$$
  Ulazna vrijednost A je na izlazu zamijenjena s $\sqrt{A^2+B^2}$, a B sa S, ako je $A > B$, odnosno s $1/C$ ako je $C \neq 0$ i $A \leq B$.
  Varijante ove procedure su:
  DROTG (DA, DB, DC, DS),
  CROTG (CA, CB, C, CS),
  ZROTG (ZA, ZB, DC, ZS).
- SROTMG računa modificiranu Givensovu rotaciju.
- SROTM primijenjuje modificiranu Givensovu rotaciju.

Matrice i vektori: BLAS 2

Na drugom nivou biblioteke BLAS su operacije sa matricama i vektorima. Kako je cijela biblioteka dostupna "*online*", mi ćemo ovdje spomenuti samo tri procedure. Slovo "x" na početku imena procedure stoji za "S", "D", "C", ili "Z".

- xGEMV računa izraz oblika
  $$y := \alpha Ax + \beta y, \quad \text{ili} \quad y := \alpha A^T x + \beta y,$$
  gdje su α i β skalari, A je $m \times n$ matrica, a x i y su vektori odgovarajućih dimenzija.
- xTRMV računa produkt $y = Ax$ ili $y = A^T x$, gdje je A $n \times n$ gornje ili donjetrokutasta matrica s općenitom ili jediničnom dijagonalom.
- xTRSV rješava linearne sustave
  $$Ax = b, \quad \text{ili} \quad A^T x = b,$$
  gdje je A gornje ili donjetrokutasta matrica s općenitom ili jediničnom dijagonalom, a b je vektor odgovarajuće dimenzije.

Čak i iz ovakvo kratkog prikaza može se vidjeti s koliko pažnje je dizajnirana funkcionalnost procedura. Odabir pojedine varijante je omogućen podešavanjem ulaznih parametara. Posebno su napisane procedure za matrice sa specijalnom strukturom kao npr. simetrične i vrpčaste matrice. Za sve detalje čitatelja upućujemo na izvor `http://www.netlib.org/blas`.

Operacije s matricama: BLAS 3

Treći nivo elementarnih operacija biblioteke BLAS implementira matrične operacije. Tek za ilustraciju funkcionalnosti procedura trećeg nivoa, opišimo tri najosnovnije.

- xGEMM računa izraz oblika

  $$C := \alpha f(A) g(B) + \beta C,$$

  gdje su α i β skalari, A, B, C matrice, $f(A) \in \{A, A^T\}$, $g(B) \in \{B, B^T\}$, pri čemu su dimenzije matrica takve da je izraz dobro definiran.

- xTRMM računa

  $$B := \alpha f(A) B, \quad \text{ili} \quad B := \alpha B f(A),$$

  gdje je α skalar, A i B su matrice, pri čemu je A gornje ili donjetrokutasta s općenitom ili jediničnom dijagonalom, te $f(A) \in \{A, A^T\}$.

- xTRSM rješava po X jednadžbe

  $$f(A) X = \alpha B, \quad \text{ili} \quad X f(A) = \alpha B,$$

  gdje je A gornje ili donjetrokutasta regularna matrica sa općenitom ili jediničnom dijagonalom, $f(A) \in \{A, A^T\}$, B i X su matrice odgovarajućih dimenzija.

Ostale procedure i kompletna dokumentacija mogu se naći na Internetu, na adresi `http://www.netlib.org`.

4.8.2. Pregled biblioteke LAPACK

LAPACK je kratica za Linear Algebra PACKage, paket (programa) za linearnu algebru. Osnovne značajke LAPACK–a su:

1. Opsežnost. LAPACK sadrži preko 1000 potprograma s algoritmima za rješavanje problema linearne algebre. Spomenimo da su na listi algoritmi za rješavanje linearnih sustava (s pripadnim algoritmima za računanje LU faktorizacije, faktorizacije Choleskog, simetrične indefinitne faktorizacije), rješavanje problema najmanjih kvadrata (s pripadnim algoritmima za računanje QR faktorizacije, generalizirane QR faktorizacije), rješavanje problema svojstvenih vrijednosti (simetrični problem, nesimetrični problem, generalizirani problemi za matrične parove), računanje obične i generalizirane dekompozicije singularnih vrijednosti, rješavanje matričnih jednadžbi (npr. Sylvesterove jednadžbe). Sve procedure su implementirane i za realne i za kompleksne tipove podataka. Posebni algoritmi su ponudjeni za specijalne tipove matrica kao što su npr. vrpčaste matrice.

2. Numerička pouzdanost. U LAPACK-u se posebna pažnja posvećuje numeričkoj stabilnosti. Uz dosta algoritama ponudjene su procedure za ocjenu greške u izračunatim rezultatima. Dani su teorijski okviri u kojima je ocjenjena stabilnost algoritama, tako da korisnik može dobiti dodatne informacije i o rezultatima ali i o svom matematičkom modelu. (Numerička nestabilnost može biti znak greške u modelu.)
3. Prenosivost i efikasnost na raznim računalima. Efikasnost biblioteke LAPACK je postignuta oslanjanjem na biblioteku BLAS. Drugim riječima, ako imamo BLAS optimiran za konkretno računalo, onda će LAPACK dobro iskoristiti resurse tog računala. Prenosivost je osigurana strogim pridržavanjem standarda konkretnog programskog jezika (FORTRAN ili C).
4. Dobra dokumentiranost i jednostavno korištenje. Sve su procedure detaljno dokumentirane na jednoobrazan i unaprijed definiran način. Matematički detalji, kao i detalji implementacije su detaljno opisani u seriji tehničkih izvještaja (LAPACK Working Notes) koji su dostupni preko Interneta. Objavljen je i priručnik [1].
5. Neprestano usavršavanje. Istraživači koji sudjeluju u projektu LAPACK dolaze iz cijelog svijeta i neprestano rade na pronalaženju novih, boljih algoritama koji nakon strogih provjera postaju dijelovi LAPACK-a, bilo kao nove procedure, bilo kao zamjena za već postojeće. U vrijeme pisanja ovih redaka, u uporabi je LAPACK, verzija 3.0.
6. Kompletan izvorni kod je dostupan preko Interneta, zajedno s MAKE datotekama koje same izvršavaju proces instaliranja. Za neka računala postoji gotova biblioteka koju samo treba kopirati koristeći FTP.

4.8.3. Rješavanje linearnih sustava pomoću LAPACK-a

Kako smo vidjeli u prethodnim odjeljcima, rješavanje linearnog sustava jednadžbi ima tri faze: LU faktorizacija, rješavanje donjetrokutastog i rješavanje gornjetrokutastog sustava jednadžbi. Pogledajmo kako je to napravljeno u LAPACK-u. Zbog jednostavnosti, opisujemo procedure u jednostrukoj preciznosti. Dat ćemo i dijelove izvornog koda, kao ilustraciju dobre prakse programiranja – kako strukture programa tako i dokumentiranosti. Naravno, cijeli kod je dostupan "*online*".

Trokutastu faktorizaciju računamo procedurom SGETRF. Pogledajmo opis parametara te procedure, kako je napisano u izvornom kodu:

```
      SUBROUTINE SGETRF( M, N, A, LDA, IPIV, INFO )
*
*  -- LAPACK routine (version 3.0) --
*     Univ. of Tennessee, Univ. of California Berkeley, NAG Ltd.,
```

```
*     Courant Institute, Argonne National Lab, and Rice University
*     March 31, 1993
*
*     .. Scalar Arguments ..
      INTEGER            INFO, LDA, M, N
*     ..
*     .. Array Arguments ..
      INTEGER            IPIV( * )
      REAL               A( LDA, * )
*     ..
*
*  Purpose
*  =======
*
*  SGETRF computes an LU factorization of a general M-by-N matrix A
*  using partial pivoting with row interchanges.
*
*  The factorization has the form
*     A = P * L * U
*  where P is a permutation matrix, L is lower triangular with unit
*  diagonal elements (lower trapezoidal if m > n), and U is upper
*  triangular (upper trapezoidal if m < n).
*
*  This is the right-looking Level 3 BLAS version of the algorithm.
*
*  Arguments
*  =========
*
*  M       (input) INTEGER
*          The number of rows of the matrix A.  M >= 0.
*
*  N       (input) INTEGER
*          The number of columns of the matrix A.  N >= 0.
*
*  A       (input/output) REAL array, dimension (LDA,N)
*          On entry, the M-by-N matrix to be factored.
*          On exit, the factors L and U from the factorization
*          A = P*L*U; the unit diagonal elements of L are not stored.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A.  LDA >= max(1,M).
*
*  IPIV    (output) INTEGER array, dimension (min(M,N))
*          The pivot indices; for 1 <= i <= min(M,N), row i of the
*          matrix was interchanged with row IPIV(i).
```

```
*
*  INFO    (output) INTEGER
*          = 0:  successful exit
*          < 0:  if INFO = -i, the i-th argument had an illegal value
*          > 0:  if INFO = i, U(i,i) is exactly zero. The factorization
*                has been completed, but the factor U is exactly
*                singular, and division by zero will occur if it is used
*                to solve a system of equations.
*
*  =====================================================================
```

Dobro dokumentiranim programima ne treba dodatni komentar!

Rješavanje trokutastih sustava s matricama L i U izvršava procedure SGETRS u kojoj se poziva procedura za rješavanje trokutastih sustava i to sa jednom ili više desnih strana. Evo kako izgleda početak procedure SGETRS.

```
      SUBROUTINE SGETRS( TRANS, N, NRHS, A, LDA, IPIV, B, LDB, INFO )
*
*  -- LAPACK routine (version 3.0) --
*     Univ. of Tennessee, Univ. of California Berkeley, NAG Ltd.,
*     Courant Institute, Argonne National Lab, and Rice University
*     March 31, 1993
*
*     .. Scalar Arguments ..
      CHARACTER          TRANS
      INTEGER            INFO, LDA, LDB, N, NRHS
*     ..
*     .. Array Arguments ..
      INTEGER            IPIV( * )
      REAL               A( LDA, * ), B( LDB, * )
*     ..
*
*  Purpose
*  =======
*
*  SGETRS solves a system of linear equations
*     A * X = B  or  A' * X = B
*  with a general N-by-N matrix A using the LU factorization computed
*  by SGETRF.
*
*  Arguments
*  =========
*
*  TRANS   (input) CHARACTER*1
*          Specifies the form of the system of equations:
```

```
*          = 'N':  A * X = B  (No transpose)
*          = 'T':  A'* X = B  (Transpose)
*          = 'C':  A'* X = B  (Conjugate transpose = Transpose)
*
*  N       (input) INTEGER
*          The order of the matrix A.  N >= 0.
*
*  NRHS    (input) INTEGER
*          The number of right hand sides, i.e., the number of columns
*          of the matrix B.  NRHS >= 0.
*
*  A       (input) REAL array, dimension (LDA,N)
*          The factors L and U from the factorization A = P*L*U
*          as computed by SGETRF.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A.  LDA >= max(1,N).
*
*  IPIV    (input) INTEGER array, dimension (N)
*          The pivot indices from SGETRF; for 1<=i<=N, row i of the
*          matrix was interchanged with row IPIV(i).
*
*  B       (input/output) REAL array, dimension (LDB,NRHS)
*          On entry, the right hand side matrix B.
*          On exit, the solution matrix X.
*
*  LDB     (input) INTEGER
*          The leading dimension of the array B.  LDB >= max(1,N).
*
*  INFO    (output) INTEGER
*          = 0:  successful exit
*          < 0:  if INFO = -i, the i-th argument had an illegal value
*
*  =====================================================================
```

I konačno, prethodne dvije procedure su integrirane u rješavač sustava $Ax = b$, odnosno $AX = B$ ako imamo više desnih strana (matrica B umjesto vektora b).

```
      SUBROUTINE SGESV( N, NRHS, A, LDA, IPIV, B, LDB, INFO )
*
*  -- LAPACK driver routine (version 3.0) --
*     Univ. of Tennessee, Univ. of California Berkeley, NAG Ltd.,
*     Courant Institute, Argonne National Lab, and Rice University
*     March 31, 1993
*
```

```
*     .. Scalar Arguments ..
      INTEGER            INFO, LDA, LDB, N, NRHS
*     ..
*     .. Array Arguments ..
      INTEGER            IPIV( * )
      REAL               A( LDA, * ), B( LDB, * )
*     ..
*
*  Purpose
*  =======
*
*  SGESV computes the solution to a real system of linear equations
*     A * X = B,
*  where A is an N-by-N matrix and X and B are N-by-NRHS matrices.
*
*  The LU decomposition with partial pivoting and row interchanges is
*  used to factor A as
*     A = P * L * U,
*  where P is a permutation matrix, L is unit lower triangular, and U is
*  upper triangular.  The factored form of A is then used to solve the
*  system of equations A * X = B.
*
*  Arguments
*  =========
*
*  N       (input) INTEGER
*          The number of linear equations, i.e., the order of the
*          matrix A.  N >= 0.
*
*  NRHS    (input) INTEGER
*          The number of right hand sides, i.e., the number of columns
*          of the matrix B.  NRHS >= 0.
*
*  A       (input/output) REAL array, dimension (LDA,N)
*          On entry, the N-by-N coefficient matrix A.
*          On exit, the factors L and U from the factorization
*          A = P*L*U; the unit diagonal elements of L are not stored.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A.  LDA >= max(1,N).
*
*  IPIV    (output) INTEGER array, dimension (N)
*          The pivot indices that define the permutation matrix P;
*          row i of the matrix was interchanged with row IPIV(i).
*
```

```
*  B       (input/output) REAL array, dimension (LDB,NRHS)
*          On entry, the N-by-NRHS matrix of right hand side matrix B.
*          On exit, if INFO = 0, the N-by-NRHS solution matrix X.
*
*  LDB     (input) INTEGER
*          The leading dimension of the array B.  LDB >= max(1,N).
*
*  INFO    (output) INTEGER
*          = 0:  successful exit
*          < 0:  if INFO = -i, the i-th argument had an illegal value
*          > 0:  if INFO = i, U(i,i) is exactly zero.  The factorization
*                has been completed, but the factor U is exactly
*                singular, so the solution could not be computed.
*
*  =====================================================================
*
*     .. External Subroutines ..
      EXTERNAL           SGETRF, SGETRS, XERBLA
*     ..
*     .. Intrinsic Functions ..
      INTRINSIC          MAX
*     ..
*     .. Executable Statements ..
*
*     Test the input parameters.
*
      INFO = 0
      IF( N.LT.0 ) THEN
         INFO = -1
      ELSE IF( NRHS.LT.0 ) THEN
         INFO = -2
      ELSE IF( LDA.LT.MAX( 1, N ) ) THEN
         INFO = -4
      ELSE IF( LDB.LT.MAX( 1, N ) ) THEN
         INFO = -7
      END IF
      IF( INFO.NE.0 ) THEN
         CALL XERBLA( 'SGESV ', -INFO )
         RETURN
      END IF
*
*     Compute the LU factorization of A.
*
      CALL SGETRF( N, N, A, LDA, IPIV, INFO )
      IF( INFO.EQ.0 ) THEN
```

```
*
*        Solve the system A*X = B, overwriting B with X.
*
         CALL SGETRS( 'No transpose', N, NRHS, A, LDA, IPIV, B, LDB,
     $                INFO )
      END IF
      RETURN
*
*     End of SGESV
*
      END
```

Na kraju, napomenimo da u LAPACK-u postoje procedure za sustave sa specijalnom strukturom (simetrične, pozitivno definitne, tridijagonalne, vrpčaste itd.). Više detalja može se naći na adresi `http://www.netlib.org/lapack`.

5. Računanje vlastitih vrijednosti i vlastitih vektora

Ovaj bi tekst trebao napisati Ivan Slapničar.

6. Izvrednjavanje funkcija

Jedan od osnovnih zadataka koji se javlja u numeričkoj matematici je izračunavanje vrijednosti funkcije u nekoj točki ili na nekom skupu točaka (tzv. izvrednjavanje funkcije). Zašto baš to?

Efikasno možemo računati samo one funkcije za koje imamo dobar algoritam za izvrednjavanje. Pri tome moramo voditi računa o tome da aritmetika računala stvarno podržava samo četiri osnovne aritmetičke operacije, pa samo njih možemo koristiti u algoritmima. Korištenje ostalih funkcija (recimo trigonometrijskih, logaritamskih ili eksponencijalnih) ovisi o kvaliteti aproksimacije upotrijebljene u odgovarajućem programskom jeziku, ili odgovarajućem sklopu u računalu. Osim toga, i kada upotrebljavamo samo četiri osnovne operacije, računanje nije egzaktno, već u svakoj operaciji imamo greške zaokruživanja. Zbog toga, pri konstrukciji algoritama imamo dva cilja uvjeta:

- efikasnost ili brzina, tj. algoritam mora imati što manji broj aritmetičkih operacija;
- točnost, tj. algoritam mora biti stabilan (unaprijed ili unazad) na greške zaokruživanja.

Oba zahtjeva su posebno bitna kod izvrednjavanja, jer se ono obično puno puta koristi, pa i mala lokalna ubrzanja daju velike ukupne uštede u vremenu, a isto vrijedi i za ukupni efekt grešaka zaokruživanja. Općenito, očekujemo da brži algoritam ima i manju grešku, jer imamo manje operacija koje doprinose ukupnoj pogrešci. Međutim, ovo **ne mora** biti istina! U mnogim slučajevima možemo drastično popraviti stabilnost algoritma tako žrtvovanjem dijela efikasnosti, a katkad je dovoljno naći pametnu reformulaciju algoritma.

U ovom poglavlju više pažnje ćemo posvetiti efikasnosti, a manje stabilnosti, osim tamo gdje realno postoji nestabilnost. Cilj nam je konstruirati efikasne algoritme i opravdati njihovu efikasnost, a ne analizirati ili dokazivati njihovu stabilnost. U skladu s tim, potencijalne nestabilnosti obrzložit ćemo na primjerima.

Neka je zadana funkcija $f : D \to \mathbb{R}$, gdje je $D \subseteq \mathbb{R}$ njena domena. Zadatak je izračunati vrijednost te funkcije f u zadanoj točki $x_0 \in D$. Preciznije, moramo naći algoritam koji računa $f(x_0)$. Naravno, točka x_0 može biti bilo koja i naš algoritam

mora raditi za sve ulaze $x_0 \in D$.

Trenutno zanemarimo pitanje kako se zadaje funkcija f. Naime, ako je f ulaz u algoritam, onda f mora biti zadana s najviše konačano mnogo podataka (o f) i ti podaci moraju jednoznačno odrediti f. To je fundamentalno ograničenje i bitno smanjuje klasu funkcija kojima vrijednosti uopće možemo algoritamski izračunati. Odgovore na takva pitanja daje tzv. teorija izračunljivosti u okviru matematičke logike i osnova matematike.

U praksi se odmah nameću i bitno jača ograničenja. Naime, ako imamo na raspolaganju samo 4 osnovne aritmetičke operacije, onda su **racionalne** funkcije jedine funkcije f kojima možemo izračunati vrijednost u bilo kojoj točki $x_0 \in D$. Takve funkcije možemo jednoznačno zadati konačnim brojem parametara — na primjer, ponašanjem ondosno vrijednostima u konačnom broju točaka (vidjeti poglavlje o interpolaciji) ili koeficijentima u nekom prikazu funkcije.

Dakle, sigurno trebamo efikasne algoritme za računanje vrijednosti racionalnih funkcija. Ako se sjetimo da racionalnu funkciju možemo zapisati kao kvocijent dva polinoma, onda je zgodno imati i algoritme za izvrednjavanje polinoma. Osim toga, polinomi su još jednostavnije funkcije, pri njihovom izvrednjavanju nema dijeljenja, a definirane su na su za sve realne (ili kompleksne) brojeve (nema problema s nultočkama nazivnika), a imaju veliku teoretsku i praktičnu primjenu.

Strogo govoreći, sve ostale funkcije moramo **aproksimirati**, polinomima ili racionalnim funkcijama. U praksi možemo pretpostaviti da za neke osnovne matematičke funkcije f već imamo dobre aproksimacije za približno računanje vrijednosti $f(x_0)$ u zadanoj točki x_0:

- procesor računala ("hardware") ima ugrađene aproksimacije i odgovarajuće instrukcije za njihov poziv (izvršavanje), ili
- koristimo neki gotovi (pot)program ("software") koji to radi.

Za osnovne matematičke funkcije kao što su: $\sqrt{x}$ (katkad i opće potencije x^α), trigonometrijske, eksponencijalne, hiperboličke i njima inverzne funkcije razvijeni su dobri algoritmi za njihove aproksimacije. O tome kako se nalaze takve aproksimacije bit će opširnije govora u poglavlju o aproksimacijama.

Bez obzira na realizaciju, u oba slučaja je bitno samo to da ih možemo direktno koristiti u našim algoritmima i da znamo da izračunata vrijednost tražene funkcije f u zadanoj točki x_0 ima malu relativnu grešku u odnosu na preciznost konačne aritmetike (tzv. točnost računanja). Drugim riječima, možemo pretpostaviti da za $f\ell(f(x_0))$ vrijede iste ili slične ocjene kao i za osnovne aritmetičke operacije (vidi odjeljke 3.3.6.) i 3.4.1.).

Reklo bi se da je ovdje sve riješeno. Međutim, nije baš tako. Računanje $f(x_0)$, u principu, traje dulje, a katkad i puno dulje, nego što je trajanje jedne osnovne

aritmetičke operacije (čak i ako sve četiri operacije nemaju isto ili podjednako trajanje). Za osnovne funkcije, trajanje može biti 10 pa i više puta dulje od trajanja jedna aritmetičke operacije. Zbog toga ponekad izbjegavamo puno poziva takvih funkcija, pogotovo ako se to može razumno izbjeći, tj. efikasno i bez većeg gubitka točnosti.

U mnogim slučajevima to se može napraviti i u ovom poglavlju ćemo pokazati neke opće algoritme tog tipa. Ideja je korištenje rekurzivnih relacija koje zadovoljavaju takve i slične, a za aproksimaciju važne funkcije. Na primjer, u teoriji aproksimacije obično se koriste ortogonalni sustavi funkcija za aproksimaciju, a funkcije takvog sustava zadovoljavaju tročlanu rekurziju.

6.1. Hornerova shema

Polinomi su najjednostavnije algebarske funkcije. Možemo ih definirati nad bilo kojim prstenom R u obliku

$$p(x) = \sum_{i=0}^{n} a_i x^i, \quad n \in \mathbb{N}_0,$$

gdje su $a_i \in R$ koeficijenti iz tog prstena, a x je simbolička "varijabla". Polinomi, kao simbolički objekti, imaju algebarsku strukturu prstena.

Međutim, polinome možemo interpretirati i kao funkcije, koje možemo izvrednjavati u svim točkama x_0 prstena R, uvrštavanjem x_0 umjesto simboličke varijable x. Dobiveni rezultat $p(x_0)$ je opet u R. Zanimaju nas efikasni algoritmi za računanje te vrijednosti.

Složenost očito ovisi o broju članova u sumi. Da broj članova ne bi bio umjetno prevelik, standardno uzimamo da je $p \neq 0$ i da je vodeći koeficijent $a_n \neq 0$, tako da je n stupanj tog polinoma p. Kada želimo naglasiti stupanj, polinom označavamo s p_n.

Algoritmi koje ćemo napraviti u principu rade nad bilo kojim prstenom R, ali neki rezultati o njihovoj složenosti vrijede samo za beskonačna neprebrojiva polja, poput $\mathbb{R}$ i $\mathbb{C}$, što su ionako najvažniji primjeri u praksi. Zbog toga, u nastavku pretpostavljamo da radimo isključivo s polinomima nad $\mathbb{R}$ ili $\mathbb{C}$. Lako je pokazati da za svako n, skup svih polinoma nad $\mathbb{R}$ ili $\mathbb{C}$ stupnja ne većeg od n čine vektorski prostor nad $\mathbb{R}$ ili $\mathbb{C}$ koji je potprostor vektorskog prostora svih polinoma nad odgovarajućim poljem $\mathbb{R}$ ili $\mathbb{C}$.

Polinom se obično zadaje stupnjem n i koeficijentima $a_0, \ldots, a_n$ u nekoj bazi vektorskog prostora polinoma stupnja ne većeg od n. Na početku razmatranja koristimo standardnu bazu $1, x, x^2, \ldots, x^n$, a kasnije ćemo algoritme modificirati i za neke druge baze.

Složenost mjerimo brojem osnovnih aritmetičkih operacija. Kad radimo s polinomima nad $\mathbb{R}$, broj operacija je korektna mjera, jer aritmetika računala modelira upravo te operacije. No, kad radimo s polinomima nad $\mathbb{C}$, treba voditi računa o tome da se kompleksne aritmetičke operacije realiziraju putem realnih, što znači da tek broj realnih operacija daje pravu mjeru složenosti. Baš zbog toga, u nekim kompleksnim algoritmima pažljivim promatranjem realnih operacija možemo ostvariti značajne uštede.

6.1.1. Računanje vrijednosti polinoma u točki

Zadan je polinom stupnja n

$$p_n(x) = \sum_{i=0}^{n} a_i x^i, \quad a_n \neq 0$$

kojemu treba izračunati vrijednost u točki x_0. To se može napraviti na više načina. Prvo, napravimo to direktno po zapisu, potencirajući. Krenemo li od nulte potencije $x^0 = 1$, svaka sljedeća potencija dobiva se rekurzivno

$$x^k = x \cdot x^{k-1}.$$

Imamo li zapamćen x^{k-1}, lako je izračunati x^k korištenjem samo jednog množenja.

Algoritam 6.1.1 (Vrijednost polinoma s pamćenjem potencija)

$sum := a_0$;
$pot := 1$;
for $i := 1$ **to** n **do**
  begin
  $pot := pot * x_0$;
  $sum := sum + a_i * pot$;
  end;
{ Na kraju je $p_n(x_0) = sum$. }

Prebrojimo zbrajanja i množenja koja se javljaju u tom algoritmu. U unutarnjoj petlji javljaju se 2 množenja i 1 zbrajanje. Budući da se petlja izvršava n puta, ukupno imamo

$$2n \text{ množenja } + n \text{ zbrajanja.}$$

Naravno, kad je polinom kompleksan, ove operacije su kompleksne.

Izvrednjavanje polinoma u točki može se izvesti i s manje množenja. Ako polinom zapišemo u obliku

$$p_n(x) = (\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_1)x + a_0.$$

Algoritam koji po prethodnoj relaciji izvrednjava polinom zove se Hornerova shema. Predložio ga je W. G. Horner, 1819. godine, ali sličan zapis je koristio i Isaac Newton, još 1669. godine.

Algoritam 6.1.2 (Hornerova shema)

$sum := a_n$;
for $i := n - 1$ **downto** 0 **do**
  $sum := sum * x_0 + a_i$;
{ Na kraju je $p_n(x_0) = sum$. }

Odmah je očito da smo korištenjem ovog algoritma broj množenja prepolovili, tj. da je njegova složenost

$$n \text{ množenja } + n \text{ zbrajanja.}$$

6.1.2. Hornerova shema je optimalan algoritam

Bitno je napomenuti da je Hornerova shema efikasan algoritam za izvrednjavaje polinoma kojima je većina koeficijenata različita od nule. Na primjer, polinom

$$p_{100}(x) = x^{100} + 1$$

besmisleno je izvrednjavati Hormerovom shemom, jer bi to predugo trajalo (binarno potenciranje je brže). Binarnim potenciranjem redom bismo potencirali bazom 2 za dani x

$$x \to x^2 \to x^4 \to x^8 \to x^{16} \to x^{32} \to x^{64},$$

i na kraju pomnožil $x^{64} \cdot x^{32} \cdot x^4$. Ukupno bismo imali 8 množenja i 1 zbrajanje. Također, kad izvrednjavamo polinom koji ima samo parne koeficijente

$$p_{2n}(x) = \sum_{i=0}^{n} a_{2i}\, x^{2i},$$

Hornerovu shemu treba modificirati tako da koristi samo parne potencije (potenciranjem $x \to x^2$). Isto vrijedi i za polinom koji ima samo neparne potencije. Sastavite pripadne algoritme.

Za Hornerovu shemu može se pokazati da je optimalan algoritam.

Teorem 6.1.1 (Borodin, Munro) *Za opći polinom n-tog stupnja potrebno je barem n aktivnih množenja. Pod aktivnim množenjem podrazumjevamo množenje između a_i i x.*

Dakle, Hornerova shema ima optimalan broj množenja. Rezultat prethodnog teorema može se poboljšati samo ako jedan te isti polinom izvrednjavamo u mnogo

točaka. Tada se koeficijenti polinoma prije samog izvrednjavanja **adaptiraju** ili **prekondicioniraju**, tako da bismo kasnije imali što manje operacija po svakoj pojedinoj točki.

Zanimljivo je Hornerova shema optimalna za polinome stupnjeva $n = 1$, 2 i 3, čak i kad računamo vrijednost polinoma u više točaka. Pokažimo jedan primjer adaptiranja koeficijenata za polinom stupnja 4.

Primjer 6.1.1 *Uzmimo opći polinom stupnja* 4

$$p_4(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

i promatrajmo sljedeću shemu računananja

$$\begin{aligned} y &= (x + c_0)x + c_1, \\ p_4(x) &= ((y + x + c_2)y + c_3)c_4. \end{aligned}$$

Primijetimo da ona koristi 3 *množenja i* 5 *zbrajanja. Međutim, prvo treba odrediti* c_i *u ovisnosti o* a_i*. Zato napišimo* p_4 *po potencijama od* x

$$\begin{aligned} p_4(x) = c_4x^4 &+ (2c_0c_4 + c_4)x^3 + (c_0^2 + 2c_1 + c_0c_4 + c_2c_4)x^2 \\ &+ (2c_0c_1c_4 + c_1c_4 + c_0c_2c_4)x + (c_1^2c_4 + c_1c_2c_4 + c_3c_4). \end{aligned}$$

Uočimo da veza između a_i *i* c_i *nije linearna. Rješavanjem po* a_i*, dobivamo*

$$\begin{aligned} c_4 &= a_4 & c_1 &= a_1/a_4 - c_0b \\ c_0 &= (a_3/a_4 - 1)/2 & c_2 &= b - 2c_1 \\ b &= a_2/a_4 - c_0(c_0 + 1) & c_3 &= a_0/a_4 - c_1(c_1 + c_2). \end{aligned}$$

Ove relacije zahtijevaju dosta računanja, ali se to obavlja samo jednom. Nakon toga će svako izvrednjavanje zahtijevati manje vremena od Hornerove sheme na polaznom polinomu, jer zbrajanje na računalu troši manje vremena od množenja.

Dapače, V. Pan je pokazao da vrijedi sljedeći teorem.

Teorem 6.1.2 (Pan) *Za bilo koji polinom* p_n *stupnja* $n \geq 3$ *postoje realni brojevi* c, d_i, e_i, *za* $0 \leq i \leq \lceil n/2 \rceil - 1$, *takvi da se* p_n *može izračunati korištenjem*

$$(\lfloor n/2 \rfloor + 2) \text{ množenja } + n \text{ zbrajanja}$$

po sljedećoj shemi

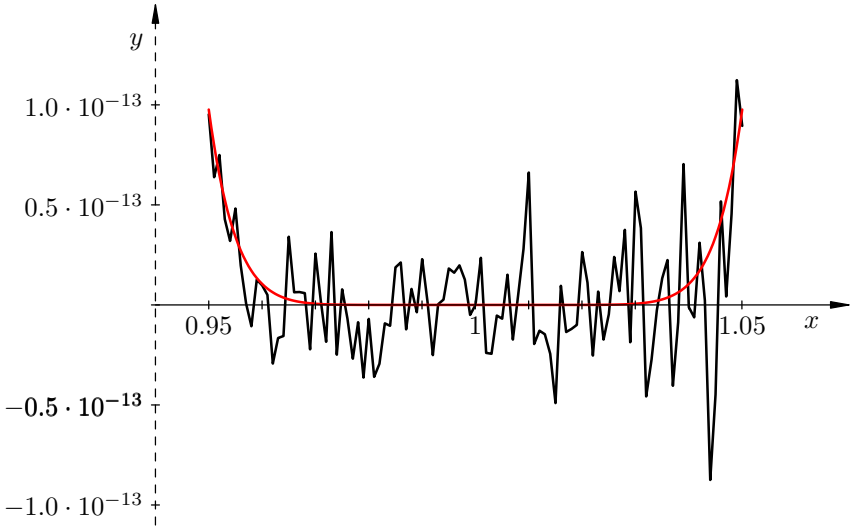
$$\begin{aligned} y &= x + c \\ w &= y^2 \\ z &:= \begin{cases} (a_ny + d_0)y + e_0, & n \text{ paran,} \\ a_ny + e_0, & n \text{ neparan,} \end{cases} \\ z &:= z(w - d_i) + e_i, \quad \text{za } i = 1, 2, \ldots \lceil n/2 \rceil - 1. \end{aligned}$$

U prethodnom teoremu nije ništa rečeno o tome koliko je operacija potrebno za računanje c, d_i i e_i. Međutim, sljedeći teorem pokazuje je red veličine broja operacija u Hornerovoj shemi optimalan za gotovo sve polinome.

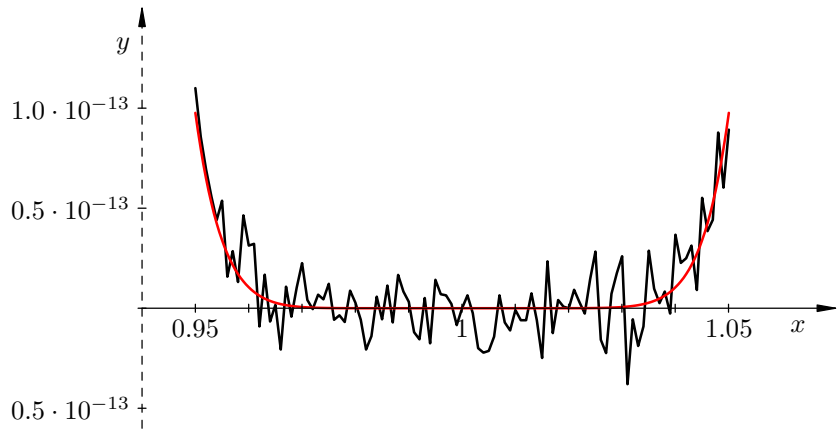
Teorem 6.1.3 (Motzkin, Belaga) *Slučajno odabrani polinom stupnja n ima vjerojatnost 0 da ga se može izračunati za strogo manje od $\lceil (n+1)/2 \rceil$ množenja/dijeljenja ili za strogo manje od n zbrajanja/oduzimanja.*

6.1.3. Stabilnost Hornerove sheme

U prošlom smo odjeljku pokazali da je Hornerova shema optimalan algoritam u smislu efikasnosti. Pažljivom analizom grešaka zaokruživanja nije teško pokazati da je Hornerova shema i stabilan algoritam.



Izvrednjavanje polinoma $(x-1)^{10}$ razvijenog po potencijama od x: korištenjem direktne sumacije u double precision aritmetici.



Izvrednjavanje polinoma $(x-1)^{10}$ razvijenog po potencijama od x:
korištenjem Hornerove sheme u double precision aritmetici.

6.1.4. Dijeljenje polinoma linearnim faktorom oblika $x - x_0$

Kako se praktično zapisuje Hornerova shema kad se radi "na ruke"? Napravi se tablica na sljedeći način. U gornjem redu se popišu svi koeficijenti polinoma p_n redom od a_n do a_0. Donji red se izračunava korištenjem gornjeg reda i broja x_0. Označimo elemente donjeg reda, gledajući slijeva nadesno, s $x_0, c_{n-1}, c_{n-2}, \ldots, c_0, r_0$, tako da se c_{n-1} nalazi ispod a_n.

	a_n	a_{n-1}	$\cdots$	a_1	a_0
x_0	c_{n-1}	c_{n-2}	$\cdots$	c_0	r_0

.

Elementi donjeg reda se računaju s lijeva na desno, kako slijedi

$$\begin{aligned} &c_{n-1} = a_n, \\ &c_{i-1} := c_i * x_0 + a_{i-1}, \quad i = n, \ldots, 1. \end{aligned} \tag{6.1.1}$$

Dakle, vodeći koeficijent a_n se prepiše, a svi ostali se računaju tako da se posljednji izračunati koeficijent c_i pomnoži s x_0, a zatim mu se doda koeficijent a_{i-1} koji se nalazi iznad. Na kraju, ispod koeficijenta a_0 se dobije r_0, tj. vrijednost polinoma u točki x_0. Pokažimo kako to funkcionira na konkretnom primjeru.

Primjer 6.1.2 *Izračunajmo vrijednost polinoma*

$$p_5(x) = 2x^5 - x^3 + 4x^2 + 1$$

u točki $x_0 = -1$.

Formirajmo tablicu:

	2	0	-1	4	0	1
-1	2	-2	1	3	-3	4

.

Dakle, $p_5(-1) = 4$.

Pogledajmo značenje koeficijenata c_i koji se javljaju u donjem redu tablice. Promatrajmo polinom koji dobijemo dijeljenjem polinoma p_n s polinomom stupnja 1 oblika $x - x_0$. Nazovimo taj kvocijent td dva polinoma s q_{n-1} (to je ponovno polinom, ali sada stupnja $n-1$), a ostatak (broj, jer mora biti stupnja manjeg od polinoma kojim dijelimo) s r_0. Tada vrijedi

$$p_n(x) = (x - x_0)q_{n-1}(x) + r_0. \tag{6.1.2}$$

Uvrštavanje $x = x_0$ u (6.1.2) pokazuje da za ostatak vrijedi $r_0 = p_n(x_0)$. Znamo da je q_{n-1} polinom stupnja $n-1$, a njegove koeficijente označimo s b_i, $1 \leq i \leq n$ (što je pomak indeksa za jedan u odnosu na dosad korištenu notaciju),

$$q_{n-1}(x) = \sum_{i=1}^{n-1} b_{i+1} x^i. \tag{6.1.3}$$

Dodatno, označimo, $b_0 = r_0$.

Uvrstimo li (6.1.3) u (6.1.2) i sredimo koeficijente uz odgovarajuće potencije, dobivamo

$$p_n(x) = b_n x^n + (b_{n-1} - x_0\, b_n)x^{n-1} + \cdots + (b_1 - x_0\, b_2)x + b_0 - x_0\, b_1.$$

Za vodeći koeficijent b_n, odmah zaključujemo $b_n = a_n$, a za a_i uz potenciju x^i, $i < n$ je

$$a_i = b_i - x_0 \cdot b_{i+1}, \quad i = n-1, \ldots, 0.$$

Zadnja relacija i veza $b_n = a_n$ pokazuju da b_i možemo izračunati iz b_{i+1} rekurzijom

$$b_i = a_i + x_0 \cdot b_{i+1}.$$

Primijetite da je to relacija istog oblika kao (6.1.1), samo s pomaknutim indeksima, a kako je inicijalno i $b_n = c_{n-1}$, zaključujemo da vrijedi

$$b_i = c_{i-1}, \quad i = 1, \ldots, n.$$

Dakle, koeficijenti koje dobijemo u Hornerovoj shemi su baš koeficijenti polinoma-kvocijenta i ostatka pri dijeljenju polinoma p_n linearnim faktorom $x - x_0$.

Primjer 6.1.3 *Podijelimo*

$$p_5(x) = 2x^5 - x^3 + 4x^2 + 1$$

linearnim polinomom $x + 1$.

Primijetite da je to ista tablica kao u primjeru 6.1.2, *pa imamo*

	2	0	−1	4	0	1
−1	2	−2	1	3	−3	4

.

Odatle lako čitamo

$$2x^5 - x^3 + 4x^2 + 1 = (x+1)\,(2x^4 - 2x^3 + x^2 + 3x - 3) + 4.$$

Provjerite zadnju relaciju množenjem polinoma!

Konačno, napišimo algoritam koji nalazi koeficijente pri dijeljenju polinoma linearnim polinomom.

Algoritam 6.1.3 (Dijeljenje polinoma s $(x - x_0)$)

$b_n := a_n$;
for $i := n - 1$ **downto** 0 **do**
  $b_i := b_{i+1} * x_0 + a_i$;
{polinom-kvocijent: $b_n x^{n-1} + \cdots + b_2 x + b_1$}

6.1.5. Potpuna Hornerova shema

Što se događa ako postupak dijeljenja polinoma linearnim faktorom nastavimo, tj. ponovimo više puta?

Vrijedi

$$\begin{aligned} p_n(x) &= (x - x_0)q_{n-1}(x) + r_0 \\ &= (x - x_0)[(x - x_0)q_{n-2}(x) + r_1] + r_0 \\ &= (x - x_0)^2 q_{n-2}(x) + r_1(x - x_0) + r_0 \\ &= \cdots \\ &= r_n(x - x_0)^n + \cdots + r_1(x - x_0) + r_0. \end{aligned}$$

Dakle, polinom p_n napisan je razvijeno po potencijama od $(x-x_0)$. Koja su značenja r_i? Usporedimo dobiveni oblik s Taylorovim polinomom oko x_0

$$p_n(x) = \sum_{i=0}^{n} \frac{p_n^{(i)}(x_0)}{i!} (x - x_0)^i,$$

pa zaključujemo da vrijedi

$$r_i = \frac{p_n^{(i)}(x_0)}{i!}, \quad 0 \leq i \leq n.$$

Potpuna Hornerova shema računa sve derivacije polinama u zadanoj točki podijeljene pripadnim faktorijelima.

Primjer 6.1.4 *Nađimo sve derivacije polinoma*

$$p_5(x) = 2x^5 - x^3 + 4x^2 + 1$$

u točki -1.

Formirajmo potpunu Hornerovu tablicu.

	2	0	−1	4	0	1
−1	2	−2	1	3	−3	4
−1	2	−4	5	−2	−1	
−1	2	−6	11	−13		
−1	2	−8	19			
−1	2	−10				
−1	2					

Odatle lako čitamo

$$p_5(-1) = 4, \qquad p_5^{(1)}(-1) = -1 \cdot 1! = -1,$$
$$p_5^{(2)}(-1) = -13 \cdot 2! = -26, \qquad p_5^{(3)}(-1) = 19 \cdot 3! = 114,$$
$$p_5^{(4)}(-1) = -10 \cdot 4! = -240, \qquad p_5^{(5)}(-1) = 2 \cdot 5! = 240.$$

Algoritam koji nalazi koeficijente r_i, odnosno koeficijente Taylorovog razvoja zadanog polinoma oko točke x_0, može se napisati koristeći samo jedno polje.

Algoritam 6.1.4 (Taylorov razvoj)

for $i := 0$ **to** n **do**
  $r_i := a_i$;
for $i := 1$ **to** n **do**
  for $j := n - 1$ **to** $i - 1$ **do**
    $r_j := r_j + x_0 * r_{j+1}$;

6.1.6. "Hornerova shema" za interpolacijske polinome

Kao što ćemo vidjeti, kod izvrednjavanja interpolacijskog polinoma u Newtonovoj formi, treba izračunati izraz oblika

$$p_n(x) = a_n (x - x_0)(x - x_1) \cdots (x - x_{n-1}) + a_{n-1} (x - x_0)(x - x_1) \cdots (x - x_{n-2}) + \cdots + a_1 (x - x_0) + a_0,$$

pri čemu su točke x_i točke interpolacije, a x točka u kojoj želimo izračunati vrijednost polinoma.

Algoritam kojim se vrši izvrednjavanje je vrlo sličan Hornerovoj shemi. Ako označimo $y_i = x - x_i$ i postavimo zagrade kao u Hornerovoj shemi, imamo

$$p_n(x) = (\cdots((a_n y_{n-1} + a_{n-1}) y_{n-2} + a_{n-2}) y_{n-3} + \cdots + a_1) y_0 + a_0.$$

Tako smo dobili "Hornerovu shemu" za interpolacijske polinome.

Algoritam 6.1.5 ("Hornerova shema" za interpolacijske polinome)

$sum := a_n$;
for $i := n-1$ **downto** 0 **do**
  $sum := sum * (x - x_i) + a_i$;
$\{p_n(x) = sum\}$

Dakle, Hornerovu shemu možemo iskoristiti i za ovakav prikaz polinoma, koji niju razvijen u standardnoj bazi.

6.1.7. Hornerova shema za realni polinom i kompleksni argument

Sve što smo dosad izvodili, vrijedi općenito, ako su koeficijenti polinoma i točka x_0 iz istog polja, $\mathbb{R}$ ili $\mathbb{C}$.

Ako želimo izračunati vrijednost realnog polinoma (tj. polinoma s realnim koeficijentima) u kompleksnoj točki, to se može napraviti uz određenu uštedu.

Neka je

$$z_0 = x_0 + iy_0.$$

Tvrdimo da postoji realan polinom s_2 stupnja 2, s vodećim koeficijentom 1, takav da je $s_2(z_0) = 0$. Dokaz je jednostavan i može se pokazati na više načina. Jedan je posljedica osnovnog teorema algebre, jer ako je $x_0 + iy_0$ nultočka polinoma s realnim koeficijentima, onda je to i $x_0 - iy_0$, tj. kod realnog polinoma kompleksne nultočke dolaze u konjugirano-kompleksnim parovima,

$$s_2(x) = (x - x_0 - iy_0)\,(x - x_0 + iy_0).$$

Dokaz se može provesti i direktno, korištenjem svojstava konjugiranja

$$\overline{z_1 + z_2} = \overline{z_1} + \overline{z_2},$$
$$\overline{z_1 \cdot z_2} = \overline{z_1} \cdot \overline{z_2}.$$

Označimo koeficijente polinoma s_2 sa

$$s_2(x) = x^2 + px + q.$$

Konjugiranjem $s_2(z_0) = 0$ izlazi

$$0 = \overline{s_2(z_0)} = \overline{z_0^2 + pz_0 + q} = \overline{z_0^2} + \overline{pz_0} + q = \overline{z_0}^2 + p\overline{z_0} + q = s_2(\overline{z_0}).$$

Dakle, $s_2(z_0) = 0$ vrijedi ako i samo ako vrijedi $s_2(\overline{z_0}) = 0$, tj. realne koeficijente od s_2 možemo napisati korištenjem realnog i imaginarnog dijela z_0

$$s_2(x) = x^2 + px + q = (x - x_0 - iy_0)\,(x - x_0 + iy_0),$$

tj.

$$p = -2x_0, \quad q = x_0^2 + y_0^2.$$

Po analogiji, podijelimo polazni realni polinom polinomom s_2. Ostatak pri dijeljenju više nije samo konstanta, nego može biti i polinom r stupnja 1

$$p_n(x) = s_2(x)q_{n-2}(x) + r(x).$$

Napišimo prošlu relaciju u pogodnoj formi, tj. napišimo na "čudan" način polinom r

$$p_n(x) = (x^2 + px + q)\, q_{n-2}(x) + b_1(x + p) + b_0. \tag{6.1.4}$$

Izaberimo oznaku za koeficijente polinoma q_{n-2}, ovaj put s indeksima pomaknutim za 2

$$q_{n-2}(x) = \sum_{i=0}^{n-2} b_{i+2}x^i. \tag{6.1.5}$$

Uvrštavanjem (6.1.5) u (6.1.4) i uspoređivanjem koeficijenata uz odgovarajuće potencije, dobivamo:

$$\begin{aligned}\sum_{i=0}^{n} a_i x^i &= \sum_{i=0}^{n-2} b_{i+2}x^{i+2} + p\sum_{i=0}^{n-2} b_{i+2}x^{i+1} + q\sum_{i=0}^{n-2} b_{i+2}x^i + b_1(x+p) + b_0 \\ &= \sum_{i=2}^{n} b_i x^i + p\sum_{i=1}^{n-1} b_{i+1}x^i + q\sum_{i=0}^{n-2} b_{i+2}x^i + (b_1x + b_0) + pb_1 \\ &= \sum_{i=0}^{n} b_i x^i + p\sum_{i=0}^{n-1} b_{i+1}x^i + q\sum_{i=0}^{n-2} b_{i+2}x^i.\end{aligned}$$

Definiramo li dodatno $b_{n+1} = b_{n+2} = 0$, onda prethodnu relaciju možemo pisati kao

$$\sum_{i=0}^{n} a_i x^i = \sum_{i=0}^{n} \left(b_i + pb_{i+1} + qb_{i+2}\right) x^i.$$

Uspoređivanjem koeficijenata lijeve i desne strane, izlazi rekurzivna relacija

$$a_i = b_i + pb_{i+1} + qb_{i+2}, \quad i = n, \ldots, 0,$$

uz start $b_{n+1} = b_{n+2} = 0$. Drugim riječima, koeficijente b_i računamo iz dva koja odgovaraju susjednim višim potencijama,

$$b_i = a_i - pb_{i+1} - qb_{i+2}, \quad i = n, \ldots, 0, \tag{6.1.6}$$

ponovno, uz $b_{n+1} = b_{n+2} = 0$.

Što je rezultat? Ako tražimo rezultat dijeljenja kvadratnim polinomom, onda moramo izračunati sve koeficijente b_i, s tim da će indeksi $i = 2, \ldots, n$ dati koeficijente

polinoma q_{n-2}, a b_1 i b_0 bit će ostaci pri dijeljenju napisani u formi (6.1.4). Ako želimo ostatak napisan u standardnoj bazi, onda ćemo r napisati kao

$$r(x) = b_1 x + (b_0 + b_1 p),$$

tj. na kraju računa, kad izračunamo sve koeficijente b_i, postavit ćemo

$$b_0 := b_0 + b_1 p$$

Sljedeći algoritam računa kvocijent polinoma p_n i kvadratnog polinoma s_2 u standardnoj bazi. Koeficijenti p i q su tada ulazni podaci, bez ikakvih ograničenja, tj. q ne mora biti nenegativan.

Algoritam 6.1.6 (Dijeljenje polinoma kvadratnim polinomom)

$b_n := a_n$;
$b_{n-1} := a_n - p * b_n$;
for $i := n - 2$ **downto** 0 **do**
  $b_i := a_i - p * b_{i+1} - q * b_{i+2}$;
$b_0 := b_0 + p * b_1$;

Konačno, vratimo se zadatku od kojeg smo krenuli. Ako želimo izračunati vrijednost polinoma p_n u točki $x_0 + iy_0$, onda nam ne trebaju svi b_i-ovi. Promotrimo relaciju (6.1.4). Uočimo da je $s_2(x_0 + iy_0) = 0$ (tako je definiran), pa se (6.1.4) svede na

$$p_n(x_0 + iy_0) = b_1(x_0 + iy_0 + p) + b_0. \tag{6.1.7}$$

To znači da nam trebaju samo koeficijenti b_1 i b_0 da bismo znali izračunati vrijednost u točki z_0. Relaciju (6.1.7) možemo i ljepše napisati jer znamo da je

$$p + x_0 = -2x_0 + x_0 = -x_0.$$

Odmah se dobije

$$p_n(x_0 + iy_0) = b_1(-x_0 + iy_0) + b_0 = (b_0 - x_0 b_1) + iy_0 b_1,$$

pa vidimo da za računanje vrijednosti realnog polinoma u kompleksnoj točki, ne moramo pamtiti čitavo polje koeficijenata b_i, nego samo "trenutna" tri koji su nam potrebni u rekurziji (6.1.6)

Algoritam 6.1.7 (Vrijednost realnog polinoma u kompleksnoj točki)

$p := -2 * x_0$;
$q := x_0^2 + y_0^2$;
$b_1 := 0$;
$b_0 := a_n$;
for $i := n - 1$ **downto** 0 **do**

begin
$b_2 := b_1$;
$b_1 := b_0$;
$b_0 := a_i - p * b_1 - q * b_2$;
end;
$\mathrm{Re}(p_n(x_0 + iy_0)) := b_0 - x_0 * b_1$;
$\mathrm{Im}(p_n(x_0 + iy_0)) := y_0 * b_1$;

Kad prebrojimo **realne** operacije u ovom algoritmu, dobivamo

$$(2n + 4) \text{ množenja } + (2n + 3) \text{ zbrajanja.}$$

S obzirom da jedno kompleksno zbrajanje zahtijeva 2 realna zbrajanja, a kompleksno množenje realiziramo na standardan način 4 realna množenja i 2 realna zbrajanja, obična Hornerova shema za kompleksni polinom u kompleksnoj točki ima

$$4n \text{ množenja } + 4n \text{ zbrajanja.}$$

U ne tako davnoj prošlosti duljina trajanja množenja u računalu bila je dosta dulja nego duljina trajanja zbrajanja, pa su se ljudi često dovijali "alkemiji" pretvaranja množenja u zbrajanja. Kod množenja kompleksnih brojeva to je lako, jer vrijedi

$$(a + bi)\,(c + di) = (ac - bd) + (ad + bc)i = (ac - bd)[(a + b)\,(c + d) - (ac + bd)]i.$$

Uočite da ova posljednja forma ima samo 3 množenja i 5 zbrajanja (produkti ac i bd se čuvaju kad se izračunaju).

Konačno, što ako želimo izračunati vrijednost polinoma p_n u točki $\bar{z}_0 = x_0 - iy_0$? Moramo li ponovno provoditi postupak? Odgovor je ne. Naime, $\bar{z}_0$ je, također nultočka od s_2, pa je u terminima izlaznih podataka iz algoritma 6.1.7,

$$p_n(x_0 - iy_0) = b_1(-x_0 - iy_0) + b_0 = (b_0 - x_0 b_1) - iy_0 b_1,$$

tj. razlika obzirom na $p_n(x_0 + iy_0)$ je samo u suprotnom predznaku imaginarnog dijela. Dakle, treba nam još jedna dodatna operacija da istovremeno izračunamo vrijednost polinoma u z_0 i $\bar{z}_0$.

Ako želimo "na ruke" izračunati vrijednost realnog polinoma u kompleksnoj točki, onda koristimo tablicu vrlo sličnu običnoj Hornerovoj shemi.

	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_2	a_1	a_0
q			$-qb_n$	$\cdots$	$-qb_4$	$-qb_3$	$-qb_2$
p		$-pb_n$	$-pb_{n-1}$	$\cdots$	$-pb_3$	$-pb_2$	$-pb_1$
$+$	b_n	b_{n-1}	b_{n-2}	$\cdots$	b_2	b_1	b_0

.

Primjer 6.1.5 *Nađimo vrijednost polinoma*

$$p_3(x) = x^3 + 8x^2 + 1$$

u točkama $2 \pm i$.

Uzmimo točku $2 + i$, *pa je* $x_0 = 2$, $y_0 = 1$, *a* $p = -4$, $q = 5$. *Formirajmo tablicu.*

	1	8	0	1
5			-5	-60
-4		4	48	172
+	1	12	43	113

.

Posljednja dva koeficijenta su $b_1 = 43$, $b_0 = 113$, *pa je*

$$\mathrm{Re}(p_3(2+i)) = b_0 - b_1 a = 113 - 86 = 27, \quad \mathrm{Im}(p_3(2+i)) = b_1 b = 43.$$

Dakle, imamo

$$p_3(2+i) = 27 + 43i, \quad p_3(2-i) = 27 - 43i.$$

6.1.8. Računanje parcijalnih derivacija kompleksnog polinoma

Kompleksni polinom u varijabli z, može se zapisati pomoću dva polinoma u i v u dvije realne varijable x i y, gdje je $z = x + iy$. Vrijedi

$$p_n(z) = u(x, y) + iv(x, y), \quad z = x + iy.$$

Funkcije u i v možemo interpretirati i kao realne funkcije kompleksnog argumenta

$$u(z) = \mathrm{Re}\,(p_n(z)), \quad v(z) = \mathrm{Im}\,(p_n(z)),$$

pa, prema prošlom odjeljku, imamo algoritam za nalaženje njihovih vrijednosti.

Da bismo oponašali dijeljenje polinoma linearnim polinomom oblika $x - x_0$, uzmimo da nam i kvadratni polinom ima oblik $x^2 - px - q$ (pa svugdje gdje u prošlom odjeljku piše p treba pisati $-p$, a gdje piše q treba pisati $-q$). Dakle, rastav u produkt kvadratnog polinoma i polinoma stupnja $n - 2$ u ovakvoj notaciji glasi:

$$p_n(x) = s_2(x) q_{n-2}(x) + r(x),$$

gdje je

$$r(x) = b_1(x - p) + b_0.$$

Algoritam za nalaženje koeficijenata u ovoj formulaciji je: start $b_{n+2} = b_{n+1} = 0$, a rekurzija je

$$b_i = a_i + p b_{i+1} + q b_{i+2}, \quad i = n, \ldots, 0. \tag{6.1.8}$$

Pokazali smo da je tada

$$p_n(z_0) = (b_0 - x_0 b_1) + i y_0 b_1.$$

Shvatimo li $p_n(z)$ kao funkciju u odgovarajućoj točki (x, y), onda su realni i imaginarni dio te funkcije

$$u(x, y) = b_0 - x b_1, \quad v(x, y) = y b_1.$$

Parcijalne derivacije funkcija u i v možemo dobiti korištenjem deriviranja složenih funkcija. Polinomi su analitičke funkcije, pa za njihove parcijalne derivacije vrijede Cauchy–Riemannovi uvjeti

$$\frac{\partial u}{\partial x}(x, y) = \frac{\partial v}{\partial y}(x, y), \quad \frac{\partial u}{\partial y}(x, y) = -\frac{\partial v}{\partial x}(x, y).$$

Zbog toga je dovoljno pronaći samo ili parcijalne derivacije jedne od funkcija ili parcijalne derivacije po jednoj varijabli. Iskoristimo li da vrijedi

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial p}\frac{\partial p}{\partial y} + \frac{\partial}{\partial q}\frac{\partial q}{\partial y},$$

dobivamo (izostavljajući pisanje točke u kojoj deriviramo)

$$\begin{aligned} \frac{\partial u}{\partial y} &= \frac{\partial b_0}{\partial y} - x\frac{\partial b_1}{\partial y} = \frac{\partial b_0}{\partial q}(-2y) - x\frac{\partial b_1}{\partial q}(-2y) = 2y\Big(x\frac{\partial b_1}{\partial q} - \frac{\partial b_0}{\partial q}\Big) \\ \frac{\partial v}{\partial y} &= b_1 + y\frac{\partial b_1}{\partial y} = b_1 + y\frac{\partial b_1}{\partial q}(-2y) = b_1 - 2y^2\frac{\partial b_1}{\partial q}. \end{aligned} \tag{6.1.9}$$

Dakle, da bismo izračunali vrijednosti parcijalnih derivacija u nekoj točki, dovoljno je znati

$$\frac{\partial b_0}{\partial p}, \frac{\partial b_0}{\partial q}, \frac{\partial b_1}{\partial p}, \frac{\partial b_1}{\partial q}.$$

Uvedimo oznake

$$c_i = \frac{\partial b_i}{\partial p}, \quad d_i = \frac{\partial b_i}{\partial q}, \quad i = n+2, \ldots, 0.$$

Deriviramo li formalno relaciju (6.1.8) prvo po p, a zatim po q, dobivamo

$$\begin{aligned} c_{n+2} &= c_{n+1} = 0 \\ c_i &= b_{i+1} + p c_{i+1} + q c_{i+2}, \quad i = n, \ldots, 0 \\ d_{n+2} &= d_{n+1} = 0 \\ d_i &= b_{i+2} + p d_{i+1} + q d_{i+2}, \quad i = n, \ldots, 0 \end{aligned}$$

Odavde odmah vidimo da c_i-ovi i d_i-ovi tvore istu rekurziju, samo s indeksom translatiranim za 1, tj. vrijedi

$$c_{i+1} = d_i, \quad i = n+1, n, n-1, \ldots, 0.$$

Zbog toga, umjesto dvije rekurzije za koeficijente možemo raditi samo s jednom, uz paralelno računanje b_i.

Algoritam dobivanja parcijalnih derivacija zove se algoritam Bairstowa.

Algoritam 6.1.8 (Algoritam Bairstowa)

$b_1 := a_n;$
$b_0 := a_{n-1} + p * b_1;$
$c_2 := 0;$
$c_1 := 0;$
$c_0 := a_n;$
for $i := n - 2$ **downto** 0 **do**
  begin
  $b_2 := b_1;$
  $b_1 := b_0;$
  $b_0 := a_i + p * b_1 + q * b_2;$
  $c_2 := c_1;$
  $c_1 := c_0;$
  $c_0 := b_1 + p * c_1 + q * c_2;$
  end;

Relacija (6.1.9) odmah daje kako se iz c-ova i b-ova dobivaju parcijalne derivacije

$$\frac{\partial u}{\partial y}(x_0, y_0) = -\frac{\partial v}{\partial x}(x_0, y_0) = 2y_0(x_0 d_1 - d_0) = 2y_0(x_0 c_2 - c_1)$$
$$\frac{\partial v}{\partial y}(x_0, y_0) = \frac{\partial u}{\partial x}(x_0, y_0) = b_1 - 2y_0^2\, d_1 = b_1 - 2y_0^2\, c_2.$$

Motivacija za ovaj algoritam potječe iz modifikacije Newtonove metode za traženje realnih nultočaka polinoma. Ako želimo naći kompleksne nultočke realnog polinoma, prvo treba izlučiti kvadratni polinom s konjugirano kompleksnim parom nultočaka. Ta generalizacija Newtonove metode zove se metoda Newton–Bairstow.

6.2. Generalizirana Hornerova shema

U prošlom odjeljku napravili smo nekoliko algoritama za izvrednjavanje polinoma i njegovih derivacija u zadanoj točki. Te algoritme, koji su egzaktni u egzaktnoj aritmetici, možemo koristiti i kao **približne** algoritme za izvrednjavanje redova potencija, tj. analitičkih funkcija.

Pretpostavimo da se funkcija f u okolini neke točke x_0 (u $\mathbb{R}$ ili $\mathbb{C}$) može razviti

u red potencija oblika

$$f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n, \tag{6.2.1}$$

s tim da znamo taj red konvergira prema f na toj okolini od x_0. Dodatno, pretpostavimo da znamo sve koeficijente a_n u ovom razvoju, u smislu da ih možemo brzo i točno izračunati. Naravno, ovu beskonačnu sumu ne možemo efektivno algoritamski izračunati, jer zahtijeva beskonačan broj aritmetičkih operacija.

Međutim, konačne komade ovog razvoja možemo iskoristiti za aproksimaciju funkcije f na toj okolini. Iz konvergencije razvoja po točkama odmah slijedi da, za bilo koju unaprijed zadanu točnost $\varepsilon > 0$, postoji $N \in \mathbb{N}$ takav da je

$$f_N(x) = \sum_{n=0}^{N} a_n (x - x_0)^n, \tag{6.2.2}$$

aproksimacija za $f(x)$ s greškom manjom od ε. Nije bitno da li grešku mjerimo u apsolutnom ili relativnom smislu, osim ako je $f(x) = 0$. Sasvim općenito, potrebna duljina razvoja N ovisi i o ε i o x. No, ako se sjetimo da redovi potencija konvergiraju uniformno na kompaktima, možemo postići i uniformnu aproksimaciju s točnošću ε na takvim kompaktima, pa N onda ovisi samo o ε.

Kad uzmemo u obzir da ionako **približno** računamo u aritmetici računala, ovim pristupom možemo bitno povećati klasu funkcija s kojima možemo računati. Ako je greška ε dovoljno mala, recimo reda veličine osnovne greške zaokruživanja u, onda je pripadna aproksimacija $f_N(x)$ gotovo jednako dobra kao i $f(x)$.

Algoritam za računanje $f_N(x)$ u zadanoj točki x bitno ovisi u tome da li N znamo unaprijed ili ne. Ako ga **ne znamo**, onda se obično koristi sumacija unaprijed, sve dok se izračunata suma ne stabilizira na zadanu točnost. Koliko to može biti opasno, već smo vidjeli u primjeru za $\sin x$. Zbog toga se sumacija unaprijed koristi samo kao "zadnje utočište".

Vrlo često se N može unaprijed odrediti iz analitičkih svojstava funkcije f, tako da dobijemo uniformnu aproksimaciju s točnošću ε na nekom kompaktu. Obično se za taj kompakt uzima neki segment u $\mathbb{R}$, odnosno neki krug u $\mathbb{C}$. Čak nije jako bitno da N bude "savršen", tj. najmanji mogući, ako je takav N teško izračunati. Katkad je sasvim dobra i približna vrijednost za N. Tada je f_N polinom poznatog stupnja N i možemo koristiti Hornerovu shemu i njene varijacije za računanje $f_N(x)$.

Trenutno ne ulazimo u to kako se nalaze takve aproksimacije. Tome ćemo posvetiti punu pažnju u poglavlju o aproksimacijama. Zasad recimo samo to da se izbjegava direktno korištenje redova potencija (6.2.1) i pripadnih polinomnih aproksimacija u obliku (6.2.2), zbog loše uvjetovanosti sustava funkcija

$$\{1, (x - x_0), (x - x_0)^2, \ldots, (x - x_0)^n, \ldots\}$$

i nejednolikog rasporeda pogreške $e(x) = f(x) - f_N(x)$ na domeni aproksimacije.

Umjesto reda potencija (6.2.1), standardno se koriste razvoji oblika

$$f(x) = \sum_{n=0}^{\infty} a_n p_n(x), \tag{6.2.3}$$

gdje je $\{p_n \mid n \in \mathbb{N}_0\}$ neki **ortogonalni** sustav funkcija na domeni aproksimacije. U aproksimaciji elementarnih i "manje elementarnih" tzv. specijalnih funkcija vrlo često se koriste tzv. Čebiševljevi polinomi, zbog skoro jednolikog rasporeda greške na domeni. Kasnije ćemo pokazati i algoritam za nalaženje takve "kvazi–uniformne" aproksimacije iz poznatog reda potencija (tzv. Čebiševljeva ekonomizacija).

Razvoj funkcije f u red oblika (6.2.3) je očita generalizacija reda potencija. Njega, također, po istom principu, možemo iskoristiti za aproksimaciju funkcije f, ako znamo da on konvergira prema f na nekoj domeni. "Rezanjem" reda dobivamo aproksimaciju funkcije f

$$f_N(x) = \sum_{n=0}^{N} a_n p_n(x), \tag{6.2.4}$$

što je očita generalizacija polinoma iz (6.2.2). Naravno, da bismo izračunali $f_N(x)$ moramo znati sve koeficijente a_n i sve funkcije p_n. Međutim, u većini primjena **nemamo** direktnu "formulu" za računanje vrijednosti $p_n(x)$ u zadanoj točki x, za sve $n \in \mathbb{N}_0$. Umjesto toga, **znamo** da funkcije p_n zadovoljavaju neku, relativno jednostavnu rekurziju po n. Funkcije p_n ne moraju biti polinomi. Dovoljno je da ih možemo rekurzivno računati!

Pristup računanju vrijednosti $f_N(x)$ je isti kao i ranije. Ako unaprijed ne znamo N, onda se sumacija vrši unaprijed, a $p_n(x)$ računa redom iz rekurzije. S druge strane, iz teorije aproksimacija, vrlo često je moguće unaprijed naći koliko članova N treba uzeti za (uniformnu) zadanu točnost. Tada bi bilo zgodno koristiti neku generalizaciju Hornerove sheme za brzo izvrednjavanje f_N oblika (6.2.4) i to je cilj ovog odjeljka.

6.2.1. Izvrednjavanje rekurzivno zadanih funkcija

Budući da ortogonalni polinomi zadovoljavaju tročlane, homogene rekurzije, a vrlo se često koriste, posebnu pažnju posvetit ćemo baš takvim rekurzijama. Osim toga, tročlane rekurzije istog općeg oblika vrijede i za mnoge specijalne funkcije koje ne moraju biti ortogonalne. Zato pretpostavljamo da funkcije p_n, za $n \in \mathbb{N}_0$, zadovoljavaju rekurziju oblika

$$p_{n+1}(x) + \alpha_n(x)p_n(x) + \beta_n(x)p_{n-1}(x) = 0, \quad n = 1, 2, \ldots, \tag{6.2.5}$$

s tim da su poznate "početne" funkcije p_0 i p_1, i sve funkcije α_n, β_n, za $n \in \mathbb{N}$, koje su obično jednostavnog oblika.

Primijetite da potencije $p_n(x) = x^n$ zadovoljavaju dvočlanu homogenu rekurziju

$$p_n(x) - xp_{n-1} = 0, \quad n \in \mathbb{N},$$

uz $p_0(x) = 1$, pa je (6.2.5) zaista generalizacija polinomnog slučaja. Sličan algoritam za brzo izvrednjavanje f_N može se napraviti i kad p_n zadovoljavaju četveročlane ili višečlane rekurzije, ali se takve rekurzije rijetko pojavljuju u praksi.

Algoritam je vrlo sličan izvrednjavanju realnog polinoma u kompleksnoj točki. Definiramo rekurziju za koeficijente

$$\begin{aligned} B_{N+2} &= B_{N+1} = 0, \\ B_n &= a_n - \alpha_n(x)B_{n+1} - \beta_{n+1}(x)B_{n+2}, \quad n = N, \ldots, 0. \end{aligned} \tag{6.2.6}$$

Uvrštavanjem u formulu (6.2.4) za $f_N(x)$, dobivamo

$$\begin{aligned} f_N(x) &= \sum_{n=0}^{N} a_n p_n(x) = \sum_{n=0}^{N} \left(B_n + \alpha_n(x)B_{n+1} + \beta_{n+1}(x)B_{n+2}\right) p_n(x) \\ &= \sum_{n=-1}^{N-1} B_{n+1}p_{n+1}(x) + \sum_{n=0}^{N} \alpha_n(x)B_{n+1}p_n(x) + \sum_{n=1}^{N+1} \beta_n(x)B_{n+1}p_{n-1}(x) \\ &= \sum_{n=1}^{N-1} B_{n+1}(p_{n+1}(x) + \alpha_n(x)p_n(x) + \beta_n(x)p_{n-1}(x)) \\ &\qquad + B_0p_0(x) + B_1p_1(x) + \alpha_0(x)B_1p_0(x) \\ &= B_0p_0(x) + B_1p_1(x) + \alpha_0(x)B_1p_0(x). \end{aligned}$$

Pripadni silazni algoritam izvrednjavanja ima sljedeći oblik.

Algoritam 6.2.1 (Generalizirana Hornerova shema za $f_N(x)$)

$B_1 := 0;$
$B_0 := a_N;$
for $k := N - 1$ **downto** 0 **do**
  begin;
  $B_2 := B_1;$
  $B_1 := B_0;$
  $B_0 := a_k - \alpha_k(x) * B_1 - \beta_{k+1}(x) * B_2;$
  end;
$f_N(x) := B_0 * p_0(x) + B_1 * (p_1(x) + \alpha_0(x) * p_0(x));$

Ako trebamo izračunati i derivaciju $f'_N(x)$, do pripadnog algoritma možemo doći deriviranjem relacije (6.2.4)

$$f'_N(x) = \sum_{n=0}^{N} a_n p'_n(x),$$

i deriviranjem rekurzije (6.2.5), tako da dobijemo i rekurziju za funkcije p'_n. Pokušajte to napraviti sami.

Međutim, postoji i jednostavniji put, deriviranjem rekurzije (6.2.6), slično kao u algoritmu Bairstowa. Ovdje je to još bitno jednostavnije, jer imamo samo jednu varijablu. Koeficijente B_n shvatimo kao funkcije od x, što oni zaista i jesu. Zatim deriviramo (6.2.6), s tim da B'_n označava derivaciju od B_n po x, u točki x. Takvim "formalnim" deriviranjem dobivamo rekurziju za koeficijente B'_n.

$$\begin{aligned}
&B_{N+2} = B_{N+1} = 0,\\
&B'_{N+2} = B'_{N+1} = 0,\\
&\quad B_n = a_n - \alpha_n(x)B_{n+1} - \beta_{n+1}(x)B_{n+2}, \quad n = N, \ldots, 0.\\
&\quad B'_n = -\alpha'_n(x)B_{n+1} - \alpha_n(x)B'_{n+1}\\
&\qquad\qquad - \beta'_{n+1}(x)B_{n+2} - \beta_{n+1}(x)B'_{n+2}, \quad n = N, \ldots, 0.
\end{aligned}$$

Odavde odmah vidimo da je i $B'_N = 0$. Uz standardnu oznaku

$$b_n = -\alpha'_n(x)B_{n+1} - \beta'_{n+1}(x)B_{n+2}, \quad n = N, \ldots, 0,$$

s tim da je očito $b_N = 0$, rekurziju za B'_n možemo napisati u obliku

$$B'_n = b_n - \alpha_n(x)B'_{n+1} - \beta_{n+1}(x)B'_{n+2}, \quad n = N, \ldots, 0,$$

što ima skoro isti oblik kao i rekurzija za B_n, osim zamjene a_n u b_n. Konačni rezultat, također, dobivamo deriviranjem ranijeg konačnog rezultata

$$f_N(x) = B_0 p_0(x) + B_1(p_1(x) + \alpha_0(x)p_0(x)),$$

odakle slijedi

$$\begin{aligned}
f'_N(x) = {} & B_0 p'_0(x) + B'_0 p_0(x) + B_1(p'_1(x) + \alpha'_0(x)p_0(x) + \alpha_0(x)p'_0(x)),\\
& + B'_1(p_1(x) + \alpha_0(x)p_0(x)).
\end{aligned}$$

Dakle, da bismo izračunali $f'_N(x)$, dovoljno je znati samo derivacije "početnih" funkcija p'_0 i p'_1, koje su obično jednostavne. Naravno, treba znati i derivacije α'_n, β'_n funkcija iz polazne tročlane rekurzije, ali i one su obično jednostavne. Rekurzija za derivacije p'_n nas uopće ne zanima, iako ju nije teško napisati.

Vidimo da nam za računanje $f'_N(x)$ treba i rekurzija za računanje $f_N(x)$, pa se te dvije vrijednosti obično zajedno računaju, a ne svaka posebno. Tada rekurzije za B_n i B'_n provodimo u istoj petlji. Konačni rezultati izgledaju komplicirano, ali kad u njih uvrstimo konkretne objekte, vrlo rijetko ostanu svi članovi. Obično se te formule svedu na

$$f_N(x) = B_0, \quad f'_N(x) = B'_0.$$

Algoritam 6.2.2 (Generalizirana Hornerova shema za $f_N(x)$ i $f'_N(x)$)

$B_1 := 0;$
$B_0 := a_N;$
$B'_1 := 0;$
$B'_0 := 0;$
for $k := N-1$ **downto** 0 **do**
  begin;
  $B_2 := B_1;$
  $B_1 := B_0;$
  $B_0 := a_k - \alpha_k(x) * B_1 - \beta_{k+1}(x) * B_2;$
  $B'_2 := B'_1;$
  $B'_1 := B'_0;$
  $b := -\alpha'_k(x) * B_1 - \beta'_{k+1}(x) * B_2;$
  $B'_0 := b - \alpha_k(x) * B'_1 - \beta_{k+1}(x) * B'_2;$
  end;
$f_N(x) := B_0 * p_0(x) + B_1 * (\alpha_0(x) * p_0(x) + p_1(x));$
$f'_N(x) := B_0 * p'_0(x) + B'_0 * p_0(x) + B_1 * (p'_1(x) + \alpha'_0(x) * p_0(x) + \alpha_0(x) * p'_0(x))$
$\qquad + B'_1 * (p_1(x) + \alpha_0(x) * p_0(x));$

Istim putem možemo izvesti i rekurzije za računanje viših derivacija $f_N^{(k)}(x)$, za $k \geq 2$. Zanimljivo je da u praksi to gotovo nikada nije potrebno. Razlog leži u činjenici da gotovo sve "korisne" familije funkcija p_n, $n \in \mathbb{N}$, zadovoljavaju neke diferencijalne jednadžbe **drugog** reda, s parametrom n. Jasno je da tada treba koristiti odgovarajuću diferencijalnu jednadžbu za računanje $f''_N(x)$, ali i to je vrlo rijetko potrebno.

Čak i algoritam za derivacije se rijetko koristi. Naime, ako znamo naći, tj. izračunati koeficijente a_n u prikazu

$$f(x) = \sum_{n=0}^{\infty} a_n p_n(x),$$

s dovoljnom točnošću, za $n \leq N$, tako da je pripadni $f_N(x)$ dovoljno dobra aproksimacija, onda se **ne isplati** koristiti

$$f'_N(x) = \sum_{n=0}^{N} a_n p'_n(x)$$

kao aproksimaciju za $f'(x)$, jer ona obično ima manju točnost od aproksimacije za f. Puno je bolje izračunati koeficijente a'_n (to nisu derivacije) u pravom razvoju derivacije f' po **istim** funkcijama p_n, a ne po njihovim derivacijama. Dakle, za f' koristimo aproksimaciju oblika

$$f'_{N'}(x) = \sum_{n=0}^{N'} a'_n p_n(x),$$

koja ne mora imati istu duljinu, ali zato ima željenu točnost.

Složenost ovih algoritama ključno ovisi o složenosti računanja svih potrebnih funkcija — p_0, p_1, α_n, β_n, i njihovih derivacija, pa je besmisleno brojati pojedinačne aritmetičke operacije na nivou općeg algoritma.

U praktičnim aproksimacijama se najčešće koriste tzv. ortogonalne familije funkcija p_n, koje čine ortogonalnu bazu u nekom prostoru funkcija, obzirom na neki skalarni produkt na tom prostoru. Vrlo često je p_n polinom stupnja n, za svaki $n \in \mathbb{N}_0$. Neke primjere klasičnih ortogonalnih polinoma i pripadnih rekurzija dajemo nešto kasnije.

Međutim, već smo rekli da funkcije p_n ne moraju biti polinomi i prvi primjer je baš tog tipa.

6.2.2. Izvrednjavanje Fourierovih redova

Za aproksimaciju periodičkih funkcija standardno koristimo Fourierove redove. Pretpostavimo, radi jednostavnosti, da je f periodička funkcija na segmentu $[-\pi, \pi]$. Tada, uz relativno blage pretpostavke, funkciju f možemo razviti u Fourierov red oblika

$$\sum_{n=0}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx).$$

Umjesto a_0, standardno se piše $a_0/2$, ali to nije bitna razlika. Zanemarimo trenutno pitanje konvergencije ovog reda i značenja njegove sume. Uočimo samo da ove trigonometrijske funkcije tvore ortogonalan sustav funkcija, obzirom na skalarni produkt definiran integralom.

Pretpostavimo da su nam koeficijenti a_n i b_n poznati. Naš zadatak je izračunati aproksimaciju oblika

$$f_N(x) = \sum_{n=0}^{N} a_n \cos(nx) + \sum_{n=1}^{N} b_n \sin(nx),$$

gdje je N unaprijed zadan. Ovakav izraz se često zove i trigonometrijski polinom. Vidimo da se on sastoji iz dva dijela, kosinusnog i sinusnog, pa ćemo tako i sastaviti algoritam. Usput, sjetimo se da Fourierov red parne funkcije $f(x) = f(-x)$ ima samo kosinusni dio, a Fourierov red neparne funkcije $f(x) = -f(-x)$ ima samo sinusni dio razvoja.

Pretpostavimo stoga da je f parna funkcija i trebamo izračunati aproksimaciju oblika

$$f_N(x) = \sum_{n=0}^{N} a_n \cos(nx).$$

U direktnoj sumaciji trebamo N računanja funkcije cos, za $\cos(nx)$, uz $n \geq 1$. Iako to danas više ne traje pretjerano dugo, možemo naći i bolji algoritam, koji treba samo jedno jedino računanje funkcije cos.

Da bismo dobili polazni oblik aproksimacije (6.2.4) iz generalizirane Hornerove sheme, očito treba definirati

$$p_n(x) = \cos(nx).$$

Fali nam još samo tročlana homogena rekurzija za ove funkcije. Međutim, i to ide lako, ako se sjetimo formule koja sumu kosinusa pretvara u produkt

$$\cos a + \cos b = 2\cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right).$$

Dovoljno je uzeti $a = (n+1)x$ i $b = (n-1)x$. Dobivamo

$$\cos((n+1)x) + \cos((n-1)x) = 2\cos(nx)\cos x,$$

pa tražena rekurzija ima oblik

$$p_{n+1}(x) - 2\cos x\, p_n(x) + p_{n-1}(x) = 0, \quad n \in \mathbb{N},$$

odakle slijedi da u općoj rekurziji (6.2.5) treba uzeti

$$\alpha_n(x) = -2\cos x, \quad \beta_n(x) = 1, \quad n \in \mathbb{N}.$$

Vidimo da $\alpha_n(x)$ i $\beta_n(x)$ ne ovise o n, a $\beta_n(x)$ ne ovisi ni o x, već je konstanta.

Rekurzija (6.2.6) za B_n ima oblik

$$\begin{aligned} B_{N+2} &= B_{N+1} = 0, \\ B_n &= a_n + 2\cos x\, B_{n+1} - B_{n+2}, \quad n = N, \ldots, 0. \end{aligned}$$

Početne funkcije su $p_0(x) = 1$ i $p_1(x) = \cos x$, pa je konačni rezultat

$$\begin{aligned} f_N(x) &= B_0 p_0(x) + B_1(p_1(x) + \alpha_0(x)p_0(x)) \\ &= B_0 \cdot 1 + B_1(\cos x - 2\cos x \cdot 1) \\ &= B_0 - B_1 \cos x. \end{aligned}$$

Sad imamo sve elemente za generaliziranu Hornerovu shemu.

Algoritam 6.2.3 (Fourierov "red" parne funkcije)

$B_1 := 0;$
$B_0 := a_N;$
$alpha := 2 * \cos x;$
for $k := N - 1$ **downto** 0 **do**
  begin;

$B_2 := B_1;$
$B_1 := B_0;$
$B_0 := a_k + alpha * B_1 - B_2;$
end;
$f_N(x) := B_0 - 0.5 * alpha * B_1;$

Ovaj algoritam zaista "troši" jedan jedini kosinus, pod cijenu jednog množenja s 0.5. Što se stabilnosti tiče, on je podjednako stabilan kao i direktna sumacija. Male vrijednosti $\cos(nx)$ ionako ne dobivamo s malom relativnom, već malom apsolutnom greškom.

Ako trebamo izračunati i derivaciju $f'_N(x)$, za pripadni algoritam trebamo

$$\alpha'_n(x) = 2\sin x, \quad \beta'_n(x) = 0, \quad n \in \mathbb{N}.$$

Onda je

$$b_n = -2\sin x\, B_{n+1}, \quad n = N, \ldots, 0,$$

i

$$B'_n = b_n + 2\cos x\, B'_{n+1} - B'_{n+2}, \quad n = N, \ldots, 0,$$

a formalnim deriviranjem $f_N(x) = B_0 - B_1 \cos x$ dobivamo

$$f'_N(x) = B'_0 - B'_1 \cos x + B_1 \sin x.$$

Dakle, cijeli taj algoritam treba još samo jedan sinus. I tog bismo mogli izbaciti, tako da sinus izrazimo preko kosinusa,

$$\sin x = \pm\sqrt{1 - \cos^2 x},$$

ali to se već ne isplati, jer moramo paziti na znak, a oduzimanje može dovesti do nepotrebnog gubitka točnosti u sinusu.

Pretpostavimo sad da je f neparna funkcija. Trebamo izračunati aproksimaciju oblika

$$f_N(x) = \sum_{n=1}^{N} b_n \sin(nx).$$

Suma ovdje ide od 1, pa treba biti malo oprezan. Zgodniji je zapis

$$f_N(x) = \sum_{n=0}^{N-1} b_{n+1} \sin((n+1)x).$$

Nije baš lijepo ostaviti indeks N u f_N, ali sad je očito da treba definirati

$$p_n(x) = \sin((n+1)x).$$

Zatim koristimo formulu

$$\sin a + \sin b = 2 \sin\left(\frac{a+b}{2}\right) \cos\left(\frac{a-b}{2}\right),$$

i uzmemo $a = (n+2)x$ i $b = nx$. Dobivamo

$$\sin((n+2)x) + \sin(nx) = 2 \sin((n+1)x) \cos x,$$

pa tražena rekurzija ima oblik

$$p_{n+1}(x) - 2 \cos x \, p_n(x) + p_{n-1}(x) = 0, \quad n \in \mathbb{N},$$

što je potpuno isti oblik kao i za parne funkcije, odnosno za $p_n(x) = \cos(nx)$. Dakle, rekurzija za pripadne B_n ima isti oblik, samo starta od $N-1$

$$\begin{aligned} B_{N+1} &= B_N = 0, \\ B_n &= b_{n+1} + 2 \cos x \, B_{n+1} - B_{n+2}, \quad n = N-1, \ldots, 0. \end{aligned}$$

Početne funkcije su $p_0(x) = \sin x$ i $p_1(x) = \sin(2x) = 2 \sin x \cos x$, pa je konačni rezultat

$$\begin{aligned} f_N(x) &= B_0 p_0(x) + B_1(p_1(x) + \alpha_0(x) p_0(x)) \\ &= B_0 \cdot \sin x + B_1(2 \sin x \cos x - 2 \cos x \cdot \sin x) \\ &= B_0 \sin x. \end{aligned}$$

Algoritam možete i sami napisati.

Za opći Fourierov red koji ima i parni i neparni dio, treba spojiti prethodne algoritme. Jedina je neugoda što je neparni za 1 kraći, jer starta s $N-1$. Ako nas to baš jako smeta, onda možemo i malo drugačije postupiti u neparnom dijelu. Umjetno definiramo da je $b_0 = 0$ i pišemo

$$f_N(x) = \sum_{n=0}^{N} b_n \sin(nx).$$

Zatim uzmemo

$$p_n(x) = \sin(nx).$$

Rekurzija za p_n, naravno, ostaje ista, a za B_n sad vrijedi "produljena" rekurzija

$$\begin{aligned} B_{N+1} &= B_N = 0, \\ B_n &= b_n + 2 \cos x \, B_{n+1} - B_{n+2}, \quad n = N, \ldots, 0. \end{aligned}$$

Početne funkcije su $p_0(x) = 0$ i $p_1(x) = \sin x$, pa je konačni rezultat

$$\begin{aligned} f_N(x) &= B_0 p_0(x) + B_1(p_1(x) + \alpha_0(x) p_0(x)) \\ &= B_0 \cdot 0 + B_1(\sin x - 2 \cos x \cdot 0) \\ &= B_1 \sin x. \end{aligned}$$

To pokazuje da B_0 uopće ne treba računati, ali baš to i očekujemo, kad smo rekurziju pomakli za jedan indeks naviše!

Spomenimo na kraju da obje funkcije $\cos(nx)$ i $\sin(nx)$ zadovoljavaju istu diferencijalnu jednadžbu drugog reda

$$y'' + n^2 y = 0.$$

6.2.3. Klasični ortogonalni polinomi

U aproksimacijama i rješavanju diferencijalnih jednadžbi najčešće se susrećemo s pet tipova klasičnih ortogonalnih polinoma. Za polinome

$$\{p_0, p_1, p_2, \ldots, p_n, \ldots\},$$

pri čemu indeks polinoma označava njegov stupanj, reći ćemo da su ortogonalni obzirom na težinsku funkciju w, $w(x) \geq 0$ na intervalu $[a, b]$, ako vrijedi

$$\int_a^b w(x)\, p_m(x)\, p_n(x)\, dx = 0, \quad \text{za } m \neq n.$$

Težinska funkcija određuje sistem polinoma do na konstantni faktor u svakom od polinoma. Izbor takvog faktora zove se još i standardizacija ili normalizacija.

Čebiševljevi polinomi prve vrste

Čebiševljevi polinomi prve vrste obično se označavaju s T_n. Oni su ortogonalni na intervalu $[-1, 1]$ obzirom na težinsku funkciju

$$w(x) = \frac{1}{\sqrt{1 - x^2}}.$$

Vrijedi

$$\int_{-1}^{1} \frac{T_m(x)\, T_n(x)}{\sqrt{1 - x^2}}\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi, & \text{za } m = n = 0, \\ \pi/2, & \text{za } m = n \neq 0. \end{cases}$$

Oni zadovoljavaju rekurzivnu relaciju

$$T_{n+1}(x) - 2xT_n(x) + T_{n-1}(x) = 0,$$

uz start

$$T_0(x) = 1, \quad T_1(x) = x.$$

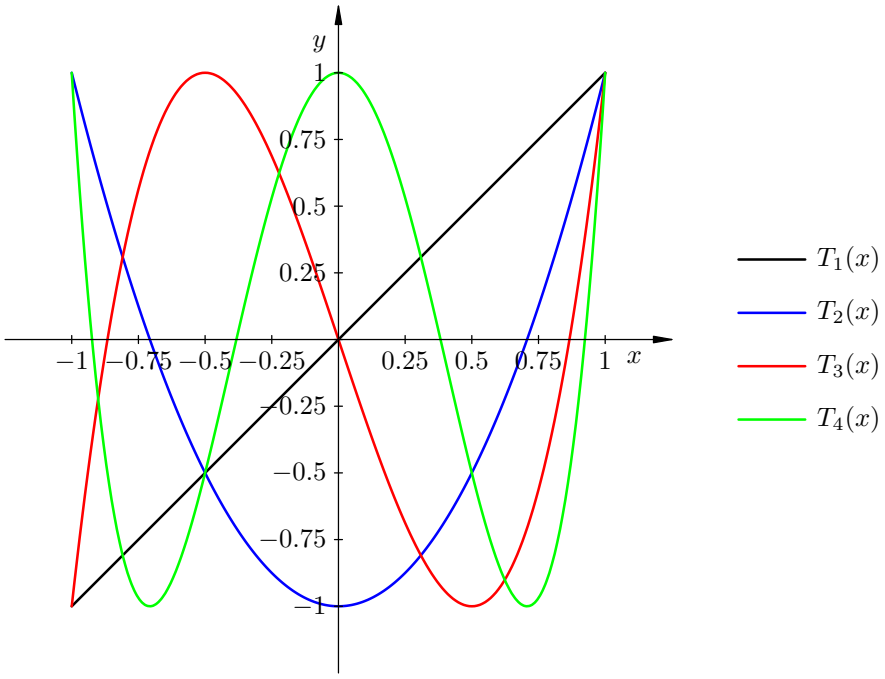
Za njih postoji i eksplicitna formula

$$T_n(x) = \cos(n \arccos x).$$

Osim toga, n-ti Čebiševljev polinom prve vrste T_n zadovoljava diferencijalnu jednadžbu

$$(1 - x^2)y'' - xy' + n^2 y = 0.$$

Graf prvih par polinoma izgleda ovako.



Katkad se koriste i Čebiševljevi polinomi prve vrste transformirani na interval $[0, 1]$, u oznaci T_n^*. Korištenjem linearne (preciznije, afine) transformacije

$$[0, 1] \ni x \mapsto \xi := 2x - 1 \in [-1, 1]$$

dolazimo do svih svojstava tih polinoma. Na primjer, relacija ortogonalnosti tada postaje

$$\int_0^1 \frac{T_m^*(x)\, T_n^*(x)}{\sqrt{x - x^2}}\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi, & \text{za } m = n = 0, \\ \pi/2, & \text{za } m = n \neq 0, \end{cases}$$

a rekurzivna relacija

$$T_{n+1}^*(x) - 2(2x - 1)T_n^*(x) + T_{n-1}^*(x) = 0,$$

uz start

$$T_0^*(x) = 1, \quad T_1^*(x) = 2x - 1.$$

Čebiševljevi polinomi druge vrste

Čebiševljevi polinomi druge vrste obično se označavaju s U_n. Oni su ortogonalni na intervalu $[-1, 1]$ obzirom na težinsku funkciju

$$w(x) = \sqrt{1 - x^2}.$$

Vrijedi

$$\int_{-1}^{1} \sqrt{1 - x^2}\, U_m(x)\, U_n(x)\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi/2, & \text{za } m = n. \end{cases}$$

Oni zadovoljavaju istu rekurzivnu relaciju kao Čebiševljevi polinomi prve vrste

$$U_{n+1}(x) - 2xU_n(x) + U_{n-1}(x) = 0,$$

samo uz malo drugačiji start

$$U_0(x) = 1, \quad U_1(x) = 2x.$$

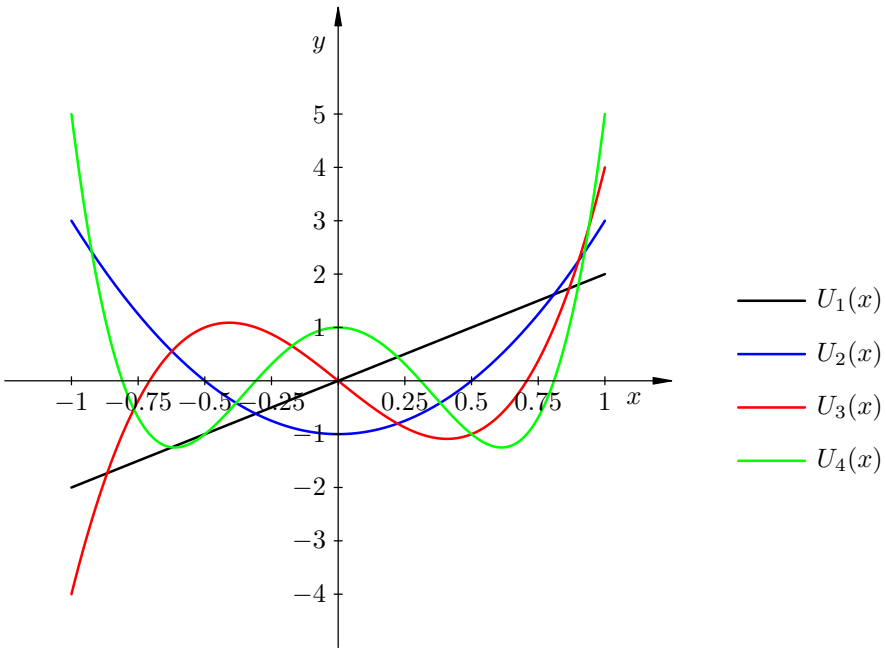
Za njih postoji i eksplicitna formula

$$U_n(x) = \frac{\sin((n+1)\arccos x)}{\sin(\arccos x)}.$$

Osim toga, n-ti Čebiševljev polinom druge vrste U_n zadovoljava diferencijalnu jednadžbu

$$(1 - x^2)y'' - 3xy' + n(n+2)y = 0.$$

Graf prvih par polinoma izgleda ovako.



Legendreovi polinomi

Legendreovi polinomi obično se označavaju s P_n. Oni su ortogonalni na intervalu $[-1, 1]$ obzirom na težinsku funkciju

$$w(x) = 1.$$

Vrijedi

$$\int_{-1}^{1} P_m(x)\, P_n(x)\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ 2/(2n+1), & \text{za } m = n. \end{cases}$$

Oni zadovoljavaju rekurzivnu relaciju

$$(n+1)P_{n+1}(x) - (2n+1)xP_n(x) + nP_{n-1}(x) = 0,$$

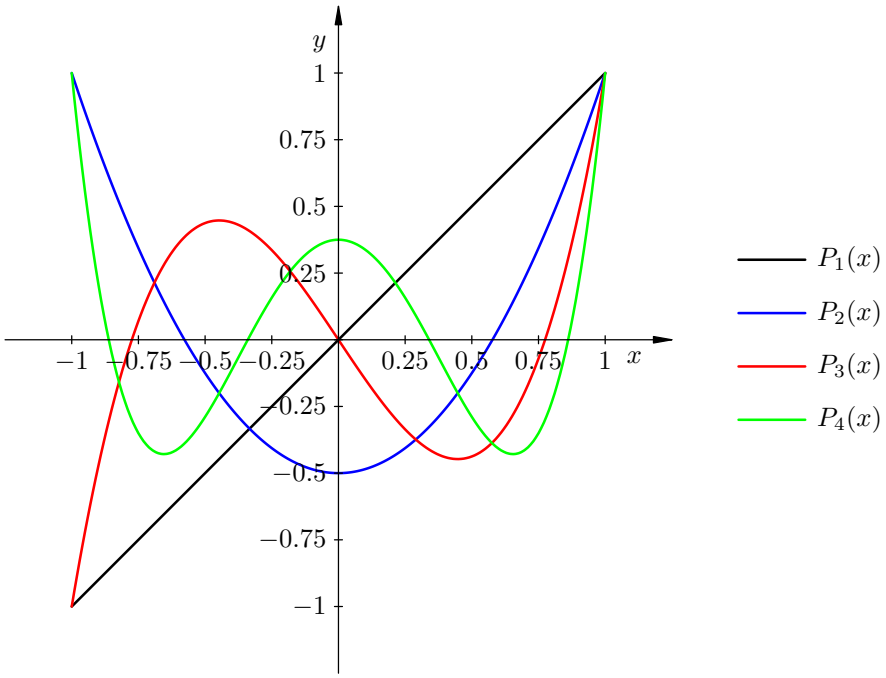
uz start

$$P_0(x) = 1, \quad P_1(x) = x.$$

Osim toga, n-ti Legendreov polinom P_n zadovoljava diferencijalnu jednadžbu

$$(1-x^2)y'' - 2xy' + n(n+1)y = 0.$$

Graf prvih par polinoma izgleda ovako.



Laguerreovi polinomi

Laguerreovi polinomi obično se označavaju s L_n. Oni su ortogonalni na intervalu $[0, \infty)$ obzirom na težinsku funkciju

$$w(x) = e^{-x}.$$

Vrijedi

$$\int_0^\infty e^{-x}\, L_m(x)\, L_n(x)\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ 1, & \text{za } m = n. \end{cases}$$

Oni zadovoljavaju rekurzivnu relaciju

$$(n+1)L_{n+1}(x) + (x - 2n - 1)L_n(x) + nL_{n-1}(x) = 0,$$

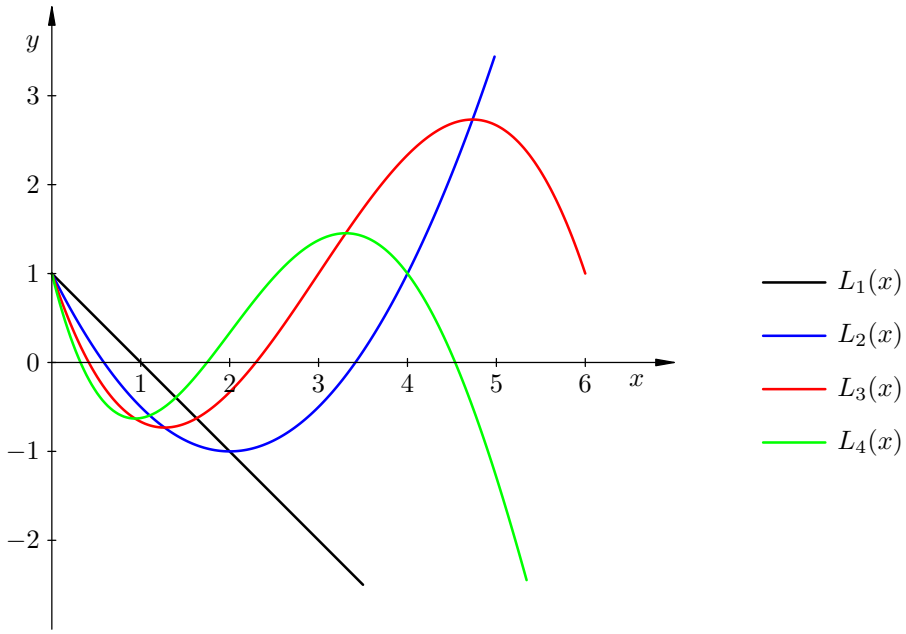
uz start

$$L_0(x) = 1, \quad L_1(x) = 1 - x.$$

Osim toga, n-ti Laguerreov polinom L_n zadovoljava diferencijalnu jednadžbu

$$xy'' + (1 - x)y' + ny = 0.$$

Graf prvih par polinoma izgleda ovako.



U literaturi se često nailazi na još jednu rekurziju za Laguerreove polinome

$$\widetilde{L}_{n+1}(x) + (x - 2n - 1)\widetilde{L}_n(x) + n^2\widetilde{L}_{n-1}(x) = 0,$$

uz jednaki start

$$\widetilde{L}_0(x) = 1, \quad \widetilde{L}_1(x) = 1 - x.$$

Uspoređivanjem ove i prethodne rekurzije dobivamo da je

$$\widetilde{L}_n(x) = n!\, L_n(x),$$

tj. radi se samo o drugačijoj normalizaciji ortogonalnih polinoma. Lako je pokazati da vrijedi

$$\int_0^\infty e^{-x}\, \widetilde{L}_m(x)\, \widetilde{L}_n(x)\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ (n!)^2, & \text{za } m = n. \end{cases}$$

Hermiteovi polinomi

Hermiteovi polinomi obično se označavaju s H_n. Oni su ortogonalni na intervalu $(-\infty, \infty)$ obzirom na težinsku funkciju

$$w(x) = e^{-x^2}.$$

Vrijedi

$$\int_{-\infty}^\infty e^{-x^2}\, H_m(x)\, H_n(x)\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ 2^n\, n!\sqrt{\pi}, & \text{za } m = n. \end{cases}$$

Oni zadovoljavaju rekurzivnu relaciju

$$H_{n+1}(x) - 2xH_n(x) + 2nH_{n-1}(x) = 0,$$

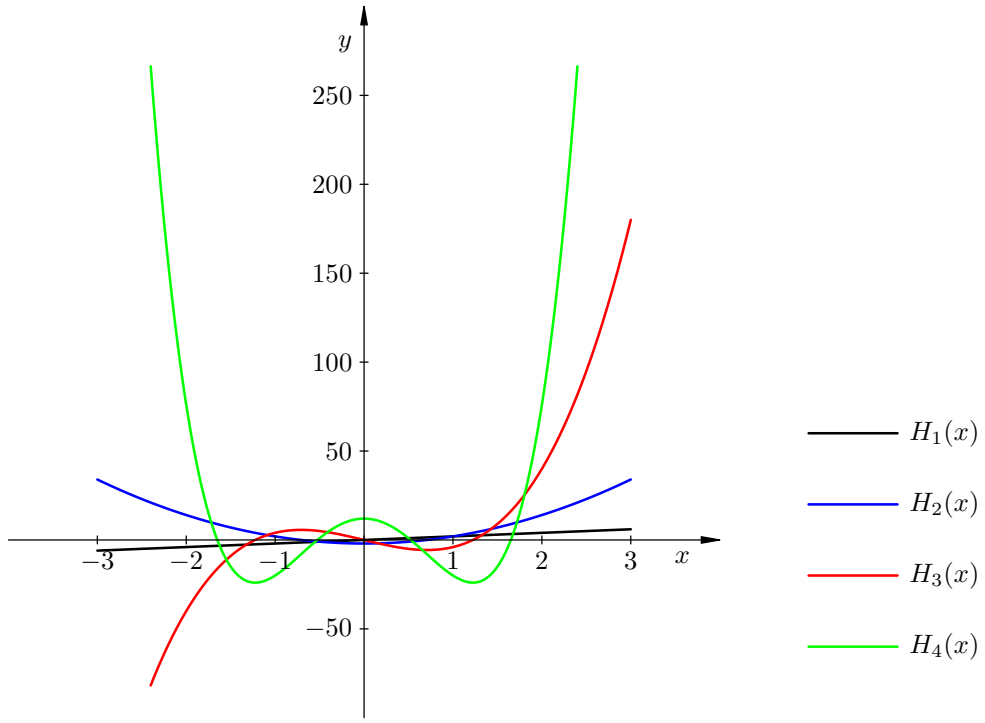
uz start

$$H_0(x) = 1, \quad H_1(x) = 2x.$$

Osim toga, n-ti Hermiteov polinom H_n zadovoljava diferencijalnu jednadžbu

$$y'' - 2xy' + 2ny = 0.$$

Graf prvih par polinoma izgleda ovako.



6.3. Stabilnost rekurzija i generalizirane Hornerove sheme

Stabilnost generalizirane Hornerove sheme u velikoj mjeri ovisi o stabilnosti rekurzije za funkcije p_n. Naime, ako u razvoju

$$f_N(x) = \sum_{n=0}^{N} a_n p_n(x),$$

uzmemo da je $a_N = 1$, i $a_n = 0$, za $n < N$, onda je $f_N = p_N$, pa točnost kojom je izračunat $p_N(x)$ može odrediti i točnost f_N. Dakle, generaliziranu Hornerovu shemu možemo koristiti i kao silazni algoritam za izvrednjavanje funkcija p_N. Postavlja se pitanje je li to bolje od direktnog računanja p_N unaprijed, po osnovnoj rekurziji

$$p_{n+1}(x) + \alpha_n(x)p_n(x) + \beta_n(x)p_{n-1}(x) = 0, \quad n = 1, 2, \ldots, N-1.$$

Za precizan odgovor, treba analizirati stabilnost ove rekurzije.

Prije toga, pogledajmo koliki je utjecaj te stabilnosti na generaliziranu Hornerovu shemu. Odgovor, naravno, bitno ovisi i o koeficijentima a_n u prikazu f_N. Za "lijepe" funkcije, ti koeficijenti obično relativno brzu teže prema nuli. Takvi mali

koeficijenti a_n, za veće n, bitno prigušuju greške greške u računanju $p_n(x)$. Isti efekt, samo manje vidljiv, postoji i u silaznom algoritmu.

U tom smislu, prethodni primjer je ekstreman, jer je zadnji koeficijent jednak 1, a svi prethodni su 0. Za taj primjer je f_N polinom. Razvoj polinoma (kao analitičke funkcije) po potencijama od varijable ili po nekom sustavu polinoma je konačan pa je uvjet smanjivanja koeficijenata a_n nepotreban. Iako to nije sasvim precizan argument, očito je ključno analizirati baš polinomni slučaj. Zbog toga, a i radi jednostavnosti, u nastavku gledamo samo polinomni slučaj $f_N = p_N$.

Općenito možemo odmah zaključiti da opasnost nastupa kad niz vrijednosti

$$p_0(x), p_1(x), \ldots, p_N(x)$$

naglo pada po apsolutnoj vrijednosti. Tada očekujemo jako kraćenja u osnovnoj rekurziji, što rezultira i gubitkom točnosti što dalje odmičemo u rekurziji. Dva su pitanja na koja bi bilo zgodno odgovoriti.

- Kako se tada ponaša silazni algoritam za računanje f_N?
- Može li se nekim trikom, poput okretanja rekurzije, popraviti stabilnost?

Umjesto općeg odgovora, koji bi koji zahtijeva dublju analizu, ilustrirajmo situaciju na jednom klasičnom primjeru.

Neka je $p_n(x) = e^{nx}$. Ove funkcije generiraju tzv. "eksponencijalne polinome" (jer umjesto potencija x^n imamo eksponencijalne funkcije e^{nx})

$$f_N(x) = \sum_{n=0}^{N} a_n e^{nx}.$$

Za takve p_n možemo sastaviti razne rekurzije. Dvočlana ima oblik

$$p_{n+1}(x) - e^x p_n(x) = 0, \quad n \in \mathbb{N}_0,$$

dok je tročlana homogenu rekurzija slična onimza za trigonometrijske funkcije,

$$p_{n+1}(x) - 2 \operatorname{ch} x \, p_n(x) + p_{n-1}(x) = 0, \quad n \in \mathbb{N},$$

pri čemu je $\operatorname{ch} x = (e^x + e^{-x})/2$ kosinus hiperbolni od x.

Očito je da $p_n(x)$ monotono raste za $x > 0$ i monotono pada za $x < 0$. Testirajmo stabilnost ove rekurzije i pripadne generalizirane Hornerove sheme za računanje $p_n(x) = e^{nx}$ u točkama $x = 1$ i $x = -1$.

6.4. Besselove funkcije i Millerov algoritam

Općenito nije potrebno znati funkcije p_0 i p_1 da bi se mogla koristiti silazna varijanta za računanje p_N. Dovoljno je znati neku vezu među funkcijama p_n koja se lako računa, a takve su često poznate. Na primjer, to su funkcije izvodnice oblika

$$F(x) = \sum_{n=0}^{\infty} q_n\, p_n(x),$$

gdje se $F(x)$ računa nekom analitičkom formulom bez upotrebe $p_n(x)$, tj. $F(x)$ možemo naći neovisno o funkcijama p_n.

Millerov algoritam (po J. C. P. Milleru, 1954. godine) primjenjuje se kada vrijednosti funkcija p_n vrlo brzo padaju kad n raste (za sve x ili u nekom području vrijednosti za argumente), a greška zaostaje.

Pretpostavimo da funkcije p_n zadovoljavaju neku homogenu rekurziju, na primjer tročlanu, koja je najčešća u praksi

$$p_{n+1}(x) + \alpha_n(x)p_n(x) + \beta_n(x)p_{n-1}(x) = 0, \quad n = 1, 2, \ldots$$

Poznavanje bilo kojeg p_n (čak ni p_0 niti p_1) nije potrebno. Treba znati samo koeficijente α_n i β_n.

6.4.1. Opća forma Millerovog algoritma

Prije no što ga primijenimo na eksponencijalnom polinomu, pokažimo kako funkcionira Millerov algoritam.

Odaberimo startnu vrijednost indeksa M od koje ćemo početi, ovisno o vrijednosti N indeksa funkcije koju tražimo. Ako tražimo $p_N(x)$ (ili $p_N(x), \ldots, p_0(x)$, ili samo neke od njih), M se obično odabere tako da je $M > N$ i vrijedi

$$\frac{p_M(x)}{p_N(x)} \approx \text{točnost računanja.}$$

To obično garantira i da je

$$F_M(x) := \sum_{n=0}^{M} q_n\, p_n(x),$$

barem jednako točna aproksimacija za $F(x)$, što ćemo kasnije iskoristiti.

Stavimo $\tilde{p}_{M+1} = 0$, $\tilde{p}_M = 1$ i računamo brojeve $\tilde{p}_n$, za $n = M-1, \ldots, 0$, unatrag po rekurziji za $p_n(x)$:

$$\tilde{p}_n = \frac{-(\alpha_{n+1}(x)\tilde{p}_{n+1} + \tilde{p}_{n+2})}{\beta_{n+1}(x)}, \quad n = M-1, \ldots, 0.$$

Zbog homogenosti rekurzije, dobiveni niz vrijednosti

$$\tilde{p}_M, \ldots, \tilde{p}_0$$

je vrlo približno proporcionalan stvarnim vrijednostima

$$p_M(x), \ldots, p_0(x),$$

barem u području od $p_N(x)$ do $p_0(x)$, tj. vrijedi $p_n(x) \approx \tilde{p}_n \cdot c$, za $n \leq N$. Treba još naći normalizacioni faktor c.

Sada iskoristimo činjenicu da znamo koeficijente q_n u razvoju funkcije izvodnice F po funkcijama p_n

$$F(x) = \sum_{n=0}^{\infty} q_n \, p_n(x).$$

Umjesto nepoznatih vrijednosti $p_n(x)$ uvrstimo $\tilde{p}_n$ i numeričkim zbrajanjem izračunamo aproksimaciju $\tilde{F}_M$

$$\widetilde{F}_M := \sum_{n=0}^{M} q_n \, \tilde{p}_n.$$

Gornji indeks sumacije može biti i bitno manji od M, ako znamo da $p_n(x)$ vrlo brzo padaju kad n raste. U prethodnoj sumi dovoljno je uzeti toliko članova da se izračunata vrijednost $\widetilde{F}_M$ stabilizira na točnost računala ili traženu točnost.

Zatim direktno analitički izračunamo $F(x)$ po poznatoj formuli i stavimo

$$c := \frac{F(x)}{\widetilde{F}_M},$$

što je traženi normalizacioni faktor, uz pretpostavku da je $F_M(x)$ dovoljno dobra aproksimacija za $F(x)$. Na kraju izračunamo

$$p_n(x) = \tilde{p}_n \cdot c$$

za sve one n između 0 i N koji nas zanimaju, jer u tom području vrijedi vrlo dobra proporcionalnost $p_n(x) \sim \tilde{p}_n$.

Vrlo često se startna vrijednost M određuje iz nekih poznatih relacija za familiju funkcija $p_n(x)$ ili eksperimentalno, povećavanjem n sve dok se ne postigne željena točnost za $p_N(x)$. Naravno, ovim se algoritmom može računati i $p_0(x)$.

6.4.2. Izvrednjavanje Besselovih funkcija

Besselove funkcije prvi puta je uveo Bessel, 1824. godine, promatrajući jedan problem iz tzv. dinamičke astronomije, vezan uz zgodan način zapisa položaja planeta koji se kreće po elipsi oko Sunca. Da bi dobio formulu prikladnu za praktično

računanje, Bessel je traženu veličinu prikazao kao red funkcija poznatih podataka. U tom redu se su se kao koeficijenti javile funkcije oblika

$$J_n(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin\theta - n\theta)\, d\theta, \quad n \in \mathbb{N}, \tag{6.4.1}$$

koje zovemo Besselovim funkcijama prve vrste. Očito se ova definicija može proširiti na $n \in \mathbb{Z}$ i tada je $J_{-n}(x) = (-1)^n J_n(x)$. Nažalost, ni za jednu od ovih funkcija ne postoji neka jednostavna formula ili oblik za računanje.

Relaciju (6.4.1) možemo iskoristiti i za numeričko računanje vrijednosti $J_n(x)$, za zadane $n \in \mathbb{N}_0$ i $x \in \mathbb{R}$, tako da upotrijebimo neku od metoda numeričke integracije. Međutim, postoje i mnogo brži algoritmi za postizanje iste tražene točnosti izračunate vrijednosti $J_n(x)$.

U klasičnom pristupu preko funkcije izvodnice, Besselove funkcije možemo definirati kao koeficijente uz t^n u razvoju

$$\exp\left(x\,\frac{t - 1/t}{2}\right) = \sum_{n=-\infty}^{\infty} J_n(x)\, t^n. \tag{6.4.2}$$

Nije teško dokazati da je ova definicija ekvivalentna integralnoj reprezentaciji (6.4.1). Iz (6.4.2) mogu se izvesti mnoge važne relacije za Besselove funkcije. Na primjer, Besselove funkcije zadovoljavaju tročlanu rekurziju

$$J_{n+1}(x) - \frac{2n}{x} J_n(x) + J_{n-1}(x) = 0, \quad n \in \mathbb{N}. \tag{6.4.3}$$

Također, vrijedi $J_1(x) = -J_0'(x)$. Dakle, kad bismo znali izračunati $J_0(x)$ i $J_1(x)$ (ili $J_0'(x)$), onda bismo iz (6.4.3) mogli izračunati i $J_n(x)$. Osim toga, generaliziranom Hornerovom shemom mogli bismo onda računati i razne razvoje po Besselovim funkcijama. Nažalost, to je tako samo u teoriji. Rekurzija (6.4.3) je izrazito nestabilna unaprijed. Da bismo to pokazali, promotrimo ponašanje vrijednosti Besselovih funkcija $J_n(x)$ u ovisnosti o n i x.

Iz (6.4.2) može se pokazati da Besselove funkcije J_n zadovoljavaju diferencijalnu jednadžbu

$$x^2 y'' + xy' + (x^2 - n^2)y = 0.$$

Ovu jednadžbu možemo promatrati na cijeloj kompleksnoj ravnini i u slučaju kad n nije cijeli broj. Tada se, umjesto n, obično koristi oznaka ν za parametar jednadžbe. Jedno od rješenja ove jednadžbe su Besselove funkcije prve vrste J_ν, koje imaju bitno svojstvo da su ograničene u 0 kad je $\operatorname{Re}\nu \geq 0$. Analitički im je oblik

$$J_\nu(x) = \left(\frac{1}{2}x\right)^\nu \sum_{k=0}^{\infty} \frac{(-x^2/4)^k}{k!\,\Gamma(\nu + k + 1)}, \tag{6.4.4}$$

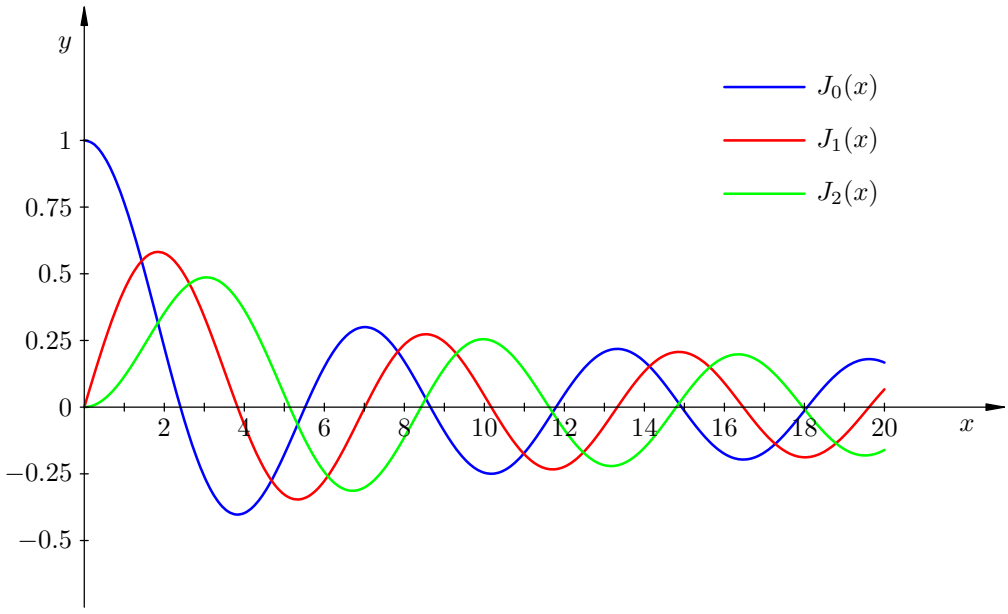
gdje su ν i x, općenito, kompleksni brojevi. U nastavku gledamo ponašanje ovih funkcija samo za nenegativne realne indekse $\nu \geq 0$ i argumente $x \geq 0$. Ako je ν cijeli broj, onda prethodna relacija glasi

$$J_n(x) = \left(\frac{1}{2}x\right)^n \sum_{k=0}^{\infty} \frac{(-x^2/4)^k}{k!\,(n+k)!}.$$

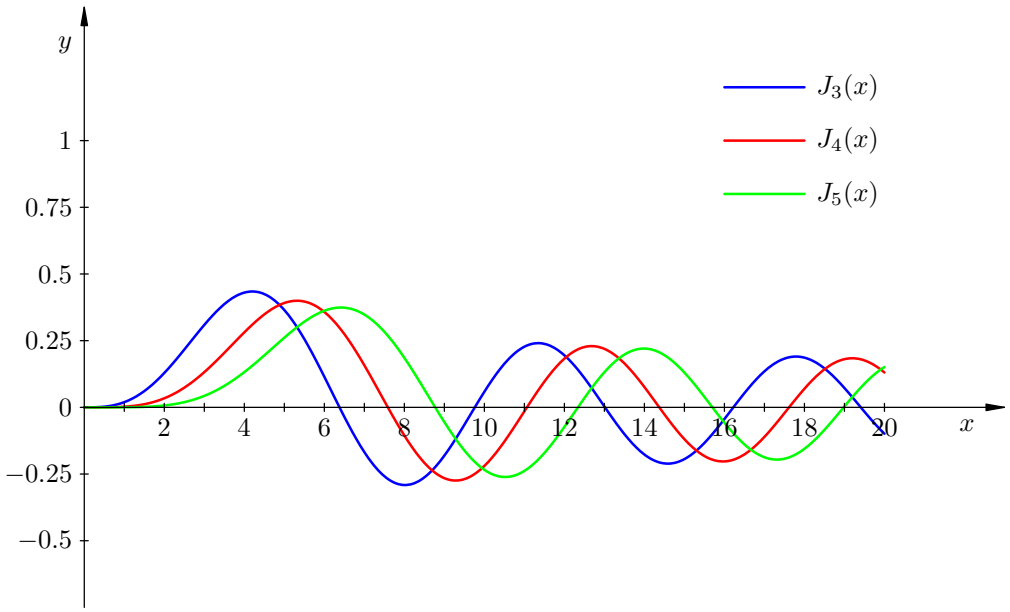
Oba reda očito konvergiraju za $x \geq 0$, a donji čak na cijelom skupu $\mathbb{C}$. Na prvi pogled izgleda kao da smo time riješili i problem računanja vrijednosti $J_0(x)$ i $J_1(x)$.

Zaista, ovaj red vrlo brzo konvergira za relativno male x, pa se može koristiti za računanje. Međutim, za malo veće x, kad je $x \approx n$ (ili ν) i dalje, dobivamo slično ponašanje kao i kod aproksimacije trigonometrijskih funkcija Taylorovim redom, tj. dolazi do sve većeg kraćenja zbrajanjem uzastopnih članova reda (6.4.4).

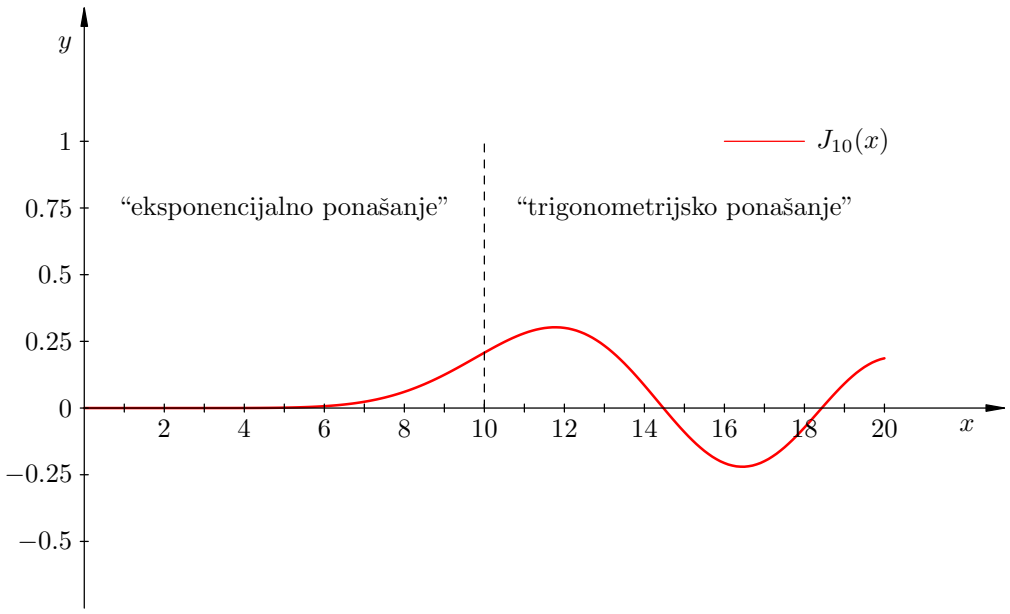
Grafički, prve tri Besselove funkcije izgledaju ovako



sljedeće tri ovako



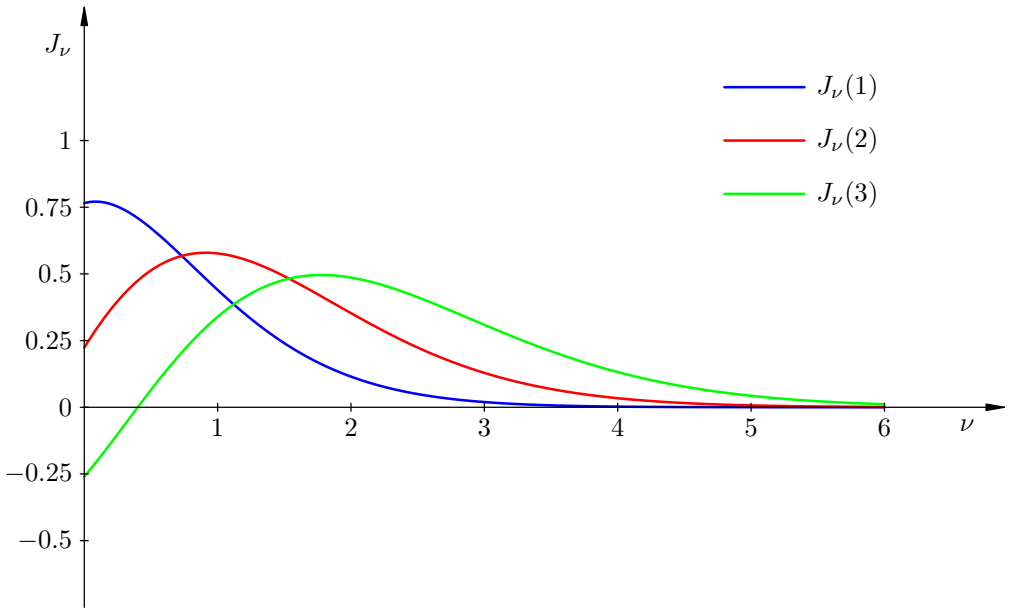
a, recimo, deseta Besselova funkcija ovako



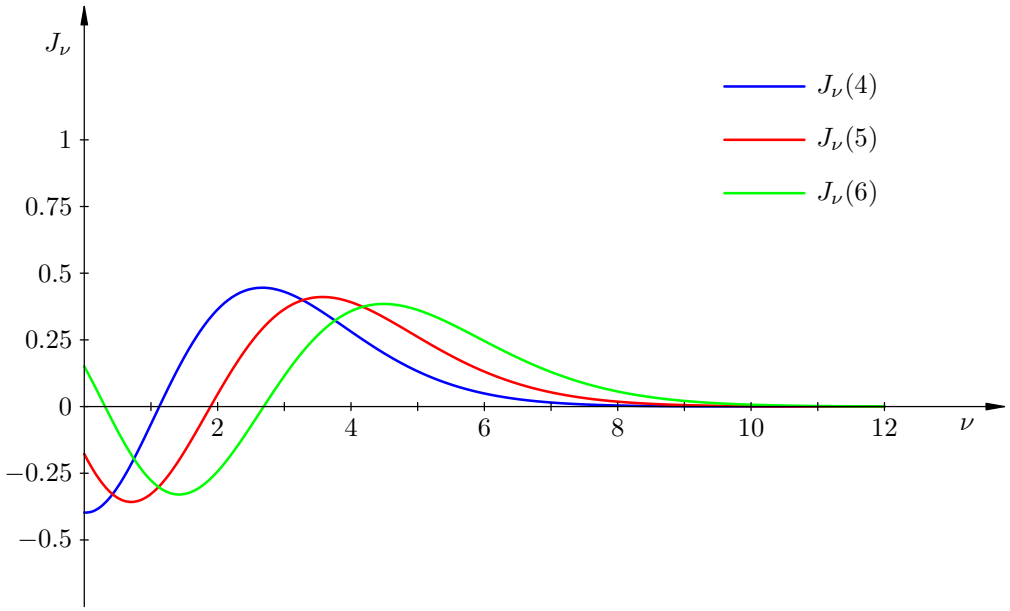
Uočite da se područje eksponencijalnog ponašanja mijenja u trigonometrijsko područje približno za $x = \nu$, kao što smo i očekivali iz Taylorovog reda.

Gledamo li Besselove funkcije ne kao funkcije od x, nego za fiksni x, kao funkcije

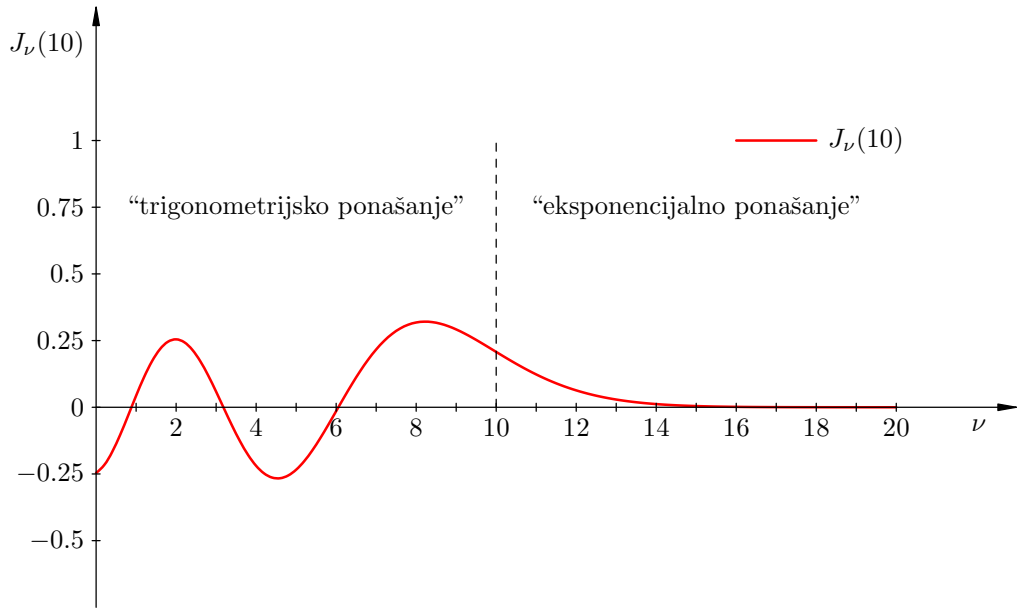
indeksa ν, onda Besselove funkcije pokazuju ovakvo ponašanje za $J_\nu(k)$, $k = 1, 2, 3$,



za $J_\nu(k)$, $k = 4, 5, 6$,



odnosno, $J_\nu(10)$ izgleda kao na sljedećoj slici:



Primijetite da i po ν postoji područje trigonometrijskog ponašanja koje za $x \approx \nu$ prelazi u eksponencijalni pad prema nuli.

Vidimo da kad n raste, u rekurziji (6.4.3) dobivamo sve manje i manje brojeve, što znači da mora doći do kraćenja. To pokazuje da je rekurzija nestabilna u rastućem smjeru po n, čim uđemo u ekponencijalno područje.

Za ilustraciju nestabilnosti možemo uzeti $x = 1$ i računati vrijednosti $J_n(x)$ koristeći rekurziju (6.4.3) uzlazno po n, u `extended` preciznosti. Dobiveni rezultati

na 18 decimala (apsolutno) dani su u sljedećoj tablici.

n	izračunati $J_n(1)$	točni $J_n(1)$
0	0.765197686557966552	0.765197686557966552
1	0.440050585744933516	0.440050585744933516
2	0.114903484931900481	0.114903484931900481
3	0.019563353982668406	0.019563353982668406
4	0.002476638964109955	0.002476638964109955
5	0.000249757730211237	0.000249757730211234
6	0.000020938338002418	0.000020938338002389
7	0.000001502325817779	0.000001502325817437
8	0.000000094223446486	0.000000094223441726
9	0.000000005249325991	0.000000005249250180
10	0.000000000264421352	0.000000000263061512
11	0.000000000039101058	0.000000000011980067
12	0.000000000595801917	0.000000000000499972

Kraćenje, a time i gubitak relativne točnosti počinje odmah za $n = 2$, ulaskom u eksponencijalno područje. Međutim, to se ne vidi u ovoj tablici, jer su rezultati prikazani apsolutno, a ne relativno. No, za $n = 11$ nemamo više niti jednu točnu znamenku, a za $n = 12$ gubimo i monotoni pad po n.

Vidimo da se događa nešto slično kao kod računanja e^{-nx}, što upućuje na okretanje rekurzije i primjenu Millerovog algoritma.

Korištenjem Millerovog algoritma dobivamo izuzetno dobre rezultate (drugi stupac tablice, koji je točan). Funkcija izvodnica koja se pritom koristi za normalizaciju je vrlo jednostavna

$$J_0(x) + 2(J_2(x) + J_4(x) + \cdots + J_{2k}(x) + \cdots) = 1.$$

Ova relacija izlazi direktno iz (6.4.2) za $t = 1$, kad iskoristimo parnost i neparnost Besselovih funkcija po n, tj. $J_{-n} = (-1)^n J_n$.

Za praktičnu primjenu Millerovog algoritma poželjno je znati precizno ponašanje rekurzije (6.4.3). Uočimo da je x fiksan, a zanima nas ponašanje $J_n(x)$ za velike n. Može se pokazati da u eksponencijalnom području vrijedi tzv. asimptotska relacija

$$J_\nu(x) \approx \frac{1}{\sqrt{2\pi\nu}} \left(\frac{ex}{2\nu}\right)^\nu, \tag{6.4.5}$$

za **fiksni** x i **velike** ν, tj. za $\nu \to \infty$. To pokazuje da se $J_n(x)$, gledano po n, za velike n ponaša kao

$$\frac{c_n}{n^{n+0.5}},$$

gdje je $c_n = (ex/2)^n/\sqrt{2\pi}$, a to vrlo brzo trne kad n raste. Uz malo pažnje, odavde se može izračunati početni indeks M za Millerov algoritam, tako da osiguramo potrebnu točnost.

Na sličan način može se opisati i ponašanje Besselovih funkcija J_ν kada je ν fiksan, a gledamo male ili velike argumente x. Za fiksni ν, kad $x \to 0$ iz prvog člana Taylorovog reda dobivamo i asimptotsku relaciju

$$J_\nu(x) \approx \left(\frac{1}{2}x\right)^\nu \frac{1}{\Gamma(\nu+1)},$$

koja je, očito, dobra aproksimacija za x u dijelu domene gdje se $J_\nu(x)$ ponaša kao eksponencijalna funkcija.

S druge strane, za $x \gtrsim n$, $J_n(x)$ se ponaša poput kosinusa, tj. oscilira. U tom trigonometrijskom području vrijedi

$$J_\nu(x) \approx \sqrt{\frac{2}{\pi x}} \cos\left(x - \frac{\pi}{4} - \frac{\nu\pi}{2}\right), \tag{6.4.6}$$

za fiksni ν, kad $x \to \infty$. Točno značenje relacija (6.4.5) i (6.4.6) bit će objašnjeno u sljedećem odjeljku o asimptotskim razvojima. Uobičajeno se koristi oznaka $\sim$, a ne $\approx$, za takve asimpotske relacije.

Napomena 6.4.1 *Napomenimo još da su sve potrebne vrijednosti $J_\nu(x)$ za crtanje prethodnih* 6 *slika izračunate sumacijom Taylorovog reda (6.4.4) unaprijed, sve dok zadnji dodani član ne padne ispod zadane točnosti. U prikazanom rasponu po ν i po x, kraćenje je minimalno (*2–3 *dekadske značajne znamenke). Jedini zanimljivi dio tog algoritma je računanje vrijednosti Γ funkcije, o čemu će uskoro biti više riječi.*

Naravno, za crtanje grafova nam i ne treba neka velika točnost funkcijskih vrijednosti. U principu, savršeno dovoljno je imati 3 *značajne znamenke u izračunatoj vrijednosti funkcije, tj. relativnu točnost reda veličine* 10^{-3}*. Za vrijednosti blizu nule, ne trebamo ni toliko. Dovoljna je relativna točnost istog reda veličine, ali obzirom na cijelu skalu, tj. raspon vrijednosti funkcije na cijelom grafu. Grešku od jedne tisućinke skale nitko neće ni primijetiti. Naime, ako je cijeli graf visok* 10 cm*, onda je ta greška na slici manja od desetinke milimetra!*

Drugo je pitanje u **koliko** *točaka treba izračunati vrijednost funkcije da bi se dobro nacrtao graf. Odgovor na to pitanje slijedi iz ocjena greške raznih vrsta aproksimacija, a posebno interpolacije, što ćemo napraviti u poglavlju o aproksimacijama. Zasad recimo samo to da je svaki od ovih grafova nacrtan korištenjem* 101 *točke, uz jednak razmak po x osi, a i to je bitno previše. Tridesetak točaka je sasvim dovoljno za vizuelno točan graf na ovim slikama.*

Besselove funkcije imaju vrlo velike primjene u fizici, u mnogim modelima, počev od difrakcije svjetlosti do energetskih nivoa u kvantnoj mehanici. Korištenjem

Millerovog algoritma, zajedno s Newtonovom metodom za nalaženje nultočaka funkcija, dobro se mogu izračunati i nultočke Besselovih funkcija, koje se, također, vrlo često koriste.

Za razliku od crtanja grafova Besselovih funkcija, za numeričko računanje njihovih nultočaka trebamo maksimalnu moguću točnost funkcijskih vrijednosti (barem u apsolutnom smislu), a to se postiže upravo Millerovim algoritmom.

6.5. Asimptotski razvoj

Sve aproksimacije koje smo do sada promatrali dobivene su "rezanjem" konvergentnih razvoja po nekom sustavu funkcija, tj. zamjenom konvergentnih redova konačnom sumom.

U relaciji (6.2.3) zamijenili smo razvoj funkcije f oblika

$$f(x) = \sum_{n=0}^{\infty} a_n p_n(x),$$

konačnom parcijalnom sumom (6.2.4)

$$f_N(x) = \sum_{n=0}^{N} a_n p_n(x),$$

podrazumijevajući da je riječ o **konvergentnom** redu u točki x. Tada za **fiksni** x, ostatak reda teži prema nuli po N,

$$\lim_{N\to\infty} (f(x) - f_N(x)) = \lim_{N\to\infty} \sum_{n=N+1}^{\infty} a_n p_n(x) = 0.$$

Strogo formalno, ako se sjetimo definicije sume reda, i sam zapis za $f(x)$ u obliku reda je sinonim za konvergenciju u ovom smislu. Eventualna uniformna konvergencija za sve x na nekoj domeni je dobrodošla, ali nije bitna za ideju ove aproksimacije.

U ovom odjeljku ćemo pokazati da zamjenom uloge N i x u iskazu o u konvergenciji razvoja dobivamo novi pojam **asimptotskog** razvoja, kojeg vrlo efikasno možemo iskoristiti i za praktično računanje. Ovaj pristup se najčešće koristi za računanje vrijednosti integrala, pa ga je zgodno uvesti baš na takvim primjerima.

Primjer 6.5.1 *Neka je f funkcija definirana integralom*

$$f(x) = \int_0^{\infty} e^{-xt} \cos t \, dt \tag{6.5.1}$$

za realne nenegativne vrijednosti parametra x. Pokušajmo ovaj intergal izračunati tako da $\cos t$ *razvijemo u red potencija po t, a zatim dobiveni red integriramo član po član. Dobivamo redom*

$$\begin{aligned} f(x) &= \int_0^\infty e^{-xt}\Big(1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \cdots\Big)\,dt \\ &= \int_0^\infty e^{-xt}\,dt - \int_0^\infty e^{-xt}\,\frac{t^2}{2!}\,dt + \int_0^\infty e^{-xt}\,\frac{t^4}{4!}\,dt - \cdots. \end{aligned}$$

Za integraciju član po član, treba izračunati integrale oblika

$$I_{2n}(x) = \int_0^\infty e^{-xt}\,\frac{t^{2n}}{(2n)!}\,dt,$$

za $n \in \mathbb{N}_0$. Za $n = 0$ odmah dobivamo

$$I_0(x) = \int_0^\infty e^{-xt}\,dt = -\frac{1}{x}\,e^{-xt}\Big|_{t=0}^\infty = \frac{1}{x},$$

jer ostaje samo vrijednost na donjoj granici, a na gornjoj granici znamo da za svaki pozitivni x, $e^{-xt} \to 0$ kad $t \to \infty$. Za $2n > 0$ parcijalnom integracijom izlazi

$$\begin{aligned} I_{2n}(x) &= \int_0^\infty \frac{t^{2n}}{(2n)!}\,d\Big(-\frac{1}{x}\,e^{-xt}\Big) \\ &= -\frac{1}{x}\,e^{-xt}\,\frac{t^{2n}}{(2n)!}\Big|_{t=0}^\infty + \frac{1}{x}\int_0^\infty e^{-xt}\,\frac{t^{2n-1}}{(2n-1)!}\,dt \\ &= \frac{1}{x}\,I_{2n-1}(x). \end{aligned}$$

Odavde indukcijom slijedi

$$I_{2n}(x) = \frac{1}{x^{2n+1}}.$$

Kad ove integrale uvrstimo natrag u red za f, dobivamo

$$f(x) = I_0(x) - I_2(x) + I_4(x) - \cdots = \frac{1}{x} - \frac{1}{x^3} + \frac{1}{x^5} - \cdots.$$

Na kraju, ako je $x > 1$, onda red na desnoj strani konvergira i vrijedi

$$f(x) = \frac{x}{x^2 + 1}. \tag{6.5.2}$$

Ovaj rezultat možemo dobiti i direktno iz (6.5.1) dvostrukom parcijalnom integracijom,

$$f(x) = \int_0^\infty e^{-xt}\cos t\,dt = \int_0^\infty e^{-xt}d(\sin t)$$
$$= e^{-xt}\,\sin t\Big|_{t=0}^{\infty} + \frac{1}{x}\int_0^\infty e^{-xt}\,\sin t\,dt.$$

Prvi član je nula na obje granice, pa ostaje

$$f(x) = \frac{1}{x}\int_0^\infty e^{-xt}d(-\cos t)$$
$$= -\frac{1}{x}\,e^{-xt}\,\cos t\Big|_{t=0}^{\infty} - \frac{1}{x^2}\int_0^\infty e^{-xt}\,\cos t\,dt$$
$$= \frac{1}{x} - \frac{1}{x^2}\,f(x).$$

Množenjem s x^2, za $x > 0$, dobivamo

$$x^2 f(x) = x - f(x),$$

pa zaključujemo da vrijedi (6.5.2), samo uz blažu pretpostavku $x > 0$.

Primjer 6.5.2 *Pokušajmo primijeniti istu ideju na funkciju g definiranu integralom*

$$g(x) = \int_0^\infty \frac{e^{-xt}}{1+t}\,dt, \quad x > 0. \tag{6.5.3}$$

Očekujemo još bolje rezultate, jer podintegralna funkcija još brže trne nego u prethodnom kad $t \to \infty$. Ovdje vrijedi $1/(1+t) \to 0$ kad $t \to \infty$, dok je $\cos t$ u funkciji f samo ograničen između -1 i 1. Supstitucijom razvoja

$$\frac{1}{1+t} = 1 - t + t^2 - \cdots$$

dobivamo redom

$$g(x) = \int_0^\infty e^{-xt}(1 - t + t^2 - \cdots)\,dt$$
$$= \int_0^\infty e^{-xt}\,dt - \int_0^\infty e^{-xt}t\,dt + \int_0^\infty e^{-xt}\,t^2\,dt - \cdots.$$

Za integraciju član po član, treba izračunati integrale oblika

$$\widehat{I}_n(x) = \int_0^\infty e^{-xt}\,t^n\,dt, \quad n \in \mathbb{N}_0.$$

Usporedbom s integralima $I_n(x)$ za funkciju f iz prethodnog primjera odmah vidimo da je

$$\widehat{I}_n(x) = n!\, I_n(x) = \frac{n!}{x^{n+1}}, \quad n \in \mathbb{N}_0.$$

Kad ove integrale uvrstimo natrag u red za g, dobivamo

$$g(x) = \widehat{I}_0(x) - \widehat{I}_1(x) + \widehat{I}_2(x) - \cdots = \frac{1}{x} - \frac{1!}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \cdots. \tag{6.5.4}$$

Međutim, ovaj red **divergira** *za sve konačne vrijednosti x i relacija (6.5.4) je besmislena. Dakle, funkciju g iz (6.5.3) ne možemo izračunati na ovaj način.*

Zašto je isti postupak bio uspješan u prvom primjeru, a ostao bez rezultata u drugom? Odgovor je jednostavan. Razvoj funkcije $\cos t$ konvergira za sve vrijednosti t, tj. na cijeloj domeni integracije, čak i jače, on konvergira uniformno na svakom konačnom intervalu za t. Zbog toga smijemo iskoristiti integraciju član po član na svakom konačnom intervalu, a zatim pustiti gornju granicu integracija na limes $t \to \infty$.

U drugom slučaju, razvoj funkcije $1/(1+t)$ konvergira samo za $t < 1$, a divergira za $t \geq 1$. Rezultat iz (6.5.4) treba shvatiti kao posljedicu integracije reda član po član, ali na intervalu na kojem taj red ne konvergira uniformno.

Sve dosad rečeno su standardne činjenice o redovima i integraciji iz matematičke analize. Međutim, ako cijelu stvar gledamo malo manje teorijski, a više praktično, onda nam nitko ne brani da pokušmo sumirati prvih nekoliko članova reda na desnoj strani u (6.5.4), za neku fiksnu vrijednost x. Pogledajmo što će se dogoditi, iako je to u potpunoj suprotnosti s poznatom teorijom!

Primjer 6.5.3 *Označimo s $g_n(x)$ parcijalne sume prvih n članova reda iz (6.5.4)*

$$g_n(x) = \frac{1}{x} - \frac{1!}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \cdots + (-1)^{n-1}\,\frac{(n-1)!}{x^n}. \tag{6.5.5}$$

Uzmimo neki malo veći x, recimo $x = 10$, tako da nazivnici relativno brzo padaju i izračunajmo prvih nekoliko vrijednosti $g_n(10)$. Za usporedbu, trebamo još i točnu vrijednost $g(10)$. Numeričkom integracijom može se izračunati da je

$$g(10) = 0.0915633339397880819\,.$$

Sljedeća tablica pokazuje izračunate vrijednosti $g_n(10)$ i pripadne pogreške.

n	*izračunati* $g_n(10)$	*greška* $g(10) - g_n(10)$
0	0.100000000000000000	−0.008436666060211918
1	0.090000000000000000	0.001563333939788082
2	0.092000000000000000	−0.000436666060211918
3	0.091400000000000000	0.000163333939788082
4	0.091640000000000000	−0.000076666060211918
5	0.091520000000000000	0.000043333939788082
6	0.091592000000000000	−0.000028666060211918
7	0.091541600000000000	0.000021733939788082
8	0.091581920000000000	−0.000018586060211918
9	0.091545632000000000	0.000017701939788082
10	0.091581920000000000	−0.000018586060211918
11	0.091542003200000000	0.000021330739788082
12	0.091589903360000000	−0.000026569420211918

Dobivene vrijednosti su sasvim dobre aproksimacije! Vidimo da je $g_9(10)$ najbolja aproksimacija, koja daje skoro 5 točnih decimala i skoro 4 točne vodeće znamenke. Naravno, za $n \geq 10$ pogreške sve više rastu i rezultati postaju beskorisni.

Sve u svemu, rezultat uopće nije loš, kad uzmemo da je nastao sumiranjem članova divergentnog reda. Nađimo još i objašnjenje za ovaj prilično neočekivani uspjeh.

Jasno je da treba promatrati grešku n-te parcijalne sume iz (6.5.5) u fiksnoj točki x. Neka je

$$e_n(x) := g(x) - g_n(x).$$

Ako još i razvoj funkcije $1/(1+t)$ napišemo u istom obliku kao zbroj prvih n članova plus ostatak,

$$\frac{1}{1+t} = 1 - t + t^2 - \cdots + (-1)^{n-1}t^{n-1} + \frac{(-1)^n t^n}{1+t},$$

i to uvrstimo u definiciju (6.5.3) za g, dobivamo da je greška $e_n(x)$ upravo integral ostatka iz prethodne relacije pomnoženog s e^{-xt},

$$e_n(x) = (-1)^n \int_0^\infty \frac{e^{-xt}t^n}{1+t}\,dt.$$

Uočimo da smo ovdje imali konačnu sumu, pa smo smjeli integrirati član po član. Ovu grešku nije teško ocijeniti. Očito je

$$\frac{1}{1+t} \leq 1, \quad t \geq 0,$$

pa je

$$|e_n(x)| = \int_0^\infty \frac{e^{-xt}t^n}{1+t}\,dt \leq \int_0^\infty e^{-xt}t^n\,dt = \frac{n!}{x^{n+1}}. \tag{6.5.6}$$

To pokazuje da je pogreška n-te parcijalne sume manja od apsolutne vrijednosti prvog odbačenog člana. Osim toga, zato što članovi alterniraju po predznaku, pogreška ima isti predznak kao i prvi odbačeni član. Dakle, članove reda možemo iskoristiti za ocjenu pogreške i zbrajanje članova treba zaustaviti točno **ispred** apsolutno najmanjeg člana.

Iz ocjene (6.5.6) za pogrešku n-te parcijalne sume odmah slijede dva zaključka. Ako gledamo konvergenciju u fiksnoj točki $x > 0$, onda je

$$|e_n(x)| \to \infty \quad \text{za} \quad n \to \infty,$$

pa nema govora o konvergenciji parcijalnih suma $g_n(x)$ prema $g(x)$. To odgovara ranijem zaključku da pripadni red divergira u svakoj točki $x > 0$.

S druge strane, ako uzmemo da je n fiksan, onda vrijedi

$$|e_n(x)| \to 0 \quad \text{za} \quad x \to \infty, \tag{6.5.7}$$

što odgovara "konvergenciji", ali sa zamijenjenim ulogama n i x. Takvu vrstu konvergencije zovemo **asimptotska konvergencija**, u ovom slučaju, u okolini točke $+\infty$.

Parcijalne sume g_n iz (6.5.5) generiraju red na desnoj strani (6.5.4), za kojeg kažemo da je **asimptotski razvoj** funkcije g u okolini točke $+\infty$, a relaciju (6.5.4) pišemo u obliku

$$g(x) \sim \frac{1}{x} - \frac{1!}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \cdots + (-1)^{n-1}\frac{(n-1)!}{x^n} + \cdots \quad (x \to +\infty). \tag{6.5.8}$$

Pritom $\sim$ označava asimptotsku konvergenciju reda na desnoj strani ove relacije, u smislu (6.5.7).

Prije precizne definicije pojma asimptotske konvergencije, objasnimo još vrijednost prethodnog zaključka. Relacija (6.5.7) vrijedi za svaki $n \in \mathbb{N}$, što znači da **svaku** od funkcija g_n možemo koristiti za aproksimaciju funkcije g, samo to moramo napraviti za dovoljno velike vrijednosti x, tj. u odgovarajućoj okolini točke $+\infty$. Takvom aproksimacijom **ne** možemo postići proizvoljno veliku točnost, odnosno

proizvoljno malu grešku, kao kod obične konvergencije. Ovisno o x, postoji minimalna greška koju možemo dobiti (gledano po n) i bolje ne ide. Međutim, i to se uspješno može iskoristiti za praktično računanje.

Osim toga, rezultat (6.5.7) ima i teorijsku vrijednost. Uzmimo da je $n = 1$ u (6.5.7). To znači da se, u prvoj aproksimaciji, funkcija g ponaša kao g_1 u okolini točke $+\infty$, tj. da $g(x)$ pada kao $1/x$, kad $x \to +\infty$. Iz relacije (6.5.6) dobivamo i ocjenu greške za takvu aproksimaciju,

$$g(x) - g_1(x) = g(x) - 1/x = e_1(x).$$

Iz (6.5.6) slijedi

$$|e_1(x)| \leq \int_0^\infty e^{-xt} t \, dt = \frac{1}{x^2},$$

pa odmah vidimo da vrijede sljedeće dvije relacije asimptotskog ponašanja. Prva je

$$g(x) - g_1(x) = O(1/x^2) \quad (x \to +\infty)$$

i ona služi kao osnova za definiciju asimptotskog razvoja. Druga je

$$g(x) - g_1(x) = o(g_1(x)) \quad (x \to +\infty),$$

ili, ekvivalentno (jer je $g_1(x) \neq 0$ za $x > 0$)

$$\lim_{x \to +\infty} \frac{g(x)}{g_1(x)} = 1.$$

Posljednju relaciju obično pišemo u obliku

$$g(x) \sim g_1(x) \quad (x \to +\infty) \tag{6.5.9}$$

i čitamo "$g(x)$ je asimptotski jednako $g_1(x)$ kad $x \to +\infty$", što znači da relativna greška od $g_1(x)$ kao aproksimacije od $g(x)$ teži prema nuli kad $x \to +\infty$.

Sve to slijedi samo iz prvog člana asimptotskog razvoja. Što znamo više članova, dobivamo sve preciznije informacije o ponašanju fukcije g u okolini $+\infty$.

Napomenimo da ista oznaka $\sim$ ima različita značenja u relacijama (6.5.8) i (6.5.9). U (6.5.9) $\sim$ je relacija asimptotskog ponašanja i ta relacija je simetrična, čak relacija ekvivalencije. Za razliku od toga, u (6.5.8) $\sim$ označava asimptotski razvoj i ta relacija nije simetrična. Lijevo je funkcija, a desno je red potencija (red funkcija). Uočimo da iz (6.5.8) slijede (6.5.9) i slični zaključci za aproksimacije g_n s više članova reda, ova oznaka $\sim$ se tradicionalno koristi u oba značenja. Naime, relaciju (6.5.9) možemo interpretirati i kao skraćeni zapis nekog asimptotskog razvoja, s tim da je naveden samo prvi član razvoja. Ali oprezno, tada zapis $g(x) \sim g_1(x)$, po definiciji, znači samo

$$\lim_{x \to +\infty} \frac{g(x)}{g_1(x)} = 1,$$

i nema govora o nekoj asimptotskoj konvergenciji koja se iskazuje relacijom (6.5.7), jer ne postoje ostali članovi razvoja (nema parametra n). Upravo tako treba interpretirati i asimptotske relacije za Besselove funkcije iz prethodnog odjeljka.

Precizna definicija asimptotskog razvoja u okolini neke točke bazirana je na definiciji asimptotskog niza u okolini te točke.

Definicija 6.5.1 (Asimptotski niz) *Neka je $D \subseteq \mathbb{R}$ neki skup i $c \in \operatorname{Cl} D$ neka točka iz zatvarača skupa D, s tim da c može biti i $+\infty$ ili $-\infty$. Nadalje, neka je $\varphi_n : D \to \mathbb{R}$, $n \in \mathbb{N}_0$, niz funkcija za kojeg vrijedi*

$$\varphi_n(x) = o(\varphi_{n-1}(x)) \quad (x \to c \text{ u } D),$$

za svaki $n \in \mathbb{N}$. Tada kažemo da je (φ_n) **asimptotski niz** *kad $x \to c$ u skupu D.*

Podsjetimo, oznaka oznaka $\varphi_n(x) = o(\varphi_{n-1}(x))$ $(x \to c \text{ u } D)$ znači da svaka funkcija φ_n raste bitno sporije od prethodne funkcije φ_{n-1} u okolini točke c, u smislu da vrijedi

$$\lim_{\substack{x \to c \\ x \in D}} \frac{\varphi_n(x)}{\varphi_{n-1}(x)} = 0,$$

što uključuje i pretpostavku da je $\varphi_{n-1}(x) \neq 0$ u nekoj okolini točke c gledano u skupu D, osim eventualno u samoj točki c. Zato, ako funkcije φ_n čine asimptotski niz, tada svaka funkcija φ_n raste bitno sporije od prethodne funkcije φ_{n-1} u okolini točke c. Ili, ako neka funkcija iz tog niza teži prema nuli kad $x \to c$ u skupu D, tada svaka sljedeća funkcija teži prema nuli bitno brže od nje.

Ista definicija vrijedi i za kompleksne domene $D \subseteq \mathbb{C}$, a točka c može biti i ∞.

Definicija 6.5.2 (Asimptotski razvoj) *Neka je $(\varphi_n,\ n \in \mathbb{N}_0)$, asimptotski niz kad $x \to c$ u skupu D. Formalni red funkcija*

$$\sum_{n=0}^{\infty} a_n \varphi_n$$

je **asimptotski razvoj** *funkcije f kad $x \to c$ u skupu D, u oznaci*

$$f(x) \sim \sum_{n=0}^{\infty} a_n \varphi_n(x) \quad (x \to c \text{ u } D), \tag{6.5.10}$$

ako za svaki $N \in \mathbb{N}$ vrijedi relacija asimptotskog ponašanja

$$f(x) = \sum_{n=0}^{N-1} a_n \varphi_n(x) + O(\varphi_N(x)) \quad (x \to c \text{ u } D),$$

tj. apsolutna greška između f i $(N-1)$-e parcijalne sume reda raste najviše jednako brzo kao i N-ti član asimptotskog niza, u okolini točke c.

Navedeni red funkcija treba interpretirati samo kao oznaku, tj. u čisto formalnom smislu, jer on može biti divergentan u svakoj točki domene.

Uočimo da iz prethodne dvije definicije odmah slijedi i

$$f(x) = \sum_{n=0}^{N-1} a_n \varphi_n(x) + o(\varphi_{N-1}(x)) \quad (x \to c \text{ u } D),$$

za svaki $N \in \mathbb{N}$. To znači da apsolutna greška između f i bilo koje parcijalne sume reda raste bitno sporije od zadnjeg člana u parcijalnoj sumi, u okolini točke c.

Tipični primjeri asimptotskih nizova su obične i logaritamske potencije

$$\varphi_n(x) = (x-c)^n \qquad \text{ili} \qquad \varphi_n(x) = (\log(x-c))^n,$$

za $n \in \mathbb{N}_0$, u okolini točke $c \in \mathbb{R}$ (ili $\mathbb{C}$). Pripadni asimptotski razvoji su obični redovi potencija

$$f(x) \sim \sum_{n=0}^{\infty} a_n (x-c)^n \quad (x \to c \text{ u } D),$$

ili logaritamskih potencija

$$f(x) \sim \sum_{n=0}^{\infty} a_n (\log(x-c))^n \quad (x \to c \text{ u } D).$$

U praksi se najčešće se koriste asimptotski nizovi u okolini točke ∞ oblika

$$\varphi_n(x) = x^{-n} \qquad \text{ili} \qquad \varphi_n(x) = (\log x)^{-n},$$

za $n \in \mathbb{N}_0$, koji nastaju supstitucijom $x \to 1/x$ iz nizova u okolini točke $c = 0$. Za potencije s negativnim eksponentima, asimptotski razvoj (6.5.10) ima već poznati oblik

$$f(x) \sim \sum_{n=0}^{\infty} \frac{a_n}{x^n} \quad (x \to \infty \text{ u } D),$$

što, po definiciji, znači da vrijedi

$$f(x) = \sum_{n=0}^{N-1} \frac{a_n}{x^n} + O(x^{-N}) \quad (x \to \infty \text{ u } D).$$

Tada nema smisla govoriti o rastu, već o padu funkcija u okolini točke ∞. Apsolutna greška između f i $(N-1)$-e parcijalne sume reda pada barem jednako brzo kao i x^{-N}, odnosno bitno brže od zadnjeg člana u sumi, koji je proporcionalan s $x^{-(N-1)}$. Povijesno gledano, H. Poincaré je 1886. godine definirao asimptotski razvoj baš u ovom obliku.

Napomenimo još da asimptotsko ponašanje može bitno ovisiti o domeni D, ne samo zbog područja definicije funkcija, već zbog moguće restrikcije točaka po kojima se gledaju limesi kad x teži prema c.

Ako je domena $D = \mathbb{R}$ ili $D = \mathbb{C}$, onda se koristi skraćena oznaka $(x \to c)$, bez navođenja domene D. Ako je još i $c = \infty$, onda se, tradicionalno, oznaka $(x \to \infty)$ može i ispustiti, tj. podrazumijeva se razvoj u okolini točke ∞.

Može se dogoditi da funkcija f nema asimptotski razvoj na zadanoj domeni D u smislu prethodne definicije. Ako je g poznata ili zadana funkcija takva da f/g ima asimptotski razvoj, onda se umjesto relacije (6.5.10) za f/g, često koristi i oznaka

$$f(x) \sim g(x) \cdot \sum_{n=0}^{\infty} a_n \varphi_n(x) \quad (x \to c \text{ u } D).$$

Ako je $a_0 \neq 0$, onda prvi član ovog razvoja daje i asimptotsko ponašanje funkcije f, tj. vrijedi $f(x) \sim a_0 g(x)$, kad $x \to c$ u D. U protivnom, ako je prvih k koeficijenata razvoja jednako nuli, tj. $a_0 = \ldots = a_{k-1} = 0$ i $a_k \neq 0$, onda vrijedi slična relacija (pokažite koja).

Analogno, ako $f - g$ ima asimptotski razvoj, gdje je g poznata funkcija, obično pišemo

$$f(x) \sim g(x) + \sum_{n=0}^{\infty} a_n \varphi_n(x) \quad (x \to c \text{ u } D).$$

6.6. Verižni razlomci i racionalne aproksimacije

Na početku ovog poglavlja zaključili smo da efikasno možemo računati samo racionalne aproksimacije funkcija. Sve dosadašnje aproksimacije imale su oblik **linearne kombinacije** nekih baznih funkcija. Kao bazne funkcije koristili smo potencije (aproksimacija polinomima) ili su potencije bile zamijenjene nekim drugim funkcijama, ali smo uvijek imali sume polinomnog oblika.

U tim aproksimacijama, dijeljenje nismo bitno iskoristili u obliku aproksimacije. Ako se sjetimo nekih rezultata iz analize, poput Weierstraßovog teorema o uniformnoj aproksimaciji funkcije polinomima na kompaktu, moglo bi nam se učiniti da uopće nema potrebe za drugim vrstama aproksimacija. S druge strane, prirodno je očekivati da "dodavanjem" operacije dijeljenja u oblik aproksimacijske funkcije, možemo postići znatno bolje aproksimacije za razne klase funkcija, korištenjem približno jednakog broja aritmetičkih operacija.

Pravu usporedbu polinomnih i racionalnih aproksimacija ostavljamo za poglavlje o aproksimacijama. Međutim, za praktičnu primjenu racionalnih aproksimacija bitni su dobri algoritmi za njihovo izvrednjavanje.

Pretpostavimo da je zadana racionalna funkcija oblika

$$R(x) = \frac{P_n(x)}{Q_m(x)},$$

gdje su P_n i Q_m polinomi stupnjeva n i m, respektivno,

$$P_n(x) = \sum_{k=0}^{n} a_k x^k, \quad Q_m(x) = \sum_{k=0}^{m} b_k x^k.$$

Očito je da izvrednjavanje prethodne funkcije možemo izvršiti korištenjem dviju Hornerovih shema (po jedna za polinom u brojniku i nazivniku) i jednim dijeljenjem na kraju. Broj potrebnih operacija je $n+m$ množenja, $n+m$ zbrajanja i jedno dijeljenje.

Ipak, takvo izvrednjavanje racionalne funkcije nije idealno iz više razloga. Prvo, jer postoje algoritmi koji zahtijevaju manje operacija. Nadalje, može se dogoditi da je vrijednost funkcije $R(x_0)$ neki broj razumnog reda veličine, ali je dobiven dijeljenjem dva vrlo velika broja (koja nisu prikaziva u računalu) ili dva vrlo mala broja (dobivena kraćenjem polinoma u brojniku ili nazivniku). Na neki način, tada bi trebalo vršiti neku normalizaciju u Hornerovoj shemi, čim nam kvocijent prijeđe neku zadanu veličinu, a tada algoritam postaje strašno kompliciran.

Na primjer, ako aproksimiramo vrijednost funkcije $\operatorname{th} x$ (slično je i za bilo koju drugu funkciju f koja ne teži u beskonačnost kada $x \to \infty$) u točki x_0, pri čemu je $x_0 \gg 1$, racionalna aproksimacija bit će omjer dva polinoma.

Ako želimo racionalnu aproksimaciju koja ima stupanj polinoma bar jedan u brojniku i nazivniku, onda će racionalna aproksimacija biti omjer vrijednosti polinoma brojnika i nazivnika "u dalekoj točki", a sama će vrijednost funkcije biti broj nekog razumnog reda veličine (u slučaju $\operatorname{th} x$ čak manji od 1).

U teoriji nepolinomnih aproksimacija moguće je pokazati da su, uz neke uvjete, najbolje racionalne aproksimacije one kojima je stupanj polinoma u brojniku jednak onom u nazivniku ili se eventualno razlikuje za jedan. U tom slučaju racionalnu aproksimaciju možemo napisati kao verižni razlomak, pa će i brzina računanja i problem preljeva (overflow) biti riješeni načinom izvrednjavanja takvog verižnog razlomka.

No prije no što upoznamo tzv. funkcijske verižne razlomke, upoznajmo se s brojevnim verižnim razlomcima.

6.6.1. Brojevni verižni razlomci

Izraz oblika

$$R = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{a_4}{b_4 + \cdots}}}}$$

zovemo (brojevni) verižni razlomak. Ovakav zapis je nespretan, pa su smišljeni alternativni zapisi. U različitoj literaturi nailazimo na tri oblika zapisa verižnih razlomaka

$$R = \left[b_0; \frac{a_1}{b_1}, \frac{a_2}{b_2}, \frac{a_3}{b_3}, \dots\right], \quad R = b_0 + \frac{a_1 \mid}{\mid b_1} + \frac{a_2 \mid}{\mid b_2} + \frac{a_3 \mid}{\mid b_3} + \cdots,$$

i možda najzgodniji zapis

$$R = b_0 + \frac{a_1}{{b_1}^+} \frac{a_2}{{b_2}^+} \frac{a_3}{{b_3}^+} \cdots. \tag{6.6.1}$$

Ako u beskonačnom verižnom razlomku uzmemo samo konačno mnogo članova,

$$R_n = b_0 + \frac{a_1}{{b_1}^+} \frac{a_2}{{b_2}^+} \frac{a_3}{{b_3}^+} \cdots \frac{a_n}{b_n}, \tag{6.6.2}$$

onda se takav izraz zove n-ta **konvergencija verižnog razlomka** R. Ako postoji vrijednost verižnog razlomka (6.6.1), onda se ona definira kao

$$R = \lim_{n\to\infty} R_n,$$

gdje je R_n n-ta konvergencija definirana izrazom (6.6.2).

Drugim riječima, važno je znati kako efikasno izračunati R_n.

6.6.2. Uzlazni algoritam za izvrednjavanje brojevnih verižnih razlomaka

Promatrajmo n-tu konvergenciju verižnog razlomka, koju možemo prikazati kao racionalni broj, kvocijent P_n i Q_n

$$R_n = \frac{P_n}{Q_n} = b_0 + \frac{a_1}{{b_1}^+} \frac{a_2}{{b_2}^+} \frac{a_3}{{b_3}^+} \cdots \frac{a_n}{b_n}.$$

Za nultu konvergenciju je

$$R_0 = \frac{P_0}{Q_0} = b_0,$$

pa možemo izabrati da je $P_0 = b_0$, $Q_0 = 1$ (mogli smo i drugačije birati, jedini je uvjet da je $P_0/Q_0 = b_0$). Za sljedeću konvergenciju vrijedi

$$R_1 = \frac{P_1}{Q_1} = b_0 + \frac{a_1}{b_1} = \frac{b_0 b_1 + a_1}{b_1} = \frac{b_1 P_0 + a_1}{b_1 Q_0}.$$

Ako još definiramo $P_{-1} = 1$, $Q_{-1} = 0$, onda prethodna relacija glasi

$$R_1 = \frac{P_1}{Q_1} = \frac{b_1 P_0 + a_1 P_{-1}}{b_1 Q_0 + a_1 Q_{-1}},$$

tj. ponovno možemo zatražiti da se brojnik određuje korištenjem prethodnih brojnika, a nazivnik korištenjem prethodnih nazivnika,

$$\begin{aligned} P_1 &= b_1 P_0 + a_1 P_{-1}, \\ Q_1 &= b_1 Q_0 + a_1 Q_{-1}. \end{aligned}$$

Te dvije relacije su baza indukcije za dokaz oblika rekurzije. Neka je $R_n = P_n/Q_n$, za P_n i Q_n vrijede relacije

$$\begin{aligned} P_n &= b_n P_{n-1} + a_n P_{n-2}, \\ Q_n &= b_n Q_{n-1} + a_n Q_{n-2}. \end{aligned}$$

Uočimo da R_{n+1} nastaje iz R_n ako b_n zamijenimo s $b_n + a_{n+1}/b_{n+1}$. No, u postupku računanja P_k i Q_k b_n se pojavljuje samo u računanju P_n i Q_n, a ne kod prethodnih $k < n$. Stoga, za R_{n+1} vrijedi

$$R_{n+1} = \frac{P'_{n+1}}{Q'_{n+1}},$$

gdje je

$$\begin{aligned} P'_{n+1} &= \left(b_n + \frac{a_{n+1}}{b_{n+1}}\right) P_{n-1} + a_n P_{n-2} \\ &= (b_n P_{n-1} + a_n P_{n-2}) + \frac{a_{n+1}}{b_{n+1}} P_{n-1} = P_n + \frac{a_{n+1}}{b_{n+1}} P_{n-1}, \\ Q'_{n+1} &= \left(b_n + \frac{a_{n+1}}{b_{n+1}}\right) Q_{n-1} + a_n Q_{n-2} \\ &= (b_n Q_{n-1} + a_n Q_{n-2}) + \frac{a_{n+1}}{b_{n+1}} Q_{n-1} = Q_n + \frac{a_{n+1}}{b_{n+1}} Q_{n-1}. \end{aligned}$$

Definiramo li

$$\begin{aligned} P_{n+1} &= b_{n+1} P'_{n+1}, \\ Q_{n+1} &= b_{n+1} Q'_{n+1}, \end{aligned}$$

onda R_{n+1} ostaje nepromijenjen (brojnik i nazivnik su skalirani), a prethodna rekurzija postaje

$$\begin{aligned} P_{n+1} &= b_{n+1} P_n + a_{n+1} P_{n-1}, \\ Q_{n+1} &= b_{n+1} Q_n + a_{n+1} Q_{n-1}. \end{aligned}$$

Time je pokazano da rekurzija za P_n i Q_n ostaje ista pri prelasku s konvergencije R_n na konvergenciju R_{n+1}, a to je upravo korak indukcije.

Drugim riječima, definiramo li

$$P_{-1} = 1, \quad Q_{-1} = 0, \quad P_0 = b_0, \quad Q_0 = 1, \tag{6.6.3}$$

onda, dobivamo tzv. uzlazni algoritam izvrednjavanja verižnog razlomka

$$\begin{aligned} P_k &= b_k P_{k-1} + a_k P_{k-2}, \\ Q_k &= b_k Q_{k-1} + a_k Q_{k-2}, \end{aligned} \qquad k = 1, 2, \ldots, n. \tag{6.6.4}$$

Primijetite da se u ovakvom zapisu algoritma lako mogu dodavati novi a_k i b_k, tzv. karike u verižnom razlomku.

Iz (6.6.4) lako se čita da su P_k i Q_k dva rješenja diferencijske jednadžbe

$$y_k - b_k y_{k-1} - a_k y_{k-2} = 0.$$

Uočite da bi nam u algoritmu (6.6.4) odgovaralo da su ili a_k ili b_k jednaki 1, tako da ne moramo množiti tim koeficijentima. To se može postići korištenjem tzv. ekvivalentne transformacije.

Neka su w_k, za $k \geq 1$, proizvoljni brojevi različiti od 0 i $w_{-1} = w_0 = 1$. Tvrdimo da izvrednjavanjem verižnog razlomka

$$R' = b_0 + \frac{w_0 w_1 a_1}{w_1 b_1{}^+} \frac{w_1 w_2 a_2}{w_2 b_2{}^+} \frac{w_2 w_3 a_3}{w_3 b_3{}^+} \cdots. \tag{6.6.5}$$

dobijemo isti R kao u (6.6.1). Označimo sa S_n i T_n brojnik i nazivnik n-te konvergencije prethodnog verižnog razlomka

$$R'_n = \frac{S_n}{T_n} = b_0 + \frac{w_0 w_1 a_1}{w_1 b_1{}^+} \frac{w_1 w_2 a_2}{w_2 b_2{}^+} \frac{w_2 w_3 a_3}{w_3 b_3{}^+} \cdots \frac{w_{n-1} w_n a_n}{w_n b_n}.$$

Pogledajmo u kojem su odnosu S_k i T_k obzirom na P_k i Q_k. Prvo napišimo rekurzije za S_k i T_k, jednostavno umetanjem "novih", proširenih a_k i b_k

$$\begin{aligned} S_k &= w_k b_k S_{k-1} + w_{k-1} w_k a_k S_{k-2}, \\ T_k &= w_k b_k T_{k-1} + w_{k-1} w_k a_k T_{k-2}, \end{aligned} \qquad k = 1, 2, \ldots, n,$$

uz

$$S_{-1} = 1, \quad T_{-1} = 0, \quad S_0 = b_0, \quad T_0 = 1.$$

Tvrdimo da postoji veza između P_k i S_k, te Q_k i T_k,

$$S_k = P_k \cdot \prod_{i=1}^{k} w_i, \quad T_k = Q_k \cdot \prod_{i=1}^{k} w_i, \quad k = 1, \ldots, n.$$

Dokaz se provodi indukcijom po k. Za bazu indukcije uzmimo $k = 1, 2$. Iz rekurzija dobivamo:

$$\begin{aligned} P_1 &= b_1 P_0 + a_1 P_{-1}, \\ P_2 &= b_2 P_1 + a_2 P_0, \\ S_1 &= w_1 b_1 S_0 + w_0 w_1 a_1 S_{-1} = w_1 b_1 P_0 + w_1 a_1 P_{-1} \\ &= w_1 (b_1 P_0 + a_1 P_{-1}) = w_1 P_1, \\ S_2 &= w_2 b_2 S_1 + w_1 w_2 a_2 S_0 = w_2 b_2 w_1 P_1 + w_1 w_2 a_2 P_0 \\ &= w_1 w_2 (b_2 P_1 + a_2 P_0) = w_1 w_2 P_2. \end{aligned}$$

Za korak indukcije, pretpostavimo da vrijedi

$$S_k = P_k \cdot \prod_{i=1}^{k} w_i, \quad S_{k-1} = P_{k-1} \cdot \prod_{i=1}^{k-1} w_i$$

za neke k, $k-1$. Tada vrijedi

$$\begin{aligned} S_{k+1} &= w_{k+1}b_{k+1}S_k + w_k w_{k+1} a_{k+1} S_{k-1} \\ &= w_{k+1}b_{k+1}P_k \cdot \prod_{i=1}^{k} w_i + w_k w_{k+1} a_{k+1} P_{k-1} \cdot \prod_{i=1}^{k-1} w_i \\ &= (b_{k+1}P_k + a_{k+1}P_{k-1}) \cdot \prod_{i=1}^{k+1} w_i = P_{k+1} \cdot \prod_{i=1}^{k+1} w_i \end{aligned}$$

što je i trebalo pokazati. Na sličan se način dokazuje i relacija za T_k i Q_k.

Dakle, vrijedi

$$R'_n = \frac{S_n}{T_n} = \frac{P_n \cdot \prod_{i=1}^{n+1} w_i}{Q_n \cdot \prod_{i=1}^{n+1} w_i} = \frac{P_n}{Q_n} = R_n.$$

Sada možemo pojednostavniti verižni razlomak R, tj. svesti ga na alternativnu formu, tako da u rekurziji (6.6.4) ili a_k ili b_k budu jednaki 1. Pretpostavimo da su $a_k \neq 0$ za sve $k \geq 1$. Budući da je izbor skalara w_k, $w_k \neq 0$, $k \geq 1$ proizvoljan w_k možemo izabrati tako da vrijedi

$$w_1 a_1 = 1, \quad w_1 w_2 a_2 = 1, \quad \ldots, \quad w_{n-1} w_n a_n = 1.$$

Odatle odmah slijedi

$$w_1 = \frac{1}{a_1}, \quad w_2 = \frac{1}{w_1 a_2} = \frac{a_1}{a_2}, \quad w_3 = \frac{1}{w_2 a_3} = \frac{a_2}{a_1 a_3}, \quad \ldots,$$

odnosno općenito

$$w_{2k} = \frac{a_1 a_3 \cdots a_{2k-1}}{a_2 a_4 \cdots a_{2k}}, \quad w_{2k+1} = \frac{a_2 a_4 \cdots a_{2k}}{a_1 a_3 \cdots a_{2k+1}},$$

što se dokazuje indukcijom. Time smo dobili tzv. II tip verižnog razlomka u kojem su brojnici jednaki 1,

$$R' = b_0 + \frac{1}{b'_1+} \frac{1}{b'_2+} \frac{1}{b'_3+} \cdots.$$

Pripadna uzlazna rekurzija za izvrednjavanje ima oblik

$$\begin{aligned} P_k &= b'_k P_{k-1} + P_{k-2}, \\ Q_k &= b'_k Q_{k-1} + Q_{k-2}, \end{aligned} \qquad k = 1, 2, \ldots, n,$$

uz startne podatke iz relacije (6.6.3).

S druge strane, možemo postići i da su nazivnici jednaki 1. Pretpostavimo da su $b_k \neq 0$ za sve $k \geq 1$. Budući da je izbor w_k, $w_k \neq 0$, $k \geq 1$ proizvoljan, izaberemo tako da vrijedi

$$w_1 b_1 = 1, \quad w_2 b_2 = 1, \quad \ldots, \quad w_n b_n = 1,$$

tj. stavljanjem

$$w_k = \frac{1}{b_k}$$

dobivamo tzv. I tip verižnog razlomka u kojem su nazivnici jednaki 1,

$$R' = b_0 + \frac{a'_1}{1^+} \frac{a'_2}{1^+} \frac{a'_3}{1^+} \cdots.$$

Koeficijenti a'_k su jednaki

$$a'_1 = \frac{a_1}{b_1}, \quad a'_2 = \frac{a_2}{b_1 b_2}, \quad a'_3 = \frac{a_3}{b_2 b_3},$$

odnosno općenito

$$a'_k = \frac{a_k}{b_{k-1} b_k}.$$

Pripadna rekurzija za uzlazno izvrednjavanje je

$$\begin{aligned} P_k &= P_{k-1} + a'_k P_{k-2}, \\ Q_k &= Q_{k-1} + a'_k Q_{k-2}, \end{aligned} \qquad k = 1, 2, \ldots, n,$$

uz start (6.6.3).

6.6.3. Eulerova forma verižnih razlomaka i neki teoremi konvergencije

Ako brojeve izaberemo tako da je zbroj brojnika i nazivnika jednak jedan (osim kod prve karike), tj. ako uzmemo

$$w_1 b_1 = 1, \quad w_{k-1} w_k a_k + w_k b_k = 1, \quad k = 2, 3, \ldots,$$

onda se verižni razlomak svede na tzv. Eulerovu formu

$$R' = b_0 + \frac{\alpha_1}{1^+} \frac{\alpha_2}{(1-\alpha_2)^+} \frac{\alpha_3}{(1-\alpha_3)^+} \cdots.$$

Da bismo uspostavili vezu početnog verižnog razlomka i dobivenog verižnog razlomka u Eulerovoj formi, napišimo rekurzije za verižni razlomak u Eulerovoj formi korištenjem veličina S_k i T_k. Promatrajmo drugu rekurziju iz (6.6.4) za Q_k odnosno

T_k, (jer nas zanima ta rekurzija za verižni razlomak u Eulerovoj formi). Uz $T_{-1} = 0$, $T_0 = 1$, dobivamo

$$T_1 = T_0 + \alpha_1 T_{-1}, \quad T_k = (1 - \alpha_k)T_{k-1} + \alpha_k T_{k-2}, \quad k \geq 2.$$

Uočite da je $T_1 = T_0 = 1$, pa vrijedi

$$T_k = 1, \quad k \geq 0.$$

Time se od dvije rekurzije za računanje n-te konvergencije verižnog razlomka koristi samo ona prva, tj. vrijedi $R_k = S_k$ za sve k. Iz $R_{-1} = S_{-1} = 1$, $R_0 = S_0 = b_0$ izlazi

$$R_1 = R_0 + \alpha_1 R_{-1}, \quad R_k = (1 - \alpha_k)R_{k-1} + \alpha_k R_{k-2}, \quad k \geq 2.$$

Ne mogu se svi verižni razlomci mogu svesti na Eulerov oblik. Verižni razlomak može svesti na Eulerovu formu ako i samo ako su svi $Q_k \neq 0$ (u rekurziji za početni verižni razlomak).

Nađimo vezu između originalnog i Eulerovog verižnog razlomka. Dokaz te tvrdenje ponovno koristi indukciju. Prvo pokažimo da za sve w_k vrijedi

$$w_k = \frac{Q_{k-1}}{Q_k}, \quad k \geq 1. \tag{6.6.6}$$

Iz $w_1 b_1 = 1$, uz pretpostavku da je $b_1 \neq 0$ slijedi

$$w_1 = \frac{1}{b_1} = \frac{Q_0}{Q_1},$$

što je baza indukcije. Pretpostavimo da za za neki w_n vrijedi

$$w_n = \frac{Q_{n-1}}{Q_n}.$$

Iz definicijske relacije za Eulerov verižni razlomak slijedi da je

$$w_n w_{n+1} a_{n+1} + w_{n+1} b_{n+1} = 1,$$

pa primjenom pretpostavke indukcije dobivamo

$$w_{n+1} = \frac{1}{w_n a_{n+1} + b_{n+1}} = \frac{1}{\frac{Q_{n-1}}{Q_n} a_{n+1} + b_{n+1}} = \frac{Q_n}{Q_{n-1} a_{n+1} + Q_n b_{n+1}} = \frac{Q_n}{Q_{n+1}},$$

čime je dokazan korak indukcije.

Iz relacije (6.6.6) i definicije

$$\alpha_k = w_{k-1} w_k a_k$$

uz dogovor $w_0 = 1$, odmah slijedi da je

$$\alpha_k = \frac{Q_{k-2}}{Q_k}\, a_k, \quad k \geq 2. \tag{6.6.7}$$

Eulerovu formu verižnog razlomka, uglavnom ćemo koristiti pri dokazivanju tvrdnji. Da bismo dokazali konvergenciju verižnog razlomka, potraban nam je jednostavan izraz za R_k, po mogućnosti neki red.

Prvo, iz rekurzija za P_k i Q_k nađimo koliko je $R_k - R_{k-1}$, a zatim i $R_k - R_{k-2}$. Za $k \geq 1$ vrijedi

$$\begin{aligned} R_k - R_{k-1} &= \frac{P_k}{Q_k} - \frac{P_{k-1}}{Q_{k-1}} = \frac{P_k Q_{k-1} - P_{k-1} Q_k}{Q_k Q_{k-1}} \\ &= \frac{(b_k P_{k-1} + a_k P_{k-2}) Q_{k-1} - P_{k-1}(b_k Q_{k-1} + a_k Q_{k-2})}{Q_k Q_{k-1}} \\ &= \frac{-a_k (P_{k-1} Q_{k-2} - P_{k-2} Q_{k-1})}{Q_k Q_{k-1}} \\ &= \frac{a_k a_{k-1} (P_{k-2} Q_{k-3} - P_{k-3} Q_{k-2})}{Q_k Q_{k-1}} = \cdots \\ &= (-1)^{k+1} \frac{a_k a_{k-1} \cdots a_1}{Q_k Q_{k-1}}. \end{aligned} \tag{6.6.8}$$

Na sličan se način dokazuje da je

$$R_k - R_{k-2} = (-1)^k \frac{b_k a_{k-1} \cdots a_1}{Q_k Q_{k-2}}, \quad k \geq 2. \tag{6.6.9}$$

Korištenjem relacije (6.6.6) možemo (6.6.8) zapisati u terminima α_k (ponovno uz pretpostavku da su svi Q_i različiti od 0). Vrijedi

$$R_k - R_{k-1} = (-1)^{k+1} \frac{a_k a_{k-1} \cdots a_1}{Q_k Q_{k-1}} = (-1)^{k+1} \alpha_k \alpha_{k-1} \cdots \alpha_1. \tag{6.6.10}$$

Rekurzivnom primjenom (6.6.10) dobivamo da je

$$R_k = (-1)^{k+1} \alpha_k \alpha_{k-1} \cdots \alpha_1 + R_{k-1} = \cdots = b_0 + \sum_{i=1}^{k} (-1)^{i+1} \alpha_i \alpha_{i-1} \cdots \alpha_1.$$

Ako verižni razlomak konvergira, onda je

$$R = \lim_{k \to \infty} R_k = b_0 + \sum_{i=1}^{\infty} (-1)^{i+1} \alpha_i \alpha_{i-1} \cdots \alpha_1, \tag{6.6.11}$$

pa je

$$|R_k - R| = \Big| \sum_{i=k+1}^{\infty} (-1)^{i+1} \alpha_i \alpha_{i-1} \cdots \alpha_1 \Big|.$$

Sada možemo izreći i dokazati neke rezultate o konvergenciji verižnih razlomaka.

Teorem 6.6.1 *Ako su $a_k, b_k > 0$, tada vrijede nejednakosti*

$$R_1 > R_3 > \cdots > R_{2k-1} > \cdots,$$
$$R_0 < R_2 < \cdots < R_{2k} < \cdots$$

i

$$R_{2m-1} > R_{2k}$$

za svako m i k.

Dokaz. Ako su $a_k, b_k > 0$, onda su to i Q_k (po rekurziji). Nakon toga, dokaz trivijalno slijedi raspisivanjem relacije (6.6.9) za parne i neparne indekse. ■

Teorem 6.6.2 *Ako su $a_k, b_k > 0$ i vrijedi $a_k \leq b_k$ i $b_k \geq \varepsilon > 0$ za $k \geq 1$, gdje je ε neka konstanta, onda je verižni razlomak (6.6.1) konvergentan.*

Dokaz. Budući da su $a_k, b_k > 0$, onda su to i Q_k, pa iz (6.6.7) izlazi

$$\alpha_1\alpha_2\cdots\alpha_i = \frac{a_1a_2\cdots a_i}{Q_iQ_{i-1}} > 0,$$

pa je red u (6.6.11) alternirajući. Po Leibnitzovom kriteriju dovoljno je pokazati da n-ti član tog reda teži u 0 i da mu članovi opadaju po apsolutnoj vrijednosti.

Da bismo pokazali konvergenciju, dovoljno je pokazati da je $\alpha_i \leq q$ za neko $q < 1$ (D'Alembertov kriterij konvergencije). Iz (6.6.7) dobivamo

$$\alpha_i = a_i\frac{Q_{i-2}}{Q_i} = \frac{a_iQ_{i-2}}{b_iQ_{i-1} + a_iQ_{i-2}} < \frac{a_iQ_{i-2}}{a_iQ_{i-1} + a_iQ_{i-2}} = \frac{Q_{i-2}}{Q_{i-1} + Q_{i-2}}.$$

U prethodnoj jednakosti potrebno je samo pokazati da postoji donja ograda za Q_{i-1}, što slijedi iz uvjeta $b_k \geq \varepsilon > 0$ i rekurzije za Q_i. Lako se dokazuje da je $Q_i \geq \varepsilon Q_{i-1}$, pa odatle slijedi da je

$$\alpha_i < \frac{Q_{i-2}}{(1+\varepsilon)Q_{i-2}} = \frac{1}{1+\varepsilon} < 1.$$

Također, red je alternirajući, pa po Leibnitzovom kriteriju, greška koju smo napravili ako R aproksimiramo s R_k manja je ili jednaka prvom odbačenom članu u redu, tj. vrijedi

$$|R_k - R| \leq \frac{a_1a_2\cdots a_{k+1}}{Q_kQ_{k+1}}.$$

■

6.6.4. Silazni algoritam za izvrednjavanje brojevnih verižnih razlomaka

Vratimo se još jednom na izvrednjavanje verižnih razlomaka. Prethodni je teorem dao neke ocjene o tome koliko dobro R_n aproksimira R. Zbog toga, možemo pretpostaviti da nam je unaprijed poznato koliko konvergencija trebamo da bismo dobro aproksimirali neki verižni razlomak.

Krenemo li od "silazno" od b_n, onda će sljedeća rekurzija izvrednjavati R_n. Definiramo $F_n = b_n$ (ili $F_{n+1} = \infty$) i računamo

$$F_k = b_k + \frac{a_{k+1}}{F_{k+1}}, \quad k = (n), n-1, \ldots, 0,$$

na kraju ćemo dobiti

$$R_n = F_0.$$

Primijetite da silazna rekurzija u svakom koraku ima točno jedno zbrajanje i jedno dijeljenje, za razliku od uzlazne, koja u svakom koraku (u općem slučaju) ima 4 množenja i 2 zbrajanja.

Ova dva tipa rekurzija analogon su izvrednjavanja polinoma: silazna rekurzija analogon je Hornerove sheme, a uzlazna je analogon potenciranja i zbrajanja.

Dapače, može se pokazati da je silazna rekurzija za izvrednjavanje verižnih razlomaka optimalna što se tiče broja operacija, čak i ako su dozvoljene transformacije koeficijenata verižnog razlomka prije početka izvrednjavanja.

6.6.5. Funkcijski verižni razlomci

Funkcijski verižni razlomci mogu se dobiti na više načina, i mogu imati više oblika. Verižne razlomke koji imaju varijablu u brojniku zovemo verižni razlomci tipa I i oni imaju opći oblik

$$f(x) = \beta_0 + \frac{x - x_1}{{\beta_1}^+} \frac{x - x_2}{{\beta_2}^+} \frac{x - x_3}{{\beta_3}^+} \cdots. \tag{6.6.12}$$

Funkcijski verižni razlomci mogu imati varijablu u nazivniku, i takve verižne razlomke zovemo verižni razlomci tipa II. Općenito, oni imaju oblik

$$f(x) = b_0 + \frac{a_1}{(x + b_1)^+} \frac{a_2}{(x + b_2)^+} \frac{a_3}{(x + b_3)^+} \cdots. \tag{6.6.13}$$

Izvrednjavanje verižnih razlomaka oba tipa vrlo je slično izvrednjavanju brojevnih verižnih razlomaka. Za izvrednjavanje n-te konvergencije $f_n(x)$ verižnih

razlomaka prvog tipa, možemo koristiti silazni algoritam. Stavimo $F_n = \beta_n$ (ili $F_{n+1} = \infty$), a zatim računamo

$$F_k = \beta_k + \frac{x - x_{k+1}}{F_{k+1}}, \quad k = (n), n-1, \ldots, 0,$$

i na kraju je

$$f_n(x) = F_0.$$

Za izvrednjavanje n-te konvergencije verižnih razlomaka drugog tipa možemo koristiti, također, silazni algoritam. Stavimo $F_n = b_n$ (ili $F_{n+1} = \infty$), a zatim računamo

$$F_k = b_k + \frac{a_{k+1}}{x + F_{k+1}}, \quad k = (n), n-1, \ldots, 0,$$

i na kraju je

$$f_n(x) = F_0.$$

Kako dolazimo do verižnih razlomaka? Obično je nešto lakše doći do verižnih razlomaka tipa I, a zatim ih možemo pretvoriti u tip II. Ako imamo zadanu funkciju f, uobičajeno se verižni razlomak nalazi nestandardiziranim postupkom kad se funkcija zapisuje "pomoću same sebe". Da bismo to bolje objasnili, pogledajmo to na primjeru jedne funkcije.

Primjer 6.6.1 *Razvijmo u verižni razlomak prvog tipa funkciju*

$$f(x) = \sqrt{1+x}.$$

Prvo, potrebno funkciju malo drugačije zapisati. Lako se provjerava da je

$$\sqrt{1+x} = 1 + \frac{x}{1 + \sqrt{1+x}}.$$

Ako ponovimo ovaj raspis u nazivniku razlomka, dobivamo verižni razlomak

$$\sqrt{1+x} = 1 + \frac{x}{2^+} \, \frac{x}{2^+} \, \frac{x}{2^+} \, \cdots.$$

Navedimo neke od poznatih verižnih razlomaka, bez njihova "izvoda":

$$\begin{aligned} e^x &= \frac{1}{1^-}\,\frac{x}{1^+}\,\frac{x}{2^-}\,\frac{x}{3^+}\,\frac{x}{2^-}\,\frac{x}{5^+}\,\frac{x}{2^-}\,\frac{x}{7^+}\,\cdots. \\ &= 1+\frac{x}{1^-}\,\frac{x}{2^+}\,\frac{x}{3^-}\,\frac{x}{2^+}\,\frac{x}{5^-}\,\frac{x}{2^+}\,\frac{x}{7^-}\,\cdots, \end{aligned}$$

$$\ln(x+1) = \frac{x}{1^+}\,\frac{x}{2^+}\,\frac{x}{3^+}\,\frac{4x}{4^+}\,\frac{4x}{5^+}\,\frac{9x}{6^+}\,\frac{9x}{7^+}\,\frac{16x}{8^+}\,\frac{16x}{9^+}\,\cdots,$$

$$x\operatorname{tg} x = \frac{x^2}{1^-}\,\frac{x^2}{3^-}\,\frac{x^2}{5^-}\,\frac{x^2}{7^-}\,\cdots, \quad x \neq \frac{(2n+1)\pi}{2},$$

$$x\operatorname{arctg} x = \frac{x^2}{1^+}\,\frac{x^2}{3^+}\,\frac{4x^2}{5^+}\,\frac{9x^2}{7^+}\,\frac{16x^2}{9^+}\,\cdots,$$

$$x\operatorname{th} x = \frac{x^2}{1^+}\,\frac{x^2}{3^+}\,\frac{x^2}{5^+}\,\frac{x^2}{7^+}\,\cdots$$

$$x\operatorname{Arth} x = \frac{x^2}{1^-}\,\frac{x^2}{3^-}\,\frac{4x^2}{5^-}\,\frac{9x^2}{7^-}\,\frac{16x^2}{9^-}\,\cdots.$$

Svi ovi verižni razlomci su prvog tipa. Ima li koristi znati kako bi izgledao njihov drugi tip? Na primjer šesta konvergencija verižnog razlomka za $\sqrt{1+x}$ bi izgledala ovako, redom, prvi tip, racionalna funkcija, drugi tip:

$$\begin{aligned} \sqrt{1+x} &= 1+\frac{x}{2^+}\,\frac{x}{2^+}\,\frac{x}{2^+}\,\frac{x}{2^+}\,\frac{x}{2^+}\,\frac{x}{2} = \frac{7x^3+56x^2+112x+64}{x^3+24x^2+80x+64} \\ &= 7+\frac{-112}{(x+20)^+}\,\frac{-24/7}{(x+8/3)^+}\,\frac{-8/63}{(x+4/3)}. \end{aligned}$$

Što vidimo? Iako drugi tip ima kompliciranije koeficijente, ima upola manje karika za izvrednjavanje, pa će to dva puta ubrzati postupak izvrednjavanja. Stoga se javlja potreba za pretvaranje verižnog razlomka prvog tipa u verižni razlomak drugog tipa. Postupak se obavlja u dva koraka.

U prvom koraku od verižnog razlomka prvog tipa dobivamo racionalnu funkciju. Pogledajmo silazni algoritam za izvrednjavanje verižnog razlomka prvog tipa. F_k želimo napisati kao kvocijent dva polinoma, pa možemo definirati

$$F_k = \frac{u_k}{v_k}.$$

Tada silazna rekurzija glasi

$$\frac{u_k}{v_k} = \beta_k + \frac{(x-x_{k+1})v_{k+1}}{u_{k+1}}.$$

Kao što smo to i prije radili, izjednačimo brojnike i nazivnike funkcija s obje strane. Dobivamo

$$u_k = \beta_k u_{k+1} + (x - x_{k+1})v_{k+1},$$
$$v_k = u_{k+1}.$$

Naravno v_k možemo eliminirati uvrštavanjem iz druge jednadžbe u prvu, pa dobivamo

$$u_k = \beta_k u_{k+1} + (x - x_{k+1})u_{k+2}, \quad k = n, n-1, \ldots, 0,$$

uz start $u_{n+2} = 0$, $u_{n+1} = 1$. Konačno, n-ta je konvergencija jednaka

$$f_n(x) = F_0 = \frac{u_0}{v_0} = \frac{u_0}{u_1}.$$

Da bismo iz racionalne funkcije dobili drugi tip verižnog razlomka, potrebno je koristiti silaznu rekurziju za drugi tip i uspoređivati s u_0/u_1. Iz silazne rekurzije za drugi tip izlazi

$$\frac{u_0}{u_1} = \tilde{b}_0 + \frac{a_1}{x + F_1},$$

pa možemo pisati

$$u_0 = u_1\tilde{b}_0 + a_1\widetilde{R}_1.$$

Zatim ponovimo postupak i dobivamo

$$u_1 = \widetilde{R}_1\tilde{b}_1 + a_2\widetilde{R}_2.$$

Ova rekurzija se prekida kad je stupanj polinoma 0.

Algoritam za pretvaranje racionalne funkcije u drugi tip verižnog razlomka je sljedeći. Definira se $\widetilde{R}_{-1} = u_0$ i $\widetilde{R}_0 = u_1$. Zatim se vrti petlja

$$\widetilde{R}_{k-1} = \widetilde{R}_k\tilde{b}_k + a_{k+1}\widetilde{R}_{k+1} \quad \text{za} \quad k = 0, 1, 2, \ldots$$

sve dok ne bude ispunjen uvjet $\widetilde{R}_x = 1$. Pritom je

$$\tilde{b}_k = \begin{cases} b_0, & k = 0, \\ b_k + x, & k \neq 0. \end{cases}$$

7. Aproksimacija i interpolacija

7.1. Opći problem aproksimacije

Što je problem aproksimacije? Ako su poznate neke informacije o funkciji f, definiranoj na nekom skupu $X \subseteq \mathbb{R}$, na osnovu tih informacija želimo f zamijeniti nekom drugom funkcijom φ na skupu X, tako da su f i φ bliske u nekom smislu. Skup X je najčešće interval oblika $[a, b]$ (može i neograničen), ili diskretni skup točaka.

Problem aproksimacije javlja se u dvije bitno različite formulacije.

(a) Poznata je funkcija f (npr. analitički), ali je njena forma prekomplicirana za računanje. U tom slučaju odabiremo neke informacije o f i po nekom kriteriju odredimo aproksimacijsku funkciju φ. U tom slučaju možemo birati informacije o f koje ćemo koristiti. Jednako tako, možemo ocijeniti grešku dobivene aproksimacije, obzirom na pravu vrijednost funkcije f.

(b) Funkcija f nije poznata, ali su poznate samo neke informacije o njoj, na primjer, vrijednosti na nekom skupu točaka. Zamjenska funkcija φ određuje se iz raspoloživih informacija, koje, osim samih podataka, uključuju i očekivani oblik ponašanja podataka, tj. funkcije φ. U ovom se slučaju **ne može** napraviti ocjena pogreške bez dodatnih informacija o nepoznatoj funkciji f.

Varijanta (b) je puno češća u praksi. Najčešće se javlja kod mjerenja raznih veličina, jer, osim izmjerenih podataka, pokušavamo aproksimirati i podatke koji se nalaze "između" izmjerenih točaka. Primijetimo da se kod mjerenja javljaju i pogreške mjerenja, pa postoje posebne tehnike za ublažavanje tako nastalih grešaka.

Funkcija φ bira se prema prirodi modela, ali tako da bude relativno jednostavna za računanje. Ona obično ovisi o parametrima a_k, $k = 0, \ldots, m$, koje treba odrediti po nekom kriteriju,

$$\varphi(x) = \varphi(x; a_0, a_1, \ldots, a_m).$$

Kad smo funkciju φ zapisali u ovom obliku, kao funkciju koja ovisi o parametrima a_k, onda kažemo da smo odabrali opći oblik aproksimacijske funkcije.

Oblike aproksimacijskih funkcija možemo (grubo) podijeliti na:

(a) linearne aproksimacijske funkcije,

(b) nelinearne aproksimacijske funkcije.

Bitne razlike između ove dvije grupe aproksimacijskih funkcija opisujemo u nastavku.

7.1.1. Linearne aproksimacijske funkcije

Opći oblik linearne aproksimacijske funkcije je

$$\varphi(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_m\varphi_m(x),$$

gdje su $\varphi_0, \ldots, \varphi_m$ poznate funkcije koje znamo računati. Primijetite da se linearnost ne odnosi na **oblik funkcije** φ, već na njenu ovisnost o parametrima a_k koje treba odrediti. Prednost ovog oblika aproksimacijske funkcije je da određivanje parametara a_k obično vodi na **sustave linearnih jednadžbi**.

Navedimo najčešće korištene oblike linearnih aproksimacijskih funkcija.

1. Algebarski polinomi, $\varphi_k(x) = x^k$, $k = 0, \ldots, m$, tj.

   $$\varphi(x) = a_0 + a_1x + \cdots + a_mx^m.$$

   Funkciju $\varphi(x)$ ne moramo nužno zapisati u standardnoj bazi običnih potencija $\{1, x, \ldots, x^m\}$. Vrlo često je neka druga baza bitno pogodnija, na primjer, tzv. ortogonalnih polinoma ili baza $\{1, (x-x_0), (x-x_0)\,(x-x_1), \ldots\}$, gdje su $x_0, x_1, \ldots$ zadane točke.

2. Trigonometrijski polinomi, pogodni za aproksimaciju periodičkih funkcija, recimo, u modeliranju signala. Za funkcije φ_k uzima se $m+1$ funkcija iz skupa

   $$\{1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots\}.$$

   Katkad se koristi i faktor u argumentu sinusa i kosinusa koji služi za kontrolu perioda, a ponekad se biraju samo parne ili samo neparne funkcije iz ovog skupa.

3. Po dijelovima polinomi, tzv. splajn funkcije. Ako su zadane točke $x_0, \ldots, x_n$, onda se splajn funkcija na svakom podintervalu svodi na polinom određenog fiksnog (niskog) stupnja, tj.

   $$\varphi\,\Big|_{[x_{k-1},x_k]} = p_k, \quad k = 1, 2, \ldots, n,$$

   a p_k su polinomi najčešće stupnjeva 1, 2, 3 ili 5. U točkama x_i obično se zahtijeva da funkcija φ zadovoljava još i tzv. "uvjete ljepljenja" vrijednosti funkcije i nekih njenih derivacija ili nekih aproksimacija tih derivacija. Splajnovi se danas često koriste zbog dobrih svojstava obzirom na grešku aproksimacije i kontrolu oblika aproksimacijske funkcije.

7.1.2. Nelinearne aproksimacijske funkcije

Navedimo najčešće korištene oblike nelinearnih aproksimacijskih funkcija.

4. Eksponencijalne aproksimacije

$$\varphi(x) = c_0 e^{b_0 x} + c_1 e^{b_1 x} + \cdots + c_r e^{b_r x},$$

koje imaju $n = 2r + 2$ nezavisna parametra, a opisuju, na primjer, procese rasta i odumiranja u raznim populacijama, s primjenom u biologiji, ekonomiji i medicini;

5. Racionalne aproksimacije

$$\varphi(x) = \frac{b_0 + b_1 x + \cdots + b_r x^r}{c_0 + c_1 x + \cdots + c_s x^s},$$

koje imaju $n = r+s+1$ nezavisni parametar, a ne $r+s+2$, kako formalno piše. Naime, razlomci se mogu proširivati (ili skalirati), pa ako su b_i, c_i parametri, onda su to i tb_i, tc_i, za $t \neq 0$. Zbog toga se uvijek fiksira jedan od koeficijenata b_i ili c_i, a koji je to — obično slijedi iz prirode modela.

Ovako definirane racionalne funkcije imaju mnogo bolja svojstva aproksimacije nego polinomi, a pripadna teorija je relativno nova.

7.1.3. Kriteriji aproksimacije

Aproksimacijske funkcije biraju se tako da "najbolje" zadovolje uvjete koji se postavljaju na njih. Najčešći su zahtjevi da graf aproksimacijske funkcije prolazi određenim točkama tj. da interpolira funkciju u tim točkama ili da je odstupanje aproksimacijske od polazne funkcije u nekom smislu minimalno, tj. tada se minimizira pogreška.

Interpolacija

Interpolacija je zahtjev da se vrijednosti funkcija f i φ podudaraju na nekom konačnom skupu argumenata (ili kraće točaka). Te točke obično nazivamo **čvorovima** interpolacije. Ovom zahtjevu se može, ali i ne mora dodati zahtjev da se u čvorovima, osim funkcijskih vrijednosti, poklapaju i vrijednosti nekih derivacija.

Drugim riječima, u najjednostavnijem obliku interpolacije, kad tražimo samo podudaranje funkcijskih vrijednosti, od podataka o funkciji f koristi se samo informacija o njenoj vrijednosti na skupu od $(n + 1)$ točaka, tj. podaci oblika (x_k, f_k), gdje je $f_k = f(x_k)$, za $k = 0, \ldots, n$.

Parametri $a_0, \ldots, a_n$ (kojih mora biti točno onoliko koliko i podataka!) određuju se iz uvjeta

$$\varphi(x_k; a_0, a_1, \ldots, a_n) = f_k, \quad k = 0, \ldots, n,$$

što je, općenito, nelinearni sustav jednadžbi. Ako je aproksimacijska funkcija φ linearna, onda za parametre a_k dobivamo sustav linearnih jednadžbi koji ima točno $n+1$ jednadžbi i $n+1$ nepoznanica. Matrica tog sustava je **kvadratna**, što bitno olakšava analizu egzistencije i jedinstvenosti rješenja za parametre interpolacije.

Minimizacija pogreške

Minimizacija pogreške je drugi kriterij određivanja parametara aproksimacije. Funkcija φ bira se tako da se minimizira neka odabrana **norma** pogreške

$$e(x) = f(x) - \varphi(x)$$

u nekom odabranom vektorskom prostoru funkcija definiranih na nekoj domeni X. Ove aproksimacije, često zvane i najbolje aproksimacije po normi, dijele se na diskretne i kontinuirane, ovisno o tome minimizira li se norma pogreške e na diskretnom ili kontinuiranom skupu podataka X.

Standardno se kao norme pogreške koriste 2-norma i ∞-norma. Za 2-normu pripadna se aproksimacija zove **srednjekvadratna**, a metoda za njeno nalaženje zove se **metoda najmanjih kvadrata**. Funkcija φ, odnosno njeni parametri, se traže tako da bude $\|e\|_2$ minimalna na X.

U diskretnom slučaju je $X = \{x_0, \ldots, x_n\}$, pa je zathjev minimalnosti

$$\sqrt{\sum_{k=0}^{n} (f(x_k) - \varphi(x_k))^2} \to \min,$$

dok je u kontinuiranom slučaju

$$\sqrt{\int_a^b (f(x) - \varphi(x))^2 \, dx} \to \min .$$

Preciznije, minimizira se samo ono pod korijenom. Tako se dobiva jednako rješenje kao da se minimizira korijen tog izraza, jer je drugi korijen rastuća funkcija.

Zašto se baš minimiziraju kvadrati grešaka? To ima veze sa statistikom, jer se izmjereni podaci obično ponašaju kao normalna slučajna varijabla, s očekivanjem koje je točna vrijednost podatka. Odgovarajući kvadrati su varijanca i nju treba minimizirati.

U slučaju ∞-norme pripadna se aproksimacija zove **minimaks** aproksimacija, a parametri se biraju tako da $\|e\|_\infty$ bude minimalna. U diskretnom slučaju problem se svodi na

$$\max_{k=0,\ldots,n} |f(x_k) - \varphi(x_k)| \to \min,$$

a u kontinuiranom

$$\max_{x\in[a,b]} |f(x) - \varphi(x)| \to \min\,.$$

U nekim problemima ovaj je tip aproksimacija poželjniji od srednjekvadratnih, jer se traži da maksimalna greška bude minimalna, tj. najmanja moguća, ali ih je općenito mnogo teže izračunati (na primjer, dobivamo problem minimizacije nederivabilne funkcije).

Napomenimo još da smo u prethodnim primjerima koristili uobičajene (diskretne i kontinuirane) norme na odgovarajućim prostorima funkcija, ovisno o domeni X. Naravno, normirani prostor u kojem tražimo aproksimacijsku funkciju ovisi o problemu kojeg rješavamo. Nerijetko u praksi, norme, pored funkcije uključuju i neke njene derivacije. Primjer takve norme je norma

$$\|f\| = \sqrt{\int_a^b (f(x))^2 + (f'(x))^2\,dx},$$

na prostoru $C^1[a,b]$ svih funkcija koje imaju neprekidnu prvu derivaciju na segmentu $[a,b]$.

Objasnimo još koja je uloga "parametrizacije" aproksimacijskih funkcija. Očito, riječ je o izboru prikaza ili "baze" u prostoru aproksimacijskih funkcija ili načinu zadavanja tog prostora. Dok prva dva problema uglavnom ne ovise o "parametrizaciji", kao što ćemo vidjeti, dobar izbor "baze" je ključan korak u konstrukciji točnih i efikasnih algoritama.

Problem interpolacije možemo smatrati specijalnim, ali posebno važnim slučajem aproksimacije po normi na diskretnom skupu X čvorova interpolacije uz neku od standardnih normi na konačnodimenzionalnim prostorima. Posebnost se ogleda u činjenici što se može postići da je minimum norme pogreške jednak nuli, a to je ekvivalentno odgovarajućim uvjetima interpolacije.

Na primjer, uzmimo da je $X = \{x_0, \ldots, x_n\}$ i tražimo aproksimacijsku funkciju φ u prostoru $\mathcal{P}_n$ svih polinoma stupnja najviše n. Kao kriterij aproksimacije uzmimo bilo koju p-normu ($1 \le p \le \infty$) vektora e pogreške funkcijskih vrijednosti na skupu X, tj.

$$\|e\|_p = \|f - \varphi\|_p = \Big(\sum_{k=0}^{n} |f(x_k) - \varphi(x_k)|^p \Big)^{1/p} \to \min, \quad 1 \le p < \infty,$$

odnosno

$$\|e\|_\infty = \|f - \varphi\|_\infty = \max_{k=0,\dots,n} |f(x_k) - \varphi(x_k)| \to \min .$$

Očito je $\|e\|_p = 0$ ekvivalentno uvjetima interpolacije

$$f(x_k) = \varphi(x_k), \quad k = 0, \dots, n,$$

samo nije jasno da li se to može postići, tj. da li postoji takva aproksimacijska funkcija $\varphi \in \mathcal{P}_n$ za koju je minimum greške jednak nuli, tako da je φ i interpolacijska funkcija. U odjeljaku 7.2. pokazat ćemo da je odgovor za ovaj primjer potvrdan.

Osnovni matematički problemi u teoriji aproksimacije

Pri kraju ovog uvoda u opći problem aproksimacije funkcija postaje jasno koji su ključni matematički problemi koje treba riješiti:

- egzistencija i jedinstvenost rješenja problema aproksimacije, što ovisi o tome koje funkcije f aproksimiramo kojim funkcijama φ (dva prostora) i kako mjerimo grešku,
- analiza kvalitete dobivene aproksimacije — vrijednost "najmanje" pogreške i ponašenje funkcije greške e (jer norma je ipak samo broj),
- konstrukcija algoritama za računanje najbolje aproksimacije,
- dokaz efikasnosti i točnosti algoritma, a ako je proces beskonačan njegovu globalnu i asimptotsku konvergenciju.

7.2. Interpolacija polinomima

Pretpostavimo da imamo funkciju f zadanu na diskretnom skupu različitih točaka x_k, $k = 0, \dots, n$, tj. $x_i \neq x_j$ za $i \neq j$. Poznate funkcijske vrijednosti u tim točkama skraćeno označavamo s $f_k = f(x_k)$.

Primijetite da pretpostavka o različitosti točaka nije bitno ograničenje. Naime, kad bismo dozvolili da je $x_i = x_j$ uz $i \neq j$, ili f ne bi bila funkcija (ako je $f_i \neq f_j$) ili bismo imali redundantan podatak (ako je $f_i = f_j$), koji možemo ispustiti.

Ako je $[a, b]$ segment na kojem koristimo interpolaciju (i promatramo grešku), u praksi su točke obično numerirane tako da vrijedi $a \leq x_0 < x_1 < \cdots < x_n \leq b$.

7.2.1. Egzistencija i jedinstvenost interpolacijskog polinoma

Za polinomnu interpolaciju vrijedi sljedeći teorem, čiji dokaz koristi činjenicu da linearni sustav s regularnom matricom ima jedinstveno rješenje. U iskazu teorema koristi se oznaka $\mathbb{N}_0$ za skup cijelih nenegativnih brojeva.

Teorem 7.2.1 *Neka je $n \in \mathbb{N}_0$. Za zadane točke (x_k, f_k), $k = 0, \ldots, n$, gdje je $x_i \neq x_j$ za $i \neq j$, postoji jedinstveni (interpolacijski) polinom stupnja najviše n*

$$\varphi(x) := p_n(x) = a_0 + a_1 x + \cdots + a_n x^n$$

za koji vrijedi

$$p_n(x_k) = f_k, \quad k = 0, \ldots, n.$$

Dokaz. Neka je $p_n = a_0 + a_1 x + \cdots + a_n x^n$ polinom stupnja najviše n. Uvjete interpolacije možemo napisati u obliku

$$\begin{aligned}
p_n(x_0) &= a_0 + a_1 x_0 + \cdots + a_n x_0^n = f_0 \\
p_n(x_1) &= a_0 + a_1 x_1 + \cdots + a_n x_1^n = f_1 \\
&\cdots\cdots\cdots \\
p_n(x_n) &= a_0 + a_1 x_n + \cdots + a_n x_n^n = f_n.
\end{aligned}$$

Drugim riječima, treba provjeriti ima li ovaj sustav od $(n+1)$-e linearne jednadžbe s $(n+1)$-om nepoznanicom $a_0, \ldots, a_n$ jedinstveno rješenje. Za to je dovoljno provjeriti je li kvadratna matrica tog linearnog sustava regularna. To možemo napraviti računanjem vrijednosti determinante te matrice. Ta determinanta je tzv. Vandermondeova determinanta

$$D_n = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^n \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix}.$$

Definirajmo determinantu koja naliči na D_n, samo umjesto potencija od x_n, posljednji redak ima potencije od x:

$$V_n(x) = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^n \\ 1 & x & x^2 & \cdots & x^n \end{vmatrix}.$$

Primijetimo da je $D_n = V_n(x_n)$. Gledamo li $V_n(x)$ kao funkciju od x, lako se vidi, razvojem po posljednjem retku, da je to polinom stupnja najviše n u varijabli x, s vodećim koeficijentom D_{n-1} uz x^n.

S druge strane, ako za x redom uvrštavamo $x_0, \ldots, x_{n-1}$, determinanta $V_n(x)$ će imati dva jednaka retka pa će biti

$$V_n(x_0) = V_n(x_1) = \cdots = V_n(x_{n-1}) = 0,$$

Dakle, točke $x_0, \ldots, x_{n-1}$ su nultočke polinoma $V_n(x)$ stupnja n. Da bismo točno odredili polinom stupnja n, ako su poznate njegove nultočke, potrebno je samo znati njegov vodeći koeficijent. U ovom slučaju, pokazali smo da je to D_{n-1}. Odatle odmah slijedi

$$V_n(x) = D_{n-1}\,(x - x_0)\,(x - x_1)\cdots(x - x_{n-1}).$$

Kad uvrstimo $x = x_n$, dobivamo rekurzivnu relaciju za D_n

$$D_n = D_{n-1}\,(x_n - x_0)\,(x_n - x_1)\cdots(x_n - x_{n-1}).$$

Ako znamo da je $D_0 = 1$, što je trivijalno, dobivamo da je

$$D_n = \prod_{0 \le j < i \le n} (x_i - x_j).$$

Budući da je $x_i \neq x_j$ za $i \neq j$, onda je $D_n \neq 0$, tj. matrica linearnog sustava je regularna, pa postoji jedinstveno rješenje $a_0, \ldots, a_n$ za koeficijente polinoma p_n, odnosno jedinstven interpolacijski polinom. ■

Ovaj teorem u potpunosti rješava prvo ključno pitanje egzistencije i jedinstvenosti rješenja problema polinomne interpolacije u njegovom najjednostavnijem obliku — kad su zadane funkcijske vrijednosti u međusobno različitim točkama.

Takav oblik interpolacije, kad tražena funkcija (u ovom slučaju polinom) mora interpolirati samo funkcijske vrijednosti zadane funkcije, obično zovemo **Lagrangeova interpolacija**. U općenitijem slučaju, možemo zahtijevati interpolaciju zadanih vrijednosti funkcije i njezinih uzastopnih derivacija. Takvu interpolaciju zovemo **Hermiteova interpolacija**. Nešto kasnije ćemo pokazati da problem Hermiteove interpolacije možemo riješiti kao granični slučaj Lagrangeove, kad dozvolimo višestruko "ponavljanje" istih čvorova, tj. otpustimo ograničenje na međusobnu različitost čvorova.

Za početak, moramo riješiti preostala dva problema vezana uz polinomnu Lagrangeovu interpolaciju, a to su: konstrukcija algoritama i analiza greške.

7.2.2. Kako naći prave algoritme?

Kakve algoritme trebamo? Odgovor ovisi o tome što želimo postići interpolacijom. Kao i kod svih aproksimacija, očita primjena je zamjena funkcijskih vrijednosti $f(x)$ vrijednostima interpolacijskog polinoma $p_n(x)$, i to u točkama x koje u

principu **nisu** čvorovi interpolacije. To je posebno bitno ako vrijednosti funkcije f ne znamo u ostalim točkama, ili se one teško računaju (recimo algoritam zahtijeva vrlo mnogo operacija ili je nestabilan, pa treba primijeniti posebne tehnike računanja). U tom smo slučaju jedva izračunali i ove vrijednosti od f koje smo iskoristili za interpolaciju.

Dakle, sigurno trebamo algoritam za računanje vrijednosti interpolacijskog polinoma u nekoj zadanoj točki x koja nije čvor, jer u čvorovima ionako vrijedi $f(x) = p_n(x)$.

Točaka x u kojima želimo izračunati $p_n(x)$ može biti vrlo mnogo, a gotovo nikad nije samo jedna. Zbog toga se problem računanja vrijednosti $p_n(x)$ uvijek rješava u dvije faze:

1. prvo nađemo polinom p_n, jer on ne ovisi o točki x, već samo o zadanim podacima (x_k, f_k), $k = 0, \ldots, n$,
2. zatim, za svaku zadanu točku x izračunamo $p_n(x)$.

Prvu fazu je dovoljno napraviti samo jednom i zato svaku od ovih faza treba realizirati posebnim algoritmom. Dodatno, želimo što brži algoritam, posebno u drugoj fazi, jer se on tamo puno puta izvršava. Međutim, nećemo zahtijevati brzinu na uštrb stabilnosti, ako se to može izbjeći, bez većeg gubitka brzine.

Pogledajmo detaljnije prvu fazu. Što znači "naći polinom p_n"? Broj podataka $n+1$ u potpunosti određuje ($(n+1)$-dimenzionalni) vektorski prostor polinoma $\mathcal{P}_n$ (stupnja manjeg ili jednakog n) u kojem, prema teoremu 7.2.1, postoji jedinstveni polinom p_n koji interpolira zadane podatke. Izaberimo neku bazu $\{b_0, b_1, \ldots, b_n\}$ u tom prostoru $\mathcal{P}_n$. Polinom p_n ima u (svakoj pa zato i izabranoj) bazi jedinstven prikaz, kao linearna kombinacija polinoma b_i iz te baze. Dakle, da bismo našli p_n, treba (i dovoljno je) naći koeficijente a_i u prikazu

$$p_n(x) = \sum_{i=0}^{n} a_i b_i(x).$$

Njih možemo naći tako da u ovu relaciju uvrstimo sve uvjete interpolacije

$$p_n(x_k) = \sum_{i=0}^{n} a_i b_i(x_k) = f_k, \quad k = 0, \ldots, n,$$

i tako dobijemo linearni sustav reda $n+1$ za nepoznate koeficijente. Matrica B tog linearnog sustava je sigurno regularna, jer su bazne funkcije b_i linearno nezavisne i svaka dva različita polinoma stupnja $\leq n$ ne mogu imati jednake vrijednosti u svih $n+1$ točaka x_k. Elementi matrice B imaju oblik $B_{i+1,k+1} = b_i(x_k)$, za $i, k = 0, \ldots, n$.

U pripadnom algoritmu, prvo treba izračunati sve elemente matrice linearnog sustava, a zatim ga riješiti. Ako pretpostavimo da znamo prikaze svih polinoma b_i

u standardnoj bazi i koristimo Hornerovu shemu za izvrednjavanje u svim točkama, onda svako izvrednjavanje $b_i(x_k)$ zahtijeva najviše $2n$ operacija (n množenja i n zbrajanja, vidi odjeljak 6.1.). Takvih izvrednjavanja ima najviše $(n+1)^2$, pa sve elemente matrice sustava možemo izračunati s najviše $2n(n+1)^2$ operacija, tj. s $O(n^3)$ operazija. Za posebne izbore baza i čvorova, ovaj broj operacija može biti i bitno manji.

Gaussovim eliminacijama (ili LR faktorizacijom) možemo riješiti dobiveni linearni sustav za najviše $O(n^3)$ operacija. Dakle, ukupan broj operacija u algoritamu za prvu fazu je najviše reda veličine $O(n^3)$. To, samo po sebi i nije tako loše, jer se izvršava samo jednom. Međutim, u nastavku ćemo pokazati da pažljivim izborom baze to računanje možemo napraviti i bitno brže.

Algoritam za izvrednjavanje $p_n(x)$ u drugoj fazi, također, fundamentalno ovisi o izboru baze u $\mathcal{P}_n$. Naravno, iz prve faze treba zapamtiti izračunati vektor koeficijenata a_i. Tada se računanje $p_n(x)$ u zadanoj točki x svodi na računanje sume

$$p_n(x) = \sum_{i=0}^{n} a_i b_i(x).$$

U najopćenitijem obliku, po ovoj relaciji imamo $(n+1)$-no računanje vrijednosti $b_i(x)$ (npr. Hornerovom shemom) i još jedan skalarni produkt vektora $(a_0, \ldots, a_n)$ i $(b_0(x), \ldots, b_n(x))$. Ukupno trajanje je $O(n^2)$, što je dosta u usporedbi s običnom Hornerovom shemom.

Uočite da ova dva opća algoritma za interpolaciju možemo sažeto prikazati u obliku:

1. izaberi bazu u $\mathcal{P}_n$ i nađi koeficijente od p_n u toj bazi,
2. u zadanoj točki x izračunaj linearnu kombinaciju polinoma baze s poznatim koeficijentima u linearnoj kombinaciji.

Iz prethodne analize slijedi da bi bilo vrlo poželjno odabrati bazu tako da druga faza zahtijeva najviše $O(n)$ operacija, tj. da traje linearno, a ne kvadratno, u funkciji od n.

Kad u ovom kontekstu pogledamo tvrdnju i dokaz teorema 7.2.1, odmah možemo zaključiti da to odgovara izboru standardne baze $b_i(x) = x^i$, $i = 0, \ldots, n$, u prostoru $\mathcal{P}_n$. U prvoj fazi za nalaženje koeficijenata interpolacijskog polinoma u standardnoj bazi ne moramo koristiti samo već spomenute numeričke metode. Osim njih, uz malo pažnje, možemo koristiti čak i Cramerovo pravilo. Determinanta D_n sustava je Vandermondeova, a sve ostale potrebne determinante se jednostavnim razvojem po stupcu svode na linearne kombinacije Vandermondeovih determinanata reda $n-1$. Ako njih izrazimo preko D_n, dobivamo opet algoritam koji treba $O(n^3)$ operacija.

Nadalje, vidimo da se druga faza svodi upravo na Hornerovu shemu, tj. ima linearno trajanje. Čak jače od toga, što se brzine tiče, ovim izborom baze dobivamo optimalan — najbrži mogući algoritam za izvrednjavanje u drugoj fazi.

Nažalost, u pogledu stabilnosti, situacija je prilično nepovoljna, posebno u prvoj fazi. Matrica sustava može imati skoro linearno zavisne retke, a da čvorovi uopće nisu "patološki" raspoređeni. Dovoljno je samo da su razumno bliski i relativno daleko od nule (što je "centar" baze). Na primjer

$$x_k = k + 10^p, \quad k = 0, \ldots, n,$$

gdje je p "iole veći" pozitivni eksponent, recimo $p = 5$ i matrica sustava će biti skoro singularna. Zbog toga se ovaj izbor baze ne koristi u praksi, već samo za dokazivanje u teoriji, jer baza ne ovisi o čvorovima.

Problemu izbora baze za prikaz interpolacijskog polinoma možemo, sasvim općenito, pristupiti na 3 načina.

1. "Jednostavna baza, komplicirani koeficijenti". Fiksiramo jednostavnu bazu u $\mathcal{P}_n$, neovisno o zadanim podacima, ali tako da dobijemo brzo izvrednjavanje. Zatim nađemo koeficijente od p_n u toj bazi. Sva ovisnost o zadanim podacima ulazi u koeficijente, pa je prva faza spora.
2. "Jednostavni koeficijenti, komplicirana baza". Podijelimo ovisnost o zadanim podacima tako da koeficijenti jednostavno ovise o zadanim podacima i lako se računaju (na primjer, jednaki su zadanim funkcijskim vrijednostima f_k). Tada je prva faza brza, ali zato baza komplicirano ovisi o čvorovima, pa je druga faza spora, jer u svakoj točki x izvrednjavamo sve funkcije baze.
3. "Kompromis između baze i koeficijenata". Pustimo da baza jednostavno ovisi o čvorovima, a koeficijenti mogu ovisiti o svim zadanim podacima, ali tako da dobijemo jednostavne algoritme u obje faze.

Ove pristupe je najlakše ilustrirati preko složenosti rješavanja linearnog sustava za koeficijente.

Prvim pristupom dobivamo puni linearni sustav za čije rješavanje treba $O(n^3)$ operacija. Ako baza ne ovisi o čvorovima, taj sustav može biti vrlo nestabilan, kao u ranijem primjeru standardne baze.

Drugi pristup vodi na dijagonalni linearni sustav u kojem se rješenje "čita" ili traje najviše $O(n)$ operacija. No, tada je izvrednjavanje u svakoj točki sporo, jer svi polinomi baze imaju puni stupanj n. Primjer takve baze je tzv. Lagrangeova baza.

U zadnjem pristupu bazu izaberemo tako da dobijemo donjetrokutasti linearni sustav. Za nalaženje koeficijenata tada trebamo "samo" $O(n^2)$ operacija. Tako dobivamo tzv. Newtonovu bazu u kojoj stupnjevi polinoma b_i rastu, tj. vrijedi $\deg b_i = i$, kao i u standardnoj bazi. Osim toga, za b_i vrijedi jednostavna rekurzija koja vodi na brzi linearni algoritam izvrednjavanja.

Sva 3 pristupa možemo vrlo lijepo ilustrirati na jednostavnom primjeru linearne interpolacije, tj. kad je $n = 1$. Problem linearne interpolacije se svodi na nalaženje jednadžbe pravca p koji prolazi kroz dvije zadane točke (x_0, f_0) i (x_1, f_1).

Standardni oblik jednadžbe pravca je $p(x) = a_0 + a_1 x$. Iz uvjeta interpolacije dobivamo linearni sustav za koeficijente a_0 i a_1

$$\begin{aligned} p(x_0) &= a_0 + a_1 x_0 = f_0 \\ p(x_1) &= a_0 + a_1 x_1 = f_1, \end{aligned}$$

odakle slijedi

$$a_0 = \frac{f_0 x_1 - f_1 x_0}{x_1 - x_0}, \quad a_1 = \frac{f_1 - f_0}{x_1 - x_0},$$

ili

$$p(x) = \frac{f_0 x_1 - f_1 x_0}{x_1 - x_0} + \frac{f_1 - f_0}{x_1 - x_0}\, x.$$

Vidimo da su koeficijenti kompliciranji, ali kad se jednom izračunaju, samo izvrednjavanje $p(x)$ je brzo (jedno množenje i jedno zbrajanje).

Pravac p možemo napisati i kao težinsku sredinu zadanih funkcijskih vrijednosti f_0 i f_1, u obliku

$$p(x) = f_0 b_0(x) + f_1 b_1(x),$$

gdje su $b_0(x)$ i $b_1(x)$ funkcije koje treba naći. Iz uvjeta interpolacije sada dobivamo jednadžbe

$$\begin{aligned} p(x_0) &= f_0 b_0(x_0) + f_1 b_1(x_0) = f_0 \\ p(x_1) &= f_0 b_0(x_1) + f_1 b_1(x_1) = f_1. \end{aligned}$$

Bez dodatnih pretpostavki, iz ovih jednadžbi ne možemo odrediti $b_0(x)$ i $b_1(x)$, jer takvih funkcija ima puno. Pretpostavimo stoga da su obje funkcije, također, polinomi prvog stupnja i to specijalnog oblika, tako da ovaj linearni sustav postane dijagonalan. Tada iz izvandijagonalnih elemenata dobivamo uvjete

$$b_1(x_0) = 0, \quad b_0(x_1) = 0,$$

a onda za dijagonalne elemente dobivamo

$$b_0(x_0) = 1, \quad b_1(x_1) = 1.$$

Vidmo da su polinomi b_0 i b_1 rješenja specijalnih problema interpolacije

$$b_i(x_k) = \delta_{ik}, \quad i, k = 0, 1,$$

tj. b_i mora biti nula u svim čvorovima osim i-tog, a u i-tom mora imati vrijednost 1. To znači da znamo sve nultočke od b_i, a vrijednost vodećeg koeficijenta izlazi iz $b_i(x_i) = 1$. Odmah možemo napisati te dvije funkcije baze u obliku

$$b_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad b_1(x) = \frac{x - x_0}{x_1 - x_0},$$

pa je

$$p(x) = f_0 \frac{x - x_1}{x_0 - x_1} + f_1 \frac{x - x_0}{x_1 - x_0}$$

što odgovara jednadžbi pravca kroz dvije točke. Ovo je Lagrangeov oblik interpolacijskog polinoma. Vidimo da funkcije baze b_0 i b_1 ovise o oba čvora interpolacije.

Jednadžbu pravca možemo napisati i tako da pravac prolazi kroz jednu točku (x_0, f_0) i ima zadani koeficijent smjera

$$p(x) = f_0 + k(x - x_0).$$

Ovaj oblik automatski zadovoljava prvi uvjet interpolacije $p(x_0) = f_0$, a iz drugog uvjeta

$$p(x_1) = f_0 + k(x_1 - x_0) = f_1$$

se lako izračuna k

$$k = \frac{f_1 - f_0}{x_1 - x_0},$$

što je poznata formula za koeficijent smjera pravca kroz dvije točke. Dobiveni oblik za p

$$p(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0)$$

je Newtonov oblik interpolacijskog polinoma. Njega možemo interpretirati na još nekoliko načina. Prvo, to je i Taylorov oblik za p napisan oko točke x_0, s tim da je **podijeljena razlika** k baš derivacija od p u točki x_0 (i, naravno, svakoj drugoj točki).

Nadalje, prvi član ovog oblika za p, u ovom slučaju konstanta f_0, je interpolacijski polinom stupnja 0 za zadanu prvu točku (x_0, f_0). Dakle, ovaj oblik za p odgovara korekciji interpolacijskog polinoma kroz prethodne točke, kad dodamo još jednu novu točku (x_1, f_1). To isto vrijedi i u općem slučaju.

Na kraju, ovaj oblik pravca možemo dobiti tako da u prostoru $\mathcal{P}_1$ izaberemo bazu b_0, b_1 koja daje donjetrokutasti linearni sustav za koeficijente c_0 i c_1 u prikazu

$$p(x) = c_0 b_0(x) + c_1 b_1(x).$$

Uvjeti interpolacije daju jednadžbe

$$\begin{aligned} p(x_0) &= c_0 b_0(x_0) + c_1 b_1(x_0) = f_0 \\ p(x_1) &= c_0 b_0(x_1) + c_1 b_1(x_1) = f_1. \end{aligned}$$

Kako ćemo dobiti donjetrokutasti linearni sustav? Postavljamo redom uvjete na polinome baze, gledajući u matrici sustava stupac po stupac, i još imamo na umu prethodnu interpretaciju "dopunjavanja" ranijeg interpolacijskog polinoma.

Za polinom b_0 u prvom stupcu matrice sustava nemamo nikavih uvjeta, pa uzimamo najjednostavniju oblik, koji odgovara interpolaciji polinomom stupnja 0 u prvom čvoru, a to je $b_0(x) = 1$. Iz prve jednadžbe supstitucijom unaprijed odmah dobivamo i $c_0 = f_0$.

Za polinom b_1 u drugom stupcu dobivamo točno jedan uvjet $b_1(x_0) = 0$. Opet uzmemo najjednostavniji oblik polinoma koji zadovoljava taj uvjet, a to je

$$b_1(x) = (x - x_0).$$

To, usput, odgovara i povećanju stupnja interpolacije kod dodavanja novog čvora. Supstitucijom unaprijed izlazi i koeficijent c_1

$$c_1 = \frac{f_1 - f_0}{x_1 - x_0}.$$

Kao što ćemo vidjeti, ovaj postupak se može nastaviti. Općenito, iz uvjeta da stupac s polinomom b_i ima donjetrokutasti oblik dobivamo da b_i mora imati nultočke u svim prethodnim čvorovima $x_0, \ldots, x_{i-1}$, pa možemo uzeti

$$b_i(x) = (x - x_0) \cdots (x - x_{i-1}),$$

što opet odgovara dizanju stupnja. Kako općenito izgledaju koeficijenti c_i, opisat ćemo malo kasnije.

7.2.3. Lagrangeov oblik interpolacijskog polinoma

Da bismo našli koeficijente interpolacijskog polinoma, nije nužno rješavati linearni sustav za koeficijente. Interpolacijski polinom p_n možemo odmah napisati korištenjem tzv. Lagrangeove baze $\{\ell_k, k = 0, \ldots, n\}$ prostora polinoma $\mathcal{P}_n$

$$p_n(x) = \sum_{k=0}^{n} f_k \ell_k(x), \tag{7.2.1}$$

pri čemu je

$$\begin{aligned} \ell_k(x) &= \frac{(x - x_0) \cdots (x - x_{k-1})\,(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})\,(x_k - x_{k+1}) \cdots (x_k - x_n)} \\ &= \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i} := \frac{\omega_k(x)}{\omega_k(x_k)}, \quad k = 0, \ldots, n. \end{aligned} \tag{7.2.2}$$

Polinomi ℓ_k su stupnja n, pa je p_n polinom stupnja najviše n. Osim toga, vrijedi

$$\ell_k(x_i) = \begin{cases} 0, & \text{za } i \neq k, \\ 1, & \text{za } i = k. \end{cases}$$

Uvrstimo li to u (7.2.1), odmah slijedi da se suma u (7.2.1) svodi na jedan jedini član za $i = k$, tj. da vrijedi

$$p_n(x_k) = f_k.$$

Oblik (7.2.1)–(7.2.2) zove se Lagrangeov oblik interpolacijskog polinoma. Taj polinom možemo napisati u još jednom, zgodnijem obliku. Definiramo

$$\omega(x) = \prod_{k=0}^{n} (x - x_k), \tag{7.2.3}$$

pa je

$$\ell_k(x) = \frac{\omega(x)}{(x - x_k)\,\omega_k(x_k)}.$$

Uvrštavanjem u (7.2.1) dobivamo da je

$$p_n(x) = \omega(x) \sum_{k=0}^{n} \frac{f_k}{(x - x_k)\,\omega_k(x_k)}. \tag{7.2.4}$$

Uočimo da je

$$\omega_k(x_k) = \omega'(x_k),$$

pa (7.2.4) možemo pisati kao

$$p_n(x) = \omega(x) \sum_{k=0}^{n} \frac{f_k}{(x - x_k)\,\omega'(x_k)}. \tag{7.2.5}$$

Ova se forma može koristiti za računanje vrijednosti polinoma u točki $x \neq x_k$, $k = 0, \ldots, n$. Prednost je što se za svaki novi x računa samo $\omega(x)$ i $(x - x_k)$, dok se $\omega_k(x_k) = \omega'(x_k)$ izračuna samo jednom za svaki k i čuva u tablici, jer ne ovisi o x.

Ukupan broj operacija je proporcionalan s n^2, a za računanje u svakoj novoj točki x, trebamo još reda veličine n operacija. Ipak, u praksi se ne koristi ovaj oblik interpolacijskog polinoma, već nešto bolji Newtonov oblik. Lagrangeov oblik interpolacijskog polinoma uglavnom se koristi u teoretske svrhe (za dokaze).

7.2.4. Ocjena greške interpolacijskog polinoma

Ako znamo još neke informacije o funkciji f, možemo napraviti i ocjenu greške interpolacijskog polinoma.

Teorem 7.2.2 *Pretpostavimo da funkcija f ima $(n+1)$-u derivaciju na segmentu $[a, b]$ za neki $n \in \mathbb{N}_0$. Neka su $x_k \in [a, b]$, $k = 0, \ldots, n$, međusobno različiti čvorovi interpolacije, tj. $x_i \neq x_j$ za $i \neq j$, i neka je p_n interpolacijski polinom za funkciju f u tim čvorovima. Za bilo koju točku $x \in [a, b]$ postoji točka ξ iz otvorenog intervala*

$$x_{\min} := \min\{x_0, \ldots, x_n, x\} < \xi < \max\{x_0, \ldots, x_n, x\} =: x_{\max}$$

takva da za grešku interpolacijskog polinoma vrijedi

$$e(x) := f(x) - p_n(x) = \frac{\omega(x)}{(n+1)!}\, f^{(n+1)}(\xi), \tag{7.2.6}$$

pri čemu je $\omega(x)$ definirana relacijom (7.2.3).

Dokaz. Ako je $x = x_k$, za neki $k \in \{0, \ldots, n\}$, iz uvjeta interpolacije i definicije polinoma ω dobivamo da su obje strane u (7.2.6) jednake 0, pa teorem očito vrijedi (ξ može biti bilo koji).

Pretpostavimo stoga da x nije čvor interpolacije. Tada je $\omega(x) \neq 0$ i grešku interpolacije možemo prikazati u obliku

$$e(x) = f(x) - p_n(x) = \omega(x)s(x),$$

gdje je $s(x)$ dobro definirano ako x nije čvor. Uzmimo sad da je x fiksan i definirajmo funkciju

$$g(t) = e(t) - \omega(t)s(x) = e(t) - \omega(t)\,\frac{e(x)}{\omega(x)}, \quad t \in [a, b]. \tag{7.2.7}$$

Funkcija pogreške e ima točno onoliko derivacija (po t) koliko i f, i one su neprekidne kad su to i odgovarajuće derivacije od f. Budući da x nije čvor, to isto vrijedi i za funkciju g, tj. $g^{(n+1)}$ postoji kao funkcija na $[a, b]$. Nađimo koliko nultočaka ima funkcija g. Ako za t uvrstimo x_k, dobivamo

$$g(x_k) = e(x_k) - \omega(x_k)\,\frac{e(x)}{\omega(x)} = 0, \quad k = 0, \ldots, n.$$

Jednako tako je i

$$g(x) = e(x) - e(x) = 0.$$

Drugim riječima, g ima barem $n+2$ nultočke na $[a, b]$. Čak i jače, sve te nultočke su na segmentu $[x_{\min}, x_{\max}]$. Budući da je g derivabilna na tom segmentu, po Rolleovom teoremu slijedi da g' ima barem $n+1$ nultočku na otvorenom intervalu $(x_{\min}, x_{\max})$. Induktivnom primjenom Rolleovog teorema zaključujemo da $g^{(j)}$ ima bar $n+2-j$ nultočaka na $(x_{\min}, x_{\max})$, za $j = 0, \ldots, n+1$. Dakle, za $j = n+1$ dobivamo da $g^{(n+1)}$ ima bar jednu nultočku $\xi \in (x_{\min}, x_{\max})$.

Iskoristimo još da je p_n polinom stupnja najviše n, a ω polinom stupnja $n+1$, pa je

$$e^{(n+1)}(t) = f^{(n+1)}(t), \quad \omega^{(n+1)}(t) = (n+1)!.$$

Deriviranjem lijeve i desne strane jednadžbe (7.2.7) $n+1$ puta, dobivamo

$$g^{(n+1)}(t) = e^{(n+1)}(t) - \omega^{(n+1)}(t)\,\frac{e(x)}{\omega(x)} = f^{(n+1)}(t) - (n+1)!\,\frac{e(x)}{\omega(x)}.$$

Konačno, ako uvažimo da je $g^{(n+1)}(\xi) = 0$, onda je

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n+1)!\,\frac{e(x)}{\omega(x)},$$

odnosno

$$e(x) = \frac{\omega(x)}{(n+1)!}\, f^{(n+1)}(\xi),$$

što je upravo (7.2.6). ■

Ako je $f^{(n+1)}$ ograničena na $[a,b]$ ili, jače, ako je $f \in C^{n+1}[a,b]$, onda se iz prethodnog teorema može dobiti sljedeća ocjena greške interpolacijskog polinoma za funkciju f u točki $x \in [a,b]$

$$|f(x) - p_n(x)| \le \frac{|\omega(x)|}{(n+1)!}\, M_{n+1}, \quad M_{n+1} := \max_{x\in[a,b]} |f^{(n+1)}(x)|.$$

Ova ocjena direktno slijedi iz (7.2.6), a korisna je ako relativno jednostavno možemo izračunati ili odozgo ocijeniti M_{n+1}.

7.2.5. Newtonov oblik interpolacijskog polinoma

Lagrangeov oblik interpolacijskog polinoma nije pogodan kad želimo povećati stupanj interpolacijskog polinoma da bismo eventualno poboljšali aproksimaciju i smanjili grešku, zbog toga što interpolacijski polinom moramo računati od početka.

Postoji forma interpolacijskog polinoma kod koje je mnogo lakše dodavati točke interpolacije, tj. povećavati stupanj interpolacijskog polinoma. Neka je p_{n-1} interpolacijski polinom koji interpolira funkciju f u točkama x_k, $k = 0, \ldots, n-1$. Neka je p_n interpolacijski polinom koji interpolira funkciju f još i u točki x_n. Polinom p_n tada možemo napisati u obliku

$$p_n(x) = p_{n-1}(x) + c(x), \tag{7.2.8}$$

gdje je c korekcija, polinom stupnja n. Također, mora vrijediti

$$c(x_k) = p_n(x_k) - p_{n-1}(x_k) = f(x_k) - f(x_k) = 0, \quad k = 0, \ldots, n-1.$$

Vidimo da su x_k nultočke od c, pa ga možemo napisati u obliku

$$c(x) = a_n\,(x - x_0)\cdots(x - x_{n-1}).$$

Nadalje, iz zadnjeg uvjeta interpolacije $p_n(x_n) = f(x_n)$, dobivamo

$$\begin{aligned} f(x_n) &= p_n(x_n) = p_{n-1}(x_n) + c(x_n) \\ &= p_{n-1}(x_n) + a_n\,(x_n - x_0)\cdots(x_n - x_{n-1}), \end{aligned}$$

odakle lako izračunavamo vodeći koeficijent a_n polinoma c

$$a_n = \frac{f(x_n) - p_{n-1}(x_n)}{(x_n - x_0)\cdots(x_n - x_{n-1})} = \frac{f(x_n) - p_{n-1}(x_n)}{\omega(x_n)}.$$

Korištenjem relacije (7.2.8), sada imamo sve elemente za računanje $p_n(x)$ u bilo kojoj točki x. Koeficijent a_n, očito je funkcija čvorova $x_0, \ldots, x_n$ i zvat ćemo ga n-ta podijeljena razlika. Formalno ćemo to označiti s

$$a_n = f[x_0, x_1, \ldots, x_n], \tag{7.2.9}$$

pa će odmah slijediti rekurzivna formula za dobivanje interpolacijskog polinoma za stupanj većeg od prethodnog

$$p_n(x) = p_{n-1}(x) + (x - x_0)\cdots(x - x_{n-1})f[x_0, \ldots, x_n]. \tag{7.2.10}$$

Da bismo bolje opisali a_n, vratimo se na Lagrangeov oblik interpolacijskog polinoma. Primijetimo da je a_n koeficijent uz vodeću potenciju x^n u p_n. Stoga iskoristimo relaciju (7.2.5), tj. nađimo koeficijent uz x^n na desnoj strani te relacije. Dobivamo

$$f[x_0, x_1, \ldots, x_n] = \sum_{k=0}^{n} \frac{f(x_k)}{\omega'(x_k)}. \tag{7.2.11}$$

Iz formule (7.2.11) slijede neka svojstva podijeljenih razlika. Primijetimo da poredak čvorova nije bitan, tj. podijeljena razlika neosjetljiva je na poredak čvorova. Druga korisna formula je formula za rekurzivno računanje podijeljenih razlika

$$f[x_0, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0}.$$

Izvedimo tu formulu. Vrijedi

$$\begin{aligned}
f[x_1, \ldots, x_n] &= \sum_{k=1}^{n} \frac{f(x_k)}{(x_k - x_1)\cdots(x_k - x_{k-1})\,(x_k - x_{k+1})\cdots(x_k - x_n)} \\
&= \sum_{k=1}^{n-1} \frac{f(x_k)(x_k - x_0)}{(x_k - x_0)\cdots(x_k - x_{k-1})\,(x_k - x_{k+1})\cdots(x_k - x_n)} \\
&\quad + \frac{f(x_n)(x_n - x_0)}{(x_n - x_0)\cdots(x_n - x_{n-1})} \\
f[x_0, \ldots, x_{n-1}] &= \sum_{k=0}^{n-1} \frac{f(x_k)}{(x_k - x_0)\cdots(x_k - x_{k-1})\,(x_k - x_{k+1})\cdots(x_k - x_{n-1})} \\
&= \sum_{k=1}^{n-1} \frac{f(x_k)(x_k - x_n)}{(x_k - x_0)\cdots(x_k - x_{k-1})\,(x_k - x_{k+1})\cdots(x_k - x_n)} \\
&\quad - \frac{f(x_0)(x_n - x_0)}{(x_0 - x_1)\cdots(x_0 - x_n)}.
\end{aligned}$$

Oduzimanjem dobivamo

$$\begin{aligned} f[x_1,\dots,x_n] - f[x_0,\dots,x_{n-1}] & \\ &= \sum_{k=1}^{n-1} \frac{f(x_k)(x_n-x_0)}{(x_k-x_0)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_n)} \\ &\quad + \frac{f(x_n)(x_n-x_0)}{(x_n-x_0)\cdots(x_n-x_{n-1})} + \frac{f(x_0)(x_n-x_0)}{(x_0-x_1)\cdots(x_0-x_n)} \\ &= (x_n-x_0)\sum_{k=0}^{n}\frac{f(x_k)}{\omega'(x_k)} = (x_n-x_0)\,f[x_0,\dots,x_n], \end{aligned}$$

čime je dokazana tražena formula. Neki autori baš tu rekurzivnu formulu koriste kao definiciju podijeljenih razlika.

Ostaje još vidjeti što je početak rekurzije za podijeljenje razlike. Ako znamo da je konstanta koja prolazi točkom $(x_0, f(x_0))$, interpolacijski polinom stupnja 0, onda je $a_0 = f[x_0] = f(x_0)$. Jednako tako vrijedi

$$f[x_k] = f(x_k),$$

pa tablicu podijeljenih razlika lako sastavljamo:

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$	$\cdots$	$f[x_0,\dots,x_n]$
x_0	$f[x_0]$				
		$f[x_0,x_1]$			
x_1	$f[x_1]$		$f[x_0,x_1,x_2]$		
		$f[x_1,x_2]$		$\ddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$f[x_0,\dots,x_n]$
		$f[x_{n-2},x_{n-1}]$		$\iddots$	
x_{n-1}	$f[x_{n-1}]$		$f[x_{n-2},x_{n-1},x_n]$		
		$f[x_{n-1},x_n]$			
x_n	$f[x_n]$				

Dakle, kad uvažimo rekurziju i oblik polinoma c u (7.2.10), dobivamo da je oblik Newtonovog interpolacijskog polinoma

$$\begin{aligned} p_n(x) = f[x_0] + (x-x_0)f[x_0,x_1] + (x-x_0)\,(x-x_1)f[x_0,x_1,x_2] \\ + \cdots + (x-x_0)\cdots(x-x_{n-1})f[x_0,x_1,\dots,x_n]. \end{aligned}$$

Primijetite da nam od tablica podijeljenih razlika treba samo "gornji rub", pa ćemo se u računanju podijeljenih razlika moći služiti jednodimenzionalnim poljem. Pretpostavimo da je na početku algoritma u i-tom elementu polja f spremljena funkcijska vrijednost $f(x_i)$. Na kraju algoritma u polju f ostavit ćemo redom $f[x_0], f[x_0,x_1], \dots, f[x_0,\dots,x_n]$.

Algoritam 7.2.1 (Algoritam za računanje podijeljenih razlika)

for $i := 1$ **to** n **do**
  for $j := n$ **to** i **do**
    $f[j] := (f[j] - f[j-1])/(x[j] - x[j-i])$;

I grešku interpolacijskog polinoma (koja je jednaka onoj kod Lagrangeovog), možemo napisati korištenjem podijeljenih razlika. Neka je $x_{n+1} \in (a, b)$ realan broj koji nije čvor. Konstruirajmo interpolacijski polinom koji prolazi točkama $x_0, \ldots, x_n$ i x_{n+1}. Dobivamo

$$\begin{aligned} p_{n+1}(x) = f[x_0] &+ (x - x_0)f[x_0, x_1] + (x - x_0)\,(x - x_1)f[x_0, x_1, x_2] \\ &+ \cdots + (x - x_0)\cdots(x - x_{n-1})f[x_0, x_1, \ldots, x_n] \\ &+ (x - x_0)\cdots(x - x_n)f[x_0, x_1, \ldots, x_n, x_{n+1}] \\ &= p_n(x) + (x - x_0)\cdots(x - x_n)f[x_0, x_1, \ldots, x_n, x_{n+1}]. \end{aligned} \tag{7.2.12}$$

Budući da je

$$p_{n+1}(x_{n+1}) = f(x_{n+1}),$$

onda iz relacije (7.2.12) slijedi

$$f(x_{n+1}) = p_n(x_{n+1}) + (x_{n+1} - x_0)\cdots(x_{n+1} - x_n)f[x_0, x_1, \ldots, x_n, x_{n+1}].$$

Usporedimo li tu formulu s ocjenom greške iz Teorema 7.2.2 (napisanu u točki x_{n+1}, a ne x)

$$f(x_{n+1}) - p_n(x_{n+1}) = \frac{\omega(x_{n+1})}{(n+1)!}\, f^{(n+1)}(\xi),$$

odmah se čita da je

$$f[x_0, x_1, \ldots, x_n, x_{n+1}] = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

za neki $\xi \in I$. Prethodna se formula uobičajeno piše u ovisnosti o varijabli x, tj. x_{n+1} se zamijeni s x (Prije nam to nije odgovaralo zbog pisanja interpolacijskog polinoma u varijabli x.)

$$f[x_0, x_1, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}. \tag{7.2.13}$$

Zajedno s (7.2.12), Newtonov interpolacijski polinom tada poprima oblik Taylorovog polinoma (s greškom nastalom zanemarivanjem viših članova), samo razvijenog oko točaka $x_0, \ldots, x_n$. To nas motivira da interpolacijski polinom u točki x izvrednjavamo na sličan način kao što se Hornerovom shemom izvrednjava vrijednost polinoma. Pretpostavimo da sukladno algoritmu 7.2.1, u polju f na mjestu i piše $f[x_0, x_1, \ldots, x_i]$.

Algoritam 7.2.2 (Algoritam izvrednjavanja interpolacijskog polinoma)

$sum := f[n];$
for $i := n-1$ **downto** 0 **do**
  $sum := sum * (x - x_i) + f[i];$
{ Na kraju je $p_n(x) = sum.$ }

7.2.6. Koliko je dobar interpolacijski polinom?

U praksi se obično koriste interpolacijski polinomi niskih stupnjeva, najčešće do 5. Zašto? Kod nekih funkcija za neki izbor točaka interpolacije, povećavanje stupnja interpolacijskog polinoma može dovesti do povećanja grešaka. Zbog toga se umjesto visokog stupnja interpolacijskog polinoma u praksi koristi po dijelovima polinomna interpolacija.

Njemački matematičar Runge prvi je uočio probleme koji nastupaju kod interpolacije na ekvidistantnoj mreži čvorova. On je konstruirao funkciju (poznatu kao Rungeova funkcija), koja ima svojstvo da niz Newtonovih interpolacijskih polinoma na ekvidistantnoj mreži ne konvergira (po točkama) prema toj funkciji kad se broj čvorova povećava.

Primjer 7.2.1 (Runge, 1901.) *Promotrimo (Rungeovu) funkciju*

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5].$$

Odaberimo n i izaberimo ekvidistantne čvorove interpolacije x_k, $k = 0, \ldots, n$

$$x_k = -5 + kh, \quad h = \frac{10}{n}, \quad k = 0, \ldots, n.$$

Zanima nas ponašanje grešaka koje nastaju povećavanjem stupnja n interpolacijskog polinoma. Po Teoremu 7.2.2*, uvažavanjem relacije* (7.2.13)*, dobivamo*

$$e_n(x) = f(x) - p_n(x) = \omega(x)\, f[x_0, x_1, \ldots, x_n, x].$$

Tvrdimo da vrijedi

$$f[x_0, x_1, \ldots, x_n, x] = f(x) \cdot \frac{(-1)^{r+1}}{\prod_{k=0}^{r}(1+x_k^2)} \cdot \begin{cases} 1, & \text{ako je } n = 2r+1, \\ x, & \text{ako je } n = 2r. \end{cases} \tag{7.2.14}$$

Prvo pokažimo tvrdnju za $n = 2r + 1$, korištenjem indukcije po r. U tom slučaju imamo paran broj interpolacijskih točaka, koje su simetrične obzirom na ishodište, tj. zadovoljavaju

$$x_k = -x_{n-k}.$$

Ako je $r = 0$, onda je $n = 1$ i $x_1 = -x_0$, a zbog parnosti funkcije f i $f(-x_0) = f(x_0)$. Izračunajmo podijeljenu razliku

$$\begin{aligned} f[x_0, x_1, x] = f[x_0, -x_0, x] &= \frac{f[-x_0, x] - f[x_0, -x_0]}{x - x_0} \\ &= \frac{f(x) - f(x_0)}{x^2 - x_0^2} = \frac{\dfrac{1}{1+x^2} - \dfrac{1}{1+x_0^2}}{x^2 - x_0^2} \\ &= \frac{1}{a + x^2}\,\frac{-1}{1+x_0^2} = f(x)\,\frac{-1}{1+x_0^2}. \end{aligned}$$

Time je pokazana baza indukcije. Provedimo korak indukcije, tj. prijelaz s r u $r+1$, (što znači da u n "skačemo" za 2). Neka vrijedi (7.2.14) za $n = 2r + 1$ i **bilo koji** *skup od r parova simetričnih točaka ($x_k = -x_{n-k}$, $k = 1, \ldots, r$). Neka je $m = n + 2 = 2(r + 1) + 1$. Definiramo funkciju*

$$g(x) = f[x_1, \ldots, x_{m-1}, x].$$

Zbog definicije g, po pretpostavci indukcije, vrijedi

$$g(x) = f(x) \cdot a_r, \quad a_r = \frac{(-1)^{r+1}}{\prod_{k=1}^{r}(1 + x_k^2)},$$

Po definiciji podijeljenjih razlika, lako je pokazati da vrijedi

$$g[x_0, x_m, x] = f[x_0, \ldots, x_m, x].$$

Osim toga je

$$g[x_0, x_m, x] = a_r f[x_0, x_m, x] = a_r\, f(x)\frac{-1}{1+x_0^2},$$

što zaključuje korak indukcije. Za paran n, dokaz je vrlo sličan.

Budući da je

$$(x - x_k)\,(x - x_{n-k}) = (x - x_k)\,(x + x_k) = x^2 - x_k^2,$$

onda je za $n = 2r + 1$

$$\prod_{k=0}^{n}(x - x_k) = \prod_{k=0}^{r}(x^2 - x_k^2).$$

U parnom je slučaju $n = 2r$, $x_r = 0$, pa izdvajanjem srednje točke dobivamo

$$\prod_{k=0}^{n}(x - x_k) = x \cdot \prod_{k=0}^{r-1}(x^2 - x_k^2) = \frac{1}{x} \cdot \prod_{k=0}^{r}(x^2 - x_k^2),$$

ili zajedno

$$\omega(x) = \prod_{k=0}^{n}(x - x_k) = \prod_{k=0}^{r}(x^2 - x_k^2) \cdot \begin{cases} 1, & ako\ je\ n = 2r+1, \\ 1/x, & ako\ je\ n = 2r. \end{cases}$$

Time smo pokazali željeni oblik formule za podijeljene razlike. Ako tu formulu uvrstimo u grešku, dobivamo

$$e_n(x) = f(x) - p_n(x) = \omega(x)\, f(x) \cdot \frac{(-1)^{r+1}}{\prod_{k=0}^{r}(1 + x_k^2)} \cdot \begin{cases} 1, & ako\ je\ n = 2r+1, \\ x, & ako\ je\ n = 2r. \end{cases}$$

$$= (-1)^{r+1}\, f(x)\, g_n(x),$$

gdje je

$$g_n(x) = \prod_{k=0}^{r} \frac{x^2 - x_k^2}{1 + x_k^2}. \tag{7.2.15}$$

Funkcija f pada od 0 do 5, pa se zbog simetrije, njena najveća vrijednost nalazi u 0, a najmanja u ± 5, pa imamo

$$\frac{1}{26} \le f(x) \le 1.$$

Zbog toga, konvergencija Newtonovog polinoma ovisi samo o $g_n(x)$. Osim toga je i g_n parna funkcija, tj. $g_n(x) = g_n(-x)$, pa možemo sve gledati na intervalu $[0, 5]$.

I apsolutnu vrijednost funkcije g_n možemo napisati na malo neobičan način

$$|g_n(x)| = \left(e^{h \ln |g_n(x)|}\right)^{1/h}.$$

Prema (7.2.15)*, za eksponent eksponencijalne funkcije imamo*

$$h \ln |g_n(x)| = h \cdot \sum_{k=0}^{r} \ln \left| \frac{x^2 - x_k^2}{1 + x_k^2} \right|.$$

Tvrdimo da je

$$\lim_{n\to\infty} h \ln |g_n(x)| = \lim_{r\to\infty} h \cdot \sum_{k=0}^{r} \ln \left| \frac{x^2 - x_k^2}{1 + x_k^2} \right|$$

$$= \int_{-5}^{0} \ln \left| \frac{x^2 - \xi^2}{1 + \xi^2} \right| d\xi =: q(x).$$

Ostavimo li zasad jednakost posljednje sume i integrala po strani (treba naći malo složeniji limes), primijetimo da se integral može izračunati analitički

$$\int_{-5}^{0} \ln \left| \frac{x^2 - \xi^2}{1 + \xi^2} \right| d\xi = (5 + x) \ln(5 + x) + (5 - x) \ln(5 - x) - 5 \ln 26 - 2 \operatorname{arctg} 5.$$

Analizom toka funkcije vidimo da $q(x)$ ima jednu nultočku u intervalu $[0,5]$, približno jednaku 3.63 *(možemo ju i točnije odrediti). Preciznije, zbog parnosti funkcije q, na $[-5,5]$ vrijedi*

$$q(x) = 0 \ za \ |x| = 3.63,$$
$$q(x) < 0 \ za \ |x| < 3.63,$$
$$q(x) > 0 \ za \ 3.63 < |x| \leq 5.$$

Za $|x| > 3.63$ i $h = 10/n$ slijedi da je

$$\lim_{n\to\infty} |g_n(x)| = \infty,$$

pa i

$$e_n(x) \to \infty,$$

tj. niz interpolacijskih polinoma divergira za $|x| > 3.63$!

Zanimljivo je da, ako umjesto ekvidistantnih točaka interpolacije u primjeru Runge uzmemo neekvidistantne, točnije tzv. Čebiševljeve točke na intervalu $[a,b]$,

$$x_k = \frac{1}{2}\Big(a + b + (a-b)\cos\frac{2k+1}{2n+2}\Big), \quad k = 0, \ldots, n.$$

onda će porastom stupnja niz interpolacijskih polinoma konvergirati prema funkciji f.

Zadatak 7.2.1 *Dokažite da vrijedi*

$$\lim_{n\to\infty} h \ln|g_n(x)| = \int_{-5}^{0} \ln\left|\frac{x^2-\xi^2}{1+\xi^2}\right| d\xi.$$

Uputa: Očito je

$$\ln\left|\frac{x^2-x_k^2}{1+x_k^2}\right| = \ln|x+x_k| + \ln|x-x_k| - \ln|1+x_k^2|$$

i lako se vidi da je

$$\lim_{r\to\infty} \sum_{k=0}^{r} h \ \ln|1+x_k^2| = \int_{-5}^{0} \ln|1+\xi^2| \, d\xi$$
$$\lim_{r\to\infty} \sum_{k=0}^{r} h \ \ln|x-x_k| = \int_{-5}^{0} \ln|x-\xi| \, d\xi,$$

zbog neprekidnosti podintegralnih funkcija i definicije Riemannovog integrala, budući je riječ o specijalnim Darbouxovim sumama. Za dokaz da je

$$\lim_{r\to\infty} \sum_{k=0}^{r} h \ \ln|x+x_k| = \int_{-5}^{0} \ln|x+\xi| \, d\xi,$$

potrebno je napraviti "finu analizu" i posebno razmatrati situacije $|x + x_k| < \delta$, $|x + x_k| > \delta$, *za neki mali* $0 < \delta < 1$ *(ili se pozvati na jače teoreme iz teorije mjere).*

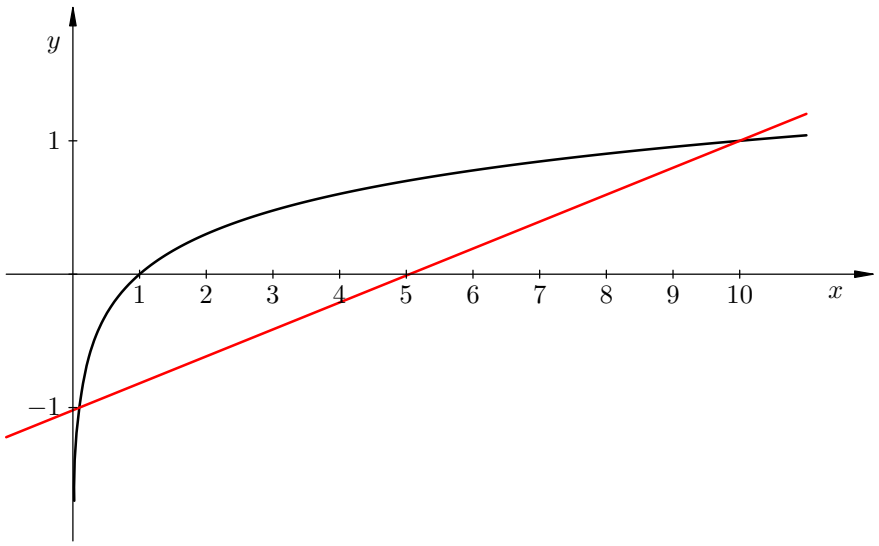
Primjer 7.2.2 *Promotrimo grafove interpolacijskih polinoma stupnjeva 1–6 koji interpoliraju funkciju*

$$f(x) = \log(x) \quad za\ x \in [0.1, 10]$$

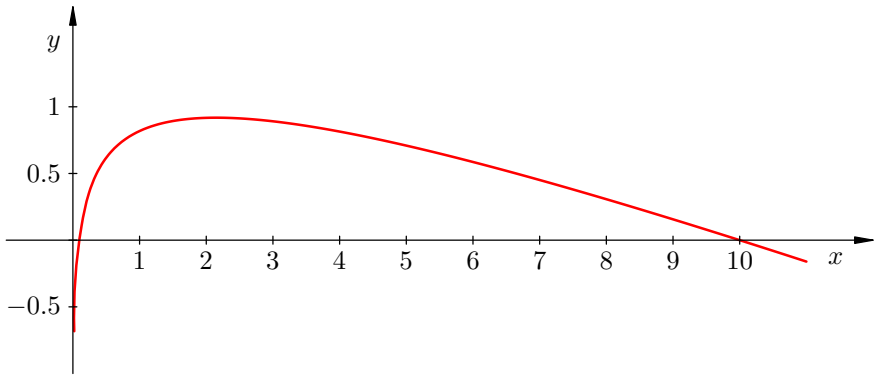
na ekvidistantnoj i Čebiševljevoj mreži.

Primijetit ćete da je greška interpolacije najveća na prvom podintervalu bez obzira na stupanj interpolacijskog polinoma. Razlog leži u činjenici da funkcija $\log(x)$ *ima singularitet u* 0*, a početna točka interpolacije je blizu.*

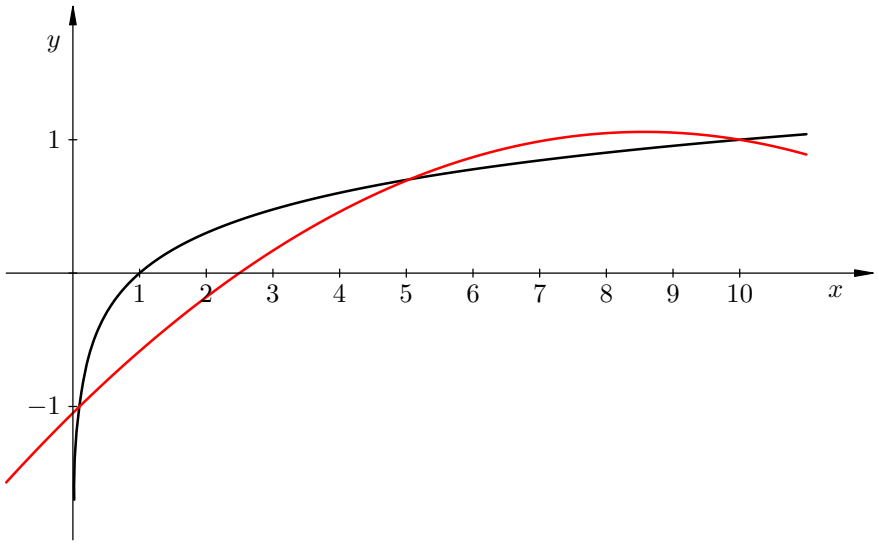
Prva grupa slika su redom funkcija (crno) i interpolacijski polinom (crveno) za ekvidistantnu mrežu, te pripadna greška, a zatim to isto za Čebiševljevu mrežu.



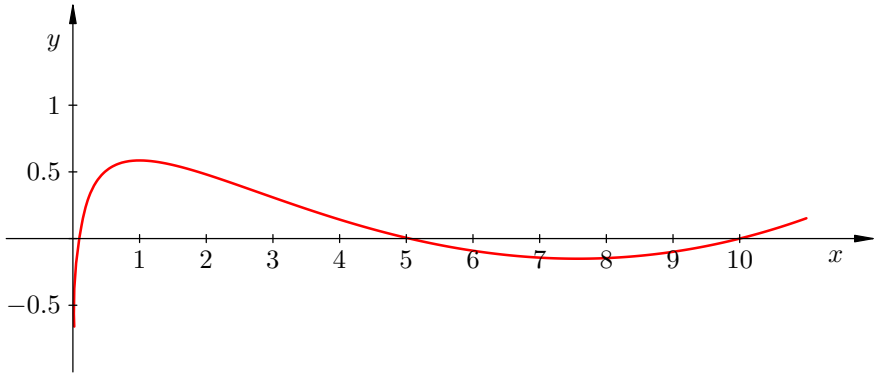
Ekvidistantna mreža, interpolacijski polinom stupnja 1.



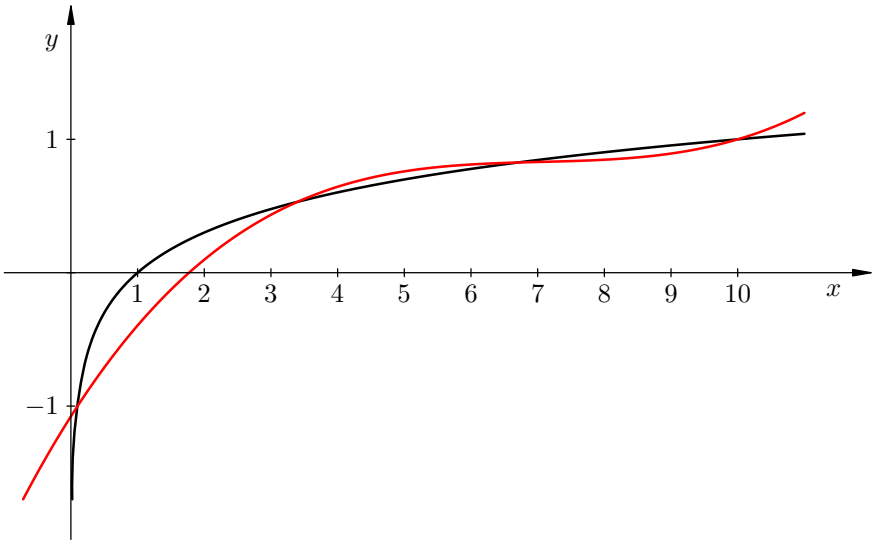
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 1.



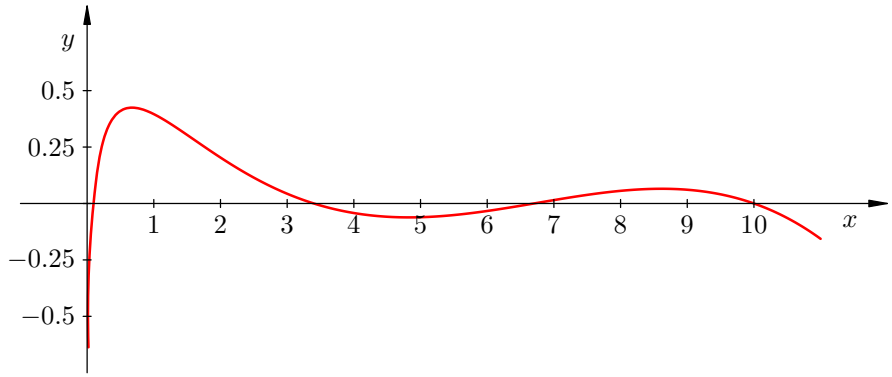
Ekvidistantna mreža, interpolacijski polinom stupnja 2.



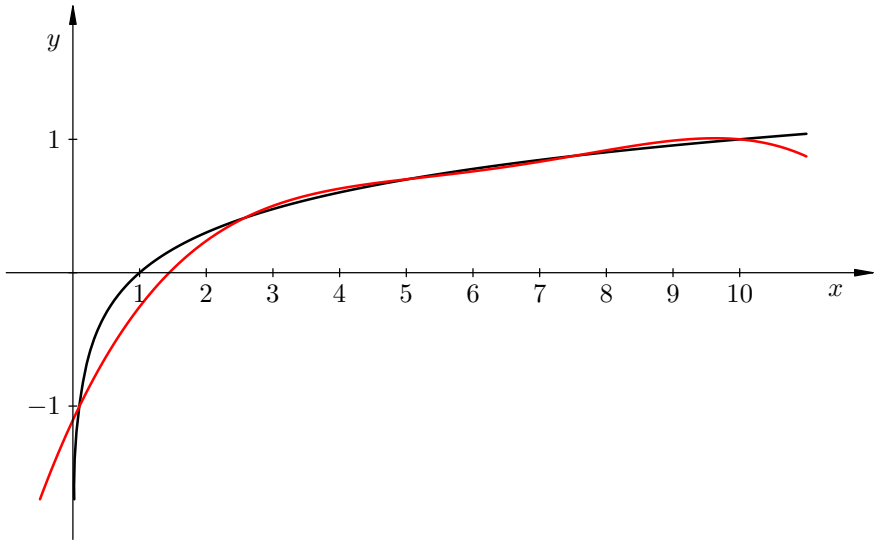
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 2.



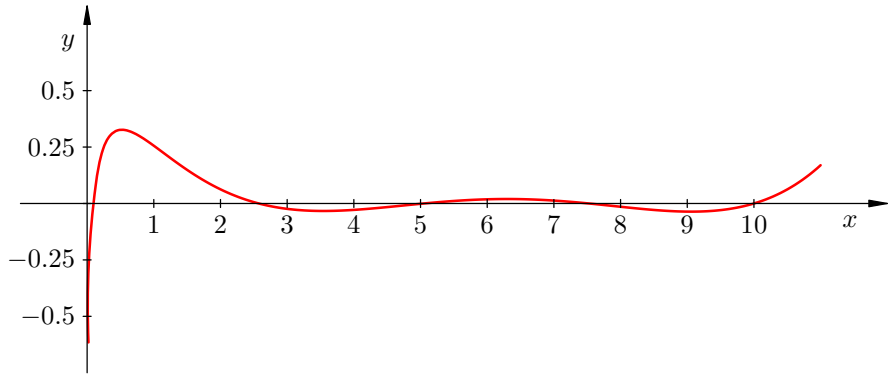
Ekvidistantna mreža, interpolacijski polinom stupnja 3.



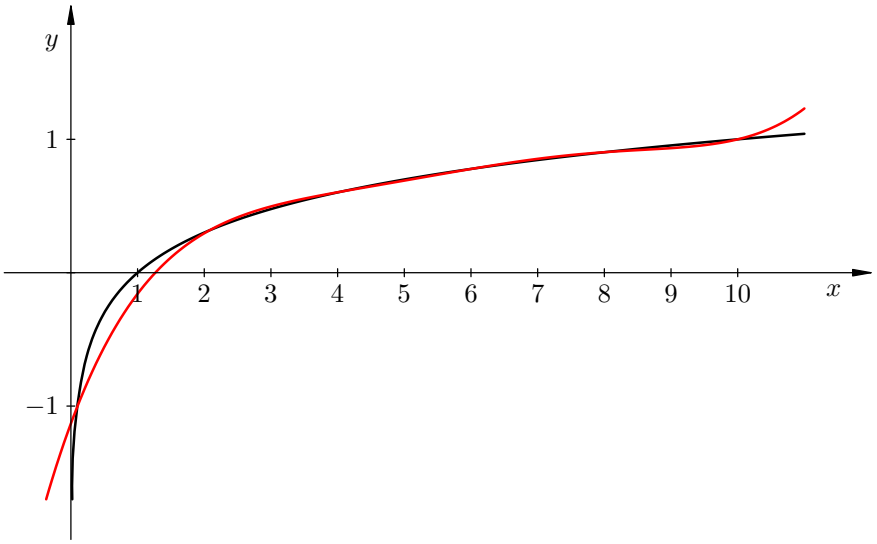
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 3.



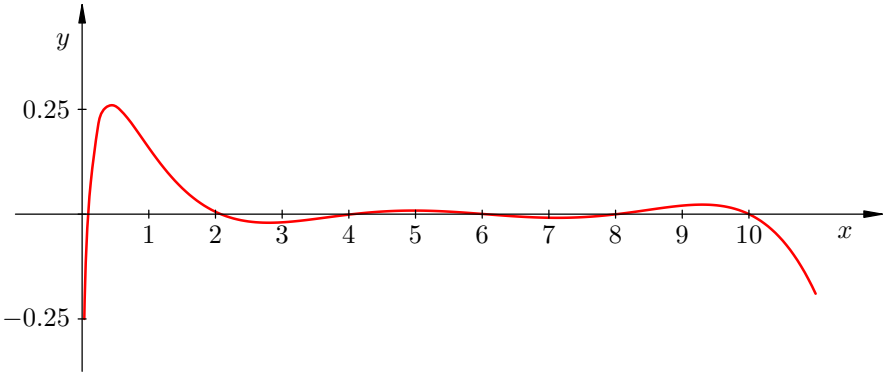
Ekvidistantna mreža, interpolacijski polinom stupnja 4.



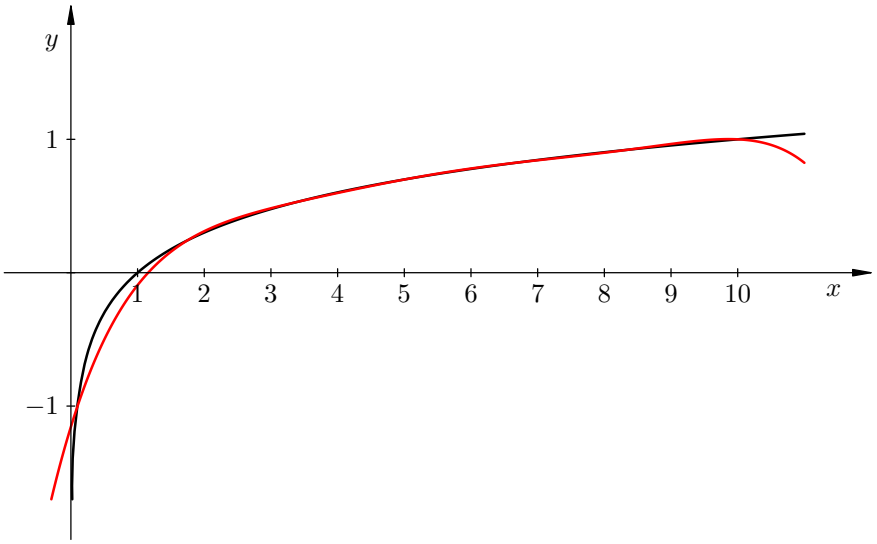
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 4.



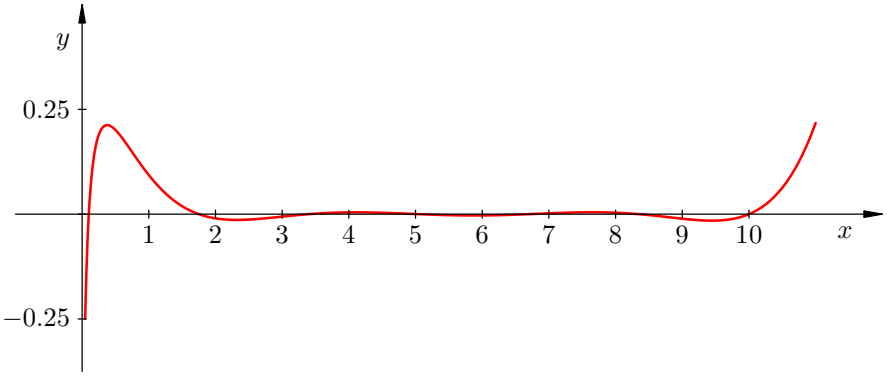
Ekvidistantna mreža, interpolacijski polinom stupnja 5.



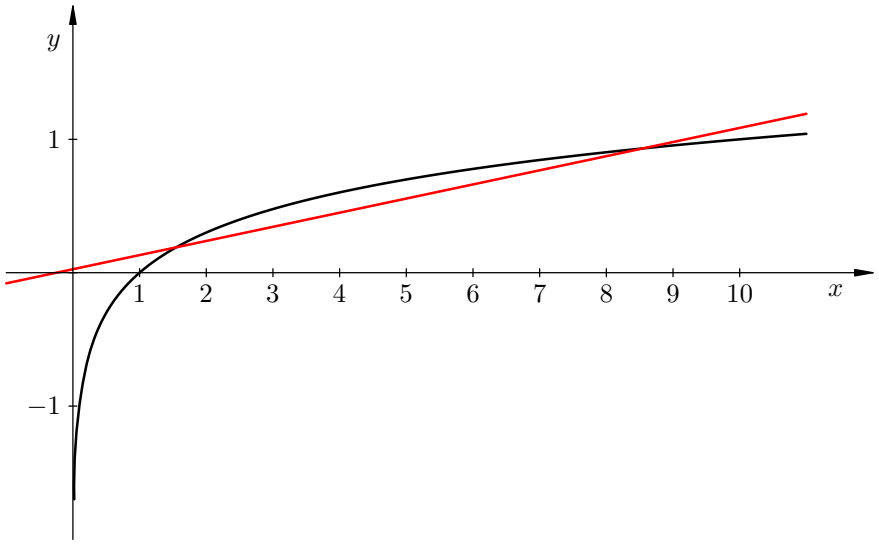
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 5.



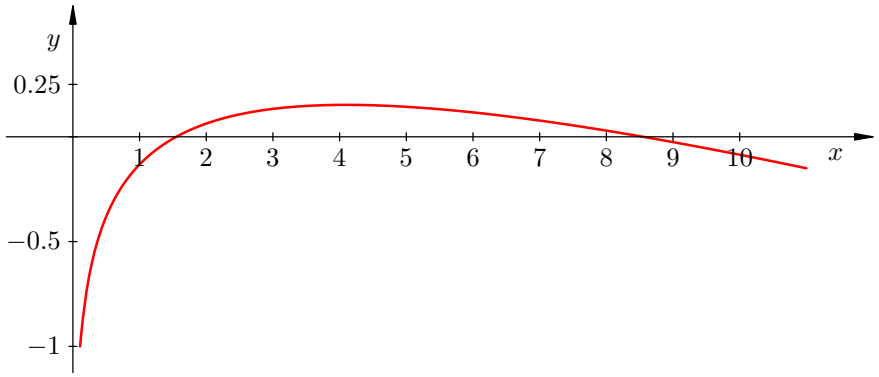
Ekvidistantna mreža, interpolacijski polinom stupnja 6.



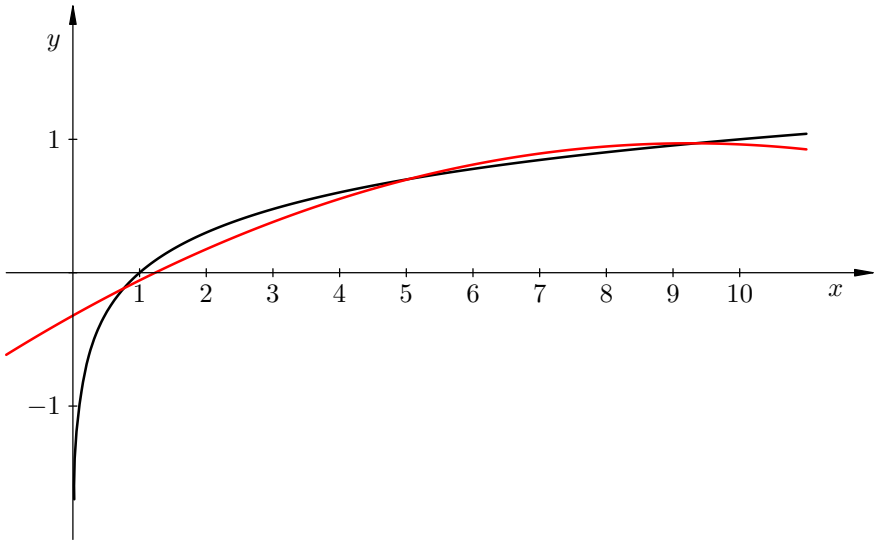
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 6.



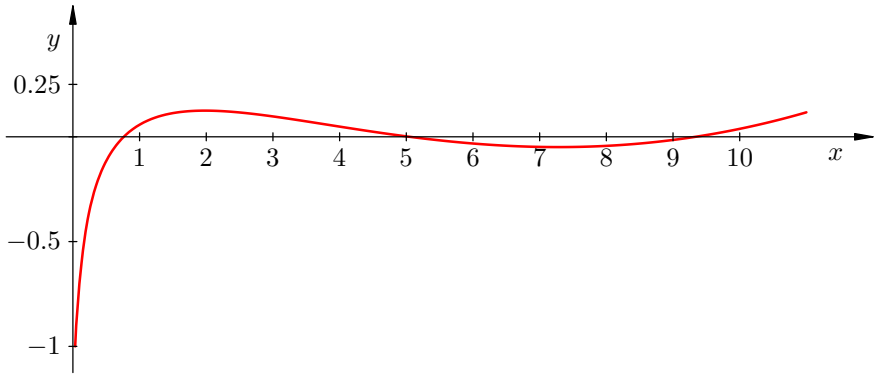
Čebiševljeva mreža, interpolacijski polinom stupnja 1.



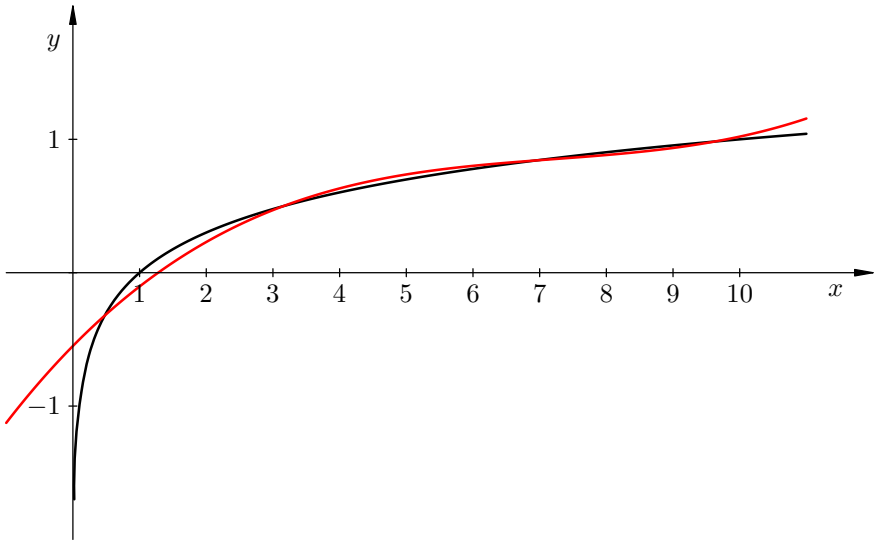
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 1.



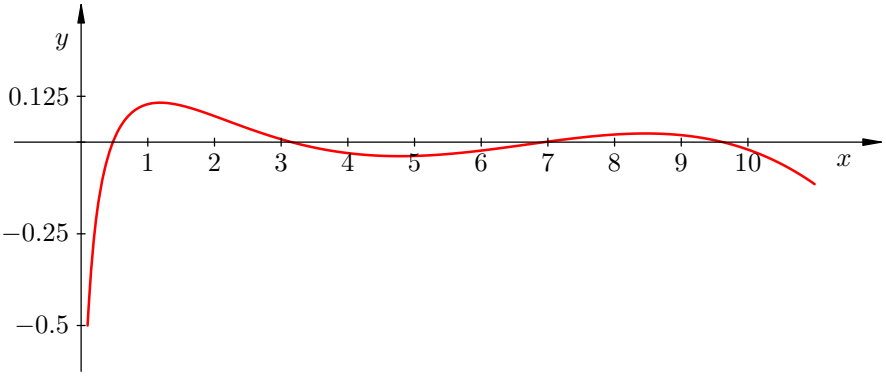
Čebiševljeva mreža, interpolacijski polinom stupnja 2.



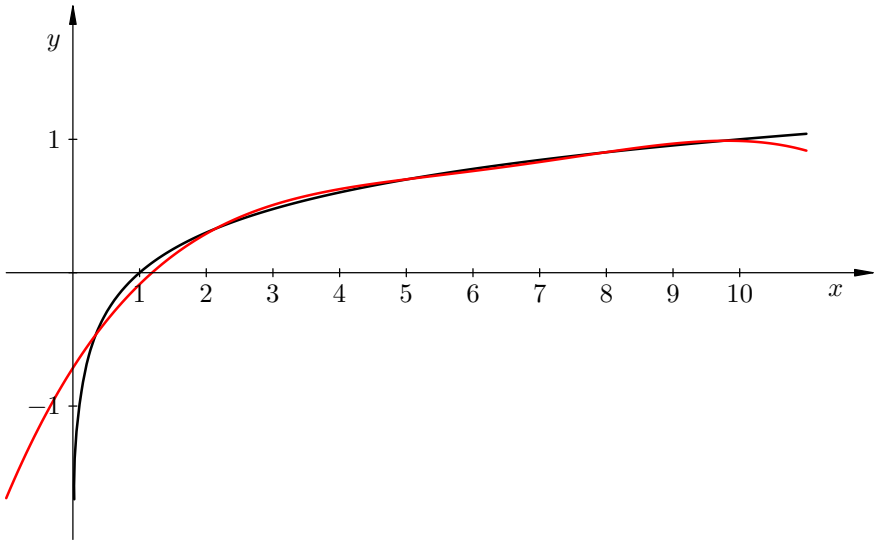
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 2.



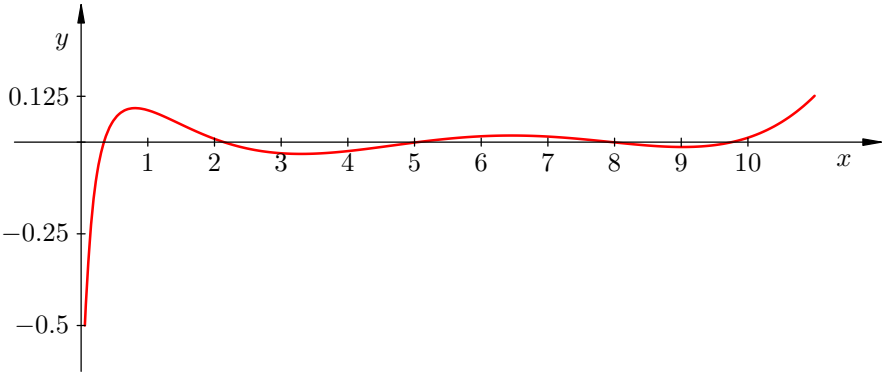
Čebiševljeva mreža, interpolacijski polinom stupnja 3.



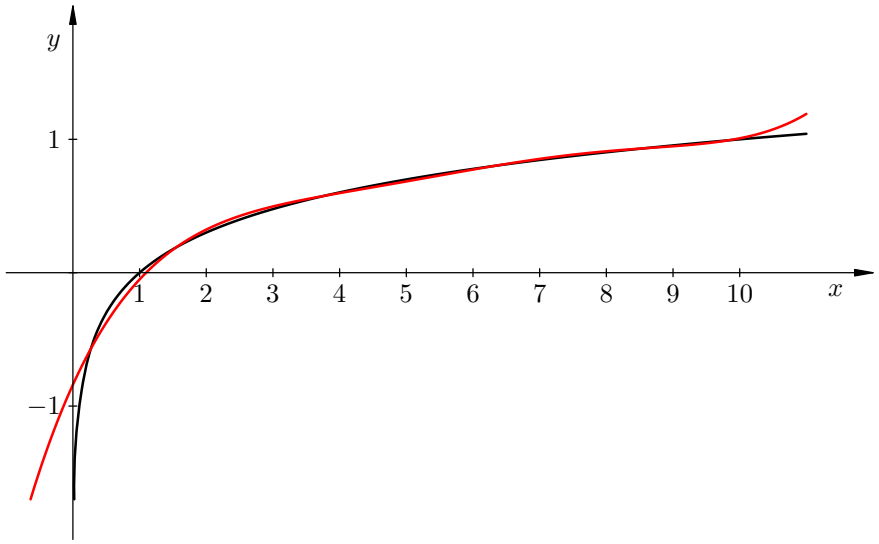
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 3.



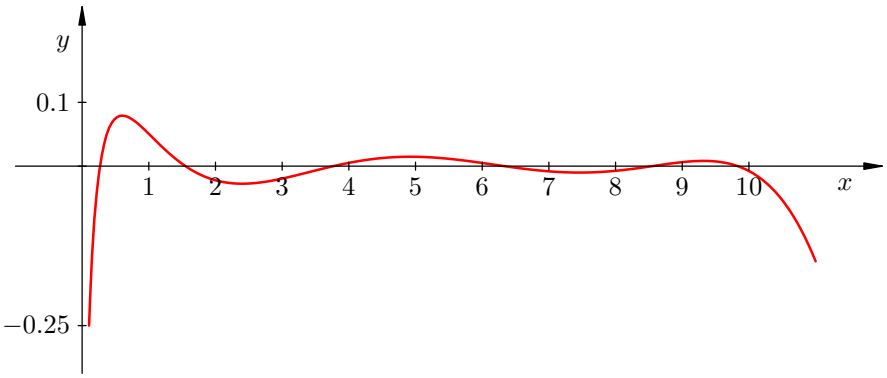
Čebiševljeva mreža, interpolacijski polinom stupnja 4.



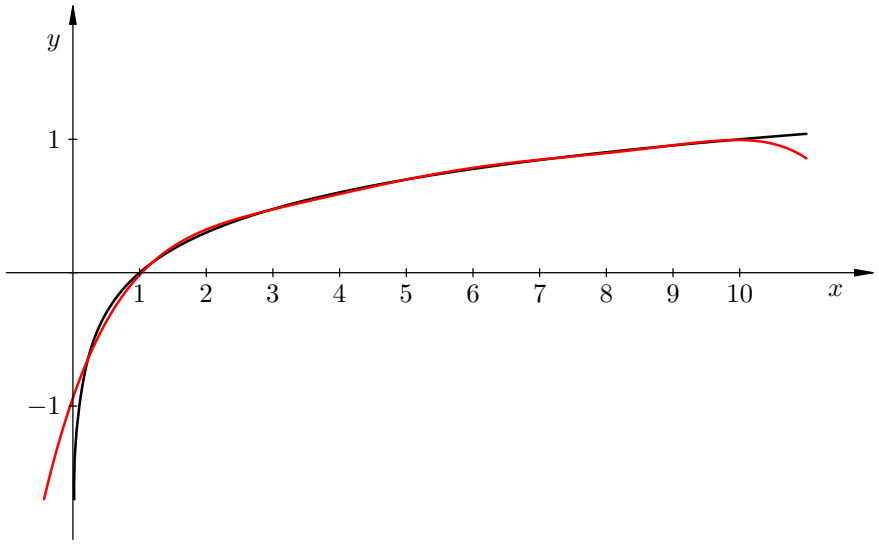
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 4.



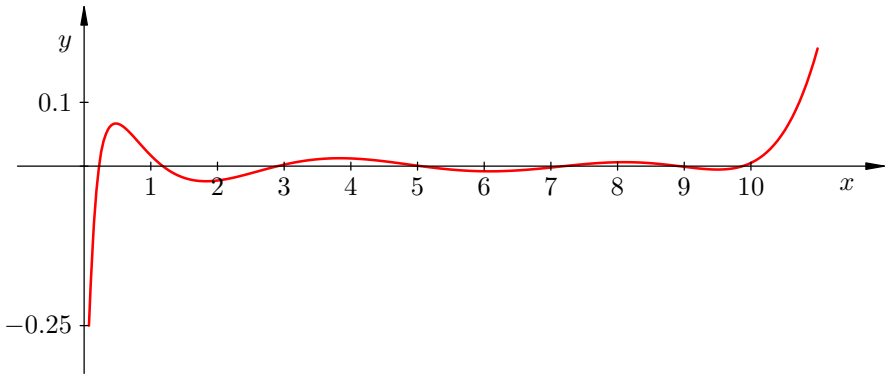
Čebiševljeva mreža, interpolacijski polinom stupnja 5.



Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 5.



Čebiševljeva mreža, interpolacijski polinom stupnja 6.



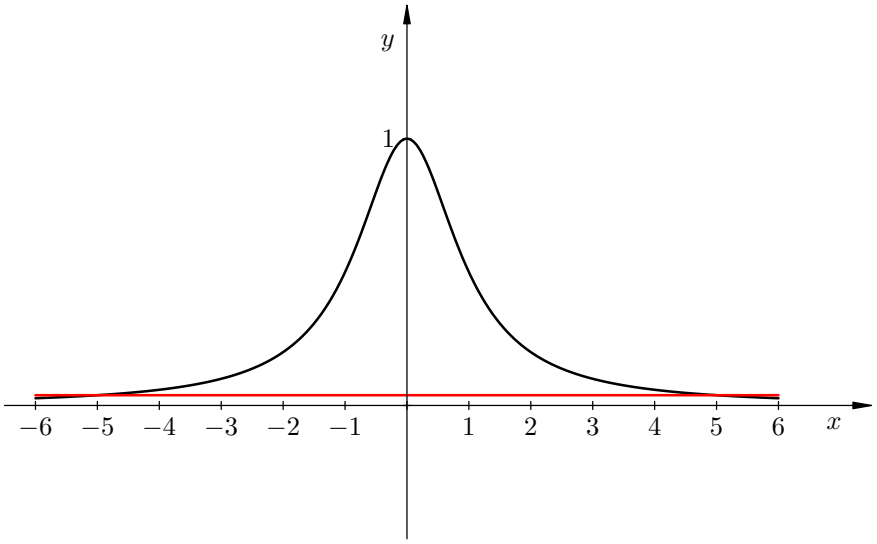
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 6.

Primjer 7.2.3 *Već smo pokazali da na primjeru Runge interpolacijski polinomi koji interpoliraju funkciju*

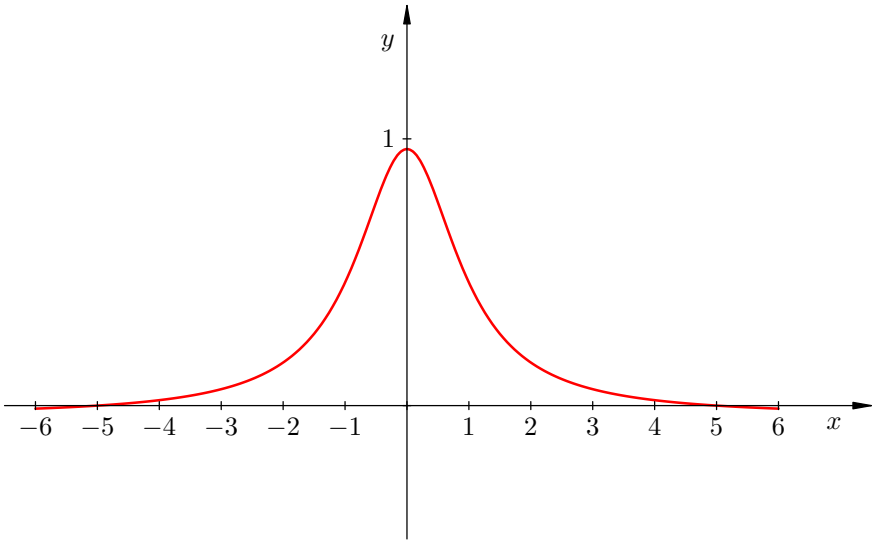
$$f(x) = \frac{1}{1+x^2} \quad za\ x \in [-5, 5]$$

na ekvidistantnoj mreži ne konvergiraju. S druge strane, pogledajmo što se događa s polinomima koji interpoliraju tu funkciju u Čebiševljevim točkama. Interpolacijski polinomi su stupnjeva 1–6, 8, 10, 12, 14, 16 (parnost funkcije!).

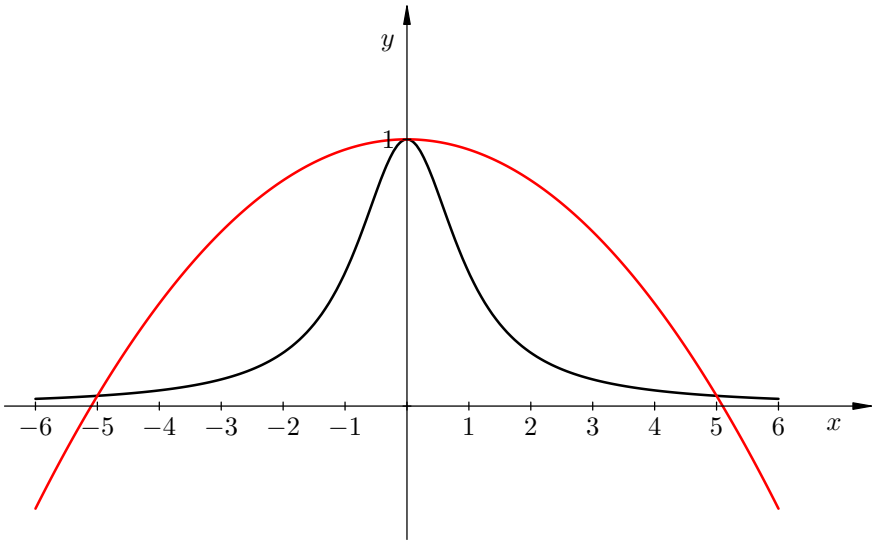
Ponovno, kao i u prošlom primjeru, grafovi su u parovima.



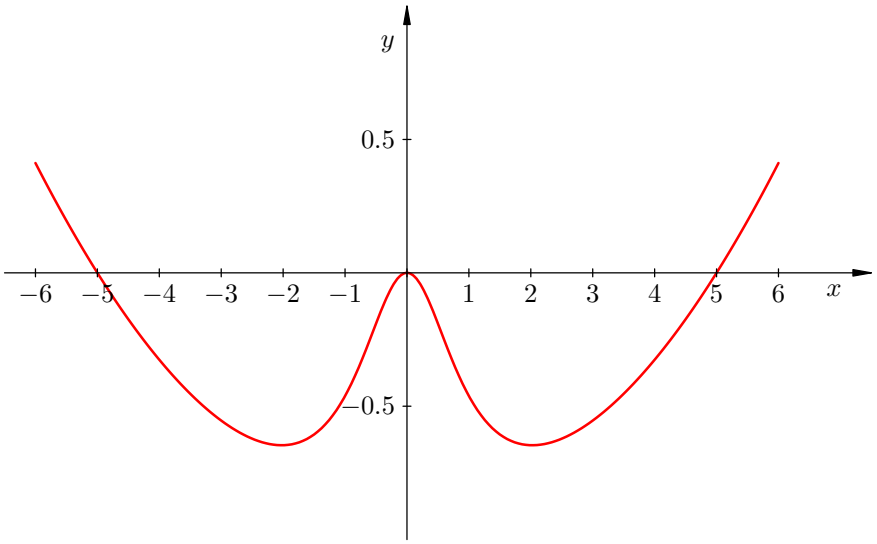
Ekvidistantna mreža, interpolacijski polinom stupnja 1.



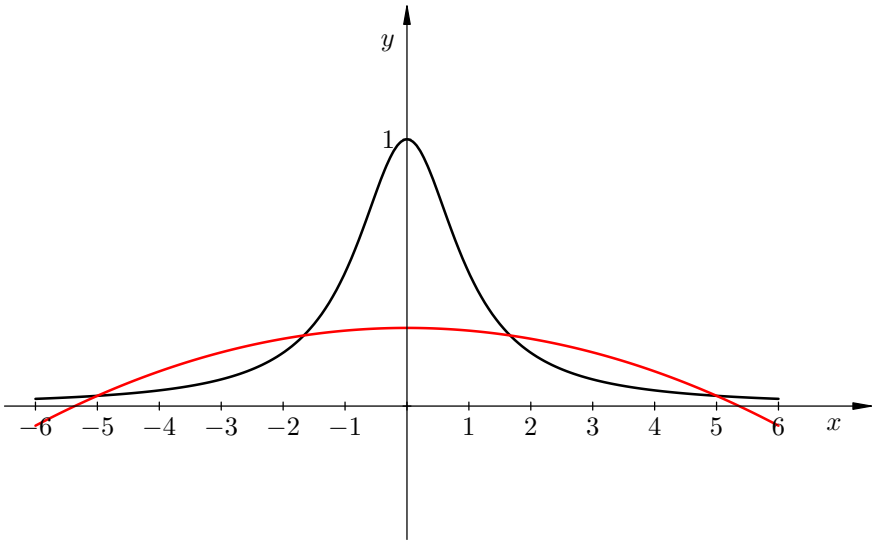
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 1.



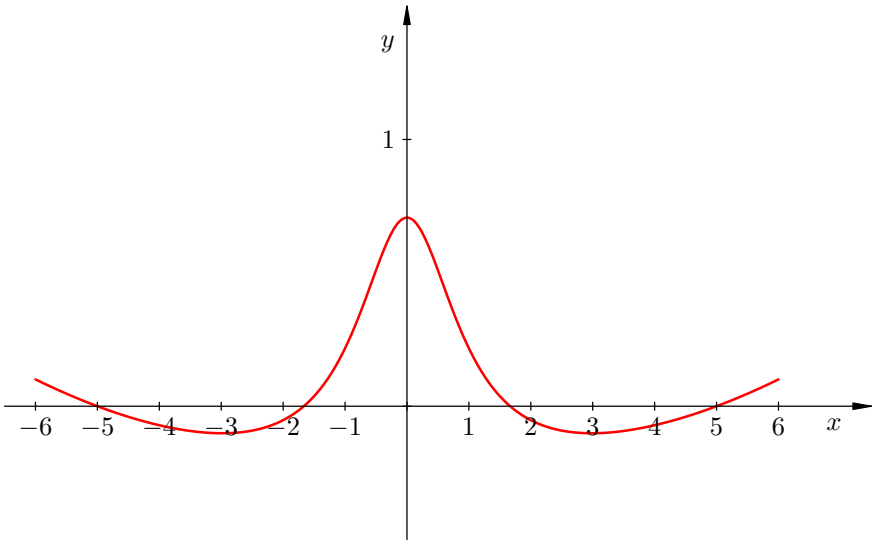
Ekvidistantna mreža, interpolacijski polinom stupnja 2.



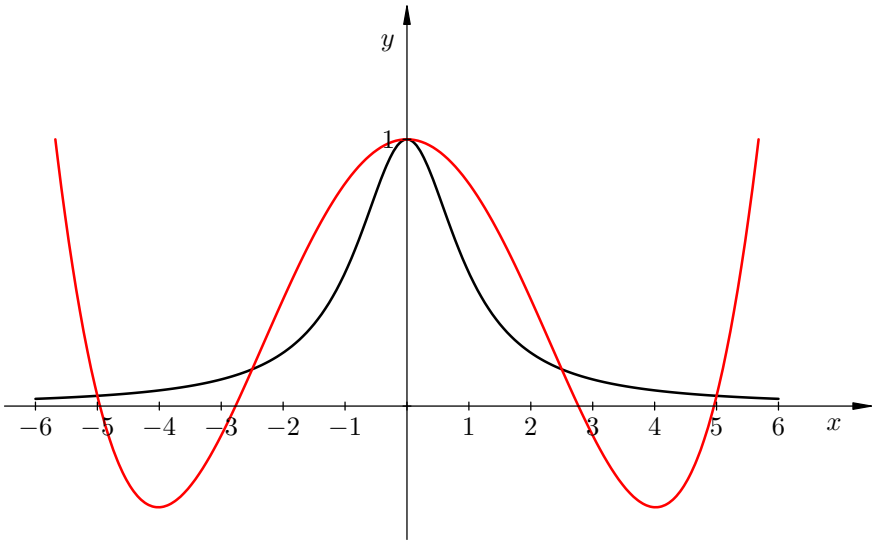
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 2.



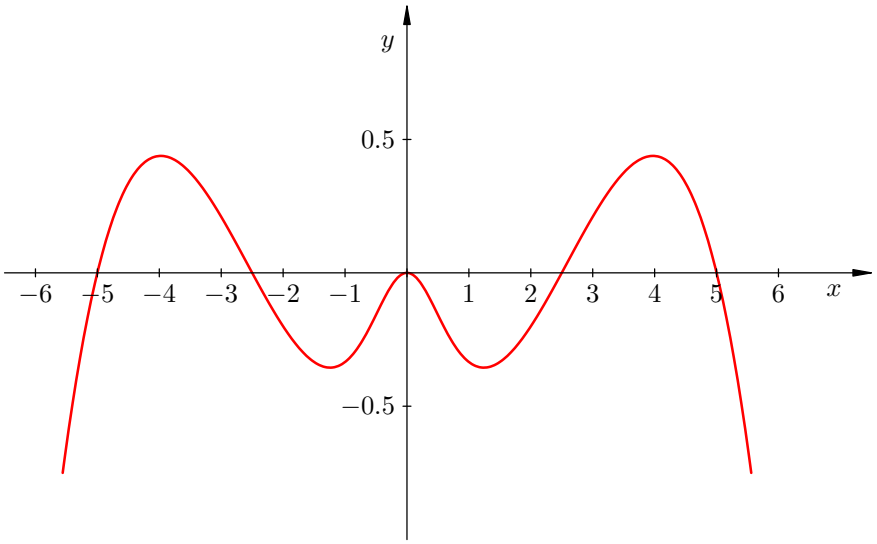
Ekvidistantna mreža, interpolacijski polinom stupnja 3.



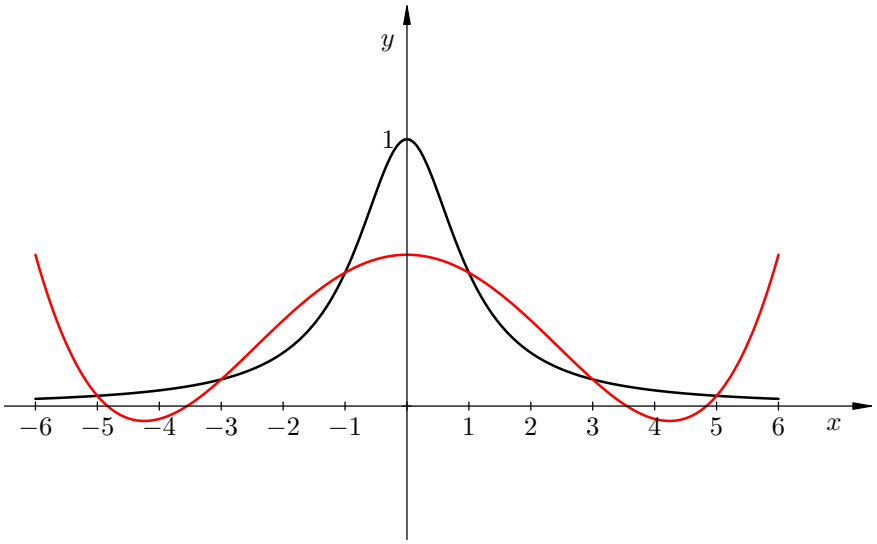
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 3.



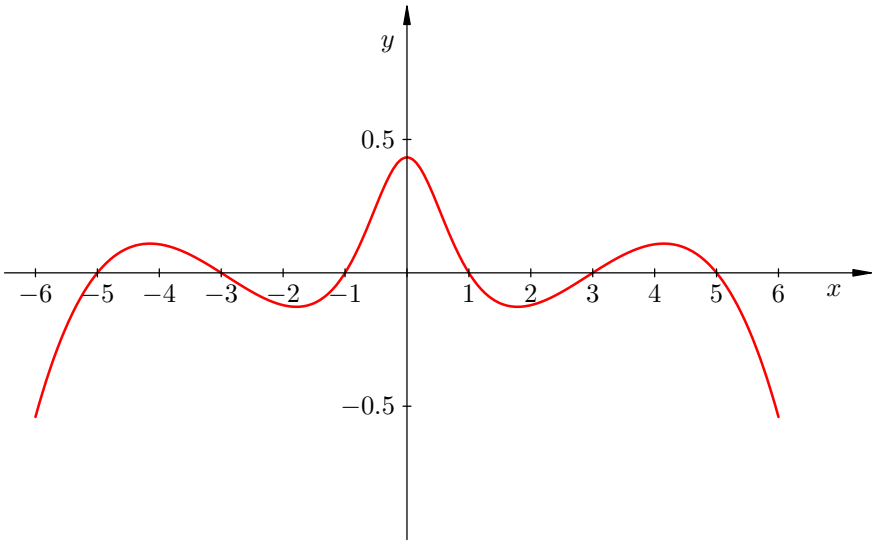
Ekvidistantna mreža, interpolacijski polinom stupnja 4.



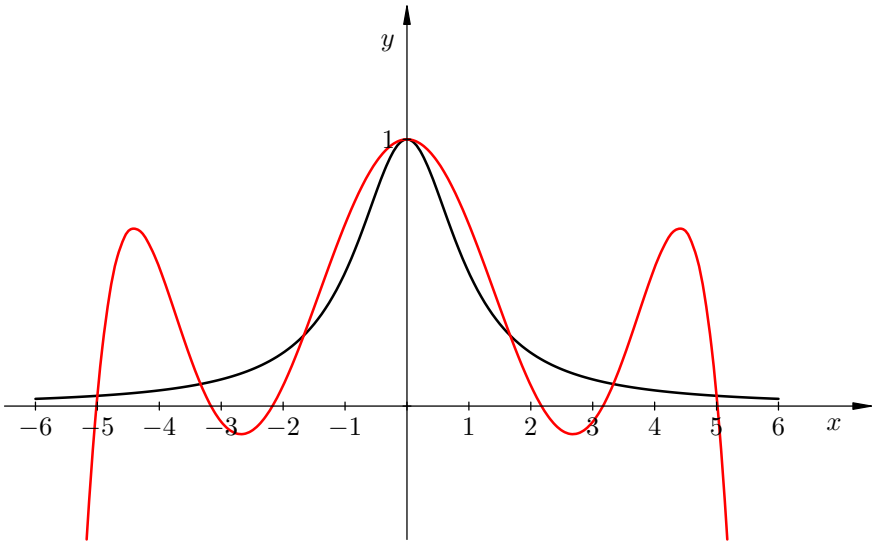
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 4.



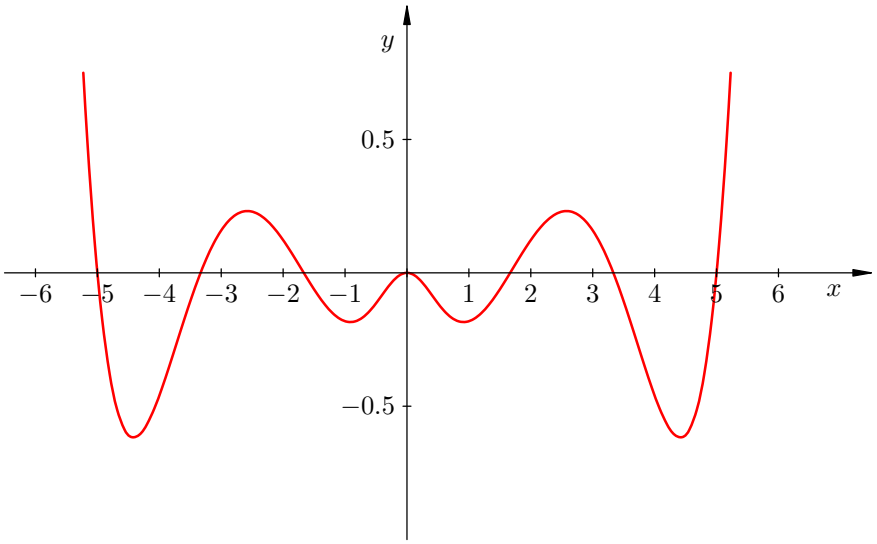
Ekvidistantna mreža, interpolacijski polinom stupnja 5.



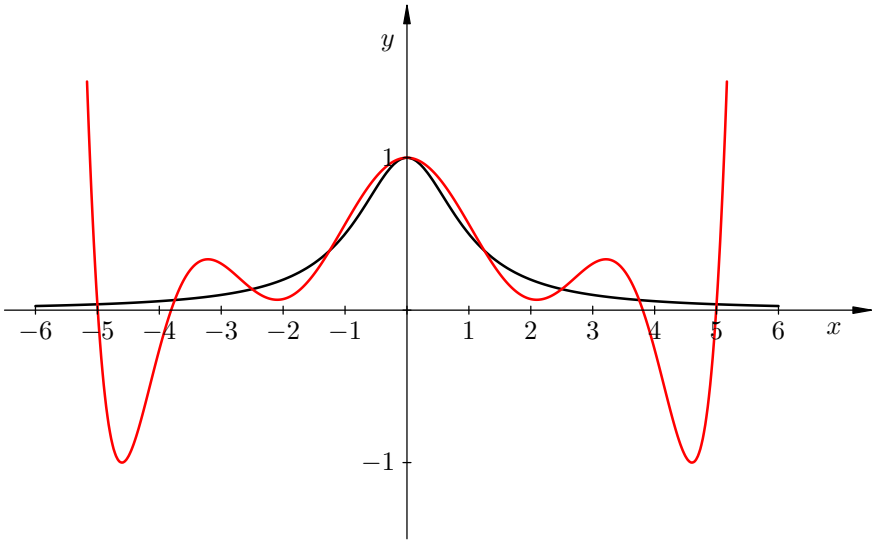
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 5.



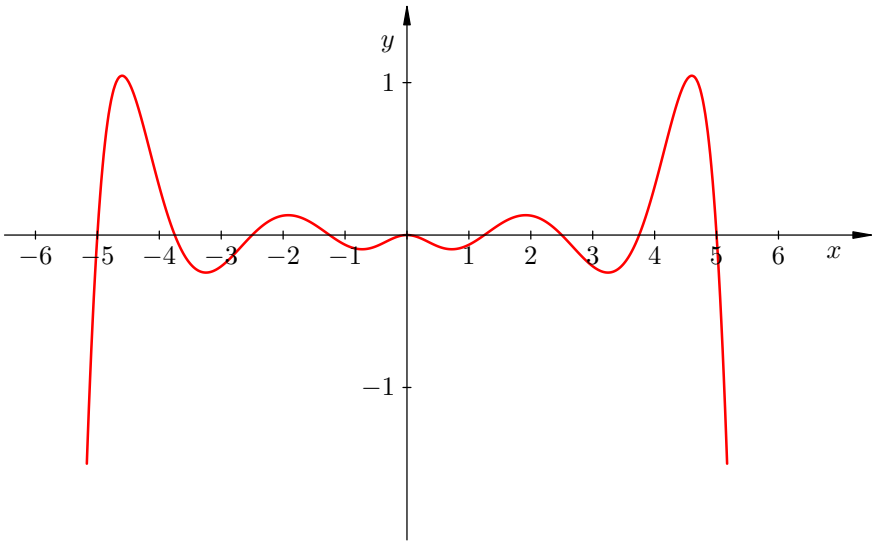
Ekvidistantna mreža, interpolacijski polinom stupnja 6.



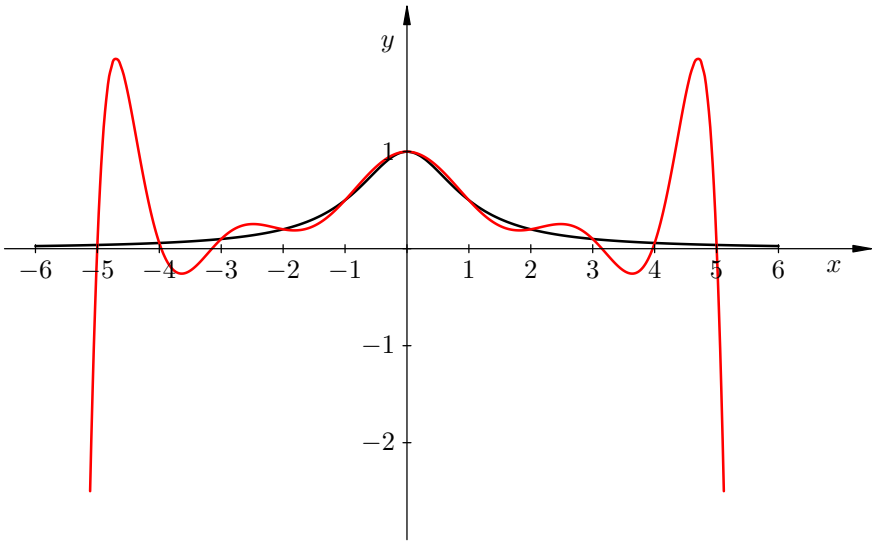
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 6.



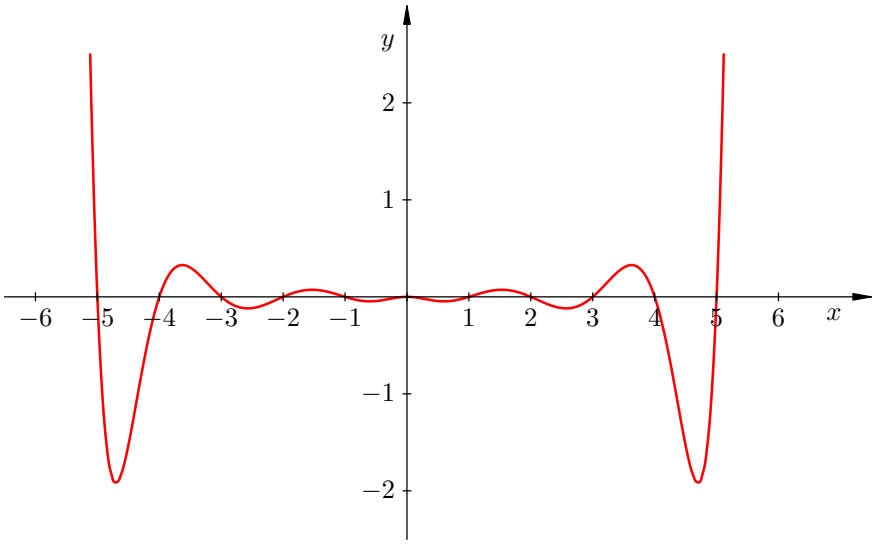
Ekvidistantna mreža, interpolacijski polinom stupnja 8.



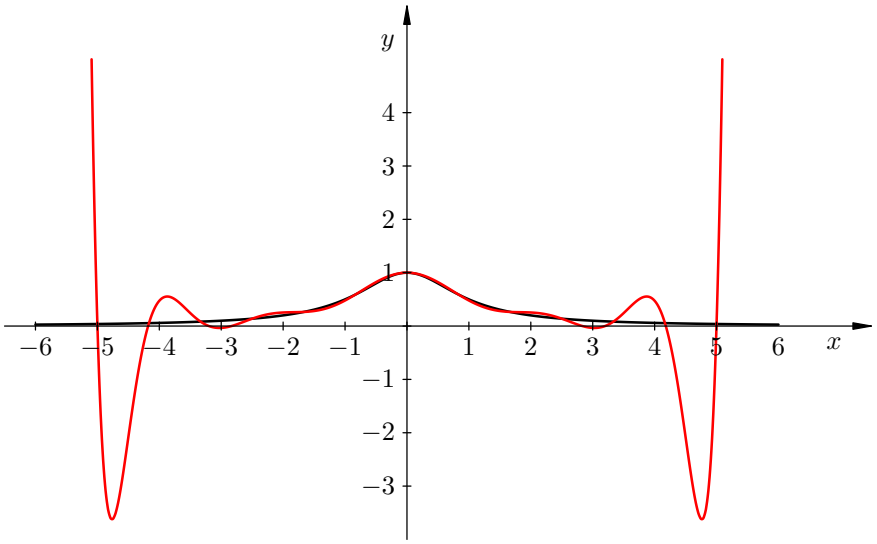
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 8.



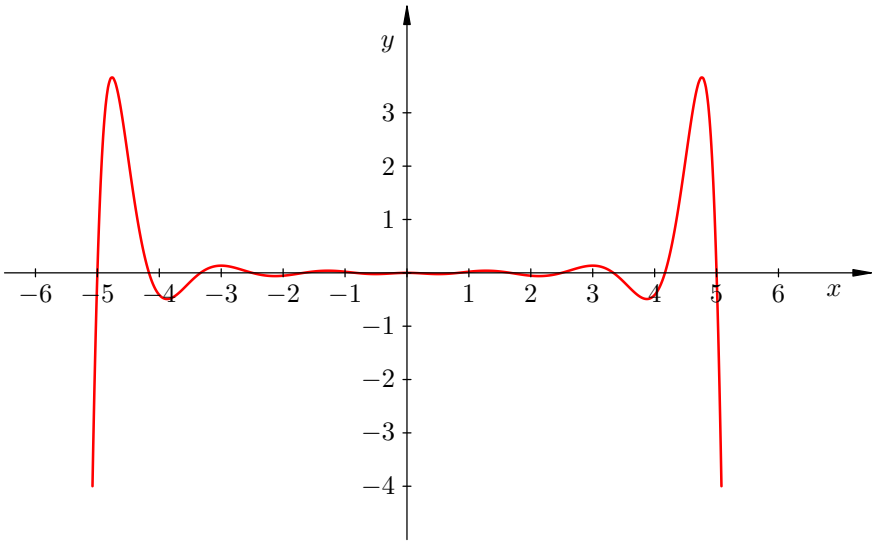
Ekvidistantna mreža, interpolacijski polinom stupnja 10.



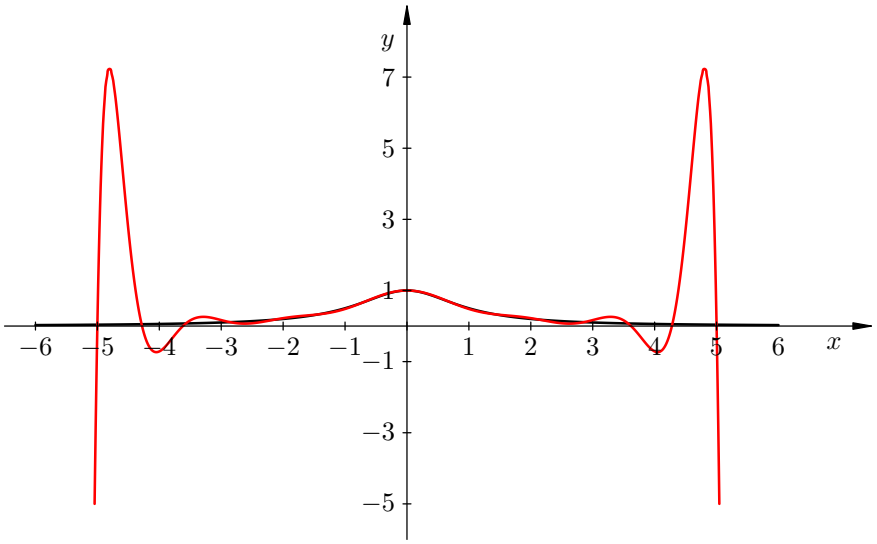
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 10.



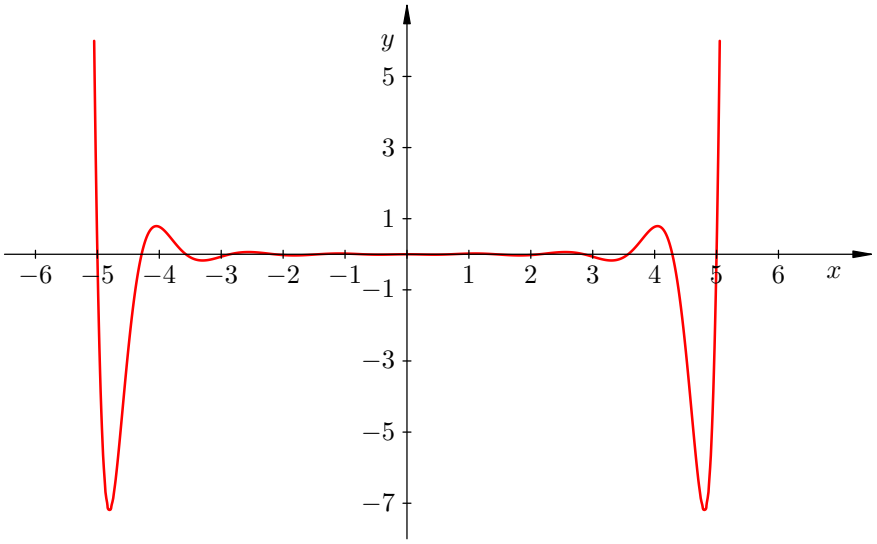
Ekvidistantna mreža, interpolacijski polinom stupnja 12.



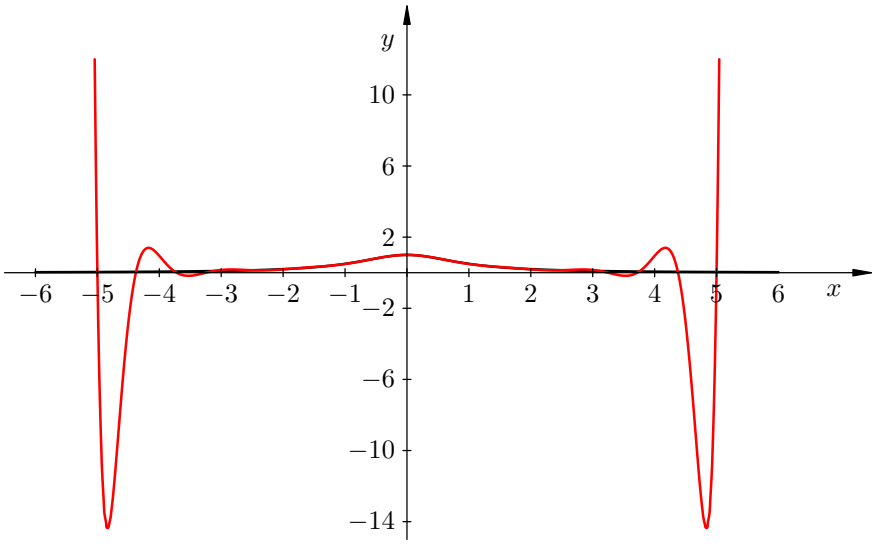
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 12.



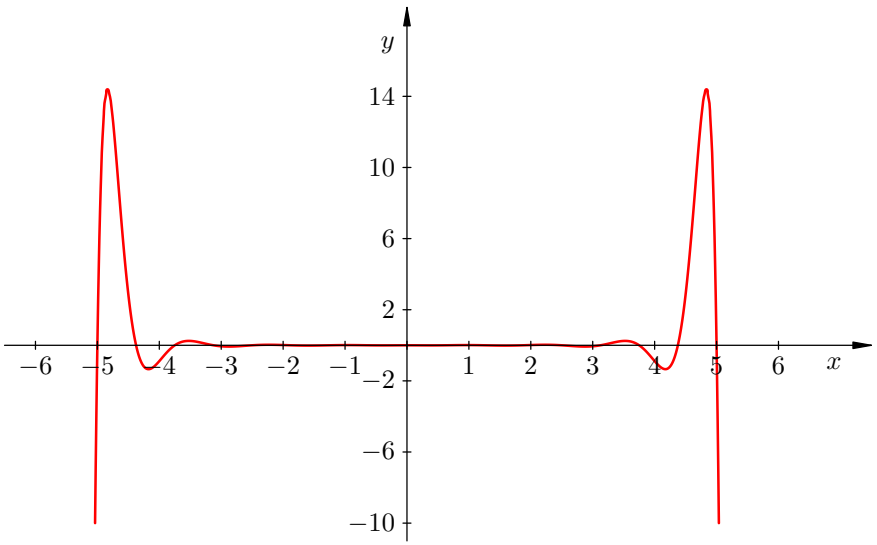
Ekvidistantna mreža, interpolacijski polinom stupnja 14.



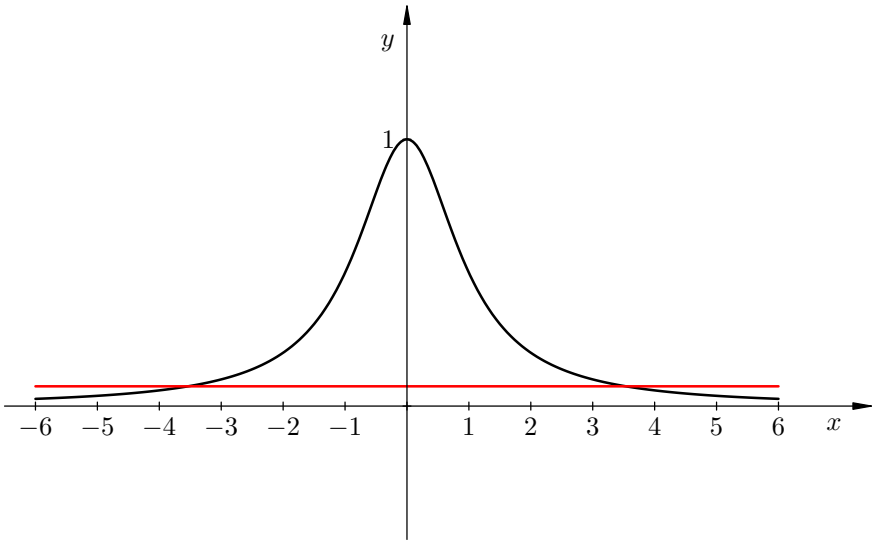
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 14.



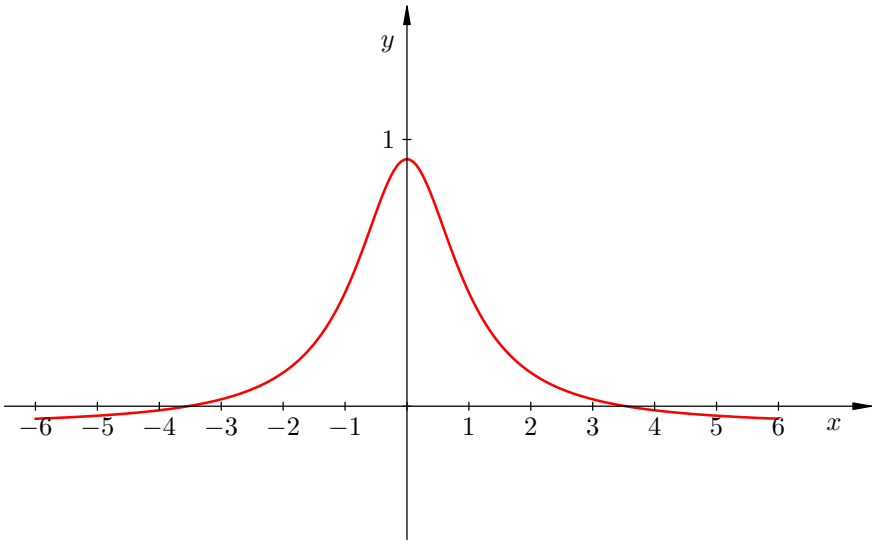
Ekvidistantna mreža, interpolacijski polinom stupnja 16.



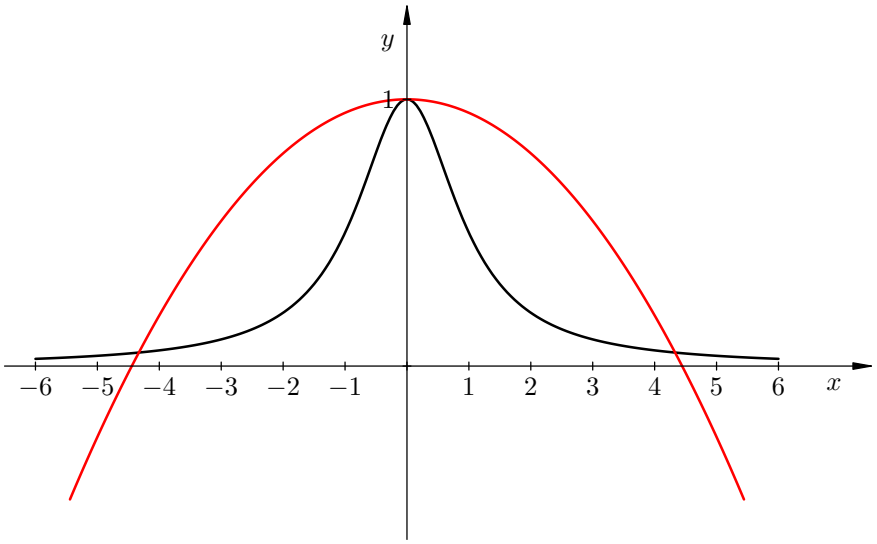
Ekvidistantna mreža, greška interpolacijskog polinoma stupnja 16.



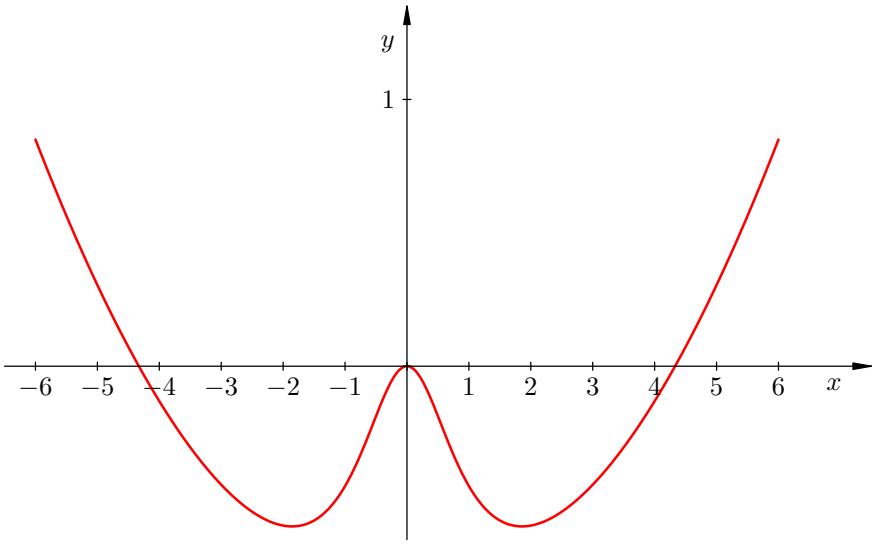
Čebiševljeva mreža, interpolacijski polinom stupnja 1.



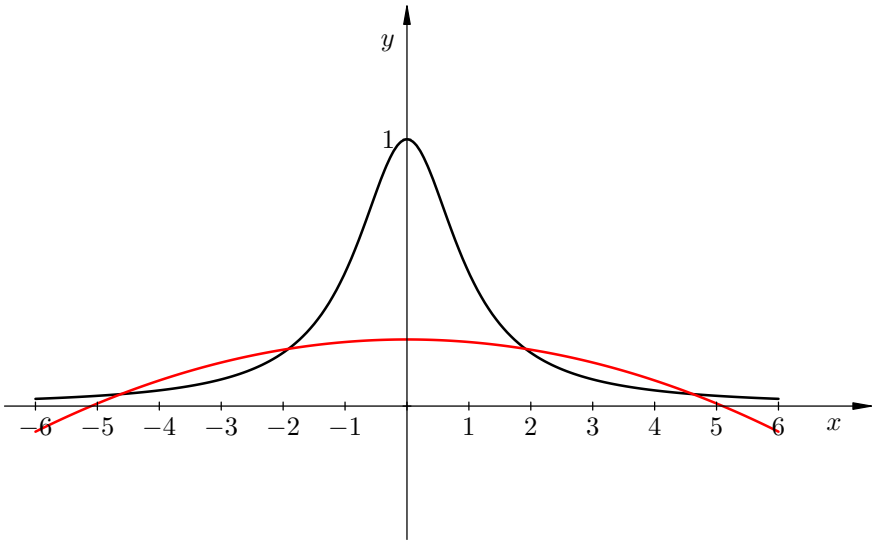
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 1.



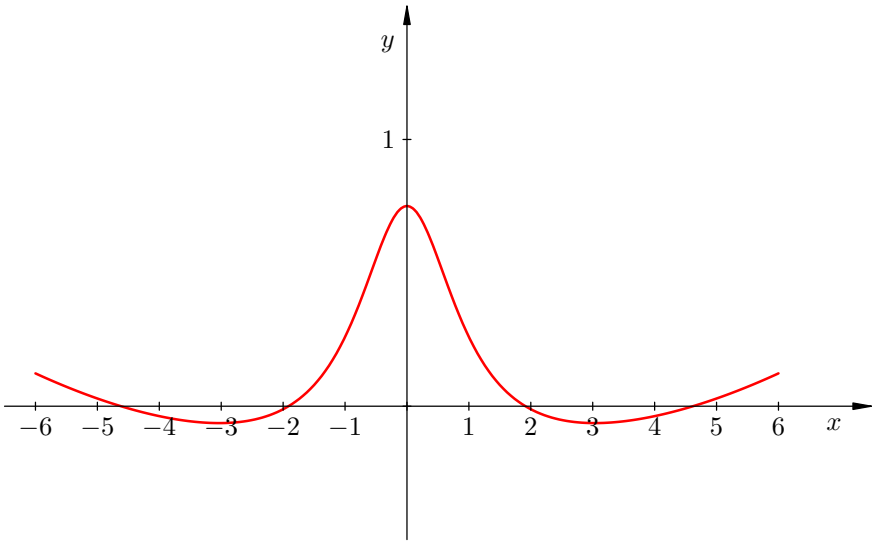
Čebiševljeva mreža, interpolacijski polinom stupnja 2.



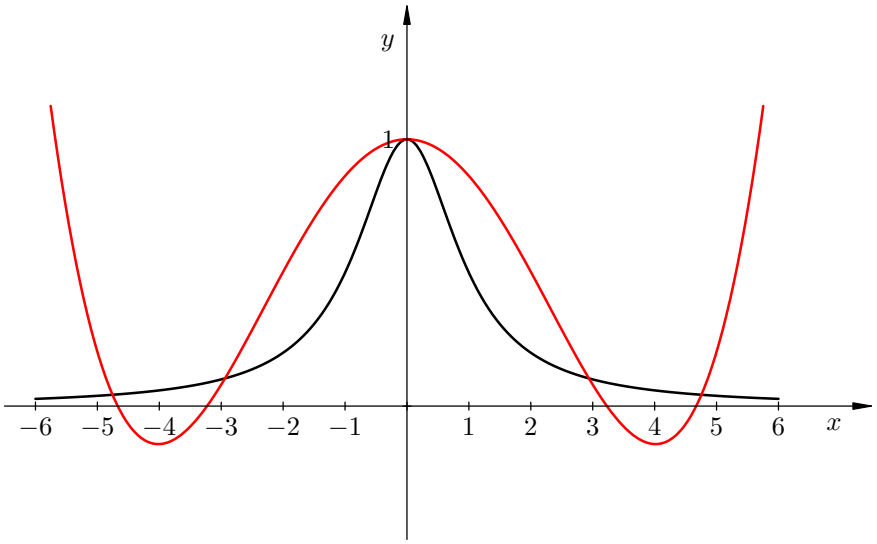
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 2.



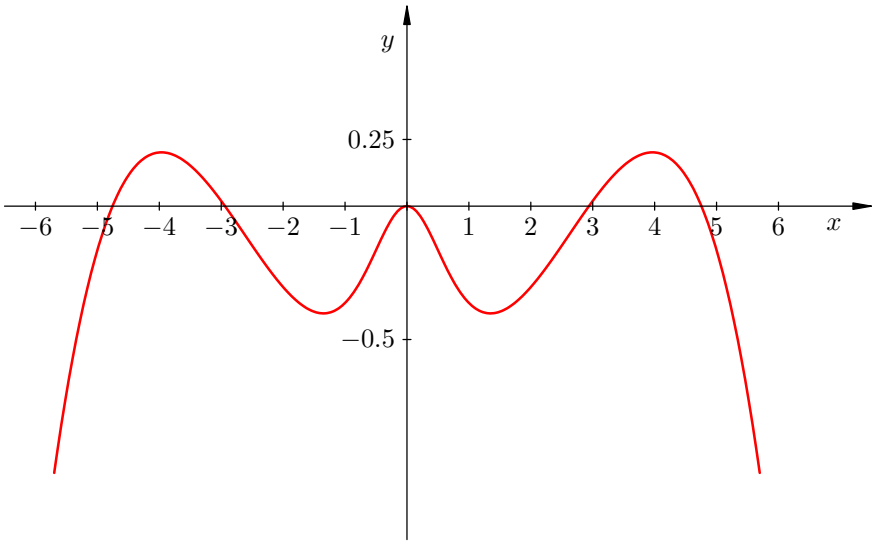
Čebiševljeva mreža, interpolacijski polinom stupnja 3.



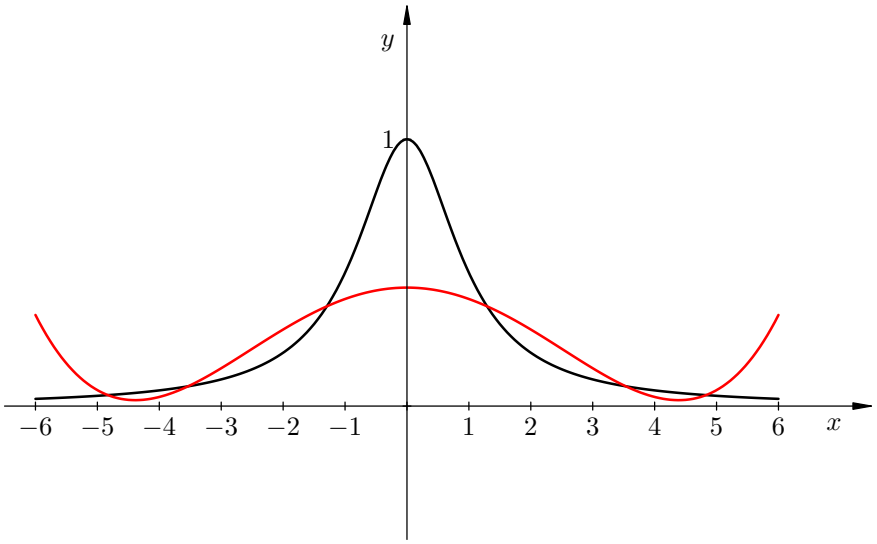
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 3.



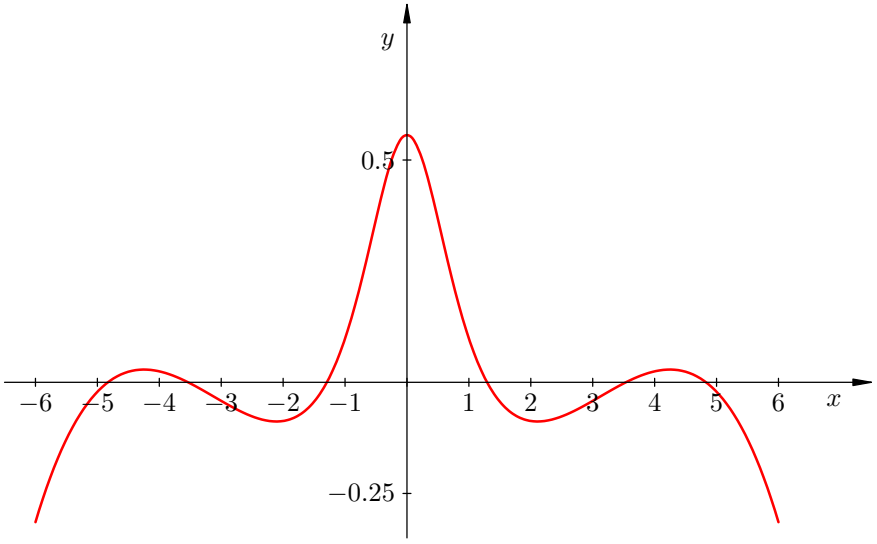
Čebiševljeva mreža, interpolacijski polinom stupnja 4.



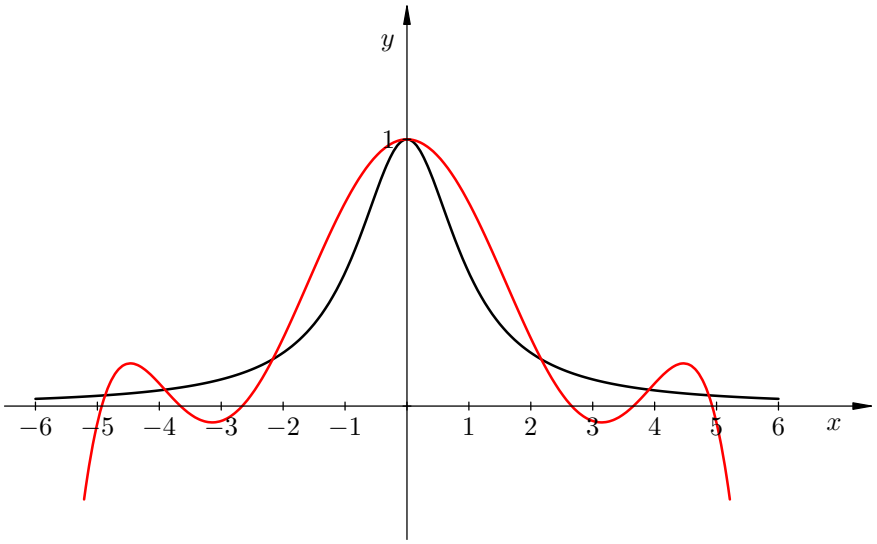
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 4.



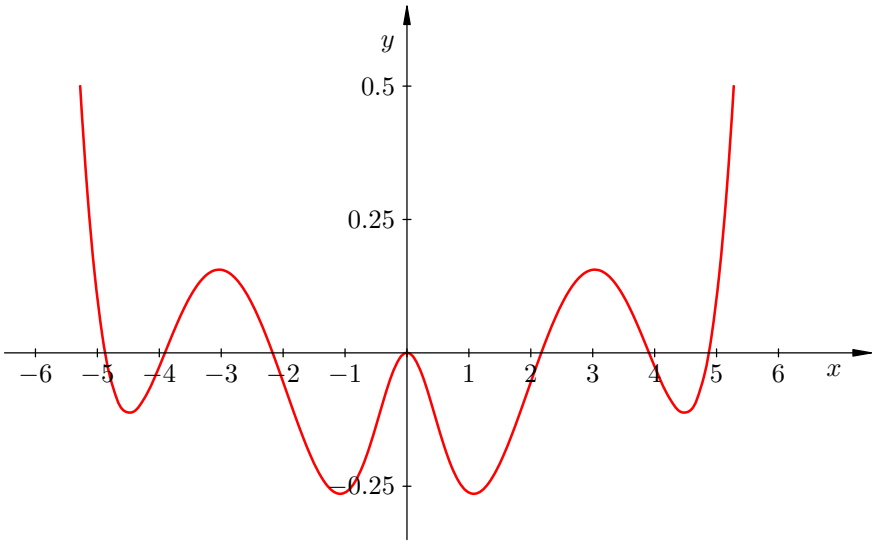
Čebiševljeva mreža, interpolacijski polinom stupnja 5.



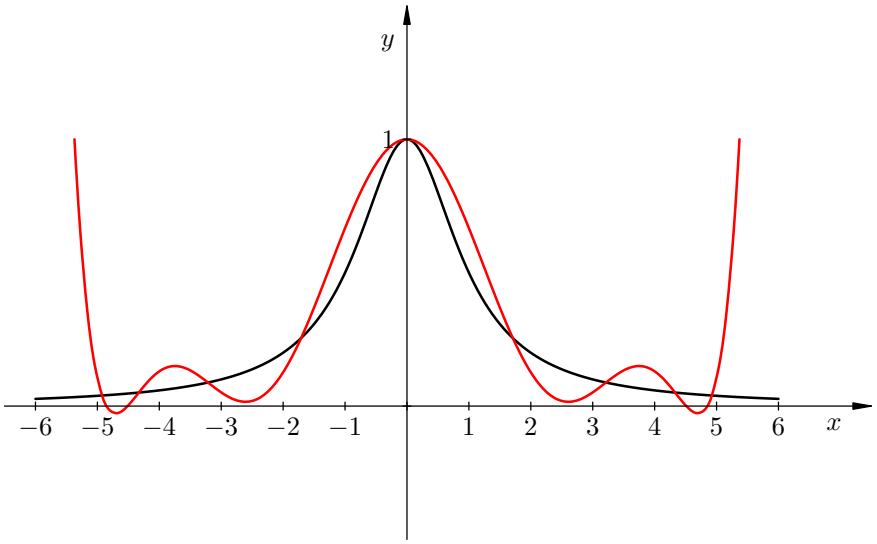
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 5.



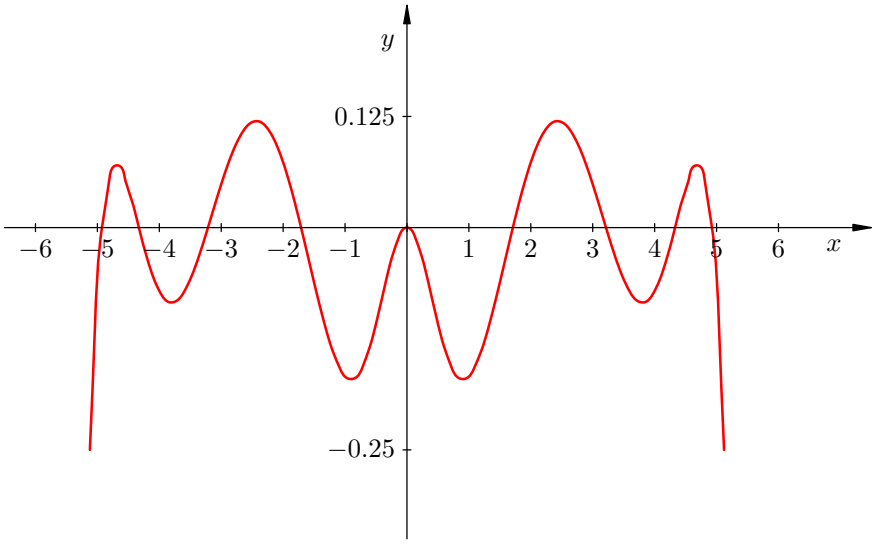
Čebiševljeva mreža, interpolacijski polinom stupnja 6.



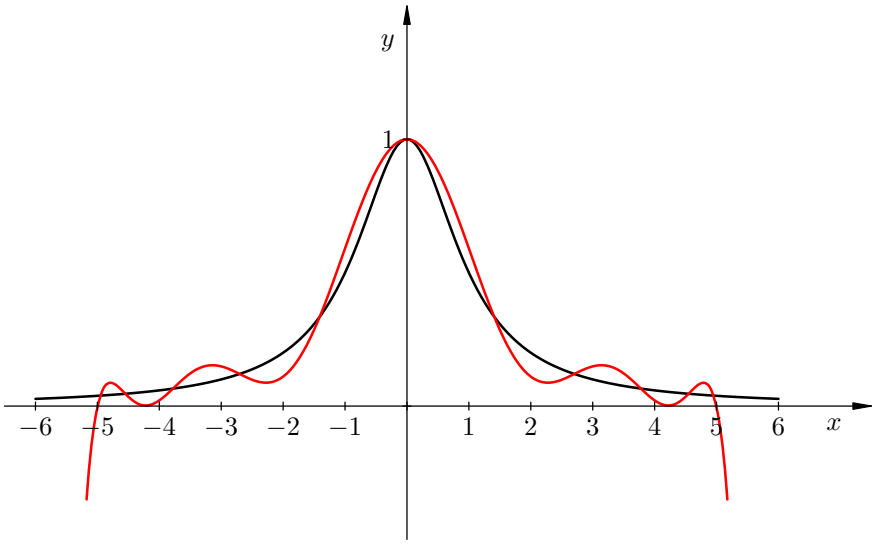
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 6.



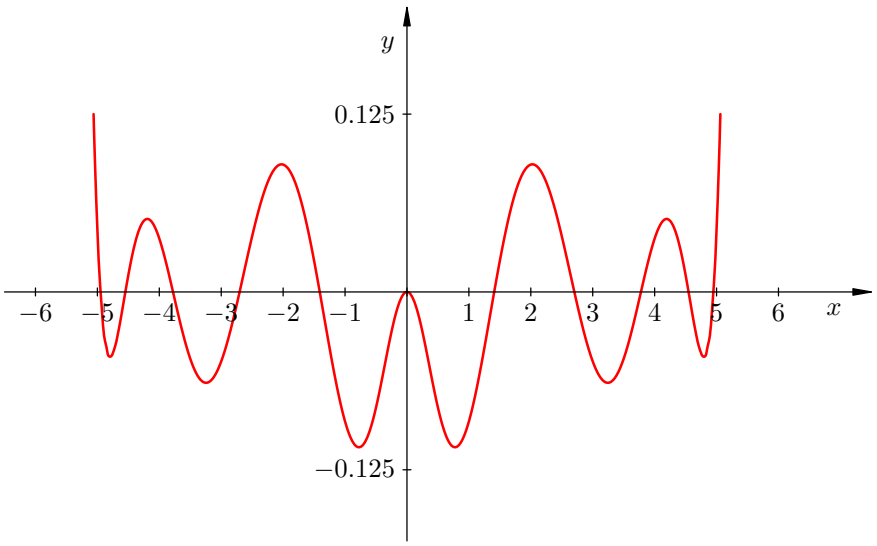
Čebiševljeva mreža, interpolacijski polinom stupnja 8.



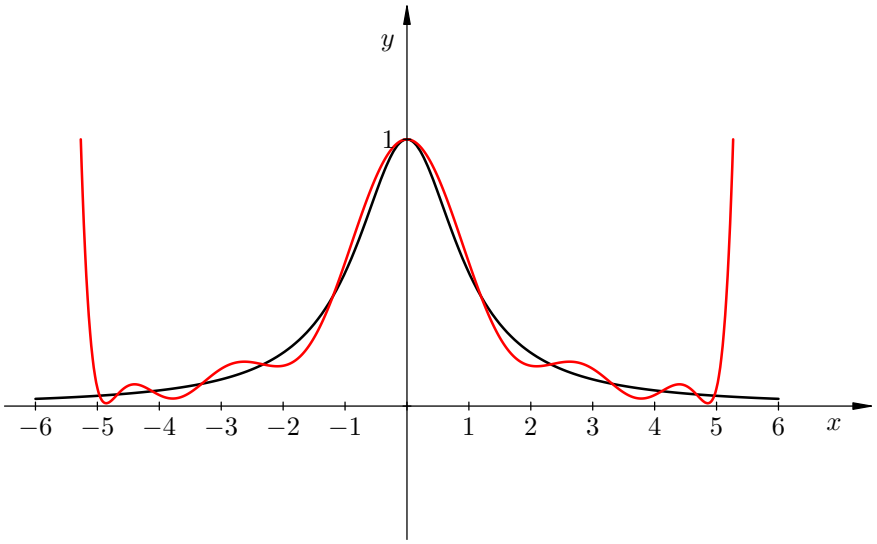
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 8.



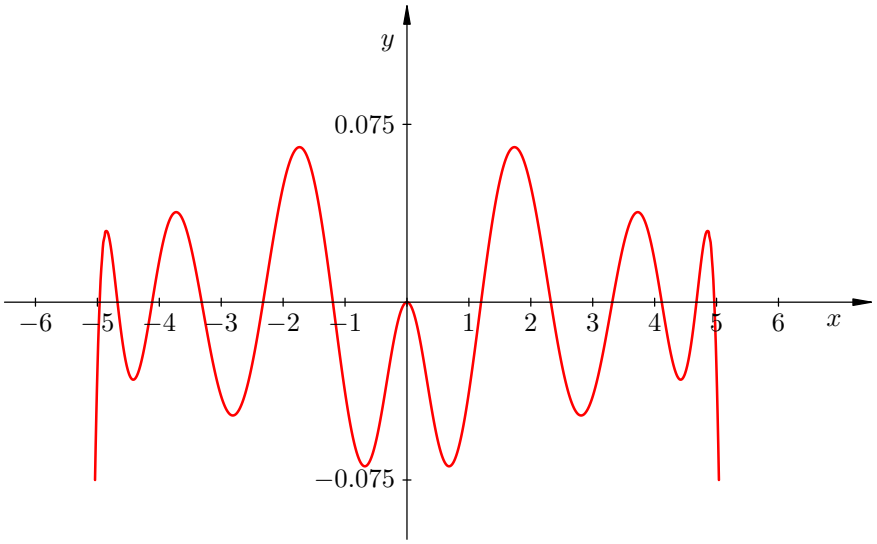
Čebiševljeva mreža, interpolacijski polinom stupnja 10.



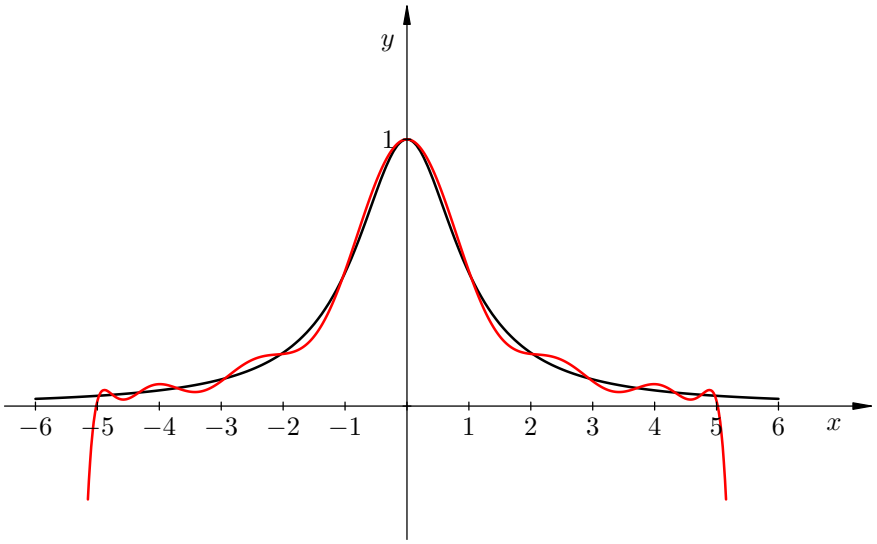
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 10.



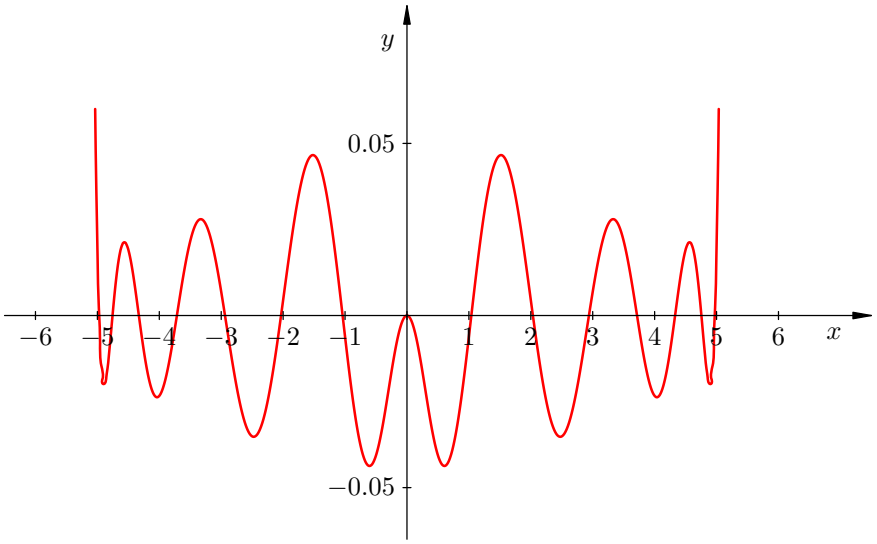
Čebiševljeva mreža, interpolacijski polinom stupnja 12.



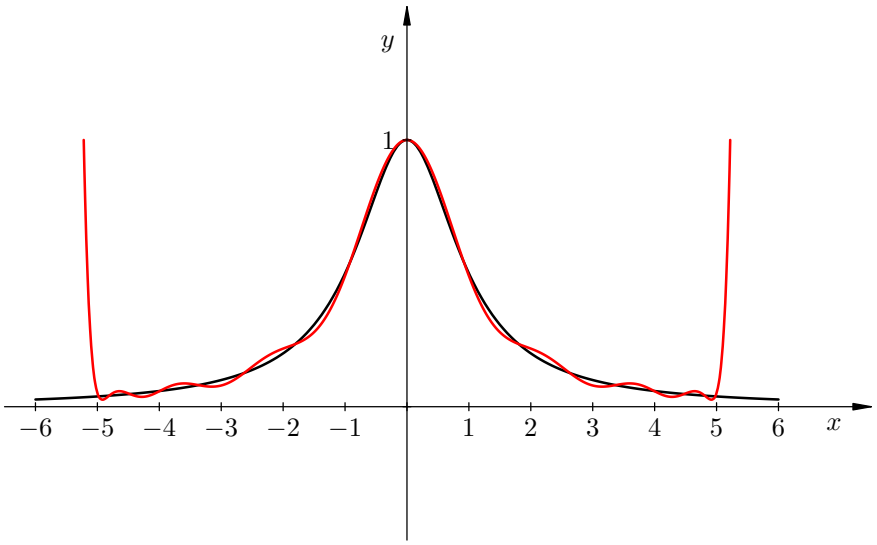
Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 12.



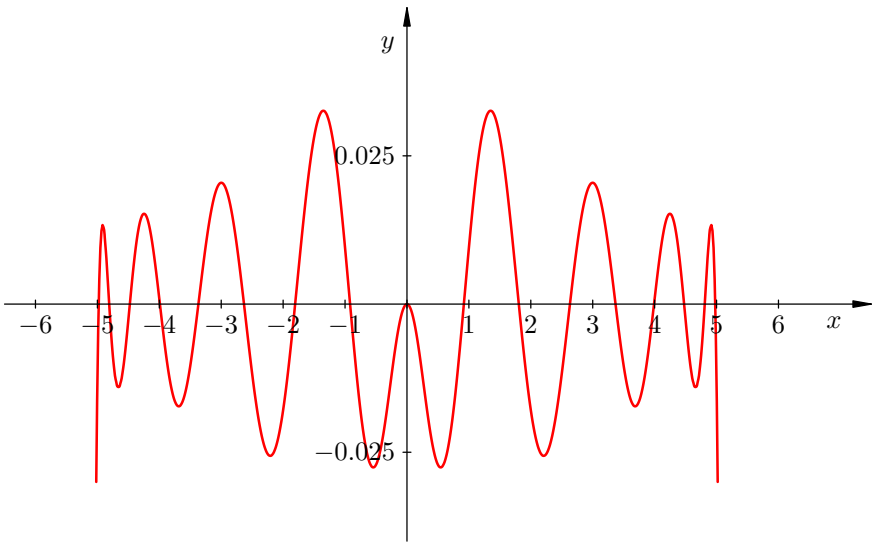
Čebiševljeva mreža, interpolacijski polinom stupnja 14.



Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 14.



Čebiševljeva mreža, interpolacijski polinom stupnja 16.



Čebiševljeva mreža, greška interpolacijskog polinoma stupnja 16.

7.2.7. Konvergencija interpolacijskih polinoma

Interpolacija polinomima vrlo je značajna zbog upotrebe u raznim postupcima u numeričkoj analizi, kao što su numerička integracija, deriviranje, rješavanje diferencijalnih jednadžbi i još mnogo toga.

Međutim, sa stanovišta teorije aproksimacije, interpolacija se ne pokazuje kao sredstvo kojim možemo doći do dobrih aproksimacija funkcija. Istina, poznati Weierstraßov teorem tvrdi da za svaku neprekidnu funkciju $f(x)$ postoji niz polinoma stupnja n, nazovimo ih $B_n(x)$, tako da

$$\|f(x) - B_n(x)\|_\infty \to 0 \qquad \text{za} \quad n \to \infty.$$

Nažalost, primjer funkcije Runge pokazuje da ovakav rezultat općenito ne vrijedi za Lagrangeove interpolacijske polinome — niz polinoma generiran ekvidistantnim mrežama ne konvergira prema toj funkciji ni po točkama (za x dovoljno blizu ruba intervala), a kamo li uniformno.

Postoje i još "gori" primjeri divergencije. Dovoljno je uzeti manje glatku funkciju od funkcije Runge.

Primjer 7.2.4 (Bernstein, 1912.) *Neka je*

$$f(x) = |x|$$

i neka je $p_n(x)$ interpolacijski polinom u $n+1$ ekvidistantnih točaka u $[-1, 1]$. Tada $|f(x) - p_n(x)| \to 0$, kad $n \to \infty$, samo u tri točke: $x = -1, 0, 1$.

Na prvi pogled se čini da to što interpolacija ne mora biti dobra aproksimacija funkcije ovisi o izboru čvorova interpolacije. To je samo djelimično točno, tj. izborom točaka interpolacije možemo poboljšati aproksimativna svojstva interpolacijskih polinoma. Drugi bitni faktor kvalitete je glatkoća funkcije.

Iz primjera funkcije Runge vidi se da je Lagrangeova interpolacija dobrih svojstava aproksimacije u sredini intervala, ali ne i na rubovima. Pitanje je, da li neki izbor neekvidistantne mreže, s čvorovima koji su bliže rubovima intervala, može popraviti konvergenciju. Odgovor nije potpuno jednostavan. Iako se mogu konstruirati mreže (poput Čebiševljeve) na kojima se funkcija Runge bolje aproksimira interpolacijskim polinomima, to je nemoguće napraviti za svaku neprekidnu funkciju.

Sljedeći teorem je egzistencijalnog tipa, ali ukazuje na to da je nemoguće naći dobar izbor točaka interpolacije za svaku funkciju.

Teorem 7.2.3 (Faber, 1914.) *Za svaki mogući izbor točaka interpolacije postoji neprekidna funkcija f, za čiji interpolacijski polinom $p_n(x)$ stupnja n vrijedi*

$$\|f(x) - p_n(x)\|_\infty \not\to 0.$$

7.2.8. Hermiteova i druge interpolacije polinomima

Do sada smo promatrali problem interpolacije polinomima u kojem su zadane samo funkcijske vrijednosti f_i u čvorovima interpolacije x_i. Takva interpolacija

funkcijskih vrijednosti se obično zove Lagrangeova interpolacija (čak i kad ne koristimo samo polinome kao aproksimacijske ili interpolacijske funkcije).

Lagrangeova interpolacija nikako ne iscrpljuje sve moguće slučajeve interpolacije polinomima. Moguće su razne generalizacije ovog problema za funkcije f koje imaju dodatna svojstva, recimo, veći broj derivacija (globalno, ili barem, u okolini svakog čvora).

Da bismo jednostavno došli do tih generalizacija, ponovimo ukratko "izvod" i konstrukciju Lagrangeove interpolacije polinomom. Traženi polinom p_n mora zadovoljavati interpolacijske jednadžbe

$$p_n(x_i) = f_i = f(x_i), \quad i = 0, \ldots, n. \tag{7.2.16}$$

Zapis polinoma p_n u standardnoj bazi potencija $1, x, \ldots, x^n$ vodi na linearni sustav s Vandermondeovom matricom, a za pripadnu Vandermondeovu determinantu (vidjeti teorem 7.2.1) pokazali smo da vrijedi

$$V(x_0, \ldots, x_n) := \det \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix} = \prod_{0 \le i < j \le n} (x_j - x_i). \tag{7.2.17}$$

Iz pretpostavke o međusobnoj različitosti čvorova x_k slijedi regularnost sustava i egzistencija i jedinstvenost polinoma p_n.

Lagrangeov interpolacijski polinom p_n može se napisati i eksplicitno u tzv. **Lagrangeovoj formi**, koja se često zove i **Lagrangeova interpolacijska formula**. Ako definiramo $n+1$ polinom $\{\ell_i(x)\}_{i=0}^n$ specijalnim interpolacijskim uvjetima

$$\ell_i(x_j) := \delta_{ij}, \tag{7.2.18}$$

gdje je δ_{ij} Kroneckerov simbol, tada Lagrangeov interpolacijski polinom koji udovoljava uvjetima (7.2.16) možemo zapisati kao

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\ell_i(x). \tag{7.2.19}$$

Tražene polinome ℓ_i stupnja n, koji su jednoznačno određeni interpolacijskim uvjetima (7.2.18), možemo "pogoditi"

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \ne i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, \ldots, n. \tag{7.2.20}$$

Funkcije ℓ_i zovu se funkcije **Lagrangeove baze**.

Zadatak 7.2.2 *Dokažite da su funkcije Lagrangeove baze linearno nezavisne i čine skup izvodnica za prostor polinoma stupnja n, što opravdava naziv baza.*

Postoji još jedan slučaj koji se može riješiti jednostavnom formulom, a posebno ga tretiramo zbog važnosti za teoriju numeričke integracije (preciznije, Gaussovih integracijskih formula). U svakom čvoru x_i, osim funkcijske vrijednosti $f_i = f(x_i)$, interpoliramo i vrijednost derivacije $f_i' = f'(x_i)$.

Teorem 7.2.4 *Postoji jedinstveni polinom h_{2n+1} stupnja najviše $2n+1$, koji zadovoljava interpolacijske uvjete*

$$h_{2n+1}(x_i) = f_i, \quad h'_{2n+1}(x_i) = f_i', \quad i = 0, \ldots, n,$$

gdje su x_i međusobno različite točke i f_i, f_i' zadani realni brojevi.

Dokaz. Egzistenciju polinoma $h_{2n+1}(x)$ možemo dokazati konstruktivnim metodama — konstrukcijom eksplicitne baze, slično kao i za Lagrangeov polinom. Neka su

$$\begin{aligned} h_{i,0}(x) &= [1 - 2(x - x_i)\ell_i'(x_i)]\, \ell_i^2(x) \\ h_{i,1}(x) &= (x - x_i)\, \ell_i^2(x), \end{aligned} \tag{7.2.21}$$

gdje su ℓ_i funkcije Lagrangeove baze (7.2.20). Direktno možemo provjeriti da su $h_{i,0}(x)$ i $h_{i,1}(x)$ polinomi stupnja $2n+1$ koji zadovoljavaju sljedeće relacije

$$\begin{aligned} h_{i,0}(x_j) &= \delta_{ij}, \quad & h_{i,1}(x_j) &= 0, \\ h'_{i,0}(x_j) &= 0, & h'_{i,1}(x_j) &= \delta_{ij}, \end{aligned} \quad \text{za} \quad i, j = 0, \ldots, n.$$

Ako definiramo polinom formulom

$$h_{2n+1}(x) = \sum_{i=0}^{n} \Big(f_i\, h_{i,0}(x) + f_i'\, h_{i,1}(x) \Big), \tag{7.2.22}$$

lagano provjerimo da h_{2n+1} zadovoljava uvjete teorema.

Obzirom da iz gornjeg ne slijedi jedinstvenost, moramo ju dokazati posebno. Neka je $q_{2n+1}(x)$ bilo koji drugi polinom koji ispunjava interpolacijske uvjete teorema. Tada je $h_{2n+1}(x) - q_{2n+1}(x)$ polinom stupnja ne većeg od $2n+1$, koji ima nultočke multipliciteta barem 2 u svakom čvoru interpolacije x_i, tj. barem $2n+2$ nultočke, što je moguće samo ako je identički jednak nuli. ■

Polinomi $h_{i,0}$, $h_{i,1}$, zovu se funkcije **Hermiteove baze**, a polinom h_{2n+1} obično se zove **Hermiteov interpolacijski polinom**.

Zadatak 7.2.3 *Pokažite da za funkcije Lagrangeove, odnosno Hermiteove baze, vrijedi*

$$\sum_{i=0}^{n} \ell_i(x) = 1, \qquad \sum_{i=0}^{n} h_{i,0}(x) = 1.$$

Zadatak 7.2.4 *Pokažite da za funkcije Lagrangeove, odnosno Hermiteove baze, vrijedi*

$$\sum_{i=0}^{n} x_i h_{i,0}(x) + h_{i,1}(x) = x, \qquad \sum_{i=0}^{n} (x - x_i)\, \ell_i^2(x)\, \ell_i'(x_i) = 0.$$

Za ocjenu greške Hermiteove interpolacije vrijedi vrlo sličan rezultat kao i za običnu Lagrangeovu interpolaciju (teorem 7.2.2).

Teorem 7.2.5 *Greška kod interpolacije Hermiteovim polinomom $h_{2n+1}(x)$ (v. teorem 7.2.4) funkcije $f \in C^{(2n+2)}[x_{\min}, x_{\max}]$ u $n+1$ čvorova $x_0, \ldots, x_n$ je oblika*

$$e(x) := f(x) - h_{2n+1}(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}\, \omega^2(x),$$

gdje su ξ i ω kao u teoremu 7.2.2.

Dokaz. Iz uvjeta interpolacije znamo da je $f(x) = h_{2n+1}(x)$ i $f'(x) = h'_{2n+1}(x)$ za $x = x_0, \ldots, x_n$, pa očekujemo da je

$$f(x) - h_{2n+1}(x) \approx C\omega^2(x)$$

za neku konstantu C. Definiramo li

$$F(x) = f(x) - h_{2n+1}(x) - C\omega^2(x),$$

vidimo da F ima nultočke multipliciteta 2 u $x_0, \ldots, x_n$, tj. $F(x_k) = F'(x_k) = 0$ za $k = 0, \ldots, n$. Izaberemo li neki $x_{n+1} \in [x_{\min}, x_{\max}]$ različit od postojećih čvorova, možemo odrediti konstantu C tako da vrijedi $F(x_{n+1}) = 0$. Kako $F(x)$ sada ima (barem) $n+2$ nule, F' ima $n+1$ nulu u nekim točkama između njih. Ona također ima nule u $x_0, \ldots, x_n$, pa ukupno ima (barem) $2n+2$ nula. No onda F'' ima bar $2n+1$ nula, F''' $2n$ nula, itd., na osnovu Rolleovog teorema. Na kraju, $F^{(2n+2)}$ ima barem jednu nulu u promatranom intervalu, označimo ju s ξ. Deriviranjem izraza za $F(x)$ dobijemo

$$F^{(2n+2)}(\xi) = f^{(2n+2)}(\xi) - C(2n+2)! = 0,$$

odakle izračunamo C. Uvrstimo li taj rezultat u izraz za grešku, dobijemo

$$F(x_{n+1}) - h_{2n+1}(x_{n+1}) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}\, \omega^2(x_{n+1}).$$

Ali kako je x_{n+1} proizvoljan, različit samo od čvorova $x_0, \ldots, x_n$, možemo ga zamijeniti s proizvoljnim x. Na kraju primijetimo da je gornji rezultat točan i za $x \in \{x_0, \ldots, x_n\}$, jer su obje strane nula, pa dokaz slijedi. ■

Hermiteov interpolacijski polinom, naravno, osim u "Lagrangeovom" obliku, možemo zapisati i u "Newtonovom" obliku — koristeći podijeljene razlike, ali sada

i s dvostrukim čvorovima. Što to znači? Pokušajte ga sami izvesti! (Taj oblik ćemo kasnije uvesti i iskoristiti za zapis po dijelovima polinomne interpolacije.)

Ponekad se naziv "Hermiteova interpolacija" koristi i za općenitiji slučaj **proširene Hermiteove interpolacije** koji uključuje i više derivacije od prvih. Bitno je samo da u određenom čvoru x_i interpoliramo **redom** funkcijsku vrijednost i prvih nekoliko (uzastopnih) derivacija.

Pretpostavimo da u čvoru x_i koristimo $l_i > 0$ podataka (funkcija i prvih $l_i - 1$ derivacija). Tada je zgodno gledati x_i kao čvor multipliciteta $l_i \geq 1$ i uvesti posebne oznake t_j za međusobno različite čvorove (uzmimo da ih je $d+1$):

$$x_0 \leq \cdots \leq x_n = \underbrace{t_0, \ldots, t_0}_{l_0}, \ldots, \underbrace{t_d, \ldots, t_d}_{l_d},$$

uz $t_i \neq t_j$ za $i \neq j$, s tim da je $l_0 + \cdots + l_d = n+1$. Problem proširene Hermiteove interpolacije, također, ima jedinstveno rješenje.

Zadatak 7.2.5 *Neka su $t_0, t_1, \ldots, t_d$ zadani međusobno različiti čvorovi i neka su $l_0, l_1, \ldots, l_d$ zadani prirodni brojevi koji zadovoljavaju $\sum_{i=0}^{d} l_i = n+1$. Pokažite da za svaki skup realnih brojeva*

$$\{f_{ij} \mid j = 1, \ldots, l_i,\ i = 0, \ldots, d\}$$

postoji jedinstveni polinom h_n, stupnja ne većeg od n, za koji vrijedi

$$h_n^{(j-1)}(t_i) = f_{ij}, \quad j = 1, \ldots, l_i, \quad i = 0, \ldots, d.$$

Uputa: Konstrukcija Hermiteove baze postaje vrlo komplicirana (pokušajte!). Zato zapišite h_n kao linearnu kombinaciju potencija, formulirajte problem interpolacije matrično i analizirajte determinantu dobivenog linearnog sustava. Ta determinanta je generalizacija Vandermondeove determinante iz (7.2.17)*, bez pretpostavke da su čvorovi različiti, pa ju, također, označavamo s $V(x_0, \ldots, x_n)$. Dokažite da vrijedi*

$$V(x_0, \ldots, x_n) = \prod_{0 \leq i < j \leq d} (t_j - t_i)^{l_i l_j} \cdot \prod_{i=0}^{d} \prod_{\nu=1}^{l_i - 1} \nu!\,,$$

odakle slijedi egzistenicija i jedinstvenost polinoma h_n.

Općeniti slučaj interpolacije funkcije i derivacija, koji obuhvaća gornje interpolacije kao specijalni slučaj, može se zapisati na sljedeći način. Neka je E matrica tipa $(m+1) \times (n+1)$ s elementima E_{ij} koji su svi 0, osim $n+1$ njih, koji su jednaki 1, i neka je zadan skup od $m+1$ točaka $x_0 < x_1 < \cdots < x_m$. Tada problem nalaženja polinoma $P(x)$ stupnja n koji zadovoljava

$$E_{ij}(P^{(j-1)}(x_i) - c_{ij}) = 0, \quad i = 0, \ldots, m, \quad j = 1, \ldots, n+1,$$

za neki izbor brojeva c_{ij}, zovemo **Hermite–Birkhoffovim** interpolacijskim problemom. U punoj općenitosti, kako je formuliran, problem može i nemati rješenje. Identifikacija matrica E koje vode na regularne sisteme jednadžbi već je dosta izučena. I na kraju, spomenimo da i time problem nije do kraja iscrpljen. Moguće je umjesto derivacija zadavati razne linearne funkcionale u čvorovima. Jedan specijalni problem u kojem su ovi linearni funkcionali linearne kombinacije derivacija, donekle je proučen. Taj se problem često naziva **proširena Hermite–Birkhoffova interpolacija**, a u vezi je s numeričkim metodama za rješavanje diferencijalnih jednadžbi.

Zadatak 7.2.6 *Zapisom polinoma P u standardnoj bazi potencija formulirajte matrično problem proširene Hermite–Birkhoffove interpolacije.*

7.3. Interpolacija po dijelovima polinomima

U prošlom smo poglavlju pokazali da polinomna interpolacija visokog stupnja može imati vrlo loša svojstva, pa se u praksi **ne smije** koristiti. Umjesto toga, koristi se po dijelovima polinomna interpolacija, tj. na svakom podintervalu vrijedi

$$\varphi\Big|_{[x_{k-1},x_k]} = p_k, \quad k = 1, 2, \ldots, n,$$

gdje su p_k polinomi niskog (ali fiksnog) stupnja. Za razliku od polinomne interpolacije funkcijskih vrijednosti, gdje je bilo dovoljno da su čvorovi interpolacije međusobno različiti, ovdje pretpostavljamo da su rubovi podintervala interpolacije uzlazno numerirani, tj. da vrijedi $a = x_0 < x_1 < \cdots < x_n = b$. To još ne osigurava da je φ funkcija jer je moguća dvoznačnost u dodirnim točkama podintervala, ali o tome ćemo voditi računa kod zadavanja uvjeta interpolacije.

Preciznije, pretpostavimo da na svakom podintervalu $[x_{k-1}, x_k]$ koristimo polinom stupnja m, tj. da je

$$\varphi\Big|_{[x_{k-1},x_k]} = p_k, \quad k = 1, \ldots, n.$$

Svaki polinom p_k stupnja m određen je s $(m+1)$-im koeficijentom. Ukupno moramo odrediti koeficijente polinoma p_k u n podintervala, tj. ukupno

$$(m+1) \cdot n \tag{7.3.1}$$

koeficijenata. Interpolacijski uvjeti su

$$\varphi(x_k) = f_k, \quad k = 0, \ldots, n,$$

što za svaki polinom daje po 2 uvjeta

$$\begin{aligned} p_k(x_{k-1}) &= f_{k-1} \\ p_k(x_k) &= f_k, \end{aligned} \quad k = 1, \ldots, n, \tag{7.3.2}$$

a ukupno daje $2n$ uvjeta interpolacije. Uočimo da smo postavljenjem prethodnih uvjeta interpolacije osigurali neprekidnost funkcije φ, jer je

$$p_{k-1}(x_{k-1}) = p_k(x_{k-1}), \quad k = 2, \ldots, n.$$

Primijetimo da uvjeta interpolacije ima $2n$, a moramo naći $(m+1) \cdot n$ koeficijenata. Bez dodatnih uvjeta to je moguće napraviti samo za $m = 1$, tj. za po dijelovima linearnu interpolaciju.

Za $m > 1$ moraju se dodati uvjeti na glatkoću interpolacijske funkcije φ u čvorovima interpolacije.

7.3.1. Po dijelovima linearna interpolacija

Osnovna ideja po dijelovima linearne interpolacije je umjesto jednog polinoma visokog stupnja koristiti više polinoma, ali stupnja 1.

Na svakom podintervalu $[x_{k-1}, x_k]$, polinom p_k je jedinstveno određen. Obično ga zapisujemo relativno obzirom na početnu točku intervala (razlog je stabilnost) u obliku

$$p_k(x) = c_{0,k} + c_{1,k}(x - x_{k-1}) \quad \text{za} \quad x \in [x_{k-1}, x_k], \quad k = 1, \ldots, n.$$

Interpolacijski polinom p_k možemo zapisati u Newtonovoj formi

$$p_k(x) = f[x_{k-1}] + f[x_{k-1}, x_k] \cdot (x - x_{k-1}),$$

pa odmah vidimo da vrijedi

$$\begin{aligned} c_{0,k} &= f[x_{k-1}] = f_{k-1} \\ c_{1,k} &= f[x_{k-1}, x_k] = \frac{f_k - f_{k-1}}{x_k - x_{k-1}}, \end{aligned} \qquad k = 1, \ldots, n.$$

Ako želimo aproksimirati vrijednost funkcije f u točki $x \in [a, b]$, prvo treba pronaći između kojih se čvorova točka x nalazi, tj za koji k vrijedi $x_{k-1} \leq x \leq x_k$. Tek tada možemo računati koeficijente pripadnog linearnog polinoma.

Za traženje tog intervala koristimo algoritam binarnog pretraživanja.

Algoritam 7.3.1 (Binarno pretraživanje)

```
low := 0;
high := n;
while (high − low) > 1 do
  begin
```

```
mid := (low + high) div 2;
if x < x_mid then
   high := mid
else
   low := mid
end;
```

Trajanje ovog algoritma je proporcionalno s $\log_2(n)$.

Ako je funkcija f klase $C^2[a,b]$, gdje je $[a,b]$ interval na kojem aproksimiramo, onda je greška takve interpolacije maksimalna pogreška od n linearnih interpolacija. Na podintervalu $[x_{k-1}, x_k]$ ocjena greške linearne interpolacije je

$$|f(x) - p_k(x)| \leq \frac{M_2^k}{2!} |\omega(x)|,$$

pri čemu je

$$\omega(x) = (x - x_{k-1})(x - x_k), \quad M_2^k = \max_{x \in [x_{k-1}, x_k]} |f''(x)|.$$

Ocijenimo $\omega(x)$ na $[x_{k-1}, x_k]$, tj. nađimo njen maksimum po apsolutnoj vrijednosti. Kako je graf od $\omega(x)$ na $[x_{k-1}, x_k]$ parabola koja siječe apscisu u x_{k-1} i x_k, maksimum od $|\omega(x)|$ je u polovištu intervala

$$x_e = \frac{(x_{k-1} + x_k)}{2}.$$

Ovo se može provjeriti i traženjem lokalnog ekstrema fukncije

$$\omega(x) = (x - x_{k-1})(x - x_k).$$

Deriviranjem izalazi

$$\omega'(x) = 2x - (x_{k-1} + x_k),$$

pa je kandidat za lokalni ekstrem točka

$$x_e = \frac{(x_{k-1} + x_k)}{2}.$$

Tvrdimo da je to lokalni minimum od ω, jer je

$$\omega''(x_e) = 2 > 0.$$

Vrijednost funkcije ω u lokalnom ekstremu je

$$\omega(x_e) = (x_e - x_{k-1})(x_e - x_k) = \frac{x_k - x_{k-1}}{2} \cdot \frac{x_{k-1} - x_k}{2} = -\frac{(x_k - x_{k-1})^2}{4}.$$

Osim toga, za bilo koji $x \in (x_{k-1}, x_k)$ vrijedi $\omega(x) < 0$. Odatle, prijelazom na apsolutnu vrijednost, slijedi da je x_e točka lokalnog maksimuma za $|\omega|$ i

$$|\omega(x)| \leq |\omega(x_e)| \leq \frac{(x_k - x_{k-1})^2}{4}, \quad \forall x \in [x_{k-1}, x_k].$$

Ako razmak između susjednih čvorova označimo s $h_k = x_k - x_{k-1}$, možemo definirati maksimalni razmak susjednih čvorova s

$$h = \max_{1 \leq k \leq n} \{h_k\}.$$

pa na čitavom $[a, b]$, možemo pisati

$$|f(x) - \varphi(x)| \leq \frac{M_2}{2!} \frac{h^2}{4} = \frac{1}{8} M_2 \cdot h^2.$$

Drugim riječima, ako ravnomjerno povećavamo broj čvorova, tako da $h \to 0$, onda i maksimalna greška teži u 0.

Na primjer, za ekvidistantne mreže, tj. za mreže za koje vrijedi

$$x_k = a + kh, \quad h = \frac{b - a}{n}$$

je greška reda veličine h^2, odnosno n^{-2} i potrebno je dosta podintervala da se dobije sasvim umjerena točnost aproksimacije. Na primjer, za $h = 0.01$, tj. za $n = 100$, greška aproksimacije je reda veličine 10^{-4}.

Druga je mana da aproksimacijska funkcija φ nije dovoljno glatka, tj. ona je samo neprekidna. Zbog ta dva razloga (dosta točaka za umjerenu točnost i pomanjkanje glatkoće), obično se na svakom podintervalu koriste polinomi viših stupnjeva.

Ako stavimo $m = 2$, tj. na svakom podintervalu postavimo kvadratni polinom (parabolu), moramo naći $3n$ koeficijenata, a imamo $2n$ uvjeta interpolacije. Ako zahtijevamo da aproksimacijska funkcija φ ima u unutarnjim čvorovima interpolacije $x_1, \ldots, x_{n-1}$ neprekidnu derivaciju, onda smo dodali još $n - 1$ uvjet. A treba nam još jedan! Ako i njega postavimo (a to ne možemo na simetričan način), onda bismo mogli naći i takvu aproksimaciju. Ona se uobičajeno ne koristi, jer kontrolu derivacije možemo napraviti samo na jednom rubu (to bi odgovaralo inicijalnim problemima). Po dijelovima parabolička interpolacija nema pravu fizikalnu podlogu, pa se vrlo rijetko koristi (katkad kod računarske grafike). Za razliku od po dijelovima parabolne interpolacije, po dijelovima kubična interpolacija ima vrlo važnu fizikalnu podlogu i vjerojatno je jedna od najčešće korištenih metoda interpolacije uopće.

7.3.2. Po dijelovima kubična interpolacija

Kod po dijelovima kubične interpolacije, restrikcija aproksimacijske funkcije φ na svaki interval je kubični polinom. Njega uobičajeno zapisujemo relativno obzirom

na početnu točku intervala $[x_{k-1}, x_k]$ u obliku

$$p_k(x) = c_{0,k} + c_{1,k}(x - x_{k-1}) + c_{2,k}(x - x_{k-1})^2 + c_{3,k}(x - x_{k-1})^3 \quad \text{za} \quad x \in [x_{k-1}, x_k], \quad k = 1, \ldots, n. \tag{7.3.3}$$

Budući da ukupno imamo n kubičnih polinoma, od kojih svakome treba odrediti 4 koeficijenta, ukupno moramo odrediti $4n$ koeficijenata. Uvjeta interpolacije je $2n$, jer svaki kubični polinom p_k mora interpolirati rubove svog podintervala $[x_{k-1}, x_k]$, tj. mora vrijediti

$$\begin{aligned} p_k(x_{k-1}) &= f_{k-1} \\ p_k(x_k) &= f_k, \end{aligned} \quad k = 1, \ldots, n.$$

Ovi uvjeti automatski osiguravaju neprekidnost funkcije φ. Obično želimo da interpolacijska funkcija bude glađa — barem klase $C^1[a, b]$, tj. da je i derivacija funkcije φ neprekidna i u čvorovima. Dodavanjem tih uvjeta za svaki kubični polinom, dobivamo još $2n$ uvjeta

$$\begin{aligned} p_k'(x_{k-1}) &= s_{k-1} \\ p_k'(x_k) &= s_k, \end{aligned} \quad k = 1, \ldots, n,$$

pri čemu su s_k neki brojevi. Njihova uloga može biti višeznačna, pa ćemo je detaljno opisati kasnije. Zasad, možemo zamišljati da su brojevi s_k neke aproksimacije derivacije u čvorovima.

Primijetite da je takvim izborom dodatnih uvjeta osigurana neprekidnost prve derivacije, jer je

$$p_{k-1}'(x_{k-1}) = p_k'(x_{k-1}) = s_{k-1}, \quad k = 2, \ldots, n.$$

Ako pretpostavimo da su s_k zadani brojevi, nađimo koeficijente interpolacijskog polinoma p_k.

Ponovno, najzgodnije je koristiti Newtonov oblik interpolacijskog polinoma, ali sada s tzv. dvostrukim čvorovima, jer su u x_{k-1} i x_k dani i funkcijska vrijednost i derivacija.

Što, zapravo, znači dvostruki čvor? Pretpostavimo li da se u podijeljenoj razlici dva čvora približavaju jedan drugom, onda je podijeljena razlika na limesu

$$\lim_{h_k \to 0} f[x_k, x_k + h_k] = \lim_{h_k \to 0} \frac{f(x_k + h_k) - f(x_k)}{h_k} = f'(x_k),$$

uz uvjet da f ima derivaciju u točki x_k. Drugim riječima, vrijedi

$$f[x_k, x_k] = f'(x_k).$$

U našem slučaju, ako u točki x_k derivaciju $f'(x_k)$ zadajemo ili aproksimiramo s s_k, onda je

$$f[x_k, x_k] = s_k.$$

Sada možemo napisati tablicu podijeljenih razlika za kubični interpolacijski polinom koji ima dva dvostruka čvora x_{k-1} i x_k. To si je najjednostavnije predočiti kao kubični interpolacijski polinom koji prolazi kroz četiri točke: x_{k-1}, točkom koja je "jako blizu" x_{k-1}, točkom koja je "jako blizu" x_k i točkom x_k. Kad se te dvije točke koje su "jako blizu" stope sa svojim parom, dobivamo dva dvostruka čvora, pa tablica podijeljenih razlika izgleda ovako:

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$	$f[x_k, x_{k+1}, x_{k+2}, x_{k+3}]$
x_{k-1}	f_{k-1}			
		s_{k-1}		
x_{k-1}	f_{k-1}		$\dfrac{f[x_{k-1}, x_k] - s_{k-1}}{h_k}$	
		$f[x_{k-1}, x_k]$		$\dfrac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k^2}$
x_k	f_k		$\dfrac{s_k - f[x_{k-1}, x_k]}{h_k}$	
		s_k		
x_k	f_k			

Forma Newtonovog interpolacijskog polinoma ostat će po obliku jednaka kao u slučaju da su sve četiri točke različite, pa imamo

$$\begin{aligned} p_k(x) = f[x_{k-1}] &+ f[x_{k-1}, x_{k-1}] \cdot (x - x_{k-1}) \\ &+ f[x_{k-1}, x_{k-1}, x_k] \cdot (x - x_{k-1})^2 \\ &+ f[x_{k-1}, x_{k-1}, x_k, x_k] \cdot (x - x_{k-1})^2 (x - x_k) \end{aligned} \tag{7.3.4}$$

uz uvažavanje da je

$$\begin{aligned} f[x_{k-1}, x_{k-1}] &= s_{k-1} \\ f[x_{k-1}, x_{k-1}, x_k] &= \frac{f[x_{k-1}, x_k] - [x_{k-1}, x_{k-1}]}{x_k - x_{k-1}} \\ &= \frac{f[x_{k-1}, x_k] - s_{k-1}}{h_k} \\ f[x_{k-1}, x_{k-1}, x_k, x_k] &= \frac{f[x_{k-1}, x_k, x_k] - f[x_{k-1}, x_{k-1}, x_k]}{x_k - x_{k-1}} \\ &= \frac{\dfrac{s_k - f[x_{k-1}, x_k]}{h_k} - \dfrac{f[x_{k-1}, x_k] - s_{k-1}}{h_k}}{h_k} \\ &= \frac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k^2}. \end{aligned}$$

Uvrštavanjem x_{k-1} i x_k u polinom p_k iz (7.3.4), te u njegovu derivaciju p_k' možemo provjeriti da je

$$\begin{aligned} p_k(x_{k-1}) &= f_{k-1}, \qquad & p_k'(x_{k-1}) &= s_{k-1}, \\ p_k(x_k) &= f_k, & p_k'(x_k) &= s_k. \end{aligned}$$

Drugim riječima, našli smo traženi p_k. Usporedimo li forme (7.3.3) i (7.3.4), dobit ćemo koeficijente $c_{i,k}$. Relaciju (7.3.4) možemo malo drugačije zapisati, tako da polinom bude napisan po potencijama od $(x - x_{k-1})$. Ako posljednji član tog polinoma možemo napišemo kao

$$\begin{aligned} (x - x_{k-1})^2(x - x_k) &= (x - x_{k-1})^2(x - x_{k-1} + x_{k-1} - x_k) \\ &= (x - x_{k-1})^2(x - x_{k-1} - h_k) \\ &= (x - x_{k-1})^3 - h_k(x - x_{k-1})^2, \end{aligned}$$

onda relacija (7.3.4) poprima oblik

$$\begin{aligned} p_k(x) = f[x_{k-1}] &+ f[x_{k-1}, x_{k-1}] \cdot (x - x_{k-1}) \\ &+ (f[x_{k-1}, x_{k-1}, x_k] - h_k\, f[x_{k-1}, x_{k-1}, x_k, x_k]) \cdot (x - x_{k-1})^2 \\ &+ f[x_{k-1}, x_{k-1}, x_k, x_k] \cdot (x - x_{k-1})^3. \end{aligned}$$

Uspoređivanjem koeficijenata uz odgovarajuće potencije prethodne relacije i relacije (7.3.3), za sve $k = 1, \ldots, n$, dobivamo

$$\begin{aligned} c_{0,k} &= p_k(x_{k-1}) = f_{k-1}, \\ c_{1,k} &= p_k'(x_{k-1}) = s_{k-1}, \\ c_{2,k} &= \frac{p_k''(x_{k-1})}{2} = f[x_{k-1}, x_{k-1}, x_k] - h_k\, f[x_{k-1}, x_{k-1}, x_k, x_k], \\ c_{3,k} &= \frac{p_k'''(x_{k-1})}{6} = f[x_{k-1}, x_{k-1}, x_k, x_k]. \end{aligned}$$

Promotrimo li bolje posljednje dvije relacije, otkrivamo da se isplati prvo izračunati koeficijent $c_{3,k}$, a zatim ga upotrijebiti za računanje $c_{2,k}$. Dobivamo

$$\begin{aligned} c_{3,k} &= \frac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k^2}, \\ c_{2,k} &= \frac{f[x_{k-1}, x_k] - s_{k-1}}{h_k} - h_k c_{3,k}. \end{aligned}$$

Drugim riječima, ako znamo skalare s_k, onda nije problem naći koeficijente po dijelovima kubične interpolacije. Ostaje nam samo pokazati kako bismo mogli birati s_k-ove. Ponovno, postoje dva bitno različita načina.

7.3.3. Po dijelovima kubična Hermiteova interpolacija

Ako su poznate vrijednosti derivacija funkcije f u čvorovima x_k, skalare s_k možemo izabrati tako da vrijedi

$$s_k = f'(x_k), \quad k = 0, \ldots, n.$$

U tom slučaju je kubični polinom određen **lokalno**, tj. ne ovisi o drugim kubičnim polinomima. Naime, ako su kubičnom polinomu na rubovima intervala zadane i funkcijske vrijednost i vrijednosti derivacija, potpuno su određena njegova četiri koeficijenta. Interpolacija koja interpolira funkcijske vrijednosti i vrijednosti derivacija u svim zadanim čvorovima zove se po dijelovima kubična Hermiteova interpolacija.

Nađimo grešku takve interpolacije, uz pretpostavku da je funkcija $f \in C^4[a, b]$. Prvo, pronađimo grešku na intervalu $[x_{k-1}, x_k]$. Interpolacijski polinom s dvostrukim čvorovima na rubu ponaša se kao polinom koji ima četiri različita čvora, takva da se parovi čvorova u rubu "stope". Zbog toga, možemo promatrati grešku interpolacijskog polinoma reda 3 koji interpolira funkciju f u točkama x_{k-1}, x_k i još dvijema točkama koje su blizu x_{k-1} i x_k. Grešku takvog interpolacijskog polinoma možemo ocijeniti s

$$|f(x) - p_k(x)| \leq \frac{M_4^k}{4!} |\omega(x)|,$$

pri čemu je, nakon "stapanja točaka",

$$\omega(x) = (x - x_{k-1})^2 (x - x_k)^2, \quad M_4^k = \max_{x \in [x_{k-1}, x_k]} |f^{(4)}(x)|.$$

Ostaje samo još pronaći u kojoj je točki intervala $[x_{k-1}, x_k]$ maksimum funkcije $|\omega|$.

Dovoljno je naći sve lokalne ekstreme funkcije ω i u njima provjeriti vrijednost. Derivirajmo

$$\begin{aligned} \omega'(x) &= 2(x - x_{k-1})(x - x_k)^2 + 2(x - x_{k-1})^2 (x - x_k) \\ &= 2(x - x_{k-1})(x - x_k)(2x - x_{k-1} - x_k). \end{aligned}$$

Budući da maksimum greške ne može biti u rubovima intervala, jer su tamo točke interpolacije (tj. minimumi i greške i $|\omega|$), onda je jedino još moguće da se ekstrem dostiže u nultočki x_e od ω', pri čemu je

$$x_e = \frac{(x_{k-1} + x_k)}{2}.$$

Lako se provjerava da je to lokalni maksimum. Vrijednost u x_e je kvadrat vrijednosti greške za po dijelovima linearnu interpolaciju na istoj mreži čvorova

$$\omega(x_e) = (x_e - x_{k-1})^2 (x_e - x_k)^2 = \frac{(x_k - x_{k-1})^4}{16}.$$

Odatle, prijelazom na apsolutnu vrijednost, odmah slijedi da je x_e točka lokalnog maksimuma za $|\omega|$ i

$$|\omega(x)| \leq |\omega(x_e)| \leq \frac{(x_k - x_{k-1})^4}{16}, \quad \forall x \in [x_{k-1}, x_k].$$

Definiramo li, ponovno, maksimalni razmak čvorova

$$h = \max_{1 \leq k \leq n} \{h_k = x_k - x_{k-1}\},$$

na čitavom $[a, b]$ možemo pisati

$$|f(x) - \varphi(x)| \leq \frac{M_4}{4!} \frac{h^4}{16} = \frac{1}{384} M_4 \cdot h^4.$$

Drugim riječima, ako ravnomjerno povećavamo broj čvorova, tako da $h \to 0$, onda i maksimalna greška teži u 0.

Ipak, u cijelom ovom pristupu ima jedan problem. Vrlo često derivacije funkcije u točkama interpolacije nisu poznate, na primjer ako su točke dobivene mjerenjem. No, tada možemo aproksimirati prave vrijednosti derivacije korištenjem vrijednosti funkcije u susjednim točkama. Ostaje još samo pokazati kako.

7.3.4. Numeričko deriviranje

Problem koji trebamo riješiti je kako aproksimirati derivaciju diferencijabilne funkcije f u nekoj točki, recimo x_0 i susjednim točkama $x_1, \ldots, x_n$, korištenjem samo vrijednosti funkcije f u zadanim točkama.

Taj problem možemo riješiti korištenjem interpolacijskog polinoma. Tada, uz pretpostavku da je f klase $C^{n+1}[a, b]$, funkciju f možemo napisati (vidjeti relaciju (7.2.6)) kao

$$f(x) = p_n(x) + e_n(x),$$

gdje je $p_n(x)$ interpolacijski polinom napisan, recimo, u Newotnovoj formi

$$p_n(x) = f[x_0] + (x - x_0) f[x_0, x_1] + \cdots + (x - x_0) \cdots (x - x_{n-1}) f[x_0, x_1, \ldots, x_n],$$

a $e_n(x)$ greška interpolacijskog polinoma

$$e_n(x) = \frac{\omega(x)}{(n+1)!} f^{(n+1)}(\xi).$$

Deriviranjem interpolacijskog polinoma, a zatim uvrštavanjem $x = x_0$ dobivamo

$$\begin{aligned} p_n'(x_0) = f[x_0, x_1] &+ (x_0 - x_1) f[x_0, x_1, x_2] \\ &+ \cdots + (x_0 - x_1) \cdots (x_0 - x_{n-1}) f[x_0, x_1, \ldots, x_n]. \end{aligned}$$

Ako pretpostavimo da f ima još jednu neprekidnu derivaciju, tj. da je f klase $C^{n+2}[a,b]$, onda dobivamo i da je

$$e_n'(x_0) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\,(x_0 - x_1)\cdots(x_0 - x_n).$$

Dakle, $p_n'(x_0)$ je aproksimacija derivacije funkcije f u točki x_0 i vrijedi

$$f'(x_0) = p_n'(x_0) + e_n'(x_0).$$

Ako označimo s

$$H = \max_k |x_0 - x_k|,$$

onda je, za $H \to 0$, greška $e_n'(x_0)$ reda veličine

$$e_n'(x_0) \leq O(H^n).$$

To nam pokazuje da aproksimacijska formula za derivaciju može biti proizvoljno visokog reda n, ali takve formule s velikim n imaju ograničenu praktičnu vrijednost.

Pokažimo kako se ta formula ponaša za male n. Za $n = 1$ imamo

$$p_1'(x_0) = f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{f_1 - f_0}{h},$$

pri čemu smo napravili grešku

$$e_1'(x_0) = \frac{f^{(2)}(\xi)}{2}\,h,$$

uz pretpostavku da je $f \in C^3[x_0, x_1]$. Greška je reda veličine $O(h)$ za $h \to 0$.

Za $n = 2$, uzmimo točke x_1 i x_2 koje se nalaze simetrično oko x_0 (to je poseban slučaj!), tj.

$$x_1 = x_0 + h, \quad x_2 = x_0 - h.$$

Puno sugestivnija notacija točaka u tom slučaju je da s x_{-1} označimo x_2, jer onda točke pišemo u prirodnom redosljedu: x_{-1}, x_0, x_1. U tom slučaju je

$$p_2'(x_0) = f[x_0, x_1] + (x_0 - x_1) f[x_0, x_1, x_{-1}].$$

Izračunajmo potrebne podijeljene razlike.

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$
x_{-1}	f_{-1}		
		$\dfrac{f_0 - f_{-1}}{h}$	
x_0	f_0		$\dfrac{f_1 - 2f_0 + f_{-1}}{2h^2}$
		$\dfrac{f_1 - f_0}{h}$	
x_1	f_1		

Uvrštavanjem dobivamo

$$p_2'(x_0) = \frac{f_1 - f_0}{h} - h\,\frac{f_1 - 2f_0 + f_{-1}}{2h^2} = \frac{f_1 - f_{-1}}{2h}.$$

Ovu posljednju formulu često zovemo simetrična (centralna) razlika, jer su točke x_1 i x_{-1} simetrične obzirom na x_0. Takva aproksimacija derivacije ima bolju ocjenu greške nego obične podijeljene razlike, tj. vrijedi

$$e_2'(x_0) = \frac{f^{(3)}(\xi)}{6}\,(x_0 - x_1)\,(x_0 - x_{-1}) = -h^2\,\frac{f^{(3)}(\xi)}{6}.$$

Pokažimo što bi se zbivalo kad točke x_1 i x_{-1} (odnosno x_2) ne bismo simetrično rasporedili oko x_0. Na primjer, uzmimo

$$x_1 = x_0 + h, \quad x_2 = x_0 + 2h.$$

Iako su i u ovom slučaju točke ekvidistantne, deriviramo u najljevijoj, a ne u srednjoj točki. Pripadna tablica podijeljenih razlika je

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$
x_0	f_0		
		$\dfrac{f_1 - f_0}{h}$	
x_1	f_1		$\dfrac{f_2 - 2f_1 + f_0}{2h^2}$
		$\dfrac{f_2 - f_1}{h}$	
x_2	f_2		

Konačno, aproksimacija derivacije u x_0 je

$$\begin{aligned} p_2'(x_0) &= f[x_0, x_1] + (x_0 - x_1) f[x_0, x_1, x_{-1}] = \frac{f_1 - f_0}{h} - h\frac{f_2 - 2f_1 + f_0}{2h^2} \\ &= \frac{-f_2 + 4f_1 - 3f_0}{2h}, \end{aligned}$$

dok je greška jednaka

$$e_2'(x_0) = \frac{f^{(3)}(\xi)}{6}\,(x_0 - x_1)\,(x_0 - x_2) = h^2\,\frac{f^{(3)}(\xi)}{3},$$

tj. greška je istog reda veličine $O(h^2)$, međutim konstanta je dvostruko veća nego u prethodnom (simetričnom) slučaju.

Primijetite da formula za derivaciju postaje sve točnija što su bliže točke iz kojih se derivacija aproksimira, tj. što je h manji, naravno, uz pretpostavku

da je funkcija f dovoljno glatka. Međutim, to vrijedi samo u teoriji. U praksi, mnogi podaci su mjereni, pa nose neku pogrešku, u najmanju ruku zbog grešaka zaokruživanja.

Kao što ste vidjeli u prethodnim primjerima, osnovu numeričkog deriviranja čine podijeljene razlike, pa ako su točke bliske, dolazi do kraćenja. To nije slučajno. Do kraćenja **mora** doći, zbog neprekidnosti funkcije f. Problem je to izrazitiji, što su točke bliže, tj. što je h manji. Dakle, za numeričko deriviranje imamo dva oprečna zahtjeva na veličinu h. Manji h daje bolju ocjenu greške, ali veću grešku zaokruživanja.

Ilustrirajmo to analizom simetrične razlike,

$$f'(x_0) = \frac{f_1 - f_{-1}}{2h} + e_2'(x_0), \quad e_2'(x_0) = -h^2\,\frac{f^{(3)}(\xi)}{6}.$$

Pretpostavimo da smo, umjesto vrijednosti f_{-1} i f_1, uzeli malo perturbirane vrijednosti

$$\widehat{f}_1 = f_1 + \varepsilon_1, \quad \widehat{f}_{-1} = f_{-1} + \varepsilon_{-1}, \quad |\varepsilon_1|, |\varepsilon_{-1}| \leq \varepsilon.$$

Ako odatle izrazimo f_1 i f_{-1} i uvrstimo ih u formulu za derivaciju, dobivamo

$$f'(x_0) = \frac{\widehat{f}_1 - \widehat{f}_{-1}}{2h} - \frac{\varepsilon_1 - \varepsilon_{-1}}{2h} + e_2'(x_0).$$

Prvi član s desne strane je ono što smo mi zaista izračunali kao aproksimaciju derivacije, a ostalo je greška. Da bismo analizu napravili jednostavnijom, pretpostavimo da je h prikaziv u računalu i da je greška pri računanju kvocijenta u podijeljenoj razlici zanemariva. U tom je slučaju napravljena ukupna greška

$$err_2 = f'(x_0) - \frac{\widehat{f}_1 - \widehat{f}_{-1}}{2h} = -\frac{\varepsilon_1 - \varepsilon_{-1}}{2h} + e_2'(x_0).$$

Ogradimo err_2 po apsolutnoj vrijednosti. Greška u prvom članu je najveća ako su ε_1 i ε_{-1} suprotnih predznaka, maksimalne apsolutne vrijednosti ε. Za drugi član koristimo ocjenu za $e_2'(x_0)$, pa zajedno dobivamo

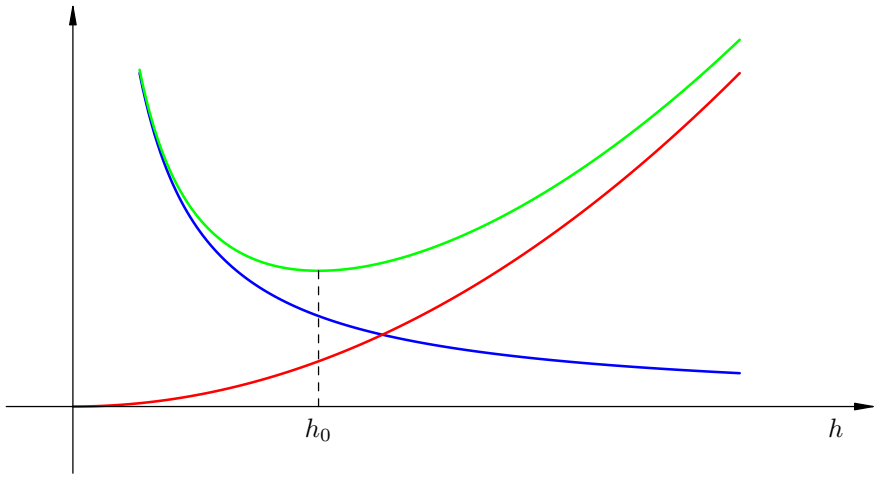
$$|err_2| \leq \frac{\varepsilon}{h} + \frac{M_3}{6}h^2, \quad M_3 = \max_{x\in[x_{-1},x_1]} |f^{(3)}(x)|.$$

Lako se vidi da je ocjena na desnoj strani najbolja moguća, tj. da se može dostići. Označimo tu ocjenu s $e(h)$

$$e(h) := \frac{\varepsilon}{h} + \frac{M_3}{6}h^2.$$

Ponašanje ove ocjene i njezina dva člana u ovisnosti od h možemo prikazati sljedećim grafom. Plavom bojom označen je prvi član ε/h oblika hiperbole, koji dolazi od greške u podacima, a crvenom bojom drugi član oblika parabole, koji predstavlja

maksimalnu grešku odbacivanja kod aproksimacije derivacije podijeljenom razlikom. Zelena boja označava njihov zbroj $e(h)$.



Odmah vidimo da $e(h)$ ima minimum po h. Taj minimum se lako računa, jer iz

$$e'(h) = -\frac{\varepsilon}{h^2} + \frac{M_3}{3}h = 0$$

izlazi da se lokalni, a onda (zbog $e''(h) > 0$ za $h > 0$) i globalni minimum postiže za

$$h_0 = \left(\frac{3\varepsilon}{M_3}\right)^{1/3}.$$

Najmanja vrijednost funkcije je

$$e(h_0) = \frac{3}{2}\left(\frac{M_3}{3}\right)^{1/3}\varepsilon^{2/3}.$$

To pokazuje da čak i u najboljem slučaju, kad je ukupna greška najmanja, dobivamo da je ona reda veličine $O(\varepsilon^{2/3})$, a ne $O(\varepsilon)$, kao što bismo željeli. To predstavlja značajni gubitak točnosti. Posebno, daljnje smanjivanje koraka h samo povećava grešku!

Isti problem se javlja, i to u još ozbiljnijem obliku, u formulama višeg reda za aproksimaciju derivacija.

Primjer 7.3.1 *Nađite po dijelovima kubičnu Hermiteovu interpolaciju za podatke*

x_k	0	1	2
f_k	1	2	0
f'_k	0	1	1

.

Očito, treba povući dva kubična polinoma p_1 i p_2. Polinom p_1 je dio funkcije φ na intervalu $[0,1]$, a p_2 na $[1,2]$.Prije računanja ovih polinoma, uvedimo još

skraćenu oznaku za podijeljene razlike reda j, po ugledu na oznaku za derivacije višeg reda,

$$f^{[j]}[x_k] := f[x_k, \ldots, x_{k+j}], \quad j \geq 0,$$

tako da tablice u prvom redu imaju kraće oznake za stupce.

Za prvi polinom imamo sljedeću tablicu podijeljenih razlika

x_k	f_k	$f^{[1]}[x_k]$	$f^{[2]}[x_k]$	$f^{[3]}[x_k]$
0	1			
		0		
0	1		1	
		1		-1
1	2		0	
		1		
1	2			

Iz nje dobivamo

$$p_1(x) = 1 + (1+1)(x-0)^2 - 1(x-0)^3 = 1 + 2x^2 - x^3.$$

Na sličan način, za p_2 dobivamo tablicu podijeljenih razlika

x_k	f_k	$f^{[1]}[x_k]$	$f^{[2]}[x_k]$	$f^{[3]}[x_k]$
1	2			
		1		
1	2		-3	
		-2		6
2	0		3	
		1		
2	0			

pa je

$$\begin{aligned} p_2(x) &= 2 + (x-1) + (-3-6)(x-1)^2 + 6(x-1)^3 \\ &= 2 + (x-1) - 9(x-1)^2 + 6(x-1)^3. \end{aligned}$$

7.3.5. Po dijelovima kubična kvazihermiteova interpolacija

Sad se možemo vratiti problemu kako napraviti po dijelovima kubičnu Hermiteovu interpolaciju, ako nemamo zadane derivacije. U tom slučaju derivacije možemo aproksimirati na različite načine, a samu interpolaciju ćemo zvati kvazihermiteova po dijelovima kubična interpolacija.

Primijetimo da u slučaju aproksimacije derivacije, greška po dijelovima kubične interpolacije ovisi o tome koliko je dobra aproksimacija derivacije.

Najjednostavnije je uzeti podijeljene razlike kao aproksimacije derivacija u čvorovima. One mogu biti **unaprijed** (do na posljednju) ili **unazad** (do na prvu).

Ako koristimo podijeljene razlike unaprijed, onda je

$$s_k = \begin{cases} \dfrac{f_{k+1} - f_k}{x_{k+1} - x_k}, & \text{za } k = 0, \ldots, n-1, \\ \dfrac{f_n - f_{n-1}}{x_n - x_{n-1}}, & \text{za } k = n, \end{cases}$$

a ako koristimo podijeljene razlike unazad, onda je

$$s_k = \begin{cases} \dfrac{f_1 - f_0}{x_1 - x_0}, & \text{za } k = 0, \\ \dfrac{f_k - f_{k-1}}{x_k - x_{k-1}}, & \text{za } k = 1, \ldots, n. \end{cases}$$

Međutim, prema prethodnom odjeljku, greška koju smo napravili takvom aproksimacijom derivacije je reda veličine $O(h)$ u derivaciji, odnosno $O(h^2)$ u funkcijskoj vrijednosti, što je dosta loše.

Prethodnu aproksimaciju možemo ponešto popraviti ako su točke x_k ekvidistantne, a koristimo simetričnu razliku (osim na lijevom i desnom rubu gdje to nije moguće). Uz oznaku $h = x_k - x_{k-1}$, u tom slučaju možemo staviti

$$s_k = \begin{cases} \dfrac{f_1 - f_0}{h}, & \text{za } k = 0, \\ \dfrac{f_{k+1} - f_{k-1}}{2h}, & \text{za } k = 1, \ldots, n-1, \\ \dfrac{f_n - f_{n-1}}{h}, & \text{za } k = n. \end{cases}$$

U ovom će se slučaju greška obzirom na obične podijeljene razlike popraviti tamo gdje se koristi simetrična razlika. Nažalost, najveće greške ostat će u prvom i posljednjem podintervalu, gdje nije moguće koristiti simetričnu razliku.

Kao što smo vidjeli, postoje i bolje aproksimacije derivacija, a pripadni kvazihermiteovi kubični polinomi obično dobivaju ime po načinu aproksimacije derivacija.

Ako derivaciju u točki x_k aproksimiramo tako da povučemo kvadratni interpolacijski polinom kroz x_{k-1}, x_k i x_{k+1}, a zatim ga deriviramo, pripadna kvazihermiteova interpolacija zove se Besselova po dijelovima kubična interpolacija. Naravno, u prvom i posljednjem čvoru ne možemo postupiti na jednak način (jer nema lijeve, odnosno desne točke). Zbog toga derivaciju u x_0 aproksimiramo tako da povučemo kvadratni interpolacijski polinom kroz x_0, x_1 i x_2, i njega deriviramo u x_0. Slično, derivaciju u x_n aproksimiramo tako da povučemo kvadratni interpolacijski polinom kroz x_{n-2}, x_{n-1} i x_n, i njega deriviramo u x_n.

U unutrašnjim čvorovima x_k, za $k = 1, \ldots, n-1$, dobivamo

$$p_{2,k}(x) = f_{k-1} + f[x_{k-1}, x_k](x - x_{k-1}) + f[x_{k-1}, x_k, x_{k+1}](x - x_{k-1})\,(x - x_k),$$

a zatim, deriviranjem i uvrštavanjem x_k

$$s_k = p'_{2,k}(x_k) = f[x_{k-1}, x_k] + f[x_{k-1}, x_k, x_{k+1}](x_k - x_{k-1}).$$

Uz oznaku

$$h_k = x_k - x_{k-1}, \quad k = 1, \ldots, n,$$

prethodna se formula može napisati i kao

$$s_k = f[x_{k-1}, x_k] + h_k \frac{f[x_k, x_{k+1}] - f[x_{k-1}, x_k]}{h_k + h_{k+1}} = \frac{h_{k+1} f[x_{k-1}, x_k] + h_k f[x_k, x_{k+1}]}{h_k + h_{k+1}},$$

tj. s_k je težinska srednja vrijednost podijeljene razlike unaprijed i unatrag.

Za $k = 0$ pripadni polinom ima oblik

$$p_{2,1}(x) = f_0 + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)\,(x - x_1).$$

Deriviranjem, pa uvrštavanjem x_0 dobivamo

$$s_0 = p'_{2,1}(x_0) = f[x_0, x_1] + f[x_0, x_1, x_2](x_0 - x_1) = \frac{(2h_1 + h_2) f[x_0, x_1] - h_1 f[x_1, x_2]}{h_1 + h_2}.$$

Za $k = n$ pripadni polinom je

$$p_{2,n-1}(x) = f_{n-2} + f[x_{n-2}, x_{n-1}](x - x_{n-2}) + f[x_{n-2}, x_{n-1}, x_n](x - x_{n-2})\,(x - x_{n-1}).$$

Deriviranjem, pa uvrštavanjem x_n dobivamo

$$\begin{aligned} s_n &= p'_{2,n-1}(x_n) = f[x_{n-2}, x_{n-1}] + f[x_{n-2}, x_{n-1}, x_n](2x_n - x_{n-1} - x_{n-2}) \\ &= \frac{(h_{n-1} + 2h_n) f[x_{n-2}, x_{n-1}] - h_n f[x_{n-1}, x_n]}{h_{n-1} + h_n}. \end{aligned}$$

Dakle, za Besselovu po dijelovima kubičnu interpolaciju stavljamo

$$s_k = \begin{cases} \dfrac{(2h_1 + h_2) f[x_0, x_1] - h_1 f[x_1, x_2]}{h_1 + h_2}, & \text{za } k = 0, \\ \dfrac{h_{k+1} f[x_{k-1}, x_k] + h_k f[x_k, x_{k+1}]}{h_k + h_{k+1}}, & \text{za } k = 1, \ldots, n-1, \\ \dfrac{(h_{n-1} + 2h_n) f[x_{n-2}, x_{n-1}] - h_n f[x_{n-1}, x_n]}{h_{n-1} + h_n}, & \text{za } k = n. \end{cases}$$

Greška u derivaciji (vidjeti prethodni odjeljak) je reda veličine $O(h^2)$, što znači da je greška u funkciji reda veličine $O(h^3)$.

Postoji još jedna varijanta aproksimacije derivacija "s imenom". Akima je 1970. godine dao sljedeću aproksimaciju koja usrednjava podijeljene razlike, s ciljem da se spriječe oscilacije interpolacijske funkcije φ:

$$s_k = \frac{w_{k+1} f[x_{k-1}, x_k] + w_{k-1} f[x_k, x_{k+1}]}{w_{k+1} + w_{k-1}}, \quad k = 0, 1, \ldots, n-1, n,$$

uz

$$w_k = |f[x_k, x_{k+1}] - f[x_{k-1}, x_k]|$$

i $w_{-1} = w_0 = w_1$, $w_{n-1} = w_n = w_{n+1}$.

Za $k = 0$ i $k = n$, ove formule se ne mogu odmah iskoristiti, bez dodatnih definicija. Naime, kraćenjem svih težina w_k u formuli za $k = 0$ dobivamo da je

$$s_0 = \frac{f[x_{-1}, x_0] + f[x_0, x_1]}{2}.$$

Ostaje nam samo još definirati što je $f[x_{-1}, x_0]$. Podijeljenu razliku $f[x_0, x_1]$ možemo interpretirati kao sredinu dvije susjedne podijeljene razlike, tj. možemo staviti

$$f[x_0, x_1] = \frac{f[x_{-1}, x_0] + f[x_1, x_2]}{2}.$$

Odatle slijedi da je

$$f[x_{-1}, x_0] = 2f[x_0, x_1] - f[x_1, x_2],$$

odnosno

$$s_0 = \frac{3f[x_0, x_1] - f[x_1, x_2]}{2}$$

i to je praktična formula za s_0. Na sličan način, možemo dobiti i relaciju za s_n

$$s_n = \frac{3f[x_{n-1}, x_n] - f[x_{n-2}, x_{n-1}]}{2}.$$

Akimin je algoritam dosta popularan u praksi i nalazi se u standardnim numeričkim paketima, poput IMSL-a, iako je točnost ovih formula za aproksimaciju derivacije relativno slaba. Općenito, za neekvidistantne točke, greška u derivaciji je reda veličine samo $O(h)$, a to znači samo $O(h^2)$ za funkcijske vrijednosti. Ako su točke ekvidistantne, onda je greška reda veličine $O(h^2)$ za derivaciju, a $O(h^3)$ za funkciju, tj. kao i kod Besselove po dijelovima kvazihermitske interpolacije.

Međutim, ova slabija točnost je potpuno u skladu s osnovnim ciljem Akimine aproksimacije derivacija. U mnogim primjenama, aproksimacijom želimo dobiti geometrijski ili vizuelno poželjan, graf aproksimacijske funkcije φ, pa makar i na uštrb točnosti. Tipičan primjer je (približno) crtanje grafova funkcija, gdje se iz nekog relativno malog broja zadanih podataka (točaka) treba, u kratkom vremenu, dobiti veliki broj točaka za crtanje vizuelno glatkog grafa. Iako nije nužno da nacrtani graf

baš interpolira zadane podatke (jer male, za oko nevidljive greške sigurno možemo tolerirati), interpolacija obično daje najbrži algoritam.

Ostaje još pitanje kako postići vizuelnu "glatkoću"? Očita heuristika je izbjegavanje naglih promjena u derivaciji. Drugim riječima, želimo "izgladiti" dobivene podatke za derivaciju, t. izračunate podijeljene razlike. Problem izglađivanja podataka je klasični problem numeričke analize. Jedan od najjednostavnijih i najbržih pristupa je zamjena podatka srednjom vrijednošću podataka preko nekoliko susjednih točaka. Ova ideja je vrlo bliska numeričkoj integraciji, jer integracija "izglađuje" funkciju, pa ćemo tamo dati precizniji opis i opravdanje numeričkog izglađivanja podataka.

Ako bolje pogledamo Akimine formule za aproksimaciju derivacije, one se svode na težinsko usrednjavanje podijeljenih razlika preko nekoliko susjednih točaka s ciljem izglađivanja derivacije (pa onda i funkcije). Vidimo da na s_k utječu točke $x_{k-2}, \ldots, x_{k+2}$, tj. usrednjavanje ide preko 5 susjednih točaka, osim na rubovima. Slično možemo interpretirati i Besselove formule. Tamo usrednjavanje ide preko 3 susjedne točke.

Aproksimacija derivacije mogla bi se napraviti još i bolje, ako povučemo interpolacijski polinom stupnja 3 koji prolazi točkama x_k, x_{k-1}, x_{k+1} i jednom od točaka x_{k-2} ili x_{k+2} (opet se javlja nesimetričnost, jer za kubični polinom trebamo 4 točke, pa s jedne strane od x_k uzimamo dvije, a s druge samo jednu točku) i njega deriviramo u x_k (uz pažljivo deriviranje na rubovima). Takvim postupkom možemo dobiti grešku u funkcijskoj vrijednosti $O(h^4)$. Primijetite da bolja aproksimacija derivacija nije potrebna, jer je greška kod po dijelovima Hermiteove kubične interpolacije također reda veličine $O(h^4)$.

Kvazihermiteova po dijelovima kubična interpolacija je također lokalna, tj. promjenom jedne točke promijenit će se samo nekoliko susjednih kubičnih polinoma. Točno koliko, ovisi o tome koju smo aproksimaciju derivacije izabrali.

7.3.6. Kubična splajn interpolacija

Brojeve $s_0, \ldots, s_n$ možemo odrediti na još jedan način. Umjesto da su skalari s_k neke aproksimacije derivacije funkcije f u čvorovima, možemo zahtijevati da se s_k biraju tako da funkcija φ bude još glađa — da joj je i druga derivacija neprekidna, tj. da je klase $C^2[a, b]$.

Nagibe $s_1, \ldots, s_{n-1}$ određujemo iz uvjeta neprekidnosti druge derivacije u unutarnjim čvorovima $x_1, \ldots x_{n-1}$. Takva se interpolacija zove (kubična) splajn interpolacija.

Možemo li iz tih uvjeta jednoznačno izračunati splajn? Prisjetimo se, imamo $4n$ koeficijenata kubičnih polinoma. Uvjeta interpolacije (svaki polinom mora inter-

polirati rubne točke svog podintervala) ima $2n$. Uvjeta ljepljenja prve derivacije u unutarnjim točkama ima $n-1$ jer je toliko unutarnjih točaka, a jednako je toliko i uvjeta ljepljenja druge derivacije.

Dakle, imamo ukupno $4n-2$ uvjeta, a moramo odrediti $4n$ koeficijenata. Odmah vidimo da nam nedostaju 2 uvjeta da bismo te koeficijente mogli odrediti. Kako se oni biraju, to ostavimo za kasnije. Za početak, prva derivacija se lijepi u unutarnjim točkama čim postavimo zahtjev da je $\varphi'(x_k) = s_k$ u tim točkama, bez obzira na to koliki je s_k i ima li on značenje aproksimacije derivacije (vidjeti početak odjeljka o po dijelovima kubičnoj interpolaciji). To nam omogućava da s_k-ove odredimo i na neki drugi način. Zbog toga, ostaje nam samo postaviti uvjete ljepljenja druge derivacije u unutarnjim čvorovima. Zahtjev je

$$p_k''(x_k) = p_{k+1}''(x_k), \quad k = 1, \ldots, n-1.$$

Ako polinome p_k pišemo u formi (7.3.3), relativno obzirom na početnu točku podintervala, tj. ako je

$$p_k(x) = c_{0,k} + c_{1,k}(x - x_{k-1}) + c_{2,k}(x - x_{k-1})^2 + c_{3,k}(x - x_{k-1})^3,$$

onda je

$$\begin{aligned} p_k''(x) &= 2c_{2,k} + 6c_{3,k}(x - x_{k-1}) \\ p_{k+1}''(x) &= 2c_{2,k+1} + 6c_{3,k+1}(x - x_k), \end{aligned}$$

pa je

$$\begin{aligned} p_k''(x_k) &= 2c_{2,k} + 6c_{3,k}(x_k - x_{k-1}) \\ p_{k+1}''(x_k) &= 2c_{2,k+1}. \end{aligned}$$

Drugim riječima, podijelimo li prethodne jednadžbe s 2, uvjet ljepljenja glasi

$$c_{2,k} + 3c_{3,k}(x_k - x_{k-1}) = c_{2,k+1}. \tag{7.3.5}$$

Ostaje samo izraziti koeficijente $c_{i,k}$ u terminima f_k i s_k iz relacija koje su dobivene iz uvjeta ljepljenja prvih derivacija. Ponovimo

$$\begin{aligned} c_{3,k} &= \frac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k^2}, \\ c_{2,k} &= \frac{f[x_{k-1}, x_k] - s_{k-1}}{h_k} - h_k c_{3,k}. \end{aligned}$$

Uvrštavanjem u (7.3.5), dobivamo

$$\begin{aligned} &\frac{f[x_{k-1}, x_k] - s_{k-1}}{h_k} + 2\,\frac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k} \\ &\qquad = \frac{f[x_k, x_{k+1}] - s_k}{h_{k+1}} - \frac{s_{k+1} + s_k - 2f[x_k, x_{k+1}]}{h_{k+1}}. \end{aligned}$$

Sređivanjem dobivamo

$$\frac{-3f[x_{k-1},x_k]+s_{k-1}+2s_k}{h_k}=\frac{3f[x_k,x_{k+1}]-2s_k-s_{k+1}}{h_{k+1}}.$$

Pomnožimo li prethodnu relaciju s h_kh_{k+1} i prebacimo li sve s_k na lijevu stranu, a članove koji nemaju s_k na desnu stranu, za $k=1,\ldots,n-1$, dobivamo

$$h_{k+1}s_{k-1}+2(h_k+h_{k+1})s_k+h_ks_{k+1}=3(h_{k+1}f[x_{k-1},x_k]+h_kf[x_k,x_{k+1}]).$$

Ovo je linearni sustav s $(n+1)$-om nepoznanicom i $(n-1)$-om jednadžbom. Ako na neki način zadamo rubne nagibe s_0 i s_n, onda ostaje točno $n-1$ nepoznanica.

Matrica tako dobivenog linearnog sustava je trodijagonalna

$$\begin{bmatrix} 2(h_1+h_2) & h_1 & & & & \\ h_3 & 2(h_2+h_3) & h_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & h_{n-1} & 2(h_{n-2}+h_{n-1}) & h_{n-2} & \\ & & & h_n & 2(h_{n-1}+h_n) & \end{bmatrix}$$

i strogo dijagonalno dominantna po retcima, jer za svako k vrijedi

$$2(h_k+h_{k+1})>h_k+h_{k+1}.$$

pa je i regularna (Geršgorin). Prema tome ovaj linearni sustav sigurno ima jedinstveno rješenje $s_1,\ldots,s_{n-1}$. Za rješavanje možemo koristiti Gaussove eliminacije ili LR faktorizaciju **bez** pivotiranja (vidjeti poglavlje o line arnim sustavima).

Primijetite, sada s_k nisu nezavisni, nego ovise jedan o drugom. To znači da aproksimacija više **nije lokalna**, jer se promjenom jedne funkcijeske vrijednosti mijenjaju **svi** polinomi. Preciznije, promjena jedne vrijednosti f_{k_0} mijenja desne strane u 3 jednadžbe (za k_0-1, k_0 i k_0+1), ali se zbog toga promijeni cijeli vektor rješenja sustava, tj. svi skalari s_k. Ipak, može se pokazati da su promjene lokalizirane — najviše se promijene s_k-ovi za k blizu k_0, a promjene padaju prema rubovima.

Posljednje otvoreno pitanje je kako možemo izabrati s_0 i s_n. Oni se ne moraju direktno zadati, već se uobičajeno zadaju rubni uvjeti na funkciju φ iz kojih se određuju s_0 i s_n ili se dodaju još dvije jednadžbe linearnog sustava (prva i zadnja).

Postoji nekoliko tradicionalnih načina zadavanja rubnih uvjeta, odnosno jednadžbi koje nedostaju.

(a) Potpuni splajn — zadana prva derivacija u rubovima

Ako je poznata derivacija funkcije f u rubovima, a to je, recimo slučaj kod rješavanja rubnih problema za običnu diferencijalnu jednadžbu, onda je prirodno zadati

$$s_0 = f'(x_0), \quad s_n = f'(x_n).$$

Takav oblik splajna se katkad zove potpuni ili kompletni splajn. Greška aproksimacije u funkcijskoj vrijednosti je $O(h^4)$.

(b) Zadana druga derivacija u rubovima

Ako je poznata druga derivacija funkcije f u rubovima, onda treba staviti

$$f''(x_0) = \varphi''(x_0) = p_1''(x_0), \quad f''(x_n) = \varphi''(x_n) = p_n''(x_n).$$

Ostaje još samo izraziti $p_1''(x_0)$ pomoću s_0, s_1 te $p_n''(x_n)$ pomoću s_{n-1} i s_n. Znamo da je

$$c_{2,1} = \frac{p_1''(x_0)}{2} = \frac{f''(x_0)}{2},$$

pa iz izraza za $c_{2,1}$ izlazi

$$\frac{3f[x_0, x_1] - 2s_0 - s_1}{h_1} = \frac{f''(x_0)}{2}.$$

Nakon sređivanja dobivamo jednadžbu

$$2s_0 + s_1 = 3f[x_0, x_1] - \frac{h_1}{2} f''(x_0),$$

koju treba dodati kao prvu jednadžbu linearnog sustava. Slično, korištenjem relacije

$$p_n''(x_n) = 2c_{2,n} + 6c_{3,n}h_n,$$

te uvrštavanjem izraza za $c_{2,n}$ i $c_{3,n}$ izlazi

$$s_{n-1} + 2s_n = 3f[x_{n-1}, x_n] + \frac{h_n}{2} f''(x_n).$$

Tu jednadžbu dodajemo kao zadnju u linearni sustav. Dobiveni linearni sustav ima $(n+1)$-u jednadžbu i isto toliko nepoznanica, a može se pokazati da ima i jedinstveno rješenje. Ponovno, greška aproksimacije u funkcijskoj vrijednosti je $O(h^4)$.

(c) Prirodni splajn — slobodni krajevi

Ako zadamo tzv. slobodne krajeve, tj. ako je

$$\varphi''(x_0) = \varphi''(x_n) = 0$$

dobivamo prirodnu splajn interpolaciju. Na isti način kao u slučaju (b), dobivamo dvije dodatne jednadžbe

$$2s_0 + s_1 = 3f[x_0, x_1], \quad s_{n-1} + 2s_n = 3f[x_{n-1}, x_n].$$

Ako aproksimirana funkcija f nema na rubu druge derivacije jednake 0, onda je greška aproksimacije u funkcijskoj vrijednosti $O(h^2)$, a ako ih ima, onda je (kao u slučaju (b)) greška reda $O(h^4)$.

(d) Numerička aproksimacija derivacija na rubu

Ako ništa ne znamo o ponašanju derivacije funkcije f na rubovima, bolje je ne zadavati njeno ponašanje.

Preostala dva parametra mogu se odrediti tako da numerički aproksimiramo φ' ili φ'' ili φ''' u rubovima. Prvo napišemo kubični interpolacijski polinom koji prolazi točkama $x_0, \ldots, x_3$, odnosno $x_{n-3}, \ldots, x_n$, a zatim analitički nađemo željenu derivaciju, koju koristimo aproksimaciju za odgovarajuću derivaciju funkcije. Bilo koja od ovih varijanti daje pogrešku reda $O(h^4)$.

(e) Not-a-knot splajn

Moguć je i drugačiji pristup. Umjesto neke aproksimacije derivacije, koristimo tzv. "not-a-knot" (prevedeno "nije čvor") uvjet. Parametre s_0 i s_n biramo tako da su prva dva i posljednja dva kubična polinoma jednaka, tj. da je

$$p_1 = p_2, \quad p_{n-1} = p_n.$$

Dakle, na "dvostrukom" intervalu $[x_0, x_2]$ povlačimo samo jedan kubični polinom p_1 koji interpolira funkcijske vrijednosti u x_0, x_1 i x_2 (u x_2 se "lijepe" još i prva i druga derivacija na sljedeći polinom). Slično vrijedi i za interval $[x_{n-2}, x_n]$.

Činjenicu da su prva dva, odnosno zadnja dva polinoma jednaka možemo izraziti i korištenjem treće derivacije, jer ako dva kubična polinoma u nekoj točki imaju istu funkcijsku vrijednost, vrijednost prve, druge i treće derivacije, oni su jednaki. Dakle, čvoru x_1 zalijepi (osim prve i druge) i treća derivacija polinoma p_1 i p_2, odnosno da se u čvoru x_{n-1} zalijepi se treća derivacija polinoma p_{n-1} i p_n. Te zahtjeve možemo pisati kao

$$p_1'''(x_1) = p_2'''(x_1), \quad p_{n-1}'''(x_{n-1}) = p_n'''(x_{n-1}).$$

Zahtjev $p_1'''(x_1) = p_2'''(x_1)$ znači da su vodeći koeficijenti polinoma p_1 i p_2 jednaki, tj.

$$c_{3,1} = c_{3,2}.$$

Pridružimo li taj zahtjev zahtjevu ljepljenja druge derivacije,

$$c_{2,1} + 3c_{3,1}h_k = c_{2,2},$$

dobivamo

$$\frac{f[x_0,x_1]-s_0}{h_1}+2\,\frac{s_1+s_0-2f[x_0,x_1]}{h_1}=\frac{f[x_1,x_2]-s_1}{h_2}-h_2\,\frac{s_1+s_0-2f[x_0,x_1]}{h_1^2}.$$

Sređivanjem, izlazi

$$h_2s_0+(h_1+h_2)s_1=\frac{(h_1+2(h_1+h_2))h_2\,f[x_0,x_1]+h_1^2\,f[x_1,x_2]}{h_1+h_2}.$$

Na sličan način dobivamo i zadnju jednadžbu

$$(h_{n-1}+h_n)s_{n-1}+h_{n-1}s_n=\frac{(h_n+2(h_{n-1}+h_n))h_{n-1}\,f[x_{n-1},x_n]+h_n^2\,f[x_{n-2},x_{n-1}]}{h_{n-1}+h_{n-2}}.$$

Kao i dosad, greška aproksimacije za funkcijske vrijednosti je $O(h^4)$.

Objasnimo još porijeklo naziva "not-a-knot" za ovaj tip određivanja dodatnih jednadžbi. Standardno, kubični splajn je klase $C^2[a,b]$, tj. funkcija φ ima neprekidne druge derivacije u unutarnjim čvorovima $x_1,\ldots,x_{n-1}$. Treća derivacija funkcije φ općenito "puca" u tim čvorovima, jer se treće derivacije polinoma p_k i p_{k+1} ne moraju zalijepiti u x_k, za $k=1,\ldots,n-1$. Kad uzmemo u obzir da su svi polinomi p_k kubični, onda je njihova treća derivacija ujedno i zadnja netrivijalna derivacija (sve više derivacije su nula). Dakle, zadnja netrivijalna derivacija splajna puca u unutarnjim čvorovima.

Ova činjenica, u terminologiji teorije splajn funkcija, odgovara tome da svi unutarnji čvorovi splajna imaju multiplicitet 1, jer je multiplicitet čvora jednak broju zadnjih derivacija koje pucaju ili mogu pucati u tom čvoru (derivacije se broje unatrag, počev od zadnje netrivijalne, koja odgovara stupnju polinoma). U tom smislu, povećanje glatkoće splajna u (unutarnjem) čvoru smanjuje multiplicitet tog čvora.

Prethodni zahtjev da se i zadnje netrivijalne derivacije splajna zalijepe u čvorovima x_1 i x_{n-1} odgovara tome da njihov multiplicitet više nije 1, nego 0. Čvorovi multipliciteta 0, naravno, nisu "pravi" čvorovi splajna, jer u njima nema pucanja derivacija (jednako kao i u svim ostalim točkama iz $[a,b]$ koje nisu čvorovi). Međutim, to **ne** znači da čvorove x_1 i x_{n-1} možemo izbaciti, jer u njima i dalje moraju biti zadovoljeni uvjeti interpolacije $\varphi(x_1)=f_1$ i $\varphi(x_{n-1})=f_{n-1}$. Dakle, te točke **ostaju** čvorovi interpolacije, iako nisu čvorovi splajna u smislu pucanja derivacija.

(f) Ostali rubni uvjeti

Svi dosad opisani načini zadavanja rubnih uvjeta "čuvaju" trodijagonalnu strukturu linearnog sustava za parametre s_k, jer eventualne dodatne jednadžbe prirodno dodajemo sustavu kao prvu i zadnju.

Za aproksimaciju periodičkih funkcija na intervalu koji odgovara periodu, ovakvi oblici zadavanja rubnih uvjeta nisu pogodni. Da bismo očuvali periodičnost, prirodno je postaviti tzv. periodičke rubne uvjete. U praksi se najčešće koristi zahtjev periodičnosti prve i druge derivacije na rubovima

$$\varphi'(x_0) = \varphi'(x_n), \quad \varphi''(x_0) = \varphi''(x_n),$$

što vodi na jednadžbe

$$p_1'(x_0) = p_n'(x_n), \quad p_1''(x_0) = p_n''(x_n).$$

Dobiveni linearni sustav više nije trodijagonalan. Za njegovo rješavanje postoje efikasni algoritmi (složenost ostaje linearna u n).

Zadatak 7.3.1 *Pokušajte naći jedan takav algoritam.*

U slučaju potrebe, dozvoljeno je i kombinirati razne oblike rubnih uvjeta u jednom i drugom rubu.

Primjer 7.3.2 *Neka je*

$$f(x) = \sin(\pi x).$$

Nađite prirodni splajn koji aproksimira funkciju f na $[0, 1]$ s čvorovima interpolacije $x_k = 0.2k$, za $k = 0, \ldots, 5$. Izračunajte vrijednost tog splajna u točki 0.55.

Budući da su točke ekvidistantne s razmakom $h = 0.2$, jednadžbe linearnog sustava za splajn su

$$hs_{k-1} + 4hs_k + hs_{k+1} = 3(hf[x_{k-1}, x_k] + hf[x_k, x_{k+1}]), \quad k = 1, \ldots, 4.$$

Dodatne jednadžbe (prva i zadnja) za prirodni splajn su

$$2s_0 + s_1 = 3f[x_0, x_1]$$
$$s_4 + 2s_5 = 3f[x_4, x_5].$$

Za desnu stranu sustava trebamo izračunati prve podijeljene razlike

x_k	f_k	$f[x_k, x_{k+1}]$
0.0	0.0000000000	
		2.9389262615
0.2	0.5877852523	
		1.8163563200
0.4	0.9510565163	
		0.0000000000
0.6	0.9510565163	
		−1.8163563200
0.8	0.5877852523	
		−2.9389262615
1.0	0.0000000000	

Iz svih ovih podataka dobivamo linearni sustav

$$\begin{bmatrix} 0.4 & 0.2 & & & & \\ 0.2 & 0.8 & 0.2 & & & \\ & 0.2 & 0.8 & 0.2 & & \\ & & 0.2 & 0.8 & 0.2 & \\ & & & 0.2 & 0.8 & 0.2 \\ & & & & 0.2 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix} = \begin{bmatrix} 1.7633557569 \\ 2.8531695489 \\ 1.0898137920 \\ -1.0898137920 \\ -2.8531695489 \\ -1.7633557569 \end{bmatrix}$$

Rješenje tog linearnog sustava za "nagibe" s_k je

$$\begin{aligned} s_0 &= -s_5 = 3.1387417029, \\ s_1 &= -s_4 = 2.5392953786, \\ s_2 &= -s_3 = 0.9699245271. \end{aligned}$$

Budući da se točka $x = 0.55$ nalazi u intervalu $[x_2, x_3] = [0.4, 0.6]$, restrikcija splajna na taj interval je polinom p_3, kojeg nalazimo iz tablice podijeljenih razlika

x_k	f_k	$f^{[1]}[x_k]$	$f^{[2]}[x_k]$	$f^{[3]}[x_k]$
0.4	0.9510565163			
		0.9699245271		
0.4	0.9510565163		-4.8496226357	
		0.0000000000		0.0000000000
0.6	0.9510565163		-4.8496226357	
		-0.9699245271		
0.6	0.9510565163			

Odavde odmah slijedi da je taj kubični polinom jednak

$$p_3(x) = 0.9510565163 + 0.9699245271(x - 0.4) - 4.8496226357(x - 0.4)^2,$$

tj. p_3 je zapravo kvadratni polinom.

Pogledajmo još aproksimacije za funkciju, prvu i drugu derivaciju u točki 0.55.

	funkcija $j = 0$	prva derivacija $j = 1$	druga derivacija $j = 2$
$f^{(j)}(0.55)$	0.9876883406	-0.4914533661	-9.7480931932
$\varphi^{(j)}(0.55)$	0.9874286861	-0.4849622636	-9.6992452715
greška	0.0002596545	-0.0064911026	-0.0488479218

Vidimo da su aproksimacije vrlo točne, iako je h relativno velik. To je zato što funkcija $f(x) = \sin(\pi x)$ zadovoljava prirodne rubne uvjete $f''(0) = f''(1) = 0$, kao i prirodni splajn. Greška aproksimacije funkcije je reda veličine $O(h^4)$, prve derivacije $O(h^3)$, a druge derivacije $O(h^2)$.

7.4. Interpolacija polinomnim splajnovima — za matematičare

Iskustvo s polinomnom interpolacijom ukazuje da polinomi imaju dobra lokalna svojstva aproksimacije, ali da globalna uniformna pogreška može biti vrlo velika. Niti posebnim izborom čvorova interpolacije ne možemo ukloniti taj fenomen. Nameće se prirodna ideja da izbjegavamo visoke stupnjeve polinoma, ali da konstruiramo polinome niskog stupnja na nekoj subdiviziji segmenta od interesa, tj. da razmotrimo **po dijelovima polinomnu interpolaciju**.

Ako je funkcija koju želimo interpolirati glatka, želimo sačuvati što je moguće veću glatkoću takvog interpolanta. To nas vodi na zahtijev da za po dijelovima linearne aproksimacije zahtijevamo globalnu neprekidnost, za po dijelovima parabolične aproksimacije globalnu diferencijabilnost, itd. Po dijelovima polinomne funkcije koje zadovoljavaju zadane uvjete glatkoće zovemo **polinomne splajn funkcije**. Koeficijente u nekoj reprezentaciji polinomnog splajna odredit ćemo iz uvjeta interpolacije, kao i u slučaju polinomne interpolacije. Takav specijalni izbor splajna zove se **interpolacijski polinomni splajn**.

U sljedeća dva odjeljka istražit ćemo konstrukciju i svojstva aproksimacije linearnog i kubičnog splajna. Dok je za linearni splajn očito moguće zahtijevati najviše neprekidnost na cijelom segmentu od interesa (zahtijev za "lijepljenjem" prve derivacije vodi na funkciju koja je globalno linearna), za kubične je splajnove moguće zahtijevati pripadnost prostorima C^1 ili C^2, tj. moguće je naći dva kubična splajna.

Splajnovi parnog stupnja mogu biti problematični, kao što pokazuje sljedeća intuitivna diskusija. Zamislimo da je segment od interesa za interpolaciju $[a, b]$, i neka je neka njegova subdivizija (podjela na podintervale) zadana mrežom čvorova

$$a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b. \tag{7.4.1}$$

Paraboličkи splajn S_2 mora biti polinom stupnja najviše 2 (parabola) na svakom intervalu subdivizije, tj. imamo po 3 nepoznata parametra (koeficijenti polinoma stupnja 2) na svakom intervalu. Ukupno dakle treba naći $3N$ slobodnih parametara.

Zahtijev da vrijedi $S_2 \in C^1[a, b]$ vezuje $2(N-1)$ od tih parametara (neprekidnost S_2 i S_2' u $N-1$ unutrašnjih čvorova $x_1, \ldots, x_{N-1}$). Ostaju dakle $N+2$ slobodna parametra. Zahtijevamo li da S_2 bude interpolacijski, tj. da vrijedi

$$S_2(x_i) = f_i, \quad i = 0, \ldots, N,$$

ostaje slobodan samo jedan parametar. Taj bismo parametar mogli odrediti dodavanjem još jednog čvora interpolacije, ili nekim dodatnim uvjetom na rubu cijelog intervala — recimo, zadavanjem derivacije. Međutim, jasno je da se taj parametar ne može odrediti **simetrično** iz podataka. To je problem i s ostalim splajn interpolantima parnog stupnja.

Zadatak 7.4.1 *Nađite što je u gornjoj diskusiji neformalno, i što je potrebno za precizan matematički dokaz. Ako je prostor polinomnih splajnova $\mathcal{S}(n)$ stupnja n definiran zahtjevima:*

(1) $s \in \mathcal{S}(n) \Longrightarrow s|_{[x_i,x_{i+1}]} \in \mathcal{P}_n$ *($\mathcal{P}_n$ je prostor polinoma stupnja n);*

(2) $s \in C^{n-1}[x_0, x_N]$,

pokažite da je $\mathcal{S}(n)$ vektorski prostor, i dokažite da je $\dim \mathcal{S}(n) = n + N$.

7.4.1. Linearni splajn

Najjednostavniji **linearni interpolacijski splajn** S_1 određen je uvjetom globalne neprekidnosti i uvjetom interpolacije

$$S_1(x_i) = f_i, \quad i = 0, \ldots, N,$$

na mreži čvorova — subdiviziji segmenta $[a, b]$ zadanoj s (7.4.1). Očito imamo

$$S_1(x) = f_i \frac{x_{i+1} - x}{h_i} + f_{i+1} \frac{x - x_i}{h_i} = f_i + \frac{x - x_i}{h_i}(f_{i+1} - f_i), \quad x \in [x_i, x_{i+1}],$$

gdje je $h_i = x_{i+1} - x_i$, za $i = 0, \ldots, N - 1$. Algoritam za računanje je trivijalan, pa možemo odmah ispitati pogrešku, odnosno, razmotriti svojstva interpolacijskog linearnog splajna obzirom na glatkoću funkcije koja se interpolira, u raznim normama koje se koriste za aproksimaciju. U dokazima ćemo koristiti jednu sporednu lemu.

Lema 7.4.1 *Ako je $f \in C[a, b]$ i α, β imaju isti znak, tada postoji $\xi \in [a, b]$ tako da vrijedi*

$$\alpha f(a) + \beta f(b) = (\alpha + \beta) f(\xi).$$

Dokaz. Ako je $f(a) = f(b)$ tvrdnja je očigledna, jer možemo uzeti $\xi = a$ ili $\xi = b$. Ako je $f(a) \neq f(b)$, tada funkcija $\psi(x) = \alpha f(a) + \beta f(b) - (\alpha + \beta) f(x)$ poprima suprotne predznake na krajevima intervala, pa zbog neprekidnosti postoji $\xi \in (a, b)$ tako da je $\psi(\xi) = 0$. Tvrdnja vrijedi i ako je $\alpha = 0$, uz $\xi = b$, odnosno, $\beta = 0$, uz $\xi = a$. ■

Za precizno određivanje reda konvergencije aproksimacija neprekidne funkcije f zgodno je uvesti oznake

$$\omega_i(f) = \max_{x', x'' \in [x_i, x_{i+1}]} |f(x'') - f(x')|, \quad i = 0, \ldots, N - 1,$$
$$\omega(f) = \max_{0 \leq i \leq N-1} \omega_i(f).$$

Vrijednost $\omega_i(f)$ zovemo **oscilacija** funkcije f na podintervalu $[x_i, x_{i+1}]$, a $\omega(f)$ je (očito) najveća oscilacija po svim podintervalima mreže.

Uočite da glatkoća funkcije f nije potrebna u definiciji ovih veličina, pa ih koristimo za ocjenu greške u slučaju da je f samo neprekidna, ali ne i derivabilna funkcija. Isto vrijedi i za zadnju (najvišu) **neprekidnu** derivaciju funkcije.

Također, kod ocjene grešaka, zgodno je uvesti skraćenu oznaku D za operator deriviranja funkcije f jedne varijable, kad je iz konteksta očito po kojoj varijabli se derivira. Onda n-tu derivaciju funkcije f u točki x možemo pisati u bilo kojem od sljedeća tri oblika

$$D^n f(x) = \frac{d^n}{dx^n} f(x) = f^{(n)}(x).$$

Pokazuje se da je prvi oblik najpregledniji u zapisu nekih dugačkih formula.

Da bismo olakšali razumijevanje teorema o ocjenama greške splajn interpolacije koji slijede, objasnimo odmah osnovnu ideju za uvođenje oznake $\omega(f)$ i njezinu ulogu u ocjeni greške. Jednostavno rečeno, $\omega(f)$ služi tome da napravimo finu razliku između ograničenosti i neprekidnosti funkcije f na nekom intervalu. Za dobivanje korisnih ocjena, obično, uz ograničenost, pretpostavljamo još i integrabilnost funkcije. Neprekidnost je, očito, jače svojstvo.

Za ilustraciju, uzmimo da je f derivabilna funkcija na $[a, b]$. Onda je prva derivacija Df i ograničena funkcija na $[a, b]$, čim postoji derivacija u svakoj točki segmenta, s tim da uzimamo jednostrane derivacije (limese) u rubovima. Drugim riječima, postoji njezina ∞–norma

$$\|Df\|_\infty = \sup_{x\in[a,b]} |Df(x)|.$$

Ako je prva derivacija i integrabilna, to označavamo s $f \in L^1_\infty[a, b]$. Gornji indeks 1 označava da je riječ o prvoj derivaciji funkcije f, a donji indeks ∞ označava ograničenost (preciznija definicija prostora $L^1_\infty[a, b]$ zahtijeva teoriju mjere i integrala). Naravno, prva derivacija **ne mora** biti neprekidna na $[a, b]$, da bi bila integrabilna. Ako je Df i neprekidna, onda je $f \in C^1[a, b]$ (oznaka koju smo odavno koristili).

Jedan od rezultata koje želimo dobiti ocjenom greške je uniformna konvergencija splajn interpolacije kad povećavamo broj čvorova, tj. "profinjujemo" mrežu (barem uz neke blage uvjete). Za uniformnu konvergenciju, očito, treba promatrati maksimalnu grešku na cijelom intervalu, tj. zanimaju nas tzv. uniformne ocjene — u ∞–normi. Iz iskustva polinomne interpolacije, jasno je da moramo iskoristiti **lokalno** ponašanje funkcije i splajn interpolacije na podintervalima mreže.

Kako ćemo lokalnost dobro ugraditi u ocjenu greške? S jedne strane, kvaliteta ocjene mora ovisiti o svojstvima (glatkoći) funkcije koju aproksimiramo (interpoliramo). Dakle, trebamo dobru globalnu mjeru lokalnog ponašanja funkcije. Za ograničene (integrabilne) funkcije koristimo ∞–normu na $[a, b]$, koja, očito, postoji. Nažalost, lokalnost tu ne pomaže, jer maksimum normi po podintervalima daje upravo normu na cijelom intervalu. Neprekidna funkcija je, naravno, i ograničena i

integrabilna. Međutim, za neprekidne funkcije, $\omega(f)$ daje bitno precizniju uniformnu ocjenu greške od globalne norme, jer uključuje lokalno ponašanje po podintervalima — najveća lokalna oscilacija može biti mnogo manja od globalne oscilacije na cijelom intervalu!

S druge strane, ocjena greške mora uključvati ovisnost o nekoj veličini koja mjeri "gustoću" mreže, odnosno razmak čvorova. Ako profinjavanjem mreže želimo dobiti konvergenciju, odmah je jasno da to profinjavanje mora biti "ravnomjerno" u cijelom $[a, b]$, tj. maksimalni razmak susjednih čvorova

$$\bar{h} := \max_{0 \le i \le N-1} h_i$$

mora težiti prema nuli. Da bismo izbjegli ovisnost o svim h_i, standardno se ocjene greške izražavaju upravo u terminima veličine $\bar{h}$, koja se još zove i **dijametar mreže**.

Teorem 7.4.1 (Uniformna ocjena pogreške linearnog splajna)

Neka je $S_1(x)$ linearni interpolacijski splajn za funkciju f. Obzirom na svojstva glatkoće funkcije f vrijedi:

(1) *ako je $f \in C[a, b]$ tada je*

$$\|S_1(x) - f(x)\|_\infty \le \omega(f);$$

(2) *ako je $f \in L^1_\infty[a, b]$ tada je*

$$\|S_1(x) - f(x)\|_\infty \le \frac{\bar{h}}{2} \|Df\|_\infty;$$

(3) *ako je $f \in C[a, b] \bigcap_{i=0}^{N-1} C^1[x_i, x_{i+1}]$ tada je*

$$\|S_1(x) - f(x)\|_\infty \le \frac{\bar{h}}{4} \omega(Df);$$

(4) *ako je $f \in C[a, b] \bigcap_{i=0}^{N-1} L^2_\infty[x_i, x_{i+1}]$ tada je*

$$\|S_1(x) - f(x)\|_\infty \le \frac{\bar{h}^2}{8} \|D^2 f\|_\infty.$$

Dokaz. Neka je $t := (x - x_i)/h_i$. Prema (7.4.1) greška je

$$E(x) := S_1(x) - f(x) = (1 - t)f_i + tf_{i+1} - f(x), \quad x \in [x_i, x_{i+1}]. \tag{7.4.2}$$

Uočimo da je $x \in [x_i, x_{i+1}]$ ekvivalentno s $t \in [0, 1]$, pa $(1-t)$ i t imaju isti (pozitivni) predznak, ili je jedan od njih jednak nula.

Ako je $f \in C[a,b]$, onda prema Lemi 7.4.1 postoji $\xi \in [x_i, x_{i+1}]$ takav da vrijedi $E(x) = f(\xi) - f(x)$, pa je $|E(x)| \le \omega_i(f) \le \omega(f)$.

Ako je prva derivacija ograničena i integrabilna, vrijedi

$$f_i = f(x) + \int_x^{x_i} Df(v)\,dv, \quad f_{i+1} = f(x) + \int_x^{x_{i+1}} Df(v)\,dv.$$

Supstitucijom u (7.4.2) dobijemo

$$E(x) = -(1-t)\int_{x_i}^{x} Df(v)\,dv + t\int_x^{x_{i+1}} Df(v)\,dv$$

i

$$|E(x)| \le (1-t)\int_{x_i}^{x} |Df(v)|\,dv + t\int_x^{x_{i+1}} |Df(v)|\,dv.$$

Slijedi

$$|E(x)| \le \left[(1-t)\int_{x_i}^{x} dv + t\int_x^{x_{i+1}} dv\right] \|Df\|_\infty = 2t(1-t)\,h_i\,\|Df\|_\infty.$$

Kako parabola $2t(1-t)$ ima maksimum $1/2$ u $t = 1/2$, dokazali smo da vrijedi

$$|E(x)| \le \frac{1}{2}\,\bar{h}\,\|Df\|_\infty.$$

Neka je sada $f \in C[a,b]$ klase C^1 na svakom podintervalu mreže (eventualni prekidi prve derivacije mogu biti samo u čvorovima mreže). Prema Taylorovoj formuli s Lagrangeovim oblikom ostatka

$$f_i = f(x) - t\,h_i\,Df(\xi), \quad f_{i+1} = f(x) + (1-t)\,h_i\,Df(\eta), \quad \xi, \eta \in [x_i, x_{i+1}].$$

Supstitucijom u (7.4.2) dobijemo

$$E(x) = t(1-t)\,h_i\,(Df(\eta) - Df(\xi)),$$

odakle slijedi

$$|E(x)| \le t(1-t)\,h_i\,\omega_i(Df),$$

pa opet ocjenom parabole na desnoj strani dobijemo da vrijedi

$$|E(x)| \le \frac{1}{4}\,\bar{h}\,\omega(Df).$$

Na kraju, ako f ima na svakom podintervalu ograničenu i integrabilnu drugu derivaciju, tada vrijedi Taylorova formula s integralnim oblikom ostatka

$$f_i = f(x) - t\,h_i\,Df(x) + \int_x^{x_i} (x_i - v)\,D^2 f(v)\,dv$$

$$f_{i+1} = f(x) + (1-t)\,h_i\,Df(x) + \int_x^{x_{i+1}} (x_{i+1} - v)\,D^2 f(v)\,dv,$$

pa iz formule (7.4.2) slijedi

$$E(x) = (1-t)\int_x^{x_i} (x_i - v)\,D^2 f(v)\,dv + t\int_x^{x_{i+1}} (x_{i+1} - v)\,D^2 f(v)\,dv.$$

Odavde lako slijedi

$$|E(x)| \le \frac{1}{2}\,h_i^2\,t(1-t)\,\|D^2 f\|_\infty \le \frac{1}{8}\,\bar{h}^2\,\|D^2 f\|_\infty.$$

■

Zadatak 7.4.2 *Dokažite da se u slučajevima* (3) *i* (4) *teorema* 7.4.1 *može ocijeniti i greška u derivaciji, točnije, da vrijedi:*

(3) $\|DS_1(x) - Df(x)\|_\infty \le \omega(Df)$;

(4) $\|DS_1(x) - Df(x)\|_\infty \le \dfrac{\bar{h}}{2}\,\|D^2 f\|_\infty$.

Teoremi poput teorema 7.4.1 pripadaju grupi teorema koji se nazivaju **direktni teoremi teorije aproksimacija**. Iako u daljnjem nećemo slijediti ovaj pristup do krajnjih detalja, primijetimo da se prirodno pojavljuju dva važna pitanja.

(1) Da li su navedene ocjene najbolje moguće, tj. da li smo zbog tehnike dokazivanja napravili na nekom mjestu pregrubu ocjenu, iskoristili nedovoljno "finu" nejednakost, pa zapravo možemo dobiti bolji red konvergencije? Da li su i konstante u ocjeni greške najbolje moguće?

(2) Ako dalje povećavamo glatkoću funkcije koja se interpolira, možemo li dobiti sve bolje i bolje ocjene za grešku, na primjer, u slučaju linearnog splajna, ocjene s h^2, h^3, i tako redom?

Teoremi koji se bave problematikom kao u (2) zovu se **inverzni teoremi teorije aproksimacija**. U većini slučajeva to su iskazi tipa "red aproksimacije naveden u direktnom teoremu je najbolji mogući". Doista, da nije tako, trebalo bi dopuniti ili popraviti direktni teorem! Ocjena optimalnosti konstanti je neugodan problem, koji

za opći stupanj splajna nije riješen — treba konstruirati primjer funkcije na kojoj se dostiže konstanta iz direktnog teorema.

Slični su i tzv. **teoremi zasićenja teorije aproksimacija**, koji pokušavaju odgovoriti na drugo pitanje: može li se bolje aproksimirati funkcija ako su pretpostavke na glatkoću jače? I ovi teoremi su u principu negativnog karaktera — na primjer, za linearni splajn možemo staviti da je $f \in C^\infty[a,b]$, ali red aproksimacije će ostati h^2. Sam prostor u kojem se aproksimira jednostavno ne može točnije reproducirati funkciju koja se aproksimira, nedostaje mu "snage aproksimacije". Iako se u daljnjem nećemo baviti općim teoremima aproksimacije, svi direktni teoremi koji slijede optimalni su u smislu postojanja odgovarajućih inverznih teorema i teorema zasićenja.

Zadatak 7.4.3 *Pokažite da u slučaju* (4) *teorema* 7.4.1 *funkcija* $f(x) = x^2$ *igra ulogu* **ekstremale**, *tj. da vrijedi "=" umjesto "≤", pa je ocjena ujedno i najbolja moguća.*

Zadatak 7.4.4 *Ako je* $f \in C[a,b] \bigcap_{i=0}^{N-1} C^3[x_i, x_{i+1}]$, *tada vrijedi*

$$DS\Big(x_i + \frac{h_i}{2}\Big) = Df\Big(x_i + \frac{h_i}{2}\Big) + O(h_i^2), \quad i = 0, \ldots, N-1.$$

Iz toga možemo zaključiti da red aproksimacije derivacije u specijalno izabranim točkama može biti i **više** *od optimalnog; to je efekt* **superkonvergencije**.

7.4.2. Hermiteov kubični splajn

Kao i u slučaju Hermiteove interpolacije polinomima, možemo razmatrati i Hermiteovu interpolaciju splajn funkcijama. Ako preskočimo paraboličke splajnove (v. raniju diskusiju), prvi je netrivijalni slučaj po dijelovima kubičnih splajnova s globalno neprekidnom derivacijom.

Definicija 7.4.1 *Neka su u čvorovima* $a = x_0 < x_1 < \cdots < x_N = b$ *zadane vrijednosti* f_i, f_i', *za* $i = 0, \ldots, N$. *Hermiteov interpolacijski kubični splajn je funkcija* $H \in C^1[a,b]$ *koja zadovoljava*

(1) $H(x) = a_{i0} + a_{i1}(x - x_i) + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3$, *za svaki* $x \in [x_i, x_{i+1}]$;

(2) $H(x_i) = f_i$, $DH(x_i) = f_i'$, *za* $i = 0, \ldots, N$.

Koristeći Hermiteovu bazu iz teorema 7.2.4 na svakom podintervalu mreže $[x_i, x_{i+1}]$, lagano vidimo da vrijedi

$$H(x) = \varphi_1(t) f_i + \varphi_2(t) f_{i+1} + \varphi_3(t) h_i f_i' + \varphi_4(t) h_i f_{i+1}', \quad t = \frac{x - x_i}{h_i}, \qquad (7.4.3)$$

gdje je

$$\varphi_1(t) = (1-t)^2(1+2t), \qquad \varphi_2(t) = t^2(3-2t),$$
$$\varphi_3(t) = t(1-t^2), \qquad \varphi_4(t) = -t^2(1-t).$$

Napomenimo još samo da kod računanja treba prvo izračunati koeficijente A_i i B_i formulama

$$\begin{aligned} A_i &= -2\,\frac{f_{i+1}-f_i}{h_i} + (f_i' + f_{i+1}'), \\ B_i &= A_i + \frac{f_{i+1}-f_i}{h_i} - f_i', \end{aligned} \qquad \text{za} \quad i = 0, \ldots, N-1, \tag{7.4.4}$$

i zapamtiti ih. Za zadanu točku $x \in [x_i, x_{i+1}]$, Hermiteov splajn računamo formulom

$$H(x) = f_i + (th_i)\,[f_i' + t(B_i + tA_i)]. \tag{7.4.5}$$

Obzirom na činjenicu da su nam derivacije f_i' najčešće nepoznate, preostaje nam samo da ih aproksimiramo iz zadanih vrijednosti funkcije. To je problem **približne Hermiteove interpolacije**, i tada ne možemo više očekivati isti red konvergencije. Vrijednost Hermiteove interpolacije je, međutim, više teorijska nego praktična, kao što ćemo vidjeti kasnije. U tom smislu koristit ćemo sljedeći direktni teorem.

Teorem 7.4.2 *Za Hermiteov kubični splajn, ovisno o glatkoći funkcije f, vrijede sljedeće uniformne ocjene pogreške:*

(1) *ako je $f \in C^1[a,b]$ tada je*

$$\|H(x) - f(x)\|_\infty \le \frac{3}{8}\,\bar{h}\,\omega(Df);$$

(2) *ako je $f \in L_\infty^2[a,b]$ tada je*

$$\|H(x) - f(x)\|_\infty \le \frac{1}{16}\,\bar{h}^2\,\|D^2 f\|_\infty;$$

(3) *ako je $f \in C^1[a,b] \bigcap_{i=0}^{N-1} C^2[x_i, x_{i+1}]$ tada je*

$$\|H(x) - f(x)\|_\infty \le \frac{1}{32}\,\bar{h}^2\,\omega(D^2 f);$$

(4) *ako je $f \in C^1[a,b] \bigcap_{i=0}^{N-1} L_\infty^3[x_i, x_{i+1}]$ tada je*

$$\|H(x) - f(x)\|_\infty \le \frac{1}{96}\,\bar{h}^3\,\|D^3 f\|_\infty;$$

(5) *ako je* $f \in C^1[a,b] \bigcap_{i=0}^{N-1} C^3[x_i, x_{i+1}]$ *tada je*

$$\|H(x) - f(x)\|_\infty \leq \frac{1}{192}\,\bar{h}^3\,\omega(D^3 f);$$

(6) *ako je* $f \in C^1[a,b] \bigcap_{i=0}^{N-1} L^4_\infty[x_i, x_{i+1}]$ *tada je*

$$\|H(x) - f(x)\|_\infty \leq \frac{1}{384}\,\bar{h}^4\,\|D^4 f\|_\infty.$$

Dokaz. U svim slučajevima treba analizirati grešku

$$E(x) := H(x) - f(x) = f_i\varphi_1(t) + f_{i+1}\varphi_2(t) + h_i f_i'\varphi_3(t) + h_i f_{i+1}'\varphi_4(t) - f(x).$$

Ako f_i, f_{i+1} zamijenimo njihovim Taylorovim razvojem oko točke $x = x_i + th_i$ s ostatkom u Lagrangeovom obliku, dobijemo

$$E(x) = h_i\,[(1-t)\,\varphi_2(t)\,Df(\xi) - t\,\varphi_1(t)\,Df(\eta) + \varphi_3(t) f_i' + \varphi_4(t) f_{i+1}'].$$

U daljnjem oznake ξ, η, ... označavaju točke u $[x_i, x_{i+1}]$. Prema Lemi 7.4.1 možemo grupirati članove istog znaka (prvi i treći, drugi i četvrti), pa dobijemo

$$E(x) = h_i\,t(1-t)(1+2t-2t^2)\,[Df(\bar{\xi}) - Df(\bar{\eta})],$$

odakle slijedi ocjena greške po točkama

$$|E(x)| \leq h_i\,t(1-t)(1+2t-2t^2)\,\omega_i(Df).$$

Odavde odmah slijedi tvrdnja (1), uzimanjem maksimuma polinoma u varijabli t.

Ako f ima drugu derivaciju ograničenu i integrabilnu, razvijemo opet f_i, f_i', f_{i+1}, f_{i+1}' oko točke x, ali koristeći Taylorovu formulu s integralnim oblikom ostatka. Nakon kraćeg računa dobijemo integralnu reprezentaciju greške

$$\begin{aligned} E(x) = &\int_{x_i}^{x} (1-t)^2\,[-th_i + (1+2t)(v-x_i)]\,D^2 f(v)\,dv \\ &+ \int_{x}^{x_{i+1}} t^2\,[-(1-t)h_i + (3-2t)(x_{i+1}-v)]\,D^2 f(v)\,dv. \end{aligned}$$

Zamjenom varijable $v - x_i = \tau h_i$ dobivamo

$$E(x) = h_i^2 \left\{ \int_0^t \psi_1(t,\tau)\,D^2 f(x_i + \tau h_i)\,d\tau + \int_t^1 \psi_2(t,\tau)\,D^2 f(x_i + \tau h_i)\,d\tau \right\}, \quad (7.4.6)$$

gdje je

$$\begin{aligned} \psi_1(t,\tau) &= (1-t)^2\,[(1+2t)\tau - t], \\ \psi_2(t,\tau) &= t^2\,[(3-2t)(1-\tau) - (1-t)]. \end{aligned}$$

Ne možemo upotrijebiti teorem o srednjoj vrijednosti za integrale, jer $\psi_1(t,\tau)$ mijenja znak; točnije $\psi_1(t,\tau^*)=0$ za $\tau^*=t/(1+2t)$. Međutim, $[0,t]=[0,\tau^*)\cup[\tau^*,t]$, a na svakom od podintervala ψ_1 je konstantnog znaka, pa teorem srednje vrijednosti za integrale možemo upotrijebiti po dijelovima.

$$\int_0^t \psi_1(t,\tau)\,D^2f(x_i+\tau h_i)\,d\tau = D^2f(\xi)\int_0^{\tau^*}\psi_1(t,\tau)\,d\tau + D^2f(\eta)\int_{\tau^*}^t \psi_1(t,\tau)\,d\tau$$
$$= \frac{t^2(1-t)^2}{2(1+2t)}\,\{4t^2\,D^2f(\eta) - D^2f(\xi)\}.$$

Analogno

$$\int_0^t \psi_2(t,\tau)\,D^2f(x_i+\tau h_i)\,d\tau = \frac{(1-t)^2t^2}{2(3-2t)}\,\{4(1-t^2)\,D^2f(\bar\xi) - D^2f(\bar\eta)\}.$$

Iz (7.4.6) dobivamo

$$E(x) = \frac{h_i^2\,t^2(1-t)^2}{2[3+4t(1-t)]}\,\{4t^2(3-2t)\,D^2f(\eta) - (3-2t)\,D^2f(\xi)$$
$$+ 4(1-t^2)(1+2t)\,D^2f(\bar\xi) - (1+2t)\,D^2f(\bar\eta)\}.$$

Primijenimo li lemu 7.4.1 na neprekidne funkcije istog znaka, dobivamo ocjenu

$$|E(x)| \le \frac{2h_i^2\,t^2(1-t)^2}{3+4t(1-t)}\,\omega_i(D^2f).$$

Maksimalna vrijednost desne strane postiže se za $t=1/2$, odakle slijedi

$$|E(x)| \le \frac{1}{32}\,h_i^2\,\omega_i(D^2f),$$

što dokazuje ocjenu (3). Ocjena (2) proizlazi lagano iz iste ocjene greške po točkama.

Ako je f po dijelovima klase C^3, slično dobivamo

$$E(x) = h_i^3\left\{\int_0^t \psi_1(t,\tau)\,D^3f(x_i+\tau h_i)\,d\tau + \int_t^1 \psi_2(t,\tau)\,D^3f(x_i+\tau h_i)\,d\tau\right\},$$

gdje su sada

$$\psi_1(t,\tau) = (1-t)^2\tau\Big[t - \frac{(1+2t)\tau}{2}\Big],$$
$$\psi_2(t,\tau) = t^2(1-\tau)\Big[-(1-t) + \frac{(3-2t)(1-\tau)}{2}\Big].$$

Zbog simetrije, dovoljno je razmatrati $t \in [0, 1/2]$, pa slijedi

$$|E(x)| \leq \frac{2}{3}\, h_i^3\, \frac{t^2(1-t)^3}{(3-2t)^2}\, \omega_i(D^3 f).$$

Odavde slijedi ocjena greške za $\|E(x)\|_\infty$. Maksimalna greška je u $x_i + h_i/2$, tj. za $t = 1/2$. Slično slijedi i ocjena (4).

Na kraju, ako f ima ograničenu i integrabilnu četvrtu derivaciju na svakom podintervalu, tada je

$$E(x) = \frac{1}{6}\, h_i^4 \left\{ \int_0^t \psi_1(t,\tau)\, D^4 f(x_i + \tau h_i)\, d\tau + \int_t^1 \psi_2(t,\tau)\, D^4 f(x_i + \tau h_i)\, d\tau \right\},$$

gdje su

$$\psi_1(t,\tau) = (1-t)^2\tau^2\,[-3t + (1+2t)\tau],$$
$$\psi_2(t,\tau) = t^2(1-\tau)^2\,[-3(1-t) + (3-2t)(1-\tau)],$$

pa zaključujemo da vrijedi

$$|E(x)| \leq \frac{t^2(1-t)^2}{4!}\, h_i^4\, \|D^4 f\|_\infty, \quad t \in [0,1]. \tag{7.4.7}$$

Odavde se lagano dobije ocjena za $\|E(x)\|_\infty$. ■

Zadatak 7.4.5 *Pokažite da za $f \in C^1[a,b] \bigcap_{i=0}^{N-1} L_\infty^4[x_i, x_{i+1}]$ (slučaj (6) iz prethodnog teorema) vrijede sljedeće ocjene za derivacije:*

$$\|DH(x) - Df(x)\|_\infty \leq \frac{\sqrt{3}}{216}\, \bar{h}^3\, \|D^4 f\|_\infty,$$
$$\|D^2 H(x) - D^2 f(x)\|_\infty \leq \frac{1}{12}\, \bar{h}^2\, \|D^4 f\|_\infty,$$
$$\|D^3 H(x) - D^3 f(x)\|_\infty \leq \frac{1}{2}\, \bar{h}\, \|D^4 f\|_\infty.$$

Uputa: Treba derivirati integralnu reprezentaciju za $E(x)$, tj. naći integralnu reprezentaciju za $D^k E(x)$, $k = 1, 2, 3$.

Zadatak 7.4.6 *Pokušajte za prvih pet slučajeva (klasa glatkoće funkcije f) iz teorema 7.4.2 izvesti slične ocjene za one derivacije koje imaju smisla obzirom na pretpostavljenu glatkoću. Prema prošlom zadatku, ocjene treba tražiti u obliku*

$$\|D^r H(x) - D^r f(x)\|_\infty \leq C_r\, \bar{h}^{e_f - r}\, M_f, \quad r \in \{0,1,2,3\},$$

gdje su C_r konstante ovisne o r, a osnovni eksponenti e_f i "mjere" funkcije M_f ovise samo o klasi funkcije (ne i o r), pa se mogu "pročitati" iz teorema ($r = 0$). Uvjerite se da ocjene imaju smisla samo za $r \leq e_f$, a dokazuju se sličnom tehnikom.

Zadatak 7.4.7 (Superkonvergencija) *Uz pretpostavke dodatne glatkoće funkcije f, u posebno izbaranim točkama može se dobiti i viši red aproksimacije pojedinih derivacija funkcije f.*

(*a*) *U točkama $x_i^* := x_i + h_i/2$ prva derivacija može se aproksimirati s $O(h_i^4)$, a treća s $O(h_i^2)$. Točnije, vrijedi*

$$DH(x^*) = Df(x^*) - \frac{h_i^4}{1920}\, D^4 f(x^*) + O(h_i^5),$$
$$D^3 H(x^*) = D^3 f(x^*) + \frac{h_i^2}{40}\, D^4 f(x^*) + O(h_i^3).$$

(*b*) *U točkama $\bar{x}_i := x_i + (3 \pm \sqrt{3})h_i/6$ druga derivacija može se aproksimirati s $O(h_i^3)$. Točnije, vrijedi*

$$D^2 H(\bar{x}) = D^2 f(\bar{x}) \pm \frac{\sqrt{3}\, h_i^3}{540}\, D^5 f(\bar{x}) + O(h_i^4).$$

Nađite uz koje pretpostavke dodatne glatkoće funkcije f vrijede ove tvrdnje i ocjene, i dokažte ih.

7.4.3. Potpuni kubični splajn

Zahtijevamo li neprekidnost druge derivacije od po dijelovima kubičnih funkcija, dolazimo prirodno na definiciju **potpunog kubičnog splajna**, koji se često još zove i samo **kubični splajn**. Cilj nam je razmotriti algoritme za konstrukciju kubičnih splajnova koji interpoliraju zadane podatke — vrijednosti funkcije, ali ne i njezine derivacije, jer tražimo veću glatkoću. Takav splajn zovemo **kubični interpolacijski splajn**.

Od svih splajn funkcija, kubični interpolacijski splajn je vjerojatno najviše korišten i najbolje izučen u smislu aproksimacije i brojnih primjena, od aproksimacije u raznim normama, do rješavanja rubnih problema za obične diferencijalne jednadžbe. Ime "splajn" (eng. **"spline"**) označava elastičnu letvicu koja se mogla učvrstiti na rebra brodova kako bi se modelirao oblik oplate; točna etimologija riječi pomalo je zaboravljena. U matematičkom smislu pojavljuje se prvi put u radovima Eulera, oko 1700. godine, i slijedi mehaničku definiciju elastičnog štapa.

Središnja linija s takvog štapa (ona koja se ne deformira kod transverzalnog opterećenja) u linearnoj teoriji elastičnosti ima jednadžbu

$$-D^2(EI\, D^2 s(x)) = f(x),$$

gdje je E Youngov modul elastičnosti štapa, a I moment inercije presjeka štapa oko njegove osi. Pretpostavimo li da je štap izrađen od homogenog materijala, i da ne

mijenja poprečni presjek (E i I su konstante), dolazimo na jednadžbu

$$-D^4 s = f,$$

gdje je f vanjska sila po jedinici duljine. U odsustvu vanjske sile ($f = 0$), središnja linija s elastične letvice je dakle kubični polinom.

Ako je letvica učvršćena u osloncima s koordinatama x_i, $i = 0, \ldots, N$, treća derivacija u tim točkama ima diskontinuitet (ova činjenica je posljedica zakona održanja momenta, i trebalo bi ju posebno izvesti). Između oslonaca, na podintervalima $[x_i, x_{i+1}]$, središnja linija je i dalje kubični polinom, ali u točkama x_i imamo prekid treće derivacije. Dakle, s je po dijelovima kubični polinom, a druga derivacija s'' je globalno neprekidna.

Definicija 7.4.2 *Neka su u čvorovima $a = x_0 < x_1 < \cdots < x_N = b$ zadane vrijednosti f_i, za $i = 0, \ldots, N$. Potpuni interpolacijski kubični splajn je funkcija $S_3 \in C^2[a, b]$ koja zadovoljava uvjete*

(1) $S_3(x) = a_{i0} + a_{i1}(x - x_i) + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3$, *za svaki* $x \in [x_i, x_{i+1}]$;

(2) $S_3(x_i) = f_i$, *za* $i = 0, \ldots, N$.

Kako se S_3 na svakom od N podintervala određuje s 4 koeficijenta, ukupno imamo $4N$ koeficijenata koje treba odrediti. Uvjeti glatkoće (funkcija, prva i druga derivacija u unutrašnjim čvorovima) vežu $3(N - 1)$ koeficijenata, a uvjeti interpolacije $N + 1$ koeficijenata. Preostaje dakle odrediti

$$4N - 3(N - 1) - (N + 1) = 2$$

dodatna koeficijenta. Dodatni uvjeti obično se zadaju u rubovima intervala, stoga naziv **rubni uvjeti**. U praksi se najčešće koriste sljedeći rubni uvjeti:

$$\begin{array}{lll}
(R1)\ DS_3(a) = Df(a), & DS_3(b) = Df(b), & \text{(potpuni rubni uvjeti);} \\
(R2)\ D^2S_3(a) = 0, & D^2S_3(b) = 0, & \text{(prirodni rubni uvjeti);} \\
(R3)\ D^2S_3(a) = D^2f(a), & D^2S_3(b) = D^2f(b); & \\
(R4)\ DS_3(a) = DS_3(b), & D^2S_3(a) = D^2S_3(b), & \text{(periodički rubni uvjeti).}
\end{array} \tag{7.4.8}$$

Tradicionalno se naziv **potpuni splajn** koristi za splajn oderđen rubnim uvjetima ($R1$) interpolacije prve derivacije u rubovima. Splajn određen prirodnim rubnim uvjetima ($R2$) zove se **prirodni splajn**. Njega možemo smatrati specijalnim slučajem rubnih uvjeta ($R3$) interpolacije druge derivacije u rubovima, naravno, uz uvjet da sama funkcija zadovoljava prirodne rubne uvjete. Na kraju, splajn određen periodičkim rubnim uvjetima ($R4$) zove se **periodički splajn**, a koristi se za interpolaciju periodičkih funkcija f s periodom $[a, b]$ (tada je $f_0 = f_N$ i f zadovoljava periodičke rubne uvjete).

Algoritam za konstrukciju interpolacijskog kubičnog splajna možemo izvesti na dva načina. U prvom, za nepoznate parametre koje treba odrediti uzimamo vrijednosti **prve** derivacije splajna u čvorovima. Tradicionalna oznaka za te parametre je $m_i := DS_3(x_i)$, za $i = 0, \ldots, N$. U drugom, za nepoznate parametre uzimamo vrijednosti **druge** derivacije splajna u čvorovima, koristeći globalnu neprekidnost D^2S_3, uz tradicionalnu oznaku $M_i := D^2S_3(x_i)$, za $i = 0, \ldots, N$. Napomenimo odmah da se ta dva algoritma dosta ravnopravno koriste u praksi, a za ocjenu greške trebamo i jednog i drugog, pa ćemo napraviti oba izvoda.

Prvi algoritam dobivamo primijenom Hermiteove interpolacije, ali ne zadajemo derivacije, već nepoznate derivacije m_i ostavljamo kao parametre, koje treba odrediti tako da se postigne globalna pripadnost splajna klasi $C^2[a, b]$.

Drugim riječima, tražimo da S_3 zadovoljava uvjete interpolacije $S_3(x_i) = f_i$, $DS_3(x_i) = m_i$, za $i = 0, \ldots, N$, gdje su f_i zadani, a m_i nepoznati. Uz standardne oznake iz prethodnog odjeljka, prema (7.4.3), S_3 možemo na svakom podintervalu napisati u obliku

$$\begin{aligned} S_3(x) = {} & f_i\,(1-t)^2(1+2t) + f_{i+1}\,t^2(3-2t) \\ & + m_i\,h_i\,t(1-t)^2 - m_{i+1}\,h_i\,t^2(1-t), \end{aligned} \tag{7.4.9}$$

gdje je $t = (x - x_i)/h_i$, za $x \in [x_i, x_{i+1}]$. Parametre m_i, m_{i+1} moramo odrediti tako da je druga derivacija D^2S_3 neprekidna u unutrašnjim čvorovima. Budući da je

$$D^2S_3(x) = \frac{f_{i+1} - f_i}{h_i^2}\,(6 - 12t) + \frac{m_i}{h_i}\,(-4 + 6t) + \frac{m_{i+1}}{h_i}\,(-2 + 6t),$$

slijedi

$$\begin{aligned} D^2S_3(x_i + 0) &= 6\,\frac{f_{i+1} - f_i}{h_i^2} - \frac{4m_i}{h_i} - \frac{2m_{i+1}}{h_i}, \\ D^2S_3(x_i - 0) &= -6\,\frac{f_i - f_{i-1}}{h_{i-1}^2} + \frac{2m_{i-1}}{h_{i-1}} + \frac{4m_i}{h_{i-1}}. \end{aligned}$$

Uz oznake

$$\mu_i = \frac{h_{i-1}}{h_{i-1} + h_i}, \qquad \lambda_i = 1 - \mu_i, \qquad c_i = 3\Big(\mu_i\,\frac{f_{i+1} - f_i}{h_i} + \lambda_i\,\frac{f_i - f_{i-1}}{h_{i-1}}\Big),$$

uvjete neprekidnosti D^2S_3 u x_i, za $i = 1, \ldots, N-1$, možemo napisati u obliku

$$\lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = c_i, \quad i = 1, \ldots, N-1. \tag{7.4.10}$$

Dobili smo $N - 1$ jednadžbi za $N + 1$ nepoznanica m_i, pa nam fale još dvije jednadžbe. Naravno, uvjetima (7.4.10) treba dodati još neke rubne uvjete.

Za rubne uvjete $(R1)$, $(R2)$ i $(R3)$ dobivamo linearni sustav oblika

$$\begin{aligned} 2m_0 + \mu_0^*\,m_1 &= c_0^*, \\ \lambda_i\,m_{i-1} + 2m_i + \mu_i\,m_{i+1} &= c_i, \qquad i = 1, \ldots, N-1, \\ \lambda_N^*\,m_{N-1} + 2m_N &= c_N^*. \end{aligned} \tag{7.4.11}$$

Koeficijenti μ_0^*, c_0^*, λ_N^* i c_N^* određuju se ovisno o rubnim uvjetima. Za rubne uvjete $(R1)$ imamo

$$\mu_0^* = \lambda_N^* = 0, \quad c_0^* = 2Df(a), \quad c_N^* = 2Df(b),$$

a za rubne uvjete $(R3)$

$$\mu_0^* = \lambda_N^* = 1, \quad c_0^* = 3\,\frac{f_1 - f_0}{h_0} - \frac{h_0}{2}\,D^2 f(a), \quad c_N^* = 3\,\frac{f_N - f_{N-1}}{h_{N-1}} + \frac{h_{N-1}}{2}\,D^2 f(b).$$

Prirodni rubni uvjeti $(R2)$ su specijalni slučaj $(R3)$, uz $D^2 f(a) = D^2 f(b) = 0$.

Ako je f periodička funkcija, onda je $f_0 = f_N$ i $m_0 = m_N$ (periodički rubni uvjet na prvu derivaciju). Da bismo zapisali uvjet periodičnosti druge derivacije, možemo na periodički način produljiti mrežu, tako da dodamo još jedan čvor x_{N+1}, ali tako da je $x_{N+1} - x_N = x_1 = x_0$, tj. $h_N = h_0$. Zbog pretpostavke periodičnosti, moramo staviti $f_{N+1} = f_1$ i $m_{N+1} = m_1$. Na taj način, uvjet periodičnosti druge derivacije postaje ekvivalentan uvjetu neprekidnosti druge derivacije u točki x_N, tj. jednadžbi oblika (7.4.10) za $i = N$. Kad iskoristimo sve pretpostavke

$$f_0 = f_N, \quad f_{N+1} = f_1, \quad m_0 = m_N, \quad m_{N+1} = m_1, \quad h_N = h_0,$$

dobivamo sustav od samo N jednadžbi

$$\begin{aligned} 2m_1 + \mu_1\, m_2 + \lambda_1\, m_N &= c_1, \\ \lambda_i\, m_{i-1} + 2m_i + \mu_i\, m_{i+1} &= c_i, \qquad i = 2, \ldots, N-1, \\ \mu_N\, m_1 + \lambda_N\, m_{N-1} + 2m_N &= c_N. \end{aligned} \tag{7.4.12}$$

Uočite da smo jednadžbu $m_0 = m_N$ već iskoristili za eliminaciju nepoznanice m_0.

Ostaje odgovoriti na očito pitanje: da li dobiveni linearni sustavi imaju jedinstveno rješenje.

Teorem 7.4.3 *Postoji jedinstveni interpolacijski kubični splajn koji zadovoljava jedan od rubnih uvjeta* $(R1)$*–*$(R4)$*.*

Dokaz. U svim navedenim slučajevima lako se vidi da je matrica linearnog sustava strogo dijagonalno dominantna, što povlači regularnost. Naime, svi dijagonalni elementi su jednaki 2, a zbroj izvandijagonalnih elemenata je najviše $\lambda_i + \mu_i = 1$, (uz dogovor $\lambda_N^* = \lambda_N$ i $\mu_0^* = \mu_0$). ■

Algoritam 7.4.1 (Interpolacijski kubični splajn)

(1) Riješi linearni sustav (7.4.11) ili (7.4.12);
(2) Binarnim pretraživanjem nađi indeks i tako da vijedi $x \in [x_i, x_{i+1})$;
(3) Hornerovom shemom (7.4.5) izračunaj $S_3(x)$.

Primijetimo da je za rješavanje sustava potrebno samo $O(N)$ operacija, obzirom na specijalnu vrpčastu strukturu matrice. Također, matrica ne ovisi o vrijednostima funkcije koja se interpolira, pa se korak (1) u Algoritmu 7.4.1 sastoji od LR faktorizacije matrice, koju treba izračunati samo jednom.

Za računanje vrijednosti $S_3(x)$ obično se koriste formule (7.4.4)–(7.4.5). Ako je potrebno računati splajn u mnogo točaka (recimo, u svrhu brze reprodukcije grafa splajna), možemo napisati **algoritam konverzije**, tj. naći vezu između definicionog oblika splajna (v. definiciju 7.4.2) i oblika danog formulama (7.4.4)–(7.4.5). Definiciona reprezentacija splajna kao kubične funkcije na svakom podintervalu subdivizije zove se ponekad i **po dijelovima polinomna** reprezentacija, ili skraćeno PP–reprezentacija.

Zadatak 7.4.8 *Kolika je točno ušteda u broju aritmetičkih operacija potrebnih za računanje $S_3(x)$ pri prijelazu na PP–reprezentaciju? Još "brži" oblik reprezentacije je standardni kubni polinom $S_3(x) = b_{i0} + b_{i1}x + b_{i2}x^2 + b_{i3}x^3$, za svaki $x \in [x_i, x_{i+1}]$. Njega* **ne treba** *koristiti. Zašto?*

Kao što smo već rekli, u nekim slučajevima ugodnija je druga reprezentacija interpolacijskog kubičnog splajna, u kojoj se, umjesto m_i, kao nepoznanice javljaju $M_i := D^2S(x_i)$, za $i = 0, \ldots, N$. Zbog popularnosti i česte implementacije izvedimo ukratko i ovu reprezentaciju.

Na svakom podintervalu $[x_i, x_{i+1}]$ kubični splajn S_3 je kubični polinom kojeg određujemo iz uvjeta interpolacije funkcije i **druge** derivacije u rubovima

$$S_3(x_i) = f_i, \quad S_3(x_{i+1}) = f_{i+1}, \quad D^2S_3(x_i) = M_i, \quad D^2S_3(x_{i+1}) = M_{i+1}.$$

Ovaj sustav jednadžbi ima jedinstveno rješenje (dokažite to), odakle onda možemo izračunati koeficijente kubnog polinoma. Međutim, traženu reprezentaciju možemo jednostavno i "pogoditi", ako $S_3(x)$ na $[x_i, x_{i+1}]$ napišemo kao linearnu interpolaciju funkcijskih vrijednosti plus neka korekcija. Odmah se vidi da tražena korekcija ima oblik linearne interpolacije druge derivacije puta neki kvadratni faktor koji se poništava u rubovima. Dobivamo oblik

$$S_3(x) = f_i\,(1-t) + f_{i+1}\,t - \frac{h_i^2}{6}\,t(1-t)\,[M_i\,(2-t) + M_{i+1}\,(1+t)],$$

gdje je opet $t = (x - x_i)/h_i$, za $x \in [x_i, x_{i+1}]$ i $i = 0, \ldots, N-1$. Odavde lako izlazi

$$\begin{aligned} DS_3(x) &= \frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{6}\,[M_i\,(2 - 6t + 3t^2) + M_{i+1}\,(1 - 3t^2)], \\ D^2S_3(x) &= M_i\,(1-t) + M_{i+1}\,t, \\ D^3S_3(x) &= \frac{M_{i+1} - M_i}{h_i}. \end{aligned}$$

Interpolacija druge derivacije u čvorovima ne garantira da je i prva derivacija neprekidna. To treba dodatno zahtijevati. Kako je

$$DS_3(x_i+0) = \frac{f_{i+1}-f_i}{h_i} - \frac{h_i}{6}\,(2M_i + M_{i+1}),$$
$$DS_3(x_i-0) = \frac{f_i-f_{i-1}}{h_{i-1}} + \frac{h_{i-1}}{6}\,(M_{i-1} + 2M_i),$$

iz uvjeta neprekidnosti prve derivacije u unutrašnjim čvorovima dobivamo $N-1$ jednadžbi traženog linearnog sustava

$$\mu_i\, M_{i-1} + 2M_i + \lambda_i\, M_i = \frac{6}{h_{i-1}+h_i}\left(\frac{f_{i+1}-f_i}{h_i} - \frac{f_i-f_{i-1}}{h_{i-1}}\right), \tag{7.4.13}$$

za $i = 1, \ldots, N-1$, gdje je, kao i prije, $\mu_i = h_{i-1}/(h_{i-1}+h_i)$ i $\lambda_i = 1-\mu_i$.

Zadatak 7.4.9 *Napišite nedostajuće jednadžbe za rubne uvjete, i pokažite da je matrica sustava strogo dijagonalno dominantna.*

Na kraju, primijetimo da je algoritam za računanje vrijednosti $S_3(x)$ vrlo sličan ranijem, s tim što treba primijeniti malo drugačiju Hornerovu shemu (ekvivalent formula (7.4.4)–(7.4.5) za algoritam 7.4.1):

$$S_3(x) = f_i + t\,\{(f_{i+1}-f_i) - (x_{i+1}-x)\,[(x_{i+1}-x+h_i)\widetilde{M}_i + (h_i+x-x_i)\widetilde{M}_{i+1}]\},$$

gdje je $\widetilde{M}_i := M_i/6$.

Ocjena greške za potpuni kubični splajn je teži problem nego za Hermiteov kubični splajn, budući da su koeficijenti zadani implicitno kao rješenje jednog linearnog sustava.

Teorem 7.4.4 *Neka je S_3 interpolacijski kubični splajn za funkciju f koji zadovoljava jedan od rubnih uvjeta $(R1)$–$(R4)$ u (7.4.8). Tada vrijedi*

$$\|D^r S_3(x) - D^r f(x)\|_\infty \le C_r\, \bar{h}^{e_f - r}\, M_f, \quad r = 0,1,2,3,$$

gdje su C_r konstante (ovisne o r), e_f osnovni eksponenti i M_f "mjere" funkcije (e_f

i M_f ovise samo o klasi funkcije, ne i o r), dani sljedećom tablicom:

Klasa funkcije	M_f	e_f	C_0	C_1	C_2	C_3
$C^1[a,b]$	$\omega(Df)$	1	$\frac{9}{8}$	4		
$L^2_\infty[a,b]$	$\|D^2f\|_\infty$	2	$\frac{13}{48}$	0.86229		
$C^2[a,b]$	$\omega(D^2f)$	2	$\frac{19}{96}$	$\frac{2}{3}$	4	
$L^3_\infty[a,b]$	$\|D^3f\|_\infty$	3	$\frac{41}{864}$	$\frac{4}{27}$	$\frac{1}{2}+\frac{4\sqrt{3}}{9}$	
$C^2[a,b]\bigcap_i C^3[x_i,x_{i+1}]$	$\omega(D^3f)$	3	$\frac{41}{1728}$	$\frac{2}{27}$	$\frac{1}{2}+\frac{2\sqrt{3}}{9}$	$1+\frac{4\sqrt{3}}{9}\,\beta$
$C^2[a,b]\bigcap_i L^4_\infty[x_i,x_{i+1}]$	$\|D^4f\|_\infty$	4	$\frac{5}{384}$	$\frac{1}{24}$	$\frac{3}{8}$	$\frac{1}{2}\left(\frac{1}{\beta}+\beta\right)$

s tim da je

$$\beta := \frac{\max_i h_i}{\min_i h_i}$$

mjera "neuniformnosti" mreže (u zadnjem stupcu tablice).

Mjesta u tablici koja nisu popunjena znače da **ne postoje** odgovarajuće ocjene. Napomenimo, također, da rijetko korištene ocjene koje odgovaraju još nižoj glatkoći funkcije f, na primjer, $f \in C[a,b]$ ili $f \in L^1_\infty[a,b]$ nisu navedene, iako se mogu izvesti (dokaz nije trivijalan). Osim toga, nije poznato da li su sve konstante optimalne, iako se to može pokazati u nekim važnim slučajevima (na primjer, u zadnjem redu tablice, koji podrazumijeva najveću glatkoću, sve su konstante najbolje moguće).

Dokaz. Dokažimo neke od ocjena u teoremu 7.4.4 (preostale pokašajte dokazati sami).

Neka je H Hermitski interpolacijski kubični splajn i $S := S_3$ interpolacijski kubični splajn. Tada grešku možemo napisati kao

$$E(x) := S(x) - f(x) = [H(x) - f(x)] + [S(x) - H(x)].$$

Oba interpolacijska splajna $S(x)$ i $H(x)$ možemo reprezentirati preko Hermiteove baze na svakom intervalu $[x_i, x_{i+1}]$ (v. (7.4.9), (7.4.3)), pa oduzimanjem tih reprezentacija slijedi

$$S(x) - f(x) = [H(x) - f(x)] + h_i\,[t(1-t)^2\,(m_i - Df(x_i)) \\ - (1-t)t^2\,(m_{i+1} - Df(x_{i+1}))].$$

Odavde je

$$|S(x) - f(x)| \leq |H(x) - f(x)| + h_i\, t(1-t)\, \max_i |m_i - Df(x_i)|. \tag{7.4.14}$$

Za derivaciju imamo

$$\begin{aligned} DS(x) - Df(x) = [DH(x) - Df(x)] + [(1-t)(1-3t)\,(m_i - Df(x_i)) \\ - t(2-3t)\,(m_{i+1} - Df(x_{i+1}))], \end{aligned}$$

pa je stoga

$$\begin{aligned} |DS(x) - Df(x)| \leq |DH(x) - Df(x)| \\ + [(1-t)|1-3t| + t|2-3t|]\, \max_i |m_i - Df(x_i)|. \end{aligned} \tag{7.4.15}$$

Ocjene za $|H(x) - f(x)|$ izveli smo u teoremu 7.4.4, a ocjene za $|DH(x) - Df(x)|$ mogu se izvesti na sličan način (v. zadatke 7.4.5 i 7.4.6). Ostaje dakle ocijeniti drugi član na desnoj strani u (7.4.14) i (7.4.15).

Za drugu derivaciju znamo da je

$$D^2S(x) = M_i\,(1-t) + M_{i+1}\,t,$$

pa zaključujemo da je

$$\begin{aligned} D^2S(x) - D^2f(x) = (1-t)\,(M_i - D^2f(x_i)) + t\,(M_{i+1} - D^2f(x_{i+1}) \\ + (1-t)\,D^2f(x_i) + t\,D^2f(x_{i+1}) - D^2f(x). \end{aligned}$$

Ali, kako je $(1-t)D^2f(x_i) + tD^2f(x_{i+1}) - D^2f(x)$ pogreška kod interpolacije funkcije D^2f linearnim splajnom S_1 (teorem 7.4.1), možemo ju i ovako ocijeniti

$$|D^2S(x) - D^2f(x)| \leq |S_1(x) - D^2f(x)| + \max_i |M_i - D^2f(x_i)|. \tag{7.4.16}$$

Slično je i za treću derivaciju

$$|D^3S(x) - D^3f(x)| \leq |DS_1(x) - D^3f(x)| + \frac{2}{\min_i h_i}\, \max_i |M_i - D^2f(x_i)|. \tag{7.4.17}$$

Ako pogledamo nejednakosti (7.4.14), (7.4.15), (7.4.16) i (7.4.17), vidimo da preostaje ocijeniti $\max_i |m_i - Df(x_i)|$ i $\max_i |M_i - D^2f(x_i)|$. Ove ocjene, kao i sve druge, ovise o klasi funkcija.

Tvrdimo da vrijedi

$$\max_i |m_i - Df(x_i)| \leq q_f,$$

gdje je q_f dan sljedećom tablicom za 6 karakterističnih klasa funkcija:

Klasa funkcije	q_f
$C^1[a,b]$	$3\,\omega(Df)$
$L^2_\infty[a,b]$	$\frac{5}{6}\,\bar{h}\,\|D^2 f\|_\infty$
$C^2[a,b]$	$\frac{2}{3}\,\bar{h}\,\omega(D^2 f)$
$L^3_\infty[a,b]$	$\frac{4}{27}\,\bar{h}^2\,\|D^3 f\|_\infty$
$C^2[a,b]\bigcap_i C^3[x_i,x_{i+1}]$	$\frac{2}{27}\,\bar{h}^2\,\omega(D^3 f)$
$C^2[a,b]\bigcap_i L^4_\infty[x_i,x_{i+1}]$	$\frac{1}{24}\,\bar{h}^3\,\|D^4 f\|_\infty$

Da dokažemo ovu tablicu, pretpostavimo rubne uvjete $(R1)$ na derivaciju. Uvedemo li u linearnom sustavu (7.4.11) nove nepoznanice $q_i := m_i - Df(x_i)$, dobijemo sustav

$$\begin{aligned} q_0 &= 0, \\ \lambda_i\, q_{i-1} + 2q_i + \mu_i\, q_{i+1} &= \tilde{c}_i, \qquad i = 1,\dots,N-1, \\ q_N &= 0, \end{aligned}$$

gdje su desne strane

$$\begin{aligned} \tilde{c}_i = 3\mu_i\,\frac{f_{i+1}-f_i}{h_i} + 3\lambda_i\,\frac{f_i - f_{i-1}}{h_{i-1}} \\ - \lambda_i\, Df(x_{i-1}) - 2Df(x_i) - \mu_i\, Df(x_{i+1}). \end{aligned} \tag{7.4.18}$$

Da bismo ocijenili $|q_i|$, zapišimo ovaj sustav u matričnom obliku $Aq = \tilde{c}$, ili $q = A^{-1}\tilde{c}$. Vidimo odmah da je $A = 2I + B$, gdje je B matrica koja sadrži samo izvandijagonalne elemente λ_i i μ_i. Zbog $\lambda_i + \mu_i \le 1$ (jednakost vrijedi u svim jednadžbama, osim prve i zadnje), slijedi $\|B\|_\infty \le 1$. Sada nije teško ocijeniti $\|A^{-1}\|_\infty$

$$A = 2\Big(I + \frac{1}{2}\,B\Big) \implies \|A^{-1}\|_\infty \le \frac{1}{2}\Big(1 - \frac{1}{2}\,\|B\|_\infty\Big)^{-1} \le 1.$$

Na kraju, iz $q = A^{-1}\tilde{c}$ slijedi

$$|q_i| \le \|q\|_\infty \le \|A^{-1}\|_\infty \|\tilde{c}\|_\infty = \max_i |\tilde{c}_i|.$$

Drugim riječima, da bismo dokazali ocjene iz tablice za q_f, dovoljno je ocijeniti $|\tilde{c}_i|$.

Pretpostavimo da je $f \in C^1[a,b]$ i iskoristimo Lagrangeov teorem o srednjoj vrijednosti za prva dva člana u izrazu (7.4.18) za $\tilde{c}_i$. Tada je $\lambda_i + \mu_i = 1$, pa je

$$\begin{aligned}\tilde{c}_i &= 3\mu_i\, Df(\xi_{i,i+1}) + 3\lambda_i\, Df(\xi_{i-1,i}) - \lambda_i\, Df(x_{i-1}) - 2Df(x_i) - \mu_i\, Df(x_{i+1})\\ &= \lambda_i\,[Df(\xi_{i-1,i}) - Df(x_{i-1})] + 2\lambda_i\,[Df(\xi_{i-1,i}) - Df(x_i)]\\ &\quad + \mu_i\,[Df(\xi_{i,i+1}) - Df(x_{i+1})] + 2\mu_i\,[Df(\xi_{i,i+1}) - Df(x_i)],\end{aligned}$$

odakle slijedi

$$|\tilde{c}_i| \le 3(\lambda_i + \mu_i)\,\omega(Df) = 3\,\omega(Df),$$

čime smo dokazali prvu ocjenu u tablici za q_f.

Ako je $f \in C^2[a,b] \bigcap_i L^4_\infty[x_i, x_{i+1}]$, u izrazu (7.4.18) za $\tilde{c}_i$ možemo razviti f_{i-1}, $Df(x_{i-1})$, f_{i+1}, $Df(x_{i+1})$ u Taylorov red oko x_i, koristeći integralni oblik ostatka. Napomenimo da nam nije potrebna neprekidnost treće derivacije. U tom slučaju imamo dakle

$$\begin{aligned}\tilde{c}_i = 3\mu_i &\left\{ Df(x_i) + \frac{h_i}{2}\, D^2 f(x_i) + \frac{h_i^2}{6}\, D^3 f(x_i + 0) \right.\\ &\qquad \left. + \frac{1}{6h_i} \int_{x_i}^{x_{i+1}} (x_{i+1} - v)^3\, D^4 f(v)\, dv \right\}\\ + 3\lambda_i &\left\{ Df(x_i) - \frac{h_{i-1}}{2}\, D^2 f(x_i) + \frac{h_{i-1}^2}{6}\, D^3 f(x_i - 0) \right.\\ &\qquad \left. - \frac{1}{6h_{i-1}} \int_{x_i}^{x_{i-1}} (x_{i-1} - v)^3\, D^4 f(v)\, dv \right\}\\ - \mu_i &\left\{ Df(x_i) + h_i\, D^2 f(x_i) + \frac{h_i^2}{2}\, D^3 f(x_i + 0) \right.\\ &\qquad \left. + \frac{1}{2} \int_{x_i}^{x_{i+1}} (x_{i+1} - v)^2\, D^4 f(v)\, dv \right\}\\ - 2Df(x_i)&\\ - \lambda_i &\left\{ Df(x_i) - h_{i-1}\, D^2 f(x_i) + \frac{h_{i-1}^2}{2}\, D^3 f(x_i - 0) \right.\\ &\qquad \left. + \frac{1}{2} \int_{x_i}^{x_{i-1}} (x_{i-1} - v)^2\, D^4 f(v)\, dv \right\}.\end{aligned}$$

Članovi s $Df(x_i)$, $D^2f(x_i)$, $D^3f(x_i+0)$ i $D^3f(x_i-0)$ se skrate, pa ostaje samo

$$\tilde{c}_i = \frac{\mu_i}{2}\int_{x_i}^{x_{i+1}}\Big[\frac{(x_{i+1}-v)^3}{h_i} - (x_{i+1}-v)^2\Big]D^4f(v)\,dv + \frac{\lambda_i}{2}\int_{x_i}^{x_{i-1}}\Big[-\frac{(x_{i-1}-v)^3}{h_{i-1}} - (x_{i-1}-v)^2\Big]D^4f(v)\,dv.$$

Zamijenimo li varijable supstitucijom $\tau h_i := v - x_i$ u prvom integralu, odnosno, $\tau h_{i-1} := v - x_{i-1}$ u drugom integralu, dobivamo

$$\tilde{c}_i = -\frac{\mu_i h_i^3}{2}\int_0^1 \tau(1-\tau)^2\,D^4f(x_i+\tau h_i)\,d\tau + \frac{\lambda_i h_{i-1}^3}{2}\int_0^1 \tau^2(1-\tau)\,D^4f(x_{i-1}+\tau h_{i-1})\,d\tau.$$

Odavde lagano ocijenimo

$$|\tilde{c}_i| \le \frac{1}{2}\,\|D^4f\|_\infty\left\{\mu_i h_i^3\int_0^1\tau(1-\tau)^2\,d\tau + \lambda_i h_{i-1}^3\int_0^1\tau^2(1-\tau)\,d\tau\right\} = \frac{1}{24}\,\|D^4f\|_\infty\,(\mu_i h_i^3 + \lambda_i h_{i-1}^3).$$

Uvrštavanjem μ_i, λ_i (v. 7.4.10) dobivamo

$$|\tilde{c}_i| \le \frac{h_i h_{i-1}}{24}\,\frac{h_i^2+h_{i-1}^2}{h_i+h_{i-1}}\,\|D^4f\|_\infty.$$

Na kraju, kako je

$$\frac{h_i^2+h_{i-1}^2}{h_i+h_{i-1}} \le \max\{h_i, h_{i-1}\},$$

dolazimo do zadnje ocjene u tablici za q_f

$$|\tilde{c}_i| \le \frac{1}{24}\,\bar{h}^3\,\|D^4f\|_\infty.$$

Upotrebom Taylorove formule, teorema o srednjoj vrijednosti i leme 7.4.1, na već poznati način, dokazuju se i ostale ocjene u tablici. Napomenimo još, da je sličnu analizu potrebno napraviti i za druge tipove rubnih uvjeta. Pokazuje se da rezultati i tehnika dokaza ne ovise mnogo o tipu rubnih uvjeta. To, naravno, vrijedi samo uz pretpostavku da funkcija f zadovoljava iste rubne uvjete kao i splajn, ako rubni uvjet ne ovisi o funkciji (na primjer, $(R2)$ ili $(R4)$). U protivnom, dobivamo slabije ocjene.

Nadalje, za ocjenu druge i treće derivacije, moramo naći ocjene oblika

$$\max_i |M_i - D^2 f(x_i)| \leq Q_f.$$

I u ovom slučaju imamo tablicu s 4 ocjene, u ovisnosti o klasi funkcije:

Klasa funkcije	Q_f
$C^2[a,b]$	$3\,\omega(D^2 f)$
$L^3_\infty[a,b]$	$\frac{4\sqrt{3}}{9}\,\bar{h}\,\|D^3 f\|_\infty$
$C^2[a,b] \bigcap_i C^3[x_i, x_{i+1}]$	$\frac{2\sqrt{3}}{9}\,\bar{h}\,\omega(D^3 f)$
$C^2[a,b] \bigcap_i L^4_\infty[x_i, x_{i+1}]$	$\frac{1}{4}\,\bar{h}^2\,\|D^4 f\|_\infty$

Tehnika dokaza ove tablice je dosta slična onoj za prethodnu tablicu, s time da se oslanja na linearni sustav (7.4.13), pa ocjene ostavljamo kao zadatak.

Da bismo na kraju dokazali ovaj teorem, ograničimo se na "najglatkiju" klasu funkcija $C^2[a,b] \bigcap_i L^4_\infty[x_i, x_{i+1}]$; tehnika dokaza potpuno je ista i za sve druge klase. Ključna je ocjena (7.4.14):

$$|S(x) - f(x)| \leq |H(x) - f(x)| + h_i\, t(1-t)\, \max_i |m_i - Df(x_i)|.$$

Prvi dio čini greška kod interpolacije Hermiteovim splajnom, za koju, prema (7.4.7), znamo da vrijedi

$$|H(x) - f(x)| \leq \frac{t^2(1-t)^2}{4!}\, h_i^4\, \|D^4 f\|_\infty, \quad t = \frac{x - x_i}{h_i} \in [0,1],$$

a drugi dio pročitamo u tablici za $\max_i |m_i - Df(x_i)|$. Ukupno je dakle

$$|S(x) - f(x)| \leq \frac{1}{24}\, t(1-t)\,[1 + t(1-t)]\, \max_i h_i^4\, \|D^4 f\|_\infty \leq \frac{5}{384}\, \bar{h}^4\, \|D^4 f\|_\infty.$$

Zanimljivo je da ova ocjena samo 5 puta veća od ocjene za Hermiteov interpolacijski splajn, koji zahtijeva poznate derivacije funkcije f u **svim** čvorovima interpolacije, a ovdje ih koristimo samo na rubu (uz rubne uvjete $(R1)$). ■

7.5. Diskretna metoda najmanjih kvadrata

Neka je funkcija f zadana na diskretnom skupu točaka $x_0, \ldots, x_n$ kojih je mnogo više nego nepoznatih parametara aproksimacijske funkcije

$$\varphi(x, a_0, \ldots, a_m).$$

Funkcija φ određuje se iz uvjeta da euklidska norma (norma 2) vektora pogrešaka u čvorovima aproksimacije bude najmanja moguća, tj. tako da minimiziramo S,

$$S = \sum_{k=0}^{n} (f(x_k) - \varphi(x_k))^2 \to \min .$$

Ovu funkciju S (kvadrat euklidske norme vektora greške) interpretiramo kao funkciju nepoznatih parametara

$$S = S(a_0, \ldots, a_m).$$

Očito je uvijek $S \geq 0$, bez obzira kakvi su parametri. Dakle, zadatak je minimizirati funkciju S kao funkciju više varijabli $a_0, \ldots, a_m$. Ako je S dovoljno glatka funkcija, a naša je (jer je funkcija u parametrima a_k), nužni uvjet ekstrema je

$$\frac{\partial S}{\partial a_k} = 0, \quad k = 0, \ldots, m.$$

Takav pristup vodi na tzv. **sustav normalnih jednadžbi**.

7.5.1. Linearni problemi i linearizacija

Ilustrirajmo to na najjednostavnijem primjeru, kad je aproksimacijska funkcija pravac.

Primjer 7.5.1 *Zadane su točke $(x_0, f_0), \ldots, (x_n, f_n)$, koje po diskretnoj metodi najmanjih kvadrata aproksimiramo pravcem*

$$\varphi(x) = a_0 + a_1 x.$$

Greška aproksimacije u čvorovima koju minimiziramo je

$$S = S(a_0, a_1) = \sum_{k=0}^{n} (f_k - \varphi(x_k))^2 = \sum_{k=0}^{n} (f_k - a_0 - a_1 x_k)^2 \to \min .$$

Nađimo parcijalne derivacije po parametrima a_0 i a_1:

$$0 = \frac{\partial S}{\partial a_0} = -2 \sum_{k=0}^{n} (f_k - a_0 - a_1 x_k),$$
$$0 = \frac{\partial S}{\partial a_1} = -2 \sum_{k=0}^{n} (f_k - a_0 - a_1 x_k) x_k.$$

Dijeljenjem s -2 i sređivanjem po nepoznanicama a_0, a_1, dobivamo linearni sustav

$$a_0 (n+1) + a_1 \sum_{k=0}^{n} x_k = \sum_{k=0}^{n} f_k$$
$$a_0 \sum_{k=0}^{n} x_k + a_1 \sum_{k=0}^{n} x_k^2 = \sum_{k=0}^{n} f_k x_k.$$

Uvedemo li standardne skraćene oznake

$$s_\ell = \sum_{k=0}^{n} x_k^\ell, \quad t_\ell = \sum_{k=0}^{n} f_k x_k^\ell, \quad \ell \geq 0,$$

linearni sustav možemo pisati kao

$$\begin{aligned} s_0 a_0 + s_1 a_1 &= t_0 \\ s_1 a_0 + s_2 a_1 &= t_1. \end{aligned} \tag{7.5.1}$$

Nije teško pokazati da je matrica sustava regularna, jer je njena determinanta različita od nule. Determinanta matrice sustava jednaka je $s_0 s_2 - s_1^2$. *Cauchy–Schwarzova nejednakost primijenjena na vektore*

$$e = [1, 1, \ldots, 1]^T \quad i \quad x = [x_0, x_1, \ldots, x_n]^T,$$

daje $s_0 s_2 \geq s_1^2$, *pri čemu jednakost vrijedi samo ako su vektori* e *i* x *linearno zavisni. Kako pretpostavljamo da imamo barem dvije različite apscise polaznih točaka* x_k *(prirodan uvjet za pravac koji nije paralelan s ordinatom), nejednakost mora biti stroga, tj. vrijedi* $s_0 s_2 > s_1^2$. *Dakle, postoji jedinstveno rješenje sustava. Samo rješenje dobiva se rješavanjem linearnog sustava (7.5.1).*

Ostaje još pitanje da li smo dobili minimum, ali to nije teško pokazati, korištenjem drugih parcijalnih derivacija (dovoljan uvjet minimuma je pozitivna definitnost Hesseove matrice). Ipak, provjera da se radi o minimumu, može i puno lakše. Budući da se radi o zbroju kvadrata, S *predstavlja paraboloid s otvorom prema gore u varijablama* a_0, a_1, *pa je jasno da takvi paraboloidi imaju minimum. Zbog toga se nikad ni ne provjerava da li je dobiveno rješenje minimum za* S.

Za funkciju φ mogli bismo uzeti i polinom višeg stupnja,

$$\varphi(x) = a_0 + a_1 x + \cdots + a_m x^m,$$

ali postoji opasnost da je za malo veće m ($m \approx 10$) dobiveni sustav vrlo loše uvjetovan (matrica sustava vrlo blizu singularne matrice), pa dobiveni rezultati mogu biti jako pogrešni. Zbog toga se za nikad ne koristi prikaz polinoma u bazi potencija. Ako se uopće koriste aproksimacije polinomima viših stupnjeva, onda se to radi korištenjem ortogonalnih polinoma.

Linearni model diskretnih najmanjih kvadrata je potpuno primjenjiv na opću linearnu funkciju

$$\varphi(x) = a_0 \varphi_0(x) + \cdots + a_m \varphi_m(x),$$

gdje su $\varphi_0, \ldots, \varphi_m$ poznate (zadane) funkcije. Ilustrirajmo to ponovno na općoj linearnoj funkciji s 2 parametra.

Primjer 7.5.2 *Zadane su točke* $(x_0, f_0), \ldots, (x_n, f_n)$*, koje po diskretnoj metodi najmanjih kvadrata aproksimiramo funkcijom oblika*

$$\varphi(x) = a_0\varphi_0(x) + a_1\varphi_1(x).$$

Postupak je potpuno isti kao u prošlom primjeru. Opet minimiziramo kvadrat euklidske norme vektora pogrešaka aproksimacije u čvorovima

$$S = S(a_0, a_1) = \sum_{k=0}^{n}(f_k - \varphi(x_k))^2 = \sum_{k=0}^{n}(f_k - a_0\varphi_0(x_k) - a_1\varphi_0(x_k))^2 \to \min.$$

Sređivanjem parcijalnih derivacija

$$0 = \frac{\partial S}{\partial a_0} = -2\sum_{k=0}^{n}(f_k - a_0\varphi_0(x_k) - a_1\varphi_1(x_k))\,\varphi_0(x_k),$$
$$0 = \frac{\partial S}{\partial a_1} = -2\sum_{k=0}^{n}(f_k - a_0\varphi_0(x_k) - a_1\varphi_1(x_k))\,\varphi_1(x_k),$$

po varijablama a_0*,* a_1*, uz dogovor da je*

$$s_0 = \sum_{k=0}^{n}\varphi_0^2(x_k), \qquad s_1 = \sum_{k=0}^{n}\varphi_0(x_k)\varphi_1(x_k), \qquad s_2 = \sum_{k=0}^{n}\varphi_1^2(x_k),$$
$$t_0 = \sum_{k=0}^{n}f_k\varphi_0(x_k), \qquad t_1 = \sum_{k=0}^{n}f_k\varphi_1(x_k),$$

dobivamo potpuno isti oblik linearnog sustava

$$s_0a_0 + s_1a_1 = t_0$$
$$s_1a_0 + s_2a_1 = t_1.$$

Ovaj sustav ima ista svojstva kao i u prethodnom primjeru. Pokažite to!

Što ako φ nelinearno ovisi o parametrima? Dobili bismo nelinearni sustav jednadžbi, koji se relativno teško rješava. Uglavnom, problem postaje ozbiljan optimizacijski problem, koji se, recimo, može rješavati metodama pretraživanja ili nekim drugim optimizacijskim metodama, posebno prilagođenim upravo za rješavanje nelinearnog problema najmanjih kvadrata (na primjer, Levenberg–Marquardt metoda).

Postoji i drugi pristup. Katkad se jednostavnim transformacijama problem može transformirati u linearni problem najmanjih kvadrata. Nažalost, rješenja linearizirano problema najmanjih kvadrata i rješenja originalnog nelinearnog problema, u principu, **nisu** jednaka. Problem je u različitim mjerama za udaljenost točaka, odnosno mjerama za grešku.

Ilustrirajmo, ponovno, nelinearni problem najmanjih kvadrata na jednom jednostavnom primjeru.

Primjer 7.5.3 *Zadane su točke* $(x_0, f_0), \ldots, (x_n, f_n)$*, koje po diskretnoj metodi najmanjih kvadrata aproksimiramo funkcijom oblika*

$$\varphi(x) = a_0 e^{a_1 x}.$$

Greška aproksimacije u čvorovima (koju minimiziramo) je

$$S = S(a_0, a_1) = \sum_{k=0}^{n} (f_k - \varphi(x_k))^2 = \sum_{k=0}^{n} (f_k - a_0 e^{a_1 x_k})^2 \to \min.$$

Parcijalnim deriviranjem po varijablama a_0 *i* a_1 *dobivamo*

$$0 = \frac{\partial S}{\partial a_0} = -2 \sum_{k=0}^{n} (f_k - a_0 e^{a_1 x_k})\, e^{a_1 x_k},$$
$$0 = \frac{\partial S}{\partial a_1} = -2 \sum_{k=0}^{n} (f_k - a_0 e^{a_1 x_k}) a_0\, x_k e^{a_1 x_k},$$

što je nelinearan sustav jednadžbi.

S druge strane, ako logaritmiramo relaciju

$$\varphi(x) = a_0 e^{a_1 x},$$

dobivamo

$$\ln \varphi(x) = \ln(a_0) + a_1 x.$$

Moramo logaritmirati još i vrijednosti funkcije f *u točkama* x_k*, pa uz supstitucije*

$$h(x) = \ln f(x), \quad h_k = h(x_k) = \ln f_k, \quad k = 0, \ldots, n,$$

i

$$\psi(x) = \ln \varphi(x) = b_0 + b_1 x,$$

gdje je

$$b_0 = \ln a_0, \quad b_1 = a_1,$$

dobivamo linearni problem najmanjih kvadrata

$$\widetilde{S} = \widetilde{S}(b_0, b_1) = \sum_{k=0}^{n} (h_k - \psi(x_k))^2 = \sum_{k=0}^{n} (h_k - b_0 - b_1 x_k)^2 \to \min.$$

Na kraju, iz rješenja b_0 *i* b_1 *ovog problema, lako očitamo* a_0 *i* a_1

$$a_0 = e^{b_0}, \quad a_1 = b_1.$$

Uočite da ovako dobiveno rješenje uvijek daje pozitivan a_0*, tj. linearizacijom dobivena funkcija* $\varphi(x)$ *će uvijek biti veća od* 0*. Jasno da to nije "pravo" rješenje za*

sve početne podatke (x_k, f_k). No, možemo li na ovako opisani način linearizirati sve početne podatke? Očito je **ne***, jer mora biti $f_k > 0$ da bismo mogli logaritmirati.*

Ipak, i kad su neki $f_k \leq 0$, nije teško, korištenjem translacije svih podataka dobiti $f_k +$ translacija > 0, pa onda nastaviti postupak linearizacije. Pokušajte korektno formulirati linearizaciju.

Na kreju odjeljka dajemo i nekoliko funkcija koje su često u upotrebi i njihovih standardnih linearizacija u problemu najmanjih kvadrata.

(a) Funkcija

$$\varphi(x) = a_0 x^{a_1}$$

linearizira se logaritmiranjem

$$\psi(x) = \log \varphi(x) = \log(a_0) + a_1 \log x, \quad h_k = \log f_k, \quad k = 0, \ldots, n.$$

Drugim riječima, dobili smo linearni problem najmanjih kvadrata

$$\widetilde{S} = \widetilde{S}(b_0, b_1) = \sum_{k=0}^{n} (h_k - b_0 - b_1 \log(x_k))^2 \to \min,$$

gdje je

$$b_0 = \log(a_0), \quad b_1 = a_1.$$

U ovom slučaju, da bismo mogli provesti linearizaciju, moraju biti i $x_k > 0$ i $f_k > 0$.

(b) Funkcija

$$\varphi(x) = \frac{1}{a_0 + a_1 x}$$

linearizira se na sljedeći način

$$\psi(x) = \frac{1}{\varphi(x)} = a_0 + a_1 x, \quad h_k = \frac{1}{f_k}, \quad k = 0, \ldots, n.$$

Pripadni linearni problem najmanjih kvadrata je

$$\widetilde{S} = \widetilde{S}(a_0, a_1) = \sum_{k=0}^{n} (h_k - a_0 - a_1 x_k)^2 \to \min\,.$$

(c) Funkciju

$$\varphi(x) = \frac{x}{a_0 + a_1 x}$$

možemo linearizirati na više načina. Prvo, možemo staviti

$$\psi(x) = \frac{1}{\varphi(x)} = a_0 \frac{1}{x} + a_1, \quad h_k = \frac{1}{f_k}, \quad k = 0, \ldots, n.$$

Pripadni linearni problem najmanjih kvadrata je

$$\widetilde{S} = \widetilde{S}(a_0, a_1) = \sum_{k=0}^{n} (h_k - a_0 \frac{1}{x_k} - a_1)^2 \to \min .$$

Može se koristiti i sljedeći način

$$\psi(x) = \frac{x}{\varphi(x)} = a_0 + a_1 x, \quad h_k = \frac{x_k}{f_k}, \quad k = 0, \ldots, n.$$

Pripadni linearni problem najmanjih kvadrata je

$$\widetilde{S} = \widetilde{S}(a_0, a_1) = \sum_{k=0}^{n} (h_k - a_0 - a_1 x_k)^2 \to \min .$$

(d) Funkcija

$$\varphi(x) = \frac{1}{a_0 + a_1 e^{-x}}$$

linearizira se stavljanjem

$$\psi(x) = \frac{1}{\varphi(x)} = a_0 + a_1 e^{-x}, \quad h_k = \frac{1}{f_k}, \quad k = 0, \ldots, n.$$

Pripadni linearni problem najmanjih kvadrata je

$$\widetilde{S} = \widetilde{S}(a_0, a_1) = \sum_{k=0}^{n} (h_k - a_0 - a_1 e^{-x_k})^2 \to \min .$$

7.5.2. Matrična formulacija linearnog problema najmanjih kvadrata

Da bismo formirali matrični zapis linearnog problema najmanjih kvadrata, moramo preimenovati nepoznanice, naprosto zato da bismo matricu, vektor desne strane i nepoznanice u linearnom sustavu pisali u uobičajenoj formi (standardno su nepoznanice $x_1, x_2, \ldots,$, a ne $a_0, a_1, \ldots,$).

Pretpostavimo da imamo skup mjerenih podataka (t_k, y_k), $k = 1, \ldots, n$, i želimo taj model aproksimirati funkcijom oblika $\varphi(t)$. Ako je $\varphi(t)$ linearna, tj. ako je

$$\varphi(t) = x_1 \varphi_1(t) + \cdots + x_m \varphi_m(t),$$

onda bismo željeli pronaći parametre x_j tako da mjereni podaci (t_k, y_k) zadovoljavaju

$$y_k = \sum_{j=1}^{m} x_j \varphi_j(t_k), \quad k = 1, \ldots, n.$$

Ako označimo

$$a_{kj} = \varphi_j(x_k), \quad b_k = y_k,$$

onda prethodne jednadžbe možemo u matričnom obliku pisati kao

$$Ax = b.$$

Ako je mjerenih podataka više nego parametara, tj. ako je $n > m$, onda ovaj sustav jednadžbi ima više jednadžbi nego nepoznanica, pa je preodređen.

Kao što smo već u uvodu rekli, postoji mnogo načina da se odredi "najbolje" rješenje, ali zbog statističkih razloga to je često metoda najmanjih kvadrata, tj. određujemo x tako da minimizira grešku $r = Ax - b$ (r se često zove rezidual)

$$\min_x \|r\|_2 = \min_x \|Ax - b\|_2, \quad A \in \mathbb{R}^{n\times m}, \quad b \in \mathbb{R}^n, \quad x \in \mathbb{R}^m. \tag{7.5.2}$$

Ako je $\text{rang}(A) < m$, onda rješenje x ovog problema očito **nije** jedinstveno, jer mu možemo dodati bilo koji vektor iz nul-potprostora od A, a da se rezidual ne promijeni. S druge strane, među svim rješenjima x problema najmanjih kvadrata uvijek postoji jedinstveno rješenje x najmanje norme, tj. koje još minimizira i $\|x\|_2$.

7.5.3. Karakterizacija rješenja

Prvo, karakterizirajmo skup svih rješenja problema najmanjih kvadrata.

Teorem 7.5.1 *Skup svih rješenja problema najmanjih kvadrata (7.5.2) označimo s*

$$\mathcal{S} = \{x \in \mathbb{R}^m \mid \|Ax - b\|_2 = \min\}.$$

Tada je $x \in \mathcal{S}$ ako i samo ako vrijedi sljedeća relacija ortogonalnosti

$$A^T(b - Ax) = 0. \tag{7.5.3}$$

Dokaz. Pretpostavimo da $\hat{x}$ zadovoljava

$$A^T\hat{r} = 0, \quad \hat{r} = b - A\hat{x}.$$

Tada za bilo koji $x \in \mathbb{R}^m$ imamo

$$r = b - Ax = \hat{r} + A\hat{x} - Ax = \hat{r} - A(x - \hat{x}).$$

Ako označimo

$$e = x - \hat{x},$$

onda je

$$\|r\|_2^2 = r^T r = (\hat{r} - Ae)^T(\hat{r} - Ae) = \hat{r}^T\hat{r} + \|Ae\|_2^2 = \|\hat{r}\|_2^2 + \|Ae\|_2^2.$$

Kako je $\|Ae\|_2 \geq 0$, vidimo da je minimalna vrijednost za $\|r\|_2$ jednaka $\|\hat{r}\|_2$, pa $x = \hat{x}$ minimizira $\|r\|_2$. Time je pokazano da je $\hat{x} \in \mathcal{S}$.

Pokažimo obratnu implikaciju. Pretpostavimo da je

$$A^T\hat{r} = z \neq 0$$

i uzmimo

$$x = \hat{x} + \varepsilon z.$$

Tada je

$$r = \hat{r} - \varepsilon Az$$

i

$$\|r\|_2^2 = r^T r = \hat{r}^T\hat{r} - 2\varepsilon z^T z + \varepsilon^2 (Az)^T(Az) < \hat{r}^T\hat{r}$$

za dovoljno mali ε, pa $\hat{x}$ nije rješenje u smislu najmanjih kvadrata, što znači da nije u $\mathcal{S}$. ■

Relacija (7.5.3) često se zove sustav normalnih jednadžbi i uobičajeno se piše u obliku

$$A^T Ax = A^T b.$$

Matrica $A^T A$ je simetrična i pozitivno semidefinitna, a sustav normalnih jednadžbi je uvijek konzistentan, jer je

$$A^T b \in \mathcal{R}(A^T) = \mathcal{R}(A^T A).$$

Štoviše, vrijedi i sljedeća propozicija.

Propozicija 7.5.1 *Matrica $A^T A$ je pozitivno definitna ako i samo ako su stupci od A linearno nezavisni, tj. ako je* $\operatorname{rang}(A) = m$.

Dokaz. Ako su stupci od $A = [a_1, \ldots, a_m]$ linearno nezavisni, tada za svaki $x \neq 0$ vrijedi (definicija linearne nezavisnosti)

$$Ax = x_1 a_1 + \cdots + x_m a_m \neq 0,$$

pri čemu su x_j komponente od x. Za takav x je

$$x^T A^T Ax = \|Ax\|_2^2 > 0,$$

tj. $A^T A$ je pozitivno definitna.

S druge strane, ako su stupci linearno zavisni, tada postoji $x_0 \neq 0$ takav da je $Ax_0 = 0$, pa je za takav x_0

$$x_0^T A^T Ax_0 = 0.$$

Ako je x takav da je $Ax \neq 0$, onda je $x^T A^T Ax \geq 0$, pa je $A^T A$ pozitivno semidefinitna. Drugim riječima, $A^T A$ je općenito pozitivno semidefinitna, a pozitivnu definitnost garantira tek puni stupčani rang od A. ■

Iz prethodne propozicije slijedi da uvjet $\text{rang}(A) = m$, osigurava postojanje jedinstvenog rješenja problema najmanjih kvadrata. U tom slučaju rješenje i pripadni rezidual zadovoljavaju

$$x = (A^T A)^{-1} A^T b, \quad r = b - A(A^T A)^{-1} A^T b.$$

Ako je $S \subset \mathbb{R}^n$ potprostor, onda je $P_S \in \mathbb{R}^{n\times n}$ **ortogonalni projektor** na S, ako je $\mathcal{R}(P_S) = S$ i

$$P_S^2 = P_S, \quad P_S^T = P_S.$$

Nadalje, vrijedi i

$$(I - P_S)^2 = I - P_S, \quad (I - P_S)P_S = 0,$$

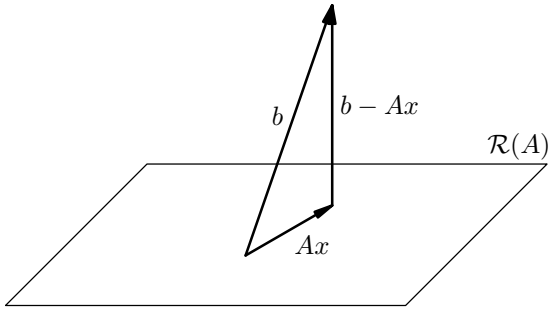
pa je $I - P_S$ projektor na ortogonalni komplement od S.

Tvrdimo da postoji jedinstveni ortogonalni projektor na S. Pretpostavimo da postoje dva ortogonalna projektora P_1 i P_2. Za sve $z \in \mathbb{R}^n$, onda vrijedi

$$\begin{aligned}\|(P_1 - P_2)z\|_2^2 &= z^T(P_1 - P_2)^T(P_1 - P_2)z = z^T(P_1^T P_1 - P_2^T P_1 - P_1^T P_2 + P_2^T P_2)z \\ &= z^T(P_1 - P_2 P_1 - P_1 P_2 + P_2)z \\ &= z^T P_1(I - P_2)z + z^T P_2(I - P_1)z = 0.\end{aligned}$$

Odatle odmah slijedi da je $P_1 = P_2$, tj. ortogonalni je projektor jedinstven.

Iz geometrijske interpretacije problema najmanjih kvadrata odmah vidimo da je Ax ortogonalna projekcija vektora b na $\mathcal{R}(A)$.



Također

$$r = (I - P_{\mathcal{R}(A)})b$$

i u slučaju punog ranga matrice A vrijedi

$$P_{\mathcal{R}(A)} = A(A^T A)^{-1} A^T.$$

Ako je $\text{rang}(A) < m$, onda A ima netrivijalni nul-potprostor i rješenje problema najmanjih kvadrata nije jedinstveno. Istaknimo jedno od rješenja $\hat{x}$. Skup svih rješenja $\mathcal{S}$ onda možemo opisati kao

$$\mathcal{S} = \{x = \hat{x} + z \mid z \in \mathcal{N}(A)\}.$$

Ako je $\hat{x} \perp \mathcal{N}(A)$, onda je

$$\|x\|_2^2 = \|\hat{x}\|_2^2 + \|z\|_2^2,$$

pa je $\hat{x}$ jedinstveno rješenje problema najmanjih kvadrata koje ima minimalnu 2-normu.

7.5.4. Numeričko rješavanje problema najmanjih kvadrata

Postoji nekoliko načina rješavanja problema najmanjih kvadrata u praksi. Obično se koristi jedna od sljedećih metoda:

1. sustav normalnih jednadžbi,
2. QR faktorizacija,
3. dekompozicija singularnih vrijednosti,
4. transformacija u linearni sustav.

Sustav normalnih jednadžbi

Prva od navedenih metoda je najbrža, ali je najmanje točna. Koristi se kad je A^TA pozitivno definitna i kad je njena uvjetovanost mala. Matrica A^TA rastavi se faktorizacijom Choleskog, a zatim se riješi linearni sustav

$$A^TAx = A^Tb.$$

Ukupan broj aritmetičkih operacija za računanje A^TA, A^Tb, te zatim faktorizaciju Choleskog je $nm^2+\frac{1}{3}m^3+O(m^2)$. Budući da je $n \geq m$, onda je prvi član dominantan u ovom izrazu, a potječe od formiranja A^TA.

Korištenje QR faktorizacije u problemu najmanjih kvadrata

Ponovno, pretpostavimo da je A^TA pozitivno definitna. Polazimo od rješenja problema najmanjih kvadrata dobivenog iz sustava normalnih jednadžbi

$$x = (A^TA)^{-1}A^Tb.$$

Zatim napišemo QR faktorizaciju matrice A

$$A = QR = Q_0R_0,$$

gdje je Q_0 ortogonalna matrica tipa (n, m), a R_0 trokutasta tipa (m, m) i uvrstimo u rješenje. Dobivamo

$$\begin{aligned} x &= (A^TA)^{-1}A^Tb = (R_0^TQ_0^TQ_0R_0)^{-1}R_0^TQ_0^Tb \\ &= (R_0^TR_0)^{-1}R_0^TQ_0^Tb = R_0^{-1}R_0^{-T}R_0^TQ_0^Tb = R_0^{-1}Q_0^Tb, \end{aligned}$$

tj. x se dobiva primjenom "invertirane" skraćene QR faktorizacije od A na b (po analogiji s rješavanjem linearnih sustava, samo što A ne mora imati inverz).

Preciznije, da bismo našli x, rješavamo trokutasti linearni sustav

$$R_0 x = Q_0^T b.$$

Na ovakav se način najčešće rješavaju problemi najmanjih kvadrata. Nije teško pokazati da je cijena računanja $2nm^2 - \frac{2}{3}m^3$, što je dvostruko više nego za sustav normalnih jednadžbi kad je $n \gg m$, a približno jednako za $m = n$.

QR faktorizacija može se koristiti i za problem najmanjih kvadrata kad matrica A nema puni stupčani rang, ali tada se koristi QR faktorizacija sa stupčanim pivotiranjem (na prvo mjesto dovodi se stupac čiji "radni dio" ima najveću normu). Zašto baš tako? Ako matrica A ima rang $r < m$, onda njena QR faktorizacija ima oblik

$$A = QR = Q \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

gdje je R_{11} nesingularna reda r, a R_{12} neka $r \times (m - r)$ matrica. Zbog grešaka zaokruživanja, umjesto pravog R, izračunamo

$$R' = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \\ 0 & 0 \end{bmatrix}.$$

Naravno, željeli bismo da je $\|R_{22}\|_2$ vrlo mala, reda veličine $\varepsilon\|A\|_2$, pa da je možemo "zaboraviti", tj. staviti $R_{22} = 0$ i tako odrediti rang od A. Nažalost, to nije uvijek tako. Na primjer, bidijagonalna matrica

$$A = \begin{bmatrix} \frac{1}{2} & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \frac{1}{2} \end{bmatrix}$$

je skoro singularna ($\det(A) = 2^{-n}$), njena QR faktorizacija je $Q = I$, $R = A$, i nema niti jednog R_{22} koji bi bio po normi malen.

Zbog toga koristimo pivotiranje, koje R_{11} pokušava držati što bolje uvjetovanim, a R_{22} po normi što manjim.

Dekompozicija singularnih vrijednosti i problem najmanjih kvadrata

Vjerojatno jedna od najkorisnijih dekompozicija i s teoretske strane (za dokazivanje činjenica) i s praktične strane je dekompozicija singularnih vrijednosti (engl. "singular value decomposition") ili, skraćeno, SVD.

Teorem 7.5.2 *Ako A ima puni rang, onda je rješenje problema najmanjih kvadrata*

$$\min_x \|Ax - b\|_2$$

jednako

$$x = V\Sigma^{-1}U^T b,$$

tj. dobiva se primjenom "invertiranog" skraćenog SVD-a od A na b.

Dokaz. Vrijedi

$$\|Ax - b\|_2^2 = \|U\Sigma V^T x - b\|_2^2.$$

Budući da je A punog ranga, to je i Σ. Zbog unitarne ekvivalencije 2-norme, vrijedi

$$\|U\Sigma V^T x - b\|_2^2 = \|\widehat{U}^T(U\Sigma V^T x - b)\|_2^2 = \left\| \begin{bmatrix} U^T \\ U_0^T \end{bmatrix} (U\Sigma V^T x - b) \right\|_2^2$$
$$= \left\| \begin{bmatrix} \Sigma V^T x - U^T b \\ -U_0^T b \end{bmatrix} \right\|_2^2 = \|\Sigma V^T x - U^T b\|_2^2 + \|U_0^T b\|_2^2.$$

Prethodni izraz se minimizira ako je prvi član jednak 0, tj. ako je

$$x = V\Sigma^{-1}U^T b.$$

Usput dobivamo i vrijednost minimuma $\min_x \|Ax - b\|_2 = \|U_0^T b\|_2$. ■

Uočite da u prethodnom teoremu rješenje problema najmanjih kvadrata kad je matrica A punog ranga. Uobičajeno se SVD primjenjuje u metodi najmanjih kvadrata i kad matrica A nema puni stupčani rang. Rješenja su istog oblika (sjetite se, više ih je), samo što moramo znati izračunati "inverz" matrice Σ kad ona nije regularna, tj. kad ima neke nule na dijagonali. Takav inverz zove se generalizirani inverz i označava sa Σ^+ ili $\Sigma^\dagger$. U slučaju da je

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix},$$

pri čemu je Σ_1 regularna, onda je

$$\Sigma^+ = \begin{bmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

Još preciznije, za problem najmanjih kvadrata tada vrijedi sljedeća propozicija.

Propozicija 7.5.2 *Neka matrica A ima rang $r < m$. Rješenje x koje minimizira $\|Ax - b\|_2$ može se karakterizirati na sljedeći način. Neka je $A = U\Sigma V^T$ SVD od A i neka je*

$$A = U\Sigma V^T = [U_1, U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} [V_1, V_2]^T = U_1\Sigma_1 V_1^T,$$

gdje je Σ_1 nesingularna, reda r, a matrice U_1 i V_1 imaju r stupaca. Neka je

$$\sigma := \sigma_{\min}(\Sigma_1),$$

najmanja ne-nula singularna vrijednost od A. Tada se sva rješenja problema najmanjih kvadrata mogu napisati u formi

$$x = V_1\Sigma_1^{-1}U_1^T b + V_2 z,$$

gdje je z proizvoljni vektor. Rješenje x koje ima minimalnu 2-normu je ono za koje je $z = 0$, tj.

$$x = V_1\Sigma_1^{-1}U_1^T b,$$

i vrijedi ocjena

$$\|x\|_2 \leq \frac{\|b\|_2}{\sigma}.$$

Dokaz. Nadopunimo matricu $[U_1, U_2]$ stupcima matrice U_3 do ortogonalne matrice reda n, i označimo je s $\widehat{U}$. Korištenjem unitarne invarijantnosti 2-norme, dobivamo

$$\begin{aligned}\|Ax-b\|_2^2 = \|\widehat{U}^T(Ax-b)\|_2^2 &= \left\| \begin{bmatrix} U_1^T \\ U_2^T \\ U_3^T \end{bmatrix} (U_1\Sigma_1V_1^Tx - b) \right\|_2^2 \\ &= \left\| \begin{bmatrix} \Sigma_1V_1^Tx - U_1^Tb \\ -U_2^Tb \\ -U_3^Tb \end{bmatrix} \right\|_2^2 = \|\Sigma_1V_1^Tx - U_1^Tb\|_2^2 + \|U_2^Tb\|_2^2 + \|U_3^Tb\|_2^2.\end{aligned}$$

Očito, izraz je minimiziran kad je prva od tri norme u posljednjem redu jednaka 0, tj. ako je

$$\Sigma_1V_1^Tx = U_1^Tb,$$

ili

$$x = V_1\Sigma_1^{-1}U_1^Tb.$$

Stupci matrica V_1 i V_2 su međusobno ortogonalni, pa je $V_1^TV_2z = 0$ za sve vektore z. Odavde vidimo da x ostaje rješenje problema najmanjih kvadrata i kad mu dodamo V_2z, za bilo koji z, tj. ako je

$$x = V_1\Sigma_1^{-1}U_1^Tb + V_2z.$$

To su ujedno i sva rješenja, jer stupci matrice V_2 razapinju nul-potprostor $\mathcal{N}(A)$. Osim toga, zbog spomenute ortogonalnosti vrijedi i

$$\|x\|_2^2 = \|V_1\Sigma_1^{-1}U_1^Tb\|_2^2 + \|V_2z\|_2^2,$$

a to je minimalno za $z = 0$. Na kraju, za to minimalno rješenje vrijedi ocjena

$$\|x\|_2 = \|V_1\Sigma_1^{-1}U_1^Tb\|_2 = \|\Sigma_1^{-1}U_1^Tb\|_2 \leq \frac{\|U_1^Tb\|_2}{\sigma} = \frac{\|b\|_2}{\sigma}.$$

Primjerom se lako pokazuje da je ova ocjena dostižna. ■

Rješenje problema najmanjih kvadrata korištenjem SVD-a je najstabilnije, a može se pokazati da je, za $n \gg m$, njegovo trajanje približno jednako kao i trajanje rješenja korištenjem QR-a. Za manje n, trajanje je približno $4nm^2 - \frac{4}{3}m^3 + O(m^2)$.

Transformiranje problema najmanjih kvadrata na linearni sustav

Ako matrica A ima puni rang po stupcima, onda problem najmanjih kvadrata možemo transformirati i na linearni sustav različit od sustava normalnih jednadžbi. Simetrični linearni sustav

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

ekvivalentan je sustavu normalnih jednadžbi. Ako napišemo prvu i drugu blok-komponentu

$$r + Ax = b, \quad A^T r = 0,$$

onda uvrštavanjem r-a iz prve blok-jednadžbe u drugu dobivamo sustav

$$A^T(b - Ax) = 0.$$

Prvi sustav ima bitno manji raspon elemenata od sustava normalnih jednadžbi. Osim toga, ako je matrica A loše uvjetovana, kod tog sustava možemo lakše koristiti iterativno profinjavanje rješenja.

7.6. Opći oblik metode najmanjih kvadrata

Nakon što smo napravili osnovni oblik diskretne metode najmanjih kvadrata, na sličan način možemo riješiti i opći problem aproksimacije po metodi najmanjih kvadrata, tj. u 2-normi. Dovoljno je uočiti da je diskretna 2-norma generirana običnim euklidskim skalarnim produktom na konačno dimenzionalnim prostorima. Po istom principu, u općem slučaju, radimo na nekom unitarnom prostoru s nekim skalarnim produktom, a pripadna norma je generirana tim skalarnim produktom.

Na početku zgodno je uvesti oznake koje nam omogućavaju da diskretni i neprekidni slučaj analiziramo odjednom, u istom općem okruženju unitarnih prostora.

7.6.1. Težinski skalarni produkti

Unitarni prostor $\mathcal{U}$ je vektorski prostor na kojem je definiran skalarni produkt.

7.7. Familije ortogonalnih funkcija

Za dvije funkcije reći ćemo da su ortogonalne, ako je njihov skalarni produkt jednak 0. Na primjer, za neprekidnu ili diskretnu mjeru $d\lambda$, te funkcije u i v koje imaju konačnu normu možemo definirati skalarni produkt kao

$$\int_{\mathbb{R}} u(x)\, v(x)\, d\lambda.$$

Postoji mnogo familija ortogonalnih funkcija. Evo nekoliko primjera takvih familija (sistema).

- Ortogonalni polinomi;
- Trigonometrijski polinomi.

7.8. Neka svojstva ortogonalnih polinoma

Ortogonalni polinomi imaju još i niz dodatnih "dobrih" svojstava, zbog kojih se mogu konstruktivno primijeniti u raznim granama numeričke matematike. Sljedeći niz teorema sadrži samo neka osnovna svojstva koja ćemo kasnije iskoristiti za konstrukciju algoritama. Sva ta svojstva su direktna posljedica ortogonalnosti polinoma i ne ovise bitno o tome da li je skalarni produkt diskretan ili kontinuiran.

Međutim, na ovom mjestu je zgodno napraviti razliku između diskretnih i kontinuiranih skalarnih produkata, prvenstveno radi jednostavnosti iskaza, dokaza i kasnijeg pozivanja na ove teoreme. Pažljivije čitanje će samo potvrditi da bitne razlike nema.

Standardno ćemo promatrati neprekidni skalarni produkt

$$\langle u, v\rangle = \int_a^b w(x)u(x)v(x)\, dx$$

generiran težinskom funkcijom $w \geq 0$ na $[a, b]$. Ako svi polinomi pripadaju odgovarajućem prostoru kvadratno integrabilnih funkcija, onda postoji pripadna familija ortogonalnih polinoma koju označavamo s $\{p_n(x) \mid n \geq 0\}$. Dogovorno smatramo da je stupanj polinoma p_n baš jednak n, za svaki $n \geq 0$.

Paralelno ćemo promatrati i diskretni skalarni produkt

$$\langle u, v\rangle = \sum_{i=0}^{n} w_i u(x_i)v(x_i)$$

generiran međusobno različitim čvorovima $x_0, \ldots, x_n$ i pripadnim pozitivnim težinama $w_0, \ldots, w_n$. Pripadni unitarni prostor "funkcija" na zadanoj mreži čvorova (izomorfno) sadrži sve polinome stupnja manjeg ili jednakog n, pa sigurno postoji pripadna baza ortogonalnih polinoma koju označavamo s $\{p_k(x) \mid 0 \leq k \leq n\}$. Opet uzimamo je stupanj polinoma p_k baš jednak k, za svaki $k \in \{0, \ldots, n\}$.

Teorem 7.8.1 *Neka je $\{p_n(x) \mid n \geq 0\}$ familija ortogonalnih polinoma na intervalu $[a, b]$ s težinskom funkcijom $w(x) \geq 0$. Ako je f polinom stupnja m, tada vrijedi*

$$f = \sum_{n=0}^{m} \frac{\langle f, p_n \rangle}{\langle p_n, p_n \rangle} p_n.$$

Dokaz. Prvo, pokažimo da se svaki polinom može napisati kao kombinacija ortogonalnih polinoma stupnja manjeg ili jednakog njegovom.

Dokaz ide korištenjem Gram–Schmidtove ortogonalizacije. Pokažimo, redom, da se monomi $\{1, x, x^2, \ldots\}$ mogu prikazati pomoću ortogonalnih polinoma.

Ako je stupanj ortogonalnog polinoma 0, on je nužno konstanta različita od nule, tj. vrijedi

$$p_0(x) = c_{0,0}, \quad c_{0,0} \neq 0,$$

pa se prvi monom 1 može napisati kao

$$1 = \frac{1}{c_{0,0}} p_0(x).$$

Za polinome stupnja jedan, konstrukcija slijedi iz Gram–Schmidtovog procesa ortogonalizacije sustava funkcija $\{1, x\}$

$$p_1(x) = c_{1,1}x + c_{1,0}p_0(x), \quad c_{1,1} \neq 0,$$

tj. vrijedi

$$x = \frac{1}{c_{1,1}} [p_1(x) - c_{1,0}p_0(x)].$$

Korištenjem indukcije u Gram–Schmidtovom procesu na $\{1, x, x^2, \ldots, x^n\}$, dobivamo

$$p_n(x) = c_{n,n}x^n + c_{n,n-1}p_{n-1}(x) + \cdots + c_{n,0}p_0(x), \quad c_{n,n} \neq 0,$$

gdje su $p_0, p_1, \ldots, p_{n-1}$ dobiveni ortogonalizacijom iz $\{1, x, \ldots, x^{n-1}\}$, pa je

$$x^n = \frac{1}{c_{n,n}} [p_n(x) - c_{n,n-1}p_{n-1}(x) - \cdots - c_{n,0}p_0(x)].$$

Neka je f bilo koji polinom stupnja (manjeg ili jednakog) m, za neki $m \in \mathbb{N}_0$. Tada se f može napisati kao linearna kombinacija monoma $\{1, x, \ldots, x^m\}$, prikazom

u standardnoj bazi. Budući da se svaki monom može napisati kao linearna kombinacija ortogonalnih polinoma stupnja manjeg ili jednakog od stupnja tog monoma, odmah slijedi da se i f može napisati kao neka linearna kombinacija ortogonalnih polinoma stupnjeva manjih ili jednakih m, tj. da vrijedi

$$f = \sum_{j=0}^{m} b_j p_j.$$

Ostaje samo odrediti koeficijente b_j. Množenjem prethodne relacije težinskom funkcijom w, pa polinomom p_n, a zatim integriranjem na $[a, b]$, tj. skalarnim množenjem s p_n, dobivamo

$$\langle f, p_n \rangle = \sum_{j=0}^{m} b_j \langle p_j, p_n \rangle = b_n \langle p_n, p_n \rangle,$$

koristeći ortogonalnost p_j i p_n, za $j \neq n$. Odatle odmah slijedi da je

$$b_n = \frac{\langle f, p_n \rangle}{\langle p_n, p_n \rangle},$$

jer je $\|p_n\|^2 = \langle p_n, p_n \rangle > 0$. ■

Razvoj polinoma f stupnja m iz prethodnog teorema možemo napisati i tako da suma ide do ∞, a ne do m, samo su svi dodatni koeficijenti $b_n = 0$, za $n > m$. To je posljedica sljedeće tvrdnje.

Korolar 7.8.1 *Ako je f polinom stupnja manjeg ili jednakog $m - 1$, onda je*

$$\langle f, p_m \rangle = 0,$$

tj. p_m je okomit na f. Dakle, p_m je okomit na sve polinome stupnja strogo manjeg od m.

Dokaz. Po prethodnom teoremu, f se može razviti po ortogonalnim polinomima stupnja manjeg ili jednakog $m - 1$

$$f(x) = \sum_{n=0}^{m-1} b_n p_n(x).$$

Množenjem s $w(x)p_m(x)$, te integriranjem, dobivamo da je

$$\langle f, p_m \rangle = \sum_{n=0}^{m-1} b_n \langle p_n, p_m \rangle = 0,$$

zbog svojstva ortogonalnosti ortogonalnih polinoma $\langle p_n, p_m \rangle = 0$, za $n \neq m$. ■

Teorem 7.8.2 *Neka je $\{p_n(x) \mid n \geq 0\}$ familija ortogonalnih polinoma na intervalu $[a, b]$ s težinskom funkcijom $w(x) \geq 0$. Tada svaki polinom p_n ima točno n različitih (jednostrukih) realnih nultočaka na otvorenom intervalu (a, b).*

Dokaz. Neka su $x_1, x_2, \ldots, x_m$ sve nultočke polinoma p_n za koje vrijedi:

- $a < x_i < b$,
- $p_n(x)$ mijenja predznak u x_i.

Budući da je p_n stupnja n, po osnovnom teoremu algebre, polinom p_n ima ukupno n nultočaka, pa onih koje zadovoljavaju prethodna dva svojstva ima manje ili jednako n. Pretpostavimo da je nultočaka koje zadovoljavaju tražena dva svojstva striktno manje od n, tj. $m < n$. Pokažimo da je to nemoguće.

Definiramo polinom

$$B(x) = (x - x_1) \cdots (x - x_m).$$

Po definiciji točaka $x_1, \ldots, x_m$, polinom

$$p_n(x)B(x) = (x - x_1) \cdots (x - x_m)p_n(x)$$

ne mijenja znak prolaskom kroz točke $x_1, \ldots, x_m$, tj. čitav polinom ne mijenja znak na (a, b). Preciznije, to implicira oblik funkcije p_n

$$p_n(x) = h(x)(x - x_1)^{r_1} \cdots (x - x_m)^{r_m},$$

pri čemu moraju biti svi r_i neparni, a $h(x)$ ne smije promijeniti predznak na (a, b). Množenjem s $B(x)$, dobivamo

$$p_n(x)B(x) = h(x)(x - x_1)^{r_1+1} \cdots (x - x_m)^{r_m+1}.$$

Nadalje, vrijedi

$$\int_a^b w(x)B(x)p_n(x)\,dx \neq 0,$$

budući da je to integral nenegativne funkcije. S druge je strane taj integral skalarni produkt od B (polinom stupnja $m < n$) i sa p_n (polinom stupnja n), pa je po prethodnom korolaru

$$\int_a^b w(x)B(x)p_n(x)\,dx = \langle B, p_n \rangle = 0.$$

To je, očito kontradikcija, pa je pretpostavka o stupnju polinoma B bila pogrešna, tj. mora biti $m = n$. Budući da p_n ima točno n nultočaka $x_1, \ldots, x_n$ u kojima mijenja predznak, one moraju biti jednostruke, jer je $p_n'(x_i) \neq 0$. ■

Neka je ponovno zadana familija ortogonalnih polinoma na intervalu $[a, b]$ i neka su prva dva koeficijenta funkcije p_n jednaki

$$p_n(x) = A_n x^n + B_n x^{n-1} + \cdots.$$

Također, tada p_n možmo napisati i kao

$$p_n(x) = A_n(x - x_{n,1})(x - x_{n,2}) \cdots (x - x_{n,n}).$$

Definiramo također i

$$a_n = \frac{A_{n+1}}{A_n}, \quad \gamma_n = \langle p_n, p_n \rangle > 0.$$

Teorem 7.8.3 (tročlana rekurzija) *Neka je* $\{p_n(x) \mid n \geq 0\}$ *familija ortogonalnih polinoma na intervalu* $[a, b]$ *s težinskom funkcijom* $w(x) \geq 0$. *Tada za* $n \geq 1$ *vrijedi rekurzija*

$$p_{n+1}(x) = (a_n x + b_n)p_n(x) - c_n p_{n-1}(x),$$

pri čemu su

$$b_n = a_n \left(\frac{B_{n+1}}{A_{n+1}} - \frac{B_n}{A_n} \right), \quad c_n = \frac{A_{n+1}A_{n-1}}{A_n^2} \cdot \frac{\gamma_n}{\gamma_{n-1}}.$$

Dokaz. Promatrajmo polinom

$$\begin{aligned} G(x) &= p_{n+1}(x) - a_n x p_n(x) \\ &= (A_{n+1}x^{n+1} + B_{n+1}x^n + \cdots) - \frac{A_{n+1}}{A_n} x (A_n x^n + B_n x^{n-1} + \cdots) \\ &= \left(B_{n+1} - \frac{A_{n+1}B_n}{A_n} \right) x^n + \cdots. \end{aligned}$$

Očito, polinom G je stupnja manjeg ili jednakog n, pa ga možemo napisati kao linearnu kombinaciju ortogonalnih polinoma stupnja manjeg ili jednakog n, tj.

$$G(x) = d_n p_n(x) + \cdots + d_0 p_0(x)$$

za neki skup konstanti d_i. Računanjem d_i izlazi

$$d_i = \frac{\langle G, p_i \rangle}{\langle p_i, p_i \rangle} = \frac{1}{\gamma_i}(\langle p_{n+1}, p_i \rangle - a_n \langle x p_n, p_i \rangle).$$

Budući da je $\langle p_{n+1}, p_i \rangle = 0$ za $i \leq n$ i da za $i \leq n - 2$ vrijedi

$$\langle x p_n, p_i \rangle = \int_a^b w(x) p_n(x) x p_i(x)\, dx = 0,$$

zaključujemo da je stupanj polinoma $x p_i(x)$ manji ili jednak $n - 1$. Kombiniranjem ta dva rezultata, dobivamo

$$d_i = 0 \quad \text{za } 0 \leq i \leq n - 2,$$

pa je zbog toga

$$\begin{aligned} G(x) &= d_n p_n(x) + d_{n-1} p_{n-1}(x) \\ p_{n+1}(x) &= (a_n x + d_n)p_n(x) + d_{n-1} p_{n-1}(x). \end{aligned}$$

Ostaje još samo pokazati koliki su koeficijenti d_{n-1} i d_n. Iz prve od dvije prethodne relacija, uspoređivanjem vodećih koeficijenata funkcije G i vodećih koeficijenata funcije s desne strane, dobivamo relaciju za d_n. ■

Teorem 7.8.4 (Christoffel–Darbouxov identitet) *Neka je* $\{p_n(x) \mid n \geq 0\}$ *familija ortogonalnih polinoma na intervalu* $[a, b]$ *s težinskom funkcijom* $w(x) \geq 0$. *Za njih vrijedi sljedeći identitet*

$$\sum_{k=0}^{n} \frac{p_k(x)p_k(y)}{\gamma_k} = \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{a_n\gamma_n(x-y)}.$$

Dokaz. Manipulacijom tročlane rekurzije. ■

7.9. Trigonometrijske funkcije

Trigonometrijske funkcije

$$\{1, \cos x, \cos 2x, \cos 3x, \ldots, \sin x, \sin 2x, \sin 3x, \ldots\}$$

čine ortogonalnu familiju funkcija na intervalu $[0, 2\pi]$ uz mjeru

$$d\lambda = \begin{cases} dx & \text{na } [0, 2\pi], \\ 0 & \text{inače.} \end{cases}$$

Pokažimo da je to zaista istina. Neka su $k, \ell \in \mathbb{N}_0$. Tada vrijedi

$$\int_0^{2\pi} \sin kx \cdot \sin \ell x \, dx = -\frac{1}{2} \int_0^{2\pi} (\cos(k+\ell)x - \cos(k-\ell)x) \, dx.$$

U slučaju da je $k = \ell$, onda je prethodni integral jednak

$$-\frac{1}{2}\left[\frac{\sin(k+\ell)x}{k+\ell} - x\right]\Bigg|_0^{2\pi} = \pi.$$

Ako je $k \neq \ell$, onda je jednak

$$-\frac{1}{2}\left[\frac{\sin(k+\ell)x}{k+\ell} - \frac{\sin(k-\ell)x}{k-\ell}\right]\Bigg|_0^{2\pi} = 0.$$

Drugim riječima, vrijedi

$$\int_0^{2\pi} \sin kx \cdot \sin \ell x \, dx = \begin{cases} 0, & k \neq \ell, \\ \pi, & k = \ell, \end{cases} \qquad k, \ell = 1, 2, \ldots,$$

Na sličan način, pretvaranjem produkta trigonometrijskih funkcija u zbroj, možemo pokazati da je

$$\int_0^{2\pi} \cos kx \cdot \cos \ell x \, dx = \begin{cases} 0, & k \neq \ell, \\ 2\pi, & k = \ell = 0, \\ \pi, & k = \ell > 0, \end{cases} \qquad k, \ell = 0, 1, \ldots,$$

te, također, da je

$$\int_0^{2\pi} \sin kx \cdot \cos \ell x \, dx = 0, \quad k = 1, 2, \ldots, \quad \ell = 0, 1, \ldots,$$

Ako periodičku funkciju f osnovnog perioda duljine 2π želimo aproksimirati redom oblika

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx),$$

onda, množenjem odgovarajućim trigonometrijskim funkcijama i integriranjem, za koeficijente u redu formalno dobivamo

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \, \cos kx \, dx, \quad b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \, \sin kx \, dx.$$

Prethodni red poznat je pod imenom Fourierov red, a koeficijenti kao Fourierovi koeficijenti.

Posebno, ako Fourierov red odsiječemo za $k = m$ i dobijemo trigonometrijski polinom, koji je najbolja L_2 aproksimacija za f u klasi trigonometrijskih polinoma stupnja manjeg ili jednakog m, obzirom na normu

$$\|u\|_2 = \left(\int_0^{2\pi} |u(t)|^2 \, dt \right)^{1/2}.$$

7.9.1. Diskretna ortogonalnost trigonometrijskih funkcija

Umjesto neprekidne, za pripadnu mjeru možemo uzeti i diskretnu mjeru, pa umjesto integrala, dobivamo sume. Da bismo dobili željeni razvoj moramo poznavati relacije diskretne relacije ortogonalnosti.

Teorem 7.9.1 *Za trigonometrijske funkcije, na mreži od točaka* $0, 1, \ldots, N$*, uz oznaku*

$$x_k = \frac{2\pi}{N+1} kx, \quad x_\ell = \frac{2\pi}{N+1} \ell x,$$

vrijede sljedeće relacije ortogonalnosti

$$\sum_{x=0}^{N} \sin x_k \sin x_\ell = \begin{cases} 0, & k \neq \ell \ i \ k = \ell = 0, \\ (N+1)/2, & k = \ell \neq 0, \end{cases}$$

$$\sum_{x=0}^{N} \sin x_k \cos x_\ell = 0$$

$$\sum_{x=0}^{N} \cos x_k \cos x_\ell = \begin{cases} 0, & k \neq \ell, \\ (N+1)/2, & k = \ell \neq 0, \\ N+1, & k = \ell = 0, \end{cases}$$

uz uvjet da je $k + \ell \leq N$.

Dokaz. Dokažimo samo prvu relaciju. Iskoristimo formulu za pretvaranje produkta dva sinusa u zbroj trigonometrijskih funkcija. Vrijedi

$$\begin{aligned} \sin x_k \cdot \sin x_\ell &= \sin\left(\frac{2\pi}{N+1} kx\right) \cdot \sin\left(\frac{2\pi}{N+1} \ell x\right) \\ &= \frac{1}{2}\left[\cos\left(\frac{2\pi}{N+1}(k-\ell)x\right) - \cos\left(\frac{2\pi}{N+1}(k+\ell)x\right)\right]. \end{aligned}$$

Ako je $k + \ell \leq N$, onda su za $x = 0, 1, \ldots, N$ onda su argumenti prvog kosinusa s desne strane redom

$$0, \frac{2\pi}{N+1}(k-\ell), \frac{4\pi}{N+1}(k-\ell), \ldots, \frac{2N\pi}{N+1}(k-\ell).$$

Ako je $k = \ell$, onda su svi argumenti prvog kosinusa 0, pa je

$$\sum_{x=0}^{N} \cos\left(\frac{2\pi}{N+1}(k-\ell)x\right) = \sum_{x=0}^{N} \cos 0 = N+1, \quad k = \ell.$$

Za slučaj $k \neq \ell$ koristimo znanje iz kompleksne analize. Članovi $\dfrac{2\pi}{N+1}(k-\ell)$ podsjećaju na argumente $(N+1)$-og korijena iz 1 ("višak" je $(k-\ell)$). Označimo s ω_j, $j = 0, \ldots, N$ sve $(N+1)$-ve korijene iz 1,

$$\omega_j = \cos\frac{2\pi j}{N+1} + i \sin\frac{2\pi j}{N+1}. \tag{7.9.1}$$

Nadalje, označimo s

$$\omega = \cos\frac{2\pi}{N+1} + i \sin\frac{2\pi}{N+1}.$$

Primijetite da se tada svi $(N+1)$-vi korijeni iz 1 mogu, korištenjem De Moivreove formule (za potenciranje kompleksnih brojeva) napisati kao

$$\omega_j = \omega^j.$$

Očito svi ω^j zadovoljavaju jednadžbu $x^{N+1} - 1 = 0$. Iskoristimo li da su ω^j svi korijeni te jednadžbe, nju možemo napisati u faktoriziranom obliku kao

$$x^{N+1} - 1 = (x - \omega^0)(x - \omega^1) \cdots (x - \omega^N).$$

Uspoređivanjem članova uz N-tu potenciju slijeva i zdesna, dobivamo

$$0 = -\sum_{j=0}^{N} \omega^j.$$

Iskoristimo li (7.9.1) dobivamo

$$0 = \sum_{j=0}^{N} \omega^j = \sum_{j=0}^{N} \cos \frac{2\pi j}{N+1} + i \sum_{j=0}^{N} \sin \frac{2\pi j}{N+1}.$$

Budući da je kompleksan broj jednak 0, nuli moraju biti jednaki i njegov realni i njegov imaginarni dio. Drugim riječima, vrijedi

$$\sum_{j=0}^{N} \cos \frac{2\pi j}{N+1} = 0, \quad \sum_{j=0}^{N} \sin \frac{2\pi j}{N+1} = 0.$$

Vratimo se na početak. Trebali smo dokazati da je

$$\sum_{j=0}^{N} \cos \frac{2\pi j}{N+1}(k - \ell) = 0.$$

Primijetimo da su argumenti kosinusa u prethodnoj formuli argumenti $(N+1)$-og korijena iz 1, pomnoženi s $(k - \ell)$, što znači da su to argumenti od $(\omega^j)^{(k-\ell)} = (\omega^{(k-\ell)})^j$. To bi odgovaralo izboru korijena

$$\widetilde{\omega} = \cos \frac{2\pi(k - \ell)}{N+1} + i \sin \frac{2\pi(k - \ell)}{N+1}$$

umjesto ω. Odmah je jasno da vrijedi

$$\sum_{j=0}^{N} \widetilde{\omega}^j = \sum_{j=0}^{N} \omega^j,$$

pa je dokazano da je

$$\sum_{j=0}^{N} \cos \frac{2\pi j(k - \ell)}{N+1} = 0.$$

Na sličan se način pokazuje da je

$$\sum_{j=0}^{N} \cos \frac{2\pi j(k + \ell)}{N+1} = 0,$$

za $j + k \neq 0$, čime je pokazana relacija ortogonalnosti za sinuse. ■

Ovo znači da restrikcije funkcija

$$\cos \frac{2\pi}{N+1} kx, \quad \sin \frac{2\pi}{N+1} kx \tag{7.9.2}$$

pri čemu su dozvoljeni $k \in \mathbb{N}_0$ za kosinuse i $k \in N$ za sinuse, na mreži $\{0, \ldots, N\}$ možemo koristiti kao ortogonalnu familiju. Linearne kombinacije funkcija (7.9.2) zvat ćemo **trigonometrijski polinom**.

Nažalost baze takvih trigonometrijskih polinoma ovise o parnosti N.

Neparan broj točaka

Neka je zadan neparan broj točaka $\mathcal{M} = \{0, 1, \ldots, N = 2L\}$. Za bazu se tada uzima prvih $L + 1$ kosinusa (prvi je konstanta) i prvih L sinusa, a pripadna trigonometrijska aproksimacija ima oblik

$$T_N(x) = \frac{a_0}{2} + \sum_{k=1}^{L} (a_k \cos x_k + b_k \sin x_k), \tag{7.9.3}$$

pri čemu je

$$x_k = \frac{2\pi}{N+1} kx := \frac{2\pi}{2L+1} kx.$$

Koeficijenti trigonometrijskog polinom određuju se iz relacija ortogonalnosti na uobičajeni način, množenjem lijeve i desne strane u (7.9.3) izabranom funkcijom baze uz koju je odgovarajući koeficijent. Ako trigonometrijski polinom interpolira T_N interpolira funkciju f u $x \in \mathcal{M}$, tj. ako je $T_n(x) = f(x)$ onda množemjem (7.9.3) s $\cos x_\ell$ $\ell \geq 0$ i upotrebom relacija ortogonalnosti dolazimo do koeficijenata a_ℓ

$$f(x) \cos x_\ell = \frac{a_0}{2} \cos x_\ell + \sum_{k=1}^{L} a_k \cos x_k \cos x_\ell + \sum_{k=1}^{L} b_k \sin x_k \cos x_\ell.$$

Zbrajanjem po svim x dobivamo

$$\begin{aligned} \sum_{x=0}^{2L} f(x) \cos x_\ell &= \frac{a_0}{2} \sum_{x=0}^{2L} \cos 0 \cos x_\ell + \sum_{k=1}^{L} a_k \sum_{x=0}^{2L} \cos x_k \cos x_\ell + \sum_{k=1}^{L} b_k \sum_{x=0}^{2L} \sin x_k \cos x_\ell \\ &= \frac{2L+1}{2} a_\ell. \end{aligned}$$

Odatle odmah zaključujemo da je (pišući k umjesto ℓ)

$$a_k = \frac{2}{2L+1} \sum_{x=0}^{2L} f(x) \cos x_k, \quad k = 0, \ldots, L.$$

Na sličan način, množenjem sa $\sin x_\ell$, $\ell > 0$ i zbrajanjem po svim x dobivamo

$$\sum_{x=0}^{2L} f(x)\sin x_\ell = \frac{a_0}{2}\sum_{x=0}^{2L}\cos 0\sin x_\ell + \sum_{k=1}^{L} a_k \sum_{x=0}^{2L}\cos x_k \sin x_\ell + \sum_{k=1}^{L} b_k \sum_{x=0}^{2L}\sin x_k \sin x_\ell$$
$$= \frac{2L+1}{2} b_\ell.$$

Slično kao kod a_k, imamo

$$b_k = \frac{2}{2L+1}\sum_{x=0}^{2L} f(x)\sin x_k, \quad k = 1, \ldots, L.$$

Dakle, u slučaju neparnog broja točaka koeficijenti u (7.9.3) su

$$a_k = \frac{2}{2L+1}\sum_{x=0}^{2L} f(x)\cos x_k, \quad k = 0, \ldots, L,$$
$$b_k = \frac{2}{2L+1}\sum_{x=0}^{2L} f(x)\sin x_k, \quad k = 1, \ldots, L.$$

Zadatak 7.9.1 *Pokažite da za bilo koju točku x^*, ne nužno iz $\mathcal{M}$ vrijedi*

$$T_N(x^*) = \frac{1}{2L+1}\sum_{x=0}^{2L} f(x)\Big(\sum_{k=0}^{2L}\cos\Big(\frac{2\pi}{2L+1}k(x-x^*)\Big)\Big).$$

Paran broj točaka

Neka je zadan paran broj točaka $\mathcal{M} = \{0, 1, \ldots, N = 2L-1\}$. Za bazu se tada uzima prvih $L+1$ kosinusa (prvi je konstanta) i prvih $L-1$ sinusa, a pripadna trigonometrijska aproksimacija ima oblik

$$T_N(x) = \frac{a_0}{2} + \sum_{k=1}^{L-1}(a_k\cos x_k + b_k \sin x_k) + \frac{1}{2}a_L \cos x_L, \tag{7.9.4}$$

pri čemu je

$$x_k = \frac{2\pi}{N+1}kx := \frac{\pi}{L}kx.$$

Na sličan način kao kod neparnog broja točaka, koeficijenti u (7.9.4) su

$$a_k = \frac{1}{L}\sum_{x=0}^{2L-1} f(x)\cos x_k, \quad k = 0, \ldots, L,$$
$$b_k = \frac{1}{2L}\sum_{x=0}^{2L-1} f(x)\sin x_k, \quad k = 1, \ldots, L-1.$$

Zadatak 7.9.2 *Pokažite da i u slučaju neparnog i u slučaju parnog broja točaka, T_N ima period $N+1$. Zbog toga se jednostavno koristi za interpolaciju trigonometrijskih funkcija, a dovoljno je zadati samo točke x iz jednog perioda.*

Primjer 7.9.1 *Funkcija f ima period 3 i zadana je tablično s*

x_k	0	1	2
f_k	0	1	1

Nađimo trigonometrijski polinom koji interpolira f u svim točkama iz $\mathbb{Z}$, a zatim izračunajmo $T_N(1/2)$ i $T_N(3/2)$.

Budući da je $N = 2$, broj točaka je neparan, pa je

$$T_2(x) = \frac{1}{2}a_0 + a_1 \cos\frac{2\pi}{3}x + b_1 \sin\frac{2\pi}{3}x.$$

Prema formulama za koeficijente, dobivamo

$$\begin{aligned} a_0 &= \frac{2}{3}\left(0\cos 0 + 1\cdot\cos 0 + 1\cdot\cos 0\right) = \frac{4}{3} \\ a_1 &= \frac{2}{3}\left(0\cos 0 + 1\cdot\cos\frac{2\pi}{3} + 1\cdot\cos\frac{4\pi}{3}\right) = -\frac{2}{3} \\ b_1 &= \frac{2}{3}\left(0\sin 0 + 1\cdot\sin\frac{2\pi}{3} + 1\cdot\sin\frac{4\pi}{3}\right) = 0. \end{aligned}$$

Prma tome, trigonometrijski polinom koji interpolira zadane točke je

$$T_2(x) = \frac{2}{3} - \frac{2}{3}\cos\frac{2\pi}{3}x.$$

Odatle se odmah može izračunati da je

$$\begin{aligned} T_2(1/2) &= \frac{2}{3} - \frac{2}{3}\cos\frac{\pi}{3} = \frac{1}{3} \\ T_2(3/2) &= \frac{2}{3} - \frac{2}{3}\cos\pi = \frac{4}{3}. \end{aligned}$$

Metoda majnanjih kvadrata za trigonometrijske funkcije

I za metodu najmanjih kvadrata možemo koristiti trigonometrijske polinome, jer je dovoljno uzeti podskup baze prostora. Slično kao kod interpolacije biramo početni dio baze (7.9.2). Također moramo paziti na parnost/neparnost broja točaka N i na parnost/neparnost stupnja trigonometrijskog polinoma M, $M \leq N$.

Ilustrirajmo to na slučaju $N = 2L$ paran (broj točaka neparan) i $M = 2m$ paran (neparna dimenzija potprostora). Trigonometrijski polinom odgovarajućeg stupnja je

$$T_M(x) = \frac{1}{2}A_0 = \sum_{k=1}^{m}(A_k \cos x_k + B_k \sin x_k), \tag{7.9.5}$$

gdje je

$$x_k = \frac{2\pi}{N+1}kx := \frac{2\pi}{2L+1}kx.$$

Metoda najmanjih kvadrata minimizira kvadrat greške

$$S = \sum_{x=0}^{2L}(f(x) - T_M(x))^2 \to \min.$$

Tvrdimo da je rješenje problema mininizacije trigonometrijski interpolacijski polinom kojemu je

$$A_k = a_k, \quad k = 0, \ldots, m$$
$$B_k = b_k, \quad k = 1, \ldots, m,$$

a koeficijenti a_k i b_k se računaju po formulama za interpolaciju. Primijetite da u točkama interpolacije x, $x = 0, \ldots, 2L$ interpolacijski polinom ima istu vrijednost kao funkcija f, pa je dovoljno (u točkama interpolacije) uspoređivati interpolacijski trigonometrijski polinom T_N, $N = 2L$ i trigonometrijski polinom T_M, $M = 2m$ dobiven metodom najmanjih kvadrata. Vrijedi

$$T_N(x) - T_M(x) = \frac{1}{2}(a_0 - A_0) + \sum_{k=1}^{m}((a_k - A_k)\cos x_k + (b_k - B_k)\sin x_k) + \sum_{k=m+1}^{L}(a_k \cos x_k + b_k \sin x_k).$$

Dakle, u točkama interpolacije x vrijedi

$$f(x) - T_M(x) = T_N(x) - T_M(x).$$

Greška S koju minimiziramo dobiva se upotrebom relacija ortogonalnosti. Izlazi

$$\begin{aligned} S &:= \sum_{x=0}^{2L}(T_N(x) - T_M(x))^2 \\ &= \sum_{x=0}^{2L}\frac{1}{4}(a_0 - A_0)^2 + \sum_{x=0}^{2L}\sum_{k=1}^{m}((a_k - A_k)\cos x_k + (b_k - B_k)\sin x_k)^2 \\ &\quad + \sum_{x=0}^{2L}\sum_{k=m+1}^{L}(a_k \cos x_k + b_k \sin x_k)^2 \\ &= \frac{1}{4}(a_0 - A_0)^2 \cdot (2L+1) + \sum_{x=0}^{2L}\sum_{k=1}^{m}\Big[(a_k - A_k)^2\cos^2 x_k \\ &\qquad + 2(a_k - A_k)(b_k - B_k)\cos x_k \sin x_k + (b_k - B_k)^2 \sin^2 x_k\Big] \\ &\quad + \sum_{x=0}^{2L}\sum_{k=m+1}^{L}(a_k^2\cos^2 x_k + 2a_k b_k \cos x_k \sin x_k + b_k^2 \sin^2 x_k) \end{aligned}$$

$$= \frac{1}{4}(a_0 - A_0)^2 \cdot (2L+1) + \frac{2L+1}{2}\sum_{k=1}^{m}(a_k - A_k)^2 + (b_k - B_k)^2)$$
$$+ \frac{2L+1}{2}\sum_{k=m+1}^{L}(a_k^2 + b_k^2).$$

Prema tome, odmah je vidljivo da je greška S minimalna ako je

$$A_k = a_k, \quad k = 0, \ldots, m$$
$$B_k = b_k, \quad k = 1, \ldots, m,$$

i njena minimalna vrijednost jednaka je

$$S_{\min} = \frac{2L+1}{2}\sum_{k=m+1}^{L}(a_k^2 + b_k^2).$$

Ovaj oblik minimalne greške nije praktičan, jer uobičajeno ne znamo a_k, b_k za $k > m$.

Zadatak 7.9.3 *Dokažite da vrijedi*

$$S_{\min} = \sum_{x=0}^{2L}(f(x))^2 - \frac{2L+1}{4}a_0^2 - \frac{2L+1}{2}\sum_{k=1}^{m}(a_k^2 + b_k^2)$$

korištenjem relacija ortogonalnosti. Prethodni oblik greške često se koristi za detekciju stupnja trigonometrijskog polinoma, jer nagli pad greške pri dizanju stupnja trigonometrijskog polinoma znači da smo otkrili stupanj polinoma. Greška pritom ne mora biti 0*, jer je pojava mogla imati slučajne greške koje smo ionako željeli maknuti.*

Zadatak 7.9.4 *Izvedite metodu najmanjih kvadrata za tri preostala slučaja:*

1. *broj točaka paran $N = 2L - 1$, dimenzija prostora neparna $M = 2m$,*
2. *broj točaka neparan $N = 2L$, dimenzija prostora parna $M = 2m - 1$,*
3. *broj točaka paran $N = 2L - 1$, dimenzija prostora parna $M = 2m - 1$.*

Zadatak 7.9.5 *Neka je funkcija f zadana na mreži točaka $\mathcal{M} = \{0, 1, \ldots, P-1\}$, P neparan (tj. točaka je paran broj) i neka je P period funkcije f, tj.*

$$f(x + P) = f(x).$$

Pokažite da su tada

$$a_k = \frac{2}{P}\sum_{x=-L+1}^{L} f(x)\cos\frac{2\pi}{P}kx, \quad k = 0, \ldots, L$$
$$b_k = \frac{2}{P}\sum_{x=-L+1}^{L} f(x)\sin\frac{2\pi}{P}kx, \quad k = 1, \ldots, L-1.$$

Ako je f neparna funkcija $f(-x) = -f(x)$, pokažite da je

$$a_k = 0, \quad k = 0, \ldots, L$$
$$b_k = \frac{4}{P} \sum_{x=1}^{L-1} f(x) \sin \frac{2\pi}{P} kx, \quad k = 1, \ldots L-1.$$

Ako je f parna funkcija $f(-x) = f(x)$, pokažite da je

$$a_k = \frac{2}{P}(f(0) + f(L)\cos k\pi) + \frac{4}{P} \sum_{x=1}^{L-1} f(x) \cos \frac{2\pi}{P} kx, \quad k = 0, \ldots, L$$
$$b_k = 0, \quad k = 1, \ldots L-1.$$

Zadatak 7.9.6 *Riješite prethodni zadatak uz uvjet da je $P = 2L+1$, tj. da funkcija ima neparan period.*

7.10. Minimaks aproksimacija

Neka je f neprekidna funkcija na $[a, b]$. Ako uspoređujemo polinomne aproksimacije funkcije f dobivene različitim metodama, pitamo se koja je od njih najbolja, tj. koja daje najmanju maksimalnu grešku. Označimo s $\rho_n(f)$ maksimalnu grešku aproksimacije

$$\rho_n(f) = \inf_{\deg(p) \le n} \|f - p\|_\infty.$$

To znači de ne postoji polinom stupnja manjeg ili jedmakog n koji bi bolje od p aproksimirao funkciju f na danom intervalu. Nas, naravno interesira za koji se polinom dostiže ta greška

$$\rho_n(f) = \|f - p_n^*\|_\infty.$$

Ako je polinom p_n jedinstven, zanima nas kako ga možemo konstruirati. Polinom p_n^* zovemo minimaks aproksimacija funkcije f na intervalu $[a, b]$.

Pokažimo kako se najbolja aproksimacija ponaša na jednom jednostavnom primjeru.

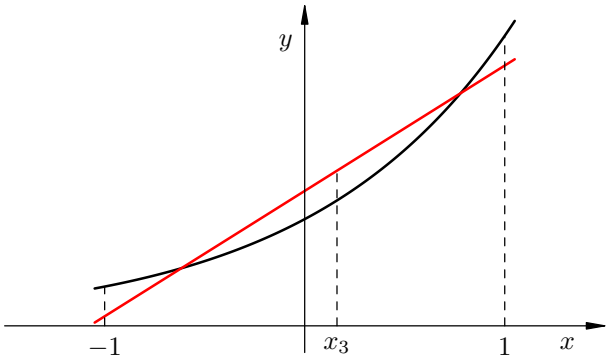
Primjer 7.10.1 *Nađimo polinom prvog stupnja $p_1^*(x) = a_0 + a_1 x$ koji je minimaks aproksimacija funkcije $f(x) = e^x$ na intervalu $[-1, 1]$, tj. da vrijedi*

$$\max_{x \in [-1,1]} |e^x - a_0 - a_1 x| \to \min.$$

Da bismo riješili problem, potrebno je malo geometrijskog zora. Nacrtajmo graf funkcije e^x i promatrajmo grešku svih polinomnih aproksimacija

$$\mathrm{err}(x) = e^x - (a_0 + a_1 x)$$

na zadanom intervalu.



Prvo, odmah je jasno da linearna minimaks aproksimacija mora sjeći graf funkcije e^x *na zadanom intervalu u točno dvije točke, nazovimo ih* x_1, x_2 *takve da je* $-1 < x_1 < x_2 < 1$. *U protivnom, ako polinom ne siječe graf niti u jednoj točki, ili ako ga siječe u točno jednoj točki, može se pokazati da postoji bolja aproksimacija. Pokažite to!*

Iz crteža odmah naslućujemo i rješenje. Nađimo jednadžbu pravca kroz točke $(-1, e^{-1})$ *i* $(1, e)$. *Zatim, nađimo koeficijent* a_0 *takav da je taj pravac tangenta funkcije* e^x *u nekoj točki* x_3. *Rješenje zadatka je pravac paralelan s prethodna dva, jednako udaljen od oba. Odmah je jasno da će postojati točno tri točke u kojima će se dostizati maksimalne pogreške: rubovi intervala i* x_3.

Pokažimo sad precizno da su naša zaključivanja ispravna. Označimo

$$\rho_1 = \max_{x\in[-1,1]} |\operatorname{err}(x)|.$$

Već smo zaključili da pogreška mora imati tri ekstrema, tj. mora vrijediti

$$\operatorname{err}(-1) = \rho_1, \quad \operatorname{err}(1) = \rho_1, \quad \operatorname{err}(x_3) = -\rho_1.$$

Budući da je $\operatorname{err}(x)$ *derivabilna, onda možemo uvjet maksimuma pogreške u* x_3 *napisati i korištenjem derivacije, tj.* $\operatorname{err}'(x_3) = 0$.

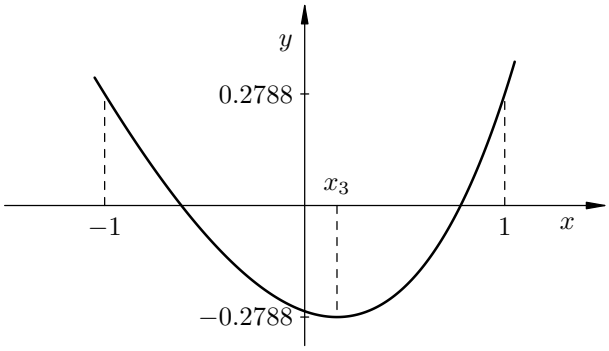
Sad možemo skupiti sve četiri jednadžbe koje trebamo zadovoljiti

$$\begin{aligned} e^{-1} - a_0 + a_1 &= \rho_1 & \qquad e^{x_3} - a_0 - a_1 x_3 &= -\rho_1 \\ e - a_0 - a_1 &= \rho_1 & \qquad e^{x_3} - a_1 &= 0. \end{aligned}$$

Rješenje te četiri jednadžbe je

$$\begin{aligned} a_1 &= \frac{e - e^{-1}}{2} \approx 1.1752 \\ x_3 &= \ln a_1 \approx 0.1614 \\ \rho_1 &= \frac{1}{2}e^{-1} + \frac{x_3}{4}(e - e^{-1}) \approx 0.2788 \\ a_0 &= \rho_1 + (1 - x_3)a_1 \approx 1.2643. \end{aligned}$$

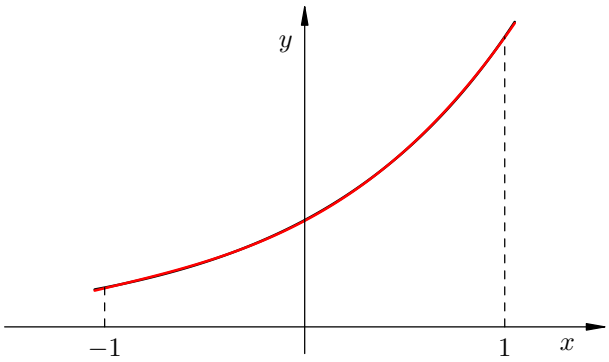
Graf pogreške ima karakterističan oscilirajući izgled.



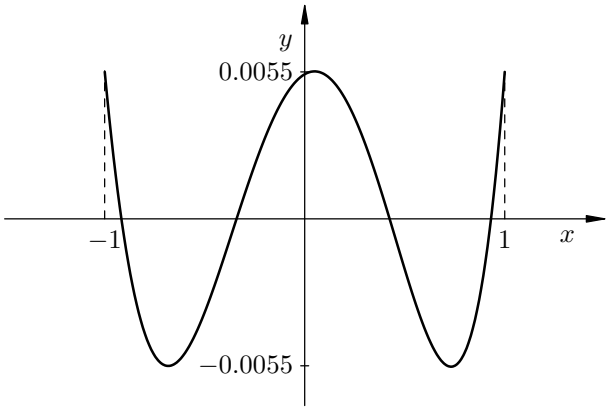
Korištenjem tzv. Remesovog algoritma možemo konstruirati i kubični polinom koji najbolje aproksimira istu funkciju. Taj polinom je

$$p_3^*(x) = 0.994579 + 0.995668x + 0.542973x^2 + 0.179533x^3.$$

Ako nacrtamo graf tog polinoma, on se na slici neće razlikovati od funkcije,



jer će greška biti iznimno mala, reda veličine 0.0055 *i opet karakteristično oscilirajuća.*



Za dobru uniformnu aproksimaciju zadane funkcije f, realno je očekivati da je greška jednako tako uniformno distribuirana na intervalu aproksimacije i da varira po predznaku. Iznijet ćemo dva vrlo važna teorema, od kojih prvi daje egzistenciju minimaks aproksimacije i važno svojstvo o oscilaciji grešaka. Drugi ocjenjuje pogrešku minimaks aproksimacije ρ_n polinoma stupnja n, bez da se sam polinom izračuna.

Teorem 7.10.1 (Čebiševljev teorem o oscilacijama grešaka) *Za danu funkciju f, $f \in C[a,b]$ i za dani $n \geq 0$ postoji jedinstven polinom p_n^* stupnja manjeg ili jednakog n za koji je*

$$\rho_n(f) = \|f - p_n^*\|_\infty.$$

Taj polinom je karakteriziran sljedećim svojstvom: postoje barem $n+2$ točke

$$a \leq x_0 < x_1 < \cdots < x_n < x_{n+1} \leq b$$

za koje je

$$f(x_j) - p_n^*(x_j) = \sigma(-1)^j \rho_n(f), \quad j = 0, \ldots, n+1,$$

pri čemu je $\sigma = \pm 1$ i ovisi samo o f i n.

Dokaz prethodnog teorema je tehnički, vrlo dugačak i provodi se obratom po kontrapoziciji.

Teorem 7.10.2 (de la Vallee–Poussin) *Neka je $f \in C[a,b]$ i $n \geq 0$. Pretpostavimo da polinom P stupnja manjeg ili jednakog n zadovoljava*

$$f(x_j) - P(x_j) = (-1)^j e_j, \quad j = 0, 1, \ldots, n+1$$

a e_j su različiti od 0 i istog predznaka, za x_j vrijedi

$$a \leq x_0 < x_1 < \cdots < x_n < x_{n+1} \leq b.$$

Tada je

$$\min_{0 \leq j \leq n+1} |e_j| \leq \rho_n(f) = \|f - p_n^*(x)\|_\infty \leq \|f - P\|_\infty.$$

Dokaz. Posljednja nejednakost (gornja ograda) u prethodnoj formuli posljedica je definicije minimaks aproksimacije, tj. da polinom p_n^* ima maksimum pogreške manji ili jednak od svih ostalih polinoma P.

Donja ograda za $\rho_n(f)$ dokazuje se pretostavljanjem suprotnog. Pretpostavimo da je

$$\rho_n(f) < \min_{0 \leq j \leq n+1} |e_j|.$$

Budući da je $\rho_n(f)$ infimum (zaboravimo načas da smo dokazali i minimum), onda sigurno postoji bar jedan polinom Q stupnja manjeg ili jedankog n koji se nalazi između $\rho_n(f)$ i minimuma $|e_j|$, tj. vrijedi

$$\rho_n(f) \leq \|f - Q\|_\infty < \min_{0 \leq j \leq n+1} |e_j|.$$

Primijetite da polinomi P i Q nisu jednaki! Definiramo

$$R(x) = P(x) - Q(x).$$

R je polinom stupnja manjeg ili jedankog n. Zbog jednostavnosti, pretpostavimo da su svi $e_j > 0$ (isti argument radit će i ako su svi manji od 0). Izračunajmo vrijednosti polinoma R u točkama x_j. Počnimo s x_0 i promotrimo predznak rezultata

$$R(x_0) = P(x_0) - Q(x_0) = (f(x_0) - Q(x_0)) - (f(x_0) - P(x_0)) = (f(x_0) - Q(x_0)) - e_0.$$

Budući da je

$$|f(x_0) - Q(x_0)| < \min |e_j| = \min e_j \leq e_0,$$

onda je

$$R(x_0) = f(x_0) - Q(x_0)) - e_0 < 0.$$

Nadalje je

$$R(x_1) = P(x_1) - Q(x_1) = (f(x_1) - Q(x_1)) - (f(x_1) - P(x_1)) = (f(x_1) - Q(x_1)) + e_1.$$

Ponovno, zbog

$$|f(x_1) - Q(x_1)| < \min |e_j| = \min e_j \leq e_1,$$

slijedi da je

$$R(x_1) = (f(x_1) - Q(x_1)) + e_1 > 0.$$

Induktivno, dobivamo da je

$$\operatorname{sign}(R(x_j)) = (-1)^{j+1}, \quad j = 0, \ldots, n+1,$$

tj. R oscilira tako da ima bar $n+2$ različita predznaka, tj. da predznak promijeni bar $n+1$ puta. Ali R je polinom stupnja n, pa predznak može mijenjati samo u nultočkama. Po osnovnom teoremu algebre znamo da polinom R stupnja $n \geq 1$ ima točno n nultočaka, a ne bar $(n+1)$-nu. Jedina mogućnost koja ostaje je da je R baš nul-polinom, tj. da je $P = Q$, što je suprotno pretpostavci. ■

Predodžbu o tome kako se ponaša najbolja aproksimacija s porastom stupnja n daje sljedeći teorem.

Teorem 7.10.3 (Jackson) *Neka funkcija f ima k neprekidnih derivacija za neki $k \geq 0$. Čak štoviše pretpostavimo da $f^{(k)}$ zadovoljava*

$$\sup_{a \leq x, y \leq b} |f^{(k)}(x) - f^{(k)}(y)| \leq M|x - y|^\alpha$$

za neki $M > 0$ i neki $0 < \alpha \leq 1$, tj. kaže se da f zadovoljava Hölderov uvjet s eksponentom α. Tada postoji konstanta d_k nezavisna o f i n za koju je

$$\rho_n(f) \leq \frac{Md_k}{n^{k+\alpha}}, \quad n \geq 1.$$

Ako u prethodnom teoremu želimo izbjeći Hölderov uvjet, dovoljno je pretpostaviti da f im k neprekidnih derivacija. Umjesto k-te derivacije svugdje dalje u teoremu koristimo $(k-1)$-u, stavljamo $\alpha = 1$ i

$$M = \|f^{(k)}\|_\infty.$$

Tada se Hölderov uvjet svede na obični teorem srednje vrijednosti, a kao rezultat dobivamo

$$\rho_n(f) \leq \frac{d_{k-1}}{n^k} \|f^{(k)}\|_\infty.$$

Nadalje, ako je f beskonačno puta derivabilna, tada p_n^* konvergira prema f uniformno na $[a, b]$ brže nego bilo koja potencija $1/n^k$, $k \geq 1$.

7.10.1. Remesov algoritam

Traženje minimaks aproksimacije p_n^* za f može se pronaći iterativnim algoritmom poznatijim kao drugi Remesov algoritam. Ovdje treba napomenuti da se taj algoritam može generalizirati i na racionalne funkcije i na slučaj kad funkcija f nije zadana na intervalu, nego na skupu točaka (tada se on zove diferencijalni algoritam korekcije).

Remesov algoritam koristi svojstvo oscilacije greške koje mora imati minimaks aproksimacija. Iteracije imaju tri dijela.

Prvi korak

Zadane su $n+2$ točke

$$a \leq x_0^{(0)} < x_1^{(0)} < \cdots < x_n^{(0)} < x_{n+1}^{(0)} \leq b.$$

koje određuju polinom p stupnja $\deg(p) \leq n$ iz uvjeta

$$f(x_i^{(0)}) - p(x_i^{(0)}) = (-1)^i E, \quad i = 0, \ldots, n+1,$$

pri čemu zahtjevamo da pogreška E (koju još ne znamo) oscilira s jednakim amplitudama. Prehodna ralacija vodi na linearni sustav s $n+2$ nepoznanice od kojih su $n+1$ koeficijenti polinoma p, a posljednja je E.

Drugi korak

Riješimo linearni sustav, tj. odredimo redom $a_0^{(0)}, \ldots, a_n^{(0)}$ (koeficijente polinoma p) i nađimo pirpadni E, zovimo ga E_0.

Treći korak

Tražimo novih $n+2$ točaka. Definiramo funkciju

$$h_0(x) = f(x) - \sum_{i=0}^{n} a_i^{(0)} x^i.$$

Funkcija h_0 ima u točkama $x_i^{(0)}$ vrijednosti $\pm E_0$ (to su greške) koje alterniraju po predznaku. Zbog toga, nije teško pokazati da u okolini svake točke $x_i^{(0)}$ postoji točka $x_i^{(1)}$ takva da u njoj $h_0(x)$ ima ekstrem i to istog predznaka kao što je predznak $f(x_i^{(0)}) - p(x_i^{(0)})$. Nakon toga zamjenjujemo $x_i^{(0)}$ s $x_i^{(1)}$. Naravno, rubne točke $x_0^{(1)}$ i $x_{n+1}^{(1)}$ moraju ostati u $[a, b]$.

Ove nove točke $x_i^{(1)}$ "lokalnih" ekstrema funkcije h_0 moraju sadržavati i točku u kojoj $|h_0|$ dostiže globalni maksimum na $[a, b]$. Naime, ako je $\bar{x}$ točka u kojoj $|h_0|$ poprima globalnu maksimalnu vrijednost na $[a, b]$

$$\|h_0\|_\infty = \|f - p\|_\infty = |f(\bar{x}) - p(\bar{x})|,$$

i ona nije među točakama $x_i^{(1)}$, onda treba onda treba zamijeniti jednu od točaka $x_i^{(1)}$ točkom $\bar{x}$, tako da h_0 i na tom novom skupu točaka alternira po znaku. Može se pokazati da se to uvijek može napraviti.

Primjernom teorema 7.10.2 dobivamo da je

$$m := \min_{i=0,\ldots,n+1} |f(x_i^{(1)}) - p(x_i^{(1)})| \leq \rho_n(f) \leq M := \max_{i=0,\ldots,n+1} |f(x_i^{(1)}) - p(x_i^{(1)})|.$$

Ako je omjer M/m dovoljno blizu 1, smatramo da je nađeni p dovoljno blizu polinomne minimaks aproksimacije za f.

U protivnom, treba ponoviti drugi, pa treći korak, samo na novim točkama $x_i^{(1)}$. Ovaj proces generirat će niz polinomnih aproksimacija će uniformno konvergirati prema minimaks polinomnoj aproksimaciji.

7.11. Skoro minimaks aproksimacije

Vidjeli smo da minimaks aproksimaciju nije jako lako izračunati. Zbog toga, bili bismo zadovoljni i približnom minimaks aproksimacijom. Ako se prisjetimo Čebiševljevog teorema o oscilaciji greške, možemo doći do metode koja daje dobru približnu minimaks aproksimaciju.

Vidjet ćemo da su u tu svrhu vrlo pogodni Čebiševljevi polinomi, koji zadovoljavaju sljedeću relaciju ortogonalnosti

$$\int_{-1}^{1} \frac{T_m(x)\,T_n(x)}{\sqrt{1-x^2}}\,dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi, & \text{za } m = n = 0, \\ \pi/2, & \text{za } m = n \neq 0. \end{cases}$$

Ako želimo funkciju f razviti po Čebiševljevim polinomima, potrebno je samo supstituirati podatke u opći algoritam. Želimo odrediti koeficijente c_k u razvoju funkcije f po Čebiševljevim polinomima. Ako je razvoj napišemo u obliku

$$f(x) = \frac{c_0}{2}T_0(x) + \sum_{i=1}^{\infty} c_i T_i(x), \tag{7.11.1}$$

onda ćemo, formalno gledajući, koeficijent c_j dobiti ako pomnožimo prethodnu relaciju s $T_j(x)$, zatim težinskom funkcijom w i integriramo od -1 do 1. Dobivamo formulu

$$c_j = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_j(x)}{\sqrt{1-x^2}}\,dx.$$

Označimo s f_n početni komad razvoja, tj. neka je

$$f_n(x) = \frac{c_0}{2}T_0(x) + \sum_{i=1}^{n} c_i T_i(x).$$

Ako je $f \in C[-1,1]$ tada razvoj (7.11.1) konvergira, u smislu da je

$$\lim_{n\to\infty} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} \Big(f(x) - f_n(x)\Big)^2\,dx = 0.$$

Za uniformnu konvergenciju, možemo pokazati dosta jak rezultat

$$\rho_n(f) \leq \|f - f_n\|_\infty \leq \Big(4 + \frac{4}{\pi^2}\ln n\Big)\rho_n(f).$$

Kako se ponaša greška odbacivanja reda? Ako se prisjetimo da je

$$T_n(x) = \cos n\vartheta, \quad x = \cos\vartheta,$$

onda se može pokazati da vrijedi

$$f(x) - f_n(x) = \sum_{i=n+1}^{\infty} c_i T_i(x) \approx c_{n+1}T_{n+1}(x) = c_{n+1}\cos(n+1)\vartheta,$$

ako je $c_{n+1} \neq 0$ i ako koeficijenti c_i brzo konvergiraju k 0. Iz definicije T_{n+1} izlazi

$$|T_{n+1}(x)| = |\cos(n+1)\vartheta| \leq 1, \quad -1 \leq x \leq 1.$$

Nultočke i ekstreme polinoma T_{n+1} nije teško izračunati. Nultočke pripadnog kosinusa su na odgovarajućem intervalu su

$$(n+1)\vartheta_j = \frac{(2j+1)\pi}{2}, \quad j = 0, \ldots, n,$$

pa su nultočke T_{n+1} jednake

$$x_j = \cos\Big(\frac{(2j+1)\pi}{2(n+1)}\Big), \quad j = 0, \ldots, n.$$

S druge strane, lokalni ekstremi se postižu kad je

$$(n+1)\vartheta_k = k\pi, \quad k = 0, \ldots, n+1,$$

pa su ekstremi T_{n+1} jednaki

$$x_k = \cos\Big(\frac{k\pi}{(n+1)}\Big), \quad k = 0, \ldots, n+1.$$

Drugim riječima, vrijedi

$$T_{n+1}(x_k) = (-1)^k, \quad k = 0, \ldots, n+1.$$

Primijetite da tih ekstrema ima točno $n+2$ i da alterniraju po znaku. Ako to iskoristimo za funkciju $c_{n+1}T_{n+1}$, onda je jasno da ona ima $n+2$ lokalna ekstrema jednakih amplituda. Po Čebiševljevom teoremu o oscilaciji grešaka, odatle odmah izlazi da je f_n skoro minimaks aproksimacija za f (ovo skoro minimaks potječe od toga što je greška odbacivanja članova reda približno jednaka $c_{n+1}T_{n+1}$).

Postoji još jedan razlog zašto se koristi razvoj po Čebiševljevim polinomima. Vrijedi sljedeći teorem.

Teorem 7.11.1 *Za fiksni prirodni broj n, promatrajmo minimizacijski problem*

$$\tau_n = \inf_{\deg(P)\leq n-1}\big(\max_{-1\leq x\leq 1}|x^n + P(x)|\big),$$

gdje je P polinom. Minimum τ_n se dostiže samo za

$$x^n + P(x) = \frac{1}{2^{n-1}}T_n(x).$$

Pripadna pogreška je

$$\tau_n = \frac{1}{2^{n-1}}.$$

Dokaz. Iz tročlane rekurzije, nije teško induktivno dokazati da je vodeći koeficijent T_n jednak

$$T_n(x) = 2^{n-1}x^n + \text{članovi nižeg stupnja}, \quad n \geq 1.$$

Zbog toga vrijedi da je

$$\frac{1}{2^{n-1}}T_n(x) = x^n + \text{članovi nižeg stupnja}.$$

Budući da su točke

$$x_k = \cos\left(\frac{k\pi}{n}\right), \quad j = 0, \ldots, n,$$

lokalni ekstremi od T_n, u kojima je

$$T_n(x_k) = (-1)^k, \quad k = 0, \ldots, n$$

i

$$-1 = x_n < x_{n-1} < \cdots < x_1 < x_0 = 1.$$

Polinom

$$\frac{1}{2^{n-1}}T_n$$

ima vodeći koeficijent 1 i vrijedi

$$\max_{-1\leq x\leq 1}\left|\frac{1}{2^{n-1}}T_n\right| = \frac{1}{2^{n-1}}.$$

Zbog toga je

$$\tau_n \leq \frac{1}{2^{n-1}}.$$

Pokažimo da je τ_n baš jednak desnoj strani. Pretpostavimo suprotno, tj. da je

$$\tau_n < \frac{1}{2^{n-1}}.$$

Pokazat ćemo da to vodi na kontradikciju. Definicija τ_n i prethodna pretpostavka pokazuju da postoji polinom M takav da je

$$M(x) = x^n + P(x), \quad \deg(P) \leq n-1,$$

gdje je

$$\tau_n \leq \max_{-1\leq x\leq 1}|M(x)| < \frac{1}{2^{n-1}}. \tag{7.11.2}$$

Definiramo

$$R(x) = \frac{1}{2^{n-1}}T_n(x) - M(x).$$

Tvrdimo da će se vodeći koeficijenti funkcija s desne strane skratiti, pa je $\deg(R) \leq n-1$. Ispitajmo vrijednosti funkcije R u lokalnim ekstremima funkcije T_n. Iz (7.11.2) redom, izlazi

$$R(x_0) = R(1) = \frac{1}{2^{n-1}} - M(1) > 0$$

$$R(x_1) = -\frac{1}{2^{n-1}} - M(x_1) < 0, \ldots$$

Tj. za polinom R vrijedi

$$\operatorname{sign}(R(x_k)) = (-1)^k.$$

Budući da ima bar $n+1$ različiti predznak, to mora postojati bar n nultočaka, što je moguće damo ako je $R = 0$. Odatle odmah izlazi da je

$$M(x) = \frac{1}{2^{n-1}} T_n(x).$$

Sad bi još trebalo pokazati da je to jedini polinom s takvim svojstvom. Taj dio dokaza vrlo je sličan ovom što je već dokazano. ■

7.12. Interpolacija u Čebiševljevim točkama

Ako se prisjetimo problema interpolacije, onda znamo da je greška interpolacionog polinoma stupnja n jednaka

$$f(x) - p_n(x) = \frac{(x - x_0) \cdots (x - x_n)}{(n+1)!} f^{(n+1)}(\xi).$$

Vrijednost $(n+1)$-ve derivacije ovisi o točkama interpolacije, ali nije jednostavno reći kako. Ipak, ono što možemo kontrolirati je izbor točaka interpolacije. Pretpostavimo da interpoliramo funkciju na intervalu $[-1, 1]$. Ako naš interval nije $[-1, 1]$, nego $[a, b]$, onda ga linearnom transformacijom

$$y = cx + d$$

možemo svesti na zadani interval. Dakle izaberimo točke interpolacije $x_j \in [-1, 1]$ tako da minimiziraju

$$\max_{-1 \leq x \leq 1} |(x - x_0) \cdots (x - x_n)|.$$

Polinom u prethodnoj relaciji je stupnja $n+1$ i ima vodeći koeficijent 1. Po Teoremu 7.11.1, minimum ćemo dobiti ako stavimo

$$(x - x_0) \cdots (x - x_n) = \frac{1}{2^n} T_{n+1}(x),$$

a minimalna će vrijednost biti $1/2^n$. Odatle odmah čitamo da su čvorovi $x_0, \ldots x_n$ nultočke polinoma T_{n+1}, a njih smo već izračunali da su jednake

$$x_j = \cos\left(\frac{(2j+1)\pi}{2n+2}\right), \quad j = 0, \ldots, n.$$

7.13. Čebiševljeva ekonomizacija

Čebiševljevi polinomi mogu se koristiti za smanjivanje stupnja interpolacionog polinoma, uz minimalni gubitak točnosti. Takav postupak zove se ekonomizacija.

Pretpostavimo da je zadan proizvoljni polinom stupnja n

$$p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0.$$

na intervalu $[-1, 1]$. Taj polinom želimo zamijeniti polinomom stupnja za jedan manjeg tako da je greška koja je pritom nastala minimalna moguća

$$\max_{x\in[-1,1]} |p_n(x) - p_{n-1}(x)| \to \min.$$

Rješenje problema se neće promijeniti ako normiramo vodeći koeficijent na 1, tj. ako tražimo

$$\max_{x\in[-1,1]} \left|\frac{1}{a_n}(p_n(x) - p_{n-1}(x))\right| \to \min.$$

Prema Teoremu 7.11.1 o minimalnom otklanjanju od polinoma x^n, izlazi da mora biti

$$\max_{x\in[-1,1]} \left|\frac{1}{a_n}(p_n(x) - p_{n-1}(x))\right| \geq \frac{1}{2^{n-1}},$$

s tim da jednakost vrijedi kad je

$$\frac{1}{a_n}(p_n(x) - p_{n-1}(x)) = \frac{1}{2^{n-1}}T_n(x).$$

Drugim riječima, izbor $p_{n-1}(x)$ je

$$p_{n-1}(x) = p_n(x) - \frac{a_n}{2^{n-1}}T_n(x),$$

a s tim izborom je maksimalna greška jednaka

$$\max_{x\in[-1,1]} |p_n(x) - p_{n-1}(x)| = |a_n| \max_{x\in[-1,1]} \left|\frac{1}{a_n}(p_n(x) - p_{n-1}(x))\right| = \frac{|a_n|}{2^{n-1}}.$$

Primjer 7.13.1 *Funkciju $f(x) = e^x$ aproksimiramo na intervalu $[-1, 1]$ Taylorovim polinomom oko 0 stupnja četiri*

$$p_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}.$$

Greška odbacivanja tog polinoma je

$$|R_4(x)| \leq \frac{M_5}{5!}|x^5|, \quad M_5 = \max_{x\in[-1,1]} |f^{(5)}(x)|.$$

Odmah se vidi da je $M_5 = e$*, pa je greška odbacivanja*

$$|R_4(x)| \leq \frac{e}{120}|x^5| \leq \frac{e}{120} \approx 0.023,$$

za $-1 \leq x \leq 1$*. Pretpostavimo da možemo tolerirati grešku* 0.05*, pa pokušajmo spustiti stupanj polinoma. Nije teško naći* T_4*, recimo iz rekurzije*

$$T_4(x) = 8x^4 - 8x^2 + 1.$$

Pokazali smo da mora biti

$$\begin{aligned} p_3(x) &= p_4(x) - \frac{a_4}{2^{4-1}}T_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} - \frac{1}{8 \cdot 24}(8x^4 - 8x^2 + 1) \\ &= \frac{191}{192} + x + \frac{13}{24}x^2 + \frac{1}{6}x^3. \end{aligned}$$

Tim snižavanjem stupnja, napravljena je greška

$$|p_4(x) - p_3(x)| \leq \frac{1}{24 \cdot 2^3} = \frac{1}{192} \leq 0.0053.$$

Pribrojimo li tu grešku grešci odbacivanja, onda je ukupna greška manja ili jednaka $0.023 + 0.0053 = 0.0283$*, što je prema uvjetima zadatka dozvoljiva greška.*

Naravno, stupanj polinoma p_3 *možemo pokušati još smanjiti. Budući da je*

$$T_3(x) = 4x^3 - 3x,$$

dobivamo

$$\begin{aligned} p_2(x) &= p_3(x) - \frac{a_3}{2^{3-1}}T_3(x) = \frac{191}{192} + x + \frac{13}{24}x^2 + \frac{1}{6}x^3 - \frac{1}{4 \cdot 6}(4x^3 - 3x) \\ &= \frac{191}{192} + \frac{9}{8}x + \frac{13}{24}x^2. \end{aligned}$$

Pritom je napravljena greška

$$|p_3(x) - p_2(x)| = \frac{1}{6 \cdot 4} = \frac{1}{24} \approx 0.042.$$

Dodamo li tu grešku na već akumuliranu grešku 0.0283*, onda je ukupna greška* $0.0283 + 0.042 > 0.05$*, što je bila dozvoljena tolerancija.*

Postoji još jedan način ekonomizacije. Znamo da se funkcije mogu razviti u red po Čebiševljevim polinomima oblika

$$f(x) = \frac{1}{2}c_0 + \sum_{k=1}^{\infty} c_k T_k(x), \quad |x| \leq 1,$$

a koeficijenti u razvoju su integrali koji u sebi sadrže f, Čebiševljev polinom i težinsku funkciju za Čebiševljeve polinome. Koeficijente c_k u ovom razvoju možemo, i to relativno brzo, numerički izračunati, koristeći algoritme na bazi diskretne ortogonalnosti Čebiševljevih polinoma. Taj postupak opisujemo u sljedećem odjeljku.

S druge strane, ako znamo (ili lako računamo) koeficijente a_m u redu potencija

$$f(x) = \sum_{m=0}^{\infty} a_m x^m,$$

onda potencije x^m možemo razviti po Čebiševljevim polinomima, pa nakon toga primijeniti postupak ekonomizacije (odbacivanjem članova) da dobijemo ravnomjernije ponašanje pogreške.

Može se pokazati da vrijedi

$$(2x)^n = 2 \sum_{k=1}^{[n/2]} \epsilon_k \binom{n}{k} T_{n-2k}(x),$$

pri čemu je

$$\epsilon_k = \begin{cases} 1/2 & \text{za } k = n/2, \\ 1 & \text{inače.} \end{cases}$$

Taj razvoj napisan posebno za parne, a posebno za neparne potencije je

$$(2x)^{2m} = \binom{2m}{m} + 2 \sum_{k=1}^{m} \binom{2m}{m-k} T_{2k}(x)$$
$$(2x)^{2m+1} = 2 \sum_{k=0}^{m} \binom{2m+1}{m-k} T_{2k+1}(x).$$

Ako za polinomnu aproksimaciju uzmemo polinom stupnja n, uvrštavanjem razvoja po Čebiševljevim polinomima dobivamo

$$f_n(x) = \sum_{m=0}^{n} a_m x^m = \frac{b_0}{2} + \sum_{m=1}^{n} b_m T_m(x),$$

gdje su koeficijenti

$$b_{2k} = 2 \sum_{i=k}^{2i \le n} \binom{2i}{i-k} \frac{a_{2i}}{2^{2i}}, \quad b_{2k+1} = 2 \sum_{i=k}^{2i+1 \le n} \binom{2i+1}{i-k} \frac{a_{2i+1}}{2^{2i}}.$$

Vrlo se često događa da su jedan ili nekoliko posljednjih koeficijenata b_m maleni, pa odbacivanje posljednjih članova bitno ne smanjuje točnost aproksimacije. Drugim riječima, smanjili smo stupanj aproksimacije. Takvo smanjivanje stupnja zove se i relaksacija stupnja aproksimacije.

7.14. Diskretne ortogonalnosti polinoma T_n

Budući da su Čebiševljevi polinomi kosinusi, onda oni zadovoljavaju diskretne relacije ortogonalnosti vrlo slične onima koje zadovoljavaju trigonometrijske funkcije.

Neka su x_α nultočke Čebiševljevog polinoma T_n, tj. neka je

$$T_n(x_\alpha) = \cos(n\vartheta_\alpha) = 0.$$

Nije teško izračunati da je tada

$$x_\alpha = \cos(\vartheta_\alpha), \quad \vartheta_\alpha = \frac{(2\alpha+1)\pi}{2n}, \quad \alpha = 0, \ldots, n-1.$$

Za Čebiševljeve polinome, u nultočkama vrijede sljedeće relacije ortogonalnosti

$$U_{j,k} = \sum_{\alpha=0}^{n-1} T_j(x_\alpha)T_k(x_\alpha) = \sum_{\alpha=0}^{n-1} \cos j\vartheta_\alpha \cos k\vartheta_\alpha,$$

gdje je

$$U_{j,k} = \begin{cases} 0 & j,k < n,\ j \neq k, \\ n/2 & j = k,\ 0 < j < n, \\ n & j = k = 0. \end{cases}$$

Sada možemo funkciju razviti po Čebiševljevim polinomima koristeći prethodnu relaciju diskretne ortogonalnosti. Može se pokazati da vrijedi sljedeći teorem.

Teorem 7.14.1 *Neka je $f_n(x)$ aproksimacija za $f(x)$,*

$$f_n(\cos\vartheta) = \frac{d_0}{2} + \sum_{k=1}^{n-1} d_k \cos k\vartheta,$$

ili

$$f_n(x) = \frac{d_0}{2} + \sum_{k=1}^{n-1} d_k T_k(x). \tag{7.14.1}$$

Tada je

$$d_k = \frac{2}{n}\sum_{\alpha=0}^{n-1} f(\cos\vartheta_\alpha)\cos k\vartheta_\alpha = \frac{2}{n}\sum_{\alpha=0}^{n-1} f(x_\alpha)T_k(x_\alpha).$$

Pretpostavimo da je f' neprekidna na $[-1,1]$, osim najviše u konačno mnogo točaka, gdje ima ograničene skokove. Tada se f može razviti u konvergentan red oblika

$$f(x) = \frac{c_0}{2} + \sum_{k=1}^{\infty} c_k T_k(x), \tag{7.14.2}$$

gdje je

$$c_k = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}}\,dx = \frac{2}{\pi}\int_0^\pi f(\cos\vartheta)\cos k\vartheta\,d\vartheta.$$

Osim toga, postoji veza između koeficijenata u diskretnom i kontinuiranom razvoju:

$$d_0 = c_0 + 2\sum_{r=1}^{\infty}(-1)^r c_{2rn}$$
$$d_k = c_k + \sum_{r=1}^{\infty}(-1)^r c_{2rn-k} + \sum_{r=1}^{\infty}(-1)^r c_{2rn+k}, \quad k = 1, \ldots, n-1.$$

Sljedeći teorem govori o greškama koje smo napravili aproksimacijom f_n obzirom na f.

Teorem 7.14.2 *Neka je*

$$\epsilon_n(x) = f(x) - f_n(x),$$

pri čemu su f_n i f zadani s (7.14.1) i (7.14.2). Za grešku ϵ_n tada vrijedi

$$\begin{aligned}\epsilon_n(\cos\vartheta) = \cos n\vartheta\Big(c_n + 2\sum_{r=1}^{2n-1} c_{n+r}\cos r\vartheta + c_{3n}\cos 2n\vartheta\Big)\\ - \sin 2n\vartheta\Big(c_{3n}\sin n\vartheta + 2\sum_{r=1}^{2n-1} c_{3n+r}\sin(n+r)\vartheta + c_{5n}\sin 3n\vartheta\Big)\\ + \cos 3n\vartheta\Big(c_{5n}\cos 2n\vartheta + 2\sum_{r=1}^{2n-1} c_{5n+r}\cos(2n+r)\vartheta + c_{7n}\cos 4n\vartheta\Big) - \cdots,\end{aligned}$$

odnosno, približno

$$\epsilon_n(\cos\vartheta) \approx c_n\cos n\vartheta\Big(1 + \frac{2c_{n+1}}{c_n}\cos\vartheta\Big).$$

Posebno, vrijedi

$$\epsilon_n(\cos\vartheta_\alpha) = 0.$$

Iz prethodnih teorema uočavamo da x_α leže u unutrašnjosti intervala. U mnogim je primjenama je korisno dozvoliti aproksimaciju i u rubnim točkama ± 1 i točkama koje leže u sredini među ϑ_α. Primijetite da su to ekstremi odgovarajućeg Čebiševljevog polinoma. Sada možemo napraviti sličan niz tvrdnji kao za diskretnu ortogonalnost u nultočkama.

Neka su x_α ekstremi Čebiševljevog polinoma T_n, tj. neka je

$$x_\alpha = \cos(\psi_\alpha), \quad \psi_\alpha = \frac{\alpha\pi}{n}, \quad \alpha = 0, \ldots, n.$$

Za Čebiševljeve polinome, u ekstremima vrijede sljedeće relacije ortogonalnosti

$$\begin{aligned}V_{j,k} &= \frac{1}{2}(T_j(x_0)T_k(x_0) + T_j(x_n)T_k(x_n)) + \sum_{\alpha=1}^{n-1} T_j(x_\alpha)T_k(x_\alpha)\\ &= \frac{1}{2}(\cos j\psi_0\cos k\psi_0 + \cos j\psi_n\cos k\psi_n) + \sum_{\alpha=1}^{n-1}\cos j\psi_\alpha\cos k\psi_\alpha,\end{aligned}$$

gdje je

$$V_{j,k} = \begin{cases} 0 & j,k < n,\ j \neq k, \\ n/2 & j = k,\ 0 < j < n, \\ n & j = k = 0 \text{ ili } j = k = n. \end{cases}$$

Sada možemo funkciju razviti po Čebiševljevim polinomima koristeći prethodnu relaciju diskretne ortogonalnosti. Može se pokazati da vrijedi sljedeći teorem.

Teorem 7.14.3 *Neka je $f_n(x)$ aproksimacija za $f(x)$,*

$$f_n(\cos\psi) = \frac{e_0}{2} + \sum_{k=1}^{n-1} e_k \cos k\psi + \frac{e_n}{2}\cos n\psi,$$

ili

$$f_n(x) = \frac{e_0}{2} + \sum_{k=1}^{n-1} e_k T_k(x) + \frac{e_n}{2} T_n(x). \tag{7.14.3}$$

Tada je

$$\begin{aligned} e_k &= \frac{2}{n}\Big(\frac{f(1)+(-1)^k f(-1)}{2} + \sum_{\alpha=1}^{n-1} f(\cos\psi_\alpha)\cos k\psi_\alpha\Big) \\ &= \frac{2}{n}\Big(\frac{f(1)+(-1)^k f(-1)}{2} + \sum_{\alpha=1}^{n-1} f(x_\alpha) T_k(x_\alpha)\Big). \end{aligned}$$

Osim toga, postoji veza između koeficijenata u diskretnom i kontinuiranom razvoju:

$$\begin{aligned} e_0 &= c_0 + 2\sum_{r=1}^{\infty} c_{2rn} \\ e_n &= 2c_n + 2\sum_{r=1}^{\infty} c_{(2r+1)n} \\ e_k &= c_k + \sum_{r=1}^{\infty} c_{2rn-k} + \sum_{r=1}^{\infty} c_{2rn+k}, \quad k = 1,\dots,n-1. \end{aligned}$$

Sljedeći teorem govori o greškama koje smo napravili aproksimacijom f_n obzirom na f.

Teorem 7.14.4 *Neka je*

$$\delta_n(x) = f(x) - f_n(x),$$

pri čemu su f_n i f zadani s (7.14.3) i (7.14.2). Za grešku δ_n tada vrijedi

$$\delta_n(\cos\psi) = -2\sin n\psi \sum_{r=1}^{\infty} c_{n+r}\sin r\psi$$

odnosno, približno

$$\delta_n(\cos\psi) \approx -2\sin n\psi \sin\psi c_{n+1}\Big(1 + \frac{2c_{n+2}}{c_{n+1}}\cos\psi\Big).$$

Posebno, vrijedi

$$\delta_n(\cos\psi_\alpha) = 0.$$

Ako se c_k iz razvoja f integrira po trapeznoj formuli (vidjeti kasnije), onda se takvom aproksimacijom dobivaju koeficijenti e_k. I d_k su koeficijenti koji se dobivaju približnom integracijom c_k (modificiranom trapeznom formulom, odnosno, tzv. "midpoint" pravilom).

7.15. Thieleova racionalna interpolacija

Racionalne funkcije bolje aproksimiraju funkcije koje imaju singularitete, nego što to mogu polinomi. Jasno je da polinomi ne mogu dobro aproksimirati funkciju u okolini točke prekida, jer ih oni sami nemaju.

Prvo, definirajmo recipročne razlike, a zatim verižni razlomak koji će interpolirati funkciju f u točkama $x_1, \ldots, x_n$ (ovdje su indeksi od 1, a ne od 0).

Recipročne razlika nultog i prvog reda definiraju se redom kao

$$\rho_0(x_0) = f(x_0), \quad \rho_1(x_0, x_1) = \frac{x_0 - x_1}{f(x_0) - f(x_1)},$$

a one viših redova rekurzivno kao

$$\rho_k(x_0, \ldots, x_k) = \frac{x_0 - x_k}{\rho_{k-1}(x_0, \ldots, x_{k-1}) - \rho_{k-1}(x_1, \ldots, x_k)} + \rho_{k-2}(x_1, \ldots, x_{k-1}), \quad k \geq 2.$$

Za računanje recipročnih razlika obično se koristi tablica vrlo slična onoj za podijeljene razlike. Kao što ćemo to pokazati kasnije, algoritam koji će koristiti recipročne razlike numerirat će točke indeksima od 1 do n (zbog toga u tablici nema x_0).

x_k	$f(x_k)$	$\rho_1(x_k, x_{k+1})$	$\rho_2(x_k, x_{k+1}, x_{k+2})$	$\cdots$	$\rho_{n-1}(x_1, \ldots, x_n)$
x_1	$f(x_1)$				
		$\rho_1(x_1, x_2)$			
x_2	$f(x_2)$		$\rho_2(x_1, x_2, x_3)$		
		$\rho_1(x_1, x_2)$		$\ddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\rho_{n-1}(x_1, \ldots, x_n)$
		$\rho_1(x_{n-2}, x_{n-1})$		$\iddots$	
x_{n-1}	$f(x_{n-1})$		$\rho_2(x_{n-2}, x_{n-1}, x_n)$		
		$\rho_1(x_{n-1}, x_n)$			
x_n	$f(x_n)$				

Uz recipročne razlike, često se definiraju i inverzne razlike

$$\phi_0(x_0) = f(x_0), \quad \phi_1(x_0, x_1) = \frac{x_1 - x_0}{\phi_0(x_1) - \phi_0(x_0)},$$

odnosno

$$\phi_k(x_0, \ldots, x_k) = \frac{x_k - x_{k-1}}{\phi_{k-1}(x_0, \ldots, x_{k-2}, x_k) - \phi_{k-1}(x_0, \ldots, x_{k-2}, x_{k-1})}, \quad k \geq 2.$$

Postoji i veza između inverznih i recipročnih razlika. Nije teško pokazati da vrijedi

$$\phi_0(x_0) = \rho_0(x_0), \quad \phi_1(x_0, x_1) = \rho_1(x_0, x_1),$$

odnosno za $k \geq 2$

$$\phi_k(x_0, \ldots, x_k) = \rho_k(x_0, \ldots, x_k) - \rho_{k-2}(x_0, \ldots, x_{k-2}).$$

Pokažimo da vrijedi jedan važan identitet iz kojeg ćemo izvesti Thieleovu formulu. Prvo, u formuli za recipročne razlike uzmemo x_0 kao varijablu i označimo je s x. Tvrdimo da je

$$f(x) = f(x_1) + \frac{x - x_1}{\phi_1(x_1, x_2)^+} \frac{x - x_2}{\phi_2(x_1, x_2, x_3)^+} \cdots \frac{x - x_{n-1}}{\phi_{n-1}(x_1, \ldots, x_n)^+} \frac{x - x_n}{\rho_n(x, x_1, \ldots, x_n) - \rho_{n-2}(x_1, \ldots, x_{n-1})} \quad (7.15.1)\,.$$

Iz

$$\rho_1(x, x_1) = \frac{x - x_1}{f(x) - f(x_1)}$$

slijedi da je

$$f(x) = f(x_1) + \frac{x - x_1}{\rho_1(x, x_1)}. \quad (7.15.2)$$

Zatim, iz formule

$$\rho_2(x, x_1, x_2) = \frac{x - x_2}{\rho_1(x, x_1) - \rho_1(x_1, x_2)} + \rho_0(x_1)$$

slijedi da je

$$\rho_1(x, x_1) = \rho_1(x_1, x_2) + \frac{x - x_2}{\rho_2(x, x_1, x_2) - \rho_0(x_1)}.$$

Uvrštavanjem tog izraza u (7.15.2) dobivamo

$$f(x) = f(x_1) + \frac{x - x_1}{\rho_1(x_1, x_2) + \dfrac{x - x_2}{\rho_2(x, x_1, x_2) - \rho_0(x_1)}}. \quad (7.15.3)$$

Konačno, formulu (7.15.1) dobivamo indukcijom po n, uz korištenje definicije inverznih razlika.

Pokažimo još jednu zanimljivu činjenicu vezanu uz formulu (7.15.1). Ako izbrišemo zadnji član, onda će za racionalnu funkciju (verižni razlomak)

$$R(x) = f(x_1) + \frac{x - x_1}{\phi_1(x_1, x_2)^+} \frac{x - x_2}{\phi_2(x_1, x_2, x_3)^+} \cdots \frac{x - x_{n-1}}{\phi_{n-1}(x_1, \ldots, x_n)} \tag{7.15.4}$$

vrijediti

$$R(x_i) = f(x_i), \quad i = 1, \ldots, n.$$

To se odmah vidi iz (7.15.1), ako krenemo od x_n, jer je član

$$\frac{x - x_n}{\rho_n(x, x_1, \ldots, x_n) - \rho_{n-2}(x_1, \ldots, x_{n-1})} \tag{7.15.5}$$

jednak 0 za $x = x_n$, pa je $R(x_n) = f(x_n)$. Nakon toga, gledamo $R(x_{n-1})$ i $f(x_{n-1})$. Oni su za jednu verigu kraći i to za onu verigu koja sadrži "član razlike" (7.15.5). U svakoj daljnjoj točki $x_{n-2}, \ldots, x_1$, verižni je razlomak kraći za jednu verigu od prethodne.

Formula (7.15.4) zove se Thielova interpolaciona formula. Pokažimo na nekoliko primjera koliko je dobra ta interpolacija.

Primjer 7.15.1 *Aproksimirajte*

$$\operatorname{tg} 1.565$$

korištenjem Thieleove interpolacione formule, ako znamo vrijednosti funkcije tg *u točkama*

$$x_i = 1.53 + 0.01 * i, \quad i = 0, \ldots, 4.$$

Prvo izračunajmo recipročne razlike.

x_k	$f(x_k)$	ρ_1	ρ_2	ρ_3	ρ_4
1.53	24.49841				
		0.001255851			
1.54	32.46114		−0.0308670		
		0.000640314		2.96838	
1.55	48.07848		−0.0207583		3.56026
		0.000224507		2.97955	
1.56	92.62050		−0.0106889		
		0.000008597			
1.57	1255.76557				

Thielova interpolacija daje

$$R(x) = 24.49841 + \frac{x-1.53}{0.001255851^+}\,\frac{x-1.54}{-24.5293^+}\,\frac{x-1.55}{2.96713^+}\,\frac{x-1.56}{3.59113}\,.$$

Uvrštavanjem 1.565 *dobivamo*

$$R(1.565) = 172.5208,$$

dok je prava vrijednost

$$\mathrm{tg}(1.565) = 172.5211.$$

I sumacija redova može se znatno ubrzati korištenjem racionalne ekstrapolacije. Pretpostavimo da treba izračunati

$$S = \sum_{n=0}^{\infty} a_n.$$

Označimo s S_N, N-tu parcijalnu sumu reda

$$S_N = \sum_{n=0}^{N} a_n.$$

Ove vrijednosti S_N možemo interpretirati kao vrijednosti neke funkcije f u točkama N, ili u nekim drugim točkama, na primjer, u točkama $1/N$,

$$S_N = f\Big(\frac{1}{N}\Big).$$

Očito je da vrijedi

$$S = S_\infty = f(0).$$

Ideja je $f(0)$ izračunati kao ekstrapoliranu vrijednost od

$$f(1), f\Big(\frac{1}{2}\Big), f\Big(\frac{1}{3}\Big), \ldots,$$

ili za neke više N, iz

$$f\Big(\frac{1}{N_1}\Big), f\Big(\frac{1}{N_2}\Big), \ldots, \quad N_1 < N_2 < \cdots$$

Primjer 7.15.2 *Treba izračunati*

$$S = \sum_{n=1}^{\infty} \frac{1}{n^2},$$

korištenjem racionalne ekstrapolacije.

Uzet ćemo $N = 1, 2, 4, 8, 16$ *i izračunati*

$$S_n = \sum_{n=1}^{N} \frac{1}{n^2}.$$

Shvatimo li to kao funkciju od $x = 1/N$ *i označimo* $S(x) = S_N$*, onda možemo formirati tablicu recipročnih razlika.*

x	$S(x)$	ρ_1	ρ_2	ρ_3	ρ_4
$\frac{1}{16}$	1.584346533				
		-1.097945891			
$\frac{1}{8}$	1.527422052		-0.238678243		
		-1.204112002		4.826059143	
$\frac{1}{4}$	1.423611111		-0.166126405		0.016938420
		-1.44		9.947195880	
$\frac{1}{2}$	1.25		-0.089285214		
		-2			
1	1				

Thielova interpolacija daje

$$R(x) = 1.584346533 + \frac{x - \frac{1}{16}}{-1.097945891^+} \frac{x - \frac{1}{8}}{-1.823024776^+} \frac{x - \frac{1}{4}}{5.924005034^+} \frac{x - \frac{1}{2}}{0.255616663}.$$

Uvrštavanjem 0 *dobivamo*

$$R(0) = 1.644927974,$$

dok je prava vrijednost

$$S_\infty = \frac{\pi^2}{6} = 1.644934067.$$

Zanimljivo je spomenuti što se dobije ako samo zbrajamo članove reda i ne ekstrapoliramo. Vidjet ćemo da taj red vrlo sporo konvergira. Na primjer, dobivamo

$$S_{3000} = 1.644601, \quad S_{30000} = 1.644901, \quad S_{10000} = 1.644834, \quad S_{100000} = 1.644924.$$

8. Rješavanje nelinearnih jednadžbi

8.1. Općenito o iterativnim metodama

Računanje nultočaka nelinearnih funkcija jedan je od najčešćih zadataka primijenjene matematike. Općenito, neka je zadana funkcija

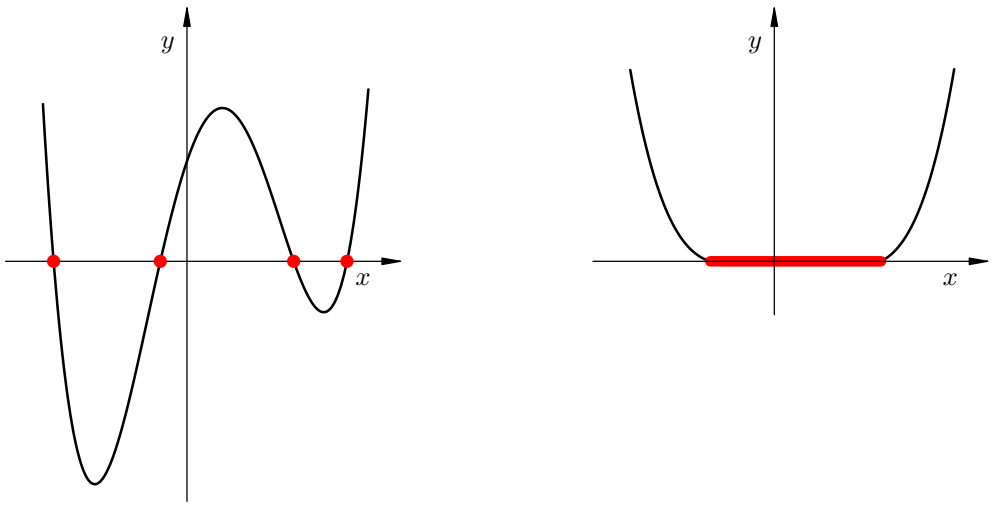
$$f : I \to \mathbb{R},$$

gdje je I neki interval. Tražimo sve one $x \in I$ za koje je

$$f(x) = 0.$$

Takve točke x zovu se rješenja, korijeni pripadne jednadžbe ili nultočke funkcije f.

U pravilu, pretpostavljamo da je f **neprekidna** na I i da su joj nultočke izolirane. U protivnom postojao bi problem konvergencije.



Traženje nultočki na zadanu točnost sastoji se od dvije faze.

1. Izolacije jedne ili više nultočki, tj. nalaženje intervala I unutar kojeg se nalazi bar jedna nultočka. Ovo je teži dio posla i obavlja se na temelju analize toka funkcije.
2. Iterativno nalaženje nultočke na traženu točnost.

Postoji mnogo metoda za nalaženje nultočaka nelinearnih funkcija na zadanu točnost. One se bitno razlikuju po tome hoće li uvijek konvergirati, tj. imamo li sigurnu konvergenciju ili ne i po brzini konvergencije. Uobičajen je slučaj da brze metode nemaju sigurnu konvergenciju, dok je sporije metode imaju.

Brzina konvergencije se definira pomoću reda konvergencije metode.

Definicija 8.1.1 *Niz iteracija* $(x_n, n \in \mathbb{N}_0)$ *konvergira prema točki* α *s redom konvergencije* p, $p \geq 1$ *ako vrijedi*

$$|\alpha - x_n| \leq c\,|\alpha - x_{n-1}|^p, \quad n \in \mathbb{N} \tag{8.1.1}$$

za neki $c > 0$. *Ako je* $p = 1$, *kažemo da niz konvergira linearno prema* α. *U tom slučaju je nužno da je* $c < 1$ *i obično se* c *naziva faktor linearne konvergencije.*

Relacija (8.1.1) katkad nije zgodna za linearne iterativne algoritme. Ako za slučaj $p = 1$ i $c < 1$ u (8.1.1), upotrijebimo indukciju po n, onda dobivamo da je

$$|\alpha - x_n| \leq c^n |\alpha - x_0|, \quad n \in \mathbb{N}. \tag{8.1.2}$$

Katkad će biti mnogo lakše pokazati (8.1.2) nego (8.1.1). I u slučaju (8.1.2), reći ćemo da niz iteracija konvergira linearno s faktorom c.

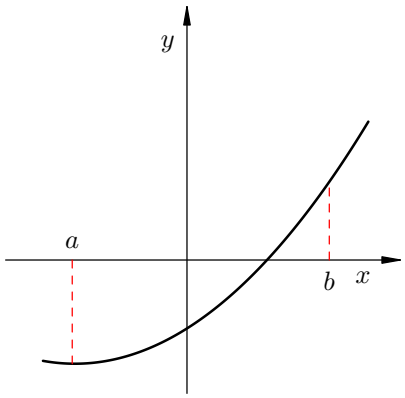
8.2. Metoda raspolavljanja (bisekcije)

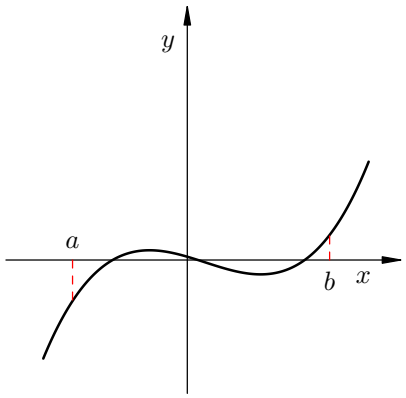
Najjednostavnija metoda nalaženja nultočaka funkcije je metoda raspolavljanja. Ona funkcionira za neprekidne funkcije, ali zbog toga ima i najlošiju ocjenu pogreške.

Osnovna pretpostavka za primjenu algoritma raspolavljanja je **neprekidnost** funkcije f na intervalu $[a, b]$ i uvjet

$$f(a) \cdot f(b) < 0.$$

Prethodna relacija znači da funkcija f ima na intervalu $[a, b]$ **barem jednu** nultočku.

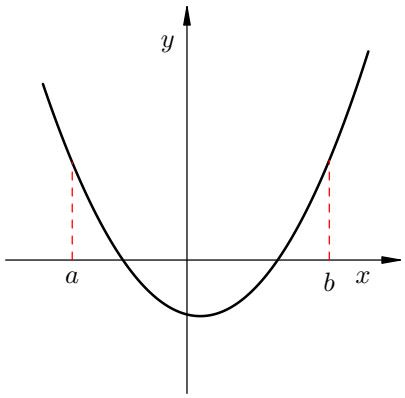


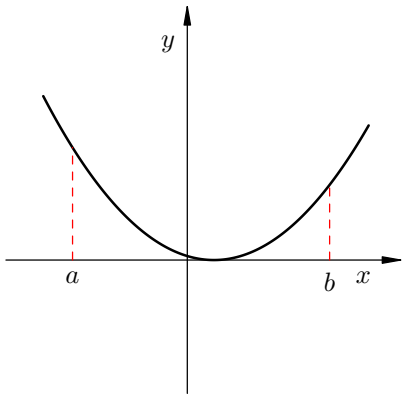


S druge strane, ako je

$$f(a) \cdot f(b) > 0,$$

to **ne mora** značiti da f nema unutar $[a, b]$ nultočku. Na primjer, moglo se dogoditi da smo loše separirali nultočke i da f ima unutar $[a, b]$ paran broj (brojeći ih s višestrukostima) nultočaka, ili nultočku parnog reda.





Dok je za prvi primjer s prethodne slike boljom separacijom nultočki lako postići $f(a) \cdot f(b) < 0$, za drugi je primjer to nemoguće! Dakle, nultočke parnog reda nemoguće je naći metodom bisekcije.

Ako vrijede polazne pretpostavke, metoda raspolavljanja (bez dodatnih uvjeta) konvergirat će prema nekoj nultočki iz intervala $[a, b]$.

Algoritam raspolavljanja je vrlo jednostavan. Označimo s α pravu nultočku funkcije, a zatim s $a_0 := a$, $b_0 := b$ i x_0 polovište $[a_0, b_0]$, tj.

$$x_0 = \frac{a_0 + b_0}{2}.$$

Neka je $n \geq 1$. U n-tom koraku algoritma konstruiramo interval $[a_n, b_n]$ kojemu je duljina polovina duljine prethodnog intervala, ali tako da je nultočka ostala unutar intervala $[a_n, b_n]$.

Konstrukcija intervala $[a_n, b_n]$ sastoji se u raspolavljanju intervala $[a_{n-1}, b_{n-1}]$ točkom x_{n-1} i to tako da je

$$\begin{aligned} a_n &= x_{n-1},\ b_n = b_{n-1} \quad \text{ako je} \quad f(a_{n-1}) \cdot f(x_{n-1}) > 0, \\ a_n &= a_{n-1},\ b_n = x_{n-1} \quad \text{ako je} \quad f(a_{n-1}) \cdot f(x_{n-1}) < 0. \end{aligned}$$

Postupak zaustavljamo kad je

$$|\alpha - x_n| \leq \varepsilon.$$

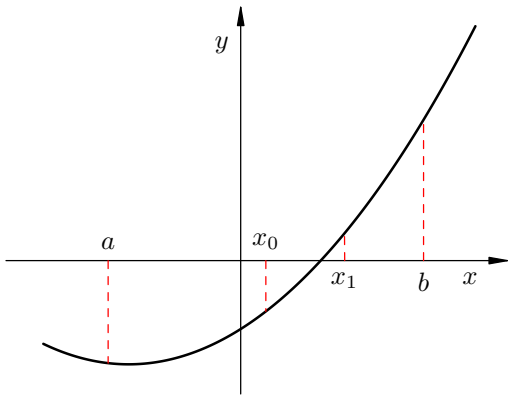
Kako ćemo znati da je prethodna relacija ispunjena ako ne znamo α? Jednostavno, budući da je x_n polovište intervala $[a_n, b_n]$, a $\alpha \in [a_n, b_n]$, onda je

$$|\alpha - x_n| \leq b_n - x_n,$$

pa je dovoljno postaviti zahtjev

$$b_n - x_n \leq \varepsilon.$$

Grafički, metoda raspolavljanja izgleda ovako



Napišimo algoritam za metodu raspolavljanja.

Algoritam 8.2.1 (Metoda raspolavljanja)

$x := (a + b)/2;$
while $b - x > \varepsilon$ **do**
  begin;
  if $f(x) * f(b) < 0.0$ **then**
    $a := x$
  else
    $b := x;$
  $x := (a + b)/2;$
  end;
{ Na kraju je $x \approx \alpha$. }

Iz konstrukcije metode lako se izvodi pogreška n-te aproksimacije nultočke. Vrijedi

$$|\alpha - x_n| \leq b_n - x_n = \frac{1}{2}(b_n - a_n) = \frac{1}{2^2}(b_{n-1} - a_{n-1}) = \cdots = \frac{1}{2^{n+1}}(b-a). \quad (8.2.1)$$

Primijetite da je

$$\frac{b-a}{2} = b - x_0,$$

pa bismo relaciju (8.2.1) mogli pisati kao

$$|\alpha - x_n| \leq \frac{1}{2^n}(b - x_0).$$

Ova relacija podsjeća na (8.1.2), ali zdesna se nigdje ne pojavljuje $|\alpha - x_0|$. Ipak desna strana daje nam naslutiti da će konvergencija biti dosta spora.

Relacija (8.2.1) omogućava da unaprijed odredimo koliko je koraka raspolavljanja potrebno da bismo postigli točnost ε. Da bismo postigli da je $|\alpha - x_n| \leq \varepsilon$, dovoljno je zahtijevati

$$\frac{1}{2^{n+1}}(b-a) \leq \varepsilon.$$

Zadnja nejednakost je ekvivalentna s

$$\frac{b-a}{\varepsilon} \leq 2^{n+1},$$

a zatim logaritmiranjem nejednakosti dobivamo $\log(b-a) - \log\varepsilon \leq (n+1)\log 2$, odnosno

$$n \geq \frac{\log(b-a) - \log\varepsilon}{\log 2} - 1, \quad n \in \mathbb{N}_0.$$

Ako je funkcija f još i klase $C^1[a,b]$, tj. ako f ima i neprekidnu prvu derivaciju, može se dobiti dinamička ocjena za udaljenost aproksimacije nultočke od prave nultočke.

Po Teoremu srednje vrijednosti za funkciju f imamo

$$f(x_n) = f(\alpha) + f'(\xi)(x_n - \alpha),$$

pri čemu je ξ između x_n i α. Prvo iskoristimo da je α nultočka, tj. $f(\alpha) = 0$, a zatim uzmemo apsolutne vrijednosti obje strane. Dobivamo

$$|f(x_n)| = |f'(\xi)|\,|\alpha - x_n|,$$

odakle slijedi

$$|\alpha - x_n| = \frac{|f(x_n)|}{|f'(\xi)|}. \quad (8.2.2)$$

Znamo da je α u intervalu $[a_n, b_n]$. Pretpostavimo da možemo ocijeniti

$$|f'(\xi)| \geq m_1, \quad m_1 = \min_{x\in[a,b]} |f'(x)|.$$

Ako je $m_1 > 0$, uvrštavanjem prethodne ocjene u (8.2.2), izlazi

$$|\alpha - x_n| \leq \frac{|f(x_n)|}{m_1}.$$

Drugim riječima, ako želimo da je $|\alpha - x_n| \leq \varepsilon$, dovoljno je zahtijevati

$$\frac{|f(x_n)|}{m_1} \leq \varepsilon,$$

odnosno

$$|f(x_n)| \leq m_1\varepsilon.$$

8.3. Regula falsi (metoda pogrešnog položaja)

U prethodnom poglavlju opisali smo metodu raspolavljanja, koja ima sigurnu konvergenciju, ali je vrlo spora. Prirodan je pokušaj ubrzavanja te metode je *regula falsi*. Konstruirat ćemo metodu koja će, ponovno biti konvergentna, čim se nultočka nalazi unutar $[a, b]$.

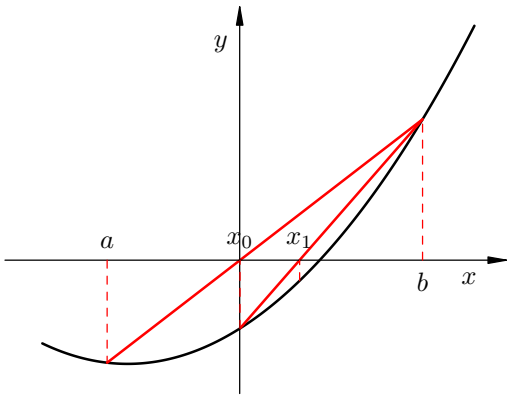
Pretpostavimo da je funkcija $f : [a, b] \to \mathbb{R}$ neprekidna na $[a, b]$ i da vrijedi

$$f(a) \cdot f(b) < 0.$$

Aproksimirajmo funkciju f pravcem koji prolazi točkama $(a, f(a))$, $(b, f(b))$. Njegova je jednadžba

$$y - f(b) = \frac{f(a) - f(b)}{a - b}(x - b), \quad \text{odnosno} \quad y - f(a) = \frac{f(b) - f(a)}{b - a}(x - a).$$

Nultočku α funkcije f možemo aproksimirati nultočkom tog pravca, točkom x_0. Nakon toga, pomaknemo ili točku a ili točku b u x_0, ali tako da nultočka α ostane unutar novodobivenog intervala. Postupak ponavljamo sve dok ne postignemo željenu točnost.



Točka x_0 dobiva se jednostavno iz jednadžbe pravca, pa je

$$x_0 = b - f(b)\,\frac{b-a}{f(b)-f(a)} = a - f(a)\,\frac{a-b}{f(a)-f(b)},$$

ili drugačije zapisano

$$x_0 = b - \frac{f(b)}{f[a,b]} = a - \frac{f(a)}{f[a,b]}, \tag{8.3.1}$$

gdje je

$$f[a,b] = \frac{f(b)-f(a)}{b-a},$$

prva podijeljena razlika (vidi primjer 1.1.4).

Postoji nekoliko ozbiljnih problema s ovom metodom, iako je aproksimacija pravcem i zatvaranje nultočke u određeni interval sasvim dobra ideja.

Izvedimo red konvergencije metode. Iskoristimo relaciju (8.3.1) za x_0, pomnožimo je s -1 i dodajmo α na obje strane. Dobivamo

$$\begin{aligned}
\alpha - x_0 &= \alpha - b + \frac{f(b)}{f[a,b]} = (\alpha-b)\Big(1 + \frac{f(b)}{(\alpha-b)f[a,b]}\Big) \\
&= (\alpha-b)\left(1 + \frac{f(b)-f(\alpha)}{(\alpha-b)f[a,b]}\right) = (\alpha-b)\left(1 + (b-\alpha)\,\frac{f[b,\alpha]}{(\alpha-b)f[a,b]}\right) \\
&= (\alpha-b)\left(1 - \frac{f[b,\alpha]}{f[a,b]}\right) = (\alpha-b)\,\frac{f[a,b]-f[b,\alpha]}{f[a,b]} \\
&= -(\alpha-b)\,(\alpha-a)\frac{f[a,b,\alpha]}{f[a,b]},
\end{aligned}$$

pri čemu je po definiciji $f[a,b,\alpha]$ druga podijeljena razlika (vidi primjer 1.1.4)

$$f[a,b,\alpha] = \frac{f[b,\alpha]-f[a,b]}{\alpha-a}.$$

Ako je f klase $C^1[a,b]$, onda po teoremu 1.1.2 srednje vrijednosti imamo

$$f[a,b] = f'(\xi), \quad \xi \in [a,b].$$

Na sličan način, ako je f klase $C^2[a,b]$, imamo

$$f[a,b,\alpha] = \frac{1}{2}\,f''(\zeta),$$

gdje se ζ nalazi između minimima i maksimuma vrijednosti a, b, α. Iskoristimo li te dvije relacije, za funkcije klase $C^2[a,b]$ dobivamo sljedeću ocjenu

$$\alpha - x_0 = -(\alpha-b)\,(\alpha-a)\frac{f''(\zeta)}{2f'(\xi)}. \tag{8.3.2}$$

Da bismo pojednostavnili analizu, pretpostavimo da je $f'(\alpha) \neq 0$ i α je jedini korijen unutar $[a, b]$. Također, pretpostavimo da je $f''(a) \geq 0$ za sve $x \in [a, b]$. Razlikujemo dva slučaja:

Slučaj $f'(x) > 0$.

U tom je slučaju f konveksna rastuća funkcija, a spojnica točaka $(a, f(a))$ i $(b, f(b))$ se uvijek nalazi **iznad** funkcije f. Uvrštavanjem podataka o prvoj i drugoj derivaciji u (8.3.2), dobivamo da je desna strana (8.3.2) veća od 0, tj. $\alpha > x_0$, pa će se u sljedećem koraku pomaknuti a. Isto će se dogoditi u svim narednim koracima. Drugim riječima, α neprestano ostaje desno od aproksimacija x_n. Promatramo li (8.3.2), to znači da je b fiksan, pa za proizvoljnu iteraciju x_n dobivamo

$$\alpha - x_n = -(\alpha - b)\,(\alpha - a_n)\frac{f''(\zeta_n)}{2f'(\xi_n)}.$$

Uzimanjem apsolutnih vrijednosti zdesna i slijeva, slijedi da je u tom slučaju konvergencija *regule falsi* linearna.

Pogled na sličnu ocjenu za metodu bisekcije, odmah kaže da ne bi trebalo biti preteško konstruirati primjere kad je metoda bisekcije brža no *regula falsi*.

Slučaj $f'(x) < 0$.

U ovom slučaju je aproksimacija nultočke uvijek desno od α, a uvijek se pomiče b. Analiza ovog slučaja vrlo je slična prethodnoj.

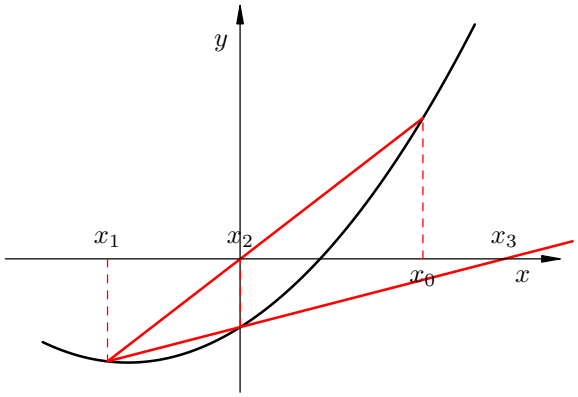
8.4. Metoda sekante

Ako graf funkcije f aproksimiramo sekantom, slično kao kod *regule falsi*, samo ne zahtijevamo da nultočka funkcije f ostane "zatvorena" unutar posljednje dvije iteracije, dobili smo metodu sekante. Time smo izgubili svojstvo sigurne konvergencije, ali se nadamo da će metoda, kad konvergira konvergirati brže nego *regula falsi*.

Počinjemo s dvije početne točke x_0 i x_1 i povlačimo sekantu kroz $(x_0, f(x_0))$, $(x_1, f(x_1))$. Ta sekanta siječe os x u točki x_2. Postupak nastavljamo povlačenjem sekante kroz posljednje dvije točke $(x_1, f(x_1))$ i $(x_2, f(x_2))$. Formule za metodu sekante dobivaju se iteriranjem početne formule za *regulu falsi*, tako da dobivamo

$$x_{n+1} = x_n - f(x_n)\,\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}. \tag{8.4.1}$$

Grafički to izgleda ovako.



Primijetite da je treće iteracija izašla izvan početnog intervala, pa metoda sekante ne mora konvergirati. Jednako tako, da smo "prirodno" numerirali prve dvije točke, tako da je $x_0 < x_1$, imali bismo konvergenciju prema rješenju.

Iskoristimo li ocjenu (8.3.2) za svaki n, dobit ćemo red konvergencije metode sekante, uz odgovarajuće pretpostavke. Imamo

$$\alpha - x_{n+1} = -(\alpha - x_n)\,(\alpha - x_{n-1})\frac{f''(\zeta_n)}{2f'(\xi_n)}. \tag{8.4.2}$$

Teorem 8.4.1 *Neka su f, f' i f'' neprekidne za sve x u nekom intervalu koji sadrži jednostruku nultočku α. Primijetite da jednostrukost nultočke osigurava $f'(\alpha) \neq 0$. Ako su početne aproksimacije x_0 i x_1 izabrane dovoljno blizu α, niz iteracija x_n konvergirat će prema α s redom konvergencije p, gdje je*

$$p = \frac{1+\sqrt{5}}{2} \approx 1.618.$$

Dokaz. Budući da je nultočka jednostruka, postoji okolina $I = [\alpha - \varepsilon, \alpha + \varepsilon]$, $\varepsilon > 0$ nultočke α, takva da je u njoj $f'(x) \neq 0$. Tada je dobro definiran broj

$$M = \frac{\max\limits_{x\in I} |f''(x)|}{2\min\limits_{x\in I} |f'(x)|}.$$

Zbog relacije (8.4.2), za sve $x_0, x_1 \in I$ vrijedi

$$|\alpha - x_2| \leq |\alpha - x_1|\,|\alpha - x_0|\,M.$$

Da bismo skratili zapis, neka je

$$e_n = \alpha - x_n$$

greška n-te iteracije (aproksimacije nultočke). Množenjem prethodne nejednakosti s M dobivamo

$$M|e_2| \leq M|e_1|\, M|e_0|.$$

Nadalje, pretpostavimo da su x_0 i x_1 izabrani toliko blizu nultočke da vrijedi

$$\delta = \max\{M|e_0|, M|e_1|\} < 1.$$

Odatle odmah slijedi

$$M|e_2| \leq \delta^2 < \delta,$$

pa je

$$|e_2| < \frac{\delta}{M} = \max\{|e_0|, |e_1|\} \leq \varepsilon,$$

odnosno

$$x_2 \in [\alpha - \varepsilon, \alpha + \varepsilon] = I.$$

Primijenimo li induktivno taj argument, dobivamo

$$\begin{aligned} M|e_3| &\leq M|e_2|M|e_1| \leq \delta^2 \cdot \delta = \delta^3 \\ M|e_4| &\leq M|e_3|M|e_2| \leq \delta^5. \end{aligned}$$

Općenito, ako je

$$M|e_{n-1}| \leq \delta^{q_{n-1}}, \quad M|e_n| \leq \delta^{q_n},$$

onda je

$$M|e_{n+1}| \leq M|e_n|M|e_{n-1}| \leq \delta^{q_n+q_{n-1}} = \delta^{q_{n+1}}.$$

Dakle, niz q_n je definiran rekurzijom

$$q_{n+1} = q_n + q_{n-1}, \quad n \geq 1,$$

s početnim uvjetima $q_0 = q_1 = 1$. Prethodna rekurzija je rekurzija za Fibonaccijeve brojeve, pa se lako izračunava njeno eksplicitno rješenje. Ono zadovoljava diferencijsku jednadžbu

$$q_{n+1} - q_n - q_{n-1} = 0,$$

uz zadane početne $q_0 = q_1 = 1$. Karakteristična jednadžba je

$$k^2 - k - 1 = 0,$$

pa su njena rješenja

$$k_{1,2} = \frac{1 \pm \sqrt{5}}{2}.$$

Označimo li

$$r_0 = \frac{1 + \sqrt{5}}{2}, \quad r_1 = \frac{1 - \sqrt{5}}{2},$$

onda je opće rješenje te diferencijske jednadžbe

$$q_n = c_0 r_0^n + c_1 r_1^n.$$

Konstante c_0 i c_1 određujemo iz početnih uvjeta. Dobivamo

$$1 = q_0 = c_0 + c_1$$
$$1 = q_1 = c_0 r_0 + c_1 r_1.$$

Rješavanjem ovog para jednadžbi, dobivamo

$$c_0 = \frac{1}{\sqrt{5}} r_0, \quad c_1 = -\frac{1}{\sqrt{5}} r_1,$$

pa je

$$q_n = \frac{1}{\sqrt{5}} (r_0^{n+1} - r_1^{n+1}), \quad n \geq 0.$$

Budući da je

$$r_0 \approx 1.618, \quad r_1 \approx -0.618,$$

onda za velike n, $r_1^{n+1} \to 0$, pa je

$$q_n \approx \frac{1}{\sqrt{5}} (1.618)^{n+1}.$$

Vratimo se na e_n. Ovim smo pokazali da je

$$|e_n| \leq \frac{1}{M} \delta^{q_n}, \quad n \geq 0.$$

Budući da $\delta < 0$ i $q_n \to \infty$ za $n \to \infty$, dobivamo da $x_n \to \alpha$.

Ovaj "dokaz" nije matematički korektan jer smo koristili samo gornje ograde. Ipak, on nam daje ideju o redu konvergencije, koji je zaista $p = r_0$, ali pravi dokaz je mnogo teži. ■

Kod metode sekante postoji nekoliko problema. Prvi je da može divergirati ako početne aproksimacije nisu dobro odabrane. Drugi problem se može javiti zbog kraćenja u brojniku i (posebno) nazivniku kvocijenta

$$\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})},$$

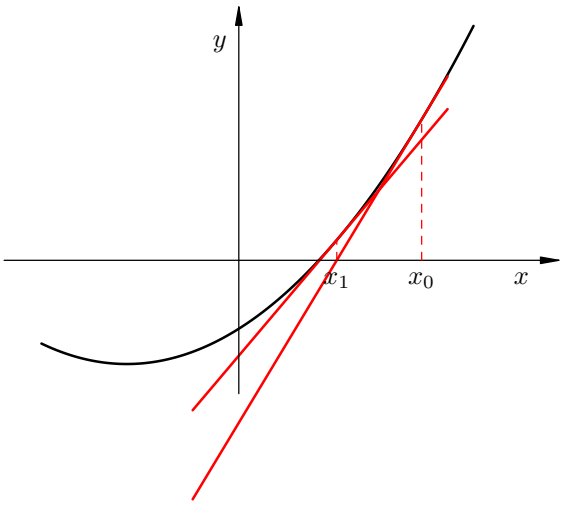
kad $x_n \to \alpha$. Osim toga, budući da iteracije ne "zatvaraju" nultočku s obje strane nije lako reći kad treba zaustaviti iterativni proces.

Konačno, primijetimo da je za svaku iteraciju metode sekante potrebno samo jednom izvrednjavati funkciju f i to u točki x_n, jer $f(x_{n-1})$ čuvamo od prethodne iteracije.

8.5. Metoda tangente (Newtonova metoda)

Ako graf funkcije f umjesto sekantom, aproksimiramo tangentom, dobili smo metodu tangente ili Newtonovu metodu. Slično kao i kod sekante, time smo izgubili svojstvo sigurne konvergencije, ali se nadamo da će metoda brzo konvergirati.

Pretpostavimo da je zadana početna točka x_0. Ideja metode je povući tangentu u točki $(x_0, f(x_0))$ i definirati novu aproksimaciju x_0 u točki gdje ona siječe os x.



Geometrijski izvod je jednostavan. U točki x_n napiše se jednadžba tangente i pogleda se gdje siječe os x. Jednadžba tangente je

$$y - f(x_n) = f'(x_n)(x - x_n),$$

odakle izlazi da je nova aproksimacija $x_{n+1} := x$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Primijetite da je prethodna formula usko vezana uz metodu sekante, jer je

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

Do Newtonove metode može se doći i na drugačiji način. Pretpostavimo li da je funkcija f dva puta neprekidno derivabilna (na nekom području oko α), onda je možemo razviti u Taylorov red oko x_n do uključivo prvog člana. Dobivamo

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(\xi_n)}{2}(x - x_n)^2,$$

pri čemu je ξ_n između x i x_n. Uvrštavanjem $x = \alpha$, dobivamo

$$0 = f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + \frac{f''(\xi_n)}{2}(\alpha - x_n)^2.$$

Premještanjem, uz pretpostavku $f'(x_n) \neq 0$, izlazi

$$\alpha = x_n - \frac{f(x_n)}{f'(x_n)} - (\alpha - x_n)^2 \frac{f''(\xi_n)}{2f'(x_n)}.$$

Primijetite da prva dva člana zdesna daju x_{n+1}, pa dobivamo

$$\alpha - x_{n+1} = -(\alpha - x_n)^2 \frac{f''(\xi_n)}{2f'(x_n)}. \tag{8.5.1}$$

Iz (8.5.1), odmah čitamo da je Newtonova metoda, kad konvergira kvadratično konvergentna. Ipak, treba biti oprezan, jer takav zaključak vrijedi samo ako $f'(x_n)$ ne teži k nuli tijeku cijelog procesa, tj. ako je $f'(\alpha) \neq 0$, dakle ako je nultočka jednostruka.

Slično, kao kod metode sekante, možemo dokazati sljedeći teorem o konvergenciji Newtonove metode.

Teorem 8.5.1 *Neka su f, f' i f'' neprekidne za sve x u nekom intervalu koji sadrži jednostruku nultočku α. Ako je početna aproksimacija x_0 izabrana dovoljno blizu nultočke α, niz iteracija x_n konvergirat će prema α s redom konvergencije $p = 2$. Čak štoviše, vrijedi*

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} = -\frac{f''(\alpha)}{2f'(\alpha)}.$$

Dokaz. Izaberimo interval $I = [\alpha - \varepsilon, \alpha + \varepsilon]$ oko nultočke u kojem su funkcije f, f' i f'' neprekidne i neka je

$$M = \frac{\max\limits_{x\in I} |f''(x)|}{2 \min\limits_{x\in I} |f'(x)|}.$$

Za sve $x_0 \in I$, korištenjem (8.5.1), dobivamo

$$|\alpha - x_1| \leq M|\alpha - x_0|^2,$$

odnosno

$$M|\alpha - x_1| \leq (M|\alpha - x_0|)^2.$$

Izaberimo x_0 tako da zadovoljava $|\alpha - x_0| \leq \varepsilon$ i $M|\alpha - x_0| < 1$. Tada je

$$M|\alpha - x_1| \leq M|\alpha - x_0|,$$

što pokazuje da je

$$|\alpha - x_1| \leq |\alpha - x_0| \leq \varepsilon.$$

Primjenom istog argumenta, induktivno dobivamo

$$|\alpha - x_n| \leq \varepsilon, \quad M|\alpha - x_n| < 1$$

za sve $n \geq 1$. Da bismo pokazali konvergenciju iskoristimo (8.5.1). Imamo

$$|\alpha - x_{n+1}| \leq M|\alpha - x_n|^2 \quad \Longrightarrow \quad M|\alpha - x_{n+1}| \leq (M|\alpha - x_n|)^2.$$

Matematičkom indukcijom lako pokazujemo

$$M|\alpha - x_n| \leq (M|\alpha - x_0|)^{2^n}, \quad \text{odnosno} \quad |\alpha - x_n| \leq \frac{1}{M}(M|\alpha - x_0|)^{2^n}.$$

Budući da je $M|\alpha - x_0| < 1$, odmah dobivamo $x_n \to \alpha$ kad $n \to \infty$.

Budući da u (8.5.1) ξ_n leži između x_n i α, onda mora biti $\xi_n \to \alpha$ kad $n \to \infty$. Zbog toga je

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} = -\lim_{n\to\infty} \frac{f''(\xi_n)}{2f'(x_n)} = -\frac{f''(\alpha)}{2f'(\alpha)}.$$

■

Jednostavnim riječima, prethodni teorem daje dovoljne uvjete za tzv. **lokalnu** konvergenciju Newtonove metode prema jednostrukoj nultočki. Lokalnost se odnosi na to da početna aproksimacija mora biti dovoljno blizu nultočke

$$|\alpha - x_0| \leq \varepsilon.$$

Veličina ε određena je uvjetom $M|\alpha - x_0| < 1$ koji osigurava konvergenciju, kao i uvjetima neprekidnosti funkcije i njenih prvih dviju derivacija. Uočimo da vrijedi

$$|\alpha - x_0| < \frac{1}{M},$$

pa nas privlči ideja da treba uzeti $\varepsilon = 1/M$. To, nažalost, nije dovoljan uvjet da vrijedi teorem (M općenito ovisi o ε). Ipak, u nekim situacijama možemo iskoristiti sličan uvjet za osiguranje konvergencije Newtonove metode.

Pretpostavimo da smo locirali nultočku funkcije f u segmentu $[a, b]$ i znamo da je f klase C^2 na tom segmentu. Neka je

$$M_2 = \max_{x\in[a,b]} |f''(x)|, \quad m_1 = \min_{x\in[a,b]} |f'(x)|.$$

Funkcija f je strogo monotona na $[a, b]$ onda i samo onda ako je $m_1 > 0$. Naime, funkcija f' je neprekidna na segmentu $[a, b]$ (kompaktan skup u $\mathbb{R}$), pa f' poprima svoj minimum i maksimum u nekoj točki segmenta. Ako je f monotono rastuća (padajuća), tada je $f'(x) > 0$ ($f'(x) < 0$) za sve x iz segmenta $[a, b]$, pa je $m_1 > 0$. Obratno, $m_1 > 0$ povlači monotonost funkcije f. Stoga, ako je $m_1 > 0$, f ima

jedinstvenu jednostruku nultočku α u $[a, b]$. U tim uvjetima umjesto "lokalnog" M, možemo koristiti "globalnu" veličinu

$$M' := \frac{M_2}{2m_1},$$

pri čemu u definiciju konstanti M_2 i m_1 ulazi cijeli interval $[a, b]$. Ako vrijedi

$$\frac{b-a}{2} < \frac{1}{M'},$$

onda možemo uzeti $\varepsilon = (b-a)/2$, a startna točka je polovište intervala $x_0 := (a+b)/2$. Zbog

$$|x_0 - \alpha| \leq \varepsilon < 1/M',$$

imamo sigurnu konvergenciju iteracija prema nultočki. Ako vrijedi i jači uvjet

$$b - a < \frac{1}{M'},$$

onda bilo koja startna točka $x_0 \in [a, b]$ daje sigurnu konvergenciju.

Naravno, to možemo iskoristiti samo ako imamo dovoljno informacija o funkciji f tako da možemo izračunati M', odnosno M_2 i m_1. Umjesto M_2, možemo uzeti i neku gornju ogradu za M_2, a umjesto m_1, neku pozitivnu donju ogradu za m_1.

Veličine M_2 i m_1 daju i lokalne ocjene greške iteracija u Newtonovoj metodi, uz uvjet da su sve iteracije u segmentu $[a, b]$. Iz ranije relacije (8.5.1)

$$\alpha - x_n = -\frac{f''(\xi_{n-1})}{2f'(x_{n-1})}\,(\alpha - x_{n-1})^2,$$

gdje je ξ_{n-1} između α i x_{n-1}, odmah slijedi

$$|\alpha - x_n| \leq \frac{M_2}{2m_1}\,(\alpha - x_{n-1})^2.$$

Sličnu ocjenu smo već imali u prethodnom teoremu, samo s M umjesto M'. Ova ocjena nije naročito korisna za praksu, jer α ne znamo.

Da bi izveli za praksu pogodniju ocjenu greške, iskoristit ćemo Taylorov teorem. Za dvije susjedne iteracije u Newtonovoj metodi vrijedi

$$f(x_n) = f(x_{n-1}) + f'(x_{n-1})(x_n - x_{n-1}) + \frac{f''(\xi_{n-1})}{2}(x_n - x_{n-1})^2,$$

pri čemu je ξ_{n-1} između x_{n-1} i x_n. Po definiciji iteracija u Newtonovoj metodi vrijedi i

$$f(x_{n-1}) + f'(x_{n-1})(x_n - x_{n-1}) = 0,$$

pa je

$$f(x_n) = \frac{f''(\xi_{n-1})}{2}(x_n - x_{n-1})^2.$$

Koristeći pretpostavku x_{n-1}, $x_n \in [a, b]$, dobivamo

$$|f(x_n)| \leq \frac{M_2}{2}(x_n - x_{n-1})^2.$$

Kao i kod metode bisekcije, ako je $m_1 > 0$, iz (8.2.2) slijedi ocjena

$$|\alpha - x_n| \leq \frac{|f(x_n)|}{m_1}.$$

Kombinacijom ovih ocjena dobivamo

$$|\alpha - x_n| \leq \frac{M_2}{2m_1}(x_n - x_{n-1})^2,$$

što se može iskoristiti u praksi. Ako je ε gornja ograda za apsolutnu grešku (uobičajeno se to kaže samo tražena točnost), onda test

$$\frac{M_2}{2m_1}(x_n - x_{n-1})^2 \leq \varepsilon$$

ili napisan u formi u kojoj se uobičajeno koristi

$$|x_n - x_{n-1}| \leq \sqrt{\frac{2m_1\varepsilon}{M_2}}$$

garantira da je $|\alpha - x_n| \leq \varepsilon$. Jasno, s obzirom da računamo na računalu, zadnja nejednakost će vrijediti do na greške zaokruživanja. Uočimo da možemo koristiti i raniji test

$$\frac{|f(x_n)|}{m_1} \leq \varepsilon.$$

U prethodnim ocjenama greške koristili smo pretpostavku da je f strogo monotona na $[a, b]$, ili ekvivalentno, da prva derivacija ima isti predznak na cijelom intervalu. Ako još i druga derivacija ima fiksni predznak na tom intervalu, onda možemo dobiti i **globalnu** konvergenciju Newtonove metode.

Teorem 8.5.2 *Neka je $f \in C^2[a, b]$, $f(a) \cdot f(b) < 0$ i neka f' i f'' nemaju nultočke u $[a, b]$, (tj. f' i f'' imaju fiksni predznak na $[a, b]$). Ako polazna iteracija x_0 iz intervala $[a, b]$ zadovoljava uvjet*

$$f(x_0) \cdot f''(x_0) > 0,$$

onda niz iteracija dobiven Newtonovom metodom konvergira prema (jedinstvenoj jednostrukoj) nultočki α funkcije f.

Dokaz. Pretpostavimo, na primjer, da je $f'(x) > 0$ i $f''(x) > 0$ na cijelom $[a, b]$. Tada f raste, pa mora biti $f(a) < 0$ i $f(b) > 0$. Zbog $f''(x) > 0$, za startnu iteraciju x_0 mora vrijediti $f(x_0) > 0$. U praksi možemo uzeti $x_0 = b$, jer je to jedina točka za koju sigurno znamo da vrijedi $f(x_0) > 0$.

Neka je $(x_n, n \in \mathbb{N}_0)$ niz iteracija generiran Newtonovom metodom iz startne točke x_0 za koju je $f(x_0) > 0$. Dakle imamo

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

i znamo da je $x_0 > \alpha$. Tvrdimo da je $\alpha < x_n \leq x_0$ za svaki $n \in N_0$. Dokaz koristi matematičku indukciju, pri čemu bazu već imamo. Pretpostavimo da je $\alpha < x_n \leq x_0$. Onda je $f(x_n) > 0$ i $f'(x_n) > 0$, pa je

$$x_{n+1} < x_n \leq x_0,$$

što pokazuje da niz (x_n) monotono pada. Da bi dokazali i drugu nejednakost za x_{n+1}, iskoristimo Taylorovu formulu

$$0 = f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + \frac{f''(\xi_n)}{2}(\alpha - x_n)^2,$$

pri čemu je $\xi_n \in (\alpha, x_n) \subset [a, b]$. Zbog $f''(\xi_n) > 0$ imamo

$$f(x_n) + f'(x_n)(\alpha - x_n) < 0,$$

odakle slijedi

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} > \alpha.$$

Time je dokazan korak indukcije, pa je tvrdnja dokazana. Uočimo da smo usput dokazali i monotonost niza (x_n). Kako je taj padajući niz omeđen s α odozdo, postoji limes

$$\alpha' := \lim_{n\to\infty} x_n,$$

za koji vrijedi $\alpha \leq \alpha' \leq x_0$, tj. $\alpha' \in [a, b]$. Prijelazom na limes u formuli za Newtonove iteracije dobivamo

$$\alpha' = \alpha' - \frac{f(\alpha')}{f'(\alpha')},$$

odakle, koristeći $f'(\alpha') \neq 0$, slijedi $f(\alpha') = 0$. Kako je α jedina nultočka od f u intervalu $[a, b]$, mora vrijediti $\alpha = \alpha'$.

Preostala tri slučaja za predznake prve i druge derivacije se dokazuju potpuno analogno. ■

Uvjet $f(x_0) \cdot f''(x_0) > 0$ na izbor startne točke u prethodnom teoremu ima vrlo jednostavnu geometrijsku interpretaciju. Ako pogledamo graf funkcije f na $[a, b]$, startnu točku x_0 treba odabrati na "strmijoj" strani funkcije.

Primijetite da računanje korištenjem Newtonove metode može trajati dulje nego računanje upotrebom metode sekante (uz upotrebu istog kriterija zaustavljanja), iako Newtonova metoda ima veći red konvergencije nego metoda sekante. Objašnjenje leži u činjenici da se za svaki korak Newtonove metode mora izračunati i vrijednost funkcije i vrijednost derivacije u točki. Ako se derivacija komplicirano računa, metoda sekante koja zahtijeva samo jedno izvrednjavanje funkcije, će biti brža.

Prethodni teoremi daju samo dovoljne uvjete konvergencije pojedinih iterativnih metoda. U praktičnom računanju često imamo samo interval $[a, b]$ u kojem smo locirali nultočku funkcije f, a **nemamo** dodatne informacije o funkciji f iz kojih bismo mogli izvući zaključak o konvergenciji bržih iterativnih metoda. Zbog toga se ove metode katkad kombiniraju s metodom bisekcije na sljedeći način. Prvo izračunamo novu iteraciju po bržoj metodi i ako ona ostaje u trenutnom intervalu, onda ju prihvaćamo i s njom nastavljamo iteracije i skraćujemo interval. U protivnom, radimo korak bisekcije za smanjivanje intervala.

8.6. Metoda jednostavne iteracije

Pretpostavimo da tražimo α, rješenje jednadžbe

$$x = g(x). \tag{8.6.1}$$

Definiramo jednostavnu iteracionu funkciju (iteracionu funkciju koja "pamti" samo jednu prethodnu točku) s

$$x_{n+1} = g(x_n), \quad n \geq 0,$$

uz x_0 kao početnu aproksimaciju za α. Primijetite da Newtonova metoda pripada klasi jednostavnih iteracija, jer je

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Rješenja, tj. točke za koje je $x = g(x)$, zovu se **fiksne točke** od g. Uobičajeno, mi smo zainteresirani $f(x) = 0$, pa taj problem treba reformulirati na problem (8.6.1). Postoji mnogo načina za tu reformulaciju.

Primjer 8.6.1 *Reformulirajmo problem*

$$x^2 - a = 0, \quad a > 0$$

u oblik (8.6.1). Na primjer, to možemo napraviti na jedan od sljedećih načina:

1. $x = x^2 + x - a$, *ili općenitije* $x = x + c(x^2 - a)$ *za neki* $c \neq 0$,
2. $x = a/x$,
3. $x = 0.5\,(x + a/x)$.

Prirodno je pitanje kako se različite jednostavne iteracije ponašaju. Odgovor ćemo dobiti nizom sljedećih tvrdnji.

Lema 8.6.1 *Neka je funkcija* g *neprekidna na intervalu* $[a, b]$ *i neka je*

$$a \leq g(x) \leq b, \quad \forall x \in [a, b],$$

u oznaci $g([a, b]) \subseteq [a, b]$. *Tada jednostavna iteracija* $x = g(x)$ *ima bar jedno rješenje na* $[a, b]$.

Dokaz. Za neprekidnu funkciju $g(x) - x$ na intervalu $[a, b]$ vrijedi

$$g(a) - a \geq 0, \quad g(b) - b \leq 0.$$

Drugim riječima, funkcija $g(x) - x$ je promijenila predznak na intervalu $[a, b]$, a to može samo prolaskom kroz nultočku (neprekidna je!). ■

Lema 8.6.2 *Neka je funkcija* g *neprekidna na* $[a, b]$ *i neka je*

$$g([a, b]) \subseteq [a, b].$$

Nadalje, pretpostavimo da postoji konstanta λ, $0 < \lambda < 1$, *takva da vrijedi*

$$|g(x) - g(y)| \leq \lambda\,|x - y|, \quad \forall x, y \in [a, b].$$

Tada $x = g(x)$ *ima jedinstveno rješenje* α *unutar* $[a, b]$. *Također, niz iteracija*

$$x_n = g(x_{n-1}), \quad n \geq 1$$

konvergira prema α *za proizvoljni* $x_0 \in [a, b]$.

Dokaz. Prema prethodnoj lemi, postoji bar jedno rješenje $\alpha \in [a, b]$. Pokažimo da ne postoji više od jednog rješenja. Da bismo to pokazali, pretpostavimo suprotno, tj. postoje barem dva rješenja. Uzmimo bilo koja dva od tih rješenja i nazovimo ih α i β iz $[a, b]$. Budući da su to rješenja, vrijedi

$$g(\alpha) = \alpha \quad \text{i} \quad g(\beta) = \beta.$$

Po pretpostavci, uvažavajući prethodne jednakosti, dobivamo

$$|\alpha - \beta| = |g(\alpha) - g(\beta)| \leq \lambda |\alpha - \beta|,$$

ili drugim riječima

$$(1-\lambda)\,|\alpha-\beta| \le 0.$$

Budući da je $1-\lambda > 0$, mora biti $\alpha = \beta$.

Dokažimo još konvergenciju jednostavnih iteracija za proizvoljnu startnu točku $x_0 \in [a,b]$. Prvo, uočimo da $x_{n-1} \in [a,b]$ povlači da je $x_n = g(x_{n-1}) \in [a,b]$. Nadalje, vrijedi

$$|\alpha - x_n| = |g(\alpha) - g(x_{n-1})| \le \lambda |\alpha - x_{n-1}|,$$

odnosno indukcijom po n dobivamo

$$|\alpha - x_n| \le \lambda^n |\alpha - x_0|, \quad n \ge 1.$$

Ako pustimo $n \to \infty$, onda $\lambda^n \to 0$, pa vrijedi $x_n \to \alpha$. ■

Ako je g derivabilna na $[a,b]$, onda je po Teoremu srednje vrijednosti

$$g(x) - g(y) = g'(\xi)(x-y), \quad \xi \text{ između } x \text{ i } y$$

za sve x, $y \in [a,b]$. Definiramo

$$\lambda = \max_{x\in[a,b]} |g'(x)|, \tag{8.6.2}$$

onda možemo pisati

$$|g(x) - g(y)| = \lambda |x-y|, \quad \forall x \in [a,b].$$

Primijetite λ može biti veći od 1!

Teorem 8.6.1 *Neka je funkcija g neprekidno diferencijabilna na $[a,b]$, neka je*

$$g([a,b]) \subseteq [a,b],$$

i neka za λ iz (8.6.2) vrijedi

$$\lambda < 1. \tag{8.6.3}$$

Tada vrijedi:

1. *$x = g(x)$ ima točno jedno rješenje na $\alpha \in [a,b]$,*
2. *za proizvoljni $x_0 \in [a,b]$, za jednostavnu iteraciju $x_{n+1} = g(x_n)$, $n \ge 0$ vrijedi*

$$\lim_{n\to\infty} x_n = \alpha,$$

$$|\alpha - x_n| \le \lambda^n |\alpha - x_0|$$

i

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = g'(\alpha).$$

Dokaz. Sve tvrdnje ovog teorema dokazane su u prethodne dvije leme, osim posljednje relacije o brzini konvergencije.

Vrijedi

$$\alpha - x_{n+1} = g(\alpha) - g(x_n) = g'(\xi_n)(\alpha - x_n), \quad n \geq 0,$$

gdje je ξ_n neki broj između α i x_n. Budući da $x_n \to \alpha$, onda i $\xi_n \to \alpha$, pa vrijedi

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = \lim_{n\to\infty} g'(\xi_n) = g'(\alpha).$$

■

Pokažimo koliko je pretpostavka (8.6.3) značajna, tj. pretpostavimo da je $|g'(\alpha)| > 1$. Tada, ako imamo niz $x_{n+1} = g(x_n)$ i rješenje $\alpha = g(\alpha)$, vrijedi

$$\alpha - x_{n+1} = g(\alpha) - g(x_n) = g'(\xi_n)(\alpha - x_n).$$

Za x_n dovoljno blizu α, onda je i i $|g'(\xi_n)| > 1$, pa je $|\alpha - x_{n+1}| \geq |\alpha - x_n|$, pa konvergencija metode nije moguća.

Prethodni teorem se može malo i pojednostavniti, tako da se ne navodi eksplicitno interval $[a, b]$.

Teorem 8.6.2 *Neka je α rješenje jednostavne iteracije $x = g(x)$ i neka je g neprekidno diferencijabilna na nekoj okolini od α i neka je $|g'(\alpha)| < 1$. Tada vrijede svi rezultati Teorema 8.6.1, uz pretpostavku da je x_0 dovoljno blizu α.*

Dokaz. Uzmimo $I = [\alpha - \varepsilon, \alpha + \varepsilon]$ takav da je

$$\max_{x\in I} |g'(x)| \leq \lambda < 1.$$

Tada je $g(I) \subseteq I$, jer $|\alpha - x| \leq \varepsilon$ povlači

$$|\alpha - g(x)| = |g(\alpha) - g(x)| = |g'(\xi)||\alpha - x| \leq \lambda|\alpha - x| \leq \varepsilon.$$

Sada možemo primijeniti prethodni teorem za $[a, b] = I$. ■

Primjer 8.6.2 *U primjeru 8.6.1, definirali smo tri iteracijske funkcije.*

1. *Ako je $g(x) = x^2 + x - a$, onda je $g'(x) = 2x + 1$ i u nultočki $\alpha = \sqrt{a}$ je*

$$g'(\sqrt{a}) = 2\sqrt{a} + 1 > 1,$$

pa ta iteracijska funkcija neće konvergirati. U općenitijem je slučaju $g(x) = x + c(x^2 - a)$, pa je $g'(x) = 1 + 2cx$ i

$$g'(\sqrt{a}) = 1 + 2c\sqrt{a}.$$

Da bismo osigurali konvergenciju, mora biti

$$-1 < 1 + 2c\sqrt{a} < 1,$$

odnosno

$$-\frac{1}{\sqrt{a}} < c < 0.$$

2. *Ako je* $g(x) = a/x$, *onda je* $g'(x) = -a/x^2$, *pa je*

$$g'(\sqrt{a}) = -1.$$

3. *Ako je* $g(x) = 0.5\,(x + a/x)$, *onda je* $g'(x) = 0.5\,(1 - a/x^2)$, *pa je*

$$g'(\sqrt{a}) = 0.$$

Ovaj odjeljak završit ćemo promatranjem jednostavnih iteracionih funkcija, ali višeg reda konvergencije, kao što je, na primjer Newtonova metoda.

Teorem 8.6.3 *Neka je* α *rješenje od* $x = g(x)$ *i neka je* g p *puta neprekidno diferencijabilna za sve* x *u okolini* α, *za neki* $p \geq 2$. *Nadalje, pretpostavimo da je*

$$g'(\alpha) = \cdots = g^{(p-1)}(\alpha) = 0. \tag{8.6.4}$$

Ako je startna vrijednost x_0 *dovoljno blizu* α, *iteracijska funkcija*

$$x_{n+1} = g(x_n), \quad n \geq 0$$

imat će red konvergencije p *i*

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^p} = (-1)^{p-1} \frac{g^{(p)}(\alpha)}{p!}.$$

Dokaz. Razvijmo $g(x)$ u okolini α do uključivo $(p-1)$-ve potencije i napišimo ostatak. Zatim, uvrstimo $x = x_n$, pa dobivamo

$$x_{n+1} = g(x_n) = g(\alpha) + g'(\alpha)(x_n - \alpha) + \cdots + \frac{g^{(p-1)}(\alpha)}{(p-1)!}\,(x_n - \alpha)^{p-1} + \frac{g^{(p)}(\xi_n)}{p!}\,(x_n - \alpha)^p,$$

za neki ξ_n između x_n i α. Iskoristimo li da je $g(\alpha) = \alpha$ i pretpostavku (8.6.4), slijedi

$$x_{n+1} = \alpha + \frac{g^{(p)}(\xi_n)}{p!}\,(x_n - \alpha)^p,$$

odnosno

$$\alpha - x_{n+1} = -\frac{g^{(p)}(\xi_n)}{p!}\,(x_n - \alpha)^p.$$

Sada možemo primijeniti prethodni Teorem, koji pokazuje da će iteracijska funkcija konvergirati. Nadalje, to znači da $x_n \to \alpha$, pa i $\xi_n \to \alpha$, što daje traženu relaciju.■

Korištenjem prethodnog teorema možemo analizirati i Newtonovu metodu za koju je

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Deriviranjem dobivamo da je

$$g'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2},$$

pa je

$$g(\alpha) = 0,$$

uz pretpostavku da je $f'(\alpha) \neq 0$. Na sličan način, dobivamo

$$g''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)},$$

pa ako je $f''(\alpha) \neq 0$, možemo pokazati da je red konvergencije Newtonove metode jednak 2. Ako je $f'(\alpha) \neq 0$, $f''(\alpha) = 0$, onda će red konvergencije biti barem 3.

8.7. Newtonova metoda za višestruke nultočke

Promotrimo što će se dogoditi s konvergencijom Newtonove metode, ako funkcija f ima neprekidnih prvih $p+1$ derivacija i p-struku, $p \geq 2$ nultočku u α. Tada vrijedi

$$f(\alpha) = f'(\alpha) = \cdots = f^{(p-1)}(\alpha) = 0, \quad f^{(p)}(\alpha) \neq 0.$$

Samu funkciju f možemo napisati i u obliku

$$f(x) = (x-\alpha)^p\, h(x), \quad h(\alpha) \neq 0. \tag{8.7.1}$$

Ograničimo se samo na cjelobrojne p i promatrajmo Newtonovu metodu kao jednostavnu iteraciju,

$$x_{n+1} = g(x_n), \quad g(x) = x - \frac{f(x)}{f'(x)}.$$

Deriviranjem (8.7.1) dobivamo jednostavniji oblik za derivaciju

$$f'(x) = p(x-\alpha)^{p-1}h(x) + (x-a)^p h'(x),$$

pa je

$$g(x) = x - \frac{(x-\alpha)h(x)}{ph(x) + (x-\alpha)h'(x)}.$$

Deriviranjem funkcije g dobivamo

$$g'(x) = 1 - \frac{h(x)}{ph(x) + (x-\alpha)h'(x)} - (x-\alpha)\frac{d}{dx}\Big(\frac{h(x)}{ph(x) + (x-\alpha)h'(x)}\Big),$$

tako da je

$$g'(\alpha) = 1 - \frac{1}{p} \neq 0 \quad \text{za} \quad p > 1,$$

što pokazuje linearnu konvergenciju. Prema teoremu 8.6.1, faktor konvergencije bit će $g'(\alpha) = 1 - 1/p$, što je vrlo sporo. U prosjeku to je podjednako brzo kao bisekcija za $p = 2$ ili čak lošije od bisekcije za $p \geq 3$.

Kako možemo popraviti (ubrzati) Newtonovu metodu za p-struke nultočke, $p \geq 2$. Prvo pretpostavimo da znamo p. Definiramo iteracionu funkciju

$$g(x) = x - p\,\frac{f(x)}{f'(x)}.$$

Tada je

$$g'(x) = 1 - p\,\frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = 1 - p + p\,\frac{f(x)f''(x)}{(f'(x))^2}.$$

Iskoristimo li oblik funkcije f, dobivamo

$$\begin{aligned} f(x) &= (x-\alpha)^p h(x) \\ f'(x) &= (x-\alpha)^{p-1}[ph(x) + (x-\alpha)h'(x)] \\ f''(x) &= (x-\alpha)^{p-2}[p(p-1)h(x) + 2p(x-\alpha)h'(x) + (x-\alpha)^2 h''(x)], \end{aligned}$$

pa je

$$\lim_{x\to\alpha} \frac{f(x)f''(x)}{(f'(x))^2} = 1 - \frac{1}{p}.$$

Odatle odmah slijedi

$$\lim_{x\to\alpha} g'(x) = 0,$$

što pokazuje da ova modifikacija osigurava barem kvadratično konvergentnu metodu.

Što ćemo napraviti ako unaprijed ne znamo p? Primijetimo da funkcija

$$u(x) = \frac{f(x)}{f'(x)} = \frac{(x-\alpha)^p h(x)}{(x-\alpha)^{p-1}[ph(x) + (x-\alpha)h'(x)]} = \frac{(x-\alpha)h(x)}{ph(x) + (x-\alpha)h'(x)}$$

ima jednostruku nultočku u α. Drugim riječima, obična Newtonova metoda, ali primijenjena na $u(x)$ konvergirat će kvadratično,

$$x_{n+1} = x_n - \frac{u(x_n)}{u'(x_n)},$$

gdje je

$$u'(x) = \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = 1 - \frac{f''(x)}{f'(x)}\,u(x),$$

što pokazuje da ćemo dobiti kvadratičnu konvergenciju, iako ne znamo red nultočke, ali uz računanje još jedne derivacije funkcije (f'').

Slično vrijedi i za metodu sekante, koju ćemo ubrzati, kao da radimo s jednostrukim nultočkama, ako primijenimo metodu sekante za funkciju u

$$x_{n+1} = x_n - u(x_n)\,\frac{x_n - x_{n-1}}{u(x_n) - u(x_{n-1})}.$$

I u ovom slučaju postoji "cijena", a to je računanje f'.

8.8. Hibridna Brent–Dekkerova metoda

Brent–Dekkerova metoda smišljena je kao metoda koja će imati sigurnu konvergenciju, a nadamo se da će konvergirati brže nego metoda sekante, u najboljem slučaju kvadratično. Ona **ne zahtijeva** računanje derivacija, pa ako joj je red konvergencije u prosjeku bolji od sekante, možemo očekivati da će metoda po brzini biti slična Newtonovoj, ali će imati sigurnu konvergenciju.

Metoda se sasoji od tri dijela, koje grubo možemo opisati kao inverznu kvadratnu interpolaciju, metodu sekante i metodu bisekcije. Algoritam počinje metodom sekante koja generira treću točku. Ako se prema nekim kriterijima ta točka prihvaća kao dobra, možemo nastaviti raditi s kvadratnom interpolacijom kroz posljednje tri točke, ali inverznom (uloga x i y zamijenjena) i time dobivamo četvrtu točku.

Ako je treća točka odbačena kao loša, radi se jedan korak metode bisekcije. Drugim riječima, metoda se "vrti" između svoja tri sastavna dijela, a mi se nadamo da će rijetko koristiti bisekciju.

Točni parametri kad se neka aproksimacija nultočke prihvaća kao dobra, odnosno odbacuje kao loša su dosta složeni. Metoda je sastavni dio velikih numeričkih biblioteka programa, kao što je IMSL.

8.9. Primjeri

Prije konkretnih primjera, zanimljivo je napomenuti da se u praksi može sasvim dobro numerički procijeniti red konvergencije iterativne metode i taj podatak iskoristiti kao dodatna informacija o konvergenciji metode.

Kako se to radi? Prisjetimo se definicijske relacije (8.1.1) za red konvergencije p niza iteracija $(x_n, n \in \mathbb{N}_0)$ koji konvergira prema nultočki α. Za većinu "brzih"

lokalno konvergentnih metoda pokazali smo da postoji eksponent $p \geq 1$ za koji vrijedi

$$\lim_{n\to\infty} \frac{|\alpha - x_n|}{|\alpha - x_{n-1}|^p} = c > 0,$$

gdje je x_n niz iteracija generiran tom metodom, uz neki start dovoljno blizu nultočke. Ovako dobiveni p i c su "teorijske" vrijednosti ovih parametara, koje vrijede asimptotski — na limesu $n \to \infty$.

Prethodnu relaciju ne možemo direktno iskoristiti za računanje p i c, jer ne znamo α. Osim toga, i konstanta c obivcno ovisi o nekim vrijednostima derivacija funkcije f u točki α, pa ni c ne znamo.

Međutim, u okolini nultočke α sigurno vrijedi

$$|\alpha - x_n| \approx c|\alpha - x_{n-1}|^p, \tag{8.9.1}$$

za dovoljno velike n, s tim da opet ne znamo α. No, ako smo dovoljno blizu nultočke, onda možemo uzeti da je $\alpha \approx x_{n+1}$, pa vrijedi i

$$|x_{n+1} - x_n| \approx c|x_{n+1} - x_{n-1}|^p,$$

za dovoljno velike n, samo možda s nešto većom greškom.

Uzmimo sad da su c i p nepoznati. Da bismo ih izračunali, trebamo dvije jednadžbe za te dvije nepoznanice. Prvu je lako dobiti, tako da umjesto $\approx$ stavimo $=$ u prethodnoj relaciji. Naravno, tada više ne smijemo očekivati da ćemo dobiti "prave" vrijednosti za c i p, već neke približne vrijednosti c_{n+1} i p_{n+1}. Dakle, pretpostavljamo da za njih vrijedi

$$|x_{n+1} - x_n| = c_{n+1}|x_{n+1} - x_{n-1}|^{p_{n+1}}.$$

Nedostaje još jedna jednadžba. Iskoristimo li (8.9.1) za x_{n-1}, dobivamo

$$|x_{n+1} - x_{n-1}| = c_n|x_{n+1} - x_{n-2}|^{p_n}.$$

Pretpostavimo li da je $c_n \approx c_{n+1}$ i $p_n \approx p_{n+1}$ tj. c_n i p_n se ne mijenjaju brzo, imamo

$$\begin{aligned} |x_{n+1} - x_n| &= c'|x_{n+1} - x_{n-1}|^{p'} \\ |x_{n+1} - x_{n-1}| &= c'|x_{n+1} - x_{n-2}|^{p'}. \end{aligned}$$

Logaritmiranjem dobivamo

$$\begin{aligned} \ln|x_{n+1} - x_n| &= \ln c' + p' \ln|x_{n+1} - x_{n-1}| \\ \ln|x_{n+1} - x_{n-1}| &= \ln c' + p' \ln|x_{n+1} - x_{n-2}|, \end{aligned}$$

pa iz ovog linearnog sustava lako izračunamo p' i c'.

Nadalje, relacija

$$|\alpha - x_k| = c|\alpha - x_{k-1}|^p, \quad k \geq 1$$

uz stavljanje $\alpha = x_{n+1}$ za dovoljno veliki n može se iskoristiti i za traženje parametara c i p metodom najmanjih kvadrata. Definiramo li $f_k = |x_{n+1} - x_k|$, $z_k = |x_{n+1} - x_{k-1}|$, tražimo aproksimaciju oblika

$$\varphi(z) = cz^p$$

koja najbolje aproksimira skup podataka (z_k, f_k), $k = 1, \ldots, n$.

Primjer 8.9.1 *Usporedimo brzinu metoda za izračunavanje* $\sqrt[3]{1.5}$.

Taj problem možemo interpretirati i kao traženje realne pozitivne nultočke funkcije $f(x) = x^3 - 1.5$.

Zatražimo li pogrešku manju od 10^{-8}, *metodi bisekcije bit će potrebno* 27 *raspolavljanja, a približna nultočka bit će* $x_{27} = 1.144714239984751$.

Za grešku manju od 10^{-15} *Newtonova metoda trebat će samo* 7 *iteracija i* $x_7 = 1.144714242553332$.

Primjer 8.9.2 *Nađite nultočku funkcije*

$$f(x) = x^3 - 5.56x^2 + 9.1389x - 4.68999$$

korištenjem Newtonove metode, tako da greška bude manja od 10^{-15}.

Poučeni prethodnim primjerom očekujemo desetak iteracija. Umjesto toga, bilo nam je potrebno 30 *iteracija za traženu točnost,* $x_{30} = 1.230000000463810$, *što je signaliziralo da smo ili loše derivirali, ili funkcija ima višestruku nultočku.*

Nije teško pokazati da je 1.23 *dvostruka nultočka zadane funkcije, pa Newtonova metoda konvergira samo linearno. Modificiramo li Newtonovu metodu tako da korekciju pomnožimo s višestrukošću nultočke, za istu točnost bilo nam je potrebno samo* 7 *iteracija,* $x_7 = 1.229999999995655$, *a konvergencija je bila kvadratična.*

Zadatak 8.9.1 *Za funkcije iz prethodna dva primjera napišite programe za nalaženje nultočaka raznim metodama i numerički nađite brzinu konvergancije korištenih metoda.*

Primjer 8.9.3 *Nultočka funkcije*

$$f(x) = \operatorname{arctg}(x)$$

je $x = 0$, *ali Newtonova metoda neće konvergirati iz svake startne točke* x_0. *Naći ćemo točku* β *za koju vrijedi*

$$\begin{cases} |x_0| < |\beta| & \textit{Newtonova metoda sa startom } x_0 \textit{ konvergira,} \\ |x_0| > |\beta| & \textit{Newtonova metoda sa startom } x_0 \textit{ divergira,} \\ |x_0| = |\beta| & \textit{Newtonova metoda sa startom } x_0 \textit{ ciklira.} \end{cases}$$

Kako ćemo naći točku "cikliranja"? Funkcija $f(x) = \operatorname{arctg} x$ *je neparna, pa da bismo dobili cikliranje, dovoljno je da tangenta na funkciju u točki* β *presiječe os* x *u točki* $-\beta$*. Jednadžba tangente na* arctg *u točki* β *je*

$$y - \operatorname{arctg}\beta = \frac{1}{1+\beta^2}(x-\beta),$$

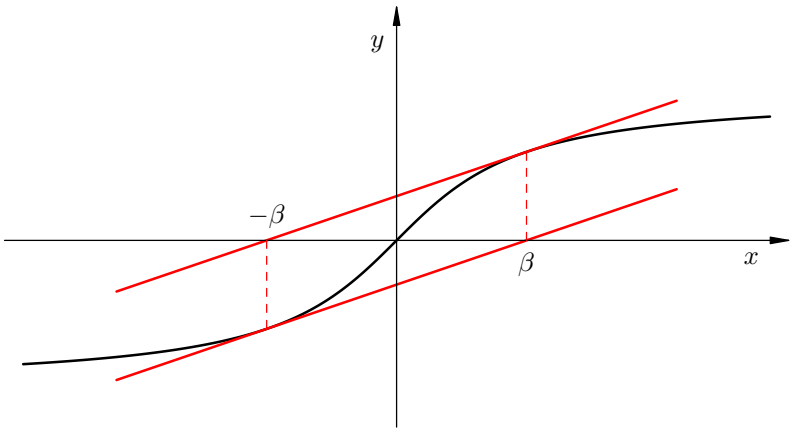
pa će tangenta sijeći os x *u* $-\beta$*, ako je*

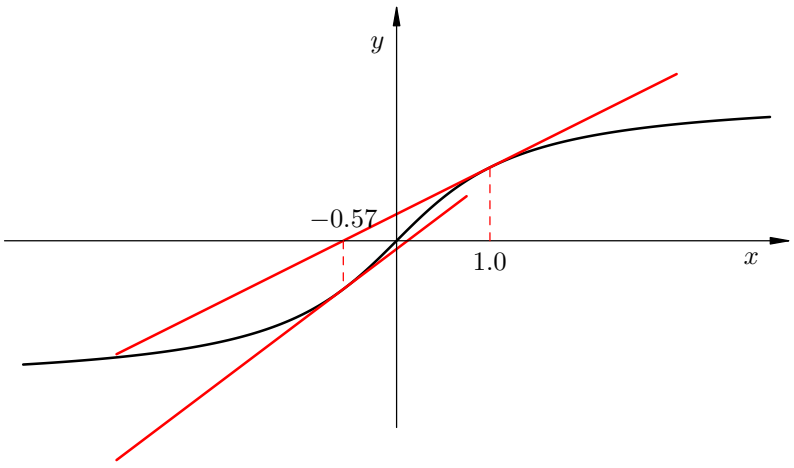
$$\operatorname{arctg}\beta = \frac{2\beta}{1+\beta^2},$$

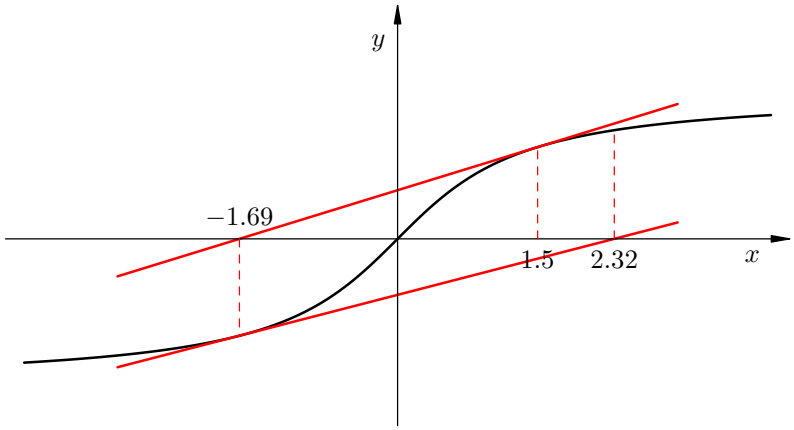
čime smo dobili nelinearnu jednadžbu po β*. Očito, postoje dva rješenja, suprotnih predznaka, i nije ih teško izračunati metodom bisekcije*

$$\beta = \pm 1.39174520027073489.$$

Nacrtajmo grafove Newtonove metode za sve tri mogućnosti za x_0*, recimo za* $x_0 = 1$*,* $x_0 = \beta$ *i* $x_0 = 1.5$*.*






y
−1.69
1.5
2.32
x

9. Numerička integracija

9.1. Općenito o integracijskim formulama

Zadana je funkcija $f : I \to \mathbb{R}$, gdje je I obično interval (može i beskonačan). Želimo izračunati integral funkcije f na intervalu $[a, b]$,

$$I(f) = \int_a^b f(x)\,dx. \tag{9.1.1}$$

Svi znamo da je deriviranje (barem analitički) jednostavan postupak, dok integriranje to nije, pa se integrali analitički u "lijepoj formi" mogu izračunati samo za malen skup funkcija f. Zbog toga, u većini slučajeva ne možemo iskoristiti osnovni teorem integralnog računa, tj. Newton–Leibnitzovu formulu za računanje $I(f)$ preko vrijednosti primitivne funkcije F od f u rubovima intervala

$$I(f) = \int_a^b f(x)\,dx = F(b) - F(a).$$

Drugim riječima, jedino što nam preostaje je približno, numeričko računanje $I(f)$.

Osnovna ideja numeričke integracije je izračunavanje $I(f)$ korištenjem vrijednosti funkcije f na nekom konačnom skupu točaka. Recimo odmah da postoje i integracijske formule koje koriste i derivacije funkcije f, ali o tome kako se one dobivaju i čemu služe, bit će više riječi nešto kasnije.

Opća integracijska formula ima oblik

$$I(f) = I_m(f) + E_m(f),$$

pri čemu je $m+1$ broj korištenih točaka, $I_m(f)$ pripadna aproksimacija integrala, a $E_m(f)$ pritom napravljena greška. Ovakve formule za približnu integraciju funkcija jedne varijable (tj. na jednodimenzionalnoj domeni) često se zovu i **kvadraturne** formule, zbog interpretacije integrala kao površine ispod krivulje.

Ako koristimo samo funkcijske vrijednosti za aproksimaciju integrala, onda aproksimacija $I_m(f)$ ima oblik

$$I_m(f) = \sum_{k=0}^{m} w_k^{(m)} f(x_k^{(m)}), \tag{9.1.2}$$

pri čemu je m neki unaprijed zadani prirodni broj. Koeficijenti $x_k^{(m)}$ zovu se čvorovi integracije, a $w_k^{(m)}$ težinski koeficijenti.

U općem slučaju, za fiksni m, moramo nekako odrediti $2m + 2$ nepoznatih koeficijenata. Uobičajen način njihovog određivanja je zahtjev da su integracijska formule egzaktne na vektorskom prostoru **polinoma** što višeg stupnja. Zašto baš tako? Ako postoji Taylorov red za funkciju f i ako on konvergira, onda bi to značilo da integracijska formula egzaktno integrira početni komad Taylorovog reda, tj. Taylorov polinom. Drugim riječima, greška bi bila mala, tj. jednaka integralu greške koji nastaje kad iz Taylorovog reda napravimo Taylorov polinom.

Zbog linearnosti integrala kao funkcionala

$$\int (\alpha f(x) + \beta g(x))\, dx = \alpha \int f(x)\, dx + \beta \int g(x)\, dx, \tag{9.1.3}$$

dovoljno je gledati egzaktnost tih formula na nekoj bazi vektorskog prostora, recimo na

$$\{1, x, x^2, x^3, \ldots, x^m, \ldots\},$$

jer svojstvo (9.1.3) onda osigurava egzaktnost za sve polinome do najvišeg stupnja baze.

Ako su čvorovi fiksirani, recimo ekvidistantni, onda dobivano tzv. Newton–Cotesove formule, za koje moramo odrediti $m + 1$ nepoznati koeficijent (težine). Uvjeti egzaktnosti na vektorskom prostoru polinoma tada vode na sustav linearnih jednadžbi. Kasnije ćemo pokazati da se te formule mogu dobiti i kao integrali interpolacijskih polinoma stupnja m za funkciju f na zadanoj (ekvidistantnoj) mreži čvorova.

S druge strane, možemo fiksirati samo neke čvorove, ili dozvoliti da su svi čvorovi "slobodni". Ove posljednje formule zovu se formule Gaussovog tipa. U slučaju Gaussovih formula (ali može se i kod težinskih Newton–Cotesovih formula) uobičajeno je (9.1.1) zapisati u obliku

$$I(f) = \int_a^b w(x) f(x)\, dx, \tag{9.1.4}$$

pri čemu je funkcija $w \geq 0$ tzv. težinska funkcija. Ona ima istu ulogu "gustoće" mjere kao i kod metode najmanjih kvadrata. Ideja je "razdvojiti" podintegralnu

funkciju na dva dijela, tako da singulariteti budu uključeni u w. Gaussove se formule nikad ne računaju "direktno" iz uvjeta egzaktnosti, jer to vodi na nelinearni sustav jednadžbi. Pokazat ćemo da postoji veza Gaussovih formula, funkcije w i ortogonalnih polinoma obzirom na funkciju w na intervalu $[a, b]$, koja omogućava efikasno računanje svih parametara za Gaussove formule.

Na kraju ovog uvoda spomenimo još da postoje primjene u kojima je korisno tražiti egzaktnost integracijskih formula na drugačijim sustavima funkcija, koji nisu prostori polinoma do određenog stupnja.

9.2. Newton–Cotesove formule

Newton–Cotesove formule zatvorenog tipa imaju ekvidistantne čvorove, s tim da je prvi čvor u točki $x_0 := a$, a posljednji u $x_m := b$. Preciznije, za zatvorenu (to se često ispušta) Newton–Cotesovu formulu s $(m + 1)$-nom točkom čvorovi su

$$x_k^{(m)} = x_0 + kh_m, \quad k = 0, \ldots, m, \quad h_m = \frac{b-a}{m}.$$

Drugim riječima, osnovni je oblik Newton–Cotesovih formula

$$\int_a^b f(x)\,dx \approx I_m(f) = \sum_{k=0}^{m} w_k^{(m)} f(x_0 + kh_m). \tag{9.2.1}$$

9.2.1. Trapezna formula

Izvedimo najjednostavniju (zatvorenu) Newton–Cotesovu formulu za $m = 1$.

Za $m = 1$, aproksimacija integrala (9.2.1) ima oblik

$$I_1(f) = w_0^{(1)} f(x_0) + w_1^{(1)} f(x_0 + h_1),$$

pri čemu je

$$h := h_1 = \frac{b-a}{1} = b - a,$$

pa je $x_0 = a$ i $x_1 = b$. Da bismo olakšali pisanje, kad znamo da je $m = 1$, možemo izostaviti gornje indekse u $w_k^{(1)}$, tj., radi jednostavnosti, pišemo $w_k := w_k^{(1)}$. Dakle, moramo pronaći težine w_0 i w_1, tako da integracijska formula egzaktno integrira polinome što višeg stupnja na intervalu $[a, b]$, tj. da za polinome f što višeg stupnja bude

$$\int_a^b f(x)\,dx = I_1(f) = w_0 f(a) + w_1 f(b).$$

Stavimo, redom, uvjete na bazu vektorskog prostora polinoma. Ako je f neki od polinoma baze vektorskog prostora, morat ćemo izračunati njegov integral. Zbog toga je zgodno odmah izračunati integrale oblika

$$\int_a^b x^k\,dx, \quad k \geq 0,$$

a zatim rezultat koristiti za razne k. Vrijedi

$$\int_a^b x^k\,dx = \frac{x^{k+1}}{k+1}\Bigg|_a^b = \frac{b^{k+1}-a^{k+1}}{k+1}. \tag{9.2.2}$$

Za $f(x) = 1 = x^0$ dobivamo

$$b - a = \int_a^b x^0\,dx = w_0 \cdot 1 + w_1 \cdot 1.$$

Odmah je očito da iz jedne jednadžbe ne možemo odrediti dva nepoznata parametra, pa moramo zahtjevati da integracijska formula bude egzaktna i na polinomima stupnja 1.

Za $f(x) = x$ izlazi

$$\frac{b^2-a^2}{2} = \int_a^b x\,dx = w_0 \cdot a + w_1 \cdot b.$$

Sada imamo dvije jednadžbe s dvije nepoznanice

$$\begin{aligned} w_0 + w_1 &= b - a \\ aw_0 + bw_1 &= \frac{b^2-a^2}{2}. \end{aligned}$$

Pomnožimo li prvu jednadžbu s $-a$ i dodamo drugoj, dobivamo

$$(b-a)w_1 = \frac{b^2-a^2}{2} - a(b-a) = \frac{b^2-2ab+a^2}{2} = \frac{(b-a)^2}{2}.$$

Budući da je $a \neq b$, dijeljenjem s $b-a$, dobivamo

$$w_1 = \frac{1}{2}(b-a) = \frac{h}{2}.$$

Drugu težinu w_0 lako izračunamo iz prve jednadžbe linearnog sustava

$$w_0 = b - a - w_1 = \frac{1}{2}(b-a) = \frac{h}{2},$$

pa je $w_0 = w_1$.

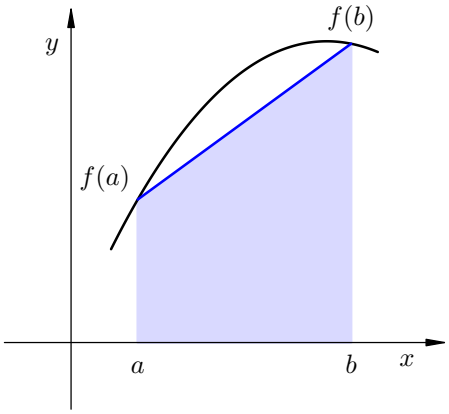
Vidimo da je integracijska formula $I_1(f)$ dobivena iz egzaktnosti na svim polinomima stupnja manjeg ili jednakog 1, i glasi

$$\int_a^b f(x)\,dx \approx \frac{h}{2}\,(f(a) + f(b)).$$

Ta formula zove se trapezna formula. Odakle joj ime? Napišemo li je na malo drugačiji način, kao

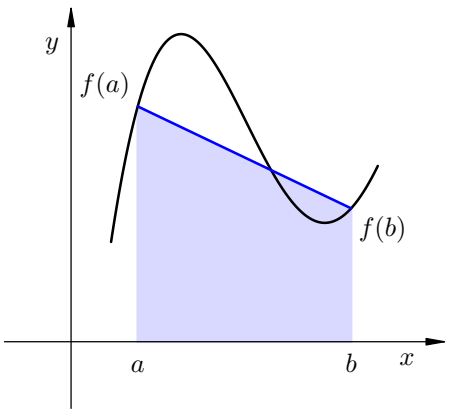
$$\int_a^b f(x)\,dx \approx \frac{f(a) + f(b)}{2}\,(b - a),$$

odmah ćemo vidjeti da je $(f(a) + f(b))/2$ srednjica, a $b - a$ visina trapeza sa slike.



Drugim riječima, površinu ispod krivulje zamijenili smo (tj. aproksimirali) površinom trapeza.

Koliko je ta zamjena dobra? Ovisi o funkciji f. Sve dok pravac razumno aproksimira oblik funkciju f, greška je mala. Na primjer, za funkciju



pravac nije dobra aproksimacija za oblik funkcije f. Da smo nacrtali funkciju f "simetričnije" oko sjecišta, moglo bi se dogoditi da je greška vrlo mala, jer bi se ono što je previše uračunato u površinu s jedne strane "skratilo" s onim što je premalo uračunato s druge strane. S numeričkog stanovišta, takav pristup je opasan.

Trapezna integracijska formula neće egzaktno integrirati sve polinome stupnja 2. To nije teško pokazati, jer već za

$$f(x) = x^2$$

vrijedi

$$\frac{b^3 - a^3}{3} = \int_a^b x^2\,dx \neq I_1(x^2) = \frac{a^2 + b^2}{2}\,(b - a).$$

Slika nas upućuje na još jednu činjenicu. Povučemo li kroz $(a, f(a))$, $(b, f(b))$ linearni interpolacijski polinom, a zatim ga egzaktno integriramo od a do b, dobivamo trapeznu formulu. Pokažimo da je to tako.

Interpolacijski polinom stupnja 1 koji prolazi kroz zadane točke je

$$p_1(x) = f(a) + f[a, b]\,(x - a).$$

Njegov integral na $[a, b]$ je

$$\begin{aligned}\int_a^b p_1(x)\,dx &= \left(f(a)x - a\,f[a, b]x + f[a, b]\,\frac{x^2}{2}\right)\Bigg|_a^b \\ &= (b - a)f(a) + \frac{(b - a)^2}{2}\,f[a, b] = (b - a)\,\frac{f(a) + f(b)}{2}.\end{aligned}$$

Ovaj nam pristup omogućava i ocjenu greške integracijska formule, preko ocjene greške interpolacijskog polinoma, uz uvjet da možemo ocijeniti grešku interpolacijskog polinoma (tj. ako f ima dovoljan broj neprekidnih derivacija).

Neka je funkcija $f \in C^2[a, b]$. Greška interpolacijskog polinoma stupnja 1 koji funkciju f interpolira u točkama $(a, f(a))$, $(b, f(b))$ na intervalu $[a, b]$ jednaka je

$$e_1(x) = f(x) - p_1(x) = \frac{f''(\xi)}{2}\,(x - a)\,(x - b).$$

Drugim riječima, vrijedi

$$E_1(f) = \int_a^b \frac{f''(\xi)}{2}\,(x - a)\,(x - b)\,dx.$$

Ostaje samo izračunati $E_1(f)$. Iskoristit ćemo generalizaciju teorema srednje vrijednosti za integrale. Ako su funkcije g i w integrabilne na $[a,b]$ i ako je $w(x) \geq 0$ na $[a,b]$, a

$$m = \inf_{x\in[a,b]} g(x), \quad M = \sup_{x\in[a,b]} g(x),$$

onda vrijedi

$$m \int_a^b w(x)\,dx \leq \int_a^b w(x)g(x)\,dx \leq M \int_a^b w(x)\,dx.$$

Prethodna formula lako se dokazuje, jer je

$$m \leq g(x) \leq M \implies mw(x) \leq g(x)w(x) \leq Mw(x),$$

pa je

$$m \int_a^b w(x)\,dx \leq \int_a^b w(x)g(x)\,dx \leq M \int_a^b w(x)\,dx. \tag{9.2.3}$$

Digresija za nematematičare. inf (čitati infimum) je minimum funkcije koji se ne mora dostići. Na primjer, funkcija

$$g(x) = x \quad \text{na} \quad (0,1) \tag{9.2.4}$$

nema minimum, ali je

$$\inf_{x\in(0,1)} x = 0.$$

Slično vrijedi i za sup (čitati supremum). Supremum je maksimum funkcije koji se ne mora dostići. Na primjer, funkcija iz relacije (9.2.4) nema ni maksimum, ali je

$$\sup_{x\in(0,1)} x = 1.$$

■

Korištenjem relacije (9.2.3), lako dokazujemo integralni teorem srednje vrijednosti s težinama.

Teorem 9.2.1 *Neka su funkcije g i w integrabilne na $[a,b]$ i neka je*

$$m = \inf_{x\in[a,b]} g(x), \quad M = \sup_{x\in[a,b]} g(x).$$

Nadalje, neka je $w(x) \geq 0$ na $[a,b]$. Tada postoji broj μ, $m \leq \mu \leq M$ takav da vrijedi

$$\int_a^b w(x)g(x)\,dx = \mu \int_a^b w(x)\,dx.$$

Posebno, ako je g neprekidna na $[a,b]$, onda postoji broj ζ takav da je

$$\int_a^b w(x)g(x)\,dx = g(\zeta)\int_a^b w(x)\,dx.$$

Dokaz. Ako je

$$\int_a^b w(x)\,dx = 0,$$

onda je po (9.2.3) i

$$\int_a^b w(x)g(x)\,dx = 0,$$

pa za μ možemo uzeti proizvoljan realan broj. Ako je

$$\int_a^b w(x)\,dx > 0,$$

onda dijeljenjem formule (9.2.3) s prethodnim integralom dobivamo

$$m \le \frac{\int_a^b w(x)g(x)\,dx}{\int_a^b w(x)\,dx} \le M,$$

pa za μ možemo uzeti

$$\mu = \frac{\int_a^b w(x)g(x)\,dx}{\int_a^b w(x)\,dx}.$$

Posljednji zaključak teorema slijedi iz činjenice da neprekidna funkcija na segmentu postiže sve vrijednosti između minimuma i maksimuma, pa mora postići i μ. Drugim riječima, postoji ζ takav da je $\mu = g(\zeta)$. ■

Prisjetite se, već smo pokazali da je

$$E_1(f) = \int_a^b \frac{f''(\xi)}{2}\,(x-a)\,(x-b)\,dx.$$

Primijetite da je funkcija

$$\frac{(x-a)\,(x-b)}{2} \le 0 \quad \text{na} \quad [a,b],$$

pa možemo uzeti

$$w(x) = -\frac{(x-a)\,(x-b)}{2}, \quad g(x) = -f''(\xi).$$

Po generaliziranom teoremu srednje vrijednosti, ako je $f \in C^2[a,b]$, (što znači da je $f'' \in C^0[a,b]$), vrijedi da je

$$E_1(f) = -f''(\zeta) \int_a^b -\frac{(x-a)\,(x-b)}{2}\,dx.$$

Ovaj se integral jednostavno računa. Integriranjem dobivamo

$$\int_a^b \frac{(x-a)\,(x-b)}{2}\,dx = -\frac{(b-a)^3}{12} = -\frac{h^3}{12},$$

pa je

$$E_1(f) = -f''(\zeta)\frac{h^3}{12}.$$

9.2.2. Simpsonova formula

Izvedimo sljedeću (zatvorenu) Newton–Cotesovu formulu za $m = 2$, poznatu pod imenom Simpsonova formula.

Za $m = 2$, aproksimacija integrala (9.2.1) ima oblik

$$I_2(f) = w_0^{(2)} f(x_0) + w_1^{(2)} f(x_0 + h_2) + w_2^{(2)} f(x_0 + 2h_2),$$

pri čemu je

$$h := h_2 = \frac{b-a}{2}.$$

Ponovno, da bismo olakšali pisanje, kad znamo da je $m = 2$, možemo, radi jednostavnosti, izostaviti gornje indekse u $w_k := w_k^{(2)}$. Oprez, to nisu isti w_k i h kao u trapeznoj formuli! Kad uvrstimo značenje h u aproksimacijsku formulu, dobivamo

$$I_2(f) = w_0 f(a) + w_1\, f\Big(\frac{a+b}{2}\Big) + w_2 f(b).$$

Stavimo uvjete na egzaktnost formule na vektorskom prostoru polinoma što višeg stupnja. Moramo postaviti najmanje tri jednadžbe, jer imamo tri nepoznata koeficijenta. Za $f(x) = 1$ dobivamo

$$b - a = \int_a^b x^0\,dx = w_0 \cdot 1 + w_1 \cdot 1 + w_2 \cdot 1.$$

Za $f(x) = x$ izlazi

$$\frac{b^2 - a^2}{2} = \int_a^b x\,dx = w_0 \cdot a + w_1 \frac{a+b}{2} + w_2 \cdot b.$$

Konačno, za $f(x) = x^2$ dobivamo

$$\frac{b^3 - a^3}{3} = \int_a^b x^2\,dx = w_0 \cdot a^2 + w_1 \frac{(a+b)^2}{4} + w_2 \cdot b^2.$$

Sada imamo linearni sustav s tri jednadžbe i tri nepoznanice

$$\begin{aligned} w_0 + w_1 + w_2 &= b - a \\ aw_0 + \frac{a+b}{2} w_1 + bw_2 &= \frac{b^2 - a^2}{2} \\ a^2 w_0 + \frac{(a+b)^2}{4} w_1 + b^2 w_2 &= \frac{b^3 - a^3}{3}. \end{aligned}$$

Rješavanjem ovog sustava, dobivamo

$$w_0 = w_2 = \frac{h}{3} = \frac{b-a}{6}, \quad w_1 = \frac{4h}{3} = \frac{4(b-a)}{6}.$$

Drugim riječima, integracijska formula $I_2(f)$ dobivena je iz egzaktnosti na svim polinomima stupnja manjeg ili jednakog 2, i glasi

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\Big(f(a) + 4f\Big(\frac{a+b}{2}\Big) + f(b)\Big).$$

Simpsonova formula ima još jednu prednost. Iako je dobivena iz uvjeta egzaktnosti na vektorskom prostoru polinoma stupnja manjeg ili jednakog 2, ona egzaktno integrira i sve polinome stupnja 3. Dovoljno je pokazati da egzaktno integrira

$$f(x) = x^3.$$

Egzaktni integral jednak je

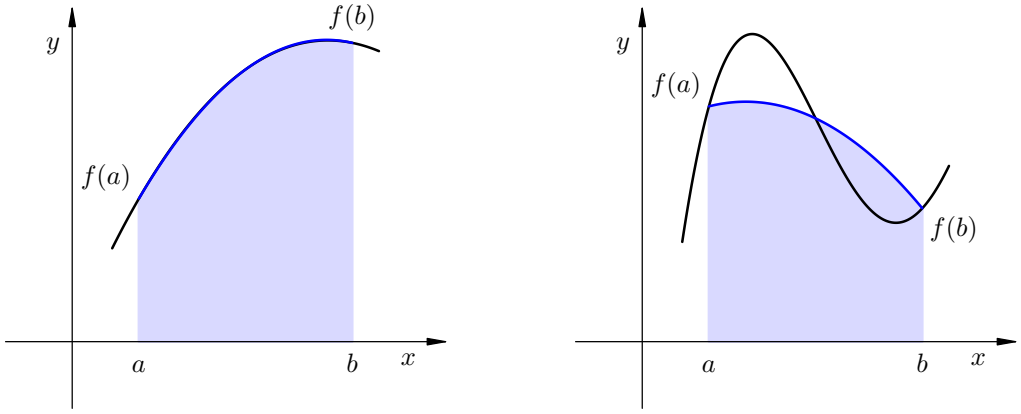
$$\int_a^b x^3\,dx = \frac{b^4 - a^4}{4},$$

a po Simpsonovoj formuli, za $f(x) = x^3$ dobivamo

$$\begin{aligned} I_2(x^3) &= \frac{b-a}{6}\Big(a^3 + 4\Big(\frac{a+b}{2}\Big)^3 + b^3\Big) \\ &= \frac{b-a}{4}\,(a^3 + a^2 b + ab^2 + b^3) = \frac{b^4 - a^4}{4}. \end{aligned}$$

Ponovno, nije teško pokazati da je i ova formula interpolacijska. Ako povučemo kvadratni interpolacijski polinom kroz $(a, f(a))$, $(\frac{a+b}{2}, f(\frac{a+b}{2}))$ i $(b, f(b))$, a zatim ga egzaktno integriramo od a do b, dobivamo Simpsonovu formulu.

Ako pogledamo kako ona funkcionira na funkcijama koje smo već integrirali trapeznom formulom, vidjet ćemo da joj je greška bitno manja. Posebno, na prvom primjeru, kvadratni interpolacijski polinom tako dobro aproksimira funkciju f, da se one na grafu ne razlikuju.



Grešku Simpsonove formule računamo slično kao kod trapezne, integracijom greške kvadratnog interpolacijskog polinoma

$$e_2(x) = f(x) - p_2(x) = \frac{f'''(\xi)}{6}\,(x-a)\Big(x - \frac{a+b}{2}\Big)(x-b).$$

Dakle, za grešku Simpsonove formule vrijedi

$$E_2(f) = \int_a^b e_2(x)\,dx.$$

Nažalost, funkcija

$$(x-a)\Big(x - \frac{a+b}{2}\Big)(x-b)$$

nije više fiksnog znaka na $[a, b]$, pa ne možemo direktno primijeniti generalizirani teorem srednje vrijednosti. Pretpostavimo da je $f \in C^4[a, b]$. Označimo

$$c := \frac{a+b}{2}$$

i definiramo

$$w(x) = \int_a^x (t-a)\,(t-c)\,(t-b)\,dt.$$

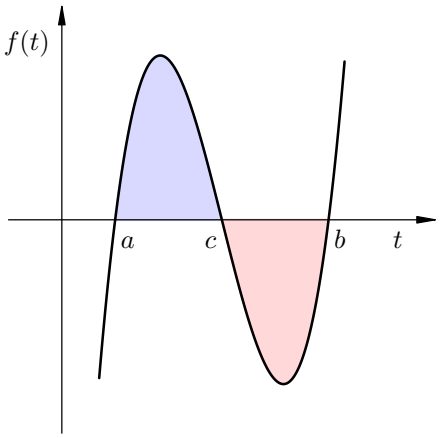
Tvrdimo da vrijedi

$$w(a) = w(b) = 0, \quad w(x) > 0, \quad x \in (a, b). \tag{9.2.5}$$

Skiciramo li funkciju

$$f(t) = (t - a)(t - c)(t - b)$$

odmah vidimo da je ona centralno simetrična oko srednje točke



pa će integral rasti od 0 do svog maksimuma (plava površina), a zatim padati (kad dođe u crveno područje) do 0.

Ostaje samo još napisati grešku interpolacijskog polinoma kao podijeljenju razliku. To smo pokazali općenito u poglavlju o Newtonovom interpolacijskom polinomu, a posebno za $n = 3$ vrijedi

$$f[a, b, c, x] = \frac{f'''(\xi)}{6}.$$

Uz oznaku (9.2.5), grešku Simpsonove formule, onda možemo napisati kao

$$E_2(f) = \int_a^b w'(x) f[a, b, c, x]\, dx.$$

Parcijalnom integracijom ovog integrala dobivamo

$$E_2(f) = w(x) f[a, b, c, x] \Big|_a^b - \int_a^b w(x) \frac{d}{dx} f[a, b, c, x]\, dx.$$

Prvi član je očito jednak 0, jer je $w(a) = w(b) = 0$. Ostaje još "srediti" drugi član. Kod splajnova smo objašnjavali da je podijeljena razlika s dvostrukim čvorom jednaka derivaciji funkcije. Na sličan je način derivacija treće podijeljene razlike

$f[a,b,c,x]$ po x, četvrta podijeljena razlika s dvostrukim čvorom x. Prema tome, dobivamo formulu greške u obliku

$$E_2(f) = -\int_a^b w(x) f[a,b,c,x,x]\,dx.$$

Sad je funkcija w nenegativna i možemo primijeniti generalizirani teorem srednje vrijednosti. Izlazi

$$E_2(f) = -f[a,b,c,\eta,\eta]\int_a^b w(x)\,dx,$$

gdje je $a \le \eta \le b$. Napišemo li $f[a,b,c,\eta,\eta]$ kao derivaciju, dobivamo

$$E_2(f) = -\frac{f^{(4)}(\zeta)}{4!}\int_a^b w(x)\,dx.$$

Ostaje još samo integrirati funkciju w. Vrijedi

$$\begin{aligned} w(x) &= \int_a^x (t-a)\,(t-c)\,(t-b)\,dt = \text{zamjena varijable } y = t-c \\ &= \int_{-h}^{x-c} (y-h)y(y+h)\,dy = \int_{-h}^{x-c} (y^3 - h^2 y)\,dy \\ &= \left(\frac{y^4}{4} - h^2\frac{y^2}{2}\right)\Bigg|_{-h}^{x-c} = \frac{(x-c)^4}{4} - h^2\frac{(x-c)^2}{2} + \frac{h^4}{4}. \end{aligned}$$

Nadalje je

$$\begin{aligned} \int_a^b w(x)\,dx &= \int_a^b \left(\frac{(x-c)^4}{4} - h^2\frac{(x-c)^2}{2} + \frac{h^4}{4}\right)dx = \text{zamjena varijable } y = x-c \\ &= \int_{-h}^{h} \left(\frac{y^4}{4} - h^2\frac{y^2}{2} + \frac{h^4}{4}\right)dy = \left(\frac{y^5}{20} - h^2\frac{y^3}{6} + \frac{h^4 y}{4}\right)\Bigg|_{-h}^{h} \\ &= 2\left(\frac{h^5}{20} - \frac{h^5}{6} + \frac{h^5}{4}\right) = \frac{4}{15}\,h^5. \end{aligned}$$

Kad to uključimo u formulu za grešku, dobivamo

$$E_2(f) = -\frac{f^{(4)}(\zeta)}{24}\cdot\frac{4}{15}\,h^5 = -\frac{h^5}{90}f^{(4)}(\zeta).$$

Primijetite, greška je za red veličine bolja no što bi po upotrijebljenom interpolacijskom polinomu trebala biti.

9.2.3. Produljene formule

Nije teško pokazati da su sve Newton–Cotesove formule integrali interpolacijskih polinoma na ekvidistantnoj mreži. Ako ne valja dizanje stupnjeva interpolacijskih polinoma na ekvidistantnoj mreži, onda neće biti dobri niti njihovi integrali.

Pokažimo to na primjeru Runge. Prava vrijednost integrala je

$$\int_{-5}^{5} \frac{dx}{1+x^2} = 2\,\mathrm{arctg}\,5 \approx 2.74680153389003172.$$

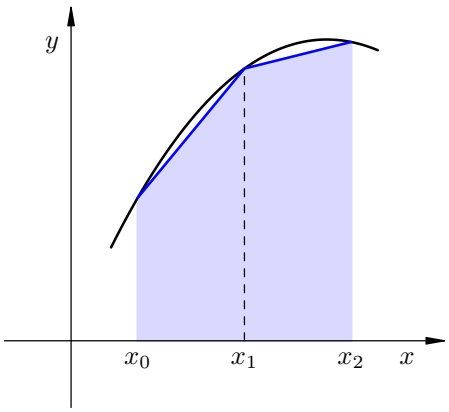
Sljedeća tablica pokazuje aproksimacije integrala izračunate Newton–Cotesovim formulama raznih redova i pripadne greške.

Red formule m	Aproksimacija integrala	Greška
1	0.38461538461538462	2.36218614927464711
2	6.79487179487179487	−4.04807026098176315
3	2.08144796380090498	0.66535357008912674
4	2.37400530503978780	0.37279622885024392
5	2.30769230769230769	0.43910922619772403
6	3.87044867347079978	−1.12364713958076805
7	2.89899440974837875	−0.15219287585834703
8	1.50048890712791179	1.24631262676211993
9	2.39861789784183472	0.34818363604819700
10	4.67330055565349876	−1.92649902176346704
11	3.24477294027858525	−0.49797140638855353
12	−0.31293651575343889	3.05973804964347061
13	1.91979721683238891	0.82700431705764282
14	7.89954464085193082	−5.15274310696189909
15	4.15555899270655713	−1.40875745881652541
16	−6.24143731477308329	8.98823884866311501
17	0.26050944143760372	2.48629209245242800
18	18.87662129010920670	−16.12981975621917490
19	7.24602608588196936	−4.49922455199193763
20	−26.84955208882447960	29.59635362271451140

Očito je da aproksimacije **ne** konvergiraju prema pravoj vrijednosti integrala. Potpunije opravdanje ovog ponašanja dajemo nešto kasnije.

I što sad? Ne smijemo dizati red formula, jer to postaje opasno. Rješenje je vrlo slično onome što smo primijenili kod interpolacije. Umjesto da dižemo red

formule, podijelimo interval $[a, b]$ na više dijelova, recimo, jednake duljine, i na svakom od njih primijenimo odgovarajuću integracijsku formulu niskog reda. Tako dobivene formule zovu se **produljene** formule. Na primjer, za funkciju koju smo već razmatrali, produljena trapezna formula s 2 podintervala izgledala bi ovako.



Općenito, produljenu trapeznu formulu dobivamo tako da cijeli interval $[a, b]$ podijelimo na n podintervala oblika $[x_{k-1}, x_k]$, za $k = 1, \ldots, n$, s tim da je

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b,$$

i na svakom od njih upotrijebimo "običnu" trapeznu formulu. Znamo da je tada

$$\int_a^b f(x)\,dx = \sum_{k=1}^{n} \int_{x_{k-1}}^{x_k} f(x)\,dx,$$

pa na isti način zbrojimo i "obične" trapezne aproksimacije u produljenu trapeznu aproksimaciju.

Najjednostavniji je slučaj kad su točke x_k ekvidistantne, tj. kad je svaki podinterval $[x_{k-1}, x_k]$ iste duljine h. To znači da je

$$x_k = a + kh, \quad k = 0, \ldots, n, \quad h = \frac{b-a}{n}.$$

Aproksimacija produljenom trapeznom formulom je

$$\int_a^b f(x)\,dx = h\Big(\frac{1}{2}f_0 + f_1 + \cdots + f_{n-1} + \frac{1}{2}f_n\Big) + E_n^T(f),$$

pri čemu je $E_n^T(f)$ greška produljene formule. Nju možemo zapisati kao zbroj grešaka osnovnih trapeznih formula na podintervalima

$$E_n^T(f) = \sum_{k=1}^{n} -f''(\zeta_k)\frac{h^3}{12}.$$

Greška ovako napisana nije naročito lijepa i korisna, pa ju je potrebno napisati malo drugačije

$$E_n^T(f) = -\frac{h^3 n}{12}\Big(\frac{1}{n}\sum_{k=1}^{n} f''(\zeta_k)\Big).$$

Izraz u zagradi je aritmetička sredina vrijednosti drugih derivacija u točkama ζ_k. Taj se broj sigurno nalazi između najmanje i najveće vrijednosti druge derivacije funkcije f na intervalu $[a,b]$. Budući da je f'' neprekidna na $[a,b]$, onda je broj u zagradi vrijednost druge derivacije u nekoj točki $\xi \in [a,b]$, pa formulu za grešku možemo pisati kao

$$E_n^T(f) = -\frac{h^3 n}{12} f''(\xi) = -\frac{(b-a)h^2}{12} f''(\xi).$$

Iz ove formule izvodimo važnu ocjenu za broj podintervala potrebnih da se postigne zadana točnost za produljenu trapeznu metodu

$$|E_n^T(f)| \le \frac{(b-a)h^2}{12} M_2 = \frac{(b-a)^3}{12n^2} M_2, \quad M_2 = \max_{x\in[a,b]} |f''(x)|.$$

Želimo li da je $|E_n^T(f)| \le \varepsilon$, onda je dovoljno tražiti da bude

$$\frac{(b-a)^3}{12n^2} M_2 \le \varepsilon,$$

odnosno da je

$$n \ge \sqrt{\frac{(b-a)^3 M_2}{12\varepsilon}}, \quad n \text{ cijeli broj.}$$

Na sličan se način izvodi i produljena Simpsonova formula. Primijetite, osnovna Simpsonova formula ima 3 točke, tj. 2 podintervala, pa produljena formula mora imati, također, paran broj podintervala. Pretpostavimo stoga da je n paran broj. Ograničimo se samo na ekvidistantni slučaj. Onda je ponovno

$$h = \frac{b-a}{n}, \quad x_k = a + kh, \quad k = 0, \ldots, n.$$

Aproksimaciju integrala produljenom Simpsonovom formulom dobivamo iz

$$\int_a^b f(x)\,dx = \sum_{k=1}^{n/2} \int_{x_{2k-2}}^{x_{2k}} f(x)\,dx,$$

tako da na svakom podintervalu $[x_{2k-2}, x_{2k}]$, duljine $2h$, primijenimo običnu Simpsonovu formulu, za $k = 1, \ldots, n/2$. Zbrajanjem izlazi

$$\int_a^b f(x)\,dx = \frac{h}{3}\Big(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 4f_{n-1} + f_n\Big) + E_n^S(f),$$

pri čemu je $E_n^S(f)$ greška produljene formule. Nju možemo zapisati kao zbroj grešaka osnovnih Simpsonovih formula na podintervalima

$$E_n^S(f) = \sum_{k=1}^{n/2} -f^{(4)}(\zeta_k)\frac{h^5}{90}.$$

Opet je grešku korisno napisati malo drugačije

$$E_n^S(f) = -\frac{h^5(n/2)}{90}\Big(\frac{2}{n}\sum_{k=1}^{n/2} f^{(4)}(\zeta_k)\Big).$$

Sličnim zaključivanjem kao kod trapezne formule, izraz u zagradi možemo zamijeniti s $f^{(4)}(\xi)$, $\xi \in [a,b]$, pa dobivamo

$$E_n^S(f) = -\frac{h^5 n}{180}f^{(4)}(\xi) = -\frac{(b-a)h^4}{180}f^{(4)}(\xi).$$

Ponovno, iz ove formule izvodimo ocjenu za broj podintervala potrebnih da se postigne zadana točnost za Simpsonovu metodu

$$|E_n^S(f)| \le \frac{(b-a)h^4}{180}M_4 = \frac{(b-a)^5}{180n^4}M_4, \quad M_4 = \max_{x\in[a,b]}|f^{(4)}(x)|.$$

Želimo li da je $|E_n^S(f)| \le \varepsilon$, onda je dovoljno tražiti da bude

$$\frac{(b-a)^5}{180n^4}M_4 \le \varepsilon,$$

odnosno da je

$$n \ge \sqrt[4]{\frac{(b-a)^5 M_4}{180\varepsilon}}, \quad n \text{ paran cijeli broj.}$$

9.2.4. Primjeri

Primjer 9.2.1 *Izračunajte vrijednost integrala*

$$\int_1^2 xe^{-x}\,dx$$

korištenjem (produljene) Simpsonove formule tako da greška bude manja ili jednaka 10^{-6}. *Nađite pravu vrijednost integrala i pogreške. Koliko je podintervala potrebno za istu točnost korištenjem (produljene) trapezne formule?*

Prmo, moramo ocijeniti pogrešku za produljenu trapeznu i produljenu Simpsonovu formulu. Za to su nam potrebni maksimumi apsolutnih vrijednosti druge i četvrte derivacije na zadanom intervalu. Derivacije su redom

$$f^{(1)}(x) = (1-x)e^{-x}, \quad f^{(2)}(x) = (x-2)e^{-x}, \quad f^{(3)}(x) = (3-x)e^{-x},$$
$$f^{(4)}(x) = (x-4)e^{-x}, \quad f^{(5)}(x) = (5-x)e^{-x}.$$

Nađimo maksimume apsolutnih vrijednosti derivacija na zadanom intervalu.

Prvo ocijenimo grešku za produljenu trapeznu formulu. Na intervalu $[1,2]$ *je* $f^{(3)}(x) > 0$*, što znači da* $f^{(2)}$ *raste. Uočimo još da je na zadanom intervalu* $f^{(2)}(x) \leq 0$*, pa je maksimum apsolutne vrijednosti druge derivacije u lijevom rubu, tj.*

$$M_2 = \max_{x\in[1,2]} |f^{(2)}(x)| = |f^{(2)}(1)| = e^{-1} \approx 0.367879441171.$$

Broj podintervala n_T *za produljenu trapeznu formulu je*

$$n_T \geq \sqrt{\frac{(b-a)^3 M_2}{12\varepsilon}} = \sqrt{\frac{e^{-1}}{12\cdot 10^{-6}}} \approx 175.09,$$

pa je najmanji broj podintervala $n_T = 176$*.*

Sada ocijenimo grešku za produljenu Simpsonovu formulu. Na intervalu $[1,2]$ *je* $f^{(5)}(x) > 0$*, što znači da* $f^{(4)}$ *raste. Također je i* $f^{(4)}(x) < 0$*, što znači da je njen maksimum po apsolutnoj vrijednosti ponovno u lijevom rubu, tj.*

$$M_4 = \max_{x\in[1,2]} |f^{(4)}(x)| = |f^{(4)}(1)| = 3\cdot e^{-1} \approx 1.103638323514.$$

Za grešku produljene Simpsonove formule imamo

$$n_S \geq \sqrt[4]{\frac{(b-a)^5 M_4}{180\varepsilon}} = \sqrt[4]{\frac{3\cdot e^{-1}}{180\cdot 10^{-6}}} \approx 8.85,$$

tj. treba najmanje $n_S = 10$ *podintervala.*

Sad možemo upotrijebiti produljenu Simpsonovu formulu s 10 *podintervala (*11

čvorova). Imamo

k	x_k	$f(x_k)$
0	1.0	0.3678794412
1	1.1	0.3661581921
2	1.2	0.3614330543
3	1.3	0.3542913309
4	1.4	0.3452357495
5	1.5	0.3346952402
6	1.6	0.3230344288
7	1.7	0.3105619909
8	1.8	0.2975379988
9	1.9	0.2841803765
10	2.0	0.2706705665

Sada je

$$\begin{aligned}
&S_0 = f(x_0) + f(x_{10}) = 0.63855000765,\\
&S_1 = 4(f(x_1) + f(x_3) = f(x_5) + f(x_7) + f(x_9)) = 6.5995485226,\\
&S_2 = 2(f(x_2) + f(x_4) + f(x_6) + f(x_8)) = 2.6544824628.
\end{aligned}$$

Vrijednost integrala po Simpsonovoj formuli je

$$I_s = \frac{0.1}{3}(S_0 + S_1 + S_2) = 0.3297526998.$$

U ovom konkretnom slučaju možemo bez puno napora izračunati i egzaktnu vrijednost integrala. Jedina korist od toga je da vidimo koliko je zaista ocjena za Simpsonovu metodu bliska sa stvarnom greškom. Parcijalna integracija daje

$$\begin{aligned}
\int_1^2 xe^{-x}\,dx &= \left\{ \begin{array}{ll} u = x, & du = dx \\ dv = e^{-x}\,dx, & v = -e^{-x} \end{array} \right\} = -xe^{-x}\Big|_1^2 + \int_1^2 e^{-x}\,dx \\
&= e^{-1} - 2e^{-2} - e^{-x}\Big|_1^2 = e^{-1} - 2e^{-2} - e^{-2} + e^{-1} \\
&= 2e^{-1} - 3e^{-2} \approx 0.3297530326.
\end{aligned}$$

Drugim riječima, prava pogreška je

$$I - I_S = 0.3297530326 - 0.3297526998 = 3.328 \cdot 10^{-7},$$

tj. ocjena greške nije daleko od prave pogreške.

9.2.5. Formula srednje točke (midpoint formula)

Ako u Newton–Cotesovim formulama ne interpoliramo (pa onda niti ne integriramo) jednu ili obje rubne točke, dobili smo otvorene Newton–Cotesove formule. Ako definiramo $x_{-1} := a$, $x_{m+1} := b$ i

$$h_m = \frac{b-a}{m+2},$$

onda otvorene Newton–Cotesove formule imaju oblik

$$\int_a^b f(x)\,dx \approx I_m(f) = \sum_{k=0}^{m} w_k^{(m)} f(x_0 + kh_m). \tag{9.2.6}$$

Vjerojatno najkorištenija i najpoznatija otvorena Newton–Cotesova formula je ona najjednostavnija za $m = 0$, poznata pod imenom "midpoint formula" (formula srednje točke).

Dakle za bismo odredili midpoint formulu, moramo naći koeficijent $w_0 := w_0^{(0)}$ takav da je

$$\int_a^b f(x)\,dx = w_0 f\Big(\frac{a+b}{2}\Big)$$

egzaktna na vektorskom prostoru polinoma što višeg stupnja.

Za $f(x) = 1$, imamo

$$b - a = \int_a^b 1\,dx = w_0,$$

odakle odmah slijedi da je

$$\int_a^b f(x)\,dx = (b-a) f\Big(\frac{a+b}{2}\Big).$$

Greška te integracijske formule je integral greške interpolacijskog polinoma stupnja 0 (konstante), koji interpolira funkciju f u srednjoj točki. Ako definiramo

$$w(x) = \int_a^x (t-c)\,dt, \quad c := \frac{a+b}{2},$$

onda koristeći istu tehniku kao kod izvoda greške za Simpsonovu formulu, izlazi da je greška midpoint formule

$$E_0(f) = \int_a^b e_0(x)\,dx = f''(\xi)\frac{(b-a)^3}{24}.$$

Da bismo izveli produljenu formulu, podijelimo interval $[a, b]$ na n podintervala i na svakom upotrijebimo midpoint formulu. Tada vrijedi

$$I_n(f) = h(f_1 + \cdots + f_n) + E_n^M(f), \quad h = \frac{b-a}{n}, \quad x_k = a + \Big(k - \frac{1}{2}\Big)h,$$

pri čemu je $E_n^M(f)$ ukupna greška koja je jednaka

$$E_n^M(f) = \sum_{k=1}^{n} f''(\xi_k)\frac{h^3}{24} = \frac{h^3 n}{24}\Big(\frac{1}{n}\sum_{k=1}^{n} f''(\xi_k)\Big) = \frac{h^3 n}{24} f''(\xi) = \frac{h^2(b-a)}{24} f''(\xi).$$

9.3. Rombergov algoritam

Pri izvodu Rombergovog algoritma koristimo se sljedećim principima:

- udvostručavanjem broja podintervala u produljenoj trapeznoj metodi,
- eliminacijom člana greške iz dvije susjedne produljene formule. Ponovljena primjena ovog principa zove se Richardsonova ekstrapolacija.

Asimptotski razvoj ocjene pogreške za trapeznu integraciju daje Euler–Mac-Laurinova formula.

Teorem 9.3.1 *Neka je $m \geq 0$, $n \geq 1$, m, n cijeli brojevi. Definiramo ekvidistantnu mrežu s n podintervala na $[a, b]$, tj.*

$$h = \frac{b-a}{n}, \quad x_k = a + kh, \quad k = 0, \ldots, n.$$

Pretpostavimo da je $f \in C^{(2m+2)}[a, b]$. Za pogrešku produljene trapezne metode vrijedi

$$E_n(f) = \int_a^b f(x)\,dx - I_n^T(f) = \sum_{i=1}^{m} \frac{d_{2i}}{n^{2i}} + F_{n,m},$$

gdje su koeficijenti

$$d_{2i} = -\frac{B_{2i}}{(2i)!}\,(b-a)^{2i}\,(f^{(2i-1)}(b) - f^{(2i-1)}(a)),$$

a ostatak je

$$F_{n,m} = \frac{(b-a)^{2m+2}}{(2m+2)!n^{2m+2}} \cdot \int_a^b \overline{B}_{2m+2}\Big(\frac{b-a}{h}\Big)\, f^{(2m+2)}(x)\,dx.$$

Ovdje su B_{2i} Bernoullijevi brojevi,

$$B_i = -\int_0^1 B_i(x)\,dx, \quad i \geq 1,$$

a $\overline{B}_i$ je periodičko proširenje običnih Bernoullijevih polinoma

$$\overline{B}_i(x) = \begin{cases} B_i(x), & za\ 0 \leq x \leq 1, \\ \overline{B}_i(x-1), & za\ x \geq 1. \end{cases}$$

Ovo je jedan od klasičnih teorema numeričke analize i njegov se dokaz može naći u mnogim knjigama.

Umjesto dokaza, nekoliko objašnjenja. Bernoullijevi polinomi zadani su implicitno funkcijom izvodnicom

$$\frac{t(e^{xt}-1)}{e^t-1} = \sum_{i=1}^{\infty} B_i(x)\frac{t^i}{i!}.$$

Prvih nekoliko Bernullijevih polinoma su:

$$B_0(x) = 1 \qquad B_1(x) = x \qquad B_2(x) = x^2 - x$$
$$B_3(x) = x^2 - \frac{3x^2}{2} + \frac{x}{2} \qquad B_4(x) = x^2(1-x)^2.$$

Uvijek vrijedi $B_i(0) = 0$ za $i \geq 1$. Rekurzivne relacije su

$$B_i'(x) = \begin{cases} iB_{i-1}(x), & \text{za } i \text{ paran i } i \geq 4, \\ i(B_{i-1}(x) + B_{i-1}), & \text{za } i \text{ neparan i } i \geq 3. \end{cases}$$

Iz prethodne se formule integracijom mogu dobiti $B_i(x)$, jer je slobodni član jednak 0.

Bernoullijevi brojevi također su definirani implicitno

$$\frac{t}{e^t-1} = \sum_{i=0}^{\infty} B_i\,\frac{t^i}{i!},$$

odakle se integracijom na $[0,1]$ po x u rekurziji za $B_i(x)$ dobiva

$$B_i = -\int_0^1 B_i(x)\,dx, \quad i \geq 1.$$

Prvih nekoliko Bernoullijevih brojeva:

$$B_0 = 1, \qquad B_1 = -\frac{1}{2}, \qquad B_2 = -\frac{1}{6}, \qquad B_4 = -\frac{1}{30}, \qquad B_6 = \frac{1}{42},$$
$$B_8 = \frac{1}{30}, \quad B_{10} = \frac{5}{66}, \quad B_{12} = -\frac{691}{2730}, \quad B_{14} = \frac{7}{6}, \qquad B_{14} = -\frac{3617}{510}$$

i dalje vrlo brzo rastu po apsolutnoj vrijednosti

$$B_{60} \approx -2.139994926 \cdot 10^{34}.$$

Rombergov algoritam dobivamo tako da eliminiramo član po član iz reda za ocjenu greške na osnovu vrijednosti integrala s duljinom koraka h i $h/2$.

Za podintegralne funkcije koje nisu dovoljno glatke, također, se može (uz blage pretpostavke) asimptotski dobiti razvoj pogreške. Posebno to vrijedi za funkcije s algebarskim (x^α) i/ili logaritamskim ($\ln x$) singularitetima.

Izvedimo sad Rombergov algoritam. Označimo s $I_n^{(0)}$ trapeznu formulu s duljinom intervala $h = (b-a)/n$. Iz Euler–MacLaurinove formule, ako je n paran, za asimptotski razvoj greške imamo

$$\begin{aligned} I - I_n^{(0)} &= \frac{d_2^{(0)}}{n^2} + \frac{d_4^{(0)}}{n^4} + \cdots + \frac{d_{2m}^{(0)}}{n^{2m}} + F_{n,m} \\ I - I_{n/2}^{(0)} &= \frac{4d_2^{(0)}}{n^2} + \frac{16d_4^{(0)}}{n^4} + \cdots + \frac{2^{2m}d_{2m}^{(0)}}{n^{2m}} + F_{n/2,m}. \end{aligned}$$

Ako prvi razvoj pomnožimo s 4 i oduzmemo mu drugi razvoj, skratit će se prva greška s desne strane $d_2^{(0)}$, tj. dobit ćemo

$$4(I - I_n^{(0)}) - (I - I_{n/2}^{(0)}) = \frac{-12d_4^{(0)}}{n^4} - \frac{60d_6^{(0)}}{n^6} + \cdots.$$

Izlučivanjem članova koji imaju I na lijevu stranu, a zatim dijeljenjem, dobivamo

$$I = \frac{4I_n^{(0)} - I_{n/2}^{(0)}}{3} - \frac{-4d_4^{(0)}}{n^4} - \frac{20d_6^{(0)}}{n^6} + \cdots.$$

Prvi član zdesna možemo uzeti kao bolju, popravljenu aproksimaciju integrala, u oznaci

$$I_n^{(1)} = \frac{4I_n^{(0)} - I_{n/2}^{(0)}}{3}, \quad n \text{ paran}, \, n \geq 2.$$

Niz $I_n^{(2)}$, $I_n^{(4)}$, $I_n^{(6)}$ je novi integracijski niz. Njegova je greška

$$I - I_n^{(1)} = \frac{d_4^{(1)}}{n^4} + \frac{d_6^{(1)}}{n^6} + \cdots,$$

gdje je

$$d_4^{(1)} = -4d_4^{(0)}, \quad d_6^{(1)} = -20d_6^{(0)}.$$

Nađimo eksplicitnu formulu za $I_n^{(1)}$. Zbog podjele na odgovarajući broj podintervala, ako je h duljina podintervala za $I_n^{(0)}$, onda je $h_1 := 2h$ duljina podintervala

za $I^{(0)}_{n/2}$, pa vrijede sljedeće formule

$$I^{(0)}_n = \frac{h}{2}(f_0 + 2f_1 + \cdots 2f_{n-1} + f_n)$$
$$I^{(0)}_{n/2} = \frac{h_1}{2}(f_0 + 2f_2 + \cdots 2f_{n-2} + f_n).$$

Uvrštavanjem u $I^{(1)}_n$, dobivamo

$$\begin{aligned} I^{(1)}_n &= \frac{4h}{3}\Big(\frac{1}{2}f_0 + 2f_1 + \cdots 2f_{n-1} + \frac{1}{2}f_n\Big) - \frac{2h}{3}\Big(\frac{1}{2}f_0 + 2f_1 + \cdots 2f_{n-1} + \frac{1}{2}f_n\Big) \\ &= \frac{h}{3}(f_0 + 4f_2 + 2f_2 + \cdots 4f_{n-2} + f_n), \end{aligned}$$

što je Simpsonova formula s n podintervala.

Sličan argument kao i prije možemo upotrijebiti i dalje. Vrijedi

$$I - I^{(1)}_{n/2} = \frac{16d^{(1)}_4}{n^4} + \frac{64d^{(1)}_6}{n^6} + \cdots.$$

Tada je

$$16(I - I^{(1)}_n) - (I - I^{(1)}_{n/2}) = \frac{-48d^{(1)}_6}{n^6} + \cdots,$$

odnosno

$$I = \frac{16I^{(1)}_n - I^{(1)}_{n/2}}{15} - \frac{-48d^{(1)}_6}{15n^6} + \cdots.$$

Ponovno, prvi član s desne strane proglasimo za novu aproksimaciju integrala

$$I^{(2)}_n = \frac{16I^{(1)}_n - I^{(1)}_{n/2}}{15}, \quad n \text{ djeljiv s } 4, \quad n \geq 4.$$

Induktivno, ako nastavimo postupak, dobivamo Richardsonovu ekstrapolaciju

$$I^{(k)}_n = \frac{4^k I^{(k-1)}_n - I^{(k-1)}_{n/2}}{4^k - 1}, \quad n \geq 2^k,$$

pri čemu je greška jednaka

$$E^{(k)}_n = I - I^{(k)}_n = \frac{d^{(k)}_{2k+2}}{n^{2k+2}} + \cdots = \beta_k(b-a)h^{2k+2}f^{(2k+2)}(\xi), \quad a \leq \xi \leq b.$$

Sada možemo definirati Rombergovu tablicu

$$\begin{array}{cccc} I^{(0)}_1 & & & \\ I^{(0)}_2 & I^{(1)}_2 & & \\ I^{(0)}_4 & I^{(1)}_4 & I^{(2)}_4 & \\ \vdots & \vdots & \vdots & \ddots \end{array}.$$

Ako pogledamo omjere grešaka članova u stupcu, uz pretpostavku dovoljne glatkoće, onda dobivamo

$$\frac{E_n^{(k)}}{E_{2n}^{(k)}} = 2^{2k+2},$$

tj. omjeri pogrešaka u stupcu se moraju ponašati kao

$$\begin{array}{ccccc} 1 & & & & \\ 4 & 1 & & & \\ 4 & 16 & 1 & & \\ 4 & 16 & 64 & 1 & \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array}.$$

Pokažimo na primjeru da prethodni omjeri pogrešaka u stupcu vrijede samo ako je funkcija dovoljno glatka.

Primjer 9.3.1 *Rombergovim algoritmom s točnošću* 10^{-12} *nađite vrijednosti integrala*

$$\int_0^1 e^x\,dx, \quad \int_0^1 x^{3/2}\,dx, \quad \int_0^1 \sqrt{x}\,dx$$

i pokažite kako se ponašaju omjeri pogrešaka u stupcima.

Pogledajmo redom funkcije. Eksponencijalna funkcija ima beskonačno mnogo neprekidnih derivacija, pa bi se računanje integrala morala ponašati po predviđanju. Kao vrijednost, nakon 2^5 *podintervala u trapeznoj formuli, dobivamo umjesto prave vrijednosti integrala* I*, približnu vrijednost*

$$\begin{aligned} I_5 &= 1.71828182845904524 \\ I &= e - 1 = 1.71828182845904524 \\ I - I_5 &= 0. \end{aligned}$$

Pokažimo omjere pogrešaka u stupcima,

```
0 1.0000
1 3.9512  1.0000
2 3.9875 15.6517  1.0000
3 3.9969 15.9913 62.4639   1.0000
4 3.9992 15.9777 63.6087 249.7197    1.0000
5 3.9998 15.9944 63.9017 254.4010 1000.5738 1.0000
```

a zatim samo eksponente omjera pogrešaka (eksponenti od 2*, koji bi ako je funkcija glatka morali biti* $2k+2$*).*

$$
\begin{array}{llllll}
0 & 1.0000 & & & & \\
1 & 1.9823 & 1.0000 & & & \\
2 & 1.9955 & 3.9682 & 1.0000 & & \\
3 & 1.9989 & 3.9920 & 5.9650 & 1.0000 & \\
4 & 1.9997 & 3.9980 & 5.9912 & 7.9642 & 1.0000 \\
5 & 1.9999 & 3.9995 & 5.9978 & 7.9910 & 9.9666 \quad 1.0000
\end{array}
$$

Što je s drugom funkcijom? Funkciji $f(x) = x^{3/2}$ *puca druga derivacija u* 0*, pa bi zanimljivo ponašanje moralo početi veću drugom stupcu (za trapez je funkcija dovoljno glatka za ocjenu pogreške). Kao vrijednost, nakon* 2^{15} *podintervala u trapeznoj formuli, dobivamo umjesto prave vrijednosti integrala* I*, približnu vrijednost*

$$
\begin{aligned}
I_{15} &= 0.40000000000004512 \\
I &= 2/5 = 0.40000000000000000 \\
I - I_{15} &= -0.00000000000004512.
\end{aligned}
$$

Primijetite da je broj intervala poprilično velik! Što je s omjerima pogrešaka?

$$
\begin{array}{llllllll}
0 & 1.0000 & & & & & & \\
1 & 3.7346 & 1.0000 & & & & & \\
2 & 3.8154 & 5.4847 & 1.0000 & & & & \\
3 & 3.8721 & 5.5912 & 5.6484 & 1.0000 & & & \\
4 & 3.9112 & 5.6331 & 5.6559 & 5.6566 & 1.0000 & & \\
5 & 3.9381 & 5.6484 & 5.6568 & 5.6568 & 5.6569 & 1.0000 & \\
6 & 3.9567 & 5.6539 & 5.6568 & 5.6569 & \cdots & 5.6569 & 1.0000 \\
\vdots & \vdots & \vdots & & & & \ddots & \ddots \\
15 & 3.9981 & 5.6569 & \cdots & & & \cdots & 5.6569 \quad 1.0000
\end{array}
$$

Primjećujemo da su se nakon prvog stupca omjeri pogrešaka stabilizirali. Bit će nam mnogo lakše provjeriti što se događa ako napišemo samo eksponente omjera pogrešaka.

$$
\begin{array}{llllllll}
0 & 1.0000 & & & & & & \\
1 & 1.9010 & 1.0000 & & & & & \\
2 & 1.9318 & 2.4554 & 1.0000 & & & & \\
3 & 1.9531 & 2.4832 & 2.4978 & 1.0000 & & & \\
4 & 1.9676 & 2.4939 & 2.5000 & 2.4999 & 1.0000 & & \\
5 & 1.9775 & 2.4978 & 2.5000 & 2.5000 & 2.5000 & 1.0000 & \\
6 & 1.9843 & 2.4992 & 2.5000 & 2.5000 & 2.5000 & 2.5000 & 1.0000 \\
\vdots & \vdots & \vdots & & & & \ddots & \ddots \\
15 & 1.9993 & 2.5000 & \cdots & & & \cdots & 2.5000 \quad 1.0000
\end{array}
$$

Primijetite da su eksponenti omjera pogrešaka od drugog stupca nadalje točno za 1 *veći od eksponenta same funkcije (integriramo!).*

Situacija s funkcijom $f(x) = \sqrt{x}$ *mora biti još gora, jer njoj puca prva derivacija u* 0. *Nakon* 2^{15} *podintervala u trapeznoj formuli (što je ograničenje zbog veličine polja u programu), ne dobivamo željenu točnost*

$$\begin{aligned} I_{15} &= 0.66666665510837633 \\ I &= 2/3 = 0.66666666666666667 \\ I - I_{15} &= 0.00000001155829033. \end{aligned}$$

Omjeri pogrešaka u tablici su:

$$\begin{array}{cccccccc}
0 & 1.0000 & & & & & & \\
1 & 2.6408 & 1.0000 & & & & & \\
2 & 2.6990 & 2.8200 & 1.0000 & & & & \\
3 & 2.7393 & 2.8267 & 2.8281 & 1.0000 & & & \\
4 & 2.7667 & 2.8281 & 2.8284 & 2.8284 & 1.0000 & & \\
5 & 2.7854 & 2.8284 & \cdots & \cdots & 2.8284 & 1.0000 & \\
\vdots & \vdots & \vdots & & & & \ddots & \ddots \\
15 & 2.8271 & 2.8284 & \cdots & & & \cdots & 2.8284 \quad 1.0000
\end{array}$$

Pripadni eksponenti su

$$\begin{array}{cccccccc}
0 & 1.0000 & & & & & & \\
1 & 1.4010 & 1.0000 & & & & & \\
2 & 1.4324 & 1.4957 & 1.0000 & & & & \\
3 & 1.4538 & 1.4991 & 1.4998 & 1.0000 & & & \\
4 & 1.4681 & 1.4998 & 1.5000 & 1.5000 & 1.0000 & & \\
5 & 1.4779 & 1.5000 & \cdots & \cdots & 1.5000 & 1.0000 & \\
\vdots & \vdots & \vdots & & & & \ddots & \ddots \\
15 & 1.4993 & 1.5000 & \cdots & & & \cdots & 1.5000 \quad 1.0000
\end{array}$$

Ipak, u ova dva jednostavna primjera, može se Rombergovom algoritmu "pomoći" tako da supstitucijom u integralu dobijemo glatku funkciju. U oba slučaja, ako stavimo supstituciju $x = t^2$, *podintegralna će funkcija imati beskonačno mnogo neprekidnih derivacija, pa će se algoritam ponašati po ocjeni pogreške.*

U literaturi postoji i malo drugačija oznaka za aproksimacije integrala u Rombergovoj tablici

$$T_m^{(k)} = \frac{4^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{4^m - 1}.$$

Sama tablica ima oblik

$$\begin{array}{llll} T_0^{(0)} & & & \\ T_0^{(1)} & T_1^{(0)} & & \\ T_0^{(2)} & T_1^{(1)} & T_2^{(0)} & \\ \vdots & \vdots & \vdots & \ddots \end{array}.$$

Pokažimo sad nekoliko primjera kako treba, odnosno ne treba koristiti Rombergov algoritam.

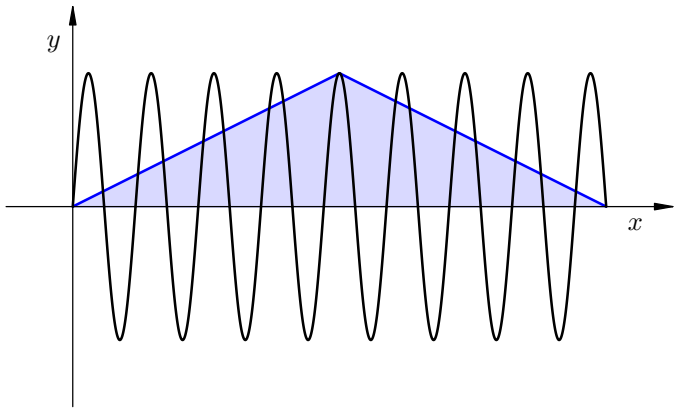
Primjer 9.3.2 *Izračunajte korištenjem Rombergovog algoritma približnu vrijednost integrala*

$$\int_0^1 \sin(17\pi x)\,dx$$

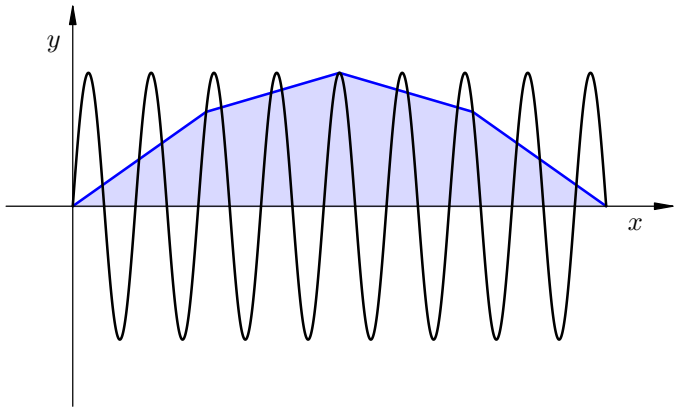
Tako da greška bude manja ili jednaka 10^{-4}*. Napišimo tablicu (samo prvih par decimala, ostale pamtimo u računalu, ali nemamo prostora za ispis)*

0	0.00000							
1	0.50000	0.66667						
2	0.60355	0.63807	0.63616					
3	0.62841	0.63671	0.63661	0.63662				
4	−0.00616	−0.21768	−0.27464	−0.28910	−0.29273			
5	0.02832	0.03982	0.05698	0.06225	0.06362	0.06397		
6	0.03525	0.03756	0.03741	0.03710	0.03700	0.03697	0.03697	
7	0.03690	0.03745	0.03745	0.03745	0.03745	0.03745	0.03745	0.03745

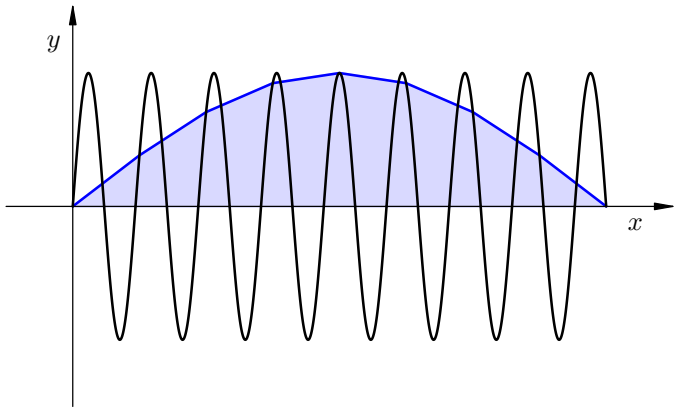
Što je razlog stabilizacije oko jedne, pa oko druge vrijednosti? Nedovoljan broj podintervala u trapezu, koji ne opisuju dobro ponašanje funkcije.



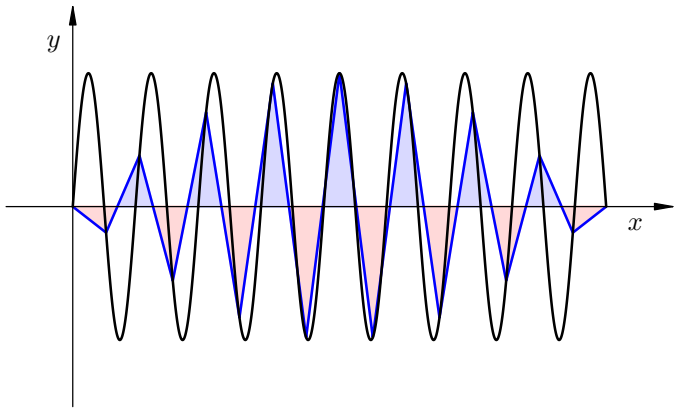
Produljena trapezna formula s 2 *podintervala.*



Produljena trapezna formula s 4 *podintervala.*



Produljena trapezna formula s 8 *podintervala.*



Produljena trapezna formula sa 16 *podintervala.*

9.4. Težinske integracijske formule

Dosad smo detaljno analizirali samo nekoliko osnovnih Newton–Cotesovih integracijskih formula s malim brojem točaka i pripadne produljene formule. U ovom odjeljku napravit ćemo opću konstrukciju i analizu točnosti za neke klase integracijskih formula, uključujući opće Newton–Cotesove i Gaussove formule.

Želimo (približno) izračunati vrijednost integrala

$$I_w(f) = \int_a^b f(x)w(x)\,dx, \tag{9.4.1}$$

gdje je w pozitivna (ili barem nenegativna) "težinska" funkcija za koju pretpostavljamo da je integrabilna na (a, b), s tim da dozvoljavamo da w nije definirana u rubovima a i b. Interval integracije može biti konačan, ali i beskonačan. Drugim riječima, promatramo opći problem jednodimenzionalne integracije zadane funkcije f po zadanoj neprekidnoj mjeri $d\lambda$ generiranoj težinskom funkcijom w na zadanoj domeni. Katkad koristimo i skraćenu oznaku $I(f)$, umjesto $I_w(f)$, za integral u (9.4.1), ako je $w(x) = 1$ na cijelom $[a, b]$, ili kad je težinska funkcija jasna iz konteksta, da skratimo pisanje.

Kao i ranije, ovaj integral aproksimiramo "težinskom" sumom funkcijskih vrijednosti funkcije f na konačnom skupu točaka, Za razliku od ranijih oznaka, ovdje je zgodnije točke numerirati od 1, a ne od 0. Dakle, opća težinska integracijska ili kvadraturna formula za aproksimaciju integrala $I_w(f)$ ima oblik

$$I_n(f) = \sum_{k=1}^{n} w_k^{(n)} f(x_k^{(n)}), \tag{9.4.2}$$

gdje je n prirodni broj. Kao i prije, gornje indekse (n) za čvorove i težine često ne pišemo, ako su očiti iz konteksta, ali ne treba zaboraviti na ovisnost o n.

Dakle, sasvim općenito možemo pisati

$$I_w(f) = \int_a^b f(x)w(x)\,dx = I_n(f) + E_n(f), \tag{9.4.3}$$

gdje je $E_n(f)$ greška aproksimacije.

Osnovnu podlogu za konstrukciju integracijskih formula i ocjenu greške $E_n(f)$ daje sljedeći rezultat.

Teorem 9.4.1 *Ako je $I_w(f)$ iz* (9.4.1) *Riemannov integral, i ako je $\widehat{f}$ bilo koja druga funkcija za koju postoji $I_w(\widehat{f})$, onda vrijedi ocjena*

$$|I_w(f) - I_w(\widehat{f})| \le \|w\|_1 \|f - \widehat{f}\|_\infty, \tag{9.4.4}$$

i postoji funkcija $\widehat{f}$ za koju se ova ocjena dostiže.

Dokaz. Prvo uočimo da w ne mora biti nenegativna, jer je riječ o Riemannovom integralu, ali zato treba pretpostaviti da je $|w|$ integrabilna.

Ocjena izlazi direktno iz osnovnih svojstava Riemannovog integrala jer podintegralne funkcije moraju biti ograničene. Dobivamo

$$\begin{aligned} |I_w(f) - I_w(\widehat{f})| &= \left| \int_a^b f(x)w(x)\,dx - \int_a^b \widehat{f}(x)w(x)\,dx \right| \\ &\leq \int_a^b |w(x)| \cdot |f(x) - \widehat{f}(x)|\,dx. \end{aligned}$$

Iskoristimo ocjenu

$$|f(x) - \widehat{f}(x)| \leq \sup_{x\in[a,b]} |f(x) - \widehat{f}(x)| = \|f - \widehat{f}\|_\infty, \quad \forall x \in [a,b],$$

i definiciju L_1 norme funkcije w (koja je apsolutno integrabilna po pretpostavci)

$$\|w\|_1 = \int_a^b |w(x)|\,dx,$$

pa dobivamo traženu ocjenu. Ako za perturbiranu funkciju $\widehat{f}$ uzmemo

$$\widehat{f}(x) := f(x) + c\,\mathrm{sign}(w(x)),$$

gdje je $c > 0$ bilo koja konstanta, onda u ocjeni (9.4.4) dobivamo jednakost, uz $\|f - \widehat{f}\|_\infty = c$. ■

U ovoj formulaciji, za klasični Riemannov integral, domena $[a,b]$ integracije mora biti konačna. Teorem onda kaže da je apsolutni broj uvjetovanosti za $I_w(f)$ upravo jednak $\|w\|_1$ i ne ovisi o f, već samo o I_w.

Ovaj rezultat može se proširiti i na neprave Riemannove integrale (beskonačna domena, singulariteti funkcija), i tada više ne vrijedi zaključak o broju uvjetovanosti. Međutim, trenutno nam to nije bitno, već je ključna malo drugačija interpretacija ocjene (9.4.4).

Zamislimo da je $\widehat{f}$ neka aproksimacija (a ne perturbacija) funkcije f, koju želimo iskoristiti za približno računanje integrala. Onda (9.4.4) daje ocjenu (apsolutne) pogreške u integralu, preko greške aproksimacije funkcije u uniformnoj (L_∞) normi na $[a,b]$.

Ono što stvarno želimo dobiti je **niz** aproksimacija integrala koji konvergira prema $I_w(f)$. Jedan od puteva da to postignemo je izbor odgovarajućeg niza aproksimacija $\widehat{f}_n$, $n \in \mathbb{N}$, za funkciju f. Prethodna ocjena upućuje na to da, u ovisnosti

o n, za aproksimacijske funkcije $\widehat{f}_n$ treba uzimati takve funkcije za koje znamo da možemo postići po volji dobru **uniformnu** aproksimaciju funkcije f, jer tada

$$\|f - \widehat{f}_n\|_\infty \to 0 \Longrightarrow |I_w(f) - I_w(\widehat{f}_n)| \to 0, \quad n \to \infty.$$

Uočimo da ove aproksimacije, naravno, ovise o konkretnoj funkciji f. Da ne bismo za svaki novi f posebno konstruirali odgovarajući niz aproksimacija, poželjno je da bilo koju funkciju f, za koju postoji integral $I_w(f)$, možemo dovoljno dobro aproksimirati nekim prostorom funkcija. Tj. umjesto niza pojedinačnih aproksimacija, koristimo niz vektorskih prostora aproksimacijskih funkcija V_n, a za svaki pojedini f nađemo pripadnu aproksimaciju $\widehat{f}_n \in V_n$.

Weierstraßov teorem o uniformnoj aproksimaciji neprekidnih funkcija polinomima na konačnom intervalu $[a, b]$ sugerira da treba uzeti V_n kao prostor polinoma $\mathcal{P}_d$ stupnja manjeg ili jednakog d, gdje d ovisi o n (i raste s n). Kao što ćemo vidjeti, korisno je dozvoliti da bude $d \neq n$.

Isti princip koristimo i za beskonačne domene, samo treba osigurati da su polinomi integrabilni s težinom w. To postižemo dodatnim zahtjevom na težinsku funkciju w, tako da pretpostavimo da svi momenti težinske funkcije

$$\mu_k := \int_a^b x^k w(x)\, dx, \quad k \in \mathbb{N}_0, \tag{9.4.5}$$

postoje i da su konačni. U nastavku pretpostavljamo da težinska funkcija w zadovoljava ovu pretpostavku. Takve težinske funkcije obično zovemo (polinomno) dopustivima.

Napomenimo odmah da se ovaj pristup može generalizirati i na bilo koji drugi sustav funkcija aproksimacijskih funkcija $\{\widehat{f}_n \mid n \in \mathbb{N}\}$ koji je gust u prostoru $C[a, b]$ neprekidnih funkcija na $[a, b]$. Pripadni prostori V_n generirani su početnim komadima ovog sustava funkcija (kao linearne ljuske).

Za praktičnu primjenu ovog pristupa moramo moći efektivno izračunati integral $I_w(\widehat{f}_n)$ aproksimacijske funkcije, i to za bilo koju funkciju f. To se najlakše postiže tako da konstruiramo pripadnu integracijsku formulu I_n koja je egzaktna na cijelom prostoru $V_n = \mathcal{P}_d$ aproksimacijskih funkcija. Dakle, uvjet egzaktnosti za I_n je

$$I_w(f) = I_n(f) \quad \text{ili} \quad E_n(f) = 0, \quad \text{za sve } f \in V_n.$$

Iz relacija (9.4.3) i (9.4.4) odmah dobivamo i ocjenu greške pripadne integracijske formule $I_n(f)$, za bilo koji f

$$|E_n(f)| = |I_w(f) - I_n(f)| = |I_w(f) - I_w(\widehat{f}_n)| \leq \|w\|_1 \|f - \widehat{f}_n\|_\infty.$$

9.5. Gaussove integracijske formule

Kao što smo već rekli, Gaussove formule imaju dvostruko više slobodnih parametara nego Newton–Cotesove, pa bi zbog toga trebale egzaktno integrirati polinome približno dvostruko većeg stupnja od Newton–Cotesovih.

Za razliku od Newton–Côtesovih formula, **Gaussove integracijske formule** su oblika

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

u kojima točke integracije x_i nisu unaprijed poznate, nego se izračunaju tako da greška takve formule bude najmanja. Motivirani praktičnim razlozima, promatrat ćemo malo općenitije integracijske formule oblika

$$\int_a^b w(x)\,f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

gdje je w **težinska funkcija**, pozitivna na otvorenom intervalu (a, b). Koeficijente w_i zovemo **težinski koeficijenti** ili, skraćeno, **težine** integracijske formule. Gornji specijalni slučaj u kojem je $w \equiv 1$ čine formule koje se zovu **Gauss–Legendreove**. Težinska funkcija u općem slučaju utječe na težine i točke integracije, ali se ne pojavljuje eksplicitno u Gaussovoj formuli.

Bitno je znati da se za neke težinske funkcije na određenim intervalima, čvorovi i težine standardno tabeliraju u priručnicima. To su

težinska funkcija w	interval	formula Gauss–
1	$[-1, 1]$	Legendre
$\dfrac{1}{\sqrt{1-x^2}}$	$[-1, 1]$	Čebišev
$\sqrt{1-x^2}$	$[-1, 1]$	Čebišev 2. vrste
e^{-x}	$[0, \infty)$	Laguerre
e^{-x^2}	$(-\infty, \infty)$	Hermite

Glavni rezultat je sljedeći: ako zahtijevamo da formula integrira egzaktno polinome što je moguće većeg stupnja, onda su točke integracije x_i nultočke polinoma koji su ortogonalni na intervalu (a, b) obzirom na težinsku funkciju w, a težine w_i mogu se eksplicitno izračunati po formuli

$$w_i = \int_a^b w(x)\,\ell_i(x)\,dx, \quad i = 1, \ldots, n.$$

Pritom je ℓ_i poseban polinom Lagrangeove baze kojeg smo razmatrali u poglavlju o polinomnoj interpolaciji, definiran uvjetom $\ell_i(x_j) = \delta_{ij}$ (v. (7.2.18)). Primijetimo samo da je kod numeričke integracije zgodnije čvorove numerirati od x_1 do x_n, (za razliku od numeracije x_0 do x_n u poglavlju o interpolaciji), pa je i ℓ_i polinom stupnja $n-1$.

Kao što se Newton–Côtesove formule mogu dobiti integracijom Lagrangeovog interpolacijskog polinoma, tako se i Gaussove formule mogu dobiti integracijom Hermiteovog interpolacijskog polinoma. Takav pristup ekvivalentan je s pristupom u kojem zahtijevamo da Gaussove formule integriraju egzaktno polinome što je moguće višeg stupnja, tj. da vrijedi

$$\int_a^b w(x)\, x^j\, dx = \sum_{i=1}^{n} w_i x_i^j, \quad j = 0, 1, \ldots, 2n-1.$$

Mogli bismo iskoristiti ovu relaciju da napišemo $2n$ jednadžbi za $2n$ nepoznanica x_i i w_i, međutim nepoznanice x_i ulaze u sistem nelinearno, pa je ovakav pristup teži. Čak i dokaz da taj nelinearni sistem ima jedinstveno rješenje nije jednostavan.

Napišimo još jednom formulu za Hermiteov interpolacijski polinom h_{2n-1}, stupnja $2n-1$, koji u čvorovima integracije x_i interpolira vrijednosti $f_i = f(x_i)$ i $f_i' = f'(x_i)$, za $i = 1, \ldots, n$. Iz relacija (7.2.21) i (7.2.22) dobivamo

$$\begin{aligned} h_{2n-1}(x) &= \sum_{i=1}^{n} \Big(h_{i,0}(x)\, f_i + h_{i,1}(x)\, f_i' \Big) \\ &= \sum_{i=1}^{n} \Big([1 - 2(x-x_i)\ell_i'(x_i)]\, \ell_i^2(x)\, f_i + (x-x_i)\, \ell_i^2(x)\, f_i' \Big). \end{aligned}$$

Integracijom dobijemo

$$\int_a^b w(x)\, h_{2n-1}(x)\, dx = \sum_{i=1}^{n} \Big(A_i f_i + B_i f_i' \Big), \tag{9.5.1}$$

gdje su

$$\begin{aligned} A_i &= \int_a^b w(x)\, [1 - 2(x-x_i)\ell_i'(x_i)]\, \ell_i^2(x)\, dx, \\ B_i &= \int_a^b w(x)\, (x-x_i)\, \ell_i^2(x)\, dx. \end{aligned} \tag{9.5.2}$$

Integracijska formula (9.5.1) sliči na Gaussovu integracijsku formulu, osim što ima dodatne članove $B_i f_i'$, koji koriste i derivacije funkcije f u čvorovima integracije.

Kad bi, kao u Newton–Cotesovim formulama, čvorovi x_i bili unaprijed zadani, iz uvjeta egzaktne integracije polinoma trebalo bi odrediti $2n$ parametara — težinskih koeficijenata A_i, B_i. Zato očekujemo da ovakva formula egzaktno integrira

polinome do stupnja $2n-1$ (dimenzija prostora je $2n$). No, za upotrebu ove formule trebamo znati ne samo funkcijske vrijednosti $f(x_i)$ u čvorovima, već i vrijednosti derivacije $f'(x_i)$ funkcije u tim čvorovima.

Zato je ideja da probamo izbjeći korištenje derivacija, tako da izborom čvorova x_i **poništimo** koeficijente B_i uz derivacije f_i'. Točnost integracijske formule mora ostati ista (egzaktna integracija polinoma stupnja do $2n-1$), ali tako dobivena formula koristila bi samo funkcijske vrijednosti u čvorovima, tj. postala bi Gaussova integracijska formula.

Zaista, odgovarajućim izborom čvorova x_i može se postići da težinski koeficijenti B_i uz derivacije budu jednaki nula. Da bismo to dokazali, uvodimo posebni "polinom čvorova" (engl. "node polynomial") ω_n, koji ima nultočke u svim čvorovima integracije

$$\omega_n := (x-x_1)(x-x_2)\cdots(x-x_n).$$

Taj polinom smo već susreli u poglavlju o Lagrangeovoj interpolaciji. Sljedeći rezultat govori o tome kako treba izabrati čvorove.

Lema 9.5.1 *Ako je* $\omega_n(x) = (x-x_1)\cdots(x-x_n)$ **ortogonalna s težinom** w *na sve polinome nižeg stupnja, tj. ako vrijedi*

$$\int_a^b w(x)\,\omega_n(x)\,x^k\,dx = 0, \quad k = 0, 1, \ldots, n-1, \tag{9.5.3}$$

onda su svi koeficijenti B_i *u* (9.5.2) *jednaki nula.*

Dokaz. Lagano provjerimo identitet

$$(x-x_i)\ell_i(x) = \frac{\omega_n(x)}{\omega_n'(x_i)}. \tag{9.5.4}$$

Supstitucijom u izraz (9.5.2) za B_i slijedi

$$B_i = \frac{1}{\omega_n'(x_i)} \int_a^b w(x)\,\omega_n(x)\,\ell_i(x)\,dx.$$

Kako je ℓ_i polinom stupnja $n-1$, i po pretpostavci je ω_n ortogonalna s težinom w na sve takve polinome, tvrdnja slijedi. ■

Lako se vidi da vrijedi i obrat ove tvrdnje, tj. da su svi koeficijenti $B_i = 0$ u (9.5.1), ako i samo ako je polinom čvorova ω_n ortogonalan na sve polinome nižeg stupnja (do $n-1$), s težinskom funkcijom w. Razlog tome je što su funkcije ℓ_i, $i = 1, \ldots, n$, Lagrangeove baze zaista baza prostora $\mathcal{P}_{n-1}$ (zadatak 7.2.2).

Iz ranijih rezultata o ortogonalnim polinomima znamo da ortogonalni polinom stupnja n obzirom na w postoji i jednoznačno je određen do na (recimo) vodeći

koeficijent. Da bismo dobili Gaussovu integracijsku formulu u (9.5.1), polinom čvorova ω_n mora biti ortogonalni polinom s vodećim koeficijentom 1, tj. ω_n postoji i jedinstven je.

Nadalje, uvjet ortogonalnosti (9.5.3) **jednoznačno** određuje raspored čvorova za Gaussovu integraciju. Iz teorema 7.8.2 slijedi da ω_n ima n jednostrukih nultočaka u otvorenom intervalu (a, b) (što nam baš odgovara za integraciju). Njegove nultočke $x_1, \ldots, x_n$ možemo samo permutirati (drugačije indeksirati), a uz standardni dogovor $x_1 < \cdots < x_n$, one su jednoznačno određene.

Time smo dokazali da postoji jedinstvena Gaussova integracijska formula oblika

$$\int_a^b w(x)\, f(x)\, dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

Čvorovi integracije x_i su nultočke ortogonalnog polinoma stupnja n na $[a, b]$ s težinskom funkcijom w, a težinske koeficijente možemo izračunati iz (9.5.2), budući da je tada $w_i = A_i$, za $i = 1, \ldots, n$.

Iskoristimo li pretpostavku ortogonalnosti iz leme 9.5.1, možemo pojednostavniti i izraze za koeficijente $w_i = A_i$ u (9.5.2). Sasvim općenito, koristeći relaciju za B_i, koeficijent A_i možemo napisati u obliku

$$A_i = \int_a^b w(x)\, [1 - 2(x - x_i)\ell_i'(x_i)]\, \ell_i^2(x)\, dx = \int_a^b w(x)\, \ell_i^2(x)\, dx - 2\ell_i'(x_i) B_i.$$

Uz uvjet ortogonalnosti (Gaussova integracija) je $B_i = 0$ i $A_i = w_i$, pa je

$$w_i = \int_a^b w(x)\, \ell_i^2(x)\, dx.$$

Podintegralna funkcija je nenegativna i ℓ_i^2 je polinom stupnja $2(n-1)$ koji nije nul-polinom, pa desna strana mora biti pozitivna. Dakle, slijedi da su svi težinski koeficijenti u Gaussovoj integraciji pozitivni, $w_i > 0$, za $i = 1, \ldots, n$, što je vrlo bitno za numeričku stabilnost i konvergenciju.

Pokažimo još da vrijedi i

$$w_i = \int_a^b w(x)\, \ell_i^2(x)\, dx = \int_a^b w(x)\, \ell_i(x)\, dx.$$

Očito, to je isto kao i dokazati

$$\int_a^b w(x)\, \ell_i^2(x)\, dx - \int_a^b w(x)\, \ell_i(x)\, dx = \int_a^b w(x)\, \ell_i(x)\, (\ell_i(x) - 1)\, dx = 0.$$

Ali polinom $\ell_i(x) - 1$ se poništava u točki $x = x_i$, po definiciji polinoma ℓ_i, jer je $\ell_i(x_j) = \delta_{ij}$. Znači da $\ell_i(x) - 1$ mora sadržavati $x - x_i$ kao faktor, tj. možemo napisati

$$\ell_i(x) - 1 = (x - x_i)q(x),$$

gdje je q neki polinom stupnja $n-2$, za jedan manje od stupnja polinoma ℓ_i. Dakle,

$$\ell_i(x)\,(\ell_i(x) - 1) = \frac{\omega_n(x)}{\omega_n'(x_i)(x - x_i)}\,(\ell_i(x) - 1) = \frac{1}{\omega_n'(x_i)}\,\omega_n(x)\,q(x),$$

pa je zbog ortogonalnosti ω_n na sve polinome nižeg stupnja

$$\int_a^b w(x)\,\ell_i(x)\,(\ell_i(x) - 1)\,dx = \frac{1}{\omega_n'(x_i)} \int_a^b w(x)\,\omega_n(x)\,q(x)\,dx = 0.$$

■

Pokazali smo da Gaussovu integracijsku formulu možemo dobiti kao integral Hermiteovog interpolacijskog polinoma, uz odgovarajući izbor čvorova, a za težinske koeficijente vrijedi

$$w_i = \int_a^b w(x)\,\ell_i(x)\,dx. \tag{9.5.5}$$

Primijetimo da je ova formula za koeficijente ista kao i ona u Newton–Côtesovim formulama, što je ovdje posljedica pretpostavke o ortogonalnosti. U oba slučaja do integracijskih formula dolazimo interpolacijom funkcije u čvorovima.

Pokažimo i primjerom da ortogonalnost produkta korijenskih faktora, tj. funkcije $\omega_n(x)$ na sve polinome nižeg stupnja zapravo određuje točke integracije x_i.

Primjer 9.5.1 *Neka je $w(x) = 1$ i $n = 3$. Odredimo točke integracije iz uvjeta ortogonalnosti. Uobičajeno je da za interval integracije uzmemo $(-1, 1)$, budući da integrale na drugim intervalima možemo lagano računati, ako podintegralnu funkciju transformiramo linearnom supstitucijom. Problem se dakle svodi na to da odredimo nultočke kubične funkcije $\omega_3(x) = a + bx + cx^2 + x^3$ za koju vrijedi*

$$\int_{-1}^{1} \omega_3(x)\,x^k\,dx = 0, \quad k = 0, 1, 2.$$

Nakon integracije dobivamo sustav jednadžbi za koeficijente a, b, c

$$2a + \frac{2}{3}c = 0, \qquad \frac{2}{3}b + \frac{2}{5} = 0, \qquad \frac{2}{3}a + \frac{2}{5}c = 0,$$

odakle nađemo $a = c = 0$ i $b = -3/5$. Dobivamo

$$\omega_3(x) = x^3 - \frac{3}{5}x = \left(x + \sqrt{\frac{3}{5}}\right)x\left(x - \sqrt{\frac{3}{5}}\right),$$

odakle slijedi da su točke integracije $x_i = -\sqrt{3/5},\ 0,\ \sqrt{3/5}$.

Teorijski, ovaj pristup možemo iskoristiti za sve moguće intervale integracije i razne težinske funkcije. Za veće n potrebno je odrediti nule polinoma visokog stupnja, što je egzaktno nemoguće, a numerički u najmanju ruku neugodno. Stoga je potrebno za specijalne težine i intervale integracije doći do dodatnih informacija o ortogonalnim polinomima. Na kraju, bilo bi dobro izračunati formulom i težinske faktore w_i u Gaussovim formulama. Analitički je moguće doći do ovakvih rezultata za mnoge specijalne težine $w(x)$ koje se pojavljuju u primjenama. Riješimo na početku važnu situaciju $w \equiv 1$, $a = -1$, $b = 1$. Pripadne formule nazvali smo Gauss–Legendreovima; u gornjem primjeru izračunali smo točke integracije za Gauss–Legendreovu formulu reda 3.

Zadatak 9.5.1 *Iz uvjeta egzaktnosti i poznatih točaka integracije za $n = 3$ izračunajte težinske koeficijente w_i. Primijetite da je sustav jednadžbi linearan, pa stoga računanje ovih faktora ne predstavlja veće probleme.*

9.5.1. Gauss–Legendreove integracijske formule

Prepostavimo u daljnjem da je $w \equiv 1$ na intervalu $(-1, 1)$ i izvedimo specijalnu Gaussovu formulu, tj. Gauss–Legendre-ovu formulu

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i).$$

Kao što znamo, Legendreov polinom stupnja n definiran je **Rodriguesovom formulom**

$$P_n(x) = \frac{1}{2^n n!}\,\frac{d^n}{dx^n}(x^2-1)^n.$$

Tako definirani polinomi čine **ortogonalnu bazu** u prostoru polinoma stupnja n, tj. oni su linearno nezavisni i ortogonalni obzirom na skalarni produkt

$$\langle P, Q\rangle := \int_{-1}^{1} P(x)\,Q(x)\,dx. \tag{9.5.6}$$

Pojavljuju se prirodno u parcijalnim diferencijalnim jednadžbama, kod metode separacije varijabli za Laplaceovu jednadžbu u kugli. Za nas je bitno samo jedno specijalno svojstvo, iz kojeg slijede sva ostala:

Lema 9.5.2 *Legendreov polinom stupnja n ortogonalan je na sve potencije x^k nižeg stupnja, tj. vrijedi*

$$\int_{-1}^{1} x^k P_n(x)\,dx = 0, \quad \text{za} \quad k = 0, 1, \ldots, n-1,$$

i vrijedi

$$\int_{-1}^{1} x^n P_n(x)\,dx = \frac{2^{n+1}(n!)^2}{(2n+1)!}.$$

Dokaz. Uvrštavanjem Rodriguesove formule, nakon k ($k < n$) parcijalnih integracija dobivamo

$$\int_{-1}^{1} x^k \frac{d^n}{dx^n}(x^2-1)^n\,dx = \underbrace{x^k \frac{d^{n-1}}{dx^{n-1}}(x^2-1)^n\Big|_{-1}^{1}}_{=0} - \int_{-1}^{1} kx^{k-1}\frac{d^{n-1}}{dx^{n-1}}(x^2-1)^n\,dx$$

$$= \cdots = (-1)^k k! \int_{-1}^{1} \frac{d^{n-k}}{dx^{n-k}}(x^2-1)^n\,dx = 0,$$

pa smo dokazali prvu formulu. Za $k = n$, na isti način imamo

$$\int_{-1}^{1} x^n \frac{d^n}{dx^n}(x^2-1)^n\,dx = (-1)^n n! \int_{-1}^{1}(x^2-1)^n\,dx = 2n! \int_{0}^{1}(1-x^2)^n\,dx$$

$$= \{x = \sin t\} = 2n! \int_{0}^{\pi/2} \cos^{2n+1} t\,dt.$$

Za zadnji integral parcijalnom integracijom izlazi

$$\int_{0}^{\pi/2} \cos^{2n+1} t\,dt = \underbrace{\frac{\cos^{2n} t\,\sin t}{2n+1}\Big|_{0}^{\pi/2}}_{=0} + \frac{2n}{2n+1}\int_{0}^{\pi/2} \cos^{2n-1} t\,dt$$

$$= \cdots = \frac{2n(2n-2)\cdots 2}{(2n+1)(2n-1)\cdots 3}\int_{0}^{\pi/2} \cos t\,dt,$$

pa je stoga

$$\int_{-1}^{1} x^n \frac{d^n}{dx^n}(x^2-1)^n\,dx = 2n!\,\frac{2n(2n-2)\cdots 2}{(2n+1)(2n-1)\cdots 3}.$$

Pomnožimo li brojnik i nazivnik s $2n(2n-2)\cdots 2 = 2^n n!$, a zatim, zbog definicije Legendreovog polinoma P_n, sve podijelimo s $2^n n!$, slijedi

$$\int_{-1}^{1} x^n P_n(x)\,dx = \frac{1}{2^n n!}\,2n!\,\frac{2^n n!\cdot 2^n n!}{(2n+1)!} = \frac{2^{n+1}(n!)^2}{(2n+1)!}.$$

■

Lema 9.5.3 *Legendreovi polinomi su ortogonalni na intervalu* $(-1,1)$ *obzirom na skalarni produkt* (9.5.6)

$$\int_{-1}^{1} P_m(x)\,P_n(x)\,dx = 0, \quad \text{za} \quad m \neq n.$$

Norma Legendreovog polinoma je

$$\|P_n\|^2 := \int_{-1}^{1} [P_n(x)]^2\,dx = \frac{2}{2n+1}.$$

Dokaz. Prva tvrdnja je direktna posljedica dokazane ortogonalnosti na potencije nižeg stupnja. Druga tvrdnja slijedi iz

$$\int_{-1}^{1} [P_n(x)]^2\,dx = \int_{-1}^{1} \Big[\frac{1}{2^n n!}\,\frac{(2n)!}{n!}\,x^n + \cdots\Big] P_n(x)\,dx.$$

Potencije manje od x^n ne doprinose integralu, pa druga tvrdnja leme 9.5.2 povlači

$$\int_{-1}^{1} [P_n(x)]^2\,dx = \frac{(2n)!}{2^n (n!)^2}\,\frac{2^{n+1}(n!)^2}{(2n+1)!} = \frac{2}{2n+1}.$$

■

Lema 9.5.4 *Legendreovi polinomi* P_n *imaju* n *nultočaka, koje su sve realne i različite, i nalaze se u otvorenom intervalu* $(-1,1)$.

Dokaz. Dokaz ide iz definicije Legendreovih polinoma

$$P_n(x) = \frac{1}{2^n n!}\,\frac{d^n}{dx^n}(x^2-1)^n,$$

induktivnom primjenom Rolleovog teorema. Polinom $(x^2-1)^n$ je stupnja $2n$ i ima višestruke (n-terostruke) nultočke u rubovima intervala ± 1. Prema Rolleovom teoremu, prva derivacija ima jednu nultočku u intervalu $(-1,1)$. Međutim, prva derivacija je, također, nula u ± 1, pa ukupno mora imati tri nultočke u zatvorenom intervalu $[-1,1]$. Druga derivacija stoga ima dvije unutarnje nule po Rolleovom teoremu, i dvije u ± 1, pa ima ukupno četiri nule u $[-1,1]$. I tako redom, vidimo da $n-1$-a derivacija ima $n-1$ unutarnju nultočku i još dvije u ± 1. Na kraju zaključimo da n-ta derivacija, koja je do na multiplikativni faktor jednaka P_n, ima n unutarnjih nultočaka. ■

Na taj način smo zapravo našli točke integracije u Gauss–Legendreovoj formuli i bez eksplicitnog rješavanja nelinearnog sistema jednadžbi za w_i i x_i, iz uvjeta egzaktne integracije potencija najvećeg mogućeg stupnja. Taj rezultat rezimiran je u sljedećem teoremu.

Teorem 9.5.1 *Čvorovi integracije u Gauss–Legendreovoj formuli reda n su nultočke Legendreovog polinoma P_n, za svaki n.*

Dokaz. Znamo da su točke integracije x_i nultočke polinoma ω_n po konstrukciji. Zbog uvjeta ortogonalnosti (9.5.3) polinom ω_n, s vodećim koeficijentom 1, proporcionalan je Legendreovom polinomu P_n. Vodeći koeficijent u P_n lako izračunamo iz Rodriguesove formule, odakle je

$$\omega_n(x) = \frac{2^n (n!)^2}{(2n)!} P_n(x),$$

pa vidimo da su sve nultočke polinoma ω_n zapravo nultočke od P_n (lema 9.5.4). ■

Primjer 9.5.2 *Iz Rodriguesove formule možemo izračunati nekoliko prvih Legendreovih polinoma.*

$$\begin{aligned}
P_0(x) &= 1,\\
P_1(x) &= \frac{1}{2}\frac{d}{dx}(x^2-1) = x,\\
P_2(x) &= \frac{1}{8}\frac{d^2}{dx^2}(x^2-1)^2 = \frac{1}{2}(3x^2-1),\\
P_3(x) &= \frac{1}{48}\frac{d^3}{dx^3}(x^2-1)^3 = \frac{1}{2}(5x^3-3x),\\
P_4(x) &= \frac{1}{16\cdot 24}\frac{d^4}{dx^4}(x^2-1)^4 = \frac{1}{8}(35x^4-30x^2+3),\\
P_5(x) &= \frac{1}{8}(63x^5-70x^3+15x),\\
P_6(x) &= \frac{1}{16}(231x^6-315x^4+105x^2-5),\\
P_7(x) &= \frac{1}{16}(429x^7-693x^5+315x^3-35x),\\
P_8(x) &= \frac{1}{128}(6435x^8-12012x^6+6930x^4-1260x^2+35).
\end{aligned}$$

Vidimo, na primjer, da su nultočke od P_3 identične s točkama integracije koje smo dobili u primjeru 9.5.1*, direktno iz uvjeta ortogonalnosti.*

Računanje nultočaka Legendreovih polinoma (na mašinsku točnost!) nije jednostavan problem, budući da egzaktne formule postoje samo za male stupnjeve. Napomenimo za sad samo toliko, da postoje specijalni algoritmi, te da je dovoljno tabelirati te nultočke jednom, pa brzina algoritma nije važna, nego samo preciznost. Tabelirane nultočke (kao i težine w_i) moguće je naći u gotovo svim standardnim knjigama i tablicama iz područja numeričke analize.

Postoji lakši način za računanje $P_n(x)$, zasnovan na činjenici da Legendreovi polinomi zadovoljavaju tročlanu rekurziju, čiji se koeficijenti mogu eksplicitno izračunati. Ova rekurzivna formula igra važnu ulogu i u konstrukciji spomenutog specijalnog algoritma za traženje nultočaka.

Lema 9.5.5 *Legendreovi polinomi zadovoljavaju rekurzivnu formulu*

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \quad n \geq 1,$$

s početnim vrijednostima $P_0(x) = 1$, $P_1(x) = x$.

Dokaz. Kako je $xP_n(x)$ polinom stupnja $n+1$ i $\{P_i\}_{i=0}^{n+1}$ baza za prostor polinoma stupnja do $n+1$, postoje koeficijenti c_i tako da vrijedi

$$xP_n(x) = \sum_{i=0}^{n+1} c_i P_i(x).$$

Pomnožimo li obje strane s $P_k(x)$ i integriramo od -1 do 1, zbog ortogonalnosti (lema 9.5.3) slijedi

$$\int_{-1}^{1} xP_k(x)\, P_n(x)\, dx = c_k \int_{-1}^{1} P_k^2(x)\, dx. \tag{9.5.7}$$

Ali za $k < n-1$ je $xP_k(x)$ polinom stupnja manjeg ili jednakog $n-1$, pa je $P_n(x)$ ortogonalan na njega (lema 9.5.2). Stoga je $c_k = 0$ za $k < n-1$, a u sumi za $xP_n(x)$ ostaju samo zadnja tri člana

$$xP_n(x) = c_{n+1}P_{n+1}(x) + c_nP_n(x) + c_{n-1}P_{n-1}(x). \tag{9.5.8}$$

Treba još izračunati koeficijente c_{n+1}, c_n i c_{n-1}. Kako je

$$P_n(x) = \frac{(2n)!}{2^n(n!)^2}\,\omega_n(x) = \frac{(2n)!}{2^n(n!)^2}\,x^n + \text{niže potencije od } x,$$

usporedimo li koeficijente uz x^{n+1} u (9.5.8), dobivamo da je

$$\frac{(2n)!}{2^n(n!)^2} = c_{n+1}\,\frac{(2n+2)!}{2^{n+1}[(n+1)!]^2},$$

odakle slijedi da je

$$c_{n+1} = \frac{n+1}{2n+1}.$$

Lagano se vidi (iz Rodriguesove formule) da se u Legendreovim polinomima pojavljuju samo alternirajuće potencije, tj. P_{2n} je linearna kombinacija parnih potencija x^{2k}, $k = 0, \ldots, n$, a P_{2n+1} je linearna kombinacija neparnih potencija x^{2k+1},

$k = 0, \ldots, n$. Iz rekurzije (9.5.8) na osnovu toga zaključimo da je $c_n = 0$, pa preostaje samo izračunati c_{n-1}. Za $k = n-1$, iz (9.5.7) imamo da je

$$\int_{-1}^{1} xP_{n-1}(x)\, P_n(x)\, dx = c_{n-1} \int_{-1}^{1} P_{n-1}^2(x)\, dx.$$

Zbog

$$xP_{n-1}(x) = \frac{(2(n-1))!}{2^{n-1}[(n-1)!]^2}\, x^n + \text{niže potencije od } x$$

i ortogonalnosti P_n na sve niže potencije od x, dobivamo

$$\frac{(2n-2)!}{2^{n-1}[(n-1)!]^2} \int_{-1}^{1} x^n P_n(x)\, dx = c_{n-1} \int_{-1}^{1} P_{n-1}^2(x)\, dx.$$

Ovi integrali su poznati (lema 9.5.2 i lema 9.5.3), pa slijedi

$$c_{n-1} = \frac{n}{2n+1}.$$

Tako smo našli sve nepoznate koeficijente u linearnoj kombinaciji (9.5.8), odakle odmah slijedi tročlana rekurzija. Primijetimo da smo usput dokazali i formulu

$$\int_{-1}^{1} xP_{n-1}(x)\, P_n(x)\, dx = \frac{n}{2n+1}\, \frac{2}{2n-1} = \frac{2n}{4n^2-1}. \tag{9.5.9}$$

■

Zadatak 9.5.2 *Budući da Legendreovi polinomi zadovoljavaju tročlanu rekurziju, moguće je napisati algoritam za brzu sumaciju parcijalnih suma redova oblika*

$$\sum_{n=0}^{\infty} a_n P_n(x),$$

poznat pod nazivom **generalizirana Hornerova shema**. *Koristeći rekurziju iz leme* 9.5.5*, napišite eksplicitno taj algoritam. Razvoji po Legendreovim polinomima pojavljuju se često kod rješavanja Laplaceove jednadžbe u sfernim koordinatama.*

Sljedeće dvije leme korisne su za dobivanje ekplicitnih formula za težine u Gauss–Legendreovim formulama.

Lema 9.5.6 (Christoffel–Darbouxov identitet) *Za Legendreove polinome P_n vrijedi*

$$(t-x) \sum_{k=0}^{n} (2k+1) P_k(x) P_k(t) = (n+1)\, [P_{n+1}(t) P_n(x) - P_n(t) P_{n+1}(x)].$$

Dokaz. Pomnožimo li rekurziju iz leme 9.5.5 (uz zamjenu $n \mapsto k$) s $P_k(t)$, dobijemo

$$(2k+1)xP_k(x)P_k(t) = (k+1)P_{k+1}(x)P_k(t) + kP_{k-1}(x)P_k(t).$$

Zamijenimo li x i t imamo

$$(2k+1)tP_k(t)P_k(x) = (k+1)P_{k+1}(t)P_k(x) + kP_{k-1}(t)P_k(x).$$

Odbijanjem prve relacije od druge, slijedi

$$\begin{aligned}(2k+1)(t-x)P_k(x)P_k(t) = (k+1)\left[P_{k+1}(t)P_k(x) - P_k(t)P_{k+1}(x)\right] \\ - k\left[P_k(t)P_{k-1}(x) - P_{k-1}(t)P_k(x)\right].\end{aligned}$$

Sumiramo li po k od 1 do n, sukcesivni članovi u sumi na desnoj strani se krate, pa ostaju samo prvi i zadnji

$$(t-x)\sum_{k=1}^{n}(2k+1)P_k(x)P_k(t) = (n+1)\left[P_{n+1}(t)P_n(x) - P_n(t)P_{n+1}(x)\right] - (t-x).$$

Zadnji član možemo prebaciti na lijevu stranu kao nulti član u sumi, a to je baš Christoffel–Darbouxov identitet. ■

Lema 9.5.7 *Derivacija Legendreovih polinoma može se rekurzivno izraziti pomoću samih Legendreovih polinoma, formulom*

$$(1-x^2)P_n'(x) + nxP_n(x) = nP_{n-1}(x), \quad n \geq 1.$$

Dokaz. Polinom $(1-x^2)P_n' + nxP_n$ je očito stupnja manjeg ili jednakog od $n+1$. Napišimo P_n kao linearnu kombinaciju potencija od x (pojavljuje se samo svaka druga potencija)

$$P_n(x) = a_n x^n + a_{n-2}x^{n-2} + \cdots,$$

pa je

$$P_n'(x) = na_n x^{n-1} + (n-2)a_{n-2}x^{n-3} + \cdots.$$

No, onda je

$$(1-x^2)P_n'(x) + nxP_n(x) = (-na_n + na_n)x^{n+1} + O(x^{n-1}),$$

tj. polinom $(1-x^2)P_n' + nxP_n$ je zapravo stupnja $n-1$. Kao i u dokazu rekurzivne formule, moraju postojati koeficijenti c_i takovi da vrijedi

$$(1-x^2)P_n'(x) + nxP_n(x) = \sum_{i=0}^{n-1} c_i P_i(x).$$

Pomnožimo ovu relaciju s $P_k(x)$ i integriramo od -1 do 1. Zbog ortogonalnosti, na desnoj strani ostaje samo jedan član

$$\frac{2}{2k+1}\,c_k = \int_{-1}^{1}(1-x^2)\,P_n'(x)\,P_k(x)\,dx + n\int_{-1}^{1} xP_n(x)\,P_k(x)\,dx.$$

Prvi integral integriramo parcijalno, pa kako se faktor $(1-x^2)$ poništava na granicama integracije, slijedi

$$\frac{2}{2k+1}\,c_k = -\int_{-1}^{1} P_n(x)\,\frac{d}{dx}[(1-x^2)P_k(x)]\,dx + n\int_{-1}^{1} xP_n(x)\,P_k(x)\,dx.$$

Za $k < n-1$, oba integranda su oblika $P_n(x) \times$ (polinom stupnja najviše $n-1$), pa su svi ovi integrali jednaki nula (lema 9.5.2), tj. $c_k = 0$ za $k < n-1$. Za $k = n-1$ treba izračunati dva integrala u prethodnoj relaciji. Drugi je jednostavan

$$n\int_{-1}^{1} xP_n(x)\,P_{n-1}(x)\,dx = (9.5.9) = \frac{2n^2}{4n^2-1}.$$

U prvom integralu

$$-\int_{-1}^{1} P_n(x)\,\frac{d}{dx}[(1-x^2)P_{n-1}(x)]\,dx,$$

zbog prve tvrdnje u lemi 9.5.2 (ortgonalnost), doprinos daje samo vodeći član u $(1-x^2)P_{n-1}(x)$, pa je taj integral jednak

$$\int_{-1}^{1} P_n(x)\,\frac{d}{dx}\left\{x^2\,\frac{(2n-2)!}{2^{n-1}[(n-1)!]^2}\,x^{n-1}\right\}dx,$$

a zbog druge tvrdnje u lemi, integral se svodi na

$$\frac{(2n-2)!}{2^{n-1}[(n-1)!]^2}\,(n+1)\,\frac{2^{n+1}(n!)^2}{(2n+1)!} = \frac{2n(n+1)}{(2n+1)(2n-1)}.$$

Na kraju je

$$c_{n-1} = \frac{2n-1}{2}\left[\frac{2n(n+1)}{(2n+1)(2n-1)} + \frac{2n^2}{(2n+1)(2n-1)}\right] = n,$$

što smo i htjeli dokazati. ■

Lema 9.5.8 *Težinski faktori u Gauss–Legendreovim formulama mogu se eksplicitno izračunati formulama*

$$w_i = \frac{2(1-x_i^2)}{n^2[P_{n-1}(x_i)]^2},$$

gdje su x_i, $i = 0, \ldots, n$, *nultočke Legendreovog polinoma* P_n.

Dokaz. Neka je x_i nultočka polinoma P_n. Stavimo li $t = x_i$ u Christoffel–Darbouxov identitet (lema 9.5.6), dobivamo

$$\frac{(n+1)P_{n+1}(x_i)P_n(x)}{x - x_i} = -\sum_{k=0}^{n}(2k+1)P_k(x)P_k(x_i).$$

Kad integriramo ovu jednakost od -1 do 1 i uzmemo u obzir da je k-ti Legendreov polinom ortogonalan na konstantu $P_k(x_i)$, na desnoj strani preostane samo član za $k = 0$

$$\int_{-1}^{1} \frac{P_n(x)}{(x - x_i)}\,dx = \frac{-2}{(n+1)P_{n+1}(x_i)}.$$

Tročlana rekurzija iz leme 9.5.5 u nultočki x_i Legendreovog polinoma P_n ima oblik $(n+1)P_{n+1}(x_i) = -nP_{n-1}(x_i)$, pa je stoga

$$\int_{-1}^{1} \frac{P_n(x)}{(x - x_i)}\,dx = \frac{2}{nP_{n-1}(x_i)}.$$

Za težinske koeficijente w_i vrijede relacije (9.5.5) i (9.5.4)

$$w_i = \int_{-1}^{1} \ell_i(x)\,dx = \int_{-1}^{1} \frac{\omega_n(x)}{\omega_n'(x_i)(x - x_i)}\,dx = \int_{-1}^{1} \frac{P_n(x)}{P_n'(x_i)(x - x_i)}\,dx,$$

pa je dakle

$$w_i = \frac{2}{nP_n'(x_i)\,P_{n-1}(x_i)}. \tag{9.5.10}$$

Primijetimo da je Christoffel–Darbouxov identitet potreban jedino zato da se izračuna neugodan integral

$$\int_{-1}^{1} \frac{P_n(x)}{(x - x_i)}\,dx,$$

u kojem podintegralna funkcija ima uklonjivi singularitet.

Na kraju, iskoristimo rekurzivnu formulu za derivacije Legendreovog polinoma iz leme 9.5.7 u specijalnom slučaju kada je $x = x_i$. Dobivamo da vrijedi

$$(1 - x_i^2)P_n'(x_i) = nP_{n-1}(x_i).$$

Uvrstimo li taj rezultat u (9.5.10), tvrdnja slijedi. ■

U dokazu prethodne leme 9.5.8 pokazali smo (usput) da u nultočki x_i Legendreovog polinoma P_n vrijedi

$$(1 - x_i^2)P_n'(x_i) = nP_{n-1}(x_i) = -(n+1)P_{n+1}(x_i).$$

Ovu relaciju možemo iskoristiti na različite načine u (9.5.10), što daje pet raznih formula za težinske koeficijente u Gauss–Legendreovim formulama

$$\begin{aligned} w_i &= \frac{2(1-x_i^2)}{[nP_{n-1}(x_i)]^2} = \frac{2(1-x_i^2)}{[(n+1)P_{n+1}(x_i)]^2} \\ &= \frac{2}{nP_n'(x_i)\,P_{n-1}(x_i)} = -\frac{2}{(n+1)P_n'(x_i)\,P_{n+1}(x_i)} \\ &= \frac{2}{(1-x_i^2)\,[P_n'(x_i)]^2}. \end{aligned} \tag{9.5.11}$$

Sljedeći teorem rezimira prethodne rezultate, i ujedno daje ocjenu greške za Gauss–Legendreovu integraciju.

Teorem 9.5.2 *Za funkciju $f \in C^{2n}[-1,1]$ Gauss–Legendreova formula integracije glasi*

$$\int_{-1}^{1} f(x)\,dx = \sum_{i=1}^{n} w_i f(x_i) + E_n(f),$$

gdje su x_i nultočke Legendreovog polinoma P_n i koeficijenti w_i dani u (9.5.11). *Za grešku $E_n(f)$ vrijedi*

$$E_n(f) = \frac{2^{2n+1}(n!)^4}{(2n+1)\,[(2n)!]^3}\, f^{(2n)}(\xi), \quad \xi \in (-1,1).$$

Dokaz. Treba samo dokazati formulu za ocjenu greške. Kako je Gauss–Legendreova formula zapravo integral Hermiteovog interpolacijskog polinoma, treba integrirati grešku kod Hermiteove interpolacije, koju smo procijenili u teoremu 7.2.5, i uvrstiti odgovarajući ω_n. Integracijom i primjenom teorema srednje vrijednosti za integrale, dobivamo

$$E_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \int_{-1}^{1} \omega_n^2(x)\,dx,$$

za neki $\xi \in (-1,1)$. Kako je

$$\omega_n(x) = \frac{2^n(n!)^2}{(2n)!}\, P_n(x),$$

zbog poznatog kvadrata norme Legendreovog polinoma (lema 9.5.3), imamo

$$E_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \left[\frac{2^n(n!)^2}{(2n)!}\right]^2 \frac{2}{2n+1} = \frac{2^{2n+1}(n!)^4}{(2n+1)\,[(2n)!]^3}\, f^{(2n)}(\xi).$$

■

Navedeni izraz za grešku nije lagano primijeniti, budući da je potrebno naći neku ogradu za vrlo visoku derivaciju funkcije f (red derivacije je dva puta veći nego kod Newton–Côtesovih formula). Član uz $f^{(2n)}(\xi)$ vrlo brzo pada s porastom n. Na primjer, za $n = 6$, greška je oblika

$$1.6 \cdot 10^{-12}\, f^{(12)}(\xi).$$

Da ocjena greške za Gaussove formule može biti previše pesimistična, pokazuje sljedeći primjer.

Primjer 9.5.3 *Primijenimo Gauss–Legendreovu formulu na integral*

$$\int_0^{\pi/2} \log(1+t)\,dt = \left(1+\frac{\pi}{2}\right)\left[\log\left(1+\frac{\pi}{2}\right)-1\right]+1.$$

Zamjena varijable $t = \pi(x+1)/4$ *prebacuje integral na standardnu formu*

$$\int_{-1}^{1} \frac{\pi}{4}\,\log\left(1+\frac{\pi(x+1)}{4}\right)dx.$$

U ovom slučaju možemo lagano izračunati bilo koju derivaciju podintegralne funkcije, koja raste s faktorijelima. Zapravo, sve ocjene greške formula za numeričku integraciju pokazuju slično ponašanje (usporedite, na primjer, trapeznu i Simpsonovu formulu), ali Gaussove formule naročito, budući da uključuju visoke derivacije. Tako je, na primjer, osma derivacija, koja je potrebna za Gaussovu formulu s četiri točke jednaka

$$\left(\frac{\pi}{4}\right)^9 \cdot \frac{-7!}{(1+t)^8},$$

pa je greška 7! *puta veća nego da smo, recimo, integrirali trigonometrijsku funkciju* sin *ili* cos*, koje imaju ograničene derivacije. Ipak, lagano vidimo da već sa šest točaka dobivamo* 6 *znamenaka točno, iako ocjena greške uključuje faktor od* 11!*. Simpsonovoj formuli treba* 64 *točke za istu točnost. Možemo slutiti, da je za analitičke funkcije moguća bolja ocjena greške.*

Korolar 9.5.1 (Uvjeti egzaktnosti) *Gauss–Legendreova formula egzaktno integrira polinome stupnja* $2n-1$*.*

Dokaz. Očito, budući da se greška, koja uključuje $2n$-tu derivaciju, poništava na takvim polinomima. ■

Svojstvo iz gornjeg korolara može se upotrijebiti za alternativni dokaz teorema 9.5.2, kao što smo napomenuli na početku. Hermiteova interpolacija poslužila je kao "trik", da izbjegnemo rješavanje nelinearnog sistema koji proizilazi iz uvjeta egzaktnosti.

Rekurziju za derivacije Legendreovih polinoma iz leme 9.5.7 možemo koristiti i za računanje vrijednosti $P_n'(x)$

$$(1-x^2)P_n'(x) = n(P_{n-1}(x) - xP_n(x)), \quad n \geq 1.$$

Nažalost, ova formulu ne možemo upotrijebiti u rubnim točakama $x = \pm 1$, zbog dijeljenja s nulom. Međutim, Legendreovi polinomi zadovoljavaju i mnoge druge rekurzivne relacije. Neke od njih dane su u sljedećem zadatku.

Zadatak 9.5.3 *Dokažite da za Legendreove polinoma vrijedi $P_n(1) = 1$, za $n \geq 0$, što opravdava izbor normalizacije. Također, dokažite da za $n \geq 1$ vrijede rekurzivne relacije*

$$\begin{aligned} P_n'(x) - xP_{n-1}'(x) &= nP_{n-1}(x), \\ xP_n'(x) - P_{n-1}'(x) &= nP_n(x), \\ P_{n+1}'(x) - P_{n-1}'(x) &= (2n+1)P_n(x) \\ \int_{-1}^{x} P_n(t)\,dt &= \frac{1}{2n+1}\left(P_{n+1}(x) - P_{n-1}(x)\right). \end{aligned}$$

Na kraju, primijetimo da Gaussove formule možemo shvatiti i kao rješenje optimizacijskog problema: naći točke integracije tako da egzaktno integriramo polinom što većeg stupnja sa što manje čvorova. Rezultat su formule visoke točnosti, koje se lagano implementiraju, i imaju vrlo mali broj izvrednjavanja podintegralne funkcije. Cijenu smo platili time što ocjena greške zahtijeva vrlo glatku funkciju, ali također i time što upotreba takvih formula na "finijoj" mreži zahtijeva ponovno računanje funkcije u drugim čvorovima, koji s čvorovima formule nižeg reda nemaju ništa zajedničko. Kod profinjavanja mreže čvorova za formule Newton–Côtesovog tipa (na primjer, raspolavljanjem h), naprotiv, jedan dio čvorova ostaje zajednički, pa već izračunate funkcijske vrijednosti možemo iskoristiti (kao u Rombergovom algoritmu).

9.5.2. Druge Gaussove integracijske formule

U praksi se često javljaju specijalni integrali koji uključuju težinske funkcije poput e^{-x}, e^{-x^2} i mnoge druge, na specijalnim intervalima, često neograničenim. Jednostavnom linearnom supstitucijom nije moguće takve intervale i/ili težinske funkcije prebaciti na interval $(-1, 1)$ i jediničnu težinsku funkciju — situaciju u kojoj možemo primijeniti Gauss–Legendreove formule.

Alternativa je iskoristiti odgovarajuće Gaussove formule s "prirodnom" težinskom funkcijom. Iz prethodnog odjeljka znamo da za čvorove integracije treba uzeti nultočke funkcije $\omega_n(x) = (x - x_1)\cdots(x - x_n)$, s tim da vrijede relacije ortogonalnosti (9.5.3). Težine w_i onda možemo odrediti rješavanjem linearnog sistema, a

možda u specijalnim slučajevima možemo doći i do eksplicitnih formula, kao što smo to učinili u slučaju Gauss–Legendreovih formula. Postavlja se pitanje da li možemo doći do formula za polinome koji su ortogonalni (obzirom na težinsku funkciju w) na polinome nižeg stupnja, uključivo i ostale formule na koje smo se oslanjali, poput tročlane rekurzije i slično (v. lema 9.5.2).

U mnogim važnim slučajevima, ali ne i uvijek, moguće je analitički doći do formula sličnim onima u slučaju Gauss–Legendreove integracije. U drugim slučajevima, koji nisu pokriveni egzaktnim formulama, u principu je moguće generirati ortogonalne polinome i numerički. Poznati postupci (Stieltjesov i Čebiševljev algoritam) ne pokrivaju, međutim, sve moguće situacije, tj. nisu uvijek numerički stabilni, što ostavlja postora za daljnja istraživanja. Slučajevi tzv. **klasičnih ortogonalnih polinoma** uglavnom se mogu karakterizirati na osnovu sljedeća dva teorema, od kojih je prvi egzistencijalni, i vezan uz teoriju rubnih problema za obične diferencijalne jednadžbe.

Teorem 9.5.3 (Generalizirana Rodriguesova formula)

Na otvorenom intervalu (a, b) postoji, do na multiplikativnu konstantu, jedinstvena funkcija $U_n(x)$ koja zadovoljava diferencijalnu jednadžbu

$$D^{n+1}\left(\frac{1}{w(x)}\, D^n U_n(x)\right) = 0$$

i rubne uvjete

$$U_n(a) = DU_n(a) = \cdots = D^{n-1}U_n(a) = 0,$$
$$U_n(b) = DU_n(b) = \cdots = D^{n-1}U_n(b) = 0.$$

Ovdje opet koristimo oznaku D za operator deriviranja funkcije f jedne varijable, kad je iz konteksta očito po kojoj varijabli se derivira, jer ta oznaka znatno skraćuje zapis nekih dugih formula. Onda n-tu derivaciju funkcije f u točki x možemo pisati u bilo kojem od sljedeća tri oblika

$$D^n f(x) = \frac{d^n}{dx^n}\, f(x) = f^{(n)}(x).$$

Budući da nas interesiraju rješenja koja se mogu eksplicitno konstruirati, nećemo dokazivati ovaj teorem. U svakom konkretnom slučaju, za zadane a, b i $w(x)$, konstruirat ćemo funkciju U_n formulom. Napomenimo još da teorem 9.5.3 vrijedi i na neograničenim i poluograničenim intervalima, tj. u slučajevima $a = -\infty$ i/ili $b = \infty$.

Funkcije U_n iz prethodnog teorema generiraju familiju ortogonalnih polinoma na (a, b) s težinskom funkcijom w.

Teorem 9.5.4 *Uz pretpostavke teorema* 9.5.3*, funkcije*

$$p_n(x) = \frac{1}{w(x)}\, D^n U_n(x)$$

su polinomi stupnja n koji su ortogonalni na sve polinome nižeg stupnja na intervalu (a, b) obzirom na težinsku funkciju $w(x)$, tj. vrijedi

$$\int_a^b w(x)\, p_n(x)\, x^k\, dx = 0, \quad \text{za} \quad k = 0, 1, \ldots, n-1.$$

Dokaz. Funkcija p_n je očito polinom stupnja n, jer je $D^{n+1} p_n(x) = 0$. Da dokažemo ortogonalnost, pretpostavimo da je $n \geq 1$. Za $k = 0$ imamo odmah po Newton–Lebnitzovoj formuli

$$\int_a^b w(x)\, p_n(x)\, dx = \int_a^b D^n U_n(x)\, dx = (n \geq 1) = D^{n-1} U_n(x)\Big|_a^b = 0,$$

zbog rubnih uvjeta $D^{n-1} U_n(a) = D^{n-1} U_n(b) = 0$.

Za $1 \leq k \leq n-1$, integriramo parcijalno k puta i iskoristimo opet rubne uvjete koje zadovoljava funkcija U_n. Dobivamo redom

$$\begin{aligned}
\int_a^b w(x) p_n(x)\, x^k\, dx &= \int_a^b x^k\, D^n U_n(x)\, dx \\
&= \underbrace{x^k\, D^{n-1} U_n(x)\Big|_a^b}_{=0} - k \int_a^b x^{k-1}\, D^{n-1} U_n(x)\, dx \\
&= -k \left(\underbrace{x^{k-1}\, D^{n-2} U_n(x)\Big|_a^b}_{=0} - (k-1) \int_a^b x^{k-2}\, D^{n-2} U_n(x)\, dx \right) \\
&= \cdots = (-1)^{k-1} k(k-1) \cdots 2 \left(\underbrace{x\, D^{n-k} U_n(x)\Big|_a^b}_{=0} - \int_a^b D^{n-k} U_n(x)\, dx \right) \\
&= (-1)^k k(k-1) \cdots 2 \cdot 1 \left(\underbrace{D^{n-k-1} U_n(x)\Big|_a^b}_{=0} \right) = 0,
\end{aligned}$$

jer je $n-k-1 \geq 0$. Primijetimo da smo za dokaz ortogonalnosti iskoristili sve rubne uvjete na funkciju U_n. ■

Ovaj teorem u mnogim slučajevima omogućava efektivnu konstrukciju ortogonalnih polinoma.

Primjer 9.5.4 *Neka je $w(x) = 1$ na intervalu $(-1, 1)$. Nađimo pripadne ortogonalne polinome. Prema teoremu* 9.5.3*, prvi korak je rješavanje diferencijalne jednadžbe*

$$D^{n+1}(D^n U_n(x)) = D^{2n+1} U_n(x) = 0,$$

uz rubne uvjete

$$U_n(\pm 1) = DU_n(\pm 1) = \cdots = D^{n-1} U_n(\pm 1) = 0.$$

Polinom $2n$-tog stupnja koji se poništava u krajevima mora, zbog simetrije, biti oblika $U_n(x) = C_n(x^2 - 1)^n$, gdje je C_n proizvoljna multiplikativna konstanta (različita od nule). Tradicionalno, konstanta C_n uzima se u obliku

$$C_n = \frac{1}{2^n n!}.$$

Pripadni ortogonalni polinomi su tada, prema teoremu 9.5.4*, dani formulom*

$$P_n(x) = \frac{1}{2^n n!} D^n (x^2 - 1)^n,$$

tj. dobivamo, očekivano, Legendreove polinome.

Zadatak 9.5.4 *Pokažite da je multiplikativna konstanta C_n odabrana tako da vrijedi $P_n(1) = 1$, za svako n. Također, pokažite da vrijedi $|P_n(x)| \leq 1$, za svaki $x \in [-1, 1]$ i svaki $n \geq 0$. To znači da P_n dostiže ekstreme u rubovima intervala, što je dodatno opravdanje za izbor normalizacije, jer je $\|P_n\|_\infty = 1$ na $[-1, 1]$.*

Primjer 9.5.5 *Neka je $w(x) = e^{-\alpha x}$ na intervalu $(0, \infty)$, za neki $\alpha > 0$. Nađimo pripadne ortogonalne polinome. Prema teoremu* 9.5.3*, trebamo prvo riješiti diferencijalnu jednadžbu*

$$D^{n+1}(e^{\alpha x} D^n U_n(x)) = 0,$$

uz rubne uvjete

$$U_n(0) = DU_n(0) = \cdots = D^{n-1} U_n(0) = 0,$$
$$U_n(\infty) = DU_n(\infty) = \cdots = D^{n-1} U_n(\infty) = 0.$$

Očito je rješenje oblika

$$U_n(x) = e^{-\alpha x}(c_0 + c_1 x + \cdots + c_n x^n) + d_0 + d_1 x + \cdots + d_{n-1} x^{n-1}.$$

Rubni uvjet u točki ∞ povlači $d_0 = \cdots = d_{n-1} = 0$, a rubni uvjet u točki 0 povlači $c_0 = \cdots = c_{n-1} = 0$, pa je

$$U_n(x) = C_n x^n e^{-\alpha x}.$$

Polinomi za koje je $\alpha = 1$ i $C_n = 1$ zovu se tradicionalno **Laguerreovi polinomi***, u oznaci $\widetilde{L}_n$. Njihova Rodriguesova formula je dakle*

$$\widetilde{L}_n(x) = e^x D^n(x^n e^{-x}).$$

U općem slučaju, za $\alpha \neq 1$, uz $C_n = 1$, lagano vidimo da je $p_n(x) = \widetilde{L}_n(\alpha x)$. Tada vrijede relacije ortogonalnosti

$$\int_0^\infty e^{-\alpha x}\, \widetilde{L}_m(\alpha x)\, \widetilde{L}_n(\alpha x)\, dx = 0, \quad m \neq n.$$

Napomenimo još da oznaku L_n koristimo za ortonormirane Laguerreove polinome. Njih dobivamo izborom normalizacijske konstante $C_n = 1/n!$, pa su $\widetilde{L}_n$ i L_n vezani relacijom $\widetilde{L}_n(x) = n!\, L_n(x)$.

Primjer 9.5.6 *Neka je $w(x) = e^{-\alpha^2 x^2}$ na intervalu $(-\infty, \infty)$, za neki $\alpha \neq 0$. Nađimo pripadne ortogonalne polinome. Prema teoremu 9.5.3, trebamo prvo riješiti diferencijalnu jednadžbu*

$$D^{n+1}(e^{\alpha^2 x^2}\, D^n U_n(x)) = 0,$$

uz rubne uvjete

$$U_n(\pm\infty) = DU_n(\pm\infty) = \cdots = D^{n-1}U_n(\pm\infty) = 0.$$

Lagano pogodimo da je

$$U_n(x) = C_n\, e^{-\alpha^2 x^2}.$$

Odaberemo li $\alpha^2 = 1$ i multiplikativnu konstantu $C_n = (-1)^n$, dolazimo do klasičnih polinoma, koji nose ime **Hermiteovi polinomi**, *u oznaci H_n, s Rodriguesovom formulom*

$$H_n(x) = (-1)^n e^{x^2}\, D^n(e^{-x^2}).$$

U općem slučaju, za $\alpha^2 \neq 1$, uz $C_n = (-\alpha)^n$, lagano vidimo da su polinomi koje tražimo oblika

$$p_n(x) = H_n(\alpha x) = (-\alpha)^n\, e^{\alpha^2 x^2}\, D^n(e^{-\alpha^2 x^2}).$$

Pripadne relacije ortogonalnosti su

$$\int_{-\infty}^\infty e^{-\alpha^2 x^2}\, H_m(\alpha x)\, H_n(\alpha x)\, dx = 0, \quad m \neq n.$$

U literaturi se ponekad može naći još jedna definicija za klasične Hermiteove polinome, koja odgovara izboru $\alpha^2 = 1/2$, uz $C_n = (-1)^n$.

Svi ortogonalni polinomi zadovoljavaju tročlane rekuzije (v. izvod Stieltjesovog algoritma uz metodu najmanjih kvadrata). Za Laguerreove i Hermiteove polinome mogu se analitički izračunati koeficijenti u rekurziji, postupkom koji je vrlo sličan onom kojeg smo u detalje proveli u slučaju Legendreovih polinoma. Primijetimo, također, da i Čebiševljevi polinomi prve vrste zadovoljavaju relacije ortogonalnosti i tročlanu rekurziju, i da smo taj slučaj do kraja proučili. Kako su čvorovi Gaussove

formule integracije reda n nultočke odgovarajućeg ortogonalnog polinoma p_n, preostaje još samo izračunati težine w_i po formuli (9.5.5). Sasvim općenito, može se pokazati da vrijedi

$$w_i = \int_a^b w(x)\,\ell_i(x)\,dx = \frac{1}{p_n'(x_i)} \int_a^b w(x)\,\frac{p_n(x)}{x - x_i}\,dx,$$

gdje su ℓ_i polinomi Lagrangeove baze, i te integrale treba naći egzaktno. Formule za težine mogu se dobiti za cijeli niz klasičnih ortogonalnih polinoma, ali njihovo računanje ovisi o specijalnim svojstvima, posebnim rekurzijama i identitetima oblika Christoffel–Darbouxovog. Obzirom na duljinu ovih izvoda, zadovoljimo se s kratkim opisom nekoliko najpoznatijih Gaussovih formula.

Gauss–Laguerreove formule

Formule oblika

$$\int_0^\infty e^{-x}\,f(x)\,dx \approx \sum_{i=1}^n w_i f(x_i)$$

zovu se **Gauss–Laguerreove formule**. Čvorovi integracije su nultočke polinoma $\widetilde{L}_n$ definiranih Rodriguesovom formulom

$$\widetilde{L}_n(x) = e^x\,\frac{d^n}{dx^n}(x^n e^{-x}),$$

a težine u Gaussovoj formuli su

$$\begin{aligned} w_i &= \frac{[(n-1)!]^2 x_i}{[n\widetilde{L}_{n-1}(x_i)]^2} = \frac{(n!)^2 x_i}{[\widetilde{L}_{n+1}(x_i)]^2} \\ &= -\frac{[(n-1)!]^2}{\widetilde{L}_n'(x_i)\,\widetilde{L}_{n-1}(x_i)} = \frac{(n!)^2}{\widetilde{L}_n'(x_i)\,\widetilde{L}_{n+1}(x_i)} \\ &= \frac{(n!)^2}{x_i[\widetilde{L}_n'(x_i)]^2}. \end{aligned}$$

Greška kod numeričke integracije dana je formulom

$$E_n(f) = \frac{(n!)^2}{(2n)!}\,f^{(2n)}(\xi), \quad \xi \in (0,\infty).$$

Gauss–Hermiteove formule

Formule oblika

$$\int_{-\infty}^\infty e^{-x^2}\,f(x)\,dx \approx \sum_{i=1}^n w_i f(x_i)$$

zovu se **Gauss–Hermiteove formule**. Čvorovi integracije su nultočke polinoma H_n definiranih Rodriguesovom formulom

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2}),$$

a težine u Gaussovoj formuli su

$$\begin{aligned} w_i &= \frac{2^{n-1}(n-1)!\sqrt{\pi}}{n[H_{n-1}(x_i)]^2} = \frac{2^{n+1}n!\sqrt{\pi}}{[H_{n+1}(x_i)]^2} \\ &= \frac{2^n(n-1)!\sqrt{\pi}}{H_n'(x_i)\,H_{n-1}(x_i)} = -\frac{2^{n+1}n!\sqrt{\pi}}{H_n'(x_i)\,H_{n+1}(x_i)} \\ &= \frac{2^{n+1}n!\sqrt{\pi}}{[H_n'(x_i)]^2}. \end{aligned}$$

Greška kod numeričke integracije dana je formulom

$$E_n(f) = \frac{n!\sqrt{\pi}}{2^n(2n)!} f^{(2n)}(\xi), \quad \xi \in (-\infty, \infty).$$

Gauss–Čebiševljeve formule

Formule oblika

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

zovu se **Gauss–Čebiševljeve formule**. Čvorovi integracije su nultočke Čebiševljevih polinoma $T_n(x) = \cos(n \arccos(x))$. Izuzetno, te se nultočke mogu eksplicitno izračunati

$$x_i = \cos\left(\frac{(2i-1)\pi}{2n}\right).$$

Sve težine w_i su jednake

$$w_i = \frac{\pi}{n}.$$

Greška kod numeričke integracije dana je formulom

$$E_n(f) = \frac{\pi}{2^{2n-1}(2n)!} f^{(2n)}(\xi), \quad \xi \in (-1, 1).$$

Zadatak 9.5.5 *Neka je težinska funkcija $w(x) = (x-a)^\alpha (b-x)^\beta$ na intervalu (a, b), gdje su $\alpha > -1$ i $\beta > -1$. Nađite funkciju U_r i napišite Rodriguesovu formulu! Pridruženi ortogonalni polinomi zovu se* **Jacobijevi polinomi**. *Legendreovi i Čebiševljevi polinomi specijalni su slučaj.*

Pomoću Gaussovih formula možemo jednostavno računati neke određene integrale analitički, kao što se vidi iz sljedećih primjera.

Primjer 9.5.7 *Ako Gauss–Laguerreovom formulom reda $n = 1$ računamo integral*

$$\int_0^\infty e^{-x}\,dx,$$

imamo približnu formulu

$$\int_0^\infty e^{-x}\,f(x)\,dx \approx f(1),$$

budući da je $\widetilde{L}_1(x) = 1 - x$, pa je $x_1 = 1$ i $w_1 = 1/[\widetilde{L}'(1)]^2 = 1$. Kako formula egzaktno integrira konstante, za $f(x) = 1$ imamo

$$\int_0^\infty e^{-x}\,dx = f(1) = 1.$$

Slično, za $f(x) = ax + b$, budući da formula egzaktno integrira i linearne funkcije,

$$\int_0^\infty e^{-x}\,(ax + b)\,dx = f(1) = a + b.$$

Primjer 9.5.8 *Ako Gauss–Čebiševljevom formulom računamo*

$$\int_{-1}^{1} \frac{x^4}{\sqrt{1 - x^2}}\,dx$$

zgodno je upotrijebiti formulu Gauss–Čebiševa reda 3, koja zahtijeva nultočke polinoma $T_3(x) = 4x^3 - 3x$, a to su $x_1 = 0$, $x_{2,3} = \pm\sqrt{3}/2$. Formula vodi na egzaktan rezultat

$$\int_{-1}^{1} \frac{x^4}{\sqrt{1 - x^2}}\,dx = \frac{\pi}{3}\left(0 + \frac{9}{16} + \frac{9}{16}\right) = \frac{3\pi}{8}.$$

10. Obične diferencijalne jednadžbe

10.1. Uvod

Rješavanje diferencijalnih jednadžbi je problem koji se često javlja u raznim primjenama. Dok je nekim jednadžbama rješenje moguće eksplicitno izraziti pomoću poznatih funkcija, daleko su brojnije one jednadžbe za koje ne možemo napisati egzaktno rješenje. Stoga takve jednadžbe rješavamo numerički. Ponekad je čak brže i jednostavnije izračunati rješenje numeričkim putem umjesto dugotrajnim analitičkim postupkom.

Opisat ćemo nekoliko najčešćih numeričkih metoda za rješavanje običnih diferencijalnih jednadžbi (skraćeno ODJ) oblika

$$y'(x) = f(x, y(x)), \quad y \in (a, b), \tag{10.1.1}$$

uz zadani početni uvjet $y(a) = y_0$ ili uz zadani rubni uvjet $r(y(a), y(b)) = 0$, gdje je r neka zadana funkcija.

Sustav običnih diferencijalnih jednadžbi je općenitiji problem:

$$\begin{aligned} y_1' &= f_1(x, y_1, \ldots, y_n), \\ y_2' &= f_2(x, y_1, \ldots, y_n), \\ &\vdots \\ y_n' &= f_n(x, y_1, \ldots, y_n). \end{aligned}$$

Međutim, koristeći vektorsku notaciju

$$\mathbf{y} = [y_1, \ldots, y_n]^T \quad \text{i} \quad \mathbf{f} = [f_1, \ldots, f_n]^T$$

sustav pišemo u obliku analognom jednadžbi (10.1.1):

$$\mathbf{y}'(x) = \mathbf{f}(x, \mathbf{y}(x)),$$

te primijenjujemo iste numeričke metode kao za rješavanje diferencijalne jednadžbe (10.1.1) vodeći računa o tome da se umjesto skalarnih funkcija y i f javljaju vektorske funkcije $\mathbf{y}$ i $\mathbf{f}$.

Diferencijalne jednadžbe višeg reda

$$y^{(n)} = f(x, y, y', y'', \ldots, y^{(n-1)})$$

supstitucijama

$$y_1 = y, \quad y_2 = y', \quad \ldots, \quad y_n = y^{(n-1)}$$

svodimo na sustav jednadžbi prvog reda:

$$\begin{aligned}
y_1' &= y' = y_2, \\
y_2' &= y'' = y_3, \\
&\vdots \\
y_{n-1}' &= y^{(n-1)} = y_n, \\
y_n' &= y^{(n)} = f(x, y, y', y'', \ldots, y^{(n-1)}) = f(x, y_1, y_2, y_3, \ldots, y_n),
\end{aligned}$$

te i u ovom slučaju možemo koristiti metode razvijene za diferencijalnu jednadžbu (10.1.1).

I za sustav diferencijalnih jednadžbi i za jednadžbu višeg reda razlikujemo inicijalni (početni ili Cauchyjev) problem i rubni problem. U ovom poglavlju prikazat ćemo sljedeće metode za rješavanje inicijalnog problema:

- Runge–Kuttine metode,
- linearne višekoračne metode.

10.2. Inicijalni problem za običnu diferencijalnu jednadžbu. Eulerova metoda

Eulerova metoda je zasigurno najjednostavnija metoda za rješavanje inicijalnog problema za ODJ oblika

$$y' = f(x, y), \quad y(a) = y_0.$$

Metoda se zasniva na ideji da se y' u gornjoj jednadžbi zamijeni s podijeljenom razlikom

$$y'(x) = \frac{y(x+h) - y(x)}{h} + \mathcal{O}(h),$$

pa rješenje diferencijalne jednadžbe zadovoljava

$$y(x+h) = y(x) + hy'(x) + \mathcal{O}(h^2) = y(x) + hf(x, y(x)) + \mathcal{O}(h^2).$$

Zanemarivanjem kvadratnog člana u gornjem razvoju dobivamo aproksimaciju

$$y(x+h) \approx y(x) + hf(x, y(x)). \tag{10.2.1}$$

Ova formula je točnija što je h manji, tako da aproksimacija

$$y(b) \approx y(a) + (b-a)hf(a, y(a)) = y_0 + hf(a, y_0)$$

može biti jako neprecizna. Stoga interval $[a, b]$ podijelimo na n jednakih dijelova te stavimo

$$h = \frac{b-a}{n}, \quad x_i = a + ih, \quad i = 0, \ldots, n.$$

Korištenjem (10.2.1), prvo aproksimiramo rješenje u točki $x_1 = a + h$

$$y(x_1) \approx y_1 = y_0 + hf(x_0, y_0).$$

Dobivenu aproksimaciju y_1 iskoristimo za računanje aproksimacije rješenja u točki $x_2 = x_1 + h$:

$$y_2 = y_1 + hf(x_1, y_1),$$

te postupak ponavljamo sve dok ne dođemo do kraja intervala $b = x_n$.

Opisani postupak nazivamo Eulerova metoda, i možemo ga kraće zapisati rekurzijom

$$y_{i+1} = y_i + hf(x_i, y_i), \quad i = 1, \ldots, n,$$

gdje je početni uvjet y_0 zadan kao inicijalni uvjet diferencijalne jednadžbe. Dobivene vrijednosti y_i su aproksimacije rješenja diferencijalne jednadžbe u točkama x_i.

10.3. Runge–Kuttine metode

Koristeći sličnu ideju kao u Eulerovoj metodi, diferencijalnu jednadžbu

$$y' = f(x, y), \quad y(a) = y_0 \tag{10.3.1}$$

na intervalu $[a, b]$, možemo rješavati tako da podijelimo interval $[a, b]$ na n jednakih podintervala, označivši

$$h = \frac{b-a}{n}, \quad x_i = a + ih, \quad i = 0, \ldots, n.$$

Sada y_{i+1}, aproksimaciju rješenja u točki x_{i+1}, računamo iz y_i korištenjem aproksimacije oblika

$$y(x+h) \approx y(x) + h\Phi(x, y(x), h; f), \tag{10.3.2}$$

te dobivamo rekurziju:

$$y_{i+1} = y_i + h\Phi(x_i, y_i, h; f), \quad i = 0, \ldots, n-1. \tag{10.3.3}$$

Funkciju Φ nazivamo **funkcija prirasta**, a različit izbor te funkcije definira različite metode. Uočimo da je funkcija f iz diferencijalne jednadžbe (10.3.1) parametar od Φ (tj. Φ zavisi o f). Tako je npr. u Eulerovoj metodi

$$\Phi(x, y, h; f) = f(x, y).$$

Metode oblika (10.3.3) zovemo **jednokoračne metode** (jer za aproksimaciju y_{i+1} koristimo samo vrijednost y_i u prethodnoj točki x_i, tj. u jednom koraku dobijemo y_{i+1} iz y_i). Da bismo pojednostavili zapis, ubuduće ćemo f izostaviti kao argument funkcije Φ.

O odabiru funkcije Φ ovisi i točnost metode. Za očekivati je da ako izaberemo Φ tako da aproksimacija točnog rješenja $y(x + h)$ dana s (10.3.2) bude što točnija, da će točnija biti i aproksimacija y_i za $y(x_i)$ dana rekurzijom (10.3.3). Pogrešku aproksimacije (10.3.2):

$$\tau(x; h) = \Delta(x; h) - \Phi(x, y(x), h), \tag{10.3.4}$$

gdje je $y(x)$ točno rješenje diferencijalne jednadžbe (10.3.1) i

$$\Delta(x; h) = \frac{y(x + h) - y(x)}{h},$$

nazivamo **lokalna pogreška diskretizacije**.

Red metode odgovara njezinoj točnosti. Općenito, za jednokoračne metode kažemo da su reda p ako je

$$\tau(x; h) = \mathcal{O}(h^p).$$

Što je veći p metoda je točnija, a to postižemo odabirom funkcije Φ.

Pod točnošću metode podrazumijevamo ponašanje pogreške $y(x_i) - y_i$. Da bismo pojednostavnili argumentaciju, promatrat ćemo pogrešku u fiksiranoj točki b. Ako je jednokoračna metoda reda p, tada je

$$y(b) - y_n = \mathcal{O}(h^p).$$

Uočimo da je $h = (b - a)/n$ te da je y_n uvijek (za svaki n) aproksimacija za $y(b)$.

Najpoznatije jednokoračne metode su svakako Runge–Kuttine metode. Kod njih je funkcija Φ oblika

$$\Phi(x, y, h) = \sum_{j=1}^{r} \omega_j k_j(x, y, h),$$

a k_j su zadani s

$$k_j(x, y, h) = f\Big(x + c_j h, y + h \sum_{l=1}^{r} a_{jl} k_l(x, y, h, f)\Big), \quad j = 1, \ldots, r. \tag{10.3.5}$$

Broj r zovemo broj stadija Runge–Kuttine (RK) metode, i on označava koliko puta moramo računati funkciju f u svakom koraku.

Različit izbor koeficijenata ω_j, c_j i a_{jl} definira različite metode. Ovi koeficijenti se najčešće biraju tako da red metode bude što je moguće veći. Iz izraza (10.3.5) vidimo da se k_j nalazi na lijevoj i na desnoj strani jednadžbe, tj. zadan je implicitno te govorimo o **implicitnoj** Runge–Kuttinoj metodi. U praksi se najviše koriste metode gdje je $a_{jl} = 0$ za $l \geq j$. Tada k_j možemo izračunati preko $k_1, \ldots, k_{j-1}$, tj. funkcije k_j su zadane eksplicitno. Takve RK metode nazivamo **eksplicitnima**. Nadalje, obično se dodaje uvjet

$$\sum_{l=1}^{r} a_{jl} = c_j.$$

Ovaj izvor koeficijenata c_j je detaljnije objašnjen u poglavlju 10.3.1..

Primjer odabira koeficijenata prikazat ćemo na RK metodi s dva stadija:

$$\begin{aligned}
\Phi(x, y, h) &= \omega_1 k_1(x, y, h) + \omega_2 k_2(x, y, h), \\
k_1(x, y, h) &= f(x, y), \\
k_2(x, y, h) &= f(x + ah, y + ahk_1).
\end{aligned}$$

Razvojem k_2 u Taylorov red po varijabli h dobivamo

$$k_2(x, y, h) = f + h(f_x a + f_y af) + \frac{h^2}{2}(f_{xx}a^2 + 2f_{xy}a^2 f + f_{yy}a^2 f^2) + \mathcal{O}(h^3),$$

gdje su f_x i f_y prve parcijalne deivacije funkcije $f = f(x, y)$ po x, odnosno y, a f_{xx}, f_{xy} i f_{yy} odgovarajuće druge parcijalne derivacije. Razvoj rješenja diferencijalne jednadžbe $y(x)$ ima oblik

$$y(x+h) = y(x) + hf + \frac{h^2}{2}(f_x + f_y f) + \frac{h^3}{6}[f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_y(f_x + f_y f)] + \mathcal{O}(h^4).$$

Ovdje smo iskoristili da je $y(x)$ rješenja diferencijalne jednadžbe:

$$y'(x) = f(x, y) = f,$$

te pravila za deriviranje

$$\begin{aligned}
y''(x) &= f_x + f_y f, \\
y'''(x) &= f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_y(f_x + f_y f).
\end{aligned}$$

Sada je lokalna pogreška diskretizacije jednaka

$$\begin{aligned}
\frac{y(x+h) - y(x)}{h} - \Phi(x, y(x), h) &= \frac{y(x+h) - y(x)}{h} - (\omega_1 k_1(x, y, h) + \omega_2 k_2(x, y, h)) \\
&= (1 - \omega_1 - \omega_2) f + h(f_x + f_y f)\left(\frac{1}{2} - \omega_2 a\right) \\
&\quad + h^2\left[(f_{xx} + 2f_{xy}f + f_{yy}f^2) \cdot \left(\frac{1}{6} - \frac{\omega_2 a^2}{2}\right) + \frac{1}{6} f_y (f_x + f_y f)\right] \\
&\quad + \mathcal{O}(h^3).
\end{aligned}$$

Da bi metoda bila reda 1 koeficijente treba odabrati tako da se poništi prvi član u gornjem razvoju:

$$1 - \omega_1 - \omega_2 = 0.$$

Ukoliko je zadovoljeno i

$$\frac{1}{2} - \omega_2 a = 0$$

metoda će biti reda 2. Uvođenjem slobodnog koeficijenta t rješenje ove dvije jednadžbe možemo napisati u obliku:

$$\omega_2 = t \neq 0, \quad \omega_1 = 1 - t, \quad a = \frac{1}{2t}.$$

Uočimo da t ne možemo odabrati tako da poništimo i član uz h^2 tako da metoda bude reda 3. Ukoliko je $\omega_2 = 0$, radi se o metodi s jednim stadijem, i to upravo o Eulerovoj metodi.

Za $t = 1/2$ dobivamo Heunovu metodu:

$$\begin{aligned} \Phi &= \frac{1}{2}\,(k_1 + k_2), \\ k_1 &= f(x, y), \\ k_2 &= f(x + h, y + hk_1), \end{aligned}$$

dok se za $t = 1$ dobiva modificirana Eulerova metoda:

$$\Phi = f\Big(x + \frac{h}{2}, y + \frac{h}{2}\, f(x, y)\Big).$$

Najraširenije su metode sa četiri stadija. Odgovarajuće jednadžbe koje moraju zadovoljavati koeficijenti RK-4 metoda su:

$$\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1, \tag{10.3.6}$$

$$\omega_2 c_2 + \omega_3 c_3 + \omega_4 c_4 = \frac{1}{2}, \tag{10.3.7}$$

$$\omega_2 c_2^2 + \omega_3 c_3^2 + \omega_4 c_4^2 = \frac{1}{3}, \tag{10.3.8}$$

$$\omega_3 c_2 a_{32} + \omega_4 (c_2 a_{42} + c_3 a_{43} = \frac{1}{6}, \tag{10.3.9}$$

$$\omega_2 c_2^3 + \omega_3 c_3^3 + \omega_4 c_4^3 = \frac{1}{4}, \tag{10.3.10}$$

$$\omega_3 c_2^2 a_{32} + \omega_4 (c_2^2 a_{42} + c_3^2 a_{43}) = \frac{1}{12}, \tag{10.3.11}$$

$$\omega_3 c_2 c_3 a_{32} + \omega_4 (c_2 a_{42} + c_3 a_{43}) c_4 = \frac{1}{8}, \tag{10.3.12}$$

$$\omega_4 c_2 a_{32} a_{43} = \frac{1}{24}, \tag{10.3.13}$$

gdje je

$$c_1 = 0, \qquad c_2 = a_{21},$$
$$c_3 = a_{31} + a_{32}, \qquad c_4 = a_{41} + a_{42} + a_{43}.$$

Uvjet (10.3.6) treba biti zadovoljen da bi metoda bila reda 1, uvjet (10.3.7) za red 2, uvjeti (10.3.8)–(10.3.9) za red 3, dok za red 4 trebaju biti ispunjeni i uvjeti (10.3.10)–(10.3.13). Ukupno imamo 10 koeficijenata i 8 jednadžbi ukoliko je metoda reda 4. Za metodu s tri stadija uvrštavanjem

$$c_4 = a_{41} = a_{42} = a_{43} = \omega_4 = 0$$

dobivamo 9 koeficijenata i 7 jednadžbi. Metoda s četiri stadija može postići najviše red četiri, tj. ne možemo dva stupnja slobode iz sustava jednadžbi iskoristiti da red metode podignemo na pet.

Općenito, za metode s jednim, dva, tri i četiri stadija najveći mogući red metode odgovara broju stadija. Za metode s 5, 6 i 7 stadija najveći mogući red je 4, 5 i 6, dok je za metode s 8 i više stadija najveći mogući red barem za dva manji od broja stadija. To je razlog zašto su metode s četiri stadija najpopularnije. Red je 4, a da bismo postigli red 5 trebamo povećati broj stadija barem za dva, što povećava složenost metode.

Evo nekoliko primjera RK-4 metoda. Najpopularnija je "klasična" Runge–Kutta metoda, koja se u literaturi najčešće naziva Runge–Kutta ili RK-4 metoda (iako je to samo jedna u nizu Runge–Kutta metoda):

$$\begin{aligned}
\Phi &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),\\
k_1 &= f(x, y),\\
k_2 &= f\Big(x + \frac{h}{2}, y + \frac{h}{2}k_1\Big),\\
k_3 &= f\Big(x + \frac{h}{2}, y + \frac{h}{2}k_2\Big),\\
k_4 &= f\Big(x + h, y + hk_3\Big).
\end{aligned}$$

Spomenimo još i 3/8-sku metodu:

$$\begin{aligned}
\Phi &= \frac{1}{8}(k_1 + 3k_2 + 3k_3 + k_4),\\
k_1 &= f(x, y),\\
k_2 &= f\Big(x + \frac{h}{3}, y + \frac{h}{3}k_1\Big),\\
k_3 &= f\Big(x + \frac{2}{3}\,h, y - \frac{h}{3}\,k_1 + hk_2\Big),\\
k_4 &= f(x + h, y + h(k_1 - k_2 + k_3))
\end{aligned}$$

i Gillovu metodu:

$$\begin{aligned}
\Phi &= \frac{1}{6}\Big(k_1 + (2-\sqrt{2})k_2 + (2+\sqrt{2})k_3 + k_4\Big),\\
k_1 &= f(x,y),\\
k_2 &= f\Big(x+\frac{h}{2}, y+\frac{h}{2}\,k_1\Big),\\
k_3 &= f\left(x+\frac{h}{2}, y+h\frac{\sqrt{2}-1}{2}\,k_1 + h\frac{2-\sqrt{2}}{2}\,k_2\right),\\
k_4 &= f\left(x+h, y-h\frac{\sqrt{2}}{2}\,k_2 + h\frac{2+\sqrt{2}}{2}\,k_3\right).
\end{aligned}$$

10.3.1. Još o koeficijentima za Runge–Kuttine metode

U prijašnjem smo poglavlju pokazali da za RK-2 i RK-4 metode treba vrijediti

$$\sum_j \omega_j = 1$$

da bi ove bile konzistentne s redom većim od 1. To vrijedi i općenito za sve RK metode, što ćemo pokazati u sljedećem teoremu.

Teorem 10.3.1 *Runge–Kuttina metoda sa s stadija ima red konzistencije veći ili jednak 1 ako i samo ako je*

$$\sum_{j=1}^{s} \omega_j = 1.$$

Dokaz. Teorem srednje vrijednosti daje nam ocjene

$$y(x+h) = y(x) + hy'(x) + \mathcal{O}(h^2)$$

i

$$k_j = f\Big(x + c_j h, y + h\sum_l a_{jl}k_l\Big) = f(x,y) + \mathcal{O}(h),$$

pa lokalna pogreška diskretizacije

$$\tau(x;h) = \frac{y(x+h)-y(x)}{h} - \sum_j \omega_j k_j$$

zadovoljava

$$\tau(x;h) = y'(x) + \mathcal{O}(h) - \sum_j \omega_j[f(x,y) + \mathcal{O}(h)].$$

Budući da je y rješenje diferencijalne jednadžbe $y' = f(x,y)$, vrijedi

$$\tau(x;h) = f(x,y) - \sum_j \omega_j f(x,y) + \mathcal{O}(h) = f(x,y)\Big(1 - \sum_j \omega_j\Big) + \mathcal{O}(h),$$

odakle lagano slijedi tvrdnja teorema. ■

Slijedeća zanimljivost vezana je uz određivanje koeficijenata c_j. U definiciji metode smo spomenuli da je uobičajeni izbor

$$c_j = \sum_l a_{jl}.$$

Međutim, ostaje pitanje da li možemo povećati red konzistencije drugačijim izborom koeficijenata c_j. Odgovor je ne, i to nam pokazuje sljedeći teorem.

Teorem 10.3.2 *Neka je RK metoda zadana s*

$$y_{i+1} = y_i + h\sum_{j=1}^{s} \tilde{k}_j, \qquad \tilde{k}_j = f\Big(x + \tilde{c}_j, y + h\sum_{l=1}^{s} a_{jl}\tilde{k}_l\Big)$$

reda konzistencije $\tilde{p}$, te neka je p red konzistencije metode

$$y_{i+1} = y_i + h\sum_{j=1}^{s} k_j, \qquad k_j = f\Big(x + c_j, y + h\sum_{l=1}^{s} a_{jl}k_l\Big),$$

gdje je $c_j = \sum_{l=1}^{s} a_{jl}$. Tada je $p \geq \tilde{p}$.

Dokaz. Neka je y rješenje diferencijalne jednadžbe

$$y' = f(x, y). \tag{10.3.14}$$

Tada je

$$\mathbf{y} = \begin{bmatrix} y \\ x \end{bmatrix}$$

rješenje jednadžbe

$$\mathbf{y}' = \begin{bmatrix} y' \\ 1 \end{bmatrix} = \begin{bmatrix} f(x,y) \\ 1 \end{bmatrix} = \mathbf{f}(\mathbf{y}).$$

Neka je $\tilde{\mathbf{y}}_i$ aproksimacija za $\mathbf{y}(x_i)$ dobivena metodom (10.3.2). S $\tilde{y}_i$ i $\tilde{x}_i$ označit ćemo komponente vektora $\tilde{\mathbf{y}}_i$:

$$\tilde{\mathbf{y}}_i = \begin{bmatrix} \tilde{y}_i \\ \tilde{x}_i \end{bmatrix}.$$

Budući da je red metode $\tilde{p}$, vrijedi

$$\|\tilde{\mathbf{y}}_i - \mathbf{y}(x_i)\| = \mathcal{O}(h^{\tilde{p}}).$$

Uočimo da je

$$\begin{aligned}\tilde{\mathbf{k}}_j = \mathbf{f}\Big(\mathbf{y} + \sum_l a_{jl}\tilde{\mathbf{k}}_l\Big) &= \begin{bmatrix} \tilde{k}_j \\ 1 \end{bmatrix} = \begin{bmatrix} f\Big(x + h\sum_l a_{jl}, y + h\sum_l a_{jl}\tilde{k}_l\Big) \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} f\Big(x + c_j h, y + h\sum_l a_{jl}\tilde{k}_l\Big) \\ 1 \end{bmatrix},\end{aligned}$$

gdje $\tilde{k}_l$ označava prvu komponentu vektora $\tilde{\mathbf{k}}_l$. Uočimo da je $\tilde{k}_j = k_j$, a k_j je definiran metodom (10.3.2), te vrijedi

$$\tilde{\mathbf{y}}_{i+1} = \tilde{\mathbf{y}}_i + h\sum_j \omega_j \tilde{\mathbf{k}}_j = \begin{bmatrix} \tilde{y}_i \\ \tilde{x}_i \end{bmatrix} + h\sum_j \omega_j \begin{bmatrix} k_j \\ 1 \end{bmatrix} = \begin{bmatrix} y_i + h\sum_j \omega_j k_j \\ \tilde{x}_i + h\sum_j \omega_j \end{bmatrix}.$$

Iz $\sum_j \omega_j = 1$ izlazi da je $\tilde{x}_{i+1} = \tilde{x}_i + h$, a upotrebom matematičke indukcije zaključimo da je $\tilde{x}_i = x_i = x_0 + ih$. Sada je

$$\begin{bmatrix} \tilde{y}_{i+1} \\ x_{i+1} \end{bmatrix} = \begin{bmatrix} \tilde{y}_i + h\sum_j \omega_j k_j \\ x_{i+1} \end{bmatrix}.$$

Ukoliko s y_i označimo aproksimaciju rješenja jednadžbe $y' = f(x,y)$ dobivenog metodom (10.3.2), zbog istih početnih uvjeta slijedi da je $y_i = \tilde{y}_i$. Ova činjenica povlači da je i

$$\tilde{\mathbf{y}}_i - \mathbf{y}(x_i) = \begin{bmatrix} \tilde{y}_i \\ \tilde{x}_i \end{bmatrix} - \begin{bmatrix} y(x_i) \\ x_i \end{bmatrix} = \begin{bmatrix} y_i - y(x_i) \\ 0 \end{bmatrix}$$

te za bilo koju od uobičajenih normi $\|\ \|_1$, $\|\ \|_2$ ili $\|\ \|_\infty$ vrijedi

$$\|y_i - y(x_i)\| = \|\tilde{\mathbf{Y}}_i - \mathbf{Y}(x_i)\| = \mathcal{O}(h^{\tilde{p}}),$$

pa je metoda (10.3.2) reda barem $\tilde{p}$, tj. vrijedi $p \geq \tilde{p}$. ■

10.3.2. Konvergencija jednokoračnih metoda

U ovom odjeljku promatramo ponašanje konvergencije približnog rješenja y_i dobivenog jednokoračnom metodom. Na početku, definirajmo prostor $F_n(a,b)$ kao prostor funkcija s trake $(a,b) \times \mathbb{R}^{d-1}$ u $\mathbb{R}$ za koje su sve parcijalne derivacije do uključivo n-tog reda neprekidne i ograničene.

U daljnjem tekstu pretpostaviti ćemo da je $f \in F_1(a,b)$, a s y ćemo označiti egzaktno rješenje inicijalnog problema

$$y' = f(x,y), \quad y(x_0) = y_0, \quad x_0 \in [a,b], \quad y_0 \in \mathbb{R}.$$

Uočimo da pretpostavka $f \in F_1(a,b)$ povlači egzistenciju i jedinstvenost rješenja y na intervalu $[a,b]$. Neka $\Phi(x,y;h)$ definira jednokoračnu metodu

$$y_{i+1} = y_i + h\Phi(x_i, y_i; h), \quad i = 0,1,2,\ldots$$

gdje je $x_{i+1} = x_i + h$. Zanima nas ponašanje pogreške

$$e_i = y_i - y(x_i).$$

Za fiksirani $x \in [a, b]$ definiramo korak

$$h_n = \frac{x - x_0}{n}$$

i **globalnu pogrešku diskretizacije**

$$e(x; h_n) = y_n - y(x).$$

Sada za svaki $n \in \mathbb{N}$ vrijedi $x_n = x$, te možemo promatrati globalnu pogrešku diskretizacije kada $n \to \infty$.

Definicija 10.3.1 *Jednokoračna metoda je* **konvergentna** *ako*

$$\lim_{n\to\infty} e(x; h_n) = 0$$

za sve $x \in [a, b]$ *i sve* $f \in F_1(a, b)$.

Pokazat ćemo da su metode reda $p > 0$ konvergentne, i štoviše, da vrijedi

$$e(x; h_n) = \mathcal{O}(h_n^p).$$

Red globalne pogreške diskretizacije je dakle jednak redu lokalne pogreške diskretizacije. Prvo ćemo dokazati sljedeću lemu.

Lema 10.3.1 *Ako brojevi* ξ_i *zadovoljavaju ocjenu oblika*

$$|\xi_{i+1}| \le (1 + \delta)|\xi_i| + B, \quad \delta > 0, \quad B \ge 0, \quad i = 0, 1, 2, \ldots$$

tada je

$$|\xi_n| \le e^{n\delta}|\xi_0| + \frac{e^{n\delta} - 1}{\delta} B.$$

Dokaz. Iz pretpostavke direktno slijedi

$$\begin{aligned}
|\xi_1| &\le (1 + \delta)|\xi_0| + B, \\
|\xi_2| &\le (1 + \delta)^2|\xi_0| + B(1 + \delta) + B, \\
&\vdots \\
|\xi_n| &\le (1 + \delta)^n|\xi_0| + B[1 + (1 + \delta) + (1 + \delta)^2 + \cdots + (1 + \delta)^{n-1}] \\
&= (1 + \delta)^n|\xi_0| + B\frac{(1 + \delta)^n - 1}{\delta} \\
&\le e^{n\delta}|\xi_0| + B\frac{e^{n\delta} - 1}{\delta},
\end{aligned}$$

jer je $0 < 1 + \delta < e^\delta$ za $\delta > -1$. ■

Pomoću ove leme dokazujemo sljedeći, glavni, teorem.

Teorem 10.3.3 *Za $x_0 \in [a,b]$, $y_0 \in \mathbb{R}$, promatramo inicijalni problem*

$$y' = f(x,y), \quad y(x_0) = y_0,$$

koji ima jedinstveno rješenje $y(x)$. Neka je funkcija Φ neprekidna na

$$G = \{(x,y,h) \mid x \in [a,b],\ |y - y(x)| \le \gamma,\ |h| \le h_0\},$$

za $h_0 > 0$, $\gamma > 0$ i neka postoje pozitivne konstante M i N takve da je

$$|\Phi(x,y_1;h) - \Phi(x,y_2;h)| \le M|y_1 - y_2|$$

za sve $(x,y_i,h) \in G$, $i = 1,2$, i

$$|\tau(x;h)| = |\Delta(x;h) - \Phi(x,y(x);h)| \le N|h|^p, \quad p > 0$$

za sve $x \in [a,b]$, $h \le h_0$. Tada postoji $\bar{h}$, $0 < \bar{h} \le h_0$, takav da globalna pogreška diskretizacije zadovoljava

$$|e(x;h_n)| \le |h_n|^p N \, \frac{e^{M|x-x_0|} - 1}{M}$$

za sve $x \in [a,b]$ i $h_n = (x - x_0)/n$, $n = 1,2,\ldots$, uz $|h_n| \le \bar{h}$. Ako je $\gamma = \infty$, tada je $\bar{h} = h_0$.

Dokaz. Funkcija

$$\widetilde{\Phi}(x,y;h) = \begin{cases} \Phi(x,y;h), & \text{za } (x,y,h) \in G, \\ \Phi(x,y(x)+\gamma;h), & \text{za } x \in [a,b],\ |h| \le h_0,\ y \ge y(x)+\gamma, \\ \Phi(x,y(x)-\gamma;h), & \text{za } x \in [a,b],\ |h| \le h_0,\ y \le y(x)-\gamma \end{cases}$$

je očito neprekidna na $\tilde{G} = \{(x,y,h) \mid x \in [a,b],\ y \in \mathbb{R},\ |h| \le h_0\}$ i zadovoljava uvjet

$$|\widetilde{\Phi}(x,y_1;h) - \widetilde{\Phi}(x,y_2;h)| \le M|y_1 - y_2| \tag{10.3.15}$$

za sve $(x,y_i,h) \in \tilde{G}$, $i = 1,2$. Zbog $\widetilde{\Phi}(x,y(x);h) = \Phi(x,y(x);h)$ također vrijedi

$$|\Delta(x;h) - \widetilde{\Phi}(x,y(x);h)| \le N|h|^p, \quad \text{za } x \in [a,b], \quad |h| \le h_0. \tag{10.3.16}$$

Neka jednokoračna metoda generirana s $\widetilde{\Phi}$ definira aproksimacije $\tilde{y}_i$ za $y(x_i)$, $x_i = x_0 + ih$:

$$\tilde{y}_{i+1} = \tilde{y}_i + h\widetilde{\Phi}(x_i, \tilde{y}_i; h).$$

Zbog

$$y(x_{i+1}) = y(x_i) + h\Delta(x_i;h),$$

za pogrešku $\tilde{e}_i = \tilde{y}_i - y(x_i)$, oduzimanjem, dobivamo rekurzivnu formulu

$$\tilde{e}_{i+1} = \tilde{e}_i + h[\widetilde{\Phi}(x_i,\tilde{y}_i;h) - \widetilde{\Phi}(x_i,y(x_i);h)] + h[\widetilde{\Phi}(x_i,y(x_i);h) - \Delta(x_i;h)].$$

No, sada iz (10.3.15) i (10.3.16) slijedi

$$|\widetilde{\Phi}(x_i,\tilde{y}_i;h)-\widetilde{\Phi}(x_i,y(x_i);h)\le M|\tilde{y}_i-y(x_i)|=M|\tilde{e}_i|,$$
$$|\widetilde{\Phi}(x_i,y(x_i);h)-\Delta(x_i;h)|\le N|h|^p,$$

te dobivamo rekurzivnu ocjenu

$$|\tilde{e}_{i+1}|\le(1+|h|M)|\tilde{e}_i|+N|h|^{p+1}.$$

Korištenjem $\tilde{e}_0=\tilde{y}_0-y(x_0)=0$, iz leme slijedi

$$|\tilde{e}_k|\le N|h|^p\frac{e^{M|x_k-x_0|}-1}{M}.$$

Neka je sada $x\in[a,b]$, $x\neq x_0$, fiksiran i $h=h_n=(x-x_0)/n$, $n>0$. Tada zbog $x_n=x_0+nh=x$ i zbog $\tilde{e}(x;h_n)=\tilde{e}_n$ slijedi

$$|\tilde{e}_(x;h_n)|\le N|h|^p\frac{e^{M|x-x_0|}-1}{M}$$

za sve $x\in[a,b]$ i h_n za koje je $|h_n|\le h_0$. Budući da je $|x-x_0|\le|b-a|$ i $\gamma>0$, postoji $\bar{h}$, $0<\bar{h}\le h_0$ takav da je $|\tilde{e}(x;h_n)|\le\gamma$ za sve $x\in[a,b]$, $|h_n|\le\bar{h}$, tj. za jednokoračnu metodu generiranu s Φ:

$$y_{i+1}=y_i+h\Phi(x_i,y_i;h)$$

imamo za $|h|\le\bar{h}$, zbog definicije za $\widetilde{\Phi}$,

$$\tilde{y}_i=y_i,\quad \tilde{e}_i=e_i\quad\text{i}\quad\widetilde{\Phi}(x_i,\tilde{y}_i;h)=\Phi(x_i,y_i;h).$$

Tvrdnja teorema dakle slijedi za sve $x\in[a,b]$ i sve $h_n=(x-x_0)/n$, $n=1,2,\ldots$, uz $|h_n|\le\bar{h}$. ■

Iz prethodnog teorema posebno slijedi da je metoda reda $p>0$, koja u okolini rješenja zadovoljava Lipschitzov uvjet, konvergentna. Uočimo da je ovaj uvjet ispunjen ako $\dfrac{\partial}{\partial y}\Phi(x,y;h)$ postoji i neprekidna je u domeni G danoj u teoremu. Teorem, također, daje i gornju ogradu za globalnu pogrešku diskretizacije, koja u principu može biti izračunata ako znamo M i N. To je u praksi nepraktično. Tako je za Eulerovu metodu

$$N\approx\frac{1}{2}\,|f_x(x,y(x))+f_y(x,y(x))f(x,y(x))|$$

i

$$M\approx\Big|\frac{\partial\Phi}{\partial y}|=|f_y(x,y(x))|,$$

dok za RK-4 metode treba ocijeniti četvrte derivacije funkcije f.

10.3.3. Runge–Kutta–Fehlbergove metode. Određivanje koraka integracije.

Iako smo u prošlom potpoglavlju pretpostavili da je korak integracije h konstantan tijekom cijelog postupka rješavanja diferencijalne jednadžbe, očito je da se h može mijenjati u svakom koraku integracije, pa jednokoračnu metodu možemo pisati u obliku:

$$y_{i+1} = y_i + h_i\Phi(x_i, y_i, h_i).$$

Prvo ćemo pokazati kako se određuje duljina koraka h_i tako da bude postignuta neka unaprijed zadana točnost ε.

Neka su s Φ i $\bar{\Phi}$ zadane dvije metode reda p i $p+1$. Tada računamo aproksimacije

$$\begin{aligned} y_{i+1} &= y_i + h_i\Phi(x_i, y_i, h_i), \\ \bar{y}_{i+1} &= y_i + h_i\bar{\Phi}(x_i, y_i, h_i). \end{aligned}$$

Iz (10.3.4) slijedi da je:

$$\begin{aligned} y(x_i + h_i) &= y(x_i) + h_i\Phi(x_i, y(x_i), h_i) + C(x_i)h_i^{p+1} + \mathcal{O}(h_i^{p+2}), \\ y(x_i + h_i) &= y(x_i) + h_i\bar{\Phi}(x_i, y(x_i), h_i) + \mathcal{O}(h_i^{p+2}). \end{aligned}$$

Cilj je da pogreška u i-tom koraku bude manja od ε. Stoga ćemo pretpostaviti da je aproksimacija y_i za $y(x_i)$ točna, tj. $y_i = y(x_i)$. Sada oduzimanjem gornje dvije jednadžbe slijedi

$$h_i[\Phi(x_i, y_i, h_i) - \bar{\Phi}(x_i, y_i, h_i)] = C(x_i)h_i^{p+1} + \mathcal{O}(h_i^{p+2}). \tag{10.3.17}$$

Iz prve dvije jednakosti oduzimanjem slijedi

$$h_i[\Phi(x_i, y_i, h_i) - \bar{\Phi}(x_i, y_i, h_i)] = \bar{y}_{i+1} - y_{i+1},$$

te uvrštavanjem u (10.3.17) dobivamo

$$y_{i+1} - \bar{y}_{i+1} = C(x_i)h_i^{p+1} + \mathcal{O}(h_i^{p+2}).$$

Zanemarivanjem viših članova u razvoju pogreške, dobivamo

$$y_{i+1} - \bar{y}_{i+1} \approx C(x_i)h_i^{p+1}$$

i

$$y_i - \bar{y}_i \approx C(x_{i-1})h_{i-1}^{p+1}.$$

Uz pretpostavku da se član $C(x)$ u pogrešci ne mijenja brzo, tj. $C(x_i) \approx C(x_{i-1})$, uvjet da pogreška u i-tom koraku bude manja od ε sada glasi:

$$\varepsilon \geq |y(x_{i+1}) - y_{i+1}| \approx |C(x_i)h_i^{p+1}| \approx |C(x_{i-1})h_i^{p+1}| \approx \left|\frac{\bar{y}_i - y_i}{h_{i-1}^{p+1}}\right| h_i^{p+1},$$

odnosno

$$h_i^{p+1} \leq h_{i-1}^{p+1} \frac{\varepsilon}{|\bar{y}_i - y_i|}.$$

Iz ovoga slijedi da za novi korak trebamo izabrati

$$h_i = h_{i-1} \sqrt[p+1]{\frac{\varepsilon}{|\bar{y}_i - y_i|}}.$$

Ukoliko je prethodni korak bio uspješan, tada je zadovoljeno

$$|\bar{y}_i - y_i| \leq \varepsilon,$$

te je stoga $h_i \geq h_{i+1}$. Ako gornja nejednakost ne vrijedi, $(i-1)$-vi korak treba ponoviti uz manji h_{i-1}. To se obično radi samo ako je nejednakost značajnije narušena, npr. ako je $|\bar{y}_i - y_i| > 2\varepsilon$.

Izbor koraka možemo modificirati, tako da zahtijevamo da je pogreška u svakom koraku proporcionalna koraku integracije, tj. da je manja od $h_i\varepsilon$. Uvrštavanjem ovog zahtjeva dobivamo izbor:

$$h_i = h_{i-1} \sqrt[p]{\frac{\varepsilon h_{i-1}}{|\bar{y}_i - y_i|}}.$$

Sljedeća modifikacija koristi korigirajući faktor α:

$$h_i = \alpha h_{i-1} \sqrt[p]{\frac{\varepsilon h_{i-1}}{|\bar{y}_i - y_i|}},$$

koji služi da ispravi pogrešku nastalu odbacivanjem viših članova u ocjeni pogreške. Obično je $\alpha = 0.9$.

Prikazani izbor koraka vrijedi za bilo koji par jednokoračnih metoda reda p i $p+1$. Primjena Runge–Kutta metoda zahtijevala bi jednu metodu sa s stadija i jednu sa $s+1$ stadija, što općenito znači da bismo u svakom koraku funkciju f iz diferencijalne jednadžbe trebali računati $2s+1$ puta. Postupak se može pojednostavniti ako prvih s stadija $k_1, \ldots, k_s, k_{s+1}$ korištenih za računanje funkcije prirasta $\bar{\Phi}$ iskoristimo za računanje funkcije Φ:

$$\Phi(x, y, h) = \sum_{i=1}^{s} \omega_i k_i(x, y, h),$$

$$\bar{\Phi}(x, y, h) = \sum_{i=1}^{s+1} \bar{\omega}_i k_i(x, y, h).$$

Sada u svakom koraku funkciju f računamo samo $s+1$ puta.

Ovu ideju ćemo ilustrirati na paru Runge–Kutta metoda reda 2 i 3. Promatramo metode s 3 i 4 stadija:

$$\Phi(x,y,h)=\sum_{i=1}^{3}\omega_i k_i(x,y,h),$$
$$\bar{\Phi}(x,y,h)=\sum_{i=1}^{4}\bar{\omega}_i k_i(x,y,h).$$

Da bi metoda definirana s $\bar{\Phi}$ bila reda 3 trebaju biti zadovoljeni uvjeti (10.3.6)–(10.3.9), što je ukupno 4 jednadžbe i 10 koeficijenata. Nadalje, metoda reda 2 s 3 stadija ima 3 dodatna koeficijenta (ω_1, ω_2 i ω_3) i treba zadovoljavati 2 dodatna uvjeta (10.3.6)–(10.3.7). Sveukupno, 13 koeficijenata u ovom paru metoda treba zadovoljavati 6 uvjeta. Preostalih 7 stupnjeva slobode iskoristit ćemo za smanjivanje broja računanja funkcije f. Naime, zahtijevat ćemo da $k_4(x_i,y_i,h_i,f)$, zadnji stadij iz računanja $\bar{\Phi}$ u i-tom koraku, iskoristimo kao $k_1(x_{i+1},y_{i+1},h_{i+1},f)$, prvi stadij u $(i+1)$-om koraku:

$$\begin{aligned} f(x+c_4h_i, y_i+h_i(a_{41}k_1+a_{42}k_2+a_{43}k_3) &= f(x_{i+1},y_{i+1}) \\ &= f(x_i+h_i, y_i+h_i\Phi(x_i,y_i,h_i,f)) \\ &= f(x_i+h_i, y_i+h_i(\omega_1k_1+\omega_2k_2+\omega_3k_3)). \end{aligned}$$

Odavde slijede dodatna 3 uvjeta:

$$a_{41}=\omega_1,\quad a_{42}=\omega_2,\quad a_{43}=\omega_3.$$

Uvjet $c_4=1$ automatski je ispunjen zbog

$$\omega_1+\omega_2+\omega_3=1.$$

Jedno od rješenja ovog sustava jednadžbi je prikazano u sljedećoj tablici.

		a_{ij}				
i	c_i	$j=1$	$j=2$	$j=3$	ω_i	$\bar{\omega}_i$
1	0				$\frac{214}{891}$	$\frac{533}{2106}$
2	$\frac{1}{4}$	$\frac{1}{4}$			$\frac{1}{33}$	0
3	$\frac{27}{40}$	$-\frac{189}{800}$	$\frac{729}{800}$		$\frac{650}{891}$	$\frac{800}{1053}$
4	1	$\frac{214}{891}$	$\frac{1}{33}$	$\frac{650}{891}$		$-\frac{1}{78}$

10.4. Linearne višekoračne metode

Kod jednokoračnih metoda je za aproksimaciju y_{i+1} u točki x_{i+1} bilo potrebno poznavanje samo aproksimacije y_i u točki x_i.

Promatrajmo ponovno diferencijalnu jednadžbu

$$y' = f(x, y), \quad y(x_0) = y_0.$$

Integracijom, te primjenom formule srednje točke za aproksimaciju integrala, slijedi da je

$$y(x_{i+1}) - y(x_{i-1}) = \int_{x_{i-1}}^{x_{i+1}} y'(x)\,dx = \int_{x_{i-1}}^{x_{i+1}} f(x, y(x))\,dx \approx 2hf(x_i, y(x_i)). \qquad (10.4.1)$$

Gornja formula vodi na rekurzivno definiranu aproksimaciju

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i).$$

U ovoj metodi za određivanje vrijednosti y_{i+1} trebamo poznavati prethodne vrijednosti y_i i y_{i-1}, a budući da je se radi o dvije točke, govorimo o dvokoračnoj metodi. Aproksimacija y_{i+1} zadana je eksplicitno s izrazom na desnoj strani, pa govorimo o eksplicitnoj metodi.

Ako umjesto formule srednje točke pri računanju integrala u (10.4.1) primijenimo Simpsonovu formulu, dobivamo drugu aproksimaciju:

$$\begin{aligned} y(x_{i+1}) - y(x_{i-1}) &= \int_{x_{i-1}}^{x_{i+1}} f(x, y(x))\,dx \\ &\approx \frac{h}{3}\left[f(x_{i-1}, y(x_{i-1})) + 4f(x_i, y(x_i)) + f(x_{i+1}, y(x_{i+1}))\right], \end{aligned}$$

što vodi na metodu

$$y_{i+1} = y_{i-1} + \frac{h}{3}\left[f(x_{i-1}, y_{i-1}) + 4f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right].$$

Ovdje se y_{i+1} javlja i na lijevoj strani i na desnoj strani kao argument, općenito nelinearne, funkcije f. Dakle, y_{i+1} je zadan implicitno, pa govorimo o implicitnoj dvokoračnoj metodi. Uočimo da gornjim formulama ne možemo odrediti y_1, pa za njegovo određivanje treba upotrebiti jednu od jednokoračnih metoda.

Općenito, linearne višekoračne metode su oblika

$$\sum_{j=0}^{r} \alpha_j y_{i+1-j} = h \sum_{j=0}^{r} \beta_j f_{i+1-j},$$

gdje je $f_k = f(x_k, y_k)$, $\alpha_0 \neq 0$ i $|\alpha_r| + |\beta_r| \neq 0$. Ovu metodu zovemo r-koračna metoda. Ukoliko je $\beta_0 = 0$ metoda je eksplicitna, a za $\beta_0 \neq 0$ metoda je implicitna.

Uočimo da prikaz višekoračne metode pomoću koeficijenata α_j i β_j nije jedinstven jer npr. koeficijenti $2\alpha_0, \ldots, 2\alpha_r$ i $2\beta_0, \ldots, 2\beta_r$ definiraju istu metodu. Često se koristi normalizacija $\alpha_0 = 1$ te je zapis metode oblika:

$$y_{i+1} + \sum_{j=1}^{r} \alpha_j y_{i+1-j} = \beta_0 f(x_{i+1}, y_{i+1}) + h \sum_{j=1}^{r} \beta_j f_{i+1-j}. \tag{10.4.2}$$

Primjenom različitih integracijskih metoda možemo dobiti cijeli niz višekoračnih metoda. Integracijom jednadžbe $y'(x) = f(x, y(x))$ na nekom zadanom intervalu $[x_{p-j}, x_{p+k}]$ dobivamo

$$y(x_{p+k}) - y(x_{p-j}) = \int_{x_{p-j}}^{x_{p+k}} f(t, y(t))\, dt.$$

Ukoliko podintegralnu funkciju $f(t, y(t))$ zamijenimo interpolacijskim polinomom P_q stupnja q koji interpolira $f(t, y(t))$ u točkama x_i, tj. ako je

$$P_q(x_i) = y'(x_i) = f(x_i, y(x_i)), \quad i = p, p-1, \ldots, p-q,$$

korištenjem interpolacijskog polinoma u Lagrangeovoj formi

$$P_q(x) = \sum_{i=0}^{q} f(x_i, y(x_i))\ell_i(x), \quad \ell_i(x) = \prod_{\substack{l=0 \\ l\neq i}}^{q} \frac{x - x_{p-l}}{x_{p-i} - x_{p-l}},$$

izraz (10.4.2) prelazi u

$$y(x_{p+k}) - y(x_{p-j}) \approx \sum_{i=0}^{q} f(x_i, y(x_i)) \int_{x_{p-j}}^{x_{p+k}} \ell_i(t)\, dt = h \sum_{i=0}^{q} \beta_{qi} f(x_i, y(x_i)),$$

gdje smo s β_{qi} označili

$$\beta_{qi} = \frac{1}{h} \int_{x_{p-j}}^{x_{p+k}} \ell_i(t)\, dt = \int_{-j}^{k} \prod_{\substack{l=0 \\ l\neq i}}^{q} \frac{s+l}{-i+l}\, ds, \quad i = 0, \ldots, q. \tag{10.4.3}$$

Zamjenom vrijednosti $y(x_i)$ aproksimacijama y_i dobivamo višekoračnu metodu oblika

$$y_{p+k} = y_{p-j} + h \sum_{i=0}^{q} \beta_{qi} f_{p-i}.$$

Najpoznatiji primjeri višekoračnih metoda ovog tipa su Adams–Bashforthova metoda i Adams–Moultonova metoda. U Adams–Bashforthovoj metodi je $k = 1$ i $j = 0$ te metoda glasi:

$$y_{p+1} = y_p + h(\beta_{q0} f_p + \beta_{q1} f_{p-1} + \cdots + \beta_{qq} f_{p-q}). \tag{10.4.4}$$

Sljedeća tablica prikazuje koeficijente β_{qi} za ovu metodu, izračunate prema formuli (10.4.3).

	i				
β_{qi}	0	1	2	3	4
β_{0i}	1				
$2\beta_{1i}$	3	-1			
$12\beta_{2i}$	23	-16	5		
$24\beta_{3i}$	55	-59	37	-9	
$720\beta_{4i}$	1901	-2774	2616	-1274	251

Izborom $k = 0$ i $j = 1$ dobijamo Adams–Moultonove metode:

$$y_{p+1} = y_p + h(\beta_{q0} f_{p+1} + \beta_{q1} f_p + \cdots + \beta_{qq} f_{p+1-q}). \tag{10.4.5}$$

Za razliku od eksplicitnih Adams–Basforthovih metoda, ove metode su implicitne. Koeficijenti su im prikazani u sljedećoj tablici.

	i				
β_{qi}	0	1	2	3	4
β_{0i}	1				
$2\beta_{1i}$	1	1			
$12\beta_{2i}$	5	8	-1		
$24\beta_{3i}$	9	19	-5	1	
$720\beta_{4i}$	251	646	-264	106	-19

Od ostalih višekoračnih metoda izvedene iz ovih formula, poznatije su još Nyströmove ($k = 1$ i $j = 1$) i Milneove ($k = 0$ i $j = 2$) metode.

Niz metoda možemo dobiti i pomoću formula za deriviranje. Neka je $P(x)$ polinom koji interpolira $y(x)$ u točkama x_{n-i}:

$$P(x_{n-i}) = y(x_{n-i}), \quad i = 0, \ldots, k.$$

Korištenjem interpolacijskog polinoma u Lagrangeovoj formi, polinom $P(x)$ možemo prikazati u obliku

$$P(x) = \sum_{i=0}^{k} \ell_i(x) y(x_{n-i}), \quad \ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{k} \frac{x - x_{n-j}}{x_{n-i} - x_{n-j}}.$$

Deriviranjem u čvoru x_{n-r} dobivamo

$$P'(x_{n-r}) = \frac{1}{h}\sum_{i=0}^{k} h\ell_i'(x_{n-r})y(x_{n-i}).$$

Supstitucijama $y_{n-i} \approx y(x_{n-i})$ i $f_{n-r} = f(x_{n-r}, y_{n-r}) \approx P'(x_{n-r})$ dobivamo metodu

$$\sum_{i=0}^{k} \alpha_i y_{n-i} = h f_{n-r}.$$

Uvođenjem oznake $p = (x - x_n)/h$, vidimo da koeficijenti

$$\alpha_i = h\ell_i'(x_{n-r}) = \frac{d}{dp}\prod_{\substack{j=0\\ j\neq i}}^{k}\frac{p+j}{j-i}\Big|_{p=-r} = \frac{1}{r-i}\prod_{\substack{j=0\\ j\neq i,r}}^{k}\frac{p+j}{j-i}$$

ne ovise o čvorovima x_{n-i} i širini mreže h. Za $r = 1$ dobivamo eksplicitnu metodu

$$\sum_{i=0}^{k} \alpha_i y_{n-i} = h f_{n-1}, \tag{10.4.6}$$

dok je za izbor $r = 0$ metoda implicitna:

$$\sum_{i=0}^{k} \alpha_i y_{n-i} = h f_{n}. \tag{10.4.7}$$

Zbog alternativnog načina izvoda ovih metoda korištenjem podijeljenih razlika unazad, ove metode poznate su pod nazivom **BDF metode** (engl. backward difference formulas). Koeficijenti za eksplicitne i implicitne BDF metode su prikazani u sljedećim dvjema tablicama.

k	η_1	α_1	α_2	α_3	α_4	α_5	α_6	α_7
1	1	1						
2	2	0	1					
3	3	$-\frac{3}{2}$	3	$-\frac{1}{2}$				
4	4	$-\frac{10}{3}$	6	$-\frac{10}{3}$	$\frac{1}{3}$			
5	5	$-\frac{65}{12}$	10	-5	$\frac{5}{3}$	$-\frac{1}{4}$		
6	6	$-\frac{77}{10}$	15	-10	30	$-\frac{3}{2}$	$\frac{1}{5}$	
7	7	$-\frac{203}{20}$	21	$-\frac{35}{2}$	$\frac{35}{3}$	$-\frac{21}{4}$	$\frac{7}{5}$	$-\frac{1}{6}$

k	η_1^*	α_1^*	α_2^*	α_3^*	α_4^*	α_5^*	α_6^*	α_7^*
1	1	1						
2	$\frac{2}{3}$	$\frac{4}{3}$	$-\frac{1}{3}$					
3	$\frac{6}{11}$	$\frac{18}{11}$	$-\frac{9}{11}$	$\frac{2}{11}$				
4	$\frac{12}{25}$	$\frac{48}{25}$	$-\frac{36}{25}$	$\frac{16}{25}$	$-\frac{3}{25}$			
5	$\frac{60}{137}$	$\frac{300}{137}$	$-\frac{300}{137}$	$\frac{200}{137}$	$-\frac{75}{137}$	$\frac{12}{137}$		
6	$\frac{60}{147}$	$\frac{360}{147}$	$-\frac{450}{147}$	$\frac{400}{147}$	$-\frac{225}{147}$	$\frac{72}{147}$	$-\frac{10}{147}$	
7	$\frac{140}{363}$	$\frac{20}{363}$	$-\frac{1470}{363}$	$\frac{4900}{1089}$	$-\frac{1225}{363}$	$\frac{588}{363}$	$-\frac{490}{1089}$	$\frac{20}{363}$

10.4.1. Konzistencija i stabilnost

Nakon ovih primjera izvoda nekih višekoračnih metoda postavlja se pitanje njihove točnosti. Kao i kod jednokoračnih metoda, prvo ćemo promatrati koliko dobro rješenje diferencijalne jednadžbe

$$y' = f(x, y), \quad y(x_0) = y_0, \quad x_0 \in [a, b], \quad y_0 \in \mathbb{R} \tag{10.4.8}$$

zadovoljava rekurzivnu formulu

$$\sum_{j=0}^{r} \alpha_j y_{i+1-j} = h \sum_{j=0}^{r} \beta_j f(x_{i+1-j}, y_{i+1-j}), \tag{10.4.9}$$

koja definira višekoračnu metodu. U gornju ćemo rekurziju umjesto y_j uvrstiti rješenje diferencijalne jednadžbe $y(x_j)$, a zatim dobiveni izraz razviti u Taylorov red:

$$\sum_{j=0}^{r} \alpha_j y(x_{i+1-j}) - h \sum_{j=0}^{r} \beta_j f(x_{i+1-j}, y(x_{i+1-j})) = \sum_{j=0}^{\infty} C_j h^j. \tag{10.4.10}$$

Rekurzivna formula biti će točnija što je više prvih članova u razvoju jednako nuli. Općenito, ukoliko je $C_0 = C_1 = \cdots = C_p = 0$ i $C_{p+1} \neq 0$, kažemo da je metoda reda p. Ukoliko je $p \geq 1$ kažemo da je metoda konzistentna.

Tako za metodu

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i)$$

uvrštavanje točnog rješenja i razvoj u red oko točke x_i daje

$$y(x_{i+1}) - y(x_{i-1}) - 2hy'(x_i) = \sum_{j=0}^{\infty} y^{(j)}(x_i)\frac{h^j}{j!} - \sum_{j=0}^{\infty} y^{(j)}(x_i)\frac{(-1)^j h^j}{j!} - 2hy'(x_i)$$
$$= \sum_{j=1}^{\infty} y^{(2j+1)}(x_i)\frac{2}{(2j+1)!}h^{2j+1} = \frac{y^{(3)}(x_i)}{3}h^3 + \mathcal{O}(h^5) = \mathcal{O}(h^3),$$

te je ova metoda reda 2. Uočimo da smo iskoristili da je y rješenje diferencijalne jednadžbe, tj. da vrijedi $y'(x_i) = f(x_i, y(x_i))$. Pretpostavili smo da je $y \in C^\infty(a,b)$, no očito je dovoljno zahtijevati da y ima neprekidnu treću derivaciju, tj. $y \in C^3(a,b)$.

Dakle, izraz (10.4.10) pokazuje kvalitetu aproksimacije višekoračne metode. U daljnjem tekstu koristiti ćemo malo promijenjen izraz za pogrešku, tzv. **lokalnu pogrešku diskretizacije**:

$$\tau(x;h) = \frac{1}{h}\sum_{j=0}^{r} \alpha_j y(x - jh) - \sum_{j=0}^{r} \beta_j y'(x - jh),$$

gdje je y egzaktno rješenje diferencijalne jednadžbe (10.4.8). Uočimo da je lokalna pogreška diskretizacije dobivena uvrštavanjem točnog rješenja u rekurziju (10.4.8) uz zamjenu $x = x_{i+1}$. Sada možemo definirati konzistenciju metode.

Definicija 10.4.1 *Višekoračnu metodu zovemo* **konzistentnom** *ako za svaki $f \in F_1(a,b)$ i $x \in [a,b]$ lokalna pogreška diskretizacije τ zadovoljava*

$$\lim_{h\to 0} \tau(x;h) = 0.$$

Ukoliko je

$$\tau(x;h) = \mathcal{O}(h^p)$$

kažemo da je metoda reda p.

Primjer 10.4.1 *Pokažimo da je red $(r+1)$-koračne Adams–Bashforthove metode (10.4.4) jednak $r+1$.*

Iskoristimo li rekurziju za Adams–Bashforthovu metodu (10.4.4) uvrštavanjem točnog rješenja $y(x)$, dobivamo

$$\tau(x;h) = \frac{1}{h}(y(x+h)) - y(x)) - \sum_{i=0}^{r} \beta_{ri} y'(x - ih).$$

Radi jednostavnosti, označimo $x_j = x + jh$, $j = -r, \ldots, 1$. Sada je lokalna pogreška diskretizacije jednaka

$$\tau(x;h) = \frac{1}{h}(y(x_{p+1})) - y(x_p)) - \sum_{i=0}^{r} \beta_{ri} y'(x_{p-i}).$$

Uvrštavanjem izraza za koeficijente β_{ri} dane s (10.4.3), dobivamo

$$\tau(x;h) = \frac{1}{h}\int_{x_p}^{x_{p+1}} y'(t)\,dt - \sum_{i=0}^{r}\Big(\frac{1}{h}\int_{x_p}^{x_{p+1}} \ell_i(t)\,dt\Big)y'(x_{p-i})$$
$$= \frac{1}{h}\int_{x_p}^{x_{p+1}} y'(t)\,dt - \frac{1}{h}\int_{x_p}^{x_{p+1}} P_r(t)\,dt = \frac{1}{h}\int_{x_p}^{x_{p+1}} [y'(t) - P_r(t)]\,dt,$$

gdje je P_r polinom koji interpolira y' u točkama $x_{p-r},\dots,x_p$. Primjenom ocjene za pogrešku interpolacije

$$y'(t) - P_r(t) = \omega(t)\,\frac{y^{(r+2)}(\xi(t))}{(r+1)!}, \quad \xi(t)\in(x_{p-r},x_p),$$
$$\omega(t) = (t-x_{p-r})(t-x_{p-r+1})\cdots(t-x_p),$$

dobivamo

$$\tau(x;h) = \frac{1}{h}\int_{x_p}^{x_{p+1}} \omega(t)\,\frac{y^{r+2}(\xi(t))}{(r+1)!}\,dt.$$

Budući da su $x_{p-r},\dots,x_p$ sve nultočke polinoma ω, ω ne mijenja predznak na intervalu $[x_p,x_{p+1}]$ pa vrijedi

$$\tau(x;h) = \frac{y^{r+2}(\eta)}{(r+1)!}\,\frac{1}{h}\int_{x_p}^{x_{p+1}} \omega(t)\,dt.$$

Supstitucijom $u=(t-x_p)/h$ dobivamo

$$t - x_{p-j} = h(u+j) \quad i \quad \omega(t) = h^{r+1}u(u+1)\cdots(u+r),$$

te gornji integral prelazi u

$$\tau(x;h) = \frac{y^{r+2}(\eta)}{(r+1)!}\,\frac{1}{h}\,h^{r+1}h\int_0^1 u(u+1)\cdots(u+r)\,du = h^{r+1}\,\frac{y^{r+2}(\eta)}{(r+1)!}\int_0^1\prod_{j=0}^{r}(u+j)\,du,$$

te je red $(r+1)$-koračne Adams–Bashforhove metode jednak $r+1$.

Razmatranjem analognim onom u prethodnom primjeru pokazuje se da je red r-koračne Adams–Moultonove metode za jedan veći od broja koraka i iznosi $r+1$. Za r-koračne eksplicitne metode (10.4.6) i r-koračne implicitne metode (10.4.7) izvedene iz formula za deriviranje, red metode odgovara broju koraka.

Korištenjem iste ideje kao kod Runge–Kuttinih metoda, koeficijente u r-koračnoj metodi

$$\sum_{j=0}^{r}\alpha_j y_{i+1-j} = h\sum_{j=0}^{r}\beta_j f(x_{i+1-j},y_{i+1-j})$$

možemo određivati tako da red metode bude što je moguće veći, tj. da poništimo što više prvih članova u Taylorovom razvoju (10.4.10). Fiksiranjem $\alpha_0 = 1$ imamo $2r$ slobodnih koeficijenata za eksplicitnu metodu. Koeficijenti C_j u Taylorovom razvoju zavisit će o koeficijentima α_j i β_j. Nije teško pokazati da je ta zavisnost linearna, te s $2r$ slobodnih koeficijenata možemo poništiti prvih $2r$ članova razvoja: $C_0 = C_1 = \cdots = C_{2k-1}$. Također, može se pokazati da će vrijediti $C_{2r} \neq 0$, pa na taj način možemo konstruirati metodu reda $2r - 1$. Slično vrijedi i za r-koračne implicitne metode. Ovdje imamo jedan koeficijent više (β_0), te možemo poništiti jedan član više u Taylorovom razvoju i dobiti metodu reda $2r$.

Prirodno se nameće pitanje zašto bismo koristili navedene metode, ako postoje metode dvostrukog reda s istim brojem koraka. Za razliku od jednokoračnih metoda gdje je kozistentnost metode bio dovoljan uvjet za konvergenciju, kod višekoračnih metoda, da bi aproksimacija konvergirala k točnom rješenju, uz konzistentnost treba biti zadovoljen još jedan dodatni uvjet, a to je stabilnost. To ćemo ilustrirati sljedećim primjerom.

Primjer 10.4.2 *Konstruirajmo dvokoračnu eksplicitnu metodu:*

$$y_{i+2} + a_1 y_{i+1} + a_0 y_i = h[b_1 f(x_{i+1}, y_{i+1}) + b_0 f(x_i, y_i)]$$

tako da red konzistencije bude što je moguće veći.

Razvojem lokalne pogreške diskretizacije u $x = x_i$

$$\tau(x; h) = \frac{1}{h}\left[y(x + 2h) + a_1 y(x + h) + a_0 y(x)\right] - \left[b_1 y'(x + h) + b_0 y'(x)\right]$$

u Taylorov red, dobivamo

$$\begin{aligned}\tau(x; h) = \frac{1}{h}\, & y(x)(1 + a_1 + a_0) + y'(x)(2 + a_1 - b_1 - b_0) \\ & + hy''(x)\Big(2 + \frac{1}{2}a_1 - b_1\Big) + h^2 y'''(x)\Big(\frac{4}{3} + \frac{1}{6}a_1 - \frac{1}{2}b_1\Big) + \mathcal{O}(h^3).\end{aligned}$$

Koeficijente a_0, a_1, b_0 i b_1 odredimo tako da poništimo što je moguće više početnih članova u Taylorovom razvoju. To nas vodi na sustav jednadžbi

$$\begin{aligned} 1 &+ a_1 + a_0 &&= 0 \\ 2 &+ a_1 - b_1 - b_0 &&= 0 \\ 2 &+ \frac{1}{2}a_1 - b_1 &&= 0 \\ \frac{4}{3} &+ \frac{1}{6}a_1 - \frac{1}{2}b_1 &&= 0 \end{aligned}$$

Rješenje ovog sustava je $a_1 = 4$, $a_0 = -5$, $b_1 = 4$, $b_0 = 2$, te je metoda definirana rekurzijom

$$y_{i+2} + 4y_{i+1} - 5y_i = h(4f_{i+1} + 2f_i).$$

Ova metoda je reda 3 (jer je $\tau(x;h) = \mathcal{O}(h^3)$, a može se provjeriti da je član uz h^3 različit od nule).

Promatrat ćemo numeričko rješavanje diferencijalne jednadžbe

$$y' = -y, \quad y(0) = 1,$$

s egzaktnim rješenjem $y = e^{-x}$. Uz korak $h = 10^{-2}$ i egzaktne početne vrijednosti $y_0 = 1$ i $y_1 = e^{-h}$ dobivamo sljedeću tablicu:

i	$y_i - y(x_i)$
2	$-0.164 \cdot 10^{-8}$
3	$+0.501 \cdot 10^{-8}$
4	$-0.300 \cdot 10^{-7}$
5	$+0.144 \cdot 10^{-6}$
⋮	⋮
96	$-0.101 \cdot 10^{58}$
97	$+0.512 \cdot 10^{58}$
98	$-0.257 \cdot 10^{59}$
99	$+0.129 \cdot 10^{60}$
100	$-0.652 \cdot 10^{60}$

Iz tablice je jasno da metoda divergira. Uzrok tome je objašnjen u daljnjem tekstu.

Višekoračna metoda

$$\sum_{j=0}^{r} \alpha_j y_{i+1-j} = h \sum_{j=0}^{r} \beta_j f(x_{i+1-j}, y_{i+1-j})$$

definira dva polinoma

$$\rho(z) = \sum_{j=0}^{r} \alpha_j z^{r-j} \qquad \text{i} \qquad \sigma(z) = \sum_{j=0}^{r} \beta_j z^{r-j}.$$

Ujedno, ova dva polinoma određuju jednu višekoračnu metodu, pa se često umjesto višekoračne metode koristi naziv (ρ, σ)-shema.

Definicija 10.4.2 *Za višekoračnu metodu kažemo da je* **stabilna** *ako nultočke z_j polinoma $\rho(z)$ zadovoljavaju*

1. *Sve nultočke su po apsolutnoj vrijednosti manje od 1 ($|z_j| \leq 1$).*
2. *Ako je $|z_j| = 1$ tada je z_j jednostruka nultočka ($\rho'(z_j) \neq 0$).*

Zajedno uvjete 1 i 2 zovemo **uvjet stabilnosti**.

Uvjet stabilnosti se često naziva i uvjet korijena.

Za metodu iz prošlog primjera je

$$\rho(z) = z^2 + 4z - 5.$$

Nultočke su mu $z_1 = 1$ i $z_2 = -5$. Budući da je $|z_2| > 1$, ρ ne zadovoljava uvjet stabilnosti, tj. metoda nije stabilna. Sljedeći teorem nam objašnjava razlog divergencije ove metode.

Teorem 10.4.1 *Linearna višekoračna metoda je konvergentna ako i samo ako je konzistentna i stabilna.*

Može se pokazati da stabilna r-koračna metoda ima red

$$p \leq \begin{cases} r+1, & \text{ako je } r \text{ neparan,} \\ r+2, & \text{ako je } r \text{ paran.} \end{cases}$$

Definicija konvergencije višekoračne metode se nešto razlikuje od konvergencije jednokoračnih metoda, pa će objašnjenje ove definicije i dokaz prethodnog teorema biti izloženi u sljedećim potpoglavljima.

10.4.2. Prediktor-korektor par

Dosad je ostalo otvoreno pitanje kako izračunati y_{i+1} u implicitnoj metodi

$$y_{i+1} + \sum_{j=1}^{k} \alpha_j^* y_{i+1-j} = \beta_0^* h f(x_{i+1}, y_{i+1}) + h \sum_{j=1}^{k} \beta_j^* f_{i+1-j}.$$

Ako označimo

$$c = -\sum_{j=1}^{k} \alpha^* y_{i+1-j} + h \sum_{j=1}^{k} \beta_j^* f_{i+1-j}, \qquad \varphi(y) = \beta_0^* h f(x_{i+1}, y) + c,$$

y_{i+1} je rješenje nelinearne jednadžbe $y = \varphi(y)$. Budući da možemo izabrati dovoljno malen korak integracije h takav da je nejednakost

$$|\varphi'(y)| = h|\beta_0^*| \left| \frac{\partial f(x_{i+1}, y)}{\partial y} \right| < 1$$

zadovoljena, slijedi da jednostavne iteracije

$$y^{[m+1]} = \varphi(y^{[m]}), \quad m = 0, 1, 2, \ldots$$

konvergiraju prema rješenju jednadžbe. Za odabir početne aproksimacije $y^{[0]}$ koristi se neka od eksplicitnih metoda

$$y_{i+1} + \sum_{j=1}^{\bar{k}} \alpha_j y_{i+1-j} = h \sum_{j=1}^{\bar{k}} \beta_j f_{i+1-j}.$$

Sada možemo zapisati cijeli algoritam:

$$y_{i+1}^{[0]} = -\sum_{j=1}^{\bar{k}} \alpha_j y_{i+1-j} + h \sum_{j=1}^{\bar{k}} \beta_j f_{i+1-j},$$

$$y_{i+1}^{[m+1]} = \beta_0^* h f(x_{i+1}, y_{i+1}^{[m]}) - \sum_{j=1}^{k} \alpha_j^* y_{i+1-j} + h \sum_{j=1}^{k} \beta_j^* f_{i+1-j}, \quad m = 0, \ldots, M-1,$$

$$y_{i+1} = y_{i+1}^{[M]}.$$

Broj iteracija (M) može biti unaprijed zadan ili se iteracije provode dok se jednadžba ne riješi do na neku unaprijed zadanu točnost. U primjeni, broj iteracije nije velik, uvijek se radi o nekoliko iteracija.

Znamo da jednostavne iteracije konvergiraju linearno prema rješenju jednadžbe. Konvergenciju možemo ubrzati primjenom Newtonove metode, što će biti opisano u sljedećem potpoglavlju.

Još nam je ostalo za promotriti kako odabrati korektor-prediktor par. Točnost metode definirana je s točnošću korektora, tj. implicitne metode. Ako je red korektora, eksplicitne metode kojom određujemo početnu aproksimaciju $y_{i+1}^{[0]}$, za jedan veći od reda korektora početna će aproksimacija biti pretočna, za jedan red točnija nego što je rješenje nelinearne jednadžbe (korektora). S druge strane, ako je red prediktora manji od reda korektora, početna aproksimacija je preslaba, te bi trebalo previše iteracija korektora da se nelinearna jednadžba riješi na zadovoljavajuću točnost. Stoga je uobičajeno da se za prediktor-korektor par uzimaju eksplicitna i implicitna metoda istoga reda. Često korišten par je k-koračna Adams–Bashforthova metoda kao prediktor i $(k-1)$-koračna Adams–Moultonova metoda kao korektor. Uz ovakav odabir prediktor-korektor para govorimo o Adams–Bashforth–Moultonovim metodama. Isto tako, eksplicitna i implicitna k-koračna metoda izvedena iz formula za numeričko deriviranje koriste se kao prediktor-korektor par.

10.4.3. Linearne diferencijske jednadžbe

Za razumijevanje stabilnosti višekoračnih metoda nužno je poznavati neke osnovne činjenice o linearnim diferencijskim jednadžbama. Pod linearnom homogenom diferencijskom jednadžbom podrazumijevamo jednadžbu oblika

$$u_{j+r} + \alpha_{r-1} u_{j+r-1} + \alpha_{r-2} u_{j+r-2} + \cdots + \alpha_0 u_j = 0, \quad j = 0, 1, 2, \ldots \tag{10.4.11}$$

Očito je da je za svaki skup početnih vrijednosti $u_0, \ldots, u_{r-1}$ jednoznačno određen niz brojeva u_j, $j = 0, 1, 2, \ldots$ koji rješava jednadžbu (10.4.11).

Uz gornju diferencijsku jednadžbu vežemo polinom

$$\rho(z) = z^r + \alpha_{r-1} z^{r-1} + \cdots + \alpha_1 z + \alpha_0. \tag{10.4.12}$$

Rješenje diferencijske jednadžbe (10.4.11) dobivamo pomoću nultočaka polinoma ρ.

Teorem 10.4.2 *Neka polinom $\rho(z)$ iz (10.4.12) ima k različitih nultočaka λ_i višestrukosti σ_i, $i = 1, \ldots, k$. Tada za proizvoljne polinome $P_i(t)$ stupnja strogo manjeg od σ_i, $i = 1, \ldots, k$, niz*

$$u_j = P_1(j)\lambda_1^j + P_2(j)\lambda_2^j + \cdots + P_k(j)\lambda_k^j, \quad j = 0, 1, 2, \ldots \tag{10.4.13}$$

je rješenje diferencijske jednadžbe (10.4.11). Obratno, svako rješenje diferencijske jednadžbe (10.4.11) može se jedinstveno prikazati u obliku (10.4.13).

Dokaz. Zbog homogenosti jednadžbe (10.4.11) vrijedi da je $u_j + v_j$ rješenje ako su u_j i v_j rješenja diferencijske jednadžbe. Stoga je dovoljno pokazati da za λ, nultočku polinoma ρ višestrukosti σ, vrijedi da je niz

$$u_j = P(j)\lambda^j, \quad j = 0, 1, 2, \ldots$$

rješenje diferencijske jednadžbe (10.4.11), gdje je P proizvoljan polinom stupnja strogo manjeg od σ, (oznaka $\partial P < \sigma$).

Za fiksni $j \geq 0$, pomoću interpolacijskog polinoma zapisanog u Newtonovom obliku prikažimo $P(j + t)$:

$$P(j + t) = a_0 + a_1 t + a_2 t(t - 1) + \cdots + a_r t(t - 1) \cdots (t - r + 1).$$

Budući da je $\partial P < \sigma$ vrijedi $a_\sigma = a_{\sigma+1} = \cdots = a_r = 0$. Uz oznaku $\alpha_r = 1$, sada slijedi

$$\begin{aligned} u_{j+r} + \alpha_{r-1}u_{j+r-1} + \cdots + \alpha_0 u_j &= \lambda^j \sum_{p=0}^{r} \alpha_p \lambda^p P(j + p) \\ &= \lambda^j \sum_{p=0}^{r} \alpha_p \lambda^p \Big[a_0 + \sum_{\tau=1}^{r} a_\tau p(p - 1) \cdots (p - \tau + 1) \Big] \\ &= \lambda^j [a_0\rho(\lambda) + a_1\rho'(\lambda) + \cdots + a_{\sigma-1}\rho^{(\sigma-1)}(\lambda)] = 0, \end{aligned}$$

jer je višestrukost nultočke λ jednak σ, tj. $\rho^{(\tau)}(\lambda) = 0$ za $0 \leq \tau \leq \sigma - 1$. Time smo dokazali prvi dio teorema.

Uočimo da je polinom $P(t) = c_0 + c_1 t + \cdots + c_{\sigma-1}t^{\sigma-1}$ stupnja manjeg od σ određen s točno σ koeficijenata c_j, tako da zbog $\sigma_1 + \sigma_2 + \cdots + \sigma_k = r$ prikaz (10.4.13) sadrži ukupno r slobodnih parametara (koeficijenata polinoma P_i).

Drugi dio teorema dakle kaže da prikladnim odabirom parametara možemo dobiti proizvoljno rješenje, tj. da za svaki izbor početnih vrijednosti $u_0, \ldots, u_{r-1}$ postoji jedinstveno rješenje sljedećeg sustava od r linearnih jednadžbi s r nepoznatih, koeficijenata polinoma P_i $(i = 1, \ldots, k)$:

$$P_1(j)\lambda_1^j + P_2(j)\lambda_2^j + \cdots + P_k(j)\lambda_k^j = u_j, \quad j = 0, \ldots, r - 1.$$

Dokaz ove tvrdnje je elementaran, ali i dugotrajan, stoga ga ovdje preskačemo. ■

U primjeni višekoračnih metoda od interesa je ponašanje rasta rješenja u_n diferencijske jednadžbe (10.4.11) kad $n \to \infty$. Posebno je važno utvrditi uvjete koji osiguravaju da je

$$\lim_{n\to\infty} \frac{u_n}{n} = 0 \tag{10.4.14}$$

za sve realne početne vrijednosti $u_0, \ldots, u_{r-1}$. To nam daje sljedeća lema.

Lema 10.4.1 *Ako polinom ρ iz (10.4.12) zadovoljava uvjete stabilnosti iz definicije 10.4.2 tada rješenje diferencijske jednadžbe (10.4.11) zadovoljava (10.4.14).*

Dokaz. Pretpostavimo da vrijedi (10.4.14), i neka je λ nultočka od ρ. Tada je niz $u_j = \lambda^j$ rješenje diferencijske jednadžbe (10.4.11). Za $|\lambda| > 1$ niz

$$\frac{u_n}{n} = \frac{\lambda^n}{n}$$

divergira, pa mora biti $\lambda \leq 1$.

Neka je sada $|\lambda| = 1$ i λ je višestruka nultočka od ρ. Tada je

$$\rho'(\lambda) = r\lambda^{r-1} + (r-1)\alpha_{r-1}\lambda^{r-2} + \cdots + 1 \cdot \alpha_1 = 0.$$

Prema tome je i niz $u_j = j\lambda^j$, $j \geq 0$ rješenje diferencijske jednadžbe (10.4.11):

$$\begin{aligned} u_{j+r} + \alpha_{r-1}u_{j+r-1} + \cdots + \alpha_0 u_j &= j\alpha^j(\lambda^r + \alpha_{r-1}\lambda^{r-1} + \cdots + \alpha_0) \\ &\quad + \lambda^{j+1}(r\lambda^{r-1} + (r-1)\alpha_{r-1}\lambda^{r-2} + \cdots + \alpha_1) \\ &= j\alpha^j\rho(\lambda) + \lambda^{j+1}\rho'(\lambda) = 0. \end{aligned}$$

Budući da niz $u_n/n = \lambda^n$ ne konvergira nuli kada $n \to \infty$, slijedi da λ mora biti jednostruka nultočka.

Obratno, neka je zadovoljen uvjet stabilnosti. Ako definiramo

$$U_j = \begin{bmatrix} u_j \\ u_{j+1} \\ \vdots \\ u_{j+r-1} \end{bmatrix} \in \mathbb{C}^r, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 & 0 \\ -\alpha_0 & \ldots & \ldots & \ldots & -\alpha_{r-1} \end{bmatrix},$$

diferencijska jednadžba (10.4.11) je ekvivalentna rekurziji

$$U_{j+1} = \mathbf{A}U_j.$$

ρ je karakteristični polinom matrice $\mathbf{A}$ te, budući da je uvjet stabilnosti zadovoljen, postoji norma $\| \ \|$ na $\mathbb{C}^r$ takva da odgovarajuća inducirana matrična norma zadovoljava $\|\mathbf{A}\| \leq 1$. Prema tome, za svaki $U_0 \in \mathbb{C}^r$ vrijedi

$$\|U_n\| = \|\mathbf{A}^n U_0\| \leq \|\mathbf{A}\|^n \|U_0\| \leq \|U_0\| \quad \text{za } n = 0, 1, \ldots$$

Budući da su na $\mathbb{C}^r$ sve norme ekvivalentne, postoji $k > 0$ takav da je

$$\frac{\|U\|}{k} \leq \|U\|_\infty \leq k\|U\|,$$

te posebno vrijedi

$$\|U_n\|_\infty \leq k^2\|U_0\|_\infty, \quad n = 0, 1, \ldots$$

Stoga je

$$\lim_{n\to\infty} \frac{\|U_n\|_\infty}{n} = 0,$$

tj. posebno vrijedi $\lim_{n\to\infty} \frac{u_n}{n} = 0$. ■

10.4.4. Konvergencija linearnih višekoračnih metoda

Kao i kod jednokoračnih metoda, kada govorimo o konvergenciji metode mislimo na ponašanje **globalne pogreške diskretizacije**:

$$e(x; h) = y_n - y(x),$$

gdje je $x \in (a, b)$, $h = h_n = (x - a)/n$.

Jasno je da globalna pogreška diskretizacije ovisi o lokalnoj pogrešci diskretizacije. Međutim, to nije jedini izvor pogreške. Da bismo startali r-koračnu metodu, prvo je potrebno izračunati r početnih vrijednosti $y_0, \ldots, y_{r-1}$. Dok y_0 možemo odrediti iz početnog uvjeta diferencijalne jednadžbe, ostale vrijednosti moramo odrediti nekom drugom, najčešće jednokoračnom, metodom. U svakom slučaju, pri njihovom određivanju javit će se određena pogreška ε_i:

$$y(x_i) = y_i + \varepsilon_i, \quad i = 0, \ldots, r - 1.$$

Ova pogreška ne ovisi o promatranoj višekoračnoj metodi, već o načinu na koji određujemo početne vrijednosti. Očito je, da ako želimo da globalna pogreška diskretizacije teži nuli kada $n \to \infty$, pogreške početnih vrijednosti trebaju zadovoljavati

$$\lim_{n\to\infty} \varepsilon_i = 0, \quad i = 0, \ldots, r - 1.$$

Sada možemo izreći definiciju konvergencije višekoračne metode.

Definicija 10.4.3 *Višekoračnu metodu danu s (10.4.9) zovemo* **konvergentnom** *ako je*

$$\lim_{n\to\infty} e(x; h_n) = 0, \quad h_n = \frac{x - a}{n}, \quad n = 1, 2, \ldots$$

za sve $x \in [a, b]$, *sve* $f \in F_1(a, b)$ *i sve* y_i, $i = 0, \ldots, r - 1$ *za koje je*

$$\lim_{n\to\infty} (y(x_i) - y_i) = 0, \quad i = 0, \ldots, r - 1.$$

Sada možemo izreći teorem o redu višekoračne metode.

Teorem 10.4.3 *Linearna višekoračna metoda je reda p ako i samo ako je $z = 1$ p-struka nultočka funkcije*

$$\frac{\rho(z)}{\ln z} - \sigma(z). \tag{10.4.15}$$

Prethodni teorem povlači i sljedeći korolar.

Korolar 10.4.1 *Linearna višekoračna metoda je konzistentna ako i samo ako je $\rho(1) = 0$ i $\rho'(1) = \sigma(1)$.*

Dokaz. Iz teorema slijedi da je metoda konzistentna ako i samo ako je 1 nultočka funkcije (10.4.15). Zbog $\ln 1 = 0$ treba vrijediti i $\rho(1) = 1$. Jednako tako, zbog

$$\frac{\rho(z)}{\ln z}\Big|_{z=1} = \lim_{z\to 1}\frac{\rho(z)}{\ln z} = \lim_{z\to 1}\frac{\rho'(z)}{\frac{1}{z}} = \rho'(1)$$

treba vrijediti i $\rho'(1) - \sigma(1) = 0$. ■

Na kraju, pokažimo vezu između stabilnosti i konzistentnosti te konvergencije linearne višekoračne metode.

Teorem 10.4.4 *Stabilne i konzistentne linearne višekoračne metode su konvergentne.*

Dokaz. Dokaz ovog teorema je sličan dokazu konvergencije jednokoračnih metoda. Neka je y egzaktno rješenje jednadžbe

$$y' = f(x, y), \quad y(x_0) = y_0, \quad f \in F_1(a, b).$$

Fiksirajmo $x \in [a, b]$ i za $n \in \mathbb{N}$ definirajmo $h = h_n = (x - x_0)/n$. S y_i označit ćemo aproksimaciju dobivenu linearnom višekoračnom metodom

$$y_i + \alpha_1 y_{i-1} + \cdots + \alpha_r y_r = h(\beta_0 f_i + \beta_1 f_{i-1} + \cdots + \beta_r f_{i-r}), \quad i = r, r+1, \ldots, \tag{10.4.16}$$

uz

$$y_i = y(x_i) + \varepsilon_i, \quad i = 0, \ldots, r-1,$$

gdje je $\lim_{n\to\infty} \varepsilon_i = 0$.

Uvjet $\lim_{n\to\infty} \varepsilon_i = 0$ znači da postoji funkcija $\varepsilon(h)$ takva da je $|\varepsilon_i| \le \varepsilon(h)$ za $i = 0, \ldots, r-1$ i $\lim_{h\to 0} \varepsilon(h) = 0$. Zbog konzistentnosti metode, lokalna pogreška diskretizacije u točki x_i

$$\tau(x_i; h) = \tau_i = \frac{1}{h}\left[y(x_i) + \sum_{j=1}^{r} \alpha_j y(x_{i-j})\right] - \sum_{j=0}^{r} \beta_j f(x_{i-j}, y(x_{i-j})) \tag{10.4.17}$$

zadovoljava

$$\lim_{h\to 0} \tau_i = 0, \tag{10.4.18}$$

tj. τ_i možemo omeđiti nekom funkcijom $t(h)$, $|\tau_i| \le t(h)$, koja zadovoljava $\lim\limits_{h\to 0} t(h) = 0$.

Oduzimanjem jednakosti (10.4.17) pomnožene s h od jednakosti (10.4.16) dobivamo rekurziju za pogreške $e_i = y_i - y(x_i)$:

$$e_i + \alpha_1 e_{i-1} + \cdots + \alpha_r e_r = c_i, \quad i = r, r+1, \ldots \tag{10.4.19}$$

uz

$$e_i = \varepsilon_i, \quad i = 0, \ldots, r-1$$

i

$$c_i = h\sum_{j=0}^{r} \beta_j [f(x_{i-j}, y_{i-j}) - f(x_{i-j}, y(x_{i-j}))] - h\tau_i.$$

Budući da je $f \in F_1(a,b)$, postoji konstanta $m > 0$ takva da je

$$\left|\frac{\partial f}{\partial y}(x,y)\right| \le m \quad \forall x \in [a,b] \quad \text{i} \quad \forall y \in \mathbb{R},$$

te vrijedi

$$|f(x_k, y_k) - f(x_k, y(x_k))| = \left|\frac{\partial f}{\partial y}(x_k, \eta)(y_k - y(x_k))\right| \le M|e_k|.$$

Iskoristivši ovu nejednakost, dobivamo da c_i zadovoljava

$$|c_i| \le hM\sum_{j=0}^{r} |e_{i-j}| + |h|t(h), \tag{10.4.20}$$

gdje je

$$M = m \max_{j=0,\ldots,r} |\beta_j|.$$

Pomoću vektora

$$\mathbf{e}_j = \begin{bmatrix} e_j \\ e_{j+1} \\ \vdots \\ e_{j+r-1} \end{bmatrix}, \quad \mathbf{b}_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^r,$$

i matrice

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 & 0 \\ -\alpha_r & \ldots & \ldots & \ldots & -\alpha_1 \end{bmatrix}$$

rekurziju (10.4.19) možemo zapisati u ekvivalentnom vektorskom zapisu

$$\mathbf{e}_{j+1} = \mathbf{A}\mathbf{e}_j + c_{j+r}\mathbf{b}, \quad \mathbf{e}_0 = \begin{bmatrix} \varepsilon_0 \\ \vdots \\ \varepsilon_{r-1} \end{bmatrix}. \tag{10.4.21}$$

Uočimo da je polinom ρ, definiran višekoračnom metodom (10.4.16), ujedno i svojstveni polinom matrice $\mathbf{A}$. Stabilnost metode povlači da su sve nultočke od ρ, tj. svojstvene vrijednosti od $\mathbf{A}$, po apsolutnoj vrijednosti manje od 1, a ako su jednake 1 tada su jednostruke. To znači da postoji vektorska norma $\| \ \|$ na $\mathbb{C}^r$ takva da za induciranu vektorsku normu vrijedi $\|\mathbf{A}\| \leq 1$. Budući da su sve norme na $\mathbb{C}^r$ ekvivalentne, postoji konstanta $k > 0$ takva da je

$$\frac{1}{k}\|\mathbf{e}_j\| \leq \sum_{i=0}^{r-1} |e_{j+i}| = \|\mathbf{e}_j\|_1 \leq k\|\mathbf{e}_j\|.$$

Uočivši da je

$$|e_{j+r}| \leq \sum_{i=1}^{r} |e_{j+i}| \leq k\|\mathbf{e}_{j+1}\|$$

iz nejednakosti (10.4.20) slijedi da je

$$|c_{j+r}| \leq |h|Mk(\|\mathbf{e}_j\| + \|\mathbf{e}_{j+1}\|) + |h|t(h).$$

Iskoristivši vektorski zapis rekurzije (10.4.21) i činjenice da je

$$\|\mathbf{b}\| \leq k\|\mathbf{b}\|_1 = k,$$

slijedi da je

$$(1 - |h|Mk^2)\|\mathbf{e}_{j+1}\| \leq (1 + |h|Mk^2)\|\mathbf{e}_j\| + k|h|t(h), \quad j = 0, 1, \ldots \tag{10.4.22}$$

i

$$\|\mathbf{e}_0\| \leq k\|\mathbf{e}_0\|_1 \leq kr\varepsilon(h).$$

Za

$$|h| \leq \frac{1}{2Mk^2}$$

je

$$1 - |h|Mk^2 \geq \frac{1}{2}$$

i

$$\frac{1 + |h|Mk^2}{1 - |h|Mk^2} \leq 1 + 4|h|Mk^2.$$

Sada nejednakost (10.4.22) prelazi u

$$\|\mathbf{e}_{j+1}\| \leq (1 + 4|h|Mk^2)\|\mathbf{e}_j\| + 2k|h|t(h), \quad j = 0, 1, \ldots$$

Iz leme 10.3.1 slijedi

$$\|\mathbf{e}_n\| \leq e^{4n|h|Mk^2} kr\varepsilon(h) + t(h)\frac{e^{4n|h|Mk^2} - 1}{2Mk},$$

tj. za $x \neq x_0$, $h = h_n = (x - x_0)/n$, $|h_n| \leq 1/(2Mk^2)$ vrijedi

$$\|\mathbf{e}_n\| \leq e^{4Mk^2|x-x_0|} kr\varepsilon(h_n) + t(h_n)\frac{e^{4Mk^2|x-x_0|} - 1}{2Mk}.$$

Dakle, postoje konstante C_1 i C_2, nezavisne o h, takve da je

$$|e(x;h)| = |e_n| = |y_n - y(x_n)| \leq C_1\varepsilon(h_n) + C_2 t(h_n) \tag{10.4.23}$$

za dovoljno veliki n. Konvergencija metode sada slijedi iz činjenice da je

$$\lim_{n\to\infty} \varepsilon(h_n) = \lim_{n\to\infty} \tau(h_n) = 0.$$

■

Iz ocjene (10.4.23) dobivamo sljedeći korolar.

Korolar 10.4.2 *Neka je linearna višekoračna metoda stabilna i konzistentna reda p, te $f \in F_p(a,b)$. Tada globalna pogreška diskretizacije zadovoljava*

$$e(x;h_n) = \mathcal{O}(h_n^p)$$

za sve $h_n = (x - x_0)/n$ čim pogreške ε_i zadovoljavaju

$$|\varepsilon_i| \leq \varepsilon(h_n), \quad i = 0, \ldots, r-1$$

uz $\varepsilon(h_n) = \mathcal{O}(h_n^p)$.

Ovaj korolar ujedno kazuje koju metodu moramo izabrati za određivanje početnih vrijednosti $y_0, \ldots, y_{r-1}$. Da bismo postigli da se pogreška ponaša kao $\mathcal{O}(h^p)$, tako se mora ponašati i pogreška početnih vrijednosti. Ukoliko za njihovo određivanje koristimo jednokoračnu metodu reda $\tilde{p}$, iz teorema 10.3.3 slijedi da pogreška početnih vrijednosti zadovoljava

$$|e(x_i;h)| \leq |h|^{\tilde{p}} N\,\frac{e^{M|x_i-x_0|} - 1}{M} = |h|^{\tilde{p}} N\frac{e^{Mi|h|} - 1}{M}.$$

Primjenom teorema srednje vrijednosti dobivamo da postoji $\theta \in (0,h)$ takav da je

$$e^{Mi|h|} - 1 = Mi|h|e^{Mi\theta} \leq Mi|h|e^{Mi|h|},$$

pa je

$$|e(x_i;h)| \leq |h|^{\tilde{p}} Ni|h|e^{Mi|h|}.$$

Kako je za početne vrijednosti r-koračne metode $0 \leq i \leq r-1$, vrijedi

$$|e(x_i; h)| \leq |h|^{\tilde{p}+1} N(r-1) e^{M(r-1)|h|},$$

te možemo izabrati metodu reda $\tilde{p} = p - 1$ da bi se pogreška višekoračne metode ponašala kao $\mathcal{O}(h^p)$.

Sljedeća dva teorema govore o svojstvima konvergentnih višekoračnih metoda.

Teorem 10.4.5 *Konvergentne linearne višekoračne metode su stabilne.*

Dokaz. Promatrajmo diferencijalnu jednadžbu

$$y' = 0, \quad y(a) = 0$$

s egzaktnim rješenjem $y = 0$. Za fiksirani $x \in [a, b]$ neka je y_n aproksimacija za $y(x)$ uz $h = (x-a)/n$, $n = 1, 2, \ldots$, te neka su zadane početne vrijednosti

$$y_i = \varepsilon_i, \quad i = 0, \ldots, r-1.$$

Budući da je metoda konvergentna, vrijedi

$$\lim_{n\to\infty} y_n = 0$$

čim je zadovoljeno

$$\lim_{n\to\infty} \varepsilon_i = 0 \quad i = 0, \ldots, r-1. \tag{10.4.24}$$

Izaberimo sada

$$\varepsilon_i = h u_i \quad i = 0, \ldots, r-1,$$

za neke fiksne konstante $u_0, \ldots, u_{r-1}$. Uz ovakav izbor ε_i zadovoljili smo uvjet (10.4.24). Sada definirajmo niz y_i rekurzivnom formulom

$$y_{j+r} + \alpha_1 y_{j+r-1} + \cdots + \alpha_r y_j = 0$$

uz početne vrijednosti

$$y_i = \varepsilon_i \quad i = 0, \ldots, r-1.$$

Sada vrijedi $y_i = h u_i$, gdje je niz u_i dobiven rekurzijom

$$u_{j+r} + \alpha_1 u_{j+r-1} + \cdots + \alpha_r u_j = 0.$$

Budući da je metoda konvergentna, vrijedi

$$0 = \lim_{n\to\infty} y_n = \lim_{n\to\infty} (h u_n) = (x-a) \lim_{n\to\infty} \frac{u_n}{n}.$$

Tvrdnja teorema sada slijedi direktno iz leme 10.4.1. ■

Teorem 10.4.6 *Konvergentne linearne višekoračne metode su konzistentne.*

Dokaz. Promatrajmo inicijalni problem

$$y' = 0, \quad y(0) = 1$$

s egzaktnim rješenjem $y(x) = 1$. Za početne vrijednosti $y_i = 1$, $i = 0, \ldots, r-1$, metoda daje vrijednosti y_{j+r}, $j = 0, 1, \ldots$ gdje je

$$y_{j+r} + \alpha_{r-1} y_{j+r-1} + \cdots + \alpha_0 y_j = 0. \tag{10.4.25}$$

Stavivši $h = x/n$, zbog konvergencije metode vrijedi

$$\lim_{n\to\infty} y_n = y(x) = 1.$$

Sada direktno iz (10.4.25) za $j \to \infty$ slijedi

$$C_0 = 1 + \alpha_{r-1} + \ldots + \alpha_0 = 0.$$

Da bismo dokazali da je i $C_1 = 0$, iskoristit ćemo činjenicu da je metoda konvergentna i za inicijalni problem

$$y' = 1, \quad y(0) = 0,$$

s egzaktnim rješenjem $y(x) = x$. Već smo vidjeli da je $C_0 = \rho(1) = 0$. Zbog konvergentnosti metode, iz teorema 10.4.5 slijedi i njena stabilnost, pa je $\lambda = 1$ jednostruka nultočka od ρ, tj. $\rho'(1) \neq 0$. Dakle, konstanta

$$K = \frac{\sigma(1)}{\rho'(1)}$$

je dobro definirana. Uz početne uvjete

$$y_j = jhK \quad j = 0, \ldots, r-1,$$

za inicijalni problem $y' = 1$, $y(0) = 0$, uzevši u obzir da je $y(x_j) = x_j = jh$, imamo

$$y_j = y(x_j) + \varepsilon_j \quad \text{uz} \quad \varepsilon_j = jh(K-1), \quad j = 0, \ldots, r-1.$$

Očito je zadovoljeno

$$\lim_{n\to\infty} \varepsilon_j = 0 \quad \text{za} \quad j = 0, \ldots, r-1.$$

Metoda, uz ove početne vrijednosti, daje niz y_j koji zadovoljava

$$y_{j+r} + \alpha_{r-1} y_{j+r-1} + \cdots + \alpha_0 y_j = h(\beta_0 + \beta_1 + \cdots + \beta_r) = h\sigma(1).$$

Uvrštavanjem u gornju jednadžbu, uzevši u obzir da je $\rho(1) = 0$, lagano se može provjeriti da je

$$y_j = jhk \quad \text{za sve } j.$$

Fiksirajmo sada x i stavimo $h = x/n$. Zbog konvergencije metode je

$$x = y(x) = \lim_{n\to\infty} y_n = \lim_{n\to\infty} (nhK) = \lim_{n\to\infty} (xK) = Kx.$$

Dakle, $K = 1$, odnosno $\rho'(1) = \sigma(1)$, te je $C_1 = \rho'(1) - \sigma(1) = 0$, što znači da je metoda konzistentna. ■

10.5. Gearova metoda

Dosada razmatrane formule za višekoračne metode zasnivale su se na ekvidistantnom izboru koraka h. Ako želimo razviti metodu koja će imati mogućnost promjene koraka integracije tijekom izvođenja algoritma, ovim metodama moramo pristupiti na drugi način. Ideju ćemo prikazati na Adams–Bashforth–Moultonovim (ABM) metodama, a ista se može poopćiti i na druge metode.

Uočimo da kod k-koračnih Adams–Bashforth–Moultonovih metoda za računanje y_{i+1} trebamo poznavati (pamtiti) prijašnje vrijednosti $y_n,\ f_n, \ldots, f_{n+1-k}$. Ovih $k+1$ vrijednosti možemo zapamtiti u vektoru $\mathbf{y}_n$:

$$\mathbf{y}_n = [y_n, hf_n, hf_{n-1}, \ldots, hf_{n+1-k}]^T.$$

Prediktor u ABM-metodi računamo prema

$$y_n^{[0]} = y_{n-1} + h\sum_{j=1}^{k} \beta_j f_{n-j},$$

dok je korektor zadan s

$$y_n^{[m+1]} = y_{n-1} + h\beta_0^* f(x_n, y_n^{[m]}) + h\sum_{j=1}^{k-1} \beta_j^* f_{n-j}, \quad m = 0, \ldots, M-1.$$

Oduzimanjem dviju uzastopnih aproksimacija dobivamo

$$y_n^{[m+1]} - y_n^{[m]} = \beta_0^*[hf(x_n, y_n^{[m]}) - hf(x_n, y_n^{[m-1]})]$$

za $m = 1, \ldots, M-1$, dok za $m = 0$ slijedi

$$y_n^{[1]} - y_n^{[0]} = h\beta_0^* f(x_n, y_n^{[0]}) + h\sum_{j=1}^{k}(\beta_j^* - \beta_j) f_{n-j} = \beta_0^*\Big[hf(x_n, y_n^{[0]}) - h\sum_{j=1}^{k}\frac{\beta_j - \beta_j^*}{\beta_0^*} hf_{n-j}\Big].$$

Ovdje smo označili $\beta_k^* = 0$. Uz oznaku

$$\delta_j = \frac{\beta_j - \beta_j^*}{\beta_0^*} \quad \text{i} \quad d_n = \sum_{j=1}^{k} h\delta_j f_{n-j},$$

možemo pisati

$$y_n^{[1]} - y_n^{[0]} = \beta_0^*[hf(x_n, y_n^{[0]}) - d_n].$$

Slično kao i vektor $\mathbf{y}_n$, definiramo vektor

$$\mathbf{y}_n^{[0]} = [y_n^{[0]}, d_n, hf_{n-1}, \ldots, hf_{n+1-k}]^T.$$

Sada vrijedi

$$\mathbf{y}_n^{[0]} = \mathbf{B}\mathbf{y}_{n-1} \tag{10.5.1}$$

gdje je

$$\mathbf{B} = \begin{bmatrix} 1 & \beta_1 & \dots & \dots & \beta_k \\ 0 & \delta_1 & \dots & \dots & \delta_k \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}.$$

Nadalje, za $m = 1, \dots, M$ definiramo vektor

$$\mathbf{y}_n^{[m]} = [y_n^{[m]}, hf(x_n, y_n^{[m-1]}, hf_{n-1}, \dots, hf_{n+1-k}]^T$$

i funkciju

$$F(\mathbf{y}_n^{[m]}) = \begin{cases} hf(x_n, y_n^{[m]}) - hf(x_n, y_n^{[m-1]}), & m = 1, \dots, M, \\ hf(x_n, y_n^{[0]} - d_n, & m = 0. \end{cases}$$

Uočimo da funkcija F koristi samo prve dvije komponente vektora $\mathbf{y}_n^{[m]}$:

$$F([u_0, u_1, \dots, u_k]^T) = hf(x_n, u_0) - u_1.$$

Sada je

$$\mathbf{y}_n^{[m+1]} = \mathbf{y}_n^{[m]} + \mathbf{c}F(\mathbf{y}_n^{[m]}), \quad m = 0, \dots, M-1, \tag{10.5.2}$$

gdje je

$$\mathbf{c} = [\beta_0^*, 1, 0, \dots, 0]^T.$$

Za $\mathbf{y}_n$ odabrat ćemo zadnju iteraciju:

$$\mathbf{y}_n = \mathbf{y}_n^{[M]}. \tag{10.5.3}$$

Za izvod Adams–Bashforthove metode iskoristili smo interpolacijski polinom P koji interpolira vrijednosti f_{n-i} u točkama x_{n-i} za $i = 1, \dots, k$ te definirali metodu pomoću

$$y_n - y_{n-1} = \int_{x_{n-1}}^{x_n} P(x)\, dx. \tag{10.5.4}$$

Definirajmo polinom

$$Q(x) = \int_{x_{n-1}}^{x} P(z)\, dz + y_{n-1}. \tag{10.5.5}$$

Zbog $Q'(x) = P(x)$ izlazi

$$Q'(x_{n-i}) = f_{n-i}, \quad i = 1, \dots, k,$$

te iz (10.5.4) i (10.5.5) slijedi

$$Q(x_{n-1}) = y_{n-1} \quad \text{i} \quad Q(x_n) = y_n.$$

Očito da iz vektora $\mathbf{y}_n$ možemo jednoznačno odrediti polinom Q, i obratno, iz polinoma Q i lagano i jednoznačno određujemo $\mathbf{y}_n$. Razvojem polinoma Q u Taylorov red oko točke x_{n-1} dobivamo

$$Q(x) = \sum_{i=0}^{k} \frac{Q^{(i)}(x_{n-1})}{i!}(x - x_{n-1})^i = \sum_{i=0}^{k} \frac{h^i}{i!} Q^{(i)}(x_{n-1}) \Big(\frac{x - x_{n-1}}{h} \Big)^i.$$

Koeficijente polinoma Q možemo zapisati vektorom $\mathbf{z}_{n-1}$:

$$\mathbf{z}_{n-1} = \Big[Q(x_{n-1}), h\frac{Q'(x_{n-1})}{1!}, h^2\frac{Q''(x_{n-1})}{2!}, \ldots, h^k\frac{Q^{(k)}(x_{n-1})}{k!} \Big]^T.$$

Važnost ovoga zapisa je u tome što su $\mathbf{y}_{n-1}$ i $\mathbf{z}_{n-1}$ linearno povezani i vrijedi:

$$\mathbf{y}_{n-1} = D\mathbf{z}_{n-1},$$

gdje je $\mathbf{D}$ regularna matrica koja ne ovisi o izboru koraka i n. Označivši još s $\mathbf{z}_{n-1}^{[m]}$:

$$\mathbf{y}_{n-1}^{[m]} = \mathbf{D}\mathbf{z}_{n-1}^{[m]}, \quad m = 0, \ldots, M,$$

korak višekoračne metode zadan s (10.5.1), (10.5.2) i (10.5.3) možemo pisati u obliku

$$\begin{aligned} \mathbf{z}_n^{[0]} &= \mathbf{D}\mathbf{B}\mathbf{D}^{-1}\mathbf{z}_{n-1}, \\ \mathbf{z}_n^{[m+1]} &= \mathbf{z}_n^{[m]} + \mathbf{D}\mathbf{c}F(\mathbf{D}^{-1}\mathbf{z}_n^{[m]}, \quad m = 0, \ldots, M-1, \\ \mathbf{z}_n &= \mathbf{z}_n^{[M]}. \end{aligned}$$

S $\mathbf{P}$ označimo matricu

$$\mathbf{P} = \mathbf{D}\mathbf{B}\mathbf{D}^{-1}.$$

Matrica $\mathbf{P}$ je gornjotrokutasta matrica čiji su elementi binomni koeficijenti:

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \ldots & 1 \\ & 1 & 2 & 3 & 4 & \ldots & k \\ & & 1 & 3 & 6 & \ldots & \\ & & & 1 & 4 & \ldots & \\ & & & & 1 & \ldots & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}.$$

Primijetimo da su prve dvije komponente vektora $\mathbf{y}_n$ i $\mathbf{z}_n$, odnosno $\mathbf{y}_{n-1}$ i $\mathbf{z}_{n-1}$ jednake:

$$Q(x_{n-1}) = y_{n-1} \quad \text{i} \quad hQ'(x_{n-1}) = hf_{n-1}.$$

Funkcija F ovisi samo o prva dva elementa vektora y_{n-1}, pa vrijedi

$$F(\mathbf{y}_n) = F(\mathbf{D}^{-1}\mathbf{z}_n) = F(\mathbf{z}_n).$$

Označimo još s

$$\mathbf{l} = \mathbf{D}\mathbf{c},$$

i višekoračnu metodu možemo pisati u obliku

$$\mathbf{z}_n^{[0]} = \mathbf{P}\mathbf{z}_{n-1}, \tag{10.5.6}$$
$$\mathbf{z}_n^{[m+1]} = \mathbf{z}_n^{[m]} + F(\mathbf{z}_n^{[m]})\mathbf{l}, \quad m = 0, \ldots, M-1, \tag{10.5.7}$$
$$\mathbf{z}_n = \mathbf{z}_n^{[M]}. \tag{10.5.8}$$

Iteracije korektora možemo napisati u obliku

$$\mathbf{z}_n = \mathbf{z}_n^{[0]} + [F(\mathbf{z}_n^{[0]}) + \cdots + F(\mathbf{z}_n^{[M-1]})]\mathbf{l}.$$

Budući da F koristi prve dvije komponente od $\mathbf{z}_n^{[m]}$, dovoljno je tijekom iteracija računati samo njih. Uz oznaku

$$e_n = F(\mathbf{z}_n^{[0]}) + \cdots + F(\mathbf{z}_n^{[M-1]})$$

prethodna relacija postaje

$$\mathbf{z}_n = \mathbf{z}_n^{[0]} + e_n\mathbf{l}. \tag{10.5.9}$$

Koeficijenti vektora $\mathbf{l}$ za Adams–Bashforth–Moultonovu metodu dani su u sljedećoj tablici.

	Red metode (k)					
	1	2	3	4	5	6
l_0	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$
l_1	1	1	1	1	1	1
l_2		$\frac{1}{2}$	$\frac{3}{4}$	$\frac{11}{12}$	$\frac{25}{24}$	$\frac{137}{120}$
l_3			$\frac{1}{6}$	$\frac{1}{3}$	$\frac{35}{72}$	$\frac{5}{24}$
l_4				$\frac{1}{24}$	$\frac{5}{48}$	$\frac{17}{96}$
l_5					$\frac{1}{120}$	$\frac{1}{40}$
l_6						$\frac{1}{720}$

Značajno je napomenuti da prediktor-korektor par izveden iz formula za deriviranje ima isti matrični prikaz s istom matricom $\mathbf{P}$ i funkcijom F, jedino se razlikuju

vektori **l**. Sljedeća tablica prikazuje koeficijente vektora **l** za BDF metodu.

	Red metode (k)					
	1	2	3	4	5	6
l_0	1	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{12}{25}$	$\frac{60}{137}$	$\frac{60}{147}$
l_1	1	1	1	1	1	1
l_2		$\frac{1}{3}$	$\frac{6}{11}$	$\frac{7}{10}$	$\frac{225}{274}$	$\frac{406}{441}$
l_3			$\frac{1}{11}$	$\frac{1}{5}$	$\frac{85}{274}$	$\frac{245}{588}$
l_4				$\frac{1}{50}$	$\frac{15}{274}$	$\frac{175}{1764}$
l_5					$\frac{1}{274}$	$\frac{7}{558}$
l_6						$\frac{1}{1764}$

Važnost ovog prikaza je u tome da vektor $\mathbf{z}_n$ čuva $k+1$ informaciju u jednoj jedinoj točki, dok prijašnji prikaz koristi isti broj informacija, ali u k različitih točaka.

Za korak integracije h vektor $\mathbf{z}_n$ je oblika

$$\mathbf{z}_n = \left[Q(x_n), Q'(x_n)\frac{h}{1!}, \ldots, Q^{(k)}(x_n)\frac{h^k}{k!}\right]^T.$$

Analogno, za korak αh imali bismo

$$\bar{\mathbf{z}}_n = \left[Q(x_n), Q'(x_n)\frac{\alpha h}{1!}, \ldots, Q^{(k)}(x_n)\frac{(\alpha h)^k}{k!}\right]^T.$$

Odavde se jasno vidi da za prijelaz s koraka h na korak αh treba i-tu komponentu vektora $\mathbf{z}_n$ pomnožiti s α^i.

Za $(k-1)$-koračnu metodu vektor $\mathbf{z}_n$ je oblika

$$\bar{\bar{\mathbf{z}}}_n = \left[Q(x_n), Q'(x_n)\frac{h}{1!}, \ldots, Q^{(k-1)}(x_n)\frac{h^{k-1}}{(k-1)!}\right]^T,$$

te ga možemo dobiti iz vektora $\mathbf{z}_n$ ispuštanjem zadnje komponente. Za prijelaz na metodu s jednim korakom više u odgovarajućem vektoru

$$\bar{\bar{\bar{\mathbf{z}}}}_n = \left[Q(x_n), Q'(x_n)\frac{h}{1!}, \ldots, Q^{(k)}(x_n)\frac{h^k}{k!}, Q^{(k+1)}(x_n)\frac{h^{k+1}}{(k+1)!}\right]^T$$

treba aproksimirati zadnju komponentu. Koristeći podijeljene razlike unazad dobivamo

$$Q^{(k+1)}(x_n) \approx y^{(k+1)}(x_n) \approx \frac{y^{(k)}(x_n) - y^{(k)}(x_{n-1})}{h}$$
$$\approx \frac{k!}{h^{k+1}}\Big(\frac{h^k y^{(k)}(x_n)}{k!} - \frac{h^k y^{(k)}(x_{n-1})}{k!}\Big) \approx \frac{k!}{h^{k+1}}(z_{n,k} - z_{n-1,k}).$$

Uočimo da su zbog oblika matrice $\mathbf{P}$ zadnje komponente vektora $\mathbf{z}_{n-1}$ i $\mathbf{z}_n^{[0]}$ jednake, te vrijedi

$$z_{n,k} - z_{n-1,k} = z_{n,k} - z_{n,k}^{[0]} = e_n l_k$$

gdje smo iskoristili prikaz (10.5.9) i s l_k označili zadnju komponentu vektora $\mathbf{l}$. Sada za aproksimaciju zadnje komponente vektora $\tilde{\tilde{\tilde{\mathbf{z}}}}_n$ možemo iskoristiti

$$\frac{h^{k+1}}{(k+1)!}Q^{(k+1)}(x_n) = \frac{e_n l_k}{k+1}.$$

Ovime smo pokazali da ovakav prikaz višekoračnih metoda omogućava laganu promjenu koraka integracije, ali i povećanje ili smanjenje broja koraka, odnosno reda metode.

Ujedno je i riješen problem početnih vrijednosti. Dovoljno je početi integraciju s jednokoračnom metodom i vektorom

$$\mathbf{z}_0 = [y_0, hf(x_0, y_0)]^T.$$

Još nam je ostalo za razmotriti izbor veličine koraka i reda metode.

Iteracije korektora u (10.5.7) nisu ništa drugo nego metoda jednostavnih iteracija za rješavanje jednadžbe

$$F(\mathbf{z}_n)\mathbf{l} = 0. \tag{10.5.10}$$

Problem se javlja kod krutih sustava. Tada iteracije konvergiraju samo ako je korak integracije h dovoljno malen. Međutim, nerealno smanjenje koraka integracije značajno povećava broj koraka integracije. Da bi se prevladao ovaj problem, jednadžbu (10.5.10) možemo rješavati Newtonovom metodom umjesto jednostavnim iteracijama:

$$\mathbf{z}_n^{[m+1]} = \mathbf{z}_n^{[m]} + WF(\mathbf{z}_n^{[m]})\mathbf{l}, \quad m = 0, \ldots, M-1, \tag{10.5.11}$$

gdje je

$$W = \Big[-\frac{\partial F}{\partial z}\Big(\mathbf{z}_n^{[m]}\Big)\mathbf{l}\Big]^{-1} = \Big[l_1 - hl_0\frac{\partial f}{\partial y}\Big]^{-1}.$$

Računanje W-a zahtijeva relativno mnogo računanja tako da se radi ubrzanja u iteracijama koristi W izračunat u prvoj iteraciji. U slučaju da nije zadovoljen kriterij konvergencije, W se ponovno računa.

Apsolutna pogreška metode dana je s

$$c_{k+1}h^{k+1}y^{(k+1)}(x_n) + \mathcal{O}(h^{k+2}),$$

pri čemu je za Adamsove metode $c_{k+1} = \gamma_k^*$ a za BDF metode je $c_{k+1} = 1/(k+1)$. Ako želimo da nam relativna točnost n-tog koraka bude manja od unaprijed zadane vrijednosti ε, tada moramo zahtijevati

$$\frac{c_{k+1}h^{k+1}|y^{(k+1)}(x_n)|}{y_\infty} \leq \varepsilon, \tag{10.5.12}$$

gdje je

$$y_\infty = \max_{0\leq i\leq} |y_i|.$$

Za aproksimaciju $y^{(k+1)}(x_n)$ koristimo stražnje diferencije

$$y^{(k+1)}(x_n) \approx \frac{y^{(k)}(x_n) - y^{(k)}(x_{n-1})}{h}.$$

Odavde slijedi

$$h^{k+1}y^{(k+1)}(x_n) \approx k!\Big(\frac{h^k}{k!}y^{(k)}(x_n) - \frac{h^k}{k!}y^{(k)}(x_{n-1})\Big). \tag{10.5.13}$$

Vrijednost $\dfrac{h^k}{k!}y^{(k)}(x_n)$ nalazi se u zadnjoj komponenti $z_{n,k}$ vektora $\mathbf{z_n}$, te (10.5.13) prelazi u

$$h^{k+1}y^{(k+1)}(x_n) \approx k!(z_{n,k} - z_{n-1,k}). \tag{10.5.14}$$

Budući da je

$$\mathbf{z_n^{[0]}} = \mathbf{P}\mathbf{z_{n-1}},$$

i jer matrica $\mathbf{P}$ ne mijenja zadnju komponentu vektora $\mathbf{z_{n-1}}$, slijedi da je

$$z_{n,k}^{[0]} = z_{n-1,k}.$$

Sada zbog (10.5.9) relaciju (10.5.14) možemo napisati u obliku

$$h^{k+1}y^{(k+1)}(x_n) \approx k!e_n l_k. \tag{10.5.15}$$

Koristeći ove aproksimacije, nejednakost (10.5.12) zamjenjujemo s

$$\frac{c_{k+1}|e_n|\,|l_k|k!}{y_\infty} \leq \varepsilon. \tag{10.5.16}$$

Uz oznake

$$D_2 = \frac{|e_n|}{y_\infty} \quad \text{i} \quad E_2 = \frac{\varepsilon}{c_{k+1}|l_k|k!},$$

kriterij (10.5.16) možemo zapisati u obliku

$$D_2 \leq E_2. \tag{10.5.17}$$

Ukoliko gornji uvjet nije zadovoljen, tj. nismo postigli traženu točnost, zadnji korak se ponavlja, ali s manjim korakom integracije:

$$h_2 = \frac{h}{p_2},$$

gdje je

$$p_2 = 1.2\Big(\frac{D_2}{E_2}\Big)^{\frac{1}{k+1}}.$$

Da smo u n-tom koraku koristili metodu jednog reda manju, dakle reda $k-1$, analogan zahtjev na relativnu pogrešku glasio bi

$$\frac{c_k h^k |y^{(k)}(x_n)|}{y_\infty} \leq \varepsilon. \tag{10.5.18}$$

Iskoristivši aproksimaciju

$$h^k y^{(k)}(x_n) \approx k! z_{n,k},$$

nejednakost (10.5.18) zamjenjujemo s

$$\frac{c_k |z_{n,k}| k!}{y_\infty} \leq \varepsilon,$$

odnosno

$$D_1 \leq E_1,$$

pri čemu je

$$D_1 = \frac{|z_{n,k}|}{y_\infty} \quad \text{i} \quad E_1 = \frac{\varepsilon}{c_k k!}.$$

Slično razmatranje možemo provesti i za slučaj da smo koristili za jedan red veću metodu, dakle reda $k+1$. Zahtjev na relativnu pogrešku glasio bi

$$\frac{c_{k+2} h^{k+2} |y^{(k+2)}(x_n)|}{y_\infty} \leq \varepsilon. \tag{10.5.19}$$

Aproksimirajući $y^{(k+2)}(x_n)$ podijeljenim razlikama unazad:

$$h^{k+2} y^{(k+2)}(x_n) \approx h^{k+1} y^{(k+1)}(x_n) - h^{k+1} |y^{(k+1)}(x_{n-1})$$

te iskoristivši (10.5.15), dobivamo

$$h^{k+2} y^{(k+2)}(x_n) \approx (e_n - e_{n-1}) l_k k!.$$

Uz oznake

$$D_3 = \frac{|e_n - e_{n-1}|}{y_\infty} \quad \text{i} \quad E_3 = \frac{\varepsilon}{c_{k+2}k!|l_k|},$$

nejednakost (10.5.19) zamjenjujemo s

$$D_3 \leq E_3.$$

Ako je u n-tom koraku zadovoljen uvjet konvergencije (10.5.17) prelazi se na izbor reda metode i koraka h u novom, $(n+1)$-om koraku. Uz konstantu p_2, pri izboru reda i koraka računaju se i konstante

$$p_1 = 1.3\Big(\frac{D_1}{E_1}\Big)^{\frac{1}{k}} \quad \text{i} \quad p_3 = 1.4\Big(\frac{D_3}{E_3}\Big)^{\frac{1}{k+2}}.$$

Ukoliko je p_1 najmanja konstanta, tada se red metode smanjuje za jedan, dok se red metode povećava za jedan ako je p_3 najmanja konstanta. Red metode se ne mijenja ako je p_2 najmanja konstanta. Za novi korak integracije uzima se

$$h_i = \frac{h}{p_i}.$$

Faktori 1.2, 1.3 i 1.4 kod računanja konstanti p_2, p_1 i p_3 su težine kojima dajemo prednost na mijenjanju i smanjivanju reda metode.

11. Optimizacija

11.1. Uvod u optimizaciju

Problem koji proučava optimizacija relativno je jednostavno opisati. Za zadanu funkciju $f : \mathbb{R}^n \to \mathbb{R}$, cilj je naći x^*, točku minimuma funkcije f:

$$x^* = \arg\min_{x \in \mathbb{R}^n} f(x). \tag{11.1.1}$$

Iako je problem jednostavno opisati, njegovo rješavanje je jedno od najtežih područja numeričke analize. Zamislimo da je naša funkcije f nadmorska visina, odnosno dubina pojedine točke na zemljinoj kugli. Naći minimum ove funkcije je zapravo traženje najveće morske dubine. Funkcija nam je poznata, u smislu da možemo dobiti njezinu vrijednost u bilo kojoj točki (npr. ehosonderom). Međutim, nije nam dostupna niti jedna dodatna informacija. Jedno rješenje je 'beskonačno' mjerenje dubina u nizu gusto izabranih točaka. Ovaj postupak očito nije prikladan, jer bismo teško izvršili toliko mjerenja u nekom razumnom vremenu.

Cilj optimizacije je definirati prikladne, tj. što brže postupke, za rješavanje ovog tipa problema. Na ovom primjeru uočimo još neke probleme s kojima ćemo se susretati. Koliko god gusto mjerili dubinu, uvijek nam ostaje nepoznata dubina između dvije susjedne točke mjerenja. Ne poznajući funkciju f, odnosno njena analitička svojstva, ne možemo reći da li između njih postoji još neka točka veće dubine.

Drugi je problem vezan uz detekciju globalnog minimuma. Ovdje je ipak područje promatranja bilo ograničeno površinom Zemljine kugle. Generalni problem optimizacije (11.1.1) vezan je uz traženje minimuma na neograničenom području. Našavši jedan minimum, nikada sa sigurnošću nećemo moći ustvrditi da je to globalni minimum jer je nemoguće provjeriti vrijednosti funkcije na cijelom neograničenom području.

Iako smo rekli da je cilj optimizacije riješiti problem (11.1.1), tj. naći minimum funkcije f, potuno analogan problem je nalaženja maksimuma funkcije f. Naime, traženje maksimuma od f ekvivalentno je traženju minimuma funkcije $-f$:

$$\max_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} -f(x).$$

To je razlog što se često umjesto termina optimizacija ravnopravno koristi termin **minimizacija funkcija**.

Često se minimum funkcije f ne traži na cijelom području $\mathbb{R}^n$, već na nekom podskupu $D \subset \mathbf{R^n}$:

$$\min_{x \in D} f(x). \tag{11.1.2}$$

U tom slučaju govorimo o problemu **minimizacije s ograničenjima**. Iako ovaj problem izgleda jednostavniji (zbog manjeg područja minimizacije), metode za njegovo rješavanje su složenije nego za polazni problem (11.1.1). Razlog leži u činjenici da uz funkciju f moramo voditi računa i o rubu područja D (često definiranog pomoću skupa funkcija). Npr., jasno je da funkcija $f(x) = x^2$ ima globalni minimum u 0. Međutim, ako tražimo minimimum na intervalu $[a, b]$ tada je minimum funkcije f jednak 0 ako je $0 \in [a, b]$ ili $\min\{f(a), f(b)\}$ ako $0 \notin [a, b]$, tj. u razmatranje smo trebali uzeti i rubove intervala.

U praksi se često javlja problem (11.1.2) s posebnim oblikom funkcije f i skupa D. Ako je D presjek poluravnina u $\mathbb{R}^n$, a f je linearna funkcija tada govorimo o **problemu linearnog programiranja**. Ukoliko je f kvadratična funkcija (polinom) tada se ovaj problem naziva **problem kvadratičnog programiranja**. Za ove probleme su razvijene posebne metode koje ovdje nećemo opisati. U narednim potpoglavljima su obrađene osnovne metode za rješavanje problema minimizacije bez ograničenja.

11.2. Metoda zlatnog reza

Prvu minimizacijsku metodu opisat ćemo za problem minimizacije realne funkcije realne varijable ($f : \mathbb{R} \to \mathbb{R}$). Neka su zadane tri točke $a < b < c$ te neka su poznate vrijednosti funkcije f u njima ($f(a)$, $f(b)$ i $f(c)$) koje zadovoljavaju $f(b) \leq f(a)$ i $f(b) \leq f(c)$ (tj. vrijednost funkcije f je najmanja u srednjoj točki b). U tom slučaju znamo da se točka lokalnog minimuma nalazi u intervalu $[a, b]$, tj. a i b omeđuju točku minimuma. Izaberimo novu točku x iz intervala $[a, b]$. Pretpostavit ćemo da je $x \in (b, c)$. Postoje dvije mogućnosti: $f(x) \geq f(b)$ ili je $f(x) < f(b)$. Ako je $f(x) \geq f(b)$ tada se minimum nalazi u intervalu $[a, x]$ (jer je $f(b) \leq f(a)$ i $f(b) \leq f(x)$) i novu točku biramo iz trojke (a, b, x). U suprotnom slučaju, minimum se nalazi u intervalu $[b, c]$ i novu točku biramo iz trojke (b, x, c). Opisani postupak je osnova metode zlatnog reza. Preostalo je još za razmotriti način izbora nove točke x.

Neka je w omjer u kojem b dijeli interval $[a, c]$:

$$w = \frac{b-a}{c-a} \quad \text{i} \quad \frac{c-b}{c-a} = 1 - w,$$

te sa z označimo

$$z = \frac{x-b}{c-a}.$$

Uočimo da je za $f(x) \geq f(b)$ minimum lociran na intervalu širine $x-a$, dok je za $f(x) < f(b)$ minimum lociran na intervalu širine $c-b$. Prvi zahtjev koji ćemo postaviti je da i u jednom i u drugom slučaju širina intervala bude jednaka:

$$x - a = c - b,$$

tj.

$$z + w = \frac{x-b+b-a}{c-a} = \frac{x-a}{c-a} = \frac{c-b}{c-a} = 1 - w.$$

Ovime dobijemo uvjet

$$z = 1 - 2w \tag{11.2.1}$$

koji znači da su točke x i b smještene simetrično s obzirom na središte intervala $[a, b]$.

Povoljnim izborom početne točke b možemo postići da x dijeli interval $[b, c]$ u istom omjeru kao što b dijeli interval $[a, c]$. Ovaj uvjet daje jednadžbu

$$\frac{c-b}{c-a} = \frac{c-x}{c-b},$$

odakle slijedi

$$1 - w = 1 - \frac{z}{1-w}.$$

Iskoristivši činjenicu da je $z = 1-2w$ (11.2.1), sređivanjem gornjeg izraza dobivamo

$$w^2 - 3w + 1 = 0.$$

Od dva rješenja gornje jednadžbe zanima nas ono koje je manje od 1 (jer je w omjer širine manjeg i većeg intervala):

$$w = \frac{3-\sqrt{5}}{2} \approx 0.38197.$$

Gornji je broj poznat kao zlatni broj pa se metoda s ovakvim izborom nove točke x naziva metodom zlatnog reza.

Algoritam za metodu zlatnog reza.

Neka točke $a^{(0)}$, $b^{(0)}$ i $c^{(0)}$ zadovoljavaju

$$\frac{b^{(0)} - a^{(0)}}{c^{(0)} - a^{(0)}} = w = \frac{3-\sqrt{5}}{2}$$

te

$$f(b^{(0)}) \leq f(a^{(0)}) \quad \text{i} \quad f(b^{(0)}) \leq f(c^{(0)}).$$

Opisani postupak možemo sada sažeto zapisati.

1. $i = 0$
2. Odredimo $x^{(i)} = c^{(i)} + a^{(i)} - b^{(i)}$.
3. Ako je $f(x^{(i)}) < f(b^{(i)})$ **tada**
   $a^{(i+1)} = b^{(i)}, \quad b^{(i+1)} = x^{(i)}, \quad c^{(i+1)} = c^{(i)},$
   inače
   $a^{(i+1)} = a^{(i)}, \quad b^{(i+1)} = b^{(i)}, \quad c^{(i+1)} = x^{(i)}.$
4. Ako je $|c^{(i+1)} - a^{(i+1)}| \leq \varepsilon$ **tada**
   $x^* = (a^{(i+1)} + c^{(i+1)})/2,$
   KRAJ.
5. $i = i + 1$
6. Vrati se na korak 2.

Uočimo da vrijedi

$$|c^{(i+1)} - a^{(i+1)}| = (1 - w)|c^{(i)} - a^{(i)}| \approx 0.61803|c^{(i)} - a^{(i)}|$$

te metoda konvergira linearno prema točki minimuma x^*.

Za inicijalno određivanje trojke $(a^{(0)}, b^{(0)}, c^{(0)})$ krenemo od bilo kojeg para točaka a i b. Bez smanjenja općenitosti pretpostavimo da je $f(a) \geq f(b)$. Točku c odredimo tako da b dijeli interval $[a, c]$ u zlatnom omjeru. Ukoliko je $f(b) \leq f(c)$ našli smo traženu trojku. U suprotnom slučaju ponovimo postupak s točkama a i c (ili b i c).

11.3. Višedimenzionalna minimizacija

Ovdje ćemo promatrati metode koje minimiziraju funkciju $f : \mathbb{R}^n \to \mathbb{R}$:

$$\min_{x \in \mathbb{R}^n} f(x). \tag{11.3.1}$$

Cijeli niz metoda zasnovan je na ideji da se za neku zadanu točku $x_0 \in \mathbb{R}^n$ odredi točka $x_1 \in \mathbb{R}^n$ takva da je $f(x_1) < f(x_0)$. Ponavljanjem ovog postupka za točku $x_i \in \mathbb{R}^n$, tražimo točku $x_{i+1} \in \mathbb{R}^n$ za koju je $f(x_{i+1}) < f(x_i)$. Na ovaj način generiramo niz točaka $x_0, x_1, x_2, \ldots$ za koje je $f(x_0) > f(x_1) > f(x_2) > \cdots$. Ako je niz $(f(x_i))_i$ odozdo ograničen, tada postoji $\lim_i f(x_i)$. No, $\lim_i x_i$ u tom slučaju ne treba nužno postojati. Na primjer, za $f(x) = e^{-x}$ i $x_i = i$ je $\lim_i f(x_i) = 0$, no $\lim_i x_i$ ne postoji. Za egzistenciju $\lim_i x_i$ nužno je da postoji $x_0 \in \mathbb{R}^n$ takav da je skup

$$\{x \mid f(x) \leq f(x_0)\}$$

kompaktan.

Osnovno je pitanje kako izabrati x_{i+1}. Neka je s smjer u kojem funkcija f lokalno pada u okolini točke x, tj.

$$f(x + \lambda s) < f(x), \quad \text{za mali } \lambda, \quad \lambda \in \mathbb{R}, \quad \lambda > 0.$$

Promatrajmo funkciju $g : \mathbb{R} \to \mathbb{R}$ definiranu s

$$g(\lambda) = f(x + \lambda s).$$

Uočimo da je $g(0) = f(x)$. Ako je f glatka funkcija ($f \in C^1$), tada je i $g \in C^1$. U tom slučaju uvjet da f pada u smjeru s oko točke x odgovara uvjetu da funkcija g pada u 0, tj. da je $g'(0) < 0$. Kako je

$$g'(\lambda) = \langle \nabla f(x + \lambda s), s \rangle,$$

gornji uvjet prelazi u

$$g'(0) = \langle \nabla f(x), s \rangle < 0.$$

Na osnovu ovog, za funkciju f i točku definiramo skup

$$D(x) = \{s \mid \langle \nabla f(x), s \rangle < 0\}. \tag{11.3.2}$$

Ovaj skup nazivamo **skup smjerova silaska** a njegove elemente **smjerovi silaska**.

Sada jednostavno možemo opisati jednu klasu minimizacijskih metoda. Za zadani $x_i \in \mathbb{R}^n$ definiramo

$$x_{i+1} = x_i + \lambda_i s_i \tag{11.3.3}$$

gdje je $s_i \in D(x_i)$. Konstantu λ_i iz (11.3.3) nazivamo **korakom** minimizacije. Odabiremo je tako da bude zadovoljeno $f(x_{i+1}) < f(x_i)$. Iz prethodnog razmatranja znamo da je to moguće jer je s_i smjer silaska. Izbor smjera silaska ovisi o izboru metode minimizacije. Različite metode na različite načine određuju izbor smjera silaska.

Veličina koraka minimizacije je bitna za konvergenciju metode, jer nije dovoljno odrediti točku x_{i+1} koja zadovoljava $f(x_{i+1}) < f(x_i)$. Jednostavan primjer je minimizacija funkcije $f(x) = x^2$. Izborom

$$x_0 = 2, \quad x_1 = 1.5, \quad \ldots, \quad x_i = 1 + \frac{1}{i+1},$$

dobivamo niz točaka $(x_i)_i$ koji konvergira i niz padajućih vrijednosti $(f(x_i))_i$ koje također konvergiraju. Međutim, ovi nizovi ne konvergiraju prema točki minimuma, odnosno minimumu funkcije f.

Jedan od načina izbora koraka λ_i je 'maksimalno spuštanje' u smjeru s_i. Tada je

$$\lambda_i = \arg\min_{\lambda > 0} f(x_i + \lambda s_i).$$

Međutim, ovakav pristup zahtijeva primjenu analitičkih metoda što nije jednostavno pri upotrebi računala a naročito ako je funkcija f složenijeg oblika.

Drugi način je približno određivanje λ_i nekom minimizacijskom metodom. U tom slučaju koristimo metode za jednodimenzionalnu minimizaciju (npr. prije opisanu metodu zlatnog reza).

Kako točka x_{i+1} nije nužno točka (globalnog ili lokalnog) minimuma mi u njoj ponavljamo cijeli postupak: izabiremo novi smjer silaska i korak minimizacije. Stoga korak minimizacije nije nužno izračunati egzaktno ili pak s prevelikom točnošću jer utrošeno vrijeme ne rezultira s odgovarajućim rezultatom. Jedan od alternativnih pristupa je 'neegzaktno pretraživanje po pravcu'. Ova metoda u konačnom broju koraka određuje λ_i koji ne minimizira $f(x_i + \lambda s_i)$, ali ipak dovoljno smanjuje vrijednost $f(x_{i+1})$ u odnosu na $f(x_i)$ tako da minimizacijska metoda konvergira, tj. da niz $(x_i)_i$ konvergira točki minimuma funkcije f. Ovaj izbor koraka minimizacije biti će detaljnije objašnjen kasnije.

11.3.1. Gradijentna metoda

Ovo je najjednostavnija metoda iz klase. Uočimo da je $-\nabla f(x)$ smjer silaska u točki x zbog

$$\langle -\nabla f(x), \nabla f(x) \rangle = -\|\nabla f(x)\|^2 < 0$$

za $\nabla f(x) \neq 0$. Ako je $\nabla f(x) = 0$ tada je, obično, zadovoljen kriterij konvergencije (u tom je slučaju skup smjerova silaska prazan skup). Važno je napomenuti da u općem slučaju to ne treba značiti da smo došli do točke minimuma.

Iako ova metoda lokalno bira smjer najbržeg silaska, zato gradijentnu metodu ponekad zovu i **metoda najbržeg silaska**, ta metoda nije najbrža u traženju globalnog minimuma. Za primjenu ove metode funkcija f treba biti glatka ($f \in C^1$), što je nedostatatak u odnosu na metode koje zahtjevaju samo neprekidnost funkcije koja se minimizira, ali je i prednost u odnosu na neke brže metode koje zahtijevaju i veću glatkoću funkcije f (npr. modificirana Newtonova metoda, zahtijeva $f \in C^2$).

11.3.2. Modificirana Newtonova metoda

Ova se metoda zasniva na Newton–Raphsonovoj metodi za rješavanje jednadžbe

$$\nabla f(x) = 0.$$

Ovo je nužan uvjet koji točka minimuma mora zadovoljavati. U slučaju da je f konveksna funkcija, tada je taj uvjet i dovoljan.

Newton-Raphsonova metoda generira niz točaka iteracijama:

$$x_{i+1} = x_i - [\nabla^2 f(x_i)]^{-1} \nabla f(x_i), \tag{11.3.4}$$

gdje je $\nabla^2 f$ Hessijan (ili Hessova matrica) funkcije f,

$$[\nabla^2 f]_{ij} = \frac{\partial^2 f}{\partial y_i \partial y_j}.$$

Ono što u ovom trenutku nije jasno je li ispunjeno $f(x_{i+1}) < f(x_i)$.

Prvo pogledajmo je li

$$-[\nabla^2 f(x_i)]^{-1} \nabla f(x_i) \in D(x_i).$$

Neka je H pozitivno definitna matrica ($(Hx, x) > 0$ za sve $x \in \mathbb{R}^n$ i $x \neq 0$). Tada je

$$\langle -H\nabla f(x), \nabla f(x)\rangle = -\langle H\nabla f(x), \nabla f(x)\rangle < 0$$

za $\nabla f(x) \neq 0$. Znači, za pozitivno definitnu matricu H je

$$-H\nabla f(x) \in D(x).$$

Sada treba primijetiti da je za x^*, točku minimuma funkcije f, matrica $\nabla^2 f(x^*)$ pozitivno definitna. No, tada je i $\nabla^2 f(x)$ pozitivno definitna za x iz neke okoline točke minimuma x^*. Kako je tada i $[\nabla^2 f(x)]^{-1}$ pozitivno definitna, onda je

$$-[\nabla^2 f(x)]^{-1} \nabla f(x) \in D(x)$$

za x u okolini točke minimuma.

Ovo još ne garantira da niz generiran s (11.3.4) zadovoljava $f(x_{i+1}) < f(x_i)$. Uočimo da je u (11.3.4) $\lambda_i = 1$. Ako za smjer silaska izaberemo

$$s_i = -[\nabla^2 f(x_i)]^{-1} \nabla f(x_i),$$

a u iteracijama

$$x_{i+1} = x_i - \lambda_i [\nabla^2 f(x_i)]^{-1} \nabla f(x_i),$$

korak minimizacije λ_i biramo na jedan od prije opisanih načina, definirali smo tzv. **modificiranu Newtonovu metodu** (modifikacija je uvođenje koraka minimizacije λ_i u Newtonovu metodu (11.3.4)).

Za primjenu modificirane Newtonove metode nužno je da je funkcija $f \in C^2$. Metoda je primjenjiva samo na području gdje je $\nabla^2 f$ pozitivno definitna. Ako f nije konveksna funkcija, tada je modificirana Newtonova metoda primjenjiva samo u okolini točke minimuma, a ne na cijelom području minimizacije.

U odnosu na gradijentnu metodu, modificirana Newtonova metoda zahtijeva konveksnost i veću glatkoću funkcije f. Nadalje, u svakom je koraku minimizacije nužno izračunati Hessijan, te riješiti sustav

$$\nabla^2 f(x_i)(x_{i+1} - x_i) = -\nabla f(x_i)$$

(ovdje nema potrebe za invertiranjem Hessijana). Radi ubrzanja metode, često se Hessijan ne računa u svakoj iteraciji. S druge strane, prednost metode je brža konvergencija generiranog niza $(x_i)_i$ k točki minimuma.

Da bismo ilustrirali prednost modificirane Newtonove metode u odnosu na gradijentnu metodu, usporedit ćemo ove dvije metode na jednome jednostavnom primjeru. Neka je A pozitivno definitna matrica. Minimizirat ćemo kvadratnu formu

$$f(y) = \frac{1}{2}\langle Ay, y\rangle + \langle b, y\rangle + c.$$

Za ovu funkciju, egzaktni minimum jednostavno se pronalazi. Gradijent funkcije f je dan s

$$\nabla f(y) = Ay + b,$$

te iz nužnog i dovoljnog uvjeta za minimum $\nabla f(y) = 0$ dobijemo da f dostiže minimum u

$$x^* = -A^{-1}b.$$

Primjenom modificirane Newtonove metode

$$x_{i+1} = x_i - \lambda_i[\nabla^2 f(x_i)]^{-1}\nabla f(x_i),$$

uz egzaktan izbor koraka

$$\lambda_i = \arg\min_{\lambda>0} f(x_i - \lambda[\nabla^2 f(x_i)]^{-1}\nabla f(x_i)) \tag{11.3.5}$$

dobivamo iteracije

$$x_{i+1} = x_i - \lambda_i A^{-1}(Ax_i + b) = (1-\lambda)x_i - A^{-1}b.$$

Ovdje smo iskoristili da je $\nabla^2 f(x_i) = A$. Odatle se lako vidi da za bilo koji izbor početne točke x_0 modificirana Newtonova metoda daje egzaktno rješenje u jednom koraku. Egzaktni izbor koraka (11.3.5) daje $\lambda_0 = 1$.

Primjena gradijentne metode na ovom primjeru pokazuje da ova metoda općenito ne dolazi do minimuma u konačnom broju koraka. Za ilustraciju promatrat ćemo dvodimenzionalni problem minimizacije funkcije

$$f(y) = \frac{1}{2}\langle Ay, y\rangle = \frac{1}{2}(\alpha_1 y_1^2 + \alpha_2 y_2^2),$$

gdje je $0 < \alpha_1 < \alpha_2$. Matrica A je oblika

$$A = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix}.$$

Ako je $\alpha_1 = \alpha_2 > 0$, lagano se vidi da, u tom slučaju, gradijentna metoda u jednom koraku dolazi do točke minimuma

$$x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Izaberimo početni vektor

$$x_0 = \begin{bmatrix} x_{0,1} \\ x_{0,2} \end{bmatrix}$$

s komponentama $x_{0,1}$ i $x_{0,2}$ različitim od nule. Ako bi jedna komponenta bila 0, tada bismo, također, u jednom koraku došli do minimuma. Pretpostavimo sada da je gradijentna metoda u konačnom broju koraka, recimo $k+1$, došla do točke minimuma. Tada je

$$x_{k+1} = x_k - \lambda_k A x_k = 0,$$

odnosno

$$x_{k+1,1} = (1 - \lambda_k \alpha_1) x_{k,1} = 0, \qquad x_{k+1,2} = (1 - \lambda_k \alpha_2) x_{k,2} = 0.$$

Budući da je $\alpha_1 \neq \alpha_2$, barem jedna od komponenti vektora x_k treba biti jednaka nuli. Pokazat ćemo da je to nemoguće ukoliko su komponente od x_0 različite od 0, te ukoliko koristimo egzaktni izbor koraka u gradijentnoj metodi. Za

$$x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix}$$

vrijedi za egzaktni izbor koraka λ_i

$$x_{i+1,1} = \frac{\alpha_2^2(\alpha_2 - \alpha_1) x_{i,1} x_{i,2}^2}{\alpha_1^3 x_{i,1}^2 + \alpha_2^3 x_{i,2}^2}, \qquad x_{i+1,2} = \frac{\alpha_1^2(\alpha_1 - \alpha_2) x_{i,2} x_{i,1}^2}{\alpha_1^3 x_{i,1}^2 + \alpha_2^3 x_{i,2}^2}.$$

Ukoliko su $x_{i,1} \neq 0$ i $x_{i,2} \neq 0$ zbog $\alpha_1 > \alpha_2 > 0$ vrijedi $x_{i+1,1} \neq 0$ i $x_{i+1,2} \neq 0$, pa indukcijom zaključujemo da niti u jednom koraku ne dolazimo do točke minimuma.

11.4. Kvazi–Newtonove metode

Prethodna usporedba gradijentne i modificirane Newtonove metode pokazuje da je modificirana Newtonova metoda brža od gradijentne, tj. do minimuma kvadratne forme dolazi u konačnom (jednom) broju minimizacijskih koraka. To plaćamo većim brojem računskih operacija u svakom koraku (za računanje Hessijana te rješavanje sustava jednadžbi).

Jedna cijela klasa minimizacijskih metoda nastala je na ideji da se iskoristi jednostavnost gradijentne i brzina modificirane Newtonove metode. S druge strane,

modificirana Newtonova metoda konvergira samo u području u kojem je funkcija koju minimiziramo konveksna, odnosno Hessijan pozitivno definitan. Izvan tog područja ne možemo ništa reći o konvergenciji metode. Metode koje ćemo ovdje opisati rješavaju i taj problem. U okolini minimuma se ponašaju kao modificirana Newtonova metoda, a izvan tog područja još uvijek garantiraju konvergenciju.

Primjena ovih metoda na problem minimizacije kvadratne forme pokazat će da se do minimuma dolazi u konačnom broju koraka, ne većem od dimenzije (broja varijabli) problema. Tako i ovaj kriterij svrstava metode koje ćemo promatrati, po efikasnosti, između gradijentne i modificirane Newtonove metode.

Već smo pokazali da je za pozitivno definitnu matricu H i $-H\nabla f(x)$ smjer silaska. Ovdje ćemo promatrati metode za koje je smjer silaska definiran s

$$s_i = -H_i \nabla f(x_i), \tag{11.4.1}$$

gdje je H_i pozitivno definitna matrica. Ako je H_i inverz Hessijana u x_i onda dobivamo modificiranu Newtonovu metodu. No, ovdje su od interesa metode koje matricu H_{i+1} relativno jednostavno računaju iz prethodne matrice H_i. Ovim pristupom ne trebamo u svakom koraku rješavati sustav $\nabla^2 f(x_i)s_i = -\nabla f(x_i)$, već u svakom koraku množimo matricu H_i s vektorom $\nabla f(x_i)$, što je daleko brže.

U svakom koraku matricu H_{i+1} generirat ćemo tako da bude zadovoljeno

$$H_{i+1}(\nabla f(x_{i+1}) - \nabla f(x_i)) = x_{i+1} - x_i. \tag{11.4.2}$$

Ovaj uvjet na H_{i+1} sličan je uvjetu koji zadovoljava Hessijan (koji se koristi u modificiranoj Newtonovoj metodi):

$$\nabla f(x_{i+1}) - \nabla f(x_i) \approx \nabla^2 f(x_{i+1})(x_{i+1} - x_i).$$

Metode koje za izbor smjera silaska koriste (11.4.1), pri čemu H_{i+1} zadovoljava ralaciju (11.4.2), nazivamo **kvazi-Newtonove metode** (često se koristi i naziv **metode promjenjive metrike**). Za niz matrica koje zadovoljavaju (11.4.2) može se pokazati da vrijedi

$$\lim_{i\to\infty} H_i = [\nabla^2 f(x^*)]^{-1},$$

gdje je x^* točka minimuma. Ovo dodatno opravdava naziv metode. Sada ćemo vidjeti kako računati H_i da bi vrijedilo (11.4.2).

Označimo

$$q_i = \nabla f(x_{i+1}) - \nabla f(x_i) \quad \text{i} \quad p_i = x_{i+1} - x_i.$$

Tražimo pozitivno definitnu matricu H_{i+1} koja zadovoljava

$$H_{i+1} q_i = p_i.$$

Htjeli bismo da je matrica H_{i+1} 'lako izračunljiva' iz matrice H_i. Radi jednostavnosti, u daljnjem računu nećemo koristiti indekse:

$$H = H_i, \quad p = p_i, \quad q = q_i, \quad H' = H_{i+1}.$$

Uz ovaj dogovor uvjet (11.4.2) možemo zapisati u obliku

$$H'q = p.$$

Pod pojmom da je H' 'lako izračunljiva' podrazumijevat ćemo da H' dobijemo iz H pribrajanjem neke relativno jednostavne matrice E:

$$H' = H + E.$$

Sada smo problem sveli na određivanje matrice E koja zadovoljava

$$(H + E)q = p,$$

odnosno

$$Eq = p - Hq.$$

Matrice H i H' su simetrične matrice, pa je i matrica E simetrična. Najjednostavnija simetrična matrica M koja zadovoljava

$$Mx = y \tag{11.4.3}$$

za zadane vektore x i y ima oblik

$$M = \frac{yy^T}{y^T x} = \frac{yy^T}{\langle y, x\rangle}.$$

Lako se vidi da ovako definirana matrica M zadovoljava (11.4.3):

$$Mx = \frac{yy^T}{\langle y, x\rangle}\, x = \frac{1}{\langle y, x\rangle}\, y(y^T x) = \frac{1}{\langle y, x\rangle}\, y\, \langle y, x\rangle = y.$$

Tako za

$$A = \frac{pp^T}{p^T q}$$

vrijedi $Aq = p$, a za

$$B = \frac{Hq(Hq)^T}{(Hq)^T q} = \frac{Hqq^T H}{q^T Hq}$$

vrijedi

$$Bq = \frac{Hq(Hq)^T q}{(Hq)^T q} = Hq.$$

Na osnovu ovoga, lako zaključimo da matrica E definirana s

$$E = A - B = \frac{pp^T}{p^T q} - \frac{Hqq^T H}{q^T Hq}$$

zadovoljava $Eq = p - Hq$.

Ovime smo pokazali da matrica H_{i+1} definirana s

$$H_{i+1} = H_i + \frac{p_i p_i^T}{p_i^T q_i} - \frac{H_i q_i q_i^T H_i}{q_i^T H_i q_i}, \tag{11.4.4}$$

gdje je $q_i = \nabla f(x_{i+1}) - \nabla f(x_i)$, a $p_i = x_{i+1} - x_i$, zadovoljava kvazi-Newtonovu jednadžbu. Ova metoda poznata je pod nazivom '**DFP metoda**' (Davidon, Fletcher, Powel).

Sada ćemo pokazati da je H_{i+1} dobro definirana, tj. da su nazivnici u (11.4.4) različiti od 0. Štoviše, vrijedi

$$p_i^T q_i > 0 \quad \text{i} \quad q_i^T H_i q_i > 0$$

ako je $\nabla f(x_i) \neq 0$. Prvo pokažimo da su kod egzaktnog izbora koraka

$$f(x_i + \lambda_i s_i) = \min_{\lambda \geq 0} f(x_i + \lambda s_i)$$

vektori s_i i $\nabla f(x_{i+1})$ okomiti. Budući da je λ_i točka minimuma, vrijedi

$$0 = \frac{d}{d\lambda} f(x_i + \lambda s_i) \Big|_{\lambda=\lambda_i} = \langle \nabla f(x_i + \lambda_i s_i), s_i \rangle = \langle \nabla f(x_{i+1}), s_i \rangle.$$

Ako korak ne biramo egzaktno, već nekom numeričkom metodom, gornja jednakost neće vrijediti. U tom slučaju je jednoznačno određen $\mu_i \neq 0$ za koji je

$$\langle \nabla f(x_{i+1}), s_i \rangle = \mu_i \langle \nabla f(x_i), s_i \rangle.$$

Za egzaktan izbor koraka je $\mu_i = 0$, dok za neegzaktan izbor očekujemo da je $\mu_i \approx 0$.

Primjenom gornjih oznaka dobivamo

$$\begin{aligned} p_i^T q_i &= \langle \nabla f(x_{i+1}) - \nabla f(x_i), x_{i+1} - x_i \rangle = \langle \nabla f(x_{i+1}) - \nabla f(x_i), \lambda_i s_i \rangle \\ &= \lambda_i [\langle \nabla f(x_{i+1}), s_i \rangle - \langle \nabla f(x_i), s_i \rangle] = \lambda_i [\mu_i \langle \nabla f(x_i), s_i \rangle - \langle \nabla f(x_i), s_i \rangle] \\ &= \lambda_i (\mu_i - 1) \langle \nabla f(x_i), s_i \rangle = -\lambda_i (\mu_i - 1) \langle \nabla f(x_i), H_i \nabla f(x_i) \rangle. \end{aligned}$$

Budući da je H_i pozitivno definitna matrica i $\nabla f(x_i) \neq 0$, gornji skalarni produkt je strogo pozitivan. Uz uvjet da je $\mu_i < 1$ vrijedit će $p_i^T q_i > 0$. Ovaj uvjet nije veliko ograničenje jer, kako smo već spomenuli, kod neegzaktnog izbora koraka očekujemo da je $\mu_i \approx 0$.

Zbog pozitivne definitnosti matrice H_i također vrijedi

$$q_i^T H_i q_i \geq 0.$$

Gornji izraz jednak je 0 samo ukoliko je i $q_i = 0$ odnosno $\nabla f(x_{i+1}) = \nabla f(x_i)$. No, u tom slučaju je i

$$\langle \nabla f(x_{i+1}), s_i \rangle = \langle \nabla f(x_i), s_i \rangle$$

te je $\mu_i = 1$. Znači, i u ovom slučaju $\mu_i < 1$ garantira da je

$$q_i^T H_i q_i > 0.$$

Ovime smo pokazali da je matrica H_{i+1} dobro definirana.

Matrica H_i je pozitivno definitna pa prema tome i simetrična. Iz definicije matrice H_{i+1} (11.4.4) jasno je da je i ona simetrična, no pozitivna definitnost nije tako očita. Uzmimo proizvoljan vektor $y \in \mathbb{R}^n$, $y \neq 0$. Kako je H_i pozitivno definitna, možemo je zapisati u obliku

$$H = LL^T,$$

gdje je L regularna donjetrokutasta matrica. Korištenjem oznaka $u = L^T y$ i $v = L^T q_i$, izlazi

$$\langle H_{i+1} y, y \rangle = u^T u + \frac{(p^T y)^2}{p^T q} - \frac{V^T u)^2}{v^T v}. \tag{11.4.5}$$

Kako je $p^T q > 0$, vrijedi

$$\frac{(p^T y)^2}{p^T q} \geq 0. \tag{11.4.6}$$

S druge strane vrijedi

$$u^T u - \frac{(V^T u)^2}{v^T v} = \|u\|^2 - \frac{\langle u, v \rangle^2}{\|v\|^2} = \frac{\|u\|^2 \|v\|^2 - \langle u, v \rangle^2}{\|v\|^2} \geq 0. \tag{11.4.7}$$

Gornja nejednakost je posljedica Schwartzove nejednakosti. Ovime smo pokazali da je izraz (11.4.5) nenegativan. Sada ćemo pokazati da je

$$\langle H_{i+1} y, y \rangle \neq 0.$$

Kad bi bilo $\langle H_{i+1} y, y \rangle = 0$, onda bi zbog (11.4.6) i (11.4.7) moralo vrijediti i

$$\|u\|^2 \|v\|^2 - \langle u, v \rangle^2 = 0,$$

a to vrijedi samo ako su vektori u i v kolinearni: $u = \alpha v$. No, tada je $y = \alpha q$, a $\alpha \neq 0$ jer je $y \neq 0$. Nadalje, zbog $\langle H_{i+1} y, y \rangle = 0$ treba vrijediti i

$$0 = \frac{(p^T y)^2}{p^T q} = \frac{\langle p, \alpha q \rangle^2}{(p, q)} = \alpha^2 \langle p, q \rangle.$$

Ovo je pak u kontradikciji s prije dokazanom činjenicom da je $0 < p^T q = \langle p, q \rangle$.

Ovime smo dokazali sljedeći teorem.

Teorem 11.4.1 *Neka je H_i pozitivno definitna matrica i $\nabla f(x_i) \neq 0$, te neka je korak minimizacije određen tako da je $\mu_i < 1$. Tada je matrica H_{i+1} definirana s (11.4.4) dobro definirana i pozitivno definitna te zadovoljava kvazi-Newtonovu jednadžbu (11.4.2).*

Gore opisana metoda može se generalizirati složenijim iteracijskim postupkom izbora matrice H_{i+1}. Neka je kao i prije $q_i = \nabla f(x_{i+1}) - \nabla f(x_i)$ i $p_i = x_{i+1} - x_i$. Za parametre

$$\gamma_i > 0, \quad \theta_i \geq 0$$

definiramo rekurziju oblika

$$H_{i+1} = \Psi(\gamma_i, \theta_i, H_i, p_i, q_i),$$

gdje je funkcija Ψ zadana s

$$\Psi(\gamma, \theta, H, p, q) = \gamma H + \Big(1 + \gamma\theta \frac{q^T H q}{p^T q}\Big) \frac{p p^T}{p^T q} - \gamma \frac{(1-\theta)}{q^T H q} H q \cdot q^T H - \frac{\gamma\theta}{p^T q} (p q^T H + H q p^T). \quad (11.4.8)$$

Funkcija Ψ je definirana jedino ako je $p^T q \neq 0$ i $q^T H q \neq 0$. Uočimo da je matrica H_{i+1} dobivena iz H_i dodavanjem korekcije reda ne većeg od 2 matrici $\gamma_i H_i$:

$$\operatorname{rank}(H_{i+1} - H_i) \leq 2.$$

Stoga se kaže da (11.4.8) definira **metodu reda dva**.

Metoda (11.4.8) obuhvaća nekoliko poznatih metoda kao specijalne slučajeve:

(a) $\gamma_i = 1$, $\theta_i = 0$; prije opisana 'DFP metoda';

(b) $\gamma_i = 1$, $\theta_i = 1$; Broyden, Fletcher, Goldfarb i Shannova metoda reda 2 ('BFGS metoda');

(c) $\gamma_i = 1$, $\theta_i = p_i^T q_i / (p_i^T q_i - q_i^T H_i q_i)$; Broydenova simetrična metoda reda jedan.

Zadnja metoda definirana je jedino u slučaju kad je $p_i^T q_i \neq q_i^T H_i q_i$. Ovdje je moguće da vrijedi $\theta_i < 0$ te u tom slučaju H_{i+1} može biti indefinitna čak i kad je H_i pozitivno definitna. Ako zamijenimo vrijednost θ_i u (11.4.8) dobivamo

$$H_{i+1} = H_i + \frac{z_i z_i^T}{\alpha_i}, \qquad z_i = p_i - H_i q_i, \qquad \alpha_i = p_i^T q_i - q_i^T H_i q_i,$$

što objašnjava zašto se ova metoda naziva metodom reda jedan. Za metode definirane s (11.4.8) vrijedi teorem analogan Teoremu 11.4.1.

Teorem 11.4.2 *Ako postoji $i \geq 0$ za koji je H_i pozitivno definitna matrica,*

$$\nabla f(x_i) \neq 0$$

i ako je korak minimizacije određen tako da je $\mu_i < 1$, tada je za sve $\gamma_i > 0$ i $\theta_i \geq 0$ matrica $H_{i+1} = \Psi(\gamma_i, \theta_i, H_i, p_i, q_i)$ dobro definirana i pozitivno definitna te zadovoljava kvazi-Newtonovu jednadžbu (11.4.2).

Na kraju razmatranja kvazi-Newtonovih metoda, pokazat ćemo da metoda definirana s (11.4.8) dolazi do točke minimuma kvadratne funkcije $f : \mathbb{R}^n \to \mathbb{R}$ u najviše n koraka, uz uvjet da je korištena egzaktna minimizacija pri izboru koraka minimizacije. Ovaj rezultat nam sugerira da će ova metoda brzo konvergirati i u slučaju minimizacije nekvadratične funkcije.

Teorem 11.4.3 *Neka je*

$$f(x) = \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle + c$$

kvadratna forma gdje je A $n \times n$ pozitivno definitna matrica. Nadalje, neka je $x_0 \in \mathbb{R}^n$ i H_0 $n \times n$ pozitivno definitna matrica. Ako je za minimizaciju $f(x)$ korištena kvazi-Newtonova metoda definirana s (11.4.8), uz egzaktan izbor koraka s početnim vrijednostima x_0 i H_0, tada generirani nizovi $(x_i)_i$, $(H_i)_i$, $(\nabla f(x_i))_i$, $(p_i)_i = (x_{i+1} - x_i)_i$ i $(q_i)_i = (\nabla f(x_{i+1}) - \nabla f(x_i))_i$ imaju sljedeća svojstva:

(a) Postoji najmanji indeks $m \leq n$ za koji je $x_m = x^* = -A^{-1}b$ minimum od f i $\nabla f(x_m) = 0$.

(b) $\langle p_i, q_k\rangle = \langle p_i, Ap_k\rangle = 0$ za $0 \leq i \neq k \leq m-1$, $\langle p_i, q_i\rangle > 0$ za $0 \leq i \leq m-1$. (Drugim riječima, vektori p_i su A-konjugirani.)

(c) $\langle p_i, \nabla f(x_k)\rangle = 0$ za sve $0 \leq i < k \leq m$.

(d) $H_k q_i = \gamma_{i,k} p_i$ za $0 \leq i < k \leq m$, gdje je

$$\gamma_{i,k} = \begin{cases} \gamma_{i+1}\gamma_{i+2}\cdots\gamma_{k-1}, & za\ i < k-1, \\ 1 & za\ i = k-1. \end{cases}$$

(e) Ako je $m = n$ tada dodatno vrijedi

$$H_m = H_n = PDP^{-1}A^{-1},$$

gdje je $D = \operatorname{diag}(\gamma_{0,n}, \gamma_{1,n}, \ldots, \gamma_{n-1,n})$, $P = (p_0, p_1, \ldots, p_{n-1})$. Ako je i $\gamma_i = 1$ tada je $H_n = A^{-1}$.

Dokaz. Promatrajmo sljedeće uvjete za proizvoljan indeks $l \geq 0$:

$$\langle p_i, q_k\rangle = \langle p_i, Ap_k\rangle = 0 \quad \text{za} \quad 0 \leq i \neq k \leq l-1, \tag{11.4.9}$$
$$\langle p_i, q_i\rangle > 0 \quad \text{za} \quad 0 \leq i \leq l-1, \tag{11.4.10}$$
$$H_l \text{ je pozitivno definitna,} \tag{11.4.11}$$
$$\langle p_i, q_k\rangle = 0 \quad \text{za sve} \quad 0 \leq i < k \leq l, \tag{11.4.12}$$
$$H_k q_i = \gamma_{i,k} p_i \quad \text{za} \quad 0 \leq i < k \leq l. \tag{11.4.13}$$

Ako su ovi uvjeti zadovoljeni za l te ako je $\nabla f(x_l) \neq 0$ tada ćemo pokazati da su uvjeti zadovoljeni i za $l+1$.

Kako je H_l pozitivno definitna (pretpostavka (11.4.11)), iz $\nabla f(x_l) \neq 0$ direktno slijedi

$$\langle H_l \nabla f(x_l), \nabla f(x_l)\rangle > 0 \quad \text{i} \quad s_l = H_l \nabla f(x_l) \neq 0.$$

Minimizacija po pravcu je egzaktna, pa je λ_l nultočka od

$$0 = \langle \nabla f(x_{l+1}), s_l\rangle = \langle \nabla f(x_l) - \lambda_l A s_l, s_l\rangle, \qquad \lambda_l = \frac{\langle H_l \nabla f(x_l), \nabla f(x_l)\rangle}{\langle A s_l, s_l\rangle}.$$

Stoga je $p_l = -\lambda_l s_l \neq 0$ i

$$\begin{aligned} \langle p_l, \nabla f(x_{l+1})\rangle &= -\lambda_l \langle s_l, \nabla f(x_{l+1})\rangle = 0, \\ \langle p_l, \nabla f(x_l)\rangle &= -\lambda_l \langle s_l, \nabla f(x_{l+1}) - \nabla f(x_l)\rangle = \lambda_l \langle s_l, \nabla f(x_l)\rangle \\ &= \lambda_l \langle H_l \nabla f(x_l), \nabla f(x_l)\rangle > 0. \end{aligned} \tag{11.4.14}$$

Prema Teoremu 11.4.2, H_{l+1} je pozitivno definitna. Nadalje, zbog $Ap_i = q_i$, (11.4.12) i (11.4.13) vrijedi

$$\begin{aligned} \langle p_i, q_l\rangle &= \langle p_i, Ap_l\rangle = \langle Ap_i, p_l\rangle = \langle q_i, p_l\rangle \\ &= -\lambda_l \langle q_i, H_l \nabla f(x_l)\rangle = -\lambda_l \gamma_{i,l} \langle p_i, \nabla f(x_l)\rangle = 0 \end{aligned}$$

za $i < l$. Time smo pokazali da (11.4.9)–(11.4.11) vrijedi za $l+1$.

Da bismo dokazali da relacija (11.4.12) vrijedi za $l+1$ trebamo pokazati da je $\langle p_i, \nabla f(x_{l+1})\rangle = 0$ za sve $i < l+1$. Slučaj $i = l$ dokazan je s (11.4.14). Za $i < l$ vrijedi

$$\langle p_i, \nabla f(x_{l+1})\rangle = \Big\langle p_i, \nabla f(x_{i+1}) + \sum_{j=i+1}^{l} q_j \Big\rangle$$

jer je $q_j = \nabla f(x_{j+1}) \nabla f(x_j)$. Zbog (11.4.12) i (11.4.9)–(11.4.11) za $l+1$ gornji je izraz jednak 0. Stoga (11.4.12) vrijedi i za $l+1$.

Korištenjem (11.4.8), direktno slijedi da je $H_{l+1} q_l = p_l$. Nadalje, (11.4.9)–(11.4.11) za $l+1$ i (11.4.13) povlače da je $\langle p_l, q_i\rangle = 0$, $\langle H_l q_l, q_i\rangle = \gamma_{i,l} \langle q_l, p_i\rangle = 0$ za $i < l$, tako da iz (11.4.8) za $i < l$ slijedi

$$H_{l+1} q_i = \gamma_l H_l q_i = \gamma_l \gamma_{i,l} p_i = \gamma_{i,l+1} p_i.$$

Stoga (11.4.13) vrijedi za $l+1$. Uočimo da su (11.4.9)–(11.4.13) trivijalno zadovoljene za $l=0$. Sve dok x_l zadovoljava (b)–(d) iz iskaza teorema i $\nabla f(x_l) \neq 0$, možemo generirati x_{l+1} koji, također, zadovoljava (b)–(d). Niz $x_0, x_1, \ldots$ mora biti konačan; (11.4.9)–(11.4.11) može vrijediti samo za $l \leq n$, jer je l vektora $p_0, \ldots, p_{l-1}$ linearno nezavisno a u $\mathbb{R}^n$ najviše n vektora može biti linearno nezavisno. Kada se niz prekine, recimo za $l=m$, $0 \leq m \leq n$, to mora biti zbog

$$\nabla f(x_m) = 0, \quad x_m = -A^{-1}b,$$

tj. vrijedi tvrdnja (a). U slučaju $m=n$, (d) povlači

$$H_n Q = PD$$

za matrice $P = (p_0, \ldots, p_{n-1})$ i $Q = (q_0, \ldots, q_{n-1})$. Zbog $AP = Q$, iz regularnosti matrice P slijedi

$$H_n = PDP^{-1}A^{-1},$$

što dokazuje tvrdnju (e), a time i teorem. ■

11.5. Konvergencija minimizacijskih metoda

U prošlom potpoglavlju smo opisali osnovne minimizacijske metode. Međutim, nismo ništa rekli o njihovoj konvergenciji, odnosno o tome, pod kojim će uvjetima generirani niz točaka konvergirati prema minimumu promatrane funkcije.

Definirali skup smjerova silaska s

$$D(x) = \{s \mid \langle \nabla f(x), s \rangle < 0\}$$

i pokazali da se za svaki vektor $s \in D(x)$ vrijednost funkcije f smanjuje u smjeru s u nekoj okolini točke x. Za promatranje konvergencije metode, promatrat ćemo nešto manji skup od skupa $D(x)$. Neka je $\| \ \|$ standardna Euklidska norma ($\| \ \| = \| \ \|_2$). Za $\gamma \in \mathbb{R}$, $0 < \gamma \leq 1$, promatrajmo skup

$$D(\gamma, x) = \{s \in \mathbb{R}^n \mid \|s\| = 1, \ \langle \nabla f(x), s \rangle \leq -\gamma \|\nabla f(x)\|\}. \tag{11.5.1}$$

Uvjet

$$\langle \nabla f(x), s \rangle \leq -\gamma \|\nabla f(x)\|, \tag{11.5.2}$$

zbog $\|s\| = 1$, možemo zapisati kao

$$\frac{\langle \nabla f(x), -s \rangle}{\|\nabla f(x)\| \, \|s\|} \geq \gamma$$

ako je $\|\nabla f(x)\| \neq 0$. Ako je $\|\nabla f(x)\| = 0$, uvjet (11.5.2) je uvijek zadovoljen. Ako s Θ označimo kut između vektora $\nabla f(x)$ i $-s$, zbog svojstva skalarnog produkta

$$\langle \nabla f(x), -s \rangle = \cos\Theta \|\nabla f(x)\| \, \|s\|,$$

dobivamo da uvjet (11.5.2) glasi

$$\cos\Theta \geq \gamma$$

te promatramo samo one smjerove silaska koji zatvaraju dovoljno mali kut s vektorom $-\nabla f(x)$. Koliko je taj kut malen, definira koeficijent γ. Sljedeća lema pokazuje pod kojim uvjetima postoji skalar λ i (smjer silaska) $s \in \mathbb{R}^n$ za koje je $f(x+\lambda s) < f(x)$.

Lema 11.5.1 *Neka je $f : \mathbb{R}^n \to \mathbb{R}$ funkcija čiji je gradijent $\nabla f(x)$ definiran i neprekidan za sve $x \in V(\bar{x})$ u okolini $V(\bar{x})$ točke $\bar{x}$. Nadalje, pretpostavimo da je $\nabla f(\bar{x}) \neq 0$ i neka je $1 \geq \gamma > 0$. Tada postoji okolina $U(\bar{x}) \subseteq V(\bar{x})$ od $\bar{x}$ i broj $\mu > 0$ takav da je*

$$f(x+\lambda s) \leq f(x) - \frac{\lambda\gamma}{4}\|\nabla f(\bar{x})\|$$

za sve $\lambda \in [0, \mu]$.

Dokaz. Skup

$$U^1(\bar{x}) := \left\{ x \in V(\bar{x}) \mid \|\nabla f(x) - \nabla f(\bar{x})\| \leq \frac{\gamma}{4}\|\nabla f(\bar{x})\| \right\}$$

je neprazan i $\bar{x} \notin \partial U^1(\bar{x})$, jer je $\nabla f(\bar{x}) \neq 0$ i ∇f je neprekidna na $V(\bar{x})$. Slično

$$U^2(\bar{x}) := \left\{ x \in V(\bar{x}) \mid d(\gamma, x) \subseteq D\left(\frac{\gamma}{2}, \bar{x}\right) \right\}$$

je neprazan i okolina od $\bar{x}$ (tj. $x \in U^2(\bar{x})$ i $x \notin \partial U^2(\bar{x})$). Izaberimo $\mu > 0$ takav da vrijedi inkluzija

$$\overline{S_{2\mu}(\bar{x})} = \{x \mid \|x - \bar{x}\| \leq 2\mu\} \subseteq U^1(\bar{x}) \cap U^2(\bar{x}),$$

i neka je

$$U(\bar{x}) := \overline{S_\mu(\bar{x})} = \{x \mid \|x - \bar{x}\| \leq \mu\}.$$

Tada za $x \in U(\bar{x})$, $0 \leq \lambda \leq \mu$, $s \in D(\gamma, x)$, postoji θ, $0 < \theta < 1$ (teorem srednje vrijednosti) takav da je

$$\begin{aligned} f(x) - f(x+\lambda s) &= \lambda\langle \nabla f(x+\theta\lambda s), s\rangle \\ &= \lambda[\langle \nabla f(x+\theta\lambda s) - \nabla f(\bar{x}), s\rangle + \langle \nabla f(\bar{x}), s\rangle]. \end{aligned}$$

Uočimo da je $x+\theta\lambda s \in U^1(\bar{x})$. Naime, zbog $x \in U(\bar{x}) = \overline{S_\mu(\bar{x})}$, $\|s\| = 1$ i $0 < \theta < 1$, vrijedi

$$\|x + \theta\lambda s - \bar{x}\| \leq \|x - \bar{x}\| + \theta\lambda\|s\| \leq \lambda + \lambda = 2\lambda,$$

to jest

$$x + \theta\lambda s \in \overline{S_{2\mu}(\bar{x})} \subseteq U^1(\bar{x}).$$

Stoga je

$$\langle \nabla f(x+\theta\lambda s) - \nabla f(\bar{x}), s\angle \geq -\|\nabla f(x+\theta\lambda s) - \nabla f(\bar{x})\|\,\|s\| \geq -\frac{\gamma}{4}\,\|\nabla f(\bar{x})\|.$$

Nadalje, jer je $s \in D(\gamma, x) \subseteq D(\gamma/2, \bar{x})$ (to vrijedi jer je $x \in U(\bar{x}) \subseteq U^2(\bar{x})$), vrijedi

$$\langle \nabla f(\bar{x}), s\rangle \geq \frac{\gamma}{2}\,\|\nabla f(\bar{x})\|.$$

Sada je

$$f(x) - f(x+\lambda s) \geq \lambda\Big(-\frac{\gamma}{4}\,\|\nabla f(\bar{x})\| + \frac{\gamma}{2}\,\|\nabla f(\bar{x})\|\Big) = \frac{\lambda\gamma}{4}\,\|\nabla f(\bar{x})\|$$

čime smo dokazali lemu. ■

Sada ćemo promatrati sljedeću metodu za minimizaciju diferencijabilne funkcije $f : \mathbb{R}^n \to \mathbb{R}$.

Metoda 1.

(a) Izaberi brojeve γ_i, σ_i, $i = 0, 1, 2, \ldots$ koji zadovoljavaju

$$\sup_i \gamma_i \leq 1, \quad \inf_i \gamma_i > 0, \quad \inf_i \sigma_i > 0,$$

i izaberi početnu točku $x_0 \in \mathbb{R}^n$.

(b) Za $i = 0, 1, 2, \ldots$ izaberi (proizvoljan) $s_i \in D(\gamma_i, x_i)$ i postavi

$$x_{i+1} := x_i + \lambda_i s_i,$$

gdje je $\lambda_i \in [0, \sigma_i\|\nabla f(x_i)\|]$ takav da je

$$f(x_{i+1}) = \min_{0\leq\lambda\leq\sigma_i\|\nabla f(x_i)\|} f(x_i + \lambda s_i).$$

Metoda 1 poopćava velik broj metoda. Jedna od njih je i gradijentna metoda. Naime, kod gradijentne metode je

$$s_i = -\frac{\nabla f(x_i)}{\|\nabla f(x_i)\|}.$$

Budući da je je

$$\langle \nabla f(x_i), s_i\rangle = \Big\langle \nabla f(x_i), -\frac{\nabla f(x_i)}{\|\nabla f(x_i)\|}\Big\rangle = -\|\nabla f(x_i)\|,$$

uvijek je zadovoljen uvjet

$$\langle \nabla f(x_i), -s_i\rangle \geq 1 \cdot \|\nabla f(x_i)\|,$$

tj. možemo uzeti $\gamma_i = 1$. Parametar σ_i kontrolira interval po kojem minimiziramo funkciju

$$g_i(\lambda) := f(x_i + \lambda s_i).$$

Traženje minimuma po neograničenom intervalu ($\lambda \in [0, \infty)$) je često zahtjevno, ponekad i nemoguće. Stoga problem minimizacije pojednostavimo traženjem minimuma na segmentu $[0, \sigma_i \|\nabla f(x_i)\|]$, širinu kojega definira σ_i. Ovaj parametar definiramo proizvoljno, vodeći računa da veći σ_i znači manju (bolju) vrijednost funkcije $f(x_i + \lambda s_i)$ ali i veći interval po kojem tražimo minimum, što znači i dulji postupak (bilo analitički, bilo numerički). Manji σ_i smanjuje interval i ubrzava jednodimenzionalnu minimizaciju u i-tom koraku, ali rezultira većom vrijednošću funkcije f u novoj točki x_{i+1}.

Kako niz $(x_i)_i$ dobivenovom metodom konvergira, opisuje sljedeći teorem.

Teorem 11.5.1 *Neka je $f : \mathbb{R}^n \to \mathbb{R}$ i neka je $x_0 \in \mathbb{R}^n$ izabran da vrijedi*

(a) $K := \{x \mid f(x) \leq f(x_0)\}$ je kompaktan,

(b) f je neprekidno diferencijabilna na nekom otvorenom skupu koji sadrži K.

Tada za svaki niz $(x_i)_i$ definiran Metodom 1 vrijedi

(a) $x_i \in K$ za sve $i = 0, 1, 2, \ldots$ i $(x_i)_i$ ima barem jedno gomilište $\bar{x}$ u K.

(b) Svako gomilište niza $(x_i)_i$ je stacionarna točka od f; $\nabla f(\bar{x}) = 0$.

Dokaz. Iz definicije niza $(x_i)_i$ direktno slijedi da je niz $(f(x_i))_i$ monoton:

$$\cdots \leq f(x_{i+1}) \leq f(x_i) \leq \cdots \leq f(x_1) \leq f(x_0),$$

te posebno vrijedi $f(x_i) \leq f(x_0)$, tj. $x_i \in K$. Budući da je K kompaktan, $(x_i)_i$ ima barem jedno gomilište u K. Time smo dokazali prvu tvrdnju teorema.

Za dokaz druge tvrdnje, pretpostavit ćemo suprotno. Neka je $(x_i)_i$ niz u K i $\bar{x}$ njegovo gomilište te vrijedi

$$\nabla f(\bar{x}) \neq 0. \tag{11.5.3}$$

Bez smanjenja općenitosti možemo pretpostaviti da je $\bar{x} = \lim_i x_i$ (tj. uzimamo konvergentan podniz). Neka je

$$\gamma := \inf_i \gamma_i > 0 \quad \text{i} \quad \sigma := \inf_i \sigma_i > 0.$$

Prema Lemi 11.5.1, postoji okolina $U(\bar{x})$ od $\bar{x}$ i broj $\mu > 0$ koji zadovoljava

$$f(x + \lambda s) \leq f(x) - \lambda \frac{\gamma}{4} \|\nabla f(x)\| \tag{11.5.4}$$

za sve $x \in U(\bar{x})$, $s \in D(\gamma, x)$ i $0 \leq \gamma \leq \mu$.

Kako je $\lim_i x_i = \bar{x}$, neprekidnost od $\nabla f(x)$ zajedno s (11.5.3) povlači egzistenciju k_0 takvog da je za sve $k \geq k_0$

$$x_i \in U(\bar{x}) \quad \text{i} \quad \|\nabla f(x_i)\| \geq \frac{1}{2}\,\|\nabla f(\bar{x})\|.$$

Neka je

$$M := \min\left\{\mu, \frac{1}{2}\,\sigma\|\nabla f(\bar{x})\|\right\} \quad \text{i} \quad \varepsilon := M\,\frac{\gamma}{4}\,\|\nabla f(\bar{x})\| > 0.$$

Iz $\sigma_i \geq \sigma$, slijedi da je $[0, M] \subseteq [0, \sigma_i\|\nabla f(x_i)\|]$ za sve $k \geq k_0$. Prema tome, iz definicije za x_{i+1} slijedi

$$f(x_{i+1}) \leq \min_{0\leq\lambda\leq M} f(x_i + \lambda s_i).$$

Zbog $M \leq \mu$, $x_i \in U(\bar{x})$, $s_i \in D(\gamma_i, x_i) \subseteq D(\gamma, x_i)$, (11.5.4) povlači

$$f(x_{i+1}) \leq f(x_i) - \frac{M\gamma}{4}\,\|\nabla f(\bar{x})\| = f(x_i) - \varepsilon$$

za sve $k \geq k_0$. To povlači da je $\lim_i f(x_i) = -\infty$, što je u kontradikciji s činjenicom da je $(f(x_i))_i$ odozdo ograničen s $f(\bar{x})$ zbog konvergencije niza $(x_i)_i$:

$$f(x_i) \geq f(x_{i+1}) \geq \cdots \geq f(\bar{x}).$$

Dakle, $\bar{x}$ je stacionarna točka funkcije f. ■

Korak (b) u Metodi 1 je minimizacija funkcije

$$g(\lambda) := f(x_i + \lambda s_i)$$

na segmentu $[0, \sigma_i\|\nabla f(x_i)\|]$. Ovaj je korak poznat pod imenom pretraživanje po pravcu (engl. 'line search'). Uočimo da uvjeti Teorema 11.5.1 zahtijevaju pronalaženje egzaktnog minimuma, što je težak (često nemoguć) zadatak. Numerička minimizacija nekom metodom za jednodimenzionalnu minimizaciju (npr. metoda zlatnog reza) može dati dosta točnu aproksimaciju minimuma. Ali i taj postupak može zahtijevati značajnu količinu računanja. Stoga Metoda 1 ima ograničenu primjenu u praksi. Sljedeća varijanta ove metode koristi ideju da se egzaktna minimizacija zamijeni 'neegzaktnim pretraživanjem po pravcu', konačnim postupkom minimizacije.

Metoda 2.

(a) Izaberu se brojevi γ_i, σ_i, $i = 0, 1, 2, \ldots$ koji zadovoljavaju

$$\sup_i \gamma_i \leq 1, \quad \inf_i \gamma_i > 0, \quad \inf_i \sigma_i > 0,$$

i početna točka $x_0 \in \mathbb{R}^n$.

(b) Za $i = 0, 1, 2, \ldots$ izračuna se x_{i+1} iz x_i na sljedeći način:

α Izabere se proizvoljan $s_i \in D(\gamma_i, x_i)$, definira

$$\rho_i := \sigma_i \|\nabla f(x_i)\|, \quad \phi_i(\lambda) := f(x_i + \lambda s_i),$$

i odredi najmanji cijeli broj $j \geq 0$ takav da je

$$\phi_i(\rho_i 2^{-j}) \leq \phi_i(0) - \rho_i 2^{-j} \frac{\gamma_i}{4} \|\nabla f(x_i)\|. \tag{11.5.5}$$

β Odredi se $\bar{i} \in \{0, 1, \ldots, j\}$ takav da je $\phi_i(\rho_i 2^{-\bar{i}})$ minimalan i definira

$$x_{i+1} := x_i + \lambda_i s_i$$

gdje je

$$\lambda_i = \rho_i 2^{-\bar{i}}.$$

Uočimo da j koji zadovoljava (11.5.5) postoji. Ako je x_i stacionarna točka ($\nabla f(x_i) = 0$) tada je $j = 0$. Ako x_i nije stacionarna točka, egzistencija od j slijedi direktno iz Leme 11.5.1 stavljanjem $\bar{x} := x_i$. U svakom slučaju, j (i λ_i) se može odrediti u konačnom broju koraka. I za ovu metodu može se dokazati teorem analogan Teoremu 11.5.1.

Teorem 11.5.2 *Pod pretpostavkom Teorema 11.5.1, svaki niz $(x_i)_i$ dobiven Metodom 2 zadovoljava tvrdnje Teorema 11.5.1.*

11.5.1. Konvergencija modificirane Newtonove metode

U modificiranoj Newtonovoj metodi smjer silaska s_i zadan je s

$$\bar{s}_i = -[\nabla^2 f(x_i)]^{-1} \nabla f(x_i), \quad s_i := \frac{\bar{s}_i}{\|\bar{s}_i\|}.$$

Radi jednostavnosti, označimo

$$H := [\nabla^2 f(x_i)]^{-1} \quad \text{i} \quad g := \nabla f(x_i),$$

pa je

$$s_i = -\frac{Hg}{\|Hg\|}.$$

Pokušajmo odrediti γ_i za koji je

$$\langle \nabla f(x_i), s_i \rangle \geq \gamma_i \|\nabla f(x_i)\|,$$

tj.

$$\langle g, Hg \rangle \geq \gamma_i \|g\| \, \|Hg\|. \tag{11.5.6}$$

Ako je f dva puta neprekidno diferencijabilna, tada je $\nabla^2 f(x_i)$ simetrična matrica, te vrijedi

$$\lambda_{\max}\|x\|^2 \geq \langle \nabla^2 f(x_i)x, x\rangle \geq \lambda_{\min}\|x\|^2, \quad \forall x \in \mathbb{R}^n,$$

gdje su $\lambda_{\max}$ i $\lambda_{\min}$ najveća i najmanja svojstvena vrijednost matrice $\nabla^2 f(x_i)$. Važno je uočiti da za euklidsku normu $\|\ \|$ vrijedi:

$$\|\nabla^2 f(x_i)\| = \max\{|\lambda_{\max}|, |\lambda_{\min}|\}.$$

Ako je matrica $\nabla^2 f(x_i)$ pozitivno definitna, onda je

$$\lambda_{\max} > \lambda_{\min} > 0,$$

te je

$$\|\nabla^2 f(x_i)\| = \lambda_{\max}.$$

Ako je λ svojstvena vrijednost matrice $\nabla^2 f(x_i)$, tada je $1/\lambda$ svojstvena vrijednost matrice $[\nabla^2 f(x_i)]^{-1}$, pa je

$$\|[\nabla^2 f(x_i)]^{-1}\| = \frac{1}{\lambda_{\min}}.$$

Znači, matrica $H = [\nabla^2 f(x_i)]^{-1}$ zadovoljava

$$\frac{1}{\lambda_{\min}}\|x\|^2 \geq \langle Hx, x\rangle \geq \frac{1}{\lambda_{\max}}\|x\|^2, \quad \forall x \in \mathbb{R}^n.$$

Nadalje vrijedi

$$\|Hx\| \leq \|H\|\,\|x\|.$$

Iskoristivši ove nejednakosti, dobivamo

$$\langle g, Hg\rangle \geq \frac{1}{\lambda_{\max}}\|g\|^2$$

i

$$\gamma_i\|g\|\,\|Hg\| \leq \gamma_i \frac{1}{\lambda_{\min}}\|g\|^2.$$

Ukoliko je zadovoljeno

$$\frac{1}{\lambda_{\max}} \geq \gamma_i \frac{1}{\lambda_{\min}}, \tag{11.5.7}$$

tada iz

$$\langle g, Hg\rangle \geq \frac{1}{\lambda_{\max}}\|g\|^2 \geq \gamma_i \frac{1}{\lambda_{\min}}\|g\|^2 \geq \gamma_i\|g\|\,\|Hg\|$$

slijedi da je zadovoljeno (11.5.5).

Uvjet (11.5.7) možemo zapisati u obliku

$$\gamma_i \leq \frac{\lambda_{\min}}{\lambda_{\max}} = \frac{1}{\|\nabla^2 f(x_i)\|\,\|[\nabla^2 f(x_i)]^{-1}\|}.$$

Veličina

$$\text{cond}(A) := \|A\| \, \|A^{-1}\|$$

naziva se broj uvjetovanosti matrice A. Sada uvjet (11.5.7) možemo zapisati u obliku

$$\gamma_i \leq \frac{1}{\text{cond}(\nabla^2 f(x_i))}.$$

Ako je matrica singularna, tada je njen broj uvjetovanosti ∞, pa je $\gamma_i = 0$ te nemamo zadovoljen uvjet konvergencije.

Međutim, ako je $\nabla^2 f$ regularna u okolini minimuma $\bar{x}$ (onda je automatski i pozitivno definitna), tada možemo staviti

$$\gamma_i = \frac{1}{\text{cond}(\nabla^2 f(x_i))}$$

te je zbog regularnosti

$$\inf_i \gamma_i = \frac{1}{\sup_i \text{cond}(\nabla^2 f(x_i))} > 0.$$

Ovo pokazuje da je modificirana Newtonova metoda konvergentna ako je funkcija f konveksna ($\nabla^2 f$ pozitivno definitna), što je mnogo jači uvjet od npr. uvjeta konvergencije gradijentne metode. Međutim, ova metoda brže konvergira ukoliko su ispunjeni uvjeti konvergencije.

Literatura

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *LAPACK Users' Guide*, Third edition, SIAM, Philadelpia, 1999.

[2] K. E. Atkinson, *An Introduction to Numerical Analysis (2^{nd} edition)*, John Wiley & Sons, New York, 1989.

[3] W. Gautschi, *Numerical Analysis (An Introduction)*, Birkhäuser, Boston, 1997.

[4] D. Goldberg, *What every computer scientist should know about floating-point arithmetic*, ACM Computing Surveys, vol. 23, no. 1, March 1991.

[5] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[6] M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM, Philadelpia, 2001.

[7] A. Ralston, P. Rabinowitz, *A First Course in Numerical Analysis*, McGraw–Hill, Singapore, 1978.

[8] G. W. Stewart, J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.

[9] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963. (Also published by Prentice–Hall, Englewood Cliffs, NJ, USA. Reprinted by Dover, New-York, 1994, ISBN 0–486–67999–5.)