Znanstveno računanje 2

2. dio vježbi
Modeli s primjenama numeričke linearne algebre

Nela Bosner

Modeli s primjenama numeričke linearne algebre

Zadatak

U MATLAB-u napišite M-file funkciju `sor()` *koja implementira SOR metodu za rješavanje linearnih sustava jednadžbi. Funkcija neka ima ulazne parametre*

- *matricu sustava A i desnu stranu sustava b*
- *početnu iteraciju x_0*
- *toleranciju na relativnu normu reziduala*
- *parametar ω*

Kriterij zaustavljanja je $\|b - Ax_k\|_2 / \|b\|_2 \leq tol$, a za računanje norme koristite MATLAB-ovu funkciju `norm()`.

Zadatak (nastavak)

Funkcija neka vraća

- *aproksimaciju rješenja* x_k
- *broj iteracija* k *potrebnih za dostizanje aproksimativnog rješenja tražene točnosti*
- *vektor duljine* $k+1$ *sa relativnim normama reziduala za svaku iteraciju* $i = 0, \ldots, k$

Zadatak

Napišite M-file funkciju `sor_konvergencija()` *koja za zadanu matricu A crta graf spektralnih radijusa matrice iteracija za SOR metodu* $T_{SOR,\omega}$.

- *Matrica A je jedini ulazni parametar.*
- *Funkcija neka generira* ω *iz ekvidistantne mreže na segmentu* $[0, 2]$ *s korakom* 0.01*, i za svaki* ω *računa* $\rho(T_{SOR,\omega})$.
- *Sve vrijednosti* ω *i odgovarajuće* $\rho(T_{SOR,\omega})$ *spremite u vektore* `omega` *i* `ro`*, koji će se koristiti za crtanje grafa s* ω *na x osi i* $\rho(T_{SOR,\omega})$ *na y osi.*

Graf će služiti za određivanje optimalnog ω.

Zadatak (nastavak)

Koristite MATLAB-ove funkcije funkcije

- `diag()`, `triu()` *i* `tril()` *za generiranje matrice iteracija* $T_{SOR,\omega}$
- `max(abs(eig(T)))` *za računanje spektralnog radijusa*
- `plot()` *za crtanje grafa*
- `axis()` *za određivanje granica na x i y osima grafa*
- `xlabel()` *i* `ylabel()` *za označavanje x i y osi*

Rizik i očekivani povrat portfelja

- Portfelj koji se sastoji od n različitih vrijednosnica može se opisati pomoću njihovih težina

$$\omega_i = \frac{x_i S_i(0)}{V(0)}, \qquad i = 1, \ldots, n,$$

gdje je x_i broj dionica tipa i u portfelju, $S_i(0)$ je početna cijena vrijednosnice i, a $V(0)$ je količina koja je početno investirana u portfelj.

- Definirajmo

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n.$$

- Iz definicije je vidljivo da je $\sum_{i=1}^{n} \omega_i = 1$, odnosno

$$\mathbf{e}^T \boldsymbol{\omega} = 1.$$

- Pretpostavimo da povrat i-te vrijednosnice R_i ima očekivanje $\mu_i = E(R_i)$, i definirajmo

$$\mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}.$$

- Nadalje, kovarijancu između dva povrata označimo sa $c_{ij} = Cov(R_i, R_j)$ i definirajmo matricu kovarijance

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}.$$

- Dobro je poznato da je matrica kovarijance simetrična pozitivno definitna matrica, pa je prema tome regularna i inverz $\mathbf{C}^{-1}$ postoji.
- Očekivani povrat $\mu_P = E(R_P)$ i varijanca $\sigma_P^2 = Var(R_P)$ portfelja sa težinama $\boldsymbol{\omega}$ dani su sa

$$\begin{aligned} \mu_P &= \boldsymbol{\mu}^T \boldsymbol{\omega} \\ \sigma_P^2 &= \boldsymbol{\omega}^T \mathbf{C} \boldsymbol{\omega} \end{aligned}$$

- Portfelj sa najmanjom varijancom ima težine

$$\boldsymbol{\omega}_{min} = \frac{\mathbf{C}^{-1}\mathbf{e}}{\mathbf{e}^T\mathbf{C}^{-1}\mathbf{e}}.$$

- Portfelj sa najmanjom varijancom i sa očekivanim povratom μ_P ima težine

$$\boldsymbol{\omega}_{\mu_p} = \frac{(\boldsymbol{\mu}^T\mathbf{C}^{-1}\boldsymbol{\mu} - \mu_P \cdot \mathbf{e}^T\mathbf{C}^{-1}\boldsymbol{\mu})\mathbf{C}^{-1}\mathbf{e} + (\mu_p \cdot \mathbf{e}^T\mathbf{C}^{-1}\mathbf{e} - \mathbf{e}^T\mathbf{C}^{-1}\boldsymbol{\mu})\mathbf{C}^{-1}\boldsymbol{\mu}}{\mathbf{e}^T\mathbf{C}^{-1}\mathbf{e} \cdot \boldsymbol{\mu}^T\mathbf{C}^{-1}\boldsymbol{\mu} - (\mathbf{e}^T\mathbf{C}^{-1}\boldsymbol{\mu})^2}.$$

Zadatak

- *Nađite efikasni način za računanje ω_{min} i ω_{μ_p}.*
- *Izračunajte ω_{min} i ω_{μ_p} za konkretan primjer, čiji su očekivani povrati i matrica kovarijance spremljeni u datoteci*

  `model_portfelj_Cm.mat`

  na adresi

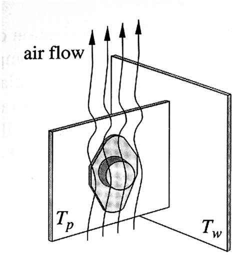
  http://www.math.hr/~nela/zr2.html

  a $\mu_p = 0.05$.
- *Koristite SOR metodu s optimalnim parametrom ω i MATLAB-ovu funkciju* `pcg()` *sa svim mogućim izlaznim varijablama.*

Zadatak (nastavak)

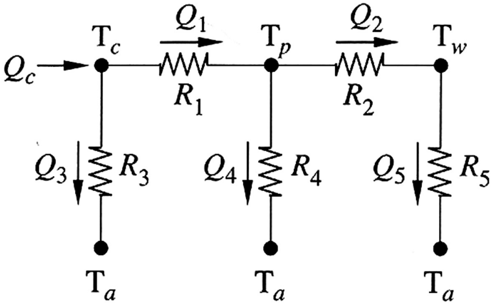
- *Kriterij zaustavljanja za obje metode neka je* $\|Ax_k - b\|_2/\|b\|_2 < 10^{-8}$.
- *Nacrtajte graf spektralnih radijusa matrice iteracija za SOR metodu.*
- *Nacrtajte grafove relativnih normi reziduala za iteracije obiju metoda, i to tako da za svaki sustav na istom grafu budu prikazani reziduali za obje metode.*
- *Relativne norme reziduala nacrtajte u logaritamskoj skali pomoću MATLAB-ove funkcije* `semilogy()`.

Disipacija topline elektroničke komponente



- Promatramo elektroničku komponentu, koja može biti tranzistor ili bilo koji drugi uređaj koji gubi značajnu količinu energije u obliku topline.
- U primjeru na slici komponenta je pričvršćena na ploču koja služi za širenje topline.
- Ta ploče je dalje pričvršćena na drugu plohu, koja može biti kućište cijelog sklopa.

- Temperature su definirane na sljedeći način:
  - T_c — temperatura elektroničke komponente
  - T_p — temperatura ploče
  - T_w — temperatura zida
  - T_a — temperatura okolnog zraka
- Elektronička komponenta troši Q_c (W) električne snage na zagrijavanje.
- Budući da je $T_c > T_a$, zrak u blizini komponente struji prema gore. To strujanje zraka pomaže kod hlađenja komponente, ploče i zida.
- Toplina komponente
  1. ili se prenosi na zrak,
  2. ili se provodi u ploču.
- Toplina dovedena do ploče
  1. ili se prenosi na zrak,
  2. ili se provodi u zid.
- Zid prenosi preostalu toplinu na zrak.



Slika: Model termalne mreže za toplinski tok od elektroničke komponente prema okolini.

- Čvorovi mreže su mjesta na kojima je definirana temperatura.
- Strelice označavaju pretpostavljeni smjer toka topline između čvorova.
- Otpornici u mreži predstavljaju termalni otpor toku topline.
  - Tok topline s visokim termalnim otporom zahtijeva veću temperaturnu razliku za prijenos određene količine topline nego tok sa manjim termalnim otporom.
- Primjenom zakona o sačuvanju energije na elektroničku komponentu i njenu potpornu konstrukciju, dobivamo sljedeće jednadžbe.

$$
\begin{aligned}
Q_1 &= \frac{1}{R_1}(T_c - T_p) & Q_2 &= \frac{1}{R_2}(T_p - T_w) \\
Q_3 &= \frac{1}{R_3}(T_c - T_a) & Q_4 &= \frac{1}{R_4}(T_p - T_a) \\
Q_5 &= \frac{1}{R_5}(T_w - T_a) & Q_c &= Q_1 + Q_3 \\
Q_1 &= Q_2 + Q_4 & Q_2 &= Q_5
\end{aligned}
$$

- Snaga disipacije elektroničke komponente Q_c i temperatura okolnog zraka T_a su poznati.
- Termalni otpori mogu se izračunati iz poznatih svojstava materijala, fizičkih dimenzija i empiričkih korelacija toka topline u različitim fizičkim konfiguracijama.

- Dakle, Q_c, T_a i R_i su poznati.
- Nepoznanice su

$$x = [\ Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad T_c \quad T_p \quad T_w\ ]^T.$$

Zadatak

1. *Definirajte matricu A i vektor b tako da sustav $Ax = b$ predstavlja matrični oblik prethodnih jednadžbi za disipaciju topline.*
2. *Napišite M-file funkciju*

```
function y=mdAx(x)
```

koja implementira množenje matrice A s vektorom x. Budući da matrica A ima puno nula, množite samo s netrivijalnim elementima matrice.

Zadatak (nastavak)

Poznate vrijednosti zadane su u tablici

R_1	R_2	R_3	R_4	R_5	T_a	Q_c
2	0.5	35	0.7	1	20	10

3. *Sustav $Ax = b$ riješite MATLAB-ovom funkcijom `gmres()` sa svim mogućim izlaznim varijablama. Umjesto ulaznog parametra za matricu A stavite pokazivač na funkciju (function handle) `@mdAx`, neka je $x_0 = [\ 1 \quad 1 \quad \cdots \quad 1\ ]^T$, a kriterij zaustavljanja neka je $\|Ax_k - b\|_2 / \|b\|_2 < 10^{-8}$.*
4. *Nacrtajte graf relativnih normi reziduala za iteracije GMRES metode.*

Orbita asteroida

- Za ovaj problem moramo se sjetiti jednog matematičkog teorema i jednog fizikalnog zakona:
  1. Opća jednadžba ravninske konike (elipsa, parabola, hiperbola) je dana sa
  $$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0,$$
  gdje su A, B, C, D, E i F konstante.
  2. **Keplerov prvi zakon**: orbita asteroida oko sunca mora biti elipsa.
- Astronom koji želi odrediti orbitu asteroida oko sunca postavlja koordinatni sustav u ravnini orbite sa ishodištem u suncu (fokus).
- Tada mjeri 5 različitih položaja asteroida u tom sustavu, čime dobivamo 5 različitih točaka na orbiti.
- 5 točaka je dovoljno da odredimo jednadžbu elipse.

- Obzirom da imamo 6 nepoznanica A, B, C, D, E i F, i da se jednadžba krivulje ne mijenja ako je pomnožimo skalarom, možemo jedan od parametara svesti na 1.
- U našem slučaju uzmimo da je $A = 1$ (jer se radi o elipsi) i da jednadžba ima oblik

$$x^2 + a_1 xy + a_2 y^2 + a_3 x + a_4 y + a_5 = 0.$$

Zadatak

- *Postavite sustav linearnih jednadžbi za nepoznanice a_i $i = 1, \ldots, 5$.*
- *Izmjereni položaji nalaze se u datoteci*

  `model_orbita_polozaji.mat`

  na adresi

  http://www.math.hr/~nela/zr2.html

Zadatak (nastavak)

- *Izaberite najpogodniju metodu za rješavanje dobivenog sustava.*
- *Definirajte anonimnu funkciju*

```
f=@(x,y) x.^2+a1*x.*y+a2*y.^2+a3*x+a4*y+a5;
```

nakon što izračunate parametre `ai`*.*

- *Napišite sljedeće:*

```
x=-5:0.1:5;
y=x;
[X,Y]=meshgrid(x,y);
Z=f(X,Y);
contour(x,y,Z,[0 0]);
grid on
```

- *Isprobajte i funkciju* `surf()`*.*

Numeričko rješavanje Poissonove jednadžbe

Numerički ćemo riješiti Poissonovu parcijalnu diferencijalnu jednadžbu (rubni problem), koja je specijalni oblik difuzijske jednadžbe (npr. distribucija temperature u objektu).

- Imamo sljedeći problem:

$$\begin{aligned} -\Delta u(x,y) &= f(x,y) && \text{na } \Omega \\ u(x,y) &= 0 && \text{na } \partial\Omega \end{aligned}$$

  gdje je Ω jedinični kvadrat $\langle 0,1\rangle \times \langle 0,1\rangle$.
- Δ je Laplaceov operator

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

- Mi ćemo rješavati stacionarnu difuzijsku jednadžbu s uniformnim toplinskim konduktivitetom materijala, s točkastim vanjskim izvorom topline u središtu kvadrata jačine 10000, i konstantnom temperaturom na rubu.
- Ovaj problem je ekvivalentan rješavanju Poissonove jednadžbe s funkcijom

$$f(x,y) = \begin{cases} 0 & \text{za } (x,y) \in \Omega \setminus \{(0.5, 0.5)\} \\ 10000 & \text{za } (x,y) = (0.5, 0.5) \end{cases}$$

- Za diskretizaciju, sada ćemo uzeti ekvidistantnu dvodimenzionalnu mrežu

$$h = 0.05, \quad x_i = ih, \quad y_j = jh, \quad i,j = 0,1,\ldots,20,$$

$$u_{i,j} = u(x_i, y_j), \quad f_{i,j} = f(x_i, y_j).$$

Zadatak

- *Matricu sustava izračunajte pomoću MATLAB-ove funkcije* `gallery('poisson',19);`, *a vektor desne strane nađite sami.*
- *Sustav riješite metodom konjugiranih gradijenata bez i sa prekondicioniranjem koristeći MATLAB-ovu funkciju* `pcg()`.
- *Za prekondicioniranje koristite nekompletnu faktorizaciju Choleskog, koju ćete izračunati pomoću MATLAB-ove funkcije* `cholinc(A,'0');`.
- *Neka je* $u^{(0)} = [0\ 0\ 0\ \cdots\ 0]^T$.
- *Kriterij zaustavljanja za oba slučaja neka je* $\|b - Au^{(k)}\|_2 / \|b\|_2 \leq 10^{-8}$.

Zadatak (nastavak)

- *Usporedite broj uvjetovanosti matrice A sa matricom prekondicioniranog sustava, broj iteracija potrebnih za dostizanje iste točnosti i grafove relativnih normi reziduala u logaritamskoj skali za oba slučaja.*
- *Za crtanje grafa u logaritamskoj skali koristite MATLAB-ovu funkciju* `semilogy()`.
- *Jednu od izračunatih aproksimacija rješenja* $u^{(k)}$ *prebacite u kvadratnu matricu* $U = [u_{i,j}^{(k)}]$, *sa* $u_{i,j}^{(k)} \approx u(x_i, y_j)$, $i, j = 1, \ldots, 19$, *i nacrtajte plohu rješenja pomoću MATLAB-ove funkcije* `surf()`.

Kreditni razred

- Pretpostavimo da korporacije mogu biti u jednom od n mogućih kreditnih razreda ("credit rating"), i da one mogu preći iz jednog razreda u bilo koji drugi u diskretnim jedinicam vremena, recimo svake godine.
- Neka je a_{ij} vjerojatnost da korporacija prijeđe u razred i sljedeće godine, ako se trenutno nalazi u razredu j.
- Pretpostavimo da je ovaj sustav zapravo **Markovljev lanac**, tj. da vjerojatnosti prelaska ovise samo o trenutnom razredu, a ne o prošlim razredima. (To je samo aproksimacija realnih sustava.)

Svojstva matrice $A = [a_{ij}]$:

- $0 \leq a_{ij} \leq 1$, jer se radi o vjerojatnostima.
- $\sum_i a_{ij} = 1$, za svako j, budući da sustav uvijek mora preći u neki novi razred.
- Kvadratna matrica $A = [a_{ij}]$ ima nenegativne elemente, i suma elemenata svakog stupca je 1.

- Pretpostavimo da imamo velik skup korporacija, i neka u_j predstavlja udio u tom skupu onih korporacija, koje su u razredu j u početnom trenutku, uz svojstva $0 \leq u_j \leq 1$ i $\sum_j u_j = 1$. Vektor $u = [u_j]$ tada nazivamo vektorom gustoće.
- Ako je skup dovoljno velik, i ako se prelazak iz razreda u razred svake korporacije odvija neovisno o drugima, tada se udio korporacija u skupu svih korporacija koje će se nakon jedne godine nalaziti u razredu i, označen sa v_i, dobiva kao

$$v_i = \sum_j a_{ij} u_j, \qquad \text{ili} \qquad v = Au.$$

- Primijetimo da je $v = [v_i]$ isto vektor gustoće

$$\sum_i v_i = \sum_i \sum_j a_{ij} u_j = \sum_j \left(\sum_i a_{ij} \right) u_j = \sum_j u_j = 1.$$

- Općenito, ako sa $u^{(k)}$ označimo vektor gustoće nakon k koraka, tada

$$u^{(k)} = Au^{(k-1)} = A^k u^{(0)}.$$

- Prema tome dugoročno ponašanje gustoće ovisi o svojstvima visokih potencija matrice A.
- Prema gornjim pretpostavkama, moguće je procijeniti vjerojatnosti prelaska na osnovu povijesnih podataka.
- U sljedećoj tablici nalaze se vjerojatnosti prelaska izraženi u postocima, za jednu godinu, objavljeni u *Credit Metrics* za 2001. godinu.

Konačni	Početni razred							
razred	AAA	AA	A	BBB	BB	B	CCC	D
AAA	90.81	0.70	0.09	0.02	0.03	0	0.22	0
AA	8.33	90.65	2.27	0.33	0.14	0.11	0	0
A	0.68	7.79	91.05	5.95	0.67	0.24	0.22	0
BBB	0.06	0.64	5.52	86.93	7.73	0.43	1.30	0
BB	0.12	0.06	0.74	5.30	80.53	6.48	2.38	0
B	0	0.14	0.26	1.17	8.84	83.46	11.24	0
CCC	0	0.02	0.01	0.12	1.00	4.07	64.86	0
D	0	0	0.06	0.18	1.06	5.20	19.79	100

- Sada se postavlja pitanje što se događa kad $k \to \infty$?
- Da li se sustav smiruje u ravnotežnom stanju?
- Ako postoji ravnotežno stanje $u^{(\infty)} = \bar{u}$, tada mora vrijediti
$$A\bar{u} = \bar{u},$$
tako da se ono ne mijenja u nadolazećim godinama.
- Dakle, $\bar{u}$ mora biti svojstveni vektor matrice A koji pripada svojstvenoj vrijednosti jednakoj 1.
- Ako pogledamo tablicu, također je jasno da je jedan takav svojstveni vektor jednak $[0, \ldots, 0, 1]^T$, tj. ako su svi u razredu D tada svi i ostaju u tom razredu.
- To nužno ne mora značiti, da svi teže ka razredu D.

- Pretpostavimo da A ima n linearno nezavisnih svojstvenih vektora $v_1, \ldots, v_n$ i n svojstvenih vrijednosti $\lambda_1, \ldots, \lambda_n$, i pretpostavimo da je v_1 svojstveni vektor koji pripada svojstvenoj vrijednosti $\lambda_1 = 1$.
- Tada $u^{(k)}$ možemo raspisati po komponentama u smjerovima $v_1, \ldots, v_n$ kao

$$u^{(k)} = \nu_1^{(k)} v_1 + \cdots + \nu_n^{(k)} v_n.$$

- Imamo

$$u^{(k+1)} = Au^{(k)} = \sum_{j=1}^{n} \nu_j^{(k)} Av_j = \sum_{j=1}^{n} \lambda_j \nu_j^{(k)} v_j.$$

- Prema tome dobiva se da je $\nu_j^{(k+1)} = \lambda_j \nu_j^{(k)}$, odnosno

$$\nu_j^{(k)} = \lambda_j^k \nu_j^{(0)}.$$

- Komponenta vektora u smjeru j-tog svojstvenog vektora ili raste ili trne eksponencijalno kad $k \to \infty$, ovisno o tome da li je odgovarajuća svojstvena vrijednost veća ili manja od 1 po apsolutnoj vrijednosti.
- Jasno je da niti jedna svojstvena vrijednost od A ne može biti veća od 1 po apsolutnoj vrijednosti,
  - jer da to nije tako, apsolutna vrijednost od u bi rasla eksponencijalno,
  - što je u suprotnosti sa činjenicom da je suma svih komponenti od u jednaka 1.
- Mi znamo da postoji najmanje jedna svojstvena vrijednost jednaka 1.
- Prema tome, ako su sve ostale svojstvene vrijednosti po apsolutnoj vrijednosti manje od 1, tada će njihove komponente utrnuti, i dugoročno gledano razred kojem će svi težiti je razred D.

Zadatak

Nacrtajte animirani graf za iteracije metode potencije primijenjene na matricu prelaska A.

- *Matrica A spremljena je u datoteci*

  `model_kredit_A.mat`

  na adresi

  http://www.math.hr/~nela/zr2.html
- *Vaša funkcija* `kredit_raz()` *treba imati ulazni parametar n koji označava broj iteracija koje će se izvršiti.*
- *Kao početni vektor uzmite* $u^{(0)} = [\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}]^T$.
- *Za svaku iteraciju nacrtajte vektor* $u^{(k)}$*, tako da se na x osi nalaze indeksi 8 komponenti, a na y osi su same komponente* $u_i^{(k)}$.

Zadatak (nastavak)

- *x os neka bude ograničena na raspon $[0.5, 8.5]$, a y os na $[0, 1]$ pomoću naredbe* `axis()`.
- *x os označite sa 'Stanje', a y os sa 'Gustoća'.*
- *Stavite naslov 'Iteracija br. k' na vaš graf pomoću MATLAB-ove funkcije* `title()`*, pri čemu je k točan broj iteracije (za to koristite* `sprintf()` *kao u C-u).*
- *Stavite svoje oznake na x os, i to*
  - *postavite crtice na vrijednosti 1, 2, 3, 4, 5, 6, 7, 8, pomoću MATLAB-ove funkcije*

    ```
    set(gca,'XTick',1:8);
    ```
  - *postavite natpise ispod crtica pomoću MATLAB-ove funkcije*

    ```
    set(gca,'XTickLabel', { 'AAA', 'AA', 'A',
        'BBB', 'BB', 'B', 'CCC', 'D' } );
    ```
  - `gca` *je oznaka za tekuće osi na koje crtamo.*

Zadatak (nastavak)

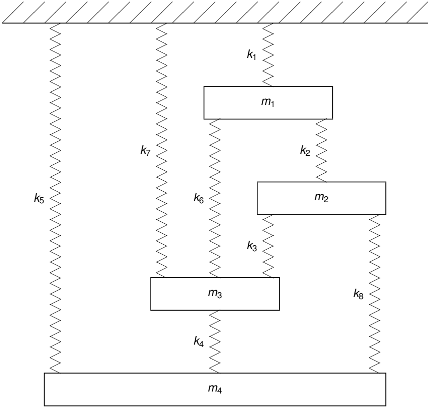
- *Nacrtajte mrežu na grafu.*
- *Svaku iteraciju zadržite 0.2 sekunde da bi animacija bila glatka pomoću MATLAB-ove funkcije* `pause(0.2)`.
- *Izračunajte svojstvene vrijednosti i svojstvene vektore matrice A pomoću* `eig()` *i provjerite dobivene rezultate.*

Napomena

- *Vidimo da je, prema očekivanom 1 najveća svojstvena vrijednost.*
- *Prva sljedeća svojstvena vrijednost je oko 0.988, što je vrlo blizu 1, i koja ukazuje da će konvergencija prema ravnotežnom stanju biti vrlo spora.*
- *Njen svojstveni vektor, osim zadnje komponente, ima najveće komponente u 3. i 4. koordinati.*
- *Zbog toga 3. i 4. koordinate od u najsporije padaju.*

Sistem masa s elastičnim oprugama

Promatramo fizikalni sistem koji se sastoji od tijela različitih masa, povezanih elastičnim oprugama.



Problem je pronaći slobodne oscilacije ovog sistema.

- U ovom konkretnom primjeru imamo četiri tijela masa m_i $i = 1, 2, 3, 4$, i osam opruga krutosti k_l $l = 1, \ldots, 8$.
- Definirat ćemo sljedeće matrice:

$$M = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix},$$

$$K = \begin{bmatrix} k_1 + k_2 + k_6 & -k_2 & -k_6 & 0 \\ -k_2 & k_2 + k_3 + k_8 & -k_3 & -k_8 \\ -k_6 & -k_3 & k_3 + k_4 + k_6 + k_7 & -k_4 \\ 0 & -k_8 & -k_4 & k_4 + k_5 + k_8 \end{bmatrix},$$

pri čemu je u matrici K prikazana interakcija među masama.

- Znamo da se ovaj problem svodi na traženje svojstvenih vrijednosti λ_i i svojstvenih vektora u_i matrice $A = M^{-\frac{1}{2}} K M^{-\frac{1}{2}}$, i da su rješenja tada oblika

$$x_i = M^{-\frac{1}{2}} u_i e^{i\sqrt{\lambda_i} t}, \qquad i = 1, 2, 3, 4.$$

- Frekvencije traženih oscilacija tada su dane sa $\phi_i = \sqrt{\lambda_i}$.
- Neka su vrijednosti masa i krutosti opruga dane u sljedećim tablicama.

i	1	2	3	4
$\mathbf{m_i}$	2	5	3	6

i	1	2	3	4	5	6	7	8
$\mathbf{k_i}$	10	9	8	7	6	5	5	5

Zadatak

- *Pretpostavimo da tražimo rješenje čija je frekvencija slobodne oscilacije $\phi \approx 2$.*
- *Za to ćemo koristiti inverzne iteracije.*
- *Neka je $u^{(k)}$ aproksimacija traženog svojstvenog vektora u k-toj iteraciji, tada kriterij zaustavljanja glasi*

$$\|Au^{(k)} - \varrho(u^{(k)})u^{(k)}\|_2 < 10^{-8}, \quad \varrho(u^{(k)}) = (u^{(k)})^T Au^{(k)}.$$

- *Za računanje rješenja sustava $(A - 4I)x = b$ koristite MATLAB-ovu funkciju `minres()`, sa parametrima `tol=1e-8` i `maxit=4`.*
- *MINRES metoda je itararativna metoda iz Krylovljevih potprostora za rješavanje sustava linearnih jednadžbi sa simetričnom matricom koja u svakoj iteraciji minimizira normu reziduala.*

Algoritam (MINRES)

x_0 *zadan;*

$d_0 = r_0 = b - Ax_0;$

while (!*kriterij_zaustavljanja*){

$$\alpha_k = \frac{r_k^T A d_k}{d_k^T A^2 d_k};$$

$$x_{k+1} = x_k + \alpha_k d_k;$$

$$r_{k+1} = r_k - \alpha_k A d_k;$$

$$\beta_{k+1} = \frac{r_{k+1}^T A^2 d_k}{d_k^T A^2 d_k};$$

$$d_{k+1} = r_{k+1} - \beta_{k+1} d_k;$$

$$k = k + 1;$$

}

Zadatak (nastavak)

- *Za svaku iteraciju nacrtajte vektor* $u^{(k)}$*, tako da se na x osi nalaze indeksi 4 komponenti, a na y osi su same komponente* $u_i^{(k)}$.
- x *os neka bude ograničena na raspon* $[0.5, 4.5]$*, a y os na* $[-1, 1]$.
- x *os označite sa 'Indeks komponente', a y os sa 'Komponenta'.*
- *Stavite naslov 'Aproksimacija svojstvenog vektora: iteracija k' na vaš graf, pri čemu je k točan broj iteracije.*
- *Nacrtajte mrežu na grafu.*
- *Svaku iteraciju zadržite 0.5 sekundi.*
- *Na kraju nacrtajte graf normi reziduala* $\|Au^{(k)} - \varrho(u^{(k)})u^{(k)}\|_2$ *za sve iteracije u logaritamskoj skali, i pravilno označite osi.*

Problem totalnih najmanjih kvadrata

- Za matricu $A \in \mathbb{R}^{m \times n}$ i vektor $b \in \mathbb{R}^m$ problem najmanjih kvadrata

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$$

možemo napisati i na sljedeći način

$$\min_{\mathrm{Im}(b+r) \subseteq \mathrm{Im}(A)} \|r\|_2, \quad \text{gdje je } r \in \mathbb{R}^m.$$

- Budući da r mora biti takav da je $\mathrm{Im}(b + r) \subseteq \mathrm{Im}(A)$, tada mora postojati $x \in \mathbb{R}^n$ takav da je $b + r = Ax$.
- Dakle, za rješenje problema je $r = Ax - b$ s minimalnom normom što smo i tvrdili.
- Na sličan način definirat ćemo problem totalnih najmanjih kvadrata.

- Neka su dani $A \in \mathbb{R}^{m\times n}$ i $B \in \mathbb{R}^{m\times k}$, i pretpostavimo da želimo riješiti sljedeći **problem totalnih najmanjih kvadrata (TLS)**

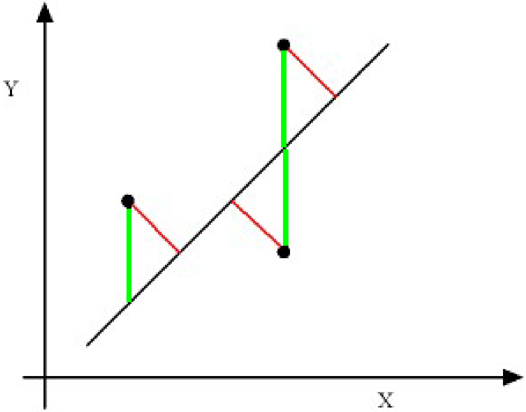
$$\min_{\mathrm{Im}(B+R)\subseteq \mathrm{Im}(A+E)} \|D \cdot [\ E \quad R\ ] \cdot T\|_F,$$

gdje su $E \in \mathbb{R}^{m\times n}$, $R \in \mathbb{R}^{m\times k}$, a matrice $D = \mathrm{diag}(d_1, \ldots, d_m)$ i $T = \mathrm{diag}(t_1, \ldots, t_{n+k})$ su regularne težinske matrice.

- Ako $[E_0\ R_0]$ rješava gornji problem, tada se bilo koji $X \in \mathbb{R}^{n\times k}$ za koji vrijedi

$$(A + E_0)X = B + R_0$$

naziva **TLS rješenje**.



Slika: Pravac kao rješenje najmanjih kvadrata i totalnih najmanjih kvadrtata.

Teorem

Neka su A, B, D i T kao u prethodnoj definiciji problema, za $m \geq n + k$. Neka

$$C = D\begin{bmatrix} A & B \end{bmatrix} T = \begin{array}{c} \begin{bmatrix} C_1 & C_2 \end{bmatrix} \\ \begin{array}{cc} n & k \end{array} \end{array}$$

ima SVD dan sa $U^T CV = \mathrm{diag}(\sigma_1, \ldots, \sigma_{n+k}) = \Sigma$ *gdje su* U, V *i* Σ *particionirani na sljedeći način:*

$$U = \begin{array}{c} \begin{bmatrix} U_1 & U_2 \end{bmatrix} \\ \begin{array}{cc} n & k \end{array} \end{array}, \qquad V = \begin{array}{cc} \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} & \begin{array}{c} n \\ k \end{array} \\ \begin{array}{cc} n & k \end{array} & \end{array},$$

$$\Sigma = \begin{array}{cc} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} & \begin{array}{c} n \\ k \end{array} \\ \begin{array}{cc} n & k \end{array} & \end{array}.$$

Teorem (nastavak)

- *Ako je $\sigma_n(C_1) > \sigma_{n+1}$, tada matrica $[E_0 \ R_0]$ definirana sa*
$$D[\ E_0 \quad R_0\ ]T = -U_2\Sigma_2[\ V_{12}^T \quad V_{22}^T\ ]$$
rješava TLS problem.
- *Ako su $T_1 = \mathrm{diag}(t_1, \ldots, t_n)$ i $T_2 = \mathrm{diag}(t_{n+1}, \ldots, t_{n+k})$, tada matrica*
$$X = -T_1 V_{12} V_{22}^{-1} T_2^{-1}$$
postoji, i ona je jedinstveno rješenje jednadžbe $(A + E_0)X = B + R_0$.
- *Ako je $\sigma_n(C_1) = \sigma_{n+1}$, tada TLS problem još uvijek može imati rješenje, ali ono ne mora biti jedinstveno. U tom slučaju traži se rješenje s minimalnom normom.*

Problem dekonvolucije

- TLS se koristi kod identifikacije sustava koji ovise o vremenu i kod procjene njihovih parametara.
- Postoje mnogi različiti modeli koji se koriste za opisivanje ponašanja takvih sustava.
- Ako se proces može izmodelirati kao linearan, vremenski invarijantan, uzročan, konačnodimenzionalan sustav sa početnim stanjem nula, tada se može koristiti sljedeći model:

$$y(t) = \sum_{k=0}^{n} h(k)u(t-k).$$

- $h(k)$ je reakcija sustava na impuls u vremenu k.
- U nekim slučajevima $h(k)$ može se procijeniti iz promatranja ulaznih vrijednosti $u(t)$ i izlaznih vrijednosti $y(t)$ za sustav u nekom vremenskom intervalu za $t = -n, \ldots, m$.

- To je takozvani *problem dekonvolucije*, i svodi se na rješavanje sustava linearnih jednadžbi $Y = UH$ za $t = 0, \dots, m$:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(m) \end{bmatrix} = \begin{bmatrix} u(0) & u(-1) & \cdots & u(-n) \\ u(1) & u(0) & \cdots & u(1-n) \\ \vdots & \vdots & & \vdots \\ u(m) & u(m-1) & \cdots & u(m-n) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(n) \end{bmatrix}$$

- Ako uzmemo više uzoraka ulaznih i izlaznih vrijednosti, tako da je $m > n$, postiže se bolja točnost.
- Pretpostavimo sada da su sve promatrane ulazne i izlazne vrijednosti perturbirane nezavisnim, vremensko invarijantnim bijelim šumom sa očekivanjem 0 i varijancom 1, tada se $h(k)$ računaju pomoću TLS-a.
- Primjena rješavanja problema dekonvolucije pomoću TLS-a koristi se u medicini kod renografije.

Zadatak

- *Riješite problem za konkretne podatke spremljene u datoteci*

  `model_dekonvolucija_uy.mat`

  na adresi

  `http://www.math.hr/~nela/zr2.html`

- *Radi se o ulaznim vrijednostima* $up(i)$, $i = 1, \ldots, m + n + 1$ *i izlaznim vrijednostima* $yp(j)$, $j = 1, \ldots, m + 1$ *za* $n = 18$ *i* $m = 102$.
- *Pri tome vrijedi da je* $up(i) = u(i - n - 1)$, *i* $yp(j) = y(j - 1)$.
- *Generirajte matricu* U *i vektor* Y *iz ovih podataka, i riješite problem totalnih najmanjih kvadrata za* T *i* D *identitete. Koristite MATLAB-ovu funkciju* `svd()`.

Zadatak (nastavak)

- *Prije rješavanja provjerite uvjete prethodnog teorema i ispišite $\sigma_{n+1}(C_1)$ i $\sigma_{n+2}(C)$.*
- *Ulazne i izlazne vrijednosti up i yp dobivene su perturbiranjem odgovarajućih vrijednosti, čije je egzaktno rješenje hh spremljeno u datoteci*

  `model_dekonvolucija_hh.mat`

  na adresi

  `http://www.math.hr/~nela/zr2.html`
- *Na istom grafu nacrtajte izračunato i egzaktno rješenje.*
- *Što možete zaključiti?*

Računanje gustoće podzemnih stijena

- Varijacija gustoće podzemnih stijena rezultira varijacijama polja gravitacije na zemljinoj površini.
- Zbog toga, na temelju mjerenja polja gravitacije na zemljinoj površini možemo izračunati gustoće podzemnih stijena.
- Varijacija vertikalne komponente polja gravitacije $g(s)$ duž pravca s na površini povezana je sa varijacijom gustoće mase $f(t)$ duž segmenta pravca t $(0 \leq t \leq 1)$ na dubini d ispod površine pomoću Fredholmove integralne jednadžbe prvog reda

$$g(s) = \int_0^1 k(s,t) f(t) dt$$

sa jezgrom

$$k(s,t) = \frac{d}{(d^2 + (s-t)^2)^{3/2}}.$$

- Nakon diskretizacije dobivamo konačnodimenzionalni problem
$$\bar{g} = \bar{K}\bar{f} + \xi,$$
gdje su
  - $\bar{g} = \begin{bmatrix} g_1 & \cdots & g_m \end{bmatrix}^T$ veličine gravitacijske varijacije izmjerene u m točaka duž pravca na površini,
  - $\bar{f} = \begin{bmatrix} f_1 & \cdots & f_n \end{bmatrix}^T$ varijacije gustoće u n točaka duž pravca ispod površine,
  - ξ vektor grešaka mjerenja,
  - $\bar{K} \in \mathbb{R}^{m \times n}$ loše uvjetovana diskretna reprezentacija Fredholmovog integralnog operatora.
- Osim rješavanja problema najmanjih kvadrata i regularizacije, ovaj problem može se još riješiti i pomoću **krnje dekompozicije singularnih vrijednosti (TSVD).**

Krnja dekompozicija singularnih vrijednosti

- Za rješavanje loše uvjetovanih problema često se koristi krnja dekompozicija singularnih vrijednosti (TSVD), koja koristi aproksimaciju ranga $p < \min\{m, n\}$.
- Ako je $A = U\Sigma V^T$ SVD matrice A, tada je prema jednom teoremu

$$A_p = \sum_{i=1}^{p} \sigma_i u_i v_i^T,$$

najbolja aproksimacija ranga p matrice A.
- Za $m \geq n$, neka su matrice $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ i $\Sigma \in \mathbb{R}^{m\times n}$ particionirane na sljedeći način

$$U = \begin{array}{c} \begin{bmatrix} U_1 & U_2 & U_3 \end{bmatrix} \\ \begin{array}{ccc} p & n-p & m-n \end{array} \end{array} \qquad V = \begin{array}{c} \begin{bmatrix} V_1 & V_2 \end{bmatrix} \\ \begin{array}{cc} p & n-p \end{array} \end{array} \qquad \Sigma = \begin{array}{cc} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{bmatrix} & \begin{array}{c} p \\ n-p \\ m-n \end{array} \\ \begin{array}{cc} p & n-p \end{array} & \end{array}$$

gdje je $\sigma_p > \zeta\sigma_1$ i $\sigma_{p+1} < \zeta\sigma_1$ za neku toleranciju ζ.

- Tada je
$$A_p = U_1 \Sigma_1 V_1^T = U(:, 1:p)\Sigma(1:p, 1:p)V(:, 1:p)^T.$$
- Rješavanje problema najmanjih kvadrata svodi se na minimizaciju $\|r_{svd}\|_2^2 = \|A\bar{x} - b\|_2^2$, gdje je
$$\|r_{svd}\|_2^2 = \|\Sigma_1 V_1^T \bar{x} - U_1^T b\|_2^2 + \|\Sigma_2 V_2^T \bar{x} - U_2^T b\|_2^2 + \|U_3^T b\|_2^2,$$
što je ekvivalentno minimizaciji prva dva izraza u gornjoj jednadžbi.
- TSVD postavlja $\sigma_i = 0$ za $i = p+1, \ldots, n$ i minimizira samo prvi izraz.
- To je ekvivalentno rješavanju problema najmanjih kvadrata za matricu A_p
$$\min \|r_{tsvd}\|_2^2 = \min(\|\Sigma_1 V_1^T \tilde{x} - U_1^T b\|_2^2 + \|U_2^T b\|_2^2 + \|U_3^T b\|_2^2).$$
- Važno je odabrati pogodnu toleranciju ζ ili rang p, tako da norma reziduala i norma rješenja budu male.

- Rješenje pomoću TSVD je tada oblika

$$\tilde{x} = \sum_{i=1}^{p} \frac{u_i^T b}{\sigma_i} v_i = V(:, 1:p)\Sigma(1:p, 1:p)^{-1} U(:, 1:p)^T b.$$

Zadatak

- *Veličine gravitacijske varijacije izmjerene u* $m = 15$ *ekvidistantnih točaka duž pravca* $s = [0, 1]$ *na površini dane su u datoteci*

  `model_gravitacija_g.mat`

  na adresi

  http://www.math.hr/~nela/zr2.html
- *Diskretizirajte Fredholmov integralni operator za* $m = 15$ *ekvidistantnih točaka na površini i za* $n = 15$ *ekvidistantnih točaka na dubini* $d = 0.25$ *ispod točaka mjerenja na površini, kao što smo to napravili u ZR1.*

Zadatak (nastavak)

- *Standardna devijacija grešaka mjerenja* ξ *je oko 0.1.*
- *Kružićima nacrtajte singularne vrijednosti matrice* $\bar{K}$*, i uvjerite se u brzinu njihovog opadanja.*
- *Nacrtajte grafove aproksimativnih rješenja rangova* $p = 1, \ldots, 15$ *i usporedite ih sa egzaktnim rješenjem koje glasi*

$$f(t) = \sin(\pi t) + 0.5 \sin(2\pi t).$$

- *Za koji rang p je aproksimacija najbolja?*