Numerička Analiza 1

Zadaća br. 1
Zadana 10.11.2015.
Rok: ponedeljak 16.11.2015. 9:00.

1 Uvod

Cilj ove zadaće je dobiti prva praktična iskustva u primjeni iterativnih metoda za rješavanje linearnih sustava. Istovremeno, navikavate se računati i upoznajete programski sustav MATLAB.

Za pomoć u korištenju MATLAB–a, "zagooglajte" npr. *matlab tutorial* i dobit ćete dugačku listu uvodnih kurseva za MATLAB. Na primjer,
http://www.mathworks.com/academia/student_center/tutorials/
nudi čak i video lekcije. S obzirom da ste na četvrtoj godini studija, sa dosta iskustva s programiranjem, učenje MATLABA bi moralo biti jednostavno i brzo. Kako budemo išli dalje s kolegijem, koristit ćemo ga sve više i na višem nivou, pa je sada pravo vrijeme da svladate osnove. Slobodno se javite ako Vam ne ide.

Implementacije metoda koje smo do sada upoznali možete naći na adresi http://www.netlib.org/. Odite na *Browse the Netlib repository* i zatim odaberite *templates*. Tamo možete naći npr. jednu knjigu s opisima metoda i savjetima kada koju koristiti (knjiga je u **ps** i **pdf** formatu), te metode implementirane kao MATLAB, FORTRAN i C++ programi.

Napomena: Za izradu zadaće možete koristiti sve knjige i druge izvore informacija (Internet), možete raditi sa koleg(ic)ama i možete se s problemima oko zadaće javiti i meni (sve do dan prije roka za predaju). Ali, ono što predate kao Vašu zadaću mora biti Vaše autorsko djelo. Cilj zadaće je učenje a ne sakupljanje bodova za ocjenu. U ocjenjivanje zadaće spada i usmena provjera što znači da možete očekivati da u terminu konzultacija ili na završnom ispitu usmeno 'branite' i objašnjavate detalje rješenja zadaće.

Programe iz ove zadaće predajete tako da ih do 16.11.2015. u 9:00 pošaljete elektroničkom poštom na adresu

```
To:    drmac@math.hr
Subject:NA1-Z1-Ime-Prezime
```

2 Upute - obrađeni materijal

Materijal u skripti koji pokriva sadržaje predavanja, ove zadaće i prvog kolokvija je:

1. Teorijska podloga:
   - Schurova dekompozicija (§1.2.)
   - Spektralni radijus (§1.5.)
   - Lociranje spektra (Geršgorinovi krugovi, §1.6.1)
   - Ireducibilne matrice (§2.1)
2. Klasične iterativne metode:
   - Jacobijeva, Gauss-Seidelova, SOR, SSOR (§3.5. - sve osim sekcije 3.5.4. i dokaza Stein-Rosenbergovog teorema (TM. 3.5.4.)).
   - Polinomijalno ubrzanje konvergencije pomoću Čebiševljevih polinoma (§3.6).

Treba svladati svu teoriju iz gore navedenih tema; neki od zadataka na kolokviju su upravo iz njih (npr. dokazati neku teorijsku tvrdnju).

3 Zadaci

1. Proučiti i implementirati SOR i SSOR metode. Usporediti sa optimalnim SOR(ω). Odaberite pogodan test primjer, npr. iz kolekcije Matrix Market, ili diskretizacija Laplaceove jednadžbe u dimenziji barem dva (npr. pomoću Matlab funkcije delsq - nakon help delsq slijedite delsqdemo za više detalja) ili koristite Vaš omiljeni primjer.

2. Metodom konačnih diferencija diskretizirajte rubni problem

$$
\begin{aligned}
-u''(x) &= 16\pi^2 \sin 4\pi x, \quad 0 < x < 1, && (1)\\
u(0) &= u(1) = 0. && (2)
\end{aligned}
$$

   Dobiveni linearni sustav $Ax = b$ riješite

- Jacobijevom metodom, te Jacobijevom metodom sa polinomijalnim ubrzanjem;
- Gauss-Seidelovom metodom, te Gauss-Seidelovom metodom sa polinomijalnim ubrzanjem.

Sve metode napisati kao Matlab funkcije (pomoć oko sintakse: `>> help function`). Ulazni podaci neka budu A, b, početna iteracija, tolerancija za numeričku konvergenciju, maksimalni broj iteracija i spektralni radijus matrice koja ganerira iteracije. Pri tome, radi jednostavnosti, pretpostavite da imate egzaktni spektralni radijus i računajte ga koristeći Matlab funkcije `eig, max, abs`. Pri implementaciji Čebiševljevih polinoma rekurziju implementirajte s minimalnim utroškom memorije. Izlazni podaci neka budu aproksimacija rješenja i vektor izračunatih reziduala. Usporediti konvergenciju reziduala za sve metode.

Varirajte broj diskretnih točaka i dobivene aproksimacije usporedite s točnim rješenjem (kojeg lako odredite analitički). Usporedbu napravite jednostavno grafički, koristite MATLAB–ovu grafiku, npr. funkciju `plot`.

3. Napišite u Matlabu *matrix free* verziju Gauss–Seidelove metode za numeričko rješavanje Poissonove dvodimenzionalne parcijalne diferencijalne jednadžbe

$$\begin{aligned} -u_{xx}(x,y) - u_{yy}(x,y) &= f(x,y), \quad (x,y) \in \Omega \\ u(x,y) &= 0, \quad (x,y) \in \partial\Omega \end{aligned}$$

gdje je $\Omega = \{(x,y) \ : \ 0 < x, y < 1\} \subset \mathbb{R}^2$ jedinični kvadrat, a $\partial\Omega$ njegov rub.

- Nepoznanice v_{ij} (v_{ij} je diskretna aproksimacija rješenja $u(\cdot,\cdot)$ u točki (x_i, y_j) diskretne mreže) **ne spremati u** $n^2 \times 1$ **vektor** x nego ih držati u prirodnoj $n \times n$ strukturi ($n \times n$ polju V) kako se i pojavljuju na diskretnoj mreži. Izvedite sustav linearnih jednadžbi

$$4v_{ij} - v_{i-1,j} - v_{i+1,j} - v_{i,j-1} - v_{i,j+1} = h^2 f_{ij}, \quad i,j = 1,\ldots,n \tag{3}$$
$$v_{0j} = v_{n+1,j} = v_{i0} = v_{i,n+1} = 0 \tag{4}$$

čije rješenje $V = (v_{ij})_{i,j=1}^n$ aproksimira $U = (u_{ij})$.

- Tome prilagoditi iteracije $x^{(k+1)} = \mathsf{F}x^{(k)} + c$ – dakle zapisati ih u terminima varijabli v_{ij}. Pri tome ne koristiti matrične elemente a_{ij} – jednostavno iskoristiti činjenicu da su oni jednaki 0, -1 ili 4 i to direktno iskoristiti u formulama.

- Dakle, programi trebaju jednostavno računati $v^{(k+1)}$ pomoću $v^{(k)}$ eksplicitnim formulama koje zapravo reprezentiraju Gauss–Seidelovu metodu.

Programi trebaju imati sljedeće dvije komponente:

- Gauss–Seidelova metoda treba biti implementirana u posebnoj funkciji `GS()`. Sami odredite ulazne i izlazne varijable.
- Test program koji treba rješiti konkretan problem u kojem je

$$f(x,y) = 2x - 2x^2 + 2y - 2y^2,$$

a diskretizacija se radi sa po 25 točaka u x i y smjeru. Egzaktno rješenje je $u(x,y) = x(x-1)y(y-1)$. Vaš program treba vratiti dva grafa (`figure(1), figure(2)`) u kojem je na prvom graf egzaktnog rješenja $u(x,y)$ i njega se može dobiti npr. sa

```
[X,Y] = meshgrid(0:0.04:1, 0:0.04:1);
Z = X .* Y.*(X-1).*(Y-1);
figure(1), surf(X,Y,Z)
xlabel('x')
ylabel('y')
zlabel('u(x,y)=x(x-1)y(y-1)')
```

(Uočite da je u naredbi `surf()` `Z(i,j)=` u(`X(i)`,`Y(j)`). Naravno, `help meshgrid`, `help surf` daju sve detalje.)

Na drugom grafu treba biti naredbom `surf(X,Y,V)` prikazana diskretna aproksimacija dobivena rješavanjem odgovarajućeg linearnog sustava, kojeg ste riješili funkcijom `GS()`.

4. Implementirajte "matrix free" Jacobijevu i SSOR(ω) metodu sa Čebiševljevim polinomijalnim ubrzanjem. Program testirajte na Poissonovoj jednadžbi sa domenom $0 \leq x \leq 1; 0 \leq y \leq 2$, $f(x,y) = 10e^{2x+y}$, i rubnim uvjetima $u(0,y) = 2e^{y}$, $u(1,y) = 2e^{2+y}$, $u(x,0) = 2e^{2x}$, $u(x,2) = 2e^{2x+2}$.

   Vaš program treba vratiti dva grafa (`figure(1), figure(2)`) u kojem je na prvom graf egzaktnog rješenja $u(x,y)$ (kojeg lako pogodite) a na drugom graf izračunate numeričke aproksimacije.