Iterativne metode za rješavanje linearnih sustava

Bošnjak, Blaženka

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike

Blaženka Bošnjak

Iterativne metode za rješavanje linearnih sustava

Završni rad

Osijek, 2020.

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike

Blaženka Bošnjak

Iterativne metode za rješavanje linearnih sustava

Završni rad

Mentor: izv.prof.dr.sc. Darija Marković

Osijek, 2020.

Sažetak

Prilikom rješavanja sustava linearnih jednadžbi $Ax = b$, pri čemu za $A \in M_{mn}(\mathbb{R})$ i $b \in M_{m1}(\mathbb{R})$ nastojimo odrediti $x \in M_{n1}(\mathbb{R})$ razlikujemo dva pristupa rješavanja sustava. Prvi način na koji se može odrediti traženi vektor x jesu direktne metode. Direktne se metode koriste uglavnom za rješavanje manjih sustava linearnih jednadžbi te se pomoću njih dobiva egzaktno rješenje sustava. Direktne metode jesu primjerice Cramerovo pravilo i Gaussove eliminacije. S druge strane, pristup rješavanja sustava koji ćemo proučavati u ovom radu jesu iterativne metode koje uvelike olakšavaju rješavanje sustava velikih dimenzija zahtjevne vremenske i prostorne složenosti. Iterativne metode su nizovi matematičkih postupaka čijim se uzastopnim ponavljanjem dolazi do aproksimacije rješenja sustava. No, takvim metodama generalno ne dolazimo do točnog rješenja sustava u konačno mnogo koraka, nego se svakim korakom odstupanje od točnog rješenja smanjuje. Ukoliko pripadna iterativna metoda konvergira, konstruiramo niz $x^{(k)},\ k \in \mathbb{N}$ za koji vrijedi $x^{(k)} \to x$, za $k \to \infty$. U ovom radu proučit ćemo neke najčešće korištene iterativne metode kao što su: Jacobijeva metoda, Gauss-Seidelova metoda, metoda najbržeg silaska, metoda konjugiranih gradijenata.

Ključne riječi: sustav linearnih jednadžbi, iterativne metode, Jacobijeva metoda, Gauss-Seidelova metoda, metoda najbržeg silaska, metoda konjugiranih gradijenata

Abstract

While solving system of linear equations $Ax = b$, where for $A \in M_{mn}(\mathbb{R})$ and $b \in M_{m1}(\mathbb{R})$ we try to determine $x \in M_{n1}(\mathbb{R})$, we distinguish two approaches. The first approach for finding the vector x is using direct methods. Direct methods are mostly used for solving simpler systems of smaller dimensions, giving the exact solution to the system. Such methods are, for example, Cramer's rule and Gaussian elimination. On the other hand, the approach for solving systems of equations we will examine are iterative methods, which greatly simplify the process of solving linear systems of big dimensions and challenging time and space complexity. Iterative methods are series of mathematical procedures which, when applied repeatedly, give an approximated solution of the system. However, in these methods, the exact solution is generally not found, but rather the deviation from the exact solution decrease with each iteration. Iterative methods construct the sequence $x^{(k)},\ k \in \mathbb{N}$, where $x^{(k)} \to x$, for $k \to \infty$. In this paper we will study some of the most commonly used iterative methods, such as the Jacobi method, Gauss-Seidel method, steepest descent method, conjugate gradient method.

Key words: system of linear equations, iterative methods, Jacobi method, Gauss-Seidel method, steepest descent method, method of conjugated gradients

Sadržaj

1 Uvod

Sustavi linearnih jednadžbi jesu jedan od najstarijih matematičkih problema čije se rješavanje proučava stoljećima. Razlikujemo dvije skupine metoda za rješavanje sustava linearnih jednadžbi: diretkne i iterativne metode. Unutar ovoga rada bit će predstavljen jedan od pristupa rješavanja linearnih sustava - iterativne metode, koje uvelike olakšavaju rješavanje velikih sustava linearnih jednadžbi zahtjevne vremenske i prostorne složenosti. Ideja iterativnih metoda datira iz 19. stoljeća kada je njemački matematičar Carl Friedrich Gauss razvio prvu metodu koju možemo smatrati iterativnom. Veliki zamah iterativne metode su doživjele 1950-ih godina razvojem računalne tehnologije.
U ovom radu bit će izložena ideja nastanka iterativnih metoda, zatim razvitak novih metoda modificiranjem ranije poznatih.
Detaljnije ćemo proučiti sljedeće četiri metode:

- Jacobijeva metoda
- Gauss-Seidelova metoda
- metoda najbržeg silaska
- metoda konjugiranih gradijenata

Osim gore navedenih iterativnih metoda navedimo još neke koje ne će biti detaljno obrađene u ovome radu. Primjerice: JOR metoda (*Jacobi overrelaxation*), SOR metoda (*successive overrelaxation*), metoda minimalnog ostatka, metoda dvostrukog konjugiranog gradijenta, metoda konjugiranih gradijanata na normalnim jednadžbama,... Više o spomenutim metodama može se pronaći u [1] i [4].

U posljednjem poglavlju bit će predstavljene implementacije spomenutih iterativnih metoda unutar programskog paketa MATLAB. Programski jezik MATLAB je jezik visokih performansi namijenjen tehničkim proračunima. MATLAB je matrično orijentiran jezik, pa je samim tim koristan za implementaciju, testiranje i vizualizaciju iterativnih metoda.

2 Iterativne metode

Prilikom korištenja direktnih metoda za rješavanje linearnih sustava možemo doći do problema jer dolazak do rješenja sustava zahtjeva velik broj aritmetičkih operacija. Upravo zbog toga su iterativne metode mnogo korisnije, posebice kada je riječ o rijetko popunjenim sustavima (engl. *sparse*), sustavima velikih redova s puno nul-elemenata.
Neke od iterativnih metoda za rješavanje linearnih sustava jesu: Jacobijeva, Gauss-Seidelova, SOR metoda, JOR metoda, metoda konjugiranih gradijenata, metoda konjugiranih smjerova, metoda najbržeg silaska, itd.
Počnimo s idejom nastanka iterativnih metoda.

Neka je zadana regularna matrica $A \in M_n(\mathbb{R})$ te neka su $M, N \in M_n(\mathbb{R})$ takve da matricu A možemo zapisati u obliku

$$A = M + N,$$

gdje se inverz matrice M računa lakše nego inverz matrice A. Tada početni sustav $Ax = b$ postaje sustav $(M + N)x = b$, odnosno

$$Mx = b - Nx,$$

gdje je M regularna matrica. Ako je dana aproksimacija $x^{(k-1)}$, k-tu aproksimaciju rješenja definiramo s :

$$Mx^{(k)} = b - Nx^{(k-1)}, \ k \in \mathbb{N}. \tag{2.1}$$

Pretpostavimo li da vrijedi $\lim\limits_{k\to\infty} x^{(k)} = x$, tada vrijedi

$$Mx = \lim_{k\to\infty} Mx^{(k)} = \lim_{k\to\infty} b - Nx^{(k-1)} = b - Nx.$$

Uočimo da vrijedi $Ax = b$. Dakle, u slučaju da niz (2.1) konvergira tada njegov limes predstavlja rješenje sustava linearnih jednadžbi.
Prilikom proučavanja konvergencije iterativne metode, proučavat ćemo ponašanje k-te pogreške $e^{(k)} = x^{(k)} - x$, pri čemu je x egzaktno rješenje sustava linearnih jednadžbi. Uz pridržavanje dosadašnjih oznaka imamo:

$$Me^{(k)} = Mx^{(k)} - Mx = (b - Nx^{(k-1)}) - (A - N)x = -Ne^{(k-1)},$$

odakle slijedi da je $e^{(k)} = -M^{-1}Ne^{(k-1)}$. Dakle, promatrana iterativna metoda će konvergirati ako i samo ako $e^{(k)} \to 0$ za $k \to \infty$.
Sljedeći teorem je karakterizacija konvergencije iterativne metode pomoću svojstava spektralnog radijusa matrice

$$C = -M^{-1}N.$$

Stoga se ukratko prisjetimo definicije spektralnog radijusa te definicija vektorske, matrične i inducirane norme. Prisjetimo se također nekih teorema koji će nam trebati prilikom dokazivanja.

Definicija 2.1. *Vektorska norma* $\|\cdot\|$ *na* $M_{n1}(\mathbb{C})$ *je preslikavanje* $\|\cdot\| : M_{n1}(\mathbb{C}) \to \mathbb{R}_0^+$ *koje zadovoljava:*

i) $\|x\| \geq 0,\ \forall x \in M_{n1}(\mathbb{C})$ *i* $\|x\| = 0$ *akko* $x = 0$*;*

ii) $\|\alpha x\| = |\alpha|\ \|x\|,\ \forall \alpha \in \mathbb{C},\ x \in M_{n1}(\mathbb{C})$*;*

iii) $\|x + y\| \leq \|x\| + \|y\|,\ \forall x, y \in M_{n1}(\mathbb{C})$.

Definicija 2.2. *Matrična norma na* $M_n(\mathbb{C})$ *je preslikavanje* $\|\cdot\| : M_n(\mathbb{C}) \to \mathbb{R}$ *za koje vrijedi:*

a) $\|A\| \geq 0,\ \forall A \in M_n(\mathbb{C})$ *i* $\|A\| = 0$ *akko je* A *nul-matrica;*

b) $\|\alpha A\| = |\alpha|\ \|A\|,\ \forall \alpha \in \mathbb{C},\ A \in M_n(\mathbb{C})$*;*

c) $\|A + B\| \leq \|A\| + \|B\|,\ \forall A, B \in M_n(\mathbb{C})$*;*

d) $\|AB\| \leq \|A\|\ \|B\|,\ \forall A, B \in M_n(\mathbb{C})$.

Definicija 2.3. *Neka je zadana vektorska norma* $\|\cdot\|$ *na* $M_{n1}(\mathbb{C})$*. Inducirana norma* $\|\cdot\|$ *na* $M_n(\mathbb{C})$ *definira se sa*

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|y\|=1} \|Ay\|, \quad A \in M_n(\mathbb{C}).$$

Definicija 2.4. *Spektralni radijus matrice* $A \in M_n(\mathbb{C})$ *definiramo sa*

$$\rho(A) = \max\{|\lambda| : \lambda \ svojstvena\ vrijednost\ matrice\ A\}.$$

Teorem 2.5. *Matrična norma inducirana vektorskom normom* ∞ *računa se pomoću jednakosti:*

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

Dokaz se može vidjeti u [5].

Teorem 2.6. *Za bilo koju matričnu normu* $\|\cdot\|$*, bilo koju matricu* $A \in M_n(\mathbb{C})$ *i bilo koji* $k \in \mathbb{N}$ *vrijedi*

$$\rho(A)^k \leq \|A^k\| \leq \|A\|^k.$$

Dokaz se može vidjeti u [5].

Teorem 2.7. *Neka je dana matrica* $C \in M_n(\mathbb{C})$ *i vektori* $e^{(0)} \in M_{n1}(\mathbb{C})$ *i* $e^{(k)} = C^k e^{(0)} \in M_{n1}(\mathbb{C}),\ \forall k \in \mathbb{N}$*. Tada* $e^{(k)} \to 0$ *za svaki* $e^{(0)} \in M_{n1}(\mathbb{C})$ *ako i samo ako je* $\rho(C) < 1$.

Prije dokaza Teorema 2.7 uočimo da $\rho(C)$ ovisi samo o matrici A odnosno njenom rastavu, a ne o vektoru b iz $Ax = b$.

Dokaz. Neka je $\rho(C) < 1$. Možemo naći induciranu matričnu normu takvu da je $\|C\| < 1$ pa slijedi

$$\|e^{(k)}\| = \|C^k e^{(0)}\| \leq \|C\|^k \|e^{(0)}\| \to 0, \text{za } k \to \infty.$$

Pokažimo sada obrat. Neka je $e^{(k)} \to 0$, kada bi $\rho(C \geq 1$, onda bi postojao barem jedan $e^{(0)} \in M_{n1}(\mathbb{C})$ za koji bi vrijedilo da je $Ce^{(0)} = \lambda e^{(0)}$, za $\lambda \in \mathbb{C}$ takav da $|\lambda| \geq 1$. Za vektor $e^{(0)}$ dobivamo

$$\|e^{(k)}\| = \|C^k e^{(0)}\| = |\lambda|^k \|e^{(0)}\|.$$

Iz ovoga vidimo da $e^{(k)}$ ne konvergira prema 0 za $k \to \infty$, što dovodi do kontradikcije. □

Napomena 2.8. *Dovoljno je provjeriti je li $\|C\| < 1$ za neku matričnu normu jer uzmemo li u Teoremu 2.6 , $k = 1$ vrijedi $\rho(C) \leq \|C\|$.*

Ideja iterativnih metoda je brzo računanje $x^{(k+1)}$ iz $x^{(k)}$. Proces se zaustavlja kada je $x^{(k)}$ dovoljno dobra aproksimacija točnog rješenja x. Međutim, kako pravo rješenje x ne znamo, nailazimo na problem. Stoga, koristimo svojstvo da je konvergentan niz Cauchyjev, tj. da susjedni članovi niza moraju postati po volji bliski. Uzima se da su $x^{(k+1)}$ i $x^{(k)}$ dovoljno bliski ako vrijedi:

$$\|x^{(k+1)} - x^{(k)}\| \leq \epsilon,$$

gdje je ϵ unaprijed zadana točnost, a $\|\cdot\|$ neka vektorska norma.

Definicija 2.9. *Za niz aproksimacija $x^{(k)}$, $k \in \mathbb{N}_0$, reći ćemo da konvergira prema x brzinom p ako je*

$$\|x^{(k+1)} - x\| \leq a\|x^{(k)} - x\|^p, a \in \mathbb{R}_0^+.$$

Za $p = 1$ kažemo da je brzina kovergencije linerana, a ako je $a < 1$ brzina je geometrijska s faktorom a.

Navedene definicije i teoremi preuzeti su iz [5].
U sljedećim potpoglavljima ćemo proučavati metode kod kojih su matrice M i N iz (2.1) specijalno konstruirane.

2.1 Jacobijeva metoda

Neka je dana matrica $A \in M_n(\mathbb{R})$ te neka su matrice $L, D, U \in M_n(\mathbb{R})$ zadane kao:

$$L = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ a_{21} & 0 & \dots & 0 & 0 \\ a_{31} & a_{32} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn-1} & 0 \end{bmatrix}, \quad D = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ 0 & 0 & a_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & 0 & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{n-1n} \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{2.2}$$

Uočimo da matricu A možemo zapisati kao sumu navedenih triju matrica. Iterativnu metodu dobivenu iz (2.1) ukoliko uzmemo da je $M = D$, a $N = L + U$, pri čemu su matrice L, D, U gore definirane matrice nazivamo *Jacobijeva metoda.* Obzirom na takav izbor matrica M i N, iterativni postupak poprima sljedeći oblik:

$$x^{(k)} = D^{-1}(b - (L + U)x^{(k-1)})$$

za sve $k \in \mathbb{N}$.
Konvergencija iterativne metode ovisi o svojstvima matrice $C_{Jacobi} = -M^{-1}N = -D^{-1}(L+U)$. Budući je D dijagonalna matrica, inverz matrice računa se trivijalno. Stoga konstrukcija matrice C_{Jacobi} ne zahtjeva velik broj računskih operacija.
Označimo sa $x^{(k)} = \left(x_1^{(k)}, \ldots, x_n^{(k)}\right)^T$, tada aproksimaciju $x^{(k+1)}$ rješenja sustava linearnih jednadžbi dobivamo kao rješenje sustava:

$$\begin{aligned}
x_1^{(k+1)} &= c_{12}x_2^{(k)} + c_{13}x_3^{(k)} + \cdots + c_{1n}x_n^{(k)} + \beta_1 \\
x_2^{(k+1)} &= c_{21}x_1^{(k)} + c_{23}x_3^{(k)} + \cdots + c_{2n}x_n^{(k)} + \beta_2 \\
&\vdots \\
x_n^{(k+1)} &= c_{n1}x_1^{(k)} + c_{n2}x_2^{(k)} + \cdots + c_{nn-1}x_{n-1}^{(k)} + \beta_n,
\end{aligned}$$

pri čemu su $c_{ij} = -\frac{a_{ij}}{a_{ii}}, \beta_i = \frac{b_i}{a_{ii}}$.
Da bi Jacobijeva metoda konvergirala moraju biti zadovoljeni neki uvjeti. Definirajmo svojstvo *stroge dijagonalne dominantnosti* matrice.

Definicija 2.10. *Neka je dana matrica A. Kažemo da je:*

i) matrica A strogo dijagonalno dominantna po retcima ako vrijedi

$$|a_{ii}| > \sum_{\substack{i=1 \\ j\neq i}}^{n} |a_{ij}|, \quad i = 1, \ldots, n;$$

ii) matrica A strogo dijagonalno dominantna po stupcima ako vrijedi

$$|a_{jj}| > \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}|, \quad j = 1, \ldots, n.$$

Teorem 2.11. *Jacobijeva metoda konvergira za sve matrice A za koje vrijedi:*

1.) stroga dijagonalna dominacija po retcima;

2.) stroga dijagonalna dominacija po stupcima.

Dokaz. Pokažimo najprije da Jacobijeva metoda konvergira za sve matrice A koje su strogo dijagonalno dominantne po retcima. Elementi matrice $C_{Jacobi} = -D^{-1}(L+U)$ su oblika

$$c_{ij} = \begin{cases} -\frac{a_{ij}}{a_{ii}}, & i \neq j \\ 0, & i = j \end{cases}.$$

Prema Teoremu 2.5 $\|C_{Jacobi}\|_\infty = \max\limits_{i=1,\ldots,n} \dfrac{1}{\|a_{ii}\|} \sum\limits_{\substack{i=1 \\ j\neq i}}^{n} |a_{ij}|$. Stroga dijagonalna dominacija po retcima implicira da je $\|C_{Jacobi}\|_\infty < 1$, što nadalje implicira da je $\rho(A) < 1$. Prema Teoremu 2.7

ova metoda konvergira.
Pokažimo sada da Jacobijeva metoda konvergira za sve matrice A uz uvjet stroge dijagonalne dominacije po stupcima, tj. ako vrijedi

$$|a_{jj}| > \sum_{\substack{i=1 \\ i \neq j}}^{n} |a_{ij}|, \quad j = 1, \ldots, n.$$

Prisjetimo se definicije adjungirane matrice:
Neka je dana matrica $A \in M_{mn}(\mathbb{C})$. Adjungiranu matricu $A^* \in M_{mn}(\mathbb{C})$ definiramo s $A^* = [\,\overline{a_{ji}}\,]$.
Tada je pripadna adjungirana matrica A^* matrice A strogo dijagonalno dominantna po retcima, odnosno vrijedi:

$$|a_{ii}| > \sum_{\substack{i=1 \\ j \neq i}}^{n} |a_{ij}|, \quad i = 1, \ldots, n$$

i metoda konvergira za matricu A^*. Prema Teoremu 2.7 slijedi da je $\rho(-D^{-*}(L^* + U^*)) < 1$. Za svaku matricu C, matrice C, C^*, $D^{-1}C^*D$ imaju iste spektre. Iz niza jednakosti:

$$\rho(-D^{-1}(L+U)) = \rho(-D^{-1}(L+U)D^{-1}D) = \rho(-(L+U)D^{-1}) = \rho(-D^{-*}(L^* + U^*)) < 1$$

slijedi konvergencija Jacobijeve metode za matricu A. $\square$

Algoritam 2.12. ***(Jacobijeva metoda)***

$$\textbf{\textit{for}}\ j := 1\ \textbf{\textit{to}}\ n\ \textbf{\textit{do}}$$

$$x_j^{(k+1)} := \Big(b_j - \sum_{\substack{i=1 \\ i \neq j}}^{n} a_{ji} x_i^{(k)}\Big) \Big/ a_{jj}$$

Uočimo da komponente novog vektora $x^{(k+1)}$ ovise samo o komponentama starog vektora $x^{(k)}$. Zato je Jacobijeva metoda idealna za paralelno računanje, jer pojedine komponente novog vektora računamo potpuno nezavisno.

2.2 Gauss-Seidelova metoda

Komponente novog vektora u Jacobijevoj metodi računaju se sekvencijalno, od prve prema zadnjoj, stoga se nameće ideja modifikacije metode. Poboljšanje Jacobijeve metode jest upravo *Gauss-Seidelova iterativna metoda.*
Pretpostavimo da je $x^{(k)} = \left(x_1^{(k)}, \ldots, x_n^{(k)}\right)^T$ k-ta aproksimacija rješanja. Tada je $(k+1)$-va aproksimacija rješenje sustava:

$$\begin{aligned}
a_{11}x_1^{(k+1)} + a_{12}x_2^{(k)} + \cdots + a_{1n}x_n^{(k)} &= b_1 \\
a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} + \cdots + a_{2n}x_n^{(k)} &= b_2 \\
&\ \vdots \\
a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + \cdots + a_{nn}x_n^{(k+1)} &= b_n.
\end{aligned}$$

Uočimo da gore navedeni sustav možemo zapisati u matričnom obliku:

$$(L+D)x^{(k+1)} + Ux^{(k)} = b,$$

odnosno

$$(L+D)x^{(k+1)} = b - Ux^{(k)}. \tag{2.3}$$

Također uočimo da formulu iterativnog postupka dobijemo specijalnim odabirom matrica M i N u jednakosti (2.1) tako da je $M = L + D$, a $N = U$, gdje su matrice L, D, U definirane u (2.2). Formula iterativnog postupka je oblika:

$$x^{(k+1)} = (L+D)^{-1}b - (L+D)^{-1}Ux^{(k)}.$$

Primjetimo da treba odrediti inverz matrice $L + D$ što je znatno teže nego odrediti inverz dijagonalne matrice D. Pomnožimo li jednadžbu (2.3) s D^{-1}, dobivamo sljedeći izraz:

$$(D^{-1}L + I)x^{(k+1)} = D^{-1}b - D^{-1}Ux^{(k)}.$$

Odakle je $x^{(k+1)} = D^{-1}b - D^{-1}Ux^{(k)} - D^{-1}Lx^{(k+1)}$. Tako dolazimo do algoritma kojeg nazivamo *Gauss-Seidelova metoda.*
Konvergencija iterativne metode ovisi o svojstvima matrice $C_{GS} = -(L+D)^{-1}U$.

Algoritam 2.13. ***(Gauss-Seidelova metoda)***

$$\textbf{\textit{for}}\ j := 1\ \textbf{\textit{to}}\ n\ \textbf{\textit{do}}$$
$$x_j^{(k+1)} := \left(b_j - \sum_{i=1}^{j-1} a_{ji}x_i^{(k+1)} - \sum_{i=j+1}^{n} a_{ji}x_i^{(k)}\right)\Big/ a_{jj}$$

Kod računanja j-te komponente $(k+1)$-ve iteracije koriste se do tada izračunate komponente od $x^{(k+1)}$. To omogućava da komponente od $x^{(k+1)}$ prebrišu stare vrijednosti komponenata od $x^{(k)}$, čim se izračunaju te tako ne moramo pamtiti dva vektora, nego samo jedan. Redoslijed računanja komponenti iteracije kod Gauss-Seidelove metode je bitan. U algoritmu smo koristili prirodni slijed, tj. od prve prema zadnjoj komponenti. Ali to nije jedini mogući redoslijed. Možemo uzeti bilo koji drugi redoslijed, odnosno bilo koju drugu od $n!$ permutacija jednadžbi. No, zbog sekvencijalnosti, rezultat će se razlikovati, tj. razlikovat će se iterativna metoda. Iskažimo nekoliko teorema o konvergenciji Gauss-Seidelove metode.

Teorem 2.14. *Ako je matrica A strogo dijagonalno dominantna po retcima, onda Gauss-Seidelova metoda konvergira.*

Dokaz se može pogledati u [1].
Uočimo da je prema Napomeni 2.8 dovoljno pokazati da je $\|C_{GS}\| < 1$.

Teorem 2.15. *Ako je A strogo dijagonalno dominantna matrica po retcima, onda i Jacobijeva i Gauss-Seidelova metoda konvergiraju i vrijedi:*

$$\|C_{GS}\|_\infty \leq \|C_{Jacobi}\|_\infty < 1.$$

Dokaz se može pogledati u [3].
Prije nego iskažemo sljedeći teorem prisjetimo se definicije hermitske i pozitivno definitne matrice.

Definicija 2.16. *Matrica $A \in M_n(\mathbb{C})$ je hermitska ako vrijedi $A^* = A$, (za realne matrice koristimo termin simetična). Hermitska matrica $A \in M_n(\mathbb{C})$ je pozitivno definitna ako vrijedi da je $x^*Ax > 0$, $\forall x \in M_{n1}(\mathbb{C} \setminus \{0\})$.*

Teorem 2.17. *Ako je matrica A hermitska i pozitivno definitna, tada Gauss-Seidelova metoda konvergira za svaku početnu iteraciju $x^{(0)}$.*

Dokaz teorema se može naći u [1].
Proučavajući navedene rezultate zaključujemo da je pod nekim uvjetima Gauss-Seidelova metoda brža u odnosu na Jacobijevu. No, ne postoji nikakva generalizacija toga. Dakle, Gauss–Seidelova metoda u najgorem slučaju konvergira barem tako brzo kao Jacobijeva metoda u najgorem slučaju. To ne znači da će Gauss–Seidelova metoda konvergirati brže nego Jacobijeva za bilo koji problem $Ax = b$.

2.3 Metoda najbržeg silaska

Unutar ovoga potpoglavlja detaljnije ćemo proučiti iterativnu metodu poznatu pod nazivom *metoda najbržeg silaska*. To je metoda za rješavanje sustava $Ax = b$, $A \in M_n(\mathbb{C})$, b, $x \in M_{n1}(\mathbb{C})$, pri čemu je uvjet da matrica sustava A bude hermitska:

- $A^* = A$;
- $y^*Ay > 0, \ \forall y \in M_{n1}(\mathbb{C}), \ y \neq 0.$

Promotrimo iterativnu metodu danu oblikom:

$$x_{k+1} = x_k + \alpha_k(b - Ax_k) = x_k + \alpha_k r_k. \tag{2.4}$$

Ideja odabira parametra α_k u k-toj iteraciji metode dane izrazom 2.4 je ta da se minimizira neka norma greške $e_{k+1} = e_k - \alpha_k r_k$. Problem je što je greška jednako tako nepoznata kao i samo rješenje, pa norma $\|\cdot\|_2$ nije opcija. Za hermitsku matricu ima smisla definirati A-normu $\|\cdot\|_A$ kao $\|x\|_A = \sqrt{(x,x)_A} = \sqrt{x^*Ax}$, pri čemu skalarni produkt $(x,x)_A$ nazivamo A-skalarni produkt. Definirajmo funkciju $f : \mathbb{R} \to \mathbb{R}$ kao

$$\begin{aligned} f(\alpha_k) &= \|e_{k+1}\|_A^2 \\ &= e_{k+1}^* A e_{k+1} \\ &= \alpha_k^2 r_k^* A r_k - 2\alpha_k r_k^* A e_k + e_k^* A e_k \\ &= \alpha_k^2 r_k^* A r_k - 2\alpha_k r_k^* r_k + e_k^* A e_k \end{aligned}$$

Traženje minimuma funkcije $f(\alpha_k)$ je ekvivalentno traženju minimuma $\|e_{k+1}\|_A$. Kako je $f(\alpha_k)$ kvadratna funkcija po varijabli α_k i parametar uz α_k^2 je $r_k^*Ar_k \geq 0$, to implicira da funkcija poprima minimum u tjemenu, koje je jedina nultočka derivacije funkcije f.

$$0 = f'(\alpha_k) = 2\alpha_k r_k^* A r_k - 2r_k^* r_k,$$

odakle slijedi da se minimalna A-norma greške e_{k+1} postiže za

$$\alpha_k = \frac{r_k^* r_k}{r_k^* A r_k}.$$

Posljedica ovakvog odabira parametra α_k je okomitost r_{k+1} i r_k, tj.vrijedi:

$$r_k^* r_{k+1} = r_k^* r_k - \alpha_k r_k^* A r_k = r_k^* r_k - \frac{r_k^* r_k}{r_k^* A r_k} r_k^* A r_k = 0.$$

Zbog okomitosti r_{k+1} i r_k slijedi da je:

$$\begin{aligned} \|e_k\|_A^2 &= e_{k+1}^* A e_{k+1} + 2\alpha_k r_k^* A e_{k+1} + \alpha_k^2 r_k^* A r_k \\ &= \|e_{k+1}\|_A^2 + \alpha_k^2 \|r_k\|_A^2 > \|e_{k+1}\|_A^2. \end{aligned}$$

Odavde se vidi da se A-norma greške smanjuje svakim korakom. Ovu metodu, prema načinu odabira parametra, nazivamo *metodom najbržeg silaska.* Analizom konvergencije metode najbržeg silaska može se pokazati da vrijedi sljedeća ocjena:

$$\|e_k\|_A \leq \left(\frac{\kappa(A) - 1}{\kappa(A) + 1} \right)^k \|e_0\|_A, \tag{2.5}$$

gdje je $\kappa(A) = \dfrac{\lambda_{max}}{\lambda_{min}}$ uvjetovanost matrice A. Iz 2.5 možemo zaključiti da za loše uvjetovane matrice možemo imati sporu konvergenciju jer $\dfrac{\kappa(A) - 1}{\kappa(A) + 1} \approx 1$.

Algoritam 2.18. ***Metoda najbržeg silaska*** $(A,\ b,\ x_0)$

$$\begin{aligned} & r_0 = b - A x_0 \\ & \textbf{\textit{for}}\ k = 1, 2, \ldots \\ & \textbf{\textit{do}}\ z = A r_{k-1} \\ & \alpha_{k-1} = \frac{r_{k-1}^* r_{k-1}}{r_{k-1}^* z} \\ & x_k = x_{k-1} + \alpha_{k-1} r_{k-1} \\ & r_k = r_{k-1} - \alpha_{k-1} z \end{aligned}$$

2.4 Metoda konjugiranih gradijenata

Kako bismo izbjegli sporu konvergenciju metode najbržeg silaska, unaprijed biramo skup A-ortogonalnih vektora, odnosno smjerove traganja $d_0, d_1, \ldots, d_{n-1}$ tako da su vektori

$$d_k = r_k + \beta_k d_{k-1}, k = 0, \ldots, n$$

budu međusobno ortogonalni u A-skalarnom produktu. Dakle, za dva vektora d_k i d_{k-1} vrijedi

$$0 = (d_k, d_{k-1})_A = d_k^* A d_{k-1} = r_k^* A d_{k-1} + \beta_k d_{k-1}^* A d_{k-1},$$

odakle slijedi da je

$$\beta_k = -\frac{r_k^* A d_{k-1}}{d_{k-1}^* A d_{k-1}}.$$

Svakim korakom iteracije biramo aproksimaciju rješenja $x_{k+1} = x_k + \alpha_k d_k$ s minimalnom A-normom greške.
Definirajmo funkciju $g : \mathbb{R} \to \mathbb{R}$ kao $g(\alpha_k) = \|e_{k+1}\|_A^2 = e_{k+1}^* A e_{k+1}$. Primjetimo da vrijedi

$$e_{k+1} = x - x_{k+1} = e_k - \alpha_k d_k;$$

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

Izjednačimo li derivaciju funkcije $g(\alpha_k)$ s 0 dobit ćemo izraz:

$$g'(\alpha_k) = \alpha_k d_k^* A d_k - d_k^* A e_k = 0$$

iz kojeg slijedi $\alpha_k = \dfrac{d_k^* A e_k}{d_k^* A d_k}$, a kako je $r_k = Ae_k$ dobivamo:

$$\alpha_k = \frac{d_k^* r_k}{d_k^* A d_k} = \frac{(r_k, d_k)}{(Ad_k, d_k)}. \tag{2.6}$$

Ovako dobivenu metodu nazivamo *metoda konjigiranih smjerova.* U ovom radu metodu konjugiranih smjerova nećemo dublje proučavati, nego ćemu ju samo iskoristiti za *metodu konjugiranih gradijenata.* Iskažimo teorem o svojstvima metode konjugiranih smjerova koji ćemo kasnije iskoristiti.

Teorem 2.19. *Za metodu konjugiranih smjerova vrijede sljedeća svojstva:*

$$(Ad_i, d_j) = 0, \ \text{za } i \neq j; \tag{2.7}$$
$$(r_i, d_j) = (Ae_i, d_j) = 0, \ \text{za } j < i; \tag{2.8}$$
$$(r_0, d_i) = (r_1, d_i) = \cdots = (r_i, d_i). \tag{2.9}$$

Skalar α_k možemo zapisati kao

$$\alpha_k = \frac{(r_0, d_k)}{(Ad_k, d_k)}.$$

Skup A-ortogonalnih vektora $\{d_0, \ldots, d_{n-1}\}$ možemo dobiti primjenom Gram-Schmidtove metode A-ortogonalizacije na skup linearno nezavisnih vektora $\{u_0, \ldots, u_{n-1}\}$ sa skalarnim produktom $(\cdot,\cdot)_A$. A-ortogonalne vektore možemo dobiti u obliku

$$d_k = u_k + \sum_{i=0}^{k-1} \beta_{ki} d_i, \ \text{gdje su koeficijenti oblika: } \beta_{ki} = -\frac{d_i^* A u_k}{d_i^* A d_i}$$

Metoda konjugiranih gradijenata jest zapravo metoda konjugiranih smjerova kod koje se smjerovi traganja konstruiraju primjenom Gram-Schmidtove metode A-ortogonalnosti na reziduale, odnosno uzima se je $u_i = r_i$. Kako su vektori r_i dobiveni metodom konjugiranih smjerova oni

su linearno nezavisni što se može provjeriti pomoću teorema 2.19.
Vrijedi da je

$$[\{d_0, \dots, d_{k-1}\}] = [\{r_0, \dots, r_{k-1}\}].$$

Budući je r_k ortogonalan na prethodne smjerove traganja vrijedi:

$$r_i^* r_j = 0, \text{ za } i \neq j.$$

Promotrimo sljedeći skalarni produkt:

$$r_{i+1}^* r_k = r_i^* r_k - \alpha_i d_i^* A r_k.$$

Odakle vrijedi

$$d_i^* A r_k = \frac{1}{\alpha_i}(r_i^* r_k - r_{i+1}^* r_k) \tag{2.10}$$

Za $i < k-1$ lijeva strana jednadžbe 2.10 je jednaka 0, pa su $\beta_{ki} = 0$ za $i = 0, 1, 2, \dots, k-2$, a za $\beta_k = \beta_{k,k-1}$ zbog izraza za α_{k-1} i 2.6 i 2.8 vrijedi:

$$\beta_k = -\frac{d_{k-1}^* A r_k}{d_{k-1}^* A d_{k-1}} = \frac{r_k^* r_k}{\alpha_{k-1} d_{k-1}^* A d_{k-1}} = \frac{r_k^* r_k}{d_{k-1}^* r_{k-1}} = \frac{r_k^* r_k}{r_{k-1}^* r_{k-1}}.$$

Također, možemo i α_k zapisati u malo drugačijem obliku:

$$\alpha_k = \frac{r_k^* r_k}{d_k^* A d_k}.$$

Uočimo da, ukoliko nismo našli egzaktno rješenje u k-tom koraku, α_k je pozitivan. Iskazat ćemo u obliku teorema ocjenu pogreške metode konjugiranih gradijenata čiji se dokaz može naći u [1].

Teorem 2.20. *Za metodu konjugiranih gradijanata vrijedi ocjena:*

$$\|e_k\|_A \leq \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k \|e_0\|_A, \tag{2.11}$$

gdje je $\kappa(A) = \dfrac{\lambda_{max}}{\lambda_{min}}$ *uvjetovanost matrice* A.

Algoritam 2.21. *Metoda konjugiranih gradijenata* $(A,\ b,\ x_0)$

$$
\begin{aligned}
& d_0 = r_0 = b - Ax_0 \\
& \textbf{\textit{for}}\ k = 1, 2, \dots, \\
& \textbf{\textit{do}}\ z = Ad_{k-1} \\
& \alpha_{k-1} = \frac{r_{k-1}^* r_{k-1}}{d_{k-1}^* z} \\
& x_k = x_{k-1} + \alpha_{k-1} d_{k-1} \\
& r_k = r_{k-1} - \alpha_{k-1} z \\
& \beta_k = \frac{r_k^* r_k}{r_{k-1}^* r_{k-1}} \\
& d_k = r_k + \beta_k d_{k-1}
\end{aligned}
$$

3 Implementacija iterativnih metoda u MATLAB

Unutar ovoga poglavlja bit će pretstavljene impelemntacije prethodno promatranih iterativnih metoda unutar programskog paketa MATLAB. MATLAB (Matix Laboratory) jest programski jezik visokih performansi namjenjen za računanje, vizualizaciju i programiranje u lako uporabljivoj okolini u kojoj su i problem i rješenje definirani poznatom matematičkom notacijom. MATLAB se koristi za razvoje algoritama, analizu i obradu podataka, izgradnju grafičkog korisničkog sučelja.

3.1 Jacobijeva metoda - implementacija u MATLAB

Riješimo sustav $Ax = b$ Jacobijevom metodom, odnosno odredimo aproksimaciju $x^{(k)}$ tako da $\|\cdot\|_\infty$ reziduala u odnosu na točno rješenje bude manja od zadane tolerancije.
Neka je zadana matrica $A \in M_n(\mathbb{R})$, početna aproksimacija $x_0 \in M_{n1}(\mathbb{R})$, vektor $b \in M_{n1}(\mathbb{R})$, te tolerancija tol.

```
function[br_koraka,spec_rad]=jacobi(A,x0,b,tol)

D=diag(diag(A));
N=A-D;

for ii=1:n
    for jj=1:n
        C(ii,jj)=-D(ii,jj)\N(ii,jj);
    end
end
e=eig(C);
spec_rad=max(abs(e));
br_koraka=0;
while norm(A*x-b,'inf')>=tol
      for kk=1:n
          x1=(D(kk,kk))\(b-N*x0));
      end
      x=x1;
      br_koraka=br_koraka+1;
      greska(broj_koraka)=norm(A*x0-b,'inf');
end
disp(sprintf('Broj iteracija Jacobijeve metode: %d',br_koraka))
```

Na početku implementacije definirali smo rastav matrice A na odgovarajuće matrice D i N. Uočimo da je matrica C matrica konvergencije Jacobijeve metode. Jedan od načina da provjerimo konvergira li Jacobijeva metoda je taj da je spektralni radijus matrice C manji od 1. To je jedan od uvjeta koji jamči konvergenciju metode. U implementaciji smo koristili while petlju koja provjerava je li $\|\cdot\|_\infty$ reziduala veća ili jednaka unaprijed zadanoj toleranciji tol. While petlja će se izvršavati sve dok je ona veća ili jednaka vrijednosti tol. Unutar while petlje koristili smo for petlju kojom smo računali sljedeću aproksimaciju. Prilikom napuštanja for petlje, vrijednost prethodne aproksimacije brišemo sadašnjom aproksimacijom i broj koraka inkrementiramo za 1. Nakon što napustimo while petlju, dobivamo aproksimaciju obzirom na zadanu toleranciju, te ispisujemo potreban broj koraka Jacobijeve metode.

3.2 Gauss-Seidelova metoda - implementacija u MATLAB

Riješimo sustav $Ax = b$ Gauss-Seidelovom iterativnom metodom, odnosno odredimo aproksimaciju $x^{(k)}$ tako da $\|\cdot\|_\infty$ reziduala u odnosu na točno rješenje bude manja od zadane tolerancije. Neka je zadana matrica $A \in M_n(\mathbb{R})$, početna aproksimacija $x_0 \in M_{n1}(\mathbb{R})$, vektor $b \in M_{n1}(\mathbb{R})$, te tolerancija tol.

```
function[br_koraka,spec_rad]=gaussseidel(A,x0,b,tol)

M=tril(A);
N=A-M;
C=zeros(n);

for ii=1:n
    for jj=1:n
        C(ii,jj) = -M(ii,jj)\N(ii,jj);
    end
end
e=eig(C);
spec_rad=max(abs(e));
br_koraka=0;
x1=x;
while norm(A*x0-b,'inf')>=tol
      for kk=1:n
          x1=(D(kk,kk))\(b-N*x0);
      end
      x=x1
      br_koraka=br_koraka + 1;
      greska(br_koraka) = norm(A*x0-b,'inf');
end
disp(sprintf('Broj iteracija Gauss-Seidelove metode: %d',br_koraka))
```

Također, kao i kod implementacije Jacobijeve metode najprije smo definirali rastav matrice A na donjetrokutastu M i strogo gornjetrokutastu matricu N. Nakon što pomoću dvaju for petlji računamo matricu konvergencije C Gauss-Seidelove metode računamo spektralni radijus spomenute matrice koji nam jamči konvergenciju metode ukoliko je manji od 1. While petlja će se ponavljati sve dok $\|\cdot\|_\infty$ reziduala bude veća ili jednaka zadanoj toleranciji tol. For petljom računamo vrijednost sljedeće aproksimacije te nakon završetka for petlje vrijednost prošle aprsimacije postaje nova aproksimacija te inkrementiramo broj koraka za 1. Nakon while petlje dobivamo aproksimaciju uvjetovan unaprijed zadanom tolerancijom tol, te ispisujemo broj koraka potrebnih prilikom postupka Gauss-Seidelove metode.

3.3 Metoda najbržeg silaska - implementacija u MATLAB

Riješimo sustav $Ax = b$ metodom najbržeg silaska, odnosno odredimo aproksimaciju $x^{(k)}$ tako da $\|\cdot\|_\infty$ reziduala u odnosu na točno rješenje bude manja od zadane tolerancije.
Neka je zadana matrica $A \in M_n(\mathbb{R})$, početna aproksimacija $x_0 \in M_{n1}(\mathbb{R})$, vektor $b \in M_{n1}(\mathbb{R})$, te tolerancija tol.

```
function[br_koraka,spec_rad]=metodanajbrzegsilaska(A,x0,b,tol)

r0= b-A*x0;
br_koraka=0;

while norm(A*x0-b,'inf') >= tol
      z=A*r0;
      alpha0=(r0'*r0)/(r0'*z);
      x1=x0+alpha0*r0;
      r1=r0-alpha0*z;
      br_koraka=br_koraka + 1;
      greska(br_koraka)=norm(A*x0-b, 'inf');
      r0=r1;
      x0=x1;
end
disp(sprintf('Broj iteracija metode najbržeg silaska: %d',br_koraka))
```

Prilikom implementacije metode najbržeg silaska najprije smo definirali rezidual u obliku razlike vektora b i Ax_0 te broj koraka inicijalizirali na 0. Zatim, unutar while petlje smo koristili ranije proučavane rezultatne koje smo sistematizirali unutar Algoritma 2.18. Nakon napuštanja while petlje ispisujemo broj potrebnih koraka iteracije

3.4 Metoda konjugiranih gradijenata - implementacija u MATLAB

Riješimo sustav $Ax = b$ metodom konjugiranih gradijenata, odnosno odredimo aproksimaciju $x^{(k)}$ tako da $\|\cdot\|_\infty$ reziduala u odnosu na točno rješenje bude manja od zadane tolerancije.
Neka je zadana matrica $A \in M_{mn}(\mathbb{R})$, početna aproksimacija $x_0 \in M_{n1}(\mathbb{R})$, vektor $b \in M_{m1}(\mathbb{R})$, te tolerancija tol.

```
function[br_koraka,spec_rad]=metodakonjugiranihgradijenata(A,x0,b,tol)

r0=b-A*x0;
d0=r0;
br_koraka=0;

while norm(A*x0-b,'inf')>=tol
      z=A*d0;
      alpha0=(r0'*r0)/(d0'*z);
      x1=x0+alpha0*d0;
      r1=r0-alpha0*z;
```

```
        beta1=(r1'*r1)/(r0'*r0);
        d1=r1+beta1*d0;
        br_koraka=br_koraka+1;
        r0=r1;
        x0=x1;
        d0=d1;
end
disp(sprintf('Broj iteracija metode konjugiranih gradijenata : %d',br_koraka))
```

U implementaciji smo najprije definirali rezidual kao razliku vektora b i Ax_0, gdje je x_0 zadana početna aproksimacija. Početni korak traganja d_0 je jednak spomenutom rezidualnu, a broj koraka smo inicijalizirali na 0. Nakon toga, while petlja se izvršava dok je $\|\cdot\|_\infty$ veća ili jednaka toleranciji tol. Unutar while petlje koristimo ranije dobivene rezultate objedinjene u Algoritmu 2.21. Potom se ispisuje potreban broj koraka iteracije.

3.5 Usporedba iterativnih metoda

Nakon što smo u prethodnim potpoglavljima ovog poglavlja analizirali implementacije iterativnih metoda unutar programskog paketa MATLAB, usporedimo sada brzinu svake od metoda, prilikom rješavanja sljedećega primjera.

Primjer 3.1. *Neka je dana matrica:* $A = \begin{bmatrix} 5 & 2 & 0 & 0 \\ 2 & 5 & 2 & 0 \\ 0 & 2 & 5 & 2 \\ 0 & 0 & 2 & 5 \end{bmatrix}$, *te vektor* $b = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$.
Za početni vektor $x^{(0)}$ *uzmimo nul-vektor te neka norma reziduala u odnosu na rješenje bude manja od tolerancije* 10^{-5}.

Uočimo da je matrica A strogo dijagonalno dominantna što nam jamči konvergenciju Jacobijeve i Gauss-Seidelove metode. Nadalje, matrica A je također simetrična te pozitivno definitna što je uvjet za metodu najbržeg silaska.
Nakon implementacije unutar programskog paketa MATLAB, u tablici možete vidjeti odnos brzina, odnosno broja koraka potrebnih pojedinoj metodi za rješavanje sustava istog $Ax = b$ iz primjera 3.1.

	Jacobijeva matoda	Gauss-Seidelova metoda	metoda najbržeg silaska	metoda konjugiranih gradijenata
broj koraka	26	14	27	4
spektralni radijus	0.6427	0.4189	-	-

Uočimo da je broj iteracija potrebnih metodi konjugiranih gradijenata za postizanje aproksimacije rješenja uz uvijet da norma reziduala bude manja od zadane tolerancije najmanji u odnosu na ostale metode te iznosi 4. Koristimo li Gauss-Seidelovu metodu za rješavanje istog sustava bit će potrebno 14 iteracija, primjenom Jacobijeve metode 26, te metode najbržeg silaska 27 iteracija.
Brzina konvergencije metode je veća što je spektralni radijus manji te je u našem primjeru

vrijednost spektralnog radijusa manja ukoliko koristimo Gauss-Seidelovu metodu u odnosu na Jacobijevu te ćemo aproksimaciju rješanja pomoću Gauss-Seidelove metode dobiti brže, odnosno potreban je manji broj iteracija.

Literatura

[1] N. Bosner, Iterativne metode za rješavanje linearnih sustava, magistarski rad, PMF, Matematički odjel, Zagreb, 2001.

[2] J.W. Demmel, Applied Numerical Algebra, SIAM, Philadelphia, 1997.

[3] M. Rogina, S. Singer, S. Singer, Numerička analiza, PMF,Matematički odjel, Zagreb, 2003.

[4] Y. Saad, Iterative Methods for Sparse Linear Systems, Second Edition, SIAM, Philadelphia, 2003.

[5] N. Truhar, Numerička linearna algebra, Sveučilište J.J. Strossmayera, Odjel za matematiku, Osijek, 2010.