Gauss-Seidelova metoda za rješavanje sustava linearnih jednadžbi

Dizdarević, Raoul

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni prijediplomski studij elektrotehnike

Završni rad

GAUSS-SEIDELOVA METODA ZA NUMERIČKO RJEŠAVANJE LINEARNIH JEDNADŽBI

Rijeka, 08.09.2024.

Raoul Dizdarević
0069090449

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni prijediplomski studij elektrotehnike

Završni rad

GAUSS-SEIDELOVA METODA ZA NUMERIČKO RJEŠAVANJE LINEARNIH JEDNADŽBI

Mentor: izv. prof. dr. sc. Ivan Dražić

Komentor: doc. dr. sc. Angela Bašić-Šiško

Rijeka, 08.09.2024.

Raoul Dizdarević
0069090449

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 21.03.2024.

Zavod: Zavod za matematiku, fiziku i strane jezike

Predmet: Inženjerska matematika ET

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Raoul Dizdarević (0069090449)**

Studij: Sveučilišni prijediplomski studij elektrotehnike (1030)

Zadatak: **Gauss-Seidelova metoda za numeričko rješavanje linearnih sustava / Gauss-Seidel method for numerical solving of linear systems**

Opis zadatka:

U radu je potrebno definirati sustav linearnih jednadžbi i odgovarajući matrični zapis. Potrebno je iskazati uvjete pod kojima sustav ima jedinstveno rješenje te navesti osnovne metode rješavanja sustava, direktne i iterativne. U radu treba izvesti Gauss-Seidelovu iterativnu metodu za približno rješavanje sustava i navesti uvjete pod kojima metoda konvergira. Metodu je potrebno primijeniti na problemima iz inženjerstva.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Mentor:
izv. prof. dr. sc. Ivan Dražić

Predsjednik povjerenstva za završni ispit:
prof. dr. sc. Dubravko Franković

Komentor:
doc. dr. sc. Angela Bašić-Šiško

IZJAVA

Sukladno članku 7. Stavku 1. Pravilnika o završnom radu, završnom ispitu i završetku sveučilišnih prijediplomskih studija Tehničkog fakulteta Sveučilišta u Rijeci od 4. travnja 2023., izjavljujem da sam samostalno izradio završni rad prema zadatku preuzetom dana 21. ožujka 2024.

Rijeka, 08.09.2024

Raoul Dizdarević

Ovim putem želim se zahvaliti mentoru izv. prof. dr. sc. Ivanu Dražiću i komentorici doc. dr. sc. Angeli Bašić-Šiško koji su uloženim vremenom, trudom, znanjem i strpljenjem pomogli da ovaj rad bude dovršen. Također se zahvaljujem svojoj obitelji, prijateljima i kolegama koji su mi bili podrška tijekom studija.

Sadržaj

1. Uvod

Mnogi matematički modeli za probleme iz inženjerstva i drugih znanosti svode se na rješavanje sustava linearnih jednadžbi, jer se mnoge fizičke i tehničke situacije mogu modelirati na ovaj način. Neki od najčešćih primjera uključuju proračun toka snaga, analizu električnih mreža, termalnu analizu i analizu strujanja fluida. Kompleksniji problemi rezultiraju kompleksnijim modelima, pa su i dobiveni sustavi u modernoj primjeni vrlo velikih dimenzija. Iako postoje direktne metode koje su teoretski vrlo važne, za takve probleme ipak nisu dovoljno efikasne.

U ovom radu bit će definiran sustav linearnih jednadžbi i odgovarajući matrični zapis, kao i uvjeti koji moraju biti zadovoljeni da bi sustav imao jedinstveno rješenje. Također će biti navedene metode kojima se mogu rješavati takvi sustavi: direktne i iterativne metode.

Najpoznatije direktne metode za rješavanje sustava linearnih jednadžbi su LU dekompozicija i Gaussove eliminacije. Ove metode nisu pogodne za sustave velikih dimenzija zbog velikog broja aritmetičkih operacija koje je potrebno provesti kako bi se došlo do točnog rješenja. Zbog toga se direktne metode koriste kod sustava s malim ili srednjim brojem nepoznanica.

Iterativne metode koriste se za rješavanje sustava kada direktne metode nisu dovoljno efikasne. Ideja iterativnih metoda je postepeno smanjivanje pogreške početne aproksimacije kako bi se došlo do rješenja koje je približno jednako točnome. Iteracije ovih metoda zaustavljaju se prema određenom kriteriju ili kada se dođe do aproksimacije koja se smatra dovoljno dobrom. Iterativna metoda kojoj će se posvetiti najviše pažnje u ovom radu je Gauss-Seidelova metoda. Uz Gauss-Seidelovu metodu spomenut će se i Jacobijeva metoda. Glavna razlika između dvije metode je ta da Gauss-Seidelova metoda konvergira brže zbog toga što koristi novije podatke prilikom računanja nove iteracije. 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 doživjele su 1950-ih godina razvojem računalne tehnologije.

Na kraju samog rada, Gauss-Seidelova metoda bit će implementirana u programskom jeziku Python, te će se na taj način riješiti nekoliko primjera iz matematike, kao i jedan primjer iz inženjerstva. Python je izuzetno koristan za rješavanje sustava linearnih jednadžbi zbog kombinacije jednostavnosti i moćnih alata koje nudi. Korištenjem biblioteka NumPy i SciPy lako se mogu implementirati i ostale numeričke metode koje su navedene u ovom radu. Algoritam Gauss-Seidelove metode se može kodirati i u programske jezike kao što su C++, Matlab, Fortrana i slično.

2. Matrice i vektori

U ovom poglavlju ćemo navesti niz definicija vezanih za matrice i vektore koje su preuzete iz [5] i [7].

Shemu brojeva

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \tag{2.1}$$

zovemo **pravokutnom matricom** tipa (m, n). Ako je $m = n$, onda kažemo da je A **kvadratna matrica** reda n. Elementi $a_{i1}, a_{i2}, \ldots, a_{in}$ čine i-ti redak dok elementi $a_{1j}, a_{2j}, \ldots, a_{mj}$ čine j-ti stupac matrice. Vektor matrice predstavlja jedan redak ili stupac te matrice.

Vektor stupc je matrica sa jednim stupcem i m redaka:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \tag{2.2}$$

dok je **vektor redak** matrica sa jednim retkom i n stupaca:

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}. \tag{2.3}$$

Neka je A matrica dimenzija $m \times n$ s elementima a_{ij}. **Podmatricu** B dobijemo na način da izbacimo pojedine retke i stupce iz matrice A tako da nam ostane matrica dimenzija $p \times q$.

Pretpostavimo da imamo matricu A dimenzija 3×3:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{2.4}$$

Podmatrica B se može dobiti, primjerice, izbacivanjem trećeg retka i trećeg stupca:

$$B = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}. \tag{2.5}$$

Sljedeće matrice se mogu definirati samo za kvadratne matrice.

Za kvadratnu matricu $n \times n$, dijagonalni elementi su a_{ii}, gdje je i redni broj od 1 do n. Matrica kojoj su svi elementi vam dijagonale jednaki nuli zove se **dijagonalna matrica**.

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \tag{2.6}$$

Matrica kojoj su svi elementi ispod glavne dijagonale jednaki nuli naziva se **gornjetrokutasta matrica**.

$$U = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \tag{2.7}$$

Matrica kojoj su svi elementi iznad glavne dijagonale jednaki nuli naziva se **donjetrokutasta matrica**.

$$L = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \tag{2.8}$$

Jedinična matrica, ili matrica identiteta, je dijagonalna matrica čiji su dijagonalni elementi jednaki jedinici.

$$I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{2.9}$$

Dijagonalno dominantne matrice

Matrica $A = [a_{ij}]$ reda $n \times n$ biti će dijagonalno dominantna po redcima ako za svaki redak i vrijedi:

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \tag{2.10}$$

To znači da apsolutna vrijednost dijagonalnog elementa mora biti veća ili jednaka sumi svakog elementa u promatranom redu.

Ako za barem jedan redak i vrijedi da je apsolutna vrijednost dijagonalnog elementa strogo veća od sume svih elemenata u tom istom redu:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \tag{2.11}$$

tada kažemo da je matrica strogo dijagonalno dominantna.

Primjer 2.1. *Primjer matrice koja je dijagonalno dominantna i matrice koja nije dijagonalno dominantna.*

- *Dijagonalno dominantna matrica*

Matrica A je dijagonalno dominantna ako za svaki redak vrijedi da je apsolutna vrijednost dijagonalnog elementa veća ili jednaka zbroju apsolutnih vrijednosti svih ostalih elemenata u tom retku:

$$A = \begin{bmatrix} 4 & -1 & 0 \\ 1 & 3 & -1 \\ 2 & -1 & 5 \end{bmatrix} \tag{2.12}$$

Provjera:

$$\begin{aligned}
|a_{11}| = 4, \qquad & \sum_{j \neq 1} |a_{1j}| = |-1| + |0| = 1, \qquad & |a_{11}| > \sum_{j \neq 1} |a_{1j}| \\
|a_{22}| = 3, \qquad & \sum_{j \neq 2} |a_{2j}| = |1| + |-1| = 2, \qquad & |a_{22}| > \sum_{j \neq 2} |a_{2j}| \\
|a_{33}| = 5, \qquad & \sum_{j \neq 3} |a_{3j}| = |2| + |-1| = 3, \qquad & |a_{33}| > \sum_{j \neq 3} |a_{3j}|
\end{aligned} \tag{2.13}$$

- *Matrica koja nije dijagonalno dominantna*

Matrica B nije dijagonalno dominantna jer ne zadovoljava uvjet dijagonalne dominacije:

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \tag{2.14}$$

Provjera:

$$\begin{aligned}
|b_{11}| = 1, \qquad & \sum_{j \neq 1} |b_{1j}| = |2| + |3| = 5, \qquad & |b_{11}| < \sum_{j \neq 1} |b_{1j}| \\
|b_{22}| = 5, \qquad & \sum_{j \neq 2} |b_{2j}| = |4| + |6| = 10, \qquad & |b_{22}| < \sum_{j \neq 2} |b_{2j}| \\
|b_{33}| = 9, \qquad & \sum_{j \neq 3} |b_{3j}| = |7| + |8| = 15, \qquad & |b_{33}| < \sum_{j \neq 3} |b_{3j}|
\end{aligned} \tag{2.15}$$

Linearna zavisnost i nezavisnost vektora

Vektori $v_1, v_2, \ldots, v_n$ u vektorskom prostoru su linearno zavisni ako postoji skup skalara $c_1, c_2, \ldots, c_n$, od kojih je barem jedan različit od nule, takav da vrijedi:

$$c_1v_1 + c_2v_2 + \cdots + c_nv_n = 0. \tag{2.16}$$

Ako je jedini način da se postigne nulti vektor 0 taj da su svi koeficijenti $c_1, c_2, \ldots, c_n$ jednaki nuli, tada su vektori $v_1, v_2, \ldots, v_n$ linearno nezavisni.

Inverzna matrica

Za kvadratnu matricu A reda $n \times n$ matrica A^{-1} je takva da zadovoljava sljedeće uvjete:

$$AA^{-1} = A^{-1}A = I. \tag{2.17}$$

A^{-1} je inverzna matrica od matrice A.

Regularna matrica (također poznata kao invertabilna matrica ili nesingularna matrica) je kvadratna matrica koja ima inverznu matricu. Vrijedi da je matrica A regularna ako i samo ako vrijedi da je $det(A) \neq 0$.

Singularna matrica je kvadratna matrica koja **nema** inverznu matricu. To znači da ne vrijedi jednadžba 2.17. Matrica A je singularna ako i samo ako vrijedi $det(A) = 0$.

Svakoj kvadratnoj matrici pridružen je skalar, njezina determinanta. Taj broj označavamo sa $det(A)$ ili $|A|$. Determinantu ćemo definirati na način da krenemo od matrice prvog reda.

Determinanta za matricu $n = 1$:

$$det(A) = |a_{11}| = a_{11} \tag{2.18}$$

Determinanta ma metricu $n = 2$:

$$det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \tag{2.19}$$

Determinanta za matricu $n = 3$:

$$det(A) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \tag{2.20}$$

Formalno, determinanta za matricu reda $n \times n$, $A = [a_{ij}]$, definira se rekurzivno pomoću kofaktorske ekspanzije:

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} \cdot a_{ij} \cdot \det(M_{ij}) \tag{2.21}$$

Rang matrice, označen kao $rang(A)$, definiran je kao maksimalni broj linearno nezavisnih redaka ili stupaca u toj matrici.

Neka je A matrica dimenzija $m \times n$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}. \tag{2.22}$$

Ako su reci matrice A linearno nezavisni, tada je rang matrice jednak broju redaka. Ako nisu svi reci linearno nezavisni, rang je jednak broju elemenata u najvećem podskupu linearno nezavisnih redaka.

Na primjer, za matricu A koja je dana kao:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \tag{2.23}$$

rang matrice A je 2, jer se iz ove matrice može izvući najveća nesingularna podmatrica dimenzija 2×2:

$$\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \tag{2.24}$$

ili

$$\begin{bmatrix} 1 & 3 \\ 4 & 6 \end{bmatrix} \tag{2.25}$$

Reci ili stupci matrice A nisu svi linearno nezavisni, što znači da rang nije jednak 3.

Svojstvena ili **vlastita** vrijednost kvadratne matrice A je skalar λ koja zadovoljava jednadžbu:

$$Av = \lambda v \tag{2.26}$$

gdje je v nenul vektor koji se naziva svojstveni (vlastiti) vektor.

Simetričnost i pozitivna definitnost matrice

Za kvadratnu matricu A kažemo da je simetrična ako je jednaka svojoj transponiranoj matrici, tj. ako vrijedi $A = A^T$. Transponirana matrica dobiva se zamjenom redaka i stupaca izvorne matrice. U tom slučaju su elementi matrice simetrični u odnosu na glavnu dijagonalu ($a_{ij} = a_{ji}$ za sve i i j).

Kvadratna matrica A je pozitivno definitna ako za svaki nenul vektor x iz $\mathbb{R}^n$ vrijedi $x^T Ax > 0$. Pozitivno definitna matrica je nužno i simetrična.

Vektorska norma

Vektorska norma je funkcija koja vektoru pridružuje nenegativan realan broj (dužinu ili veličinu vektora)

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \tag{2.27}$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{2.28}$$

$$\|x\|_\infty = \max_i |x_i|. \tag{2.29}$$

Vektorsku normu koristit ćemo za mjerenje udaljenosti dvaju vektora, naime, smatramo da su dva vektora x i y "blizu" ako je $||x - y||$ malo, odnosno "daleko" ako je veliko.

Kažemo da niz vektora $(x^n)_{n\in\mathbb{N}} \subset \mathbb{R}^n$ konvergira ka vektoru $x \in \mathbb{R}^n$ ako vrijedi $\lim_{n\to\infty} ||x^n - x|| = 0$, gdje je $|| \cdot ||$ bilo koja vektorska norma.

3. Sustav linearnih jednadžbi

Informacije vezane uz ovo poglavlje uzete su iz literature [2] i [3]. Sustav s m jednadžbi i n nepoznanica, gdje su m i $n \in \mathbb{N}$, zovemo sustav linearnih jednadžbi čiji je opći oblik:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n &= b_m. \end{aligned} \tag{3.1}$$

Skalare a_{ij}, $i = 1, 2, \ldots, m$, j = 1, 2, $\ldots$, n zovemo koeficijentima sustava, skalare b_i, $i = 1, 2$, $\ldots$, m slobodnim članovima, a skalare x_j, $j = 1, 2, \ldots, j$ nepoznanicama.
Sustav (3.1) možemo prikazati i sljedećim matricama:

$$\begin{gathered} \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{mw} & \ldots & a_{mn} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \\ \mathbf{A_p} = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array}\right] \end{gathered} \tag{3.2}$$

koje se redom zovu matrica sustava, matrica nepoznanica, matrica slobodnih članova i proširena matrica sustava. Navedene matrice daju matričnu jednadžbu koja je ekvivalentna sustavu (3.1).

$$Ax = b \tag{3.3}$$

Ako je broj jednadžbi m jednak broju nepoznanica n tada sustav zovemo kvadratnim. U slučaju kada je desna strana sustava sastavljena od samih nula tada sustav zovemo homogenim, a ako je barem jedan slobodni član $b_i \neq 0$, tada sustav zovemo nehomogenim. Pri razmatranju direktnih i iterativnih metoda za rješavanje sustava linearnih jednadžbi biti će korišteni kvadratni sustavi.

3.1. Egzistencija i broj rješenja sustava linearnih jednadžbi

Rješenje sustava linearnih jednadžbi je svaka uređena n-torka koja zadovoljava sustav jednadžbi. Navedenu n-torku možemo prikazati na sljedeći način:

$$(x_1, x_2, ..., x_n) \in \mathbb{R}^n.$$

Vrijedi da uređena n-torka $(x_1, x_2, ..., x_n)$ zadovoljava sustav (3.1) ako i samo ako vektor

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{3.4}$$

zadovoljava jednadžbu (3.3), kažemo da su sustav (3.1) i jednadžba (3.3) ekvivalentni.

Za rješenje sustava linearnih jednadžbi vrijedi sljedeće:

1. Sustav nema rješenja ako je rang matrice sustava A manji od ranga proširene matrice sustava A_p.
2. Sustav ima jedinstveno rješenje ako je rang matrice sustava A jednak rangu proširene matrice sustava A_p i broju nepoznanica n.
3. Sustav ima beskonačno mnogo rješenja ako je rang matrice sustava A jednak rangu proširene matrice sustava A_p i ako je manji od broja nepoznanica n.

Za sustav kažemo da je konzistentan (rješiv) ako ima barem jedno rješenje, u slučaju da sustav nema niti jedno rješenje tada za njega kažemo da je nekonzistentan (nerješiv).
Homogeni sustav je uvijek rješiv tj. ima barem trivijalno rješenje $x_1 = 0$, $x_2 = 0, \ldots, x_n = 0$. Problem nastaje kada želimo pronaći rješenje koje nije trivijalno. U tom slučaju se koriste direktne i iterativne metode za rješavanje sustava linearnih jednadžbi. Odabir metode ovisiti će ponajviše o dimenzijama samog sustava što će biti detaljnije obajašnjeno u nastavku rada. Rješivost kvadratnog sustava možemo karakterizirati i pomoću determinante.

Teorem 3.1. *Sustav ima jedinstveno rješenje ako i samo ako je matrica A regularna.*

4. Direktne metode

U ovom poglavlju koristila se literatura [4]. Uz pomoć direktnih metoda možemo izračunati točno rješenje sustava linearnih jednadžbi u konačnom broju koraka. Ovakve metode su posebno prikladne za sustave s manjim ili srednjim brojem nepoznanica zbog svojstva koje nam govori da je za izračun točnog rješenja potrebno $\mathcal{O}(n^3)$ algebarskih operacija (rješavanje sustava dimenzije n zahtijeva Cn^3 računskih operacija, gdje je C neka konstanta). Problem nastaje kada sustav linearnih jednadžbi ima jako veliki broj nepoznanica (nekoliko desetaka tisuća). U takvom bi slučaju prosječnom računalu trebalo nekoliko dana da dođe do željenog rješenja.

Postoji nekoliko različitih metoda za direktno rješavanje sustava linearnih jednadžbi od kojih su najpoznatije metode Gaussove eliminacije i LU faktorizacija.

4.1. Gaussove eliminacije

Rješavanje sustava Gaussovm eliminacijama se svodi na pretvaranje proširene matrice sustava na gornjetrokutastu matricu primjenom elementarnih transformacija kao što su:

1. Zamjena jednadžbi
2. Množenje jednadžbi konstantom različitom od nule
3. Množenje jednadžbi brojem i dodavanje drugoj jednadžbi

Rješenja ovako transformiranog sustava ostaju ista kao rješenja originalnog sustava bez obzira na promjenu njegovih koeficijenata. Dobiveni gornjetrokutasti sustav lako se rješava takozvanom supstitucijom unatrag, što će detaljnije biti pojašnjeno u idućem poglavlju o LU faktorizaciji.

Primjer 4.1. *Korištenjem Gaussovih eliminacija riješiti sustav* (4.1)

Zadan je sustav jednadžbi:

$$\begin{aligned} 2x + 3y + z &= 1, \\ 4x + y - 2z &= -2, \\ -2x + y + 2z &= 7. \end{aligned} \tag{4.1}$$

Najprije zapisujemo proširenu matricu sustava:

$$\left[\begin{array}{ccc|c} 2 & 3 & 1 & 1 \\ 4 & 1 & -2 & -2 \\ -2 & 1 & 2 & 7 \end{array}\right]. \tag{4.2}$$

U prvom koraku ćemo eliminirati x iz druge jednadžbe na način da prvu jednadžbu pomnoženu s 2 oduzmemo od druge jednadžbe:

$$\left[\begin{array}{ccc|c} 2 & 3 & 1 & 1 \\ 0 & -5 & -4 & -4 \\ -2 & 1 & 2 & 7 \end{array}\right]. \tag{4.3}$$

Zatim zbrajamo prvu i treću jednadžbu kako bismo eliminirali x iz treće jednadžbe:

$$\left[\begin{array}{ccc|c} 2 & 3 & 1 & 1 \\ 0 & -5 & -4 & -4 \\ 0 & 4 & 3 & 8 \end{array}\right]. \tag{4.4}$$

U drugom koraku je cilj eliminirati y iz treće jednadžbe. To možemo postići tako da drugu jednadžbu pomnoženu s $\frac{4}{5}$ dodamo trećoj jednadžbi:

$$\left[\begin{array}{ccc|c} 2 & 3 & 1 & 1 \\ 0 & -5 & -4 & -4 \\ 0 & 0 & -\frac{1}{5} & \frac{24}{5} \end{array}\right]. \tag{4.5}$$

U trećem koraku supstitucijom unatrag dolazimo do željenih rješenja.

Iz treće jednadžbe imamo:

$$-\frac{1}{5}z = \frac{24}{5} \implies z = -24. \tag{4.6}$$

Sada koristimo $z = -24$ u drugoj jednadžbi:

$$-5y - 4(-24) = -4 \implies -5y + 96 = -4 \implies -5y = -100 \implies y = 20. \tag{4.7}$$

Na kraju, koristimo $y = 20$ i $z = -24$ u prvoj jednadžbi:

$$\begin{gathered} 2x + 3(20) + (-24) = 1 \implies 2x + 60 - 24 = 1 \implies \\ 2x + 36 = 1 \implies 2x = -35 \implies x = -\frac{35}{2} \end{gathered} \tag{4.8}$$

Rješenje sustava je:

$$x = -\frac{35}{2}, \quad y = 20, \quad z = -24. \tag{4.9}$$

4.2. LU faktorizacija

Pretpostavimo da matricu sustava možemo napisati kao umnožak donjetrokutaste matrice L koja na glavnoj dijagonali ima jedinice i gornjetrokutaste matrice U:

$$A = LU. \tag{4.10}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix}, U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{bmatrix} \tag{4.11}$$

Množenjem matrica slijede izrazi za elemente l_{ij} i u_{ij}: za $i = 1, 2, \ldots, n$ vrijedi:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, \quad \text{za } i \leq j \tag{4.12}$$

$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right), \quad \text{za } i > j. \tag{4.13}$$

Zadani sustav se tada rješava u dva jednostavna koraka:

1. Supstitucijom unaprijed riješit će se donjetrokutasti sustav $Ly = b$

$$y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j, \quad \text{za } i = 1, 2, \ldots, n \tag{4.14}$$

2. Supstitucijom unatrag riješit će se gornjetrotutasti sustav $Ux = y$

$$x_i = \frac{1}{u_{ii}} \left(y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right), \quad \text{za } i = n, n-1, \ldots, 1 \tag{4.15}$$

Primjer 4.2. *U ovom primjeru dan je detaljan raspis LU faktorizacije za opću* 3×3 *matricu*

Zadan je sustav linearnih jednadžbi:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases} \tag{4.16}$$

čija pripadna matrica sustava i slobodnih elemenata izgleda ovako:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}. \tag{4.17}$$

Matricu A razlažemo ma matrice L i U prema formulama (4.14) *i* (4.15)*:*

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}. \tag{4.18}$$

Formule za elemente matrice L i U glase:

$$u_{11} = a_{11}, u_{12} = a_{12}, u_{13} = a_{13} \tag{4.19}$$

$$\begin{aligned} u_{22} &= a_{22} - l_{21}u_{12} \\ u_{23} &= a_{23} - l_{21}u_{13} \\ u_{33} &= a_{33} - l_{31}u_{13} - l_{32}u_{23} \end{aligned} \tag{4.20}$$

$$l_{21} = \frac{a_{21}}{u_{11}}, l_{31} = \frac{a_{31}}{u_{11}}, l_{32} = \frac{a_{32} - l_{31}u_{12}}{u_{22}}. \tag{4.21}$$

Nakon što smo matricu sustava podijelili na gornje i donjetrokutastu matricu slijedi rješavanje sustava. U prvom koraku rješavamo donjetrokutasti sustav $Ly = b$ supstitucijom unaprijed:

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{4.22}$$

te redom po elementima dobivamo:

$$\begin{aligned} & y_1 = b_1 \\ & l_{21}y_1 + y_2 = b_2 \quad \Rightarrow \quad y_2 = b_2 - l_{21}y_1 \\ & l_{31}y_1 + l_{32}y_2 + y_3 = b_3 \quad \Rightarrow \quad y_3 = b_3 - l_{31}y_1 - l_{32}y_2 \end{aligned} \tag{4.23}$$

U drugom koraku rješavamo gornjetrokutasti sustav $Ux = y$ supstitucijom unatrag:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{4.24}$$

te redom po elementima dobivamo:

$$\begin{aligned} & u_{33}x_3 = y_3 \quad \Rightarrow \quad x_3 = \frac{y_3}{u_{33}} \\ & u_{22}x_2 + u_{23}x_3 = y_2 \quad \Rightarrow \quad x_2 = \frac{y_2 - u_{23}x_3}{u_{22}} \\ & u_{11}x_1 + u_{12}x_2 + u_{13}x_3 = y_1 \quad \Rightarrow \quad x_1 = \frac{y_1 - u_{12}x_2 - u_{13}x_3}{u_{11}} \end{aligned} \tag{4.25}$$

Primjer 4.3. *Korištenjem LU faktorizacije riješiti sustav* (4.1).

Neka je matrica koeficijenata A zapisana kao:

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 4 & 1 & -2 \\ -2 & 1 & 2 \end{bmatrix}. \tag{4.26}$$

Želimo faktorizirati matricu A kao $A = LU$, gdje su L donja trokutasta, a U gornja trokutasta matrica.

U prvom koraku tražimo elemente matrica L i U prema formulama (4.14) *i* (4.15)

Prvi redak matrice U jednak je prvom retku matrice A:

$$u_{11} = 2, \quad u_{12} = 3, \quad u_{13} = 1 \tag{4.27}$$

Elementi prvog stupca matrice L dobivaju se dijeljenjem elemenata prvog stupca matrice A elementom u_{11}:

$$l_{11} = \frac{2}{2} = 1, \quad l_{21} = \frac{4}{2} = 2, \quad l_{31} = \frac{-2}{2} = -1 \tag{4.28}$$

Drugi redak matrice U dobiva se tako da od drugog retka matrice A oduzmemo prvi redak pomnožem s 2:

$$u_{22} = 1 - 2(3) = 1 - 6 = -5, \quad u_{23} = -2 - 2(1) = -2 - 2 = -4 \tag{4.29}$$

Element l_{32} dobija se djeljenjem odgovarajućih elemenata modificirane matrice A elementom u_{22}:

$$l_{32} = \frac{a_{32} - l_{31}u_{12}}{u_{22}} = \frac{1+3}{-5} = -\frac{4}{5} \tag{4.30}$$

Posljednji element matrice U dobije se na sljedeći način:

$$u_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23} \tag{4.31}$$

$$u_{33} = 2 - (-1)1 - (-\frac{4}{5})(-4) = -\frac{1}{5} \tag{4.32}$$

Tako dobijemo:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -\frac{4}{5} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & 3 & 1 \\ 0 & -5 & -4 \\ 0 & 0 & -\frac{1}{5} \end{bmatrix} \tag{4.33}$$

U drugom koraku rješavamo sustav $Ly = b$ supstitucijom unaprijed.

Zapišimo sustav $Ly = b$, gdje je $b = \begin{bmatrix} 1 \\ -2 \\ 7 \end{bmatrix}$:

$$\begin{aligned} y_1 &= 1, \\ 2y_1 + y_2 = -2 &\implies y_2 = -4, \\ -1y_1 - \frac{4}{5}y_2 + y_3 = 7 &\implies y_3 = \frac{24}{5}. \end{aligned} \tag{4.34}$$

Dakle, dobijemo:

$$y = \begin{bmatrix} 1 \\ -4 \\ \frac{24}{5} \end{bmatrix} \tag{4.35}$$

U trećem koraku rješavamo sustav $Ux = y$ supstitucijom unatrag:

Zapišimo sustav $Ux = y$:

$$\begin{aligned} 2x_1 + 3x_2 + x_3 &= 1, \\ -5x_2 - 4x_3 &= -4 \implies x_2 = 20, \\ -\frac{1}{5}x_3 &= \frac{24}{5} \implies x_3 = -24. \end{aligned} \tag{4.36}$$

Zamjenom $x_2 = 20$ i $x_3 = -24$ u prvu jednadžbu:

$$2x_1 + 3(20) + (-24) = 1 \implies 2x_1 + 60 + (-24) = 1 \implies x_1 = -\frac{35}{2} \tag{4.37}$$

Rješenje sustava linearnih jednadžbi je:

$$x_1 = -\frac{35}{2}, \quad x_2 = 20, \quad x_3 = -24. \tag{4.38}$$

Gornjetrokutasta matrica U iz LU faktorizacije odgovara matrici U iz Gaussovih eliminacija, a matrica L se može interpretirati kao matrica elementarnih transformacija iz Gaussovih eliminacija.

Rješavanje sustava linearnih jednadžbi korištenjem LU faktorizacije efikasnije je od Gaussovih eliminacija ako je potrebno rješavati više sustava s istom matricom sustava i različitim desnim stranama budući da faktorizacija ne ovisi o desnoj strani pa se može izračunati samo jednom.

5. Iterativne metode

U ovom poglavlju koristila se literatura [4] i [6]. Potreba za iterativnim metodama polazi iz činjenice da je za rješavanje složenijih sustava direktnom metodom potreban veliki broj aritmetičkih operacija. Dakle, iterativne metode manje su složene od direktnih zbog toga što zahtijevaju manje operacija. Korisne su kod rijetko popunjenih sustava i sustava velikih redova s puno nul-elemenata. One daju samo približna rješenja na način da se postupno smanjuje pogreška početne aprosimacije svakim korakom. Postupak staje zaustavnim kriterijem nakon čega se dobije rješenje koje je dovoljno dobro aproksimirano. Pogreške nastale iterativnom metodom su manje od pogrešaka nastalih zaokrzživanjem zbog spremanja u konačnu memoriju računala uporabom direktnih metoda.

Proučavanjem iterativnih metoda za rješavanje ovog, ali i mnogih drugih problema, bavi se grana matematike koja se zove numerička matematika tj. numerička analiza. Numerička matematika je grana matematike koja se bavi razvojem i analizom algoritama za približno rješavanje matematičkih problema. Ona uključuje metode za rješavanje jednadžbi, integraciju, derivaciju, optimizaciju i druge matematičke operacije kada se to ne može učiniti analitički ili kada je analitičko rješenje nepraktično. Numerička matematika je ključna u primjenama u znanosti i inženjerstvu, gdje se koristi za simulacije, modeliranje i analizu podataka. Sve numeričke metode daju aproksimaciju tj. rješenje problema s nekom greškom.

Iterativne metode za rješavanje sustava linearnih jednadžbi djelimo na dvije osnovne skupine, stacionarne i nestacionarne. U skupinu stacionarnih metoda spadaju:

1. **Jacobijeva metoda**
2. **Gauss - Seidelova metoda**
3. **JOR metoda** (Jacobi overrelaxation)
4. **SOR metoda** (Successive overrelaxation)

Dok u nestacionarne spadaju metoda konjugiranih gradijenata, metoda minimalnog ostatka, itd. U ovom radu će detaljnije biti proučene Jacobijeva i Gauss-Seidelova metoda.

5.1. Ideja iterativnih metoda

Neka je zadani sustav jednadžbi:

$$Ax = b. \tag{5.1}$$

Iterativne metode započinju s inicijalnom aproksimacijom $x^{(0)}$ za rješenje x. Svaka iteracija računa novo rješenje $x^{(k+1)}$ na osnovi prethodnog $x^{(k)}$ prema nekoj funkciji iteracije:

$$x^{(k+1)} = f(x^{(k)}). \tag{5.2}$$

Cilj je definirati funkciju iteracije $f(x)$ na takav način da niz rješenja $x^{(k)}$ konvergira prema točnom rješenju x. Iterativne metode poput Jacobi i Gauss-Seidelove metode postižu ovo iterativno poboljšanje raspodjelom matrice A u dijagonalne, donje trokutaste i gornje trokutaste komponente, što omogućuje izračun iteracija na jednostavan način.

Prednost iterativnih metoda je njihova sposobnost rješavanja velikih sustava korištenjem aproksimacija, što smanjuje potrebne računalne resurse u usporedbi s dorektnim metodama.

Dakle, postupak rješavanja iterativnih metoda glasi:

1. **Početna pretpostavka:** Postavimo početne vrijednosti za sve nepoznanice $x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}$.
2. **Iteracija:** Za svaku jednadžbu i za svaku nepoznanicu x_i, izračunamo novu vrijednost $x_i^{(k+1)}$ prema formuli metode.
3. **Kriterij zaustavljanja:** Iteracije se ponavljaju dok se ne zadovolji neki kriterij zaustavljanja, npr. dok razlika između vrijednosti iz dvije uzastopne iteracije ne postane manja od zadanog praga ϵ (ovo je samo jedan od nekoliko različitih kriterija koji mogu biti zadani):

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \epsilon. \tag{5.3}$$

5.2. Jacobijeva metoda

Za sustav linearnih jednadžbi $Ax = b$, pretpostavimo da A možemo razložiti na $D + L + U$, gdje su D dijagonalna matrica, L donja trokutasta matrica (bez dijagonale) i U gornja trokutasta matrica (bez dijagonale):

$$A = D + L + U. \tag{5.4}$$

Jednadžba se može zapisati kao:

$$(D + L + U)x = b, \tag{5.5}$$

iz čega slijedi:

$$Dx = b - (L + U)x. \tag{5.6}$$

Odavde slijedi iterativni proces:

$$x^{(k+1)} = D^{-1}(b - (L + U)x^{(k)}) \tag{5.7}$$

Opća formula za $x_i^{(k+1)}$ je:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right), \tag{5.8}$$

gdje je $x_i^{(k+1)}$ i-ta komponenta nove iteracije $x^{(k)}$, a $x_j^{(k)}$ je j-ta komponenta prethodne iteracije $x^{(k)}$

Primjer 5.1. *Rješavanje sustava Jacobijevom metodom za matricu* 3×3*.*

***Napomena**: Analogan postupak vrijedi i za matrice većih dimenzija.*

U matričnom obliku to izgleda ovako:

$$\begin{gathered} A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad D = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}, \\ L = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & a_{32} & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} \end{gathered} \tag{5.9}$$

Iz jednadžbe (5.8) *imamo formule za iterativni proces:*

$$x_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - (a_{12}x_2^{(k)} + a_{13}x_3^{(k)}) \right) \tag{5.10}$$

$$x_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - (a_{21}x_1^{(k)} + a_{23}x_3^{(k)}) \right) \tag{5.11}$$

$$x_3^{(k+1)} = \frac{1}{a_{33}} \left(b_3 - (a_{31}x_1^{(k)} + a_{32}x_2^{(k)}) \right) \tag{5.12}$$

Iterativni proces se ponavlja sve dok se ne zadovolji zadani kriterij zaustavljanja.

Primjer 5.2. *Korištenjem Jacobijeve metode izračunati prve dvije iteracije za zadani sustav.*

Pretpostavimo da imamo konkretne vrijednosti:

$$\begin{cases} 4x_1 - x_2 + 2x_3 = 4 \\ -3x_1 + 12x_2 - x_3 = 9 \\ x_1 + x_2 + 5x_3 = 7 \end{cases} \tag{5.13}$$

Matrice sustava i slobodnih elemenata izgledaju ovako:

$$A = \begin{bmatrix} 4 & -1 & 2 \\ -3 & 12 & -1 \\ 1 & 1 & 5 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 9 \\ 7 \end{bmatrix} \tag{5.14}$$

Prva iteracija:

$$x_1^{(1)} = \frac{1}{4}\left(4 - (-1 \cdot 0 + 2 \cdot 0)\right) = 1 \tag{5.15}$$

$$x_2^{(1)} = \frac{1}{12}\left(9 - (-3 \cdot 0 + (-1) \cdot 0)\right) = 0.75 \tag{5.16}$$

$$x_3^{(1)} = \frac{1}{5}\left(7 - (1 \cdot 0 + 1 \cdot 0)\right) = 1.4 \tag{5.17}$$

Druga iteracija:

$$x_1^{(2)} = \frac{1}{4}\left(4 - (-1 \cdot 0.75 + 2 \cdot 1.4)\right) = 0.3125 \tag{5.18}$$

$$x_2^{(2)} = \frac{1}{12}\left(9 - (-3 \cdot 1 + (-1) \cdot 1.4)\right) = 0.8667 \tag{5.19}$$

$$x_3^{(2)} = \frac{1}{5}\left(7 - (1 \cdot 0.3125 + 1 \cdot 0.8667)\right) = 1.1642 \tag{5.20}$$

Komponente nove iteracije ovise isključivo o parametrima sustava i rezultatima prethodne iteracije zbog čega se one mogu računati bilo kojim redoslijedom. To Jacobi metodu čini idealnom za paralelno računanje. Iterativni proces se ponavlja dok ne postignemo zadovoljavajuću točnost tj. dok se ne zadovolji zaustavni kriterij.

5.3. Gauss-Seidelova metoda

Slično kao i kod Jacobijeve metode matricu A dijelimo na matrice L, D i U:

$$A = L + D + U. \tag{5.21}$$

Jednadžba se može zapisati kao:

$$(L + D + U)x = b, \tag{5.22}$$

z čega slijedi:

$$(D + L)x^{(k+1)} = b - Ux^{(k)}. \tag{5.23}$$

Odavde slijedi iterativni proces:

$$x^{(k+1)} = (D + L)^{-1}(b - Ux^{(k)}) \tag{5.24}$$

Opća formula za $x_i^{(k+1)}$ je:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \tag{5.25}$$

gdje je $x_j^{(k+1)}$ j-ta komponenta nove iteracije $x^{(k)}$, a $x_j^{(k)}$ je j-ta komponenta prethodne iteracije $x^{(k)}$.

Primjer 5.3. *Rješavanje sustava Gauss-Seidelovom metodom za matricu* 3×3.

***Napomena**: Analogan postupak vrijedi i za matrice većih dimenzija.*

Matrice sustava su:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & a_{32} & 0 \end{bmatrix},$$
$$D = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} \tag{5.26}$$

Iz prethodnih jednadžbi imamo:

$$x_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - (a_{12}x_2^{(k)} + a_{13}x_3^{(k)}) \right) \tag{5.27}$$

$$x_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - (a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)}) \right) \tag{5.28}$$

$$x_3^{(k+1)} = \frac{1}{a_{33}} \left(b_3 - (a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)}) \right) \tag{5.29}$$

Iterativni proces se ponavlja sve dok se ne zadovolji zadani kriterij zaustavljanja.

Primjer 5.4. *Korištenjem Gauss-Seidelove metode riješiti sustav* (5.30).

Pretpostavimo da imamo konkretne vrijednosti:

$$\begin{cases} 4x_1 - x_2 + 2x_3 = 4 \\ -3x_1 + 12x_2 - x_3 = 9 \\ x_1 + x_2 + 5x_3 = 7 \end{cases} \tag{5.30}$$

Matrice su:

$$A = \begin{bmatrix} 4 & -1 & 2 \\ -3 & 12 & -1 \\ 1 & 1 & 5 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 9 \\ 7 \end{bmatrix} \tag{5.31}$$

Prva iteracija:

$$x_1^{(1)} = \frac{1}{4} (4 - (-1 \cdot 0 + 2 \cdot 0)) = 1 \tag{5.32}$$

$$x_2^{(1)} = \frac{1}{12} (9 - (-3 \cdot 1 - 1 \cdot 0)) = 1 \tag{5.33}$$

$$x_3^{(1)} = \frac{1}{5} (7 - (1 \cdot 1 + 1 \cdot 1)) = 1 \tag{5.34}$$

Druga iteracija:

$$x_1^{(2)} = \frac{1}{4} (4 - (-1 \cdot 1 + 2 \cdot 1)) = 0.75 \tag{5.35}$$

$$x_2^{(2)} = \frac{1}{12}\left(9 - (-3 \cdot 0.75 - 1 \cdot 1)\right) = 0.75 \tag{5.36}$$

$$x_3^{(2)} = \frac{1}{5}\left(7 - (1 \cdot 0.75 + 1 \cdot 0.75)\right) = 1.1 \tag{5.37}$$

Glavna razlika u odnosu na Jacobijevu metodu je ta da kod Gauss-Seidelove metode komponente nove iteracije ovise o trenutnoj i prethodnoj iteraciji. Zbog toga možemo očekivati da će Gauss-Seidelova metoda brže konvergirati od Jacobijeve. Poredak računanja je kod ovakve metode bitan.

5.4. Konvergencija iterativnih metoda

Rješavanje sustava linearnih jednadžbi iterativnim metodama kao što su Jacobi i Gauss-Seidelova metoda ovisi o konvergenciji ovih metoda. Kako bismo razumjeli kada i zašto ove metode konvergiraju, potrebno je detaljnije proučiti pojmove spektralnog polumjera i dijagonalne dominantnosti.

5.4.1. Opći oblik iterativnih metoda

Neka je $Ax = b$ sustav linearnih jednadžbi, gdje je A matrica koeficijenata, x vektor nepoznanica, a b vektor slobodnih članova. Iterativne metode mogu se općenito zapisati kao:

$$x^{(k+1)} = Hx^{(k)} + c \tag{5.38}$$

gdje je H iterativna matrica koja ovisi o vrsti metode, a c je vektor koji također ovisi o metodi i vektoru b.

Konvergencija metode ovisi o spektralnom polumjeru matrice H, kojeg definiramo kao najveću apsolutnu vrijednost vlastitih brojeva te matrice:

$$\rho(H) = \max\left\{|\lambda| : \lambda \text{ je vlastita vrijednost matrice } H\right\}. \tag{5.39}$$

Za iterativne metode vrijedi:

- Ako je $\rho(H) < 1$, iterativna metoda će konvergirati
- Ako je $\rho(H) = 1$, metoda je na granici konvergencije i ovisi o dodatnim uvjetima
- Ako je $\rho(H) > 1$, metoda neće konvergirati

Brzina konvergencije je veća što je $\rho(H)$ manji.

Iz jednadžbe (5.7) vidimo da je iterativna matrica za Jacobi metodu:

$$H_J = D^{-1}(L + U). \tag{5.40}$$

Jacobi metoda konvergira ako je spektralni polumjer matrice H_J manji od 1:

$$\rho(H_J) < 1. \tag{5.41}$$

Konvergencija Jacobi metode se postiže kada je matrica A strogo dijagonalno dominantna.

Iz jednadžbe (5.24) vidimo da je iterativna matrica za Gauss-Seidelovu metodu:

$$H_{GS} = (D + L)^{-1}U. \tag{5.42}$$

Gauss-Seidelova metoda konvergira ako je spektralni polumjer matrice H_{GS} manji od 1:

$$\rho(H_{GS}) < 1. \tag{5.43}$$

5.4.2. Povezanost dijagonalne dominantnosti i pozitivne definitnosti s konvergencijom

U sustavu linearnih jednadžni $Ax = b$, matrica A opisuje odnose između varijabli u sustavu. Kada rješavamo navedeni sustav cilj je da svaka nepoznanica x_i što više ovisi o koeficijentu a_{ii} uz kojeg uz kojeg stoji u i-toj jednadžbi jer to znači da ćemo lakše i brže doći do željenog rješenja. Ako je a_{ii} veći od ostalih koeficijenata u retku, to znači da će utjecaj ostalih varijabli x_j na x_i biti relativno mali. U tom slučaju, možemo iterativno prilagođavati x_i prema rješenju bez značajnog utjecaja grešaka ili utjecaja drugih varijabli, što vodi do stabilnosti i konvergencije. U suprotnom, ako je a_{ij}, za $j \neq i$ usporediv sa a_{ii}, to znači da će varijabla x_j imati velik utjecaj na varijablu x_i što otežava pronalaženje stabilnog rješenja kroz iteracije.

Teorem 5.1. *Jacobijeva i Gauss-Seidelova metoda će konvergirati kada je matrica dijagonalno dominantna.*

Teorem 5.2. *Obje metode konvergiraju kada je matrica simetrična i pozitivno definitna.*

Gauss-Seidelova metoda uvijek brže konvergira od Jacobijeve.

6. Primjer iterativnih metoda u programskom jeziku Python

U ovom poglavlju će biti prikazano kako u programskom jeziku Python možemo koristiti iterativne metode.

6.1. Gauss-Seidelova metoda

Na slici 6.1 prikazan je kod koji implementira Gauss-Seidelovu metodu:

```
import numpy as np

def GaussSeidel(A, b, x_0, eps = 1e-8):
    n = np.size(b)
    x_nova = np.zeros(np.shape(b))
    for iteracija in range(1000):

        for i in range(0, n):
            x_nova[i] = b[i]
            for j in range(0, i):
                    x_nova[i] = x_nova[i] - A[i, j] * x_nova[j]
            for j in range(i+1, n):
                    x_nova[i] = x_nova[i] - A[i, j] * x_0[j]
            x_nova[i] = x_nova[i] / A[i, i]

        print(iteracija+1, '. iteracija:', x_nova)
        stop = np.linalg.norm(x_nova-x_0, ord=np.inf)
        print('Uvjet zaustavljanja:', stop, '\n')
        if(stop < eps):
            print('Gauss_Seidelova metoda je u ', iteracija, ' iteracija pronasla rjesenje.')
            return x_nova

        x_0[:] = x_nova[:]

    print('Nije pronadeno rjesenje u dozvoljenom broju iteracija!')
    return None
```

Slika 6.1. Implementacija Gauss-Seidelove metode u Python-u, izvor:autor

Gornja Python funkcija kao ulazne argumente redom prima:

- matricu A
- vektor desne strane b
- početnu iteraciju x_0
- prag uvjeta zaustavljanja ϵ koji ima pretpostavljenu vrijednost 10^{-8}

Kriterij zaustavljanja je trenutak kada maksimalna apsolutna greška (norma $||\cdot||_\infty$) bude manja od zadanog praga tolerancije ϵ. Funkcija ispisuje aproksimaciju te vrijednosti norme razlike susjednih iteracija (uvjet zaustavljanja) za svaku iteraciju te kao povratnu vrijednost vraća zadnje izračunatu aproksimaciju ako je metoda konvergirala. Inače vraća vrijednost None. Smatramo da metoda nije konvergirala ako je potrebno više od 1000 iteracija da se ispuni uvjet zaustavljanja.

Primjer 6.1. *Sustav Ax = b rješava se Gauss-Seidelovom metodom (A i b su zadani u kodu). Početna aproksimacija je nul-vektor.*

```
A = np.array([[4,1,0,0,0], [1,4,1,0,0], [0,1,4,1,0],
[0,0,1,4,1], [0,0,0,1,4]])
b = [5,6,6,6,5]
print(GaussSeidel(A, b, np.zeros(np.shape(b))))
```

Primijetmo da je matrica A strogo dijagonalno dominantna što garantira konvergenciju. Egzaktno rješenje zadanog sustava je vektor jedinica.

Pokretanje gornjeg koda dobivamo sljedeći ispis:

```
1 . iteracija: [1.25       1.1875     1.203125   1.19921875 0.95019531]
Uvjet zaustavljanja: 1.25

2 . iteracija: [0.953125   0.9609375  0.95996094 1.02246094 0.99438477]
Uvjet zaustavljanja: 0.296875

3 . iteracija: [1.00976562 1.00756836 0.99249268 1.00328064 0.99917984]
Uvjet zaustavljanja: 0.056640625

4 . iteracija: [0.99810791 1.00234985 0.99859238 1.00055695 0.99986076]
Uvjet zaustavljanja: 0.01165771484375

5 . iteracija: [0.99941254 1.00049877 0.99973607 1.00010079 0.9999748 ]
```

```
Uvjet zaustavljanja: 0.0018510818481445312

6 . iteracija: [0.99987531 1.00009716 0.99995051 1.00001867 0.99999533]
Uvjet zaustavljanja: 0.0004627704620361328


7 . iteracija: [0.99997571 1.00001844 0.99999072 1.00000349 0.99999913]
Uvjet zaustavljanja: 0.00010040402412414551

8 . iteracija: [0.99999539 1.00000347 0.99999826 1.00000065 0.99999984]
Uvjet zaustavljanja: 1.9677914679050446e-05

9 . iteracija: [0.99999913 1.00000065 0.99999967 1.00000012 0.99999997]
Uvjet zaustavljanja: 3.7428690120577812e-06

10 . iteracija: [0.99999984 1.00000012 0.99999994 1.00000002 0.99999999]
Uvjet zaustavljanja: 7.051166903693229e-07

11 . iteracija: [0.99999997 1.00000002 0.99999999 1.         1.        ]
Uvjet zaustavljanja: 1.324174263572786e-07

12 . iteracija: [0.99999999 1.         1.         1.         1.        ]
Uvjet zaustavljanja: 2.4841270374054147e-08

13 . iteracija: [1. 1. 1. 1. 1.]
Uvjet zaustavljanja: 4.658550878389178e-09
```

Iz priloženog vidimo da je Gauss-Seidelova metoda nakon 12 iteracija pronašla rješenje.

$$x^{13} = [1.1.1.1.1.]. \tag{6.1}$$

Primjer 6.2. *Sustav Ax = b rješava se Gauss-Seidelovom metodom. Uzmimo sada neku drugu početnu iteraciju, recimo vektor [10, 20, 17, -15, 31].*

```
A = np.array([[4,1,0,0,0], [1,4,1,0,0], [0,1,4,1,0],
[0,0,1,4,1], [0,0,0,1,4]])
b = [5,6,6,6,5]
print(GaussSeidel(A, b, [10,20,17,-15,31]))
```

Egzaktno rješenje zadanog sustava je vektor jedinica.

Ispis:

```
0 . iteracija: [-3.75 -1.8125 5.703125 -7.67578125 3.16894531]
[10, 20, 17, -15, 31]
Uvjet zaustavljanja: 27.8310546875

1 . iteracija: [1.703125 -0.3515625 3.50683594 -0.16894531 1.29223633]
[-3.75, -1.8125, 5.703125, -7.67578125, 3.1689453125]
Uvjet zaustavljanja: 7.5068359375

2 . iteracija: [1.33789062 0.28881836 1.47003174 0.80943298 1.04764175]
[1.703125, -0.3515625, 3.5068359375, -0.1689453125, 1.292236328125]
Uvjet zaustavljanja: 2.03680419921875

3 . iteracija: [1.17779541 0.83804321 1.08813095 0.96605682 1.00848579]
[1.337890625, 0.288818359375, 1.47003173828125,

0.8094329833984375, 1.0476417541503906]
Uvjet zaustavljanja: 0.549224853515625
```

⋮

```
13 . iteracija: [1.00000001 0.99999999 1. 1. 1.]
[1.0000000673136693, 0.9999999495143317, 1.0000000252429382,

0.9999999905338851, 1.0000000023665288]
Uvjet zaustavljanja: 5.4692252282606546e-08

14 . iteracija: [1. 1. 1. 1. 1.]
[1.000000012621417, 0.9999999905339112, 1.0000000047330508,

0.999999998225105, 1.0000000004437237]
Uvjet zaustavljanja: 1.0254894933225955e-08

15 . iteracija: [1. 1. 1. 1. 1.]
[1.000000002366522, 0.9999999982251068, 1.000000000887447,

0.9999999996672074, 1.0000000000831981]
Uvjet zaustavljanja: 1.9227988090619874e-09

Gauss_Seidelova metoda je u  15  iteracija pronasla rjesenje.
```

```
[1. 1. 1. 1. 1.]
```

I u ovom slučaju je metoda pronašla točno rješenje bez obzira što je početna iteracija puno dalje od pravog rješenja. Jedina razlika je u tome što je u ovom primjeru bilo potrabno više iteracija.

Primjer 6.3. *Sustav $Ax = b$ rješava se Gauss-Seidelovom metodom (A i b su zadani u kodu). Početna aproksimacija je nul-vektor.*

```
A = np.array([[1,2,-3,0,1], [2,1,2,-4,5], [-3,2,3,8,-1],
[0,-4,8,5,2], [1,5,-1,2,-3]])
b = [1, 12, 4, 5, -9]
print(GaussSeidel(A, b, np.zeros(np.shape(b))))
```

Primijetimo da matrica A više nije dijagonalno dominantna i da Gauss-Seidelova metoda neće konvergirati iako sustav ima jedinstveno rješenje [1, 1, 1, 1, ,1].

Iteracije:

```
0 . iteracija: [1 10 -4.33333333 15.93333333 32.06666667]
[0. 0. 0. 0. 0.]
Uvjet zaustavljanja: 32.06666666666666

1 . iteracija: [-64.06666667 52.2 -129.33333333 236.86666667 269.66666667]
[ 1. 10. -4.33333333 15.93333333 32.06666667]
Uvjet zaustavljanja: 237.5999999999999

2 . iteracija: [-761.06666667 1391.93333333 -2229.44444444 4573.79111111
5861.54222222]
[-64.06666667 52.2 -129.33333333 236.86666667 269.66666667]
Uvjet zaustavljanja: 5591.875555555556
```

$\vdots$

```
997 . iteracija: [nan nan nan nan nan]
[nan nan nan nan nan]
Uvjet zaustavljanja: nan

998 . iteracija: [nan nan nan nan nan]
[nan nan nan nan nan]
```

```
Uvjet zaustavljanja: nan

999 . iteracija: [nan nan nan nan nan]
[nan nan nan nan nan]
Uvjet zaustavljanja: nan

Nije pronadeno rjesenje u dozvoljenom broju iteracija!
```

Primjer 6.4. *Sustav $Ax = b$ rješava se Gauss-Seidelovom metodom (A i b su zadani u kodu). Uzmite da je početna aproksimacija nul-vektor.*

$$A = \begin{pmatrix} 35 & 32 & 25 & 34 & 33 \\ 32 & 42 & 36 & 44 & 40 \\ 25 & 36 & 43 & 40 & 41 \\ 34 & 44 & 40 & 52 & 48 \\ 33 & 40 & 41 & 48 & 49 \end{pmatrix} \tag{6.2}$$

```
A = np.array([[35,32,25,34,33], [32,42,36,44,40], [25,36,43,40,41],
[34,44,40,52,48], [33,40,41,48,49]])
b = [159,194,185,218,211]
print(GaussSeidel(A, b, np.zeros(np.shape(b))))
```

Egzaktno rješenje zadanog sustava je vektor jedinica.

Iteracije:

```
1 . iteracija: [4.54285714 1.15782313 0.69178927 -0.28986407 0.00658988]
Uvjet zaustavljanja: 4.542857142857143

2 . iteracija: [3.26550972 1.83546902 1.1304593 -0.37158949 0.02667108]
Uvjet zaustavljanja: 1.2773474209410787

3 . iteracija: [2.39306875 2.19067524 1.39718694 -0.32541802 0.05585964]
Uvjet zaustavljanja: 0.8724409760657732

4 . iteracija: [1.8054161 2.33361815 1.54839116 -0.20538952 0.09084086]
Uvjet zaustavljanja: 0.5876526459462177
```

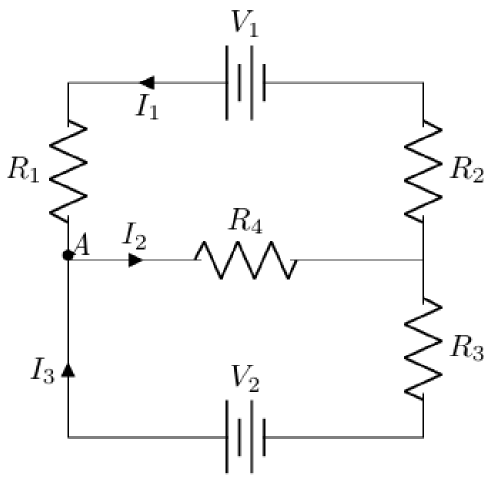
$$\vdots$$

```
435 . iteracija: [1.0000001 0.99999973 1.0000001 1.00000031 0.99999976]
Uvjet zaustavljanja: 1.100962920830284e-08

436 . iteracija: [1.00000009 0.99999974 1.0000001 1.0000003 0.99999977]
Uvjet zaustavljanja: 1.063646992705003e-08

437 . iteracija: [1.00000009 0.99999975 1.0000001 1.00000029 0.99999978]
Uvjet zaustavljanja: 1.0275958750582959e-08


438 . iteracija: [1.00000009 0.99999975 1.00000009 1.00000028 0.99999979]
Uvjet zaustavljanja: 9.927667354858727e-09

Gauss_Seidelova metoda je u 437 iteracija pronasla rjesenje.
[1.00000009 0.99999975 1.00000009 1.00000028 0.99999979]
```

Primijetimo da matrica (6.2) *nije dijagonalno dominantna, ali Gauss-Seidelova metoda ipak konvergia. To je zbog toga što je matrica ovog sustava pozitivno definitna.*

U sljedećem primjeru primjenom Gauss-Seidelove metode riješit ćemo jedan primjer iz elektrotehnike.

Primjer 6.5. *Odrediti struje u strujnom krugu prikazanom na slici 6.2 prikazanom strujnom krugu koristeći Gauss-Seidelovu metodu ako je R1 = R3 = 1Ω, R2 = 2Ω, R4 = 4Ω, V1 = 15V i V2 = 14V.*



Slika 6.2. Strujni krug, izvor [1]

Gore:

$$R_1 I_1 + R_4 I_2 + R_2 I_1 = V_1 \tag{6.3}$$

Dolje:

$$R_4 I_2 + R_3 I_3 = V_2 \tag{6.4}$$

Točka A:

$$I_1 + I_3 = I_2 \tag{6.5}$$

Sustav:

$$\begin{aligned} 3I_1 + 4I_2 &= 15 \\ 4I_2 + I_3 &= 14 \\ I_1 - I_2 + I_3 &= 0 \end{aligned} \tag{6.6}$$

Pozivanje funkcije:

```
A = np.array([[3,4,0], [0,4,1], [1,-1,1]])
b = [15,14,0]
print(GaussSeidel(A, b, np.zeros(np.shape(b))))
```

Ispis:

```
1 . iteracija: [5. 3.5 -1.5]
Uvjet zaustavljanja: 5.0

2 . iteracija: [0.33333333 3.875 3.54166667]
Uvjet zaustavljanja: 5.041666666666666

3 . iteracija: [-0.16666667 2.61458333 2.78125]
Uvjet zaustavljanja: 1.2604166666666665
```

$$\vdots$$

```
36 . iteracija: [1.00000002 3. 1.99999998]
Uvjet zaustavljanja: 2.869879922595686e-08

37 . iteracija: [1. 3. 2.00000001]
Uvjet zaustavljanja: 2.2238351826686653e-08

38 . iteracija: [0.99999999 3. 2.]
Uvjet zaustavljanja: 5.559587901160512e-09

Gauss_Seidelova metoda je u  37  iteracija pronasla rjesenje.
[0.99999999 3. 2.]
```

Dakle, struje u ovom strujnom krugu iznose $I_1 = 1, I_2 = 3$ i $I_3 = 2$.

6.2. Jacobijeva metoda

Implementaciju Jacobijeve metode možemo dobiti tako da u 9. liniji koda Gauss-Seidelove metode zamijenimo `A[ij]*x_nova[j]` s `A[ij]*x_0[j]`.

Primjer 6.6. *Sustav Ax = b rješava se Jacobijevom metodom (A i b su isti kao i kod prvog primjera za Gauss-Seidelovu metodu). Uzmite da je početna aproksimacija nul-vektor*

```
A = np.array([[4,1,0,0,0], [1,4,1,0,0], [0,1,4,1,0],
[0,0,1,4,1], [0,0,0,1,4]])
b = [5,6,6,6,5]
print(Jacobi(A, b, np.zeros(np.shape(b))))
```

Ispis:

```
0 . iteracija: [1.25 1.5  1.5  1.5  1.25]
Uvjet zaustavljanja: 1.5

1 . iteracija: [0.875 0.8125 0.75 0.8125 0.875]
Uvjet zaustavljanja: 0.75

2 . iteracija: [1.046875 1.09375 1.09375 1.09375 1.046875]
Uvjet zaustavljanja: 0.34375

3 . iteracija: [0.9765625 0.96484375 0.953125 0.96484375 0.9765625]
Uvjet zaustavljanja: 0.140625
```

⋮

```
21 . iteracija: [0.99999999 0.99999999 0.99999999 0.99999999 0.99999999]
Uvjet zaustavljanja: 4.027856448374223e-08

22 . iteracija: [1. 1.00000001 1.00000001 1.00000001 1.]
Uvjet zaustavljanja: 1.846100872171519e-08

23 . iteracija: [1. 1. 1. 1. 1.]
Uvjet zaustavljanja: 7.552230840701668e-09

Jacobijeva metoda je u  23  iteracija pronasla rjesenje.
```

```
[1. 1. 1. 1. 1.]
```

Jacobijevoj metodi treba više iteracija za isti zadatak u odnosu na Gauss-Seidelovu metodu.

7. Zaključak

Direktne i iterativne metode za rješavanje sustava linearnih jednadžbi ključan su alat u mnogim znanstvenim, matematičkim i inženjerskim disciplinama jer znatno olakšavaju i povećavaju efikasnost rješavanja raznovrsnih problema.

Direktne metode koriste se za rješavanje sustava u konačnom broju koraka. Bitne su zbog svoje točnosti i pouzdanosti te su idealne za sustave s manjim brojem nepoznanica. Iterativne metode često su jedini praktični izbor za rješavanje sustava velikih dimenzija zbog velikog broja računskih operacija koje zahtijevaju direktne metode. Iterativne metode mogu se prilagoditi specifičnostima problema, poput rijetkih matrica (gdje je većina elemenata nula), koje se često javljaju u praksi. Rješenje dobiveno iterativnim metodama uglavnom je točnije u usporedbi s rješenjem dobivenim korištenjem direktnih metoda.

Gauss-Seidelova metoda se, u usporedbi s Jacobi metodom, pokazala efikasnijom za rješavanje velikih sustava zbog veće brzine konvergencije. To je zato što Gauss-Seidelova metoda koristi komponente trenutne i prethodne iteracije za računanje novih komponenata. Također, Gauss-Seidelova metoda zahtijeva manje memorije jer ne treba pohranjivati cijeli sustav.

Kao što je već navedeno u uvodu rada, Gauss-Seidelova metoda se jednostavno može implementirati u razne programske jezike. U ovom radu je dan primjer kako se, koristeći Gauss-Seidelovu metodu u programskom jeziku Python, može doći do vrijednsoti struja u strujnom krugu.

Literatura

[1] Bašić-Šiško A. "Numerička matematika - skripta", Sveučilište u Rijeci, Tehnički fakultet, 2024.

[2] Ninoslav Truhar, "Numerička linearna algebra", Sveučilište J.J. Strossmayera, Odjel za matematiku, Osijek, 2010.

[3] Brzić, L. (2020). Jacobijeva i Gauss-Seidelova metoda za rješavanje sustava linearnih jednadžbi (Završni rad). Osijek: Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku.

[4] Rudolf Scitovski, "Numerička matematika", Sveučilište J.J. Strossmayera, Odjel za Matematiku, Osijek, 2015.

[5] Matrice, s interneta https://element.hr/wp-content/uploads/2020/06/unutra-13520.pdf (preuzeto 04.09.2024)

[6] Yousef Saad, "Iterative Methods for Sparse Linear Systems", Second edition, SIAM, Philadephia, 2003.

[7] D. Bakić, Linearna algebra, PMF, Matematički odjel, Zagreb, 2008.

Sažetak i ključne riječi

U ovom završnom radu definiran je sustav linearnih jednadžbi i odgovarajući matrični zapis. Opisane su metode za rješavanje takvih sustava: direktne i iterativne metode. Direktne metode daju egzaktno rješenje u konačnom broju koraka i idealne su za rješavanje sustava malih i srednjih veličina. Iterativne metode se koriste kada direktne metode nisu dovoljno efikasne. Idealne su za sustave s velikim brojem nepoznanica i daju aproksimirana rješenja, tj. rješenja s određenom pogreškom. Ovaj rad će najviše pažnje posvetiti Gauss-Seidelovoj metodi, uz pomoć koje će na kraju rada biti riješeno nekoliko primjera iz matematike i inženjerstva koristeći programski jezik Python.

Ključne riječi: sustavi linearnih jednadžbi, numerička metoda, iterativne metode, Gauss-Seidelova metoda

Summary and key words

In this thesis, a system of linear equations and the corresponding matrix notation are defined. Methods for solving such systems, both direct and iterative, are described. Direct methods provide an exact solution in a finite number of steps and are ideal for solving small to medium-sized systems. Iterative methods are used when direct methods are not sufficiently efficient. They are ideal for systems with a large number of unknowns and provide approximate solutions, i.e., solutions with a certain error margin. This thesis will focus primarily on the Gauss-Seidel method, which will be used at the end of the thesis to solve several examples from mathematics and engineering using the Python programming language.

Keywords: systems of linear equations, numerical method, iterative methods, Gauss-Seidel method