Sveučilište J.J. Strossmayera,
Odjel za matematiku

Numerička linearna algebra

Ninoslav Truhar

Osijek, 2010.

Dr.sc. Ninoslav Truhar, redoviti profesor

Odjel za matematiku
Sveučilište J. J. Strossmayera u Osijeku
Trg Ljudevita Gaja 6
HR-31000 Osijek

e-mail: ntruhar@math.hr

Izdavač:
Sveučilište J. J. Strossmayera u Osijeku, Odjel za matematiku

Recenzenti:
dr.sc. Rudolf Scitovski, redoviti profesor u trajnom zvanju,
Odjel za matematiku, Sveučilište u Osijeku
dr.sc. Dražen Adamović, redoviti profesor, PMF - Matematički odjel,
Sveučilište u Zagrebu

Lektor:
Ivanka Ferčec, Elektrotehnički fakultet, Osijek

CIP zapis dostupan u računalnom katalogu Gradske i sveučilišne knjižnice Osijek pod brojem 121125049

ISBN 978-953-6931-42-2

Ovaj udžbenik objavljuje se uz suglasnost Senata Sveučilišta J. J. Strossmayera u Osijeku pod brojem 34/10.

Tisak:
Grafika d.o.o.
Osijek

Sadržaj

Poglavlje 1

Uvod

Ovaj tekst je prije svega namijenjen studentima Odjela za matematiku Sveučilišta J. J. Strossmayera u Osijeku za potrebe predmeta *Numerička linearna algebra* **M033** koji se sluša u 5. semestru preddiplomskog studija.

Što je **numerička linearna algebra**? Naravno, osnova je linearna algebra, tj. u okviru ovog predmeta (odnosno matematičkog područja) proučavaju se isti problemi kao i u linearnoj algebri, ali ovdje s potpuno drugačijeg aspekta.

Nas će prije svega zanimati:

- rješavanje velikih problema pomoću računala, te
- proučavanje brzine, točnosti i stabilnosti pojedinih algoritama.

Veliki se problemi često pojavljuju u primjenama, npr. pri diskretizaciji nekog kontinuiranog problema, kod analize slike ili kod obrade velikih skupova podataka, npr. konstrukcije kvalitetnih web pretraživača. U okviru našeg predavanja mi ćemo se većinom baviti sa sljedeća tri glavna problema.

SLJ (sustav linearnih jednadžbi): za zadanu matricu $A \in \mathbb{C}^{m\times n}$ i vektor $b \in \mathbb{C}^m$ treba odrediti vektor $x \in \mathbb{C}^n$ takav da je

$$Ax = b.$$

LPNK (linearni problem najmanjih kvadrata): za zadanu matricu $A \in \mathbb{C}^{m\times n}$ i vektor $b \in \mathbb{C}^m (m \geq n)$ treba odrediti $x \in \mathbb{C}^n$ koji minimizira (udaljenost) normu

$$\|Ax - b\|.$$

PSV (problem svojstvenih vrijednosti, (svojstveni problem)): za zadanu matricu $A \in \mathbb{C}^{n\times n}$ treba odrediti vektor $x \in \mathbb{C}^n, \lambda \in \mathbb{C}$ takav da

$$Ax = \lambda x \text{ za } x \neq 0.$$

Predavanja su poredana na sljedeći način:

Poglavlje 1 (Linearna algebra i MatLab) sadrži kratki pregled osnovnih pojmova linearne algebre i daje osnovne teorijske rezultate koji će nam biti neophodni u kasnijim analizama. Također dajemo kratki uvod u programski paket MatLab koji će nam služiti kao osnovno *oruđe* s kojim ćemo provjeravati kvalitetu proučavanih algoritama.

Poglavlje 2 (Složenost algoritama) proučava osnovne načine pomoću kojih mjerimo kvalitetu izvođenja algoritama, što ćemo ilustrirati na nekoliko jednostavnih primjera.

Poglavlje 3 (Stabilnost i uvjetovanost) proučava osjetljivost rješenja problema linearne algebre u ovisnosti o malim promjenama (perturbacijama) veličina koje definiraju te probleme, npr. kako rješenje ovisi o zaokruživanju brojeva pri rješavanju (pogreške zaokruživanja).

Poglavlje 4 (SLJ) uspoređuje nekoliko algoritama za rješavanje sustava linearnih jednadžbi. Istovremeno ćemo proučavati stabilnost i složenost tih algoritama.

Poglavlje 5 (LPNK) proučava osnove linearnih problema najmanjih kvadrata i uspoređuje nekoliko algoritama za njihovo rješavanje.

Poglavlje 6 (Iterativne metode) proučava drugačiju skupinu algoritama za rješavanje sustava linearnih jednadžbi: to su iterativne metode koje daju približno rješenje promatranog sustava, ali mogu biti znatno brže, stabilnije te bolje mogu koristiti moguću strukturu problema u odnosu na direktne metode proučavane u Poglavlju 4.

Poglavlje 7 (PSV) bavi se algoritmima za računanje svojstvenih vrijednosti i pripadnih svojstvenih vektora.

Iako ćemo proučavati primjere koji neće prelaziti dimenzije od nekoliko tisuća, sve navedene metode su konstruirane tako da se mogu primijeniti na problemima čija je dimenzija nekoliko desetaka tisuća.

Zahvale

Želio bih se zahvaliti kolegama i studentima koji su savjetima, diskusijom, ili strpljivim čitanjem ovog teksta pridonijeli njegovom poboljšanju. Posebno

se zahvaljujem kolegi Zoranu Tomljanoviću na izradi zadataka u tekstu.

Poglavlje 2

Linearna algebra i MatLab

U ovom poglavlju nalazi se pregled osnovnih rezultata iz linearne algebre koji će nam služiti za bolje razumijevanje teorije koju ćemo koristiti u našim analizama.

2.1 Vektorske norme i skalarni umnožak

Definicija 2.1 *Vektorska norma* $\|\cdot\|$ *na* $\mathbb{C}^n$ *je preslikavanje* $\|\cdot\| : \mathbb{C}^n \to \mathbb{R}$ *koje zadovoljava:*

a) $\|x\| \geq 0$ *za sve* $x \in \mathbb{C}^n$ *i* $\|x\| = 0$ *ako i samo ako* $x = 0$,

b) $\|\alpha x\| = |\alpha|\|x\|$ *za sve* $\alpha \in \mathbb{C}$, $x \in \mathbb{C}^n$, *i*

c) $\|x + y\| \leq \|x\| + \|y\|$ *za sve* $x, y \in \mathbb{C}^n$.

Primjer:

- p-norma za $1 \leq p < \infty$:

$$\|x\|_p = \Big(\sum_{j=1}^{n} |x_j|^p \Big)^{1/p} \qquad \forall x \in \mathbb{C}^n$$

- za p=2 imamo euklidsku normu:

$$\|x\|_2 = \sqrt{\sum_{j=1}^{n} |x_j|^2} \qquad \forall x \in \mathbb{C}^n$$

- za p=1 dobivamo

$$\|x\|_1 = \sum_{j=1}^{n} |x_j| \qquad \forall x \in \mathbb{C}^n$$

- norma L_∞ : $\|x\|_\infty = \max\limits_{1\le j\le n} |x_j|$

Definicija 2.2 *Skalarni umnožak na* $\mathbb{C}^n$ *je preslikavanje* $\langle\cdot,\cdot\rangle : \mathbb{C}^n \times \mathbb{C}^n \longrightarrow \mathbb{C}$ *koje zadovoljava:*

a) $\langle x,\, x\rangle \in \mathbb{R}^+$ *za sve* $x \in \mathbb{C}^n$ *i* $\langle x,\, x\rangle = 0$ *ako i samo ako je* $x = 0$.

b) $\langle x,\, y\rangle = \overline{\langle y,\, x\rangle}$ *za sve* $x,\, y \in \mathbb{C}^n$.

c) $\langle x,\, \alpha\, y\rangle = \alpha\, \langle x,\, y\rangle$ *za sve* $\alpha \in \mathbb{C}$, $x,\, y \in \mathbb{C}^n$.

d) $\langle x,\, y + z\rangle = \langle x,\, y\rangle + \langle x,\, z\rangle$ *za sve* $x,\, y,\, z \in \mathbb{C}^n$.

Primjer 2.1 *Standardni skalarni umnožak na* $\mathbb{C}^n$ *dan je sa*

$$\langle x,\, y\rangle = \sum_{j=1}^{n} \overline{x}_j\, y_j, \tag{2.1}$$

gdje su $x, y \in \mathbb{C}^{\ltimes}$, $x = (x_1, \ldots, x_n)$,$y = (y_1, \ldots, y_n)$.

Napomena 2.1 *Uvjeti c) i d) definicije 2.2 ukazuju da je skalarni umnožak* $\langle\cdot,\cdot\rangle$ *linearna funkcija u drugoj komponenti. To se može jednostavno provjeriti. S druge strane, koristeći svojstva skalarnog umnoška vidimo da*

$$\langle x+y, z\rangle = \overline{\langle z, x+y\rangle} = \overline{\langle z, x\rangle + \langle z, y\rangle} = \langle x, z\rangle + \langle y, z\rangle \qquad za\ sve \quad x, y, z \in \mathbb{C}^n\,,$$

i

$$\langle \alpha\, x, y\rangle = \overline{\alpha}\langle x, y\rangle \qquad za\ sve \quad \alpha \in \mathbb{C},\, x, y \in \mathbb{C}^n.$$

Vidimo da je skalarni umnožak anti-linearan s obzirom na prvu komponentu.

Definicija 2.3 *Dva vektora* x, y *su okomita (*ortogonalna*) s obzirom na skalarni umnožak* $\langle\cdot,\cdot\rangle$ *ako je* $\langle x,\, y\rangle = 0$.

Lema 2.1 *Neka je* $\langle \cdot, \cdot \rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ *skalarni umnožak. Tada je preslikavanje* $\| \cdot \| : \mathbb{C}^n \to \mathbb{R}$ *definirano sa*

$$\|x\| = \sqrt{\langle x, x \rangle} \qquad \forall x \in \mathbb{C}^n$$

vektorska norma.

Dokaz. Trebamo provjeriti zadovoljava li preslikavanje $\| \cdot \| : \mathbb{C}^n \to \mathbb{R}$ uvjete $a) - c)$ definicije 2.1.

a) Budući da je $\langle \cdot, \cdot \rangle$ skalarni umnožak, imamo $\langle x, x \rangle \geq 0$ za sve $x \in \mathbb{C}^n$, što znači da je $\sqrt{\langle x, x \rangle}$ dobro definirano i nenegativno. Posebno imamo

$$\|x\| = 0 \iff \langle x, x \rangle = 0 \iff x = 0$$

b) Neka su $\alpha \in \mathbb{C}$ i $x \in \mathbb{C}^n$. Imamo

$$\|\alpha x\| = \sqrt{\langle \alpha\, x,\, \alpha\, x \rangle} = \sqrt{\overline{\alpha}\, \alpha \langle x,\, x \rangle} = |\alpha| \cdot \|x\|$$

c) Za provjeru trećeg svojstva iz definicije 2.1 koristit ćemo Cauchy-Schwarzovu nejednakost

$$|\langle x, y \rangle| \leq \|x\| \|y\| \qquad \forall x, y \in \mathbb{C}^n,$$

koja slijedi iz svojstava skalarnog umnoška bez pretpostavke o tome je li $\| \cdot \|$ norma ili ne. Imamo

$$\begin{aligned} \|x + y\|^2 &= \langle x + y, x + y \rangle \\ &= \langle x, x \rangle + \langle x, y \rangle + \langle y, x \rangle + \langle y, y \rangle \\ &\leq \|x\|^2 + 2|\langle x, y \rangle| + \|y\|^2 \\ &\leq \|x\|^2 + 2\|x\|\|y\| + \|y\|^2 \\ &= (\|x\| + \|y\|)^2 \qquad \forall x, y \in \mathbb{C}^n. \end{aligned}$$

Time smo dokazali lemu. □

Matricu $A \in \mathbb{C}^{m \times n}$ zapisujemo kao

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} = (a_{ij})_{ij}$$

Definicija 2.4 *Neka je dana matrica $A \in \mathbb{C}^{m\times n}$. Pripadnu adjungiranu matricu $A^* \in \mathbb{C}^{n\times m}$ definiramo s $A^* = (\overline{a_{ji}})_{ij}$. (Ako je A realna matrica, tj. za $A \in \mathbb{R}^{m\times n}$ imamo $A^* = A^T$.)*

Koristeći gornju definiciju, standardni skalarni umnožak možemo pisati kao

$$\langle x, y\rangle = x^* y$$

Definicija 2.5 *Za matricu $Q \in \mathbb{C}^{n\times n}$ reći ćemo da je unitarna ako je $Q^*Q = I$, to jest ako su stupci matrice Q ortonormirani s obzirom na standardni skalarni umnožak. (Ako $Q \in \mathbb{R}^{n\times n}$ zadovoljava $Q^TQ = I$, kažemo da je matrica Q ortogonalna.)*

Matrica $A \in \mathbb{C}^{n\times n}$ je *hermitska* ako vrijedi $A^* = A$ (za realne matrice koristimo izraz *simetrična*).

Hermitska matrica $A \in \mathbb{C}^{n\times n}$ je *pozitivno definitna* ako vrijedi da je $x^*Ax > 0$ za sve $x \in \mathbb{C}^n \setminus \{0\}$ (*pozitivno semidefinitna* ako je za $x^*Ax \geq 0$ za sve $x \in \mathbb{C}^n$).

Napomena 2.2

1) *U većem dijelu teksta (ako drugačije ne bude istaknuto) oznaka $\langle\cdot,\cdot\rangle$ predstavljat će standardni skalarni umnožak iz definicije 2.2. Primijetimo da standardni skalarni umnožak zadovoljava*

$$\langle Ax, y\rangle = (Ax)^* y = x^* A^* y = \langle x, A^* y\rangle$$

za sve $x, y \in \mathbb{C}^n$.

2) *Ako je $A \in \mathbb{C}^{n\times n}$ hermitska i pozitivno definitna matrica, tada izraz*

$$\langle x,\, y\rangle_A = \langle x,\, Ay\rangle \qquad \forall x, y \in \mathbb{C}^n$$

definira skalarni umnožak, a izraz

$$\|x\|_A = \sqrt{\langle x,\, x\rangle_A} \qquad \forall x \in \mathbb{C}^n$$

definira vektorsku normu na $\mathbb{C}^n$.

2.1.1 Zadaci

1. Neka je $A \in \mathbb{C}^{n\times n}$ hermitska i pozitivno definitna matrica. Pokažite da tada izraz
$$\langle x, y\rangle_A = \langle x, Ay\rangle$$
definira skalarni umnožak.

2. Pokažite da je za hermitsku pozitivno definitnu matricu $A \in \mathbb{C}^{n\times n}$ izrazom
$$\|x\|_A = \sqrt{\langle x,\, x\rangle_A} \qquad \forall x \in \mathbb{C}^n$$
definirana vektorska norma na $\mathbb{C}^n$.

2.2 Matrične norme

Definicija 2.6 *Matrična norma na $\mathbb{C}^{n\times n}$ je preslikavanje $\|\cdot\| : \mathbb{C}^{n\times n} \to \mathbb{R}$ za koje vrijedi:*

a) $\|A\| \geq 0$ *za sve* $A \in \mathbb{C}^{n\times n}$ *i* $\|A\| = 0$ *ako i samo ako je* $A = 0$

b) $\|\alpha\, A\| = |\alpha|\, \|A\|$ *za sve* $\alpha \in \mathbb{C}, A \in \mathbb{C}^{n\times n}$

c) $\|A + B\| \leq \|A\| + \|B\|$ *za sve* $A, B \in \mathbb{C}^{n\times n}$

d) $\|AB\| \leq \|A\|\, \|B\|$ *za sve* $A, B \in \mathbb{C}^{n\times n}$.

Napomena 2.3 *Uvjeti a), b) i c) pokazuju da je $\|\cdot\|$ vektorska norma na vektorskom prostoru $\mathbb{C}^{n\times n}$ ($\mathbb{C}^{n\times n}$ je vektorski prostor kvadratnih matrica). Međutim, uvjet d) ima smisla jedino za matrice, jer općenito množenje vektora nije definirano (vektorski prostori nisu općenito snabdjeveni vektorskim umnoškom).*

Definicija 2.7 *Neka je zadana vektorska norma $\|\cdot\|_v$ na $\mathbb{C}^n$. Inducirana norma $\|\cdot\|$ na $\mathbb{C}^{n\times n}$ definira se sa*

$$\|A\|_m = \max_{x\neq 0} \frac{\|Ax\|_v}{\|x\|_v} = \max_{\|y\|_v=1} \|Ay\|_v\,,$$

za sve $A \in \mathbb{C}^{n\times n}$.

Teorem 2.1 *Inducirana norma* $\|\cdot\|_m$ *vektorskom normom* $\|\cdot\|_v$ *je matrična norma za koju vrijedi*

$$\|I\|_m = 1$$

i

$$\|Ax\|_v \leq \|A\|_m \|x\|_v$$

za sve $A \in \mathbb{C}^{n\times n}$ *i* $x \in \mathbb{C}^n$.

Dokaz. a) Očito je iz definicije da je $\|A\|_m \in \mathbb{R}$ za sve $A \in \mathbb{C}^{n\times n}$. Iz definicije također slijedi

$$\begin{aligned} \|A\|_m = 0 &\iff \frac{\|Ax\|_v}{\|x\|_v} = 0 \qquad \forall x \neq 0 \\ &\iff \|Ax\|_v = 0 \qquad \forall x \neq 0 \iff Ax = 0 \ \forall x \neq 0 \iff A = 0. \end{aligned}$$

b) Za $\alpha \in \mathbb{C}$ i $A \in \mathbb{C}^{n\times n}$ imamo

$$\|\alpha A\|_m = \max_{x\neq 0} \frac{\|\alpha Ax\|_v}{\|x\|_v} = \max_{x\neq 0} \frac{|\alpha| \, \|Ax\|_v}{\|x\|_v} = |\alpha| \, \|A\|_m$$

c) Za $A, B \in \mathbb{C}^{n\times n}$ vrijedi

$$\begin{aligned} \|A+B\|_m &= \max_{x\neq 0} \frac{\|Ax+Bx\|_v}{\|x\|_v} \\ &\leq \max_{x\neq 0} \frac{\|Ax\|_v + \|Bx\|_v}{\|x\|_v} \\ &\leq \max_{x\neq 0} \frac{\|Ax\|_v}{\|x\|_v} + \max_{x\neq 0} \frac{\|Bx\|_v}{\|x\|_v} = \|A\|_m + \|B\|_m . \end{aligned}$$

Prije nego provjerimo vrijedi li uvjet d) iz definicije 2.6, primijetimo

$$\|I\| = \max_{x\neq 0} \frac{\|Ix\|_v}{\|x\|_v} = \max_{x\neq 0} \frac{\|x\|_v}{\|x\|_v} = 1$$

i

$$\|A\|_m = \max_{y\neq 0} \frac{\|Ay\|_v}{\|y\|_v} \geq \frac{\|Ax\|_v}{\|x\|_v} \quad \forall x \in \mathbb{C}^n \backslash \{0\}$$

što daje

$$\|Ax\|_v \leq \|A\|_m \|x\|_v \quad \forall x \in \mathbb{C}^n.$$

d) Koristeći gornju nejednakost sada možemo pisati

$$\begin{aligned} \|AB\|_m &= \max_{x\neq 0} \frac{\|ABx\|_v}{\|x\|_v} \\ &\leq \max_{x\neq 0} \frac{\|A\|_m \|Bx\|_v}{\|x\|_v} = \|A\|_m \|B\|_m . \end{aligned}$$

□

Uobičajeno je koristiti istu oznaku za induciranu normu i za vektorsku normu. U ovom ćemo se tekstu držati te konvencije.

Prisjetite se da za vektor $x \in \mathbb{C}^n$ kažemo da je *svojstveni vektor* matrice $A \in \mathbb{C}^{n\times n}$, kome pripada *svojstvena vrijednost* $\lambda \in \mathbb{C}$, ako vrijedi

$$Ax = \lambda x , \qquad x \neq 0. \tag{2.2}$$

Definicija 2.8 Spektralni radijus *matrice* $A \in \mathbb{C}^{n\times n}$ *definiramo sa*

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

Teorem 2.2 *Za bilo koju matričnu normu* $\|\cdot\|$, *bilo koju matricu* $A \in \mathbb{C}^{n\times n}$ *i bilo koji* $\ell \in \mathbb{N}$ *vrijedi*

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

Dokaz. Iz definicije spektralnog radijusa $\rho(A)$ vidimo da možemo naći svojstveni vektor x za koji vrijedi $Ax = \lambda\, x$ i $\rho(A) = |\lambda|$. Neka je $X \in \mathbb{C}^{n\times n}$ matrica kojoj su svi stupci jednaki vektoru x. Tada imamo $A^l X = \lambda^l X$ i stoga

$$\|A^l\|\|X\| \geq \|A^l X\| = \|\lambda^l X\| = |\lambda|^l \|X\| = \rho(A)^l \|X\|.$$

Podijelimo li sve sa $\|X\|$ dobivamo $\rho(A)^l \leq \|A^l\|$. Druga nejednakost slijedi iz svojstva d) definicije 2.6. □

Definicija 2.9 *Matrica* $A \in \mathbb{C}^{n\times n}$ *je normalna ako* $A^*A = AA^*$.

Lema 2.2 $A \in \mathbb{C}^{n\times n}$ *je* normalna ako i samo ako ima n ortonormiranih svojstvenih vektora, odnosno n svojstenih vektora $x_1, \ldots, x_n$ takvih da je $\langle x_i, x_j\rangle = \delta_{ij}$ za sve $i, j = 1, \ldots, n$.

(δ_{ij} je Kroneckerov simbol i definira se kao: $\delta_{ij} = 1$ za $i = j$ i $\delta_{ij} = 0$ za $i \neq j$)

Teorem 2.3 *Neka je* $A \in \mathbb{C}^{n\times n}$ normalna matrica*, tada*

$$\rho(A)^l = \|A^l\|_2 = \|A\|_2^l \qquad \forall\, l \in \mathbb{N}.$$

Dokaz. Neka je $x_1, \ldots, x_n$ ortonormirana baza sastavljena od svojstvenih vektora matrice A s odgovarajućim svojstvenim vrijednostima $\lambda_1, \ldots, \lambda_n$. Bez smanjenja općenitosti možemo pretpostaviti da je $\rho(A) = |\lambda_1|$.

Neka je $x \in \mathbb{C}^n$. Tada možemo pisati

$$x = \sum_{j=1}^{n} \alpha_j x_j$$

što daje

$$\|x\|_2^2 = \sum_{j=1}^{n} |\alpha_j|^2.$$

Slično dobivamo

$$Ax = \sum_{j=1}^{n} \alpha_j \lambda_j x_j \qquad \text{i} \qquad \|Ax\|_2^2 = \sum_{j=1}^{n} |\alpha_j \lambda_j|^2.$$

Iz ovoga slijedi

$$\|Ax\|_2^2 \le |\lambda_1|^2 \sum_{j=1}^{n} |x_j|^2 = \rho(A)^2 \|x\|_2^2,$$

što povlači $\|A\|_2 \le \rho(A)$.

Primjenom teorema 2.2 dobivamo

$$\rho(A)^l \le \|A^l\|_2 \le \|A\|_2^l \le \rho(A)^l$$

za sve $l \in \mathbb{N}$. Time je teorem dokazan. □

Koristeći metode slične onima koje smo koristili u dokazu prethodnog teorema, možemo dokazati sljedeći teorem.

Teorem 2.4 *Za sve matrice* $A \in \mathbb{C}^{n\times n}$ *vrijedi*

$$\|A\|_2 = \sqrt{\rho(A^*A)}.$$

Prethodni teorem između ostalog daje jednu karakterizaciju matrične norme inducirane vektorskom 2-normom. Sljedeći teorem eksplicitno opisuje matričnu normu induciranu vektorskom ∞-normom.

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

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

Dokaz. Za $x \in \mathbb{C}^n$ vrijedi

$$\begin{aligned} \|Ax\|_\infty &= \max_{1\le i\le n} |(Ax)_i| = \max_{1\le i\le n} \left| \sum_{j=1}^{n} a_{ij}x_j \right| \\ &\le \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}| \|x\|_\infty \end{aligned}$$

iz čega slijedi

$$\frac{\|Ax\|_\infty}{\|x\|_\infty} \le \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$$

za sve $x \in \mathbb{C}^n$ i stoga $\|A\|_\infty \le \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$.

Za dokaz donje ograde izaberimo $k \in \{1, 2, \ldots, n\}$ takav da

$$\max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}| = \sum_{j=1}^{n} |a_{kj}|$$

i definirajmo $x \in \mathbb{C}^n$ tako da $x_j = \overline{a_{kj}}/|a_{kj}|$ za sve $j = 1, \ldots, n$. Tada imamo $\|x\|_\infty = 1$ i

$$\begin{aligned} \|A\|_\infty &\geq \frac{\|Ax\|_\infty}{\|x\|_\infty} = \frac{\max_{1\leq i\leq n}\left|\sum_{j=1}^n a_{ij}\ \frac{\overline{a_{kj}}}{|a_{kj}|}\right|}{1} \\ &\geq \left|\sum_{j=1}^n a_{kj}\frac{\overline{a_{kj}}}{|a_{kj}|}\right| \\ &= \sum_{j=1}^n |a_{kj}| \\ &= \max_{1\leq i\leq n}\sum_{j=1}^n |a_{ij}| \end{aligned}$$

Time je teorem dokazan. □

2.2.1 Zadaci

1. Pokažite da je preslikavanje $\nu(A) : \mathbb{C}^{n\times n} \to \mathbb{R}, n \in \mathbb{N}$

$$\nu(A) = n \cdot \max_{i,j} |a_{ij}|, \quad A \in \mathbb{C}^{n\times n},$$

   matrična norma.

2. Pokažite da za matričnu normu induciranu s vektorskom normom $\|\cdot\|_1$ na $\mathbb{C}^n$ vrijedi

$$\|A\|_1 = \max_{1\leq j\leq n}\sum_{i=1}^n |a_{ij}|.$$

3. Pokažite da za sve kvadratne matrice $A \in \mathbb{C}^{n\times n}$ funkcija $\|A\|_{max} = \max_{i,j} |a_{ij}|$ definira vektorsku normu na prostoru matrica $n \times n$, ali to nije matrična norma.

4. Neka su $A, B, C \in \mathbb{C}^{n\times n}$ takve da je $A = BC$. Pokažite da je

$$\|A\|_{max} \leq \|B\|_\infty \|C\|_{max}$$

   i

$$\|A\|_{max} \leq \|B\|_{max} \|C\|_1.$$

5. Ako su U i V unitarne matrice odgovarajuće dimenzije, tada matričnu normu $\|\cdot\|$ za koju vrijedi

$$\|UAV\| = \|A\|,$$

nazivamo *unitarno invarijantna norma*. Dokažite da je matrična norma $\|\cdot\|_2$ unitarno invarijantna norma.

6. Preslikavanje $\|A\|_F : \mathbb{C}^{n\times n} \to \mathbb{R}, n \in \mathbb{N}$ zadano sa

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2}, \quad A \in \mathbb{C}^{n\times n},$$

je Frobeniusova norma na matricama $\mathbb{C}^{n\times n}$. Pokažite da je $\|A\|_F = \sqrt{\text{trag}(A^*A)}$ i dokažite da je $\|\cdot\|_F$ unitarno invarijantna norma.

7. Dokažite da matrične norme $\|\cdot\|_1$ i $\|\cdot\|_\infty$ nisu unitarno invarijantne norme.

8. Neka je $A = \begin{pmatrix} -1 & -3 \\ -2 & 4 \end{pmatrix}$. Izračunajte $\|A\|_1$, $\|A\|_\infty$, $\|A\|_F$ i $\|A\|_2$.

[Rješenje: $\|A\|_1 = 7$, $\|A\|_\infty = 6$, $\|A\|_F = \sqrt{30}$ i $\|A\|_2 = \sqrt{15 + 5\sqrt{5}}$]

9. Dokažite da je

$$\|A\|_2^2 \leq \|A\|_1 \|A\|_\infty.$$

10. Dokažite da za norme $\|\cdot\|_2$ i $\|\cdot\|_F$ vrijedi

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\|A\|_2.$$

[Uputa za rješavanje zadatka: Iskoristite tvrdnju da za normalnu matricu $N \in \mathbb{C}^{n\times n}$ postoji unitarna matrica $U \in \mathbb{C}^{n\times n}$ takva da je $U^*NU = \Lambda$, gdje je Λ dijagonalna matrica koja sadrži svojstvene vrijednosti matrice N na dijagonali.]

2.3 Kratki uvod u MatLab

2.3.1 Što je MatLab?

MatLab (Matrix Laboratory) je programski jezik visokih performansi namijenjen za tehničke proračune. Objedinjava računanje, vizualizaciju i programiranje u lako uporabljivoj okolini u kojoj su problem i rješenje definirani poznatom matematičkom notacijom. Uobičajena je uporaba MatLab-a za:

- matematiku i izračune,
- razvoj algoritama,
- modeliranje, simulaciju, analizu,
- analizu i obradu podataka, vizualizaciju,
- znanstvenu i inženjersku grafiku,
- razvoj aplikacija, uključujući i izgradnju grafičkog korisničkog sučelja.

MatLab je i okružje i programski jezik. Jedna od jačih strana MatLab-a je činjenica da njegov programski jezik omogućava izgradnju vlastitih alata za višekratnu uporabu. Možete lako sami kreirati vlastite funkcije i programe (poznate kao M-datoteke) u kodu MatLab-a. Skup specijaliziranih M-datoteka za rad na određenoj klasi problema naziva se Toolbox. S MatLab-om dolazi nekoliko Toolbox-ova koji su i više od kolekcije korisnih funkcija. Oni predstavljaju rezultate istraživanja vrhunskih stručnjaka iz područja upravljanja, obrade signala, identifikacije procesa, i drugih. Dakle, uz pomoć MatLab-a možete sami razviti nove ili adaptirati postojeće Toolbox-ove za rješavanje određenih problema.

Naredbe za MatLab unosimo u komandni prozor, osnovni prozor MatLab-a. Taj je prozor neka vrsta terminala operacijskog sustava i u njemu vrijede i osnovne terminalske operacijske naredbe za manipulaciju datotekama. Trenutni direktorij možemo promijeniti poznatom naredbom cd, a izvršavati možemo funkcije/naredbe koje su u path-u. Osim toga, uz MatLab novije verzije dolazi i vlastiti editor M-datoteka s debugerom.

Osnovni elementi programskog paketa MatLab su:

Razvojna okolina.

Skup alata za lakšu uporabu MatLab-a i njegovih funkcija. Mnogi od ovih alata (u verziji 6) realizirani su u grafičkom sučelju. To su MatLab desktop, komandni prozor (engl. command window), povijest naredbi (engl. command history), editor i debuger, te preglednici help-a, radnog prostora (engl. workplace), datoteka te path-a.

Biblioteka matematičkih funkcija.

Ogromna kolekcija računalnih algoritama.

Programski jezik.

MatLab programski jezik je jezik visokog stupnja, matrično orijentiran, s naredbama uvjetnih struktura, funkcija, strukturiranim podacima, ulazom/izlazom te nekim svojstvima objektno-orijentiranog programiranja. Ovaj programski jezik omogućava programiranje na nižoj razini (kao npr. za potrebe studenata) ali i za složenije programe većih razmjera.

Grafički alat.

MatLab raspolaže velikim mogućnostima za grafički prikaz podataka, vektora i matrica, kao i notaciju i printanje tih dijagrama. Postoje funkcije visokog stupnja za 2D i 3D vizualizaciju podataka, obradu slike, animaciju. Također postoje i funkcije za izgradnju grafičkih sučelja za vaše MatLab aplikacije.

Sučelje programskih aplikacija. Biblioteka (engl. Application Program Interface - API) koja omogućava razvoj C i Fortran programa koji mogu biti u interakciji s MatLab-om.

U ovom dokumentu bit će pokazane samo osnovne funkcije MatLab-a dostatne za početak samostalnog rada u MatLab-u. Za sve daljnje informacije proučite MatLab-ov help ili dodatne upute koje dolaze s instalacijom.

2.3.2 Jednostavni matematički računi

MatLab može poslužiti kao vrhunski linijski kalkulator, ali krenimo od osnovnih funkcija:

```
>> 4*25+3
ans =
   103
```

U radnom prostoru MatLab-a možemo definirati varijable:

```
>> a=4
```

```
a =
     4
>> b=25;
>> c=3;
>> d=a*b+c
d =
   103
```

Primijetimo da ukoliko naredbu završimo s ';' MatLab ne ispisuje rezultat te naredbe. Osim toga, ';' služi za razdvajanje više naredbi u jednom redu. Ukoliko je naredba predugačka za jedan red,ako na kraju tog reda dodamo '...', ista se nastavlja u sljedećem redu.

Kao i kod svih računalnih jezika tako i ovdje postoje neka osnovna pravila kod imenovanja varijabli:

- potrebno je razlikovati uporabu velikih/malih slova,
- maksimalni broj znakova je 31,
- prvi znak mora biti slovo.

MatLab ima sljedeće posebne varijable čiji su nazivi rezervirani: `ans`, `pi`, `eps`, `flops`, `inf`, `nan`, `i`, `j`, `nargin`, `nargout`, `realmin`, `realmax`.

2.3.3 Kompleksni brojevi

Jedna od pogodnosti MatLab-a je jednostavnost izvršavanja operacija s kompleksnim brojevima. Možemo ih formirati na više načina, a operacije s kompleksnim brojevima analogne su operacijama s realnim brojevima.

```
>> kmpl1=2-3i
kmpl1 =
    2.0000 - 3.0000i
>> kmpl2=6+sin(pi/3)*i
kmpl2 =
    6.0000 + 0.8660i
>> kmpl3=(kmpl1-kmpl2*2)*kmpl1
kmpl3 =
  -34.1962 +20.5359i
```

Također je moguća jednostavna konverzija između različitih zapisa kompleksnih brojeva

$$A \cdot e^{\theta \cdot j} \qquad \text{polarne koordinate}$$

$$a + b \cdot i \qquad \text{pravokutne koordinate}$$

pomoću funkcija `real`, `imag`, `abs`, `angle`:

```
>> p_kmpl1=abs(kmpl1)*exp(angle(kmpl1)*j)
p_kmpl1 =
   2.0000 - 3.0000i
>> real(p_kmpl1)
ans =
     2
>> imag(p_kmpl1)
ans =
    -3
```

2.3.4 Osnovne matematičke funkcije

MatLab podržava osnovne matematičke funkcije (kao npr. `abs(x)`, `acos(x)`, `sqrt(x)`, `sin(x)`, `tan(x)`, `asin(x)`, `atan(x)`, ...).

```
>> x=sqrt(2)/2
x =
    0.7071
>> y=asin(x)
y =
    0.7854
>> y_s=y*180/pi
y_s =
   45.0000
```

Zanimljiv je primjer funkcije arkus tangensa za koju pored funkcije atan postoji i funkcija `atan2` koja ima domenu u sva četiri kvadranta:

```
>> 180/pi*atan(-2/3)
ans =
   -33.6901
>> 180/pi*atan2(-2,3)
ans =
   -33.6901
>> 180/pi*atan2(2,-3)
ans =
   146.3099
```

2.3.5 MatLab-ov radni prostor

Dok radite u komandnom prozoru, MatLab pamti varijable koje ste koristili. Varijable koje su u radnom prostoru možemo vidjeti u pregledniku radnog prostora (prozor Workplace) ili u komandnom prozoru naredbom `who` (ispis varijabli) i `whos` (detaljniji ispis varijabli). Na bilo koji od ovih načina možemo doći do informacija o tipu određene varijable, njenoj veličini i dimenziji.

Tako je u našem primjeru (u komandnom prozoru):

```
>> who
Your variables are:
a         b         d         kmpl2     p_kmpl1   y
ans       c         kmpl1     kmpl3     x

>> whos
        Name        Size     Elements     Bytes    Density   Complex
          a       1 by 1           1         8       Full      No
         ans      1 by 1           1         8       Full      No
          b       1 by 1           1         8       Full      No
          c       1 by 1           1         8       Full      No
          d       1 by 1           1         8       Full      No
       kmpl1      1 by 1           1        16       Full      Yes
       kmpl2      1 by 1           1        16       Full      Yes
       kmpl3      1 by 1           1        16       Full      Yes
     p_kmpl1      1 by 1           1        16       Full      Yes
          x       1 by 1           1         8       Full      No
          y       1 by 1           1         8       Full      No
```

```
Grand total is 11 elements using 120 bytes
```

Ukoliko neku vrijablu želimo izbrisati iz radnog prostora koristimo naredbu clear na način:

```
>> clear p_kmpl1 x y ans
>> who
Your variables are:
a         c         kmpl1     kmpl3
b         d         kmpl2
```

Samom naredbom `clear` bez dodatnih opcija izbrisali bismo sve varijable iz radnog prostora.

Ista mogućnost stoji nam na raspolaganju s preglednikom radnog prostora - pritiskom desne tipke miša možemo odabrati `delete` za brisanje odabrane varijable (ili ikonicu za brisanje) te `clear` Workplace za brisanje svih varijabli iz radnog prostora.

2.3.6 Spremanje i ponovna uporaba podataka

Sadržaj radnog prostora možemo spremiti u binarnom formatu u željenu datoteku `ime.mat` i to na nekoliko načina:

- iz menija File—Save Workspace as...
- iz preglednika radnog prostora - pritiskom desne tipke miša odabirom Save Workplace as... ili ikonicom za spremanje.
- naredbom u komandnom prozoru

```
>> save ime
```

Opis uporabe svake naredbe/funkcije u komandnom prozoru MatLab-a možemo dobiti na sljedeći način:

```
>> help save
 SAVE   Save workspace variables on disk.
    SAVE fname saves all workspace variables to the binary "MAT-file"
        named fname.mat. The data may be retrieved with LOAD. Omitting the
        filename causes SAVE to use the default filename "MatLab.mat".
```

```
SAVE fname X  saves only X.
SAVE fname X Y Z  saves X, Y, and Z.
SAVE fname X Y Z  -ascii  uses 8-digit ASCII form instead of binary.
SAVE fname X Y Z  -ascii -double  uses 16-digit ASCII form.
SAVE fname X Y Z  -ascii -double -tabs  delimits with tabs.
    If fname is "stdio", SAVE sends the data to standard output.
The binary formats used in MAT-files depend upon the size and type of
each matrix.  Small matrices and matrices with any noninteger entries
are saved in floating point format requiring 8 bytes per real element.
Large, integer matrices may be saved in compact formats requiring
only 1, 2 or 4 bytes per element.  See the External Interface Library
for more details, including C and Fortran routines to read and write
MAT-files from external programs.
See also LOAD, DIARY, FWRITE, FPRINTF, IMWRITE.
```

Ukoliko želimo spremiti samo neke varijable u datoteku data.mat odgovarajuća naredba je

```
>> save data kmpl1 kmpl2 kmpl3
```

Tada je datoteka pohranjena u trenutnom direktoriju.

Osim binarnog formata za spremanje se može koristiti i ASCII format, ali samo iz komandnog prozora. Tako s naredbom

```
>> save data.dat kmpl1 kmpl2 kmpl3 -ascii
```

snimamo željene varijable iz popisa u ACII datoteku `data.dat`.

Učitavanje željenih podataka iz vanjske datoteke `ime.mat` može se provesti na nekoliko načina:

- iz menija File—Load Workspace...
- iz preglednika radnog prostora - pritiskom desne tipke miša i odabirom Import Data... ili ikonicom za otvaranje
- naredbom u komandnom prozoru

```
>> load ime
```

Podatke zapisane u datoteci s ASCII formatom možemo učitati u MatLab primjenom čarobnjaka iz menija File—Import Data.... U komandnom prozoru ASCII datoteka se učitava naredbom `>> load data.dat`. Pri tome datoteka data.dat mora imati oblik matrice, tj. jednak broj elemenata u

svakom retku (ti elementi mogu biti odvojeni razmakom, zarezom ili tabularom). Ukoliko redak započinje znakom %, taj se redak ne učitava. MatLab ovakvom naredbom u radni prostor sprema varijablu (matricu) imena data.

Ponovnim učitavanjem (ili definiranjem) varijable s istim imenom varijabla poprima novu vrijednost.

2.3.7 Formiranje matrica

Krenimo za početak od zapisa matrica u MatLab-u. Elemente matrice unosimo između zagrada [ ]:

```
>> A=[1,2,3;4,5,6;7,8,9]
A =
     1     2     3
     4     5     6
     7     8     9
>> B=[1 2 3;4 5 6;7 8 9]
B =
     1     2     3
     4     5     6
     7     8     9
```

Vektor, odnosno jednodimenzionalno polje možemo zapisati na sljedeći način

```
>> x=[1 2 3]
x =
     1     2     3
```

Dakle, razdvajanjem elemenata s razmacima ili zarezima definiramo elemente u različitim stupcima, dok razdvajanjem s ; ili novim redom <enter> definiramo elemente u različitim redovima.

Moguće je i automatizirano formiranje vektora

```
>> x=(0:0.1:1)
x =
  Columns 1 through 7
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
  Columns 8 through 11
```

```
    0.7000    0.8000    0.9000    1.0000


>> y=linspace(0,1,11)
y =
  Columns 1 through 7
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
  Columns 8 through 11
    0.7000    0.8000    0.9000    1.0000
```

U MatLab-u postoje funkcije kojima se mogu definirati matrice čiji su elementi jednaki jedinici i nuli (nul matrica).

```
>> P=ones(3)
P =
     1     1     1
     1     1     1
     1     1     1
>> Q=zeros(3)
Q =
     0     0     0
     0     0     0
     0     0     0
```

2.3.8 Pristupanje dijelu matrice

Pojedini element matrice možemo ispisati definiranjem njegova retka i stupca (npr. element matrice `A` u prvom retku i drugom stupcu).

```
 >> A(1,2)
 ans =
     2
```

Ukoliko želimo vidjeti prva dva retka matrice `A`

```
>> A(1:2,:)
ans =
     1     2     3
     4     5     6
```

Moguća je korekcija pojedinih elemenata (npr. želimo li promijeniti zadnji redak matrice A vektorom-retkom r)

```
>> r=[101 102 103];
>> A(3,:)=r
A =
     1     2     3
     4     5     6
   101   102   103
```

ili nadopuna matrice (npr. želimo li matricu B proširiti s dodatnim redom jednakim vektoru-retku r)

```
>> B=[B;r]
B =
     1     2     3
     4     5     6
     7     8     9
   101   102   103
```

Ukoliko bismo kod matrice A definirali element u drugom redu i šestom stupcu

```
>> A(2,6)=1
A =
     1     2     3     0     0     0
     4     5     6     0     0     1
   101   102   103     0     0     0
```

matrica se proširuje na potrebnu dimenziju dodavajući na novodefiniranim mjestima nule. Ukoliko dio matrice izjednačimo s praznom matricom `[ ]`, isti dio se briše čime se početna matrica svodi na ostatak:

```
>> A(:,4:5)=[ ]
A =
     1     2     3     0
     4     5     6     1
   101   102   103     0
```

2.3.9 Operacije skalar - matrica

Matrici možemo dodati i/ili oduzeti skalar pri čemu rezultat zadržava izvornu dimenziju

```
>> A=[1,2,3;4,5,6;7,8,9];
>> A-1
ans =
     0     1     2
     3     4     5
     6     7     8
>> 2*A-1
ans =
     1     3     5
     7     9    11
    13    15    17
```

Također je definirana operacija potenciranja matrice sa skalarom

```
>> A.^2
ans =
     1     4     9
    16    25    36
    49    64    81
```

Pri tome se koristi `'.^'` za označavanje operacije potenciranja koja se odnosi na elemente.

2.3.10 Operacije matrica - matrica

Za početak pokažimo kako se dobija transponirana matrica

```
>> A=[1,2,3;4,5,6;7,8,9]
A =
     1     2     3
     4     5     6
     7     8     9
>> B=A'
B =
```

```
     1     4     7
     2     5     8
     3     6     9
```

Za matrice jednake dimenzije moguće je definirati operaciju zbrajanja

```
>> A+B
ans =
     2     6    10
     6    10    14
    10    14    18
>> 2*A-B
ans =
     1     0    -1
     6     5     4
    11    10     9
```

Ukoliko matrice imaju odgovarajuće dimenzije (ako je broj stupaca prve jednak broju redaka druge), moguće je izvršiti i operaciju množenja

```
>> A*B
ans =
    14    32    50
    32    77   122
    50   122   194
>> C=[1 1;2 2;3 3]
C =
     1     1
     2     2
     3     3
>> A*C
ans =
    14    14
    32    32
    50    50
>> D=[1 1 1; 2 2 2]
D =
     1     1     1
```

```
     2     2     2
>> D*A
ans =
    12    15    18
    24    30    36
```

U prethodnom poglavlju navedeno je potenciranje koje se odnosilo na elemente a koje je bilo naznačeno s '.^'. Potenciranje koje bi se odnosilo na cijelu matricu je

```
>> A^2
ans =
    30    36    42
    66    81    96
   102   126   150
```

što je zapravo

```
>> A*A
ans =
    30    36    42
    66    81    96
   102   126   150
```

2.3.11 Operacije na elementima matrica

Ovdje su opisane operacije koje se provode po odgovarajućim elementima matrica. Ukoliko su matrice jednakih dimenzija, moguće je primijeniti operacije množenja ('.*'), dijeljenja ('./' s desne i '.\' s lijeve strane) i potenciranja ('.^') po elementima

```
>> A,D
A =
     1     2     3
     4     5     6
     7     8     9
D =
     1     1     1
     2     2     2
```

```
     3     3     3
>> A.*D
ans =
     1     2     3
     8    10    12
    21    24    27
>> A./D
ans =
    1.0000    2.0000    3.0000
    2.0000    2.5000    3.0000
    2.3333    2.6667    3.0000
>> D./A
ans =
    1.0000    0.5000    0.3333
    0.5000    0.4000    0.3333
    0.4286    0.3750    0.3333
>> A.\D
ans =
    1.0000    0.5000    0.3333
    0.5000    0.4000    0.3333
    0.4286    0.3750    0.3333
>> A.^D
ans =
     1     2     3
    16    25    36
   343   512   729
```

2.3.12 Dimenzije matrica i vektora

Već je ranije opisano kako doznati informaciju o određenoj varijabli. Tako se dimenzija podatka iz radnog prostora može vidjeti u pregledniku radnog prostora ili iz komandnog prozora.

```
>> whos
Name      Size          Elements     Bytes    Density   Complex
 A       3 by 3           9          72       Full       No
 B       3 by 3           9          72       Full       No
 C       3 by 2           6          48       Full       No
```

```
 D       3 by 3           9          72       Full       No
 P       3 by 3           9          72       Full       No
 Q       3 by 3           9          72       Full       No
ans       1 by 1          1           8       Full       No
 r       1 by 3           3          24       Full       No
 total is 55 elements using 440 bytes
```

Kad nam je potrebna dimenzija određene matrice u nekim algoritmima možemo koristiti sljedeće funkcije

```
>> R=[1 2 3 4;5 6 7 8]
R =
     1     2     3     4
     5     6     7     8
>> [red,stup]=size(R)
red =
     2
stup =
     4
>> length(R)
ans =
     4
```

2.3.13 Sustav linearnih jednadžbi

Već je spomenuto da je problem rješavanja sustava linearnih jednadžbi jedan od najčešćih problema linearne algebre. Promotrimo na primjer sljedeći sustav

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 366 \\ 844 \\ 351 \end{bmatrix}$$

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

Kad je osigurana egzistencija rješenja sustava linearnih jednadžbi postoji nekoliko pristupa za rješavanje sustava: Gaussova eliminacija, LU faktorizacija ili izravna uporaba inverzne matrice $\mathbf{A}^{-1}$. Analitičko rješenje može se zapisati u obliku

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b} \, .$$

Diskusija o analitičkom i numeričkom rješenju sustava linearnih jednadžbi izvan je interesa ovog teksta. Cilj nam je pokazati kako se MatLab može primijeniti u rješavanju ovih problema. Riješimo gornji sustav pomoću inverzne matrice:

```
>> A=[1 2 3;4 5 6;7 8 0]
A =
     1     2     3
     4     5     6
     7     8     0
>> b=[366;804;351]
b =
   366
   804
   351
>> x=inv(A)*b
x =
   25.0000
   22.0000
   99.0000
```

Uz korištenje inverzne matrice moguće je riješiti sustav primjenom operacije dijeljenja s lijeve strane pri čemu se koristi pristup LU faktorizacije

```
>> x=A\b
x =
   25.0000
   22.0000
   99.0000
```

Ovaj drugi pristup određivanju rješenja češće se primjenjuje iz nekoliko razloga. Jedan od osnovnih je u tome što ima manje operacija od pristupa s inverznom matricom, što ga čini bržim. Osim toga, u slučaju velikih matrica drugi pristup daje točnija rješenja.

Zanimljivo je ovdje iznijeti i funkciju kojom možemo odrediti determinantu matrice

```
>> det(A)
ans =
    27
```

Od interesa mogu biti i sljedeće funkcije: `eig(A)` (vlastite vrijednosti i vlastiti vektori matrice), `norm(A)` (norma matrice), `poly(A)` (karakteristični polinom matrice), `rank(A)` (rang matrice), `trace(A)` (trag matrice), te specijalne matrice `eye(n)` (jedinična matrica dimenzije n), `ones(n,m)` (matrica dimenzije $n \times m$ s elementima jednakim 1), `zeros(n,m)` (matrica dimenzije $n \times m$ s elementima jednakim 0 - nul matrica)

2.3.14 Programi i funkcije u MatLab-u

MatLab ima i mogućnost razvoja algoritama u vlastitom programskom jeziku. Datoteke s takvim algoritmima nazivamo M-datoteke, a pohranjuju se s ekstenzijom '.m'. Pri tome možemo razlikovati dvije vrste M-datoteka: skripte i funkcije. Skripte su jednostavno skup naredbi koje se prenose i izvršavaju u komandnom prozoru (adekvatno glavnom programu). Funkcije su na neki način crne kutije kojima dajemo određeni ulaz i dobijamo traženi izlaz (slično potprogramima ili bibliotekama).

Sve naredbe koje smo do sada upoznali i primijenili izravno u komandnom prozoru MatLab-a mogu se primijeniti i u M-datotekama.

2.3.15 Funkcijske M-datoteke

Kod funkcijskih datoteka varijable su lokalne i nema ih u radnom prostoru i zato možemo reći da je funkcija na neki način crna kutija. Funkcijska datoteka komunicira s radnim prostorom samo preko varijabli ulaza i varijabli izlaza.

Osnovna forma funkcijske M-datoteke dana je na sljedeći način:

Glavni element je prva linija u kojoj se definira funkcija sa svojim imenom (to ime određuje i ime datoteke u kojoj je spremljena funkcija), ulaznim i izlaznim varijablama. Nakon nje slijedi niz komentar linija koje predstavljaju help funkcije i pri tome je prva od njih (naziva se H1 linija) posebna jer se ona pretražuje naredbom lookfor i uobičajeno je da sadrži ime i kratki opis funkcije. Nakon komentar linija slijedi samo tijelo funkcije.

Uz pretpostavku da je funkcijska M-datoteka (imedatoteke.m) smještena u MAT LAB-ovu path-u, funkcija se izvršava pozivom u MatLab-ovu ko-

mandnom prozoru (ili u nekoj drugoj M-datoteci) na sljedeći način:
`[izl1,izl2,...]=imedatoteke(ul1,ul2,...)` pri čemu su `ul1,ul2,...` ulazne varijable, a `izl1,izl2,...` izlazne.

Analizirajmo funkcijske M-datoteke na primjeru funkcije average.m koja određuje srednju vrijednost za elemente jednog vektora. Pri tome ulazna varijabla ne smije biti skalar ili matrica.

```
function a = average(b)
% AVERAGE Srednja vrijednost elemenata vektora.
% AVERAGE(B), gdje je B vektor, predstavlja srednju
% vrijednost elemenata vektora.
% Za ne-vektorski ulaz funkcija dojavljuje gresku.
[m,n] = size(b); if (~((m == 1) | (n == 1)) | (m == 1 & n == 1))
    error('Ulaz mora biti vektor!')
end
a = sum(b)/length(b);      % izracun srednje vrijednosti
```

U komandnom prozoru funkciju pozivamo na sljedeći način

```
>> y=average(x)
y =
    0.7850
```

2.3.16 Petlje i uvjetne strukture

Ukoliko niste od ranije upoznati s mogućnostima kontrole toka i strukture algoritama koje pružaju razni programski jezici, ovo poglavlje može vam biti složeno. U tom ga slučaju pažljivo prijeđite.

Uvjetne strukture jak su alat, budući da omogućavaju utjecaj prijašnjih operacija algoritma na buduće. MatLab pruža četiri oblika petlji, odnosno uvjetnih struktura: for petlje, while petlje, if-else-end struktura i switch-case struktura.

2.3.17 for petlje

`for` petlje omogućavaju da se grupa naredbi ponavlja unaprijed određeni broj puta. Opći oblik for petlje je

```
for x = array
    naredbe...
end
```

Naredbe između for i end izvršavaju se jednom za svaki stupac u array. Na primjer,

```
>> for n=1:10
     x(n)=sin(n*pi/10);
   end
>> x
x =
 Columns 1 through 7
 0.3090    0.5878    0.8090    0.9511    1.0000    0.9511    0.8090
 Columns 8 through 10
 0.5878    0.3090    0.0000
```

Osim automatski generiranog polja 1:10 može se primijeniti bilo koje polje, npr.

```
>> data=[3 9 45 6; 7 16 -1 5];
>> for n=data
      y=n(1)-n(2)
   end
y =
    -4
y =
    -7
y =
    46
y =
     1
```

Pored ovih mogućnosti for petlja može biti ugniježđena jedna u drugoj.

2.3.18 while petlja

Kod `while` petlje naredbe između while i end izvršavaju se sve dok su svi elementi izraza istiniti:

```
>> while izraz
      naredbe...
   end
```

Razmotrimo sljedeći primjer:

```
>> num=0;EPS=1;
>> while (1+EPS)>1
      EPS=EPS/2;
      num=num+1;
   end
>> num
num =
    53
>> EPS=2*EPS
EPS =
  2.2204e-016
>> eps
eps =
  2.2204e-016
```

Ovaj primjer prikazuje jedan od načina određivanja vrijednosti najmanjeg broja (EPS) koji može biti dodan broju 1 tako da je rezultat tog zbroja $1+\text{EPS} > 1$ koristeći konačnu preciznost. Našu smo varijablu EPS usporedili sa specijalnom funkcijom MatLab-a - eps. Što se zapravo događa u ovoj petlji? Vrijednost EPS stalno se smanjuje od početne vrijednosti (1) pri svakom prolazu kroz petlju. Budući da MatLab koristi 16 znamenaka pri prezentiranju brojeva za očekivati je da EPS (odnosno *eps*) bude oko 10^{-16}. Tako je uvjet $1+\text{EPS} > 1$ neistinit (jednak nuli) i petlja se prekida. Na kraju EPS množimo sa 2 jer se u zadnjem prolazu kroz petlju stvarna vrijednost EPS smanjila za 2.

2.3.19 if-else-end struktura

Često želimo izvršiti neke operacije pod uvjetom da su zadovoljeni određeni uvjeti. To nam omogućava if-else-end struktura. Oblik ove strukture u općem slučaju je

```
if izraz1
```

```
   naredbe1 ... izvr\v{s}ene ako je izraz1 istinit
elseif izraz2
   naredbe2 ... izvrsene ako je izraz2 istinit
elseif izraz3
   naredbe3 ... izvrsene ako je izraz3 istinit
elseif ...
   naredbe4 ... izvrsene ako je izraz4 istinit
...
else
   naredbe ... izvrsene ako nijedan izraz nije istinit
end
```

Pri izvršavanju ove strukture provjerava se izraz1, pa izraz2, ... Ukoliko je neki od tih izraza istinit, izvršavaju se pripadajuće naredbe - narebe1, naredbe2, naredbe3, ... Ako nijedan od izraza nije istinit, izvršavaju se naredbe iza else. Završni dio strukture else ne mora se uvijek primijeniti. Promotrimo primjer iz predhodnog poglavlja:

```
>> EPS=1;
>> for num=1:1000
      EPS=EPS/2;
      if(1+EPS)<=1
         EPS=EPS*2
         break
      end
   end
EPS =
  2.2204e-016
>> num
num =
    53
```

Naredba break uzrokuje izlazak iz for petlje na prvu naredbu koja slijedi. U ovom slučaju vraća se u komandni prozor i prikazuje vrijednost varijable EPS.

2.3.20 Grafika

Već su u prethodnom poglavlju korištene neke grafičke mogućnosti MatLab-a. U ovom poglavlju bit će dan detaljniji prikaz osnovnih naredbi za grafičku

prezentaciju podataka te kratki prikaz naredbi za specijalnu i 3D grafiku.

2.3.21 Tipična grafička sesija u MatLab-u

Procedura pri grafičkoj prezentaciji podataka u MatLab-u, bilo da se radi u komandom prozoru s podacima iz radnog prostora ili u M-datoteci s definiranim podacima, ima sljedeće korake:

1. priprema podataka,
2. odabir prozora, pozicije za dijagram,
3. iscrtavanje podataka,
4. postavljanje karakteristika linija i markera,
5. postavljanje karakteristika osi dijagrama i mreže,
6. notacije na dijagramu,
7. eksportiranje grafike.

Određeni koraci mogu se naravno po potrebi i preskočiti. U ovom poglavlju bit će dane osnovne naredbe za provedbu gore navedenih koraka. Prvi korak neće se detaljnije opisivati budući da je njegov sadržaj bio predmet prethodnih poglavlja. Ukoliko je grafika samo alat brze analize vaših rezultata, možda će prva tri koraka biti dovoljna, no ukoliko želite vaše rezultate grafički prezentirati, vjerojatno će biti potrebno odraditi sve korake.

2.3.22 Osnove grafike

Ovdje će se kroz kratke primjere koji slijede prezentirati naredbe uz njihov grafički rezultat, te kraći komentar postupka. Operacije izmjene izgleda i parametara samih dijagrama moguće je izvesti i preko grafičkog sučelja - Plot Edit Mode grafičkih prozora, no on ovdje neće biti opisan iz razloga što je njegovo savladavanje poslije ovog poglavlja vrlo intuitivno. Osim toga, pristup s naredbama ima prednost jer se može zapisati i ponovno izvršavati iz M-datoteke.

2.3.23 Osnovne grafičke naredbe

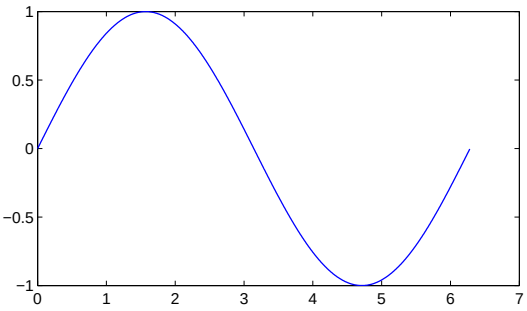
Ovo su najčešće naredbe koje se primjenjuju za grafički prikaz podataka u MatLab-u kao funkcije jedne varijable:

- naredba za crtanje dijagrama linearnog mjerila za obje osi, `plot`,
- naredba za crtanje dijagrama logaritamskog mjerila za obje osi, `loglog`,
- naredba za crtanje dijagrama logaritamskog mjerila za x-os i linearnog mjerila za y-os, `semilogx`,
- naredba za crtanje dijagrama logaritamskog mjerila za y-os i linearnog mjerila za x-os, `semilogy` ,
- dijagram s dvije y-osi (lijevo i desno), `plotyy` .

Sve ove funkcije imaju različitu sintaksu za različite potrebe korisnika. Detaljnije ćemo se upoznati samo s naredbom plot koja je i najčešća, a samo se dotaknuti naredbe plotyy zbog njene specifičnosti. Osim ovih naredbi postoji i mnoštvo naredbi za ostale oblike dijagrama (polarne dijagrame, bar, pite, histograme, ruže, ...), crtanje poligona i tijela te izradu animacija kao i prikaz bitmap grafike. No, njih ovdje nećemo opisivati.

2.3.24 Linijski prikaz podataka

Primjer najjednostavnijeg prikaza elemenata vektora `y=sin(t)` dan je na slici 2.1.



Slika 2.1: Graf funkcije $\sin x$

Dakle, u ovoj osnovnoj sintaksi naredbe `plot` ulazne varijable su dva vektora s apscisom i ordinatom svake točke . Na primjeru `plot(t,y)` želimo

prikazati diskretnu funkciju jedne varijable. Potrebna su nam dva vektora: `t` - vektor s vrijednostima apscisa svake točke i `y` - vektor s vrijednostima ordinata svake točke. Jasno je da ta dva vektora moraju imati jednak broj elemenata, u suprotnom MatLab javlja pogrešku. Ukoliko se kao ulazni parametar da samo jedan vektor kao varijabla, plot(y), grafički rezultat takve naredbe je prikaz vektora, ali s brojem elemenata kao parametrom na x-osi. Točke vektora su, u ovom defaultnom načinu poziva naredbe `plot(t,y)` međusobno spojene punim plavim linijama.

Ukoliko na istom dijagramu želimo istovremeno prikazati više vektora, to činimo na sljedeći način:

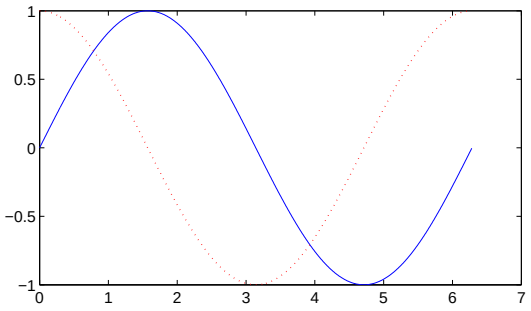
```
y1=sin(t-pi/6);
y2=sin(t-pi*2/6);
plot(t,y,t,y1,t,y2)
```

Ako želimo promijeniti oblik ili boju linije (primjeri sa slika 2.2 i 2.3)potrebno je iza svake kombinacije x i y koordinata dati odabranu šifru u jednostrukom apostrofu: `plot(t,cos(t),'r:')`. Dakle, oznaka `'r:'` predstavlja šifru za liniju točkastog oblika: u crvenoj boji `r`. Od raspoloživih oblika linija postoje puna, isprekidana, točkasta i linija-točka. Osnovne boje imaju svoje znakovne oznake, a uz to je moguće redefinirati bilo koju RGB boju. Više o samim oznakama proučite naredbom help plot.

Linijski prikaz: dvije krivulje - primjena naredbe hold on:

```
figure
plot(t,y)
hold on
plot(t,cos(t),'r:')
```

Prolaskom kroz ove primjere naredbe plot u komandnom prozoru MatLab-a primijetili ste da je nakon izvršenja prve naredbe za crtanje otvoren novi prozor s vašim dijagramom. Ponovne naredbe za crtanje izvodile su se na tom istom prozoru ali na način da su prethodne krivulje izbrisane. Ukoliko želimo zadržati prethodne krivulje i na dijagram dodati nove, potrebno je aktivirati naredbu hold on (slika 2.2). Kad želimo u jednom trenutku izbrisati sve stare i ostaviti samo zadnju krivulju dovoljno je prethodno dati naredbu hold off. Ako želimo izbrisati cijeli dijagram iz grafičkog prozora primjenjujemo naredbu clf, a naravno možemo i jednostavno zatvoriti grafički prozor - kao i svaki prozor u operacijskom sustavu ili koristiti naredbu close. Kad



Slika 2.2: Grafički prikaz dviju krivulja, primjena naredbe *hold on*

želimo zadržati prvi grafički prozor i na novom grafičkom prozoru nacrtati novi dijagram, dovoljno je dati naredbu figure.

Naredbom figure otvaramo novi grafički prozor (koji nosi oznaku redom sljedećeg broja). No postavlja se pitanje kako znamo na koji grafički prozor MatLab iscrtava našu krivulju. Iscrtava je na onaj prozor na kojemu je zadnjem bio fokus (windows terminologijom: onaj koji je zadnji bio aktiviran - tipkovnicom ili mišem). Fokus možemo definirati i naredbom figure(1) koja će postaviti 1. grafički prozor za aktivni prozor. To uvijek možemo provjeriti naredbom gcf koja nam kao rezultat vraća numeričku oznaku aktivnog prozora. Kod rada s većim brojem grafičkih prozora korisna je i naredba close all koja zatvara sve grafičke prozore.

2.3.25 Točkasti prikaz podataka

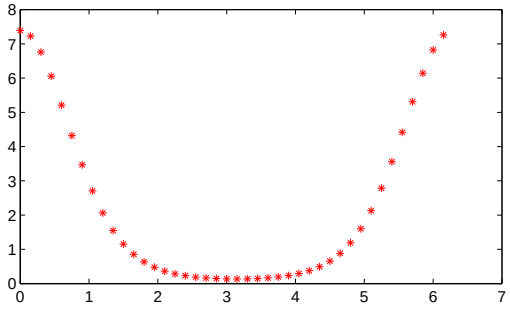
Kad elemente vektora želimo prikazati kao točke određenim znakom (markerom) ponovo primjenjujemo istu sintaksu naredbe plot, ali dodatne opcije umjesto linije definiraju odabrani znak: npr. `plot(t,y,'r*')`. Dostupne oblike znakova možete pregledati naredbom `help plot`.

Grafički prikaz funkcije $y = exp(2 * cos(t))$, za $t \in [0, \pi/2]$, možemo zapisati kao:

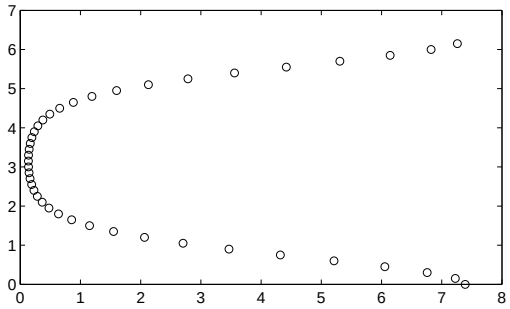
```
t=[0:0.15:pi/2]; y=exp(2*cos(t)); plot(t,y,'r*')
```

U isto vrijeme grafički prikaz funkcije t u ovisnosti o „varijabli" y možemo zapisati kao

```
plot(y,t,'ko')
```



Slika 2.3: Grafički prikaz podataka pomoću točaka funkcije $y = f(x)$



Slika 2.4: Grafički prikaz podataka pomoću točaka funkcije $x = f(y)$

Poglavlje 3

Složenost algoritama

U ovom poglavlju proučavat ćemo kako na pravilan način analizirati vrijeme potrebno za izvršavanje nekog algoritma. Posebno ćemo proučavati vezu između trajanja izvođenja pojedinog programa, njegove složenosti („troška programa" ili „koštanja programa", engl. costs - trošak) i dimenzija ulaznih podataka (matrica).

3.1 Složenost računanja

Složenost računanja nekog algoritma predstavlja ukupan broj svih operacija (računskih operacija, spremanja podataka u memoriju, vremena potrebnog za komunikaciju između procesora i memorije, itd.) potrebnih za izvođenje tog algoritma na računalu. Zbog jednostavnosti mi ćemo proučavati samo broj računskih operacija, točnije broj floating point operacija, potrebnih da se izvede neki algoritam (zbrajanje, množenje, oduzimanje, dijeljenje). Za detaljniju analizu kvalitete kojom se pojedini algoritam izvršava trebalo bi uzeti u obzir upotrebu memorije i komunikacijske potrebe.

Definicija 3.1 *Složenost računanja algoritma definiramo s $C(n)$, tako da je*

$$C(n) = \text{ ukupan broj zbrajanja, množenja, oduzimanja i dijeljenja}$$

pri čemu je n dimenzija ulaznih podataka (npr. broj jednadžbi u linearnom sustavu, itd.).

Sljedeća definicija sadrži potrebne pojmove pomoću kojih ćemo proučavati asimptotsko ponašanje pojedinog algoritma, tj. pomoću kojih ćemo izračunavati $C(n)$ za $n \to \infty$.

Definicija 3.2 *Za funkcije* $f, g : \mathbb{N} \to \mathbb{N}$ *ili* $f, g : \mathbb{R}^+ \to \mathbb{R}^+$ *pišemo*

$$g(n) = \mathcal{O}(f(n)) \quad \text{ako je} \quad \limsup_{n\to\infty} \frac{g(n)}{f(n)} < \infty,$$
$$g(n) = \Omega(f(n)) \quad \text{ako je} \quad \liminf_{n\to\infty} \frac{g(n)}{f(n)} > 0,$$
$$g(n) = \Theta(f(n)) \quad \text{ako je} \quad g(n) = \Omega((f(n)) \quad i \quad g(n) = \mathcal{O}(f(n)) .$$

Oznake Θ, Ω i $\mathcal{O}$ predstavljaju točnu, donju odnosno gornju ocjenu. Općenito se koriste i u drugim područjima matematika kao i računarstva.

Alternativa definiciji 3.2 bila bi sljedeća definicija:
Kažemo da je $g(n) = \Theta(f(n))$ ako postoje konstante c_1, c_2 i n_0 takve da za sve $n > n_0$ vrijedi da je

$$c_1 f(n) < g(n) < c_2 f(n) .$$

Primjer 3.1 *Koristeći gore navedenu notaciju možemo pisati:* $5n^2+2n-3 = \Theta(n^2), n^2 = \mathcal{O}(n^2)$ *i* $n^2 = \Omega(n)$.

Algoritam za standardni skalarni umnožak.
ulaz: $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$
izlaz: $s = \langle x, y \rangle$
1: $s = 0$
2: **for** $i = 1, \ldots, n$ **do**
3: $s = s + x_i y_i$
4: **end for**
5: return s

Teorem 3.1 *Trošak računanja algoritma za standardni skalarni umnožak na* $\mathbb{C}^n$ *iznosi* $C(n) = \Theta(n)$. *Najmanja složenost računanja standardnog skalarnog umnoška (donja granica za složenost računanja) iznosi* $C(n) = \Omega(n)$.

Dokaz. Iz gornjeg algoritma za standardni skalarni umnožak vidimo da nam je za skalarni umnožak dvaju n-dimenzionalnih vektora potrebno n množenja i n zbrajanja, to znači ukupno $C(n) = 2n = \Theta(n)$.

Skica dokaza za drugi dio teorema, tj. za donju granicu $C(n) = \Omega(n)$. Budući da su umnošci $x_i y_i$ međusobno neovisni, najmanje što moramo učiniti jest izračunati n takvih produkata. □

Napomena 3.1 *Kako bismo dali točan dokaz za donju granicu u gornjem teoremu, bila bi nam potrebna točna definicija pojma algoritam. Budući da to prelazi okvire našeg interesa u ovom pregledu mi ćemo se zadovoljiti samo jednostavnim skicama odnosno osnovnim idejama takvih dokaza. (Na primjer, netko bi mogao pomisliti da je pogađanje rezultata takodjer algoritam, a za to je potrebna samo jedna operacija. Stoga bi se takvi i slični slučajevi morali isključiti uvođenjem stroge definicije algoritma.)*

Teorem 3.2 *Za standardnu metodu za množenje kvadratnih matrica iz* $\mathbb{C}^{n\times n}$ *vrijedi da je* $C(n) = \Theta(n^3)$*, dok je donja ograda dana sa* $C(n) = \Omega(n^2)$ *(tj. najboljom metodom matrice možemo pomnožiti za* $\Omega(n^2)$*).*

Dokaz. Označimo redove matrice $A \in \mathbb{C}^{n\times n}$ sa $a_1^*, \ldots, a_n^*$ a stupce matrice $B \in \mathbb{C}^{n\times n}$ sa $b_1, \ldots, b_n$. Budući da je

$$(AB)_{ij} = a_i^* b_j \quad \text{za sve} \quad i, j = 1, \ldots, n\,,$$

vidimo da za jedan element matrice umnoška trebamo izračunati n umnožaka. Stoga je ukupni trošak dan sa $C(n) = n^2\Theta(n) = \Theta(n^3)$ (analiza je mogla ići i ovako: budući da imamo n^2 elemenata u matrici produkta AB, a za jedan elemenat nam treba n umnožaka, vidimo da je $C(n) = \Theta(n^3)$).

Skica dokaza za donju granicu izgledala bi ovako : najmanje što trebamo izračunati je n^2 elemenata iz matrice umnoška AB. Stoga je najbolje što možemo očekivati dano sa $C(n) \geq n^2$. □

Poglavlje 4

Stabilnost i uvjetovanost

Pogreške zokruživanja često uzrokuju to da je rezultat koji smo dobili na nekom računalu različit od rezultata očekivanog na osnovi teorije. Stoga ćemo u ovom poglavlju proučavati tehnike koje će nam pomoći pri dobivanju odgovora na pitanje: „Koliko je dobiveni rezultat blizu točnom (teorijski dobivenom) rezultatu?"

4.1 Pogreške

Prije nego se posvetimo proučavanju tehnika koje će nam poslužiti za dobivanje odgovora na pitanje „Koliko je dobiveni rezultat blizu točnom (teorijski dobivenom) rezultatu?" reći ćemo ponešto općenito o pogreškama. Prilikom numeričkog (približnog) rješavanja određenih matematičkih problema dolazi do pojave pogrešaka. Ugrubo ih možemo podijeliti u tri kategorije:

- pogreške matematičkog modela,
- pogreške metode,
- pogreške računanja.

Pogreške matematičkog modela.

Vrlo često matematički modeli idealiziraju stvarnu situaciju, pri čemu se pojedine stvari zanemaruju ili pojednostavljuju. Tako su na primjer mnogi matematički modeli koji opisuju određene fizikalne pojave (valna jednadžba za oscilacije žice ili proučavanje balističkih putanja) dobiveni zanemarivanjem nekih veličina, i time je već na početku načinjena pogreška, koja se kasnijim

postupcima ne može ispraviti. Također, ulazni podaci kao što su duljina žice, vanjske sile, početni položaj, početna brzina nisu potpuno točni, jer su oni rezultat nekih mjerenja.

Pogreške metode.

Metoda kojom rješavamo problem može sadržavati beskonačnu sumaciju, računanje neelementarnih integrala, rješavanje algebarskih jednadžbi višeg reda, itd. U takvim slučajevima pogreške na primjer dolaze zamjenom beskonačnih suma konačnim sumama (nekim od parcijalnih suma) reda.

Pogreške računanja.

Računala računaju sa strojnim (floating-point) brojevima koji su u njemu pohranjeni i pomoću kojih aproksimiramo skup realnih brojeva $\mathbb{R}$. Ako broj koji unosimo u računalo, ili s kojim on u nekom međukoraku mora baratati, nije strojni broj, onda ga računalo zaokružuje i time čini pogrešku. Jednostavniji proračun možemo izvršiti i bez računala, no i tada, ako želimo dobiti ispis rezultata pomoću znamenaka, brojeve kao što su $\pi, e, \ldots$ moramo zamijeniti približnim vrijednostima. Kako bismo se mogli pouzdati u rezultat nekog proračuna, moramo biti u stanju kontrolirati pogrešku.

Recimo sada nešto o pogreškama i o tome kako se one ponašaju prilikom izvođenja operacija.

Neka je a točna vrijednost nekog broja, i $\widetilde{a}$ njegova približna vrijednost. Kažemo da $\widetilde{a}$ aproksimira a.

Broj $|\tilde{a} - a|$ zovemo *apsolutnom pogreškom*, a broj $\dfrac{|\tilde{a} - a|}{|\tilde{a}|}$ zovemo *relativnom pogreškom* aproksimacije.

Kažemo da je $\tilde{a} \approx a$ s točnošću ε, ako je $|\tilde{a} - a| \leq \varepsilon$.

Neka je $\tilde{a} \approx a$ s točnošću ε_1 i $\tilde{b} \approx b$ s točnošću ε_2, tj. neka je $\tilde{a} = a \pm \Delta a$ i $\tilde{b} = b \pm \Delta b$, pri čemu je $\Delta a \leq \varepsilon_1$ i $\Delta b \leq \varepsilon_2$. Tada je

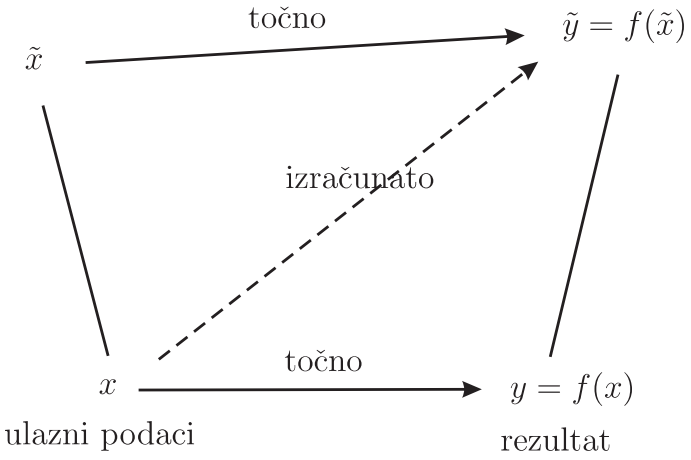
- $\tilde{a} \pm \tilde{b} \approx a \pm b$ s točnošću $\varepsilon_1 + \varepsilon_2$,
- $\tilde{a}\,\tilde{b} \approx a\,b$ s točnošću $\varepsilon_1\,|\tilde{b}| + \varepsilon_2\,|\tilde{a}| + \varepsilon_1\,\varepsilon_2$,
- $\dfrac{\tilde{a}}{\tilde{b}} \approx \dfrac{a}{b}$ s točnošću $\dfrac{\varepsilon_1\,|\tilde{b}| + \varepsilon_2\,|\tilde{a}|}{(|\tilde{b}| - \varepsilon_2)\,|\tilde{b}|}$.

Analizi gornjih svojstava vratit ćemo se nešto kasnije promatrajući ih u okviru analize stabilnosti pojedinih računskih operacija sa strojnim brojevima.

4.1.1 Stabilnost i točnost

Stabilnost pojedinog algoritma mjeri njegovu osjetljivost na pogreške zaokruživanja tijekom računanja. Od sada pa nadalje tijekom naših razmatranja razlikovat ćemo *izračunati rezultat* (to je rezultat koji nam daje računalo) od *točnog rezultata* (to je rezultat koji je matematički ili teorijski točan) promatranog problema.

Definicija 4.1 *Prepostavimo da želimo numerički izračunati vrijednost* $y = f(x)$. *Međutim, algoritam daje rezultat* $\tilde{y} \neq y$, *koji možemo shvatiti kao točnu vrijednost (egzaktnu sliku) promatrane funkcije* $\tilde{y} = f(\tilde{x})$ *ali za neku drugačiju ulaznu vrijednost* $\tilde{x}$ *(drugačiji ulazni podatak). Tada veličinu* $\Delta y = \tilde{y} - y$ *zovemo* pogreška unaprijed, *a* $\Delta x = \tilde{x} - x$ *zovemo* pogreška unazad *ili* povratna pogreška.



Ako $\tilde{x}$ nije jedinstveno određen, tada izabiremo onaj za koji će $\|\Delta x\|$ biti minimalno. U većini slučajeva mi ćemo promatrati *relativnu pogrešku unazad* $\|\Delta x\|/\|x\|$, odnosno *relativnu pogrešku unaprijed* $\|\Delta y\|/\|y\|$.

Računala za prikaz brojeva koriste konačan broj bitova. Stoga ona mogu prikazati samo konačno mnogo brojeva pomoću kojih aproksimiraju realne brojeve. Posljedica takve aproksimacije jest pojava pogreške zaokruživanja. Sa $\mathcal{F}$ označimo skup svih brojeva koji su pohranjeni u računalu (strojni brojevi). Neka je $\mathrm{fl} : \mathbb{R} \to \mathcal{F}$ funkcija zaokruživanja realnih brojeva s najbližim strojnim brojem iz $\mathcal{F}$.

U svrhu naših istraživanja mi ćemo se poslužiti pojednostavljenim modelom računalne aritmetike koji isključuje probleme s brojevima koje ne možemo prikazati jer su preveliki (overflows) ili su premali (underflows).

U nastavku ćemo iznijeti pretpostavke za promatrani model računalne aritmetike.

Pretpostavke. Postoji parameter $\varepsilon_m > 0$ (*strojna preciznost* ili *strojni epsilon*) takav da vrijedi sljedeće:

A1: Za sve $x \in \mathbb{R}$ postoji $\varepsilon \in (-\varepsilon_m, +\varepsilon_m)$ za koji je

$$\mathrm{fl}(x) = x \cdot (1 + \varepsilon).$$

A2: Za svaku od operacija $* \in \{+, -, \cdot, /\}$ i svaki $x, y \in \mathcal{F}$ postoji $\varepsilon \in (-\varepsilon_m, +\varepsilon_m)$ takav da je

$$x \circledast y = (x * y) \cdot (1 + \varepsilon)$$

pri čemu oznaka $\circledast$ označava izračunatu vrijednost za $*$.

Definicija 4.2 *Za algoritam ćemo reći da je* povratno stabilan *ili* stabilan unazad *(engl. backward stable), ako relativna povratna pogreška (relativna pogreška unazad) zadovoljava*

$$\frac{\|\Delta x\|}{\|x\|} = \mathcal{O}(\varepsilon_m).$$

Tipičan način upotrebe gore navedenog koncepta jest postupak u dva koraka koji zovemo *analiza povratne pogreške* ili *povratna analiza pogreške* (engl. backward error analysis). U prvom koraku analize pokazuje se da je promatrani algoritam *povratno stabilan* ili *stabilan unazad*, to jest da posljedice pogreške zaokruživanja možemo prikazati malom perturbacijom Δx ulaznih podataka izvornog problema. U drugom koraku koriste se rezultati, kao što je na primjer teorem 4.2, koji govore o uvjetovanosti problema (ti rezultati ne ovise o vrsti algoritma koji koristimo). Pomoću tih rezultata pokazuje se da je odgovarajuća pogreška unaprijed također mala. Tada smo pokazali, koristeći oba koraka zajedno, da će izračunati rezultat biti blizu točnom (egzaktnom) rezultatu.

Lema 4.1 *Izračunati rezultat oduzimanjem $\ominus$ povratno je stabilan.*

Dokaz. Točan rezultat oduzimanja označimo s $f(x_1, x_2) = x_1 - x_2$, a izračunati rezultat s $\tilde{f}(x_1, x_2) = \mathrm{fl}(x_1) \ominus \mathrm{fl}(x_2)$. Koristeći pretpostavku A1 možemo pisati

$$\mathrm{fl}(x_1) = x_1(1 + \varepsilon_1) \quad \text{i} \quad \mathrm{fl}(x_2) = x_2(1 + \varepsilon_2)$$

pri čemu je $|\varepsilon_1|, |\varepsilon_2| < \varepsilon_m$. Nadalje, koristeći pretpostavku A2 možemo pisati

$$\text{fl}(x_1) \ominus \text{fl}(x_2) = (\text{fl}(x_1) - \text{fl}(x_2))(1 + \varepsilon_3)$$

pri čemu je $|\varepsilon_3| < \varepsilon_m$. Sve zajedno daje:

$$\begin{aligned} \text{fl}(x_1) \ominus \text{fl}(x_2) &= (x_1(1+\varepsilon_1) - x_2(1+\varepsilon_2))(1+\varepsilon_3) \\ &= x_1(1+\varepsilon_1)(1+\varepsilon_3) - x_2(1+\varepsilon_2)(1+\varepsilon_3) \\ &= x_1(1+\varepsilon_4) - x_2(1+\varepsilon_5) \end{aligned}$$

gdje je $\varepsilon_4 = \varepsilon_1 + \varepsilon_3 + \varepsilon_1\varepsilon_3$ i $\varepsilon_5 = \varepsilon_2 + \varepsilon_3 + \varepsilon_2\varepsilon_3$. Oni zadovoljavaju $|\varepsilon_4|, |\varepsilon_5| \leq 2\varepsilon_m + \mathcal{O}(\varepsilon_m^2) = \mathcal{O}(\varepsilon_m)$ za $\varepsilon_m \to 0$.

Stoga za ulaznu pogrešku možemo izabrati

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \tilde{x} = \begin{pmatrix} x_1(1+\varepsilon_4) \\ x_2(1+\varepsilon_5) \end{pmatrix}, \quad \Delta x = \tilde{x} - x\,,$$

što daje $\|\Delta x\|_2 = \sqrt{\varepsilon_4^2 x_1^2 + \varepsilon_5^2 x_2^2} \leq \mathcal{O}(\varepsilon_m)\|x\|_2$. Time je teorem dokazan. □

Napomena 4.1

1) Primijetite da u gornjem dokazu u definiciji povratne pogreške $\tilde{x}$ nije jedinstveno određen. Budući da nas zanima $\tilde{x}$ koji minimizira povratnu pogrešku, možemo izabrati bilo koji $\tilde{x}$ za koji vrijedi $\|\Delta x\|_2 \leq \mathcal{O}(\varepsilon_m)\|x\|_2$. „Pravi" $\tilde{x}$ može biti samo bolji.

2) Slično se može dokazati da su operacije $\oplus, \odot$ i $\oslash$ također povratno stabilne.

3) U većini dokaza povratne stabilnosti mora se analizirati utjecaj pogrešaka zaokruživanja, pa su takvi dokazi zasnovani na pretpostavkama A1 i A2. Budući da su većinom takvi dokazi dugački ali ne i teški, mi ćemo većinu izostaviti u okviru ovog materijala.

4.2 Uvjetovanost

Definicija 4.3 *Za problem koji rješavamo reći ćemo da je* dobro uvjetovan *ako male promjene veličina koje se pojavljuju u problemu dovode do malih promjena u rješenju. S druge strane, reći ćemo da je problem* loše uvjetovan *ako male promjene veličina koje se pojavljuju u problemu dovode do velikih promjena u rješenju.*

Kroz ovo poglavlje $\|\cdot\|$ će nam predstavljati izabranu vektorsku normu ili matričnu normu suglasnu s tom vektorskom normom, što znači da vrijedi

$$\|Ax\| \leq \|A\| \cdot \|x\| \qquad \text{za sve } x \in \mathbb{C}^n,\ A \in \mathbb{C}^{n\times n}.$$

Definicija 4.4 *Uvjetovanost matrice A s oznakom $\kappa(A)$ definira se kao*

$$\kappa(A) = \begin{cases} \|A\|\|A^{-1}\|, & \text{ako je } A \text{ invertibilna;} \\ +\infty, & \text{u suprotnom.} \end{cases}$$

Napomena 4.2 *Uvijek vrijedi da je $\|I\| = \|A\,A^{-1}\| \leq \|A\|\,\|A^{-1}\| = \kappa(A)$, što za inducirane norme povlači $\kappa(A) \geq 1$ za svaki $A \in \mathbb{C}^{n\times n}$.*

Primjer 4.1 *Neka je A realna simetrična i pozitivno definitna matrica sa svojstvenim vrijednostima*

$$\lambda_{\max} = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = \lambda_{\min} > 0.$$

Tada je $\|A\|_2 = \lambda_{\max}$. Budući da inverzna matrica A^{-1} ima svojstvene vrijednosti $1/\lambda_1, \ldots, 1/\lambda_n$, vidimo da je $\|A^{-1}\|_2 = \lambda_{\min}$. Stoga je uvjetovanost matrice A u standardnoj 2-normi (euklidskoj normi) jednaka

$$\kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}}.$$

Lema 4.2 *Neka je $Ax = b$ i $A(x + \Delta x) = b + \Delta b$, za $b \neq 0$. Tada je*

$$\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A)\frac{\|\Delta b\|}{\|b\|}. \tag{4.1}$$

Dokaz. U slučaju da je A singularna, na desnoj strani nejednakosti (4.1) imamo $+\infty$ pa nejednakost svakako vrijedi. U suprotnom (A je regularna) imamo

$$\|b\| = \|A\,x\| \leq \|A\|\,\|x\| \tag{4.2}$$

i

$$A^{-1}\Delta b = (x + \Delta x) - A^{-1}b = x + \Delta x - x = \Delta x.$$

Stoga imamo

$$\frac{\|\Delta x\|}{\|x\|} = \frac{\|A^{-1}\Delta b\|}{\|x\|} \leq \frac{\|A^{-1}\|\|\Delta b\|}{\|x\|} \leq \|A\|\|A^{-1}\|\frac{\|\Delta b\|}{\|b\|},$$

pri čemu je posljednja nejednakost posljedica (4.2). □

Gornja lema nam govori o tome koliko se može promijeniti rješenje jednadžbe $Ax = b$ pri promjeni desne strane. Rezultat pokazuje da je problem rješavanja linearnog sustava $Ax = b$ *dobro uvjetovan* ako je uvjetovanost matrice $\kappa(A)$ mala. Teorem 4.1 koji ćemo iskazati malo kasnije daje sličan rezultat ali za slučaj kad se mijenja matrica A, a ne kao gore desna strana b. Za dokaz tog teorema trebat ćemo sljedeću lemu.

Lema 4.3 *Neka matrica $A \in \mathbb{C}^{n\times n}$ zadovoljava $\|A\| < 1$ za bilo koju induciranu normu. Tada je $I + A$ regularno i*

$$\|(I + A)^{-1}\| \leq (1 - \|A\|)^{-1}.$$

Dokaz. Koristeći nejednakost trokuta imamo

$$\begin{aligned} \|x\| &= \|(I + A)x - Ax\| \\ &\leq \|(I + A)x\| + \| - Ax\| \\ &\leq \|(I + A)x\| + \|A\|\|x\| \end{aligned}$$

i stoga

$$\|(I + A)x\| \geq (1 - \|A\|)\|x\|$$

za svaki $x \in \mathbb{C}^n$. Budući da to povlači $(I + A)x \neq 0$ za svaki $x \neq 0$, vidimo da je matrica $I + A$ regularna.

Pretpostavimo sada da je $b \neq 0$ i $x = (I + A)^{-1}b$. Tada

$$\frac{\|(I + A)^{-1}b\|}{\|b\|} = \frac{\|x\|}{\|(I + A)x\|} \leq \frac{1}{1 - \|A\|}.$$

Budući da je to istinito za sve $b \neq 0$, imamo

$$\|(I + A)^{-1}\| = \max_{b\neq 0} \frac{\|(I + A)^{-1}b\|}{\|b\|} \leq \frac{1}{1 - \|A\|}.$$

Ovime je dokaz završen. □

Teorem 4.1 (uvjetovanost sustava linearnih jednadžbi) *Neka je x rješenje jednadžbi*

$$Ax = b \qquad i \qquad (A + \Delta A)(x + \Delta x) = b.$$

Pretpostavimo da je A regularna tako da je $\|A^{-1}\|\|\Delta A\| < 1$ za neku induciranu matričnu normu. Tada vrijedi

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1-\kappa(A)\frac{\|\Delta A\|}{\|A\|}} \cdot \frac{\|\Delta A\|}{\|A\|}.$$

Dokaz. Uočimo da je

$$(A+\Delta A)\Delta x = b - (A+\Delta A)x = -\Delta A x$$

i stoga je $(I + A^{-1}\Delta A)\Delta x = -A^{-1}\Delta A x$. Koristeći lemu 4.3 možemo pisati

$$\Delta x = -(I + A^{-1}\Delta A)^{-1}A^{-1}\Delta A x$$

pa imamo

$$\|\Delta x\| \leq \|(I + A^{-1}\Delta A)^{-1}\|\|A^{-1}\Delta A\|\|x\| \leq \frac{\|A^{-1}\Delta A\|}{1-\|A^{-1}\Delta A\|}\|x\|.$$

Budući da je

$$\|A^{-1}\Delta A\| \leq \|A^{-1}\|\|\Delta A\| = \kappa(A)\frac{\|\Delta A\|}{\|A\|},$$

i preslikavanje $x \mapsto x/(1-x)$ je rastuće na intervalu $[0,1)$ slijedi

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)\frac{\|\Delta A\|}{\|A\|}}{1-\kappa(A)\frac{\|\Delta A\|}{\|A\|}}.$$

Ovime smo pokazali traženi rezultat. □

Koristeći lemu 4.2, jednostavno se može pokazati da za sustave linearnih jednažbi vrijedi da je relativna pogreška rješenja omeđena umnoškom uvjetovanosti matrice sustava i relativne pogreške u slobodnom vektoru (vektoru desne strane). O tome nam govori sljedeći teorem:

Teorem 4.2 *Za problem rješavanja sustava linearnih jednadžbi (SLJ), pri promjeni vektora na desnoj strani ($b \to b + \Delta b$) vrijedi*

relativna pogreška unaprijed $\leq \kappa(A) \cdot$ *relativna pogreška unazad.*

Dokaz. Primijetite da koristeći oznake koje smo koristili u proučavanju SLJ, relativna pogreška unaprijed dana je sa $\|\Delta x\|/\|x\|$ (pogreška u rezultatu), dok je relativna pogreška unazad dana sa $\|\Delta b\|/\|b\|$ (pogreška u ulaznim podacima odnosno desna strana linearnog sustava). Sada je jasno da je teorem izravna posljedica leme 4.2. □

4.2.1 Zadaci

1. Izaberite neku matričnu normu i izračunajte uvjetovanost matrice

$$A = \begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}$$

za tu normu.

2. Neka je x rješenje za $Ax = b$ i neka je $\tilde{x}$ rješenje za $(A+\Delta A)\tilde{x} = b+\Delta b$. Pokažite da vrijedi

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}} \cdot \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

3. Pokažite da su aritmetičke operacije $\oplus$, $\odot$ i $\oslash$ povratno stabilne.

4. Odredite $\kappa_2(A)$, $\kappa_F(A)$, $\kappa_1(A)$ i $\kappa_\infty(A)$ ako je

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 0 & -3 & 3 \\ 0 & 3 & 5 \end{pmatrix}.$$

[Rješenje: $\kappa_2(A) = 3$, $\kappa_F(A) = 4.3653$, $\kappa_1(A) = 4$, $\kappa_\infty(A) = 4$.]

5. Rješenje sustava linearnih jednadžbi, gdje su matrica sustava A i vektor b zadani s

$$A = \begin{pmatrix} 0.434 & 0.26 \\ 0.79 & 0.473 \end{pmatrix}, \quad b = \begin{pmatrix} 0.694 \\ 1.263 \end{pmatrix} \quad \text{iznosi} \quad x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Promijenimo vektor b za $\Delta b = \begin{pmatrix} 0.00006 \\ 0.000003 \end{pmatrix}$. Odredite rješenje sustava $A\tilde{x} = b + \Delta b$ i izračunajte relativne pogreške vektora b i x. Koliko je puta relativna pogreška u rješenju veća od relativne pogreške u vektoru b u $\|\cdot\|_\infty$? Izračunajte $\kappa_\infty(A)$.

[Rješenje: Relativna pogreška rješenja je više od 822 puta veća od relativne pogreške u vektoru b, $\kappa_\infty(A) = 13101$.]

6. U prethodnom zadatku načinimo promjenu u elementu a_{11} tako da je $\Delta A = \begin{pmatrix} -0.001 & 0 \\ 0 & 0 \end{pmatrix}$. Odredite rješenje sustava $(A + \Delta A)\tilde{x} = b$. Koliko puta je relativna pogreška u rješenju x veća od relativne pogreške u matrici?

   [Rješenje: Relativna pogreška u rješenju x veća je od relativne pogreške u matrici 1688 puta.]

Poglavlje 5

Sustavi linearnih jednadžbi

U okviru ovog poglavlja proučavat će se sljedeći problem: neka je dana matrica $A \in \mathbb{C}^{n\times n}$ i vektor $b \in \mathbb{C}^n$, treba odrediti vektor $x \in \mathbb{C}^n$ takav da vrijedi $Ax = b$. Taj problem odgovara matričnom zapisu sustava linearnih jednadžbi i kraće ćemo ga zapisivati sa (SLJ).

Proučavat ćemo nekoliko metoda za njegovo rješavanje. Jedna od uobičajenih ideja bi se ukratko mogla opisati na sljedeći način: prikazati matricu A pomoću jednostavnijih matrica M i N, tako da je $A = MN$, zatim riješiti prvo (jednostavniji linearni sustav) $My = b$ i nakon toga $Nx = y$. Na taj smo način dobili vektor x koji rješava početni (SLJ), zaista vidimo da je $Ax = MNx = My = b$.

5.1 Gaussove eliminacije

Metoda Gaussovih eliminacija je svakako najstariji, najjednostavniji i najpoznatiji algoritam za rješavanje sustava linearnih jednadžbi $Ax = b$. Ilustrirajmo osnovnu ideju Gaussovih eliminacija na sljedećem jednostavnom primjeru. Kako bismo riješili sustav

$$\begin{aligned} 2\,x_1 - x_2 &= 1 \\ -x_1 + 2\,x_2 &= 1 \end{aligned}$$

dovoljno je primijetiti da zbog prve jednadžbe vrijedi $x_1 = \frac{1}{2}(1 + x_2)$, pa je druga jednadžba

$$\underbrace{-\frac{1}{2}(1 + x_2)}_{x_1} + 2\,x_2 = 1\,,\ \text{to jest}\ ,x_2 = 1\,,$$

odakle je $x_1 = 1$. Kažemo da smo x_1 eliminirali iz druge jednadžbe.

Ovu ideju možemo lako generalizirati na dimenziju $n > 1$, gdje sustavno eliminiramo neke nepoznanice iz nekih jednadžbi. Pokazuje se da takav algoritam ima dosta zanimljivu strukturu i da ga se može ekvivalentno zapisati u terminima matričnih operacija. Kvalitativno novi moment u analizi metode eliminacija nastaje kad sam proces eliminacija interpretiramo kao faktorizaciju matrice sustava A na produkt trokutastih matrica.

Sljedeći će teorem sadržavati teorijsku osnovu za takav algoritam, ali prije nego ga iskažemo, uvest ćemo neke važne pojmove.

Definicija 5.1 *Za matricu $A \in \mathbb{C}^{n\times m}$ kažemo da je* gornjetrokutasta *ako je $a_{ij} = 0$ za sve $i > j$ i* donjetrokutasta *ako $a_{ij} = 0$ za sve $i < j$.*

Definicija 5.2 *Za matricu $A^j \in \mathbb{C}^{j\times j}$ kažemo da je j-ta* glavna podmatrica *matrice $A \in \mathbb{C}^{n\times n}$, ako za njene elemente vrijedi $(A^j)_{kl} = a_{kl}$ za $1 \le k, l \le j$.*

Teorem 5.1 (LU faktorizacija) *a) Neka je $A \in \mathbb{C}^{n\times n}$ takva da su joj sve j-te* glavne podmatrice *A^j regularne za $j = 1, \ldots, n$. Tada postoji jedinstvena dekompozicija matrice A takva da je $A = LU$, gdje je L donja trokutasta matrica s jedinicama na glavnoj dijagonali, a U je regularna gornje trokutasta matrica.*

b) Ako je A^j singularna za neko $j \in \{1, \ldots, n\}$, tada ne postoji takva dekompozicija.

Prije nego dokažemo teorem ilustrirat ćemo oblik LU faktorizacije za matricu A dimenzije 3×3:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

Dokaz. a) Za dokaz prve tvrdnje koristit ćemo indukciju. Ako je $n = 1$, tada imamo trivijalan slučaj. Matrica $L = (1) \in \mathbb{C}^{1\times 1}$, a $U = (a_{11}) \in \mathbb{C}^{1\times 1}$.

Očito je $A = LU$, budući da L mora biti 1×1 donja trokutasta matrica s jedinicama na dijagonali, to je ova dekompozicija jedinstvena.

Neka je sada $n > 1$. Pretpostavimo da za $A \in \mathbb{C}^{(n-1)\times(n-1)}$ postoji jedinstvena dekompozicija $A = LU$, pri čemu su sve glavne podmatrice A^j regularne za $j = 1, \ldots, n-1$.

Ostaje nam pokazati da tvrdnja vrijedi za matricu $A \in \mathbb{C}^{n\times n}$. Stoga ćemo matricu $A \in \mathbb{C}^{n\times n}$ zapisati u obliku:

$$A = \begin{pmatrix} A^{n-1} & b \\ c^* & a_{nn} \end{pmatrix}, \tag{5.1}$$

gdje je A^{n-1} $n-1$-va glavna podmatrica matrice A, a preostali blokovi su $b, c \in \mathbb{C}^{n-1}$ i $a_{nn} \in \mathbb{C}$. Mi želimo odrediti dekompoziciju oblika:

$$A = \begin{pmatrix} L_1 & 0 \\ l^* & 1 \end{pmatrix} \begin{pmatrix} U_1 & u \\ 0 & \eta \end{pmatrix} = \begin{pmatrix} L_1U_1 & L_1u \\ l^*U_1 & l^*u + \eta \end{pmatrix}, \tag{5.2}$$

gdje je $L_1 \in \mathbb{C}^{(n-1)\times(n-1)}$ donja trokutasta s jedinicama na glavnoj dijagonali, $U_1 \in \mathbb{C}^{(n-1)\times(n-1)}$ je gornja trokutasta, a $l, u \in \mathbb{C}^{n-1}$ i $\eta \in \mathbb{C}$.

Uspoređivanjem blokova iz (5.1) i (5.2) slijedi:

Po pretpostavci indukcije postoje matrice donje trokutasta L_1 i gornja trokutasta U_1 takve da je $A^{n-1} = L_1U_1$ i ta je dekompozicija jedinstvena. Budući da je L_1 regularna, to znači da je vektor u jedinstveno određen iz uvjeta $L_1u = b$. Nadalje, zbog regularnosti matrice U_1, vektor l koji zadovoljava jednakost $l^*U_1 = c^*$ (vidimo da je $l^* = c^*U_1^{-1}$, odnosno $l = U_1^{-*}c$). Konačno iz uvjeta $l^*u + \eta = a_{nn}$ slijedi da je $0 \neq \eta \in \mathbb{C}$ jedinstveno određeno. Činjenica da je $\eta \neq 0$ slijedi iz pretpostavke teorema o regularnosti matrice A, tj. $\det(A) = 1 \cdot \eta \cdot \det(U_1) \neq 0$, povlači $\eta \neq 0$. To znači da je matrica

$$U = \begin{pmatrix} U_1 & u \\ 0 & \eta \end{pmatrix}$$

regularna.

b) Pretpostavimo da je dana LU faktorizacija matrice A. Tada za po volji izabrano $j \in \{1, \ldots, n\}$ možemo pisati

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix} = \begin{pmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{pmatrix},$$

gdje su $A_{11}, L_{11}, U_{11} \in \mathbb{C}^{j\times j}$. Iz

$$\det(A^j) = \det(A_{11}) = \det(L_{11}U_{11}) = \det(L_{11}) \cdot \det(U_{11}) = 1 \cdot \det(U_{11}) \neq 0\,,$$

vidimo da za sve $j \in \{1, \ldots, n\}$ slijedi da je A^j regularna, što je kontradikcija s pretpostavkom. □

Primjer 5.1 *Može se pokazati da se matrica A može transformirati u gornju trokutastu množenjem s lijeva donjim trokutastim matricama. To ćemo ilustrirati sljedećim primjerom. Neka je*

$$A = \begin{pmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ -15 & 5 & -9 \end{pmatrix}.$$

Gornju trokutastu formu postići ćemo množenjem matrice A s lijeva donjim trokutastim matricama na sljedeći način:

$$L_1 A = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ -15 & 5 & -9 \end{pmatrix} = \begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 8 & 3 \end{pmatrix}.$$

Sada ponovno množimo donjom trokutastom matricom s lijeva dobivenu matricu $L_1 A$

$$L_2(L_1 A) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{pmatrix} \begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 8 & 3 \end{pmatrix} = \begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 0 & 7 \end{pmatrix}$$

Budući da je produkt donjih trokutastih matrica opet donja trokutasta matrica (provjerite ovu tvrdnju), dobili smo donju trokutastu matricu $\hat{L} = L_2 L_1$ za koju vrijedi da je $\hat{L}A = U$.

Iz gornjeg primjera se vidi da se matrica A može prikazati kao $A = L^{-1}U = L_1^{-1}L_2^{-1}U$. Sljedeća lema objašnjava kako se može izračunavati inverz donje trokutaste matrice. No prije nego iskažemo tu lemu, primijetimo da matrice L_1 i L_2 iz primjera 5.1 možemo zapisati u obliku:

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} = I + u_1 e_1^T, \quad u_1 = \begin{pmatrix} 0 \\ -2 \\ 3 \end{pmatrix}, e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{pmatrix} = I + u_2 e_2^T, \quad u_2 = \begin{pmatrix} 0 \\ 0 \\ -4 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

gdje je I jedinična matrica odgovarajuće dimenzije.

Lema 5.1 *a) Neka je $L = (l_{ij})$ donjetrokutasta matrica s jedinicama na dijagonali s elementima različitim od nule ispod dijagonale samo u k-tom stupcu, drugim riječima neka je*

$$L_k = I + u_k\, e_k^T\,, \quad u_k = \begin{bmatrix} 0 & \dots & 0 & m_{k+1\,k} & \dots & m_{n\,k} \end{bmatrix}^T,$$

i neka je e_k k-ti stupac jedinične matrice I. Tada je L_k^{-1} također donjetrokutasta matrica s jedinicama na dijagonali s elementima različitim od nule ispod dijagonale samo u k-tom stupcu i vrijedi:

$$L_k^{-1} = I - u_k\, e_k^T\,.$$

b) Produkt dviju donjetrokutastih matrica s jedinicama na dijagonali opet je donjetrokutasta matrica s jedinicama na dijagonali.

Dokaz. a) Jednostavnim množenjem lako se provjeri da je

$$L_k L_k^{-1} = \left(I + u_k\, e_k^T\right) \cdot \left(I - u_k\, e_k^T\right) = I\,,$$

pri čemu se iskoristi činjenica da je $e_k^T u_k = 0$.

b) Ovo se jednostavno pokazuje izravno množenjem. □

Primjer 5.2 *Za matrice L_1 i L_2 iz primjera 5.1 dobivamo*

$$(L_2 L_1)^{-1} = L_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 4 & 1 \end{pmatrix},$$

tako da je LU faktorizacija matrice A iz primjera 5.1 dana sa

$$\begin{pmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ -15 & 5 & -9 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 4 & 1 \end{pmatrix} \begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 0 & 7 \end{pmatrix}.$$

Prije nego zapišemo algoritam za LU faktorizaciju kvadratne matrice reda n, prisjetimo se Gaussovog postupaka eliminacije za slučaj n lineranih jednadžbi s n nepoznanica. U tom slučaju sustav linearnih jednadžbi možemo zapisati kao:

$$\begin{aligned} a_{11}\, x_1 + a_{12}\, x_2 + \dots + a_{1n}\, x_n &= b_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + \dots + a_{2n}\, x_n &= b_2 \\ &\dots \\ a_{n1}\, x_1 + a_{n2}\, x_2 + \dots + a_{nn}\, x_n &= b_n\,. \end{aligned}$$

Pretpostavimo da je matrica sustava $A = (a_{ij})$ takva da je $\det(A^j) \neq 0$ za $j = 1, \ldots, n$. To znači da je element $a_{11} \neq 0$. Prvo primjenom Gaussovih transformacija matricu A svodimo na

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

gdje je

$$a_{ik}^{(2)} = a_{ik} - m_{i1}\, a_{1k},\ b_i^{(2)} = b_i - m_{i1} b_1, \qquad i, k = 2, \ldots, n\,,$$

a $m_{i1} = \dfrac{a_{i1}}{a_{11}}$. Koristeći oznake leme 5.1 gornji postupak možemo zapisati kao

$$A_1 = L_1 A \quad \text{gdje su} \quad L_1 = I + u_1\, e_1^T\,, \quad \text{i} \quad u_1 = \begin{bmatrix} 0 & m_{2\,1} & \ldots & m_{n\,1} \end{bmatrix}^T,$$

$$A_1 = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}.$$

Nadalje, prema teoremu 5.1 slijedi da će $a_{22}^{(2)} \neq 0$, stoga slično kao gore dobijamo

$$\begin{aligned} a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 + \cdots + a_{1n}\, x_n &= b_1 \\ a_{22}^{(2)}\, x_2 + a_{23}^{(2)}\, x_3 + \cdots + a_{2n}^{(2)}\, x_n &= b_2^{(2)} \\ a_{33}^{(3)}\, x_3 + \cdots + a_{3n}^{(3)}\, x_n &= b_3^{(3)} \\ &\cdots \\ a_{n3}^{(3)}\, x_3 + \cdots + a_{nn}^{(3)}\, x_n &= b_n^{(3)}\,, \end{aligned}$$

gdje je

$$a_{ik}^{(3)} = a_{ik}^{(2)} - m_{i2}\, a_{2k}^{(2)},\ b_i^{(3)} = b_i^{(2)} - m_{i2} b_2^{(2)}, \qquad i, k = 3, \ldots, n\,,$$

i $m_{i2} = \dfrac{a_{i2}^{(2)}}{a_{22}^{(2)}}$. Slično kao i u prošlom koraku koristeći oznake leme 5.1 ovaj korak možemo zapisati kao

$$A_2 = L_2 A_1 = L_2 L_1 A \quad \text{gdje su} \quad L_2 = I + u_2 e_2^T \ \text{ i } \ u_2 = \begin{bmatrix} 0 & 0 & m_{3\,2} & \dots & m_{n2} \end{bmatrix}^T,$$

$$A_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ & & \vdots & \vdots & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{bmatrix}.$$

Gore navedenu rutinu možemo zapisati u matričnom obliku, pri čemu se matrica A preoblikuje u gornju trokutastu ($A_{n-1} \equiv U$) množenjem s donje trokutastim matricama s lijeva. Taj je postupak dan sljedećim algoritmom:

Algoritam LU (LU faktorizacija).
ulaz: $A \in \mathbb{C}^{n\times n}$, takva da $\det(A^j) \neq 0$ za $j = 1, \dots, n$
izlaz: $L, U \in \mathbb{C}^{n\times n}$, gdje je $A = LU$ tražena LU faktorizacija matrice A

1: $U = A, L = I$
2: **for** $k = 1, \dots, n-1$ **do**
3: **for** $j = k+1, \dots, n$ **do**
4: $l_{jk} = u_{jk}/u_{kk}$
5: $(u_{j\,k}, \dots, u_{j\,n}) = (u_{j\,k}, \dots, u_{j\,n}) - l_{j\,k}(u_{k\,k}, \dots, u_{k\,n})$
6: **end for**
7: **end for**

Napomena 5.1 *Primijetimo da u gornjem algoritmu 5. red odgovara oduzimanju k-tog retka pomnoženog s brojem $l_{jk} = u_{jk}/u_{kk}$ od j-tog retka, što je standardni korak Gaussovih eliminacija, a to se vidi i iz činjenice da su l_{jk} upravo jednaki brojevima m_{jk} iz gore navedenih koraka Gaussovih eliminacija. Na taj način će biti $u_{jk} = 0$, za $j = k+1, \dots, n$, pri čemu se retci $1, \dots, k$ neće promijeniti.*

To točno odgovara produktu matrice A s donjetrokutastom matricom L_k s lijeva, kao što smo ilustrirali u prethodnom primjeru. Stoga, nakon što završimo sve operacije u petlji koja završava u 6. retku, matrica U poprima vrijednost $L_k \cdots L_1 A$ i ima nule ispod glavne dijagonale u stupcima $1, \dots, k$.

Budući da su sve glavne podmatrice A^j regularne i matrica L_j je donje trokutasta s jedinicama na glavnoj dijagonali, to znači da imamo

$$det(L_k \cdots L_1 A^{k+1}) = det\, A^{k+1} \neq 0$$

i stoga je $u_{kk} \neq 0$ u 4. retku. Lema 5.1 pokazuje da algoritam ispravno izračunava vrijednosti elemenata l_{jk} matrice $L = (L_n \cdots L_1)^{-1}$.

Za kompletiranje rješavanja linearnih sustava Gaussovom metodom eliminacija, potrebno je još konstruirati algoritam za rješavanje trokutastih sustava, tj. u ovom slučaju za rješavanje sustava s gornjom trokutastom matricom.

Algoritam PS (povratnih supstitucija).
ulaz: $U \in \mathbb{C}^{n\times n}$ regularna gornje trokutasta matrica i vektor $b \in \mathbb{C}^n$
izlaz: $x \in \mathbb{C}^n$ koji zadovoljava $Ux = b$

1: $x_n = \dfrac{b_n}{u_{nn}}$;
2: **for** $j = n-1, \ldots, 1$ **do**
3: $\qquad x_j = \dfrac{1}{u_{jj}} \left(b_j - \displaystyle\sum_{k=j+1}^{n} u_{jk} x_k \right)$
4: **end for**

Engleski naziv za *povratne supstitucije* (*supstitucije unazad*) glasi *backward substitution.*

Napomena 5.2 *Budući da je U gornje trokutasta, vidimo da je*

$$\begin{aligned} (Ux)_i &= \sum_{j=1}^{n} u_{ij} x_j \\ &= u_{ii} \left(\frac{1}{u_{ii}} (b_i - \sum_{k=i+1}^{n} u_{ik} x_k) \right) + \sum_{k=i+1}^{n} u_{ik} x_k \\ &= b_i. \end{aligned}$$

*Stoga možemo zaključiti da je **Algoritam PS** ispravan.*

Odgovarajući algoritam za rješavanje sustava $Lx = b$, pri čemu je L donje trokutasta matrica, nazivamo *supstitucijama unaprijed* (engl. forward substitution).

Algoritam SU (supstitucija unaprijed).
ulaz: $L \in \mathbb{C}^{n\times n}$ regularna donje trokutasta matrica i vektor $b \in \mathbb{C}^n$
izlaz: $x \in \mathbb{C}^n$ koji zadovoljava $Lx = b$

1: $x_1 = b_1/l_{11}$;
2: **for** $j = 2, \ldots, n$ **do**
3: $$x_j = \frac{1}{l_{jj}} \cdot \left(b_j - \sum_{k=1}^{j-1} l_{jk} x_k \right) ;$$
4: **end for**

Uzimajući u obzir sve do sada izneseno, odgovarajući algoritam za metodu Gaussovih eliminacija za rješavanje sustava linearnih jednadžbi možemo pisati:

Algoritam GE (Gaussove eliminacije).
ulaz: $A \in \mathbb{C}^{n\times n}$, gdje je $det(A^j) \neq 0$ za $j = 1,\ldots,n$
izlaz: $x \in \mathbb{C}^n$, takav da $Ax = b$

1: Odrediti LU faktorizaciju matrice A.
2: Riješiti $Ly = b$ pomoću supstitucija unaprijed.
3: Riješiti $Ux = y$ pomoću povratnih supstitucija.

Dobiveni rezultat je vektor $x \in \mathbb{C}^n$ za koji vrijedi $Ax = LUx = Ly = b$, stoga vidimo da gornji algoritam daje dobar rezultat.

5.1.1 Zadaci

1. Odredite LU faktorizaciju matrice A, gdje je

$$A = \begin{pmatrix} 4 & 6 & 7 \\ -4 & -11 & -3 \\ 16 & -1 & 50 \end{pmatrix}.$$

$$[\text{ Rješenje: } U = \begin{pmatrix} 4 & 6 & 7 \\ 0 & -5 & 4 \\ 0 & 0 & 2 \end{pmatrix}, L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 4 & 5 & 1 \end{pmatrix}.]$$

2. Odredite LU faktorizaciju matrice A, ako je

$$A = \begin{pmatrix} 2 & 1 & -3 & 4 \\ -4 & 1 & 5 & -6 \\ 6 & 9 & -10 & 17 \\ 8 & 19 & -14 & 32 \end{pmatrix}.$$

[Rješenje: $U = \begin{pmatrix} 2 & 1 & -3 & 4 \\ 0 & 3 & -1 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 3 \end{pmatrix}$, $L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 4 & 5 & 3 & 1 \end{pmatrix}$.]

3. Odredite LU faktorizaciju matrice A i $\det A$ ako je

$$A = \begin{pmatrix} -2 & 4 & 3 & 5 \\ -8 & 13 & 13 & 24 \\ -4 & 14 & 8 & 4 \\ -2 & -20 & 27 & 47 \end{pmatrix}.$$

[Rješenje: $\det A = 48$.]

4. Odredite LU faktorizaciju matrice A i riješite sustav $Ax = b$, ako je

$$A = \begin{pmatrix} -1 & 4 & 4 & -6 \\ 3 & -14 & -11 & 15 \\ 1 & 2 & -10 & 13 \\ -1 & 0 & -9 & -21 \end{pmatrix}, \quad b = \begin{pmatrix} 13 \\ -25 \\ -52 \\ 53 \end{pmatrix}.$$

[Rješenje: $x = \begin{pmatrix} 1 & -2 & 1 & -3 \end{pmatrix}^T$]

5. Neka je $A \in \mathbb{R}^{n \times n}$ za $n = 2^k$. Primijetimo da se LU faktorizacija matrice A može zapisati u obliku

$$A = LU = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix}.$$

Konstruirajte algoritam na osnovi strategije *podijeli i osvoji* (*divide and conquer*) kojemu će za izračunavanje LU-faktorizacije matrice A biti potreban red veličine $\mathcal{O}(n^\alpha)$ operacija, pri čemu je $\mathcal{O}(n^\alpha)$ broj operacija potrebnih za matrična množenja.

5.2 Numerička svojstva Gaussovih eliminacija

U ovom ćemo poglavlju prikazati neka od osnovnih numeričkih svojstava Gaussovih eliminacija, kao što su (*ne*)*stabilnost* ili *efikasnost*. Sljedeća lema sadrži rezultat koji govori o potrebnom broju računskih operacija za LU faktorizaciju (*efikasnost*).

Lema 5.2 *Algoritam za LU faktorizaciju kvadratne matrice reda n ima složenost računanja:*

$$C(n) = \frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n\,.$$

Dokaz. Trebamo odrediti broj računskih operacija potrebnih za izvođenje LU faktorizacije. U 5. redu **Algoritma LU**, imamo $(n-k+1)$ produkt i $(n-k+1)$ oduzimanja, što znači ukupno $2(n-k+1)$ operacija. U 4. redu imamo jedno dijeljenje. Stoga se u petlji koja počinje u 3. redu ukupno izvodi $(n-k)(1+2(n-k+1))$ operacija. Uzimajući u obzir vanjsku petlju, ukupan broj operacija iznosi

$$C(n) = \sum_{k=1}^{n-1}(n-k)\,(1+2(n-k+1)) = \sum_{k=1}^{n-1} 2(n-k)^2 + 3(n-k)\,.$$

Tvrdnja teorema slijedi iz činjenice da je

$$\sum_{k=1}^{n-1} 2(n-k)^2 + 3(n-k) = \frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n\,,$$

koju lako dokazujemo matematičkom indukcijom. □

Zadatak 5.1 *Matematičkom indukcijom dokažite da vrijedi*

$$\sum_{k=1}^{n-1} 2(n-k)^2 + 3(n-k) = \frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n\,.$$

Pomoć: $\sum_{k=1}^{n-1} k^2 = \frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6}$.

Napomena 5.3 *Za funkcije $f, g : \mathbb{N} \to \mathbb{N}$ ili $f, g : \mathbb{R}^+ \to \mathbb{R}^+$ pišemo*

$$f(x) \sim g(x) \qquad \text{za } x \to \infty$$

ako $\lim_{x\to\infty} f(x)/g(x) = 1$. *Ovo povlači da je* $f(x) = \Theta\Big(g(x)\Big)$. *Međutim, ovo je strožiji pristup jer se na ovakav način zahtijeva da se pri zapisivanju „glavnog dijela" broja operacija zadrži i vodeća konstanta. Koristeći ovaj način označavanja vidimo da je* $C(n) \sim \frac{2}{3}n^3$.

Lema 5.3 *Računska složenost supstitucija unaprijed i supstitucija unazad iznosi* $C(n) \sim n^2$.

Dokaz. Za izračunavanje x_j pomoću supstitucija unazad, potrebno je $2(n-j)+1$ operacija. Stoga je ukupan broj operacija potrebnih za izračunavanje rješenja linearnog sustava dan sa

$$C(n) = \sum_{j=1}^{n} \Big(2(n-j)+1\Big) = 2\sum_{k=0}^{n-1} k + n = n(n-1)+n = n^2.$$

Slično se može pokazati da je računska složenost supstitucija unaprijed dana sa $C(n) \sim n^2$. □

Teorem 5.2 (Računska složenost Gaussovih eliminacija) *Asimptotski red veličine računske složenosti Gaussovih eliminacija iznosi:*

$$C(n) \sim \frac{2}{3}n^3\,.$$

Dokaz. Dokaz je jednostavna posljedica prethodne dvije leme. □

5.2.1 Analiza pogreške Gaussovih eliminacija

Teorem 5.3 *Algoritam supstitucije unazad je povratno stabilan (backward stable), odnosno izračunato rješenje $\tilde{x}$ zadovoljava $(U+\Delta U)\tilde{x} = b$ za gornju trokutastu matricu $\Delta U \in \mathcal{C}^{n\times n}$ za koju vrijedi*

$$\frac{\|\Delta U\|}{\|U\|} = \mathcal{O}(\varepsilon_m).$$

Dokaz gornjeg teorema je dugačak i temelji se na pretpostavkama (A1) i (A2) aritmetike računala i mi ćemo ga izostaviti.

Napomena 5.4 *Gornji teorem možemo iskazati točnije: izračunato rješenje $\tilde{x}$ zadovoljava $(U+\Delta U)\tilde{x} = b$ za gornju trokutastu matricu $\Delta U \in \mathcal{C}^{n\times n}$ za koju vrijedi*

$$\frac{\|\Delta U\|}{\|U\|} = \frac{n\,\varepsilon_m}{1-n\,\varepsilon_m}\,,$$

gdje je parameter $\varepsilon_m > 0$ (*strojna preciznost* (za detalje vidi [3]).

Sada koristeći teorem 4.1 koji govori o uvjetovanosti SLJ dobivamo gornju ocjenu za pogrešku izračunatog rezultata pomoću supstitucije unazad i ona glasi:

$$\frac{\|\tilde{x}-x\|}{\|x\|} \leq \frac{\kappa(U)}{1-\kappa(U)\dfrac{\|\Delta U\|}{\|U\|}} \cdot \frac{\|\Delta U\|}{\|U\|} = \kappa(U)\mathcal{O}(\varepsilon_m).$$

Stoga možemo zaključiti da je dio Gaussovih eliminacija, koji se sastoji od supstitucija unazad, numerički stabilan i ne dovodi to dodatnih poteškoća. Isto vrijedi i za supstitucije unaprijed.

Problem. Na sljedećem ćemo jednostavnom primjeru ilustrirati da LU-factorizacija nije povratno stabilna. Neka je

$$A = \begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}$$

za neko $\varepsilon > 0$. Tada za A postoji LU faktorizacija oblika $A = LU$, gdje su

$$L = \begin{pmatrix} 1 & 0 \\ \varepsilon^{-1} & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} \varepsilon & 1 \\ 0 & 1-\varepsilon^{-1} \end{pmatrix}.$$

Pretpostavimo sada da je $\varepsilon \ll 1$. Tada je ε^{-1} vrlo veliki broj, pa će pri spremanju tog broja u računalo doći do pogreške zaokruživanja. Matrice mogu biti spremljene kao

$$\tilde{L} = \begin{pmatrix} 1 & 0 \\ \varepsilon^{-1} & 1 \end{pmatrix}, \qquad \tilde{U} = \begin{pmatrix} \varepsilon & 1 \\ 0 & -\varepsilon^{-1} \end{pmatrix},$$

što je u skladu s pretpostavkom (A1) o pogreškama zaokruživanja. Vidimo da vrijedi $\tilde{L} = L$ i $\tilde{U} \approx U$. Međutim, množenjem dviju matrica sa zaokruženim brojevima dobivamo

$$\tilde{L}\tilde{U} = \begin{pmatrix} \varepsilon & 1 \\ 1 & 0 \end{pmatrix} = A + \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}.$$

Vidimo da je mala pogreška pri zaokruživanju broja dovela do velike pogreške u rezultatu! Ovaj jednostavni primjer pokazuje da analiza Gaussovih eliminacija općenito pokazuje da perturbirani problem ne mora biti blizu originalnom (polaznom) problemu.

Primijetimo da gore promatrani problem nema veze s uvjetovanošću matrice A. Uistinu, imamo

$$A^{-1} = (1-\varepsilon)^{-1} \begin{pmatrix} -1 & 1 \\ 1 & -\varepsilon \end{pmatrix}$$

i stoga je $\kappa(A) = \|A\|_\infty \|A^{-1}\|_\infty \approx 4$ za malo $\varepsilon > 0$, što znači da je matrica A dobro uvjetovana.

Zbog ovih nestabilnosti standardne (klasične) metode Gaussovih eliminacija, ova se metoda u takvom obliku izbjegava u numeričkim primjenama. U sljedećem ćemo poglavlju pokazati jednu njezinu modifikaciju koja nema navedeni problem sa stabilnošću.

5.3 Gaussove eliminacije s djelomičnim pivotiranjem

Definicija 5.3 ***Matrica permutacije*** *je matrica kojoj su elementi nule ili jedinice i koja u svakom retku i u svakom stupcu ima točno jednu jedinicu.*

Iz definicije odmah slijedi da je svaka matrica permutacije kvadratna. Primjetimo da je svaka matrica permutacije regularna.

Primjer 5.3 *Neka je* $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ *permutacija. Tada je matrica* $P = (p_{ij})$

$$p_{ij} = \begin{cases} 1, & \text{ako } j = \pi(i) \\ 0, & \text{u suprotnom} \end{cases}$$

matrica permutacija. Istaknimo da je jedinična matrica također permutacijska matrica.

Napomena 5.5

1) Ako je P permutacijska matrica, tada je

$$(P^T P)_{ij} = \sum_{k=1}^{n} p_{ki}\, p_{kj} = \delta_{ij},$$

stoga je $P^T P = I$. Ovim smo pokazali da je matrica permutacije ortogonalna i da je $P^{-1} = P^T$.

2) Ako je P matrica permutacija koja odgovara permutaciji π, tada za njezin inverz vrijedi da je $(P^{-1})_{ij} = 1$ ako i samo ako je $j = \pi^{-1}(i)$. Stoga matrica permutacija P^{-1} odgovara permutaciji π^{-1}.

3) Vrijedi sljedeće:

$$(PA)_{ij} = \sum_{k=1}^{n} p_{ik}\, a_{kj} = a_{\pi(i),j},$$

(posljednja jednakost slijedi iz $p_{ik} = 1$ za $k = \pi(i)$) za sve $i, j \in \{1, \ldots, n\}$, što pokazuje da množenje s matricom permutacija slijeva odgovara permutaciji (zamjeni mjesta) redaka matrice A.

Nadalje, vrijedi

$$(AP)_{ij} = \sum_{k=1}^{n} a_{ik}\, p_{kj} = a_{i,\pi^{-1}(j)}$$

za sve $i, j \in \{1, \ldots, n\}$, odnosno množenje s matricom permutacija zdesna odgovara permutaciji (zamjeni mjesta) stupaca matrice A.

U prošlom poglavlju prikazan je primjer koji obrađuje slučaj kad LU faktorizacija nije povratno stabilna. Problem sa stabilnošću nastaje zbog toga što u 4. koraku LU faktorizacije dolazi do dijeljenja s vrlo malim brojem $u_{kk} = \varepsilon$.

Nadalje, primijetimo da za postojanje LU faktorizacije matrica A mora imati specijalnu strukturu: sve njene glavne podmatrice do uključivo reda $n - 1$ moraju biti regularne. Sljedeći primjer ilustrira taj problem.

Primjer 5.4 *Neka je*

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

matrica sustava $Ax = b$. Očito je regularna ($\det A = -1$), pa sustav ima rješenje, ali za A ne postoji LU faktorizacija. Zaista, kad bi LU faktorizacija postojala, vrijedilo bi:

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_{21} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{pmatrix}$$

što povlači

$$\begin{aligned} 1 \cdot u_{11} &= 0 \\ 1 \cdot u_{12} &= 1 \\ l_{21} \cdot u_{11} &= 1 \\ l_{21} \cdot u_{12} + u_{22} &= 1 \end{aligned}$$

Iz prve jednadžbe odmah vidimo da je $u_{11} = 0$, a iz treće slijedi da $l_{21} \cdot 0 = 1$, što je u kontradikciji s regularnošću matrice A.

S druge strane, sustav $Ax = b$ je linearni sustav

$$\begin{aligned} 0\,x_1 + x_2 &= b_1 \\ x_1 + x_2 &= b_2 \end{aligned}$$

čija su rješenja dana sa $x_1 = b_2 - b_1$ i $x_2 = b_1$. Primijetimo da taj sustav možemo ekvivalentno zapisati kao

$$\begin{aligned} x_1 + x_2 &= b_2 \\ 0\,x_1 + x_2 &= b_1\,, \end{aligned}$$

pri čemu promatranom sustavu sada pripada matrica:

$$\tilde{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Vidimo da matricu $\tilde{A}$ jednostavno dobivamo iz matrice A zamjenom redaka, odnosno $\tilde{A} = PA$, pri čemu je p **matrica permutacije** oblika

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Dakle, kako bismo izbjegli probleme kao što su dijeljenje s jako malim brojevima u LU faktorizaciji ili nemogućnost izračunavanja LU faktorizacije, mi ćemo se poslužiti zamjenom redaka matrice A. Može se pokazati da za proizvoljnu kvadratnu matricu A postoji permutacija P takva da za permutiranu matricu $\tilde{A} = PA$ postoji LU faktorizacija, tj. $\tilde{A} = LU$.

Algoritam LUDP (LU faktorizacija s djelomičnim pivotiranjem)

ulaz: $A \in \mathbb{C}^{n\times n}$ regularna

izlaz: $L, U, P \in \mathbb{C}^{n\times n}$, takve da je $PA = LU$, gdje je L donje trokutasta s jedinicama na dijagonali, U regularna gornja trokutasta i P matrica permutacija

1: $U = A$, $L = I$, $P = I$

2: **for** $k = 1, \ldots, n-1$ **do**

3: izaberi $i \in k, \ldots, n$ tako da je $|u_{ik}|$ najveće

4: zamijeni red $(u_{k,k}, \ldots, u_{k,n})$ redom $(u_{i,k}, \ldots, u_{i,n})$

5: zamijeni red $(l_{k,1}, \ldots, l_{k,k-1})$ redom $(l_{i,1}, \ldots, l_{i,k-1})$

6: zamijeni red $(p_{k,1}, \ldots, p_{k,n})$ redom $(p_{i,1}, \ldots, p_{i,n})$

7: **for** $j = k+1, \ldots, n$ **do**

8: $l_{jk} = u_{jk}/u_{kk}$

9: $(u_{j,k}, \ldots, u_{j,n}) = (u_{j,k}, \ldots, u_{j,n}) - l_{j,k}(u_{k,k}, \ldots, u_{k,n})$

10: **end for**

11: **end for**

Napomena 5.6

1) Za sve elemente matrice L dobivene gornjim algoritmom vrijedi $l_{ij} \leq 1$ za sve $i, j \in \{1, \ldots, n\}$.

2) Složenost računanja ostaje nepromijenjena kao i u slučaju običnog LU algoritma (bez pivotiranja). Ovo je očito, jer je jedina razlika između ta dva algoritma dodatna zamjena redova, za što nam ne trebaju dodatne računske operacije. Dodatne permutacije zahtijevaju $\mathcal{O}(n^2)$ pridruživanja, ali se mogu zanemariti u odnosu na dominantni dio od $\Theta(n^3)$ računskih operacija iz LU algoritma.

Dakle, modificirani algoritam (Gaussove eliminacije s djelomičnim pivotiranjem) izvode se na sljedeći način. Izračunavamo

$$U = L_{n-1}P_{n-1} \cdots L_1 P_1 A.$$

Budući da množenje slijeva matricama P_k zamjenjuje k-ti s i_k-tim retkom, redak i_k izabiremo tako da element $u_{i_k k}$ bude najveći po apsolutnoj vrijednosti. Gornju jednakost možemo zapisati i kao

$$U = \widetilde{L}_{n-1} \cdots \widetilde{L}_1 P_{n-1} \cdots P_1 A,$$

gdje je

$$\widetilde{L}_k = P_{n-1} \cdots P_{k+1} L_k P_{k+1}^{-1} \cdots P_{n-1}^{-1}$$

za $k = 1, \ldots, n-1$. Primijetimo da $P_{n-1} \cdots P_{k+1}$ zamjenjuju retke $k+1, \ldots, n$, a $P_{k+1}^{-1} \cdots P_{n-1}^{-1}$ zamjenjuju odgovarajuće stupce, $k+1, \ldots, n$. Stoga

vidimo da je oblik matrice $\widetilde{L}_k$ isti kao i oblik matrice L_k, odnosno obje matrice su donje trokutaste s jedinicama na dijagonali, a jedini elementi različiti od nule nalaze im se u k-tom stupcu.

Sve zajedno daje sljedeći algoritam za Gaussove eliminacije s djelomičnim pivotiranjem:

Algoritam GEDP (Gaussove eliminacije s djelomičnim pivotiranjem).
ulaz: $A \in \mathbb{C}^{n\times n}$ regularna matrica, $b \in \mathbb{C}^n$
izlaz: $x \in \mathbb{C}^n$ za koji vrijedi $Ax = b$
1: odrediti $PA = LU$ koristeći algoritam LUDP
2: riješiti $Ly = Pb$ pomoću supstitucija unaprijed
3: riješiti $Ux = y$ pomoću povratnih supstitucija

Rezultat ovog algoritma je vektor $x \in \mathbb{C}^n$ za koji je $Ax = P^{-1}LUx = P^{-1}Ly = P^{-1}Pb = b$, pa vidimo da je algoritam korektan.

5.3.1 Analiza pogreške za GEDP

Teorem 5.4 (Povratna analiza pogreške za LUDP) *Neka je LU faktorizacija kvadratne matrice A reda n izračunata s pivotiranjem redaka u aritmetici s relativnom točnošću ε_m i neka su $\widetilde{L}$ i $\widetilde{U}$ dobivene aproksimacije za L i U. Ako je pri tome korištena permutacija P, onda je*

$$\widetilde{L}\widetilde{U} = P(A + \Delta A)$$

i vrijedi

$$|\Delta A| \leq \frac{2n\varepsilon_m}{1 - 2n\varepsilon_m} P^T |\tilde{L}|\, |\tilde{U}|\,. \tag{5.3}$$

Važno je napomenuti da standardno pivotiranje redaka osigurava da su u matrici L svi elementi po apsolutnoj vrijednosti manji ili jednaki jedinici (što vrijedi i za matricu $|\tilde{L}|$), dok elementi matrice $\tilde{U}$ ovise o (i dobiveni su pomoću) elementima matrice $\tilde{A}^{(k)}$, $k = 0, 1, \ldots, n-1$. Stoga je broj $g_n(A)$ *(faktor rasta elemenata)* definiran sa

$$g_n(A) = \frac{\max_{ij} |u_{ij}|}{\max_{ij} |a_{ij}|}\,,$$

dobra mjera za relativni rast (u odnosu na A) elemenata u produktu $|\tilde{L}||\tilde{U}|$.

Vrijedi sljedeća ocjena:

Napomena 5.7 *Umjesto ocjene iz teorema 5.4 često se koristi sljedeća ocjena. Za rastav*

$$\widetilde{L}\widetilde{U} = P(A + \Delta A)$$

vrijedi

$$\frac{\|\Delta A\|}{\|A\|} \leq \varepsilon_m p(n) g_n(A),$$

gdje je p polinom a $g_n(A)$ (faktor rasta elemenata) definiran sa

$$g_n(A) = \frac{\max_{ij} |u_{ij}|}{\max_{ij} |a_{ij}|}.$$

Primijetimo da pivotiranje redaka doprinosi numeričkoj stabilnosti tako što se za *faktor rasta elemenata* $g_n(A)$ *može osigurati umjeren rast u toku* LU *faktorizacije.*

Sljedeća lema pokazuje da u slučaju pivotiranja redaka faktor rasta elemenata $g_n(A)$ ima gornju ogradu koja je funkcija samo dimenzije problema.

Lema 5.4 *Faktor rasta elemenata* $g_n(A)$ *omeđen je odozgo i vrijedi ocjena*

$$g_n(A) \leq 2^{n-1}$$

za svaki $A \in \mathbb{C}^{n\times n}$.

Dokaz. Matricu U dobivamo kao $U = \widetilde{L}_{n-1} \cdots \widetilde{L}_1 PA$, gdje su $\widetilde{L}_k$ oblika $I + u_k e_k^T$. Za po volji izabran $k \in \{1, \ldots, n-1\}$ i za bilo koju matricu $A \in \mathbb{C}^{n\times n}$ vrijedi

$$\begin{aligned}
\max_{i,j} |(\widetilde{L}_k A)_{ij}| &= \max_{i,j=1,\ldots,n} |\sum_{l=1}^{n} (\widetilde{L}_k)_{il}\, a_{lj}| \\
&\leq \max_{i=1,\ldots,n} \left| \sum_{l=1}^{n} (\widetilde{L}_k)_{il} \right| \max_{i,j} |a_{ij}| \\
&\leq 2 \max_{i,j} |a_{ij}|,
\end{aligned}$$

stoga redom množeći s matricama $\widetilde{L}_1, \ldots, \widetilde{L}_{n-1}$ slijeva matricu PA dobivamo

$$\max_{i,j} |u_{ij}| \leq 2^{n-1} \max_{i,j} |(PA)_{ij}| \leq 2^{n-1} \max_{i,j} |a_{ij}|.$$

Ovime smo dokazali teorem. □

Primjer 5.5 *Neka je*

$$(A_1 =)A = \begin{pmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -1 & -1 & \cdots & -1 & 1 \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

Budući da svi elementi koji su nam važni za provođenje Guassovih eliminacija imaju modul 1 *(to su elementi na i ispod glavne dijagonale), nije potrebno permutirati retke (pivotirati). LU faktorizacija nam daje*

$$(A_{n-1} =)U = \begin{pmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & \ddots & \vdots \\ & & & 1 & 2^{n-2} \\ & & & & 2^{n-1} \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

Vidimo da za matricu A faktor rasta elemenata iznosi

$$g_n(A) = \frac{\max_{i,j}|u_{ij}|}{\max\limits_{i,j}|a_{ij}|} = 2^{n-1}.$$

Gornji primjer pokazuje da je ocjena leme 5.4 oštra, tj. da postoje matrice na kojima se ona dostiže. Stoga konstanta iz ocjene povratne pogreške iz teorema 5.4 može rasti eksponencijalno u ovisnosti o dimenziji n.

Teorem 5.4 i lema 5.4 zajedno pokazuju da je LU algoritam s djelomičnim pivotiranjem povratno stabilan.

Također se može pokazati da vrijedi sljedeći teorem:

Teorem 5.5 *Neka je $\tilde{x}$ rješenje regularnog sustava jednadžbi $Ax = b$, dobiveno Gaussovim eliminacijama s pivotiranjem redaka. Tada postoji perturbacija ΔA za koju vrijedi*

$$(A + \Delta A)\tilde{x} = b, \qquad |\Delta A| \leq \frac{5n\varepsilon}{1 - 2n\varepsilon} P^T|\tilde{L}||\tilde{U}|.$$

Ovdje je P permutacija koja realizira zamjenu redaka. Također pretpostavljamo da je $2n\varepsilon < 1$.

Konačno, neka je $\tilde{x}$ rješenje regularnog sustava jednadžbi $Ax = b$. Tada napomena 5.7 zajedno s teoremom 4.1 daje ocjenu za relativnu pogrešku rješenja (pri čemu zanemarujemo pogrešku desne strane, tj. $\delta b = 0$):

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \frac{\kappa(A)\, g_n(A)\, p(n)}{1 - \kappa(A)\varepsilon_m p(n) g_n(A)} \cdot \varepsilon_m.$$

5.3.2 Zadaci

1. Analogno kao kod povratnih supstitucija konstruirajte algoritam koji rješava $Ly = b$, gdje je $L \in \mathbb{C}^{n\times n}$ donja trokutasta regularna matrica, a $b \in \mathbb{C}^n$ je vektor. Pokažite da dobiveni algoritam daje ispravan rezultat te da je asimptotski broj računskih operacija reda veličine $\Theta(n^2)$.

2. Odredite LU-faktorizaciju matrice

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}.$$

   Kako glasi faktor rasta $g_n(A)$ u ovom slučaju?

3. Odredite matrice P, L i U koje određuju LU faktorizaciju matrice A s djelomičnim pivotiranjem $PA = LU$ ako je

$$A = \begin{pmatrix} 3 & 1 & 2 & 5 \\ 2 & -2 & 3 & -2 \\ 1 & 5 & 5 & 4 \\ -4 & 8 & -4 & 12 \end{pmatrix}.$$

   [Rješenje: $L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1/4 & 1 & 0 & 0 \\ -3/4 & 1 & 1 & 0 \\ -1/2 & 2/7 & 1/35 & 1 \end{pmatrix}$; $U = \begin{pmatrix} -4 & 8 & -4 & 12 \\ 0 & 7 & 4 & 7 \\ 0 & 0 & -5 & 7 \\ 0 & 0 & 0 & 9/5 \end{pmatrix}$; $P = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$.]

4. Gaussovom metodom eliminacija s djelomičnim pivotiranjem riješite sustav $Ax = b$ i odredite matrice P, L i U takve da je $PA = LU$, ako je

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 0 & 1 \\ 1 & -1 & 0 & -1 \\ 2 & 3 & -2 & -2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 7 \\ 0 \\ 10 \end{pmatrix}.$$

[Rješenje: $x = \begin{pmatrix} 1 & 2 & 0 & -1 \end{pmatrix}^T$.]

5. Koristeći LU faktorizaciju s djelomičnim pivotiranjem izračunajte inverz matrice A, ako je

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 2 & -3 & 6 \\ 5 & -4 & 10 \end{pmatrix}.$$

[Rješenje: $A^{-1} = \frac{1}{46} \begin{pmatrix} 6 & -46 & 30 \\ 10 & 0 & -4 \\ 7 & 23 & -12 \end{pmatrix}$.]

5.4 QR dekompozicija

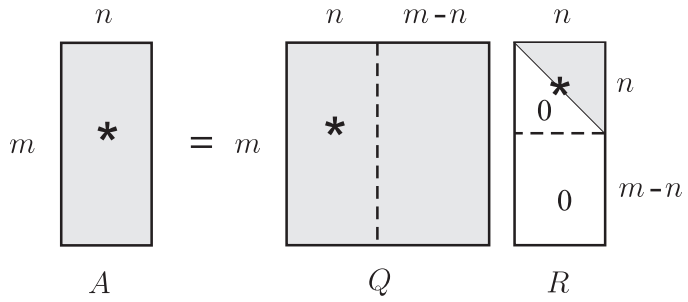
Sljedeća matrična faktorizacija (dekompozicija) koju ćemo proučavati u okviru ovog poglavlja jest QR dekompozicija. Ona je često vrlo korisna i može se primjenjivati u raznim problemima. U okviru ovog poglavlja pokazat ćemo njenu primjenu na rješavanje linearnih sustava, a u sljedećem poglavlju vidjet ćemo i neke druge moguće primjene QR dekompozicije.

U biti, QR dekompozicija jest rastav matrice A na dvije matrice $A = QR$, od kojih je jedna unitarna a druga gornja trokutasta. U sljedećem ćemo teoremu dokazat postojanje takve dekompozicije.

Teorem 5.6 (QR dekompozicija) *Za bilo koju matricu* $\mathrm{A} \in \mathbb{C}^{m\times n}$*, takvu da je* $m \geq n$ *postoje matrice* Q *i* R *takve da* $A = QR$*, pri čemu je* $Q \in \mathbb{C}^{m\times m}$ *unitarna, a* $R \in \mathbb{C}^{m\times n}$ *je gornja trokutasta.*

Napomena 5.8 *Faktorizaciju iz teorema 5.6 zovemo* puna QR dekompozicija*. Budući da su svi elementi u matrici* R *ispod glavne dijagonale jednaki* 0*, vidimo da stupci* $n+1, \ldots, m$ *matrice* Q *ne doprinose umnošku* QR*. Neka*

se matrica $\widetilde{Q} \in \mathbb{C}^{m\times n}$ *sastoji od prvih* n *stupaca matrice* Q *i neka se matrica* $\widetilde{R} \in \mathbb{C}^{n\times n}$ *sastoji od prvih* n *redaka matrice* R. *Tada se lako može provjeriti da vrijedi* $A = \widetilde{Q}\,\widetilde{R}$. *Ovakvu dekompoziciju zovemo* reducirana QR dekompozicija *matrice* A. *Sljedeća slika ilustrira navedeno:*



Dokaz. Neka su $a_1, \ldots, a_n \in \mathbb{C}^m$ stupci matrice A, a $q_1, \ldots, q_m \in \mathbb{C}^m$ stupci matrice Q. Teorem 5.6 dokazujemo tako da ćemo pomoću Gramm-Schmidtovog postupka ortogonalizacije konstruirati matrice Q i R.

Postupak kojim se mogu konstruirati matrice Q i R dan je sljedećim algoritmom:

1: $q_1 = a_1/\|a_1\|_2, \qquad r_{11} = \|a_1\|_2$
2: **for** $j = 2, \ldots, n$ **do**
3: $\widehat{q}_j = a_j$
4: **for** $k = 1, \ldots, j-1$ **do**
5: $r_{kj} = \langle q_k, a_j \rangle$
6: $\widehat{q}_j = \widehat{q}_j - r_{kj}\, q_k\,, \qquad \left(\widehat{q}_j = a_j - \sum_{k=1}^{j-1} r_{kj}\, q_k\right)$
7: **end for**
8: $r_{jj} = \|\widehat{q}_j\|_2$
9: **if** $r_{jj} > 0$ **then**
10: $q_j = \widehat{q}_j / r_{jj}$
11: **else**
12: izaberimo q_j tako da je okomit na q_1,..., q_{j-1}
13: **end if**
14: **end for**
15: izaberimo $q_{n+1}, \ldots, q_m$ tako da $q_1, \ldots, q_m$ čine ortonormalan sustav

Navedeni algoritam izračunava stupce $q_1, \ldots, q_m$ matrice Q i elemente matrice R koji se nalaze iznad i na glavnoj dijagonali. Svi su elementi matrice R ispod glavne dijagonale jednaki 0. Uz korištenje MatLab oznaka $A_{(i,:)}$ ($A_{(:,j)}$) za i-ti redak (j-ti stupac) matrice A, provjerimo kako izgleda (i,j)-ti element matrice QR za ovako izračunate matrice Q i R:

$$(QR)_{ij} = (QR_{(:,j)})_{(i,:)} = \left(\sum_{k=1}^{j} q_k r_{kj}\right)_{(i,:)} = \left(\sum_{k=1}^{j-1} q_k r_{kj} + q_j r_{jj}\right)_{(i,:)}$$
$$= \left(\sum_{k=1}^{j-1} q_k r_{kj} + \widehat{q}_j\right)_{(i,:)} = \left(\sum_{k=1}^{j-1} q_k r_{kj} + \left(a_j - \sum_{k=1}^{j-1} q_k r_{kj}\right)\right)_{(i,:)} = a_{ij}$$

stoga vidimo da je $A = QR$.

Iz konstrukcije slijedi da za sve stupce matrice Q vrijedi $\|q_j\| = 1$ za $j = 1,...,m$. Ostaje nam još provjeriti da su stupci matrice Q međusobno okomiti, tj. da za $q_1, \ldots, q_j$ vrijedi $\langle q_i, q_j\rangle = \delta_{ij}$, za $i, j \in \{1, \ldots, n\}$. Dokaz provodimo matematičkom indukcijom, za $j = 1$ se nema što dokazivati, a lako se provjeri da je za $j = 2$, $\langle q_1, q_2\rangle = 0$, zaista $q_1 = a_1/\|a_1\|_2$, i $\langle q_1, q_1\rangle = 1$, pa imamo:

$$\begin{aligned}
\langle q_1, q_2\rangle &= \frac{1}{r_{22}}\langle q_1, \widehat{q}_2\rangle \\
&= \frac{1}{r_{22}}\langle q_1, a_2 - r_{12}q_1\rangle \\
&= \frac{1}{r_{22}}\langle q_1, a_2 - \langle q_1, a_2\rangle\, q_1\rangle \\
&= \frac{1}{r_{22}}(\langle q_1, a_2\rangle - \langle q_1, a_2\rangle) = 0\,.
\end{aligned}$$

Pretpostavimo da je $j > 2$ i neka su vektori $q_1, \ldots, q_{j-1}$ međusobno okomiti. Trebamo pokazati da $\langle q_i, q_j\rangle = 0$ za $i = 1, \ldots, j-1$. Ako je $r_{jj} = 0$, tada tvrdnja vrijedi iz definicije vektora q_j. U suprotnom imamo:

$$\begin{aligned}
\langle q_i, q_j\rangle &= \frac{1}{r_{jj}}\langle q_i, \widehat{q}_j\rangle \\
&= \frac{1}{r_{jj}}(\langle q_i, a_j\rangle - \sum_{k=1}^{j-1} r_{kj}\langle q_i, q_k\rangle) \\
&= \frac{1}{r_{jj}}(\langle q_i, a_j\rangle - r_{ij}) = 0.
\end{aligned}$$

Ovim smo pokazali da su stupci matrice Q međusobno okomiti, a budući da su svi normirani (jedinične duljine), vidimo da je Q unitarna. □

Napomena 5.9

1) Ovdje je potrebno napomenuti da Gram-Schmidtov postupak ortogonalizacije koji smo koristili u dokazu teorema 5.6 nije numerički stabilan, pa nije preporučljivo koristiti ga za izračunavanje QR dekompozicija.

2) Za $m = n$ matrice $Q, R \in \mathbb{C}^{n\times n}$ su kvadratne. Budući da je

$$\det(A) = \det(QR) = det(Q)\ \det(R)\,,$$

a $\det(Q) \in \{1, -1\}$, vidimo da će matrica R biti regularna ako i samo ako je A regularna.

Kao što smo naveli u uvodu u ovo poglavlje, jedna od mogućih primjena QR dekompozicija je rješavanje sustava linearnih jednadžbi (SLJ). Stoga ćemo navesti algoritam koji koristi QR dekompoziciju za rješavanje problema SLJ.

Algoritam (rješavanje SLJ pomoću QR dekompozicije).
ulaz: $A \in \mathbb{C}^{n\times n}$
izlaz: $x \in \mathbb{C}^n$ takav da $Ax = b$

1: odrediti QR dekompoziciju matrice A, tako da je $A = QR$

2: izačunati $y = Q^* b$

3: izačunati x iz sustava $Rx = y$ pomoću povratnih supstitucija

Rezultat gornjeg algoritma jest vektor $x \in \mathbb{C}^n$ takav da je $Ax = QRx = Qy = QQ^*b = b$, pa vidimo da je algoritam ispravan.

Kako bi gornji algoritam bio upotrebljiv, trebat će konstruirati algoritam za QR dekompoziciju koji je stabilan. U tu svrhu morat ćemo prikazati drugačiju metodu za računanje QR dekompozicija, koja je stabilna. Nakon toga ćemo analizirati kako on radi. U algoritmu koji ćemo prikazati koristi se funkcije predznaka, stoga se prisjetimo njezine definicije:

$$\operatorname{sign}(x) = \begin{cases} +1, & \text{ako je } x \geq 0; \\ -1, & \text{ako je } x < 0. \end{cases}$$

5.4.1 Householderova QR dekompozicija (Householderovi reflektori)

U Gram-Schmidtovom postupku ortogonalizacije osnovni cilj je bio ortogonalizirati stupce matrice A. U tu svrhu je matrica A postupno transformirana u ortogonalnu matricu, a gornje trokutasta matrica R nastala je kao sporedni proizvod te ortogonalizacije.

Budući da je poznato da Gram-Schmidtov postupak ortogonalizacije nije numerički stabilan, američki je matematičar Alston Scott Householder predložio drugačiji pristup: osnovni cilj jest transformirati matricu A u gornju trokutastu pomoću unitarnih transformacija primijenjenih na stupce matrice A.

Osnovnu ideju možemo vidjeti u sljedećem algoritmu:

QR algoritam (Householderova QR dekompozicija).
ulaz: A $\in \mathbb{C}^{m\times n}$, za $m \geq n$
izlaz: Q $\in \mathbb{C}^{m\times m}$ ortogonalna matrica i $R \in \mathbb{C}^{m\times n}$ gornje trokutasta,
takva da je $A = QR$

1: $Q = I, R = A$
2: **for** $k = 1, ..., n-1$ **do**
3: $u = (r_{kk}, ..., r_{mk})^* \in C^{m-k+1}$
4: $\bar{v} = \text{sign}(u_1)\|u\|_2 e_1 + u$ gdje je $e_1 = (1, 0, ..., 0)^* \in C^{m-k+1}$
5: $v = \bar{v}/\|\bar{v}\|_2$
6: $H_k = (I_{m-k+1} - 2\, vv^*) \in \mathbb{C}^{(m-k+1)\times(m-k+1)}$
7: $Q_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H_k \end{pmatrix}$
8: $R = Q_k \cdot R$
9: $Q = Q \cdot Q_k$
10: **end for**

Napomena 5.10

1) Gornji algoritam izračunava matrice Q_k, za koje vrijedi $Q_k = Q_k^*$ za $k = 1, ..., n-1$, kao i matricu $R = Q_{n-1} \cdots Q_1 A$, odnosno ako definiramo $Q = Q_1 \cdots Q_{n-1}$, onda je $R = Q^* A$.

2) Pokazat ćemo da matrica $Q_k \cdots Q_1 A$ ispod glavne dijagonale u stupcima $1, ..., k$ ima samo nul-elemente, to znači da je matrica $R = Q_{n-1} \cdots Q_1 A$ gornja trokutasta.

3) Važno je primijetiti da pri rješavanju linearnih sustava oblika $Ax = b$, matricu Q trebamo jedino za izračunavanje vektora desne strane Q^*b. Stoga

u algoritmu koji će rješavati linearni sustav $Ax = b$ ne moramo izračunati cijelu matricu Q (ne moramo formirati matricu Q), nego možemo zamijeniti 9. redak QR algoritma sa $b = Q_k b$. Na taj način će rezultat algoritma biti novi vektor desne strane b, za koji vrijedi

$$Q_{n-1} \cdots Q_1 b = (Q_1 \cdots Q_{n-1})^* b = Q^* b$$

Householderove matrice (Householderovi reflektori)

U 8. koraku Householderove QR dekompozicije računa se umnožak

$$\mathbf{Q}_k \cdot \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} \\ 0 & H_k R_{22} \end{pmatrix}, \tag{5.4}$$

gdje su $R_{11} \in \mathbb{R}^{(k-1)\times(k-1)}$ i $H_k, R_{22} \in \mathbb{R}^{(m-k+1)\times(n-k+1)}$. U ovom dijelu ćemo podrobnije razjasniti taj korak.

Lako se može provjeriti da je H_k koju smo izračunali u 6. koraku Householderovog QR algoritma hermitska matrica, tj. da vrijedi $H_k^* = H_k$. Nadalje, vrijedi

$$H_k^* H_k = (I - 2vv^*)\,(I - 2vv^*) = I - 4vv^* + 4v\,(v^*v)\,v^* = I\,.$$

Time smo pokazali da vrijedi

$$H_k^* H_k = H_k H_k = I\,, \tag{5.5}$$

što pokazuje da su matrice H_k, a time i Q_k, unitarne za svaki $k \in \{1, \ldots, n-1\}$.

Primijetimo da se za bilo koji vektor $u = \begin{bmatrix} u_1 & u_2 & \ldots & u_{m-k+1} \end{bmatrix}^T$, takav da je $u \in \mathbb{R}^{m-k+1}$, vektor $H_k u$ može izračunati kao

$$H_k u = u - 2vv^*u = u - 2v\langle v, u\rangle = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{m-k+1} \end{bmatrix} - 2\langle v, u\rangle\, v\,. \tag{5.6}$$

Nadalje uočimo da vektor v izračunat u 5. koraku Householderova QR algoritma ima sljedeći oblik:

$$\bar{v} = \begin{bmatrix} u_1 + \operatorname{sign}(u_1)\|u\|_2 \\ u_2 \\ \vdots \\ u_{m-k+1} \end{bmatrix}, \quad u_1 = r_{kk}\,, \ldots, u_{m-k+1} = r_{mk}\,.$$

Za izračunavanje vektora $H_k u$, gdje je u definiran u 3. koraku Householderova QR algoritma, potrebno je uočiti sljedeće

$$\begin{aligned} & \langle \bar{v}, u \rangle \equiv \bar{v}^* u = \operatorname{sign}(u_1) \|u\|_2 \cdot u_1 + \|u\|_2^2 \\ \text{i} \qquad & \|\bar{v}\|^2 = \|u\|_2^2 + 2 \operatorname{sign}(u_1) \|u\|_2 \cdot u_1 + \|u\|_2^2 \\ & = 2 \left(\|u\|^2 + \operatorname{sign}(u_1) \cdot u_1 \|u\|_2 \right). \end{aligned}$$

Iz gornjih jednakosti slijedi da je

$$\frac{2}{\|\bar{v}\|^2} \langle \bar{v}, u \rangle \equiv \frac{2}{\|\bar{v}\|^2} \bar{v}^* u = 1 \,.$$

Sada iz gornjih jednakosti i (5.6) slijedi da vektor $H_k u$ ima oblik:

$$H_k u = u - \frac{2}{\|\bar{v}\|^2} \bar{v} \bar{v}^* u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{m-k+1} \end{bmatrix} - \begin{bmatrix} u_1 + \operatorname{sign}(u_1) \|u\|_2 \\ u_2 \\ \vdots \\ u_{m-k+1} \end{bmatrix} = - \begin{bmatrix} \operatorname{sign}(u_1) \|u\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{5.7}$$

Važno je primijetiti da iz (5.5) slijedi da je H_k projektor. Zaista, budući daje vektor $v\langle v, x \rangle$ projekcija vektora x na vektor v, vidimo da je vektor $x - v\langle v, x \rangle$ projekcija vektora x na ravninu okomitu na vektor v. Stoga možemo zaključiti da je $x - 2v\langle v, x \rangle$ refleksija vektora x u toj ravnini.

Vektor kojim je definirana ravnina refleksije dan je sa: $\bar{v} = u - \|u\|_2 e_1$ ili $\bar{v} = u - (-\|u\|_2 e_1)$, u ovisnosti o predznaku u_1. Odgovarajuća slika (refleksija) vektora u dana je sa $H_k u = \|u\|_2 e_1$ ili $H_k u = -\|u\|_2 e_1$. U oba slučaja slika je proporcionalna s vektorom e_1, a budući da je vektor u izabran kao prvi stupac blok-matrice R_{22} iz (5.4), tj. vrijedi $u_1 = r_{kk}, \ldots, u_{m-k+1} = r_{mk}$, vidimo da će u umnošku $H_k R_{22}$ svi elementi ispod glavne dijagonale biti jednaki nuli. Prvi stupac blok-matrice R_{22} je u stvari k-ti stupac matrice R, pa zaključujemo da matrica $Q_k R$ ima samo nul elemente ispod glavne dijagonale u stupcima $1, \ldots, k$. Nastavimo li zaključivati na isti način vidimo da je za $k = n - 1$ matrica $R = Q_{n-1} \cdots Q_1 A$ gornja trokutasta, što smo i trebali pokazati.

Napomena 5.11

i) Matrice H_k ili Q_k ponekad se zovu Householderovi reflektori.

ii) Izbor predznaka u definiciji vektora u pomaže u povećanju stabilnosti algoritma u slučajevima kad je $u \approx \|u\|_2 e_1$ ili $u \approx -\|u\|_2 e_1$.

5.4.2 Broj računskih operacija (trošak računanja)

Broj računskih operacija QR algoritma posebno ćemo određivati za slučaj kad nas zanima cijela matrica Q kao što je to definirano u 9. koraku QR algoritma, a posebno za slučaj rješavanja SLJ kad nam treba samo vektor Q^*b, što se postiže promjenom 9. koraka definiranjem novog koraka danog sa $b = Q_k b$.

Prvo ćemo izbrojati potrebne računske operacije bez 9. koraka.

Lema 5.5 *Broj računskih operacija $C(m,n)$ Householderovog QR algoritma u slučaju matrice $m \times n$, ne računajući matricu Q niti vektor Q^*b, dan je sa:*

$$C(m,n) \sim 2\,m\,n^2 - \frac{2}{3}n^3 \quad \text{za} \quad m,n \to \infty \quad \text{za slučaj} \quad m = \Theta(n).$$

Dokaz. Broj računskih operacija odredit ćemo za svaki pojedini korak QR algoritma. Prije nego što to učinimo uočimo da iz relacije (5.4) slijedi da nam je za računanje umnožaka $Q_k R$ u 8. koraku jedino potrebno izračunati $H_k R_{22} = R_{22} - 2vv^* R_{22}$. Budući da je $v = \bar{v}/\|\bar{v}\|_2$ i $\|\bar{v}\|_2 = \sqrt{\bar{v}^*\bar{v}}$, taj se produkt može računati pomoću sljedeće jednakosti:

$$H_k R_{22} = R_{22} - \frac{\bar{v}}{\bar{v}^*\bar{v}/2}\bar{v}^* R_{22} \in \mathbb{R}^{(m-k+1)\times(n-k+1)}\,. \tag{5.8}$$

Koristeći jednakost (5.8) računske operacije ćemo izbrojati na sljedeći način:

1. Za izračunavanje vektora $\bar{v}$ trebamo $2\,(m-k+1)+1$ operacija, točnije $(m-k+1)$ množenja, odnosno $(m-k)$ zbrajanja i jedno korijenovanje za izračunavanje $\|u\|_2$ ($\sqrt{\cdot}$ računamo kao jednu operaciju) i još jedno zbrajanje prve komponente matrice u sa $\text{sign}(u_1)\|u\|_2$.

2. Za izračunavanje svake od $n-k+1$ komponenti umnožaka matrica-vektor $\bar{v}^* R_{22}$ trebamo $m-k+1$ množenja i $m-k$ zbrajanja, znači da ukupan broj operacija za izračunavanje $\bar{v}^* R_{22}$ iznosi $(n-k+1)(2\,(m-k+1)-1)$.

3. Za izračunavanje $\bar{v}^*\bar{v}/2$ potrebno je $2(m-k+1)$ operacija ($(m-k+1)$ množenja, $(m-k)$ zbrajanja i 1 dijeljenje), te dodatno za dijeljenje vektora $\bar{v}$ s dobivenim brojem još $(m-k+1)$ dijeljenja.

4. Za izračunavanje produkta između dobivenog vektora u gornjem razmatranju i vektora $\bar{v}^* R_{22}$, tj. za izračunavanje $(\cdots)(\bar{v}^* R_{22})$ treba nam $(m-k+1)(n-k+1)$ množenja.

5 Za izračunavanje razlike $R_{22} - (\cdots)$ potrebno je $(m-k+1)(n-k+1)$ oduzimanja.

Stoga je ukupan broj računskih operacija dan sa

$$C(m,n) \;=\; \sum_{k=1}^{n-1} 5(m-k+1) + 1 + (n-k+1)(4(m-k+1)-1)\,.$$

Uvođenjem supstitucije $l = m-k+1$, iz $k = 1, \ldots, n-1$ slijedi da možemo pisati da $l = m-n+2, \ldots, m$, dobivamo:

$$\begin{aligned} C(m,n) &= \sum_{l=m-n+2}^{m} 5l + 1 + (n-m+l)(4l-1) \\ &= 4\sum_{l=m-n+2}^{m} (n-m)\,l + 4\sum_{l=m-n+2}^{m} l^2 + \text{ članovi s najviše dva faktora } m,n \\ &\sim 4(n-m)(mn - \frac{n^2}{2}) + \frac{4}{6}\left(2m^3 - 2(m-n)^3\right) \\ &\sim 2mn^2 - \frac{2}{3}n^3 \end{aligned}$$

za $m, n \to \infty$, uz uvjet $m = \Theta(n)$. □

U slučaju ako trebamo izračunavati i matricu Q, dodatno je potrebno izvršiti $(m-k+1) \times (m-k+1)$ matričnih množenja definiranih u 9. koraku. Ako pretpostavimo da se to množenje izvodi standardnom metodom za matrično množenje, tada dodani broj računskih operacija iznosi $\Theta(m^3)$, tj. asimptotski doprinos broju računskih operacija QR algoritma bit će uvećan upravo za taj red veličine. Međutim, ako koristimo QR algoritam za rješavanje SLJ, tada je jedino potrebno u svakom koraku izračunati Q^*b umjesto cijele matrice Q. Gledajući preko algoritma vidimo da je to analogno kao da računamo QR dekompoziciju matice A proširene vektorom b (QR dekompozicija matrice $[A, b]$). Stoga ukupni broj računskih operacija u tom slučaju možemo izračunati kao

$$C(m, n+1) \sim 2m(n+1)^2 - \frac{2}{3}(n+1)^3 \sim 2mn^2 - \frac{2}{3}n^3$$

za $m, n \to \infty$ i za $m = \Theta(n)$, što znači da se za velike n i m broj računskih operacija ne mijenja. Primijetimo da kod rješavanja SLJ imamo $m = n$ pa

za broj računskih operacija pri rješavanju SLJ pomoću Householderove QR dekompozicija dobivamo

$$C(n) \sim 2nn^2 - \frac{2}{3}n^3 = \frac{4}{3}n^3 \quad \text{za } n \to \infty.$$

Primijetimo da je broj operacija za rješavanje SLJ pomoću Householderove QR dekompozicije dvostruko veći od broja operacija za rješavanje SLJ pomoću Gaussovih eliminacija s djelomičnim pivotiranjem. Međutim, metoda rješavanja SLJ pomoću Householderove QR dekompozicije je stabilnija od Gaussovih eliminacija s djelomičnim pivotiranjem.

5.4.3 Analiza točnosti

Teorem 5.7 *a) Za matricu $A \in \mathbb{R}^{m\times n}$, neka su $\tilde{R}, \tilde{v}_1, \ldots, \tilde{v}_{n-1}$ veličine izračunate Householderovim QR algoritmom. Neka je*

$$\tilde{Q}_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & I_{m-k+1} - 2\tilde{v}_k\tilde{v}_k^* \end{pmatrix}$$

i $\tilde{Q} = \tilde{Q}_1 \cdots \tilde{Q}_{n-1}$. Tada je $\tilde{Q}\tilde{R} = A + \Delta A$ za neku matricu $\Delta A \in \mathbb{R}^{m\times n}$, pri čemu vrijedi

$$\frac{\|\Delta A\|}{\|A\|} = \mathcal{O}(\varepsilon_m).$$

b) Neka je $\tilde{y}$ izračunata vrijednost za $\tilde{Q}^*b$. Tada

$$(\tilde{Q} + \Delta Q)\tilde{y} = b, \quad \|\Delta Q\| = \mathcal{O}(\varepsilon_m).$$

Napomena 5.12

1) Veličine $\tilde{v}_1, \ldots, \tilde{v}_{n-1}$ iz gornjeg teorema predstavljaju vrijednosti vektora v izračunatog u 5. koraku algoritma za k-tu iteraciju. Budući da se matrica Q_k ne računa eksplicitno (koristimo jednakost (5.8)), veličine $\tilde{v}_1, \ldots, \tilde{v}_{n-1}$ su izračunate relevantne veličine. U našoj ćemo analizi proučavati matrice $\tilde{Q}_k$ koje su točno izračunate pomoću vektora v_k, stoga su egzaktno ortogonalne.

2) Primijetite da se drugi dio gornjeg teorema (dio b)) odnosi na apsolutnu pogrešku za $\tilde{Q}^*$, koja se na prvi pogled ne čini baš korisnom. No kako ćemo vidjeti u sljedećem teoremu, koji govori o povratnoj stabilnosti Householderovog algoritma, ta se ocjena može uspješno iskoristiti.

Teorem 5.8 *Householderov algoritam za rješavanje SLJ je povratno stabilan: izračunato rješenje $\tilde{x}$ linearnog sustava $Ax = b$ zadovoljava ocjenu*

$$(A + \Delta\bar{A})\tilde{x} = b, \qquad \frac{\|\Delta\bar{A}\|}{\|A\|} = \mathcal{O}(\varepsilon_m).$$

Dokaz. Neka su $\tilde{Q}$, $\tilde{R}$ i $\tilde{y}$ dane kao u teoremu 5.7. Rješenje $\tilde{x}$, SLJ $Ax = b$, izračunava se kao $\tilde{R}x = \tilde{y}$ pomoću povratnih supstitucija. Iz činjenice da je algoritam povratnih supstitucija povratno stabilan (vidi teorem 5.3) slijedi da izračunato rješenje $\tilde{x}$ zadovoljava

$$(\tilde{R} + \Delta R)\tilde{x} = \tilde{y}, \quad \frac{\|\Delta R\|}{\|R\|} = \mathcal{O}(\varepsilon_m).$$

To nam za izračunate $\tilde{Q}$ i $\tilde{y}$ daje

$$\begin{aligned} b &= (\tilde{Q} + \Delta Q)\tilde{y} \\ &= (\tilde{Q} + \Delta Q)(\tilde{R} + \Delta R)\tilde{x} \\ &= (\tilde{Q}\tilde{R} + \Delta Q\tilde{R} + \tilde{Q}\Delta R + \Delta Q\Delta R)\tilde{x} \\ &= (A + \Delta\bar{A})\tilde{x}, \end{aligned}$$

pri čemu je $\Delta\bar{A} = \Delta A + \Delta Q\tilde{R} + \tilde{Q}\Delta R + \Delta Q\Delta R$ i ΔA je pogreška izračunate QR dekompozicije dane u dijelu a) teorema 5.7.

Proučit ćemo sva četiri izraza iz definicije $\Delta\bar{A}$, tj. proučavat ćemo omjer norme svakog izraza s normom matrice A. Za prvi izraz $\|\Delta A\|/\|A\|$ primjenom teorema 5.7 slijedi da je $\|\Delta A\|/\|A\| = \mathcal{O}(\varepsilon_m)$.

Budući da je matrica $\tilde{Q}$ ortogonalna, imamo $\tilde{R} = \tilde{Q}^*(A + \Delta A)$ i stoga

$$\frac{\|\tilde{R}\|}{\|A\|} \leq \|\tilde{Q}^*\| \cdot \frac{\|A + \Delta A\|}{\|A\|} = \mathcal{O}(1).$$

Zbog toga vrijedi

$$\frac{\|\Delta Q\tilde{R}\|}{\|A\|} \leq \|\Delta Q\| \cdot \mathcal{O}(1) = \mathcal{O}(\varepsilon_m).$$

Slično se može vidjeti da vrijedi

$$\frac{\|\tilde{Q}\Delta R\|}{\|A\|} \leq \|\tilde{Q}\|\frac{\|\Delta R\|}{\|R\|}\frac{\|R\|}{\|A\|} = \mathcal{O}(\varepsilon_m)$$

i

$$\frac{\|\Delta Q \Delta R\|}{\|A\|} \leq \|\Delta Q\| \frac{\|\Delta R\|}{\|R\|} \frac{\|R\|}{\|A\|} = \mathcal{O}(\varepsilon_m^2).$$

Uzimajući sve zajedno u obzir vidimo da vrijedi $\|\Delta \bar{A}\|/\|A\| = \mathcal{O}(\varepsilon_m)$. □

Teorem 5.8 pokazuje da je Householderova metoda za rješavanje SLJ povratno stabilna. Zbog jednostavnosti smo u našoj analizi izostavili konstantu u ocjeni stabilnosti $\|\Delta \bar{A}\|/\|A\| = \mathcal{O}(\varepsilon_m)$. Detaljnija bi analiza pokazala da u toj ocjeni postoji dodatna konstanta koja nije oblika eksponencijalne funkcije za razliku od metode GEDP koja posjeduje konstantu, koja je osjetljiva na eksponencijalni rast, s gornjom međom 2^{n-1} (vidi lemu 5.4 iz poglavlja 5.3.1).

5.4.4 Zadaci

1. Neka je
$$A = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}.$$
a) Odredite QR dekompoziciju matrice A, koristeći Gram-Schmidtov postupak i postupak primjenom Householderovih matrica.
b) Izračunajte broj uvjetovanosti matrice A za norme $\|\cdot\|_\infty$, $\|\cdot\|_2$ odnosno $\|\cdot\|_1$.

2. a) Neka je $A = a \otimes b$. Izračunajte sve svojstvene vrijednosti i svojstvene vektore matrice A. Kad će A biti normalna matrica?
b) Neka je $H \in \mathbb{R}^{n\times n}$ Householderova matrica. Pokažite da je samo jedna svojstvena vrijednost matrice H jednaka -1, dok su sve ostale $+1$ s mnogostrukošću $(n-1)$. Kolika je $\det(H)$?

3. Odredite asimptotski broj računskih operacija QR algoritma u slučaju da izračunavamo cijelu matricu Q.

4. Neka je $u = \begin{pmatrix} 2 & 2 & -4 & 5 \end{pmatrix}^T$. Konstruirajte Householderovu matricu H tako da poništi
a) sve elemente vektora u osim elementa u_1.
b) samo treću i četvrtu komponentu vektora u.
Izračunajte Hu.
[Rješenje: a) $Hu = \begin{pmatrix} -7 & 0 & 0 & 0 \end{pmatrix}$, b) $Hu = \begin{pmatrix} 2 & -3\sqrt{5} & 0 & 0 \end{pmatrix}$.]

5. Odredite reduciranu QR dekompoziciju matrice

$$A = \begin{pmatrix} -4 & 3 \\ 8 & 3 \\ 8 & 12 \end{pmatrix}$$

[Rješenje: $\hat{R} = \begin{pmatrix} 12 & 9 \\ 0 & 9 \end{pmatrix}$, $\hat{Q} = \begin{pmatrix} -\frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{pmatrix}$.]

6. Primjenom Householderovih matrica odredite QR dekompoziciju matrice A i riješite sustav $Ax = b$, ako je

$$A = \begin{pmatrix} 10 & 9 & 18 \\ 20 & -15 & -15 \\ 20 & -12 & 51 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 20 \\ -43 \end{pmatrix}.$$

[Rješenje: $R = \begin{pmatrix} -30 & 15 & -30 \\ 0 & 15 & 15 \\ 0 & 0 & 45 \end{pmatrix}$, $Q = \frac{1}{15}\begin{pmatrix} -5 & 14 & -2 \\ -10 & -5 & -10 \\ -10 & -2 & 11 \end{pmatrix}$, $x = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$]

Poglavlje 6

Iterativne metode

U dosadašnjim razmatranjima proučavali smo tzv. direktne metode za izračunavanje rješenja SLJ. Osnovno svojstvo tih metoda jest da je njima za izračunavanje rješenja $x = A^{-1}b$ potrebno $\Theta(n^3)$ operacija. U ovom ćemo poglavlju proučavati metode koje ne određuju rješenje direktno nego iterativno (postupno). Takvim se metodama konstruira niz $x^{(k)}$, $k \in \mathbb{N}$ za koji vijedi

$$x^{(k)} \to x \text{ za } k \to \infty.$$

Takve metode nazivamo iterativne metode, a kako ćemo vidjeti postoje matrice A s posebnom strukturom kod kojih pogreška $\|x^{(k)} - x\|$ postaje dovoljno mala nakon znatno manje operacija od $\Theta(n^3)$.

6.1 Linearne metode

Osnovna ideja iterativnih metoda jest napisati matricu A u obliku $A = M + N$, pri čemu je matrica M izabrana tako da se njen inverz računa lakše od inverza matrice A. Tada sustav $Ax = b$ prelazi u sustav $(M + N)x = b$, odnosno slijedi da je $Mx = b - Nx$. Tada za zadanu $(k-1)$-vu iteraciju $x^{(k-1)}$, k-tu iteraciju $k \in \mathbb{N}$ (k-tu aproksimaciju rješenja) definiramo sa:

$$Mx^{(k)} = b - Nx^{(k-1)}. \tag{6.1}$$

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

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

stoga za taj limes x vidimo da vrijedi $Ax = b$. To nam pokazuje da u slučaju konvergencije niza (6.1) njegov limes predstavlja rješenje SLJ.

U svrhu proučavanja konvergencije iterativne metode, proučavat ćemo ponašanje pogreške $e^{(k)} = x^{(k)} - x$, gdje je x točno (egzaktno rješenje) SLJ. Koristeći navedene oznake dobivamo

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

iz čega slijedi $e^{(k)} = -M^{-1}Ne^{(k-1)}$. To znači da će promatrana metoda konvergirati ako $e^{(k)} \to 0$ za $k \to \infty$.

Prije nego nastavimo s proučavanjem konvergencije iterativnih metoda potrebno je uvesti nekoliko pojmova koji će nam poslužiti kao teorijska osnova za njihovo proučavanje.

Prvi se rezultat odnosi na Jordanovu kanonski oblik matrice. Rezultat ćemo iskazati u obliku teorema bez dokaza.

Definicija 6.1 *Jordanov blok $J_k(\lambda) \in \mathbb{C}^{k\times k}$ za $\lambda \in \mathbb{C}$ je matrica za koju vrijedi $J_k(\lambda)_{ii} = \lambda$, $J_k(\lambda)_{i,i+1} = 1$ i $J_k(\lambda)_{i,j} = 0$ za sve ostale $i, j = 1, ..., k$, tj.*

$$J_k(\lambda) = \begin{pmatrix} \lambda & 1 & 0 & \dots & 0 \\ 0 & \lambda & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda & 1 \\ 0 & 0 & \dots & \dots & \lambda \end{pmatrix}.$$

Jordanova matrica je blok diagonalna matrica $J \in \mathbb{C}^{n\times n}$ sljedećeg oblika

$$J = \begin{pmatrix} J_{n_1}(\lambda_1) & & & \\ & J_{n_2}(\lambda_2) & & \\ & & \ddots & \\ & & & J_{n_k}(\lambda_k) \end{pmatrix}$$

gdje je $\sum_{j=1}^{n} n_j = n$.

Teorem 6.1 (Jordanov kanonski oblik) *Za bilo koju kvadratnu matricu $A \in \mathbb{C}^{n\times n}$ postoji regularna matrica $S \in \mathbb{C}^{n\times n}$ i Jordanova matrica $J \in \mathbb{C}^{n\times n}$, koje zadovoljavaju jednakost*

$$A = SJS^{-1}\,,$$

pri čemu su dijagonalni elementi $\lambda_1, \dots, \lambda_k$ Jordanovih blokova J_{n_i}, $i = 1, \dots, k$, svojstvene vrijednosti matrice A.

Pomoću prethodnog teorema o Jordanovom kanonskom obliku matrice u sljedećoj lemi izvest ćemo jedno od važnih svojstava spektralnog radijusa matrice A.

Lema 6.1 *Neka je dana matrica $A \in \mathbb{C}^{n\times n}$ i broj $\varepsilon > 0$. Tada postoji vektorska norma na $\mathbb{C}^n$ takva da njoj pripadna inducirana matrična norma zadovoljava*

$$\rho(A) \leq \|A\| \leq \rho(A) + \varepsilon\,.$$

Dokaz. Iz teorema 2.2 slijedi $\rho(A) \leq \|A\|$, za bilo koju matričnu normu. Ostaje nam samo pokazati drugu nejednakost iz tvrdnje. Neka je sa $J = S^{-1}AS$ dan Jordanov oblik matrice A i neka je dana dijagonalna matrica $D_\varepsilon = \operatorname{diag}(1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{n-1})$. Tada je

$$(SD_\varepsilon)^{-1}A(SD_\varepsilon) = D_\varepsilon^{-1}JD_\varepsilon = \begin{pmatrix} \lambda_1 & \varepsilon & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & \ddots & \varepsilon & & & & & \\ & & & \lambda_1 & & & & & \\ & & & & \lambda_2 & \varepsilon & & & \\ & & & & & \ddots & \ddots & & \\ & & & & & & \ddots & \varepsilon & \\ & & & & & & & \lambda_2 & \\ & & & & & & & & \ddots \end{pmatrix}.$$

Definirajmo vektorsku normu $\|\cdot\|$ na $\mathbb{C}^n$ na sljedeći način:

$$\|x\| = \|(SD_\varepsilon)^{-1}x\|_\infty$$

za sve vektore $x \in \mathbb{C}^n$. Tada inducirana matrična norma zadovoljava

$$\begin{aligned} \|A\| &= \max_{x\neq 0} \frac{\|Ax\|}{\|x\|} = \max_{x\neq 0} \frac{\|(SD_\varepsilon)^{-1}Ax\|_\infty}{\|(SD_\varepsilon)^{-1}x\|_\infty} \\ &= \max_{y\neq 0} \frac{\|(SD_\varepsilon)^{-1}A(SD_\varepsilon)y\|_\infty}{\|y\|_\infty} \\ &= \|(SD_\varepsilon)^{-1}A(SD_\varepsilon)\|_\infty\,. \end{aligned}$$

Iz teorema 2.5 slijedi da se $\|\cdot\|_\infty$ neke matrice izračunava kao najveći zbroj apsolutnih vrijednosti elemenata po retcima. Prema tome, iz eksplicitnog

oblika matrice $(SD_\varepsilon)^{-1}A(SD_\varepsilon)$ koji smo gore izračunali vidimo da je $\|A\| \leq \max_i |\lambda_i| + \varepsilon = \rho(A) + \varepsilon$. Time je teorem dokazan. □

Sada se možemo vratiti proučavanju konvergencije iterativnog postupka (6.1). Prisjetimo se da za taj postupak pogreška $e^{(k)} = x^{(k)} - x$ zadovoljava $e^{(k)} = Ce^{(k-1)} = C^2e^{(k-2)} = \cdots = C^ke^{(0)}$ za sve $k \in \mathbb{N}$. To znači da će metoda konvergirati ako i samo ako $e^{(k)} \to 0$ za $k \to \infty$. Sljedeći teorem karakterizira konvergenciju iterativne metode pomoću svojstava spektralnog radijusa matrice

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

Teorem 6.2 *Neka je dana matrica $C \in \mathbb{C}^{n\times n}$ i vektori $e^{(0)} \in \mathbb{C}^n$ i $e^{(k)} = C^ke^{(0)} \in \mathbb{C}^n$, za sve $k \in \mathbb{N}$. Tada $e^{(k)} \to 0$ za svaki $e^{(0)} \in \mathbb{C}^n$ ako i samo ako je $\rho(C) < 1$.*

Dokaz. Prvo pretpostavimo da je $\rho(C) < 1$. Tada prema 6.1 možemo pronaći induciranu matričnu normu takvu da je $\|C\| < 1$ pa imamo

$$\|e^{(k)}\| = \|C^ke^{(0)}\| \leq \|C\|^k\|e^{(0)}\| \to 0$$

za $k \to \infty$. S druge strane, neka je $e^{(k)} \to 0$. Kada bi $\rho(C) \geq 1$, tada bi postojao barem jedan $e^{(0)} \in \mathbb{C}^n$ za koji vrijedi da je $Ce^{(0)} = \lambda e^{(0)}$ za neko $\lambda \in \mathbb{C}$ takvo da je $|\lambda| \geq 1$. Za taj vektor $e^{(0)}$ dobivamo

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

iz čega se vidi da $e^{(k)}$ ne konvergira ka 0 za $k \to \infty$ za taj $e^{(0)}$, što dovodi do kontradikcije. □

Napomena 6.1

a) Gornji teorem govori o tome da će iterativni postupak (6.1) konvergirati ako i samo ako je $\rho(C) < 1$. Brzina konvergencije je veća što je $\rho(C)$ manji.

b) Budući da je $\rho(C) \leq \|C\|$ za svaku matričnu normu $\|\cdot\|$, dovoljan uvjet za konvergenciju promatrane iterativne metode je $\|C\| < 1$, za neku matričnu normu.

U nastavku ćemo proučavati specifične metode kod kojih su matrice M i N iz (6.1) posebno konstruirane. Tu ćemo konstrukciju izvoditi na sljedeći

način: Za matricu $A \in \mathbb{C}^{n\times n}$ definiramo $n \times n$ matrice

$$L = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ a_{2\,1} & 0 & \dots & 0 & 0 \\ a_{3\,1} & a_{3\,2} & \dots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ a_{n\,1} & a_{n\,2} & \dots & a_{n\,n-1} & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{2\,2} & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & a_{n\,n} \end{pmatrix},$$

$$U = \begin{pmatrix} 0 & a_{1\,2} & a_{13} & \dots & a_{1n} \\ 0 & 0 & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 & a_{n-1\,n} \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}, \tag{6.2}$$

Vidimo da vrijedi $A = L + D + U$.

6.1.1 Jacobijeva metoda

Iterativna metoda koju dobivamo iz (6.1) za izbor matrica $M = D$ i $N = L + U$, gdje su L, D i U definirane kao u (6.2) zovemo *Jacobijeva metoda*. Uz takav izbor matrica M i N iterativni postupak poprima sljedeći oblik

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

za sve $k \in \mathbb{N}$, pa konvergencija promatrane metode ovisi o svojstvima matrice $C = -M^{-1}N = -D^{-1}(L + U)$. Budući da je D dijagonalna matrica, njen inverz se izračunava trivijalno, tako da konstrukcija matrice C nije *skupa*.

Označimo li sa $x^{(k)} = \left(x_1^{(k)}, \ \dots, \ x_n^{(k)}\right)^T$, tada se $x^{(k+1)}$ aproksimacija rješenja SLJ pomoću **Jacobijeve metode** dobiva kao rješenje sustava:

$$\begin{aligned} x_1^{(k+1)} &= c_{1\,2}\, x_2^{(k)} + c_{1\,3}\, x_3^{(k)} + \dots + c_{1\,n}\, x_n^{(k)} + \beta_1 \\ x_2^{(k+1)} &= c_{2\,1}\, x_1^{(k)} + c_{2\,3}\, x_3^{(k)} + \dots + c_{2\,n}\, x_n^{(k)} + \beta_2 \\ &\vdots \\ x_n^{(k+1)} &= c_{n\,1}\, x_1^{(k)} + c_{n\,2}\, x_2^{(k)} + \dots + c_{n\,n-1}\, x_{n-1}^{(k)} + \beta_n, \end{aligned}$$

gdje su $c_{ij} = -\dfrac{a_{ij}}{a_{ii}}, \beta_i = \dfrac{b_i}{a_{ii}}$.

Teorem 6.3
a) Jacobijeva metoda konvergira za sve matrice A za koje vrijedi da je

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \tag{6.3}$$

za $i = 1, \ldots, n$. (Ovaj uvjet nazivamo stroga dijagonalna dominacija po retcima).
b) Jacobijeva metoda konvergira za sve matrice A za koje vrijedi da je

$$|a_{jj}| > \sum_{i \neq j} |a_{ij}| \tag{6.4}$$

za $j = 1, \ldots, n$. (Ovaj uvjet nazivamo stroga dijagonalna dominacija po stupcima).

Dokaz. a) Kao što smo već vidjeli, elementi matrice $C = -D^{-1}(L+U)$ su oblika

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

Koristeći teorem 2.5 vidimo da je

$$\|C\|_\infty = \max_{i=1,\ldots,n} \frac{1}{|a_{ii}|} \sum_{j \neq i}^{n} |a_{ij}| .$$

Stoga stroga dijagonalna dominacija po retcima povlači $\|C\|_\infty < 1$, što dalje povlači da je $\rho(A) < 1$, a to po teoremu 6.2 znači da metoda konvergira.

b) Ako je matrica A strogo dijagonalno dominantna po stupcima, tj. ako vrijedi (6.4), tada je matrica A^* strogo dijagonalno dominantna po retcima, tj. vrijedi (6.3) i metoda konvergira za A^*. Prema teoremu 6.2 to znači da je $\rho(-D^{-*}(L^* + U^*)) < 1$. Budući da za bilo koju matricu C, matrice C, C^* i $D^{-1}C^*D$ imaju iste spektre, zaključujemo da $\rho(-D^{-1}(L+U)) = \rho(-D^{-*}(L^*+U^*)) < 1$, (zaista, vrijedi $\rho(-D^{-1}(L+U)) = \rho(-D^{-1}(L+U)D^{-1}D) = \rho(-(L+U)D^{-1}) = \rho(-D^{-*}(L^*+U^*))$ što povlači konvergenciju metode za matricu A. □

6.1.2 Gauss-Seidelova metoda

Primijetimo da se pri izračunavanju $(k+1)$-ve iteracije rješenja $x^{(k+1)}$ pomoću Jacobijeve metode svaka komponenta vektora $x^{(k+1)}$ izračunavala pomoću komponenata vektora $x^{(k)}$. Gauss-Seidelova metoda je poboljšanje Jacobijeve metode u sljedećem smislu.

Pretpostavimo da je $x^{(k)} = \left(x_1^{(k)}, \quad \ldots, \quad x_n^{(k)}\right)^T$ k-ta aproksimacija rješenja. Tada se $(k+1)$-va aproksimacija dobiva kao rješenje sustava:

$$\begin{aligned}
a_{1\,1}x_1^{(k+1)} + a_{1\,2}\,x_2^{(k)} + \cdots + a_{1\,n}\,x_n^{(k)} &= b_1 \\
a_{2\,1}x_1^{(k+1)} + a_{2\,2}\,x_2^{(k+1)} + \cdots + a_{2\,n}\,x_n^{(k)} &= b_2 \\
&\vdots \\
a_{n\,1}x_1^{(k+1)} + a_{n\,2}\,x_2^{(k+1)} + \cdots + a_{n\,n}\,x_n^{(k+1)} &= b_n\,.
\end{aligned}$$

Važno je primijetiti da se u gornjem sustavu prva komponenta $x_1^{(k+1)}$ vektora $x^{(k+1)}$ izračunava iz prve jednadžbe pomoću komponenti k-te aproksimacije rješenja. Ali tu komponentu odmah koristimo u drugoj, trećoj, ..., n-toj jednadžbi. Nadalje, komponentu $x_2^{(k+1)}$ koristimo u trećoj, četvrtoj, ..., n-toj jednadžbi, itd. Taj sustav možemo zapisati u matričnom obliku

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

odnosno

$$(L+D)\,x^{(k+1)} = -U\,x^{(k)} + b\,. \tag{6.5}$$

Tako da formula iterativnog postupka ima oblik:

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

Nedostatak ove formule je u tome što u gornjem algoritmu trebamo odrediti inverz matrice $L+D$ što je teže nego odrediti inverz matrice D. Zato se u praksi radi malo drukčije. Pomnožimo li jednadžbu (6.5) s D^{-1}, dobijamo:

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

Odatle je

$$x^{(k+1)} = -D^{-1}\,Lx^{(k+1)} - D^{-1}\,U\,x^{(k)} + D^{-1}\,b\,.$$

Tako dolazimo do algoritma koji nazivamo **Gauss-Seidelova** metoda.

Teorem 6.4 *Ako vrijedi da je matrica A strogo dijagonalno dominantna matrica po retcima, tj. A zadovoljava (6.3), onda Gauss-Seidelova metoda konvergira.*

6.1.3 Zadaci

1. Može li se sustav $Ax = b$ riješiti Jacobijevom metodom? Ako može, odredite potreban broj koraka tako da pogreška u svakoj koordinati bude manja od 10^{-3}, ako je početna aproksimacija $x^{(0)} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}^T$ i

$$A = \begin{pmatrix} 10 & 1 & 0 & 1 \\ 1 & 10 & 1 & 0 \\ 0 & 1 & 10 & 1 \\ 1 & 0 & 1 & 10 \end{pmatrix}, b = \begin{pmatrix} -8 \\ 0 \\ 12 \\ 20 \end{pmatrix}.$$

[Rješenje: $k = 5$.
Uputa: ako metoda konvergira, onda vrijedi sljedeća formula $\|x - x^{(k)}\| \leq \frac{\|C\|^k}{1-\|C\|}\|x^{(1)} - x^{(0)}\|$,]

2. Može li se sustav $Ax = b$ riješiti Jacobijevom metodom? Ako može, odredite potreban broj koraka tako da apsolutna pogreška metode u $\|\cdot\|_1$ bude manja od 10^{-3}, ako je

$$A = \begin{pmatrix} 4 & 2 & 3 \\ 5 & 10 & 2 \\ 1 & 1 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 7 \\ 4 \end{pmatrix} \quad \text{i} \quad x^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

[Rješenje: $k = 207$.]

3. a) Odredite matricu P tako da se sustav $Ax = b$, gdje je $A = \begin{pmatrix} 5 & 8 & 1 \\ 1 & 2 & 10 \\ 10 & 1 & 0 \end{pmatrix}$, $b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, može riješiti primjenom Jacobijeve metode na sustav $PAx = Pb$.
b) Odredite potreban broj koraka da bi pogreška aproksimacije bila manja od 10^{-3} u $\|\cdot\|_\infty$ ako je $x^{(0)} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$.

c) Odredite prve dvije aproksimacije ako je $x^{(0)} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$.
[Rješenje: $P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $k = 22$, $x^{(1)} = \begin{pmatrix} \frac{1}{10} \\ \frac{1}{8} \\ \frac{1}{10} \end{pmatrix}$, $x^{(2)} = \begin{pmatrix} \frac{7}{80} \\ \frac{1}{20} \\ \frac{13}{200} \end{pmatrix}$.]

4. Neka su matrice A, b i $x^{(0)}$ kao u 1. zadatku. Gauss-Seidelovom metodom riješite sustav $Ax = b$ tako da pogreška ne prijeđe 0.05 u normi $\|\cdot\|_\infty$.
[Rješenje: potreban broj koraka za danu toleranciju je 3,
$x^{(3)} = \begin{pmatrix} -0.9981 & -0.00077 & 1.00009 & 1.9998 \end{pmatrix}$.]

6.2 Metoda najbržeg silaska i konjugiranih gradijenata

U ovom ćemo poglavlju proučavati jednu od iterativnih metoda za rješavanje SLJ $Ax = b$, u slučaju kada je A pozitivno definitna matrica. Za razliku od metoda jednostavnih iteracija, metoda koju ćemo obrađivati u ovom poglavlju spada u takozvane aproksimacije iz Krylovljevih potprostora.

Standardni rezultat iz linearne algebre kaže da svaka matrica poništava svoj karakteristični i minimalni polinom. Za $A \in \mathbb{C}^{n\times n}$ i $b \in \mathbb{C}^n$ to možemo zapisati na sljedeći način

$$\kappa_A(A) = a_0 I + a_1 A + \ldots + a_{n-1} A^{n-1} + a_n A^n \,,$$

gdje je $\kappa_A(\lambda) = \det(A - \lambda I) = \sum_{i=0}^{n} a_i \lambda^i$ karakteristični polinom matrice A. Slično možemo napisati da je $\mu_A(A) = 0$, gdje je μ_A minimilni polinom stupnja manjeg od ili jednakog n. Prepostavimo li da je matrica A regularna, to znači da nula ne može biti korijen karakterističnog polinoma, pa je $a_0 \neq 0$. Odavde jednostavnim računom možemo dobiti da vrijedi

$$\begin{aligned} &-\frac{1}{a_0}\left(a_1 I + \ldots + a_{n-1} A^{n-2} + a_n A^{n-1}\right) A = \\ &= A\left(-\frac{1}{a_0}\right)\left(a_1 I + \ldots + a_{n-1} A^{n-2} + a_n A^{n-1}\right) = I \,, \end{aligned}$$

to jest, da je

$$A^{-1} = -\frac{1}{a_0}\left(a_1 I + \ldots + a_{n-1}A^{n-2} + a_n A^{n-1}\right). \tag{6.6}$$

Budući da rješenje sustava $Ax = b$ možemo zapisati kao $x = A^{-1}b$, uz uvažavanje prethodnog zapisa za A^{-1} možemo zaključiti da je

$$x = -\frac{a_1}{a_0}b - \cdots - \frac{a_{n-1}}{a_0}A^{n-2}b - \frac{a_n}{a_0}A^{n-1}b\,,$$

odnosno

$$x \in \operatorname{span}\{b, Ab, \ldots, A^{n-1}b\} = \mathcal{K}_n(A, b). \tag{6.7}$$

Prostor koji se pojavljuje na desnoj strani u (6.7) zovemo Krylovljevim prostorom matrice A i inicijalnog vektora b. Upravo iz (6.7) dobivamo ideju za metode rješavanja sustava linearnih jednadžbi koje bi se temeljile na aproksimacijama iz Krylovljevih potprostora. Naime, kako je vektor b jedini vektor izravno vezan za problem rješavanja sustava $Ax = b$, čini nam se prirodnim uzeti neki „višekratnik" od b kao prvu aproksimaciju [1] rješenja, tj.

$$x_1 \in \operatorname{span}\{b\}\,.$$

Nakon toga računamo produkt Ab i zahtijevamo da nam je sljedeća aproksimacija jednaka nekoj linearnoj kombinaciji od b i Ab, tj.

$$x_2 \in \operatorname{span}\{b, Ab\}\,.$$

Taj se proces nastavlja, tako da nam aproksimacija u k-tom koraku zadovoljava

$$x_k \in \operatorname{span}\{b, Ab, \ldots, A^{k-1}b\}\,, k = 1, 2, \ldots\,.$$

Metoda, naravno, mora odrediti kriterij po kojemu biramo vektore iz pojedinih Krylovljevih potprostora u svakom koraku, tako da bismo u optimalnom slučaju mogli dobiti rješenje u k-tom koraku, za $k \ll n$. Ako pak račun vršimo na računalu, u obzir još moramo uzeti i pogreške zaokruživanja. Konkretne bi se metode zasnivale na aproksimacijama upravo ovako definiranih vektora x_k.

[1] Zbog lakšeg zapisivanja, u ovom ćemo poglavlju k-tu aproksimaciju označavati sa x_k.

Prije svega promotrimo jednu vrlo jednostavnu činjenicu. Ako je x_k aproksimacija rješenja u k-tom koraku neke iterativne metode za rješavanje linearnih sustava, i ako u $(k+1)$-om koraku uzmemo

$$x_{k+1} = x_k + A^{-1}(b - Ax_k)\,, \quad k = 1, 2, \ldots$$

tada je $x_{k+1} = A^{-1}b$ rješenje sustava. Budući da je izračunavanje vektora $A^{-1}(b - Ax_k)$ ekvivalentno problemu rješavanja polaznog sustava, korekciju vektora x_k radit ćemo pomoću neke njegove aproksimacije, koja se lakše izračunava i koja će nas držati unutar Krylovljevih potprostora.

Zato najprije pretpostavimo da polazimo od iteracije jednostavnog oblika

$$x_{k+1} = x_k + \alpha_k(b - Ax_k)\,, \tag{6.8}$$

gdje je α_k parametar koji određuje točku na pravcu koji prolazi kroz točku x_k i pruža se u smjeru vektora $b - Ax_k$. Promotrimo kako se na taj način možemo približiti vektoru iz (6.7). Lako se vidi da je na temelju iteracije (6.8) za $k = 0, 1, 2, \ldots$

$$x_k \in x_0 + \operatorname{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\}\,, \tag{6.9}$$

$$r_k \in r_0 + \operatorname{span}\{Ar_0, A^2r_0, A^3r_0, \ldots, A^{k}r_0\}\,, \tag{6.10}$$

pri čemu je $r_k = b - Ax_k$ rezidual k-te iteracije, a sa $e_k = A^{-1}r_k = A^{-1}b - x_k$ označavamo pogrešku. Prostor koji se pojavljuje na desnoj strani (6.9) također je Krylovljev potprostor, ali određen matricom A i vektorom r_0, koji ne mora biti jednak vektoru b. Međutim, ako napišemo da je prema jednakosti (6.6) inverz matrice A jednak $A^{-1} = q_{k-1}(A)$, gdje je $q_{k-1}(\lambda)$ polinom stupnja $k-1$ za neki $k \leq n$, tada za bilo koji r_0 u prostoru na desnoj strani izraza (6.9) možemo naći vektor oblika $x_0 + q_{k-1}(A)r_0 = x_0 + A^{-1}r_0 = A^{-1}b$ koji je rješenje linearnog sustava $Ax = b$.

Općenito će iterativne metode biti oblika

$$x_k = x_{k-1} + z_{k-1}$$

ili

$$x_k = x_0 + w_k,$$

pri čemu će se z_{k-1} odnosno w_k birati tako da x_k zadovoljava (6.9) i neki uvjet optimalnosti, najčešće vezan uz normu reziduala ili neku normu pogreške.

6.2.1 Metoda najbržeg silaska

Ako promatramo iterativnu metodu danu u obliku

$$x_{k+1} = x_k + \alpha_k(b - Ax_k) = x_k + \alpha_k r_k\,, \tag{6.11}$$

jedan od mogućih načina odabira parametra α_k u iteraciji (6.11) je takav da u k-tom koraku iterativnog postupka dobijemo pogrešku $e_{k+1} = x - x_{k+1} = A^{-1}r_{k+1}$ s minimalnom A-normom, pri čemu je A, matrica sustava $Ax = b$, pozitivno definitna. A-norma definira se kao $\|z\|_A = \sqrt{z^*Az}$. Neka je zadana funkcija

$$\begin{aligned} F(\alpha_k) \equiv \frac{1}{2}\|e_{k+1}\|_A^2 &= \frac{1}{2}\|x - x_{k+1}\|_A^2 = \frac{1}{2}(x - x_{k+1})^* A(x - x_{k+1}) \\ &= \frac{1}{2}(A^{-1}r_k - \alpha_k r_k)^* A(A^{-1}r_k - \alpha_k r_k) \\ &= \frac{1}{2}(r_k - \alpha_k Ar_k)^* A^{-1}(r_k - \alpha_k Ar_k)\,, \end{aligned}$$

gdje je x točno rješenje ($Ax = b$), a x_k k-ta iteracija dana sa (6.11) i $e_k \equiv x - x_k = A^{-1}r_k$.

Primijetite da je gornja funkcija $F(\alpha_k) \equiv \frac{1}{2}e_{k+1}^* Ae_{k+1} : \mathbb{R} \to \mathbb{R}$. Parametar α_k biramo tako da bude točka minimuma funkcije $F(\alpha_k)$. Stoga ćemo njenu derivaciju izjednačit s nulom, što odgovara minimalnoj A-normi vektora e_{k+1}. Ovu metodu nazivamo *metodom najbržeg silaska*:

$$\begin{aligned} F'(\alpha_k) &= \frac{1}{2}(-Ar_k)^* A^{-1}(r_k - \alpha_k Ar_k) + \frac{1}{2}(r_k - \alpha_k Ar_k)^* A^{-1}(-Ar_k) \\ &= -r_k^* r_k + \alpha_k r_k^* Ar_k\,, \end{aligned}$$

iz čega slijedi da će biti $F'(\alpha_k) = 0$ za

$$\alpha_k = \frac{r_k^* r_k}{r_k^* Ar_k}\,. \tag{6.12}$$

U ovom je slučaju r_{k+1} okomit na r_k. (Zaista, $r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha_k r_k) = r_k - \alpha_k Ar_k$, te je $r_{k+1}^* r_k = r_k^* r_k - \alpha_k r_k^* Ar_k$, što za izbor α_k iz (6.12) daje $r_{k+1}^* r_k = 0$). Slično se može pokazati da je i e_{k+1} okomit na r_k u skalarnom produktu induciranom matricom A, tj. da vrijedi $e_{k+1}^* Ar_k = 0$. Važno je još primijetiti da tako dugo dok nismo našli egzaktno rješenje, to jest dok je $r_k \neq 0$, α_k je strogo veći od nule. Zbog okomitosti e_{k+1} i r_k slijedi

$$\|e_k\|_A^2 = \|e_{k+1}\|_A^2 + \alpha_k^2\|r_k\|_A^2 > \|e_{k+1}\|_A^2\,,$$

odakle se vidi da se A-norma pogreške smanjuje u svakom koraku.

Može se pokazati da iz analize konvergencije metode najbržeg silaska slijedi sljedeća ocjena:

$$\|e_k\|_A \leq \left(\frac{\kappa_2(A)-1}{\kappa_2(A)+1}\right)^k \|e_0\|_A \,,$$

gdje je $\kappa_2(A) = \lambda_{\max}/\lambda_{\min}$ uvjetovanost matrice A.

6.2.2 Metoda konjugiranih gradijenata

Jedan od osnovnih nedostataka metoda najbržeg silaska jest da ne osigurava moguće ponavljanje smjerova, tj. metoda ne vodi brigu o tome hoće li se neki od generiranih smjerova ponoviti.

Stoga, da metoda ne bi radila korake u smjeru kojim je neki raniji korak prošao, mi u metodi najbržeg silaska, unaprijed odabiremo skup A-ortogonalnih vektora, odnosno smjerove traganja $d_0, d_1, \ldots, d_{n_1}$ i to tako da vektori

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

za $k = 0, 1, 2, \ldots, n$ budu međusobno ortogonalni u A-skalarnom produktu. To znači da

$$0 = \langle d_k, d_{k-1}\rangle_A = d_k^* A d_{k-1} = r_k^* A d_{k-1} + \beta_k d_{k-1}^* A d_{k-1} \,,$$

što povlači

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

S druge strane u svakom koraku novu iteraciju (aproksimaciju rješenja) biramo kao

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

s minimalnom A-normom pogreške.

Prije nego izračunamo minimum funkcije $F(\alpha_k) = \frac{1}{2}\|e_{k+1}\|_A^2$, primjetimo da vrijedi

$$e_{k+1} = x - x_{k+1} = e_k - \alpha_k d_k \,, \qquad r_{k+1} = A e_{k+1} = r_k - \alpha_k A d_k \,.$$

Izraz za α_k dobit ćemo izjednačavanjem $F'(\alpha_k) = 0$, tj.

$$F'(\alpha_k) = \frac{d}{d\alpha_k}\frac{1}{2}(e_k - \alpha_k d_k)^* A (e_k - \alpha_k d_k) = -\, d_k^* A e_k + \, \alpha_k \, d_k^* A d_k \,,$$

što nakon izjednačavanja s nulom daje

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

Uzimajući u obzir $r_k = Ae_k$ dobivamo

$$\alpha_k = \frac{d_k^* r_k}{d_k^* A d_k} = \frac{\langle d_k, r_k\rangle}{\langle Ad_k, d_k\rangle}\,. \tag{6.13}$$

Primjetite da je $r_k^* A d_k = d_k^* A d_k$, što slijedi iz $r_k^* A d_k = (d_k - \alpha_k d_{k-1}^*) A d_k = d_k^* A d_k$ jer su $d_0, \ldots, d_n$ A- ortogonalni. Slijedi da je

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

Ovako dobivena metoda naziva se *metoda konjugiranih smjerova*.

Metoda konjugiranih gradijenata (CG) je zapravo metoda konjugiranih smjerova kod koje se smjerovi traganja konstruiraju primjenom Gram—Schmidtove metode A-ortogonalnosti na reziduale, tj. uzima se da $d_0, d_1, \ldots, d_i$ čine A-ortonormiranu bazu za $\operatorname{span}\{r_0, r_1, \ldots, r_{k-1}\}$. Iz te činjenice slijedi:

$$\operatorname{span}\{d_0, d_1, \ldots, d_{k-1}\} = \operatorname{span}\{r_0, r_1, \ldots, r_{k-1}\}\,.$$

U nastavku ćemo pokazati da za $r_{k+1} = r_k - \alpha_k A d_k$ i $e_{k+1} = e_k - \alpha_k d_k$, vrijedi

$$r_i^* d_j = e_i^* A d_j = 0 \qquad \text{za} \quad j < i\,. \tag{6.14}$$

Za $j = i - 1$ imamo, koristeći (6.13), dobivamo

$$r_i^* d_{i-1} = (r_{i-1} - \alpha_{i-1} A d_{i-1})^* d_{i-1} = r_{i-1}^* d_{i-1} - \alpha_{i-1} d_{i-1}^* A d_{i-1} = 0\,.$$

Nadalje, sličnim računom, koristeći (6.13) i jednakost $r_{i-1}^* r_{i-1} = d_{i-1}^* r_{i-1}$, dobivamo

$$r_i^* r_{i-1} = d_{i-1}^* r_{i-1} - \alpha_{i-1} d_{i-1}^* A r_{i-1} = 0\,.$$

Nastavljajući na sličan načini dobivamo da (6.14) vrijedi. Primjetimo da također vrijedi

$$r_i^* r_j = 0 \qquad \text{za} \quad 0 \le j < i\,. \tag{6.15}$$

Također iz (6.14) slijedi da su i-te pogreške (e_i) okomite u A-skalarnom produktu na j-te smjerove (d_j) za sve $i > j$.

U nastavku promotrimo jednakost

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

iz koje slijedi

$$r_k^* A d_i = \frac{1}{\alpha_i}(r_k^* r_i - r_k^* r_{i+1}). \tag{6.16}$$

Za $i < k-1$ lijeva strana u (6.16) jednaka je 0, pa ako smjerove $d_0, d_1, \ldots, d_i$ konstruiramo Gram—Schmidtovom metodom A-ortogonalnosti imat ćemo

$$d_k = r_k - \sum_{i=0}^{k-1} \frac{r_k^* A d_i}{d_i^* A d_i}\, d_i = r_k - \frac{r_k^* A d_{k-1}}{d_{k-1}^* A d_{k-1}}\, d_{k-1}\,.$$

Budući da je $d_k = r_k + \beta_k d_{k-1}$, vidimo da je β_k dano sa

$$\beta_k \;=\; -\frac{r_k^* A d_{k-1}}{d_{k-1}^* A d_{k-1}}\,. \tag{6.17}$$

Pokažimo da se β_k može računati samo pomoću reziduala. Zaista

$$\begin{aligned}\beta_k &= -\frac{r_k^* A d_{k-1}}{d_{k-1}^* A d_{k-1}} = -\frac{\frac{1}{\alpha_{k-1}}(r_k^* r_{k-1} - r_k^* r_k)}{\frac{1}{\alpha_{k-1}}(d_{k-1}^* r_{k-1} - d_{k-1}^* r_k)} \\ &= \frac{r_k^* r_k}{d_{k-1}^* r_{k-1}} = \frac{r_k^* r_k}{r_{k-1}^* r_{k-1}}\,.\end{aligned}$$

U posljednjoj jednakosti smo iskoristili da je $r_{k-1}^* r_{k-1} = d_{k-1}^* r_{k-1}$.

Algoritam CG (*Metoda konjugiranih gradijenta*)

ulaz: dana je početna iteracija x_0, $d_0 = r_0 = b - Ax_0$.

izlaz: x_k, za koji vrijedi $Ax_k \approx b$.

for $k = 1, 2, \ldots, n$

izračunaj Ad_{k-1},

$\alpha_{k-1} = \dfrac{r_{k-1}^* r_{k-1}}{d_{k-1}^* A d_{k-1}}$,

$x_k = x_{k-1} + \alpha_{k-1} d_{k-1}$,

$r_k = r_{k-1} - \alpha_{k-1} A d_{k-1}$,

$\beta_k = \dfrac{r_k^* r_k}{r_{k-1}^* r_{k-1}}$,

$d_k = r_k + \beta_k d_{k-1}$

end for

Teorem 6.5 (Brzina konvergencije) *Metoda konjugiranih gradijenata zadovoljava*

$$\|e_k\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \cdot \|e_0\|_A$$

6.2.3 Zadaci

1. Izračunajte prve tri aproksimacije rješenja sustava $Ax = b$ uz pomoć metode najbržeg silaska i metode konjugiranih gradijenata ako je

$$A = \begin{pmatrix} 4 & 2 & -1 \\ 2 & 8 & 4 \\ -1 & 4 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 30 \\ 37 \end{pmatrix}, \quad x^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

[Rješenje: za metodu najbržeg silaska $x^{(3)} = \begin{pmatrix} 0.87853 & 2.08418 & 2.97173 \end{pmatrix}^T$
za metodu konjugiranih gradijenata $x^{(3)} = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}^T$]

Poglavlje 7

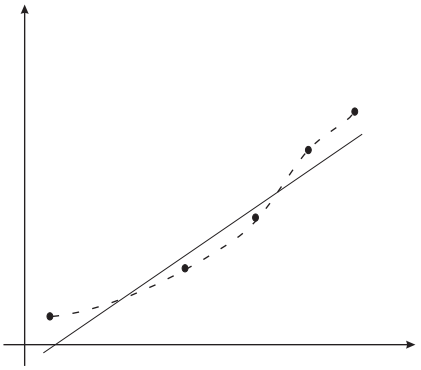
Problem najmanjih kvadrata

U ovom ćemo poglavlju proučavati problem rješavanja sustava koji ima više jednadžbi nego nepoznanica, tj. za zadane $A \in \mathbb{C}^{m\times n}$ i $b \in \mathbb{C}^m$, pri čemu je $m \geq n$, treba odrediti $x \in \mathbb{C}^n$ koji minimizira normu $\|Ax - b\|_2$.

Promatrajmo sljedeći primjer:

Primjer 7.1 *Pretpostavimo da su nam zadane vrijednosti funkcije samo u nekim točkama, tj. neka su $x_1, x_2, \ldots, x_n$ dane točke i neka su $f(x_1)$, $f(x_2)$, $\ldots$, $f(x_n)$ pripadne vrijednosti funkcije f u tim točkama. Želimo odrediti pravac $g(x) = ax + b$ koji najbolje aproksimira funkciju f.*

Na sljedećoj slici je funkcija (nacrtana iscrtkano) zadana svojim vrijednostima u nekim točkama i pravac koji je najbolja aproksimacija zadane funkcije u smislu metode najmanjih kvadrata.



Ako pretpostavimo da postoji pravac koji se u promatranim točkama podudara s funkcijskim vrijednostima, onda taj pravac mora zadovoljavati sljedeći

sustav

$$\begin{aligned} ax_1 + b &= f(x_1) \\ ax_2 + b &= f(x_2) \\ &\dots \\ ax_n + b &= f(x_n) \end{aligned}$$

što u matričnom zapisu ima oblik

$$A\mathbf{x} = y, \qquad \textit{gdje su} \qquad A = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix}, \ \mathbf{x} = \begin{pmatrix} a \\ b \end{pmatrix}, \ y = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix}.$$

Očito je da općenito gornji sustav ne mora imati rješenje, pa je prirodno da promatrani pravac određujemo tako da odredimo $\mathbf{x} \in \mathbb{C}^2$ *koji minimizira normu* $\|A\mathbf{x} - y\|_2$ *(što je ekvivalentno minimizaciji* $\|A\mathbf{x} - y\|_2^2$*).*

Označimo li sa $S(a,b) = \|A\mathbf{x} - y\|_2^2$*, vidimo da desna strana ove jednakosti predstavlja sumu kvadrata razlika funkcija* $y \equiv f(x)$ *i* $g(x) = ax + b$ *u zadanim točkama, tj.*

$$S(a,b) = \sum_{k=1}^{n} \left[f(x_k) - g(x_k) \right]^2 = \sum_{k=1}^{n} \left[f(x_k) - ax_k - b \right]^2 .$$

Na taj smo način definirali funkciju $S(a,b) : \mathbb{R}^2 \to R$*, za koju vrijedi da je* $S(x,y) \geq 0$*.*

Nužan uvjet za postojanje ekstrema funkcije S *glasi*

$$\frac{\partial S}{\partial a} = 0, \qquad \frac{\partial S}{\partial b} = 0,$$

dakle

$$\sum_{k=1}^{n} \left[f(x_k) - a\, x_k - b \right] x_k = 0,$$

i

$$\sum_{k=1}^{n} \left[f(x_k) - a\, x_k - b \right] = 0,$$

što nakon sređivanja daje sustav jednadžbi

$$a \sum_{k=1}^{n} x_k^2 + b \sum_{k=1}^{n} x_k = \sum_{k=1}^{n} f(x_k)\, x_k,$$

i

$$a \sum_{k=1}^{n} x_k + n\, b = \sum_{k=1}^{n} f(x_k)$$

Iz gornjeg sustava izračunamo a i b. Iz prirode problema jasno je da funkcija S ima samo minimum, pa tako određeni parametri doista daju najbolju aproksimaciju funkcije f.

7.1 Normalne jednadžbe

U ovom poglavlju prvo ćemo izvesti teorijski kriterij za rješenje problema najmanjih kvadrata nakon čega ćemo prikazati tri različita algoritma za rješavanje tog problema.

Teorem 7.1 *Vektor $x \in \mathbb{C}^n$ minimizira $\|Ax-b\|_2$ za sve $A \in \mathbb{C}^{m\times n}$ i $b \in \mathbb{C}^m$ ako i samo ako je $Ax - b$ ortogonalno na $Im(A)$.*

Dokaz. Neka je $g(x) = \frac{1}{2}\|Ax - b\|_2^2$. Tada je minimiziranje $\|Ax - b\|_2$ ekvivalentno minimiziranju funkcije g.

(1) Pretpostavimo da je $Ax - b$ ortogonalno na $Im(A)$ i neka je $y \in \mathbb{C}^n$. Tada je

$$Ay - Ax = A(y - x) \perp Ax - b,$$

pa primjenom Pitagorinog teorema vidimo da je

$$\|Ay - b\|_2^2 = \|Ay - Ax\|_2^2 + \|Ax - b\|_2^2 \geq \|Ax - b\|_2^2,$$

za sve $y \in \mathbb{C}^n$. Stoga x minimizira g.

(2) Pretpostavimo da vektor x minimizira g. Tada za svaki $y \in \mathbb{C}^n$ imamo

$$\frac{\partial}{\partial\lambda} g(x + \lambda y) = \frac{1}{2}\left(\langle Ay, Ax + \lambda Ay - b\rangle + \langle Ax + \lambda Ay - b, Ay\rangle\right).$$

Budući x minimizira g, to znači da će funkcija $G(\lambda) = g(x+\lambda y)$ poprimiti minimum za $\lambda = 0$, odnosno vrijedi

$$0 = \frac{\partial}{\partial\lambda} g(x+\lambda y) = \frac{1}{2}\left(\langle Ay, Ax-b\rangle + \langle Ax-b, Ay\rangle\right) = Re\langle Ax-b, Ay\rangle.$$

Slično vrijedi i za

$$0 = \frac{\partial}{\partial\lambda} g(x+\lambda iy) = \frac{1}{2}\left(-i\langle Ay, Ax-b\rangle + i\langle Ax-b, Ay\rangle\right) = iIm\langle Ax-b, Ay\rangle.$$

Time smo pokazali da je $\langle Ax-b, Ay\rangle = 0$ pa je $Ax-b \perp Ay$ za sve $y \in \mathbb{C}^n$. □

Korolar 7.1 *Vektor $x \in \mathbb{C}^n$ rješava linearni problem najmanjih kvadrata ako i samo ako*

$$A^*Ax = A^*b. \tag{7.1}$$

Dokaz. Iz teorema 7.1 znamo da će x biti rješenje linearnog problem najmanjih kvadrata ako i samo ako $Ax-b \perp Im(A)$. To je ekvivalentno činjenici da je $Ax-b \perp a_i$ za svaki stupac a_i matrice A, odnosno $A^*(Ax-b) = 0$. □

Definicija 7.1 *Sustav linearnih jednadžbi (7.1) zovemo normalnim jednadžbama za problem najmanjih kvadrata.*

Promatrat ćemo razne algoritme za rješavanje linearnog problema najmanjih kvadrata. Prvi najčešće nazivamo *algoritam za rješavanje linearnog problema najmanjih kvadrata* (LPNK) *preko sustava normalnih jednadžbi* i on se izravno temelji na sustavu normalnih jednadžbi.

Algoritam za LPNK pomoću sustava normalnih jednadžbi.
ulaz: $A \in \mathbb{C}^{m\times n}$, $m \geq n$ i $rang(A) = n$, $b \in \mathbb{C}^m$
izlaz: $x \in \mathbb{C}^n$ koji minimizira $\|Ax-b\|_2$
1: izračunaj A^*A i A^*b
2: riješi $(A^*A)x = A^*b$

Napomena 7.1 *Izračunavanje produkata A^*A i A^*b zahtijeva asimptotski $\sim 2mn^2$ operacija za $m, n \to \infty$. Budući da je A^*A hermitska, trebamo izračunati samo pola elemenata što se može napraviti u $\sim mn^2$ operacija. Rješavanje $(A^*A)x = A^*b$ pomoću QR faktorizacije zahtijeva $\frac{4}{3}n^3$ operacija, što nam sve zajedno daje sljedeću asimptotsku ocjenu*

$$C(m,n) \sim mn^2 + \frac{4}{3}n^3.$$

Kod hermitskih matrica postoje metode za rješavanje sustava jednadžbi u manje koraka. Koristeći faktorizaciju Choleskog[1] dobije se asimptotska ocjena $C(m,n) \sim mn^2 + \frac{1}{3}n^3$.

7.2 Rješavanje LPNK pomoću SVD dekompozicije

7.2.1 Dekompozicija na singularne vrijednosti

Dekompozicija na singularne vrijednosti matrice jedna je od najkorištenijih dekompozicija u numeričkoj linearnoj algebri. Imamo sljedeću definiciju

Definicija 7.2 *Neka je $A \in \mathbb{C}^{m\times n}$ za $m,n \in \mathbb{N}$. Rastav matrice*

$$A = U\Sigma V^*$$

zovemo dekompozicija na singularne vrijednosti *A, ako su $U \in \mathbb{C}^{m\times m}$ i $V \in \mathbb{C}^{n\times n}$ unitarne, a $\Sigma \in \mathbb{C}^{m\times n}$ dijagonalna*

$$\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}),$$

pri čemu vrijedi $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(m,n)} \geq 0$, a brojeve $\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}$ zovemo singularne vrijednosti matrice A. Stupce matrice U zovemo lijevi, a stupce matrice V desni singularni vektori matrice A.

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

Sljedeći je teorem o egzistenciji dekompozicije na singularne vrijednosti.

[1] Matrica A je hermitska pozitivno definitna matrica ako i samo ako postoji jedinstvena donja trokutasta matrica L takva da je $A = LL^*$. Takvu dekompoziciju nazivamo faktorizacija Choleskog a L nazivamo Cholesky faktor.

Teorem 7.2 (Dekompozicija na singularne vrijednosti) *Ako je $A \in \mathbb{C}^{m\times n}$, tada postoje unitarne matrice $U \in \mathbb{C}^{m\times m}$ i $V \in \mathbb{C}^{n\times n}$ takve da je*

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

pri čemu vrijedi $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(m,n)} \geq 0$.

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

$$\|Av\|_2 = \max\{\|Ax\|_2 : \|x\|_2 = 1, x \in \mathbb{C}^n\}.$$

Ako je $\|Av\|_2 = 0$, onda je $A = 0$ i faktorizacija u iskazu teorema je trivijalna uz $\Sigma = 0$ i s proizvoljnim unitarnim matricama U i V reda m odnosno n. Ako je $\|Av\|_2 > 0$, stavimo $\sigma_1 = \|Av\|_2$ i formiramo jedinični vektor

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

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

$$A_1 = U_1^* A V_1 = \begin{pmatrix} u_1^* \\ u_2^* \\ \vdots \\ u_m^* \end{pmatrix} \begin{pmatrix} Av_1 & Av_2 & \ldots & Av_n \end{pmatrix} = \begin{pmatrix} u_1^* \\ u_2^* \\ \vdots \\ u_m^* \end{pmatrix} \begin{pmatrix} \sigma_1 u_1 & Av_2 & \ldots & Av_n \end{pmatrix}$$

$$= \begin{pmatrix} \sigma_1 & u_1^* A v_2 & \ldots & u_1^* A v_n \\ 0 & u_2^* A v_2 & \ldots & u_2^* A v_n \\ \vdots & \vdots & & \vdots \\ 0 & u_m^* A v_2 & \ldots & u_m^* A v_n \end{pmatrix} = \begin{pmatrix} \sigma_1 & w^* \\ 0 & A_2 \end{pmatrix},$$

gdje je $w \in \mathbb{C}^{n-1}$, $A_2 \in \mathbb{C}^{(m-1)\times(n-1)}$. Za jedinični vektor (jedinični vektor koji odgovara prvom retku matrice s desne strane iz gornje jednakosti)

$$y = \frac{1}{\sqrt{\sigma_1^2 + w^* w}} \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \tag{7.2}$$

zbog unitarne invarijantnosti euklidske norme, vrijedi

$$\|A(V_1 y)\|_2^2 = \|(U_1 A_1 V_1^*)V_1 y\|_2^2 = \|A_1 y\|_2^2 = \frac{(\sigma_1^2 + w^* w)^2 + \|A_2 w\|_2^2}{\sigma_1^2 + w^* w}$$

a ovo je strogo veće od σ_1^2 ako je $w \neq 0$. Budući da je to u suprotnosti s maksimalnošću od σ_1, zaključujemo da je $w = 0$. Stoga je

$$A_1 = U_1^* A V_1 = \begin{pmatrix} \sigma_1 & 0 \\ 0 & A_2 \end{pmatrix}. \tag{7.3}$$

Sada ponavljamo iste argumente za matricu $A_2 \in \mathbb{C}^{(m-1)\times(n-1)}$. Na taj način dobivamo unitarne matrice U i V kao produkt unitarnih matrica koje su dobijene nakon svakog koraka. Ako je $m \geq n$, taj postupak vodi do dijagonalne matrice Σ.

S druge strane, ako je $m < n$, primijenimo postupak opisan u dokazu teorema na matricu A^*, za koju je broj redaka n veći od broja stupaca m. Nakon dobivene dekompozicije $A^* = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^*$, kompleksno transponirajmo obje matrice u toj jednakosti. Dobijemo

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

što je tražena dekompozicija na singularne vrijednosti od A, pri čemu moramo još preimenovati $\tilde{V}$ u U, $\tilde{U}$ u V i $\tilde{\Sigma}$ u S. □

Sljedeći nam teorem ilustrira neka od korisnih svojstava SVD dekompozicije.

Teorem 7.3 *Neka je $A \in \mathbb{C}^{m\times n}$ matrica sa dekompozicijom na singularne vrijednosti, $A = U\Sigma V^*$ i singularnim vrijednostima $\sigma_1 \geq \cdots \geq \sigma_r > 0 = \cdots = 0$. Tada vrijedi:*

a) rang matrice A iznosi r, tj. $\operatorname{rank}(A) = r$.

b) $\mathcal{R}(A) = \operatorname{range}(A) = \operatorname{span}(u_1, \ldots, u_r)$

c) $\mathcal{N}(A) = \ker(A) = \operatorname{span}(v_{r+1}, \ldots, v_n)$.

Dokaz.
a) Budući da su U i V^* regularne matrice, slijedi da $\operatorname{rank}(A) = \operatorname{rank}(\Sigma) = r$.

b) Znamo da je $\mathcal{R}(A) = \text{span}(Av_1, \ldots, Av_n)$, jer je $v_1, \ldots, v_n$ baza (*svaki linearni operator jedinstveno je određen svojim djelovanjem na bazi*). Stoga je $\text{span}(Av_1, \ldots, Av_r) \subseteq \mathcal{R}(A)$. Međutim, za $1 \leq i \leq r$ vrijedi

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

Vektori u_i linearno su nezavisni, pa potprostor

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

ima dimenziju r kao i $\mathcal{R}(A)$, što znači da je $\mathcal{R}(A) = \text{span}(u_1, \ldots, u_r)$.
c) Budući da je $Av_i = 0$, za $r+1 \leq i \leq n$, zaključujemo da je

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

Iz rezultata o defektu koji kaže da je $\text{defekt}(A) = n - \text{rang}(A) = n - r$ slijedi da je dimenzija potprostora $\text{span}(v_{r+1}, \ldots, v_n)$ ista kao i dimenzija od $\mathcal{N}(A)$. □

Teorem 7.4 *Ako je A hermitska matrica, tj. $A = A^*$, tada su njene singularne vrijednosti jednake: $|\lambda_1|, \ldots, |\lambda_n|$, gdje su $\lambda_1, \ldots, \lambda_n$ svojstvene vrijednosti matrice A.*

7.2.2 Rješavanje LPNK pomoću dekompozicije na singularne vrijednosti

Kao što smo pokazali i u prethodnom poglavlju, linearni problem najmanjih kvadrata svodi se na problem minimizacije norme $\|Ax - b\|$, gdje su $A \in \mathbb{C}^{m\times n}$ i $b \in \mathbb{C}^m$, pri čemu je $m \geq n$, tj. svodi se na izračunavanje vektora $x \in \mathbb{C}^n$ koji minimizira $\|Ax - b\|$.

U svrhu toga definiramo funkciju

$$F(x) = \|Ax - b\|\,.$$

Neka je $m \geq n$ i neka je $A = U\Sigma V^*$ dekompozicija na singularne vrijednosti matrice A, gdje su $U \in \mathbb{C}^{m\times m}$ i $V \in \mathbb{C}^{n\times n}$ unitarne a

$$\Sigma = \begin{bmatrix} \Sigma_n \\ 0 \end{bmatrix}, \qquad \Sigma_n = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)\,,$$

pri čemu vrijedi $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ (vidi teorem 7.2).

Pretpostavimo nadalje da je $\text{rang}(A) = r \leq n \leq m$. Tada prema Teoremu 7.2 slijedi da dekompoziciju na singularne vrijednosti matrice A možemo zapisati u obliku $A = U\Sigma V^*$, pri čemu

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}, \qquad \Sigma_r = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)\,,$$

i $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

Neka je $\|\cdot\|$ bilo koja unitarno invarijantna norma. Koristeći činjenicu da produkt s unitarnim matricama ne mijenja normu, možemo pisati

$$F(x) = \|Ax - b\| = \|U\Sigma V^* x - b\| = \|U(\Sigma V^* x - U^* b)\| = \|\Sigma V^* x - U^* b\|\,.$$

Sada, uz oznaku $z = V^* x$ i $\hat{b} = U^* b$, imamo

$$\|Ax - b\|^2 = \|\Sigma z - \hat{b}\|^2 = \left\| \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \end{bmatrix} \right\|,$$

pri čemu su $z_1, \hat{b}_1 \in \mathbb{C}^r$, a $z_2 \in \mathbb{C}^{n-r}$ i $\hat{b}_2 \in \mathbb{C}^{m-r}$. Iz gornje jednakosti slijedi

$$\|Ax - b\|^2 = \| \begin{bmatrix} \Sigma_r z_1 - \hat{b}_1 \\ -\hat{b}_2 \end{bmatrix} \|^2 = \|\Sigma_r z_1 - \hat{b}_1\|^2 + \|\hat{b}_2\|^2\,. \tag{7.4}$$

Primijetite da smo u posljednjem izrazu na desnoj strani jednakosti (7.4) koristili istu oznaku $\|\cdot\|$ za vektorske norme definirane na prostorima $\mathbb{C}^r$ odnosno $\mathbb{C}^{m-r}$.

Istaknimo da u slučaju $r < n$ rješenje x problema LPNK nije jednoznačno određeno. U tom slučaju, kao što ćemo pokazati, jednoznačno ćemo moći izračunati samo r komponenata rješenja x. Stoga označimo sa $x = \begin{bmatrix} \mathbf{x}^* & x_0^* \end{bmatrix}^*$, gdje je $\mathbf{x} \in \mathbb{C}^r$ i $x_0 \in \mathbb{C}^{n-r}$, i neka u_i i v_i označavaju $i-ti$ stupac matrica U odnosno V.

Iz (7.4) slijedi da će $F(x)$ biti minimalno ako

$$\Sigma_r z_1 = \hat{b}_1,$$

iz čega slijedi da je

$$z_1 = (\Sigma_r)^{-1} \hat{b}_1\,.$$

Budući da iz $z = V^*x$ slijedi da je $\mathbf{x} = \begin{bmatrix} v_1, & \dots, & v_r \end{bmatrix} z$, što sve zajedno daje rješenje LPNK $x = \begin{bmatrix} \mathbf{x}^* & x_0^* \end{bmatrix}^*$, gdje je:

$$\mathbf{x} = \begin{bmatrix} v_1, & \dots & v_r \end{bmatrix} \begin{bmatrix} 1/\sigma_1 & & & \\ & 1/\sigma_2 & & \\ & & \ddots & \\ & & & 1/\sigma_r \end{bmatrix} \begin{bmatrix} u_1^* b \\ u_2^* b \\ \vdots \\ u_r^* b \end{bmatrix} = \sum_{i=1}^{r} \frac{u_i^* b}{\sigma_i} v_i \,.$$

Navedeni rezultati sažeti su u sljedećem algoritmu pomoću kojeg se izračunava rješenje LPNK u slučaju kad matrica A ima puni rang stupaca.

Algoritam za LPNK pomoću SVD faktorizacije.
ulaz: $A \in \mathbb{C}^{m\times n}, \quad m \geq n \quad \text{i} \quad rang(A) = n, \quad b \in \mathbb{C}^m$
izlaz: $x \in \mathbb{C}^n$ koji minimizira $\|Ax - b\|_2$
1: izračunaj reduciranu SVD-faktorizaciju $A = \hat{U}\hat{\Sigma}\hat{V}^*$
2: izračunaj $\hat{U}^*b \in \mathbb{C}^n$
3: riješi $\hat{\Sigma}y = \hat{U}^*b$
4: vrati $x = Vy$

Tada rezultat algoritma zadovoljava

$$A^*Ax = A^*\hat{U}\hat{\Sigma}V^*Vy = A^*\hat{U}\hat{U}^*b = A^*b$$

pa stoga i rješava problem najmanjih kvadrata.

7.2.3 Rješavanje LPNK pomoću QR dekompozicije

Slično kao u prethodnom poglavlju i u ovom ćemo tražiti minimum funkcije

$$F(x) = \|Ax - b\| \,,$$

gdje su $A \in \mathbb{C}^{m\times n}$ i $b \in \mathbb{C}^m$, pri čemu je $m \geq n$.

Nadalje, neka je $\text{rang}(A) = r \leq n \leq m$ rang matrice A te neka je $A = QR$, QR dekompozicija matrice A, gdje je $Q \in \mathbb{C}^{m\times m}$ unitarna matrica, a R gornja trokutasta matrica oblika

$$R = \begin{bmatrix} R_r & R_{12} \\ 0_{m-r,r} & 0_{m-r,n-r} \end{bmatrix}, \qquad R_r = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1r} \\ 0 & r_{22} & \cdots & r_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{rr} \end{bmatrix},$$

pri čemu oznaka $0_{i,j}$ označava nul-matricu dimenzije $i \times j$, a R_{12} je dimenzije $r \times n - r$.

Budući da matrica A ne mora imati puni rang stupaca ($r < n$), rješenje x problema LPNK nije jednoznačno određeno. Stoga, slično kao i u prošlom poglavlju, označimo sa $x = \begin{bmatrix}\mathbf{x}^* & x_0^*\end{bmatrix}^*$, gdje je $\mathbf{x} \in \mathbb{C}^r$ i $x_0 \in \mathbb{C}^{n-r}$, i neka q_i označava $i - ti$ stupac matrice Q.

Neka je $\|\cdot\|$ bilo koja unitarno invarijantna norma, koristeći činjenicu da produkt s unitarnim matricama ne mijenja normu, možemo pisati

$$F(x) = \|Ax - b\| = \|QRx - b\| = \|Q(Rx - Q^*b)\| = \|Rx - Q^*b\|\,.$$

Iz gornje jednakosti slijedi

$$\|Ax - b\|^2 = \| \begin{bmatrix} R_r\mathbf{x} - R_{12}x_0 - \hat{b}_1 \\ -\hat{b}_2 \end{bmatrix} \|^2 = \|R_r\mathbf{x} - R_{12}x_0 - \hat{b}_1\|^2 + \|\hat{b}_2\|^2\,, \tag{7.5}$$

gdje je $\hat{b} \equiv Q^*b = \begin{bmatrix}\hat{b}_1^* & \hat{b}_2^*\end{bmatrix}^*$, a $\hat{b}_1$ je r dimenzionalni vektor. Zbog razlike u rangu matrice A i broja stupaca, dio rješenja x_0 je proizvoljan, te ga možemo izabrati po volji. Jedan od standardnih načina izbora jest $x_0 = 0$, što osigurava da riješenje x, LPNK problema, ima najmanju normu. Primijetite da smo i ovog puta zlorabili oznake, pa smo u izrazu na desnoj strani jednakosti (7.5) koristili istu oznaku $\|\cdot\|$ za vektorske norme definirane na prostorima $\mathbb{C}^r$ odnosno $\mathbb{C}^{m-r}$.

Iz (7.5), uz pretpostavku $x_0 = 0$, slijedi da će $F(x)$ biti minimalno ako

$$R_r\mathbf{x} = \hat{b}_1$$

iz čega slijedi da je rješenje $\mathbf{x}$ LPNK dano sa

$$\mathbf{x} = (R_r)^{-1}\hat{b}_1\,, \quad \text{ili} \quad \mathbf{x} = (R_r)^{-1}\begin{bmatrix} q_1^*b \\ q_2^*b \\ \vdots \\ q_r^*b \end{bmatrix}\,.$$

Navedeni rezultati sažeti su u sljedećem algoritmu pomoću kojeg se izračunava rješenje LPNK u slučaju matrice punog ranga stupaca.

Algoritam za LPNK pomoću QR dekompozicije.
ulaz: $A \in \mathbb{C}^{m\times n}$, $m \geq n$ i $rang(A) = n$, $b \in \mathbb{C}^m$

izlaz: $x \in \mathbb{C}^n$ koji minimizira $\|Ax - b\|_2$
1: izračunaj reduciranu QR dekompoziciju $A = \hat{Q}\hat{R}$
2: izračunaj $\hat{Q}^*b \in \mathbb{C}^n$
3: riješi $\hat{R}x = \hat{Q}^*b$ koristeći supstitucije unatrag

Tada rezultat algoritma zadovoljava

$$A^*Ax = A^*\hat{Q}\hat{R}x = A^*\hat{Q}\hat{Q}^*b = A^*b,$$

pa stoga i rješava problem najmanjih kvadrata. Korak 1 i 2 zajedno daju $C_1(m,n) \sim 2mn^2 - \frac{2}{3}n^3$, a korak 3 daje $C_2(m,n) = \mathcal{O}(n^2)$. Stoga imamo sljedeću asimptotsku ocjenu

$$C(m,n) \sim 2mn^2 - \frac{2}{3}n^3$$

za $m, n \to \infty$, gdje je $m = \Theta(n)$.

7.3 Uvjetovanost problema najmanjih kvadrata

Kao i prije, neka je $A \in \mathbb{C}^{m\times n}$, gdje je $m \geq n$, $rang(A) = n$ i $b \in \mathbb{C}^m$. Iz korolara 7.1 znamo da rješenje problema najmanjih kvadrata možemo izračunati ovako

$$x = (A^*A)^{-1}A^*b.$$

Definicija 7.3 *Matrica $A^\dagger = (A^*A)^{-1}A^*$ zove se pseudo-inverz matrice $A \in \mathbb{C}^{m\times n}$.*

Definicija 7.4 *Za $A \in \mathbb{C}^{m\times n}$ definiramo uvjetovanost matrice A na sljedeći način*

$$\kappa(A) = \|A\|\|A^\dagger\|.$$

Napomena 7.2 *Ako je $rang(A) < n$, tada A^*A nije invertibilna, pa stavljamo $\kappa(A) = +\infty$. Za kvadratne invertibilne matrice je $A^\dagger = A^{-1}$ pa je ova definicija konzistentna s prethodnom definicijom uvjetovanosti. Kao i prije uvjetovanost ovisi o odabranoj matričnoj normi.*

Lema 7.1 *Neka su $\sigma_1 \geq \ldots \geq \sigma_n > 0$ netrivijalne singularne vrijednosti matrice A. Tada je uvjetovanost u normi $\|\cdot\|_2$ matrice A dana sa*

$$\kappa(A) = \frac{\sigma_1}{\sigma_n}.$$

Dokaz. Koristeći SVD matrice A slijedi

$$A^\dagger = (A^*A)^{-1}A^* = (V\Sigma^*\Sigma V^*)^{-1}V\Sigma^*U^* = V(\Sigma^*\Sigma)^{-1}\Sigma^*U^*. \tag{7.6}$$

Matrica $(\Sigma^*\Sigma)^{-1}\Sigma^* \in \mathbb{C}^{n\times m}$ je dijagonalna s dijagonalnim elementima

$$\frac{1}{\sigma_i^2}\sigma_i = \frac{1}{\sigma_i}$$

za $i = 1, \ldots, n$. Tada je jednadžba (7.6) (do na raspored singularnih vrijednosti) dekompozicija na singularne vrijednosti matrice $A^\dagger$ i imamo

$$\kappa(A) = \|A\|_2\|A^\dagger\|_2 = \sigma_1 \cdot \frac{1}{\sigma_n}.$$

Teorem 7.5 *Pretpostavimo da x rješava problem najmanjih kvadrata za b i $x + \Delta x$ rješava problem najmanjih kvadrata za $b + \Delta b$. Definirajmo $\vartheta \in [0, \pi/2]$ sa $\cos(\vartheta) = \frac{\|Ax\|_2}{\|b\|_2}$ i $\eta = \|A\|\|x\|/\|Ax\| \geq 1$. Tada vrijedi*

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \frac{\kappa(A)}{\eta\cos(\vartheta)} \cdot \frac{\|\Delta b\|_2}{\|b\|_2},$$

gdje je $\kappa(A)$ uvjetovanost matrice A obzirom na normu $\|\cdot\|_2$.
Dokaz. Znamo da je $x = A^\dagger b$ i $x + \Delta x = A^\dagger(b + \Delta b)$. Zbog linearnosti je $\Delta x = A^\dagger \Delta b$ iz čega slijedi

$$\begin{aligned}\frac{\|\Delta x\|_2}{\|x\|_2} &\leq \frac{\|A^\dagger\|_2\|\Delta b\|_2}{\|x\|_2} = \frac{\kappa(A)\|\Delta b\|_2}{\|A\|_2\|x\|_2} \\ &= \frac{\kappa(A)\|\Delta b\|_2}{\eta\|Ax\|_2} = \frac{\kappa(A)}{\eta\cos(\vartheta)} \cdot \frac{\|\Delta b\|_2}{\|b\|_2}.\end{aligned}$$

Napomena 7.3 *Konstanta $\kappa(A)/\eta\cos(\vartheta)$ postaje velika ako je $\kappa(A)$ veliko ili $\vartheta \approx \pi/2$. U bilo kojem od ta dva slučaja problem je loše uvjetovan.*

Teorem 7.6 *Neka su ϑ i η kao što smo ih prije definirali. Pretpostavimo da x rješava problem najmanjih kvadrata za A, b i $x + \Delta x$ rješava problem najmanjih kvadrata za $A + \Delta A$, b. Tada*

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \left(\kappa(A) + \frac{\kappa(A)^2\tan(\vartheta)}{\eta}\right) \cdot \frac{\|\Delta A\|_2}{\|A\|_2},$$

gdje je $\kappa(A)$ uvjetovanost matrice A u normi $\|A\|_2$.

Teorem 7.7 *Algoritam koji rješava problem najmanjih kvadrata pomoću QR dekompozicija s Householderom je povratno stabilan: izračunato rješenje $\hat{x}$ minimizira*
$\|(A + \Delta A)\hat{x} - b\|_2$ *za neku matricu ΔA gdje je*

$$\frac{\|\Delta A\|_2}{\|A\|_2} = \mathcal{O}(\varepsilon_m).$$

Teoremi 7.6 i 7.7 zajedno daju ocjenu točnosti izračunatog rezultata x; daju da je ϑ "daleko" od $\pi/2$.

7.3.1 Zadaci

1. Prateći dokaz egzistencije SVD faktorizacije odredite SVD dekompoziciju matrice
$$\begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}.$$

2. Neka je dano m točaka $(u^{(i)}, v^{(i)})$, gdje je $u^{(i)} \in \mathbb{R}^{n-1}$ i $v^{(i)} \in \mathbb{R}$ za $i = 1, \ldots, m$. Želimo odrediti $\alpha \in \mathbb{R}^{n-1}$ i $\beta \in \mathbb{R}$ koji minimiziraju
$$\sum_{i=1}^{m} |\alpha^T u^{(i)} + \beta - v^{(i)}|^2.$$
Pokažite da se taj problem može preformulirati kao standardni problem najmanjih kvadrata za specijalan odabir matrice $A \in \mathbb{R}^{m\times n}$ i $b \in \mathbb{R}^m$.

3. Odredite x koji minimizira normu $\|Ax - b\|_2$, koristeći algoritam za LPNK pomoću QR faktorizacije, ako je
$$A = \begin{pmatrix} -3 & 3 \\ 2 & 0 \\ 6 & -9 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$
[Rješenje: $x = \begin{pmatrix} -\frac{12}{49} \\ -\frac{5}{21} \end{pmatrix}$.]

4. Neka je dana dekompozicija na singularne vrijednosti matrice A

$$A = \frac{1}{3}\begin{pmatrix} -2 & -2 & 1 \\ 1 & -2 & -2 \\ 2 & -1 & 2 \end{pmatrix}\begin{pmatrix} 8 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}^T.$$

Ako je $b = \begin{pmatrix} -1 & 2 & 4 \end{pmatrix}^T$, odredite x koji minimizira $\|Ax - b\|_2$ i odredite uvjetovanost matrice A u normi $\|\cdot\|_2$.
[Rješenje: $x = \begin{pmatrix} \frac{3}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} \end{pmatrix}$, $\kappa(A) = 4$.]

Poglavlje 8

Svojstvene vrijednosti i svojstveni vektori

U ovom ćemo se poglavlju baviti izračunavanjem svojstvenih vrijednosti i pripadnih svojstvenih vektora matrica općeg oblika. Također ćemo prezentirati neke algoritme za njihovo izračunavanje za matrice s određenom strukturom, npr. simetrične ili pozitivno definitne matrice.

8.1 Problem svojstvenih vrijednosti

Prije nego predstavimo neke od algoritama pomoću kojih ćemo izračunavati svojstvene vrijednosti i pripadne svojstvene vektore ponovimo neke teorijske činjenice u vezi s njima.

Vektor $x \in \mathbb{C}^n$ je svojstveni vektor matrice $A \in \mathbb{C}^{n\times n}$ s pripadnom svojstvenom vrijednosti $\lambda \in \mathbb{C}$ ako je

$$Ax = \lambda x, \quad x \neq 0.$$

U ovom ćemo poglavlju naučiti kako za danu matricu A izračunati svojstvene vrijednosti i svojstvene vektore.

Definicija 8.1 *Za matricu $A \in \mathbb{C}^{n\times n}$ definiramo karakteristični polinom od A kao*

$$\rho_A(z) := \det(A - zI).$$

Teorem 8.1 *$\lambda \in \mathbb{C}$ je svojstvena vrijednost matrice A ako i samo ako je $\rho_A(z) = 0$.*

Dokaz. λ je svojstvena vrijednost matrice A ako i samo ako postoji $x \neq 0$ takav da je $(A-\lambda I)x = 0$. To je ekvivalentno uvjetu da je $A-\lambda I$ singularna, što je opet ekvivalentno činjenici da je $\det(A - zI) = 0$. □

Odatle slijedi da ako možemo odrediti nultočke proizvoljnog polinoma, tada možemo odrediti svojstvene vrijednosti proizvoljne matrice. Sada ćemo pokazati da su ta dva problema čak ekvivalentna, odnosno da za svaki polinom možemo naći matricu kojoj je taj polinom karakteristični polinom. Neka je dan polinom $p(z) = a_0 + a_1 z + \cdots + a_{n-1}z^{n-1} + z^n$. Definirajmo $A \in \mathbb{C}^{n\times n}$ sa

$$A = \begin{pmatrix} 0 & & & -a_0 \\ 1 & & & -a_1 \\ & \ddots & & \vdots \\ & & \ddots & -a_{n-2} \\ & & 1 & -a_{n-1} \end{pmatrix}. \tag{8.1}$$

Indukcijom se pokaže da je $\rho_A(z) = \det(A - zI) = (-1)^n p(z)$. Time smo pokazali da je problem određivanja svojstvenih vrijednosti ekvivalentan problemu određivanja nultočaka polinoma.

Teorem 8.2 (Abel, 1824) *Za svaki $n \geq 5$ postoji polinom p stupnja n s racionalnim koeficijentima koji ima realnu nultočku koja se ne može izraziti koristeći samo racionalne brojeve, zbrajanje, oduzimanje, množenje, dijeljenje i vađenje n-tog korijena.*

Budući da će se rješenje izračunato pomoću računala uvijek temeljiti na operacijama spomenutim u teoremu, možemo zaključiti da nije moguće naći algoritam koji će izračunati svojstvene vrijednosti u konačno mnogo koraka. Zaključak: svaki algoritam za određivanje svojstvenih vrijednosti mora biti iterativan.

Osim toga, treba istaknuti da veza između određivanja svojstvenih vrijednosti i pripadnih svojstvenih vektora i nultočaka polinoma je prije svega teorijskog karaktera. Naime, sljedeći primjer ilustrira kako u jednostavnom slučaju utjecaj grešaka zaokruživanja može biti „poguban" za točnost izračunatih svojstvenih vrijednosti, kad se one računaju kao nultočke odgovarajućeg karakterističnog polinoma.

Primjer 8.1 *Neka je A dijagonalna matrica 20×20, s elementima $1, 2, \ldots, 20$ na dijagonali. Očito je da su tada dijagonalni elementi svojstvene vrijednosti matrice A. Međutim, želimo li na matricu A izravno primijeniti teo-*

rem 8.1 dobivamo sljedeće. Karakteristični polinom matrice A definiran sa $P_{20}(z) \equiv \det(A - zI)$ ima sljedeći oblik:

$$\begin{aligned}P_{20}(z) = {} & z^{20} - 210\,z^{19} + 20615\,z^{18} - 1256850\,z^{17} + 53327946\,z^{16} - 1672280820\,z^{15} \\ & + 40171771630\,z^{14} - 756111184500\,z^{13} + 11310276995381\,z^{12} \\ & - 135585182899530\,z^{11} + 1307535010540395\,z^{1}0 - 10142299865511450\,z^{9} \\ & + 63030812099294896\,z^{8} - 311333643161390640\,z^{7} + 1206647803780373360\,z^{6} + \\ & 3599979517947607200\,z^{5} + 8037811822645051776\,z^{4} + 12870931245150988800\,z^{3} \\ & + 13803759753640704000\,z^{2} + 8752948036761600000\,z + 2432902008176640000\end{aligned}$$

Primijetimo, koeficijent uz $-z^{19}$ je 210, što odgovara zbroju svih svojstvenih vrijednosti. S druge strane, koeficijent uz z^0, odnosno slobodni koeficijent iznosi $20!$*, što je produkt svih svojstvenih vrijednosti. Ostali su koeficijenti različite kombinacije zbroja i produkata svojstvenih vrijednosti matrice A.*

Sada je očito da bilo koji račun s gore navedenim koeficijentima, s točnošću od 16 znamenaka dovodi do značajnih grešaka zaokruživanja. Tako ćemo npr. koristeći MatLab, izračunati sljedeće nultočke gornjeg polinoma s točnošću od

16 znamenaka:

1.00000000000000
2.00000000000096
2.99999999986640
4.00000000495944
4.99999991473414
6.00000084571661
6.99999455544845
8.00002443256894
8.99992001186835
10.00019696490537
10.99962843024064
12.00054374363591
12.99938073455790
14.00054798867380
14.99962658217055
16.00019208303847
16.99992773461773
18.00001875170604
18.99999699774389
20.00000022354640

Gornji primjer ilustrira da već pri spremanju koeficijenata karakterističnog polinoma u dvostrukoj preciznosti aritmetika pomičnog zareza može znatno pokvariti točnost izračunatih svojstvenih vrijednosti.

Stoga ćemo sljedeće poglavlje posvetiti iterativnim metodama za izračunavanje svojstvenih vrijednosti simetričnih (hermitskih) matrica.

8.1.1 Iterativne metode za simetrične matrice

Prisjetimo se, realna matrica A je simetrična ako je $A = A^T$. Kompleksna matrica A je hermitska ako je $A = A^*$

Teorem 8.3 *Ako je $A \in \mathbb{C}^{n\times n}$ hermitska, tada postoji unitarna matrica $Q \in \mathbb{C}^{n\times n}$ i dijagonalna $\Lambda \in \mathbb{R}^{n\times n}$ tako da je*

$$A = Q\Lambda Q^*.$$

Napomena 8.1

1) Ortonormalni stupci matrice Q svojstveni su vektori matrice A, a dijagonalni elementi od Λ odgovarajuće su svojstvene vrijednosti.

2) Iz teorema slijedi da hermitske matrice imaju realne svojstvene vrijednosti.

Iterativne metode koje obrađujemo u ovom poglavlju računaju aproksimacije svojstvenih vektora, a pomoću njih lako određujemo pripadne svojstvene vrijednosti. Zaista, ako poznamo svojstveni vektor, onda će nam sljedeće razmatranje pomoći u dobivanju aproksimacije odgovarajuće svojstvene vrijednosti. Dana je matrica $A \in \mathbb{C}^{n\times n}$. Želimo naći $\alpha \in \mathbb{C}$ koji minimizira $\|Ax - \alpha x\|_2$. Ako je x svojstveni vekto, tada se minimum dostiže u odgovarajućoj svojstvenoj vrijednosti. Inače, promotrimo normalne jednadžbe:

$$\|Ax - \alpha x\|_2 = \|x \cdot \alpha - Ax\|_2$$

na x gledamo kao na $n\times 1$ matricu, $\alpha \in \mathbb{C}^1$ je "vektor" nepoznanica i $Ax \in \mathbb{C}^n$ neka je desna strana. Tada se prema korolaru 7.1 minimum postiže za

$$\alpha = (x^*x)^{-1}x^*(Ax) = \frac{\langle x, Ax\rangle}{\langle x, x\rangle}.$$

Definicija 8.2 *Reyleigh-jev kvocijent matrice $A \in \mathbb{C}^{n\times n}$ definira se sa*

$$r_A(x) = \frac{\langle x, Ax\rangle}{\langle x, x\rangle}$$

za sve $x \in \mathbb{C}^n$.

Teorem 8.4 *Neka je $A \in \mathbb{R}^{n\times n}$ simetrična i $x \in \mathbb{R}^n$, $x \neq 0$. Tada je x svojstveni vektor od A s pripadnom svojstvenom vrijednosti λ ako i samo ako je $\nabla r_A(x) = 0$ i $r_A(x) = \lambda$.*

Dokaz. Gradijent od r_A možemo izračunati na sljedeći način

$$\begin{aligned}
\nabla r_A(x) &= \left(\frac{\partial}{\partial x_i} \frac{\sum_{j,k=1}^n x_j a_{jk} x_k}{\sum_{j=1}^n x_j^2} \right)_{i=1,\dots,n} \\
&= \left(\frac{\left(\sum_{k \neq i} a_{ik} x_k + \sum_{j \neq i} x_j a_{ji} + 2a_{ii} x_i \right) \sum_{j=1}^n x_j^2 - \sum_{j,k} x_j a_{jk} x_k \cdot 2x_i}{\left(\sum_{j=1}^n x_j^2 \right)^2} \right)_{i=1,\dots,n} \\
&= \frac{2Ax \cdot \langle x, x \rangle - 2\langle x, Ax \rangle \cdot x}{\langle x, x \rangle^2} \\
&= \frac{2}{\|x\|_2} \left(Ax - r_A(x) \cdot x \right).
\end{aligned}$$

Pretpostavimo da je $Ax = \lambda x$. Tada je $r_A(x) = \langle x, \lambda x \rangle / \langle x, x \rangle = \lambda$ i

$$\nabla r_A(x) = \frac{2}{\|x\|_2^2} (\lambda x - \lambda x) = 0.$$

S druge strane, ako je $\nabla r_A(x) = 0$, onda je $Ax - r_A(x)x = 0$, pa stoga imamo $Ax = r_A(x) \cdot x$, odnosno $\lambda = r_A(x)$. □

Za ostatak poglavlja neka su $x_1, \dots, x_n$ ortonormalni sustav svojstvenih vektora i $\lambda_1, \dots, \lambda_n$ odgovarajuće svojstvene vrijednosti realne simetrične matrice A. Poredajmo svojstvene vrijednosti tako da je $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$.

Algoritam (metoda potencija).
ulaz: $A \in \mathbb{C}^{n \times n}$ simetrična sa $|\lambda_1| > |\lambda_2|$
izlaz: $z^{(k)} \in \mathbb{R}^n, \quad \lambda^{(k)} \in \mathbb{R}$ gdje je $z^{(k)} \approx x_1$ i $\lambda^{(k)} \approx \lambda_1$
1: odaberite $z^{(0)} \in \mathbb{R}^n$ takav da je $\|z^{(0)}\|_2 = 1$
2: **for** $k = 1, 2, 3, \dots$ **do**
3: $\quad w^{(k)} = Az^{(k-1)}$
4: $\quad z^{(k)} = \dfrac{w^{(k)}}{\|w^{(k)}\|_2}$
5: $\quad \lambda^{(k)} = \langle z^{(k)}, Az^{(k)} \rangle$
6: **end for**

Napomena 8.2

1) *U konkretnoj primjeni metoda se, kao i svaka iterativna metoda, zaustavlja u određenom trenutku kad je rezultat "dovoljno blizu" egzaktnom rješenju.*

2) *Algoritam računa* $z^{(k)} = A^k z^{(0)}/\|A^k z^{(0)}\|_2$ *i* $\lambda^{(k)} = r_A(z^{(k)})$. *Kako bismo izbjegli overflow/underflow pogreške, vektor* $z^{(k)}$ *se u svakom koraku iteracija normalizira.*

3) *Metoda se temelji na sljedećoj ideji: ako izrazimo* $z^{(0)}$ *u bazi* $x_1, \ldots, x_n$ *imamo*

$$z^{(0)} = \sum_{i=1}^{n} \alpha_i x_i$$

i

$$A^k z^{(0)} = \sum_{i=1}^{n} \alpha_i A^k x_i = \sum_{i=1}^{n} \alpha_i \lambda_i^k x_i.$$

Za veliko k *u tom izrazu dominira član koji odgovara svojstvenoj vrijednosti s najvećim modulom.*

Teorem 8.5 *Neka je* $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$ *i* $\langle z^{(0)}, x_1 \rangle \neq 0$. *Tada postoji niz* $(\sigma^{(k)})_{k\in\mathbb{N}}$ *gdje je* $\sigma^{(k)} \in \{-1, +1\}$ *za sve* $k \in \mathbb{N}$ *tako da nizovi* $(z^{(k)})$ *i* $(\lambda^{(k)})$ *dobiveni algoritmom metode potencija zadovoljavaju*

$$\|z^{(k)} - \sigma^{(k)} x_1\|_2 = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right) \tag{8.2}$$

i

$$|\lambda^{(k)} - \lambda_1| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right). \tag{8.3}$$

Dokaz. a) Neka je $x_1, \ldots, x_n$ ortonormalan sustav svojstvenih vektora sa svojstvenim vrijednostima $\lambda_1, \ldots, \lambda_n$ i

$$z^{(0)} = \sum_{i=1}^{n} \alpha_i x_i.$$

Budući da smo pretpostavili da je $\alpha_1 = \langle z^{(0)}, x_1 \rangle \neq 0$, imamo

$$A^k z^{(0)} = \sum_{i=1}^{n} \alpha_i \lambda^k x_i = \alpha_1 \lambda_1^k \left(x_1 + \sum_{i=2}^{n} \frac{\alpha_i}{\alpha_1} \left|\frac{\lambda_i}{\lambda_1}\right|^k x_i \right),$$

pa iz Pitagorinog teorema slijedi

$$\|A^k z^{(0)}\|_2^2 = |\alpha_1 \lambda_1^k|^2 \left(1 + \sum_{i=2}^{n} \left(\frac{\alpha_i}{\alpha_1}\right)^2 \left|\frac{\lambda_i}{\lambda_1}\right|^{2k}\right) \leq |\alpha_1 \lambda_1^k|^2 \left(1 + \left|\frac{\lambda_2}{\lambda_1}\right|^{2k} \sum_{i=2}^{n} \left(\frac{\alpha_i}{\alpha_1}\right)^2\right).$$

Ako iskoristimo Taylorovu aproksimaciju $\sqrt{1+x^2} \approx 1 + x^2/2$, možemo zaključiti

$$\|A^k z^{(0)}\|_2 = |\alpha_1 \lambda_1^k| \left(1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)\right).$$

Definirajmo $\sigma^{(k)} = sign(\alpha_1 \lambda_1^k)$. Tada je

$$\left\| \frac{A^k z^{(0)}}{|\alpha_1 \lambda_1^k|} - \sigma^{(k)} x_1 \right\|_2^2 = \left\| \frac{A^k z^{(0)}}{\alpha_1 \lambda_1^k} - x_1 \right\|_2^2 = \sum_{i=2}^{n} \left(\frac{\alpha_i}{\alpha_1}\right)^2 \left|\frac{\lambda_i}{\lambda_1}\right|^{2k} = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$$

i stoga je

$$\begin{aligned}
\|z^{(k)} - \sigma^{(k)} x_1\|_2 &\leq \left\| \frac{A^k z^{(0)}}{\|A^k z^{(0)}\|_2} - \frac{A^k z^{(0)}}{|\alpha_1 \lambda_1^k|} \right\|_2 + \left\| \frac{A^k z^{(0)}}{|\alpha_1 \lambda_1^k|} - \sigma^{(k)} x_1 \right\|_2 \\
&\leq \left\| \frac{A^k z^{(0)}}{|\alpha_1 \lambda_1^k|} \right\| \cdot \left\| \frac{1}{1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)} - 1 \right\|_2 + \left\| \frac{A^k z^{(0)}}{|\alpha_1 \lambda_1^k|} - \sigma^{(k)} x_1 \right\|_2 \\
&= \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right) + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{k}\right) = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{k}\right) \qquad (8.4)
\end{aligned}$$

što dokazuje tvrdnju (8.2).

b) Iz teorema 8.4 znamo da je $r_A(\sigma^{(k)} x_1) = \lambda_1$ i $\nabla r_A(\sigma^{(k)} x_1) = 0$. Taylorov razvoj r_A oko $\sigma^{(k)} x_1$ daje nam

$$\begin{aligned}
r_A(z) &= r_A(\sigma^{(k)} x_1) + \langle \nabla r_A(\sigma^{(k)} x_1), z - \sigma^{(k)} x_1 \rangle + \mathcal{O}(\|z - \sigma^{(k)} x_1\|_2^2) \\
&= \lambda_1 + 0 + \mathcal{O}(\|z - \sigma^{(k)} x_1\|_2^2).
\end{aligned}$$

Koristeći (8.2) slijedi

$$|\lambda^{(k)} - \lambda_1| = |r_A(z^{(k)}) - \lambda_1| = \mathcal{O}(\|z^{(k)} - \sigma^{(k)} x_1\|_2^2) = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$$

čime smo dokazali relaciju (8.3). □

Algoritam metode potencija omogućava nam da odredimo svojstvenu vrijednost s najvećim modulom i odgovarajući svojstveni vektor. Metoda se može modificirati za pronalaženje različitih svojstvenih vrijednosti matrice A. To je napravljeno u sljedećem algoritmu.

Algoritam (inverzna metoda potencija s pomakom ("shiftom")).

ulaz: $A \in \mathbb{R}^{n\times n}$ simetrična sa $\lambda \in \mathbb{R}$
izlaz: $z^{(k)} \in \mathbb{R}^n$, $\lambda^{(k)} \in \mathbb{R}$ gdje je $z^{(k)} \approx x_j$ i $\lambda^{(k)} \approx \lambda_j$,
λ_j je svojstvena vrijednost najbliža λ
1: odaberite $z^{(0)} \in \mathbb{R}^n$ takav da je $\|z^{(0)}\|_2 = 1$
2: **for** $k = 1, 2, 3, \ldots$ **radi**
3: $v = z^{(k-1)}/\|z^{(k-1)}\|_2$
4: riješi $(A - \lambda I)z^{(k)} = v$
5: $\mu^{(k)} = \langle v^*, z^{(k)}\rangle$
6: **if** $\|z^{(k)} - \mu^{(k)}v\|_2 \leq$ tolerancija, **stop**
7: **end for**
8: vrati $\lambda^{(k)} = \lambda + \frac{1}{\mu^{(k)}}$.

Napomena 8.3

1) *Kod praktične primjene metoda se zaustavlja u koraku kad je rezultat "dovoljno blizu" egzaktnom rješenju.*

2) *Kod usporedbe s jednostavnim iteracijama vidimo da $\mu^{(k)}$ aproksimira svojstvenu vrijednost matrice $(A-\lambda I)^{-1}$ s najvećim modulom. Budući da $(A-\lambda I)^{-1}$ ima svojstvene vrijednosti $\mu_i = 1/(\lambda_i - \lambda)$, ta vrijednost je μ_j gdje je λ_j svojstvena vrijednost najbliža λ i imamo*

$$\lambda^{(k)} = \lambda + \frac{1}{\mu^{(k)}} \approx \lambda + \frac{1}{\mu_j} = \lambda + (\lambda_j - \lambda) = \lambda_j.$$

Ako iskoristimo taj argument, jednostavno bismo mogli modificirati teorem 8.5 u teorem koji govori o brzini konvergencije za algoritam inverzne metode potencija s "shiftom".

Metoda potencija radi samo u slučaju kad početni vektor $z^{(0)}$ zadovoljava uvjet $\langle z^{(0)}, x_1\rangle \neq 0$. To nije problem jer $\dim\{z \in \mathbb{R}^n | \langle z, x\rangle = 0\} = n-1 < n$.

Vjerojatnost da ćemo pogoditi upravo tu hiperravninu je nula uz dodatni uvjet da bar za jedan od vektora baze $e_1, \ldots, e_n$ vrijedi $\langle e_i, x_1 \rangle \neq 0$.

Sljedeći algoritam proširuje ideju algoritma metode potencija: pokreće metodu potencija za n ortonormalnih vektora istodobno i ponovno ih ortonormalizira u svakom koraku. Rezultat je algoritam koji aproksimira sve svojstvene vektore i sve svojstvene vrijednosti odjednom.

Algoritam ("simultana" metoda potencija).
ulaz: $A \in \mathbb{R}^{n \times n}$ simetrična
izlaz: $Q^{(k)}, \Lambda^{(k)} \in \mathbb{R}^{n \times n}$
gdje je $Q^{(k)} \approx (x_1, x_2, \ldots, x_n)$ i $\Lambda_{ii}^{(k)} \approx \lambda_i$ za $i = 1, \ldots, n$
1: odaberite ortogonalnu $Q^{(0)} \in \mathbb{R}^{n \times n}$
2: **for** $k = 1, 2, 3, \ldots$ **radi**
3: $W^{(k)} = AQ^{(k-1)}$
4: izračunajte QR dekompoziciju $W^{(k)} = Q^{(k)} R^{(k)}$
5: $\Lambda^{(k)} = (Q^{(k)})^T W^{(k)} Q^{(k)}$
6: **end for**

Teorem 8.6 *Neka je $A \in \mathbb{R}^{n \times n}$ simetrična matrica sa svojstvenim vrijednostima $\lambda_1, \ldots, \lambda_n$ takvim da je $\lambda_1 > \ldots > \lambda_n$. Pretpostavimo da je $\langle Q_i^{(0)}, x_i \rangle \neq 0$ za $i = 1, \ldots, n$. Tada postoje nizovi $(\sigma_i^{(k)})_{k \in \mathbb{N}}$ za $i = 1, \ldots, n$ gdje je $\sigma_i^{(k)} \in \{+1, -1\}$ za sve i, k gdje je*

$$\|q_i^{(k)} - \sigma_i^{(k)} x_i\|_2 = \mathcal{O}\left(|\xi|^k\right)$$

i

$$|\Lambda_{ii}^{(k)} - \lambda_i| = \mathcal{O}\left(|\xi|^{2k}\right)$$

za sve $i = 1, \ldots, n$, a $q_1^{(k)}, \ldots, q_n^{(k)}$ su stupci matrice $Q^{(k)}$ i $\xi = \max\limits_{i=1,\ldots,n-1} |\lambda_i|/|\lambda_{i+1}|$.

Napomena 8.4 *Budući da su matrice $R^{(k)}$ gornjetrokutaste, prvi stupac matrice $Q^{(k)}$ u koraku 4 algoritma je multipl prvog stupca matrice $W^{(k)}$. Stoga prvi stupac $q_1^{(k)}$ matrice $Q^{(k)}$ prikazuje početnu metodu potencija s početnim vektorom $q_1^{(0)}$.*

8.1.2 Zadaci

1. Dokažite indukcijom da matrica A iz (8.1) ima determinantu $(-1)^n p(z)$ gdje je $p(z) = a_0 + a_1 z + \cdots + a_{n-1} z^{n-1} + z^n$.

2. Dokažite teorem 8.6.

3. Neka je dana matrica $A = \begin{pmatrix} 1 & 10 & -2 \\ 10 & 100 & 6 \\ -2 & -6 & 10 \end{pmatrix}$

   a) Odredite svojstvenu vrijednost najveću po modulu.

   b) Odredite svojstvenu vrijednost najbližu broju 10.

   Za toleranciju uzmite 0.05.
   [Rješenje: a) $z_4 = \begin{pmatrix} 0.09777 \\ 0.993224 \\ 0.062826 \end{pmatrix}$, b) $z_3 = \begin{pmatrix} 0.87545 \\ 0.13061 \\ -3.41204 \end{pmatrix}$.]

Literatura

[1] J. W. Demmel, *Applied Numerical Linear Algebra*. SIAM, 1997.

[2] Z. Drmač, V. Hari, M. Marušić, M. Rogina, I. Slapničar, S. Singer, S. Singer, *Numerička analiza, Predavanja i vježbe*. Zagreb, 2003. `http://web.math.hr/~rogina/2001096/num_anal.pdf`

[3] G.H. Golub and Ch.F. van Loan, *Matrix Computations*. J. Hopkins University Press, Baltimore, 1989.

[4] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 1990.

[5] R. A. Horn and C. R. Johnson, *Topics in matrix analysis*. Cambridge University, 1991.

[6] R. Scitovski, *Numerička matematika, 2.* izdanje, Odjel za matematiku Sveučilišta u Osijeku, Osijek, 2004.

[7] A. Stuart and J. Voss, *Matrix Analysis and Algorithms*, 2009. `http://seehuhn.de/media/papers/numlinalg.pdf`,

Index