* Sustavi Linearnih Jednadžbi *

Zlatko Drmač

18. veljače 2002

Sadržaj

Glava 1

Sustavi linearnih jednadžbi

Jedan od osnovnih problema numeričke matematike je rješavanje linearnih sustava jednadžbi. Uovom poglavlju ćemo istraživati metode za rješavanje kvadratnih $n \times n$ sustava, tj. sustava od n jednadžbi sa n nepoznanica,

$$\begin{array}{ccccccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1j}x_j & + & \cdots & + & a_{1n}x_n & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2j}x_j & + & \cdots & + & a_{2n}x_n & = & b_2 \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & \vdots & \vdots \\
a_{i1}x_1 & + & a_{i2}x_2 & + & \cdots & + & a_{ij}x_j & + & \cdots & + & a_{in}x_n & = & b_i \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & \vdots & \vdots \\
a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nj}x_j & + & \cdots & + & a_{nn}x_n & = & b_n
\end{array}$$

Matrica $A = (\, a_{ij} \,)_{i,j=1}^{n} \in \mathbf{R}^{n \times n}$ je *matrica sustava*, a njeni elementi su *koeficijenti sustava*. Vektor $b = (\, b_i \,)_{i=1}^{n} \in \mathbf{R}^n$ je *vektor desne strane* sustava. Treba odrediti *vektor nepoznanica* $x = (\, x_i \,)_{i=1}^{n} \in \mathbf{R}^n$ tako da vrijedi $Ax = b$.
Kako znamo iz linearne algebre, za teorijsku matematiku je rješavanje sustava $Ax = b$ gotovo trivijalan problem, posebno u slučaju kada je matrica sustava kvadratna i regularna. Rješenje x je dano formulom $x = A^{-1}b$ u kojoj je A^{-1} inverzna matrica od A ($AA^{-1} = A^{-1}A = I$). Pri tome postoje eksplicitne formule i za elemente matrice A^{-1} i za samo rješenje x. Osim toga, svima dobro poznata Gaussova metoda eliminacija dolazi do rješenja u $O(n^3)$ elementarnih operacija[1]. Dakle, situacija je potpuno jasna: rješenje $x = A^{-1}b$ postoji, i to samo jedno, i znamo jednostavan algoritam koji to rješenje eksplicitno računa koristeći samo jednostavne aritmetičke operacije.
U primijenjenoj matematici, posebno u numeričkoj linearnoj algebri (grana numeričke matematike koja se bavi problemima linearne algebre) je situacija puno kompliciranija. Zašto? U numeričkoj matematici danas riješiti problem znači biti u stanju u

[1]Ovdje elementarna operacija označava zbrajanje, oduzimanje, množenje ili dijeljenje.

konkretnoj situaciji sa konkretnim podacima koristeći računalo *brzo* doći do dovoljno *točne numeričke aproksimacije* rješenja. Na primjer, ako su $n \times n$ matrica A i vektor b zapisani u nekim datotekama na disku, ili su dane procedure (potprogrami) koji generiraju A i b, onda je zadatak izračunati numeričke vrijednosti x_i, $i = 1, \ldots, n$.
Ono po čemu su računala poznata je brzina – dobro jednoprocesorsko računalo može napraviti npr. 10^9 operacija u sekundi. Međutim, ono što je osnovna značajka moderne numeričke matematike je da joj u primjenama dolaze problemi sve većih dimenzija. Kako je broj operacija u Gaussovim eliminacijama $O(n^3)$, to znači da npr. za $n = 10^5$ je broj operacija reda veličine 10^{15} pa brznimo od 10^9 operacija u sekundi dobivamo vrijeme izvršavanja[2] oko 10^6 sekundi, što je više od deset dana. U ozbiljnim primjenama treba u procesu projektiranja puno puta rješavati takve sustave – jasno nam je onda da brzina računala ne rješava uvijek problem razumno (ili dovoljno) brzog rješavanja linearnog sustava. Dakle, problem koji je matematički posve jednostavan u praksi može biti puno izazovniji i netrivijalniji.
U divljenju brzini kojom računalo zbraja, oduzima ili množi brojeve često zaboravljamo da su rezultati tih operacija uglavnom **netočni**. Sjetimo se, računalo reprezentira brojeve i izvršava računske operacije koristeći fiksirani broj znamenki – to znači da se rezultati operacija zaokružuju. Dakle, moguće je da je svaka od $O(n^3)$ operacija u Gaussovom algoritmu izvršena sa greškom zaokruživanja. Koliko onda možemo vjerovati izračunatom rješenju?

1.1 Vodič kroz ovo poglavlje

Materijal u ovom poglavlju je organiziran u više nivoa tako da ga mogu čitati i početnici i napredniji čitatelji. Sljedeći pregled bi trebao pomoći čitatelju pri planiranju proučavanja ponuđenog materijala.

> *1. nivo*: Proučiti i shvatiti barem jedan primjer iz sekcije 1.2. Materijal iz sekcije 1.3 čitati do iskaza teorema. Pažljivo, uz papir i olovku, obraditi primjere i opis Gaussovih eliminacija na primjeru male dimenzije. Čitatelj na ovom nivou treba naučiti kako funkcioniraju Gaussove eliminacije, uočiti vezu između procesa eliminacija i faktorizacije matrice koeficijenata sustava, svladati algoritme za rješavanje trokutastih sustava, te biti u stanju na ruke riješiti sustave manje dimenzije.
>
> *2. nivo*: Materijal iz sekcije 1.3 svladati u potpunosti.

[2]Ovdje namjerno pojednostavljujemo ocjenu vremena izvršavanja. Za precizniju procjenu je potrebno uračunati i vrijeme pristupa memoriji i dohvaćanje podataka, veličinu cache memorije itd. U ovom trenutku je važno je dobiti osjećaj za red veličine.

3. nivo: Sekcije 1.3 i 1.4 svladati u potpunosti. Po potrebi se služiti materijalom iz dodataka ili druge literature. Analizirati i shvatiti sve primjere. Shvatiti važnost pivotiranja za numeričku stabilnost Gaussovih eliminacija.

4. nivo: Na ovom nivou čitatelj bet poteškoća čita sav materijal i služi se programima za rješavanje sustava na računalu.

Vježbe se sastoje od zadataka koji su također podijeljeni u grupe po težiti: čitatelj koji čita materijal na i-tom nivou bi trebao bez posebnih poteškoća riješiti zadatke iz prvih i grupa zadataka.

1.2 Primjeri: Kako nastaje linearni sustav jednadžbi

U ovoj sekciji dajemo niz primjera problema iz primijenjene matematike čije rješavanje je bazirano na sustavima linearnih jednadžbi.

Primjer 1.2.1 *Promotrimo sljedeći rubni problem:*

$$-\frac{d^2}{dx^2}u(x) = f(x), \quad 0 < x < 1, \tag{1.1}$$

$$u(0) = u(1) = 0. \tag{1.2}$$

Rješenje u problema (1.1, 1.2) ćemo aproksimirati na skupu od konačno mnogo točaka iz segmenta $[0,1]$. *Odaberimo prirodan broj* n *i definirajmo*

$$h = \frac{1}{n+1}, \quad x_i = ih, \quad i = 0, \ldots, n+1. \tag{1.3}$$

Sada promatramo vrijednosti $u_i = u(x_i)$, $i = 0, \ldots, n+1$. *Iz uvjeta (1.2) je odmah* $u_0 = u_{n+1} = 0$. *Iz Taylorovog teorema je*

$$u(x_i + h) = u(x_i) + u'(x_i)h + \frac{u''(x_i)}{2}h^2 + \frac{u'''(x_i)}{6}h^3 + \frac{u^{(4)}(x_i + \alpha_i)}{24}h^4, \tag{1.4}$$

$$u(x_i - h) = u(x_i) - u'(x_i)h + \frac{u''(x_i)}{2}h^2 - \frac{u'''(x_i)}{6}h^3 + \frac{u^{(4)}(x_i + \zeta_i)}{24}h^4, \tag{1.5}$$

gdje su $\alpha_i \in (x_i, x_i + h)$, $\zeta_i \in (x_i - h, x_i)$. *Zbrajanjem jednadžbi (1.4) i (1.5) dobijemo za* $i = 1, \ldots, n$

$$u_{i+1} + u_{i-1} = 2u_i + u''(x_i)h^2 + (u^{(4)}(x_i + \alpha_i) + u^{(4)}(x_i + \zeta_i))\frac{h^4}{24}, \quad tj. \tag{1.6}$$

$$-u''(x_i) = \frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} - \mathbf{e}_i, \tag{1.7}$$

gdje je

$$\mathbf{e}_i = -(u^{(4)}(x_i + \alpha_i) + u^{(4)}(x_i + \zeta_i))\frac{h^2}{24}.$$

Matrično to možemo zapisati kao

$$\underbrace{\begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix}}_{T_n} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-2} \\ u_{n-1} \\ u_n \end{bmatrix}}_{u} = h^2 \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-2} \\ f_{n-1} \\ f_n \end{bmatrix}}_{f} + h^2 \underbrace{\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \\ \vdots \\ \mathbf{e}_{n-2} \\ \mathbf{e}_{n-1} \\ \mathbf{e}_n \end{bmatrix}}_{\mathbf{e}} \tag{1.8}$$

U jednadžbi $T_n u = h^2 f + h^2 \mathbf{e}$ član $\mathbf{e}$ ne poznamo pa ga zanemarujemo, tj. pokušat ćemo riješiti $T_n \hat{u} = h^2 f$. Primijetimo da pod određenim uvjetima možemo očekivati da je $\|\mathbf{e}\|_2$ mali broj, $\|\mathbf{e}\|_2 = O(h^2)$.
Tek da ilustriramo, uzmimo npr. $f(x) = 16\pi^2 \sin 4\pi x$, za kojeg znamo točno rješenje $u(x) = \sin 4\pi x$. Uzet ćemo $n = 10$ i $n = 30$ i vidjeti koliko su dobre dobivene aproksimacije. Rezultati su dani na slikama 1.1 i **??**.
– U pripremi –

Slika 1.1: Numeričko rješenje dobiveno sa 30 čvorova.

Za iskusnije čitatelje ćemo odmah malo prodiskutirati dobiveno rješenje. (Ostali mogu preskočiti sljedeću diskusiju.) Lako se pokaže da je matrica T_n regularna. Neka je $\hat{u} = h^2 T_n^{-1} f$. Imamo

$$u - \hat{u} = T_n^{-1}(h^2 f + h^2 \mathbf{e}) - h^2 T_n^{-1} f = h^2 T_n^{-1} \mathbf{e}, \tag{1.9}$$

pa je

$$\frac{\|u - \hat{u}\|_2}{\|\hat{u}\|_2} = \frac{\|T_n^{-1}\mathbf{e}\|_2}{\|T_n^{-1} f\|_2} \leq \|T_n\|_2 \|T_n^{-1}\|_2 \frac{\|\mathbf{e}\|_2}{\|f\|_2}. \tag{1.10}$$

Pri tome smo koristili sljedeće nejednakosti:

$$\begin{aligned} \|T_n^{-1}\mathbf{e}\|_2 &\leq \|T_n^{-1}\|_2 \|\mathbf{e}\|_2 \quad \textit{(jer je } \|T_n^{-1}\|_2 = \max_{x \neq 0} \frac{\|T_n^{-1} x\|_2}{\|x\|_2}\textit{)}, \\ \|T_n^{-1} f\|_2 &\geq \frac{\|f\|_2}{\|T_n\|_2} \quad \textit{(jer je } \|T_n\|_2 = \max_{x \neq 0} \frac{\|T_n x\|_2}{\|x\|_2} = \max_{y \neq 0} \frac{\|y\|_2}{\|T_n^{-1} y\|_2}\textit{)}. \end{aligned}$$

Vidimo da nejednakost u relaciji (1.10) može za neke izbore vektora $\mathbf{e}$ *i* f *preći u jednakost, što znači da je relacija (1.10) realistična ocjena greške diskretizacije* $\mathbf{e}$ *na aproksimaciju rješenja polaznog rubnog problema (1.1, 1.2).*
Mi naravno ne možemo točno odrediti niti $\hat{u}$ *jer računamo u aritmetici sa konačnom preciznosti. Neka je* $\tilde{u}$ *izračunata aproksimacija vektora* $\hat{u}$*. Pitanje je koliku točnost aproksimacije treba imati* $\tilde{u}$*. Iz relacije (1.10) slijedi da je zadovoljavajuća točnost postignuta ako je* $\|\hat{u}-\tilde{u}\|_2/\|\tilde{u}\|_2$ *najviše reda veličine* $\kappa_2(T_n)\|\mathbf{e}\|_2/\|f\|_2$ *(sjetimo se da mi zapravo želimo aproksimirati* u*).*

Primjer 1.2.2 *U primjenama se često javljaju integralne jednadžbe – tražena funkcija* $u(x)$ *zadovoljava jednadžbu u kojoj se ona javlja i kao podintegralna funkcija.*

Primjer 1.2.3 *U pripremi.*

Primjer 1.2.4 *U pripremi.*

Primjer 1.2.5 *U pripremi.*

1.3 Gaussove eliminacije i trokutaste faktorizacije

Metoda Gaussovih eliminacija je svakako najstariji, najjednostavniji i najpoznatiji algoritam za rješavanje sustava linearnih jednadžbi $Ax = b$. Ideja je jednostavna: Za riješiti sustav

$$\begin{array}{rcrcl} 2x_1 & - & x_2 & = & 1 \\ -x_1 & + & 2x_2 & = & 1 \end{array}$$

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

$$-\underbrace{\frac{1}{2}(1+x_2)}_{x_1} + 2x_2 = 1, \quad \text{tj.} \quad \frac{3}{2}x_2 = \frac{3}{2}, \quad \text{tj.} \quad x_2 = 1,$$

odakle je $x_1 = 1$. Kažemo da smo x_1 *eliminirali* iz druge jednadžbe.
Ova ideja se lako generalizira 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 kada sam proces eliminacija interpretiramo kao faktorizaciju matrice sustava A na produkt trokutastih matrica.

1.3.1 Matrični zapis metode eliminacija

Primjer 1.3.1 *Riješimo sljedeći sustav jednadžbi:*

$$\begin{array}{rcrcrcr} 5x_1 & + & x_2 & + & 4x_3 & = & 19 \\ 10x_1 & + & 4x_2 & + & 7x_3 & = & 39 \\ -15x_1 & + & 5x_2 & - & 9x_3 & = & -32 \end{array} \equiv \underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ -15 & 5 & -9 \end{pmatrix}}_{A=(a_{ij})_{i,j=1}^{3}} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 19 \\ 39 \\ -32 \end{pmatrix}}_{b=(b_i)_{i=1}^{3}}. \tag{1.11}$$

Koristimo metodu supstitucija, odnosno eliminacija: prvo iz prve jednadžbe izrazimo x_1 pomoću x_2 i x_3, te to uvrstimo u zadnje dvije jednadžbe, koje postaju dvije jednadžbe sa dvije nepoznanice (x_2 i x_3). Dobijemo

$$x_1 = \frac{1}{5}(19 - x_2 - 4x_3),$$

pa druga jednadžba sada glasi

$$\frac{10}{5}(19 - x_2 - 4x_3) + 4x_2 + 7x_3 = 39, \quad tj. \quad -\frac{10}{5}(x_2 + 4x_3) + 4x_2 + 7x_3 = 39 + (-\frac{10}{5}19).$$

Dakle, efekt ove transformacije je ekvivalentno prikazan kao rezultat množenja prve jednadžbe s

$$-\frac{a_{21}}{a_{11}} = -\frac{10}{5} = -2$$

i zatim njenim dodavanjem (pribrajanjem) drugoj jednadžbi. Druga jednadžba sada glasi

$$2x_2 - x_3 = 1.$$

Ako ovu transformaciju sustava zapišemo matrično, imamo

$$\underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ 15 & 5 & -9 \end{pmatrix}}_{A} \mapsto \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{L^{(2,1)}} \underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 10 & 4 & 7 \\ 15 & 5 & -9 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 15 & 5 & -9 \end{pmatrix}}_{A^{(1)}=(a_{ij}^{(1)})_{i,j=1}^{3}}.$$

Nepoznanicu x_1 eliminiramo iz zadnje jednadžbe ako prvu pomnožimo s

$$-\frac{a_{31}^{(1)}}{a_{11}^{(1)}} = -\frac{15}{5} = -3$$

i onda je pribrojimo zadnjoj. To znači sljedeću promjenu matrice koeficijenata:

$$\underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 15 & 5 & -9 \end{pmatrix}}_{A^{(1)}} \mapsto \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}}_{L^{(3,1)}} \underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 15 & 5 & -9 \end{pmatrix}}_{A^{(1)}} = \underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 2 & -21 \end{pmatrix}}_{A^{(2)} = (a_{ij}^{(2)})_{ij=1}^{3}}.$$

Vektor desne strane je u ove dvije transformacije promijenjen u

$$\underbrace{\begin{pmatrix} 19 \\ 39 \\ -32 \end{pmatrix}}_{b} \mapsto \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{L^{(2,1)}} \begin{pmatrix} 19 \\ 39 \\ -32 \end{pmatrix} = \underbrace{\begin{pmatrix} 19 \\ 1 \\ -32 \end{pmatrix}}_{b^{(1)}} \mapsto \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}}_{L^{(3,1)}} \begin{pmatrix} 19 \\ 1 \\ -32 \end{pmatrix} = \underbrace{\begin{pmatrix} 19 \\ 1 \\ -59 \end{pmatrix}}_{b^{(2)}}.$$

Novi, ekvivalentni, sustav je $A^{(2)}x = b^{(2)}$, tj.

$$\begin{array}{rcrcrcr} 5x_1 & + & x_2 & + & 4x_3 & = & 19 \\ & & 2x_2 & - & 1x_3 & = & 1 \\ & & 2x_2 & - & 21x_3 & = & -59, \end{array} \tag{1.12}$$

u kojem su druga i treća jednadžba sustav od dvije jednadžbe sa dvije nepoznanice. Očito je da rješenje $x = (x_1, x_2, x_3)^\tau$ sustava (1.11) zadovoljava i sustav (1.12). Obratno, ako trojka x_1, x_2, x_3 zadovoljava (1.12), onda množenjem prve jednadžbe u (1.12) s 2 i zatim pribrajanjem drugoj jednadbi, dobijemo drugu jednadžbu sustava (1.11). Na sličan način iz prve i treće jednadžbe sustava (1.12) rekonstruiramo treću jednadžbu polaznog sustava (1.11). U tom smislu kažemo da sus sustavi (1.11) i (1.12) ekvivalentni: imaju isto rješenje.

Nadalje, primijetimo da smo proces eliminacija (tj. izražavanja nepoznanice x_1 pomoću x_2 i x_3 i eliminiranjem x_1 iz zadnje dvije jednadžbe) jednostavno opisali matričnim operacijama. Eliminaciju nepoznanice x_1 smo prikazali kao rezultat množenja matrice koeficijenata i vektora desne strane s lijeva jednostavnim matricama $L^{(2,1)}$ i $L^{(3,1)}$.

Jasno je da je sustav (1.12) jednostavniji od polaznog. Zato sada nastavljamo s primjenom iste strategije: iz treće jednadžbe eliminiramo x_2 tako što drugu jednadžbu pomonožimo s

$$-\frac{a_{32}^{(2)}}{a_{22}^{(2)}} = -1$$

i pribrojimo je trećoj. Tako treća jednadžba postaje

$$-20x_3 = -60,$$

a cijeli sustav ima oblik

$$\begin{array}{rcrcrcl} 5x_1 & + & x_2 & + & 4x_3 & = & 19 \\ & & 2x_2 & - & x_3 & = & 1 \\ & & & & -20x_3 & = & -60 \end{array} . \tag{1.13}$$

Transformaciju eliminacije x_2 iz treće jednadžbe možemo matrično zapisati kao transformaciju matrice koeficijenata

$$\underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 2 & -21 \end{pmatrix}}_{A^{(2)}} \mapsto \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}}_{L^{(3,2)}} \underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 2 & -21 \end{pmatrix}}_{A^{(2)}} = \underbrace{\begin{pmatrix} 5 & 1 & 4 \\ 0 & 2 & -1 \\ 0 & 0 & -20 \end{pmatrix}}_{A^{(3)}} \tag{1.14}$$

i transformaciju vektora desne strane

$$\underbrace{\begin{pmatrix} 19 \\ 1 \\ -59 \end{pmatrix}}_{b^{(2)}} \mapsto \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}}_{L^{(3,2)}} \underbrace{\begin{pmatrix} 19 \\ 1 \\ -59 \end{pmatrix}}_{b^{(2)}} = \underbrace{\begin{pmatrix} 19 \\ 1 \\ -60 \end{pmatrix}}_{b^{(3)} = (b_i^{(3)})_{i=1}^3} \tag{1.15}$$

Sustav (1.13), koji je ekvivalentan polaznom, lako riješimo:

1. *Iz treće jednadžbe je* $x_3 = \dfrac{-60}{-20} = 3$ *;*
2. *Iz druge jednadžbe je* $x_2 = \dfrac{1}{2}(1 + x_3) = 2$ *;*
3. *Iz prve jednadžbe je* $x_1 = \dfrac{1}{5}(19 - x_2 - 4x_3) = 1.$

Jednostavna provjera potvrđuje da je sa ovim x_1, x_2, x_3 riješen polazni sustav (1.11).

Analizirajmo postupak rješavanja u prethodnom primjeru. Relacija

$$A^{(3)} = L^{(3,2)} L^{(3,1)} L^{(2,1)} A$$

zaslužuje posebnu pažnju. Matrica $A^{(3)}$ je gornje trokutasta, a produkt $L^{(3,2)} L^{(3,1)} L^{(2,1)}$ je donje trokutasta matrica. Dakle, polaznu matricu A smo množenjem s lijeva donje trokutastom matricom načinili gornje trokutastom. To možemo pročitati i ovako:

$$A = LA^{(3)}, \quad L = (L^{(2,1)})^{-1} (L^{(3,1)})^{-1} (L^{(3,2)})^{-1},$$

gdje je L donje trokutasta matrica. Lako provjerimo da je

$$L = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{(L^{(2,1)})^{-1}} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix}}_{(L^{(3,1)})^{-1}} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}}_{(L^{(3,2)})^{-1}} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix}.$$

Dakle, matricu A smo napisali kao produkt donje trokutaste i gornje trokutaste matrice, $A = LA^{(3)}$. Gornje trokutastu matricu u ovom kontekstu obično označavamo s $U = A^{(3)}$, pa je A rastavljen na produkt $A = LU$. Govorimo o LU *faktorizaciji* matrice A. Uočimo da je računanje produkta koji definira matricu L jednostavno: Inverze od $L^{(2,1)}$, $L^{(3,1)}$, $L^{(3,2)}$ dobijemo samo promjenom predznaka netrivijalnih elemenata u donjem trokutu, a cijeli produkt je jednostavno stavljanje tih elemenata na odgovarajuće pozicije u matrici L. Sada još primijetimo da je relacija (1.13) zapravo linearni sustav

$$Ux = b^{(3)}, \text{ gdje je } b^{(3)} = L^{(3,2)}L^{(3,1)}L^{(2,1)}b = L^{-1}b.$$

Jasno, $x = A^{-1}b = (LU)^{-1}b = U^{-1}L^{-1}b$.
Dakle, u terminima matrice A i vektora b, linearni sustav u primjeru 1.3.1 je riješen metodom koja se sastoji od tri glavna koraka:

1. *Matricu sustava A faktorizirati u obliku $A = LU$, gdje je L donje trokutasta, a U gornje trokutasta matrica.*

2. *Rješavanjem donje trokutastog sustava $Ly = b$ odrediti vektor $y = L^{-1}b$.*

3. *Rješavanjem gornje trokutastog sustava $Ux = y$ odrediti vektor $x = U^{-1}y = U^{-1}(L^{-1}b)$.*

Ovakav zapis metode opisane u primjeru 1.3.1 ima niz prednosti:

– Operacije su iskazane u terminima matrice A i desne strane b, a ne u terminima izražavanja neke nepoznanice pomoću ostalih. Umjesto "x_1 izrazimo pomoću $x_2, x_3, \ldots$" i sl., operacije izražavamo jezikom operacija sa matricama i vektorima. To omogućuje jednostavnu i sustavnu primjenu opisane metode na sustav sa proizvoljnim brojem nepoznanica. Sam linearni sustav je u računalu pohranjen kao matrica koeficijenata A i vektor desne strane b. Dakle, ovakav zapis metode eliminacija je prirodan.

– Ponekad u primjenama rješavamo nekoliko linearnih sustava sa istom matricom A, ali sa nizom različitih desnih strana b. Vidimo da je u tom slučaju transformacije na matrici A dovoljno napraviti jednom (prvi korak u gornjem zapisu metode), a zatim za različite desne strane provesti samo zadnja dva koraka.

1.3.2 Trokutasti sustavi: rješavanje supstitucijama naprijed i unazad

Trokutasti sustavi jednadžbi su laki za riješiti. Pogledajmo na primjer donje trokutasti sustav $Lx = b$ dimenzije $n = 4$:

$$\begin{pmatrix} \ell_{11} & 0 & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} & 0 \\ \ell_{41} & \ell_{42} & \ell_{43} & \ell_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}.$$

Neka je matrica L regularna. To znači da je $\ell_{ii} \neq 0$ za $i = 1, 2, 3, 4$. Očito je

$$\begin{aligned} x_1 &= \frac{b_1}{\ell_{11}} \\ x_2 &= \frac{1}{\ell_{22}}(b_2 - \ell_{21}x_1) \\ x_3 &= \frac{1}{\ell_{33}}(b_3 - \ell_{31}x_1 - \ell_{32}x_2) \\ x_4 &= \frac{1}{\ell_{44}}(b_4 - \ell_{41}x_1 - \ell_{42}x_2 - \ell_{43}x_3). \end{aligned}$$

Vidimo da x_1 možemo odmah izračunati, a za $i > 1$ formula za x_i je funkcija od b_i, i–tog retka matrice L i nepoznanica $x_1, \ldots, x_{i-1}$ koje su prethodno već izračunate. Dakle, izračunamo prvo x_1 pa tu vrijednost uvrstimo u izraz koji daje x_2; zatim x_1 i x_2 uvrstimo u izraz za x_3 itd. Ovakav postupak zovemo *supstitucije naprijed.*

Algoritam 1.3.1 *Rješavanje linearnog sustava jednadžbi $Lx = b$ sa regularnom donje trokutastom matricom $L \in \mathbf{R}^{n\times n}$.*

/ Supstitucije naprijed za $Lx = b$ */*
$x_1 = \dfrac{b_1}{\ell_{11}}$;
za $i = 2, \ldots, n$ {
$\quad x_i = \dfrac{1}{\ell_{ii}}(b_i - \sum_{j=1}^{i-1} \ell_{ij}x_j)$; }

Prebrojimo operacije u gornjem algoritmu:

- Dijeljenja: n ;
- Množenja: $1 + 2 + \cdots + (n-1) = \dfrac{1}{2}n(n-1)$;

– Zbrajanja i oduzimanja: $1 + 2 + \cdots + (n-1) = \frac{1}{2}n(n-1)$.

Dakle, ukupna složenost je $O(n^2)$.
Gornje trokutaste sustave rješavamo na sličan način. Ako je sustav $Ux = b$ oblika

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}, \quad \prod_{i=1}^{4} u_{ii} \neq 0,$$

onda, polazeći od zadnje jednadžbe unazad, imamo

$$\begin{aligned} x_4 &= \frac{b_4}{u_{44}} \\ x_3 &= \frac{1}{u_{33}}(b_3 - u_{34}x_4) \\ x_2 &= \frac{1}{u_{22}}(b_2 - u_{23}x_3 - u_{24}x_4) \\ x_1 &= \frac{1}{u_{11}}(b_1 - u_{12}x_2 - u_{13}x_3 - u_{14}x_4). \end{aligned}$$

Ovakav postupak zovemo *supstitucije unazad.*

Algoritam 1.3.2 *Rješavanje linearnog sustava jednadžbi $Ux = b$ sa regularnom gornje trokutastom matricom $U \in \mathbf{R}^{n\times n}$.*

/ Supstitucije unazad za $Ux = b$ */*
$x_n = \frac{b_n}{u_{nn}}$;
za $i = n-1, \ldots, 1$ {
$x_i = \frac{1}{u_{ii}}(b_i - \sum_{j=i+1}^{n} u_{ij}x_j)$; }

Kao i kod supstitucija naprijed, složenost ovog algoritma je $O(n^2)$.

1.3.3 LU faktorizacija

Sada kada smo uočili da se rješavanje linearnog sustava $Ax = b$ faktoriziranjem matrice A svodi na trokutaste sustave, ostaje nam posebno proučiti faktorizaciju matrice $A \in \mathbf{R}^{n\times n}$ na produkt donje i gornje trokutaste matrice. Zanima nas proizvoljna dimenzija n, ali ćemo zbog jednostavnosti razmatranja na početku sve ideje ilustrirati

na primjeru $n = 5$. Neka je

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix}.$$

Sjetimo se, eliminacija prve nepoznanice je manifestirana poništavanjem koeficijenata na pozicijama $(2,1),(3,1),\ldots,(n,1)$. To možemo napraviti u jednom potezu.[3] Ako definiramo matricu

$$L^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -\dfrac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ -\dfrac{a_{31}}{a_{11}} & 0 & 1 & 0 & 0 \\ -\dfrac{a_{41}}{a_{11}} & 0 & 0 & 1 & 0 \\ -\dfrac{a_{51}}{a_{11}} & 0 & 0 & 0 & 1 \end{pmatrix},$$

onda je x_1 eliminiran iz svih jednadžbi osim prve, tj.

$$A^{(1)} \equiv L^{(1)}A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} & a_{35}^{(1)} \\ 0 & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} & a_{45}^{(1)} \\ 0 & a_{52}^{(1)} & a_{53}^{(1)} & a_{54}^{(1)} & a_{55}^{(1)} \end{pmatrix}.$$

Objasnimo oznake koje koristimo za elemente matrice $A^{(1)}$. Općenito, elementi od $A^{(1)}$ su označeni s $a_{ij}^{(1)}$, $1 \le i,j \le n$. Međutim, elementi prvog retka u $A^{(1)}$ su jednaki prvom retku u A, $a_{1j}^{(1)} = a_{1j}$, $1 \le j \le n$, pa smo to eksplicitno naznačili u zapisu matrice $A^{(1)}$.

Primijetimo da je transformaciju $A \mapsto A^{(1)}$ moguće izvesti samo ako je

$$a_{11} \neq 0. \tag{1.16}$$

[3] U primjeru 1.3.1 smo zbog jednostavnosti poništavali koeficijente jedan po jedan.

Također, lako se uvjerimo da je

$$(L^{(1)})^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & 0 & 1 & 0 & 0 \\ \frac{a_{41}}{a_{11}} & 0 & 0 & 1 & 0 \\ \frac{a_{51}}{a_{11}} & 0 & 0 & 0 & 1 \end{pmatrix},$$

te da iz $A = (L^{(1)})^{-1}A^{(1)}$ slijedi

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{a_{21}}{a_{11}} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22}^{(1)} \end{pmatrix}.$$

Jednostavno, dobili smo faktorizaciju vodeće 2×2 podmatrice od A. Uvjet za izvod ove faktorizacije je bio (1.16). Stavimo

$$\alpha_2 \equiv \det \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = a_{11}a_{22}^{(1)}.$$

Ako je $\alpha_2 \neq 0$, onda je i $a_{22}^{(1)} \neq 0$ pa je dobro definirana matrica

$$L^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ 0 & -\frac{a_{42}^{(1)}}{a_{22}^{(1)}} & 0 & 1 & 0 \\ 0 & -\frac{a_{52}^{(1)}}{a_{22}^{(1)}} & 0 & 0 & 1 \end{pmatrix} \quad \text{i njen inverz} \quad (L^{(2)})^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ 0 & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & 0 & 1 & 0 \\ 0 & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & 0 & 0 & 1 \end{pmatrix}.$$

Vrijedi

$$A^{(2)} \equiv L^{(2)}A^{(1)} = L^{(2)}L^{(1)}A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \end{pmatrix}. \tag{1.17}$$

(Uočimo da oznake u relaciji (1.17) naglašavaju da je u matrici $A^{(2)} = \left(a_{ij}^{(2)}\right)_{i,j=1}^{n}$ prvi redak jednak prvom retku od A, a drugi redak jednak drugom retku matrice $A^{(1)}$.)

Ako sada u relaciji $A = (L^{(1)})^{-1}(L^{(2)})^{-1}A^{(2)}$ izračunamo produkt $(L^{(1)})^{-1}(L^{(2)})^{-1}$ dobijemo

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & 0 & 1 & 0 \\ \frac{a_{51}}{a_{11}} & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \end{pmatrix}, \tag{1.18}$$

odakle zaključujemo da vrijedi

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} \\ 0 & 0 & a_{33}^{(2)} \end{pmatrix}.$$

Dakle, ako je $a_{11} \neq 0$ i $\alpha_2 \neq 0$, onda smo dobili trokutastu faktorizaciju vodeće 3×3 podmatrice od A. Stavimo

$$\alpha_3 \equiv \det \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = a_{11}a_{22}^{(1)}a_{33}^{(2)}.$$

Ako je $\alpha_3 \neq 0$ onda je i $a_{33}^{(2)} \neq 0$ pa su dobro definirane matrice

$$L^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ 0 & 0 & -\frac{a_{53}^{(2)}}{a_{33}^{(2)}} & 0 & 1 \end{pmatrix}, \quad (L^{(3)})^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ 0 & 0 & \frac{a_{53}^{(2)}}{a_{33}^{(2)}} & 0 & 1 \end{pmatrix},$$

i vrijedi

$$A^{(3)} \equiv L^{(3)}A^{(2)} = L^{(3)}L^{(2)}L^{(1)}A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & a_{54}^{(3)} & a_{55}^{(3)} \end{pmatrix}.$$

Ako izračunamo produkt $(L^{(1)})^{-1}(L^{(2)})^{-1}(L^{(3)})^{-1}$, onda vidimo da vrijedi

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ \frac{a_{51}}{a_{11}} & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & \frac{a_{53}^{(2)}}{a_{33}^{(2)}} & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & a_{54}^{(3)} & a_{55}^{(3)} \end{pmatrix},$$

te da je

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} \end{pmatrix}.$$

Ponovo zaključujemo na isti način: definiramo

$$\alpha_4 \equiv \det \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = a_{11}a_{22}^{(1)}a_{33}^{(2)}a_{44}^{(3)}$$

Ako je $\alpha_4 \neq 0$, onda je i $a_{44}^{(3)} \neq 0$, pa su dobro definirane matrice

$$L^{(4)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{a_{54}^{(3)}}{a_{44}^{(3)}} & 1 \end{pmatrix}, \quad (L^{(4)})^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{a_{54}^{(3)}}{a_{44}^{(3)}} & 1 \end{pmatrix}. \tag{1.19}$$

Lako provjerimo da vrijedi

$$A^{(4)} \equiv L^{(4)}A^{(3)} = L^{(4)}L^{(3)}L^{(2)}L^{(1)}A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & 0 & a_{55}^{(4)} \end{pmatrix}.$$

te da je, nakon računanja produkta $(L^{(1)})^{-1}(L^{(2)})^{-1}(L^{(3)})^{-1}(L^{(4)})^{-1}$,

$$A = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ \frac{a_{51}}{a_{11}} & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & \frac{a_{53}^{(2)}}{a_{33}^{(2)}} & \frac{a_{54}^{(3)}}{a_{44}^{(3)}} & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & 0 & a_{55}^{(4)} \end{pmatrix}}_{U}. \tag{1.20}$$

Vidimo da je izvedivost operacija koje su dovele do faktorizacije $A = LU$ ovisila o uvjetima

$$a_{11} \neq 0, \quad a_{22}^{(1)} \neq 0, \quad a_{33}^{(2)} \neq 0, \quad a_{44}^{(3)} \neq 0.$$

Također smo uočili da su ti uvjeti osigurani ako su u matrici A determinante glavnih podmatrica dimenzija $1, 2, \ldots, n-1$ različite od nule. To je u našem primjeru značilo uvjete

$$\alpha_1 \equiv a_{11} \neq 0, \quad \alpha_2 \neq 0, \quad \alpha_3 \neq 0, \quad \alpha_4 \neq 0.$$

Brojeve a_{11}, $a_{22}^{(1)}$, $a_{33}^{(2)}$, $a_{44}^{(3)}$ zovemo *pivotni elementi* ili kratko *pivoti*. Brojevi α_1, α_2, α_3, α_4 su *glavne minore* matrice A. Dakle, možemo zaključiti sljedeće:

♣ *Ako je prvih $n-1$ minora matrice A različito od nule, onda su i svi pivotni elementi različiti od nule i Gaussove eliminacije daju LU faktorizaciju matrice A.*

U tom slučaju sljedeći algoritam računa faktorizaciju $A = LU$.

Algoritam 1.3.3 *Računanje LU faktorizacije matrice A.*

$L = I$;
za $k = 1, \ldots, n-1$ {
  za $j = k+1, \ldots, n$ {
    $\ell_{jk} = \dfrac{a_{jk}^{(k-1)}}{a_{kk}^{(k-1)}}$;
    $a_{jk}^{(k)} = 0$; }
  za $j = k+1, \ldots, n$ {
    za $i = k+1, \ldots, n$ {
      $a_{ij}^{(k)} = a_{ij}^{(k-1)} - \ell_{ik} a_{kj}^{(k-1)}$; }}}
$U = A^{(n-1)} = \left(a_{ij}^{(n-1)} \right)$.

Sljedeći teorem i formalno dokazuje egzistenciju i jedinstvenost LU faktorizacije.

Teorem 1.3.1 *Neka je* $A \in \mathbf{R}^{n\times n}$ *i neka su determinante glavnih podmatrica* $A(1:k, 1:k)$ *različite od nule za* $k = 1, 2, \ldots, n-1$. *Tada postoji donje trokutasta matrica* L *sa jedinicama na dijagonali i postoji gornje trokutasta matrica* U, *tako da vrijedi* $A = LU$. *Ako takva faktorizacija* $A = LU$ *postoji i ako je još i matrica* A *regularna, onda je faktorizacija jedinstvena: postoji točno jedna matrica* L *i točno jedna matrica* U *sa ovim svojstvima. Tada je i* $\det(A) = \prod_{i=1}^{n} u_{ii}$.

Dokaz: Dokažimo prvo jedinstvenost LU faktorizacije. Neka postoje dvije takve faktorizacije,

$$A = LU = L'U'.$$

Ako je A regularna onda su i L, U, L', U' također regularne matrice pa vrijedi

$$L^{-1}L' = U(U')^{-1}$$

U gornjoj relaciji imamo jednakost donje trokutaste i gornje trokutaste matrice – znači da na obe strane jednakosti stoje dijagonalne matrice. Nadalje, L i L' po pretpostavci imaju jedinice na dijagonali, a zbog činjenice da se na dijagonali produkta donje trokutastih matrica nalaze produkti dijagonalinih elemenata matrica koje se množe su na dijagonali od $L^{-1}L'$ jedinice. Dakle, $L^{-1}L' = I$, tj. $L = L'$. Tada je i $U = U'$. Dokažimo sada egzistenciju LU faktorizacije. Induktivni dokaz je zapravo već skiciran u opisu računanja faktorizacije 5×5 matrice. Pogledajmo kako uvjeti teorema omogućuju prelaz sa $A^{(k)}$ na $A^{(k+1)}$, gdje je

$$A^{(k)} = L^{(k)} \cdots L^{(1)} A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & \cdots & a_{1,n-1} & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & \cdots & \cdots & a_{2,n-1}^{(1)} & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & \cdots & \cdots & a_{3,n-1}^{(2)} & a_{3n}^{(2)} \\ 0 & 0 & 0 & \ddots & \cdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & & \ddots & a_{kk}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ 0 & 0 & & & 0 & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ \vdots & \vdots & & & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & & 0 & a_{n,k+1}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}.$$

Kako je produkt $(L^{(k)} \cdots L^{(1)})^{-1}$ donje trokutasta matrica sa jedinicama na dijagonali, zaključujemo da je

$$\det(A(1:k+1, 1:k+1)) = a_{11} a_{22}^{(1)} a_{33}^{(2)} \cdots a_{kk}^{(k-1)} a_{k+1,k+1}^{(k)} \neq 0.$$

Odavde je i $a_{k+1,k+1}^{(k)} \neq 0$ pa možemo definirati matricu $L^{(k+1)}$ koja će poništiti elemente ispod dijagonale u $(k+1)$–om stupcu i dati $A^{(k+1)} = L^{(k+1)} A^{(k)}$. Jasno je da nakon konačno koraka dobijemo matricu $A^{(n-1)}$ koja je gornje trokutasta. ■

Komentar 1.3.1 *Primijetimo, ako je A regularna i ako ima LU faktorizaciju, onda su nužno i sve glavne podmatrice $A(1:k,1:k)$ regularne. To slijedi iz činjenice da je*

$$\det(A(1:k,1:k)) = \prod_{i=1}^{k} u_{ii}, \quad k = 1,\ldots,n.$$

1.3.4 LU faktorizacija sa pivotiranjem

Jedan očit problem sa LU faktorizacijom koju smo opisali u prethodnoj sekciji je da za njeno računanje prema opisanom algoritmu matrica A mora imati specijalnu strukturu: sve njene glavne podmatrice do uključivo reda $n-1$ moraju biti regularne. Sljedeći primjer ilustrira taj problem.

Primjer 1.3.2 *Neka je matrica sustava $Ax = b$ dana s*

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

Ova matrica je regularna, $\det(A) = -1$, pa sustav uvijek ima rješenje, ali A očito nema LU faktorizaciju. Jer,

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

povlači da je $\ell_{11}u_{11} = 0$, pa je $\ell_{11} = 0$ ili $u_{11} = 0$, a tada je $1 = \ell_{11}u_{12} = 0$ ili $1 = \ell_{21}u_{11} = 0$.
S druge strane, matrica A reprezentira linearni sustav

$$\begin{array}{rcrcl} 0x_1 & + & x_2 & = & b_1 \\ x_1 & + & x_2 & = & b_2 \end{array}$$

koji uvijek ima rješenje $x_1 = b_2 - b_1$, $x_2 = b_1$, i kojeg možemo ekvivalento zapisati kao[4]

$$\begin{array}{rcrcl} x_1 & + & x_2 & = & b_2 \\ 0x_1 & + & x_2 & = & b_1 \end{array}.$$

Matrica ovog sustava je

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

i očito ima jednostavnu LU faktorizaciju sa $L = I$, $U = A'$. Vezu između A i A' lako opišemo matrično:

$$A' = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{P} \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}}_{A}$$

[4] Zamjena redoslijeda jednadžbi ne mijenja sustav.

Matricu P zovemo matrica permutacije ili jednostavno permutacija. Njeno djelovanje na matricu A je jednostavno permutiranje stupaca.

Da bismo ilustrirali kako zamjenama redaka uvijek možemo dobiti LU faktorizaciju, vratimo se našem 5×5 primjeru i pogledajmo npr. relacije (1.17), (1.18):

$$A^{(2)} \equiv L^{(2)}A^{(1)} = L^{(2)}L^{(1)}A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \end{pmatrix},$$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \dfrac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \dfrac{a_{31}}{a_{11}} & \dfrac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \dfrac{a_{41}}{a_{11}} & \dfrac{a_{42}^{(1)}}{a_{22}^{(1)}} & 0 & 1 & 0 \\ \dfrac{a_{51}}{a_{11}} & \dfrac{a_{52}^{(1)}}{a_{22}^{(1)}} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \end{pmatrix}.$$

Neka je $a_{33}^{(2)} = 0$. Dakle, više ne možemo kao ranije definirati $L^{(3)}$. Pogledajmo elemente $a_{43}^{(2)}$ i $a_{53}^{(2)}$. Ako su obadva jednaki nuli, onda možemo staviti $L^{(3)} = I$ i nastaviti dalje. Jer, cilj transformacije $L^{(3)}$ je poništiti $a_{43}^{(2)}$ i $a_{53}^{(2)}$ – ako su oni već jednaki nuli onda u ovom koraku ne treba ništa raditi pa je transformacija jednaka jediničnoj matrici. Neka je sada npr. $a_{53}^{(2)} \neq 0$. Ako definiramo matricu

$$P^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad \text{onda je} \quad P^{(3)}A^{(2)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} & a_{45}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \end{pmatrix}.$$

Sada možemo definirati matrice

$$L^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{a_{43}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \\ 0 & 0 & -\dfrac{a_{33}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \end{pmatrix}, \quad (L^{(3)})^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{a_{43}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \\ 0 & 0 & \dfrac{a_{33}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \end{pmatrix},$$

i postići

$$A^{(3)} \equiv L^{(3)}P^{(3)}A^{(2)} = L^{(3)}P^{(3)}L^{(2)}L^{(1)}A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & a_{54}^{(3)} & a_{55}^{(3)} \end{pmatrix}.$$

Primijetimo da je treći redak od $A^{(3)}$ jednak petom retku od $A^{(2)}$. Za sljedeći korak eliminacija provjeravamo vrijednost $a_{44}^{(3)}$. Ako je $a_{44}^{(3)} \neq 0$, postupamo kao i ranije, tj. definiramo matricu $L^{(4)}$ kao u relaciji (1.19). Ako je $a_{44}^{(3)} = a_{54}^{(3)} = 0$, onda možemo staviti $L^{(4)} = I$, jer je u tom slučaju $A^{(3)}$ već gornje trokutasta. Neka je $a_{44}^{(3)} = 0$, ali $a_{54}^{(3)} \neq 0$, tako da $L^{(4)}$ nije definirana. Lako provjerimo da permutacijska matrica

$$P^{(4)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{daje} \quad P^{(4)}A^{(3)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{53}^{(2)} & a_{54}^{(2)} & a_{55}^{(2)} \\ 0 & 0 & 0 & a_{54}^{(3)} & a_{55}^{(3)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \end{pmatrix}.$$

Kako je po pretpostavci $a_{44}^{(3)} = 0$, možemo staviti $L^{(4)} = I$ i matrica $U = L^{(4)}P^{(4)}A^{(3)}$ je gornje trokutasta. Sve zajedno, vrijedi relacija

$$U = L^{(4)}P^{(4)}L^{(3)}P^{(3)}L^{(2)}L^{(1)}A.$$

Vidjeli smo ranije da je množenje inverza trokutastih matrica $L^{(k)}$ jednostavno. Međutim, mi sada imamo permutacijske matrice između, pa ostaje istražiti kako one djeluju na strukturu produkta. Primijetimo,

$$P^{(4)}L^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{a_{43}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \\ 0 & 0 & -\frac{a_{33}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{a_{33}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \\ 0 & 0 & -\frac{a_{43}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \end{pmatrix}$$

$$= \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{a_{33}^{(2)}}{a_{53}^{(2)}} & 1 & 0 \\ 0 & 0 & -\frac{a_{43}^{(2)}}{a_{53}^{(2)}} & 0 & 1 \end{pmatrix}}_{\tilde{L}^{(3)}} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} = \tilde{L}^{(3)} P^{(4)}.$$

Dakle, $P^{(4)}$ možemo *prebaciti s lijeve na desnu stranu od* $L^{(3)}$, ako u $L^{(3)}$ ispermutiramo elemente ispod dijagonale u trećem stupcu. Tako dobivena matrica $\tilde{L}^{(3)}$ ima istu strukturu kao i $L^{(3)}$. Na isti način je $P^{(3)}L^{(2)}L^{(1)} = \tilde{L}^{(2)}\tilde{L}^{(1)}P^{(3)}$ i $P^{(4)}\tilde{L}^{(2)}\tilde{L}^{(1)} = \tilde{\tilde{L}}^{(2)}\tilde{\tilde{L}}^{(1)}P^{(4)}$ pa je

$$U = L^{(4)}P^{(4)}L^{(3)}P^{(3)}L^{(2)}L^{(1)}A = L^{(4)}\tilde{L}^{(3)}\tilde{\tilde{L}}^{(2)}\tilde{\tilde{L}}^{(1)}P^{(4)}P^{(3)}A,$$

tj.

$$\underbrace{P^{(4)}P^{(3)}}_{P}A = \underbrace{(L^{(4)})^{-1}(\tilde{L}^{(3)})^{-1}(\tilde{\tilde{L}}^{(2)})^{-1}(\tilde{\tilde{L}}^{(1)})^{-1}}_{L}U.$$

Produkt koji definira matricu L je iste strukture kao i ranije – dakle jednostavno slaganje odgovarajućih elemenata. Nadalje matrica

$$P = P^{(4)}P^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

je opet matrica permutacije. Jasno je kako bi ovaj postupak izgledao općenito. Na kraju eliminacija bi vrijedilo

$$U = A^{(n-1)} = L^{(n-1)}P^{(n-1)}(\cdots(\underbrace{L^{(3)}P^{(3)}\,(\underbrace{L^{(2)}P^{(2)}\,(\underbrace{L^{(1)}P^{(1)}A}_{A^{(1)}})}_{A^{(2)}})}_{A^{(3)}})\cdots), \tag{1.21}$$

i $P = P^{(n-1)}P^{(n-2)}\cdots P^{(2)}P^{(1)}$, gdje neke od permutacija $P^{(k)}$ mogu biti jednake identitetama (jediničnim matricama).
Ilustrirajmo opisanu proceduru jednim numeričkim primjerom.

Primjer 1.3.3 *Neka je*

$$A = \begin{pmatrix} 1 & 1 & 4 & 1 \\ 2 & 1 & 1 & 6 \\ 5 & 1 & 1 & 0 \\ 1 & 4 & 1 & 3 \end{pmatrix}.$$

Najveći element u prvom stupcu od A je na poziciji $(3,1)$ – to znači da prvi pivot maksimiziramo ako zamijenimo prvi i treći redak od A. Tu zamjenu realizira permutacija $P^{(1)}$, gdje je

$$P^{(1)} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad P^{(1)}A = \begin{pmatrix} 5 & 1 & 1 & 0 \\ 2 & 1 & 1 & 6 \\ 1 & 1 & 4 & 1 \\ 1 & 4 & 1 & 3 \end{pmatrix}.$$

Sada definiramo

$$L^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{2}{5} & 1 & 0 & 0 \\ -\frac{1}{5} & 0 & 1 & 0 \\ -\frac{1}{5} & 0 & 0 & 1 \end{pmatrix}, \quad \text{pa je} \quad A^{(1)} = L^{(1)}P^{(1)}A = \begin{pmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{3}{5} & \frac{3}{5} & 6 \\ 0 & \frac{4}{5} & \frac{19}{5} & 1 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \end{pmatrix}.$$

Sljedeći pivot je maksimiziran permutacijom $P^{(2)}$, gdje je

$$P^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad P^{(2)}A^{(1)} = \begin{pmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \\ 0 & \frac{4}{5} & \frac{19}{5} & 1 \\ 0 & \frac{3}{5} & \frac{3}{5} & 6 \end{pmatrix}.$$

Sljedeći korak eliminacija glasi

$$L^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\frac{4}{19} & 1 & 0 \\ 0 & -\frac{3}{19} & 0 & 1 \end{pmatrix}, \quad A^{(2)} = L^{(2)}P^{(2)}A^{(1)} = \begin{pmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \\ 0 & 0 & \frac{69}{19} & \frac{7}{19} \\ 0 & 0 & \frac{9}{19} & \frac{105}{19} \end{pmatrix}.$$

Sljedeća permutacija je identiteta, $P^{(3)} = I$, pa u zadnjem koraku imamo

$$L^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{9}{69} & 1 \end{pmatrix}, \quad A^{(3)} = L^{(3)}P^{(3)}A^{(2)} = \begin{pmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \\ 0 & 0 & \frac{69}{19} & \frac{7}{19} \\ 0 & 0 & 0 & \frac{7182}{1311} \end{pmatrix}.$$

Sada primijetimo da je $A^{(3)} = L^{(3)} I L^{(2)} P^{(2)} L^{(1)} P^{(1)} A$, *gdje je*

$$P^{(2)} L^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{5} & 0 & 0 & 1 \\ -\frac{1}{5} & 0 & 1 & 0 \\ -\frac{2}{5} & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{5} & 1 & 0 & 0 \\ -\frac{1}{5} & 0 & 1 & 0 \\ -\frac{2}{5} & 0 & 0 & 1 \end{pmatrix} P^{(2)} = \tilde{L}^{(1)} P^{(2)}.$$

Dakle, $U \equiv A^{(3)} = L^{(3)} L^{(2)} \tilde{L}^{(1)} P^{(2)} P^{(1)} A$. *Ako stavimo* $P = P^{(2)} P^{(1)}$, *onda vrijedi*

$$\begin{aligned} PA &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 4 & 1 \\ 2 & 1 & 1 & 6 \\ 5 & 1 & 1 & 0 \\ 1 & 4 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 5 & 1 & 1 & 0 \\ 1 & 4 & 1 & 3 \\ 1 & 1 & 4 & 1 \\ 2 & 1 & 1 & 6 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{5} & 1 & 0 & 0 \\ \frac{1}{5} & 0 & 1 & 0 \\ \frac{2}{5} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{4}{19} & 1 & 0 \\ 0 & \frac{3}{19} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{9}{69} & 1 \end{pmatrix} U \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{5} & 1 & 0 & 0 \\ \frac{1}{5} & \frac{4}{19} & 1 & 0 \\ \frac{2}{5} & \frac{3}{19} & \frac{9}{69} & 1 \end{pmatrix} \begin{pmatrix} 5 & 1 & 1 & 0 \\ 0 & \frac{19}{5} & \frac{4}{5} & 3 \\ 0 & 0 & \frac{69}{19} & \frac{7}{19} \\ 0 & 0 & 0 & \frac{7182}{1311} \end{pmatrix}. \end{aligned}$$

Dakle, možemo zaključiti sljedeće:

- ♣ *Za proizvoljnu* $n \times n$ *matricu* A *postoji permutacija* P *tako da Gaussove eliminacije daju* LU *faktorizaciju od* PA, *tj.* $PA = LU$, *gdje je* L *donje trokutasta matrica sa jedinicama na dijagonali, a* U *je gornje trokutasta matrica. Permutaciju* P *možemo odabrati tako da su svi elementi matrice* L *po apsolutnoj vrijednosti najviše jednaki jedinici.*

Preciznije, vrijedi sljedeći teorem :

Teorem 1.3.2 *Neka je* $A \in \mathbf{R}^{n \times n}$ *proizvoljna matrica. Tada postoji permutacija* P *takva da Gaussove eliminacije daju* LU *faktorizaciju* $PA = LU$ *matrice* PA. *Matrica* $L = (\ell_{ij})$ *je donje trokutasta sa jedinicama na dijagonali, a* U *je gornje trokutasta. Pri tome, ako je* P *produkt od* p *inverzija, vrijedi da je* $\det(A) = (-1)^p \prod_{i=1}^{n} u_{ii}$.

Ako su matrice $P^{(k)}$ *odabrane tako da vrijedi*

$$|(P^{(k)} A^{(k-1)})_{kk}| = \max_{k \le j \le n} |(P^{(k)} A^{(k-1)})_{jk}|$$

onda je

$$\max_{1\le k\le n}\max_{1\le i,j\le n}|(L^{(k)})_{ij}| = \max_{1\le i,j\le n}|\ell_{ij}| = 1.$$

U tom slučaju faktorizaciju $PA = LU$ zovemo LU faktorizacijom sa (standardnim) pivotiranjem redaka.

Dokaz: Za početak, primijetimo da za matricu

$$L^{(k)} = \begin{pmatrix} I_k & 0 \\ 0 & v \quad I_{n-k} \end{pmatrix}, \quad v = \begin{pmatrix} \ell_{k+1,k}^{(k)} \\ \vdots \\ \ell_{nk}^{(k)} \end{pmatrix}$$

i permutaciju $\Pi \in \mathcal{S}_n$ oblika

$$\Pi = \begin{pmatrix} I_k & 0 \\ 0 & \hat{\Pi}_{n-k} \end{pmatrix}, \quad \hat{\Pi} \in \mathcal{S}_{n-k}$$

vrijedi

$$\Pi L^{(k)} = \begin{pmatrix} I_k & 0 \\ 0 \quad \hat{\Pi}_{n-k}v & \hat{\Pi}_{n-k} \end{pmatrix} = \begin{pmatrix} I_k & 0 \\ 0 \quad \hat{\Pi}_{n-k}v & I_{n-k} \end{pmatrix} \Pi = \tilde{L}^{(k)}\Pi.$$

Nadalje, svaka permutacija Π oblika

$$\Pi = \begin{pmatrix} I_m & 0 \\ 0 & \hat{\Pi}_{n-m} \end{pmatrix}, \quad m > k, \quad \hat{\Pi} \in \mathcal{S}_{n-m}$$

je trivijalno i oblika

$$\Pi = \begin{pmatrix} I_k & 0 \\ 0 & \tilde{\Pi}_{n-k} \end{pmatrix}, \quad \tilde{\Pi}_{n-k} = \begin{pmatrix} I_{m-k} & 0 \\ 0 & \hat{\Pi}_{n-m} \end{pmatrix} \in \mathcal{S}_{n-k},$$

pa je množenje analogno slučaju $m = k$. Kratko kažemo da "Π prolazi kroz $L^{(k)}$". Nadalje, jasno je da u svakom koraku možemo odrediti permutaciju $P^{(k)}$ tako da postoji donje trokutasta transformacija $L^{(k)}$ sa jedinicama na dijagonali za koju $L^{(k)}P^{(k)}A^{(k-1)}$ ima sve nule ispod dijagonale u k–tom stupcu. Dakle, kao u relaciji (1.21), možemo postići da je $U = A^{(n-1)}$ gornje trokutasta matrica. U produktu

$$U = L^{(n-1)}P^{(n-1)}L^{(n-2)}P^{(n-2)}L^{(n-3)}P^{(n-3)}L^{(n-4)}P^{(n-4)}\dots L^{(2)}P^{(2)}L^{(1)}P^{(1)}A$$

je $P^{(k+1)}$ oblika

$$P^{(k+1)} = \begin{pmatrix} I_k & 0 \\ 0 & \hat{\Pi}_{n-k} \end{pmatrix}, \quad \hat{\Pi} \in \mathcal{S}_{n-k},$$

što znači da $P^{(n-1)}$ prolazi kroz $L^{(n-2)}$, produkt $P^{(n-1)}P^{(n-2)}$ prolazi kroz $L^{(n-3)}$, produkt $P^{(n-1)}P^{(n-2)}P^{(n-3)}$ prolazi kroz $L^{(n-4)}$ itd.
Ako stavimo $P = P^{(n-1)}P^{(n-2)}\cdots P^{(2)}P^{(1)}$, onda je

$$U = \tilde{L}^{(n-1)}\tilde{L}^{(n-2)}\cdots\tilde{L}^{(2)}\tilde{L}^{(1)}PA,$$

odakle kao i ranije dobijemo $PA = LU$. Jasno je da strategija odabira permutacija iz iskaza teorema osigurava da su svi elementi od L po apsolutnoj vrijednosti najviše jednaki jedinici. ■

1.3.5 Vježbe

1.4 Numerička svojstva Gaussovih eliminacija

U prethodnim sekcijama smo se Gaussovim eliminacijama bavili u okvirima linearne algebre. Preciznije, nismo razmatrali praktične detalje realizacije izvedenih algoritama na računalu. Zapravo, termin *praktični detalji* bi trebalo čitati kao **problemi**. Zašto?
Računalo je ograničen, konačan stroj. Imamo ograničenu količinu memorijskog prostora u kojem možemo držati polazne podatke, međurezultate i rezultate računanja.[5] Umjesto skupa realnih brojeva $\mathbf{R}$ imamo njegovu aproksimaciju pomoću konačno mnogo strojnih brojeva (strojni brojevi su zapravo konačan skup razlomaka) što znači da računske operacije ne možemo izvršavati niti točno niti rezultat možemo po volji dobro aproksimirati.
Za one čitatelje koji nisu svladali osnove numeričkih operacija linearne algebre na računalu, kao i za one koji taj materijal žele ponoviti, osnovne činjenice su dane u dodatku u sekciji 1.4.4. Preporučamo da čitatelj svakako baci pogled na tu sekciju prije nastavka čitanja ovog materijala.
Praktično je odvojeno analizirati LU faktorizaciju i rješenje trokutastog sustava. Počinjemo sa LU faktorizacijom, gdje nas očekuje niz zanimljivih zaključaka.

1.4.1 Analiza LU faktorizacije. Važnost pivotiranja.

Prije nego pređemo na numeričku analizu algoritma, pogledajmo kako ga možemo implementirati na računalu s minimalnim korištenjem dodatnog memorijskog prostora.

[5] Svaka operacija zahtijeva izvjesno vrijeme izvršavanja pa je ukupno trajanje algoritma također važan faktor. U ovoj sekciji ćemo prvenstveno analizirati problem točnosti.

Prisjetimo se našeg 5×5 primjera i relacije (1.20):

$$A = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & 0 & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & 1 & 0 & 0 \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 & 0 \\ \frac{a_{51}}{a_{11}} & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & \frac{a_{53}^{(2)}}{a_{33}^{(2)}} & \frac{a_{54}^{(3)}}{a_{44}^{(3)}} & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & 0 & 0 & a_{55}^{(4)} \end{pmatrix}}_{U}.$$

Vidimo da je za spremiti sve elemente matrica L i U dovoljno n^2 varijabli (lokacija u memoriji), dakle onoliko koliko zauzima originalna matrica A. Ako pažljivo pogledamo proces računanja LU faktorizacije, uočavamo da ga možemo izvesti tako da matrica U ostane zapisana u gornjem trokutu matrice A, a strogo donji trokut matrice L bude napisan na mjestu elemenata strogo donjeg trokuta polazne matrice A. Kako matrica L po definiciji ima jedinice na dijagonali, te elemente ne treba nigdje posebno zapisivati. Na taj način se elementi polazne matrice gube, a računanje možemo shvatiti kao promjenu sadržaja polja A koje sadrži matricu A:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \mapsto \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \frac{a_{21}}{a_{11}} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} \\ \frac{a_{41}}{a_{11}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & a_{44}^{(3)} & a_{45}^{(3)} \\ \frac{a_{51}}{a_{11}} & \frac{a_{52}^{(1)}}{a_{22}^{(1)}} & \frac{a_{53}^{(2)}}{a_{33}^{(2)}} & \frac{a_{54}^{(3)}}{a_{44}^{(3)}} & a_{55}^{(4)} \end{pmatrix}.$$

Sve matrice $A^{(k)}$, $k = 1, 2, \ldots, n-1$ su pohranjene u istom $n \times n$ polju koje na početku sadrži matricu $A \equiv A^{(0)}$. Na ovaj način zapis algoritma 1.3.3 postaje još jednostavniji i elegantniji.

Algoritam 1.4.1 *Računanje LU faktorizacije matrice A bez dodatne memorije.*

za $k = 1, \ldots, n-1$ {
    za $j = k+1, \ldots, n$ {

$$
\begin{aligned}
A(j,k) &= \frac{A(j,k)}{A(k,k)} \ ; \} \\
za\ j &= k+1, \ldots, n \ \{ \\
&za\ i = k+1, \ldots, n \ \{ \\
&\qquad A(i,j) = A(i,j) - A(i,k)A(k,j) \ ; \}\}\}
\end{aligned}
$$

Primijetimo da smo koristili oznake uobičajene u programskim jezicima – element matrice (dvodimenzionalnog polja) smo označili s $A(i,j)$. Isto tako, vidimo da konkretna realizacija algoritma na računalu uključuje dodatne trikove i modifikacije kako bi se što racionalnije koristili resursi računala (npr. memorija). Dodatnu pažnju zahtijeva izvođenje aritmetičkih operacija pri čemu ne možemo izbjeći greške zaokruživanja.

Analiza grešaka zaokruživanja je ponekad tehnički kompliciranа. Ono što je važno uočiti je da cilj takve analize nije jednostavno tehničko prebrojavanje svih grešaka zaokruživanja nego izvođenje složenijih i dubljih zaključaka o numeričkoj stabilnosti algoritma i o pouzdanosti korištenja dobivenih rezultata.

Da bismo dobili ideju o kvaliteti izračunate faktorizacije, analizirat ćemo primjer faktorizacije 4×4 matrice

$$
A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}.
$$

Izračunate aproksimacije matrica $L = (\,\ell_{ij}\,)$ i $U = (\,u_{ij}\,)$ ćemo označiti s $\tilde{L} = (\,\tilde{\ell}_{ij}\,)$ i $\tilde{U} = (\,\tilde{u}_{ij}\,)$. Kao u opisu algoritam za računanje LU faktorizacije u sekciji 1.3.3, koristit ćemo matrice $L^{(i)}$ i transformacije oblika $A^{(i)} = L^{(i)}A^{(i-1)}$, $i = 1, \ldots, n-1$. Izračunate aproksimacije označavamo s $\tilde{L}^{(i)}$ i $\tilde{A}^{(i)}$.
Primijetimo da je prvi redak matrice $\tilde{U}$ jednak prvom retku polazne matrice A,

$$
\tilde{U} = \begin{pmatrix} \tilde{u}_{11} & \tilde{u}_{12} & \tilde{u}_{13} & \tilde{u}_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & 0 & \tilde{u}_{33} & \tilde{u}_{34} \\ 0 & 0 & 0 & \tilde{u}_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & 0 & \tilde{u}_{33} & \tilde{u}_{34} \\ 0 & 0 & 0 & \tilde{u}_{44} \end{pmatrix}, \quad \tilde{u}_{1j} = a_{1j}, \quad j = 1, \ldots, 4.
$$

Sada umjesto matrica $L^{(1)}$ i $A^{(1)} = L^{(1)}A$ imamo izračunate matrice

$$
\tilde{L}^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\tilde{\ell}_{21} & 1 & 0 & 0 \\ -\tilde{\ell}_{31} & 0 & 1 & 0 \\ -\tilde{\ell}_{41} & 0 & 0 & 1 \end{pmatrix}
$$

$$\tilde{A}^{(1)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} \ominus \tilde{\ell}_{21} \odot \tilde{u}_{12} & a_{23} \ominus \tilde{\ell}_{21} \odot \tilde{u}_{13} & a_{24} \ominus \tilde{\ell}_{21} \odot \tilde{u}_{14} \\ 0 & a_{32} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{12} & a_{33} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{13} & a_{34} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{14} \\ 0 & a_{42} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{12} & a_{43} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{13} & a_{44} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{14} \end{pmatrix}$$

$$= \begin{pmatrix} \tilde{u}_{11} & \tilde{u}_{12} & \tilde{u}_{13} & \tilde{u}_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & \star & \star & \star \\ 0 & \star & \star & \star \end{pmatrix}, \quad \tilde{u}_{2j} = a_{2j} \ominus \tilde{\ell}_{21} \odot \tilde{u}_{1j}, \quad j = 2, 3, 4.$$

Ovdje smo sa $\star$ označili one elemente koje ćemo mijenjati u sljedećem koraku. Primijetimo da su u prva dva retka matrice $\tilde{A}^{(1)}$ već izračunata prva dva retka matrice $\tilde{U}$. U sljedećem koraku računamo

$$\tilde{L}^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\tilde{\ell}_{32} & 1 & 0 \\ 0 & -\tilde{\ell}_{42} & 0 & 1 \end{pmatrix}$$

$$\tilde{A}^{(2)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & 0 & (a_{33} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{13}) \ominus \tilde{\ell}_{32} \odot \tilde{u}_{23} & (a_{34} \ominus \tilde{\ell}_{31} \odot \tilde{u}_{14}) \ominus \tilde{\ell}_{32} \odot \tilde{u}_{24} \\ 0 & 0 & (a_{43} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{14}) \ominus \tilde{\ell}_{42} \odot \tilde{u}_{23} & (a_{44} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{14}) \ominus \tilde{\ell}_{42} \odot \tilde{u}_{24} \end{pmatrix}$$

$$= \begin{pmatrix} \tilde{u}_{11} & \tilde{u}_{12} & \tilde{u}_{13} & \tilde{u}_{14} \\ 0 & \tilde{u}_{22} & \tilde{u}_{23} & \tilde{u}_{24} \\ 0 & 0 & \tilde{u}_{33} & \tilde{u}_{34} \\ 0 & 0 & \star & \star \end{pmatrix}, \quad \tilde{u}_{3j} = (a_{3j} \ominus \tilde{\ell}_{31}\tilde{u}_{1j}) \ominus \tilde{\ell}_{32} \odot \tilde{u}_{2j}, \quad j = 3, 4.$$

I, u zadnjem koraku je ostala transformacija

$$\tilde{L}^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\tilde{\ell}_{43} & 1 \end{pmatrix},$$

koja primjenom na $\tilde{A}^{(2)}$ daje i preostali element matrice $\tilde{U}$,

$$\tilde{u}_{44} = ((a_{44} \ominus \tilde{\ell}_{41} \odot \tilde{u}_{14}) \ominus \tilde{\ell}_{42} \odot \tilde{u}_{24}) \ominus \tilde{\ell}_{43} \odot \tilde{u}_{34}.$$

Uočavamo da se elementi u_{ij} računaju prema formuli

$$u_{ij} = a_{ij} - \sum_{m=1}^{i-1} \ell_{im} u_{mj}, \quad 2 \leq i \leq n, \quad i \leq j \leq n,$$

pri čemu je $u_{1j} = a_{1j}$, $1 \leq j \leq n$. Ovu formulu je lako provjeriti raspisivanjem produkta $A = LU$ po elementima. U našem algoritmu, zbog grešaka zaokruživanja, vrijedi

$$\tilde{u}_{ij} = (\cdots((a_{ij} \ominus \tilde{\ell}_{i1} \odot \tilde{u}_{1j}) \ominus \tilde{\ell}_{i2} \odot \tilde{u}_{2j}) \ominus \cdots) \ominus \tilde{\ell}_{i,i-1} \odot \tilde{u}_{i-1,j}. \tag{1.22}$$

Formula (1.22) je samo specijalan slučaj računanja općenitog izraza oblika

$$s = v_1 w_1 \pm v_2 w_2 \pm v_3 w_3 \pm \cdots \pm v_p w_p,$$

pri čemu se koristi algoritam

$\tilde{u}_{ij} = a_{ij}$;
za $m = 1, \ldots, i-1$ {
$\quad \tilde{u}_{ij} = \tilde{u}_{ij} \ominus \tilde{\ell}_{im} \odot \tilde{u}_{mj}$; }

Koristeći Propoziciju 1.4.3, zaključujemo da postoje ξ_{ij}, ζ_{ijm} tako da je u (1.22)

$$\tilde{u}_{ij} = a_{ij}(1 + \xi_{ij}) - \sum_{m=1}^{i-1} \tilde{\ell}_{im}\tilde{u}_{mj}(1 + \zeta_{ijm}). \tag{1.23}$$

Pri tome je za sve i, j, m

$$|\xi_{ij}|,\ |\zeta_{ijm}| \leq \frac{n\boldsymbol{\varepsilon}}{1 - n\boldsymbol{\varepsilon}}.$$

Relaciju (1.23) možemo pročitati i kao

$$a_{ij} = \sum_{m=1}^{i} \tilde{\ell}_{im}\tilde{u}_{mj} + \delta a_{ij},\quad \delta a_{ij} = \sum_{m=1}^{i} \tilde{\ell}_{im}\tilde{u}_{mj}\zeta_{ijm} - \xi_{ij}a_{ij}, \tag{1.24}$$

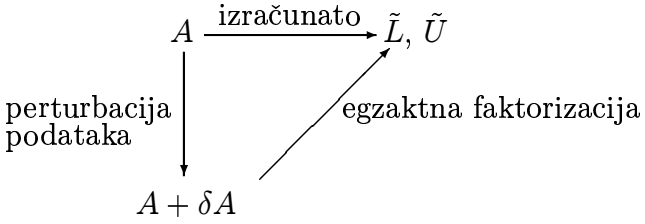
gdje smo $\tilde{u}_{ij}$ napisali kao $\tilde{\ell}_{ii}\tilde{u}_{ij}(1 + \zeta_{iji})$, uz $\tilde{\ell}_{ii} = 1$ i $\zeta_{iji} = 0$.
Time smo dokazali sljedeći teorem.

Teorem 1.4.1 *Neka je algoritam 1.4.1 primijenjen na matricu $A \in \mathbf{R}^{n\times n}$ i neka su uspješno izvršene sve njegove operacije. Ako su $\tilde{L}$ i $\tilde{U}$ izračunati trokutasti faktori, onda je*

$$\tilde{L}\tilde{U} = A + \delta A,\ \ |\delta A| \leq \frac{n\boldsymbol{\varepsilon}}{1 - n\boldsymbol{\varepsilon}}(|A| + |\tilde{L}||\tilde{U}|) \leq \frac{2n\boldsymbol{\varepsilon}}{1 - 2n\boldsymbol{\varepsilon}}|\tilde{L}||\tilde{U}|,$$

gdje prva nejednakost vrijedi za $n\boldsymbol{\varepsilon} < 1$, a druga za $2n\boldsymbol{\varepsilon} < 1$.

Komentar 1.4.1 *Rezultat teorema zaslužuje poseban komentar. Naša analiza nije dala odgovor na pitanje koliko su $\tilde{L}$ i $\tilde{U}$ daleko od točnih matrica L i U. Umjesto toga, dobili smo zaključak da $\tilde{L}$ i $\tilde{U}$ čine egzaktnu LU faktorizaciju matrice $A+\delta A$. Drugim riječima, ako bismo A promijenili u $A+\delta A$ i zatim uzeli egzaktnu faktorizaciju, dobili bismo upravo $\tilde{L}$ i $\tilde{U}$. Ovu situaciju možemo ilustrirati komutativnim dijagramom na slici 1.2. Dobiveni rezultat je u praksi od izuzetne važnosti. Jer, često je u primjenama nemoguće raditi sa egzaktnim podacima – matrica A može biti rezultat mjerenja ili nekih prethodnih proračuna, dakle netočna. Ako je egzaktna (nepoznata) matrica $\hat{A}$ i $A = \hat{A} + \delta\hat{A}$, onda je $\tilde{L}\tilde{U} = \hat{A} + \delta\hat{A} + \delta A$ i $LU = \hat{A} + \delta\hat{A}$. Ako su δA i $\delta\hat{A}$ usporedivi po veličini, onda možemo u mnogim primjenama $\tilde{L}$ i $\tilde{U}$ smatrati jednako dobrim kao i L i U.*



Slika 1.2: *Komutativni dijagram LU faktorizacije u aritmetici konačne preciznosti. Izračunati rezultat je ekvivalentan egzaktnom računu sa promijenjenim polaznim podacima.*

Iz prethodne analize je jasno da je δA mala ako produkt $|\tilde{L}||\tilde{U}|$ nije prevelik u usporedbi s $|A|$. To na žalost nije osigurano u LU faktorizaciji. Sljedeći primjer pokazuje numeričku nestabilnost algoritma.

Primjer 1.4.1 *Neka je α mali parametar, $|\alpha| \ll 1$, i neka je matrica A definirana s*

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

U egzaktnom računanju imamo

$$L^{(2,1)} = \begin{pmatrix} 1 & 0 \\ -\dfrac{1}{\alpha} & 1 \end{pmatrix}, \quad L^{(2,1)}A = \begin{pmatrix} \alpha & 1 \\ 0 & 1 - \dfrac{1}{\alpha} \end{pmatrix},$$

pa je LU faktorizacija matrice A dana s

$$\underbrace{\begin{pmatrix} \alpha & 1 \\ 1 & 1 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 \\ \dfrac{1}{\alpha} & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} \alpha & 1 \\ 0 & 1 - \dfrac{1}{\alpha} \end{pmatrix}}_{U}.$$

Pretpostavimo sada da ovaj račun provodimo na računalu u aritmetici sa 8 decimalnih znamenki, tj. sa točnosti $\varepsilon \approx 10^{-8}$. Neka je $|\alpha| < \varepsilon$, npr. neka je $\alpha = 10^{-10}$. Kako je problem jednostavan, vrijedi $\tilde{\ell}_{21} = \ell_{21}(1+\epsilon_1)$, $|\epsilon_1| \leq \varepsilon$, $\tilde{u}_{11} = u_{11}$, $\tilde{u}_{12} = u_{12}$ i

$$\tilde{u}_{22} = 1 \ominus 1 \oslash \alpha = -1 \oslash \alpha = -\frac{1}{\alpha}(1+\epsilon_1).$$

Primijetimo da je

$$\left|\frac{\tilde{u}_{22} - u_{22}}{u_{22}}\right| \leq \frac{2\varepsilon}{1-\varepsilon}.$$

Dakle svi elementi matrica $\tilde{L}$ i $\tilde{U}$ su izračunati sa malom relativnom pogreškom. Sjetimo se da je ovaj primjer najavljen kao primjer numeričke nestabilnosti procesa eliminacija, odnosno LU faktorizacije. Gdje je tu nestabilnost ako su svi izračunati elementi matrica $\tilde{L}$ i $\tilde{U}$ gotovo jednaki točnim vrijednostima? Odstupanje (relativna greška) je najviše reda veličine dvije greške zaokruživanja – gdje je onda problem? Izračunajmo (egzaktno) $\tilde{L}\tilde{U}$:

$$\tilde{L}\tilde{U} = \begin{pmatrix} 1 & 0 \\ 1 \oslash \alpha & 1 \end{pmatrix} \begin{pmatrix} \alpha & 1 \\ 0 & -1 \oslash \alpha \end{pmatrix} = \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha & 1 \\ 1 & 1 \end{pmatrix}}_{A} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}}_{\delta A}.$$

Primijetimo da δA ne možemo smatrati malom perturbacijom polazne matrice A – jedan od najvećih elemenata u matrici A, $a_{22} = 1$, je promijenjen u nulu. Ako bismo koristeći $\tilde{L}$ i $\tilde{U}$ pokušali riješiti linearni sustav $Ax = b$, zapravo bismo radili na sustavu $(A + \delta A)x = b$. Teka da dobijemo osjećaj kako katastrofalno loš rezultat možemo dobiti, pogledajmo linearne sustave

$$\begin{pmatrix} \alpha & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Njihova rješenja su

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \dfrac{-1}{\alpha - 1} \\ \dfrac{2\alpha - 1}{\alpha - 1} \end{pmatrix} \approx \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 - 2\alpha \end{pmatrix}.$$

Vidimo da se x_1 i $\tilde{x}_1$ potpuno razlikuju. Zaključujemo da Gaussove eliminacije mogu biti numerički nestabilne – dovoljna je jedna greška zaokruživanja " u krivo vrijeme na krivom mjestu" pa da dobiveni rezultat bude potpuno netočan.

Komentar 1.4.2 *I ovaj primjer zaslužuje komentar. Vidimo da katastrofalno velika greška nije uzrokovana akumuliranjem velikog broja grešaka zaokruživanja. Cijeli*

problem je u samo jednoj aritmetičkoj operaciji (pri računanju $\tilde{u}_{22}$) koja je zapravo izvedena jako točno, sa malom greškom zaokruživanja. Cilj numeričke analize algoritma je da otkrije moguće uzroke nestabilnosti, objasni fenomene vezane za numeričku nestabilnost i ponudi rješenja za njihovo uklanjanje.
Primijetimo da je nestabilnost ilustrirana u primjeru u skladu sa teoremom 1.4.1. Naime, ako izračunamo $|\tilde{L}||\tilde{U}|$ dobijemo

$$|\tilde{L}||\tilde{U}| = \begin{pmatrix} |\alpha| & 1 \\ 1+\epsilon & 2|1 \oslash \alpha| \end{pmatrix},$$

gdje je $1+\epsilon = \alpha(1 \oslash \alpha)$, $|\epsilon| \leq \boldsymbol{\varepsilon}$. Kako je na poziciji $(2,2)$ u matrici $|\tilde{L}||\tilde{U}|$ element koji je reda veličine $1/|\alpha| > 1/\boldsymbol{\varepsilon}$, vidimo da nam teorem ne može garantirati mali δA.

Jasno nam je da je, zbog nenegativnosti matrica $|\tilde{L}|$ i $|\tilde{U}|$, mali produkt $|\tilde{L}||\tilde{U}|$ moguć samo ako su elementi od $\tilde{L}$ i $\tilde{U}$ mali po apsolutnoj vrijednosti. Pogledajmo nastavak primjera 1.4.1.

Primjer 1.4.2 *Neka je A matrica iz primjera 1.4.1. Zamijenimo joj poredak redaka,*

$$A' = PA = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \alpha & 1 \end{pmatrix}.$$

LU faktorizacija matrice $A' = LU$ je

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

Ako je $|\alpha| < \boldsymbol{\varepsilon}$, onda su izračunate matrice

$$\tilde{L} = \begin{pmatrix} 1 & 0 \\ \alpha & 1 \end{pmatrix}, \quad \tilde{U} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},$$

i vrijedi

$$\tilde{L}\tilde{U} = \begin{pmatrix} 1 & 1 \\ \alpha & 1+\alpha \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 \\ \alpha & 1 \end{pmatrix}}_{A'} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & \alpha \end{pmatrix}}_{\delta A'}, \quad |\delta A'| \leq \boldsymbol{\varepsilon}|A|.$$

Primijetimo i da je produkt

$$|\tilde{L}||\tilde{U}| = \begin{pmatrix} 1 & 1 \\ |\alpha| & 1+\alpha \end{pmatrix}$$

po elementima istog reda veličine kao i $|A'|$. Dakle, u ovom primjeru je bilo dovoljno zamijeniti poredak redaka u A (redoslijed jednadžbi) pa da imamo garantirano dobru faktorizaciju u smislu da je $\tilde{L}\tilde{U} = A' + \delta A'$ sa malom perturbacijom $\delta A'$.

Iz prethodnih primjera i diskusija je jasno da standardno pivotiranje redaka, koje osigurava da su u matrici L svi elementi po apsolutnoj vrijednosti najviše jednaki jedinici[6], doprinosi numeričkoj stabilnosti. Naime, lako se vidi sa su tada i svi elementi

[6] Vidi teorem 1.3.2.

matrice $|\tilde{L}|$ manji ili jednaki od jedan. U tom slučaju veličina produkta $|\tilde{L}||\tilde{U}|$ bitno ovisi o elementima matrice $\tilde{U}$. S druge strane, elementi matrice $\tilde{U}$ su dobiveni iz matrica $\tilde{A}^{(k)}$, $k = 0, 1, \ldots, n-1$, pa je broj

$$\rho = \frac{\max\limits_{i,j,k} \tilde{a}_{ij}^{(k)}}{\max\limits_{ij} a_{ij}} \tag{1.25}$$

dobra mjera za relativni rast (u odnosu na A) elemenata u produktu $|\tilde{L}||\tilde{U}|$. Broj ρ zovemo faktor rasta elemenata u LU faktorizaciji i definiran je bez obzira da li koristimo pivotiranje redaka. Primijetimo da u analizi grešaka zaokruživanja pivotiranje ne predstavlja dodatnu tehničku poteškoću, pa odmah možemo iskazati sljedeći teorem.

Teorem 1.4.2 *Neka je LU faktorizacija $n \times n$ matrice A izračunata sa pivotiranjem redaka u aritmetici sa relativnom točnosti ε i neka su $\tilde{L}$ i $\tilde{U}$ dobivene aproksimacije za L i U. Ako je pri tome korištena permutacija P, onda je*

$$\tilde{L}\tilde{U} = P(A + \delta A), \quad |\delta A| \leq \frac{2n\varepsilon}{1 - 2n\varepsilon} P^\tau |\tilde{L}||\tilde{U}|.$$

Specijalno je, bez obzira na pivotiranje,

$$\|\delta A\|_F \leq O(n^3)\varepsilon\rho\|A\|_F.$$

Dokaz: Nakon ponovnog čitanja dokaza teorema 1.3.2 bi trebalo biti jasno da permutacije prolaze kroz elementarne transformacije $\tilde{L}^{(i)}$ neovisno o točnosti računanja (egzaktno ili do na greške zaokruživanja). Dakle, možemo zaključiti da čak i računanje faktorizacije s pivotiranjem na stroju odgovara računu bez pivotiranja ali sa polaznom matricom $A' = PA$. Sada primjenom teorema 1.4.1 dobijemo da vrijedi

$$\tilde{L}\tilde{U} = A' + \delta A', \quad |\delta A'| \leq \frac{2n\varepsilon}{1 - 2n\varepsilon} |\tilde{L}||\tilde{U}|.$$

Kako je $A' + \delta A' = P(A + P^\tau \delta A')$, stavljanjem $\delta A = P^\tau \delta A'$ dobivamo tvrdnju teorema. Primijetimo i da je, bez obzira da li pivotiramo retke ili ne,

$$\|\delta A\|_F \leq \frac{2n\varepsilon}{1 - 2n\varepsilon} \sqrt{\frac{n(n+1)}{2}} \sqrt{\frac{n(n+1)}{2}} \rho \|A\|_F.$$

■

Razlika u numeričkoj stabilnosti koju donosi pivotiranje redaka je bolje ponašanje parmetra ρ, tj. pivotiranjem možemo osigurati umjeren rast elemenata u toku LU faktorizacije. U primjeru 1.4.1 smo vidjeli da u LU faktorizaciji bez pivotiranja rast elemenata tokom faktorizacije može biti po volji veliki. Sljedeća propozicija pokazuje da u slučaju pivotiranja redaka faktor ρ ima gornju ogradu koja je funkcija samo dimenzije problema.

Propozicija 1.4.1 *Ako LU faktorizaciju računamo s pivotiranjem redaka u aritmetici sa maksimalnom greškom zaokruživanja ε, onda je*

$$\rho \leq 2^{n-1}(1+\varepsilon)^{2(n-1)}.$$

Specijalno je u slučaju egzaktnog računanja $\rho \leq 2^{n-1}$.

Dokaz: Dokaz ostavljamo čitatelju za vježbu. ■

Primijetimo da je gornja ograda za ρ reda veličine 2^n, što brzo raste kao funkcija od n. Postoje primjeri na kojima se ta gornja ograda i dostiže. Ipak, iskustvo iz prakse govori da su takvi primjeri rijetki i da je LU faktorizacija sa pivotiranjem redaka dobar algoritam za rješavanje sustava linearnih jednadžbi. Možemo zaključiti i preporučiti sljedeće:

- ♣ *Gaussove eliminacije, odnosno LU faktorizaciju, valja u praksi uvijek raditi sa pivotiranjem redaka.*

1.4.2 Analiza numeričkog rješenja trokutastog sustava

Kako smo vidjeli u sekciji 1.3.2, trokutaste sustave rješavamo jednostavnim i elegantnim supstitucijama naprijed ili unazad. Ta jednostavnost se odražava i na dobra numerička svojstva supstitucija, kada ih provedemo na računalu. Sljedeća propozicija opisuje kvalitetu numerički izračunatog rješenja trokutastog sustava jednadžbi.

Propozicija 1.4.2 *Neka je T donje (gornje) trokutasta matrica reda n i neka je sustav $Tv = d$ riješen supstitucijama naprijed (unazad) kako je opisano u sekciji 1.3.2. Ako je $\tilde{v}$ rješenje dobiveno primjenom strojne aritmetike preciznosti ε, onda postoji donje (gornje) trokutasta matrica δT tako da vrijedi*

$$(T+\delta T)\tilde{v} = d, \quad |\delta T| \leq \eta_{\triangleright}|T|, \quad 0 \leq \eta_{\triangleright} \leq \frac{n\varepsilon}{1-n\varepsilon}.$$

Dokaz: Dokaz zbog jednostavnosti provodimo samo za donje trokutastu matricu T. Pretpostavljamo da se i–ta komponenta rješenja za $i > 1$ računa na sljedeći način:

$\tilde{v}_i = T_{i1} \odot \tilde{v}_1$;
za $j = 2, \ldots, i-1$ {
$\quad \tilde{v}_i = \tilde{v}_i \oplus T_{ij} \odot \tilde{v}_j$; }
$\tilde{v}_i = (d_i \ominus \tilde{v}_i) \oslash T_{ii}$.

Primjenom pravila strojne aritmetike dobijemo[7] $\tilde{v}_1 = (d_1/T_{11})(1+\epsilon_1)$, $|\epsilon_1| \leq \varepsilon$, te za $i = 2, \ldots, n$

$$\tilde{v}_i = \frac{d_i - \sum_{j=1}^{i-1} T_{ij}(1+\zeta_j)\tilde{v}_j}{\frac{T_{ii}}{(1+\epsilon_{1,i})(1+\epsilon_{2,i})}}, \quad |\zeta_j| \leq \frac{(i-1)\varepsilon}{1-(i-1)\varepsilon}, \quad |\epsilon_{1,i}| \leq \varepsilon, \quad |\epsilon_{2,i}| \leq \varepsilon.$$

■

Komentar 1.4.3 *Koliko god da je prethodni rezultat tehnički jednostavan, valja naglasiti da je zaključak o točnosti rješenja trokutastog sustava važan: izračunato rješenje zadovoljava trokutasti sustav sa matricom koeficijenata koja se po elementima malo razlikuje od zadane. Pojednostavljeno govoreći, ako radimo sa $\varepsilon \approx 10^{-8}$ i ako je $n = 1000$, onda izračunati vektor $\tilde{v}$ zadovoljava $\tilde{T}\tilde{v} = d$, gdje se elemeni od $\tilde{T}$ i T poklapaju u barem 5 decimalnih znamenki (od 8 na koliko je zadana matrica T).*

1.4.3 Točnost izračunatog rješenja sustava

Sada nam ostaje napraviti kompoziciju dobivenih rezultata i ocijeniti koliko točno možemo na računalu riješiti linearni sustav $Ax = b$ u kojem smo izračuali LU faktorizaciju $PA = LU$ i supstitucijama naprijed i unazad izračunali $x = U^{-1}(L^{-1}(Pb))$. Kako smo vidjeli u prethodnim sekcijama permutacija P se može (za potrebe analize) odmah primjeniti na polazne podatke i numeričku analizu možemo provesti bez pivotiranja. Kako to pojednostavljuje oznake, mi ćemo pretpostaviti da su na polazne podatke A i b već primijenjene zamjene redaka, tako da su formule jednostavno $A = LU$ i $x = U^{-1}(L^{-1}b)$).
Neka su $\tilde{L}$ i $\tilde{U}$ izračunate trokutaste matrice, gdje je $\tilde{L}\tilde{U} = A + \delta A$, kao u teoremu 1.4.1. Izračunato rješenje $\tilde{y}$ sustava $\tilde{L}y = b$ zadovoljava (prema propoziciji 1.4.2)

$$(\tilde{L} + \delta\tilde{L})\tilde{y} = b, \quad |\delta\tilde{L}| \leq \frac{n\varepsilon}{1-n\varepsilon}|\tilde{L}|.$$

Na isti način rješenje $\tilde{x}$ sustava $\tilde{U}x = \tilde{y}$ zadovoljava

$$(\tilde{U} + \delta\tilde{U})\tilde{x} = \tilde{y}, \quad |\delta\tilde{U}| \leq \frac{n\varepsilon}{1-n\varepsilon}|\tilde{U}|.$$

Dakle, $(\tilde{L} + \delta\tilde{L})(\tilde{U} + \delta\tilde{U})\tilde{x} = b$, tj.

$$\begin{aligned} (A + \delta A + E)\tilde{x} = b, \quad E &= \tilde{L}\delta\tilde{U} + \delta\tilde{L}\tilde{U} + \delta\tilde{L}\delta\tilde{U}, \\ |E| &\leq |\tilde{L}||\delta\tilde{U}| + |\delta\tilde{L}||\tilde{U}| + |\delta\tilde{L}||\delta\tilde{U}|. \end{aligned}$$

Time smo dokazali sljedeći teorem.

[7] Vidi sekciju 1.4.4.

Teorem 1.4.3 *Neka je $\tilde{x}$ rješenje regularnog $n \times n$ sustava jednadžbi $Ax = b$, dobiveno Gaussovim eliminacijama sa pivotiranjem redaka. Tada postoji perturbacija ΔA za koju vrijedi*

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

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

Na kraju ove sekcije, pokažimo kako cijeli algoritam na računalu možemo implementirati bez dodatne memorije. Kako smo prije vidjeli, LU faktorizaciju možemo napraviti tako da L i U smjestimo u matricu A. Sada još primijetimo da sustave $Ly = b$ i $Ux = y$ možemo riješiti tako da y i x u memoriju zapisujemo na mjesto vektora b. Tako dobijemo sljedeću implementaciju Gaussovih eliminacija:

Algoritam 1.4.2 *Rješavanje trokutastog sustava jednadžbi $Ax = b$ Gaussovim eliminacijama bez dodatne memorije.*

/ LU faktorizacija, $A = LU$ */*
za $k = 1, \ldots, n-1$ {
    za $j = k+1, \ldots, n$ {
        $A(j,k) = \dfrac{A(j,k)}{A(k,k)}$; }
    za $j = k+1, \ldots, n$ {
        za $i = k+1, \ldots, n$ {
            $A(i,j) = A(i,j) - A(i,k)A(k,j)$; }}}
/ Rješavanje sustava $Ly = b$, y napisan na mjesto b. */*
za $i = 2, \ldots, n$ {
    za $j = 1, \ldots, i-1$ {
        $b(i) = b(i) - A(i,j)b(j)$; } }
*/*Rješavanje sustava $Ux = y$, x napisan na mjesto b.*/*
$b(n) = b(n)/A(n,n)$;
za $i = n-1, \ldots, 1$ {
    za $j = i+1, \ldots, n$ {
        $b(i) = b(i) - A(i,j)b(j)$; }
    $b(i) = b(i)/A(i,i)$; }

1.4.4 Dodatak: Osnove matričnog računa na računalu

Na računalu općenito ne možemo egzaktno izvršavati aritmetičke operacije. Rezultat zbrajanja, oduzimanja, množenja ili dijeljenja dva strojna broja x i y je po definiciji strojni broj koji je najbliži egzaktnom zbroju, razlici, umnošku, odnosno kvocijentu

x i y. Pri tome je relativna greška tako izvedenih operacija manja ili jednaka polovini najvećeg relativnog razmaka dva susjedna strojna broja. Na primjer, u standardnoj jednostrukoj preciznosti (32–bitna reprezentacija) je relativni razmak susjednih brojeva omeđen s 2^{-23} pa je relativna greška aritmetičkih operacija najviše $\boldsymbol{\varepsilon} \approx 10^{-8}$. Navedena pravila za izvršavanje elementarnih aritmetičkih operacija lako zapišemo na sljedeći način:

$$\begin{aligned}
\text{zbrajanje: } x \oplus y &= (x+y)(1+\epsilon_1), \quad |\epsilon_1| \leq \boldsymbol{\varepsilon} \\
\text{oduzimanje: } x \ominus y &= (x-y)(1+\epsilon_2), \quad |\epsilon_2| \leq \boldsymbol{\varepsilon} \\
\text{množenje: } x \odot y &= xy(1+\epsilon_3), \quad |\epsilon_3| \leq \boldsymbol{\varepsilon} \\
\text{dijeljenje: } x \oslash y &= \frac{x}{y}(1+\epsilon_4), \quad |\epsilon_4| \leq \boldsymbol{\varepsilon}, \quad y \neq 0.
\end{aligned}$$

Ove relacije vrijede ako su rezultati navedenih operacija po apsolutnoj vrijednosti u inervalu (μ, M) gdje je npr. u 32–bitnoj reprezentaciji $\mu = 2^{-126} \approx 10^{-38}$ najmanji a $M = (1+2^{-1}+\cdots+2^{-23})2^{127} \approx 10^{38}$ najveći normalizirani strojni broj. (U dvostrukoj preciznosti (64–bitna reprezentacija brojeva) je $\mu \approx 10^{-308}$, $M \approx 10^{308}$.) Analiza za rezultate izvan intervala (μ, M) je nešto složenija pa je nećemo raditi.
Kako na računalu izgledaju osnovne operacije linearne algebre? Lako se uvjerimo da je većina operacija (skalarni produkt, norma, linearne kombinacije, matrične operacije) bazirana na računanju

$$s = \sum_{i=1}^{m} x_i y_i,$$

gdje su x_i, y_i skalari (realni ili kompleksni brojevi ili njihove aproksimacije na računalu). Ako s računamo na standardan način, u računalu dobijemo, npr. sa $m = 4$, izraz oblika

$$\tilde{s} = ((((x_1 \odot y_1) \oplus x_2 \odot y_2) \oplus x_3 \odot y_3) \oplus x_4 \odot y_4).$$

Sustavnom primjenom osnovnih svojstava aritmetike na stroju, lako se provjeri da je

$$\begin{aligned}
\tilde{s} &= (((x_1y_1(1+\epsilon_1) + x_2y_2(1+\epsilon_2))(1+\xi_2) + x_3y_3(1+\epsilon_3))(1+\xi_3) \\
&\quad + x_4y_4(1+\epsilon_4))(1+\xi_4) \\
&= x_1y_1 \underbrace{(1+\epsilon_1)(1+\xi_2)(1+\xi_3)(1+\xi_4)}_{1+\zeta_1} + x_2y_2 \underbrace{(1+\epsilon_2)(1+\xi_2)(1+\xi_3)(1+\xi_4)}_{1+\zeta_2} \\
&\quad + x_3y_3 \underbrace{(1+\epsilon_3)(1+\xi_3)(1+\xi_4)}_{1+\zeta_3} + x_4y_4 \underbrace{(1+\epsilon_4)(1+\xi_4)}_{1+\zeta_4} = \sum_{i=1}^{m=4} x_iy_i(1+\zeta_i),
\end{aligned}$$

gdje su sve vrijednosti ϵ_i, ξ_i po modulu manje od $\boldsymbol{\varepsilon}$. Sada je jasno kako bi izgledala formula za proizvoljan broj od m sumanada. Primijetimo da $1+\zeta_k$ možemo ocijeniti

s

$$1 - m\varepsilon \leq 1 + \zeta_k \leq \frac{1}{1 - m\varepsilon}, \text{ tj. vrijedi } |\zeta_k| \leq \frac{m\varepsilon}{1 - m\varepsilon}, \quad k = 1, 2, \ldots, m.$$

Propozicija 1.4.3 *Neka su* $x_1, \ldots, x_m$, $y_1, \ldots, y_m$ *brojevi u računalu,* $m \geq 1$. *Ako vrijednost* $s = \sum_{i=1}^{m} x_i y_i$ *računamo kao*

$\tilde{s} = x_1 \odot y_1$;
za $i = 2, \ldots, m$ {
$\quad \tilde{s} = \tilde{s} \oplus x_i \odot y_i$; }

onda postoje brojevi ζ_i, $i = 1, \ldots, m$, *tako da vrijedi*

$$\tilde{s} = \sum_{i=1}^{m} x_i y_i (1 + \zeta_i), \quad |\zeta_i| \leq \frac{m\varepsilon}{1 - m\varepsilon}, \quad i = 1, 2, \ldots, m.$$

Dokaz: Dokaz smo već skicirali na primjeru $m = 4$. Očito je formalni dokaz najlakše izvesti matematičkom indukcijom po m. Dovoljno je primijetiti da je u koraku indukcije

$$\begin{aligned} \tilde{s} \oplus x_{m+1} \odot y_{m+1} &= (\tilde{s} + x_{m+1} y_{m+1}(1 + \omega_1))(1 + \omega_2) \\ &= \tilde{s}(1 + \omega_2) + x_{m+1} y_{m+1}(1 + \omega_1)(1 + \omega_2), \quad |\omega_1| \leq \varepsilon, \quad |\omega_2| \leq \varepsilon, \end{aligned}$$

te da je $1 - (m+1)\varepsilon \leq (1 - \varepsilon)(1 - m\varepsilon)$ i

$$\frac{1 + \varepsilon}{1 - m\varepsilon} \leq 1 + \frac{(m+1)\varepsilon}{1 - (m+1)\varepsilon}.$$

■

1.4.5 Vježbe

1.5 Numeričko rješavanje simetričnih sustava jednadžbi

U mnogim važnim primjenama, posebno u inženjerskim znanostima, je linearni sustav jednadžbi *simetričan*. To znači da je u sustavu $Ax = b$ matrica $A = (a_{ij})_{i,j=1}^{n}$ simetrična, $A = A^\tau$, tj. za sve i, j je $a_{ij} = a_{ji}$.

Primjer 1.5.1 *U pripremi.*

Primjer 1.5.2 *U pripremi.*

Naravno, ako pametno iskoristimo simetriju, $A = A^\tau$, onda Gaussove eliminacije možemo provesti puno efikasnije. Pokazuje se da u slučaju simetrične matrice LU faktorizacija ima općeniti oblik $A = R^\tau JR$, gdje je R gornje trokutasta matrica, a $J = \mathrm{diag}(\pm 1)$. Da bismo dobili ideju zašto je to tako, pogledajmo LU faktorizaciju $A = LU$ simetrične regularne matrice A. Iz $A = LU$ slijedi da je $U^{-\tau}AU^{-1} = U^{-\tau}L$ istovremeno simetrična i gornje trokutasta matrica. Dakle, $D = U^{-\tau}L$ je dijagonalna matrica, pa je $L = U^\tau D$, tj. $A = U^\tau DU$. Ako je $D = \mathrm{diag}(d_i)_{i=1}^n$, onda definiramo $|D|^{1/2} = \mathrm{diag}(|d_i|^{1/2})$, $J = \mathrm{diag}(\mathrm{sign}(d_i))$ i $R = |D|^{1/2}U$. Slijedi da je $A = R^\tau JR$.
Postojanje LU faktorizacije je uvijek osigurano ako pivotiramo. Kako nam je cilj sačuvati simetriju, kod simetrične matrice ćemo istovremeno permutirati retke i stupce, tj. radit ćemo s PAP^τ, gdje je P matrica permutacije.

1.5.1 Pozitivno definitni sustavi. Faktorizacija Choleskog

Kažemo da je simetrična $n \times n$ matrica A *pozitivno definitna* ako za sve $x \in \mathbf{R}^n$, $x \neq 0$, vrijedi

$$x^\tau Ax > 0.$$

Ako npr. uzmemo $x = e_i$, i–ti stupac jedinične matrice, onda je $a_{ii} = e_i^\tau Ae_i > 0$, tj. dijagonalni elementi pozitivno definitne matrice su uvijek pozitivni. Nadalje, ako je S bilo koja pozitivno definitna matrica i $x \neq 0$, onda je i $y = Sx \neq 0$ i vrijedi

$$x^\tau(S^\tau AS)x = (Sx)^\tau A(Sx) = y^\tau Sy > 0,$$

pa zaključujemo da je i $S^\tau AS$ pozitivno definitna matrica.
Pozitivna definitnost osigurava egzistenciju LU faktorizacije bez pivotiranja. Pogledajmo kako. Ako A particioniramo s

$$A = \begin{pmatrix} a_{11} & a^\tau \\ a & \hat{A} \end{pmatrix}, \quad \hat{A} \in \mathbf{R}^{(n-1)\times(n-1)},$$

onda je $a_{11} > 0$ i prvi korak eliminacija je

$$\begin{pmatrix} 1 & 0 \\ -\dfrac{a}{a_{11}} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{11} & a^\tau \\ a & \hat{A} \end{pmatrix} = \begin{pmatrix} a_{11} & a^\tau \\ 0 & \hat{A} - \dfrac{aa^\tau}{a_{11}} \end{pmatrix}.$$

Sada primijetimo i da vrijedi

$$\begin{pmatrix} 1 & 0 \\ -\dfrac{a}{a_{11}} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{11} & a^\tau \\ a & \hat{A} \end{pmatrix} \begin{pmatrix} 1 & -\dfrac{a^\tau}{a_{11}} \\ 0 & I_{n-1} \end{pmatrix} = \begin{pmatrix} a_{11} & 0 \\ 0 & \hat{A} - \dfrac{aa^\tau}{a_{11}} \end{pmatrix},$$

pri čemu je dobivena matrica također pozitivno definitna. Sada se lako provjeri da je i matrica $\hat{A} - \dfrac{aa^\tau}{a_{11}}$ pozitivno definitna – dakle njen prvi dijagonalni element je strogo

veći od nule pa se postupak eliminacija može nastaviti. Time je pokazana egzistencija faktorizacije $A = R^\tau R$ u kojoj je R gornje trokutasta matrica. Faktorizaciju $A = R^\tau R$ zovemo *faktorizacija Choleskog* ili *trokutasta faktorizacija* simetrične pozitivno definitne matrice.
Elemente matrice R možemo izračunati jednostavnim nizom formula. Raspisivanjem relacije

$$A = R^\tau R = \begin{pmatrix} r_{11} & 0 & \cdots & 0 & 0 \\ r_{12} & r_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & r_{n-1,n-1} & 0 \\ r_{1n} & r_{2n} & \cdots & r_{n-1,n} & r_{nn} \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & \cdots & r_{2n} \\ \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & \vdots & \ddots & r_{n-1,n-1} & r_{n-1,n} \\ 0 & 0 & \cdots & 0 & r_{nn} \end{pmatrix}$$

po komponentama, za $i \leq j$, dobijemo

$$a_{ij} = \sum_{k=1}^{i} r_{ki} r_{kj}$$

odakle direktno slijedi sljedeći algoritam za računanje matrice R.

Algoritam 1.5.1 *Računanje faktorizacije Choleskog simetrične pozitivno definitne matrice* $A \in \mathbf{R}^{n\times n}$.

za $i = 1, \ldots, n$ {

$$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}\ ; \qquad /^*\ za\ i = 1,\ r_{ii} = \sqrt{a_{ii}}\ ^*/$$

za $j = i+1, \ldots, n$ {

$$r_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}}{r_{ii}}\ ;\ \}\ \}$$

Promotrimo sada linearni sustav jednadžbi u kojem je $A \in \mathbf{R}^{n\times n}$ simetrična, pozitivno definitna matrica. Ako je $A = R^\tau R$ trokutasta faktorizacija, onda rješenje $x = A^{-1}b = R^{-1}R^{-\tau}b$ možemo dobiti tako da prvo nađemo rješenje y sustava $R^\tau y = b$, a zatim riješimo sustav $Rx = y$. Kako je R gornje trokutasta matrica, cijeli postupak je vrlo jednostavan i možemo ga zapisati na sljedeći način:

Algoritam 1.5.2 *Rješavanje linearnog sustava jednadžbi* $Ax = b$ *sa pozitivno definitnom matricom* $A \in \mathbf{R}^{n\times n}$.

/ Trokutasta faktorizacija* $A = R^\tau R$ **/*
za $i = 1, \ldots, n$ {

$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}$; /* za $i = 1$, $r_{ii} = \sqrt{a_{ii}}$ */
za $j = i+1, \ldots, n$ {
$r_{ij} = \dfrac{a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}}{r_{ii}}$; } }
/* Supstitucije naprijed za $R^\tau y = b$ */
$y_1 = \dfrac{b_1}{r_{11}}$;
za $i = 2, \ldots, n$ {
$y_i = \dfrac{1}{r_{ii}}(b_i - \sum_{j=1}^{i-1} r_{ji} y_j)$; }
/* Supstitucije unazad za $Rx = y$ */
$x_n = \dfrac{y_n}{r_{nn}}$;
za $i = n-1, \ldots, 1$ {
$x_i = \dfrac{1}{r_{ii}}(y_i - \sum_{j=i+1}^{n} r_{ij} x_j)$; }

Naš cilj je ispitati numerička svojstva algoritma 1.5.1, ako njegove operacije izvedemo na računalu u (strojnoj) aritmetici konačne preciznosti ε.

Propozicija 1.5.1 *Neka je za zadanu $n \times n$ pozitivno definitnu matricu A algoritam 1.5.1 uspješno izvršio sve operacije u konačnoj aritmetici s greškom zaokruživanja ε. Ako je $\tilde{R}$ izračunata aproksimacija matrice R, onda je $\tilde{R}^\tau \tilde{R} = A + \delta A$, gdje je $\delta A = (\,\delta a_{ij}\,)$ simetrična matrica i za sve $1 \le i, j \le n$ vrijedi*

$$|\delta a_{ij}| \le \eta_C \sqrt{a_{ii} a_{jj}}, \quad \eta_C = \frac{c(n)\varepsilon}{1 - 2c(n)\varepsilon}, \quad c(n) = \max\{3, n\}. \tag{1.26}$$

Dokaz: U i–tom koraku, $2 \le i \le n$, u algoritmu 1.5.1 vrijedi[8]

$$\tilde{r}_{ii} = (1+\varepsilon_2)\sqrt{(1+\varepsilon_1)(a_{ii} - \sum_{k=1}^{i-1} \tilde{r}_{ki}^2 (1+\zeta_k))}, \tag{1.27}$$

pa kvadriranjem i uz oznaku $1 + \eta = (1+\varepsilon_1)(1+\varepsilon_2)^2$ dobijemo

$$\sum_{k=1}^{i} \tilde{r}_{ki}^2 = a_{ii} + \frac{\eta}{1+\eta} \tilde{r}_{ii}^2 - \sum_{k=1}^{i-1} \tilde{r}_{ki}^2 \zeta_k = a_{ii} + \delta a_{ii}. \tag{1.28}$$

[8] Ovdje koristimo pretpostavku da je algoritam uspješno završio sve operacije, tj. da je uspješno izračunao $\tilde{r}_{ii} > 0$ za sve i.

Ovu relaciju možemo zapisati i u obliku

$$\tilde{r}_{ii} = \sqrt{a_{ii} + \delta a_{ii} - \sum_{k=1}^{i-1} \tilde{r}_{ki}^2}. \tag{1.29}$$

Za $i = 1$ je trivijalno $\tilde{r}_{11} = \sqrt{(1+\varepsilon_2)^2 a_{11}} = \sqrt{a_{11} + \delta a_{11}}$.
Stavimo $\eta_i = \max\{|\eta|/(1+\eta), \max_{1\le k\le i-1} |\zeta_k|\}$. Lako provjerimo da u relaciji (1.29) vrijedi

$$|\delta a_{ii}| \le \eta_i \sum_{k=1}^{i} \tilde{r}_{ki}^2 \le \frac{\eta_i}{1-\eta_i} a_{ii}, \quad \eta_i \le \frac{c(n)\varepsilon}{1 - c(n)\varepsilon}.$$

Na isti način analiziramo računanje vrijednosti $\tilde{r}_{ij}$, $j > i$. Imamo

$$\tilde{r}_{ij} = (1+\epsilon_1)(1+\epsilon_2)\frac{a_{ij} - \sum_{k=1}^{i-1} \tilde{r}_{kj}\tilde{r}_{ki}(1+\zeta_k')}{\tilde{r}_{ii}}, \quad |\zeta_k'| \le \frac{(i-1)\varepsilon}{1-(i-1)\varepsilon},$$

pa stavljanjem $1 + \tau = (1+\epsilon_1)(1+\epsilon_2)$ laganim računom izvedemo da vrijedi

$$\sum_{k=1}^{i} \tilde{r}_{kj}\tilde{r}_{ki} = a_{ij} + \frac{\tau}{1+\tau}\tilde{r}_{ij}\tilde{r}_{ii} - \sum_{k=1}^{i-1} \tilde{r}_{kj}\tilde{r}_{ki}\zeta_k' = a_{ij} + \delta a_{ij} = a_{ji} + \delta a_{ji}. \tag{1.30}$$

Ovu relaciju možemo pročitati i kao

$$\tilde{r}_{ij} = \frac{a_{ij} + \delta a_{ij} - \sum_{k=1}^{i-1} \tilde{r}_{kj}\tilde{r}_{ki}}{\tilde{r}_{ii}}. \tag{1.31}$$

Ako sada definiramo $\tau_i = \max\{|\tau|/(1+\tau), \max_{1\le k\le i-1} |\zeta_k'|\}$, možemo pisati da je

$$\begin{aligned} |\delta a_{ij}| &\le \tau_i \sum_{k=1}^{i} |\tilde{r}_{kj}||\tilde{r}_{ki}| \le \tau_i \sqrt{\sum_{k=1}^{i} \tilde{r}_{kj}^2} \sqrt{\sum_{k=1}^{i} \tilde{r}_{ki}^2} \\ &\le \frac{\tau_i}{\sqrt{(1-\eta_i)(1-\eta_j)}} \sqrt{a_{ii}a_{jj}}, \quad \tau_i \le \frac{c(n)\varepsilon}{1-c(n)\varepsilon}. \end{aligned}$$

Konačno, primijetimo da relacije (1.28) i (1.30) pokazuju da je $\tilde{R}^\tau \tilde{R} = A + \delta A$, gdje je $\delta A = (\,\delta a_{ij}\,)_{i,j=1}^n$. Time je tvrdnja propozicije dokazana. ■

Zanimaju nas numerička svojstva algoritma 1.5.2. Ako je $\tilde{x}$ izračunata aproksimacija točnog rješenja $x = A^{-1}b$, što možemo reći o $\tilde{x}$? Iz propozicije 1.5.1 znamo da izračunata matrica $\tilde{R}$ zadovoljava

$$\tilde{R}^\tau \tilde{R} = A + \delta A, \quad \max_{i,j} \frac{|\delta a_{ij}|}{\sqrt{a_{ii}a_{jj}}} \le \eta_C$$

U sljedećem koraku rješavamo dva trokutasta sustava, $\tilde{R}^\tau y = b$ i $\tilde{R}x = y$. Neka su $\tilde{y} \approx y$ i $\tilde{x} \approx x$ izračunati vektori. Prema propoziciji 1.4.2, postoje gornje trokutaste matrice $\delta_1\tilde{R}$, $\delta_2\tilde{R}$ tako da vrijedi

$$(\tilde{R} + \delta_1\tilde{R})^\tau\tilde{y} = b, \quad (\tilde{R} + \delta_2\tilde{R})\tilde{x} = \tilde{y}.$$

Pri tome je $|\delta_1\tilde{R}| \leq \eta_\triangleright|\tilde{R}|$, $|\delta_2\tilde{R}| \leq \eta_\triangleright|\tilde{R}|$, gdje je $\eta_\triangleright = n\varepsilon/(1 - n\varepsilon)$. Slijedi da izračunati $\tilde{x}$ zadovoljava

$$(\tilde{R} + \delta_1\tilde{R})^\tau(\tilde{R} + \delta_2\tilde{R})\tilde{x} = b, \text{ tj. } (\tilde{R}^\tau\tilde{R} + \tilde{R}^\tau\delta_2\tilde{R} + (\delta_1\tilde{R})^\tau\tilde{R} + (\delta_1\tilde{R})^\tau\delta_2\tilde{R})\tilde{x} = b$$

Stavimo $E = \tilde{R}^\tau\delta_2\tilde{R} + (\delta_1\tilde{R})^\tau\tilde{R} + (\delta_1\tilde{R})^\tau\delta_2\tilde{R}$. Vrijedi $|E| \leq (2\eta_\triangleright + \eta_\triangleright^2)|\tilde{R}|^\tau|\tilde{R}|$, pa zaključujemo da $\tilde{x}$ zadovoljava sustav $(\tilde{R}^\tau\tilde{R} + E)\tilde{x} = b$, u kojem je E po elementima mala perturbacija matrice sustava $\tilde{A} = \tilde{R}^\tau\tilde{R}$. Primijetimo i da je

$$|E_{ij}| \leq (2\eta_\triangleright + \eta_\triangleright^2)\sqrt{\tilde{a}_{ii}\tilde{a}_{jj}} \leq (2\eta_\triangleright + \eta_\triangleright^2)(1 + \eta_C)\sqrt{a_{ii}a_{jj}}, \quad 1 \leq i, j \leq n.$$

Kako je $\tilde{A} = A + \delta A$, dobijemo sljedeću vezu između polaznog sustava i izračunatog rješenja $\tilde{x}$:

$$(A + F)\tilde{x} = b, \text{ gdje je } F = \delta A + E.$$

Pri tome su i E i δA ocijenjeni po elementima, na isti način i sa otprilike istom gornjom ogradom. Možemo reći da smo riješili sustav sa matricom $A + F$ koja je blizu zadane matrice A u smislu da je

$$\max_{i,j} \frac{|F_{ij}|}{\sqrt{a_{ii}a_{jj}}} \leq \xi, \quad \xi = \eta_C + (2\eta_\triangleright + \eta_\triangleright^2)(1 + \eta_C).$$

Da smo, na primjer, nakon faktorizacije $A + \delta A = \tilde{R}^\tau\tilde{R}$ trokutaste sustave po y i x riješili egzaktno, imali bismo $F = \delta A$ i $\xi = \eta_C$.
Ipak, nismo sasvim zadovoljni zaključkom. Zašto? Pažljivi čitatelj je sigurno već uočio da općenito E nije simetrična, što povlači da F nije simetrična, pa niti $A + F$ nije simetrična. Mi jesmo riješili sustav blizak zadanom, ali smo izgubili važnu strukturu polaznog sustava: simetriju. Simetrija matrice A sustava je posljedica strukture problema kojeg opisujemo sustavom $Ax = b$, pa nam je važno znati da $\tilde{x}$ odgovara rješenju bliskog problema, sa istom strukturom, tj. simetrijom. To nas vodi do sljedećeg problema:

- *Ako je $(A + F)\tilde{x} = b$, da li postoji simetrična perturbacija ΔA tako da je $(A + \Delta A)\tilde{x} = b$ i tako da za veličinu od ΔA postoje ocjene analogne onima za F?*

Sljedeća propozicija daje potvrdan odgovor na to pitanje.

Propozicija 1.5.2 *Neka je* $(A+F)\tilde{x} = b$, *gdje je* A *simetrična i pozitivno definitna i neka vrijedi*

$$\max_{i,j} \frac{|F_{ij}|}{\sqrt{a_{ii}a_{jj}}} \leq \xi.$$

Tada postoji simetrična perturbacija ΔA *tako da je* $(A+\Delta A)\tilde{x} = b$. *Pri tome je*

$$\max_{i\neq j} \frac{|\Delta a_{ij}|}{\sqrt{a_{ii}a_{jj}}} \leq \xi, \quad \max_{i} \frac{|\Delta a_{ii}|}{a_{ii}} \leq (2n-1)\xi.$$

Dokaz: Primijetimo da ΔA mora zadovoljavati jednadžbu $\Delta A\tilde{x} = F\tilde{x}$, koja daje n uvjeta za $n(n+1)/2$ stupnjeva slobode u ΔA. Stavimo $D = \mathrm{diag}(\sqrt{a_{ii}})_{i=1}^n$ i promotrimo skalirani sustav $D^{-1}(A+F)D^{-1}D\tilde{x} = D^{-1}b$, tj.

$$(A_s + F_s)z = D^{-1}b, \;\; A_s = D^{-1}AD^{-1}, \;\; F_s = D^{-1}FD^{-1}.$$

Neka je P permutacija takva da vektor $\tilde{z} = P^\tau z$ zadovoljava $|\tilde{z}_1| \leq |\tilde{z}_2| \leq \cdots \leq |\tilde{z}_n|$. Gornji sustav zapisat ćemo u ekvivalentnom obliku

$$P^\tau(A_s + F_s)P\tilde{z} = P^\tau D^{-1}b, \;\; \text{tj.} \;\; (A_{s,p} + F_{s,p})\tilde{z} = P^\tau D^{-1}b,$$

gdje smo stavili $A_{s,p} = P^\tau A_s P$, $F_{s,p} = P^\tau F_s P$. Konstruirat ćemo simetričnu matricu $M = (\, m_{ij}\,)$ za koju je $M\tilde{z} = F_{s,p}\tilde{z}$. Definirajmo

$$\begin{aligned} m_{ij} &= (F_{s,p})_{ij} \;\; \text{za } i < j \;; \\ m_{ij} &= (F_{s,p})_{ji} \;\; \text{za } j < i \;; \end{aligned}$$

i odredimo dijagonalne elemente m_{ii} tako da je

$$m_{ii}\tilde{z}_i + \sum_{j\neq i} m_{ij}\tilde{z}_j = (F_{s,p})_{ii}\tilde{z}_i + \sum_{j\neq i}(F_{s,p})_{ij}\tilde{z}_j.$$

Jednostavnom operacijom, koristeći definiciju izvandijagonalnih elemenata matrice M, dobijemo relaciju

$$m_{ii}\tilde{z}_i = (F_{s,p})_{ii}\tilde{z}_i + \sum_{j=1}^{i-1}((F_{s,p})_{ij} - (F_{s,p})_{ji})\tilde{z}_j.$$

Ako je $\tilde{z}_i = 0$, stavimo $m_{ii} = 0$. Inače, definiramo

$$m_{11} = (F_{s,p})_{11}, \quad m_{ii} = (F_{s,p})_{ii} + \sum_{j=1}^{i-1}((F_{s,p})_{ij} - (F_{s,p})_{ji})\frac{\tilde{z}_j}{\tilde{z}_i}, \;\; i = 2, \ldots, n.$$

Očito je $|m_{ii}| \leq (2i-1)\xi$, za sve i, te $\max_{i\neq j} |m_{ij}| \leq \xi$. Po konstrukciji matrice A vrijedi

$$(A_{s,p} + M)\tilde{z} = P^\tau D^{-1} b \text{ ili, ekvivalentno, } (A_s + PMP^\tau)z = D^{-1}b. \tag{1.32}$$

Pri tome je $\max_{i\neq j} |(PMP^\tau)_{ij}| \leq \xi$ i $\max_i |(PMP^\tau)_{ii}| \leq (2n-1)\xi$. Skaliranjem sustava (1.32) dobijemo

$$(A + \Delta A)\tilde{x} = b, \;\; \Delta A = D(PMP^\tau)D,$$

čime je dokaz završen. ■

Komentar 1.5.1 *Rezultat ove sekcije možemo sažeti u jednostavan zaključak: Pozitivno definitne sustave na računalu možemo riješiti sa pogreškom koja je ekvivalentna malim promjenama koeficijenata u matrici sustava.*

1.5.2 Indefinitni sustavi

1.5.3 Vježbe

1.6 Teorija perturbacija za linearne sustave

Iz prethodnih razmatranja je jasno da u primjenama rijetko možemo izračunati egzaktno rješenje sustava $Ax = b$. Jer, i formiranje samog sustava (računanje koeficijenata sustava i desne strane) i njegovo rješavanje na računalu uzrokuju greške. Analizom tih grešaka dolazimo do zaključka da izračunata aproksimacija rješenja $\tilde{x} = x + \delta x$ zadovoljava tzv. perturbirani sustav, $(A + \delta A)(x + \delta x) = b + \delta b$. Sada se postavlja pitanje kako ocijeniti veličinu greške $\delta x = \tilde{x} - x$, ako je poznata informacija o veličini grešaka δA i δb.

U primjeru 1.4.1 smo vidjeli da čak i mala perturbacija δA može potpuno promijeniti rješenje x. Kako mi u realnoj primjeni ne znamo točno rješenje, cilj nam je otkriti kako možemo iz matrice A i vektora b dobiti ne samo (što je moguće bolju) aproksimaciju $\tilde{x}$, nego i procjenu koliko je ta aproksimacija dobra.

Za početak teorijske analize, promotrimo jednostavniji slučaj u kojem je $\delta b = 0$. Dakle, jedina perturbacija je ona koja A promijeni u $A + \delta A$. Zbog jednostavnosti ćemo promatrati samo (inače, važan) slučaj u kojem je matrica koeficijenata $A + \delta A$ i dalje regularna, pa je $x + \delta x$ jedinstveno određen.

Iz jednakosti $A + \delta A = A(I + A^{-1}\delta A)$ vidimo da je regularnost matrice $A + \delta A$ osigurana ako je $I + A^{-1}\delta A$ regularna. Uvjet pod kojim možemo garantirati regularnost matrice $I + A^{-1}\delta A$ daje sljedeća propozicija.

Propozicija 1.6.1 *Neka je X $n \times n$ matrica i $\|\cdot\|$ proizvoljna matrična norma. Ako je $\|X\| < 1$ onda je $I - X$ regularna matrica i*

$$(I - X)^{-1} = I + X + X^2 + \cdots = \sum_{k=0}^{\infty} X^k.$$

Dokaz: Primijetimo da za svaki prirodan broj m vrijedi

$$(I-X)(I+X+\cdots+X^m) = I+X+\cdots+X^m - X - \cdots - X^m - X^{m+1} = I - X^{m+1}.$$

Ako označimo $S_m = I + X + \cdots + X^m$, onda možemo pisati

$$(I - X)S_m = S_m(I - X) = I - X^{m+1}, \text{ tj. } S_m = (I - X)^{-1} - (I - X)^{-1}X^{m+1}.$$

Kako je, zbog $\|X\| < 1$,

$$\|(I - X)^{-1}X^{m+1}\| \leq \|(I - X)^{-1}\|\|X\|^{m+1} \longrightarrow 0, \text{ kada } m \to \infty,$$

zaključujemo da je za dovoljno veliki indeks m matrica S_m po volji blizu matrici $(I - X)^{-1}$. ■

Koristeći ovu propoziciju, regularnost matrice $A + \delta A$ obično osiguravamo tako da zahtijevamo da je $\|A^{-1}\delta A\| < 1$. Izbor matrične norme $\|\cdot\|$ ovisi o konkretnoj situaciji, npr. o tipu informacije o δA ili o teorijskim rezultatima koje koristimo u analizi. Neka je matrična norma jednaka Frobeniusovoj normi, $\|\cdot\| = \|\cdot\|_F$,

$$\|X\|_F = \sqrt{\sum_{i,j=1}^{n} |X_{ij}|^2} = \sqrt{\text{Trag}(X^\tau X)}.$$

Informacija o perturbaciji δA je važan faktor u razvoju analize. Neka je na primjer zadano (poznato) da je

$$\epsilon \equiv \frac{\|\delta A\|_F}{\|A\|_F} \ll 1$$

mali broj, tj. da je perturbacija *mala po normi*. Regularnost matrice $A + \delta A$ je osigurana ako je npr.

$$\|A^{-1}\|_F\|\delta A\|_F = \epsilon(\|A\|_F\|A^{-1}\|_F) < 1, \text{ tj. } \epsilon < \frac{1}{\|A\|_F\|A^{-1}\|_F}.$$

Tada je $\|A^{-1}\delta A\|_F < 1$ i $(A + \delta A)^{-1} = (I + A^{-1}\delta A)^{-1}A^{-1}$, pa $\tilde{x} = (A + \delta A)^{-1}b$ možemo pisati kao

$$\tilde{x} = (I + A^{-1}\delta A)^{-1}A^{-1}b = (I + A^{-1}\delta A)^{-1}x, \text{ tj. } (I + A^{-1}\delta A)\tilde{x} = x.$$

Znači, $x - \tilde{x} = A^{-1}\delta A\tilde{x}$, pa je

$$\|x - \tilde{x}\|_2 \leq \|A^{-1}\delta A\|_F \|\tilde{x}\|_2.$$

Kako je $\|A^{-1}\delta A\|_F \leq \|A^{-1}\|_F \|\delta A\|_F$, dobijamo

$$\frac{\|x - \tilde{x}\|_2}{\|\tilde{x}\|_2} \leq \|A^{-1}\|_F \|A\|_F \frac{\|\delta A\|_F}{\|A\|_F} = \epsilon \|A^{-1}\|_F \|A\|_F. \tag{1.33}$$

Relacija (1.33) pokazuje da relativna greška u izračunatom rješenju $\tilde{x}$ može biti uvećana najviše sa faktorom $\kappa_F(A) = \|A^{-1}\|_F \|A\|_F$ u odnosu na relativnu promjenu $\epsilon = \|\delta A\|_F / \|A\|_F$ u polaznoj matrici A.

1.6.1 Perturbacije male po normi

Sljedeći teorem daje potpuni opis greške ako je perturbacija dana po normi. Općenito ćemo promatrati i δA i δb, a mjerenja perturbacija će biti u proizvoljnoj vektorskoj normi $\|\cdot\|$ i pripadnoj matričnoj normi $\|\cdot\|$.

Teorem 1.6.1 *Neka je $Ax = b$, $(A + \delta A)(x + \delta x) = b + \delta b$, gdje je $\|\delta A\| \leq \epsilon \|A\|$, $\|\delta b\| \leq \epsilon \|b\|$. Ako je $\epsilon \|A^{-1}\| \|A\| < 1$, onda je*

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\epsilon}{1 - \epsilon \|A^{-1}\| \|A\|} \left(\frac{\|A^{-1}\| \|b\|}{\|x\|} + \|A^{-1}\| \|A\| \right) \leq 2 \frac{\epsilon \|A^{-1}\| \|A\|}{1 - \epsilon \|A^{-1}\| \|A\|}.$$

Pri tome postoje perturbacije δA i δb za koje je gornja nejednakost skoro dostignuta. Preciznije, postoje δA i δb tako da je $\|\delta A\| = \epsilon \|A\|$, $\|\delta b\| = \epsilon \|b\|$, te

$$\frac{\|\delta x\|}{\|x\|} \geq \frac{\epsilon}{1 + \epsilon \|A^{-1}\| \|A\|} \left(\frac{\|A^{-1}\| \|b\|}{\|x\|} + \|A^{-1}\| \|A\| \right).$$

Dokaz: Iz pretpostavki teorema je

$$\delta x = A^{-1}\delta b - A^{-1}\delta A x - A^{-1}\delta A \delta x, \tag{1.34}$$

pa uzimanjem norme dobijemo

$$\|\delta x\| \leq \epsilon \|A^{-1}\| \|b\| + \epsilon \|A^{-1}\| \|A\| \|x\| + \epsilon \|A^{-1}\| \|A\| \|\delta x\|,$$

odakle, rješavanjem nejednakosti po $\|\delta x\|$ slijedi tvrdnja.
Da bismo konstruirali perturbacije za koje dobivena nejednakost skoro postaje jednakost, pogledajmo desnu stranu jednakosti (1.34). Vrijedi

$$\|\delta x\| \geq \|A^{-1}\delta b - A^{-1}\delta A x\| - \|A^{-1}\delta A \delta x\|.$$

Pokušajmo odrediti perturbacije δA, δb tako da vrijedi

$$\|A^{-1}\delta b - A^{-1}\delta A x\| = \epsilon\|A^{-1}\|\|b\| + \epsilon\|A^{-1}\|\|A\|\|x\|.$$

Dakle, δA i δb treba odabrati tako da je $\|\delta A\| \leq \epsilon\|A\|$, $\|\delta b\| \leq \epsilon\|b\|$,

$$\|A^{-1}\delta b\| = \epsilon\|A^{-1}\|\|b\|, \quad \|A^{-1}\delta A x\| = \epsilon\|A^{-1}\|\|A\|\|x\|,$$

te da je norma razlike $A^{-1}\delta b - A^{-1}\delta A x$ jednaka sumi normi vektora. Ovaj zadnji uvjet znači da $A^{-1}\delta b$ i $-A^{-1}\delta A x$ moraju biti kolinearni.
Ako je u jedinični vektor za kojeg je $\|A^{-1}u\| = \|A^{-1}\|$, onda $\delta b = \epsilon\|b\|u$ zadovoljava $\|\delta b\| = \epsilon\|b\|$ i $\|A^{-1}\delta b\| = \epsilon\|A^{-1}\|\|b\|$. Sada stavimo $\delta A = \epsilon\|A\|uv^\tau$, gdje je $v \in \mathbf{R}^n$ vektor kojeg ćemo odrediti da postignemo željene relacije:

(i) $\|\delta A\| = \epsilon\|A\| \max\limits_{z\neq 0} \dfrac{\|uv^\tau z\|}{\|z\|} = \epsilon\|A\| \max\limits_{z\neq 0} \dfrac{|v^\tau z|}{\|z\|}$ treba postati $\|\delta A\| = \epsilon\|A\|$;

(ii) $\|A^{-1}\delta A x\| = \epsilon\|A\|\|A^{-1}\||v^\tau x|$ treba postati $\|A^{-1}\delta A x\| = \epsilon\|A\|\|A^{-1}\|\|x\|$.

Dakle, treba nam vektor v sa svojstvom da je, za sve z, $|v^\tau z| \leq \|z\|$, te da je $|v^\tau x| = \|x\|$. Postojanje takvog vektora je rezultat Hahn–Banachovog teorema: takav vektor v postoji. Dakle, konstruirali smo δA i δb za koje je

$$\|\delta x\| \geq \epsilon\|A^{-1}\|\|b\| + \epsilon\|A^{-1}\|\|A\|\|x\| - \epsilon\|A^{-1}\|\|A\|\|\delta x\|,$$

čime je dokaz druge tvrdnje teorema završen. ■
Vidimo da teorem 1.6.1 iz zadane informacije o veličini perturbacija po normi ($\|\delta A\| \leq \epsilon\|A\|$, $\|\delta b\| \leq \epsilon\|b\|$) izvodi optimalnu[9] ocjenu iz koje se jasno vidi da je broj

$$\kappa(A) = \|A^{-1}\|\|A\| \tag{1.35}$$

odlučujući faktor u donošenju suda o numeričkoj kvaliteti izračunate aproksimacije $\tilde{x} = x + \delta x$ sustava $Ax = b$. Pravilo je jednostavno:

♣ *Ako je relativna greška (po normi) u podacima najviše ϵ, onda se relativna greška u rješenju ponaša kao $\kappa(A)\epsilon$.*

[9]Ovdje pod optimalnosti podrazumijevamo činjenicu da je gornju ogradu za $\|\delta x\|/\|x\|$ nemoguće bitno poboljšati.

1.6.2 Rezidualni vektor i stabilnost

Postoji još jedan jednostavan i koristan način kako prosuditi kvalitetu aproksimacije $\tilde{x}$ rješenja sustava $Ax = b$. Radi se o *rezidualnom vektoru*

$$r = b - A\tilde{x}. \tag{1.36}$$

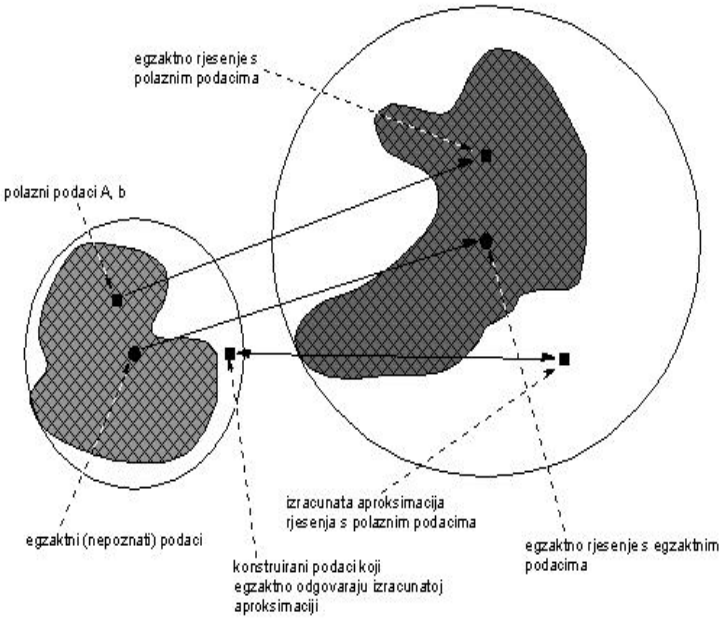
Kako je $b - Ax = 0$, jasno je da bi za dobru aproksimaciju $\tilde{x}$ pripadni rezidual trebao biti mali po normi. Ako relaciju (1.36) pročitamo kako

$$A\tilde{x} = b - r, \quad \text{tj. kao} \quad A\tilde{x} = b + \delta b, \quad \text{gdje je} \quad \delta b = -r,$$

onda vidimo da je $\tilde{x}$ egzaktno rješenje sustava koji je dobiven iz originalnog sustava promjenom desne strane u $b + \delta b$. Ako je

$$\epsilon \equiv \frac{\|r\|}{\|b\|}$$

dovoljno mali broj, onda možemo $\tilde{x}$ prihvatiti kao zadovoljavajuću aproksimaciju. Zašto? Recimo da su naša polazna matrica A i vektor b rezultati mjerenja ili nekih prethodnih proračuna.



Slika 1.3: Interpretacija izračunatog rješenja $\tilde{x}$ kao egzaktnog rješenja promijenjenog sustava jednadžbi.

Teorem 1.6.2 *Neka je $\tilde{x}$ aproksimacija rješenja sustava $Ax = b$ i neka je*

$$\beta(\tilde{x}) = \min\{\epsilon \ : \ (A+\delta A)\tilde{x} = b+\delta b, \ \ \|\delta A\| \leq \epsilon\|A\|, \ \ \|\delta b\| \leq \epsilon\|b\|\}.$$

Tada je

$$\beta(\tilde{x}) = \frac{\|b - A\tilde{x}\|}{\|A\|\|\tilde{x}\| + \|b\|}.$$

Dokaz: Neka je $r = b - A\tilde{x}$ rezidualni vektor. Ako je $\epsilon \geq 0$ takav da postoje δA, δb takvi da je $\|\delta A\| \leq \epsilon\|A\|$, $\|\delta b\| \leq \epsilon\|b\|$, $(A+\delta A)\tilde{x} = b + \delta b$, onda vrijedi

$$r = \delta A\tilde{x} - \delta b, \ \ \text{pa je} \ \ \|r\| \leq \epsilon(\|A\|\|\tilde{x}\| + \|b\|), \ \ \text{tj.} \ \ \epsilon \geq \underline{\epsilon} \equiv \frac{\|b - A\tilde{x}\|}{\|A\|\|\tilde{x}\| + \|b\|}.$$

Stavimo sada

$$\delta b = -\frac{\|b\|}{\|A\|\|\tilde{x}\| + \|b\|} r.$$

Očito je $\|\delta b\| = \underline{\epsilon}\|b\|$. Odredimo δA tako da je $\|\delta A\| = \underline{\epsilon}\|A\|$ i

$$(A+\delta A)\tilde{x} = b - \frac{\|b\|}{\|A\|\|\tilde{x}\| + \|b\|} r, \ \ \text{tj.} \ \ \delta A\tilde{x} = \frac{\|A\|\|\tilde{x}\|}{\|A\|\|\tilde{x}\| + \|b\|} r.$$

Definirajmo

$$\delta A = \frac{\|A\|}{\|A\|\|\tilde{x}\| + \|b\|} r v^\tau,$$

gdje je v vektor sa svojstvima

$$v^\tau \tilde{x} = \|\tilde{x}\| \ \ \text{i za sve } z \text{ vrijedi } \ |v^\tau z| \leq \|z\|.$$

Takav vektor v postoji po Hahn–Banachovom teoremu i lako provjerimo da δA ima sva tražena svojstva. ■

1.6.3 Perturbacije po elementima

Najpreciznija ocjena perturbacije u matrici A je kada imamo informaciju o perturbaciji svakog njenog elementa, tj. svakog koeficijenta u sustavu jednadžbi. Ako je $A = (\, a_{ij} \,)_{i,j=1}^n$ i $A + \delta A = (\, a_{ij} + \delta a_{ij} \,)_{i,j=1}^n$, onda je takva ocjena dana relacijama

$$|\delta a_{ij}| \leq \varepsilon |a_{ij}|, \ \ i,j = 1,2,\ldots,n,$$

gdje je $0 \leq \varepsilon \ll 1$. Ove nejednakosti jednostavno zapisujemo kao $|\delta A| \leq \varepsilon|A|$, tj. apsolutne vrijednosti matrica i nejednakost među matricama shvaćamo po elementima. Na isti način pišemo $|\delta b| \leq \varepsilon|b|$. Primijetimo da ovakve perturbacije ($|\delta A| \leq \varepsilon|A|$, $|\delta b| \leq \varepsilon|b|$) čuvaju strukturu u smislu da nule u matrici A i vektoru b ostaju nepromijenjene. Nadalje, ove perturbacije su neizbježne pri pohranjivanju podataka u računalo.

Teorem 1.6.3 *Neka je* $Ax = b$ *i* $(A + \delta A)(x + \delta x) = b + \delta b$, *gdje je*

$$|\delta A| \leq \varepsilon |A|, \quad |\delta b| \leq \varepsilon |b|.$$

Uzmimo proizvoljnu apsolutnu vektorsku normu $\|\cdot\|$ *i njenu induciranu matričnu normu, također označenu s* $\|\cdot\|$. *Neka je* $\varepsilon \||A^{-1}||A|\| < 1$. *Tada vrijedi*

$$\frac{\|\delta x\|}{\|x\|} \leq \varepsilon \frac{\||A^{-1}||A||x| + |A^{-1}||b|\|}{(1 - \varepsilon \||A^{-1}||A|\|)\|x\|}. \tag{1.37}$$

Nadalje, postoje perturbacije δA *i* δb *takve da je* $|\delta A| = \varepsilon |A|$, $|\delta b| = \varepsilon |b|$ *i da za rješenje* $x + \delta x = (A + \delta A)^{-1}(b + \delta b)$ *vrijedi*

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty} \geq \varepsilon \frac{\||A^{-1}||A||x| + |A^{-1}||b|\|_\infty}{(1 + \varepsilon \||A^{-1}||A|\|_\infty)\|x\|_\infty}. \tag{1.38}$$

Dokaz: Prije svega, uvjet $\varepsilon \||A^{-1}||A|\| < 1$ osigurava da je $A + \delta A$ regularna matrica, pa je $x + \delta x$ jedinstveno određen. Sada lako provjerimo da vrijedi

$$\delta x = -A^{-1}\delta A(x + \delta x) + A^{-1}\delta b, \tag{1.39}$$

pa primjenom nejednakosti trokuta (mnogokuta) dobijemo da je

$$\begin{aligned} |\delta x| &\leq |A^{-1}||\delta A||x| + |A^{-1}||\delta A||\delta x| + |A^{-1}||\delta b| \\ &\leq \varepsilon |A^{-1}||A||x| + \varepsilon |A^{-1}||A||\delta x| + \varepsilon |A^{-1}||b| \end{aligned}$$

pa je

$$\|\delta x\| \leq \varepsilon(\||A^{-1}||A||x| + |A^{-1}||b|\|) + \varepsilon \||A^{-1}||A|\|\|\delta x\|$$

Neka je $m \in \{1, 2, \ldots, n\}$ odabran tako da je

$$(|A^{-1}||A||x| + |A^{-1}||b|)_m = \||A^{-1}||A||x| + |A^{-1}||b|\|_\infty.$$

Definirajmo dijagonalne matrice

$$D_1 = \operatorname{diag}(sign((A^{-1})_{mi}))_{i=1}^n, \quad D_2 = \operatorname{diag}(sign(x_i))_{i=1}^n,$$

i perturbacije $\delta A = \varepsilon D_1 |A| D_2$, $\delta b = -\varepsilon D_1 |b|$. Sada lako računamo

$$\begin{aligned} (A^{-1}\delta A x - A^{-1}\delta b)_m &= \sum_{j=1}^n \sum_{i=1}^n (A^{-1})_{mj}(\delta A)_{ji} x_i - \sum_{j=1}^n (A^{-1})_{mj}\delta b_j \\ &= \varepsilon(|A^{-1}||A||x| + |A^{-1}||b|)_m \\ &= \varepsilon \||A^{-1}||A||x| + |A^{-1}||b|\|_\infty. \end{aligned}$$

S druge strane, iz relacije 1.39 lako izvedemo da je

$$(A^{-1}\delta A x - A^{-1}\delta b)_m = -(\delta x + A^{-1}\delta A \delta x)_m,$$

pa je

$$\varepsilon \| |A^{-1}||A||x| + |A^{-1}||b| \|_\infty \leq \|\delta x\|_\infty + \varepsilon \| |A^{-1}||A| \|_\infty \|\delta x\|_\infty.$$

Korolar 1.6.1 *Relativni faktor osjetljivosti je dan relacijom*

$$\lim_{\varepsilon \to 0} \sup \left\{ \frac{\frac{\|\delta x\|_\infty}{\|x\|_\infty}}{\varepsilon} \ : \ (A+\delta A)(x+\delta x) = b + \delta b,\ |\delta A| \leq \varepsilon |A|,\ |\delta b| \leq \varepsilon |b| \right\}$$
$$= \frac{\| |A^{-1}||A||x| + |A^{-1}||b| \|_\infty}{\|x\|_\infty}.$$

Primijetimo da je

$$\| |A^{-1}||A||x| \|_\infty \leq \| |A^{-1}||A||x| + |A^{-1}||b| \|_\infty \leq 2 \| |A^{-1}||A||x| \|_\infty.$$

Zato kao koeficijent osjetljivosti možemo koristiti veličinu

$$\kappa_\infty(A, x) = \frac{\| |A^{-1}||A||x| \|_\infty}{\|x\|_\infty}.$$

Teorem 1.6.4 *Neka je $\tilde{x}$ izračunata aproksimacija rješenja sustava $Ax = b$ i neka je $r = b - A\tilde{x}$ izračunati rezidual. Vrijedi*

$$\min\{\varepsilon \geq 0 \ : \ (A+\delta A)\tilde{x} = b + \delta b,\ |\delta A| \leq \varepsilon |A|,\ |\delta b| \leq \varepsilon |b|\} = \max_i \frac{|r_i|}{(|A||\tilde{x}| + |b|)_i}$$

Optimalna perturbacija polaznih podataka koja reproducira izračunato rješenje je dana s

$$\delta A = D_1 |A| D_2, \quad \delta b = -D_1 |b|,$$

gdje je $D_1 = \operatorname{diag}\left(\frac{|r_i|}{(|A||\tilde{x}| + |b|)_i} \right)_{i=1}^n$, $D_2 = \operatorname{diag}(sign(\tilde{x}_i))_{i=1}^n$.

Primjer 1.6.1 *U ovom primjeru istražujemo stabilnost Gaussovih eliminacija po elementima matrice. Neka su $\alpha \neq 0$ i $\beta \neq 0$ zadani i neka je*

$$A = \begin{pmatrix} \alpha & \beta \\ \alpha & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad x = A^{-1}b = \begin{pmatrix} 0 \\ \frac{1}{\beta} \end{pmatrix}.$$

Trokutasta LU faktorizacija matrice A je

$$\underbrace{\begin{pmatrix} \alpha & \beta \\ \alpha & 0 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} \alpha & \beta \\ 0 & -\beta \end{pmatrix}}_{U}.$$

Vektor $y = L^{-1}b = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ *je izračunat bez greške Ako pogledamo proces supstitucija unazad, egzaktne formule* $x_2 = 1/\beta$, $x_1 = 0$, *prelaze u*

$$\begin{aligned} \tilde{x}_2 &= \frac{1}{\beta}(1+\xi_1), \\ \tilde{x}_1 &= \frac{1}{\alpha}(1-\beta\tilde{x}_2(1+\xi_2))(1+\xi_3)(1+\xi_4) = \frac{1}{\alpha}(-\xi_1-\xi_2-\xi_1\xi_2)(1+\xi_3)(1+\xi_4), \end{aligned}$$

gdje su $\xi_1, \xi_2, \xi_3, \xi_4$ *male greške reda veličine strojne točnosti. Primijetimo da općenito ne možemo garantirati* $\tilde{x}_1 = 0$. *Ako je* β *strojni broj takav da je* $\beta \odot (1 \oslash \beta) \neq 1$, *bit će* $\tilde{x}_1 \neq 0$. *Na primjer u IEEE aritmetici je takav broj npr.* $\beta = 4.057062130620955e-001$, *pri čemu je* $\beta \odot (1 \oslash \beta) = 9.999999999999999e-001$. *(Čitatelju za vježbu ostavljamo da pokuša naći još takvih brojeva.)*

Pokušajmo sada izračunato rješenje $\tilde{x}_1, \tilde{x}_2$ *interpretirati kao egzaktno rješenje sustava* $(A+\delta A)\tilde{x} = b + \delta b$, *gdje su elementi matrice* δA *oblika* $a_{ij}(1+\epsilon_{ij})$, *a elementi od* δb *su* $b_i(1+\varepsilon_i)$. *Drugim riječima, treba odrediti* ϵ_{ij}, ε_i, $i, j = 1, \ldots, n$, *tako da vrijedi*

$$\begin{pmatrix} \alpha(1+\epsilon_{11}) & \beta(1+\epsilon_{12}) \\ \alpha(1+\epsilon_{21}) & 0 \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} 1+\varepsilon_1 \\ 0 \end{pmatrix}. \tag{1.40}$$

Ako je $\tilde{x}_1 \neq 0$, *jasno je da je za zadovoljavanje druge jednadžbe u gornjem sustavu nužno uzeti* $\epsilon_{21} = -1$, *što znači da element* a_{21} *treba promijeniti u nulu. Znači da imamo veliku promjenu elementa,* $\delta a_{21} = -a_{21}$, *tj.* $(\delta A)_{22} = -\alpha$, *pa je i*

$$\frac{\|\delta A\|_F}{\|A\|_F} \geq \frac{|\alpha|}{\sqrt{3}|\alpha|} > \frac{1}{2}.$$

Iz prethodnog primjera zaključujemo da bez obzira na točnost koju koristimo u računanju na stroju, općenito ne možemo garantirati da će izračunato rješenje $\tilde{x}$ biti točno rješenje sustava $\tilde{A}\tilde{x} = \tilde{b}$ u kojem su $\tilde{A}$ i $\tilde{b}$ nastali malim relativnim perturbacijama koeficijenata u A i b.

1.6.4 Dodatak: Udaljenost matrice do skupa singularnih matrica

1.6.5 Dodatak: Dualne norme i Hahn–Banachov teorem

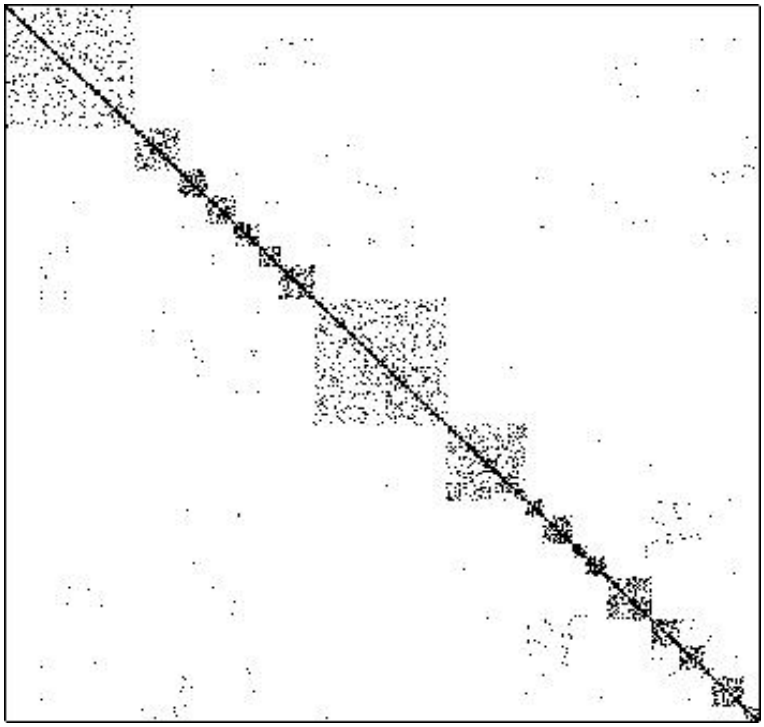
1.6.6 Vježbe

1.7 Iterativne metode

U prethodnim sekcijama smo vidjeli da rješenje linearnog sustava $Ax = b$ općenito ne možemo izračunati apsolutno točno. Također, u nekim primjenama niti točno rješenje $x = A^{-1}b$ nije puno bolje od neke dovoljno dobre aproksimacije $\tilde{x}$, gdje obično $\tilde{x}$ zadovoljava sustav $(A+\delta A)\tilde{x} = b$, blizak polaznom. Dakle, Gaussove eliminacije koje su jednostavno konačan niz formula koje vode rješenju sustava ne garantiraju idealnu točnost.

Nadalje, u praksi moramo biti svjesni da je stroj (računalo) omeđen ne samo u pitanju numeričke točnosti nego i u još dva važna aspekta: raspoloživi memorijski prostor i vrijeme izvršavanja. Moderne primjene matematike zahtijevaju rješenja sustava velikih dimenzija, npr. $n > 10^5$. Lako se uvjeriti da je u takvim primjerima proces Gaussovih eliminacija često praktično neupotrebljiv. Jer, matrica dimenzije $n = 10^5$ zahtijeva $n^2 = 10^{10}$ lokacija u memoriji, svaka barem 4 bajta (veličina reprezentacije realnog broja u jednostrukoj preciznosti). Dakle, moguće je da samo spremanje matrice koeficijenata u memoriju računala predstavalja poteškoću – ponekad matricu držimo na vanjskoj, sporoj memoriji (datoteka na disku) i onda možemo dijelove matrice učitavati dio po dio u radnu memoriju. Kako u takvim uvjetima implementirati Gaussove eliminacije?

U puno važnih primjena je matrica sustava A velike dimenzije, ali je *rijetko popunjena*. To znači da je velika većina elemenata od A jednaka nuli, a elementi koji nisu nula su obično pravilno raspoređni po matrici ili čak imaju i pravilno raspoređene numeričke vrijednosti. Na primjer matrica iz primjera 1.2.1 ima broj 2 na dijagonali i -1 na dvije sporedne dijagonale, a ostali elementi su nule. To znači da za takvu matricu A računanje produkta Av, gdje je $v \in \mathbf{R}^n$, zahtijeva $3n$ množenja i $2n$ zbrajanja – ukupno $5n$ operacija. (Ako A nema strukturu, onda općenito Av zahtijeva $2n^2 - n$ operacija.) Još jedan primjer rijetko popunjene matrice je dan na slici 1.4. Ponovo uočavamo da je u svakom retku broj elemenata koji nisu nula unaprijed poznat (kao i pozicije gdje se ti elementi nalaze) i da je broj takvih elemenata puno manji od dimenzije n. To znači da je računanje produkta Av složenosti puno manje od $2n^2 - n$. Ponekad je matrica A velike dimenzije i gusto popunjena (svi ili velika većina elemenata su različiti od nule) tako da jednostavno ne može stati u memoriju računala. Najbolje što možemo je učitavanje dijelova matrice iz vanjske u radnu memoriju.



Slika 1.4: Rijetko popunjena matrica. Točkice pokazuju pozicije elemenata u matrici koji su različiti od nule.

Ponekad je poznat način kako su generirani elementi matrice (npr. poznato da je $a_{ij} = f_{ij}(\cdots)$, gdje su $f_{ij}(\cdots)$ poznate funkcije nekih parametara) pa uvijek možemo generirati dijelove matrice. Moguće je i da je u konkretnoj aplikaciji jedino zadano kako A djeluje kao linearni operator – postoji potprogram koji za zadani v računa Av. Ako je to jedini način kako doći do matrice A, kako onda riješiti $Ax = b$?
Prethodna diskusija nas motivira da potražimo i drugačije pristupe za rješavanje linearnog sustava $Ax = b$. Primijetimo da ne moramo nužno težiti pronalaženju egzaktnog rješenja – umjesto toga želimo *dovoljno dobru* aproksimaciju $\tilde{x}$. Zato ima smisla pokušati konstruirati niz $x^{(0)}, x^{(1)}, \ldots, x^{(k)}, \ldots$ vektora iz $\mathbf{R}^n$ sa sljedećim svojstvima:

(i) za svaki k je formula za računanje $x^{(k)}$ jednostavna ;

(ii) $x^{(k)}$ teži prema $x = A^{-1}b$ i za neki k (obično $k \ll n$) je $x^{(k)}$ prihvatljiva aproksimacija za x.

Nabrojana svojstva su namjerno za sada dana u nepreciznoj, ali lako razumljivoj formi. Detalji, koji ovise o konkretnom problemu i o konkretnom načinu konstruiranja niza $(x^{(k)})$, će biti dani malo kasnije.

1.7.1 Jacobijeva i Gauss–Seidelova metoda

Jacobijeva i Gauss–Seidelova metoda spadaju u klasične i najjednostavnije iterativne metode za rješavanje linearnih sustava. Ideju Jacobijeve metode ćemo ilustrirati na primjeru 2×2 sustava

$$\begin{array}{rcrcl} a_{11}x_1 & + & a_{12}x_2 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & = & b_2 \end{array}, \quad a_{11} \neq 0, a_{22} \neq 0.$$

Uočimo da rješenje $x = (\, x_1, x_2 \,)^\tau$ zadovoljava

$$\begin{array}{rcl} x_1 & = & \dfrac{1}{a_{11}}(b_1 - a_{12}x_2) \\ x_2 & = & \dfrac{1}{a_{22}}(b_2 - a_{21}x_1) \end{array}.$$

Te relacije motiviraju da neku približnu vrijednost rješenja $x^{(0)} = \left(x_1^{(0)}, x_2^{(0)} \right)^\tau$ korigiramo pomoću formula

$$\begin{array}{rcl} x_1^{(1)} & = & \dfrac{1}{a_{11}}(b_1 - a_{12}x_2^{(0)}) \\ x_2^{(1)} & = & \dfrac{1}{a_{22}}(b_2 - a_{21}x_1^{(0)}) \end{array}. \tag{1.41}$$

Nadamo se da je $x^{(1)} = \left(x_1^{(1)}, x_2^{(1)} \right)^\tau$ bolja aproksimacija nego što je to $x^{(0)}$. Na isti način možemo iskoristiti $x^{(1)}$ da dobijemo sljedeću aproksimaciju, $x^{(2)}$, zatim $x^{(3)}$, itd. Pitanje je, naravno, pod kojim uvjetima te iteracije teže prema rješenju x.
Prije same analize, zgodno je uočiti strukturu računanja vektora $x^{(k+1)}$ pomoću $x^{(k)}$. Primijetimo da vrijedi

$$\begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \end{pmatrix} = \begin{pmatrix} \dfrac{1}{a_{11}} & 0 \\ 0 & \dfrac{1}{a_{22}} \end{pmatrix} \left(\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \begin{pmatrix} 0 & -a_{12} \\ -a_{21} & 0 \end{pmatrix} \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \end{pmatrix} \right)$$

Dakle, ako stavimo

$$A = D - N, \;\; D = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}, \;\; N = \begin{pmatrix} 0 & -a_{12} \\ -a_{21} & 0 \end{pmatrix}, \tag{1.42}$$

onda možemo jednostavno pisati

$$x^{(k+1)} = D^{-1}(b + Nx^{(k)}) = D^{-1}Nx^{(k)} + D^{-1}b. \tag{1.43}$$

Relacijom (1.43) je definirana □*acobijeva iterativna metoda.*
Pokušajmo ovaj proces ilustrirati na jednom primjeru. Zbog jednostavnosti je primjer dimenzije 2×2 tako da svatko može lako slijediti račun.

Primjer 1.7.1 *Neka je*

$$A = \begin{pmatrix} 2 & 0.1 \\ -0.1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 19.9 \\ -3 \end{pmatrix}, \quad x = A^{-1}b = \begin{pmatrix} 10 \\ -1 \end{pmatrix}.$$

□atricu A napišimo kao u relaciji (□.□□). Za početnu iteraciju uzmimo vektor

$$x^{(0)} = D^{-1}b = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 19.9 \\ -3 \end{pmatrix} = \begin{pmatrix} 9.949999999999999 \\ -1.5 \end{pmatrix}.$$

Primijetimo da u početnoj iteraciji pokušavamo □pogoditi□ rješenje. Ponekad je dovoljno uzeti slučajno odabran vektor. Ipak, poželjno je da je polazna iteracija što je moguće bliže cilju. Naš izbor je bio rezultat jednostavne ideje□matricu A aproksimiramo s D (jer su dijagonalni elementi veći od izvandijagonalnih), pa $A^{-1}b$ aproksimiramo s $D^{-1}b$. Naravno da je ovo gruba aproksimacija, ali ipak ima smisla. □ada iteriramo kao u relaciji (□.□□) i dobijemo

$$\begin{aligned} x^{(1)} &= \begin{pmatrix} 1.002500000000000e+001 \\ -1.002500000000000e+000 \end{pmatrix}, \quad x^{(2)} = \begin{pmatrix} 1.000012500000000e+001 \\ -9.987500000000000e-001 \end{pmatrix} \\ x^{(3)} &= \begin{pmatrix} 9.999937500000000e+000 \\ -9.999937500000000e-001 \end{pmatrix}, \quad x^{(4)} = \begin{pmatrix} 9.999999687499999e+000 \\ -1.000003125000000e+000 \end{pmatrix} \\ x^{(5)} &= \begin{pmatrix} 1.000000015625000e+001 \\ -1.000000015625000e+000 \end{pmatrix}. \end{aligned}$$

Ako izračunamo relativne greške $\epsilon_k = \|x - x^{(k)}\|_\infty / \|x\|_\infty$, onda je

$$\begin{aligned} \epsilon_0 &= 5.000000000000000e-001 \\ \epsilon_1 &= 2.499999999999858e-002 \\ \epsilon_2 &= 1.249999999999973e-003 \\ \epsilon_3 &= 6.250000000029843e-005 \\ \epsilon_4 &= 3.125000000103739e-006 \\ \epsilon_5 &= 1.562499996055067e-007 \end{aligned}$$

Koristeći (1.43) i relaciju $b = Ax = (D - N)x$, lako izračunamo ponašanje pogreške $e^{(k)} = x^{(k)} - x$. Vrijedi

$$e^{(k+1)} = x^{(k+1)} - x = D^{-1}N(x^{(k)} - x) = D^{-1}Ne^{(k)}. \tag{1.44}$$

Primijetimo da izvedene relacije (1.42), (1.43), (1.44) vrijede za proizvoljni $n \geq 2$, gdje je

$$A = D - N, \quad D = \operatorname{diag}(a_{11}, \ldots, a_{nn}), \quad \prod_{i=1}^{n} a_{ii} \neq 0.$$

Sada iz (1.44) slijedi

$$e^{(k)} = D^{-1}Ne^{(k-1)} = (D^{-1}N)^2e^{(k-2)} = (D^{-1}N)^ke^{(0)}, \tag{1.45}$$

gdje je $e^{(0)} = x^{(0)} - x$ pogreška prve iteracije $x^{(0)}$. Uzimanjem proizvoljne vektorske norme $\|\cdot\|$ i odgovarajuće matrične norme, dobijemo

$$\|e^{(k)}\| \leq \|(D^{-1}N)^k\|\|e^{(0)}\| \leq \|D^{-1}N\|^k\|e^{(0)}\|. \tag{1.46}$$

Iz relacije (1.46) zaključujemo da će $e^{(k)}$ težiti nuli za svaki početni $x^{(0)}$ ako matrice $(D^{-1}N)^k$ teže nuli za $k \to \infty$. Na primjer, ako je $\|D^{-1}N\| < 1$, onda svakako $\|D^{-1}N\|^k \to 0$ za $k \to \infty$. Vidimo i da je nakon k–tog koraka greška $\|e^{(k)}\|$ barem $\|D^{-1}N\|^k$ puta manja od polazne $\|e^{(0)}\|$. Zapišimo ove zaključke u obliku propozicije.

Propozicija 1.7.1 *Ako je u rastavu $A = D - N$ u nekoj matričnoj normi ispunjeno $\|D^{-1}N\| < 1$, onda za svaku početnu iteraciju $x^{(0)}$ niz*

$$x^{(k+1)} = D^{-1}(b + Nx^{(k)}), \quad k = 0, 1, 2, \ldots \tag{1.47}$$

konvergira rješenju x sustava $Ax = b$.

Propozicija 1.7.2 *Ako je matrica A dijagonalno dominantna u smislu da je*

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

onda niz generiran Jacobijevom metodom sa proizvoljnim $x^{(0)}$ konvergira prema rješenju sustava $Ax = b$.

Dokaz: Lako je pokazati da je $\|D^{-1}N\|_\infty < 1$. ■

Primjer 1.7.2 *Neka je ...*

Primijetimo da smo u relacijama (1.41) $x_1^{(1)}$ i $x_2^{(1)}$ računali neovisno, pomoću $x_1^{(0)}$ i $x_2^{(0)}$. Ako pažljivije promotrimo formule (1.41), vidimo da bi imalo smisla u formuli za $x_2^{(1)}$ umjesto $x_1^{(0)}$ koristiti novu, upravo izračunatu (i vjerojatno bolju) vrijednost $x_1^{(1)}$. Općenito, formulu za $x^{(k+1)}$ modificiramo tako da u svakoj komponenti koristimo najsvježije izračunate vrijednosti. Na primjer, u slučaju $n = 4$ bismo imali

$$\begin{aligned}
x_1^{(k+1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)}) \\
x_2^{(k+1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - a_{24}x_4^{(k)}) \\
x_3^{(k+1)} &= \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - a_{34}x_4^{(k)}) \\
x_4^{(k+1)} &= \frac{1}{a_{44}}(b_4 - a_{41}x_1^{(k+1)} - a_{42}x_2^{(k+1)} - a_{43}x_3^{(k+1)})
\end{aligned} \quad . \tag{1.48}$$

Jasno je da je u općenitom slučaju

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}), \quad i = 1, \ldots, n. \tag{1.49}$$

Gornje formule definiraju *Gauss–Seidelovu iterativnu metodu.*
Da bismo bolje shvatili strukturu izvedenih formula, vratimo se primjeru $n = 4$. Uočimo da je u relaciji (1.48)

$$-\begin{pmatrix} 0 & a_{12} & a_{13} & a_{14} \\ 0 & 0 & a_{23} & a_{24} \\ 0 & 0 & 0 & a_{34} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \\ x_4^{(k)} \end{pmatrix} = \begin{pmatrix} -a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} \\ -a_{23}x_3^{(k)} - a_{24}x_4^{(k)} \\ -a_{34}x_4^{(k)} \\ 0 \end{pmatrix},$$

te da je vektor $x^{(k+1)}$ zapravo rješenje donje trokutastog sustava

$$\begin{pmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \\ x_4^{(k+1)} \end{pmatrix} = \begin{pmatrix} -a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} \\ -a_{23}x_3^{(k)} - a_{24}x_4^{(k)} \\ -a_{34}x_4^{(k)} \\ 0 \end{pmatrix}. \tag{1.50}$$

Stavimo

$$L = \begin{pmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}, \quad U = -\begin{pmatrix} 0 & a_{12} & a_{13} & a_{14} \\ 0 & 0 & a_{23} & a_{24} \\ 0 & 0 & 0 & a_{34} \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{1.51}$$

Vrijedi $A = L - U$ i, uz uvjet regularnosti matrice L, Gauss–Seidelovu metodu možemo zapisati kao

$$x^{(k+1)} = L^{-1}(b + Ux^{(k)}), \quad k = 1, 2, \ldots \tag{1.52}$$

Kao i u analizi Jacobijeve metode, dobijemo da je

$$e^{(k)} = (L^{-1}U)^k e^{(0)}, \quad k = 1, 2, \ldots \tag{1.53}$$

Propozicija 1.7.3 *Ako je A simetrična i pozitivno definitna matrica, onda niz generiran Gauss–Seidelovom metodom sa proizvoljnim početnim $x^{(0)}$ konvergira prema rješenju sustava $Ax = b$.*

Primjer 1.7.3 *Ilustrirajmo ponašanje Gauss–Seidelove metode na jednom primjeru. U pripremi.*

I Jacobijeva i Gauss–Seidelova metoda su generirane istom shemom: Matrica A je napisana u obliku $A = M - S$, gdje je M regularna matrica, i iteracije su dane s

$$x^{(k+1)} = M^{-1}(b + Sx^{(k)}), \quad k = 1, 2, \ldots \tag{1.54}$$

Pri tome je matrica M odabrana tako da ju je lako invertirati, tj. da je lako riješiti sustav

$$Mx^{(k+1)} = b + Sx^{(k)}.$$

U Jacobijevoj metodi je $M = D$ dijagonalna, a u Gauss–Seidelovoj $M = L$ je donje trokutasta matrica. Konvergencija prema rješenju sustava $Ax = b$ je osigurana ako je $\|M^{-1}N\| < 1$ za neku matričnu notmu $\|\cdot\|$.

1.7.2 Vježbe

1.8 Matematički software za problem $Ax = b$

U ovom poglavlju opisujemo BLAS i LAPACK, dvije trenutno najpopularnije biblioteke programa za rješavanje problema numeričke linearne algebre. Biblioteka BLAS (Basic Linear Algebra Subroutines, osnovni potprogrami za linearnu algebru) sadrži potprograme za elementarne operacije sa vektorima i matricama, dok je LAPACK opsežna biblioteka sa gotovim rješavačima za probleme kao što su linearni sustavi jednadžbi, problemi najmanjih kvadrata, problemi svojstvenih i singularnih vrijednosti za matrice i matrične parove. U LAPACK–u su sve elementarne operacije nad matricama i vektorima izvedene pomoću poziva biblioteke BLAS.
Obadvije biblioteke su dostupne preko Interneta, i na adresi `http://www.netlib.org` se mogu naći implementacije u FORTRAN–u i C–u.
Cilj ove sekcije nije detaljan opis tih biblioteka. Umjesto toga želimo čitatelju dati osnovnu informaciju i uputiti ga na izvore informacija. Za potrebe ove knjige smo sakupili neke bilbioteke programa, tako da ih čitatelj može lako kopirati na svoje računalo i koristiti.

1.8.1 Pregled biblioteke BLAS

Biblioteka BLAS je nastala iz potrebe da se elementarne operacije u linearnoj algebri na neki način standardiziraju. Korist od takvog standardiziranja je višestruka. Kao prvo, programiranje kompliciranih algoritama je pojednostavljeno: pri proračunu konstrukcije tankera ne treba trošiti vrijeme oko detalja kao što je npr. duljina vektora ili produkt dvije matrice. Nadalje, i najsloženiji proračuni su sastavljeni od takvih elementarnih operacija, pa je poželjno da su elementarne operacije implementirane na najefikasniji mogući način.

Optimalna implementacija algoritma kao što je na primjer množenje dvije matrice nije uvijek jednostavan posao. Za optimalni rezultat je potrebno poznavati detalje građe i funkcioniranja pojedinog računala. Sve to zahtijeva dodatne napore što onda poskupljuje izradu programa. Osim toga, prelaskom na drugo računalo se mijenjaju parametri optimalne implementacije i postupak treba ponoviti. Time je i održavanje dobrog, efikasnog programa skupo.

Činilo se razumnim odabrati jedan, ne preveliki, skup operacija koji je pak dovoljno velik da se iz njega može izvesti većina ostalih operacija. Za odabrane operacije se specificiraju procedure i to *(i)* imenom; *(ii)* listom ulaznih parametara, *(iii)* listom izlaznih vrijednosti. Takodjer se zahtijeva da su operacije implementirane za sve tipove strojnih brojeva. Kako je BLAS povijesno vezan za programski jezik FORTRAN, ti tipovi su REAL, DOUBLE PRECISION, COMPLEX i DOUBLE COMPLEX. U nekim operacijama, kao na primjer pri traženju komponente vektora sa najvećom apsolutnom vrijednosti, je rezultat cjelobrojan (INTEGER). Zbog preglednosti, uzeto je da je prvo slovo imena procedure oznaka tipa. Tako imena koja počinju sa I označavaju cjelobrojni tip (INTEGER), S označava jednostruku preciznost (SINGLE), D označava dvostruku preciznost (DOUBLE), C označava kompleksni tip (COMPLEX), a Z označava kompleksni tip u dvostrukoj preciznosti. (Implementacija BLAS–a u jeziku C ima sličnu strukturu, gdje je hijerarhija tipova ona iz C–a.)

Na ovaj način je pojednostavljeno programiranje, a optimizacija implementacije je svedena na mali skup standardiziranih procedura. Implementacija odabranih procedura se onda može prepustiti stručnjacima za *software*. Tržišna utakmica između proizvođača računala je dovela do toga da proizvođači nude tvornički optimirane verzije BLAS biblioteke za svoja računala.[10] Takve implementacije imaju vrijeme izvršavanja znatno kraće od "naivne" implementacije.

Operacije s vektorima: BLAS 1

Zbog jednostavnosti, u ovoj sekciji je sa (n,X,kx) zadan vektor x tipa REAL. To znači da su elementi vektora x na pozicijama $X(1 + (i-1)kx)$, $i = 1, \ldots, n$. Parametar kx je *korak* (engleski termin je *stride*) i njegovo korištenje je vezano za način spremanja vektora u memoriji računala.

Slično je sa (n,DX,kx) zadan vektor tipa DOUBLE PRECISION, sa (n,CX,kx) vektor tipa COMPLEX i sa (n,ZX,kx) vektor tipa DOUBLE COMPLEX.

Dat ćemo kratak pregled osnovnih operacija u paketu BLAS 1. Kako su i izvorni kod i dokumentacija dostupni preko Interneta, nećemo ulaziti u sve detalje, niti ćemo dati pregled cijelog paketa. U skupini operacija sa vektorima izdvajamo:

[10] Naravno, ne besplatno. Obično se takve biblioteke kupuju zajedno sa ostalom programskom podrškom.

- SCOPY kopira vektor x u vektor y.
  Deklarirana je sa SUBROUTINE SCOPY(n,X,kx,Y,ky). Varijante su:
  DCOPY(n,DX,kx,DY,ky),
  CCOPY(n,CX,kx,CY,ky),
  ZCOPY(n,ZX,kx,ZY,ky).

- SSWAP razmjenjuje sadžaj vektora x i y.
  Deklarirana je sa SUBROUTINE SSWAP(n,X,kx,Y,ky). Varijante su:
  DSWAP(n,DX,kx,DY,ky),
  CSWAP(n,CX,kx,CY,ky),
  ZSWAP(n,ZX,kx,ZY,ky).

- ISAMAX računa najmanji indeks i za kojeg je $|x_i| = \max_{1 \le j \le n} |x_j|$.
  Deklarirana je sa INTEGER FUNCTION ISAMAX(n,X,kx). Varijante su:
  INTEGER FUNCTION IDAMAX(n,DX,kx),
  INTEGER FUNCTION ICAMAX(n,CX,kx),
  INTEGER FUNCTION IZAMAX(n,ZX,kx).

- SNRM2 računa euklidsku duljinu $\|x\|_2$ vektora $x =$ (n,X,kx) tipa REAL.
  Deklarirana je sa REAL FUNCTION SNRM2(n,X,kx). Varijante su:
  DOUBLE PRECISION FUNCTION DNRM2(n,DX,kx),
  REAL FUNCTION SCNRM2(n,CX,kx),
  DOUBLE PRECISION FUNCTION DZNRM2(n,ZX,kx).

- SASUM računa ℓ_1 normu $\|x\|_1 = |x_1| + \cdots + |x_n|$.
  Deklarirana je sa REAL FUNCTION SASUM(n,X,kx). Varijante su:
  DOUBLE PRECISION FUNCTION DASUM(n,DX,kx),
  REAL FUNCTION SCASUM(n,CX,kx),
  DOUBLE PRECISION FUNCTION DZASUM(n,ZX,kx).

- SSCAL računa $a \cdot x$, gdje je a skalar istog tipa kao i vektor x. Rezultat je u vektoru x.
  Deklarirana je sa SUBROUTINE SSCAL(n,SA,SX,kx). Varijante su:
  DSCAL(n,DA,DX,kx),
  CSCAL(n,CA,CX,kx),
  ZSCAL(n,ZA,ZX,kx).

- SDOT računa skalarni produkt $(x, y) = y^* x$ vektora x i y.
  Deklarirana je sa REAL FUNCTION SDOT(n,X,kx,Y,ky). Varijante su:
  DOUBLE PRECISION FUNCTION DDOT(n,DX,kx,DY,ky),
  COMPLEX FUNCTION CDOTC(n,CX,kx,CY,ky) (računa $\sum_i \overline{x}_i y_i$),
  COMPLEX FUNCTION CDOTU(n,CX,kx,CY,ky) (računa $\sum_i x_i y_i$),
  DOUBLE COMPLEX FUNCTION ZDOTC(n,ZX,kx,ZY,ky) i
  DOUBLE COMPLEX FUNCTION (ZDOTU(n,ZX,kx, ZY,ky).

- SROT primijenjuje ravninsku rotaciju na vektore x i y. Preciznije, matrica $[x\ y]$ je zamijenjena sa $[c \cdot x + s \cdot y, c \cdot y - s \cdot x]$.
  Procedura je deklarirana sa SUBROUTINE SROT(n,X,kx,Y,ky,C,S). Varijante su:
  DROT(n,DX,kx,DY,ky,DC,DS),
  CSROT(n,CX,kx,CY,ky,C,S),
  ZDROT(n,ZX,kx,ZY,ky,DC,DS).

- SROTG računa Givensovu ravninsku rotaciju.
  Deklarirana je sa SUBROUTINE SROTG(A,B,C,S). Izlazni parametri C i S su izračunati tako da je
  $$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sqrt{a^2+b^2} \\ 0 \end{bmatrix}.$$
  Ulazna vrijednost A je na izlazu zamijenjena sa $\sqrt{a^2+b^2}$, a B sa ?? Varijante ove procedure su DROTG(DA,DB,DC,DS)

- SROTMG računa modificiranu Givensovu rotaciju.

- SROTM primijenjuje modificiranu Givensovu rotaciju.

Matrice i vektori: BLAS 2

Na drugom nivou biblioteke BLAS su operacije sa matricama i vektorima. Kako je cijela biblioteka dostupna *online*, mi ćemo ovdje spomenuti samo tri procedure. Slovo "x" na početku imena procedure stoji za "S", "D", "C", ili "Z".

- xGEMV računa izraz oblika
  $$y := \alpha Ax + \beta y, \text{ ili } y := \alpha A^\tau x + \beta y,$$
  gdje su α i β skalari, A je $m \times n$ matrica, a x i y su vektori odgovarajućih dimenzija.

- xTRMV računa produkt $y = Ax$ ili $y = A^\tau x$, gdje je A $n \times n$ gornje ili donje trokutasta matrica sa općenitom ili jediničnom dijagonalom.

- xTRSV rješava linearne sustave
  $$Ax = b, \text{ ili } A^\tau x = b,$$
  gdje je A gornje ili donje trokutasta matrica sa općenitom ili jediničnom dijagonalom, a b je vektor odgovarajuće dimenzije.

Čak i iz ovakvo kratkog prikaza se može vidjeti sa koliko pažnje je dizajnirana funkcionalnost procedura. Odabir pojedine varijante je omogućen podešavanjem ulaznih parametara. Posebno su napisane procedure za matrice sa specijlnom strukturom kao npr. simetrične i vrpčaste matrice. Za sve detalje čitatelja upućujemo na *online* izvor `http://www.netlib.org/blas`.

Operacije s matricama: BLAS 3

Treći nivo elementarnih operacija biblioteke BLAS implementira matrične operacije. Tek za ilustraciju funkcionalnosti procedura trećeg nivoa, opišimo tri najosnovnije.

- xGEMM računa izraz oblika

$$C := \alpha f(A)g(B) + \beta C,$$

gdje su α i β skalari, A, B, C matrice, $f(A) \in \{A, A^\tau\}$, $g(B) \in \{B, B^\tau\}$, pri čemu su dimenzije matrica takve da je izraz dobro definiran.

- xTRMM računa

$$B := \alpha f(A)B, \text{ ili } B := \alpha B f(A),$$

gdje je α skalar, A i B su matrice, pri čemu je A gornje ili donje trokutasta sa općenitom ili jediničnom dijagonalom, te $f(A) \in \{A, A^\tau\}$.

- xTRSM rješava po X jednadžbe

$$f(A)X = \alpha B, \text{ ili } Xf(A) = \alpha B,$$

gdje je A gornje ili donje trokutasta regularna matrica sa općenitom ili jediničnom dijagonalom, $f(A) \in \{A, A^\tau\}$, B i X su matrice odgovarajućih dimenzija.

Ostale procedure i kompletna dokumentacija se može naći na *Internetu*, na adresi `http://www.netlib.org`, ili na *Internet* stranicama ove knjige.

1.8.2 Pregled biblioteke LAPACK

LAPACK je kratica za *Linear Algebra PACKage*, paket (programa) za linearnu algebru. Osnovne značajke LAPACK–a su

1. Opsežnost. LAPACK sadrži preko 1000 potprograma sa algoritmima za rješavanje problema linearne algebre. Spomenimo da su na listi algoritmi za rješavanje linearnih sustava (sa pripadnim algoritmima za računanje LU faktorizacije, faktorizacije Choleskog, simetrične indefinitne faktorizacije), rješavanje problema najmanjih kvadrata (sa pripadnim algoritmima za računanje QR faktorizacije, generalizirane QR faktorizacije), rješavanje problema svojstvenih vrijednosti (simetrični problem, nesimetrični problem, generalizirani problemi za matrične parove), računanje obične i generalizirane dekompozicije singularnih vrijednosti, rješavanje matričnih jednadžbi (npr. Sylvesterove jednadžbe). Sve procedure su implementirane i za realne i za kompleksne tipove podataka. Posebni algoritmi su ponudjeni za specijalne tipove matrica kao što su npr. vrpčaste matrice.

2. Numerička pouzdanost. U LAPACK–u se posebna pažnja posvećuje numeričkoj stabilnosti. Uz dosta algoritama su ponudjene procedure za ocjenu greške u izračunatim rezultatima. Dani su teorijski okviri u kojima je ocjenjena stabilnost algoritama, tako da korisnik može dobiti dodatne informacije i o rezultatima ali i o svom matematičkom modelu. (Numerička nestabilnost može biti znak greške u modelu.)

3. Prenosivost i efikasnost na raznim računalima. Efikasnost biblioteke LAPACK je postignuta oslanjanjem na biblioteku BLAS. Drugim riječima, ako imamo BLAS optimiran za konkretni stroj, onda će LAPACK dobro iskoristiti resurse tog stroja. Prenosivost je osigurana strogim pridržavanjem standarda konkretnog programskog jezika (FORTRAN ili C).

4. Dobra dokumentiranost i jednostavno korištenje. Sve su procedure detaljno dokumentirane na jednoobrazan i unaprijed definiran način. Matematički detalji, kao i detalji implementacije su detaljno opisani u seriji tehničkih izvještaja (LAPACK Working Notes) koji su dostupni preko Interneta. Objavljen je i priručnik [?].

5. Neprestano usavršavanje. Istraživači koji sudjeluju u projektu LAPACK dolaze iz cijelog svijeta i neprestano rade na pronalaženju novih, boljih algoritama koji nakon strogih provjera postaju dijelovi LAPACK–a, bilo kao nove procedure, bilo kao zamjena za već postojeće. U vrijeme pisanja ovih redaka, u uporabi je LAPACK, verzija 3.0.

6. Kompletan izvorni kod je dostupan preko Interneta, zajedno sa MAKE datotekama koje same izvršavaju proces instaliranja. Za neka računala postoji gotova biblioteka koju samo treba kopirati koristeći FTP.

1.8.3 Rješavanje linearnih sustava pomoću LAPACK–a

Kako smo vidjeli u prethodnim sekcijama, rješavanje linearnog sustava jednadžbi ima tri faze: LU faktorizacija, rješavanje donje trokutastog i rješavanje gornje trokutastog sustava jednadžbi. Pogledajmo kako je to napravljeno u LAPACK–u. Zbog jednostavnosti, opisujemo procedure u jednostrukoj preciznosti. Dat ćemo i dijelove izvornog koda, kao ilustraciju dobre prakse programiranja – kako strukture programa tako i dokumentiranosti. Naravno, cijeli kod je dostupan *online*.
Trokutastu faktorizaciju računamo procedurom SGETRF. Pogledajmo opis parametara te procedure, kako je napisano u izvornom kodu:

```
SUBROUTINE SGETRF( M, N, A, LDA, IPIV, INFO )
```

```
*
*  -- LAPACK routine (version 3.0) --
*     Univ. of Tennessee, Univ. of California Berkeley, NAG Ltd.,
*     Courant Institute, Argonne National Lab, and Rice University
*     March 31, 1993
*
*     .. Scalar Arguments ..
      INTEGER            INFO, LDA, M, N
*     ..
*     .. Array Arguments ..
      INTEGER            IPIV( * )
      REAL               A( LDA, * )
*     ..
*
*  Purpose
*  =======
*
*  SGETRF computes an LU factorization of a general M-by-N matrix A
*  using partial pivoting with row interchanges.
*
*  The factorization has the form
*     A = P * L * U
*  where P is a permutation matrix, L is lower triangular with unit
*  diagonal elements (lower trapezoidal if m > n), and U is upper
*  triangular (upper trapezoidal if m < n).
*
*  This is the right-looking Level 3 BLAS version of the algorithm.
*
*  Arguments
*  =========
*
*  M       (input) INTEGER
*          The number of rows of the matrix A.  M >= 0.
*
*  N       (input) INTEGER
*          The number of columns of the matrix A.  N >= 0.
*
*  A       (input/output) REAL array, dimension (LDA,N)
*          On entry, the M-by-N matrix to be factored.
*          On exit, the factors L and U from the factorization
```

```
*          A = P*L*U; the unit diagonal elements of L are not stored.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A.  LDA >= max(1,M).
*
*  IPIV    (output) INTEGER array, dimension (min(M,N))
*          The pivot indices; for 1 <= i <= min(M,N), row i of the
*          matrix was interchanged with row IPIV(i).
*
*  INFO    (output) INTEGER
*          = 0:  successful exit
*          < 0:  if INFO = -i, the i-th argument had an illegal value
*          > 0:  if INFO = i, U(i,i) is exactly zero. The factorization
*                has been completed, but the factor U is exactly
*                singular, and division by zero will occur if it is used
*                to solve a system of equations.
*
*  =====================================================================
```

Dobro dokumentiranim programima ne treba dodatni komentar!
Rješavanje trokutastih sustava s matricama L i U izvršava procedure SGETRS u kojoj se poziva procedura za rješavanje trokutastih sustava i to sa jednom ili više desnih strana. Evo kako izgleda početak procedure SGETRS.

```
      SUBROUTINE SGETRS( TRANS, N, NRHS, A, LDA, IPIV, B, LDB, INFO )
*
*  -- LAPACK routine (version 3.0) --
*     Univ. of Tennessee, Univ. of California Berkeley, NAG Ltd.,
*     Courant Institute, Argonne National Lab, and Rice University
*     March 31, 1993
*
*     .. Scalar Arguments ..
      CHARACTER          TRANS
      INTEGER            INFO, LDA, LDB, N, NRHS
*     ..
*     .. Array Arguments ..
      INTEGER            IPIV( * )
      REAL               A( LDA, * ), B( LDB, * )
*     ..
*
*  Purpose
```

```
*  =======
*
*  SGETRS solves a system of linear equations
*     A * X = B  or  A' * X = B
*  with a general N-by-N matrix A using the LU factorization computed
*  by SGETRF.
*
*  Arguments
*  =========
*
*  TRANS   (input) CHARACTER*1
*          Specifies the form of the system of equations:
*          = 'N':  A * X = B  (No transpose)
*          = 'T':  A'* X = B  (Transpose)
*          = 'C':  A'* X = B  (Conjugate transpose = Transpose)
*
*  N       (input) INTEGER
*          The order of the matrix A.  N >= 0.
*
*  NRHS    (input) INTEGER
*          The number of right hand sides, i.e., the number of columns
*          of the matrix B.  NRHS >= 0.
*
*  A       (input) REAL array, dimension (LDA,N)
*          The factors L and U from the factorization A = P*L*U
*          as computed by SGETRF.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A.  LDA >= max(1,N).
*
*  IPIV    (input) INTEGER array, dimension (N)
*          The pivot indices from SGETRF; for 1<=i<=N, row i of the
*          matrix was interchanged with row IPIV(i).
*
*  B       (input/output) REAL array, dimension (LDB,NRHS)
*          On entry, the right hand side matrix B.
*          On exit, the solution matrix X.
*
*  LDB     (input) INTEGER
*          The leading dimension of the array B.  LDB >= max(1,N).
```

```
*
*  INFO    (output) INTEGER
*          = 0:  successful exit
*          < 0:  if INFO = -i, the i-th argument had an illegal value
*
*  =====================================================================
```

I konačno, prethodne dvije procedure su integrirane u rješavač sustava $Ax = b$, odnosno $AX = B$ ako imamo više desnih strana (matrica B umjesto vektora b).

```
      SUBROUTINE SGESV( N, NRHS, A, LDA, IPIV, B, LDB, INFO )
*
*  -- LAPACK driver routine (version 3.0) --
*     Univ. of Tennessee, Univ. of California Berkeley, NAG Ltd.,
*     Courant Institute, Argonne National Lab, and Rice University
*     March 31, 1993
*
*     .. Scalar Arguments ..
      INTEGER            INFO, LDA, LDB, N, NRHS
*     ..
*     .. Array Arguments ..
      INTEGER            IPIV( * )
      REAL               A( LDA, * ), B( LDB, * )
*     ..
*
*  Purpose
*  =======
*
*  SGESV computes the solution to a real system of linear equations
*     A * X = B,
*  where A is an N-by-N matrix and X and B are N-by-NRHS matrices.
*
*  The LU decomposition with partial pivoting and row interchanges is
*  used to factor A as
*     A = P * L * U,
*  where P is a permutation matrix, L is unit lower triangular, and U is
*  upper triangular.  The factored form of A is then used to solve the
*  system of equations A * X = B.
*
*  Arguments
*  =========
```

```
*
*  N       (input) INTEGER
*          The number of linear equations, i.e., the order of the
*          matrix A.  N >= 0.
*
*  NRHS    (input) INTEGER
*          The number of right hand sides, i.e., the number of columns
*          of the matrix B.  NRHS >= 0.
*
*  A       (input/output) REAL array, dimension (LDA,N)
*          On entry, the N-by-N coefficient matrix A.
*          On exit, the factors L and U from the factorization
*          A = P*L*U; the unit diagonal elements of L are not stored.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A.  LDA >= max(1,N).
*
*  IPIV    (output) INTEGER array, dimension (N)
*          The pivot indices that define the permutation matrix P;
*          row i of the matrix was interchanged with row IPIV(i).
*
*  B       (input/output) REAL array, dimension (LDB,NRHS)
*          On entry, the N-by-NRHS matrix of right hand side matrix B.
*          On exit, if INFO = 0, the N-by-NRHS solution matrix X.
*
*  LDB     (input) INTEGER
*          The leading dimension of the array B.  LDB >= max(1,N).
*
*  INFO    (output) INTEGER
*          = 0:  successful exit
*          < 0:  if INFO = -i, the i-th argument had an illegal value
*          > 0:  if INFO = i, U(i,i) is exactly zero.  The factorization
*                has been completed, but the factor U is exactly
*                singular, so the solution could not be computed.
*
*  =====================================================================
*
*     .. External Subroutines ..
      EXTERNAL           SGETRF, SGETRS, XERBLA
*     ..
```

```
*     .. Intrinsic Functions ..
      INTRINSIC          MAX
*     ..
*     .. Executable Statements ..
*
*     Test the input parameters.
*
      INFO = 0
      IF( N.LT.0 ) THEN
         INFO = -1
      ELSE IF( NRHS.LT.0 ) THEN
         INFO = -2
      ELSE IF( LDA.LT.MAX( 1, N ) ) THEN
         INFO = -4
      ELSE IF( LDB.LT.MAX( 1, N ) ) THEN
         INFO = -7
      END IF
      IF( INFO.NE.0 ) THEN
         CALL XERBLA( 'SGESV ', -INFO )
         RETURN
      END IF
*
*     Compute the LU factorization of A.
*
      CALL SGETRF( N, N, A, LDA, IPIV, INFO )
      IF( INFO.EQ.0 ) THEN
*
*        Solve the system A*X = B, overwriting B with X.
*
         CALL SGETRS( 'No transpose', N, NRHS, A, LDA, IPIV, B, LDB,
     $                INFO )
      END IF
      RETURN
*
*     End of SGESV
*
      END
```

Na kraju, napomenimo da u LAPACK–u postoje i procedure za sustave sa specijalnom strukturom (simetrični, pozitivno definitni, tridijagonalni, vrpčasti itd.). Više detalja

se može naći na adresi `http://www.netlib.org/lapack` i u *online* verziji ove knjige.

1.8.4 Dodatak: Vektori i matrice u programskim jezicima

1.8.5 Vježbe