

Numerička matematika

3. predavanje

Saša Singer

singer@math.hr

web.math.hr/~singer

PMF – Matematički odsjek, Zagreb

Sadržaj predavanja

- Rješavanje linearnih sustava:
 - Gaussove eliminacije.
 - Zamjene jednadžbi (redaka) — parcijalno pivotiranje.
 - Zamjene redaka i stupaca — potpuno pivotiranje.
 - Gaussove eliminacije u praksi — LR (LU) faktorizacija.
 - Veza Gaussovih eliminacija i LR faktorizacije.
 - Pivotiranje u LR (LU) faktorizaciji.
 - Pivotni rast kao mjera nestabilnosti.
 - Teorija perturbacije linearnih sustava (početak).

Informacije

Moja web stranica za Numeričku matematiku je

http://web.math.hr/~singer/num_mat/

Tamo su kompletna predavanja od prošle četiri godine, a stizat će i nova (kako nastaju). Prva 2 su još nesređena — onako kako ste ih vidjeli!

Skraćena verzija skripte — 1. dio (prvih 7 tjedana):

http://web.math.hr/~singer/num_mat/num_mat1.pdf

Skraćena verzija skripte — 2. dio (drugih 6 tjedana):

http://web.math.hr/~singer/num_mat/num_mat2.pdf

Informacije — demonstratori

Kolegij “Numerička matematika” ima čak **tri demonstratora**:

- **Mario Berljafa** — termin: **petak, 10–12**, u **Pr3**, **poželjna** najava mailom,
- **Anastasia Kruchinina** — termin: **ponedjeljak, 17–19**, **poželjna** najava mailom,
- **Melkior Ornik** — termin: **utorak, 16–18**, sastanak pred oglasnom pločom, **nužna** najava mailom.

Demosi lijepo **mole** da im se **najavite** mailom koji dan ranije!

- Njihove mail adrese nađete na **oglasnoj ploči**,
- ili se javite meni.

Rješavanje linearnih sustava

Općenito o linearnim sustavima — teorija

Neka je $\mathbb{F} = \mathbb{R}$ polje **realnih** brojeva (može i $\mathbb{F} = \mathbb{C}$).

Zadani su:

• (pravokutna) matrica $A \in \mathbb{F}^{m \times n}$ i vektor $b \in \mathbb{F}^m$.

Tražimo **rješenje** linearnog sustava

$$Ax = b.$$

Teorem **Kronecker–Capelli** kaže da linearni sustav $Ax = b$

- ima rješenje $x \in \mathbb{F}^n$ — **ako i samo ako** je rang matrice A , u oznaci r , **jednak** rang u proširene matrice $\hat{A} = [A \mid b]$,
- rješenje sustava je **jedinstveno** ako je $r = n$.

Znamo čak i malo više!

Linearni sustavi — teorija (nastavak)

Opće rješenje sustava $Ax = b$ (ako postoji) ima oblik

$$x = x_p + \mathcal{N}(A),$$

gdje je

- x_p jedno partikularno rješenje polaznog sustava $Ax = b$,
- $\mathcal{N}(A)$ je nul-potprostor od A , ili opće rješenje pripadnog homogenog sustava $Ax = 0$.

Iz teorema o rangu i defektu za matricu A

$$r + \dim \mathcal{N}(A) = n,$$

slijedi tvrdnja o jedinstvenosti rješenja:

$$\dim \mathcal{N}(A) = 0 \iff r = n.$$

Linearni sustavi — od teorije prema praksi

U praksi se najčešće rješavaju linearni sustavi $Ax = b$ kod kojih je matrica A kvadratna i regularna.

- A kvadratna — znači da je $m = n$ (A je reda n).
- A regularna — znači, na primjer, $\det A \neq 0$.

Iz teorema Kronecker–Capelli onda izlazi da

- rješenje x takvog sustava postoji i jedinstveno je.

⇒ ima smisla promatrati algoritme za računanje rješenja.

Nema smisla računati nešto što (možda) i ne postoji, ili nije jedinstveno (koje od mnogo rješenja računamo).

Oprez: To što unaprijed znamo da je A regularna

- ne znači da to vrijedi i numerički!

Kako naći rješenje? — Inverz matrice

Teorija (1). Možemo naći **inverz** matrice A , tj. matricu A^{-1}

• i **slijeva** pomnožiti sustav $Ax = b$ matricom A^{-1} .

Dobivamo

$$x = A^{-1}b.$$

Samo je **pitanje**: kako ćemo **izračunati** inverz A^{-1} ?

Zaključak: **Lakši** problem sveli smo na **teži** — u prijevodu, **pali smo s konja na magarca**.

Zašto? Jednostavno, zato što je

• j -ti stupac inverza = rješenje sustava $Ax = e_j$.

Dakle, za n stupaca od A^{-1} treba **riješiti** n linearnih sustava. A krenuli smo od **jednog** (sustava)! **Ne tako!**

Kako naći rješenje? — Cramerovo pravilo

Teorija (2). Iz linearne algebre znate za Cramerovo pravilo:

• j -ta komponenta rješenja sustava je

$$x_j = \frac{\det A_j}{\det A}, \quad j = 1, \dots, n,$$

• pri čemu je matrica A_j jednaka matrici A ,

• osim što je j -ti stupac u A_j zamijenjen desnom stranom b .

Treba još “samo” izračunati determinante — i to $n + 1$ njih.
A kako ćemo to?

“Klasični” odgovor: pa ... recimo, Laplaceovim razvojem.

Jao, jao ... Bilo kako, samo ne tako!

Kako naći rješenje? — Zaboravite Cramera

Zašto? Laplaceovim razvojem dobijemo

- “samo” $n!$ pribrojnika u **svakoj** determinanti,
- a **svaki** pribrojnik je produkt od n faktora.

Prava “sitnica”. I tako to, još $n + 1$ puta ...

Zaključak: Ako determinante **računamo** na ovaj način,

- složenost **Cramerovog** pravila za rješavanja linearnog sustava je **eksponencijalna** u n (dokažite to!)
- i **nikad** se ne koristi kao metoda **numeričkog** rješavanja.

Zaboravite determinante i Cramera — finale

Komentar: Determinante možemo računati i puuuno brže,

- tako da matricu svedemo na trokutasti oblik,
- postupkom sličnim Gausovim eliminacijama.

Naime, determinanta trokutaste matrice (gornje ili donje) je

- produkt dijagonalnih elemenata,

pa se lako računa!

No, isti postupak eliminacija koristimo i za

- rješavanje “cijelog” linearnog sustava $Ax = b$,
- i to samo jednom, a ne $n + 1$ puta.

Dakle, Cramerovo pravilo se ne isplati ni kad ovako računamo determinante.

Kako naći rješenje? — Gaussove eliminacije!

Najjednostavnija metoda za rješavanje linearnih sustava su

- Gaussove eliminacije, odnosno
- slične metode svodenja na trokutastu formu.

Ideja: Sustav $Ax = b$ se ekvivalentnim transformacijama svodi na sustav oblika

$$Rx = y,$$

gdje je

- R trokutasta matrica (recimo, gornja),
- iz kojeg se lako, tzv. povratnom supstitucijom, nalazi rješenje.

Oznaka R — “right” (desna) = gornja trokutasta matrica.

Ekvivalentne transformacije sustava

Ekvivalentne transformacije sustava su

- one koje ne mijenjaju rješenje sustava.

Standardne ekvivalentne transformacije su (v. LA):

- zamjena poretka jednačbi (nužno!),
- množenje jednačbe brojem različitim od nule,
 - ova transformacija “skaliranja” se obično ne koristi, ili se vrlo pažljivo koristi — za povećanje stabilnosti,
- množenje jedne jednačbe nekim brojem i dodavanje drugoj jednačbi (ključno!),
= dodavanje linearne kombinacije preostalih jednačbi, s tim da uzmemo samo jednu preostalu jednačbu.

Gaussove eliminacije — komentari

Par komentara, prije detaljnog opisa metode.

Gaussove eliminacije su metoda **direktnog** transformiranja linearnog sustava $Ax = b$, zajedno s desnom stranom b .

Možemo ih implementirati i tako da se desna strana b **ne transformira** istovremeno kad i matrica A .

- Tada se formiraju dvije matrice L i R takve da je $A = LR$, gdje je R **gornja** trokutasta matrica iz Gaussovih eliminacija, a L je **donja** trokutasta matrica.
- Tako implementirane Gaussove eliminacije zovemo **LR** (ili **LU**) **faktorizacija** matrice A — **standard** u praksi.
- Ovaj pristup je posebno zgodan kad imamo **više desnih** strana za **isti** A .

Gaussove eliminacije — komentari (nastavak)

U praksi se koriste za “opće”, ali ne pretjerano velike matrice (n u tisućama), ili za sustave s tzv. “vrpčastom” strukturom.

Složenost: polinomna i to kubna, tj. $O(n^3)$, što je sporo za još veće sustave. Za njih se koriste iterativne metode.

Mnogi sustavi imaju specijalna svojstva koja koristimo za brže i/ili točnije rješenje. Na primjer,

- za simetrične, pozitivno definitne matrice koristi se “simetrična” LR faktorizacija, tzv. faktorizacija Choleskog,
- za dijagonalno dominantne sustava ne treba pivotiranje,
- za vrpčaste, posebno trodijagonalne matrice, algoritam se drastično skraćuje (v. kubična spline interpolacija).

Gaussove eliminacije

Gaussove eliminacije — algoritam

Označimo $A^{(1)} := A$, $b^{(1)} := b$ na početku **prvog** koraka.

U skraćenoj notaciji, **bez** pisanja nepoznanica x_i , linearni sustav $Ax = b$ možemo zapisati **proširenom** matricom, kao

$$\left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array} \right] .$$

Svođenje na **trokutastu** formu radimo u $n - 1$ **koraka**.

Gaussove eliminacije — algoritam (nastavak)

1. korak.

- U prvom stupcu matrice $A^{(1)}$ poništimo sve elemente, osim prvog.

Kako se to radi?

Ako je element $a_{11}^{(1)} \neq 0$, onda redom, možemo

- od i -te jednačbe oduzeti
- prvu jednačbu pomnoženu s

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \quad i = 2, \dots, n.$$

Pritom se prva jednačba se ne mijenja.

Gaussove eliminacije — algoritam (nastavak)

Prva jednađba — kao **redak** proširene matrice $[A^{(1)} \mid b^{(1)}]$, je

$$a_{11}^{(1)} \quad a_{12}^{(1)} \quad \cdots \quad a_{1n}^{(1)} \quad \left| \quad b_1^{(1)} \right. .$$

Polazna i -ta jednađba — pisana na isti naćin, za $i = 2, \dots, n$

$$a_{i1}^{(1)} \quad a_{i2}^{(1)} \quad \cdots \quad a_{in}^{(1)} \quad \left| \quad b_i^{(1)} \right. .$$

Nova i -ta jednađba — pisana na isti naćin, za $i = 2, \dots, n$

$$a_{i1}^{(2)} \quad a_{i2}^{(2)} \quad \cdots \quad a_{in}^{(2)} \quad \left| \quad b_i^{(2)} \right. .$$

Relacije za **nove** elemente su

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)}, \quad b_i^{(2)} = b_i^{(1)} - m_{i1}b_1^{(1)},$$

za $j = 1, \dots, n$, $i = 2, \dots, n$. **Prvi** redak ($i = 1$) ostaje **isti**.

Gaussove eliminacije — algoritam (nastavak)

Iz ovih relacija za **nove** elemente (prepisane još jednom)

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)}, \quad b_i^{(2)} = b_i^{(1)} - m_{i1}b_1^{(1)},$$

vidimo da su **multiplikatori**

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \quad i = 2, \dots, n.$$

odabrani upravo tako da je $a_{i1}^{(2)} = 0$, za $i = 2, \dots, n$.

Dakle, nakon **prvog** koraka dobivamo proširenu matricu $[A^{(2)} \mid b^{(2)}]$ u kojoj

- **prvi stupac** ima **nule ispod** dijagonale, tj. **gornju** trokutastu formu.

Gaussove eliminacije — algoritam (nastavak)

Time smo dobili **ekvivalentni** linearni sustav $A^{(2)}x = b^{(2)}$ s **proširenom** matricom

$$\left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array} \right] .$$

Postupak **ponišćavanja** možemo nastaviti s **drugim** stupcem matrice $A^{(2)}$ — na isti naćin.

Ako je $a_{22}^{(2)} \neq 0$, biramo faktore m_{i2} tako da poništimo sve elemente **drugog** stupca **ispod** dijagonale. I tako redom.

Gaussove eliminacije — algoritam (nastavak)

Općenito, k -ti korak izgleda ovako, za $k = 1, \dots, n - 1$:

- 🔴 iz proširene matrice $[A^{(k)} \mid b^{(k)}]$ dobivamo novu proširenu matricu $[A^{(k+1)} \mid b^{(k+1)}]$,
- 🔴 tako da **poništimo** sve elemente **ispod** dijagonale u k -tom **stupcu** matrice $A^{(k)}$.

Relacije za **nove** elemente koje treba **izračunati** u matrici $A^{(k+1)}$ i vektoru $b^{(k+1)}$ su

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)},$$

za $i, j = k + 1, \dots, n$, a **multiplikatori** m_{ik} su

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k + 1, \dots, n.$$

Gaussove eliminacije — algoritam (nastavak)

Prvih k redaka u $[A^{(k+1)} \mid b^{(k+1)}]$ ostaju isti kao u $[A^{(k)} \mid b^{(k)}]$.

Konačno, ako su svi $a_{ii}^{(i)} \neq 0$, za $i = 1, \dots, n - 1$, završni linearni sustav $[A^{(n)} \mid b^{(n)}]$, ekvivalentan polaznom, je

$$\left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & a_{nn}^{(n)} & b_n^{(n)} \end{array} \right].$$

Dobili smo gornju trokutastu matricu $R = A^{(n)}$ (nule u donjem trokutu matrice R ne pišemo).

Gaussove eliminacije — algoritam (nastavak)

Uz pretpostavku da je $a_{nn}^{(n)} \neq 0$, ovaj se linearni sustav lako rješava **povratnom supstitucijom**

$$x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}},$$

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

Pitanje: Ako je A kvadratna i regularna,

🔴 moraju li **svi** elementi $a_{ii}^{(i)}$ biti **različiti** od **nule**?

To je **nužno** (i dovoljno) da algoritam “**prođe**” u **ovom** obliku.

Gaussove eliminacije — primjedba na algoritam

Odgovor: Ne!

Primjer. Linearni sustav $Ax = b$ s proširenom matricom

$$[A | b] = \left[\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & 0 & 1 \end{array} \right]$$

je **regularan** ($\det A = -1$), sustav ima **jedinstveno** rješenje $x_1 = x_2 = 1$,

- a ipak ga **ne možemo** riješiti Gausovim eliminacijama,
- ako **ne mijenjamo poredak** jednadžbi.

Zaključak: Moramo dozvoliti **promjenu poretka** jednadžbi.

Gaussove eliminacije s pivotiranjem

Pitanje: Dozvolimo li **promjene poretka** jednadžbi — tzv. “**pivotiranje**” u **stupcu** kojeg sređujemo,

- može li se Gaussovima eliminacijama s **pivotiranjem** riješiti **svaki** sustav kojemu je matrica kvadratna i regularna?

Odgovor: Ako dozvolimo **pivotiranje**

- zamjenom “**ključne**” jednadžbe i **bilo koje** druge koja ima **ne-nula** element (u tom **stupcu**),
- Gaussovima eliminacijama **rješiv** je **svaki** regularni kvadratni linearni sustav.

Objašnjenje: Ako u **prvom** stupcu **nemamo ne-nula** elemenata, matrica je **singularna**. Isto vrijedi i za **svaki** sljedeći **korak** (Laplaceov razvoj determinante!).

Gaussove eliminacije — kako pivotiramo?

Pitanje: Kako vršiti **pivotiranje**, tj. **zamjene** jednadžbi?

- Zamjenom “**ključne**” jednadžbe i **bilo koje** druge koja ima **ne-nula** element (u tom **stupcu**)?

Odgovor: Tu je **ključna** razlika između **egzaktnog** i **približnog** računanja (kad imamo greške zaokruživanja).

- U **teoriji** — kod **egzaktnog** računanja, **dovoljno** je naći **bilo koji ne-nula** element (u tom stupcu).
- U **praksi** — kad računamo **približno**, to **može** dovesti do potpuno **pogrešnog** rezultata.

Jedna jedina operacija može **upropastiti** rezultat!

- Postoji u **puno bolja** strategija za **pivotiranje**, kojom se to (barem dijelom) može **izbjeći**.

Gaussove eliminacije — primjer

Primjer. Zadan je linearni sustav

$$0.0001 x_1 + x_2 = 1$$

$$x_1 + x_2 = 2.$$

Matrica sustava je regularna, pa postoji jedinstveno rješenje

$$x_1 = 1.0001, \quad x_2 = 0.9999.$$

Riješimo taj sustav “**računalom**” koje ima 4 decimalne znamenke mantise i 2 znamenke eksponenta.

Uočiti: Broj 0.0001 je “**mali**”, ali **nije nula**. Po teoriji,

🔴 možemo ga uzeti kao **prvi** (ili bilo koji) **ne-nula** element.

Gaussove eliminacije — primjer (nastavak)

Sustav zapisan u takvom “računalu” pamti se kao

$$1.000 \cdot 10^{-4} x_1 + 1.000 \cdot 10^0 x_2 = 1.000 \cdot 10^0$$

$$1.000 \cdot 10^0 x_1 + 1.000 \cdot 10^0 x_2 = 2.000 \cdot 10^0.$$

Množenjem prve jednadžbe s -10^4 i dodavanjem drugoj, dobivamo **novu drugu** jednadžbu

$$(1.000 \cdot 10^0 - 1.000 \cdot 10^4) x_2 = 2.000 \cdot 10^0 - 1.000 \cdot 10^4.$$

Oduzimanje u računalu se vrši tako da manji eksponent postane jednak većem, a mantisa se denormalizira. Dobivamo

$$\begin{aligned} 1.000 \cdot 10^0 &= 0.100 \cdot 10^1 = 0.010 \cdot 10^2 = 0.001 \cdot 10^3 \\ &= 0.000|1 \cdot 10^4. \end{aligned}$$

Gaussove eliminacije — primjer (nastavak)

Za zadnju jedinicu **nema mjesta** u mantisi, pa je mantisa postala 0, tj. **prvi** broj “nema” utjecaja na rezultat. Slično se dogodi i s **desnom** stranom (i 2 je “zanemariv” prema 10^4).

Dakle, **nova druga** jednačba glasi

$$-1.000 \cdot 10^4 x_2 = -1.000 \cdot 10^4.$$

Rješenje ove jednačbe je očito $x_2 = 1.000 \cdot 10^0$. Uvrštavanjem u **prvu** jednačbu, dobivamo

$$\begin{aligned} 1.000 \cdot 10^{-4} x_1 &= 1.000 \cdot 10^0 - 1.000 \cdot 10^0 \cdot 1.000 \cdot 10^0 \\ &= 0.000 \cdot 10^0, \end{aligned}$$

pa je $x_1 = 0$, što **nije niti približno točan rezultat**.

Gaussove eliminacije — primjer (nastavak)

Razlog za **ogromnu** relativnu grešku (100%):

- **prvu** jednadžbu množimo **velikim** brojem -10^4 (po apsolutnoj vrijednosti) i **odajemo drugoj**,
- što “**uništava**” **drugu** jednadžbu.

Drugim riječima, utjecaj **polazne druge** jednadžbe

- postaje **zanemariv** u **novoj drugoj** jednadžbi.

U **polaznoj drugoj** je moglo pisati “bilo što”!

Gaussove eliminacije s pivotiranjem — primjer

Promijenimo li poredak jednadžbi, dobivamo

$$1.000 \cdot 10^0 x_1 + 1.000 \cdot 10^0 x_2 = 2.000 \cdot 10^0$$

$$1.000 \cdot 10^{-4} x_1 + 1.000 \cdot 10^0 x_2 = 1.000 \cdot 10^0.$$

Množenjem prve jednadžbe s -10^{-4} i dodavanjem drugoj, dobivamo novu drugu jednadžbu

$$(1.000 \cdot 10^0 - 1.000 \cdot 10^{-4}) x_2 = 1.000 \cdot 10^0 - 2.000 \cdot 10^{-4}.$$

Ovdje nema oduzimanja — drugi broj s 10^{-4} je “zanemariv” prema 1. Dakle, nova druga jednadžba sad glasi

$$1.000 \cdot 10^0 x_2 = 1.000 \cdot 10^0.$$

Gaussove eliminacije s pivotiranjem — primjer

Ponovno dobivamo rješenje $x_2 = 1.000 \cdot 10^0$. Međutim, uvrštavanjem u **prvu** jednadžbu dobivamo

$$\begin{aligned} 1.000 \cdot 10^0 x_1 &= 2.000 \cdot 10^0 - 1.000 \cdot 10^0 \cdot 1.000 \cdot 10^0 \\ &= 1.000 \cdot 10^0, \end{aligned}$$

pa je $x_1 = 1.000 \cdot 10^0$, što je **točan** rezultat — **korektno zaokruženo** egzaktno rješenje na četiri decimalne znamenke!

Razlog za **vrlo malu** relativnu grešku:

- **prvu** jednadžbu sad množimo **malim** brojem -10^{-4} (po apsolutnoj vrijednosti) i **dodajemo drugoj**,
- što **nema utjecaja** na **drugu** jednadžbu — tj. ovdje **nema** “**uništavanja**” jednadžbi.

Gaussove eliminacije s pivotiranjem — primjer

Kao i ranije, u koraku eliminacije,

- (bivša) druga jednadžba nema utjecaja na (bivšu) prvu.

Međutim, nakon zamjene

- prva jednadžba (bivša druga) ostaje netaknuta u prvom koraku eliminacije i uredno utječe na rješenje.

Zaključak: Sigurno nije dovoljno uzeti

- prvi (bilo koji) ne-nula element u stupcu

kao ključni element za eliminacije,

- jer možemo dobiti potpuno pogrešan rezultat.

Gaussove eliminacije s pivotiranjem — primjer

Primjer. Usporedimo izračunata rješenja sustava

$$\varepsilon x_1 + x_2 = 1$$

$$x_1 + x_2 = 2,$$

za $\varepsilon = 10^{-1}, \dots, 10^{-25}$, Gaussovima eliminacijama **bez** zamjena i **sa zamjenom** poretka jednačbi, u aritmetici **računala**.

Računanjem u **najvećoj** mogućoj preciznosti (**extended**) dobivamo sljedeću tablicu.

U tablici je x_2 naveden samo **jednom** — jer ga obje metode izračunaju **jednako** (i točno)!

U prvom stupcu pišu samo **eksponenti** p , pri čemu je $\varepsilon = 10^p$.

Gaussove eliminacije s pivotiranjem — primjer

p	x_1 bez pivotiranja	x_1 s pivotiranjem	x_2
-4	1.00010001000100000	1.00010001000100010	0.99989998999899990
-5	1.00001000010000200	1.00001000010000100	0.99998999989999900
-6	1.00000100000099609	1.00000100000100000	0.99999899999900000
-7	1.00000009999978538	1.00000010000001000	0.99999989999990000
	⋮	⋮	
-17	0.99746599868666408	1.00000000000000001	0.99999999999999999
-18	0.97578195523695399	1.00000000000000000	1.00000000000000000
-19	1.08420217248550443	1.00000000000000000	1.00000000000000000
-20	0.00000000000000000	1.00000000000000000	1.00000000000000000

Gaussove eliminacije s parcijalnim pivotiranjem

Pivotni element uobičajeno se bira korištenjem **parcijalnog pivotiranja**

● pivotni element je **po apsolutnoj vrijednosti najveći** u “ostatku” **stupca** — na glavnoj dijagonali ili ispod nje, tj. ako je u k -tom koraku

$$|a_{rk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|,$$

onda ćemo **zamijeniti** r -ti i k -ti redak i početi korak eliminacije elemenata k -tog stupca.

Gaussove eliminacije s parcijalnim pivotiranjem

Motivacija: elementi “ostatka” linearnog sustava koje treba izračunati u matrici $A^{(k+1)}$ u k -tom koraku transformacije su

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)},$$

za $i, j = k + 1, \dots, n$, a multiplikatori m_{ik} su

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k + 1, \dots, n.$$

Ako je multiplikator m_{ik} **velik**, u aritmetici pomičnog zareza može doći do **kraćenja** najmanje značajnih znamenki $a_{ij}^{(k)}$, tako da izračunati $a_{ij}^{(k+1)}$ može imati **veliku** relativnu grešku.

Gaussove eliminacije s parcijalnim pivotiranjem

Sasvim općenito, ideja pivotiranja je **minimizirati korekcije elemenata** pri prijelazu s $A^{(k)}$ na $A^{(k+1)}$. Dakle, multiplikatori trebaju biti **što manji**.

Za multiplikatore kod parcijalnog pivotiranja vrijedi

$$|m_{ik}| \leq 1, \quad i = k + 1, \dots, n.$$

U praksi, parcijalno pivotiranje **funkcionira izvrsno**, ali matematičari su konstruirali primjere kad ono “**nije savršeno**”.

Gaussove eliminacije s potpunim pivotiranjem

Osim parcijalnog pivotiranja, može se provoditi i **potpuno pivotiranje**. U k -tom koraku, bira se maksimalni element u cijelom “ostatku” matrice $A^{(k)}$, a ne samo u k -tom stupcu.

Ako je u k -tom koraku

$$|a_{rs}^{(k)}| = \max_{k \leq i, j \leq n} |a_{ij}^{(k)}|,$$

onda ćemo zamijeniti r -ti i k -ti redak, s -ti i k -ti stupac i početi korak eliminacije elemenata k -tog stupca.

Oprez: zamjenom s -tog i k -tog stupca zamijenili smo ulogu varijabli x_s i x_k .

Ovo **nisu jedine** mogućnosti pivotiranja kod rješavanja linearnih sustava.

GE s parcijalnim pivotiranjem

— algoritam i složenost

Algoritam

Gaussove eliminacije s parcijalnim pivotiranjem

```
/* Trokutasta redukcija */
```

```
za k = 1 do n - 1 radi {
```

```
  /* Nađi maksimalni |element| u ostatku stupca */
```

```
  max_elt = 0.0;
```

```
  ind_max = k;
```

```
  za i = k do n radi {
```

```
    ako je  $|A[i, k]| > \text{max\_elt}$  onda {
```

```
      max_elt =  $|A[i, k]|$ ;
```

```
      ind_max = i;
```

```
    }
```

```
  };
```

Algoritam (nastavak)

```
ako je max_elt > 0.0 onda {
  /* Matrica ima ne-nula element u stupcu */
  ako je ind_max <> k onda {
    /* Zamijeni k-ti i ind_max-ti redak */
    za j = k do n radi {
      temp = A[ind_max, j];
      A[ind_max, j] = A[k, j];
      A[k, j] = temp;
    };
    temp = b[ind_max];
    b[ind_max] = b[k];
    b[k] = temp;
  };
};
```

Algoritam (nastavak)

```
za i = k + 1 do n radi {
    /* Izračunaj multiplikator */
    mult = A[i, k] / A[k, k];
    /* Ažuriraj i-ti redak */
    za j = k + 1 do n radi {
        A[i, j] = A[i, j] - mult * A[k, j];
    };
    b[i] = b[i] - mult * b[k];
}
inače
    /* Matrica je singularna, STOP */
};
```

Algoritam (nastavak)

```
/* Povratna supstitucija */  
  
/* Rješenje x izračunaj u b */  
b[n] = b[n] / A[n, n];  
za i = n - 1 do 1 radi {  
    sum = b[i];  
    za j = i + 1 do n radi {  
        sum = sum - A[i, j] * b[j];  
    };  
    b[i] = sum / A[i, i];  
};
```

Složenost algoritma

Prebrojimo sve **aritmetičke operacije** ovog algoritma:

U prvom koraku **trokutaste redukcije** obavlja se:

- $n - 1$ dijeljenje — računanje **mult**,
- $n(n - 1)$ množenje — za **svaki** od $n - 1$ redaka imamo:
 - $n - 1$ množenje za računanje elemenata matrice A ;
 - **jedno** množenje za računanje elementa vektora b ,
- $n(n - 1)$ oduzimanje — u istoj naredbi gdje i prethodna množenja.

Na sličan način zaključujemo da se u k -tom koraku obavlja:

- $n - k$ dijeljenja,
- $(n - k + 1)(n - k)$ množenja i $(n - k + 1)(n - k)$ oduzimanja.

Složenost algoritma (nastavak)

Ukupno, u k -tom koraku imamo

$$n - k + 2(n - k + 1)(n - k) = 2(n - k)^2 + 3(n - k)$$

aritmetičkih operacija.

Broj koraka k varira od 1 do $n - 1$, pa je ukupan broj operacija potrebnih za svođenje na trokutastu formu jednak

$$\begin{aligned} \sum_{k=1}^{n-1} [2(n - k)^2 + 3(n - k)] &= \sum_{k=1}^{n-1} (2k^2 + 3k) \\ &= \frac{1}{6}(4n^3 + 3n^2 - 7n). \end{aligned}$$

Druga suma se dobije se iz prve zamjenom indeksa $n - k \mapsto k$.

Složenost algoritma (nastavak)

Potpuno istim zaključivanjem dobivamo da u **povratnoj supstituciji** ima:

- $(n - 1) n/2$ množenja i $(n - 1) n/2$ zbrajanja,
- n dijeljenja,

što je zajedno tačno n^2 operacija.

Dakle, **ukupan broj** operacija u Gaussovima eliminacijama je

$$OP(n) = \frac{1}{6}(4n^3 + 9n^2 - 7n),$$

što je približno $2n^3/3$, za malo veće n .

LR faktorizacija

LR faktorizacija

U praksi se linearni sustavi najčešće rješavaju korištenjem **LR faktorizacije** — A faktoriziramo kao

$$A = LR,$$

pri čemu je

- L donja trokutasta matrica s jedinicama na dijagonali,
- R gornja trokutasta.

Matrica L je **regularna**, jer je $\det L = 1$, pa regularnost matrice A povlači i regularnost matrice R , jer je

$$\det A = \det L \cdot \det R = \det R.$$

LR faktorizacija (nastavak)

Ako znamo LR faktorizaciju od A , onda linearni sustav $Ax = b$ postaje

$$LRx = b.$$

Uz oznaku $y = Rx$, sustav $LRx = b$ svodi se na dva sustava

$$Ly = b, \quad Rx = y.$$

Prednost LR faktorizacije:

- rješavaju se dva **jednostavna** sustava,
- desna strana b **ne transformira** se istovremeno s matricom A , pa promjena **desne strane** košta samo $O(n^2)$ operacija.

LR faktorizacija (nastavak)

Oba sustava se lako rješavaju:

- prvi $Ly = b$ — supstitucijom unaprijed

$$y_1 = b_1,$$

$$y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j, \quad i = 2, \dots, n,$$

- drugi $Rx = y$ — povratnom supstitucijom

$$x_n = \frac{y_n}{r_{nn}},$$

$$x_i = \frac{1}{r_{ii}} \left(y_i - \sum_{j=i+1}^n r_{ij} x_j \right), \quad i = n-1, \dots, 1.$$

LR faktorizacija (nastavak)

Kako izračunati elemente l_{ij} i r_{ij} matrica L i R ?

- Iskoristimo poznatu strukturu L i R
- i činjenicu da je $A = L \cdot R$.

Dobivamo:

$$a_{ij} = \sum_{k=1}^{\min\{i,j\}} l_{ik} r_{kj}, \quad \text{uz } l_{ii} = 1.$$

Iz ovih n^2 jednažbi računamo, redom, one elemente matrica L i R koje možemo.

- Za $i = 1$, zbog $l_{11} = 1$, dobivamo prvi redak matrice R .
- Zatim, za $j = 1$, dobivamo prvi stupac matrice L , jer znamo r_{11} . Itd ...

LR faktorizacija (nastavak)

Tako dobivamo rekurziju za elemente matrica L i R

$$r_{1j} = a_{1j}, \quad j = 1, \dots, n,$$

$$l_{j1} = \frac{a_{j1}}{r_{11}}, \quad j = 2, \dots, n,$$

za $i = 2, \dots, n$:

$$r_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} r_{kj}, \quad j = i, \dots, n,$$

$$l_{ji} = \frac{1}{r_{ii}} \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} r_{ki} \right), \quad j = i + 1, \dots, n.$$

U zadnjem koraku, za $i = n$, računamo samo r_{nn} .

LR faktorizacija (nastavak)

Ako je $r_{ii} \neq 0$, za $i = 1, \dots, n - 1$, onda iz prethodnih relacija

- možemo izračunati **sve netrivialne** elemente matrica L i R .

Drugim riječima,

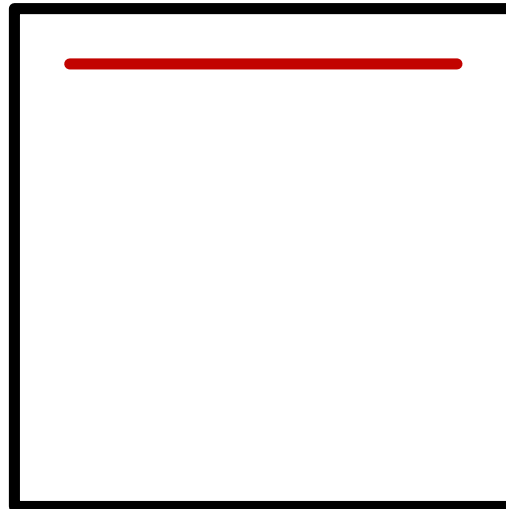
- imamo **egzistenciju** i **jedinstvenost** matrica L i R .

Primijetite, $r_{nn} \neq 0$ treba samo za **povratnu** supstituciju.

Pitanje: Kojim se **redom** računaju elementi?

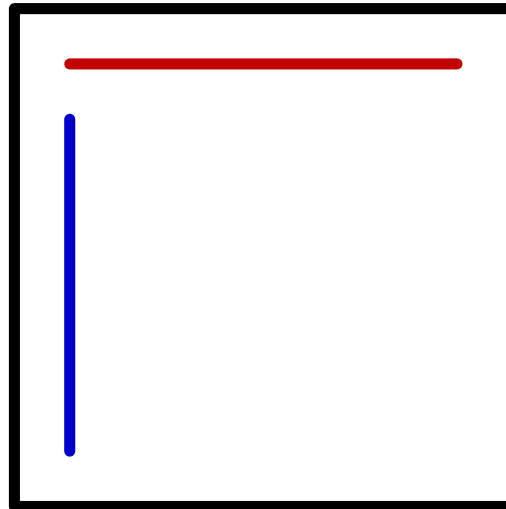
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



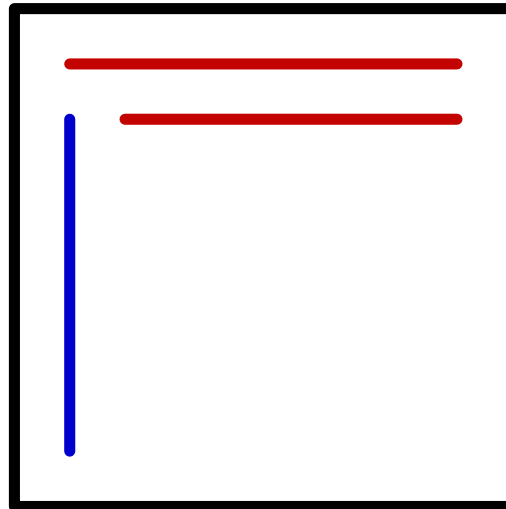
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



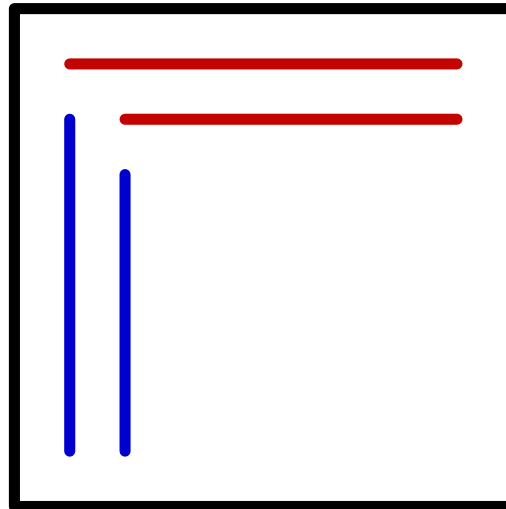
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



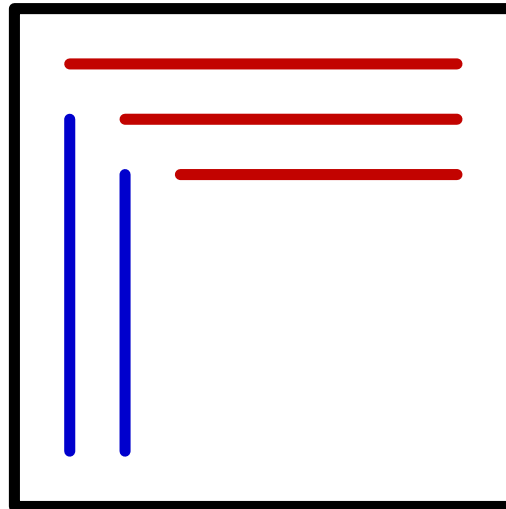
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



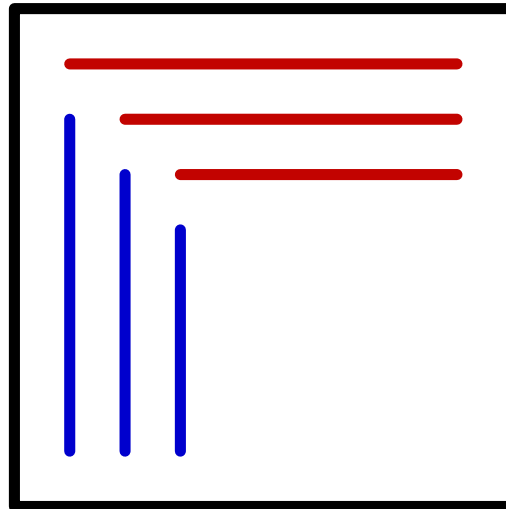
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



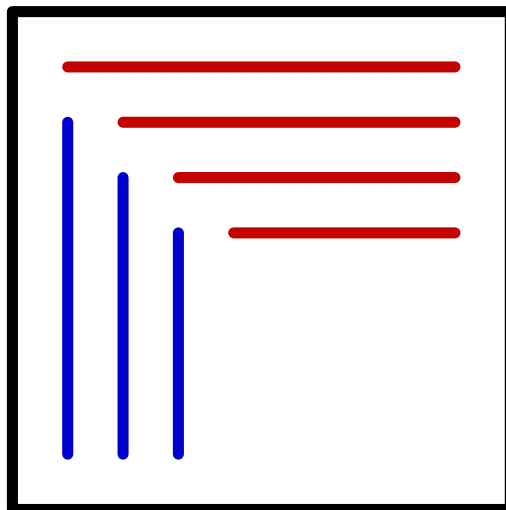
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



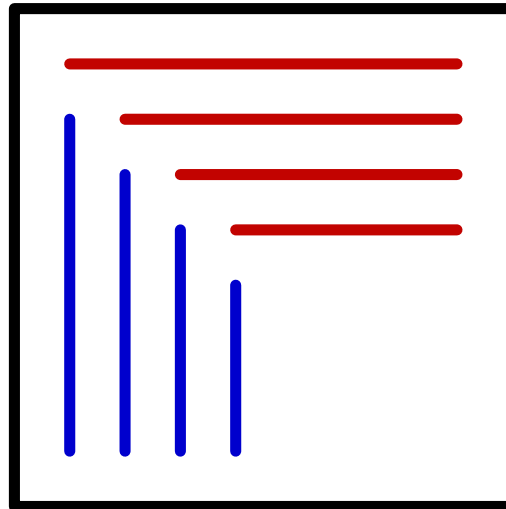
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



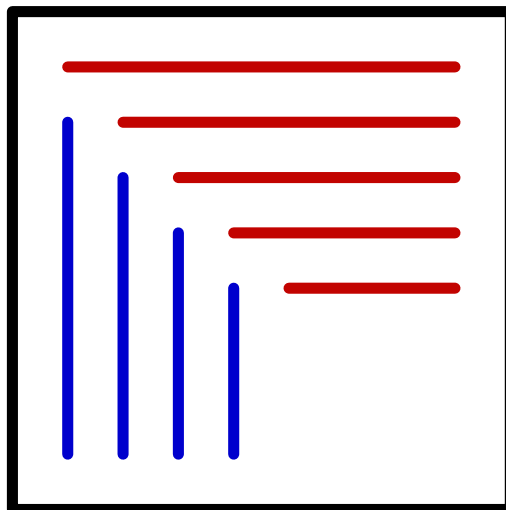
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



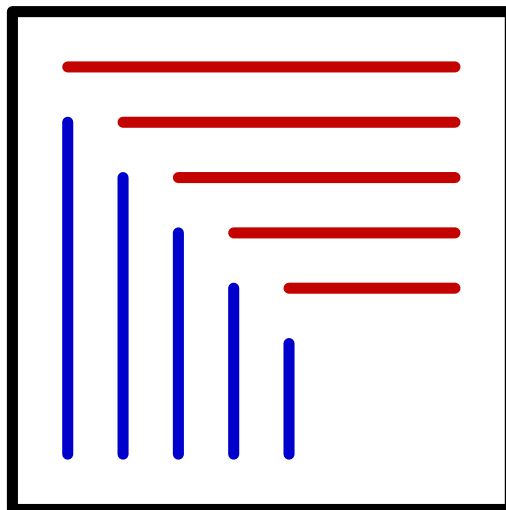
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



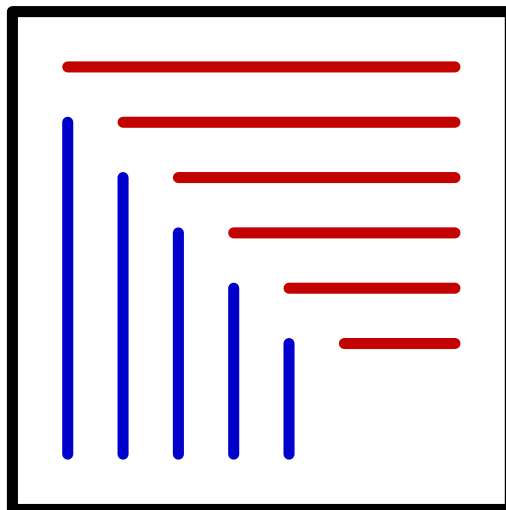
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



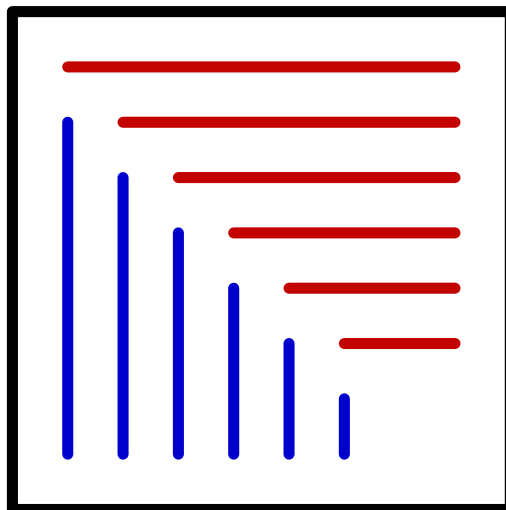
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



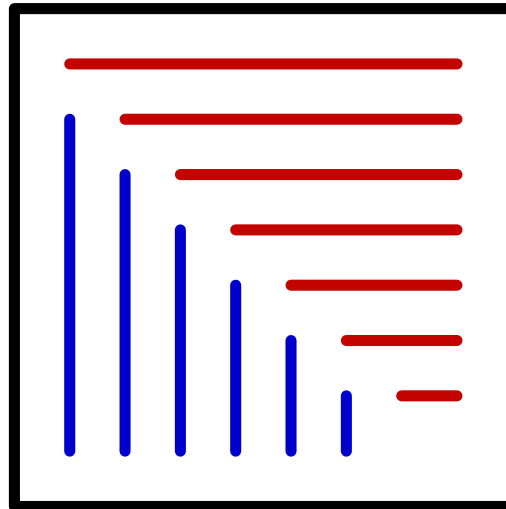
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



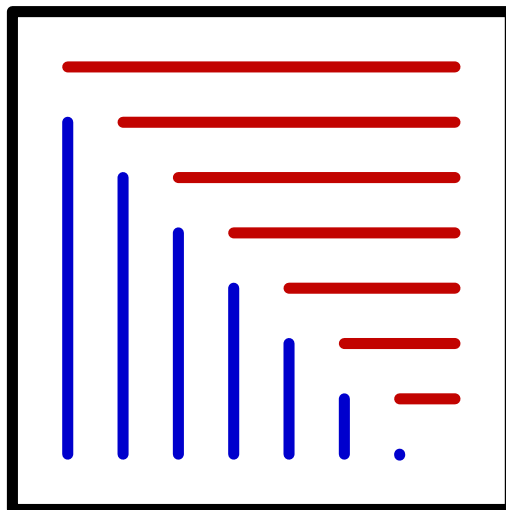
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



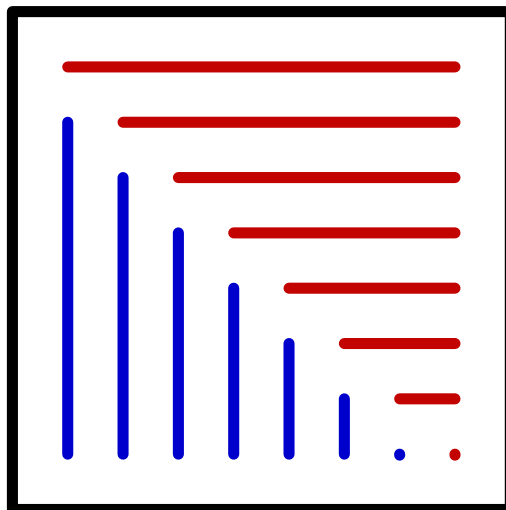
LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



LR faktorizacija (nastavak)

Poredak računanja elemenata — plavo L i crveno R :



LR faktorizacija (nastavak)

Uobičajeno se LR faktorizacija matrice A izvodi tako da se njezina “radna kopija”

• postupno uništava i prepisuje elementima matrica L i R na sljedeći način:

• elementi matrice R spremaju se u gornjem trokutu i na dijagonali,

• elementi matrice L spremaju se u donjem trokutu, s tim da se dijagonala matrice L ne sprema.

Redosljed spremanja — kao na prošlim slikama.

Egzistencija i jedinstvenost LR faktorizacije

Ostaje vidjeti uz koje je uvjete $r_{ii} \neq 0$, za $i = 1, \dots, n$.

Teorem. Postoji **jedinstvena** LR faktorizacija matrice A **ako i samo ako** su vodeće glavne podmatrice $A_k := A(1 : k, 1 : k)$ **regularne**, za $k = 1, \dots, n - 1$.

Ako je A_k **singularna** za neki k , faktorizacija **može postojati**, ali **nije jedinstvena**.

Dokaz. Ide indukcijom po dimenziji matrice.

Prvi smjer. Pretpostavimo da su sve podmatrice A_k **regularne**.

Baza indukcije: Za $k = 1$, postoji jedinstvena LR faktorizacija

$$A_1 = [1] [a_{11}].$$

Egzistencija i jedinstvenost LR faktorizacije

Korak indukcije: Pretpostavimo da A_{k-1} ima jedinstvenu faktorizaciju $A_{k-1} = L_{k-1} R_{k-1}$.

Tražimo faktorizaciju matrice A_k , gdje je

$$A_k = \begin{bmatrix} A_{k-1} & b \\ c^T & a_{kk} \end{bmatrix} = \begin{bmatrix} L_{k-1} & 0 \\ \ell^T & 1 \end{bmatrix} \begin{bmatrix} R_{k-1} & r \\ 0 & r_{kk} \end{bmatrix} := L_k R_k.$$

Da bi jednačbe bile zadovoljene, množenjem dobivamo

$$L_{k-1}r = b, \quad R_{k-1}^T \ell = c, \quad a_{kk} = \ell^T r + r_{kk}.$$

Matrice L_{k-1} i R_{k-1} su regularne, pa postoje **jedinstvena** rješenja prva dva sustava, r , ℓ . Iz zadnje jednačbe dobivamo da je onda i r_{kk} jedinstven.

Egzistencija i jedinstvenost LR faktorizacije

Obrat. Pretpostavimo da je A regularna i postoji njezina LR faktorizacija. Tada je $A_k = L_k R_k$, za $k = 1, \dots, n$. Zbog regularnosti matrice A izlazi

$$\det A = \det R = r_{11} r_{22} \cdots r_{nn} \neq 0,$$

pa je $\det A_k = r_{11} r_{22} \cdots r_{kk} \neq 0$, tj. **sve** podmatrice A_k moraju biti regularne.

Oдавде se lako dobije i **jedinstvenost**.

Tvrdnja vrijedi i kad je A singularna i ima **jedinstvenu LR** faktorizaciju — onda podmatrice A_k moraju biti **regularne**, za $k = 1, \dots, n - 1$ (zadnja podmatrica $A_n = A$ nije regularna).

Egzistencija i jedinstvenost LR faktorizacije

Primjer singularne matrice A za koju postoji LR faktorizacija, ali nije jedinstvena:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Element l_{21} može biti bilo što.

S druge strane, matrica

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

nema LR faktorizaciju, iako je regularna (fali pivotiranje). ■

Gaussove eliminacije i LR faktorizacija

Veza Gaussovih eliminacija i LR faktorizacije

Može se pokazati da je

- matrica R dobivena LR faktorizacijom jednaka
- matrici R dobivenoj Gausovim eliminacijama.

Neka je

- $A^{(k)}$ matrica na početku k -tog koraka Gausovih eliminacija,
- a $A^{(k+1)}$ matrica dobivena na kraju tog koraka.

Onda se $A^{(k+1)}$ može matično napisati kao produkt

$$A^{(k+1)} = M_k A^{(k)},$$

pri čemu je ...

Veza Gaussovih eliminacija i LR faktorizacije

$$M_k = \left[\begin{array}{c|cccc} I_{k-1} & & & & \\ \hline & 1 & & & \\ & -m_{k+1,k} & 1 & & \\ & -m_{k+2,k} & & \ddots & \\ & \vdots & & & \ddots \\ & -m_{n,k} & & & 1 \end{array} \right]$$

a m_{ik} su odgovarajući **multiplikatori** u k -tom koraku.

Na **kraju** eliminacija, nakon $n - 1$ koraka, dobijemo **gornju trokutastu** matricu \tilde{R} ,

$$\tilde{R} := A^{(n)} = M_{n-1} M_{n-2} \cdots M_1 A.$$

Veza Gaussovih eliminacija i LR faktorizacije

Sve matrice M_k su **regularne**, jer su M_k donje trokutaste s 1 na dijagonali, pa postoje njihovi **inverzi**. Onda se A može napisati kao

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} \tilde{R} := \tilde{L}\tilde{R},$$

gdje je

$$\tilde{L} = \begin{bmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ \vdots & m_{32} & \ddots & & \\ \vdots & \vdots & & \ddots & \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1 \end{bmatrix}.$$

Iz **jedinstvenosti** LR faktorizacije slijedi da je $\tilde{R} = R$.

Parcijalno pivotiranje u LR faktorizaciji

Veza LR faktorizacije i Gaussovih eliminacija upućuje nas da pivotiranje vršimo na isti način kao kod Gaussovih eliminacija.

Ako koristimo parcijalno pivotiranje, onda se LR faktorizacija tako dobivene matrice — permutiranih redaka, zapisuje kao

$$PA = LR,$$

pri čemu je P matrica permutacije.

Matrica permutacije P u svakom retku i stupcu

ima točno jednu jedinicu, a ostalo su nule.

P je uvijek regularna matrica — pokažite to!

Parcijalno pivotiranje u LR faktORIZACIJI

Ako znamo “permutiranu” faktORIZACIJU $PA = LR$, kako ćemo riješiti linearni sustav $Ax = b$?

Najjednostavnije je lijevu i desnu stranu (slijeva) pomnožiti s P , pa dobivamo

$$PAx = LRx = Pb.$$

Oprez: kad permutiramo, istovremeno zamjenjujemo retke u obje “radne matrice” — $(L - I)$ i R ,

tj. permutiramo dosadašnje multiplikatore i jednadžbe.

Kako realiziramo permutacije u algoritmu?

Parcijalno pivotiranje u LR faktORIZACIJI

Realizacija permutacija:

- Fizički zamjenjujemo **retke** u **radnoj matrici** A u kojoj formiramo L i R ,
 - $L - I$ u **strogo donjem** trokutu od A ,
 - R u **gornjem** trokutu od A .
- Moramo **pamtiti** permutaciju P , zbog naknadne permutacije desne strane — vektora b .
- Matrica P se pamti kao **vektor** p , koji na mjestu i ima
 - **indeks stupca** j gdje se nalazi **jedinica** u i -tom retku od P ,tj. $p[i] = j \iff P_{ij} = 1$.

Parcijalno pivotiranje u LR faktorizaciji

Primjer. Ako u LR faktorizaciji sustava s 3 jednačbe zamijenimo

prvi i treći redak,

pa onda trenutni drugi i treći redak,

onda će se P , odnosno, p mijenjati ovako:

$$P : \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$p : \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}.$$

Potpuno pivotiranje u LR faktorizaciji

Ako koristimo **potpuno pivotiranje**, dobivamo LR faktorizaciju matrice koja ima permutirane **retke** i **stupce** obzirom na A , tj.

$$PAQ = LR,$$

gdje su P i Q matrice permutacije.

Skica rješenja. Q je unitarna, pa iz $PA = LRQ^T$, uz pokratu $Q^T x = z$, imamo

$$PAx = LR(Q^T x) = LRz = Pb.$$

Dakle, jedina **razlika** obzirom na **parcijalno pivotiranje** je

- da na **kraju** treba “**izokretati**” rješenje z da se dobije x , tj. $x = Qz$.

Pivotni rast

Parcijalno vs. potpuno pivotiranje

Možemo li i na temelju čega reći da je potpuno pivotiranje “bolje” od parcijalnog? Tradicionalno to se čini na temelju pivotnog rasta.

Pivotni rast (ili “faktor rasta”) je omjer

- najvećeg (po apsolutnoj vrijednosti) elementa u svim koracima eliminacije,
- i najvećeg elementa u originalnoj matrici

$$\rho_n = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}.$$

Intuitivno je jasno da nije dobro da elementi rastu po apsolutnoj vrijednosti, jer bi to moglo dovesti do gubitka točnosti.

Pivotni rast

Koliki je **pivotni rast** kod **parcijalnog** pivotiranja?

Korištenjem relacija za **ponišćavanje elemenata**

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, \quad |m_{ik}| \leq 1,$$

za **parcijalno pivotiranje** vrijedi

$$|a_{ij}^{(k+1)}| \leq |a_{ij}^{(k)}| + |a_{kj}^{(k)}| \leq 2 \max_{i,j} |a_{ij}^{(k)}|.$$

Ova ocjena, nakon $n - 1$ koraka algoritma, daje **pivotni rast** ρ_n^p

$$\rho_n^p \leq 2^{n-1}.$$

Pivotni rast

Već je J. H. Wilkinson primijetio da se taj pivotni rast **može doći** za sve matrice oblika

$$\begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & \ddots & & 1 \\ -1 & -1 & \ddots & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

Eksponencijalno rastu elementi **posljednjeg** stupca.

Ovo je samo “**umjetno**” konstruirani primjer, a u praksi je takvih matrica **izrazito malo**, pa se **parcijalno** pivotiranje ponaša **mного bolje** od očekivanog.

Pivotni rast

Za potpuno pivotiranje pivotni rast ρ_n^c može se ograditi odozgo s

$$\rho_n^c \leq n^{1/2} \left(2 \cdot 3^{1/2} \dots n^{1/(n-1)} \right)^{1/2} \approx c n^{1/2} n^{(\log n)/4},$$

ali ta ograda nije dostižna. Ovo je dokazao J. H. Wilkinson, šezdesetih godina prošlog stoljeća.

Dugo se mislilo da vrijedi

$$\rho_n^c \leq n.$$

Međutim, nađeni su primjeri matrica kad to ne vrijedi.

Kontraprimjer (konstruiran 1991. godine), matrice reda 13 ima pivotni rast $\rho_n^c = 13.0205$.

Teorija perturbacije linearnih sustava

Teorija perturbacije linearnih sustava

Teorija perturbacije linearnih sustava bavi se **ocjenom** (po elementima i/ili po normi) koliko se **najviše** promijeni rješenje sustava x , ako se malo promijene elementi A i/ili b .

Problem. Neka je

$$Ax = b,$$

$A \in \mathbb{F}^{n \times n}$ regularna, a b zadani vektor.

Zanima nas koliko će se **najviše** promijeniti rješenje x ovog problema, ako **perturbiramo** A , odnosno, b .

- Pojednostavnimo problem i pretpostavimo da je A **fiksna** matrica, a dozvoljene su perturbacije **samo** vektora b .

Koliko je dobro uvjetovan linearni sustav?

Za prethodni problem

- ulazni podaci su elementi od A i b — njih $n^2 + n$,
- rezultat je vektor $x \in \mathbb{F}^n$.

Uzmimo kao da b varira, a A je “fiksna” matrica.

- Pripadna funkcija problema je $f_A : \mathbb{F}^n \rightarrow \mathbb{F}^n$, uz

$$x = f_A(b) := A^{-1}b.$$

Iz prethodnog predavanja znamo da je relativna uvjetovanost problema (samo, umjesto x , pišemo b)

$$\kappa_{\text{rel}}(b) = (\text{cond } f_A)(b) := \left| \frac{b f'_A(b)}{f_A(b)} \right|.$$

Koliko je dobro uvjetovan linearni sustav?

Za višedimenzionalne probleme, da bismo dobili **jedan broj** u prethodnoj formuli, uzmemo **normu**, a derivacija je **gradijent**,

$$\frac{\partial x}{\partial b} = \nabla f_A(b) = A^{-1},$$

pa je

$$\kappa_{\text{rel}}(b) = \frac{\|b\|_2 \|A^{-1}\|_2}{\|A^{-1}b\|_2} = \frac{\|Ax\|_2 \|A^{-1}\|_2}{\|x\|_2}.$$

Gledamo **najgoru** moguću uvjetovanost, po **svim** vektorima b ,

$$\max_{\substack{b \in \mathbb{F}^n \\ b \neq 0}} \kappa_{\text{rel}}(b) = \max_{\substack{x \in \mathbb{F}^n \\ x \neq 0}} \frac{\|Ax\|_2}{\|x\|_2} \|A^{-1}\|_2 = \|A\|_2 \|A^{-1}\|_2.$$

Primjer

Za mjeru uvjetovanosti sustava, možemo uzeti:

$$\text{cond } A := \|A\|_2 \|A^{-1}\|_2.$$

Ako je uvjetovanost mala, mora li onda rješenje računalom biti dobro?

Primjer. Sjetimo se sustava $Ax = b$, gdje je

$$A = \begin{bmatrix} 0.0001 & 1 \\ 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Za vježbu izračunajte da je

$$\text{cond } A \approx 2.6183852736548268689.$$

Primjer

Je li to dobro uvjetovan sustav? Jest!

Digresija. Nije teško pokazati da za regularne matrice vrijedi

$$1 = \|A \cdot A^{-1}\| \leq \|A\|_2 \|A^{-1}\|_2 = \text{cond}(A),$$

a jednakost se dostiže za unitarne matrice. Uvjetovanost je loša ako je $\text{cond}(A) \gg 1$. ■

U prethodnom sustavu je nešto pošlo po zlu! Što?

Na bitnom mjestu u računu došlo je do “underflow-a”

🕒 mali broj je pretvoren u nulu,

i više ne možemo govoriti o malim relativnim perturbacijama.