

Numerička matematika

3. predavanje

Saša Singer

`singer@math.hr`

`web.math.pmf.unizg.hr/~singer`

PMF – Matematički odsjek, Zagreb

Sadržaj predavanja

- Rješavanje linearnih sustava:
 - Uvod i ponavljanje teorije o linearnim sustavima.
 - Gaussove eliminacije.
 - Zamjene jednadžbi (redaka) — parcijalno pivotiranje.
 - Zamjene redaka i stupaca — potpuno pivotiranje.
 - Gaussove eliminacije — algoritam i složenost.
 - LR (LU) faktorizacija.
 - Veza Gaussovih eliminacija i LR faktorizacije.
 - Pivotiranje u LR (LU) faktorizaciji.
 - Perturbacije linearnih sustava i uvjetovanost matrice.

Informacije — predavanja

Trenutno nema konkretnih informacija.

- Pratite web stranice fakulteta!
- Za “nastavu na daljinu” pratite web stranice kolegija.

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** rangu 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) ne postoji, ili nije jedinstveno (koje od beskonačno 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, upravo, 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 (kao u definiciji \det)

- “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 rješavanje 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 svođenja 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

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, tj. sve elemente strogo ispod dijagonale.

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 (tzv. “ključna”) jednačba 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** (strogo) **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 **ponišćimo** 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 strogo **ispod** dijagonale u k -tom **stupcu** matrice $A^{(k)}$, koristeći “**ključni**” k -ti redak.

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 $i, j = k + 1, \dots, n$. Iz $a_{ik}^{(k+1)} = 0$, **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 $i = 1, \dots, n - 1$ (ne treba n), 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 strogo 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 tzv. **povratnom supstitucijom** (supstitucijom unatrag)

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

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

Pitanje: Ako je A kvadratna i **regularna** matrica,

• 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

$$[A \mid b] = \begin{bmatrix} 0 & 1 & \vdots & 1 \\ 1 & 0 & \vdots & 1 \end{bmatrix},$$

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

- a ipak ga **ne možemo** riješiti Gaussovima eliminacijama,
- ako **ne mijenjamo poredak** jednačbi.

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

Gaussove eliminacije s pivotiranjem

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

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

Odgovor: Ako dozvolimo **pivotiranje** u svakom koraku,

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

Objašnjenje: Ako u **prvom** stupcu **ne postoji ne-nula** element, matrica je **singularna**. Isto vrijedi i za **svaki** sljedeći **korak** = stupac ispod dijag. (Dokaz: 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**, **ispod** dijagonale)?

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, ispod dijag.).
- U **praksi** — kad računamo **približno**, to **može** dovesti do potpuno **pogrešnog** rezultata.

Jedna jedina operacija može **upropastiti** rezultat!

- Postoji i **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

$$x_1 = \frac{10000}{9999} = 1.\dot{0}00\dot{1}, \quad x_2 = \frac{9998}{9999} = 0.\dot{9}99\dot{8}.$$

Riješimo taj sustav “računalom” u bazi 10 — s 4 značajne znamenke mantise i 2 znamenke eksponenta (ovo nije bitno).

Uočiti: Broj $0.0001 = 10^{-4}$ je “mali”, ali nije nula. Po teoriji,  možemo ga uzeti kao prvi (ili bilo koji) ne-nula element.

Gaussove eliminacije — primjer (nastavak)

U takvom “računalu”, sustav se pamti **bez greške**, 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 ni približno točan rezultat**.

Gaussove eliminacije — primjer (nastavak)

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

- prvu jednadžbu množimo **velikim** brojem -10^4 (po apsolutnoj vrijednosti) i **dodajemo 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**” (dovoljno **malo**)!

Isto bi nam se dogodilo za **bilo koju drugu** jednadžbu oblika

$$x_1 + \alpha x_2 = \beta,$$

gdje su $|\alpha|, |\beta| < 5$. (Za ovo “računalo” je $u = 5 \cdot 10^{-4}$.)

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, na obje strane. 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

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

za $\varepsilon = 10^{-4}, \dots, 10^{-25}$, Gaussovima eliminacijama **bez** zamjena i **sa zamjenom** poretka jednačbi, u aritmetici **računala**.

Računanjem u **najvećoj** dostupnoj preciznosti = tip **extended** ($u \approx 5.42 \cdot 10^{-20}$), dobivamo tablicu na sljedećoj stranici.

Za x_1 — **crvene** znamenke su **pogrešne**, **ljubičaste** su korektne kad ih **zaokružimo**, a **zelene** su **točne**. Za x_2 — obje metode daju **iste** (i **točne**) vrijednosti, pa je naveden samo **jednom**.

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	0.99999989999999000
⋮	⋮	⋮	
-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
⋮	isto	isto	isto

Gaussove eliminacije s parcijalnim pivotiranjem

Standardni naziv: **pivotni** element = element koji se **prije** k -tog koraka eliminacije dovodi na **dijagonalno** mjesto $a_{kk}^{(k)}$.

U praksi se obično bira korištenjem **parcijalnog pivotiranja**.

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

Preciznije, ako je u k -tom koraku

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

onda ćemo **prvo zamijeniti** r -ti i k -ti **redak**, a zatim 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 računala može doći do **kraćenja** ili **gubitka** najmanje značajnih znamenki $a_{ij}^{(k)}$, tako da izračunati $a_{ij}^{(k+1)}$ može imati **veliku** relativnu grešku, ili bitno **narasti** → **gubitak** informacija iz originalne jednadžbe.

Gaussove eliminacije s parcijalnim pivotiranjem

Sasvim općenito, ideja pivotiranja je **minimizirati** “korekcije” **elemenata** pri prijelazu s $A^{(k)}$ na $A^{(k+1)}$. Dakle,

- multiplikatori m_{ik} trebaju biti **što manji**, po apsolutnoj vrijednosti.

Ekvivalentno, **pivotni** element treba biti **što veći**, jer ulazi u **nazivnik** multiplikatora.

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**” (v. kasnije).

Gaussove eliminacije s potpunim pivotiranjem

Osim parcijalnog, može se provoditi i **potpuno pivotiranje**.

- U k -tom koraku, bira se **najveći** 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 **prvo** zamijeniti r -ti i k -ti redak, s -ti i k -ti stupac, a zatim početi korak eliminacije elemenata k -tog stupca.

Oprez: zamjenom s -tog i k -tog stupca zamijenili smo ulogu nepoznanica (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 maks. |element| u ostatku stupca */  
    max_elt = |A[k, k]|;  
    ind_max = k;  
    za i = k + 1 do n radi {  
        ako je |A[i, k]| > 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)

```
    /* Korak Gaussovih eliminacija */
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 */
ako je A[n, n] <> 0.0 onda {
    /* Rješenje x */
    x[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] * x[j];
        }
        x[i] = sum / A[i, i];
    }
}
inače
    /* Matrica je singularna, STOP */
```

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)

Ukupan broj aritmetičkih operacija u k -tom koraku je

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

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 iz prve zamjenom indeksa $n - k \mapsto k$ (granice za sumu ostaju iste). Onda iskoristimo formule za sumu i sumu kvadrata prvih $n - 1$ prirodnih brojeva.

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, točno n^2 operacija.

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

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

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

Gaussove eliminacije — komentari

Par završnih komentara, nakon 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 sustave ne treba pivotiranje,
- za vrpčaste, posebno, trodijagonalne matrice, algoritam se drastično skraćuje (v. kubična spline interpolacija).

LR faktorizacija

LR faktorizacija

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

$$A = LR,$$

pri čemu je

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

Nazivi: “**LR**” = (left, right), “**LU**” = (lower, upper).

Matrica L je **regularna**, jer je $\det L = 1$. Onda je regularnost matrice A **ekvivalentna** regularnosti matrice R , jer vrijedi

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

LR faktorizacija — rješenje sustava

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

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

Lako pamćenje: matrice u sustavima idu slijeva \mapsto udesno.

Prednost LR faktorizacije:

- za zadani b , 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 — rješenje sustava (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 (unatrag)

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

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

LR faktorizacija — nalaženje

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

- Iskoristimo **poznatu strukturu** matrica 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** izračunati iz već **poznatih** elemenata.

- 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} .
- I tako **redom**, $i = 2, j = 2, \dots, i = n$ (**bez** $j = n$).

LR faktorizacija — nalaženje (nastavak)

Tako dobivamo **rekurzivne** relacije 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} (nema l -ova).

LR faktorizacija — nalaženje (nastavak)

Napomena. Ako je $r_{ii} \neq 0$, za $i = 1, \dots, n - 1$ (bez n), onda iz prethodnih relacija možemo

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

Drugim riječima,

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

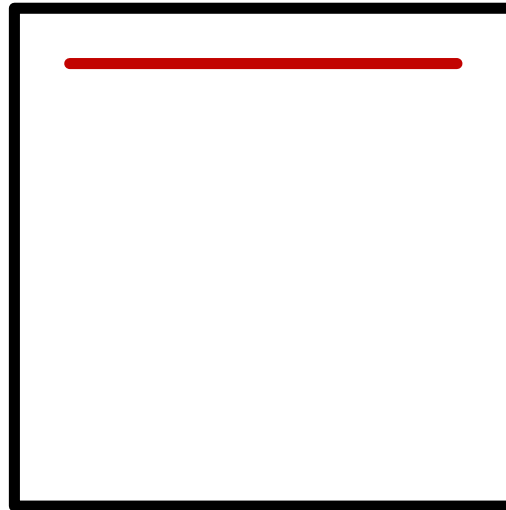
Primijetite da $r_{nn} \neq 0$ treba samo za **povratnu** supstituciju.

Pitanje: Kojim se **redom** računaju elementi od L i R ?

- Može **točno** prema prethodnim relacijama (v. slikice), ali
- **neke** elemente smijemo računati i **kasnije** — za efikasno korištenje tzv. **cache** memorije (granice i poredak petlji).

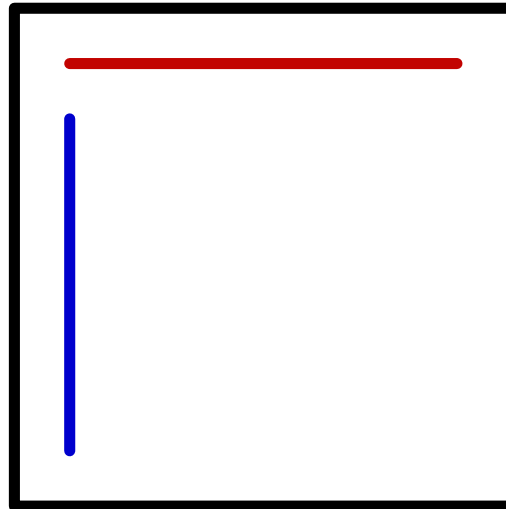
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



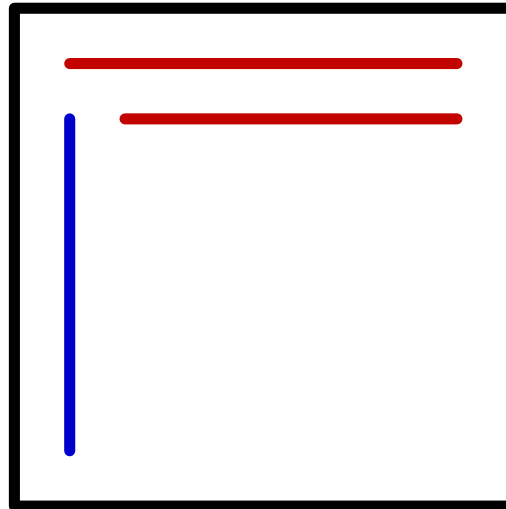
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



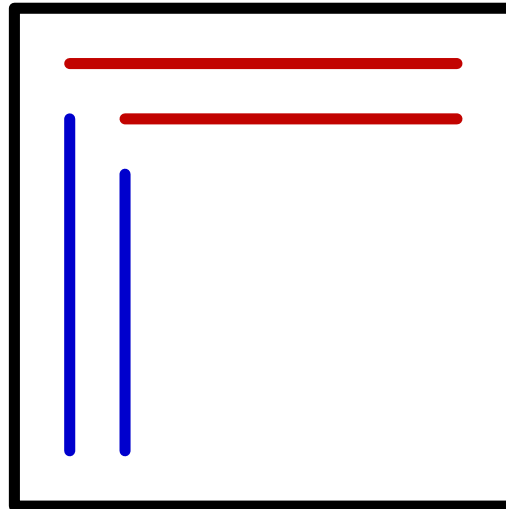
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



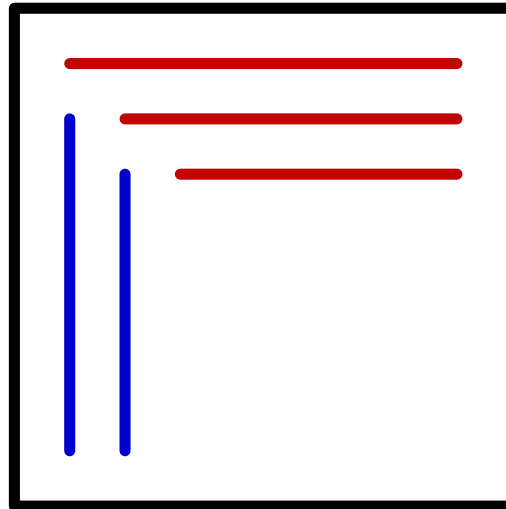
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



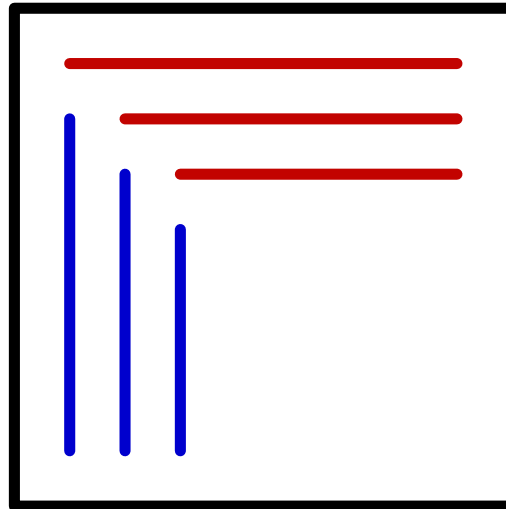
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



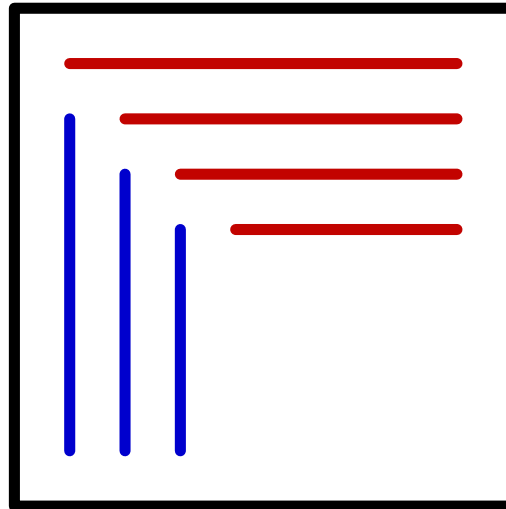
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



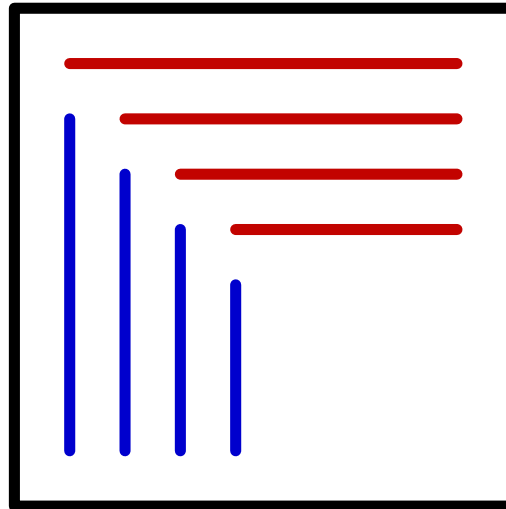
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



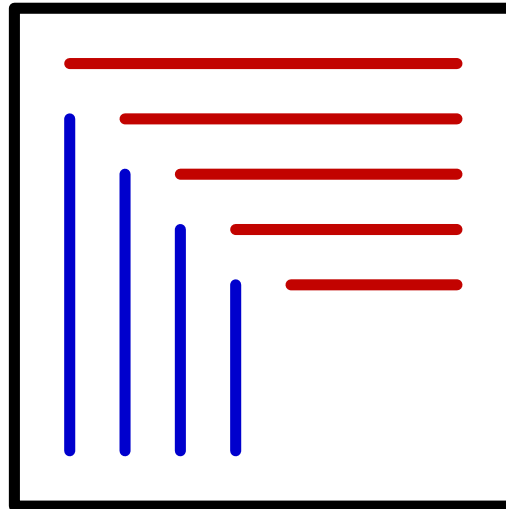
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



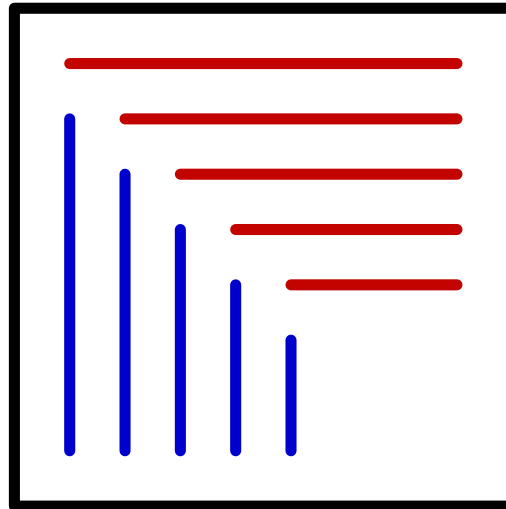
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



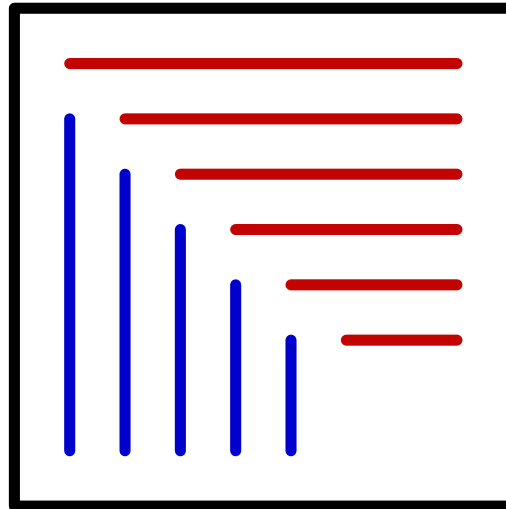
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



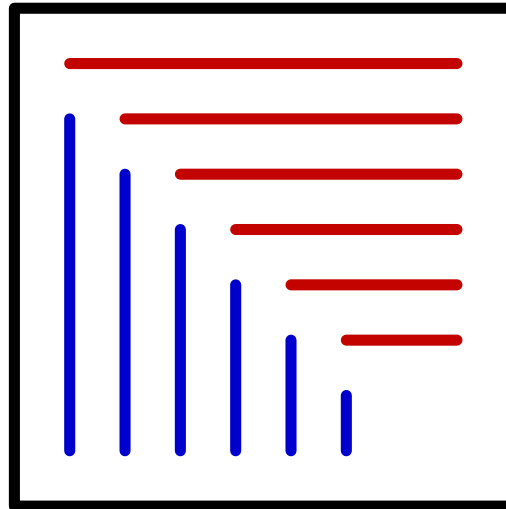
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



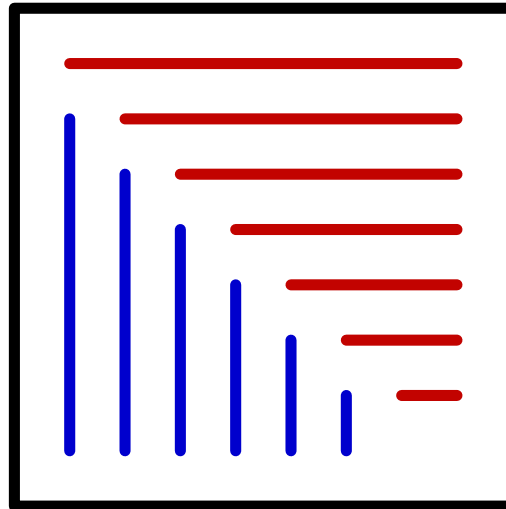
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



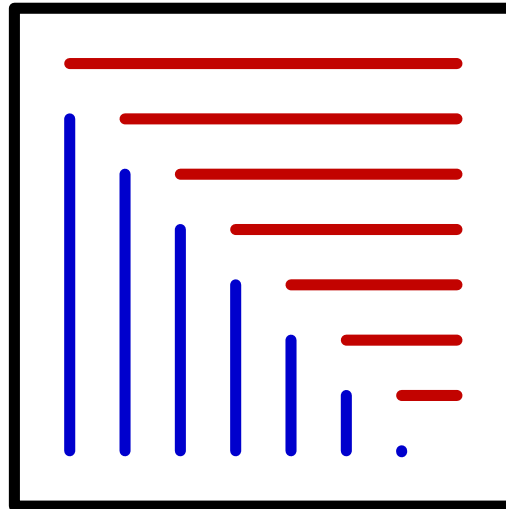
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



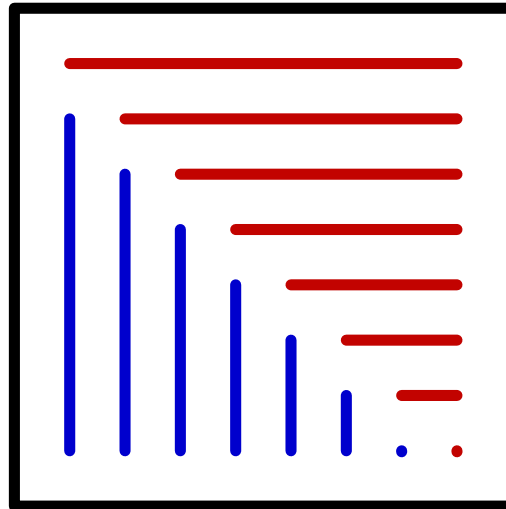
LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



LR faktorizacija — poredak nalaženja

Poredak računanja elemenata — plavo za L i crveno za R :



LR faktorizacija — spremanje elemenata

Uobičajeno se LR faktorizacija matrice A izvodi tako da se njezina “radna kopija” = neko polje u memoriji računala,

- koje, na početku, sadrži matricu A ,
- 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 (znamo da su 1).

Redosljed spremanja — kao na prošlim slikama, ili drugačije (ovisno o granicama i poretku petlji).

Digresija — “Matlab” oznake za podmatrice

Ostaje još vidjeti uz koje **uvjete** na matricu A vrijedi $r_{ii} \neq 0$, za $i = 1, \dots, n - 1$.

Za iskaz teorema, korisno je uvesti tzv. **Matlab** oznake za one **podmatrice** matrice A , koje se dobivaju na **presjeku** redaka i stupaca, s indeksima u zadanim **rasponima** (oblika “od : do”).

- $A(i : j, k : \ell)$ = podmatrica od A na **presjeku** redaka, od i -tog do j -tog, i stupaca, od k -tog do ℓ -tog.
Tip ove podmatrice je $(j - i + 1) \times (\ell - k + 1)$.
- Ako za raspon napišemo **samo** “ : ”, podrazumijevaju se **svi** dozvoljeni indeksi, od prvog do zadnjeg iz A .

Definiramo još i **apsolutnu** vrijednost matrice A , u oznaci $|A|$. To je **matrica** istog tipa kao i A , čiji elementi su **apsolutne** vrijednosti elemenata od A , tj. $(|A|)_{ij} = |a_{ij}|$, za sve i, j .

Digresija — Blok–matrice i operacije

Za skraćeni zapis, matricu A možemo podijeliti na **blokove** ili **podmatrice** — vodoravnim i okomitim “crtama”, kao na slici

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1q} \\ A_{21} & A_{22} & \cdots & A_{2q} \\ \vdots & \vdots & \cdots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pq} \end{bmatrix},$$

gdje su A_{ij} , općenito, **pravokutne** matrice.

Kad imamo dvije matrice A i B , podijeljene u blokove, onda se matrice operacije $+$ i \cdot “**po blokovima**” rade **isto** kao i **po elementima**, uz **uvjet** da su operacije **korektno definirane** za pripadne blokove, tj. da ti blokovi imaju odgovarajući **tip**.

Digresija — Blok–matrice i operacije (nastavak)

Primjer. Neka su A i B kvadratne matrice reda n , s **istom** podjelom na blokove, tj. blokovi A_{ij} i B_{ij} su **istog tipa**,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

Ako u produktu $C = A \cdot B$ napravimo **istu** podjelu na blokove, onda za blokove C_{ij} vrijedi **ista** formula kao i “po elementima”

$$C_{ij} = \sum_{k=1}^2 A_{ik} \cdot B_{kj}, \quad i, j = 1, 2.$$

Napomena. Ovo koristimo u dokazu teorema. Dodatno, kad je blok A_{11} reda $n - 1$, onda blok A_{12} pišemo kao **vektor** (v), blok A_{21} kao **transponirani vektor** (w^T), a “blok” A_{22} je **skalar** a_{nn} .

Egzistencija i jedinstvenost LR faktorizacije

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 onda sigurno **nije jedinstvena** (v. primjere na kraju dokaza).

Dokaz. Za **prvi smjer**, pretpostavimo da su sve podmatrice A_k **regularne**, za $k = 1, \dots, n - 1$. Konstrukcija LR faktorizacije za $A = A_n$ napreduje induktivno po dimenziji k .

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

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

Egzistencija i jedinstvenost LR faktorizacije

Korak indukcije: Pretpostavimo da je $k > 1$ i da podmatrica A_{k-1} ima jedinstvenu LR faktorizaciju $A_{k-1} = L_{k-1} R_{k-1}$.

Tražimo LR faktorizaciju podmatrice A_k , gdje je

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

Množenjem dobivamo da moraju vrijediti sljedeće jednačbe

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

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

Egzistencija i jedinstvenost LR faktorizacije

Obrat. Pretpostavimo da matrica A ima jedinstvenu LR faktorizaciju $A = LR$ i označimo

$$L_k := L(1 : k, 1 : k), \quad R_k := R(1 : k, 1 : k), \quad k = 1, \dots, n.$$

Onda je, raspisom kao na prethodnoj stranici, za $k = n$

$$A = \begin{bmatrix} A_{n-1} & b \\ c^T & a_{nn} \end{bmatrix} = \begin{bmatrix} L_{n-1} & 0 \\ \ell^T & 1 \end{bmatrix} \begin{bmatrix} R_{n-1} & r \\ 0 & r_{nn} \end{bmatrix} := LR.$$

Množenjem dobivamo da onda vrijede sljedeće četiri jednačbe

$$A_{n-1} = L_{n-1} R_{n-1}, \quad L_{n-1} r = b, \quad a_{nn} = \ell^T r + r_{nn}, \\ R_{n-1}^T \ell = c,$$

Sad iskoristimo jedinstvenost matrica L i R u faktorizaciji.

Egzistencija i jedinstvenost LR faktorizacije

To znači da vektor ℓ mora biti **jedinstveno** rješenje sustava

$$R_{n-1}^T \ell = c,$$

pa matrica R_{n-1} mora biti **regularna**, tj. vrijedi

$$\det R_{n-1} = r_{11} r_{22} \cdots r_{n-1,n-1} \neq 0.$$

Iz strukture matrica L i R (rastavom unatrag) vidimo da je $A_k = L_k R_k$, za **sve** $k = 1, \dots, n - 1$, pa je $\det A_k = \det R_k$. Iz regularnosti R_{n-1} onda slijedi

$$\det A_k = \det R_k = r_{11} r_{22} \cdots r_{kk} \neq 0, \quad k = 1, \dots, n - 1.$$

Dakle, **sve** podmatrice A_k su **regularne**, za $k = 1, \dots, n - 1$.

Samo **zadnja** matrica $A_n = A$ može biti **singularna** ($r_{nn} = 0$).

Egzistencija i jedinstvenost LR faktorizacije

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

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

Ovdje je $A_1 = R_1 = 0$, sustav za ℓ_{21} je $0 \cdot \ell_{21} = 0$ (u skladu s prethodnim dokazom), pa 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). Sustav za ℓ_{21} ovdje glasi $0 \cdot \ell_{21} = 1$ i nema rješenja. ■

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, kao ranije,

- $A^{(k)}$ 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 matrica “transformacije” M_k ima sljedeći oblik ...

Veza Gaussovih eliminacija i LR faktorizacije

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

Matrica I_{k-1} je **jedinična** matrica reda $k - 1$, a m_{ik} su odgovarajući **multiplikatori** u k -tom koraku eliminacija.

Na **kraju** eliminacija, nakon $n - 1$ koraka, dobijemo **gornju trokutastu** matricu

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

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 $\tilde{R} = R$. Usput, slijedi i $\tilde{L} = L$, pa imamo vezu matrice L s multiplikatorima.

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, čak ortogonalna — pokažite to!
Zato P ima inverz i vrijedi $P^{-1} = P^T$.

Parcijalno pivotiranje u LR faktorizaciji

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

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

$$PAx = LRx = Pb.$$

Dakle, u prvom koraku rješavamo sustav $Ly = Pb$.

Oprez: kad permutiramo, istovremeno zamjenjujemo retke u obje “radne matrice” u polju — to su $(L - I)$ 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.

$$p[i] = j \iff P_{ij} = 1.$$

Za velike matrice — može i bez zamjena, dovoljan je vektor p .

Parcijalno pivotiranje u LR faktORIZACIJI

Primjer. Ako u LR faktORIZACIJI sustava s 3 jednaDžbe

• prvo zamijenimo prvi i treći redak,

• pa onda trenutni drugi i treći redak,

onda će se matrica P , odnosno, vektor 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 (P za **retke**, Q za **stupce**).

Rješenje sustava $Ax = b$ dobivamo kao i prije — iz $PAx = Pb$.

Q je ortogonalna, pa je $PA = LRQ^T$. Uz pokratu $Q^T x = z$, imamo

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

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

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

Perturbacije linearnih sustava i uvjetovanost matrice

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 od A i/ili b .

Problem. Neka je

$$Ax = b,$$

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

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

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

U **općem** obliku problema, ulaznih podataka je **puno**.

Koliko je dobro uvjetovan linearni sustav?

Zato, pojednostavnimo problem i pretpostavimo da je

- A “fiksna” matrica (ne varira),
- a dozvoljene su perturbacije samo vektora b (on varira).

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

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

Iskoristimo ranije rezultate — samo, umjesto x, y , pišemo b, x .

U grubljoj analizi gledamo relativne perturbacije “po normi”, a relativna uvjetovanost problema je

$$(\text{cond } f_A)(b) := \frac{\|b\|}{\|f_A(b)\|} \cdot \left\| \frac{\partial f_A}{\partial b} \right\|.$$

Koliko je dobro uvjetovan linearni sustav?

Funkcija problema $f_A(b) = A^{-1}b$ je **linearna**, pa je **Jacobijeva matrica** te funkcije

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

Onda je

$$(\text{cond } f_A)(b) = \frac{\|b\| \cdot \|A^{-1}\|}{\|A^{-1}b\|} = \frac{\|Ax\| \cdot \|A^{-1}\|}{\|x\|}.$$

Nađimo **najgoru** moguću relativnu uvjetovanost sustava, po **svim** vektorima b — u bilo kojoj **operatorskoj** normi $\|\cdot\|$:

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

Uvjetovanost matrice

Dobili smo broj koji ovisi **samo** o matrici A .

Definicija. Broj uvjetovanosti ili **uvjetovanost** matrice A je

$$\text{cond}(A) = \kappa(A) := \|A\| \cdot \|A^{-1}\|.$$

U ovoj definiciji, norma može biti **bilo koja** matrična norma, a najčešće se koriste **operatorske** norme.

Oznaka norme = uvjetovanost dobije **indeks** norme. Na pr.

$$\text{cond}_2(A) = \kappa_2(A) := \|A\|_2 \cdot \|A^{-1}\|_2.$$

Napomena. Kao mjera uvjetovanosti **linearnog sustava**, uvjetovanost matrice je **dostižna** u **operatorskim** normama, tj.

• **postoji** desna strana b za koju je $(\text{cond } f_A)(b) = \text{cond}(A)$.

Osnovna svojstva uvjetovanosti matrice

Za regularne matrice, u bilo kojoj **operatorskoj** normi vrijedi

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

Zato kažemo da je A **loše** uvjetovana ako je $\text{cond}(A) \gg 1$. ■

Posebno, u **unitarno** invarijantnoj **2-normi** vrijedi

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

a jednakost se **dostiže** za **unitarne** matrice A i za αA .

Dodatno, za bilo koje dvije **unitarne** matrice U i V vrijedi

$$\text{cond}_2(UAV) = \text{cond}_2(A),$$

jer su inverzi $U^{-1} = U^*$ i $V^{-1} = V^*$, opet, **unitarne** matrice. ■

Perturbacije linearnih sustava (nastavak)

Još malo o perturbacijama linearnih sustava

Ocjenu koliko se **najviše** promijenilo rješenje sustava $Ax = b$,

• ako perturbiramo samo b ili samo A ,

• ako perturbiramo i A i b ,

možemo dobiti **direktno** — po **normi** i po **elementima**.

U nastavku, gledamo samo **relativne** perturbacije po **normi**.

Razumna **pretpostavka**: perturbacije Δb vektora b , odnosno, ΔA matrice A , su relativno, po normi, **odozgo** ogradržene nekim brojem ε , tj. vrijedi

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

Komentar. Ako je ε točnost računanja, tolika perturbacija je **napravljena** već pri **spremanju** elemenata od b ili A u računalo.

Perturbacija vektora b

Za početak, pretpostavimo da smo perturbirali **samo** vektor b i da za **vektorsku** normu **perturbacije** vektora b vrijedi

$$\|\Delta b\| \leq \varepsilon \|b\|.$$

Umjesto sustava $Ax = b$, onda rješavamo sustav

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

Od ovog sustava **oduzmemo** $Ax = b$, pa ostaje

$$A \Delta x = \Delta b.$$

Množenjem slijeva s A^{-1} dobivamo

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

Uzmemo normu obje strane i desnu stranu **ocijenimo odozgo**.

Perturbacija vektora b (nastavak)

Korištenjem pretpostavke $\|\Delta b\| \leq \varepsilon \|b\|$, dobivamo

$$\begin{aligned}\|\Delta x\| &\leq \|A^{-1}\| \|\Delta b\| \leq \varepsilon \|A^{-1}\| \|b\| = \varepsilon \|A^{-1}\| \|Ax\| \\ &\leq \varepsilon \|A^{-1}\| \|A\| \|x\| = \varepsilon \kappa(A) \|x\|,\end{aligned}$$

pri čemu je $\kappa(A) = \|A\| \|A^{-1}\|$ uvjetovanost matrice A .

To pokazuje da je pogreška u rješenju (relativno, po normi)

• proporcionalna uvjetovanosti matrice A .

Korektno bi bilo dodati “u najgorem slučaju po b ”, jer imamo ocjenu odozgo, ali se ona može dostići.

Ovaj rezultat odgovara ranijem za relativnu uvjetovanost po normi — na temelju kojeg smo definirali uvjetovanost matrice.

Perturbacija matrice A

Pretpostavimo da smo perturbirali **samo** matricu A i da za **operatorsku** normu **perturbacije** vrijedi

$$\|\Delta A\| \leq \varepsilon \|A\|.$$

Umjesto sustava $Ax = b$, onda rješavamo sustav

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

Od ovog sustava **oduzmemo** $Ax = b$, pa ostaje

$$A \Delta x + \Delta A (x + \Delta x) = 0.$$

Množenjem slijeva s A^{-1} i sređivanjem dobivamo

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

Uzmemo normu obje strane i desnu stranu **ocijenimo odozgo**.

Perturbacija matrice A (nastavak)

Korištenjem pretpostavke $\|\Delta A\| \leq \varepsilon \|A\|$, dobivamo

$$\begin{aligned}\|\Delta x\| &\leq \|A^{-1}\| \|\Delta A\| \|x + \Delta x\| \leq \varepsilon \|A^{-1}\| \|A\| \|x + \Delta x\| \\ &\leq \varepsilon \kappa(A) (\|x\| + \|\Delta x\|).\end{aligned}$$

Na lijevu stranu **prebacimo** sve članove koji sadrže Δx . Izlazi

$$(1 - \varepsilon \kappa(A)) \|\Delta x\| \leq \varepsilon \kappa(A) \|x\|.$$

Ako je $\varepsilon \kappa(A) < 1$, a to znači da je i $\|\Delta A\| \|A^{-1}\| < 1$, **onda** je

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

pa je **pogreška** u rješenju (relativno, po normi)

- približno **proporcionalna uvjetovanosti** matrice A .

Perturbacija matrice A i vektora b

Kad perturbiramo A i b — zbrojimo ranije ocjene. Poopćenje:

Teorem (v. Higham, ASNA2). Neka je $Ax = b$ i neka je

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

gdje je

$$\|\Delta A\| \leq \varepsilon \|E\|, \quad \|\Delta b\| \leq \varepsilon \|f\|,$$

pri čemu je E neka matrica, a f neki vektor. Također, neka je

$$\varepsilon \|A^{-1}\| \|E\| < 1.$$

Tada, za $x \neq 0$, vrijedi ocjena

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\varepsilon}{1 - \varepsilon \|A^{-1}\| \|E\|} \left(\frac{\|A^{-1}\| \|f\|}{\|x\|} + \|A^{-1}\| \|E\| \right).$$

Perturbacija matrice A i vektora b (nastavak)

Komentar. Uobičajeno se za E uzima A , jer je to **pogreška** koju napravimo spremanjem matrice A u računalo. Jednako tako, za f se obično uzima b . U tom slučaju je

$$\begin{aligned}\frac{\|\Delta x\|}{\|x\|} &\leq \frac{\varepsilon}{1 - \varepsilon \|A^{-1}\| \|A\|} \left(\frac{\|A^{-1}\| \|b\|}{\|x\|} + \|A^{-1}\| \|A\| \right) \\ &= \frac{\varepsilon}{1 - \varepsilon \kappa(A)} \left(\frac{\|A^{-1}\| \|Ax\|}{\|x\|} + \kappa(A) \right) \\ &\leq \frac{\varepsilon}{1 - \varepsilon \kappa(A)} \left(\frac{\|A^{-1}\| \|A\| \|x\|}{\|x\|} + \kappa(A) \right) \\ &= \frac{2\varepsilon \kappa(A)}{1 - \varepsilon \kappa(A)}.\end{aligned}$$

Perturbacija matrice A i vektora b (nastavak)

Dokaz (skica). Provodi se na sličan način kao za pojedinačne perturbacije vektora b , odnosno, matrice A .

Od $(A + \Delta A)(x + \Delta x) = b + \Delta b$ oduzmemo $Ax = b$, pa ostaje

$$A \Delta x = \Delta b - \Delta A x - \Delta A \Delta x.$$

Množenjem s A^{-1} slijeva, a zatim korištenjem svojstava **operatorskih** normi, s malo truda, izlazi traženo. ■

Malo kompliciranije, mogu se dobiti i ocjene za perturbacije po **elementima**. Na primjer, uz pretpostavke da je

$$|\Delta A| \leq \varepsilon |E|, \quad |\Delta b| \leq \varepsilon |f|,$$

gdje je E neka matrica, a f neki vektor (v. Higham, ASNA2). **Nejednakost** za matrice \Leftrightarrow vrijedi po elementima, za svaki.

Komentar rezultata teorije perturbacija

Uočimo da sve ocjene vrijede

- samo za “dovoljno male” perturbacije matrice A .

U općem teoremu, za relativne perturbacije po normi, mora biti

$$\varepsilon \|A^{-1}\| \|E\| < 1, \quad \text{odnosno,} \quad \varepsilon \kappa(A) < 1.$$

Druga relacija se dobiva za $E = A$.

U protivnom, ocjena **ne vrijedi** (nazivnik nula ili krivi znak),

- tj. **relativna** greška (po normi) može biti **po volji velika**.

Pitanje: Što kaže **obratna** analiza **grešaka zaokruživanja**, tj.

- koje su **ocjene** na perturbacije, kad računamo **približno**?

Odgovor na to — malo kasnije (sljedeće predavanje).

Primjer — loša uvjetovanost

Primjer. Na prvom predavanju imali smo primjer sustava

$$2x_1 + 6x_2 = 8$$

$$2x_1 + 6.0001x_2 = 8.0001,$$

s rješenjem $x_1 = 1$, $x_2 = 1$, i malo perturbiranog sustava

$$2x_1 + 6x_2 = 8$$

$$2x_1 + 5.99999x_2 = 8.00002.$$

s rješenjem $x_1 = 10$, $x_2 = -2$.

Ovdje je $\|\Delta A\|_2 < 10^{-4} \|A\|_2$ i $\|\Delta b\|_2 < 10^{-4} \|b\|_2$. Krivac za veliku perturbaciju u rješenju je loša uvjetovanost matrice A

$$\kappa_2(A) \approx 4.00006 \cdot 10^5.$$

Zato je $\|\Delta A\|_2 \|A^{-1}\|_2 > 1$, pa ranija ocjena ne vrijedi.

Primjer — uvjetovanost i izračunato rješenje

Pitanje: Ako je **uvjetovanost** matrice **mala**, mora li onda rješenje izračunato računalom biti **dobro**?

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

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

Na tom sustavu smo pokazali korisnost **zamjene** jednadžbi.

Za vježbu izračunajte da je

$$\text{cond}_2(A) = \frac{30000001 + 10001\sqrt{499980001}}{199980000} \approx 2.61839.$$

Dakle, ovo je **dobro** uvjetovan sustav.

Primjer (nastavak)

Međutim, u Gaussovima eliminacijama **bez pivotiranja**,

☛ u prethodnom sustavu je nešto “**pošlo po zlu**”! Što?

Na **bitnom mjestu** u računu došlo je do “**underflow-a**”, tj.

☛ **mali** broj je pretvoren u **nulu**,

i više **ne možemo** govoriti o **malim relativnim** perturbacijama!

Za **razliku** od toga, s **parcijalnim** pivotiranjem

☛ **nije** bilo nikakvih problema — dobivamo **malu** grešku.

Dakle, ponašanje izračunatog rješenja **bitno** ovisi o **algoritmu**!

☛ **Gdje** se ta “razlika” **vidi**?

Završni komentar — perturbacije i algoritmi

Uočite još da pivotiranje **ne mijenja** uvjetovanost matrice A (bar u 2-normi), jer je

$$\text{cond}_2(PAQ) = \text{cond}_2(A).$$

Ključna **razlika** između algoritama s **raznim** matricama P i Q :

- različite PAQ imaju različite faktore L, R u $PAQ = LR$.
- Zato **obratna** analiza grešaka zaokruživanja daje bitno **različite ocjene** na perturbacije za **razne** algoritme!

Zadatak. Izračunajte $\kappa_2(A)$ i LR faktorizacije matrica PAQ , za **sve** moguće zamjene redaka P i zamjene stupaca Q , gdje je

$$A = \begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}, \quad |\varepsilon| < 1.$$