

# *Numerička matematika*

## *3. predavanje*

Saša Singer

`singer@math.hr`

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

PMF – Matematički odsjek, Zagreb

# Sadržaj predavanja

- Rješavanje linearnih sustava (nastavak):
  - Pivotiranje u Gaussovima eliminacijama (ponavljanje).
  - 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.
  - Obratna analiza grešaka i pivotiranje.
  - Pivotni rast kao mjera nestabilnosti.
  - Hilbertove matrice.
  - Uloga reziduala i iterativno poboljšanje rješenja.

# Informacije

Moja web stranica za **Numeričku matematiku** je

[http://web.math.hr/~singer/num\\_mat/](http://web.math.hr/~singer/num_mat/)

Tamo su kompletna **predavanja** iz prošlih godina, a stizat će i **nova** (kako nastaju).

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

[http://web.math.hr/~singer/num\\_mat/num\\_mat1.pdf](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](http://web.math.hr/~singer/num_mat/num_mat2.pdf)

# Informacije — demonstratori

Kolegij “Numerička matematika” ima **dvije demonstratorice**:

● **Angela Bašić–Šiško**

● termin: **ponedjeljak, 12–14.**

● e-mail: **[abasic@student.math.hr](mailto:abasic@student.math.hr)**

● **Mia Jukić**

● termin: **srijeda, 14–16.**

● e-mail: **[mia.jukic2@gmail.com](mailto:mia.jukic2@gmail.com)**

Demosice lijepo **mole** da im se **najavite** mailom bar **dan ranije!**

● Sastanak za demonstrature je pred **oglasnom pločom**  
(bar zasad).

# Pivotiranje u Gaussovima eliminacijama

# 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** 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  $m_{ik}$  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. malo 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 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, 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** rješenja. 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  $\det L = 1$ , pa je regularnost matrice  $A$  **ekvivalentna** regularnosti matrice  $R$ , jer je

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

# 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,  $i = 2, j = 2, \dots$



## 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  $r_{nn}$ .

# LR faktorizacija — nalaženje (nastavak)

**Napomena.** 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  $R$ .

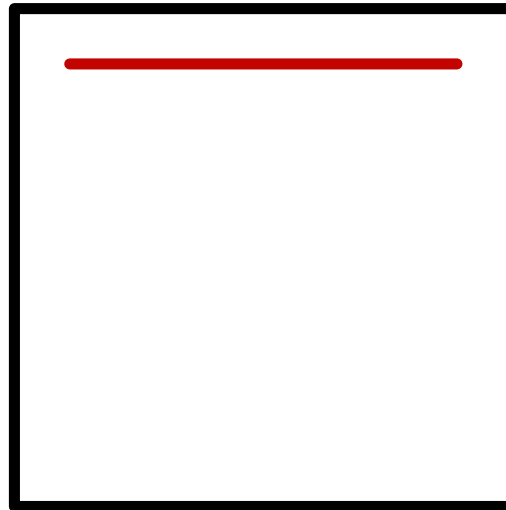
Primijetite,  $r_{nn} \neq 0$  treba samo za **povratnu** supstituciju.

**Pitanje:** Kojim se **redom** računaju elementi 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).

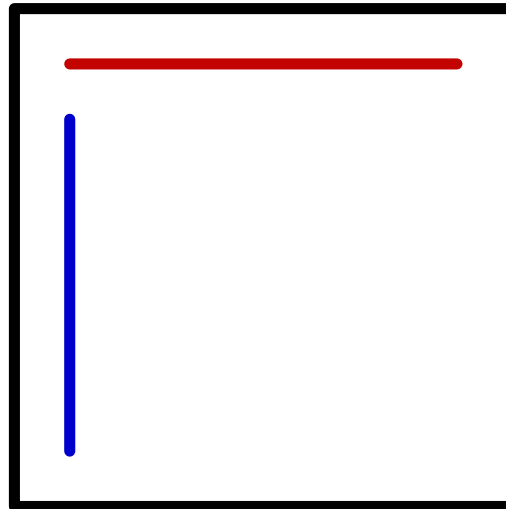
# LR faktorizacija — poredak nalaženja

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



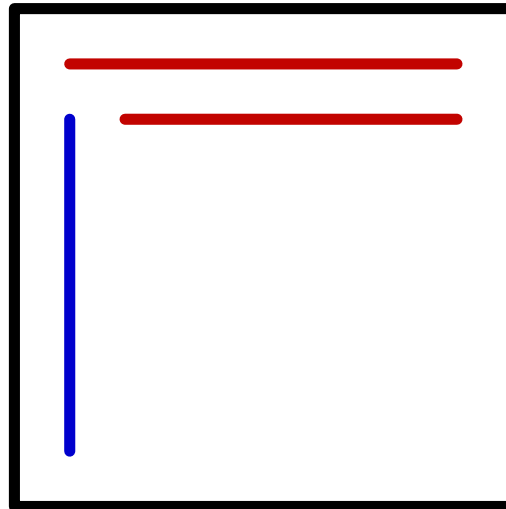
# LR faktorizacija — poredak nalaženja

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



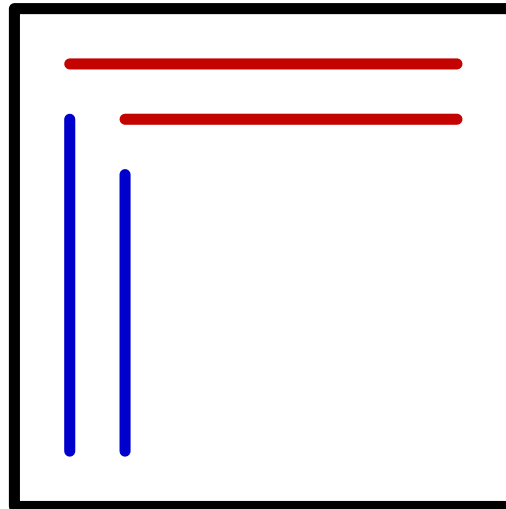
# LR faktorizacija — poredak nalaženja

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



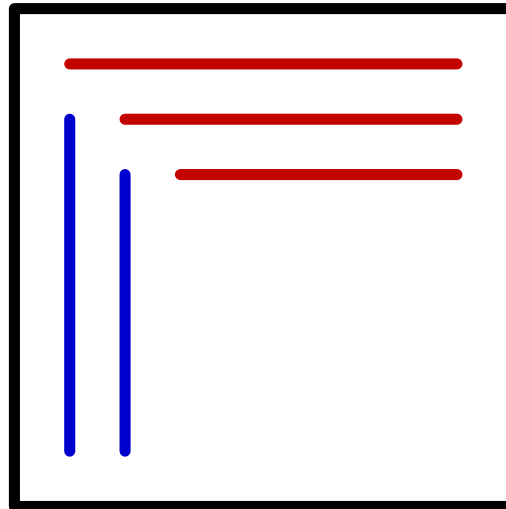
# LR faktorizacija — poredak nalaženja

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



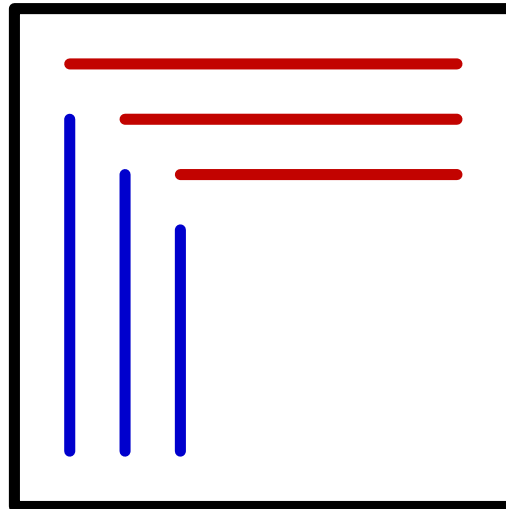
# LR faktorizacija — poredak nalaženja

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



# LR faktorizacija — poredak nalaženja

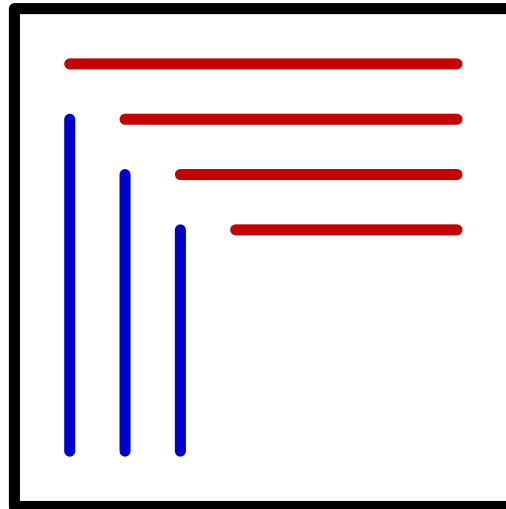
Poredak računanja elemenata — plavo za  $L$  i crveno za  $R$ :





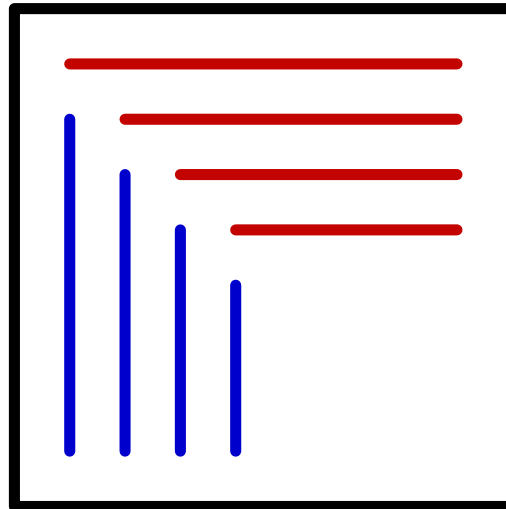
# LR faktorizacija — poredak nalaženja

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



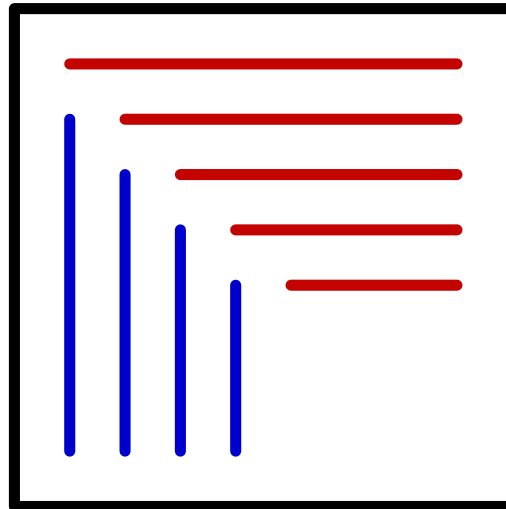
# LR faktorizacija — poredak nalaženja

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



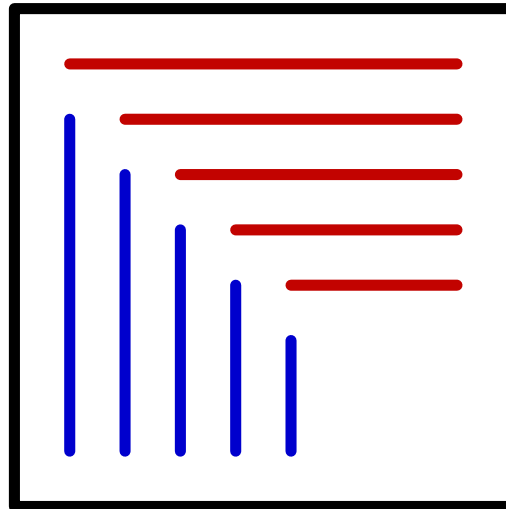
# LR faktorizacija — poredak nalaženja

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



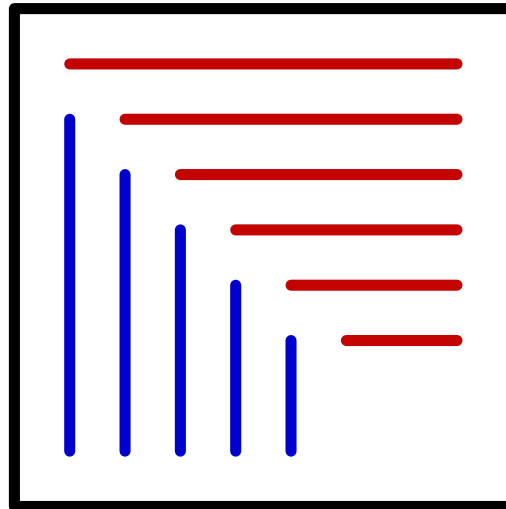
# LR faktorizacija — poredak nalaženja

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



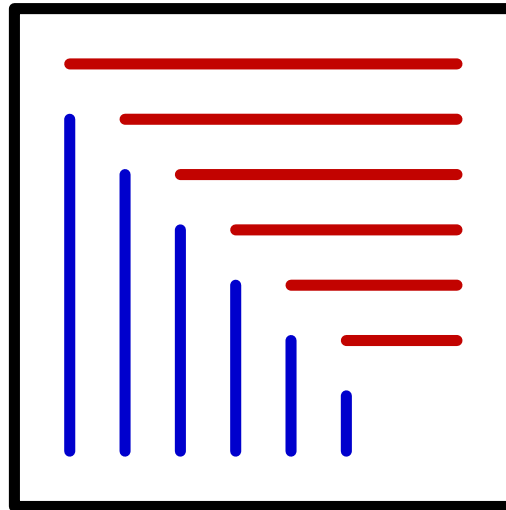
# LR faktorizacija — poredak nalaženja

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



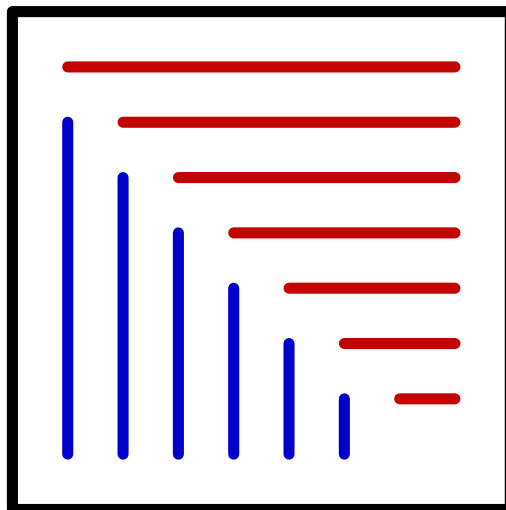
# LR faktorizacija — poredak nalaženja

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



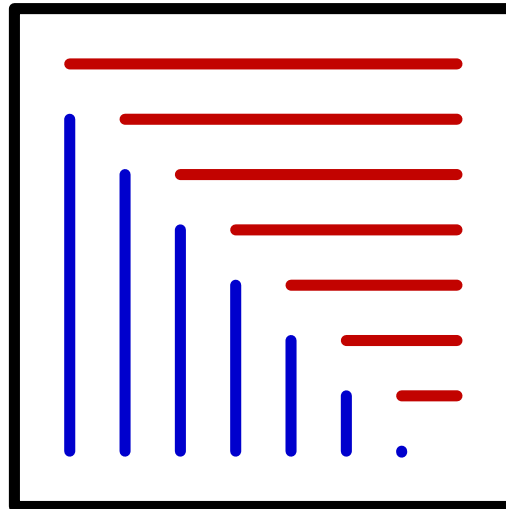
# LR faktorizacija — poredak nalaženja

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



# LR faktorizacija — poredak nalaženja

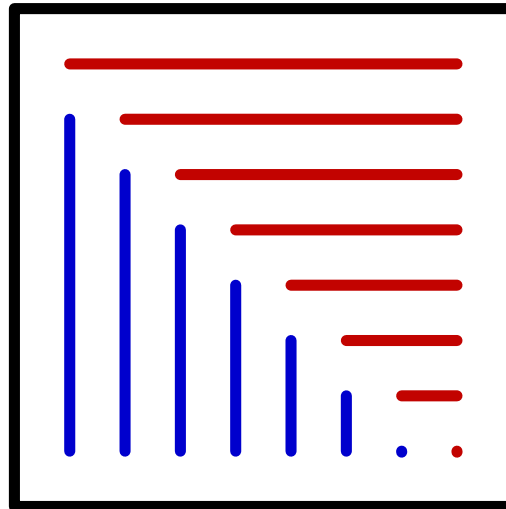
Poredak računanja elemenata — plavo za  $L$  i crveno za  $R$ :





# 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).

# Egzistencija i jedinstvenost LR faktorizacije

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

**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 **nije jedinstvena**.

**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  $r$ ,  $\ell$ . 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  $k = n - 1$

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

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  $\ell$  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  $\ell_{21}$  je  $0 \cdot \ell_{21} = 0$  (u skladu s prethodnim dokazom), pa element  $\ell_{21}$  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  $\ell_{21}$  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 Gaussovih 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

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

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

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 (slijeva) 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 jednačbe

🔴 prvo 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.

**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

- da 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 konzistentnoj normi vrijedi

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

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  $\alpha A$ .

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  $\Delta b$  vektora  $b$ , odnosno,  $\Delta A$  matrice  $A$ , su relativno po normi **odozgo** ograđene nekim brojem  $\varepsilon$ , tj. vrijedi

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

**Komentar.** Ako je  $\varepsilon$  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 onom 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  $\Delta x$ . Izlazi

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

**Ako** je  $\varepsilon \kappa(A) < 1$ , a to znači 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.** 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).

# Komentar rezultata teorije perturbacija

Uočimo da sve ocjene vrijede

• samo za “dovoljno male” perturbacije matrice  $A$ .

Na primjer, 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?



## 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

$$\text{cond}_2(A) = \frac{300000001 + 10001\sqrt{499980001}}{199980000} \approx 2.61839.$$

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 o perturbacijama

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 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

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

# Rezultati obratne analize grešaka zaokruživanja

# Obratna ocjena za LR faktorizaciju

**Teorem.** U aritmetici računala računamo LR faktorizaciju zadane matrice  $A$ , reda  $n$ . Pretpostavimo da je algoritam uspješno završio,

- bez pojave prevelikih ili premalih brojeva koji nisu prikazivi,
- i bez pokušaja dijeljenja s nulom.

Izračunati trokutasti faktori  $\hat{L}$  i  $\hat{R}$  onda zadovoljavaju

$$\hat{L} \hat{R} = A + \Delta A, \quad |\Delta A| \leq \gamma_n |\hat{L}| |\hat{R}|,$$

gdje je  $\gamma_n$  standardna oznaka za mjeru grešaka zaokruživanja

$$\gamma_n := \frac{nu}{1 - nu}.$$



# Obratna ocjena za rješenje sustava

**Teorem.** U aritmetici računala računamo rješenje linearnog sustava  $Ax = b$ , s matricom  $A$ , reda  $n$ .

Uz iste pretpostavke kao u prošlom teoremu, neka su

- $\hat{L}$  i  $\hat{R}$  izračunati trokutasti faktori u LR faktorizaciji matrice  $A$ ,
- i neka je  $\hat{x}$  izračunato rješenje sustava  $Ax = b$ .

Onda postoji perturbacija  $\Delta A$  matrice  $A$ , za koju vrijedi

$$(A + \Delta A) \hat{x} = b, \quad |\Delta A| \leq \gamma_{3n} |\hat{L}| |\hat{R}|.$$

Za zaključak o relativnoj grešci, fali nam još

- neka veza između matrica  $|\hat{L}| |\hat{R}|$  i  $|A|$ .

# Put do relativnih ocjena

U idealnom slučaju, željeli bismo da je

$$|\Delta A| \leq u |A|.$$

To bi odgovaralo grešci zaokruživanja koju napravimo samo

- početnim spremanjem elemenata matrice  $A$  u memoriju računala.

No, to nije realistično. Nad svakim elementom matrice  $A$

- vrši se još najviše  $n$  aritmetičkih operacija (za  $A = LR$ ).

Zato ne možemo očekivati nešto bolje od ocjene oblika

$$|\Delta A| \leq c_n u |A|,$$

gdje je  $c_n$  “konstanta” reda veličine  $n$ , odnosno,  $c_n u \approx c \gamma_n$ .



## Relativne ocjene — idealni slučaj

Na primjer, takvu ocjenu **dobivamo** pod uvjetom da  $\hat{L}$  i  $\hat{R}$  zadovoljavaju da je

$$|\hat{L}| |\hat{R}| = |\hat{L}\hat{R}|.$$

To je **idealni** slučaj — i, naravno, **ne vrijedi** uvijek.

Ako to **vrijedi**, onda iz **prvog** teorema izlazi

$$|\hat{L}| |\hat{R}| = |\hat{L}\hat{R}| = |A + \Delta A| \leq |A| + |\Delta A| \leq |A| + \gamma_n |\hat{L}| |\hat{R}|,$$

pa, prebacivanjem članova dobivamo

$$|\hat{L}| |\hat{R}| \leq \frac{1}{1 - \gamma_n} |A|.$$

## Relativne ocjene — idealni slučaj (nastavak)

Ako tu relaciju uvrstimo u drugi teorem, onda izlazi

$$(A + \Delta A) \hat{x} = b, \quad |\Delta A| \leq \frac{\gamma_{3n}}{1 - \gamma_n} |A|,$$

tj. **izračunato** rješenje  $\hat{x}$  ima

- malu obratnu **relativnu** grešku po **komponentama**.

Za koje matrice **vrijedi** “idealno”  $|\hat{L}| |\hat{R}| = |\hat{L}\hat{R}|$ ?

Na primjer, ako LR faktorizacija daje **nenegativne** elemente u faktorima  $L$  i  $R$ , tj. vrijedi  $L, R \geq 0$  (po elementima).

- Takve su tzv. **totalno nenegativne** ili **totalno pozitivne** matrice — i zato se kod njih **ne pivotira** u GE ili LR.

Javljaju se, na primjer, kod **splajn interpolacije** (v. kasnije).

# Što je bitno za stabilnost?

Iz prethodna dva teorema slijedi da stabilnost LR faktorizacije i rješenja linearnog sustava

- ne ovisi o veličini multiplikatora,
- već o veličini elemenata koji se javljaju u matrici  $|\hat{L}| |\hat{R}|$ , relativno obzirom na odgovarajuće elemente matrice  $A$  (toliko kraćenje može nastati računanjem  $\hat{L}\hat{R} \approx A$ ).

Naime, ta matrica  $|\hat{L}| |\hat{R}|$

- može imati male elemente, iako su joj multiplikatori  $m_{ij} = l_{ij}$  veliki — pripadni elementi u  $\hat{R}$  su jako mali,
- ali može imati i velike elemente, a da su joj multiplikatori reda veličine 1 — pripadni elementi u  $\hat{R}$  su veliki.

# Analiza i procjena stabilnosti algoritma

Za lakšu analizu, ne gleda se po svim elementima, već se analizira omjer normi

$$\frac{\|\hat{L}\| \|\hat{R}\|}{\|A\|}.$$

**Bitno:** Ovaj omjer ovisi o algoritmu kojim računamo LR faktorizaciju matrice  $A$ !

Kod LR faktorizacije bez pivotiranja, ovaj omjer normi može biti proizvoljno velik. Na primjer, pokažite da je za matricu

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}$$

taj omjer jednak  $\varepsilon^{-1}$ .

# Stabilnost parcijalnog pivotiranja

Kod **parcijalnog** pivotiranja ( $PA = LR$ ) znamo da vrijedi

$$|\ell_{ij}| \leq 1 \quad \text{za sve } i \geq j.$$

Kad uvrstimo  $m_{ik} = \ell_{ik}$  u formule **transformacije** elemenata

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

indukcijom po koracima eliminacije, dobivamo da vrijedi

$$|r_{ij}| \leq 2^{i-1} \max_{k \leq i} |(PA)_{kj}| \leq 2^{i-1} \max_k |a_{kj}|,$$

jer element  $r_{ij}$  nastaje nakon  $i - 1$  koraka eliminacije.

Dakle, kod **parcijalnog** pivotiranja

•  $L$  je **malen**, a  $R$  je **ograđen relativno** obzirom na  $A$ .

# 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 tzv. pivotnog rasta.

Pivotni rast ili “faktor rasta”, u oznaci  $\rho_n$ , je omjer

- najvećeg (po apsolutnoj vrijednosti) elementa u svim koracima eliminacije — ovisi o pivotiranju,
- i (apsolutno) najvećeg elementa u originalnoj matrici  $A$ ,

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

Intuitivno je jasno da nije dobro da elementi jako narastu po apsolutnoj vrijednosti, jer to može dovesti do gubitka točnosti. To je analogno “uništavanju” polaznih jednadžbi!

# Pivotni rast — parcijalno pivotiranje

Koliki je pivotni rast  $\rho_n^p$  kod parcijalnog pivotiranja?

Transformacije elemenata u  $k$ -tom koraku eliminacija su

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}.$$

Kod parcijalnog pivotiranja, za multiplikatore  $m_{ik}$  vrijedi

$$|m_{ik}| \leq 1,$$

pa je

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

Nakon  $n - 1$  koraka algoritma, ova ocjena daje pivotni rast  $\rho_n^p$

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



## Pivotni rast — parcijalno pivotiranje (nastavak)

Već je J. H. Wilkinson primijetio da se taj pivotni rast može dostići 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 mnogo bolje od očekivanog.

# Pivotni rast — potpuno pivotiranje

J. H. Wilkinson je 1961. godine dokazao da za pivotni rast  $\rho_n^c$  kod **potpunog** pivotiranja vrijedi **ocjena odozgo**

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

i da se ta ocjena **ne može** dostići.

Dugo se mislilo da vrijedi

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

Međutim, **nađeni** su kontraprimjeri matrica kad to **ne vrijedi**.

● 1991. g. — matrica reda **13** za koju je  $\rho_{13}^c = 13.0205$ ,

● 1992. g. — matrica reda **25** za koju je  $\rho_{25}^c = 32.986341$ .

Točno ponašanje  $\rho_n^c$  je **otvoren** problem!

# Ocjena stabilnosti preko faktora rasta

Tradicionalno, **obratna** analiza greške izražava se preko **pivotnog rasta** ili **faktora rasta** (engl. growth factor)

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

U procesu **Gaussovih** eliminacija, očito vrijedi da je

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

što daje ogradu za  $R$ , **relativno** obzirom na  $A$ . Naravno, faktor rasta  $\rho_n$  ovisi o algoritmu kojim računamo.

Može se naći i precizna ocjena **omjera normi** odozgo, preko **faktora rasta**, i obratno (ovisi o algoritmu i izabranoj normi).

# Obratna ocjena za sustav preko faktora rasta

**Teorem** (Wilkinson). Neka je  $A$  regularna kvadratna matrica reda  $n$  i neka je  $\hat{x}$  izračunato rješenje sustava  $Ax = b$

- Gaussovima eliminacijama s parcijalnim pivotiranjem u aritmetici računala.

Tada vrijedi

$$(A + \Delta A) \hat{x} = b, \quad \|\Delta A\|_{\infty} \leq n^2 \gamma_{3n} \rho_n \|A\|_{\infty}. \quad \blacksquare$$

U prethodnom teoremu, pretpostavka da koristimo parcijalno pivotiranje nije nužna.

Naime, slično vrijedi i za Gaussove eliminacije bez pivotiranja, samo s malo drugačijim oblikom faktora ispred  $\|A\|_{\infty}$ .

Naravno, faktor rasta  $\rho_n$  može biti puno veći!

# Hilbertove matrice

# Hilbertova matrica

**Primjer.** Kod aproksimacije polinomima javljaju se linearni sustavi oblika

$$H_n x = b,$$

gdje je  $H_n$  Hilbertova matrica reda  $n$ ,  $(H_n)_{ij} = \frac{1}{i+j-1}$ , ili

$$H_n = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix}.$$

# Hilbertova matrica

Da bismo ispitali **točnost** rješenja, stavimo **desnu** stranu

$$b(i) := \sum_{j=1}^n H_n(i, j) = \sum_{j=1}^n \frac{1}{i+j-1}, \quad i = 1, \dots, n,$$

tako da je egzaktno **rješenje** sustava  $x^T = [1, 1, \dots, 1]$ .

Što možemo očekivati od rješenja takvog sustava?

Pogled na **Frobeniusovu normu** matrice  $H_n$  kaže da ona **nije naročito velika**,

$$\|H_n\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n \left| \frac{1}{i+j-1} \right|^2} \leq \sqrt{\sum_{i=1}^n \sum_{j=1}^n 1} = n.$$

# Hilbertova matrica — uvjetovanost

Međutim ... ne treba gledati samo normu matrice!!!

Uvjetovanost Hilbertovih matrica je vrlo visoka:

$n$	$\kappa_2(H_n)$	$n$	$\kappa_2(H_n)$	$n$	$\kappa_2(H_n)$
2	$1.928 \cdot 10^1$	9	$4.932 \cdot 10^{11}$	15	$6.117 \cdot 10^{20}$
3	$5.241 \cdot 10^2$	10	$1.603 \cdot 10^{13}$	16	$2.022 \cdot 10^{22}$
4	$1.551 \cdot 10^4$	11	$5.231 \cdot 10^{14}$	17	$6.697 \cdot 10^{23}$
5	$4.766 \cdot 10^5$	12	$1.713 \cdot 10^{16}$	18	$2.221 \cdot 10^{25}$
6	$1.495 \cdot 10^7$	13	$5.628 \cdot 10^{17}$	19	$7.376 \cdot 10^{26}$
7	$4.754 \cdot 10^8$	14	$1.853 \cdot 10^{19}$	20	$2.452 \cdot 10^{28}$
8	$1.526 \cdot 10^{10}$				



## Hilbertova matrica — rješenje za $n = 2, 5$

Za sustav  $H_n x = b$  s Hilbertovom matricom, za razne  $n$ ,

• GE s parcijalnim pivotiranjem, u extended točnosti, umjesto svih jedinica u rješenju, dobivamo ove rezultate:

Red  $n = 2$

$$x(1) = 1.0000000000000000 \quad x(2) = 1.0000000000000000$$

Red  $n = 5$

$$x(1) = 1.0000000000000000 \quad x(4) = 0.99999999999999990$$

$$x(2) = 0.9999999999999999 \quad x(5) = 1.00000000000000005$$

$$x(3) = 1.00000000000000007$$

Uvjetovanost:  $\approx 4.766 \cdot 10^5$ .

## Hilbertova matrica — rješenje za $n = 10$

$$x(1) = 1.00000000000003436$$

$$x(6) = 0.9999999294831902$$

$$x(2) = 0.9999999999710395$$

$$x(7) = 1.0000001151701616$$

$$x(3) = 1.0000000006068386$$

$$x(8) = 0.9999998890931838$$

$$x(4) = 0.9999999945453735$$

$$x(9) = 1.0000000580638087$$

$$x(5) = 1.0000000258066880$$

$$x(10) = 0.9999999872591526$$

Uvjetovanost:  $\approx 1.603 \cdot 10^{13}$ .

# Hilbertova matrica — rješenje za $n = 15$

$$x(1) = 1.00000000005406387 \quad x(9) = 1.0952919444304200$$

$$x(2) = 0.9999999069805858 \quad x(10) = 0.8797820363884070$$

$$x(3) = 1.0000039790948573 \quad x(11) = 1.0994671444236333$$

$$x(4) = 0.9999257525660447 \quad x(12) = 0.9508102511158300$$

$$x(5) = 1.0007543452271621 \quad x(13) = 1.0106027108940050$$

$$x(6) = 0.9953234190795597 \quad x(14) = 1.0012346841153261$$

$$x(7) = 1.0188643674562383 \quad x(15) = 0.9992252029377023$$

$$x(8) = 0.9487142544341838$$

Uvjetovanost:  $\approx 6.117 \cdot 10^{20}$ .

# Hilbertova matrica — rješenje za $n = 20$

$x(1) =$	1.0000000486333029	$x(11) =$	231.3608002738048500
$x(2) =$	0.9999865995557111	$x(12) =$	-60.5143391625873562
$x(3) =$	1.0008720556363132	$x(13) =$	-57.6674972682886125
$x(4) =$	0.9760210562677670	$x(14) =$	5.1760567992057506
$x(5) =$	1.3512820600312678	$x(15) =$	8.7242780841976215
$x(6) =$	-2.0883247796748707	$x(16) =$	210.1722288687690970
$x(7) =$	18.4001541798146106	$x(17) =$	-413.9544667202651170
$x(8) =$	-63.8982130462650081	$x(18) =$	349.7671855031355400
$x(9) =$	161.8392478869777220	$x(19) =$	-142.9134532513063250
$x(10) =$	-254.7902985140752950	$x(20) =$	25.0584794423327874

Uvjetovanost  $\approx 2.452 \cdot 10^{28}$ .

# Uvjetovanost Hilbertovih matrica

Može se pokazati da za **uvjetovanost** Hilbertove matrice  $H_n$  vrijedi formula

$$\kappa_2(H_n) \approx \frac{(\sqrt{2} + 1)^{4n+4}}{2^{15/4} \sqrt{\pi n}} \quad \text{za } n \rightarrow \infty.$$

Dakle, iako Hilbertove matrice imaju “**idealna**” svojstva,

• **simetrične**, **pozitivno definitne** (čak **totalno pozitivne** = determinanta **svake** kvadratne podmatrice je **pozitivna**), njihova uvjetovanost **katastrofalno brzo raste!**

“**Krivci**” za to su elementi **inverza**  $H_n^{-1}$ .

## Inverz Hilbertove matrice

Recimo,  $H_5^{-1}$  izgleda ovako:

$$H_5^{-1} = \begin{bmatrix} 25 & -300 & 1050 & -1400 & 630 \\ -300 & 4800 & -18900 & 26880 & -12600 \\ 1050 & -18900 & 79380 & -117600 & 56700 \\ -1400 & 26880 & -117600 & 179200 & -88200 \\ 630 & -12600 & 56700 & -88200 & 44100 \end{bmatrix} .$$

A kako tek izgledaju elementi  $H_{20}^{-1}$ ?

# Inverz Hilbertove matrice

Elementi inverza  $H_n^{-1}$  Hilbertove matrice mogu se eksplicitno izračunati u terminima binomnih koeficijenata

$$(H_n^{-1})_{ij} = (-1)^{i+j} (i + j - 1) \cdot \binom{n+i-1}{n-j} \binom{n+j-1}{n-i} \binom{i+j-2}{i-1}^2.$$

Lako se vidi da ovi elementi vrlo brzo rastu za malo veće  $n$ .

Pogledajte

<http://mathworld.wolfram.com/HilbertMatrix.html>

# Rezidual i iterativno poboljšanje rješenja



## Rezidual približnog rješenja

Kad rješenje sustava  $Ax = b$  računamo približno (računalom),

☛ umjesto pravog rješenja  $x$ , dobivamo približno rješenje  $\hat{x}$ .

Veličinu

$$r = r(\hat{x}) = b - A\hat{x},$$

zovemo **rezidual** izračunatog rješenja  $\hat{x}$ .

**Napomena.** Egzaktni rezidual pravog rješenja  $x$  je  $r = 0$ !

Međutim, ako je (egzaktni) rezidual

- ☛ **velik**, onda sigurno **nismo blizu** pravom rješenju,
- ☛ ali rezidual može biti **malen**, a da izračunato rješenje  $\hat{x}$  sustava nije **ni blizu** pravom rješenju  $x$ .

## Izračunati rezidual

Primjer. Gledamo **izračunato** rješenje  $\hat{x}$  linearnog sustava

$$H_{20}x = b$$

s desnom stranom takvom da je  $x^T = [1, 1, \dots, 1]$ .

Kad računamo u **extended** točnosti,

• **izračunati** rezidual  $\hat{r} = b - A\hat{x}$  je **nula-vektor** (kraćenje), a komponente rješenja  $\hat{x}$  su bile u **stotinama**.

Ovo ponašanje je u **skladu** s **teorijom perturbacija**, koja

- **garantira mali** rezidual  $r$ , za iole razumne perturbacije,
- a **izračunato rješenje  $\hat{x}$**  može biti **katastrofalno**, ako je **uvjetovanost** matrice  $A$  **velika**.

# Uloga reziduala — poboljšanje točnosti

Reziduali se mogu iskoristiti za **poboljšavanje** netočnog rješenja linearnog sustava.

To se obično provodi u **tri** koraka — može i **iterativno**.

- **Izračuna** se rezidual  $r = b - A\hat{x}$ , pri čemu je  $\hat{x}$  **izračunato** (ili približno) rješenje sustava.
- **Riješi** se sustav  $Ad = r$ , gdje je  $d$  **korekcija**.
- **Korekcija** se **doda** izračunatom rješenju

$$y = \hat{x} + d,$$

što bi **trebalo** dati **bolje** rješenje  $y$ . Egzaktno bi bilo  $r(y) = b - A\hat{x} - Ad = r - r = 0$ .

Postupak se može **ponoviti** s  $y$ , umjesto  $\hat{x}$ .

## Računanje reziduala — mora u većoj točnosti

Ovo ima smisla **samo** ako se **prvi** korak

- računanje reziduala  $r = b - A\hat{x}$
- radi u **većoj** točnosti od **one** u kojoj je izračunat  $\hat{x}$ .

To je nužno zbog **kraćenja**, tako da

- **izračunati**  $\hat{r}$  ima **dovoljnu** relativnu točnost.

Preostala **dva** koraka standardno se rade

- u “**običnoj**” točnosti, kao za  $\hat{x}$  (baš to je **ideja!**).

Tipično se računanje reziduala radi u

- **dvostrukoj** točnosti — jedinična greška zaokruživanja je reda veličine  $u^2$ , obzirom na **jednostruku**.

Na pr. **double**, prema **single**.