

Reševanje sistemov linearnih enačb

Meteorologija z geofiziko, I. stopnja

<http://ucilnica.fmf.uni-lj.si/>

Matrični zapis sistema linearnih enačb

Sistem m linearnih enačb z n neznankami

$$a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n-1} x_{n-1} + a_{1n} x_n = b_1$$

$$a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n-1} x_{n-1} + a_{2n} x_n = b_2$$

$$\vdots$$

$$a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn-1} x_{n-1} + a_{mn} x_n = b_m,$$

krajše zapišemo v obliki

$$A \mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{m \times n}, \quad \mathbf{x} \in \mathbb{R}^{n \times 1}, \quad \mathbf{b} \in \mathbb{R}^{m \times 1}.$$

Pri tem je

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

V veliki večini primerov (vsaj v okviru tega poglavja) bomo predpostavili, da je $m = n$.

Posebej pomembni bodo sistemi, katerih matrike imajo določeno strukturo:

- zgornje trikotni sistemi (matrika je **zgornja trikotna**)

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix},$$

- spodnje trikotni sistemi (matrika je **spodnja trikotna**)

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix},$$

- tridiagonalni sistemi (matrika je **tridiagonalna**)

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ 0 & \cdots & 0 & a_{nn-1} & a_{nn} \end{bmatrix},$$

- diagonalni sistemi (matrika je **diagonalna**)

$$A = \text{diag}(a_{11}, a_{22}, \dots, a_{nn}).$$

Vektorske in matrične norme

Definicija

Vektorska norma je preslikava $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}^+ \cup \{0\}$, za katero je

1 $\|\mathbf{x}\| = 0$ natanko tedaj, ko je $\mathbf{x} = \mathbf{0}$,

2 $\|\alpha \mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$,

3 $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$,

za vsak $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ in $\alpha \in \mathbb{R}$.

Pomembnejše vektorske norme so

- $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$ (1-norma),
- $\|\mathbf{x}\|_2 = (\sum_{i=1}^n x_i^2)^{1/2}$ (2-norma),
- $\|\mathbf{x}\|_\infty = \max_{i=1,2,\dots,n} |x_i|$ (∞ -norma ali max norma).

Definicija

Matrična norma je preslikava $\| \cdot \| : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^+ \cup \{0\}$, za katero je

- 1 $\|A\| = 0$ natanko tedaj, ko je $A = 0$,
- 2 $\|\alpha A\| = |\alpha| \cdot \|A\|$,
- 3 $\|A + B\| \leq \|A\| + \|B\|$,
- 4 $\|AC\| \leq \|A\| \|C\|$,

za vsak $A, B \in \mathbb{R}^{m \times n}$, $C \in \mathbb{R}^{n \times p}$ in $\alpha \in \mathbb{R}$.

Najbolj znane matrične norme so:

- $\|A\|_1 = \max_{j=1,2,\dots,n} \left(\sum_{i=1}^m |a_{ij}| \right)$ (1-norma),
- $\|A\|_\infty = \max_{i=1,2,\dots,m} \left(\sum_{j=1}^n |a_{ij}| \right)$ (∞ -norma),
- $\|A\|_2 = \max_{i=1,2,\dots,n} \sqrt{\lambda_i(A^\top A)}$ (spektralna norma),
- $\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{1/2}$ (Frobeniusova norma).

Za obrnljivo matriko A definiramo **število občutljivosti**

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

Pri reševanju sistemov linearnih enačb nas pogosto zanima, kako je rešitev sistema $A\mathbf{x} = \mathbf{b}$ občutljiva na motnje v A in $\mathbf{b}$.

Lema

Naj bo $A\mathbf{x} = \mathbf{b}$ in $(A + \delta A)(\mathbf{x} + \delta \mathbf{x}) = \mathbf{b} + \delta \mathbf{b}$. Če je $\|A^{-1}\| \|\delta A\| < 1$, je

$$\frac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\kappa(A)}{1 - \kappa(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta \mathbf{b}\|}{\|\mathbf{b}\|} \right).$$

Primer

Naj bo

$$A = \left(\frac{1}{i+j-1} \right)_{i,j=1}^{10}, \quad \mathbf{b} = A [1, 1, \dots, 1]^\top.$$

Točna rešitev sistema $A\mathbf{x} = \mathbf{b}$ je torej $\mathbf{x} = [1, 1, \dots, 1]^\top$. Z Matlabovim algoritmom za reševanje sistemov linearnih enačb pa dobimo rešitev $\tilde{\mathbf{x}}$, za katero je $\|\mathbf{x} - \tilde{\mathbf{x}}\|_2 \approx 8.7 \cdot 10^{-4}$.

Direktno vstavljanje

Oglejmo si najprej preprost algoritem za reševanje sistema linearnih enačb s **spodnjo trikotno matriko**:

```
function x=direktno(L,b)
%DIREKTNO resuje spodnji trikotni sistem
%x=DIREKTNO(L,b)
%L je spodnja trikotna matrika
%b je desna stran
%x je resitev sistema

n=size(L,1);
x=zeros(n,1);
for i=1:n
    x(i)=(b(i)-L(i,1:i-1)*x(1:i-1))/L(i,i);
end
```

Podoben algoritem lahko izvedemo za sistem linearnih enačb z
zgornjo trikotno matriko:

```
function x=obratno(U,b)
%OB RATNO resuje spodnji trikotni sistem
%x=OB RATNO(U,b)
%U je zgornja trikotna matrika
%b je desna stran
%x je resitev sistema

n=size(U,1);
x=zeros(n,1);
for i=n:-1:1
    x(i)=(b(i)-U(i,i+1:n)*x(i+1:n))/U(i,i);
end
```

- Časovna zahtevnost obeh algoritmov je $\mathcal{O}(n^2)$. Preštejmo število operacij $(+, -, *, /)$ za direktno vstavljanje:

$$\sum_{i=1}^n (1 + i - 1 + i - 2 + 1) = -n + 2 \sum_{i=1}^n i = -n + n(n+1) = n^2.$$

- Algoritma sta osnova za reševanje sistema linearnih enačb s poljubno matriko A .

LU razcep

Osnovna ideja večine algoritmov za reševanje sistema linearnih enačb je

- Problem **prevedite na enostavnejšega**, ki ima posebno obliko.
- Ekonomično **rešite enostavnejši sistem**.
- Iz rešitve enostavnejšega **pridobite rešitev originalnega sistema**.

Izrek

Za matriko $A \in \mathbb{R}^{n \times n}$ sta ekvivalentni izjavi:

- a) Obstaja enolični razcep $A = L U$ (LU razcep), kjer je L spodnja trikotna matrika z enkami na diagonali in U nesingularna zgornja trikotna matrika.
- b) Vse vodilne podmatrike $A(1 : k, 1 : k)$, $k = 1, 2, \dots, n$, so nesingularne.

Izrek seveda ne zagotavlja, da LU razcep obstaja za vsako matriko. Preprost protiprimer je na primer že matrika

$$A = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix}.$$

Velja pa naslednji izrek.

Izrek

Za vsako nesingularno matriko $A \in \mathbb{R}^{n \times n}$, najdemo tako permutacijsko matriko P , da obstaja LU razcep za matriko $P A$.

Za primer s prejšnje prosojnice je

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

in je

$$P A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}.$$

- Matrika P v resnici pove, kako bi morali na začetku premenjati **vrstice** matrike A , da za tako dobljeno matriko obstaja LU razcep.
- V splošnem je možnih veliko matrik P . Najbolj znan je postopek za določitev ene od njih je **delno pivotiranje**.
- Prejšnji izrek torej lahko preberemo tudi takole: za vsako nesingularno matriko A s postopkom delnega pivotiranja pridemo do LU razcepa matrike PA .

Osnovne prijeme pri postopku LU racepa z delnim pivotiranjem **na j -tem koraku** lahko povzamemo takole:

- Naj bo $A^{(j)}$ matrika A na j -tem koraku. V j -tem stolpcu v vrsticah od j do n poiščite največji element po absolutni vrednosti, torej

$$\max_{j \leq i \leq n} |a_{ij}^{(j)}|.$$

- Zamenjajte vrstico, kjer je največji element, z j -to vrstico (**delno pivotiranje**).
- Izvedite **Gaussovo eliminacijo** vrstic $j + 1, j + 2, \dots, n$:

$$A^{(j+1)} = L_j A^{(j)},$$

kjer je

$$L_j = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & -\ell_{j+1j} & 1 & \\ & & & \vdots & & \ddots \\ & & & -\ell_{nj} & & & 1 \end{bmatrix}$$

in

$$\ell_{kj} = \frac{a_{kj}^{(j)}}{a_{jj}^{(j)}}.$$

LU razcep z delnim pivotiranjem

Algoritem za postopek *LU* razcepa z delnim pivotiranjem je naslednji:

```
function [L,U,P]=lu_delno(A)
%LU_DELNO je LU razcep z celnim pivotiranjem
%[L,U,P]=LU_DELNO(A)
%A je vhodna matrika
%L je spodnja trikotna z enkami na diagonali
%U je zgornja trikotna
%P je permutacijska

n=size(A,1);
P=eye(n);
L=zeros(n);
U=zeros(n);
```

```

for j=1:n-1
    [M,maxi]=max(abs(A(j:n,j)));
    maxi=maxi+j-1;%popravimo na globalni indeks
    P([j,maxi],:)=P([maxi,j],:);
    A([j,maxi],:)=A([maxi,j],:);
    A(j+1:n,j)=A(j+1:n,j)/A(j,j);
    A(j+1:n,j+1:n)=A(j+1:n,j+1:n)-A(j+1:n,j)*A(j,j+1:n);
end
L=tril(A,-1)+eye(n);
U=triu(A);

```

Primer

Poiščimo LU razcep z delnim pivotiranjem za matriko

$$A = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 0 & 1 \end{bmatrix}.$$

Sledimo zgornjemu algoritmu in dobimo

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & -1/2 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -2 & -2 \\ 0 & 0 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Preštejmo število operacij:

$$\sum_{j=1}^{n-1} \left(\overbrace{n-j}^{\text{/}} + \overbrace{(n-j)^2}^{\text{*}} + \overbrace{(n-j)^2}^{\text{-}} \right) = \sum_{j=1}^{n-1} j + 2 \sum_{j=1}^{n-1} j^2$$
$$= \frac{n(n-1)}{2} + \frac{2}{6}(n-1)n(2n-1) = \frac{2}{3}n^3 - \frac{1}{2}n^2 - \frac{1}{6}n = \mathcal{O}(n^3).$$

Če se dimenzija matrike torej **podvoji**, potrebujemo približno **osemkrat** več operacij.

Uporaba LU razcepa z delnim pivotiranjem

- Reševanje sistema linearnih enačb $A\mathbf{x} = \mathbf{b}$, $A \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$:
 - ▶ $PA = LU$, LU razcep $\mathcal{O}(n^3)$ operacij,
 - ▶ $L\mathbf{y} = P\mathbf{b}$, direktno vstavljanje $\mathcal{O}(n^2)$ operacij,
 - ▶ $U\mathbf{x} = \mathbf{y}$, obratno vstavljanje $\mathcal{O}(n^2)$ operacij.
- Reševanje matričnega sistema $AX = B$, $A \in \mathbb{R}^{n \times n}$, $X \in \mathbb{R}^{n \times p}$, $B \in \mathbb{R}^{n \times p}$:
 - ▶ Problem zapišemo po stolpcih v obliki

$$A [\mathbf{x}_1, \mathbf{x}_2 \dots, \mathbf{x}_p] = [\mathbf{b}_1, \mathbf{b}_2 \dots, \mathbf{b}_p]$$

in rešujemo $A\mathbf{x}_j = \mathbf{b}_j$, $j = 1, 2, \dots, p$.

- ▶ Ponovimo postopek iz prejšnje točke, pri čemer LU razcep naredimo le enkrat.

- Iščemo inverz matrike $A \in \mathbb{R}^{n \times n}$ (zelo redko zares potrebujemo inverz!!!):
 - ▶ Uporabimo prejšnjo točko, kjer je $X = A^{-1}$ in $B = I$ (identiteta).
- Računamo determinanto matrike $A \in \mathbb{R}^{n \times n}$:
 - ▶ Naredimo LU razcep matrike A , torej $PA = LU$.
 - ▶ Upoštevamo zvezo

$$\det PA = \det L \det U = \prod_{i=1}^n u_{ii}$$

in dejstvo, da je $\det P = (-1)^k$, kjer je k število menjav vrstic v matriki A pri delnem pivotiranju.

Primer

Izračunajmo determinanto matrike

$$A = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 0 & 1 \end{bmatrix}$$

z LU razcepom z delnim pivotiranjem. Iz prejšnjega primera dobimo

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & -1/2 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -2 & -2 \\ 0 & 0 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Torej je

$$\det PA = \det L \det U = 1 * (-2).$$

Ker je $\det P = 1$, je torej $\det A = -2$.

Kompletno pivotiranje

- Poleg delnega pivotiranja je znan tudi postopek **kompletnega pivotiranja**.
- Na j -tem koraku v matriki $A^{(j)}$ poiščemo največji element v **celi matriki $A^{(j)}$** , torej

$$\max_{j \leq k, l \leq n} |a_{kl}^{(j)}|$$

in zamenjamo ustrezni vrstici **ter stolpca**.

- Nato nadaljujemo z eliminacijami kot v primeru delnega pivotiranja.
- Razcep s kompletnim pivotiranjem ima obliko

$$PAQ = LU,$$

kjer sta P in Q permutacijski matriki (v matriki Q je informacija o menjavi stolpcev).

- Kompletno pivotiranje večinoma ne izboljša bistveno natančnosti v primerjavi z delnim, zato se v praksi redko uporablja.

Primer

Razcepimo s kompletnim pivotiranjem matriko

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix}.$$

Dobimo

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 1/3 & -1/2 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 3 & 1 & 2 \\ 0 & 2/3 & 1/3 \\ 0 & 0 & 1/2 \end{bmatrix},$$

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Reševanje posebnih sistemov

- Med posebnimi sistemi smo že omenili zgornje in spodnje trikotne, tridiagonalne in diagonalne.
- **Tridiagonalne** bomo obravnavali ob koncu tega razdelka.
- Med zelo pomembnimi so tudi sistemi, pri katerih je matrika A **simetrična pozitivno definitna**.

Definicija

Matrika $A \in \mathbb{R}^{n \times n}$ je simetrična pozitivno definitna, če je $A = A^\top$ in je $\mathbf{x}^\top A \mathbf{x} \geq 0$ za vsak $\mathbf{x} \in \mathbb{R}^n$.

Izrek

Matrika $A \in \mathbb{R}^{n \times n}$ je simetrična pozitivno definitna natanko tedaj, ko je $A = A^\top$ in so vse lastne vrednosti pozitivne.

Za simetrično pozitivno definitno matriko obstaja posebna oblika razcepa, t.i. **razcep Choleskega**.

Izrek

Za simetrično pozitivno definitno matriko A obstaja nesingularna spodnja trikotna matrika V , da je $A = VV^\top$.

Razcep Choleskega

```
function V=cholesky(A)
%CHOLESKY je razcep Choleskega
%V=CHOLESKY(A)
%A je simetricna pozitivno definitna
%V je spodnja trikotna, da je  $V*V'=A$ 
n=size(A,1);
V=zeros(n);
V(1,1)=sqrt(A(1,1));
V(2:n,1)=A(2:n,1)/V(1,1);
for k=2:n
    V(k,k)=sqrt(A(k,k)-V(k,1:k-1)*V(k,1:k-1)');
    for j=k+1:n
        V(j,k)=(A(j,k)-V(k,1:k-1)*V(j,1:k-1)')/V(k,k);
    end
end
```

- Časovna zahtevnost je (brez kvadratnih korenov)

$$\sum_{k=1}^n (2(k-1) + (n-k)(2k-1)) = \frac{1}{3}n^3 + \mathcal{O}(n^2).$$

- V primerjavi s klasičnim LU razcepom približno pol manj, kar zaradi simetričnosti ni presenetljivo.
- Razcep Choleskega je tudi najcenejši način, kako preverimo pozitivno definitnost matrike A (ne sme priti do korenjenja negativnih števil).

Primer

Izračunajmo razcep Choleskega za matriko

$$A = \begin{bmatrix} 4 & -2 & 4 & -2 & 4 \\ -2 & 10 & 1 & -5 & -5 \\ 4 & 1 & 9 & -2 & 1 \\ -2 & -5 & -2 & 22 & 7 \\ 4 & -5 & 1 & 7 & 14 \end{bmatrix}.$$

Neposredno po algoritmu dobimo

$$V = \begin{bmatrix} 2 & & & & \\ -1 & 3 & & & \\ 2 & 1 & 2 & & \\ -1 & -2 & 1 & 4 & \\ 2 & -1 & -1 & 2 & 2 \end{bmatrix}.$$

- Če je matrika $A \in \mathbb{R}^{n \times n}$ simetrična pozitivno definitna, potem sistem linearnih enačb $A\mathbf{x} = \mathbf{b}$ rešimo takole:
 - ▶ Razcepimo A z razcepom Choleskega: $A = V V^\top$.
 - ▶ Rešimo sistem $V\mathbf{y} = \mathbf{b}$ (direktno vstavljanje).
 - ▶ Rešimo sistem $V^\top \mathbf{x} = \mathbf{y}$ (obratno vstavljanje).
- Pri razcepu Choleskega lahko pivotiramo na poseben način, da ohranjamo simetrijo (**diagonalno pivotiranje**). Na vsakem koraku poiščemo največji element po absolutni vrednosti na preostanku diagonale in izvedemo zamenjavo ustreznih vrstic in stolpcev.
- Če simetrična matrika ni pozitivno definitna, razcep Choleskega ne obstaja. Obstaja pa razcep $PAP^\top = LDL^\top$, kjer je L spodnja trikotna matrika z enkami na diagonali, D pa bločno diagonalna matrika z bloki 1×1 ali 2×2 .

Tridiagonalni sistemi

- Rešujemo sistem $A\mathbf{x} = \mathbf{b}$, kjer je A tridiagonalna matrika

$$A = \begin{bmatrix} a_{11} & a_{12} & & & & \\ a_{21} & a_{22} & a_{23} & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} & \\ & & & a_{n,n-1} & a_{n,n} & \end{bmatrix}.$$

- Pri LU razcepu brez pivotiranja dobimo

$$L = \begin{bmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ & \ddots & \ddots & & \\ & & \ell_{n,n-1} & 1 & \end{bmatrix}, U = \begin{bmatrix} u_{11} & a_{12} & & & \\ & \ddots & \ddots & & \\ & & & u_{n-1,n-1} & a_{n-1,n} \\ & & & & u_{n,n} \end{bmatrix}$$

- Pri tem za razcep in nadaljnje reševanje potrebujemo $\mathcal{O}(n)$ operacij.

- Če moramo pivotirati, se v matriki U pojavi še ena obdiagonala

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & & & \\ & \ddots & \ddots & \ddots & & \\ & & & u_{n-2,n-2} & u_{n-2,n-1} & u_{n-2,n} \\ & & & & u_{n-1,n-1} & u_{n-1,n} \\ & & & & & u_{n,n} \end{bmatrix}.$$

Izrek

Če je matrika $A \in \mathbb{R}^{n \times n}$ diagonalno dominantna po stolpcih (in vsaj v enem stolpcu strogo diagonalno dominantna), pivotiranje ni potrebno.

(Glejte dokaz z vaj!)

- Torej lahko tridiagonalni sistem, pri katerem je matrika diagonalno dominantna po stolpcih, rešujemo v **kompaktni obliki**.
- Sistem predstavimo z matriko $M \in \mathbb{R}^{n \times n}$ in vektorjem desnih strani ter priredimo algoritem LU razcepa.
- Dobljeni časovna in prostorska zahtevnost sta $\mathcal{O}(n)$.