

## Semestrální práce - numerické metody

Název: LU rozklad matice  $\mathbf{A}$  s částečným výběrem hlavních prvků.

## 1. Zadání

Zadání: LU rozklad matice **A** s částečným výběrem hlavních prvků.

Vstup: matice **A**

Výstup: dolní trojúhelníková matice **L**, horní trojúhelníková matice **U**, permutační matice **P**

## 2. Stručný popis použité metody řešení

LU rozklad matice  $\mathbf{A}$  s částečným výběrem hlavních prvků vychází z Gaussovy eliminační metody (GEM), která se používá pro řešení soustav lineárních rovnic ( $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ ). Při GEM se postupně prochází všechny sloupce zleva doprava a provádíme takové úpravy, abychom pod diagonálou

dostali nuly. Tyto úpravy spočívají v tom, že dle vztahu  $m_{ik} = \frac{a_{ik}}{a_{kk}}$  prvků matice  $\mathbf{A}$  vypočítáme tzv. multiplikátor, který odečteme od prvku matice, který chceme vynulovat.

LU rozklad znamená, že tyto multiplikátory zaznamenáme do dolní trojúhelníkové matice  $\mathbf{L}$  a do horní trojúhelníkové matice  $\mathbf{U}$  zaznamenáme diagonální a naddiagonální prvky matice  $\mathbf{A}$ . Dá se dokázat, že platí  $\mathbf{A} = \mathbf{LU}$ .

Částečný výběr hlavního prvku je modifikace GEM zajišťující, aby absolutní hodnota multiplikátorů byla menší nebo rovna jedné. Toho se dosáhne tím, že řádky v dosud neeliminované části matice se přesouvají tak, aby diagonální prvek v aktuálním sloupci měl větší absolutní hodnotou než zbývající poddiagonální prvky aktuálního sloupce. Tím se zajistí nejmenší nepřesnosti vlivem zakrouhlování. Tyto přesuny řádků se zaznamenají do permutační matice  $\mathbf{P}$ .

### 3. Opis programu

```
%semestralni prace - LU rozklad matice A s castecnym vyberem  
hlavnich prvku.  
  
%autor  
  
%VSTUP A - matice 3x3  
  
%  
  
%VYSTUP dolni 3-uhelnikova matice L, horni 3-uhelnikova matice U,  
permutacni matice P; matice 3x3  
  
function [L, U, P] = LURozklad(A)  
  
    P = eye(size(A)); % jednotkova matice  
  
    L = P;  
  
    A_predchozi = A;  
    P_predchozi = P;  
  
    for k = 1:size(A) - 1  
        % urceni indexu "r" pivotniho radku  
        r = IndexPivotnihoRadku(A, k, k);  
  
        % prohozeni radku k a r matice A_predchozi  
        A_predchozi = A;  
        A = VymenaRadkuMatice(A, k, r);  
  
        % prohozeni radku k a r matice P_predchozi:  
        P_predchozi = VymenaRadkuMatice(P_predchozi, k, r);  
        P = P_predchozi;  
  
        for i = k + 1:size(A)  
            m_ik = A(i, k)/A(k, k);
```

```
        for j = k + 1:size(A)
            A(i, j) = A(i, j) - m_ik * A(k, j);
        end

        A(i, k) = m_ik;
    end
end

printf('matice A: ', A);

%sestaveni dolni 3-uhelnikove matice L:
for i = 2:size(A)
    for j = 1:size(A)
        if i > j
            L(i, j) = A(i, j); % ve skriptech je
A_predchozi(i, j); chyba?
        end
    end
end

printf('matice L: ', L);

%sestaveni horni 3-uhelnikove matice U:
for i = 1:size(A)
    for j = 1:size(A)
        if i <= j
            U(i, j) = A(i, j); % ve skriptech je A_predchozi(i,
j); chyba?
        end
    end
end

printf('matice U: ', U);

P = P_predchozi;
```

```
printf('matice P:', P);
```

```
end
```

```
function [r] = IndexPivotnihoRadku(A, OdRadku, IndexSloupce)
```

```
    MaxHodnota = 0;
```

```
    for i = OdRadku:size(A)
```

```
        if abs(A(i,IndexSloupce)) > MaxHodnota
```

```
            MaxHodnota = abs(A(i,IndexSloupce));
```

```
            r = i;
```

```
        end
```

```
end
```

```
function [A] = VymenaRadkuMatice(A, radek1, radek2)
```

```
    PomocnyRadek = A(radek1,:);
```

```
    A(radek1, :) = A(radek2,:);
```

```
    A(radek2, :) = PomocnyRadek;
```

```
end
```

```
function printf(string, value)
```

```
    disp(string);
```

```
    disp(value);
```

```
end
```

## 4. Demonstrace programu

LU-rozklad bude demonstrován na dvou maticích:

$$\text{a) } \mathbf{A} = \begin{pmatrix} -0,4 & -0,95 & -0,4 & -7,34 \\ 0,5 & -0,3 & 2,15 & -2,45 \\ -2 & 4 & 1 & -3 \\ -1 & 5,5 & 2,5 & 3,5 \end{pmatrix}$$

Výstup programu Matlab:

```
krok c.1:

matice A:
-2.0000    4.0000    1.0000   -3.0000
-0.2500    0.7000    2.4000   -3.2000
 0.2000   -1.7500   -0.6000   -6.7400
 0.5000    3.5000    2.0000    5.0000

krok c.2:

matice A:
-2.0000    4.0000    1.0000   -3.0000
 0.5000    3.5000    2.0000    5.0000
 0.2000   -0.5000    0.4000   -4.2400
-0.2500    0.2000    2.0000   -4.2000

krok c.3:

matice A:
-2.0000    4.0000    1.0000   -3.0000
 0.5000    3.5000    2.0000    5.0000
-0.2500    0.2000    2.0000   -4.2000
 0.2000   -0.5000    0.2000   -3.4000

matice L:
 1.0000         0         0         0
 0.5000    1.0000         0         0
-0.2500    0.2000    1.0000         0
 0.2000   -0.5000    0.2000    1.0000
```

obr. č.1: výstup programu Matlab

```
matice U:  
  -2.0000    4.0000    1.0000   -3.0000  
         0    3.5000    2.0000    5.0000  
         0         0    2.0000   -4.2000  
         0         0         0   -3.4000  
  
matice P:  
    0     0     1     0  
    0     0     0     1  
    0     1     0     0  
    1     0     0     0
```

*obr. č.1: výstup programu Matlab*



b)  $A = \begin{pmatrix} -3 & 6 & 1 \\ 12 & 3 & -1 \\ -2 & 4 & -1 \end{pmatrix}$

Výstup programu Matlab:

```
krok c.1:

matice A:
    12.0000    3.0000   -1.0000
   -0.2500    6.7500    0.7500
   -0.1667    4.5000   -1.1667

ans =

krok c.2:

matice A:
    12.0000    3.0000   -1.0000
   -0.2500    6.7500    0.7500
   -0.1667    0.6667   -1.6667

matice L:
    1.0000         0         0
   -0.2500    1.0000         0
   -0.1667    0.6667    1.0000

matice U:
    12.0000    3.0000   -1.0000
         0    6.7500    0.7500
         0         0   -1.6667

matice P:
     0     1     0
     1     0     0
     0     0     1
```

obr. č.2: výstup programu Matlab

## 5. Závěr

Použitá metoda LU-rozkladu s částečným výběrem hlavních prvků vyžaduje poměrně málo výpočetních kroků: 3 kroky pro matici  $4 \times 4$  a 2 kroky pro matici  $3 \times 3$ . Výběr hlavního prvku zaručuje malé zaokrouhlovací chyby. Tato metoda je velmi výhodná pro řešení soustav lineárních rovnic.