Masarykova univerzita

# Maticové rozklady

## DOKUMENTACE

**Vypracoval: Martin Nováček, 393995**

Vyučující: Mgr. Jiří Zelinka, Dr.

Datum odevzdání: 1/2012

Masarykova univerzita

BRNO

# Maticové rozklady

## DOKUMENTACE

www.martinnovacek.cz

## Obsah

# 1. Anotace

Funkce *Maticové rozklady* slouží k provádění maticových rozkladů, konkrétně rozkladů typu LU, Choleského a Croutova. Tyto výpočty jsou blíže specifikovány v 6. kapitole této dokumentace.

Tento program vznikl jako zápočtový projekt z předmětu Numerické výpočty I na Přírodovědecké fakultě Masarykovy univerzity v Brně v podzimním semestru roku 2011.

Byl napsán jakožto M-file funkce pro výpočetní prostředí MATLAB. Součástí projektu je také tato dokumentace.

Funkci je možné stáhnout na webu:

*www.martinnovacek.cz/edu/muni/rozklady*

*Tato dokumentace nebyla součástí zadání projektu. Dle mého názoru je však dobrým zvykem ke každému programu, skriptu a podobným projektům dokumentaci psát, aby bylo později jasné, jaký význam mají jednotlivé části a jaký význam má program jako celek.*

# 2. Příprava programu

## 2.1 Instalace

Pro úspěšné použití je třeba mít nainstalovaný software MATLAB. Soubor M-File *rozklady.m* se zdrojovým kódem se nakopíruje do příslušné složky. V MATLABu je označená jako *„Current Folder“*.

## 2.2 Spuštění

Chcete-li program spustit, napište do příkazového řádku systému MATLAB příkaz `rozklady(A)`, kde A je předem daná čtvercová a neprázdná matice.

## 2.3 Odinstalace

Odinstalaci této funkce – *rozklady.m* – provedete jejím smazáním ze složky, do které jste program zkopírovali při instalaci.

## 3. Práce s programem

Spustíte-li program příkazem `rozklady(A)`, objeví se nabídka maticových rozkladů:

```
Vyberte typ rozkladu (1/2/3):
  1. LU rozklad
  2. Choleského rozklad
  3. Croutův rozklad
```

Konkrétní rozklad vyberete vepsáním příslušného čísla a stisknutím klávesy Enter. Například pro Choleského rozklad to bude:

```
>> 2
```

Po provedení výpočtu se zobrazí příslušné matice rozkladu (L a U, T a T', nebo L a U) a také jejich násobek pro kontrolu správnosti výpočtu.

Pro vyvolání nápovědy v systému MATLAB zadejte příkaz:

```
>> help rozklady
```

# 4. Maticové rozklady obecně

## 4.1 LU rozklad

LU rozklad matice je rozklad, který nám pro jednu vstupní matici *A* vrátí matice *L* a *U* takové, že:

$$A = L \cdot U$$

Matice *L* je dolní trojúhelníková matice s jedničkami na diagonále a matice *U* je horní trojúhelníková matice. Obecně vypadají matice rozkladu takto:

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{pmatrix}}_{L} \cdot \underbrace{\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & \vdots \\ \vdots & \ddots & \ddots & u_{n-1,n} \\ 0 & \cdots & 0 & u_{nn} \end{pmatrix}}_{U}$$

LU rozklad se standardně používá pro numerický výpočet lineárních rovnic $A \cdot x = b$. Matici $A$ totiž můžeme nahradit maticemi rozkladu, tedy $L \cdot U \cdot x = b$, a také můžeme označit $U \cdot x = z$ . Tímto způsobem dostaneme následující:

$$L \cdot U \cdot x = b \Leftrightarrow L \cdot z = b,\ U \cdot x = z$$

Nejprve vyřešíme rovnice $L \cdot z = b$ dopřednou substitucí a pak rovnice $U \cdot x = z$ substitucí zpětnou.

## 4.2 Choleského rozklad

**Věta:** *Nechť matice $A \in \mathcal{M}_n$ je symetrická a všechny její hlavní minory jsou různé od nuly, tedy $a_{11} \neq 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0, \det A \neq 0$, pak existuje horní trojúhelníková matice $T \in \mathcal{M}_n$ taková, že $A = T \cdot T^T$.*

Pokud je rozkládaná matice $A$ pozitivně definitní, dostaneme zpět reálnou matici $T$. Jestli však pozitivně definitní není, dostneme matici s ryze imaginárními prvky na některých pozicích.

Tento rozklad se, stejně jako rozklad LU, používá k řešení systému lineárních rovnic. Rozkladem získáme dvě trojúhelníkové matice, resp. $T$ a $T^T$, pomocí kterých řešíme dva systémy.

## 4.3 Croutův rozklad

**Věta:** *Mějme třídiagonální matici A, která splňuje tyto podmínky:*

1. $a_{i,i-1}a_{i,i+1} \neq 0,\qquad i = 2, \dots, n-1$
2. $|a_{11}| > |a_{12}|$
3. $|a_{ii}| \geq |a_{i,i-1}| + |a_{i,i+1}|,\qquad i = 2, \dots, n-1$
4. $|a_{nn}| > |a_{n,n-1}|$

Pokud tyto podmínky matice splňuje, tak je regulární a vypočtené hodnoty v maticích L a U jsou různé od nuly.

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

Hledáme její rozklad na matice L a U, kde

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & & 0 \\ 0 & l_{32} & l_{33} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & l_{n,n-1} & l_{nn} \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & u_{12} & \cdots & 0 & 0 \\ 0 & 1 & u_{23} & & 0 \\ \vdots & \ddots & 1 & \ddots & \vdots \\ 0 & & \ddots & 1 & u_{n-1,n} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

# 5. Autoři maticových rozkladů – krátký exkurs do historie

## 5.1 Alan Turing, autor LU rozkladu

Alan Turing se narodil 12. června roku 1912 v londýnském distriktu Maida Vale. V letech 1931 až 1934 studoval matematiku na Cambridge King's College.

Největším dílem Alana Turinga je zavedení tzv. Turingova stroje. Prvně se o něm zmínil roku 1936. Jedná se o teoretický matematický model počítače, který má nekonečně velkou operační paměť.

V letech 1937 až 1938 studoval na americké Princeton University. Taktéž je, společně s americkým matematikem Alonzem Churchem, autorem Church-Turingovy teze, která neformálně říká, že každý proveditelný výpočet je možné provést algoritmem běžícím na počítači s dostatečnou kapacitou paměti a času.

V průběhu druhé světové války pobýval v anglickém Buckinghamshire, kde spolu s dalšími vědci pomáhal luštit německé zprávy šifrované Enigmou.

Roku 1948 představil maticový LU rozklad, který dodnes používáme k řešení soustav lineárních rovnic.

Mezi další zájmy Alana Turinga patřila například biologie. Studoval a snažil se vysvětlit zejména tzv. morfogenezi.

O jeho osobním životě víme, že byl homosexuál, což bylo v tehdejší Anglii trestné. Turing tak byl odsouzen k roční hormonální „léčbě“ estrogenem. Zemřel 7. června 1954 ve Wilmslow. Podle oficiální zprávy se jednalo o sebevražednou otravu kyanidem draselným.

V roce 2009 byla v The Daily Telegraph zveřejněna omluva britské vlády za tehdejší Turingovo odsouzení.

Na počest Alana Turinga se za významné přínosy informatice každoročně od roku 1966 uděluje Turingova cena.

## 5.2 André-Louis Cholesky

André-Louis Cholesky se narodil 15. října roku 1875 v Montguyon na jihozápadním pobřeží Francie v departmentu Charente-Maritime. Cholesky byl francouzský oficír, topograf, geodetik a také matematik.

V letech 1892 až 1893 v Bordeaux maturoval. Od roku 1985 studoval na polytechnické škole a mezi lety 1897 až 1899 chodil na vojenskou L'École d'application de l'artillerie et du génie v Metách. Po studiu se věnoval právě topografii a geografii ve službách armády.

Proslavil se zejména díky jeho způsobu řešení systému lineárních rovnic – Choleského rozkladu (Factorisation de Cholesky). Tento rozklad však nevznikl původně jakožto výsledek matematického výzkumu. Vznikl jako výsledek studie o kompenzaci geodetických sítí.

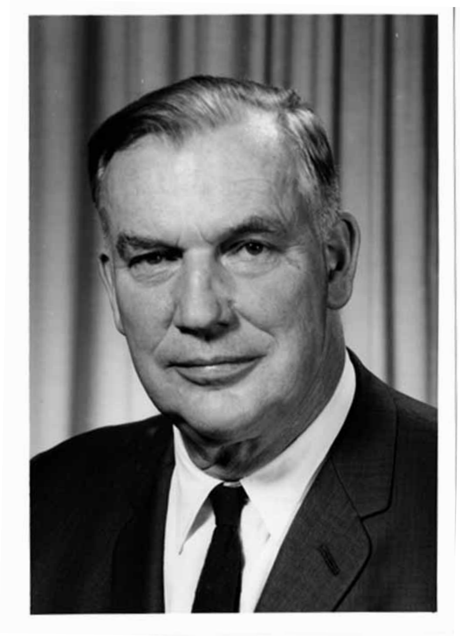
Roku 1909 se stal druhým nejvyšším velitelem, později kapitánem 13. dělostřeleckého pluku.

André-Louis Cholesky zemřel 31. srpna 1918 v Bagneaux v Picardii na následky zranění z bitvy. Jeho ostatky byly v roce 1921 převezeny na hřbitov v Cuts.

André-Louis Cholesky v době studia l'École Polytechnique

## 5.3 Prescott Durand Crout

Prescott Durand Crout se narodil 28. července roku 1907. V letech 1934 až 1973 působil na Fakultě matematiky MIT (Massachusetts Institute of Technology) ve Spojených státech amerických a je autorem tzv. Croutova maticového rozkladu.



# 6. Teoretická programátorská část

## 6.1 Základní topologie zdrojového kódu

Zdrojový kód začíná jako standardní M-File funkce v MATLABu, tedy uvedením klíčového slova `function`, názvu funkce a proměnnou, která je přebírána na vstupu. Návratové proměnné uvedeny nejsou, jelikož každý rozklad navrací jinak pojmenovanou matici. Proto je výpis výsledků řešen přímým zobrazením do příkazového řádku funkcí `disp`. Nevýhodou je, že se s maticemi nedá později pracovat jako se standardními proměnnými.

Následuje krákÿ Help k funkci, který popisuje, jaké typy rozkladu lze provést a jaké navrací matice. Jejich popis je však velmi krátký s odvoláním na tuto Dokumentaci.

Na začátku programové části jsou uvedeny podmínky, které kontrolují, zda-li matice na vstupu splňuje základní kritéria, tedy jestli se náhodou nejedná o prázdnou matici a jestli je čtvercová (stejný počet řádků a slopců). V případě, že jedna z podmínek není splněna, MATLAB pomocí funkce `error` vypíše příslušnou hlášku – ohlásí konkrétní typ chyby – a poté provádění funkce ihned ukončí.

Dále se program uživatele ptá, který maticový rozklad si přeje provést. Jako odpověď očekává číslo z intervalu ⟨1,3⟩. Při zadání jakékoliv jiné hodnoty MATLAB taktéž ohlásí chybu funkcí `error`.

Poslední, nejdůležitější, částí kódu je vícecestná rozhodovací konstrukce `struct`. Tento struct má tři větve – pro každý maticový rozklad jednu.

Většina příkazů je taktéž opatřena komentářem pro lepší orientaci, co který příkaz nebo výpočet dělá.

### 6.2 Rozbor kódu – LU rozklad

Nejprve pomocí funkce `zeros`, která slouží pro vytvoření matice o dané velikosti s nulovými prvky, program předalokuje matice $L$ a $U$. Tento krok není nezbytně nutný, ale plnit matice přímo za chodu programu a měnit tak dynamicky jejich velikost je časově i režijně poměrně náročné. Proto jsou obě předalokovány na odpovídající velikost a naplněny nulami.

Dále program „rozbíhá“ **hlavní for cyklus** od 1 do $n$, kde $n$ je velikost matice a při každém průchodu tímto cyklem umístí na odpovídající diagonální prvek matice $L$ jedničku. Tento cyklus obsahuje dva vnořené cykly a první vnořený obsahuje ještě jeden vnořený.

**První vnořený cyklus** počítá prvky pod diagonálou matice $L$. Výpočet probíhá pomocí vztahu: $L(k,k+1) = \frac{A(k,k+1)}{A(k+1,k+1)}$.

V tomto cyklu se také kontroluje dělení nulou[1]. Tento rozklad neumí zaměňovat řádky matice, proto nelze provést rozklady matic s nulou na diagonále. V takovém případě dojde k ohlášení patřičné chyby.

**Cyklus vnořený do prvního vnořeného cyklu** přepočítává prvky matice $A$ podle již spočítané matice $L$ tak, aby horní trojúhelník matice $A$ odpovídal hodnotami jednotlivých prvků matici $U$. Podle vztahu: $A(k+1,k+2) = A(k+1,k+2) - (L(k+1,k) \cdot A(k,k+2))$.

**Druhý vnořený cyklus** dopočítá horní trojúhelníkové prvky matice $U$ tak, že jednoduše vybere prvky z matice $A$ na horní diagonále a umístí je do horního trojúhelníku matice $U$.

**Pro názornost ještě přehlednější schéma cyklů:**

```
Hlavní cyklus {
    1. vnořený {
        → matice L
        vnitřní vnořený {
            → úprava matice A
        }
    }
    2. vnořený {
        → dopočítání matice U
    }
}
```

Dalšími kroky se jen vypíší matice $L$, $U$ a jejich násobek $LU$ na obrazovku.

## 6.3 Rozbor kódu – Choleského rozklad

Jelikož je Choleského rozklad možné provést pouze ze symetrické matice, algoritmus nejprve kontroluje, je-li matice na vstupu symetrická, nebo není. Tato kontrola probíhá tak, že se nejprve sečtou všechny prvky (logické hodnoty), které jsou v matici $A$ i v matici $A^T$ shodné. Toto porovnání je logická operace a tudíž vrací jedničky na místech, kde jsou prvky shodné. Dále se vynásobí jeden rozměr matice s druhým. Pokud je matice symetrická, oba vysledky budou shodné.

V dalším kroku se opět předalokuje, ze stejného důvodu jako v případě LU rozkladu, výsledná matice $T$ pomocí příkazu `zeros`.

Samotný výpočet prvků probíhá podle standardních vztahů uvedených v knize Numerické metody[1]. Algoritmizovaný postup využívá dvou for cyklů:

**Hlavní cyklus {**

→ diagonální prvky $T$

**1. vnořený {**

→ dolní trojúhelník v $T$

**}**

**}**

Vztahy:

$$T(i,i) = \sqrt{a_{ii} - \sum_{l=1}^{i-1} t_{li}^2}$$ pro všechny diagonální prvky $T$

$$T(j,i) = \left(A(j,i) - \left(T(i,:)\cdot T^T(j,:)\right)\right)\cdot \frac{1}{T(i,i)}$$ pro všechny prvky pod diagonálou $T$

**Pozn.:** Standardně se Choleského rozklad provádí pouze u pozitivně definitních symetrických matic. Symetrická matice musí být na vstupu v každém případě. Pozitivně definitní však býti nemusí – budou vycházet záporné hodnoty pod odmocninou. Výsledkem bude matice s ryze imaginární prvky.

## 6.4 Rozbor kódu – Croutův rozklad

V případě výpočtu Croutova rozkladu na vstupu očekáváme třídiagonální matici. To musíme ověřit podmínkou. Tato podmínka je:

```
if (sum(sum(triu(A,+2)))==0) && (sum(sum(tril(A,-2)))==0) … zdrojový kód výpočtu … end
```

Co tyto řádky podmínkového konstruktu `if` říkají? Podmínka se „ptá“ na dvě subpodmínky, které musí být splněny obě dvě zároveň:

1. Jsou všechny prvky nad horní diagonálou nulové?
2. Jsou všechny prvky pod dolní diagonálou nulové?

Jsou-li všechny tyto podmínky splněny, matice je třídiagonální a výpočet může pokračovat. V opačném případě MATLAB ohlásí chybu pomocí funkce `error`.

V dalším kroku se zjistí rozměr vstupní čtvercové matice a jeho hodnota se uloží do proměnné $n$. Podle této hodnoty se předalokují matice $L$ a $U$ na rozměr odpovídající vstupní matici, přičemž matice $L$ je tvořená nulami a matice $U$ je jedničková.

Dále následuje for cyklus, který spočítá prvky v dolním trojúhelníku (bez diagonály) matice $L$ a to pouhým překopírováním prvků z matice vstupní. Pak se na diagonální prvek $L_{1,1}$ uloží hodnota prvku $A_{1,1}$.

V dalším kroku následuje další for cyklus, tentokráte od 1 do $n-1$. Jsou počítány prvky horního trojúhelníku v matici $U$ a diagonální prvky v matici $L$ pomocí vztahů:

$$U_{j,j+1} = \frac{A_{j,j+1}}{L_{j,j}}$$

$$L_{j+1,j+1} = A_{j+1,j+1} \cdot \left(L_{j+1,j} \cdot U_{j,j+1}\right).$$

V tomto cyklu je také podmínka, která kontroluje dělení nulou[1] ve výpočtu prvku v matici $U$.

# 7. Poznámky k dokumentaci

### *ad 1. Kontrola dělení nulou*

Dělení nulou se provádí tak, že se spočítá absolutní hodnota dělitele. Pokud je tato absolutní hodnota menší než hodnota $\varepsilon$, pak dojde k ohlášení chyby – dělení nulou. Hodnota $\varepsilon = 1{,}4901 \cdot 10^{-008}$, je to tedy dostatečně malé číslo, aby eliminovalo případné chyby v počítačové reprezentaci čísel.

Pouze v případě Choleského rozkladu, kde se mohou objevovat komplexní čísla, se dělení nulou kontroluje touto podmínkou: `abs(real(T(k,k)))<eps && abs(imag(T(k,k)))<eps`. Jiné případy dělení komplexním číslem jsou optimalizované přímo MATLABem.

## 8. Přepis reálného kódu funkce *rozklady.m* z M-File

```
function rozklady(A)

% Tato funkce umožňuje provést maticový rozklad dle výběrů:
%   1. LU rozklad
%         - vrací matice L a U
%   2. Choleského rozklad
%         - vrací matice T a T'
%   3. Croutův rozklad
%         - vrací matice L a U
%
% Funkce ve všech třech případech přebírá jako argument jednu matici.
%
% Tento program vznikl jako zápočtový projekt do předmětu Numerické
% výpočty I v podzimním semestru roku 2011.
%
% Vypracoval: Martin Nováček, 393995
% Vyučující: Mgr. Jiří Zelinka, Dr.
%
% Pro více informací, prosím, čtěte dokumentaci na:
%   www.martinnovacek.cz/edu/muni/rozklady
%


[m,n]=size(A);

if m*n==0
    error('Matice je prázdná! Zadejte neprázdnou matici.');
end

if(m~=n)
    error('Matice není čtvercová! Zadejte čtvercovou matici.');
end

disp('Vyberte typ rozkladu (1/2/3): ')      % main title of the selection
disp('  1. LU rozklad')                     % 1st choice for LU decomposition
disp('  2. Choleského rozklad')             % 2nd choice for Cholesky decomposition
disp('  3. Croutův rozklad')                % 3rd choice for Crout decomposition

choice = input('>> ','s');         % print ">>" and wait for user entered input

switch choice                      % switch which enable compute chosen decompositon
    case '1'                       % LU decomposition
        disp('LU rozklad: ');      % visual checkpoint for input to case 1

        L=zeros(n);                % preallocating L matrix
        U=zeros(n);                % preallocating U matrix

        for k=1:n                  % for loop from 1 to n
            L(k,k)=1;              % gives ones to diagonal of L matrix
            for i=k+1:n            % for loop from k+1 to n
                if abs(A(k,k))<eps
                    error(Nedefinované dělení nulou. LU rozklad zadané matice
                    nelze provést bez výměny řádku.');
                end
                L(i,k)=A(i,k)/A(k,k); % formula for under-diagonal elements of L
                                      matrix
                for j=k+1:n           % for loop from k+1 to n
```

```
                A(i,j)=A(i,j)-(L(i,k)*A(k,j)); % recompute A matrix according
                                               to L matrix
            end
        end

        for j=k:n               % for loop from k to n
            U(k,j)=A(k,j);      % formula for upper-diagonal elements of U matrix
        end
    end

    disp('L =');                % print matrix L
    disp(L);
    disp('U =');                % print matrix U
    disp(U);
    disp('L*U =');              % print matrix L*U
    disp(L*U);


case '2'                        % Cholesky decomposition
    disp('Choleského rozklad: ');   % visual checkpoint for input to case 2

    x=sum(sum(A==A'));          % summary of all elements
    y=m*n;                      % multiply m*n

    if x==y                     % if symmetric matrix

       T = zeros(n);            % preallocating T(n,n) matrix

       for k=1:n                % for loop from 1 to n
           T(k,k)=sqrt(A(k,k)-(T(k,:)*T(k,:).')); % formula for diagonál
                                                     elements

           for l=(k+1):n        % for loop from i+1 to n (elements under
                                diagonal)
               if (abs(real(T(k,k)))<eps) && (abs(imag(T(k,k)))<eps)
                   error('Nedefinované dělení nulou. Choleského rozklad zadané
                  matice nelze provést.');
               end
               T(l,k)=(A(l,k)-T(k,:)*T(l,:).')*(1/T(k,k));    % formula for
                                                               under-diagonal
                                                               elements

           end
       end

       disp('T =');             % print matrix T
       disp(T);
       disp('T'' =');           % print matrix T'
       disp(T.');
       disp('T*T'' =');         % print matrix T*T'
       disp(T*T.');

    else                        % error message for non-symetric matrix
        error('Matice není symetrická! Zadejte symetrickou matici.')
    end

case '3'                        % Crout decomposition
    disp('Croutův rozklad: ');  % visual checkpoint for input to case 3
```

```
        if (sum(sum(triu(A,+2)))==0) && (sum(sum(tril(A,-2)))==0)
                                        % check if A matrix is tridiagnal
            [~,n]=size(A);              % dimension of matrix A

            L=zeros(n);                 % preallocating L matrix
            U=eye(n);                   % preallocating U eye-matrix

            for i=2:n                   % for loop from 2 to n - access to lower-
                                        diagonal elements of L matrix
                L(i,i-1)=A(i,i-1);      % computes lower-diagonal elements in L
                                        matrix
            end

            L(1,1)=A(1,1);              % A(1,1) value is copied to element L(1,1)

            for j=1:n-1                 % for loop from 1 to n-1 - access to:
                if abs(L(j,j))<eps
                                        %
                    error('Nedefinované dělení nulou. Croutův rozklad zadané matice
                           nelze provést.');
                end                      %
                U(j,j+1)=A(j,j+1)/L(j,j);%   >> upper-diagonal elements in U matrix
                L(j+1,j+1)=A(j+1,j+1)-(L(j+1,j)*U(j,j+1));%   >> diagonal elements
                                                               in L matrix
            end

            disp('L =');                % print matrix L
            disp(L);
            disp('U =');                % print matrix U
            disp(U);
            disp('L*U =');              % print matrix L*U
            disp(L*U);


        else error('Zadaná matice není třídiagonální!');

        end

    otherwise
        error('Chybný výběr rozkladu!');

end

%-------------------------------------------------------------------------%
% END - FUNCTION rozklady(A)                                              %
%-------------------------------------------------------------------------%
```

## 9. Literatura, zdroje k matematické a informatické části


[1] Horová, Ivana, Zelinka, Jiří: *Numerické metody*. Masarykova univerzita, Brno 2008.

[2] Zelinka, Jiří, Koláček, Jan: *Jak pracovat s MATLABem*. Masarykova univerzita.

[3] Slovák, Jan, Panák, Martin: *Drsná matematika.* Dosud nevydáno.

[4] Olšák, Petr: *Lineární algebra*: Vydavatelství ČVUT, Praha 2010.


## 10. Webové stránky, zdroje k historické části


http://en.wikipedia.org/wiki/LU_decomposition
http://cs.wikipedia.org/wiki/Alan_Turing
http://cs.wikipedia.org/wiki/Turingova_cena
http://cs.wikipedia.org/wiki/Gordon_Brown
http://cs.wikipedia.org/wiki/Enigma
http://en.wikipedia.org/wiki/Princeton_University
http://cs.wikipedia.org/wiki/Turingův_stroj
http://en.wikipedia.org/wiki/Church-Turing_thesis
http://en.wikipedia.org/wiki/Bletchley_Park
http://cs.wikipedia.org/wiki/Alonzo_Church
http://fr.wikipedia.org/wiki/André-Louis_Cholesky
http://en.wikipedia.org/wiki/André-Louis_Cholesky
http://fr.wikipedia.org/wiki/Montguyon
http://familytreemaker.genealogy.com/users/g/r/a/Kathleen-Grace/WEBSITE-0001/UHP-0269.html
http://webmuseum.mit.edu/detail.php?t=people&type=browse&f=preferred_name&s=Crout%2C+Prescott+Durand&record=0


## 11. Fotografie


http://www.ieee.org/portal/cms_docs_sscs/sscs/08Spring/KFig6_turing.jpg
http://www.reunion.iufm.fr/dep/mathematiques/calculsavant/Textes/Resources/Portrait_Cholesky.gif
http://webmuseum.mit.edu/grabimg.php?kv=65129