MATLAB II

Ing. Bohumil Kovář

12. března 2002

1 Funkce

V současné době má MATLAB několik set funkcí. Funkce jsou buď vnitřní, nebo se jedná o tzv. M-funkce, které jsou uloženy na disku. Uživatel může vytvářet vlastní M-funkce (pomocí příkazů MATLABu, nebo pomocí mex-funkcí naprogramovaných jazykem C) a tím rozšiřovat funkčnost MATLABu.

Funkce mohou mít více vstupních a výstupních parametrů. Příklad zápisu funkce s jedním výstupním parametrem je `y=sin(x)`. Funkce však může vracet i více parametrů (např. `[m,n]=size(X)`).

2 M-soubory (M-files)

Příkazy MATLABu jsou prováděny buď ihned (v command window) nebo mohou být uloženy do m-souboru, do textového souboru s příponou .m, a prováděny sekvenčně. Příkazy m-souboru se mohou odkazovat na jiné m-soubory, případně rekurzivně volat sami sebe. M-soubory jsou typu *script* nebo *m-funkce*.

2.1 Script soubory

Scriptové soubory jsou sekvencemi příkazů ve kterých jsou všechny proměnné globální.

Příklad:

```
x = [-pi:0.1:pi]';
y = sin(x);
plot(y)
```

2.2 M-funkce

M-funkce rozšiřují možnosti MATLABu, protože umožňují definovat nové funkce. Proměnné jsou uvnitř funkce lokální a s prostředím MATLABu komunikují pomocí vstupních a výstupních parametrů. Nová funkce se definuje příkazem `function` a seznamem vstupních a výstupních parametrů (bez tohoto příkazu by se jednalo o script soubor).

Příklad:

```
function y=prumer(x)
% vstupni parametr vektor x
% vystupni parametr aritmeticky prumer
% pouziti y=prumer(x)
y=0;
soucet=0;
for i=1:length(x)
    soucet=soucet+x(i);
end
y=soucet/length(x);
```

2.3 Mex-funkce

Z MATLABu lze volat i externí funkce napsané v jazycích C nebo Fortran. Funkce naprogramovaná v jazyce C se skládá ze dvou částí. První, tzv. gateway interface, zajišťuje komunikaci s MATLABem a je obdobou *main()* funkce v jazyce C. Druhá část je vlastní kód. Výhodou mex funkcí je jejich mnohonásobně vyšší rychlost oproti m-funkcím.

3 Programování M-souborů

Při programování m-souborů je často požadováno opakované provádění skupiny příkazů. V případě, že je počet opakování dán, je vhodné použít příkaz `for`.

```
for <parametr>=<vyraz>
    <prikazy>
end
```

Příkazy cyklu se mohou vkladat do sebe, každý příkaz `for` je ukončen příkazem `end`.

Příklad:

```
for i=1:m
    for j=1:n
        x(i,j)=0;
        y(i,j)=0;
    end
end
```

V případě, že počet opakování není předem dán, je možné použít příkaz `while`. Skupina příkazů je prováděna tolikrát, dokud je splněna logická podmínka.

```
while <podminka>
    <prikazy>
end
```

Příklad:

```
n=1;
while prod(1:n) < 1000
    n=n+1;
end;
```

Příkaz `if` umožňuje větvení programu podle logické podmínky.

```
if <podminka>
    <prikaz>
elseif <podminka>
    <prikaz>
    ...
else
    <prikaz>
end
```

Příklad:

```
j=a;
for i=1:n
    if X(i,j)>0
       m=m+1;
    elseif X(i,j)<0
       m=m-1;
    else
       break;
    end
end
```

Příkaz `break` umožňuje přerušit provádění `for` nebo `while` cyklů. Příkaz `return` předá řízení volající funkci, případně umožní návrat do command window. Příkaz `input` umožňuje vstup dat z obrazovky.

```
<promenna> = input('<text>'[,'s'])
```

Příklad:

```
jmeno=input('Vase jmeno:','s');
rok=input('Rok narozeni:');
```

Příkaz `eval` interpretuje znakový řetězec.

Příklad:

```
string='clc,clear,a=[1 2;3 4],...
b=[0 0; 1 0], c=a*b'
eval string
a =
     1     2
     3     4
b =
     0     0
     1     0
c =
     2     0
     4     0
```

4 Koeficienty Čebyševových polynomů

Čebyševovy polynomy mohou být odvozeny takto:

Nechť $x = cos(\varphi), |x| \leq 1$ a předpokládejme, že existují polynomy $T_k(x) = cos(k\varphi)$, $cos(\varphi) = x$. Řešením diferenciální rovnice, pro $|x| \leq 1$,

$$(1-x^2)y'' - xy' + n^2 y = 0, \;\; y = T_n(x), \;\; n \geq 0 \quad (1)$$

dostaneme tyto polynomy:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1, \\ T_5(x) &= 16x^5 - 20x^3 + 5x \ldots, \end{aligned} \quad (2)$$

Tyto polynomy splňují rekurentní rovnici

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x),\ k = 1, \ldots, n, \end{aligned} \tag{3}$$

Model v MATLABu

```
function F=T(n)
% ************************************
% * Koeficienty Cebysevovych polynomu *
% *                                   *
% ************************************
pocitadlo = 1;
arg = [1 0];
a = [1];
b = [1 0];
shift = [0 0 1];
if n == 0
   F = a;
elseif n == 1
   F = b;
else
   while pocitadlo < n
      c = conv(2*arg,b) - conv(shift,a);
      a = b;
      b = c;
      pocitadlo = pocitadlo + 1;
   end
   F = c;
end
```

5 Jednoduchý dynamický model Nabídka-Poptávka

Vyjděme z ekonomického modelu, kdy nabídka v n-tém časovém intervalu $s(n)$ je přímo úměrná ceně produktu v minulém časovém intervalu $p(n-1)$ a naopak poptávka v n-tém časovém intervalu $dem(n)$ klesá se současnou cenou produktu $p(n)$, to znamená, že je úměrná $-dp(n)$. Zavedeme-li proměnnou $u(n)$, která charakterizuje počet vyrobených kusů v čase n, můžeme systém popsat soustavou tří diferenčních rovnic.

$$\begin{aligned} s(n) &= c \cdot p(n-1) + a \cdot u(n), \\ dem(n) &= -d \cdot p(n) + b \cdot u(n), \\ s(n) &= p(n). \end{aligned} \tag{4}$$

Pro obecně zvolené parametry a, b, c, d a $u(n) = 1(n)$ můžeme v MATLABu vytvořit následující model.

Model v MATLABu

```
% ********************************
% * Nabídka a poptávka           *
% * (jednoduchý dynamický model) *
% ********************************
clear
a = 100;
b = 120;
c = 3;
d = 4;
N = 50;
% vstupní posloupnost
u = ones(1,N);
p(1) = (b-a)/d;
% rovnice pro cenu p(n)
for k = 2:N
    p(k) = -c/d*p(k-1)+(b-a)/d*u(k);
end
figure(1)
subplot(3,1,1), stem(p)
% nabídka s(n)
for m = 1:N
    s(m) = c*p(m) + a*u(m);
end
subplot(3,1,2), plot(p,s,'*')
% poptávka dem(n)
for m = 1:N
    dem(m) = -d*p(m) + b*u(m);
end
subplot(3,1,3), plot(p,dem,'o',p,s,'*')
```