

Cvičení 3 – Programování v Matlabu

Modelování systémů a procesů

Lucie Kárná

karna@fd.cvut.cz

March 3, 2020

1 Grafy v Matlabu

2 Programování v Matlabu

- Funkce
- Model epidemie
- Klouzavý průměr
- Hod mincí

Základní grafické příkazy

Graf funkce

`figure` aktivuje okno pro obrázek

`plot` kreslí graf

- `plot(v)`, v je vektor:
 - na vodorovné ose index i
 - na svislé ose hodnoty $v(i)$
- `plot(A)`, A je matice (= tabulka):
 - na vodorovné ose řádkový index i
 - na svislé ose hodnoty $A(i, j)$
 - tj. pro každý **sloupec** j jeden graf
- `plot(x,y)`, x a y vektory stejné délky: XY-graf

Graf jedné funkce

Úloha 1

Nakreslete graf funkce $y = t \cdot e^{-\frac{1}{2}t}$ na intervalu $\langle 0, 5 \rangle$.

Řešení

```
» t = [0:0.1:5];  
» y = t.*exp(-0.5*t);  
» figure(1);  
» plot(t,y);
```

Doplnění grafu

Popis grafu

```
title titulek grafu  
xlabel, ylabel popisky os  
nesmím zavřít okno s grafem!
```

Vylepšení řešení úlohy 1

```
» title('Obrazek funkce v Matlabu');  
» xlabel('t');  
» ylabel('f(t)');
```

Grafy více funkcí

Úloha 2

Nakreslete do jednoho obrázku grafy funkcí

$$f_1(t) = \frac{1}{4}t,$$

$$f_2(t) = e^{-\frac{1}{2}t},$$

$$f_3(t) = \frac{1}{4}t \cdot e^{-\frac{1}{2}t}.$$

Řešení

```
» t = [0:0.1:5];  
» y1 = 0.25*t;  
» y2 = exp(-0.5*t);  
» y3 = y1 .* y2;
```

`legend` legenda grafu
`xlim, ylim` limity os

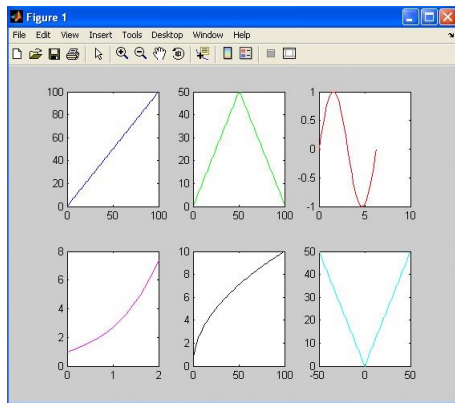
Grafy více funkcí

Řešení úlohy 2 - pokračování

```
» ...  
» figure(2);  
» plot(t,y1,t,y2,t,y3);          % 'plot' musí být první  
» title('Obrazek tri funkci v Matlabu');  
» xlabel('t');  
» ylabel('y');  
» legend('jedna', 'druha', 'treti');  
» legend('jedna', 'druha', 'treti', 'Location', 'nw');  
  
» % nebo 'northwest'
```

Více grafů v jednom okně

`subplot(m,n,p)` podobrázek



- » `figure(1);`
- » `subplot (2,3,1);`
- » `plot(modra_fce);`
- » `subplot (2,3,2);`
- » `plot(zelena_fce);`
- » ...
- » `subplot (2,3,4);`
- » `plot(fial_fce);`
- » ...

m-funkce

Typy m-souborů

scripty sekvence příkazů

- všechny proměnné globální
- volají se jménem souboru

m-funkce funkce

- všechny proměnné lokální
- vstupní a výstupní parametry
- volají se jménem funkce a parametry
- jméno souboru **musí být totožné se jménem funkce**

Funkce

```
function [y1,y2] = fce(x1,x2) <příkazy> end  
funkce v .m souboru fce.m
```

Příklad 1

Naprogramujte funkci dvou proměnných $geom(x_1, x_2) = \sqrt{x_1 \cdot x_2}$.

Řešení

- v editoru napíšeme

```
function y = geom( x1, x2 )  
y = sqrt( x1 * x2 );  
end
```

- uložíme **pod jménem geom.m**
- voláme z *Command Window* např. » `a = geom(7, 11)`

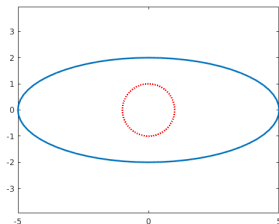
XY graf

Úkol

Naprogramovat funkci, která vykreslí elipsu s danými poloosami.
Do téhož obrázku nakreslit pro srovnání jednotkovou kružnici.

Řešení

```
function kresliElipsu(a, b)
t=[0:pi/32:2*pi];
x=cos(t);
y=sin(t);
plot(x,y,'r:',a*x,b*y)
axis equal
end
```



Kermack-McKendrickův SIR model jako funkce

Argumenty funkce (vstupní parametry):

```
% alpha ... koef. nakazlivosti  
% beta ... koeficient uzdraveni  
% S0 ... pocatecni pocet vnimavych jedincu  
% I0 ... pocatecni pocet infekcnich jedincu  
% n ... pocet iteraci
```

Návratové hodnoty (výstupní parametry):

```
% S ... vnimavi jedinci (vektor - pocty po dnech)  
% I ... infekcni jedinci (vektor - pocty po dnech)  
% R ... uzdraveni jedinci (vektor - pocty po dnech)
```

Hlavička funkce:

```
function [S,I,R] = SIR2(alpha,beta,S0,I0,n)
```

m-funkce

soubor SIR2.m

```
function [S,I,R] = SIR2( alpha, beta, S0 ,I0, n)
S = zeros(1,n+1); % pocatecni hodnota + n iteraci
S(1) = S0;
I = zeros(1,n+1);
I(1) = I0;
R = zeros(1,n+1); % R0 = 0

for j = 1:n
S(j+1) = S(j) - alpha*I(j)*S(j);
I(j+1) = I(j) + alpha*I(j)*S(j) - beta*I(j);
R(j+1) = R(j) + beta*I(j);
end end
```

Volání funkce

Funkci voláme z *Command Window* např. takto:

```
» [S,I,R] = SIR2(3e-5, 0.07, 10000, 5, 60);  
» A = [ S' I' R' ];  
» plot(A);  
» title('Epidemiologicky model');  
» xlabel('Dny');  
» ylabel('Pocet');  
» legend('Vnimavi' , 'Nakazeni', 'Uzdraveni');
```

Klouzavý průměr

- průměr konstantního počtu za sebou jdoucích hodnot
- slouží k vyhlazení časové řady

Úkol

- naprogramovat funkci `y = ravg(x, w)`
 - `x` vektor vstupních dat
 - `w` šířka okna (počet průměrovaných hodnot)
 - `y` výstup
- zobrazit `x` a `y` v grafu v různých barvách
- k průměrování použít funkci `mean()`
- počáteční hodnoty pro $i < w$ zkopírovat ze vstupu
- vstup: dopravní data
<http://zolotarev.fd.cvut.cz/static/msap/data.mat>

Klouzavý průměr – řešení

Funkce v souboru ravg.m

```
function y = ravg( x, w)
y = x;
for j = w : length( x )
i = j - w + 1;
y(j) = mean( x(i:j) );
end
```

Vykreslení výsledku

```
» load data.mat
» y = ravg ( data, 10 );
» plot( data );
» hold on;
» plot(y,'r','linewidth',2)
```


Simulace hodu férovou mincí s pravděpodobností panna-orel 50–50

Úkol

Naprogramovat funkci `[s,p,o]=coin(n)` simulující n hodů mincí.
Výstupem je

- vektor (řetězec) s obsahující symboly 'P' a 'O',
- celé číslo p udávající, kolikrát padla panna, a
- celé číslo o udávající, kolikrát padl orel.

Hod mincí – řešení

Řešení

```
function [s,p,o] = coin( n )  
s = char ( zeros ( 1, n ));  
p = 0, o = 0;  
for i = 1:n  
r = rand();  
if r < 0.5  
s(i) = 'P'; p = p + 1;  
else  
s(i) = '0'; o = o + 1;  
end end
```



T
E

H
N

E
D

Three white dice are arranged in a horizontal row, spelling out the word 'TEND'. The first die shows 'T' on top and 'E' on the front. The second die shows 'H' on top and 'N' on the front. The third die shows 'E' on top and 'D' on the front. A soft shadow is cast beneath the dice.