VÝUKA MOŽNOSTÍ MATLABU

Miroslav Olehla

Technická univerzita v Liberci, Fakulta strojní, Katedra aplikované kybernetiky

V následujícím příspěvku jsou uvedeny některé oblasti MATLABU ve výuce. Vychází se z učebních textů Metody zpracování dat – MATLAB a snahou je, provést studenty možnostmi uvedeného programovacího jazyka. Jsou uvedeny možnosti vizualizace, řízení průběhu výpočtu, relační a logické operátory, základní operace s maticemi.

1 Řešení soustavy lineárních rovnic.

Řešme soustavu přeurčených lineárních rovnic různými metodami – Gausova metoda, metoda Gauss Jordanova eliminace s částečnou pivotací, inverzí a pomocí determinantů.
Uvažujme soustavu **A.x** = **b**

$$3x_1 + 7x_2 - 2x_3 = 2$$
$$4x_1 + 3x_2 + 2x_3 = 5$$
$$2x_1 + 11x_2 - 6x_3 = -1$$
$$-1x_1 + 7x_2 + 3x_3 = -1$$

Zadání koeficientů řešené soustavy:

```
>> A=[ 3  7 -2; 4 3 2; 2 11 -6; -1 7 3]
A=
3            7           -2
4            3            2
2           11           -6
-1            7            3
>> b=[2;5;-1;-1]%nebo b=[2 5 -1 -1]'
b =
2
5
-1
-1
```

Kontrola řešitelnosti soustavy určením hodnosti a determinantu matice:

```
>> rank(A)
ans =
3
>> rank([A,b])
ans =
3
>>det(A'*A)
ans =
183750
```

- Gausova metoda

```
>> x=A\b
x =
1.1714
-0.1200
0.3371
```

- Metoda Gauss Jordanova eliminace s částečnou pivotací

```
>> C=rref([A,b])
C =
```

```
1.0000         0         0    1.1714
0    1.0000         0   -0.1200
0         0    1.0000    0.3371
0         0         0         0
```

- Řešení pomocí inverze

```
>>x=inv(A'*A)*(A'*b)
x =
1.1714
-0.1200
0.3371
```

- Řešení pomocí determinantů.

```
>> bt=A'*b
bt =
    25
    11
     9
>> At=A'*A
At =
    30    48   -13
    48   228   -53
   -13   -53    53
>> A1=[bt,At(:,2:3)]
A1 =
    25    48   -13
    11   228   -53
     9   -53    53
>> A2=[At(:,1),bt,At(:,3)]
A2 =
    30    25   -13
    48    11   -53
   -13     9    53
>> A3=[At(:,1:2),bt]
A3 =
    30    48    25
    48   228    11
   -13   -53     9
>> x1=det(A1)/det(At)
x1 =
    1.1714
>> x2=det(A2)/det(At)
x2 =
   -0.1200
>> x3=det(A3)/det(At)
x3 =
    0.3371
```

2 Aproximace periodické funkce

V případě, že je nutné aproximovat periodické funkce $f(x)$, získané na základě měření, lze aproximace získat funkcemi ve tvaru trigonometrických polynomů

$$f_m(x_i)=\frac{a_0}{2}+\sum_{n=1}^{M}(a_n\cos nx_i+b_n\sin nx_i)$$

Uvažujme, že aproximovaná funkce f má periodu 2π a že jsou známy její hodnoty v intervalu $\langle -\pi,\pi\rangle$. Označme

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos nx\,dx, \quad n = 0,1,2,....$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin nx\,dx, \quad n = 0,1,2,....$$

Předpokládejme, že známe $2N+1$ hodnot funkce f, která má periodu 2π. Využitím numerické integrace obdržíme v bodech $x_i = -\pi, -\frac{N-1}{N}\pi, ..., -\frac{\pi}{N}, 0, \frac{\pi}{N}, ..., \frac{N-1}{N}\pi, \pi$ a pro $M \le N$, kde M je stupeň trigonometrického polynomu (počet koeficientů a, b) vztahy

$$a_0 = \frac{1}{N}\sum_{i=-N}^{N-1} f(x_i)$$

$$a_n = \frac{1}{N}\sum_{i=-N}^{N-1} f(x_i)\cos mx_i, \quad m = 0,1,2,...., M$$

$$b_n = \frac{1}{N}\sum_{i=-N}^{N-1} f(x_i)\sin mx_i, \quad m = 0,1,2,...., M$$

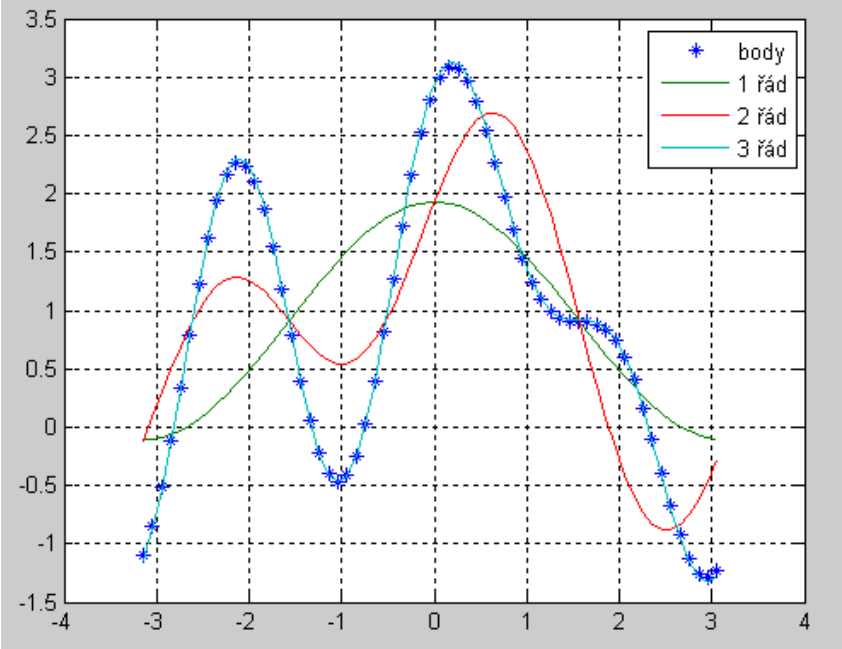
```
%Program v MATLABU:
% simulace vstupních dat
x=-pi:0.1:pi; N1=length(x)
y=(cos(x)+sin(2*x)+cos(3*x))+0.9;
% výpočet koeficientů
N=(N1-1)/2;
a0=1/N*sum(y)
a1=1/N*(sum(y*cos(x)'))
a2=1/N*( sum(y*cos(2*x)'))
a3=1/N*(sum(y*cos(3*x)'))
b1=1/N*(sum(y*sin(x)'))
b2=1/N*( sum(y*sin(2*x)'))
b3=1/N*(sum(y*sin(3*x)'))
%výpočet aproximační funkce
f1=a0/2+(a1*cos(x) +b1*sin(x));
f2=a0/2+(a1*cos(x) +b1*sin(x)+ a2*cos(2*x) +b2*sin(2*x) );
f3=a0/2+(a1*cos(x) +b1*sin(x)+ a2*cos(2*x) +b2*sin(2*x) +
a3*cos(3*x) +b3*sin(3*x));
%grafické zobrazení
plot(x,y, '*',x,f1, x,f2, x,f3)
grid on
legend('body', '1 řád', '2 řád', '3 řád')

nebo lépe:
% simulace vstupních dat
x=-pi:0.1:pi; N1=length(x)
y=(cos(x)+sin(2*x)+cos(3*x))+0.9;
% výpočet koeficientů
N=(N1-1)/2;
a0=1/N*sum(y);
M=3; % stupeň trigonometrickeho polynomu M
for k=1:M
a(k)=1/N*(sum(y*cos(k*x)'));
b(k)=1/N*(sum(y*sin(k*x)'));
end
a0
a
b
```

```
%výpočet aproximační funkce
f1=a0/2+(a(1)*cos(x) +b(1)*sin(x));
f2=a0/2+(a(1)*cos(x) +b(1)*sin(x)+ a(2)*cos(2*x) +b(2)*sin(2*x) );
f3=a0/2+(a(1)*cos(x) +b(1)*sin(x)+ a(2)*cos(2*x) +b(2)*sin(2*x) +
a(3)*cos(3*x) +b(3)*sin(3*x));
%grafické zobrazení
plot(x,y, '*',x,f1, x,f2, x,f3)
grid on
legend('body',  '1 řád', '2 řád', '3 řád')
```

Výsledek:

```
N1 =
   63
a0 =
   1.8178
a =
   1.0198   -0.0064    1.0198
b =
  -0.0003    1.0139   -0.0008
```



obr.2-1 Aproximace periodických funkcí pomocí Fourierovy řady

3 Lineární regrese

3.1 Jednoparametrová lineární regrese

Obecný tvar lineárního modelu s jednou nezávisle proměnnou je dán vztahem $y = a_0 + a_1 g(x)$, kde $g(x)$ může být nelineární, např. $g(x) = x^j$, $j = 2,3,.....$, nebo hyperbola, logaritmická funkce atd. U těchto funkcí můžeme k určení regresních koeficientů použít přímo metodu nejmenších čtverců.

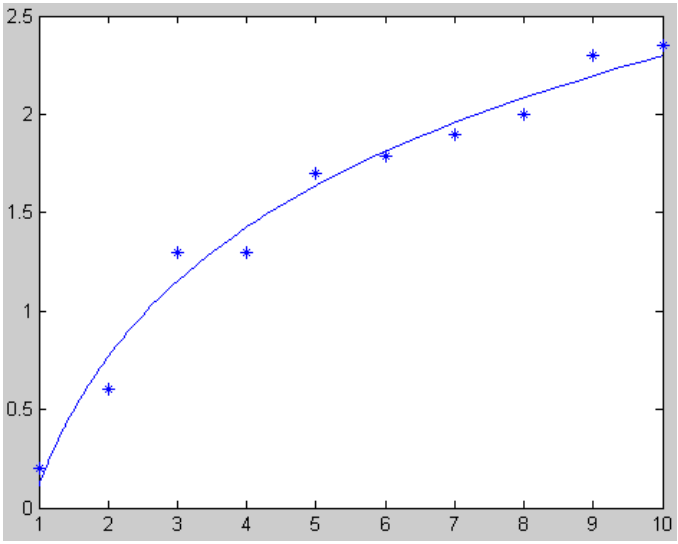
$y = a_0 + a_1 x^r$ pro $r<0$ musí být $x \neq 0$

$y = a_0 + a_1 \log x$ pro $x>0$

$y = a_0 + a_1 e^x$

Určeme koeficienty pro aproximační funkci $y = a_0 + a_1 \ln x$ pro naměřená data

x	y
1	0.20
2	0.60
3	1.30
4	1.30
5	1.70
6	1.79
7	1.90
8	2.00
9	2.30
10	2.35



Obr. 3-1 Lineární regrese

```
x=1:1:10;
y=[0.2 0.6 1.3 1.3 1.7 1.79 1.9 2.0 2.3 2.35];
X=[1 1 1 1 1 1 1 1 1 1 ; log(x)];
a=X'\y'
xgraf=1:0.1:10;
ygraf=a(1)+a(2)*log(xgraf);
plot(x,y,'*', xgraf,ygraf)
% nebo po dosazení a(1), a(2)
% plot(x,y,'*')
% hold on
%fplot('0.1106+0.949*log(x)',[1,10])

a =
 0.1106
 0.9490
```

V některých případech je možno provést následující transformaci

$y = a_0 a_1^{\,x}$	$\log y = \log a_0 + x \log a_1$
$y = a_0 x^{a_1}$	$\log y = \log a_0 + \log a_1 x$
$y = \dfrac{1}{a_0 + a_1 x}$	$1/y = a_0 + a_1 x$

3.2 Víceparametrová lineární regrese

Uvažujme naměřené hodnoty:

x_1	4	2	4	1	5	3	4
x_2	10	8	12	3	15	12	13
y	4	3	4	1	6	4	5

a určeme koeficienty regresní funkce

$y=a_0.1+a_1.x_1 + a_2.x_2.$

Doplněním vektoru jedniček má matice **X** tvar

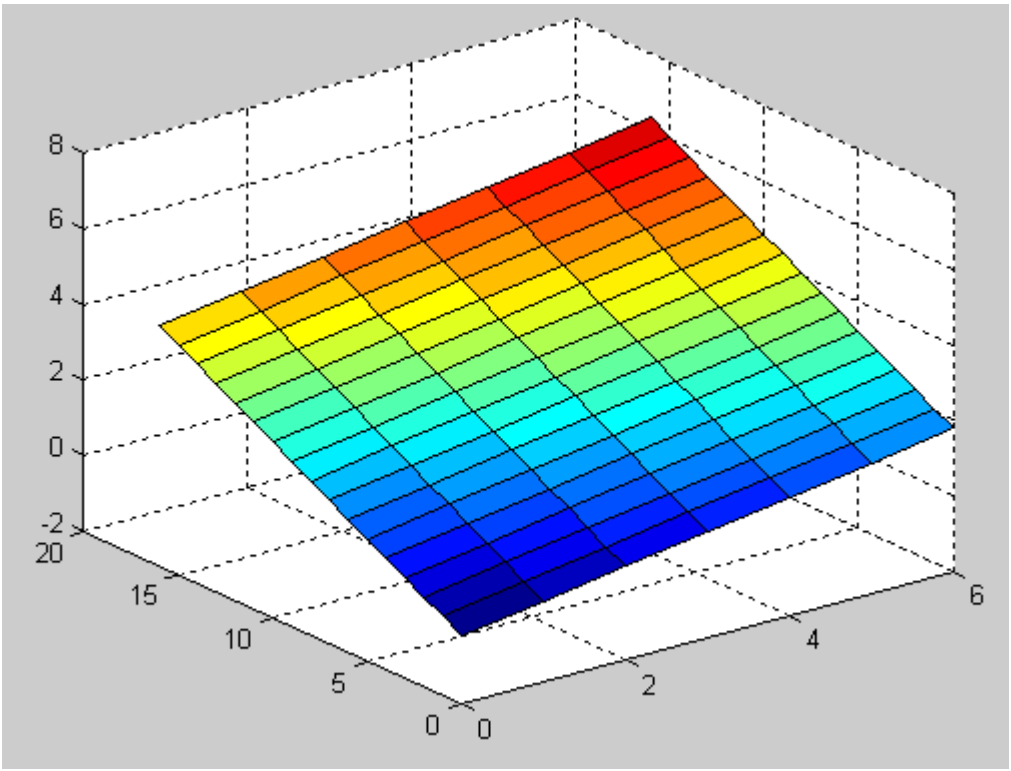
```
X=[1 1 1 1 1 1 1;4 2 4 1 5 3 4;10 8 12 3 15 12 13];
y=[4 3 4 1 6 4 5];
a=X'\y'
a =
-0.1741
 0.3281
 0.2832
```

Poznámka: Výsledkem $\mathbf{X}*\mathbf{X}^T$ je matice symetrická

```
X*X'
ans =
     7    23    73
    23    87   270
    73   270   855
```

Vytvoření grafu:

```
xp=[0:1:6];
yp=[0:1:16];
[XP,YP]=meshgrid(xp,yp);
zp=a(1)+a(2)*XP+a(3)*YP;
surf(xp,yp,zp)
```



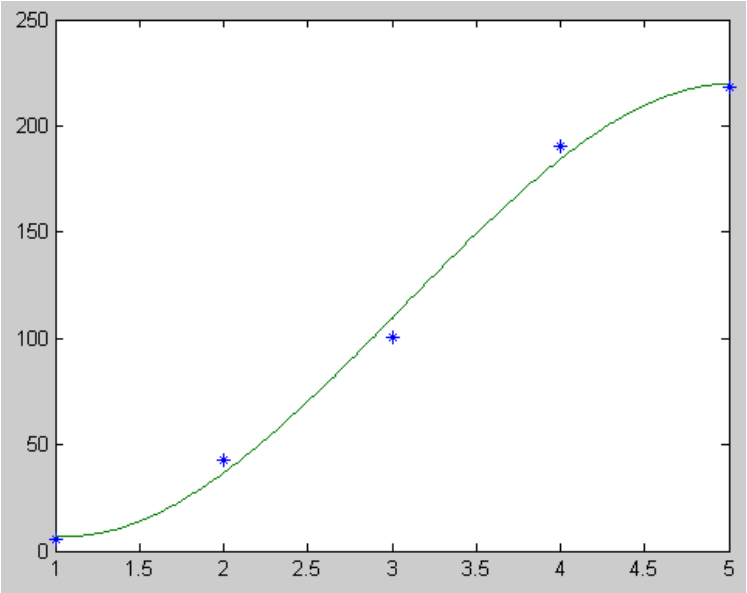
obr. 3-2 Víceparametrová lineární regrese

3.3 Aproximace pomocí polynomu

Pro uvedené výsledky měření uvažujme aproximaci ve tvaru polynomu 3. stupně, tedy $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$.

```
x=[1 2 3 4 5];
X=[1 1 1 1 1 ;x;x.^2;x.^3];
y=[5.5 43.1 100.2 190.7 218.4];
a=X'\y'
 a=
```

```
   61.5800
 -110.3452
   62.6964
   -6.8583
% Vykreslení grafu:
xgraf=1:0.01:5;
graf=a(1)+a(2)*xgraf+a(3)*xgraf.^2+a(4)*xgraf.^3;
plot(x,y,'*',xgraf,ygraf);
%nebo po dosazení koeficientů
% plot(x,y,'*')
% hold on
%fplot('61.58-110.345*x+62.696*x.^2-6.858*x.^3',[1,5])
%není povolen zápis
fplot('a(1)+a(2)*x+a(3)*x.^2',[1,5])
```

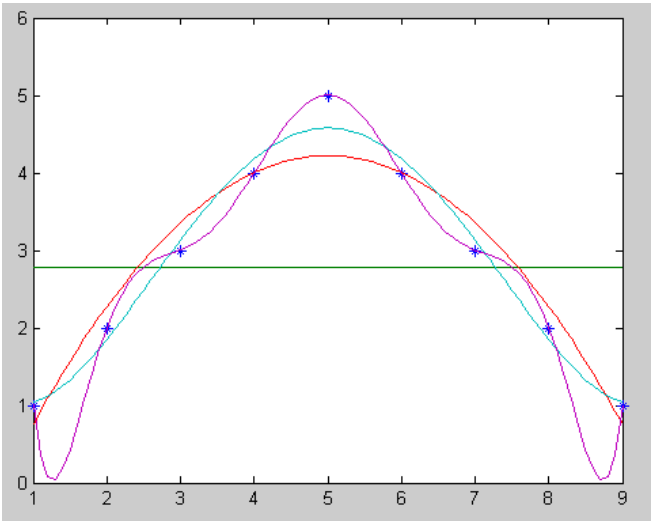


Obr. 3-3 Regrese polynomem 3. stupně

V případě, že chceme data aproximovat polynomem *n*-tého stupně, můžeme to provést pomocí funkce `p=polyfit(x,y,n).` Výsledek je stejný, jako v předchozích případech.
x je vektor hodnot nezávisle proměnné,
y je vektor hodnot závisle proměnné,
n je stupeň polynomu, jímž chceme aproximovat body $[x_i,y_i]$,
p je vektor koeficientů výsledného polynomu $P(x) = a_n.x^n + a_{n-1}.x^{n-1} + ... + a_1.x + a_0$

```
x=[10 21 35 37 50 61 70];
y=[0.52 1.7 1.58 2.11 2.63 3.2 4.9];
x1=10:0.1:70; p=polyfit(x,y,1)
y1=polyval(p,x1);
plot(x,y,'*',x1,y1)
p =
       0.618   -0.1290
```

Musíme připravit datové vektory odpovídající jednotlivým grafům. Zkusme nadefinovat několik bodů a proložit jimi polynomy různých stupňů pomocí funkce `polyfit`. Prokládané body jsou dány vektory **x** a **y**. V grafu budou vyznačeny hvězdičkami. Za povšimnutí stojí průběh polynomu *y*8 – aproximace tohoto stupně je interpolací.



```
x=[1 2 3 4 5 6 7 8 9];
y=[1 2 3 4 5 4 3 2 1];
x1=1:0.1:9;
y1=polyval(polyfit(x,y,1),x1);
y2=polyval(polyfit(x,y,2),x1);
y5=polyval(polyfit(x,y,5),x1);
y8=polyval(polyfit(x,y,8),x1);
plot(x,y,'*',x1,y1,x1,y2,x1,y5
,x1,y8)
```

Obr. 3-4 Grafické okno funkce polyfit

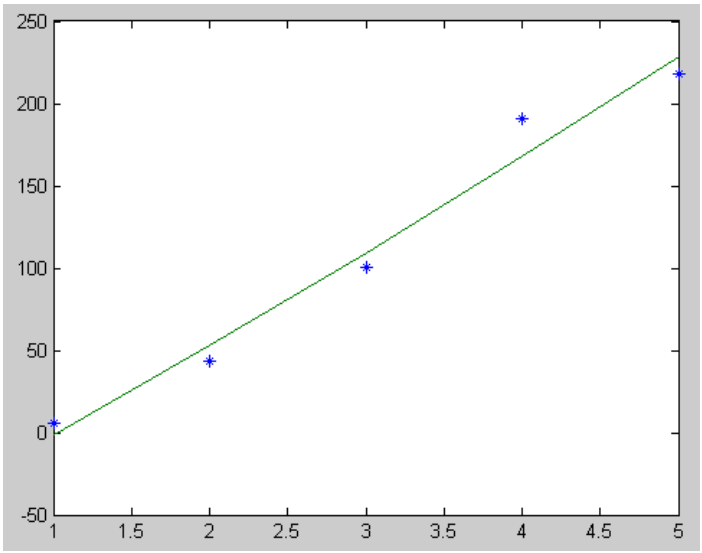
Dalším příkladem je nalezení koeficientů polynomu zvoleného stupně prokládajícího naměřená data podle kritéria nejmenších čtverců:

```
x=[1 2 3 4 5];
y=[5.5 43.1 100.2 190.7 218.4];
r=polyfit(x,y,2)
% nalezení koeficientů polynomu 2. stupně
yp2=polyval(r,x);
plot(x,y,'*',x,yp2)
```

Výsledek:

```
r =
     0.9714 51.5114 -53.6400
```

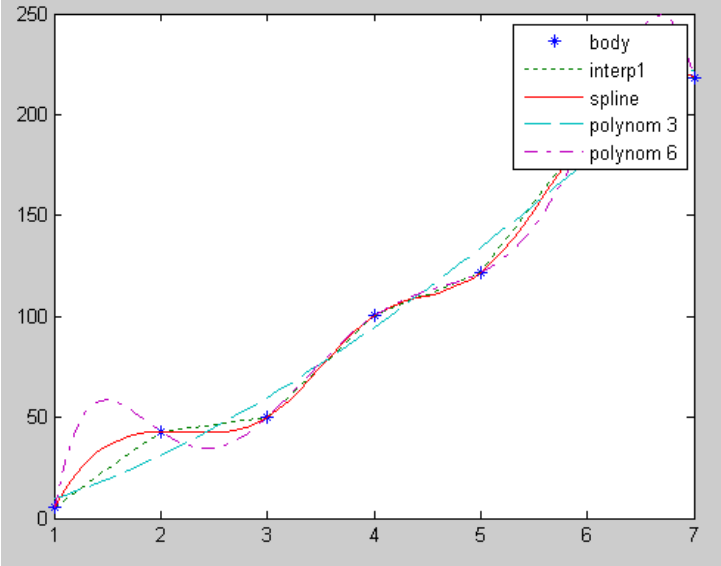
tedy polynom ve tvaru `0.9714.x^2+51.5114.x-53.6400`



obr. 3-5 Polynomická aproximace

S proložením bodů pomocí polynomu úzce souvisí interpolace[1], respektive proložení polynomem je jeden typ interpolace. MATLAB poskytuje několik standardních funkcí pro různé typy interpolačních funkcí. Popis všech funkcí a jejich použití přesahuje rozsah tohoto textu.

Uveďme názornou ukázku použití funkce `polyval` pro aproximaci, `interp1, spline` pro interpolaci a použijeme příkaz `plot` pro grafické zobrazení zadaných bodů a bodů získaných proložením polynomem 3. a 6. stupně, interpolační funkcí a kubickými spline.



obr. 3-6 Nevhodně použitá polynomiální interpolace

```
x = [1:7];
y=[5.5 43 50 100.2 122 190.9 218];
% interpolace
xp=1:0.1:7;
yi=interp1(x,y,xp);
%spline
yp=spline(x,y,xp);
%aproximace
yp3=polyval(polyfit(x,y,3),xp);
%aproximace = interpolace
yp6=polyval(polyfit(x,y,6),xp); plot(x,y,'*',xp,yi,
':',xp,yp,xp,yp3,'--',xp,yp6, '-.')
legend('body', 'interp1','spline','polynom 3','polynom 6')
```

Příklad ukazuje nebezpečí bezmyšlenkového použití polynomiální aproximace. Pokud je aproximujeme polynomem 6. stupně a dopočítáme hodnoty pro dané *x* dostaneme ideální shodu. Pokud se však podíváme i na hodnoty mimo sledované body už spokojeni nebudeme. Aproximace se sice přesně shoduje v zadaných bodech, ovšem průběh mezi zadanými body se značně odchyluje od očekávaného průběhu. Polynomiální aproximace není rovněž vhodná pro aproximaci průběhů s ostrými zlomy.

Literatura:

Olehla, M., Dušek, F.: *Metody zpracování dat – MATLAB.* Technická Univerzita Liberec 2013, ISBN 978-80-7372-952-3

[1] Aproximace polynomem n-1 stupně