Interpolace a aproximace dat.

Interpolace dat křivkou (funkcí) - křivka (graf funkce) prochází daty (body) **přesně**.

Aproximace dat křivkou (funkcí) - křivka (graf funkce) prochází daty (body) **přibližně**.

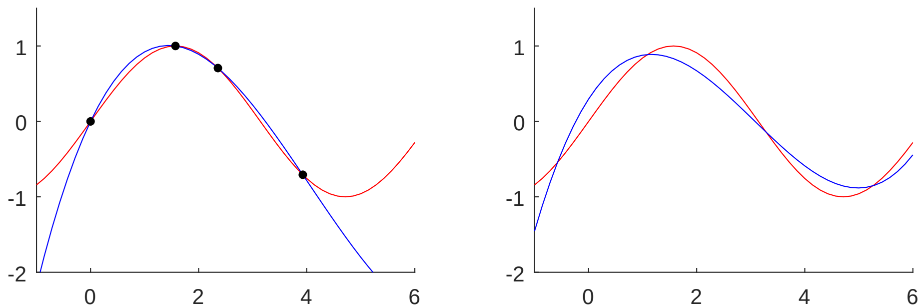


Figure 1: Náhrada $f(x) = \sin(x)$ na $(-1, 6)$ polynomem čtvrtého stupně.

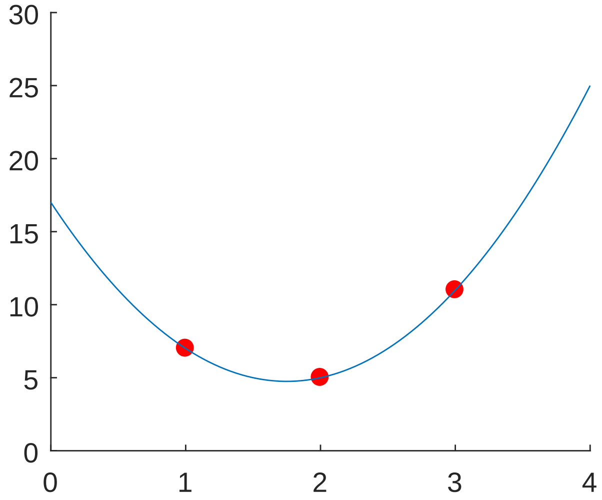
Nejčastěji se za křivky (funkce) používají polynomy nebo lineární kombinace goniometrických funkcí $\sin(kx)$ a $\cos(kx)$.

Interpolace

Příklad. Proložte body $[1, 7]$, $[2, 5]$ a $[3, 11]$ polynomem 2. stupně.

[... řešení na tabuli ... soustava lineárních rovnic ...]

Výsledek: polynom $p(x) = 4x^2 - 14x + 17$



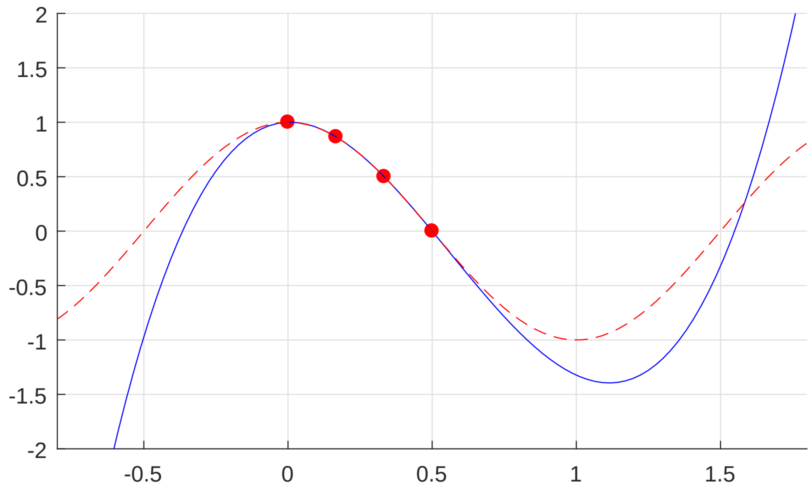

```
cla; hold on;
plot([1,2,3],[7,5,11],'r.','MarkerSize',20);
p = polyfit([1,2,3],[7,5,11],2) % koef. pol.
x = linspace(0,4,100); % vektor x pro kresleni
y = polyval(p,x); % hodnoty polynomu v x
plot(x,y);
```

K čemu potřebujeme interpolaci polynomem?

Příklad. Nahraďte funkci $f(x) = \cos(\pi x)$ na intervalu $(0, \frac{1}{2})$ polynomem 3. stupně.

Zvolíme 5 bodů na intervalu $(0, \frac{1}{2}) \longrightarrow$ vypočítáme v nich funkční hodnoty funkce $f(x)$ $\longrightarrow$ získané body v rovině proložíme (jako v předchozím příkladě) polynomem (tentokrát stupně 3).

Výsledek: $p(x) = 1 + 0.0885x - 5.9423x^2 + 3.5307x^3$.

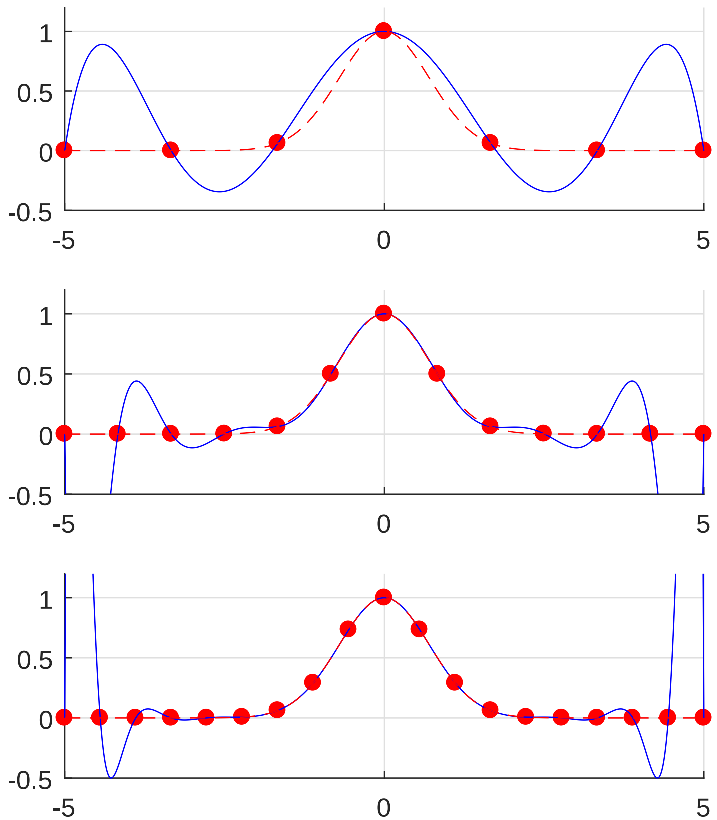



```
cla; hold on;
plot([0,1/6,1/3,1/2],[1,sqrt(3)/2,1/2,0],...
     'r.','MarkerSize',20);
p = polyfit([0,1/6,1/3,1/2],[1,sqrt(3)/2,1/2,0],3)
x = linspace(-0.8,1.8,100);
y = polyval(p,x);
plot(x,y,'b');
plot(x,cos(pi*x),'r--');
grid on
axis([-0.8,1.8,-2,2]);
```

Nyní, potřebujeme-li **přibližnou** hodnotu $\cos(0.3\pi)$, stačí spočítat
$p(0.3) = 1 + 0.0885 \cdot 0.3 - 5.9423 \cdot 0.3^2 + 3.5307 \cdot 0.3^3$.

Nebo:

Příklad. Nahraďte funkci $f(x) = \mathrm{e}^{-x^2}$ (Gaussovo rozdělení) na intervalu $(-5, 5)$ polynomem 6., 12. a 18. stupně,




```
cla; hold on;
N = 7; %7, 13 nebo 19
a = 5; % interval (-a,a)
x1 = linspace(-a,a,N); % data
y1 = exp(-x1.^2); % data
plot(x1,y1,'r.','MarkerSize',20);
p = polyfit(x1,y1,N-1)
x = linspace(-a,a,1000);
y = polyval(p,x);
plot(x,y,'b');
plot(x,exp(-x.^2),'r--');
grid on
axis([-a,a,-0.5,1.2]);
```

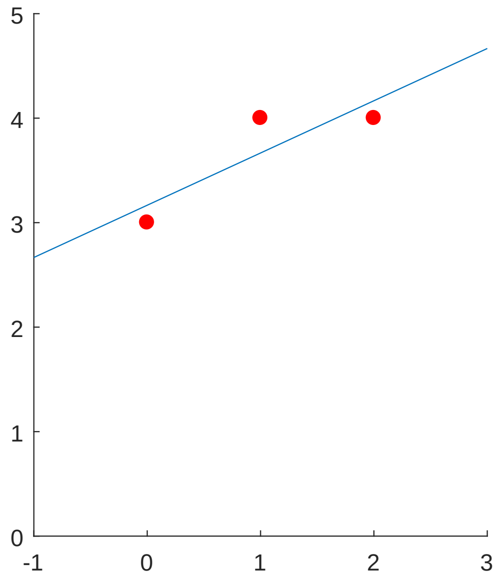
Oscilace na krajích intervalu! Interpolace polynomem vysokých stupňů je nevhodná.

Aproximace - křivka nepochází daty přesně ale jen **přibližně**.

Příklad. Proložte body $[0, 3]$, $[1, 4]$ a $[2, 4]$ polynomem 1. stupně. (Lineární regrese.)

[... řešení na tabuli ... soustava lineárních rovnic ...]

Výsledek: $p(x) = \frac{19}{6} + \frac{1}{2}x \approx 3.1667 + 0.5\,x.$



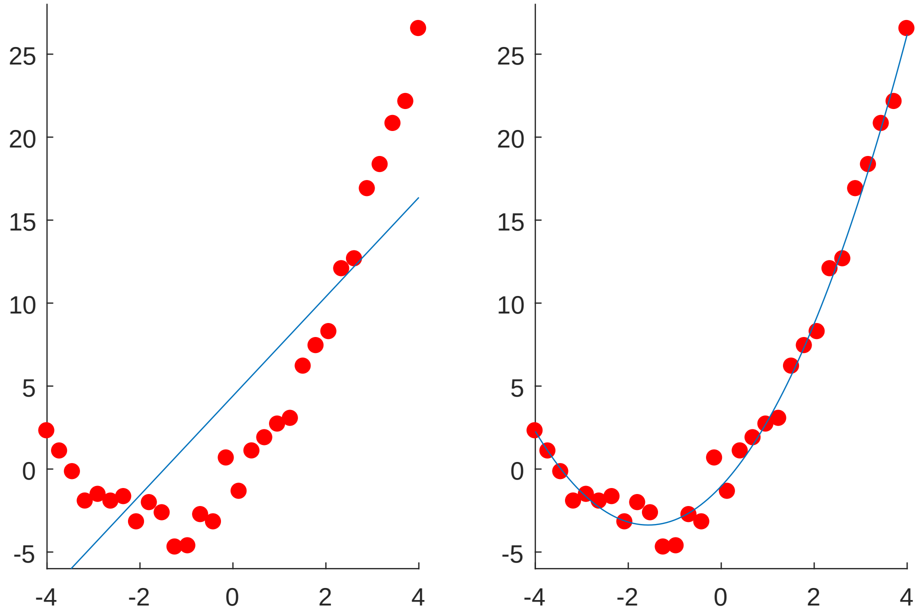

```
cla; hold on;
plot([0,1,2],[3,4,4],'r.','MarkerSize',20);
p = polyfit([0,1,2],[3,4,4],1)
x = linspace(-1,3,100);
y = polyval(p,x);
plot(x,y);
axis([-1,3,0,5]);
```

Je vidět, že funkce $p(x)$ neprochází zadanými body přesně, ale přesto "dobře vystihuje jejich chování (průběh)".

Příklad. Proložte body $[x_1, y_1], [x_2, y_2], \ldots, [x_N, y_N]$ polynomem 1. stupně a polynomem 2. stupně.

Lineární regrese; polynomická regrese; metoda nejmenších čtverců.

[... řešení na tabuli ... soustava lineárních rovnic ...]




```
cla; hold on;
% 30 bodu na ose x:
x1 = linspace(-4,4,30);
% data s nahodnym sumem:
y1 = x1.^2+3*x1-1+randn(1,30);
plot(x1,y1,'r.','MarkerSize',20);
p = polyfit(x1,y1,2) % 1 nebo 2
x = linspace(-4,4,100);
y = polyval(p,x);
plot(x,y);
axis([-4,4,-6,28])
```

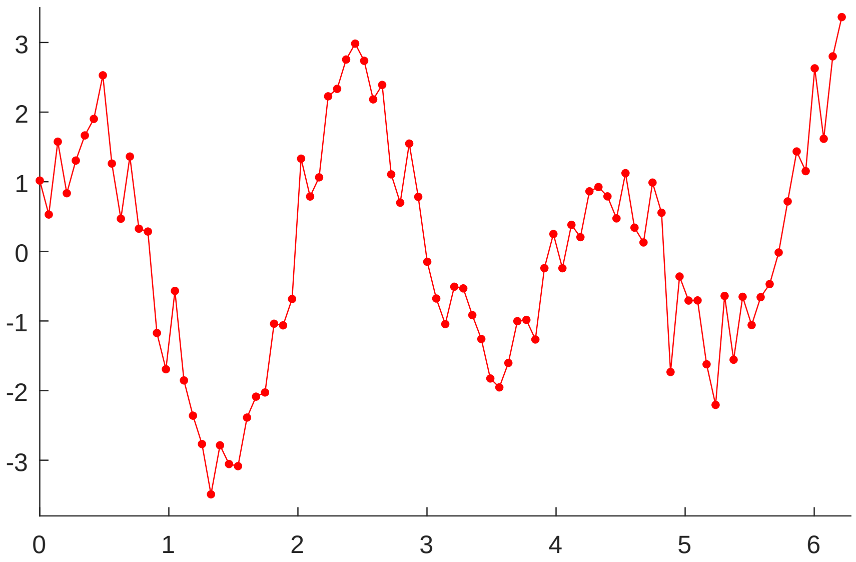
Dosud pouze aproximace pomocí polynomů. Nyní lineární kombinace goniometrických funkcí $\sin(kx)$ a $\cos(kx)$ a 1, $k = 1, 2, \ldots, N$.

- důležité pro fyzikální aplikace

- odstranění šumu u zvukových signálů, MRI

- elektromagnetické vlnění (radary), atd.

Příklad. Aproximujte zadaná data $[x_1, y_1]$, $[x_2, y_2]$, $\ldots$, $[x_N, y_N]$ lineární kombinací funkcí $\sin(x)$, $\sin(2x)$, $\sin(3x)$, $\sin(4x)$, 1, $\cos(x)$, $\cos(2x)$, $\cos(3x)$, $\cos(4x)$,



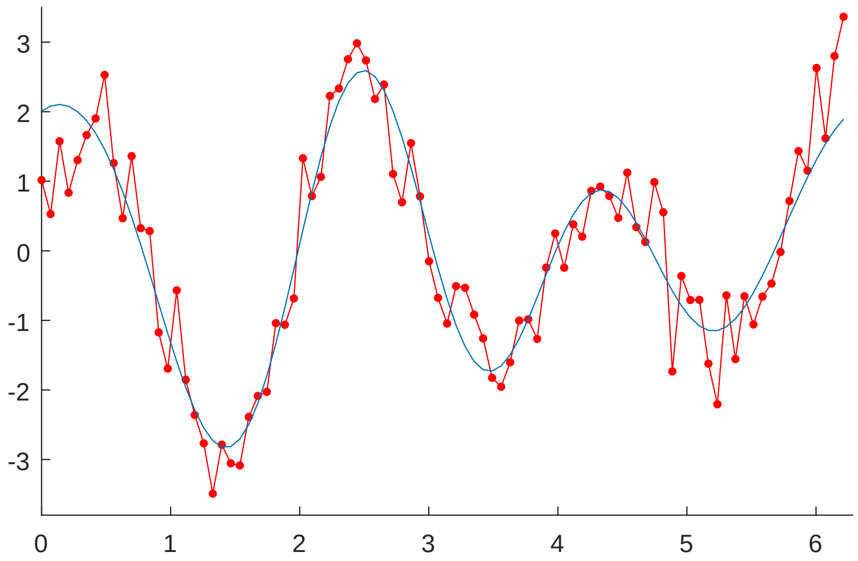
Příklad. (Opak.) Aproximujte zadaná data $[x_1, y_1]$, $[x_2, y_2]$, …, $[x_N, y_N]$ lineární kombinací funkcí $\sin(x)$, $\sin(2x)$, $\sin(3x)$, $\sin(4x)$, 1, $\cos(x)$, $\cos(2x)$, $\cos(3x)$, $\cos(4x)$.

Jak se to spočítá:

V bodech $x_1, x_2, \ldots, x_N$ se vyhodnotí všechny funkce $\sin(x)$, $\sin(2x)$, $\sin(3x)$, $\sin(4x)$, 1, $\cos(x)$, $\cos(2x)$, $\cos(3x)$ a $\cos(4x)$. Získaných 9 vektorů hodnot se jako sloupce sestaví do matice, kterou označíme Y. Potom sestrojíme matici $A = Y^T Y$ a vektor $b = Y^T y$, kde y je sloupcový vektor dat $(y_1, y_2, \ldots, y_N)^T$. Potom vyřešíme soustavu rovnic $Au = b$. Získaný vektor $u = (u_1, \ldots, u_N)^T$ obsahuje koeficienty takové, že hledaná funkce je

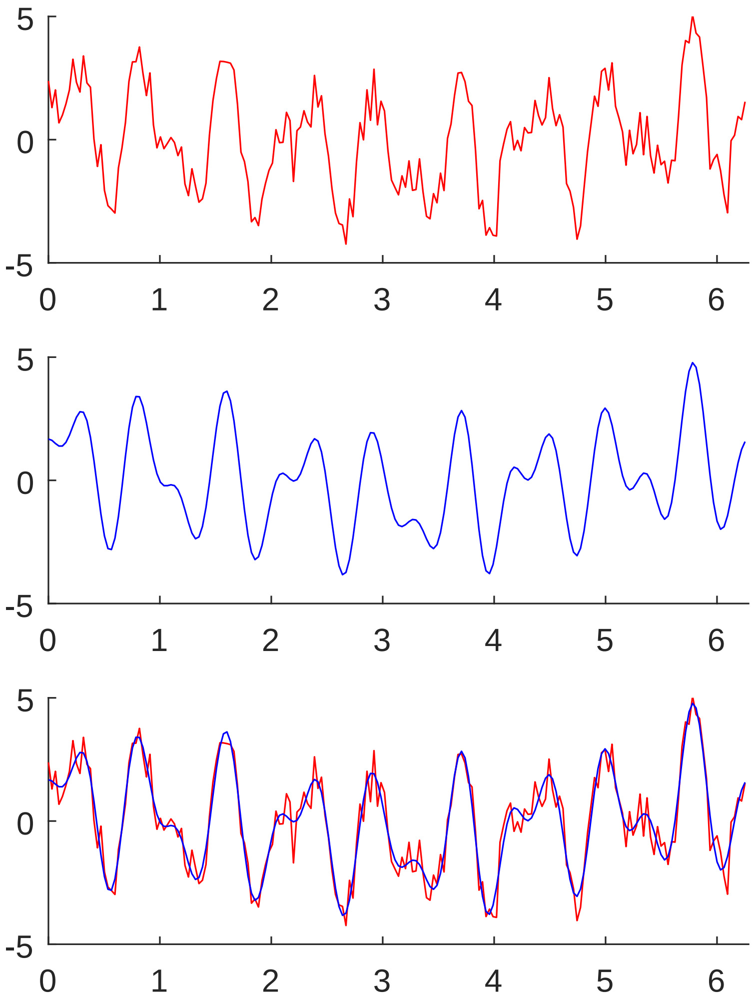
$$p(x) = u_1 \sin(x) + u_2 \sin(2x) + \ldots + u_4 \sin(4x) + u_5 + u_6 \cos(x) + \ldots + u_9 \cos(4x).$$

Diskrétní Fourierova transformace. Funkce fft a ifft v Matlabu.




```
cla; hold on;
% 90 bodu na ose x:
x = linspace(0,2*pi,91); x(end) = [];
% data s nahodnym sumem:
y = 2*sin(3.2*x)+1*cos(1.9*x+1)+randn(1,90)/2;
plot(x,y,'r.','MarkerSize',10);
plot(x,y,'r');
% diskretni Fourierova transformace:
z = fft(y);
% vynulovani vysokych frekvenci:
z(6:end-4) = 0;
% inverzni diskretni Fourierova transformace:
y = ifft(z);
plot(x,y);
```

Příklad. Odstraňte šum v zadaném akustickém signálu. Šum obvykle odpovídá vysokým frekvencím v datech.



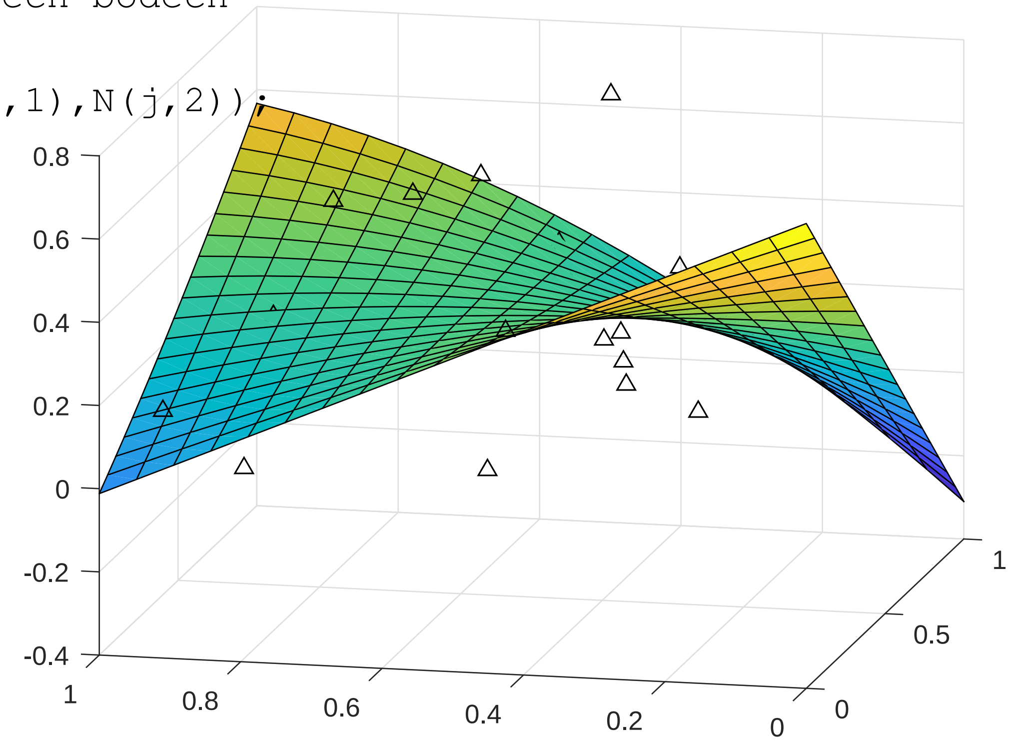

```
cla; hold on;
% N bodu na ose x:
N = 200;
x = linspace(0,2*pi,N+1); x(end) = [];
% data s nahodnym sumem:
y = cos(x)+2*sin(9*x)+1.5*cos(15*x+1)+...
    randn(1,N)*0.6;
% plot(x,y,'r.','MarkerSize',10);
plot(x,y,'r');
% diskretni Fourierova transformace:
z = fft(y);
% vynulovani vysokych frekvenci:
M = round(N/10);
z(M+1:end-M+1) = 0;
% inverzni diskretni Fourierova transformace:
y = ifft(z);
plot(x,y,'b');
axis([0,2*pi,-5,5]);
```

Příklad. Aproximujte (náhodná) data v prostoru.

```
N = [0,0;1,0;0,1;1,1;1,2;2,1]; % exponenty x a y
Nmax = size(N,1); % pocet polynomu
M = 20; % pocet dat
data = rand(M,3); % data v (0,1)x(0,1)
Y = zeros(M,Nmax);
for k = 1:M % vyhodnoceni vsech pol. ve vsech bodech
    for j = 1:Nmax
        Y(k,j) = F(data(k,1),data(k,2),N(j,1),N(j,2));
    end;
end;
A = Y'*Y;
B = Y'*data(:,3);
u = A\B; % reseni (koeficienty)

// Zde kresleni ...

function z = F(x,y,n1,n2)
z = x^n1*y^n2;
```



…dokončení:

```
// Kresleni:

x = linspace(0,1,20);
y = linspace(0,1,20);
[X,Y] = meshgrid(x,y);
for k = 1:20 % vypocet pro kresleni
    for j = 1:20
        pom = 0
        for n = 1:Nmax
            pom = pom + u(n)*F(X(k,j),Y(k,j),N(n,1),N(n,2));
        end;
        Z(k,j) = pom;
    end;
end;
cla; hold on;
plot3(data(:,1),data(:,2),data(:,3),'k^');
surf(X,Y,Z); grid on;
```