

რიცხვითი ინტერპოლაცია

არ არსებული (დაკარგული) ინფორმაციის აღდგენა არსებული მონაცემებიდან გარკვეული დაშვების ფარგლებში
(ფუნქციის უწყვეტობა, გლუვობა, ფორმა, ტრაექტორიის ჩაკეტილობა ...)

პირდაპირი (გულუბრყვილო) ინტერპოლაციის მეთოდი:

$$F(1) = 1;$$

$$F(2) = 2;$$

$$F(3) = 3;$$

$$F(4) = \dots$$

$$F(5) = 4;$$

$$F(6) = 4;$$

$$F(7) = 4;$$

ნულოვანი რიგის ინტერპოლაცია

$$F(k) = F(k-1);$$

პირველი რიგის ინტერპოლაცია

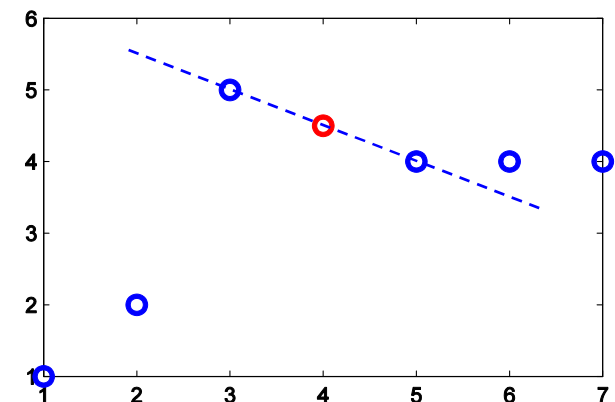
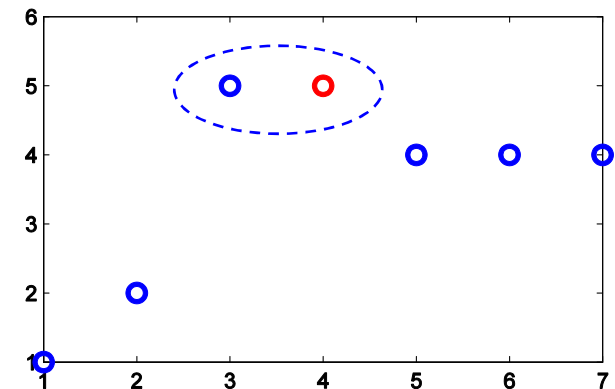
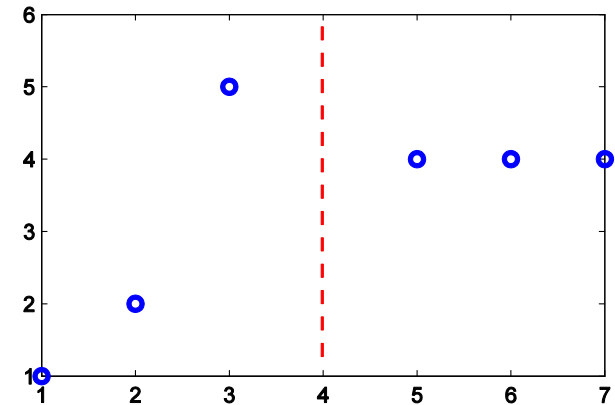
Piecewise Linear interpolation:

$$F(k-1) = a \cdot x(k-1) + b$$

$$F(k+1) = a \cdot x(k+1) + b$$

გამოვითვალათ a, b ;

$$F(k) = a \cdot X(k) + b$$



მაღალი რიგის ინტერპოლაცია

ნაწილ-ნაწილ კვადრატული ინტერპოლაცია
Piecewise Quadratic interpolation:

$$F(k) = a \cdot x(k)^2 + b \cdot x(k) + c$$

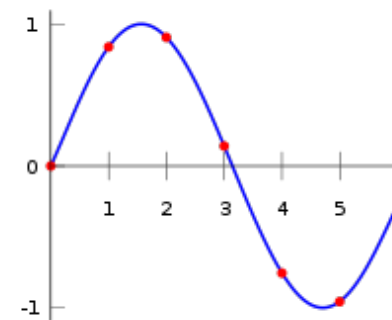
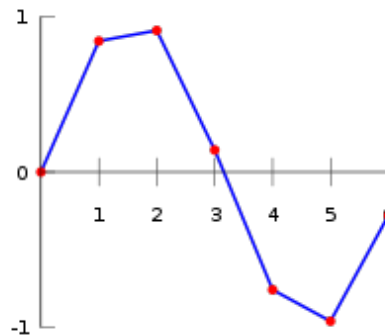
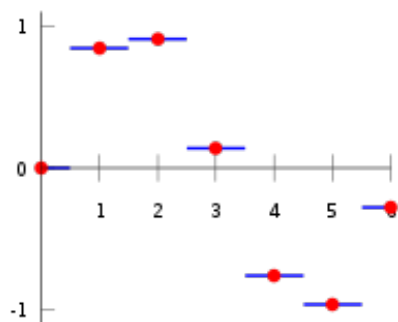
გამოვითვალთ: a, b, c

გამოთვლებისას გამოვიყენოთ წერტილები

Forward scheme: $F(k-2), F(k-1), F(k+1)$

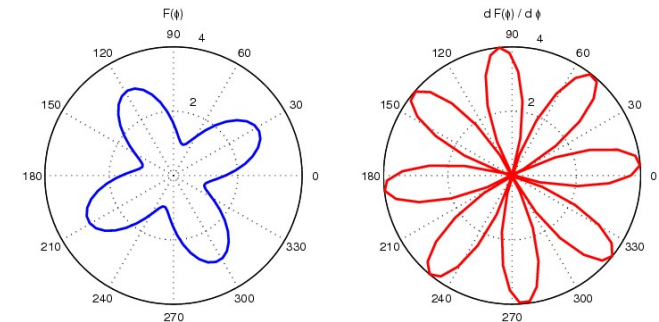
Backward scheme: $F(k-1), F(k+1), F(k+2)$

ნულოვანი, პირველი და მაღალი რიგის ინტერპოლაცია:





როგორ მუშაობენ გრაფიკული ფუნქციები:
polar



დავალება 5.1.

რიცხვითი მეთოდების გამოყენებით გამოითვალეთ $F(\varphi) = 2 + \sin(\omega\varphi - \pi/8)$ ფუნქციის პირველი რიგის წარმომადგენელი $G(\varphi) = dF(\varphi)/d\varphi$ ცენტრალური წარმომადგენლის (G_2) და ხუთ წერტილიანი წარმომადგენლის (G_5) გამოყენებით, როდესაც $\Omega=4$ და φ დისკრეტულ ვექტორში წერტილების რაოდენობაა $N_\varphi = 80$. გამოითვალეთ F ფუნქციის მეორე რიგის წარმომადგენლები $H(\varphi) = d^2F(\varphi)/d\varphi^2$ ცენტრალური წარმომადგენლის (H_2) და ხუთ წერტილიანი წარმომადგენლის (H_5) გამოყენებით.

$G_{2,5}$ ვექტორებიდან აღადგინეთ N_φ წერტილის მქონე $L_{2,5}$ ვექტორები ნულოვანი ინტერპოლაციის გამოყენებით.

$G_{2,5}$ ვექტორებიდან აღადგინეთ N_φ წერტილის მქონე $M_{2,5}$ ვექტორები წრფივი ინტერპოლაციის გამოყენებით.

$H_{2,5}$ ვექტორებიდან აღადგინეთ N_φ წერტილის მქონე $N_{2,5}$ ვექტორები ნულოვანი ინტერპოლაციის გამოყენებით.

$H_{2,5}$ ვექტორებიდან აღადგინეთ N_φ წერტილის მქონე $O_{2,5}$ ვექტორები წრფივი ინტერპოლაციის გამოყენებით.

ააგეთ პოლარული მრუდები:

პირველ გრაფიკზე: L_2 , L_5 ; მეორე გრაფიკზე M_2 , M_5 ; მესამე გრაფიკზე N_2 , N_5 , მეოთხე გრაფიკზე O_2 , O_5 .

ინტერპოლაცია ლაგრანჟის პოლინომებით

ზოგიერთ შემთხვევაში გლუვი ფუნქციების პირობებში ინტერპოლაციის მიზანი შეიძლება იყოს K წერტილიდან N წერტილის აღდგენა, როდესაც $N \gg K$ (!)

მეთოდი: აღვადგინოთ ანალიზური ფუნქცია, ხოლო ანალიზური ფუნქციიდან აღვადგინოთ N დისკრეტული წერტილი.

ვთქვათ მოცემულია $k+1$ წერტილი რომელთაგანაც არცერთი არ მეორდება:

$$(x_0, y_0), \dots, (x_j, y_j), \dots, (x_k, y_k)$$

ლაგრანჟის პოლინომი:

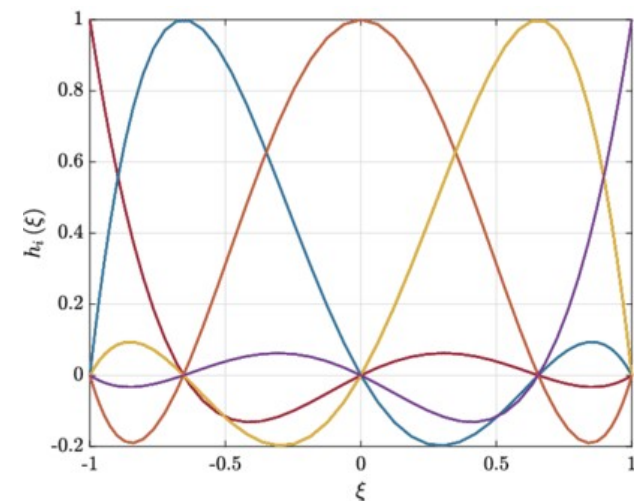
$$L(x) := \sum_{j=0}^k y_j \ell_j(x)$$

ლაგრანჟის საბაზისო პოლინომები:

$$\ell_j(x) := \prod_{\substack{0 \leq m \leq k \\ m \neq j}} \frac{x - x_m}{x_j - x_m} = \frac{(x - x_0) \dots (x - x_{j-1}) (x - x_{j+1}) \dots (x - x_k)}{(x_j - x_0) \dots (x_j - x_{j-1}) (x_j - x_{j+1}) \dots (x_j - x_k)},$$

აუცილებელი პირობა: წერტილები არ მეორდება

$$x_j - x_m \neq 0$$



(a) Lagrange polynomials

მაგალითი 1.

ავიღოთ $f(x) = x^2$ ფუნქციის 3 წერტილი:

$$x_0 = 1 \quad f(x_0) = 1$$

$$x_1 = 2 \quad f(x_1) = 4$$

$$x_2 = 3 \quad f(x_2) = 9.$$

ავაგოთ ინტერპოლირებული ფუნქცია ლაგრანჟის პოლინომის გამოყენებით:

$$\begin{aligned} L(x) &= 1 \cdot \frac{x-2}{1-2} \cdot \frac{x-3}{1-3} + 4 \cdot \frac{x-1}{2-1} \cdot \frac{x-3}{2-3} + 9 \cdot \frac{x-1}{3-1} \cdot \frac{x-2}{3-2} \\ &= x^2. \end{aligned}$$

მაგალითი 2.

ავიღოთ $f(x) = x^3$ ფუნქციის 3 წერტილი:

$$x_0 = 1 \quad f(x_0) = 1$$

$$x_1 = 2 \quad f(x_1) = 8$$

$$x_2 = 3 \quad f(x_2) = 27$$

ავაგოთ ინტერპოლირებული ფუნქცია ლაგრანჟის პოლინომის გამოყენებით:

$$\begin{aligned} L(x) &= 1 \cdot \frac{x-2}{1-2} \cdot \frac{x-3}{1-3} + 8 \cdot \frac{x-1}{2-1} \cdot \frac{x-3}{2-3} + 27 \cdot \frac{x-1}{3-1} \cdot \frac{x-2}{3-2} \\ &= 6x^2 - 11x + 6. \end{aligned}$$

n წერტილი: $(n-1)$ რიგის პოლინომიური ინტერპოლანტი;

აპროქსიმაცია ინტერპოლაციით:

- პოლინომიური აპროქსიმაცია ფიქსირებული წერტილებით;
- პოლინომის რიგის კლების აპროქსიმაცია;

ლაგრანჟის ინტერპოლაციის რეალიზაცია მატლაბში

$$P(x_k) = y_k, \quad k = 1, \dots, n.$$

$$P(x) = \sum_k \left(\prod_{j \neq k} \frac{x - x_j}{x_k - x_j} \right) y_k.$$

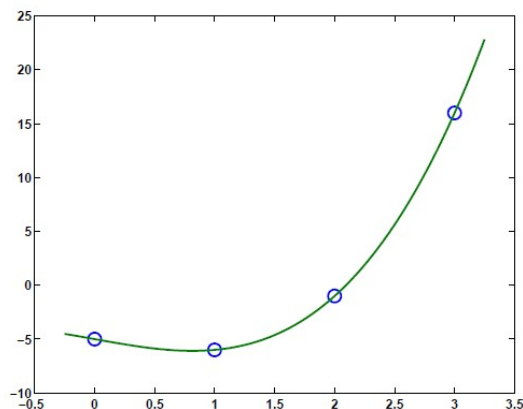
$$P(x) = c_1 x^{n-1} + c_2 x^{n-2} + \dots + c_{n-1} x + c_n,$$

ამოცანის ანალიზური ფორმა:

$$\begin{pmatrix} x_1^{n-1} & x_1^{n-2} & \dots & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \dots & x_2 & 1 \\ \dots & \dots & \dots & \dots & 1 \\ x_n^{n-1} & x_n^{n-2} & \dots & x_n & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

გამოყენება:

```
>>
>> v = polyinterp(x,y,u);
>> plot(x,y,'o',u,v,'-');
>>
```



polyinterp.m

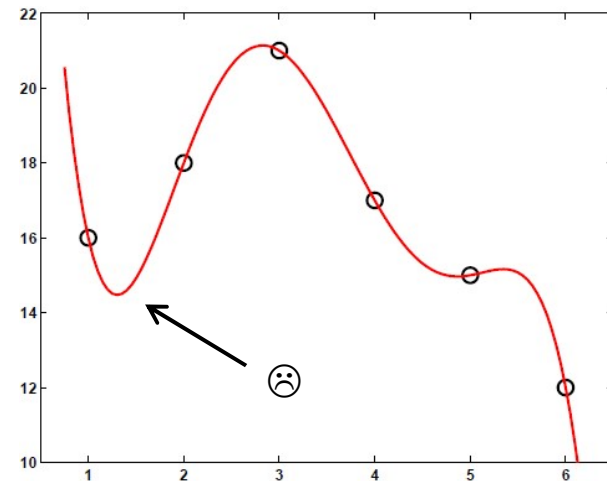
```
function v = polyinterp(x,y,u)

Nx = length(x);
Nu = length(u);
v = zeros(1,Nu);
for k = 1:Nx
    w = ones(1,Nu);
    for j = [1:k-1 k+1:Nx]
        w = (u-x(j))./(x(k)-x(j)).*w;
    end
    v = v + w*y(k);
end
```

უწყვეტი ინტერპოლაციის პრობლემები

მაგალითი:

```
>>  
>> x = 1:6;  
>> y = [16 18 21 17 15 12];  
>>  
>> u = .75:.05:6.25;  
>> v = polyinterp(x,y,u);  
>>
```



დავალება 5.2.

`polyinterp` ფუნქციის გამოყენებით მოახდინეთ $(-1, 1)$ ინტერვალში 5 თქვენს მიერ შერჩეული წერტილიდან ლაგრანჟის პოლინომიალური ინტერპოლაცია ვექტორში: $u = (-1:0.1:1)$;

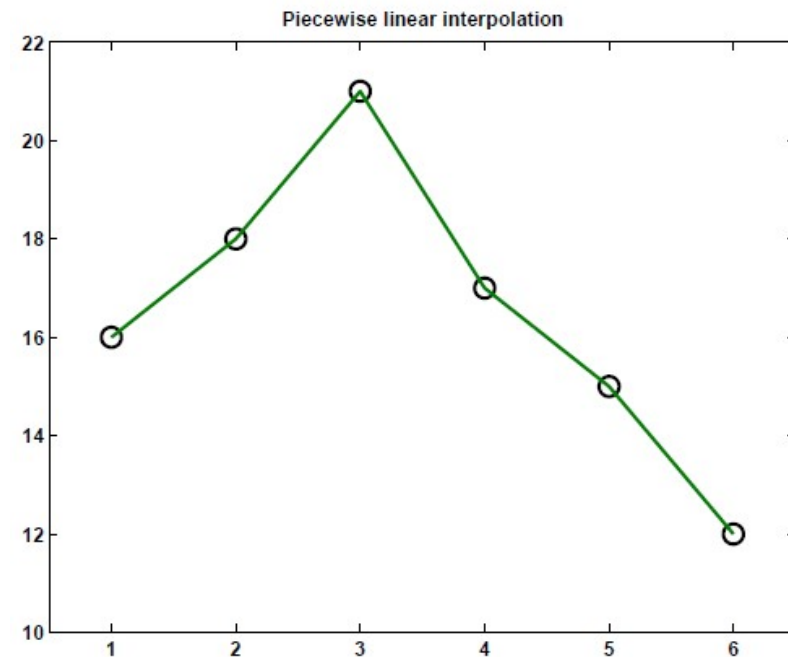
უბან-უბან წრფივი ინტერპოლაცია

ნაწყვეტებისაგან შედგენილი ფუნქცია - spline

ნაწყვეტ-ნაწყვეტ ანალიზური ფუნქცია, რომლითაც შესაძლებელია ინტერვალში ნებისმიერი წერტილის აღდგენა;

$$x_k \leq x < x_{k+1}$$

$$L(x) = y_k + (x - x_k) \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$$



უბან-უბან კუბური ერმიტული ინტერპოლაცია (pchip)

ინტერპოლანტი წერტილები: (x_k, y_k)

უბან-უბან კუბური ინტერპოლაციური ფუნქცია: $y = P(x)$

ინტერვალი ორ მოცემულ წერტილს შორის:

$$h_k = x_{k+1} - x_k.$$

ცვლადი ინტერვალს შიგნით:

$$s = x - x_k$$

ინტერპოლანტის დახრის კუთხის შეფასება:

$$d_k = P'(x_k).$$

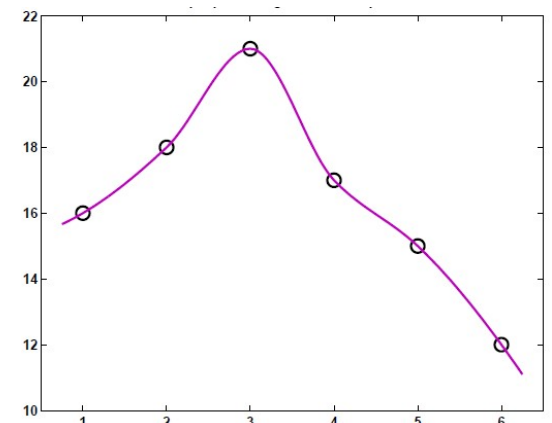
ორ ინტერპოლანტ წერტილს შორის ინტერვალში კუბური პოლინომიალი (spline):

$$P(x) = \frac{3hs^2 - 2s^3}{h^3}y_{k+1} + \frac{h^3 - 3hs^2 + 2s^3}{h^3}y_k + \frac{s^2(s-h)}{h^2}d_{k+1} + \frac{s(s-h)^2}{h^2}d_k.$$

ინტერპოლაციური ფუნქციის მნიშვნელობა ემთხვევა ინტერპოლანტ წერტილების მნიშვნელობებს:

$$P(x_k) = y_k, \quad P(x_{k+1}) = y_{k+1}, \\ P'(x_k) = d_k, \quad P'(x_{k+1}) = d_{k+1}.$$

არის კუბური პოლინომი (s -ის, ანუ x -ის მიმართ).



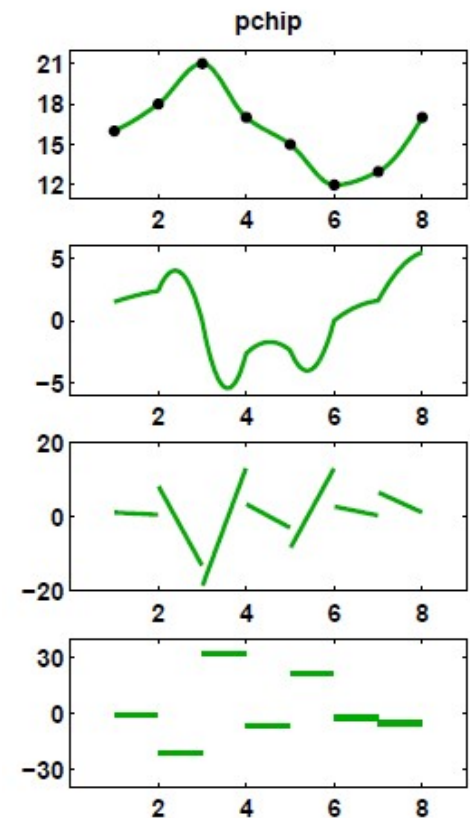
შესაძლებელია სხვადასხვა იმპლემენტაცია. მიდგომები: დახრის კუთხის შეფასება დისკრეტულ ინტერპოლანტში (d_k).

$$P(x) = y_k + sd_k + s^2c_k + s^3b_k,$$

$$c_k = \frac{3\delta_k - 2d_k - d_{k+1}}{h},$$

$$b_k = \frac{d_k - 2\delta_k + d_{k+1}}{h^2}.$$

pchip და მისი სამი ანალიზური წარმოდგენა.



დავალება 5.3.

შეადარეთ ნახაზზე მოყვანილ წერტილებზე უბან-უბან კუბური ინტერპოლაციის 1,2 და 3 რიგის ანალიზური წარმოდგენები კუბური ინტერპოლაციური ფუნქციის რიცხვით ცენტრალური წარმოდგენის მეთოდით აღებულ 3 წარმოდგენას.

pchiptx.m

```
function v = pchiptx(x,y,u)
%PCHIPTX piecewise cubic Hermite interpolation.

h = diff(x);
delta = diff(y)./h;
d = pchipslopes(h,delta);

% Piecewise polynomial coefficients
n = length(x);
c = (3*delta - 2*d(1:n-1) - d(2:n))./h;
b = (d(1:n-1) - 2*delta + d(2:n))./h.^2;

% Find subinterval indices k so that x(k) <= u < x(k+1)
k = ones(size(u));
for j = 2:n-1
    k(x(j) <= u) = j;
end

% Evaluate interpolant
s = u - x(k);
v = y(k) + s.*(d(k) + s.*(c(k) + s.*b(k)));
```

pchipslopes.m

```
function d = pchipslopes(h,delta)
% PCHIPSLOPES Slopes for shape-preserving Hermite cubic
% pchipslopes(h,delta) computes  $d(k) = P'(x(k))$ .

% Slopes at interior points
% delta = diff(y)./diff(x).
% d(k) = 0 if delta(k-1) and delta(k) have opposites
% signs or either is zero.
% d(k) = weighted harmonic mean of delta(k-1) and
% delta(k) if they have the same sign.

n = length(h)+1;
d = zeros(size(h));
k = find(sign(delta(1:n-2)).*sign(delta(2:n-1))>0)+1;
w1 = 2*h(k)+h(k-1);
w2 = h(k)+2*h(k-1);
d(k) = (w1+w2)./(w1./delta(k-1) + w2./delta(k));

% Slopes at endpoints
d(1) = pchipend(h(1),h(2),delta(1),delta(2));
d(n) = pchipend(h(n-1),h(n-2),delta(n-1),delta(n-2));
```

pchipend.m

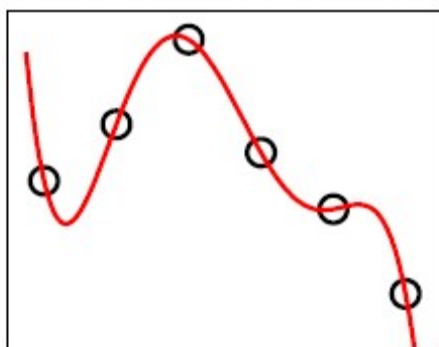
```
function d = pchipend(h1,h2,del1,del2)
% Noncentered, shape-preserving, three-point formula.

d = ((2*h1+h2)*del1 - h1*del2)/(h1+h2);

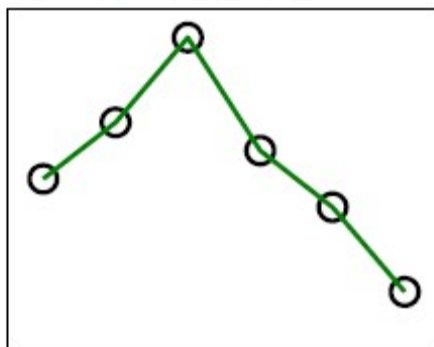
if sign(d) ~= sign(del1)
    d = 0;
elseif (sign(del1)~=sign(del2)) & (abs(d)>abs(3*del1))
    d = 3*del1;
end
```

6 წერტილიანი ინტერპოლანტის ინტერპოლაციის შედეგები

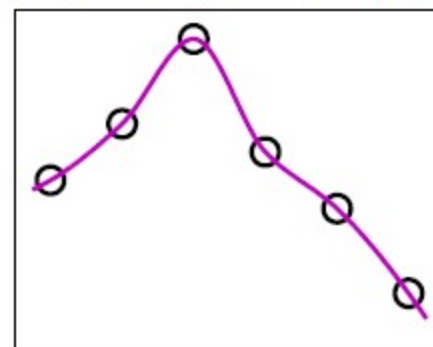
Full degree polynomial interpolation



Piecewise linear interpolation



Shape-preserving Hermite interpolation



უბან-უბან კუბური ერმიტული ინტერპოლაცია:

+ ნაკლები „ამოვარდნები“

+ მონოტონური მონაცემები

- ვიზუალურად არც ისე გლუვი როგორც ლაგრანჟის ინტერპოლაცია

დავალება 5.4.

ააგეთ თქვენი ხელის ფუნქცია კუბური ინტერპოლაციის გამოყენებით. გამოიყენეთ მაქსიმუმ 30-40 საინტერპოლაციო წერტილი. მიღებული მრუდი უნდა შეიცავდეს 1000 წერტილს.

გამოიყენეთ პარამეტრული ინტერპოლაცია: $x=x(t)$, $y=y(t)$,

Matlab-ში წერტილების ინტერაქტიული შეყვანის კოდი:

```
figure('position',get(0,'screensize'));  
axes('position',[0 0 1 1]);  
[x,y] = ginput;
```

