Altai State University

Discipline "Numerical Methods in Physics"

Lecture 4. Spline interpolation

Shevchuk Evgeniya Petrovna,
Senior Lecturer, Department of General and Experimental Physics,
Institute of Digital Technologies, Electronics and Physics

Methods for setting functions

1. Analytical
2. Tabular
3. Graphic

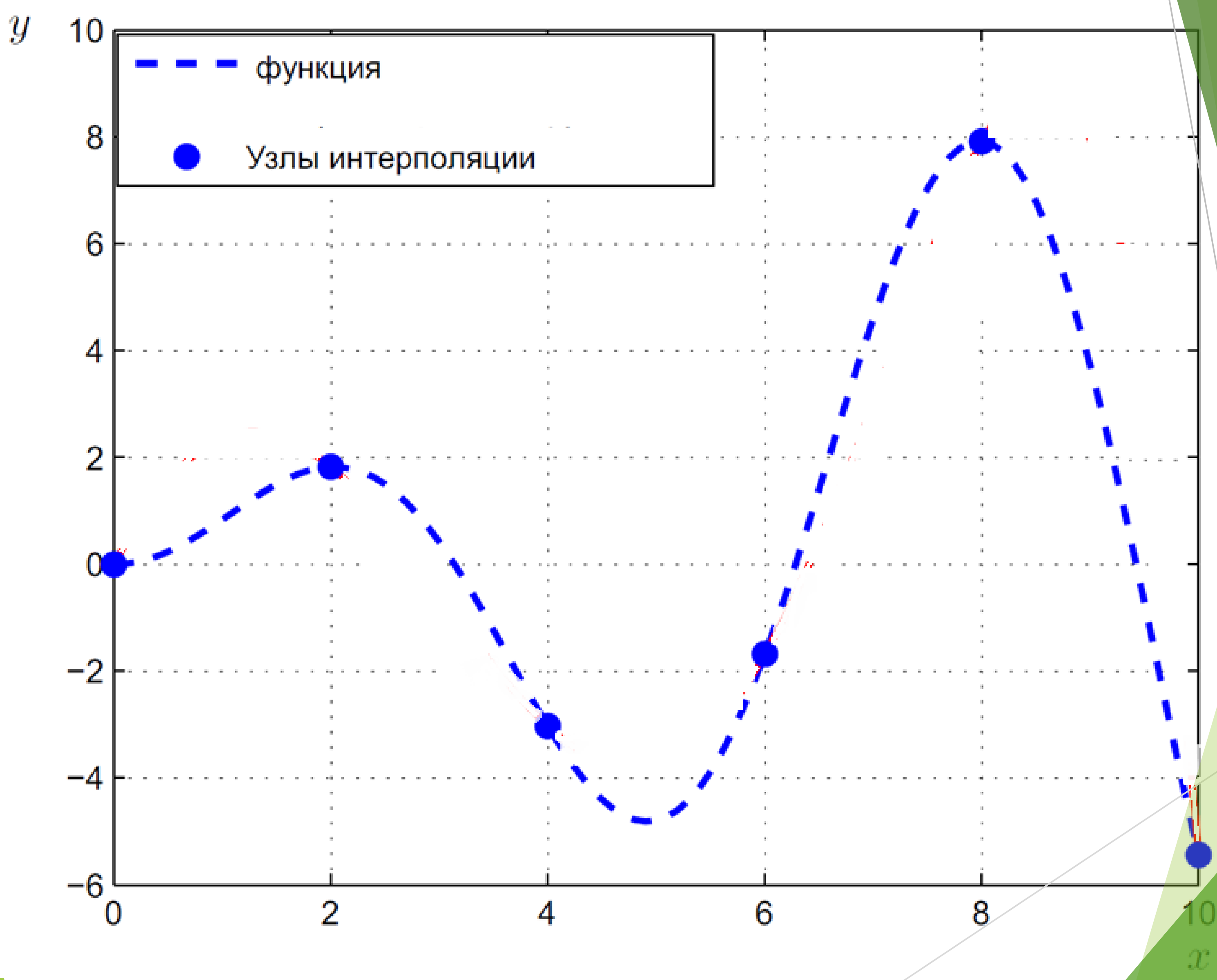
Buckley-Leverett function (fraction of water in the stream)

S - water saturation

η - the ratio of the viscosities of water and oil

$$F(S) = \frac{S^2}{S^2 + \eta(1 - S)^2}$$

Interpolation problem



Interpolation, interpolation is a method of finding intermediate values of a quantity from an available set of known values.
The values of the function y = f (x) are known only at these points

$$y_i = f(x_i),\ i=1,\ldots N$$

The interpolation problem is to find such a function F from

a given class of functions that $F(x_i) = y_i,\ i=1,\ldots N$

Points x_i are called interpolation nodes,

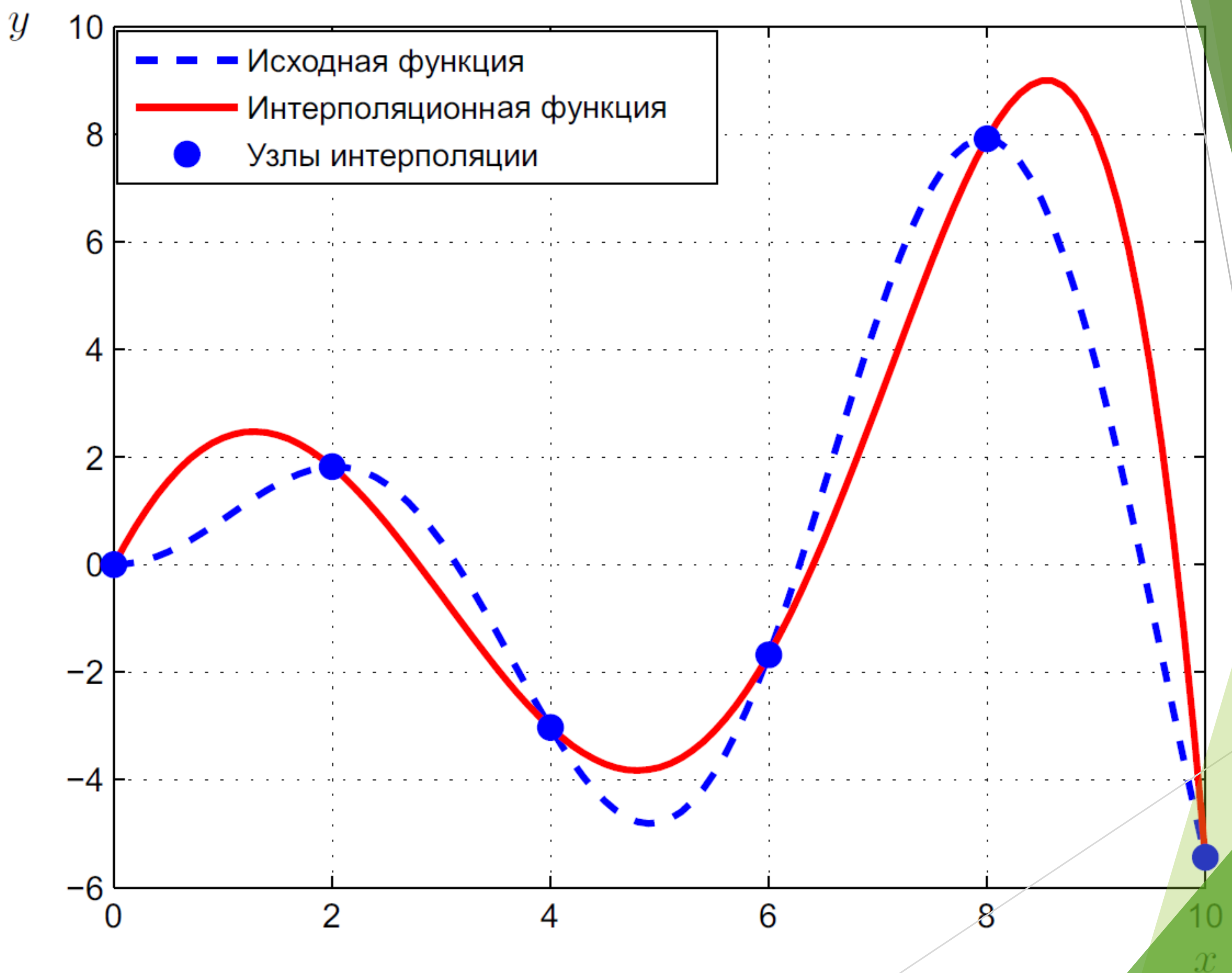
and their totality - by an interpolation grid.

Pairs x_i, y_i called data points or base points.

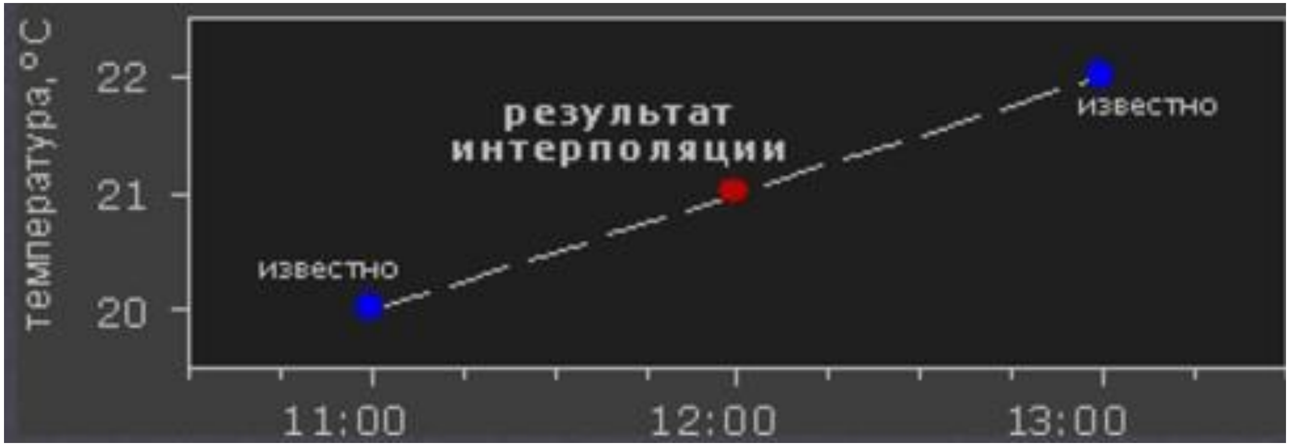
Difference between "adjacent" values $\Delta_i = x_i - x_{i-1}$ — step of the interpolation grid. The step can be both variable and constant.

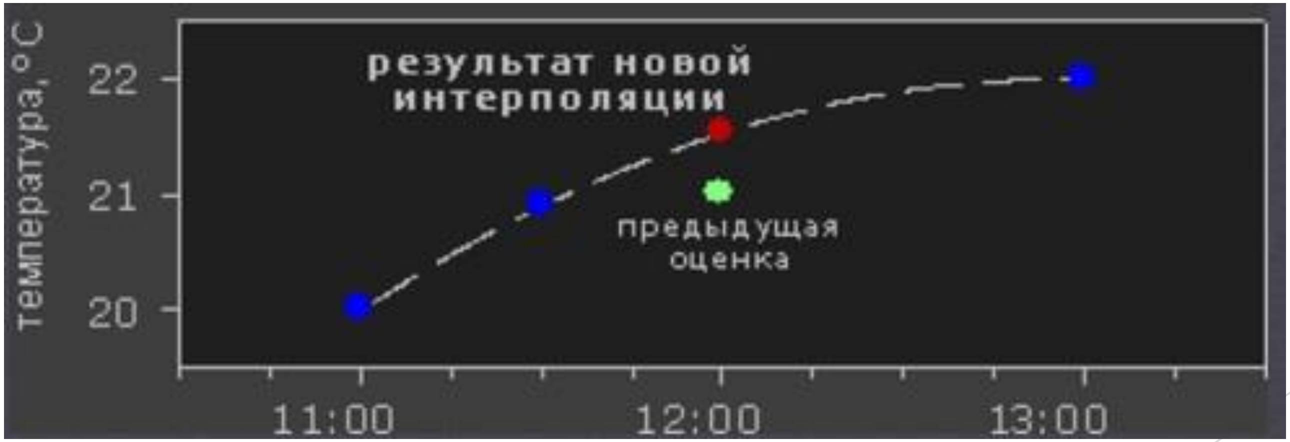
Function $F(x)$ called interpolation function or interpolant.

Interpolation function



Interpolation example

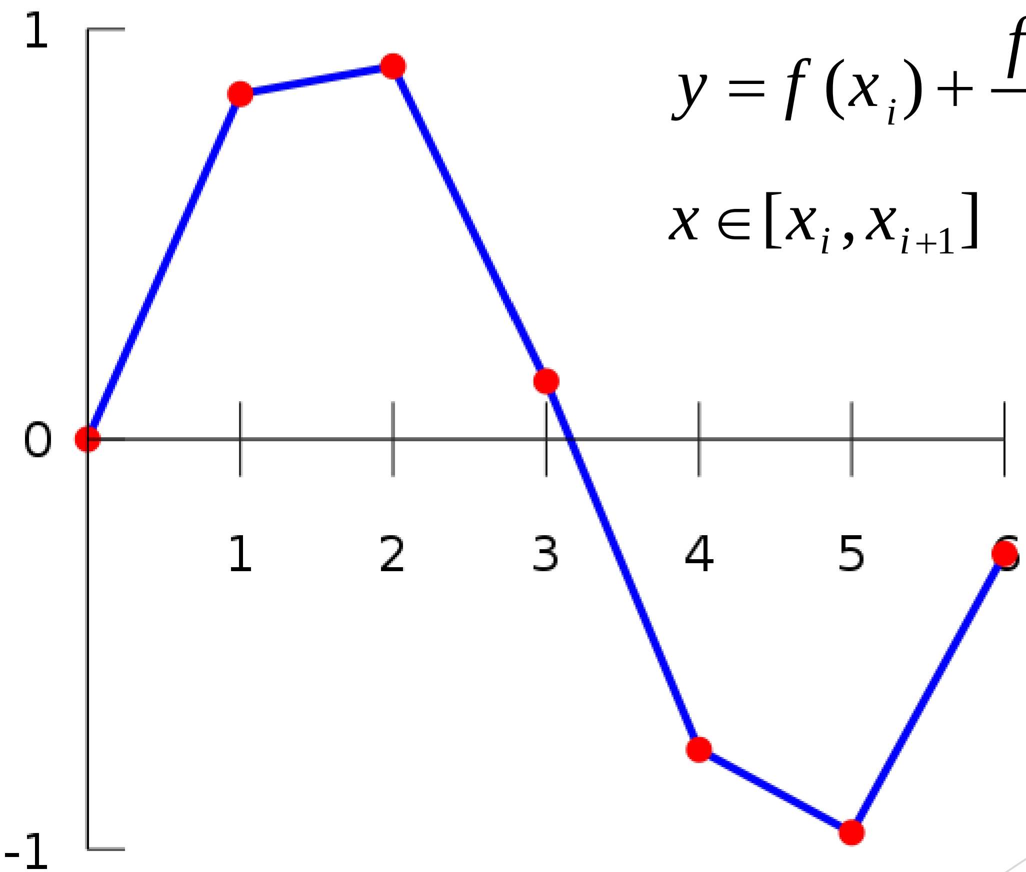




Piecewise linear interpolation

Equation of a straight line passing through $(x_i, f(x_i)),(x_{i+1}, f(x_{i+1}))$

$$\frac{y - f(x_i)}{f(x_{i+1}) - f(x_i)} = \frac{x - x_i}{x_{i+1} - x_i}$$

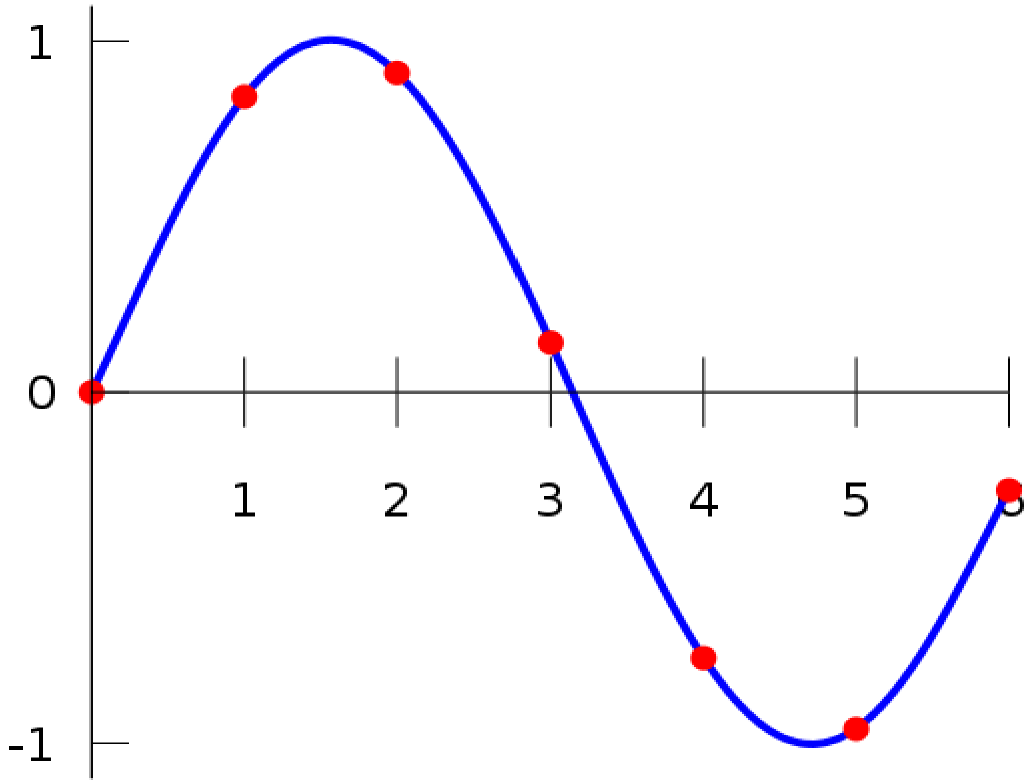


$$y = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i)$$

$$x \in [x_i, x_{i+1}]$$

Polynomial interpolation

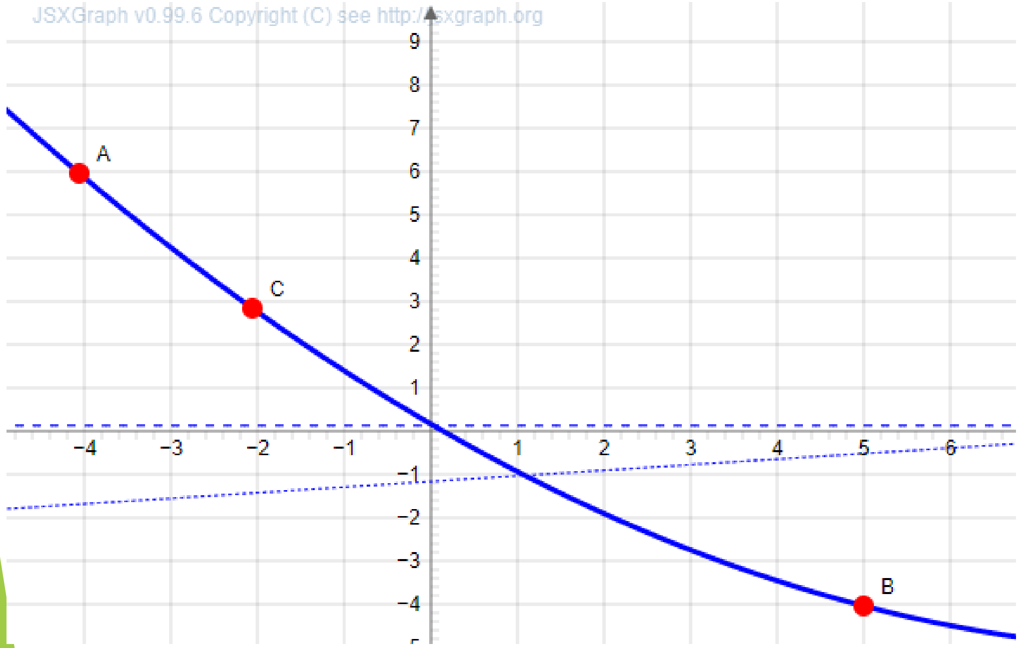
As an interpolating function for n points, we choose polynomial of degree at most n-1



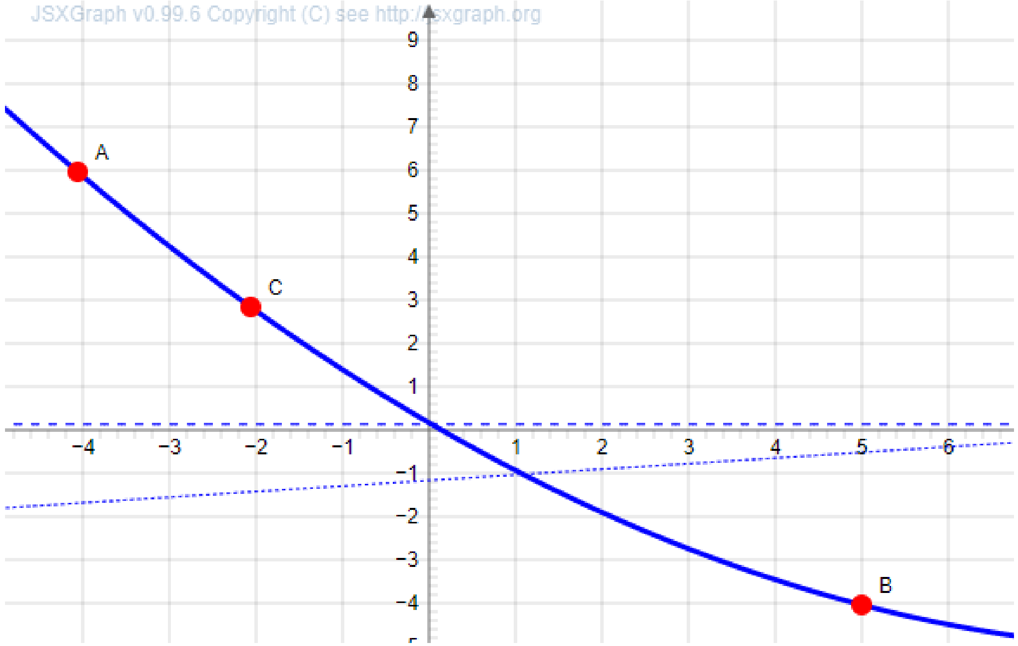
For n points

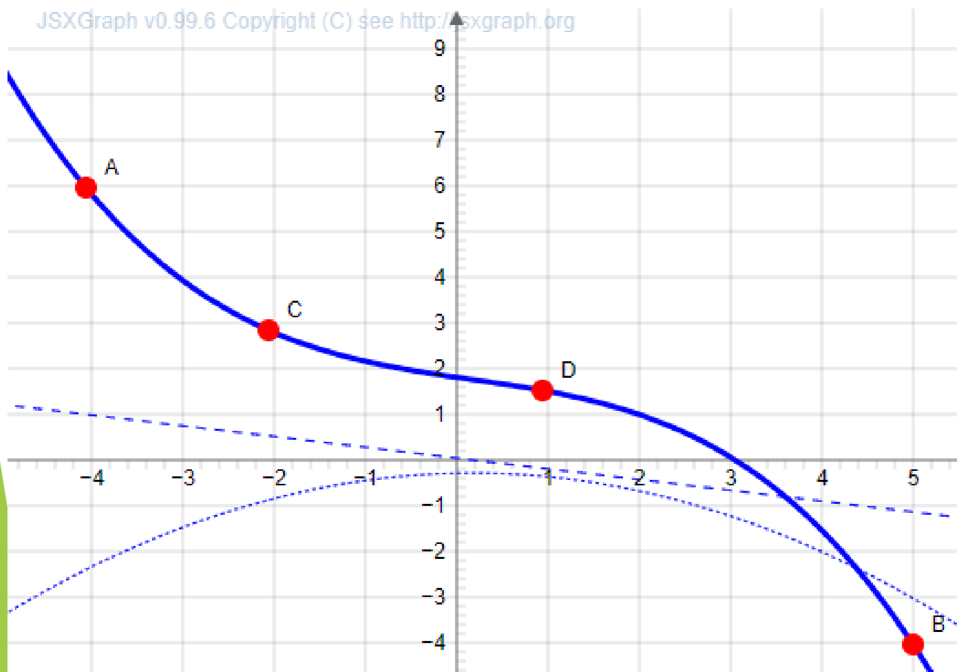
http://jsxgraph.uni-bayreuth.de/wiki/index.php/Lagrange_interpolation

Polynomial interpolation

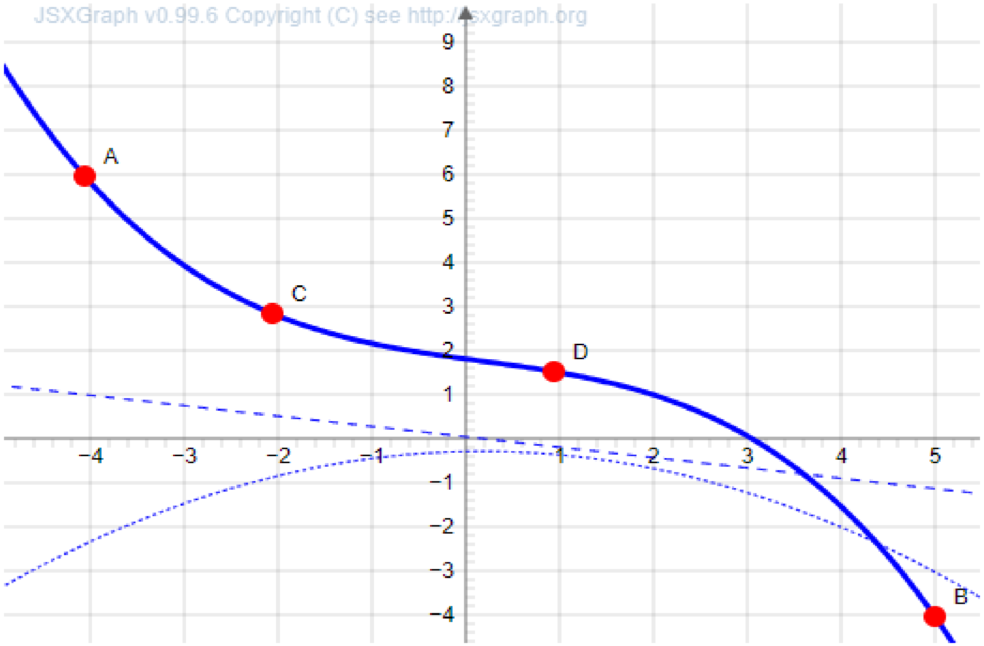


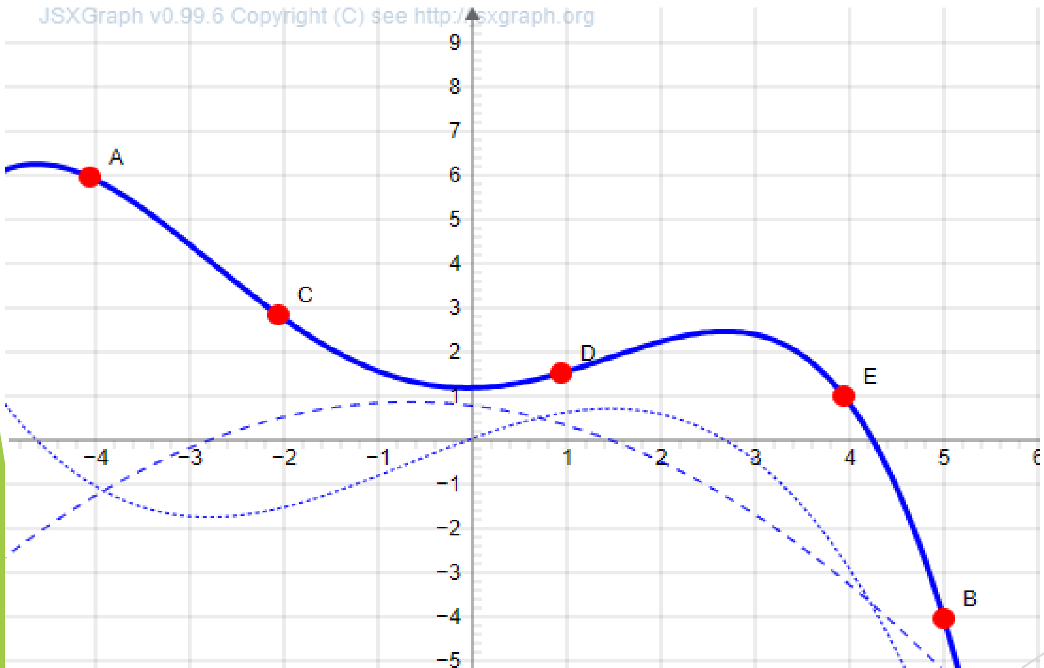
Polynomial interpolation





Polynomial interpolation





Polynomial interpolation

Through n points (x_i, y_i) one can hold a single polynomial of degree at most n-1. Different forms of notation of the polynomial

$$y(x) = P_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 =$$

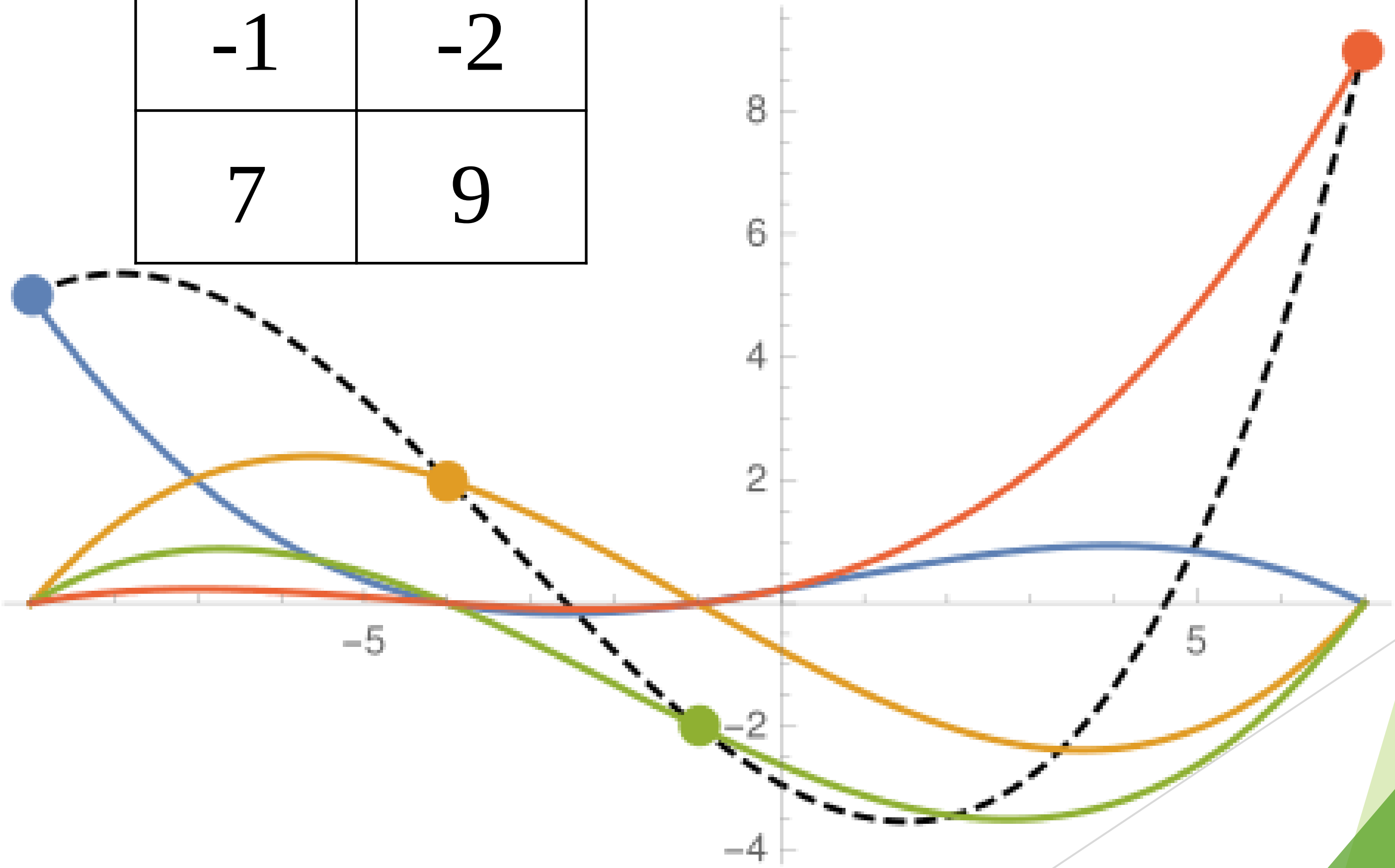
$$= x \cdot (x \cdot (x \cdot a_3 + a_2) + a_1) + a_0 =$$

$$= a_3 \cdot (x - x_1) \cdot (x - x_2) \cdot (x - x_3)$$

What function will be obtained by adding 2 polynomials of degree 3?

x	y
-9	5
-4	2
-1	-2
7	9

$$L_3(x) = \sum_{j=1}^{4} y_j p_j(x)$$



Lagrange interpolation polynomial

$$L_{n-1}(x) = \sum_{j=1}^{n} y_j \prod_{k=1,\ k\neq j}^{n} \frac{(x - x_k)}{(x_j - x_k)}$$

Another form of notation

$$L_{n-1}(x) = \sum_{j=1}^{n} y_j \, p_j(x)$$

$$p_j(x) = \prod_{k=1, k\neq j}^{n} \frac{(x - x_k)}{(x_j - x_k)}$$

Lagrange interpolation polynomial for n = 4

$$L_3(x) = \sum_{j=1}^{4} y_j p_j(x)$$

$$p_1(x) = \frac{(x - x_2)\cdot(x - x_3)\cdot(x - x_4)}{(x_1 - x_2)\cdot(x_1 - x_3)\cdot(x_1 - x_4)}$$

$$p_2(x) = \frac{(x - x_1)\cdot(x - x_3)\cdot(x - x_4)}{(x_2 - x_1)\cdot(x_2 - x_3)\cdot(x_2 - x_4)}$$

$$p_3(x) = \frac{(x - x_1)\cdot(x - x_2)\cdot(x - x_4)}{(x_3 - x_1)\cdot(x_3 - x_2)\cdot(x_3 - x_4)}$$

$$p_4(x) = \frac{(x - x_1)\cdot(x - x_2)\cdot(x - x_3)}{(x_4 - x_1)\cdot(x_4 - x_2)\cdot(x_4 - x_3)}$$

x	**y**
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4

x	y
-9	5
-4	2
-1	-2
7	9

$$L_3(x) = \sum_{j=1}^{4} y_j p_j(x)$$

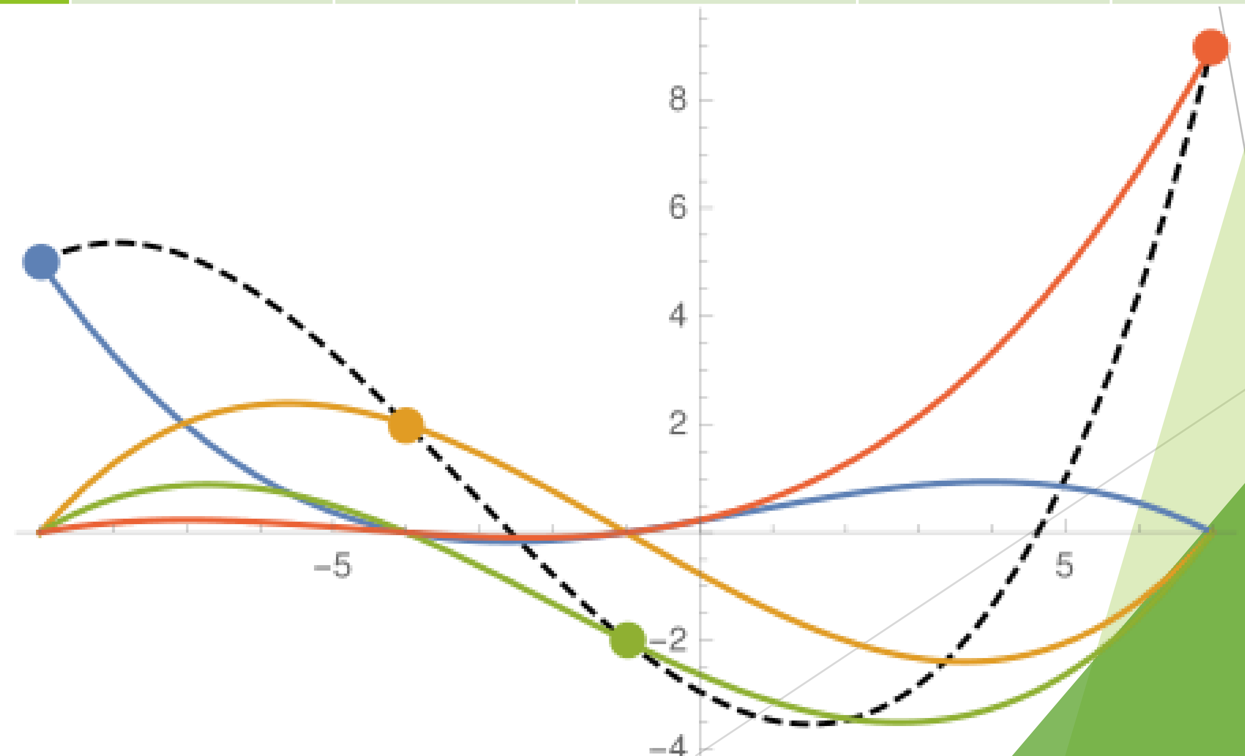
$$p_1(x) = \frac{\square(x-(-4))\cdot(x-(-1))\cdot(x-7)}{(-9-(-4))\cdot(-9-(-1))\cdot(-9-7)}$$

$$p_2(x) = \frac{\square(x-(-9))\cdot(x-(-1))\cdot(x-7)}{(-4-(-9))\cdot(-4-(-1))\cdot(-4-7)}$$

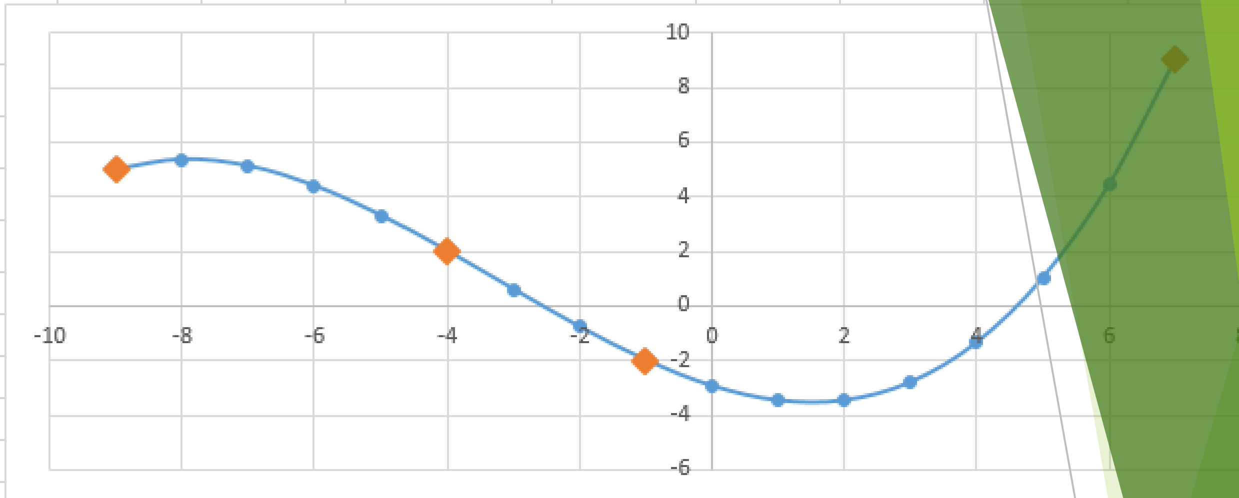
$$p_3(x) = \frac{(x-\ldots)(\ldots)}{(\ldots-\ldots)(\ldots)}$$

$$p_4(x) = \frac{(x-\ldots)(\ldots)}{(\ldots-\ldots)(\ldots)}$$

x	p1	p2	p3	p4	y
-9	1	0	0	0	5
-4	0	1	0	0	2
0	=-28/-640	=-63/165	=-252/-192	=36/1408	-2,94
-1	0	0	1	0	-2
7	0	0	0	1	9



x	y
-9	5
-4	2
-1	-2
7	9

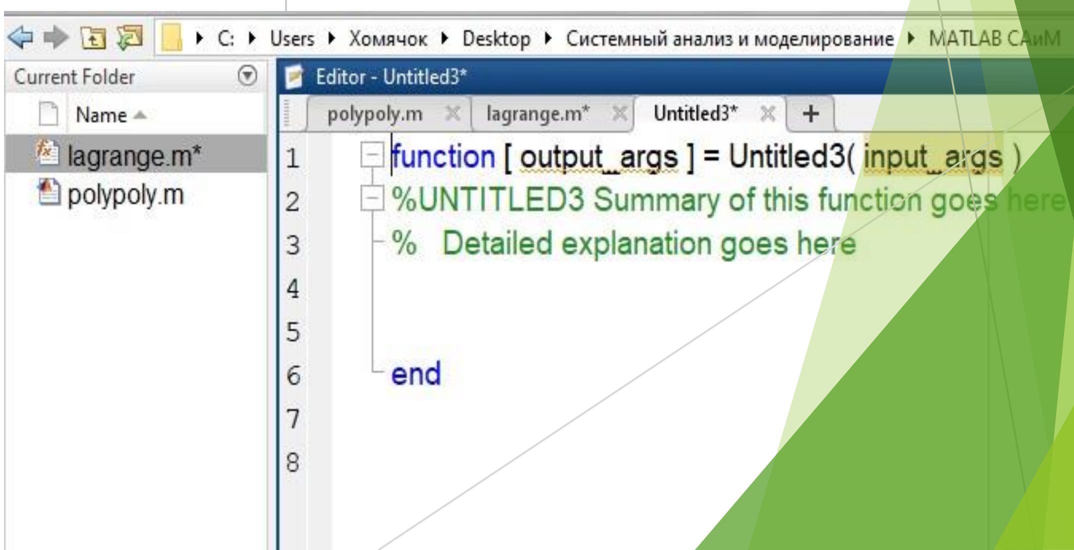


x	p1	p2	p3	p4	Y Lagrange
-9	1	0	0	0	5,00
-8	0,65625	0,636364	-0,3125	0,019886	5,36
-7	0,39375	1,018182	-0,4375	0,025568	5,11
-6	0,203125	1,181818	-0,40625	0,021307	4,38
-5	0,075	1,163636	-0,25	0,011364	3,30
-4	0	1	0	0	2,00
-3	-0,03125	0,727273	0,3125	-0,00852	0,60
-2	-0,028125	0,381818	0,65625	-0,00994	-0,78
-1	0	0	1	0	-2,00
0	0,04375	-0,38182	1,3125	0,025568	-2,94
1	0,09375	-0,72727	1,5625	0,071023	-3,47
2	0,140625	-1	1,71875	0,140625	-3,47
3	0,175	-1,16364	1,75	0,238636	-2,80
4	0,1875	-1,18182	1,625	0,369318	-1,35
5	0,16875	-1,01818	1,3125	0,536932	1,01
6	0,109375	-0,63636	0,78125	0,745739	4,42
7	0	0	0	1	9,00

Editor - C:\Users\Хомячок\Desktop\Системный анализ и моделирование\MATLAB САиМ\lagrange.m*

polypoly.m | lagrange.m*

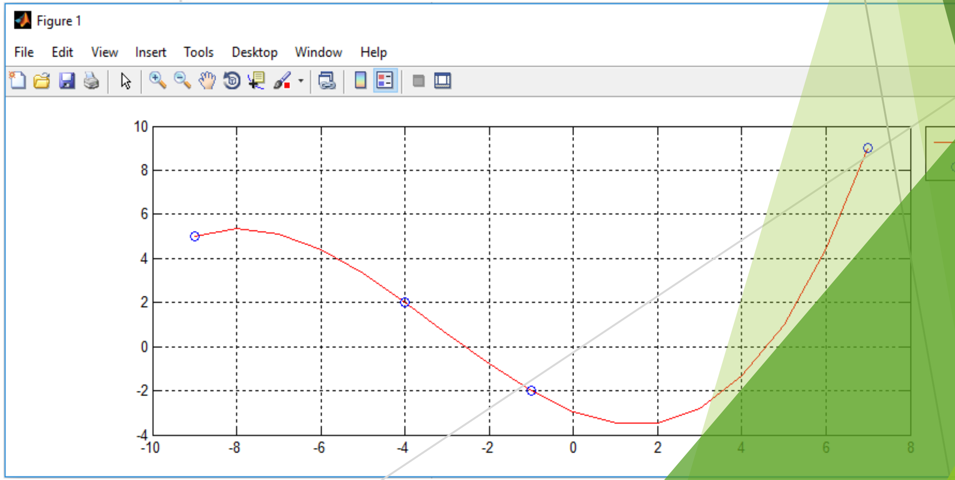
```
function yy=lagrange(x,y,xx)
% вычисление интерполяционного полинома в форме Лагранжа
% x - массив координат узлов
% y - массив значений интерполируемой функции
% xx - массив значений аргумента, для которых надо вычислить значения полинома
% yy - массив значений полинома в точках xx

% узнаем число узлов интерполяции (N=n+1)
N=length(x);
% создаем нулевой массив значений интерполяционного полинома
yy=zeros(size(xx));
% в цикле считаем сумму по узлам
for k=1:N
   % вычисляем произведения
   t=ones(size(xx));
   for j=[1:k-1, k+1:N]
      t=t.*(xx-x(j))/(x(k)-x(j));
   end
   % накапливаем сумму
   yy = yy + y(k)*t;
end
```



Editor - C:\Users\Хомячок\Desktop\Системный анализ и моделирование\MATLAB САиМ\polypoly.m

polypoly.m | lagrange.m

```
clear all
clc
close all
% задание узлов интерполяции
x=[-9 -4 -1 7];
y=[5 2 -2 9];
% задание точек, в которых требуется найти значения интерполяционного полинома
%xx=linspace(-9,7,20)
xx=-9:7
% нахождение значений интерполяционного полинома
yy=lagrange(x,y,xx);
% построение графиков
figure('Color','w')
% вывод графика полинома
plot(xx,yy,'r')
hold on
grid on
% вывод узлов интерполяции
plot(x,y,'bo')
% размещение легенды
legend('{\itL_n}({\itx})','nodes',-1)
```



Newton's method
(finite difference method)

For functions y = f (x), given tabularly with a constant step, the differences between the values of the function at neighboring interpolation nodes are called finite differences of the first order

$$\Delta y_i = \Delta f(x) = f(x+h) - f(x) = y_{i+1} - y_i$$

From finite differences of the first order, second order finite differences

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i = f(x+2h) - 2f(x+h) + f(x)$$

Newton's method
(finite difference method)

Finite Difference Table for n = 4

$\Delta^0 y_i$	$\Delta^1 y_i$	$\Delta^2 y_i$	$\Delta^3 y_i$
y_1	$y_2 - y_1$	$\Delta^1 y_2 - \Delta^1 y_1$	$\Delta^2 y_2 - \Delta^2 y_1$
y_2	$y_3 - y_2$	$\Delta^1 y_3 - \Delta^1 y_2$	
y_3	$y_4 - y_3$		
y_4			

Newton's method (finite difference method)

Finite Difference Table for n = 4

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
150	1,40			0,60
200	1,85			
250	1,75			
300	1,60			

Fill the table

Метод Ньютона

Newton's first interpolation formula for equidistant points. It is used if you need to calculate the value of a function near a point x_1

$$P_{n-1}(x) = y_1 + \frac{\Delta^1 y_1}{h}(x - x_1) + \frac{\Delta^2 y_1}{2!h^2}(x - x_1)(x - x_2) + \ldots +$$

$$+ \frac{\Delta^{n-1} y_1}{(n-1)!h^{n-1}}(x - x_1)(x - x_2)\ldots(x - x_{n-1})$$

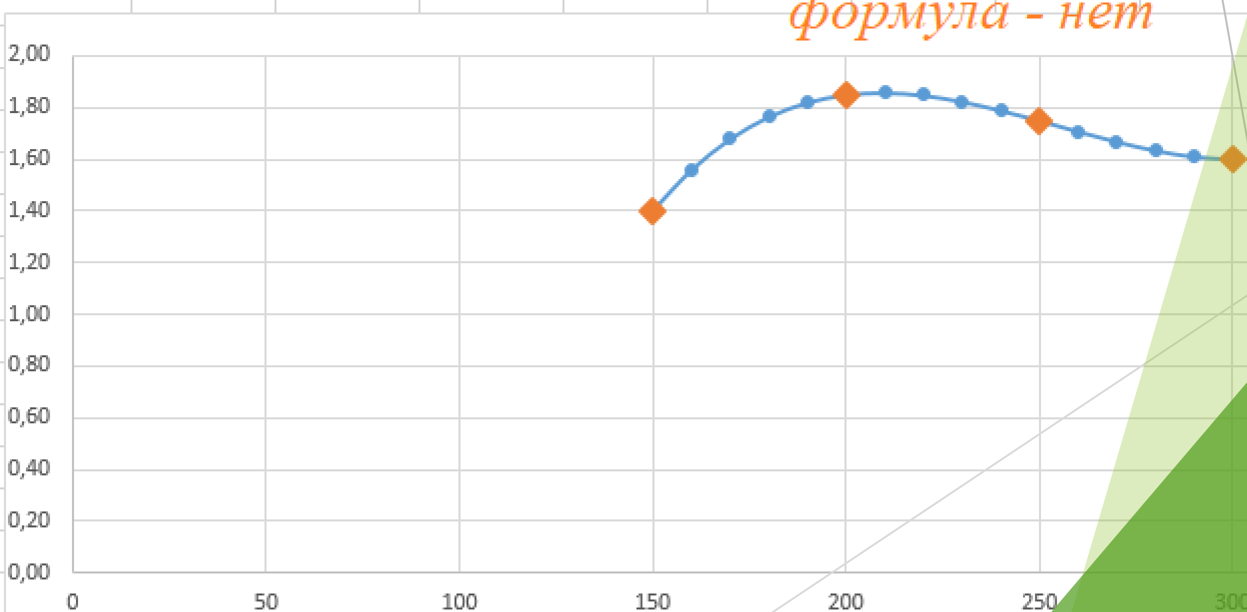
x	y	Δy	Δ^2y	Δ^3y
150	1,40	0,45	-0,55	0,50
200	1,85	-0,10	-0,05	
250	1,75	-0,15		
300	1,60			

Write a formula to calculate the value of a function at the point x=220

Newton's first interpolation formula for equidistant points.

	A	B	C	D	E
1	x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
2	150	1,40	0,45	-0,55	0,50
3	200	1,85	-0,10	-0,05	
4	250	1,75	-0,15		
5	300	1,60			
6					
7					
8	x	Y1H			
9	150	1,40			
10	160	=$B$2+$C$2/50*(A10-$A$2)+$D$2/(2*50^2)*(A10-$A$2)*(A10-$A$3)+:			
11	170	1,68			
12	180	1,76			
13	190	1,82			
14	200	1,85			
15	210	1,86			
16	220	1,85			
17	230	1,82			
18	240	1,79			
19	250	1,75			
20	260	1,71			
21	270	1,67			
22	280	1,63			
23	290	1,61			
24	300	1,60			
25					

подсказки закончились, формула - нет



Метод Ньютона

Newton's second interpolation formula for equidistant points. It is used if you need to calculate the value of a function near a point $\boldsymbol{x_n}$

$$P_{n-1}(x) = y_n + \frac{\Delta^1 y_{n-1}}{h}(x - x_n) + \frac{\Delta^2 y_{n-2}}{2!h^2}(x - x_n)(x - x_{n-1}) + \ldots +$$

$$+ \frac{\Delta^{n-1} y_1}{(n-1)!h^{n-1}}(x - x_n)(x - x_{n-1})\ldots(x - x_2)$$

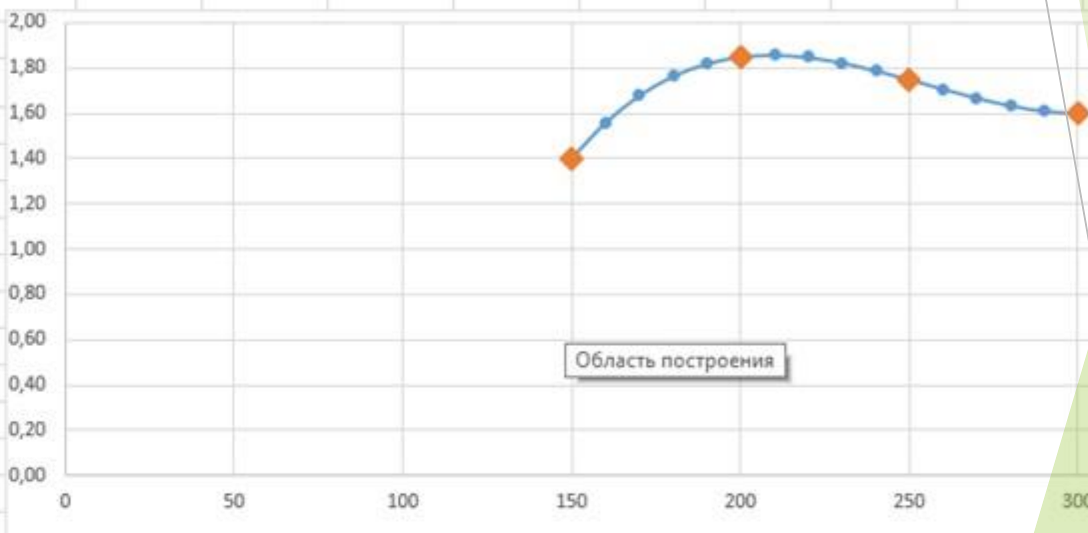
	A	B	C	D	E
1	**x**	**y**	Δy	Δ^2y	Δ^3y
2	150	1,40	0,45	-0,55	0,50
3	200	1,85	-0,10	-0,05	
4	250	1,75	-0,15		
5	300	1,60			

Newton's second interpolation formula for equidistant points.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
150	1,40	0,45	-0,55	0,50
200	1,85	-0,10	-0,05	
250	1,75	-0,15		
300	1,60			

x	Y1H	Y2H
150	1,40	1,40
160	1,56	1,56
170	1,68	1,68
180	1,76	1,76
190	1,82	1,82
200	1,85	1,85
210	1,86	1,86
220	1,85	1,85
230	1,82	1,82
240	1,79	1,79
250	1,75	1,75
260	1,71	1,71
270	1,67	1,67
280	1,63	1,63
290	1,61	=$B$5+$C$4/50*(A23-$A$5)+$D$3/(2*50^2)*(A23-$A$5)*(A23-$A$4)+
300	1,60	1,60

продолжите сами



Newton's method

An example of the need to use the first or second Newton formula

x	y	$\Delta^1 y$	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1.415	0.88855	0.001048	-1E-05	2E-06	-4E-06
1.420	0.8896	0.001038	-8E-06	-2E-06	3E-06
1.425	0.89064	0.001030	-1E-05	1E-06	-1E-06
1.430	0.89167	0.001020	-9E-06	1.11E-16	-3.3E-16
1.435	0.89269	0.001011	-9E-06	-2.2E-16	4.44E-16
1.440	0.8937	0.001002	-9E-06	2.22E-16	1E-06
1.445	0.8947	0.000993	-9E-06	1E-06	-3E-06
1.450	0.89569	0.000984	-8E-06	-2E-06	
1.455	0.89668	0.000976	-1E-05		
1.460	0.89765	0.000966			
1.465	0.89862				

Newton's method

Newton's formula for randomly located points $P_{n-1}(x) = y(x_1) + (x - x_1)\cdot[x_1, x_2] + (x - x_1)(x - x_2)\cdot[x_1, x_2, x_3] + \ldots$

$$+(x - x_1)(x - x_2)\cdot\ldots\cdot(x - x_{n-1})\cdot[x_1, x_2, \ldots, x_n]$$

$[x_1, x_2], [x_1, x_2, x_3], \ldots, [x_1, x_2, \ldots, x_n]$ separated differences

$$[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$ separated 1st order differences

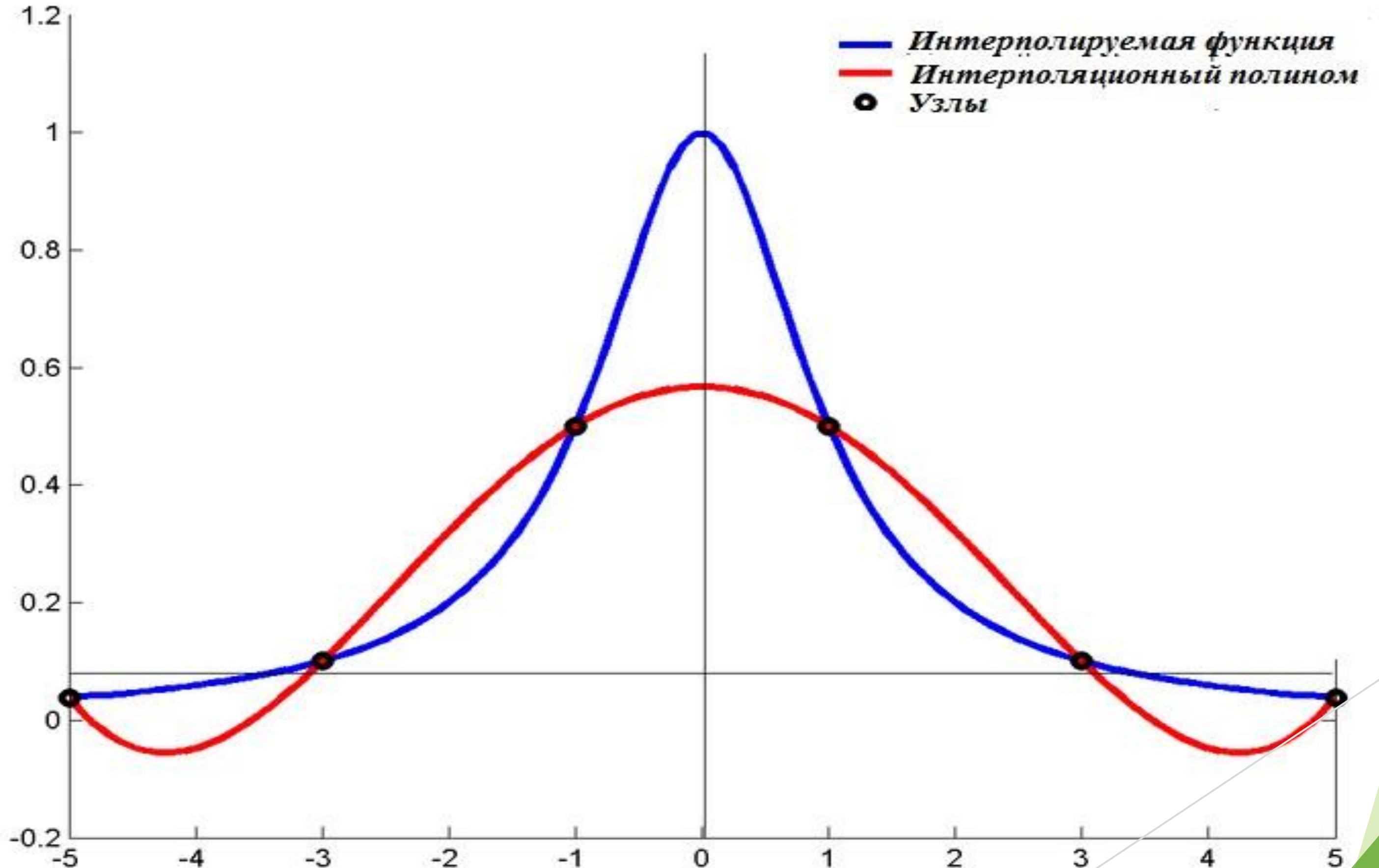
$$[x_i, x_{i+1}, x_{i+2}] = \frac{[x_{i+1}, x_{i+2}] - [x_i, x_{i+1}]}{x_{i+2} - x_i}$$ separated 2st order differences

$$[x_i, x_{i+1}, \ldots, x_{k+i}] = \frac{[x_{i+1}, x_{i+2}, \ldots, x_{k+i}] - [x_i, \ldots, x_{k+i-1}]}{x_{k+i} - x_i}$$

separated kth order differences

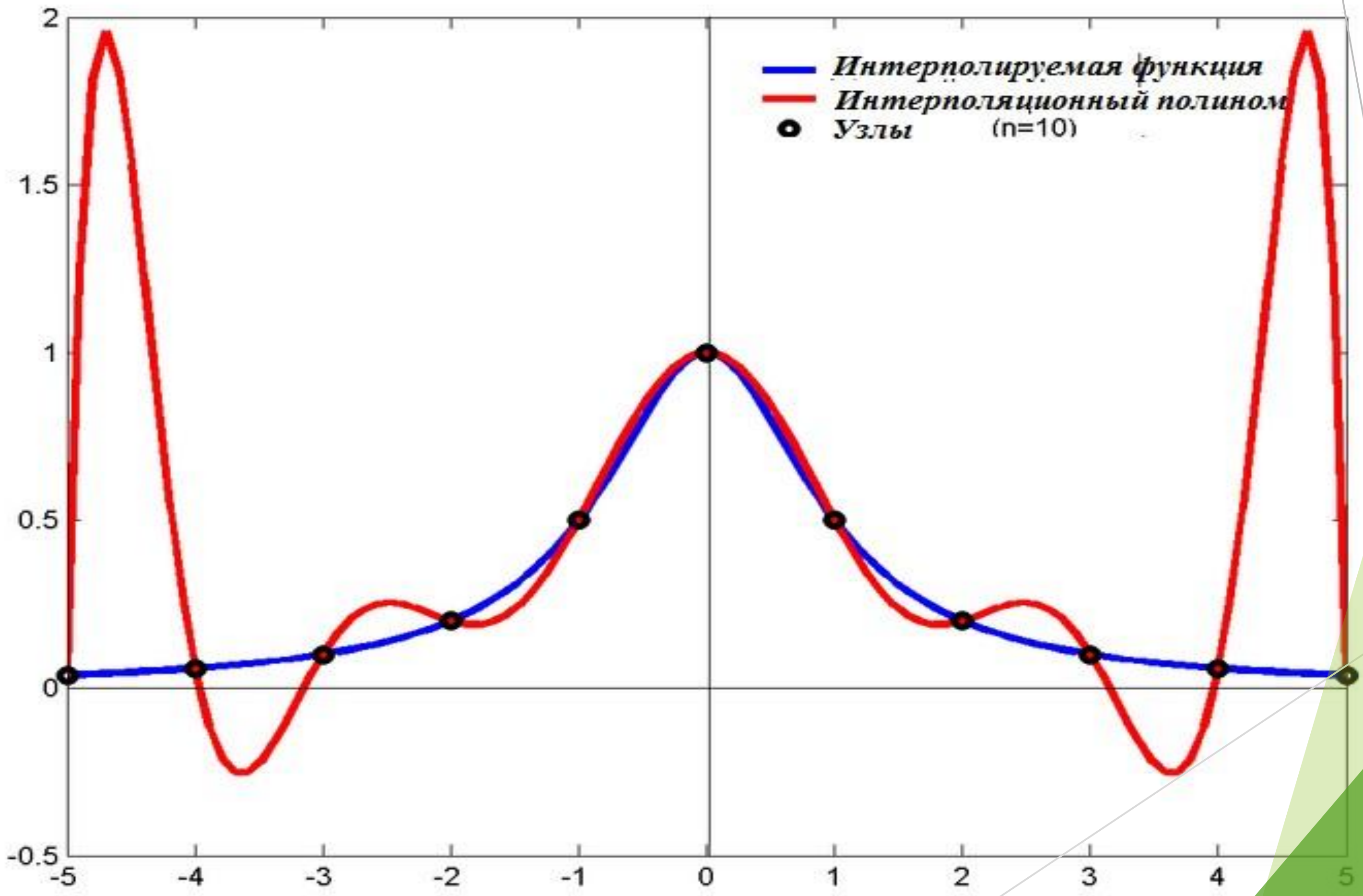
Polynomial interpolation problems

n=6 The interpolating function does not accurately approximate the original one. What happens if you increase the number of interpolation nodes?



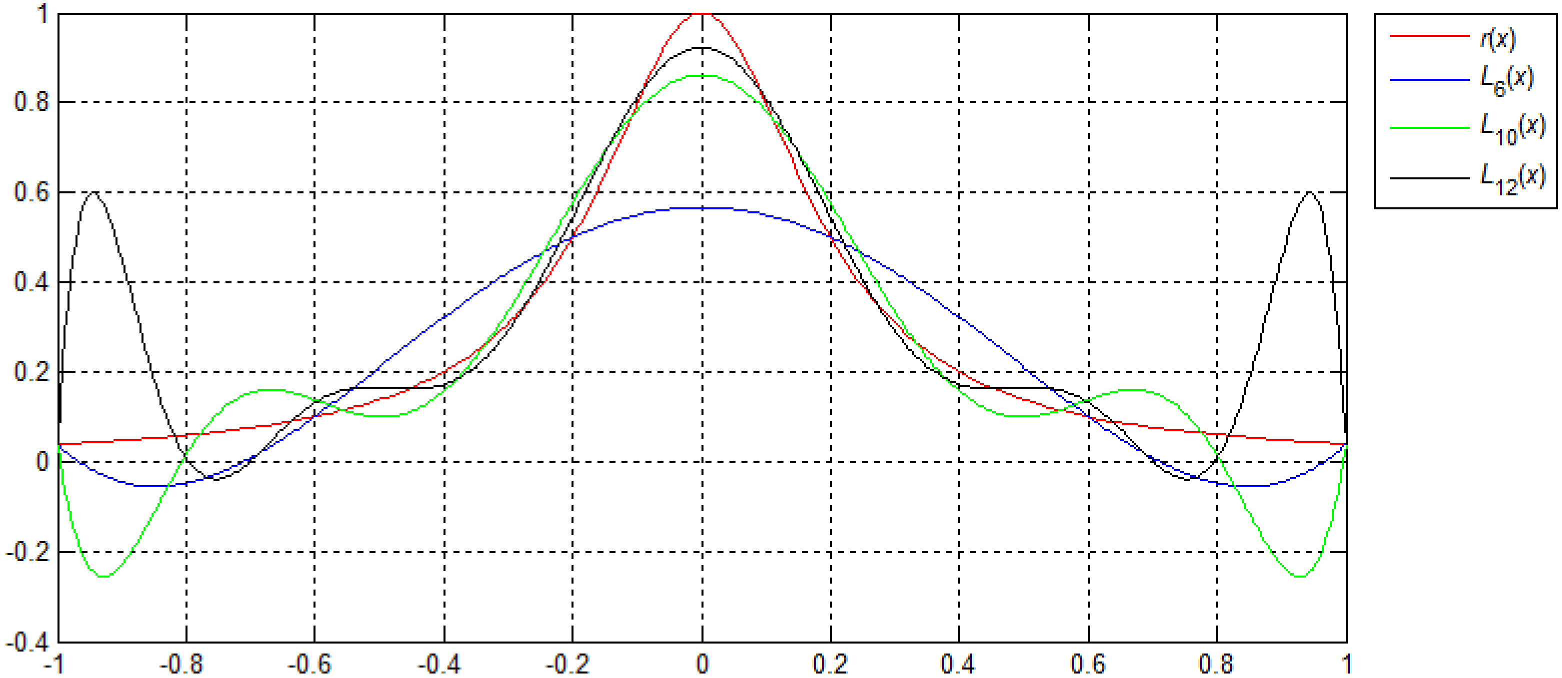
Polynomial interpolation problems

Possible behavior of the interpolation function with an increase in the number of interpolation nodes n=10



Polynomial interpolation problems

For some functions, for example $y(x) = \frac{1}{1+25x^2}$, observed a phenomenon called the Runge phenomenon, when with an increase in the number of nodes, the error tends to infinity. Moreover, even for a relatively small number of nodes, the error turns out to be unacceptably large

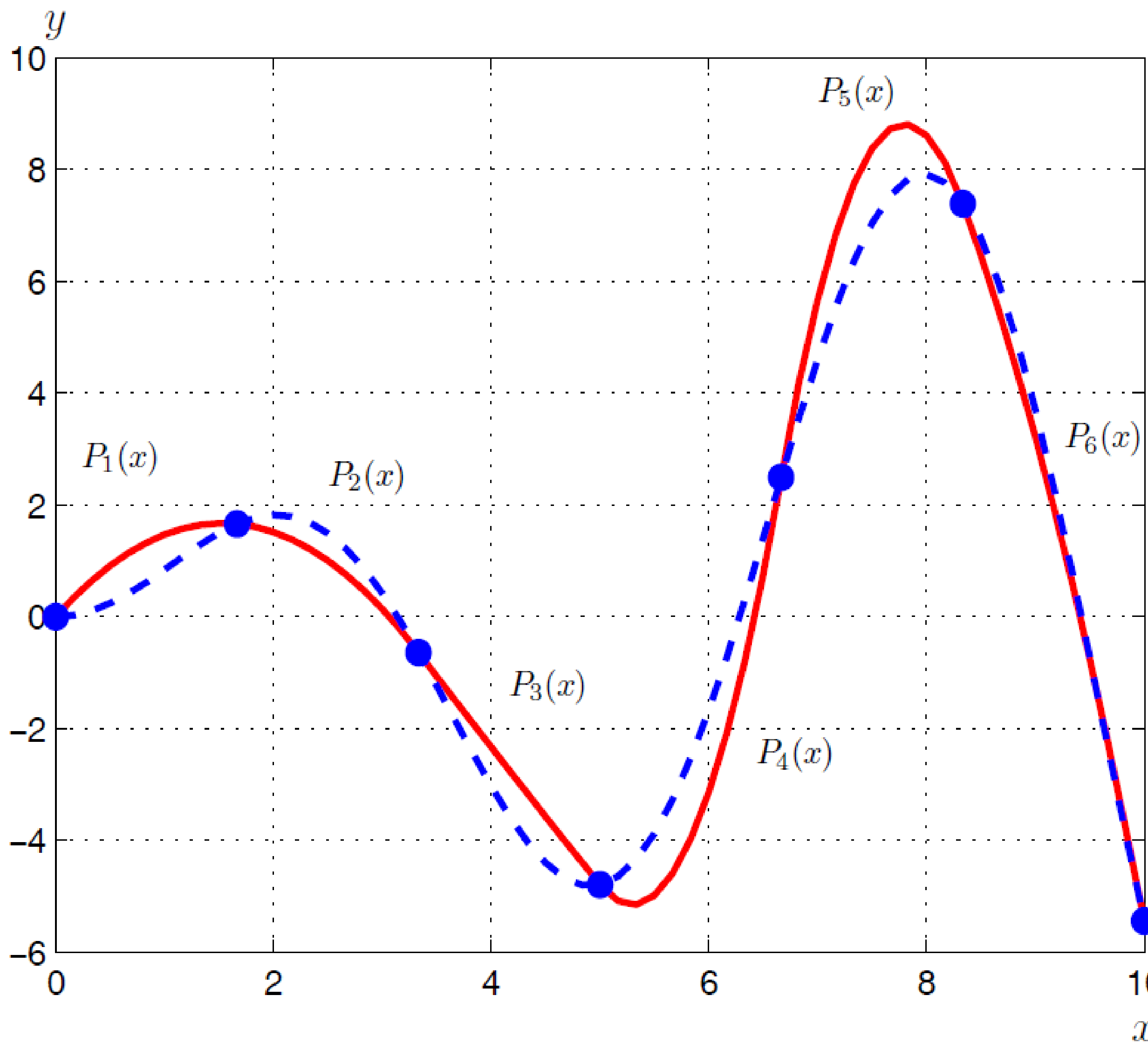


Spline interpolation

A spline is a function that, together with several derivatives, is continuous on the entire segment, and on each partial segment given by some algebraic polynomial is

$$P(x) = \begin{cases} P_1(x), & x_1 \le x < x_2, \\ P_2(x), & x_2 \le x < x_3, \\ \quad \dots & \\ P_{n-1}(x), & x_{n-1} \le x \le x_n. \end{cases}$$

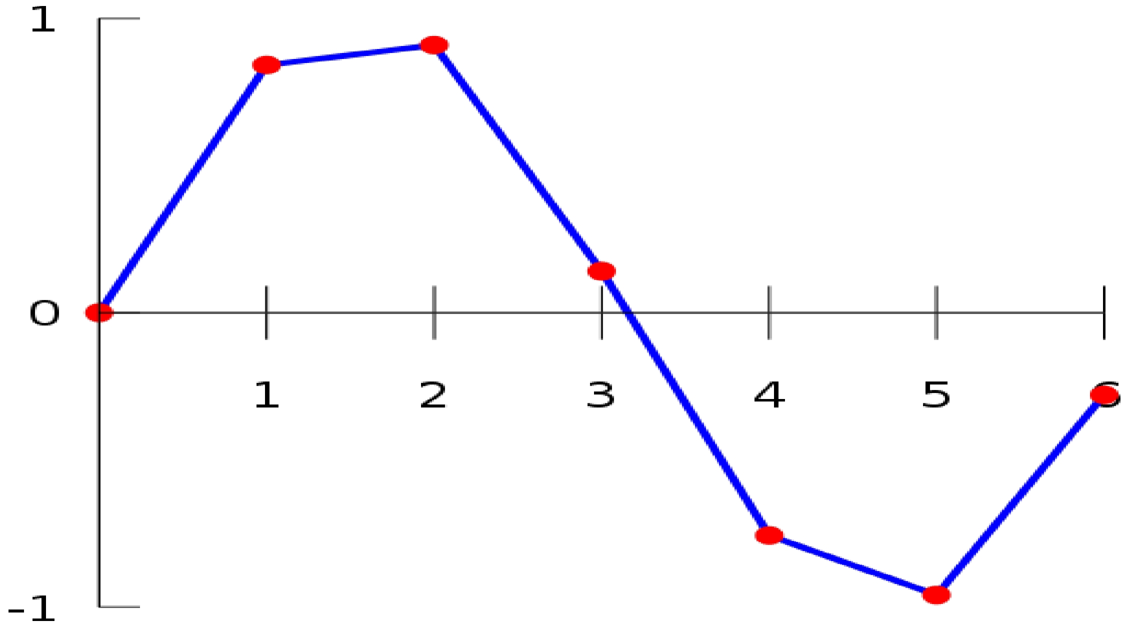
$$x_{i+1} - x_i = h = \text{const}, \quad i = \overline{1, n-1}.$$



y
10
8
6
4
2
0
−2
−4
−6
0
2
4
6
8
10
x
$P_1(x)$
$P_2(x)$
$P_3(x)$
$P_4(x)$
$P_5(x)$
$P_6(x)$

Spline interpolation

The degree of a spline is the maximum degree of polynomials over all partial segments, and the defect of a spline is the difference between the degree of the spline and the order of the highest continuity on the segment $[x_{1,} x_n]$ of the derivative.



In practice, cubic splines are most often used - splines of the third degree with continuous first and second derivatives.

Quadratic splines.

Derivation of formulas.

x	y
x_1	y_1
x_2	y_2
x_3	y_3
…	…
x_i	y_i
x_{i+1}	y_{i+1}
x_{i+2}	y_{i+2}
…	…
x_n	y_n

On every segment

we define a polynomial of degree 2 in the form

$$\boxed{\begin{array}{l} P_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)(x - x_{i+1}) \\ \qquad i = 1,\ldots,n-1 \end{array}}$$

How to determine the odds?

What conditions should the interpolation function satisfy?

At the nodes, the values of the polynomial must coincide with the values of the function

$$P_i(x_i) = a_i = y_i$$

$$P_i(x_{i+1}) = a_i + b_i(x_{i+1} - x_i) = y_{i+1}, \quad 1 \le i \le n-1$$

$$b_i = (y_{i+1} - y_i)/(x_{i+1} - x_i)$$

At the nodes, the values of the first derivatives of neighboring polynomials must coincide

$$P'_i(x_{i+1}) = P'_{i+1}(x_{i+1})$$

$$P_i(x) = a_i + b_i(x - x_i) + c_i(x^2 - x_i x - x_{i+1}x + x_i x_{i+1})$$

$$i = 1, \ldots, n-1$$

$$P'_i(x) = \quad b_i + c_i(2x - x_i - x_{i+1})$$

$$P'_{i+1}(x) = b_{i+1} + c_{i+1}(2x - x_{i+1} - x_{i+2})$$

At the nodes, the values of the first derivatives of neighboring polynomials must coincide

$$P'_i(x_{i+1}) = P'_{i+1}(x_{i+1})$$

$$=$$

$$=$$

$$b_{i+1} + c_{i+1}(x_{i+1} - x_{i+2})$$

$$b_i + c_i(2x_{i+1} - x_i - x_{i+1})$$

$$b_i + c_i(x_{i+1} - x_i)$$

$$b_{i+1} + c_{i+1}(2x_{i+1} - x_{i+1} - x_{i+2})$$

At the nodes, the values of the first derivatives of neighboring polynomials must coincide

$$P'_i(x_{i+1}) = P'_{i+1}(x_{i+1})$$

$$b_i + c_i(x_{i+1} - x_i) = b_{i+1} + c_{i+1}(x_{i+1} - x_{i+2})$$

$$c_{i+1}(x_{i+2} - x_{i+1}) = (b_{i+1} - b_i) - c_i(x_{i+1} - x_i)$$

$$c_{i+1} = \frac{(b_{i+1} - b_i) - c_i(x_{i+1} - x_i)}{(x_{i+2} - x_{i+1})}$$

recurrent formula

$$i = 2, 3, \dots, n-2$$

To calculate c1, an additional condition is needed

- natural spline

$$P''_1(x) = 0$$

$$c_1 = 0$$

or

- *2nd order smoothness condition at 2 node*

$$P''_1(x_2) = P''_2(x_2)$$

$$c_1 = c_2$$

$$c_1 = c_2 = \frac{(b_2 - b_1)}{(x_3 - x_1)}$$

Formulas for the coefficients of all polynomials for the natural spline condition

$$a_i = y_i$$

$$b_i = (y_{i+1} - y_i)/(x_{i+1} - x_i)$$

$$1 \le i \le n-1$$

$$c_1 = 0$$

$$c_{i+1} = \frac{(b_{i+1} - b_i) - c_i(x_{i+1} - x_i)}{(x_{i+2} - x_{i+1})}$$

$$1 \le i \le n-2$$

Formulas for the coefficients of all polynomials for the second order smoothness condition at the node x_2

$$a_i = y_i$$

$$b_i = (y_{i+1} - y_i)/(x_{i+1} - x_i)$$

$$2 \le i \le n-1$$

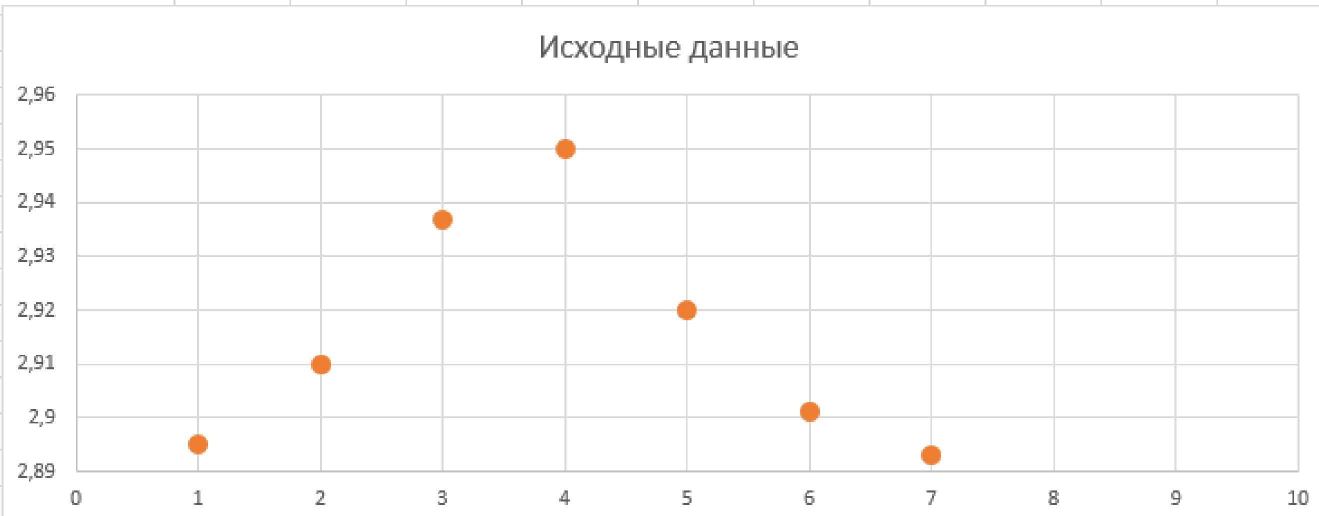
$$c_2 = \frac{(b_2 - b_1)}{(x_3 - x_1)}$$

$$c_{i+1} = \frac{(b_{i+1} - b_i) - c_i(x_{i+1} - x_i)}{(x_{i+2} - x_{i+1})}$$

$$2 \le i \le n-2$$

Quadratic Spline Interpolation Example

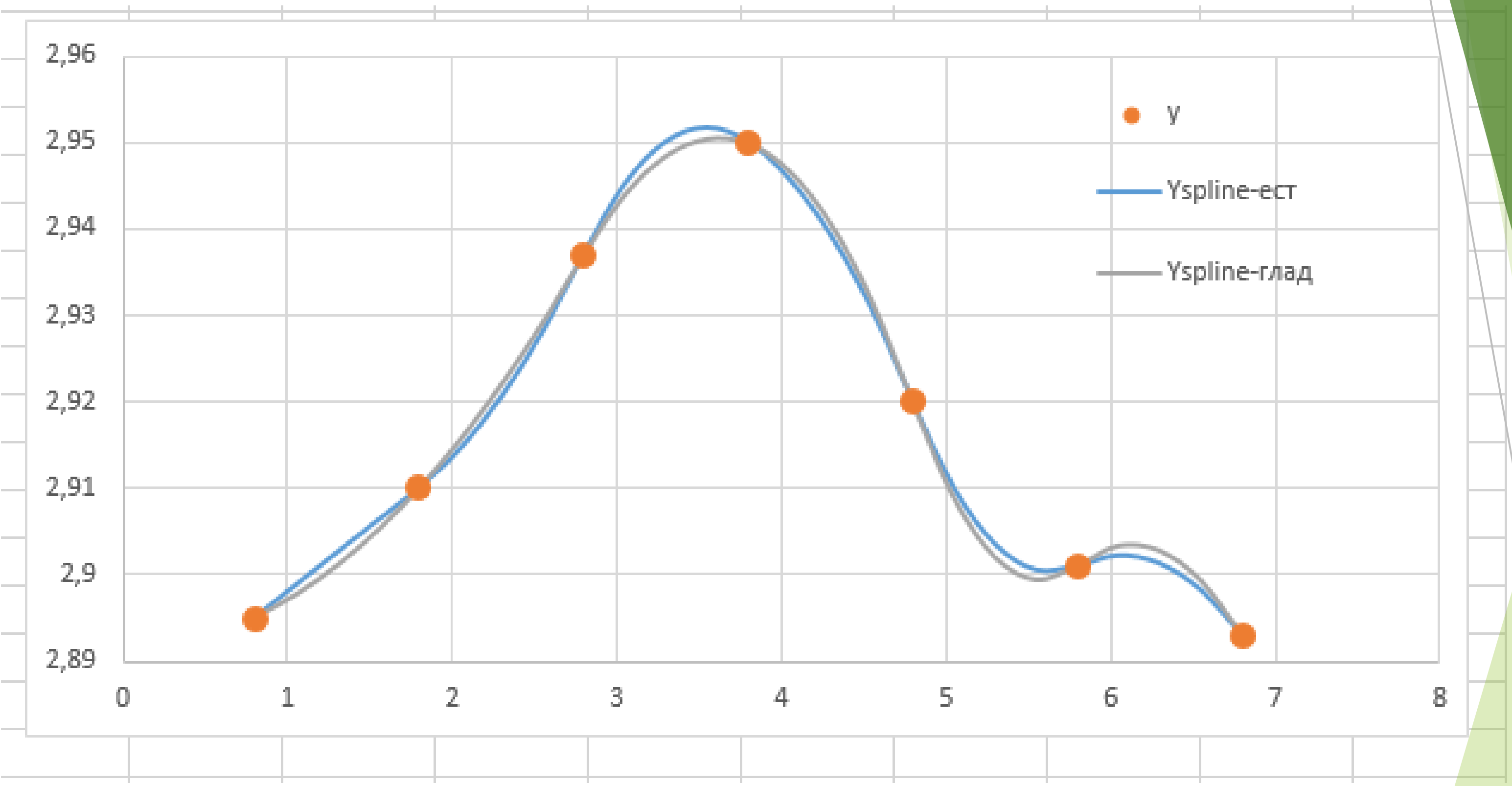
	A	B
1	x	Исходные данные
2	0,8	2,895
3	1,8	2,91
4	2,8	2,937
5	3,8	2,95
6	4,8	2,92
7	5,8	2,901
8	6,8	2,893
9		
10	x	
11	0,8	
12	1,05	
13	1,3	
14	1,55	
15	1,8	
16	2,05	
17	2,3	
18	2,55	
19	2,8	
20	3,05	
21	3,3	
22	3,55	
23	3,8	
24	4,05	
25	4,3	
26	4,55	
27	4,8	
28	5,05	
29	5,3	
30	5,55	
31	5,8	
32	6,05	



Quadratic spline coefficients for each interpolation segment

	A	B	C	D	E	F	G	H
1	x	y		a	b	c	N сплайна	
2	0,8	2,895		2,895	0,015	0	1	
3	1,8	2,91		2,91	0,027	0,012	2	
4	2,8	2,937		2,937	0,013	-0,026	3	
5	3,8	2,95		2,95	-0,03	-0,017	4	
6	4,8	2,92		2,92	-0,019	0,028	5	
7	5,8	2,901		2,901	-0,008	-0,017	6	
8	6,8	2,893						

Spline function plots for different calculation conditions c_1



Cubic splines

On each segment, a polynomial of degree 3 is given in the form

$$pi(x) = ai + bi\,(x - xi) + ci(x - xi)^2 + di(x - xi)^3$$

$$a_i = y_i$$

$$\begin{cases} c_1 = 0 \\ h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\left[\dfrac{y_{i+1} - y_i}{h_i} - \dfrac{y_i - y_{i-1}}{h_{i-1}}\right] & 2 \le i \le n-2 \\ c_{n-1} = 0 \end{cases}$$

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{1}{3}h_i(c_{i+1} + 2c_i), \quad 1 \le i \le n-2$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \quad i = 1, n-2$$

$$b_{n-1} = \frac{y_n - y_{n-1}}{h_{n-1}} - \frac{2}{3}h_{n-1}c_{n-1}$$

$$d_{n-1} = -\frac{c_{n-1}}{3h_{n-1}}$$

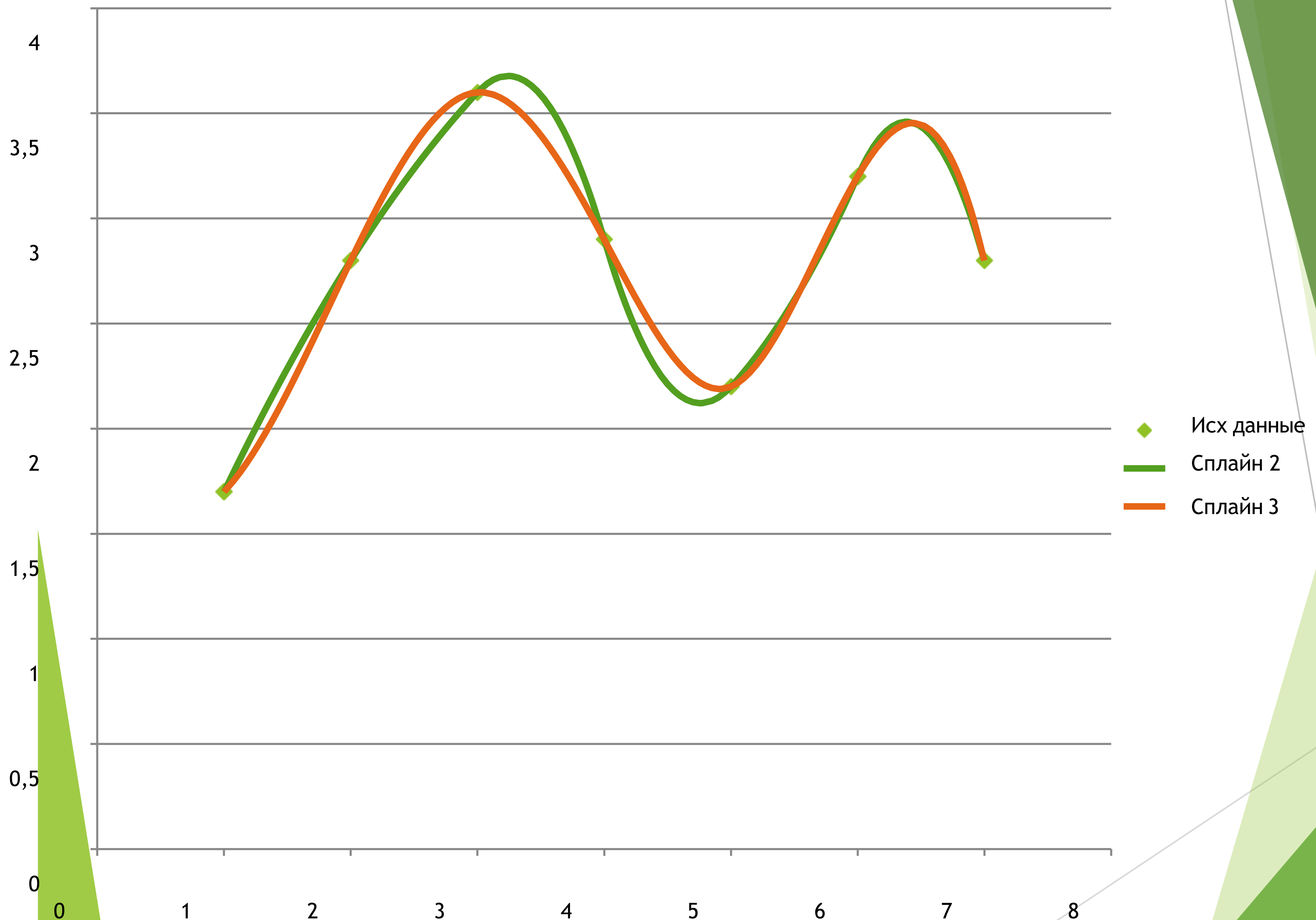
Conditions for the derivation of formulas for the coefficient of splines

1. At the nodes, the values of the polynomial must coincide with the values of the function
2. At the nodes, the 1st derivatives of neighboring polynomials must coincide
3. At the nodes, the 2nd derivatives of the neighboring polynomials must coincide.
4. Additional conditions for extreme polynomials: Equality of 1, 2 and 3 derivatives for extreme polynomials means that these polynomials are identical.

Cubic Spline Interpolation Example

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	x	y	Δy	$\Delta^2 y$		1	0	0	0	0	-0.15		
2	1	1.7	1.1	-0.3		1	4	1	0	0	-4.5		
3	2	2.8	0.8	-1.5		0	1	4	1	0	0		
4	3	3.6	-0.7	0		0	0	1	4	1	5.1		
5	4	2.9	-0.7	1.7		0	0	0	0	1	-0.7		
6	5	2.2	1	-1.4									
7	6	3.2	-0.4			c1	-0.150		1	0	0	0	0
8	7	2.8				c2	-1.062		-0.26786	0.267857	-0.07143	0.017857	-0.01786
9						c3	-0.104		0.071429	-0.07143	0.285714	-0.07143	0.071429
10						c4	1.476		-0.01786	0.017857	-0.07143	0.267857	-0.26786
11						c5	-0.700		0	0	0	0	1
12													
13	a	b	c	d	i								
14	2.8	0.8	-1.062	-0.3039	2								
15	3.6	-0.7	-0.104	0.3193	3								
16	2.9	-0.7	1.476	0.5265	4								
17	2.2	1	-0.700	-0.7253	5								

Cubic Spline Interpolation Example



An example of one-dimensional spline interpolation in a package MATLAB

To solve the problem, the **interp1 ()** function is used, which has the following syntax: **yi = interp1(x,y,xi)**

linear interpolation of table values x, y at points whose abscissas are in vector xi

yi = interp1(x,y,xi,method)

interpolate linear values using the selected interpolation method

Possible variable values **method**:

'nearest' interpolation using nearest nodes

'linear' linear interpolation (default)

'spline' cubic spline interpolation

'pchip' interpolation by Hermite polynomials of the third degree

'cubic' similarly **pchip**

In addition, there are standard functions

spline**()**

In addition, there are standard functions

The function p = **polyfit** (x, y, n) finds the coefficients of the polynomial p (x) of degree n, which approximates the function y (x) in the sense of the least squares method. The output is a string p of length n +1, containing the coefficients of the approximating polynomial.

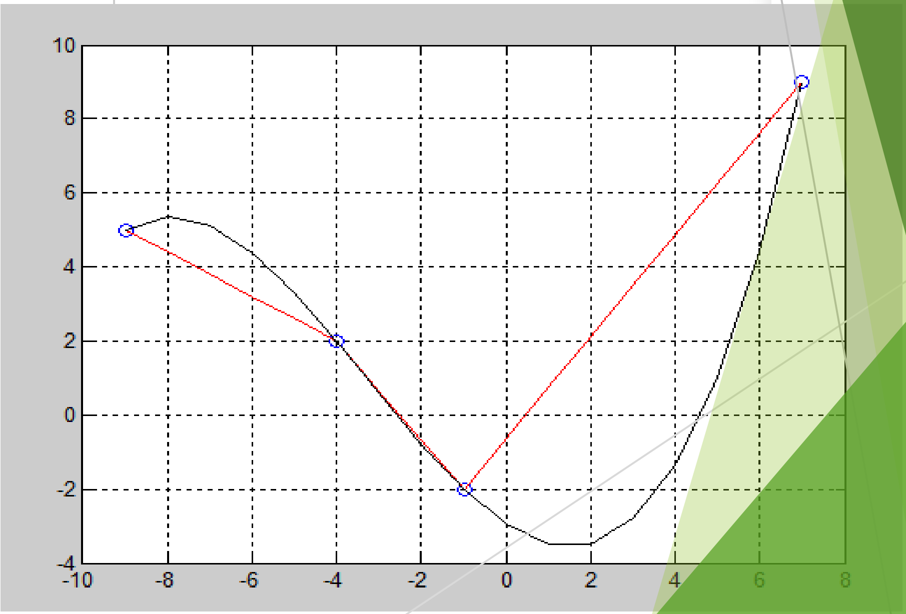
The function y **= polyval** (p, s), where p = [p1 p2 ... pn pn + 1] is the vector of coefficients of the polynomial p (x), calculates the value of this polynomial at the point x = s.
Function Y = **polyval** (p, S), where S is a one-dimensional or two-dimensional array, calculates the value of this polynomial for each element in the array.

The function yi = **spline** (x, y, xi) interpolates the values of the function y at the points xi within the domain of the function using cubic splines.

The function v = **ppval** (pp, xx) evaluates the value of the piecewise smooth polynomial pp for the values of the argument xx.

The pp = **mkpp** (breaks, coefs) function forms a piecewise smooth pp polynomial by its characteristics:
breaks - vector of argument splitting;
coefs - Coefficients of cubic splines.

```
clear all
clc
close all
% Задание табличных значений интерполируемой функции
 x=[-9 -4 -1 7];
y=[5 2 -2 9];
plot(x,y,'bo')
hold on
grid on
%Задание значения абсцисс точек, в которых вычисляется значение интерполя
xx=-9:7
%Вычисление интерполируемых значений функции в узлах координатной сетки
yi=interp1(x,y,xx);
plot(xx,yi, 'r')
hold on
yy=interp1(x,y,xx, 'spline');
plot(xx,yy, 'k')
hold on
```



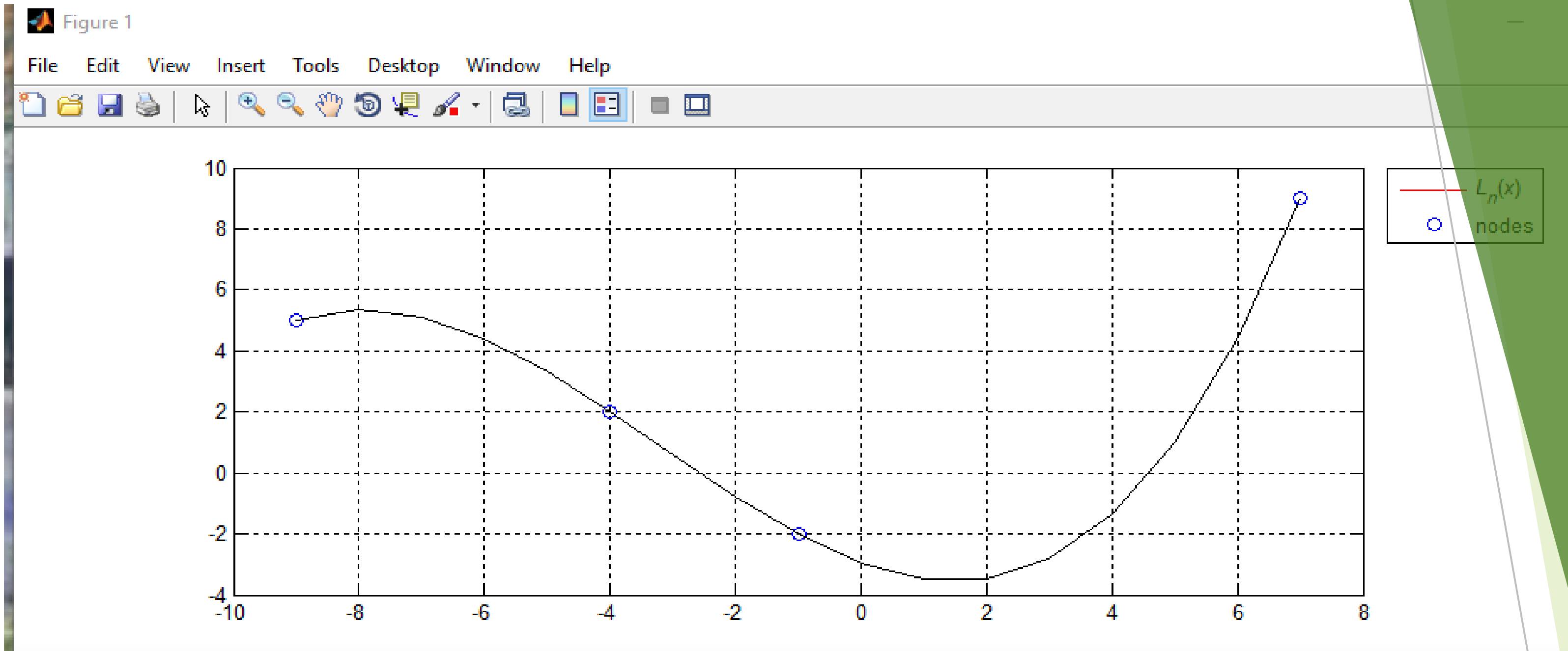


Figure 1
File Edit View Insert Tools Desktop Window Help
10
8
6
4
2
0
-2
-4
-10 -8 -6 -4 -2 0 2 4 6 8
$L_n(x)$
nodes

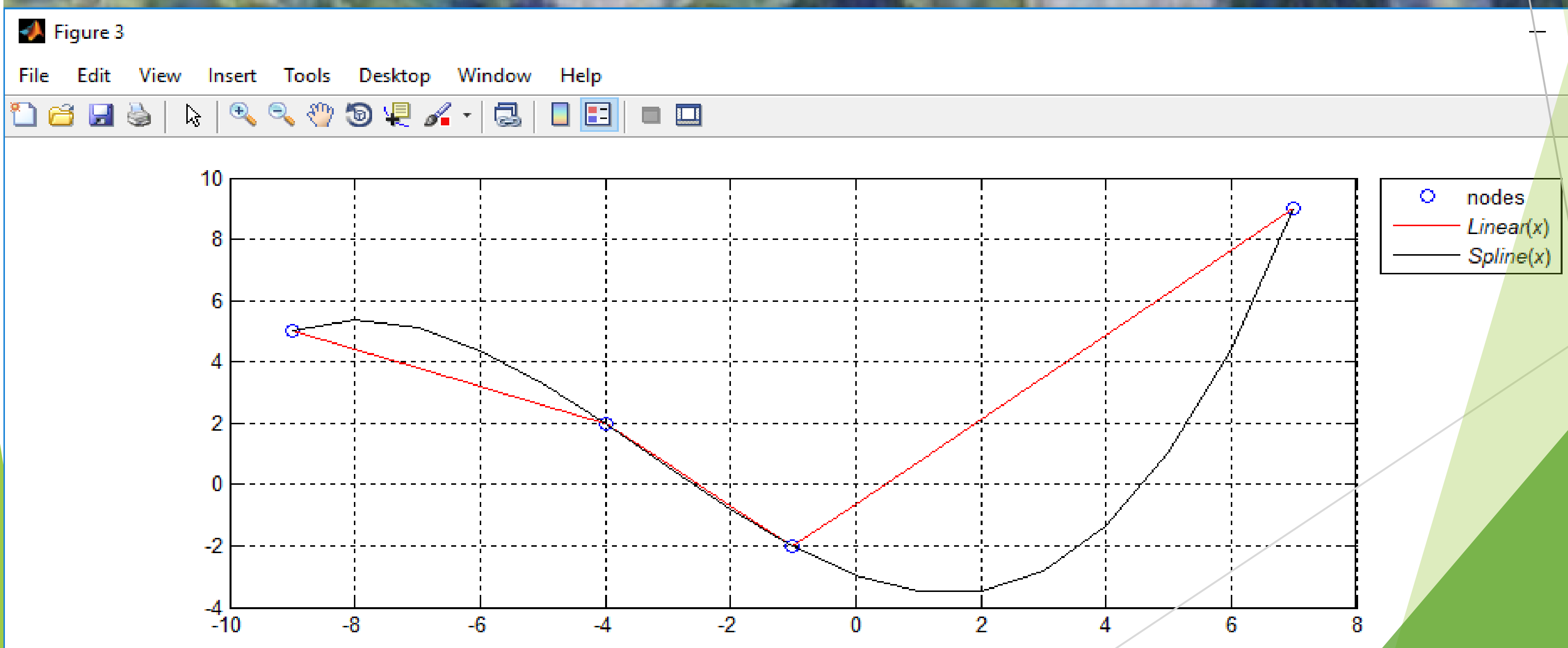
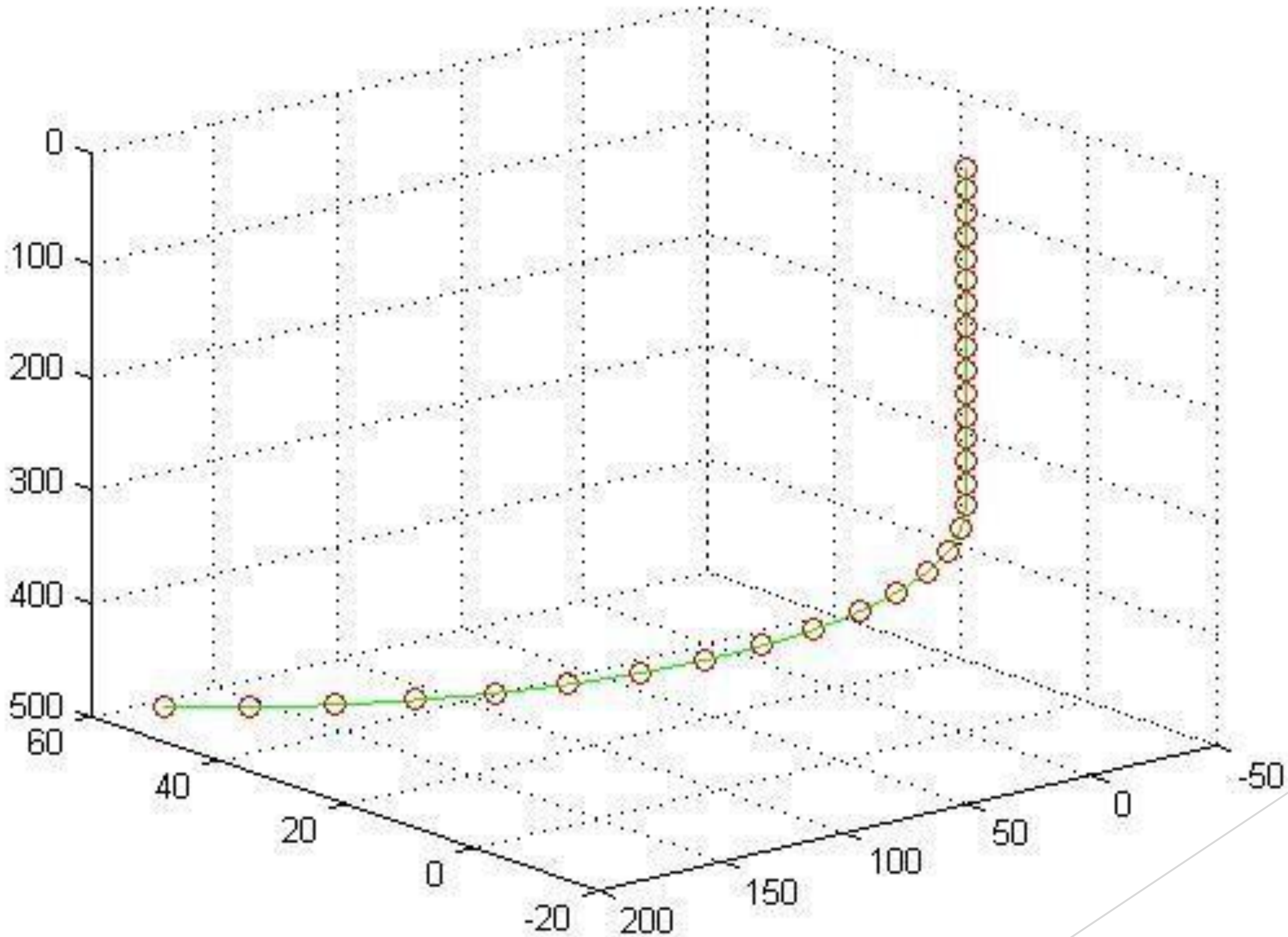


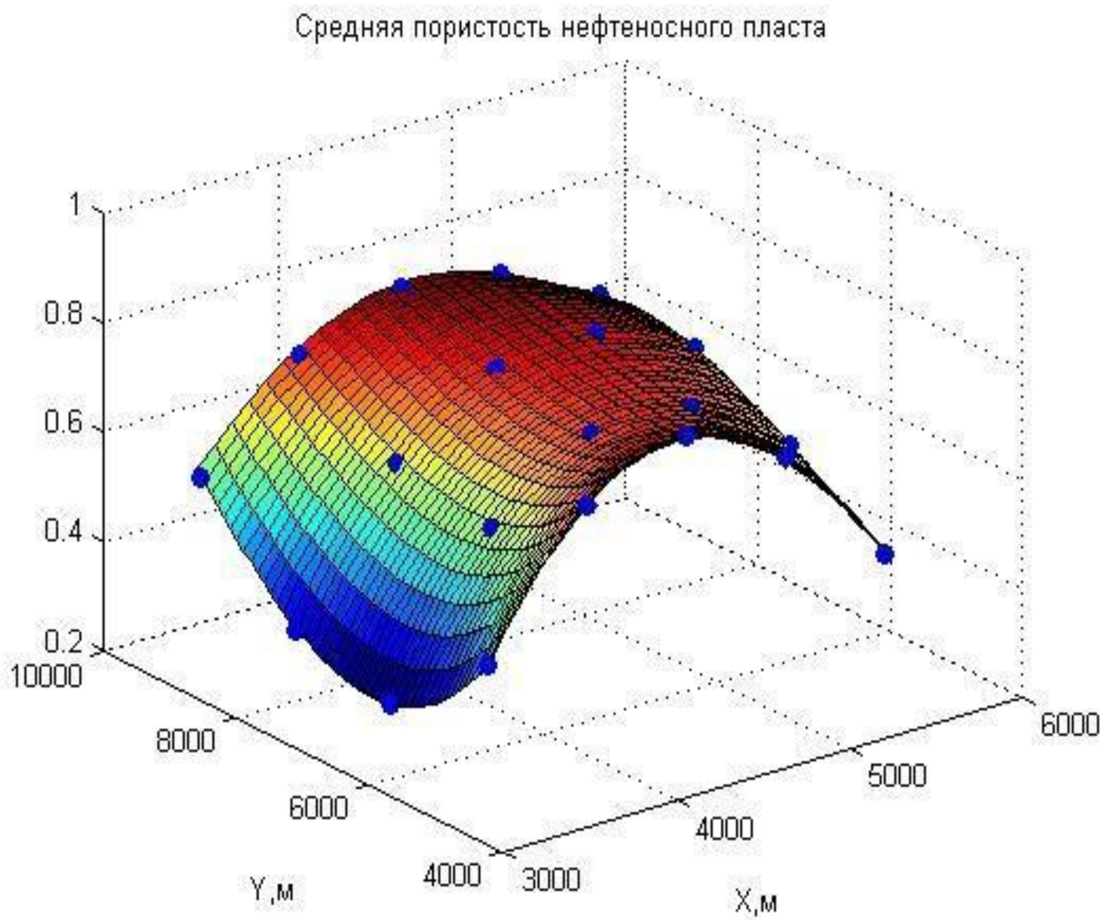
Figure 3
File Edit View Insert Tools Desktop Window Help
10
8
6
4
2
0
-2
-4
-10 -8 -6 -4 -2 0 2 4 6 8
nodes
Linear(x)
Spline(x)

An example of using spline interpolation to construct a horizontal well profile



Interpolation of a function of 2 variables

An example of plotting an interpolation function of two variables in Matlab



THANK YOU FOR ATTENTION