**Отчет**

**по Методам численного анализа**

Исполнитель

студент 161 группы

специальности

Прикладная математика

Борис Д. Ю.

«13» Апреля 2018 г.

Гродно, 2018

**ЛАБОРАТОРНАЯ РАБОТА № 6**

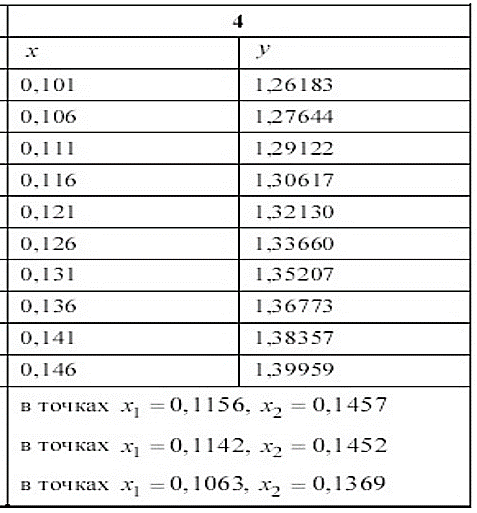
**Тема. Интерполяционный многочлен Ньютона**

1.Построить график интерполяционного многочлена Ньютона (первый и второй).

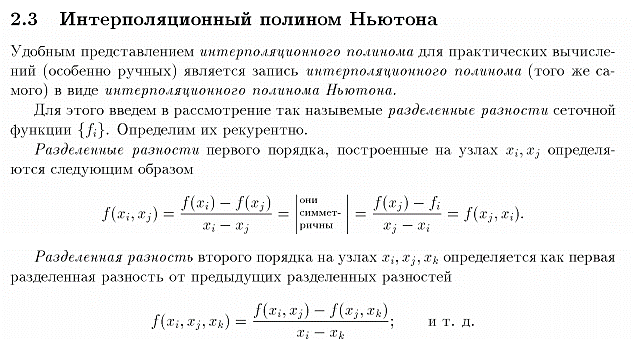
2.Отметить на нем узлы интерполяции.

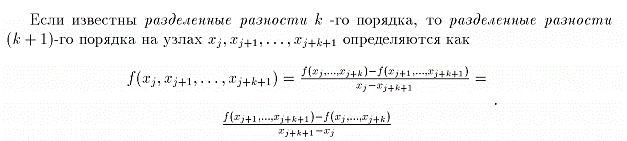
3. Вычислить значения многочлена в некоторых точках.

Условие:



1. **Теоретический материал**





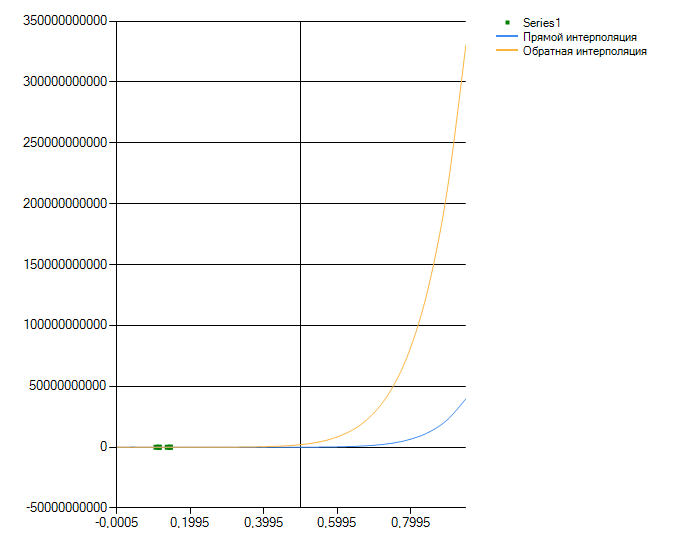
Прямая интерполяционная формула Ньютона



## Обратная интерполяционная формула Ньютона



1. **Результат вычислений**



1. **Текст программы**

*Метод прямой интерполяции*

static public double Newton(double x, int n, double[] MasX, double[] MasY, double step)

{

double[,] mas = new double[n + 2, n + 1];

for ( int i = 0 ; i < 2 ; i++ )

{

for ( int j = 0 ; j < n + 1 ; j++ )

{

if ( i == 0 )

mas[i, j] = MasX[j];

else if ( i == 1 )

mas[i, j] = MasY[j];

}

}

int m = n;

for ( int i = 2 ; i < n + 2 ; i++ )

{

for ( int j = 0 ; j < m ; j++ )

{

mas[i, j] = mas[i - 1, j + 1] - mas[i - 1, j];

}

m--;

}

double[] dy0 = new double[n + 1];

for ( int i = 0 ; i < n + 1 ; i++ )

{

dy0[i] = mas[i + 1, 0];

}

double res = dy0[0];

double[] xn = new double[n];

xn[0] = x - mas[0, 0];

for ( int i = 1 ; i < n ; i++ )

{

double ans = xn[i - 1] \* ( x - mas[0, i] );

xn[i] = ans;

ans = 0;

}

int m1 = n + 1;

int fact = 1;

for ( int i = 1 ; i < m1 ; i++ )

{

fact = fact \* i;

res = res + ( dy0[i] \* xn[i - 1] ) / ( fact \* Math.Pow(step, i) );

}

return res;

}

*Метод обратной интерполяции*

static public double inverse\_Newton(double x, int n, double[] MasX, double[] MasY, double step)

{

double[,] mas = new double[n + 2, n + 1];

for ( int i = 0 ; i < 2 ; i++ )

{

for ( int j = 0 ; j < n + 1 ; j++ )

{

if ( i == 0 )

mas[i, j] = MasX[j];

else if ( i == 1 )

mas[i, j] = MasY[j];

}

}

int m = n;

for ( int i = 2 ; i < n + 2 ; i++ )

{

for ( int j = 0 ; j < m ; j++ )

{

mas[i, j] = mas[i - 1, j + 1] - mas[i - 1, j];

}

m--;

}

double[] dy = new double[n + 1];

for ( int i = 0 ; i < n + 1 ; i++ )

{

dy[i] = mas[i + 1, 0];

}

double res = dy[n];

double[] xn = new double[n];

xn[0] = x - mas[0, 0];

for ( int i = 1 ; i < n ; i++ )

{

double ans = xn[i - 1] \* ( x + mas[0, i] );

xn[i] = ans;

ans = 0;

}

int m1 = n + 1;

int fact = 1;

for ( int i = 1, j=n ; i < m1 ; i++,j-- )

{

fact = fact \* i;

res = res + ( dy[n] \* xn[i - 1] ) / ( fact \* Math.Pow(step, i) );

}

return res;

}