**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Кафедра вычислительной математики**

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**”Приближение функции** **с помощью интерполяционного полинома Лагранжа”**

Возовикова Никиты Александровича

студента 2 курса группы 10

специальности «Компьютерная Безопасность»

дневной формы получения

высшего образования

Научный руководитель:

старший преподаватель

Никифоров Иван Васильевич

Минск, 2021

**ОГЛАВЛЕНИЕ**

**1. Постановка задачи**

**2. Краткие теоретические сведения**

**3. Листинг программы**

**4. Результаты**

**5. Выводы**

**Постановка задачи**

Функция на отрезке задана таблицей своих значений с шагом . Определить наибольшую погрешность линейной интерполяции, если:

Построить графики функций.

**Краткие теоретические сведения**

Рассмотрим многочлены степени n, удовлетворяющих условиям

которые называются многочленами Лагранжа. Их можно записать в виде

Тогда интерполяционный многочлен в форме Лагранжа принимает вид

По определению интерполяционного полинома



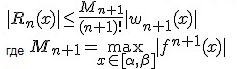
поэтому речь идет об оценке при значениях /

Пусть имеет непрерывную (n+1) производную на отрезке [a, b]. Тогда погрешность определяется формулой:



- точка из [a, b].

Так как точка  наизвестна, то эта формула позволяет только оценить погрешность:



**Листинг программы**

namespace MHA1

{

public partial class Form1 : Form

{

private double leftBound = -5;

private double rightBound = -1;

private double h = 0.25;

private List<double> nodes = new List<double>();

private List<double> nodes1 = new List<double>();

public double fun(double x)

{

return Math.Exp(2 \* x);

}

public double interpolFun(double x)

{

return summarize(x);

}

public double summarize(double x)

{

double summ = 0;

for (int i = 0; i < nodes.Count; i++)

summ += fun(nodes[i]) \* multiplication(x, i, nodes.Count);

return summ;

}

public double multiplication(double x, int i, int n)

{

double mult = 1.0;

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

if (i != j)

mult \*= ((x - nodes[j]) / (nodes[i] - nodes[j]));

return mult;

}

public void printFun(Series series, List<double> nodes, Func<double ,double> fun)

{

foreach(double el in nodes)

series.Points.AddXY(el, fun(el));

}

public void nodesInit(List<double> nodes, double leftBound, double rightBound, double h)

{

for (double i = leftBound; i <= rightBound; i += h)

nodes.Add(i);

}

public double mesureError()

{

double max = maxAbs(leftBound, rightBound, nodes.Count) \* omega(leftBound+0.0013, nodes.Count) / factorial(nodes.Count);

for (double i = leftBound + 0.001; i < rightBound; i += 0.0013)

max = Math.Min(max, maxAbs(leftBound, rightBound, nodes.Count) \* omega(i, nodes.Count) / factorial(nodes.Count));

return Math.Abs(max);

}

public double maxAbs(double leftBound, double rightBound, int n)

{

double max = Math.Abs(derivateExp(n) \* fun(leftBound));

for (double i = leftBound; i <= rightBound; i += 0.0001)

max = Math.Max(max, derivateExp(nodes.Count) \* fun(i));

return max;

}

public double derivateExp(int n)

{

return n \* 2;

}

public double factorial(int n)

{

int mult = 1;

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

mult \*= i;

return mult;

}

public double omega(double x,int n)

{

double mult = 1;

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

mult \*= (x - nodes[i]);

return Math.Abs(mult);

}

public Form1()

{

InitializeComponent();

chart1.Series[0].Color = Color.Indigo;

chart1.Series[0].BorderWidth = 2;

chart1.Series[0].ChartType = SeriesChartType.Line;

chart1.Series.Add(new Series("Polinom"));

chart1.Series[1].Color = Color.LightGreen;

chart1.Series[1].BorderWidth = 2;

chart1.Series[1].ChartType = SeriesChartType.Line;

nodesInit(nodes1, leftBound, rightBound, 0.001);

nodesInit(nodes, leftBound, rightBound, h);

printFun(chart1.Series[0], nodes1, fun);

printFun(chart1.Series[1], nodes, interpolFun);

chart1.Series.Add(new Series("Error: "+mesureError().ToString()));

chart1.Series[2].Color = Color.Red;

}

private void chart1\_Click(object sender, EventArgs e)

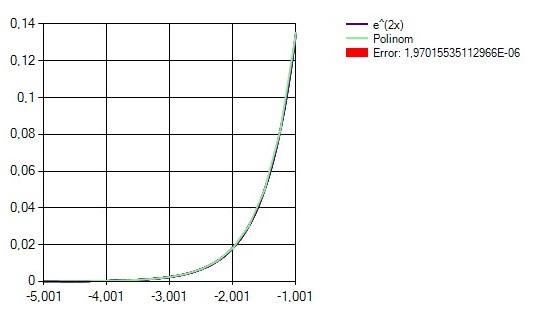
{

}

}

}

**Результаты**

****

*Рис 1.1 График функции и итерполяционного полинома Лагранжа*

Наибольшая погрешность интерполяции: **1,97015535112966E-06**

**Выводы**

По окончанию выполнения лабораторной работы было реазизованно приближение функции с помощью интерполяционного полинома Лагранжа и выведены соотвествующие графики. В результате решения была найдена наибольшая погрешность интерполяции равная 1,97015535112966E-06

Данная задача, а именно интерполяция функций, часто встречается при ограниченности возможностей при проведении эксперимента. В частности из-за дороговизны и трудоемкости проведения эксперимента размер выборки (x0, x1, x2,..., xn) может быть достаточно мал. При этом во многих случаях аналитическое выражение функции y(x) не известно и получить его по таблице ее значений в большинстве случаев невозможно.

Интерполяционный полином Лагранжа обычно применяется в теоретических исследованиях (при доказательстве теорем, аналитическом решении задач и т. п.).