Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №3**

По дисципліні «Алгоритми та методи обчислень»

Тема: «Інтерполяція функцій»

Виконав: Перевірив:

Студент групи ІО-21 ст. викладач

Коноз А.О. Порєв В. М.

Дата здачі\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з балом\_\_\_\_\_

Київ 2014

**Мета:** Ознайомлення з інтерполяційними формулами Лагранжа, Ньютона,

рекурентним співвідношенням Ейткена, методами оцінки похибки інтерполяції.

**Завдання:** Закріплення, поглиблення і розширення знань студентів при

вирішенні практичних обчислювальних завдань. Оволодіння обчислювальними

методами і практичними методами оцінки похибки обчислень. Придбання умінь

і навичок при програмуванні та налагодженні обчислювальних завдань на

комп'ютері.

**Код програми:**

package Lab3;

import org.jfree.chart.ChartFactory;

import org.jfree.chart.ChartPanel;

import org.jfree.chart.JFreeChart;

import org.jfree.chart.plot.PlotOrientation;

import org.jfree.data.xy.XYSeries;

import org.jfree.data.xy.XYSeriesCollection;

import org.jfree.ui.ApplicationFrame;

import org.jfree.ui.RefineryUtilities;

public class XYSeriesDemo extends ApplicationFrame {

double arg = Math.PI\*0.25;

public XYSeriesDemo(final String title) {

super(title);

Lagr lagr = new Lagr();

lagr.countTable(0, 5, 10);

double functValues [][] = lagr.getFunctValues();

final XYSeries series = new XYSeries("Відомі значення функції");

for (int i = 0; i < functValues.length; i++) {

series.add(functValues[i][0], functValues[i][1]);

}

final XYSeries series2 = new XYSeries("Результат інтерполяції");

double step = 0.05;

for (int i = 0; i < 100; i++) {

series2.add(step\*i, lagr.lagr(0, 5, step\*i, 10));

}

lagr.countTable(0, 5, 10);

lagr.showTable();

lagr.countTable(0, 5, 10);

final XYSeriesCollection data = new XYSeriesCollection(series);

data.addSeries(series2);

final JFreeChart chart = ChartFactory.createXYLineChart(

"Інтерполяція",

"X",

"Y",

data,

PlotOrientation.VERTICAL,

true,

true,

false

);

final ChartPanel chartPanel = new ChartPanel(chart);

chartPanel.setPreferredSize(new java.awt.Dimension(800, 400));

setContentPane(chartPanel);

}

public static void main(final String[] args) {

final XYSeriesDemo demo = new XYSeriesDemo("Інтерполяція");

demo.pack();

RefineryUtilities.centerFrameOnScreen(demo);

demo.setVisible(true);

}

}

**package** Lab3;

**public** **class** Lagr {

**private** **double** values [][];

**private** **double** a;

**private** **double** b;

**public** **double**[][] getFunctValues(){

**return** values;

}

**public** **void** showTable(){

**double** interpol[] = **new** **double**[12];

**for** (**int** i = 1; i < interpol.length; i++) {

countTable(0, 5, i);

interpol[i] = lagr(a, b, 3.232323, i);

}

**double** delta;

**double** deltaExact;

**double** kDelta;

**for** (**int** i = 1; i < interpol.length - 1; i++) {

delta = interpol[i] - interpol[i + 1];

countTable(0, 5, i);

deltaExact = interpol[i] - getFunctionValue(3.232323);

kDelta = Math.*abs*(1 - delta/deltaExact);

System.*out*.print("n = " + i);

System.*out*.print(" delta = " + delta);

System.*out*.print(" deltaExact = " + deltaExact);

System.*out*.print(" kDelta = " + kDelta);

System.*out*.println();

}

}

**public** **int** fact(**int** n){

**int** result = 1;

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

result \*= i;

}

**return** result;

}

**public** **void** countTable(**double** a, **double** b, **int** n){

**this**.a = a;

**this**.b = b;

**double** h = (b - a)/n;

values = **new** **double** [n+1][2];

**for** (**int** i = 0; i < values.length; i++) {

values[i][0] = a + i\*h;

values[i][1] = getFunctionValue(a + i\*h);

}

}

**public** **double** lagr(**double** a, **double** b, **double** x, **int** n){

**double** h = (b - a)/n;

**double** m = (x - a)/h;

**double** result = 0;

**double** inres = 1;

**for** (**int** i = 0; i < values.length; i++) {

inres = 1;

**for** (**int** j = 0; j < values.length; j++) {

**if**(i != j){

inres \*= (m - j);

}

}

result += (Math.*pow*(-1, (n - i)))\*inres\*values[i][1]/(fact(i)\*fact(n - i));

}

**return** result;

}

**public** **double** getFunctionValue(**double** x){

**return** Math.*sin*(x);

// return Math.pow(Math.E, Math.sin(x));

}

**public** **static** **void** main(String[] args) {

**double** arg = Math.*PI*\*0.25;

Lagr ob = **new** Lagr();

ob.countTable(0, 5, 10);

System.*out*.println(ob.lagr(0, 5, 1.5, 10));

}

}

**Результат виконання програми:**

n = 1 delta = -0.9854394181050863 deltaExact = -0.5293046819846319 kDelta = 0.8617621412494856

n = 2 delta = 0.4699512845138256 deltaExact = 0.45613473612045435 kDelta = 0.030290498177982794

n = 3 delta = -0.03953593014785653 deltaExact = -0.013816548393371278 kDelta = 1.861491091857975

n = 4 delta = 0.024033689042098086 deltaExact = 0.02571938175448525 kDelta = 0.06554172757644905

n = 5 delta = 0.0014669465078387817 deltaExact = 0.0016856927123871657 kDelta = 0.1297663583290991

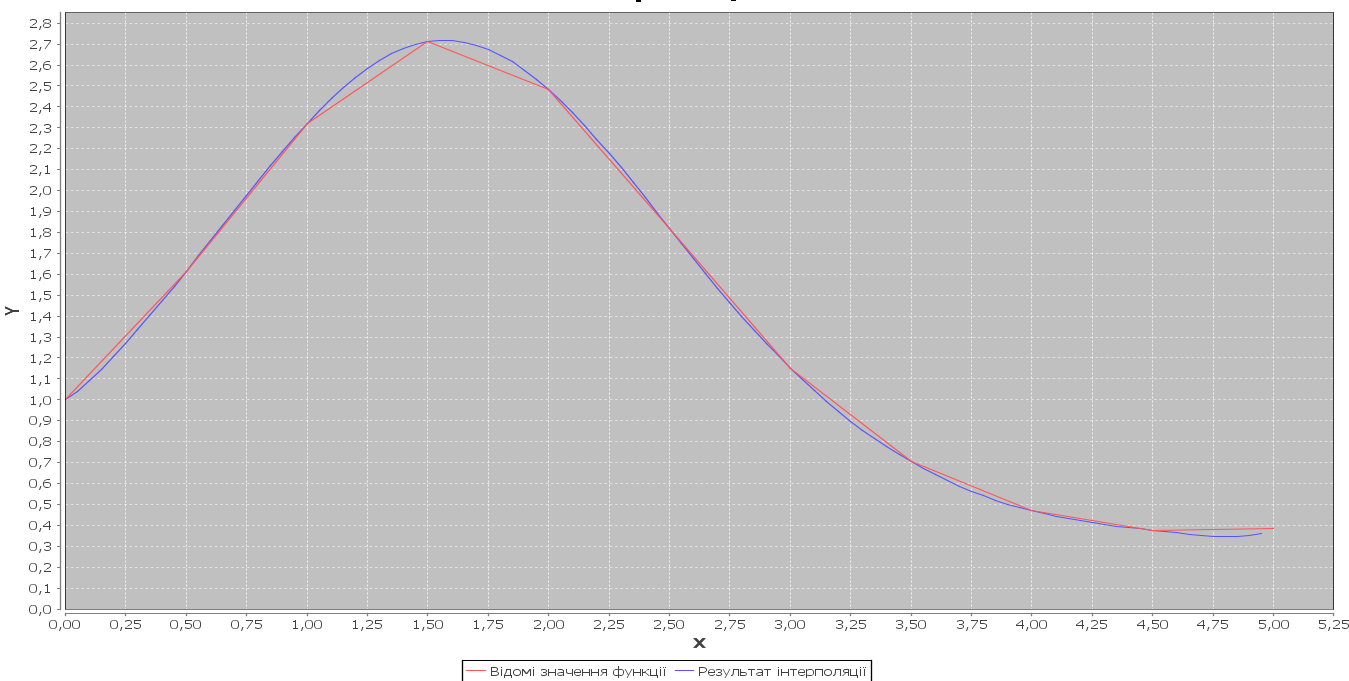
n = 6 delta = 1.7466092457912863E-4 deltaExact = 2.18746204548384E-4 kDelta = 0.20153620521221083

n = 7 delta = 4.797000649488237E-5 deltaExact = 4.4085279969255375E-5 kDelta = 0.08811844970330607

n = 8 delta = -4.138988559790335E-6 deltaExact = -3.884726525626991E-6 kDelta = 0.06545172034273539

n = 9 delta = 3.1742850543914347E-7 deltaExact = 2.542620341633439E-7 kDelta = 0.248430606180158

n = 10 delta = -6.201385353843403E-8 deltaExact = -6.316647127579955E-8 kDelta = 0.01824730294546495

**Висновок:** Під час виконання даної лабораторної роботи були закріплені навички інтерполяції функції різними методами. Програмно був реалізований метод інтерполяції поліномом Лагранджа. Окрім того були закріплені основні навички роботи з графічною бібліотекою JFreeChart.