**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**Л А Б О Р А Т О Р Н А Р О Б О Т А № 3**

***Інтерполяція функцій***

Виконав:

Студент ФІОТ

Групи ІО – 31

Долинний Олександр

**Київ**

**2015**

**Лабораторна робота №3**

**1. Тема 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.

**2. Завдання:**

1) За вказівкою викладача вибрати метод інтерполяції (многочлени Лагранжа, Ньютона або рекурентне співвідношення Ейткена).

2) Скласти програму, що обчислює значення заданої функції у вузлах інтерполяції на відрізку [a, b].

3) Передбачити в програмі оцінку похибки на основі порівняння значень, отриманих за допомогою інтерполяційних многочленів різного степеня.

4) Оцінити розмитість оцінки похибки.

5) Налагодити програму шляхом інтерполяції функції sinx (див. «Чисельний експеримент»).

6) Застосувати програму для інтерполяції функції, з таблиці 2 за номером у списку.

7) Результат оцінки похибки представити у вигляді графіка (рис. 3, 4) і для одного з значень x у вигляді таблиці 1.

|  |  |  |
| --- | --- | --- |
| № Варіанту | f(x) | [a, b] |
| 11 | C:\Users\Alexandr\Desktop\0233.jpg | [0, 3] |

**3. Лістинг програми:**

public class Main {

public static void main(String[] args) {

Out dr = new Out();

}

}

public class Interpolation {

private double a;

private double b;

private double step;

private int count;

private double[] xNodes;

private double[] yNodes;

private double[] zNodes;

public Interpolation(double a, double b, int count) {

this.a = a;

this.b = b;

this.count = count;

this.step = (b - a) / this.count;

this.xNodes = new double[count + 1];

this.yNodes = new double[count + 1];

this.zNodes = new double[count + 1];

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

this.xNodes[i] = this.a + i \* this.step;

}

public void setY(String str) {

if (str.equals("sin(x)")) {

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

yNodes[i] = Math.sin(xNodes[i]);

}

}

if (str.equals("sin(x^2)\*exp(-(x^2)/4)")) {

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

yNodes[i] = Math.sin(Math.pow(xNodes[i], 2)) \* Math.exp(-xNodes[i] \* xNodes[i] / 4.0);

}

}

}

public double interp(String str, double xx) {

double result = 0;

if (str.equals("Лагранж")) {

result = this.interpLagranj(xx);

}

if (str.equals("Ньютон")) {

result = this.interpNewton(xx);

}

return result;

}

public double interpLagranj(double xx) {

double result = 0;

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

double temp = 1;

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

temp = temp \* (xx - xNodes[j]) / (xNodes[i] - xNodes[j]);

}

for (int j = i + 1; j <= count; j++) {

temp = temp \* (xx - xNodes[j]) / (xNodes[i] - xNodes[j]);

}

result = result + temp \* yNodes[i];

}

return result;

}

public double interpNewton(double xx) {

double result = yNodes[0];

double buf = 1;

for (int k = 1; k <= count; k++) {

double tempSum = 0;

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

double temp = 1;

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

temp = temp \* (xNodes[i] - xNodes[j]);

for (int j = i + 1; j <= k; j++)

temp = temp \* (xNodes[i] - xNodes[j]);

temp = yNodes[i] / temp;

tempSum += temp;

}

buf = buf \* (xx - xNodes[k - 1]);

result = result + tempSum \* buf;

}

return result;

}

public double getX(int i) {

return xNodes[i];

}

public double getY(int i) {

return yNodes[i];

}

public double getError(String strFun, String strInt, double xx){

Interpolation interp1 = new Interpolation(a,b,count+1);

interp1.setY(strFun);

return (interp1.interp(strInt,xx)-this.interp(strInt, xx));

}

public double getErrorOfError(String strFun, String strInt, double xx){

Interpolation interp1 = new Interpolation(a,b,count+1);

interp1.setY(strFun);

return interp1.getError(strFun, strInt,xx);

}

public double getBluriness(String strFun, String strInt, double xx){

return getErrorOfError(strFun, strInt,xx)/getError(strFun, strInt,xx);

}

}

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 javax.swing.\*;

import javax.swing.table.TableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Out {

public static int FRAME\_WIDTH = 450;

public static int FRAME\_HEIGHT = 300;

JPanel panel1 = new JPanel();

GridLayout g1 = new GridLayout(12, 2);

JPanel panel2 = new JPanel();

GridLayout g2 = new GridLayout(2, 1);

JPanel panel3 = new JPanel();

GridLayout g3 = new GridLayout(1, 1);

JPanel panel4 = new JPanel();

GridLayout g4 = new GridLayout(2, 2);

JLabel label1 = new JLabel("Оберіть тип інтерполяції");

JComboBox<String> typeInterpolation = new JComboBox<String>(new String[]{"Лагранж", "Ньютон"});

JLabel label2 = new JLabel("Оберіть функцію");

JComboBox<String> typeFunction = new JComboBox<String>(new String[]{"sin(x)", "sin(x^2)\*exp(-(x^2)/4)"});

JLabel label3 = new JLabel("Границя a");

JTextField field3 = new JTextField(10);

JLabel label4 = new JLabel("Границя b");

JTextField field4 = new JTextField(10);

JLabel label5 = new JLabel("Кількість точок");

JTextField field5 = new JTextField(10);

JButton task1 = new JButton("Обчислити");

JLabel label6 = new JLabel("");

JLabel label7 = new JLabel("");

JLabel label7A = new JLabel("");

JLabel label8 = new JLabel("x");

JTextField field8 = new JTextField(10);

JLabel label9 = new JLabel("y(x)=");

JTextField field9 = new JTextField(10);

JLabel label10 = new JLabel("Похибка");

JTextField field10 = new JTextField(10);

JLabel label11 = new JLabel("Похибка похибки");

JTextField field11 = new JTextField(10);

JLabel label12 = new JLabel("Відносна розмитість похибки");

JTextField field12 = new JTextField(10);

JLabel label13 = new JLabel(" ");

XYSeriesCollection data = new XYSeriesCollection();

XYSeriesCollection data1 = new XYSeriesCollection();

XYSeries series1 = new XYSeries("Вихідна функція");

XYSeries series2 = new XYSeries("Інтерпольована функція");

XYSeries series3 = new XYSeries("Похибки");

JTable table = new JTable(13, 4);

public Out() {

//створюємо вкладку

JFrame frame = new JFrame();

frame.setTitle("Третя лабораторна робота");

frame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT);

JTabbedPane tabby = new JTabbedPane();

panel1.setBackground(Color.white);

panel1.setLayout(g1);

panel2.setBackground(Color.white);

panel2.setLayout(g2);

panel3.setBackground(Color.white);

panel3.setLayout(g3);

panel4.setBackground(Color.white);

panel4.setLayout(g4);

panel1.add(label1);

panel1.add(typeInterpolation);

panel1.add(label2);

panel1.add(typeFunction);

panel1.add(label3);

panel1.add(field3);

panel1.add(label4);

panel1.add(field4);

panel1.add(label5);

panel1.add(field5);

panel1.add(task1);

panel1.add(label6);

panel1.add(label7);

panel1.add(label7A);

panel1.add(label8);

panel1.add(field8);

panel1.add(label9);

panel1.add(field9);

panel1.add(label10);

panel1.add(field10);

panel1.add(label11);

panel1.add(field11);

panel1.add(label12);

panel1.add(field12);

task1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

series1.clear();

series2.clear();

series3.clear();

double a = Double.parseDouble(field3.getText());

double b = Double.parseDouble(field4.getText());

double xx = Double.parseDouble(field8.getText());

int count = Integer.parseInt(field5.getText());

double step = (b - a) / count;

String typeInt = (String) typeInterpolation.getSelectedItem();

String typeFun = (String) typeFunction.getSelectedItem();

Interpolation src = new Interpolation(a, b, count);

src.setY(typeFun);

int c1 = 100;

double step1 = step / c1;

if (typeFun.equals("sin(x)")) {

for (int i = 0; i <= (count \* c1); i++) {

series1.add(a + step1 \* i, Math.sin(a + step1 \* i));

series2.add(a + step1 \* i, src.interp(typeInt, a + step1 \* i));

series3.add(a + step1 \* i, src.getError(typeFun, typeInt, a + step1 \* i));

}

}

if (typeFun.equals("sin(x^2)\*exp(-(x^2)/4)")) {

for (int i = 0; i <= (count \* c1); i++) {

series1.add(a + step1 \* i, Math.sin(Math.pow((a + step1 \* i), 2)) \* Math.exp(-(a + step1 \* i) \* (a + step1 \* i) / 4.0));

series2.add(a + step1 \* i, src.interp(typeInt, a + step1 \* i));

series3.add(a + step1 \* i, src.getError(typeFun, typeInt, a + step1 \* i));

}

}

field9.setText(String.valueOf(src.interp(typeInt, xx)));

field10.setText(String.valueOf(src.getError(typeFun, typeInt, xx)));

field11.setText(String.valueOf(src.getErrorOfError(typeFun, typeInt, xx)));

field12.setText(String.valueOf(src.getBluriness(typeFun, typeInt, xx)));

TableModel model = table.getModel();

model.setValueAt("n", 0, 0);

model.setValueAt("Оцінка похибки", 0, 1);

model.setValueAt("Точна похибка", 0, 2);

model.setValueAt("Коефіцієнт уточнення", 0, 3);

for (int i = 1; i <= count; i++) {

Interpolation src1 = new Interpolation(a, b, i);

src1.setY(typeFun);

double p1 = src1.getError(typeFun, typeInt, xx);

double p2 = 0;

if (typeFun.equals("sin(x)"))

p2 = src1.interp(typeInt, xx) - Math.sin(xx);

if (typeFun.equals("sin(x^2)\*exp(-(x^2)/4)"))

p2 = src1.interp(typeInt, xx) - Math.sin(Math.pow(xx, 2)) \* Math.exp(-xx \* xx / 4.0);

double p3 = 1 - p2 / p1;

model.setValueAt(i, i, 0);

model.setValueAt(p1, i, 1);

model.setValueAt(p2, i, 2);

model.setValueAt(p3, i, 3);

}

}

});

data.addSeries(series1);

data.addSeries(series2);

data1.addSeries(series3);

JFreeChart chart1 = ChartFactory.createXYLineChart(

"Графіки вихідної та інтерпольованої функцій",

"X",

"Y",

data,

PlotOrientation.VERTICAL,

true,

true,

false);

final ChartPanel chartPanel1 = new ChartPanel(chart1);

JFreeChart chart2 = ChartFactory.createXYLineChart(

"Похибки",

"X",

"Y",

data1,

PlotOrientation.VERTICAL,

true,

true,

false);

final ChartPanel chartPanel2 = new ChartPanel(chart2);

chartPanel1.setPreferredSize(new java.awt.Dimension(300, 400));

chartPanel2.setPreferredSize(new java.awt.Dimension(300, 400));

panel3.add(table, BorderLayout.SOUTH);

panel3.setPreferredSize(new java.awt.Dimension(800, 800));

panel4.add(panel1);

panel4.add(chartPanel1);

panel4.add(panel3);

panel4.add(chartPanel2);

frame.add(panel4);

frame.setVisible(true);

}

}

**4. Приклад**

