Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №3

З алгоритмів та методів обчислень

*Виконав:*

Студент групи ІО-12

Нестерук Ю.О.

*Перевірив:*

Подрубайло О.О.

м. Київ

2013 р.

**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] |
| 5 (20) |  | [2, 5] |

**3. Лістинг програми:**

package lab3;

import java.util.Arrays;

public class Main {

private static final double *a* = 2.0;

private static final double *b* = 5.0;

private static final int *n* = 11;

private static final int *k* = 20;

private static double[] *x\_poli* = new double[*n*];

private static double[] *y\_poli* = new double[*n*];

private static double[] *y\_func\_grid* = new double[*n*];

private static double[] *buffer* = new double[*n*];

private static double[] *x\_func* = new double[*n*\**k*];

private static double[] *y\_func* = new double[*n*\**k*];

private static double[] *delta* = new double[*n*\**k*];

public static void main(String[] args) {

double h = (*b* - *a*) / (*n*\**k* - 1);

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

*x\_func*[i] = *a* + h\*i;

*y\_func*[i] = *function*(*x\_func*[i]);

}

h = (*b* - *a*) / (*n* - 1);

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

*x\_poli*[i] = *a* + h \* i;

*y\_func\_grid*[i] = *function*(*x\_poli*[i]);

}

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

*y\_poli*[i] = *getInterpolatedValue*(*x\_poli*[i]);

for(int i = 0; i < *n*\**k*; i++)

*delta*[i] = Math.*abs*(*y\_func*[i] - *getInterpolatedValue*(*x\_func*[i]));

Viewer viewer = new Viewer(*x\_poli*, *y\_poli*, *x\_func*, *y\_func*, *delta*);

}

public static double function(double x) {

//return Math.sin(x);

return Math.*exp*( -(x + Math.*sin*(x)) );

}

public static double getInterpolatedValue(double xx) {

*buffer* = Arrays.*copyOf*(*y\_func\_grid*, *n*);

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

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

*buffer*[j] = ((xx - *x\_poli*[i])\**buffer*[j] - (xx - *x\_poli*[j])\**buffer*[i])/(*x\_poli*[j] - *x\_poli*[i]);

return *buffer*[*n* - 1];

}

}

package lab3;

import java.awt.BorderLayout;

import javax.swing.JFrame;

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;

public class Viewer {

public Viewer(double[] x\_poli, double[] y\_poli, double[] x\_func, double[] y\_func, double[] delta) {

JFrame frame = new JFrame();

frame.setTitle("Графіки інтерполяції та похибок");

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

XYSeriesCollection data = new XYSeriesCollection();

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

for(int i = 0; i < x\_func.length; i++)

series1.add(x\_func[i], y\_func[i]);

data.addSeries(series1);

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

for(int i = 0; i < x\_poli.length; i++)

series2.add(x\_poli[i], y\_poli[i]);

data.addSeries(series2);

JFreeChart chart = ChartFactory.*createXYLineChart*(

"Графіки вихідної та інтерпольваної функцій",

"X",

"Y",

data,

PlotOrientation.*VERTICAL*,

true,

true,

false);

final ChartPanel chartPanel = new ChartPanel(chart);

chartPanel.setPreferredSize(new java.awt.Dimension(600, 600));

//frame.setContentPane(chartPanel);

frame.setLayout(new BorderLayout());

frame.add(chartPanel, BorderLayout.*WEST*);

XYSeriesCollection mistake = new XYSeriesCollection();

XYSeries deltas = new XYSeries("Похибка");

for(int i = 0; i < x\_func.length; i++)

deltas.add(x\_func[i], delta[i]);

mistake.addSeries(deltas);

JFreeChart chart2 = ChartFactory.*createXYLineChart*(

"Похибки інтерполяції у невузлових точках",

"X",

"delta",

mistake,

PlotOrientation.*VERTICAL*,

true,

true,

false);

final ChartPanel chartPanel2 = new ChartPanel(chart2);

chartPanel2.setPreferredSize(new java.awt.Dimension(600, 600));

//frame.setContentPane(chartPanel2);

frame.add(chartPanel2, BorderLayout.*EAST*);

frame.pack();

frame.setVisible(true);

}

}

**4. Аналіз результатів:**

При великій кількості вузлів різниця між значеннями функції та інтерпольованого многочлена стає дуже великою. Найоптимальніша кількість вузлів інтерполяції – 20 – тоді відхилення не перевищує .