Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

З алгоритмів та методів обчислень

*Виконав:*

Студент групи ІО-22

Бас А. В.

м. Київ

2014 р.

**1. Тема завдання:**

Закріплення знань студентів при вирішенні практичних завдань з розв’язування нелінійних рівнянь. Оволодіння методами і практичними навичками розв’язування нелінійних рівнянь на ЕОМ. Набуття умінь і навичок при програмуванні та налагодженні програм для розв’язування нелінійних рівнянь на комп'ютері.

**2. Завдання:**

Скласти програму розв’язання нелінійного рівняння, користуючись схемою алгоритму.

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Номер варіанту | Рівняння | Примітка |
| Метод половинного ділення | 1 |  | -1.35 |

**3. Лістинг програми:**

public interface Function {

public double value(double x);

}

public class HalfDivider {

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

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

HalfDividerFrame demo = new HalfDividerFrame("Метод половинного ділення");

}

}

public class HalfDividerFrame extends ApplicationFrame {

private JPanel rootPanel;

private JTextField mTextFieldLeft;

private JTextField mTextFieldRight;

private JButton mFindButton;

private ChartPanel chartPanelGraph = null;

private Function function = new Function() {

@Override

public double value(double x) {

return x \* (x \* x - 1) + 1;

}

};

private double EPS = 0.000000001D;

private double zeroValue = Double.MAX\_VALUE;

public HalfDividerFrame(String title) {

super(title);

setContentPane(rootPanel);

setPreferredSize(new Dimension(750, 550));

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

mFindButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

drawGraphic();

}

});

pack();

setVisible(true);

}

private boolean findZero(double left, double right) {

if (Math.abs(function.value(left)) < EPS) {

zeroValue = left;

return true;

}

if (right - left < 0.1 \* EPS) {

return false;

}

double m = 0.5 \* (left + right);

if (function.value(m) \* function.value(left) < 0) {

return findZero(left, m);

} else {

return findZero(m, right);

}

}

private void drawGraphic() {

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

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

XYSeries series = new XYSeries("Функція");

for (double i = a; i < b; i += 0.05) {

series.add(i, function.value(i));

}

XYSeriesCollection dataGraph = new XYSeriesCollection();

dataGraph.addSeries(series);

JFreeChart chartGraph = ChartFactory.createXYLineChart(

"Декартова система", "X", "Y", dataGraph, PlotOrientation.VERTICAL,

true, true, false);

if (findZero(a, b)) {

ValueAxis valueAxis = chartGraph.getXYPlot().getRangeAxis();

System.out.println(valueAxis.getUpperBound());

double h = 0.03 \* (valueAxis.getUpperBound() - valueAxis.getLowerBound());

double w = 0.03 \* (b - a);

System.out.println(w + " , " + h);

XYShapeAnnotation xyShapeAnnotation = new XYShapeAnnotation(

new Ellipse2D.Double(zeroValue - 0.5 \* w, function.value(zeroValue) - 0.5 \* h, w, h),

null,

null,

Color.ORANGE);

chartGraph.getXYPlot().addAnnotation(xyShapeAnnotation);

}

if (chartPanelGraph != null) {

rootPanel.remove(chartPanelGraph);

}

chartPanelGraph = new ChartPanel(chartGraph);

rootPanel.add(chartPanelGraph, BorderLayout.CENTER);

pack();

rootPanel.setVisible(true);

}

}