НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА №1

з навчальної дисципліни

«Математичні моделі динамічних систем»

Варіант № 2

|  |  |
| --- | --- |
| Тема: | Цифрові математичні моделі аеродинамічних характеристик середньомагістрального літака |

Задані аеродинамічні характеристики:  **- 3.45 - 3.23**

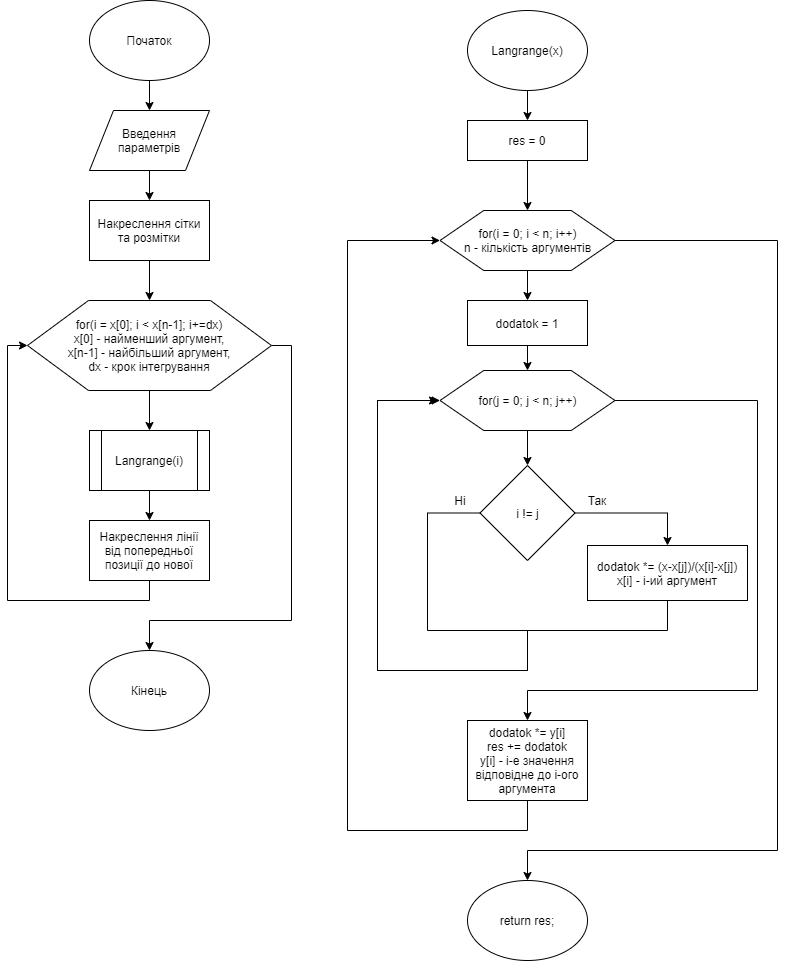
Задана інтерполяційна формула: Лагранжа

Виконавець: студент групи УС-213 Дзиговський В.І.

**Київ 2020**

Хід роботи

Схема алгоритму моделювання:



Програмна реалізація на Java:

1. Для характеристики  **- 3.45** :

public class Rabotat extends JFrame {

Graphics graph;

double maxx = 1.1, minx = 0.6;

double maxgx = 640, mingx = 100;

double maxy = 0, miny = -0.003;

double maxgy = 70, mingy = 460;

int n = 5;

double[] x = new double[] {0.6, 0.7, 0.8, 0.9, 1};

double[] y = new double[] {-0.0004, -0.0004, -0.00037, -0.00033, -0.0003};

public Rabotat(String s){

super(s);

setLayout(null);

setSize(640, 480);

setVisible(true);

this.setLocation(0, 0);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

this.setResizable(false);

graph = getGraphics();

graph.setColor(Color.white);

graph.fillRect(0, 0, 640, 480);

graph.setColor(Color.black);

graph.fillRect(0, 0, 2, 480);

graph.setFont(new Font("TimesRoman", Font.PLAIN, 20));

for(double i = 0; i > miny-0.0001; i-=0.001) {

int bub = (int)map(i, miny, maxy, mingy, maxgy);

graph.fillRect(0, bub, 640, 2);

graph.drawString(String.valueOf((double)((int)(i\*10000))/10000), 10, bub-10);

}

for(double i = minx; i < maxx; i+=maxx/10) {

int bub = (int)map(i, minx, maxx, mingx, maxgx);

graph.fillRect(bub, 0, 2, 480);

graph.drawString(String.valueOf((double)((int)(i\*10))/10), bub+10, 60);

}

graph.fillRect(4, 70, 640, 2);

int lx = (int)map(x[0], minx, maxx, mingx, maxgx), ly = (int)map(y[0], miny, maxy, mingy, maxgy);

graph.setColor(Color.red);

for (double i = x[0]; i<=x[n-1]+0.05; i+=0.0001) {

System.out.print(i + " ");

int nx = (int)map(i, minx, maxx, mingx, maxgx), ny = (int)map(Langrange(i), miny, maxy, mingy, maxgy);

graph.drawLine(lx,ly,nx,ny);

graph.drawLine(lx,ly,nx+1,ny);

graph.drawLine(lx,ly,nx,ny+1);

graph.drawLine(lx,ly+1,nx,ny);

graph.drawLine(lx+1,ly,nx,ny);

graph.drawLine(lx,ly,nx-1,ny);

graph.drawLine(lx,ly,nx,ny-1);

graph.drawLine(lx,ly-1,nx,ny);

graph.drawLine(lx-1,ly,nx,ny);

lx = nx;

ly = ny;

}

}

public double Langrange(double dx) {//Інтерполяційна формула Лагранжа

double res = 0;

for(int i = 0; i < n; i++) {//Цикл, що розраховує суму додатків

double dodatok = 1;

for(int j = 0; j < n; j++) {// Цикл, що розраховує 1 додаток

if(j != i)

dodatok \*= (dx-x[j])/(x[i]-x[j]);

}

dodatok \*= y[i];

res += dodatok;

}

System.out.println(res);

return res;

}

public double map(double input, double INmin, double INmax, double OUTmin, double OUTmax) {

return (input-INmin)\*(OUTmax-OUTmin)/(INmax-INmin)+OUTmin;

}

public static void main(String[] args) {

new Rabotat("Graphics");

}

}

1. Для характеристики  **- 3.23**:

public class Rabota2 extends JFrame{

Graphics graph;

double maxx = 1, minx = 0.4;

double maxgx = 640, mingx = 100;

double maxy = 0.12, miny = 0.08;

double maxgy = 120, mingy = 440;

int n = 7;

double[] x = new double[] {0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95};

double[] y = new double[] {0.094, 0.095, 0.1, 0.105, 0.1125, 0.122, 0.111};

public Rabota2(String s){

super(s);

setLayout(null);

setSize(640, 480);

setVisible(true);

this.setLocation(0, 0);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

this.setResizable(false);

graph = getGraphics();

graph.setColor(Color.white);

graph.fillRect(0, 0, 640, 480);

graph.setColor(Color.black);

graph.fillRect(0, 0, 2, 480);

graph.setFont(new Font("TimesRoman", Font.PLAIN, 20));

for(double i = miny; i < maxy+0.021; i+=0.02) {

int bub = (int)map(i, miny, maxy, mingy, maxgy);

graph.fillRect(0, bub, 640, 2);

graph.drawString(String.valueOf((double)((int)(i\*100))/100), 10, bub-10);

}

graph.fillRect(0, 440, 640, 2);

graph.fillRect(0, 70, 640, 2);

for(double i = minx; i < maxx; i+=maxx/10) {

int bub = (int)map(i, minx, maxx, mingx, maxgx);

graph.fillRect(bub, 0, 2, 480);

graph.drawString(String.valueOf((double)((int)Math.round(i\*100))/100), bub+10, 60);

}

int lx = (int)map(x[0], minx, maxx, mingx, maxgx), ly = (int)map(y[0], miny, maxy, mingy, maxgy);

graph.setColor(Color.red);

for (double i = x[0]; i<=x[n-1]; i+=0.0001) {

System.out.print(i + " ");

int nx = (int)map(i, minx, maxx, mingx, maxgx), ny = (int)map(Langrange(i), miny, maxy, mingy, maxgy);

graph.drawLine(lx,ly,nx,ny);

graph.drawLine(lx,ly,nx+1,ny);

graph.drawLine(lx,ly,nx,ny+1);

graph.drawLine(lx,ly+1,nx,ny);

graph.drawLine(lx+1,ly,nx,ny);

graph.drawLine(lx,ly,nx-1,ny);

graph.drawLine(lx,ly,nx,ny-1);

graph.drawLine(lx,ly-1,nx,ny);

graph.drawLine(lx-1,ly,nx,ny);

lx = nx;

ly = ny;

}

}

public double Langrange(double dx) {

double res = 0;

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

double dodatok = 1;

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

if(j != i)

dodatok \*= (dx-x[j])/(x[i]-x[j]);

}

dodatok \*= y[i];

res += dodatok;

}

System.out.println(res);

return res;

}

public double map(double input, double INmin, double INmax, double OUTmin, double OUTmax) {

return (input-INmin)\*(OUTmax-OUTmin)/(INmax-INmin)+OUTmin;

}

public static void main(String[] args) {

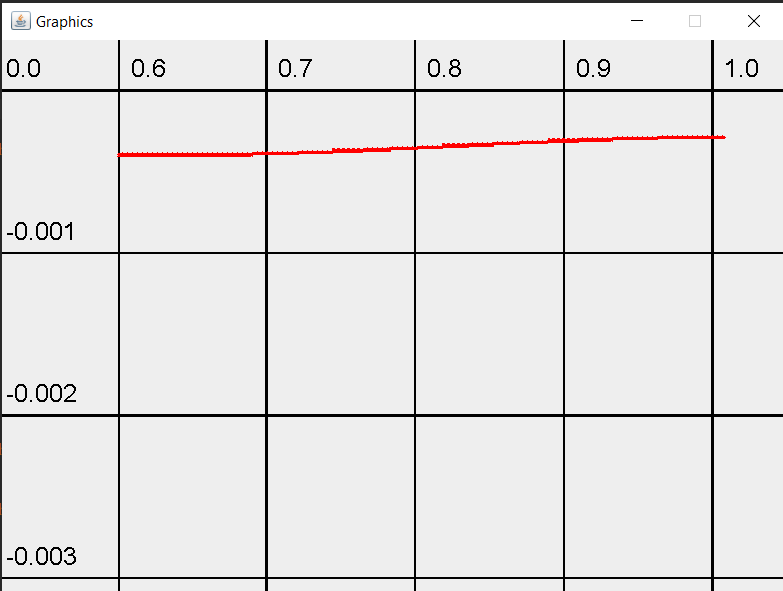
new Rabota2("Graphics");

}

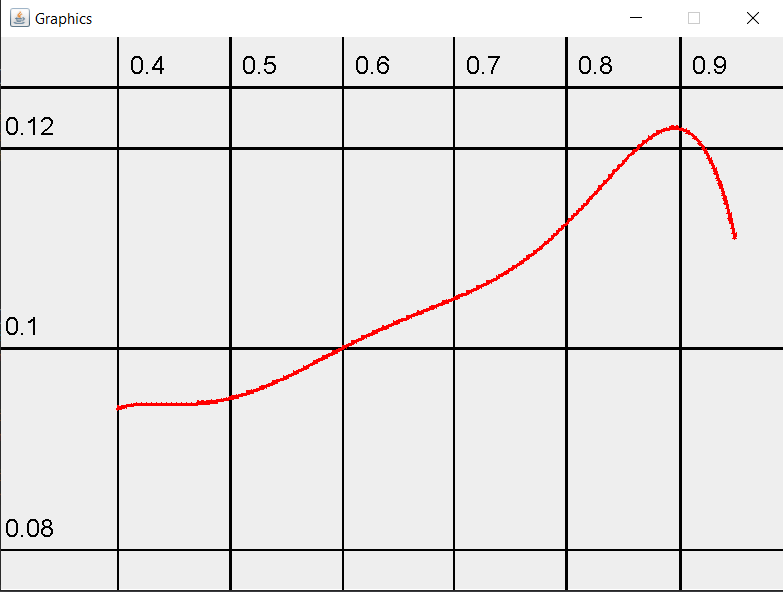
}

Результати:

Графік  **- 3.45**



Графік  **- 3.23**



Висновки: Було досліджено та вивчено інтерполяційні формули для моделювання аеродинамічних характеристик. Інтерполяційна формула Лагранжа досить точна та простіша для великої кількості аргументів ніж канонічна, але многочлен має коливальні властивості, тому більш доречно буде використовувати інтерполяційну формулу Ньютона. Створений алгоритм дозволяє використовувати формулу Лагранжа для будь якої кількості аргументів.