МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное

образовательное учреждение

высшего образования

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Отчёт по лабораторной работе №3

по дисциплине «Математические модели в естествознании и методы их исследования»

Исследование движения планет

Выполнил: студент группы ФИб-4302-51-00 \_\_\_\_\_\_\_\_\_\_\_ / Савин Д. А/

Проверил: к. б. н., доцент каф. ПМиИ \_\_\_\_\_\_\_\_\_\_\_/ Чупраков П. Г./

Киров 2021

**Цель работы:** приобретение навыков самостоятельного решения прикладной задачи математического моделирования “Исследование движения планет”.

**План:**

1. Выписать уравнения системы для всех случаев со своим вариантом (указан в таблице коэффициентов и значений начальных величин).
2. Аппроксимировать полученные уравнения с использованием методов Рунге-Кутты.
3. Построить модели по полученным результатам.

Проанализировать результаты вычислительных экспериментов, сделать выводы.

**Задание**

1. Решить систему уравнений для двух тел (1).

2. Построить траектории движения планет на плоскости.

3. Решить систему уравнений для трёх тел (2).

4. Построить траектории движения планет на плоскости.

5. Решить систему уравнений для трёх тел (3). Начальные условия для третьей координаты задать самостоятельно, например, равные нулю.

6. Построить траектории движения планет в пространстве.

7. Подобрать начальные данные и массу тел для получения устойчивой системы трёх тел.

Уравнения:

X:

Y:

Z:

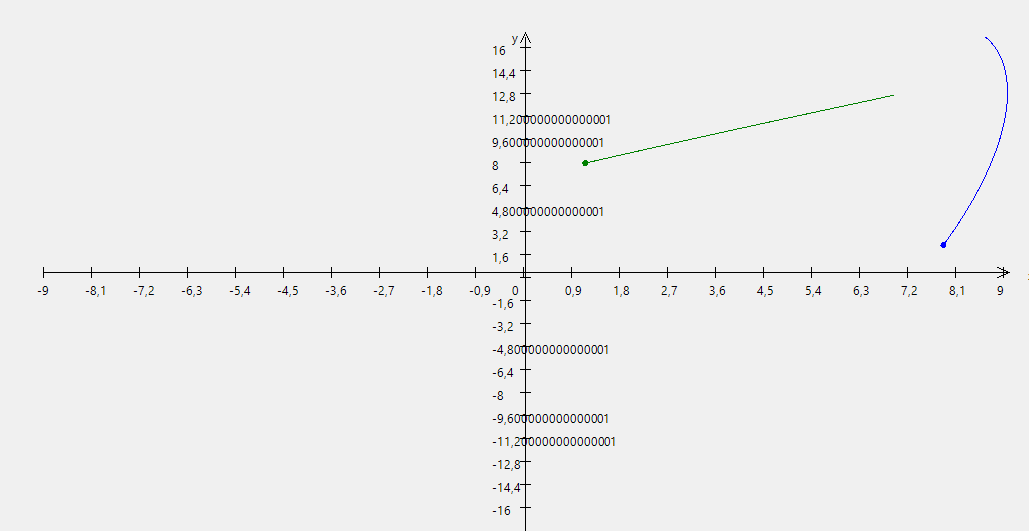
Таблица вариантов:

Изображение выглядит как текст, белый

Автоматически созданное описание

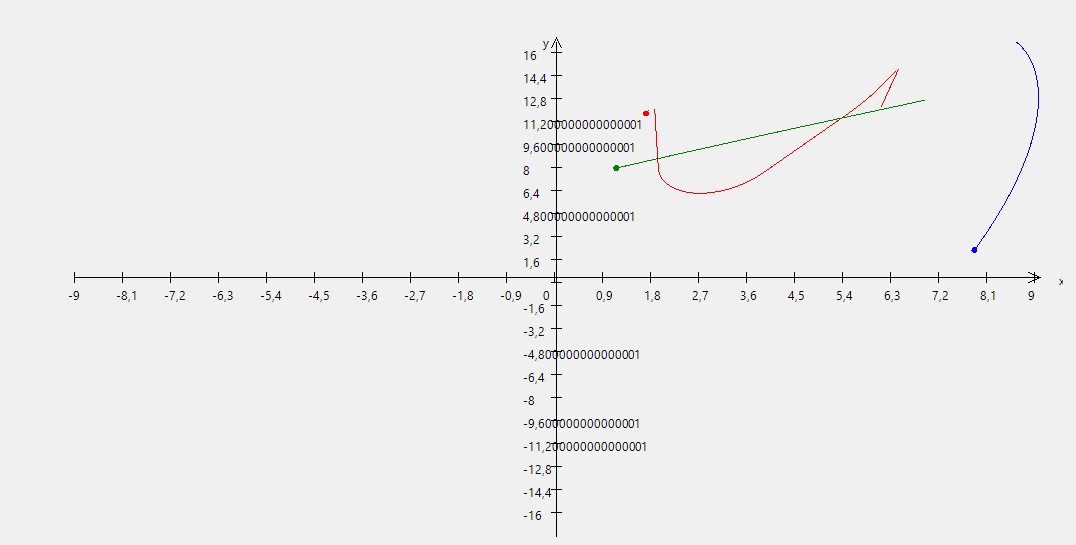
Вариант 9.

1. Для двух тел (2 координаты)



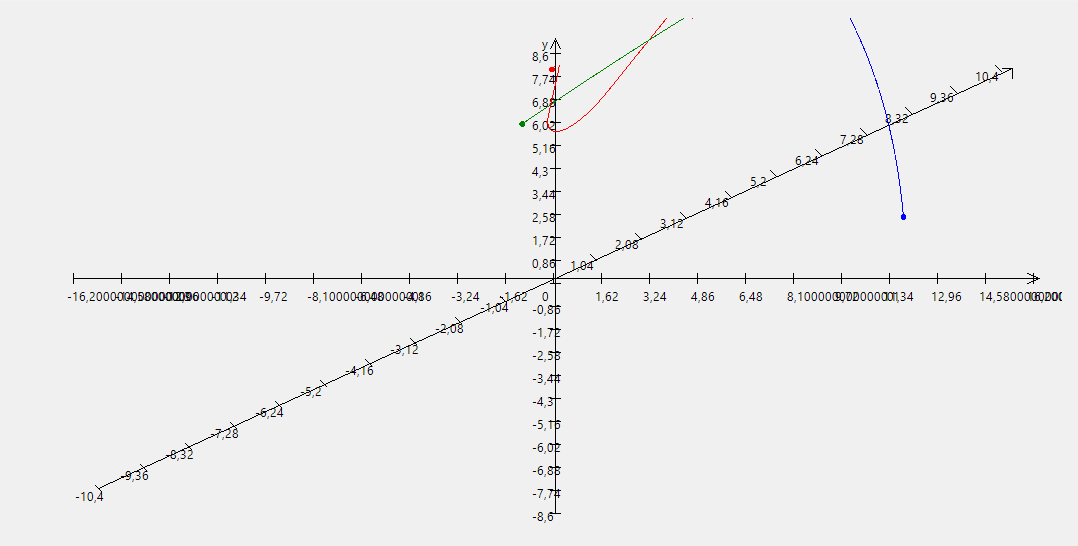
Более тяжелое зеленое тело движется с постоянной скоростью, синее тело, приближаясь к зеленому, начинает поворачиваться.

1. Для трех тел (2 координаты)



Самое легкое красное тело начинает притягиваться к зеленому телу и лететь за ним

1. Для трех тел (3 координаты)



Синее и красное тело стремятся к зеленому телу, которое движется с постоянной скоростью. Синее тело, приближаясь к зеленому, начинает поворачиваться, после того как сближается с ним.

# Заключение

В результате выполнения лабораторной работы были приобретены навыки решения прикладной задачи математического моделирования «Исследование движения планет».

**Листинг программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinFormsApp1

{

public partial class Form1 : Form

{

double[] M;

double[,] C;

int N, K;

public Form1()

{

InitializeComponent();

}

private double Function0(double[,] x, double[,] y, int indexN, int indexK)

{

return y[indexN, indexK];

}

private double Function1(double[,] x, double[,] y, int indexN, int indexK)

{

double[] f = new double[N];

double[] r = new double[N];

double result = 0;

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

{

if (i != indexN)

{

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

{

r[i] += Math.Pow(x[indexN, j] - x[i, j], 2);

}

f[i] = M[indexN] \* M[i] / Math.Pow(Math.Sqrt(r[i]), 3) \* (x[indexN, indexK] - x[i, indexK]) \* -1;

if (r[i] < 2)

{

f[i] = 0;

}

result += f[i];

}

}

result /= M[indexN];

return result;

}

private void BeginToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void toolStrip2\_Click(object sender, EventArgs e)

{

try

{

Stream myStream;

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

if ((myStream = openFileDialog1.OpenFile()) != null)

{

StreamReader streamReader = new StreamReader(myStream);

string line;

int i = 0;

line = streamReader.ReadLine();

N = int.Parse(line);

line = streamReader.ReadLine();

K = int.Parse(line);

C = new double[3, 7];

while (!streamReader.EndOfStream)

{

line = streamReader.ReadLine();

string[] s = line.Split(' ');

for (int j = 0; j < s.Length; j++)

{

C[i, j] = double.Parse(s[j]);

}

i++;

}

streamReader.Close();

}

myStream.Close();

}

double[,] coordinates = new double[N, K];

double[,] speeds = new double[N, K];

double t = 0;

double h = 0.00001;

M = new double[N];

List<double[,]> coordinateValues = new List<double[,]>();

List<double[,]> speedValues = new List<double[,]>();

List<double> tValue = new List<double>();

double maxt = 0;

double[] maxCoordinate = new double[K];

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

{

M[i] = C[i, 0];

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

{

coordinates[i, j] = C[i, 1 + j];

}

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

{

speeds[i, j] = C[i, 4 + j];

}

}

for (int i = 0; t < 1000; t += 400 \* h, i++)

{

coordinateValues.Add(new double[N, K]);

speedValues.Add(new double[N, K]);

for (int n = 0; n < N; n++)

{

for (int k = 0; k < K; k++)

{

coordinateValues[i][n, k] = coordinates[n, k];

speedValues[i][n, k] = speeds[n, k];

}

}

for (int n = 0; n < N; n++)

{

for (int k = 0; k < K; k++)

{

maxCoordinate[k] = Math.Max(maxCoordinate[k], Math.Abs(coordinates[n, k]));

}

}

tValue.Add(t);

Runge\_kutt4(coordinates, speeds, h, t, out coordinates, out speeds);

maxt = t;

}

int X0, Y0;

X0 = 25;

Y0 = pictureBox1.Height - 25;

Graphics graph = pictureBox1.CreateGraphics();

graph.Clear(BackColor);

Pen myPen = new Pen(Brushes.Black, 1);

graph.DrawLine(myPen, X0, (Y0 + 25) / 2, pictureBox1.Width - 25, (Y0 + 25) / 2);

graph.DrawLine(myPen, (pictureBox1.Width) / 2, Y0 + 25, (pictureBox1.Width) / 2, 25);

int x = (pictureBox1.Width - 50) / 20;

int y = (pictureBox1.Height - 50) / 20;

int Y = Y0;

int X = X0;

for (int i = 0; i <= 20; i++)

{

graph.DrawLine(myPen, (pictureBox1.Width) / 2 - 5, Y, (pictureBox1.Width) / 2 + 5, Y);

graph.DrawLine(myPen, X, (Y0 + 25) / 2 - 5, X, (Y0 + 25) / 2 + 5);

if (i != 10)

graph.DrawString((Math.Round(maxCoordinate[1] / 10, 1) \* (i - 10)).ToString(), Font, Brushes.Black, (float)(X - 8), (Y0 + 25) / 2 + 10);

if (i != 10)

graph.DrawString((Math.Round(maxCoordinate[0] / 10, 1) \* (i - 10)).ToString(), Font, Brushes.Black, (pictureBox1.Width) / 2 - 35, (float)(Y - 5));

if (i == 10)

graph.DrawString((Math.Round(maxCoordinate[1] / 10, 1) \* (i - 10)).ToString(), Font, Brushes.Black, (float)(X - 13), (Y0 + 25) / 2 + 10);

X += x;

Y -= y;

}

graph.DrawString("x", Font, Brushes.Black, (float)(X - 25), (Y0 + 25) / 2 - 5);

graph.DrawString("y", Font, Brushes.Black, (pictureBox1.Width) / 2 - 15, (float)(Y + 5));

graph.DrawLine(myPen, pictureBox1.Width - 35, (Y0 + 25) / 2 + 5, pictureBox1.Width - 23, (Y0 + 25) / 2);

graph.DrawLine(myPen, pictureBox1.Width - 35, (Y0 + 25) / 2 - 5, pictureBox1.Width - 23, (Y0 + 25) / 2);

graph.DrawLine(myPen, (pictureBox1.Width) / 2 - 5, 30, (pictureBox1.Width) / 2, 20);

graph.DrawLine(myPen, (pictureBox1.Width) / 2 + 5, 30, (pictureBox1.Width) / 2, 20);

Pen[] myPens1 = new Pen[N];

Color[] colors = new Color[3];

colors[0] = Color.Green;

colors[1] = Color.Blue;

colors[2] = Color.Red;

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

{

graph.FillEllipse(new SolidBrush(colors[i]), (pictureBox1.Width) / 2 + (float)(coordinateValues[0][i, 0] / maxCoordinate[0] \* (pictureBox1.Width / 2 - 25)) - 3, (Y0 + 25) / 2 - (float)(coordinateValues[0][i, 1] / maxCoordinate[1] \* (pictureBox1.Height / 2 - 25)) - 3, 6, 6);

myPens1[i] = new Pen(colors[i], 1);

}

for (int n = N - 1; n > -1; n--)

{

for (int i = 0; i < coordinateValues.Count - 1; i++)

{

graph.DrawLine(myPens1[n], ((float)pictureBox1.Width) / 2 + (float)(coordinateValues[i][n, 0] / maxCoordinate[0] \* ((float)pictureBox1.Width / 2 - 25)), ((float)Y0 + 25) / 2 - (float)(coordinateValues[i][n, 1] / maxCoordinate[1] \* ((float)pictureBox1.Height / 2 - 25)),

((float)pictureBox1.Width) / 2 + (float)(coordinateValues[i + 1][n, 0] / maxCoordinate[0] \* ((float)pictureBox1.Width / 2 - 25)), ((float)Y0 + 25) / 2 - (float)(coordinateValues[i + 1][n, 1] / maxCoordinate[1] \* ((float)pictureBox1.Height / 2 - 25)));

}

}

if (K > 2)

{

graph = pictureBox1.CreateGraphics();

graph.Clear(BackColor);

X0 = 25;

Y0 = pictureBox1.Height - 25;

int Z0X = X0 + 25;

int Z0Y = Y0 - 25;

graph.DrawLine(myPen, X0, (Y0 + 25) / 2, pictureBox1.Width - 25, (Y0 + 25) / 2);

graph.DrawLine(myPen, (pictureBox1.Width) / 2, Y0, (pictureBox1.Width) / 2, 23);

graph.DrawLine(myPen, X0 + 25, Y0 - 25, pictureBox1.Width - 50, 50);

x = (pictureBox1.Width - 50) / 20;

y = (pictureBox1.Height - 50) / 20;

float zx = (pictureBox1.Width - 100) / 20;

float zy = (pictureBox1.Height - 100) / 20;

Y = Y0;

X = X0;

float ZX = Z0X;

float ZY = Z0Y;

for (int i = 0; i <= 20; i++)

{

graph.DrawLine(myPen, (pictureBox1.Width) / 2 - 5, Y, (pictureBox1.Width) / 2 + 5, Y);

graph.DrawLine(myPen, X, (Y0 + 25) / 2 - 5, X, (Y0 + 25) / 2 + 5);

if (i != 10)

graph.DrawString((Math.Round(maxCoordinate[0] / 10, 2) \* (i - 10)).ToString(), Font, Brushes.Black, (float)(X - 8), (Y0 + 25) / 2 + 10);

if (i != 10)

graph.DrawString((Math.Round(maxCoordinate[1] / 10, 2) \* (i - 10)).ToString(), Font, Brushes.Black, (pictureBox1.Width) / 2 - 25, (float)(Y - 5));

if (i != 10 && K > 2)

{

graph.DrawLine(myPen, ZX - 3, ZY - 3, ZX + 3, ZY + 3);

graph.DrawString((Math.Round(maxCoordinate[2] / 10, 2) \* (i - 10)).ToString(), Font, Brushes.Black, (float)(ZX - 25), (float)(ZY));

}

if (i == 10)

graph.DrawString((Math.Round(maxCoordinate[0] / 10, 2) \* (i - 10)).ToString(), Font, Brushes.Black, (float)(X - 13), (Y0 + 25) / 2 + 10);

X += x;

Y -= y;

ZX += zx;

ZY -= zy;

}

graph.DrawString("x", Font, Brushes.Black, (float)(X - 20), (Y0 + 25) / 2 - 5);

graph.DrawString("y", Font, Brushes.Black, (pictureBox1.Width) / 2 - 15, (float)(Y + 5));

graph.DrawLine(myPen, pictureBox1.Width - 35, (Y0 + 25) / 2 + 5, pictureBox1.Width - 23, (Y0 + 25) / 2);

graph.DrawLine(myPen, pictureBox1.Width - 35, (Y0 + 25) / 2 - 5, pictureBox1.Width - 23, (Y0 + 25) / 2);

graph.DrawLine(myPen, (pictureBox1.Width) / 2 - 5, 30, (pictureBox1.Width) / 2, 20);

graph.DrawLine(myPen, (pictureBox1.Width) / 2 + 5, 30, (pictureBox1.Width) / 2, 20);

graph.DrawLine(myPen, (pictureBox1.Width) - 50, 50, (pictureBox1.Width) - 50, 60);

graph.DrawLine(myPen, (pictureBox1.Width) - 50, 50, (pictureBox1.Width) - 60, 50);

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

{

graph.FillEllipse(new SolidBrush(colors[i]), (pictureBox1.Width) / 2 + (float)((coordinateValues[0][i, 0] / maxCoordinate[0] - coordinateValues[i][i, 2] / maxCoordinate[2] / 2) \* (pictureBox1.Width / 2 - 25)) - 3, (Y0 + 25) / 2 - (float)((coordinateValues[0][i, 1] / maxCoordinate[1] + coordinateValues[i][i, 2] / maxCoordinate[2] / 2) \* (pictureBox1.Height / 2 - 25)) - 3, 6, 6);

}

for (int i = 0; i < coordinateValues.Count - 1; i++)

{

for (int n = N - 1; n > -1; n--)

{

graph.DrawLine(myPens1[n], (pictureBox1.Width) / 2 + (float)((coordinateValues[i][n, 0] / maxCoordinate[0] - coordinateValues[i][n, 2] / maxCoordinate[2] / 2) \* (pictureBox1.Width / 2 - 25)), (Y0 + 25) / 2 - (float)((coordinateValues[i][n, 1] / maxCoordinate[1] + coordinateValues[i][n, 2] / maxCoordinate[2] / 2) \* (pictureBox1.Height / 2 - 25)),

(pictureBox1.Width) / 2 + (float)((coordinateValues[i + 1][n, 0] / maxCoordinate[0] - coordinateValues[i + 1][n, 2] / maxCoordinate[2] / 2) \* (pictureBox1.Width / 2 - 25)), (Y0 + 25) / 2 - (float)((coordinateValues[i + 1][n, 1] / maxCoordinate[1] + coordinateValues[i][n, 2] / maxCoordinate[2] / 2) \* (pictureBox1.Height / 2 - 25)));

}

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Runge\_kutt4(double[,] x, double[,] y, double h, double t, out double[,] x1, out double[,] y1)

{

double[,,] l = new double[4, N, K];

double[,,] m = new double[4, N, K];

x1 = new double[N, K];

y1 = new double[N, K];

for (int i = 0; i < 4; i++)

{

for (int n = 0; n < N; n++)

{

for (int k = 0; k < K; k++)

{

l[i, n, k] = h \* Function0(x, y, n, k);

m[i, n, k] = h \* Function1(x, y, n, k);

}

}

}

for (int n = 0; n < N; n++)

{

for (int k = 0; k < K; k++)

{

x1[n, k] = x[n, k] + (l[0, n, k] + 2 \* l[1, n, k] + 2 \* l[2, n, k] + l[3, n, k]) / 6;

y1[n, k] = y[n, k] + (m[0, n, k] + 2 \* m[1, n, k] + 2 \* m[2, n, k] + m[3, n, k]) / 6;

}

}

}

}

}