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 №6

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

Модель химической кинетики «Брюсселятор»

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

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

Киров 2021

**Цель лабораторной работы:** провести исследование автоколебаний в системе химических реакций на примере модели «Брюсселятор».

**Задание:**

1. Ознакомиться с теоретическим материалом.

2. Разработать программу, реализующую точечную модель «Брюсселятор».

3. Провести исследование модели: исследовать точечную модель при различных соотношениях между параметрами. Построить фазовый портрет и графики зависимости концентраций веществ от времени для двух случаев: ∙

автоколебания;

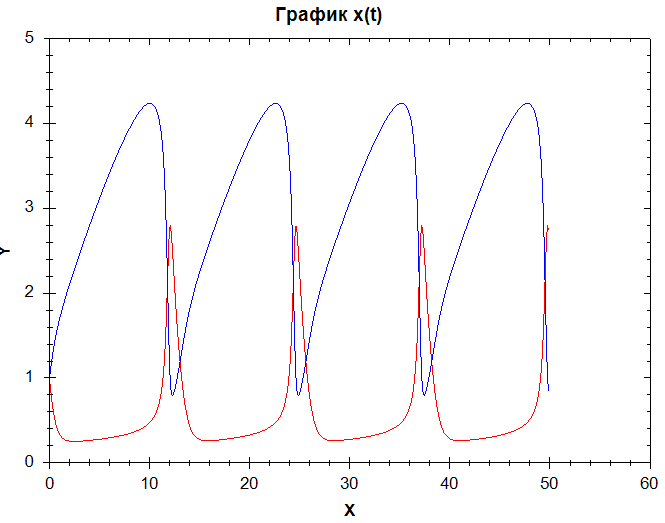
затухающие колебания.

4. Оформить отчёт по результатам исследований.

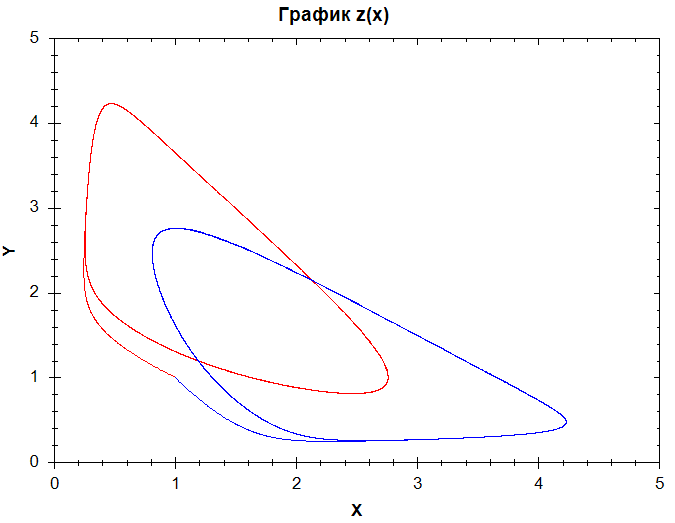
1. Автоколебания

При a = 1, b = 2, x = 1, y = 1.

Графики зависимости концентрации веществ от времени (синий – y(t), красный – x(t)).



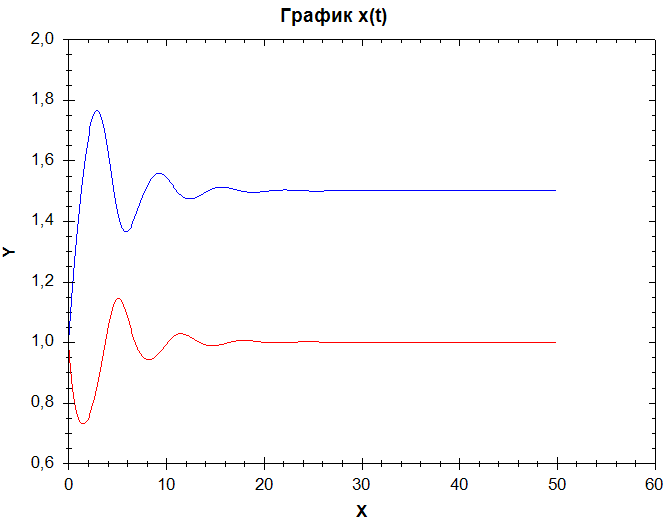
Фазовые портреты (синий – y(x), красный x(y))



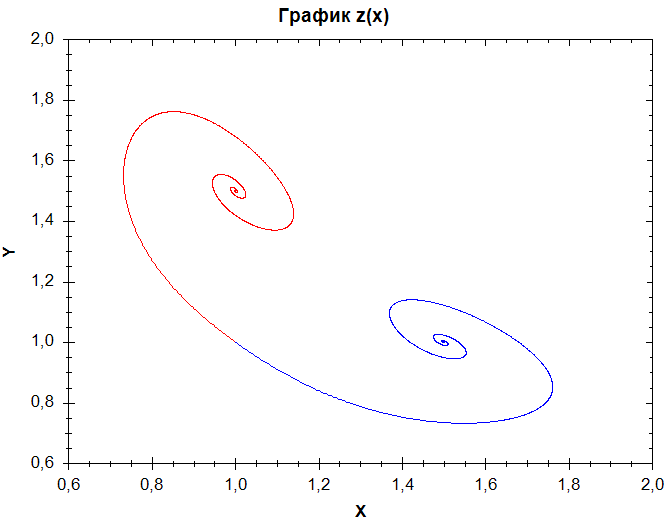
1. Затухающие колебания

При a = 1, b = 1.5, x = 1, y = 1

Графики зависимости концентрации веществ от времени (синий – y(t), красный – x(t)).



Фазовые портреты (синий – y(x), красный x(y))



**Вывод:** в ходе лабораторной работы было проведено исследование автоколебаний и затухающих колебаний в системе химических реакций «Брюсселятор», были построены графики зависимости концентраций вещества от времени и фазовые портреты для обоих веществ.

**Листинг программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using ZedGraph;

namespace lab6\_graphics\_

{

public partial class Form1 : Form

{

public static List<double> listOfPointsX = new List<double>();

public static List<double> listOfPointsY = new List<double>();

public static List<double> listOfPointsT = new List<double>();

public Form1()

{

InitializeComponent();

DrawGraph();

}

static double Function\_f(double a, double b, double x, double y)

{

return a - (b + 1) \* x + Math.Pow(x, 2) \* y;

}

static double Function\_g(double b, double x, double y)

{

return b \* x - Math.Pow(x, 2) \* y;

}

static void Calculation()

{

double a = 1, b = 1.5, x = 1, y = 1, t = 0;

double h = 0.1,

k1, k2, k3, k4, l1, l2, l3, l4;

Console.WriteLine("\tX1\t\t\t\tX2");

for (int i = 0; i < 500; i++)

{

listOfPointsX.Add(x);

listOfPointsY.Add(y);

listOfPointsT.Add(t);

k1 = h \* Function\_f(a, b, x, y);

l1 = h \* Function\_g(b, x, y);

k2 = h \* Function\_f(a, b, x + k1 / 2, y + l1 / 2);

l2 = h \* Function\_g(b, x + k1 / 2, y + l1 / 2);

k3 = h \* Function\_f(a, b, x + k2 / 2, y + l2 / 2);

l3 = h \* Function\_g(b, x + k2 / 2, y + l2 / 2);

k4 = h \* Function\_f(a, b, x + k3 / 2, y + l3 / 2);

l4 = h \* Function\_g(b, x + k3 / 2, y + l3 / 2);

double fRez = Function\_f(a, b, x, y);

double gRez = Function\_g(b, x, y);

Console.WriteLine(Convert.ToString("\t" + fRez + "\t\t\t" + gRez));

//Console.WriteLine(Convert.ToString("\t" + gRez + "\t\t"));

x += 1 / 6.0 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

y += 1 / 6.0 \* (l1 + 2 \* l2 + 2 \* l3 + l4);

t += h;

}

}

private void DrawGraph()

{

GraphPane pane = zedGraph.GraphPane;

pane.CurveList.Clear();

pane.Title.Text = "График x(t)";

pane.YAxis.Title.Text = "Y";

pane.XAxis.Title.Text = "X";

PointPairList list = new PointPairList();

PointPairList list2 = new PointPairList();

Calculation();

for (int i = 0; i < listOfPointsX.Count; i++)

{

list.Add(listOfPointsT.ElementAt(i), listOfPointsY.ElementAt(i));

list2.Add(listOfPointsT.ElementAt(i), listOfPointsX.ElementAt(i));

}

LineItem myCurve = pane.AddCurve("", list, Color.Blue, SymbolType.None);

LineItem myCurve2 = pane.AddCurve("", list2, Color.Red, SymbolType.None);

zedGraph.AxisChange();

zedGraph.Invalidate();

}

}

}

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Windows.Forms;

using ZedGraph;

namespace lab6\_graphics\_

{

public partial class Form2 : Form

{

public static List<double> listOfPointsX = new List<double>();

public static List<double> listOfPointsY = new List<double>();

public static List<double> listOfPointsT = new List<double>();

public Form2()

{

InitializeComponent();

DrawGraph();

}

static double Function\_f(double a, double b, double x, double y)

{

return a - (b + 1) \* x + Math.Pow(x, 2) \* y;

}

static double Function\_g(double b, double x, double y)

{

return b \* x - Math.Pow(x, 2) \* y;

}

static void Calculation()

{

double a = 1, b = 1.5, x = 1, y = 1, t = 0;

double h = 0.01,

k1, k2, k3, k4, l1, l2, l3, l4;

Console.WriteLine("\tX1\t\t\t\tX2");

for (int i = 0; i < 5000; i++)

{

listOfPointsX.Add(x);

listOfPointsY.Add(y);

listOfPointsT.Add(t);

k1 = h \* Function\_f(a, b, x, y);

l1 = h \* Function\_g(b, x, y);

k2 = h \* Function\_f(a, b, x + k1 / 2, y + l1 / 2);

l2 = h \* Function\_g(b, x + k1 / 2, y + l1 / 2);

k3 = h \* Function\_f(a, b, x + k2 / 2, y + l2 / 2);

l3 = h \* Function\_g(b, x + k2 / 2, y + l2 / 2);

k4 = h \* Function\_f(a, b, x + k3 / 2, y + l3 / 2);

l4 = h \* Function\_g(b, x + k3 / 2, y + l3 / 2);

double fRez = Function\_f(a, b, x, y);

double gRez = Function\_g(b, x, y);

x += 1 / 6.0 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

y += 1 / 6.0 \* (l1 + 2 \* l2 + 2 \* l3 + l4);

t += h;

}

}

private void DrawGraph()

{

GraphPane pane = zedGraph.GraphPane;

pane.CurveList.Clear();

pane.Title.Text = "График z(x)";

pane.YAxis.Title.Text = "Y";

pane.XAxis.Title.Text = "X";

PointPairList list = new PointPairList();

PointPairList list2 = new PointPairList();

Calculation();

for (int i = 0; i < listOfPointsX.Count; i++)

{

list.Add(listOfPointsY.ElementAt(i), listOfPointsX.ElementAt(i));

list2.Add(listOfPointsX.ElementAt(i), listOfPointsY.ElementAt(i));

}

LineItem myCurve = pane.AddCurve("", list, Color.Blue, SymbolType.None);

LineItem myCurve2 = pane.AddCurve("", list2, Color.Red, SymbolType.None);

zedGraph.AxisChange();

zedGraph.Invalidate();

}

}

}