Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

Тамбовский государственный технический университет

Кафедра

**КУРСОВАЯ РАБОТА**

по дисциплине «Модели и методы анализа проектных решений»

16 вариант

Выполнил: студент гр. –41

Д. Ю.

Проверил: Ю.В.

Тамбов

**Содержание**

1. Введение 3

2. Объект проектирования 3

3. Математическая модель заданного объекта 3

4. Метод оптимизации 6

5. Оптимизация объекта проектирования 9

6. Проверка работоспособности спроектированного объекта 9

7. Выводы 11

Литература 12

Приложение 13

# Введение

***Задание:***

Найти методом локальных вариаций по пяти сечениям методом дихотомии функцию распределения температуры теплоносителя по длине трубчатого реактора получения фталевого ангидрида, при которой длина трубы реактора будет минимальной:

α1 = α2 = α3 = 2; А1 = 8000; Е1 = 72600 Дж/моль; A2 = 4000;

E2 = 77 000 Дж/моль; А3 = 3\*105; Е3 = 87 000 Дж/моль;

Kt = 500 Вт/м2град; u = 0,1 м/с; Твх = 650 К;

Q1 = 1,47\*107 Дж/моль; Q2 = 1,4\*108 Дж/моль;

Q3 = 19,6\*106 Дж/моль; Сt = 1100 Дж/кг·град; ρ = 1,4 кг/м3;

D = 0,025 м; Снаф.вх.= 0,1; СО2вв = 0,2; Сфт.анг.вых. ≥ 4%;

260 ≤ Tt ≤ 300°С.

При имитационном моделировании выходная координата Cфт.анг.вых, входная координата Снаф. вх.; M0 = 1 моль/м3; K = 0,4exp(–0,071dt).

# Объект проектирования

Объектом проектирования является реактор для получения фталевого ангидрида.

Реактор по своему типу является трубчатым, поэтому используют пренебрежения характерные для математической модели «Идеальное вытеснение».

# Математическая модель заданного объекта

Для описания рассматриваемого объекта целесообразно использовать математическую модель «Идеальное вытеснение», так как среда в реакторе движется без смешивания каждого слоя с впереди идущими и последующими слоями, без перемешивания по диаметру трубы:

В реакторе происходит экзотермическая бимолекулярная реакция:

Где С – концентрация вещества, – время, l – длина трубы, u – скорость движения среды, S – стехиометрический коэффициент, k – скорость химической реакции.

Таким образом, изменение концентрации веществ происходит за счет:

1. идеального вытеснения;
2. протекающих химических реакций.

Тогда математическая модель будет выглядеть следующим образом:

С1 – концентрация нафталина

C2 – концентрация кислорода

Входные концентрации исходных веществ, даны в процентах, их необходимо перевести в моль/м3:

ρ – средняя плотность реакционной среды;

μ – молярная масса вещества;

μ1=0.128 кг/моль; – молярная масса нафталина

μ2=0.032 кг/моль; – молярная масса кислорода

μ3= 0.148 кг/моль – молярная масса фталиевого ангидрида.

Для решения дифференциальных уравнений, входящих в систему использовался метод Эйлера.

***Проверка адекватности модели:***

Подавая на вход модели статики концентрацию нафталина и оксида кислорода, видим что концентрация выходного вещества постепенно растет доходя до заданного порога в 4%, определив минимальную длину равной 2. 8 метра при температуре 286 градусов цельси. Следовательно, математическая модель является адекватной. (Рисунок 1)

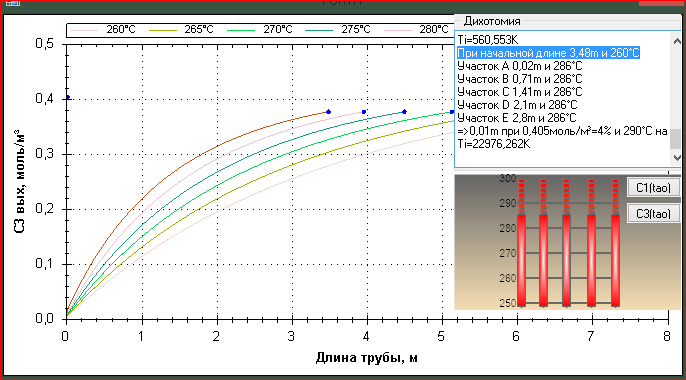


Рисунок Длина трубы 3.48 м

# Метод оптимизации

Минимальную длину трубы будем искать используя метод локальных вариации по пяти сечениям методом дихотомии.

В данном методе задается начальная температура всего участка трубы, разбивая на пять частей, проверяем каждый участок в отдельности, отвечая на вопрос – уменьшилась ли длина при повышении температуры отдельно взятого участка.

***Алгоритм:***

1. Найденную длину делим на пять равных отрезков:
2. Задаем начальнйю температуру равной 260 градусов цельси, просчитываем концентрацию выходного вещества для заданого порога в 4%
3. Если длина уменьшилась, тоделаем повышение на текущем участке, пока не получим результат, больщий предидущего по длине.
4. Если результат отрицателен, необходимо перейти к следующему отрезку и повторить действия начиная с пункта 2.
5. Действия повторяют до тех пор, пока получаемая в результате длина трубы не изменится на порядок, отличный от предыдущих.

***Проверка работоспособности алгоритма:***

Для проверки задаем начальную температуру в 300 градусов цельси, рассматриваем длину равной 2.1 метра. Минимальный участок дает результат в 1.3 метра при достижении концентрации выходного вещества равным 4%. Алгоритм удовлетворяет условиям поиска минимума длины.

***Блок-схема метода оптимизации***



**5. Оптимизация объекта проектирования**

При оптимизации объекта проектирования в качестве критерия выступает длина трубы реактора, а в качестве варьируемых параметров – температура в реакторе.

В результате выполнения программы были получены следующие данные: ***минимальная длинна трубы реактора, при которой концентрация конечного вещества Cфтал.анг.вых. =< 4 %, равна 2.8м***

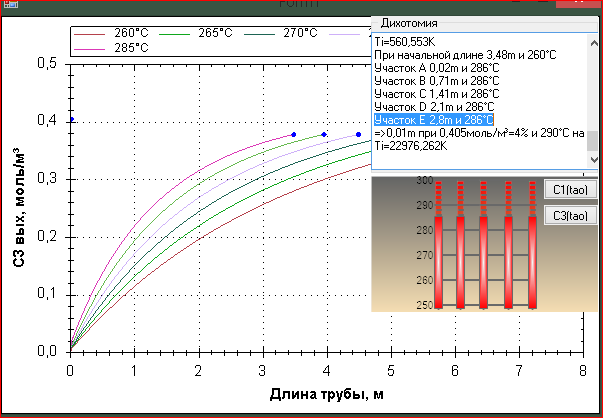


Рисунок Минимум равен 2.8 м и температура участка 286\*С

**6. Проверка работоспособности спроектированного объекта**

Проверку работоспособности спроектированного объекта проводится методом имитационного моделирования.

В качестве выходной координаты принимается – Cфтал.анг.вых, входной координаты – Снафт.вх.

Построим генератор случайного процесса со следующими характеристиками:

Математическое ожидание: M0=1 моль/;

Корреляционная функция: K=0.4exp(-0,071dt);

тогда дисперсия σ2=0.4,

параметр аппроксимации α=0,071.

Для построения генератора случайного процесса воспользуемсяприближенной функцией вида:



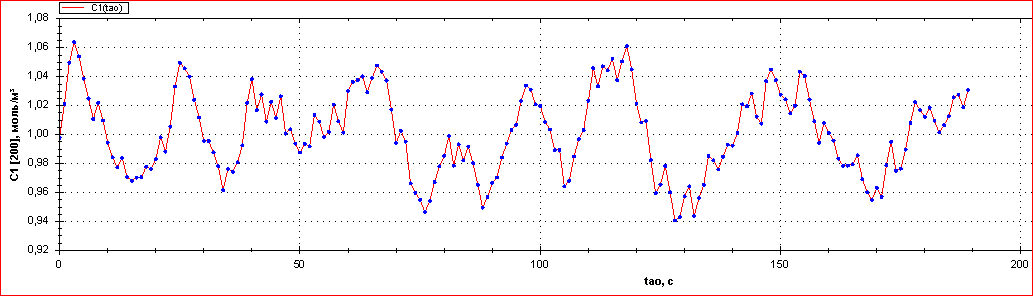


Рисунок Значения, полученные генератором в количестве 200 чисел

Выходная координата лежит в диапазоне [0.94;1.061], не выходя за его пределы. Следовательно, объект является устойчивым и работоспособным.

**7. Выводы**

В процессе выполнения курсовой работы был изучен ректор получения фталевого ангидрида, выявлены его особенности, протекающие в нем процессы, построена математическая модель этого объекта.

Далее, методом дихотомии была проведена оптимизация объекта по полученной математической модели, выявлены оптимальные параметры (концентрация в реакторе ангидрида и температура), при которых достигается минимальная длина трубы реактора.

Методом имитационного моделирования была осуществлена проверка математической модели. Эксперимент показал, что колебания выходной величина находятся в заданных пределах, следовательно, математическая модель работоспособна.

**Литература**

1. Построение математических моделей химико-технологических объектов \Дудников Е.Г., Балакирев В.С., Кривсунов В.Н., Цирлин А.М. – Л.: Химия. 1970. - 312 с.

2. Шеннон Р. Имитационное моделирование систем - искусство и наука. – М.: Мир, 1975. - 502 с.

3. Закгейм А.Ю. Введение в моделирование химико-технологических процессов. М.: Химия, 1982. - 288 с.

4. Общая химическая технология / Под ред И. П. Мухленова. В 2 ч. – М.: Высшая школа, ч. 2, 1984. - 264 с.

**Приложение**

**Описание программы**

***Общие сведения***

Программа называется Razbor, написана на языке программирования C#, в среде разработки Mono.

***Функциональное назначение программы***

Программа решает систему уравнений математической модели, построенной по заданному объекту; оптимизирует объект по критерию с заданными ограничениями и условиями, а также проверяет модель на работоспособность методом имитационного моделирования.

***Описание логической структуры программы***

Программа состоит из 2 частей:

1. Решение системы уравнений математической модели статики методом Эйлера. Оптимизация модели по заданному критерию, в условиях и ограничениях методом дихотомии.
2. Проверка математической модели на работоспособность методом имитационного моделирования. С помощью генератора случайных чисел формируется случайная последовательность, далее она проходит через фильтр, который формирует случайный процесс с заданным мат.ожиданием и корреляционной функцией. Эта последовательность будет являться колебаниями объемного расхода вещества, поступающего на вход реактора. Она подается на вход мат.модели динамики, и исследуется выходная величина (концентрация фталевого ангидрида). Если ее колебания не покидают заданных пределов, то полученная мат.модель является работоспособной.

***Используемые технические средства***

Программа была написана в среде разработки Mono на компьютере под управлением операционной системы Windows 8.1.

***Вызов и загрузка программы***

Запуск программы осуществляется через файл Razbor.exe. Программа состоит из двух областей.

1. В первой области строится график модели динамики с генератором.
2. Во второй области происходит вывод результатов работы динамики.

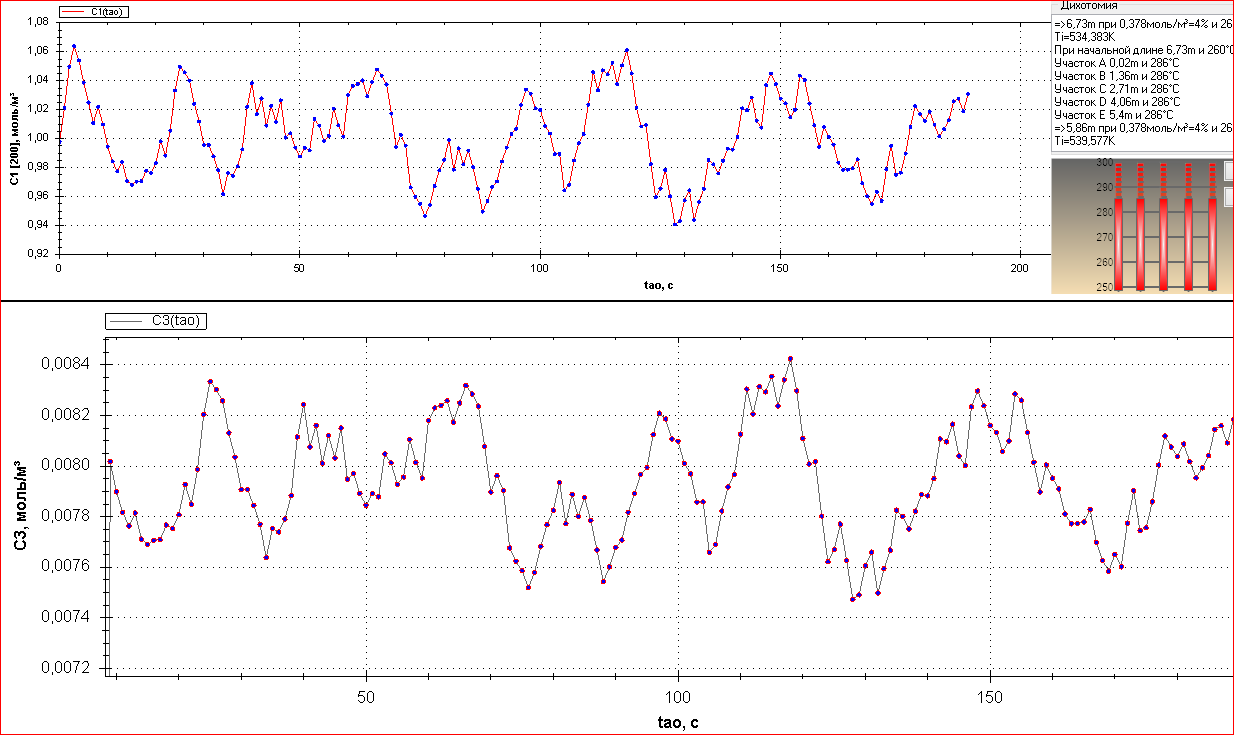


Рисунок Консентрации вещества на выходе повторяют случайные процессы генератора

***Текст программы***

Часть А

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

using ZedGraph;

namespace Razbor

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

zedGraphControl1.GraphPane.Title.Text = "";

zedGraph.GraphPane.Title.Text = "";

zedGraph.GraphPane.YAxis.Title.Text = "С3 вых, моль/м³";

zedGraph.GraphPane.XAxis.Title.Text = "Длина трубы, м";

var listTi = new List<double>();

//Статика//Найти мин длину трубы при 4% С3//Ответ Lmin=2.8m при 286°C и С3=4%

Random rnd = new Random();

listTi.Add(560);

for (double i = 260, k=0; i <= 300; i+=5, k++)

{

var Ti = this.C1(i);

listTi.Add(Ti);

//при разнице температур в 1000К делаем выход

if (Math.Abs(Ti-listTi[listTi.Count-2]) > 1000)

break;

//Для минимальной длины при температуре

clDixotomia = new Дихотомия(Lmin);

for (int j = 0; j < 5; j++)

gaugeControl1.LinearScales[j].Pointers[0].Value = clDixotomia.HiComerce[j];

DrawGraph(\_list, Color.FromArgb(rnd.Next(0, 255), rnd.Next(0, 255), rnd.Next(0, 255)), i + "°C");//i + "°C");

//DrawGraph(clDixotomia.\_list, Color.FromArgb(rnd.Next(0, 255), rnd.Next(0, 255), rnd.Next(0, 255)), "°C");//i + "°C");

textBox1.Text += clDixotomia.ResultLen;

}

//Динамика//ГСЧ лаб6 + лаб4 Характеристик

//получаем 200 чисел

//cтроим С1(tao) по кнопке Динамика

//передаем на метод характеристик

//\_model = new clModel(massivГСЧ, 2.8);

}

private double[] \_arr\_200 { get; set; }

private clГСЧ \_гсч = new clГСЧ();

private clModel \_model;

// Создадим список точек

private PointPairList \_list = new PointPairList();

private Дихотомия clDixotomia;

//private Console clConsole = new Console();

public const double \_A1 = 8000;

public const double \_A2 = 4000;

public const double \_A3 = 300000;

public const double \_E1 = 72600;//Дж\моль

public const double \_E2 = 77000;//Дж\моль

public const double \_E3 = 87000;//Дж\моль

public const double \_R = 8.31;

public const double \_C1\_нафВх = 1.09;//моль\м3

public const double \_C2\_О2вв = 8.75;//моль\м3

public const double \_C3\_вых = 6.62;//моль\м3

public const double \_dL = 0.1;//метр

public const double \_U = 0.1;//м\с

public const double \_Ct = 1100;//Дж\кг\*град

public const double \_p = 1.4;//кг\м3

public const double \_Kt = 500;//Вт\м2\*град

public const double \_D = 0.025;//метр

public const double \_deltaTao = 0.1;//секунд

#region Расчер k

/// <summary>

/// k1=A1exp(-E1/RTt)

/// </summary>

/// <returns></returns>

public double K1(double Tt)

{

//+градус в кельвин

return \_A1 \* Math.Exp(-\_E1 / (\_R \* (Tt + 273.15)));

}

/// <summary>

/// k2=A2exp(-E2/RTt)

/// </summary>

/// <returns></returns>

public double K2(double Tt)

{

return \_A2 \* Math.Exp(-\_E2 / (\_R \* (Tt + 273.15)));

}

/// <summary>

/// k3=A3exp(-E3/RTt)

/// </summary>

/// <returns></returns>

public double K3(double Tt)

{

return \_A3 \* Math.Exp(-\_E3 / (\_R \* (Tt + 273.15)));

}

#endregion

#region Расчет Сi и Ti

private double C1(double temper)

{

var flag = !false;

var listC1 = new PointPairList();

var listC2 = new PointPairList();

var lenArr = 7000;

var C1i = new double[lenArr];

C1i[0] = \_C1\_нафВх;

var C2i = new double[lenArr];

C2i[0] = \_C2\_О2вв;

var C3i = new double[lenArr];

C3i[0] = 0.0;//\_C3\_вых;

var Ti = new double[lenArr];

Ti[0] = 650.0;//Кельвин//Tвх

var \_dL = 0.001;

var limitC3 = ((4.0 \* \_p) / (100.0 \* 0.148));// 4%==>0.37моль\м3

var j = 0.0;//для шага по трубе [0..max]

//var temper = 260;//Цельси

var длинаТрубы = 110.0;//m

for (int i = 1; j < длинаТрубы; i++)

{

var k1 = K1(Ti[i - 1] - 273.15);//в цельси т.к. метод преобразует

var k2 = K2(Ti[i - 1] - 273.15);

var k3 = K3(Ti[i - 1] - 273.15);

var Tt = temper + 273.15;//в кельвин

var rp1 = k1 \* C1i[i-1]\*C2i[i-1];//\_C1\_нафВх \* \_C2\_О2вв;

var rp2 = k2 \* C1i[i-1]\*C2i[i-1]\*C3i[i-1];//\_C1\_нафВх \* \_C2\_О2вв \* \_C3\_вых;

var rp3 = k3 \* C1i[i - 1] \* C2i[i - 1];//\_C1\_нафВх \* \_C2\_О2вв;

var Qp1 = 1.47 \* Math.Pow(10, 7);

var Qp2 = 1.4 \* Math.Pow(10, 8);

var Qp3 = 19.6 \* Math.Pow(10, 6);

Ti[i] = Ti[i-1]+((rp1\*Qp1)/(\_Ct\*\_p) - ((4\*\_Kt)/(\_Ct\*\_p\*\_D))\*(Ti[i-1]-Tt) + (rp2\*Qp2)/(\_Ct\*\_p) + (rp3\*Qp3)/(\_Ct\*\_p))/\_U\*\_dL;

C1i[i] = C1i[i - 1] + (-2.0 \* k1 \* C1i[i - 1] \* C2i[i - 1] - k2 \* C1i[i - 1] \* C2i[i - 1]) / \_U \* \_dL;

C2i[i] = C2i[i - 1] +

(-9.0\*k1\*C1i[i - 1]\*C2i[i - 1] - 12.0\*k2\*C1i[i - 1]\*C2i[i - 1] - 9.0\*k3\*C3i[i - 1]\*C2i[i - 1])/

\_U \* \_dL;

C3i[i] = C3i[i - 1] + (-k3 \* C3i[i - 1] \* C2i[i - 1] + 2.0 \* k1 \* C1i[i - 1] \* C2i[i - 1]) / \_U \* \_dL;

var res = C3i[i];

j += \_dL;//шаг по трубе

\_list.Add(j, res);//вывод на график

//Если достигнут порог вещества на выходе

if (res >= limitC3 && flag)//моль\м3 >= моль\м3

{

SetPointCurve(j, res);//вывод контрольной точки на график

// flag = false;

textBox1.Text +=

String.Format("=>{0:#0.##}m при {1:#0.###}моль/м³=4% и {2:#0.###}°C на Ti={3:#0.###}K", j, res, temper, Ti[i]) +

Environment.NewLine;

Lmin = j;

//exit

return Ti[i];

}

listC1.Add(j, C1i[i]);

listC2.Add(j, C2i[i]);

}

return Ti[Ti.Length-1];

}

/// <summary>

/// Храним найденый минимум в статике

/// </summary>

internal double Lmin { get; set; }

#endregion

//событие Построить график динамики

private void StartDinamiks(object sender, EventArgs e)

{

//очистить график

pane.CurveList.Clear();

zedGraph.GraphPane.YAxis.Title.Text = "С1 [200], моль/м³";

zedGraph.GraphPane.XAxis.Title.Text = "tao, с";

// Обновляем график

zedGraph.Invalidate();

var massivГСЧ = \_гсч.ГСЧрезульт;

\_arr\_200 = massivГСЧ;

C1\_200();

}

/// <summary>

/// График С1(tao)

/// </summary>

private void C1\_200()

{

var list1 = new PointPairList();

foreach (PointD pointD in \_гсч.NaGrafik)

{

if(pointD==null)continue;

list1.Add(pointD.X, pointD.Y);

SetPointCurve(pointD.X, pointD.Y);

}

DrawGraph(list1, Color.Red, "C1(tao)");

}

/// <summary>

/// Нарисовать график зависимости C1(L)

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Start\_C1L(object sender, EventArgs e)

{

zedGraphControl1.GraphPane.YAxis.Title.Text = "С3, моль/м³";

zedGraphControl1.GraphPane.XAxis.Title.Text = "tao, c";

// Обновляем график

zedGraphControl1.Invalidate();

C1\_L();

}

private void C1\_L()

{

//передаем на метод характеристик

\_model = new clModel(\_arr\_200, Lmin);

var list2 = new PointPairList();

foreach (PointD pointD in \_model.NaGrafik2)

{

if (pointD != null)

{

list2.Add(pointD.X, pointD.Y);

SetPointCurve(pointD.X, pointD.Y, true);

}

}

DrawGraph(list2, Color.DimGray, "C3(tao)", new bool[] { true });

}

}

}

Часть Б

using System;

using System.Collections.Generic;

using ZedGraph;

namespace Razbor

{

class Дихотомия

{

public const double \_A1 = 8000;

public const double \_A2 = 4000;

public const double \_A3 = 300000;

public const double \_E1 = 72600;//Дж\моль

public const double \_E2 = 77000;//Дж\моль

public const double \_E3 = 87000;//Дж\моль

public const double \_R = 8.31;

public const double \_C1\_нафВх = 1.09;//моль\м3

public const double \_C2\_О2вв = 8.75;//моль\м3

public const double \_C3\_вых = 6.62;//моль\м3

public const double \_dL = 0.1;//метр

public const double \_U = 0.1;//м\с

public const double \_Ct = 1100;//Дж\кг\*град

public const double \_p = 1.4;//кг\м3

public const double \_Kt = 500;//Вт\м2\*град

public const double \_D = 0.025;//метр

public Дихотомия(double len)

{

Temperature = halfDivision(len);

}

private int[] halfDivision(double len)

{

var maxDegres = new int[5];

HiComerce = new double[5];

var result = 0d;

//Делим len на пять отрезков

var mass = new PointD[5];

var tmp = len/5.0;

for (var i = 0; i < mass.Length; i++)

mass[i] = new PointD((i == 0 ? 0.0 : mass[i - 1].Y), (i == 0 ? tmp : mass[i - 1].Y + tmp));//0..2//от до

ResultLen = Console.WriteLine("При начальной длине {0:#0.##}m и {1}°C", len, 260);

var t\_p = 260;//цельси

var e1 = C3(mass[0].X, mass[0].Y, t\_p);

ResultLen += Console.WriteLine("Участок А {0:#0.##}m и {1}°C", Lmin, trass);

HiComerce[0] = Ttmax-273.15;

var e = C3(mass[1].X, mass[1].Y, t\_p);

ResultLen += Console.WriteLine("Участок B {0:#0.##}m и {1}°C", Lmin, trass);

HiComerce[1] = Ttmax - 273.15;

e = C3(mass[2].X, mass[2].Y, t\_p);

ResultLen += Console.WriteLine("Участок C {0:#0.##}m и {1}°C", Lmin, trass);

HiComerce[2] = Ttmax - 273.15;

e = C3(mass[3].X, mass[3].Y, t\_p);

ResultLen += Console.WriteLine("Участок D {0:#0.##}m и {1}°C", Lmin, trass);

HiComerce[3] = Ttmax - 273.15;

e = C3(mass[4].X, mass[4].Y, t\_p);

ResultLen += Console.WriteLine("Участок E {0:#0.##}m и {1}°C", Lmin, trass);

HiComerce[4] = Ttmax - 273.15;

return maxDegres;

}

internal double[] HiComerce;

private PointD C3(double a, double b, double temper)

{

var flag = !false;

var lenArr = 7000;

var C1i = new double[lenArr];

C1i[0] = Form1.\_C1\_нафВх;

var C2i = new double[lenArr];

C2i[0] = Form1.\_C2\_О2вв;

var C3i = new double[lenArr];

C3i[0] = 0.0;//\_C3\_вых;

var Ti = new double[lenArr];

Ti[0] = 650.0;//Кельвин//Tвх

var \_dL = 0.001;

var limitC3 = ((4.0 \* Form1.\_p) / (100.0 \* 0.148));// 4%==>0.37моль\м3

var Qp1 = 1.47 \* Math.Pow(10, 7);

var Qp2 = 1.4 \* Math.Pow(10, 8);

var Qp3 = 19.6 \* Math.Pow(10, 6);

//var temper = 260;//Цельси

var Tt = temper + 273.15;//в кельвин

var j = a;//для шага по трубе [0..max]

for (int i = 1; j < b; i++)

{

var k1 = K1(Ti[i - 1] - 273.15);//в цельси т.к. метод преобразует

var k2 = K2(Ti[i - 1] - 273.15);

var k3 = K3(Ti[i - 1] - 273.15);

var rp1 = k1 \* C1i[i - 1] \* C2i[i - 1];//\_C1\_нафВх \* \_C2\_О2вв;

var rp2 = k2 \* C1i[i - 1] \* C2i[i - 1] \* C3i[i - 1];//\_C1\_нафВх \* \_C2\_О2вв \* \_C3\_вых;

var rp3 = k3 \* C1i[i - 1] \* C2i[i - 1];//\_C1\_нафВх \* \_C2\_О2вв;

Ti[i] = Ti[i - 1] +

((rp1\*Qp1)/(\_Ct\*\_p) - ((4\*\_Kt)/(\_Ct\*\_p\*\_D))\*(Ti[i - 1] - Tt) + (rp2\*Qp2)/(\_Ct\*\_p) +

(rp3\*Qp3)/(\_Ct\*\_p))/\_U\*\_dL;

C1i[i] = C1i[i - 1] + (-2.0\*k1\*C1i[i - 1]\*C2i[i - 1] - k2\*C1i[i - 1]\*C2i[i - 1])/\_U\*\_dL;

C2i[i] = C2i[i - 1] +

(-9.0\*k1\*C1i[i - 1]\*C2i[i - 1] - 12.0\*k2\*C1i[i - 1]\*C2i[i - 1] - 9.0\*k3\*C3i[i - 1]\*C2i[i - 1])/

\_U\*\_dL;

C3i[i] = C3i[i - 1] + (-k3\*C3i[i - 1]\*C2i[i - 1] + 2.0\*k1\*C1i[i - 1]\*C2i[i - 1])/\_U\*\_dL;

var res = C3i[i];

j += \_dL;//шаг по трубе

//Если достигнут порог вещества на выходе

if (res >= limitC3)//моль\м3 >= моль\м3

{

if (j < (b - a))

{

//поднять температуру

//Tt += 283.15;// 10°C=283.15 Kельвин

Ttmax = Tt;

trass = temper;

Lmin = j;

normal.Add(temper);

return new PointD(j, Tt);

}

else//иначе переход на след отрезок

{

Ttmax = Tt;

trass = temper;

Lmin = j;

normal.Add(temper);

return new PointD(j, Tt);

}

}

}

C3(a, b, temper + 1.0);

return new PointD(j, Ttmax);

}

private double Ttmax=0;

private double trass = 0;

private double Lmin = 0;

internal PointPairList \_list = new PointPairList();

private List<double> normal = new List<double>();

internal string ResultLen { get; set; }

internal int[] Temperature { get; set; }

internal string ResultDxt { get; set; }

}

Часть С

using System;

//Вариант 16 конгратуэнтный метод - генератор

namespace Razbor

{

/// <summary>

/// метод генератор

/// </summary>

class clГСЧ

{

//Количество чисел ряда

private int \_N = 200;

//дисперсия:

private double \_G0 = 0.4;

private double \_a0 = 0.071;

//задать 1

private double \_A1 = 1d;

//математическое ожидание:

private double \_M0 = 1.0;

//длина трубы в статике

private double \_Lmin = 2.8;//m

//скорость распространения в трубе

private double \_U = 0.1;//m\c

public clГСЧ()

{

double x0 = \_N;

//случайный ряд:

var arrL = new double[\_N];

//центрируем ряд

arrL = Xi(arrL, ref x0);

var sum = 0.0;

for (int i = 0; i < \_N; i++)

sum += arrL[i];

//Расчитать матожидание Mx

var Mx = this.Mx(sum);

double w = 0.0;

double sun = 0.0;

for (int i = 0; i < \_N; i++)

w = Math.Pow((arrL[i] - Mx), 2);

sun += w;

var z = new double[1590];//Случайный процес

var v = 0.0;

double Ns = 10.0;

var tauSrednee = \_Lmin / \_U; //среднее время пребывания, c

z = Z(arrL, (int) Ns, tauSrednee);

NaGrafik = new PointD[\_N-10];

for (int i = 0; i < \_N-10; i++)

NaGrafik[i] = new PointD(i, z[i]);

//рассчитаные случайные процессы вернуть на метод характеристик

ГСЧрезульт = z;

}

internal double [] ГСЧрезульт { get; set; }

/// <summary>

/// Расчитать случайный процесс z(t)=>z(k)

/// k=1, Ns=200

/// K = 0.4exp(-0.071dt)

/// </summary>

/// <returns></returns>

private double[] Z(double[] arr, double Ns, double tauSrednee)

{

var z = new double[\_N];

var v = 0d;

//Расчитать случайный процесс z(t)=>z(k)

for (int i = (int)tauSrednee; i < \_N-10; i++)

{

for (int j = i; j < i + 10; j++)

{

v += arr[j] \* \_G0 \* Math.Exp(-\_a0 \* (j - i));

}

z[i] = (\_A1 / Ns \* v + \_M0);

v = 0;

}

return z;

}

/// <summary>

/// Расчитать матожидание Mx

/// </summary>

/// <param name="arr"></param>

/// <returns></returns>

private double Mx(double sum)

{

return 1.0/\_N\*sum;

}

/// <summary>

/// Расчитать X[0..\_N]

/// Центрированная последовательность случайных чисел

/// </summary>

private double[] Xi(double[] arrL, ref double x0)

{

//Центрированная последовательность случайных чисел

for (int i = 0; i < \_N; i++)

arrL[i] = Metod(ref x0) - 0.5061;

return arrL;

}

/// <summary>

/// Найти лямбда[0..\_N]

/// Случайные числа

/// </summary>

private double Metod(ref double x)

{

const double m = 1000000.0;

double a = 8; double inc = 65.0;

x = ((a \* x) + inc) % m;

return x / m;

}

internal PointD[] NaGrafik { get; set; }

}

}

Часть D

using System;

using System.Collections.Generic;

using ZedGraph;

//17 varik

namespace Razbor

{

class clModel

{

//--------------------

public const double \_A1 = 8000;

public const double \_A2 = 4000;

public const double \_A3 = 300000;

public const double \_E1 = 72600;//Дж\моль

public const double \_E2 = 77000;//Дж\моль

public const double \_E3 = 87000;//Дж\моль

public const double \_R = 8.31;

public const double \_C1\_нафВх = 1.09;//моль\м3

public const double \_C2\_О2вв = 8.75;//моль\м3

public const double \_C3\_вых = 6.62;//моль\м3

//public const double \_dL = 0.1;//метр

public const double \_U = 0.1;//м\с

public const double \_Ct = 1100;//Дж\кг\*град

public const double \_p = 1.4;//кг\м3

public const double \_Kt = 500;//Вт\м2\*град

public const double \_D = 0.025;//метр

//----------------------

private const double \_deltaAlf = 0.01;

private const double \_deltaBet = 0.01;

public const double \_T = 650d;//Кельвин//Tвх

//дисперсия:

private double \_G0 = 0.4;

/// <summary>

/// Метод характеристик

/// </summary>

/// <param name="ArrГСЧ">200 чисел с генератора</param>

/// <param name="Lmin">минимальная длина трубы найдено в статике</param>

public clModel(double[] ArrГСЧ, double Lmin)

{

var zz = new double[1000];

var tay = new double[1000];

var count = 0;

var tao\_Max = ArrГСЧ.Length\*\_U; //c

var k1 = K1(\_T);

var k2 = K2(\_T);

var k3 = K3(\_T);

var C3\_ang = new double[1000];

var tauSrednee = Lmin/\_U; //среднее время пребывания

double C1;

var C2 = 8.75; //моль\м3

var C3 = 0.0;

//Note: ВТОРАЯ ОБЛАСТЬ

for (double B = 0; B < (tao\_Max - tauSrednee)/2.0; B += \_deltaBet)

{

if(count>=ArrГСЧ.Length)break;

C1 = ArrГСЧ[count];

C2 = 8.75;

C3 = 0;

for (double A = B; A < tauSrednee + B; A += \_deltaAlf)

{

C1 = C1 + ((-2\*k1\*C1\*C2) - (k2\*C1\*C2))\*\_deltaAlf;

C2 = C2 + (-9.0\*k1\*C1\*C2 - 12.0\*k2\*C1\*C2 - 9.0\*k3\*C3\*C2)\*\_deltaAlf;

C3 = C3 + (-k3\*C3\*C2 + k1\*C1\*C2)\*\_deltaAlf;

tay[count] = A + B;

zz[count] = \_U\*(A - B);

}

C3\_ang[count] = C3;

count++;

}

//Note: ТРЕТЬЯ ОБЛАСТЬ

for (double B = (tao\_Max - tauSrednee)/2; B < tao\_Max/2; B += \_deltaBet)

{

if (count >= ArrГСЧ.Length) break;

C1 = ArrГСЧ[count];

C2 = 8.75;

C3 = 0;

for (double A = B; A < tao\_Max - B; A += \_deltaAlf)

{

C1 = C1 + ((-2\*k1\*C1\*C2) - (k2\*C1\*C2))\*\_deltaAlf;

C2 = C2 + (-9.0\*k1\*C1\*C2 - 12.0\*k2\*C1\*C2 - 9.0\*k3\*C3\*C2)\*\_deltaAlf;

C3 = C3 + (-k3\*C3\*C2 + k1\*C1\*C2)\*\_deltaAlf;

tay[count] = A + B;

zz[count] = \_U\*(A - B);

}

}

NaGrafik2 = new PointD[ArrГСЧ.Length];

for (int i = 0; i < ArrГСЧ.Length-10; i++)

{

NaGrafik2[i] = new PointD(i, C3\_ang[i]);

}

}

internal PointD[] NaGrafik2 { get; set; }

}

}