Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

Тамбовский государственный технический университет

Кафедра

Пояснительная записка

к курсовой работе по дисциплине

«Модели и методы анализа проектных решений»

Выполнил студент гр. –41

Д.В.

Проверил: Ю.В.

Тамбов

**Содержание**

Содержание……………………………………………………………..…………………....…2

1. Введение…………………………………………………………………..………….....3
2. Объект проектирования...................……………………………………..…………….4
3. Математическая модель заданного объекта………………………………………….5
4. Исследование метода оптимизации………………………………………...................7
5. Оптимизация объекта проектирования……………………………...…......................9
6. Проверка работоспособности спроектированного объекта методом имитационного моделирования……………….. …………………………….……………..…...….......9
7. Выводы ………..…………………………………………….………………..….….....10
8. Список литературы…………………………………………….………………..….….11
9. Приложение. Листинг программы………………………………….………………...11
10. **Введение**

***Задание:***

Найти методом штрафных функций площадь поверхности теплообмена и объемный расход реакционной смеси на входе в контактный аппарат производства синильной кислоты, при которых концентрация синильной кислоты на выходе из реактора будет максимальной:

2СH4 + 2NH3 + 3О2 2HCN + 6H2O + Q1;

СН4 + Н2О СО + 3Н2 – Q2;

2СН4 +О2 2СО + 4Н2;

4NH3 + 3О2 2N2 + 6H2O + Q4;

α1 = 3; α2 = α3 = α4 = 2; А1 = 10 000; Е1 = 156 000 Дж/моль; A2 = 20 000; E2 = 166 400 Дж/моль; А3 = 50 000; Е3 = 177 000 Дж/моль; А4 = 500; Е4 = 160 000 Дж/моль; Kt = 2000 Вт/м2·град; Tt = 500 K; V = 10 м3; Твх = 1200 К; Q1 = 361 000 Дж/моль; Q2 = 950 000 Дж/моль; Q4 = 1520 кДж/моль; Сt = 1200 Дж/кг·град; ρ = 1,9 кг/м3; ССН4вх = 13 %; СО2вх = 15 %; С3NHвх = 12 %; 5 ≤ F ≤ 20 м2; 0,01 ≤ vвх ≤ 0,2 м3/c.

При имитационном моделировании выходная координата Cциан.вых.; входная координата Сметан вх; M0 = 15 моль/м3; K = 12exp(–0,1dt).

1. **Объект проектирования**

Объектом проектирования в данной работе является контактный аппарат для производства синильной кислоты. Реакторы смешения – это емкостные аппараты с перемешиванием механической мешалкой или циркуляционным насосом. Благодаря перемешиванию для описания данного реактора можно применить модель «Идеальное смешение». Для идеального смешения характерно абсолютно полное выравнивание всех характеризующих реакцию параметров по объёму реактора.

Реактор смешения периодического действия - аппарат, в который единовременно загружают исходные компоненты, взаимодействующие между собой определенное время, до необходимой степени превращения. Затем полученную смесь выгружают.

Задача оптимального проектирования ставится следующим образом:

Найти методом штрафных функций площадь поверхности теплообмена и объемный расход реакционной смеси на входе в контактный аппарат производства синильной кислоты, при которых концентрация синильной кислоты на выходе из реактора будет максимальной.

Для решения задачи оптимизации было принято решение составить математическую модель данного объекта.

1. **Математическая модель заданного объекта**

Построим математическую модель (ММ) периодического реактора с мешалкой.

Допущения:

Будем использовать ММ идеального смешения, так как дан реактор с мешалкой.

Запишем уравнения кинетики для участвующих в реакции веществ:

Зависимость константы скорости k химической реакции от температуры T выражается уравнением Аррениуса:

Уравнение теплового баланса будет иметь следующий вид:

где *dQ* – изменение тепла в объеме реактора;

– температура на входе и выходе соответсвенно.

тепло, выделяющееся с реакциями (;

qт – поток тепла, уходящий вследствие теплообмена;

Распишем тепловые потоки:

Подставив эти уравнения в уравнение теплового баланса, получим:

Получили систему уравнений:

Полученную систему уравнений будем решать, используя метод Эйлера:

.

***Проверка адекватности модели:***

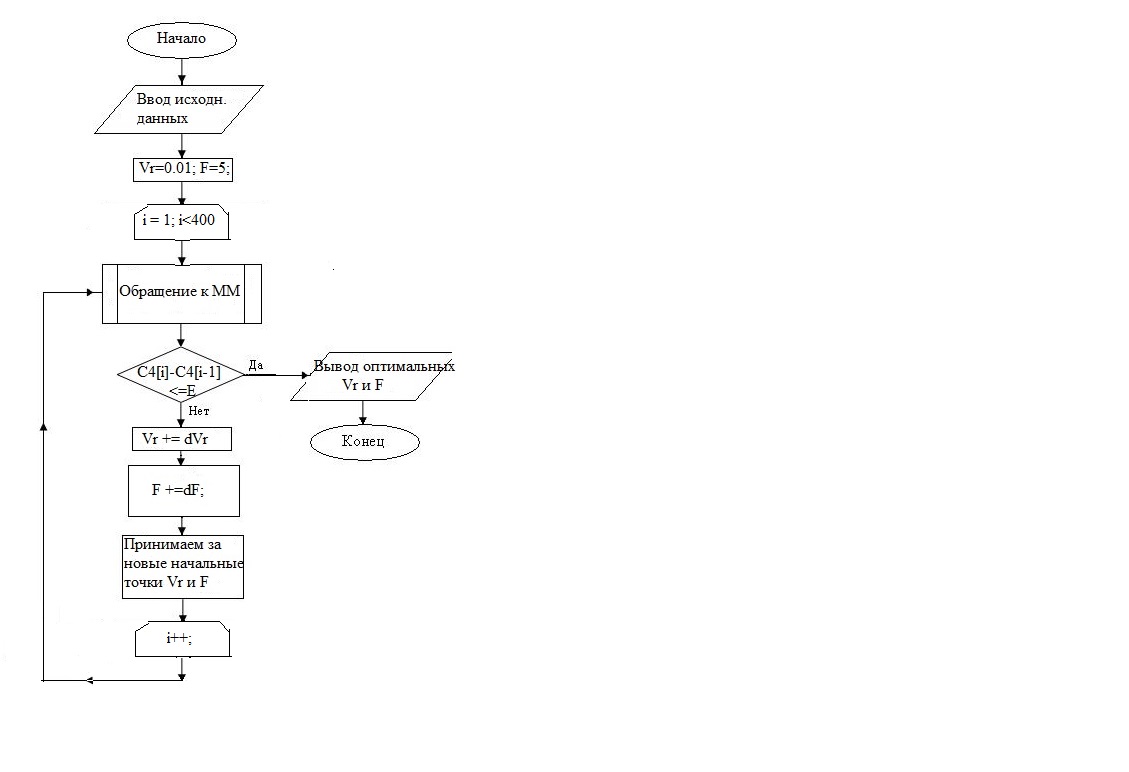
Подавая на вход модели последовательно из диапазона ограничения по площади и объемного расхода реакционной смеси на входе, я заметил при увеличении соответствующих значений уменьшается производительность установки, величина конечного продукта падает. Это являлось логичным доказательством работоспособности математической мадели.

1. **Исследование метода оптимизации**

Основная идея метода штрафных функций состоит в преобразовании задачи минимизации функции с соответствующими ограничениями, наложенными на аргументы http://dit.isuct.ru/ivt/sitanov/Literatura/M171/Pages/Glava3_2.files/image002.gif, в задачу поиска минимума функции без ограничений. Значение площади F и объемного расхода Vr(вх), где F изменяется в пределах 5 ≤ F ≤ 20 м2, а Vr(вх): 0,01 ≤ Vr(вх) ≤ 0,2 м3/с.

Исследуемая функция:

Составим блок-схему алгоритма оптимизации:



1. **Оптимизация объекта проектирования**

Процесс оптимизации состоит в решении системы уравнений и нахождении необходимого параметра. Используя полученную ММ и метод Эйлера, можно легко написать программу, занимающуюся этой оптимизацией.

В результате оптимизации были получены следующие результаты:

Площадь поверхности теплообмена составила F = 5 м2, при объемном расходе реакционной смеси на входе в контактный аппарат равным Vrвх = 0.01 м3/с.

1. **Проверка работоспособности спроектированного объекта методом имитационного моделирования**

Проверка работоспособности спроектированного объекта проводится методом имитационного моделирования.

Выходной координатой является С4(циан.), входной координатой С1(метан).

Для проверки работоспособности спроектированного объекта был построен генератор случайных процессов с заданными характеристиками:

Мат. ожидание М0 = 15 моль/м3;

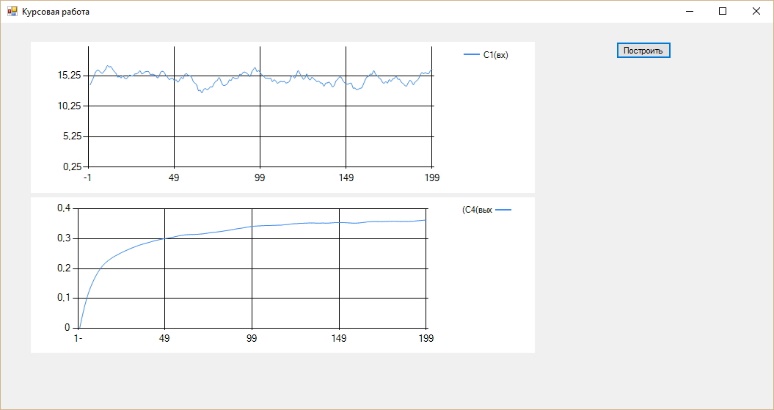
Корреляционная функция K = 12exp(–0,1dt).

Тогда дисперсия σ2=12,

параметр аппроксимации α=0,1.

Для построения генератора случайного процесса воспользуемсяприближенной функцией вида:





В результате проведенного имитационного эксперимента можно заметить, что даже резкие изменения значения входной концентрации С1 дают незначительное влияния на получаемую концентрацию целевого вещества.

1. **Выводы**

В процессе выполнения курсовой работы был изучен контактный аппарат для производства синильной кислоты, выявлены его особенности, протекающие в нем процессы, построена математическая модель этого объекта.

Далее методом штрафных функций была найдена оптимальная площадь поверхности теплообмена, а также оптимальный объемный расход реакционной смеси на входе в контактный аппарат производства синильной кислоты, при которой концентрация синильной кислоты на выходе будет оптимальной. Методом имитационного моделирования была осуществлена проверка математической модели.

Эксперимент показал, что резкие изменения значения входной концентрации С1 не дает резкого влияния на получаемую концентрацию целевого вещества, следовательно, математическая модель работоспособна.

1. **Литература**
2. Построение математических моделей химико-технологических объектов / Е.Г. Дудников, В.С. Балаки- рев, В.Н. Кривсунов, А.М. Цирлин. – Л. : Химия, 1970. – 312 с.
3. Шеннон, Р. Имитационное моделирование систем – искусство и наука / Р. Шеннон. – М. : Мир, 1975. – 502 с.
4. Закгейм А.Ю. Введение в моделирование химико-технологических процессов. М.: Химия, 1982. - 288 с.
5. 25. Бусленко, В.Н. Автоматизация имитационного моделирования сложных систем / В.Н. Бусленко. – М. : Наука. 1977. – 239 с.
6. Юкельсон, И.И. Технология основного органического синтеза / И.И. Юкельсон. – М. : Химия, 1968. – 848 с.
7. **Приложение**

Общие сведения:

Программа называется Modeli(Kursovaya), написана на языке программирования C#, в среде разработки Microsoft Visual Studio 2013.

Функциональное назначение

Программа решает систему уравнений математической модели, построенной по заданному объекту, оптимизирует объект по критерию с заданными ограничениями и условиями и находит значение необходимого параметра.

Описание логической структуры

Программа статического режима делится на 2 части

1. Решение системы уравнений математической модели методом Эйлера.

Программа динамического режима делится на 3 части:

1. Построение генератора случайных чисел для входной концентрации С1.
2. Нахождение выходной концентрации С4.
3. Вывод результатов на график.

Используемые технические средства

Программ была разработана в программном комплексе Microsoft Visual Studio 2013. На компьютере с ОС Windows 8.1,Intel core i-5,ОЗУ 6GB, HDD 750GB.

Вывод и загрузка программы:

Запуск программы осуществляется через файл «Modeli(Kursovaya).exe»

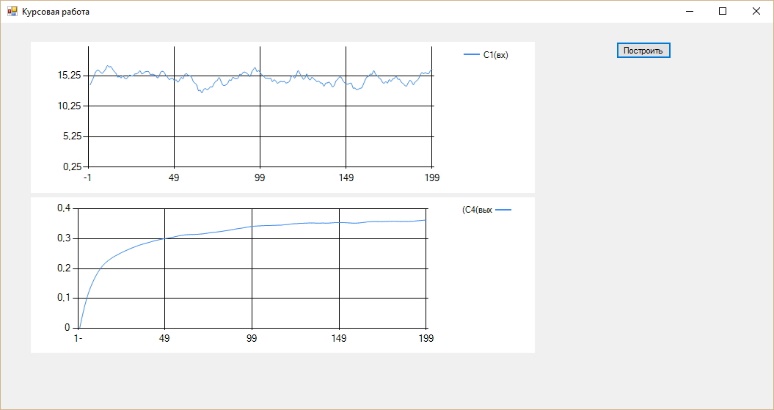
Входные данные:

α1 = 3; α2 = α3 = α4 = 2; А1 = 10 000; Е1 = 156 000 Дж/моль; A2 = 20 000; E2 = 166 400 Дж/моль; А3 = 50 000; Е3 = 177 000 Дж/моль; А4 = 500; Е4 = 160 000 Дж/моль; Kt = 2000 Вт/м2·град; Tt = 500 K; V = 10 м3; Твх = 1200 К; Q1 = 361 000 Дж/моль; Q2 = 950 000 Дж/моль; Q4 = 1520 кДж/моль; Сt = 1200 Дж/кг·град; ρ = 1,9 кг/м3; ССН4вх = 13 %; СО2вх = 15 %; С3NHвх = 12 %; 5 ≤ F ≤ 20 м2; 0,01 ≤ vвх ≤ 0,2 м3/c.

Выходные данные:

Fopt = 5 м2; Vr opt = 0.01 ; C4max = 4,58

График распределения:



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 System.Windows.Forms.DataVisualization.Charting;

using Sluch;

namespace Modeli\_Kursovik\_

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

double d = 0.01;

private void button1\_Click(object sender, EventArgs e)

{

// double C1\_vh = 15.4375, C2\_vh = 13.4118, C3\_vh = 8.90625, C5\_vh = 0;

var C1 = new double[400];

var C2 = new double[400];

var C3 = new double[400];

var C4 = new double[400];

var C5 = new double[400];

var T\_vyh = new double[400];

double E1 = 156000, E2 = 166400, E3 = 177000, E4 = 160000;

double A1 = 10000, A2 = 20000, A3 = 50000, A4 = 500;

double Q1 = 361000, Q2 = 950000, Q4 = 1520000;

double T\_t = 500, M = 19, V = 10, K\_t = 2000, F = 5, V\_vh = 0.01, R = 8.31, Ct = 1200, ro = 1.9;

double k1, k2, k3, k4;

double T\_vh = 1200.0, C1\_vh = 15.4375, C2\_vh = 13.4118, C3\_vh = 8.90625, C4\_vh = 0.0, C5\_vh = 0.0;

double Vr = 0.01 , d = 0.01 , eps = 0.0001;

// ввод входных параметров на начальном этапе

T\_vyh[0] = 1200.0;

C1[0] = 15.4375;

C2[0] = 13.4118;

C3[0] = 8.90625;

C4[0] = 0;

C5[0] = 0;

// блок расчета параметра kn

k1 = A1 \* Math.Exp(-E1 / (R \* T\_vyh[0]));

k2 = A2 \* Math.Exp(-E2 / (R \* T\_vyh[0]));

k3 = A3 \* Math.Exp(-E3 / (R \* T\_vyh[0]));

k4 = A4 \* Math.Exp(-E4 / (R \* T\_vyh[0]));

C5[1] = C5[0] + (((1.0 / (V / Vr)) \* (C5\_vh - C5[0])) - (6 \* k1 \* C1[0] \* C2[0] \* C3[0]) - (k2 \* C1[0] \* C5[0]) + (6 \* k4 \* C2[0] \* C3[0])) \* d;

C4[1] = C4[0] + (((1.0 / (V / Vr)) \* (C4\_vh - C4[0])) + (2 \* k1 \* C1[0] \* C2[0] \* C3[0])) \* d;

//блок расчета дифферинциальных уравнений по методу Эйлера

for (int i = 1; i < C5.Length; i++)

{

// блок расчета уравнений Аррениуса

k1 = A1 \* Math.Exp(-E1 / (R \* T\_vyh[i - 1]));

k2 = A2 \* Math.Exp(-E2 / (R \* T\_vyh[i - 1]));

k3 = A3 \* Math.Exp(-E3 / (R \* T\_vyh[i - 1]));

k4 = A4 \* Math.Exp(-E4 / (R \* T\_vyh[i - 1]));

d = d + 0.001;

//Vr = Vr + 0.01;

// F = F + 0.1;

if (Vr <= 0.2)

Vr = Vr + 0.01; // 0.01 <= Vr <=0.2

else break;

if (F <= 20)

F = F + 0.1; //5 <= F <= 20

else break;

// блок расчета уравнения теплового баланса

if(Math.Abs(T\_vyh[i] - T\_vyh [i-1]) > eps)

{

T\_vyh[i] = T\_vyh[i-1] + ((1 / (V / Vr)) \* (T\_vh - T\_vyh[i-1]) + (k1 \* C1[i-1] \* C2[i-1] \* C3[i-1] \* Q1 / (Ct \* ro)) + (k4 \* C2[i-1] \* C3[i-1] \* Q4 / (Ct \* ro)) - (k2 \* C1[i-1] \* C5[i-1] \* Q2 / (Ct \* ro)) - ((K\_t / (Ct \* ro \*V)) \* (T\_vyh[i-1] - T\_t) \* F)) \* d;

}

if(Math.Abs(C1[i] - C1[i-1]) > eps)

{

C1[i] = C1[i-1] + (((1.0 / (V / Vr)) \* (C1\_vh - C1[i-1])) - (2 \* k1 \* C1[i-1] \*C2 [i-1]\* C3[i-1]) - (k2 \* C1[i-1]\*C3[i-1]) - (2\*k3\*C1[i-1]\*C3[i-1])) \* d;

}

if(Math.Abs(C2[i] - C2[i-1]) > eps)

{

C2[i] = C2[i - 1] + (((1.0 / (V / Vr)) \* (C2\_vh - C1[i - 1])) - (2 \* k1 \* C1[i - 1] \* C2[i - 1] \* C3[i - 1]) - (4 \* k4 \* C2[i-1] \* C3[i-1])) \* d;

}

if(Math.Abs(C3[i] - C3[i-1]) > eps)

{

C3[i] = C3[i - 1] + (((1.0 / (V / Vr)) \* (C3\_vh - C3[i - 1])) - (3 \* k1 \* C1[i - 1] \* C2[i - 1] \* C3[i - 1]) - (k3 \* C1[i - 1] \* C3[i - 1]) - (3 \* k4 \* C2[i - 1] \* C3[i - 1])) \* d;

}

if(Math.Abs(C5[i] - C5[i-1]) > eps)

{

C5[i+1] = C5[i] + (((1.0 / (V / Vr)) \* (C5\_vh - C5[i])) - (6 \* k1 \* C1[i - 1] \* C2[i - 1] \* C3[i - 1]) - (k2 \* C1[i - 1] \* C5[i - 1]) + (6 \* k4 \* C2[i - 1] \* C3[i - 1])) \* d;

}

if(Math.Abs(C4[i] - C4[i-1]) > eps)

{

C4[i+1] = C4[i] + (((1.0 / (V / Vr)) \* (C4\_vh - C4[i])) + (2 \* k1 \* C1[i - 1] \* C2[i - 1] \* C3[i - 1])) \* d;

}

// блок вывода результатов на график

chart1.Series[0].Points.AddXY(d, C4[i]);

chart1.Series[0].ChartType = SeriesChartType.Spline;

}

//C1 = ((1.0 / (V / Vr)) \* (C1\_vh - 0)) - (2 \* k1 \* C1\_vh \* C2\_vh \* C3\_vh) - (k2 \* C1\_vh \* C3\_vh) - (2 \* k3 \* C1\_vh \* C3\_vh);

//C2 = ((1.0 / (V / Vr)) \* (C2\_vh - 0)) - (2 \* k1 \* C1\_vh \* C2\_vh \* C3\_vh) - (4 \* k4 \* C2\_vh \* C3\_vh);

//C3 = ((1.0 / (V / Vr)) \* (C3\_vh - 0)) - (3 \* k1 \* C1\_vh \* C2\_vh \* C3\_vh) - (k3 \* C1\_vh \* C3\_vh) - (3 \* k4 \* C2\_vh \* C3\_vh);

//C5 = ((1.0 / (V / Vr)) \* (C5\_vh - 0)) + (6 \* k1 \* C1\_vh \* C2\_vh \* C3\_vh) - (k2 \* C1\_vh \* C5\_vh) + (6 \* k4 \* C2\_vh \* C3\_vh);

//C4 = ((1 / (V / Vr)) \* (0 - 0)) + (2 \* k1 \* C1 \* C2 \* C3);

// T\_vyh = T\_vh + ((1 / (V / Vr)) \* (T\_vh - T\_vyh) + (k1 \* C1 \* C2 \* C3 \* Q1 / (Ct \* ro)) + (k4 \* C2 \* C3 / (Ct \* ro)) - (k2 \* C1 \* C5 / (Ct \* ro)) - ((K\_t / (Ct \* ro \* V)) \* (T\_vyh - T\_t) \* F)) \* d;

textBox1.Text = Convert.ToString(Vr);//

textBox2.Text = Convert.ToString(F); //

textBox5.Text = Convert.ToString(C4);

}

// пишем генератор случайных чисел

private double generator(ref double x)

{

const double m = 1000000.0;

double a = 8; double inc = 65.0;

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

return x / m;

}

private void button3\_Click(object sender, EventArgs e)

{

// double C1\_vh = 15.4375, C2\_vh = 13.4118, C3\_vh = 8.90625, C5\_vh = 0;

double x0 = 500;

double[] x = new double[500];

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

x[i] = generator(ref x0) - 0.5;

double M, sum = 0.0;

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

sum += x[i];

M = 1.0 / 500.0 \* sum;

double w = 0;

double sun = 0;

//int sun;

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

w = Math.Pow((x[i] - M), 2);

sun += w;

double[] z = new double[490];

double k = 1, v = 0;

double alf0 = 0.1, M0 = 15.0, A1 = 1, A2 = 1, N = 10;

double constanta = 12;//Math.Pow((gam0 / gamx / alf0), (double)1 / (double)2);

for (int i = 0; i < 490; i++)

{

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

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

v += x[j] \* constanta \* Math.Exp(-alf0 \* (j - i));

}

z[i] = (1.0 / N \* v + M0);

v = 0;

}

// блок вывода на график набор случайных чисел для C1

for (int i = 0; i < 490; i++) {

chart2.Series[0].Points.AddXY(i, z[i]);

chart2.Series[0].ChartType = SeriesChartType.Spline;

}

// конец генератора случайных чисел

var C1 = new double[200];

var C2 = new double[200];

var C3 = new double[200];

var C4 = new double[200];

var C5 = new double[200];

var T\_vyh = new double[200];

double E1 = 156000, E2 = 166400, E3 = 177000, E4 = 160000;

double a1 = 10000, a2 = 20000, A3 = 50000, A4 = 500;

double Q1 = 361000, Q2 = 950000, Q4 = 1520000;

double T\_t = 500, Mv = 19, V = 10, K\_t = 2000, F = 5, V\_vh = 0.01, R = 8.31, Ct = 1200, ro = 1.9;

double k1, k2, k3, k4;

double T\_vh = 1200.0, C1\_vh = z[0], C2\_vh = 13.4118, C3\_vh = 8.90625, C4\_vh = 0.0, C5\_vh = 0.0;

double Vr = 0.01, d = 0.0001, eps = 0.001;

T\_vyh[0] = 1200.0;

C1[0] = z[0];

C2[0] = 13.4118;

C3[0] = 8.90625;

C4[0] = 0;

C5[0] = 0;

k1 = a1 \* Math.Exp(-E1 / (R \* T\_vyh[0]));

k2 = a2 \* Math.Exp(-E2 / (R \* T\_vyh[0]));

k3 = A3 \* Math.Exp(-E3 / (R \* T\_vyh[0]));

k4 = A4 \* Math.Exp(-E4 / (R \* T\_vyh[0]));

C5[1] = C5[0] + (((1.0 / (V / Vr)) \* (C5\_vh - C5[0])) - (6 \* k1 \* C1[0] \* C2[0] \* C3[0]) - (k2 \* C1[0] \* C5[0]) + (6 \* k4 \* C2[0] \* C3[0])) \* d;

C4[1] = C4[0] + (((1.0 / (V / Vr)) \* (C4\_vh - C4[0])) + (2 \* k1 \* C1[0] \* C2[0] \* C3[0])) \* d;

// T\_vyh[1] = T\_vyh[0] + ((1 / (V / Vr)) \* (T\_vh - T\_vyh[0]) + (k1 \* C1[0] \* C2[0] \* C3[0] \* Q1 / (Ct \* ro)) + (k4 \* C2[0] \* C3[0] / (Ct \* ro)) - (k2 \* C1[0] \* C5[0] / (Ct \* ro)) - ((K\_t / (Ct \* ro \* V)) \* (T\_vyh[0] - T\_t) \* F)) \* d;

for (int i = 1; i < C5.Length; i++)

{

k1 = A1 \* Math.Exp(-E1 / (R \* T\_vyh[i - 1]));

k2 = A2 \* Math.Exp(-E2 / (R \* T\_vyh[i - 1]));

k3 = A3 \* Math.Exp(-E3 / (R \* T\_vyh[i - 1]));

k4 = A4 \* Math.Exp(-E4 / (R \* T\_vyh[i - 1]));

d = d + 0.01;

// Vr = Vr + 0.01;

// F = F + 0.1;

if (Vr <= 0.2)

Vr = Vr + 0.01; // 0.01 <= Vr <=0.2

else break;

if (F <= 20)

F = F + 0.1; //5 <= F <= 20

else break;

if (Math.Abs(T\_vyh[i] - T\_vyh[i - 1]) > eps)

{

T\_vyh[i] = T\_vyh[i - 1] + ((1 / (V / Vr)) \* (T\_vh - T\_vyh[i - 1]) + (k1 \* z[i] \* C2[i - 1] \* C3[i - 1] \* Q1 / (Ct \* ro)) + (k4 \* C2[i - 1] \* C3[i - 1] \* Q4 / (Ct \* ro)) - (k2 \* z[i] \* C5[i - 1] \* Q2 / (Ct \* ro)) - ((K\_t / (Ct \* Mv)) \* (T\_vyh[i - 1] - T\_t) \* F)) \* d;

}

if (Math.Abs(C1[i] - C1[i - 1]) > eps)

{

C1[i] = z[i] + (((1.0 / (V / Vr)) \* (C1\_vh - z[i])) - (2 \* k1 \* z[i] \* C2[i - 1] \* C3[i - 1]) - (k2 \* z[i] \* C3[i - 1]) - (2 \* k3 \* z[i] \* C3[i - 1])) \* d;

}

if (Math.Abs(C2[i] - C2[i - 1]) > eps)

{

C2[i] = C2[i - 1] + (((1.0 / (V / Vr)) \* (C2\_vh - C2[i - 1])) - (2 \* k1 \* z[i] \* C2[i - 1] \* C3[i - 1]) - (4 \* k4 \* C2[i - 1] \* C3[i - 1])) \* d;

}

if (Math.Abs(C3[i] - C3[i - 1]) > eps)

{

C3[i] = C3[i - 1] + (((1.0 / (V / Vr)) \* (C3\_vh - C3[i - 1])) - (3 \* k1 \* z[i] \* C2[i - 1] \* C3[i - 1]) - (k3 \* z[i] \* C3[i - 1]) - (3 \* k4 \* C2[i - 1] \* C3[i - 1])) \* d;

}

if (Math.Abs(C5[i] - C5[i - 1]) > eps)

{

C5[i + 1] = C5[i] + (((1.0 / (V / Vr)) \* (C5\_vh - C5[i])) - (6 \* k1 \* z[i] \* C2[i - 1] \* C3[i - 1]) - (k2 \* z[i] \* C5[i - 1]) + (6 \* k4 \* C2[i - 1] \* C3[i - 1])) \* d;

}

if (Math.Abs(C4[i] - C4[i - 1]) > eps)

{

C4[i + 1] = C4[i] + (((1.0 / (V / Vr)) \* (C4\_vh - C4[i])) + (2 \* k1 \* z[i] \* C2[i - 1] \* C3[i - 1])) \* d;

}

//chart3.Series[0].Points.AddXY(d, C4[i-1]);

// chart3.Series[0].ChartType = SeriesChartType.Spline;

// chart1.Series[0].Points.AddXY(d, C4);

}

}

}

}