**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа природных ресурсов

Направление подготовки Химическая технология

Отделение химической инженерии

**АППРОКСИМАЦИЯ С ИСПОЛЬЗОВАНИЕМ ЛИНЕЙНОЙ, ЭКСПОНЕНЦИАЛЬНОЙ И СТЕПЕННОЙ ФУНКЦИЙ**

**Лабораторная работа по дисциплине «Углубленный курс информатики»**

Выполнил студент гр. 2Д91 Е.С.Мухина

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Томск 2020 г.

**Цель работы:** научиться составлять программы аппроксимации с использованием линейной, экспоненциальной и степенной функций.

**Теоретическая часть**

Аппроксимация опытных данных – это метод, основанный на замене экспериментально полученных данных аналитической функцией наиболее близко проходящей или совпадающей в узловых точках с исходными значениями (данными полученными в ходе опыта или эксперимента). В настоящее время существует два способа определения аналитической функции:

- с помощью построения интерполяционного многочлена n-степени, который проходит непосредственно через все точки заданного массива данных. В данном случае аппроксимирующая функция представляется в виде: интерполяционного многочлена в форме Лагранжа или  интерполяционного многочлена в форме Ньютона.

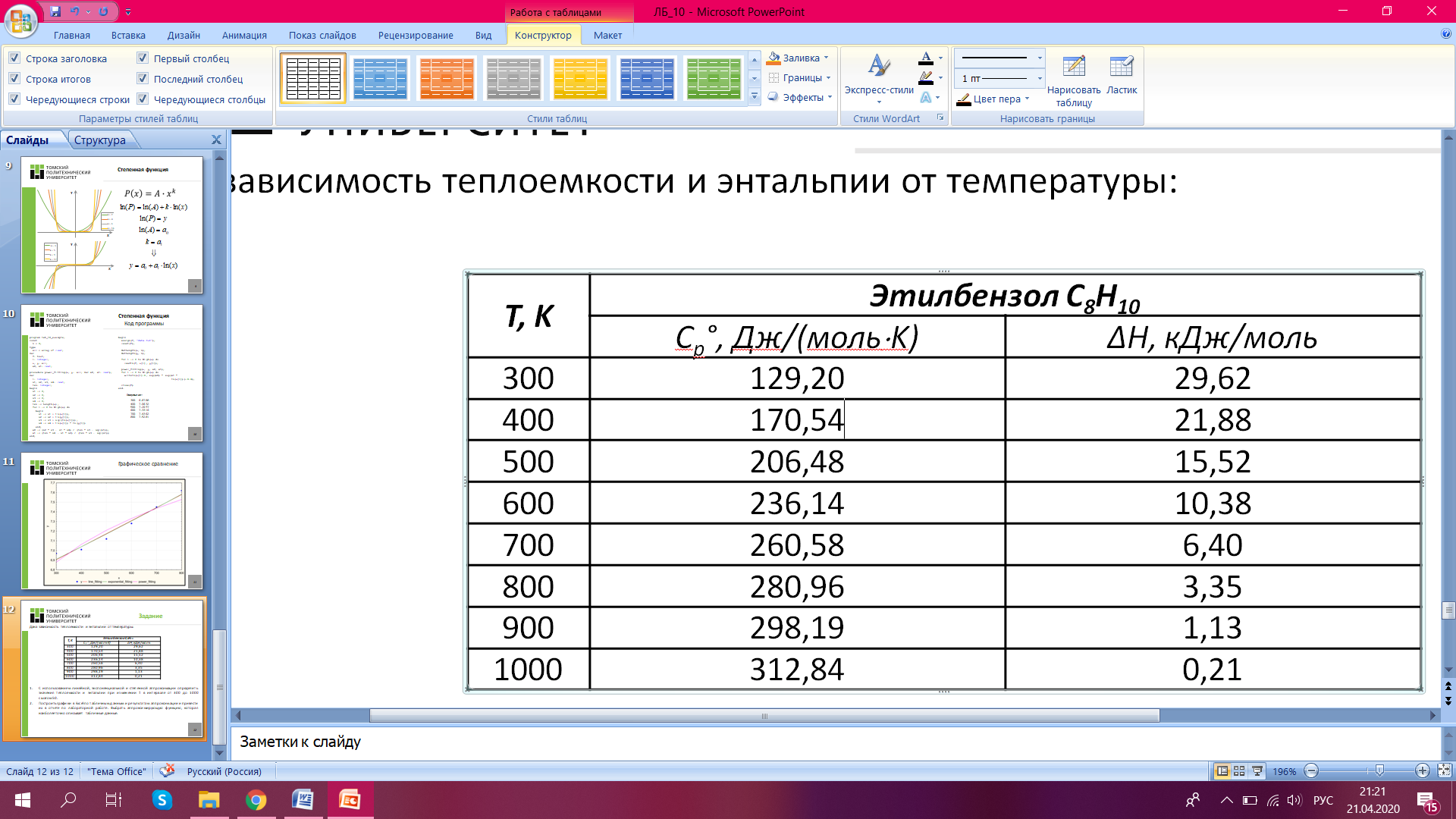
- с помощью построения аппроксимирующего многочлена n-степени, который проходит в ближайшей близости от точек из заданного массива данных. Таким образом, аппроксимирующая функция сглаживает все случайные помехи (или погрешности), которые могут возникать при выполнении эксперимента: измеряемые значения в ходе опыта зависят от случайных факторов, которые колеблются по своим собственным случайным законам (погрешности измерений или приборов, неточность или ошибки опыта). В данном случае аппроксимирующая функция определяется по методу наименьших квадратов.

Метод наименьших квадратов (в англоязычной литературе Ordinary Least Squares, OLS) - математический метод, основанный на определении аппроксимирующей функции, которая строится в ближайшей близости от точек из заданного массива экспериментальных данных. Близость исходной и аппроксимирующей функции F(x) определяется числовой мерой, а именно: сумма квадратов отклонений экспериментальных данных от аппроксимирующей кривой F(x) должна быть наименьшей.

**Практическая часть**

**Задание 1**

**Исходные данные**: Дана зависимость теплоемкости и энтальпии от температуры:



**Задание**

1. С использованием линейной, экспоненциальной и степенной аппроксимации определить значения теплоемкости и энтальпии при изменении Т в интервале от 300 до 1000   
   с шагом 50.
2. Построить графики в Excel по табличным данным и результатам аппроксимации и привести их в отчете по лабораторной работе. Выбрать аппроксимирующую функцию, которая наиболее точно описывает табличные данные.

**Программная реализация**

**Линейная – теплоёмкость:**

**Program** lb10;

**const**

n = 8;

**type**

arr = **array of** real;

**var**

f: text;

i, t: integer;

x, y: arr;

a0, a1: real;

**procedure** line\_fitting(x, y: arr; **var** a0, a1: real);

**var**

i: integer;

s1, s2, s3, s4: real;

len: integer;

**begin**

s1 := 0;

s2 := 0;

s3 := 0;

s4 := 0;

len := Length(x);

**for** i := 0 **to** High(x) **do**

**begin**

s1 := s1 + x[i];

s2 := s2 + y[i];

s3 := s3 + sqr(x[i]);

s4 := s4 + x[i] \* y[i]

**end**;

a0 := (s2 \* s3 - s1 \* s4) / (len \* s3 - sqr(s1));

a1 := (len \* s4 - s1 \* s2) / (len \* s3 - sqr(s1))

**end**;

**begin**

assign(f, 'data.txt');

reset(f);

t:= 300;

SetLength(x, n);

SetLength(y, n);

**for** i := 0 **to** High(x) **do**

readln(f, x[i], y[i]);

line\_fitting(x, y, a0, a1);

**while** t <= 1000 **do**

**begin**

writeln(t:6, a0 + a1 \* t:10:4);

t:= t + 50

**end**;

close(f)

**end**.

**Ответ:**

300 146.3825

350 159.3088

400 172.2350

450 185.1613

500 198.0875

550 211.0138

600 223.9400

650 236.8663

700 249.7925

750 262.7188

800 275.6450

850 288.5713

900 301.4975

950 314.4238

1000 327.3500

**Экспоненциальная - теплоемкость:**

**Program** lb10;

**const**

n = 8;

**type**

arr = **array of** real;

**var**

f: text;

i, t: integer;

x, y: arr;

a0, a1: real;

**procedure** exponential\_fitting(x, y: arr; **var** a0, a1: real);

**var**

i: integer;

s1, s2, s3, s4: real;

len: integer;

**begin**

s1 := 0;

s2 := 0;

s3 := 0;

s4 := 0;

len := Length(x);

**for** i := 0 **to** High(x) **do**

**begin**

s1 := s1 + x[i];

s2 := s2 + ln(y[i]);

s3 := s3 + sqr(x[i]);

s4 := s4 + x[i] \* ln(y[i])

**end**;

a0 := (s2 \* s3 - s1 \* s4) / (len \* s3 - sqr(s1));

a1 := (len \* s4 - s1 \* s2) / (len \* s3 - sqr(s1))

**end**;

**begin**

assign(f, 'data.txt');

reset(f);

t:= 300;

SetLength(x, n);

SetLength(y, n);

**for** i := 0 **to** High(x) **do**

readln(f, x[i], y[i]);

exponential\_fitting(x, y, a0, a1);

**while** t <= 1000 **do**

**begin**

writeln(t:6, exp(a0) \* exp(a1 \* t):10:4);

t:= t + 50

**end**;

close(f)

**end**.

**Ответ:**

300 150.3544

350 159.5821

400 169.3761

450 179.7712

500 190.8043

550 202.5145

600 214.9434

650 228.1351

700 242.1364

750 256.9971

800 272.7697

850 289.5104

900 307.2785

950 326.1370

1000 346.1530

**Степенная - теплоемкость:**

**Program** lb10;

**const**

n = 8;

**type**

arr = **array of** real;

**var**

f: text;

i, t: integer;

x, y: arr;

a0, a1: real;

**procedure** power\_fitting(x, y: arr; **var** a0, a1: real);

**var**

i: integer;

s1, s2, s3, s4: real;

len: integer;

**begin**

s1 := 0;

s2 := 0;

s3 := 0;

s4 := 0;

len := Length(x);

**for** i := 0 **to** High(x) **do**

**begin**

s1 := s1 + ln(x[i]);

s2 := s2 + ln(y[i]);

s3 := s3 + sqr(ln(x[i]));

s4 := s4 + ln(x[i]) \* ln(y[i])

**end**;

a0 := (s2 \* s3 - s1 \* s4) / (len \* s3 - sqr(s1));

a1 := (len \* s4 - s1 \* s2) / (len \* s3 - sqr(s1))

**end**;

**begin**

assign(f, 'data.txt');

reset(f);

t:= 300;

SetLength(x, n);

SetLength(y, n);

**for** i := 0 **to** High(x) **do**

readln(f, x[i], y[i]);

power\_fitting(x, y, a0, a1);

**while** t <= 1000 **do**

**begin**

writeln(t:6, exp(a0) \* exp(a1 \* ln(t)):10:4);

t:= t + 50

**end**;

close(f)

**end**.

**Ответ:**

300 136.7507

350 152.9961

400 168.6205

450 183.7214

500 198.3721

550 212.6292

600 226.5378

650 240.1345

700 253.4496

750 266.5084

800 279.3326

850 291.9406

900 304.3484

950 316.5703

1000 328.6184

**Линейная – энтальпия:**

**Program** lb10;

**const**

n = 8;

**type**

arr = **array of** real;

**var**

f: text;

i, t: integer;

x, y: arr;

a0, a1: real;

**procedure** line\_fitting(x, y: arr; **var** a0, a1: real);

**var**

i: integer;

s1, s2, s3, s4: real;

len: integer;

**begin**

s1 := 0;

s2 := 0;

s3 := 0;

s4 := 0;

len := Length(x);

**for** i := 0 **to** High(x) **do**

**begin**

s1 := s1 + x[i];

s2 := s2 + y[i];

s3 := s3 + sqr(x[i]);

s4 := s4 + x[i] \* y[i]

**end**;

a0 := (s2 \* s3 - s1 \* s4) / (len \* s3 - sqr(s1));

a1 := (len \* s4 - s1 \* s2) / (len \* s3 - sqr(s1))

**end**;

**begin**

assign(f, 'data2.txt');

reset(f);

t:= 300;

SetLength(x, n);

SetLength(y, n);

**for** i := 0 **to** High(x) **do**

readln(f, x[i], y[i]);

line\_fitting(x, y, a0, a1);

**while** t <= 1000 **do**

**begin**

writeln(t:6, a0 + a1 \* t:10:4);

t:= t + 50

**end**;

close(f)

**end**.

**Ответ:**

300 25.6492

350 23.5652

400 21.4812

450 19.3972

500 17.3132

550 15.2292

600 13.1452

650 11.0613

700 8.9773

750 6.8933

800 4.8093

850 2.7253

900 0.6413

950 -1.4427

1000 -3.5267

**Экспоненциальная - энтальпия:**

**Program** lb10;

**const**

n = 8;

**type**

arr = **array of** real;

**var**

f: text;

i, t: integer;

x, y: arr;

a0, a1: real;

**procedure** exponential\_fitting(x, y: arr; **var** a0, a1: real);

**var**

i: integer;

s1, s2, s3, s4: real;

len: integer;

**begin**

s1 := 0;

s2 := 0;

s3 := 0;

s4 := 0;

len := Length(x);

**for** i := 0 **to** High(x) **do**

**begin**

s1 := s1 + x[i];

s2 := s2 + ln(y[i]);

s3 := s3 + sqr(x[i]);

s4 := s4 + x[i] \* ln(y[i])

**end**;

a0 := (s2 \* s3 - s1 \* s4) / (len \* s3 - sqr(s1));

a1 := (len \* s4 - s1 \* s2) / (len \* s3 - sqr(s1))

**end**;

**begin**

assign(f, 'data2.txt');

reset(f);

t:= 300;

SetLength(x, n);

SetLength(y, n);

**for** i := 0 **to** High(x) **do**

readln(f, x[i], y[i]);

exponential\_fitting(x, y, a0, a1);

**while** t <= 1000 **do**

**begin**

writeln(t:6, exp(a0) \* exp(a1 \* t):10:4);

t:= t + 50

**end**;

close(f)

**end**.

**Ответ:**

300 50.4259

350 36.4464

400 26.3423

450 19.0395

500 13.7612

550 9.9462

600 7.1888

650 5.1958

700 3.7554

750 2.7143

800 1.9618

850 1.4179

900 1.0248

950 0.7407

1000 0.5354

**Степенная – энтальпия:**

**Program** lb10;

**const**

n = 8;

**type**

arr = **array of** real;

**var**

f: text;

i, t: integer;

x, y: arr;

a0, a1: real;

**procedure** power\_fitting(x, y: arr; **var** a0, a1: real);

**var**

i: integer;

s1, s2, s3, s4: real;

len: integer;

**begin**

s1 := 0;

s2 := 0;

s3 := 0;

s4 := 0;

len := Length(x);

**for** i := 0 **to** High(x) **do**

**begin**

s1 := s1 + ln(x[i]);

s2 := s2 + ln(y[i]);

s3 := s3 + sqr(ln(x[i]));

s4 := s4 + ln(x[i]) \* ln(y[i])

**end**;

a0 := (s2 \* s3 - s1 \* s4) / (len \* s3 - sqr(s1));

a1 := (len \* s4 - s1 \* s2) / (len \* s3 - sqr(s1))

**end**;

**begin**

assign(f, 'data2.txt');

reset(f);

t:= 300;

SetLength(x, n);

SetLength(y, n);

**for** i := 0 **to** High(x) **do**

readln(f, x[i], y[i]);

power\_fitting(x, y, a0, a1);

**while** t <= 1000 **do**

**begin**

writeln(t:6, exp(a0) \* exp(a1 \* ln(t)):10:4);

t:= t + 50

**end**;

close(f)

**end**.

**Ответ:**

300 64.7566

350 37.2366

400 23.0569

450 15.1071

500 10.3497

550 7.3510

600 5.3789

650 4.0357

700 3.0930

750 2.4145

800 1.9152

850 1.5406

900 1.2549

950 1.0335

1000 0.8597

Степенная аппроксимирующая функция наиболее точно описывает табличные данные зависимости теплоёмкости от температуры.

Линейная аппроксимирующая функция наиболее точно описывает табличные данные зависимости энтальпии от температуры.

**Выводы**

Подводя итог, можно смело утверждать, что я научилась составлять программы аппроксимации с использованием линейной, экспоненциальной и степенной функций.