

НИУ ИТМО
Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе №4
по дисциплине Вычислительная математика

Студент группы № Р32151
Преподаватель

Шипулин Павел Андреевич
Машина Екатерина Алексеевна

Санкт-Петербург

2023

Цель работы

Найти функцию, являющуюся наилучшим приближением заданной табличной функции по методу наименьших квадратов.

Вычислительная часть лабораторной работы

$$y(x) = \frac{31x}{x^4 + 13}, x \in [0; 4], h = 0.4$$

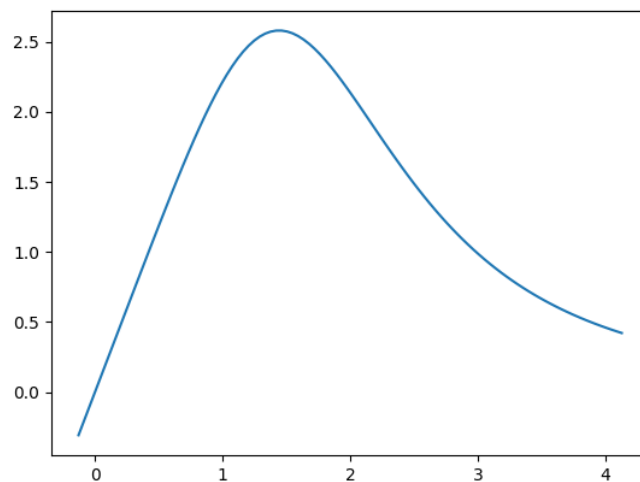


Рисунок 1. График исходной функции

n	x	x^2	x^3	x^4	y	xy	x^2y
1	0,000	0,000	0,000	0,000	0,000	0,000	0,000
2	0,400	0,160	0,064	0,026	0,952	0,381	0,152
3	0,800	0,640	0,512	0,410	1,849	1,480	1,184
4	1,200	1,440	1,728	2,074	2,468	2,961	3,554
5	1,600	2,560	4,096	6,554	2,537	4,059	6,494
6	2,000	4,000	8,000	16,000	2,138	4,276	8,552
7	2,400	5,760	13,824	33,178	1,611	3,867	9,280
8	2,800	7,840	21,952	61,466	1,166	3,264	9,139
9	3,200	10,240	32,768	104,858	0,842	2,693	8,619
10	3,600	12,960	46,656	167,962	0,617	2,220	7,993
11	4,000	16,000	64,000	256,000	0,461	1,844	7,375
Σ	22,000	61,600	193,600	648,525	14,640	27,044	62,341

Линейная аппроксимация

$$y(x) = a + bx$$

$$\begin{cases} an + b\sum x_i &= \sum y_i \\ a\sum x_i + b\sum x_i^2 &= \sum x_i y_i \end{cases}$$

$$\Delta = n \cdot \sum x_i^2 - (\sum x_i)^2 = 193,6$$

$$\Delta_a = \sum y_i \cdot \sum x_i^2 - \sum x_i \cdot \sum x_i y_i = 306,856$$

$$a = \frac{\Delta_a}{\Delta} 1,585$$

$$\Delta_b = n \cdot \sum x_i y_i - \sum y_i \cdot \sum x_i = -24,596$$

$$b = \frac{\Delta_b}{\Delta} - 0,127$$

$$\varphi(x) = 1,585 - 0,127x$$

x_i	y_i	$\varphi(x_i)$	$(\varphi(x_i) - y_i)^2$
0,000	0,000	1,585	2,512
0,400	0,952	1,534	0,339
0,800	1,849	1,483	0,134
1,200	2,468	1,433	1,072
1,600	2,537	1,382	1,334
2,000	2,138	1,331	0,651
2,400	1,611	1,280	0,110
2,800	1,166	1,229	0,004
3,200	0,842	1,179	0,114
3,600	0,617	1,128	0,261
4,000	0,461	1,077	0,379
		Σ	6,910

$$\sigma = \sqrt{\frac{\sum (\varphi(x_i) - y_i)^2}{n}} = 0,793$$

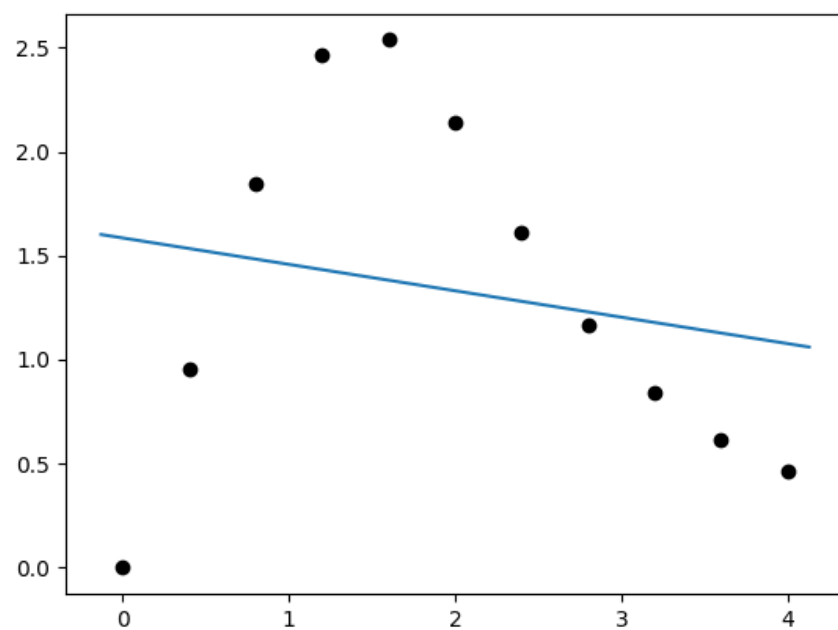


Рисунок 2. Аппроксимация прямой.

Квадратичная аппроксимация

$$y(x) = a + bx + cx^2$$

$$\begin{cases} an + b\sum x_i + c\sum x_i^2 = \sum y_i \\ a\sum x_i + b\sum x_i^2 + c\sum x_i^3 = \sum x_i y_i \\ a\sum x_i^2 + b\sum x_i^3 + c\sum x_i^4 = \sum x_i^2 y_i \end{cases}$$

$$\Delta = \begin{vmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{vmatrix} = 4252,385$$

$$\Delta_a = \begin{vmatrix} \sum y_i & \sum x_i & \sum x_i^2 \\ \sum x_i y_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 y_i & \sum x_i^3 & \sum x_i^4 \end{vmatrix} = 1768,253 \Rightarrow a = \frac{\Delta_a}{\Delta} = 0,416$$

$$\Delta_b = \begin{vmatrix} n & \sum y_i & \sum x_i^2 \\ \sum x_i & \sum x_i y_i & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^2 y_i & \sum x_i^4 \end{vmatrix} = 7745,912 \Rightarrow b = \frac{\Delta_b}{\Delta} = 1,822$$

$$\Delta_c = \begin{vmatrix} n & \sum x_i & \sum y_i \\ \sum x_i & \sum x_i^2 & \sum x_i y_i \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^2 y_i \end{vmatrix} = -2071,524 \Rightarrow c = \frac{\Delta_c}{\Delta} = -0,487$$

$$\varphi(x) = 0,416 + 1,822x - 0,487x^2$$

x_i	y_i	$\varphi(x_i)$	$(\varphi(x_i) - y_i)^2$
0,000	0,000	0,416	0,173
0,400	0,952	1,066	0,013
0,800	1,849	1,561	0,083
1,200	2,468	1,900	0,323
1,600	2,537	2,083	0,206
2,000	2,138	2,110	0,001
2,400	1,611	1,981	0,137
2,800	1,166	1,697	0,282
3,200	0,842	1,256	0,172
3,600	0,617	0,660	0,002
4,000	0,461	-0,092	0,306
		Σ	1,697

$$\sigma = \sqrt{\frac{\sum (\varphi(x_i) - y_i)^2}{n}} = 0,393$$

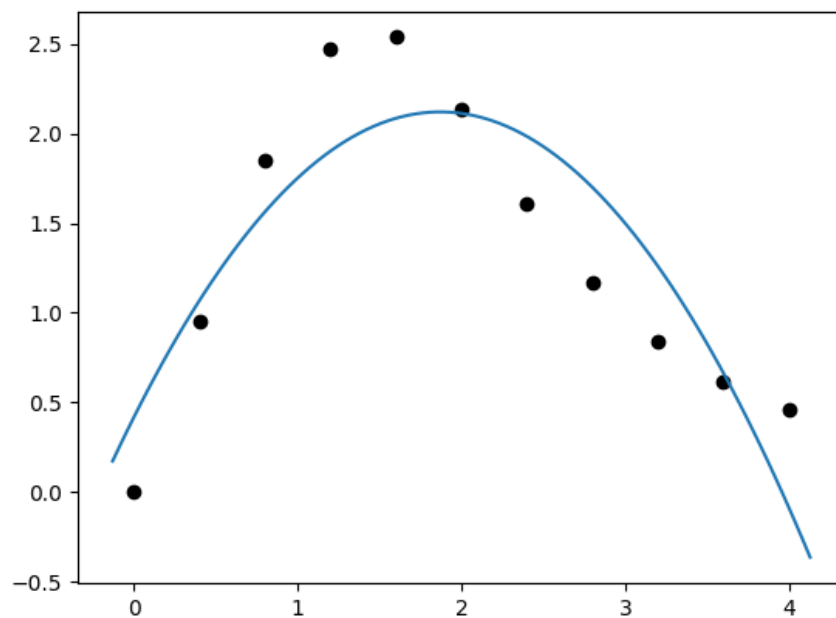


Рисунок 3. Аппроксимация параболой.

Код численных методов

```
from Labs.Lab4.data.matrix import Matrix
from Labs.Lab4.data.matrix import gauss_linear_solve
import math

def pierson_score(x=[], y=[]):
    if len(x) != len(y):
        raise Exception("Размеры данных не совпадают")

    n = len(x)

    mean_x = sum(x) / n
    mean_y = sum(y) / n

    return sum([(x[i] - mean_x) * (y[i] - mean_y) for i in range(n)]) / (
        sum([(x[i] - mean_x) ** 2 for i in range(n)]) * sum([(y[i] -
mean_y) ** 2 for i in range(n)])) ** 0.5

class AnyMethod:
    def __init__(self, x=[], y=[], name="Some method", view="y(x) = ..."):
        self.name = name
        self.view = view
        self.x = x
        self.y = y

    def set_x(self, x=[]):
        self.x = x

    def set_y(self, y=[]):
        self.y = y

    def approximation(self):
        return lambda x: 0, []

class PolyMethod(AnyMethod):
    def __init__(self, x=[], y=[], m=1):
        view = "y(x) = a_0"
        for i in range(1, m + 1):
            view += f" + a_{i} * x ** {i}"
        super().__init__(x, y, f"Полином степени {m}", view)
        self.m = m

    def approximation(self):
        if len(self.x) != len(self.y):
            raise Exception(f"Размеры данных не подходят: {len(self.x)},
{len(self.y)}")

        self.m = int(self.m)
        if self.m < 1:
            raise Exception("Степень полинома должна быть не меньше 1")

        X_sums = [sum([self.x[j] ** i for j in range(len(self.x))]) for i in
range(2 * self.m + 1)]
        XY_sums = [sum([(self.x[j] ** i) * self.y[j] for j in
range(len(self.x))]) for i in range(self.m + 1)]

        A = Matrix(self.m + 1, self.m + 1)
        for i in range(self.m + 1):
            A.set_row(i, [X_sums[j] for j in range(i, i + self.m + 1)])
```

```

        B = Matrix(self.m + 1, 1)
        B.set_col(0, XY_sums)
        a_numbers = gauss_linear_solve(A, B)

        return lambda t: sum([a_numbers[i] * (t ** i) for i in range(self.m + 1)]), a_numbers

class PowMethod(AnyMethod):
    def __init__(self, x=[], y=[]):
        super().__init__(x, y, "Степенная функция", "y(x) = a_0 * x ** a_1")

    def approximation(self):
        if len(self.x) != len(self.y):
            raise Exception("Размеры данных не совпадают")

        try:
            ln_x = [math.log(v, math.e) for v in self.x]
            ln_y = [math.log(v, math.e) for v in self.y]
        except Exception as e:
            raise Exception("Логарифм отрицательного числа, брух")

        func, a_numbers = PolyMethod(ln_x, ln_y, 1).approximation()
        a_numbers[0] = math.exp(a_numbers[0])

        return lambda t: a_numbers[0] * math.pow(t, a_numbers[1]), a_numbers

class ExpMethod(AnyMethod):
    def __init__(self, x=[], y=[]):
        super().__init__(x, y, "Экспонента", "y(x) = a_0 * exp(a_1 * x)")

    def approximation(self):
        if len(self.x) != len(self.y):
            raise Exception("Размеры данных не совпадают")

        try:
            ln_y = [math.log(v, math.e) for v in self.y]
        except Exception as e:
            raise Exception("Логарифм отрицательного числа, брух")

        func, a_numbers = PolyMethod(self.x, ln_y, 1).approximation()
        a_numbers[0] = math.exp(a_numbers[0])

        return lambda t: a_numbers[0] * math.exp(a_numbers[1] * t), a_numbers

class LnMethod(AnyMethod):
    def __init__(self, x=[], y=[]):
        super().__init__(x, y, "Натуральный логарифм", "y(x) = a_0 + a_1 * ln(x)")

    def approximation(self):
        if len(self.x) != len(self.y):
            raise Exception("Размеры данных не совпадают")

        try:
            ln_x = [math.log(v, math.e) for v in self.x]
        except Exception as e:
            raise Exception("Логарифм отрицательного числа, брух")

        return PolyMethod(ln_x, self.y, 1).approximation()

```



```

class LessSquares:
    def __init__(self, method: AnyMethod):
        self.method = method

    def calc(self):
        try:
            phi, a_numbers = self.method.approximation()

            deviation = [phi(self.method.x[i]) - self.method.y[i] for i in
range(len(self.method.x))]
            standard_deviation = math.sqrt(sum([v ** 2 for v in deviation]) /
len(self.method.x))
            r2_score = 1 - standard_deviation / (sum([phi(v ** 2) for v in
self.method.x])
                                                - sum([phi(v) ** 2 for v in
self.method.x]) / len(self.method.x))

            return phi, a_numbers, deviation, standard_deviation, r2_score

        except Exception as e:
            raise Exception(f"Метод ({self.method.name}) не удалось
выполнить: {e}")

```

Результаты выполнения программы

Пример 1

F:\Programming\python\CalcMath\venv\Scripts\python.exe

F:/Programming/python/CalcMath/Labs/Lab4/main.py

[Info]: Введите команду:

lab4

[Input]: lab4

[Info]: Введите путь файла или пустую строку

input.txt

[Input]: input.txt

[Info]: Ввод из файла

[Info]: Введите количество точек

[Input]: 8

[Info]: Введите координату x точки 1

[Input]: 1

[Info]: Введите координату y точки 1

[Input]: 3

[Info]: Введите координату x точки 2

[Input]: 1.1

[Info]: Введите координату y точки 2

[Input]: 3.5

[Info]: Введите координату x точки 3

[Input]: 2.3

[Info]: Введите координату y точки 3
[Input]: 4.1
[Info]: Введите координату x точки 4
[Input]: 3.7
[Info]: Введите координату y точки 4
[Input]: 5.2
[Info]: Введите координату x точки 5
[Input]: 4.5
[Info]: Введите координату y точки 5
[Input]: 6.9
[Info]: Введите координату x точки 6
[Input]: 5.4
[Info]: Введите координату y точки 6
[Input]: 8.3
[Info]: Введите координату x точки 7
[Input]: 6.8
[Info]: Введите координату y точки 7
[Input]: 14.8
[Info]: Введите координату x точки 8
[Input]: 7.5
[Info]: Введите координату y точки 8
[Input]: 21.2

[Info]: Лучшая функция: $y(x) = a_0 + a_1 * x ** 1 + a_2 * x ** 2 + a_3 * x ** 3$

[Info]: Коэффициенты аппроксимирующей функции:

$a_0 = 0.5278637158989511$

$a_1 = 3.509125033479098$

$a_2 = -1.0809148017215509$

$a_3 = 0.13002074642508074$

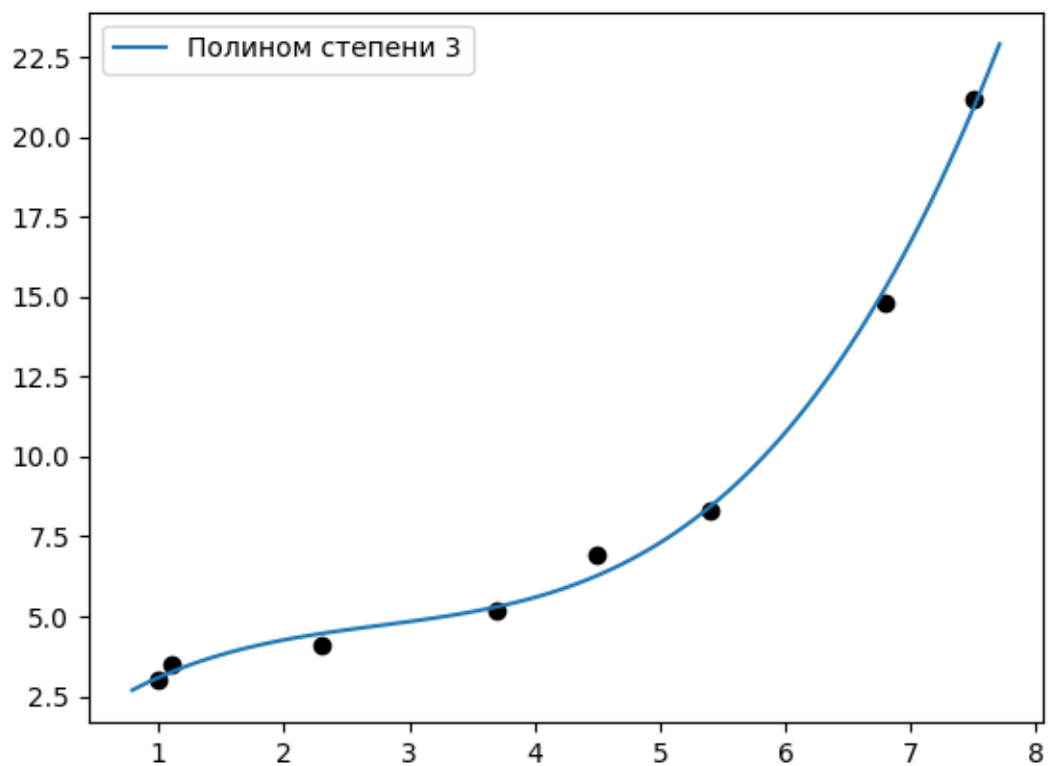
[Info]:

x_i	y_i	ϕ_i	ϵ_i
1.000	3.000	3.086	0.086
1.100	3.500	3.253	-0.247
2.300	4.100	4.463	0.363
3.700	5.200	5.300	0.100
4.500	6.900	6.279	-0.621
5.400	8.300	8.431	0.131

6.800	14.800	15.291	0.491
7.500	21.200	20.897	-0.303

[Info]: Среднеквадратическое отклонение:
0.3439079624248588

[Info]: $R^2 = 0.9999720303378542$



[Info]: Лабораторная работа 4 (аппроксимация)
завершилась

Пример 2

[Info]: Введите команду:

lab4

[Input]: lab4

[Info]: Введите путь файла или пустую строку

input.txt

[Input]: input.txt

[Info]: Ввод из файла

[Info]: Введите количество точек

[Input]: 11

[Info]: Введите координату x точки 1

[Input]: 0,000

[Info]: Введите координату y точки 1

[Input]: 0,000

[Info]: Введите координату x точки 2

[Input]: 0,400

[Info]: Введите координату y точки 2

[Input]: 0,952

[Info]: Введите координату x точки 3

[Input]: 0,800

[Info]: Введите координату y точки 3

[Input]: 1,849

[Info]: Введите координату x точки 4

[Input]: 1,200

[Info]: Введите координату y точки 4

[Input]: 2,468

[Info]: Введите координату x точки 5

[Input]: 1,600

[Info]: Введите координату y точки 5

[Input]: 2,537

[Info]: Введите координату x точки 6

[Input]: 2,000

[Info]: Введите координату y точки 6

[Input]: 2,138

[Info]: Введите координату x точки 7

[Input]: 2,400

[Info]: Введите координату y точки 7

[Input]: 1,611

[Info]: Введите координату x точки 8

[Input]: 2,800

[Info]: Введите координату y точки 8

[Input]: 1,166

[Info]: Введите координату x точки 9

[Input]: 3,200

[Info]: Введите координату y точки 9

[Input]: 0,842

[Info]: Введите координату x точки 10

[Input]: 3,600

[Info]: Введите координату y точки 10

[Input]: 0,617

[Info]: Введите координату x точки 11

[Input]: 4,000

[Info]: Введите координату y точки 11

[Input]: 0,461

[Error]: Метод (Степенная функция) не удалось
выполнить: Логарифм отрицательного числа, брух

[Error]: Метод (Экспонента) не удалось выполнить:
Логарифм отрицательного числа, брух

[Error]: Метод (Натуральный логарифм) не удалось
выполнить: Логарифм отрицательного числа, брух

[Info]: Лучшая функция: $y(x) = a_0 + a_1 * x ** 1 + a_2$
 $* x ** 2 + a_3 * x ** 3$

[Info]: Коэффициенты аппроксимирующей функции:

$a_0 = -0.15004895104890803$

$a_1 = 4.06903846153831$

$a_2 = -1.960504079253987$

$a_3 = 0.24555652680651224$

[Info]:

+-----+-----+-----+-----+

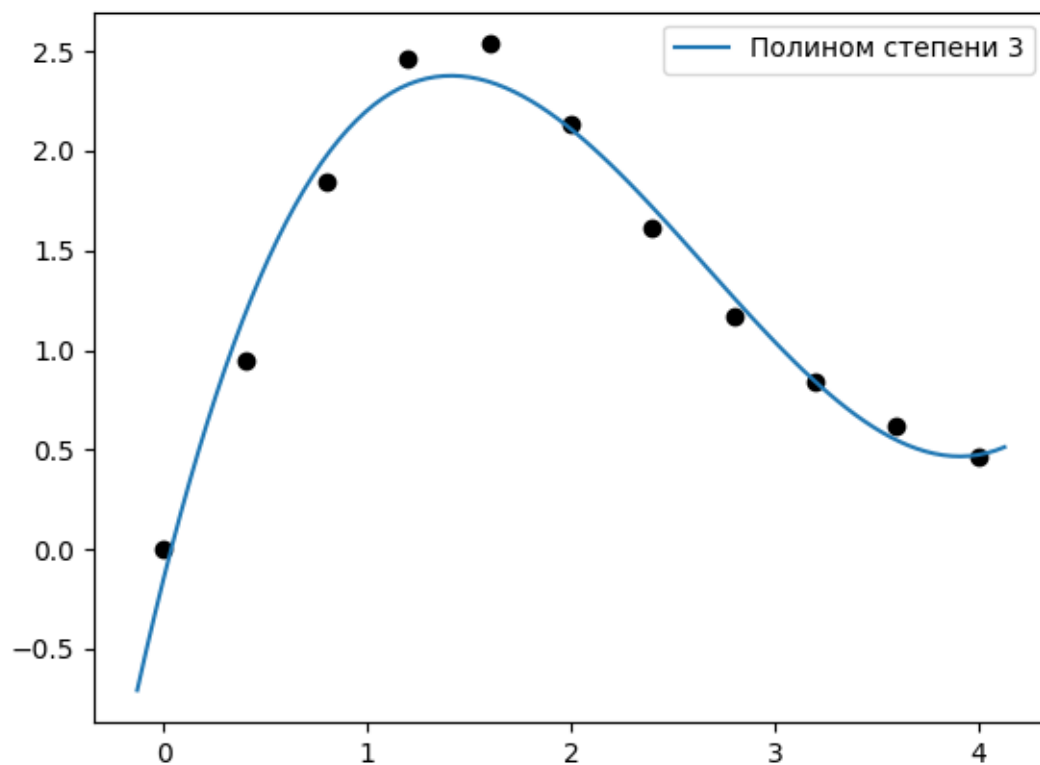
x_i	y_i	phi_i	epsilon_i
0.000	0.000	-0.150	-0.150
0.400	0.952	1.180	0.228
0.800	1.849	1.976	0.127
1.200	2.468	2.334	-0.134
1.600	2.537	2.347	-0.190
2.000	2.138	2.110	-0.028
2.400	1.611	1.718	0.107
2.800	1.166	1.263	0.097
3.200	0.842	0.842	-0.000
3.600	0.617	0.547	-0.070
4.000	0.461	0.474	0.013

+-----+-----+-----+-----+

[Info]: Среднеквадратическое отклонение:

0.12472103221287481

[Info]: $R^2 = 0.9998707601732936$



[Info]: Лабораторная работа 4 (аппроксимация)

завершилась

Пример 3

[Info]: Введите команду:

lab4

[Input]: lab4

[Info]: Введите путь файла или пустую строку

input.txt

[Input]: input.txt

[Info]: Ввод из файла

[Info]: Введите количество точек

[Input]: 9

[Info]: Введите координату x точки 1

[Input]: 1

[Info]: Введите координату y точки 1

[Input]: 1.54

[Info]: Введите координату x точки 2

[Input]: 2

[Info]: Введите координату y точки 2

[Input]: 1.17

[Info]: Введите координату x точки 3

[Input]: 3

[Info]: Введите координату y точки 3

[Input]: 0.03

[Info]: Введите координату x точки 4

[Input]: 4

[Info]: Введите координату y точки 4

[Input]: 1.39

[Info]: Введите координату x точки 5

[Input]: 5

[Info]: Введите координату y точки 5

[Input]: 6.42

[Info]: Введите координату x точки 6

[Input]: 6

[Info]: Введите координату y точки 6

[Input]: 11.76

[Info]: Введите координату x точки 7

[Input]: 7

[Info]: Введите координату y точки 7

[Input]: 12.28

[Info]: Введите координату x точки 8

[Input]: 8

[Info]: Введите координату y точки 8

[Input]: 6.84

[Info]: Введите координату x точки 9

[Input]: 9

[Info]: Введите координату y точки 9

[Input]: 0.8

[Info]: Лучшая функция: $y(x) = a_0 + a_1 * x ** 1 + a_2 * x ** 2 + a_3 * x ** 3$

[Info]: Коэффициенты аппроксимирующей функции:

$a_0 = 10.903730158726656$

$a_1 = -11.725300625297052$

a_2 = 3.4651984126975464

a_3 = -0.25473905723899903

[Info]:

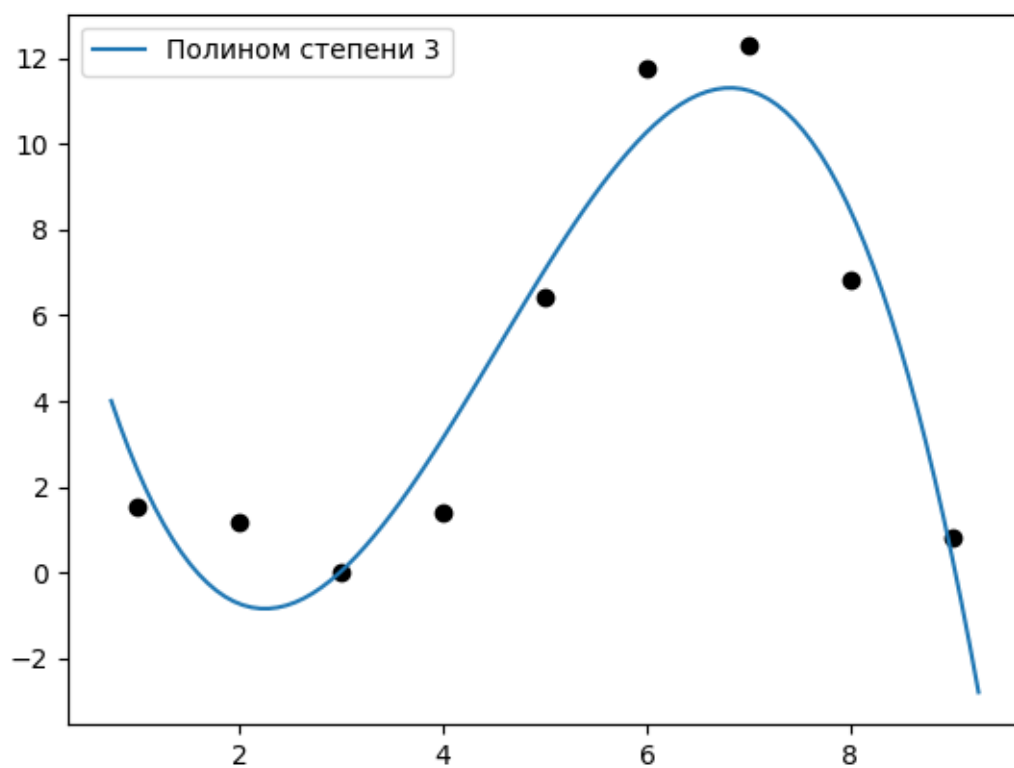
x_i	y_i	phi_i	epsilon_i	
1.000	1.540	2.389	0.849	
2.000	1.170	-0.724	-1.894	
3.000	0.030	0.037	0.007	
4.000	1.390	3.142	1.752	
5.000	6.420	7.065	0.645	
6.000	11.760	10.275	-1.485	
7.000	12.280	11.246	-1.034	
8.000	6.840	8.448	1.608	

	9.000		0.800		0.352		-0.448	
--	-------	--	-------	--	-------	--	--------	--

+	-----	+	-----	+	-----	+	-----	+
---	-------	---	-------	---	-------	---	-------	---

[Info]: Среднеквадратическое отклонение:
1.240646004797171

[Info]: $R^2 = 1.0000062221853137$



[Info]: Лабораторная работа 4 (аппроксимация)
завершилась

Выводы

Научился аппроксимировать функции полиномами (до 3-й степени) и экспонентами по методу наименьших квадратов.