Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО».

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4
Вариант №12
Аппроксимация функции методом наименьших квадратов

Выполнил
Путинцев Данил Денисович
Группа P3207
Проверил(а)
Преподаватель: Рыбаков Степан Дмитриевич

Санкт-Петербург 2025 год

Цель работы

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

Рабочие формулы метода

Вычислительная часть лабораторной работы

Функция:

$$y=\frac{4x}{x^4+12}, x \in [-2;0], h=0.2$$

Таблица 1: Таблица табулирования заданной функции

x	-2	-1.8	-1.6	-1.4	-1.2	-1	-0.8	-0.6	-0.4	-0.2	0
y	-0.286	-0.32	-0.345	-0.353	-0.341	-0.308	-0.258	-0.198	-0.133	-0.067	0
f(x)	-0.396	-0.364	-0.333	-0.301	-0.27	-0.238	-0.206	-0.175	-0.143	-0.112	-0.08
(f(x) - y)^2	0.012	0.002	0	0.003	0.005	0.005	0.003	0	0	0.002	0.006
g(x)	-0.297	-0.324	-0.339	-0.34	-0.328	-0.303	-0.265	-0.214	-0.149	-0.07	0.019
(g(x) - y)^2	0	0	0	0	0	0	0	0	0	0	0

Построим линейное приближение

$SX=-11, SXX=15.4, SY=-2.609, SXY=3.3032$

Получаем систему линейных уравнений:

{15.4a - 11b = 3.3032; -11a + 11b = -2.609}

Решая систему, получим значение коэффициентов: a = 0.158, b = -0.08

f(x) = 0.158x — 0.08

$$\delta=\sqrt{\frac{\sum_{i=1}^{n}\left(f(x_i)-y_i\right)^2}{n}}=0.0035$$

Построим квадратичное приближение

$SX=-11, SXX=15.4, SXXX=-24.2, SXXXX=40.5328, SY=-2.609, SXY=3.3032, SXXY=-4.8153$

Получим систему линейных уравнений:

{11a -11b + 15.4c = -2.609}

{-11a + 15.4b -24.2c = 3.3032}

{15.4a - 24.2b + 40.5328c = -4.8153}

Ответ: c = 0.019, b = 0.4866, a = 0.1644

g(x) = $0.1644x^2 + 0.4866x + 0.019$

$$\delta=\sqrt{\frac{\sum_{i=1}^{n}\left(g\left(x_i\right)-y_i\right)^2}{n}}=0$$

Следовательно, квадратичное приближение лучше

Листинг программы (по крайней мере, коды используемого метода)

```
import math

import numpy as np

def linial_approx(x, y):
   sum_x = 0
   sum_xx = 0
   sum_y = 0
   sum_xy = 0
   for i in range(len(x)):
      sum_x += x[i]
      sum_xx += (x[i]**2)
      sum_y += y[i]
      sum_xy += (x[i]*y[i])
   delta = len(x) * sum_xx - sum_x ** 2

   b = (sum_y * sum_xx - sum_x * sum_xy) / delta
   a = (len(x) * sum_xy - sum_x * sum_y) / delta

   return a, b

def bipolin_approx(x, y):
   sum_x = 0
   sum_xx = 0
   sum_xxx = 0
   sum_xxxx = 0
   sum_y = 0
   sum_xy = 0
   sum_xxy = 0
```

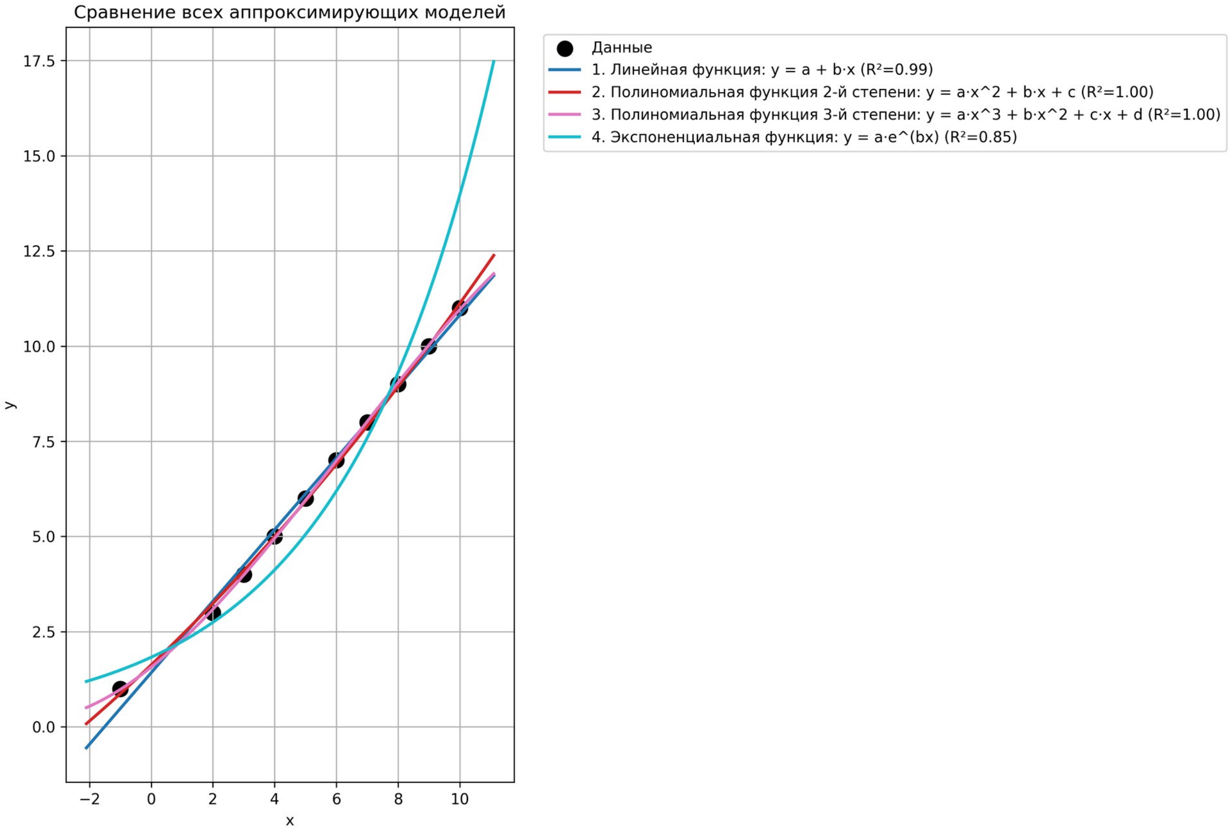
```python
    for i in range(len(x)):
        sum_x += x[i]
        sum_xx += (x[i]**2)
        sum_xxx += (x[i]**3)
        sum_xxxx += (x[i]**4)
        sum_y += y[i]
        sum_xy += (x[i]*y[i])
        sum_xxy += (x[i]**2 * y[i])
    A = [
        [len(x), sum_x, sum_xx],
        [sum_x, sum_xx, sum_xxx],
        [sum_xx, sum_xxx, sum_xxxx]
    ]

    B = [sum_y, sum_xy, sum_xxy]
    c, b, a = np.linalg.solve(A, B)

    return a, b, c

def cubic_approx(x, y):
    sum_x = 0
    sum_xx = 0
    sum_xxx = 0
    sum_xxxx = 0
    sum_xxxxx = 0
    sum_xxxxxx = 0
    sum_y = 0
    sum_xy = 0
    sum_xxy = 0
    sum_xxxy = 0
    for i in range(len(x)):
        sum_x += x[i]
        sum_xx += (x[i]**2)
        sum_xxx += (x[i]**3)
        sum_xxxx += (x[i]**4)
        sum_xxxxx += (x[i]**5)
        sum_xxxxxx += (x[i]**6)
        sum_y += y[i]
        sum_xy += (x[i] * y[i])
        sum_xxy += (x[i]**2 * y[i])
        sum_xxxy += (x[i]**3 * y[i])

    A = [
        [len(x), sum_x, sum_xx, sum_xxx],
```

```
        [sum_x, sum_xx, sum_xxx, sum_xxxx],
        [sum_xx, sum_xxx, sum_xxxx, sum_xxxxx],
        [sum_xxx, sum_xxxx, sum_xxxxx, sum_xxxxxx]
    ]

    B = [sum_y, sum_xy, sum_xxy, sum_xxxy]

    d, c, b, a = np.linalg.solve(A, B)

    return a, b, c, d

def exp_approx(x, y):
    log_y = np.log(y)
    sum_x = 0
    sum_xx = 0
    sum_logy = 0
    sum_x_logy = 0
    for i in range(len(x)):
        sum_x += x[i]
        sum_xx += (x[i]**2)
        sum_logy += log_y[i]
        sum_x_logy += (x[i] * log_y[i])

    A = [
        [len(x), sum_x],
        [sum_x, sum_xx]
    ]
    B = [sum_logy, sum_x_logy]

    A_coeff, b = np.linalg.solve(A, B)
    a = np.exp(A_coeff)

    return b, a

def log_approx(x, y):
    log_x = np.log(x)

    sum_logx = 0
    sum_logx2 = 0
    sum_y = 0
    sum_logx_y = 0
    for i in range(len(x)):
        sum_logx += log_x[i]
        sum_logx2 += (log_x[i]**2)
```

```python
        sum_y += y[i]
        sum_logx_y += (log_x[i]*y[i])

    A = [
        [len(x), sum_logx],
        [sum_logx, sum_logx2]
    ]
    B = [sum_y, sum_logx_y]

    a, b = np.linalg.solve(A, B)
    return b, a

def power_approx(x, y):
    log_x = np.log(x)
    log_y = np.log(y)

    sum_logx = 0
    sum_logx2 = 0
    sum_logy = 0
    sum_logx_logy = 0
    for i in range(len(x)):
        sum_logx += log_x[i]
        sum_logx2 += (log_x[i]**2)
        sum_logy += log_y[i]
        sum_logx_logy += (log_x[i] * log_y[i])

    A = [
        [len(x), sum_logx],
        [sum_logx, sum_logx2]
    ]
    B = [sum_logy, sum_logx_logy]

    A_coeff, b = np.linalg.solve(A, B)
    a = np.exp(A_coeff)

    return a, b

def corr_pirson(x, y):
    x_sr = sum(x) / len(x)
    y_sr = sum(y) / len(y)
    sum_1 = 0
    sum_2 = 0
    sum_3 = 0
    for i in range(len(x)):
```

```
    sum_1 += (x[i] - x_sr)*(y[i] - y_sr)
    sum_2 += (x[i] - x_sr)**2
    sum_3 += (y[i] - y_sr)**2

  r = sum_1 / math.sqrt(sum_2 * sum_3)
  return r
```

Графики аппроксимирующих функций



Результаты выполнения программы при различных исходных данных (не менее трех)

!("/exit" to quit) Введите команду: => 7

Отправка: 7

Введите путь к файлу: table.txt

!("/exit" to quit) Введите команду: => /info

Отправка: /info

Введена таблица:

Значения x: [-1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0]

Значения y: [1.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0]

!("/exit" to quit) Введите команду: => 4

Отправка: 4

Выберете тип функции для исследования:

1. Линейная функция

2. Полиномиальная функция 2-й степени

3. Полиномиальная функция 3-й степени

4. Экспоненциальная функция

5. Логарифмическая функция

6. Степенная функция

7. Исследовать все типы функции

Введите тип функции: 7

Коэффициент корреляции Пирсона: 0.9971890580229869

1.827242291856008 * e^0.20341663784936884x

Аппроксимация невозможна. Значения по X должны быть положительными

Аппроксимация невозможна. Значения по X должны быть положительными

Аппроксимация невозможна. Значения по Y должны быть положительными

Наилучшая модель:

Тип: 3. Полиномиальная функция 3-й степени: y = a·x^3 + b·x^2 + c·x + d

Коэффициенты: a = -0.0033, b = 0.0619, c = 0.6504, d = 1.5671

R^2: 0.9998

Интерпретация: отлично объясняет данные ($R^2 \geq 0.9$)

1. Линейная функция: y = a + b·x

```
--------------------------------------------------
```

Коэффициенты: a = 0.9395, b = 1.4207

Среднеквадратичное отклонение (SSE): 0.5187

R^2 = 0.9944 (отлично объясняет данные ($R^2 \geq 0.9$))

```
+--------+--------+--------+--------+
|   x    |   y    |  f(x)  |  eps   |
+--------+--------+--------+--------+
```

-1.0000	1.0000	0.4813	0.5187
2.0000	3.0000	3.2997	-0.2997
3.0000	4.0000	4.2392	-0.2392
4.0000	5.0000	5.1787	-0.1787
5.0000	6.0000	6.1182	-0.1182
6.0000	7.0000	7.0576	-0.0576
7.0000	8.0000	7.9971	0.0029
8.0000	9.0000	8.9366	0.0634
9.0000	10.0000	9.8761	0.1239
10.0000	11.0000	10.8156	0.1844

2. Полиномиальная функция 2-й степени: y = a·x^2 + b·x + c

Коэффициенты: a = 0.0177, b = 0.7720, c = 1.6252

Среднеквадратичное отклонение (SSE): 0.1291

R^2 = 0.9986 (отлично объясняет данные ($R^2 \geq 0.9$))

x	y	f(x)	eps
-1.0000	1.0000	0.8709	0.1291
2.0000	3.0000	3.2402	-0.2402
3.0000	4.0000	4.1009	-0.1009
4.0000	5.0000	4.9971	0.0029
5.0000	6.0000	5.9289	0.0711
6.0000	7.0000	6.8961	0.1039
7.0000	8.0000	7.8987	0.1013
8.0000	9.0000	8.9369	0.0631

9.0000	10.0000	10.0106	-0.0106
10.0000	11.0000	11.1197	-0.1197

3. Полиномиальная функция 3-й степени: $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$

Коэффициенты: $a = -0.0033$, $b = 0.0619$, $c = 0.6504$, $d = 1.5671$

Среднеквадратичное отклонение (SSE): 0.0181

$R^2 = 0.9998$ (отлично объясняет данные ($R^2 \geq 0.9$))

x	y	f(x)	eps
-1.0000	1.0000	0.9819	0.0181
2.0000	3.0000	3.0891	-0.0891
3.0000	4.0000	3.9864	0.0136
4.0000	5.0000	4.9481	0.0519
5.0000	6.0000	5.9544	0.0456
6.0000	7.0000	6.9854	0.0146
7.0000	8.0000	8.0215	-0.0215
8.0000	9.0000	9.0427	-0.0427
9.0000	10.0000	10.0292	-0.0292
10.0000	11.0000	10.9613	0.0387

4. Экспоненциальная функция: $y = a \cdot e^{(bx)}$

Коэффициенты: a = 1.8272, b = 0.2034

Среднеквадратичное отклонение (SSE): 14.0747

R^2 = 0.8479 (хорошо объясняет данные (0.7 ≤ R^2 < 0.9))

x	y	f(x)	eps
-1.0000	1.0000	1.4909	-0.4909
2.0000	3.0000	2.7446	0.2554
3.0000	4.0000	3.3638	0.6362
4.0000	5.0000	4.1226	0.8774
5.0000	6.0000	5.0525	0.9475
6.0000	7.0000	6.1923	0.8077
7.0000	8.0000	7.5892	0.4108
8.0000	9.0000	9.3012	-0.3012
9.0000	10.0000	11.3994	-1.3994
10.0000	11.0000	13.9709	-2.9709

!("/exit" to quit) Введите команду: => 7

Отправка: 7

Введите путь к файлу: table.txt

!("/exit" to quit) Введите команду: => /info

Отправка: /info

Введена таблица:

Значения x: [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0]

Значения y: [1.0, 2.4, 5.0, 3.0, 7.0, 8.0, 10.0, 21.0, 33.0, 44.0]

!("/exit" to quit) Введите команду: => 4

Отправка: 4

Выберете тип функции для исследования:

1. Линейная функция

2. Полиномиальная функция 2-й степени

3. Полиномиальная функция 3-й степени

4. Экспоненциальная функция

5. Логарифмическая функция

6. Степенная функция

7. Исследовать все типы функции

Введите тип функции: 7

Коэффициент корреляции Пирсона: 0.8849450077853759

0.9164991296568911 * e^0.3837930357945538x

Наилучшая модель:

Тип: 3. Полиномиальная функция 3-й степени: $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$

Коэффициенты: a = 0.1143, b = -1.0691, c = 3.9711, d = -1.8267

R^2: 0.9886

Интерпретация: отлично объясняет данные ($R^2 \geq 0.9$)

1. Линейная функция: $y = a + b \cdot x$

Коэффициенты: a = 4.2618, b = -10.0000

Среднеквадратичное отклонение (SSE): 414.9687

$R^2 = 0.7831$ (хорошо объясняет данные ($0.7 \leq R^2 < 0.9$))

```
+--------+--------+--------+--------+
|   x    |   y    |  f(x)  |  eps   |
+--------+--------+--------+--------+
| 1.0000 | 1.0000 | -5.7382 | 6.7382 |
| 2.0000 | 2.4000 | -1.4764 | 3.8764 |
```

3.0000	5.0000	2.7855	2.2145
4.0000	3.0000	7.0473	-4.0473
5.0000	7.0000	11.3091	-4.3091
6.0000	8.0000	15.5709	-7.5709
7.0000	10.0000	19.8327	-9.8327
8.0000	21.0000	24.0945	-3.0945
9.0000	33.0000	28.3564	4.6436
10.0000	44.0000	32.6182	11.3818

2. Полиномиальная функция 2-й степени: y = a·x^2 + b·x + c

Коэффициенты: a = 0.8174, b = -4.7298, c = 7.9833

Среднеквадратичное отклонение (SSE): 62.1684

R^2 = 0.9675 (отлично объясняет данные ($R^2 \geq 0.9$))

x	y	f(x)	eps
1.0000	1.0000	4.0709	-3.0709
2.0000	2.4000	1.7933	0.6067
3.0000	5.0000	1.1506	3.8494
4.0000	3.0000	2.1427	0.8573
5.0000	7.0000	4.7697	2.2303
6.0000	8.0000	9.0315	-1.0315

7.0000	10.0000	14.9282	-4.9282
8.0000	21.0000	22.4597	-1.4597
9.0000	33.0000	31.6261	1.3739
10.0000	44.0000	42.4273	1.5727

3. Полиномиальная функция 3-й степени: y = a·x^3 + b·x^2 + c·x + d

----

Коэффициенты: a = 0.1143, b = -1.0691, c = 3.9711, d = -1.8267

Среднеквадратичное отклонение (SSE): 21.7896

R^2 = 0.9886 (отлично объясняет данные ($R^2 \geq 0.9$))

x	y	f(x)	eps
1.0000	1.0000	1.1897	-0.1897
2.0000	2.4000	2.7538	-0.3538
3.0000	5.0000	3.5517	1.4483
4.0000	3.0000	4.2694	-1.2694
5.0000	7.0000	5.5929	1.4071
6.0000	8.0000	8.2083	-0.2083
7.0000	10.0000	12.8015	-2.8015
8.0000	21.0000	20.0586	0.9414
9.0000	33.0000	30.6656	2.3344
10.0000	44.0000	45.3085	-1.3085

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

4. Экспоненциальная функция: $y = a \cdot e^{(bx)}$

Коэффициенты: a = 0.9165, b = 0.3838

Среднеквадратичное отклонение (SSE): 39.8991

$R^2 = 0.9792$ (отлично объясняет данные ($R^2 \geq 0.9$))

x	y	f(x)	eps
1.0000	1.0000	1.3453	-0.3453
2.0000	2.4000	1.9747	0.4253
3.0000	5.0000	2.8985	2.1015
4.0000	3.0000	4.2545	-1.2545
5.0000	7.0000	6.2449	0.7551
6.0000	8.0000	9.1666	-1.1666
7.0000	10.0000	13.4551	-3.4551
8.0000	21.0000	19.7499	1.2501
9.0000	33.0000	28.9898	4.0102
10.0000	44.0000	42.5524	1.4476

5. Логарифмическая функция: y = a·log(x) + b

Коэффициенты: a = 14.6532, b = -8.6929

Среднеквадратичное отклонение (SSE): 875.0677

R^2 = 0.5427 (удовлетворительно объясняет данные (0.5 ≤ R^2 < 0.7))

x	y	f(x)	eps
1.0000	1.0000	-8.6929	9.6929
2.0000	2.4000	1.4640	0.9360
3.0000	5.0000	7.4054	-2.4054
4.0000	3.0000	11.6208	-8.6208
5.0000	7.0000	14.8906	-7.8906
6.0000	8.0000	17.5622	-9.5622
7.0000	10.0000	19.8210	-9.8210
8.0000	21.0000	21.7777	-0.7777
9.0000	33.0000	23.5036	9.4964
10.0000	44.0000	25.0475	18.9525

6. Степенная функция: y = a * x^b

Коэффициенты: a = 0.7508, b = 1.5296

Среднеквадратичное отклонение (SSE): 533.7613

$R^2 = 0.7273$ (хорошо объясняет данные ($0.7 \le R^2 < 0.9$))

x	y	f(x)	eps
1.0000	1.0000	0.7508	0.2492
2.0000	2.4000	2.1675	0.2325
3.0000	5.0000	4.0299	0.9701
4.0000	3.0000	6.2575	-3.2575
5.0000	7.0000	8.8030	-1.8030
6.0000	8.0000	11.6344	-3.6344
7.0000	10.0000	14.7280	-4.7280
8.0000	21.0000	18.0654	2.9346
9.0000	33.0000	21.6316	11.3684
10.0000	44.0000	25.4143	18.5857

!("/exit" to quit) Введите команду: => 7

Отправка: 7

Введите путь к файлу: table.txt

!("/exit" to quit) Введите команду: => 4

Отправка: 4

Выберете тип функции для исследования:

1. Линейная функция

2. Полиномиальная функция 2-й степени

3. Полиномиальная функция 3-й степени

4. Экспоненциальная функция

5. Логарифмическая функция

6. Степенная функция

7. Исследовать все типы функции

Введите тип функции: 7

Коэффициент корреляции Пирсона: 0.5759608182611893

1.3592901725153945 * e^0.39911804199690903x

Наилучшая модель:

Тип: 3. Полиномиальная функция 3-й степени: y = a·x^3 + b·x^2 + c·x + d

Коэффициенты: a = -0.1314, b = 1.2695, c = 4.3584, d = -6.1600

R^2: 0.4281

Интерпретация: плохо объясняет данные ($R^2 < 0.5$)

1. Линейная функция: y = a + b·x

Коэффициенты: a = 4.4739, b = 2.3333

Среднеквадратичное отклонение (SSE): 3326.5930

$R^2 = 0.3317$ (плохо объясняет данные ($R^2 < 0.5$))

x	y	f(x)	eps
1.0000	1.0000	6.8073	-5.8073
2.0000	3.4000	11.2812	-7.8812
3.0000	2.0000	15.7552	-13.7552
4.0000	63.0000	20.2291	42.7709
5.0000	2.0000	24.7030	-22.7030
6.0000	30.0000	29.1770	0.8230
7.0000	50.0000	33.6509	16.3491

8.0000	51.0000	38.1248	12.8752
9.0000	33.0000	42.5988	-9.5988
10.0000	34.0000	47.0727	-13.0727

2. Полиномиальная функция 2-й степени: y = a·x^2 + b·x + c

Коэффициенты: a = -0.8985, b = 14.3573, c = -17.4333

Среднеквадратичное отклонение (SSE): 2900.3518

R^2 = 0.4174 (плохо объясняет данные (R^2 < 0.5))

x	y	f(x)	eps
1.0000	1.0000	-3.9745	4.9745
2.0000	3.4000	7.6873	-4.2873
3.0000	2.0000	17.5521	-15.5521
4.0000	63.0000	25.6200	37.3800
5.0000	2.0000	31.8909	-29.8909
6.0000	30.0000	36.3648	-6.3648
7.0000	50.0000	39.0418	10.9582
8.0000	51.0000	39.9218	11.0782
9.0000	33.0000	39.0048	-6.0048
10.0000	34.0000	36.2909	-2.2909

3. Полиномиальная функция 3-й степени: $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$

Коэффициенты: a = -0.1314, b = 1.2695, c = 4.3584, d = -6.1600

Среднеквадратичное отклонение (SSE): 2847.0281

R^2 = 0.4281 (плохо объясняет данные ($R^2 < 0.5$))

x	y	f(x)	eps
1.0000	1.0000	-0.6635	1.6635
2.0000	3.4000	6.5836	-3.1836
3.0000	2.0000	14.7929	-12.7929
4.0000	63.0000	23.1761	39.8239
5.0000	2.0000	30.9449	-28.9449
6.0000	30.0000	37.3109	-7.3109
7.0000	50.0000	41.4857	8.5143
8.0000	51.0000	42.6810	8.3190
9.0000	33.0000	40.1085	-7.1085
10.0000	34.0000	32.9799	1.0201

4. Экспоненциальная функция: $y = a \cdot e^{(bx)}$

\---

Коэффициенты: a = 1.3593, b = 0.3991

Среднеквадратичное отклонение (SSE): 6392.6529

$R^2 = -0.2228$ (плохо объясняет данные ($R^2 < 0.5$))

x	y	f(x)	eps
1.0000	1.0000	2.0260	-1.0260
2.0000	3.4000	3.0198	0.3802
3.0000	2.0000	4.5011	-2.5011
4.0000	63.0000	6.7089	56.2911
5.0000	2.0000	9.9997	-7.9997
6.0000	30.0000	14.9046	15.0954
7.0000	50.0000	22.2155	27.7845
8.0000	51.0000	33.1124	17.8876
9.0000	33.0000	49.3543	-16.3543
10.0000	34.0000	73.5631	-39.5631

5. Логарифмическая функция: $y = a \cdot \log(x) + b$

\---

Коэффициенты: a = 19.8706, b = -3.0734

Среднеквадратичное отклонение (SSE): 3068.5060

$R^2 = 0.3836$ (плохо объясняет данные ($R^2 < 0.5$))

x	y	f(x)	eps
1.0000	1.0000	-3.0734	4.0734
2.0000	3.4000	10.6999	-7.2999
3.0000	2.0000	18.7567	-16.7567
4.0000	63.0000	24.4731	38.5269
5.0000	2.0000	28.9071	-26.9071
6.0000	30.0000	32.5300	-2.5300
7.0000	50.0000	35.5930	14.4070
8.0000	51.0000	38.2464	12.7536
9.0000	33.0000	40.5868	-7.5868
10.0000	34.0000	42.6804	-8.6804

6. Степенная функция: y = a * x^b

Коэффициенты: a = 0.8964, b = 1.7289

Среднеквадратичное отклонение (SSE): 4262.4097

$R^2 = 0.2180$ (плохо объясняет данные ($R^2 < 0.5$))

x	y	f(x)	eps

1.0000	1.0000	0.8964	0.1036
2.0000	3.4000	2.9715	0.4285
3.0000	2.0000	5.9900	-3.9900
4.0000	63.0000	9.8500	53.1500
5.0000	2.0000	14.4873	-12.4873
6.0000	30.0000	19.8557	10.1443
7.0000	50.0000	25.9198	24.0802
8.0000	51.0000	32.6509	18.3491
9.0000	33.0000	40.0252	-7.0252
10.0000	34.0000	48.0225	-14.0225

Выводы

В рамках лабораторной работы были проведены аппроксимации функций с применением различных методов: линейного, квадратичного, кубического, экспоненциального и логарифмического. Для этого был разработан Python-скрипт, использующий метод наименьших квадратов, который строит графики исходной функции и её аппроксимаций.

В результате работы удалось определить наиболее точное приближение, а также рассчитать среднеквадратические отклонения и коэффициент корреляции Пирсона для линейной зависимости.