

Университет ИТМО
Мегафакультет компьютерных технологий и управления
Факультет программной инженерии и компьютерной техники



ЛАБОРАТОРНАЯ РАБОТА №4
АППРОКСИМАЦИЯ ФУНКЦИИ МЕТОДОМ НАИМЕНЬШИХ
КВАДРАТОВ
Вариант №10

Группа: Р3211
Студент: Орчиков Даниил Валерьевич
Преподаватель: Малышева Татьяна Алексеевна

г. Санкт-Петербург
2024

Оглавление

Цель работы.....	2
Вычислительная реализация задачи:.....	2
Рабочие формулы	2
Решение.....	2
Линейное приближение	2
Квадратичное приближение	3
Программная реализация задачи	4
Листинг программы	4
Примеры и результаты работы программы	7
Пример 1	7
Пример 2.....	8
Пример 3.....	10
Вывод.....	12

Цель работы

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

Вычислительная реализация задачи:

Рабочие формулы

$$S = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n [\varphi(x_i) - y_i]^2 \rightarrow \min$$
$$\delta = \sqrt{\frac{\sum_{i=1}^n (\varphi(x_i) - y_i)^2}{n}}$$

Решение

$$y = \frac{18x}{x^4 + 10}, \quad x \in [0, 4]$$

	1	2	3	4	5	6	7	8	9	10	11
X	0	0.4	0.8	1.2	1.6	2	2.4	2.8	3.2	3.6	4
Y	0	0.718	1.383	1.789	1.74	1.385	1.001	0.705	0.501	0.364	0.271

Линейное приближение

$$SX = \sum_{i=1}^n x_i, \quad SXX = \sum_{i=1}^n x_i^2, \quad SY = \sum_{i=1}^n y_i, \quad SXY = \sum_{i=1}^n x_i y_i$$
$$SX = 22, \quad SXX = 61, \quad SY = 9.857, \quad SXY = 17.468$$

$$\begin{cases} aSXX + bSX = SXY \\ aSX + bn = SY \end{cases}$$

$$\Delta = SXX * n - SX^2 = 193.6$$

$$\Delta_1 = SXY * n - SX * SY = -24.707$$

$$\Delta_2 = SXX * SY - SX * SXY = 222.897$$

$$a = \frac{\Delta_1}{\Delta} = -\frac{24.707}{193.6} = -0.128$$

$$b = \frac{\Delta_2}{\Delta} = \frac{222.897}{193.6} = 1.151$$

$$P_{1(x)} = -0.128x + 1.151$$

	1	2	3	4	5	6	7	8	9	10	11
X	0	0.4	0.8	1.2	1.6	2	2.4	2.8	3.2	3.6	4
Y	0	0.718	1.383	1.789	1.74	1.385	1.001	0.705	0.501	0.364	0.271
$P_1(x)$ $= ax + b$	1.151	1.1	1.049	0.997	0.946	0.895	0.844	0.793	0.741	0.69	0.639
ε_i	1.151	0.382	-0.335	-0.792	-0.794	-0.49	-0.157	0.087	0.24	0.326	0.368

$$\begin{aligned} \delta &= \sqrt{\frac{(1.151 - 0.0)^2 + (1.1 - 0.718)^2 + (1.049 - 1.383)^2 + (0.997 - 1.789)^2 + (0.946 - 1.74)^2 + (0.895 - 1.385)^2 + (0.844 - 1.001)^2 + (0.793 - 0.705)^2 + (0.741 - 0.501)^2 + (0.69 - 0.364)^2 + (0.639 - 0.271)^2}{11}} \\ &= \sqrt{\frac{1.325 + 0.146 + 0.112 + 0.627 + 0.63 + 0.24 + 0.025 + 0.008 + 0.058 + 0.106 + 0.136}{11}} \\ &= 0.557 \end{aligned}$$

Квадратичное приближение

$$\sum_{i=1}^n x_i = 22, \quad \sum_{i=1}^n x_i^2 = 61.6, \quad \sum_{i=1}^n x_i^3 = 193.6, \quad \sum_{i=1}^n x_i^4 = 648.525$$

$$\sum_{i=1}^n y_i = 9.857, \quad \sum_{i=1}^n y_i x_i = 17.468, \quad \sum_{i=1}^n y_i x_i^2 = 39.046$$

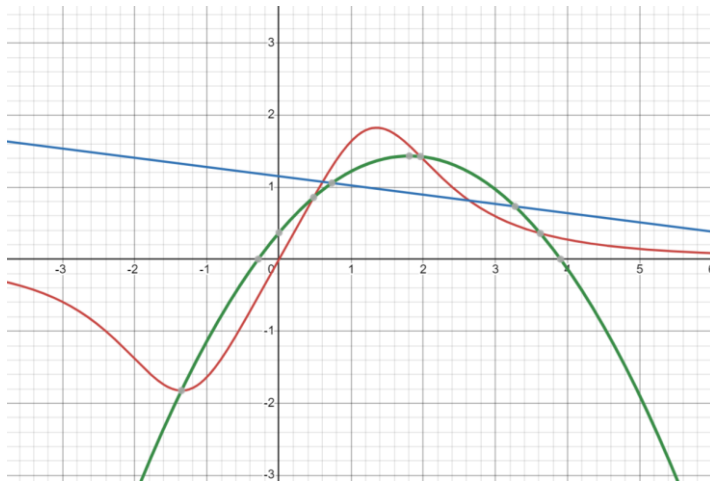
$$\begin{cases} 11a_0 + 22a_1 + 61.6a_2 = 9.857 \\ 22a_0 + 61.6a_1 + 193.6a_2 = 17.468 \\ 61.6a_0 + 193.6a_1 + 648.525a_2 = 39.046 \end{cases}$$

$$a_0 = 0.368, \quad a_1 = 1.178, \quad a_2 = -0.326$$

$$P_2(x) = -0.326x^2 + 1.178x + 0.368$$

	1	2	3	4	5	6	7	8	9	10	11
X	0	0.4	0.8	1.2	1.6	2	2.4	2.8	3.2	3.6	4
Y	0	0.718	1.383	1.789	1.74	1.385	1.001	0.705	0.501	0.364	0.271
$P_1(x)$ $= ax + b$	0.368	0.787	1.102	1.312	1.418	1.42	1.317	1.111	0.799	0.384	-0.136
ε_i	0.368	0.069	-0.282	-0.477	-0.322	0.035	0.317	0.405	0.298	0.02	-0.407

$$\delta = 0.312$$



Программная реализация задачи

Листинг программы

Представлен только код, непосредственно выполняющий вычисления

Весь код можно посмотреть [тут \(GitHub\)](#)

```
function polynomial(m) {
  m += 1
  let A = create2DArray(m, m)
  for (let i = 0; i < m; i++) {
    for (let j = 0; j < m; j++) {
      let s = 0
      XY.forEach(x => s += x[0] ** (i + j))
      A[i][j] = s
    }
  }
  A[0][0] = n
  let B = Array(m)
  for (let i = 0; i < m; i++) {
    let s = 0
    XY.forEach(x => s += x[0] ** i * x[1])
    B[i] = s
  }
  let coeffs = solveLinearSystem(A, B);
  let w = getting_rid_of_scientific_notation(coeffs)

  let latex_fun = []
  for (let i = 0; i < m; i++) {
    latex_fun.push(w[i] + "*x^" + i)
  }
  latex_fun = latex_fun.join("+")
  calculator.setExpression({id: 'graph' + m, latex: `f_{${m - 1}}=` + latex_fun})
  let S = 0
  latex_fun = latex_fun.replaceAll("^", "**").replaceAll("{", "(").replaceAll("}", ")")
  XY.forEach(x => S += (eval(latex_fun.replaceAll("x", `${x[0]}`))) - x[1] ** 2)
  document.getElementById("res").innerHTML += `S_{$m - 1} = ${Math.round(S * 10000) / 10000}<br>`
  draw_table(m - 1, latex_fun)
  return [coeffs, S]
}

function exponential() {
  let sx = 0, sxx = 0, sy = 0, sxy = 0
  XY.forEach(x => {
    sx += x[0]
    sxx += x[0] ** 2
    sy += Math.log(x[1])
  })
}
```

```

    sxy += x[0] * Math.log(x[1])
  })
  let A = [
    [n, sx],
    [sx, sxx]
  ]
  let B = [sy, sxy]
  let coeffs = solveLinearSystem(A, B)
  coeffs[0] = Math.exp(coeffs[0])
  let w = getting_rid_of_scientific_notation(coeffs)
  calculator.setExpression({id: 'graph6', latex: `f_e=${w[0]}*e^{{w[1]}x}`})
  let latex_fun = `${w[0]}*Math.E ** (${w[1]}*x)`.replaceAll("{", "(").replaceAll(")", ")")
  let S = 0
  XY.forEach(x => S += (eval(latex_fun.replaceAll("x", `${x[0]}`)) - x[1]) ** 2)
  if (isNaN(S))
    document.getElementById("res").innerHTML += `S_e - невозможно аппроксимировать с помощью экспоненциальной
функции<br>`
  else
    document.getElementById("res").innerHTML += `S_e = ${Math.round(S * 10000) / 10000}<br>`
  draw_table("e", latex_fun)
  return [coeffs, S]
}

function logarithmic() {
  let sx = 0, sxx = 0, sy = 0, sxy = 0
  XY.forEach(x => {
    sx += Math.log(x[0])
    sxx += Math.log(x[0]) ** 2
    sy += x[1]
    sxy += Math.log(x[0]) * x[1]
  })
  let A = [
    [n, sx],
    [sx, sxx]
  ]
  let B = [sy, sxy]
  let coeffs = solveLinearSystem(A, B)
  let w = getting_rid_of_scientific_notation(coeffs)
  calculator.setExpression({id: 'graph7', latex: `f_log=${w[1]}*\ln{x}+${w[0]}`})
  let latex_fun = `${w[1]}*Math.log(x) + ${w[0]}`.replaceAll("{", "(").replaceAll(")", ")")
  let S = 0
  XY.forEach(x => S += (eval(latex_fun.replaceAll("x", `${x[0]}`)) - x[1]) ** 2)
  if (isNaN(S))
    document.getElementById("res").innerHTML += `S_log - невозможно аппроксимировать с помощью логарифмической
функции<br>`
  else
    document.getElementById("res").innerHTML += `S_log = ${Math.round(S * 10000) / 10000}<br>`
  draw_table("log", latex_fun)
  return [coeffs, S]
}

function power() {
  let sx = 0, sxx = 0, sy = 0, sxy = 0
  XY.forEach(x => {
    sx += Math.log(x[0])
    sxx += Math.log(x[0]) ** 2
    sy += Math.log(x[1])
    sxy += Math.log(x[0]) * Math.log(x[1])
  })
  let A = [
    [n, sx],
    [sx, sxx]
  ]
  let B = [sy, sxy]
  let coeffs = solveLinearSystem(A, B)
  coeffs[0] = Math.exp(coeffs[0])
  let w = getting_rid_of_scientific_notation(coeffs)
  calculator.setExpression({id: 'graph8', latex: `f_power=${w[0]}*x^{{w[1]}`})
  let latex_fun = `${w[0]}*x** (${w[1]}`.replaceAll("{", "(").replaceAll(")", ")")

```

```

let S = 0
XY.forEach(x => S += (eval(latex_fun.replaceAll("x", `${x[0]}`)) - x[1]) ** 2)
if (isNaN(S))
    document.getElementById("res").innerHTML += `S_power - невозможно аппроксимировать с помощью степенной функции<br>`
else
    document.getElementById("res").innerHTML += `S_power = ${Math.round(S * 10000) / 10000}<br>`
draw_table("power", latex_fun)
return [coeffs, S]
}

// Решение системы линейных уравнений Ax = B методом Гаусса
function solveLinearSystem(A, B) {
    let n = A.length;

    for (let i = 0; i < n; i++) {
        // Поиск главного элемента
        let maxEl = Math.abs(A[i][i]);
        let maxRow = i
        for (let k = i + 1; k < n; k++)
            if (Math.abs(A[k][i]) > maxEl) {
                maxEl = Math.abs(A[k][i])
                maxRow = k
            }

        // Меняем местами строки
        for (let k = i; k < n; k++) {
            let tmp = A[maxRow][k]
            A[maxRow][k] = A[i][k]
            A[i][k] = tmp
        }
        let tmp = B[maxRow]
        B[maxRow] = B[i]
        B[i] = tmp

        // Приводим к нижнетреугольной форме
        for (let k = i + 1; k < n; k++) {
            let c = -A[k][i] / A[i][i];
            for (let j = i; j < n; j++) {
                if (i === j)
                    A[k][j] = 0
                else
                    A[k][j] += c * A[i][j]
            }
            B[k] += c * B[i]
        }
    }

    // Обратная подстановка
    let x = new Array(n).fill(0);
    for (let i = n - 1; i >= 0; i--) {
        x[i] = B[i] / A[i][i]
        for (let k = i - 1; k >= 0; k--) {
            B[k] -= A[k][i] * x[i]
        }
    }

    return x;
}

function create2DArray(n, m) {
    let array = new Array(n);

    for (let i = 0; i < n; i++)
        array[i] = new Array(m).fill(0);

    return array;
}

```

Примеры и результаты работы программы

Пример 1

Ввод:

1 2.7

2 7.38

3 20.08

4 54.59

5 148.41

6 403.29

-3 0.049

Выберите файл Файл не выбран

Введите точки

1 2.7
2 7.38
3 20.08
4 54.59
5 148.41
6 403.29
-3 0.049

S_1 = 68503.3146
S_2 = 17245.7372
S_3 = 1826.9186
S_4 = 90.5909
S_e = 1.5967

S_log - невозможно аппроксимировать с помощью логарифмической функции
S_power - невозможно аппроксимировать с помощью степенной функции

Наилучшее приближение:
экспоненциальное
 $f(x) = 1.002 * \exp(0.992 * x)$

Линейная аппроксимация

№	0	1	2	3	4	5	6
X	1	2	3	4	5	6	-3
Y	2.7	7.38	20.08	54.59	148.41	403.29	0.049
P_1	37.6547	71.5562	105.4576	139.359	173.2605	207.1619	-97.951
E_i	34.9547	64.1762	85.3776	84.769	24.8505	-196.1281	-98

r = 0.6884856457893946

Квадратичная аппроксимация

№	0	1	2	3	4	5	6
X	1	2	3	4	5	6	-3
Y	2.7	7.38	20.08	54.59	148.41	403.29	0.049
P_2	-53.2277	-18.4925	37.1429	113.6785	211.1143	329.4503	16.8333
E_i	-55.9277	-25.8725	17.0629	59.0885	62.7043	-73.8397	16.7843

Аппроксимация полиномом 3 степени

№	0	1	2	3	4	5	6
X	1	2	3	4	5	6	-3
Y	2.7	7.38	20.08	54.59	148.41	403.29	0.049
P_3	19.8935	-3.9816	0.8352	54.5208	177.2523	389.2066	-1.2277
E_i	17.1935	-11.3616	-19.2448	-0.0692	28.8423	-14.0834	-1.2767

Аппроксимация полиномом 4 степени

№	0	1	2	3	4	5	6
X	1	2	3	4	5	6	-3
Y	2.7	7.38	20.08	54.59	148.41	403.29	0.049
P_4	0.2978	12.5378	18.9022	49.2407	153.5323	401.8848	0.1034
E_i	-2.4022	5.1578	-1.1778	-5.3493	5.1223	-1.4052	0.0544

Аппроксимация экспоненциальной функцией

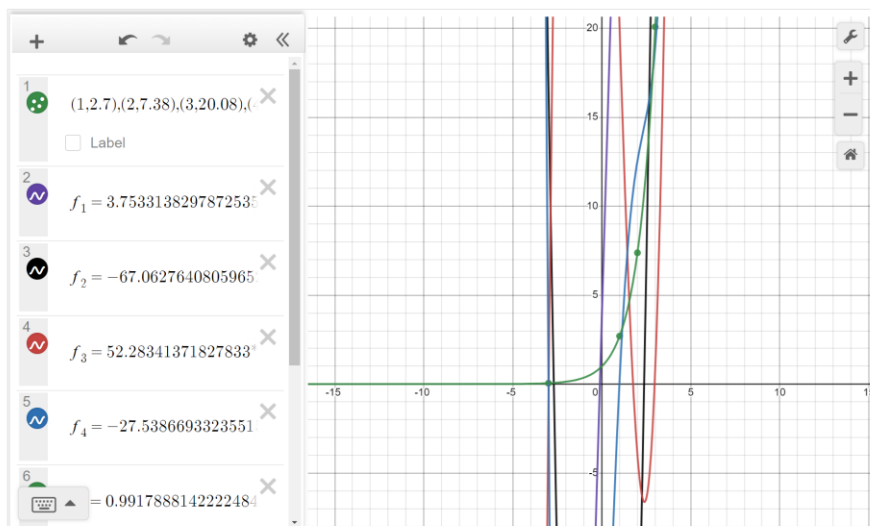
№	0	1	2	3	4	5	6
X	1	2	3	4	5	6	-3
Y	2.7	7.38	20.08	54.59	148.41	403.29	0.049
P_e	2.7009	7.3553	20.0305	54.5486	148.5506	404.5438	0.0491
E_i	0.0009	-0.0247	-0.0495	-0.0414	0.1406	1.2538	0.0001

Аппроксимация логарифмической функцией

№	0	1	2	3	4	5	6
X	1	2	3	4	5	6	-3
Y	2.7	7.38	20.08	54.59	148.41	403.29	0.049
P_log	NaN	NaN	NaN	NaN	NaN	NaN	NaN
E_i	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Аппроксимация степенной функцией

№	0	1	2	3	4	5	6
X	1	2	3	4	5	6	-3
Y	2.7	7.38	20.08	54.59	148.41	403.29	0.049
P_power	NaN	NaN	NaN	NaN	NaN	NaN	NaN
E_i	NaN	NaN	NaN	NaN	NaN	NaN	NaN



Пример 2

Ввод

1 0.1

2 0.69

3 1.09
4 1.38
5 1.6
6 1.79
7 1.94
8 2.08
0.5 -0.69
0.3 -1.2
0.002 -6.214
0.0000002 -15.42

Выберите файл Файл не выбран

Введите точки

1 0.1
2 0.69
3 1.09
4 1.38
5 1.6
6 1.79
7 1.94
8 2.08
0.5 -0.69
0.3 -1.2
0.002 -6.214
0.000002 -15.42

S_1 = 177.1212

S_2 = 133.5811

S_3 = 102.5348

S_4 = 76.8753

S_e - невозможно аппроксимировать с

помощью экспоненциальной функции

S_log = 0.0098

S_power - невозможно аппроксимировать с

помощью степенной функции

Наилучшее приближение:

логарифмическое

$f(x) = 1 * \log(x) + 0.006$

Линейная аппроксимация

Ni	0	1	2	3	4	5	6	7	8	9	10	11
X	1	2	3	4	5	6	7	8	0.5	0.3	0.002	2e-7
Y	0.1	0.69	1.09	1.38	1.6	1.79	1.94	2.08	-0.69	-1.2	-6.214	-15.42
P_1	-3.2839	-2.2133	-1.1427	-0.0721	0.9985	2.0691	3.1397	4.2103	-3.8193	-4.0334	-4.3524	-4.3546
E_i	-3.3839	-2.9033	-2.2327	-1.4521	-0.6015	0.2791	1.1997	2.1303	-3.1293	-2.8334	1.8616	11.0654

$r = 0.6097902920566524$

Квадратичная аппроксимация

Ni	0	1	2	3	4	5	6	7	8	9	10	11
X	1	2	3	4	5	6	7	8	0.5	0.3	0.002	2e-7
Y	0.1	0.69	1.09	1.38	1.6	1.79	1.94	2.08	-0.69	-1.2	-6.214	-15.42
P_2	-2.9837	-0.4753	1.3673	2.5441	3.0549	2.8999	2.079	0.5923	-4.4876	-5.1358	-6.151	-6.158
E_i	-3.0837	-1.1653	0.2773	1.1641	1.4549	1.1099	0.139	-1.4877	-3.7976	-3.9358	0.063	9.262

Аппроксимация полиномом 3 степени

Ni	0	1	2	3	4	5	6	7	8	9	10	11
X	1	2	3	4	5	6	7	8	0.5	0.3	0.002	2e-7
Y	0.1	0.69	1.09	1.38	1.6	1.79	1.94	2.08	-0.69	-1.2	-6.214	-15.42
P_3	-1.3264	1.8909	2.8574	2.4068	1.3729	0.5894	0.8901	3.1088	-4.0396	-5.3642	-7.6119	-7.6282
E_i	-1.4264	1.2009	1.7674	1.0268	-0.2271	-1.2006	-1.0499	1.0288	-3.3496	-4.1642	-1.3979	7.7918

Аппроксимация полиномом 4 степени

Ni	0	1	2	3	4	5	6	7	8	9	10	11
X	1	2	3	4	5	6	7	8	0.5	0.3	0.002	2e-7
Y	0.1	0.69	1.09	1.38	1.6	1.79	1.94	2.08	-0.69	-1.2	-6.214	-15.42
P_4	0.7185	2.7806	1.6943	0.4295	0.4855	1.8913	3.2053	1.5155	-2.8234	-4.917	-8.9018	-8.9323
E_i	0.6185	2.0906	0.6043	-0.9505	-1.1145	0.1013	1.2653	-0.5645	-2.1334	-3.717	-2.6878	6.4877

Аппроксимация экспоненциальной функцией

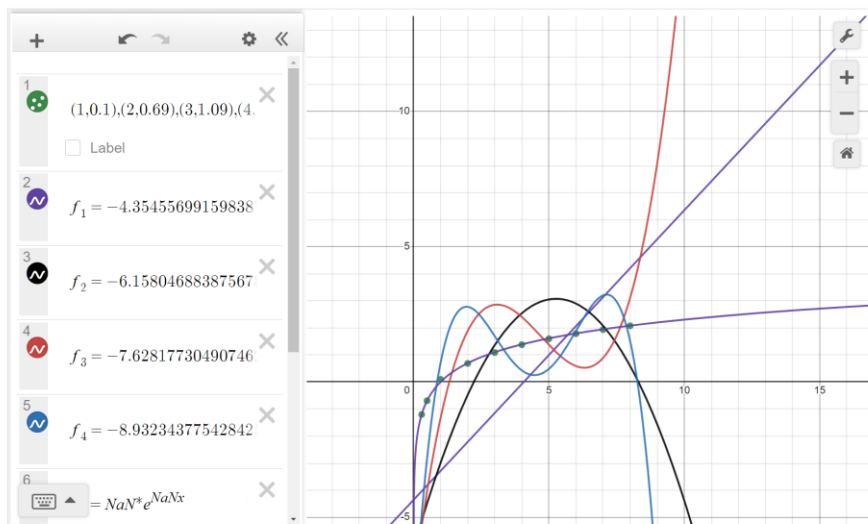
Ni	0	1	2	3	4	5	6	7	8	9	10	11
X	1	2	3	4	5	6	7	8	0.5	0.3	0.002	2e-7
Y	0.1	0.69	1.09	1.38	1.6	1.79	1.94	2.08	-0.69	-1.2	-6.214	-15.42
P_e	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
E_i	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Аппроксимация логарифмической функцией

Ni	0	1	2	3	4	5	6	7	8	9	10	11
X	1	2	3	4	5	6	7	8	0.5	0.3	0.002	2e-7
Y	0.1	0.69	1.09	1.38	1.6	1.79	1.94	2.08	-0.69	-1.2	-6.214	-15.42
P_log	0.0063	0.6993	1.1047	1.3923	1.6154	1.7977	1.9519	2.0854	-0.6867	-1.1974	-6.2072	-15.4158
E_i	-0.0937	0.0093	0.0147	0.0123	0.0154	0.0077	0.0119	0.0054	0.0033	0.0026	0.0068	0.0042

Аппроксимация степенной функцией

Ni	0	1	2	3	4	5	6	7	8	9	10	11
X	1	2	3	4	5	6	7	8	0.5	0.3	0.002	2e-7
Y	0.1	0.69	1.09	1.38	1.6	1.79	1.94	2.08	-0.69	-1.2	-6.214	-15.42
P_power	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
E_i	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN



Пример 3

Ввод

1 1

2 0.5

3 0.3333

4 0.25

5 0.2

6 0.16667

7 0.142

8 0.125

9 0.111111

10 0.1

Выберите файл

Файл не выбран

Введите точки

1 1

2 0.5

3 0.3333

4 0.25

5 0.2

6 0.16667

7 0.142

8 0.125

9 0.111111

10 0.1

S_1 = 0.2395

S_2 = 0.0707

S_3 = 0.0177

S_4 = 0.0037

S_e = 0.1538

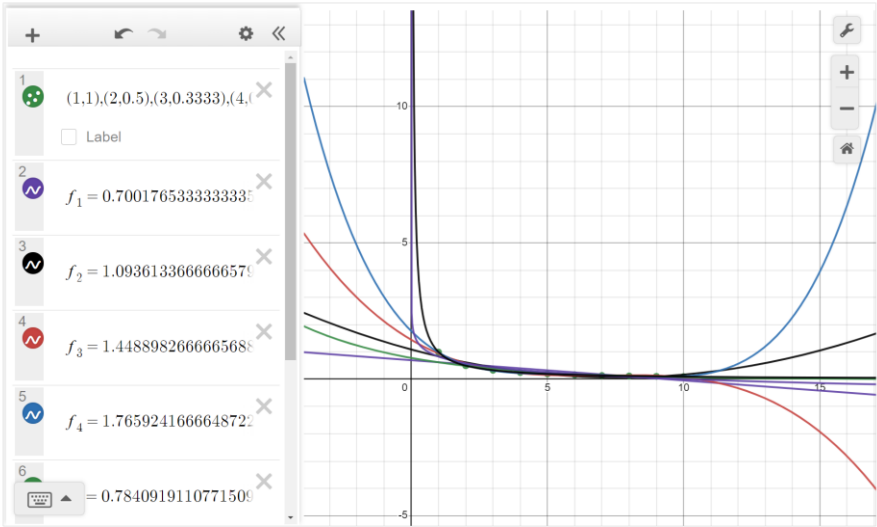
S_log = 0.0717

S_power = 0

Наилучшее приближение:
степенное
 $f(x) = 1 * x^{(-1.001)}$

Линейная аппроксимация

№	0	1	2	3	4	5	6	7	8	9
X	1	2	3	4	5	6	7	8	9	10
Y	1	0.5	0.3333	0.25	0.2	0.16667	0.142	0.125	0.111111	0.1
P_1	0.6261	0.552	0.478	0.4039	0.3298	0.2558	0.1817	0.1076	0.0336	-0.0405
E_i	-0.3739	0.052	0.1447	0.1539	0.1298	0.0891	0.0397	-0.0174	-0.0775	-0.1405



r = -0.8086409869814156

Квадратичная аппроксимация

№	0	1	2	3	4	5	6	7	8	9
X	1	2	3	4	5	6	7	8	9	10
Y	1	0.5	0.3333	0.25	0.2	0.16667	0.142	0.125	0.111111	0.1
P_2	0.8407	0.6236	0.4422	0.2966	0.1868	0.1127	0.0744	0.0719	0.1051	0.1741
E_i	-0.1593	0.1236	0.1089	0.0466	-0.0132	-0.054	-0.0676	-0.0531	-0.006	0.0741

Аппроксимация полиномом 3 степени

№	0	1	2	3	4	5	6	7	8	9
X	1	2	3	4	5	6	7	8	9	10
Y	1	0.5	0.3333	0.25	0.2	0.16667	0.142	0.125	0.111111	0.1
P_3	0.9451	0.5888	0.3553	0.2196	0.157	0.1425	0.1514	0.1588	0.1399	0.0698
E_i	-0.0549	0.0888	0.022	-0.0304	-0.043	-0.0241	0.0094	0.0338	0.0288	-0.0302

Аппроксимация полиномом 4 степени

№	0	1	2	3	4	5	6	7	8	9
X	1	2	3	4	5	6	7	8	9	10
Y	1	0.5	0.3333	0.25	0.2	0.16667	0.142	0.125	0.111111	0.1
P_4	0.985	0.54	0.3176	0.2262	0.1969	0.1824	0.1581	0.1211	0.0911	0.1097
E_i	-0.015	0.04	-0.0157	-0.0238	-0.0031	0.0158	0.0161	-0.0039	-0.02	0.0097

Аппроксимация экспоненциальной функцией

№	0	1	2	3	4	5	6	7	8	9
X	1	2	3	4	5	6	7	8	9	10
Y	1	0.5	0.3333	0.25	0.2	0.16667	0.142	0.125	0.111111	0.1
P_e	0.6227	0.4945	0.3927	0.3118	0.2476	0.1967	0.1562	0.124	0.0985	0.0782
E_i	-0.3773	-0.0055	0.0594	0.0618	0.0476	0.03	0.0142	-0.001	-0.0126	-0.0218

Аппроксимация логарифмической функцией

№	0	1	2	3	4	5	6	7	8	9
X	1	2	3	4	5	6	7	8	9	10
Y	1	0.5	0.3333	0.25	0.2	0.16667	0.142	0.125	0.111111	0.1
P_log	0.8338	0.5856	0.4403	0.3373	0.2573	0.192	0.1368	0.089	0.0468	0.0091
E_i	-0.1662	0.0856	0.107	0.0873	0.0573	0.0254	-0.0052	-0.036	-0.0643	-0.0909

Аппроксимация степенной функцией

№	0	1	2	3	4	5	6	7	8	9
X	1	2	3	4	5	6	7	8	9	10
Y	1	0.5	0.3333	0.25	0.2	0.16667	0.142	0.125	0.111111	0.1
P_power	1.0002	0.4999	0.3332	0.2499	0.1999	0.1665	0.1427	0.1249	0.111	0.0999
E_i	0.0002	-0.0001	-0.0001	-0.0001	-0.0001	-0.0001	0.0007	-0.0001	-0.0001	-0.0001

Вывод

Во время выполнения данной лабораторной работы я познакомился с методом наименьших квадратов для аппроксимации табличной функции с помощью функций и реализовал это на языке JavaScript.