Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №4**

по «Вычислительной математике»

Аппроксимация функции методом наименьших квадратов

Выполнил:

Студент группы P32151

Кортыш А.О.

Преподаватель:

Малышева Т.А.

Санкт-Петербург

2023

**Цель работы**

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

**Вычислительная реализация**

Функция . Интервал

| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0,0000 | 0,2000 | 0,4000 | 0,6000 | 0,8000 | 1,0000 | 1,2000 | 1,4000 | 1,6000 | 1,8000 | 2,0000 |
|  | 0,000  0 | 0,2399 | 0,4776 | 0,7018 | 0,8873 | 1,0000 | 1,0179 | 0,9501 | 0,8309 | 0,6969 | 0,5714 |
|  | 0,0000 | 0,0400 | 0,1600 | 0,3600 | 0,6400 | 1,0000 | 1,4400 | 1,9600 | 2,5600 | 3,2400 | 4,0000 |
|  | 0,0000 | 0,0080 | 0,0640 | 0,2160 | 0,5120 | 1,0000 | 1,7280 | 2,7440 | 4,0960 | 5,8320 | 8,0000 |
|  | 0,0000 | 0,0016 | 0,0256 | 0,1296 | 0,4096 | 1,0000 | 2,0736 | 3,8416 | 6,5536 | 10,4976 | 16,0000 |
|  | 0,0000 | 0,0480 | 0,1910 | 0,4211 | 0,7098 | 1,0000 | 1,2214 | 1,3301 | 1,3295 | 1,2544 | 1,1429 |
|  | 0,0000 | 0,0096 | 0,0764 | 0,2527 | 0,5679 | 1,0000 | 1,4657 | 1,8621 | 2,1271 | 2,2579 | 2,2857 |

**Линейная аппроксимация**

Получаем систему линейных уравнений:

Решая ее получаем:

a = 0.2897, b = 0.3807

| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0,0000 | 0,2000 | 0,4000 | 0,6000 | 0,8000 | 1,0000 | 1,2000 | 1,4000 | 1,6000 | 1,8000 | 2,0000 |
|  | 0,000 | 0,2399 | 0,4776 | 0,7018 | 0,8873 | 1,0000 | 1,0179 | 0,9501 | 0,8309 | 0,6969 | 0,5714 |
|  | 0,3807 | 0,43864 | 0,49658 | 0,55452 | 0,61246 | 0,6704 | 0,72834 | 0,78628 | 0,84422 | 0,90216 | 0,9601 |
|  | 0,3807 | 0,1987 | 0,0190 | -0,1473 | -0,2749 | -0,3296 | -0,2895 | -0,1638 | 0,0133 | 0,2053 | 0,3887 |

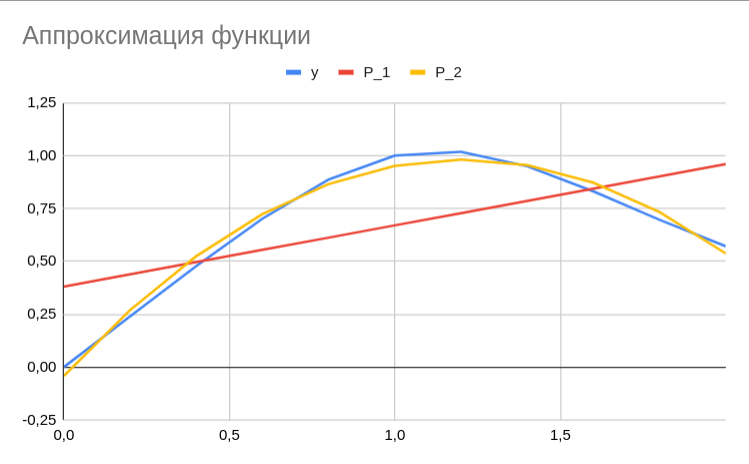
**Квадратичная аппроксимация**

Получаем систему уравнений:

Откуда получаем:

| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0,0000 | 0,2000 | 0,4000 | 0,6000 | 0,8000 | 1,0000 | 1,2000 | 1,4000 | 1,6000 | 1,8000 | 2,0000 |
|  | 0,000 | 0,2399 | 0,4776 | 0,7018 | 0,8873 | 1,0000 | 1,0179 | 0,9501 | 0,8309 | 0,6969 | 0,5714 |
|  | -0,042 | 0,26954 | 0,52472 | 0,72354 | 0,866 | 0,9521 | 0,98184 | 0,95522 | 0,87224 | 0,7329 | 0,5372 |
|  | -0,0420 | 0,0296 | 0,0472 | 0,0217 | -0,0213 | -0,0479 | -0,0360 | 0,0052 | 0,0413 | 0,0360 | -0,0342 |

Наилучшим приближением будет квадратичное, так как его среднеквадратичное отклонение наименьшее.



**Листинг программы**

def get\_polynom\_coefficients(x: List[float], y: List[float], deg: int) -> List[float]:

A = [[sum(x\_i\*\*(i+j) for x\_i in x) for i in range(deg+1)] for j in range(deg+1)]

B = [sum(x\_i\*\*i \* y\_i for x\_i, y\_i in zip(x, y)) for i in range(deg+1)]

det = determinant(A)

coefficients = []

for i in range(deg+1):

new\_A = [[A[j][k] if k != i else B[j] for k in range(deg+1)] for j in range(deg+1)]

cur\_det = determinant(new\_A)

coefficients += [cur\_det/det]

return coefficients

def get\_polynom(x: List[float], y: List[float], deg: int, precision: float) -> Tuple[Add, float]:

func: Add = Add(0)

for index, coef in enumerate(get\_polynom\_coefficients(x, y, deg)):

func += (var\*\*index)\*round(coef, precision)

return func, get\_standard\_deviation(func, x, y)

def get\_exponential(x: List[float], y: List[float], precision: float) -> Tuple[Add, float]:

lny = list(map(math.log, y))

a, b = get\_polynom\_coefficients(x, lny, 1)

func: Add = round(math.exp(a), precision)\*sympy.exp(round(b, precision)\*var)

return func, get\_standard\_deviation(func, x, y)

def get\_power(x: List[float], y: List[float], precision: float) -> Tuple[Add, float]:

lnx = list(map(math.log, x))

lny = list(map(math.log, y))

a, b = get\_polynom\_coefficients(lnx, lny, 1)

func: Add = round(math.exp(a), precision) \* var\*\*(round(b, precision))

return func, get\_standard\_deviation(func, x, y)

def get\_logarithmic(x: List[float], y: List[float], precision: float) -> Tuple[Add, float]:

lnx = list(map(math.log, x))

a, b = get\_polynom\_coefficients(lnx, y, 1)

func: Add = round(b, precision)\*sympy.ln(var) + round(a, precision)

return func, get\_standard\_deviation(func, x, y)

def get\_standard\_deviation(func: Add, x: List[float], y: List[float]) -> float:

callable\_function: Callable[[float], float] = lambdify(var, func, "numpy")

return (sum((callable\_function(x\_i) - y\_i) \*\* 2 for x\_i, y\_i in zip(x, y)) / len(x)) \*\* .5

def get\_extra\_information(func: Add, x: List[float], y: List[float]) -> Tuple[List[float], List[float]]:

callable\_function = lambdify(var, func, "numpy")

phi = [callable\_function(x\_i) for x\_i in x]

epsilon = [phi\_i - y\_i for phi\_i, y\_i in zip(phi, y)]

return phi, epsilon

def get\_pirson\_coefficient(x: List[float], y: List[float]) -> float:

mean\_x = sum(x) / len(x)

mean\_y = sum(y) / len(y)

return sum((x\_i - mean\_x)\*(y\_i - mean\_y) for x\_i, y\_i in zip(x, y)) /\

((sum((x\_i - mean\_x)\*\*2 for x\_i in x)\*sum((y\_i - mean\_y)\*\*2 for y\_i in y))\*\*.5)

**Примеры и результат работы программы**

**Пример 1**

Введите 0 для ввода из консоли, 1 для ввода из файла:

1

Введите имя файла:

3.txt

Введите 0 для вывода результатов в консоль, 1 для вывода в файл:

0

Текущая функция:

0.29\*var + 0.381

Среднеквадратичное отклонение:

0.251

Значения функции на текущих x:

0.381 0.439 0.497 0.555 0.613 0.671 0.729 0.787 0.845 0.903 0.961

Отклонение значений функции от реальных:

0.381 0.199 0.019 -0.147 -0.274 -0.329 -0.289 -0.163 0.014 0.206 0.39

Коэффициент Пирсона равен: 0.589

Текущая функция:

-0.704\*var\*\*2 + 1.698\*var - 0.042

Среднеквадратичное отклонение:

0.035

Значения функции на текущих x:

-0.042 0.269 0.525 0.723 0.866 0.952 0.982 0.955 0.873 0.733 0.538

Отклонение значений функции от реальных:

-0.042 0.03 0.047 0.022 -0.021 -0.048 -0.036 0.005 0.042 0.037 -0.033

Текущая функция:

-0.022\*var\*\*3 - 0.638\*var\*\*2 + 1.648\*var - 0.036

Среднеквадратичное отклонение:

0.035

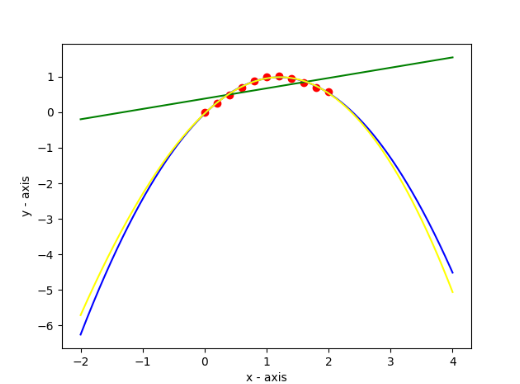
Значения функции на текущих x:

-0.036 0.268 0.52 0.718 0.863 0.952 0.985 0.96 0.877 0.735 0.532

Отклонение значений функции от реальных:

-0.036 0.028 0.042 0.017 -0.024 -0.048 -0.033 0.01 0.047 0.038 -0.039

Лучшая аппроксимирующая функция: -0.022\*var\*\*3 - 0.638\*var\*\*2 + 1.648\*var - 0.036

****

**Пример 2**

Введите 0 для ввода из консоли, 1 для ввода из файла:

1

Введите имя файла:

1.txt

Введите 0 для вывода результатов в консоль, 1 для вывода в файл:

0

Текущая функция:

1.685\*var + 1.217

Среднеквадратичное отклонение:

0.26

Значения функции на текущих x:

3.071 5.092 7.452 8.8 10.316 12.675 13.855

Отклонение значений функции от реальных:

0.341 -0.028 -0.288 -0.111 -0.274 -0.075 0.425

Коэффициент Пирсона равен: 0.997

Текущая функция:

-0.059\*var\*\*2 + 2.197\*var + 0.374

Среднеквадратичное отклонение:

0.1

Значения функции на текущих x:

2.719 5.115 7.695 9.066 10.517 12.585 13.533

Отклонение значений функции от реальных:

-0.011 -0.005 -0.045 0.156 -0.073 -0.165 0.103

Текущая функция:

-0.006\*var\*\*3 + 0.019\*var\*\*2 + 1.912\*var + 0.64

Среднеквадратичное отклонение:

0.092

Значения функции на текущих x:

2.758 5.065 7.671 9.082 10.574 12.634 13.518

Отклонение значений функции от реальных:

0.028 -0.055 -0.069 0.172 -0.016 -0.116 0.088

Текущая функция:

2.731\*exp(0.235\*var)

Среднеквадратичное отклонение:

1.249

Значения функции на текущих x:

3.537 4.689 6.515 7.863 9.715 13.5 15.914

Отклонение значений функции от реальных:

0.807 -0.431 -1.225 -1.047 -0.875 0.75 2.484

Текущая функция:

2.542\*var\*\*0.838

Среднеквадратичное отклонение:

0.148

Значения функции на текущих x:

2.753 5.109 7.609 8.965 10.445 12.671 13.756

Отклонение значений функции от реальных:

0.023 -0.011 -0.131 0.055 -0.145 -0.079 0.326

Текущая функция:

5.65\*log(var) + 1.199

Среднеквадратичное отклонение:

0.775

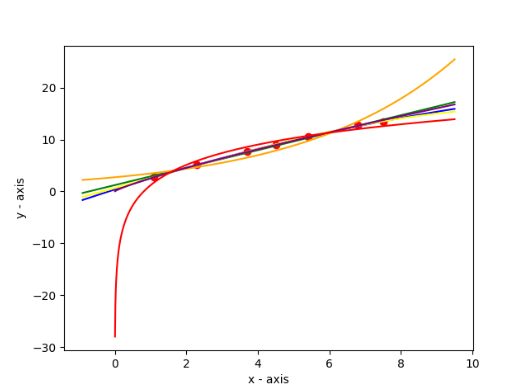
Значения функции на текущих x:

1.738 5.905 8.591 9.697 10.727 12.03 12.583

Отклонение значений функции от реальных:

-0.992 0.785 0.851 0.787 0.137 -0.72 -0.847

Лучшая аппроксимирующая функция: -0.006\*var\*\*3 + 0.019\*var\*\*2 + 1.912\*var + 0.64



**Пример 3**

Введите 0 для ввода из консоли, 1 для ввода из файла:

1

Введите имя файла:

2.txt

Введите 0 для вывода результатов в консоль, 1 для вывода в файл:

0

Текущая функция:

2.547\*var - 2.248

Среднеквадратичное отклонение:

2.603

Значения функции на текущих x:

0.554 3.61 7.176 9.213 11.506 15.072 16.855

Отклонение значений функции от реальных:

-2.946 -0.49 1.976 2.313 3.206 0.272 -4.345

Коэффициент Пирсона равен: 0.903

Текущая функция:

0.602\*var\*\*2 - 2.687\*var + 6.365

Среднеквадратичное отклонение:

0.861

Значения функции на текущих x:

4.138 3.369 4.664 6.464 9.41 15.93 20.075

Отклонение значений функции от реальных:

0.638 -0.731 -0.536 -0.436 1.11 1.13 -1.125

Текущая функция:

0.127\*var\*\*3 - 1.04\*var\*\*2 + 3.325\*var + 0.775

Среднеквадратичное отклонение:

0.368

Значения функции на текущих x:

3.343 4.466 5.273 6.25 8.402 15.228 20.791

Отклонение значений функции от реальных:

-0.157 0.366 0.073 -0.65 0.102 0.428 -0.409

Текущая функция:

2.15\*exp(0.28\*var)

Среднеквадратичное отклонение:

1.56

Значения функции на текущих x:

2.926 4.094 6.059 7.58 9.752 14.432 17.557

Отклонение значений функции от реальных:

-0.574 -0.006 0.859 0.68 1.452 -0.368 -3.643

Текущая функция:

2.351\*var\*\*0.868

Среднеквадратичное отклонение:

3.323

Значения функции на текущих x:

2.554 4.844 7.319 8.674 10.162 12.413 13.515

Отклонение значений функции от реальных:

-0.946 0.744 2.119 1.774 1.862 -2.387 -7.685

Текущая функция:

7.455\*log(var) - 0.825

Среднеквадратичное отклонение:

3.845

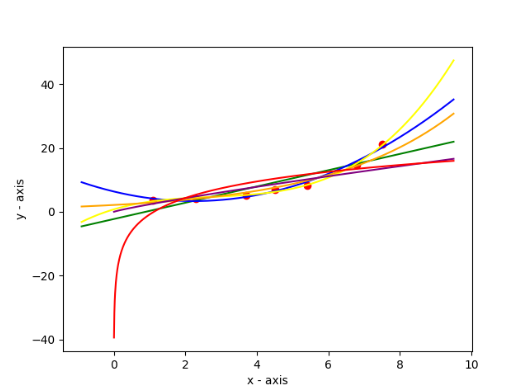
Значения функции на текущих x:

-0.114 5.384 8.929 10.388 11.747 13.466 14.196

Отклонение значений функции от реальных:

-3.614 1.284 3.729 3.488 3.447 -1.334 -7.004

Лучшая аппроксимирующая функция: 0.127\*var\*\*3 - 1.04\*var\*\*2 + 3.325\*var + 0.775



**Выводы**

Приятная лабораторная работа: очень понятные идеи, аппроксимация всех видов строится на основе решения СЛАУ, много разноцветных графиков и текста в выводе - эти и многие другие причины наслаждаться итогом этой лабораторной.