



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

ЛАБОРАТОРНАЯ РАБОТА № 3

ПО КУРСУ:

«*Численные методы*»

Студент *Наумов С. А.*

Преподаватель *Домрачева А.Б.*

Москва, 2024 г.

ОГЛАВЛЕНИЕ

1. Постановка задачи	3
2. Практическая реализация	4
3. Результат	4

1. Постановка задачи

По данному набору точек на плоскости (x, y) :

1. Построить эти точки на плоскости, а также аппроксимирующий многочлен степени количества этих точек - 1 – функцию $z(x)$.

Графики см. в 1 пункте части ‘Результат’

2. Пользуясь формулами для трех типов средних найти эти средние значения для переменных x , y и z :

$$\text{Арифметическое: } x_a = \frac{(x_0 + x_{n-1})}{2}$$

$$\text{Геометрическое: } x_g = \sqrt{x_0 x_{n-1}}$$

$$\text{Гармоническое: } x_h = \frac{2}{\frac{1}{x_0} + \frac{1}{x_{n-1}}}$$

3. Построить значения $\text{abs}(\{z(x_a), z(x_g), z(x_h)\} \times \{y_a, y_g, y_h\})$

Сопоставить наименьшее значение из этих 9 - одной из 9 формул аппроксимации

4. Составить систему уравнений для определения коэффициентов a, b формулы аппроксимации

5. Найти среднеквадратичное отклонение по заданным точкам множества x

2. Практическая реализация

Код реализации можно посмотреть на гитхабе:

[BMSTU/Numerical Methods/lab3.py at main · pear2jam/BMSTU \(github.com\)](https://github.com/pear2jam/BMSTU/blob/main/BMSTU/Numerical%20Methods/lab3.py)

Реализация лабораторной работы выполнена с помощью языка python и библиотек: numpy - для работы с массивами и математикой, и matplotlib - для работы с графиками

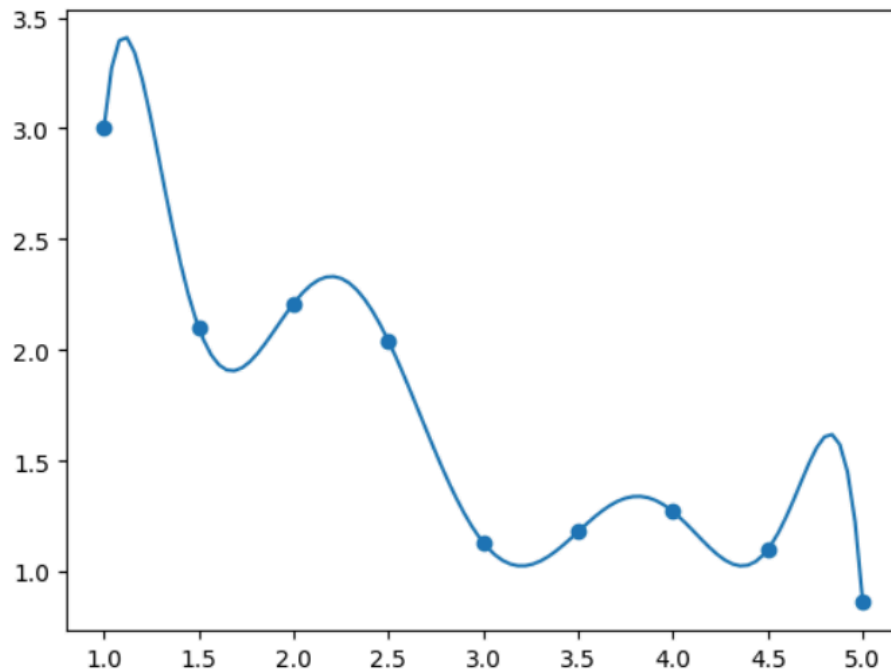
Реализация функций вычисления средних:

```
def calc_a(x, y):  
    return (x+y)/2  
  
def calc_g(x, y):  
    return np.sqrt(x*y)  
  
def calc_h(x, y):  
    return 2/(1/x+1/y)
```

С помощью функции numpy.polyfit был получен аппроксимирующий многочлен z для пары входных массивов (x, y) (Графики см. в 1 пункте части ‘Результат’)

3. Результат

1. Аппроксимирующий многочлен и соответствующие точки на графике:



2. Найдены средние значения для переменных x, y, z:

```
X: 3.0 2.23606797749979 1.66666666666667
Y: 1.93 1.606237840420901 1.3367875647668395
Z: 1.9299999999969373 1.6062378404336217 1.3367875647901342
```

Где средние идут в порядке: арифметическое, геометрическое, гармоническое.

3. Вычислены значения необходимых 9-ти значений:

```
1: 3.062661235730957e-12
2: 1.2720713371550119e-11
3: 0.32376215957603627
4: 0.3237621595663782
5: 0.5932124352098658
6: 0.5932124352300978
7: 2.3294699502685035e-11
8: 0.26945027563076684
9: 0.26945027566678226
```

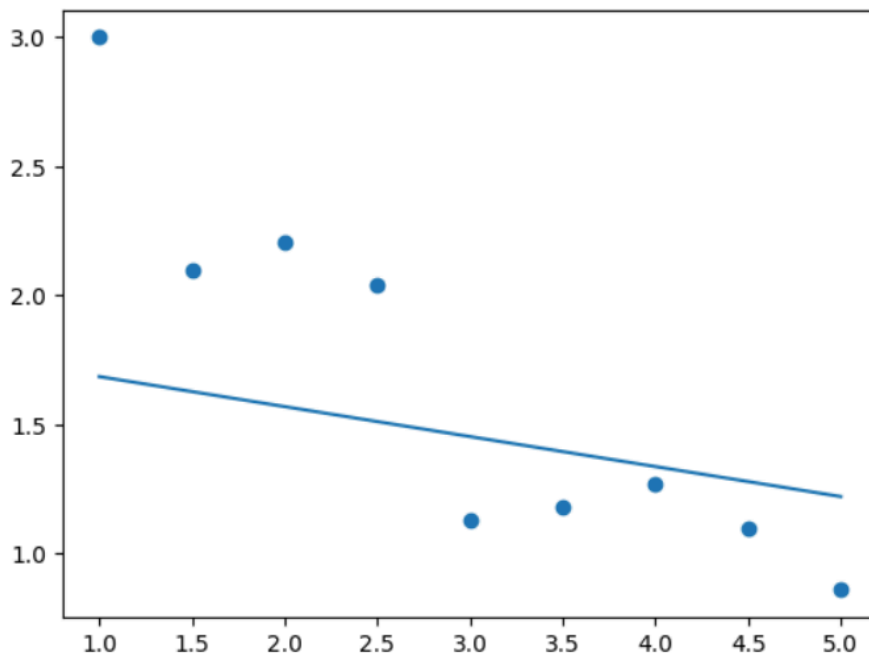
Минимальному значению (1) соответствует функция аппроксимации - линейная ($y(x) = ax+b$)

4. Составлена система уравнений для нахождения неизвестных коэффициентов a , b аппроксимирующей функции

```
a1, b1, c1 = sum(x*x), sum(x), sum(x*y)
a2, b2, c2 = sum(x), len(x)+1, sum(y)
```

```
a, b = np.linalg.solve(np.array([[a1, b1], [a2, b2]]), np.array([c1, c2]))
```

Получим следующую аппроксимацию данных:



5. Посчитаем среднее квадратичное отклонение:

```
np.mean((f(x)-y)*(f(x)-y))
```

```
0.3283677448573552
```