

МГТУ им. Н.Э. Баумана

Информатика, искусственный интеллект и системы управления

**отчет по заданию:**  
**Аппроксимация заданных величин**

Специалитет  
ИУ2

Выполнил студент 3 курса:  
**Александр Шмигельский**

Сентябрь 2023

# 1 Введение

Для анализа и аппроксимации данных, представленных в данной работе, был использован язык программирования Python. Для выполнения расчетов и построения графиков использовались две основные библиотеки:

- NumPy - библиотека для работы с массивами и матричными операциями, что позволяет эффективно обрабатывать и анализировать данные.
- Matplotlib - библиотека для визуализации данных, которая предоставляет множество инструментов для создания информативных графиков и диаграмм.

В данной работе будут представлены результаты аппроксимации данных о зависимости различных параметров от высоты. Для аппроксимации данных использовались полиномы определенной степени, а результаты представлены в виде графиков и таблиц с подробным описанием.

```
# Пробую реализовать метод наименьших квадратов для подбора коэффициентов
def least_squares_fit(x, y, degree, num_points=None):
    if num_points is None:
        num_points = len(x) # Используем все точки по умолчанию

    # Выбираем только первые num_points точек для аппроксимации
    x_subset = x[:num_points]
    y_subset = y[:num_points]

    # Создаем матрицу X со степенями x_subset
    X = np.vander(x_subset, degree + 1)

    # Решаем уравнение  $X^T * X * coeffs = X^T * y_subset$  для coeffs
    # Решение этой системы линейных уравнений производится для получения коэффициентов coeffs.
    coeffs = np.linalg.solve(np.dot(X.T, X), np.dot(X.T, y_subset))

    return coeffs

height_table = np.array([0, 500, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 11000, 12000, 13000])
g_table = np.array([9.8066, 9.8051, 9.8036, 9.8005, 9.7974, 9.7943, 9.7912, 9.7882, 9.7851, 9.7820, 9.7789, 9.7758, 9.7727, 9.7696, 9.7665])
data_dict_table = {height: g for height, g in zip(height_table, g_table)} # Создаем словарь

degree = 3 # Выбираем степень полинома

coeffs = least_squares_fit(height_table, g_table, degree)

# Создание полинома на основе коэффициентов
poly = np.poly1d(coeffs)

# Генерация значений для аппроксимации
# height_range = np.linspace(700, 8000, 100) # Будет 100 значений, расположенных через равные интервалы
height_range = np.arange(700, 8000 + 1, 10) # Интервал каждые 10 метров
g_approximated = poly(height_range)

# Визуализация результатов
plot_result(height_table, g_table, height_range, g_approximated, degree, 'Ускорение свободного падения (м/с^2)')

# Визуализация результатов поближе
plot_results_close(height_table, g_table, height_range, g_approximated, degree, 'Ускорение свободного падения (м/с^2)')

# Вывод коэффициентов полинома
print(f'Коэффициенты полинома: {coeffs}')

calculate_error(height_range, g_approximated, data_dict_table)
```

Рис. 1: Пример кода на языке Python для аппроксимации ускорения свободного падения

## 2 Аппроксимация ускорения свободного падения

В этой части отчета я рассмотрел аппроксимацию ускорения свободного падения на высотах 700м - 8000м.

### 2.1 Исходные данные

Для начала, рассмотрим исходные данные, представленные в таблице:

Высота (м)	Ускорение свободного падения (м/с <sup>2</sup> )
0	9.8066
500	9.8051
1000	9.8036
2000	9.8005
3000	9.7974
4000	9.7943
5000	9.7912
6000	9.7882
7000	9.7851
8000	9.7820
9000	9.7789
10000	9.7759
11000	9.7728
12000	9.7697
13000	9.7667
14000	9.7636
15000	9.7605
16000	9.7575
17000	9.7544
18000	9.7513
19000	9.7483
20000	9.7452

Таблица 1: Исходные данные

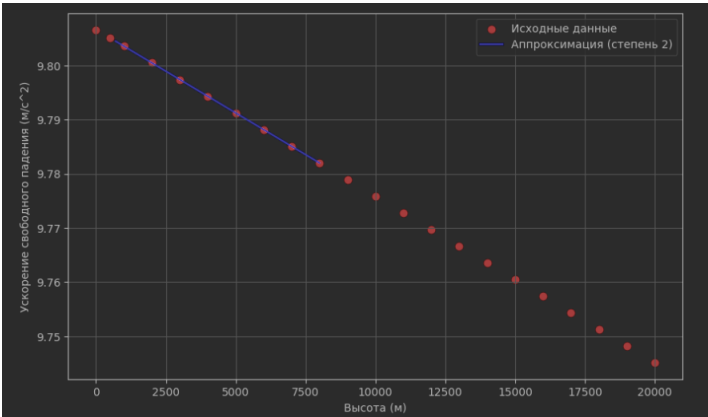
### 2.2 Аппроксимация полиномом

Затем были аппроксимированны данные полиномом второй степени. Коэффициенты полинома:

$$5.58847218 \times 10^{-13}, -3.08249938 \times 10^{-06}, 9.80663919.$$

$$f(h) = 5.58847218 \times 10^{-13}h^2 - 3.08249938 \times 10^{-06}h + 9.80663919$$

Для визуализации результатов аппроксимации, на рисунке 2 показаны исходные данные (красные точки) и аппроксимированный полином (синяя линия).



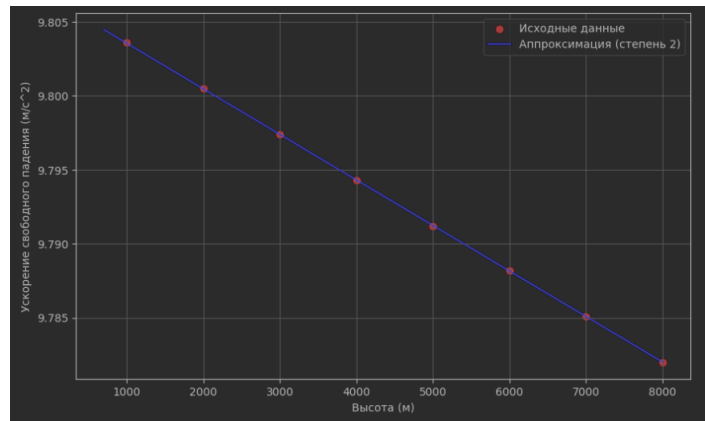


Рис. 2: Аппроксимация данных об ускорении свободного падения.

Также мною были рассмотрены погрешности аппроксимированной величины на различных высотах. Результаты представлены в таблице 2.

Высота (м)	Ошибка
1000	$4.275 \times 10^{-05}$
2000	$2.358 \times 10^{-05}$
3000	$3.283 \times 10^{-06}$
4000	$-1.813 \times 10^{-05}$
5000	$-4.066 \times 10^{-05}$
6000	$3.569 \times 10^{-05}$
7000	$1.093 \times 10^{-05}$
8000	$-1.496 \times 10^{-05}$

Таблица 2: Значения ошибок аппроксимации для определенных высот.

### 3 Скорость звука от высоты

#### 3.1 Исходные данные

Для анализа зависимости скорости звука от высоты были использованы следующие данные:

Высота (м)	Скорость звука (м/с)
0	340.294
500	338.370
1000	336.435
2000	332.532
3000	328.584
4000	324.589
5000	320.545
6000	316.452
7000	312.306
8000	308.105
9000	303.848
10000	299.532
11000	295.154
12000	295.069
13000	295.069
14000	295.069
15000	295.069
16000	295.065

Таблица 3: Исходные данные

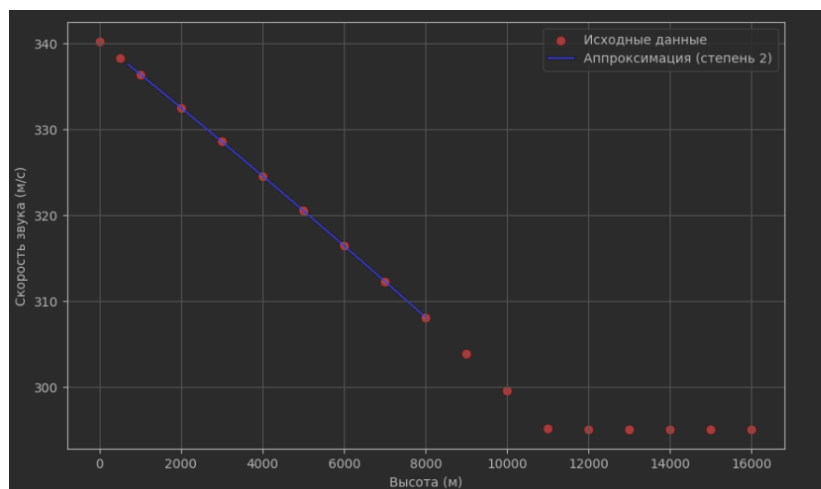
#### 3.2 Аппроксимация полиномом

Для аппроксимации данных был использован полином второй степени. Также для более точной аппроксимации на необходимых высотах были использованы только первые 10 значений. Коэффициенты полинома:

$$-2.43265270 \times 10^{-8}, -3.82778332e \times 10^{-3}, 3.40289478 \times 10^2$$

$$f(h) = -2.43265270 \times 10^{-8}h^2 - 3.82778332 \times 10^{-3}h + 340.289478$$

Визуализация результатов аппроксимации приведена на графиках ниже (рис. 3):



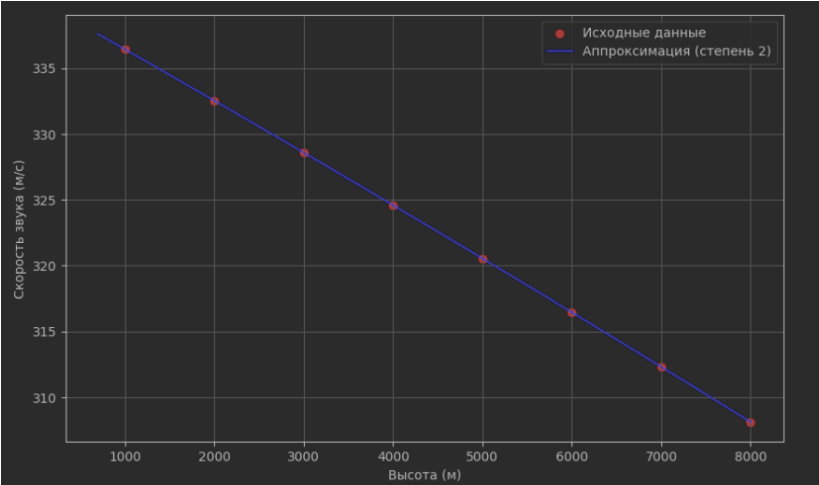


Рис. 3: Аппроксимация данных об ускорении свободного падения.

Значения ошибок:

Высота (м)	Ошибка
1000	−0.002368579305368712
2000	−0.004605680613508412
3000	−0.0031897278436758825
4000	−0.00012072099599436115
5000	0.002601339929583446
6000	0.004976454933000696
7000	0.0030046240142382885
8000	−0.005314152826542795

Таблица 4: Значения ошибок аппроксимации для определенных высот.

4 Плотность воздуха от высоты

4.1 Исходные данные

Для анализа зависимости плотности воздуха от высоты были использованы следующие данные:

Высота (м)	Плотность воздуха (кг/м³)
0	1.2255
500	1.16727
1000	1.11166
2000	1.00655
3000	0.909254
4000	0.819347
5000	0.736429
6000	0.660111
7000	0.590018
8000	0.526783
9000	0.467063
10000	0.41351
11000	0.364801
12000	0.311937
13000	0.266595
14000	0.227855
15000	0.194755
16000	0.16647
17000	0.142301
18000	0.121647
19000	0.103995
20000	0.0889097

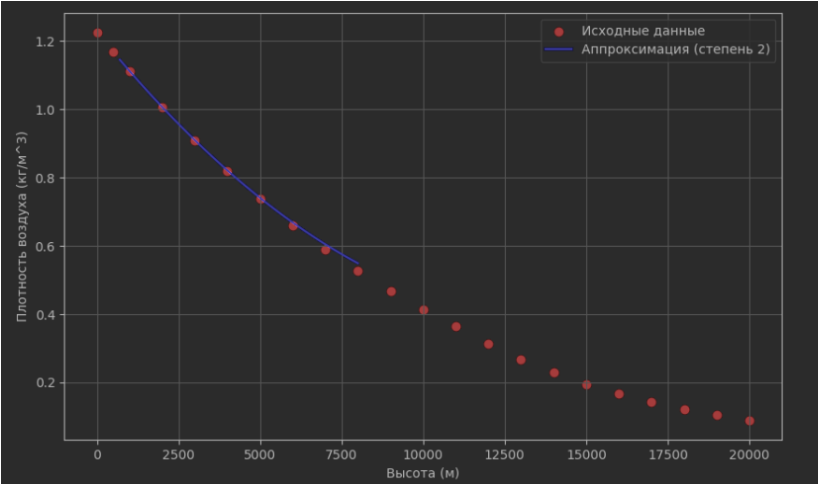
Таблица 5: Исходные данные

4.2 Аппроксимация полиномом

Для аппроксимации данных был использован полином второй степени. Коэффициенты полинома:

$$f(h) = 4.14972281 \times 10^{-9}h^2 - 1.17790055 \times 10^{-4}h + 1.22534716$$

Визуализация результатов аппроксимации приведена на графиках ниже (рис. 4):



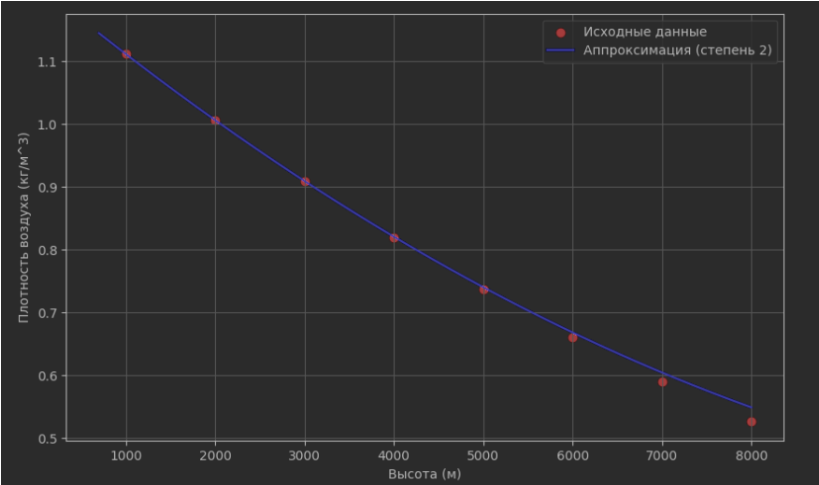


Рис. 4: Аппроксимация данных о плотности воздуха.

Значения ошибок:

Высота (м)	Ошибка
1000	$-4.6829424307226475 \times 10^{-05}$
2000	0.00018405756929595896
3000	$-7.050106609773898 \times 10^{-05}$
4000	-0.0012355053304877739
5000	-0.003710955223874768
6000	-0.007885850746258471
7000	-0.014135191897638899
8000	-0.021825978678016056

Таблица 6: Значения ошибок аппроксимации для определенных высот.



## 5 Вспомогательные функции

В данном разделе представлены вспомогательные функции, которые используются для анализа данных и аппроксимации в данной работе.

### 5.1 Функция `plot_result`

Функция `plot_result` предназначена для визуализации результатов аппроксимации данных. Она принимает следующие аргументы:

- `X_table` - массив значений высоты из таблицы исходных данных.
- `Y_table` - массив соответствующих значений из таблицы исходных данных.
- `X_range` - массив значений высоты для аппроксимации.
- `Y_approximated` - массив аппроксимированных значений.
- `degree` - степень полинома, используемого для аппроксимации.
- `y_label` - подпись оси Y на графике.

Функция строит график, на котором исходные данные представлены точками (красные точки), а аппроксимация полиномом указанной степени представлена линией (синяя линия). На графике также присутствует подпись осей, легенда и сетка.

### 5.2 Функция `plot_results_close`

Функция `plot_results_close` аналогична предыдущей, но используется для визуализации результатов более близких к определенному диапазону высот. Она также принимает аргументы для массивов и степени полинома, а также подпись оси Y.

### 5.3 Функция `calculate_error`

Функция `calculate_error` предназначена для расчета ошибок аппроксимации данных. Она принимает аргументы:

- `X_range` - массив значений высоты для аппроксимации.
- `Y_approximated` - массив аппроксимированных значений.
- `data_dict` - словарь с исходными данными (высотой и соответствующими значениями).

Функция считает ошибки между аппроксимированными данными и данными из словаря для определенных высот и выводит их на экран.

### 5.4 Функция `least_squares_fit`

Функция `least_squares_fit` выполняет аппроксимацию данных методом наименьших квадратов. Она принимает аргументы:

- `x` - массив значений X.
- `y` - массив значений Y.
- `degree` - степень полинома для аппроксимации.
- `num_points` - количество точек, используемых для аппроксимации (по умолчанию все точки).

Функция возвращает коэффициенты полинома, который наилучшим образом аппроксимирует данные.