



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 _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 1
по курсу «Численные методы линейной алгебры»
«Подготовка вспомогательных средств разработки»

Студент группы ИУ9-71Б Баев Д.А

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

1. Подготовить методы для выполнения операций с векторами: вычисление скалярного произведения двух векторов, вычисление Евклидовой нормы вектора.

2. Подготовить методы для выполнения операций с матрицами: умножение матрицы на матрицу, умножение матрицы на вектор, транспонирование матрицы.

3. Подготовить метод построения графика произвольной функции от одной переменной.

4. Реализовать оценку погрешности вычисления объема шара зажатого цилиндром тремя способами при двух различных приближениях вычисления значения

2 Исходный код

Исходный код программы представлен в листингах 1–4.

Листинг 1 — Операции над векторами

```
1 def scalar(vector1, vector2):
2     assert len(vector1) == len(vector2)
3     return sum(x * y for x, y in zip(vector1, vector2))
4
5 def norm(vector):
6     return np.sqrt(sum(x**2 for x in vector))
7
8 vector1 = [1, 2, 3]
9 vector2 = [4, 5, 6]
10
11 print(scalar(vector1, vector2))
12 print(norm(vector2))
```

Листинг 2 — Операции над матрицами

```
1 def mul_matrix(matrix1, matrix2):
2     assert len(matrix1[0]) == len(matrix2)
3     return [
4         [sum(matrix1[i][k] * matrix2[k][j] for k in range(len(matrix2[0])))
5          for j in range(len(matrix2[0]))]
6         for i in range(len(matrix1))]
7
8
9 def mul_matrix_by_vector(matrix, vector):
10     assert len(matrix[0]) == len(vector)
11     return [sum(matrix[i][j] * vector[j] for j in range(len(vector)))
12             for i in range(len(matrix))]
13
14 def transpose_matrix(matrix):
15     return [list(row) for row in zip(*matrix)]
16
17 matrix1 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
18 matrix2 = [[3, 4, 6], [1, 2, 7], [9, 8, 1]]
19 matrix3 = [[2, 3], [9, 4], [5, 6]]
20
21 print(mul_matrix(matrix1, matrix2))
22 print(mul_matrix_by_vector(matrix1, vector1))
23 print(transpose_matrix(matrix1))
24 print(transpose_matrix(matrix3))
```

Листинг 3 — Построение графика

```
1 def plot(function, x1, x2, step, x = None):
2     x = [x for x in range(x1, x2 + step, step)] if x is None else x
3     plt.plot(x, function(np.array(x)))
4     plt.show()
5
6 def f(x):
7     return x**2
8
9 plot(f, -10, 10, 1)
```

Листинг 4 — Оценка погрешности

```

1 def S1(sqrt):
2     return (sqrt - 1) ** 6
3
4 def S2(sqrt):
5     return (3 - 2 * sqrt) ** 3
6
7 def S3(sqrt):
8     return 99 - 70 * sqrt
9
10 def S1_derivative(sqrt):
11     return 6 * (sqrt - 1) ** 5
12
13 def S2_derivative(sqrt):
14     return -6 * (3 - 2 * sqrt) ** 2
15
16 def S3_derivative(sqrt):
17     return -70
18
19 def V(S):
20     return 4/3 * np.pi * S
21
22 R = 3
23
24 ways = {
25     "S1": [S1, S1_derivative],
26     "S2": [S2, S2_derivative],
27     "S3": [S3, S3_derivative]
28 }
29
30 sqrts = {
31     "7/5": 7/5,
32     "17/12": 17/12
33 }
34
35 S_list = []
36 V_list = []
37 estimation_list = []
38 for way, func in ways.items():
39     for sqrt, value in sqrts.items():
40         S = func[0](value)
41         V_calc = V(S)
42         estimation = abs(func[1](value) * (value - np.sqrt(2)) / S)
43         S_list.append(S)
44         V_list.append(V_calc)
45         estimation_list.append(estimation * 100)
46
47
48 table = pd.DataFrame(
49     {
50         "Way": sorted(list(ways.keys())) * 2,
51         "Sqrt": list(sqrts.keys()) * 3,
52         "S": S_list,
53         "V": V_list,
54         "Error estimation": estimation_list
55     }
56 )
57 print(table)

```

3 Результаты

Результат проверки операций над векторами представлен на рисунке 1.

```
32
8.774964387392123
```

Рис. 1 — Результат проверки операций над векторами

Результат проверки операций над матрицами представлен на рисунке 2.

```
[[32, 32, 23], [71, 74, 65], [110, 116, 107]]
[14, 32, 50]
[[1, 4, 7], [2, 5, 8], [3, 6, 9]]
[[2, 9, 5], [3, 4, 6]]
```

Рис. 2 — Результат проверки операций над матрицами

Результат проверки построения графика представлен на рисунке 3.

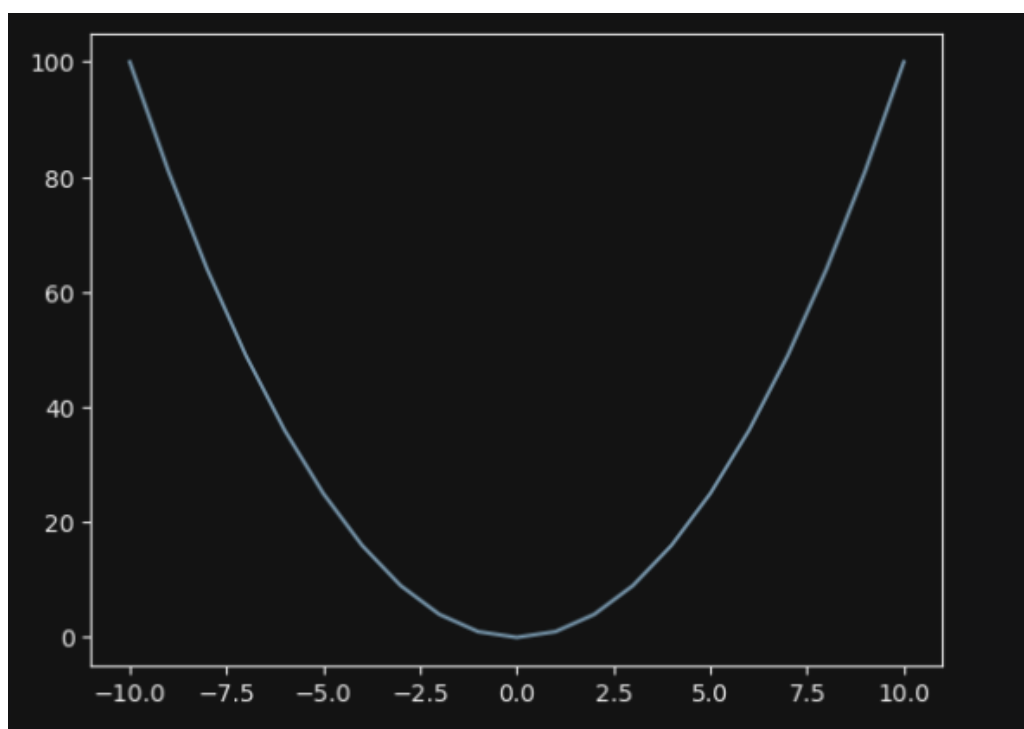


Рис. 3 — Результат проверки построения графика

Результат проверки оценки погрешности представлен на рисунке 4.

	Way	Sqrt	S	V	Error estimation
0	S1	7/5	0.004096	0.017157	21.320344
1	S1	17/12	0.005233	0.021919	3.532470
2	S2	7/5	0.008000	0.033510	42.640687
3	S2	17/12	0.004630	0.019393	8.831175
4	S3	7/5	1.000000	4.188790	99.494937
5	S3	17/12	-0.166667	-0.698132	103.030380

Рис. 4 — Результат проверки оценки погрешности

4 Выводы

В результате выполнения лабораторной работы были реализованы основные операции с векторами и матрицами, построение графика произвольной функции с помощью Matplotlib, а также метод оценки погрешности вычислений.

Анализируя результат оценки погрешности, можно отметить, что, при разных приближениях данных и при разных, но аналитически тождественных способах вычисления, результат может значительно отличаться.