



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»

Студентка группы ИУ9-72Б Самохвалова П. С.

Преподаватель Посевин Д. П.

Москва 2023

1 Цель работы

Подготовить методы необходимые для решения задач численных методов линейной алгебры.

2 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 $2^{0.5}$: $7/5$ и $17/12$.

3 Практическая реализация

Исходный код программы представлен в листингах 1– 2.

Листинг 1: Методы для выполнения операций с матрицами и векторами

```
1 from matplotlib import pyplot as plt
2
3
4 def scalar_mult_vec(vec1, vec2):
5     n = len(vec1)
6     s = 0
7     for i in range(n):
8         s += vec1[i] * vec2[i]
9     return s
10
11
```

```

12 def norm_vec(vec):
13     n = len(vec)
14     s = 0
15     for i in range(n):
16         s += vec[i] ** 2
17     norm = s ** 0.5
18     return norm
19
20
21 def mult_matr_matr(matr1, matr2):
22     n1 = len(matr1)
23     m1 = len(matr1[0])
24     m2 = len(matr2[0])
25     matr = [[0] * m2 for i in range(n1)]
26     for i in range(n1):
27         for j in range(m2):
28             for k in range(m1):
29                 matr[i][j] += matr1[i][k] * matr2[k][j]
30     return matr
31
32
33 def mult_matr_vec(matr, vec):
34     n = len(vec)
35     m = len(matr[0])
36     vec1 = [0] * n
37     for i in range(n):
38         for j in range(m):
39             vec1[i] += matr[i][j] * vec[j]
40     return vec1
41
42
43 def transp_matr(matr):
44     n = len(matr)
45     m = len(matr[0])
46     matr1 = [[0] * n for i in range(m)]
47     for i in range(m):
48         for j in range(n):
49             matr1[i][j] = matr[j][i]
50     return matr1
51
52
53 def plot_func(func, x):
54
55     y = [func(x[i]) for i in range(len(x))]
56
57     plt.xlabel('x')

```

```

58 plt.ylabel('y')
59 plt.grid()
60 plt.plot(x, y)
61 plt.show()

```

Листинг 2: Оценка погрешностей

```

1 import math
2
3 from num_methods import *
4
5
6 def calc_err(x):
7
8     true_value = 0.005051
9
10    v1 = (x - 1) ** 6
11    v2 = (3 - 2 * x) ** 3
12    v3 = 99 - 70 * x
13
14    print("Relative error at the root of 2, equal to", x)
15
16    print("First method -", abs(true_value - v1) / true_value)
17    print("Second method -", abs(true_value - v2) / true_value)
18    print("Third method -", abs(true_value - v3) / true_value)
19    print()
20
21
22 def calc_err1(x):
23
24    delta_x = abs(math.sqrt(2) - x)
25
26    v1 = 6 / (math.sqrt(2) - 1)
27    v2 = 6 / (3 - 2 * math.sqrt(2))
28    v3 = 70 / (99 - 70 * math.sqrt(2))
29
30    print("Error estimation with a root of 2, equal to", x)
31
32    print("First method -", v1 * delta_x)
33    print("Second method -", v2 * delta_x)
34    print("Third method -", v3 * delta_x)
35    print()
36
37
38 print(scalar_mult_vec([1, 2, 3], [4, 5, 6]))
39 print(norm_vec([1, 2, 3]))
40 print()

```

```

41
42 print(mult_matr_matr([[1, 2], [3, 4], [5, 6]], [[1, 2, 3, 4, 5],
43                                     [6, 7, 8, 9, 10]]))
44 print(mult_matr_vec([[1, 2, 3], [4, 5, 6], [7, 8, 9]], [1, 2, 3]))
45 print(transp_matr([[1, 2], [3, 4], [5, 6]]))
46 print()
47
48 plot_func(lambda x: x ** 2, [1, 2, 3, 4, 5])
49
50 calc_err(7 / 5)
51 calc_err(17 / 12)
52
53 calc_err1(7 / 5)
54 calc_err1(17 / 12)

```

4 Результаты

Результаты работы программы представлены на рисунках 1 – 3.

```

32
3.7416573867739413

[[13, 16, 19, 22, 25], [27, 34, 41, 48, 55], [41, 52, 63, 74, 85]]
[14, 32, 50]
[[1, 3, 5], [2, 4, 6]]

```

Рис. 1 — Результаты работы операций с матрицами и векторами

```

Relative error at the root of 2, equal to 1.4
First method - 0.18907147099584345
Second method - 0.5838447832112494
Third method - 196.98059790140567

Relative error at the root of 2, equal to 1.4166666666666667
First method - 0.03598908842427385
Second method - 0.08342315786386512
Third method - 33.99676631690188

```

Рис. 2 — Относительные погрешности

```
Error estimation with a root of 2, equal to 1.4
First method - 0.20588745030457453
Second method - 0.4970562748477211
Third method - 196.99494936626192

Error estimation with a root of 2, equal to 1.4166666666666667
First method - 0.03553390593273729
Second method - 0.08578643762690426
Third method - 33.999158227710424
```

Рис. 3 — Оценка погрешностей

5 Выводы

В результате выполнения лабораторной работы были реализованы методы для выполнения операций с матрицами и векторами, был реализован метод построения графика произвольной функции от одной переменной, была реализована оценка погрешности вычисления объема шара зажатого цилиндром тремя способами при двух различных приближениях вычисления значения $2^{0.5}$: $7/5$ и $17/12$.