

Отчет по лабораторной работе №3

Дисциплина: Научное программирование

Выполнила Дяченко Злата Константиновна, НПМмд-02-22

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Шаг 1	7
3.2	Шаг 2	8
3.3	Шаг 3	8
3.4	Шаг 4	9
3.5	Шаг 5	11
3.6	Шаг 6	12
3.7	Шаг 7	13
3.8	Шаг 8	14
4	Выводы	16

Список иллюстраций

3.1	Простейшие операции	7
3.2	Операции с векторами	8
3.3	Вычисление проектора	9
3.4	Матричные операции	10
3.5	Определитель, обратная матрица, собственные значения и ранг матрицы	10
3.6	График функции	11
3.7	Улучшенный график функции	12
3.8	График точек	12
3.9	Два графика на одном чертеже	13
3.10	График $y = x^2 \sin x$	13
3.11	Файл loop_for.m	14
3.12	Файл loop_vec.m	14
3.13	Запуск файлов loop_vec.m и loop_for.m	15

Список таблиц

1 Цель работы

Научиться работать с Octave.

2 Задание

Ознакомиться с простейшими операциями, операциями с векторами и матрицами, построить графики функций и сравнить эффективность работы с циклами и операций с векторами.

3 Выполнение лабораторной работы

3.1 Шаг 1

Включила журналирование сессии с помощью *diary*, вычислила значение выражения, задала вектор-строку и вектор-столбец, а также матрицу, используя показанные на Рисунке 1 (рис - fig. 3.1) строки.

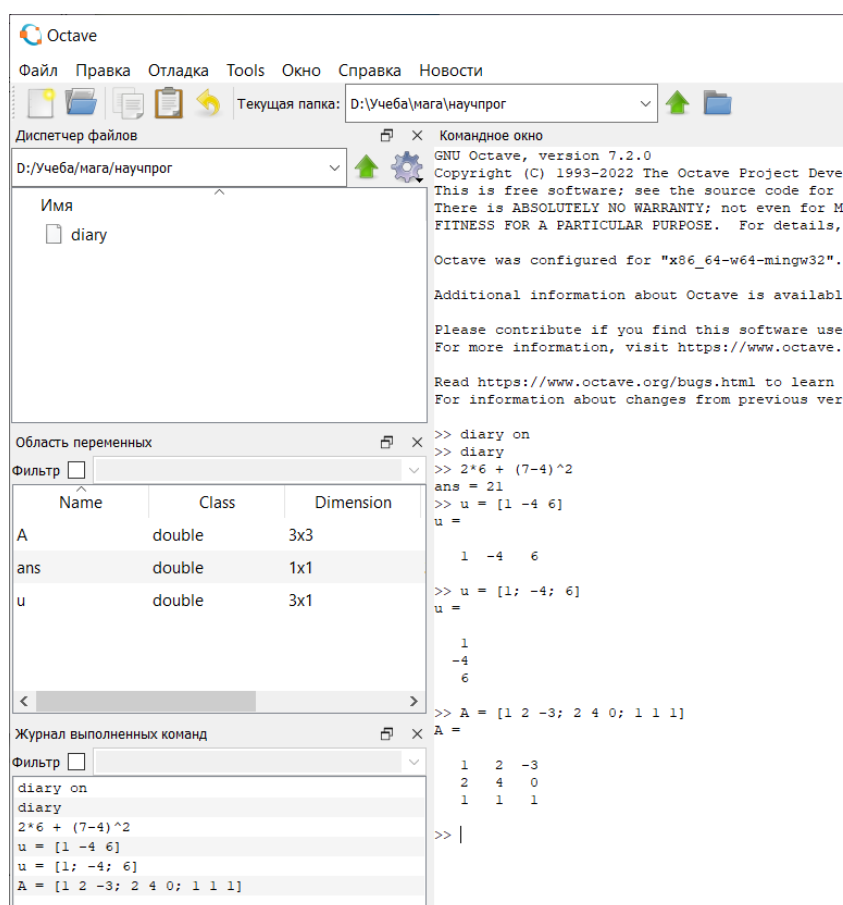


Рис. 3.1: Простейшие операции

3.2 Шаг 2

Задала два вектор-столбца \vec{u} и \vec{v} . Выполнила их сложение, скалярное и векторное умножения и вычислила норму вектора, используя строки, показанные на Рисунке 2 (рис - fig. 3.2).

```
>> u = [1; -4; 6]
u =
     1
    -4
     6

>> v = [2; 1; -1]
v =
     2
     1
    -1

>> 2*v + 3*u
ans =
     7
    -10
    16

>> dot(u, v)
ans = -8
>> cross(u, v)
ans =
    -2
    13
     9

>> norm(u)
ans = 7.2801
```

Рис. 3.2: Операции с векторами

3.3 Шаг 3

Задала два вектор-строки \vec{u} и \vec{v} и вычислила проекцию вектора \vec{u} на вектор \vec{v} с помощью строки $proj = dot(u, v)/(norm(v))^2 \cdot v^*$, что показано на Рисунке 3 (рис - fig. 3.3).


```

>> u = [3 5]
u =

    3    5

>> v = [7 2]
v =

    7    2

>> proj = dot(u, v) / (norm(v))^2 * v
proj =

    4.0943    1.1698

>> |

```

Рис. 3.3: Вычисление проектора

3.4 Шаг 4

Задала две матрицы A и B , вычислила их произведение, произведение транспонированной матрицы B и A , разницу матриц, где $e_{ue}(3)$ - единичная матрица. Выполнение операций показано на Рисунке 4 (рис - fig. 3.4). Кроме того, нашла определитель, обратную матрицу, собственные значения матрицы и ее ранг, используя показанные на Рисунке 5 (рис - fig. 3.5) строки.

```

>> A = [1 2 -3; 2 4 0; 1 1 1]
A =

     1     2    -3
     2     4     0
     1     1     1

>> B = [1 2 3 4; 0 -2 -4 6; 1 -1 0 0]
B =

     1     2     3     4
     0    -2    -4     6
     1    -1     0     0

>> A*B
ans =

    -2     1    -5    16
     2    -4   -10    32
     2    -1    -1    10

>> B' * A
ans =

     2     3    -2
    -3    -5    -7
    -5   -10    -9
    16    32   -12

>> 2 * A - 4 * eye(3)
ans =

    -2     4    -6
     4     4     0
     2     2    -2

>> eye(3)
ans =

Diagonal Matrix

     1     0     0
     0     1     0
     0     0     1

```

Рис. 3.4: Матричные операции

```

>> det(A)
ans = 6

>> inv(A)
ans =

     0.6667    -0.8333     2.0000
    -0.3333     0.6667    -1.0000
    -0.3333     0.1667         0

>> eig(A)
ans =

     4.5251 + 0i
     0.7374 + 0.8844i
     0.7374 - 0.8844i

>> rank(A)
ans = 3

```

Рис. 3.5: Определитель, обратная матрица, собственные значения и ранг матрицы

3.5 Шаг 5

Создала вектор значений x с помощью команды `linspace(начальное_значение, конечное_значение, n)`, которая создаёт вектор-строку из n равномерно распределённых значений на заданном интервале. Чем меньше приращение, тем более гладкой будет выглядеть кривая. Точка с запятой в конце строки предназначена для подавления вывода на экран. Задала также вектор y и построила график, показанный на Рисунке 6 (рис - fig. 3.6).

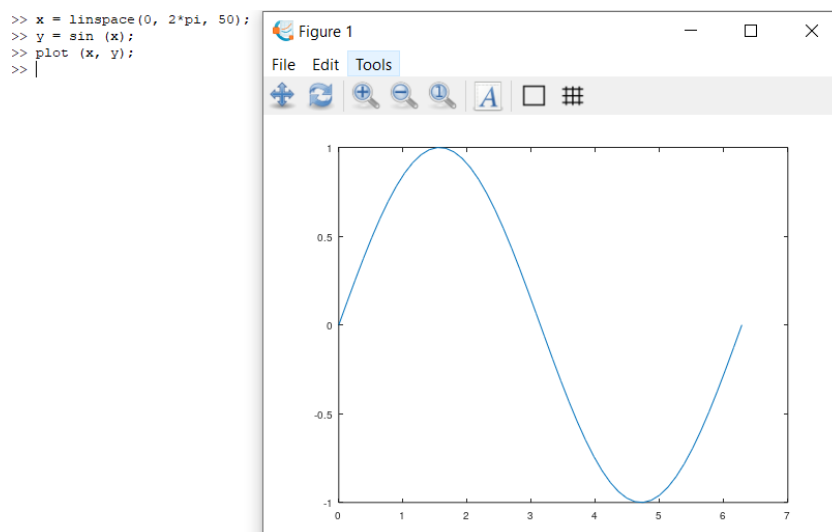


Рис. 3.6: График функции

Для улучшения внешнего вида графика очистила получившийся до этого график, задала красный цвет для линии и ее толщину, подогнала диапазон осей, нарисовала сетку, подписала оси, добавила заголовок графика и задала легенду. Получившийся график показан на Рисунке 7 (рис - fig. 3.7).

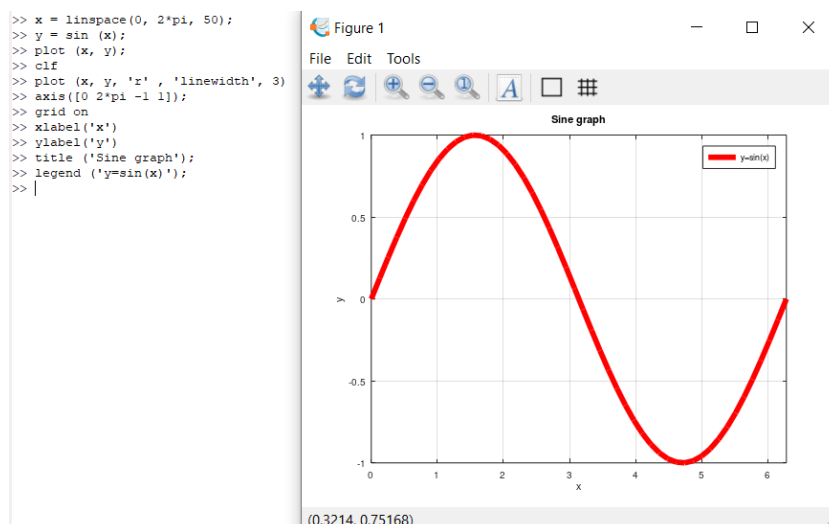


Рис. 3.7: Улучшенный график функции

3.6 Шаг 6

Очистила память и рабочую область фигуры. Задала два вектора и начертила точки, используя в качестве маркеров кружочки. График представлен на Рисунке 8 (рис - fig. 3.8).

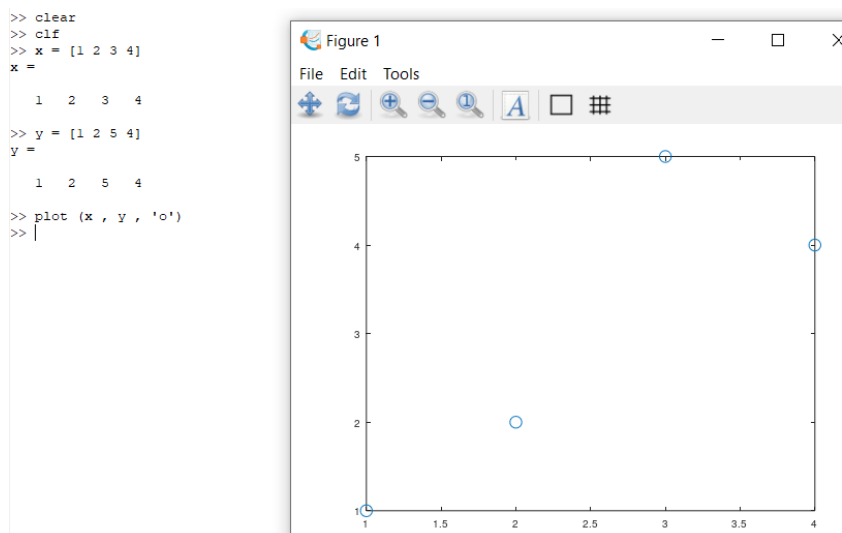


Рис. 3.8: График точек

Для добавления еще одного графика, а именно графика регрессии, использова-

ла команду *hold on*. Задала сетку, оси и легенду. Получившийся график представлен на Рисунке 9 (рис - fig. 3.9).

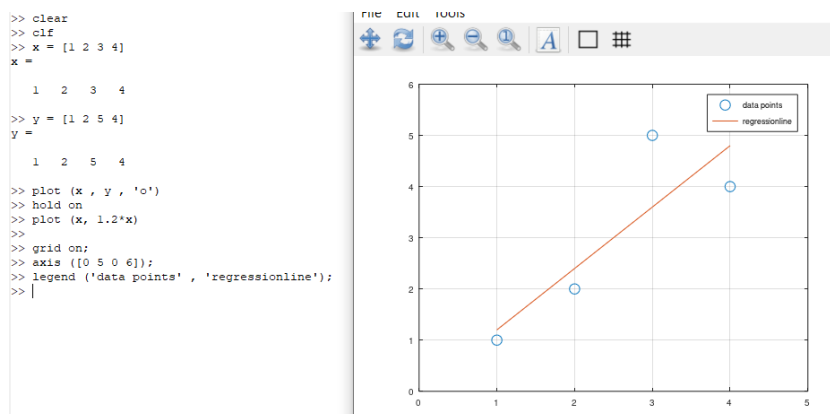


Рис. 3.9: Два графика на одном чертеже

3.7 Шаг 7

Для построения графика $y = x^2 \sin x$ очистила память и рабочую область фигуры. Задала вектор \vec{x} и использовала поэлементное возведение в степень и поэлементное умножение для построения графика. Получившийся график, представленный на Рисунке 10 (рис - fig. 3.10), сохранила в двух форматах.

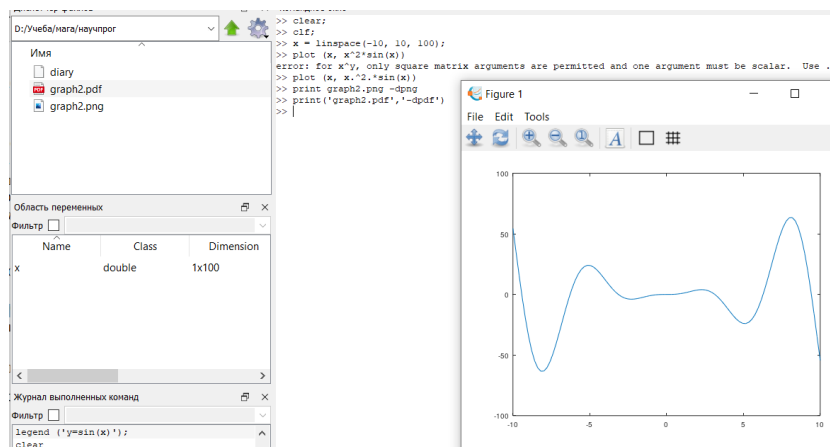


Рис. 3.10: График $y = x^2 \sin x$

3.8 Шаг 8

Для сравнения эффективности работы с циклами и операций с векторами очистила память и рабочую область фигуры. Создала файл *loop_for.m*, содержимое которого представлено на Рисунке 11 (рис - fig. 3.11). Код в данном файле вычисляет сумму $\sum_{n=1}^{1000000} \frac{1}{n^2}$ с помощью цикла. Код, содержащийся в файле *loop_vec.m*, представленный на Рисунке 12 (рис - fig. 3.12), вычисляет эту сумму с помощью операций с векторами. Результаты выполнения данных файлов представлены на Рисунке 13 (рис - fig. 3.13). Вычисление суммы с помощью операций с векторами оказалось в 17,5 раз быстрее, чем вычисление с помощью цикла.



Рис. 3.11: Файл loop_for.m



Рис. 3.12: Файл loop_vec.m

```
>> clear;  
>> clf;  
>> loop_for  
Elapsed time is 0.163952 seconds.  
>> loop_vec  
Elapsed time is 0.00938487 seconds.  
>> diary off  
>> |
```

Рис. 3.13: Запуск файлов loop_vec.m и loop_for.m

4 Выводы

Я ознакомилась с Octave, а именно с работой с простейшими операциями, операциями с векторами и матрицами, построить графики функций и сравнить эффективность работы с циклами и операций с векторами. Результаты работы находятся в репозитории на GitHub, а также есть скринкаст выполнения лабораторной работы.