Лабораторная работа №3

Введение в Octave

Сырцов Александр Юрьевич

Содержание

# Цель работы

Познакомиться с языком Octave, изучить основные инструменты работы с векторами и матрцами.

# Задание

Подготовить отчёт.

# Выполнение лабораторной работы

1. Включаю листинг команд и записываю всё в файл list.txt. Сам файл и его содержимое (рис. -fig. 1)

diary list.txt

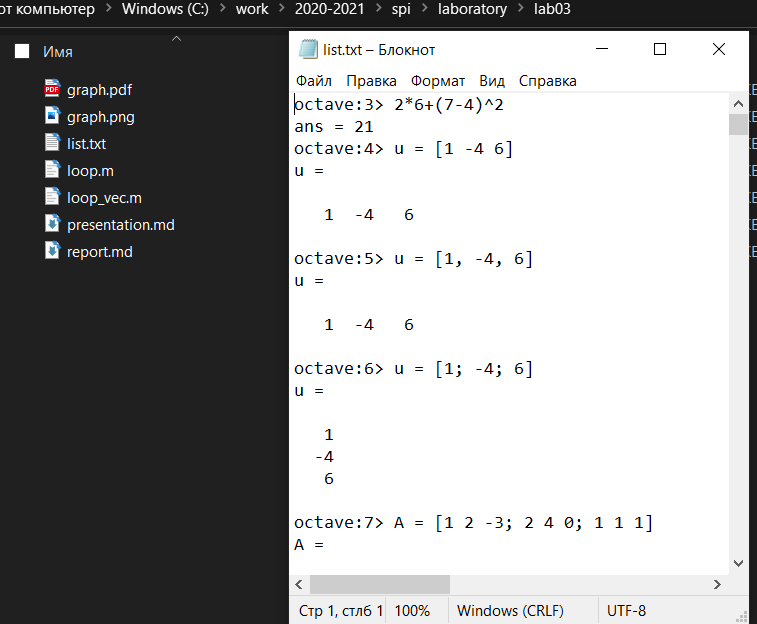


Figure 1: 1

1. Использую Octave для вычисления математического выражения (рис. -fig. 2)

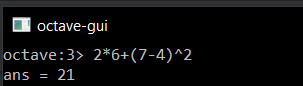


Figure 2: 2

1. Задаю конвектор u, или вектор-строку, двумя способами (используя стандартный синтаксис и синтаксис с запятыми). Также создаю вектор с теми же значениями, переопределяя u, и матрицу A (рис. -fig. 3)

уже сейчас можно сказать, что Octave обладает ленивой динамической типизацией.

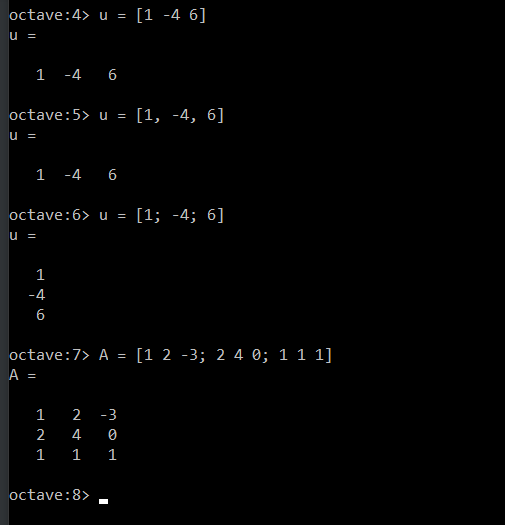


Figure 3: 3

1. Задав два вектора u и v, нашёл значение выражения 2v + 3u, затем скалярное (в ответе число), векторное произведение (в ответе вектор) двух векторов и норму вектора u (рис. -fig. 4)

u = [1; -4; 6]  
 v = [2; 1; -1]

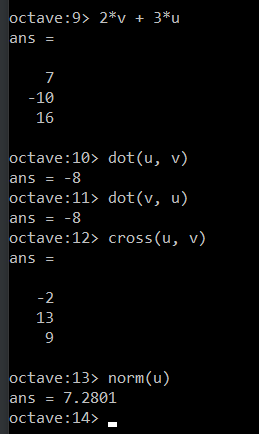


Figure 4: 4

1. Вычисляю проекцию вектора u на v, перед этим переопределив их (рис. -fig. 5)

Стандартный инструмент для решения не задан, поэтому пользуемся выведением формулы проекции через формулу скалярного произведения.

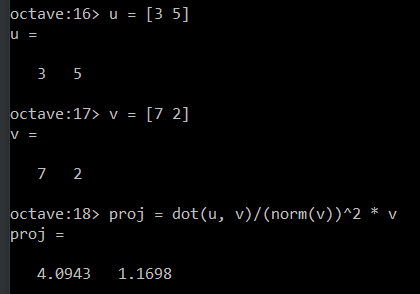


Figure 5: 5

1. Задаём матрицы A и B (рис. -fig. 6)

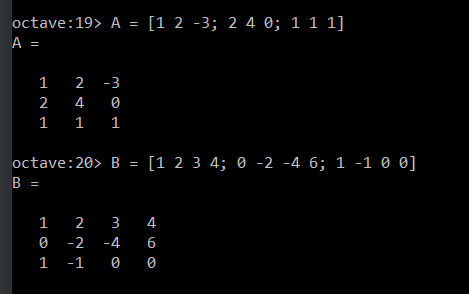


Figure 6: 6

1. Вычислил их произведение, произведение с транспонированной матрицей B и выражение вида 2A - 4I, I - единичная матрица (рис. -fig. 7)

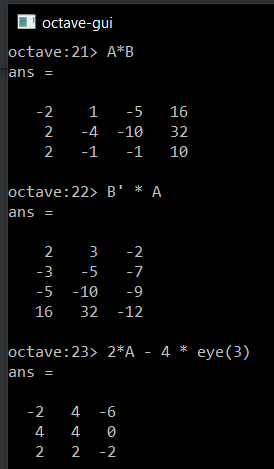


Figure 7: 7

Пример единичной матрицы 5x5 (рис. -fig. 8)

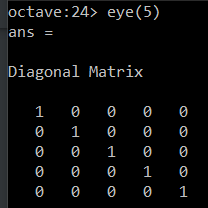


Figure 8: 8

1. высчитаем ранг матрицы A (рис. -fig. 8)

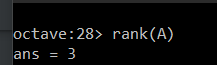


Figure 9: 9

1. Построим график, задав все значения х через range-функцию linspace() и определим нашу переменную y со всеми значениями во всех точках x. Чтобы получить график, используем функцию plot(), которая выводит неприрывную линию по умолчанию (рис. -fig. 10)

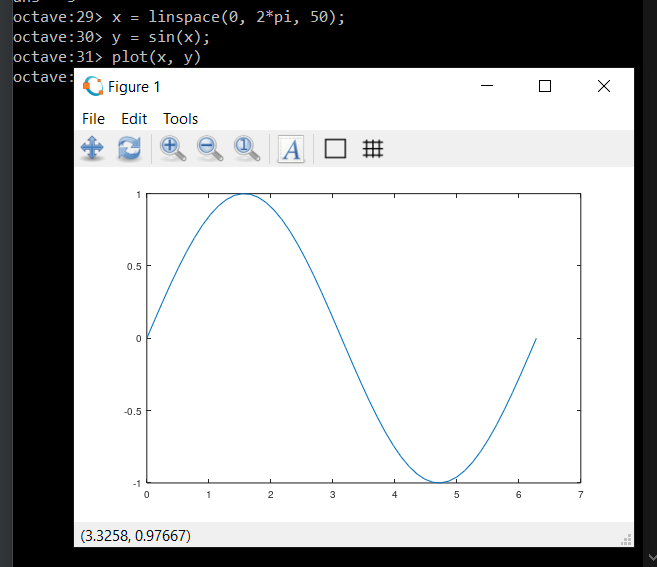


Figure 10: 10

1. Улучшим внешний вид графика, определив параметр цвета 'r' (можно было задать его через параметр 'r-', явно задав отображение в виде линии) и параметр ширины линии. Кроме того функцией axis() подраниваем значения осей, то есть меняем масштаб изображения, включаем сетку, добавляем подписи к осям и легенду (рис. -fig. 11)

Функция plot() принимает множественные параметры, поэтому, считывая параметр 'linewidth', считывает следующий за ним параметр, как значение ширины линии. По сути это своеобразная замена инструмента ключевых параметров.

% предварительно отчистим фигуру  
 clf

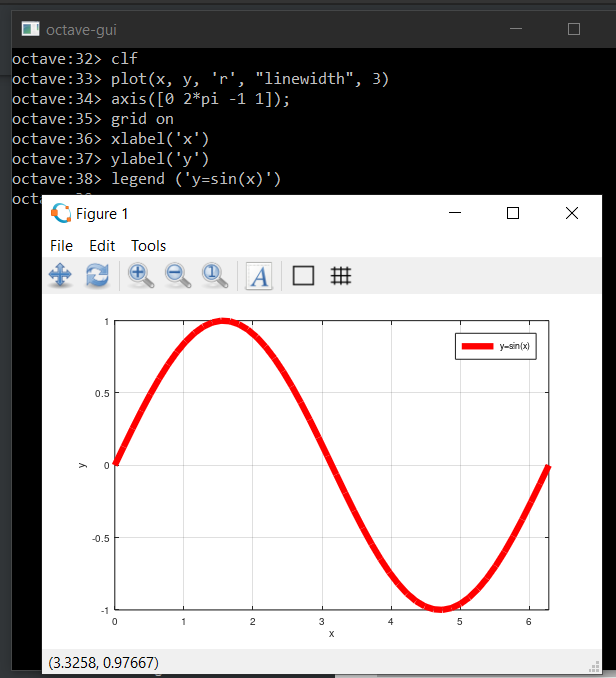


Figure 11: 11

1. Теперь изображаем две диаграммы: точки и прямую линейной регрессии, задав нужные значения (рис. -fig. 12)

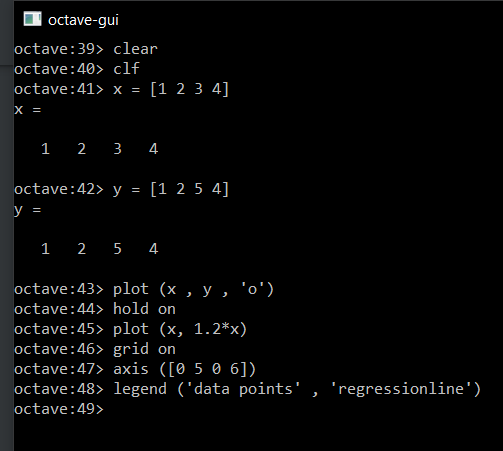


Figure 12: 12

% аналогично можно было  
 % отобразить две диаграммы на одной фигуре так  
  
 plot(x, y, 'o', x, 1.2 \* x, '-')

Итоговое изображение выглядит так (рис. -fig. 13)

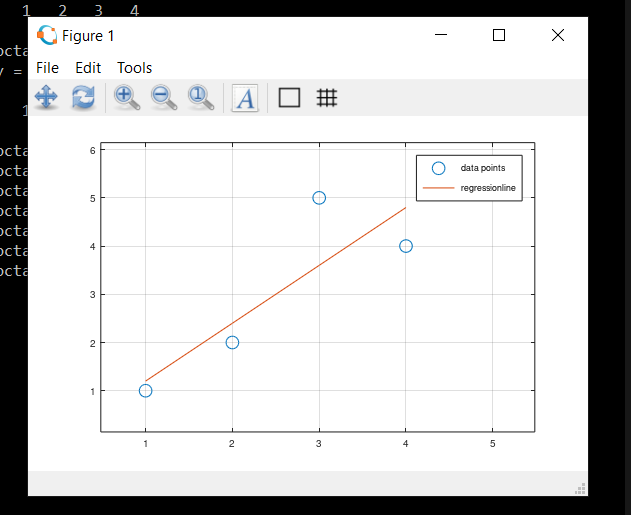


Figure 13: 13

1. Изобразим третий график, который потом сохраним двумя способами. для начала зададим x (рис. -fig. 14), далее отобразим график и сохраним в формате png (рис. -fig. 14)

Figure 14: 14

Figure 14: 14

Figure 15: 15

Figure 15: 15

% второй способ отличается тем,  
 % что вместо макроса,  
 % мы используем print в качестве функции  
  
 % конкретно тут сохраняем в формате pdf  
 print('graph2.pdf','-dpdf')

Сам график выглядит так (рис. -fig. 16)

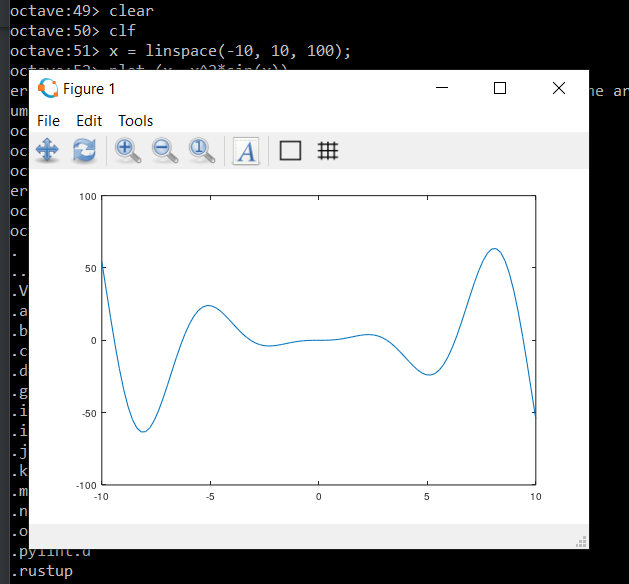


Figure 16: 16

То, что всё было сохранено, можно увидеть выше (рис. -fig. 1)

1. Сравним работу двух алгоритмов вычисления суммы элементов последовательности от 1 до 1000000: через range-цикл for с итератором и векторную форму (то есть через векторы и орифметические операторы с точкой)

Код с циклом for (рис. -fig. 17)

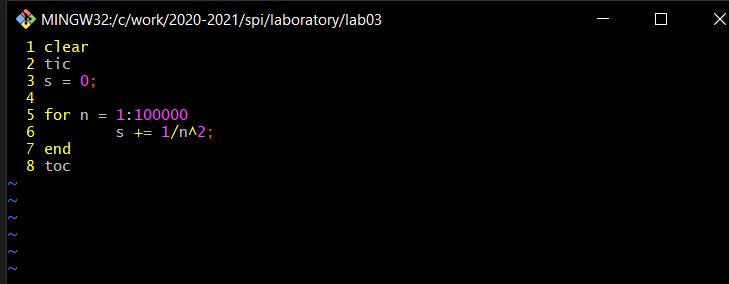


Figure 17: 17

Код с векторами (рис. -fig. 17)

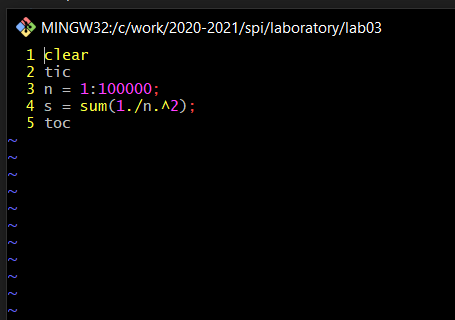


Figure 18: 18

Результаты оказались неожиданными – векторная форма отработала намного быстрее (рис. -fig. 19)

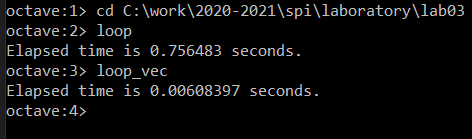


Figure 19: 19

Скорее всего, дело в реализации инструментов языка. По аналогии с Python, где цикл for реализован через язык C, поэтому работает быстрее, чем более нативный цикл while, можно предположить, что в Octave присутствует реализация на языках более низкого уровня абстракции.

# Выводы

Мне удалось освоить начальные навыки владения языка Octave, с помощью его инструментов произвести операции с векторами, конвекторами, матрицами: создать и сохранить в разных форматах диаграммы и понять некоторые принципы его работы.