Лабораторная работа №3

Научное программирование

Викторов Егор

Содержание

# 1 Цель работы

Освоить базовые навыки работы в Octave: простейшие вычислительные операции, операции с веторами и матрицами, построение простейших графиков, сравнение циклов и операций с векторами.

# 2 Теоретическое введение

**Octave** — высокоуровневый интерпретируемый язык программирования, предназначенный для решения задач вычислительной математики. По возможностям и качеству реализации интерпретатора язык Octave можно сравнивать с проприетарной программой MATLAB, причём синтаксис обоих языков очень схож. В состав пакета входит интерактивный командный интерфейс (интерпретатор Octave). Интерпретатор Octave запускается из терминала ОС Linux или из его порта в Windows. После запуска Octave пользователь видит окно интерпретатора.

В окне интерпретатора пользователь может вводить как отдельные команды языка Octave, так и группы команд, объединяемые в программы. Если строка заканчивается символом “;”, результаты на экран не выводятся. Если же в конце строки символ “;” отсутствует, результаты работы выводятся на экран. Текст в строке после символа % является комментарием и интерпретатором не обрабатывается.

Octave имеет большое количество инструментов для решения распространенных задач числовой линейной алгебры, поиска корней нелинейных уравнений, интегрирования обычных функций, управления полиномами, интегрирования обыкновенных дифференциальных и дифференциально-алгебраических уравнений и т.д. Помимо этого в Octave можно рисовать графики. Функционал этой программной системы может быть легко расширен и перенастроен с помощью пользовательских функций, написанных на собственном языке программирования Octave, либо с помощью динамически загружаемых модулей, написанных на C++, C, Fortran или других языках.

В Octave реализованы многие возможности Matlab, включая использование матриц в качестве основных типов данных, поддержку комплексных чисел, поддержку математических функции и больших библиотек функций, а также возможность создания пользовательских функций для расширения функциональности системы.

Более подробно см. в [@Octave\_1:bash] и [@Octave\_2:bash].

# 3 Выполнение лабораторной работы

Для начала включили журналирование сессии, создав документ “diary”. Вычислили значение выражения. Задали вектор-строку (ковектор), вектор-столбец (вектор) и матрицу (рис. fig. 1).

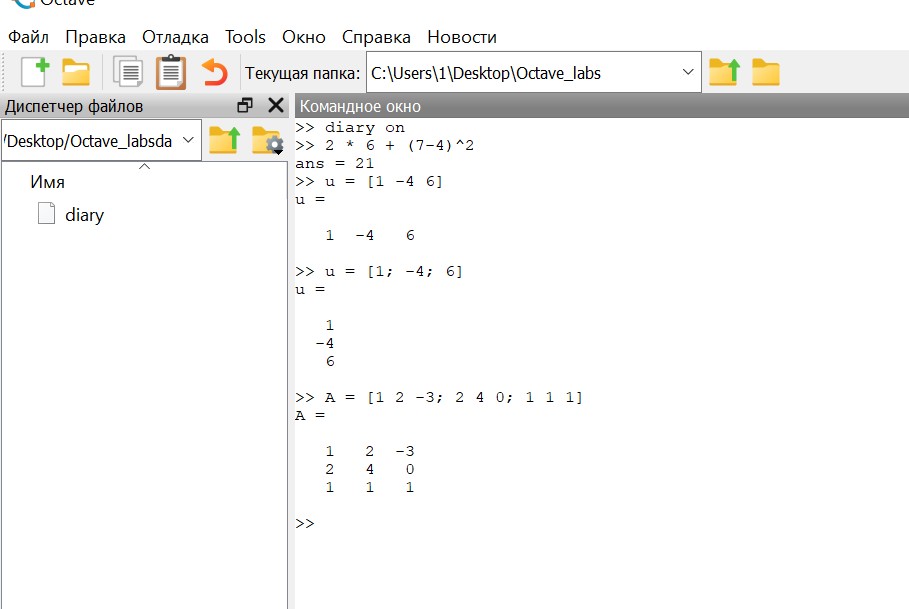


Рис. 1: Простейшие операции

Задали два вектора-столбца. Осуществили сложение заданных векторов, их скалярное и векторное умножение. Нашли норму одного из векторов (рис. fig. 2).

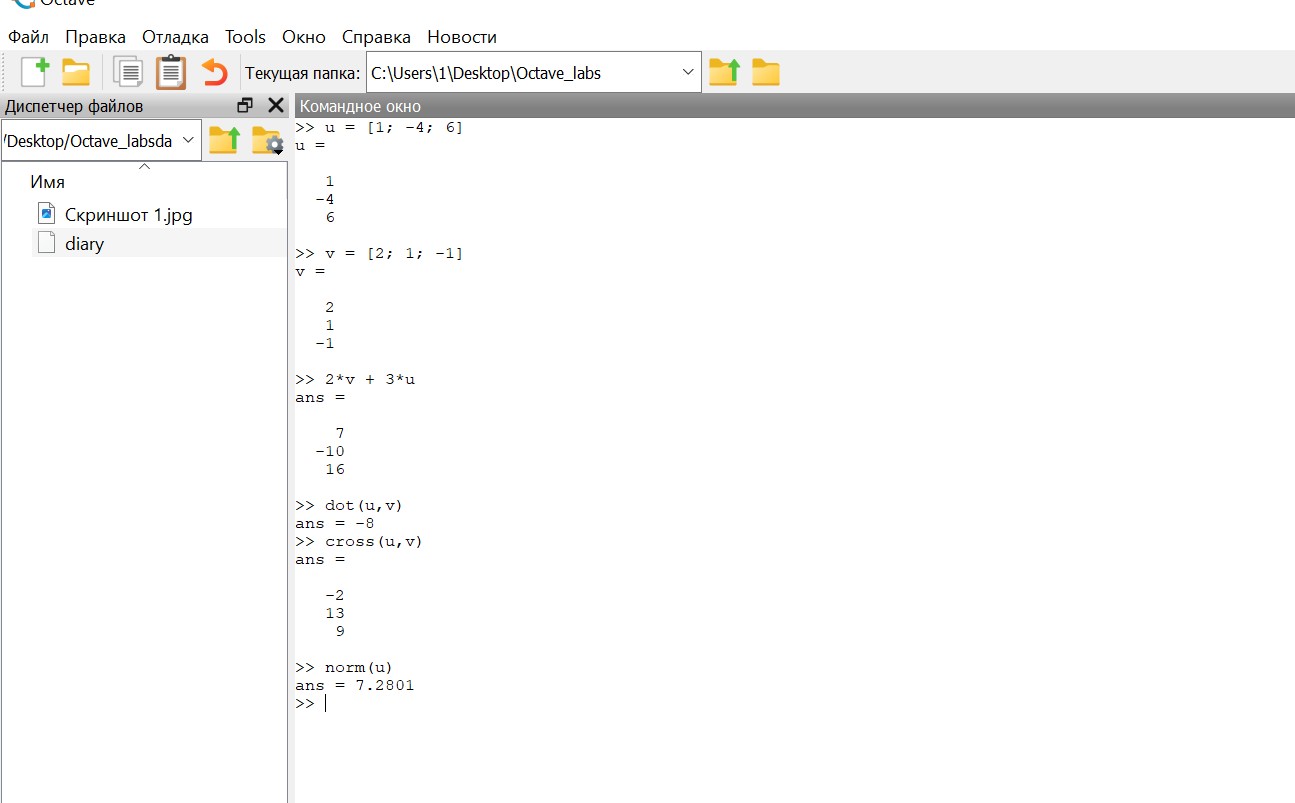


Рис. 2: Операции с векторами

Ввели два вектора-строки. Вычислили проекцию одного вектора на другой (рис. fig. 3).

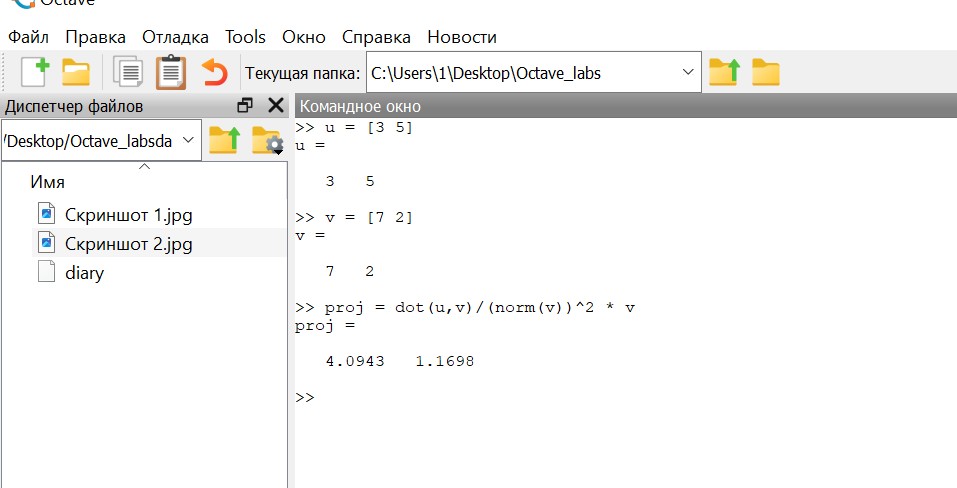


Рис. 3: Вычисление проекции одного вектора на другой

Ввели две матрицы. Вычислили их произведение, а также их произведение с учетом транспонирования одной из матриц. Вычислили выражение, содержащее единичную матрицу. Нашли определитель матрицы, обратную для нее, вычислили собственные значения матрицы и ее ранг (рис. fig. 4 и рис. fig. 5).

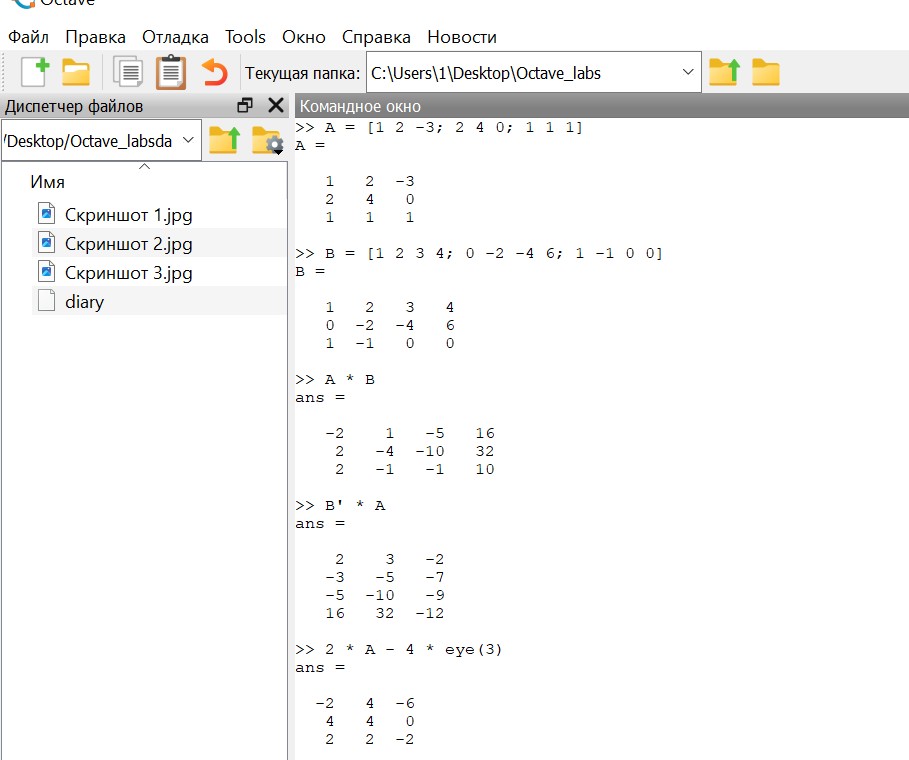


Рис. 4: Матричные операции

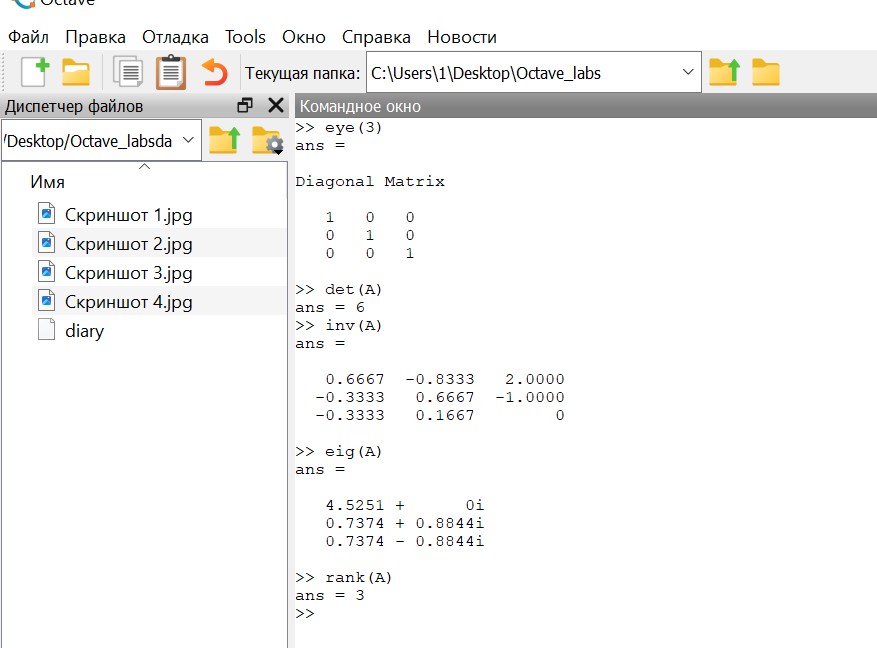


Рис. 5: Матричные операции

Создали вектор значений . Задали вектор и построили график. Очистили рабочую область фигуры для дальнейшего улучшения графика. Задали красный цвет для линии и сделали ее потолще. Подогнали диапазон осей и нарисовали сетку. Подписали оси и сделали заголовок графика, а также задали легенду (рис. fig. 6 и рис. fig. 7).

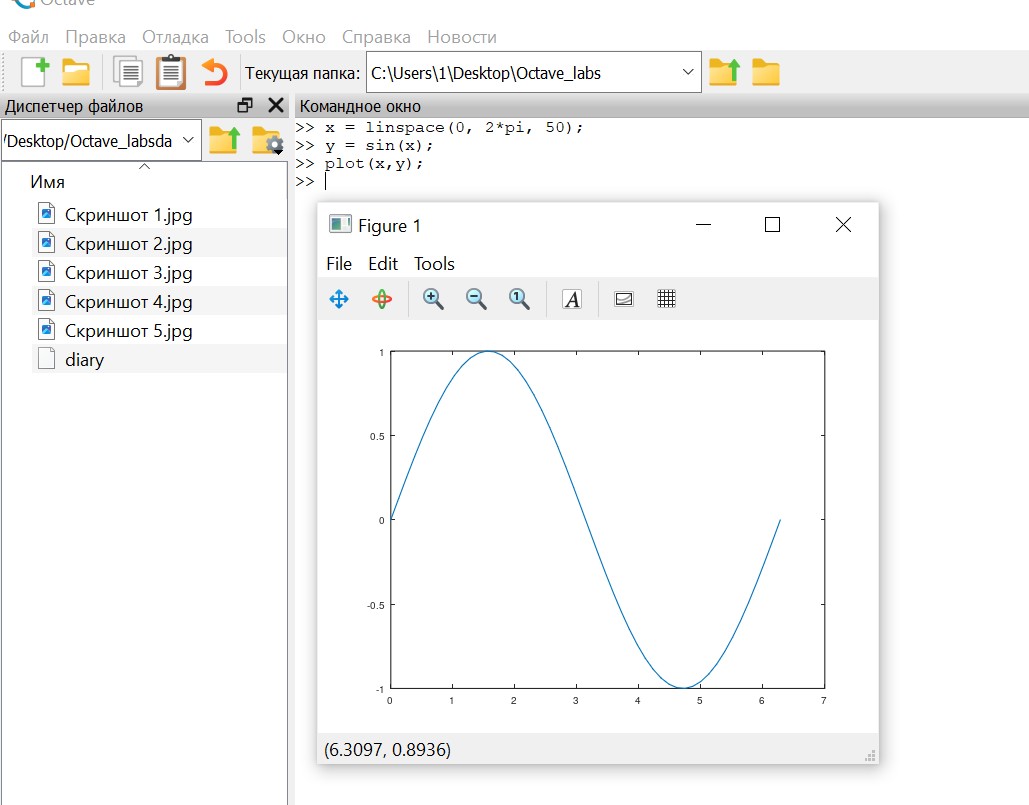


Рис. 6: Построение простейших графиков

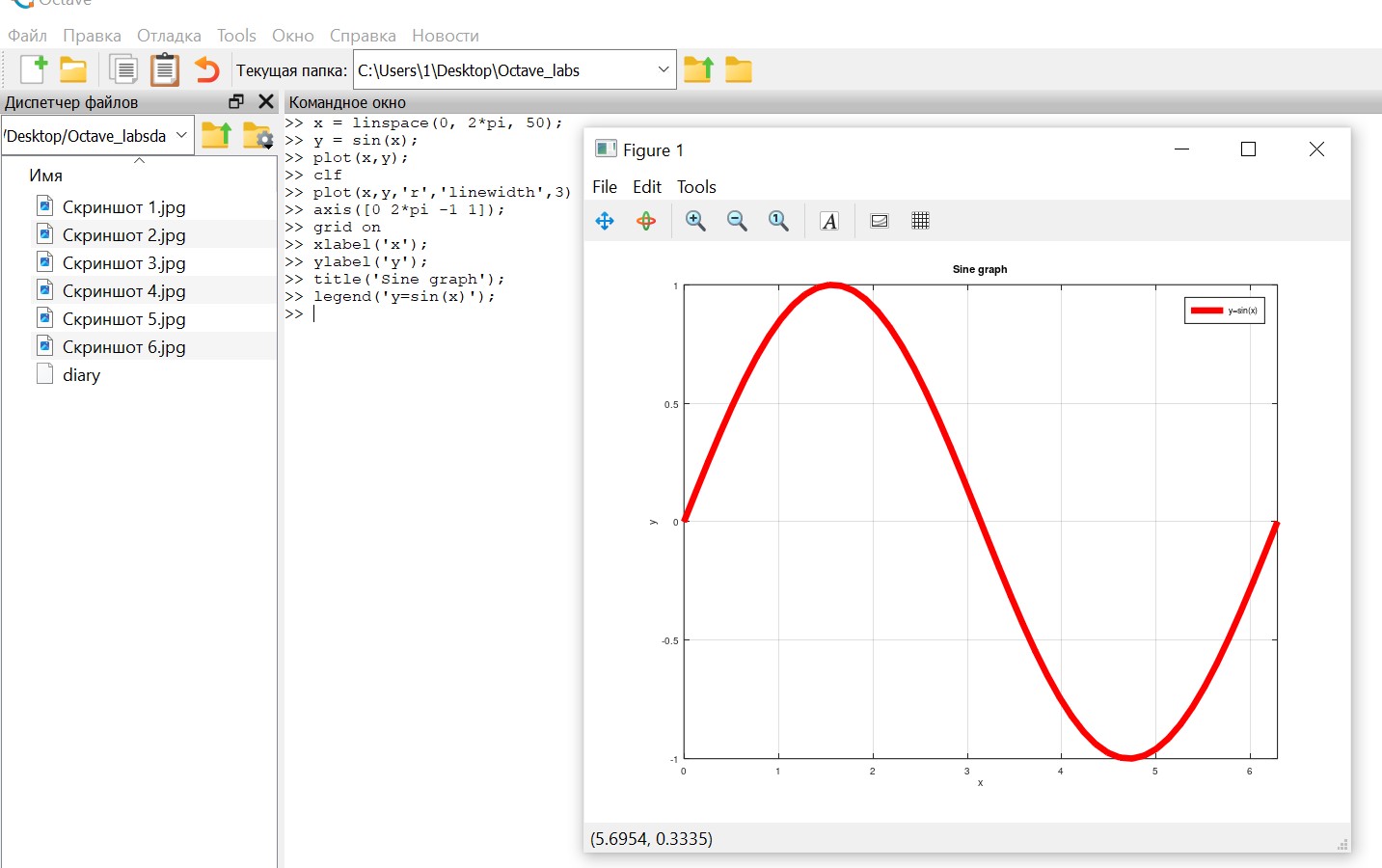


Рис. 7: Построение простейших графиков

Очистили память и рабочую область фигуры. Задали два вектора и начертили эти точки, используя кружочки как маркеры. Ввели команду для добавления еще одного графика к текущему. Добавили график регрессии. Задали сетку, оси и легенду (рис. fig. 8).

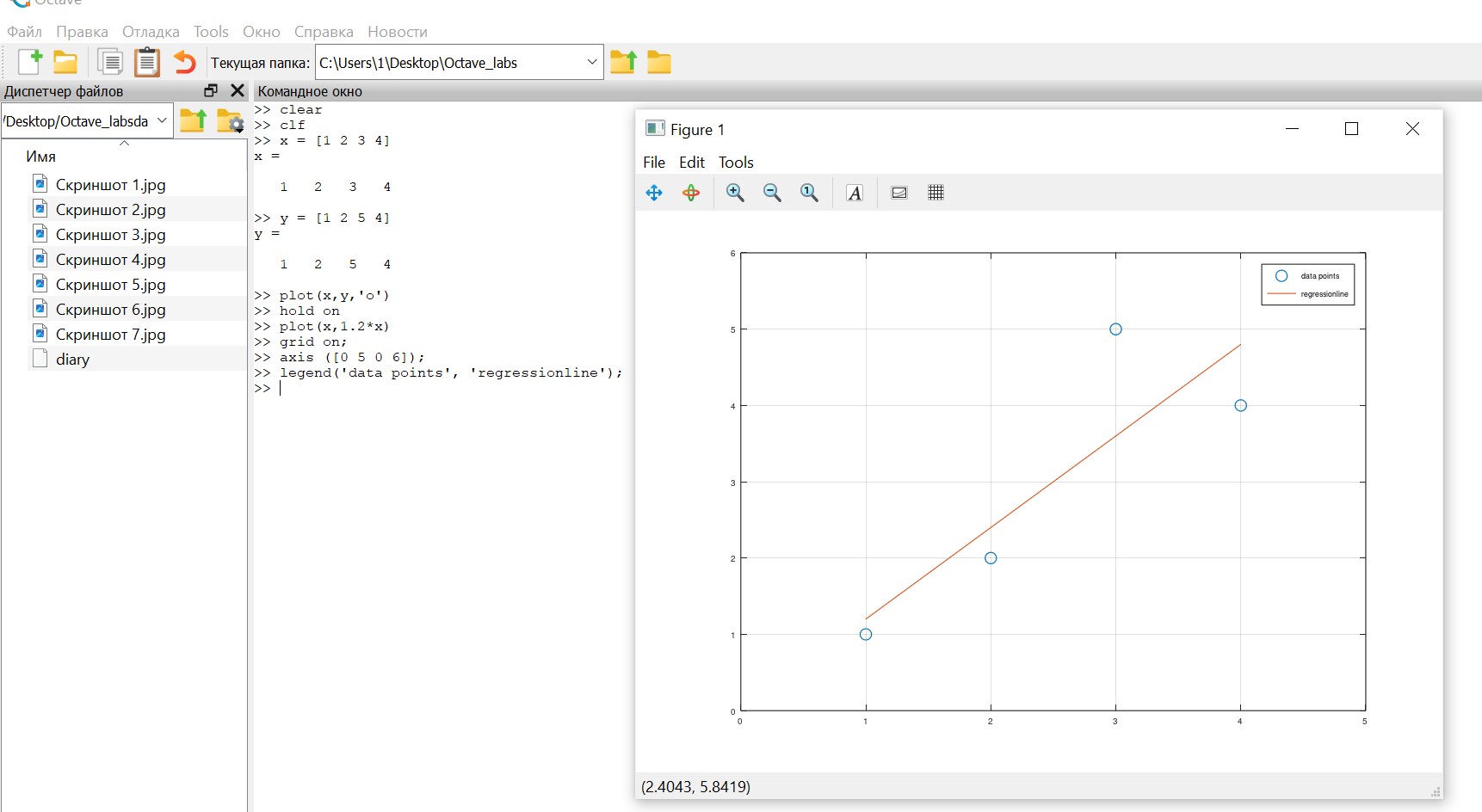


Рис. 8: Два графика на одном чертеже

Строим график . Очистим память и рабочую область фигуры. Зададим вектор и построим график, используя поэлементное возведение в степень и поэлементное умножение (в противном случае выдаст ошибку). Сохраним графики в виде файлов (на сохранении файла в формате pdf программа зависает) (рис. fig. 9).

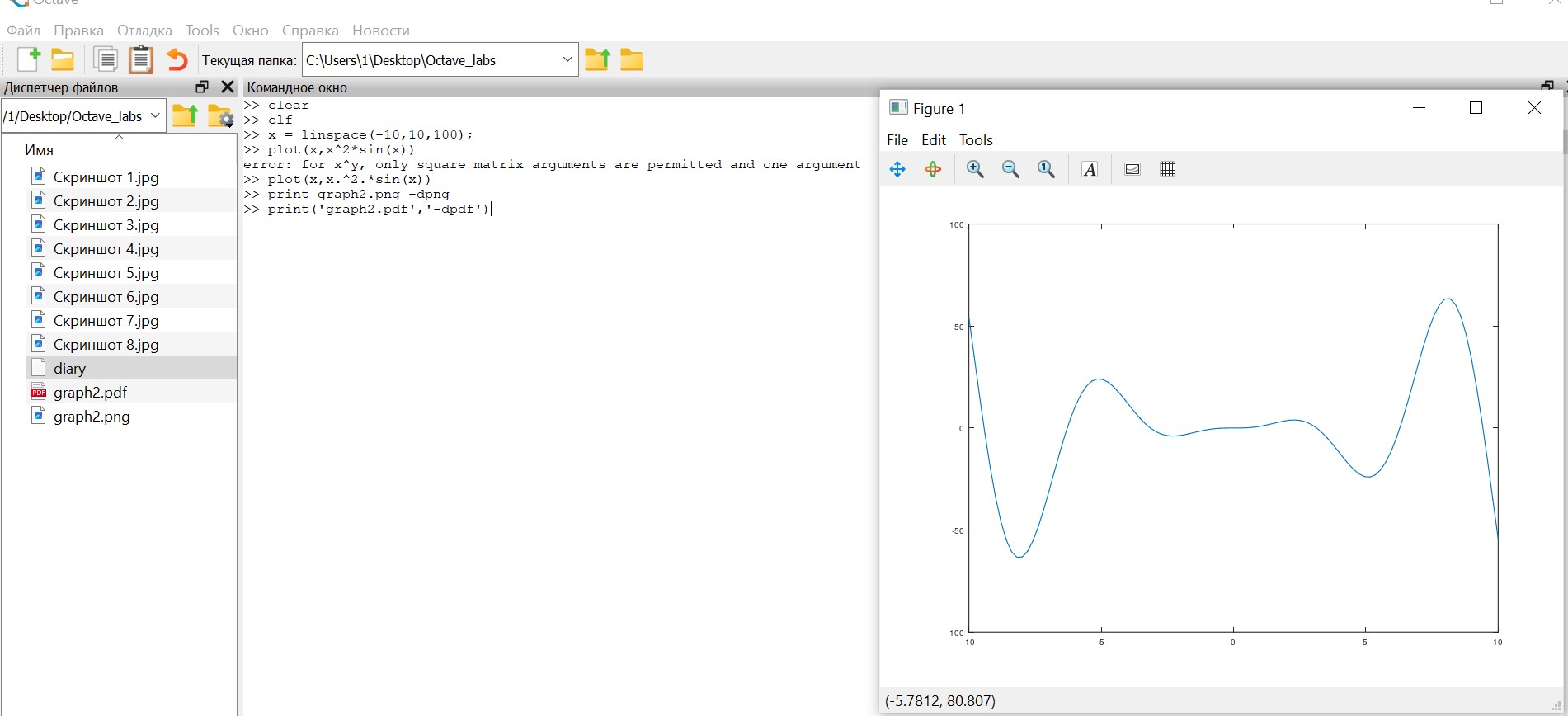


Рис. 9: График

Сравним эффективность работы с циклами и операций с векторами. Для этого вычислим сумму: .

Очистим память и рабочую область фигуры. Вычислим сумму с помощью цикла, создав файл loop\_for.m (рис. fig. 10).

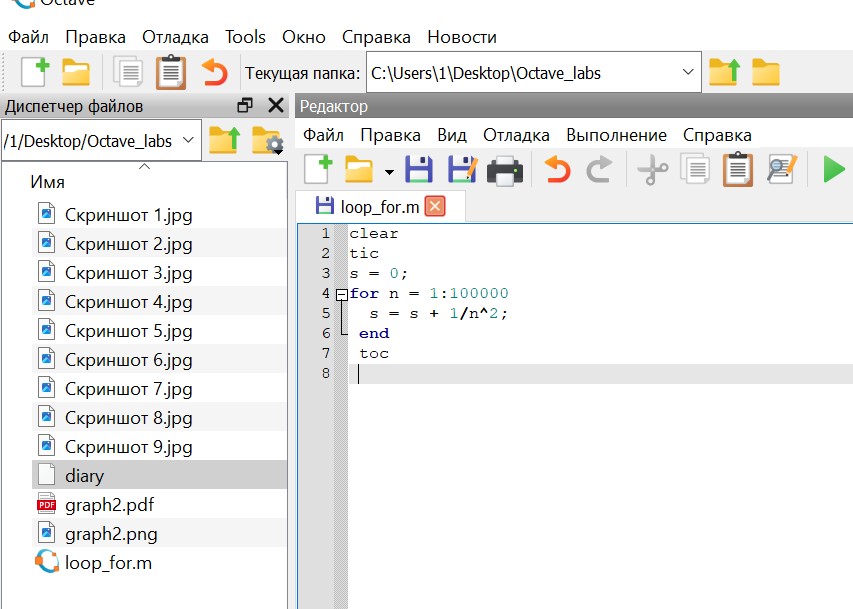


Рис. 10: Вычисление суммы с помощью цикла

Вычислим сумму с помощью операций с векторами, создав файл loop\_vec.m (рис. fig. 11).

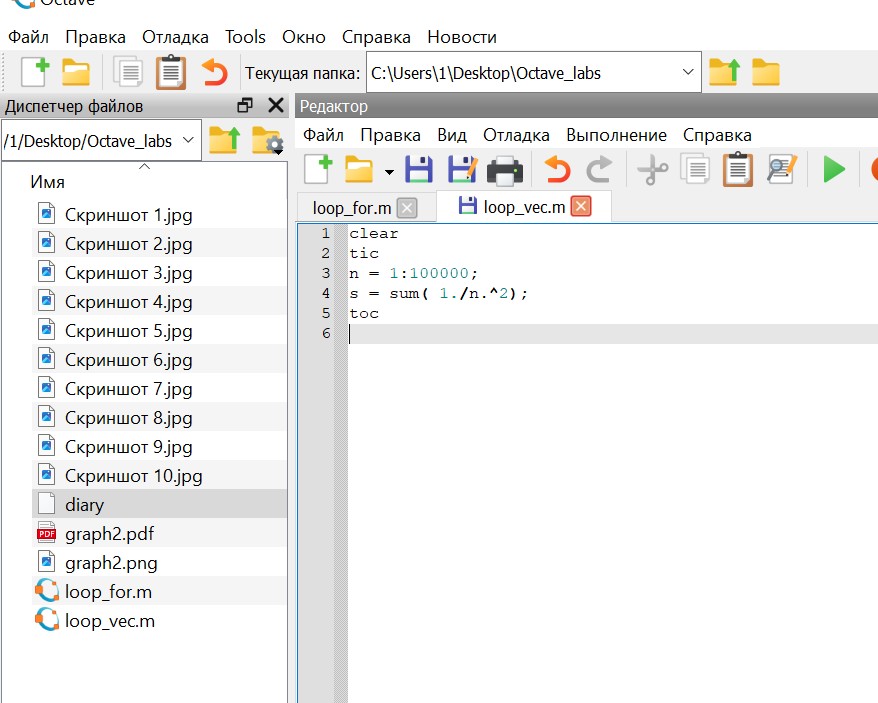


Рис. 11: Вычисление суммы с помощью операций с векторами

Запустив оба файла, видим, что время на выполнение второй программы в 100 раз меньше, чем на выполнение первой (рис. fig. 12). Таким образом, в данном случае операции с векторами эффективнее, чем циклы. В конце завершим запись в файл.

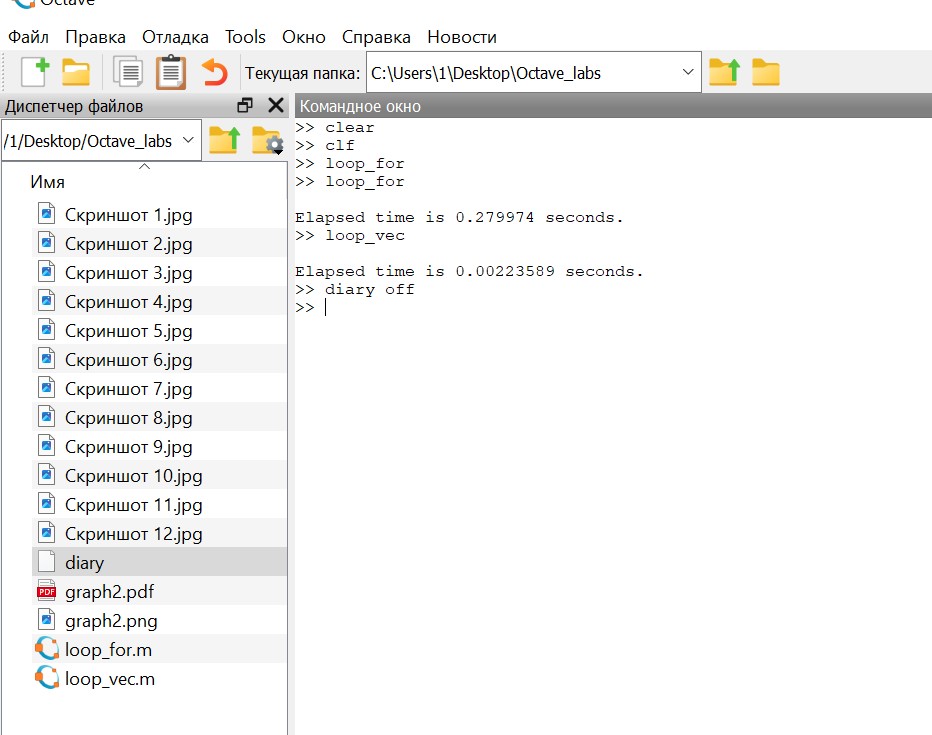


Рис. 12: Сравнение двух программ

# 4 Вывод

В ходе выполнения данной лабораторной работы я освоил базовые навыки работы в Octave: простейшие вычислительные операции, операции с веторами и матрицами, построение простейших графиков, сравнение циклов и операций с векторами.

# Список литературы