Лабораторная работа №4

Линейная алгебра

Клюкин Михаил Александрович

Содержание

# 1 Цель работы

Основной целью работы является изучение возможностей специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.

# 2 Задание

1. Используя Jupyter Lab, повторитm примеры из раздела 4.2.
2. Выполнитm задания для самостоятельной работы (раздел 4.4).

# 3 Выполнение лабораторной работы

## 3.1 Поэлементные операции над многомерными массивами

Для матрицы 4 x 3 рассмотрим поэлементные операции сложения и произведения её элементов (рис. 1, 2):

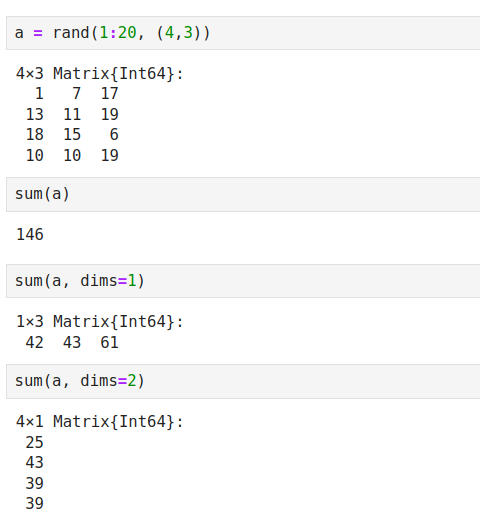


Рис. 1: Пример поэлементного суммирования по столбцам и строкам

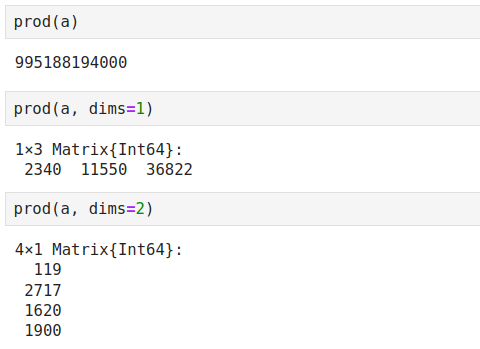


Рис. 2: Пример поэлементного произведения по столбцам и строкам

Для работы со средними значениями можно воспользоваться возможностями пакета Statistics (рис. 3):



Рис. 3: Импорт пакета Statistics

Используя этот пакет, найдем среднее значение всей матрицы, средние значения по столбцам и по строкам (рис. 4).

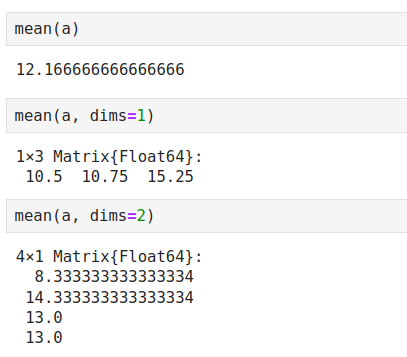


Рис. 4: Пример нахождения средних во всей матрице, по столбцам и по строкам

Для выполнения таких операций над матрицами, как транспонирование, диагонализация, определение следа, ранга определителя матрицы и т.п. можно воспользоваться библиотекой (пакетом) LinearAlgebra (рис. 5):

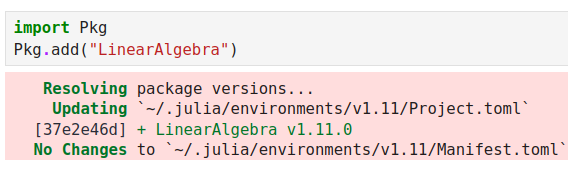


Рис. 5: Импорт пакета LinearAlgebra

Создадим матрицу 4 х 4 со случайными целыми числами из интервала от 1 до 20, транспонируем ее, найдем след, создадим массив из диагональных элементов, найдем ранг, обратную матрицу, детерминант и псевдообратную матрицу для матрицы из предыдущего примера (рис. 6, 7).

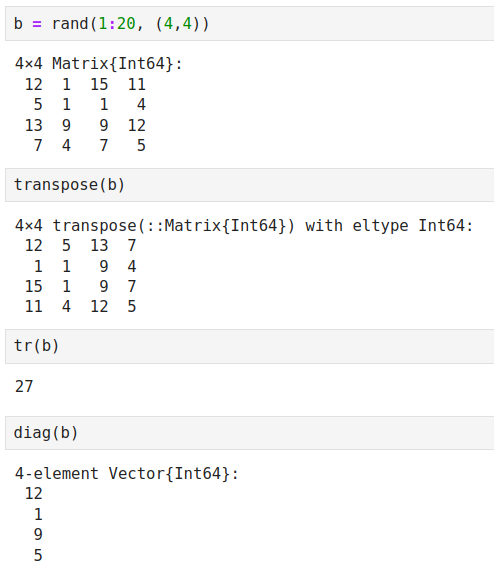


Рис. 6: Транспонирование, след и диагональные элементы

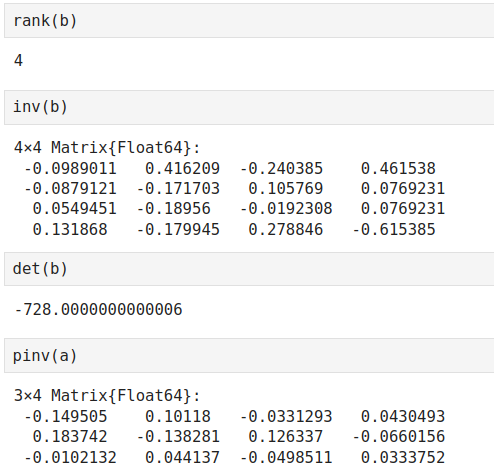


Рис. 7: Ранг, обратная матрица, детерминант и псевдообратная матрица

## 3.2 Вычисление нормы векторов и матриц, повороты, вращения

Для вычисления нормы используется LinearAlgebra.norm(x).

Создадим вектор X, вычислим евклидову норму и p-норму (рис. 8).

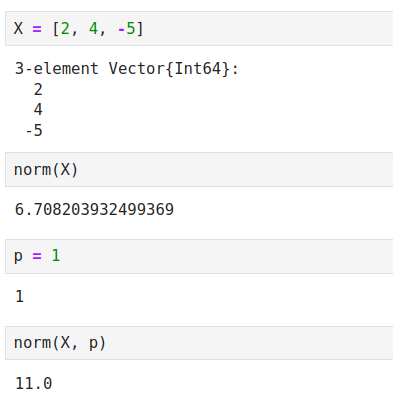


Рис. 8: Евклидова норма и p-норма вектора

Создадим второй вектор Y, найдем расстояние между X и Y (рис. 9).

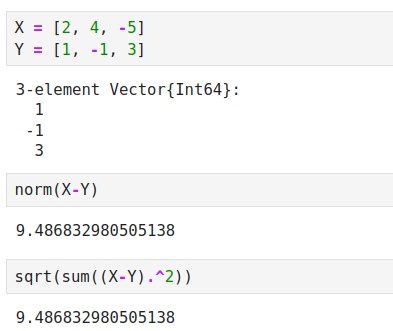


Рис. 9: Расстояние между векторами

Найдем угол между векторами X и Y (рис. 10).

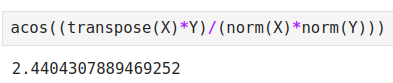


Рис. 10: Угол между векторами

Вычислим нормы для двумерной матрицы, а также повернем ее на 180 градусов, перевернем строки и столбцы (рис. 11).

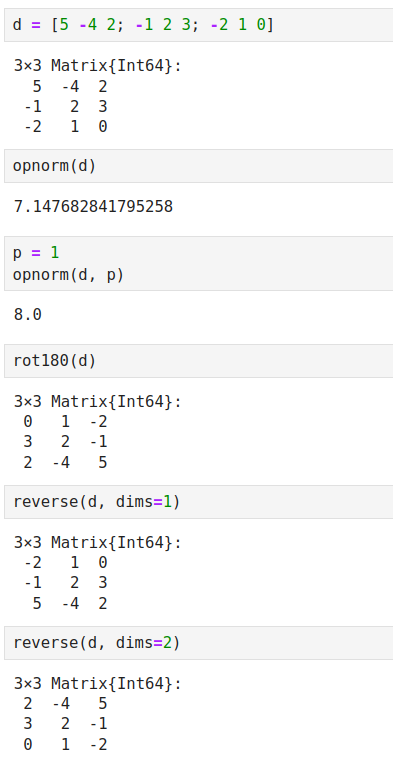


Рис. 11: Угол между векторами

## 3.3 Матричное умножение, единичная матрица, скалярное произведение

Создадим случайные матрицы 2 х 3 и 3 х 4, найдем их произведение. Создадим единичную матрицу. Создадим два вектора X и Y, найдем их скалярное произведение (рис. 12).

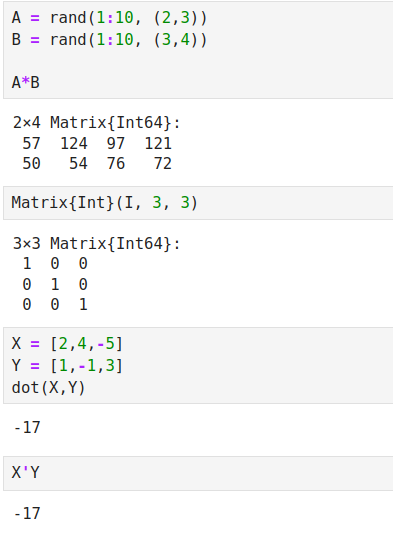


Рис. 12: Произведение матриц и скалярного векторов

## 3.4 Факторизация. Специальные матричные структуры

Для работы со специальными матричными структурами потребуется пакет LinearAlgebra.

Решим систему линейный алгебраических уравнений Ax = b (рис. 13):

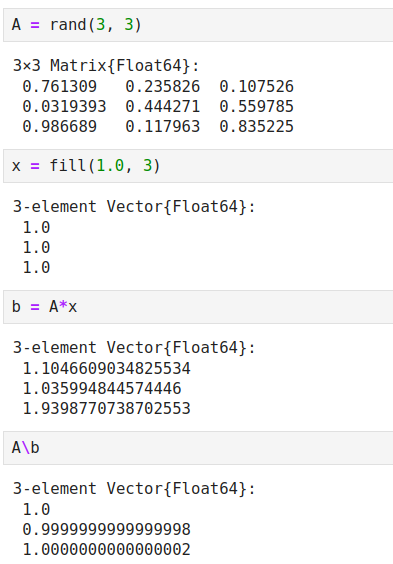


Рис. 13: Решение СЛАУ

Julia позволяет вычислять LU-факторизацию и определяет составной тип факторизации для его хранения (рис. 14):

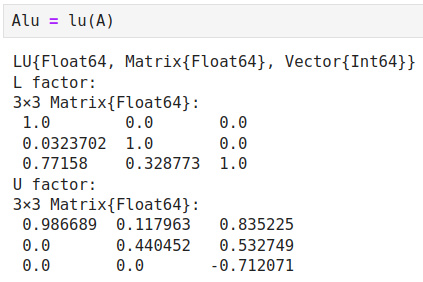


Рис. 14: Пример LU-факторизации

Различные части факторизации могут быть извлечены путём доступа к их специальным свойствам (рис. 15).

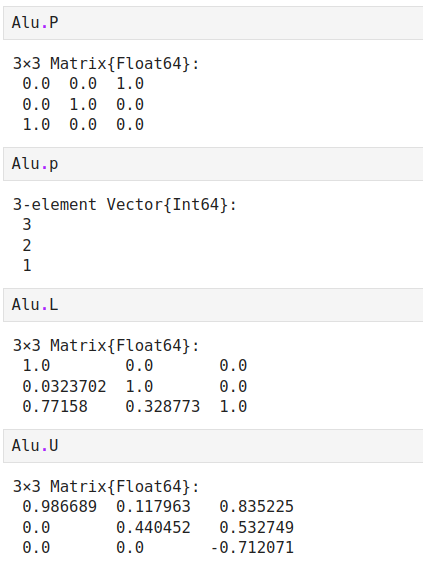


Рис. 15: Пример извлечения различных частей факторизации

Решим исходную систему, используя объекты факторизации (рис. 16).

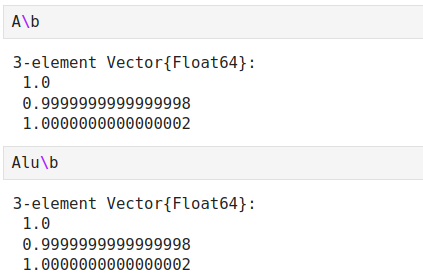


Рис. 16: Пример решения системы с использованием объектов факторизации

Найдем детерминант матрицы A (рис. 17).

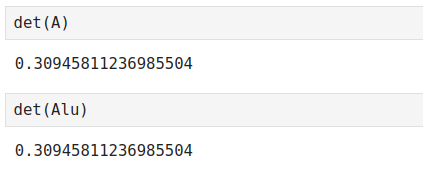


Рис. 17: Пример нахождения детерминанта матрицы А

Julia позволяет вычислять QR-факторизацию и определяет составной тип факторизации для его хранения (рис. 18).

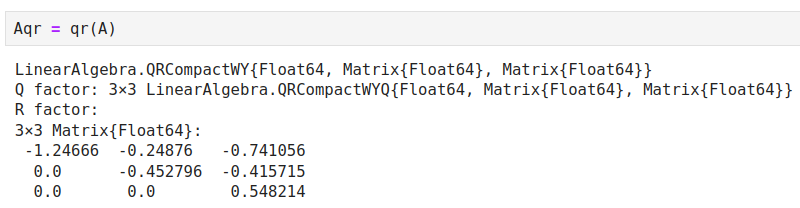


Рис. 18: Пример вычисления QR-факторизации

По аналогии с LU-факторизацией различные части QR-факторизации могут быть извлечены путём доступа к их специальным свойствам (рис. 19).

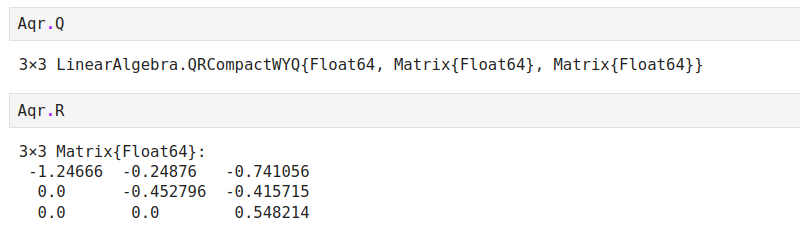


Рис. 19: Пример извлечения различных частей QR-факторизации

Проведем проверку ортогональности матрицы Q, проведем симметризацию матрицы A (рис. 20).

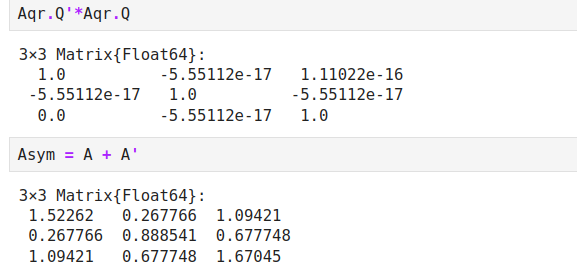


Рис. 20: Проверка ортогональности и симметризация

Сделаем спектральное разложение симметризованной матрицы, найдем собственные векторы и собственные значения (рис. 21).

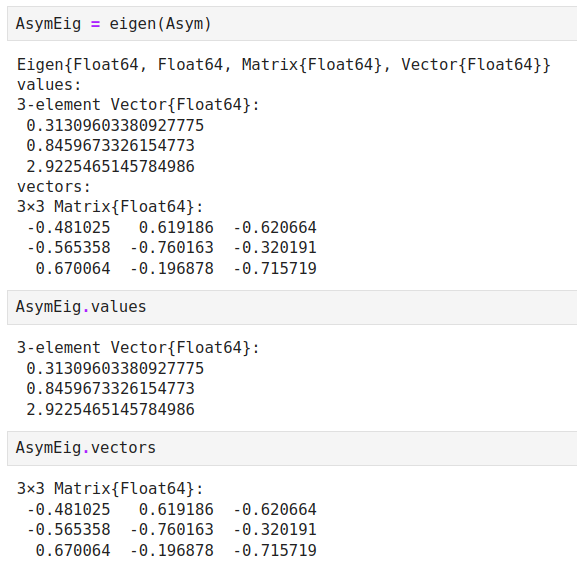


Рис. 21: Спектральное разложение, вычисление собственных векторов и собственных значений

Проверим, что получилась единичная матрица (рис. 22).

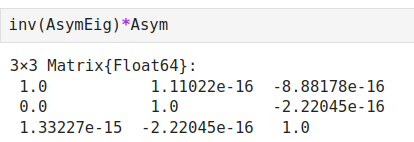


Рис. 22: Проверка на единичную матрицу

Далее рассмотрим примеры работы с матрицами большой размерности и специальной структуры. Создадим матрицу 1000 х 1000 (рис. 23).

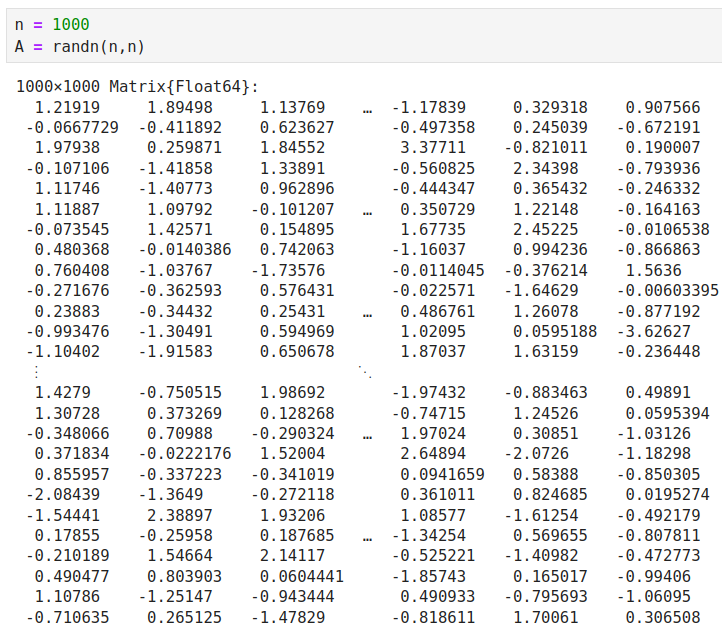


Рис. 23: Создание матрицы большой размерности

Симметризуем матрицу A и проверим, что она явдяется симметричной (рис. 24).

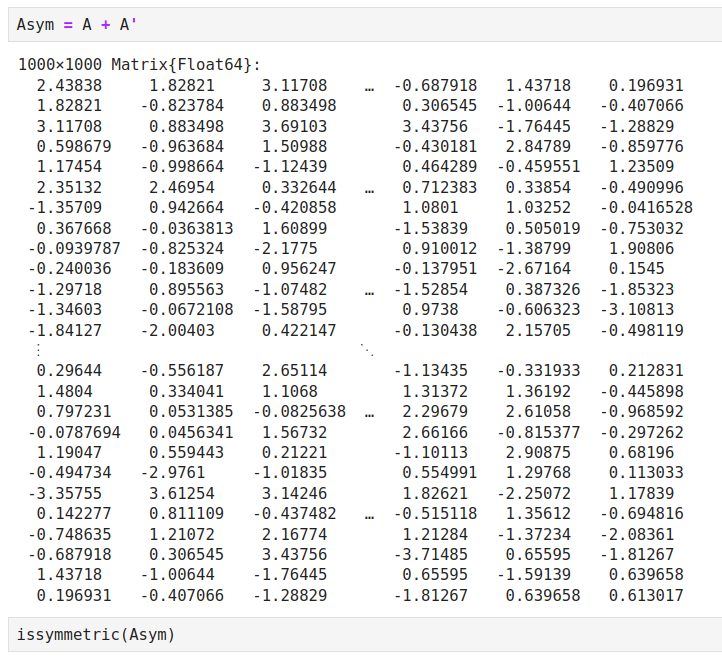


Рис. 24: Симметризация матрицы А

Добавим шум в симметричную матрицу. После этого она уже не будет симметричной (рис. 25).

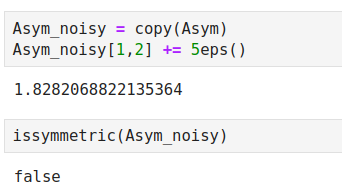


Рис. 25: Пример добавления шума к симметричной матрице

В Julia можно объявить структуру матрица явно, например, используя Diagonal, Triangular, Symmetric, Hermitian, Tridiagonal и SymTridiagonal. Явно укажем, что матрица является симметричной (рис. 26).

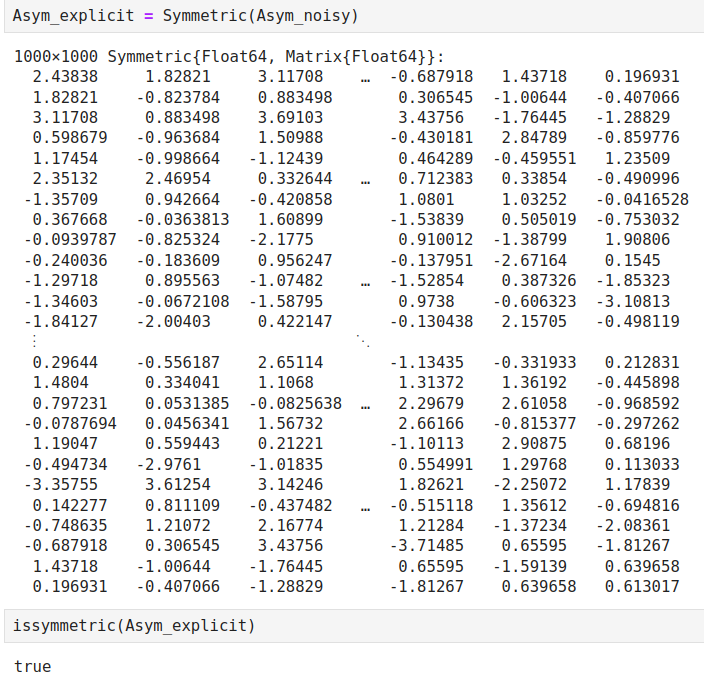


Рис. 26: Явное создание симметричной матрицы

Далее для оценки эффективности выполнения операций над матрицами большой размерности и специальной структуры воспользуемся пакетом BenchmarkTools (рис. 27).



Рис. 27: Оценка эффективности выполнения операций над матрицами большой размерости

## 3.5 Общая линейная алгебра

ulia также поддерживает общую линейную алгебру, что позволяет, например, работать с матрицами и векторами рациональных чисел. В следующем примере показано, как можно решить систему линейных уравнений с рациональными элементами без преобразования в типы элементов с плавающей запятой (для избежания проблемы с переполнением используем BigInt) (рис. 28, 29).

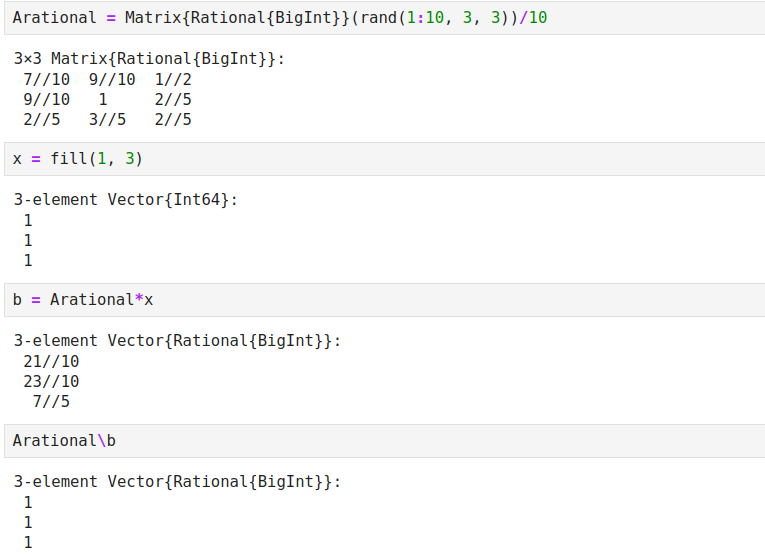


Рис. 28: Решение СЛАУ с рациональными элементами

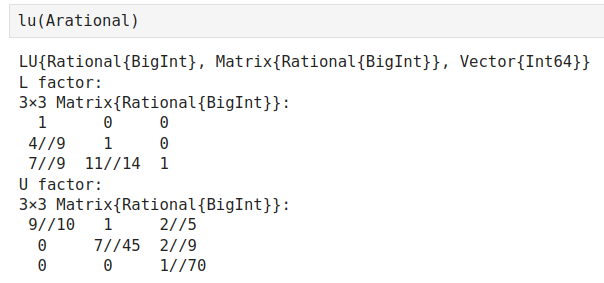


Рис. 29: LU разложение матрицы A

## 3.6 Задания для самостоятельного выполнения

Зададим вектор v. Умножим вектор v скалярно сам на себя и сохраним результат в dot\_v (рис. 30).

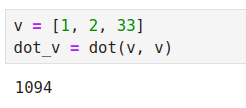


Рис. 30: Выполнение задания 1

Умножим v матрично на себя (внешнее произведение), присвоив результат переменной outer\_v (рис. 31).

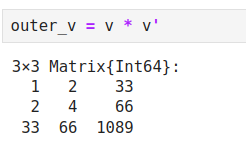


Рис. 31: Выполнение задания 1

Решим СЛАУ с двумя неизвестными.

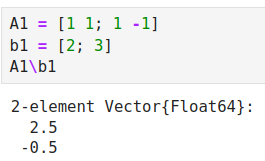


Рис. 32: Выполнение задания 2

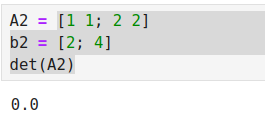


Рис. 33: Выполнение задания 2

Система не имеет решения (рис. 33).

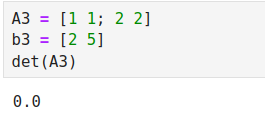


Рис. 34: Выполнение задания 2

Система не имеет решения (рис. 34).

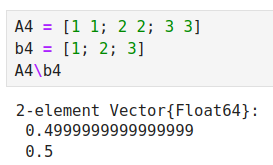


Рис. 35: Выполнение задания 2

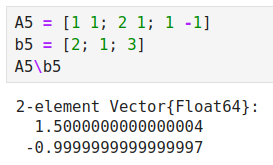


Рис. 36: Выполнение задания 2

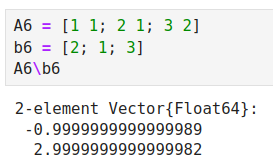


Рис. 37: Выполнение задания 2

Решим СЛАУ с тремя неизветными.

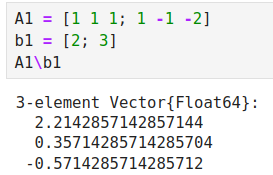


Рис. 38: Выполнение задания 2

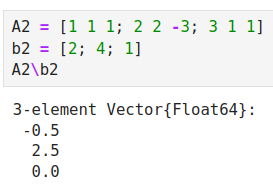


Рис. 39: Выполнение задания 2

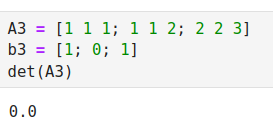


Рис. 40: Выполнение задания 2

Система не имеет решения (рис. 40).

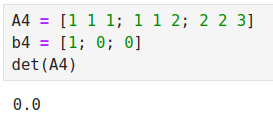


Рис. 41: Выполнение задания 2

Система не имеет решения (рис. 41).

Приведем приведенные ниже матрицы к диагональному виду.

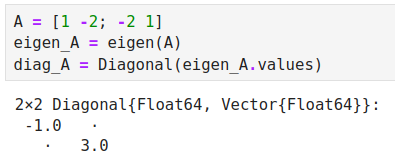


Рис. 42: Выполнение задания 3

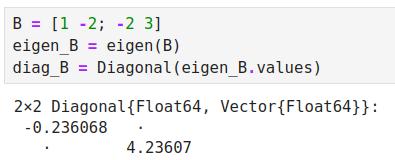


Рис. 43: Выполнение задания 3

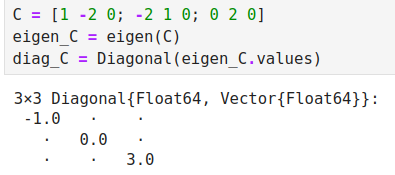


Рис. 44: Выполнение задания 3

Вычислим выражения.

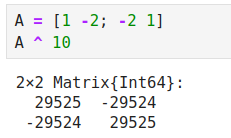


Рис. 45: Выполнение задания 3

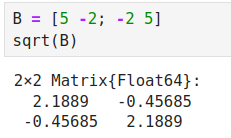


Рис. 46: Выполнение задания 3

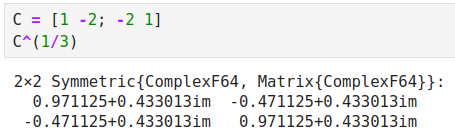


Рис. 47: Выполнение задания 3

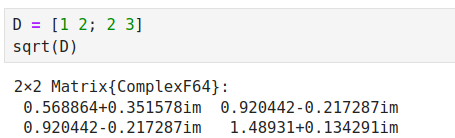


Рис. 48: Выполнение задания 3

Найдем собственные значения матрицы А.

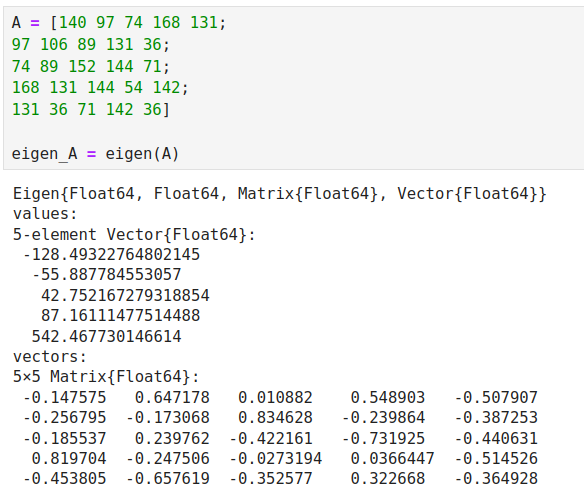


Рис. 49: Выполнение задания 3

Оценим эффективность выполняемых операций (рис. 50).

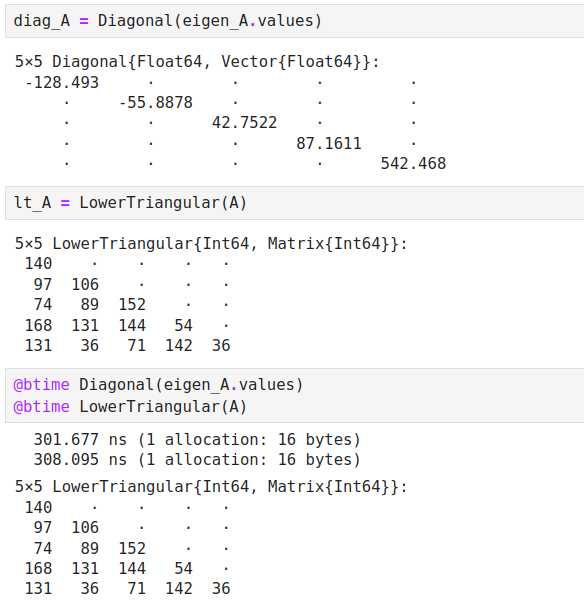


Рис. 50: Выполнение задания 3

Определим, являются ли матрицы продуктивными (рис. 51-53).

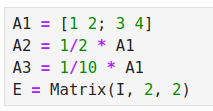


Рис. 51: Выполнение задания 4

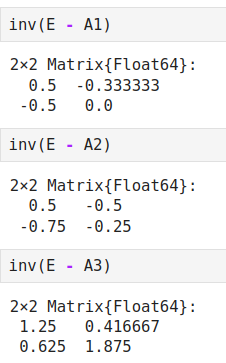


Рис. 52: Выполнение задания 4



Рис. 53: Выполнение задания 4

# 4 Вывод

Изучили возможности специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.