Отчёт по лабораторной работе №4  
Компьютерный практикум по статистическому анализу данных

Линейная алгебра

Выполнил: Махорин Иван Сергеевич,  
НПИбд-02-21, 1032211221

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

## 2.1 Поэлементные операции над многомерными массивами

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

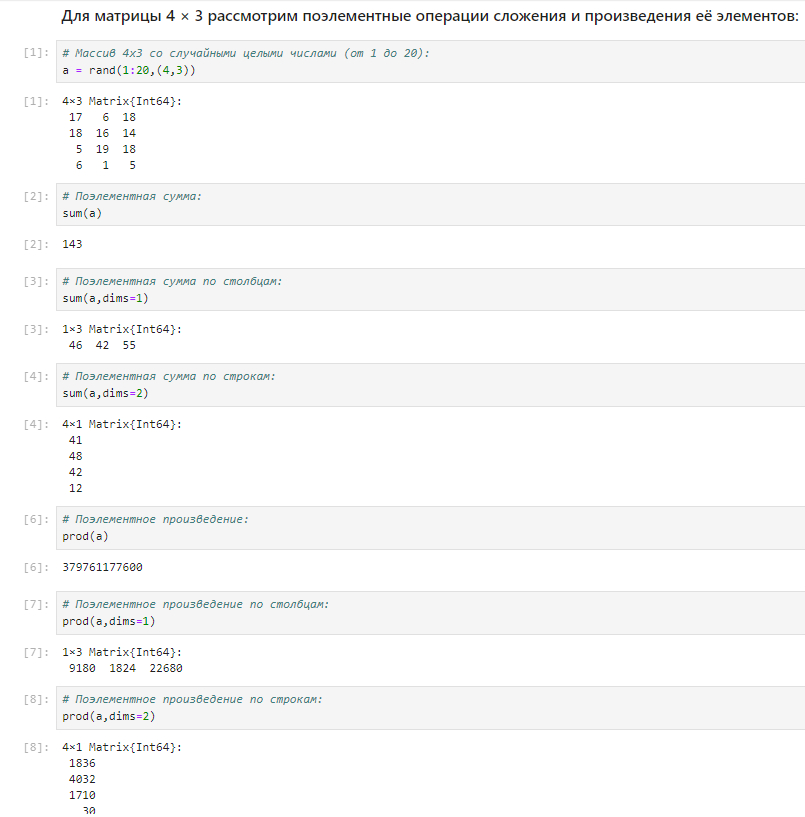


Рис. 1: Поэлементные операции сложения и произведения элементов матрицы

Для работы со средними значениями можно воспользоваться возможностями пакета Statistics (рис. 2):

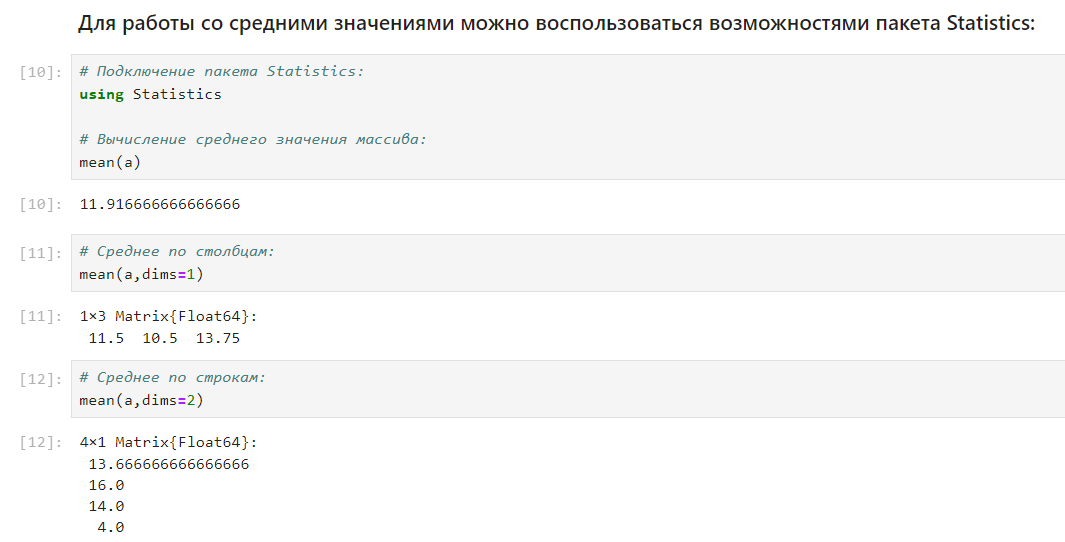


Рис. 2: Использование возможностей пакета Statistics для работы со средними значениями

## 2.2 Транспонирование, след, ранг, определитель и инверсия матрицы

Для выполнения таких операций над матрицами, как транспонирование, диагонализация, определение следа, ранга, определителя матрицы и т.п. можно воспользоваться библиотекой (пакетом) LinearAlgebra (рис. 3 - рис. 4):

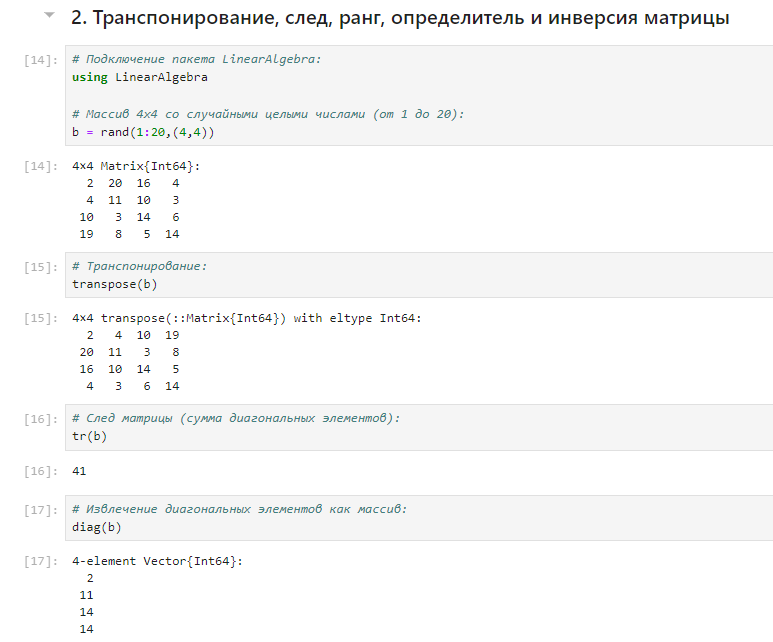


Рис. 3: Использование библиотеки LinearAlgebra для выполнения определённых операций

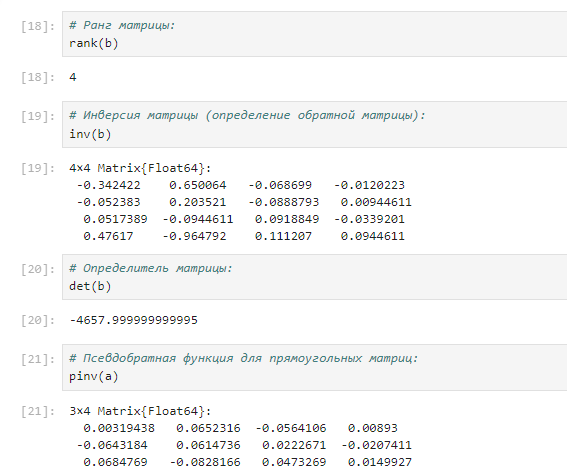


Рис. 4: Использование библиотеки LinearAlgebra для выполнения определённых операций

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

Для вычисления нормы используется LinearAlgebra.norm(x) (рис. 5):

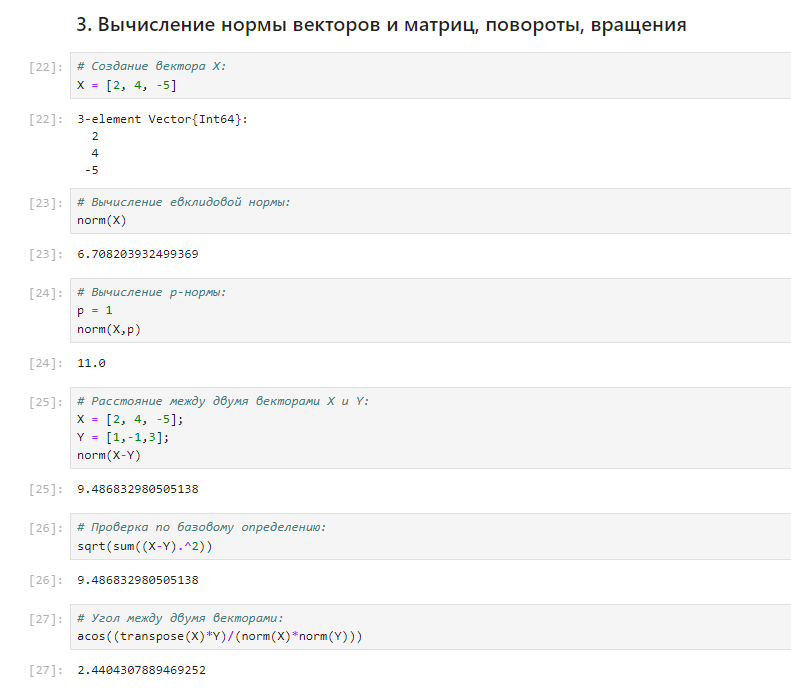


Рис. 5: Использование LinearAlgebra.norm(x)

Вычислим нормы для двумерной матрицы (рис. 6):

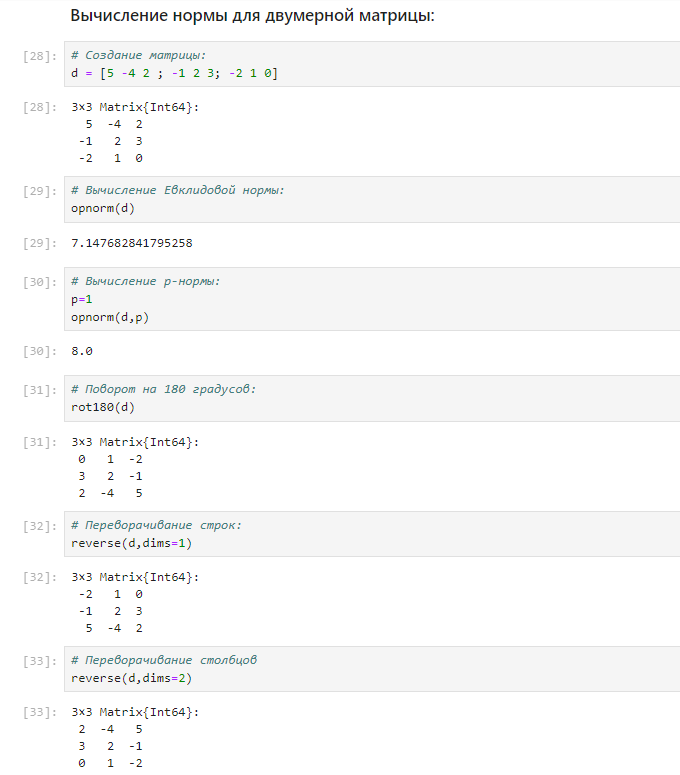


Рис. 6: Вычисление нормы для двумерной матрицы

## 2.4 Матричное умножение, единичная матрица, скалярное произведение

Выполним примеры матричного умножения, единичной матрицы и скалярного произведения (рис. 7):

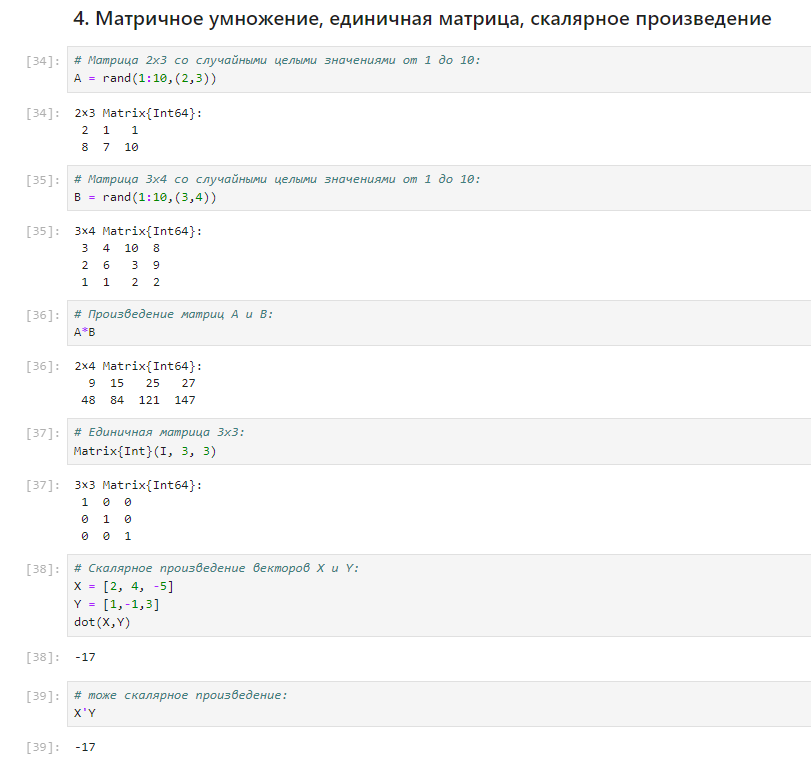


Рис. 7: Примеры матричного умножения, единичной матрицы и скалярного произведения

## 2.5 Факторизация. Специальные матричные структуры

Рассмотрим несколько примеров. Для работы со специальными матричными структурами потребуется пакет LinearAlgebra.

Решение систем линейный алгебраических уравнений 𝐴𝑥 = 𝑏 (рис. 8):

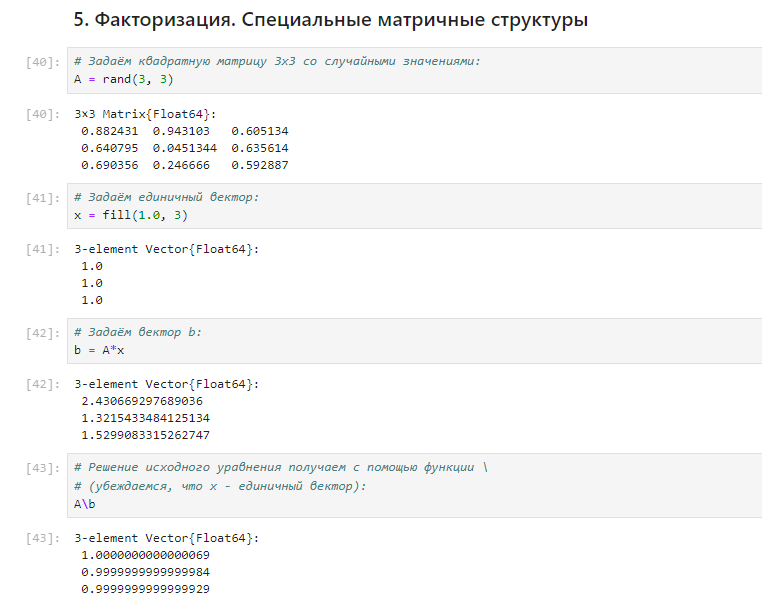


Рис. 8: Решение систем линейный алгебраических уравнений 𝐴𝑥 = 𝑏

Julia позволяет вычислять LU-факторизацию и определяет составной тип факторизации для его хранения (рис. 9):

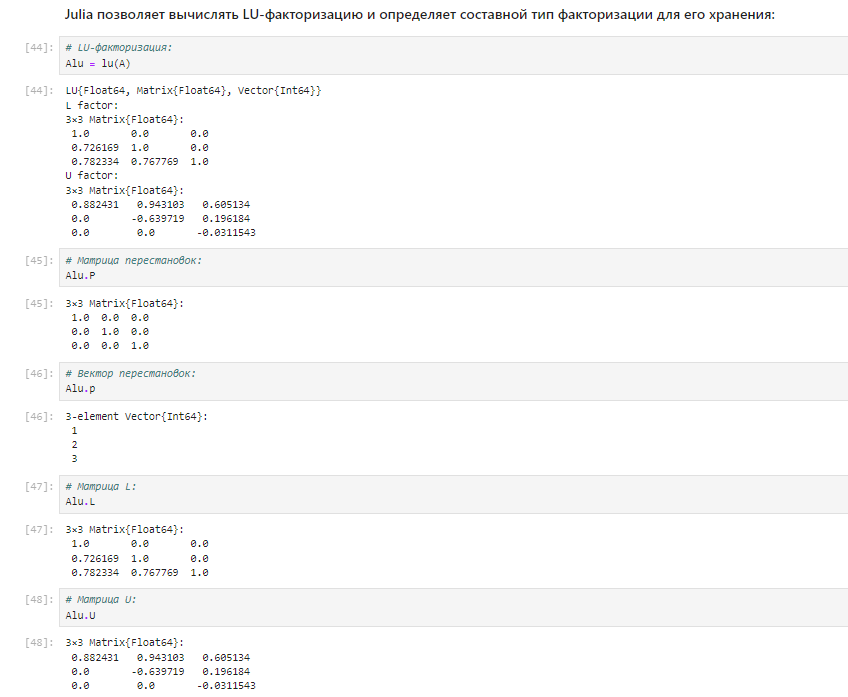


Рис. 9: Пример вычисления LU-факторизации и определение составного типа факторизации для его хранения

Исходная система уравнений 𝐴𝑥 = 𝑏 может быть решена или с использованием исходной матрицы, или с использованием объекта факторизации (рис. 10):

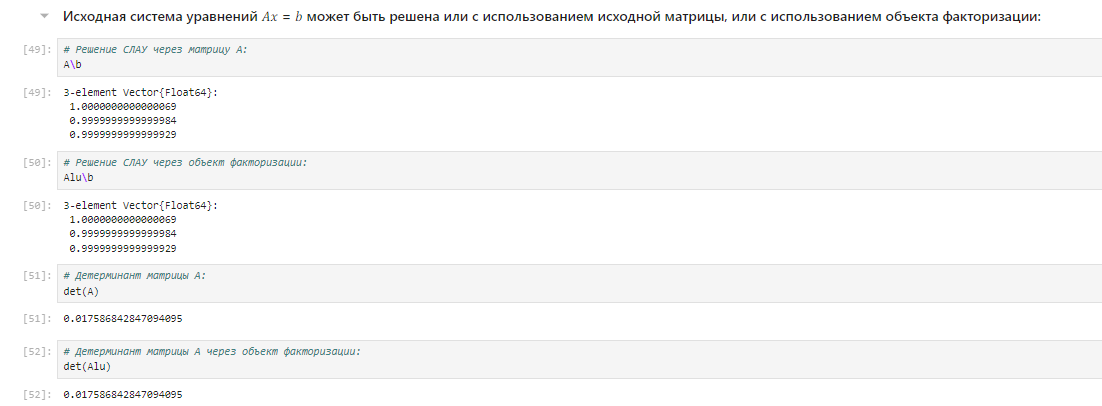


Рис. 10: Пример решения с использованием исходной матрицы и с использованием объекта факторизации

Julia позволяет вычислять QR-факторизацию и определяет составной тип факторизации для его хранения (рис. 11):

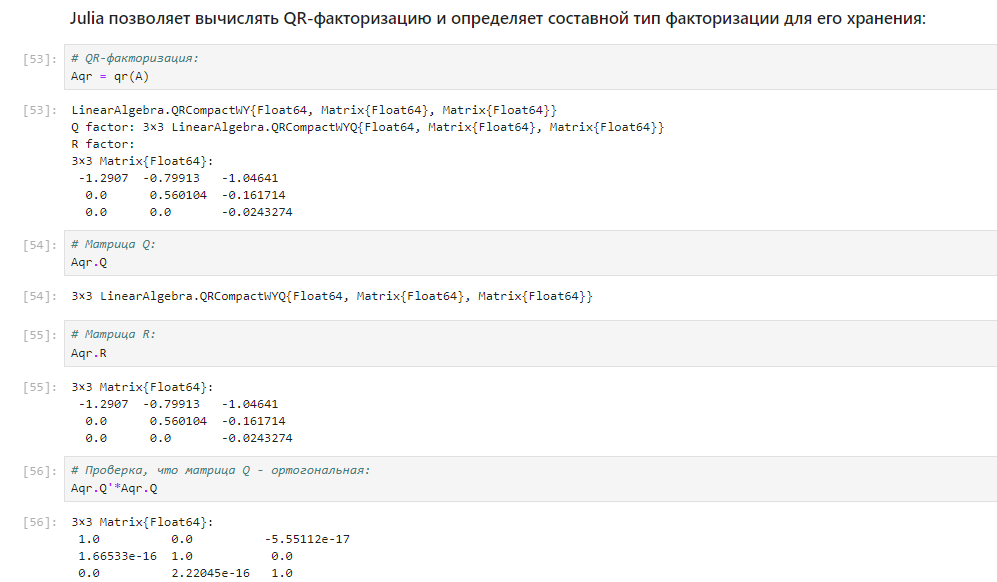


Рис. 11: Пример вычисления QR-факторизации и определение составного типа факторизации для его хранения

Примеры собственной декомпозиции матрицы 𝐴 (рис. 12):

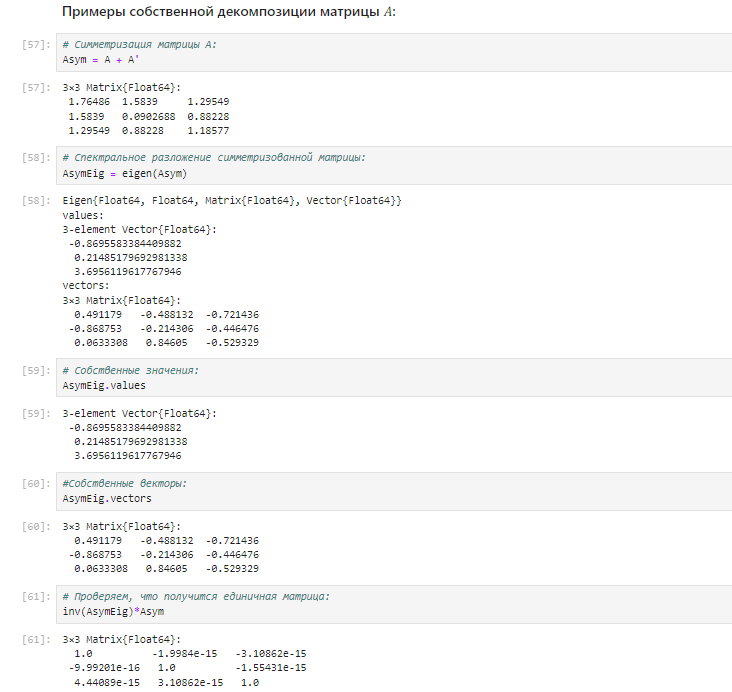


Рис. 12: Примеры собственной декомпозиции матрицы 𝐴

Далее рассмотрим примеры работы с матрицами большой размерности и специальной структуры (рис. 13):

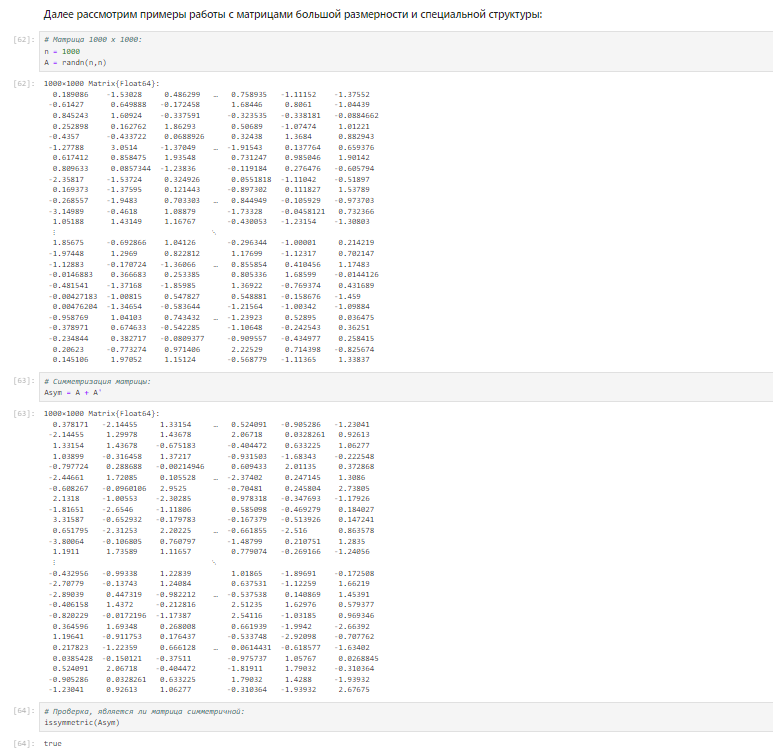


Рис. 13: Примеры работы с матрицами большой размерности и специальной структуры

Пример добавления шума в симметричную матрицу (матрица уже не будет симметричной) (рис. 14):

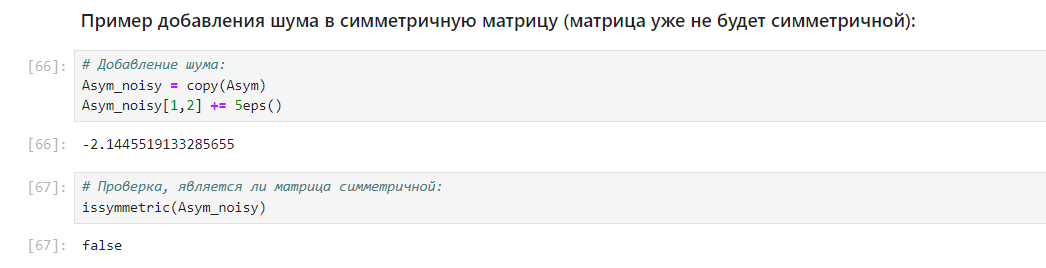


Рис. 14: Пример добавления шума в симметричную матрицу

В Julia можно объявить структуру матрица явно, например, используя Diagonal, Triangular, Symmetric, Hermitian, Tridiagonal и SymTridiagonal (рис. 15):

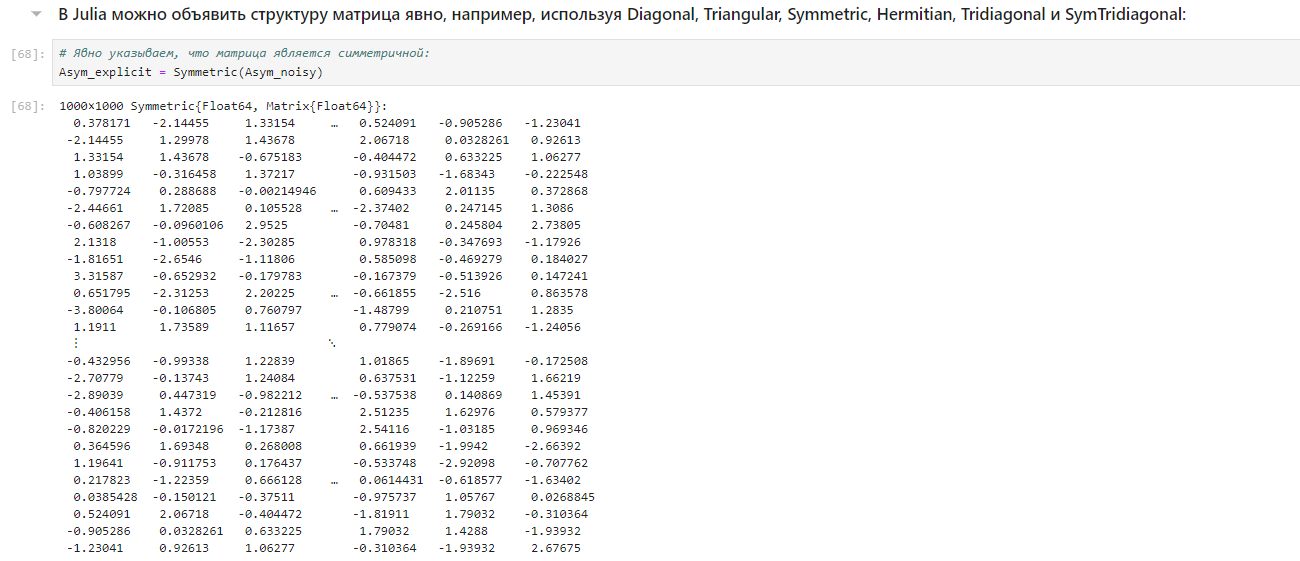


Рис. 15: Пример явного объявления структуры матрицы

Далее для оценки эффективности выполнения операций над матрицами большой размерности и специальной структуры воспользуемся пакетом BenchmarkTools (рис. 16):

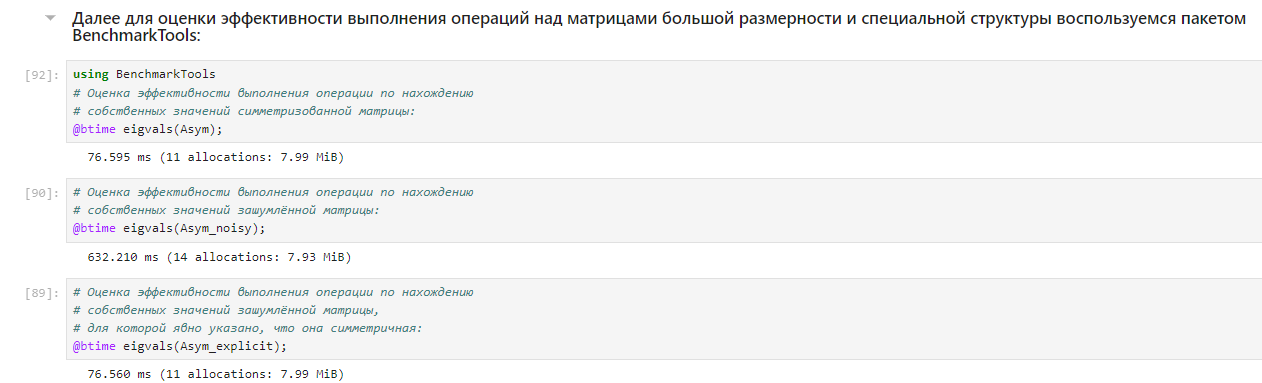


Рис. 16: Использование пакета BenchmarkTools

Далее рассмотрим примеры работы с разряженными матрицами большой размерности.

Использование типов Tridiagonal и SymTridiagonal для хранения трёхдиагональных матриц позволяет работать с потенциально очень большими трёхдиагональными матрицами (рис. 17):

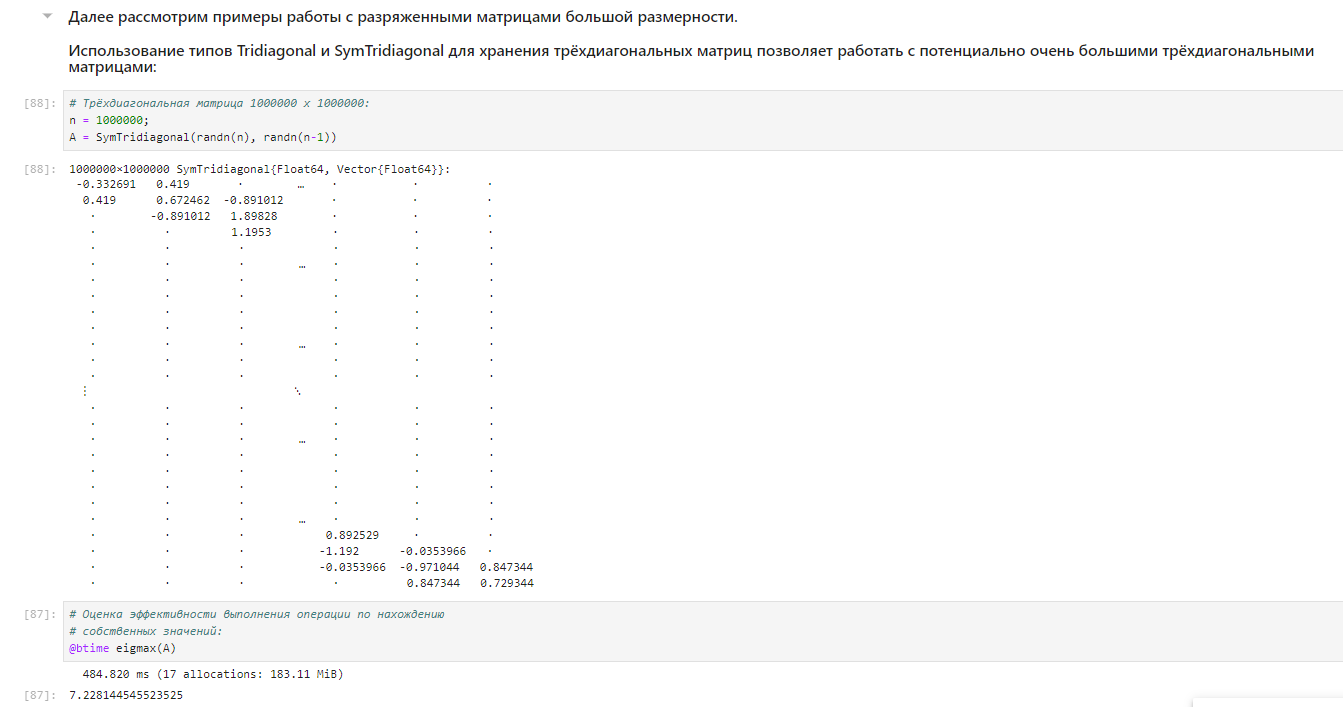


Рис. 17: Примеры работы с разряженными матрицами большой размерности

## 2.6 Общая линейная алгебра

В 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. 18):

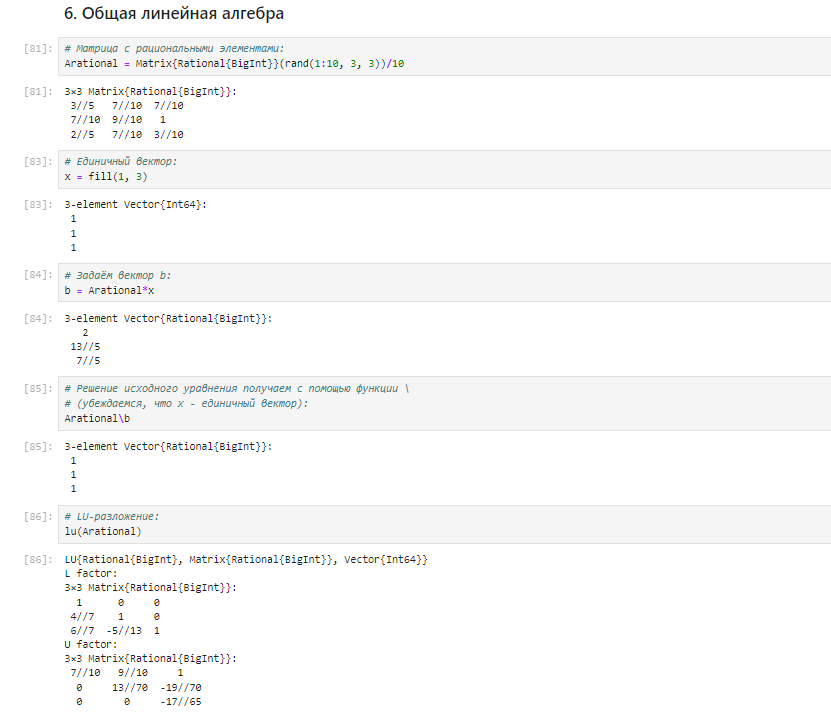


Рис. 18: Решение системы линейных уравнений с рациональными элементами без преобразования в типы элементов с плавающей запятой

## 2.7 Самостоятельная работа

Выполнение задания “Произведение векторов” (рис. 19):

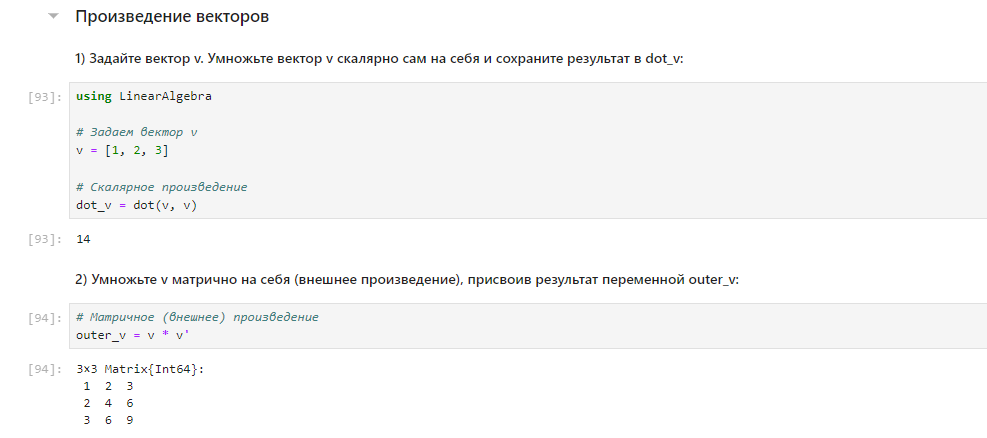


Рис. 19: Решение задания “Произведение векторов”

Выполнение задания “Системы линейных уравнений” (рис. 20 - рис. 21):

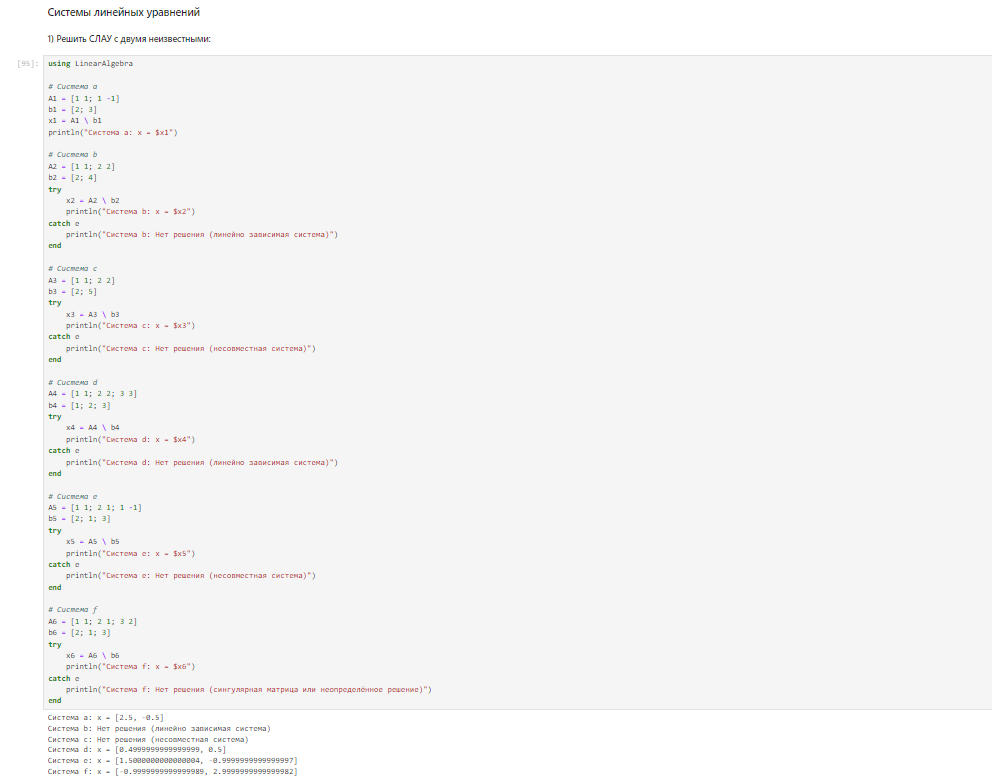


Рис. 20: Решение задания “Системы линейных уравнений”



Рис. 21: Решение задания “Системы линейных уравнений”

Выполнение задания “Операции с матрицами” (рис. 22 - рис. 26):

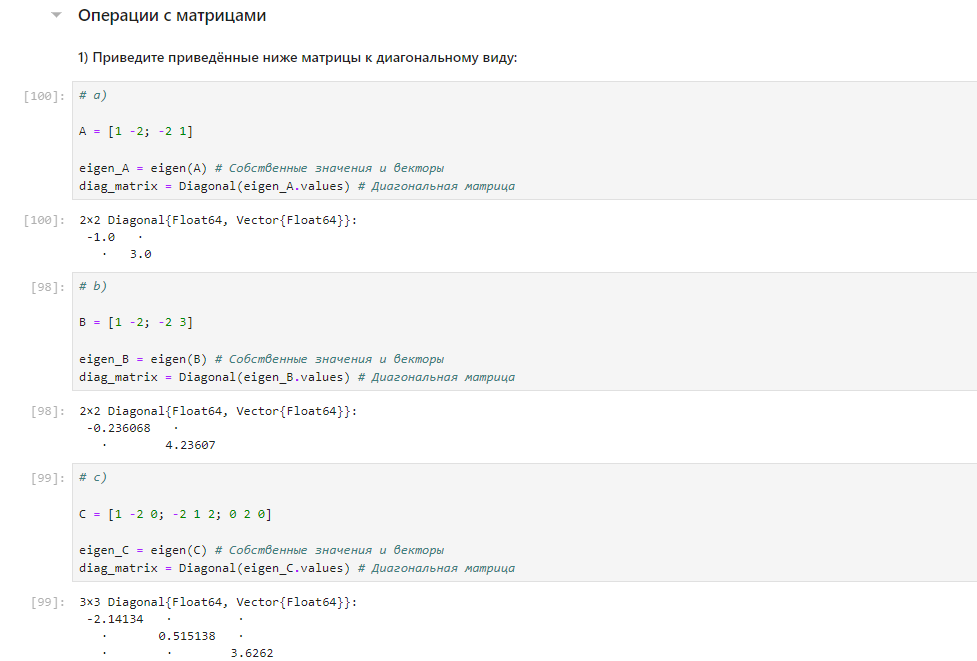


Рис. 22: Решение задания “Операции с матрицами”

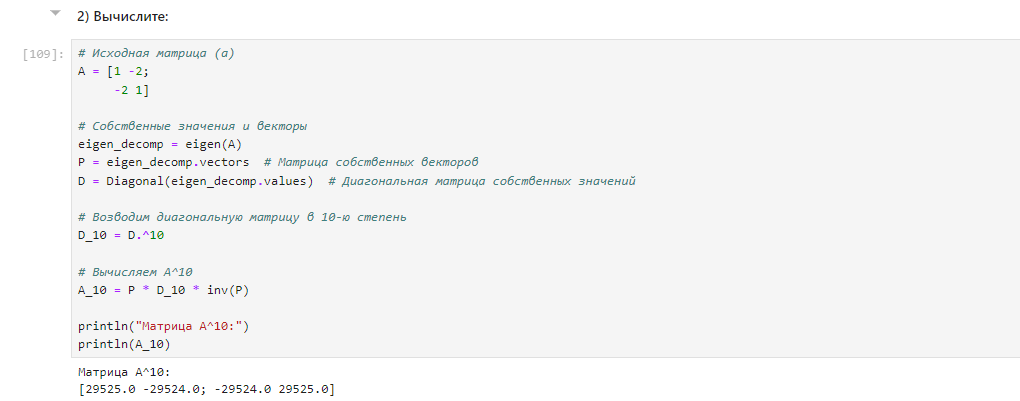


Рис. 23: Решение задания “Операции с матрицами”



Рис. 24: Решение задания “Операции с матрицами”

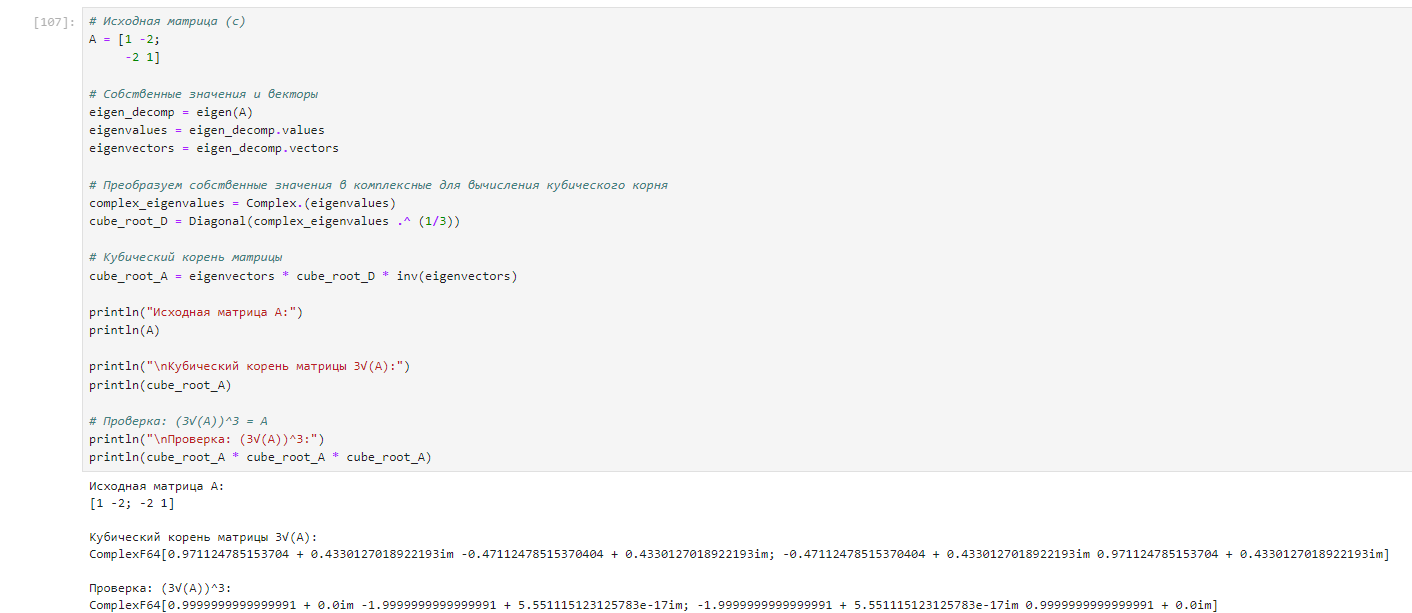


Рис. 25: Решение задания “Операции с матрицами”

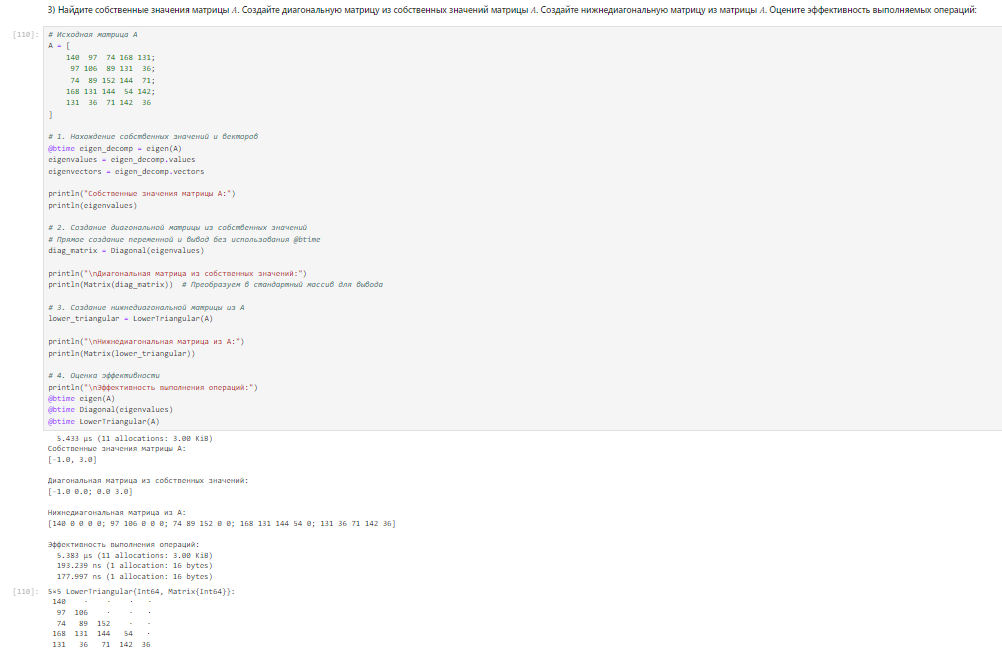


Рис. 26: Решение задания “Операции с матрицами”

Выполнение задания “Линейные модели экономики” (рис. 27):

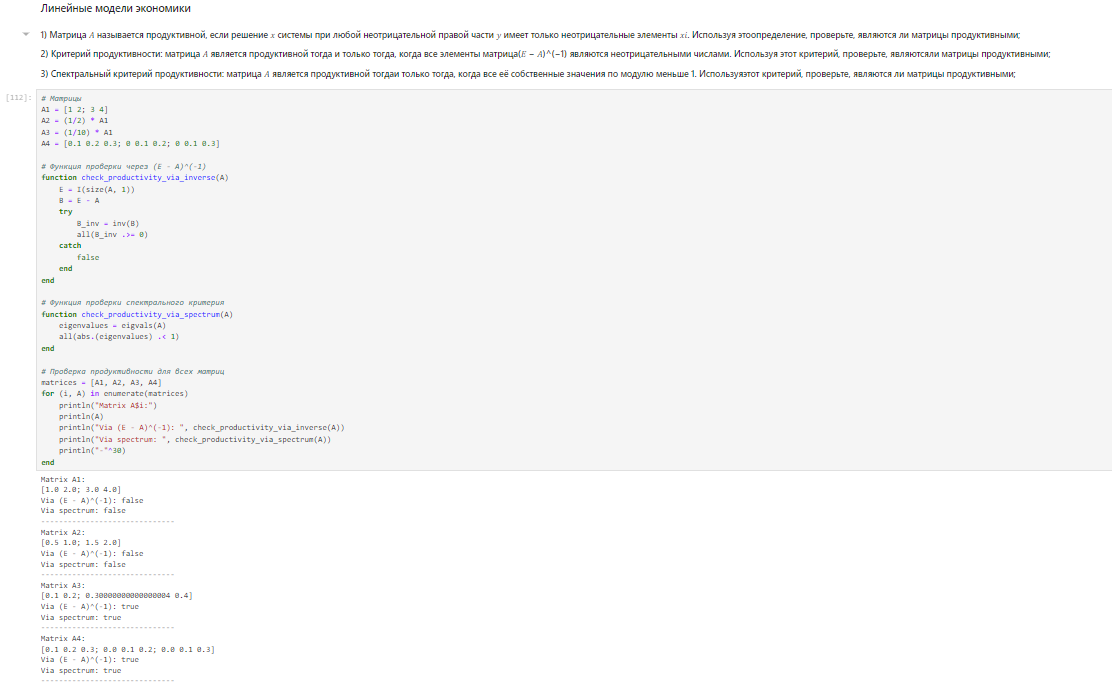


Рис. 27: Решение задания “Линейные модели экономики”

# 3 Вывод

В ходе выполнения лабораторной работы были изучены возможности специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.

# 4 Список литературы. Библиография

[1] Julia Documentation: https://docs.julialang.org/en/v1/