Отчет по лабораторной работе 4

Петрушов Дмитрий, 1032212287

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](#fig:001)]):

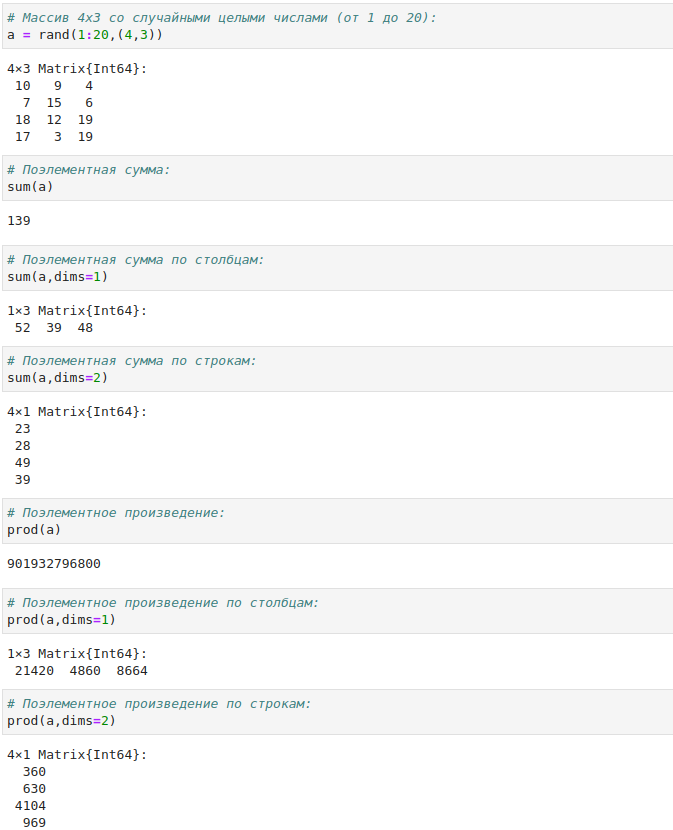


Figure 1: Поэлементные операции сложения и произведения элементов матрицы

Для работы со средними значениями можно воспользоваться возможностями пакета Statistics (рис. [[2](#fig:002)]):

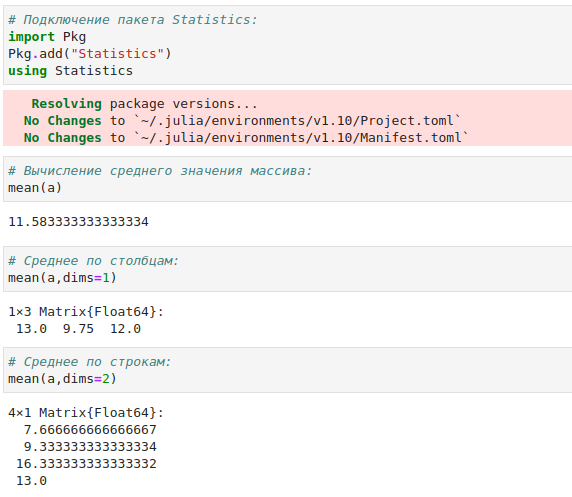


Figure 2: Использование возможностей пакета Statistics для работы со средними значениями

## 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](#fig:003)] - рис. [[4](#fig:004)]):

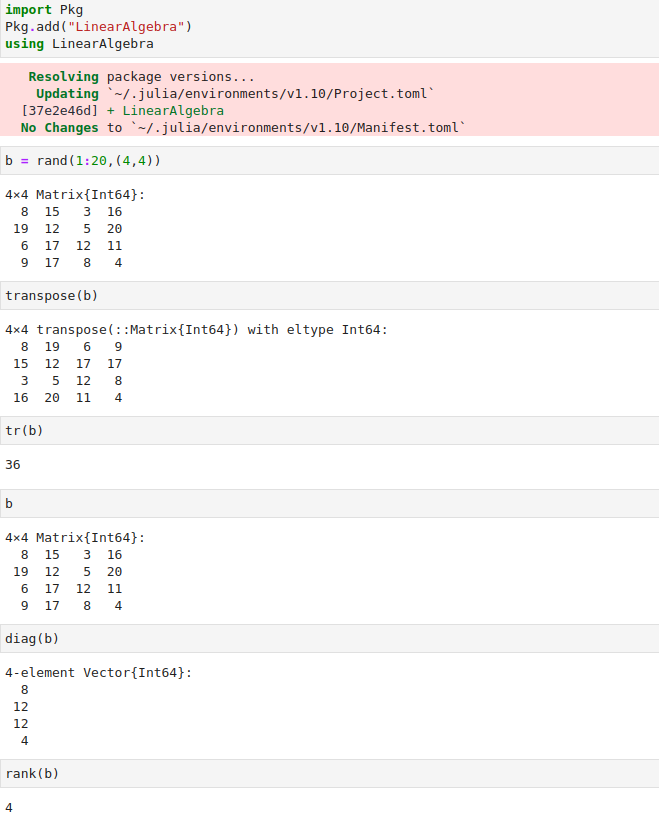


Figure 3: Использование библиотеки LinearAlgebra для выполнения определённых операций

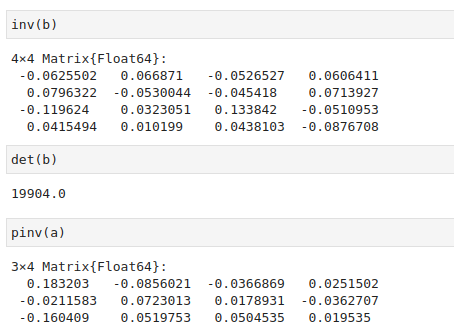


Figure 4: Использование библиотеки LinearAlgebra для выполнения определённых операций

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

Для вычисления нормы используется LinearAlgebra.norm(x) (рис. [[5](#fig:005)]):



Figure 5: Использование LinearAlgebra.norm(x)

Вычислим нормы для двумерной матрицы (рис. [[6](#fig:006)]):

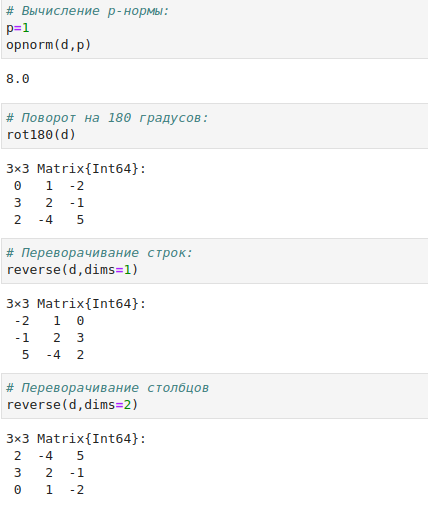


Figure 6: Вычисление нормы для двумерной матрицы

## 2.4 Матричное умножение, единичная матрица, скалярное произведение

Выполним примеры матричного умножения, единичной матрицы и скалярного произведения (рис. [[7](#fig:007)]):



Figure 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](#fig:008)]):

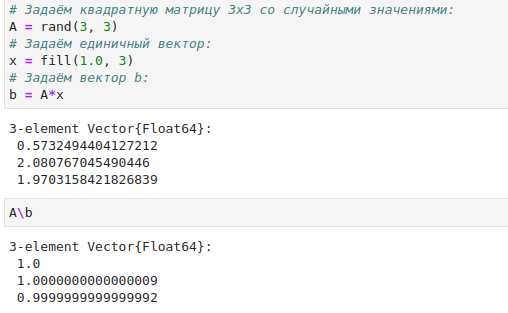


Figure 8: Решение систем линейный алгебраических уравнений Ax = b

Julia позволяет вычислять LU-факторизацию и определяет составной тип факторизации для его хранения (рис. [[9](#fig:009)]):

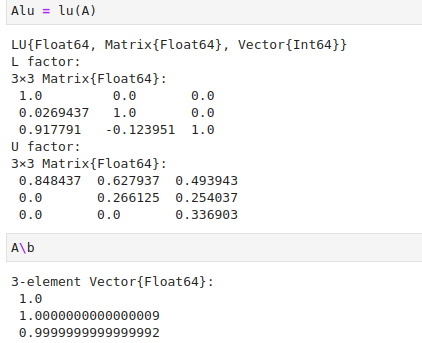


Figure 9: Пример вычисления LU-факторизации и определение составного типа факторизации для его хранения

Исходная система уравнений Ax = b может быть решена или с использованием исходной матрицы, или с использованием объекта факторизации (рис. [[10](#fig:010)]):

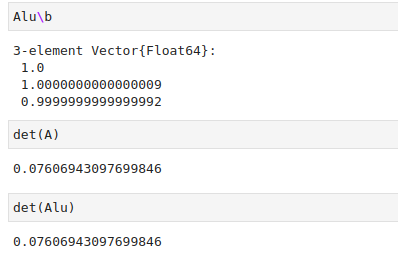


Figure 10: Пример решения с использованием исходной матрицы и с использованием объекта факторизации

Julia позволяет вычислять QR-факторизацию и определяет составной тип факторизации для его хранения (рис. [[11](#fig:011)]):

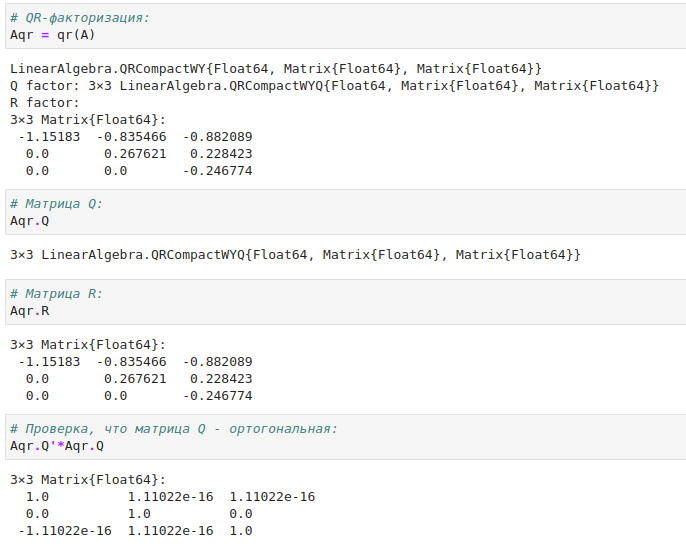


Figure 11: Пример вычисления QR-факторизации и определение составного типа факторизации для его хранения

Примеры собственной декомпозиции матрицы A (рис. [[12](#fig:012)]):

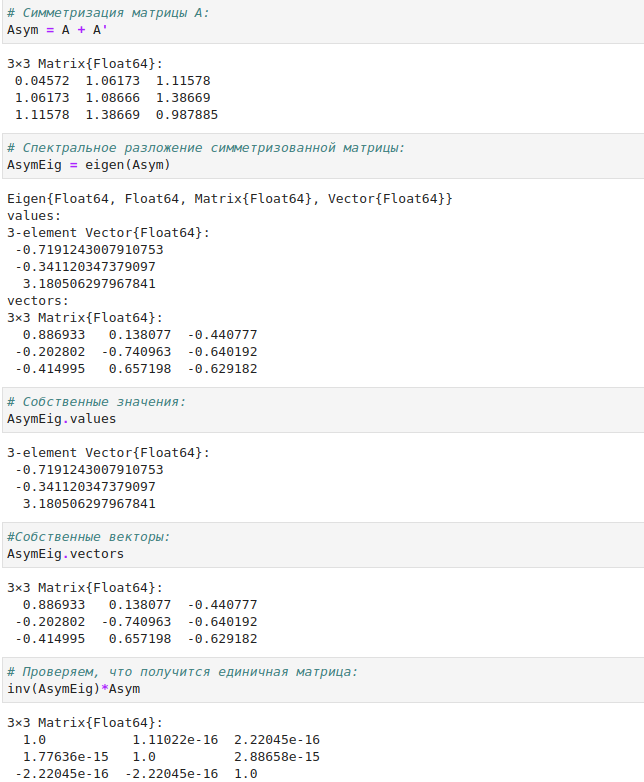


Figure 12: Примеры собственной декомпозиции матрицы A

Далее рассмотрим примеры работы с матрицами большой размерности и специальной структуры (рис. [[13](#fig:013)]):

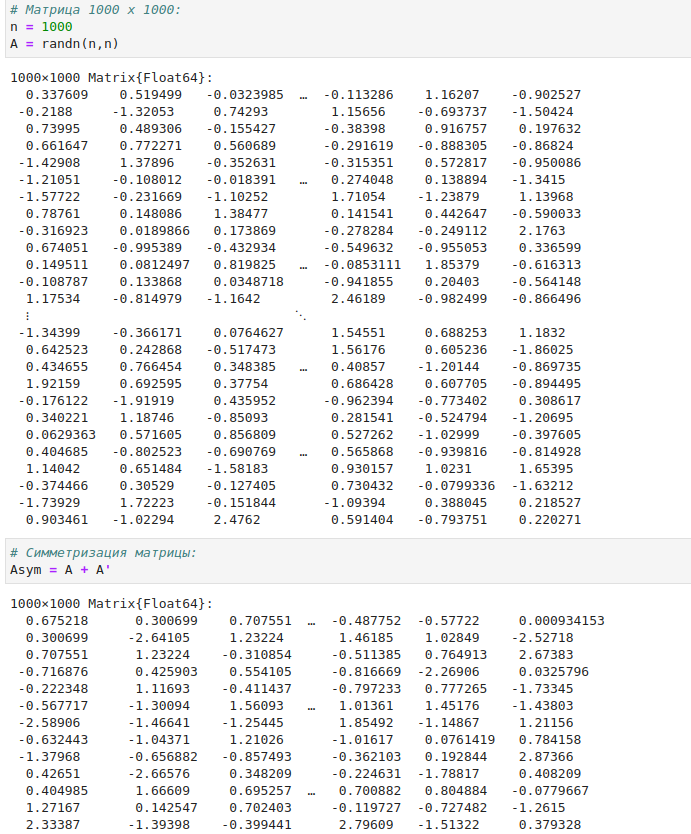


Figure 13: Примеры работы с матрицами большой размерности и специальной структуры

Пример добавления шума в симметричную матрицу (матрица уже не будет симметричной) (рис. [[14](#fig:014)]):

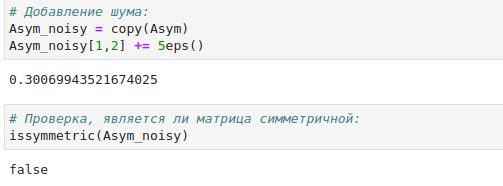


Figure 14: Пример добавления шума в симметричную матрицу

В Julia можно объявить структуру матрица явно, например, используя Diagonal, Triangular, Symmetric, Hermitian, Tridiagonal и SymTridiagonal (рис. [[15](#fig:015)]):

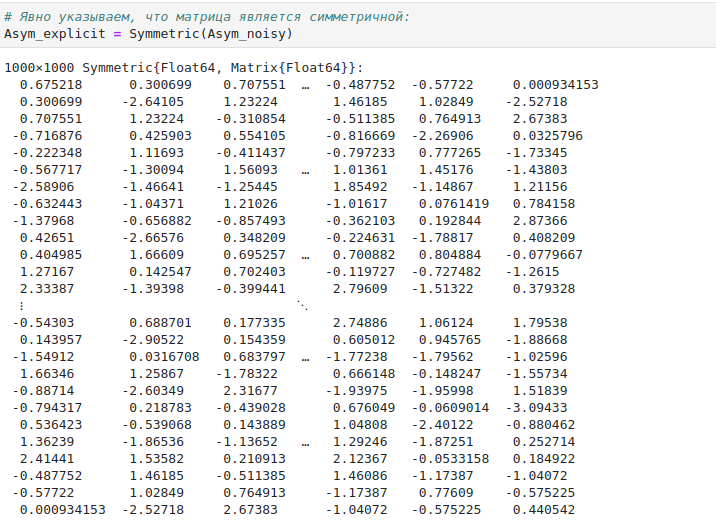


Figure 15: Пример явного объявления структуры матрицы

Далее для оценки эффективности выполнения операций над матрицами большой размерности и специальной структуры воспользуемся пакетом BenchmarkTools (рис. [[16](#fig:016)]):



Figure 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](#fig:017)]):

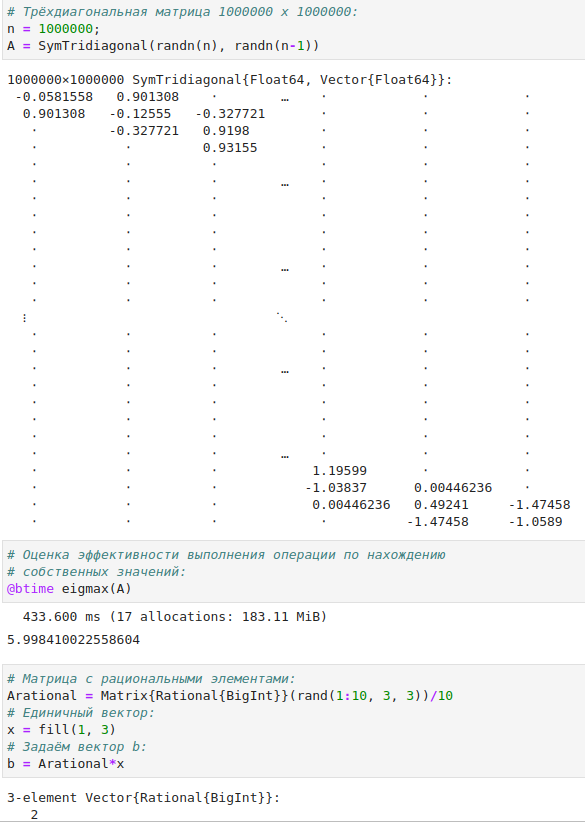


Figure 17: Примеры работы с разряженными матрицами большой размерности

## 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](#fig:018)]):

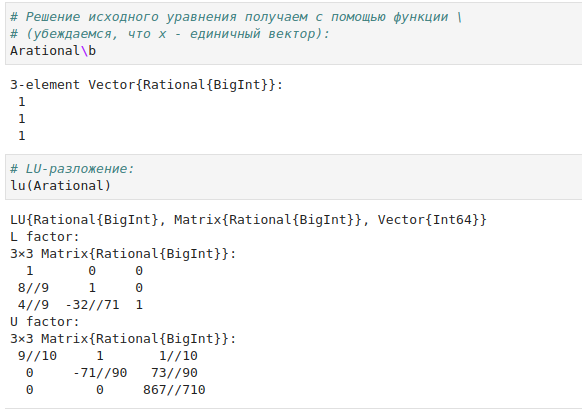


Figure 18: Решение системы линейных уравнений с рациональными элементами без преобразования в типы элементов с плавающей запятой

## 2.7 Самостоятельная работа

Выполнение задания “Произведение векторов” (рис. [[19](#fig:019)]):

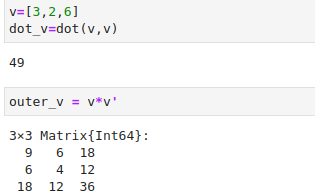


Figure 19: Решение задания “Произведение векторов”

Выполнение задания “Системы линейных уравнений” (рис. [[20](#fig:020)] - рис. [[21](#fig:021)]):

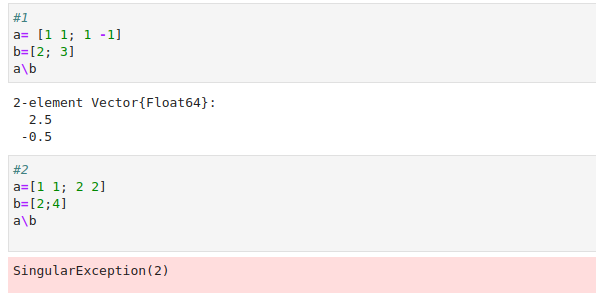


Figure 20: Решение задания “Системы линейных уравнений”

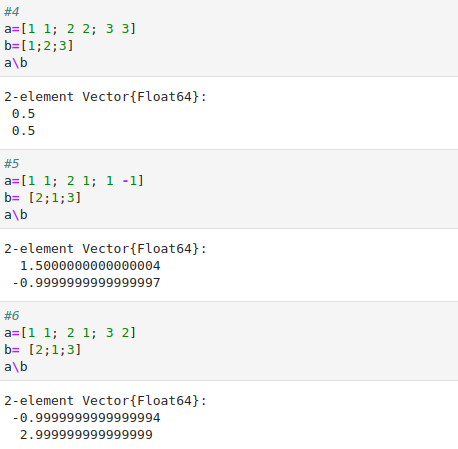


Figure 21: Решение задания “Системы линейных уравнений”

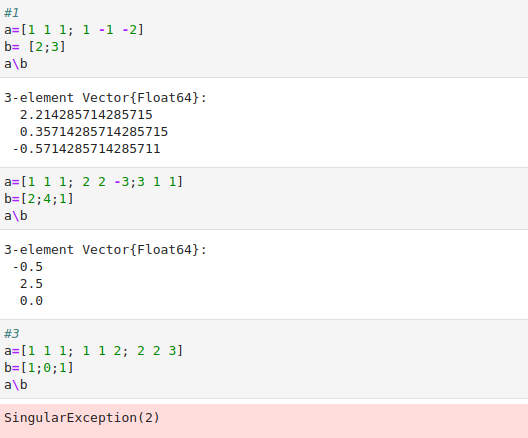


Figure 22: Решение задания “Системы линейных уравнений”

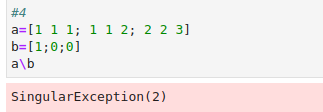


Figure 23: Решение задания “Системы линейных уравнений”

Выполнение задания “Операции с матрицами” (рис. [[22](#fig:022)] - рис. [[26](#fig:026)]):

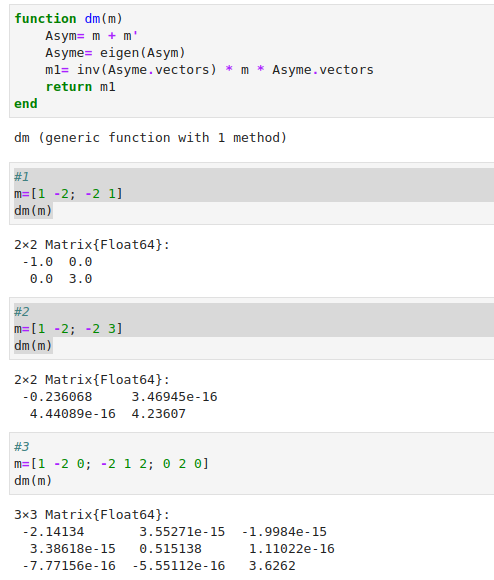


Figure 24: Решение задания “Операции с матрицами”

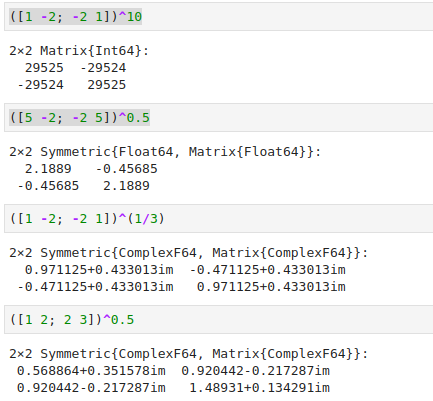


Figure 25: Решение задания “Операции с матрицами”

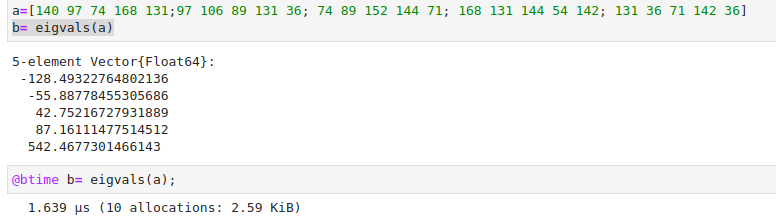


Figure 26: Решение задания “Операции с матрицами”

Выполнение задания “Линейные модели экономики” (рис. [[27](#fig:027)]):

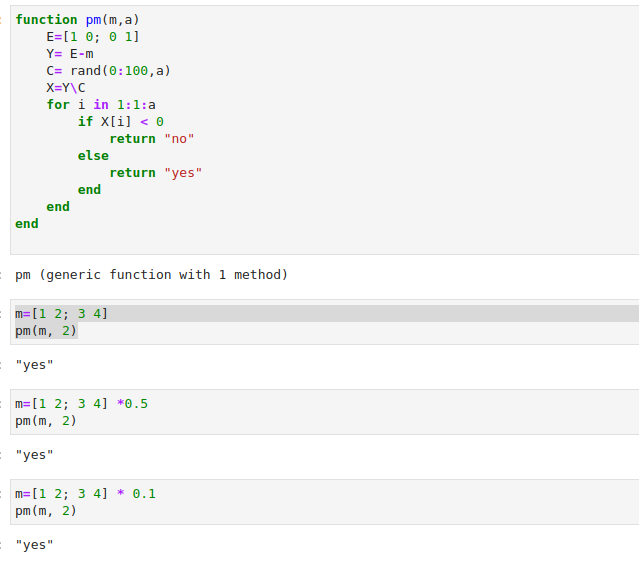


Figure 27: Решение задания “Линейные модели экономики”

# 3 Вывод

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

# Список литературы