**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

*дисциплина: Компьютерный практикум по математическому моделированию*

Студент: Абрамян Артём

Группа: НПИбд-01-20

**МОСКВА**

2023 г.

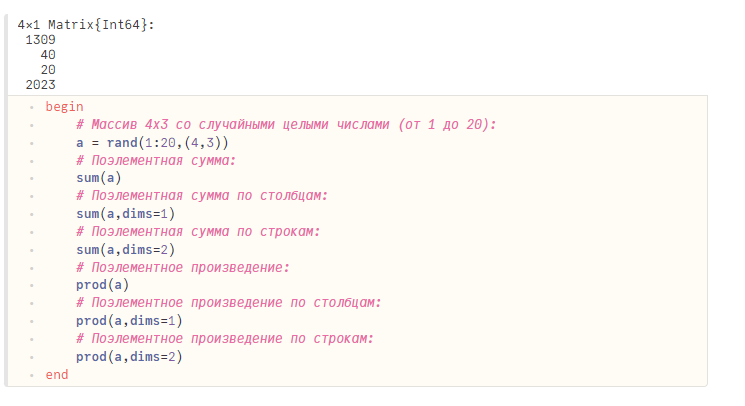
**Постановка задачи**

Изучение возможностей специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.

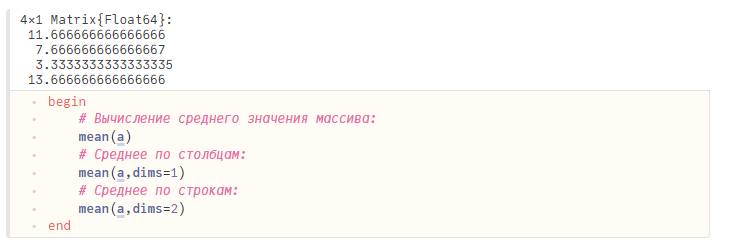
**Выполнение работы**

1. **Используя Jupyter Lab, повторите примеры из раздела 4.2.**
   1. **Поэлементные операции над многомерными массивами**

Для матрицы 4 × 3 рассмотрим поэлементные операции сложения и произведения её элементов:

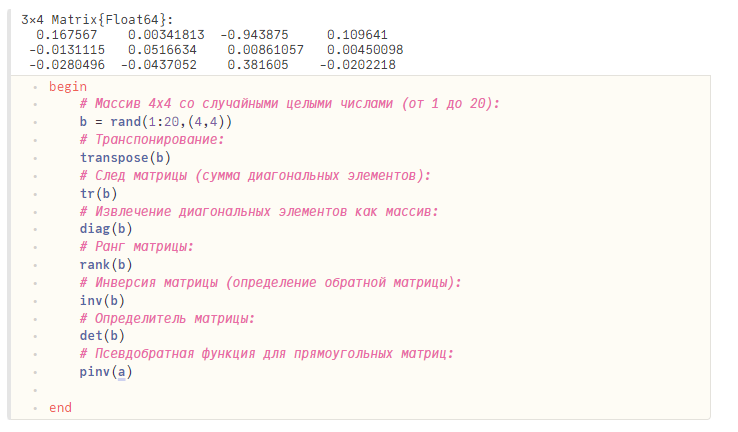


Для работы со средними значениями можно воспользоваться возможностями пакета Statistics:



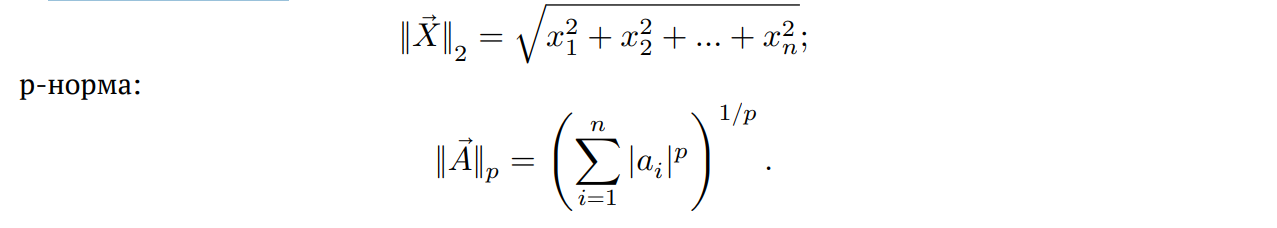
* 1. **Транспонирование, след, ранг, определитель и инверсия матрицы**

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



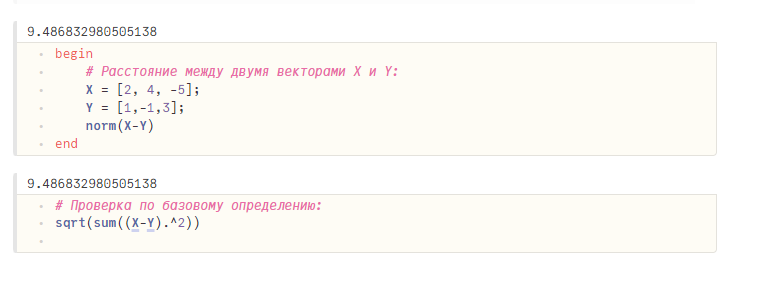
* 1. **Вычисление нормы векторов и матриц, повороты, вращения**

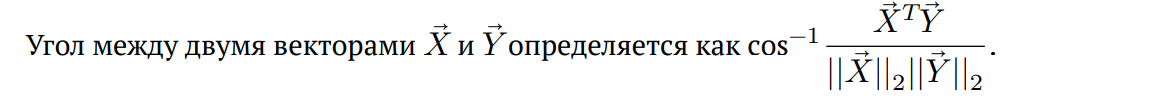
Для вычисления нормы используется LinearAlgebra.norm(x). Евклидова норма:

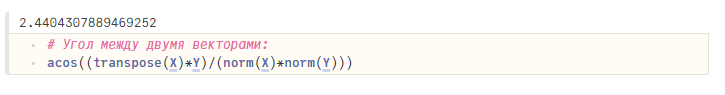








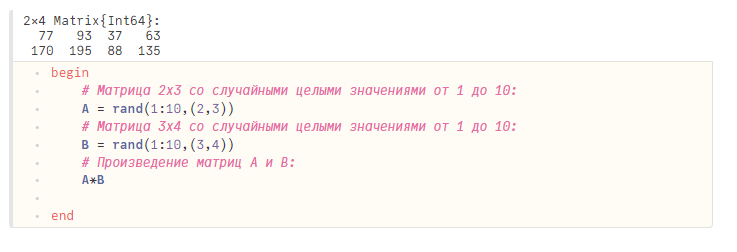


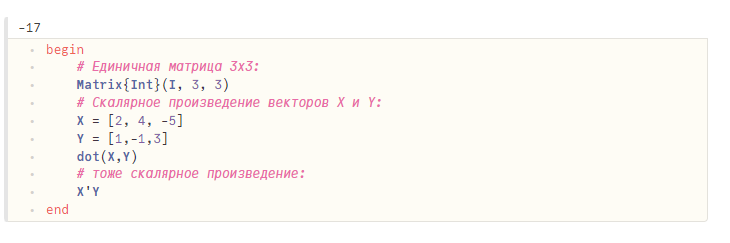


Вычисление нормы для двумерной матрицы:



* 1. **Матричное умножение, единичная матрица, скалярное произведение**





* 1. **Факторизация. Специальные матричные структуры**

В математике факторизация (или разложение) объекта — его декомпозиция (например, числа, полинома или матрицы) в произведение других объектов или факторов, которые, будучи перемноженными, дают исходный объект.

Матрица может быть факторизована на произведение матриц специального вида

для приложений, в которых эта форма удобна. К специальным видам матриц относят

ортогональные, унитарные и треугольные матрицы.

LU-разложение — представление матрицы 𝐴 в виде произведения двух матриц 𝐿 и 𝑈,

𝐿 — нижняя треугольная матрица, а 𝑈 — верхняя треугольная матрица. LU-разложение

существует только в том случае, когда матрица 𝐴 обратима, а все её ведущие (угловые)

главные миноры невырождены.

Обращение матрицы 𝐴 эквивалентно решению линейной системы 𝐴𝑋 = 𝐼, где 𝑋

— неизвестная матрица, 𝐼 — единичная матрица. Решение 𝑋 этой системы является

обратной матрицей 𝐴−1

.

LUP-разложение — представление матрицы 𝐴 в виде произведения 𝑃 𝐴 = 𝐿𝑈, где

матрица 𝐿 является нижнетреугольной с единицами на главной диагонали, 𝑈 — верхнетреугольная общего вида матрица,𝑃 — матрица перестановок, получаемая из единичной

матрицы путём перестановки строк или столбцов.

QR-разложение матрицы — представление матрицы в виде произведения унитарной (или ортогональной) матрицы 𝑄 и верхнетреугольной матрицы 𝑅. QR-разложение

применяется для нахождения собственных векторов и собственных значений матрицы.

𝑄 является ортогональной матрицей, если 𝑄𝑇𝑄 = 𝐼, где 𝐼 — единичная матрица.

Спектральное разложение матрицы 𝐴 — представление её в виде произведения

𝐴 = 𝑉 Λ𝑉 −1, где 𝑉 — матрица, столбцы которой являются собственными векторами

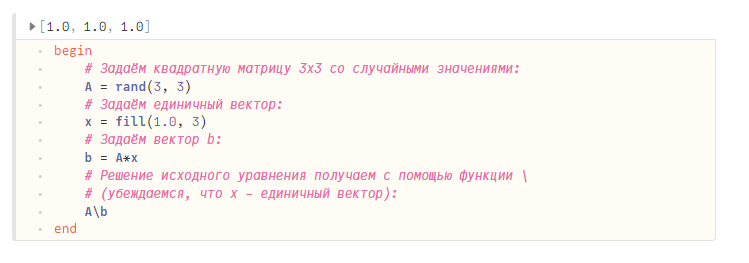
матрицы 𝐴, Λ — диагональная матрица с соответствующими собственными значениями

на главной диагонали, 𝑉

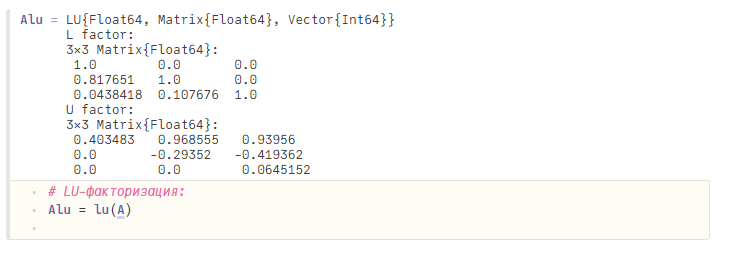
−1 — матрица, обратная матрице 𝑉.

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

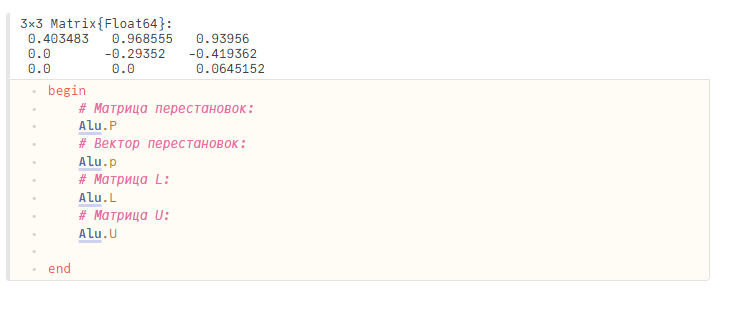
Решение систем линейный алгебраических уравнений 𝐴𝑥 = 𝑏:



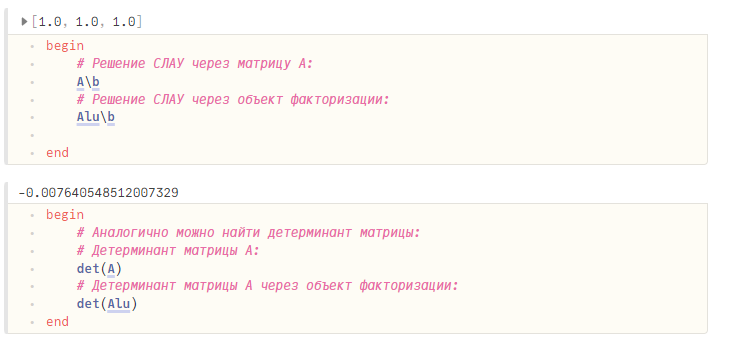
Julia позволяет вычислять LU-факторизацию и определяет составной тип факторизации для его хранения:



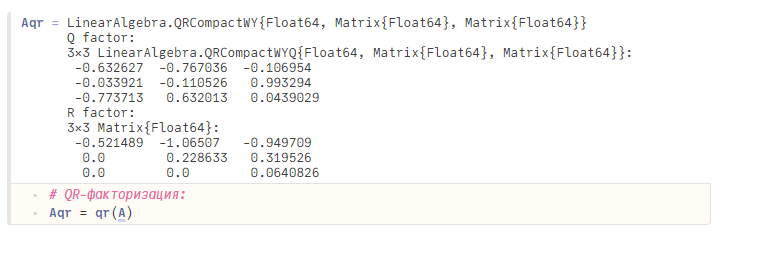
Различные части факторизации могут быть извлечены путём доступа к их специальным свойствам:

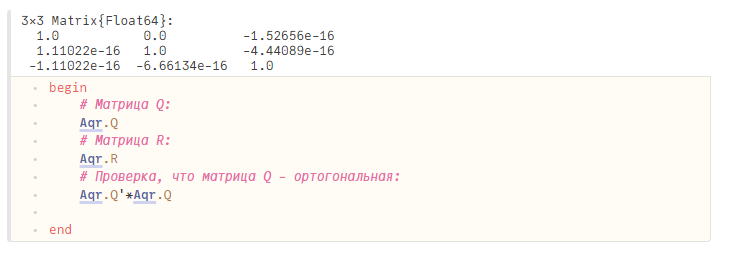


Исходная система уравнений 𝐴𝑥 = 𝑏 может быть решена или с использованием исходной матрицы, или с использованием объекта факторизации:

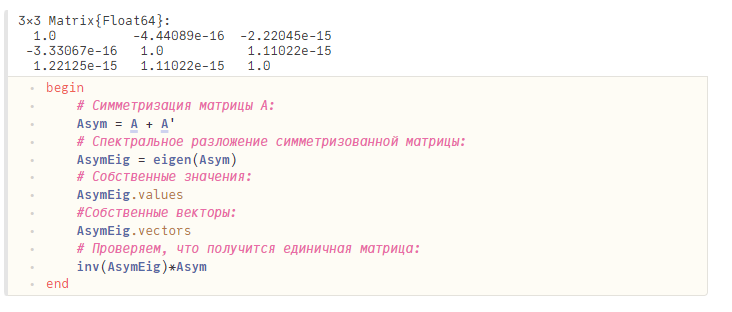


Julia позволяет вычислять QR-факторизацию и определяет составной тип факторизации для его хранения:

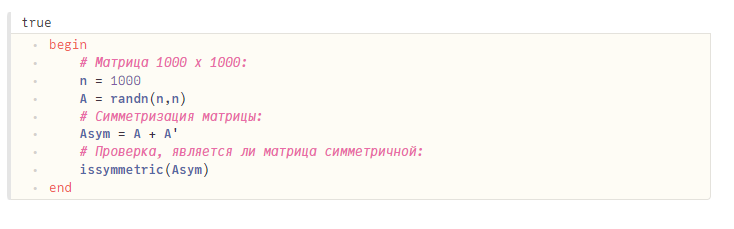


По аналогии с LU-факторизацией различные части QR-факторизации могут быть извлечены путём доступа к их специальным свойствам:

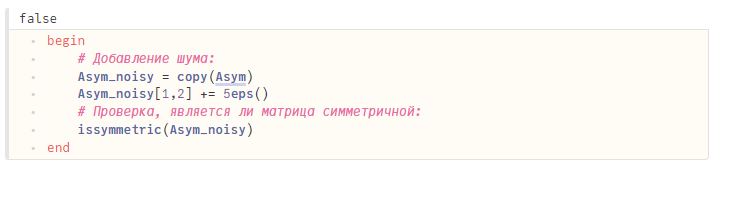
Примеры собственной декомпозиции матрицы 𝐴:



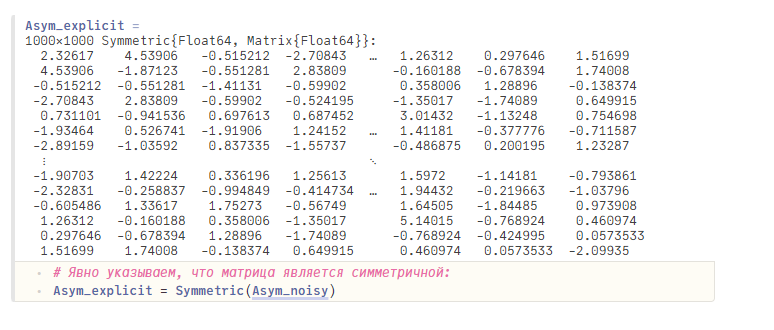
Далее рассмотрим примеры работы с матрицами большой размерности и специальной структуры.



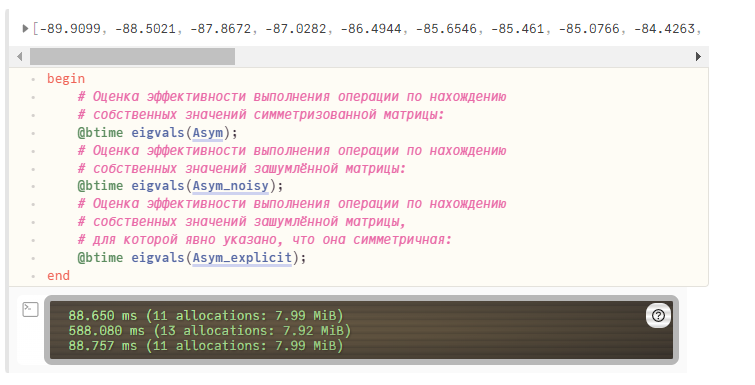
Пример добавления шума в симметричную матрицу (матрица уже не будет симметричной):



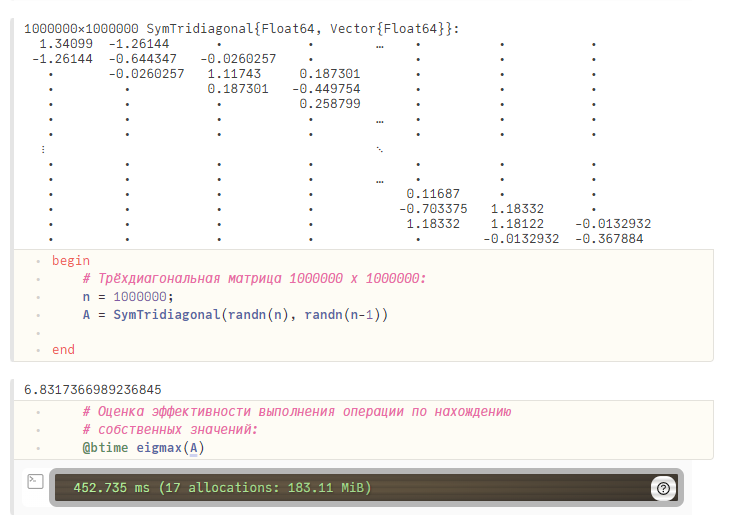
В Julia можно объявить структуру матрица явно, например, используя Diagonal, Triangular, Symmetric, Hermitian, Tridiagonal и SymTridiagonal:



Далее для оценки эффективности выполнения операций над матрицами большой размерности и специальной структуры воспользуемся пакетом 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:



При попытке задать подобную матрицу обычным способом и посчитать её собственные

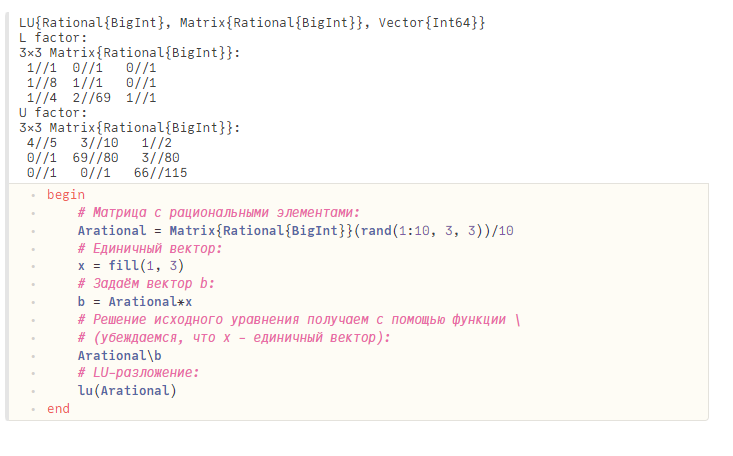
значения, вы скорее всего получите ошибку переполнения памяти:

B = Matrix(A)

OutOfMemoryError()

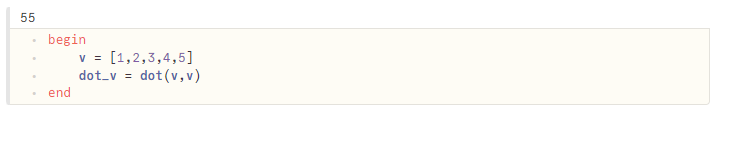
* 1. **Общая линейная алгебра**

Обычный способ добавить поддержку числовой линейной алгебры - это обернуть подпрограммы BLAS и LAPACK. Собственно, для матриц с элементами Float32,Float64, Complex {Float32} или Complex {Float64} разработчики Julia использовали такое же решение. Однако Julia также поддерживает общую линейную алгебру, что позволяет, например, работать с матрицами и векторами рациональных чисел. Для задания рационального числа используется двойная косая черта: 1//2 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):

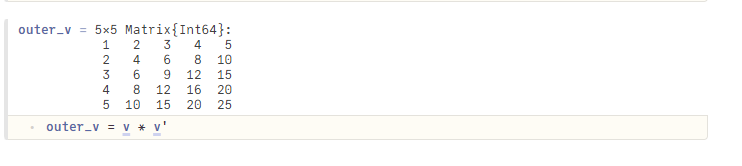


1. **Задания для самостоятельной работы**
   1. **Произведение векторов**

Задайте вектор v. Умножьте вектор v скалярно сам на себя и сохраните результат в dot\_v.



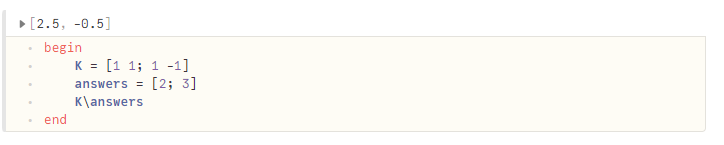
Умножьте v матрично на себя (внешнее произведение), присвоив результат переменной outer\_v.

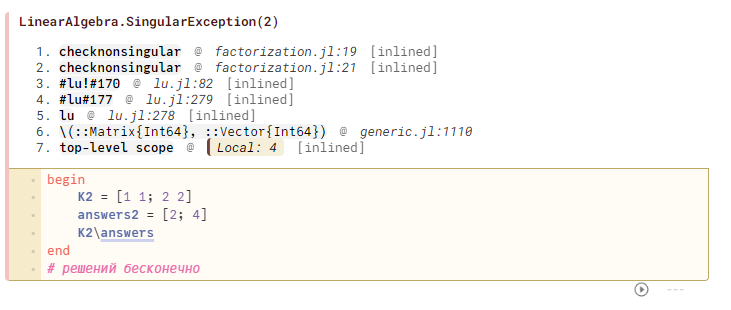


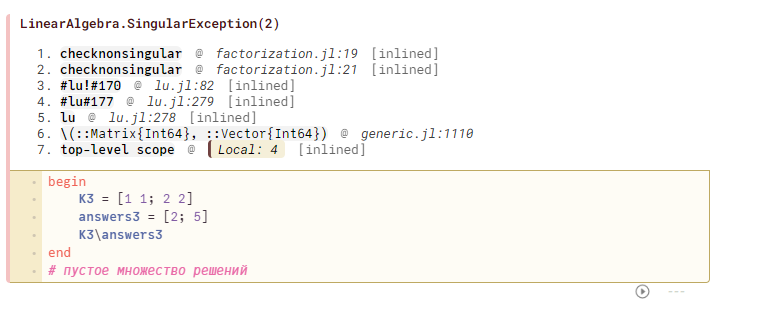
* 1. **Системы линейных уравнений**

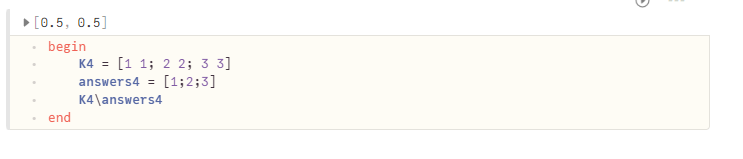
Решить СЛАУ с двумя неизвестными.

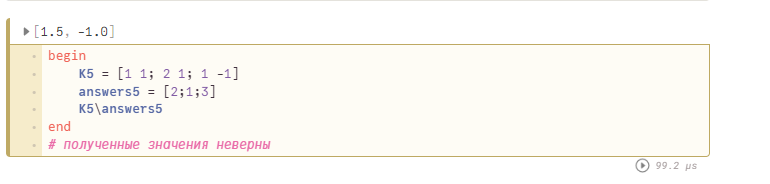


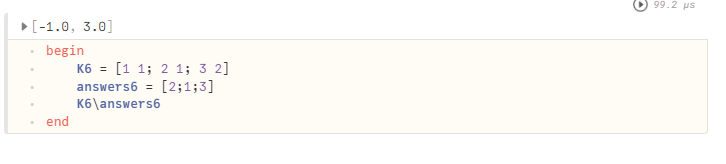






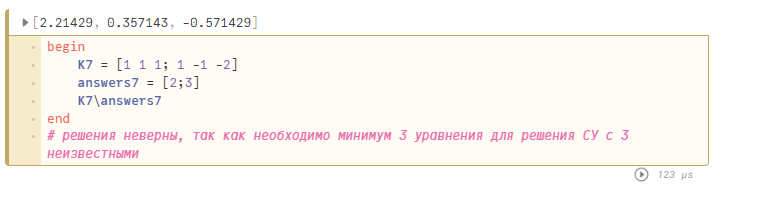


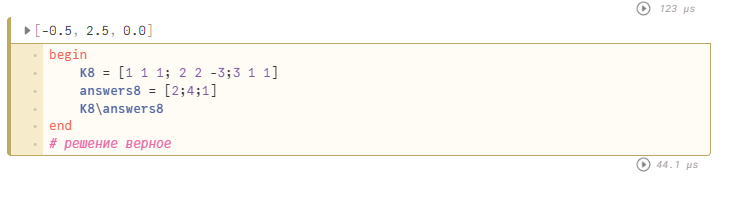


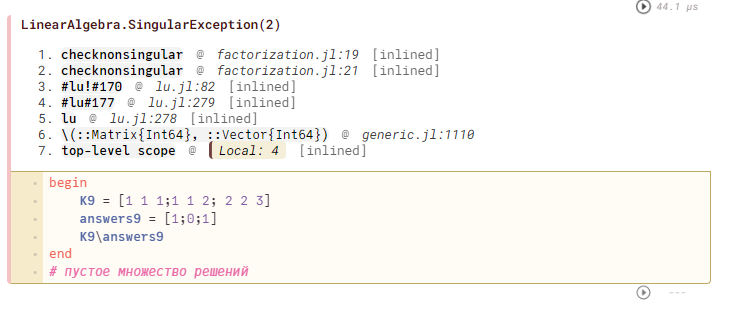


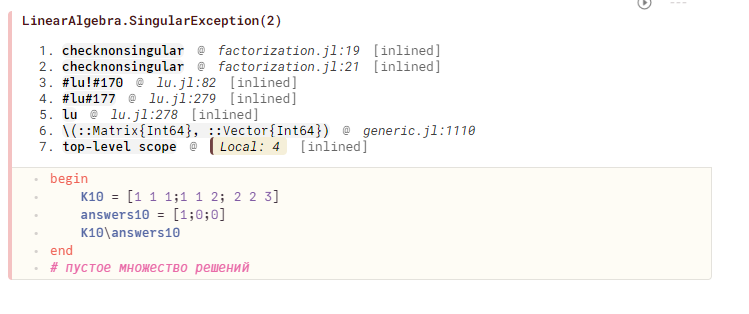
Решить СЛАУ с тремя неизвестными











* 1. **Операции с матрицами**

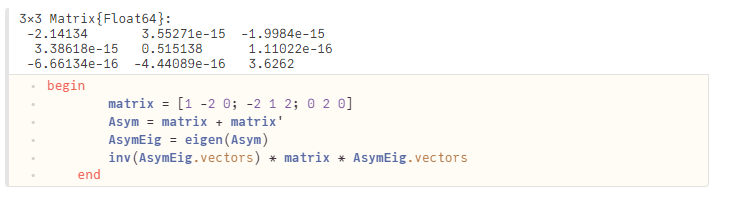
Приведите приведённые ниже матрицы к диагональному виду



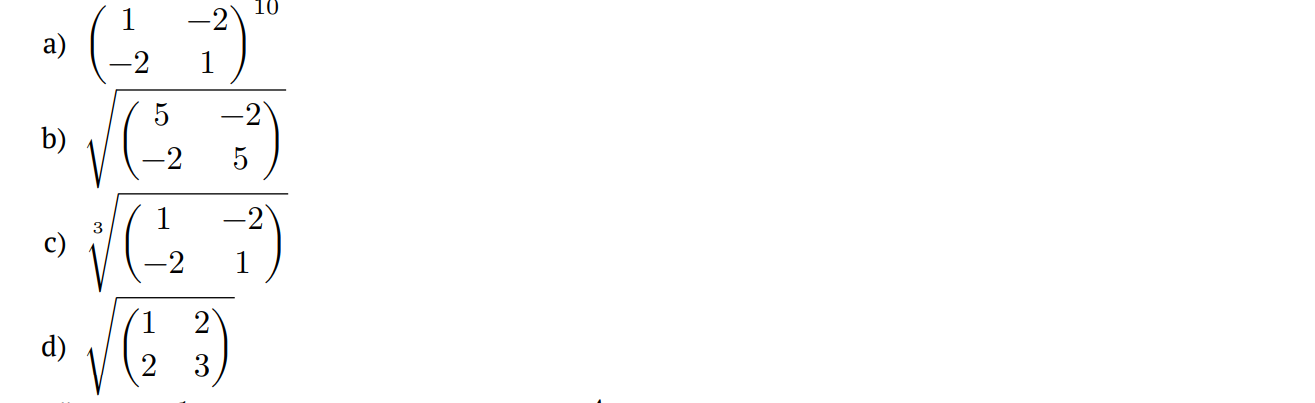


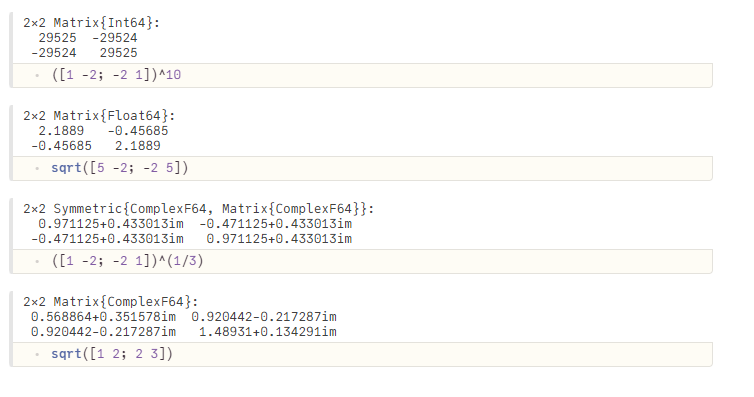




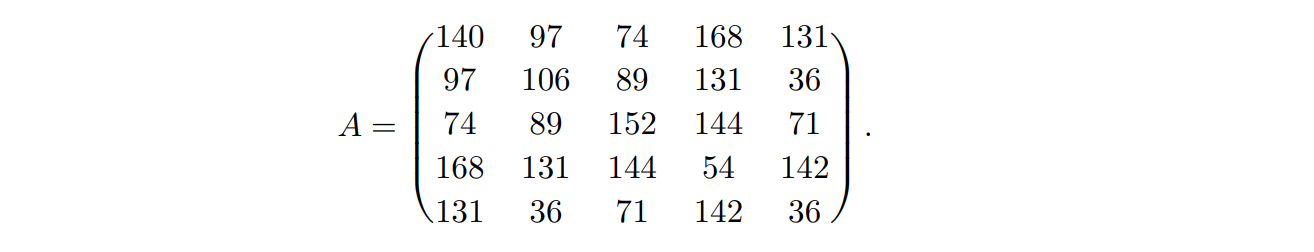


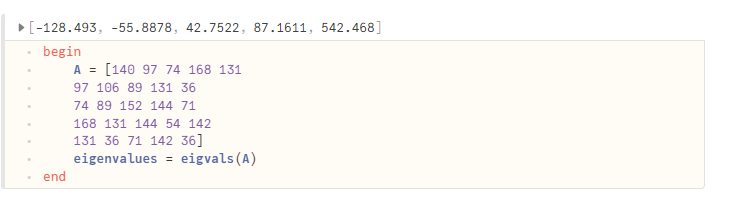
Вычислите



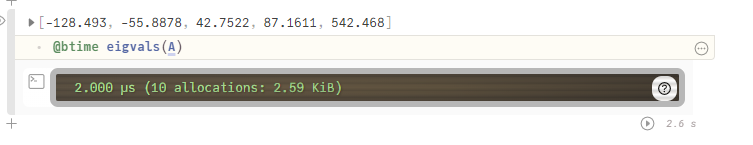


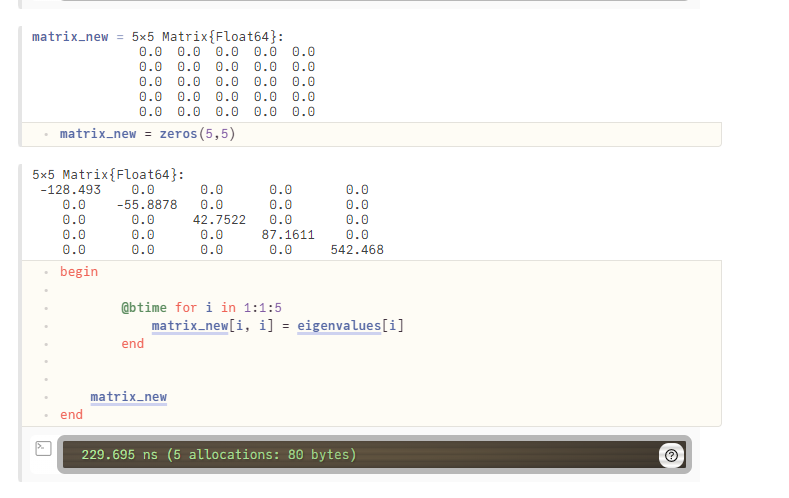
* 1. Найдите собственные значения матрицы 𝐴, если

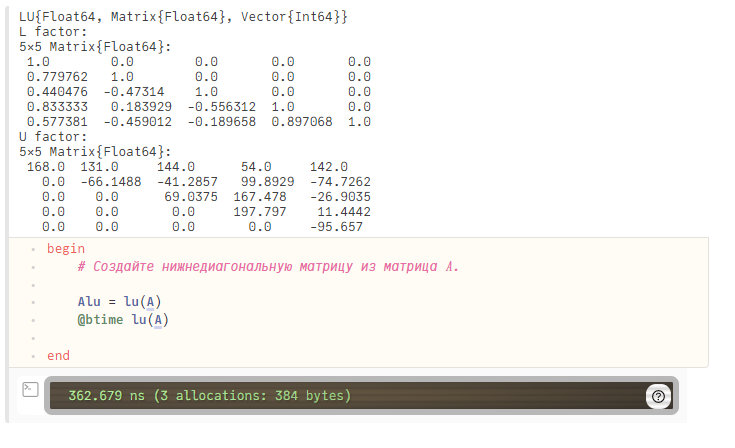




Создайте диагональную матрицу из собственных значений матрицы 𝐴. Создайте нижнедиагональную матрицу из матрица 𝐴. Оцените эффективность выполняемых операций







2.5 Линейные модели экономики

Линейная модель экономики может быть записана как СЛАУ

𝑥 − 𝐴𝑥 = 𝑦,

где элементы матрицы 𝐴 и столбца 𝑦 — неотрицательные числа. По своему смыслу в экономике элементы матрицы 𝐴 и столбцов 𝑥, 𝑦 не могут быть отрицательными числами.

1. Матрица 𝐴 называется продуктивной, если решение 𝑥 системы при любой неотрицательной правой части 𝑦 имеет только неотрицательные элементы 𝑥𝑖. Используя это

определение, проверьте, являются ли матрицы продуктивными.





Критерий продуктивности: матрица 𝐴 является продуктивной тогда и только тогда,

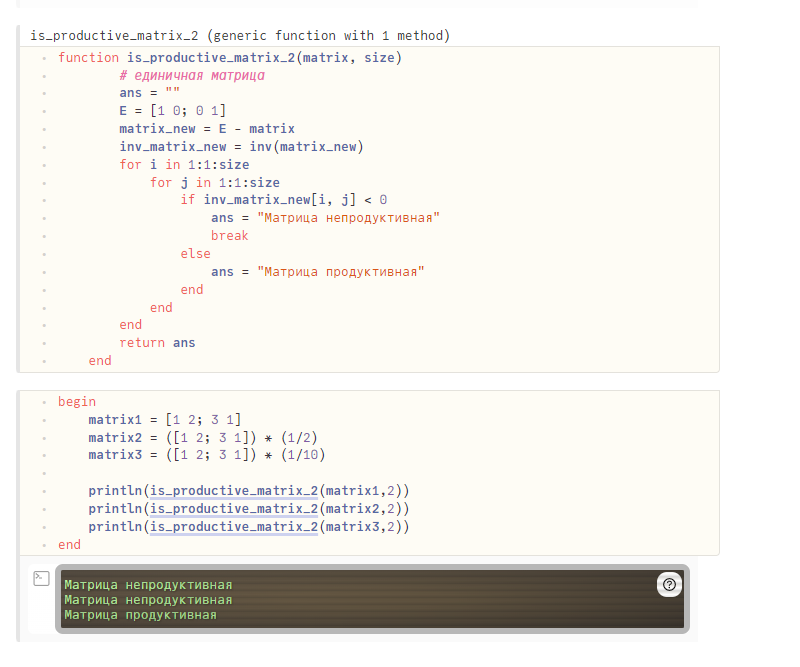
когда все элементы матрица

(𝐸 − 𝐴)−1

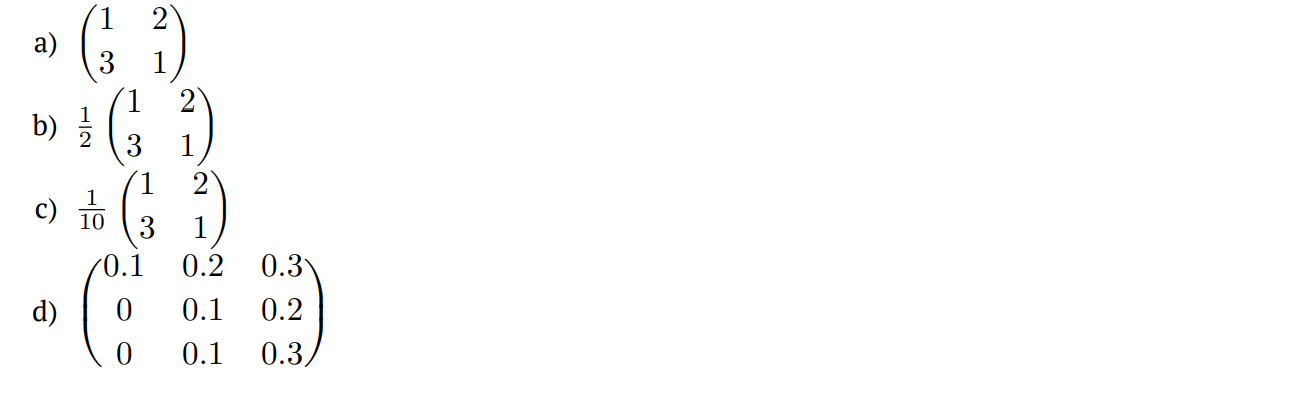
являются неотрицательными числами. Используя этот критерий, проверьте, являются

ли матрицы продуктивными.





Спектральный критерий продуктивности: матрица 𝐴 является продуктивной тогда и только тогда, когда все её собственные значения по модулю меньше 1. Используя этот критерий, проверьте, являются ли матрицы продуктивными.





**Выводы**

В ходе выполнения лабораторной работы успешно удалось изучить возможности специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.