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: Яссин Мохамад Аламин

Группа: НКНбд-01-20

**МОСКВА**

2023 г.

# Постановка задачи

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

алгебры.

1. Используя Jupyter Lab, повторите примеры

2. Выполните задания для самостоятельной работы.

Выполнение работы

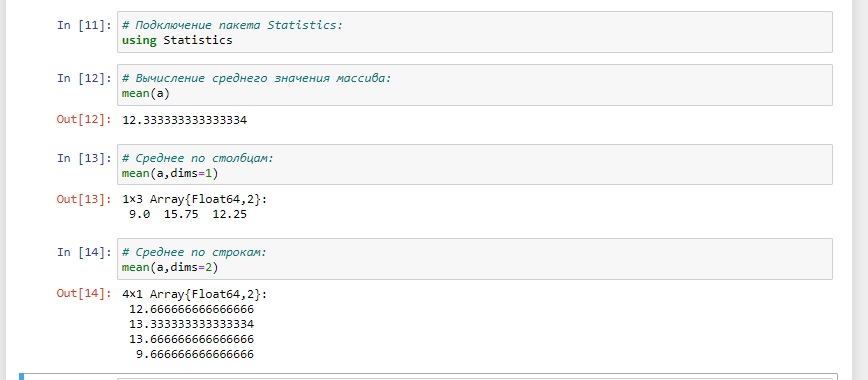
1. Повторение примером

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



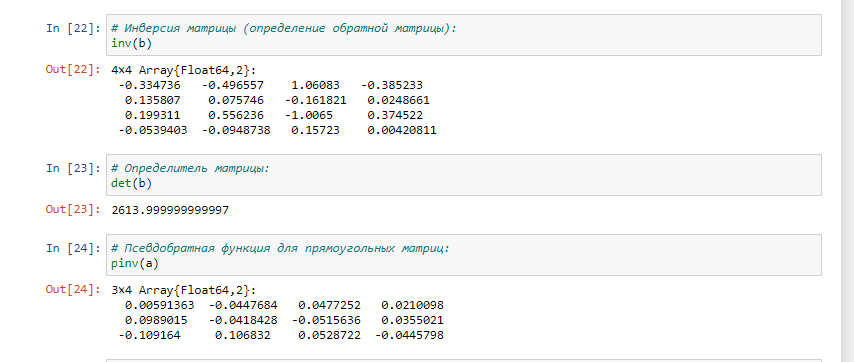
Для работы со средними значениями можно воспользоваться возможностями пакета

Statistics:



4.2.2. 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:



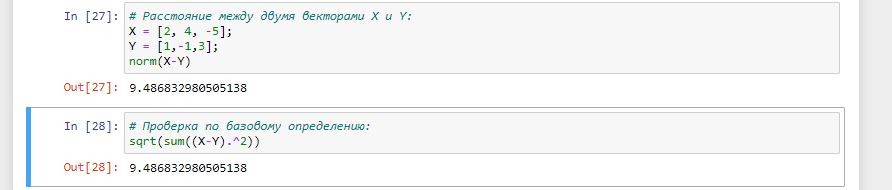
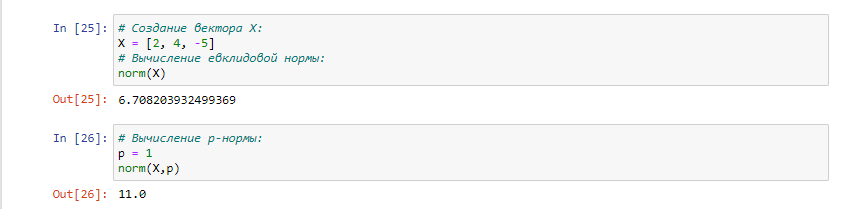
4.2.3. Вычисление нормы векторов и матриц, повороты, вращения

Для вычисления нормы используется LinearAlgebra.norm(x).

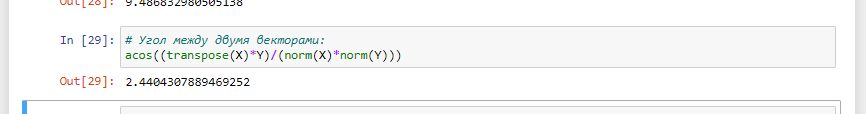
Евклидова норма:

p-норма:

Евклидово расстояние между двумя векторами и определяется как



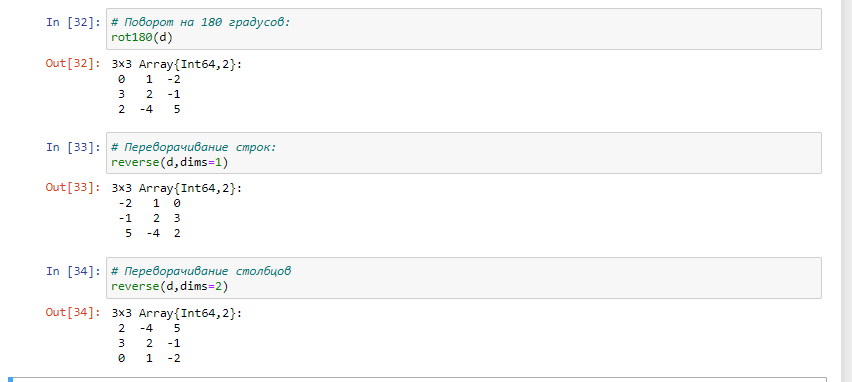
Угол между двумя векторами и определяется как



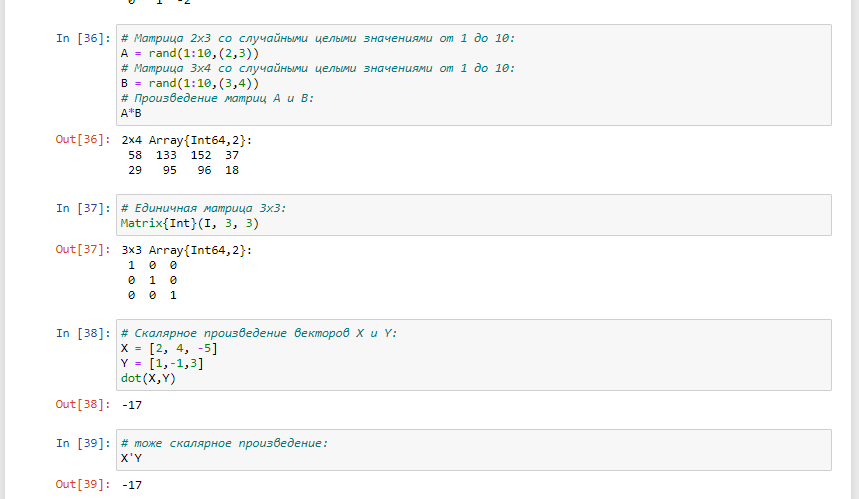
Вычисление нормы для двумерной матрицы:



Операции поворота и перестановки



4.2.4. Матричное умножение, единичная матрица, скалярное произведение



4.2.5. 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.

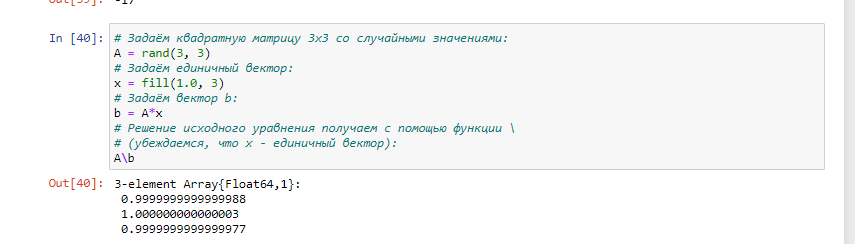
Матрица может быть факторизована на произведение матриц специального вида

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

ортогональные, унитарные и треугольные матрицы.

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

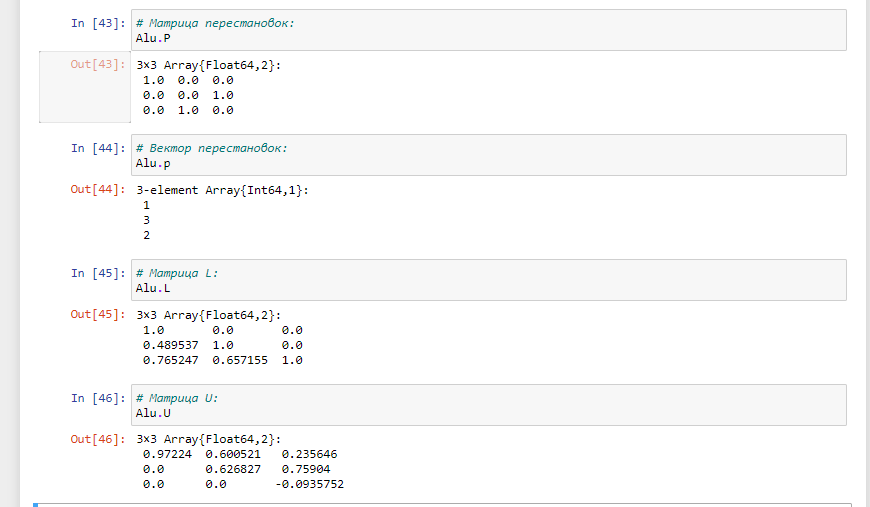
Решение систем линейный алгебраических уравнений 𝐴𝑥 = 𝑏:



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

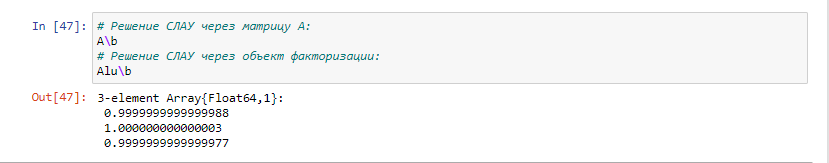


Различные части факторизации могут быть извлечены путём доступа к их специальным свойствам:

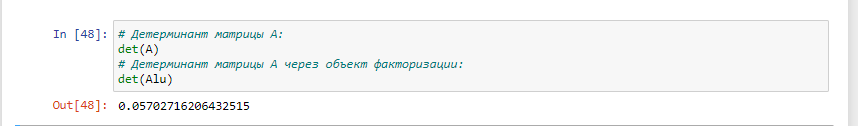


Исходная система уравнений 𝐴𝑥 = 𝑏 может быть решена или с использованием

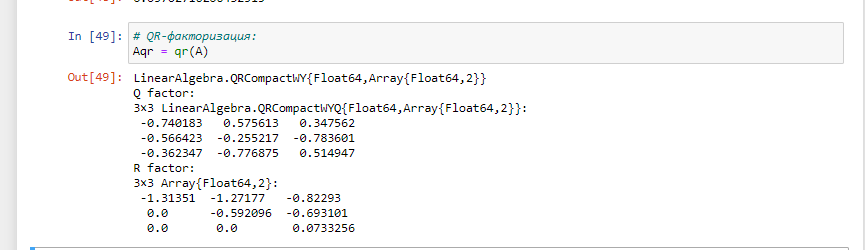
исходной матрицы, или с использованием объекта факторизации:



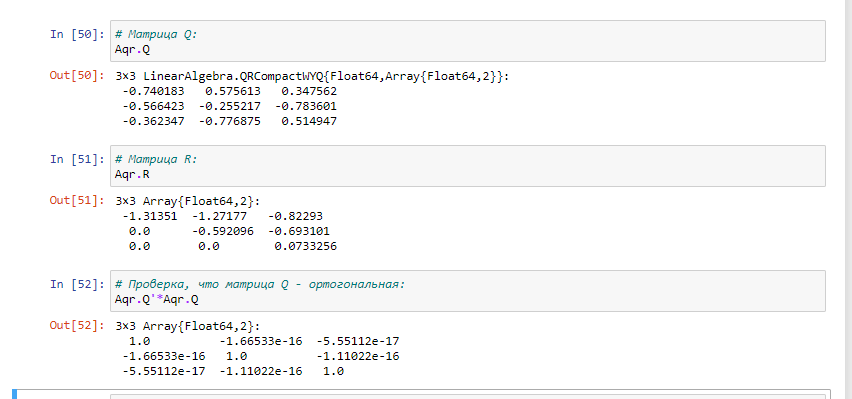
Аналогично можно найти детерминант матрицы:



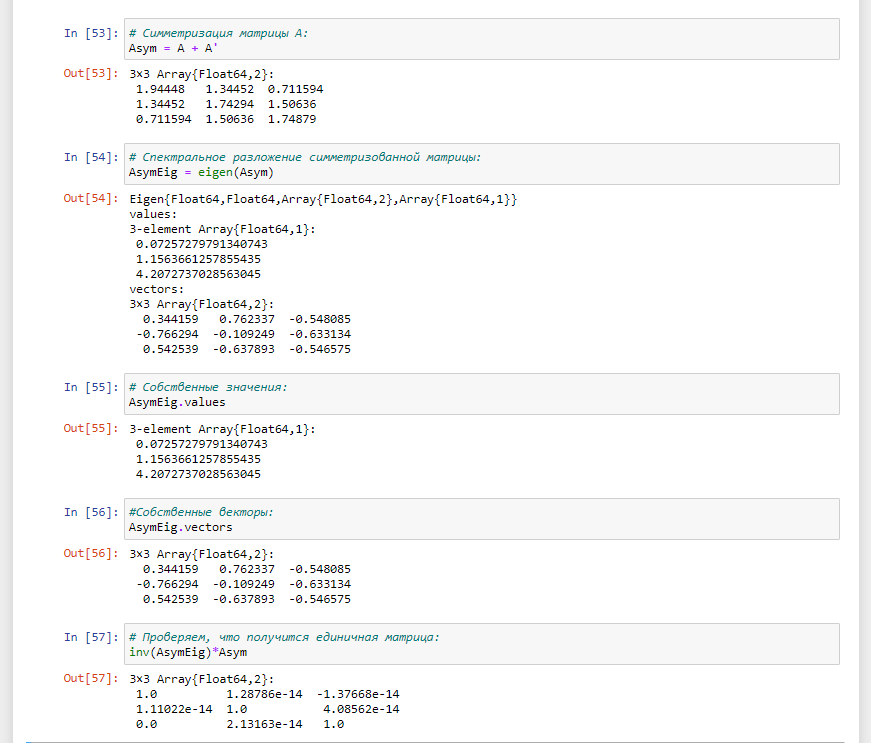
Julia позволяет вычислять QR-факторизацию и определяет составной тип факторизации для его хранения:



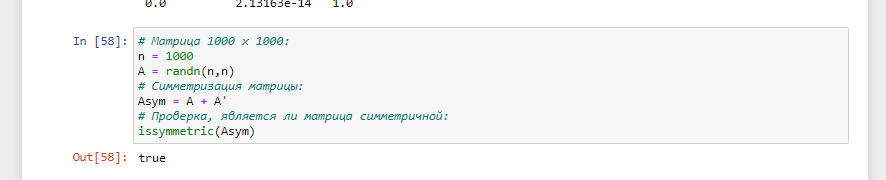
По аналогии с LU-факторизацией различные части QR-факторизации могут быть извлечены путём доступа к их специальным свойствам:



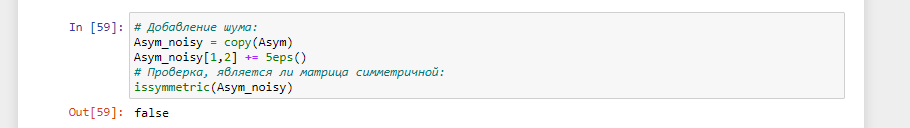
Примеры собственной декомпозиции матрицы 𝐴:



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

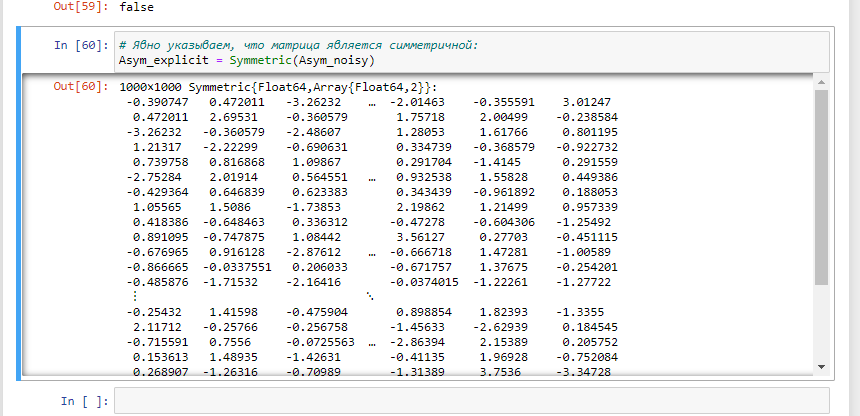


Пример добавления шума в симметричную матрицу (матрица уже не будет симметричной):



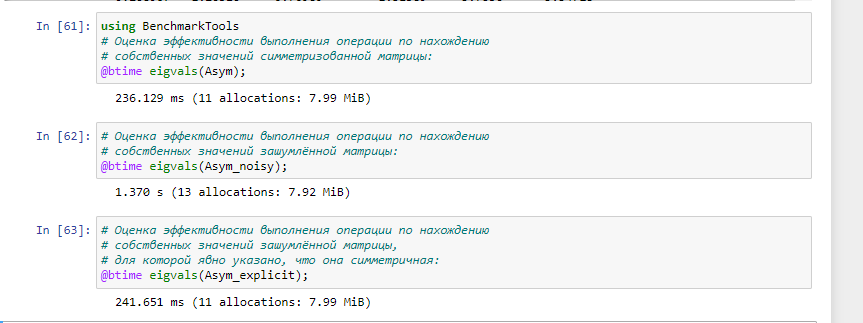
В Julia можно объявить структуру матрица явно, например, используя Diagonal,

Triangular, Symmetric, Hermitian, Tridiagonal и SymTridiagonal:



Далее для оценки эффективности выполнения операций над матрицами большой

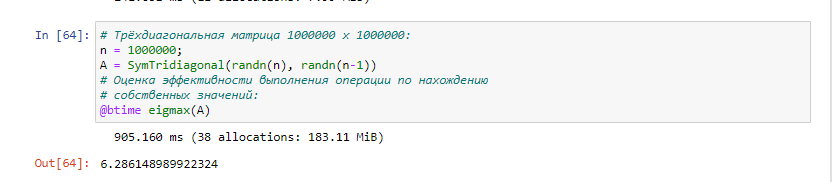
размерности и специальной структуры воспользуемся пакетом BenchmarkTools:



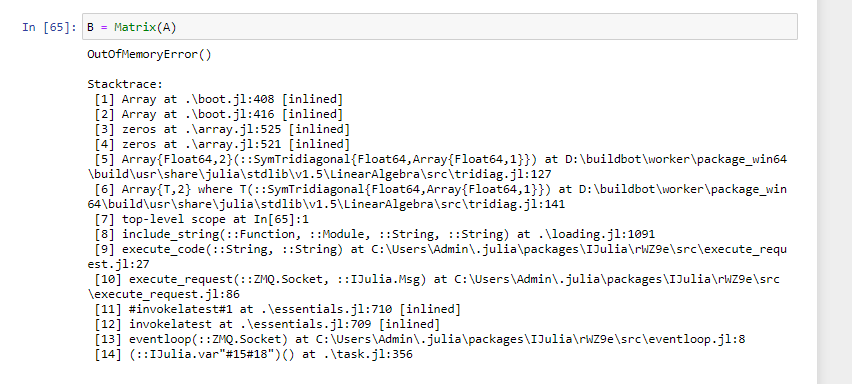
Далее рассмотрим примеры работы с разряженными матрицами большой размерности.

Использование типов Tridiagonal и SymTridiagonal для хранения трёхдиагональных матриц позволяет работать с потенциально очень большими трёхдиагональными

матрицами:



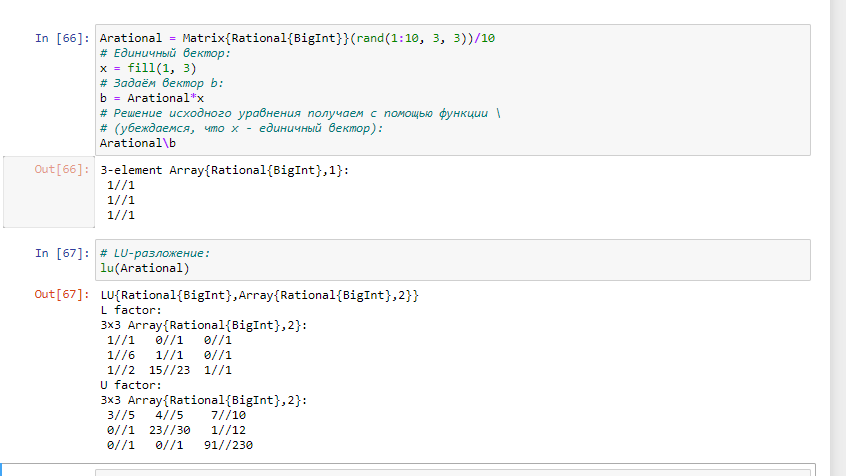
При попытке задать подобную матрицу обычным способом и посчитать её собственные значения, вы скорее всего получите ошибку переполнения памяти:



4.2.6. 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.

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):



4.4. Задания для самостоятельного выполнения

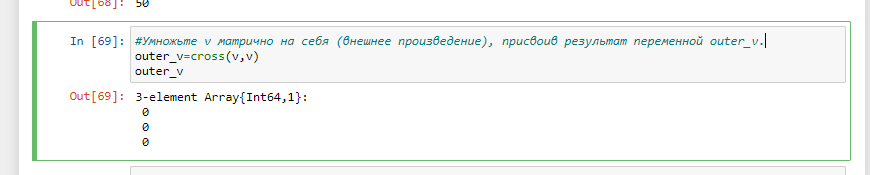
4.4.1. Произведение векторов

1. Задайте вектор v. Умножьте вектор v скалярно сам на себя и сохраните результат

в dot\_v.



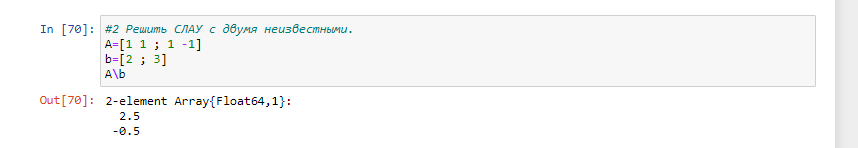
2. Умножьте v матрично на себя (внешнее произведение), присвоив результат переменной outer\_v.



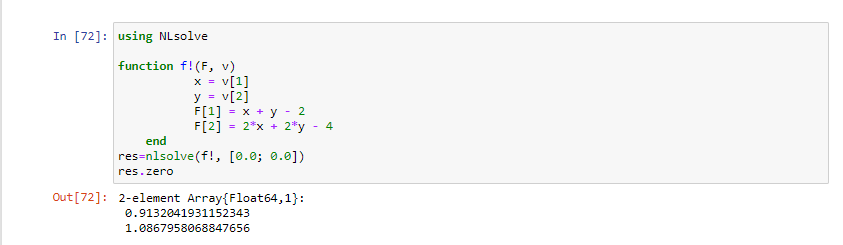
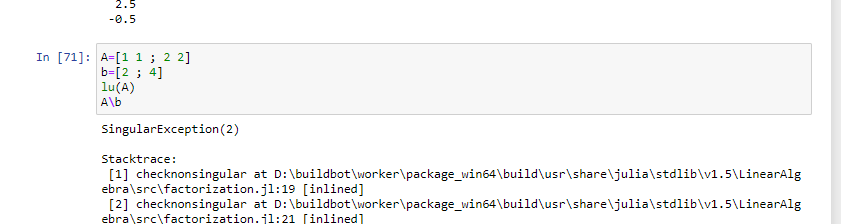
4.4.2. Системы линейных уравнений

1. Решить СЛАУ с двумя неизвестными.

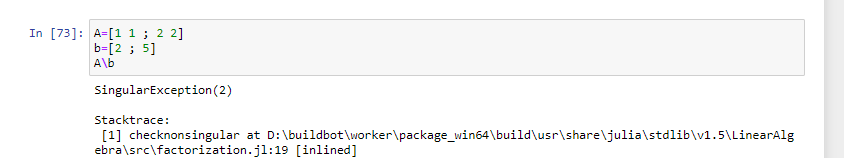
Системы решаются способом, указанном при повторении примеров (через матричное деление)



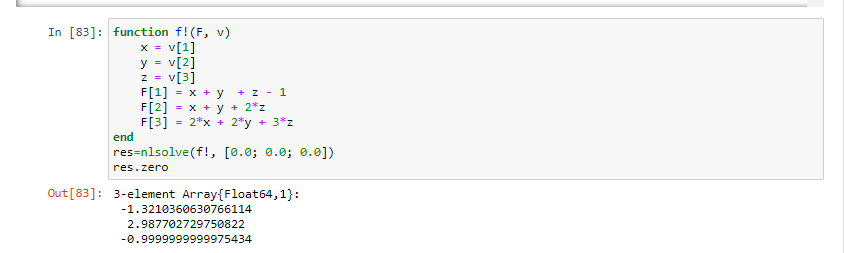
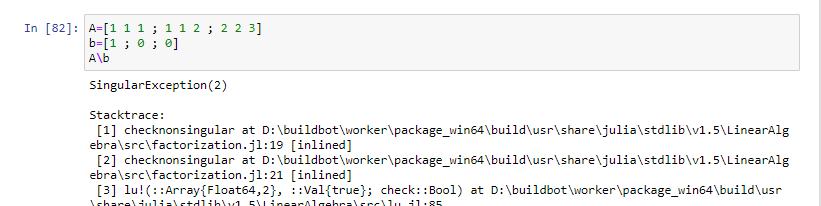
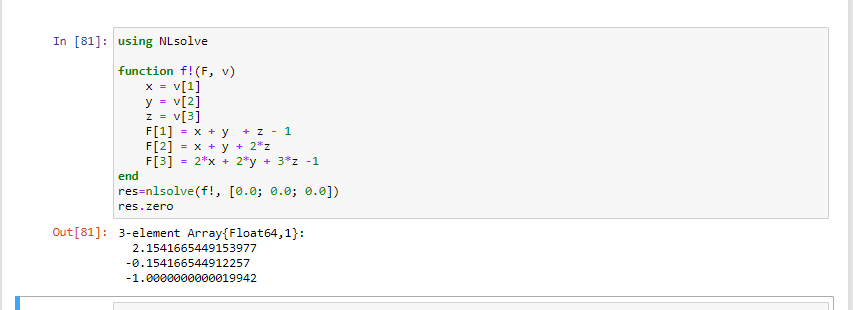
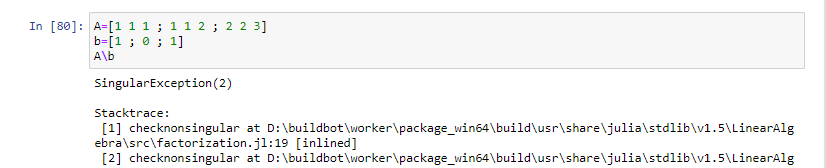
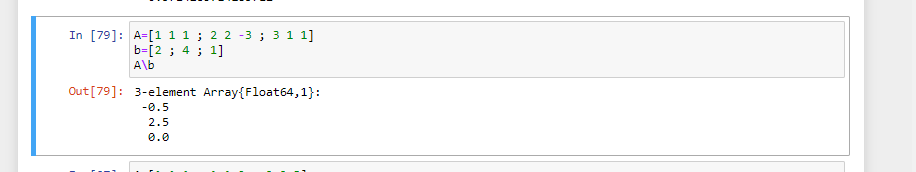
Данная система имеет линейно-зависимые строки, следовательно, бесконечное множество решений. Предыдущий способ решения результата не дает, обращаемся к библиотеке для решения нелинейых уравнений и получаем частное решение



Аналогично предыдущей системе



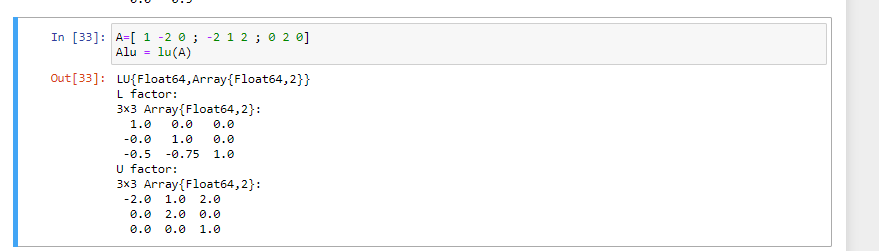
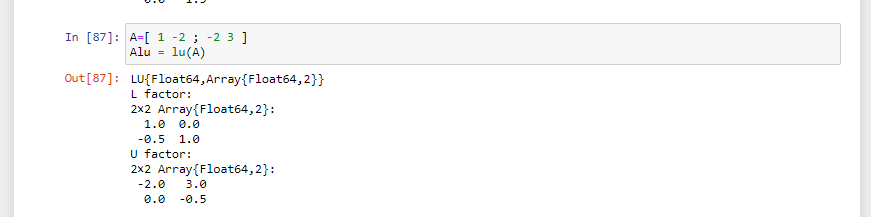
1. Решить СЛАУ с тремя неизвестными.



4.4.3. Операции с матрицами

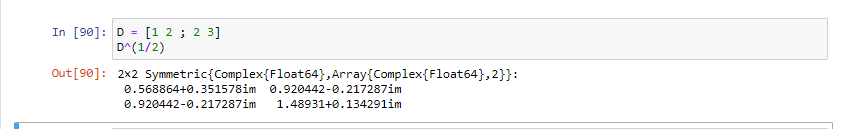
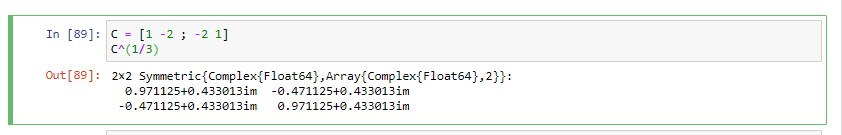
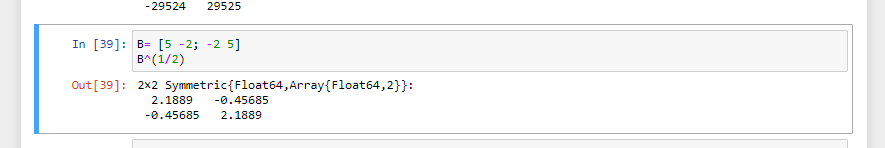
1. Приведите приведённые ниже матрицы к диагональному виду

Для приведения матриц к диагональному виду используем функцию LU разложения



2. Вычислите

Для вычислений применяем к матрицам операцию возведения в степень и извлечения корня



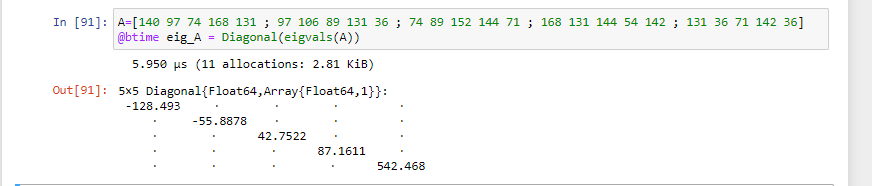
3 Найдите собственные значения матрицы 𝐴

Создайте диагональную матрицу из собственных значений матрицы 𝐴. Создайте

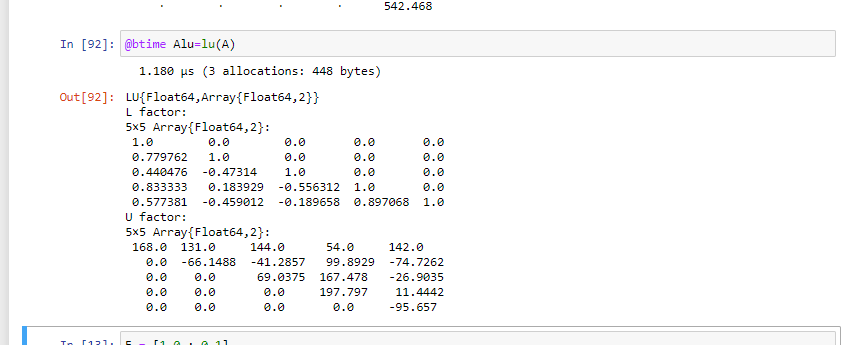
нижнедиагональную матрицу из матрица 𝐴. Оцените эффективность выполняемых

операций.

Для нахождения собственных значений используем функцию eigvals(), затем результат записываем в матрицу, указав что она диагональная. Вместе с этим проводим бенчмаркинг.



Выполняем LU разложение и получаем верхне- и нижне-диагональную матрицу



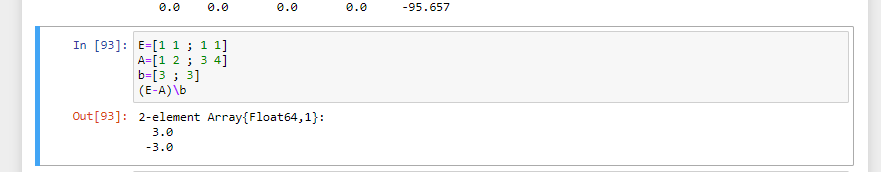
4.4.4. Линейные модели экономики

Линейная модель экономики может быть записана как СЛАУ 𝑥 − 𝐴𝑥 = 𝑦,

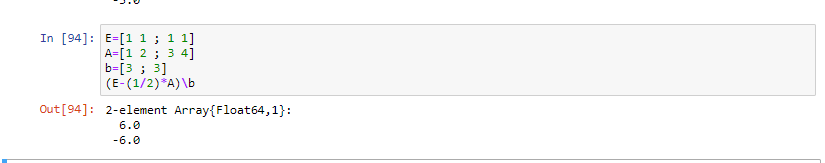
где элементы матрицы 𝐴 и столбца 𝑦 — неотрицательные числа.

По своему смыслу в экономике элементы матрицы 𝐴 и столбцов 𝑥, 𝑦 не могут быть отрицательными числами.

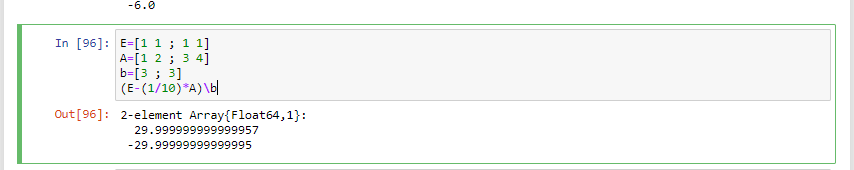
1. Матрица 𝐴 называется продуктивной, если решение 𝑥 системы при любой неотрицательной правой части 𝑦 имеет только неотрицательные элементы 𝑥 Используя это определение, проверьте, являются ли матрицы продуктивными.



Один из корней отрицательный – матрица не является продуктивной



Один из корней отрицательный – матрица не является продуктивной



Один из корней отрицательный – матрица не является продуктивной

2 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.



3 элемента отрицательны – матрица не является продуктивной



4 элемента отрицательны – матрица не является продуктивной



4 элемента положительны– матрица является продуктивной

3 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.



оба значения по модулю превосходят 1 – матрица не является продуктивной

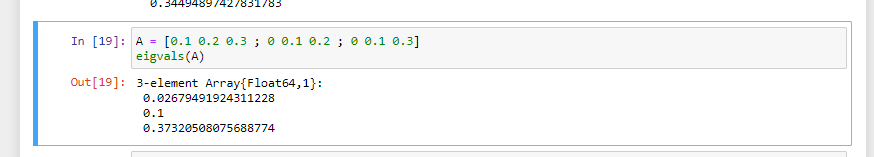


одно из значений по модулю превосходит 1 – матрица не является продуктивной



оба значения по модулю меньше 1 – матрица является продуктивной

все три значения по модулю меньше 1 – матрица является продуктивной



**Выводы**

Получены навыки работы с матричными функциями и операциями линейной алгебры