БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

ОТЧЕТ ПО МЕТОДАМ ВЫЧИСЛЕНИЙ

студента 2 курса 13 группы

Лабораторная работа №2

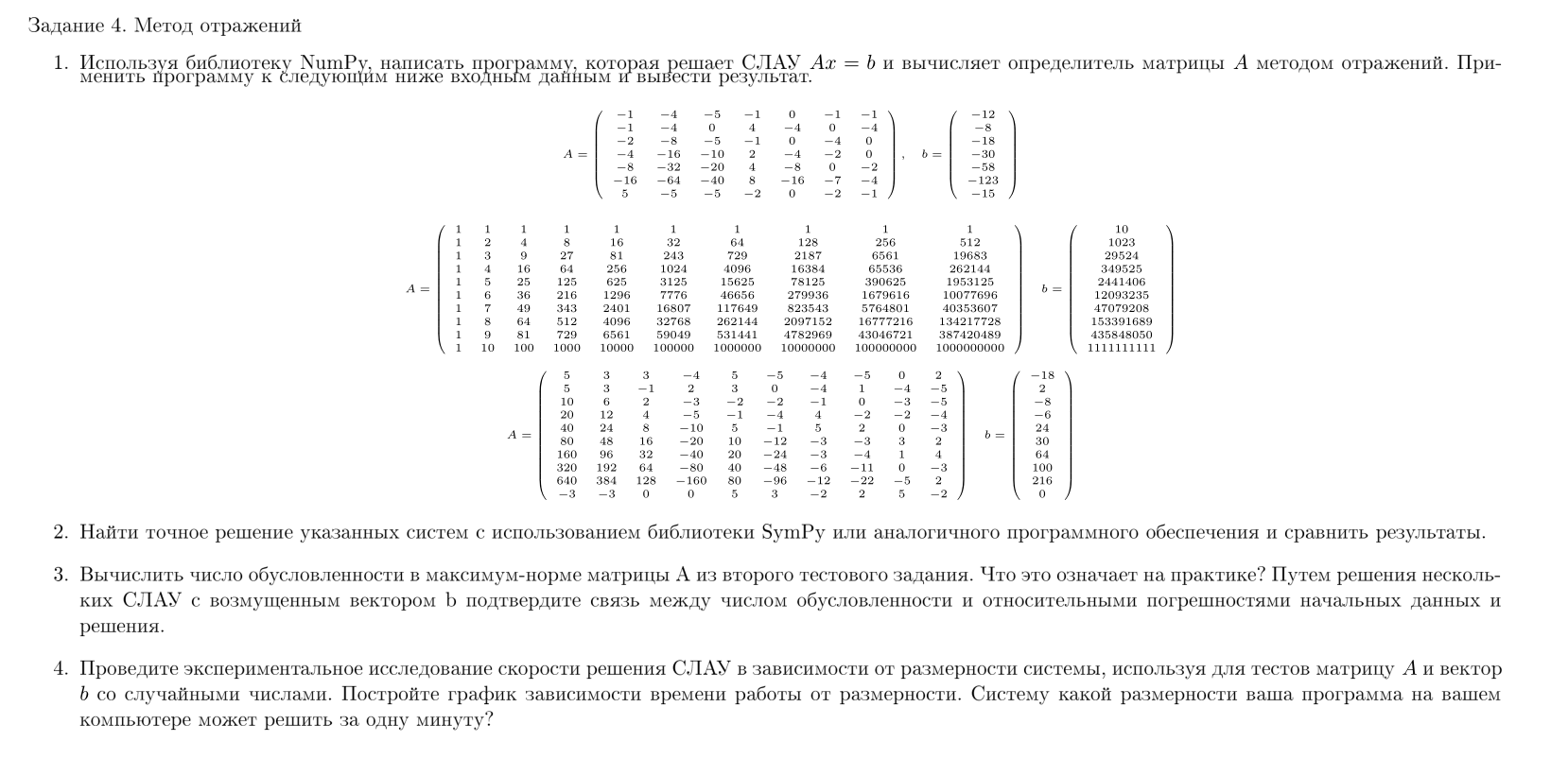
Преподаватель

Бондарь И.В.

Минск 2020

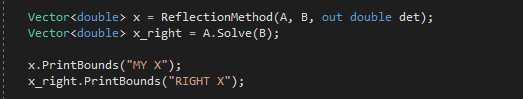
Вариант 4

**Условие**

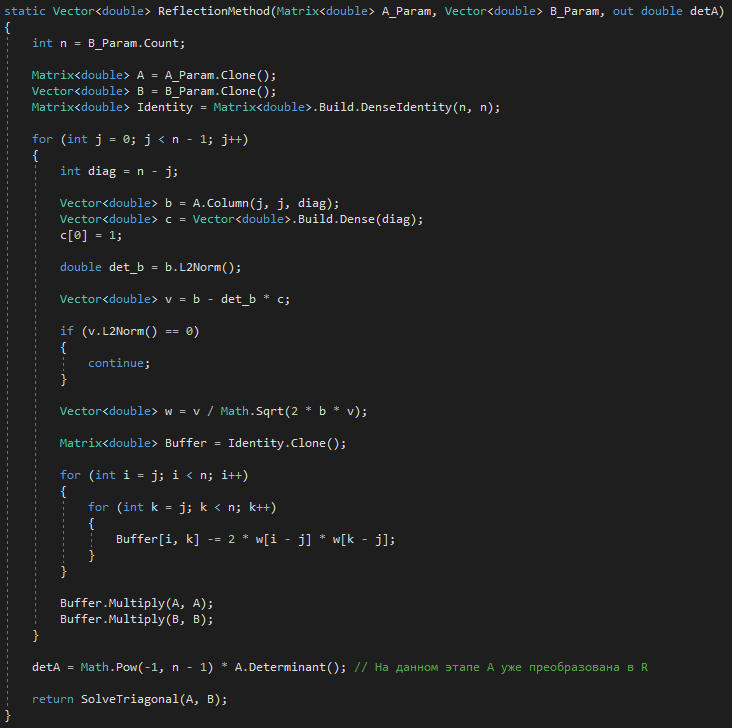


**Задание 1, 2**

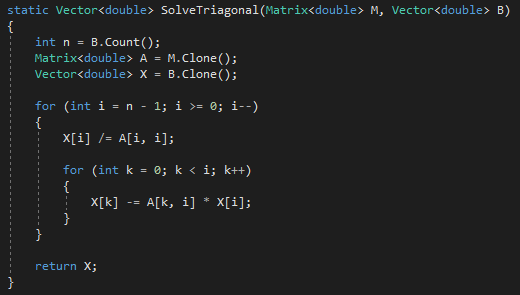
Я выполнял работу на C# и пользовался библиотекой MathNet. Точные результаты вычислял методами из этой же библиотеки:



Вот как выглядит мой метод:



Цель метода – привести матрицу к верхнетреугольной, постепенно зануляя столбцы под главной диагональю, после чего решить как треугольную:

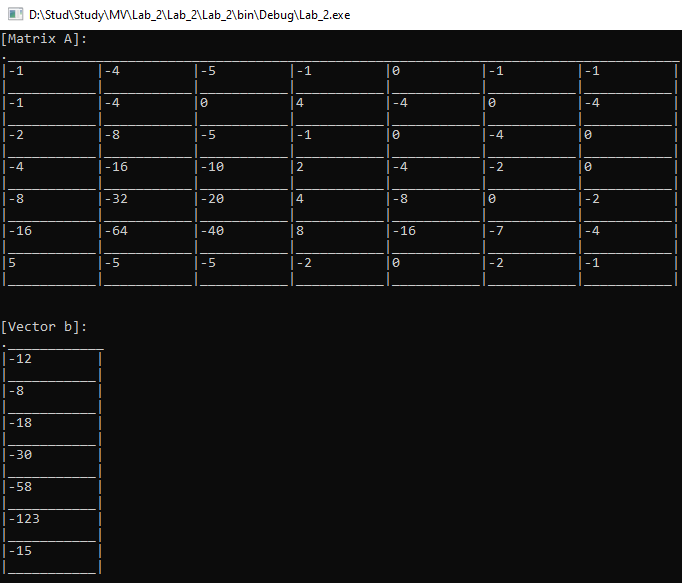


Для этого нужно вычислить ряд ортогональных матриц Q и применить к результату предыдущего преобразования матрицы A и вектора b. Сначала я хранил эти матрицы, но потом решил сразу выполнять преобразования чтобы не тратить место в пустую. Определитель в конце вычисляется по формуле:

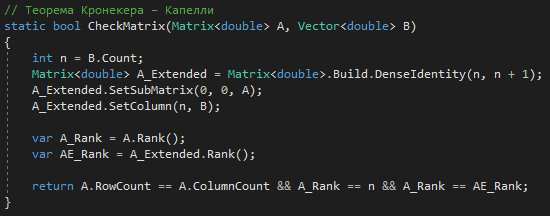


Хочу отметить, что реализация на самом деле не самая оптимальная, так как классы матриц и векторов в библиотеке, хоть и не иммутабельны, но довольно близки к этому. Поэтому многие операторы возвращают измененную копию объекта, выполняя при этом аллокации для массивов, что не очень хорошо, так как провоцирует сборку мусора и на больших матрицах это значительно снижает производительность. По-хорошему можно было обойтись всего несколькими матрицами и векторами для конкретной СЛАУ, оперируя исключительно индексами.

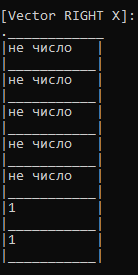
Результаты:

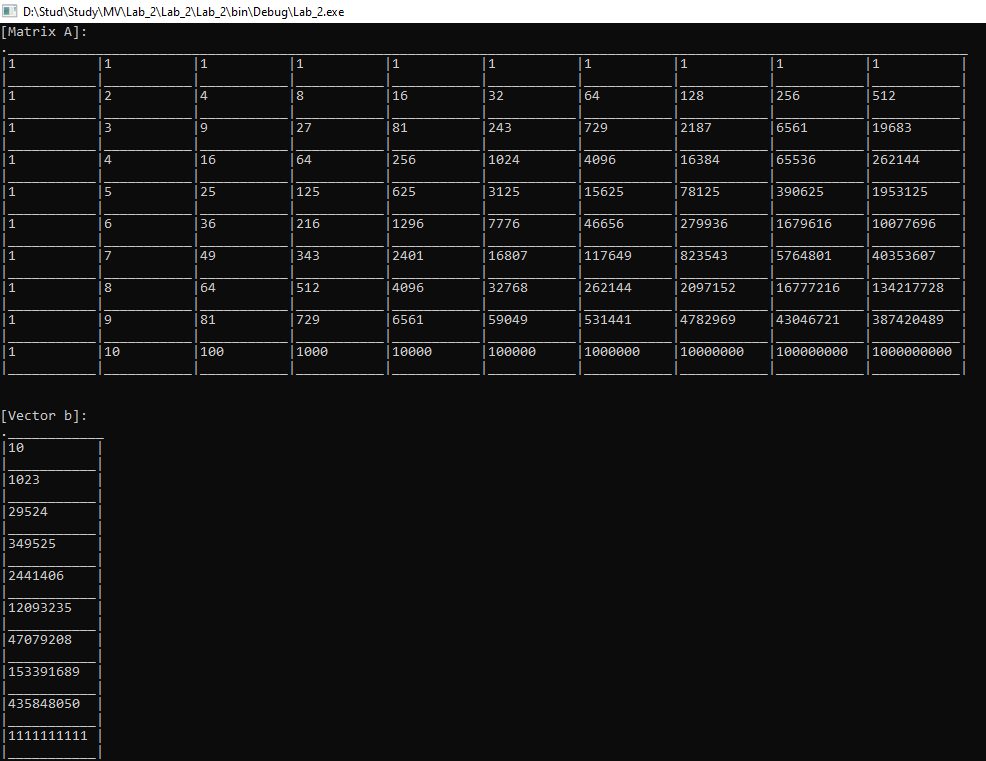
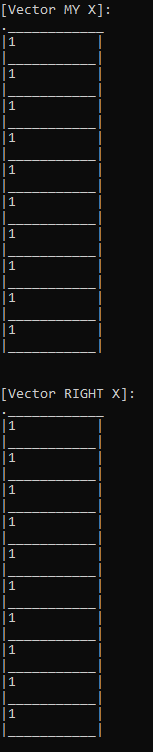


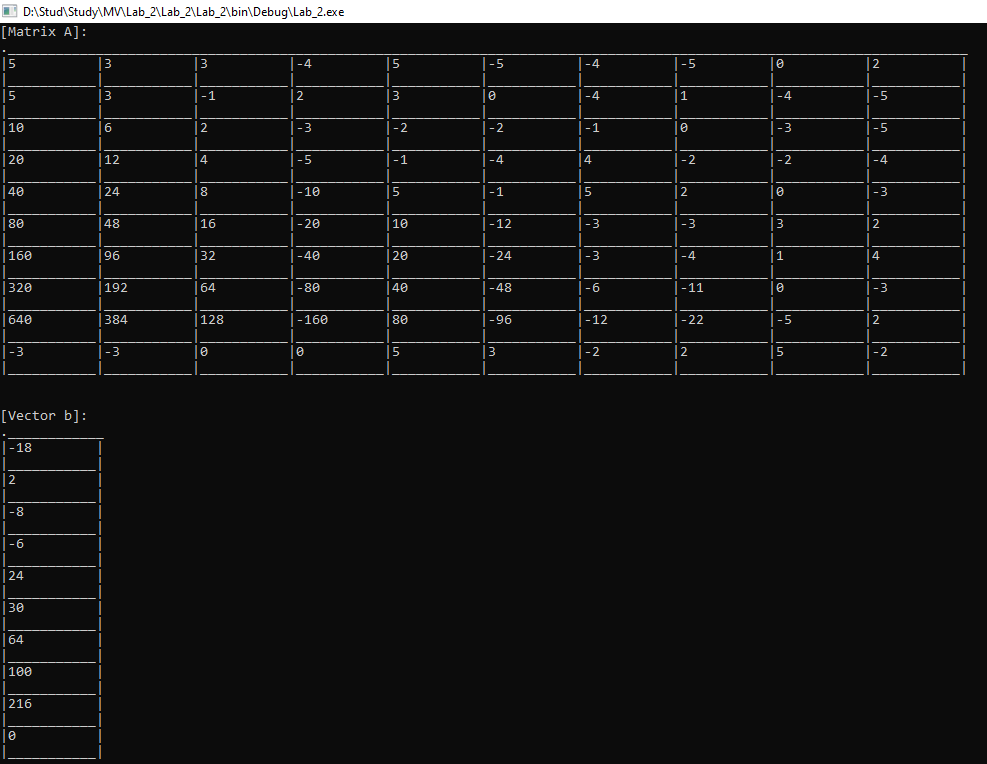
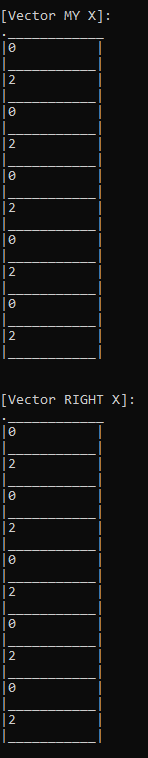
Первая матрица не прошла проверку на единственное решение, ранг матрицы оказался меньше числа переменных. Сама проверка:



Метод решения из библиотеки среагировал на это вернув вектор с NAN:

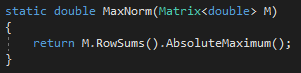


**Задание 3**

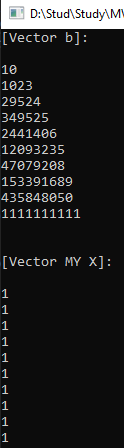
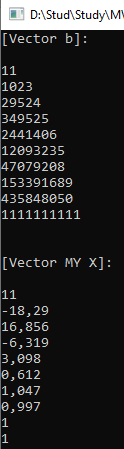
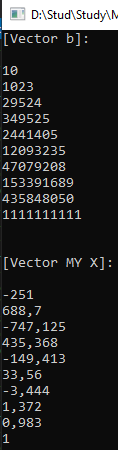
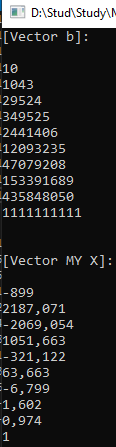
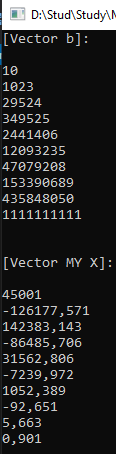
**Рассчитываю максимум-норму**



**Вычисляю число обусловленности:**

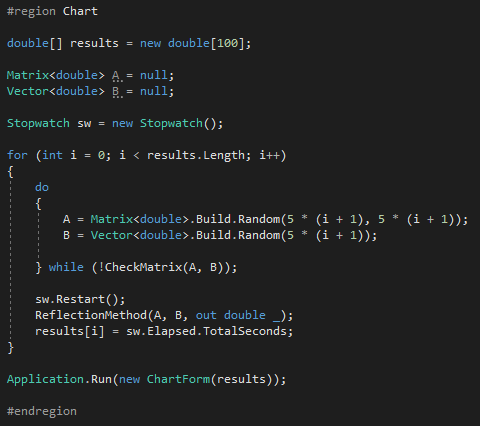


**Такое громадное число говорит о том, что задача плохо обусловлена и результат очень чувствителен к точности входных данных. Это можно увидеть немного “возмутив” вектор b (слева-направо изменения сильнее):**

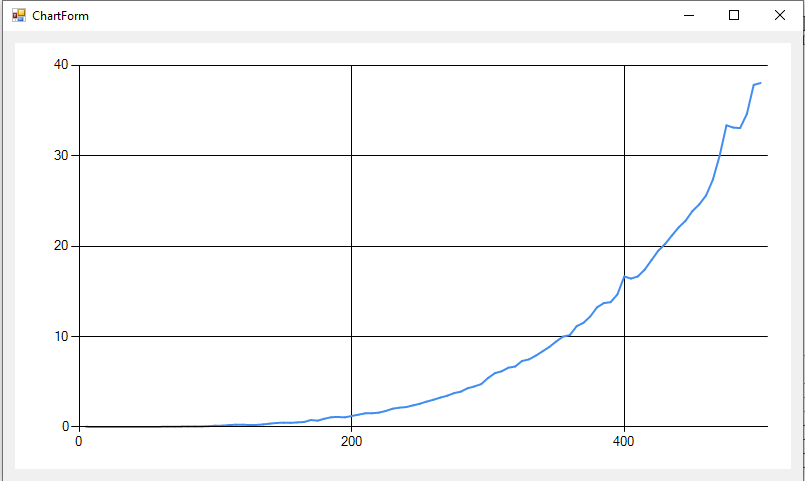
**Задание 4**

Я проверял СЛАУ размерности до 500 с интервалом в 5:



Как я уже писал выше, реализация не очень оптимальна и много времени уходит на сборку мусора. Так же, насколько я знаю, трудоемкость прямого метода отражений выше метода Гаусса, поэтому результаты по скорости хуже, чем у одногруппников с которыми я сверялся.

Результат (X/Y – размерность/секунды):



Таким образом за 1 минуту решается СЛАУ размерностью в 600. Метод из библиотеки (QR разложение) справляется с ней за доли секунды.

**Заключение**

В процессе работы я изучил стандартную для платформы .Net библиотеку для работы с линейной алгеброй, реализовал один из точных методов решений СЛАУ и на практике ощутил значимость обусловленности задачи т.к. результаты задания 3 меня очень удивили.