**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Романова Елизавета Юрьевна

Вариант 11

Отчет по лабораторной работе №1

«Методы решения СЛАУ» студента 2 курса 14 группы

**Преподаватель**

Воробьев А.В.

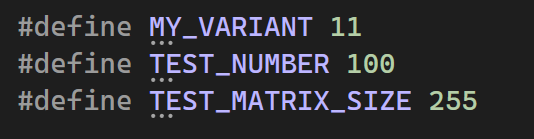
Минск 2022

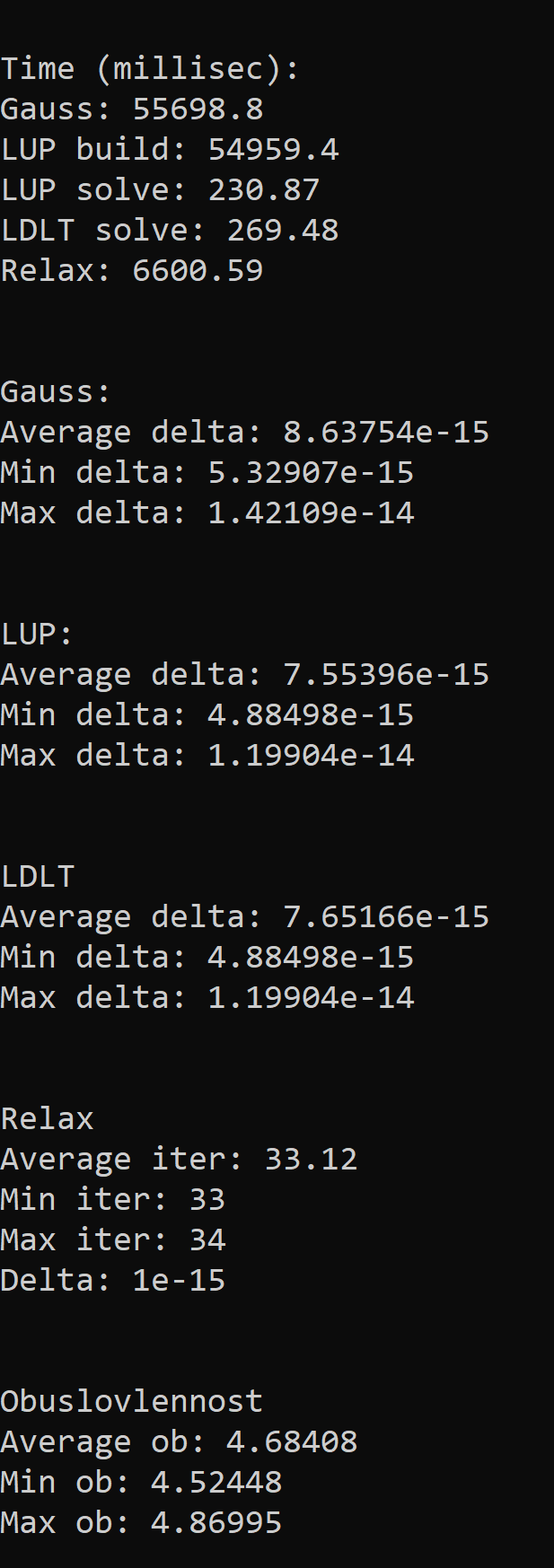
Ссылка на github - <https://github.com/veta1428/MV-year2-sem4-Lab1>

Visual Studio 2022, можно сказать, на С

**Результаты 100 тестов:**

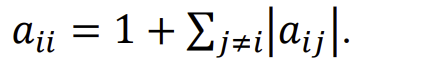
В файле Source.cpp есть константы, если захочется проверить на меньшем количестве тестов, с матрицами другого размера или другим вариантом. (Не касается матриц А1 и А2)





Здесь приведено полное время для метода Гаусса – и приведение к треугольному виду и решение. LUP build – только время на построение, LUP solve и LDLT solve – только на решение, Relax – все итерации + получение L и R.

Можно сделать вывод, что основное время идет на получение треугольного вида, т.к. построение LUP и метод гаусса уже с решением работают за сравнимое время, а вот решение системы, имея треугольные матрицы (LUP LDLT) происходит во много раз быстрее.

Метод релаксаций сходится довольно быстро, за 33 34 итерации, т.к. имеется строгое диагональное преобладание (мы строили симметричными эти матрицы, а диагональ – сумма модулей всех элементов в строке + 1): 

А потом, делая B = E - DA мы получили матрицу B, кубическая норма которой строго меньше 1, значит выполняется условие сходимости.

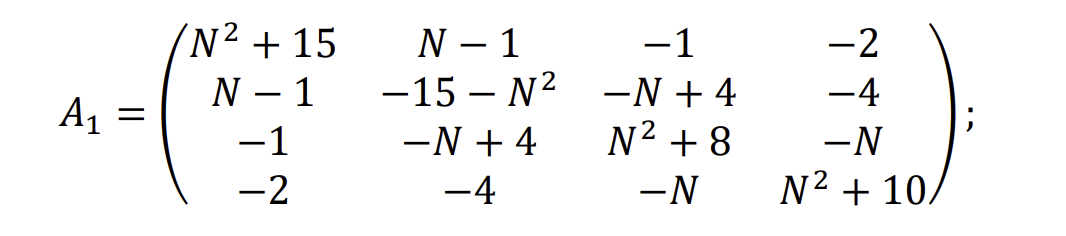
На больших матрицах действительно метод релаксаций работает быстрее метода гаусса.

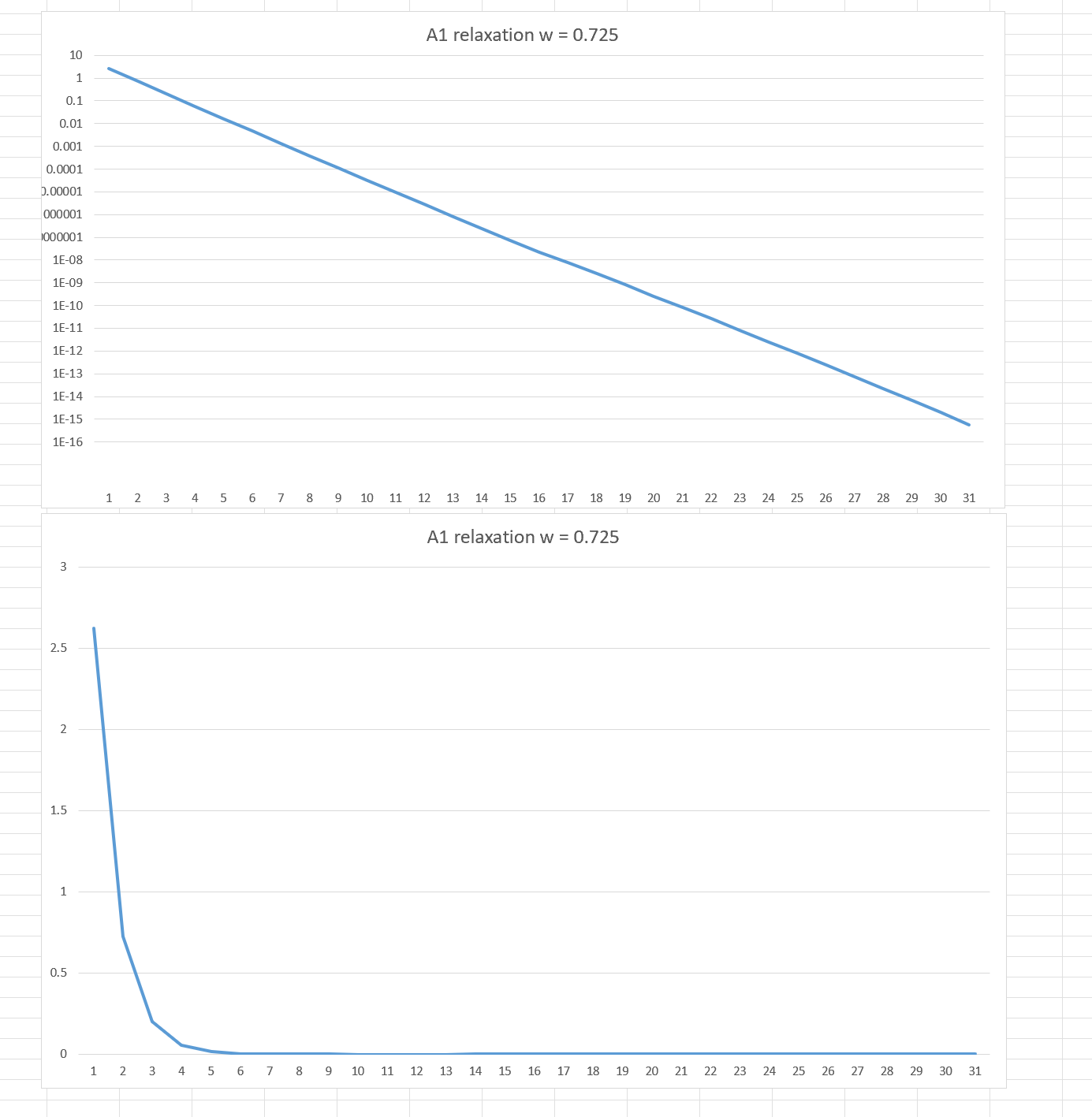
Причем, я считаю, что для алгоритма он не хуже (не стоит пугаться того, что он дает лишь приближенное решение), т.к. точное решение с double методом Гаусса мы не получим никогда (в общем случае).

Меня сначала удивило, что у матриц так мало отличается число обусловленности (не знаю, насколько 0.3 это малое для числа обусловленности, но я думала, будет больше). Но, возможно, тут просто статистика. Я использую рандом с равномерным распределением => модуль большинства элементов будет около 2 для моего варианта, всего элементов 255, а еще есть диагональный со своей суммой => около 1100 выйдет число обусловленности сгенерированной мной матрицы. Ну а дальше, обратная матрица, возможно, тоже не будет сильно отличаться и получится примерно одинаково. Также, у большего размера матриц по идее обусловленность должна отличаться еще меньше (больше выборка, ближе к равномерному).

Точность немного меньше у метода Гаусса, чем у остальных (не рассматриваю тут метод релаксаций, его точность мы сами задали). Я думаю, это потому, что в методе гаусса мы портим вектор b пока строим треугольник, а мы уже знаем, что это влияет на решение и чем больше обусловленность, тем хуже. Тут обусловленность не сильно большая, поэтому, я думаю, метод и не сильно хуже.

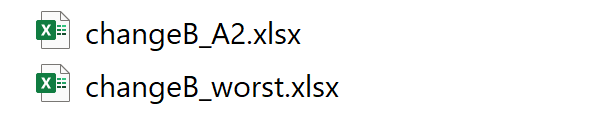
**Релаксация матрицы A1 c очень маленьким числом обусловленности**:



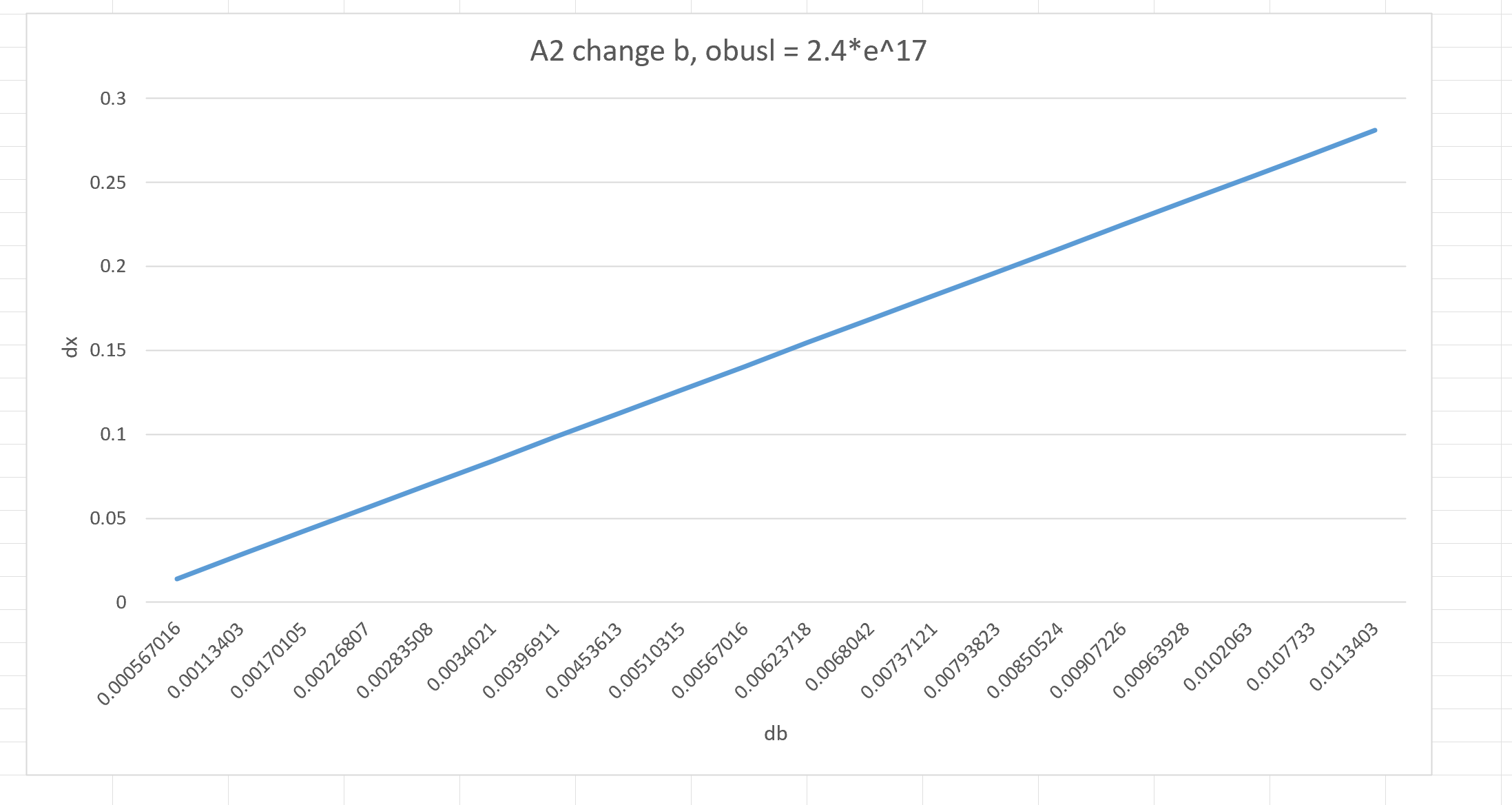


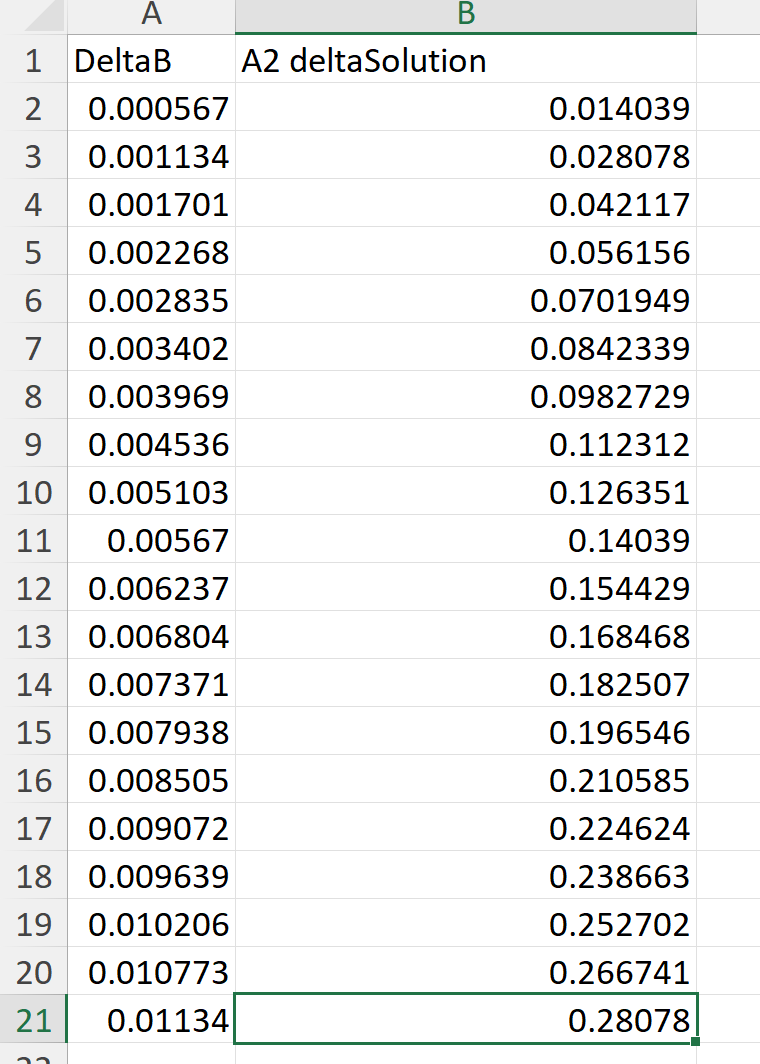
Видно, что довольно быстро (относительно) за 31 итерацию достигается максимальная точность (ограниченная double).

**Влияние возмущения вектора b на решение:**

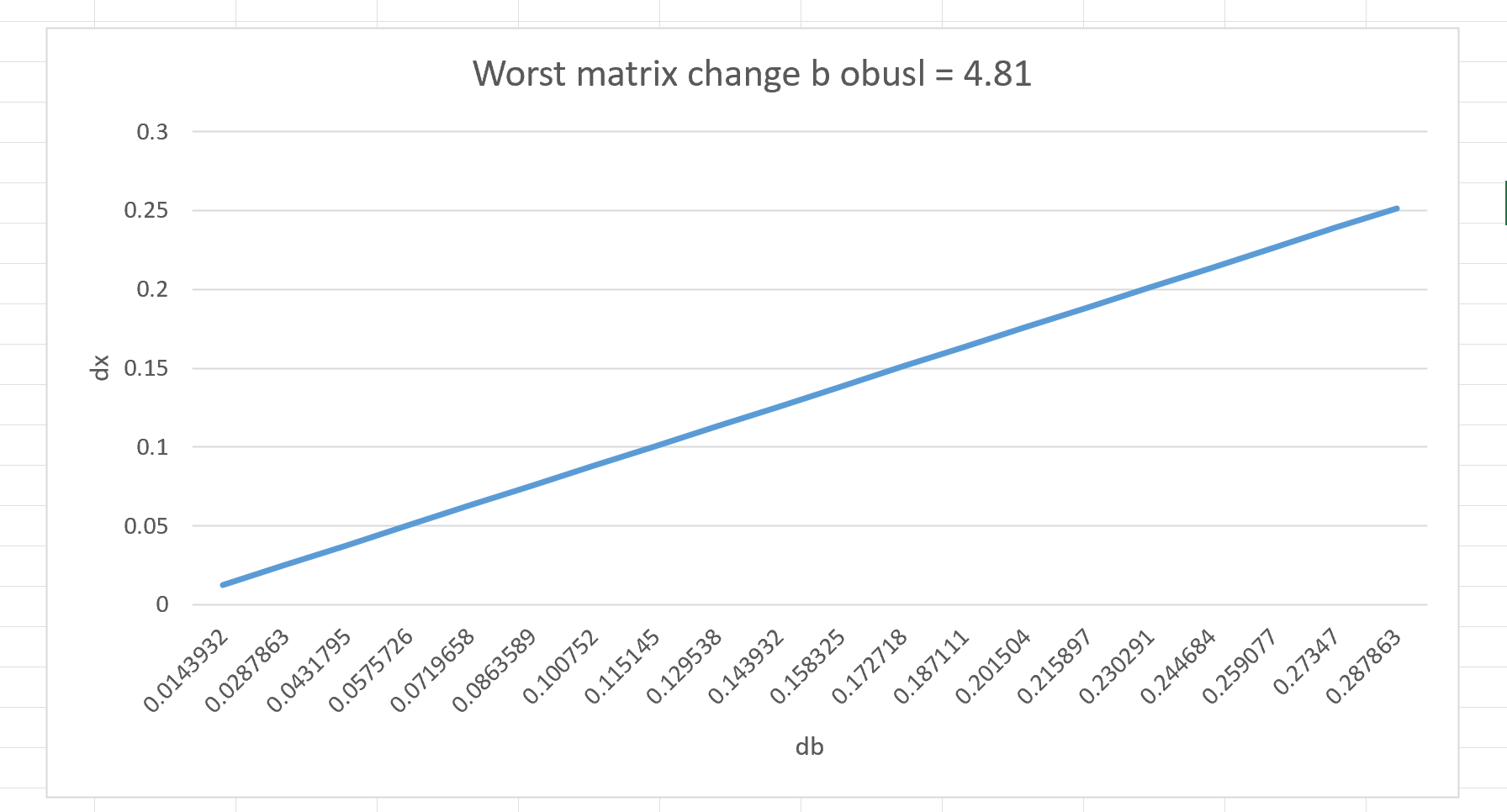
Вот в этих файлах данные: 

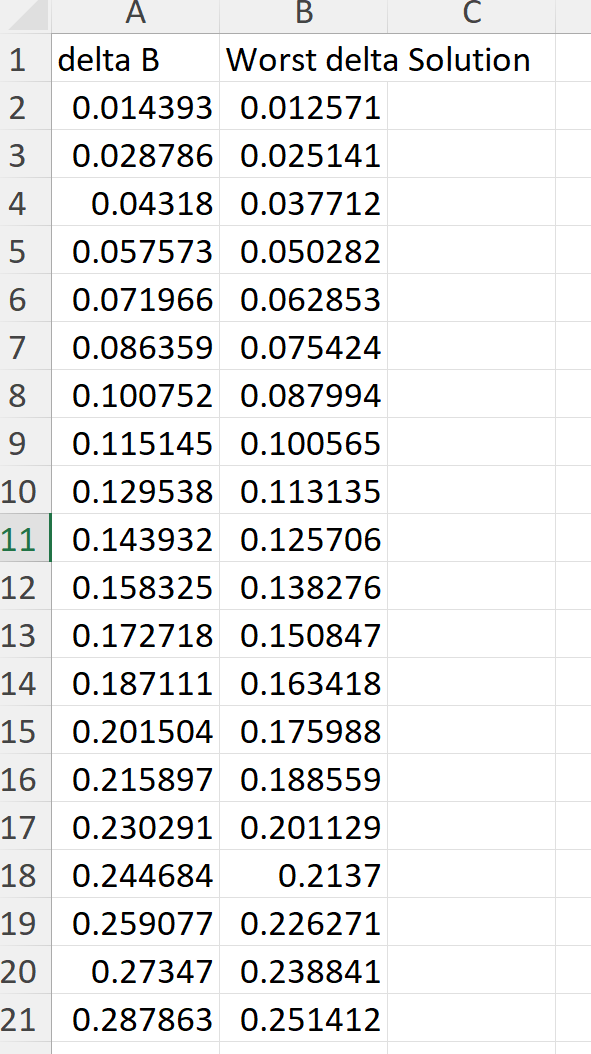
А2:





Худшая матрица (наибольшее число обусловленности):





Проверялось путем изменения каждой координаты решения на одно число и получением нового вектора b. Потом расчет относительных погрешностей.

Худшая матрица разнится от запуска к запуску из-за рандома, у меня была вот такая. Dx, db – относительные погрешности. Видно, что у матрицы с большей обусловленностью прям катастрофически влияет на решение возмущение вектора b. Неравенство dx <= obusl \* db выполняется, но это лишь граница сверху.

В матрице А2 при изменении вектора b на один процент решение изменяется на 28%! В моей худшей матрице.

Линейность можно объяснить изменениями, которые я проводила с решением.

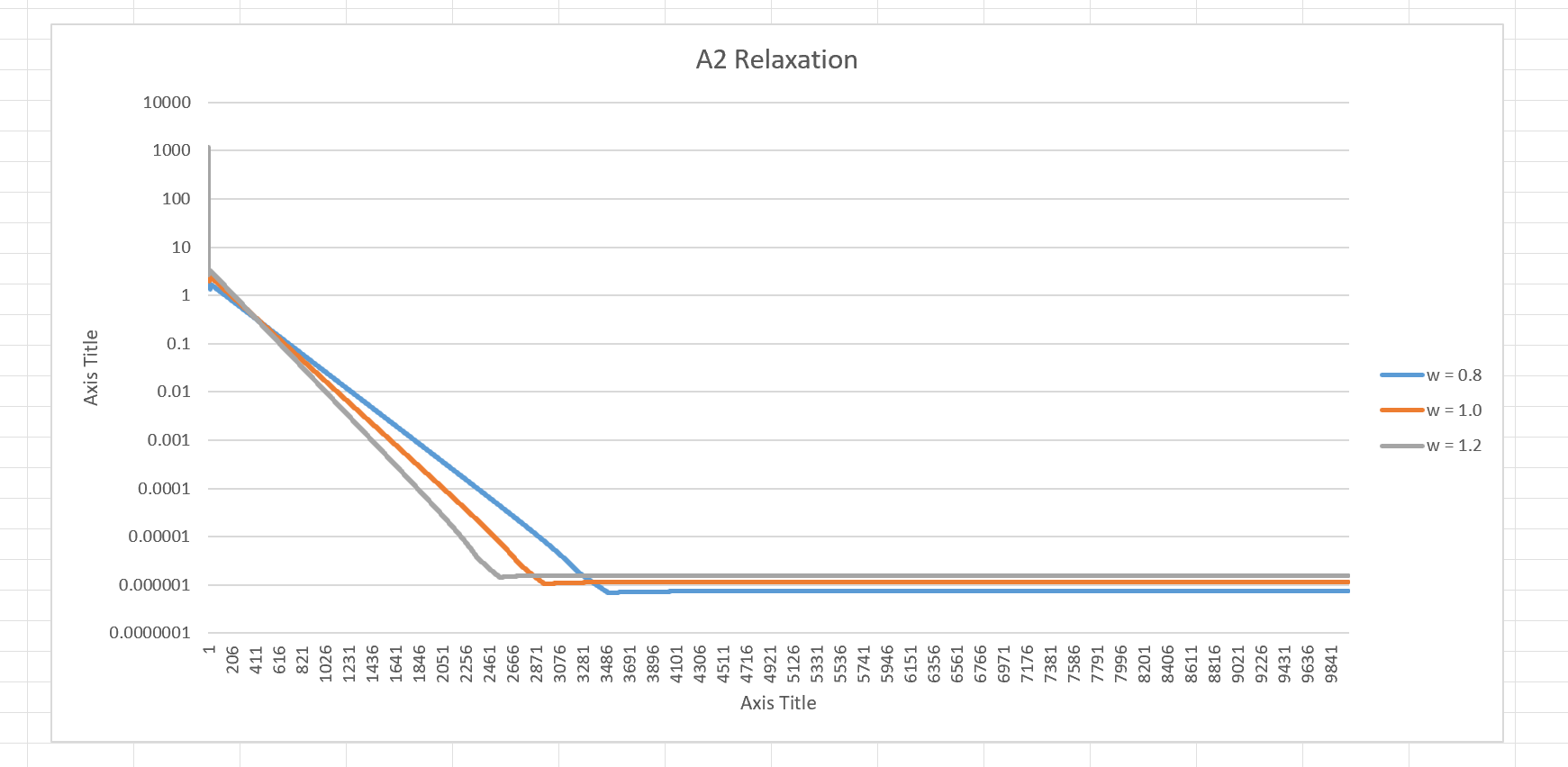
Т е в данном случае обусловленность влияет угол наклона прямой. Она больше – угол больше.

**Метод релаксаций с разными параметрами:**

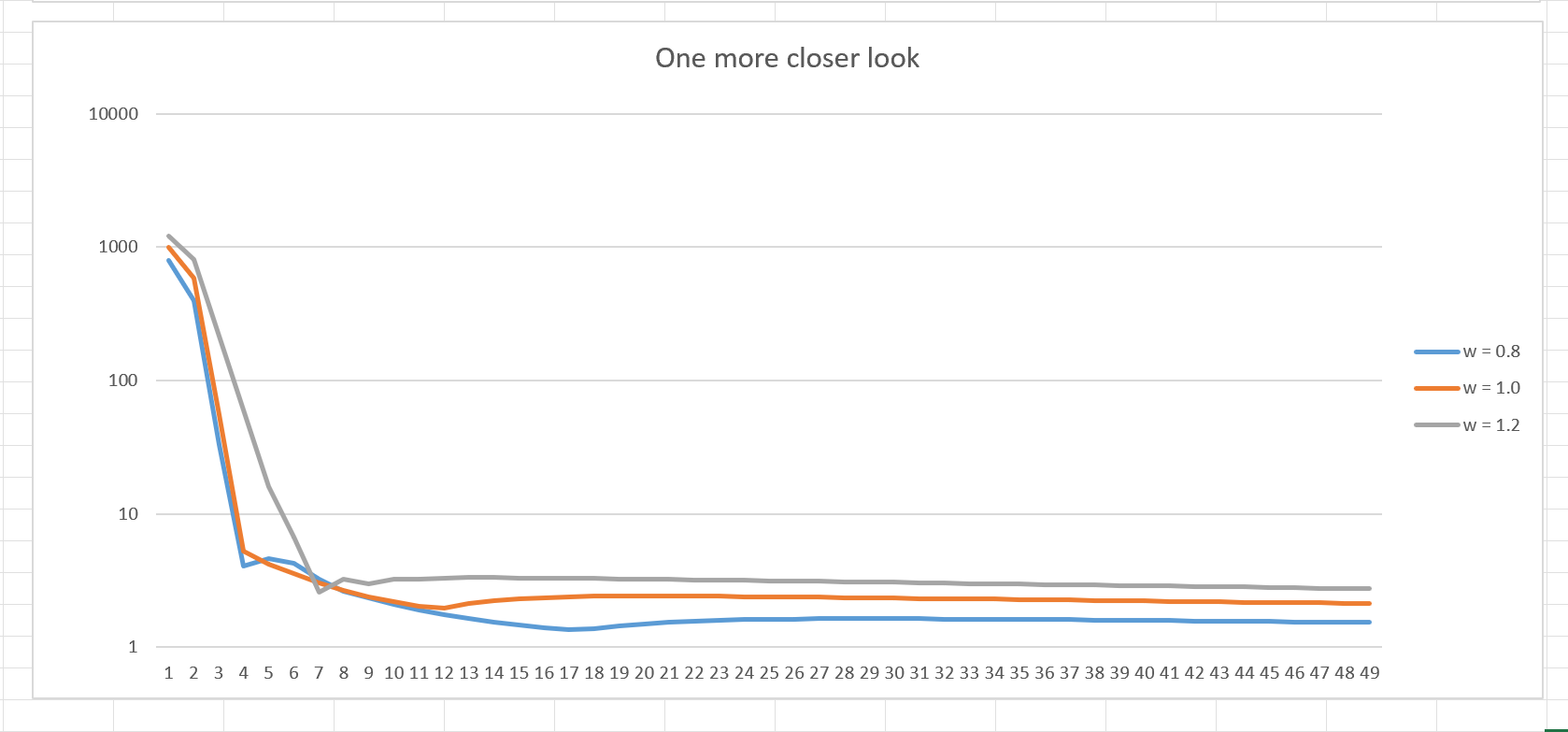
****

****

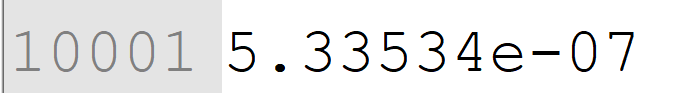
Моя А2:





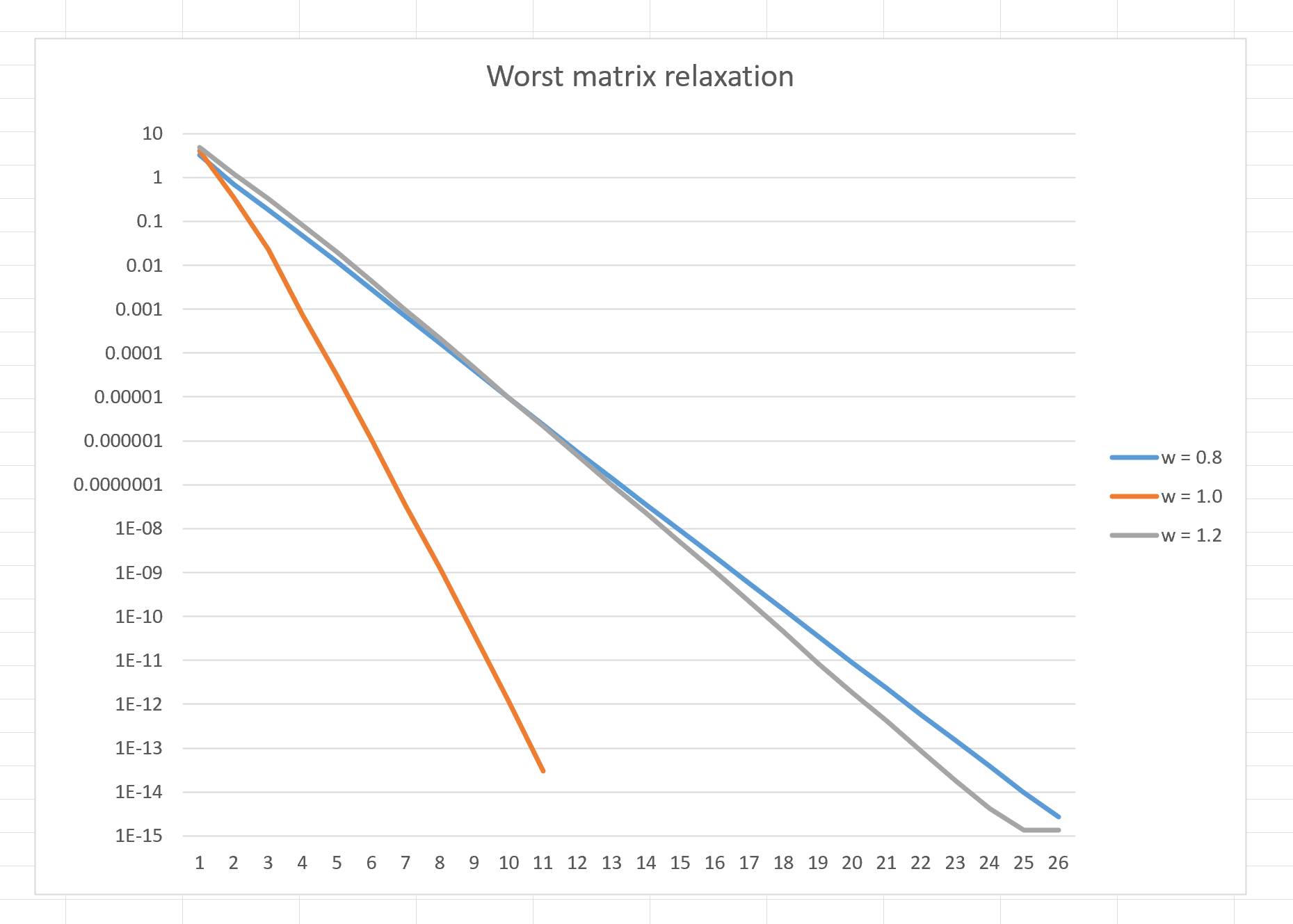


Это один и тот же график, просто разные масштабы.

Вот это уже интересно. У многих ребят метод релаксаций разошелся на этой матрице. У меня же просто сходимость резко замедлилась, но, подозреваю, если долго ждать, очень долго, она сойдется. По крайней мере, я смогла на 10000 достичь такого 

Так что в каком-то смысле, можно сказать, что сходимся) Немного лучше тут себя показал w = 0.8

Моя худшая матрица:

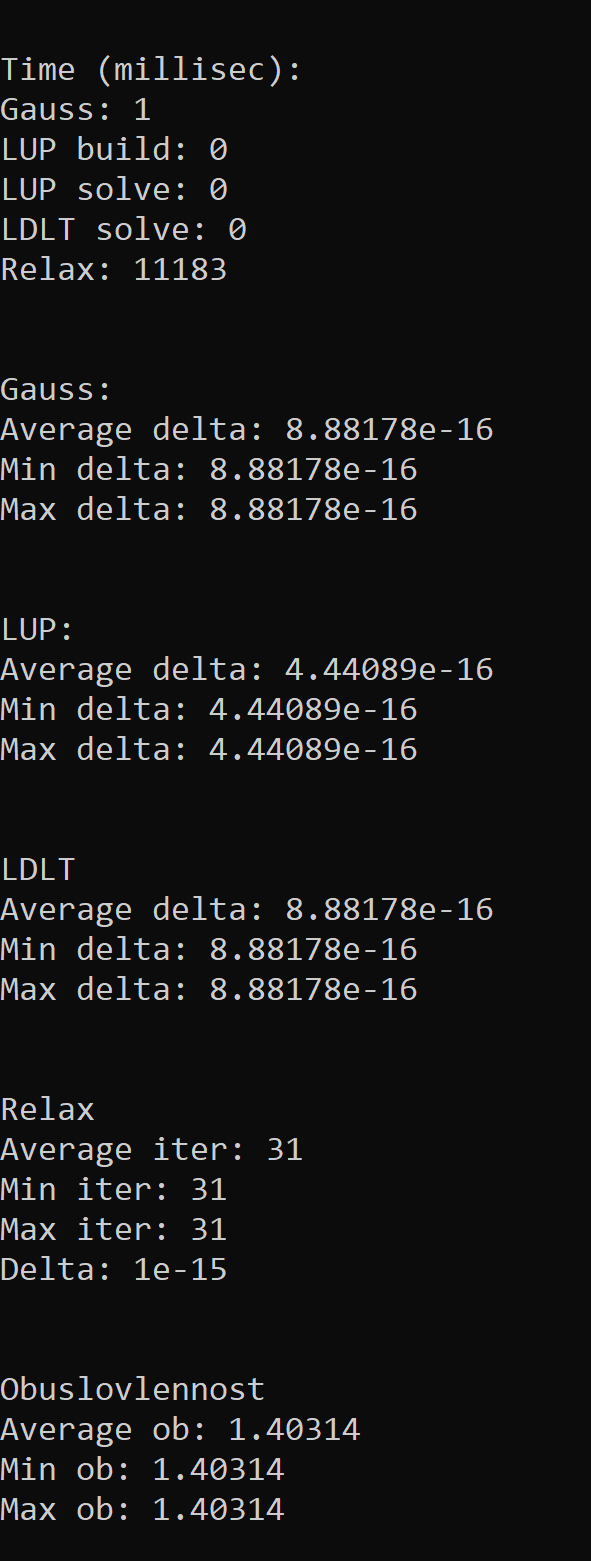


Тут мы сошлись) На w = 1 за 11 итераций, на w = 0.8 за 27 итераций. На w = 1.2 не совсем, там все резко замедлилось. Тут лучше всего себя показал w = 1. Сложно сделать вывод, какой параметр оптимален, так как, очевидно, это зависит от матриц, а значит и от ситуации. Ну или придется тогда посчитать в % при каких W какие матрицы сходятся быстрее) Видимо, как правило, в большинстве случаев метод сойдется быстрее при w = 1.

Сложно (и неправильно) делать какой-то вывод на примере 2 матриц)

**Индивидуальные тесты:**

A1:

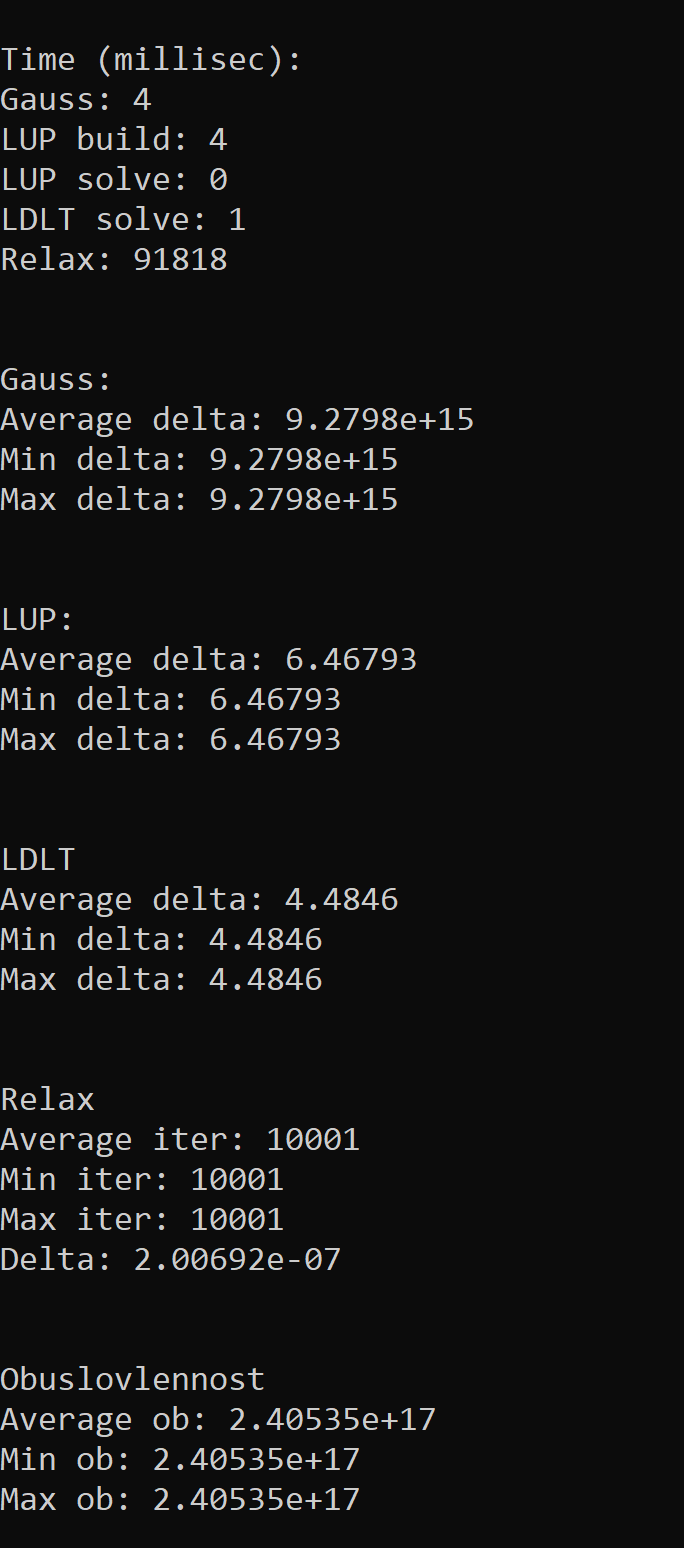
****

У этой матрицы маленькое число обусловленности => возмущение вектора b влияет не так сильно на решения. Отсюда маленькие погрешности.

Метод итераций на маленьких матрицах не эффективен.

Решение треугольных систем даже меньше микросекунды.

A2:



У этой матрицы просто гигантское число обусловленности. Из-за этого все сходит с ума) Большие погрешности (огромные просто, если на контрасте смотреть) У метода Гаусса что-то пошло не так, хотя если вспомнить наше неравенство, ограничивающее сверху погрешность вектора решений, то при такой обусловленности и такая погрешность возможна. LUP LDLT чувствуют себя лучше (возможно, потому что мы в них не портим так сильно вектор b). Метод итераций пришлось остановить из-за того, что он долго слишком сходился (описано выше).

Для себя я вынесла то, что метод релаксаций, не дающий точного решения, не так и плох. (Но на контрольной все равно не рискнула его использовать, т к после того, как увидела, что во 2 задании LUP по матрице, которого по обещанию не должно было быть, решила, что в 1 задании 100% миллиард итераций придется сделать)). Также, если меняется только вектор b, то очень круто можно разложить матрицу и гораздо быстрее все решать. Еще, то что матрицы бывают ужасными для таких вычислений и отсюда у меня вопрос, какая часть всех матриц неудобна для вычислений) Надеюсь, небольшая)