БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

ОТЧЕТ ПО МЕТОДАМ ВЫЧИСЛЕНИЙ

студента 2 курса 13 группы

Лабораторная работа №3

Преподаватель

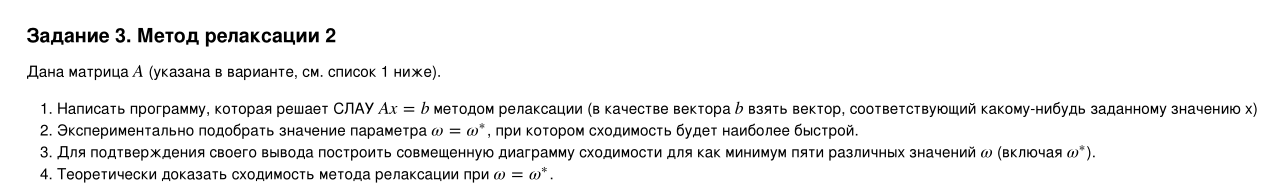
Бондарь И.В.

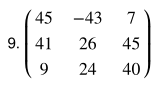
Минск 2020

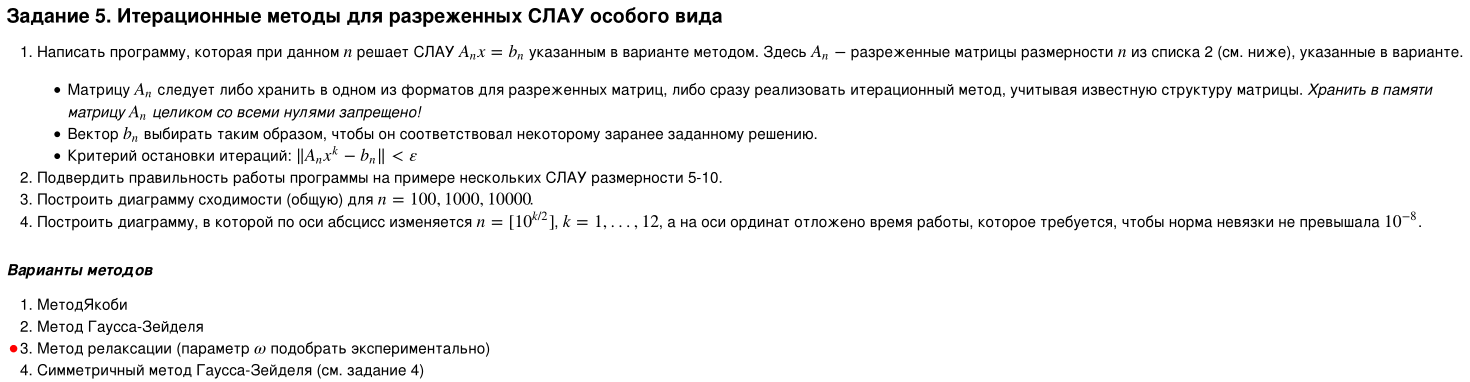
Вариант 9

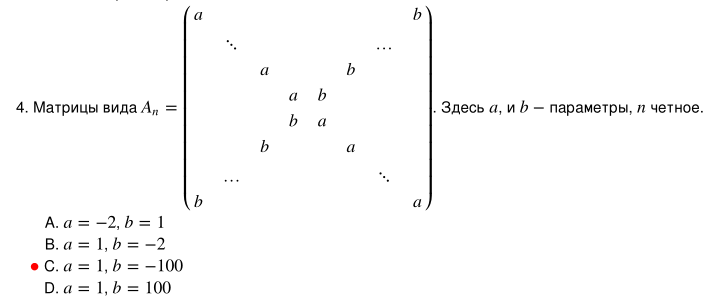
**Условие**





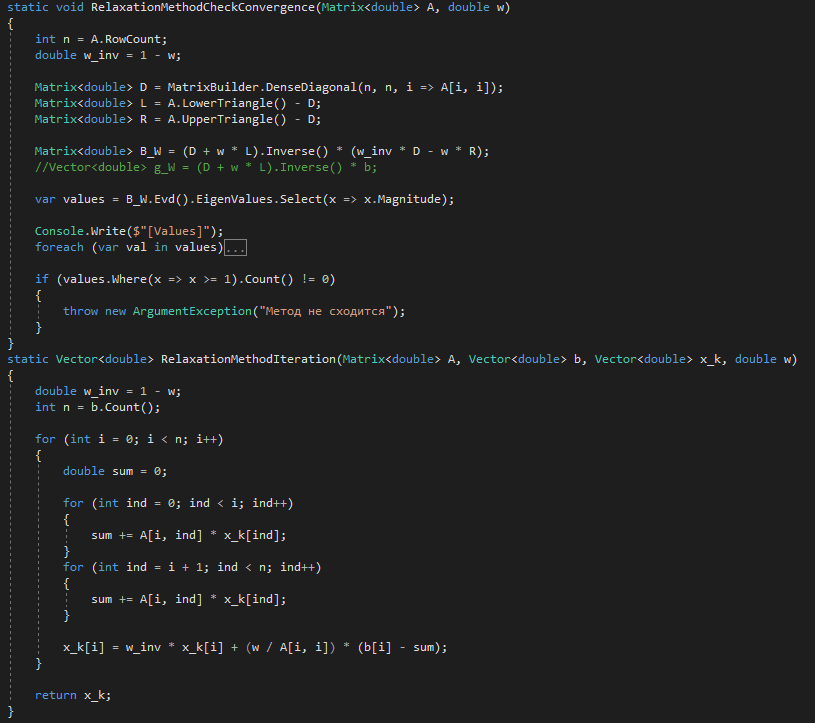




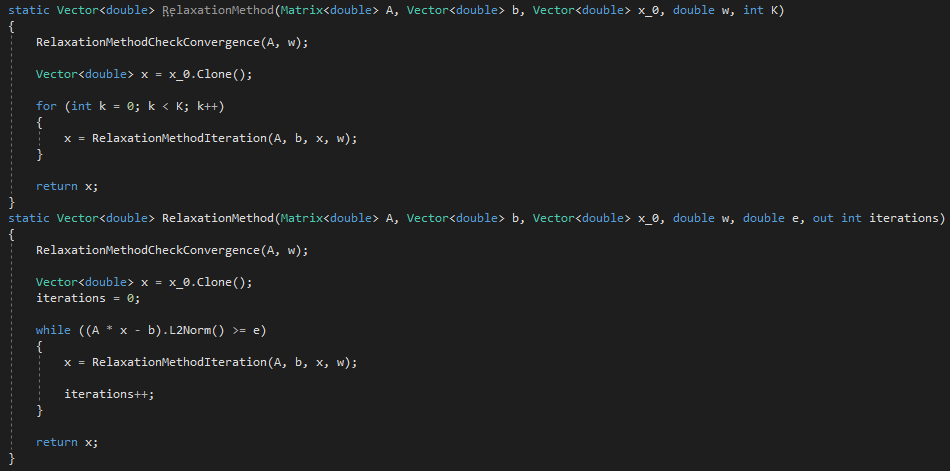


**Задание 3**

Реализовал методы проверки сходимости и итерации алгоритма:



Реализовал методы для алгоритма по количеству итераций / требуемой точности:

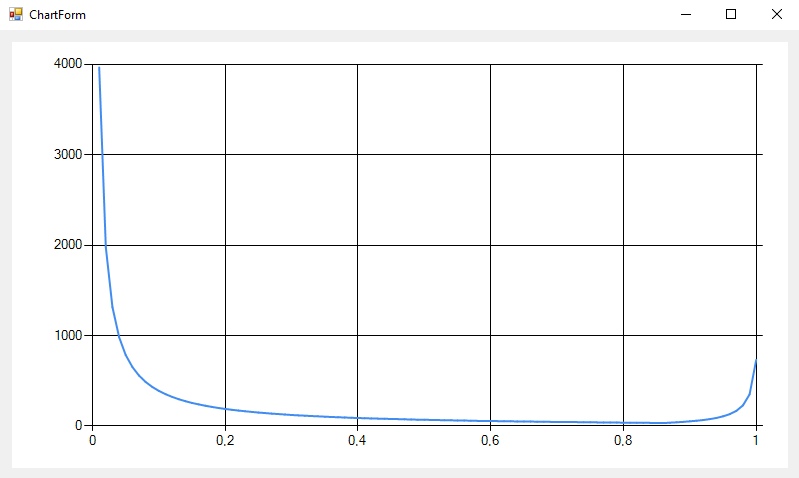


При вычислениях брал параметр точности E как 1 \* 10-8.

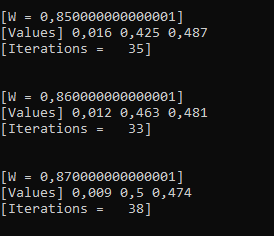


Так же, метод с моей матрицей сходится с параметром w из диапазона (0,1), что проверено экспериментально и видно на графике ниже.

Далее экспериментальным путем подобрал параметр, при котором сходимость была наиболее быстрой. Для этого построил график зависимости числа итераций от w + делал вывод в консоль:



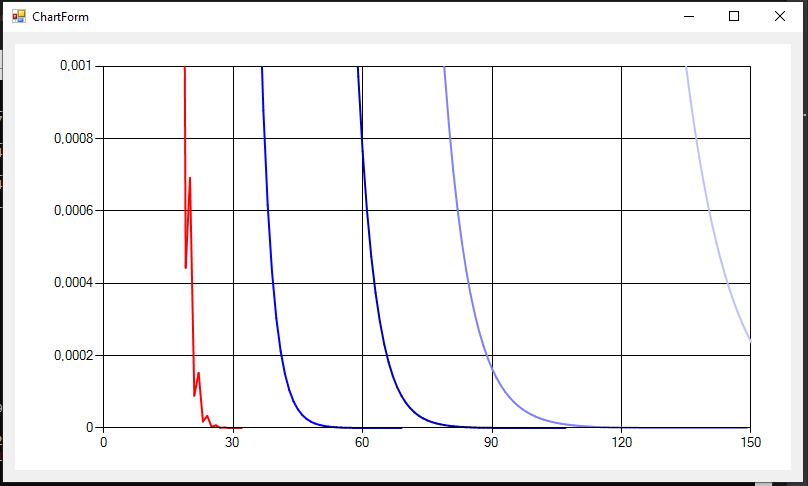
Можно видеть, что минимум находится где-то между 0.8 и 0.9, сверяясь с консолью, получаю w\* = 0.86.



Тут же можно видеть абсолютные значения собственных чисел матрицы Bw (блок Values), максимум из которых является спектральным радиусом. Видно что он < 1, а значит метод сходится по критерию сходимости. Метод проверки значений и определения матрицы Bw был приведен выше.

Далее построил совмещенную диаграмму сходимости для значений

w = {0.05, 0.25, 0.5, 0.86, 0.95}:



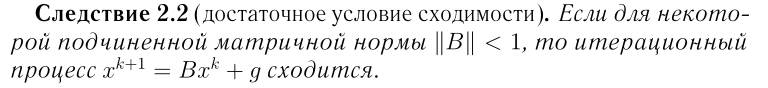
На графике слева – направо линии для значений 0.86 / 0.5 / 0.95 / 0.25 / 0.05 соответственно. Видно, что при значении w = w\* = 0.86 метод и правда сходится быстрее.

**Задание 5**

Для начала я решил научиться определять сходимость данной матрицы, для воспользовался уравнением

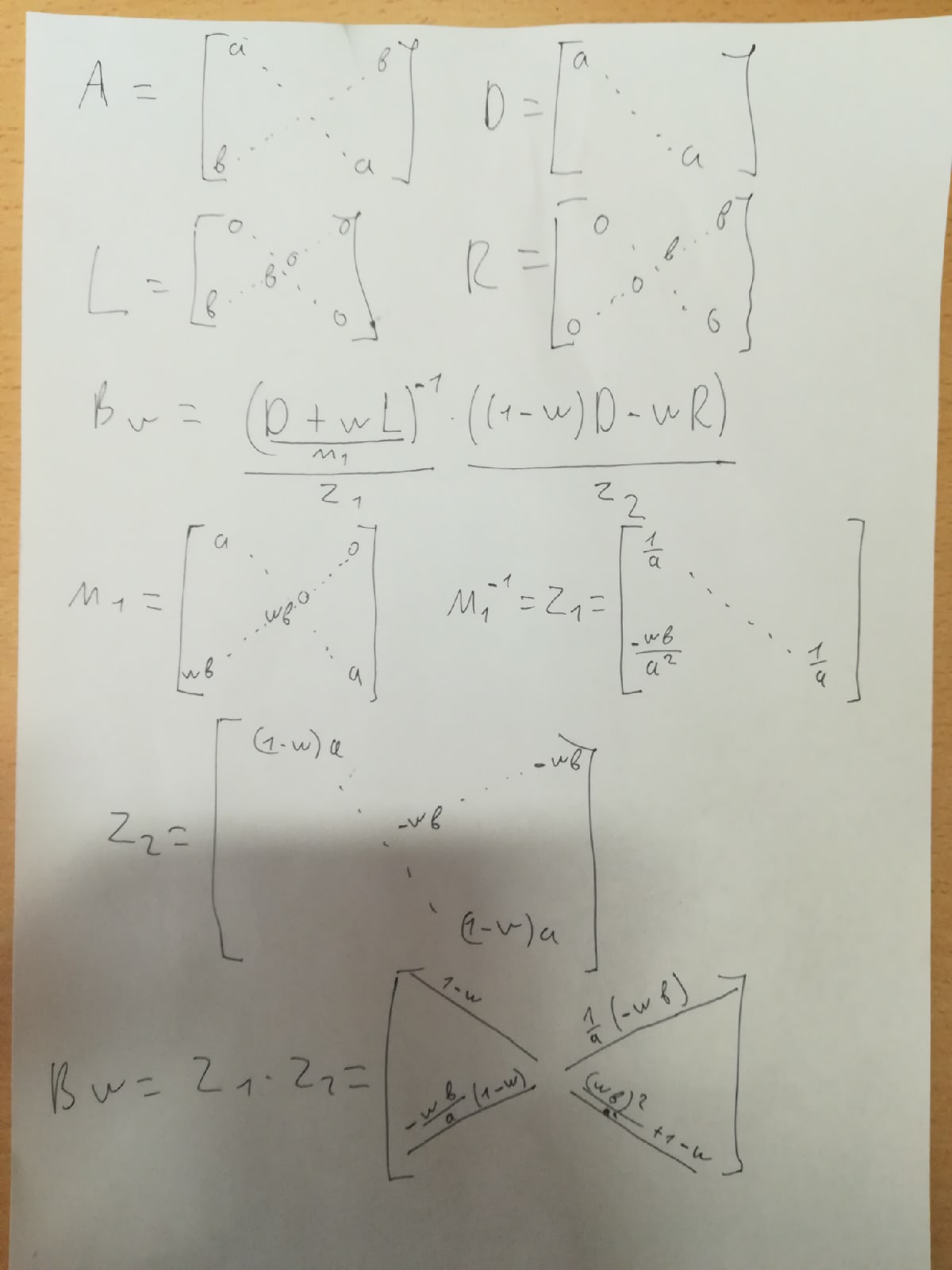


преобразовав его к такому виду, где я мог бы проверять эту матрицу, не храня значения. Сначала я протестировал вид этой матрицы на конкретных значениях, оказалось, что эта матрица так же будет состоять из перекрещивающихся диагоналей, при этом каждая половина каждой диагонали, будет иметь своё постоянное значение, зависящее от параметров *a, b, w* матрицы и независящее от n. Идея была в том что в таком случае я могу легко определить сходимость исходя из достаточного условия:

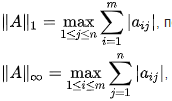


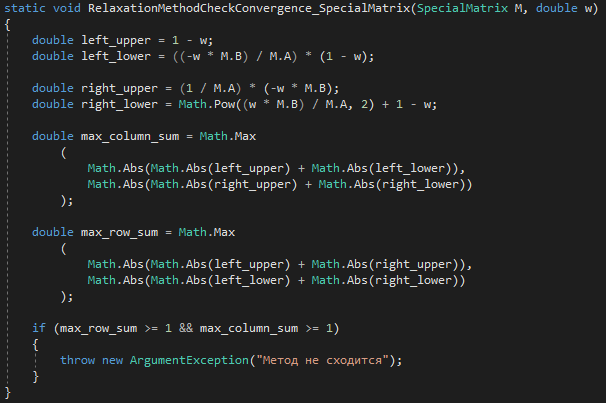
Подчиненные нормы для такой матрицы считаются элементарно, а вот собственные значения для матриц размера n = 10^6, я бы искать не хотел.

Без промежуточных вычислений у меня получилось следующее:

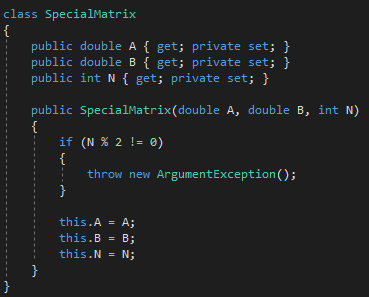


По этим формулам я реализовал проверку на нормах (строка и столбец всегда задаются 2-я комбинациями 2-ух элементов половин диагоналей):

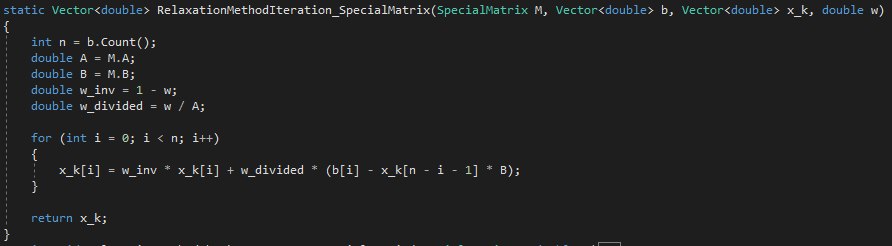




Где SpecialMatrix – очень простой класс, задающий параметры матрицы:

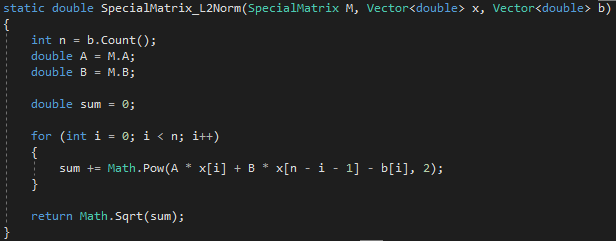


После этого реализовал метод итерации:



Для моей матрицы, суммы в правой части формулы всегда состоят из одного элемента, умноженного на значение B.

Так же реализовал расчет нормы, формула легко выводится из формулы умножения матрицы на вектор:



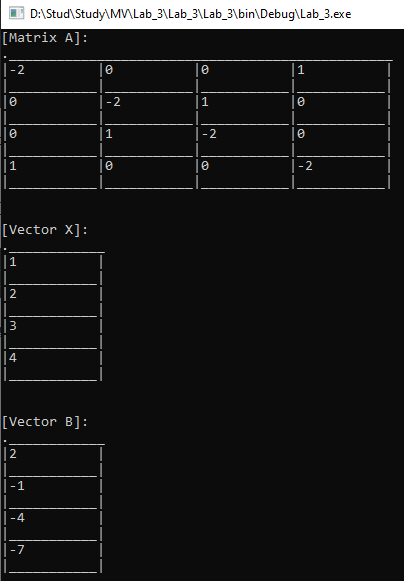
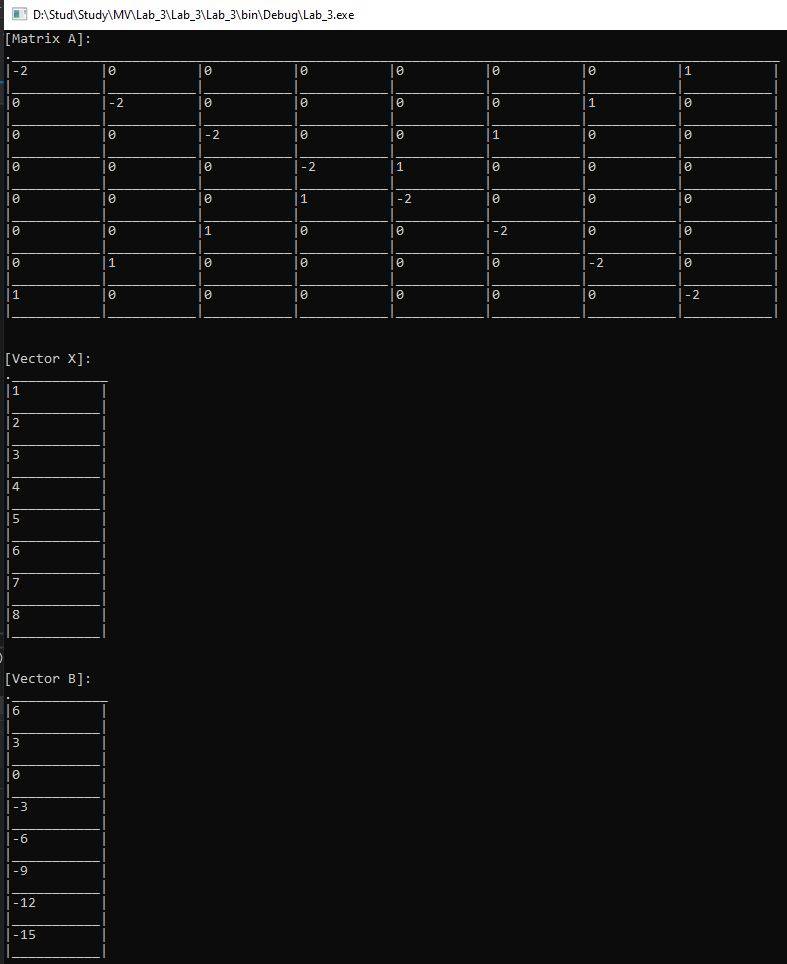
Проверка метода:

Параметры: a = -2, b = 1, w = 1

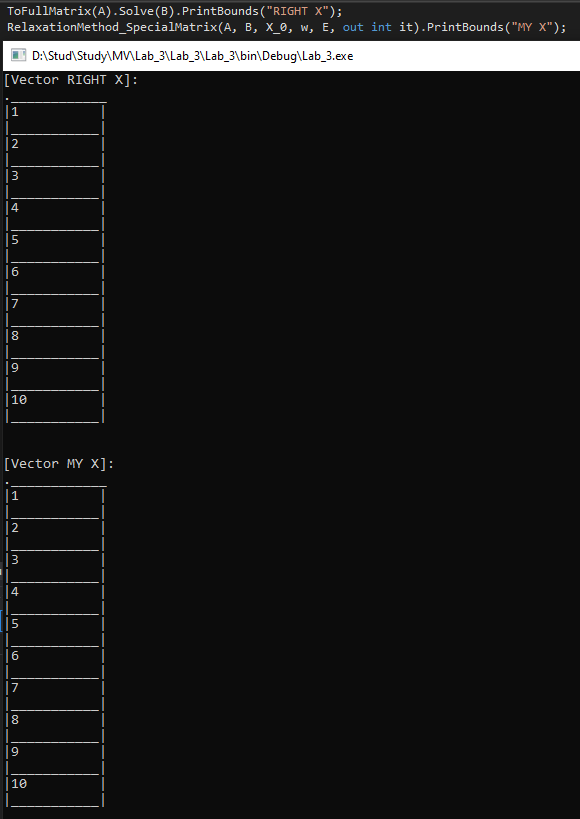
Вектора: x = 1…n, bi = (n – 2) – (3 \* i), где i = 0…n – 1 – универсальная формула для любых n, при параметрах выше:



Вот, например, как выглядят матрицы и вектора для n = 4, 8:

Собственно, проверка для n = 10:

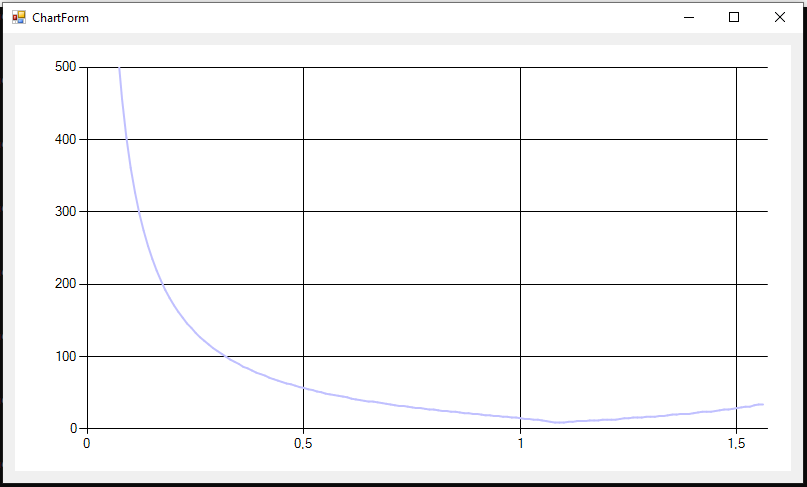


Где метод ToFullMatrix преобразует мою матрицу в матрицу MathNet, после чего вычисляется точное решение методом из библиотеки.

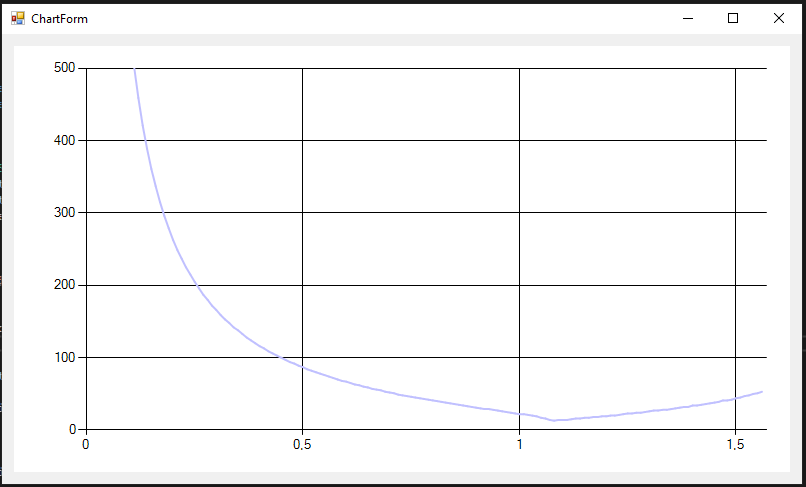
После этого я вычислил оптимальное значение w = w\* для параметров из задания. Единственное что оказалось, что значения параметров a и b не играют никакой роли для сходимости (нормы матрицы Bw не изменялись, как и сама матрица, что видно из выведенной формулы) и не влияют на выбор оптимального w, на это влияет лишь их отношение, поэтому вместо (a,b) = (-200, 100) я брал (-2,1), исключительно для удобства и адекватности значений X.

Так же, как и в задании 3, я построил диаграмму числа итераций от значения w + делал вывод на консоль. Оказалось, что значение w\* не зависит от размера матрицы, вот, например графики:

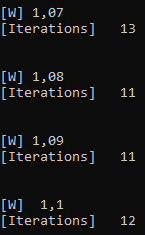
Для n = 10



И для n = 10000

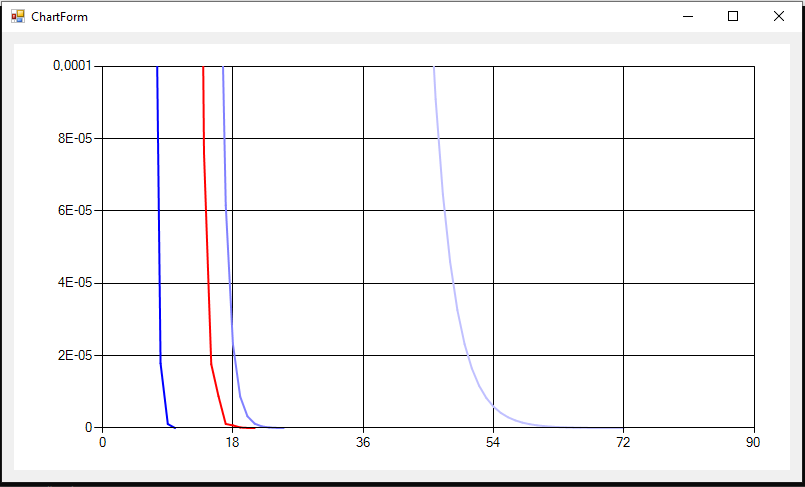


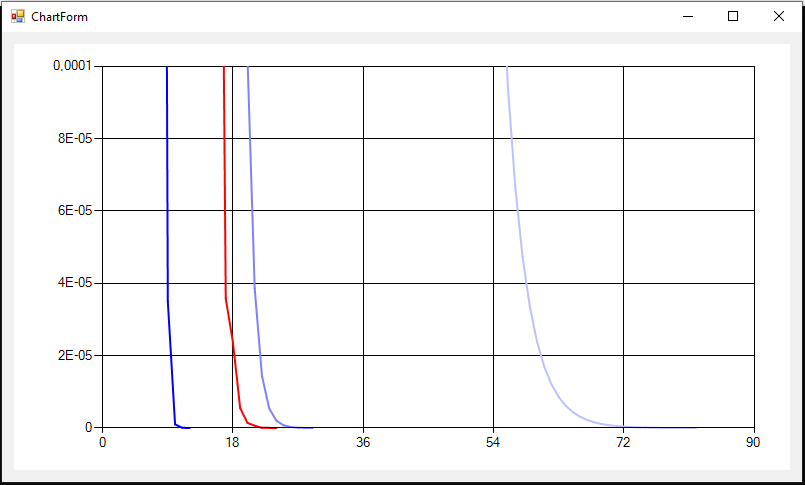
Видно, что они буквально идентичны. Сверяясь с консолью:

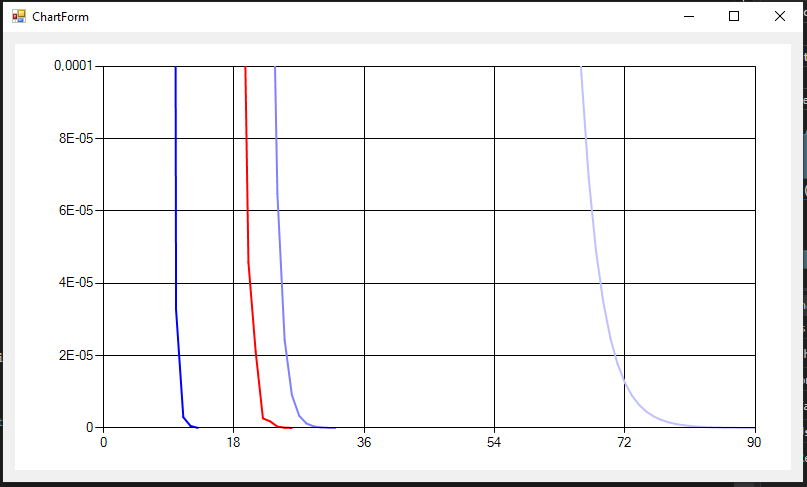


оптимальное значение w = w\* = 1.08.

Диаграммы сходимости для n = 100, 1000, 10000 (w = 1.08 / 0.9 / 1.3 / 0.5 слева - направо)

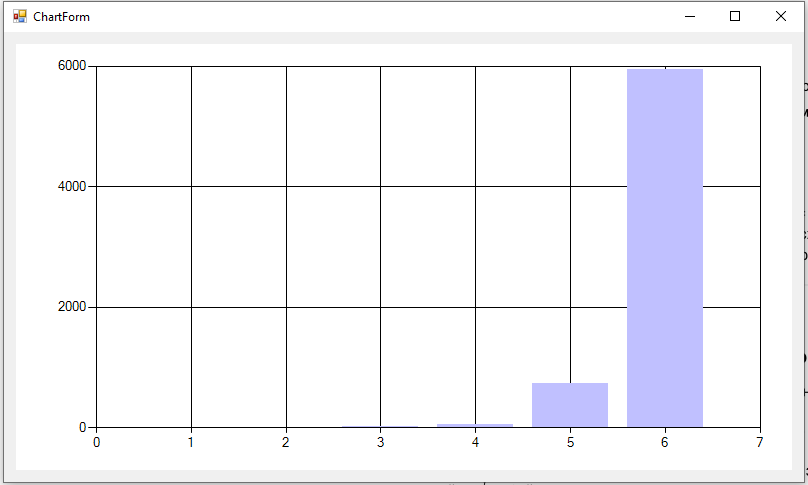






Для построения гистограммы времени работы от размера матрицы, я использовал точность 1 \* 10^7 так как более точный результат у меня не сходился для n = 10^6.

По оси Y – время (мс), по X – степень 10 для n:



Видно, что время работы метода возрастает, если не линейно, то близко к этому.

**Заключение**

В этой работе было практически доказано преимущество метода релаксации по сравнению с методом Гаусса-Зейделя, так как выбор параметра релаксации, часто отличный от 1, давал в разы лучший результат. Во второй части работы, была рассмотрена специальная разреженная СЛАУ, хранение которой, а также специальный метод релаксации, позволяли СЛАУ огромных размеров в кратчайшие сроки. Для хранения полных матриц размерностью n = 10^6, потребовалось бы 8 \* 10^12 байт, или 800 Гб памяти. В правильной реализации, место занимали лишь вектора x и b, суммарного размера в 2\*8\*10^6 байт или 12 мегабайт, что я и наблюдал при дебаге своей программы. Особый вид так же обеспечивает почти линейный рост сложности итерационного метода, в то время как для обычных СЛАУ это как минимум квадрат с коэффициентом >= 1.

Пользовался калькулятором в этой и других дз. Не вижу если честно смысла считать это вручную, особенно зная, что со своей собранностью потрачу на это часы, а не минуты.