Методы вычислений

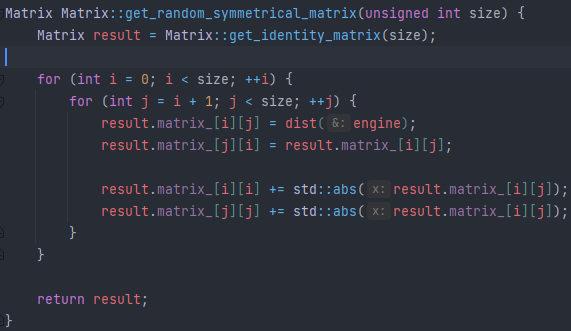
Лабораторная работа 1. Методы решения СЛАУ

Выполнил Емельяненко Павел (Вариант 6)

**Задание 1.**

Для того, чтобы корректно сгенерировать случайное число, был использован генератор C++:  

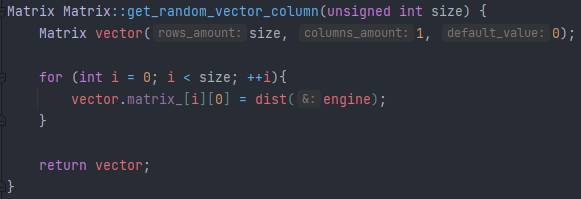

Ограничения, согласно варианту .

Для того, чтобы корректно заполнить матрицу случайными числами и чтобы при этом выполнялось свойство симметричности, мы заполняем верхний треугольник без главной диагонали, параллельно заполняя нижний треугольник соответствующими элементами . Диагональные элементы, согласно условию, получаем как сумму модулей элементов в строке + 1:  


Для реализации функционала матриц и СЛАУ был использован класс Matrix, решения СЛАУ реализованы как статические методы.

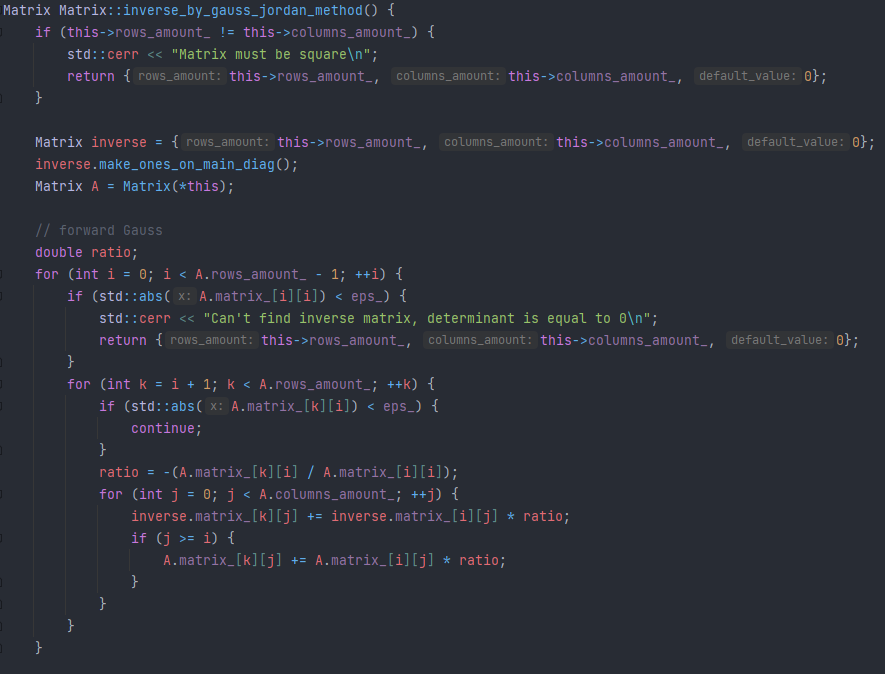
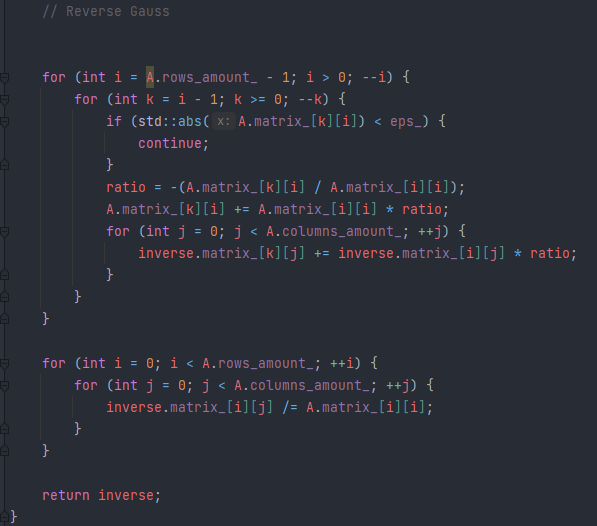
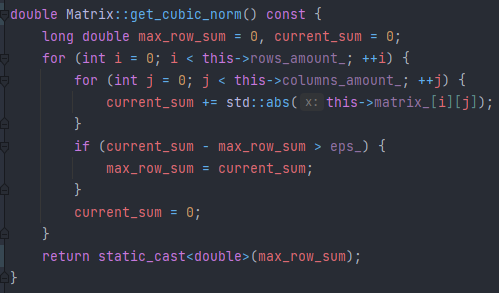
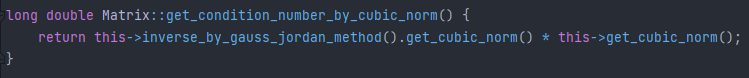
**Задание 2.**

Проделываем аналогично действие, но для вектора. Соблюдается только условие генератора.



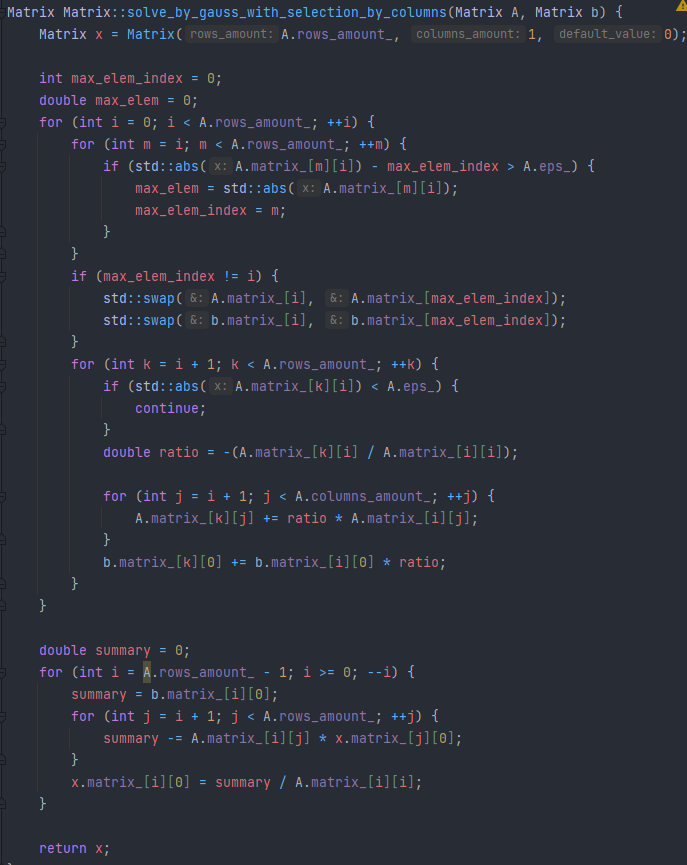
**Задание 3.**

Состоит из 3 пунктов:

1. Построение обратной матрицы методом Гаусса-Жордана:  
   Метод заключается в том, чтобы сделать прямой ход Гаусса, чтобы занулить нижний треугольник, а потом обратный ход, чтобы занулить верхний треугольник, аналогичные действия нужно провести для единичной матрицы. После прямого и обратного хода получаем диагональную матрицу, разделив строки на соответствующие диагональные элементы матрицы преобразованной матрицы А, получаем единичную матрицу слева и обратную справа. Обратную и возвращаем:  
     
    
2. Далее – нахождение кубической нормы – максимальная сумма элементов по модулю в строке:  
   
3. И последнее – подсчет числа обусловленности для матрицы. Общая формула: :  
   

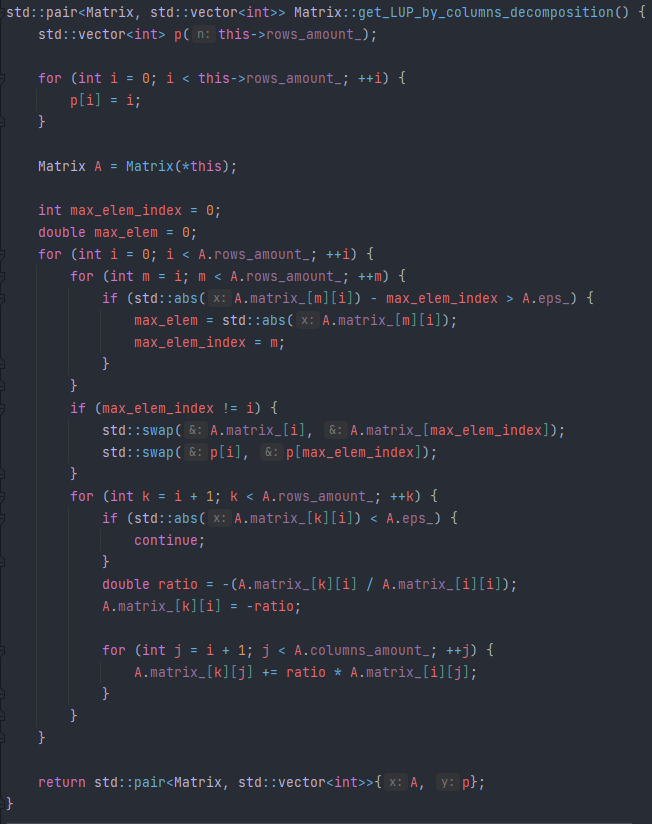
**Задание 4.**

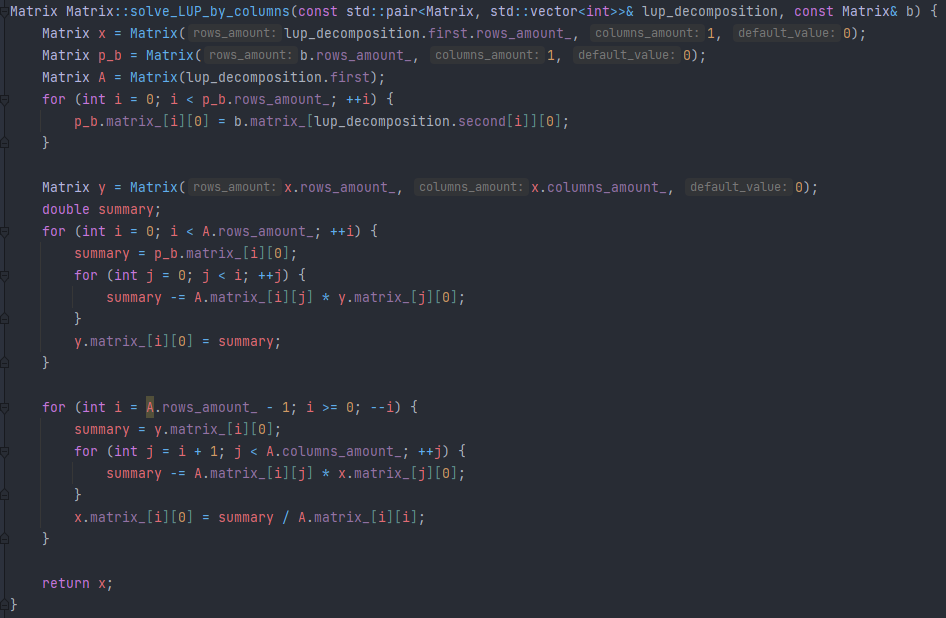
Решить СЛАУ методом Гаусса с выбором по столбцам. Идея метода в том, что мы находим максимальный элемент ниже диагонального в столбце, ставим его на место диагонального, после чего зануляем все элементы под диагональным. Таким образом мы уменьшаем погрешность вычислений, которая возникает, когда мы делим малое значение на большое. Параллельно со сменой строк местами мы меняем элементы в векторе b. В итоге мы получаем верхнетреугольную матрицу и постепенно заполняем вектор x с конца.



**Задание 5.**

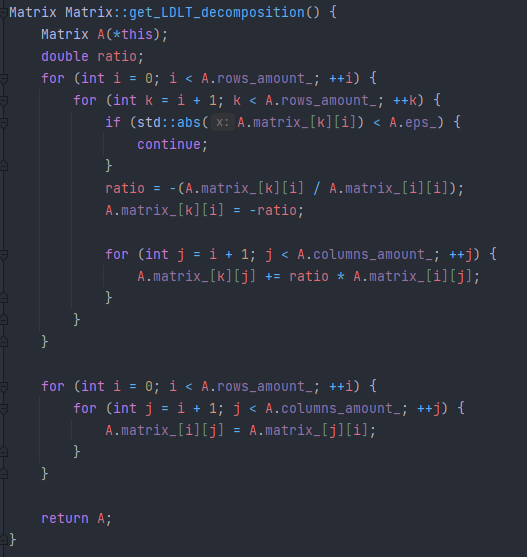
Разделим этот пункт на два:

1. Нахождение LUP разложения, где L – нижнетреугольная матрица, U – верхнетреугольная, P – матрица перестановок.  
   LU разложение получается прямым ходом Гаусса, но на место зануленных элементов будем ставить коэффициент , где – коэффициент, который домножали на i строку и которым непосредственно зануляли элемент . Кроме того, в эту схему нам нужно добавить выбор по столбцу. Мы храним матрицу U и L в одной, что дает нам некоторое преимущество. Если решать, сохраняя матрицы , конечная формула получилась бы , и нам нужно было бы переносить для того, чтобы сгруппировать все , а это много перестановок или же умножений (оба варианта плохи). Если же мы храним все в одной матрице, то меняя строки местами во время выбора по столбцу, мы автоматически формируем матрицу L. После прямого хода получим конечные матрицы L, U, вектор, по которому сделаем перестановки(храним именно вектор, чтобы сэкономить память).  
   
2. Имея разложение LUP разложение, а именно , т.к. выбор именно по стобцу. Подставляем матрицу в начальное СЛАУ, получаем Далее решаем систему поэтапно. Сначала введем замену . Решим систему . Далее решим систему . Таким образом получаем искомый вектор x.

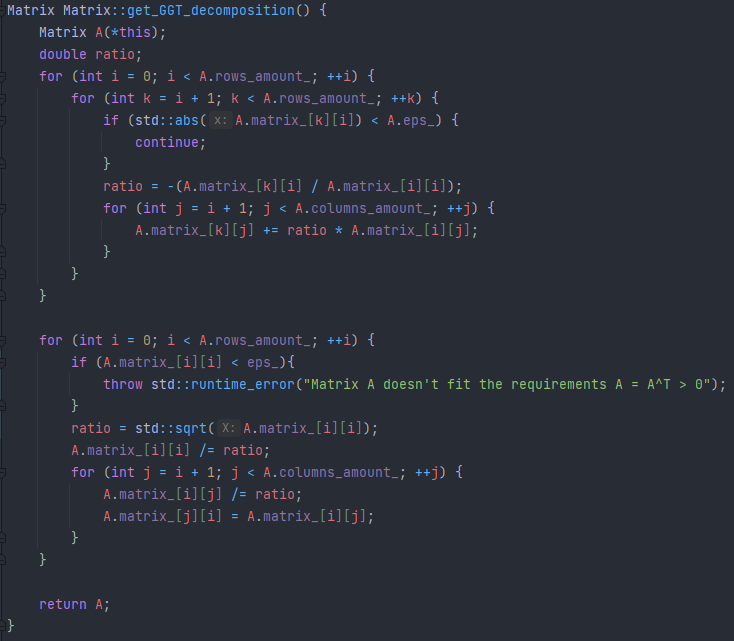


**Задание 6.**

– разложение и решение методом квадратного корня. Для необходимо, чтобы матрица была симметричной и вещественной. В нашем случае все генерируемые матрицы соответствуют требованиям, разложение будет работать. Чтобы получить разложение, мы находим L матрицу из LU разложение, матрица D – диагональная матрица, где главная диагональ соответствует главной диагонали U матрицы.

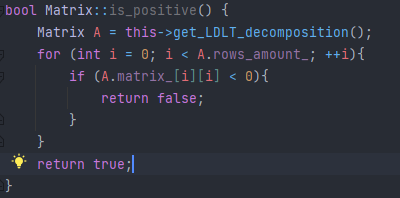
****

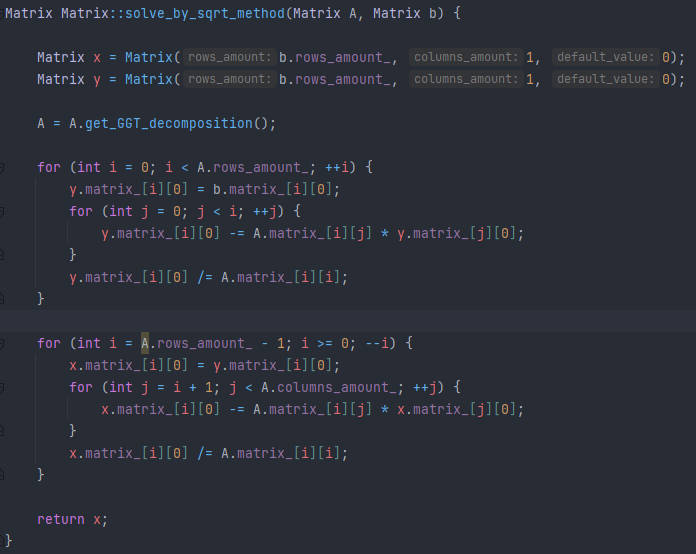
Для решения методом квадратного корня возникает дополнительное условие для матрицы – она должна быть положительно определенной, тогда мы можем использовать преобразование или же разложение Холецкого . Генерируемые матрицы положительно определенные, соответственно мы можем применить разложение Холецкого и решить Полученную СЛАУ по аналогии с тем, как мы решали LUP-разложение:



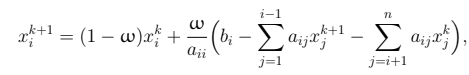
В случае, если матрица не положительно определенная, мы получаем исключение, т.к. будет предпринята попытка извлечения корня из отрицательного числа.

Дополнительно можно сделать проверку на то, является ли матрица положительно определенной, для этого нужно проверить тот факт, что все главные угловые миноры матрицы A положительные.

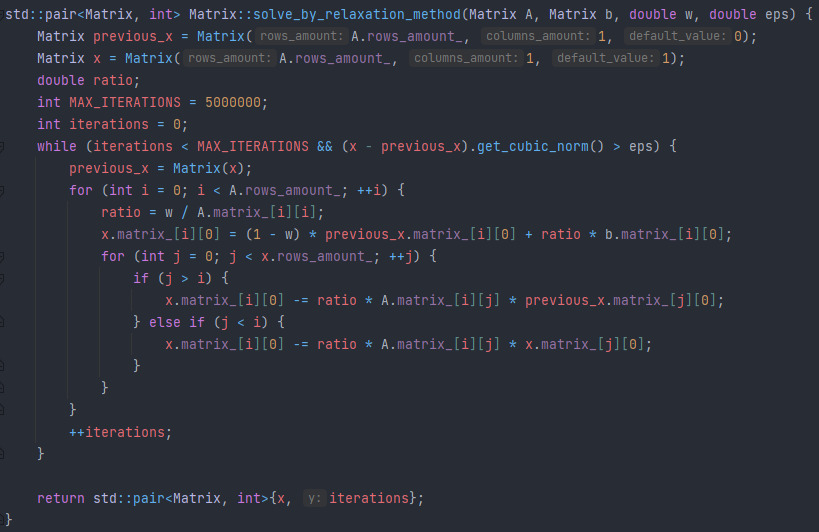


**

**Задание 7.**

Метод релаксации. Суть реализации состоит в том, чтобы запрограммировать следующую формулу:  


В качестве начального значения вектора я брал единичный вектор. В случае с генерируемыми матрицами, число обусловленности относительно мало(см. отчет задания 8 далее в этом документе, или в файле report.txt), присутствует диагональное преобладание, что обеспечивает сходимость итерационного процесса для метода Якоби и метода Гаусса-Зейделя. В случае с методом релаксации, итерационный процесс сходится, если матрица положительно определенная и симметричная, а w находится на интервале (0, 2). Согласно теореме 2.9, коспект Фалейчика.

**

Задание 8. Текст отчета будет тут, сам файл report.txt будет приложен отдельно в проекте.  
Текст отчета:  
Данные в нем заполняются при запуске, значения могут отличаться, т.к. матрицы генерируются.

=============== [Condition Numbers Stats] ===============  
Min condition number is 4.56351  
Max condition number is 4.90404  
Average condition number is 4.69056  
  
================ [Inverse Matrix Stats] =================  
Average time is 11379.9 mcs  
  
===================== [Gauss Method] ====================  
Min difference norm is 4.02567e-13  
Max difference norm is 6.26171e-11  
Average difference norm is 6.06553e-12  
Average time is 3271.96 mcs  
  
================== [LUP-decomposition] ==================  
Average time is 3141.4 mcs  
  
==================== [Solve LUx = b'] ===================  
Min difference norm is 4.02567e-13  
Max difference norm is 6.26171e-11  
Average difference norm is 6.06553e-12  
Average time is 189.62 mcs  
  
================ [Solve by SQRT method] =================  
Min difference norm is 3.10862e-15  
Max difference norm is 9.76996e-15  
Average difference norm is 5.9619e-15  
Average time is 3291.17 mcs  
  
============= [Solve by Relaxation method] ==============  
Min difference norm is 3.55271e-15  
Max difference norm is 9.76996e-15  
Average difference norm is 6.3749e-15  
Average time is 4454.28 mcs  
Min iterations number is 23  
Max iterations number is 24  
Average iterations number is 23.21

Для начала, матрицы с максимальным и минимальным числом обусловленности записаны в файл “min\_max\_condition\_matrices.txt”.

Пройдем по порядку, для того чтобы значения, которые мы получаем в ходе решения СЛАУ были достаточно точными, число обусловленности матрицы А должно быть близко к 1. В нашем случае, среднее значения числа обусловленности матриц – 4.69056, что вполне приемлемо, дальнейшие данные тому доказательство.

Самая тяжеловесная операция, не считая подсчет числа обусловленности, - нахождение обратной матрицы, в нашем случае среднее время выполнения примерно 11.4 мс. Точнее указано в самом отчете.

Далее интересно сравнить время решения методом Гаусса и время решения с помощью LUP разложения. Если рассматривать единоразовое решение СЛАУ, то нет смысла делать разложение, т.к. суммарно разложение и его решение равносильно решению методом Гаусса. Но если мы, например, исследуем влияние возмущения вектора, см. задание 10, не изменяя при этом матрицу А, то при готовом разложении решаем СЛАУ быстрее примерно в 17,26 раз, что является ощутимой разницей.

Далее заметим, что решение методом квадратного корня равноценно по времени выполнения с методом Гаусса(в рамках погрешности), но при этом дает большую точность. Разложение Холецкого также можно использовать с той же целью, что и LUP-разложение.

Насчет метода релаксации, в данном случае матрицы хорошо обусловлены, положительно определены и симметричны, подобран хороший параметр w, что по итогу дает быструю сходимость(<25 итераций) и относительно точное решение.

**Задание 9.**

В данном отчете организован форматированный вывод матриц, но в формате docx все отступы сломались, ширины листа не хватает, поэтому полный отчет можно посмотреть в файле task9+.txt

=============== [Condition Numbers Stats] ===============  
A1 condition number is 1.72336  
A2 condition number is 5.77524e+16  
  
================ [Inverse Matrix Stats] =================  
A1 inversion time is 0 mcs  
A2 inversion time is 0 mcs  
  
===================== [Gauss Method] ====================  
A1 solution difference norm is 0  
A1 solution time is 0 mcs  
A2 solution difference norm is 1.98968  
A2 solution time is 0 mcs  
  
============= [LUP-decomposition & solution] ============  
\_\_\_-------== ( A1 ) ==-------\_\_\_  
LUP-decomposition time is 0 mcs  
LUP-decomposition:  
  
LUP solution time is 0 mcs  
Solution difference norm is 9.32587e-15  
  
\_\_\_-------== ( A2 ) ==-------\_\_\_  
LUP-decomposition time is 0 mcs  
LUP-decomposition:  
L matrix:   
U matrix  
P matrix:   
  
LUP solution time is 0 mcs  
Solution difference norm is 1.98968  
  
================= [LDLT-decomposition] ==================  
\_\_\_-------== ( A1 ) ==-------\_\_\_  
L matrix:

D matrix:

L^T matrix:   
  
  
\_\_\_-------== ( A2 ) ==-------\_\_\_  
L matrix  
D matrix  
L^T matrix:  
  
  
================ [Solve by SQRT method] =================  
A1 \* x = b1 can't be solved by sqrt method: A1 == A1^T > 0 is not fit  
A2 solution difference norm is 1.46219  
A2 solution time is 0 mcs  
  
============= [Solve by Relaxation method] ==============  
A1 solution difference norm is 1.19904e-14  
A1 solution time is 0 mcs  
A1 solution iterations number is 20  
A2 solution difference norm is 2.49984  
A2 solution time is 7980 mcs  
A2 solution iterations number is 3126  
  
===== [Vector b Perturbation for Max Cond A Matrix] =====  
Iteration 1:   
Perturbation for 1e-06  
Practical error = 1.22169e-09  
Theoretical error = 4.61098e-09  
Practical error <= Theoretical error : True  
  
Iteration 2:   
Perturbation for 2e-06  
Practical error = 2.37441e-09  
Theoretical error = 9.22196e-09  
Practical error <= Theoretical error : True  
  
Iteration 3:   
Perturbation for 3e-06  
Practical error = 3.66508e-09  
Theoretical error = 1.38329e-08  
Practical error <= Theoretical error : True  
  
Iteration 4:   
Perturbation for 4e-06  
Practical error = 4.74881e-09  
Theoretical error = 1.84439e-08  
Practical error <= Theoretical error : True  
  
Iteration 5:   
Perturbation for 5e-06  
Practical error = 6.10847e-09  
Theoretical error = 2.30549e-08  
Practical error <= Theoretical error : True  
  
Iteration 6:   
Perturbation for 6e-06  
Practical error = 7.12322e-09  
Theoretical error = 2.76659e-08  
Practical error <= Theoretical error : True  
  
Iteration 7:   
Perturbation for 7e-06  
Practical error = 8.55187e-09  
Theoretical error = 3.22769e-08  
Practical error <= Theoretical error : True  
  
Iteration 8:   
Perturbation for 8e-06  
Practical error = 9.49763e-09  
Theoretical error = 3.68878e-08  
Practical error <= Theoretical error : True  
  
Iteration 9:   
Perturbation for 9e-06  
Practical error = 1.09953e-08  
Theoretical error = 4.14988e-08  
Practical error <= Theoretical error : True  
  
Iteration 10:   
Perturbation for 1e-05  
Practical error = 1.1872e-08  
Theoretical error = 4.61098e-08  
Practical error <= Theoretical error : True  
  
========= [Relaxation method A2 VS Max Cond A] ==========  
Diagrams for both matrices are in files A\_plot.jpeg and A2\_plot.jpeg

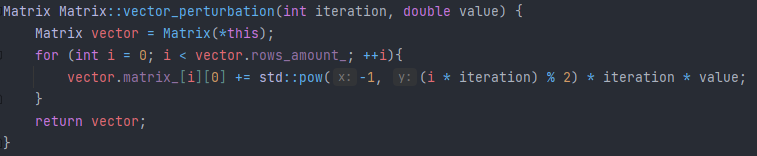
Для начала отметим, что матрица A1 – симметричная, хорошо обусловленная матрица, но при этом не является положительно определенной, что повлияет далее на возможность решения методом квадратного корня. А вот ситуация с матрицей А2 в разы хуже, там число обусловленности примерно 5.8e+16, т.е. матрица очень плохо обусловлена, но при этом симметрическая и положительно определенная. При большом числе обусловленности сильно снижается точность вычислений, что видно далее.

За счет малой размерности матриц решения СЛАУ, разложение и нахождение обратной матрицы выполняется быстро, таймер, реализованный с помощью std::chrono::high\_resolution\_clock не позволяет вывести более точное значение, но по моим подсчетам, обратная матрица для матрицы A2 находится за примерно 21 микросекунду.

Первое, что нас интересует, так это тот факт, что решения СЛАУ для матрицы А1 не уступают, а в некоторых случаях и превосходят точность решения СЛАУ для генерируемых матриц. Второе, для матрицы А1 нельзя использовать разложение, т.к. она не положительно определенная. Третье, точность решения СЛАУ для матрицы А2 очень низка, ни в одном из методов не опускается ниже 1(было произведено несколько прогонов для исключения неудачно сгенерированного точного решения).

**Задание 10.**

Это задание также отражено в отчете task9+.txt. Результаты будут продублированы в этом задании.

* Возмущение вектора для генерируемой матрицы с максимальным числом обусловленности. Восстановив матрицу, для нее сгенерировал точное решение x, подсчитал вектор b. Далее нужно было провести несколько возмущений вектора b и посчитать погрешность полученного решения x’.  
  Метод для возмущения вектора:  
    
  Данный метод меняет величину каждой координаты вектора на некоторое значение. После чего считалась абсолютная погрешность:  
    
  Результатами исследования являются:

===== [Vector b Perturbation for Max Cond A Matrix] =====  
Iteration 1:   
Perturbation for 1e-06  
Practical error = 1.22169e-09  
Theoretical error = 4.61098e-09  
Practical error <= Theoretical error : True  
  
Iteration 2:   
Perturbation for 2e-06  
Practical error = 2.37441e-09  
Theoretical error = 9.22196e-09  
Practical error <= Theoretical error : True  
  
Iteration 3:   
Perturbation for 3e-06  
Practical error = 3.66508e-09  
Theoretical error = 1.38329e-08  
Practical error <= Theoretical error : True  
  
Iteration 4:   
Perturbation for 4e-06  
Practical error = 4.74881e-09  
Theoretical error = 1.84439e-08  
Practical error <= Theoretical error : True  
  
Iteration 5:   
Perturbation for 5e-06  
Practical error = 6.10847e-09  
Theoretical error = 2.30549e-08  
Practical error <= Theoretical error : True  
  
Iteration 6:   
Perturbation for 6e-06  
Practical error = 7.12322e-09  
Theoretical error = 2.76659e-08  
Practical error <= Theoretical error : True  
  
Iteration 7:   
Perturbation for 7e-06  
Practical error = 8.55187e-09  
Theoretical error = 3.22769e-08  
Practical error <= Theoretical error : True  
  
Iteration 8:   
Perturbation for 8e-06  
Practical error = 9.49763e-09  
Theoretical error = 3.68878e-08  
Practical error <= Theoretical error : True  
  
Iteration 9:   
Perturbation for 9e-06  
Practical error = 1.09953e-08  
Theoretical error = 4.14988e-08  
Practical error <= Theoretical error : True  
  
Iteration 10:   
Perturbation for 1e-05  
Practical error = 1.1872e-08  
Theoretical error = 4.61098e-08  
Practical error <= Theoretical error : True

В случае начального смещение координат на 1e-6 все работает корректно, совпадает с условием и неравенству относительно теоритической погрешности, которая считается по формуле . Но ситуация ухудшается, если уменьшать величину сдвига. При начальном сдвиге на 1e-10 неравенство не выполняется. В остальном пока матрица имеет небольшое число обусловленности, ограничение теоритеческой погрешностью выполняется.

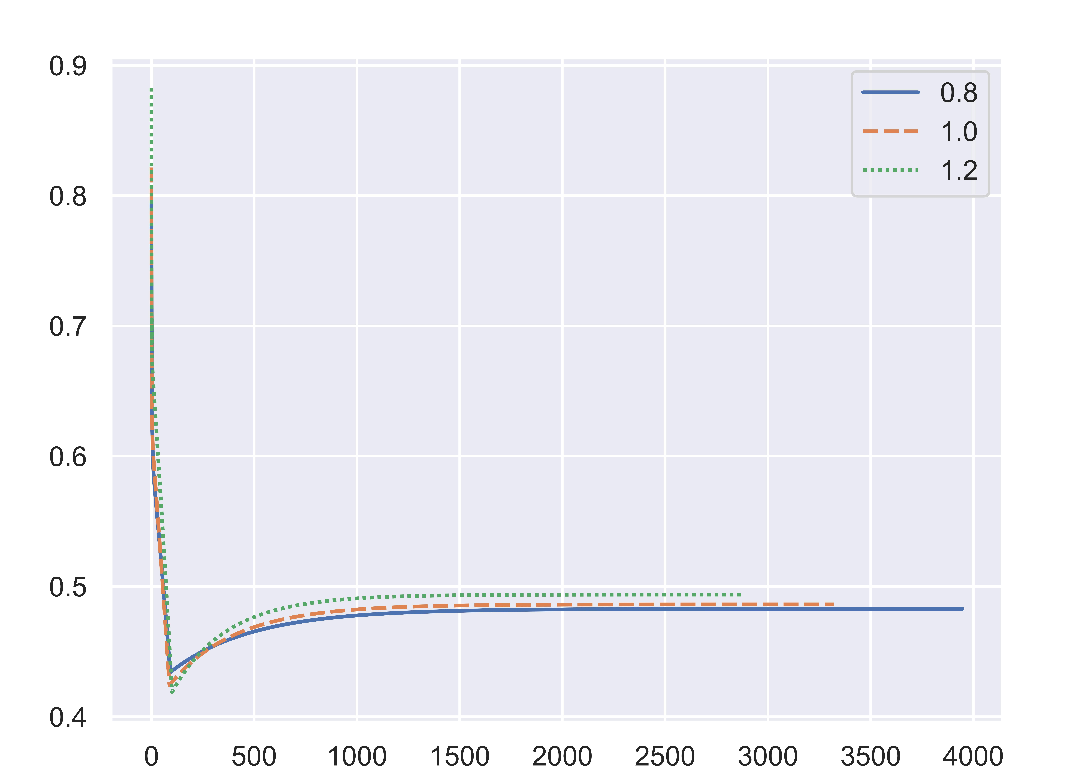
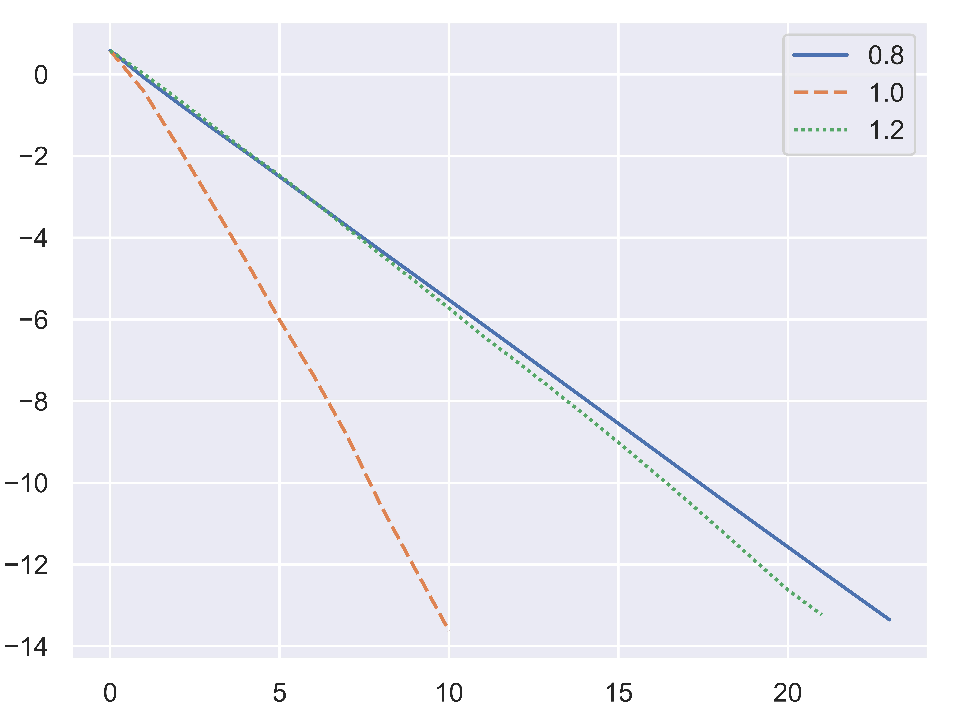
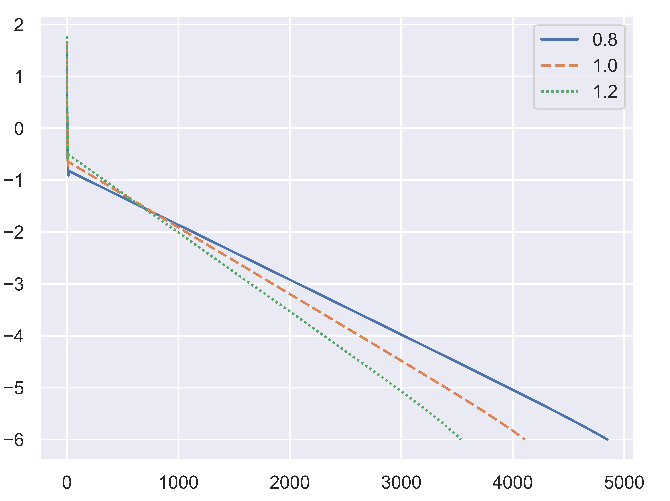
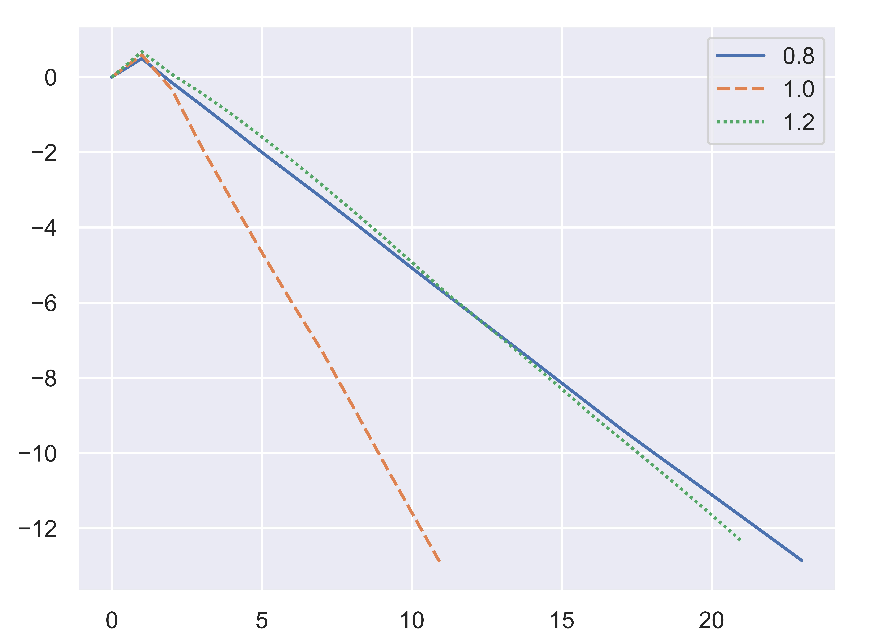
* Метод релаксации и диаграммы сходимости для матрицы А из прошлого пункта и матрицы А2. Для всех графиков верно следующее: ось Oy – логарифмированная по основанию 10 норма разности, Ox – количество итераций, в легенде указаны варианты параметра w. В качестве теста я решил сделать сначала график сходимости относительно искомого решения, а после – график сходимости относительно предыдущей итерации.  
    
  График сходимости относительно точного решения для матрицы А2:  
  

График сходимости относительно точного решения для матрицы А:  
  
График сходимости относительно решения на предыдущей итерации для матрицы А2:  
  
График сходимости относительно решения на предыдущей итерации для матрицы А:  


Для матрицы А2 была поставлена меньшая точность, т.к. решение сходилось очень медленно(5000000 итераций не хватало на точность выше 1e-9), соответственно точность была понижена до 1е-6.

В целом по графикам можно заметить, что, во-первых, решение СЛАУ для матрицы А2 медленно сходится, но сходится к ответу с большой погрешностью(из пункта 9 норма разности решений равна примерно 2.5, может меняться в зависимости от запуска).  
Для матрицы А наилучшим выбором параметра релаксации будет w = 1.0, т.к. в этом случае решение сходится быстрее всего(т.е. использован метод Гаусса-Зейделя).  
Для матрицы А2 ситуация разнится. Если смотреть на график сходимости относительно последней итерации, то быстрее сходится при w = 1.2, иначе большая точность обеспечивается при w = 0.8.