Arsham Mehrani
Project 1 report
CS3010

Summary:

In this project I have implemented the Jacobi and the Gauss-Seidel method to solving the system of linear equations. This project is implemented using Java. The functions, just like the handwritten way, are quite similar in that they both use iterative ways to solve the system of linear equations. With the only difference be that in Gauss-Seidel method we use each calculated X value in the next equations to solve and as a result reducing the iteration number by marginal amounts. This could be clearly observed by comparing the number of iterations each method produces and it is always the case that the gauss-Seidel method requires less iterations to converge.

Example:

The test case located in the input file has three equations. With desired error of 0.01 and starting values of X_1 X_2 $X_3 = 0$ we can observe that these equations will converge within 13 iterations using the Gauss-Seidel method and it simply will not converge within 50 iterations using the Jacobi method.

Code Explanation:

This implementation uses object-oriented programming to categorize each method into their own class. I use the main method to collect all the data required for each method and send that data to the constructor to then create an instance of each class to be solved.

Because these methods are so similar to each other, I used the same methods as the Jacobi method with a slight tweak to make the improvement that Gauss-Seidel has.

Each method first tries to solve each equation in terms of its respective X variable. For example, the first equation will solve with respect to X_1 and second equations will solve with respect to X_2 and etc. This is done by substituting all other X variable with the starting inputs and calculating the sum of each equations left hand side then bringing the sum to the other side of the equation and subtracting it from the value of b. In the Augmented matrix in the source code the b value is the last value in each row. Then simply dividing the result by the respective X variable's multiple will result in the value of X. From here the difference between the two methods is that the Jacobi method will store these back into the vector X and waits until all equations are solved then restarts the whole process with the new vector values. But the Gauss-Seidel will store that X value directly back to the X vector in the same loop to make sure that the next round of the loop will run using the newly calculated X value.

Conclusion:

Overall, after executing these methods for different test cases it was quite apparent that the Gauss-Seidel method has a big advantage over Jacobi. Even though there is only a small difference between how each method treats the calculated X values.