CMPT-439 Numerical Computation
Project 6

1. **All students may choose one of the following two options:**

   Follow a sequence of steps

   a)→c) → d)→e)→f) (you may get 100 points)

   Or

   b→c)→d)→e)→f) (you may get 115 points)

   a) (**40 points**) Design in any high-level language or in Matlab two functions to utilize the Jacobi and the Gauss-Seidel iterative methods for solving a system of ***n*** linear algebraic equations in ***n*** unknowns. Follow the algorithms from the class notes (Lecture 7) or from the textbook [1] (pp. 123-125). Use both approximate mean absolute error (MAE) and approximate root mean square error (RMSE) as stopping criteria creating two corresponding branches in both of your functions. Use vector [1, …, 1] or random numbers as the starting approximation for the unknowns.
   Your functions should accept as calling arguments:

   - an augmented matrix of the system;
   - a tolerance threshold;
   - a parameter determining a stopping criterion;
   - you may also use ***n*** (the number of equations and unknowns) as a parameter, if necessary.

   Your function shall test whether a matrix of a system to be solved is a diagonally dominant one and if not, prompt the user to transform it.

   b) **(55 points**) Design **two functions** to utilize the Jacobi and the Gauss-Seidel iterative methods, respectively. Follow the algorithms from the class notes (Lecture 7) or from the textbook [1] (pp. 123-125). Use vector [1, …, 1] or random numbers as the starting approximation for the unknowns.
   Use the following four stopping criteria in both of your functions:
   - approximate mean absolute error (MAE)
   - approximate root mean square error (RMSE)
   - true mean absolute error (MAE)
   - true root mean square error (RMSE).

   Your functions should accept as calling arguments:

   - an augmented matrix of the system;
   - a tolerance threshold;
   - a parameter determining a stopping criterion;
   - you may also use ***n*** (the number of equations and unknowns) as a parameter, if necessary.

Your function shall test whether a matrix of a system to be solved is a diagonally dominant one and if not, prompt the user to transform it.

c) (25 points) Solve the following two systems of linear equations given by their augmented matrices using all stopping criteria implemented in your functions utilizing the Jacobi and Gauss-Seidel iterative methods. Use 0.001 as a tolerance threshold

d) $\left[\begin{array}{ccc|c} 3 & 1 & -4 & 7 \\ -2 & 3 & 1 & -5 \\ 2 & 0 & 5 & 10 \end{array}\right]$

e) $\left[\begin{array}{ccc|c} 1 & -2 & 4 & 6 \\ 8 & -3 & 2 & 2 \\ -1 & 10 & 2 & 4 \end{array}\right]$

Hint (important): One of these systems, while it is not a diagonally dominant one, can be solved using iterative methods without transformation of its matrix (to transform its matrix, only complete pivoting may help, but this is not necessary, as this system is an exception). Another system also is not a diagonally dominant one, and it can be solved only after its matrix is transformed using partial pivoting.

f) (**5 points**) Test roots, which you found plugging them in all equations of the corresponding system

2. (**10 points**) Ask a generative AI tool (ChatGPT or any other) to create functions utilizing Jacobi and Gauss-Seidel methods in the same programming language, which you used, meeting the same requirements as your functions had to meet (in particular, utilizing the same stopping criteria).
Evaluate differences between your code and generative AI code. **Is an AI-generated code correct? Does it meet requirements?**

3. (**15 points**) Solve the same two systems (1d) and (1e) of linear equations using AI-generated code. Use 0.001 as a tolerance threshold

4. (**Optional - extra credit 15 points**). Design a function transforming a matrix of the system to a diagonally dominant one. You may focus only on partial pivoting, not involving complete pivoting (<u>**extra 10 points** will be added if you utilize complete pivoting</u>).
Use this function in your assignment calling it at the beginning of both functions utilizing Jacobi and Gauss-Zeidel methods and checking how it works.

5. (**5 points**) Write a brief report presenting your solution and demonstrating your understanding of this solution.

6. Turn in your source code and a report presenting solutions.

Reference

[1] Gerald, C.F. and Wheatley, P.O., Applied Numerical Analysis, 7th Edition, Pearson, 2004