MAE384, Spring 2022 Homework #2

A statement of collaboration is required. If there is no collaboration, write "No collaboration". For the computation of *condition number*, you are allowed to use Matlab built-in function `inv`. Please <u>include computer codes</u> in your work.

Problem 1 (6 points)
A system of linear equations is given:

$$\begin{pmatrix} 7 & 4 & 2 & 0.5 \\ 0.5 & 3 & 1 & 0.5 \\ 1.5 & 0.5 & 5 & 2 \\ 1 & 2 & 0.5 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 23.5 \\ 12 \\ 15 \\ 12 \end{pmatrix} . \qquad Eq.(1)$$

Symbolically, we write *Eq*. (1) as

$$\boldsymbol{A}\,\vec{\boldsymbol{x}} = \vec{\boldsymbol{b}} \ , \qquad Eq.(2)$$

which also gives the definitions of the matrix $\boldsymbol{A}$ and vectors $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{b}}$. For this system, the exact solution is known to be

$$\vec{\boldsymbol{x}}_S = \begin{pmatrix} 1 \\ 3 \\ 2 \\ 1 \end{pmatrix} .$$

Let us also recall a few relevant definitions. Given $\vec{\boldsymbol{x}}_N$ as a numerical solution, the **relative true error** (*RTE*) is defined as

$$RTE = \frac{||\vec{\boldsymbol{e}}||}{||\vec{\boldsymbol{x}}_S||} \ , \qquad Eq.(3)$$

where $\vec{\boldsymbol{e}} = \vec{\boldsymbol{x}}_N - \vec{\boldsymbol{x}}_S$ is the vector of absolute error. The **relative residual** (*RRS*) is defined as

$$RRS = \frac{||\vec{\boldsymbol{r}}||}{||\vec{\boldsymbol{b}}||} \ , \qquad Eq.(4)$$

where $\vec{\boldsymbol{r}} = \boldsymbol{A}\vec{\boldsymbol{x}}_N - \vec{\boldsymbol{b}}$ is the residual vector. The key conclusion of the discussion (in Lecture 11) on **condition number**, *C*, is summarized by the inequality,

$$\frac{1}{C} * (RRS) \le (RTE) \le C * (RRS) \ . \qquad Eq.(5)$$

Conceptually, *Eq*. (5) provides the upper and lower bounds of *RTE* from *RRS* and *C*. The condition number itself is defined by

$$C = ||\boldsymbol{A}||\ ||\boldsymbol{A}^{-1}|| \ . \qquad Eq.(6)$$

In *Eqs*. (3)-(6), Euclidean norm is used for a vector and Euclidean 2-norm (or "Frobenius norm") is used for a matrix. Let x_j be the j-th element of a vector $\vec{\boldsymbol{x}}$ with N elements, then

$$\|\vec{\boldsymbol{x}}\| = \sqrt{\sum_{j=1}^{N} (x_j)^2} \ .$$

Let A_{ij} be the element of an N x N matrix $\boldsymbol{A}$ at the i-th row and j-th column, then

$$\|\boldsymbol{A}\| = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} (A_{ij})^2} \ .$$

(a) Solve the system in *Eq*. (1) with two methods: (i) *Jacobi iterative method*, using (x_1, x_2, x_3, x_4) = (0,0,0,0) as the initial guess to perform 20 iterations. (ii) *Gauss-Seidel iterative method*, using (x_2, x_3, x_4) = (0,0,0) as the initial guess to perform 20 iterations. (Note that the *G-S* method does not require an initial guess for x_1.) Here, one iteration means completing the update of all x_1, x_2, x_3, and x_4 once. From the results, make a plot of the relative true error (*RTE*) vs. the number of iterations for both methods. Please collect the two curves of *RTE* in the same plot for a clear comparison. Since *RTE* decreases dramatically with the number of iterations, for clarity we require that the actual plot be made by taking the 10-based logarithm of *RTE*. In other words, the deliverable is the line plot of $log_{10}(RTE)$ vs. the number of iterations. (In Matlab, the built-in function for 10-based log is `log10`.)

(b) Compute the condition number, C, for the system in *Eq*. (1). Using the result from the case in Part (a) with *Gauss-Seidel method*, make a plot with 3 curves (must be collected in the same plot): The first is the curve of $log_{10}(RTE)$ vs. the number of iterations, as recycled from Part (a). (Despite the redundancy, for clarity we ask that a new plot be made for Part (b). Do not merge the deliverables in Part (a) and (b) into a single plot.) The other two are the curves for $log_{10}(RRS*C)$ and $log_{10}(RRS*C^{-1})$, vs. the number of iterations. This plot serves the purpose of demonstrating the relations in *Eq*. (5). Namely, $RRS*C$ and $RRS*C^{-1}$ provide the upper bound and lower bound of *RTE*. (If C is close to 1, these bounds are "tight" and *RRS* is a useful proxy of *RTE*. If C is very large, *RTE* is detached from *RRS*.)