NE 155 Spring 2017
Homework #5 Due March 22, 2017

SSID:

On homework:

- If you work with anyone else, document what you worked on together.
- Show your work.
- Always clearly label plots (axis labels, a title, and a legend if applicable).
- Homework should be done "by hand" (i.e. not with a numerical program such as MATLAB, Python, or Wolfram Alpha) unless otherwise specified. You may use a numerical program to check your work.
- If you use a numerical program to solve a problem, submit the associated code, input, and output (email submission is fine).
- If using Python, be aware of `copy` vs. `deep copy`: https://docs.python.org/2/library/copy.html

1. (10 points) For $n = 100$ we will use this tridiagonal system of equations

$$\begin{aligned} 2x_0 - x_1 &= 0 \\ -x_{j-1} + 2x_j - x_{j+1} &= j\,, \qquad j = 1, \ldots, n-2 \\ -x_{n-2} + 2x_{n-1} &= n-1 \end{aligned}$$

in a few different ways. You may use Python, MATLAB or another package or code language of your choice. Note: `NumPy` and `SciPy` are libraries that can be imported into Python and would be useful for this assignment.

   (a) (2.5 points) Use built in Python or MATLAB commands to construct $\mathbf{A}$ and $\vec{b}$.

   (b) (2.5 points) What is the condition number of $\mathbf{A}$?

   (c) (2.5 points) Solve this problem by explicitly inverting $\mathbf{A}$ and multiplying $\vec{b}$.

   (d) (2.5 points) Use `scipy.linalg.solve` (Python) or the backslash operator (MATLAB) to solve the system.

   Plot both your explicit (c) and numerical (d) solutions on the same plot. Include axis labels, a title, and a legend.

2. (5 points) Discuss the significance of the spectral radius for the iterative solution of $\mathbf{A}\vec{x} = \vec{b}$, including how it is used to determine convergence and how it is related to rate of convergence.

3. (10 points) By hand, find the first two iterations (report the vectors $\vec{x}^{(1)}$ and $\vec{x}^{(2)}$) of the

   - Jacobi method
   - Gauss Seidel method
   - SOR (with $\omega = 1.1$) method

   for the following system of equations. Use $\vec{x}^{(0)} = \vec{0}$. Briefly compare the results between methods.

$$\text{(b).} \quad \begin{aligned} 2x_1 + -2x_2 + 1x_3 + 1x_4 &= 0.8 \\ 0x_1 + -3x_2 + 0.5x_3 + 1x_4 &= -6.6 \\ 0x_1 + 0x_2 + 5x_3 + -1x_4 &= 4.5 \\ 0x_1 + 0x_2 + 0x_3 + 2x_4 &= 2 \end{aligned}$$

4. (30 points) We will use the following system of n equations:

$$\mathbf{A}\vec{x} \equiv \begin{pmatrix} 3 & -1 & 0 & \cdots & 0 \\ -1 & 3 & -1 & \ddots & \vdots \\ 0 & -1 & 3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 3 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} 100 \\ 100 \\ 100 \\ \vdots \\ 100 \end{pmatrix} \equiv \vec{b}$$

   Write a program to implement the

   (a) Jacobi method
   (b) Gauss Seidel method
   (c) SOR method

   for a matrix with n unknowns. Turn in your source code electronically; include instructions for how to run it, input files, etc. if necessary.

   Solve the above system of equations with each program.
   Use $\omega = 1.15$ for SOR; use $\vec{x}^{(0)} = \vec{0}$ and $n = 5$.
   Print the solution vector from each method converged to an **absolute** tolerance of 10^{-6}.
   Indicate the number of iterations required to meet this tolerance for each method.

5. (20 points) Use the programs you just wrote with the same matrix and using the same settings to answer the following.

   (a) (10 points) How many iterations are required for each method to reach the stopping criterion (*relative error*):

   $$\frac{||x^{(k+1)} - x^{(k)}||}{||x^{(k+1)}||} < \epsilon$$

   for $\epsilon = 10^{-6}$ and $\epsilon = 10^{-8}$?
   Also:

   - For each method, how does the absolute error (from the previous question) with $\epsilon = 10^{-6}$ compare to the relative error?
   - Which method required the fewest iterations?
   - What do you observe about reaching a tighter convergence tolerance?

   (b) (10 points) Perform an experiment to determine ω_{opt} for SOR. Explain your procedure and include the results.

BONUS: submit your code by providing read/clone access to an online version control repository where your code is stored (e.g. github or bitbucket). If you don't know what that means and want to learn about it, come talk to me or check out resources here: http://software-carpentry.org/lessons.html.

NOTE: If you are unsure if your code is working properly you can check with me before submitting as that is a big part of this homework.