Homework 4.

Amath 301
Beginning Scientific Computing

Due: 4/25/23 at 11:59pm to Gradescope

Directions:

For the **written exercises** of the homework, please do the following:

- Write your name, student number, and section at the top of your written exercises. A half point will be deducted if these are not included.
- Complete all written exercises to the best of your ability and as neatly as possible.
- For those who are writing their answers by hand, please scan your work and save it as a pdf file.
- You are encouraged to type your written homework using LaTeX. Doing so will earn you a half bonus point. Check out my LaTeX beginner document and overleaf.com if you are new to LaTeX.

For the **coding exercises** of the homework, please do the following:

- Create a MATLAB script titled `hw4.m` and divide your script into sections according to the various coding exercises.
- Make sure your final answers in each coding exercise are assigned to the correct variable names.
- Publish your script as a pdf file. You can do this by typing `publish('hw4.m','pdf')` in the Command Window.
- Attach your published code to the end of your written exercises. Submit your entire homework assignment (written + coding exercises) to Gradescope.

Pro-Tips:

- You can check many of your answers to the written exercises of this homework by having MATLAB, Wolfram Alpha, or Chat GPT do the calculations for you.
- Teamwork makes the dream work, but please make sure you write up your solutions.
- Don't wait until the last minute. ☺

Written Exercises

(10 points total)

Exercise 1. (Component Skill 4.1)

Consider the following ridiculously simple linear system:

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

a. What is the solution of this linear system? You don't have to show your work.

b. Set up a Neumann iteration with initial guess $\mathbf{x}_0 = (0,0)^T$. Calculate $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$. Show your work.

c. Based on your calculations in (b), what will $\mathbf{x}_k$ equal for $k > 0$ when (i) k is odd and (ii) k is even? You don't have to show your work.

d. Based on your answer to (c), does the Neumann iteration converge?

e. Let M represent the matrix used in your Neumann iteration. Does $M^n \to 0$ as $n \to \infty$? What does this tell you about the convergence of the Neumann iteration method? Is this consistent with your answer to (d)?

Exercise 2. (Component Skill 4.2)

For each linear system, decide whether Jacobi iteration is guaranteed or not guaranteed to converge to the solution. Justify each of your decisions in a sentence or two.

a. $\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

b. $\begin{pmatrix} -3 & 1 & 0 \\ 1 & -3 & 1 \\ 0 & 1 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

c. $\begin{pmatrix} 1 & 3 & 0 \\ 3 & 1 & 3 \\ 0 & 3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

d. $\begin{pmatrix} 1 & -3 & 0 \\ -3 & 1 & -3 \\ 0 & -3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

e. $\begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

Coding Exercises

(10 points total)

Exercise 1. (Component Skill 4.2)

The following MATLAB function file executes Jacobi iteration:

```
function[x1] = JacobiHW4(A,b,x0,N)
% INPUTS:     A is a nxn coefficient matrix
%             b is a nx1 column vector of knowns
%             x0 is the nx1 initial guess of the solution
%             N is the number of iterations
% OUTPUTS:    x1 is the nx1 solution

D = diag(diag(A));
L = tril(A,-1);
U = triu(A,1);

for j = 1:N
    x1 = D \ (-(L+U)*x0 + b);
    x0 = x1;
end

end
```

a. Use the function above to solve

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{1}$$

by Jacobi iteration. Be sure to put the code for the function at the very bottom of your `hw4` script file. Take the initial guess to be $\mathbf{x}_0 = (0, 0, 0)^T$ and iterate $N = 10$ times. Assign the value of $\mathbf{x}_{10}$ to the variable `A1`. Use `format long` to show as many decimal places as possible.

b. What is the exact solution of (1) according to the MATLAB backslash command? Assign this solution to the variable `A2`.

c. Let $\mathbf{x}_*$ be the exact solution of (1). Define the error of the kth Jacobi iterate $\mathbf{x}_k$ as the maximum difference in absolute value between corresponding components of $\mathbf{x}_k$ and $\mathbf{x}_*$:

$$\mathcal{E} = \max_{1 \le i \le 3} \{ |\mathbf{x}_{k,i} - \mathbf{x}_{*,i}| \} .$$

Assign the variable `A3` to the value of $\mathcal{E}$ for your $\mathbf{x}_{10}$ computed in (a). Use `format long` to show as many decimal places as possible.

Answers:

```
A1 =

   0.285559450625751
   0.142779725312876
   0.285559450625751


A2 =

   0.285714285714286
   0.142857142857143
   0.285714285714286


A3 =

     1.548350885341998e-04
```

Exercise 2. (Component Skill 4.3)

Modify the Jacobi function file given in Coding Exercise 1 to create a new function file in MATLAB that executes Gauss-Seidel iteration. Call your new function file `GaussSeidelHW4`. Retain the same inputs and output as before.

a. Use your new function to solve

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{2}$$

by Gauss-Seidel iteration. Be sure to put the code for your function at the very bottom of your `hw4` script file. Take the initial guess to be $\mathbf{x}_0 = (0,0,0)^T$ and iterate $N = 10$ times. Assign the value of $\mathbf{x}_{10}$ to the variable `A4`. Use `format long` to show as many decimal places as possible.

b. Let $\mathbf{x}_*$ be the exact solution of (2). Define the error of the kth Gauss-Seidel iterate $\mathbf{x}_k$ as the maximum difference in absolute value between corresponding components of $\mathbf{x}_k$ and $\mathbf{x}_*$:

$$\mathcal{E} = \max_{1 \le i \le 3} \left\{ |\mathbf{x}_{k,i} - \mathbf{x}_{*,i}| \right\}.$$

Assign the variable `A5` to the value of $\mathcal{E}$ for your $\mathbf{x}_{10}$ computed in (a). How does this error compare to your error in Coding Exercise 1(c)? Use `format long` to show as many decimal places as possible.

Answers:

```
A4 =

   0.285714128385537
   0.142857247742976
   0.285714250752341


A5 =
```

[redacted]

```
     1.573287491951625e-07
```

Exercise 3. (Component Skill 4.4)

Modify the Jacobi function file given in Written Exercise 3 to create a new function file in MATLAB that executes SOR iteration. Call your new function file `SORHW4`. Retain the same inputs as before, but add one more input ω for the relaxation parameter. Keep the output the same.

a. Use your new function to solve

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{3}$$

by SOR iteration with $\omega = 1.09$. Be sure to put the code for your function at the very bottom of your `hw4` script file. Take the initial guess to be $\mathbf{x}_0 = (0, 0, 0)^T$ and iterate $N = 10$ times. Assign the value of $\mathbf{x}_{10}$ to the variable `A6`. Use `format long` to show as many decimal places as possible.

Remark: For this linear system, $\omega = 1.09$ is very close to the optimal choice.

b. Let $\mathbf{x}_*$ be the exact solution of (3). Define the error of the kth SOR iterate $\mathbf{x}_k$ as the maximum difference in absolute value between corresponding components of $\mathbf{x}_k$ and $\mathbf{x}_*$:

$$\mathcal{E} = \max_{1 \le i \le 3} \left\{ |\mathbf{x}_{k,i} - \mathbf{x}_{*,i}| \right\}.$$

Assign the variable `A7` to the value of $\mathcal{E}$ for your $\mathbf{x}_{10}$ computed in (a). How does this error compare to your errors in Coding Exercises 1(c) and 2(b)? Use `format long` to show as many decimal places as possible.

Answers:

```
A6 =

   0.285714285711890
   0.142857142816904
   0.285714285716453


A7 =

     4.023845145972871e-11
```

Exercise 4. (Component Skill 4.2)

Rather than specify the number of Jacobi iterations, we could specify a desired error threshold to achieve instead. In particular, if the kth Jacobi iterate is within this specified threshold of error of the true solution, we terminate our Jacobi iterations. It's a great idea, but the problem is that we usually don't know the exact solution of our linear system and, hence, can't exactly measure the error of our kth iteration.

What we do instead is we keep track of what's called the Cauchy error. This is the maximum difference in absolute value between two successive Jacobi iterations:

$$\mathcal{C} = \max_i |\mathbf{x}_{k,i} - \mathbf{x}_{k-1,i}|.$$

Here, the index i keeps track of the components of our iterates. If the Cauchy error is decreasing, our iterates must be converging closer and closer to the exact solution of the linear system.

The following represents a MATLAB function file that executes Jacobi iteration until the Cauchy error falls below a certain threshold:

```
function[x1,N] = JacobiCauchyHW4(A,b,x0,threshold)
% INPUTS:     A is a nxn coefficient matrix
%             b is a nx1 column vector of knowns
%             x0 is the nx1 initial guess of the solution
%             threshold is the Cauchy error threshold
% OUTPUTS:    x1 is the nx1 solution
%             N is the number of iterations

D = diag(diag(A));
L = tril(A,-1);
U = triu(A,1);

x1 = D \ (-(L+U)*x0 + b);
err = max(abs(x1-x0));
N = 1;
```

```

while err > threshold
    xtemp = x1;
    x1 = D \ (-(L+U)*x1 + b);
    x0 = xtemp;
    err = max(abs(x1-x0));
    N = N+1;
end

end
```

a. Use the function above to solve

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

by Jacobi iteration. Be sure to put the code for the function above at the very bottom of your `hw4` script file. Take the initial guess to be $\mathbf{x}_0 = (0,0,0)^T$ and the error threshold to be 10^{-10}. Assign the number of iterations necessary to achieve this error threshold to the variable `A8`.

b. Modify the function above to solve

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

by Gauss-Seidel iteration. Call your new function `GaussSeidelCauchyHW4`. Be sure to put the code for your function at the very bottom of your `hw4` script file. Take the initial guess to be $\mathbf{x}_0 = (0,0,0)^T$ and the error threshold to be 10^{-10}. Assign the number of iterations necessary to achieve this error threshold to the variable `A9`. Compare your answer with (a).

c. Modify the function above to solve

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

by SOR iteration. Call your new function `SORCauchyHW4`. Make sure you add an extra input for the relaxation parameter ω. Be sure to put the code for your function at the very bottom of your `hw4` script file. Take the initial guess to be $\mathbf{x}_0 = (0,0,0)^T$, the relaxation parameter to be $\omega = 1.09$, and the error threshold to be 10^{-10}. Assign the number of iterations necessary to achieve this error threshold to the variable `A10`. Compare your answer with (a) and (b).

Answers:

```
A8 =

    31


A9 =

    15
```

```
A10 =

    11
```