NENG 685 Fall 2017
HW 5 Due Dec. 4, 2017

Name:

On homework:

- Well organized and documented work scores better. If I cannot figure out what is going on, then I am less likely to "intuit" what you intended, and the score will be reflective of this fact.
- 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.
- ***I should not have to run your code to see your answers.*** The attached code is an additional form of feedback for me and a method to give partial credit. If you want full credit, then include the outputs (plots, tables, answers, etc.) in your write-up.

Do not write in the table to the right.

Problem	Points	Score
1	5	
2	25	
3	20	
4	10	
5	20	
6	15	
Total:	95	

1. (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.

2. (25 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 final error and the number of iterations required to meet this tolerance for each method.

3. (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 number of iterations using 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.

4. (10 points) Harness your knowledge from your differential equations class to analytically solve the fixed-source diffusion equation (assuming D and Σ_a are constant):

$$-D\frac{d}{dx}\frac{d\phi(x)}{dx} + \Sigma_a\phi(x) = S(x)$$
$$\text{Boundary Conditions: } \phi(\pm a) = 0$$

in the following situations:

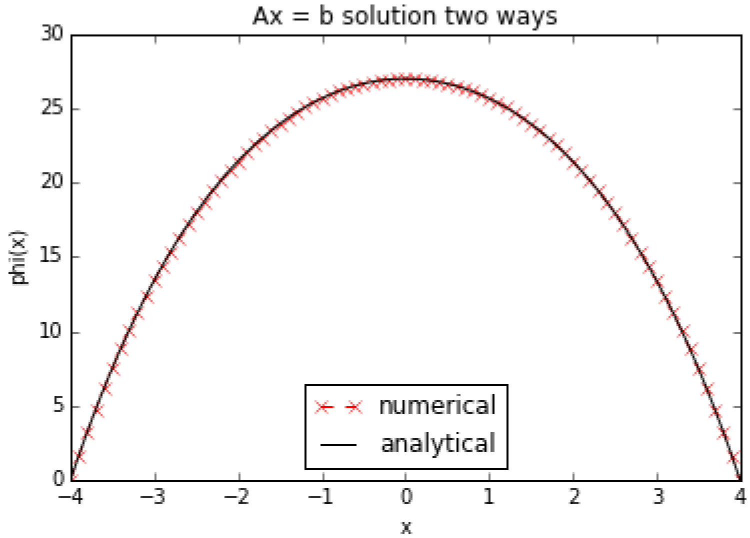
$S(x) = S_0$ (a constant) for $x \in [-a, a]$

Hint: You can deduce an additional boundary condition that may make things slightly simpler.

5. (20 points) Numerically solve the fixed-source diffusion equation as described in Question 4 using the finite *difference method* for discretization of the spatial variable and *Gaussian elimination* (a.k.a. the Thomas algorithm; note that you will have a tridiagonal system to solve) for solving the system of linear algebraic equations.

Use the following parameters:

- $a = 4$ cm,
- $D = 1$ cm,
- $\Sigma_a = 0.2$ cm^{-1},
- $S = 8$ n/(cm^3 s), and
- $h = 0.1$ cm.



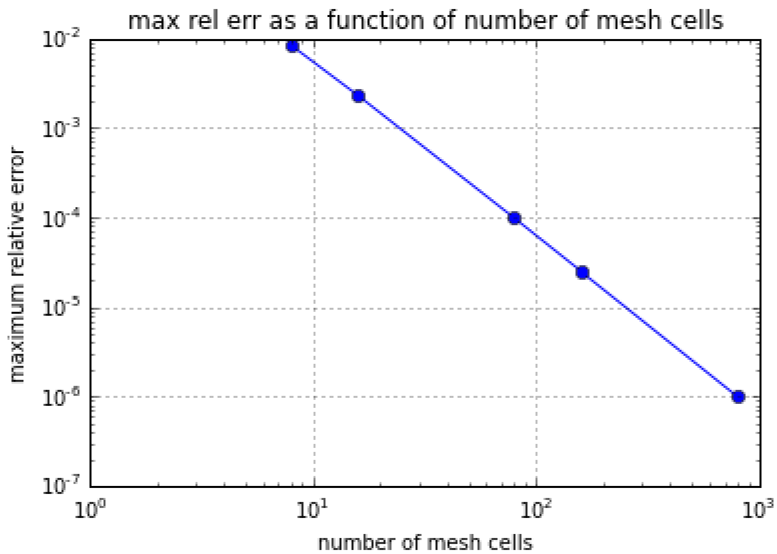
Plot the solution from $x = -a$ to $x = a$. Compare your answer (in terms of max error) to your solution from Question 4.

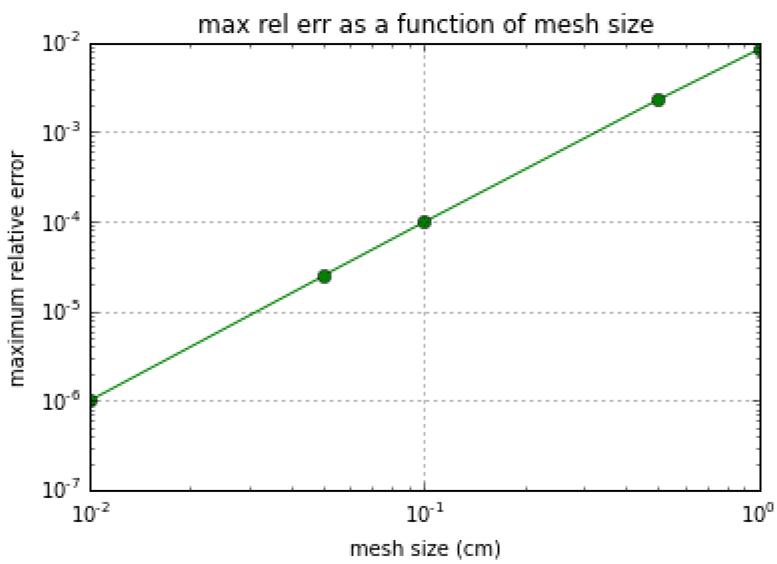
6. (15 points) Investigate how well your numerical solution approximates the analytical solution by computing ϕ_i for various constant mesh sizes: h = *1 cm, 0.5 cm, 0.1 cm, 0.05 cm, 0.01 cm*.

   For each mesh length calculate the relative error between your numerical and analytical solutions. Plot the maximum relative error as a function of total number of meshes for each case. What can you conclude about the relationship between the maximum error and the total number of meshes? What is the order of convergence?

BONUS (5 points): submit your code by providing read/clone access to an online version control repository where your code is stored (e.g. github or bitbucket).

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.


max rel err as a function of number of mesh cells
maximum relative error
number of mesh cells
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{0}
10^{1}
10^{2}
10^{3}


max rel err as a function of mesh size
maximum relative error
mesh size (cm)
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-2}
10^{-1}
10^{0}