PROGRAMMING ASSIGNMENT 5

Due: Thursday, November 12 by 11am. Upload *one* Jupyter notebook to Gradescope.

1. SOLVING SYSTEMS OF EQUATIONS

Numerically solve the following system of equations using the Jacobi (relaxation) method:

$$4x_1 + x_2 - x_3 = 3$$

$$x_1 + 6x_2 - 2x_3 + x_4 - x_5 = -6$$

$$x_2 + 5x_3 - x_5 + x_6 = -5$$

$$2x_2 + 5x_4 - x_5 - x_7 - x_8 = 0$$

$$-x_3 - x_4 + 6x_5 - x_6 - x_8 = 12$$

$$-x_3 - x_5 + 5x_6 = -12$$

$$-x_4 + 4x_7 - x_8 = -2$$

$$-x_4 - x_5 - x_7 + 5x_8 = 2$$

Use a convergence limit of 10^{-5}, and *verify* that your numerical solution works to solve the system of equations.

Grading Rubric: Program has required features described in prompt (25%), graphs show correct behavior and are easy to interpret (25%), written response is factually correct (25%) and uses proper grammar and spelling (15%), code compiles without errors and reproduces all output (10%)

2. ENERGY IN A CAPACITOR

Solve for the potential and electric field in the prism geometry in Figure 5.4 of your textbook. Replicate the plots in Figure 5.5: show the equipotential contours, a perspective plot of the potential, and a plot of the electric field. Assume that the inner conductor is held at $V = 2$, and is a square of length 1 on a side; assume the boundary condition $V = 0$ at the outer edge of the prism, which is also a square of length 4 on a side. In other words, the gray shaded box in Fig. 5.4 should span from -0.5 to +0.5 in along both the x and y axes, and the solid black outline should span from -2 to 2. Use a grid spacing of 0.1.

Hints:

- You may find it useful to establish two arrays for this problem: one to hold the potential at each point in your grid, and one to hold the boundary conditions. Structure your code to check the boundary condition array, and if a given "flag" is set, do not update your potential array.
- To make a 3D surface plot of the potential, you may find the tool `np.meshgrid` useful, and I highly recommend you look at the "surface3d_demo" example in the matplotlib gallery.
- Look up the pyplot tool *quiver* to make the electric field diagram (e.g., you can call it with `plt.quiver`)

Extra Credit Opportunity: Use the symmetry of the problem to only solve for V in *half* of the first quadrant (thereby speeding up your code by a factor of 8). **+4 points for correct implementation.**

Grading Rubric: Program has required features described in prompt (25%), graphs show correct behavior and are easy to interpret (25%), written response is factually correct (25%) and uses proper grammar and spelling (15%), code compiles without errors and reproduces all output (10%)

3. A Wave on a String

Simulate a wave propagating along a 1D string that has a length $L = 20$. The ends of the string (at $x = 0$ and $x = L$) should remain fixed at 0 for the duration of the simulation ($t_{\max} = 30$). Assume that the wave initially follows the "pulse" profile

$$y_0(0) = e^{-k(x-x_0)^2}$$

where x_0 is the initial starting location of the pulse along the x-axis, and $k = 2$ is a factor that determines the width of the pulse. Start the wave at the center of the string (in other words, $x_0 = 10$). Assume that the wave speed is $c = 1$. Hint: You should select values of Δx and Δt such that $(c\,\Delta t/\Delta x)^2 \leq 0.25$.

Show time evolution of the string[1]. Specifically show that wave correctly reflects off boundaries, and exhibits constructive or destructive interference (as appropriate), and demonstrate that energy is conserved in your simulation. Include a few grammatically correct sentences describing how the plots you are presenting support your assertions (e.g., "This plot shows that energy is conserved because…").

Grading Rubric: Program has required features described in prompt (25%), graphs show correct behavior and are easy to interpret (25%), written response is factually correct (25%) and uses proper grammar and spelling (15%), code compiles without errors and reproduces all output (10%)

4. Testing the Limits Your Simulation

Re-run your simulation from #3, but this time vary your value of $(c\,\Delta t/\Delta x)^2$ to range from 0.2 to 1.0 in steps of 0.1, and measure the percent change in energy during the simulation. Make a graph of $\%\Delta E$ vs. $(c\,\Delta t/\Delta x)^2$. Do you observe a power law dependence on $(c\,\Delta t/\Delta x)^2$? A linear dependence? Exponential? Something else? Describe in a few grammatically correct sentences your conclusions, and discuss the importance of the $(c\,\Delta t/\Delta x)^2$ parameter in this type of simulation.

Grading Rubric: Program has required features described in prompt (25%), graphs show correct behavior and are easy to interpret (25%), written response is factually correct (25%) and uses proper grammar and spelling (15%), code compiles without errors and reproduces all output (10%)

5. Ripples in a Pond (Simulating the Motion of a 2D Wave)

---

[1] If you want to make an animation of your results, check out this blog post:
http://louistiao.me/posts/notebooks/embedding-matplotlib-animations-in-jupyter-notebooks/
I put the sample code in a notebook that is available on Blackboard.

We're now going to take what we learned in problem 3 about simulating a 1D wave and extend it to two dimensions. Note that, in the 2D case, the wave equation becomes:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

We derived the 1D version of equation 6.6 in your text in class; apply the finite difference expression for second partial derivatives to the 2D wave equation and derive the 2D analog equation of equation 6.6. *You must show your work* in your notebook using LaTeX typesetting.

We will use similar boundary and initial conditions as those in problem 3. Assume a square box with a width of 20 on each side ($L = L_x = L_y = 20$). The edges of the box are to remain fixed at 0 for the duration of the simulation which runs from $t = 0 \rightarrow 30$. Make your life easy and assume $\Delta x = \Delta y$. The wave will initially have a 2D Gaussian pulse profile:

$$z_0(0) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-[(x-x_0)^2/(2\sigma_x)+(y-y_0)^2/(2\sigma_y)]}$$

Where $\sigma_x = \sigma_y = 2$, and the initial displacement is in the center of the box ($x_0 = y_0 = L/2$). Assume the wave speed is $c = 3.5$ and select a value of Δt such that $(c\,\Delta t/\Delta x)^2 \leq 0.25$. Simulate the evolution of the wave; show 3D surface plots (like you did for the electric potential) at several interesting time steps[2] to convince me that your wave is propagating correctly. (Hint: You should get some surface maps that resemble to the front of your textbook).

Grading Rubric: Program has required features described in prompt (25%), graphs show correct behavior and are easy to interpret (25%), written response is factually correct (25%) and uses proper grammar and spelling (15%), code compiles without errors and reproduces all output (10%)

[2] Or, again, make an animation! Beware that 3D animations can take much longer to render, so don't be surprised if your code runs for a while. Try making a few surface plots *first*, to make sure your code works, before attempting to animate the results.