Homework 3

AMATH 481/581

Problem 1

The BVP

$$\theta''(t) + 0.1\theta'(t) + \sin(\theta(t)) = 0 \text{ with } \theta(0) = 0.5 = \theta(6)$$

describes the motion of a pendulum. The function $\theta(t)$ gives the angle of a pendulum (where $\theta = 0$ means the pendulum is hanging straight down) at time t. The boundary conditions mean that the pendulum begins and ends at an angle of 0.5 radians.

In this problem, you should solve this problem with the direct method as described in class. That is, you should discretize the ODE using a second order central difference scheme for both θ'' and θ'. For each part of this problem, you should use $\Delta t = 0.06$.

Since the ODE is nonlinear, you will obtain a nonlinear system of equations, which you will need to solve with Newton's method. In this problem, you will explore the effects of different initial guesses on the resulting solution. For both parts, you should use a tolerance of 10^{-8} in Newton's method.

(a) Solve this BVP using a vector of all 0.5's as your initial guess. That is, start Newton's method with the guess $(0.5, 0.5, 0.5, \ldots, 0.5)$. Store your approximation of the value of θ at time $t = 3$ in a variable named `A1`. Store your approximation of the maximum value of θ (i.e., the largest value of θ from your N approximated values) in a variable named `A2`. Store your approximation of the minimum value of θ (i.e., the smallest value of θ from your N approximated values) in a variable named `A3`.

(b) Solve this BVP using a vector the form

$$\theta = 0.005t^4 - 0.07t^3 + 0.66t^2 - 2.56t + 0.55$$

(where $t = (0, 0.06, 0.12, \ldots, 6)$) as your initial guess. Store your approximation of the value of θ at time $t = 3$ in a variable named `A4`. Store your approximation of the maximum value of θ (i.e., the largest value of θ from your N approximated values) in a variable named `A5`. Store your approximation of the minimum value of θ (i.e., the smallest value of θ from your N approximated values) in a variable named `A6`.

Note: The original version of the homework had a very different initial guess. However, if your initial guess is far from the solution you want to converge to, then Newton's method is often highly sensitive to small changes (even as small as rounding error issues) and can lead to very different behavior. If you're curious, try out the initial guess $\theta = -3t^2 + 18t + 0.5$. Does this converge to the same solution? Why/why not? It might be illuminating to plot each of the intermediate guesses produced by Newton's method.

Problem 2

Consider the boundary value problem

$$\begin{aligned}
&u_{xx} + u_{yy} = 0 \;\text{ for all }\; 0 < x < 3 \;\text{ and }\; 0 < y < 3 \\
&u(x,0) = x^2 - 3x \;\text{ and }\; u(x,3) = \sin\left(\frac{2\pi x}{3}\right) \;\text{ for all }\; 0 \le x \le 3 \\
&u(0,y) = \sin\left(\frac{\pi y}{3}\right) \;\text{ and }\; u(3,y) = 3y - y^2 \;\text{ for all }\; 0 \le y \le 3
\end{aligned}$$

Solve this problem using the direct method like we did in class (i.e., discretize the derivatives using a 5-point Laplacian and solve the resulting system of equations). Your system should only include one equation for each interior grid point. The boundary values should be incorporated into the right-hand side of your system.

(a) Use $\Delta x = \Delta y = 0.05$. This is a small enough problem that you do not have to use a sparse matrix (although you can). Store your approximation of the value of $u(1,1)$ (i.e., u at $x = 1$ and $y = 1$) in a variable named `A7`. Store your approximation of the value of $u(2,2)$ (i.e., u at $x = 2$ and $y = 2$) in a variable named `A8`.

(b) Use $\Delta x = \Delta y = 0.015$. This is a large enough problem that you **do** need to use a sparse matrix. (It's possible this will run on your machine without sparse storage, but it definitely won't run on the autograder.) Store your approximation of the value of $u(1.005, 1.005)$ (i.e., u at $x = 1.005$ and $y = 1.005$) in a variable named `A9`. Store your approximation of the value of $u(1.995, 1.995)$ (i.e., u at $x = 1.995$ and $y = 1.995$) in a variable named `A10`.

Problem 3

Consider the boundary value problem

$$u_{xx} + u_{yy} = -e^{-2(x^2+y^2)} \text{ for all } -1 < x < 1 \text{ and } -1 < y < 1$$
$$u(x,1) = \frac{x^3 - x}{3} \text{ and } u(x,-1) = 0 \text{ for all } -1 \le x \le 1$$
$$u(-1,y) = 0 = u(1,y) \text{ for all } -1 \le y \le 1$$

Solve this problem using the direct method like we did in class (i.e., discretize the derivatives using a 5-point Laplacian and solve the resulting system of equations). Your system should only include one equation for each interior grid point. The boundary values should be incorporated into the right-hand side of your system.

(a) Use $\Delta x = 0.1$ and $\Delta y = 0.05$. This is a small enough problem that you do not have to use a sparse matrix (although you can). Store your approximation of the value of $u(0,0)$ (i.e., u at $x = 0$ and $y = 0$) in a variable named `A11`. Store your approximation of the value of $u(-0.5, 0.5)$ (i.e., u at $x = -0.5$ and $y = 0.5$) in a variable named `A12`.

(b) Use $\Delta x = 0.01$ and $\Delta y = 0.025$. This is a large enough problem that you **do** need to use a sparse matrix. (It's possible that this will run on your machine without sparse storage, but it definitely won't run on the autograder.) Store your approximation of the value of $u(0,0)$ (i.e., u at $x = 0$ and $y = 0$) in a variable named `A13`. Store your approximation of the value of $u(-0.5, 0.5)$ (i.e., u at $x = -0.5$ and $y = 0.5$) in a variable named `A14`.

Problem 4 - Relevant to the written report

This problem is only for the written report. You don't need to include the solutions to this problem when submitting to the autograder.

Consider the boundary value problem

$$u_{xx} + u_{yy} = -\sin(\pi x)\sin(\pi y) \text{ for all } 0 < x < 2 \text{ and } 0 < y < 2$$
$$u(x,0) = 0 = u(x,2) \text{ for all } 0 \le x \le 2$$
$$u(0,y) = 0 = u(2,y) \text{ for all } 0 \le y \le 2$$

The exact solution to this BVP is

$$u(x, y) = \frac{1}{2\pi^2} \sin(\pi x) \sin(\pi y)$$

(a) Solve this problem using the direct method like we did in class (i.e., discretize the derivatives using a 5-point Laplacian and solve the resulting system of equations). Your system should only include one equation for each interior grid point. The boundary values should be incorporated into the right-hand side of your system.

Use several different values of Δx (but keep $\Delta x = \Delta y$ throughout) and use a method similar to that of the previous two reports (i.e., make a log-log plot of the errors) to determine the order of accuracy. Include any relevant code, plots and/or data in your report.

(b) Now solve this BVP using the following 9-point Laplacian (with $\Delta x = \Delta y$):

$$\begin{aligned} u_{xx}(x, y) + u_{yy}(x, y) \approx \nabla_9^2 u(x, y) = \\ \frac{1}{6\Delta x^2} \Big(& 4u(x - \Delta x, y) + 4u(x + \Delta x, y) + 4u(x, y - \Delta x) + 4u(x, y + \Delta x) \\ & + u(x - \Delta x, y - \Delta x) + u(x - \Delta x, y + \Delta x) + u(x + \Delta x, y - \Delta x) \\ & + u(x + \Delta x, y + \Delta x) - 20u(x, y) \Big) \end{aligned}$$

Use several different values of Δx to solve the BVP, then use a method similar to that of the previous two reports (i.e., make a log-log plot of the errors) to determine the order of accuracy. Include any relevant code, plots and/or data in your report.

(c) Calculate $\nabla_9^2 u(x, y) - f(x, y)$ applied to the true solution u (not your approximation) at each grid point. This is the discretization error of the 9-point Laplacian. Plot this error as a 3-dimensional surface plot. On a separate graph, plot the function $f_{xx} + f_{yy}$, where $f(x, y) = -\sin(\pi x) \sin(\pi y)$ (the right-hand side of our original PDE). You should find that these surfaces have very similar shapes. It turns out that this is a general feature of this 9-point Laplacian: The leading error term of the approximation is proportional to $f_{xx} + f_{yy}$.

Write a sentence or two about how you could use this fact to more accurately/efficiently solve the BVP. How do you think that would affect the order of accuracy? (You **do not** have to implement this improvement in code.)