PHY407 Computational Lab 8
Solving PDEs, Part 1

Time management advice: In this lab, Question 2 is worth 40% of the grade but probably will require more than half the work to be done well. Consider splitting the work with your partner on that question. Q1 and Q3 are more straightforward. If you do them first, and quickly, you will have covered 60% of the lab's marks, and then can concentrate on Q2. We will accept incomplete solutions for Q2, there are a lot of parts to it.

Computational Background

- **Boundary value problem methods (for Q1 and Q2):** The first class of PDEs that Newman discusses in Chapter 9 are solutions of elliptic partial differential equations of which the Poisson equation (9.10) is a classic example. He discusses the Jacobi method, which is a relaxational method for solving Laplace's or Poisson's equation, and then speedups to this method using overrelaxation and Gauss-Seidel replacement. For this lab we would like you to focus on Gauss-Seidel as the speedup method for reasons related to Question 2.

  In Question 2, we will ask you to solve the Laplace equation problem for which the boundary condition is time dependent. Thus, with each time step, the Laplace equation needs to be solved again because the boundary condition changes. It's good to have a fast method at hand and to check how many iterations are required for convergence. Unfortunately, I have found that the overrelaxation method is unstable for this particular application and so I will not encourage you to use it for this application, even though it is potentially a lot faster.

- **Point to remember: leapfrog timestepping and the computational mode (for Q2):** Leapfrog timestepping for PDEs has some advantages because it is a conservative explicit method that only requires information at two time steps. However, leapfrog timestepping has a hidden problem that only becomes apparent once you start using it for long integrations: it is unstable because of a "ghost" or "computational" mode that shows up as a signal that flashes at each time step. A simple way to filter out this mode is to apply a filter that blends the two timesteps together. To work around this instability, you need to carry out a filtering step as follows, using a small adjustable parameter called r which is around 0.01 to 0.05.

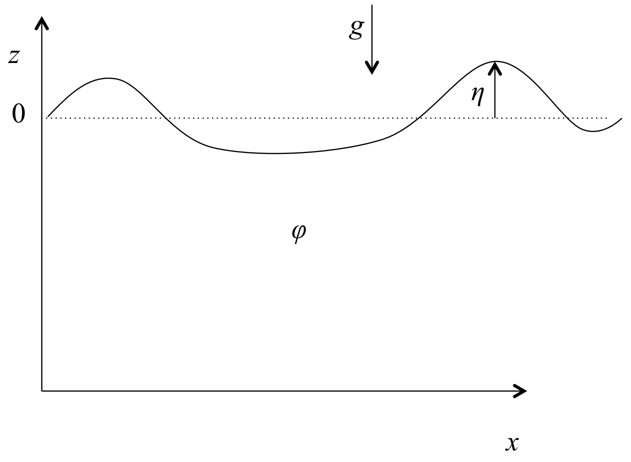
$$x(t+h) \leftarrow x(t+h) + r\left[x\left(t+\frac{3}{2}h\right) - 2x(t+h) + x\left(t+\frac{1}{2}h\right)\right],$$

  where the left arrow indicates that you replace the quantity on the left with the quantity on the right. A similar filter is applied to the half step. This filtering is known as "Robert-Asselin" filtering after the (Canadian!) scientists who invented it. This filter is complicated to implement and will not make a big difference to your solution to Q2. *So, Robert-Asselin filtering will not be a required part of Q2!* But it is something to keep in mind if you see evidence of "flashing" in your solution.

- **The FTCS method (for Q3):** Newman's text in Chapters 9.3.1-9.3.2 discusses the forward-time-centered-space method of solving PDEs, which is easy to program but has some issues of stability that are discussed in detail in the text. I will ask you to do one of the exercises from the book; it is an example that will run stably. Next lab we will discuss improvements on FTCS that can be used to solve various wave equations.

Physics Background

- **Deep water waves (for Q2):** Let's construct a simple model of gravity waves on the free surface of a constant density water layer that is quite deep (see below).



This is a model for waves out in the open ocean. The coordinates are $\boldsymbol{x} = (x, z)$, where x is the horizontal coordinate and z is the vertical coordinate. The components of the velocity of the fluid are $\boldsymbol{u} = [u(x, z, t), w(x, z, t)]$. The waves are generated by the restoring force of gravity in response to perturbations of the free surface which has height $\eta(x, t)$ above a resting level. We'll call this resting level $z = 0$. We'll assume that the slopes of the free surface aren't too steep and make similar assumptions to keep the system linear and easier to deal with mathematically. The key characteristics of the system are that

- The fluid is non-divergent.
- The free surface is a material surface, meaning the fluid elements can't cross it.
- We don't need to worry about pressure effects.

It is conventional to express the fluid velocity in terms of a potential function (not the gravitational potential): $\boldsymbol{u} = (u, v) = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}\right)$. The fluid is non-divergent, meaning that $\nabla \cdot \boldsymbol{u} = \partial u/\partial x + \partial w/\partial z = 0$, which implies that ϕ satisfies Laplace's equation: $\nabla^2\phi = 0$. Since the free surface is a material surface, the vertical velocity is related to the displacement of the free surface by $w(z = \eta) \approx w(z = 0) \approx \frac{\partial\phi}{\partial z} \approx \frac{\partial\eta}{\partial t}$. The momentum equation in the x direction says that the acceleration of the fluid $\frac{\partial u}{\partial t}$ is generated by the gravitational force $-g\frac{\partial \eta}{\partial x}$, which means that if the slope of the free surface height is positive, the fluid will tend to flow to the left, which hopefully makes sense to you. This force balance is calculated at the free surface height $z = \eta \approx 0$. Thus the momentum equation is

$$\frac{\partial u}{\partial t} = -g\frac{\partial \eta}{\partial x},$$

and with $u = \partial\phi/\partial x$ this can be integrated in x to obtain

$$\frac{\partial \phi(z=0)}{\partial t} = -g\eta.$$

The three equations of motion are then

$$\begin{aligned} \nabla^2\phi(x,z,t) &= 0,\ -\infty < z < 0, & (1)\\ \frac{\partial \phi(z=0)}{\partial t} &= -g\eta, \text{ and} & (2)\\ \frac{\partial \eta}{\partial t} &= \frac{\partial \phi(z=0)}{\partial z} & (3)\end{aligned}$$

In this set of equations, Laplace's equation (1) is solved in a domain for $z < 0$, and the boundary conditions on ϕ are updated according to Equations (3) and (2).

This seems complicated, but consider a plane wave of the form $\phi = \hat{\phi}(z)\cos(kx - \omega t)$, where k is a given wavenumber and $\hat{\phi}(z)$ reflects the vertical dependence of the waves. You can show that $\hat{\phi}(z) \propto \exp(kz)$, which is exponentially decreasing as z becomes more negative (that is, as depth increases). This solution lets the plane wave solve Laplace's equation and shows that the waves are trapped at the surface. By substitution you can show that dispersion relation for this system, which expresses ω as a function of k, is $\omega = \pm\sqrt{gk}$.[1] The phase speed of these waves depends on wavenumber: $c = \omega/k = \pm\sqrt{g/k}$. Because the phase speed depends strongly on wavelength, these water waves are highly dispersive, with longer length scale waves travelling faster. Water waves of 1 m wavelength travel about 8 m/s and of 4 m wavelength about 4 m/s. The group velocity, which tells you how fast wave packets travel, is $c_g = \partial\omega/\partial k = \pm c/2$. This system is symmetric in x so waves can propagate in either the plus or minus x direction.

In Q2 you will be asked to solve this system numerically for an intial depression in the fluid. In this two dimensional situation, this is something like dropping a long solid pipe in a lake and watching the waves move off in the direction perpendicular to the pipe's axis. Can you imagine what would happen? Let's see if your simulation matches your intuition.

Lab Instructions

For all codes mentioned, it is always recommended to write pseudocode first!

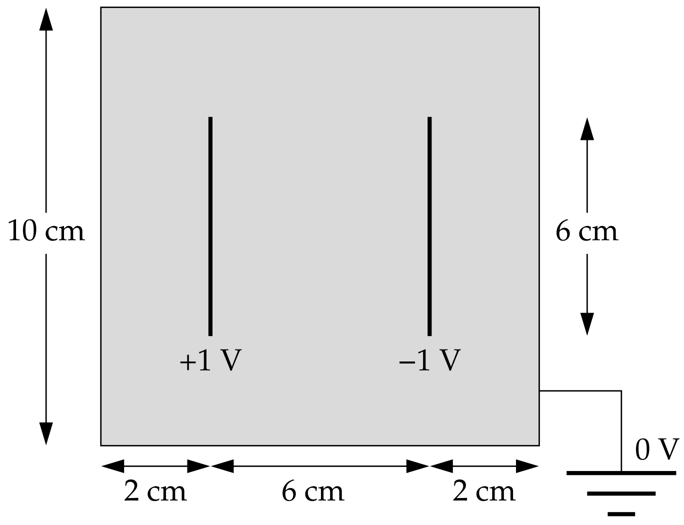
[1]Does this remind you of another formula? Think about the formula for the frequency of the linear pendulum, with length l being replaced by the wavelength of the wave.

For grading purposes, only hand in the following parts. Ensure that your codes are well commented and readable by an outsider:

- Q1: Hand in the plot and any written answer you wish to provide.
- Q2: Hand in your PSEUDOCODE, your plots and written answers.
- Q3: Hand in requested plot and written answers; justify your numerical parameter settings (but there is no need to hand in additional plots).

Lab Questions

1. **Electrostatics and Laplace's equation (30% of the lab):** See Ex. 9.3 of Newman. Consider the following simple model of an electronic capacitor, consisting of two flat metal plates enclosed in a square metal box:



For simplicity let us model the system in two dimensions. Using the *Gauss-Seidel* method without overrelaxation, write a program to calculate the electrostatic potential in the box on a grid of 100×100 points, where the walls of the box are at voltage zero and the two plates (which are of negligible thickness) are at voltages ± 1 V as shown. Have your program calculate the value of the potential at each grid point to a precision of 10^{-6} volts and then make a contour plot of the result.

Hint: This exercise is similar to Exercise 9.1, which you don't have to do. Notice that the capacitor plates are at fixed *voltage*, not fixed charge, so this problem differs from the problem with the two charges in Exercise 9.1. In effect, the capacitor plates are part of the boundary condition in this case: they behave the same way as the walls of the box, with potentials that are fixed at a certain value and cannot change.

2. **Linear water waves (40% of the lab):** In this exercise, we will adopt the Gauss- Seidel method to the situation of the deep water waves described in the background material. We will use the domain $0 < x < L$ and $-D < z < 0$ and examine the evolution of an

initial perturbation $\eta(x, t = 0) = \eta_0(x) = -A \exp[-(x - L/2)^2/\Delta^2)]$ and $\phi(x, z, t = 0) = 0$. We will choose L and D large enough that the domain boundaries are not important to the problem. This will allow us to set $\phi = 0$ on all domain boundaries, although this is not a valid boundary condition for reasons you can ask Paul or Oliver about.

Initially, try the following settings: $L = 400\,\text{m}$, $D = 50\,\text{m}$, and $\Delta = 2\,\text{m}$. A is arbitrary in this linear problem so set it to 1 m, which will provide a scale for convergence of the Gauss-Seidel method.

A tricky part of this calculation is that you have to calculate (3), which involves a vertical derivative of ϕ. To do this, you need to calculate $\partial\phi(z = 0)/\partial z$ numerically. Use a simple finite difference, according to

$$\frac{\partial\phi(z = 0)}{\partial z} \approx \frac{\phi(z = 0) - \phi(z = -dz)}{dz}.$$

Remember to set $\phi = 0$ on the sides and bottom of the domain for all time.

Your algorithm will proceed along the lines of the following pseudocode:

```
• Initialize $\eta$ and $\phi$ as above.
• Do a half-step Euler forward to get $\eta(x, t = h/2)$ and $\phi(x, z = 0, t = h/2)$.
• Solve Laplace's equation for $\phi(x, z, t = h/2)$ given this boundary condition.
• Then (and only then!) calculate $\partial\phi(x, z = 0, t = h/2)/\partial z$ as per the recommendation above.
• Then begin your main loop:
    – Update $\eta$ and $\phi(z = 0)$ to the next full step using the information from the half step.
    – Solve Laplace's equation with the updated boundary condition for $\phi(z = 0)$.
    – Update $\eta$ and $\phi(z = 0)$ to the next half step using the information from the previous full step.
    – Solve Laplace's equation again.
    – Save $\eta$ and $\phi$ from each full step as required for plotting purposes.
    – Repeat.
```

For numerical parameters, try $dz = dx = 1\,\text{m}$ and $h = 0.1\,\text{s}$ initially. Set an initial threshold for your iterations of the Gauss-Seidel method of $10^{-4}\,\text{m}^2/\text{s}$.

Now that we have the instructions, let's work on the exercise!

(a) The first thing to do is to get the simulation working. We want to have you write some pseudocode outlining the steps you will follow to get this working. Feel free to use the steps above. YOU WILL NEED TO HAND IN THIS PSEUDOCODE.

(b) Using the parameters above, plot $\eta(x, t)$ at t =0, 2, and 5 s. You will hand these in, but feel free to show your plots to Oliver and Paul to see if you are on the right track.

(c) Now we will adjust numerical parameters.

- With the parameter settings above, the timestep is large, and you will find that if you reduce it, you will be able to reduce the number of iterations required before the Gauss-Seidel solution converges. Adjust the timestep and document the impact on the number of iterations and on the total runtime of the simulation. The runtime will increase with decreasing timestep, but not quite proportionally, and your simulation will get better. You don't have to do an exhaustive job here, but show to us that you have investigated the tradeoff between timestep length and number of iterations (always keeping the target threshold for the Gauss-Seidel fixed). Your test can be 5s long. Use this testing to decide on a final timestep (you might adjust it for other work below).
- If you zoom your graphical windows to the center of the domain, you will see that the x-resolution recommended as a starting point is a bit too coarse to properly resolve the waves generated. Adjust the x-resolution to account for this - tell us briefly what you find.
- The other remaining important parameter to adjust is the width L of the domain. Since the boundary condition is incorrect, L should be increased, especially for longer simulations.

We are finally in a position to look at the physics of this problem.

(d) Using the settings you have found above, run a simulation approximately 40 s in length. For snapshots at t=0, 2, 10, and 40 seconds, plot for each snapshot $\eta(x,t)$ in one subplot and $\phi(x,z,t)$ immediately below as a contour plot. Make sure you know your contour interval and the units of ϕ, and choose a plotting window that highlights important features. In addition, create a filled contour, regular contour, or `imshow` plot of $\eta(x,t)$ with t on the vertical axis and x on the x axis. It is always fun to animate these kinds of solutions, but we will not ask you to hand in animations. You can create additional plots if they help you answer the following:

- The potential ϕ is defined so the fluid tends to travel up its gradient, from low to high values. Does the relationship between ϕ and η make sense to you?
- What has happened to the initial gaussian fluid bump as time evolves? Where are the shortest waves located, and where are the longest waves located?
- Track a pair of wavecrests as a function of time, and from this estimate a wavelength and a phase speed. Does this calculation agree with the predicted wave phasespeed?
- Can you see the wave groups in your figure? What is their approximate group speed? Does it agree with theory?

3. **Application of FCTS: Thermal diffusion in the Earth's crust (Newman Exercise 9.4; 30% of the lab):** A classic example of a diffusion problem with a time-varying boundary condition is the diffusion of heat into the crust of the Earth, as surface temperature varies with the seasons. Suppose the mean daily temperature at a particular point on the surface varies as:

$$T_0(t) = A + B\sin\frac{2\pi t}{\tau},$$

where $\tau = 365$ days, $A = 10°\text{C}$ and $B = 12°\text{C}$. At a depth of 20 m below the surface almost all annual temperature variation is ironed out and the temperature is, to a good approximation, a constant 11°C (which is higher than the mean surface temperature of 10°C—temperature increases with depth, due to heating from the hot core of the planet). The thermal diffusivity of the Earth's crust varies somewhat from place to place, but for our purposes we will treat it as constant with value $D = 0.1\,\text{m}^2\,\text{day}^{-1}$.

Write a program, or modify one of the ones given in this chapter, to calculate the temperature profile of the crust as a function of depth up to 20 m and time up to 10 years. Start with temperature everywhere equal to 10°C, except at the surface and the deepest point, choose values for the number of grid points and the time-step h, then run your program for the first nine simulated years, to allow it to settle down into whatever pattern it reaches. Then for the tenth and final year plot four temperature profiles taken at 3-month intervals on a single graph to illustrate how the temperature changes as a function of depth and time.