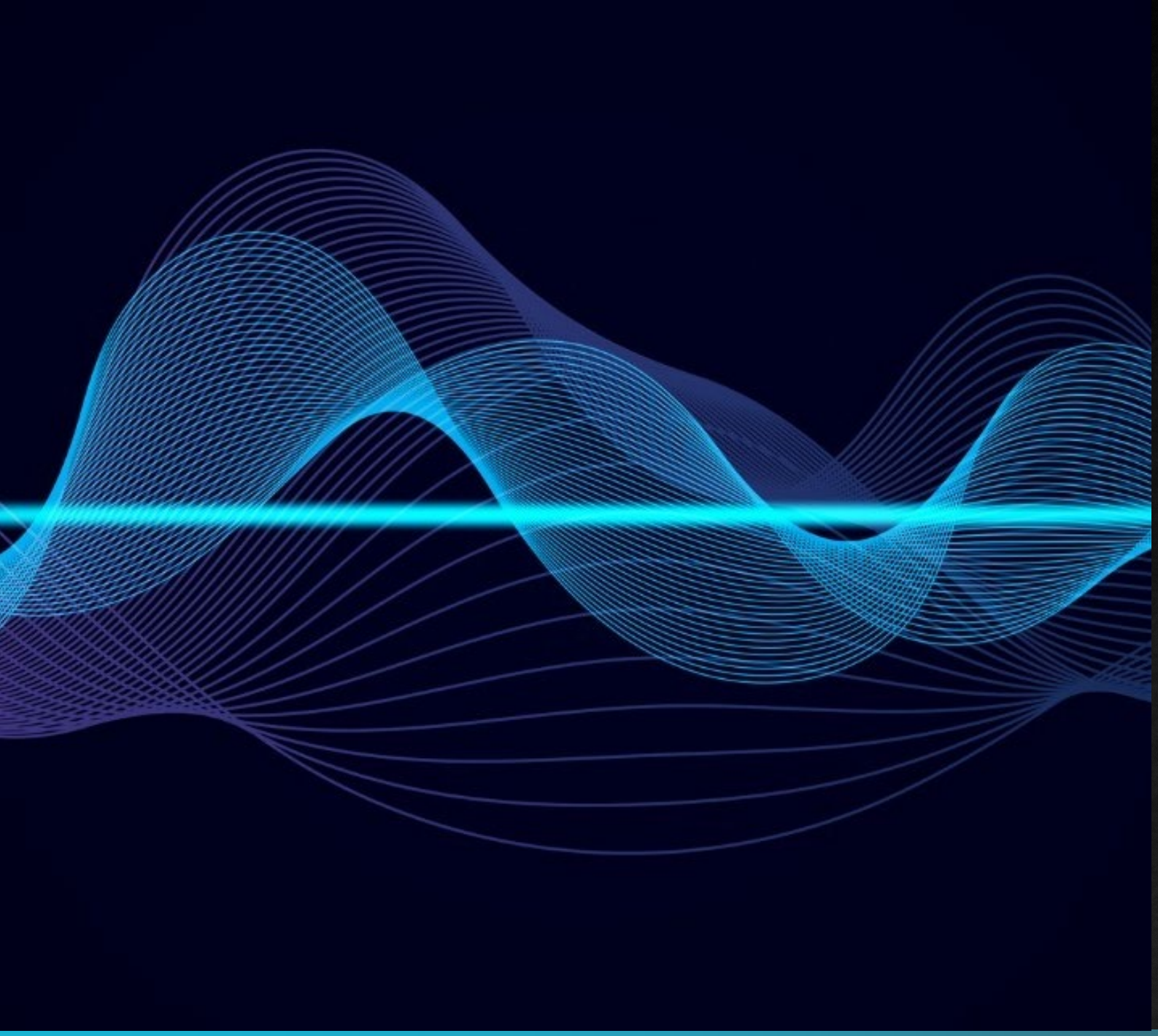


Computational Physics

PHYS 6260

Solving Partial differential equations (PDEs)

Announcements:

- HW3: Due Friday 1/31

Lecture Outline

We will cover these topics

- Relaxation method

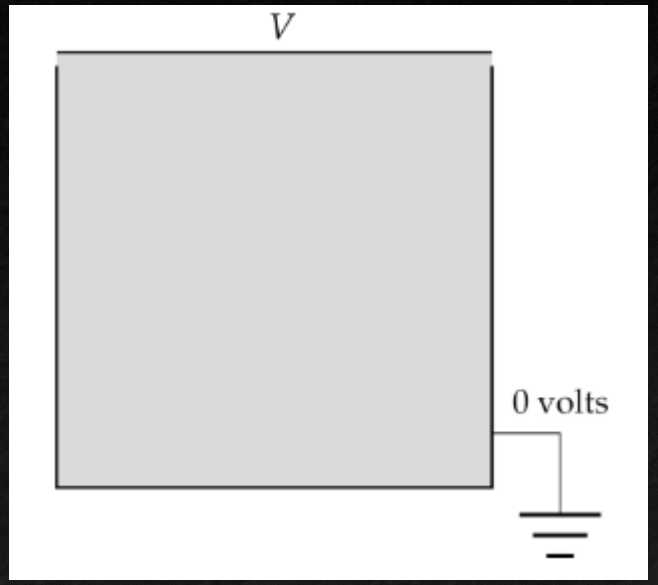
Faster methods

- Over-relaxation
- Gauss-Seidel method

Initial value problems

Introduction

- Some of the most interesting physics problems involve PDEs
  - Wave eqn, heat eqn, diffusion eqn, Laplace/Poisson eqn, Maxwell's eqns, Schrödinger eqn
- PDEs are straightforward to solve but are computationally intense because of their 3D nature



- First we'll cover boundary value problems (BVP) today
- Next lecture we'll start to cover initial value problems (e.g. a string being plucked)

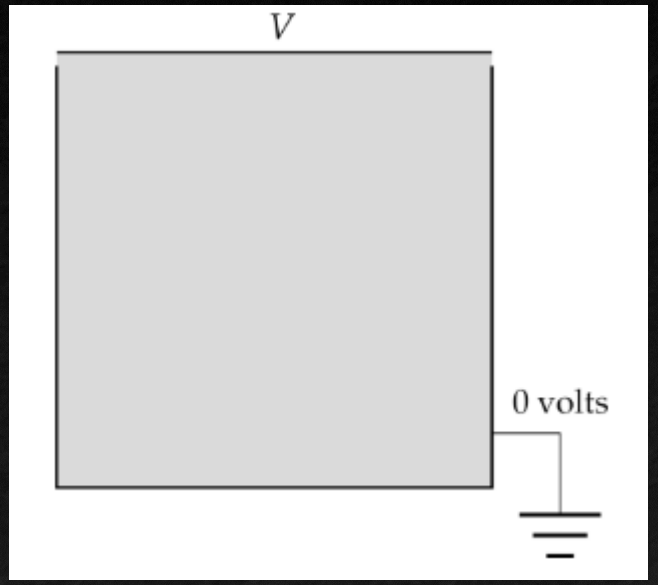
- BVPs have constraints on their boundaries while the interior obeys a set of PDEs

Introduction

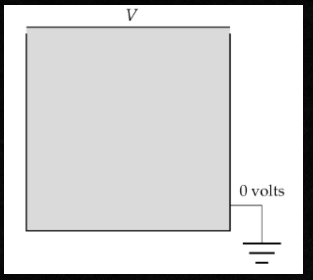
- BVPs have constraints on their boundaries while the interior obeys a set of PDEs
- Consider an empty conducting box with a voltage V on the top wall while the other 5 walls are grounded
- What is the electric potential V inside the box?
- We have Gauss's Law $\nabla \cdot \vec{E} = 0$ and $\vec{E} = -\nabla\phi$, giving

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0$$

- This is time-independent (electrostatics) that makes the solution much easier



Relaxation method: BC problems



- A fundamental technique to solve PDEs is the *method of finite differences*
- Let's consider our electrostatics problem in 2D

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$$

- We want to solve for ϕ given the boundary conditions
- Set up a grid of points where we'll solve for ϕ
- Because it's a box, we'll use Cartesian coordinates
- We can use central differencing to numerically calculate the partials

$$\frac{\partial^2 \phi}{\partial^2 x} = \frac{1}{a^2}\left[\phi(x+a,y) + \phi(x-a,y) - 2\phi(x,y)\right]$$
$$\frac{\partial^2 \phi}{\partial^2 y} = \frac{1}{a^2}\left[\phi(x,y+a) + \phi(x,y-a) - 2\phi(x,y)\right]$$

Relaxation method: BC problems

$$\frac{\partial^2\phi}{\partial^2 x} = \frac{1}{a^2}\left[\phi(x+a,y)+\phi(x-a,y)-2\phi(x,y)\right]$$
$$\frac{\partial^2\phi}{\partial^2 y} = \frac{1}{a^2}\left[\phi(x,y+a)+\phi(x,y-a)-2\phi(x,y)\right]$$

- We insert these expressions into the Laplace equation and obtain

$$\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)-4\phi(x,y)=0.$$

- Solving for the potential at (x,y), we find

$$\phi(x,y)=\tfrac{1}{4}\left[\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)\right],$$

- This is just an average of the surrounding points
- Known as the **Jacobi method**

Relaxation method: BC problems

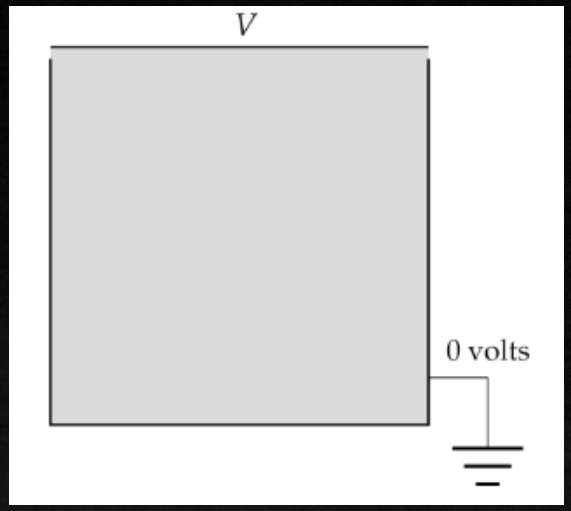
$$\phi(x,y) = \tfrac{1}{4}\left[\phi(x+a,y) + \phi(x-a,y) + \phi(x,y+a) + \phi(x,y-a)\right],$$

- How do we solve this system of equations?
  - Gaussian elimination or LU decomposition is too difficult with so many terms
- We can use the relaxation method
  - Choose an initial guess
  - Iteratively update the solution
  - Halt when a desired accuracy is reached (that is, the change between current and updated solutions)

- The Jacobi method is always stable and will converge to the correct solution

Jacobi method: Laplace's Eqn

In-class problem



- Let's solve this example problem for a top voltage V = 1 V
- Download the skeleton code `06_laplace0.py`
- The box is 1 meter on a side
- We set up a 101^2 grid (including the boundaries) to have a 1cm spacing
- Use the Jacobi method to solve for the potential inside the box

$$\phi(x,y) = \tfrac{1}{4}\left[\phi(x+a,y) + \phi(x-a,y) + \phi(x,y+a) + \phi(x,y-a)\right],$$

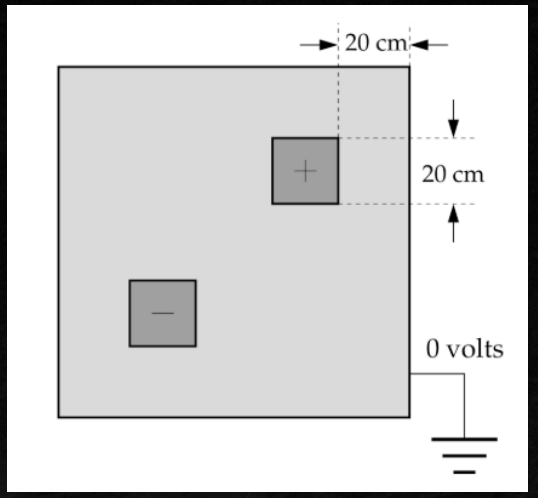
Relaxation method: BC problems

- This example took over 700 iterations to converge to a solution, coming from the 2nd derivative being only 1st-order accurate
- We can use higher-order estimates (involving more neighboring grid points) that are more accurate and converge in fewer iterations. However this requires more calculations

- As in any physics problem, you want to choose a grid that aligns with the system's symmetry or boundary

Jacobi method: Poisson Eqn

In-class problem



- Let's look at a more complex electrostatics problem
- Consider a conducting and grounded 1m box
- There are two (20cm)^2 squares with a charge density of ± 1 C m^{-2}. There lower left corners are at (x,y) = (0.2, 0.2) m and (0.6, 0.6) m. The system is shown above.
- We need to solve the Poisson equation

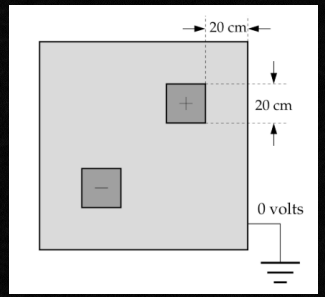
$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0}$$

- We can use finite differencing to express the Poisson equation as

$$\frac{1}{a^2}\left[\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)-4\phi(x,y)\right] = -\frac{\rho(x,y)}{\epsilon_0}.$$

Jacobi method: Poisson Eqn

In-class problem



$$\frac{1}{a^2}\left[\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)-4\phi(x,y)\right]=-\frac{\rho(x,y)}{\epsilon_0}.$$

- Solving for the potential, we find

$$\phi(x,y)=\frac{1}{4}\left[\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)\right]+\frac{a^2}{4\epsilon_0}\rho(x,y).$$

- The last term is known as the source/sink term that supplies the charge density for the potential

- Modify the skeleton code `06_poisson0.py` and use the Jacobi method to calculate the potential

Over-relaxation

- The Jacobi method is straightforward but it converges slowly
- We can accelerate the convergence by overshooting the solution
  - For example at some point, suppose our initial guess is 0.1, but the solution is 0.5
  - In the first iteration, the solution is updated from 0.1 to 0.3
  - We can overshoot a little and go to 0.4
- We don't know the solution, but the method of over-relaxation will converge with less iterations

- Let's consider the Laplace equation again, where we're updating the solution by some amount $\Delta\phi$

$$\phi'(x,y) = \phi(x,y) + \Delta\phi(x,y),$$

Over-relaxation

$$\phi'(x,y) = \phi(x,y) + \Delta\phi(x,y),$$

- Define a set of over-relaxed values

$$\phi_\omega(x,y) = \phi(x,y) + (1+\omega)\Delta\phi(x,y),$$

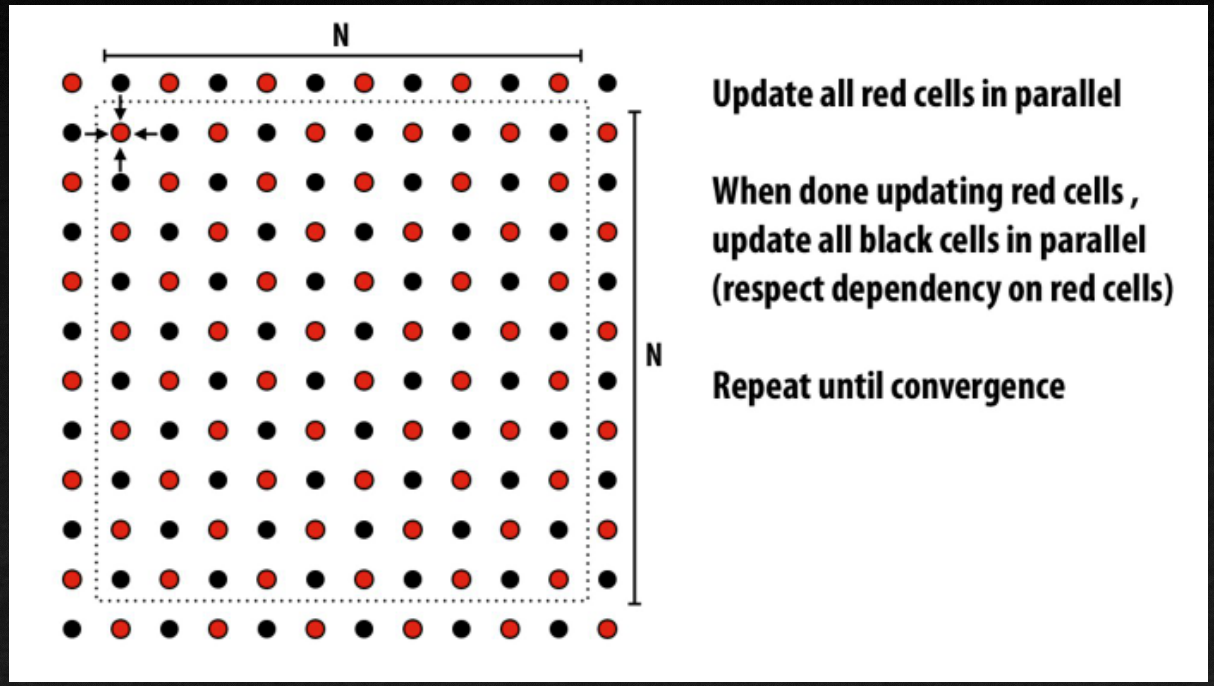
- Here $\omega > 0$, meaning that we let the solution change a little more than the Jacobi method
- Now substitute $\Delta\phi$ from its definition into the over-relaxed solution to obtain

$$\begin{aligned}\phi_\omega(x,y) &= \phi(x,y) + (1+\omega)[\phi'(x,y) - \phi(x,y)] \\ &= (1+\omega)\phi'(x,y) - \omega\phi(x,y) \\ &= \frac{1+\omega}{4}[\phi(x+a,y) + \phi(x-a,y) + \phi(x,y+a) + \phi(x,y-a)] - \omega\phi(x,y)\end{aligned}$$

- The over-relaxation method is very similar to the Jacobi method, but we have to choose some ω value
  - No exact method to choose ω, it is usually stable for <1
  - If too large, the method will oscillate around the true solution
  - In the previous example, $\omega = 0.9$ accelerates the convergence tremendously by 10-20 times

Gauss-Seidel method

- A widely-used “numerical trick” to speed up the Jacobi method is to use the updated values as they are calculated. This is known as the Gauss-Seidel method.
- In practice, this means we will have one array for the solution and replace the values as they are calculated
- To the right, the red-black variant is shown
- We can also use over-relaxation in combination with the Gauss-Seidel method
- One benefit is that the over-relaxed Gauss-Seidel method is formally stable and never diverges



Initial value problems: Overview

- Boundary value problems are only one type of PDE that are time-independent
- There are many problems that evolve and are time-dependent
- These are called initial value problems
- Goal: Predict their future variation

- A simple example is the diffusion (heat) equation in one dimension

$$\frac{\partial \phi}{\partial t} = D \frac{\partial^2 \phi}{\partial x^2}$$

- Here D is the diffusion coefficient

Initial value problems: Overview

$$\frac{\partial \phi}{\partial t} = D \frac{\partial^2 \phi}{\partial x^2}$$

- The variable $\phi(x, t)$ depends on space and time
- Can think of this as a 2D problem, like our previous example using the 2D Laplace equation
- Now we can consider a space-time grid, where both space and time are discretized

- We have an initial system setup, but we don't know the endpoint
- Goal: Predict how it behaves afterward

- The only feasible method is based on forward integration

The FTCS Method

$$\frac{\partial \phi}{\partial t} = D \frac{\partial^2 \phi}{\partial x^2}$$

- Just like how we represented partial derivatives in boundary value problems, we can do the same with the time derivative but forward in time
- We can write the right-hand side as

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{1}{a^2} \left[\phi(x+a,t) + \phi(x-a,t) - 2\phi(x,t) \right]$$

- That gives

$$\frac{\partial \phi}{\partial t} = \frac{D}{a^2} \left[\phi(x+a,t) + \phi(x-a,t) - 2\phi(x,t) \right]$$

- We can think of the variables at different points in space as separate variables
- Meaning that we have a set of simulataneous ODEs
- On a single compute core, it's no problem to solve thousands or even millions of them

The FTCS Method

$$\frac{\partial \phi}{\partial t} = D \frac{\partial^2 \phi}{\partial x^2}$$

- Now we need to choose a method to solve the left-hand side, $\partial\phi/dt$
- The right-hand side is only first-order accurate
- So there's little benefit to represent the left-hand side with a higher-order method

- We choose Euler's method to advance the system forward in time
  - RK methods would be a waste because the increased accuracy would be lost to the RHS 2nd order error

$$\begin{aligned}\phi(t+h) &\simeq \phi(t) + h\frac{d\phi}{dt} \\ &\simeq \phi(x,t) + h\frac{D}{a^2}\left[\phi(x+a,t) + \phi(x-a,t) - 2\phi(x,t)\right]\end{aligned}$$

- This is known as the forward-time, centered-space (FTCS) method for solving PDEs

The Heat Equation: FTCS

In-class example

$$\phi(t+h) \simeq \phi(t) + h\frac{d\phi}{dt}$$
$$\simeq \phi(x,t) + h\frac{D}{a^2}\left[\phi(x+a,t) + \phi(x-a,t) - 2\phi(x,t)\right]$$

- The flat base of a container made of 1cm thick stainless steel is initially at a uniform temperature of 20C
- The contained is placed in a cold bath at 0C and filled with soup at 80C (assume constant w/ time)
- Goal: Calculate the temperature profile from the hot → cold side w.r.t. position and time
- Thermal conduction is governed by the diffusion equation (shown above)

- Divide the x-axis into 100 equal grid integrals
- Boundary conditions: hot and cold bath temperatures
- Initial condtions: 20C everywhere except boundaries
- Heat diffusion coefficient for stainless steel: $D = 4.25 \times 10^{-6}\ m^2 s^{-1}$
- Make a temperature profile plot at t = (0.01, 0.1, 0.4, 1, and 10) seconds all on the same graph

Numerical stability

- The FTCS method works well for the diffusion equation
- But there are many other equations where it's stable for only some conditions
- Consider the wave equation

$$\frac{\partial^2 \phi}{dx^2} - \frac{1}{v^2}\frac{\partial^2 \phi}{dt^2} = 0$$

- To solve this, we would use central spatial differencing, like before
- But for the 2nd order time derivative, we can write two 1st order ODEs

$$\frac{\partial \phi}{\partial t} = \psi(x,t), \qquad \frac{\partial \psi}{\partial t} = \frac{v^2}{a^2}\left[\phi(x+a,t) + \phi(x-a,t) - 2\phi(x,t)\right]$$

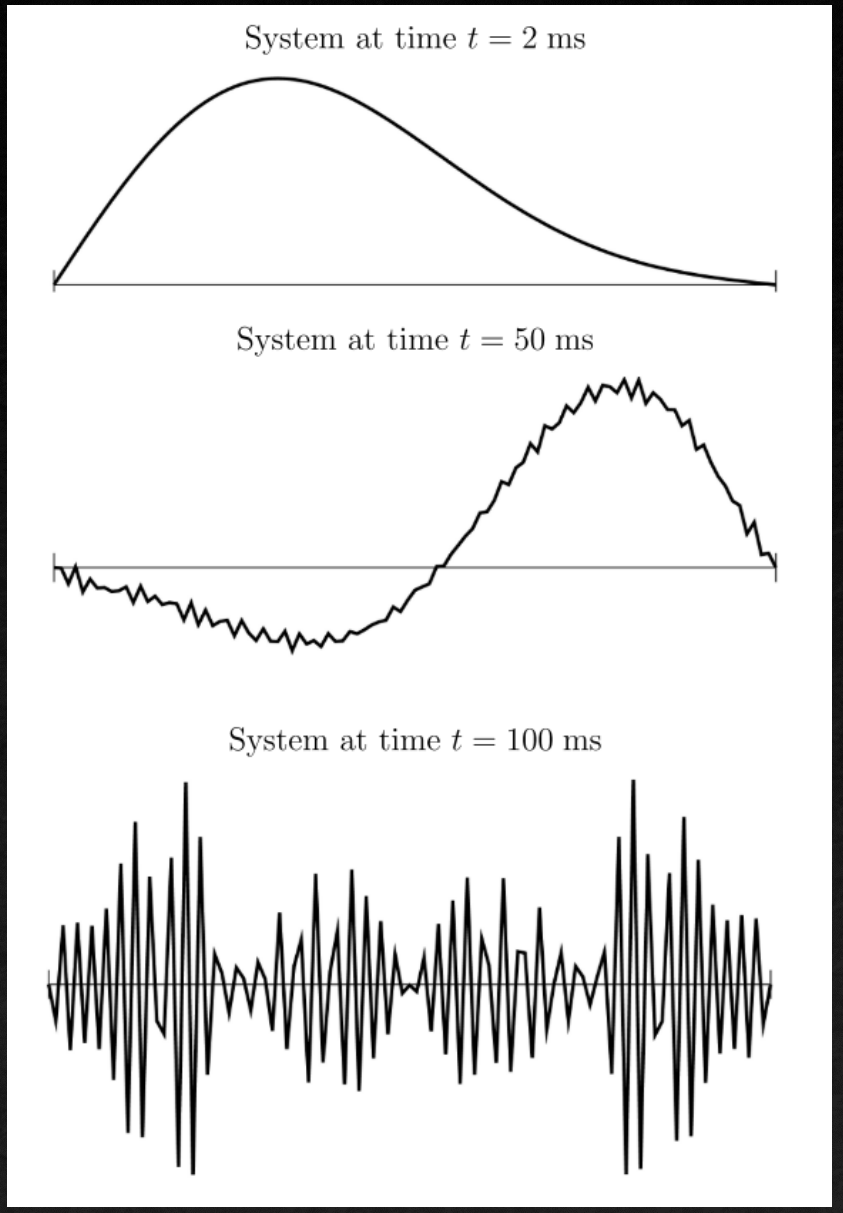
Numerical stability

$$\frac{\partial \phi}{\partial t} = \psi(x,t), \qquad \frac{\partial \psi}{\partial t} = \frac{v^2}{a^2}\left[\phi(x+a,t) + \phi(x-a,t) - 2\phi(x,t)\right]$$

- Apply Euler's method to both variables

$$\phi(x,t+h) = \phi(x,t) + h\psi(x,t)$$
$$\psi(x,t+h) = \phi(x,t) + h\frac{v^2}{a^2}\left[\phi(x+a,t) + \phi(x-a,t) - 2\phi(x,t)\right]$$

- Evolve the system with some timestep h



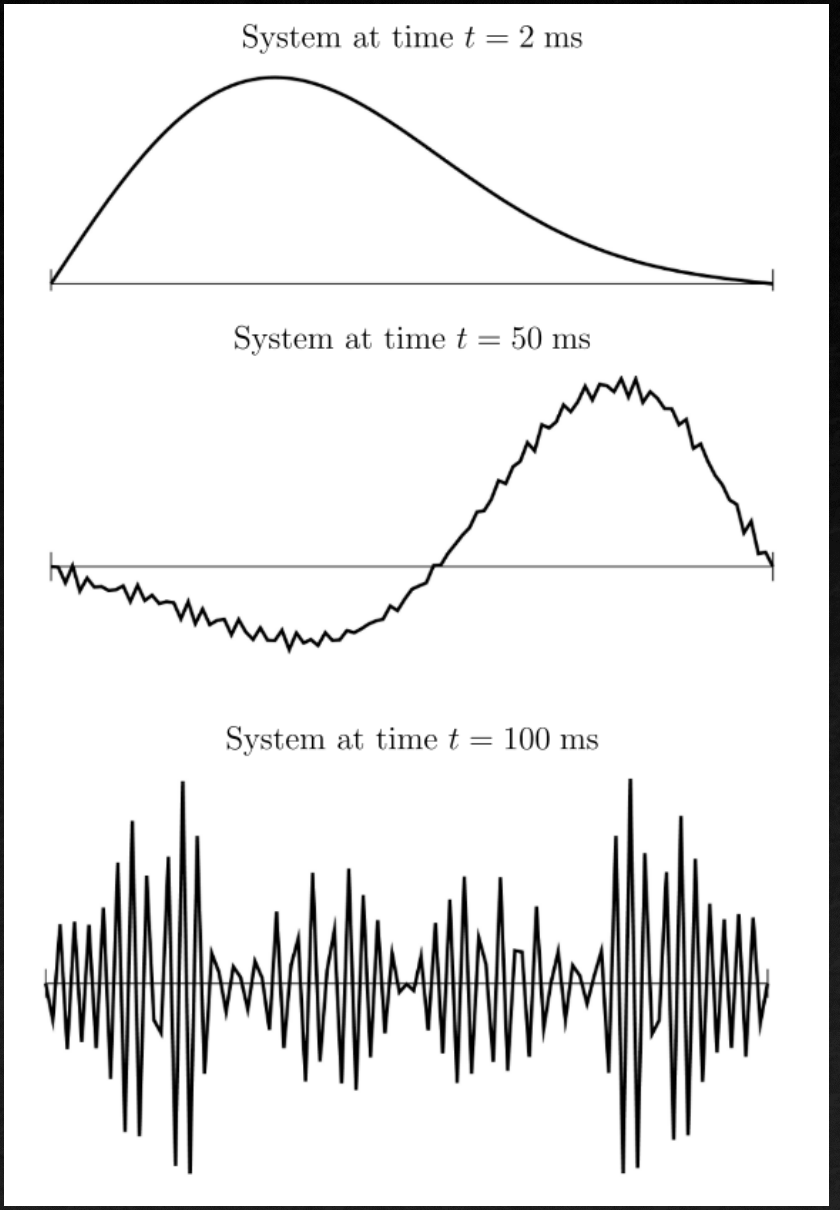
Numerical stability

- This is not roundoff or truncation error
- The errors will acculumate, eventually causing the calculation to crash

- We can perform a von Neumann stability analysis
- Here we express the solution as a Fourier series

$$\phi(x,t) = \sum_k c_k(t) e^{ikx}$$

- Given such a solution, we can see how each term changes in the next timestep
- *Note: We cannot perform this for non-linear equations*



Numerical stability

- Let's plug this solution into the FTCS diffusion equation

$$\begin{aligned}\phi(x,t+h) &= c_k(t)e^{ikx} + h\frac{D}{a^2}\left[e^{ik(x+a)} + e^{ik(x-a)} - 2e^{ikx}\right] \\ &= \left[1 + h\frac{D}{a^2}\left(e^{ika} + e^{-ika} - 2\right)\right] c_k(t)e^{ikx} \\ &= \left[1 - h\frac{4D}{a^2}\sin^2\left(\frac{ka}{2}\right)\right] c_k(t)e^{ikx},\end{aligned}$$

- Here we've used $e^{i\theta} + e^{-i\theta} = 2$ and $1 - \cos\theta = 2\sin^2\left(\frac{\theta}{2}\right)$

Numerical stability

- Notice that each Fourier coefficient is independent of each other, evolving as

$$c_k(t+h) = \left[1 - h\frac{4D}{a^2}\sin^2\left(\frac{ka}{2}\right)\right] c_k(t).$$

- The solution will be unstable if these grow with them, otherwise it's stable
- The largest the $\sin^2$ term can be is 1, so the solution is stable when $h \leq \frac{a}{2D}$
- If the timestep is larger than this, the solution can diverge

Numerical stability: Wave eqn

- Let's apply the von Neumann analysis to the wave equation (two 1st order ODEs)

$$\begin{pmatrix} \phi(x,t) \\ \psi(x,t) \end{pmatrix} = \begin{pmatrix} c_\phi(t) \\ c_\psi(t) \end{pmatrix} e^{ikx}$$

- Making the same Fourier series substitution, we find

$$\begin{aligned} c_\phi(x+h) &= c_\phi(t) + h c_\psi(t), \\ c_\psi(x+h) &= c_\psi(t) - h c_\phi(t) \frac{4v^2}{a^2} \sin^2 \frac{ka}{2}. \end{aligned}$$

Numerical stability: Wave eqn

$$c_\phi(x+h) = c_\phi(t) + hc_\psi(t),$$
$$c_\psi(x+h) = c_\psi(t) - hc_\phi(t)\frac{4v^2}{a^2}\sin^2\frac{ka}{2}.$$

- Write this in vector form as **c**(t+h) = **A**c(t), where

$$\mathbf{A} = \begin{pmatrix} 1 & h \\ -hr^2 & 1 \end{pmatrix} \qquad \text{with} \qquad r = \frac{2v}{a}\sin\frac{ka}{2}$$

- We can write **c**(t) as a linear combination of two eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ of **A**

$$\mathbf{c}(t+h) = \mathbf{A}(\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2) = \alpha_1\lambda_1\mathbf{v}_1 + \alpha_2\lambda_2\mathbf{v}_2,$$

- After m timesteps, we would have

$$\mathbf{c}(t+h) = \alpha_1\lambda_1^m\mathbf{v}_1 + \alpha_2\lambda_2^m\mathbf{v}_2,$$

Numerical stability: Wave eqn

- The solution is stable if both eigenvalues $(\lambda_1, \lambda_2) \leq 1$
- The eigenvalues are given by the determinant equation $A - \lambda I = 0$
- Using A for the wave equation (see above), both solutions have the same magnitude and always greater than 1

$$|\lambda| = \sqrt{1 + h^2 r^2} = \sqrt{1 + \frac{4h^2 v^2}{a^2} \sin^2 \frac{ka}{2}},$$

- Meaning that the numerical solution with FTCS will always diverge
- We need to find another technique to solve the wave equation