PHY407: Computational Physics
Fall, 2017

Lecture 8: Partial differential equations, Part 1

Summary & Status

- ☑ Weeks 1-3: Programming basics, numerical errors, numerical integration and differntiation.
- ☑ Weeks 4-5: Solving linear & nonlinear systems and Fourier transforms.
- ☑ Week 6: ODEs Part 1: RK4, Leapfrog, Verlet, adaptive time stepping; customizing python output
- ☑ Week 7: ODEs Part 2: Bulirsch-Stoer, Boundary Value Problems/shooting,
- ❑ Week 8: PDEs Part 1
  - Intro, elliptic equation solvers, FTCS
- ❑ Week 9: PDEs Parts 2
- ❑ Weeks 10-11: Random numbers & Monte Carlo methods

PHY407: Computational Physics
Fall, 2017

Lecture 8: Partial Differential Equations, Part 1

- Classifying PDEs
- Elliptic equations: Jacobi, Gauss-Seidel
- FTCS and stability

Solving PDEs

- Solving partial differential equations is one of the pinnacles of computational physics, bringing together many methods.

Solving PDEs

- Solving partial differential equations is one of the pinnacles of computational physics, bringing together many methods.

Elliptic equations: $\nabla^2 \Phi = \rho$ Poisson equation

Parabolic equations: $\dfrac{\partial T}{\partial t} = \kappa \nabla^2 T$ Diffusion equation

Hyperbolic equations: $\dfrac{\partial^2 \phi}{\partial t^2} = c^2 \nabla^2 \phi$ Wave equation

Solving PDEs

- Solving partial differential equations is one of the pinnacles of computational physics, bringing together many methods.

$$\text{Elliptic equations: } \nabla^2\Phi = \rho \qquad \text{Poisson equation}$$

$$\text{Parabolic equations: } \frac{\partial T}{\partial t} = \kappa\nabla^2 T \qquad \text{Diffusion equation}$$

$$\text{Hyperbolic equations: } \frac{\partial^2 \phi}{\partial t^2} = c^2\nabla^2\phi \qquad \text{Wave equation}$$

- We are faced with design decisions on how to discretize and implement numerical methods.
- Stability is something we need to deal with a lot.

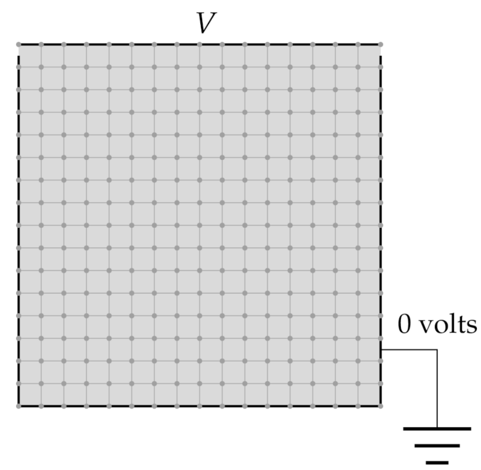
Calculating the Second Derivative

- Recall central difference calculation of second derivative (Section 5.10.5):

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{1}{12}h^2 f''''(x) + \cdots$$

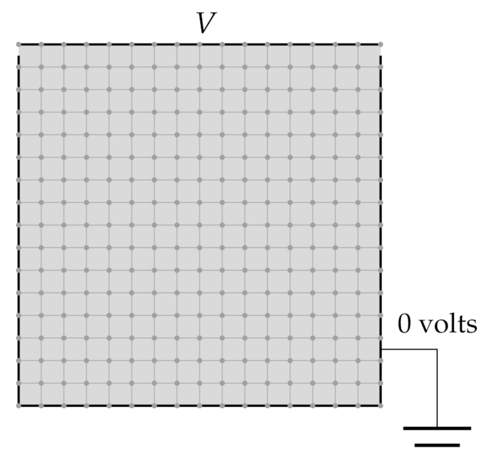
General approach

- Discretize system spatially and temporally.



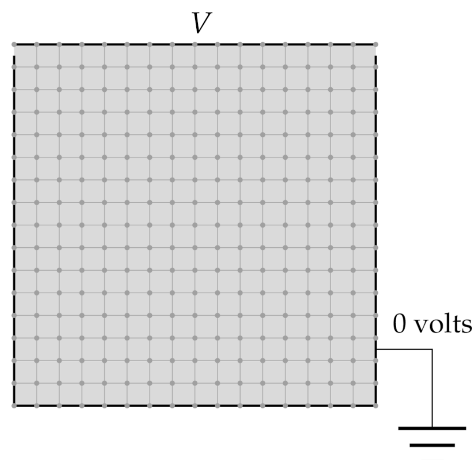
General approach

- Discretize system spatially and temporally. Can use
  - Finite difference
  - Spectral coefficients
  - Gaussian quadrature
  - Etc.



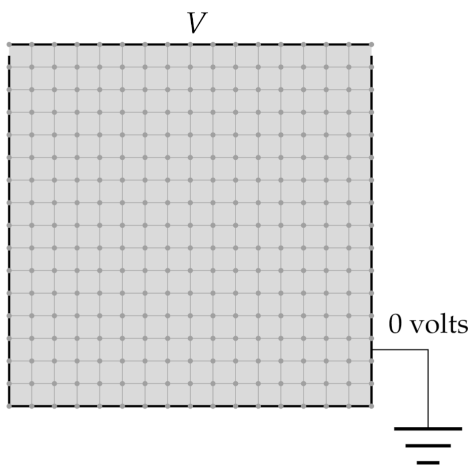
General approach

- Discretize system spatially and temporally. Can use
  - Finite difference
  - Spectral coefficients
  - Gaussian quadrature
  - Etc.
- You obtain a set of coupled ODEs that you need to solve in an efficient way.



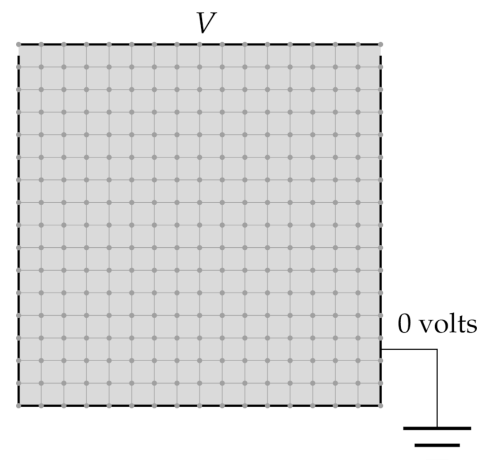
General approach

- Discretize system spatially and temporally. Can use
  - Finite difference
  - Spectral coefficients
  - Gaussian quadrature
  - Etc.
- You obtain a set of coupled ODEs that you need to solve in an efficient way.
- Coupling occurs because spatial derivatives bring information in from neighbouring points.



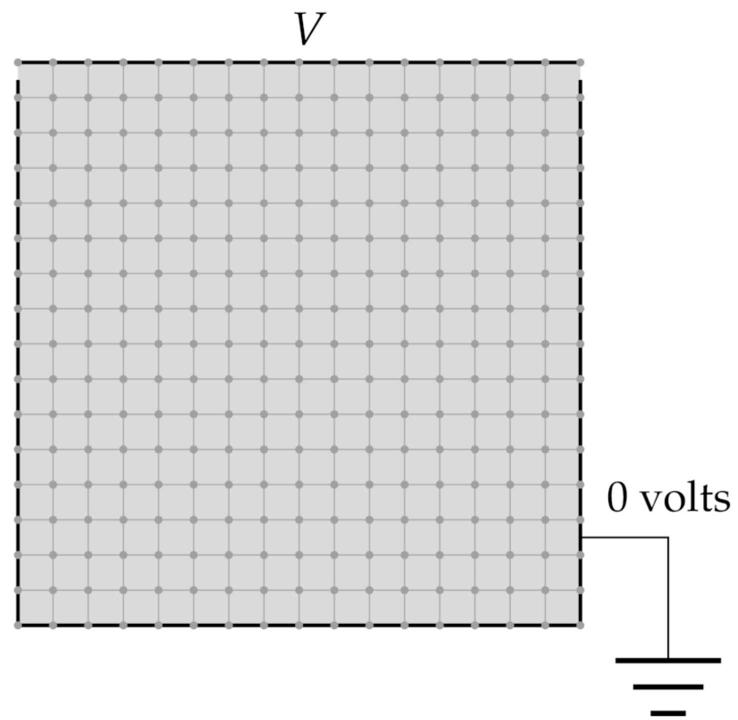
General approach

- Discretize system spatially and temporally. Can use
  - Finite difference
  - Spectral coefficients
  - Gaussian quadrature
  - Etc.
- You obtain a set of coupled ODEs that you need to solve in an efficient way.
- Coupling occurs because spatial derivatives bring information in from neighbouring points.
- Because of this coupling, errors depend on space and time and can get wave like characteristics.



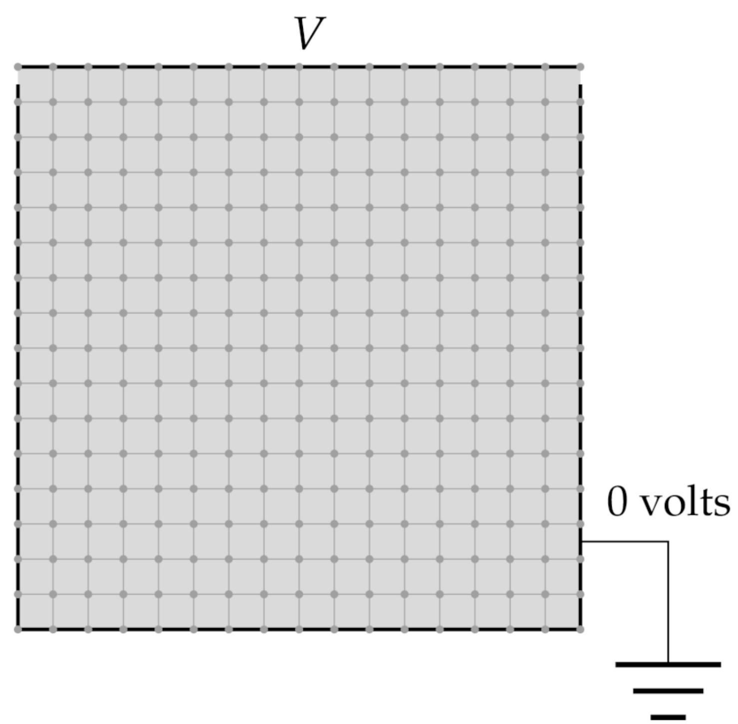
Elliptic Equations

- For solution of Laplace's equation or Poisson equation.



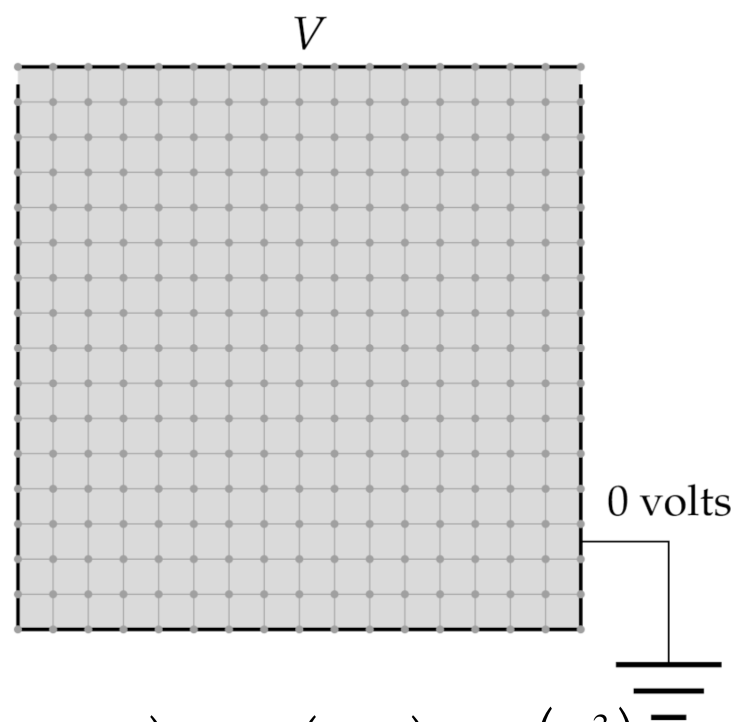
Elliptic Equations

- For solution of Laplace's equation or Poisson equation.
- On regular grid, finite difference form of Laplacian is



Elliptic Equations

- For solution of Laplace's equation or Poisson equation.
- On regular grid, finite difference form of Laplacian is



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

$$\frac{\partial^2 \phi}{\partial x^2} \approx \frac{\phi(x+a,y) - 2\phi(x,y) + \phi(x-a,y)}{a^2}$$

$$\frac{\partial^2 \phi}{\partial y^2} \approx \frac{\phi(x,y+a) - 2\phi(x,y) + \phi(x,y-a)}{a^2}$$

$$0 \approx \phi(x+a,y) + \phi(x-a,y) + \phi(x,y+a) + \phi(x,y-a) - 4\phi(x,y) + O(a^3)$$

Elliptic Equations

- Put together a series of equations of the form

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

for each x and y, subject to boundary conditions.

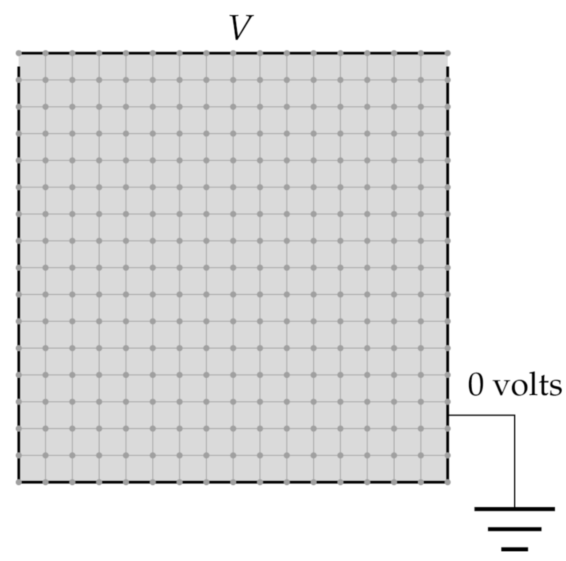
Elliptic Equations

- Put together a series of equations of the form

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

for each x and y, subject to boundary conditions.

- φ or derivative given on boundary. How would you handle these?

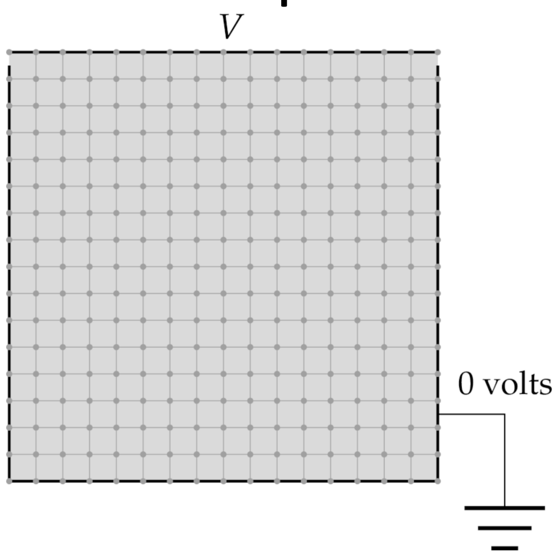


Elliptic Equations

- Put together a series of equations of the form

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

  for each x and y, subject to boundary conditions.
- φ or derivative given on boundary. How would you handle these?
- If φ given, use this value for adjacent points.
- If φ derivative given, find algebraic relationship between points near to boundary using finite difference.



Elliptic Equations

- Put together a series of equations of the form

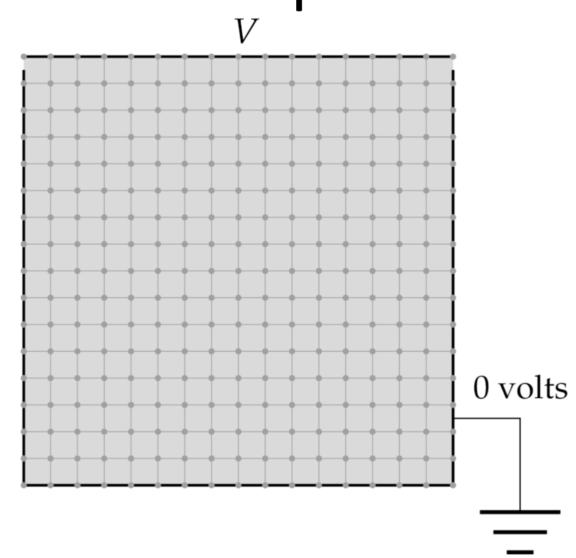
$$\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)-4\phi(x,y)=0$$

for each x and y, subject to boundary conditions.

- φ or derivative given on boundary. How would you handle these?
- If φ given, use this value for adjacent points.
- If φ derivative given, find algebraic relationship between points near to boundary using finite difference.
- Could solve using matrix methods:

$$\mathbf{L}\phi = \mathbf{R}\phi$$

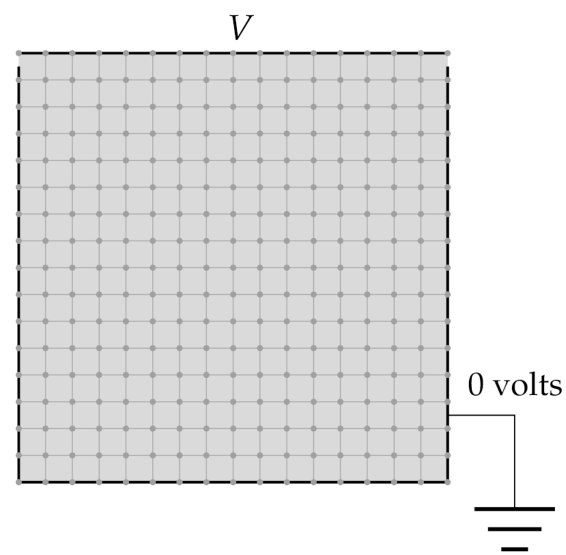
- But a simpler method is possible.



Jacobi Relaxation

- Iterate the rule

$$\phi'(x,y) = \frac{\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)}{4}$$

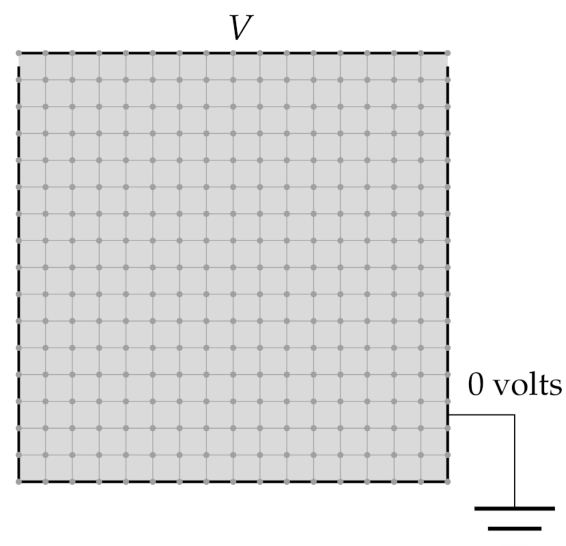


Jacobi Relaxation

- Iterate the rule

$$\phi'(x,y) = \frac{\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)}{4}$$

- For this problem it turns out that Jacobi Relaxation is always stable and so always gives a solution!
- Let's look at `laplace.py`



Other methods

- Gauss Seidel: replace function on the fly as in

$$\phi(x,y) \leftarrow \frac{\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)}{4}$$

Other methods

- Gauss Seidel: replace function on the fly as in

$$\phi(x,y) \leftarrow \frac{\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)}{4}$$

- This can be shown to run faster.
- Can also implement overrelaxation:

$$\phi(x,y) \leftarrow (1+\omega)\left[\frac{\phi(x+a,y)+\phi(x-a,y)+\phi(x,y+a)+\phi(x,y-a)}{4}\right] \\ -\omega\phi(x,y)$$

FTCS Solution of Heat Equation

Consider the 1-D heat equation:

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2},$$

BC: $T(0,t) = T_0, T(L,t) = T_L,$

Initial condition: $T(x,0) = \left(T_L - T_0\right)\left(\frac{f(x) - f(x_0)}{f(x_L) - f(x_0)}\right) + T_0$

FTCS Solution of Heat Equation

Consider the 1-D heat equation:

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2},$$

BC: $T(0,t) = T_0, T(L,t) = T_L,$

Initial condition: $T(x,0) = \left(T_L - T_0\right)\left(\frac{f(x) - f(x_0)}{f(x_L) - f(x_0)}\right) + T_0$

STEP 1: Discretize in space

$$x_j = \frac{L}{J} j = aj,\ j = 0 \ldots J,\ a = L/J$$

$$T_j(t) = \left[T_0(t), \ldots, T_J(t)\right]$$

$$\frac{\partial^2 T_j(t)}{\partial x^2} \approx \frac{T_{j+1} - 2T_j + T_{j-1}}{a^2},\ j = 1, \ldots, J-1$$

This is called *centered spatial* (CS) differencing.

FTCS Solution of Heat Equation

STEP 2: Discretize in time

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2},$$

$$\frac{dT_j(t)}{dt} \approx \kappa \frac{\left(T_{j+1} - 2T_j + T_{j-1}\right)}{a^2},\ j = 1,\ldots,J-1$$

$t_n = hn,\ h$ is time step.

Forward Euler (Forward Time - FT): use RHS temperature at current time

$$T_j(t_n) \equiv T_j^n$$

$$\frac{dT_j^n}{dt} \approx \frac{\left(T_j^{n+1} - T_j^n\right)}{h} \equiv \kappa\left(\frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{a^2}\right)$$

$$\boxed{T_j^{n+1} = T_j^n + \frac{\kappa h}{a^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right)}$$

This is the explicit FTCS method.

Introduction to Stability

Von Neumann Stability Analysis

- How can we determine stability in PDEs?

Von Neumann Stability Analysis

- How can we determine stability in PDEs? Consider a single spatial Fourier mode.

$$T(x_j, t_n) = T_j^n = \sum_k \hat{T}_k^n \exp\left(ikx_j\right)$$

Von Neumann Stability Analysis

- How can we determine stability in PDEs? Consider a single spatial Fourier mode.

$$T(x_j, t_n) = T_j^n = \sum_k \hat{T}_k^n \exp\left(ikx_j\right)$$

For single Fourier mode $\hat{T}_k^n \exp\left(ikx_j\right) = \hat{T}_k^n \exp\left(iajk\right)$,

$$T_j^{n+1} = T_j^n + \frac{\kappa h}{a^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right)$$

Von Neumann Stability Analysis

- How can we determine stability in PDEs? Consider a single spatial Fourier mode.

$$T(x_j, t_n) = T_j^n = \sum_k \hat{T}_k^n \exp(ikx_j)$$

For single Fourier mode $\hat{T}_k^n \exp(ikx_j) = \hat{T}_k^n \exp(iajk)$,

$$T_j^{n+1} = T_j^n + \frac{\kappa h}{a^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right)$$

$$\hat{T}_k^{n+1}\exp(iaj) = \hat{T}_k^n \exp(iajk)\left(1 - \frac{2\kappa h}{a^2}\right) + \frac{\kappa h}{a^2}\left(\hat{T}_k^n \exp(ia(j+1)k) - \hat{T}_k^n \exp(ia(j-1)k)\right)$$

$$\left|\frac{\hat{T}_k^{n+1}}{\hat{T}_k^n}\right| = 1 + \frac{\kappa h}{a^2}\left(e^{ika} + e^{-ika} - 2\right) = \left|1 - \frac{4h\kappa}{a^2}\sin^2\left(\tfrac{1}{2}ka\right)\right|$$

Von Neumann Stability Analysis

- How can we determine stability in PDEs? Consider a single spatial Fourier mode.

$$T(x_j, t_n) = T_j^n = \sum_k \hat{T}_k^n \exp(ikx_j)$$

For single Fourier mode $\hat{T}_k^n \exp(ikx_j) = \hat{T}_k^n \exp(iajk)$,

$$T_j^{n+1} = T_j^n + \frac{\kappa h}{a^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right)$$

$$\hat{T}_k^{n+1}\exp(iajk) = \hat{T}_k^n \exp(iajk)\left(1 - \frac{2\kappa h}{a^2}\right) + \frac{\kappa h}{a^2}\left(\hat{T}_k^n \exp\left(ia(j+1)k\right) - \hat{T}_k^n \exp\left(ia(j-1)k\right)\right)$$

$$\left|\frac{\hat{T}_k^{n+1}}{\hat{T}_k^n}\right| = 1 + \frac{\kappa h}{a^2}\left(e^{ika} + e^{-ika} - 2\right) = \left|1 - \frac{4h\kappa}{a^2}\sin^2\left(\tfrac{1}{2}ka\right)\right|$$

This is the growth factor and it should be less than unity for stability:

$$h \le \frac{a^2}{2\kappa}$$

Notice this is independent of k!

FTCS for the Wave Equation

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

(Subject to suitable boundary and initial conditions.)

FTCS for the Wave Equation

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

(Subject to suitable boundary and initial conditions.)

$$\frac{d^2 \phi_j(t)}{dt^2} \approx \frac{c^2}{a^2}\left(\phi_{j+1} - 2\phi_j + \phi_{j-1}\right),\ j = 1,\ldots,J-1$$

FTCS for the Wave Equation

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

(Subject to suitable boundary and initial conditions.)

$$\frac{d^2 \phi_j(t)}{dt^2} \approx \frac{c^2}{a^2}\left(\phi_{j+1} - 2\phi_j + \phi_{j-1}\right),\ j = 1, \ldots, J-1$$

Now transform to pairs of first order ODEs:

$$\frac{d\phi_j}{dt} = \psi_j$$

$$\frac{d\psi_j}{dt} = \frac{c^2}{a^2}\left(\phi_{j+1} - 2\phi_j + \phi_{j-1}\right)$$

and discretize using Euler-Forward (so there about 2J ODEs).

Stability for this method.

$$\begin{pmatrix} \phi_j^{n+1} \\ \psi_j^{n+1} \end{pmatrix} = \begin{pmatrix} 1 & h \\ \dfrac{-2hc^2}{a^2} & 1 \end{pmatrix} \begin{pmatrix} \phi_j^n \\ \psi_j^n \end{pmatrix} + \begin{pmatrix} 0 \\ \dfrac{c^2 h}{a^2}\left(\phi_{j+1}^n + \phi_{j-1}^n\right) \end{pmatrix}$$

Now consider a single Fourier component with wavenumber k as before:

$$\begin{pmatrix} \hat{\phi}_k^n \\ \hat{\psi}_k^n \end{pmatrix} \exp(ikja)$$

and we obtain

$$\begin{pmatrix} \hat{\phi}_k^{n+1} \\ \hat{\psi}_k^{n+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \hat{\phi}_k^n \\ \hat{\psi}_k^n \end{pmatrix}, \ \mathbf{A} = \begin{pmatrix} 1 & h \\ -hr^2 & 1 \end{pmatrix}, \ r^2 = \frac{2c}{a} \sin\left(\frac{ka}{2}\right)$$

which does depend on k.

Stability for this method.

$$\begin{pmatrix} \hat{\phi}_k^{n+1} \\ \hat{\psi}_k^{n+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \hat{\phi}_k^{n} \\ \hat{\psi}_k^{n} \end{pmatrix}, \ \mathbf{A} = \begin{pmatrix} 1 & h \\ -hr^2 & 1 \end{pmatrix}, \ r^2 = \frac{2c}{a} \sin\left(\frac{ka}{2}\right)$$

Now the eigenvalues of $\mathbf{A}$ are $\lambda_\pm = 1 \pm ihr$ and $\left|\lambda_\pm\right|^2 = 1 + h^2 r^2 \geq 1$ with corresponding eigenvectors $\mathbf{v}_+, \mathbf{v}_-$.

Suppose initial condition is $\alpha_+ \mathbf{v}_+ + \alpha_- \mathbf{v}_-$. After m timesteps, this becomes $\alpha_+ \lambda_+^m \mathbf{v}_+ + \alpha_- \lambda_-^m \mathbf{v}_-$, which will grow without bound!

So, FTCS is never stable for the wave equation!

PHY407: Computational Physics
Fall, 2017

Lecture 8: Partial Differential Equations, Part 1

- Classifying PDEs
- Elliptic equations: Jacobi, Gauss-Seidel
- FTCS and stability