Partial Differential Equations (PDEs)

Primary Reference: "*Numerical Methods for Physics*" by A. Garcia, Chapters 6-9

- A large and very important component of computational physics is devoted to numerically solving PDEs of some form or another
- The numerical solution of PDEs is a vast and complex area of study and can only hope to scratch the surface here
- The goal of these next few chapters (6-9) are to introduce you to some of the basic techniques and challenges associated with solving PDE's on a computer

Classification of PDEs

There are 2 classes of problems:

Initial value problems such as the *wave equation* is a hyperbolic equation

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2}$$

Or the diffusion equation which is a parabolic equation

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial}{\partial x}\left(\kappa \frac{\partial u(x,t)}{\partial x}\right)$$

The task here is given *u(x,0),* find *u(x,t)*

The second type are elliptic equations such as Poisson's equation where the solution has reached a steady state, where Φ(x,y) or a normal gradient is given at the boundary to be covered later (Chapter 8)

$$\frac{\partial^2 \Phi(x,y)}{\partial x^2} + \frac{\partial^2 \Phi(x,y)}{\partial x^2} = -\frac{\rho(x,y)}{\epsilon_0}$$

Classification of PDEs -2

- Parabolic PDEs describe time dependent, dissipative physical processes that *are* evolving towards a steady state
  - Dissipative means that energy, mass, etc. are not constant
  - Typically have exponentially decaying or growing solutions
  - Typically yield smooth solutions
  - Are time irreversible
- Hyperbolic PDEs describe time dependent, conservative physical processes that are *not* necessarily evolving towards a steady state
  - Conservative means energy, mass, etc
  - Can produce non-smooth solutions (e.g. shocks)
  - Are time reversible, in principal
  - Are probably the most challenging to solve numerically
- Elliptic PDEs describe physical systems that have reached steady state, or equilibrium and are time-independent

Classification of PDEs -3

For the following 2nd order PDE

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu = g$$

Where *a*-*g* can be constants or given functions (non-linear).

There are 3 basic types of PDES:

(a) parabolic: $b^2 - 4ac = 0$ $(b=0,\ c=0)$, *e.g., (diffusion)* $\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2}$

(b) hyperbolic: $b^2 - 4ac > 0$ $(b=0,\ a=-c)$, *e.g.,(wave)* $\frac{\partial^2 u(x,t)}{\partial t^2} = c^2\frac{\partial^2 u(x,t)}{\partial x^2}$

(c) elliptic: $b^2 - 4ac < 0$ $(b=0,\ a=c)$, *e.g., (Laplace)* $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$

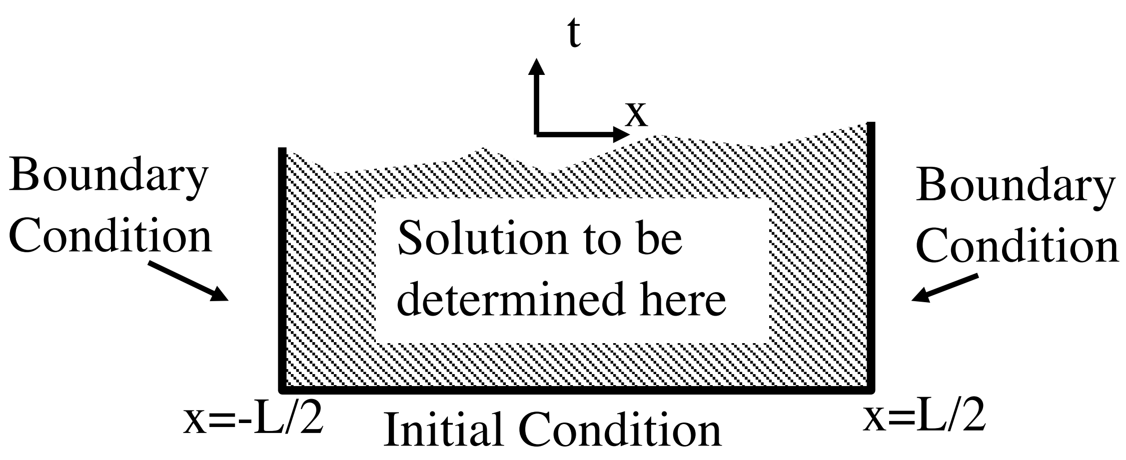
Initial Value Problems

Hyperbolic and Parabolic equations are both treated as initial value problems.

For the diffusion equation for temperature, *T(x,t)*

$$\frac{\partial T(x,t)}{\partial t} = \frac{\partial}{\partial x}\left(\kappa \frac{\partial T(x,t)}{\partial x}\right)$$

One would be given the initial *T(x,0)* and some boundary condition since the solution is likely constrained to some region of space



Boundary Conditions

Consider the Diffusion equation as applied to the Heat equation, *T(x,t)* is then the temperature

Dirichlet Boundary Conditions

$$T\left(\frac{L}{2},t\right) = T_b \qquad T\left(-\frac{L}{2},t\right) = T_a$$

where the temperature *T* has been specified at the boundary

Neumann Boundary (Flux) Conditions

$$-\kappa \left.\frac{dT}{dx}\right|_{x=-\frac{L}{2}} = F_a \quad and \quad -\kappa \left.\frac{dT}{dx}\right|_{x=\frac{L}{2}} = F_b$$

which specifies the heat flux at the boundaries

A boundary condition that specifies both Dirichlet and Neumann Boundary conditions is know as a *Cauchy* Boundary condition

Another boundary condition we will encounter are *Periodic boundary conditions*

$$T\left(\frac{L}{2},t\right) = T(-\frac{L}{2},t) \qquad \left.\frac{dT}{dx}\right|_{x=-\frac{L}{2}} = \left.\frac{dT}{dx}\right|_{x=\frac{L}{2}}$$

The choice of boundary/initial conditions impacts the solution (from *Arfken*, Table 8.1)

Boundary Conditions	**Type of partial differential equation**		
	Elliptic	Hyperbolic	Parabolic
Example equation	Laplace's equation in (x, y)	Wave equation in (x, t)	Diffusion equation in (x, t)
Dirichlet Open Surface	Insufficient	Insufficient	Unique, stable solution in one direction
Dirichlet Closed Surface	Unique stable solution	Solution not unique	Too restrictive
Neumann Open Surface	Insufficient	Insufficient	Unique, stable solution in one direction
Neumann Closed Surface	Unique stable solution (to an arbitrary const.)	Solution not unique	Too restrictive
Cauchy Open Surface	Unphysical Result (instability)	Unique stable solution	Too restrictive
Cauchy Closed Surface	Too restrictive	Too restrictive	Too restrictive

Discretization

Numerical solutions of PDEs produce a solution on a grid, both space and time are discretized. We will use finite-difference methods in the class, which is the simplest. Other approaches, such as finite element or finite volume are quite popular but more complex to implement.

Discretization *N* evenly space points in space in 1D on a line length *L* as

$$x_j = -\frac{L}{2} + \frac{j-1}{N-1} L$$

Or where h is the grid spacing $x_j = -\frac{L}{2} + (j-1)h$

Where $h = \frac{L}{N-1}$

The solution is evolved in time a what is called a marching method where the solution is computed one step into the future $t_n = (n-1)\tau$

1D Diffusion Equation

Analytic solution by method of images

Start with the equation, where κ is a constant

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

The solution can be verified to be a Gaussian

$$T_G(x,t) = \frac{1}{\sqrt{4\pi\kappa t}} e^{-\left[\frac{(x-x_0)^2}{4\kappa t}\right]}$$

Note that

$$\int_{-\infty}^{\infty} T_G(x,t) = 1$$

and

$$\lim_{t\to 0} T_G(x,t) = \delta(x - x_0)$$

Method of Images

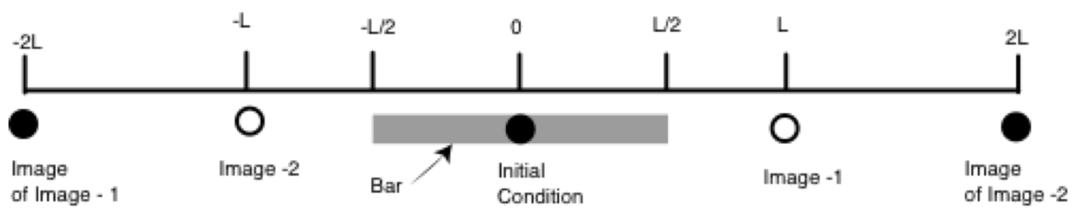
The method of images can be used to solve the diffusion equation with the boundary condition $T(x, t = 0) = \delta(x)$with the Dirichlet boundary condition

$$T\left(-\frac{L}{2}, t\right) = T\left(\frac{L}{2}, t\right) = 0$$

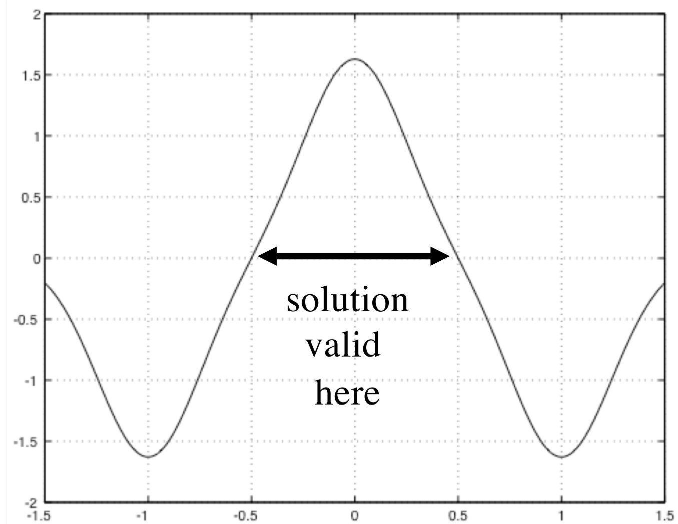
So we have to find $T(x,t)$.

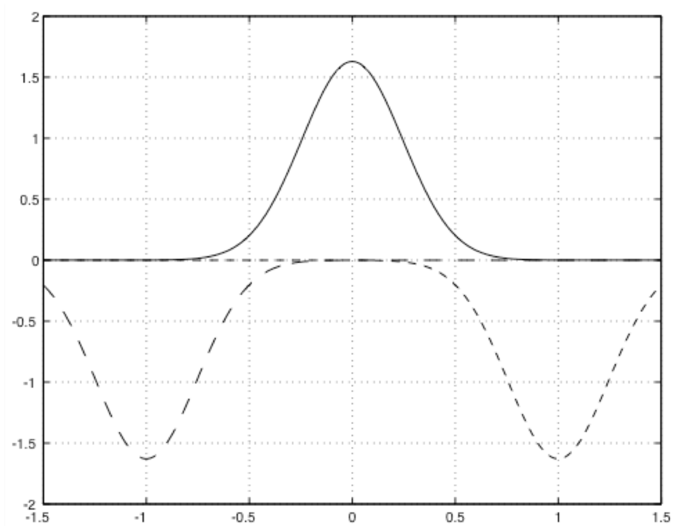
The method of images can be used to construct the solution

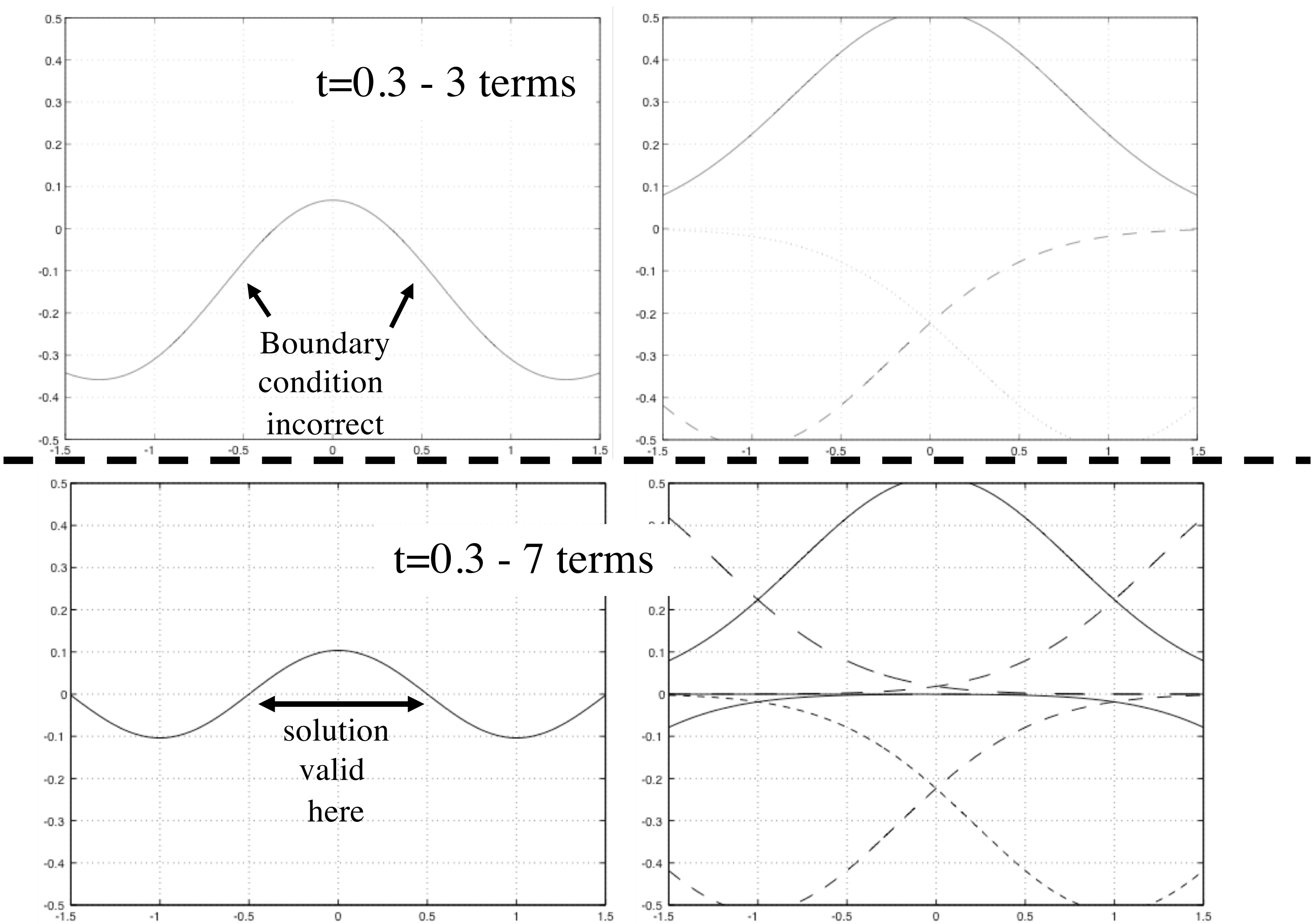
$$T(x, t) = \sum_{n=-\infty}^{\infty} (-1)^n T_G(x + nL, t)$$



t=0.03 - 3 terms







t=0.3 - 3 terms
Boundary
condition
incorrect
t=0.3 - 7 terms
solution
valid
here

Forward in time centered in space method (FTCS)

The simplest method to numerically solve an initial value problem is with the *Forward in Time Centered in Space* (FTCS) method

Rewrite the diffusion equation in finite difference form as

$$\frac{T_j^{n+1} - T_j^n}{\tau} = \kappa \frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{h^2}$$

Using the notation: $T_j^n \equiv T(x_j, t_n) = T(-\frac{L}{2} + (j-1)h, (n-1)\tau)$

We get

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

The FTCS method is an *explicit* method and is related to the Euler method for ODEs. It can be shown that the scheme is be stable when $\tau < \frac{h^2}{2\kappa}$

Aside: Derivation of the second derivative

We encountered this in Chapter 2 for the Verlet method in time. This is in space, but basically the same idea where we want a second order accurate finite different approximation to the derivative $\frac{d^2T}{dx^2}$

Using a central difference approximation on a grid of spacing h at index location j, this can be written as $\frac{d^2T}{dx^2} \approx \frac{d}{dx}\left(\frac{T_{j+\frac{1}{2}} - T_{j-\frac{1}{2}}}{h}\right)$

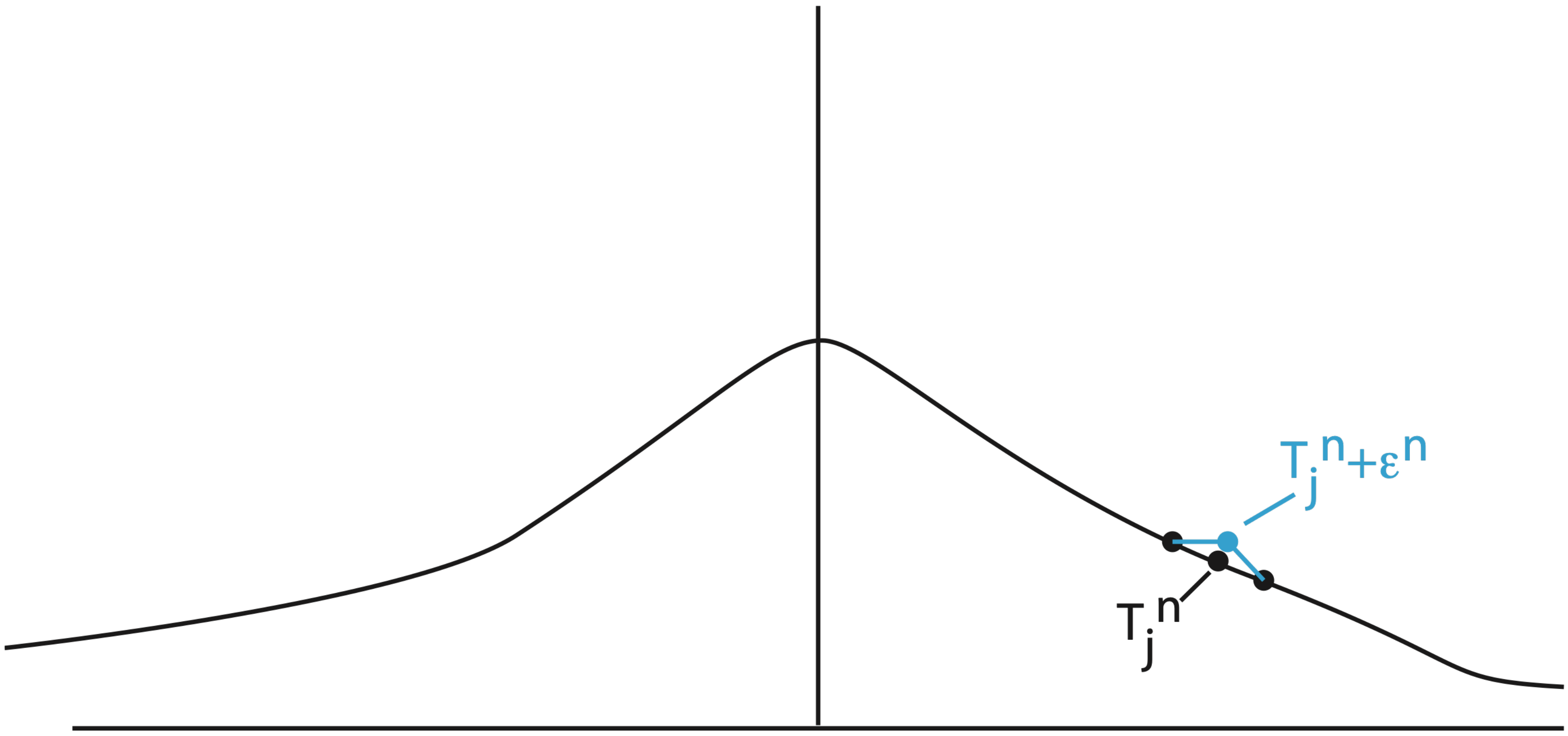
$$= \frac{1}{h}\left(\frac{dT_{j+\frac{1}{2}}}{dx} - \frac{dT_{j-\frac{1}{2}}}{dx}\right)$$

$$= \frac{1}{h}\left(\frac{T_{j+1} - T_j}{h} - \frac{T_j - T_{j-1}}{h}\right)$$

$$= \frac{T_{j+1} - 2T_j + T_{j-1}}{h^2}$$

Which is second order accurate in space

Stability analysis - Perturbation



Stability analysis - Perturbation

For the original equation $\frac{T_j^{n+1}-T_j^n}{\tau} = \kappa \frac{T_{j+1}^n-2T_j^n+T_{j-1}^n}{h^2}$

Assume that we have a solution T_j^n where we introduce a disturbance ε_j^n and see what happens to it by calculating the resulting perturbation at the next timestep ε_j^{n+1}We do this by adding the perturbation to the discretized diffusion equation as such

$$\frac{T_j^{n+1} + \varepsilon_j^{n+1} - T_j^n - \varepsilon_j^n}{\tau} = \kappa \frac{T_{j+1}^n - 2(T_j^n + \varepsilon_j^n) + T_{j-1}^n}{h^2}$$

Subtracting the top and the second equations we get and equation for ε

$$\frac{\varepsilon_j^{n+1}-\varepsilon_j^n}{\tau} = \kappa \frac{-2\varepsilon_j^n}{h^2}$$

Stability analysis - Perturbation -2

Or solving for ε_j^{n+1} we get

$$\varepsilon_j^{n+1} = \varepsilon_j^n(1 - \frac{2\tau\kappa}{h^2})$$

and taking the ratio

$$\frac{\varepsilon_j^{n+1}}{\varepsilon_j^n} = \left(1 - \frac{2\tau\kappa}{h^2}\right) = (1 - 2d)$$

where $d = \frac{\tau\kappa}{h^2}$

Since we want the perturbation to die out, we require that

$$\left|\frac{\varepsilon_j^{n+1}}{\varepsilon_j^n}\right| \leq 1$$

This means that the following 2 conditions must be satisfied

$$1 - 2d \geq -1 \text{ and } 1 - 2d \leq 1$$

Stability analysis - Perturbation -3

Taking the 2 conditions then

$$\Rightarrow 1 - 2d \geq -1$$
$$\Rightarrow \mathrm{d} \leq 1$$
$$\Rightarrow \frac{\tau\kappa}{h^2} \leq 1$$
$$\Rightarrow \tau \leq \frac{h^2}{\kappa}$$

Which basically puts a restriction on the timestep

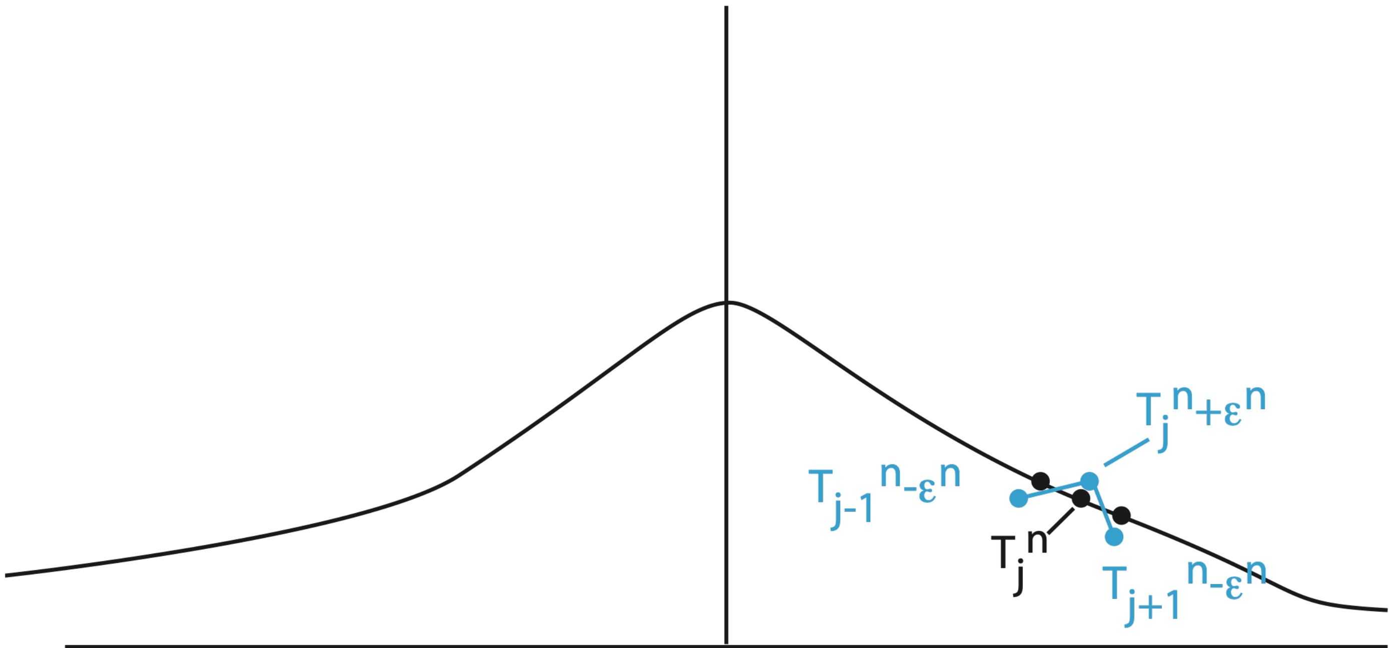
The second condition

$$1 - 2d \leq 1$$

Is always satisfied for $d > 0$

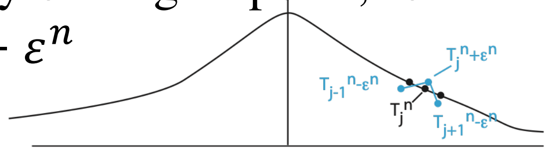
The first condition is not as restrictive as was mentioned on slide 13, that's because there is another perturbation that is more unstable, as discussed on the next slide.

Stability analysis part 2 - Perturbation



Stability analysis part 2 - Perturbation

A more restrictive case is where we have a perturbation on every other grid point, so that $\mathrm{T}_{\mathrm{j}}^{\mathrm{n}} \to \mathrm{T}_{\mathrm{j}}^{\mathrm{n}} + \varepsilon^{n}$ and $\mathrm{T}_{\mathrm{j+1}}^{\mathrm{n}} \to \mathrm{T}_{\mathrm{j+1}}^{\mathrm{n}} - \varepsilon^{n}$ and $\mathrm{T}_{\mathrm{j-1}}^{\mathrm{n}} \to \mathrm{T}_{\mathrm{j-1}}^{\mathrm{n}} - \varepsilon^{n}$



Putting this in the discretized diffusion equation, we get

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

Subtracting from the original diffusion equation yields

$$\frac{\varepsilon_j^{n+1} - \varepsilon_j^n}{\tau} = \kappa \frac{-4\varepsilon_j^n}{h^2} \text{ or } \frac{\varepsilon_j^{n+1}}{\varepsilon_j^n} = \left(1 - \frac{4\tau\kappa}{h^2}\right) = (1 - 4d)$$

Appling the same criteria as before that $\left|\frac{\varepsilon_j^{n+1}}{\varepsilon_j^n}\right| \leq 1$ for stability reduces to the more restrictive timestep that

$$\tau \leq \frac{h^2}{2\kappa}$$

Stability Analysis - Von Neumann

- Based on the analysis we did on the effect of the perturbation, which suggests that stability also depends on the type of perturbation, wherein the one with a perturbation that occurs in every other grid point suggests that there is a more general approach to stability analysis that depends on the spatial size of the perturbation.
- This approach is know as the Von Neumann stability analysis and will be discussed on the next slides.
- A more general approach to stability analysis that takes into account the effect of the boundary conditions will be introduced in chapter 9.

Stability Analysis - Von Neumann

This is a commonly used procedure where we look at the effect of the numerical method on all Fourier components, if we define the solution in the form

$$T_{j\pm1}^{n} = \mathrm{T^n e^{ikh(j\pm1)}} T_{j\pm1}^{n+1} = \mathrm{T^{n+1} e^{ikh(j\pm1)}}$$

where $i \equiv \sqrt{-1}$, k is the Fourier mode number, and j is the index location on the grid.

Putting this into the discretized version of the diffusion equation on slide 13 gives

$$T^{n+1}e^{ikhj} = T^{n}e^{ikhj} + dT^{n}e^{ikhj}(e^{-ikh} - 2 + e^{ikh})$$

Where $d = \frac{\kappa\tau}{h^2}$ as before. Defining an amplification factor A defined as $A = \frac{T^{n+1}}{T^n}$ we get

$$A = \frac{T^{n+1}}{T^{n}} = 1 + 2\mathrm{d}(\mathrm{coskh} - 1)$$

Von Neumann - continued

For the method to be stable, we would require that $|A| \leq 1$ for all k. Removing the absolute value sign gives us 2 cases:

$$1 + 2d(coskh - 1) \leq 1$$
$$\Rightarrow 2d(coskh - 1) \leq 0$$

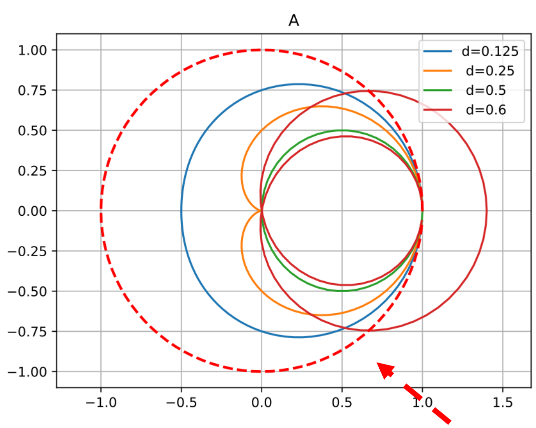
Which is always true, since cos is always less than or equal to 1

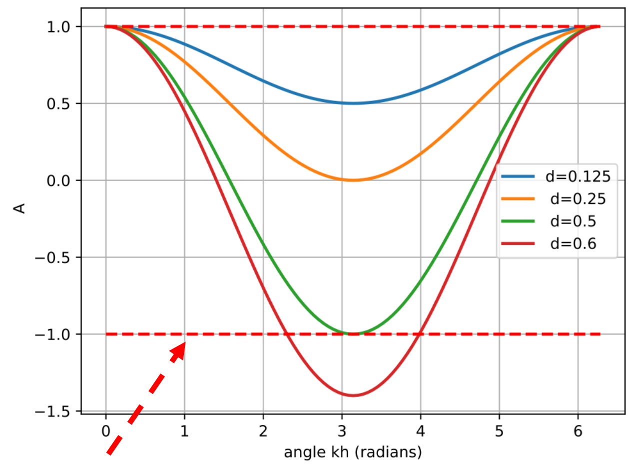
The other case must be true for all k, so we pick the smallest value of cos(kh), which is -1 so we get:

$$1 + 2d(coskh - 1) \geq -1$$
$$\Rightarrow 2d(coskh - 1) \geq -2$$
$$\Rightarrow 2d(-2) \geq -2$$
$$\Rightarrow d \leq \frac{1}{2} \ or$$
$$\tau \leq \frac{h^2}{2\kappa}$$

Which is the stability condition derived earlier.

Another way to look at the stability is the plot the amplification factor A as a function of kh. Below is a plot of the amplification factor A as a for different values of d plotted in polar coordinates and as a line plot. (Uses the program ‘vn_demo.m’)





Method is unstable if line crosses a circle of *radius*=1 (red- dashed line) which happens if $d > 0.5$

Neutron Diffusion
(Garcia, section 6.3)

This section describes a model of neutron interaction with a fissionable material such as ^{235}U. The basic equation that describes the neutron density in the material is given by

$$\frac{\partial n(r,t)}{\partial t} = D\nabla^2 n(r,t) + Cn(r,t)$$

Where n(r,t) is the neutron density, D is the diffusion constant and C is the creation rate for neutrons. Typical values are D~10^5 m^2/s and C~10/s (In the program, D=C=1). For a 1-D model, we will solve

$$\frac{\partial n(x,t)}{\partial t} = D\,\frac{\partial^2 n(x,t)}{\partial x^2} + Cn(x,t)$$

Along a 1D rod of length L, centered at the origin.

Neutron Diffusion -2

The boundary condition is if the neutrons encounter the boundary they will escape, which is modeled by

$$n\left(-\frac{L}{2}, t\right) = n\left(\frac{L}{2}, t\right) = 0$$

Which is the Dirichlet Boundary condition. The analytic solution is via separation of variables, by assuming that

$$n(x, t) = X(x)T(t)$$

The equation then becomes

$$X\frac{\partial T}{\partial t} = DT\frac{\partial^2 X}{\partial x^2} + CXT$$

or

$$\frac{1}{T}\frac{\partial T}{\partial t} = \frac{D}{X}\frac{\partial^2 X}{\partial x^2} + c$$

Neutron Diffusion - 3

Since the LHS only depends on t and the RHS on x, the equations are equal to a constant α

$$\frac{1}{T}\frac{\partial T}{\partial t} = \frac{D}{X}\frac{d^2X}{dx^2} + C = \alpha$$

The solution is then

$$T(t) = T(0)e^{\alpha t}$$

And

$$\frac{d^2X}{dx^2} = \frac{\alpha - C}{D}x$$

The spatial solution that satisfies the boundary conditions is

$$X(x) = \sum_{j=0}^{\infty} a_j cos\left(\frac{(2j+1)\pi}{L}x\right)$$

Neutron Diffusion - 4

Putting this back into the equation for X and equating coefficients we get

$$-\left(\frac{(2j+1)\pi}{L}\right)^2 = \frac{\alpha - C}{D}$$

[Note: this is a bit different from the book, but I think equivalent and simpler.] For exponential growth, we need $\alpha > 0$, which, solving the above equation for α, gives

$$C - D\left(\frac{(2j+1)\pi}{L}\right)^2 > 0$$

This inequality is satisfied for any j, if L is longer than the critical length L_c defined as

$$L_c = \pi\sqrt{\frac{D}{C}}$$

Neutron - Program

The MATLAB program 'neutrn.m' simulates the 1D diffusion equation described above. It uses the FTCS method used for the diffusion equation with a delta function initial condition. The basic timestep, in the notation of the diffusion equation is

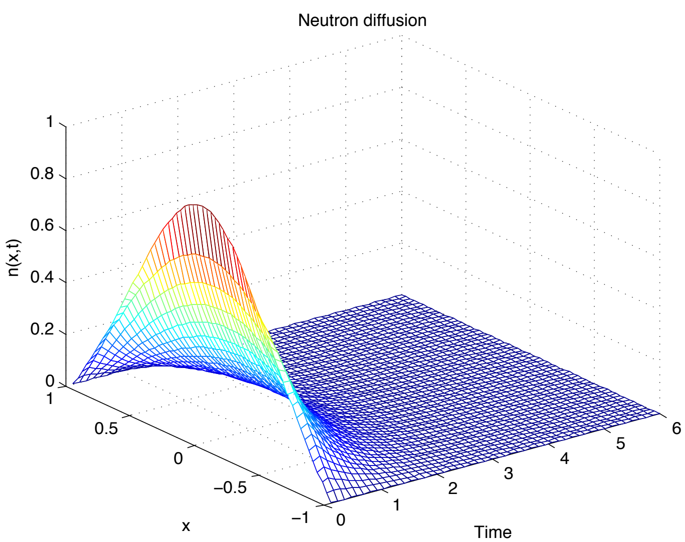
$$n_i^{n+1} = n_i^n + \frac{D\tau}{h^2}(n_{i+1}^n - 2n_i^n + n_{i-1}^n) + C\tau n_i^n$$

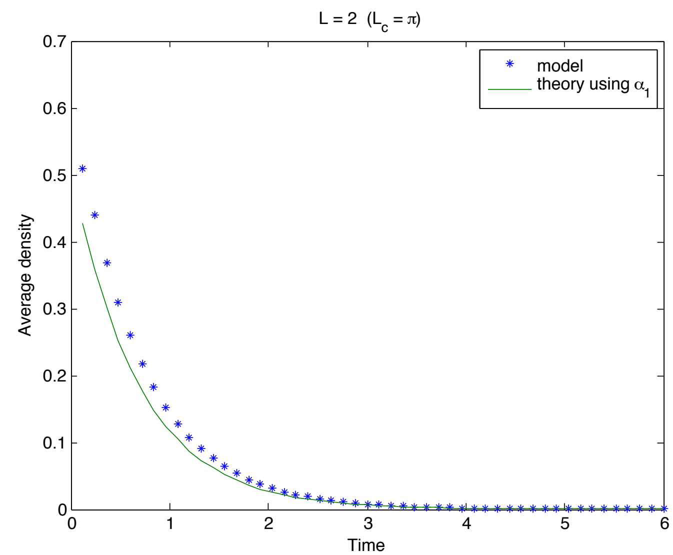
Where $x_i = (i-1)h - \frac{L}{2}$ and $t_n = (n-1)\tau$, $h = \frac{L}{N-1}$ where n is the timestep and N is the number of grid points. The code also outputs the average value of defined as which decays as (*j=1*)

$$\bar{n}^n \equiv \frac{1}{N}\sum_{i=1}^{N} n_i^n$$

And the analytic solution as $\overline{n(t)} \sim e^{\alpha_1 t}$ where $\alpha_1 = C - \frac{D\pi^2}{L^2}$

Example from neutrn.m
(L=2, τ=5x10^{-4},12000 timesteps)





Example from neutrn.m – critical example ($L=4$, $\tau=5\text{x}10^{-4}$,12000 timesteps)