Heat Flow in Space and Time

Time-Stepping Via the Leap Frog Algorithm

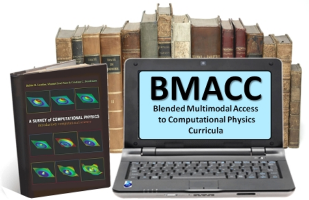
Rubin H Landau

Sally Haerer, Producer-Director

Based on *A Survey of Computational Physics* by Landau, Páez, & Bordeianu

with Support from the National Science Foundation

Course: **Computational Physics II**



Problem: How Does a Bar Cool?

Insulated Metallic Bar Touching Ice

- Aluminum bar, $L = 1$ m, w along x
- Insulated along length, ends in ice ($T = 0$ C)
- Initially $T = 100$ C
- How does temperature vary in space and time?

The Parabolic Heat Equation (Theory)

1. Nature: heat flow hot $\rightarrow$ cold
   K = conductivity
   C = sp heat, ρ = density
2. $Q(t)$ = contained heat
3. Heat Eqn: ΔT from flow
4. Parabolic PDE in x & t
5. "Analytic" Solution

   IC: $T(x, t = 0) = 100\text{C}$
   BC: $T(x = 0) = T(x = L) = 0\text{C}$

$$\mathbf{H} = -K\,\nabla T(\mathbf{x}, t) \tag{1}$$

$$Q(t) = \int d\mathbf{x}\, C\rho(\mathbf{x})\, T(\mathbf{x}, t) \tag{2}$$

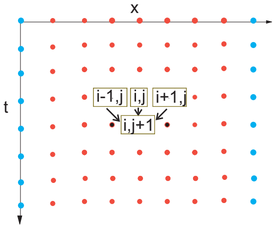
$$\frac{\partial T(\mathbf{x}, t)}{\partial t} = \frac{K}{C\rho}\nabla^2 T(\mathbf{x}, t) \tag{3}$$

$$\frac{\partial T(x, t)}{\partial t} = \frac{K}{C\rho}\frac{\partial^2 T(x, t)}{\partial x^2} \tag{4}$$

$$T(x, t) = \sum_{n=1,3,\ldots}^{\infty} \frac{400 \sin\, k_n x\, e^{-\alpha k_n^2 t}}{n\pi} \tag{5}$$

$$(k_n = \frac{n\pi}{L},\ \alpha = \frac{K}{C\rho}) \tag{6}$$

Solution Via Time Stepping



- Differential $\rightarrow$ difference eqtn
- Solve at $x - t$ lattice sites
- Vert blue = BC, row 0 = IC
- Relax: if knew T_{bottom}
- Leapfrog $\downarrow$ one t to next
- FD ∂t, CD for $\partial^2 x$ (can)
- $\Rightarrow$ difference heat eqtn
- *Explicit Soltn*: known values
- Not space-time symmetric

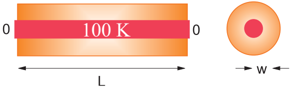
$$\frac{\partial T}{\partial t} \simeq \frac{T(x, t + \Delta t) - T(x, t)}{\Delta t}$$

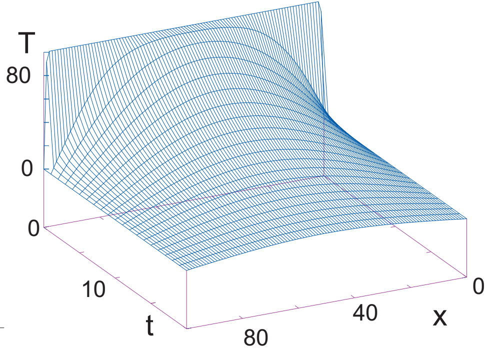
$$\frac{\partial^2 T}{\partial x^2} \simeq \frac{T(x + \Delta x) + T(x - \Delta x) - 2T(x)}{(\Delta x)^2}$$

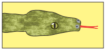
$$\frac{T(x, t + \Delta t) - T(x, t)}{\Delta t} = \frac{K}{C\rho} \frac{T(x + \Delta x, t) + T(x - \Delta x, t) - 2T(x, t)}{\Delta x^2} \quad (1)$$

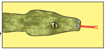
$$T_{i,j+1} = T_{i,j} + \eta \left[T_{i+1,j} + T_{i-1,j} - 2T_{i,j}\right] \quad (2)$$

Solution of Heat Equation





EqHeat.py 

EqHeat_Animate.py 

Von Neumann Stability Analysis

$$T_{m,j+1} = T_{m,j} + \eta\,[T_{m+1,j} + T_{m-1,j} - 2T_{m,j}]\,, \qquad x = m\Delta x,\ \ t = j\Delta t \qquad (1)$$

- Difference soltn $\simeq$ PDE soltn??
- Bad: difference diverges

$$T_{m,j} = \xi(k)^j\, e^{ikm\Delta x} \qquad (2)$$

$$\Rightarrow \quad \xi(k) = 1 + 2\eta[\cos(k\Delta x) - 1] \qquad (3)$$

$$|\xi(k)| < 1 \qquad (4)$$

$$\Rightarrow \ \eta = \frac{K\,\Delta t}{C\rho\,\Delta x^2} < \frac{1}{2} \qquad (5)$$

- Assume $T_{m,j}$ = eigenmodes
- $k = 2\pi/\lambda = ?$
- Stable if eigenmodes stable
- i.e. $|\xi(k)| < 1$
- Sub (3) into diff eqtn (1)
- $\Rightarrow$ Smaller Δt more stable
- $\downarrow \Delta x$ must $\uparrow \Delta t$
- Always try analysis

Implementation

EqHeat.py CODE

- Build in BC & IC
- Heart: 2-D array `T[101][2] = T[x][present, future]`
- Set future to present, calculate future
- Output t & T ever 300 t steps
- Surface $T(x, t)$ plots with *isotherms*, must be smooth
- Vary Δt & Δx
- Compare analytic & numeric solutions
- **Stability:** Diverges $\eta > \frac{1}{4}$?

Crank–Nicolson Algorithm Next

Take a break, or quit if not proceeding to Crank Nicolson.