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?

- 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
   $K = \text{conductivity}$
   $C = \text{sp heat, } \rho = \text{density}$

2. $Q(t) = \text{contained heat}$

3. Heat Eqn: $\Delta T$ from flow

4. Parabolic PDE in $x$ & $t$

5. “Analytic” Solution

   **IC:** $T(x, t = 0) = 100\,^\circ C$
   **BC:** $T(x = 0) = T(x = L) = 0\,^\circ C$

\[
\mathbf{H} = -K \nabla T(x, t) \quad \quad (1)
\]
\[
Q(t) = \int d\mathbf{x} \, C\rho(\mathbf{x}) \, T(\mathbf{x}, t) \quad \quad (2)
\]
\[
\frac{\partial T(\mathbf{x}, t)}{\partial t} = \frac{K}{C\rho} \nabla^2 T(\mathbf{x}, t) \quad \quad (3)
\]
\[
\frac{\partial T(x, t)}{\partial t} = \frac{K}{C\rho} \frac{\partial^2 T(x, t)}{\partial x^2} \quad \quad (4)
\]
\[
T(x, t) = \sum_{n=1,3,\ldots}^{\infty} \frac{400 \sin k_n x \, e^{-\alpha k_n^2 t}}{n\pi} \quad \quad (5)
\]
\[
(k_n = \frac{n\pi}{L}, \quad \alpha = \frac{K}{C\rho}) \quad \quad (6)
\]
Solution Via Time Stepping

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

\[
\frac{\partial T}{\partial t} \approx \frac{T(x, t + \Delta t) - T(x, t)}{\Delta t}
\]

\[
\frac{\partial^2 T}{\partial x^2} \approx \frac{T(x + \Delta x) + T(x - \Delta x) - 2T(x)}{(\Delta x)^2}
\]

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

\[
T_{i,j+1} = T_{i,j} + \eta [T_{i+1,j} + T_{i-1,j} - 2T_{i,j}] \] (2)
Solution of Heat Equation

```
EqHeat.py
```

```
EqHeat_Animate.py
```
Von Neumann Stability Analysis

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

- Difference soltn \( \simeq \) PDE soltn??
- Bad: difference diverges
  
  \[ T_{m,j} = \xi(k)^j e^{ikm\Delta x} \quad (2) \]

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

  \[ |\xi(k)| < 1 \quad (4) \]

  \[ \Rightarrow \quad \eta = \frac{K\Delta t}{C\rho \Delta x^2} < \frac{1}{2} \quad (5) \]

\[ \Rightarrow \quad k = \frac{2\pi}{\lambda} = ? \]

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

EqHeat.py

- Build in BC & IC
- Heart: 2-D array $T[101][2] = T[x][\text{present, future}]$
- Set future to present, calculate future
- Output $t$ & $T$ ever 300 $t$ steps
- Surface $T(x, t)$ plots with isotherms, must be smooth
- Vary $\Delta t$ & $\Delta x$
- Compare analytic & numeric solutions
- **Stability:** Diverges $\eta > \frac{1}{4}$?
Take a break, or quit if not proceeding to Crank Nicolson.