Recap: FTCS for the 1D Heat Equation

We consider the one-dimensional heat equation

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

Using finite differences in space and an explicit time update, the *Forward Time, Centered Space (FTCS)* scheme is

$$T_j^{n+1} = T_j^n + \lambda \left(T_{j+1}^n - 2T_j^n + T_{j-1}^n \right), \qquad \lambda = \frac{\alpha \, \Delta t}{\Delta x^2}.$$

Key features:

- Fully explicit
- Easy to implement
- Conditionally stable

FTCS Stability: What We Know

For the one-dimensional heat equation, FTCS is stable only if

$$\lambda = \frac{\alpha\,\Delta t}{\Delta x^2} \le \frac{1}{2}.$$

Implications:

- Time step is restricted by spatial resolution
- Small $\Delta x \Rightarrow$ very small Δt
- Long-time simulations can become expensive

This motivates the use of *implicit* time integration schemes.

Motivation for Crank–Nicolson

FTCS evaluates spatial diffusion entirely at time level n.

Crank–Nicolson evaluates diffusion at the *midpoint in time*:

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha \frac{1}{2} \left(\frac{\partial^2 T^{n+1}}{\partial x^2} + \frac{\partial^2 T^n}{\partial x^2} \right).$$

Interpretation:

- Average of explicit and implicit diffusion
- Second-order accurate in time
- Improved stability properties

Crank–Nicolson: Discrete Form

Using centered finite differences in space, we obtain

$$T_j^{n+1} - \frac{\lambda}{2}\left(T_{j+1}^{n+1} - 2T_j^{n+1} + T_{j-1}^{n+1}\right) = T_j^n + \frac{\lambda}{2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right).$$

Unlike FTCS, all unknown temperatures at time level $n+1$ appear on the *left-hand side*.

This leads to a linear system that must be solved at each time step.

Crank–Nicolson as a Linear System

At each time step, we solve

$$\mathbf{A}\,\mathbf{T}^{n+1} = \mathbf{d}.$$

The coefficient matrix $\mathbf{A}$ has a special structure:

$$\mathbf{A} = \begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ 0 & & & a_N & b_N \end{bmatrix}.$$

This is called a *tridiagonal* matrix.

Structure of the Linear System

The Crank–Nicolson system has the form

$$
\begin{bmatrix}
b_1 & c_1 & & & 0 \\
a_2 & b_2 & c_2 & & \\
& a_3 & b_3 & c_3 & \\
& & \ddots & \ddots & \ddots \\
0 & & & a_N & b_N
\end{bmatrix}
\begin{bmatrix}
T_1^{n+1} \\ T_2^{n+1} \\ \vdots \\ T_N^{n+1}
\end{bmatrix}
=
\begin{bmatrix}
d_1 \\ d_2 \\ \vdots \\ d_N
\end{bmatrix}.
$$

- Lower diagonal: a_i
- Main diagonal: b_i
- Upper diagonal: c_i

Looking Ahead

- Crank–Nicolson is unconditionally stable for linear diffusion
- Each time step requires solving a tridiagonal system
- The special matrix structure enables very efficient solvers

Coming next:

- Efficient solution of tridiagonal systems
- Implementation details in Python
- Comparison of FTCS and Crank–Nicolson on real problems