EMPossible

Computational Science:
Computational Methods in Engineering

Introduction to Time-Domain Finite-Difference Method

Outline

- Introduction
- Time-Domain Solution of the Heat Equation

Introduction

Basic Approach

Formulation

1. Identify governing equation
2. Approximate equation using finite-differences
3. Solve finite-difference equation for the function at the future time-value
4. Collect constants into update coefficients.
5. Write final update equation.

Implementation

1. Dashboard
2. Calculate grid
3. Build device on grid
4. Calculate update coefficients
5. Initialize unknown function(s)
6. Main loop – iterate over time
7. Post process data
8. Visualize data

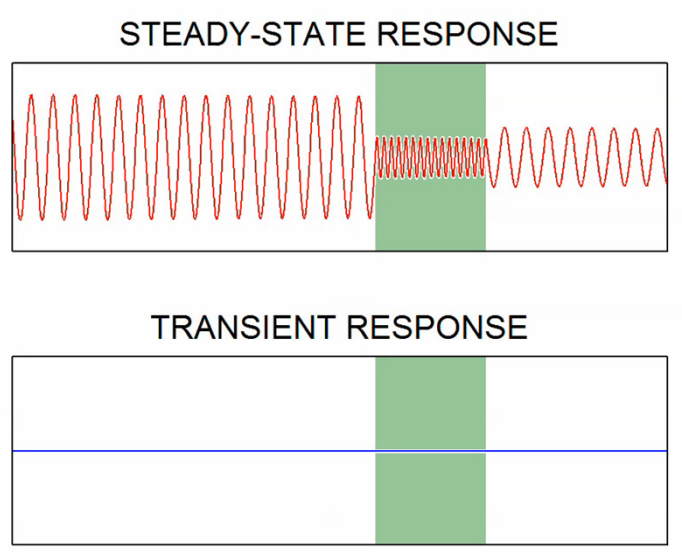
EMPossible

Notes

- Time-domain FDM does not require linear algebra.
- Scales nearly linearly
- Excellent technique for very large scale simulations
- Excellent for simulating transient response
- Excellent for incorporating nonlinear devices and materials
- Excellent technique for visualizing motion and waves
- Excellent technique for learning about devices

EMPossible

Transient Vs. Steady-State

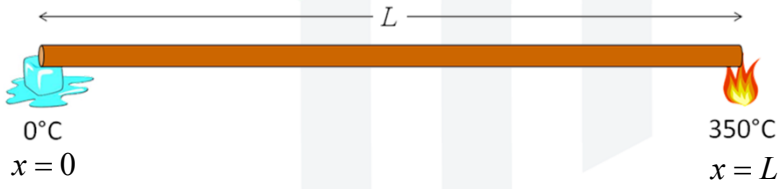


EMPossible

Time-Domain Solution of the Heat Equation

Define Problem

A long bar of length L is a temperature 0 °C. At time t = 0, a 350 °C heat source is applied to the far end of the bar while the other end is held at 0 °C. Calculate and show how the temperature evolves with time over the length of the bar if the thermal diffusivity $\alpha(x)$ is



$$\alpha(x) = 25 + 50\sin\left(\frac{\pi x}{L}\right) \qquad 0 \le x \le L$$

EMPossible

Governing Equation

The governing equation is the time-domain form of the heat equation.

$$\frac{\partial T}{\partial t} - \alpha \nabla^2 T = 0$$

$T(x) \equiv$ temperature

$\alpha(x) \equiv$ thermal diffusivity

$t \equiv$ time

$\nabla \equiv$ del operator

Material	α (m²/s)
Metal	10^{-4}
Water	10^{-7}
Air	10^{-5}
Nylon	10^{-7}
Wood	10^{-7}

EMPossible

9

Reduce to 1D

$$\frac{\partial T}{\partial t} - \alpha \nabla^2 T = 0$$

$$\downarrow$$

$$\frac{\partial T(x,t)}{\partial t} - \alpha(x)\left[\frac{\partial^2 T(x,t)}{\partial x^2} + \cancel{\frac{\partial^2 T(x,t)}{\partial y^2}} + \cancel{\frac{\partial^2 T(x,t)}{\partial z^2}}\right] = 0$$

$$\downarrow$$

$$\frac{\partial T(x,t)}{\partial t} - \alpha(x)\frac{\partial^2 T(x,t)}{\partial x^2} = 0$$

EMPossible

10

Approximate with Finite-Differences

The governing equation is

$$\frac{\partial T(x,t)}{\partial t} - \alpha(x)\frac{\partial^2 T(x,t)}{\partial x^2} = 0$$

Make a guess at the finite-difference approximation

$$\underbrace{\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t}}_{\text{Exists at } (x,\,t+\Delta t/2)} - \alpha(x)\underbrace{\frac{T(x+\Delta x,t)-2T(x,t)+T(x-\Delta x,t)}{(\Delta x)^2}}_{\text{Exists at } (x,\,t)} = 0$$

Exists at (x,t)

Not all finite-difference approximations in this equation exist at the same instant in time.

EMPossible

11

Fixing the Finite-Difference Equation (1 of 2)

Solution 1 – Interpret the time finite-difference as a backward finite-difference.

$$\underbrace{\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t}}_{\text{Exists at } (x,t)} - \alpha(x)\frac{T(x+\Delta x,t)-2T(x,t)+T(x-\Delta x,t)}{(\Delta x)^2} = 0$$

Solution 2 – Interpret the time finite-difference as a forward finite-difference.

$$\underbrace{\frac{T(x,t)-T(x,t-\Delta t)}{\Delta t}}_{\text{Exists at } (x,t)} - \alpha(x)\frac{T(x+\Delta x,t)-2T(x,t)+T(x-\Delta x,t)}{(\Delta x)^2} = 0$$

EMPossible

12

Fixing the Finite-Difference Equation (2 of 2)

Solution 3 – Interpolate the second term in the equation to exist at $t + \Delta t/2$.

$$\underbrace{\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t}}_{\text{Exists at } (x,\, t+\Delta t/2)} - \underbrace{\frac{\alpha(x)\dfrac{T(x+\Delta x,t+\Delta t)-2T(x,t+\Delta t)+T(x-\Delta x,t+\Delta t)}{(\Delta x)^2}+\alpha(x)\dfrac{T(x+\Delta x,t)-2T(x,t)+T(x-\Delta x,t)}{(\Delta x)^2}}{2}}_{\text{Exists at } (x,\, t+\Delta t/2)} = 0$$

This is called the *Crank-Nicholson* scheme.

Very often, Solution 1 or 2 can be used successfully, but the Crank-Nicholson scheme is unconditionally stable.

EMPossible

13

Solve for Temperature at Future Step

Proceed with Solution 1 because it is the simplest, but not necessarily the most accurate or stable.

$$\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t}-\alpha(x)\frac{T(x+\Delta x,t)-2T(x,t)+T(x-\Delta x,t)}{(\Delta x)^2}=0$$

Write this equation in terms of array indices where k is the time step (i.e. $t = k\Delta t$).

$$\frac{T_i^{k+1}-T_i^k}{\Delta t}-\alpha_i\frac{T_{i+1}^k-2T_i^k+T_{i-1}^k}{(\Delta x)^2}=0$$

Solve for temperature at the future time step T_i^{k+1}.

$$T_i^{k+1}=T_i^k+\frac{\alpha_i\Delta t}{(\Delta x)^2}\left(T_{i+1}^k-2T_i^k+T_{i-1}^k\right)$$

EMPossible

14

Write Update Equation

The update equation is

$$T_i^{k+1} = T_i^k + c_i\left(T_{i+1}^k - 2T_i^k + T_{i-1}^k\right)$$

When performing the update, calculate the new values based only on old values. Do not mix the two!

It is usually necessary to store the old and the new values in two different arrays to do this.

The update coefficient is

$$c_i = \frac{\alpha_i \Delta t}{(\Delta x)^2}$$

The update coefficients c_i do not change with time so they should be calculated before the main time loop to improve speed and efficiency.

EMPossible

15

Stability Condition (1 of 2)

The finite-difference equation couples only the immediately adjacent points on the grid. Therefore, it is impossible for numbers to propagate across the grid faster than one cell Δx in on time step Δt. This defines the numerical speed v_{FDM} as

$$v_{\text{FDM}} \cong \frac{\Delta x}{\Delta t}$$

The speed v_{heat} at which a physical heat front would diffuse through a 1D medium is

$$v_{\text{heat}} \cong \frac{\alpha_i}{\Delta x}$$

EMPossible

16

Stability Condition (2 of 2)

In order for the simulation to be stable, the numerical parameters must be chosen so that the physical speed of the heat front cannot is not faster than is numerically possible on the grid.

$$v_{\text{heat}} < \frac{1}{2} v_{\text{FDM}}$$

A factor of $\frac{1}{2}$ was included as a safety margin.

Substituting in the expressions for speed gives

$$\frac{\alpha_i}{\Delta x} < \frac{1}{2}\frac{\Delta x}{\Delta t}$$

Solving this for Δt gives the stability condition.

$$\boxed{\Delta t < \frac{(\Delta x)^2}{2\alpha_i}}$$

EMPossible

17

Revised Algorithm

1. Initialize MATLAB
2. Dashboard – define all simulation parameters
3. Calculate grid: `Nx` and `dx`.
4. Build device on grid: `alpha`
5. Calculate a stable time step (and number of iterations): `dt` and `NT`
6. Calculate update coefficients: $[c_1\ c_2\ c_3\ \ldots\ c_N]$
7. Initialize arrays: `T`, `T2`, etc.
8. Main loop – iterate over time t
   a. Update T_i all the way across grid (loop over x)
   b. Enforce boundary conditions
   c. Record intermediate results (if needed)
   d. Visualize intermediate results (if needed)
9. Post process results
10. Visualize results
11. Finished!

EMPossible

18