Course Instructor
Dr. Raymond C. Rumpf
Office: A-337
Phone: (915) 747-6958
E-Mail: rcrumpf@utep.edu

Computational Methods in EE

Topic 7f –

# Time-Domain FDM

*EE 4386/5301 Computational Methods in EE*



## Outline

Computational Methods in EE

- Introduction
- Example #1

Computational Methods in EE

# Introduction



## Basic Algorithm

Computational Methods in EE

**Formulation**

1. Identify governing equation
2. Approximate equation using finite-differences
3. Solve finite-difference equation for 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. Initial unknown function(s)
6. Main loop – iterate over time
7. Post process data
8. Visualize data

# Notes

Computational Methods in EE

- Time-domain FDM does not require linear algebra.
- Scales nearly linearly
- Excellent technique for very large scale simulations
- Excellent for simulating transient response
- Excellent technique for learning about devices



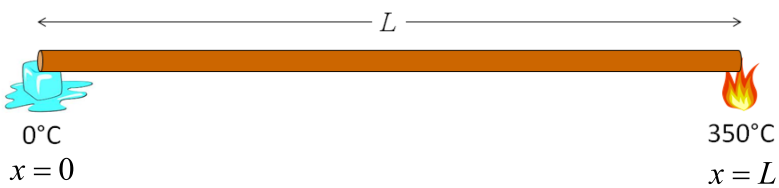
Computational Methods in EE

# Example #1

# Define Problem

Computational Methods in EE

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. Calculate and show how the temperature evolves with time over the length of the bar if the thermal diffusivity $\alpha(x)$ is




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



# Governing Equation

Computational Methods in EE

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 | $\alpha$ |
|---|---|
| Metal | $10^{-4}$ |
| Water | $10^{-7}$ |
| Air | $10^{-5}$ |
| Nylon | $10^{-7}$ |
| Wood | $10^{-7}$ |

# Reduce to 1D

Computational Methods in EE

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

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



# Approximate with Finite-Differences

Computational Methods in EE

Over governing equation is

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

Let's make a guess at the finite-difference approximation

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

Exists at $(x,\ t + \Delta t/2)$

Exists at $(x,\ t)$

Exists at $x$

# Fixing the Finite-Difference Equation (1 of 2)

Computational Methods in EE

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

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



# Fixing the Finite-Difference Equation (2 of 2)

Computational Methods in EE

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

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

This is called the Crank-Nicholson scheme.

## Solve for Future Value

Computational Methods in EE

We will choose Solution 1 because it is the simplest, but not necessarily the most accurate.

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

We now write this equation in terms of array indices.

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

Solve for future value of $T$.

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



## Write Update Equation

Computational Methods in EE

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

The update coefficient is

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

The update coefficient does not change with time so it should be calculated before the main time loop.

# Stability Condition

Computational Methods in EE

Our finite-difference equations couples only adjacent points on the grid.

Thus, it is numerically impossible for a number to travel farther than one grid cell in one time step.

We need to make sure that the speed of the heat front is slower than what is numerically possible.

The stability condition that ensures this is

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

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

Typically we calculate the time step based on this stability condition.



# Revised Algorithm

Computational Methods in EE

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`, `DT2`, 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!