Transient 1D Conductive Heat Transfer

Wayne Pafko
11/19/01

Background

This Mathcad document shows how to use an finite difference algorithm to solve an intial valu transient heat transfer problem involving conduction in a slab. In this problem, the temperature the slab is initially uniform (Initial Condition). The edges are then instantly changed to a const temperature boundary condition (Dirichlet BC). The finite difference algorithm then calculates how the temperature profile in the slab changes over time.

Physical Model

Conduction of heat in a slab is usually described using a parabolic partial differential equation.

$$\frac{d}{dt}T = \lambda \cdot \frac{d^2}{dx^2}T$$ Transient conduction of heat in a slab.

This partial differential equation can be approximated using finite differences...

$$\frac{\delta T}{\Delta t} = \lambda \cdot \frac{\Delta(\Delta T)}{(\Delta x)^2}$$ Where... $\Delta(\Delta T) = (T_{i+1} - T_i) - (T_i - T_{i-1}) = T_{i+1} - 2 \cdot T_i + T_{i-1}$

$\delta T = T_{j+1} - T_j$ ...with i indexing across the node and j indexing over time.

If the time step and nodal spacing is sufficiently small the finite difference solution will approa analytical solution and we will find the numerical solution to our partial differential equation.

Problem Setup

1) Input the physical properties the slab (density, heat capacity, & thermal conductivity)...

$\rho := 1 \cdot \frac{gm}{m^3}$ $Cp := 5 \cdot \frac{J}{gm \cdot K}$ $k := 2 \cdot 10^{-5} \cdot \frac{W}{m \cdot K}$

$\lambda := \frac{k}{\rho \cdot Cp}$ $\lambda = 4 \times 10^{-6} \frac{m^2}{s}$ Thermal diffusivity.

2) Input finite difference parameters (time step, node spacing, # nodes, # time steps)...

$\Delta time := 0.1 \cdot sec$ $\Delta x := 0.1 \cdot cm$ $\lambda \cdot \frac{\Delta time}{\Delta x^2} = 0.4$

Note: This term should be less than 0.5 to ensure stability, less than 0.25 to ensure convergence, and about 0.166 to minimize truncation error (similar to a Courant-Friedrishs-Lewy stability criterion).

$timesteps := 600$ $nodes := 30$

$timesteps \cdot \Delta time = 60\,s$ Total simulation time

$nodes \cdot \Delta x = 3\,cm$ Total simulation width

$i := 0 .. (nodes - 1)$

3) Input initial conditions for the slab.

$Tinitial_i := 300 \cdot K$ $Tinitial^T =$

	0	1	2	3	4	5	6	7	8
0	300	300	300	300	300	300	300	300	300

K

4) Input left & right boundary conditions.

$BC_{left} := 350K$ $BC_{right} := 440 \cdot K$

Problem Solution

5) Solve the problem using a finite difference algorithm.

When you pass a vector containing the temperature at each node to the "updateTemps" functio calculates what the temperature at each node will be at the next time step and returns that vecto updated temperatures. Note that the temperature at the edges are always just set to equal the le and right boundary conditions (Dirichlet BC).

$$updateTemps(T) := \left| \begin{array}{l} Temp_0 \leftarrow BC_{left} \\ Temp_{nodes-1} \leftarrow BC_{right} \\ \text{for } i \in 1 .. nodes - 2 \\ \quad Temp_i \leftarrow T_i + \lambda \cdot \frac{\Delta time}{\Delta x^2} \cdot \left(T_{i-1} - 2 \cdot T_i + T_{i+1}\right) \\ Temp \end{array} \right.$$

The "SolveFiniteDifference" function first sets the temperature at each node to be equal to the temperature. It then loops over the specified number of time steps and calculates what the temperature profile will be at each step using the "updateTemps" function. This is stored in a s array. Each row contains the temperature profile for that time step.

$$SolveFiniteDifference(steps) := \left| \begin{array}{l} T^{\langle 0 \rangle} \leftarrow Tinitial \\ \text{for } i \in 1 .. floor(steps) \\ \quad T^{\langle i \rangle} \leftarrow updateTemps\left(T^{\langle i-1 \rangle}\right) \\ T \end{array} \right.$$

Tanswer := SolveFiniteDifference(timesteps)

The "Tanswer" array contains solution. Each row contains a temperature profile for a given tin step. There are as many rows as time steps and as many columns as nodes (array is only partia shown, scroll around to see entire solution).

$Tanswer^{T} =$ K

	0	1	2	3	4	5	6	7
0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0
1	350.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0
2	350.0	320.0	300.0	300.0	300.0	300.0	300.0	300.0
3	350.0	324.0	308.0	300.0	300.0	300.0	300.0	300.0
4	350.0	328.0	311.2	303.2	300.0	300.0	300.0	300.0
5	350.0	330.1	314.7	305.1	301.3	300.0	300.0	300.0
6	350.0	331.9	317.0	307.4	302.3	300.5	300.0	300.0
7	350.0	333.2	319.1	309.2	303.6	301.0	300.2	300.0
8	350.0	334.3	320.8	311.0	304.8	301.7	300.5	300.1

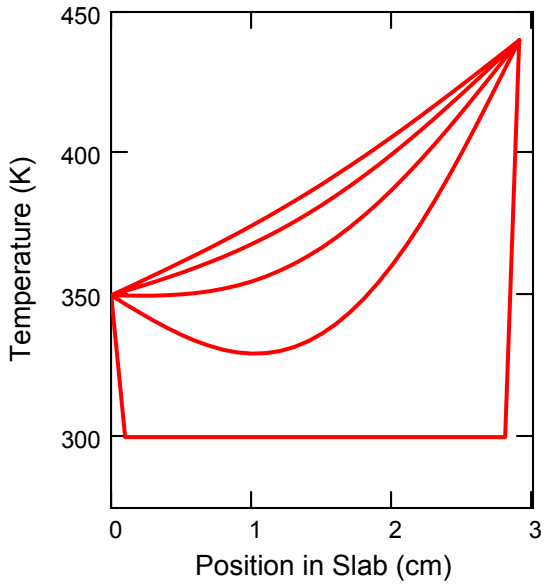
The last row contains the temperature profile at the end of the simulation. This may or may no the steady-state solution depending on how much simulation time elapses.

$Tanswer^{\langle timesteps-1\rangle^{T}} =$ K

	0	1	2	3	4	5	6
0	350.0	352.3	354.6	357.0	359.4	361.8	364.2

6) View the results

$x_i := i \cdot \Delta x$ Position of each node in slab



We can also animate the results...

time := 10·FRAME·Δtime Calculate time at each FRAME

$\frac{\text{timesteps}}{10} = 60$ Animate FRAME up to this number.