Finite Difference Model for Joule Heating

Tristan Wolff Schwab

April 13, Spring 2022

Introduction

In complex multiphysical problems with many variables it is particularly useful to model the problem in simulation. The author implements the Finite Difference Method to model the heat equation in 1D to simulate transient thermal effects of a process. Examples of transient thermal effects in a manufacturing process include spot welding, sintering, and laser engraving. In part one, a model for heat transfer under various boundary conditions is applied using a Finite Difference Method solution scheme. In part two, the process is optimized using a genetic algorithm to find the optimal system parameters for the Joule-heating process that closely matches the desired results.

Methodology

First Law of Thermodynamics: Steady-State Temperature Profile with Joule Heating

Discussion begins by considering the First Law of Thermodynamics including the Joule-Heating term (shown in bold and will be represented by the symbol, H for the remainder of the paper) for steady-state one-dimensional conduction. Here, we assume that the thermal conductivity K is constant.

$$\frac{d}{dx}K(x)\frac{d\theta}{dx} + \mathbf{aJE} = \rho C\frac{d\theta}{dt} = 0 \tag{1}$$

Integrating eq.1 twice with bounds from 0 to x and solving for the unknown constants C_1 and C_2 where $\theta(x=0) = \theta_0$:

$$\theta(x) = \theta_0 - \int_0^x \frac{Hx}{K} + \frac{C_1}{K}dx + C_2 \tag{2}$$

$$\theta(x=L) = \theta_L - \theta_0 = \frac{-HL^2}{2K} + \frac{C_1 L}{K} \tag{3}$$

The generalized steady-state equation for one-dimensional conduction is therefore:

$$\theta(x) = \theta_0 - \frac{Hx^2}{2K} + \frac{1}{L}(\theta_L - \theta_0)x + \frac{HLx}{2K} \tag{4}$$

Now, we will apply the various initial conditions to the rod using the single constant boundary condition for the fixed-temperature at one end:

$$\theta(x=0, t=0) = \theta_0 \tag{5}$$

Part A. Identical (Dirichlet) Condition

$$\theta(x,t) = \theta_0 - \frac{Hx^2}{2K_0} + \frac{HLx}{2K_0} \tag{6}$$

Part B. Non-matching (Dirichlet) Condition

$$\theta(x,t) = \theta_0 - \frac{Hx^2}{2K_0} + \frac{(\theta_L + 100 - \theta_0)x}{L} + \frac{HLx}{2K_0} \tag{7}$$

Part C. Fixed-Derivative, Fixed Flux (Neumann) Condition The following is evaluated by taking the partial derivative of eq.3 w.r.t x and setting it equal to zero.

$$K\nabla\theta(L)_x = 0 \tag{8}$$

$$\theta(x,t) = \theta_0 - \frac{Hx^2}{2K_0} + \frac{HLx}{2K_0} \tag{9}$$

Part D. Fixed-Flux, Constant Cooling (Neumann) The following is evaluated by taking the partial derivative of eq.3 w.r.t x and setting it equal to the constant:

$$K\nabla\theta(L)_x = -1 \times 10^7 \tag{10}$$

$$\theta(x,t) = \theta_0 - \frac{Hx^2}{2K_0} + \frac{(-10^7 + HL)x}{K} \tag{11}$$

Where the final temperature has units $[Kelvin]$. $H = aJ\epsilon$ has units of $[W/m^3]$ where J is the current density $[A/m^2]$, ϵ is the imposed electric field magnitude $[V/m]$. The isotropic thermal conductivity, K_0 has units of $[W/m * K]$.

Results and Discussion

Process Simulation

Using the finite difference method, one-dimensional transient temperature is solved numerically using different mesh resolution cases, $N = [3, 11, 21]$ to discretize the spatial domain. The plots for A-D are shown for each mesh resolution and the results are discussed.

For case A and B, the temperature along the rod approaches the analytical solution, which reaches a maximum around 440K and 475K, respectively. For case C however, the final temperature at t=tf plateaus at 560K at the center of the rod, whereas the analytical solution levels off more towards the end of the rod and reaches upwards of 850K. A similar scenario occurs in case D, where the analytical temperature solution reaches a maximum larger and further down the length of the rod than from the finite difference simulation. This difference is due to the difference between evaluating a solution using the finite difference method and solving analytically. In the finite difference method for N=3, the temperature gradient is evaluated between nodes (for example, $T(N = 2) = T(N = 3)$ in part C) whereas the analytical temperature gradient is between two "points" on a continuous function. Increasing the number of nodes in the simulation increases the resolution of the temperature profile, which increases accuracy. Table 1 below shows the maximum temperatures from the finite difference method evaluated at the final time.

Case	Maximum Temperature [K]
A	439.9
B	489.7
C	561.7
D	445.4

Table 1: Maximum Temperature at final time for cases A-D

Process Optimization

To optimize the process discussed above, a genetic algorithm is used to find t he o ptimal system parameters for the Joule Heating process that matches the desired results: $\theta_{max} = 460$, the highest temperature at the last time step and $\theta_{last} = 495$, the last value of the temperature array at the

last time step. The cost function that we seek to minimize is shown in eq.12 which will attempt to merge the maximum desired temperature and the desired temperature at the final length. The genetic algorithm will stop at a pre-defined tolerance or once the maximum number of generations is met.

$$\Pi(\lambda) = \omega_1 \frac{(|\theta_{max,tf} - \theta_{max,des}|)^3}{\theta_{max,des}} + \omega_2 \frac{(|\theta_{L,tf} - \theta_{L,des}|)^3}{\theta_{L,des}} \tag{12}$$

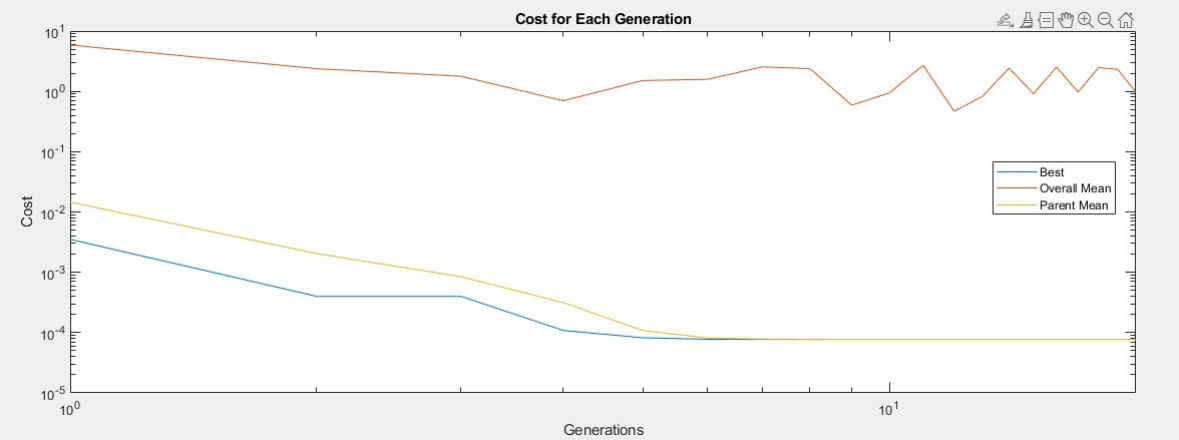


Figure 1: Cost function for 20 generations

Comparing my results with another students, shows that our cost function converges occasionally but there are attempts when the best and overall never converge after 20 generations. In one case, adjusting the system tolerance was found to decrease the computational run time and the Best and Average converged in under 9 generations. Another classmate assigned all variable associated with the process simulation as global variables which was said to decrease run time as well.

There is variation between the top system performers. The best performer was design 3, shown in Table 1. Comparing design 3 with the second lowest cost function, design 2, it is evident that the optimal thermal conductivity value falls in the range of 486 while the optimal Neumann boundary condition falls near $-1.4243 * 10^6$. From the other designs, Neumann boundary conditions above $-1.4243 * 10^6$ tend to increase the cost function but higher thermal conductivity does not necessarily indicate that the rod will reach the maximum optimized temperature, based on the 2nd design. The temperature profile for the optimized system parameters is shown in figure 2. Note that the optimized curve actually reaches a much higher temperature than before at a maximum of 529K compared to 516K for the analytical solution which shows a 2.3 % increase in peak temperature. The analytical and optimized temperatures at the end of the rod is 469K and 498K, respectively, yielding a 6.2 % increase. The temperature profile for the optimized system parameters also has a much more parabolic shape, and reaches the maximum temperature further down the length of the rod.

Conclusion

In this report, a steady-state multiphysical heat model with many system parameters is simulated and optimized to maximize temperature along the length of a one-dimensional rod using the finite difference method. The results are compared to the analytical solution at different boundary conditions. The process is optimized using a genetic algorithm to identify the optimal parameters to maximize temperature. Future analysis could expand the one-dimensional conduction problem to two-dimensions, considering transient conditions, and other boundary conditions which has applications to a broad range of heat problems.

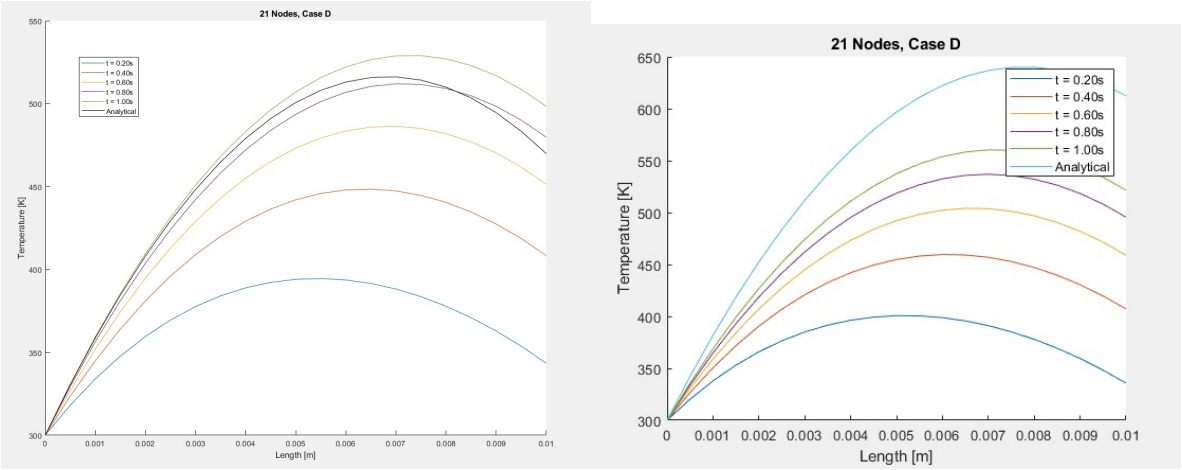
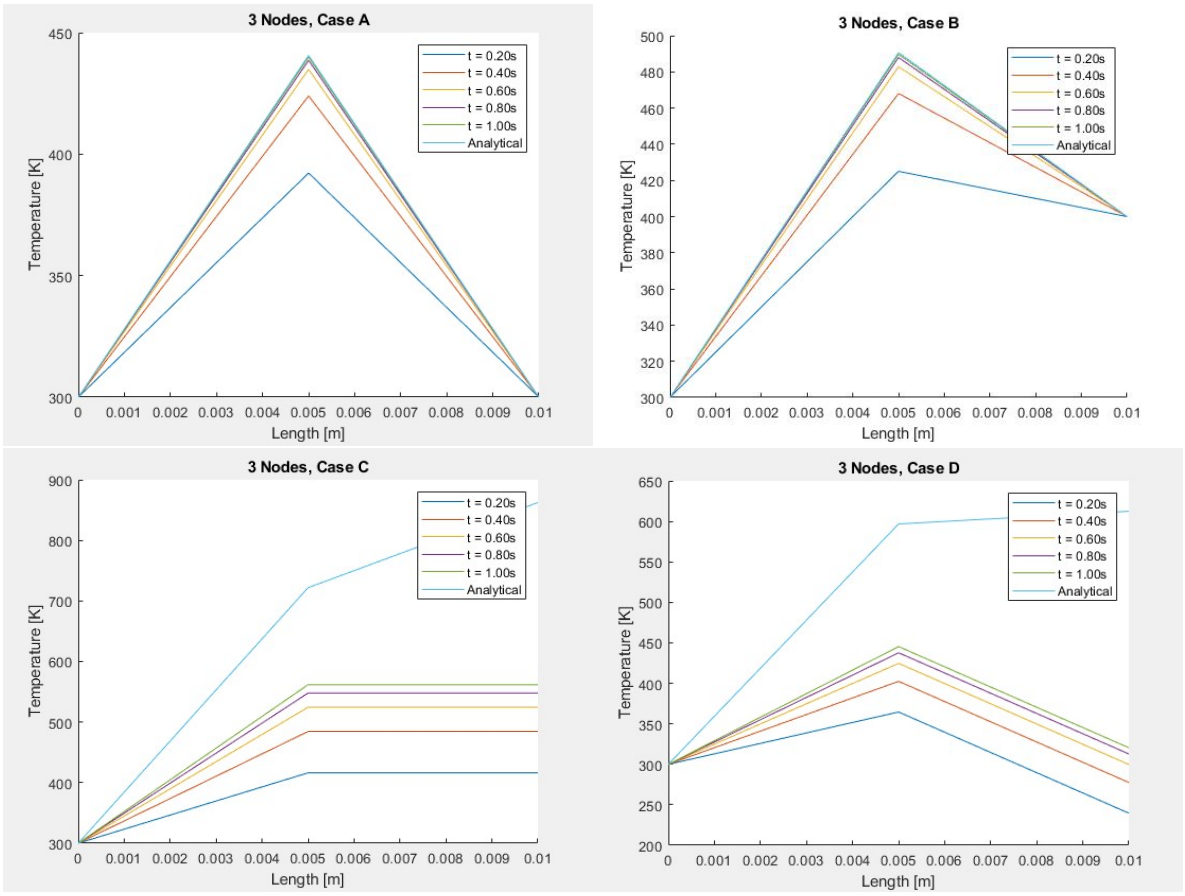
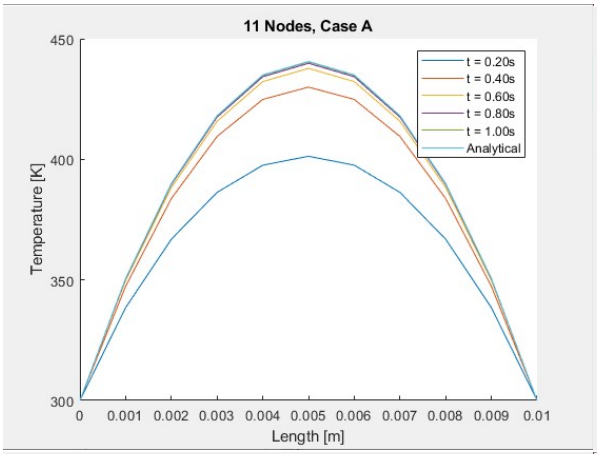


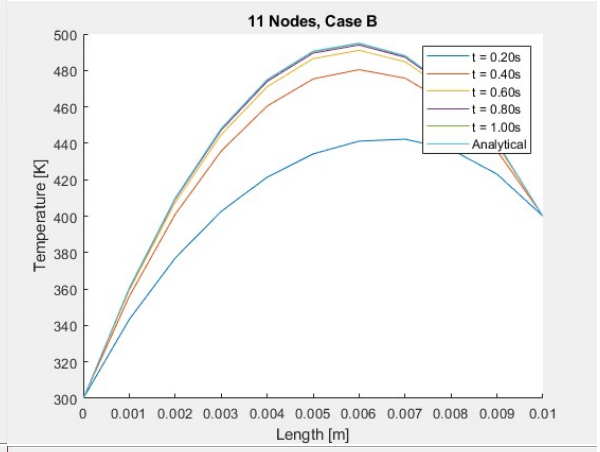
Figure 2: Optimized system parameters (left) and unoptimized (right) for the Neumann Boundary Condition

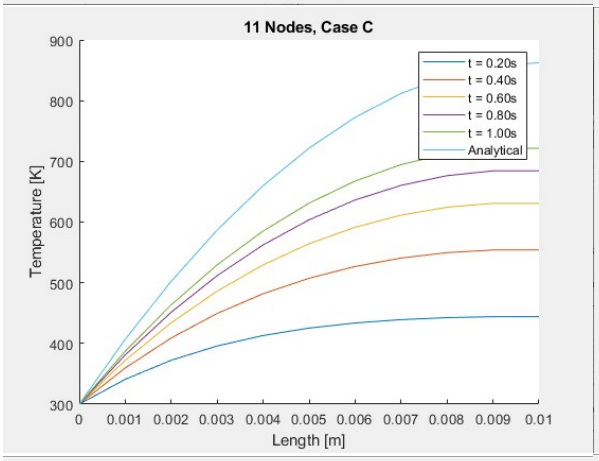
Design	Λ_1	Λ_2	Π
1	475.0650	-1.6355*10^7	2.64*10^{-}5
2	576.5245	-1.2143*10^7	7.2261^{-}6
3	486.3164	-1.4243*10^7	3.0055*10^{-}8
4	571.6130	-1.2276*10^7	1.2153*10^{-}5

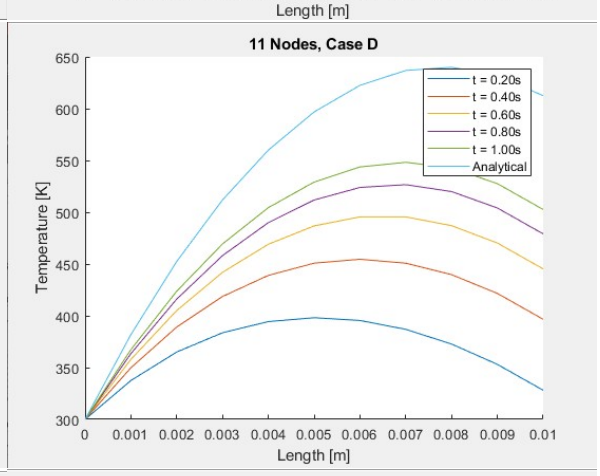
Table 2: Maximum Temperature at final time in Optimized Process

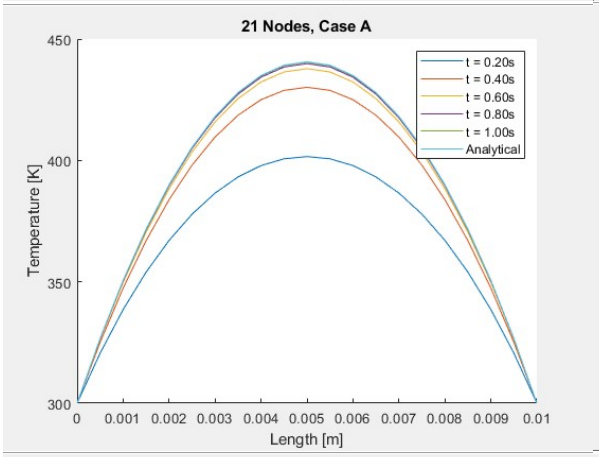

3 Nodes, Case A
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical
3 Nodes, Case B
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical
3 Nodes, Case C
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical
3 Nodes, Case D
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical

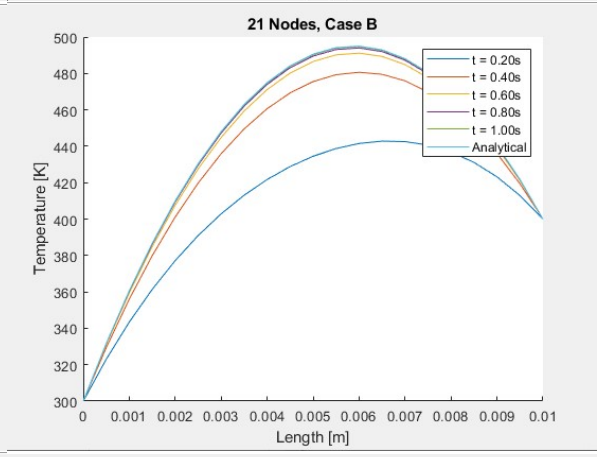

11 Nodes, Case A
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical

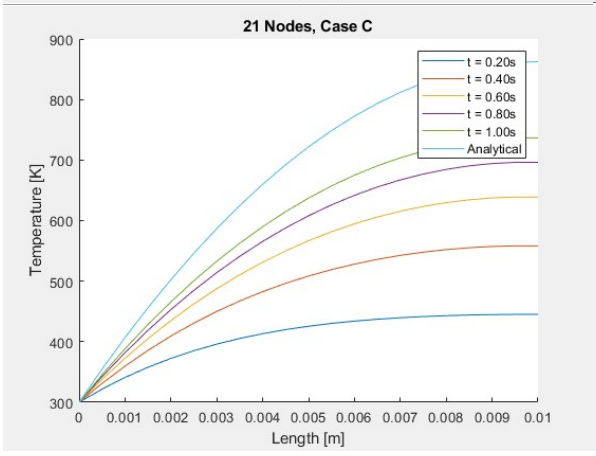

11 Nodes, Case B
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical

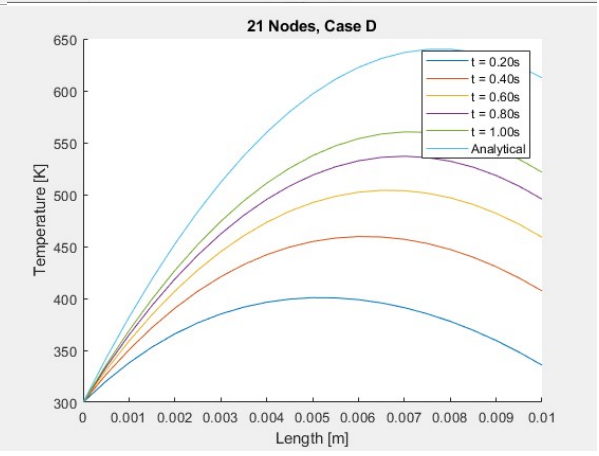

11 Nodes, Case C
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical


11 Nodes, Case D
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical


21 Nodes, Case A
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical


21 Nodes, Case B
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical


21 Nodes, Case C
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical


21 Nodes, Case D
Temperature [K]
Length [m]
t = 0.20s
t = 0.40s
t = 0.60s
t = 0.80s
t = 1.00s
Analytical