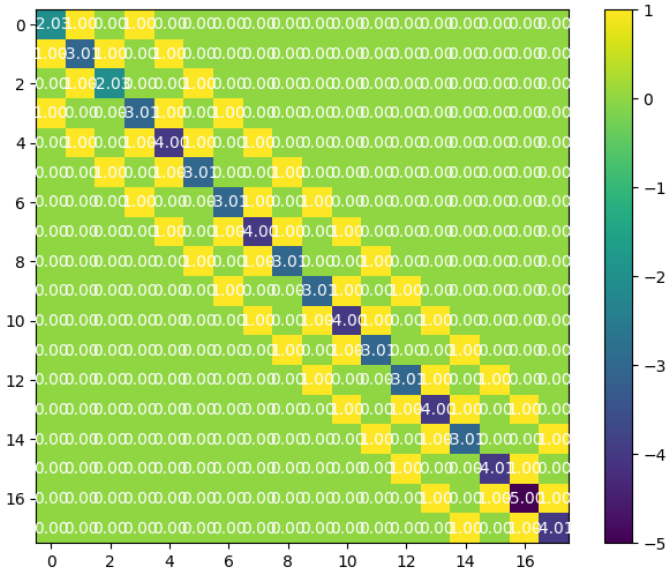

Figure 3.4: Example 3.1.

Illustration of the coefficient A matrix. The value of the coefficients are labeled.

Source: Author's creation.

estimated as $h(T - T_\infty) = 100 \times 70 = 7000 W/m^2$. Therefore, the left and right boundaries are almost exposed to the same boundary conditions. Interested readers can increase the radiative flux to be comparable to the convective heat flux or larger, and observe the temperature distribution.

3.2.3. Treatment of time

So far we have dealt with discretization of steady-state problems. For transient problems, an additional term that involves time derivative is present in the governing equations, for example,

$$\frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k\frac{\partial T}{\partial y}) = \nabla \cdot (k\nabla T) = \rho c \frac{\partial T}{\partial t} \tag{3.34}$$

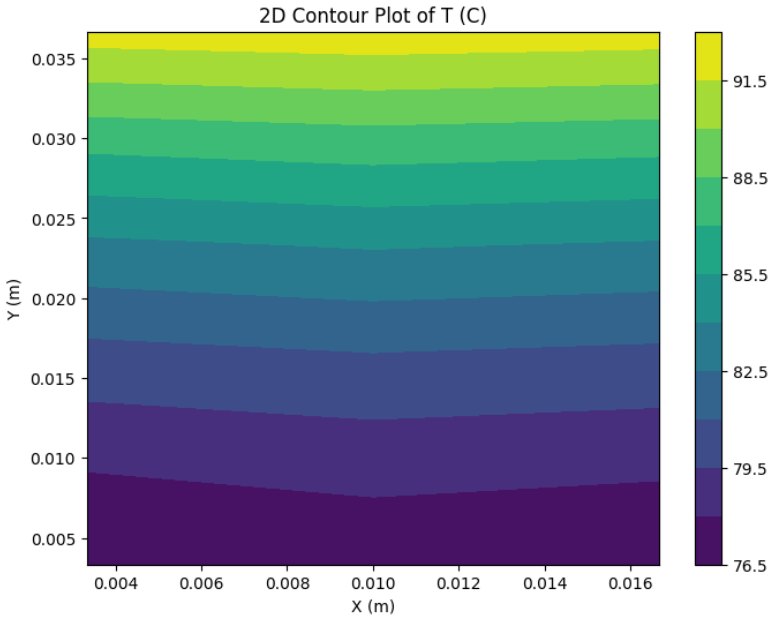

Figure 3.5: Example 3.1. Continued.

Iso-contour of temperature obtained using $N_x = 3$ and $N_y = 6$.

Source: Author's creation.

To discretize Equation 3.34, we need to integrate the entire equation over the user-specified time interval Δt, in addition to integrating the entire equation over the particular control volume $\mathcal{V}$, as shown below.

$$\int_t^{t+\Delta t} (\iiint_{\mathcal{V}} \nabla \cdot (k \nabla T) dV) dt = \iiint_{\mathcal{V}} (\int_t^{t+\Delta t} \rho c \frac{\partial T}{\partial t} dt) dV \tag{3.35}$$

Note that both sides integrate over the volume $\mathcal{V}$ and time interval Δt, albeit in different orders to simplify the computation. For the right-hand side, after integrating the time derivative over time, we simply obtain

$$\iiint_{\mathcal{V}} \rho c (T(t + \Delta t) - T(t)) dV = \rho c (T(t + \Delta t) - T(t)) \mathcal{V}$$

leveraging the assumption that within a computational cell states and properties are constant. To differentiate time and location, we often use superscripts to indicate time and subscripts to indicate spatial locations. For example, $T(t+\Delta t)$ for the cell with spatial identifier i, j is conventionally denoted as $T_{i,j}^{n+1}$, where the superscript $(n+1)$ indicates that it is the $t+\Delta t$ time step (n is used to indicate the current time step at t).

For the left-hand side, we can treat the volumetric integration first, using the discretization scheme discussed in the previous section. For an interior point in the 2D conduction Example 3.1, the left-hand side is reduced to

$$\int_t^{t+\Delta t}(k\frac{\Delta y}{\Delta x}T(i+1,j)+k\frac{\Delta y}{\Delta x}T(i-1,j)+k\frac{\Delta x}{\Delta y}T(i,j+1)+k\frac{\Delta x}{\Delta y}T(i,j-1)-2k(\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y})T(i,j))dt \tag{3.36}$$

The question is then how to approximate the integration $\int_t^{t+\Delta t} T$ for temperature or any other time dependent quantities in Equation 3.36. Conventionally, a linear blending of the quantity from the current time step t and future time step $t+\Delta t$ is a sufficient approximation, which can be expressed mathematically as

$$\int_t^{t+\Delta t} T dt = (\theta T^n + (1-\theta)T^{n+1})\Delta t \tag{3.37}$$

θ can be considered as a blending factor, and it conventionally takes the value of 1 (called Explicit scheme), 0 (Implicit Scheme) and 0.5 (Crank Nicholson Scheme). For the explicit scheme with $\theta = 1$, Equation 3.36 becomes

$$(k\frac{\Delta y}{\Delta x}T^n(i+1,j)+k\frac{\Delta y}{\Delta x}T^n(i-1,j)+k\frac{\Delta x}{\Delta y}T^n(i,j+1)+k\frac{\Delta x}{\Delta y}T^n(i,j-1)-2k(\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y})T^n(i,j))\Delta t \tag{3.38}$$

Connecting it with the left-hand side of equation, for the interior cell (i,j) in

Example 3.1, we obtain

$$\rho c(T^{n+1}(i,j) - T^n(i,j))\Delta x\Delta y = (k\frac{\Delta y}{\Delta x}T^n(i+1,j) + k\frac{\Delta y}{\Delta x}T^n(i-1,j) + k\frac{\Delta x}{\Delta y}T^n(i,j+1) + k\frac{\Delta x}{\Delta y}T^n(i,j-1) - 2k(\frac{\Delta y}{\Delta x} + \frac{\Delta x}{\Delta y})T^n(i,j))\Delta t \tag{3.39}$$

After dividing both sides of the equation by the constants $\rho c\Delta x\Delta y$ and rearranging it, we obtain

$$T^{n+1}(i,j) = (\frac{\alpha\Delta t}{(\Delta x)^2}T^n(i+1,j) + \frac{\alpha\Delta t}{(\Delta x)^2}T^n(i-1,j) + \frac{\alpha\Delta t}{(\Delta y)^2}T^n(i,j+1) + \frac{\alpha\Delta t}{(\Delta y)^2}T^n(i,j-1) + (1 - \frac{2\alpha\Delta t}{(\Delta x)^2} - \frac{2\alpha\Delta t}{(\Delta y)^2})T^n(i,j)) \tag{3.40}$$

For all the transient problems, initial conditions of the solution quantities, in this case temperature T^0, need to be specified. Given the knowledge of that initial state, temperature at location (i,j) at the $t + \Delta t$ time step $T^{n+1}(i,j)$ can be determined directly from Equation 3.40, starting from $n = 0$.

If θ is taken to be 0 or 0.5, Equation 3.39 involves T^n and T^{n+1} for any location. Therefore, a similar A coefficient matrix as in Example 3.1 needs to be constructed and inverted for each time step, and a linear equation solver is required to obtain temperature within each time step.

3.2.4. Accuracy versus stability

For the transient problems described above, it seems that the explicit numerical scheme does not involve inversion of a large sparse matrix and is straightforward to implement. So what is the catch? The answer to this question involves two important concepts in numerical simulations: accuracy versus stability. The accuracy of a numerical simulation of the physical problem depends on many factors: accuracy of the physical models involved in the governing equations, numerical schemes (e.g., first order or second order or higher), the size of the mesh (e.g., Δx and Δy in Example 3.1), as well as the computer used to solve the problem (e.g., single precision or double precision). Stability, on the other hand, is mostly related to mathematical properties of coefficient A matrix. In particular, for explicit method, one requirement to obtain solutions without getting astronomically large/small/unphysical numbers, is to have the time step Δt smaller than $\min(\frac{(\Delta x)^2}{2\alpha}, \frac{(\Delta y)^2}{2\alpha})$. This stability criterion effectively states that the time step should be smaller than the time required to diffuse heat from one end of the cell to the other end. Larger time step contradicts the physical process of heat diffusion and can invalidate the numerical solution. This constraint can be rather limiting when mesh sizes Δx and Δy are small or when the thermal diffusivity α is large. Therefore, people may prefer to use the implicit method, which is unconditionally stable to solve the same problem. However, one needs to keep in the mind the accuracy requirement and not to choose overly large time step Δt to compromise the accuracy.

3.2.5. Virtual Laboratory: 1D transient conduction in a semi-infinite wall

Transient conduction in a semi-infinite wall is a common prototype problem that can be encountered in engineering, for example, when heat transfer along a large thin wall with small thermal conductivity is of interest. The governing equation for this type of problem can be simplified to be

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}(k \frac{\partial T}{\partial x}) \tag{3.41}$$

Suppose that the left-hand side of the wall is exposed to convective fluids at T_∞ with a convective heat transfer coefficient of h. The right-hand side of wall is adiabatic, or assuming the same temperature as the ambient fluids with a very long domain (consistent with the semi-infinite assumption). The boundary conditions can be expressed mathematically as

$$T(x, 0) = T_i, \qquad x > 0 \tag{3.42}$$

$$T(0, t) = T_s, \qquad t > 0 \tag{3.43}$$

$$T(x \to \infty, t) = T_i, \qquad t > 0 \tag{3.44}$$

where T_i is the initial temperature of the solid and T_s is the new surface temperature on the left boundary starting from time 0.

Analytical solution. Analytical solution, as described by Sec. 3.1 exists, and a brief derivation is provided below.

Define the dimensionless temperature θ as

$$\theta(x,t) = \frac{T(x,t) - T_s}{T_i - T_s} \tag{3.45}$$

Then the boundary and initial conditions become:

$$\theta(x,0) = 1 \tag{3.46}$$

$$\theta(0,t) = 0 \tag{3.47}$$

$$\theta(x \to \infty, t) = 1 \tag{3.48}$$

and the governing equation becomes

$$\frac{\partial \theta}{\partial t} = \alpha \frac{\partial^2 \theta}{\partial x^2} \tag{3.49}$$

where $\alpha = k/\rho c$. A similarity solution can be obtained by introducing the similarity variable η to simplify the solution procedure,

$$\eta = \frac{x}{2\sqrt{\alpha t}}, \qquad \theta(x,t) = f(\eta) \tag{3.50}$$

Computing derivatives using the chain rule, we get

$$\frac{\partial \theta}{\partial t} = f'(\eta) \cdot \frac{\partial \eta}{\partial t} = f'(\eta) \cdot \left(-\frac{x}{4\sqrt{\alpha} t^{3/2}} \right) = -\frac{\eta}{2t} f'(\eta) \tag{3.51}$$

$$\frac{\partial^2 \theta}{\partial x^2} = f''(\eta) \cdot \left(\frac{1}{2\sqrt{\alpha t}} \right)^2 = \frac{f''(\eta)}{4\alpha t} \tag{3.52}$$

Substituting the derivatives into the partial differential equation, we obtain

$$-\frac{\eta}{2t} f'(\eta) = \alpha \cdot \frac{f''(\eta)}{4\alpha t} \quad \Rightarrow \quad -2\eta f'(\eta) = f''(\eta) \tag{3.53}$$

The resulting ODE is then

$$f''(\eta) + 2\eta f'(\eta) = 0 \tag{3.54}$$

Let $g(\eta) = f'(\eta)$,

$$g'(\eta) + 2\eta g(\eta) = 0 \tag{3.55}$$

which is a linear ordinary differential equation(ODE). Multiplying by the integrating factor $\mu(\eta) = e^{\eta^2}$, the equation is transformed into

$$\frac{d}{d\eta}\left[g(\eta)e^{\eta^2}\right] = 0 \quad \Rightarrow \quad g(\eta) = C_1 e^{-\eta^2} \tag{3.56}$$

Integrating Equation 3.56, we obtain

$$f(\eta) = C_1 \int e^{-\eta^2} d\eta + C_2 = C_1 \cdot \frac{\sqrt{\pi}}{2} \cdot \mathrm{erf}(\eta) + C_2 \tag{3.57}$$

Here, $\mathrm{erf}(\eta) \equiv \frac{2}{\sqrt{\pi}} \int_0^\eta e^{-p^2} dp$ is the error function that can be tabulated or directly called as built-in functions from many programming packages. C_1 and C_2 are constants from the integration, which can be found from applying the boundary condition,

$$f(0) = 0 \Rightarrow C_2 = 0 \tag{3.58}$$

$$f(\infty) = 1 \Rightarrow C_1 = \frac{2}{\sqrt{\pi}} \tag{3.59}$$

The final solution is

$$\theta(x,t) = \mathrm{erf}\left(\frac{x}{2\sqrt{\alpha t}}\right) \tag{3.60}$$

or in dimensional form,

$$T(x,t) = T_s + (T_i - T_s) \cdot \text{erf}\left(\frac{x}{2\sqrt{\alpha t}}\right) \tag{3.61}$$

Numerical solutions. Numerically, using the explicit approach for advancing in time, Equation 3.39 can be used. In its one-dimensional version, the discretized equation for any interior cell i is

$$\rho c(T^{n+1}(i) - T^n(i))\Delta x = (\frac{k}{\Delta x}T^n(i+1) + \frac{k}{\Delta x}T^n(i-1) - 2\frac{k}{\Delta x}T^n(i))\Delta t \tag{3.62}$$

After re-arranging, we obtain

$$T^{n+1}(i) = T^n(i) + \frac{k}{\rho c}\frac{\Delta t}{\Delta x^2}(T^n(i+1) + T^n(i-1) - 2T^n(i)) \tag{3.63}$$

$k/\rho c$ is the thermal diffusivity α, and subsequently $\frac{\alpha \Delta t}{\Delta x^2}$ is the local Fourier number **Fo**.

Once an initial condition ($n = 0$) is given, the temperature at the next time steps ($n = 1, 2, \cdots$) can be obtained. Two boundary conditions are required to advance temperature since the original partial differential equation is second-order in space. For the left boundary at $x = 0$, the discretized form is (assuming the cell ID for the first cell from the left is $i = 0$)

$$\rho c(T^{n+1}(0) - T^n(0))\Delta x = (\frac{k}{\Delta x}T^n(1) + \frac{2k}{\Delta x}T_s - 3\frac{k}{\Delta x}T^n(0))\Delta t \tag{3.64}$$

For the right-hand side boundary (assuming the cell ID is $i = N$), we obtain

$$\rho c(T^{n+1}(N-1) - T^n(N-1))\Delta x = (\frac{k}{\Delta x}T^n(N-2) + \frac{2k}{\Delta x}T_i - 3\frac{k}{\Delta x}T^n(N-1))\Delta t \tag{3.65}$$

The Jupyter Notebook provided in the Supplementary Materials details the Python program that solves the problem and compares it with the analytical solution. Since thermal conductivity k always appears with density ρ and specific heat c in the form of thermal diffusivity, only thermal diffusivity α is needed in the program, which is assigned to be $\alpha = 10^{-5}$ m^2/s. A domain of $L = 0.3$ m is selected to represent the semi-infinite domain. The domain is discretized into 150 subdomains, with a cell width $\Delta x = 0.002$ m. Based on the stability criterion in Sec. 3.2.3, the maximum time step is $\Delta t = \frac{\Delta x^2}{2\alpha} = 0.2$ s.

Figure 3.6 shows the temperature evolution at 20, 100, 200 and 1000 s, compared to the analytical solutions, using $\Delta t = 0.1$ s. Great agreement is achieved between the numerical and analytical solutions, verifying that the numerical procedure is correct. It is noteworthy that the temperature beyond 0.2 m is unaffected by the sudden increase of temperature (or appears to have no knowledge of the change) on the left boundary by 200 seconds. This is because diffusion occurs at finite rates, and at 200 seconds with a diffusivity of 10^{-5}, the length scale for diffusion can be estimated using $2\sqrt{\alpha t}$ as indicated by the similarity variable, which gives a value of 0.04 m. This estimated distance is on the same order of magnitude with the observation from the results here. Further advancement in time makes the temperature significantly different from the initial temperature at the right boundary when using $L = 0.3$. In other words, $L = 0.3$ m violates the assumptions of semi-infinite domain in the analytical solution. As a result, a deviation between the numerical solution is observed from the analytical solution at $t = 1000$ s. Similarly, if a very early time instant is considered, a smaller L could qualify as a semi-infinite domain. In other words, the validity of a semi-infinite domain depends on the time of interest and the thermal diffusivity of the solid material.

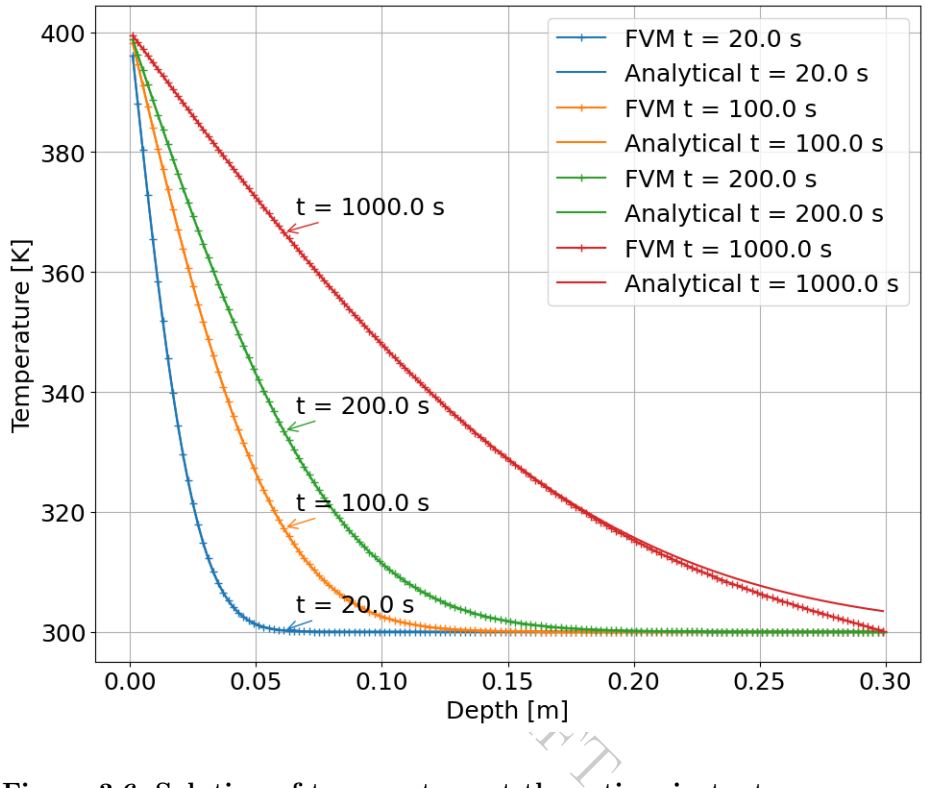

Figure 3.6: Solution of temperature at three time instants

Source: Author's creation.

Now we use the Jupyter notebook to explore some concepts introduced in this chapter. Figure 3.7(a) shows the same solutions after normalization, using the variables θ and η. All the results at different physical time instants collapse with one another, which is the meaning or signature of a self-similar solution. The solution at $t = 1000$ s slightly deviates from the rest of the profiles, since it violates the semi-infinite domain assumption, as discussed above. Figure 3.7(b) are obtained by setting the time step Δt to be twice the maximum value that is determined by the stability criterion. Fluctuations of the solution are the most prominent feature, indicating instability in the solution. Moreover, the magnitude of temperature overshoots and becomes unphysical. To avoid such

instability, the time steps should be smaller than the maximum allowable value, as demonstrated in Figure 3.6.

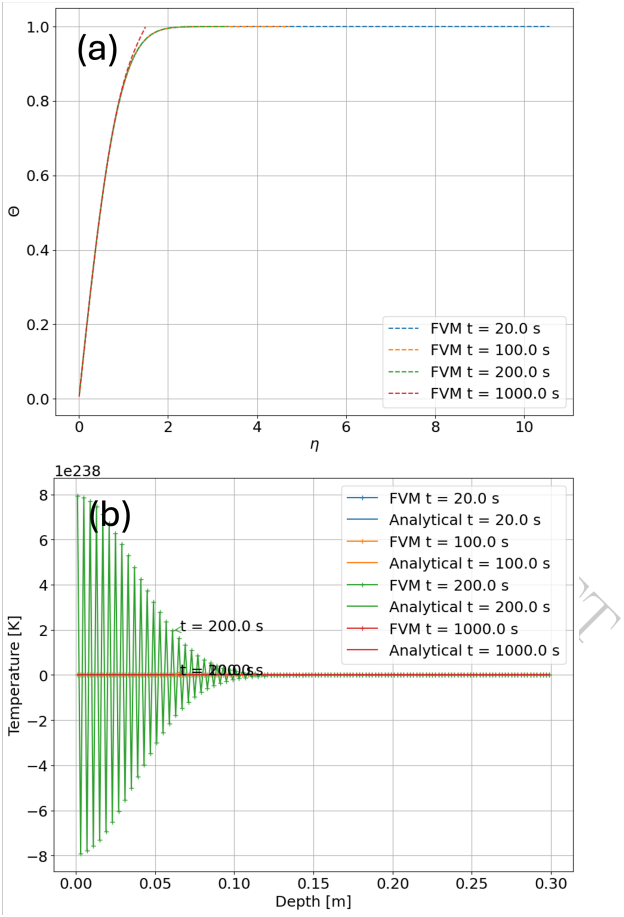

Figure 3.7: Parametric analysis of the results.

(a)Non-dimensional solution of temperature at four time instants in the self-similarity coordinate. (b) Diverged solution when $\Delta t = 0.4$ s is used. *Source:* Author's creation.

Comments. The example shows a particular solution method for partial differential equation, where self-similarity exists. The solution procedure can be derived with some mathematical intuition so that a simple ordinary differential

equation is formed. A more general approach to solve the second-order ordinary differential equation will be introduced in Chapter 10. A similar solution procedure is also employed in deriving solution for two-dimensional laminar boundary layer problems; interested readers can find more details in [Bergman and Lavine, 2017]. The numerical solutions, on the other hand, require less physical intuition, and can be obtained by the numerical procedure introduced in this chapter. The accuracy and stability of the solution are demonstrated through comparison with analytical solutions. It should be noted that the time step constraint for stability is only associated with the explicit method. For implicit method (i.e., by setting $\theta = 0$ in Equation 3.37), the solution is unconditionally stable; the choice of Δt is solely constrained by accuracy consideration because larger time step also introduces larger errors.

3.3. Summary

This chapter covers a wide variety of topics, including analytical solutions to multi-dimensional problems, numerical methods for solving these problems, and the lumped capacitance method to provide time or temperature estimate for an object without any spatial temperature variation. These topics are deeply connected and form the foundation for solving an actual heat transfer problem, either through back-of-an-envelope calculation or using high-performance computing facilities involving hundreds of computers. It is important to remember that we are solving a physical problem, and the solution needs to make physical sense. For example, a numerical value of 100,000 Kelvin is mathematically possible as a solution to the heat equation, however, the value is physically almost impossible for heat transfer problems on earth. With more

and more computer-aided computations involved in everyday engineering activities, it is particularly important to verify the numbers that come out of a computer. The analytical solutions introduced in Sec. 3.1 are useful tools to conduct such verification and to debug a software or computer programs. The lumped capacitance method in Sec. 3.1.1, on the other hand, provides an estimation of time required to reach certain temperature or thermal equilibrium. With time scale estimated using a length scale and thermal diffusivity, L_c^2/α, one can understand how long they need to run their computer programs and how large the time step needs to be chosen.

Bibliography

Guillermo Araya and Gustavo Gutierrez. Analytical solution for a transient, three-dimensional temperature distribution due to a moving laser beam. *International Journal of Heat and Mass Transfer*, 49(21):4124–4131, 2006.

Theodore L. Bergman and Adrienne S. Lavine. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2017.

D. Poulikakos. *Conduction heat transfer)*. Prentice Hall, Englewood Cliffs, New Jersey 06732, 1994.