Approximating a plane wall as a semi-infinite solid

In class, you have solved for temperature profiles in a semi-infinite solid exposed to different boundary conditions. Sometimes, it is also appropriate to treat a transient, 1-D, plane wall with no energy generation as a semi-infinite solid as long as $Fo_L = \alpha t/L^2 < 0.2$. We will be testing the validity of this. Your task will be to numerically simulate the evolution of the temperature profiles in a plane wall subject to different boundary conditions and compare those profiles with those obtained analytically using the semi-infinite solid assumption, and with the infinite series solution.

For this problem, consider a plane wall of thickness $2L$ = 2 [m], thermal conductivity k = 21.2 [W/m-K], density ρ = 8885 [kg/m^3] and specific heat capacity c = 390 [J/kg-K]. Initially, it was at T_i = 300 [K]. At $t > 0$, it is exposed to three different boundary conditions:

Case 1	Constant surface temperature of 500 [K] at both the boundaries
Case 2	Constant heat flux of 3000 [W/m^2] at both the boundaries **coming into the wall**
Case 3	Convection at both the boundaries to a fluid with h = 200 [W/m^{2}K] and T_{fluid} = 500 [K]

In this project, you will be solving for the temperature profiles numerically and comparing these profiles with the infinite series solutions (which are exact and are provided to you). You will also be comparing the numerically solved temperature profiles with the temperature profiles for semi-infinite solids.

Project Deliverable 1:

Due: 10 pm, Thursday, July 13, 2023

Task A: Treating the plane wall as a semi-infinite solid, compute the temperatures within the solid for the three different boundary conditions. Then, non-dimensionalize the temperatures as

Case 1	$\theta(x,t) = (T(x,t) - T_i)/(T_s - T_i)$ where T_i and T_s are the initial temperature and the constant surface temperature respectively.
Case 2	$\theta(x,t) = (T(x,t) - T_i)/(q_s" \cdot L/2k)$ where T_i, q_s, L and k are the initial temperature, constant surface heat flux, wall thickness and wall thermal conductivity respectively.
Case 3	$\theta(x,t) = (T(x,t) - T_i)/(T_{fluid} - T_i)$

	where T_i and T_{fluid} are the initial temperature and the fluid temperature respectively.

Deliverable(s) for task A: On three separate plots (one for each boundary condition), plot the non-dimensionalized temperature profiles in the half-thickness of the plane wall as dashed lines of different colors, i.e., plot $\theta(x, t)$ as a function of *x* for $0 < x < L = 1.0$ [m] for Fo_L = 0.01, 0.1, 0.2, 0.5, and 1.0, where *L* is the half-thickness of the plane wall, and the boundary condition is applied at *x* = 0.

Note: Fo_L is a non-dimensional time scale defined as

$$Fo_L = \alpha t/L^2$$

where t is the time in seconds. This is different to the Fo defined in chapter 5 of your textbook which is based on the time step, Δt and space step Δx as

$$Fo = \alpha \cdot \Delta t/(\Delta x)^2$$

Task B: Next, you will compare the non-dimensional temperature profiles from the semi-infinite solid with the non-dimensional temperature profiles from the infinite series, which are the exact solution. On the Canvas page, you will find three additional MATLAB scripts titled '*ExactSolution_CST.m*', '*ExactSolution_CHF.m*', and '*ExactSolution_Convection.m*' which calculate and plot the temperature profiles for 0 < Fo_L < 1 using a finite number of terms of the infinite series solution. The scripts also generate comma-separated-values (CSV) files with the temperatures stored in them. These files may be viewed using Excel. The temperature values at each Fo_L is organized in rows, with each row containing the temperature at a user-defined number of linearly spaced points within the plane wall.

Update (10 July, 2023): In the csv files for the exact solutions, the first column holds the value of Fo_L for the temperature profile in that row.

Deliverable(s) for task B: On the three plots from task A, also plot the non-dimensional temperature profiles as solid lines of different colors (but same color for the same Fo_L) from the infinite series solutions provided to you in csv files on the Canvas page.

Task C: Now, you will compare the wall heat flux for the constant surface temperature and the convective boundary condition, and the surface temperature for the constant heat flux from the two solutions. The heat flux for the exact solution can be estimated using the temperatures at *x* = 0 and *x* = 0.05 [m] as

$$q_s" = k(T(x = 0) - T(x = 0.05))/\Delta x$$

Deliverable(s) for task C: On two new plots (one for the constant surface temperature and the other, for convective BC), plot the wall heat flux as a function of Fo_L for 0.01 ≤ Fo_L ≤ 1 as dashed lines for the semi-infinite solid approximation. Then, use the infinite series solutions to plot the wall heat flux as a function of Fo_L for $0.01 \le Fo_L \le 1$ as solid lines.

On a third plot (for the constant heat flux), plot the non-dimensional surface temperature as a function of Fo_L for $0.01 < Fo_L < 1$ as a dashed line for the semi-infinite solid approximation. Then, use the infinite series solutions to plot the same as a function of Fo_L for $0.01 < Fo_L < 1$ as a solid line.

Deliverable(s) for task D: Based on the plots from Tasks A and B, and C, comment on the validity of the semi-infinite solid approximation for the plane wall for different Fo_L. When do the curves start diverging from each other? What does that tell you about the validity of the semi-infinite approximation when applied to a plane wall of a finite thickness?

Project Deliverable 2:

Due: 10 pm, Thursday, July 27, 2023

Your next task is to derive the finite-difference form of the heat diffusion equation and the boundary conditions that you will be implementing in your final code.

Deliverables:

Task E.1. Writing the appropriate form of the heat diffusion equation

Start with the most general form of the heat diffusion equation in the relevant coordinate system (Be careful about what coordinate system you use). Then, make, and state, reasonable assumptions to simplify the equation as much as possible. Present your work neatly.

Task E.2. Writing the boundary and initial conditions

State your boundary and initial conditions (in mathematical form) for each of the three cases.

Task F.1. Finite-Difference Form of the Heat Diffusion Equation

Use finite-difference approximations to derive the finite-difference form of the heat diffusion equation to solve it explicitly, and implicitly. For the explicit method, what is your stability criteria (see section 5.10.1 or see the relevant pdf under the Textbook module on Canvas on how to determine the stability criteria)?

Task F.2. Finite-Difference Form of the Boundary Conditions for Each Case

Come up with a finite difference form for the boundary conditions for the implicit and the explicit method for each case (The implicit and explicit methods will have a separate finite difference form of the boundary condition). (For cases 2 and 3, refer to the text around Figure 5.12 or see the relevant pdf under the Textbook module on Canvas. Perform a similar energy balance for your coordinate system). State the stability criteria for the explicit method of cases 2 and 3?

Final Report:

Due: 10 pm, August 3, 2023

For your final report, you have to solve for the temperature profiles for all the three cases using both the implicit and the explicit schemes.

Task G. Implement the codes

If *M* denotes the number of spatial nodes, write your codes for *M* = 6, such that $\Delta x = L/(M - 1)$. Keep the following in mind:

1. It would be best if *M* is a variable that you can change easily because in the Grip Independence Study that will follow, you will see the effect on your solution when you change *M*.
2. You want to solve temperature profiles for $0 \leq Fo_L \leq 1$ where $Fo_L = \alpha t/L^2$ is the non-dimensional time. Therefore, the maximum time for which you want to solve the temperature profiles is $t = L^2/\alpha$.
3. For the explicit scheme, you have multiple stability criteria - one from the heat diffusion equation, and two (or one for the constant surface temperature case) from the boundary conditions. All of them must be satisfied.
4. For the implicit scheme, you do not have a stability criteria. So initially, write code for *nt* = 1001. You will change this value based on the grid independence study in the next step.

Note: To check if your codes are working correctly, compare temperature profiles from the implicit and explicit schemes for Fo_L *= 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, and 1.0. They should roughly match but depending on the value of nt, the implicit scheme may act funny at the boundary with convection and heat flux. You will see in the next step that increasing nt gets rid of this problem.*

Task H. Grid Independence Study

Once you have correctly implemented the code, your next task is to find the appropriate value of *M*. The purpose of the grid independence study is to ensure that your results are independent of the grid size (values of Δr and Δt). The idea behind the grid independence study is to change the value of *M* or *nt* and monitor the change on the solution. The solution is determined to be independent of the grid size once the solution changes less than a certain threshold which depends on how critical the accuracy of the simulation is.

H1. First, do a grid independence study for the explicit scheme. Change the values of *M* to 6, 11, 21, 51, and 101 and monitor the temperature at the following nodes for the different cases:

1. For the constant surface temperature, monitor the node right next to the boundary node that is exposed to the constant boundary condition at $Fo_L = 1$.

2. For the constant heat flux and convection cases, monitor the temperature of the node at the boundary exposed to those conditions at Fo_L = 1.

Note that your value of Δt should be one which satisfies the most critical of the stability criteria, so it will be tied to the value of M. What is the percentage change in the value of the temperatures as M changes from one value to the next? Do this for all the three boundary conditions and **tabulate** your results.

H2. Now, choose a fixed value for M from Task H1 that you feel is reasonable for the application at-hand. Use that value of M in the implicit scheme now but this time, change values of Δt since they are not tied to the value of M for implicit schemes. You may start with values of Δt that is about 10 times the value which satisfies the most critical stability criteria from task **H1**, and then keep halving the time steps as you note the percentage change in the value of the surface temperature. Additionally, this time, use MATLAB functions tic and toc to monitor how long the code takes to run for the different values of Δt. **Tabulate** the results for each case.

Note: The choice to monitor the temperature of a particular node is motivated by the fact that this is where the temperature changes are expected to occur the earliest. For the constant surface temperature case, the surface node should not be considered because the constant surface temperature is imposed on solution, and hence, it will not change.

Note 2: A characteristic of the implicit scheme is that the code may take significantly longer to run depending on the value of M, since the matrix constructed in a M × M matrix, which will take significantly longer times to invert the bigger the matrix is. You could look at how much longer the implicit scheme takes to run compared to the implicit scheme for the same value of M and nt.

Task I. Plots for the final report

You have to compare your results from the finite difference methods and the semi-infinite approximation against the 'exact solutions'. To generate the exact solution, use the MATLAB codes provided on the Canvas page ('*ExactSolution_CST.m*, *ExactSolution_CHF.m*, and *ExactSolution_Convection.m*). In these MATLAB codes, you can change the values of the variables *nx*, and *nt* to match what you have in your finite difference code. When you run the code, it will generate a plot of the temperature profiles at Fo_L = 0.01, 0.1, 0.2, 0.5, and 1.0, and generate a CSV file in the current folder. This CSV file contains the temperature profiles for the half-plane wall (for 0 [m] ≤ x ≤ 1 [m]) at different values of 0 ≤ Fo_L ≤ 1. You may use the matlab function readmatrix to read the temperatures from the CSV file into MATLAB.

I1. For all three cases, show the RMS (root-mean-square error) error in the temperature profiles against 0 ≤ Fo_L ≤ 1 for the semi-infinite solid approximation and the finite difference methods (either implicit or explicit). For instance, at a particular value of Fo_L, take the temperature profile from your finite difference code (a matrix with M elements, T_{FD}) and the exact solution (a matrix, T_{ES}). Then, you can calculate the RMS as

$$RMS_{Fo_L} = \sqrt{\left(\sum_{i=1}^{M} (T_{FD,i} - T_{ES,i})^2\right)/M}$$

where M is the number of spatial nodes. Similarly, you can compute the RMS error for the semi-infinite solid approximation temperature profiles. For each case, plot two RMS error curves on the same graph for $0 \leq Fo_L \leq 1$ - one for the finite difference method, and the other for the semi-infinite solid approximation. The curves should be lines of different colors without markers.

I2. The RMS error quantifies the overall deviation of the finite difference method and semi-infinite solid approximation from the exact solution. But the deviation in the temperature profiles might be more significant in certain locations of the plane wall than others. A location of interest is the surface exposed to the boundary condition. Plot the following:

I2.1. For the constant surface temperature case, plot the relative error in the wall heat flux against $0 \leq Fo_L \leq 1$ for the finite difference method and the semi-infinite solid approximation. The wall heat flux at any time step may be calculated as

$$q_s'' = k(T_0 - T_1)/\Delta x$$

The relative error in the wall heat flux is defined as

$$\text{Relative Error} = (|q_s''_{FD} - q_s''_{ES}|)/q_s''_{ES}$$

Plot two relative error curves on the same graph for $0 \leq Fo_L \leq 1$ - one for the finite difference method, and the other for the semi-infinite solid approximation. The curves should be lines of different colors without markers.

I2.2. For the constant heat flux case, plot the relative error in the surface temperature against $0 \leq Fo_L \leq 1$ for the finite difference method and the semi-infinite solid approximation. The relative error in the surface temperature is defined as

$$\text{Relative Error} = (|T_{M,FD} - T_{M,ES}|)/T_{M,ES}$$

Plot two relative error curves on the same graph for $0 \leq Fo_L \leq 1$ - one for the finite difference method, and the other for the semi-infinite solid approximation. The curves should be lines of different colors without markers.

I2.3. For the convective boundary case, plot both the relative error in the surface temperature and wall heat flux against $0 \leq Fo_L \leq 1$ for the finite difference method and the semi-infinite solid approximation.

Plot two relative error curves for the surface temperature on the same graph for $0 \leq Fo_L \leq 1$ - one for the finite difference method, and the other for the semi-infinite solid approximation.

Then, plot two corresponding relative error curves for the wall heat flux on a separate graph. The curves should be lines of different colors without markers.