Cooling of a sphere in a well-stirred liquid

6EMA08, Multiphase Computational Fluid Dynamics - Q2 (2024)

Group 16

Full Name	Student ID
Adam Jordani Misa	2208512
Artemis Angelopoulou	2192977
Srinivasan Seshadri	1823671

Lecturer: Maike Baltussen

Eindhoven, November 24, 2024

Contents

1 | Introduction

A homogeneous solid sphere of radius R, initially at a uniform temperature T_1, is suddenly immersed at time $t = 0$ in a volume V_f of well-stirred fluid of temperature T_0 in an insulated tank. It is desired to find the thermal diffusivity $\alpha_s = \frac{k_s}{\rho_s C_{p,s}}$ of the solid by observing the change of the fluid temperature T_f with time. In this study, we proceed to formulate the governing equations for the solid phase and fluid phase, establish the necessary initial and boundary conditions, and solve the system to predict the evolution of T_f. This process enables us to derive a relationship that connects the thermal properties of the solid sphere to measurable changes in the fluid temperature.

2 | Mathematical Formulation

Governing equations

The first step of our analysis was to derive the governing equations that describe the heat transfer dynamics in the system. For the solid phase we proceed by formulating the governing equation which match with the heat conduction equation for a sphere and is given by:

$$\frac{\partial T_s}{\partial t} = \alpha_s \left(\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial T_s}{\partial r} \right) \right) \tag{2.1}$$

with thermal diffusivity defined as:

$$\alpha_s = \frac{k_s}{\rho_s C_{p,s}} \tag{2.2}$$

For the fluid phase, we derive the governing equation, which corresponds to the energy balance equation for a well-stirred fluid, and is given by:

$$\rho_f C_{p,f} V_f \frac{dT_f}{dt} = -k_s A \left. \frac{\partial T_s}{\partial r} \right|_{r=R} \tag{2.3}$$

Rearranging:

$$\frac{dT_f}{dt} = -\frac{k_s A}{\rho_f C_{p,f} V_f} \left. \frac{\partial T_s}{\partial r} \right|_{r=R} \tag{2.4}$$

Substituting the surface area of the sphere $A = 4\pi R^2$ and volume V_f:

$$\frac{dT_f}{dt} = -\frac{4\pi R^2 k_s}{\rho_f C_{p,f} V_f} \left. \frac{\partial T_s}{\partial r} \right|_{r=R} \tag{2.5}$$

To fully define the problem and ensure the governing equations are solvable, it is essential to specify the appropriate boundary conditions (BCs). These boundary conditions describe the behavior of the system at the spatial and temporal boundaries, including the initial state of the system and the interactions between the solid sphere and the surrounding fluid.

$$\text{At } t = 0, \quad T_s = T_1 \quad \text{(initial condition)} \tag{2.6}$$

$$\text{At } r = R, \quad T_s = T_0 \tag{2.7}$$

$$\text{At } r = 0, \quad T_s = \text{Finite} \tag{2.8}$$

Dimensionless Numbers

We introduce dimensionless variables to simplify the mathematical formulation and to generalize the problem. The dimensionless solid and fluid temperatures normalize the temperature difference, allowing

the analysis to focus on relative changes rather than absolute values. The dimensionless radial coordinate normalizes the radial distance by the sphere's radius, ensuring that the analysis is scale-independent and valid for spheres of different sizes. Similarly, the dimensionless time incorporates thermal diffusivity and removes reliance on specific time units, emphasizing the relative progression of heat transfer over time. The dimensionless variables used in this study are as follows:

$$\begin{aligned}
\Theta_s(\xi,\tau) &= \frac{T_1 - T_s}{T_1 - T_0}, && \text{dimensionless sphere temperature} \\
\Theta_f(\tau) &= \frac{T_1 - T_f}{T_1 - T_0}, && \text{dimensionless fluid temperature} \\
\xi &= \frac{r}{R}, && \text{dimensionless radial coordinate} \\
\tau &= \frac{\alpha_s t}{R^2}, && \text{dimensionless time}
\end{aligned}$$

Based on the dimensionless variables introduced above, the governing equation for the solid phase can be rewritten in its dimensionless form. It becomes:

$$\frac{\partial T_s}{\partial t} = \alpha_s \left(\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial T_s}{\partial r} \right) \right) \tag{2.9}$$

Substitute dimensionless variables:

$$\frac{\partial \Theta_s}{\partial \tau} = \frac{\partial}{\partial t} \left(\frac{T_1 - T_s}{T_1 - T_0} \right) = \frac{1}{\xi^2} \frac{\partial}{\partial \xi} \left(\xi^2 \frac{\partial \Theta_s}{\partial \xi} \right) \tag{2.10}$$

To simplify:

$$\frac{\partial \Theta_s}{\partial \tau} = \frac{1}{\xi^2} \frac{\partial}{\partial \xi} \left(\xi^2 \frac{\partial \Theta_s}{\partial \xi} \right) \tag{2.11}$$

After transforming the governing equations into their dimensionless form, the next step involves defining the boundary conditions for the dimensionless solid temperature. The boundary conditions for the dimensionless solid temperature are:

$$\text{At } \tau = 0, \quad \Theta_s = 0 \tag{2.12}$$

$$\text{At } \xi = 1, \quad \Theta_s = 1 \tag{2.13}$$

$$\text{At } \xi = 0, \quad \Theta_s \text{ is finite} \tag{2.14}$$

The dimensionless form of the fluid energy balance becomes:

$$\frac{d\Theta_f}{d\tau} = -\frac{3}{B} \left. \frac{\partial \Theta_s}{\partial \xi} \right|_{\xi=1} \tag{2.15}$$

with:

$$B = \frac{\rho_f C_{p,f} V_f}{\rho_s C_{p,s} V_s} \tag{2.16}$$

with initial condition for dimensionless variables of the fluid:

$$\text{At } \tau = 0, \quad \Theta_f = 1 \tag{2.17}$$

Solid	Fluid
$\frac{\partial \Theta_s}{\partial \tau} = \frac{1}{\xi^2}\frac{\partial}{\partial \xi}\left(\xi^2 \frac{\partial \Theta_s}{\partial \xi}\right)$	$\frac{d\Theta_f}{d\tau} = -\frac{3}{B}\frac{\partial \Theta_s}{\partial \xi}\Big\vert_{\xi=1}$
At $\tau = 0$, $\Theta_s = 0$ At $\xi = 1$, $\Theta_s = \Theta_f$ At $\xi = 0$, Θ_s is finite	At $\tau = 0$, $\Theta_f = 1$

Table 2.1: Dimensionless governing equations and boundary conditions for solid and fluid phases.

3 | Discretization of Governing Equations

To numerically solve the governing equations, it is necessary to discretize them. Discretization involves approximating continuous derivatives with finite difference expressions, enabling the transformation of the partial differential equation into a solvable algebraic form.

In this section, the heat diffusion equation for the solid is discretized using finite difference approximations for time, first-order spatial, and second-order spatial derivatives. Discretizing the governing equation for the solid phase:

$$\frac{\partial \Theta}{\partial \tau} = \frac{1}{\xi^2}\frac{\partial}{\partial \xi}\left(\xi^2 \frac{\partial \Theta}{\partial \xi}\right)$$

Solid Temperature Distribution Verification

1.Analytical Solution

The analytical solution provides an exact expression for the temperature distribution in a solid sphere under transient heat conduction. Using the method of separation of variables, the solution is expressed as a Fourier series that accounts for the radial position and time. This serves as a benchmark to validate numerical methods by comparing results with the exact temperature distribution under specified boundary conditions. The solution involves expressing the temperature distribution as a series expansion.For the analytical part of the equation we used, the equations are available in Book of Bird, page 381 [1]. For a sphere, the solution can be written as:

$$\theta(\xi, \tau) = 1 + \sum_{n=1}^{\infty} \frac{2(-1)^n}{n\pi\xi} \sin(n\pi\xi) \exp(-n^2\pi^2\tau)$$

Where:

- θ is the dimensionless temperature.
- $\xi = \frac{r}{a}$ is the dimensionless radial position (with r being the radial position and a the radius of the sphere).
- $\tau = \frac{\alpha t}{a^2}$ is the dimensionless time (with α being the thermal diffusivity and t the time).
- The series terms involve $\sin(n\pi\xi)$ and the exponential decay term $\exp(-n^2\pi^2\tau)$.

2. Implicit Numerical Method

This method ensures stability and accuracy, even for larger time steps, making it suitable for our spherical geometry. By discretizing the governing equation, we calculate the temperature at each grid point

iteratively, incorporating the effects of radial symmetry and the Fourier number (Fo). This approach allows us to model the heat diffusion process effectively and compare it directly with the analytical solution for validation. The temperature T_s in the solid is calculated using the implicit finite difference method.

The implicit discretization scheme for the heat equation is given as:

$$\frac{\partial \theta}{\partial \tau} = \frac{1}{\xi^2} \frac{\partial}{\partial \xi} \left(\xi^2 \frac{\partial \theta}{\partial \xi} \right)$$

The discretized form of the equation is:

$$\theta_i^{n+1} - \theta_i^n = \text{Fo} \left(\frac{1}{\xi_i^2} \left[\frac{\xi_{i+\frac{1}{2}}^2 \left(\theta_{i+1}^{n+1} - \theta_i^{n+1} \right)}{\Delta \xi} - \frac{\xi_{i-\frac{1}{2}}^2 \left(\theta_i^{n+1} - \theta_{i-1}^{n+1} \right)}{\Delta \xi} \right] \right)$$

Rearranging the Terms Rearranging gives:

$$\left[\theta_i^{n+1} - \text{Fo} \left(\frac{\xi_{i+\frac{1}{2}}^2}{\Delta \xi^2} \theta_{i+1}^{n+1} - \left(\frac{\xi_{i+\frac{1}{2}}^2 + \xi_{i-\frac{1}{2}}^2}{\Delta \xi^2} \right) \theta_i^{n+1} + \frac{\xi_{i-\frac{1}{2}}^2}{\Delta \xi^2} \theta_{i-1}^{n+1} \right) \right] = \theta_i^n$$

The matrix form for the implicit method can be written as:

$$A \theta^{n+1} = b$$

where A is a tridiagonal matrix with the following components:

$$A_{i,i-1} = -\text{Fo} \cdot \frac{\xi_{i-\frac{1}{2}}^2}{\xi_i^2 \Delta \xi^2}$$

$$A_{i,i} = 1 + \text{Fo} \cdot \frac{\xi_{i+\frac{1}{2}}^2 + \xi_{i-\frac{1}{2}}^2}{\xi_i^2 \Delta \xi^2}$$

$$A_{i,i+1} = -\text{Fo} \cdot \frac{\xi_{i+\frac{1}{2}}^2}{\xi_i^2 \Delta \xi^2}$$

The temperature at each grid point is updated using the implicit finite difference approximation. This approach ensures stability for the numerical solution and involves solving a system of equations for the unknown temperatures at the next time step.

For each grid point i, the implicit scheme calculates the updated temperature by considering contributions from neighboring points and the current state of the system, as defined by the discretized governing equations.

$$\Theta_s^{n+1}(i) = \Theta_s^n(i) + \frac{\text{Fo}}{\xi_i^2} \left(\xi_{i+1/2}^2 \Theta_s^{n+1}(i+1) - \left(\xi_{i+1/2}^2 + \xi_{i-1/2}^2 \right) \Theta_s^{n+1}(i) + \xi_{i-1/2}^2 \Theta_s^{n+1}(i-1) \right)$$

where:

$$\text{Fo} = \frac{\alpha \Delta t}{\Delta \xi^2}$$

is the Fourier number, α is the thermal diffusivity, and ξ_i represents the spatial grid location.

This equation is derived from the heat diffusion equation in spherical coordinates, accounting for the geometry of the sphere and discretizing both time and space. The symmetry of the sphere and boundary conditions are applied:

- At $\xi = 0$: $T_s(0) = T_s(1)$,
- At $\xi = 1$: $T_s(\text{end}) = T_l$, where T_l is the temperature of the interacting liquid phase.

3. Explicit Numerical Method

Using a forward difference scheme for the time derivative, the method directly computes the temperature at each step based on the previous one. Boundary conditions ensure symmetry, continuity at the interface, and proper initial conditions, making this method efficient for modeling heat transfer with small time steps. This approach provides a clear comparison with the analytical and implicit methods.

Solid Phase:

$$\Theta_s^{n+1} = \frac{1}{\xi_i^2} Fo \left(\xi_{i+\frac{1}{2}}^2 \Theta_{s_{i+1}} - \Theta_{s_i} \left(\xi_{i+\frac{1}{2}}^2 + \xi_{i-\frac{1}{2}}^2 \right) + \xi_{i-\frac{1}{2}}^2 \Theta_{s_{i-1}} \right) + \Theta_{s_i}$$

Liquid Phase:

$$\Theta_f^{n+1} = -\frac{3\Delta\tau}{B} \frac{\left(3\Theta_{s_{i+1}}^n - 4\Theta_{s_i}^n + \Theta_{s_{i-1}}^n \right)}{2\Delta\xi} + \Theta_f^n$$

The boundary conditions for the solid and liquid phases are defined as follows:

Solid Phase:

$$\begin{aligned} &\text{At } \tau = 0, \quad \Theta_s = 0 \\ &\text{At } \xi = 1, \quad \Theta_s = \Theta_f \\ &\text{At } \xi = 0, \quad \frac{\partial \Theta_s}{\partial \xi} = 0, \Theta_{s_{i+1}} = \Theta_{s_{i-1}} \end{aligned}$$

Liquid Phase:

$$\text{At } \tau = 0, \quad \Theta_f = 1$$

where:

- $\Delta\tau$ is the dimensionless time step.
- Θ_f^n is the dimensionless fluid temperature at time step n.
- Θ_f^{n+1} is the dimensionless fluid temperature at the next time step $n+1$.
- k_s is the thermal conductivity of the particle.
- V_s is the volume of the particle.
- ρ_f is the density of the fluid.
- $C_{p,f}$ is the specific heat capacity of the fluid.
- V_f is the volume of the fluid.
- $\left.\frac{\partial \Theta_s}{\partial \xi}\right|_{\xi=1}$ is the dimensionless temperature gradient at the surface of the particle.

4 | Computational Methodology

1. Implicit Method

The computational methodology employed in this analysis involves solving the transient heat conduction problem in a solid-liquid system using finite difference methods.This Python code simulates the heat transfer process between a solid sphere and a liquid phase using an implicit finite difference method. The objective is to analyze the rescaled evolution of the liquid temperature over dimensionless time (τ) for different values of the parameter B, which governs the heat transfer interaction between the solid and liquid phases. The code snippet can be found in the Appendix, listing 1. The key steps in the code are outlined as follows:

Initialization

- The spatial grid (ξ) and temporal grid (τ) are defined, along with the thermal diffusivity (α), time step size ($\Delta\tau$), spatial step size ($\Delta\xi$), and the Fourier number (Fo).
- A range of B values is specified to study the interaction strength between the solid and liquid phases.

Matrix Assembly

- An implicit tridiagonal matrix A is constructed to solve the heat equation in the solid phase. This matrix incorporates the effects of radial symmetry and the Fourier number.

Time Evolution

- For each time step, the temperature in the solid phase is updated by solving the linear system $A \cdot T_s^{n+1} = b$, where b is the right-hand side vector that accounts for boundary conditions.
- The liquid temperature is updated using a finite difference approximation for the interaction term at the solid-liquid interface.

Rescaling and Plotting

- The liquid temperature is rescaled as $(1+B)\cdot(1-\theta_{\text{liquid}})$, and its evolution is tracked for all specified B values.
- Finally, the rescaled liquid temperature is plotted against the dimensionless time τ, providing a visualization of the effect of B on the heat transfer dynamics.

This simulation efficiently models the coupled heat transfer problem and provides insights into the relationship between the solid and liquid phases under varying interaction strengths.

2. Explicit Method

Explicit Python code simulates the heat transfer process between a solid sphere and a liquid phase, utilizing an implicit finite difference method. The purpose is to analyze how the rescaled liquid temperature evolves over dimensionless time (τ) for different values of the parameter B, which governs the solid-liquid interaction strength. The main steps are described below:

Initialization

- Simulation parameters, including the number of time steps (timesteps), maximum dimensionless time ($\tau_{\max}$), and grid resolution ($\Delta\xi$), are defined.
- The Fourier number (Fo) is calculated using the thermal diffusivity (α) and spatial and temporal step sizes.
- A range of B values is specified to investigate different coupling strengths between the solid and liquid.

Implicit Matrix Setup

- A tridiagonal matrix A is constructed to model the heat equation for the solid phase. The entries of A account for the effects of radial symmetry, the Fourier number, and spatial discretization. Boundary conditions are applied at $\xi = 0$ (symmetry) and $\xi = 1$ (solid-liquid interaction).

Time Evolution

- At each time step, the right-hand side (RHS) vector b is assembled, incorporating the boundary conditions.
- The temperature distribution in the solid (T_s) is updated by solving the linear system $A \cdot T_s^{n+1} = b$.
- The liquid temperature (T_l) is updated using a finite difference approximation for the interaction term at the solid-liquid interface.

Rescaling and Plotting

- The liquid temperature is rescaled using $(1 + B) \cdot (1 - \theta_{\text{liquid}})$ to observe its evolution for each B.
- The rescaled values are plotted against the dimensionless time τ, providing insights into the heat transfer dynamics for various interaction strengths.

This code offers an efficient and stable way to model coupled heat transfer and visualize how the interaction strength B affects the liquid temperature evolution.

5 | Analysis of Results

Figure 5.1 compares the numerical and analytical solutions for the dimensionless solid temperature profile at various maximum dimensionless times, τ_{max}. The agreement between the numerical and analytical solutions confirms the correctness of the numerical method for computing solid temperature profiles.

As τ_{max} increases, the profiles exhibit more pronounced curvature, which aligns well with theoretical expectations. The validation of numerical results against the analytical solution provides confidence in the robustness of the implemented method.

Additionally, the numerical results were cross-verified with similar temperature profiles provided in *Transport Phenomena* by Bird, Stewart, and Lightfoot [1]. This further corroborates the reliability of the numerical approach used in this study.

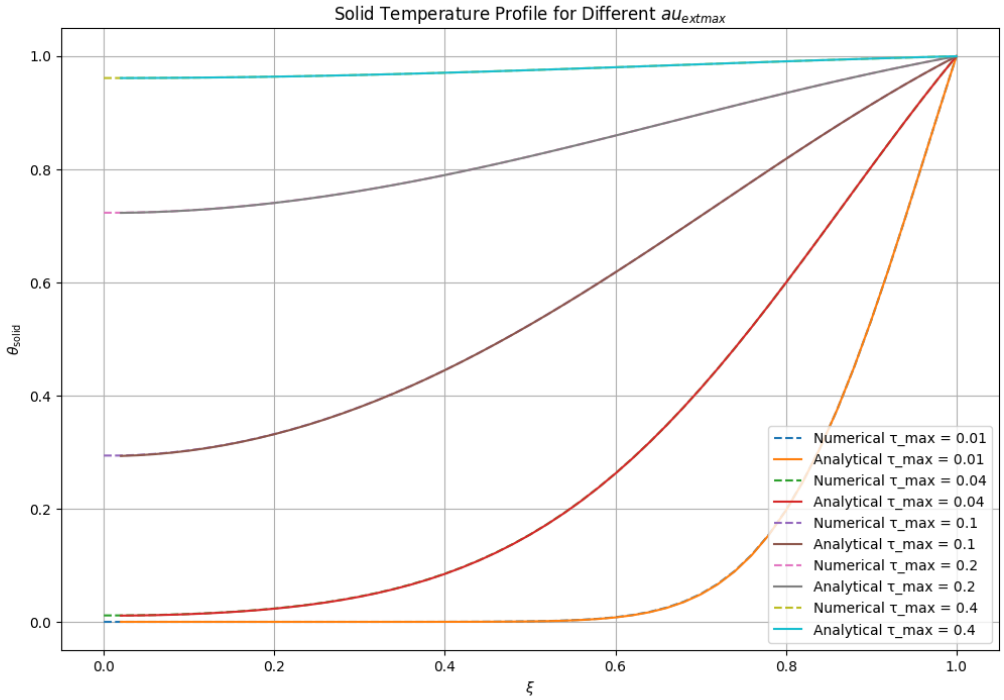

Figure 5.1: Comparison of numerical and analytical solutions for solid temperature for different τ

Numerical Methods for Dimensionless Liquid Temperature

In this study, both explicit and implicit numerical methods were employed to analyze the dimensionless temperature profiles of the liquid. These methods were chosen for their distinct characteristics: the explicit method, being straightforward and computationally efficient, provides insight into stability requirements, while the implicit method ensures numerical stability even for larger time steps. The combination of these methods allows for a comprehensive validation of the numerical results against the analytical solution.

To evaluate the accuracy of the numerical methods, the error between the numerical and analytical solutions was quantified using various norms:

$$MAE = L_1 = \frac{1}{n}\sum_{i=1}^{n} |\phi - \phi_{\text{real}}|\,,$$

$$RMSE = L_2 = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |\phi - \phi_{\text{real}}|^2},$$
$$L_\infty = \max |\phi - \phi_{\text{real}}|,$$

where ϕ is the numerical solution, ϕ_{real} is the analytical solution, and n is the number of spatial grid points. These norms provide insights into the convergence and accuracy of the numerical schemes under varying grid resolutions and time steps.

Explicit Method Analysis

Figure 5.2 illustrates the comparison between the explicit numerical method and the analytical solution for the dimensionless liquid temperature. The analytical solution, derived and validated using the book *Transport Phenomena* by Bird, Stewart, and Lightfoot [1], serves as the benchmark.

The numerical results obtained using the explicit method align closely with the analytical solution, confirming the accuracy of the discretized equations. For this comparison, the explicit method was implemented with a spatial grid of 51 steps, a time step of 1100 iterations, and a diffusivity parameter $\alpha = 1$. Despite the inherent stability limitations of the explicit scheme, the agreement demonstrates that the timestep size was appropriately chosen to ensure numerical stability and convergence.

This validation of the explicit method highlights its reliability for computing the temperature profiles in this problem, provided the timestep adheres to the stability criteria of the scheme.

The analytical solution [1] for the dimensionless temperature Θ_f of the fluid was derived using the following equations. The Heaviside partial fractions expansion theorem was applied, with:

$$\frac{N(0)}{D'(0)} = -\frac{1}{3(1+B)}, \quad \frac{N(p_k)}{D'(p_k)} = \frac{2B}{9(1+B)+B^2 b_k^2},$$

where b_k are the nonzero roots of $\tan b_k = \frac{3b_k}{3+Bb_k^2}$, and $p_k = ib_k$ (for $k = 1, 2, 3, \ldots, \infty$). Substituting these into the general form yields:

$$\Theta_f = \frac{B}{1+B} + 6B\sum_{k=1}^{\infty} \frac{\exp\left(-b_k^2\tau\right)}{9(1+B)+B^2 b_k^2},$$

where $\tau = \alpha_s t/R^2$ is the dimensionless time.

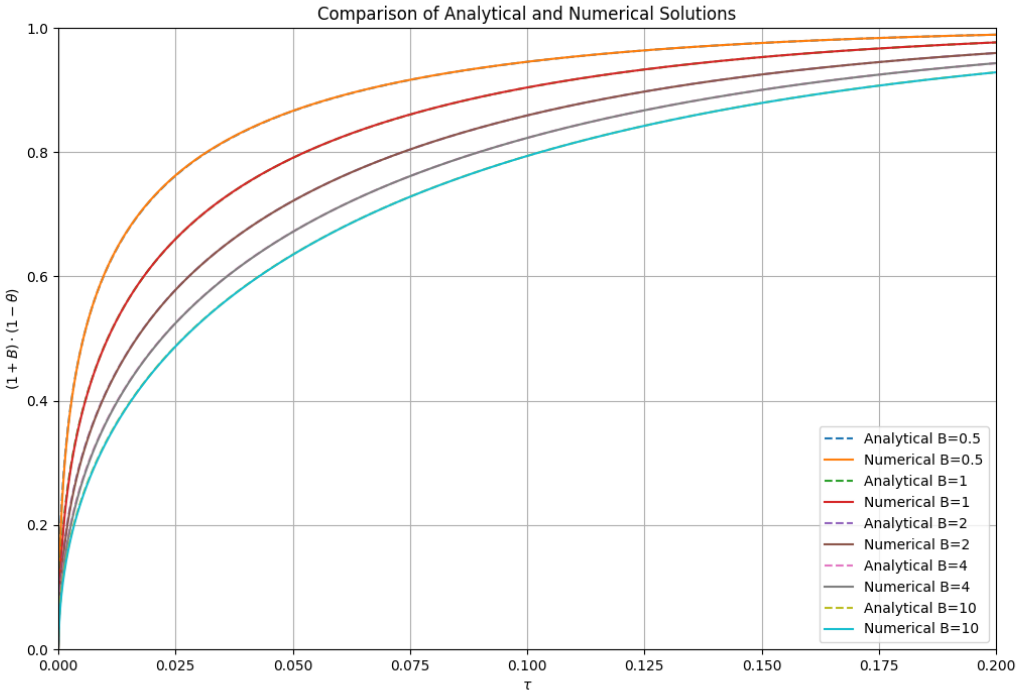

Figure 5.2: Comparison of Explicit method with analytical solution, grid steps = 51, $\alpha = 1$, time steps = 1100

The explicit method was used to compute the dimensionless liquid temperature profiles, and its results were compared with the analytical solution (Figure 5.2). The comparison demonstrates close agreement between the numerical and analytical solutions for grid steps of 51, a time step count of 1100, and $\alpha = 1$, validating the implementation of the explicit method.

To further evaluate the performance of the explicit method, norm errors (L_1, L_2, and L_∞) were computed for varying values of thermal diffusivity α and parameter B. Figure 5.3 shows the error norms for different α values, indicating that the error is minimized for $\alpha = 1$.

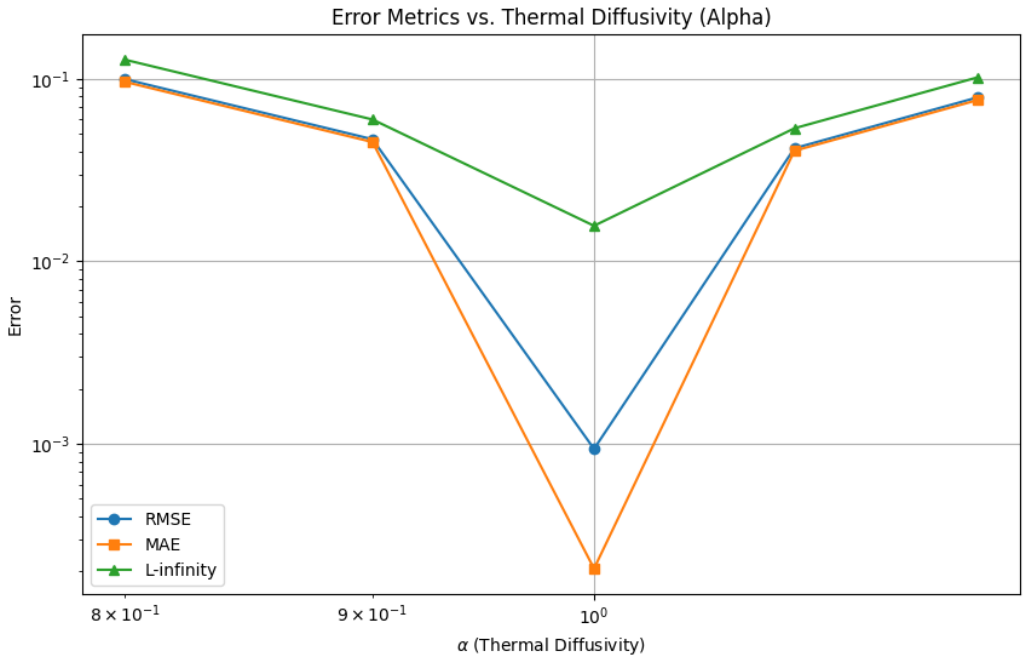

Figure 5.3: Explicit: Norm error for different alpha values

Similarly, Figure 5.4 depicts the norm errors for varying B values, with $B = 2$ used as a representative case. The results confirm that the explicit method performs well within stability and accuracy constraints when appropriate parameter values are selected.

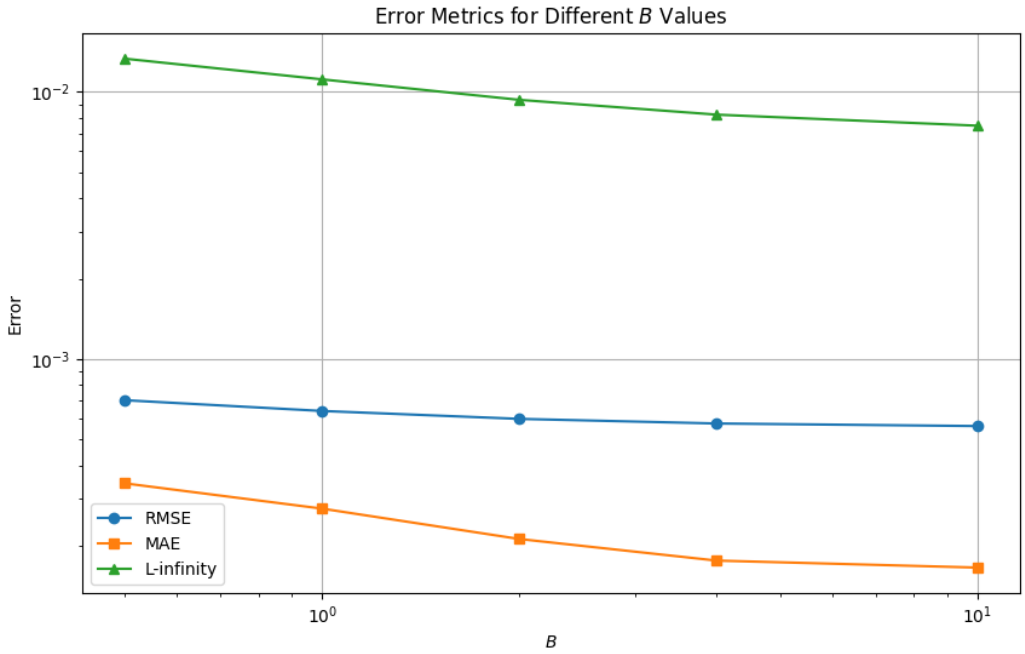

Figure 5.4: Explicit: Norm error for different B values

Figure 5.5 illustrates the error trends for different timesteps in the explicit method. The results indicate

that a timestep count of 1100 serves as a threshold for minimizing the error. Beyond this threshold, the accuracy does not significantly improve, making 1100 timesteps an optimal choice for balancing computational cost and accuracy.

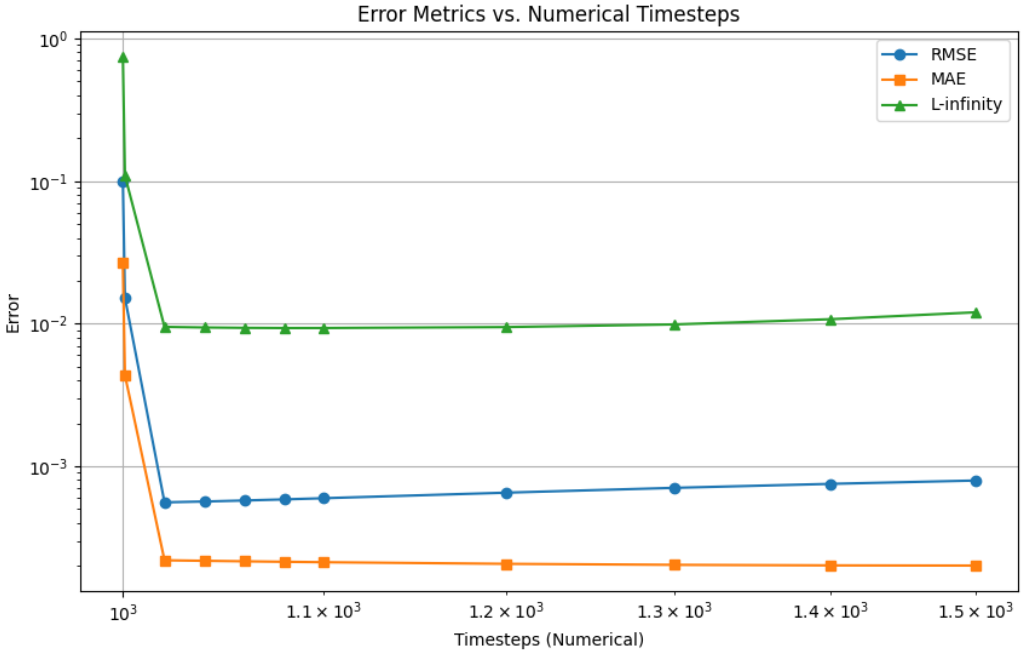

Figure 5.5: Explicit: Norm error for different time steps

Attempts were also made to compute errors for different grid steps. However, due to the limitations of the explicit method related to stability (with the stability criterion Fo < 0.5, where Fo $= \alpha \cdot \Delta t / \Delta \xi^2$), it was challenging to analyze this parameter effectively. The numerical instability restricted the range of grid steps that could be studied. Consequently, it was decided to investigate how the grid steps influence the accuracy of the numerical solution using the implicit method, which does not face these stability constraints.

Implicit Method Analysis

Similar to the explicit method, the implicit method was employed to compute the dimensionless liquid temperature profiles, and its accuracy was analyzed by evaluating norm errors (L_1, L_2, and L_∞). The parameter B was fixed at a value of 2 for all cases.

The first analysis focused on the effect of the thermal diffusivity α on the numerical accuracy. As shown in Figure 5.6, the error is minimized for $\alpha = 1$, similar to the explicit method. This indicates that $\alpha = 1$ is an optimal parameter for achieving accurate results.

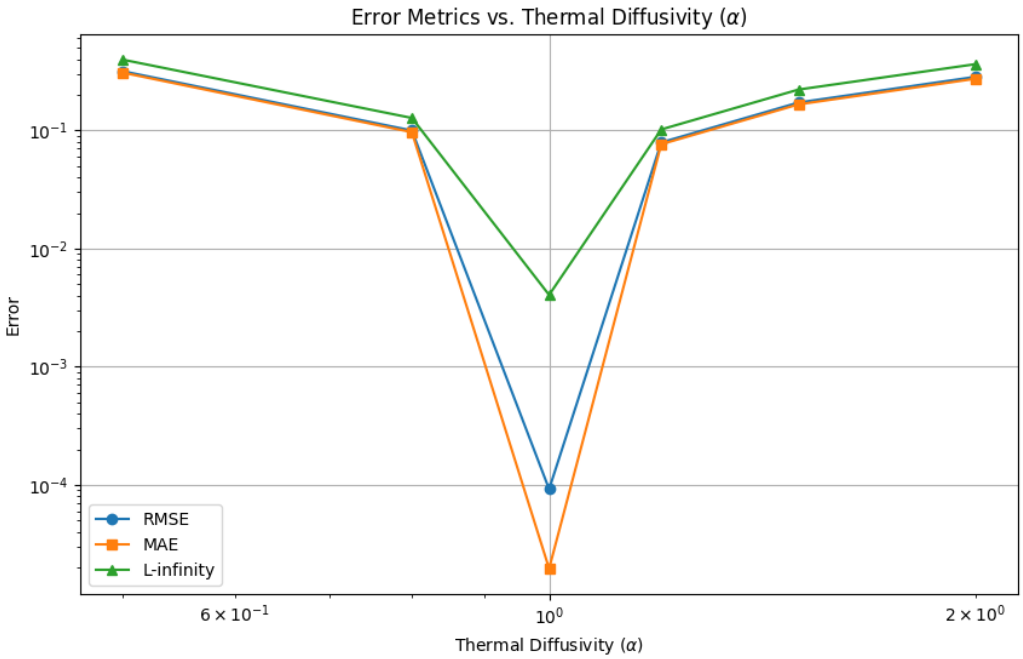

Figure 5.6: Implicit: Norm error for different alpha values

Next, the influence of the grid steps on the numerical solution was analyzed. Figure 5.7 illustrates that using a grid step count of 401 is sufficient, as larger grid steps have an insignificant impact on improving the accuracy of the solution. This makes 401 grid steps an optimal choice for balancing computational efficiency and accuracy.

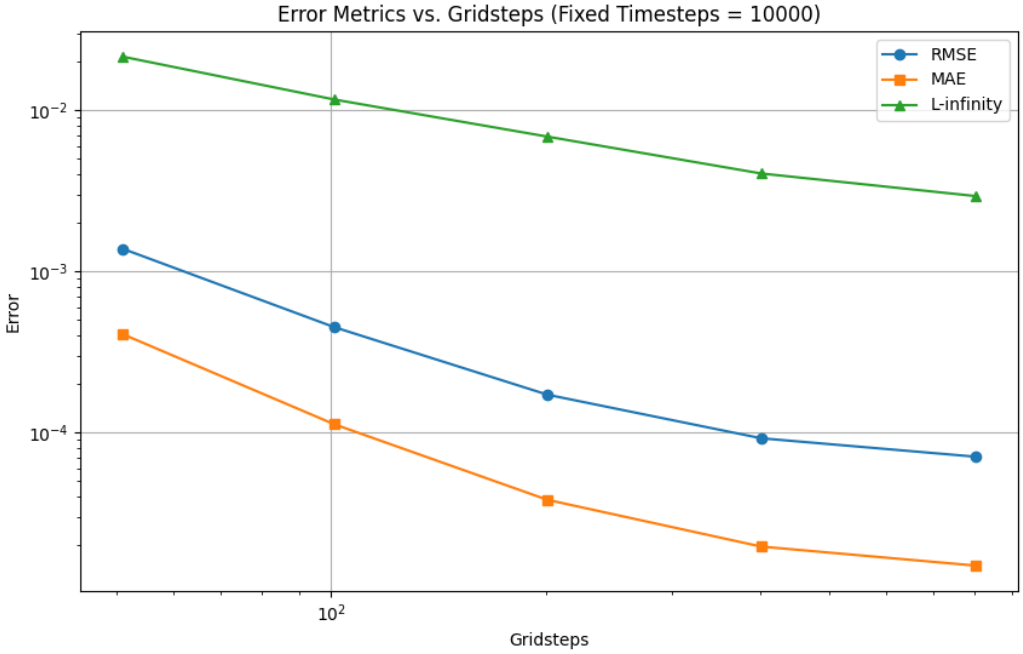

Figure 5.7: Implicit: Norm error for different grid steps

Lastly, the effect of the timestep was evaluated. Figure 5.8 demonstrates that the implicit method achieves better accuracy even for larger timesteps compared to the explicit method. This behavior is expected due to the inherent stability of the implicit scheme. A timestep count of 10,000 was selected, as it balances accuracy and computational cost effectively. Notably, the implicit method provides accurate results even for smaller timesteps (e.g., 100), which is a significant improvement over the explicit method.

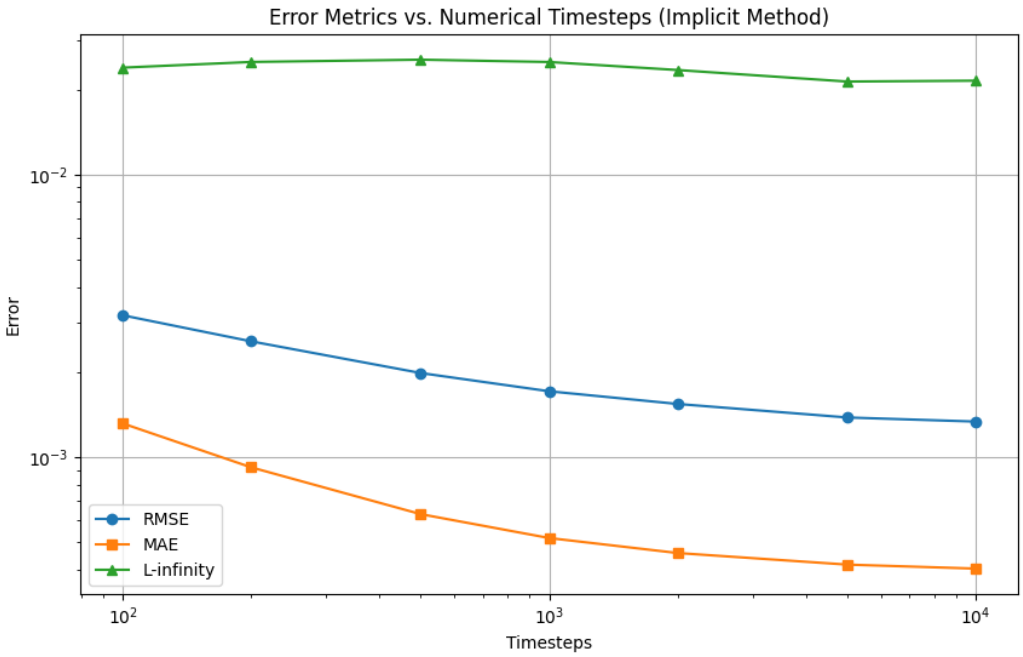

Figure 5.8: Implicit: Norm error for different time steps

Based on the chosen grid steps, timesteps, and thermal diffusivity α, the final graph of rescaled liquid temperature evolution was created for the implicit method. Figure 5.9 shows the numerical solution for various B values.

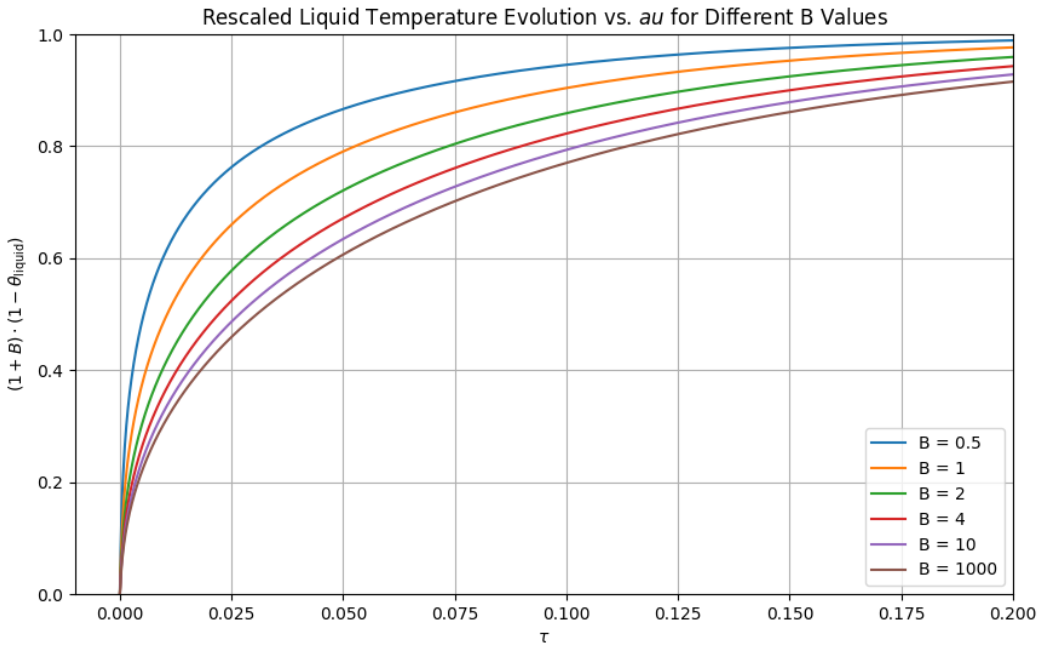

Figure 5.9: Implicit method, with grid steps $= 51$, $\alpha = 1, timesteps = 1100$

6 | Conclusion

This study explored the transient heat transfer dynamics of a homogeneous solid sphere immersed in a well-stirred liquid, employing both explicit and implicit numerical methods. The governing equations for the solid and liquid phases were derived and discretized to facilitate numerical solution, while analytical solutions served as benchmarks for validation.

The explicit method demonstrated close agreement with the analytical solution, confirming its reliability for small time steps. However, its inherent stability limitations restricted its applicability, particularly when analyzing the effects of grid steps. In contrast, the implicit method exhibited superior stability and accuracy, enabling the use of larger time steps and providing robust results across all parameter configurations. The norm errors (L_1, L_2, and L_∞) confirmed that the optimal thermal diffusivity $\alpha = 1$, grid step count of 401, and timestep of 10,000 achieved the best balance between accuracy and computational efficiency.

The final graph, showcasing the rescaled liquid temperature evolution for varying interaction strengths (B values), highlighted the implicit method's robustness and accuracy in capturing the coupled heat transfer dynamics. This comprehensive analysis underscores the strengths and limitations of each numerical method, providing valuable insights for solving similar heat transfer problems in complex systems.

7 | References

[1] R. Byron Bird, Warren E. Stewart, and Edwin N. Lightfoot. *Transport Phenomena*. John Wiley & Sons, Inc., 2nd edition, 2002.

A | Appendix A

Python code - Implicit method

Listing 1: Code snippet to compute liquid temperature with implicit method

```python
import numpy as np
import matplotlib.pyplot as plt

# Simulation parameters
timesteps = 1100
tau_max = 0.2
gridsteps = 51
alpha = 1  # Example thermal diffusivity
delta_tau = tau_max / timesteps  # Time step size
delta_xi = 1 / (gridsteps - 1)  # Spatial step size

# Calculate the Fourier number
Fo = alpha * delta_tau / delta_xi**2

# Values of B to consider
B_values = [0.5, 1, 2, 4, 10, 1000]

# Initialize plot
plt.figure(figsize=(10, 6))

for B in B_values:
    # Initialize temperatures for solid and liquid
    T_s = np.zeros(gridsteps)  # Solid initially zero
    T_l = 1.0  # Liquid initially one (normalized)

    # Time evolution of temperatures
    theta_l_all = [T_l]  # Track liquid temperature

    # Implicit matrix setup
    A = np.zeros((gridsteps, gridsteps))
    for i in range(1, gridsteps - 1):
        xi = i * delta_xi
        xi_plus_half = xi + delta_xi / 2
        xi_minus_half = xi - delta_xi / 2
        A[i, i - 1] = -Fo * xi_minus_half**2 / xi**2
        A[i, i] = 1 + Fo * (xi_plus_half**2 + xi_minus_half**2) / xi**2
        A[i, i + 1] = -Fo * xi_plus_half**2 / xi**2

    A[0, 0] = 1
    A[gridsteps - 1, gridsteps - 1] = 1

    for _ in range(timesteps):
        # Setup the right-hand side (RHS) of the implicit equation
        b = T_s.copy()
        b[0] = T_s[1]  # Boundary condition at xi = 0 (symmetry)
        b[gridsteps - 1] = T_l  # Boundary condition at xi = 1 (
            interaction)

        # Solve the linear system A * T_s_new = b
        T_s_new = np.linalg.solve(A, b)
        T_s = T_s_new.copy()  # Update the old array with new values

```

```
        # Update liquid temperature
        interaction_term = (-3 * delta_tau / (B * 2 * delta_xi)) * (
            3 * T_s[-1] - 4 * T_s[-2] + T_s[-3])
        T_l += interaction_term
        theta_l_all.append(T_l)  # Track liquid temperature

    # Calculate the rescaled values for (1 + B) * (1 - theta_liquid)
    rescaled_theta_l = [(1 + B) * (1 - theta_l) for theta_l in
        theta_l_all]

    # Plot for the current B
    tau = np.linspace(0, timesteps * delta_tau, timesteps + 1)
    plt.plot(tau, rescaled_theta_l, label=f'B = {B}')

# Finalize plot
plt.xlabel(r'$\tau$')
plt.ylabel(r'$(1 + B) \cdot (1 - \theta_{\mathrm{liquid}})$')
plt.title('Rescaled Liquid Temperature Evolution vs. $\tau$ for
    Different B Values')
plt.legend()
plt.grid()
plt.xlim(-0.01, 0.2)  # Set x-axis limits
plt.ylim(0, 1.0)  # Set y-axis limits
plt.show()
```

Python code - Explicit method

Listing 2: Code snippet to compute liquid temperature with implicit method

```
import numpy as np
import matplotlib.pyplot as plt

# Simulation parameters
timesteps = 1100
tau_max = 0.2
gridsteps = 51
alpha = 1  # Example thermal diffusivity
delta_tau = tau_max / timesteps  # Time step size
delta_xi = 1 / (gridsteps - 1)  # Spatial step size

# Calculate the Fourier number
Fo = alpha * delta_tau / delta_xi**2

# Values of B to consider
B_values = [0.5, 1, 2, 4, 10, 1000]

# Initialize plot
plt.figure(figsize=(10, 6))

for B in B_values:
    # Initialize temperatures for solid and liquid
    T_s = np.zeros(gridsteps)  # Solid initially zero
    T_l = 1.0  # Liquid initially one (normalized)

    # Time evolution of temperatures
    theta_l_all = [T_l]  # Track liquid temperature

```

```
    # Implicit matrix setup
    A = np.zeros((gridsteps, gridsteps))
    for i in range(1, gridsteps - 1):
        xi = i * delta_xi
        xi_plus_half = xi + delta_xi / 2
        xi_minus_half = xi - delta_xi / 2
        A[i, i - 1] = -Fo * xi_minus_half*2 / xi*2
        A[i, i] = 1 + Fo * (xi_plus_half*2 + xi_minus_half2) / xi*2
        A[i, i + 1] = -Fo * xi_plus_half*2 / xi*2

    A[0, 0] = 1
    A[gridsteps - 1, gridsteps - 1] = 1

    for _ in range(timesteps):
        # Setup the right-hand side (RHS) of the implicit equation
        b = T_s.copy()
        b[0] = T_s[1]  # Boundary condition at    = 0 (symmetry)
        b[gridsteps - 1] = T_l  # Boundary condition at    = 1 (
            interaction)

        # Solve the linear system A * T_s_new = b
        T_s_new = np.linalg.solve(A, b)
        T_s = T_s_new.copy()  # Update the old array with new values

        # Update liquid temperature
        interaction_term = (-3 * delta_tau / (B * 2 * delta_xi)) * (
            3 * T_s[-1] - 4 * T_s[-2] + T_s[-3])
        T_l += interaction_term
        theta_l_all.append(T_l)  # Track liquid temperature

    # Calculate the rescaled values for (1 + B) * (1 - theta_liquid)
    rescaled_theta_l = [(1 + B) * (1 - theta_l) for theta_l in
        theta_l_all]

    # Plot for the current B
    tau = np.linspace(0, timesteps * delta_tau, timesteps + 1)
    plt.plot(tau, rescaled_theta_l, label=f'B = {B}')

# Finalize plot
plt.xlabel(r'$\tau$')
plt.ylabel(r'$(1 + B) \cdot (1 - \theta_{\mathrm{liquid}})$')
plt.title('Rescaled Liquid Temperature Evolution vs. $\tau$ for
    Different B Values')
plt.legend()
plt.grid()
plt.xlim(-0.01, 0.2)  # Set x-axis limits
plt.ylim(0, 1.0)  # Set y-axis limits
plt.show()
```