Partial Differential Equation-Based Heat Transfer and Solidification Model

Developed and Implemented by Roozbeh Aghabarari

Contact Me
ro.aghabarari@gmail.com
www.roozbehaghabarari.com

Partial Differential Equation-Based Heat Transfer and Solidification Model

This simulation aims to model the heat transfer and solidification processes in a steel ingot using partial differential equations (PDEs). The physical properties employed in the model are derived from values reported in textbooks and authoritative references in materials science. The MATLAB code developed for this study is subsequently presented, followed by a detailed explanation of each section of the manuscript.

1. MATLAB code

```
% PDE-Based Heat Transfer & Solidification Model
%                    Coded by
%               Roozbeh Aghabarari
%
%                   Contact Me
%             ro.aghabarari@gmail.com
%           www.roozbehaghabarari.com
%
% Parameters
L = 0.762;                % Length of the ingot (m)
Tmax = 6000;              % Maximum simulation time (s)
dx = 0.009525;            % Spatial step (m)
dt = 0.1;                 % Time step (s)
K = 30;                   % Thermal conductivity (W/(m*K))
T_wall = 1150;            % Fixed wall temperature (°C)
T_inf = 20;               % Ambient temperature (°C)
rho = 7200;               % Density of steel (kg/m^3)
cp_solid = 682;           % Specific heat capacity of solid steel (J/(kg*K))
cp_liquid = 710;          % Specific heat capacity of liquid steel (J/(kg*K))
Lf = 262500;              % Latent heat of fusion (J/kg)
Tm = 1500;                % Melting temperature of steel (°C)

% Discretization
x = 0:dx:L;               % Spatial grid in x-direction
y = 0:dx:L;               % Spatial grid in y-direction
time = 0:dt:Tmax;         % Time grid
Nx = length(x);           % Number of spatial points in x-direction
Ny = length(y);           % Number of spatial points in y-direction
Nt = length(time);        % Number of time points

% Initial condition
T = ones(Nx, Ny) * 1535;  % Initial temperature (molten steel at 1535°C)

% Pre-allocate temperature and solidification fraction arrays
T_all = zeros(Nx, Ny, Nt);
solid_frac_all = zeros(Nx, Ny, Nt);
T_all(:, :, 1) = T;
solid_frac_all(:, :, 1) = solid_fraction(T, Tm);  % Initial solidification fraction
should be zero
```

```
% Main simulation loop
for n = 1:Nt-1
    T_new = T;
    cp_eff = effective_cp(T, cp_solid, cp_liquid, Lf, Tm);

    for i = 2:Nx-1
        for j = 2:Ny-1
            T_new(i, j) = T(i, j) + (K / (rho * cp_eff(i, j))) * dt / dx^2 * ...
                          ((T(i+1, j) - 2*T(i, j) + T(i-1, j)) + ...
                           (T(i, j+1) - 2*T(i, j) + T(i, j-1)));
        end
    end

    % Fixed boundary condition at all walls
    T_new(1, :) = T_wall;
    T_new(Nx, :) = T_wall;
    T_new(:, 1) = T_wall;
    T_new(:, Ny) = T_wall;

    T = T_new;
    T_all(:, :, n+1) = T;
    solid_frac_all(:, :, n+1) = solid_fraction(T, Tm);  % Calculate solidification
fraction
end

% Indices for the center of the quarter of the ingot and the center of the ingot
quarter_idx = floor(Nx/4);
half_idx = floor(Nx/2);

% Extract temperatures versus time
temperature_quarter = squeeze(T_all(quarter_idx, quarter_idx, :));
temperature_center = squeeze(T_all(half_idx, half_idx, :));

% Plot temperature versus time
figure;
plot(time, temperature_quarter, 'r', 'LineWidth', 2); hold on;
plot(time, temperature_center, 'b', 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Temperature (°C)');
title('Temperature vs Time');
legend('Quarter Center', 'Ingot Center');
grid on;

% Plot contour plots of temperature distribution at specific time points
time_points = 0:1000:Tmax;  % Specific time points for contour plots (based on hw-pde
file)

figure;
for i = 1:length(time_points)
    subplot(2, 4, i);
    contourf(x, y, T_all(:, :, time_points(i)/dt + 1)', 20, 'LineColor', 'none');
    colorbar;
    axis equal;  % Ensure the plot is cubic
    xlabel('x (m)');
    ylabel('y (m)');
```

```
    title(['t = ', num2str(time_points(i)), ' s']);
    hold on;
    plot(x(quarter_idx), y(quarter_idx), 'ro', 'MarkerSize', 10, 'LineWidth', 2);
    plot(x(half_idx), y(half_idx), 'bo', 'MarkerSize', 10, 'LineWidth', 2);
end
sgtitle('Temperature Contours at Different Time Points');
hold off;

% Calculate average solidification fraction versus time
avg_solid_frac = squeeze(mean(mean(solid_frac_all, 1), 2));

% Plot average solidification fraction versus time
figure;
plot(time, avg_solid_frac, 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Solidification Fraction');
title('Solidification Fraction vs Time');
grid on;

% Solidification fraction calculation function
function sf = solid_fraction(T, Tm)
    Tm_range = 1;  % Define a range around Tm in order to consider solidus and
liquidus temperatures
    sf = zeros(size(T));
    sf(T <= Tm - Tm_range) = 1;
    sf(T >= Tm + Tm_range) = 0;
    phase_change_indices = (T > Tm - Tm_range) & (T < Tm + Tm_range);
    sf(phase_change_indices) = (Tm + Tm_range - T(phase_change_indices)) / (2 *
Tm_range);
end

% Effective heat capacity function
function cp_eff = effective_cp(T, cp_solid, cp_liquid, Lf, Tm)
    cp_eff = cp_solid * ones(size(T));
    liquid_indices = T > Tm;
    cp_eff(liquid_indices) = cp_liquid;
    % Linear approximation of the phase change region
    phase_change_indices = (T > Tm - 1) & (T < Tm + 1);
    cp_eff(phase_change_indices) = cp_solid + (Lf / ((Tm + 1)-(Tm - 1)));
end
```

1.1. Parameters

```
% Parameters
L = 0.762;                % Length of the ingot (m)
Tmax = 6000;              % Maximum simulation time (s)
dx = 0.009525;            % Spatial step (m)
dt = 0.1;                 % Time step (s)
K = 30;                   % Thermal conductivity (W/(m*K))
T_wall = 1150;            % Fixed wall temperature (°C)
T_inf = 20;               % Ambient temperature (°C)
rho = 7200;               % Density of steel (kg/m^3)
cp_solid = 682;           % Specific heat capacity of solid steel (J/(kg*K))
cp_liquid = 710;          % Specific heat capacity of liquid steel (J/(kg*K))
Lf = 262500;              % Latent heat of fusion (J/kg)
Tm = 1500;                % Melting temperature of steel (°C)
```

This report describes the MATLAB code developed to simulate the temperature evolution and solidification process in a steel ingot. The simulation employs a two-dimensional explicit finite difference approach to solve the heat conduction equation with phase change.
This section outlines the physical and simulation parameters, including the domain dimensions, time step, material properties, and initial conditions, as summarized in Table 1 and derived from relevant reference sources. The simulation focuses on isothermal solidification in a transverse cross-section of a steel ingot with dimensions of $0.762\,m \times 0.762\,m$. The thermophysical properties are assumed to be identical for both the solid and liquid phases, with the exception of specific heat capacity, which differs between the two states to more accurately reflect real physical behavior. The initial temperature of the molten steel is set at 1535 ℃, exceeding the melting temperature of 1500 ℃. A fixed temperature boundary condition of 1150 ℃ is applied along the ingot–mold interface. The computational domain is discretized using a uniform 80 × 80 grid of square elements.

$$dx = \frac{Ingot\ length}{Number\ of\ grids\ in\ one\ direction} = \frac{0.762\,m}{80} = 0.009525\,m$$

The time step is selected based on the discrete Fourier number, defined as:

$$Fo = \frac{K\Delta t}{\rho C \Delta x^2}$$

For the explicit scheme, a time step of 0.1 is used to ensure stability (the stability limit is 0.375), while a larger time step of 16 is employed for the implicit scheme. The temperature history, temperature distribution (temperature counters), and solid fraction are evaluated at two key locations:

- Center of the ingot x = y = 0.381 m
- Center of the ingot quarter x = y = 0.1905 m

These points are indicated in Figure 1. The temporal evolution of temperature and solid fraction at these locations during the solidification process is presented in Figures 3 and 4, respectively.

Table 1
Thermophysical properties, boundary conditions, and initial conditions for steel ingot solidification.

k	30.0 $W/m.K$
c	750.0 $J/kg.K$
ρ	7200.0 kg/m^3
L	262.5 kJ/kg
T_m	1500.0 ℃
T_{ini}	1535.0 ℃
T_{wall}	1150.0 ℃

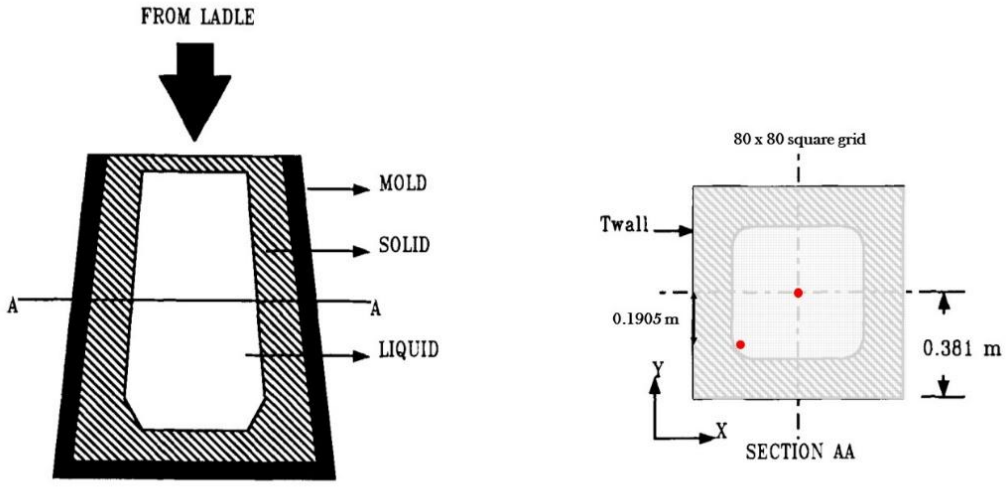

Fig. 1. Isothermal solidification of a steel ingot, and two mentioned points which are x = y = 0.381 m (center of the ingot) and x = y = 0.1905 m (center of the ingot quarter).

1.2. Discretization and main simulation loop

```
% Discretization
x = 0:dx:L;                % Spatial grid in x-direction
y = 0:dx:L;                % Spatial grid in y-direction
time = 0:dt:Tmax;          % Time grid
Nx = length(x);            % Number of spatial points in x-direction
Ny = length(y);            % Number of spatial points in y-direction
Nt = length(time);         % Number of time points

% Initial condition
T = ones(Nx, Ny) * 1535;  % Initial temperature (molten steel at 1535°C)

% Pre-allocate temperature and solidification fraction arrays
T_all = zeros(Nx, Ny, Nt);
solid_frac_all = zeros(Nx, Ny, Nt);
T_all(:, :, 1) = T;
solid_frac_all(:, :, 1) = solid_fraction(T, Tm);  % Initial solidification fraction
should be zero

% Main simulation loop
for n = 1:Nt-1
    T_new = T;
    cp_eff = effective_cp(T, cp_solid, cp_liquid, Lf, Tm);

    for i = 2:Nx-1
        for j = 2:Ny-1
            T_new(i, j) = T(i, j) + (K / (rho * cp_eff(i, j))) * dt / dx^2 * ...
                          ((T(i+1, j) - 2*T(i, j) + T(i-1, j)) + ...
                           (T(i, j+1) - 2*T(i, j) + T(i, j-1)));
        end
    end

    % Fixed boundary condition at all walls
    T_new(1, :) = T_wall;
    T_new(Nx, :) = T_wall;
    T_new(:, 1) = T_wall;
    T_new(:, Ny) = T_wall;

    T = T_new;
    T_all(:, :, n+1) = T;
    solid_frac_all(:, :, n+1) = solid_fraction(T, Tm);  % Calculate solidification
fraction
end
```

This section establishes the spatial and temporal grids for the simulation and determines the number of grid points. The temperature of the entire ingot is initially set to 1535 ℃, representing molten steel. Subsequently, arrays are pre-allocated to store the temperature and solidification fraction at each grid point for all time steps, with the initial solidification fraction set to zero.

The main simulation loop then iterates over each time step to update the temperature and solidification fraction arrays. These updates are performed using the `effective_cp` function, which is defined at the end of the code. The discretization of the governing equations—considered the

core of this simulation—is based on the explicit finite difference method for solving partial differential equations and is explained in detail in the following section.

1.2.1. Governing equations

The transient heat conduction during the solidification of a steel ingot is described by the one-dimensional heat equation.

$$\rho C_p \frac{\partial T}{\partial t} = \nabla.(K.\nabla T) + \dot{q}$$

Where ρ is the density (kg/m^3), C_p is the specific heat capacity ($J/kg.K$), T is the temperature (K), t is time (s), K is the thermal conductivity ($W/m.K$), and $\dot{q}$ is the internal heat generation term (W/m^3), which is assumed to be zero in this case. During solidification, C_p varies with temperature due to the release of latent heat. To account for this effect, an effective specific heat capacity ($C_{P,eff}$) is used in the model.

For a two-dimensional domain, the heat conduction during solidification is described by the transient heat equation as follows:

$$\rho C_p \frac{\partial T}{\partial t} = K\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

1.2.2. Discretization

To solve the heat conduction equation numerically, the explicit finite difference method is employed. The computational domain is discretized into a grid with spatial step sizes Δx and Δy, and a temporal step size Δt. Using central difference approximations for the spatial derivatives and a forward difference for the time derivative, the two-dimensional transient heat conduction equation is discretized as follows:

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2}$$
$$\frac{\partial^2 T}{\partial y^2} \approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2}$$

$$\frac{\partial T}{\partial t} \approx \frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t}$$

Finally:

$$\rho C_{p,eff} \frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} = K\left(\frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{(\Delta x)^2} + \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{(\Delta y)^2}\right)$$

Rearranging:

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \frac{K\Delta t}{\rho C_{p,eff}(\Delta x)^2}\left(T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}\right) + \frac{K\Delta t}{\rho C_{p,eff}(\Delta y)^2}\left(T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}\right)$$

In the implementation, to ensure an equal number of grid points in each spatial direction, it is assumed that $\Delta x = \Delta y$. Under this assumption, the discretized heat conduction equation simplifies to:

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \frac{K\Delta t}{\rho C_{p,eff}(\Delta x)^2}\left(T_{i+1,j}^{n} + T_{i-1,j}^{n} + T_{i,j+1}^{n} + T_{i,j-1}^{n} - 4T_{i,j}^{n}\right)$$

This formulation employs a five-point stencil to approximate the Laplacian in two dimensions, which is central to the explicit finite difference scheme utilized in this study. Here, $T_{i,j}^{n}$ represents the temperature at grid point (i, j) at time step n, and $C_{P,eff}$ denotes the effective specific heat capacity, which incorporates the effects of latent heat release during the solidification process.

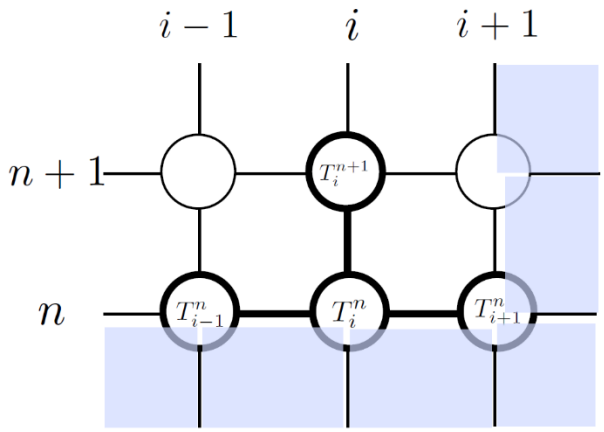

Fig. 2. Schematic representation of grids for explicit finite difference formulation.

1.3. Plots

```
% Indices for the center of the quarter of the ingot and the center of the ingot
quarter_idx = floor(Nx/4);
half_idx = floor(Nx/2);

% Extract temperatures versus time
temperature_quarter = squeeze(T_all(quarter_idx, quarter_idx, :));
temperature_center = squeeze(T_all(half_idx, half_idx, :));

% Plot temperature versus time
figure;
plot(time, temperature_quarter, 'r', 'LineWidth', 2); hold on;
plot(time, temperature_center, 'b', 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Temperature (°C)');
title('Temperature vs Time');
legend('Quarter Center', 'Ingot Center');
grid on;

% Plot contour plots of temperature distribution at specific time points
time_points = 0:1000:Tmax;  % Specific time points for contour plots (based on hw-pde
file)

figure;
for i = 1:length(time_points)
    subplot(2, 4, i);
    contourf(x, y, T_all(:, :, time_points(i)/dt + 1)', 20, 'LineColor', 'none');
    colorbar;
    axis equal;  % Ensure the plot is cubic
    xlabel('x (m)');
    ylabel('y (m)');
    title(['t = ', num2str(time_points(i)), ' s']);
    hold on;
    plot(x(quarter_idx), y(quarter_idx), 'ro', 'MarkerSize', 10, 'LineWidth', 2);
    plot(x(half_idx), y(half_idx), 'bo', 'MarkerSize', 10, 'LineWidth', 2);
end
sgtitle('Temperature Contours at Different Time Points');
hold off;

% Calculate average solidification fraction versus time
avg_solid_frac = squeeze(mean(mean(solid_frac_all, 1), 2));

% Plot average solidification fraction versus time
figure;
plot(time, avg_solid_frac, 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Solidification Fraction');
title('Solidification Fraction vs Time');
grid on;
```

This section extracts temperature data at selected locations within the domain—namely, at the center of the ingot and at the center of one of its quarter—and plots temperature as a function of time to observe thermal evolution at these points. In addition, contour plots of the temperature

distribution are generated at various time steps to visualize the spatial and temporal evolution of the temperature field. The solidification fraction is also computed across the entire ingot for each time step, and the average solidification fraction is plotted over time to illustrate the overall solidification progress.

1.4. Functions

```
% Solidification fraction calculation function
function sf = solid_fraction(T, Tm)
    Tm_range = 1;  % Define a range around Tm in order to consider solidus and
liquidus temperatures
    sf = zeros(size(T));
    sf(T <= Tm - Tm_range) = 1;
    sf(T >= Tm + Tm_range) = 0;
    phase_change_indices = (T > Tm - Tm_range) & (T < Tm + Tm_range);
    sf(phase_change_indices) = (Tm + Tm_range - T(phase_change_indices)) / (2 *
Tm_range);
end

% Effective heat capacity function
function cp_eff = effective_cp(T, cp_solid, cp_liquid, Lf, Tm)
    cp_eff = cp_solid * ones(size(T));
    liquid_indices = T > Tm;
    cp_eff(liquid_indices) = cp_liquid;
    % Linear approximation of the phase change region
    phase_change_indices = (T > Tm - 1) & (T < Tm + 1);
    cp_eff(phase_change_indices) = cp_solid + (Lf / ((Tm + 1)-(Tm - 1)));
end
```

First, a function is defined to calculate the effective specific heat capacity by incorporating the effects of phase change. The `effective_cp` function evaluates the effective heat capacity of the steel ingot during the solid-to-liquid transition. To identify liquid regions, the function checks for temperatures exceeding the melting temperature, assigning the specific heat capacity of liquid steel in these areas. To account for the phase change, a linear approximation is applied within a narrow temperature range around the melting point. Specifically, the phase change region is defined as temperatures within ± 1 ℃ of T_m, with the solidus and liquidus temperatures taken as $T_m - 1$ and $T_m + 1$, respectively. Within this interval, the effective specific heat capacity is modified to account for the latent heat of fusion (L_f) as follows:

$$C_{p,eff} = C_p + \frac{L_f}{T_l - T_s}$$

Subsequently, a separate function is defined to compute the solidification fraction of the ingot based on the temperature field. This `solid_fraction` function determines the fraction of material that has solidified at each grid point. The solidification fraction is set to 1 for regions where the temperature is less than or equal to $T_m - T_{m\_range}$, indicating complete solidification, and to 0 for regions where the temperature is greater than or equal to $T_m + T_{m\_range}$, indicating fully liquid steel. For temperatures between the solidus and liquidus limits, the fraction is interpolated linearly to represent partial solidification.

2. Results and discussion

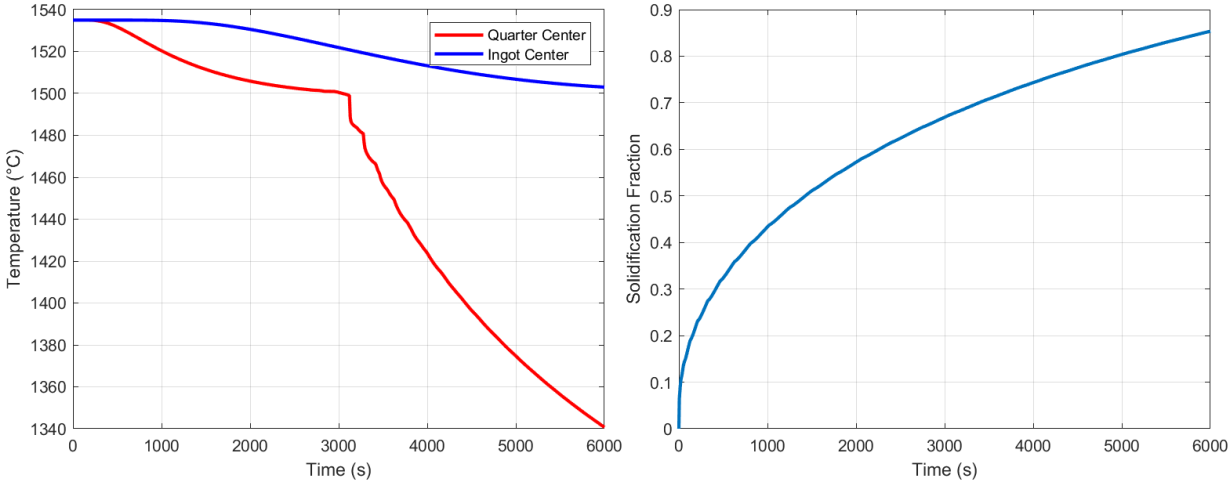

Fig. 3. Isothermal solidification of the steel ingot. The plots are generated using MATLAB. The left subplot shows the temperature history at two representative points: the quarter center and the ingot center. The right subplot illustrates the temporal progression of solidification throughout the entire ingot.

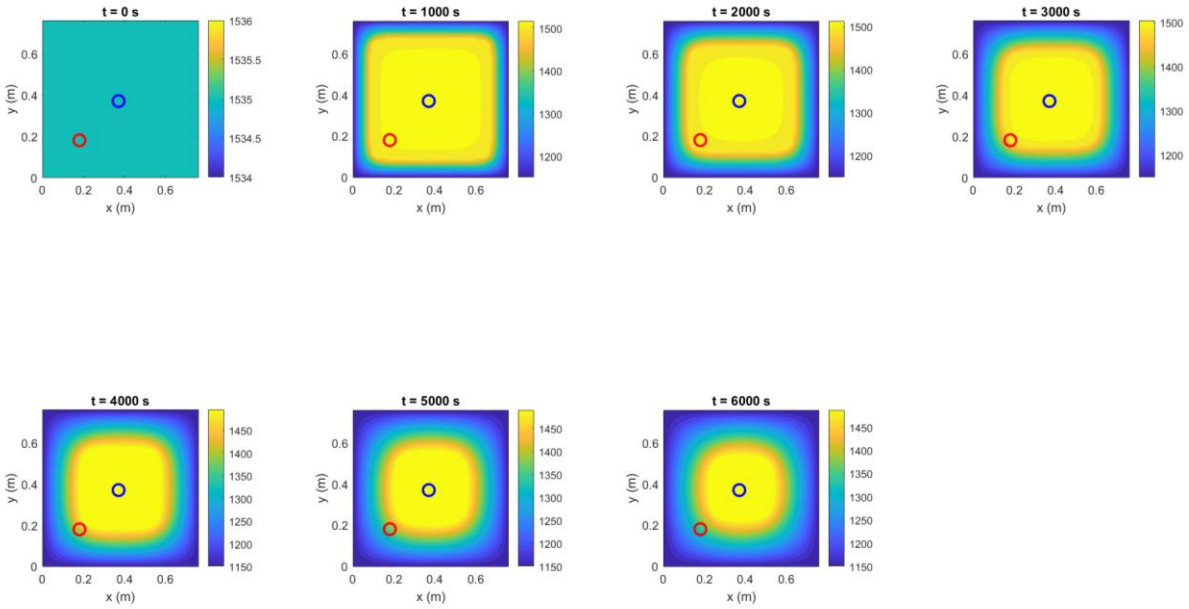

Fig. 4. Temperature contour plots of the entire ingot at selected time steps. The quarter center and ingot center points—used for temperature tracking—are indicated by circles.

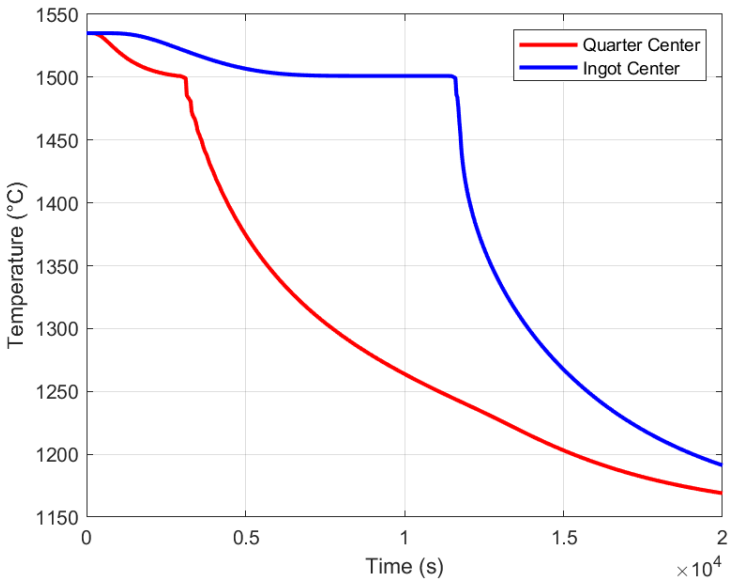

Fig. 5. Temperature history at the quarter center and the center of the ingot.

Figure 5 presents the results of a simulation run for 20000 seconds to capture the complete solidification process at the center of the ingot, which, due to slower cooling rates, naturally requires a longer time. This extended duration is both physically justified and consistent with the thermal behavior expected in such systems. The cooling rate at the center of the ingot is significantly lower than that at the quarter center, primarily due to the insulation effect of the surrounding material and the longer thermal path to the surface. This results in delayed solidification in the ingot's core region.