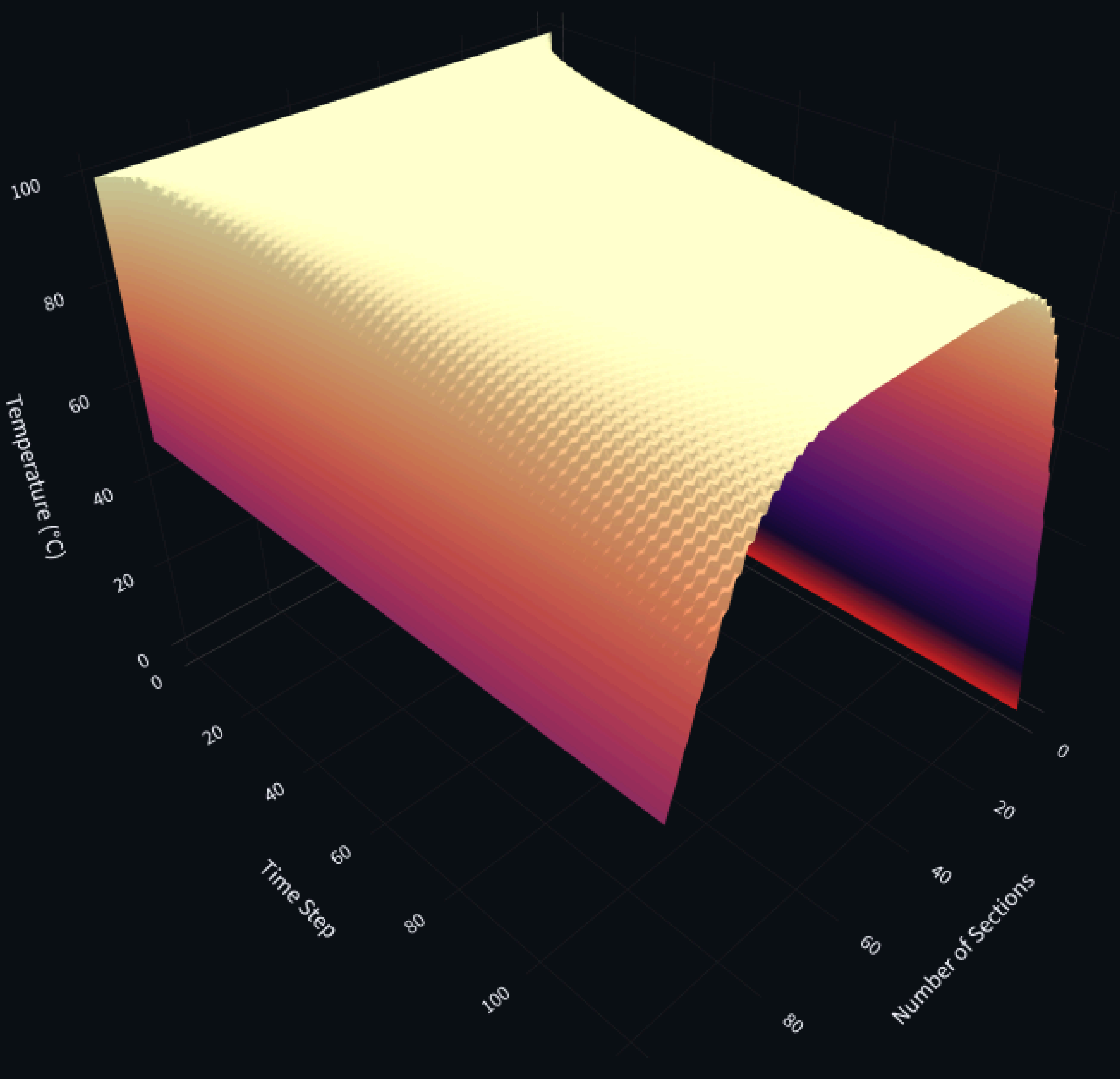
Heat-Conduction Equation with PDE Method



PARTIAL DIFFERENTIAL EQUATION

Heat transfer in this model is governed by the one-dimensional heat equation, describing the temporal evolution of temperature due to thermal diffusion, where the diffusion rate is controlled by the material's thermal diffusivity determined by its thermal conductivity, density, and specific heat capacity, and the equation is solved numerically using an explicit finite difference scheme that updates the temperature at each grid point based on neighboring values and the chosen time and space discretization.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

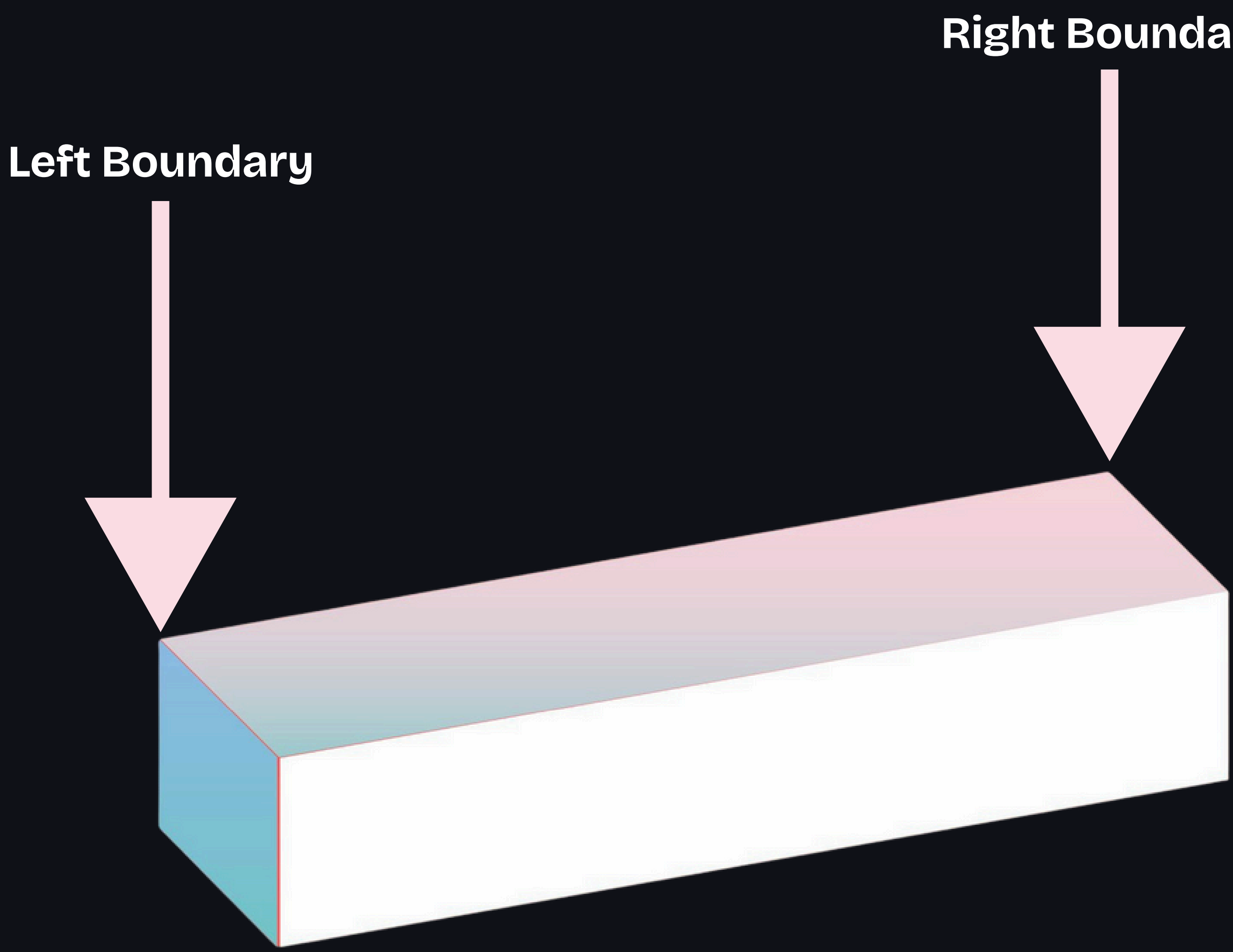
$$\alpha = \frac{k}{\rho C_p}$$

$$T_i^{n+1} = T_i^n + r\left(T_{i+1}^n - 2T_i^n + T_{i-1}^n\right), \quad r = \frac{\alpha \Delta t}{\Delta x^2}$$

T	: temperature (K or °C)
t	: time (s)
x	: spatial coordinate (m)
α	: thermal diffusivity (m^2/s)
k	: thermal conductivity (W/m·K)
ρ	: density (kg/m^3)
C_p	: specific heat capacity at constant pressure (J/kg·K)
T_i^n	: temperature at spatial index $i$and time step n
Δt	: time step size (s)
Δx	: spatial grid spacing (m)
r	: numerical stability parameter

BOUNDARY CONDITION

This section allows the user to define the thermal behavior at the left and right boundaries of the 1D domain. Boundary conditions control how heat enters, leaves, or is constrained at the edges of the system, and they directly influence the temperature distribution over time. The application provides two types of boundary conditions for each boundary: **Dirichlet** and **Neumann**.



DIRICHLET

When the Dirichlet boundary condition is selected, the temperature at the left boundary is fixed at a constant value specified by the input TL (°C), representing a boundary that is maintained at a constant temperature throughout the simulation, while similarly, the temperature at the right boundary is fixed at a constant value specified by the input TR (°C), indicating that the right boundary is also held at a constant temperature for the entire duration of the simulation.

DIRICHLET BOUNDARY CONDITIONS

$$T_0^{\,n} = T_{\text{left}}\,, \quad T_N^{\,n} = T_{\text{right}}$$

$$T_i^{\,n+1} = T_i^{\,n} + r\left(T_{i+1}^{\,n} - 2T_i^{\,n} + T_{i-1}^{\,n}\right), \quad 1 \le i \le N-1$$

T_i^{n+1} **: temperature at spatial index i at the next time step $n+1$**

T_i^n **: temperature at spatial index i at the current time step n**

T_{i-1}^n **: temperature at spatial index $i-1$ at the current time step n**

T_{i+1}^n **: temperature at spatial index $i+1$ at the current time step n**

NEUMAN

For the Neumann boundary condition, the temperature gradient is prescribed to represent the heat flux at the boundary, where the boundary temperature is inferred from the adjacent interior grid point; a zero gradient indicates an insulated boundary, a positive gradient signifies heat entering the domain, and a negative gradient represents heat leaving the domain, with the same interpretation applied consistently at both the left and right boundaries of the domain.

NEUMAN BOUNDARY CONDITIONS

Left Boundary (Flux at x=0)

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = q_L \quad \Rightarrow \quad T_0^{n+1} = T_1^{n+1} - q_L\,\Delta x$$

Right Boundary (Flux at x=L)

$$\left.\frac{\partial T}{\partial x}\right|_{x=L} = q_R \quad \Rightarrow \quad T_N^{n+1} = T_{N-1}^{n+1} + q_R\,\Delta x$$

q_L : heat flux at the left boundary

q_R : heat flux at the right boundary

T_0^{n+1} : temperature at the left boundary node

T_N^{n+1} : temperature at the right boundary node

T_1^{n+1} : temperature at the first interior grid point

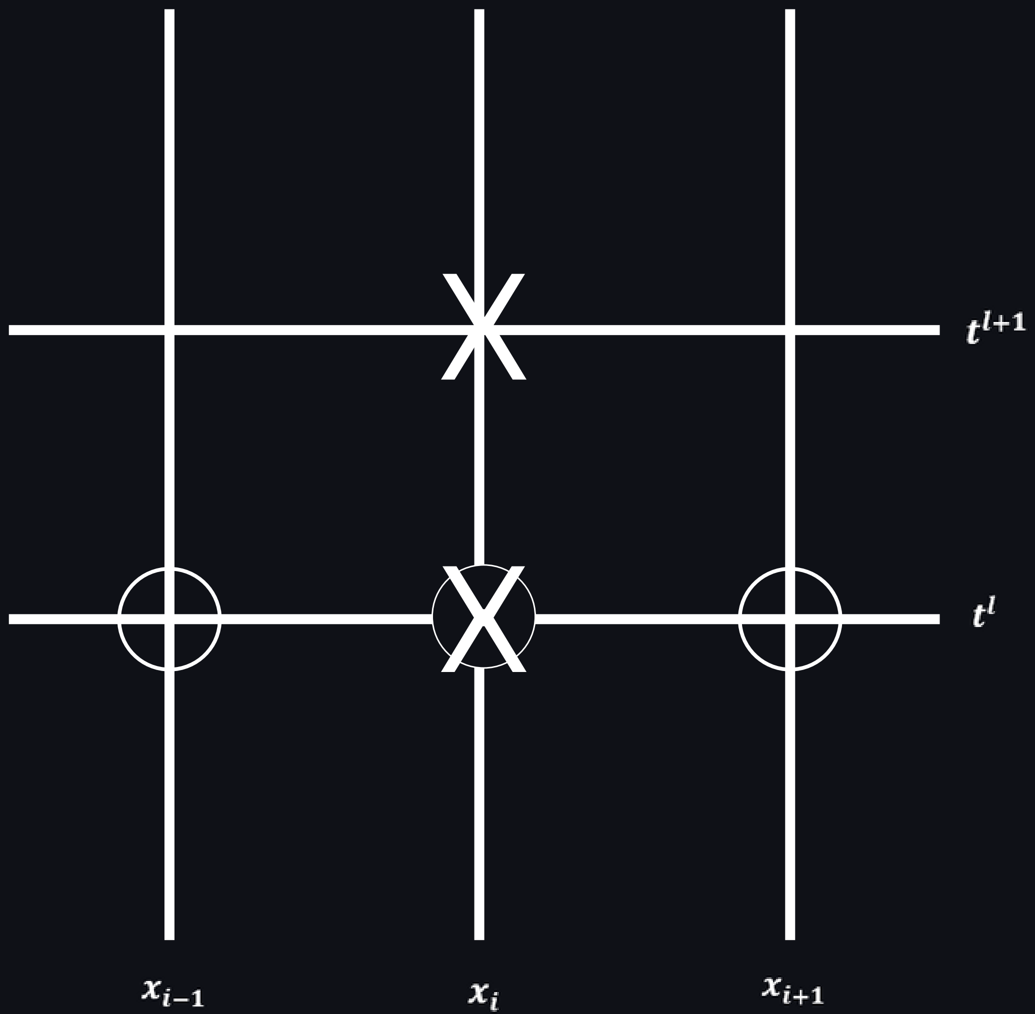
T_{N-1}^{n+1} : temperature at the last interior grid point

Δx : spatial grid spacing

SOLUTION (EXPLICIT)

EXPLICIT

The explicit method computes the temperature at the next time step directly from known values at the current time step. This approach is simple to implement and computationally efficient, but it requires the stability condition to be satisfied. If the time step is too large, the solution may become unstable.



llustration: Computational molecule for the explicit scheme

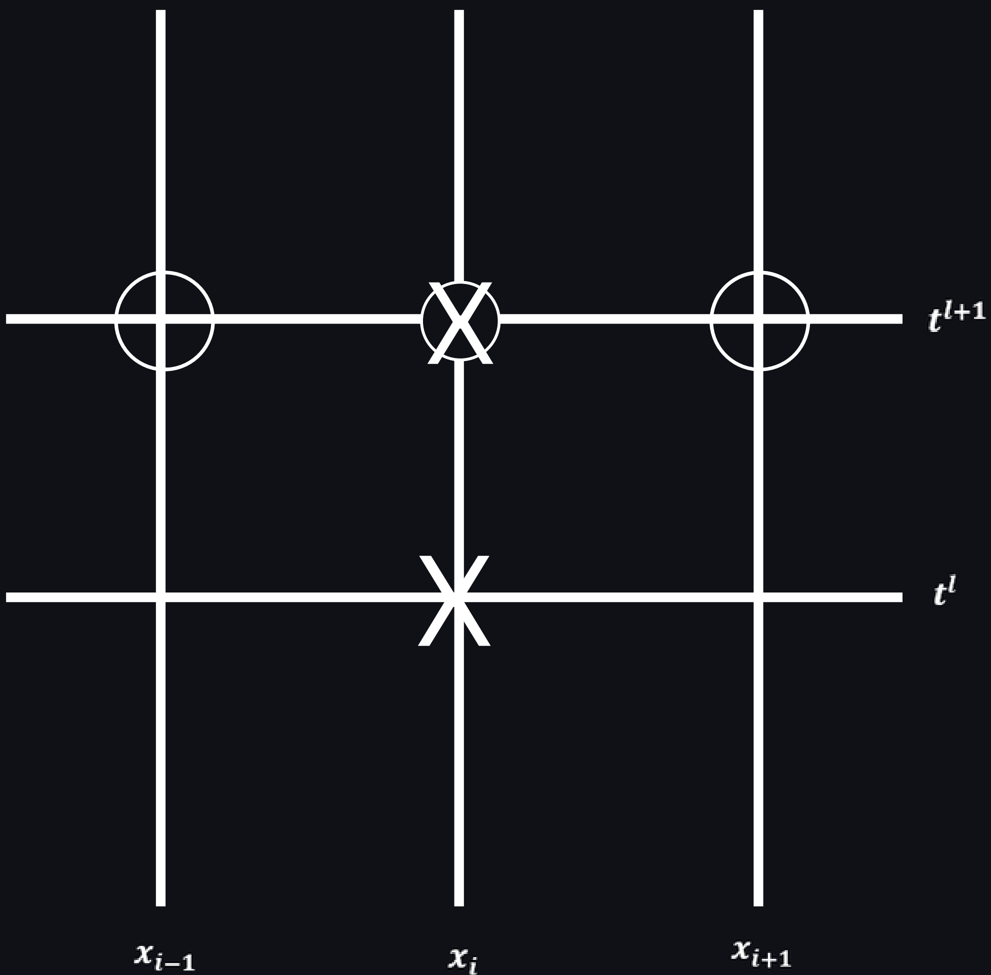
Grid point involved in space difference

Grid point involved in time difference

SOLUTION IMPLICIT)

IMPLICIT

The implicit method computes the temperature at the next time step by solving a system of equations involving future values. This approach is more stable and allows larger time steps, but it requires higher computational effort compared to the explicit method.alues at the current time step. This approach is simple to implement and computationally efficient, but it requires the stability condition to be satisfied. If the time step is too large, the solution may become unstable.



llustration: Computational molecule for the implicit scheme

Grid point involved in space difference

Grid point involved in time difference

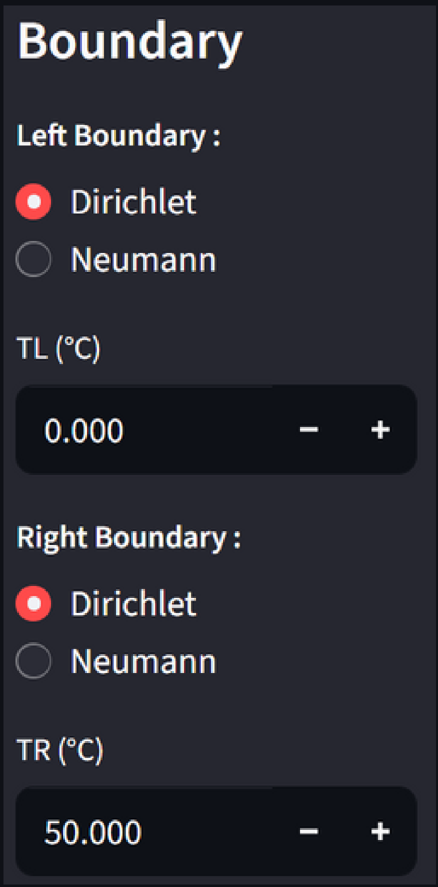
STABILITY CONDITION

The stability condition defines the numerical requirement for obtaining a stable and physically meaningful solution when using the explicit finite difference scheme, limiting the relationship between thermal diffusivity, time step size, and spatial grid spacing through the stability parameter, which controls how information propagates across the grid at each time step; if this condition is violated, the solution may become unstable and produce oscillatory or non-physical temperature values, whereas selecting a sufficiently small time step relative to the grid spacing and material diffusivity ensures smooth temperature evolution and reliable simulation results.

$$r = \frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2}$$

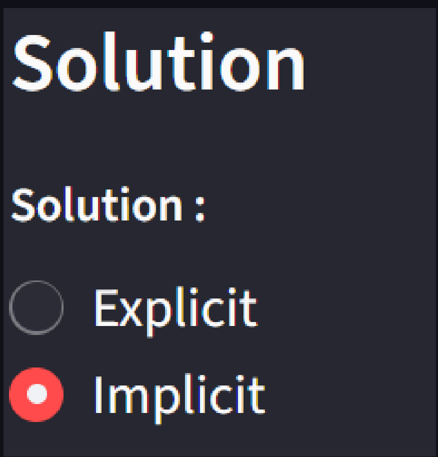
r	: numerical stability parameter
α	: thermal diffusivity of the material (m^2/s)
Δt	: time step size (s)
Δx	: spatial grid spacing (m)
$r \leq \frac{1}{2}$	: stability condition required to ensure a stable explicit finite difference solution

TUTORIAL



1

“The first step is to select the type of boundary condition for the left and right boundaries, either Dirichlet or Neumann (°C for Dirichlet, °C/m for Neumann).”



2

“The second step is to select the solution method to be used, either explicit or implicit.”

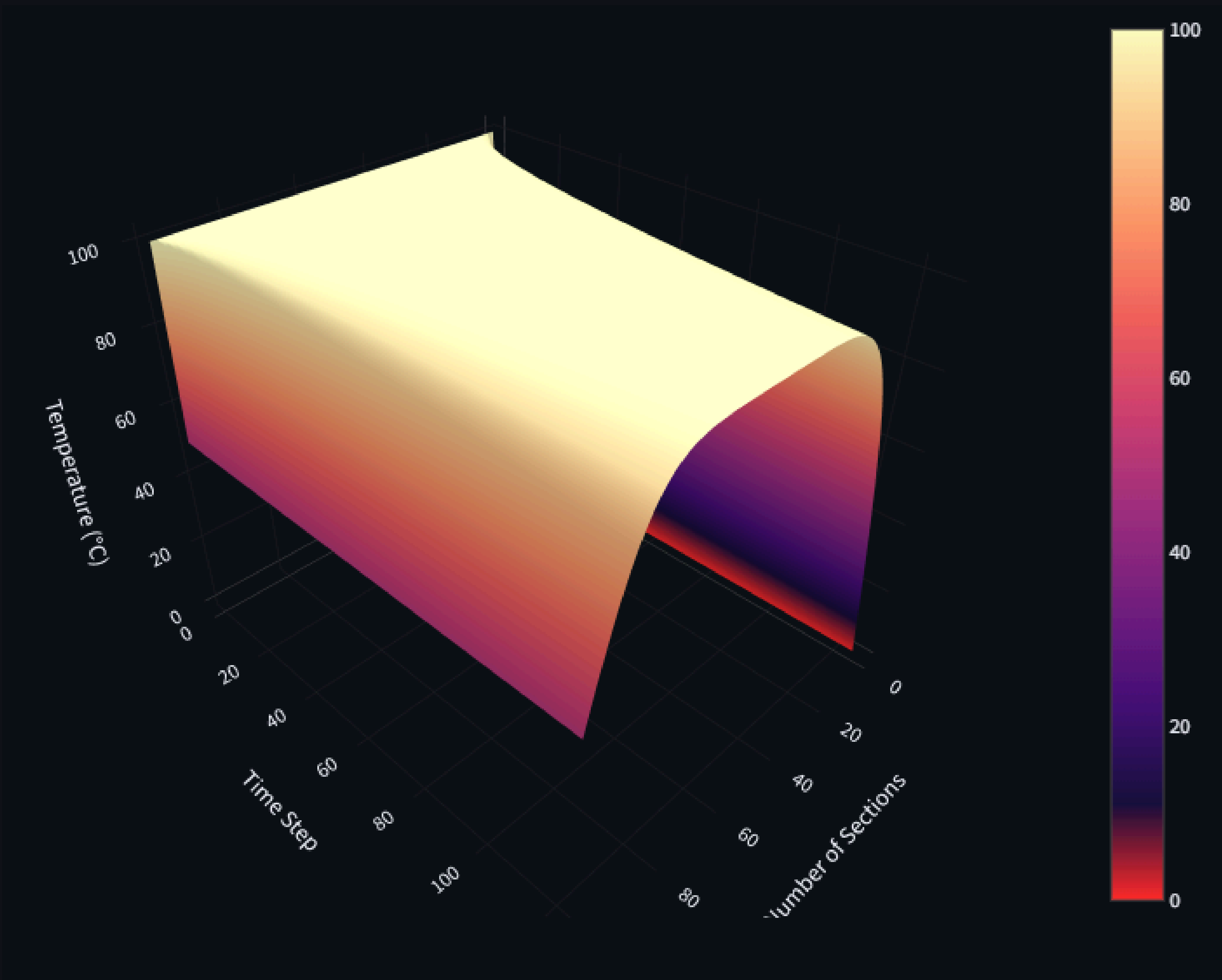
3

"The final step is to input the property values of the 1D component and the simulation time. The list of required properties is as follows:"

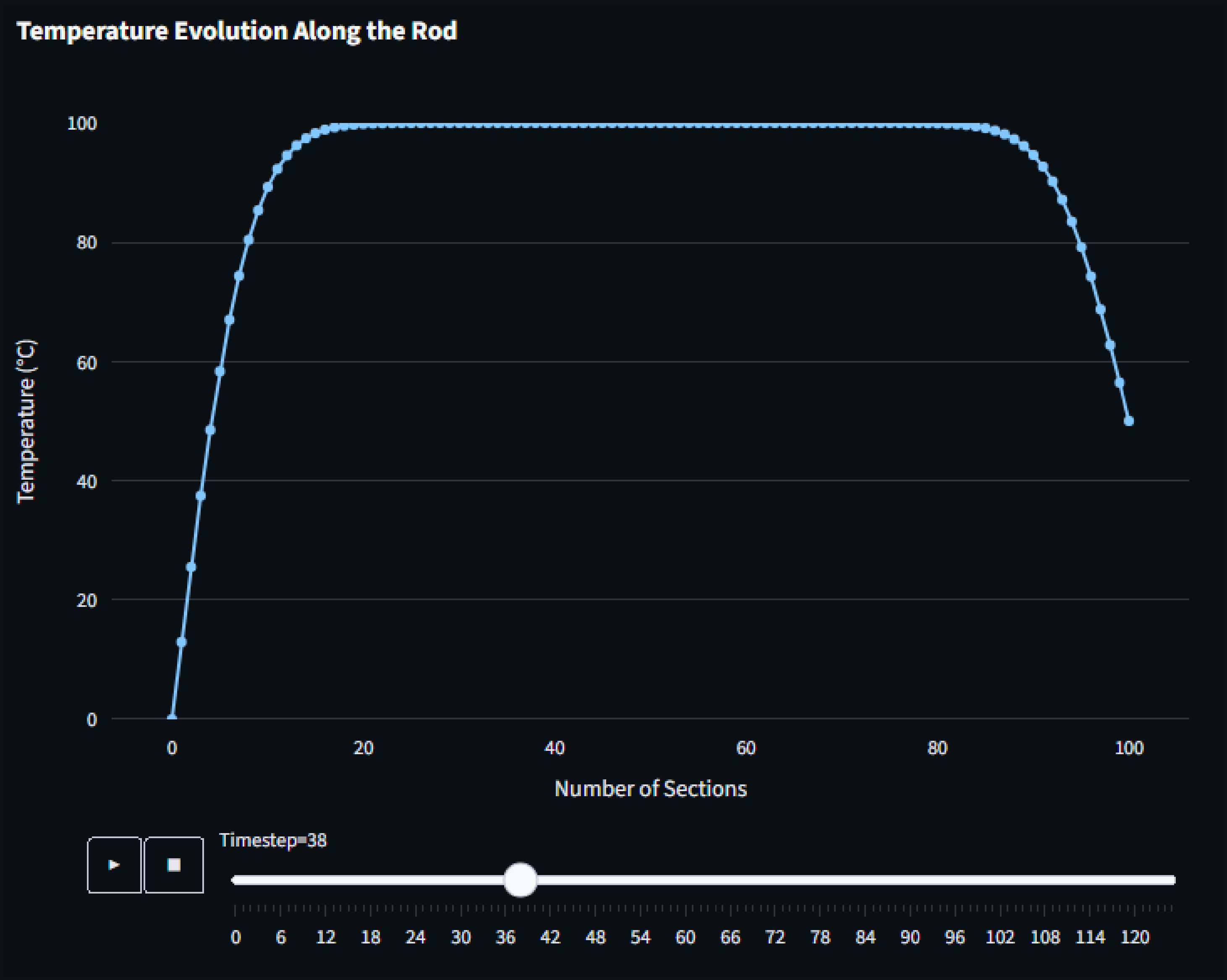
- T0 = Initial temperature (°C)
- K = Thermal conductivity (W/(m·K))
- cp = Specific heat capacity (J/(kg·K))
- Density = Material density (kg/m³)
- Alpha = Thermal diffusivity (m²/s)
- Length = Domain length (m)
- Grid number = Total number of grid points
- Time = Total simulation duration (s)
- dt = Time step size (s)

RESULT

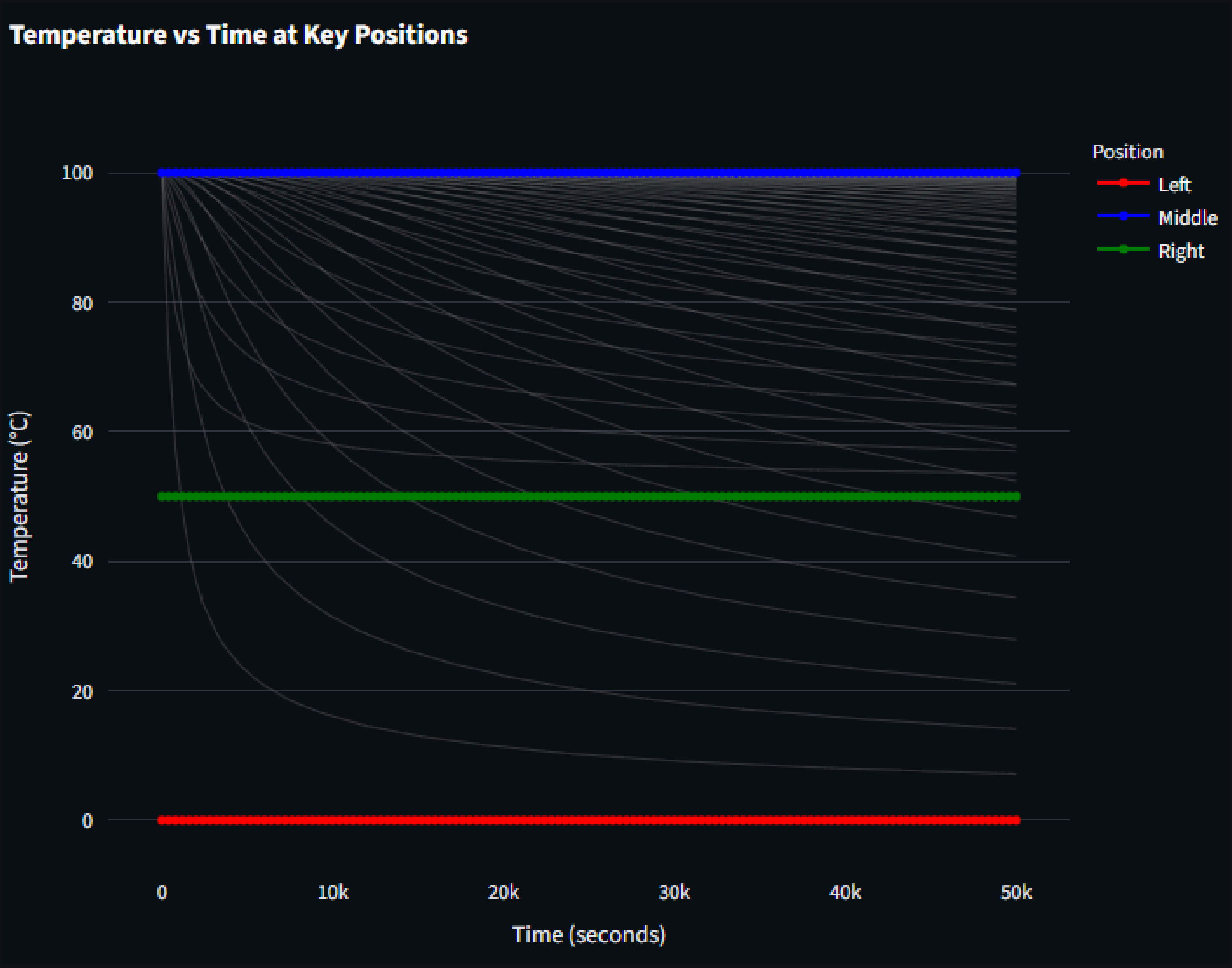
3D VISUALIZE TEMPERATURE DISTRIBUTION



TEMPERATURE DISTRIBUTION PLOTS FOR EACH SECTION SHOWING TEMPERATURE CHANGES AT EVERY TIME STEP



TEMPERATURE VS TIME AT KEY POSITIONS



2D VISUALIZATION OF TEMPERATURE DISTRIBUTION WITH TEMPERATURE CHANGES AT EACH TIME STEP

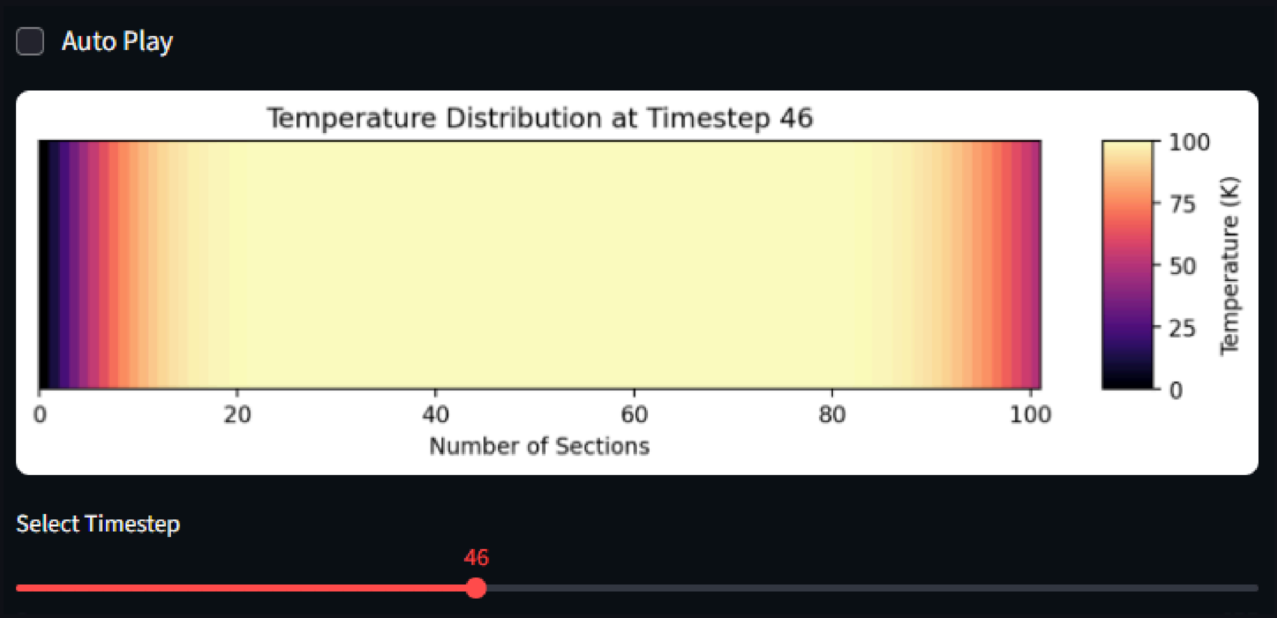


TABLE OF COMPUTED TEMPERATURE DISTRIBUTION FOR EACH SECTION AND TIME STEP

0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	100	100	100	100	100	100	100	100	100	100	100	100	100
0	73.02	2.7208	8.0361	9.4701	99.857	9.9614	9.9896	9.9972	9.9992	9.9998	9.9999	100	100
0	7.5051	84.349	94.648	8.2513	99.446	9.8284	9.9477	9.9843	9.9953	9.9986	9.9996	9.9999	100
0	7.8829	6.7497	0.5984	6.4441	8.7183	9.5545	9.8495	9.9503	9.9839	9.9949	9.9984	9.9995	9.9998
0	1.4807	0.3093	6.4102	4.2341	7.6905	9.1156	9.6733	9.8828	99.959	9.9859	9.9953	9.9984	9.9995
0	6.9517	64.953	82.369	1.7985	6.4146	8.5091	9.4049	9.7705	99.914	9.9685	9.9887	99.996	9.9986
0	3.5799	0.4962	8.6047	89.274	4.9541	7.7482	9.0384	9.6045	9.8424	9.9389	9.9769	9.9914	9.9969
0	0.9636	6.7567	5.1597	6.7544	93.37	96.855	8.5751	9.3793	9.7387	9.8932	9.9575	9.9835	9.9937
0	8.8653	3.5838	72.032	4.2991	1.7138	5.8554	8.0213	99.093	9.5988	99.828	9.9283	9.9708	9.9884
0	7.1374	50.86	9.1997	81.942	90.026	4.7745	7.3871	8.7462	9.4199	9.7402	99.887	9.9521	9.9802

THANK YOU

Temperature (°C)
100
80
60
40
20
0
Time Step
0
20
40
60
80
100
Number of Sections
0
20
40
60
80