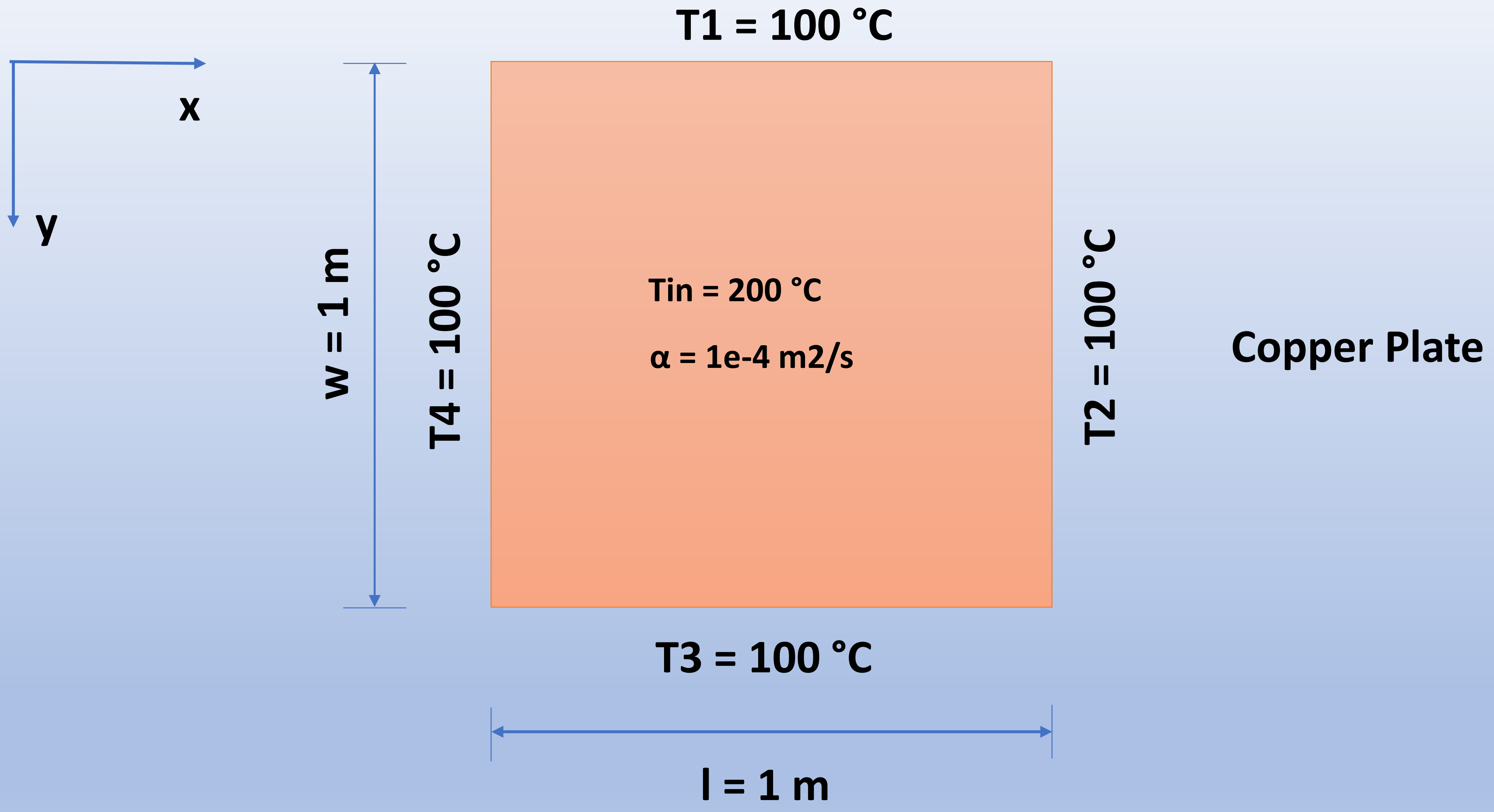
Solve 2D Transient Heat Conduction Problem in Cartesian Coordinates Using FTCS Finite Difference Method

Objectives

- Present a simple 2D Transient heat conduction problem with no heat generation
- We discretize the domain into 4 x 4 grid
- We consider 2 time steps and solve the problem using Forward time centered-space (FTCS) finite difference method
- Vary grid spacings and time steps and resolve the problem and obtain temperatures inside the domain



2D Transient Heat Conduction Problem

General Heat Conduction Equation in 3D Cartesian Coordinates

- $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{g}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t}$ (1) ; T(Temp) = T(x,y,z,t); x,y,z – spatial coordinates, t - time
- $\alpha\ - Thermal\ diffusivity, \frac{m2}{s}$
- $\alpha\ = k\ /(\rho * c)$
- k – thermal conductivity of the material , W/(m K)
- ρ - density of the material, kg/m3
- c – specific heat capacity of the material, J/(kg K)
- g – volumetric rate of internal heat generation, W/(m3)

Assumption of material thermal conductivity

- kx, ky, kz does not vary along each of x, y or z axis (homogeneous)
- kx = ky = kz = k (isotropic)

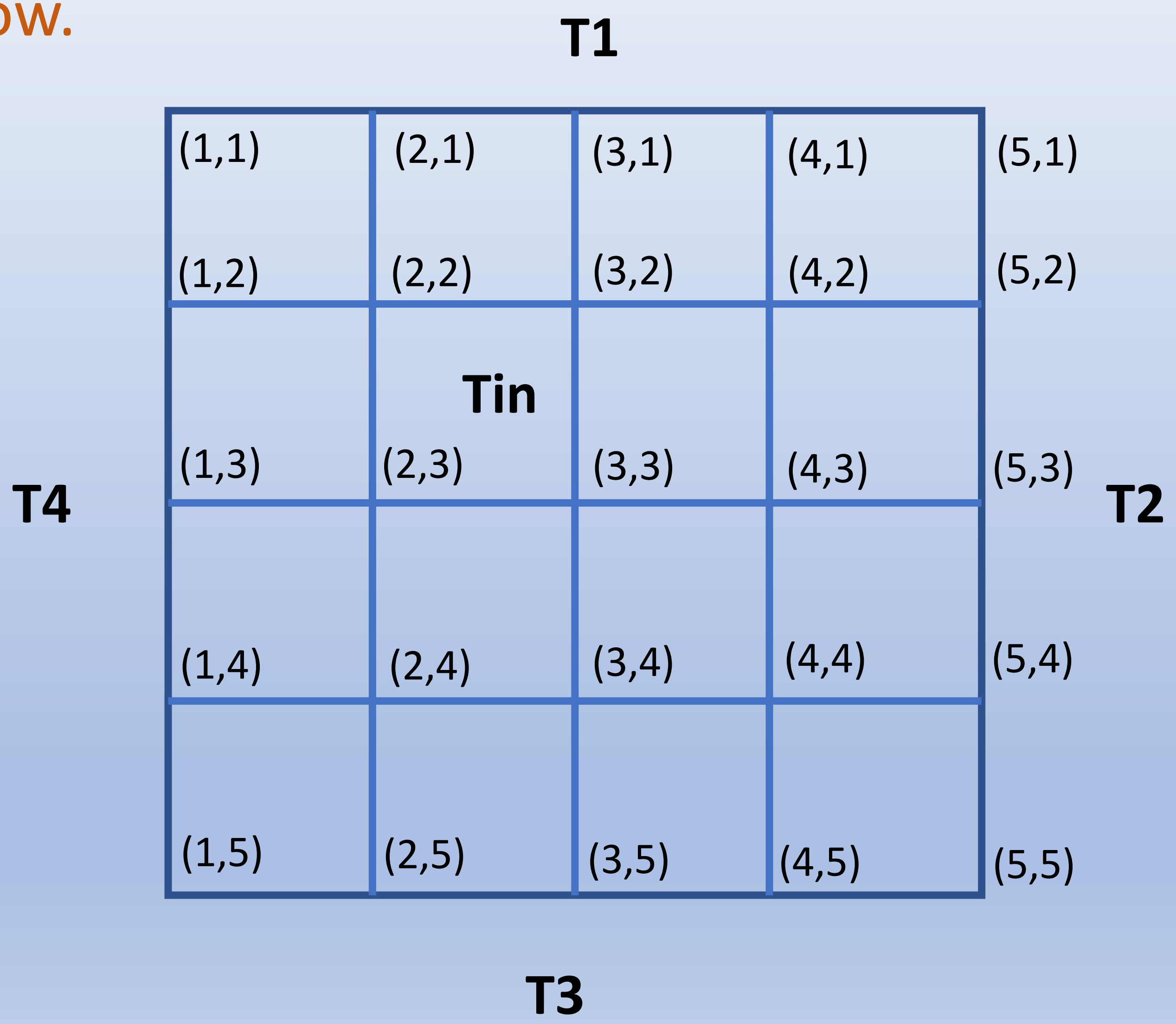
For 2D transient heat conduction with no heat generation, equation (1) reduces to a simpler form

- We assume temperature does not vary significantly along the z direction when compared with x, y directions.
- $g = 0$ (no heat generation);
- $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \left(\frac{1}{\alpha}\right)\left(\frac{\partial T}{\partial t}\right)$ .... (2); T = T(x,y,t);
- $\left(\frac{\partial T}{\partial t}\right) = \alpha * \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$; T = T(x,y,t); t >=0;
- IC: T(x,y,t=0) = Tin;
- BCs: T(x,y = 0,t) = T1; T(x=l,y,t) = T2; T(x,y=w,t) = T3; T(x=0,y,t) = T4 (3)
- l = length of the domain in x direction; w = length of the domain in y direction.
- To obtain T, we need to solve the above PDE.
- We will utilize Finite Difference Method to solve the above PDE.
- To do so, we need to replace the partial derivatives with finite difference approximations
- We replace the time derivative with first order forward difference and the space derivatives with second order centered-difference approximations.

- $\left(\frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t}\right) = \alpha * \left[\left(\frac{T_{i-1,j}^{n} - 2 * T_{i,j}^{n} + T_{i+1,j}^{n}}{\Delta x^2}\right) + \left(\frac{T_{i,j-1}^{n} - 2 * T_{i,j}^{n} + T_{i,j+1}^{n}}{\Delta y^2}\right)\right];$
- For simplicity, let Δx = Δy
- $T_{i,j}^{n+1} = T_{i,j}^{n} + \left(\frac{\alpha * \Delta t}{\Delta x^2}\right) * \left[T_{i-1,j}^{n} + T_{i+1,j}^{n} - 4 * T_{i,j}^{n} + T_{i,j-1}^{n} + T_{i,j+1}^{n}\right]$…(4)
- Let d = $\left(\frac{\alpha * \Delta t}{\Delta x^2}\right)$; where d – diffusion number, we then get
- $T_{i,j}^{n+1} = T_{i,j}^{n} + d * \left[T_{i-1,j}^{n} + T_{i+1,j}^{n} - 4 * T_{i,j}^{n} + T_{i,j-1}^{n} + T_{i,j+1}^{n}\right]$…(5)
- Equation (5) is the finite difference approximation of the original equation we were trying to solve.
- Here i represents the node location along the x direction, j represents the node location along the y direction and n represents the time step.

- The above approximation is called Forward-Time Centered-Space (FTCS) method.
- This is an explicit method. Hence, temperatures $T_{i,j}$'s at future times (n+1) can be directly obtained based on $T_{i,j}$'s at present times as shown in equation (5)
- Explicit methods are conditionally stable.
- The stability criteria is given as d <= 0.25 (for 2D problems)
- Also, time step (Δt) needs to be small based on the accuracy desired
- The error is $O(\Delta t) + O(\Delta x^2) + O(\Delta y^2)$

- Now let us discretize the 2D domain into a 4 x 4 grid equally spaced, as shown below.



- We have 25 nodes in total
- Temperatures are fixed at the boundary nodes as shown
- Our interest is on the (9) interior nodes (2,2)……(4,4)
- Let us apply equation (5) on the (9) interior nodes
- $T_{i,j}^{n+1} = T_{i,j}^{n} + d * \left[T_{i-1,j}^{n} + T_{i+1,j}^{n} - 4 * T_{i,j}^{n} + T_{i,j-1}^{n} + T_{i,j+1}^{n}\right]$ …(5)
- $T_{2,2}^{n+1} = T_{2,2}^{n} + d * \left[T_{1,2}^{n} + T_{3,2}^{n} - 4 * T_{2,2}^{n} + T_{2,1}^{n} + T_{2,3}^{n}\right]$; for (i = 2, j = 2)
- $T_{3,2}^{n+1} = T_{3,2}^{n} + d * \left[T_{2,2}^{n} + T_{4,2}^{n} - 4 * T_{3,2}^{n} + T_{3,1}^{n} + T_{3,3}^{n}\right]$; for (i = 3, j = 2)
- $T_{4,2}^{n+1} = T_{4,2}^{n} + d * \left[T_{3,2}^{n} + T_{5,2}^{n} - 4 * T_{4,2}^{n} + T_{4,1}^{n} + T_{4,3}^{n}\right]$; for (i = 4, j = 2)
- $T_{2,3}^{n+1} = T_{2,3}^{n} + d * \left[T_{1,3}^{n} + T_{3,3}^{n} - 4 * T_{2,3}^{n} + T_{2,2}^{n} + T_{2,4}^{n}\right]$; for (i = 2, j = 3)
- $T_{3,3}^{n+1} = T_{3,3}^{n} + d * \left[T_{2,3}^{n} + T_{4,3}^{n} - 4 * T_{3,3}^{n} + T_{3,2}^{n} + T_{3,4}^{n}\right]$; for (i = 3, j = 3)
- $T_{4,3}^{n+1} = T_{4,3}^{n} + d * \left[T_{3,3}^{n} + T_{5,3}^{n} - 4 * T_{4,3}^{n} + T_{4,2}^{n} + T_{4,4}^{n}\right]$; for (i = 4, j = 3)
- $T_{2,4}^{n+1} = T_{2,4}^{n} + d * \left[T_{1,4}^{n} + T_{3,4}^{n} - 4 * T_{2,4}^{n} + T_{2,3}^{n} + T_{2,5}^{n}\right]$; for (i = 2, j = 4)
- $T_{3,4}^{n+1} = T_{3,4}^{n} + d * \left[T_{2,4}^{n} + T_{4,4}^{n} - 4 * T_{3,4}^{n} + T_{3,3}^{n} + T_{3,5}^{n}\right]$; for (i = 3, j = 4)
- $T_{4,4}^{n+1} = T_{4,4}^{n} + d * \left[T_{3,4}^{n} + T_{5,4}^{n} - 4 * T_{4,4}^{n} + T_{4,3}^{n} + T_{4,5}^{n}\right]$; for (i = 4, j = 4)

- For the first time step, let n = 0, we get
- $T_{2,2}^{1} = T_{2,2}^{0} + d * [T_{1,2}^{0} + T_{3,2}^{0} - 4 * T_{2,2}^{0} + T_{2,1}^{0} + T_{2,3}^{0}];$ for (i = 2, j = 2)
- $T_{3,2}^{1} = T_{3,2}^{0} + d * [T_{2,2}^{0} + T_{4,2}^{0} - 4 * T_{3,2}^{0} + T_{3,1}^{0} + T_{3,3}^{0}];$ for (i = 3, j = 2)
- $T_{4,2}^{1} = T_{4,2}^{0} + d * [T_{3,2}^{0} + T_{5,2}^{0} - 4 * T_{4,2}^{0} + T_{4,1}^{0} + T_{4,3}^{0}];$ for (i = 4, j = 2)
- $T_{2,3}^{1} = T_{2,3}^{0} + d * [T_{1,3}^{0} + T_{3,3}^{0} - 4 * T_{2,3}^{0} + T_{2,2}^{0} + T_{2,4}^{0}];$ for (i = 2, j = 3)
- $T_{3,3}^{1} = T_{3,3}^{0} + d * [T_{2,3}^{0} + T_{4,3}^{0} - 4 * T_{3,3}^{0} + T_{3,2}^{0} + T_{3,4}^{0}];$ for (i = 3, j = 3)
- $T_{4,3}^{1} = T_{4,3}^{0} + d * [T_{3,3}^{0} + T_{5,3}^{0} - 4 * T_{4,3}^{0} + T_{4,2}^{0} + T_{4,4}^{0}];$ for (i = 4, j = 3)
- $T_{2,4}^{1} = T_{2,4}^{0} + d * [T_{1,4}^{0} + T_{3,4}^{0} - 4 * T_{2,4}^{0} + T_{2,3}^{0} + T_{2,5}^{0}];$ for (i = 2, j = 4)
- $T_{3,4}^{1} = T_{3,4}^{0} + d * [T_{2,4}^{0} + T_{4,4}^{0} - 4 * T_{3,4}^{0} + T_{3,3}^{0} + T_{3,5}^{0}];$ for (i = 3, j = 4)
- $T_{4,4}^{1} = T_{4,4}^{0} + d * [T_{3,4}^{0} + T_{5,4}^{0} - 4 * T_{4,4}^{0} + T_{4,3}^{0} + T_{4,5}^{0}];$ for (i = 4, j = 4)

- Let Δt = 100 s (Note: higher value chosen for illustration purposes; For accuracy purposes use smaller value);
- Δx = l/nx = 1/4 = 0.25 m; Δy = Δx = 0.25 m.
- Then d = $(\frac{\alpha * \Delta t}{\Delta x^2})$ = $\left(\frac{1 \text{ x } 10^{-4} \text{ x } 100}{0.25^2}\right)$ = 0.16 <= 0.25 (stability criteria met)
- Substituting the diffusion no. d and initial and boundary temperature values into the previous set of equations, we get,
- $T_{2,2}^{1}$ = 200 + 0.16 * $[100 + 200 - 4 * 200 + 100 + 200]$ = 168 °C; for (i = 2, j = 2)
- $T_{3,2}^{1}$ = 200 + 0.16 * $[200 + 200 - 4 * 200 + 100 + 200]$ = 184 °C; for (i = 3, j = 2)
- $T_{4,2}^{1}$ = 200 + 0.16 * $[200 + 100 - 4 * 200 + 100 + 200]$ = 168 °C; for (i = 4, j = 2)
- $T_{2,3}^{1}$ = 200 + 0.16 * $[100 + 200 - 4 * 200 + 200 + 200]$ = 184 °C; for (i = 2, j = 3)
- $T_{3,3}^{1}$ = 200 + 0.16 * $[200 + 200 - 4 * 200 + 200 + 200]$ = 200 °C; for (i = 3, j = 3)
- $T_{4,3}^{1}$ = 200 + 0.16 * $[200 + 100 - 4 * 200 + 200 + 200]$ = 184 °C; for (i = 4, j = 3)
- $T_{2,4}^{1}$ = 200 + 0.16 * $[100 + 200 - 4 * 200 + 200 + 100]$ = 168 °C; for (i = 2, j = 4)
- $T_{3,4}^{1}$ = 200 + 0.16 * $[200 + 200 - 4 * 200 + 200 + 100]$ = 184 °C; for (i = 3, j = 4)
- $T_{4,4}^{1}$ = 200 + 0.16 * $[200 + 100 - 4 * 200 + 200 + 100]$ = 168 °C; for (i = 4, j = 4)

- For the second time step, let n = 1, we get
- $T_{2,2}^{2} = T_{2,2}^{1} + d * \left[T_{1,2}^{1} + T_{3,2}^{1} - 4 * T_{2,2}^{1} + T_{2,1}^{1} + T_{2,3}^{1}\right]$; for (i = 2, j = 2)
- $T_{3,2}^{2} = T_{3,2}^{1} + d * \left[T_{2,2}^{1} + T_{4,2}^{1} - 4 * T_{3,2}^{1} + T_{3,1}^{1} + T_{3,3}^{1}\right]$; for (i = 3, j = 2)
- $T_{4,2}^{2} = T_{4,2}^{1} + d * \left[T_{3,2}^{1} + T_{5,2}^{1} - 4 * T_{4,2}^{1} + T_{4,1}^{1} + T_{4,3}^{1}\right]$; for (i = 4, j = 2)
- $T_{2,3}^{2} = T_{2,3}^{1} + d * \left[T_{1,3}^{1} + T_{3,3}^{1} - 4 * T_{2,3}^{1} + T_{2,2}^{1} + T_{2,4}^{1}\right]$; for (i = 2, j = 3)
- $T_{3,3}^{2} = T_{3,3}^{1} + d * \left[T_{2,3}^{1} + T_{4,3}^{1} - 4 * T_{3,3}^{1} + T_{3,2}^{1} + T_{3,4}^{1}\right]$; for (i = 3, j = 3)
- $T_{4,3}^{2} = T_{4,3}^{1} + d * \left[T_{3,3}^{1} + T_{5,3}^{1} - 4 * T_{4,3}^{1} + T_{4,2}^{1} + T_{4,4}^{1}\right]$; for (i = 4, j = 3)
- $T_{2,4}^{2} = T_{2,4}^{1} + d * \left[T_{1,4}^{1} + T_{3,4}^{1} - 4 * T_{2,4}^{1} + T_{2,3}^{1} + T_{2,5}^{1}\right]$; for (i = 2, j = 4)
- $T_{3,4}^{2} = T_{3,4}^{1} + d * \left[T_{2,4}^{1} + T_{4,4}^{1} - 4 * T_{3,4}^{1} + T_{3,3}^{1} + T_{3,5}^{1}\right]$; for (i = 3, j = 4)
- $T_{4,4}^{2} = T_{4,4}^{1} + d * \left[T_{3,4}^{1} + T_{5,4}^{1} - 4 * T_{4,4}^{1} + T_{4,3}^{1} + T_{4,5}^{1}\right]$; for (i = 4, j = 4)

- Substituting the diffusion no. d and previous and boundary temperature values into the previous set of equations, we get,

- $T^1_{2,2}$ = 168 + 0.16 * $[100 + 184 - 4 * 168 + 100 + 184\,]$ = 151.36 °C; for (i = 2, j = 2)
- $T^1_{3,2}$ = 184 + 0.16 * $[168 + 168 - 4 * 184 + 100 + 200\,]$ = 168 °C; for (i = 3, j = 2)
- $T^1_{4,2}$ = 168 + 0.16 * $[184 + 100 - 4 * 168 + 100 + 184\,]$ = 151.36 °C; for (i = 4, j = 2)
- $T^1_{2,3}$ = 184 + 0.16 * $[100 + 200 - 4 * 184 + 168 + 168\,]$ = 168 °C; for (i = 2, j = 3)
- $T^1_{3,3}$ = 200 + 0.16 * $[184 + 184 - 4 * 200 + 184 + 184\,]$ = 189.76°C; for (i = 3, j = 3)
- $T^1_{4,3}$ = 184 + 0.16 * $[200 + 100 - 4 * 184 + 168 + 168\,]$ = 168 °C; for (i = 4, j = 3)
- $T^1_{2,4}$ = 168 + 0.16 * $[100 + 184 - 4 * 168 + 184 + 100\,]$ = 151.36°C; for (i = 2, j = 4)
- $T^1_{3,4}$ = 184 + 0.16 * $[168 + 168 - 4 * 184 + 200 + 100\,]$ = 168 °C; for (i = 3, j = 4)
- $T^1_{4,4}$ = 168 + 0.16 * $[184 + 100 - 4 * 168 + 184 + 100\,]$ = 151.36 °C; for (i = 4, j = 4)

- Likewise, we can find the temperatures at these interior nodes at the next time step by choosing $n = 2$ and so on
- Graphical results are presented using MATLAB for this case.
- Using MATLAB or other software, we can develop codes for a general case where the number of grid spacings and time steps can be varied as desired.
- And solutions obtained accordingly.

Summary

In this video,

- We presented a 2D transient heat conduction problem in cartesian coordinates
- The temperature at each of the 4 sides of the square copper plate is fixed.
- The initial temperature in the domain is known.
- Using FTCS Finite difference method, we discretized our domain, obtained temperatures inside the domain at various times and solved the problem
- We varied the grid spacings and time steps and presented the results using MATLAB
- In future videos, we can explore more challenging problems.