# ***TEMPERATURE DISTRIBUTION FOR 2D STEADY STATE HEAT CONDUCTION IN NUCLEAR FUEL ROD BY FINITE DIFFERENCE METHOD***

PRESENTED BY: AYAN PANJA

DEPT. OF MECHANICAL ENGINEERING

JADAVPUR UNIVERSITY

# Content

# Introduction

In this project we’ve solved 2D heat conduction equation using CFD techniques to get the solution of temperature distribution inside a nuclear rod with a coolant in it’s surroundings with specific convective boundary conditions.

The system of equation is solved by Gauss Seidel approach through a Matlab code. Here the corresponding partial equations are non dimensionalized and further discretized by finite difference schemes, central and backward difference to be precise. Then another matrix is generated which contains the x, y coordinates and values of temperature respectively. That has been further implemented in Tecplot , a post processing software.

Tecplot gives the temperature contour inside the cross section of the rod which is the objective of this experiment.

## Problem Formulation

- Assumption: I. Heat transfer is steady and two dimensional.

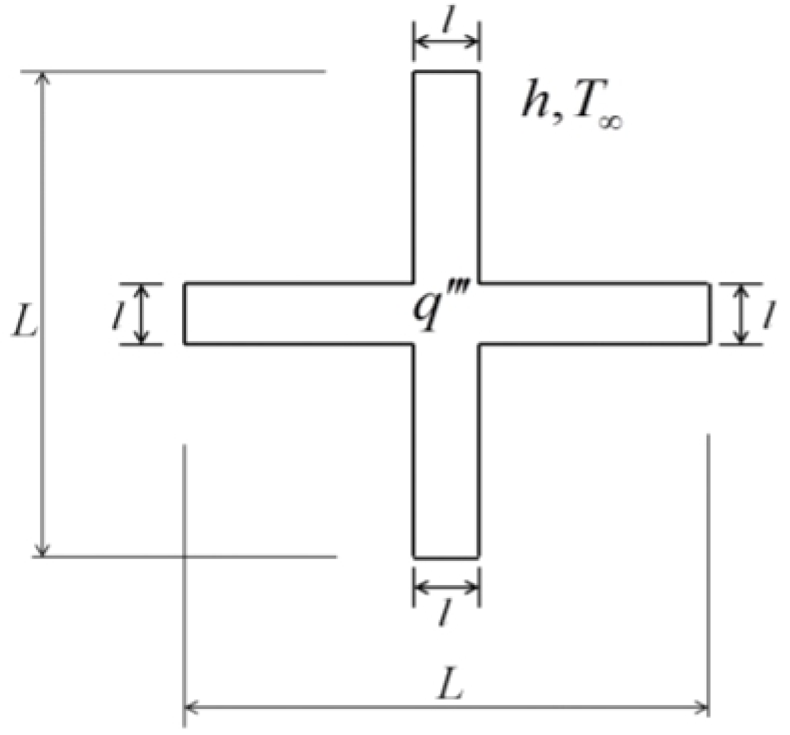
  II. Thermal conductivity is constant throughout.

  III. Heat generation is uniform,

  IV. Radiation heat transfer is negligible.

It’s a case of steady state heat conduction where heat is generated uniformly throughout the body at a uniform rate of q’’’ W/m$^3$.

The problem can be assumed to be a 2-D as the dimension of the rod is much longer to the dimension normal to the cross sectional plane; therefore end effects can be neglected.



Heat generation in a 2D Nuclear Fuel Rod

➢ <u>Consideration of symmetry:</u>

From a minute observation of the physical problem can be interpreted as geometrically and thermally symmetric. So calculation of one quarter is enough to compute distribution in whole domain.

- <u>Governing equations:</u>

  The heat energy equation for steady state heat (for constant "k") is

$$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+q'''=0$$

- <u>Boundary conditions:</u>

  BC1: at x=0, $\frac{\partial T}{\partial x}=0$

  BC2: at y=0, $\frac{\partial T}{\partial y}=0$

  BC3: Along the horizontal boundary, $-k\frac{\partial T}{\partial y}=h(T-T_\infty)$

  BC4: Along the vertical boundary, $-k\frac{\partial T}{\partial x}=h(T-T_\infty)$

- <u>Dimensionless form</u>: $\theta=\frac{T-T_\infty}{(q'''L^2/k)}$, X=$\frac{x}{L}$, Y=$\frac{y}{L}$

- <u>Non-dimensionalizing using the dimensionless variables:</u>

  $\frac{\partial^2\theta}{\partial X^2}+\frac{\partial^2\theta}{\partial Y^2}+1=0$

  BC1: at x=0, $\frac{\partial\theta}{\partial x}=0$

  BC2: at y=0, $\frac{\partial\theta}{\partial y}=0$

  BC3: Along the horizontal boundary, $\frac{\partial\theta}{\partial Y}+Bi\cdot\theta=0$

  BC4: Along the vertical boundary, $\frac{\partial\theta}{\partial X}+Bi\cdot\theta=0$.

## COMPUTATIONAL METHOD

In this problem central difference scheme is used for interior points and backward difference is used at the convection surfaces to discretize the equations. we're considering a (100x100) grid here. And the equations are discretized at any interior grid point(i,j).

- Discretized equations:

$$\frac{\partial^2\theta}{\partial X^2}+\frac{\partial^2\theta}{\partial Y^2}+1=0$$

$$=>\frac{\theta_{i+1,j}-2\theta_{i,j}+\theta_{i-1,j}}{(\Delta X)^2}+\frac{\theta_{i,j-1}-2\theta_{i,j}+\theta_{i,j+1}}{(\Delta Y)^2}+1=0$$

$$=>-\theta_{i-1,j}-\theta_{i,j-1}+4\theta_{i,j}-\theta_{i,j+1}-\theta_{i+1,j}=(\Delta X)^2 \text{ --(i)}$$

- Boundary condition along X=0:

  Using image point technique $\theta_{i-1,j}=\theta_{i+1,j}$

  Setting i=1 , we have:

$$-2\theta_{2,j}-\theta_{1,j-1}+4\theta_{1,j}---\theta_{1,j+1}=(\Delta X)^2 \text{ --(ii)}$$

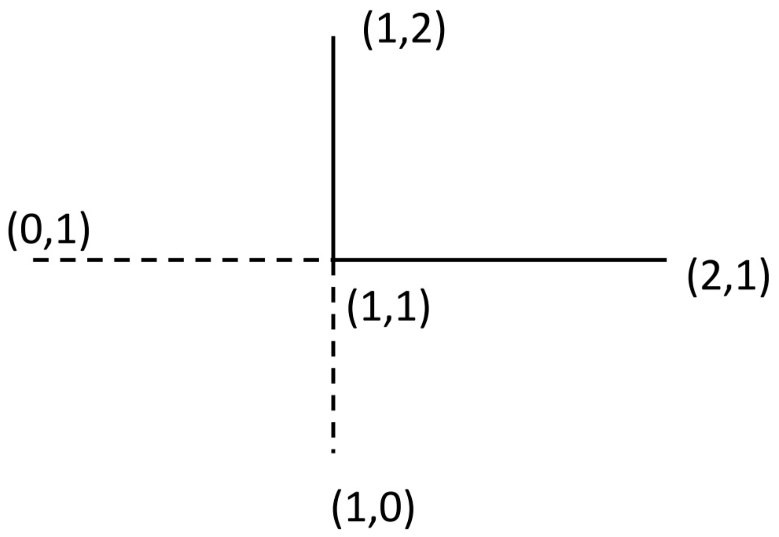
- Boundary condition along Y=0:

  Using image point technique $\theta_{i,j-1}=\theta_{i,j+1}$

  Setting j=1, we have:

$$-\theta_{i+1,1}-2\theta_{i,2}+4\theta_{i,1}-\theta_{i-1,1}=(\Delta X)^2 \text{ --(iii)}$$

➢ <u>Dealing with corner point:</u> Using image point technique…



$$\theta_{0,1} = \theta_{2,1} \ \& \ \theta_{1,0} = \theta_{1,2}$$

$$\left(\frac{\partial^2\theta}{\partial X^2}\right)_{1,1} + \left(\frac{\partial^2\theta}{\partial Y^2}\right)_{1,1} + 1 = 0$$

$$=> \frac{\theta_{2,1}-2\theta_{1,1}+\theta_{0,1}}{(\Delta X)^2} + \frac{\theta_{1,2}-2\theta_{1,1}+\theta_{1,0}}{(\Delta Y)^2} + 1 = 0$$

$$=> \frac{\theta_{2,1}-2\theta_{1,1}+\theta_{2,1}}{(\Delta X)^2} + \frac{\theta_{1,2}-2\theta_{1,1}+\theta_{1,2}}{(\Delta Y)^2} + 1 = 0$$

$$=> 2\theta_{2,1} - 4\theta_{1,1} + 2\theta_{1,2} + (\Delta X)^2 = 0 \text{ ----(iv)}$$

➢ <u>Boundary condition along horizontal boundary:</u>

$$\frac{\theta_{i,j}-\theta_{i,j-1}}{\Delta X} + Bi \cdot \theta_{i,j} = 0$$

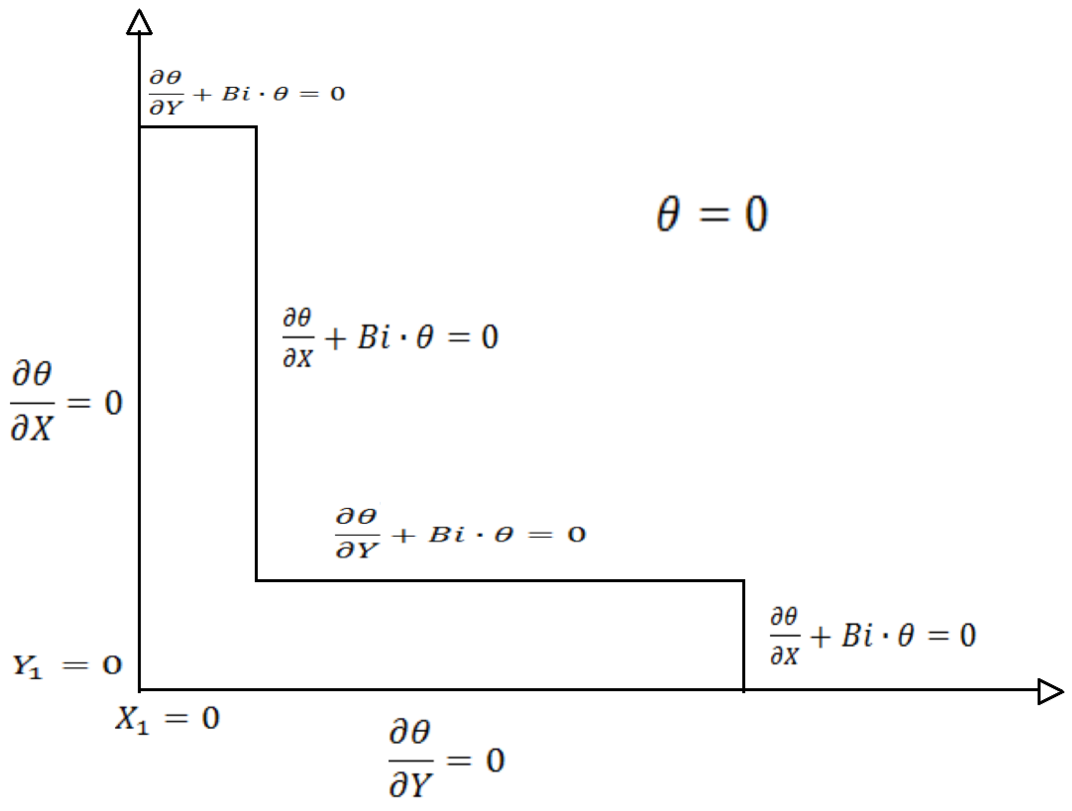
$$=> \theta_{i,j} = \frac{\theta_{i,j-1}}{1+(\Delta x)Bi} \text{ ---(v)}$$

- <u>Boundary condition along vertical boundary:</u>

$$\frac{\theta_{i,j}-\theta_{i-1,j}}{\Delta X} + Bi\cdot\theta_{i,j} = 0$$

$$=>\theta_{i,j} = \frac{\theta_{i-1,j}}{1+(\Delta x)Bi} \text{ ---(vi)}$$

- <u>Computational geometry:</u>



$\frac{\partial\theta}{\partial Y}+Bi\cdot\theta=0$
$\theta=0$
$\frac{\partial\theta}{\partial X}+Bi\cdot\theta=0$
$\frac{\partial\theta}{\partial X}=0$
$\frac{\partial\theta}{\partial Y}+Bi\cdot\theta=0$
$\frac{\partial\theta}{\partial x}+Bi\cdot\theta=0$
$Y_1=0$
$X_1=0$
$\frac{\partial\theta}{\partial Y}=0$

## Method of Solution:

Here we've six system of linear equations for which we'll be using 'Gauss-Seidel' iterative method as it'll store less amount of data than a direct method.

- Gauss-Seidel Iterative Method:

Let us consider a system of linear equations as follows

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N = c_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N = c_2$$

.

.

$$a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{NN}x_N = c_N$$

The method assumes the solution to the system of equations in general

as : $x_i = \frac{c_i - \sum_{j=1, j\neq i}^{N} a_{ij}x_j}{a_{ii}}$ for i=1,2,3,.....,N .

To start the iteration process initial guess values are assigned to all unknowns. If $x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)}$ are initial guess values, then value of $x_1$ after the first iteration,

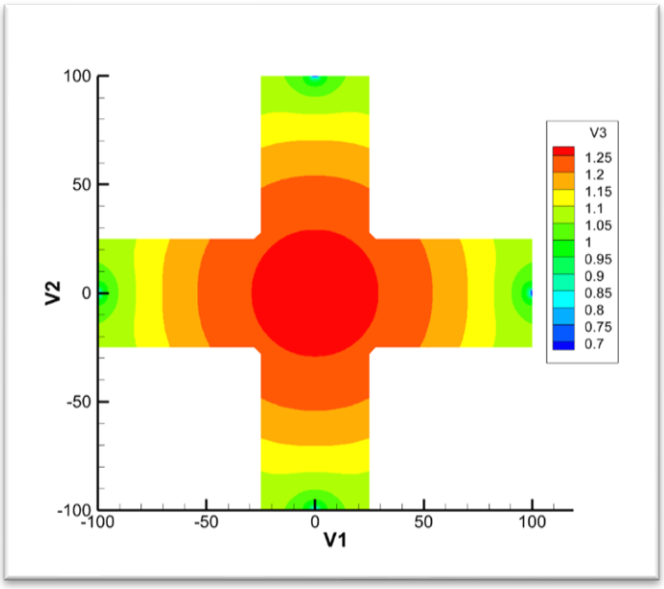
$$x_1^{(1)} = \frac{c_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - \cdots - a_{1N}x_N^{(0)}}{a_{11}}$$

The values obtained from the current iteration replaces the previously computed values hence rate of convergence increases. The values obtained in current iteration is further implemented in computing the remaining variables in the same iteration.

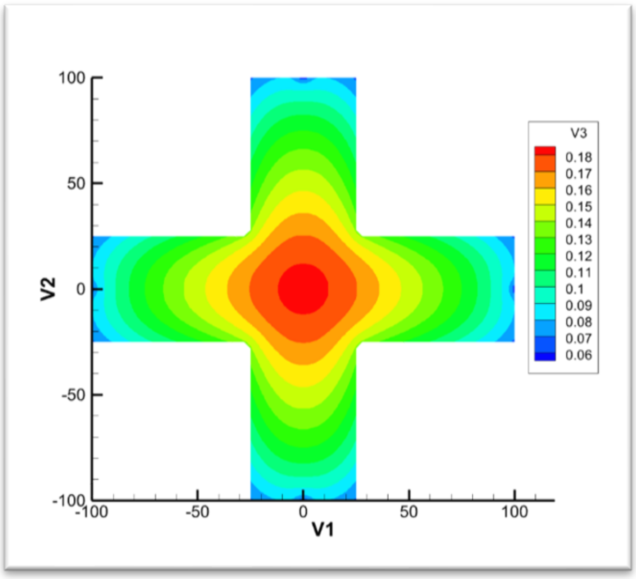
Therefore in general ,

$$x_i^{(p+1)} = \frac{c_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(p+1)} - \sum_{j=i+1}^{N} a_{ij} x_j^{(p)}}{a_{ii}}$$

## Results and Discussions

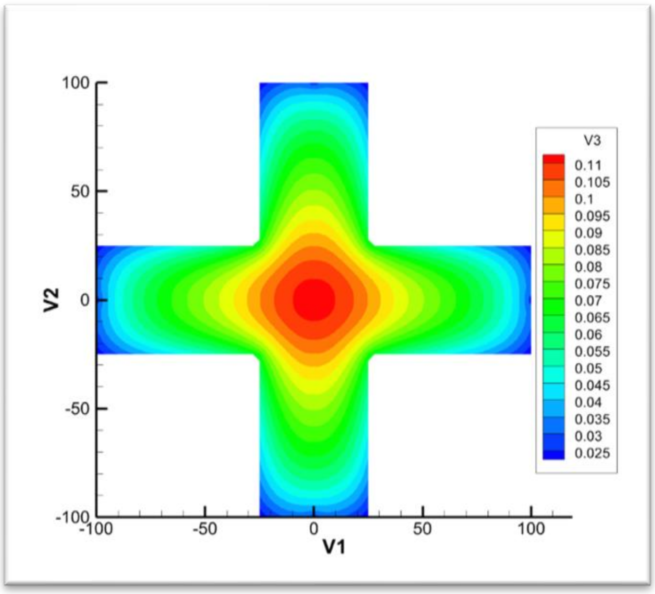


**Bi=0.1**



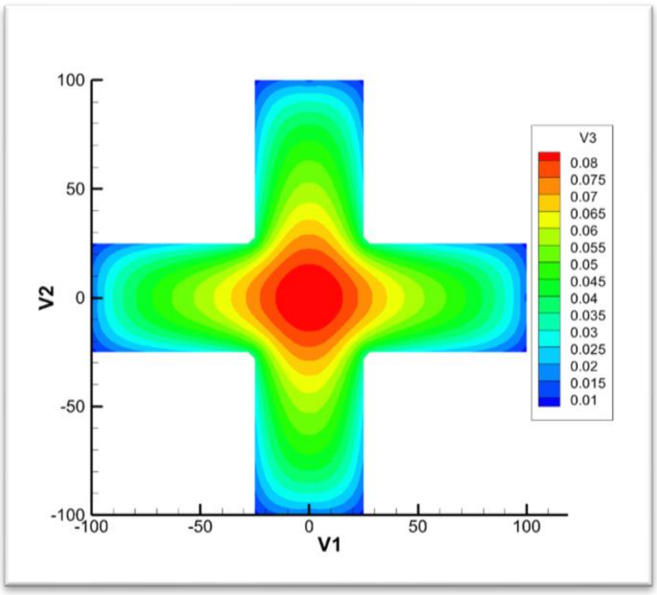
V2
100
50
0
-50
-100
V1
-100
-50
0
50
100
V3
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0.09
0.08
0.07
0.06

**Bi=2**



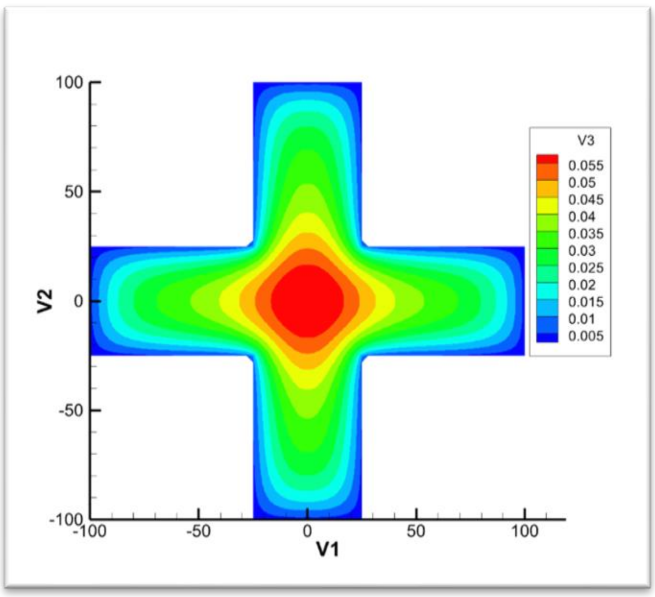
V2
100
50
0
-50
-100
V1
-100
-50
0
50
100
V3
0.11
0.105
0.1
0.095
0.09
0.085
0.08
0.075
0.07
0.065
0.06
0.055
0.05
0.045
0.04
0.035
0.03
0.025

**Bi=5**



V3
0.08
0.075
0.07
0.065
0.06
0.055
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
V1
V2

**Bi=10**



V3
0.055
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
V1
V2

**Bi=50**

## Comment:

As we know with increment in Biot no convection start dominating over conduction heat transfer as conduction resistance increases with decrease in the counterpart.

Now from the post processing results we see that the gradient in the temperature contour gradually increases with increase in Biot no which actually depicts the above mentioned fact of dominance of convection.

# Submission of Code

```
close all;
clc;
%Specifying no of grids
n=101;
nx=100;
ny=100;
Y=1;
%Grid size same for both
X=Y;
%Giving input value of Biot no.
Bi=input('enter Biot no:  ');
%Initializing the variables
T=zeros(n);
itr=0;
w=input('enter no of iteration:  ');
%Implementing Gauss-Siedel method
while itr<=w
       itr=itr+1;
       Told=T;
          %Insulated boundary conditions
             for j=52:100

T(51,j)=0.25*(X^2+T(51,j+1)+2*T(50,j)+T(51,j-1));
             end

             for i=2:50

T(i,51)=0.25*(X^2+2*T(i,52)+T(i+1,51)+T(i-1,51));
             end

             T(51,51)=0.5*(X^2+2*T(51,102)+2*T(52,51));
%calculation of interior points
 for i=2:50
 for j=52:60
 T(i,j)=0.25*(X^2+T(i,j-1)+T(i+1,j)+T(i-1,j)+T(i,j+1));
 end
 end
```

```
for i=42:50
  for j=61:100
   T(i,j)=0.25*(X^2+T(i,j-1)+T(i+1,j)+T(i-1,j)+T(i,j+1));
  end
 end
%convective boundary conditions
             for i=42:51
                 T(i,101)=T(i,100)/(1+Bi*X);
             end

             for j=61:101
                 T(60,j)=T(42,j)/(1+Bi*X);
             end

             for i=2:40
                 T(i,41)=T(i,40)/(1+Bi*X);
             end

             for j=51:61
                 T(1,j)=T(2,j)/(1+Bi*X);
             end
             Tnew=T;
            error=max(max(abs(Tnew-Told)));
 end
%Implementing symmetry to other quarters
 for i=1:51
     j=1:50;
     Tnew(i,51-j)=Tnew(i,51+j);
 end

 for i=1:50
     j=1:51;
     Tnew(51+i,j)=Tnew(51-i,j);
 end
```

```matlab
% Generating matrix for Tecplot
Tec=zeros(10201,3);
j=0;
k=0;
l=0;
m=101;
n=1;
for i=1:10201
       Tec(i,1)=j;
       Tec(i,2)=k;
       j=j+1;
       l=l+1;
       Tec(i,3)=Tnew(m,n);
       n=n+1;
       while n>51
           n=1;
       end
       while j>100
           j=0;
       end
       while l==51
           l=0; k=k+1; m=m-1;
       end

end
```