Pandit Deendayal Energy University
Computational Engineering Laboratory
Matlab Case study Report

By,

Vora Khyaat Darshan (21BME118D)

Narinder Singh Bhadwal (21BME006)

Under the Guidance of

Dr Annirudh Kulkarni

School of Technology

Pandit Deendayal Energy University

Gandhinagar-382426

March - April 2024

Case Study 1

Problem Statement :-

To Solve a problem on Heat Transfer of Natural Convection in H – Shaped Cavity.

Derivation :-

As Per the Source of paper

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{1}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho_{hnf}}\frac{\partial p}{\partial x}+\frac{\mu_{hnf}}{\rho_{hnf}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\begin{aligned}u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=&-\frac{1}{\rho_{hnf}}\frac{\partial p}{\partial y}+\frac{\mu_{hnf}}{\rho_{hnf}}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)\\&+g\beta_{T,hnf}(T-T_c)+g\beta_{c,hnf}(c-c_c)\end{aligned} \tag{3}$$

$$u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}=\alpha_{hnf}\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right) \tag{4}$$

$$u\frac{\partial c}{\partial x}+v\frac{\partial c}{\partial y}=D\left(\frac{\partial^2 c}{\partial x^2}+\frac{\partial^2 c}{\partial y^2}\right) \tag{5}$$

Introducing the following dimensionless variables:

$$U=\frac{uL}{\alpha_f}, V=\frac{vL}{\alpha_f}, \theta=\frac{(T-T_c)}{\Delta T}, C=\frac{(c-c_c)}{\Delta c}, P=\frac{pL^2}{\rho_f\alpha_f^2} \tag{6}$$

The non-dimensional shape of governing equations becomes:

$$U\frac{\partial U}{\partial X}+V\frac{\partial U}{\partial Y}=-\frac{\rho_f}{\rho_{hnf}}\frac{\partial P}{\partial X}+\mathrm{Pr}\frac{\rho_f}{\rho_{hnf}}\frac{\mu_{hnf}}{\mu_f}\left(\frac{\partial^2 U}{\partial X^2}+\frac{\partial^2 U}{\partial Y^2}\right) \tag{7}$$

$$\begin{aligned}U\frac{\partial V}{\partial X}+V\frac{\partial V}{\partial Y}=&-\frac{\rho_f}{\rho_{hnf}}\frac{\partial P}{\partial Y}+\mathrm{Pr}\frac{\rho_f}{\rho_{hnf}}\frac{\mu_{hnf}}{\mu_f}\left(\frac{\partial^2 V}{\partial X^2}+\frac{\partial^2 V}{\partial Y^2}\right)\\&+\frac{\beta_{hnf}}{\beta_f}Ra\mathrm{Pr}(\theta+NC)\end{aligned} \tag{8}$$

$$U\frac{\partial \theta}{\partial X}+V\frac{\partial \theta}{\partial Y}=\frac{(k_{hnf}/k_f)}{\left((\rho C_p)_{hnf}/(\rho C_p)_f\right)}\left(\frac{\partial^2 \theta}{\partial X^2}+\frac{\partial^2 \theta}{\partial Y^2}\right) \tag{9}$$

$$U\frac{\partial C}{\partial X}+V\frac{\partial C}{\partial Y}=\frac{1}{Le}\left(\frac{\partial^2 C}{\partial X^2}+\frac{\partial^2 C}{\partial Y^2}\right) \tag{10}$$

Where;

$$Ra=\frac{g\beta_f\Delta TL^3}{\nu_f\alpha_f}, \mathrm{Pr}=\frac{\upsilon_f}{\alpha_f}, \mathrm{Le}=\frac{\alpha_f}{D} \tag{11}$$

Boundary conditions:

$$\text{Bottom wall: } \frac{\partial\theta}{\partial Y}=0, U=0, V=0, \Psi=0, \frac{\partial C}{\partial Y}=0$$

$$\text{Bottom rib: } \frac{\partial\theta}{\partial n}=-1, U=0, V=0, \Psi=0, \frac{\partial C}{\partial n}=-1$$

$$\text{Right and left walls: } \theta=0, U=0, V=0, \Psi=0, C=0$$

$$\text{Top wall: } \frac{\partial\theta}{\partial Y}=0, U=0, V=0, \Psi=0, \frac{\partial C}{\partial Y}=0$$

$$\text{Top rib: } \frac{\partial\theta}{\partial n}=-1, U=0, V=0, \Psi=0, \frac{\partial C}{\partial n}=-1$$

$$\text{Baffle: } \frac{\partial\theta}{\partial n}=-1, U=0, V=0, \Psi=0, \frac{\partial C}{\partial n}=-1 \tag{12}$$

The target physical parameters in this paper are the local Nusselt and Sherwood numbers. These parameters are defined as:

$$Nu=\left(\frac{k_{hnf}}{k_f}\right)\left(\frac{1}{\theta}\right) \tag{13}$$

$$Sh=\left(\frac{1}{C}\right) \tag{14}$$

And the average ones are:

$$Nu_{avg}=\frac{1}{s}\int Nu.ds \tag{15}$$

$$Sh_{avg}=\frac{1}{s}\int Sh.ds \tag{16}$$

Where s is the length of the bottom rib.

K, (ρC_p), μ, $(\rho\beta)$, and ρ of hybrid nanoliquid could be represented as:

$$\frac{k_{hnf}}{k_f}=\left[(1-\lambda)\varphi\left(k_f-k_s\right)+k_s-(1-\lambda)k_f\right]\times\left[k_s-(1-\lambda)k_f+\varphi\left(k_f-k_s\right)\right]^{-1} \tag{17}$$

$$\left(\rho C_p\right)_{hnf}=\varphi_{Cu}\left(\rho C_p\right)_{Cu}+\varphi_{Al_2O_3}\left(\rho C_p\right)_{Al_2O_3}+(1-\varphi)\left(\rho C_p\right)_f \tag{18}$$

$$\mu_{hnf}=(1-\varphi)^{-2.5}\mu_f \tag{19}$$

$$(\rho\beta)_{hnf}=(1-\varphi)(\rho\beta)_f+\varphi_{Cu}(\rho\beta)_{Cu}+\varphi_{Al_2O_3}(\rho\beta)_{Al_2O_3} \tag{20}$$

$$\rho_{hnf}=(1-\varphi)\rho_f+\varphi_{Cu}\rho_{Cu}+\varphi_{Al_2O_3}\rho_{Al_2O_3} \tag{21}$$

And the correlations of entropy generation would be as:

$$S_{gen}=S_{HT}+S_{FF}+S_{Mass} \tag{22}$$

In which

$$S_{HT}=\frac{k_{hnf}}{k_f}\left(\left(\frac{\partial\theta}{\partial X}\right)^2+\left(\frac{\partial\theta}{\partial Y}\right)^2\right) \tag{23}$$

$$S_{FF}=\varphi_1\frac{\mu_{hnf}}{\mu_f}\left[2\left(\frac{\partial V}{\partial Y}\right)^2+2\left(\frac{\partial U}{\partial X}\right)^2+\left(\frac{\partial U}{\partial Y}+\frac{\partial V}{\partial X}\right)^2\right] \tag{24}$$

$$S_{Mass}=\varphi_2\left(\left(\frac{\partial C}{\partial X}\right)^2+\left(\frac{\partial C}{\partial Y}\right)^2\right)+\varphi_3\left(\frac{\partial C}{\partial X}\frac{\partial\theta}{\partial X}+\frac{\partial C}{\partial Y}\frac{\partial\theta}{\partial Y}\right) \tag{25}$$

And the Bejan number is:

$$Be=\frac{S_{HT}+S_{Mass}}{S_{gen}} \tag{26}$$

Equations (Our Source (Handwritten))

Case Study -1 Equations

* The derivation ~~the~~ the equations we need :-

1) Governing equations :-

$\nabla \cdot V = 0$

Its represents conservation of Mass.

2) Navier Stokes equation :-

$$\rho \left(\frac{\partial v}{\partial t} + V \cdot \nabla v \right)$$

$$= -\nabla p + \mu \nabla^2 v + \rho g.$$

It's represents the Conservation Momentum

3) Energy equations :-

$$\rho c_p \left(\frac{\partial T}{\partial t} + V \cdot \nabla T \right) = k \nabla^2 T$$

It's represents the Conservation of energy.

* Assumptions and Simplifications :-

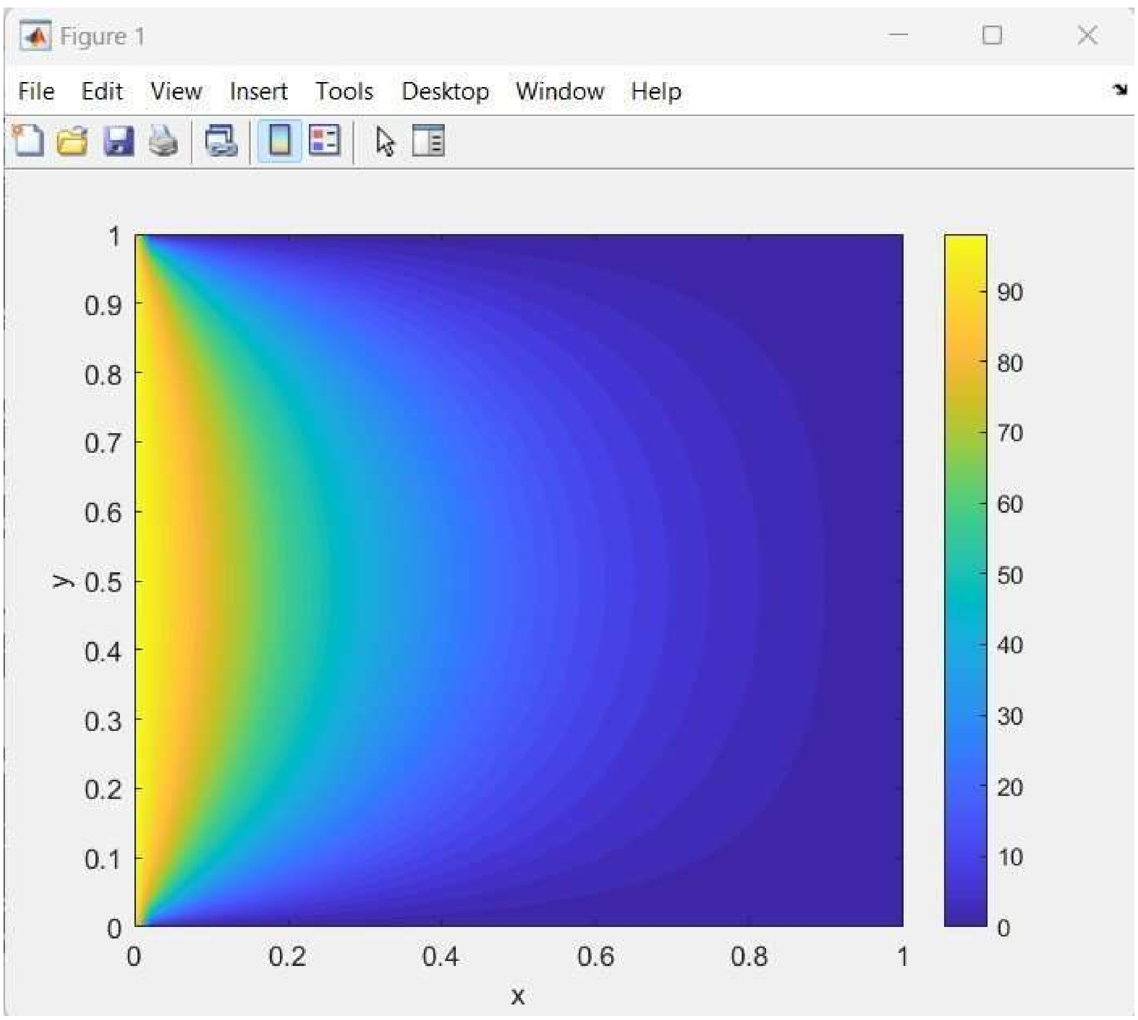
- Steady State Condition : $\frac{\partial}{\partial t} = 0$
- Incompressible flow : $\nabla \cdot V = 0$
- Boussinesq approximation (density variation with temperature) : $\rho = \rho_0 (1 - \beta (T - T_0))$
- Negligible viscous dissipation : Neglecting the viscous term $\mu \nabla^2 v$
- Constant fluid properties : ρ, μ, k and C_p are constant within the Domain.

Code

```
title('Temperature Distribution in H-Shaped Cavity');
% Parameters
L = 1; % Length of cavity
W = 1; % Width of cavity
H = 0.1; % Height of cavity
T_hot = 100; % Temperature of hot wall
T_cold = 0; % Temperature of cold walls
nx = 51; % Number of grid points in x-direction
ny = 51; % Number of grid points in y-direction
dx = L/(nx-1); % Grid spacing in x-direction
dy = W/(ny-1); % Grid spacing in y-direction
max_iter = 1000; % Maximum number of iterations
tolerance = 1e-5; % Convergence tolerance

% Initialization
T = ones(nx, ny) * T_cold; % Initialize temperature array

% Main loop (iterate until convergence)
for iter = 1:max_iter
    % Boundary conditions
    T(:,1) = T_cold; % Left cold wall
    T(:,end) = T_cold; % Right cold wall
    T(1,:) = T_hot; % Hot wall

    % Compute temperature field using simple averaging
    T_new = T;
    for i = 2:nx-1
        for j = 2:ny-1
            % Average of neighboring points
            T_new(i,j) = 0.25*(T(i+1,j) + T(i-1,j) + T(i,j+1) + T(i,j-1));
        end
    end

    % Check for convergence
    if max(abs(T_new(:) - T(:))) < tolerance
        disp(['Converged at iteration ', num2str(iter)]);
        break;
    end
```

```
    % Update temperature array
    T = T_new;
end

% Plotting the results
[X, Y] = meshgrid(linspace(0,L,nx), linspace(0,W,ny));
contourf(X, Y, T', 50, 'LineColor', 'none');
colorbar;
xlabel('x');
ylabel('y');
```

Results :-



Case Study 2

Problem Statement:-

To Update the code on the problem on Heat Transfer of Natural Convection in H – Shaped Cavity.

Derivations: -

Case Study 2 Equations

* To derive the equations :-

⇒ Governing equations :-

Start with the continuity equation :-

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

By using Navier-Stokes equations (momentum equation):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + g_x$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + g_y$$

Then,

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

Code :-

```
% Parameters
L = 1;           % Length of cavity
W = 1;           % Width of cavity
H = 0.1;         % Height of cavity
T_hot = 100;     % Temperature of hot wall
T_cold = 0;      % Temperature of cold walls
nx = 51;         % Number of grid points in x-direction
ny = 51;         % Number of grid points in y-direction
dx = L/(nx-1);   % Grid spacing in x-direction
dy = W/(ny-1);   % Grid spacing in y-direction
max_iter = 100; % Maximum number of iterations
tolerance = 1e-5;% Convergence tolerance
alpha = 0.1;     % Thermal diffusivity

% Initialize temperature array
T = ones(nx, ny) * T_cold;

% Set up VideoWriter object

writerObj = VideoWriter('temperature_evolution.avi'); % Specify the file name
writerObj.FrameRate = 10; % Set the frame rate

% Open the video writer object
open(writerObj);

% Create a figure for animation
figure;

% Main loop (iterate until convergence)
for iter = 1:max_iter
    % Boundary conditions
    T(:,1) = T_cold;            % Left cold wall
    T(:,end) = T_cold;          % Right cold wall
    T(1,:) = T_hot;             % Hot wall
    T(ceil(end/2-round(H/(2*dy))):floor(end/2+round(H/(2*dy))), end) = T_cold; % Middle cold wall

    % Compute temperature field using finite difference method
    T_new = T;
    for i = 2:nx-1
        for j = 2:ny-1
            % Finite difference equation
            T_new(i,j) = T(i,j) + alpha * ( (T(i+1,j) - 2*T(i,j) + T(i-1,j))/dx^2 + (T(i,j+1) - 2*T(i,j) + T(i,j-1))/dy^2 );
        end
    end

    % Check for convergence
    if max(abs(T_new(:) - T(:))) < tolerance
        disp(['Converged at iteration ', num2str(iter)]);
        break;
    end

    % Update temperature array
    T = T_new;

    % Plot the current temperature distribution in 3D
    [X, Y] = meshgrid(linspace(0,L,nx), linspace(0,W,ny));
    surf(X, Y, T', 'EdgeColor', 'none');
    colorbar;
    xlabel('x');
    title(['Iteration: ', num2str(iter)]);
    axis([0 L 0 W 0 100]); % Set axis limits for consistency

    % Capture the frame
    frame = getframe(gcf);

    % Write the frame to the video file
    writeVideo(writerObj, frame);
end

% Close the video writer object
close(writerObj);
```

Results :-

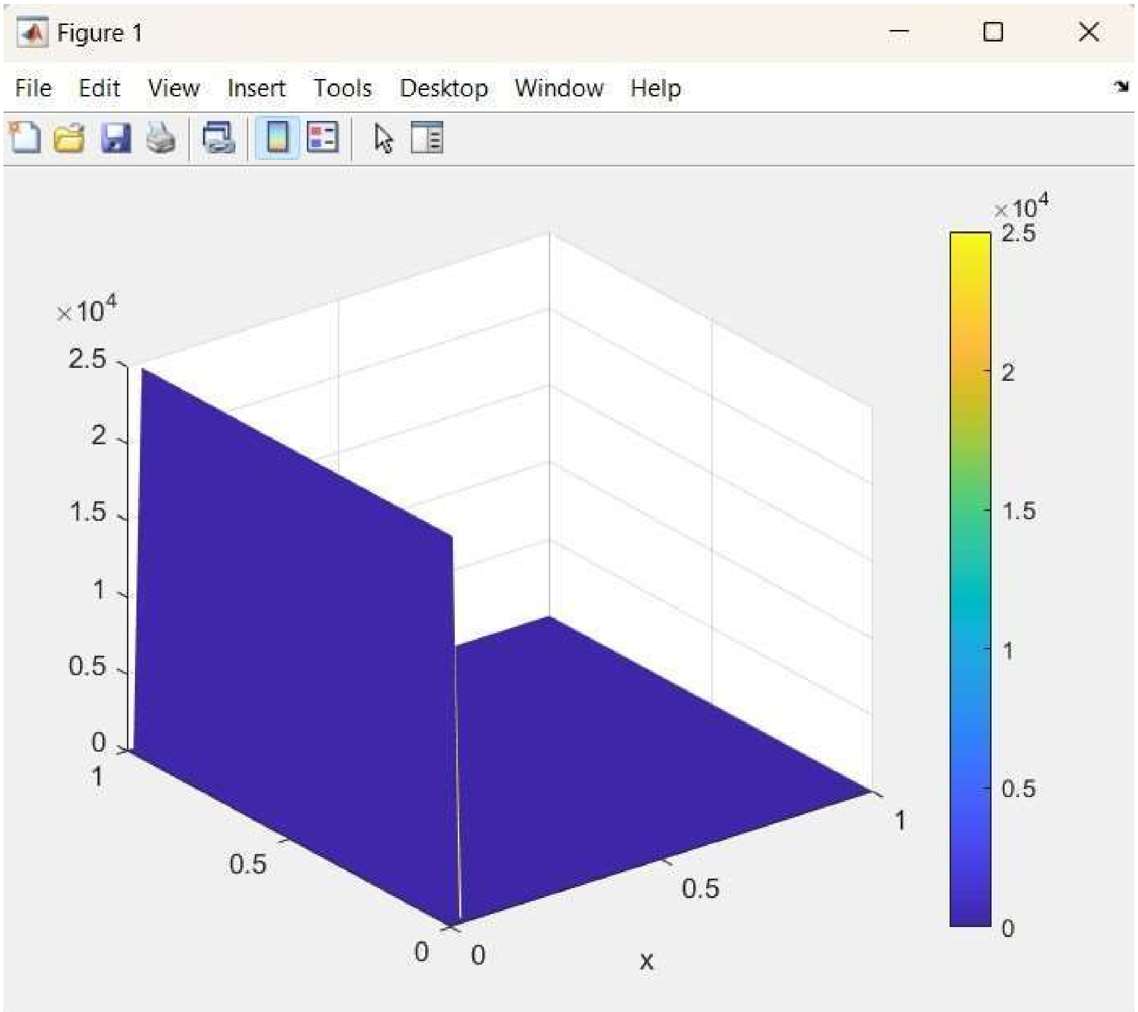


Figure 1
File Edit View Insert Tools Desktop Window Help
×10^4
2.5
2
1.5
1
0.5
0
1
0.5
0
0
0.5
1
x
×10^4
2.5
2
1.5
1
0.5
0