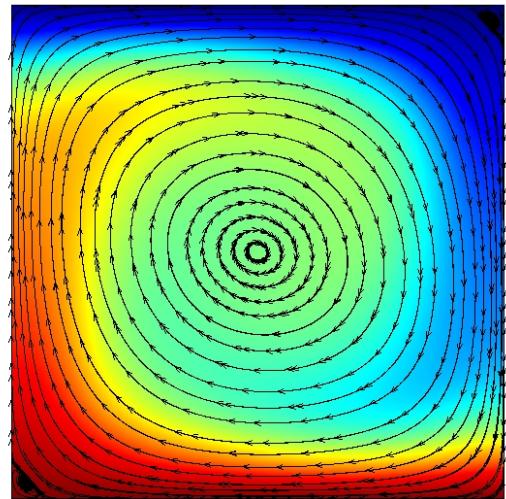
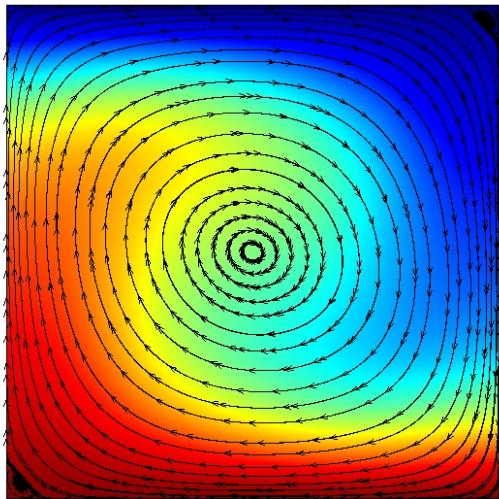


NUMERICAL SIMULATION OF DOUBLE DIFFUSIVE CONVECTION



ASHUTOSH BEHERA

RITAM MAJUMDAR

Contents

1	Introduction	2
2	Problem Description	2
2.1	Boundary Conditions	3
2.1.1	Traction free boundary condition	3
2.1.2	Symmetry boundary condition	3
2.1.3	No-slip boundary condition	3
3	Numerical Method	4
3.1	Method of relaxation	10
4	Visualisation of data	11
5	Mesh Convergence	12
6	Results and Validation	13
7	Appendix-I : MATLAB Program linked with the simulation	14
8	Appendix-II : A note on post processing and visualisation of data and the relevant software	19
9	Appendix-III : Code analysis and flow chart development	20
10	Appendix-IV : Some variations of the original problem	23
10.1	Double diffusive convection in a triangular cavity	23
10.2	Double diffusive convection with effect of a conductive wall	25
10.3	Double diffusive convection with variable wall temperature	27
11	Appendix-V : An alternative method of solving convection diffusion problems - The stream function and vorticity approach	29
11.1	Numerical Method	30
11.2	Boundary Condition	31
11.3	Mesh Convergence	32
11.4	Results and Validation	32
12	Appendix VI : Use of Mapped Grids	33

1 Introduction

Double diffusive convection is a phenomenon that describes a form of convection driven by density gradients, which have different diffusion rates. Double diffusive convection phenomenon is widely seen in process industries, where the interplay between thermal and solutal (mass) buoyancy forces play a crucial role in governing the outcome. Convection in fluids is driven by density variations within them under the influence of gravity. These density variations may be caused by gradients in the fluid composition or temperature differences (through thermal expansion). Thermal and compositional gradients can often diffuse with time, reducing their ability to drive the convection and requiring that gradients in other regions of the flow exist in order for convection to continue. A typical example of double-diffusive convection is in oceanography, where heat and salt concentrations exist with different gradients and diffuse at differing rates. An effect that affects both of these variables is the input of cold freshwater from an iceberg. Double diffusive convection is vital in understanding the evolution of several systems with multiple causes for density variations. These include convection in the Earth's oceans (as mentioned above), in magma chambers, and in the sun (where heat and helium diffuse at differing rates). Sediment can also be thought of as having a slow Brownian diffusion rate compared to salt or heat, so double-diffusive convection is significant, below sediment laden rivers in lakes and the ocean.

2 Problem Description

A square cavity is involved in the current problem, with upper and lower walls maintained at constant temperature and concentration. Sidewalls are insulated both thermally and solutally. Flow inside the cavity is a buoyancy-driven flow, governed by temperature and concentration gradients.

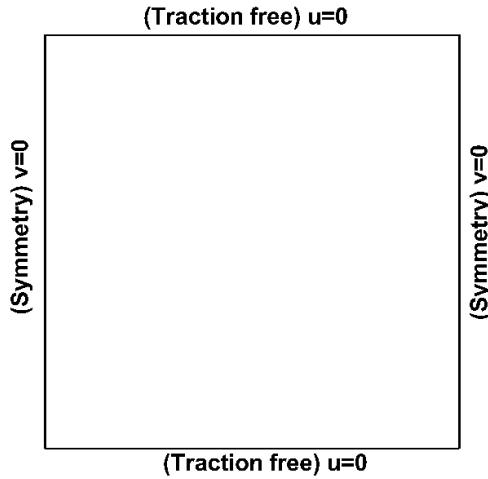


Figure 1: Schematic of problem description

The governing equations of the flow are given, in non-dimensional form as :

$$\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} = 0 \quad (1)$$

$$\frac{1}{Pr} \left(u' \frac{\partial u'}{\partial x'} + v' \frac{\partial u'}{\partial y'} \right) = -\frac{\partial p'}{\partial x'} + \left(\frac{\partial^2 u'}{\partial x'^2} + \frac{\partial^2 u'}{\partial y'^2} \right) \quad (2)$$

$$\frac{1}{Pr} \left(u' \frac{\partial v'}{\partial x'} + v' \frac{\partial v'}{\partial y'} \right) - Ra_T T' - Ra_S C' = -\frac{\partial p'}{\partial y'} + \left(\frac{\partial^2 v'}{\partial x'^2} + \frac{\partial^2 v'}{\partial y'^2} \right) \quad (3)$$

$$u' \frac{\partial T'}{\partial x'} + v' \frac{\partial T'}{\partial y'} = \left(\frac{\partial^2 T'}{\partial x'^2} + \frac{\partial^2 T'}{\partial y'^2} \right) \quad (4)$$

$$u' \frac{\partial C'}{\partial x'} + v' \frac{\partial C'}{\partial y'} = \tau \left(\frac{\partial^2 C'}{\partial x'^2} + \frac{\partial^2 C'}{\partial y'^2} \right) \quad (5)$$

where, $u' = u/U_\infty$, $v' = v/U_\infty$, $p' = P/0.5\rho U_\infty^2$, $T' = (T - T_L) / (T_H - T_L)$ and $C' = (C - C_L) / (C_H - C_L)$, are the non dimensional velocities, pressure, temperature and concentration respectively.

2.1 Boundary Conditions

Boundary and initial conditions are an essential part of solving any system of ODE or PDE. Moreover, in situations like these, where no analytical solution exists or is relevant, boundary conditions play a crucial role in the convergence of the iterations. Due to the elliptic nature of the Navier-Stokes equations and the advection-diffusion equations for temperature and concentration, all four boundaries of the domain demand different boundary conditions. Three kinds of boundary conditions on the momentum equations were studied in this numerical simulation, i.e., traction free, symmetry, and no-slip, elaborated in a later sub-section. These conditions were imposed on various sides of the domain in 3 cases :

Case	Top and bottom walls	side walls
1	traction free ($v = 0$)	symmetry ($u = 0$)
2	traction free ($v = 0$)	no slip ($u = v = 0$)
3	no slip ($u = v = 0$)	symmetry ($u = 0$)

Higher concentration and temperature are maintained on the bottom wall while the top wall is subjected to lower concentration and temperatures (Dirichlet boundary condition). The side walls are both thermally and solutally insulated (Neumann boundary condition):

2.1.1 Traction free boundary condition

In fluid mechanics, traction or the stress vector on a fluid element is given as :

$$\tau = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \quad (6)$$

Where,

$$\sigma_{ii} = -p + 2\mu \frac{\partial u_i}{\partial x_i} - \frac{2\mu}{3} \nabla \cdot \mathbf{V} \text{ and } \tau_{ij} = \mu \Sigma \left(\frac{\partial u_i}{\partial x_j} \right) \quad (7)$$

If the incompressible continuity equation (equation(1)) is satisfied, $\nabla \cdot \mathbf{V} = 0$, for which, $\sigma_{ii} = -p + 2\mu \partial u_i / \partial x_i$. The term "traction-free" signifies no shear or normal stresses on a face i.e. $\tau \hat{n} = 0$. This postulate is simplified for top and bottom walls, where $\hat{n} = 0, 1, 0$ as :

$$\begin{aligned} \sigma_{xx} &= 0 \text{ and } \tau_{xy} = \tau_{yx} = 0 \\ \Rightarrow -p + 2\mu \frac{\partial u}{\partial x} &= 0 \text{ and } \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) = 0 \end{aligned} \quad (8)$$

The viscosity term in the normal stress component would make $2\mu \partial u / \partial x \rightarrow 0$. As $v = 0$, we've $\partial v / \partial x = 0$. Thus the boundary conditions at the top and bottom wall become :

$$p = 0 \text{ and } \frac{\partial u}{\partial y} = 0 \quad (9)$$

2.1.2 Symmetry boundary condition

The term 'symmetry' implies that the face acts as a mirror or plane of symmetry to the value of variables across it. In the staggered formulation (discussed further), variables are positioned on either side of a face/gridline, for which this boundary condition would make sense while implementing the code, which can be represented mathematically as $\vec{\nabla} \phi \cdot \hat{n} = 0$, where ϕ is any variable of interest.

2.1.3 No-slip boundary condition

The property of a fluid to stick to a surface causes this kind of situation. Here, the fluid velocity is made equal to the surface velocity at a solid wall. If the wall is stationary, this boundary condition is mathematically represented as $\mathbf{V} \times \hat{n} = 0$. If impenetrable/nonporous wall is considered, $\mathbf{V} \cdot \hat{n} = 0$, as well.

Normalising these parameters, we've the simplified boundary conditions as :

- at top wall : $\partial u' / \partial y' = v' = p' = 0$, $T' = 0$ and $C' = 0$
- at bottom wall : $\partial u' / \partial y' = v' = p' = 0$, $T' = 1$ and $C' = 1$
- at side walls : $u' = \partial v / \partial x' = \partial p' / \partial x' = 0$, $\partial T' / \partial x' = 0$ and $\partial C' / \partial x' = 0$.

As these parameters are coupled, we need to solve them iteratively in a numerical method. Finite Volume Method, with under-relaxation, is employed for this purpose.

3 Numerical Method

As has been seen, the major difficulty encountered during the solution of incompressible flow is the non-availability of any obvious equation for the pressure. This difficulty can be resolved in the stream-function-vorticity approach. This approach loses its advantage when the three-dimensional flow is computed because a single scalar stream-function does not exist in three-dimensional space. A three-dimensional problem demands a primitive-variable approach. Efforts have been made so that two-dimensional and three-dimensional problems could be computed following a primitive variable approach without encountering non-physical wiggles in the pressure distribution. As a remedy, it has been suggested to employ a different grid for each dependent variable. Such a staggered grid for the dependant variables in a flow field was first used by Harlow and Welch (1965) in their very well-known MAC (Marker and Cell) method. Since then, it has been used by many researchers. Specifically, the SIMPLE (Semi Implicit Method for Pressure Linked Equations) procedure of Patankar and Spalding (1972) has become popular. Figure 2 shows a two-dimensional staggered grid where independent variables ($u_{i,j}$, $v_{i,j}$ and $p_{i,j}$) with the same indices staggered to one another. Extension to three-dimensions is straight-forward. The computational domain is divided into several cells, shown as "main control volume" in Fig. 2. The velocity components' location is at the center of the cell faces to which they are normal. If a uniform grid is used, the locations are exactly at the midway between the grid points. In such cases, the pressure difference between the two adjacent cells is the driving force for the velocity component located at these cells' interface. The finite-difference approximation is now physically meaningful, and the pressure field will accept a reasonable pressure distribution for a correct velocity field. The goal is to solve for u , v , p , T and C as a function

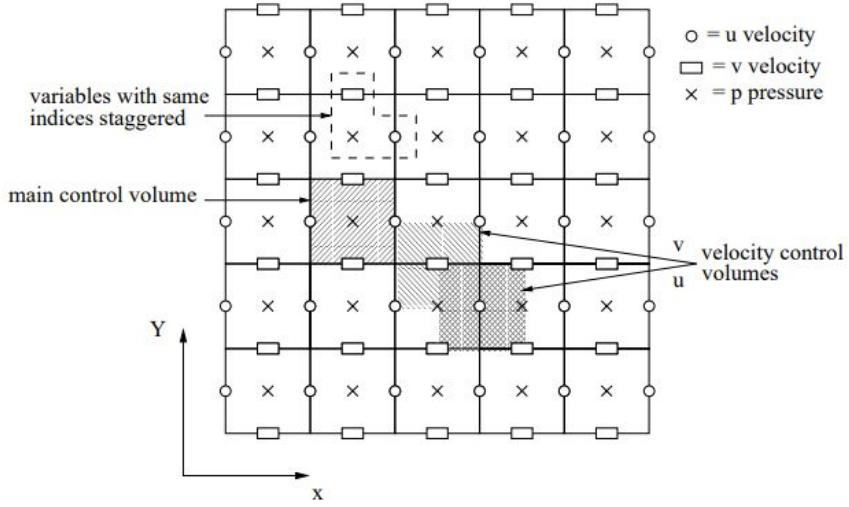


Figure 2: Staggered Grid

of x and y over the grid/mesh shown above. However, from equations (1)-(5), we see an explicit equation for pressure is not available. Thus, the incompressible continuity equation or equation (1) helps derive a Poisson equation for pressure in the numerical form. In our methodology, pressure, temperature, and concentration control volumes are coinciding. Those for x and y velocities are shown in figure 2. To carry out the finite volume method, a weak form of the equations (1)-(5) is needed. These are derived as follows :

From equation (2) and (1),

$$\begin{aligned} -\frac{\partial p'}{\partial x'} + \left(\frac{\partial^2 u'}{\partial x'^2} + \frac{\partial^2 u'}{\partial y'^2} \right) &= \frac{1}{Pr} \left(2u' \frac{\partial u'}{\partial x'} - u' \frac{\partial u'}{\partial x'} + v' \frac{\partial u'}{\partial y'} + u' \frac{\partial v'}{\partial y'} - u' \frac{\partial v'}{\partial y'} \right) \\ &= \frac{1}{Pr} \left(\frac{\partial u'^2}{\partial x'} + \frac{\partial v'u'}{\partial y'} - u' \left(\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} \right) \right) \\ &= \frac{1}{Pr} \left(\frac{\partial u'^2}{\partial x'} + \frac{\partial v'u'}{\partial y'} \right) \end{aligned}$$

\therefore equation(2) becomes :

$$\frac{1}{Pr} \left(\frac{\partial u'^2}{\partial x'} + \frac{\partial v'u'}{\partial y'} \right) = -\frac{\partial p'}{\partial x'} + \left(\frac{\partial^2 u'}{\partial x'^2} + \frac{\partial^2 u'}{\partial y'^2} \right) \quad (10)$$

similarly, equations (3)-(5) become :

$$\frac{1}{Pr} \left(\frac{\partial u' v'}{\partial x'} + \frac{\partial v'^2}{\partial y'} \right) - Ra_T T' - Ra_S C' = -\frac{\partial p'}{\partial y'} + \left(\frac{\partial^2 v'}{\partial x'^2} + \frac{\partial^2 v'}{\partial y'^2} \right) \quad (11)$$

$$\frac{\partial u' T'}{\partial x'} + \frac{\partial v' T'}{\partial y'} = \left(\frac{\partial^2 T'}{\partial x'^2} + \frac{\partial^2 T'}{\partial y'^2} \right) \quad (12)$$

$$\frac{\partial u' C'}{\partial x'} + \frac{\partial v' C'}{\partial y'} = \tau \left(\frac{\partial^2 C'}{\partial x'^2} + \frac{\partial^2 C'}{\partial y'^2} \right) \quad (13)$$

In order to derive the finite volume equations, equations (6)-(7) are integrated over their respected control volumes.

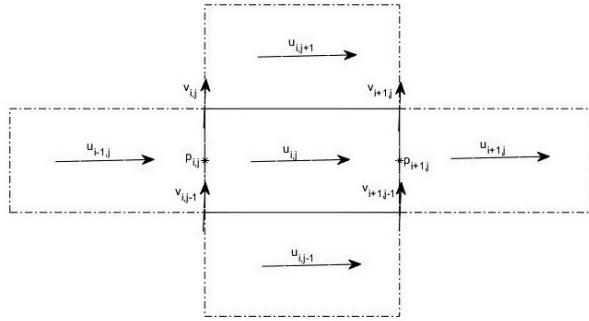


Figure 3: x-velocity control volume

$$\begin{aligned} \iint_{CV} \left(\frac{1}{Pr} \left(\frac{\partial u'^2}{\partial x'} + \frac{\partial v' u'}{\partial y'} \right) \right) dV &= \iint_{CV} \left(-\frac{\partial p'}{\partial x'} + \left(\frac{\partial^2 u'}{\partial x'^2} + \frac{\partial^2 u'}{\partial y'^2} \right) \right) dV \\ \Rightarrow \frac{1}{Pr} \iint \frac{\partial u'^2}{\partial x'} dx dy + \frac{1}{Pr} \iint \frac{\partial v' u'}{\partial y'} dx dy &= \iint -\frac{\partial p'}{\partial x'} dx dy + \iint \frac{\partial^2 u'}{\partial x'^2} dx dy + \iint \frac{\partial^2 u'}{\partial y'^2} dy dx \\ \Rightarrow T_{1u} + T_{2u} &= T_{3u} + T_{4u} + T_{5u} \end{aligned}$$

term T_{1u} and T_{2u} are the convective terms, which are given as :

$$\begin{aligned} T_{1u} &= \frac{1}{Pr} \frac{1}{\Delta x} \left(\left(\frac{u_{i+1,j}^k + u_{i,j}^k}{2} \right) \left(\frac{u_{i+1,j}^{k-1} + u_{i,j}^{k-1}}{2} \right) - \left(\frac{u_{i,j}^k + u_{i-1,j}^k}{2} \right) \left(\frac{u_{i,j}^{k-1} + u_{i-1,j}^{k-1}}{2} \right) \right) \Delta x \Delta y \\ &= \frac{1}{Pr} \left(\left(\frac{u_{i+1,j}^k + u_{i,j}^k}{2} \right) \left(\frac{u_{i+1,j}^{k-1} + u_{i,j}^{k-1}}{2} \right) - \left(\frac{u_{i,j}^k + u_{i-1,j}^k}{2} \right) \left(\frac{u_{i,j}^{k-1} + u_{i-1,j}^{k-1}}{2} \right) \right) \Delta y \end{aligned} \quad (14)$$

$$\begin{aligned} T_{2u} &= \frac{1}{Pr} \frac{1}{\Delta y} \left(\left(\frac{u_{i,j+1}^k + u_{i,j}^k}{2} \right) \left(\frac{v_{i+1,j}^{k-1} + v_{i,j}^{k-1}}{2} \right) - \left(\frac{u_{i,j}^k + u_{i,j-1}^k}{2} \right) \left(\frac{v_{i,j-1}^{k-1} + v_{i+1,j-1}^{k-1}}{2} \right) \right) \Delta x \Delta y \\ &= \frac{1}{Pr} \left(\left(\frac{u_{i,j+1}^k + u_{i,j}^k}{2} \right) \left(\frac{v_{i+1,j}^{k-1} + v_{i,j}^{k-1}}{2} \right) - \left(\frac{u_{i,j}^k + u_{i,j-1}^k}{2} \right) \left(\frac{v_{i,j-1}^{k-1} + v_{i+1,j-1}^{k-1}}{2} \right) \right) \Delta x \end{aligned} \quad (15)$$

The pressure gradient term T_{3u} is given as :

$$\begin{aligned} T_{3u} &= - \left(\frac{p_{i+1,j} - p_{i,j}}{\Delta x} \right) \Delta x \Delta y \\ &= -(p_{i+1,j} - p_{i,j}) \Delta y \end{aligned} \quad (16)$$

Terms T_{4u} and T_{5u} , give the diffusive parts as :

$$\begin{aligned} T_{4u} &= \frac{1}{\Delta x} \left(\left(\frac{u_{i+1,j}^k - u_{i,j}^k}{\Delta x} \right) - \left(\frac{u_{i,j}^k - u_{i-1,j}^k}{\Delta x} \right) \right) \Delta x \Delta y \\ &= (u_{i+1,j}^k - 2u_{i,j}^k + u_{i-1,j}^k) \frac{\Delta y}{\Delta x} \end{aligned} \quad (17)$$

$$\begin{aligned}
T_{5u} &= \frac{1}{\Delta y} \left(\left(\frac{u_{i,j+1}^k - u_{i,j}^k}{\Delta y} \right) - \left(\frac{u_{i,j}^k - u_{i,j-1}^k}{\Delta y} \right) \right) \Delta x \Delta y \\
&= (u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k) \frac{\Delta x}{\Delta y}
\end{aligned} \tag{18}$$

where, k represents the no. of iterations. Similar scheme is carried out of equation (7) as well.

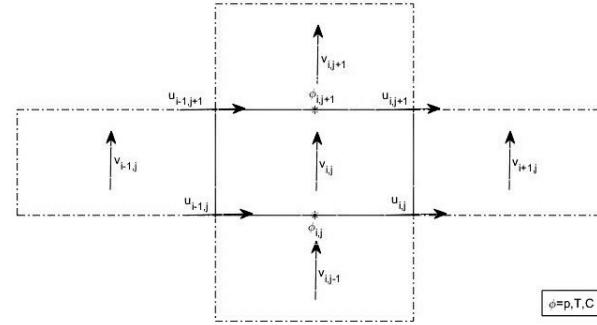


Figure 4: y-velocity control volume

$$\begin{aligned}
&\iint_{CV} \left(\frac{1}{Pr} \left(\frac{\partial u'^2}{\partial x'} + \frac{\partial v' u'}{\partial y'} \right) \right) dV = \iint_{CV} \left(-\frac{\partial p'}{\partial x'} + \left(\frac{\partial^2 u'}{\partial x'^2} + \frac{\partial^2 u'}{\partial y'^2} \right) \right) dV \\
&\implies \frac{1}{Pr} \iint \frac{\partial u' v'}{\partial x'} dx dy + \frac{1}{Pr} \iint \frac{\partial v'^2}{\partial y'} dy dx - Ra_T \iint T dx dy - Ra_S \iint C dx dy \\
&\quad = \iint -\frac{\partial p'}{\partial y'} dy dx + \iint \frac{\partial^2 v'}{\partial x'^2} dx dy + \iint \frac{\partial^2 v'}{\partial y'^2} dy dx \\
&\implies T_{1v} + T_{2v} + T_{3v} + T_{4v} = T_{5v} + T_{6v} + T_{7v}
\end{aligned}$$

Terms T_{1v} and T_{2v} are given as :

$$\begin{aligned}
T_{1v} &= \frac{1}{Pr} \frac{1}{\Delta y} \left(\left(\frac{v_{i+1,j}^k + v_{i,j}^k}{2} \right) \left(\frac{u_{i,j+1}^{k-1} + u_{i,j}^{k-1}}{2} \right) - \left(\frac{v_{i,j}^k + v_{i-1,j}^k}{2} \right) \left(\frac{u_{i-1,j+1}^{k-1} + u_{i-1,j}^{k-1}}{2} \right) \right) \Delta x \Delta y \\
&= \frac{1}{Pr} \left(\left(\frac{v_{i+1,j}^k + v_{i,j}^k}{2} \right) \left(\frac{u_{i,j+1}^{k-1} + u_{i,j}^{k-1}}{2} \right) - \left(\frac{v_{i,j}^k + v_{i-1,j}^k}{2} \right) \left(\frac{u_{i-1,j+1}^{k-1} + u_{i-1,j}^{k-1}}{2} \right) \right) \Delta y
\end{aligned} \tag{19}$$

$$\begin{aligned}
T_{2v} &= \frac{1}{Pr} \frac{1}{\Delta y} \left(\left(\frac{v_{i,j+1}^k + v_{i,j}^k}{2} \right) \left(\frac{v_{i,j+1}^{k-1} + v_{i,j}^{k-1}}{2} \right) - \left(\frac{v_{i,j}^k + v_{i,j-1}^k}{2} \right) \left(\frac{v_{i,j}^{k-1} + v_{i,j-1}^{k-1}}{2} \right) \right) \Delta x \Delta y \\
&= \frac{1}{Pr} \left(\left(\frac{v_{i,j+1}^k + v_{i,j}^k}{2} \right) \left(\frac{v_{i,j+1}^{k-1} + v_{i,j}^{k-1}}{2} \right) - \left(\frac{v_{i,j}^k + v_{i,j-1}^k}{2} \right) \left(\frac{v_{i,j}^{k-1} + v_{i,j-1}^{k-1}}{2} \right) \right) \Delta x
\end{aligned} \tag{20}$$

Term T_{5v} is given as :

$$\begin{aligned}
T_{5v} &= - \left(\frac{p_{i,j+1} - p_{i,j}}{\Delta y} \right) \Delta x \Delta y \\
&= - (p_{i,j+1} - p_{i,j}) \Delta x
\end{aligned} \tag{21}$$

for the terms T_{3v} and T_{4v} , we assume a linear profile for T and C over the control volume, i.e. $T(y) = a + by$. \therefore ,

$$T_{i,j} = a + by_j \tag{22}$$

$$T_{i,j+1} = a + by_{j+1} \tag{23}$$

solving equations (18) and (19), we get :

$$b = \frac{T_{i,j+1} - T_{i,j}}{y_{j+1} - y_j} \text{ and } a = \frac{T_{i,j}y_{j+1} - T_{i,j+1}y_j}{y_{j+1} - y_j}$$

Thus, if $y_{j+1} - y_j = \Delta y$,

$$T(y) = T_{i,j} \left(\frac{y_{j+1} - y}{\Delta y} \right) + T_{i,j+1} \left(\frac{y - y_j}{\Delta y} \right) \quad (24)$$

from equation (20), term T_{3v} is given as :

$$\begin{aligned} T_{3v} &= -Ra_T \int \left(\int_{y_j}^{y_{j+1}} T_{i,j} \left(\frac{y_{j+1} - y}{dy} \right) dy + \int_{y_j}^{y_{j+1}} T_{i,j+1} \left(\frac{y - y_j}{dy} \right) dy \right) dx \\ &= -Ra_T \Delta x \left(\frac{T_{i,j}}{\Delta y} \left(y_{j+1} \Delta y - \frac{y_{j+1}^2 - y_j^2}{2 \Delta y} \right) + \frac{T_{i,j+1}}{\Delta y} \left(\frac{y_{j+1}^2 - y_j^2}{2 \Delta y} - y_j \Delta y \right) \right) \\ &= -Ra_T \Delta x \left(\frac{T_{i,j}}{2} (y_{j+1} - y_j) + \frac{T_{i,j+1}}{2} (y_{j+1} - y_j) \right) \\ &= -Ra_T \Delta x \Delta y \left(\frac{T_{i,j} + T_{i,j+1}}{2} \right) \end{aligned} \quad (25)$$

Similarly,

$$T_{4v} = -Ra_S \Delta x \Delta y \left(\frac{C_{i,j} + C_{i,j+1}}{2} \right) \quad (26)$$

The diffusive terms take a form similar to equations (13) and (14), i.e.,

$$T_{6v} = (v_{i+1,j}^k - 2v_{i,j}^k + v_{i-1,j}^k) \frac{\Delta y}{\Delta x} \quad (27)$$

$$T_{7v} = (v_{i,j+1}^k - 2v_{i,j}^k + v_{i,j-1}^k) \frac{\Delta x}{\Delta y} \quad (28)$$

Rearranging the terms, equations (10)-(14) form the discretised x -momentum equation as :

$$\begin{aligned} &\left(\left(\frac{u_{i+1,j}^{k-1} + u_{i,j}^{k-1}}{2} \right) \Delta y - \left(\frac{u_{i,j}^{k-1} + u_{i-1,j}^{k-1}}{2} \right) \Delta y + \left(\frac{v_{i+1,j}^{k-1} + v_{i,j}^{k-1}}{2} \right) \Delta x - \left(\frac{v_{i,j-1}^{k-1} + v_{i+1,j-1}^{k-1}}{2} \right) \Delta x + 4 \frac{\Delta y}{\Delta x} + 4 \frac{\Delta x}{\Delta y} \right) \frac{u_{i,j}^k}{2} \\ &+ \left(\left(\frac{u_{i+1,j}^{k-1} + u_{i,j}^{k-1}}{2} \right) \Delta y - 2 \frac{\Delta y}{\Delta x} \right) \frac{u_{i+1,j}^k}{2} - \left(\left(\frac{u_{i,j}^{k-1} + u_{i-1,j}^{k-1}}{2} \right) \Delta y + 2 \frac{\Delta y}{\Delta x} \right) \frac{u_{i-1,j}^k}{2} + \left(\left(\frac{v_{i+1,j}^{k-1} + v_{i,j}^{k-1}}{2} \right) \Delta x - 2 \frac{\Delta x}{\Delta y} \right) \frac{u_{i,j+1}^k}{2} \\ &- \left(\left(\frac{v_{i,j-1}^{k-1} + v_{i+1,j-1}^{k-1}}{2} \right) \Delta x + 2 \frac{\Delta x}{\Delta y} \right) \frac{u_{i,j-1}^k}{2} + Pr(p_{i+1,j} - p_{i,j}) \Delta y = 0 \\ \implies a_P u_{i,j}^K &= a_E u_{i+1,j}^k + a_W u_{i-1,j}^k + a_N u_{i,j+1}^k + a_S u_{i,j-1}^k + b^u \end{aligned} \quad (29)$$

where, a_P , a_E , etc. contain the convective and diffusive coefficients, represented as a^C and a^D respectively. In other words, $a_P = a_P^C + a_P^D$, etc. the term b_u contains the pressure gradient term. A similar form is derived for y -momentum equation as well :

$$\begin{aligned} &\left(\left(\frac{u_{i,j+1}^{k-1} + u_{i,j}^{k-1}}{2} \right) \Delta y - \left(\frac{u_{i-1,j+1}^{k-1} + u_{i-1,j}^{k-1}}{2} \right) \Delta y + \left(\frac{v_{i,j+1}^{k-1} + v_{i,j}^{k-1}}{2} \right) \Delta x - \left(\frac{v_{i,j-1}^{k-1} + v_{i+1,j-1}^{k-1}}{2} \right) \Delta x + 4 \frac{\Delta y}{\Delta x} + 4 \frac{\Delta x}{\Delta y} \right) \frac{v_{i,j}^k}{2} \\ &+ \left(\left(\frac{u_{i,j+1}^{k-1} + u_{i,j}^{k-1}}{2} \right) \Delta y - 2 \frac{\Delta y}{\Delta x} \right) \frac{v_{i+1,j}^k}{2} - \left(\left(\frac{u_{i,j}^{k-1} + u_{i-1,j}^{k-1}}{2} \right) \Delta y + 2 \frac{\Delta y}{\Delta x} \right) \frac{v_{i-1,j}^k}{2} + \left(\left(\frac{v_{i,j+1}^{k-1} + v_{i,j}^{k-1}}{2} \right) \Delta x - 2 \frac{\Delta x}{\Delta y} \right) \frac{v_{i,j+1}^k}{2} \\ &- \left(\left(\frac{v_{i,j-1}^{k-1} + v_{i+1,j-1}^{k-1}}{2} \right) \Delta x + 2 \frac{\Delta x}{\Delta y} \right) \frac{v_{i,j-1}^k}{2} + Pr(p_{i,j+1} - p_{i,j}) \Delta x \\ &- \Delta x \Delta y \left(Ra_T \left(\frac{T_{i,j} + T_{i,j+1}}{2} \right) - Ra_S \left(\frac{C_{i,j} + C_{i,j+1}}{2} \right) \right) = 0 \\ \implies a_P v_{i,j}^k &= a_E v_{i+1,j}^k + a_W v_{i-1,j}^k + a_N v_{i,j+1}^k + a_S v_{i,j-1}^k + b_v \end{aligned} \quad (30)$$

Here b_v will contain the temperature and concentration terms as well.

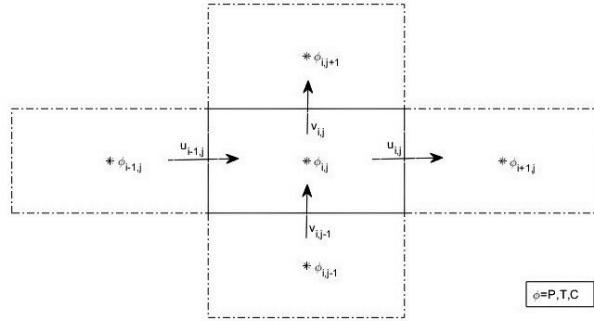


Figure 5: pressure/temperature/concentration control volume

Coming to equations (8) and (9),

$$\begin{aligned}
 & \iint_{CV} \left(\frac{\partial uT}{\partial x} + \frac{\partial vT}{\partial y} \right) dV = \iint_{CV} \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) dV \\
 \Rightarrow & \iint \left(\frac{\partial uT}{\partial x} \right) dx dy + \iint \left(\frac{\partial vT}{\partial y} \right) dy dx = \iint \frac{\partial^2 T}{\partial x^2} dx dy + \iint \frac{\partial^2 T}{\partial y^2} dy dx \\
 \Rightarrow & T_{1T} + T_{2T} = T_{3T} + T_{4T}
 \end{aligned} \tag{31}$$

where,

$$T_{1T} = \Delta y \left(u_{i,j}^{k-1} \left(\frac{T_{i+1,j}^k + T_{i,j}^k}{2} \right) - u_{i-1,j}^{k-1} \left(\frac{T_{i,j}^k + T_{i-1,j}^k}{2} \right) \right) \tag{32}$$

$$T_{2T} = \Delta x \left(v_{i,j}^{k-1} \left(\frac{T_{i,j+1}^k + T_{i,j}^k}{2} \right) - v_{i,j-1}^{k-1} \left(\frac{T_{i,j}^k + T_{i,j-1}^k}{2} \right) \right) \tag{33}$$

$$T_{3T} = \frac{\Delta y}{\Delta x} (T_{i+1,j} - 2T_{i,j} + T_{i-1,j}) \tag{34}$$

$$T_{4T} = \frac{\Delta x}{\Delta y} (T_{i,j+1} - 2T_{i,j} + T_{i,j-1}) \tag{35}$$

Similar forms are obtained for equation (9). Rearranging the terms, we get,

$$\begin{aligned}
 & \left(\frac{\Delta y}{2} (u_{i,j}^{k-1} - u_{i-1,j}^{k-1}) + \frac{\Delta x}{2} (v_{i,j}^{k-1} - v_{i,j-1}^{k-1}) + 2 \frac{\Delta y}{\Delta x} + 2 \frac{\Delta x}{\Delta y} \right) T_{i,j}^k + \\
 & \left(\frac{\Delta y}{2} u_{i,j}^{k-1} - \frac{\Delta y}{\Delta x} \right) T_{i+1,j}^k + \left(-\frac{\Delta y}{2} u_{i-1,j}^{k-1} - \frac{\Delta x}{\Delta y} \right) T_{i-1,j}^k + \left(-\frac{\Delta x}{2} v_{i,j-1}^{k-1} - \frac{\Delta x}{\Delta y} \right) T_{i,j-1}^k = 0 \\
 \Rightarrow & a_P \phi_{i,j}^k = a_E \phi_{i+1,j}^k + a_W \phi_{i-1,j}^k + a_N \phi_{i,j+1}^k + a_S \phi_{i,j-1}^k
 \end{aligned} \tag{36}$$

where, $\phi \in \{T, C\}$. Here, too the coefficients contain both convective and diffusive coefficients, which were dealt separately in the numerical program. However, till now, we have'nt yet got any explicit equation for pressure term. For this purpose, a predictor-corrector method is used, which is summarised as :

- We start with initial estimates of u , v and p as u^* , v^* and p^* ,such that,

$$a_P u_{i,j}^* = \sum a_{nb} u_{nb}^* - (p_{i+1,j}^* - p_{i,j}^*) \Delta y \tag{37}$$

$$a_P v_{i,j}^* = \sum a_{nb} v_{nb}^* - (p_{i,j+1}^* - p_{i,j}^*) \Delta x + b_v \tag{38}$$

- These velocities may or may not satisfy the continuity equation, for which a correction term is added to each of the estimates, i.e.

$$u = u^* + u' \implies u' = u - u^* \tag{39}$$

$$v = v^* + v' \implies v' = v - v^* \tag{40}$$

$$p = p^* + p' \implies p' = p - p^* \tag{41}$$

The corrected velocities are given by :

$$a_P u_{i,j} = \sum a_{nb} u_{nb} - (p_{i+1,j} - p_{i,j}) \Delta y \quad (42)$$

$$a_P v_{i,j} = \sum a_{nb} v_{nb} - (p_{i,j+1} - p_{i,j}) \Delta x + b_v \quad (43)$$

- Subtracting equation (33) from (38) and (34) from (39), we get :

$$a_P u'_{i,j} = \sum a_{nb} u'_{nb} - (p'_{i+1,j} - p'_{i,j}) \Delta y \quad (44)$$

$$a_P v'_{i,j} = \sum a_{nb} v'_{nb} - (p'_{i,j+1} - p'_{i,j}) \Delta x \quad (45)$$

Putting $i \rightarrow i-1$ and $j \rightarrow j-1$ in above equations,

$$a_P u'_{i-1,j} = \sum a_{nb} u'_{nb} - (p'_{i,j} - p'_{i-1,j}) \Delta y \quad (46)$$

$$a_P v'_{i,j-1} = \sum a_{nb} v'_{nb} - (p'_{i,j} - p'_{i,j-1}) \Delta x \quad (47)$$

The velocity corrections in the neighbouring cells are considered negligible. Thus, equations (40)-(43) become :

$$u'_{i,j} = - (p'_{i+1,j} - p'_{i,j}) \frac{\Delta y}{a_P} \quad (48)$$

$$v'_{i,j} = - (p'_{i,j+1} - p'_{i,j}) \frac{\Delta x}{a_P} \quad (49)$$

$$u'_{i-1,j} = - (p'_{i,j} - p'_{i-1,j}) \frac{\Delta y}{a_P} \quad (50)$$

$$v'_{i,j-1} = - (p'_{i,j} - p'_{i,j-1}) \frac{\Delta x}{a_P} \quad (51)$$

- Integrating the continuity equation (Equation (1)) over the control volume shown in figure 5, we get :

$$\begin{aligned} & \iint_{CV} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) dV = 0 \\ \implies & \iint \frac{\partial u}{\partial x} dx dy + \iint \frac{\partial v}{\partial y} dy dx = 0 \\ \implies & (u_{i,j} - u_{i-1,j}) \Delta y + (u_{i,j} - u_{i,j-1}) \Delta x = 0 \end{aligned}$$

Thus, putting $u = u^* + u'$,etc. from equations (44)-(47),

$$\begin{aligned} & (u_{i,j}^* - u_{i-1,j}^*) \Delta y + (v_{i,j}^* - v_{i,j-1}^*) \Delta x + (u'_{i,j} - u'_{i-1,j}) \Delta y + (v'_{i,j} - v'_{i,j-1}) \Delta x = 0 \\ \implies & (u_{i,j}^* - u_{i-1,j}^*) \Delta y + (v_{i,j}^* - v_{i,j-1}^*) \Delta x + (-p'_{i+1,j} + p'_{i,j} + p'_{i,j} - p'_{i-1,j}) \frac{\Delta y^2}{a_P} + (-p'_{i,j+1} + p'_{i,j} + p'_{i,j} - p'_{i,j-1}) \frac{\Delta x^2}{a_P} = 0 \\ \implies & (-p'_{i+1,j} + 2p'_{i,j} - p'_{i-1,j}) \frac{\Delta y^2}{a_P} + (-p'_{i,j+1} + 2p'_{i,j} - p'_{i,j-1}) \frac{\Delta x^2}{a_P} = -(u_{i,j}^* - u_{i-1,j}^*) \Delta y - (v_{i,j}^* - v_{i,j-1}^*) \Delta x \\ \implies & \left(\frac{2\Delta y^2}{a_P} + \frac{2\Delta x^2}{a_P} \right) p'_{i,j} = \frac{\Delta y^2}{a_P} p'_{i+1,j} + \frac{\Delta y^2}{a_P} p'_{i-1,j} + \frac{\Delta x^2}{a_P} p'_{i,j+1} + \frac{\Delta x^2}{a_P} p'_{i,j-1} - (u_{i,j}^* - u_{i-1,j}^*) \Delta y - (v_{i,j}^* - v_{i,j-1}^*) \Delta x \\ \implies & a_P p'_{i,j} = \sum a_{nb} p'_{nb} + b^p \end{aligned} \quad (52)$$

such that, $a_p = \Delta y^2/a_P$ and $b_p = -(u_{i,j}^* - u_{i-1,j}^*) \Delta y - (v_{i,j}^* - v_{i,j-1}^*) \Delta x$. Correction of pressure requires application of under-relaxation such that,

$$p = p^* + \alpha_p p' \quad (53)$$

- From the above parameters, the corrected velocities are used to solve equations (32).

3.1 Method of relaxation

The system of linear equations derived above, can be cast into a form $A\phi = B$ such that $\phi \in \{u, v, p, T, c\}$. Iterative methods are generally of the form

$$M\phi^k = N\phi^{k-1} + B \quad (54)$$

where, M and N are iteration matrices and k is the iteration counter. At convergence, when $\phi^k \rightarrow \phi^{k-1}$, we must have $A = M - N$. Thus, equation (55) can be re written as

$$\begin{aligned} M\phi^k - M\phi^{k-1} &= N\phi^{k-1} - M\phi^{k-1} + B \\ \implies M(\phi^k - \phi^{k-1}) &= B - (M - N)\phi^{k-1} \\ \therefore M\delta^k &= \rho^{k-1} \end{aligned} \quad (55)$$

where, δ is the residual at the current iteration. ϕ^k can be found out by solving $\delta^k = M^{-1}\rho^{k-1}$ and then doing $\phi^k = \phi^{k-1} + \delta^k$. This depends on the initial guess of M , which may or may not lead to a convergent solution. Thus, a relaxation parameter α is used for this purpose. Depending on the value of α the relaxation technique is classified as over-relaxation ($\alpha > 1$) or under-relaxation ($\alpha < 1$). The under relaxation factors ensure that the solution from one step to the next does not change too much as it then might get unstable. However, this takes a lot of steps even after a steady solution is reached. On the other hand, over-relaxation is used in cases where, chances of blowing up of solution is least, like low Reynold's no. flows, etc. where a faster solution is expected. The over-relaxation factor gives solution of the next step, by skipping a few intermediate steps in between. The method of under-relaxation was used for the current problem, because of high values of Rayleigh no.s, which give a possibility of diverging solutions at some point of time. This formulation is discussed below.

The general linear equation for the above system is given as :

$$\begin{aligned} a_P\phi_{i,j}^k + a_E\phi_{i+1,j}^k + a_W\phi_{i-1,j}^k + a_N\phi_{i,j+1}^k + a_S\phi_{i,j-1}^k &= b^\phi \\ \implies a_P\phi_{i,j}^k &= b^\phi - (a_E\phi_{i+1,j}^k + a_W\phi_{i-1,j}^k + a_N\phi_{i,j+1}^k + a_S\phi_{i,j-1}^k) \\ \implies a_P\phi_{i,j}^k &= a_P\phi_{i,j}^{k-1} + b^\phi - (a_E\phi_{i+1,j}^k + a_W\phi_{i-1,j}^k + a_N\phi_{i,j+1}^k + a_S\phi_{i,j-1}^k + a_P\phi_{i,j}^{k-1}) \\ \implies \phi_{i,j}^k &= \phi_{i,j}^{k-1} + \frac{b^\phi - \sum a_{nb}\phi_{nb} - a_P\phi_{i,j}^{k-1}}{a_P} \end{aligned} \quad (56)$$

Incorporating under-relaxation.

$$\begin{aligned} \phi_{i,j}^k &= \phi_{i,j}^{k-1} + \alpha \left(\frac{b^\phi - \sum a_{nb}\phi_{nb} - a_P\phi_{i,j}^{k-1}}{a_P} \right) \\ &= \frac{a_P\phi_{i,j}^{k-1} + \alpha(b^\phi - \sum a_{nb}\phi_{nb} - a_P\phi_{i,j}^{k-1})}{a_P} \\ &= \alpha \left(\frac{b^\phi - \sum a_{nb}\phi_{nb} + (\frac{1}{\alpha} - 1)a_P\phi_{i,j}^{k-1}}{a_P} \right) \\ &= \alpha \left(\frac{b^\phi - \sum a_{nb}\phi_{nb} + (\frac{1-\alpha}{\alpha})a_P\phi_{i,j}^{k-1}}{a_P} \right) \end{aligned} \quad (57)$$

If $a_P/\alpha = a'_P$, the above equation becomes :

$$\phi_{i,j}^k = \left(\frac{b^\phi - \sum a_{nb}\phi_{nb} + (1 - \alpha)a'_P\phi_{i,j}^{k-1}}{a'_P} \right) \quad (58)$$

The corresponding under-relaxation values for pressure and velocity equations are α_P ad α_V . The same value of α_P was used for temperature and concentration equations as well. For steady flow systems, α_p is optimal when $\alpha_V = 1 - \alpha_p$ (Ferziger & Peric, 2002). However, optimal under-relaxation factors are problem-dependent (Patankar, 1980) and may not explicitly follow the above equation. For instance, Pantakar (1980) suggests a $\alpha_V = 0.5$ and $\alpha_p = 0.8$ for the lid-driven cavity problem. In general, small values of α_V allow α_p to take any value between 0.1 and 1.0, but with slow convergence, while large values of α_V results in faster convergence, but with a restricted range of useful α_p (Ferziger & Peric, 2002). A trial and error method can be used wherein a small under-relaxation factor is applied in the early iterations and increased until convergence is achieved. Here, the velocity under-relaxation factor applied line-by-line by manipulation of equation (33) and (34) using a pseudo-Jacobi iteration method (Staerdahl, 2016; McDonough, 2007). Equations (33) and (34) can be solved using TDMA or Point Gauss-Siedel method. However, Patankar (1980), Versteeg and Malalasekera (2002), and Recktenweld (2014) suggest the use of a hybrid scheme for the pseudo-Jacobi iterative method as well :

Coefficient	Forward	Central	Hybrid
a_E	$a_E^D + \max(0, -a_E^C)$	$a_E^D - \frac{a_E^C}{2}$	$\max(-a_E^C, \left(a_E^D - \frac{a_E^C}{2}\right), 0)$
a_W	$a_W^D + \max(0, a_W^C)$	$a_W^D + \frac{a_W^C}{2}$	$\max(a_W^C, \left(a_W^D + \frac{a_W^C}{2}\right), 0)$
a_N	$a_N^D + \max(0, -a_N^C)$	$a_N^D - \frac{a_N^C}{2}$	$\max(-a_N^C, \left(a_N^D - \frac{a_N^C}{2}\right), 0)$
a_S	$a_S^D + \max(0, a_S^C)$	$a_S^D + \frac{a_S^C}{2}$	$\max(a_S^C, \left(a_S^D + \frac{a_S^C}{2}\right), 0)$
a_P	$\Sigma a_{nb} + \Sigma(-1)^{nb} a_{nb}^C$	$\Sigma a_{nb} + \Sigma(-1)^{nb} a_{nb}^C$	$\Sigma a_{nb} + \Sigma(-1)^{nb} a_{nb}^C$

4 Visualisation of data

In order to draw inferences from an experiment, post-processing and visualization of data are necessary. Many CFD post-processing tools like Tecplot, GNU Plot, ANSYS CFD post, and even MATLAB have their own module in the form of the 'figure()' command. However, the data needs to be filtered from the corresponding variables obtained from the simulation to use these modules. In our problem, plots of streamlines, isotherms, and iso-concentration lines were generated using Tecplot. Isotherms and Iso-concentration lines are contour levels at various x and y coordinates. However, streamlines need to be looked into in detail.

In CFD post-processing, streamlines are generated from the stream function, which is, in turn, obtained from velocity or vorticity levels. Once the stream function is obtained at various points, streamlines can be plotted as contour stream function levels. It is known that velocity ($\{u, v\}$) is related to stream function (ψ) as

$$u = \frac{\partial \psi}{\partial y} \text{ and } v = -\frac{\partial \psi}{\partial x} \quad (59)$$

and vorticity is obtained from velocities as :

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \quad (60)$$

Combining above equations, we get :

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \quad (61)$$

The above equation is discretised in a central difference as :

$$\begin{aligned} & \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta x^2} + \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\Delta y^2} = -\omega_{i,j} \\ \implies & \psi_{i,j} = \frac{(\psi_{i+1,j} + \psi_{i-1,j}) \Delta y^2 + (\psi_{i,j+1} + \psi_{i,j-1}) \Delta x^2 + \omega_{i,j} \Delta x^2 \Delta y^2}{2(\Delta x^2 + \Delta y^2)} \end{aligned} \quad (62)$$

The above equation is also solved in an iterative manner using under-relaxation as discussed previously. For vorticity, equation (59) is used as :

$$\omega_{i,j} = \left(\frac{v_{i+1,j}^k - v_{i-1,j}^k}{2\Delta x} \right) - \left(\frac{u_{i,j+1}^k - u_{i,j-1}^k}{2\Delta y} \right) \quad (63)$$

It can be noted that streamlines can be obtained from velocities as well, using any numerical integration method like trapezoidal rule, etc.

$$\psi_{i,j} = \int u dy = \frac{\Delta y}{2} \left(u_{i,0}^k + 2 \sum_{J=1}^{j-1} u_{i,J}^k + u_{i,j}^k \right) \quad (64)$$

$$\psi_{i,j} = - \int v dx = - \frac{\Delta x}{2} \left(v_{0,j}^k + 2 \sum_{I=1}^{i-1} v_{I,j}^k + v_{i,j}^k \right) \quad (65)$$

From the values of u^k , v^k , etc., we have to obtain the cell point values to feed them into the plot matrix. This can be done by looking into the control volume arrangement of different variables. Figure (2) approximates the values at cell points as the corresponding average values of neighboring control volumes. With reference to figures (3), (4), and (5), this is done as :

$$u_{i,j}^{cell} = \frac{u_{i,j+1}^k + u_{i,j}^k}{2} \quad (66)$$

$$v_{i,j}^{cell} = \frac{v_{i+1,j}^k + v_{i,j}^k}{2} \quad (67)$$

$$\phi_{i,j}^{cell} = \frac{\phi_{i,j}^k + \phi_{i+1,j}^k + \phi_{i+1,j+1}^k + \phi_{i,j+1}^k}{4} \quad (68)$$

Such that, $\phi \in \{p, T, C\}$. The above variables including ψ and ω can be plotted in any post processing software.

5 Mesh Convergence

The importance of grid in numerical solutions is that discretized equations need to be solved at each point of the domain, for which grid/mesh gives a visual interpretation of the solution domain and governs the solver's accuracy. A coarse grid will give a reasonable solution, but finer grids increase accuracy. However, refining a mesh comes with a cost of CPU timing. Thus, an optimum grid size is expected with a trade-off between solution accuracy and CPU timing. This task is commonly referred to as "mesh convergence" among numerical programmers and researchers. If we keep refining the grid size, we reach a point, after which the solution ceases to vary by a substantial amount. The earliest limit of such grid size is used as an optimum mesh for further calculation.

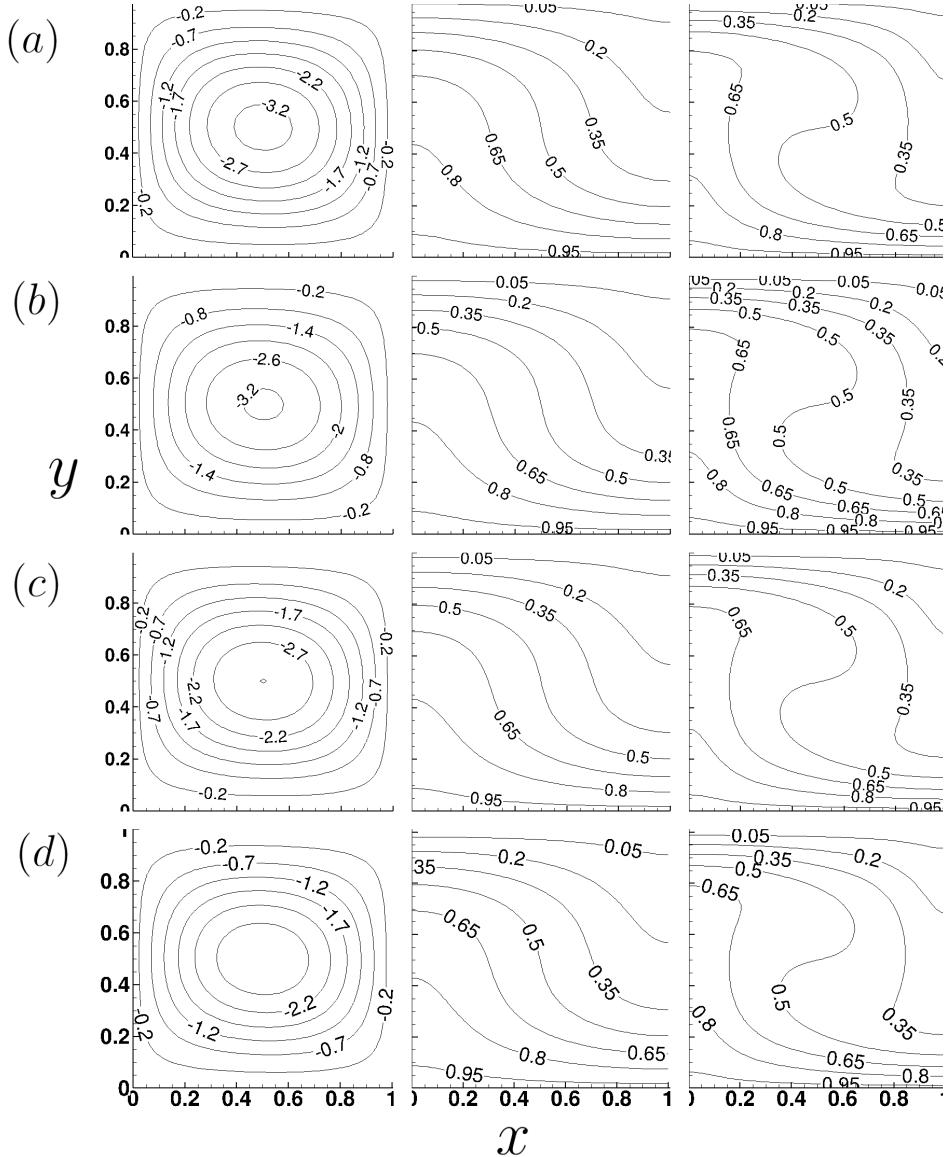


Figure 6: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 2000$, for grid size 30×30 , 40×40 , 50×50 and 60×60 from top to bottom.

From the above figure, it can be seen, after a grid of 50×50 , the solutions don't vary much. Thus, optimum grid size is considered $1/50 = 0.02$ in each direction (x, y) for the current simulation.

6 Results and Validation

The solutal rayleigh no. (Ra_S), is kept constant at 1000. Simulations are done for thermal Rayleigh no. (Ra_T), varying from 2000 to 10000. The simulation results, i.e., streamlines, isotherms, and iso-concentration lines, at different Rayleigh numbers, for Prandtl no. (Pr) 1.0, are presented below. The results were verified with the work of Murty et al. The results of their work are presented below. The results of the current study match with great accuracy to their work.

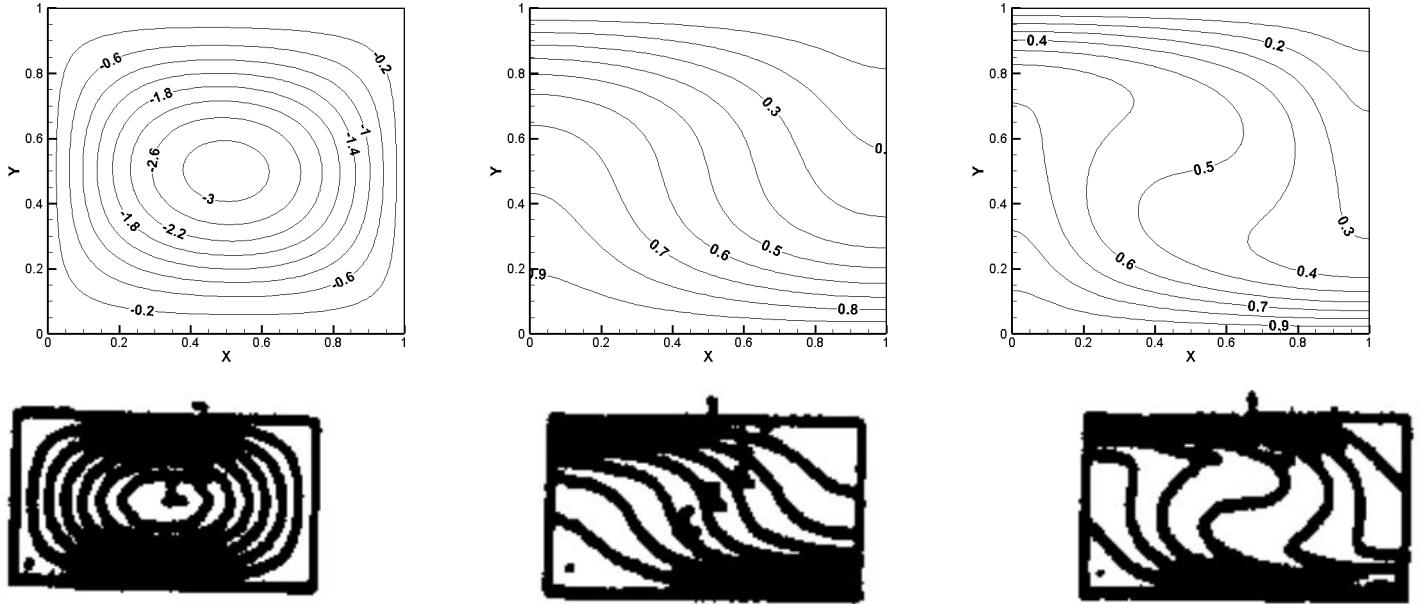


Figure 7: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 2000$ (top row current work) compared with the work of Murty (bottom row).

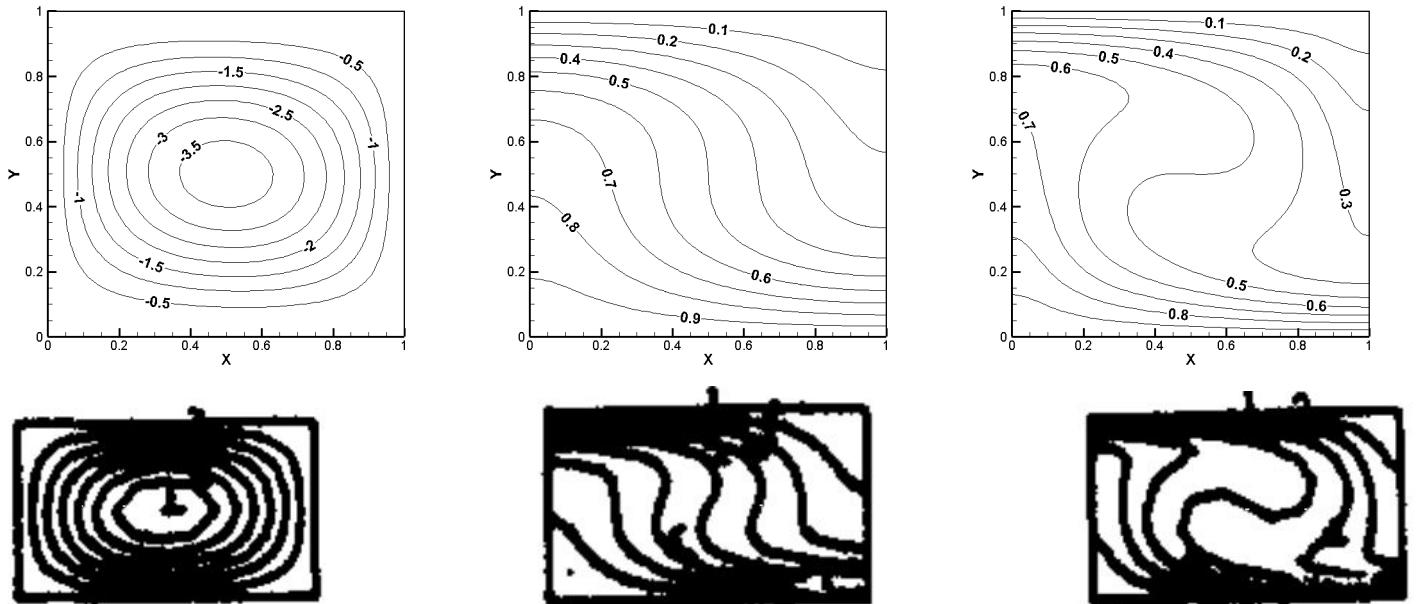


Figure 8: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 2500$ (top row current work) compared with the work of Murty (bottom row)

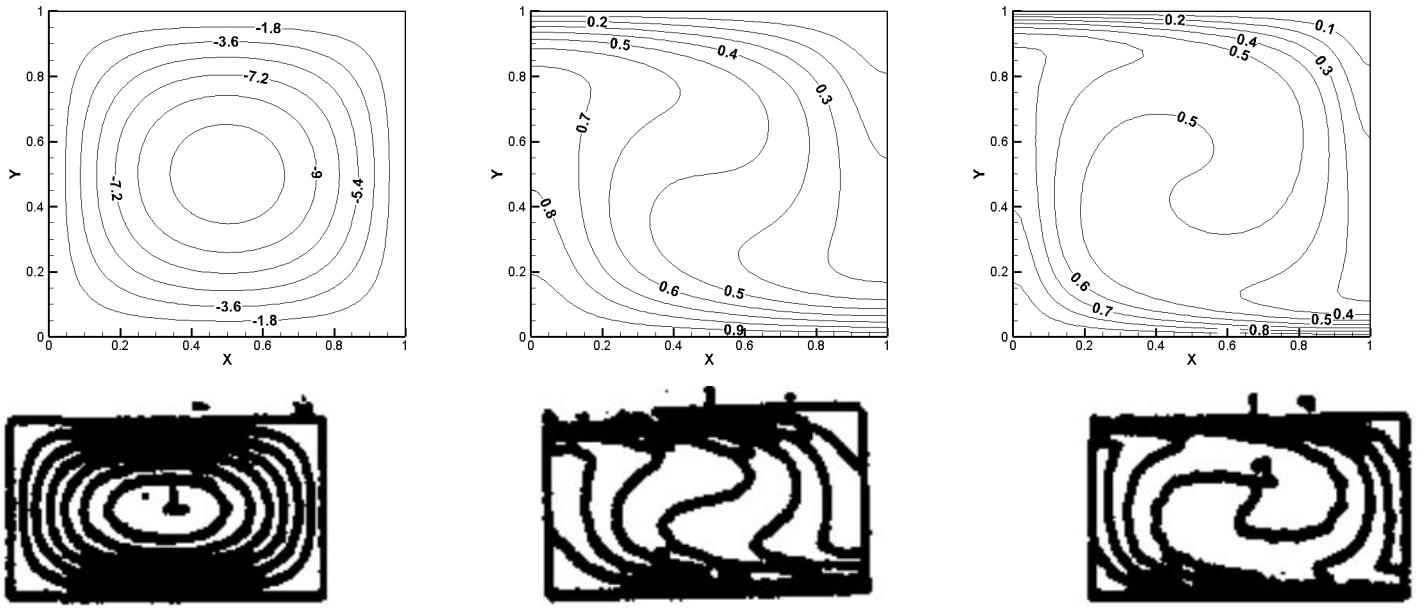


Figure 9: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 5000$ (top row current work) compared with the work of Murty (bottom row)

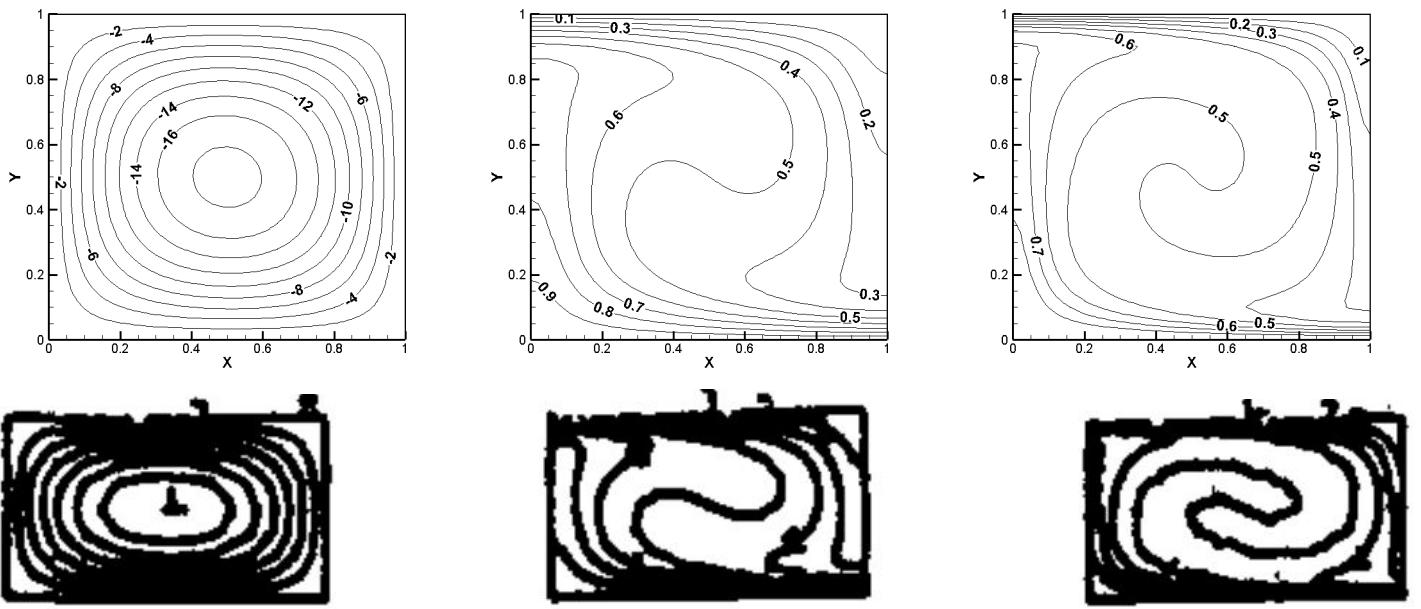


Figure 10: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 10000$ (top row current work) compared with the work of Murty (bottom row)

7 Appendix-I : MATLAB Program linked with the simulation

```

nx=50; ny=50; D=1.0;
dx=D/nx; dy=D/ny;
Pr=1.0; Ra_T=10000; Ra_S=1000;
alpha_V=0.3; alpha_P=0.7;
maxit=120000; maxdiv=1e-6;

```

```

u_k=zeros(nx+1,ny+2); u_star=zeros(nx+1,ny+2); u_dash=zeros(nx+1,ny+2); u_cell=zeros(nx+1,ny+1);
aUE=zeros(nx+1,ny+2); aUW=zeros(nx+1,ny+2); aUN=zeros(nx+1,ny+2); aUS=zeros(nx+1,ny+2);
aUP=zeros(nx+1,ny+2); dU=zeros(nx+1,ny+2);
v_k=zeros(nx+2,ny+1); v_star=zeros(nx+2,ny+1); v_dash=zeros(nx+2,ny+1); v_cell=zeros(nx+1,ny+1);
aVE=zeros(nx+2,ny+1); aVW=zeros(nx+2,ny+1); aVN=zeros(nx+2,ny+1); aVS=zeros(nx+2,ny+1);
aVP=zeros(nx+2,ny+1); dV=zeros(nx+2,ny+1);
p=zeros(nx+2,ny+2); p_dash=zeros(nx+2,ny+2); p_cell=zeros(nx+1,ny+1);
aPE=zeros(nx+2,ny+2); aPW=zeros(nx+2,ny+2); aPN=zeros(nx+2,ny+2); aPS=zeros(nx+2,ny+2);
aPP=zeros(nx+2,ny+2); bPP=zeros(nx+2,ny+2);
T_k=0.5*ones(nx+2,ny+2); T_k_minus_1=0.5*ones(nx+2,ny+2); T_cell=zeros(nx+1,ny+1);
aTE=zeros(nx+2,ny+2); aTW=zeros(nx+2,ny+2); aTN=zeros(nx+2,ny+2); aTS=zeros(nx+2,ny+2);
aTP=zeros(nx+2,ny+2);
C_k=1.0*ones(nx+2,ny+2); C_k_minus_1=1.0*ones(nx+2,ny+2); C_cell=zeros(nx+1,ny+1);
aCE=zeros(nx+2,ny+2); aCW=zeros(nx+2,ny+2); aCN=zeros(nx+2,ny+2); aCS=zeros(nx+2,ny+2);
aCP=zeros(nx+2,ny+2);

u_k_minus_1=u_k; v_k_minus_1=v_k; p_star=p;

for t=1:maxit
    %x-momentum eqn :
    for i=2:nx
        for j=2:ny+1
            %convection coefficients
            aCe=dy*0.5*(u_k_minus_1(i+1,j)+u_k_minus_1(i,j))/Pr;
            aCw=dy*0.5*(u_k_minus_1(i,j)+u_k_minus_1(i-1,j))/Pr;
            aCn=dx*0.5*(v_k_minus_1(i+1,j)+v_k_minus_1(i,j))/Pr;
            aCs=dx*0.5*(v_k_minus_1(i+1,j-1)+v_k_minus_1(i,j-1))/Pr;
            %diffusive coefficients
            aDx=dy/dx; aDy=dx/dy;
            %hybrid scheme
            aUE(i,j)=max([-aCe,(aDx-0.5*aCe),0]);
            aUW(i,j)=max([aCw,(aDx+0.5*aCw),0]);
            aUN(i,j)=max([-aCn,(aDy-0.5*aCn),0]);
            aUS(i,j)=max([aCs,(aDy+0.5*aCs),0]);
            aUP(i,j)=(aUE(i,j)+aUW(i,j)+aUN(i,j)+aUS(i,j)+aCe-aCw+aCn-aCs)/alpha_V;
            %coefficient of pressure correction equation
            dU(i,j)=dy/aUP(i,j);
        end
    end
    u_star=u_k_minus_1;
    for i=2:nx
        for j=2:ny+1
            u_star(i,j)=(aUE(i,j)*u_star(i+1,j)+aUW(i,j)*u_star(i-1,j)+aUN(i,j)*u_star(i,j+1)...
                +aUS(i,j)*u_star(i,j-1)-dy*(p_star(i+1,j)-p_star(i,j))...
                +(1-alpha_V)*aUP(i,j)*u_k_minus_1(i,j))/aUP(i,j);
        end
    end
    %y-momentum eqn
    for i=2:nx+1
        for j=2:ny
            %convection coefficients
            aCe=dy*0.5*(u_k_minus_1(i,j+1)+u_k_minus_1(i,j))/Pr;
            aCw=dy*0.5*(u_k_minus_1(i-1,j+1)+u_k_minus_1(i-1,j))/Pr;
            aCn=dx*0.5*(v_k_minus_1(i,j+1)+v_k_minus_1(i,j))/Pr;
            aCs=dx*0.5*(v_k_minus_1(i,j)+v_k_minus_1(i,j-1))/Pr;
            %diffusive coefficients
            aDx=dy/dx; aDy=dx/dy;
            %hybrid scheme

```

```

aVE(i,j)=max([-aCe,(aDx-0.5*aCe),0]);
aVW(i,j)=max([aCw,(aDx+0.5*aCw),0]);
aVN(i,j)=max([-aCn,(aDy-0.5*aCn),0]);
aVS(i,j)=max([aCs,(aDy+0.5*aCs),0]);
aVP(i,j)=(aVE(i,j)+aVW(i,j)+aVN(i,j)+aVS(i,j)+aCe-aCw+aCn-aCs)/alpha_V;
%coefficient of pressure correction equation
dV(i,j)=dx/aVP(i,j);
end
end
v_star=v_k_minus_1;
for i=2:nx+1
    for j=2:ny
        v_star(i,j)=(aVE(i,j)*v_star(i+1,j)+aVW(i,j)*v_star(i-1,j)+aVN(i,j)*v_star(i,j+1)...
        +aVS(i,j)*v_star(i,j-1)-dy*(p_star(i,j+1)-p_star(i,j))...
        +(1-alpha_V)*aVP(i,j)*v_k_minus_1(i,j)+...
        0.5*Ra_T*(T_k_minus_1(i,j+1)+T_k_minus_1(i,j))*dx*dy+...
        0.5*Ra_S*(C_k_minus_1(i,j+1)+C_k_minus_1(i,j))*dx*dy)/aVP(i,j);
    end
end
%pressure correction equation (poisson eqn) :
for i=2:nx+1
    for j=2:ny+1
        aPE(i,j)=dU(i,j)*dy;
        aPW(i,j)=dU(i-1,j)*dy;
        aPN(i,j)=dV(i,j)*dx;
        aPS(i,j)=dV(i,j-1)*dx;
        aPP(i,j)=aPE(i,j)+aPW(i,j)+aPN(i,j)+aPS(i,j);
        bPP(i,j)=(u_star(i-1,j)-u_star(i,j))*dy+(v_star(i,j-1)-v_star(i,j))*dx;
    end
end
for i=1:nx+2
    for j=1:ny+2
        p_dash(i,j)=0.0;
    end
end
for i=2:nx+1
    for j=2:ny+1
        p_dash(i,j)=(aPE(i,j)*p_dash(i+1,j)+aPW(i,j)*p_dash(i-1,j)+aPN(i,j)*p_dash(i,j+1)...
        +aPS(i,j)*p_dash(i,j-1)+bPP(i,j))/aPP(i,j);
    end
end
%velocity correction terms :
for i=2:nx
    for j=2:ny+1
        u_dash(i,j)=dU(i,j)*(p_dash(i,j)-p_dash(i+1,j));
    end
end
for i=2:nx+1
    for j=2:ny
        v_dash(i,j)=dV(i,j)*(p_dash(i,j)-p_dash(i,j+1));
    end
end
end
%pressure updatation
for i=2:nx+1
    for j=2:ny+1
        p_dash(i,j)=p_star(i,j)+p_dash(i,j)*alpha_P;
    end
end
end

```

```

for i=2:nx
    for j=2:ny+1
        u_k(i,j)=u_star(i,j)+u_dash(i,j);
    end
end
for i=2:nx+1
    for j=2:ny
        v_k(i,j)=v_star(i,j)+v_dash(i,j);
    end
end
%Temperqature and concentration equations :
for i=2:nx+1
    for j=2:ny+1
        aCe=u_k(i,j)*dy; aCw=u_k(i-1,j)*dy; aCn=v_k(i,j)*dx; aCs=v_k(i,j-1)*dx; aDx=dy/dx; aDy=dx/dy;
        %hybrid scheme
        aTE(i,j)=max([-aCe,(aDx-0.5*aCe),0]);
        aTW(i,j)=max([aCw,(aDx+0.5*aCw),0]);
        aTN(i,j)=max([-aCn,(aDy-0.5*aCn),0]);
        aTS(i,j)=max([aCs,(aDy+0.5*aCs),0]);
        aTP(i,j)=(aTE(i,j)+aTW(i,j)+aTN(i,j)+aTS(i,j)+aCe-aCw+aCn-aCs)/alpha_P;
    end
end
T_k=T_k_minus_1;
for i=2:nx+1
    for j=2:ny+1
        T_k(i,j)=(aTE(i,j)*T_k(i+1,j)+aTW(i,j)*T_k(i-1,j)+aTN(i,j)*T_k(i,j+1)+...
                    aTS(i,j)*T_k(i,j-1)+(1-alpha_P)*T_k_minus_1(i,j)*aTP(i,j))/aTP(i,j);
    end
end
%alpha_C=(10^-0.5)*alpha_V;
for i=2:nx+1
    for j=2:ny+1
        aCe=u_k(i,j)*dy; aCw=u_k(i-1,j)*dy; aCn=v_k(i,j)*dx; aCs=v_k(i,j-1)*dx;
        aDx=(10^-0.5)*dy/dx; aDy=(10^-0.5)*dx/dy;
        aCE(i,j)=max([-aCe,(aDx-0.5*aCe),0]);
        aCW(i,j)=max([aCw,(aDx+0.5*aCw),0]);
        aCN(i,j)=max([-aCn,(aDy-0.5*aCn),0]);
        aCS(i,j)=max([aCs,(aDy+0.5*aCs),0]);
        aCP(i,j)=(aCE(i,j)+aCW(i,j)+aCN(i,j)+aCS(i,j)+aCe-aCw+aCn-aCs)/alpha_P;
    end
end
C_k=C_k_minus_1;
for i=2:nx+1
    for j=2:ny+1
        C_k(i,j)=(aCE(i,j)*C_k(i+1,j)+aCW(i,j)*C_k(i-1,j)+aCN(i,j)*C_k(i,j+1)+...
                    aCS(i,j)*C_k(i,j-1)+(1-alpha_P)*C_k_minus_1(i,j)*aCP(i,j))/aCP(i,j);
    end
end
error1=max(max(abs(u_k_minus_1(2:nx,2:ny+1)-u_k(2:nx,2:ny+1))));
error2=max(max(abs(v_k_minus_1(2:nx+1,2:ny)-v_k(2:nx+1,2:ny))));
error3=max(max(abs(T_k_minus_1(2:nx+1,2:ny+1)-T_k(2:nx+1,2:ny+1))));
error4=max(max(abs(C_k_minus_1(2:nx+1,2:ny+1)-C_k(2:nx+1,2:ny+1))));
if(t>100)
    error=max([error1,error2,error3,error4]);
    if(error<maxdiv)
        break;
    end
end

```

```

u_k_minus_1=u_k; v_k_minus_1=v_k; p_star=p_dash; T_k_minus_1=T_k; C_k_minus_1=C_k;
%boundary conditions :
for i=2:nx
    u_k_minus_1(i,1)=u_k_minus_1(i,2); u_k_minus_1(i,ny+2)=u_k_minus_1(i,ny+1);
end
for j=1:ny+2
    u_k_minus_1(1,j)=0.0; u_k_minus_1(nx+1,j)=0.0;
end
for j=2:ny
    v_k_minus_1(1,j)=v_k_minus_1(2,j); v_k_minus_1(nx+2,j)=v_k_minus_1(nx+1,j);
end
for i=1:nx+2
    v_k_minus_1(i,1)=0.0; v_k_minus_1(i,ny+1)=0.0;
end
for i=1:nx+2
    p_dash(i,1)=-p_dash(i,2); p_dash(i,ny+2)=-p_dash(i,ny+1);
end
for j=2:ny+1
    p_dash(nx+2,j)=p_dash(nx+1,j); p_dash(1,j)=p_dash(2,j);
end
for i=2:nx+1
    T_k_minus_1(i,1)=2-T_k_minus_1(i,2); T_k_minus_1(i,ny+2)=-T_k_minus_1(i,ny+1);
end
for j=1:ny+2
    T_k_minus_1(1,j)=T_k_minus_1(2,j); T_k_minus_1(nx+2,j)=T_k_minus_1(nx+1,j);
end
for i=2:nx+1
    C_k_minus_1(i,1)=2-C_k_minus_1(i,2); C_k_minus_1(i,ny+2)=-C_k_minus_1(i,ny+1);
end
for j=1:ny+2
    C_k_minus_1(1,j)=C_k_minus_1(2,j); C_k_minus_1(nx+2,j)=C_k_minus_1(nx+1,j);
end
end

%calculation of grid point parameters by taking average of staggered grid parameters :
for i=1:nx+1
    for j=1:ny+1
        u_cell(i,j)=(u_k(i,j+1)+u_k(i,j))/2;
        v_cell(i,j)=(v_k(i+1,j)+v_k(i,j))/2;
        p_cell(i,j)=(p_dash(i,j)+p_dash(i+1,j)+p_dash(i+1,j+1)+p_dash(i,j+1))/4;
        T_cell(i,j)=(T_k(i,j)+T_k(i+1,j)+T_k(i+1,j+1)+T_k(i,j+1))/4;
        C_cell(i,j)=(C_k(i,j)+C_k(i+1,j)+C_k(i+1,j+1)+C_k(i,j+1))/4;
    end
end

%vorticity :
w=zeros(nx+1,ny+1);
for i=2:nx
    for j=2:ny
        w(i,j)=(-(u_cell(i,j+1)-u_cell(i,j-1))/(2*dy))+((v_cell(i+1,j)-v_cell(i-1,j))/(2*dx));
    end
end

%generation of stream function for streamlines :
psi=zeros(nx+1,ny+1); psi_new=zeros(nx+1,ny+1); errorS=10.0; maxS=0.0;
while(errorS>0.0001)
    for i=2:nx
        for j=2:ny

```

```

psi_new(i,j)=0.25*0.5*(psi(i+1,j)+psi(i-1,j)+psi(i,j+1)+psi(i,j-1)+...
(dx*dx*w(i,j)))+((1-0.5)*psi(i,j));
end
end
maxS=0.0;
for i=2:nx-1
    for j=2:ny-1
        errorS=abs(psi_new(i,j)-psi(i,j));
        if(errorS>maxS)
            maxS=errorS;
        end
    end
end
errorS=maxS;
psi=psi_new;
end

x=linspace(0,1,nx+1); y=linspace(0,1,ny+1); data=zeros((nx+1)*(ny+1),9); k=1;
for i=1:nx+1
    for j=1:ny+1
        data(k,:)=[x(1,i),y(1,j),u_cell(i,j),v_cell(i,j),p_cell(i,j),...
        w(i,j),psi_new(i,j),T_cell(i,j),C_cell(i,j)];
        k=k+1;
    end
end

```

8 Appendix-II : A note on post processing and visualisation of data and the relevant software

The cell point variables obtained from the program, can be easily used to plot streamlines (variable used : "psi_new") in MATLAB using the following subroutine :

```

F=figure(1);
subplot(1,3,1);
[c1,h1]=contour(linspace(0,1,nx+1),linspace(0,1,ny+1),psi_new.',15);
set(h1,'edgecolor','k');
title('Streamline');
set(F,'WindowStyle','docked');

```

The same when done for "subplots", (1,3,2) and (1,3,3), could be used to make isotherm (variable : "T_cell") and iso-concentrations (variable : "C_cell") as well.

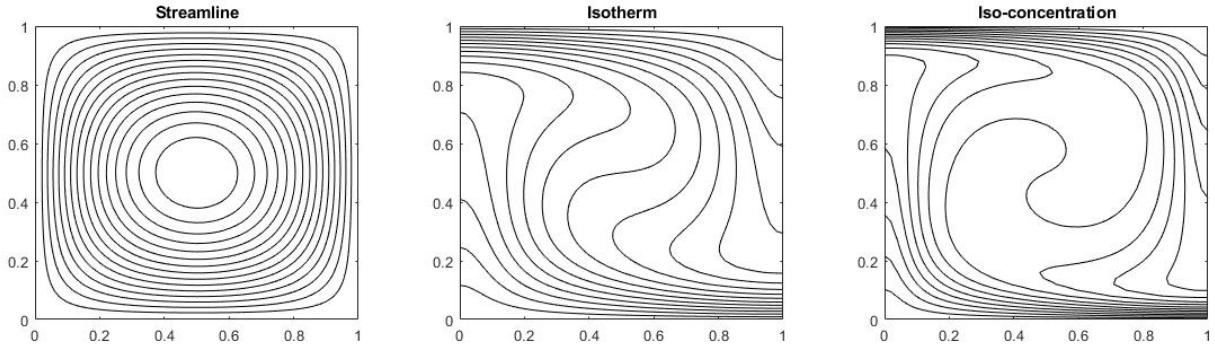


Figure 11: Post Processing image generated from MATLAB, for $Ra_T = 5000$

However, Tecplot was used for better accuracy and clarity of plots. For this purpose, all the matrices (u^{cell} , v^{cell} , p^{cell} , etc) are reshaped into column vectors along with corresponding x and y coordinates. The data was generated as a /.dat or a /.txt file to be compatible with the software's plot function.

9 Appendix-III : Code analysis and flow chart development

A detailed analysis of any computer program is necessary so as to debug logical errors, or such errors which may run effortlessly under some configurations, but fail to produce results when the inputs are changed. Development of flow chart is done to summarise the solution algorithm and to make a code language independent. Flow charts help programmers to export the code into any other convenient languages, without going into detail of the solution algorithm.

One method of analysing numerical codes is to change the grid size and run test cases for every such example. It is inspected whether the results are consistent, even though not exactly similar. Such a method is employed for code analysis, using hand calculations for small no. of grids. The flow chart for the algorithm is given as : According to the above flow

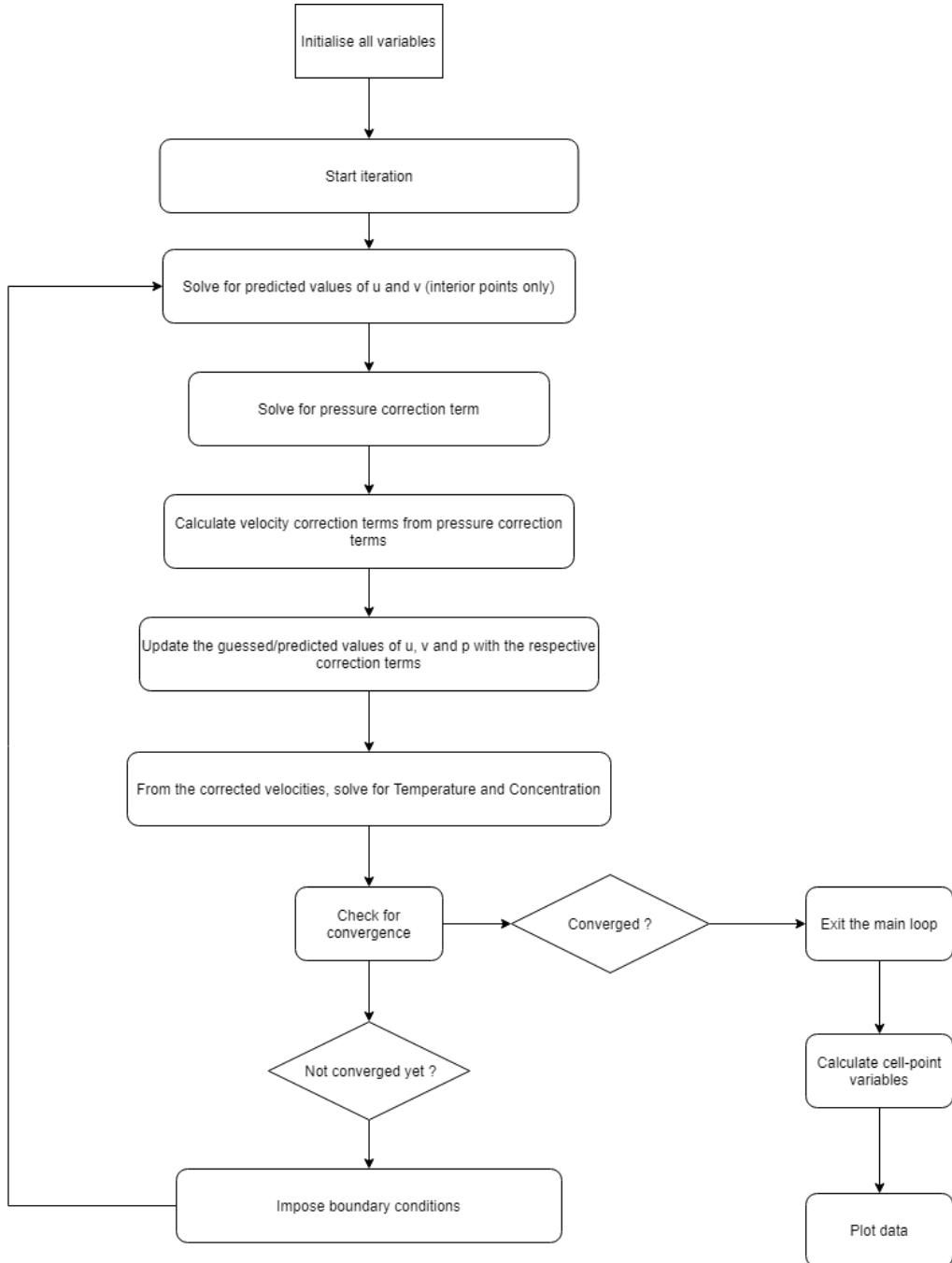


Figure 12: Flow chart of the code discussed in the previous section

chart, a few hand calculations were carried out for 5 nodes or 4 cells. Values of u , v , p , T and C are initialised as 0.0, 0.0, 0.5 and 1.0 for all staggered grid points. The subsequent calculations for 3 iterations are shown below :

- **Iteration 1** The convective and diffusive coefficients of each node are first calculated as :

$$\begin{aligned}
a_e^C &= \Delta y \frac{u_{2+1,2}^{k-1} + u_{2,2}^{k-1}}{2Pr} = 0 \\
a_w^C &= \Delta y \frac{u_{2,2}^{k-1} + u_{2-1,2}^{k-1}}{2Pr} = 0 \\
a_n^C &= \Delta y \frac{v_{2+1,2}^{k-1} + v_{2,2}^{k-1}}{Pr} = 0 \\
a_s^C &= \Delta y \frac{u_{2+1,2-1}^{k-1} + u_{2,2-1}^{k-1}}{2Pr} = 0 \\
a_e^D &= a_w^D = \frac{\Delta y}{\Delta x} = 1 \\
a_n^D &= a_s^D = \frac{\Delta x}{\Delta y} = 1
\end{aligned} \tag{69}$$

By the hybrid scheme formula :

$$\begin{aligned}
a^E &= \max\{-a_e^C, a_e^D - 0.5a_e^C, 0\} = \max\{0, 1, 0\} = 0 \\
a^W &= \max\{-a_w^C, a_w^D - 0.5a_w^C, 0\} = \max\{0, 1, 0\} = 0 \\
a^N &= \max\{-a_n^C, a_n^D - 0.5a_n^C, 0\} = \max\{0, 1, 0\} = 0 \\
a^S &= \max\{-a_s^C, a_s^D - 0.5a_s^C, 0\} = \max\{0, 1, 0\} = 0 \\
a^P &= \frac{\sum a_{nb} + \sum (-1)^n a_{nb}^C}{\alpha_V} = 13.333
\end{aligned} \tag{70}$$

Similar calculations are carried out for v^* as well. The coefficient matrices of v^* are given as :

$$a_{nb}^v = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad a_P^v = \sum a_{nb}^v - \sum a_{nb}^C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 13.333 & 13.333 & 13.333 & 0 \\ 0 & 13.333 & 13.333 & 13.333 & 0 \\ 0 & 13.333 & 13.333 & 13.333 & 0 \\ 0 & 13.333 & 13.333 & 13.333 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{71}$$

Solving for v using under relaxation, we get :

$$v_{i,j}^* = \left(\frac{b^v - \sum a_{nb} v_{nb}^{k-1} + (1-\alpha) a_P v_{i,j}^{k-1}}{a'_P} \right) \tag{72}$$

we have :

$$v^* = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 28.125 & 30.234 & 30.392 & 0 \\ 0 & 30.234 & 32.660 & 32.853 & 0 \\ 0 & 30.392 & 32.853 & 33.053 & 0 \\ 0 & 30.404 & 32.869 & 33.069 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{73}$$

From here, the pressure correction equation is solved. Coefficients are calculated as

$$a_E^P = \frac{dy^2}{a_{P,i,j}^u}, \quad a_W^P = \frac{dy^2}{a_{P,i-1,j}^u}, \quad a_N^P = \frac{dx^2}{a_{P,i,j}^u}, \quad a_S^P = \frac{dx^2}{a_{P,i,j-1}^u}, \quad a_P^P = \sum a_{nb}^P \tag{74}$$

The coefficient matrices at each node is given as :

$$a_E^P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad a_W^P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{75}$$

$$a_N^P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0 & 0 \\ 0 & 0.004 & 0.004 & 0.004 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, a_S^P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0 & 0.004 & 0.004 & 0.004 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (76)$$

$$a_P^P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.009 & 0.014 & 0.014 & 0.009 & 0 \\ 0 & 0.014 & 0.018 & 0.018 & 0.014 & 0 \\ 0 & 0.014 & 0.018 & 0.018 & 0.014 & 0 \\ 0 & 0.009 & 0.014 & 0.014 & 0.009 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, b^P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -7.031 & -0.527 & -0.039 & 7.598 & 0 \\ 0 & -7.558 & -0.606 & -0.048 & 8.213 & 0 \\ 0 & -7.598 & -0.615 & -0.049 & 8.263 & 0 \\ 0 & -7.601 & -0.616 & -0.049 & 8.267 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (77)$$

Solving equation of pressure, using under-relaxation, we have :

$$p'_{i,j} = \frac{\sum a_{nb}^P p'_{nb} + b^P}{a_{i,j}^P} \quad (78)$$

Thus, for interior points, pressure correction matrix equals :

$$p'_{i,j} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -525 & -201.250 & -69.052 & 532.802 & 0 \\ 0 & -551.250 & -210.765 & -71.763 & 562.528 & 0 \\ 0 & -561.968 & -216.156 & -73.838 & 574.224 & 0 \\ 0 & -848.533 & -385.571 & -155.623 & 826.591 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (79)$$

From the pressure correction term velocity correction terms are derived as :

$$u'_{i,j} = \frac{\Delta y}{a_P} (p'_{i,j} - p'_{i+1,j}) \quad (80)$$

$$v'_{i,j} = \frac{\Delta x}{a_P} (p'_{i,j} - p'_{i,j+1}) \quad (81)$$

For which the velocity correction matrices are given as:

$$u'_{i,j} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.703 & 0.254 & 0.072 & -0.796 & 0 \\ 0 & 0.287 & 0.144 & 0.055 & -0.313 & 0 \\ 0 & 7.675 & 4.537 & 2.190 & -6.759 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, v'_{i,j} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -8.671 & -3.541 & -16.121 & 0 & 0 \\ 0 & -9.120 & -3.723 & -16.989 & 0 & 0 \\ 0 & -9.262 & -3.812 & -17.358 & 0 & 0 \\ 0 & -12.400 & -6.159 & -26.309 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (82)$$

The predicted velocities are now corrected as :

$$u_{i,j}^k = u_{i,j}^{k-1} + u'_{i,j}, v_{i,j}^k = v_{i,j}^{k-1} + v'_{i,j} \quad (83)$$

Which leads to the matrices :

$$u_{i,j}^k = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.703 & 0.254 & 0.072 & -0.796 & 0 \\ 0 & 0.287 & 0.144 & 0.055 & -0.313 & 0 \\ 0 & 7.675 & 4.537 & 2.190 & -6.759 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, v_{i,j}^k = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 19.453 & 26.693 & 14.271 & 0 & 0 \\ 0 & 21.114 & 28.936 & 15.863 & 0 & 0 \\ 0 & 21.129 & 29.041 & 15.694 & 0 & 0 \\ 0 & 18.003 & 26.710 & 6.759 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (84)$$

The above values of u and v are used to solve temperature and concentration variables as :

$$T_{i,j}^k = 0.5 \forall \{i, j\}, \text{ and } C_{i,j}^k = 1.0 \forall \{i, j\} \quad (85)$$

We don't see any appreciable changes in u , T and C , but in the subsequent iterations, the solution varies. At each step, boundary conditions are imposed on the corected variables, and after some 200 iterations, convergence for 5 nodes is reached.

10 Appendix-IV : Some variations of the original problem

In this section, we discuss about various situations of double diffusive convection like change in geometry, solver algorithm, etc. It is important to carry out such simulations in order to check the versatility of the original code and its extendability to modelling more complex problems. A range of different situations were studied and implemented into the code which are discussed further.

10.1 Double diffusive convection in a triangular cavity

Unlike the original problem, a three sided enclosure is considered for calculation of flow variables. Due to irregularity of the 3rd side, implementation of boundary conditions becomes more complicated. Two cases of different boundary temperatures are studied which is illustrated below. The domain is meshed into a cartesian grid, however the mesh is set such

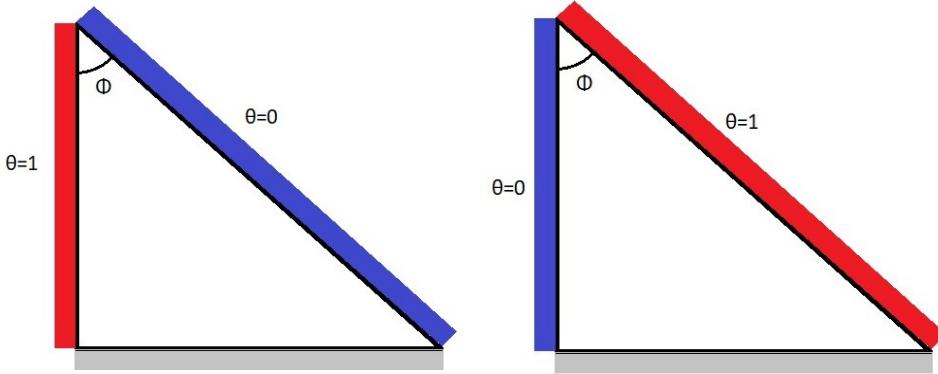


Figure 13: Schematic of problem description

that the inclined side of the triangle lies along the diagonal of the grid points, i.e. no. of grid points or cells is considered same in both x and y directions, such that Δx and Δy vary according to the length of domain in respective directions. The mesh is shown below for a better understanding of the problem. The boundary conditions of the problem are :

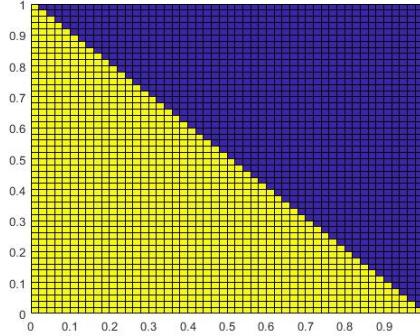


Figure 14: Discretized domain for $\phi = 45^\circ$. Part in yellow denotes fluid domain while purple region houses the void or solid domain

- Bottom wall

$$\frac{\partial u}{\partial y} = 0, v = 0, p = 0, \frac{\partial T}{\partial y} = 0, \frac{\partial C}{\partial y} = 0 \quad (86)$$

- Side wall

$$u = 0, \frac{\partial v}{\partial x} = 0, T = 1, C = 1, \frac{\partial p}{\partial x} = 0 \quad (87)$$

- Inclined wall

$$\frac{\partial u}{\partial y} = 0, \frac{\partial v}{\partial x} = 0, T = 0, C = 0, p = 0 \quad (88)$$

Results for case (1) of the problem definition are given as :

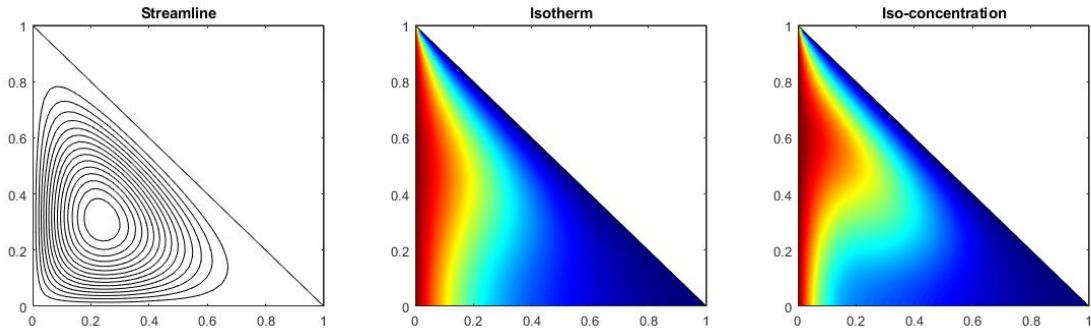


Figure 15: $Ra_T = 2000$

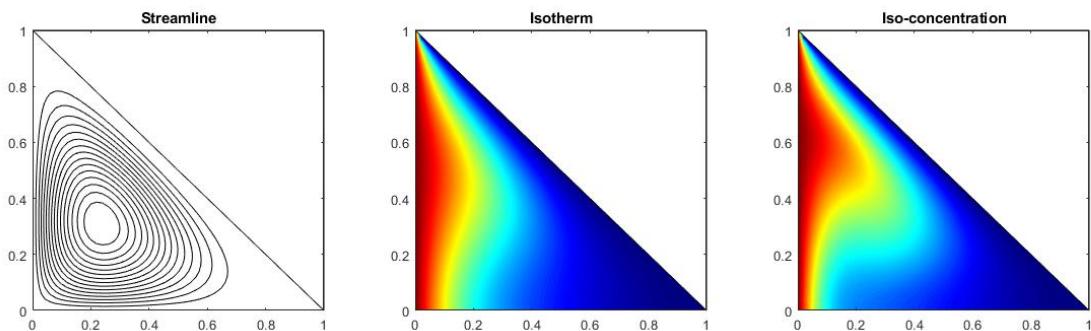


Figure 16: $Ra_T = 2500$

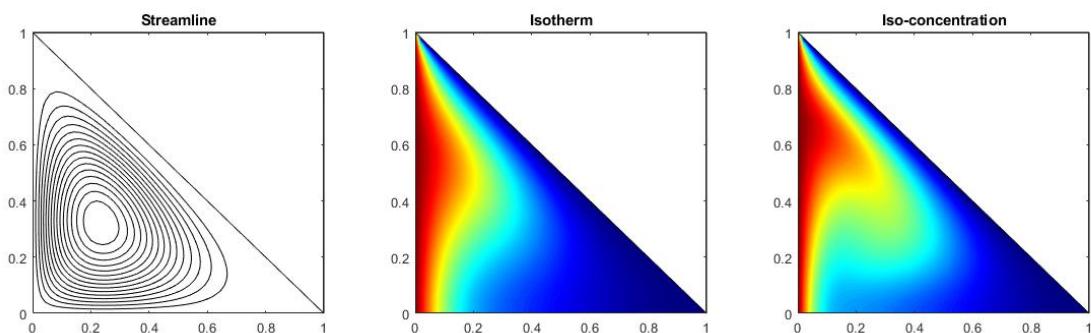


Figure 17: $Ra_T = 5000$

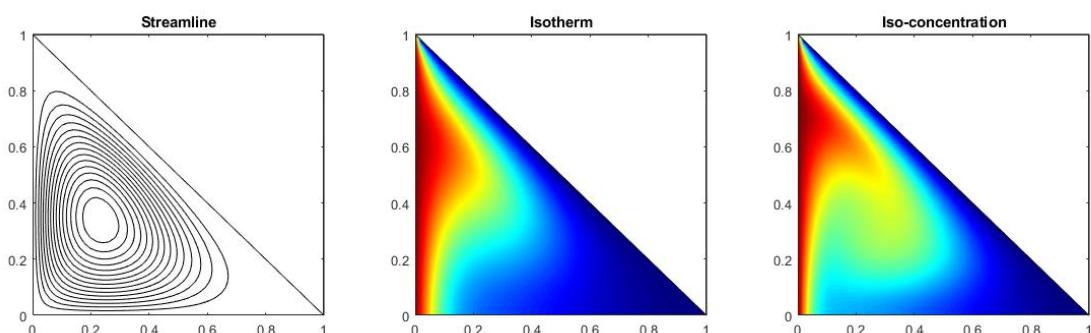


Figure 18: $Ra_T = 10000$

10.2 Double diffusive convection with effect of a conductive wall

In this case, the right wall is replaced by a conducting solid domain of finite length. Here convection and conduction are coupled as, both have effect on each other at the solid-fluid interface. The problem is graphically represented below: The

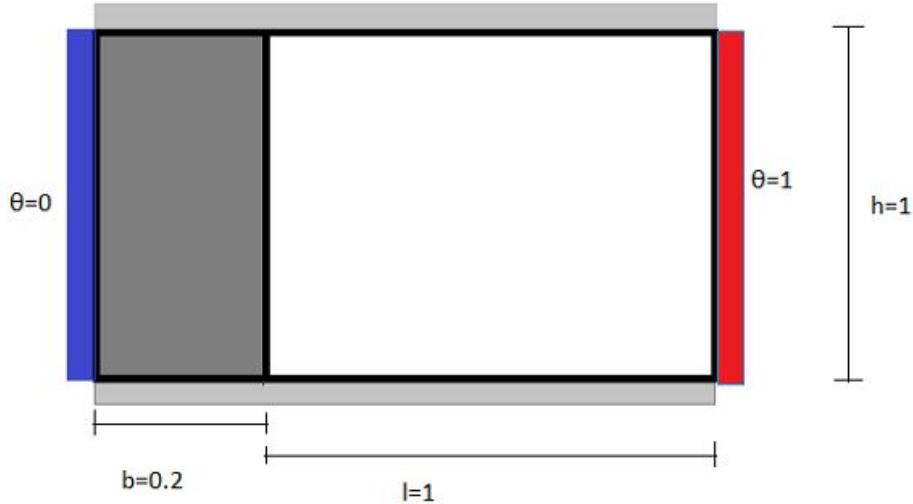


Figure 19: Schematic representation of the problem of interest

mesh is set such that the domain form $x \in (0, 0.2)$ is the solid body, while the rest is fluid. Inside the solid part the velocities are set as 0. This is described pictorially as : Boundary conditions of the problem are given as :

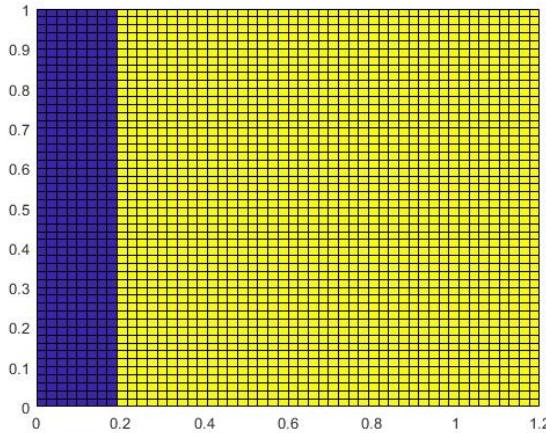


Figure 20: Computational domain

- top and bottom walls

$$v = 0, \frac{\partial \phi}{\partial y} = 0, \phi \in \{u, T, C\} \quad (89)$$

- left wall

$$u = 0, \frac{\partial v}{\partial x} = 0, T = C = 1 \quad (90)$$

- right wall

$$\text{at } x = 0, u = v = 0, T = C = 1; \text{ at } x = 0.2, u = 0, \frac{\partial v}{\partial x} = 0 \quad (91)$$

Results for Rayleigh numbers 2000 – 10000 are produced for $Pr = 1.0$ and displayed below :

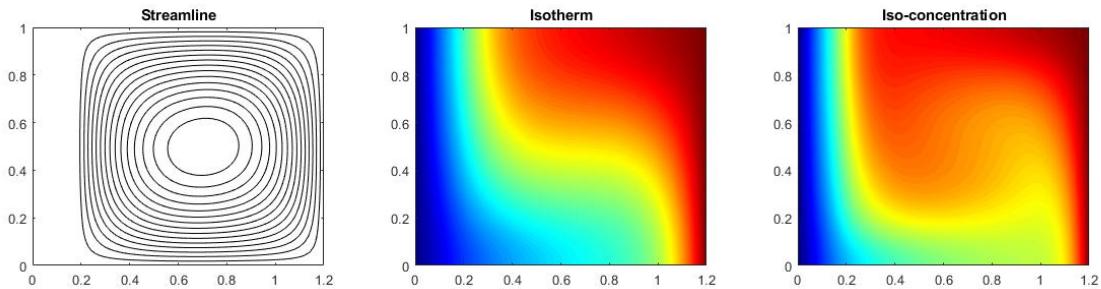


Figure 21: $Ra_T = 2000$

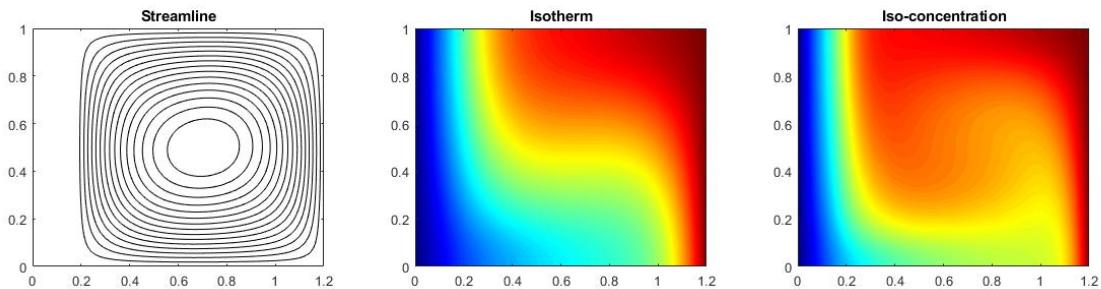


Figure 22: $Ra_T = 2500$

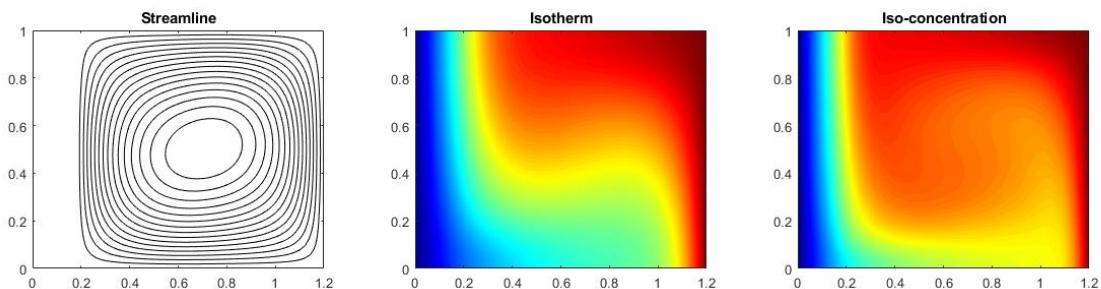


Figure 23: $Ra_T = 5000$

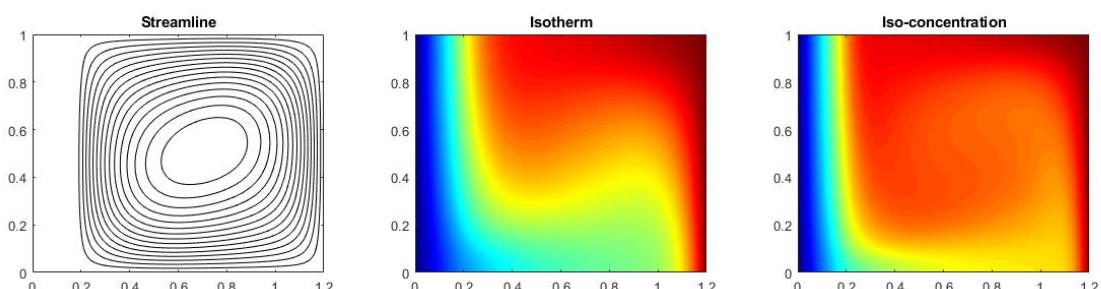


Figure 24: $Ra_T = 10000$

10.3 Double diffusive convection with variable wall temperature

In this problem, the bottom wall is subjected to segmented heating and salting. This leads to many interesting flow patterns at different Rayleigh numbers. The boundary conditions of the problem are the same, except for the temperature and

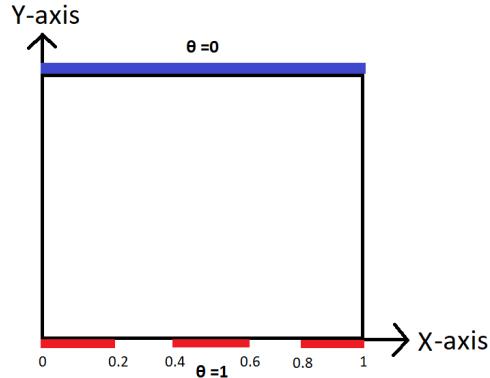


Figure 25: Schematic of the problem

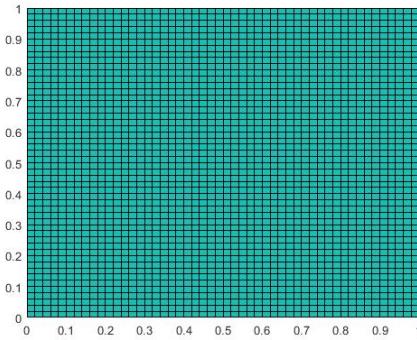


Figure 26: Computational domain

concentrations at the lower wall are segmented as shown in figure (25). This is implemented in the code as :

```
%constant wall temperature
for i=2:nx+1
    if(i>0 && i<=round(0.2/dx))
        T_k_minus_1(i,1)=2-T_k_minus_1(i,2);%bottom wall
    end
    if(i>round(0.2/dx) && i<=round(0.4/dx))
        T_k_minus_1(i,1)=T_k_minus_1(i,2);%bottom wall
    end
    if(i>round(0.4/dx) && i<=round(0.6/dx))
        T_k_minus_1(i,1)=2-T_k_minus_1(i,2);%bottom wall
    end
    if(i>round(0.6/dx) && i<=round(0.8/dx))
        T_k_minus_1(i,1)=T_k_minus_1(i,2);%bottom wall
    end
    if(i>round(0.8/dx) && i<=nx+1)
        T_k_minus_1(i,1)=2-T_k_minus_1(i,2);%bottom wall
    end
    T_k_minus_1(i,ny+2)=-T_k_minus_1(i,ny+1);%top wall
end
```

The same is done for concentration as well. Results of the simulation for Rayleigh numbers 2000 – 10000 are produced and presented below :

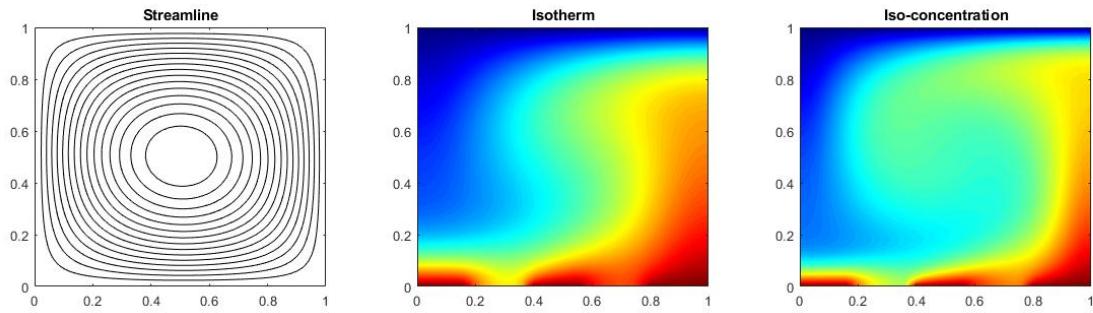


Figure 27: $Ra_T = 2000$

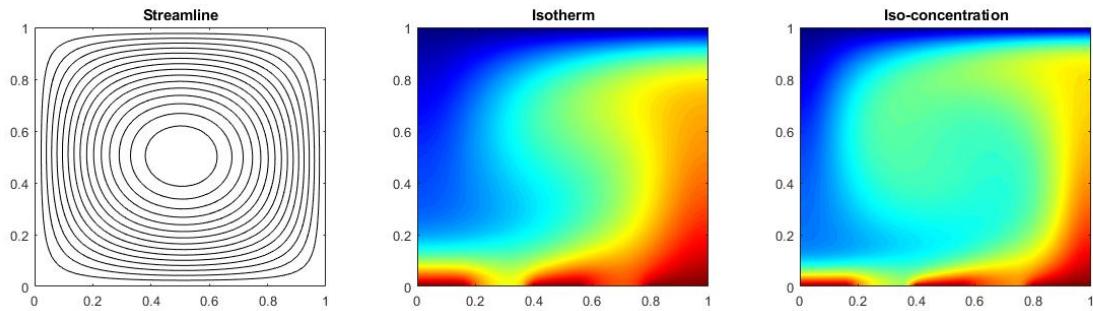


Figure 28: $Ra_T = 2500$

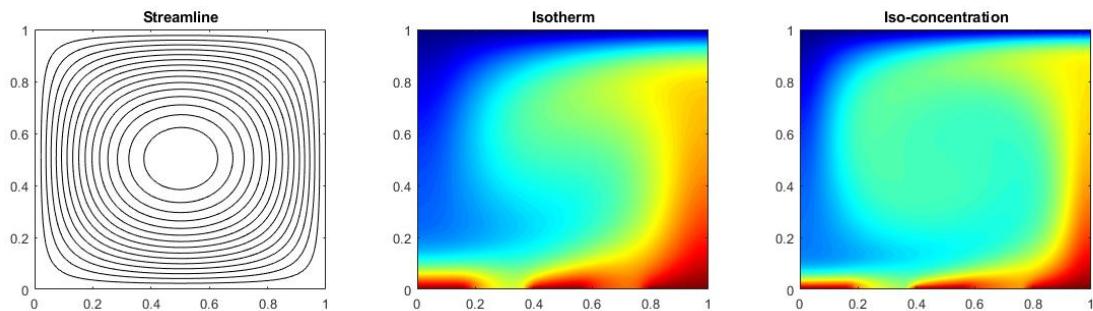


Figure 29: $Ra_T = 5000$

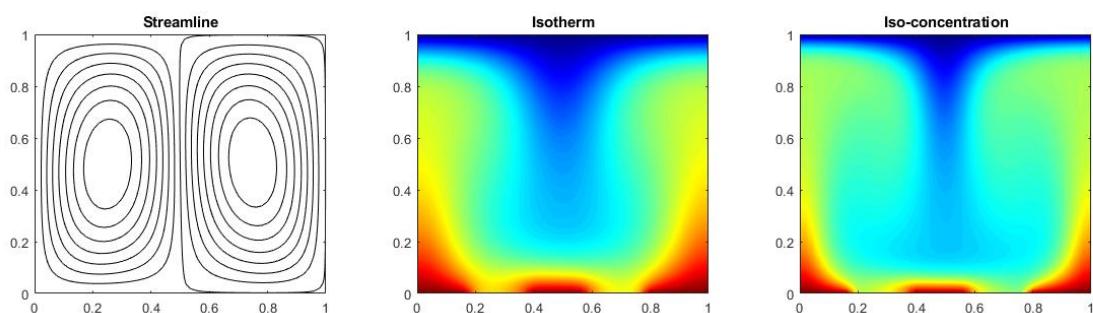


Figure 30: $Ra_T = 10000$

11 Appendix-V : An alternative method of solving convection diffusion problems - The stream function and vorticity approach

Vorticity-Stream Function approach to two-dimensional problem of solving Navier-Stokes equations is rather easy. A different form of equations can be scary at the beginning but, mathematically, we have only two variables which have to be obtained during computations: stream vorticity vector ω and stream function ψ . The vorticity-streamfunction equations present some advantages over the velocity-pressure equations in the case of two-dimensional flows in simply connected domains. These advantages are well known: (1) the velocity field is automatically divergence-free, (2) the mathematical properties of the equations permit the construction of simple and robust solution methods, (3) computing time is saved because of the smaller number of equations. The governing equations of the flow can be easily derived from equations (1)-(5).

We start by differentiating equation (2) w.r.t y ,

$$\begin{aligned} \frac{\partial}{\partial y} \left(\frac{1}{Pr} \left(u' \frac{\partial u'}{\partial x'} + v' \frac{\partial u'}{\partial y'} \right) \right) &= \frac{\partial}{\partial y} \left(-\frac{\partial p'}{\partial x'} + \left(\frac{\partial^2 u'}{\partial x'^2} + \frac{\partial^2 u'}{\partial y'^2} \right) \right) \\ \Rightarrow \frac{1}{Pr} \left(\frac{\partial u'}{\partial y'} \frac{\partial u'}{\partial x'} + u' \frac{\partial^2 u'}{\partial x' \partial y'} + \frac{\partial v}{\partial y} \frac{\partial u'}{\partial y'} + v' \frac{\partial^2 u'}{\partial y'^2} \right) &= -\frac{\partial^2 p'}{\partial x' \partial y'} + \frac{\partial^3 u'}{\partial y' \partial x'^2} + \frac{\partial^3 u'}{\partial y'^3} \end{aligned} \quad (92)$$

and equation (3) w.r.t x

$$\begin{aligned} \frac{\partial}{\partial x'} \left(\frac{1}{Pr} \left(u' \frac{\partial v'}{\partial x'} + v' \frac{\partial v'}{\partial y'} \right) - Ra_T T' - Ra_S C' \right) &= \frac{\partial}{\partial x} \left(-\frac{\partial p'}{\partial y'} + \left(\frac{\partial^2 v'}{\partial x'^2} + \frac{\partial^2 v'}{\partial y'^2} \right) \right) \\ \Rightarrow \frac{1}{Pr} \left(\frac{\partial u'}{\partial x'} \frac{\partial v'}{\partial x'} + u' \frac{\partial v'}{\partial x'^2} + \frac{\partial v'}{\partial x'} \frac{\partial v'}{\partial y'} + v' \frac{\partial^2 v'}{\partial x' \partial y'} \right) - Ra_T \frac{\partial T'}{\partial x'} - Ra_S \frac{\partial C'}{\partial x'} &= -\frac{\partial^2 p'}{\partial y' \partial x'} + \frac{\partial^3 v'}{\partial x'^3} + \frac{\partial^3 v'}{\partial x' \partial y'^2} \end{aligned} \quad (93)$$

Subtracting equation (69) from (70), we get :

$$\begin{aligned} \frac{1}{Pr} \left(\frac{\partial u'}{\partial x'} \frac{\partial v'}{\partial x'} + u' \frac{\partial v'}{\partial x'^2} + \frac{\partial v'}{\partial x'} \frac{\partial v'}{\partial y'} + v' \frac{\partial^2 v'}{\partial x' \partial y'} - \frac{\partial u'}{\partial y'} \frac{\partial u'}{\partial x'} - u' \frac{\partial^2 u'}{\partial x' \partial y'} - \frac{\partial v}{\partial y} \frac{\partial u'}{\partial y'} - v' \frac{\partial^2 u'}{\partial y'^2} \right) \\ = Ra_T \frac{\partial T'}{\partial x'} + Ra_S \frac{\partial C'}{\partial x'} + \frac{\partial^3 v'}{\partial x'^3} + \frac{\partial^3 v'}{\partial x' \partial y'^2} - \frac{\partial^3 u'}{\partial y' \partial x'^2} - \frac{\partial^3 u'}{\partial y'^3} \\ \Rightarrow \frac{1}{Pr} \left(\frac{\partial u'}{\partial x'} \left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} \right) + \frac{\partial v'}{\partial y'} \left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} \right) + u' \frac{\partial}{\partial x'} \left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} \right) + v' \frac{\partial}{\partial y'} \left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} \right) \right) \\ = Ra_T \frac{\partial T'}{\partial x'} + Ra_S \frac{\partial C'}{\partial x'} + \frac{\partial^2}{\partial x'^2} \left(\frac{\partial v'}{\partial y'} - \frac{\partial u'}{\partial y'} \right) + \frac{\partial^2}{\partial y'^2} \left(\frac{\partial v'}{\partial y'} - \frac{\partial u'}{\partial y'} \right) \\ \Rightarrow \frac{1}{Pr} \left(\left(\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} \right) \left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} \right) + u' \frac{\partial}{\partial x'} \left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} \right) + v' \frac{\partial}{\partial y'} \left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} \right) \right) \\ = Ra_T \frac{\partial T'}{\partial x'} + Ra_S \frac{\partial C'}{\partial x'} + \frac{\partial^2}{\partial x'^2} \left(\frac{\partial v'}{\partial y'} - \frac{\partial u'}{\partial y'} \right) + \frac{\partial^2}{\partial y'^2} \left(\frac{\partial v'}{\partial y'} - \frac{\partial u'}{\partial y'} \right) \end{aligned} \quad (94)$$

From equation (1), cosidering $\nabla \cdot \mathbf{V} = 0$ and $\omega = \nabla \times \mathbf{V}$, we get

$$\frac{1}{Pr} \left(u' \frac{\partial \omega}{\partial x'} + v' \frac{\partial \omega}{\partial y'} \right) = Ra_T \frac{\partial T'}{\partial x'} + Ra_S \frac{\partial C'}{\partial y'} + \frac{\partial^2 \omega}{\partial x'^2} + \frac{\partial^2 \omega}{\partial y'^2} \quad (95)$$

Considering the stream function ψ to be defined as

$$u' = \frac{\partial \psi}{\partial y'} \text{ and } v' = -\frac{\partial \psi}{\partial x'} \quad (96)$$

from the equation of vorticity, we've

$$\omega = \frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'} = \frac{\partial}{\partial x'} \left(-\frac{\partial \psi}{\partial x'} \right) - \frac{\partial}{\partial y'} \left(\frac{\partial \psi}{\partial y'} \right) = -\frac{\partial^2 \psi}{\partial x'^2} - \frac{\partial^2 \psi}{\partial y'^2} \therefore \frac{\partial^2 \psi}{\partial x'^2} + \frac{\partial^2 \psi}{\partial y'^2} = -\omega \quad (97)$$

Thus equations (72) and (74), alongwith equations (4) and (5) are used to solve fluid flow equations. Unlike equations (2) and (3) we don't have a pressure term in the above equations, for which the use of a predictor corrector stepping can be ruled out. Thus, this formulation is actually **simple** unlike the SIMPLE algorithm. The above equations were discretised using a central difference scheme of Finite Difference Method and the use of under-relaxation is retained.

11.1 Numerical Method

Equation (74) can be discretised using a central difference as :

$$\begin{aligned}
& \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta x^2} + \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\Delta y^2} = -\omega_{i,j} \\
\implies & \left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} \right) \psi_{i,j} = \frac{\psi_{i+1,j} + \psi_{i-1,j}}{\Delta x^2} + \frac{\psi_{i,j+1} + \psi_{i,j-1}}{\Delta y^2} + \omega_{i,j} \\
\implies & 2 \left(\frac{\Delta x^2 + \Delta y^2}{\Delta x^2 \Delta y^2} \right) \psi_{i,j} = \frac{(\psi_{i+1,j} + \psi_{i-1,j}) \Delta y^2 + (\psi_{i,j+1} + \psi_{i,j-1}) \Delta x^2 + \omega_{i,j} \Delta x^2 \Delta y^2}{\Delta x^2 \Delta y^2} \\
\implies & 2 (\Delta x^2 + \Delta y^2) \psi_{i,j} = \Delta y^2 \psi_{i+1,j} + \Delta y^2 \psi_{i-1,j} + \Delta x^2 \psi_{i,j+1} + \Delta x^2 \psi_{i,j-1} + \Delta x^2 \Delta y^2 \omega_{i,j} \\
\implies & a_P \psi_{i,j} = a_E \psi_{i+1,j} + a_W \psi_{i-1,j} + a_N \psi_{i,j+1} + a_S \psi_{i,j-1} + b^\psi
\end{aligned} \tag{98}$$

From equation (72) each term is dealt separately as,

$$\frac{1}{Pr} u \frac{\partial \omega}{\partial x} = \frac{1}{Pr} u_{i,j} \frac{\omega_{i+1,j} - \omega_{i-1,j}}{2 \Delta x} \tag{99}$$

$$\frac{1}{Pr} v \frac{\partial \omega}{\partial y} = \frac{1}{Pr} u_{i,j} \frac{\omega_{i+1,j} - \omega_{i-1,j}}{2 \Delta y} \tag{100}$$

$$Ra_T \frac{\partial T}{\partial x} = Ra_T \frac{T_{i+1,j} - T_{i-1,j}}{2 \Delta x} \tag{101}$$

$$Ra_S \frac{\partial C}{\partial x} = Ra_S \frac{C_{i+1,j} - C_{i-1,j}}{2 \Delta x} \tag{102}$$

$$\frac{\partial^2 \omega}{\partial x^2} = \frac{\omega_{i+1,j} - 2\omega_{i,j} + \omega_{i-1,j}}{\Delta x^2} \tag{103}$$

$$\frac{\partial^2 \omega}{\partial y^2} = \frac{\omega_{i,j+1} - 2\omega_{i,j} + \omega_{i,j-1}}{\Delta y^2} \tag{104}$$

Thus, from equation (72), we have

$$\begin{aligned}
& \frac{1}{Pr} \left(u_{i,j} \left(\frac{\omega_{i+1,j} - \omega_{i-1,j}}{2 \Delta x} \right) + v_{i,j} \left(\frac{\omega_{i,j+1} - \omega_{i,j-1}}{2 \Delta y} \right) \right) \\
= & Ra_T \left(\frac{T_{i+1,j} - T_{i-1,j}}{2 \Delta x} \right) + Ra_S \left(\frac{C_{i+1,j} - C_{i-1,j}}{2 \Delta x} \right) + \frac{\omega_{i+1,j} - 2\omega_{i,j} + \omega_{i-1,j}}{\Delta x^2} + \frac{\omega_{i,j+1} - 2\omega_{i,j} + \omega_{i,j-1}}{\Delta y^2}
\end{aligned} \tag{105}$$

A transient model with a very small time-step is used to ensure stability throughout the iteration, which is given as

$$\frac{1}{Pr} \left(\frac{\partial \omega}{\partial t'} + u' \frac{\partial \omega}{\partial x'} + v' \frac{\partial \omega}{\partial y'} \right) = Ra_T \frac{\partial T'}{\partial x'} + Ra_S \frac{\partial C'}{\partial x'} + \frac{\partial^2 \omega}{\partial x'^2} + \frac{\partial^2 \omega}{\partial y'^2} \tag{106}$$

The transient term can be discretised as :

$$\frac{\partial \omega}{\partial t'} = \frac{\omega_{i,j}^{n+1} - \omega_{i,j}^n}{\Delta t} \tag{107}$$

Thus, the discretised equation of vorticity is given as :

$$\begin{aligned}
\omega_{i,j}^{n+1} = & \omega_{i,j}^n + \Delta t \left(-u_{i,j} \left(\frac{\omega_{i+1,j}^n - \omega_{i-1,j}^n}{2 \Delta x} \right) Pr - v_{i,j} \left(\frac{\omega_{i,j+1}^n - \omega_{i,j-1}^n}{2 \Delta y} \right) Pr \right. \\
& \left. + Ra_T \left(\frac{T_{i+1,j}^n - T_{i-1,j}^n}{2 \Delta x} \right) + Ra_S \left(\frac{C_{i+1,j}^n - C_{i-1,j}^n}{2 \Delta x} \right) \right. \\
& \left. + \frac{\omega_{i+1,j}^n - 2\omega_{i,j}^n + \omega_{i-1,j}^n}{\Delta x^2} + \frac{\omega_{i,j+1}^n - 2\omega_{i,j}^n + \omega_{i,j-1}^n}{\Delta y^2} \right)
\end{aligned} \tag{108}$$

The same iteration scheme is used for temperature and concentration equations as well, i.e. :

$$\begin{aligned}
T_{i,j}^{n+1} = & T_{i,j}^n + \Delta t \left(-u_{i,j} \left(\frac{T_{i+1,j}^n - T_{i-1,j}^n}{2 \Delta x} \right) Pr - v_{i,j} \left(\frac{T_{i,j+1}^n - T_{i,j-1}^n}{2 \Delta y} \right) Pr \right. \\
& \left. + \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{\Delta x^2} + \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{\Delta y^2} \right)
\end{aligned} \tag{109}$$

$$C_{i,j}^{n+1} = C_{i,j}^n + \Delta t \left(-u_{i,j} \left(\frac{C_{i+1,j}^n - C_{i-1,j}^n}{2\Delta x} \right) Pr - v_{i,j} \left(\frac{C_{i,j+1}^n - C_{i,j-1}^n}{2\Delta y} \right) Pr \right. \\ \left. + \tau \left(\frac{C_{i+1,j}^n - 2C_{i,j}^n + C_{i-1,j}^n}{\Delta x^2} + \frac{C_{i,j+1}^n - 2C_{i,j}^n + C_{i,j-1}^n}{\Delta y^2} \right) \right) \quad (110)$$

If the convective-diffusive terms on the right hand sides of equations (85)-(87) are aliased as $RHS_{i,j}$, they can be summarised as :

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n + \Delta t \times RHS_{i,j} \quad (111)$$

Equation of stream function is solved using over-relaxation as discussed earlier. The boundary conditions need to be implemented in accordance with the original problem.

11.2 Boundary Condition

Along the domain boundaries/walls, as there is no flow accross, they can be considered as streamlines themselves. Thus, along the walls, stream function is set to 0, i.e.

$$\psi_{1,j} = \psi_{n,j} = \psi_{i,1} = \psi_{i,n} = 0 \quad (112)$$

Velocities at the boundary are extrapolated from thee interior points as :

$$\left(\frac{\partial u}{\partial y} \right)_{1,j} = \frac{u_{i,j} - u_{i,j-1}}{\Delta y} = 0 \\ \implies u_{i,j} = u_{i,j-1} \quad (113)$$

Boundary conditions for vorticity are derived as :

$$\omega = -\frac{\partial^2 \psi}{\partial n^2} \quad (114)$$

where n is the outward normal of a surface. If we consider a wall, where the outwar normal is \hat{x} , using Taylor series expansion for ψ we have :

$$\psi_{i+1,j} = \psi_{i,j} + \left(\frac{\partial \psi}{\partial x} \right)_{i,j} \Delta x + \left(\frac{\partial^2 \psi}{\partial x^2} \right)_{i,j} \frac{\Delta x^2}{2} + \\ \implies \frac{2(\psi_{i+1,j} - \psi_{i,j})}{\Delta x} - 2 \left(\frac{\partial \psi}{\partial x} \right)_{i,j} = \left(\frac{\partial^2 \psi}{\partial x^2} \right)_{i,j} \\ \implies \left(\frac{\partial^2 \psi}{\partial x^2} \right)_{i,j} = \frac{2(\psi_{i+1,j} - \psi_{i,j})}{\Delta x} + \frac{2v_{i,j}}{\Delta x} \quad (115)$$

Thus, from equation (19), considering $\psi = 0$ at the wall, we have

$$\omega_{i,j} = \frac{-2\psi_{i+1,j}}{\Delta x} - \frac{2v_{i,j}}{\Delta x} \quad (116)$$

Similarly for a wall with $-\hat{x}$ as outward normal, a backward differencing is carried ut in the taylor series expansion. The boundary conitions at different walls are summarised as :

- Top wall :

$$u_{i,jmax} = u_{i,jmax-1}, v_{i,jmax} = 0, \psi_{i,jmax} = 0, \omega_{i,jmax} = \frac{-2\psi_{i,jmax-1}}{\Delta y^2} - \frac{2u_{i,jmax}}{\Delta y} \quad (117)$$

- Bottom wall :

$$u_{i,jmin} = u_{i,jmin+1}, v_{i,jmin} = 0, \psi_{i,jmin} = 0, \omega_{i,jmin} = \frac{-2\psi_{i,jmin+1}}{\Delta y^2} - \frac{2u_{i,jmin}}{\Delta y} \quad (118)$$

- Right wall :

$$u_{imax,j} = 0, v_{imax,j} = v_{imax-1,j}, \psi_{imax,j} = 0, \omega_{imax,j} = \frac{-2\psi_{imax-1,j}}{\Delta x^2} - \frac{2v_{imax,j}}{\Delta x} \quad (119)$$

- Left wall :

$$u_{imin,j} = 0, v_{imin,j} = v_{imin+1,j}, \psi_{imin,j} = 0, \omega_{imin,j} = \frac{-2\psi_{imin+1,j}}{\Delta x^2} - \frac{2v_{imin,j}}{\Delta x} \quad (120)$$

11.3 Mesh Convergence

Mesh convergence test is carried out for three progressively refined meshes with $\Delta = 1/30, 1/40, 1/50$. From figure 13 it is evident that the measured parameters are converging with mesh refinement. $\Delta = 1/50$ is chosen as the value for carrying out the simulations.

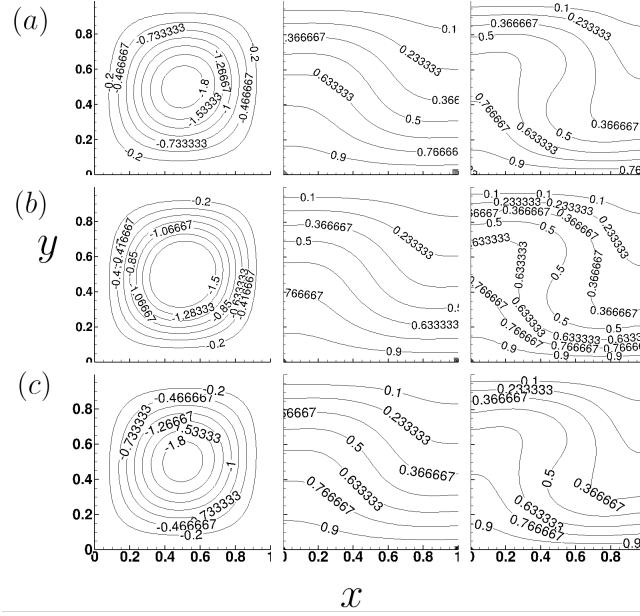


Figure 31: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 2000$, for grid size (a) 30×30 , (b) 40×40 and (c) 50×50 from top to bottom.

11.4 Results and Validation

The results for $Ra_T = 2000$, $Ra_T = 2500$, $Ra_T = 5000$, $Ra_T = 10000$ are shown and compared with the work of Murty.

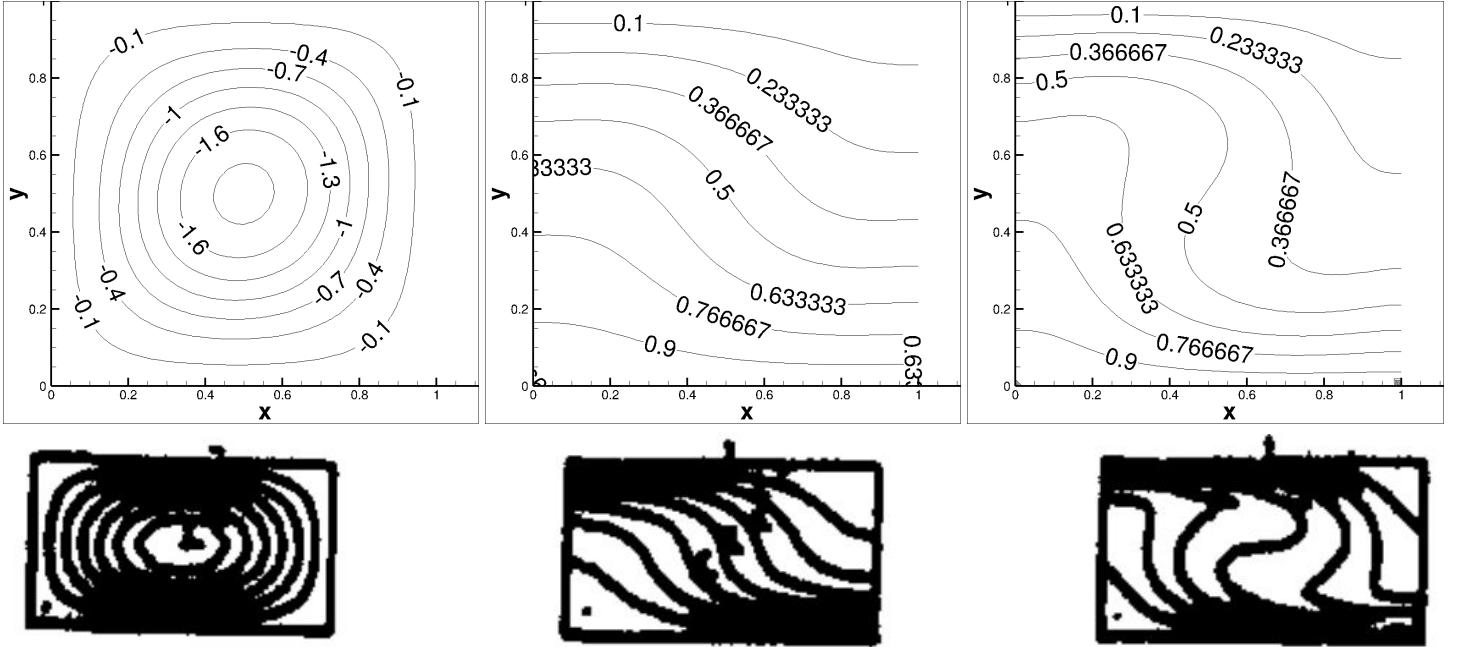


Figure 32: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 2000$ (top row current work) compared with the work of Murty (bottom row).

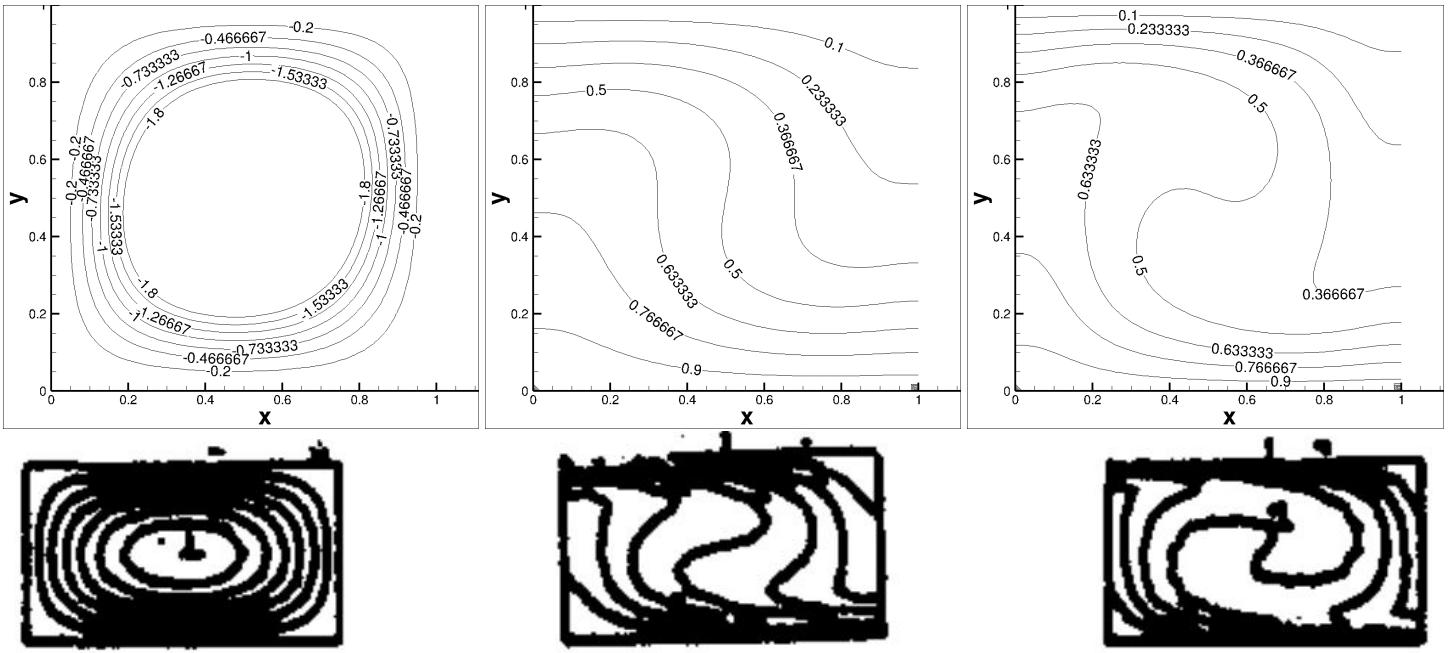


Figure 33: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 5000$ (top row current work) compared with the work of Murty (bottom row).

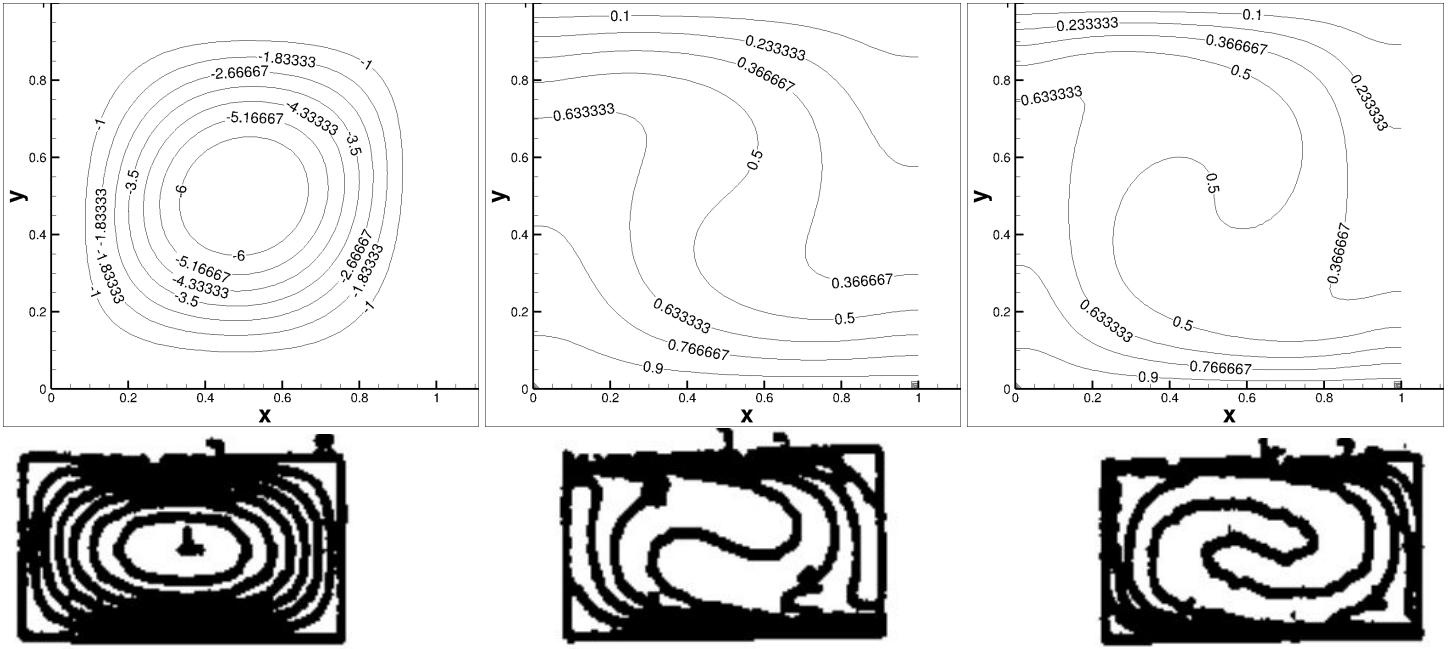


Figure 34: Streamlines, Isotherms and Isoconcentration lines at $Ra_T = 10000$ (top row current work) compared with the work of Murty (bottom row).

12 Appendix VI : Use of Mapped Grids

The main problem with cartesian grid is modelling of complex boundaries. For example, a cartesian grid as used in the original problem, is ideal for rectangular boundaries, however it may not give very accurate results for non-rectangular domains, like the one discussed in section 10.1. Another problem with handling non rectangular boundaries in a cartesian system is the wastage of memory for unused grid points. Such problems can be adressed by downright transforming or mapping the cartesian coordinate system into an appropriate grid system, suitable for the geometry of the problem. An example is demonstrated by repeating the original problem in a parallelogram shaped cavity, whose walls are inclined at

angle θ with the horizontal. The cartesian system $(\zeta - \eta)$ is converted into the $x - y$ system by using bilinear interpolation

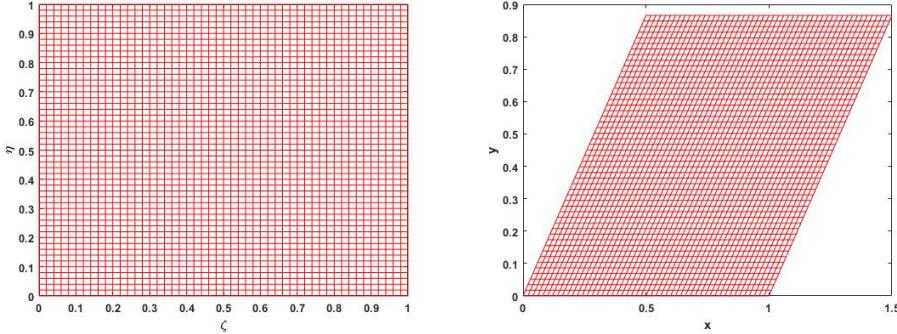


Figure 35: Coordinate mapping for generating appropriate grid

as

$$x(\zeta, \eta) = a_0^x + a_1^x \zeta + a_2^x \eta + a_3^x \zeta \eta \quad (121)$$

$$y(\zeta, \eta) = a_0^y + a_1^y \zeta + a_2^y \eta + a_3^y \zeta \eta \quad (122)$$

In the above equations, the unknown constants are solved using the corresponding domain points as,

- at $(\zeta, \eta) = (0, 0)$,

$$x = 0 \implies a_0^x + (a_1^x \times 0) + (a_2^x \times 0) + (a_3^x \times 0 \times 0) = 0 \therefore a_0^x = 0 \quad (123)$$

$$y = 0 \implies a_0^y + (a_1^y \times 0) + (a_2^y \times 0) + (a_3^y \times 0 \times 0) = 0 \therefore a_0^y = 0 \quad (124)$$

- at $(\zeta, \eta) = (1, 0)$,

$$x = 1 \implies a_0^x + (a_1^x \times 1) + (a_2^x \times 0) + (a_3^x \times 1 \times 0) = 1 \therefore a_1^x = 1 \quad (125)$$

$$y = 0 \implies a_0^y + (a_1^y \times 1) + (a_2^y \times 0) + (a_3^y \times 1 \times 0) = 0 \therefore a_1^y = 0 \quad (126)$$

- at $(\zeta, \eta) = (1, 1)$,

$$x = 1 + \cos(\theta) \implies a_0^x + (a_1^x \times 1) + (a_2^x \times 1) + (a_3^x \times 1 \times 1) = \cos(\theta) \therefore a_2^x + a_3^x = \cos(\theta) \quad (127)$$

$$y = \sin(\theta) \implies a_0^y + (a_1^y \times 1) + (a_2^y \times 1) + (a_3^y \times 1 \times 1) = \sin(\theta) \therefore a_2^y + a_3^y = \sin(\theta) \quad (128)$$

- at $(\zeta, \eta) = (0, 1)$,

$$x = \cos(\theta) \implies a_0^x + (a_1^x \times 0) + (a_2^x \times 1) + (a_3^x \times 0 \times 1) = \cos(\theta) \therefore a_2^x = \cos(\theta) \quad (129)$$

$$y = \sin(\theta) \implies a_0^y + (a_1^y \times 0) + (a_2^y \times 1) + (a_3^y \times 0 \times 1) = \sin(\theta) \therefore a_2^y = \sin(\theta) \quad (130)$$

From equations (127) and (128), we get $a_3^x = a_3^y = 0$. Thus, using the above quations, we get the required transformation as,

$$x(\zeta, \eta) = \zeta + \cos(\theta) \eta \quad (131)$$

$$y(\zeta, \eta) = \sin(\theta) \eta \quad (132)$$

or,

$$\zeta = x - \cot(\theta) y \quad (133)$$

$$\eta = \operatorname{cosec}(\theta) y \quad (134)$$

The real world phenomenon is modelled in the $x - y$ coordinate system, for which it is known as the physical plane. However the solution is done in the cartesian plane or the $\zeta - \eta$ system, for which it is known as the computational plane. So, any differential equation in the $x - y$ system needs to be transformed into the $\zeta - \eta$ system for any further calculations. This is done as,

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial \zeta} \frac{\partial \zeta}{\partial x} + \frac{\partial \phi}{\partial \eta} \frac{\partial \eta}{\partial x} = \frac{\partial \phi}{\partial \zeta} \times 1 + \frac{\partial \phi}{\partial \eta} \times 0 = \frac{\partial \phi}{\partial \zeta} \quad (135)$$

$$\frac{\partial \phi}{\partial y} = \frac{\partial \phi}{\partial \zeta} \frac{\partial \zeta}{\partial y} + \frac{\partial \phi}{\partial \eta} \frac{\partial \eta}{\partial y} = \frac{\partial \phi}{\partial \zeta} \times -\cot(\theta) + \frac{\partial \phi}{\partial \eta} \times \operatorname{cosec}(\theta) = \frac{1}{\sin(\theta)} \left(-\cos(\theta) \frac{\partial \phi}{\partial \zeta} + \frac{\partial \phi}{\partial \eta} \right) \quad (136)$$

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right) = \frac{\partial}{\partial \zeta} \left(\frac{\partial \phi}{\partial \zeta} \right) \frac{\partial \zeta}{\partial x} + \frac{\partial}{\partial \eta} \left(\frac{\partial \phi}{\partial \zeta} \right) \frac{\partial \eta}{\partial x} = \frac{\partial^2 \phi}{\partial \zeta^2} \times 1 + \frac{\partial^2 \phi}{\partial \zeta \partial \eta} \times 0 = \frac{\partial^2 \phi}{\partial \zeta^2} \quad (137)$$

$$\begin{aligned} \frac{\partial^2 \phi}{\partial y^2} &= \frac{\partial}{\partial y} \left(\frac{\partial \phi}{\partial y} \right) \\ &= \frac{\partial}{\partial y} \left(\frac{1}{\sin(\theta)} \left(-\cos(\theta) \frac{\partial \phi}{\partial \zeta} + \frac{\partial \phi}{\partial \eta} \right) \right) \\ &= \frac{1}{\sin(\theta)} \left(\frac{\partial}{\partial \zeta} \left(-\cos(\theta) \frac{\partial \phi}{\partial \zeta} \right) \frac{\partial \zeta}{\partial y} + \frac{\partial}{\partial \eta} \left(-\cos(\theta) \frac{\partial \phi}{\partial \zeta} \right) \frac{\partial \eta}{\partial y} \right) \\ &= \frac{1}{\sin(\theta)} \left(-\cos(\theta) \frac{\partial^2 \phi}{\partial \zeta^2} \times -\cot(\theta) - \cos(\theta) \frac{\partial^2 \phi}{\partial \zeta \partial \eta} \times \operatorname{cosec}(\theta) \right) \\ &= \cot^2(\theta) \frac{\partial^2 \phi}{\partial \zeta^2} - 2\operatorname{cosec}(\theta) \cot(\theta) \frac{\partial^2 \phi}{\partial \zeta \partial \eta} + \operatorname{cosec}^2(\theta) \frac{\partial^2 \phi}{\partial \eta^2} \end{aligned} \quad (138)$$

Thus, the convection-diffusion equations in $x - y$ system are transformed into cartesian form as,

$$\begin{aligned} \frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} &= \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + S(x, y) \\ \Rightarrow \frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial \zeta} - \frac{v}{\sin(\theta)} \left(\cos(\theta) \frac{\partial \phi}{\partial \zeta} - \frac{\partial \phi}{\partial \eta} \right) &= (1 + \cot^2(\theta)) \frac{\partial^2 \phi}{\partial \zeta^2} - 2\operatorname{cosec}(\theta) \cot(\theta) \frac{\partial^2 \phi}{\partial \zeta \partial \eta} + \operatorname{cosec}^2(\theta) \frac{\partial^2 \phi}{\partial \eta^2} + S(\zeta, \eta) \\ \Rightarrow \frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial \zeta} - \frac{v}{\sin(\theta)} \left(\cos(\theta) \frac{\partial \phi}{\partial \zeta} - \frac{\partial \phi}{\partial \eta} \right) &= \operatorname{cosec}^2(\theta) \frac{\partial^2 \phi}{\partial \zeta^2} - 2\operatorname{cosec}(\theta) \cot(\theta) \frac{\partial^2 \phi}{\partial \zeta \partial \eta} + \operatorname{cosec}^2(\theta) \frac{\partial^2 \phi}{\partial \eta^2} + S(\zeta, \eta) \\ \Rightarrow \frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial \zeta} - \frac{v}{\sin(\theta)} \left(\cos(\theta) \frac{\partial \phi}{\partial \zeta} - \frac{\partial \phi}{\partial \eta} \right) &= \operatorname{cosec}^2(\theta) \left(\frac{\partial^2 \phi}{\partial \zeta^2} - 2\cos(\theta) \frac{\partial^2 \phi}{\partial \zeta \partial \eta} + \frac{\partial^2 \phi}{\partial \eta^2} \right) + S(\zeta, \eta) \end{aligned} \quad (139)$$

The stream function vorticity formulation gives a relatively simpler discretisation scheme than the actual navier stokes equations, because of which, $\phi \in \{\omega, T, C\}$, in the above equation. The source term S exists only for vorticity (ω) and is transformed as,

$$S(x, y) = Ra_T \frac{\partial T}{\partial x} + Ra_S \frac{\partial C}{\partial x} \implies S(\zeta, \eta) = Ra_T \frac{\partial T}{\partial \zeta} + Ra_S \frac{\partial C}{\partial \zeta} \quad (140)$$

Similarly the poisson equation of stream function is transformed as,

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \therefore \frac{\partial^2 \psi}{\partial \zeta^2} - 2\cos(\theta) \frac{\partial^2 \psi}{\partial \zeta \partial \eta} + \frac{\partial^2 \psi}{\partial \eta^2} = -\omega \sin^2(\theta) \quad (141)$$

Velocities in the x and y directions are derived as :

$$u(x, y) = \frac{\partial \psi}{\partial y} \implies u(\zeta, \eta) = \frac{1}{\sin(\theta)} \left(-\cos(\theta) \frac{\partial \psi}{\partial \zeta} + \frac{\partial \psi}{\partial \eta} \right) \quad (142)$$

$$v(x, y) = -\frac{\partial \psi}{\partial x} \implies v(\zeta, \eta) = -\frac{\partial \psi}{\partial \zeta} \quad (143)$$

Discretisation scheme and boundary conditions are kept pretty same as discussed in the previous section. The only additional term appering in equation (139) is the mixed derivative, which is derived as,

$$\begin{aligned} \frac{\partial^2 \phi}{\partial \zeta \partial \eta} &= \frac{\partial}{\partial \zeta} \left(\frac{\partial \phi}{\partial \eta} \right) \\ &= \frac{\partial}{\partial \zeta} \left(\frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta \eta} \right) \\ &= \frac{1}{2\Delta \zeta} \left(\frac{\phi_{i+1,j+1} - \phi_{i+1,j-1}}{2\Delta \eta} - \frac{\phi_{i-1,j+1} - \phi_{i-1,j-1}}{2\Delta \eta} \right) \\ &= \frac{\phi_{i+1,j+1} - \phi_{i-1,j+1} - \phi_{i+1,j-1} + \phi_{i-1,j-1}}{4\Delta \zeta \Delta \eta} \end{aligned} \quad (144)$$

Simulations for $\theta = 60^\circ$ at different rayleigh no.s were carried out and results are presented below:

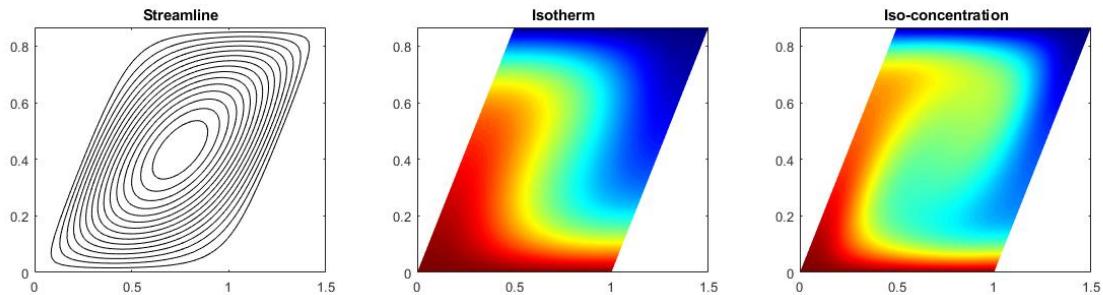


Figure 36: $Ra_T = 2000$

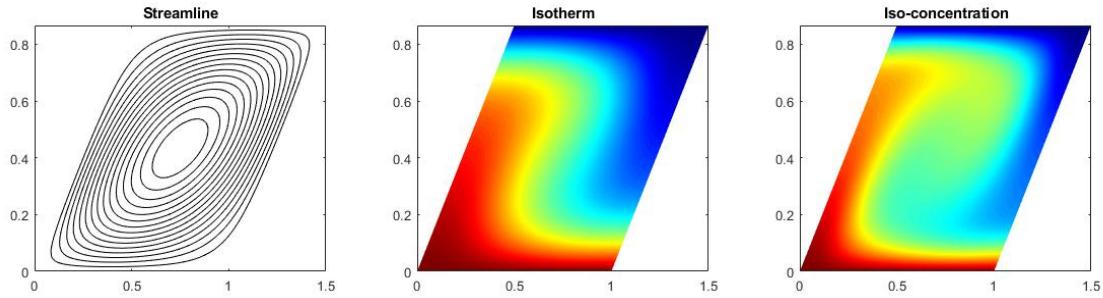


Figure 37: $Ra_T = 2500$

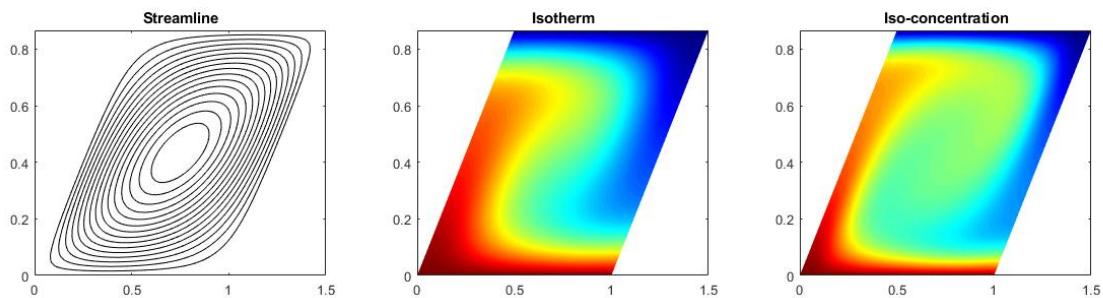


Figure 38: $Ra_T = 5000$

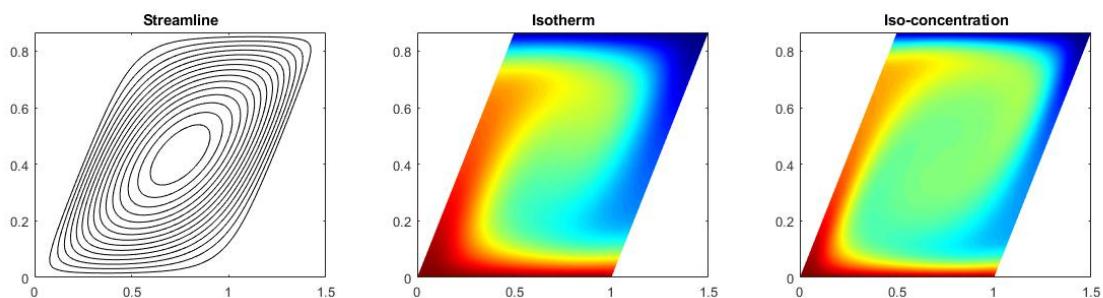


Figure 39: $Ra_T = 10000$