Computational Fluid Dynamics (AE 706)

Computer Assignment 5:

Numerical computation of Incompressible flow inside venturi

By

Parth Nawkar
210010044

DEPARTMENT OF AEROSPACE ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

Apr, 2024

Abstract:

This report presents a numerical computation of incompressible flow inside a venturi using the stream-function equation solved via finite difference methods. The problem involves solving the Laplace equation

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0$$

on a specific grid of size 101×26 generated using transfinite interpolation. The boundary conditions are defined as $\psi = 0$ along the lower wall and $\psi = 100$ along the upper wall, with inlet and outlet boundary conditions consistent with the flow physics.

The numerical scheme utilizes second-order accurate finite difference approximations for the derivatives, employing the Point Gauss Seidel (PGS) method for solving the resulting discretized equations. A convergence criterion of $ERROR < 10^{-5}$ is imposed, where ERROR represents the maximum absolute difference between successive iterations of the stream function ψ.

The report discusses the formulation of the problem, including grid transformation, discretized equations for derivatives, and the matrix generated from the discretization. It also presents the numerical implementation, providing finite difference expressions for each derivative, along with a detailed description of the convergence history and plots of ψ, u, v, and total velocity variations along mid-vertical-lines at specified locations within the venturi.

Finally, observations based on the results are discussed, and the correctness of the numerical outcomes is justified, validating the effectiveness of the applied numerical methods for simulating incompressible flow inside a venturi.

Problem Statement:

Compute incompressible flow inside the venturi (shown in the Figure 1) by

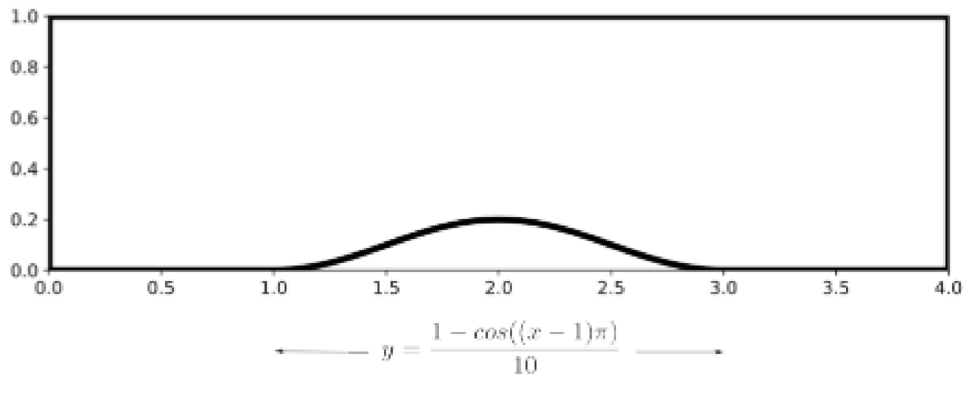


Figure 1: Venturi

numerically solving the stream-function equation,

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0$$

on the grid (101 × 26) as shown in Figure 2. The transfinite interpolation used here for grid generation is the same as that given in question 2 of Quiz2. Here, you need to have the grid orientation as shown in Figure 2.

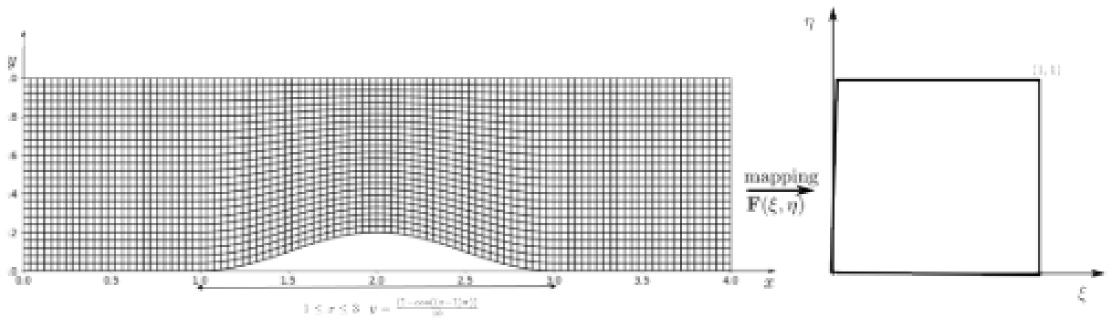


Figure 2: Typical grid inside venturi produced using Transfinite Interpolation

Initial and Boundary Condition:

The boundary conditions are defined as ψ = 0 along the lower wall and ψ = 100 along the upper wall, with inlet and outlet boundary conditions consistent with the flow physics.

To follow the flow physics a linear variation of streamlines has been deployed at left and right boundary

The code can be referred from Transform.py

All other initial Parameters are stored in params.py

Grid Transformation:

Computational grid has been found using **Transfinite Interpolation**

$$P_\xi[F(\eta)] = \phi_0(\xi)F(0,\eta) + \phi_1(\xi)F(1,\eta)$$

$$P_\eta[F(\xi)] = \psi_0(\eta)F(\xi,0) + \psi_1(\eta)F(\xi,1)$$

$$\begin{aligned} P_\xi P_\eta(\mathbf{F}) &= \sum_{i=0}^{1}\sum_{j=0}^{1} \phi_i(\xi)\psi_i(\eta)F(\xi_i,\eta_j) \\ &= \phi_0(\xi)\psi_0(\eta)\mathbf{F}(0,0) + \phi_0(\xi)\psi_1(\eta)\mathbf{F}(0,1) \\ &+ \phi(\xi)\psi_0(\eta)\mathbf{F}(1,0) + \phi_1(\xi)\Psi_1(\eta)\mathbf{F}(1,1) \end{aligned}$$

$$F(0{,}0) = \begin{bmatrix}0\\0\end{bmatrix} \qquad F(1{,}0) = \begin{bmatrix}4\\0\end{bmatrix}$$

$$F(0{,}1) = \begin{bmatrix}0\\1\end{bmatrix} \qquad F(1{,}1) = \begin{bmatrix}4\\1\end{bmatrix}$$

$$F(\xi,0) = \begin{bmatrix}4\xi\\ 0.1\left(1 - cos\left((4\xi-1)\pi\right)\right)\end{bmatrix}$$

$$F(0,\eta) = \begin{bmatrix}0\\ \eta\end{bmatrix}$$

$$F(\xi,1) = \begin{bmatrix}4\xi\\ 1\end{bmatrix}$$

$$F(1,\eta) = \begin{bmatrix}4\\ \eta\end{bmatrix}$$

And $F(\xi,\eta)$ is given by:

$$F(\xi,\eta) = P_\xi\big(F(\eta)\big) + P_\eta\big(F(\xi)\big) - P_\xi P_\eta\big(F(\xi,\eta)\big)$$

Which Results in following transformation function:

$F(\xi,\eta) = \begin{bmatrix}4\xi\\ \eta + (1-\eta)(0.1)(1 - cos(4\xi - 1)\pi)\end{bmatrix}$ for the middle section of venturi

$F(\xi,\eta) = \begin{bmatrix}4\xi\\ \eta\end{bmatrix}$ for the rest of venturi the transformation is given by this function

This transformation function has been coded in Transform.py.

Laplace Equation Transformation to Computation Domain:

$$\frac{\partial^2\phi}{\partial\xi^2}\left[\left(\frac{\partial\xi}{\partial x}\right)^2+\left(\frac{\partial\xi}{\partial y}\right)^2\right]+\frac{\partial^2\phi}{\partial\eta^2}\left[\left(\frac{\partial\eta}{\partial x}\right)^2+\left(\frac{\partial\eta}{\partial y}\right)^2\right]$$
$$+2\frac{\partial^2\phi}{\partial\xi\,\partial\eta}\left[\left(\frac{\partial\eta}{\partial x}\right)\left(\frac{\partial\xi}{\partial x}\right)+\left(\frac{\partial\eta}{\partial y}\right)\left(\frac{\partial\xi}{\partial y}\right)\right]$$
$$+\frac{\partial\phi}{\partial\xi}\left(\frac{\partial^2\xi}{\partial x^2}+\frac{\partial^2\xi}{\partial y^2}\right)+\frac{\partial\phi}{\partial\eta}\left(\frac{\partial^2\eta}{\partial x^2}+\frac{\partial^2\eta}{\partial y^2}\right)=0 \quad (5.17)$$

Following the reference literature we have arrived at this Laplace Equation in Computational Domain.

Metric Terms and Jacobian:

The terms $\frac{\partial\xi}{\partial x}$, $\frac{\partial\xi}{\partial y}$, $\frac{\partial\eta}{\partial x}$ and $\frac{\partial\eta}{\partial y}$ are referred as metric terms and are calculated as following

Since we knew the analytical equation of the grid computing derivatives was simple

$$x = 4\,\xi$$
$$y = \eta + (1-\eta)(0.1)(1-cos(4\xi-1)\pi)$$

We got **for middle section**:

$$\frac{\partial x}{\partial\xi} = 0.25 \qquad \frac{\partial x}{\partial\eta} = 0$$

$$\frac{\partial y}{\partial\xi} = (1-\eta)\left(0.1\pi 4 sin\left((4\xi-1)\pi\right)\right) \qquad \frac{\partial y}{\partial\eta} = 1-0.1*\left(1-cos\left((4\xi-1)\pi\right)\right)$$

For other sections

$$\frac{\partial x}{\partial\xi} = 0.25 \qquad \frac{\partial x}{\partial\eta} = 0$$

$$\frac{\partial y}{\partial\xi} = 0 \qquad \frac{\partial y}{\partial\eta} = 1$$

To get the metric term we need to take inverse of these derivatives using the next steps:

J here is the Jacobian

$$\begin{bmatrix}\frac{\partial\xi}{\partial x} & \frac{\partial\xi}{\partial y}\\ \frac{\partial\eta}{\partial x} & \frac{\partial\eta}{\partial y}\end{bmatrix} = \begin{bmatrix}\frac{\partial x}{\partial\xi} & \frac{\partial x}{\partial\eta}\\ \frac{\partial y}{\partial\xi} & \frac{\partial y}{\partial\eta}\end{bmatrix}^{-1}$$

$$J \equiv \frac{\partial(x, y)}{\partial(\xi, \eta)} \equiv \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix}$$

$$\begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \frac{\begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{bmatrix}}{\begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix}}$$

Which finally gives us the Metric terms as following:

$$\boxed{\begin{aligned} \frac{\partial \xi}{\partial x} &= \frac{1}{J}\frac{\partial y}{\partial \eta} \\ \frac{\partial \eta}{\partial x} &= -\frac{1}{J}\frac{\partial y}{\partial \xi} \\ \frac{\partial \xi}{\partial y} &= -\frac{1}{J}\frac{\partial x}{\partial \eta} \\ \frac{\partial \eta}{\partial y} &= \frac{1}{J}\frac{\partial x}{\partial \xi} \end{aligned}}$$

This part is coded in Derivatives.py module following the above relation.

To improve efficiency of the code, I have calculated all four derivatives at all grid points in computational domain as they remain same rather than calculating for each iteration.

Numerical Implementation:

All the derivatives of the Laplace Equation (Computational domain) were numerically approximated using **Central Differencing** as following**:**

$$\frac{\partial^2 \psi}{\partial \xi^2} = \frac{\psi_{\xi+1,\eta} - 2\psi_{\xi,\eta} + \psi_{\xi-1,\eta}}{\Delta \xi^2}$$

$$\frac{\partial^2 \psi}{\partial \eta^2} = \frac{\psi_{\xi,\eta+1} - 2\psi_{\xi,\eta} + \psi_{\xi,\eta\text{-}1}}{\Delta \eta^2}$$

$$\frac{\partial^2 \psi}{\partial \xi \partial \eta} = \frac{\psi_{\xi+1,\eta+1} - \psi_{\xi\text{-}1,\eta} - \psi_{\xi,\eta\text{-}1} + \psi_{\xi\text{-}1,\eta\text{-}1}}{4\Delta\eta\Delta\xi}$$

$$\frac{\partial \psi}{\partial \xi} = \frac{\psi_{\xi+1,\eta} - \psi_{\xi\text{-}1,\eta}}{2\Delta \xi}$$

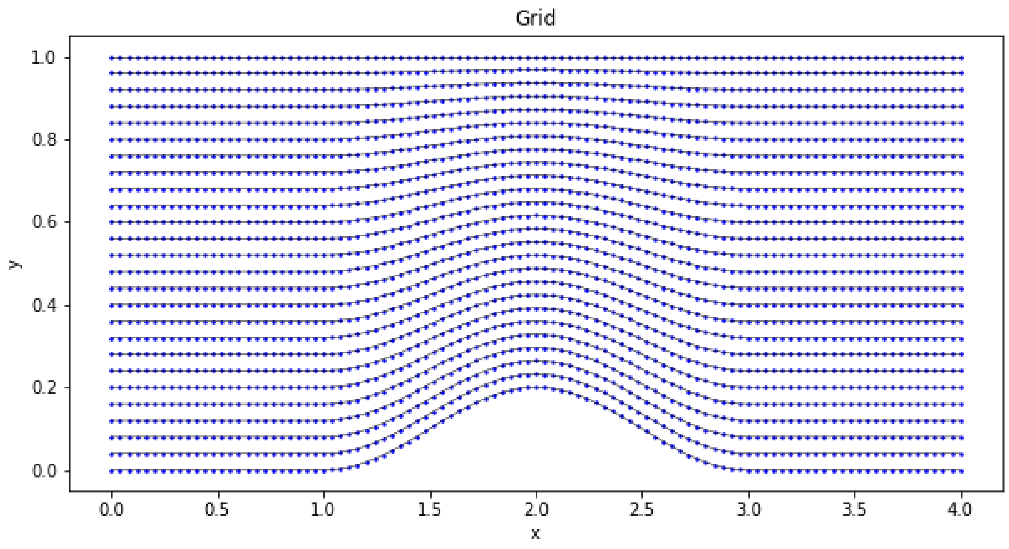
$$\frac{\partial\psi}{\partial\eta} = \frac{\psi_{\xi,\eta\text{-}1} - \psi_{\xi,\eta\text{-}1}}{2\Delta\eta}$$

Finally to update iteratively to convergence we used **Point Gauss Seidel Method:**

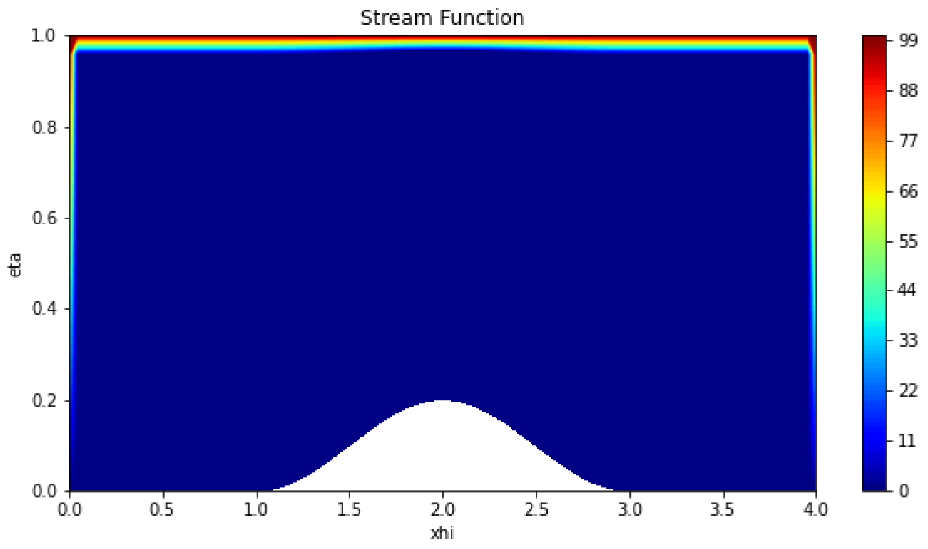
$$\psi_{\xi,\eta}^{n+1} = \left[\frac{\psi^{\mathrm{n}}{}_{\xi+1,\eta} + \psi^{\mathrm{n+1}}{}_{\xi-1,\eta}}{\Delta\xi^2}\left(\frac{\partial\xi}{\partial \mathrm{x}}^2 + \frac{\partial\xi}{\partial \mathrm{y}}^2\right) + \frac{\psi^{\mathrm{n}}{}_{\xi,\eta+1} + \psi^{\mathrm{n+1}}{}_{\xi,\eta\text{-}1}}{\Delta\eta^2}\left(\frac{\partial\eta}{\partial \mathrm{x}}^2 + \frac{\partial\eta}{\partial \mathrm{y}}^2\right)\right.$$
$$+ \frac{\psi^{\mathrm{n}}{}_{\xi+1,\eta+1} - \psi^{\mathrm{n+1}}{}_{\xi\text{-}1,\eta} - \psi^{\mathrm{n+1}}{}_{\xi,\eta\text{-}1} + \psi^{\mathrm{n+1}}{}_{\xi\text{-}1,\eta\text{-}1}}{4\Delta\eta\Delta\xi} + \frac{\psi^{\mathrm{n}}{}_{\xi+1,\eta} - \psi^{\mathrm{n+1}}{}_{\xi\text{-}1,\eta}}{2\Delta\xi}$$
$$\left. + \frac{\psi^{\mathrm{n+1}}{}_{\xi,\eta\text{-}1} - \psi^{\mathrm{n+1}}{}_{\xi,\eta\text{-}1}}{2\Delta\eta}\right] \times \left[\frac{1}{2\frac{\frac{\partial\xi}{\partial \mathrm{x}}^2 + \frac{\partial\xi}{\partial \mathrm{y}}^2}{\Delta\xi^2} + 2\frac{\frac{\partial\eta}{\partial \mathrm{x}}^2 + \frac{\partial\eta}{\partial \mathrm{y}}^2}{\Delta\eta^2}}\right]$$

This turns out to be the discretized update equation.

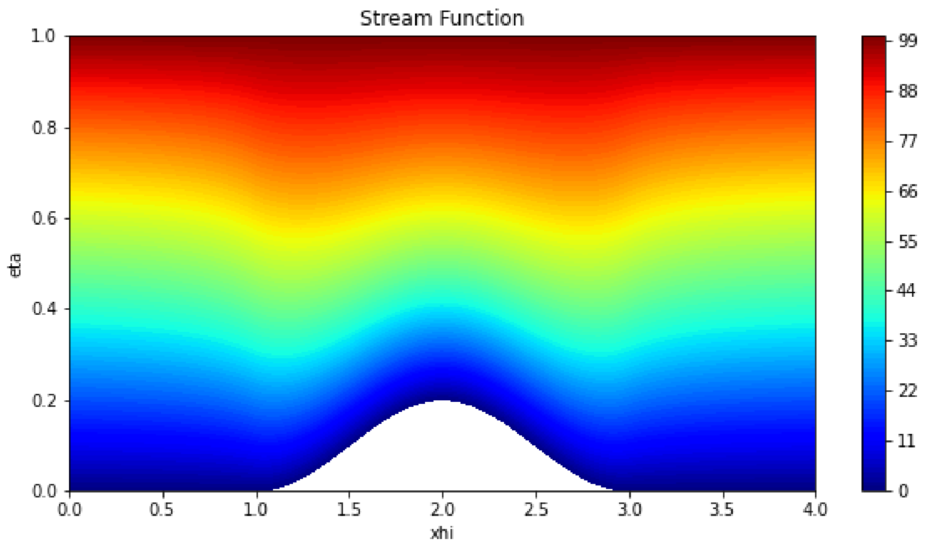
Results: (These all plots are generated by the code)



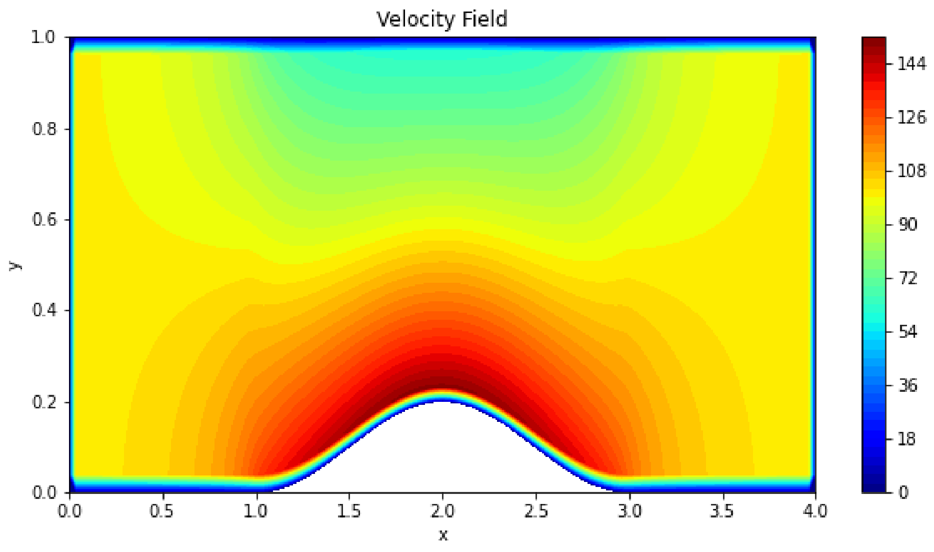
Initialized Grid



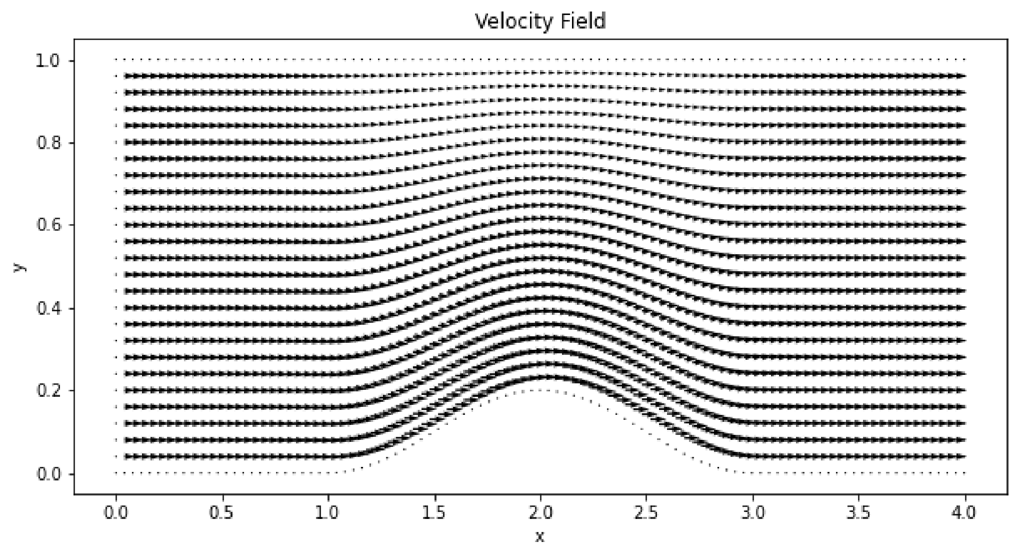
Initial Stream Function



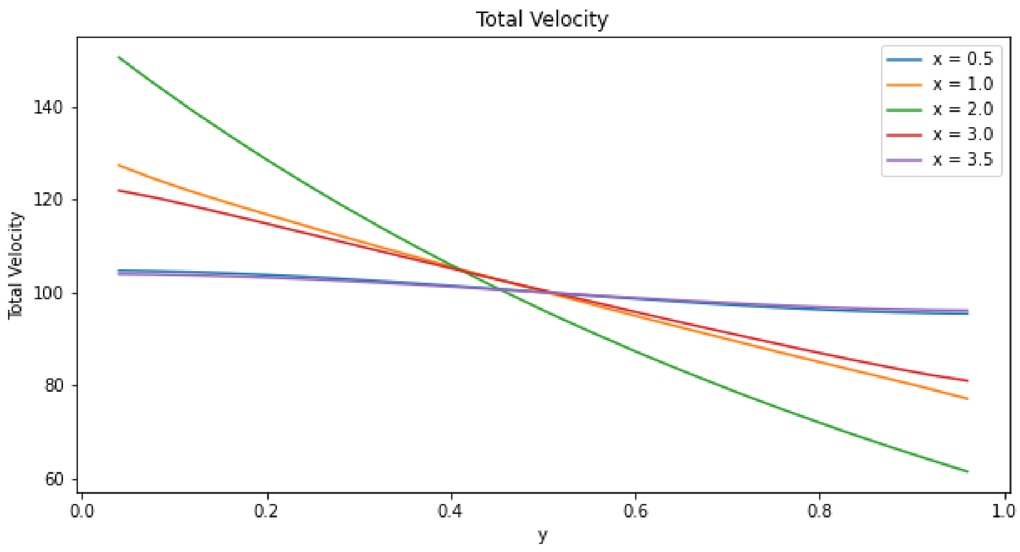
Stream Function after Convergence



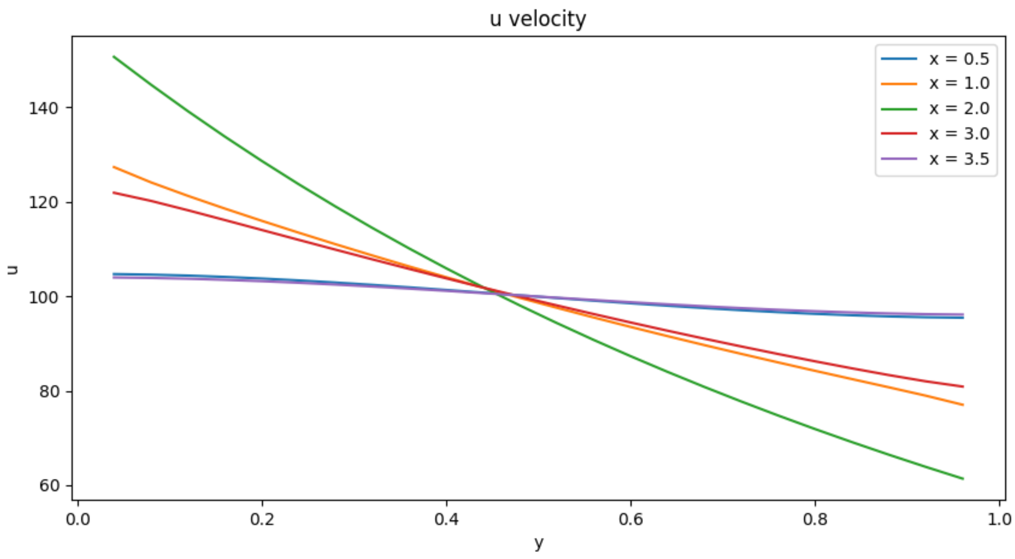
Final Velocity Field



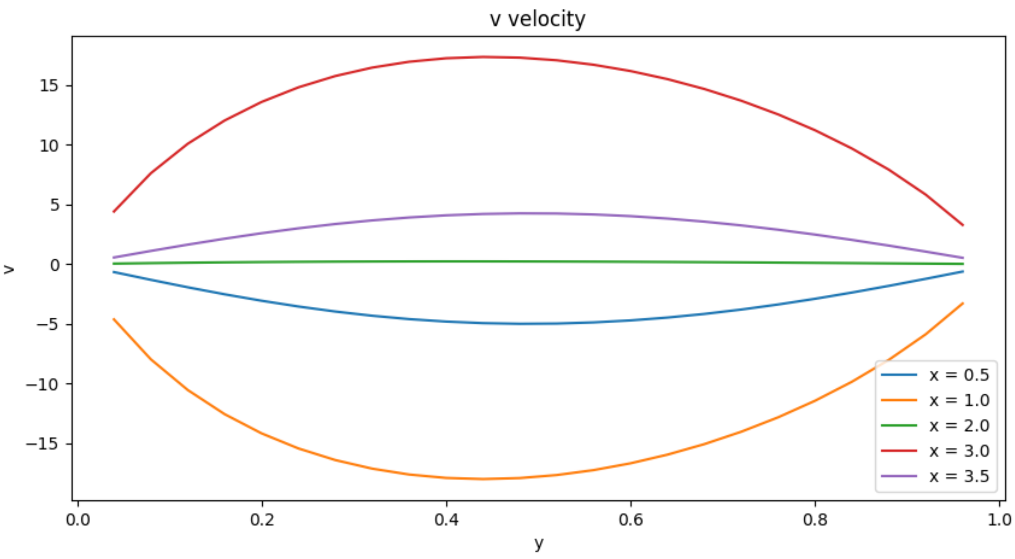
Quiver Plot of Velocity Field



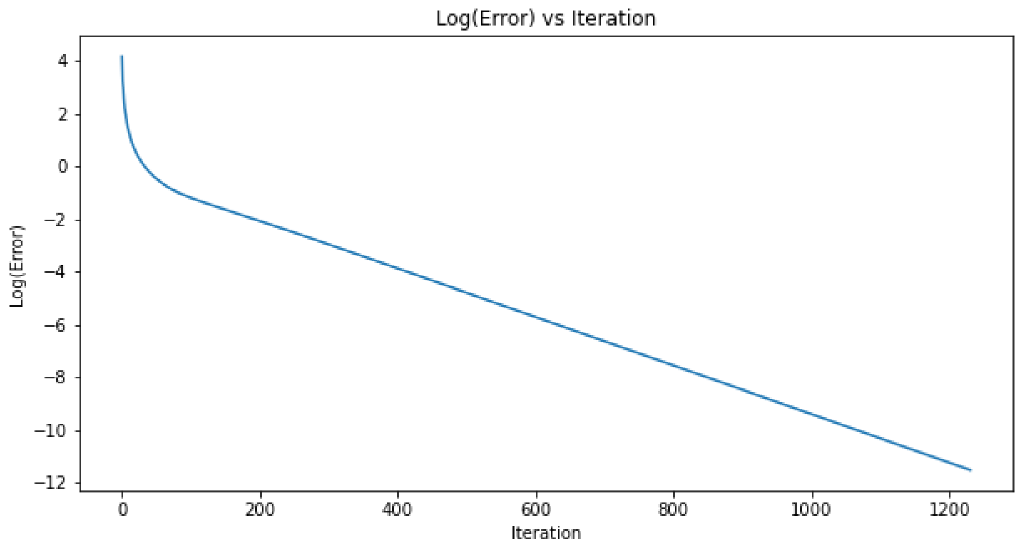
Variation of Total Velocity at x = (0.5, 1.0, 2.0, 3.0, 3.5) vs y



Variation of U component of Velocity at x = (0.5, 1.0, 3.0, 3.5) vs y



Variation of V component of Velocity at x = (0.5, 1.0, 3.0, 3.5) vs y



Plot of Log (Error) vs Iteration

Conclusions:

1. The Solution converged after 1231 iterations.
2. The variation of ψ with respect to y is linear, indicating that the velocity in the x-direction remains constant along a given x-coordinate. This behavior can be attributed to local mass conservation for each streamline, reflecting the laminar nature of the flow.
3. The slope of ψ is steeper towards the center of the venturi where the width narrows, resulting in faster velocity changes. In contrast, the ψ remains nearly constant for x-values within [0,1] ∪ [3,4] because there are no flow disturbances or geometric changes in this region until reaching the curved portion of the venturi.
4. At x = 2, the x-component of velocity (u) is calculated is consistent with theoretical predictions derived from mass conservation principles. Concurrently, the y-component of velocity (v) is observed to be negligible in comparison.
5. The velocity in the y-direction is significantly smaller than that in the x-direction, underscoring the dominance of x-directed flow velocities within the venturi configuration.

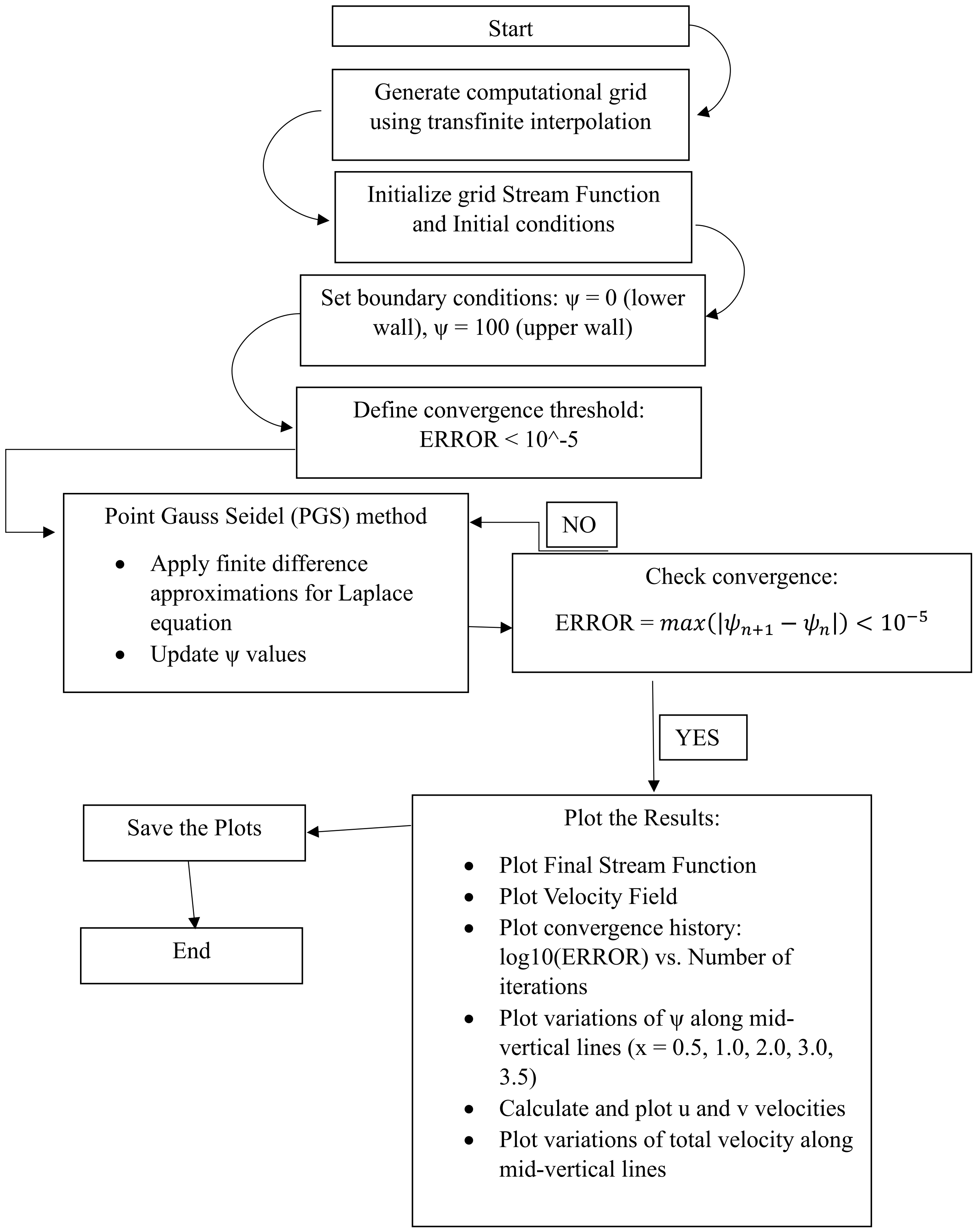
References

1. Prof. J. C. Mandal, AE706 Lecture Notes, Moodle
2. Anderson, J. D. (1995). Computational Fluid Dynamics: The Basics With Applications. McGraw-Hill Education.

Note:

Flowchart, zip file of code repository and README.txt has been provided with instructions on running the code.

FLOW CHART


Start
Generate computational grid using transfinite interpolation
Initialize grid Stream Function and Initial conditions
Set boundary conditions: ψ = 0 (lower wall), ψ = 100 (upper wall)
Define convergence threshold: ERROR < 10^-5
Point Gauss Seidel (PGS) method
• Apply finite difference approximations for Laplace equation
• Update ψ values
NO
Check convergence:
ERROR = $max(|\psi_{n+1} - \psi_n|) < 10^{-5}$
YES
Plot the Results:
• Plot Final Stream Function
• Plot Velocity Field
• Plot convergence history: log10(ERROR) vs. Number of iterations
• Plot variations of ψ along mid-vertical lines (x = 0.5, 1.0, 2.0, 3.0, 3.5)
• Calculate and plot u and v velocities
• Plot variations of total velocity along mid-vertical lines
Save the Plots
End