Two-dimensional fluid flow of a stream over a rectangular beam

April 4, 2024

1 Methodology

1.1 Problem Definition

1.1.1 Aims and Objectives

We aim to find and graphically visualise the flow profile of a rectangular beam submerged in an oncoming stream with steady flow and constant speed U, as shown in Figure 1. Additionally, we aim to investigate how the flow changes for different parameters such as the Reynolds number R, and where we choose to place the beam. Our objectives are to discretise the equations governing fluid flow in our problem and solve them numerically in a two-dimensional solution domain, which is a grid of discrete points.

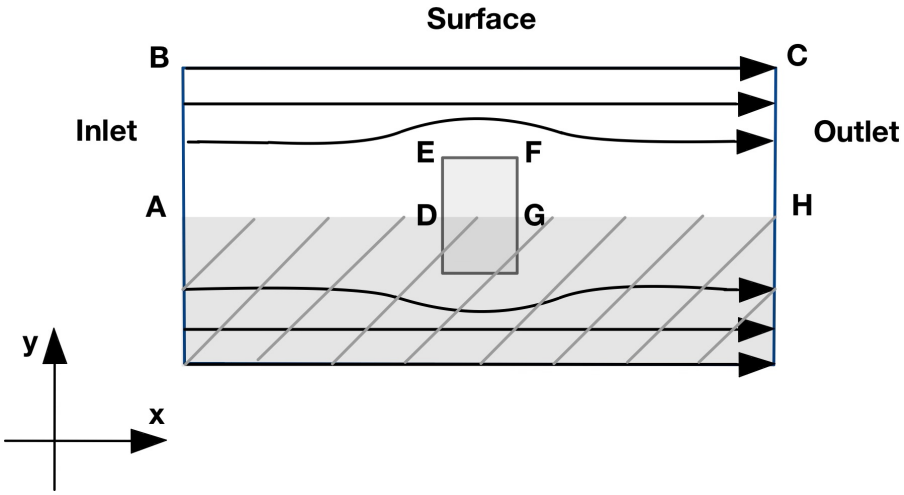

Figure 1: A diagram of the fluid flow around a beam in the centre of an oncoming stream. The arrows show the direction of flow and the boundary between the shaded and non-shaded regions indicates the plane of symmetry.

1.1.2 Governing Equations

We introduce the stream ψ and vorticity ω functions to characterise the fluid behaviour for our problem. From the contours of ψ, we obtain the streamlines that map the paths of the fluid particles in the flow. We use ω to describe the local rotation of the fluid particles. We express ψ and ω in terms of the velocity components $\underline{u} = (u, v)$.

$$\frac{\partial \psi}{\partial y} = u, \quad \frac{\partial \psi}{\partial x} = -v, \quad \omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{1}$$

For our problem, we require steady solutions that satisfy $\frac{\partial \omega}{\partial t} = 0$. By applying this condition to the conservation of mass and momentum equations, we obtain the coupled partial differential

equations that govern our system.

$$
\begin{aligned}
-\omega &= \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} \\
\frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} &= \nu \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right)
\end{aligned}
\tag{2}
$$

After solving ψ and ω numerically, we visualise fluid flow by plotting their contours.

1.2 Problem Geometry

As Figure 1 illustrates, the fluid flow is symmetric about the centreline AH. Therefore, we only need to find half of the stream's flow profile, as we can obtain the other half by reflecting in the centreline AH. This halves the computations we need to perform. We define our problem geometry in Figure 2

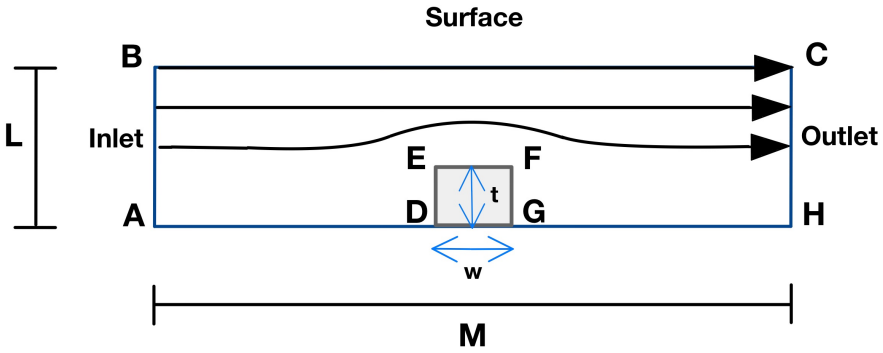

Figure 2: A diagram of the fluid flow over a beam with height t and width w, in a channel of length M. The stream enters the channel at the inlet AB and exits at the outlet CH. We define the channel width L as the characteristic length for our problem. The vertices of the beam are labelled $D - G$.

The fluid behaviour we observe depends on the specific geometry and fluid properties we choose for our system. We want the ability to easily compare the fluid behaviour for different conditions. Hence, we use the Reynolds number R to characterise the fluid flow specific to our system with a single dimensionless parameter.

$$
R = \frac{UL}{\nu} \tag{3}
$$

Equation 3 shows that increasing the characteristic length L i.e. the channel width has the same effect as increasing the fluid velocity U or decreasing the fluid's kinematic viscosity ν. This allows us to scale our problem for differently-sized systems. For example, if we want to apply our problem to a physical system with a larger channel width L, we can ensure the same flow behaviour by adjusting the other variables in Equation 3 to maintain R. For our problem, we set the constant fluid velocity $U = 1$ and the characteristic length $L = 1$. Under these conditions, changing R is equivalent to changing the kinematic viscosity ν of the fluid.

Our discretised solution domain is a two-dimensional grid with equal grid spacing h in both the x and y direction. We have two separate solution grids, one to solve ψ and the other to solve ω. We choose the number of grid points n along the y-axis to set the grid-resolution and we use $2n$ grid points along the x-axis to maintain the rectangular shape of the channel. We set $L = 1$, by restricting the y axis of our grid to span $0 - 1$ and set $M = 2L$ accordingly. We summarise our parameters in Table1.

We choose the beam dimensions $w \times t$ and placement AD that gives the best visualisation of the flow patterns and maintains enough distance between the beam and the surface so that surface flow is unchanged by the beam.

Parameter	Description	Value
h	The uniform grid spacing between points,equal in both the x and y direction.	variable and dependent on n
n	The number of discrete points on the y-axis, determines grid resolution.	variable
L	The y-axis spanning 0-1, which is the characteristic length.	1 n grid points.
M	The x-axis spanning 0-2.	2 n grid points.
t	Height of the beam.	0.14 n grid points
w	Width of the beam.	0.08 n grid points
AD	Length between origin and beamfront DE, determining the beam placement.	0.15 n grid points
U	Background flow velocity of the stream.	1
R	The Reynolds number.	variable

Table 1: A table showing the key parameters of the problem, their descriptions, and their associated value either given as a variable parameter, a constant number, or a value in terms of the number of grid points

1.3 Numerical Method

1.3.1 Central Difference Formulae

We use the central difference method to discretise the coupled expressions in Equation 2. Computationally, we cannot treat a derivative as the instantaneous rate of change because computers have finite precision and cannot process infinitesimal quantities. Instead, we approximate derivatives by discretising a function into finite intervals and finding the slope between neighbouring points over each interval.

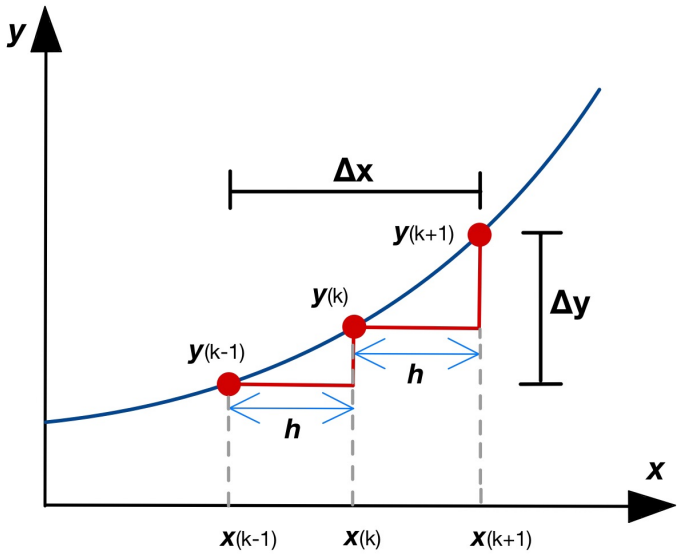

Figure 3: A diagram illustrating the midpoint method. The function (shown in blue) is represented by discrete points (shown in red) indexed by k and separated by the equal interval h. The derivative at the point y_k is determined over the interval y_{k-1} to y_{k+1}.

The central difference method uses three points in the interval, as shown in Figure 3. We define the derivative $\frac{dy}{dx}$ at the middle point y_k using the points either side, y_{k-1} and y_{k+1}, and can therefore write the discretised derivative as follows.

$$\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x} = \frac{y_{k+1} - y_{k+1}}{2h} \tag{4}$$

Similarly, we derive the central difference formula for a second order derivative as follows.

$$\frac{d^2y}{dx^2} \approx \frac{\frac{y_{k+1}-y_k}{h} - \frac{y_k-y_{k-1}}{h}}{h} = \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} \tag{5}$$

Using these principles, we find each point in a two-dimensional function $U_{i,j}$ in terms of its 4 nearest neighbouring points as shown in Figure 4.

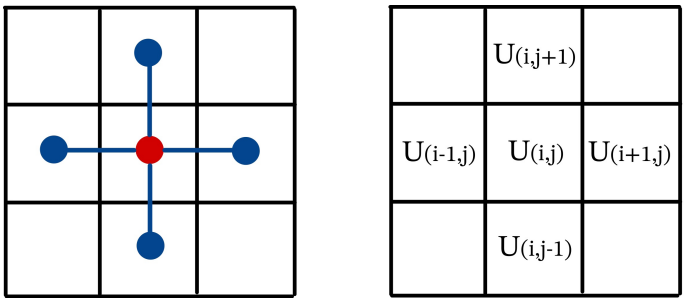

Figure 4: On the left, the point $U_{(i,j)}$ (shown in red) and its 4 nearest neighbouring points (shown in blue). On the right, the corresponding indexes of the points are shown.

This allows us to express each point as follows.

$$\begin{aligned} U_{i,j} &= \frac{1}{4}\left[U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - h^2 f(x_i, y_j)\right] \\ \text{where} \quad \nabla^2 U &= f(x_i, y_j) = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \end{aligned} \tag{6}$$

1.3.2 Gauss-Seidel Iteration

We utilise the Gauss-Seidel method, which updates each grid point using the updated values of its neighboring points obtained during the current loop through the grid, shown in Figure 5. We loop iteratively over the grid until the solution satisfactorily converges. Points along the grid boundaries are excluded from the loop range since their neighboring points fall outside beyond the solution domain. To account for this, we introduce "ghost" points around the grid boundaries, which allow the boundary points to be updated. The ghost points are removed after the solution converges to restore the original domain.

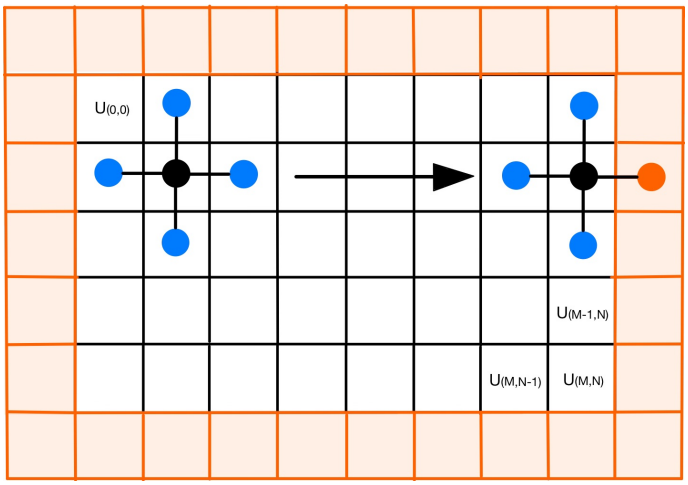

Figure 5: A schematic showing an two-dimensional grid U with M rows and N columns undergoing a Gauss-Seidel iteration. The point (shown in black) is at the index $U_{(1,1)}$ being updated using its 4 nearest neighbouring points (shown in blue) and then moves across the grid, indicated by the arrow. At the index $U_{(1,N)}$, one of the neighbouring points is a "ghost point" (shown in orange), outside the normal solution domain. The padding is highlighted in orange around the boundaries of the grid.

1.4 Discretisation

1.4.1 Update Rules

We can apply the central difference formulae from Equations 4, 5, and 6 to Equation 2 to obtain the following update rules for ψ and ω.

$$\psi_{i,j} = \frac{1}{4}\left(\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1} + h^2\omega_{i,j}\right) \tag{7}$$

$$\begin{aligned}\omega_{i,j} = &\frac{1}{4}\left(\omega_{i+1,j} + \omega_{i-1,j} + \omega_{i,j+1} + \omega_{i,j-1}\right) \\ &- \frac{R_G}{16}\left[\left(\psi_{i,j+1} - \psi_{i,j-1}\right)\left(\omega_{i+1,j} - \omega_{i-1,j}\right) - \left(\psi_{i+1,j} - \psi_{i-1,j}\right)\left(\omega_{i,j+1} - \omega_{i,j-1}\right)\right]\end{aligned} \tag{8}$$

To ensure the ψ and ν variables in Equation 2 are dimensionless, we introduce the Grid Reynolds number R_G, which is the physical Reynolds number R in Equation 3 scaled to the grid spacing h as shown in Equation 9.

$$R_G = R\left(\frac{h}{L}\right) = \frac{UhL}{L\nu} \implies R_G = \frac{Uh}{\nu} \tag{9}$$

We also note that (i, j) in Equations 7 and 8 are the indices for (x, y) or (columns,rows). For these indices, $i + 1$ represents moving a step along the x-axis, i.e. across one column to right. Similarly, $j+1$ represents moving up a step along the y-axis, i.e. up by one row. In Python however, (i, j) are the indices for (rows,columns) —and the row index starts at the top and increases as we descend the rows. Hence $i+1$ and $j+1$ represent moving down a row and moving a column across to the right, respectively. Instead of converting our discretised equations, we use them as they are with their original indexing and correct the updated solution grids by transposing and flipping along the y-axis as shown in Figure 6.

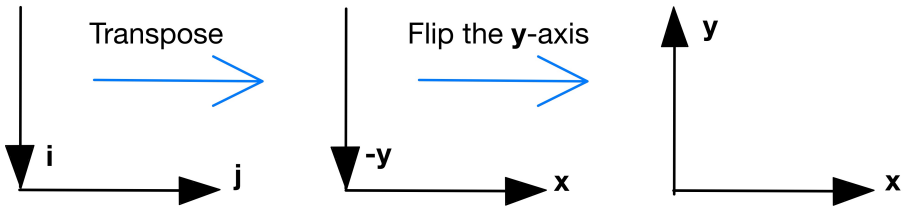

Figure 6: A diagram showing the transformation of the Axes to convert the indexing of the grids so that $(i, j) = (x, y)$

1.4.2 Boundary Conditions

We derive our boundary conditions by considering the physics of our system and discretising derivative terms using the central difference formulae.

Upstream of the beam we have laminar flow, so the fluid has no local rotation and the streamlines are parallel in the x direction.

$$\text{Inlet (AB)}: \quad \frac{\partial \psi}{\partial x} = 0, \quad \omega = 0 \tag{10}$$

Downstream of the beam, we assume the conditions do not affect the fluid behaviour upstream. Hence, ω and ψ are constant in the x direction.

$$\text{Outlet (CH)}: \quad \frac{\partial \psi}{\partial x} = 0, \quad \frac{\partial \omega}{\partial x} = 0 \tag{11}$$

We place the beam far enough away from the surface to ensure that the surface flow is unaffected. At the surface, we expect a streamline with free flow conditions.

$$\text{Surface (BC)}: \quad \frac{\partial \psi}{\partial y} = U, \quad \omega = 0 \tag{12}$$

We delineate a streamline along the centreline and beam surfaces. Because the beam is solid and fixed in place, there is no slip or through-flow on its surfaces and the velocity components are zero. Hence, $\psi = 0$ along the whole streamline. We observe ω on the streamline arising from a perpendicular velocity gradient, which is present on the beam surfaces, but zero on the centreline to maintain symmetric flow.

$$\text{Centreline (AH)}: \quad \psi = 0, \quad \omega = 0 \tag{13}$$

$$\text{Beam (DEFG)} \begin{cases} \text{Beamfront (DE)}: & \psi = 0, \quad \omega = -\frac{\partial^2 \psi}{\partial x^2} \\ \text{Beamback (FG)}: & \psi = 0, \quad \omega = -\frac{\partial^2 \psi}{\partial x^2} \\ \text{Beamtop (EF)}: & \psi = 0, \quad \omega = -\frac{\partial^2 \psi}{\partial y^2} \end{cases} \tag{14}$$

Recalling $U = 1$, we discretise all the derivative terms in the boundary conditions by applying Equations 4 and 5. Our calculations to find the discretised boundary conditions are shown below and then summarised in Table 2.

$$\begin{aligned}
\text{For Inlet and Outlet}: \quad & \frac{\partial \psi}{\partial x} = \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h} = 0 \implies \psi_{i+1,j} = \psi_{i-1,j} \\
\text{For Outlet}: \quad & \frac{\partial \omega}{\partial x} = \frac{\omega_{i+1,j} - \omega_{i-1,j}}{2h} = 0 \implies \omega_{i+1,j} = \omega_{i-1,j} \\
\text{For Surface}: \quad & \frac{\partial \psi}{\partial y} = \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2h} = 1 \implies \psi_{i,j+1} = \psi_{i,j-1} + 2h \\
\text{For Beamfront and Beamback}: \quad & -\frac{\partial^2 \psi}{\partial x^2} = -\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{h^2} \\
\text{For Beamtop}: \quad & -\frac{\partial^2 \psi}{\partial y^2} = -\frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{h^2}
\end{aligned}$$

Table 2: Boundary Conditions

Boundary	**Condition**	
	ψ	ω
Inlet AB	$\psi_{i+1,j} = \psi_{i-1,j}$	$\omega = 0$
Outlet CH	$\psi_{i+1,j} = \psi_{i-1,j}$	$\omega_{i+1,j} = \omega_{i-1,j}$
Surface BC	$\psi_{i,j+1} = \psi_{i,j-1} + 2h$	$\omega = 0$
Centreline AH	$\psi = 0$	$\omega = 0$
Beamfront DE	$\psi = 0$	$\omega = -\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{h^2}$
Beamback FG	$\psi = 0$	$\omega = -\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{h^2}$
Beamtop EF	$\psi = 0$	$\omega = -\frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{h^2}$

For ψ, the Inlet, Outlet, and Surface conditions are applied across the boundary and involve a ghost point, requiring padding along the top row and outermost columns. For ω we only require padding along the Outlet, as this is only condition applied across a grid boundary that uses a ghost point. However, Equation 8 shows that the ω and ψ update rules are coupled and must therefore be applied in the same Gauss-Seidel iteration loop. To avoid indexing issues, we apply the same padding to the solution grids for ψ and ω to ensure they have the same numbers of rows and columns.

Instead of attempting to initialising a grid with a missing space for the beam, we include the beam in solution domain but impose boundary conditions where the beam surfaces are. After solutions are obtained, we then remove the beam from the domain of fluid flow by covering the beam region in our solutions with a patch of the beam's dimensions.

1.5 Convergence

Our computational procedure to find the solutions and ensure they have converged is summarised in Figure 7. We calculate the relative error as shown in Equation 15.

$$\begin{aligned} \psi_{\text{Relative Error}} &= \frac{|\psi - \psi_{Snapshot}|}{|\psi|} \\ \omega_{\text{Relative Error}} &= \frac{|\omega - \omega_{Snapshot}|}{|\omega|} \end{aligned} \tag{15}$$

Figure 7 shows the flowchart for the iterative process.

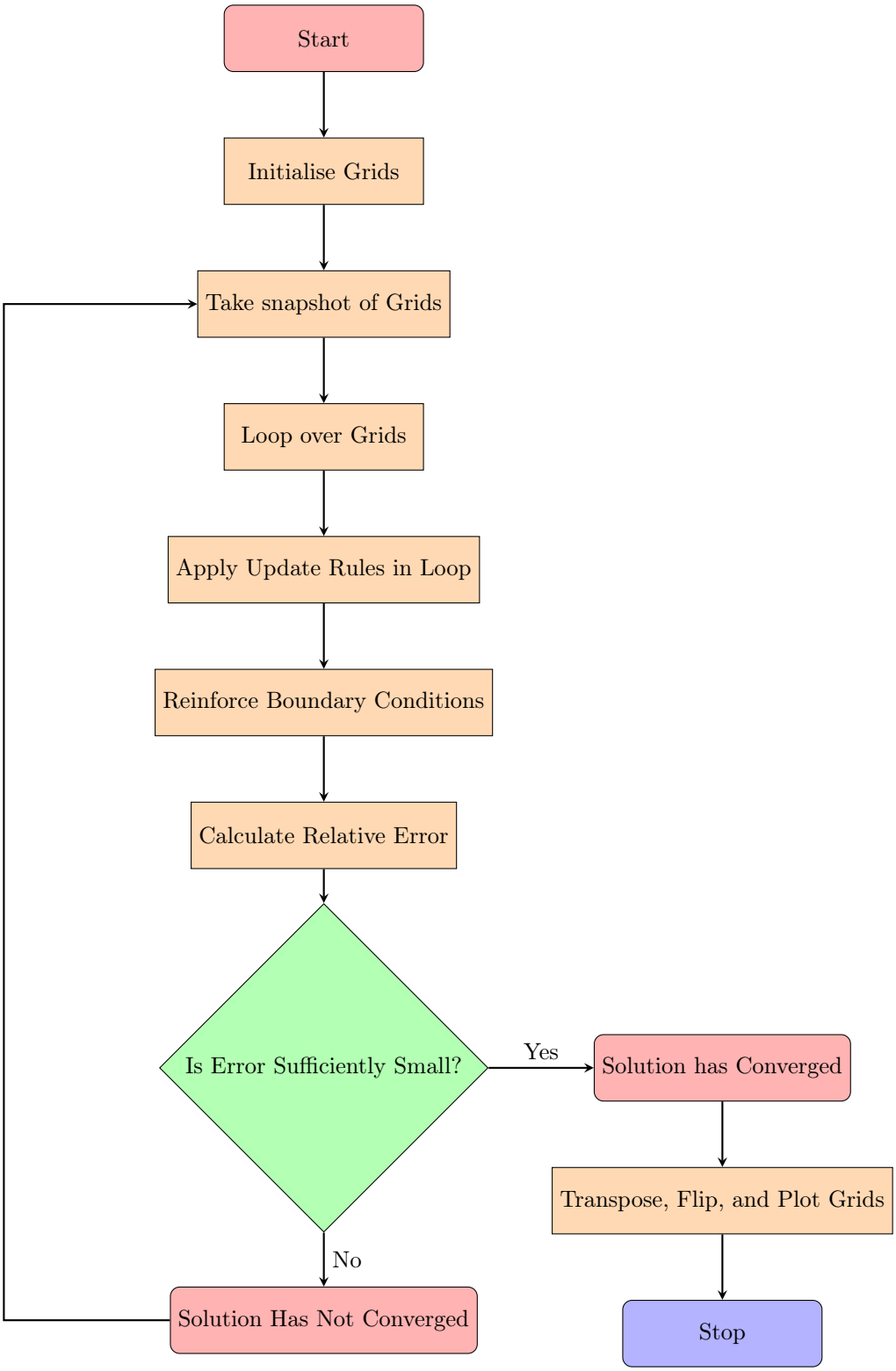

Figure 7: Flowchart for the Gauss-Seidel Iterative Process

2 Results

2.1 Varying Reynolds Number

In Figures 8, 10, 12, and 14 we show the stream and vorticity contour plots for the Reynolds numbers $R = 1000$, $R = 3000$, $R = 6000$, and $R = 15000$, respectively. We confirm these solutions have converged by breaking the Gauss-Seidel iteration loop when the relative errors for both the ψ and ω solution grids are less than or equal to a predefined tolerance. The relative error is defined as insert Frob Norm By plotting their associated errors as shown in Figures 9 , 11, 13, and 15 we can also confirm visually that errors drop and then do not increase, indicating that the solutions have converged.

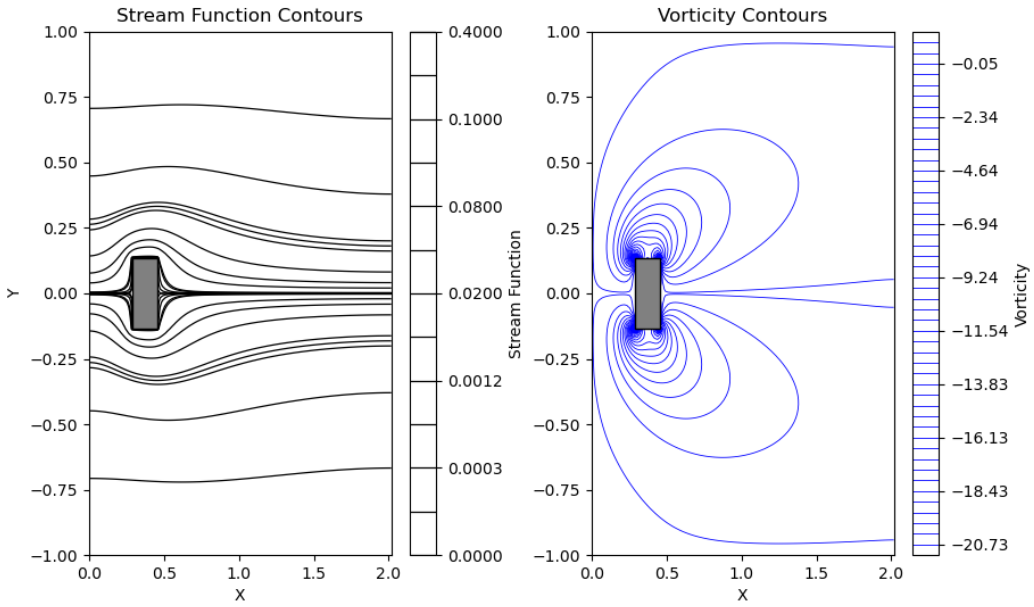

Figure 8: Contour plots of the stream ψ function (left) and the vorticity ω function (right) for the fluid flow problem in the full solution space for a Reynolds number $R = 1000$. For this solution, $n = 60$, and it took 3131 Gauss-Seidel iterations of the grids to reach convergence, set at a tolerance of 1.8×10^{-4}. This is equivalent to the relative error being 0.018% of the solution plots. The starting values for the ghost points were 0.5 and -1 for ψ and ω, respectively

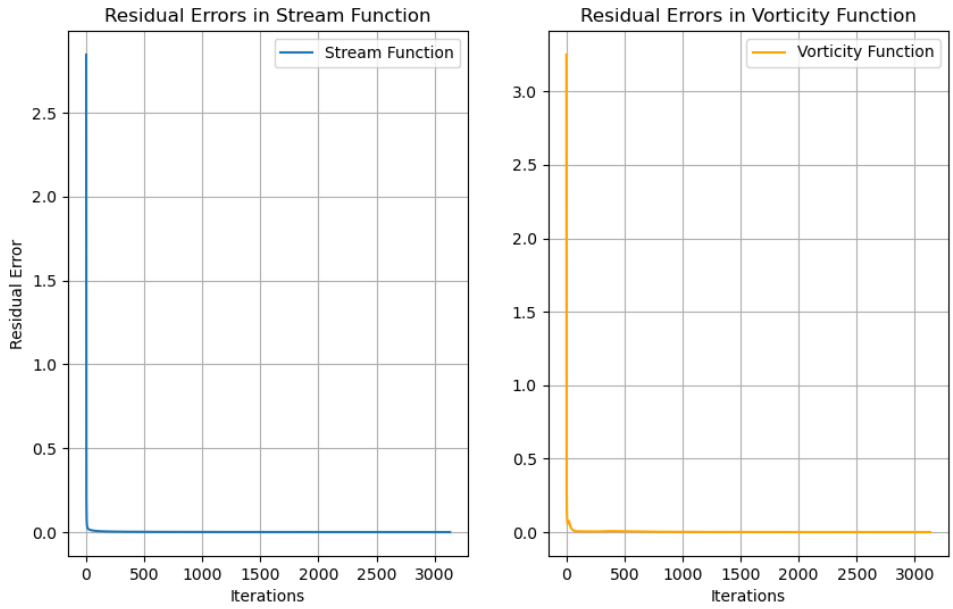

Figure 9: Plots of the relative error against the number of Gauss-Seidel iterations for the stream ψ function (left) and the vorticity ω function (right), for a Reynolds Number $R = 1000$. For this solution, $n = 60$, and it took 3131 Gauss-Seidel iterations of the grids to reach convergence, set at a tolerance of 1.8×10^{-4}. This is equivalent to the relative error being 0.018% of the solution plots. The starting values for the ghost points were 0.5 and -1 for ψ and ω, respectively

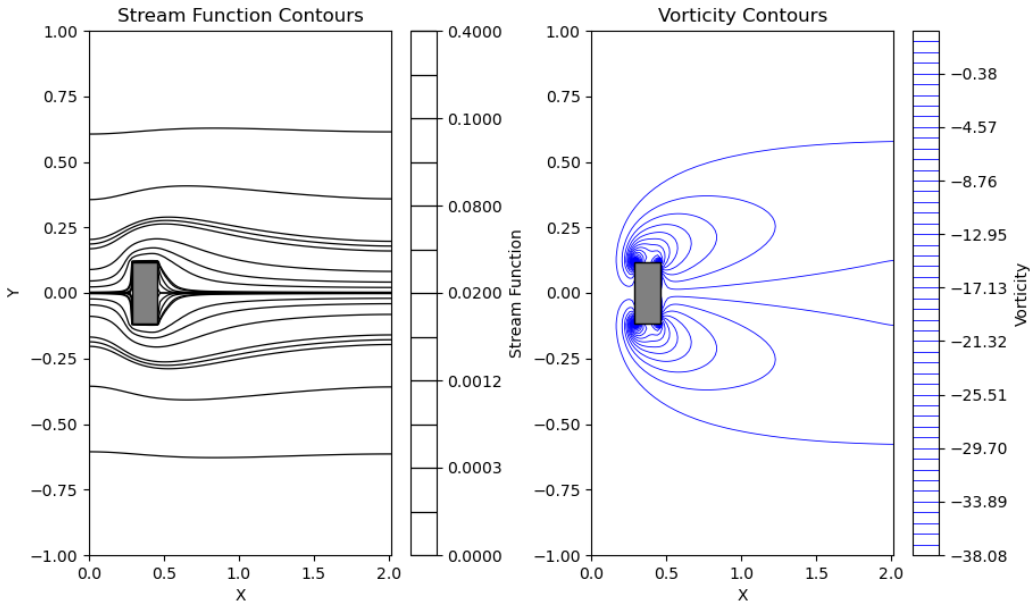

Figure 10: Contour plots of the stream ψ function (left) and the vorticity ω function (right) for the fluid flow problem in the full solution space for a Reynolds number $R = 3438$. For this solution, $n = 60$, and it took 4520 Gauss-Seidel iterations of the grids to reach convergence, set at a tolerance of 1.8×10^{-4}. This is equivalent to the relative error being 0.018% of the solution plots.

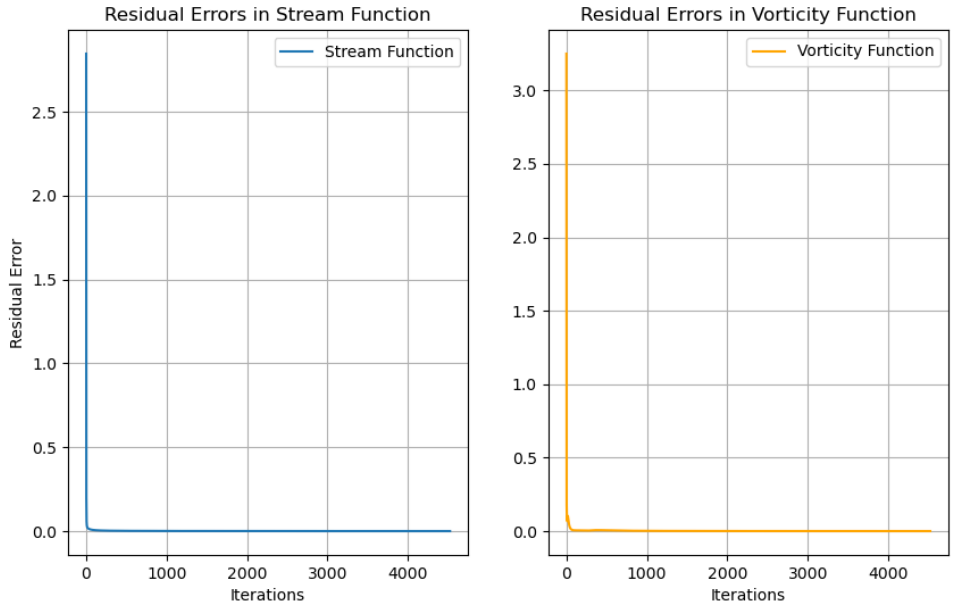

Figure 11: Plots of the relative error against the number of Gauss-Seidel iterations for the stream ψ function (left) and the vorticity ω function (right), for a Reynolds Number $R = 3000$. For this solution, $n = 60$, and it took 4520 Gauss-Seidel iterations of the grids to reach convergence, set at a tolerance of 1×10^{-4}. This is equivalent to the relative error being 0.01% of the solution plots.

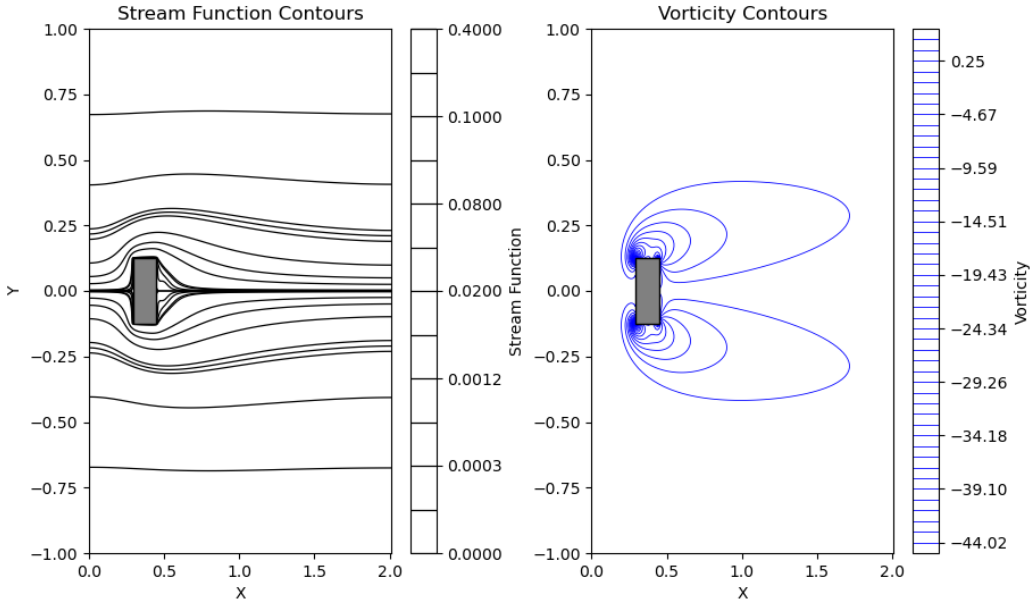

Figure 12: Contour plots of the stream ψ function (left) and the vorticity ω function (right) for the fluid flow problem in the full solution space for a Reynolds number $R = 6000$. For this solution, $n = 80$, and it took 4718 Gauss-Seidel iterations of the grids to reach convergence, set at a tolerance of 1.8×10^{-4}. This is equivalent to the relative error being 0.018% of the solution plots.

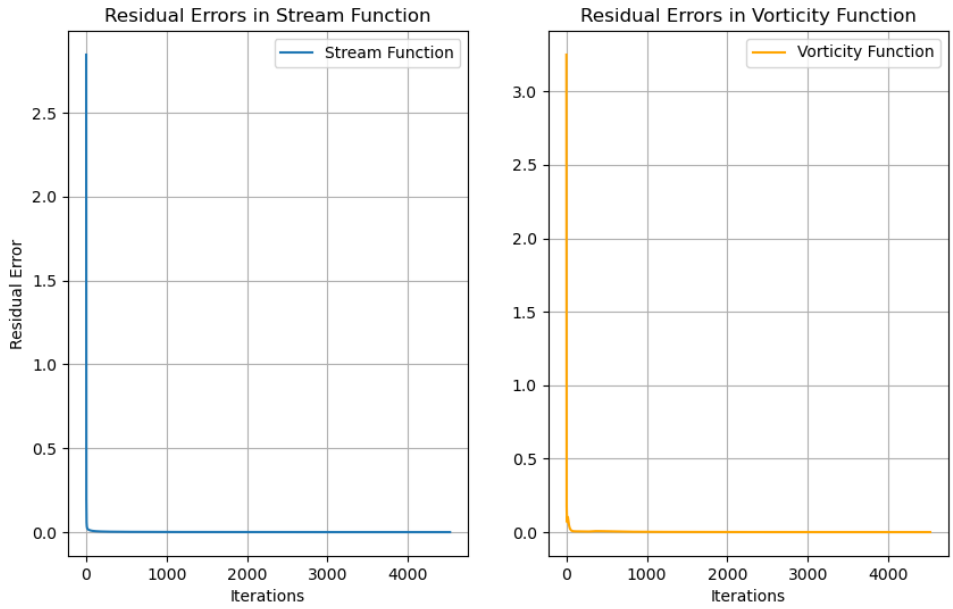

Figure 13: Plots of the relative error against the number of Gauss-Seidel iterations for the stream ψ function (left) and the vorticity ω function (right), for a Reynolds Number $R = 6000$. For this solution, $n = 80$, and it took 4718 Gauss-Seidel iterations of the grids to reach convergence, set at a tolerance of 1.8×10^{-4}. This is equivalent to the relative error being 0.018% of the solution plots.

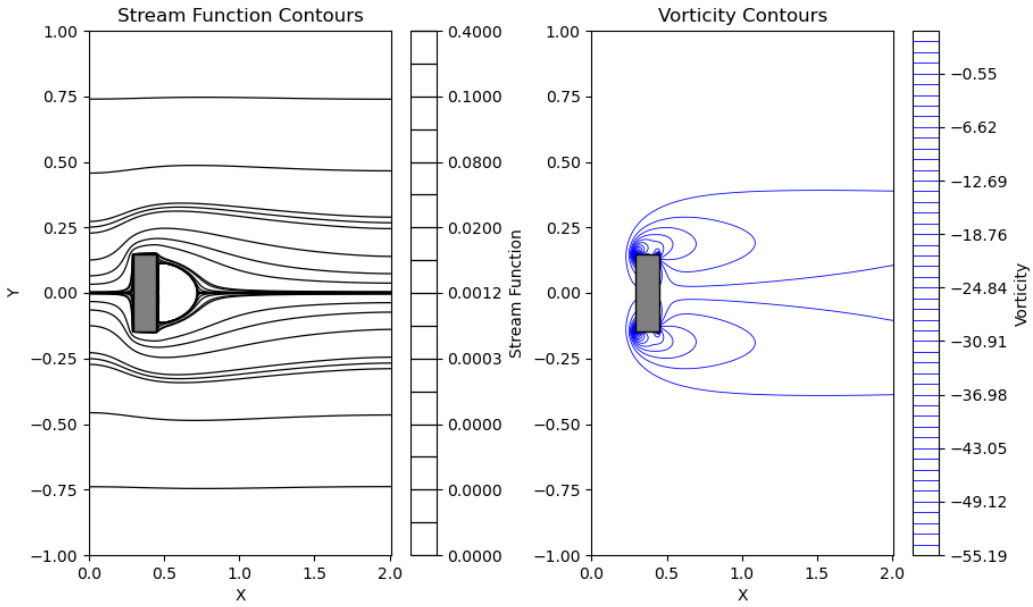

Figure 14: Contour plots of the stream ψ function (left) and the vorticity ω function (right) for the fluid flow problem in the full solution space for a Reynolds number $R = 15000$. For this solution, $n = 90$, and it took 5477 Gauss-Seidel iterations of the grids to reach convergence, set at a tolerance of 1.8×10^{-4}. This is equivalent to the relative error being 0.018% of the solution plots.

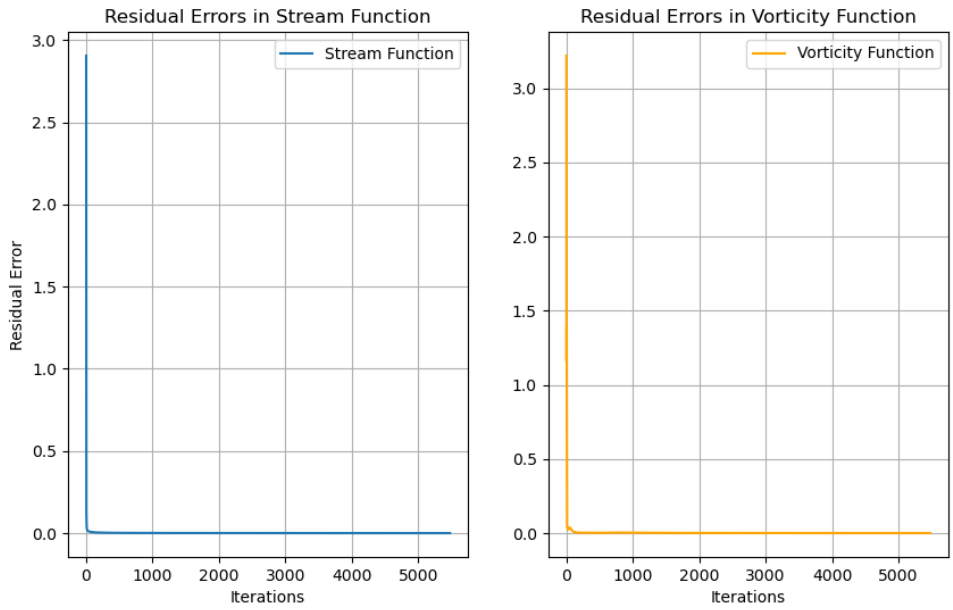

Figure 15: Plots of the relative error against the number of Gauss-Seidel iterations for the stream ψ function (left) and the vorticity ω function (right), for a Reynolds Number $R = 15000$. For this solution, $n = 90$, and it took 5477 Gauss-Seidel iterations of the grids to reach convergence, set at a tolerance of 1.8×10^{-4}. This is equivalent to the relative error being 0.018% of the solution plots.

2.2 Observations

We note that the general trend is that solutions for greater Reynolds numbers require more iterations to converge. Critically, numerical values that are too large to be calculated with our software begin to appear in computations for Reynolds numbers greater than $R = 3000$ when using $n = 60$. The error occurs when applying the update rule in Equation 8 , due to the magnitude of the Grid Reynolds number R_G. Because $L = 1$ is fixed, the only way to reduce R_G without changing the fluid's properties is by increasing n which decreases h and, in turn, decreases R_G. Hence, $n > 60$ are needed to compute solutions for $R > 3000$.