Computational Thermo-Fluid Dynamics
Sommer Semester 2015

Vorticity equation in a two dimensional channel flow

Dustin Draper[1]*, Ben Cappuyns[2]

Abstract
This report deals with the two-dimensional vorticity equation in Cartesian coordinates. A finite difference method is used to approximate the vorticity equation. The vorticity equation is derived from the classic Navier-Stokes equations for incompressible flow, which is valid for Mach-numbers under 0.3. An iterative solution strategy allows coupling of different fluidmechanical parameters, in particular velocity, stream function and vorticity. This strategy is used to simulate the behaviour of vortices in a two-dimensional channel flow.

Keywords
Vorticity — 2D flow channel flow

[1] *Master Student, TUM, Munich, Germany*
[2] *Exchange student, KUleuven, Belgium*
***e-mails**: dustin.draper@tum.de, ben.cappuyns@student.kuleuven.be

Contents

1. Introduction

The study of vortices is a fundamental part of fluid mechanics due to its many applications. A vortex is a region of space in a fluid in which the fluid rotates around a common axis. Vortices appear in real life as hurricanes, tornadoes, in the wake of planes, in the draining of sinks, and in many other applications

The ability to simulate the behavior of vortices is of particular importance in weather forecasts. On a global scale, vortices appear in the form of hurricanes and typhoons. On this scale, the Coriolis effect due to the roatation of the earth becomes relevant. The initial set-up of this report was to simulate the behavior of vortices on a global scale. However, due to the magnitude of this problem, several simplifications were made to reduce to complexity of the simulation. The focus now lies on the behavior of vortices in a two dimensional channel flow.

As is often the case in fluid mechanics, the governing equations are the Navier-Stokes equations. These are transformed into the vorticity equation by use of the curl operator, as shown in [1]. A classical finite difference method approximates the vorticity equation in able to compute numerical results.

After the explanation of the configuration, the report describes the mathematical model and derives the relevant equations. The solution strategy and derivations of the boundary conditions then follow. In the last paragraph, the results of the simulation are discussed with some propositions for future work.

2. Configuration under study

The configuration under study is a two dimensional channel flow. The channel is a rectangle with the long side along the x-axis. Fig. 1 shows a schematic drawing.

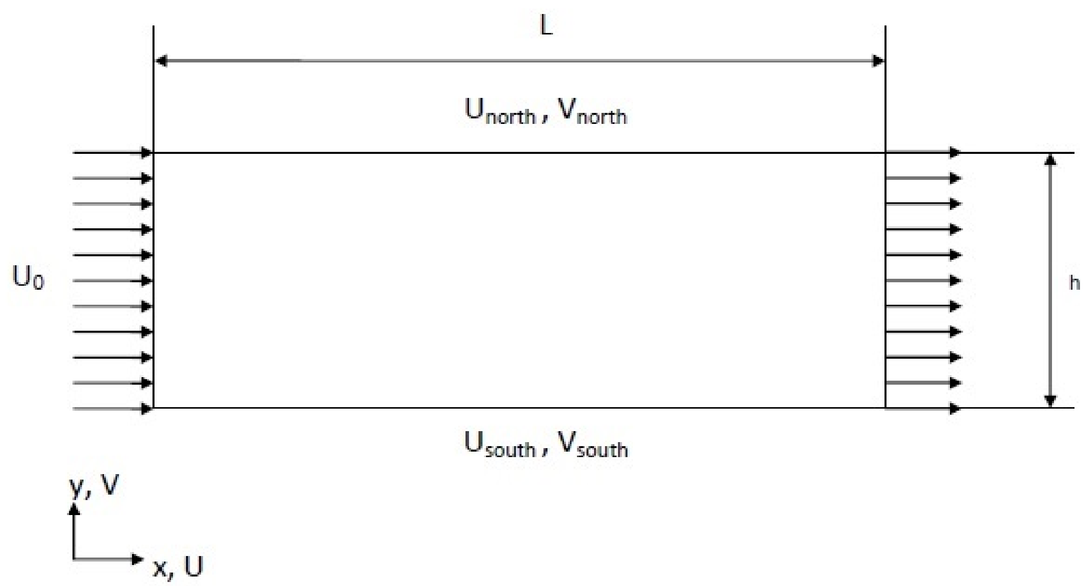


Figure 1. Schematic drawing of two dimensional channel flow

3. A finite difference approach of the vorticity equation

3.1 Mathematical description

The starting point of the proposed model are the continuity equation and the Navier-Stokes equations. The equations are simplified for this project into the 2D case for Cartesian coordinates, with incompressible, isothermal flow. The continuity equation is shown by Eq. (1).

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

with u and v the velocity in the x-direction and in the y-direction, respectively.

The Navier-Stokes equations for incompressible, isothermal flow are shown in Eq. (2) and Eq. (3).

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \boldsymbol{\nu}\left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y}\right). \tag{2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \boldsymbol{\nu}\left(\frac{\partial^2 v}{\partial^2 x} + \frac{\partial^2 v}{\partial^2 y}\right). \tag{3}$$

Here, u, v, P and ν are the velocity in the x-direction, the velocity in the y-direction, the pressure and the kinematic viscosity, respectively.

Next, we need the definition of the vorticity in the 2D case. This is given by (4).

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{4}$$

Here, ω denotes the vorticity. For the sake of simplicity, the subscript z has been omitted. We should therefore always keep in mind that the vorticity is in the positive z-direction.

To derive the vorticity equation, we take the partial derivative with respect to x from Eq. (3) and the partial derivative with respect to y from Eq. (2). After that equation (2) is subtracted from (3). Together with the continuity equation (1) and the definition of vorticity (4), this results in (5).

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \boldsymbol{\nu}\left(\frac{\partial^2 \omega}{\partial^2 x} + \frac{\partial^2 \omega}{\partial^2 y}\right) \tag{5}$$

Equation (5) describes the progression of a vortex throughout a fluid.

Introducing the stream function in equation (6)

$$u = \frac{\partial \psi}{\partial y} \qquad v = -\frac{\partial \psi}{\partial x} \tag{6}$$

where ψ is the stream function. Combining (4) with (6), results in (7)

$$\frac{\partial^2 \psi}{\partial^2 x} + \frac{\partial^2 \psi}{\partial^2 y} = \omega \tag{7}$$

Equation (7) is the equation on which the solution strategy is based.

3.2 Finite difference approach

Since it is still impossible to find a closed form solution of the Navier-Stokes equations, (2) and (3), (and thus by extension also the vorticity equation (5)), a numerical approach is needed.
A vorticity field can be obtained by using a finite difference scheme (FDS) and an explicit first order time scheme. Both velocity field and the streamlines can be calculated from the vorticity field. The FDS and time scheme are applied to (5), (6), (7). A distinction must be made between the inner nodes of the FDS grid and the boundary nodes. A derivation of the Finite Difference scheme can be found in [2]

Inner nodes

Stream function To derive the FDS for the stream function, the FDS is applied to Eq. (7). This results in (8)

$$\frac{\psi_E - 2\psi_P + \psi_W}{(\Delta x)^2} + \frac{\psi_N - 2\psi_P + \psi_S}{(\Delta y)^2} = \omega_P \tag{8}$$

where ψ_E is the stream function east (negative x as shown in figure 1) to the considered point, ψ_P the stream function at the considered point, ψ_W the stream function west (positive x) to the considered point, ψ_N the stream function north (positive y), ψ_S the stream function south (negative y) and ω_P the vorticity of the considered point.

Eq. (8) can be interpreted as a linear system where ψ_P is the variable. This is the system the computer simulation will solve.

Velocity field A similar approach is used for Eq. (6) to find a FDS for the velocity field.

$$u_P = \frac{\psi_E - \psi_W}{2\Delta y} \qquad v_p = -\frac{\psi_N - \psi_S}{2\Delta x} \tag{9}$$

where u_p and v_p are the discretized velocities in the x and y direction, respectively.

Vorticity The last part of the FDS for the inner nodes is the vorticity. For this, the FDS is applied to Eq. (5). Be aware that the FDS only discretizes spatial dimensions. The time derivative remains analytical for now. This leads to Eq.(10)

$$\frac{\partial \omega}{\partial t} = -u\frac{\partial \omega}{\partial x} - v\frac{\partial \omega}{\partial y} + \boldsymbol{\nu}\left(\frac{\partial^2 \omega}{\partial^2 x} + \frac{\partial^2 \omega}{\partial^2 y}\right) \tag{10}$$

The right hand side of Eq. (10) consists of the stationary term. The first two are convective, the third term is the viscous term. With this in mind, a forward-Euler time scheme can be used to discretize the time derivative which leads to

$$\omega^{t_{n+1}} = \omega^{t_n} + \omega_{stat}\Delta t \tag{11}$$

where ω_{stat} is the stationary term and Δt a finite time step (see section 3.3).
The stationary term of Eq. (11) can be evaluated as

$$\omega_{stat} = -u_p \frac{\omega_E - \omega_W}{\Delta x} - v_p \frac{\omega_N - \omega_S}{\Delta y} + \boldsymbol{\nu}\left(\frac{\omega_E - 2\omega_P + \omega_W}{(\Delta x)^2} + \frac{\omega_N - 2\omega_p + \omega_S}{(\Delta y)^2}\right) \tag{12}$$

Boundary nodes

The boundary conditions for the stream function and vorticity function are derived from the velocity boundary conditions. At every boundary, a condition is needed.

Velocity The inlet boundary conditions are Dirichlet boundaries conditions, where the value of both u and v are prescribed.

$$u_{\vec{inlet}} = [U_0 \;\; V_0] \tag{13}$$

V_0 is chosen to be 0.
The outlet boundary condition states that the flow is fully developed.

$$\left.\frac{\partial u}{\partial y}\right|_{outlet} = 0 \tag{14}$$

On both the walls, a no-slip condition is applied.

$$\vec{u} = \vec{0} \tag{15}$$

Stream function By plugging Eq. (9) into Eq. (20), Eq. (14) and Eq. (15), the boundary conditions for the stream function for the inlet, outlet and wall, respectively.

$$\frac{\psi_N - \psi_S}{2\Delta y} = U_0 \qquad \frac{-3\psi_P + 4\psi_E - \psi_{EE}}{2\Delta x} = 0 \tag{16}$$

$$\frac{\psi_W - 2\psi_P + \psi_E}{(\Delta w)^2} = 0 \tag{17}$$

$$\frac{-3\psi_P + 4\psi_S - \psi_{SS}}{(\Delta y)^2} = 0 \qquad \frac{\psi_E - \psi_W}{2\Delta x} = 0 \tag{18}$$

Vorticity A standard Taylor approximation can now be used to derive the vorticity boundary conditions. The derivation will be done only for the inlet for readability purposes. The other three boundary conditions are derived in the same way.

$$\psi_E - \psi_P = \Delta x \frac{\partial \psi}{\partial x}\bigg|_P + \frac{(\Delta x)^2}{2}\frac{\partial^2 \psi}{\partial x^2}\bigg|_P + \frac{(\Delta x)^3}{6}\frac{\partial^3 \psi}{\partial x^3}\bigg|_P + \mathcal{O}((\Delta x)^4) \tag{19}$$

The first term on the right side can now be replaced by using Eq. (6), the second term can be replaced by using Eq (5) and Eq. (6) and the third term can be replaced by deriving Eq. (5) with respect to x. The fourth order term is neglected. These steps lead to

$$\psi_E - \psi_P = -\Delta x V_0 + \frac{(\Delta x)^2}{2}\left(\omega_P - \frac{\partial u}{\partial y}\bigg|_P\right) + \frac{(\Delta x)^3}{6}\frac{\partial \omega}{\partial x}\bigg|_P \tag{20}$$

Note that the derivative of the inlet velocity in the x-direction is still apparent in this boundary conditions. This is just for the sake of completeness and will be omitted since the inlet velocity in the x-direction is a given constant. V_0 also dissapears since it is proscribed as 0.
Applying the FDS to Eq.(20) and rearranging the equation, a inlet boundary condition for ω is obtained. Since the inlet boundary is being considered, a second order FDS is applied to discretize $\frac{\partial \omega}{\partial x}\Big|_P$.

$$-\frac{\omega_P}{3} - \frac{\omega_E}{6} = \frac{\psi_E - \psi_P}{(\Delta x)^2} + \frac{v}{\Delta x} + \frac{1}{2}\frac{\partial u}{\partial y} + \frac{\Delta x}{6}\frac{\partial^2 u}{\partial x \partial y} \tag{21}$$

In a similar fashion, the other boundary conditions can be derived. For the upper and lower wall, these are given by Eq.(22) and Eq. (23), respectively.

$$-\frac{\omega_P}{3} - \frac{\omega_S}{6} = \frac{\psi_S - \psi_P}{(\Delta y)^2} + \frac{u}{\Delta y} \tag{22}$$

$$-\frac{\omega_P}{3}-\frac{\omega_N}{6}=\frac{\psi_N-\psi_P}{(\Delta y)^2}-\frac{u}{\Delta y} \tag{23}$$

For the outlet boundary, based on Eq. (14)

$$\frac{\omega_P-\omega_W}{\Delta x}=0 \tag{24}$$

3.3 Stability

Everything is almost set up to start the simulation. The only thing missing is the time step Δt. This can not be chosen at random. The stability of a forward Euler scheme is sensitive to the size of the time step. To guarantee stability, stability criteria are needed. Those criteria are based on the geometry of the chosen mesh, on the properties of the fluid and the velocity field. Using these criteria, a time step can be determined. This time step will guarantee that the problem does not 'explode'. An explosion is an amplification of numerical errors which accumulate. Two conditions are proposed by [2] and [3].

Criterion 1

$$\Delta t \leqslant \frac{1}{2\boldsymbol{\nu}}\left[\frac{(\Delta x)^2(\Delta y)^2}{(\Delta x)^2+(\Delta y)^2}\right] \tag{25}$$

Criterion 2

$$\frac{u\Delta t}{\Delta x}+\frac{v\Delta t}{\Delta y}\leqslant 1 \tag{26}$$

Both criteria focus on a different aspect of the simulation. Criterion (25) takes the viscosity into account, where as criterion (26) handles with convection. Now, a time step is chosen which satisfies both criteria.

3.4 Solution Strategy

The solution strategy is based upon the strategy proposed in [4]. The idea they propose is to solve equation (8) iteratively. First,the velocity boundary conditions, the initial velocity field and the initial vorticity field are set. From those, the stream functions boundary conditions are computed as discussed in subsection 3.2. These stream function boundary conditions are the starting point for the time loop. Using Eq. (8), the stream function is evaluated. From here, the boundary conditions for the vorticity field are computed using Eq (21), (22), (23) and (24). Parallel to this, the velocity field is updated using Eq. (9). From the vorticity boundary conditions, the stationary part of the vorticity field can be calculated with Eq. (12). Using the forward Euler scheme, Eq. (11), the vorticity field is computed with time. This result is plugged in back into Eq. (8) to continue the iteration. The time loop will end

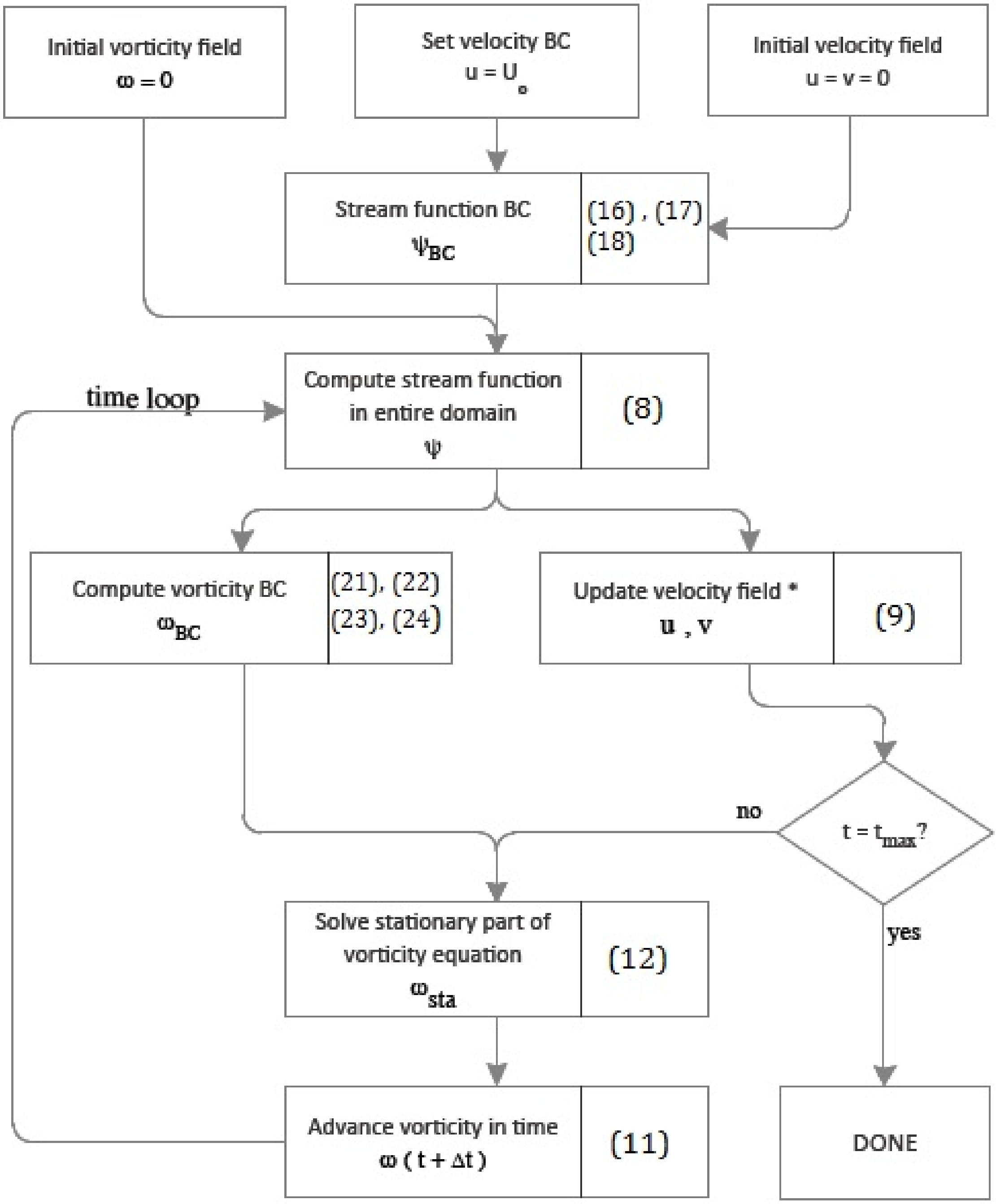


Figure 2. Schematic representation of the solution strategy

when the prescribed maximum number of time steps has been reached. Fig. 2 shows a schematic representation of the solution strategy.

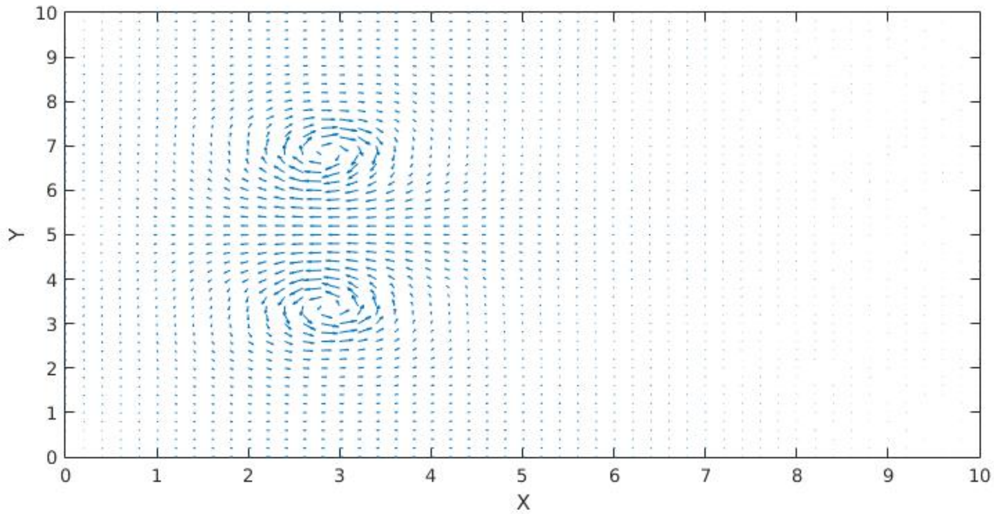


Figure 3. Vector plot of the starting condition (before the first iteration) of the flow field. The two source vortices are oriented in opposite directions.

4. Results and Discussion

To test the code, two basic input flows are used: a Poiseuille profile and a constant profile. Antisymmetric paired sources are added at one quarter into the x direction separating the y direction into thirds. The flow is then allowed to develop over one second of physical time, which means that the entry injected velocity has time to reach the exit of the domain.

4.1 Various Test Cases and Observations

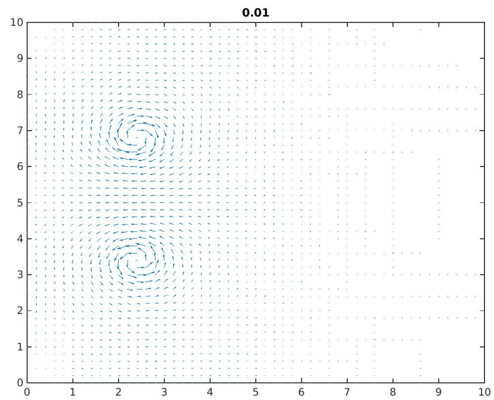
4.1.1 Zero Velocity Initial Conditions

At the start of the project, the code was very unstable and exploded after reaching only 0.001 of the final time. One way to address this issue was to push the Courant-Friedrichs-Lewy-number (26) very small by making the U velocity go to zero. Now, the code maintains its stability for small values of viscosity ν = 1e-6. In Figures 4 and 5 a source with an initial vorticity magnitude of 50 s^{-1} is introduced and maintained until a physical time of $t = 0.10$ seconds at which time the source is removed. The rest of the physical time of the simulation is used to dissipate this vortex.

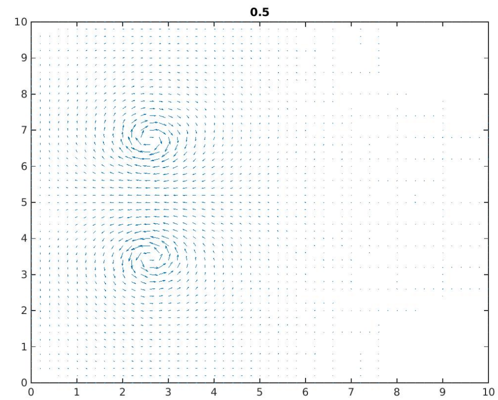
The small difference between the flows shown in Figure 4 and Figure 5 by only changing the ν value is very encouraging and a good indication that the diffusion equations are working properly.

4.1.2 Source Free Constant Input Flows

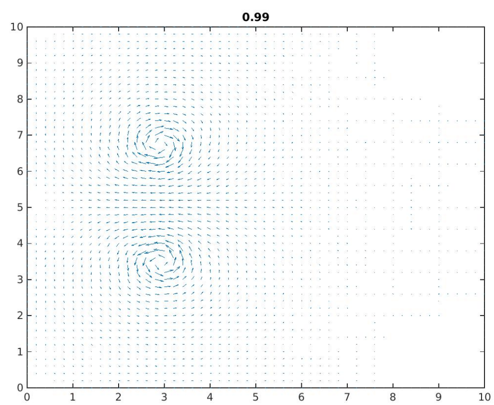
The next step is to see if Poiseuille flow could be developed from a constant input flow. It is expected that the vortex starts at the walls and slowly advances towards the center of the channel while at the same time the velocity should go to zero at the walls to form a parabolic profile.



(a)



(b)

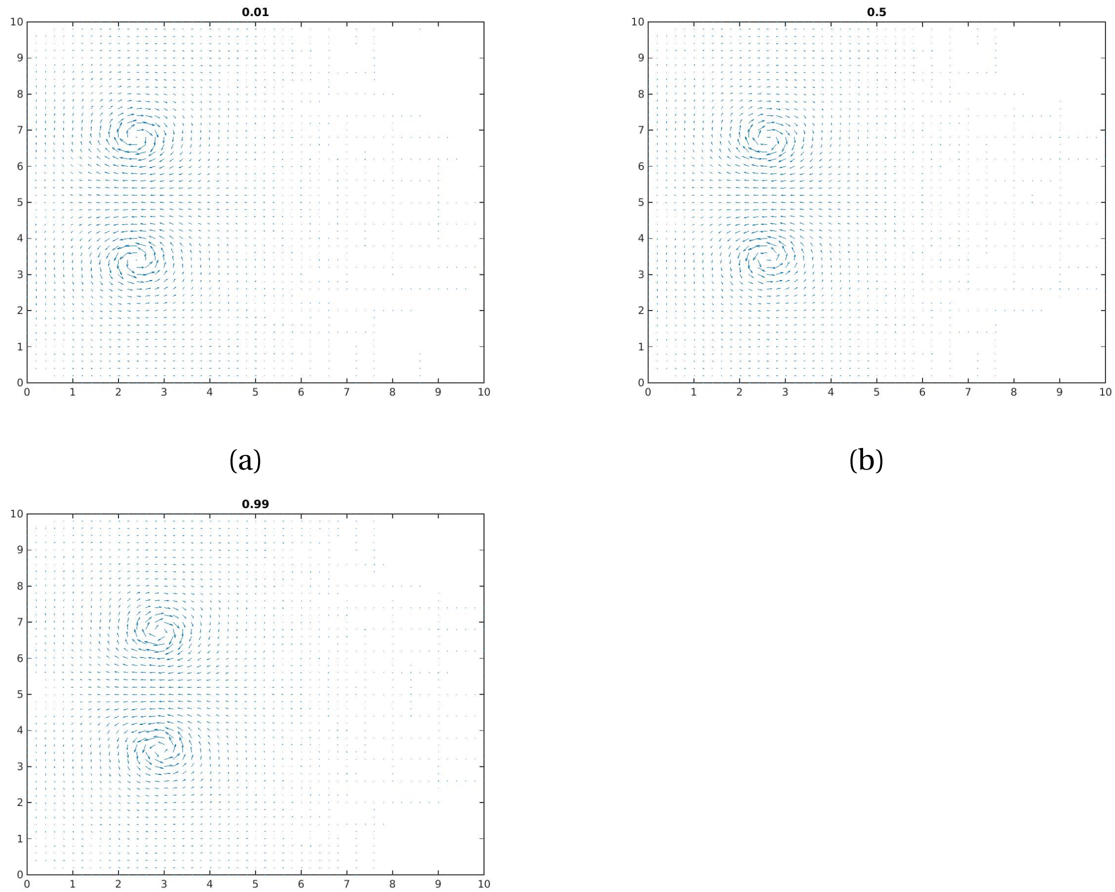


(c)

Figure 4. Development of the flow starting at (a) progressing to (b) and ending at (c)

Vorticity is developing at the walls as expected but does not travel towards the center of the channel throughout the simulation. One method suggested to perhaps find if this might have been due to boundary conditions was to set the initial flow field to zero everywhere and then to inject to input velocity. Unfortunately for this code that suggestion was infeasible because both the U and V velocities are slave variables of the stream function ϕ, which is defined as monotonically increasing. Because the velocities are the gradients of a monotonically increasing function, it is not possible to have a true zero and would involve manually enforcing values onto the stream functions. However, the stream function should be calculated entirely from the initial conditions, which cannot enforce that the stream yields a no-slip boundary condition.

It is also interesting to note that, when a Poiseuille is given as the input condition, it will be maintained and not revert to a constant profile. This enforces the idea that somehow the vorticity is not being transported properly which means the diffusion somehow cannot cross the stream lines.



(a) (b) (c)

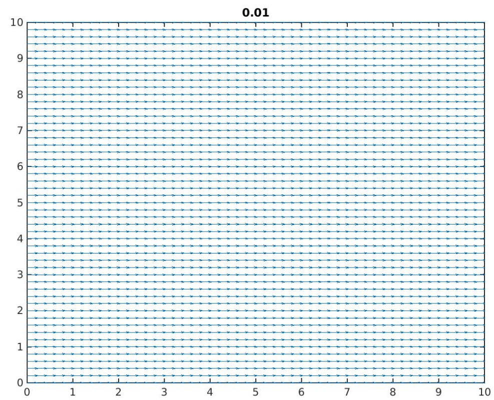
Figure 5. The same flow with an increased ν of 1e-3 dampens out a bit faster than with a viscosity ν of 1e-6, as expected

4.1.3 Convection Flows

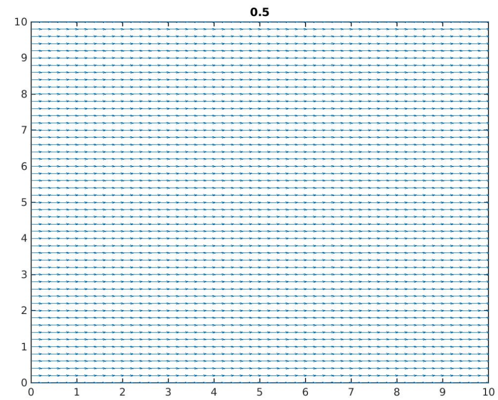
The next logical step is to add convection into the flow and convect vortices along the channel.

In Figure 8 an apparent convection is going against the current of the flow, which is a very troubling development, with no apparent explanations. It is likely related to the dissipation problem mentioned earlier, in which the vorticity is not diffused in the cross-axial direction.

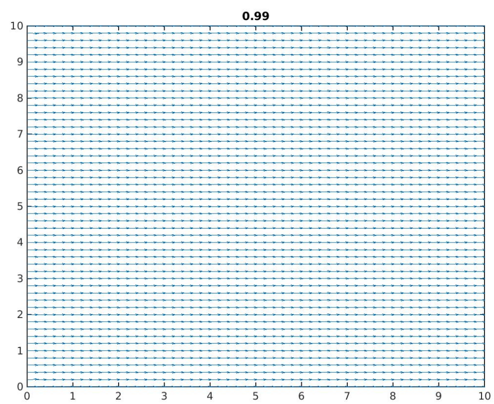
Figure 9 better illustrates the upstream convection, but also shows the the vorticity is in fact being convected downstream as expected at the same time, which is a promising development. The vorticity being convected in the correct direction exhibits a classical checkerboard patterning, which is one of the reasons it is difficult to visualize with a vector plot.



(a)



(b)



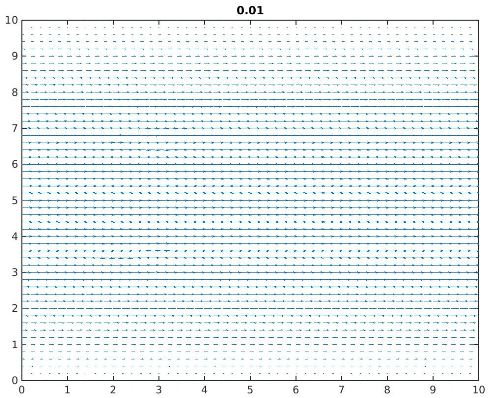
(c)

Figure 6. The aforementioned prediction of flow does not match how the actual simulation behaves: there is no observable change from the constant flow.

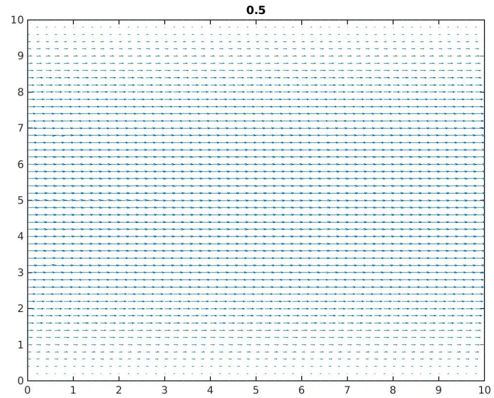
5. Conclusions

It is clear that this project is by no means finished. What is needed, is a clear strategy for debugging which our team was unable to provide. Firstly, there needs to be an investigation on why vorticity generated at the walls does not diffuse into the center of the flow. As mentioned earlier, a good strategy to address this would be to have a zero velocity flow field and then to inject velocity into this field. For this particular code, that requires a major restructuring of how initial conditions are set and handled. The second and possibly more difficult problem to address is that of the upstream convection. Perhaps the only feasible step to debugging this error would be to look at the vorticity that is being corrected properly and try to figure out where the checkerboarding comes from. The downstream flow exhibits checkerboarding while the upstream un-physical flow does not, as shown in Figure [**?**].

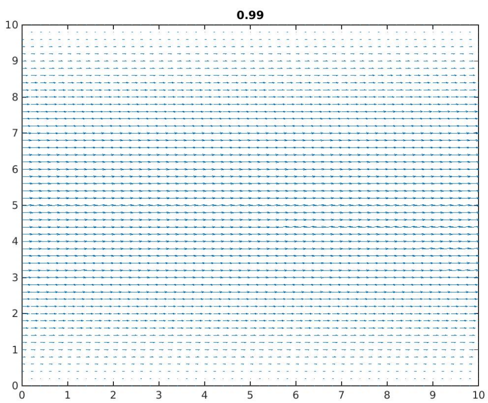
In the long term, after this code can accurately represent convection-diffusion flows



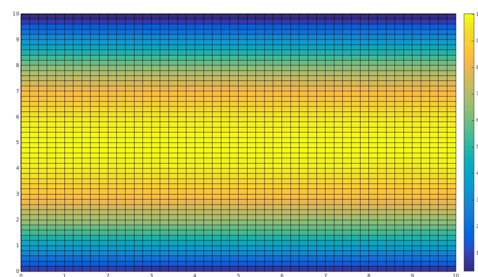
(a)



(b)



(c)



(d)

Figure 7. A given Poiseuille flow is maintained in the exact same way as a given constant profile flow.

with vorticity sources, the derivation of the model could be extended to include frame of reference accelerations. These accelerations would represent correolis effects and allow the model to simulate flows which are not as intuitive as simple convection-diffusion channel flows.

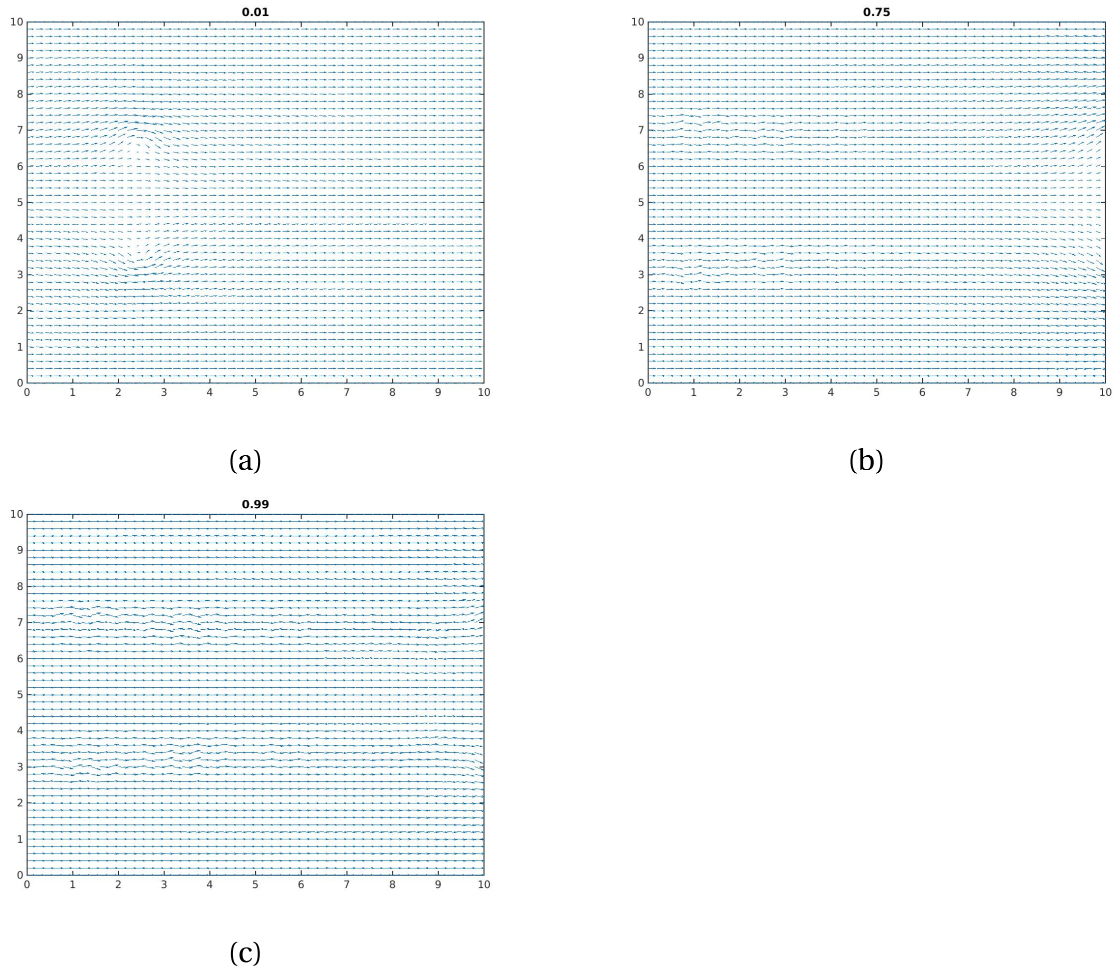
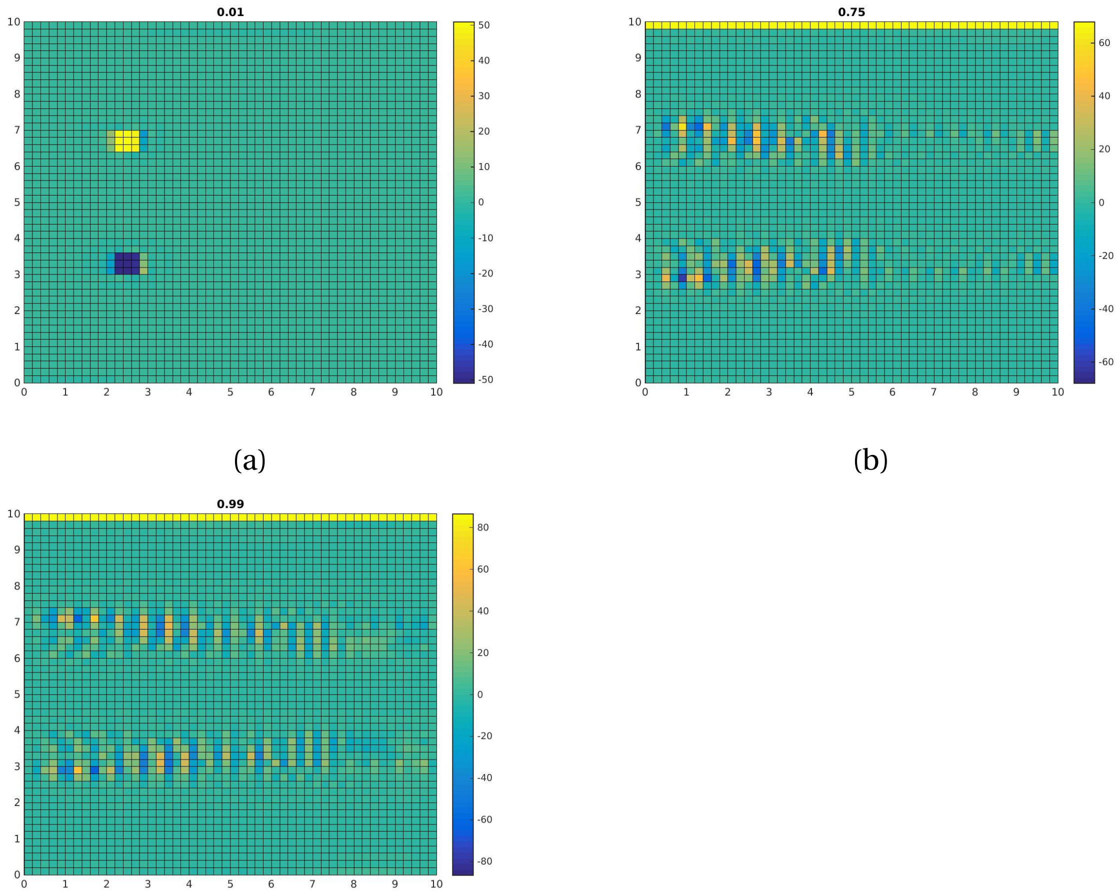


Figure 8. A plot of the vorticity being convected with a constant inflow velocity of 10 m/s.



(a) (b) (c)

Figure 9. The same flow as above except now plotted as vorticity in the z direction, where color corresponds to intensity.

Acknowledgements

We would like to thank Camilo Silva for his unwavering help and patience with us and our project. Without him, we would still be debugging our first lines of code. We would also like to thank Enzo Maier and Tobias Gschnaidtner. Most of our work is based on their paper [4].

References

[1] Massachusetts Institute of Technology. *Marine Hydrodynamics, Lecture 9.* http://web.mit.edu/2.20/www/lectures/lec09/lecture9.pdf.

[2] Camilo F. Silva, Kilian Förner, Wolfgang Polifke. *Notes on Computation Thermo-Fluid Dynamics*. Professur für Thermofluiddynamik Technische Universität München, April 2015.

[3] Volker Gravemeier, *Vorlesungsmanuskript "Finite Elemente in der Fluidmechanik"*. Lehrstuhl für Numerische Mechanik Technische Universität München, Sommersemester 2015.

[4] Enzo Maier, Tobias Gschnaidtner, *Kelvin–Helmholtz instabilities in an axisymmetric sudden expansion*. Professur für Thermofluiddynamik Technische Universität München, Winter Semester 2014-2015.