Dilip Thekkoodan

18.086 Project Report

A 2D Navier Stokes solver for the flow past bluff bodies

Introduction

This report describes a 2D Navier-Stokes solver that was used to study the flow past a square cylinder under different Reynolds numbers. The objectives of this project are:

1. Understanding the general solution process of a projection-based approach
2. Implementing this solver in MATLAB.
3. Implement an inflow/outflow boundary condition.
4. Study the flow past a square cylinder at various Reynolds numbers.

Some key details of this implementation are first covered.

Solution Process

The solution process used in this project is a Chorin-projection based approach as described in [1]. The sample implementation for the driven box problem, available on the CSE website [2], was the basis of this re-implementation.

The Navier-Stokes equations in 2D are:

$$u_t + p_x = -(u^2)_x - (uv)_y + \frac{1}{Re}(u_{xx} + u_{yy})$$

$$v_t + p_y = -(uv)_x - (v^2)_y + \frac{1}{Re}(v_{xx} + v_{yy})$$

$$u_x + v_y = 0$$

The solution at time step ***n+1*** can be computed from that of time step ***n***, with the following three step approach [1]:

1. Computation of the non-linear terms explicitly in time to obtain intermediate velocities (U*,V*):

$$\frac{U^* - U^n}{\Delta t} = -((U^n)^2)_x - \left((U^n V^n)\right)_y$$

By rearranging to the explicit time-integration:

$$U^* = U^n + \Delta t(-((U^n)^2)_x - \left((U^n V^n)\right)_y)$$

Similarly,

$$V^* = V^n + \Delta t(-\left((U^n V^n)\right)_x - ((V^n)^2)_y)$$

2. Computation of the viscosity term implicitly to modify these intermediate velocities. We compute U** from U* and V** from V* through the following equations:

$$\frac{U^{**} - U^*}{\Delta t} = \frac{1}{Re}(U_{xx}^{**} + U_{yy}^{**})$$

$$\frac{V^{**} - V^*}{\Delta t} = \frac{1}{Re}(V_{xx}^{**} + V_{yy}^{**})$$

3. Applying a pressure correction on the intermediate velocity fields (U**,V**) to obtain the final velocities (U,V), by enforcing the incompressibility constraint.

   This involves the following substeps:

   a. Computing the field divergence

   $$D^n = \nabla . U^n$$

   b. Solving a Poisson equation

   $$-\Delta P^{n+1} = -\frac{1}{\Delta t} D^n$$

   c. Computing the gradient of pressure:

   $$G = \nabla P^{n+1}$$

   d. Finally, progressing the velocity to the current time-step:

$$U^{n+1} = U^n - \Delta t - G^{n+1}$$

Staggered Grids & Discretization

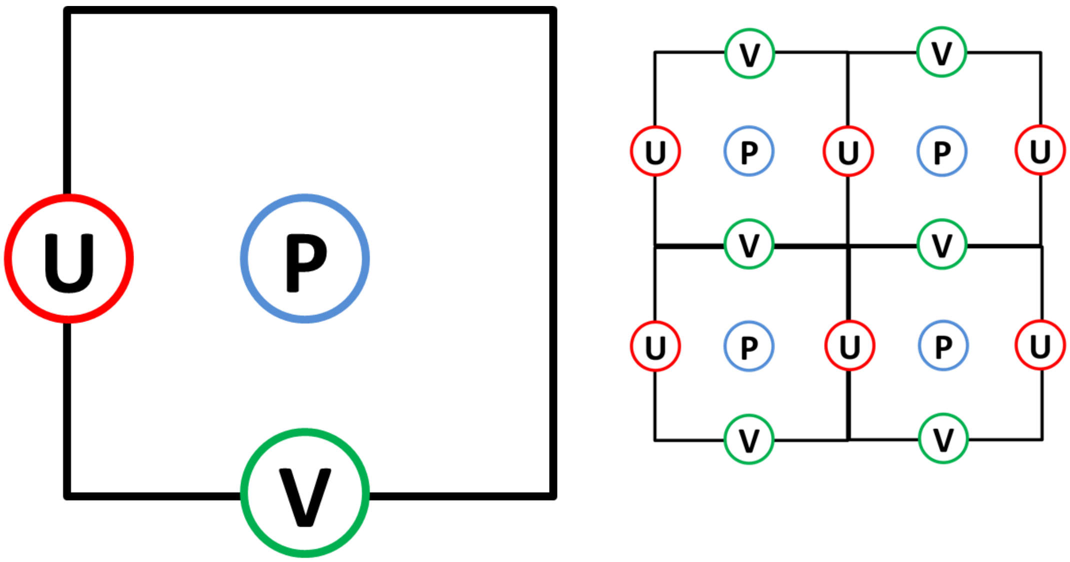


Fig 1: Layout of the staggered grid arrangement showing where the U, V, P values are defined

A staggered grid-approach is used to represent the physical quantities of this problem as shown in Fig.1. The U velocity is defined on the vertical cell boundaries, V-velocity is defined on the horizontal cell boundaries and P is defined in the interior of each cell.

First order derivatives are discretized with a centered differencing scheme. The center-point is chosen to be collocated with the pressure (and used in the pressure correction step), which conveniently removes the instability associated with centered schemes. Similarly, second order derivatives are discretized by a 5-point centered differencing scheme.

A final comment on the treatment of non-linear terms: to handle faster flows, it is necessary to incorporate an upwinding effect in the treatment of non-linear terms. This is achieved by combining an upwinding-type scheme with the centered differences that are used and determining the relative contribution of each by measuring the maximum velocity in the domain. Details are given in [3].

Boundary Points & Boundary Conditions

Dirichlet BC

The Dirichlet boundary condition specifies a value at the boundary and must be directly applied to the corresponding points in the fluid domain. Given the staggered grid approach, applying a Dirichlet boundary condition for U-velocity on the horizontal cell-boundaries and V-velocities on vertical cell-boundaries are not straightforward.

To resolve this, we can express the boundary value as an average of two points, one on either side of the boundary. For instance:

$$U_{boundary,prescribed} = \frac{U_{inside} + U_{outside}}{2}$$

$$U_{inside} = 2U_{boundary,prescribed} - U_{outside} = 2U_{boundary,prescribed}$$

In the second equation $U_{outside}$ is zero as it is no defined outside the boundary. Note that inside refers to a point in the fluid domain, while outside refers to a point outside the fluid domain. In this study, a point inside the bluff body is considered outside the fluid domain.

The Dirichlet BC is encountered in the following instances:

1. For the flow past a bluff body, the body boundaries (four walls of the square) have no-slip and no-penetration constraints (u=v=0), which are Dirichlet BCs.
2. In addition, if the flow is a defined in a channel with the top and bottom boundaries defining the walls, then Dirichlet BCs are also enforced on these walls (u=v=0).
3. If the flow is defined in an unbounded domain, given the constraints of the mesh size, we could define Dirichlet BC (u=freestream velocity and v=0) along the top and bottom boundaries.

Neumann BC

The Neumann boundary condition specifies value for the normal derivative at the boundary point. Neumann BCs encountered in the current problem specifies a zero value for the normal derivative along the boundary. A straightforward way to enforce this is to set the value of the points immediately inside the boundary to that on the boundary. For a boundary point and one that immediately lies inside, this reads:

$$NBC = 0 = \frac{U_{boundary} - U_{inside}}{h}$$

$$U_{boundary} = U_{inside}$$

Neumann BCs are encountered in the following instances:

1. Outflow of the domain, on the right, where both the velocity terms are Neumann type BC with the gradient defined to be zero ([1], p.537).
2. All pressure boundaries are treated with a Neumann BC for the pressure solve step.

Cholesky Matrix Decomposition & Reordering

A Hermitian positive semi-definite matrix A can be decomposed into the form:

$$A = LL^*$$

where, L is a lower triangular matrix with real, non-negative diagonal entries and L* the conjugate transpose of L.

Using a Cholesky decomposition, in conjunction with a minimum-degree reordering for symmetric matrices, is more advantageous than an LU decomposition as the resultant systems are sparser and hence can be solved much more efficiently.

Simulation Cases

Validation of Inflow/Outflow BC

A simple test is done to check if the channel flow is implemented properly. The domain chosen is similar to our test problems and look at the evolution of the u-velocity profile. Poiseuille's flow between parallel plates for a viscous liquid (low Reynolds number) will be correctly indicated if the flow profile becomes parabolic as the flow becomes fully developed. Furthermore, the peak of the parabolic profile is calculated to be 1.5 times the free-stream velocity.

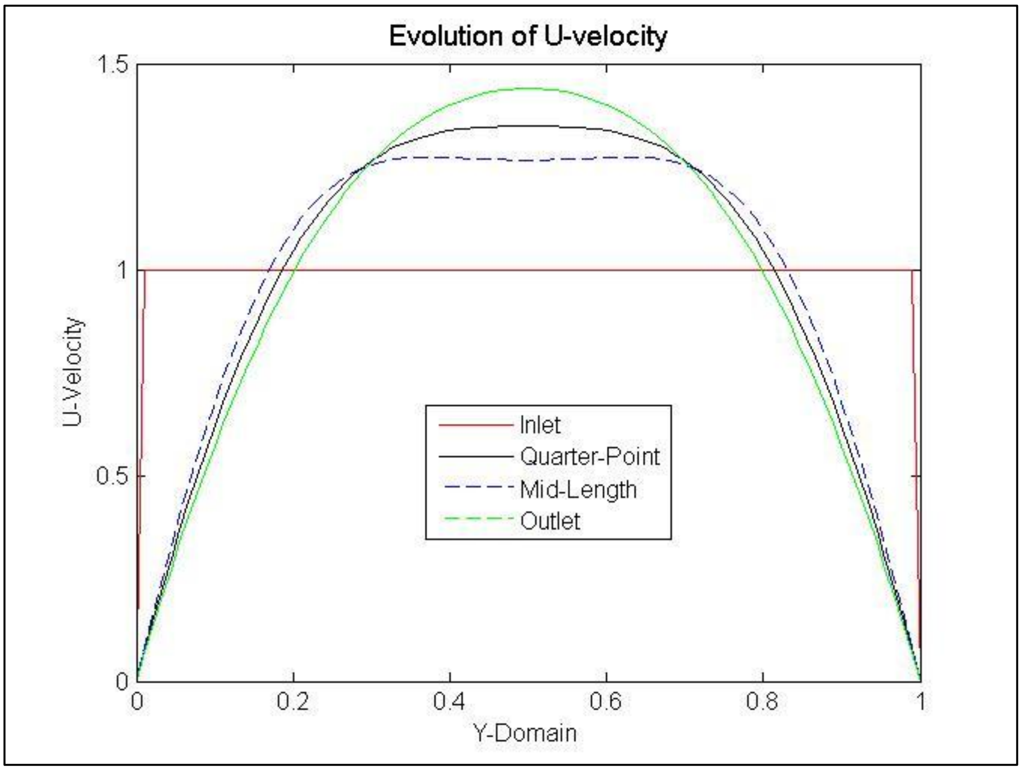


Figure 3: The u-velocity profile of the flow between parallel-plates (the channel) showing the gradual evolution from the uniform free-stream velocity to a fully-developed parabolic profile.

As shown in Fig.3 above, the fully developed profile is parabolic and the peak of the profile is close to the predicted value of 1.5 times the free-stream velocity. This validates the implementation of the inflow/outflow condition.

Problem Geometry

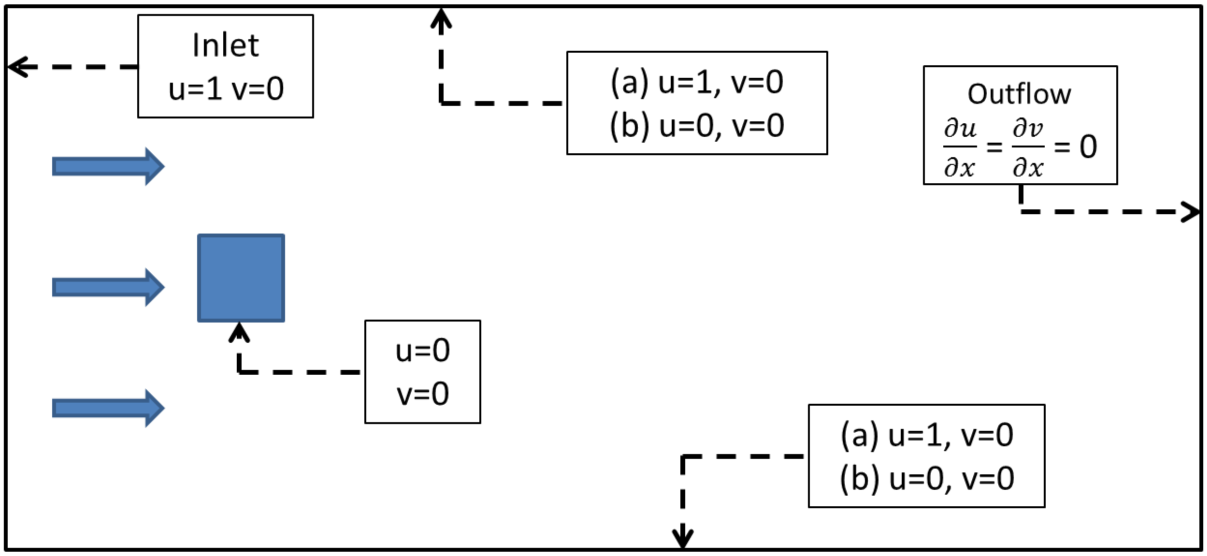


Figure 2: Geometry of the problem indicating the flow direction, the bluff body (square cylinder) and the various boundary conditions on the different boundaries.

The geometry of the problem that is simulated is shown in the Fig.2; the flow enters from the inlet on the left and exits from the outlet on the right. This configuration allows for two problems to be simulated:

a) The flow past a square cylinder in a channel.
b) The flow past a square cylinder in an unbounded domain. This is a heavy approximation, as by forcing the side boundaries to move with the free-stream we are injecting some momentum into the flow around the bluff-body.

We generate streamline patters for the two cases mentioned above for four Reynolds number values: 10, 25, 50 and 100. The Reynolds number is based on the side length of the square, which is set to 0.1. The domain used is 3 units along x-axis and 1 unit along the y-axis.

Vorticity Patterns

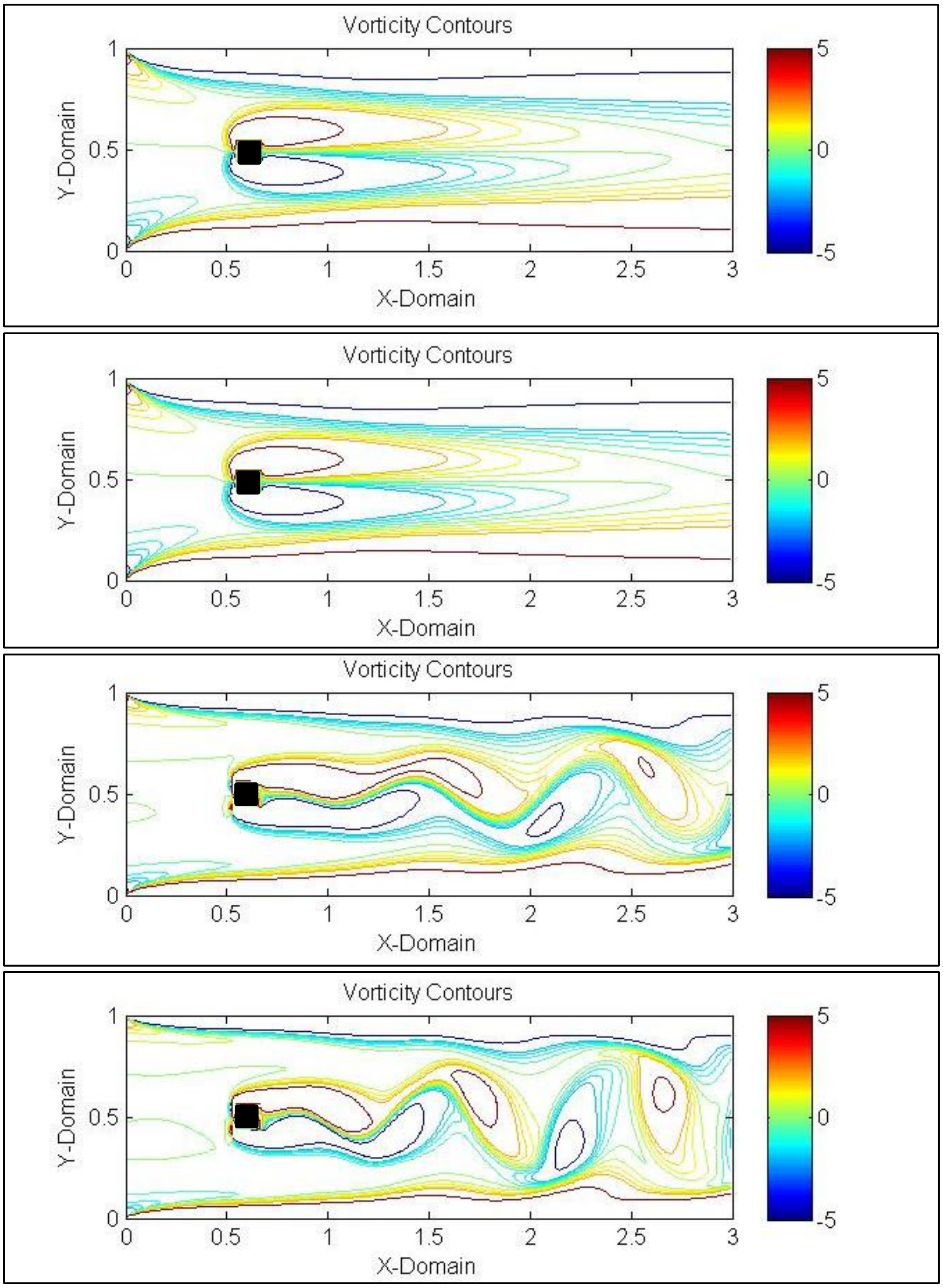


Figure 3: Vorticity pattern for the flow past a square cylinder in a channel: (a) Re=10, (b) Re=25, (c) Re=50 and (d) Re=100

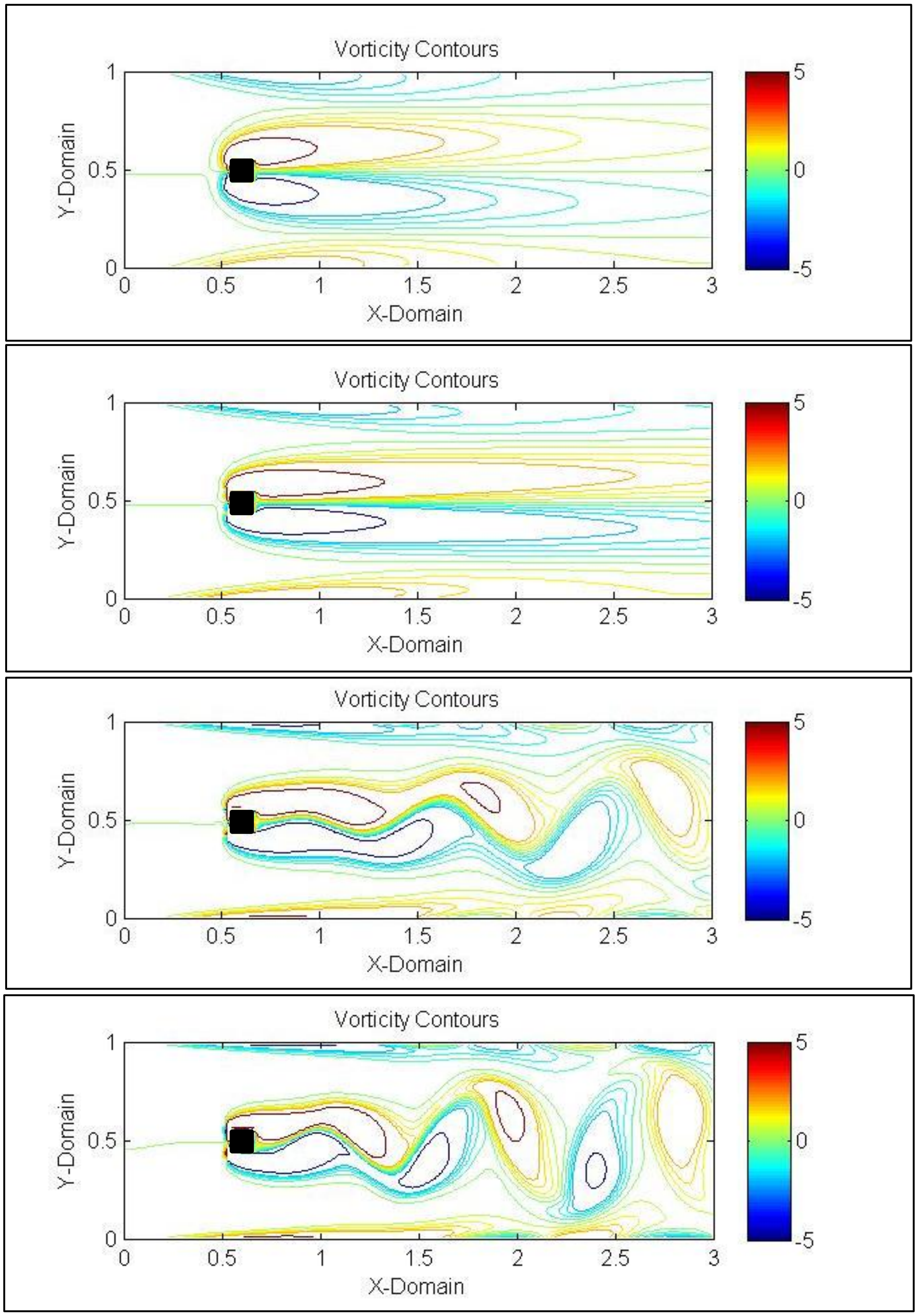
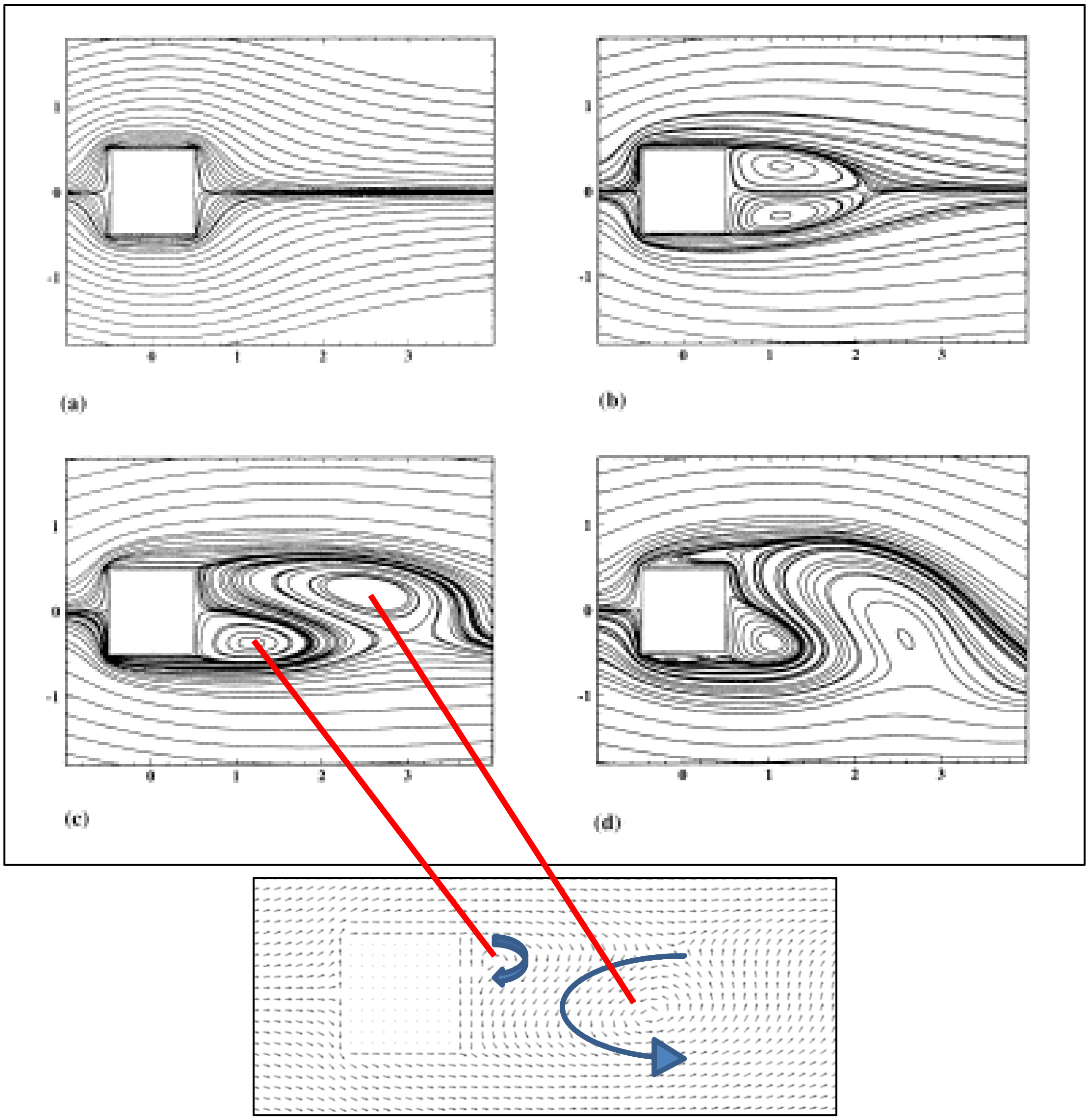


Figure 4: Vorticity pattern for four simulation cases: (a) Re=10, (b) Re=25, (c) Re=50 and (d) Re=100

Comparison with Simulations from Literature



(e)

Figure 5: Figure adapted from [4], showing the streamline pattern for a flow past a square cylinder at various Reynolds numbers: (a) Re=1, (b) Re=30, (c) Re=60 and (d) Re=200 and (e) streamline pattern generated from this project for Re=60.

A qualitative comparison with the results published by Breuer et.al [4] is done here. As shown in [3], the onset of the production of a Karman street is around Re=60. In the simulation cases performed (Figs.4-5) with the solver developed in this project, a similar behavior is observed: low Re cases do not indicate vorticity patches that is reminiscent of the Karman street. Starting with Re=50, the street with alternating patches of vorticity is visible. Fig.6 also demonstrates a qualitative agreement of the flow-patterns for the Re=60 case.

A qualitative comparison is only a preliminary step in validating a code and as such it cannot be claimed based on what is presented that the solver works well for the flow past a square.

Conclusion

The following were achieved in this project:

1. An implementation of a Navier-Stokes 2D solver, employing a projection-based approach on a staggered grid.
2. The implementation of the inflow/outflow boundary condition and a sample validation against the fully developed flow-profile of Poiseuille's flow.
3. An implementation of the flow-past a square cylinder in a channel and also in an approximation of an unbounded flow. A qualitative comparison is presented as a first step to more rigorous validation.

References

[1] Strang, G, *Computational Science and Engienering*, 2007, Wellesley-Cambridge Press.

[2] Seibold, B., *A compact and fast Matlab code solving the incompressible Navier-Stokes equations on rectangular domains*, 2008

[3] 18.086 Course Website, *Computational Science and Engineering,* math.mit.edu/cse, Accessed on 04/30/14', Last updated 04/18/04'

[4] Breuer, M., Bernsdorf, J., Zeiser, T., Durst, F., *Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume*, International Journal of Heat and Fluid Flow, 21(2000).