

## CFD ANALYSIS OF 2D STEADY FLOW OVER A SQUARE CYLINDER USING SIMPLE ALGORITHM

Lalit Yadneshwar Attarde

ME21S034

Department of Mechanical Engineering, IIT Madras  
me21s034@smail.iitm.ac.in

G V Rama Jaswanth Voleti

ME21M018

Department of Mechanical Engineering, IIT Madras  
me21m018@smail.iitm.ac.in

### ABSTRACT

*The present work models a 2D flow around a square cylinder using a finite volume approach with Semi Implicit method for Pressure Linked Equations. The analysis focuses on the effect of low Reynolds number, as determined by the input velocity parameters, on the flow around the square cylinder. Velocity profiles along with an integral drag coefficient are investigated for laminar flows, and validated with the literature.*

### NOMENCLATURE

$\mathbf{u}$	Velocity vector
$u$	x velocity
$v$	y velocity
$p$	scalar pressure
$\mathbf{i}$	x direction unit vector
$\mathbf{j}$	y direction unit vector
$\mu$	diffusion constant
$\rho$	density
$Re$	Reynolds number
$\nabla$	Divergence operator
$\phi$	unit quantity in transportation equation
$\Gamma$	Diffusion coefficient
$u_b$	boundary velocity
$a_{nb}$	neighbor coefficient
$u_{nb}$	neighbor velocity
$V$	volume
$u^*$	velocity guess
$u'$	velocity correction

### INTRODUCTION

The flow over square cylinder bluff body in a low speed flow has various applications, ranging from aerodynamics, wind engineering, and electronics. There is a difficulty associated with the sequential solution of the continuity and momentum equations for incompressible flow. If we intend to use the continuity equation to solve for pressure, we encounter a problem for incompressible flows because the pressure does not appear in the continuity equation directly. The density does appear in the continuity equation, but for incompressible flows, the density is unrelated to the pressure and cannot be used instead. If sequential, iterative methods are to be used, it is necessary to introduce pressure into the the continuity equation. Methods which use pressure as the solution variable are called pressure-based methods. One popular method among these pressure-based method is SIMPLE[1] method.

### PROBLEM DESCRIPTION

The domain is a rectangular duct with a square cylinder centered on y-axis. The geometry is made to match the domain used in and is as shown in Fig Fig1 where  $D = 0.01$ ,  $H = 0.1$ ,  $L_h = 0.15$ ,  $L = 0.6$ . The inlet is a constant velocity inlet using a uniform profile with a value of  $u = 1$ , the outflow at the outlet is upwinded using infinite Peclet number assumption and the walls are all no slip boundaries. The case was run using  $Re$  of 0.1, 0.5, 1, 2, 5. Where  $Re$  is calculated using the velocity inlet and the length scale  $D$  where  $Re = Du/\mu$

## GOVERNING EQUATIONS

The governing equations for this problem are the 2D steady, Navier-Stokes equations, which are nothing but the continuity equation and the momentum equations in two directions as given below:

$$\nabla \cdot (\rho u) = 0 \quad (1)$$

$$\nabla \cdot (\rho V u) = \nabla \cdot (\mu \nabla u) - \nabla p \cdot i + S_u \quad (2)$$

$$\nabla \cdot (\rho V v) = \nabla \cdot (\mu \nabla v) - \nabla p \cdot j + S_v \quad (3)$$

The source terms  $S_u$ ,  $S_v$  appearing in above equations for a Newtonian fluid can be written as

$$S_u = f_u + \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v}{\partial x} \right) - \frac{2}{3} \frac{\partial}{\partial x} (\mu \nabla \cdot V) \quad (4)$$

$$S_v = f_v + \frac{\partial}{\partial y} \left( \mu \frac{\partial v}{\partial y} \right) + \frac{\partial}{\partial x} \left( \mu \frac{\partial v}{\partial x} \right) - \frac{2}{3} \frac{\partial}{\partial y} (\mu \nabla \cdot V) \quad (5)$$

But for the transport of any scalar  $\phi$  we have a general transport equation which is given as follows:

$$\frac{\partial}{\partial t} (\rho \phi) + \nabla \cdot (\rho V \phi) = \nabla \cdot (\Gamma \nabla \phi) + S \quad (6)$$

In a steady state case the general transport equation reduces to the convection term balanced by diffusion and source terms as follows:

$$\nabla \cdot (\rho V \phi) = \nabla \cdot (\Gamma \nabla \phi) + S \quad (7)$$

Comparing the u-momentum equation 2 to general transport equation 7 we get  $\phi = u$ ,  $\Gamma = \mu$ , and  $S = S_u - \partial p / \partial x$ . Further in the present case viscosity ( $\mu$ ) and density ( $\rho$ ) of the fluid are constant, and assuming that there are no body forces acting the flow domain, the source terms  $S_u$  and  $S_v$  are zero. This can be shown by invoking the continuity equation for compressible fluid  $\nabla \cdot V = 0$  in eqs. 4, 5.

The Navier-Stokes equations are nonlinear (because of the convection term) and they are coupled because the u-momentum equation has v velocity in it and vice versa). In order to solve these equations we have to linearize the convection terms and the equations are decoupled using the current iterate values and are iterated till convergence using SIMPLE algorithm.

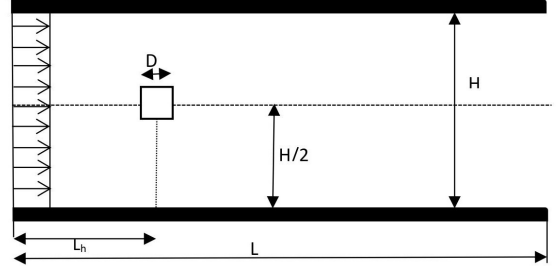


FIGURE 1: Problem Domain

## DISCRETIZATION

In the finite volume method the first step is to integrate the governing equations on the control volume. When the continuity equation is integrated on a cell P, the resulting equation does not contain the velocity for cell P. Consequently, a checker-board velocity pattern can be sustained by the continuity equation. Checkerboarding means the solution could converge to a value whose alternate cell values are equal but this kind of solution is not physically possible. Of the momentum equations can sustain this pattern, the checker boarding would persist in the final solution. Since the pressure gradient, is computed as a part of the solution, it is possible to create pressure fields whose gradients exactly compensate the checkerboarding of momentum transport implied by the checkerboard velocity field. Under these circumstances, the final pressure and velocity fields would exhibit checkerboarding.

A popular remedy for the checkerboarding is the use of a staggered mesh. This Staggered mesh consists of two types on control volumes namely main cell or main control volume and the staggered cell or staggered control volume. The pressure is stored at the centroids of the main cells. The velocity components are stored on the faces of the main cells and are associated with the staggered cells. The u velocity control volumes which are staggered from the pressure control volumes by  $(\Delta x/2)$  in x-direction and the v-velocity control volumes which are staggered from the pressure control volumes by  $(\Delta y/2)$  in y-direction. Because the control volumes are different for u and v momentum equation discretizations the corresponding coefficients will turn out to be different.

The discretized equations are:

$$a_{i,j}^u u_{i,j} = \sum a_{nb}^u u_{nb} + \Delta y (P_{i-1,j} - P_{i,j}) \quad (8)$$

$$a_{i,j}^v v_{i,j} = \sum a_{nb}^v v_{nb} + \Delta x (P_{i,j-1} - P_{i,j}) \quad (9)$$

In the above equations we have used the hybrid differencing scheme (by Spalding, 1972) for the both the velocity equations

8,9. This hybrid differencing scheme is based on a combination of central and upwind differencing schemes. The central differencing scheme which is second order accurate is implemented for small Peclet numbers ( $Pe < 2$ ) and upwind scheme, which is first order accurate, is employed for large Peclet numbers ( $Pe \geq 2$ ). The hybrid differencing scheme uses piecewise formulae based on the local Peclet number to evaluate the net flux through each control volume face.

On the right hand side of the equation 8 we have the source term, for u-momentum equation this can be evaluated as below:

$$S\Delta V = -\frac{\partial p}{\partial x}\Delta y\Delta x = (p_{i-1,j} - p_{i,j})\Delta y \quad (10)$$

### Application of Boundary Condition

The boundary conditions imposed in the problem are briefly described in this section. The inflow is prescribed with a specific inlet velocity  $u_{inlet}$  and  $v_{inlet}$ . In this problem the  $u_{inlet}$  is taken equal to 1m/s and to be entering the domain as uniform flow, and  $v_{inlet}$  is taken as zero.

The outflow is set with a Neumann Boundary condition, described by  $\partial u/\partial x = \partial v/\partial x = 0$ . This implies that the flow velocity does not change across the outflow, or in other terms the flow is developed at the outlet. The velocities directly adjacent to the outlet are set to be equal in this case. It should also be noted that a global mass conservation should be imposed on the outlet.

The wall boundaries are handled by using a ghost cell across the physical boundary. As the solver sweeps across the physical domain the values across the boundaries are used in the solution and then updated after each iteration. The values for the u and v velocity are assigned across the boundaries [2]. The opposite values are assigned to the outer cell resulting in a value of zero along the boundary. Along the faces where the staggered cell center is on the boundary the set value is used. The pressure boundary outside values are set to the same as the cells along the boundary resulting in  $\partial p$  being equal to zero along all boundaries. At exit boundary, the pressure is considered as zero.

The cylinder should also be considered as a no-slip wall. The boundaries for the square cylinder are handled in the same manner as the outer boundaries for velocity and pressure. The cells inside could be set to arbitrary values as no information is transferred between the inside and outside. Similarly for pressure, no information is being obtained from inside the cylinder.

### SOLUTION METHOD

In SIMPLE algorithm, the discrete momentum equations are substituted into the discrete continuity equation. Then by proposing corrections to velocity and pressure a pressure correction equation is derived. As shown in the flow chart, first we guess

the velocity and pressure fields over the domain. Now using these guess values we solve the discrete momentum equations 8,9 and obtain  $u_e^*, v_n^*$ . Now these newly computed starred velocities satisfy momentum equations for the guessed pressure field but they do not satisfy the continuity equation since the guessed pressure is not the correct pressure field. So, corrections are proposed for velocity and pressure as follows:

$$u = u^* + u' \quad (11)$$

$$v = v^* + v' \quad (12)$$

$$p = p^* + p' \quad (13)$$

$$u'_{i,j} = d_{i,j}(p'_{i-1,j} - p'_{i,j}) \quad (14)$$

SIMPLE algorithm demands that these corrected values should satisfy the discrete continuity equation. Substituting these proposed corrected velocities 11 into discrete continuity equation a discrete equation for pressure correction  $p'$  can be derived which reads as follows:

$$a_P p'_P = \sum_{nb} a_{nb} p'_{nb} + (F_w^* - F_e^* + F_s^* - F_n^*) \quad (15)$$

$$a_E = \rho_e d_e \Delta y \quad (16)$$

$$a_W = \rho_w d_w \Delta y \quad (17)$$

$$a_N = \rho_n d_n \Delta x \quad (18)$$

$$a_S = \rho_s d_s \Delta x \quad (19)$$

$$a_P = \sum_{nb} a_{nb} \quad (20)$$

The Gauss-Seidel method is used for converging both u,v momentum equations, whereas the pressure correction equation is converged using line-by-line TDMA (Tri Diagonal Matrix Algorithm). The reason behind this choice is based on the equation environments. In case of the pressure correction equation, the number of equations are large and so to reduce the number of iterations, we have used line-by-line TDMA method. This reduced the iterations from 30000 (guass-seidel) to 5000 (considerably lower number) for convergence. On the contrary, the momentum equations doesn't require a large no.of iterations to converge

making it viable to proceed with Gauss-Seidel method. The flow chart of SIMPLE algorithm is shown in Fig 2.

In the SIMPLE algorithm, the pressure correction equation eq.14 is solved rather than the pressure itself. The omission of  $\Sigma_{nb} a_{nb} u'_{nb}$  and  $\Sigma_{nb} a_{nb} v'_{nb}$  terms in driving the pressure correction equation is of no consequence as far as the final solution is concerned. Only discrete momentum equation, and the discrete continuity equation determine the final answer.

The dropping of the terms does have consequences for rate of convergence. The u-velocity correction given by the eqn. 14 is obtained after the removal of the term  $\Sigma_{nb} a_{nb} u'_{nb}$ . This places the entire burden of correcting the u-velocity component upon the pressure correction. The resulting corrected velocity will satisfy the continuity equation all the same, but the resulting pressure is over-corrected. Due to this over-correction of pressure, the SIMPLE algorithm is prone to divergence unless under-relaxation is used. The momentum equations are under-relaxed as mentioned in eqn. 21., while the pressure can be under-relaxed in a conventional way given by eqn. 22.

$$\frac{a_{i,j} u_{i,j}}{\alpha} = \Sigma_{nb} a_{nb} u_{nb} + \Delta y (p_{i-1,j} - p_{i,j}) + \frac{(1-\alpha)}{\alpha} a_{i,j} u_{i,j}^0 \quad (21)$$

Similarly the pressure correction is under-relaxed when adding them to the starred values as follows:

$$p = p^* + \alpha_p p' \quad (22)$$

In the present simulations we have taken  $\alpha = 0.9$ , and  $\alpha_p = 0.001$ .

## RESULTS

The velocity profiles are obtained and plotted across the flow domain. The u-velocity profile for various Re are given in figure 4. The coefficient of drag is obtained and is compared with the literature [3]. The results are quite in the range of the experimentally obtained values. For  $Re=0.1$ , proper validation is not available. The stokes hypothesis on flow over bluff bodies at very low Re is taken for validation. The obtained resultant coefficient of drag for  $Re = 0.1$ , is in good agreement with the value obtained from Stokes hypothesis. The obtained drag coefficients are plotted, and the comparison is shown in figure 3.

The staggered mesh worked to eliminate checker boarding but is only applicable on simple geometries and use of the col-located scheme is more robust in regards to mesh complexity. The SIMPLE algorithm, however, is robust and provides quality solutions to incompressible flows. This algorithm have the advantage of low, storage, and reasonably good performance over a broad range of problems. However, they are known to require a large number of iterations on problems with large body forces

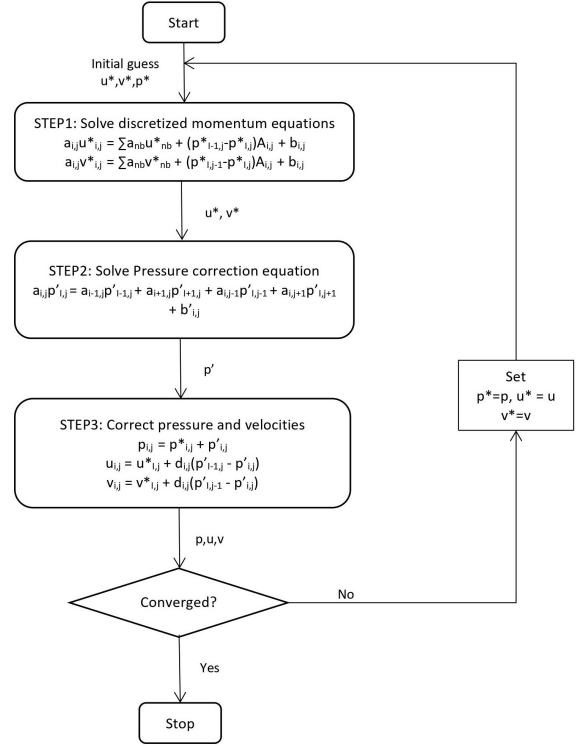


FIGURE 2: Flowchart of SIMPLE algorithm

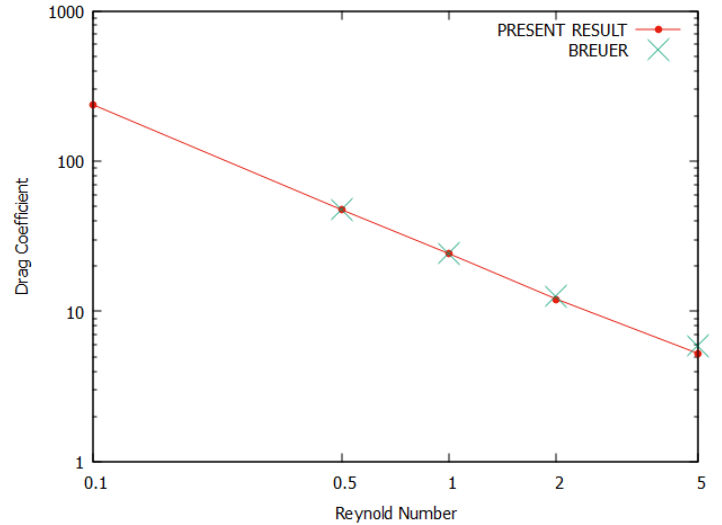
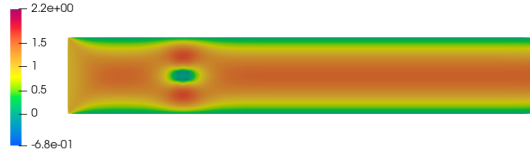
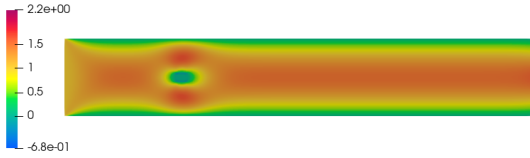


FIGURE 3: Drag Coefficient vs Re

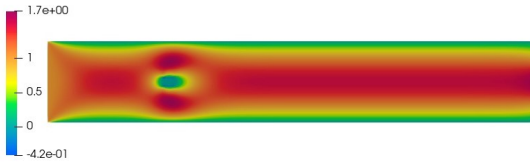
resulting from buoyancy, swirl and other agents. In strongly non-linear case divergence may occur despite under-relaxation. Many of these difficulties are a result of sequential nature of the momentum and continuity solutions which are strongly coupled through the pressure gradient term when strong body forces are present.



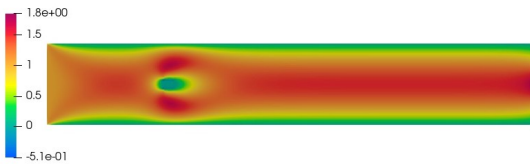
(a)  $Re = 0.1$



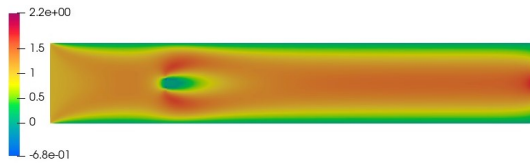
(b)  $Re = 0.5$



(c)  $Re = 1$



(d)  $Re = 2$



(e)  $Re = 5$

**FIGURE 4:** u-velocity profiles

## CONCLUSIONS

SIMPLE solver has been implemented on steady incompressible 2D flow around square cylinder using staggered grid, for  $Re$  ranging from 0.1 to 5. Coefficient of drag for the considered  $Re$  is seen to be in good agreement with the reference values from the literature[3]. Other than some minor difficulties in handling the effect of corrected pressure, SIMPLE algorithm is efficient in solving this problem.

SIMPLER technique could be implemented for obtaining better convergence and to do deal with the consequence of incorrect pressure guess on good velocity guess.

## REFERENCES

1. Suhas V. Patankar, Textbook, Numerical Heat Transfer and Fluid Flow
2. Fadl Moukalled, Luca Mangani, Marwan Darwish (2016) Implementation of boundary conditions in the finite-volume pressure-based method—Part I: Segregated solvers, Numerical Heat Transfer, Part B: Fundamentals, 69:6, 534-562, DOI: 10.1080/10407790.2016.1138748
3. Breuer, M, Bernsdorf, J., and T. Zeiser, F. D., 1999. "Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice boltzmann and finite-volume"
4. H K Versteeg, W Malalasekara, Textbook, An Introduction to computational Fluid Dynamics, Finite Volume Method

## ACKNOWLEDGMENT

Support from Prof. Kameswararao Anupindi.