

Validation of a Fast Transient Solver based on the Projection Method

Darrin W. Stephens ^{*1}, Chris Sideroff², and Aleksandar Jemcov³

¹Applied CCM Pty Ltd, Dandenong, Victoria 3175, Australia.

²Applied CCM Canada, Ottawa, ON K1J 6K3, Canada.

³Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA.

May 15, 2015

Abstract

This paper presents a fast transient solver suitable for the simulation of incompressible flows. The main characteristic of the solver is that it is based on the projection method and requires only one pressure and momentum solve per time step. Furthermore, advantage of using the projection method in the formulation is the particularly efficient form of the pressure equation that has the Laplacian term depending only on geometric quantities. This form is highly suitable for the high performance computing that utilises the Algebraic Multi-grid Method (AMG) as the coarse levels produced by the algebraic multi-grid can be stored if the grid is not changing. Fractional step error near the boundaries is removed by utilising the incremental version of the algorithm. The solver is implemented using version 5.04 of the open source library, Caelus [Applied CCM (2015)]. Accuracy of the solver was investigated through several validation cases. The results indicate the solver is accurate and has good computational efficiency.

Keywords: Projection Method; Fractional Step Algorithm; Algebraic Multi-grid Solver; Transient solutions; Validation

Introduction

Fast transient solution of incompressible turbulent flows still occupy a dominant place in engineering computations. Majority of the solvers that use finite volume method on unstructured meshes with the arbitrary number of faces use either transient SIMPLE [Patankar and Splading (1972)] or PISO algorithm [Issa (1985)]. While these algorithms are known to produce spatially and temporally accurate solutions, they are not particularly efficient due to algorithmic constraints. Typically they require a multiple solution of the pressure equation (PISO algorithm) or multiple momentum and pressure solves per time step (SIMPLE algorithm).

*Corresponding author: d.stephens@appliedccm.com.au

Multiple solutions of the pressure pressure and/or momentum equation per time step removes the fractional step error due to splitting of equations and recovers the time accuracy. Unlike SIMPLE and PISO algorithms, the projection algorithm introduced originally by [Chorin (1967)], does not require multiple pressure and momentum solves per time step. In this paper we describe an efficient implementation of the projection algorithm that utilises the Algebraic Multi-grid Method for the pressure equations suitable for high fidelity transient solutions.

Governing equations and boundary conditions

Incompressible Navier-Stokes equations are given by the following set of partial differential equations:

$$\partial_t u_i + u_j \partial_j u_i = -\partial_i \hat{p} + \nu \partial_{ii} u_i. \quad (1)$$

It should be noted that the density is absorbed in the pressure field since it is considered to be constant in incompressible flow, i.e. $\hat{p} = p/\rho$. Discretization in time of Eq. (1) yields time discretized Navier-Stokes equations with ignored pressure term

$$\delta_t^{*,n} u_i = -u_j^n \partial_j u_i^n + \nu \partial_{ii} u_i^n. \quad (2)$$

Here symbol $\delta_t^{*,n}$ is a finite difference applied in time coordinate

$$\delta_t^{*,n} u_i = \frac{u^* - u^n}{\Delta t}, \quad (3)$$

where u^* is the intermediate velocity field from which the pressure effects were removed, and superscript n denotes the previous time level. This step is called a momentum predictor step or simply predictor step in the language of fractional step methods. Introduction of the intermediate velocity u^* is the part of the algorithm in which the Hodge decomposition was performed in order to compute solenoidal free velocity field. The Hodge decomposition in this case becomes

$$u^* = u_{sol} + u_{irrot} = u^{n+1} + \nabla \phi, \quad (4)$$

where $u_{sol} = u^{n+1}$ is the incompressible flow field we are solving for and $u_{irrot} = \nabla \phi$ is the irrotational part of the flow field that is computed from the scalar potential.

The scalar potential ϕ is obtained using the second part of the fractional step algorithm in which pressure gradient is brought to bare on the intermediate velocity field:

$$\delta_t^{n+1,*} u_i = \partial_i \hat{p}^{n+1}, \quad (5)$$

where $\delta_t^{n+1,*}$ denotes the following finite difference in time

$$\delta_t^{n+1,*} u_i = \frac{u^{n+1} - u^*}{\Delta t}. \quad (6)$$

The Hodge decomposition now becomes

$$u^* = u^{n+1} + \Delta t \partial_i \hat{p}^{n+1}. \quad (7)$$

Since the field u^{n+1} is solenoidal, application of the divergence operator to Eq. (7) produces the equation for the scalar potential

$$\partial_{ii}\hat{p}^{n+1} = \frac{1}{\Delta t}\partial_i u_i^*. \quad (8)$$

Eq. (8) is used to compute the pressure field that can be used in the corrector step to produce the solenoidal field u^{n+1} :

$$u^{n+1} = u^* - \Delta t \partial_i \hat{p}^{n+1}. \quad (9)$$

Therefore, the scalar potential ϕ in the Hodge decomposition in Eq. (7) is identified to be

$$\phi = \Delta t \hat{p}^{n+1}. \quad (10)$$

The fractional step error near the boundaries of the computational domain is removed by reintroducing the pressure in the discretized momentum equation:

$$\delta_t^{*,n} u_i = -u_j^n \partial_j u_i^n + \nu \partial_{ii} u_i^n - \partial_i \hat{p}^n. \quad (11)$$

This practice is commonly called incremental pressure method and is shown to remove the fractional step errors near the boundaries [Strikwerda and Lee (1999)].

Method of solution

With the fractional steps identified above, the solution algorithm takes the following form:

1. Compute intermediate velocity field u_i^* according to Eq. (2) (momentum predictor step)
2. Compute pressure field $\hat{p}$ from Eq. (8)
3. Compute divergence free velocity field u_i^{n+1} according to Eq. (9)
4. Repeat the process in the next time step

The most expensive part of the projection algorithm corresponds to the linear solution of the pressure equation given by Eq. (8). The structure of the pressure equation is strongly elliptic thus requiring the multi-level solver such as Algebraic Multi-grid Method (AMG) for efficiency reasons. The expensive part of the algorithm corresponds to the agglomeration procedure in which the coarse matrix hierarchy is created using the heuristic rules for determining the link strengths among fine equations. One advantage of the projection algorithm is that the coefficients in the Laplacian in Eq. (8) are purely geometrical. Discretization coefficients entering the discrete matrix on fixed grids are constant thus allowing the coarse hierarchy to be computed only once. Since the geometric coefficients do not depend on any other variables in the discretization process, this does not constitute any approximation beyond the usual approximations associated with the discretization process.

Given the geometric nature of the Laplacian in Eq. (8), the algorithm can be modified as follows:

1. Compute coarse mesh hierarchy for Eq. (8) and store it for the duration of the simulation
2. Compute intermediate velocity field u_i^* according to Eq. (2) (momentum predictor step)
3. Compute pressure field p from Eq. (8) by using the stored hierarchy of coarse matrices. The only change from time step to time step is in the right-hand-side of Eq. (8).
4. Repeat the process in the next time step starting from the step 2

The above algorithm termed the Semi-Linear Implicit Method (SLIM) is implemented as a solver using version 5.04 of the Caelus [Applied CCM (2015)] library. The origin of the name can be traced to the fact that the coefficients of the Laplacian are always stored and only the right-hand-side changes.

Furthermore, additional efficiency can be gained if the momentum predictor equations are solved using explicit time stepping thus avoiding any need for creating and storing matrices for the momentum equations. With the stored hierarchy for the Laplacian and explicit time stepping for the momentum equations, the algorithm would completely avoid creation of matrices and computationally would perform close to the fully explicit algorithms that are pervasive in compressible flow computations.

Results and discussions

In the following section, several validation cases are presented for conditions ranging from attached to separated flow. Whilst the SLIM algorithm is by nature a time accurate solver, both steady state and transient cases have been considered under laminar and turbulent flows to fully investigate the performance of the algorithm and its implementation.

Laminar flat plate

In this case, steady, incompressible, laminar flow over a two-dimensional sharp-leading edge flat-plate at zero angle of incidence was investigated. The flow generates a laminar boundary layer and the computational results are compared with the Blasius solution for incompressible flow. Blasius, in his work [Blasius (1908)] obtained the solution to the Boundary Layer Equations using a transformation technique. Here, equations of continuity and momentum in two-dimensional form are converted into a single ordinary differential equation (ODE). The solution to this ODE can be numerically obtained and is regarded as the exact solution to the boundary layer equations. One of the highlights of Blasius solution is the analytical expression for the skin friction coefficient (c_f) distribution along the flat-plate given by

$$c_f \approx \frac{0.644}{\sqrt{Re_x}}, \quad (12)$$

where Re_x is the local Reynolds number defined as

$$Re_x = \frac{ux}{\nu}, \quad (13)$$

u is the freestream velocity, x is the distance starting from the leading edge and ν is the kinematic viscosity.

This case is based on the validation work carried out by the [NASA NPARC Alliance (2008)] for flow over a flat plate using the same conditions in the incompressible limit. A schematic of the geometric configuration is shown in Fig. 1. The length of the plate is $L = 0.3048 \text{ m}$ wherein, $x = 0$ is the leading edge, the Reynolds number of the flow based on the length of the plate is 200,000 and u is the velocity in the x -direction. Assuming the inlet flow is at a temperature of 300 K, the kinematic viscosity ($1.58963 \times 10^{-5} \text{ m}^2/\text{s}$) can be determined from dynamic viscosity and density of the fluid. The value of dynamic viscosity is obtained from the Sutherland viscosity formulation [Sutherland (1893)]. Using the Reynolds number, plate length and kinematic viscosity, the freestream velocity evaluates to $u = 10.4306 \text{ m/s}$. As we have assumed the flow incompressible, the density (ρ) remains constant. In addition, since the fluid temperature is not considered, the viscosity remains constant. For incompressible flows, the kinematic forms of pressure and viscosity are always used in Caelus 5.04.

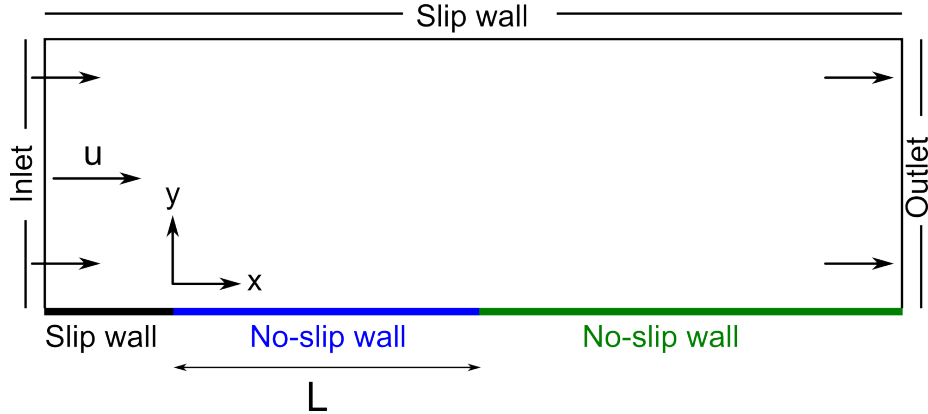


Figure 1: Flat plate computational domain

The computational domain is a rectangular block encompassing the flat-plate. Fig. 1 shows the details of the boundaries used in two-dimensions ($x - y$ plane). The region of interest extends between $0 \leq x \leq 0.3048 \text{ m}$ and has a no-slip boundary condition. Upstream of the leading edge, a slip boundary is used to simulate freestream uniform flow approaching the flat-plate. However, downstream of the plate, there is an additional no-slip wall a further three plate lengths. This ensures that the boundary layer in the vicinity of the trailing edge is not influenced by the outlet boundary. Since the flow is subsonic, disturbances cause the pressure to propagate both upstream and downstream. Therefore, placement of the inlet and outlet boundaries were chosen to have minimal effect on the solution. The inlet boundary is placed at start of the slip-wall ($x = -0.06 \text{ m}$) and the outlet at the end of the second no-slip wall ($x = 1.2192 \text{ m}$). Both inlet and outlet boundaries are between $0 \leq y \leq 0.15 \text{ m}$. A slip-wall condition is used for the entire top boundary. At the inlet a fixed uniform velocity $u = 10.4306 \text{ m/s}$ in x direction and zero pressure gradient is applied and at the outlet a fixed uniform pressure

$p = 0 \text{ m}^2/\text{s}^2$ and zero gradient velocity are applied. The simulation is initialised with a constant velocity $u = 10.4306 \text{ m/s}$ in the x direction and uniform zero pressure field.

A 3D hexahedral mesh was generated using [Pointwise (2014)] by extruding the 2D quadrilateral mesh in the $x - y$ plane. The two $x - y$ planes obtained as a result of grid extrusion need boundary conditions to be specified. As the flow over a flat-plate is generally 2D, we do not need to solve the flow in the third dimension. This is achieved in Caelus by specifying an empty boundary condition for each plane. Although, no flow is computed in the z direction, a velocity of $w = 0$ has to be specified for the velocity boundary condition as indicated above.

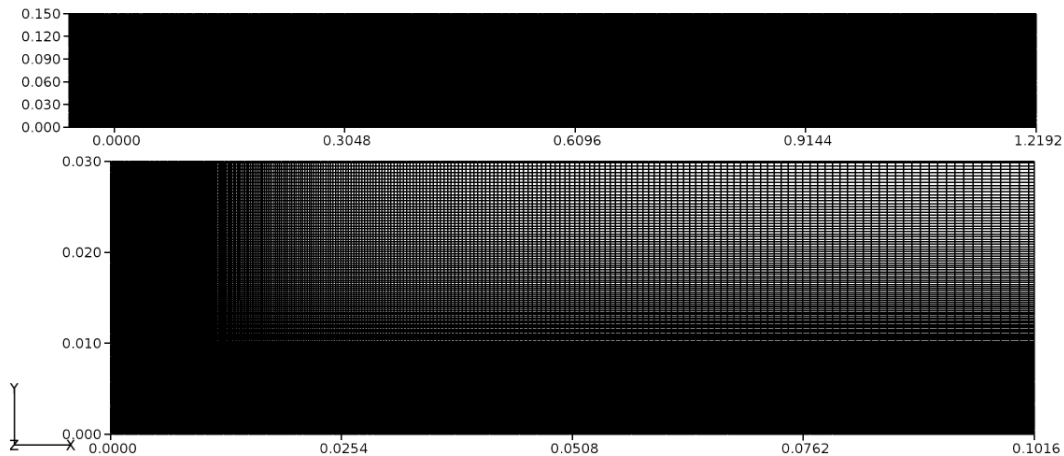


Figure 2: Computational mesh for the flat plate domain.

Fig. 2 shows the mesh in the $x - y$ plane. As can be seen, the grid is refined perpendicular to the wall in order to resolve the viscous effects. To ensure that the gradients in boundary layer are well resolved, about 50 grid nodes are placed between the wall and the boundary layer edge. Grid refinement is also added at the leading edge so that the growth of the boundary layer is also well resolved. In this particular case, 399 cells were used in the streamwise (x) direction ($x \leq 0 \leq 0.3048 \text{ m}$) and 297 in the wall normal (y) direction. For the no-slip wall beyond $x > 0.3048$, a similar distribution is used, resulting in a total cell count of 217,998.

A time-dependent solution was obtained using the SLIM solver. The flow was simulated sufficiently long (several plate length flow times) such that steady flow was established. For the discretization of time-dependent terms, the first-order Euler scheme was used. Pressure and velocity gradients were calculated using the Gauss method. A 2nd order linear upwind discretization was used for the divergence of velocity.

In Fig. 3, the skin-friction distribution along the flat-plate obtained from the SLIM solver is compared with that of the Blasius analytical solution. Here, the distance x is normalised with the length of the plate (L). Excellent agreement is observed along the entire length of the flat-plate. At the exit plane of the flat-plate at $x = 0.3048 \text{ m}$, velocity data was extracted across the boundary layer and compared with the Blasius analytical solution. This is shown in Fig. 4 where the velocity profile is plotted using similarity variables from the Blasius solution.

Here, η is the non-dimensional distance from the wall to the boundary layer edge and U_e is the velocity at the boundary layer edge. Similar to skin-friction, the velocity profile also exhibits excellent agreement with the Blasius solution.

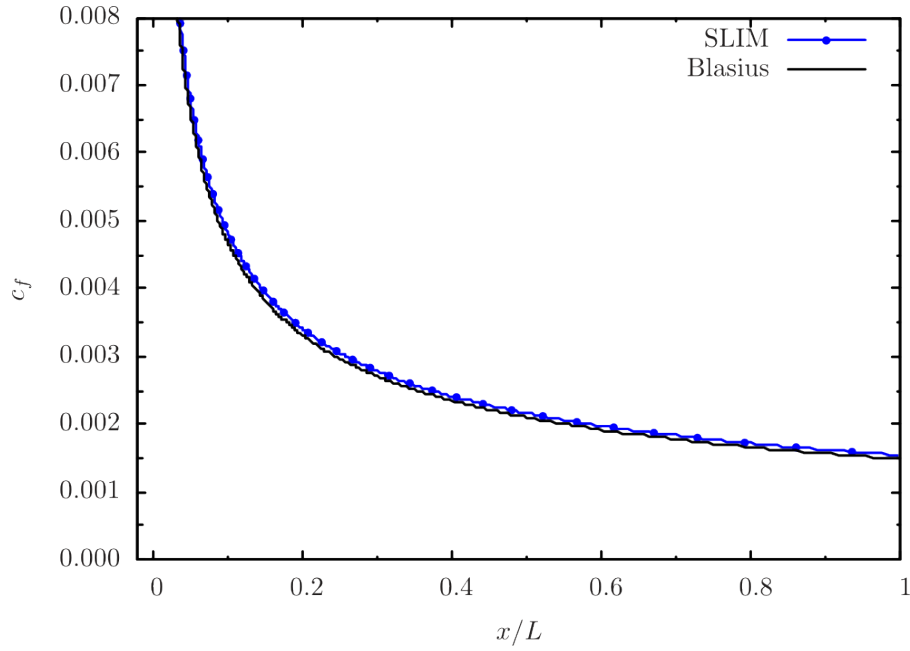


Figure 3: Skin-friction comparison between SLIM and Blasius solutions.

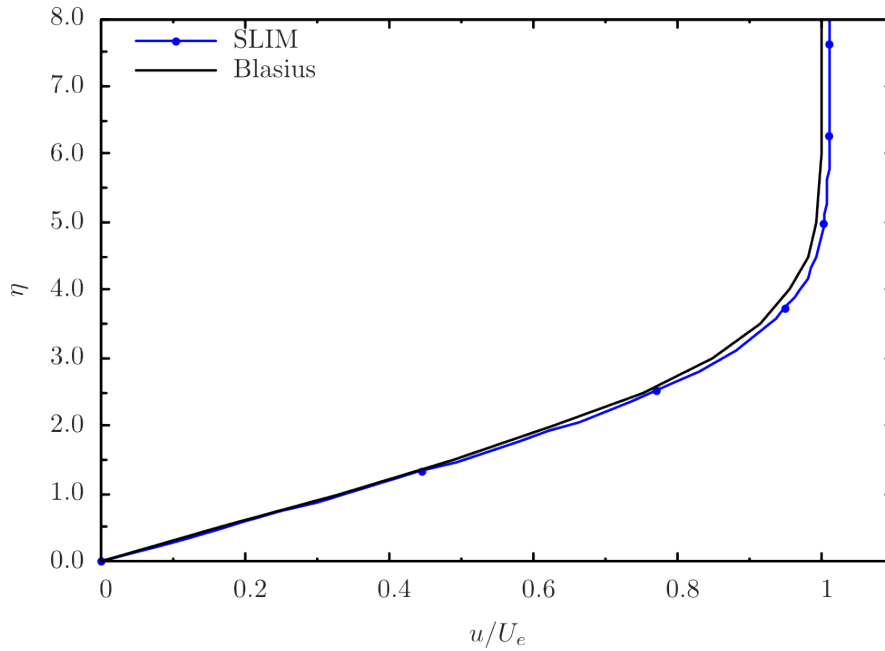


Figure 4: Non-dimensional velocity profile comparison between SLIM and Blasius solutions.

Tee junction

In this validation case, laminar, incompressible flow through a two-dimensional 90° tee junction was investigated. Due to the presence of the side branch, the flow separates and forms a recirculation region. The recirculating region influences the mass flow through the main and side branches. The numerically computed mass flow ratio was calculated and compared with experiment. A comprehensive study of flow through planar branches has been carried out by [Hayes et al. (1989)] due to its prevalence in the bio-mechanical industry.

Fig. 5 shows the schematic of the tee-junction. Here, $L = 3.0\text{ m}$ and $W = 1.0\text{ m}$ respectively, the Reynolds number based on the width is 300, and v is the velocity in the y-direction. For simplicity, we have assumed the velocity, $v = 1\text{ m/s}$. Using these values the resulting kinematic viscosity was $0.00333\text{ m}^2/\text{s}$. Since this is an internal flow problem, the computational domain is contained within tee-junction geometry. All tee-junction walls have a no-slip boundary condition. At the inlet, a fully developed laminar flow parabolic profile is applied with a mean velocity $v = 1.0\text{ m/s}$, otherwise a much longer main branch would be required for the flow to develop. The domain has two outlets, one at the end of the main channel and the other at the end of side branch. Exit pressures at the two outlets are equal ($p = 0\text{ m}^2/\text{s}^2$) and a zero gradient condition is applied to the velocity. The simulation is initialised with uniform zero velocity and pressure fields.

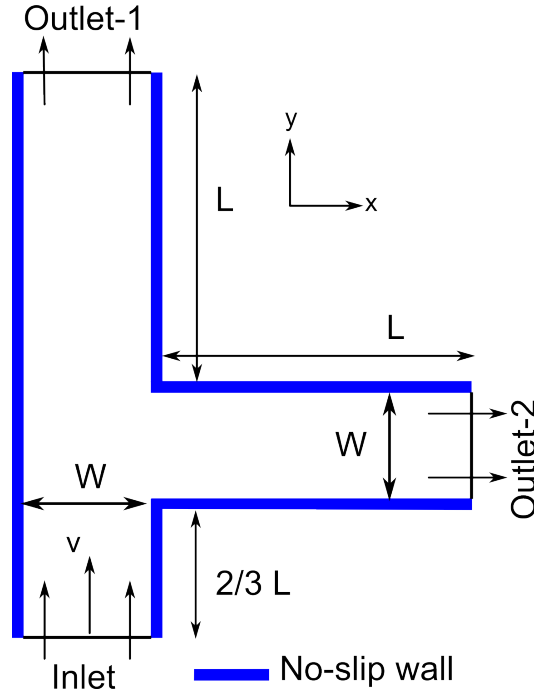


Figure 5: Computational domain representing tee-junction.

The computational mesh is shown in Fig. 6. The two $x - y$ planes obtained as a result of grid extrusion have empty boundary conditions applied to achieve 2D flow. A total of 2,025 hexahedral cells comprise the tee-junction of which, 90 cells are distributed along the height of the main channel, and 45 along the length of the side branch. The distribution is such that a dimensional length of $L = 1\text{ m}$

has a total of 45 cells, giving a distribution of 30 cells for the $(2/3)L$ segment of the main channel. The width, W , consists of 15 cells.

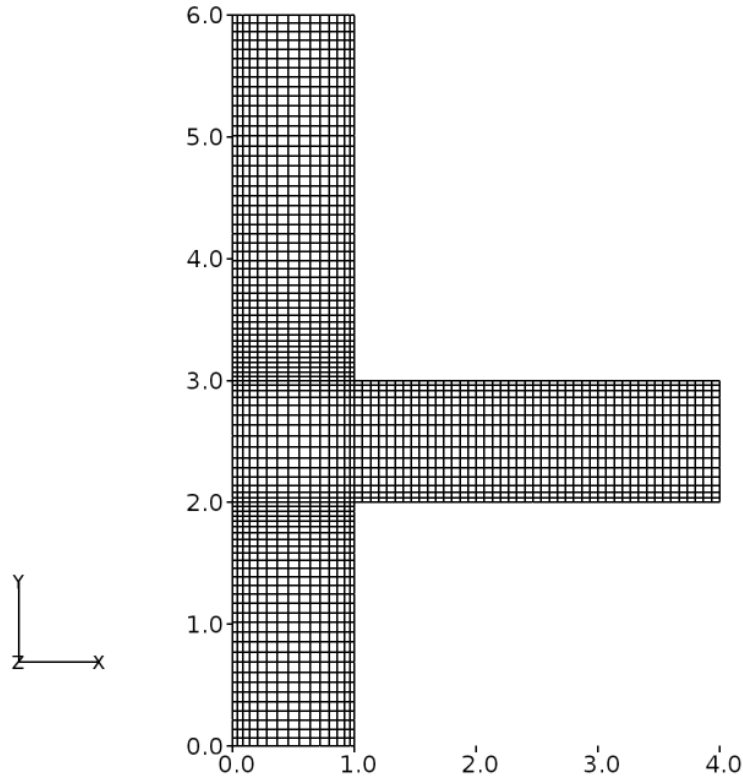


Figure 6: Structured grid for tee-junction domain.

A time-dependent solution was obtained using the SLIM solver. The flow was simulated sufficiently long such that steady separated flow was established. To ensure this, the shear-stress distribution was monitored on the lower wall of the side branch. For the discretization of time-dependent terms, the first-order Euler scheme was used. Pressure and velocity gradients were calculated using the Gauss method. A 2nd order linear upwind discretization was used for the divergence of velocity.

The mass flow rate was calculated at the inlet and at the main outlet (outlet-1) and the ratio was subsequently calculated. Table 1 compares the SLIM result with the experimental value. As can be noted, the agreement between the two is excellent.

Table 1: Comparison of mass flow rate split for SLIM and experiment.

	Experimental	SLIM	Percentage Difference
Flow Split	0.887	0.886	0.112 %

Triangular Cavity

This validation study concerns the laminar, incompressible flow inside a lid driven triangular cavity. Here, the top wall of the cavity moves at a constant velocity

initiating a recirculating motion within the cavity. Experiments on this configuration have been reported in [Jyotsna and Vanka (1995)] for a Reynolds number of 800. The main objective of this validation case was to compare the x velocity distribution against experimental data.

A schematic of the triangular cavity is presented in Fig. 7 where the depth of the cavity $D = 4\text{ m}$ and the width $W = 2\text{ m}$. The Reynolds number based on the cavity depth is 800 and the wall velocity is $u = 2\text{ m/s}$. Using the Reynolds number, u , and D , kinematic viscosity was calculated to be $0.01\text{ m}^2/\text{s}$. The side walls of the cavity have a no-slip boundary condition while the top wall, has a uniform velocity in the x direction. The simulation is initialised with zero velocity and pressure fields.

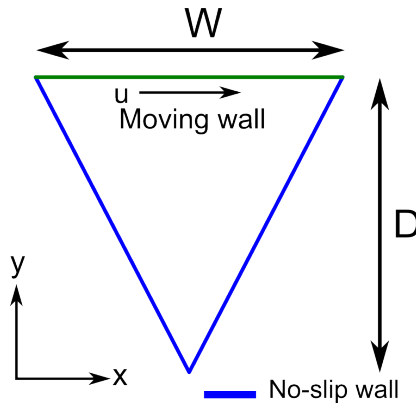


Figure 7: Computational domain of the triangular cavity.

The mesh in $x - y$ plane is shown in Fig. 8. A hybrid mesh is employed for this case with a total of 5,538 cells. Up to a depth of $D = 1.35\text{ m}$ hexahedral elements are used while below that value triangular prisms are used. The triangular prisms are used in the bottom portion because they resulted in better cell quality. For the hexahedral region, 39 cells are distributed across the width of the cavity and 39 along the depth. The cavity walls in the prism region have 100 cells along each. The interface of the two regions is node matched and has 39 cells across the width. The mesh close to the cavity lid was refined to better capture the shear layer. The flow characteristics in the cavity can be assumed to be 2D and here it has been solved with the same assumption.

A steady solution to the cavity was obtained using the SLIM solver. While a time-dependent approach was used, the solution was simulated sufficiently long so that steady flow was achieved. To determine when this occurred the velocity distribution along the cavity centre-line was monitored with respect to time. For the discretization of time-dependent terms, the first-order Euler scheme was used. Pressure and velocity gradients were calculated using the Gauss method. A 2nd order linear upwind discretization was used for the divergence of velocity. A linear corrected scheme was used for Laplacian discretization to account for mesh non-orthogonality.

In Fig. 9, the x velocity distribution along the cavity centre-line is compared with that of the experimental data [Jyotsna and Vanka (1995)]. The y distance is normalised with the cavity depth (D) which gives $y/d = 0$ at the cavity lid and $y/d = -1$ at the bottom vertex. Similarly, the u velocity is normalised with

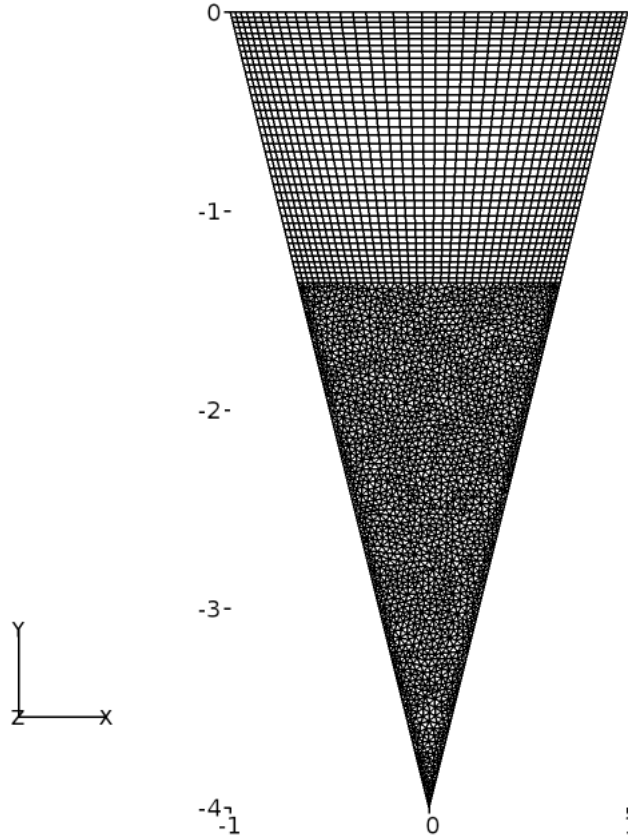


Figure 8: Hybrid grid for the triangular cavity.

the velocity of the cavity lid (u_L). As seen in Fig. 9 the comparison with the experiment is excellent.

2D Circular Cylinder

In this validation study, laminar incompressible flow over a 2D circular cylinder is investigated at a Reynolds number of 100. This classical configuration represents flow over a bluff body dominated by a wake region. For flows having a low Reynolds number ($40 \leq Re_D \leq 150$), periodic vortex shedding occurs in the wake. The phenomenon of vortex shedding behind bluff bodies is referred to as the Karman Vortex Street [Roshko (1954)] and provides a transient case for CFD code validation.

In his work, [Roshko (1954)] experimentally studied wake development behind 2D circular cylinders from Reynolds number ranging from 40 to 10000. For Reynolds numbers of 40 to 150, the so called the stable range [Roshko (1954)], regular vortex streets are formed with no evidence of turbulence motion in the wake. Therefore, at a Reynolds number of 100, the vortex shedding exhibits smooth, coherent structures making it ideally suited for validating laminar CFD calculations. The frequency associated with the oscillations of the vortex streets can be characterised by the Strouhal Number (St). The Strouhal Number is a non-dimensional number defined as

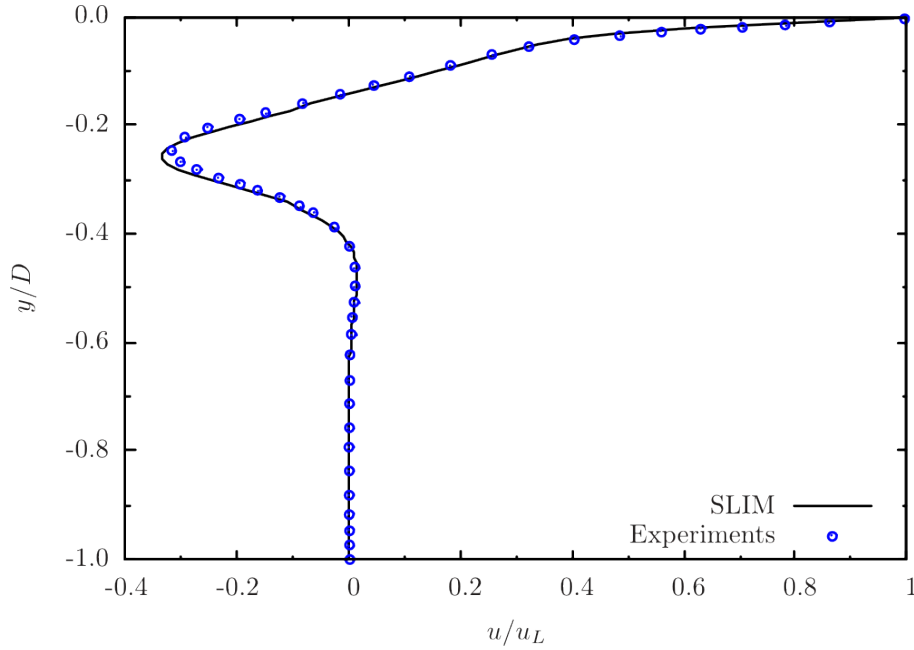


Figure 9: Comparison of experimental and computational x velocity distribution along the cavity's centre-line.

$$St = \frac{fD}{u} \quad (14)$$

where, f is the frequency of oscillations of vortex shedding, D is the diameter of the cylinder and u is the freestream velocity of the flow. Experimentally [Roshko (1954)], it has been determined that for a Reynolds number based on the diameter of the cylinder of 100, the Strouhal number $St \approx 0.16 - 0.17$. The main objective of this study was to compare the St for the SLIM calculation to the experimental data of [Roshko (1954)]. Provided the cylinder has a sufficient span length, the flow characteristics can be assumed to be 2D as the experiments suggest.

Fig. 10 shows the schematic of the 2D circular cylinder. Here, the diameter $D = 2\text{ m}$ and is the characteristic length for the Reynolds number. For simplicity, the freestream velocity was taken to be $u = 1\text{ m/s}$ in the x -direction. Using these values the kinematic viscosity was calculated to be $0.02\text{ m}^2/\text{s}$. The domain extends by 5 diameters upstream and 20 diameters downstream. In the y direction, the domain extends 5 diameters on either side. From the figure, multiple inlet boundaries to this domain can be seen, one at the upstream boundary and the other two for the top and bottom boundaries. This type of configuration is needed to appropriately model the inflow, similar to an undisturbed flow in an experimental set-up. It is noted that for top and bottom boundaries, the flow is in the x direction. The outlet is located at the downstream boundary. The cylindrical wall is a no-slip boundary condition. The solution is initialised with a uniform velocity $u = 1.0\text{ m/s}$ and uniform zero pressure field.

The computational mesh was generated using Pointwise in the $x - y$ plane and subsequently extruded one cell in the z direction. The mesh consisted of 9,260 cells. An O-grid topology was constructed around the cylinder with 10 cells in the radial direction and 84 cells in the circumferential direction. 31 cells were used

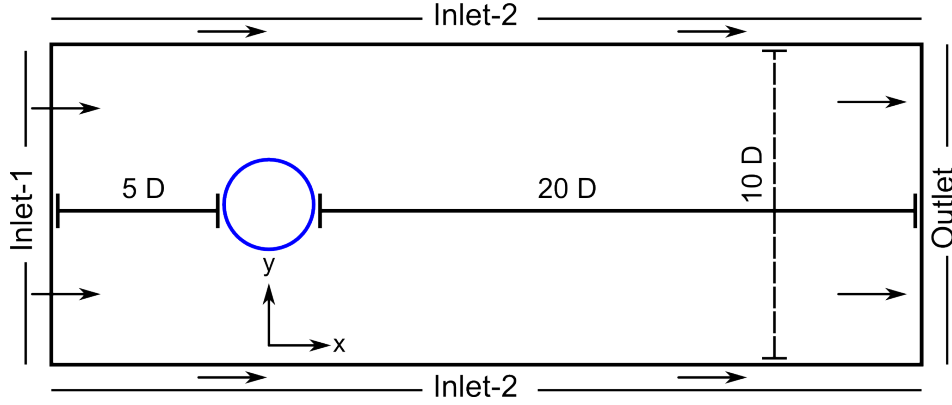


Figure 10: Computational domain of a circular cylinder.

upstream of the O-grid, in the x direction while 100 cells were used downstream. The region of interest is about 10 diameters downstream, where the grids are refined. In the y direction, 21 cells were used above and below the O-grid region.

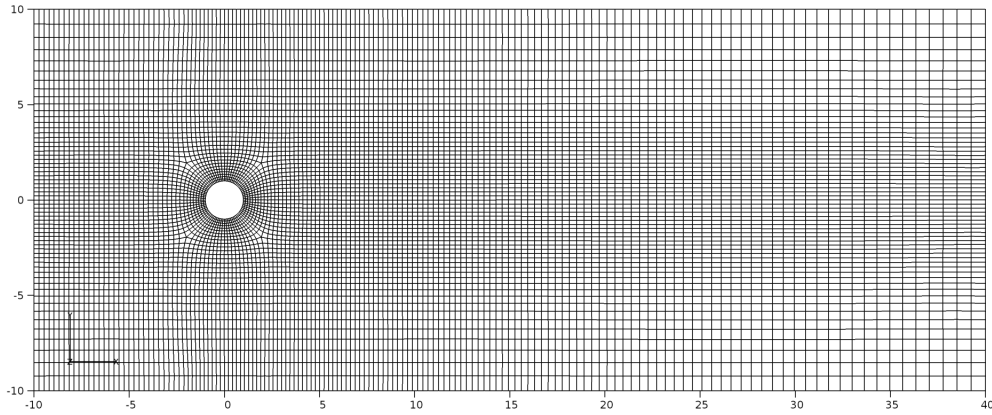


Figure 11: O-grid around the circular cylinder.

A time-dependent simulation was carried out using the SLIM solver. For the discretization of time-dependent terms, the 2nd order backward scheme was used. Pressure and velocity gradients were calculated using the Gauss method. A 2nd order linear upwind discretization was used for the divergence of velocity. A linear corrected scheme was used for Laplacian discretization to account for mesh non-orthogonality.

To capture the transient start-up process, the simulation was started from time $t = 0$ s and was simulated up to $t = 360$ s, while lift and drag forces over the cylindrical surface were monitored at a frequency of 2 Hz. It was found that the on-set of vortex shedding occurred after about $t = 90$ s which was then followed by a steady shedding process. A Fast Fourier transformation (FFT) was carried out on the lift force data and the peak frequency of vortex shedding occurred at $f = 0.0888$ Hz. Based on this value, it takes about 7.8 cycles for the shedding to start. Table 2 compares the computed value from SLIM with that of the experiment. The agreement is good given that experimental uncertainty can be

relatively high at low Reynolds numbers.

Table 2: Comparison of experimental and numerical frequency results for the circular cylinder.

	Frequency (Hz)	Strouhal Number
Experimental	0.0835	0.167
SLIM	0.0888	0.177

Square cylinder

This case considers the turbulent, incompressible flow around a square cylinder, as studied experimentally by [Lyn and Rodi (1994) and Lyn et al. (1995)]. The side of the square cylinder (D) is 1 m and it extends along the width ($4D$) of the channel. All distances are made non-dimensional with reference to D . The mean velocity at the inlet, u , is assumed to be 0.214 m/s . All velocities are made non-dimensional using the inlet velocity. The Reynolds number, based on u and D is 21,400.

A rectangular computational domain in the $x - y$ plane was constructed surrounding the square cylinder as shown in Fig. 12. The domain extends in the z direction a length of $4D$. The domain extends by 5 diameters upstream and 14.5 diameters downstream. In the y direction, the domain extends 7 diameters on either side. At the inlet a uniform velocity $u = 0.214 m/s$ in x direction and a zero pressure gradient are applied. At the outlet a uniform zero pressure and zero velocity gradient conditions are applied. The top and bottom boundaries have a slip condition applied to velocity and a zero gradient condition applied to pressure. Cyclic boundaries conditions are used in the span wise direction and a no-slip boundary on the cylinder. The simulation is initialised with a uniform velocity $u = 0.214 m/s$ in x direction and zero pressure field.

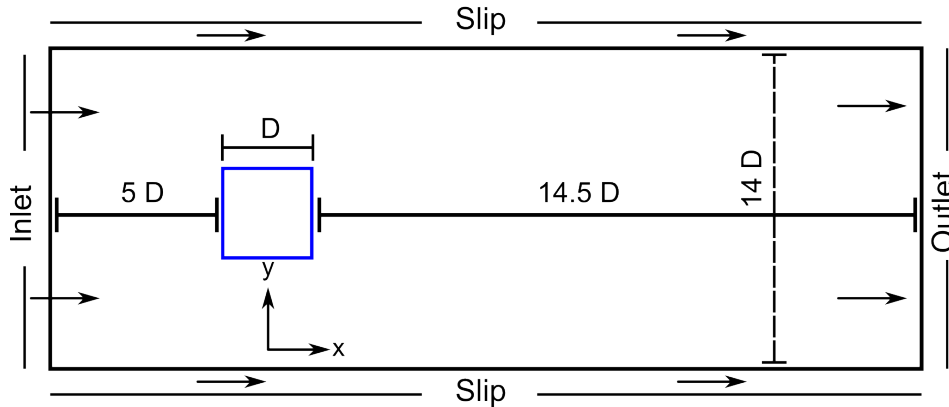


Figure 12: Computational domain of the square cylinder.

The non-uniform computational mesh shown in Fig. 13 was generated using Pointwise and consisted of $216 \times 150 \times 21$ (668,850) cells (being denser near the cylinder). For the discretization of time-dependent terms, the 2nd order backward

scheme was used. Pressure and velocity gradients were calculated using the Gauss method. A 2nd order linear upwind discretization with multidimensional interpolation linear scheme utilising Barth-Jespersen limiter [Berger et al. (2005)] was used for the divergence of velocity.

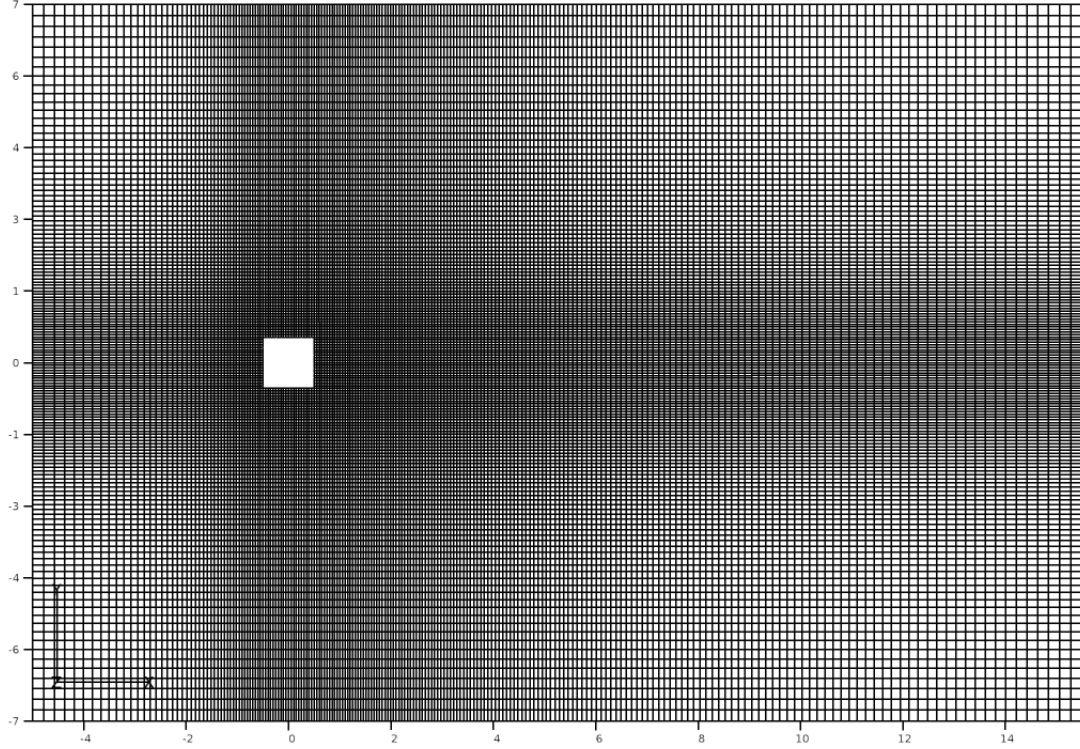


Figure 13: Square cylinder mesh.

Large Eddy Simulation (LES) computes the large-scale motions of the flow directly. The small-scale, dissipative motions of turbulence tend to be more amenable to modelling because of their more uniform character, whereas the large-scale motions contain the majority of the energy and anisotropy. As a result, LES is expected to be more accurate, particularly in complex flows where the assumptions inherent to RANS models rarely exist. The drawback is that LES simulations are always three-dimensional and unsteady. For this particular case, the Smagorinsky SGS model [Smagorinsky (1963)] was chosen. The model is based on the SGS kinetic energy $k = \frac{1}{2} (\overline{\mathbf{u}^2} - \tilde{\mathbf{u}}^2)$ where $\tilde{\mathbf{u}}$ is the filtered velocity. The following assumptions for the SGS stress tensor ($\mathbf{B}$) and the filtered deviatoric part of the rate of strain tensor ($\widetilde{\mathbf{D}}_D$) in incompressible flows are used

$$\mathbf{B} = \left[\frac{2}{3} k \mathbf{I} - 2\nu \widetilde{\mathbf{D}}_D \right], \quad (15)$$

$$\widetilde{\mathbf{D}}_D = \left[\widetilde{\mathbf{D}} - \frac{1}{3} (tr \widetilde{\mathbf{D}}) \mathbf{I} \right], \quad (16)$$

$$\widetilde{\mathbf{D}} = \frac{1}{2} [\nabla \tilde{\mathbf{u}} + \nabla \tilde{\mathbf{u}}^T], \quad (17)$$

where $\mathbf{I}$ is the unit tensor. The Smagorinsky model is an algebraic model for the SGS viscosity ν_{SGS} . Caelus 5.04 implements the SGS viscosity as

$$\nu_{SGS} = c_k \sqrt{(k_{SGS})} \Delta \quad (18)$$

where the k_{SGS} is given by

$$k_{SGS} = \frac{(c_k \Delta^2)}{c_\epsilon} \|\widetilde{\mathbf{D}}\|^2. \quad (19)$$

and Δ represents the top-hat filter with a characteristic filter width estimated as the cubic root of the cell volume. The relationship between the classical C_s constant and the constants c_k and c_ϵ from the Smagorinsky model implementation in Caelus 5.04 is

$$C_s = \left(\frac{c_k^3}{c_\epsilon} \right)^{0.25}. \quad (20)$$

In this work the value used for C_s was 0.1. In some circumstances the turbulent viscosity near the wall is over predicted by the LES SGS model due to a lack of constraint on the turbulent viscosity. To remedy this, damping is added to the length scale (Δ) using the [Van Driest (1956)] formulation. In Caelus 5.04 the implementation takes the form

$$\Delta = \min \left[\Delta_{cr}, \frac{\kappa y}{C_\Delta} \left(1 - e^{-\frac{y}{y^* A^+}} \right) \right], \quad (21)$$

where Δ_{cr} is the filter width calculated from the cubic root of the cell volume, y is the distance from the wall, C_Δ , A^+ , κ are constants and y^* is given by the formula $y^* = \nu/u_\tau$. Here u_τ is the wall friction velocity.

The simulation was started from time $t = 0$ s and was simulated up to $t = 1000$ s using a fixed Courant number of 1, while lift and drag forces over the cylindrical surface were monitored. It was found that the on-set of vortex shedding occurred after about $t = 100$ s which was then followed by a steady shedding process. The velocity field was spanwise and time averaged for the period between $200 \leq t \leq 1000$ s. A Fast Fourier transformation (FFT) was carried out on the lift force data and the peak frequency of vortex shedding occurred at $f = 0.028$ Hz. Based on this value, it takes about 2.7 cycles for the shedding to start. The most important time-averaged parameters are presented in Table 3, where they are compared to experimental and numerical data, available from several authors from the ERCOFTAC database. The labels used are the same as in [Voke (1997)]. The parameters compared are the recirculation length, Strouhal number, the drag and lift coefficients and the R.M.S variation of the drag and lift coefficients.

The values predicted by SLIM agree reasonably well with the corresponding experimental and numerical data. The drag and lift coefficients are slightly greater than experimental data but agree as well as other numerical data. Fig. 14 shows the normalised time-averaged horizontal velocity along the centreline 10 diameters downstream of the cylinder. The experiment shows the velocity reaching approximately 0.6 of the freestream value and then leveling off thereafter. It can be observed from this figure that there is a fair amount of disparity among the numerical results. Some of the LES simulations, in particular UK1, UK3 and

Table 3: Comparison among time-averaged squared cylinder data. The labels are the same as used in [Voke (1997)].

Set	l_r	St	C_D	C_L	$\overline{C_D}$	$\overline{C_L}$
IS3	1.24	0.133	2.79	-0.125	0.36	1.68
NT7	1.39	0.131	2.05	-0.050	0.12	1.39
ST3	1.24	0.150	2.66	-0.005	0.27	1.33
TIT	1.23	0.131	2.62	0.0093	0.23	1.39
UK1	1.32	0.130	2.20	-0.020	0.14	1.01
UK3	1.44	0.130	2.23	-0.050	0.13	1.02
UOI	1.20	0.130	2.03	0.0400	0.18	1.29
[Lyn et al. (1995)]	1.38	0.132	2.1			
SLIM	1.41	0.131	2.44	0.076	0.33	1.29

NT7, show the velocity approaching the free stream value; others such as ST5 and UOI show a distinct decline beyond $x/D = 5$. The results from this work agree reasonably well with the experimental data.

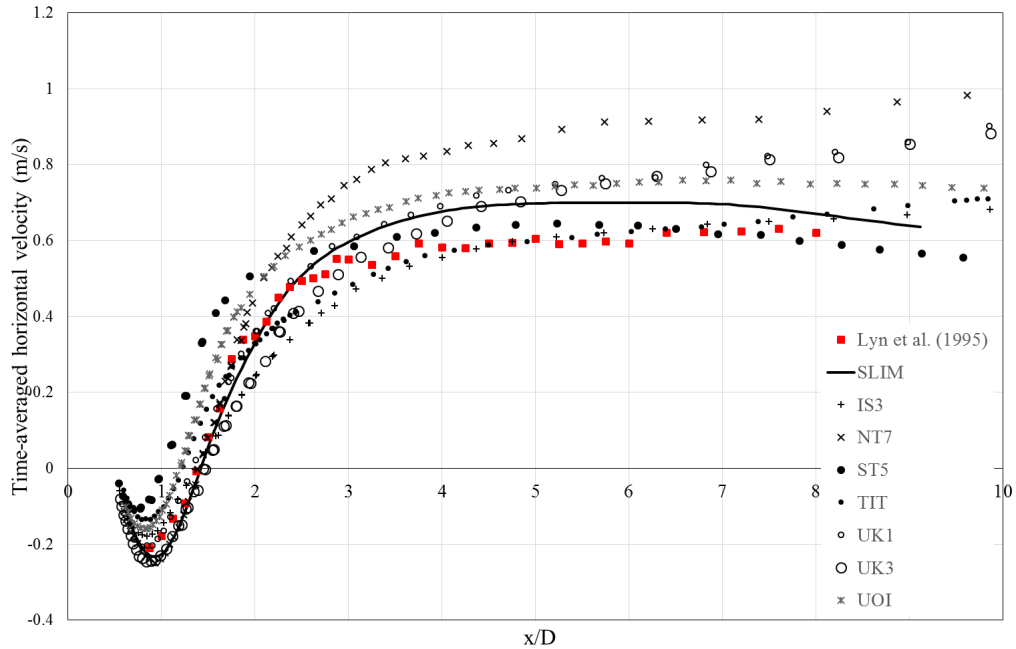


Figure 14: Streamwise distribution of the normalised time-averaged horizontal velocity along the centreline $y = 0$ for the experimental and numerical data sets.

The prediction of fluctuating horizontal and vertical velocities along the centre

line are shown in Figs. 15 and 16 respectively. These figures also display disparity amongst the numerical simulations and reveal that no simulation matches closely. The result from this work shows a slightly higher peak in the horizontal velocity fluctuation just behind the cylinder but does a much better job than most of the other simulations at matching the experimental data beyond this point. The fluctuations in the vertical velocity show reasonable agreement with the experimental data over the entire measurement region.

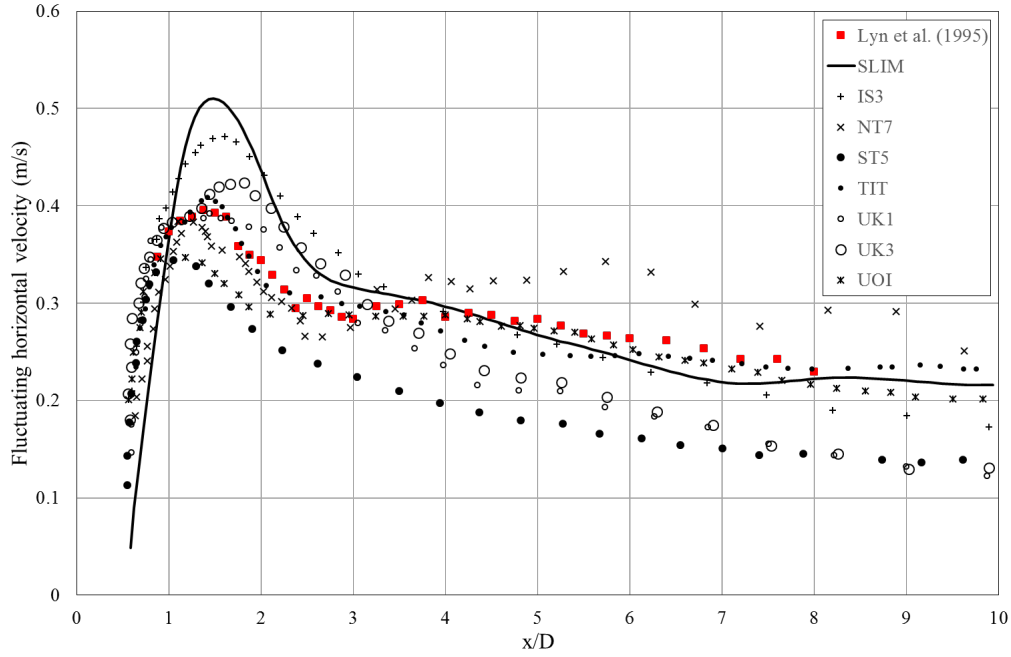


Figure 15: Streamwise distribution of the normalised horizontal velocity fluctuation along the centreline $y=0$ for the experimental and numerical data sets.

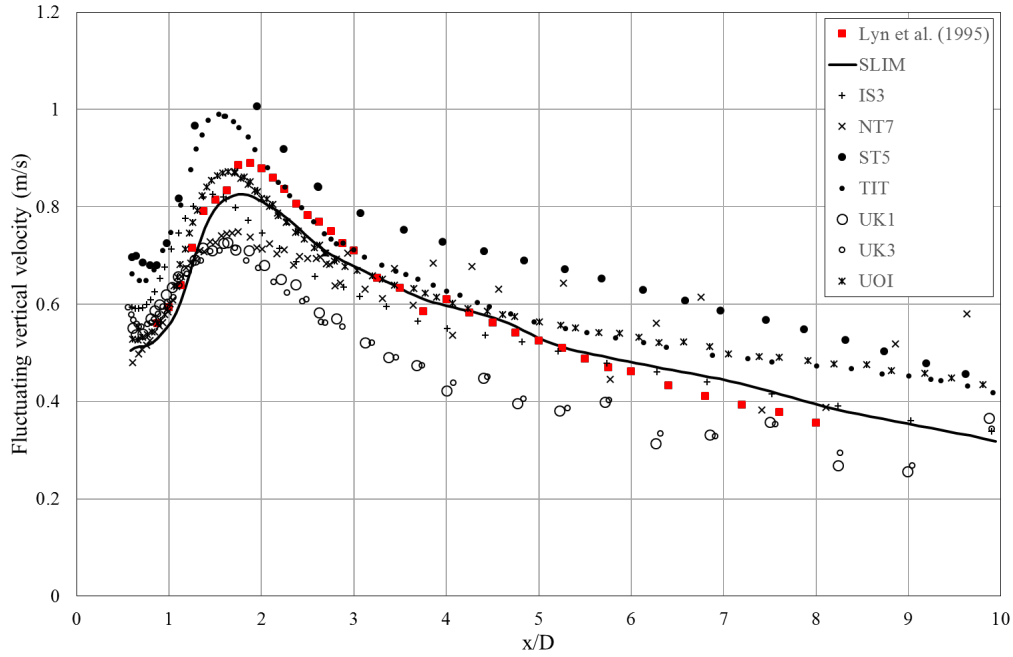


Figure 16: Streamwise distribution of the normalised vertical velocity fluctuation along the centreline $y = 0$ for the experimental and numerical data sets.

Conclusions

In this paper, the Semi-Linear Implicit Method (SLIM) algorithm was described. The algorithm has many benefits including the exact satisfaction of the continuity equation for each time step. In addition, the algorithm is computationally efficient due to the geometric nature of the pressure Laplacian and the hyperbolic nature of the momentum equations. The algorithm was implemented as a solver using version 5.04 of the Caelus library and its accuracy tested through several validation cases comprising steady and transient laminar problems along with a transient turbulent case.

References

- Applied CCM (2015). Caelus: Computational Mechanics Library (Version 5.04) [Software]. Available at <http://www.caelus-cml.com>.
- Berger, M., Aftosmis, M.J., and Murman, S.M. (2005). Analysis of Slope Limiter on Irregular Grids. *43rd AIAA Aerospace Sciences Meeting*, Jan. 10-13, 2005, Reno, NV
- Blasius H. (1908). Grenzschichten in Flüssigkeiten mit kleiner Reibung. *Z. Math. Phys.*, **56**, 1–37.
- Chorin A.J. (1967). A numerical method for solving incompressible viscous flow problems. *J. Comp. Phys.*, **2**, 12–26.
- Hayes, R.E., Nandkumar, K. and Nasr-El-Din, H. (1989). Steady Laminar Flow in a 90 Degree Planar Branch. *Computers and Fluids*, **17**(4), 537–553.
- Issa, R. (1985). Solution of the implicitly discretised fluid flow equations by operator splitting. *J. Comp. Phys.*, **61**, 40.

- Jyotsna, R. and Vanka, P. (1995). Multigrid Calculation of Steady, Viscous Flow in a Triangular Cavity. *J. Comp. Phys.*, **122**, 107–117.
- Lyn, D.A. and Rodi, W. (1994). The flapping shear layer formed by flow separation from the forward corner of a square cylinder. *J. Fluid Mech.*, **267**, 353.
- Lyn, D.A., Einav, S., Rodi, W. and Park, J.H. (1995). A laser-Doppler velocimetry study of ensemble averaged characteristics of the turbulent near wake of a square cylinder. *J. Fluid Mech.*, **304**, 285.
- NPARC Alliance. (2008). NPARC Alliance Validation Archive. [ONLINE] Available at <http://www.grc.nasa.gov/WWW/wind/valid/fplam/fplam.htm>. [Accessed 13 May 15].
- Patankar, S. V., and Splading, D.B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. of Heat and Mass Transfer*, **15(10)**: 1787–1806.
- Pointwise Inc. (2014). Pointwise, Mesh Generation Software for CFD, Version 17.2R2. Fort Worth, TX. <http://www.pointwise.com/>.
- Roshko, A. (1954). On the Development of Turbulent Wakes from Vortex Streets. Technical Report TR-1191, National Advisory Committee for Aeronautics.
- Smagorinsky, J. (1963). General Circulation Experiments with Primitive Equations, *Month. Weath. Rev.*, **93**, 99–165.
- Strikwerda, J.C., and Lee, Y.S. (1999). The accuracy of the fractional Step method. *SIAM J. Numer. Anal.*, Vol. **37**, No. **1**, 37–47.
- Sutherland, W. (1893). The viscosity of gases and molecular force. *Philosophical Magazine*, **S. 5**, 507–531.
- Van Driest, E.R. (1956). On the turbulent flow near a wall, *J. Aeronautical Sci.*, **23**, 1007–1011.
- Voke, P. (1997) *Direct and Large-Eddy Simulation II*, chapter Flow past a Square Cylinder: Test Case LES2. Kluwer Academic.