Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) – solution in MATLAB®

H Khawaja*, M Moatamedi
UiT - The Arctic University of Norway

ABSTRACT

This work presents a method for the solution of fundamental governing equations of computational fluid dynamics (CFD) using the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) in MATLAB®. The fundamental governing equations of fluid mechanics are based on three laws of conservation, referred to as the law of conservation of mass, the law of conservation of momentum and the law of conservation of energy. The continuity equation represents the law of conservation of mass, the Navier-Stokes equations represent the law of conservation of momentum, and the energy equation represents the law of conservation of energy. In SIMPLE, the continuity and Navier-Stokes equations are required to be discretized and solved in a semi-implicit way. This article presents the discretization and method of solution applied to the flow around a 2-D square body. Code is written in MATLAB®. The results show the pressure and velocity fields of the converged solution.

1. INTRODUCTION

Computational fluid dynamics (CFD) is based on three basic physical principles: conservation of mass, of momentum and of energy. The governing equations in CFD are based on these conservation principles. The continuity equation is based on conservation of mass, as shown in Equation (1),

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \tag{1}$$

where ρ is fluid density, u is fluid velocity in the x-direction, v is fluid velocity in the y-direction, w is fluid velocity in the z-direction and t is time.

The Navier-Stokes equations based on conservation of momentum are shown in Equation (2) for the x-direction,

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot \left(\rho u \vec{U}\right) = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \tag{2}$$

in Equation (3) for the y-direction,

$$\frac{\partial(\rho v)}{\partial t} + \nabla \cdot \left(\rho v \vec{U}\right) = -\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y \tag{3}$$

*Corresponding Author: hassan.a.khawaja@uit.no and in Equation (4) for the z-direction,

$$\frac{\partial(\rho w)}{\partial t}+\nabla\cdot\left(\rho w\vec{U}\right)=-\frac{\partial p}{\partial z}+\frac{\partial\tau_{xz}}{\partial x}+\frac{\partial\tau_{yz}}{\partial y}+\frac{\partial\tau_{zz}}{\partial z}+\rho f_z \tag{4}$$

where p represents pressure, $\vec{U}$ the velocity vector $(ui+vj+wk)$, $\vec{f}=f_x i+f_y j+f_z k$ the body force vector and τ is the shear stress tensor as shown in Equation (5),

$$\tau=\begin{bmatrix}\tau_{xx} & \tau_{xy} & \tau_{xz}\\ \tau_{yx} & \tau_{yy} & \tau_{yz}\\ \tau_{zx} & \tau_{zy} & \tau_{zz}\end{bmatrix} \tag{5}$$

The effect of volumetric viscosity is ignored. If the fluid is considered to be a Newtonian fluid, then shear stress is as shown in Equation (6),

$$\tau_{ij}=\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\delta_{ij}\left(\frac{2}{3}\right)\frac{\partial u_k}{\partial x_k}\right) \tag{6}$$

where μ is the dynamic viscosity, δ_{ij} is the kronecker delta, and subscripts i, j and k refer to linear dimensions x, y and z.

The equation based on the conservation of total energy is shown in Equation (7),

$$\frac{\partial}{\partial t}\left[\rho\left(e+\frac{U^2}{2}\right)\right]+\nabla\cdot\left[\rho\left(e+\frac{U^2}{2}\right)\vec{U}\right]=\rho\dot{q}+\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)-\frac{\partial(up)}{\partial x}-\frac{\partial(vp)}{\partial y}-\frac{\partial(wp)}{\partial z}+\frac{\partial(u\tau_{xx})}{\partial x}+\frac{\partial(u\tau_{yx})}{\partial y}+\frac{\partial(u\tau_{zx})}{\partial z}+\frac{\partial(v\tau_{xy})}{\partial x}+\frac{\partial(v\tau_{yy})}{\partial y}+\frac{\partial(v\tau_{zy})}{\partial z}+\frac{\partial(w\tau_{xz})}{\partial x}+\frac{\partial(w\tau_{yz})}{\partial y}+\frac{\partial(w\tau_{zz})}{\partial z}+\rho\vec{f}\cdot\vec{U} \tag{7}$$

where $e=c_vT$ is internal energy, $U^2=u^2+v^2+w^2$ is the magnitude of velocity vector, $\dot{q}$ is the energy source, k is the thermal diffusion coefficient and T is temperature.

By subtracting the product of momentum equations with their corresponding velocity from the total energy equation, an equation for the conservation of internal energy can be written as shown in Equation (8),

$$\frac{\partial(\rho e)}{\partial t}+\nabla\cdot(\rho e\vec{U})=\rho\dot{q}+\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)-p\left(\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}-\frac{\partial w}{\partial z}\right)+\lambda\left(\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}-\frac{\partial w}{\partial z}\right)^2+\mu\left[2\left(\frac{\partial u}{\partial x}\right)^2+2\left(\frac{\partial v}{\partial y}\right)^2+2\left(\frac{\partial w}{\partial z}\right)^2+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)^2+\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)^2\right] \tag{8}$$

where $\lambda=-\frac{2}{3}\mu$ and μ is the coefficient of dynamic viscosity.

These governing equations can be applied to a physical system in a closed volume. Equations (1) to (8) are non-linear partial differential equations and difficult to solve analytically. The numerical method of solution involves the discretization in time and space (closed volume) domains. The solution is achieved using iterative numerical techniques such as the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE).

A SIMPLE algorithm is a widely used numerical procedure in computational fluid dynamics (CFD) to solve the fundamental governing equations of fluid mechanics. Prof. Brian Spalding and Suhas Patankar developed the SIMPLE algorithm at Imperial College, London, in the early 1970s [1]. Since then, it has been extensively used by many researchers to solve different kinds of fluid flow and heat transfer problems [2-4]. The literature on computational fluid dynamics discusses the SIMPLE algorithm in detail [5]. A modified variant is the SIMPLER algorithm (SIMPLE Revised) introduced by Patankar in 1979 [6].

In order to be solved using numerical techniques, CFD problems need to be discretized in space dimensions [2, 7-15]. For such numerical problems to be solved, they need to be discretized in nodes and elements [16-18]. The nodes represent the position in space where parameters are being calculated (for example, pressure, velocities), and the elements define the equations relating to the parameters (for example, continuity and Navier-Stokes) [19]. The geometry of the element defines the extent of calculations regarding space dimensions. These space dimensions categorize one-dimensional (1-D), two-dimensional (2-D) or three-dimensional (3-D) elements, as shown in Fig. 1 [20].

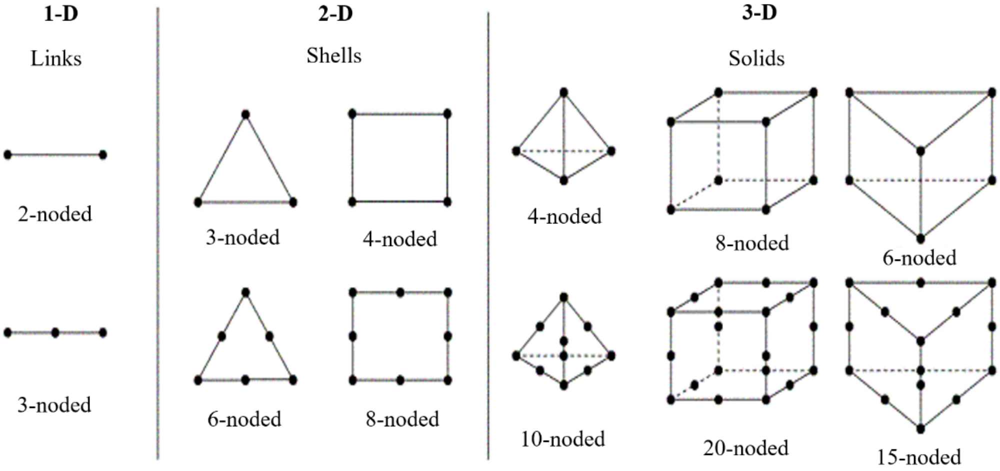


Figure 1. One-dimensional (1-D), two-dimensional (2-D) or three-dimensional (3-D) elements [20].

This physical behavior of various real-time structures and physical phenomena allows engineers and mathematicians to assume a realistic 3-D situation to be a 2-D or even a 1-D problem. The advantage of reducing the dimensions is that it allows complex equations to be solved much more easily and relatively faster [7].

This paper provides a step by step methodology for SIMPLE, defining the problem, giving the assumptions, degrees of freedom, discretization, and method of solution. The methodology also discusses the flow chart of the method written in MATLAB®. The results section of the paper discusses the pressure and velocity fields obtained from the solution. The paper concludes by discussing the effectiveness and limitations of the given method.

2. METHODOLOGY

The methodology is divided into two sections: setting up the SIMPLE and the method of solution in MATLAB®. Setting up the SIMPLE includes: assumptions, degrees of freedom, discretization, and pressure and velocity corrections. The method of solution is given in the form of a flowchart as shown in Fig. 4.

This work assumes the CFD problem to be incompressible. This means that the density does not change in time and space and, henceforth, can be considered as a constant value. This assumption is very valid in a range of problems when flow velocities are not too high. In addition, the given CFD problem does not constitute any thermal effects. This assumption allows us to simplify the problem to the extent that there is no need to solve the energy equation.

The domain is defined as 2-D space, as shown in Fig. 2, presenting a 2-D problem of flow around a square. The variables constituting continuity and Navier-Stokes are pressure, x-velocity and y-velocity.

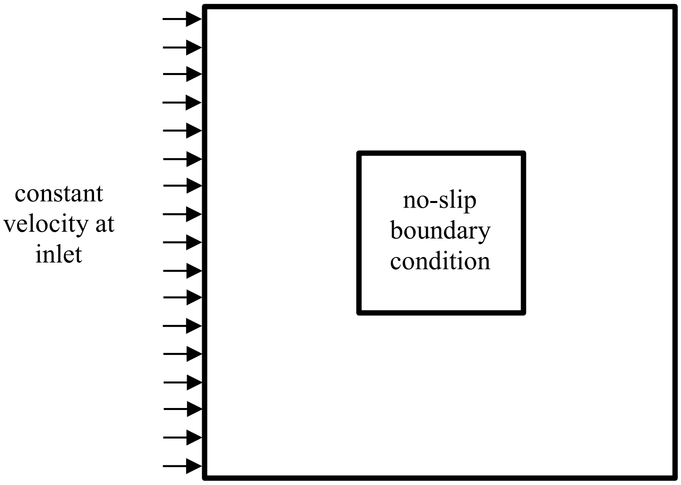


Figure 2. Domain of a 2-D problem of flow around a square.

With the above-given assumptions, continuity and Navier-Stokes equations can be simplified as shown in Equations (9) to (11),

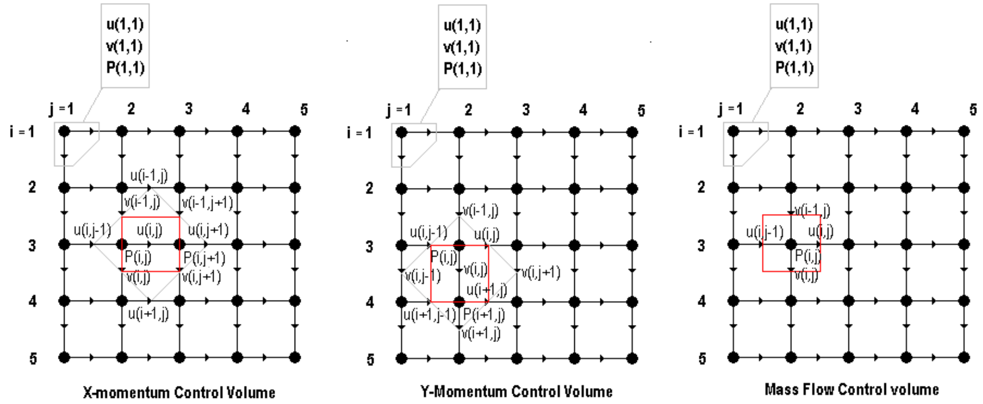
$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y} = 0 \tag{9}$$

$$\frac{\partial u}{\partial t} = -\frac{u\partial u}{\partial x}-\frac{u\partial v}{\partial y}-\frac{\partial p}{\partial x}+\frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{10}$$

$$\frac{\partial v}{\partial t} = -\frac{v\partial v}{\partial y}-\frac{v\partial u}{\partial x}-\frac{\partial p}{\partial y}+\frac{\mu}{\rho}\left(\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial x^2}\right) \tag{11}$$

where u is the fluid velocity in the x-direction, v is the fluid velocity in the y-direction, p is the fluid pressure, ρ is the fluid density, μ is the coefficient of dynamic viscosity of the fluid, and t is time.

In order to apply a numerical scheme such as SIMPLE for the solution of Equations (9) to (11), each term needs to be discretized. Discretization is the mathematical process of transforming continuous equations into their discrete counterparts. This process is a first step toward making equations suitable for a numerical solution and for implementation on digital machines. In this work, the 2-D domain has been discretized in a staggered grid, as shown in Fig. 3.



X-momentum Control Volume

Y-Momentum Control Volume

Mass Flow Control volume

Figure 3. 2-D domain discretized in a staggered grid.

The transient terms from Navier-Stokes equations (Equations (10) and (11)) can be discretized using the forward difference method as shown in Equations (12) and (13),

$$\frac{\partial u}{\partial t} = \frac{u^{t+1}(i,j) - u^{t}(i,j)}{\Delta t} \tag{12}$$

$$\frac{\partial v}{\partial t} = \frac{v^{t+1}(i,j) - v^{t}(i,j)}{\Delta t} \tag{13}$$

where Δt is the value of the timestep.

Different methods can be used for the discretization of the convective term. The decision as to which method to select is based on the Reynolds number. This non-dimensional number is the ratio of convection and diffusion. If this number is small enough, we can use the central difference method; however, if the Reynolds number is higher, it is better to use the upwind scheme. Convective terms discretized using the central difference method are shown in Equations (14) to (17),

$$-\frac{u\partial u}{\partial x} = -\frac{u^{t}(i,j-1)^{2} - u^{t}(i,j+1)^{2}}{2\Delta x} \tag{14}$$

$$-\frac{u\partial v}{\partial y} = -\frac{u^{t}(i-1,j)\left(v^{t}(i-1,j)+v(i-1,j+1)\right)}{4\Delta y} + \frac{u^{t}(i+1,j)\left(v^{t}(i,j)+v^{t}(i,j+1)\right)}{4\Delta y} \tag{15}$$

$$-\frac{v\partial v}{\partial y} = -\frac{v^t(i-1,j)^2 - v^t(i+1,j)^2}{2\Delta y} \tag{16}$$

$$-\frac{v\partial u}{\partial x} = -\frac{v^t(i,j-1)\left(u^t(i,j-1)+u(i+1,j-1)\right)}{4\Delta x} + \frac{v^t(i,j+1)\left(u^t(i,j)+u^t(i+1,j)\right)}{4\Delta x} \tag{17}$$

while convective terms discretized using the upwind scheme are shown in Equations (18) to (21),

$$\begin{aligned}-\frac{u\partial u}{\partial x} = -max\left(\frac{u^t(i,j-1)\left(u^t(i,j-1)+u^t(j,i)\right)}{2\Delta x},0\right) + \\ max\left(-\frac{u^t(i,j+1)\left(u^t(j,i)+u^t(i,j+1)\right)}{2\Delta x},0\right)\end{aligned} \tag{18}$$

$$\begin{aligned}-\frac{u\partial v}{\partial y} = -max\left(\frac{u^t(i-1,j)\left(v^t(i-1,j)+v^t(i-1,j+1)\right)}{2\Delta y},0\right) + \\ max\left(-\frac{u^t(i+1,j)\left(v^t(i,j)+v^t(i,j+1)\right)}{2\Delta y},0\right)\end{aligned} \tag{19}$$

$$\begin{aligned}-\frac{v\partial v}{\partial y} = -max\left(\frac{v^t(i-1,j)\left(v^t(i-1,j)+v^t(j,i)\right)}{2\Delta y},0\right) + \\ max\left(-\frac{v^t(i+1,j)\left(v^t(j,i)+v^t(i+1,j)\right)}{2\Delta y},0\right)\end{aligned} \tag{20}$$

$$\begin{aligned}-\frac{v\partial u}{\partial x} = -max\left(\frac{v^t(i,j-1)\left(u^t(i,j-1)+u^t(i+1,j-1)\right)}{2\Delta x},0\right) + \\ max\left(-\frac{v^t(i,j+1)\left(u^t(i,j)+u^t(i+1,j)\right)}{2\Delta x},0\right)\end{aligned} \tag{21}$$

where Δx and Δy are the values representing the size of space in the discretized domain.

The pressure terms are discretized using the forward difference method, as shown in Equations (22) and (23),

$$-\frac{\partial p}{\partial x} = -\frac{p^t(i,j)-p^t(i,j+1)}{\Delta x} \tag{22}$$

$$-\frac{\partial p}{\partial y} = -\frac{p^t(i,j)-p^t(i+1,j)}{\Delta y} \tag{23}$$

The diffusion terms are discretized using the central difference method, as shown in Equations (24) to (27),

$$\frac{\partial^2 u}{\partial x^2} = \frac{u^t(i,j+1) - 2u^t(i,j) + u^t(i,j-1)}{\Delta x^2} \tag{24}$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u^t(i+1,j) - 2u^t(i,j) + u^t(i-1,j)}{\Delta y^2} \tag{25}$$

$$\frac{\partial^2 v}{\partial x^2} = \frac{v^t(i,j+1) - 2v^t(i,j) + v^t(i,j-1)}{\Delta x^2} \tag{26}$$

$$\frac{\partial^2 v}{\partial y^2} = \frac{v^t(i+1,j) - 2v^t(i,j) + v^t(i-1,j)}{\Delta y^2} \tag{27}$$

The discretized terms can be substituted in Navier-Stokes Equations (10) and (11), resulting in the following forms, as shown in Equations (28) and (29),

$$u^{t+1}(i,j) = u^t(i,j) + \Delta t \left(function(u^t, v^t, \rho, \mu, \Delta x, \Delta y) - \frac{(p(i,j) - p(i,j+1))}{\Delta x} \right) \tag{28}$$

$$v^{t+1}(i,j) = v^t(i,j) + \Delta t \left(function(u^t, v^t, \rho, \mu, \Delta x, \Delta y) - \frac{(p(i,j) - p(i+1,j))}{\Delta y} \right) \tag{29}$$

The continuity equation (Equation (9)) is discretized using the forward difference method, as shown in Equation (30),

$$\frac{(u^{t+1}(i,j-1) - u^{t+1}(i,j))}{\Delta x} + \frac{(v^{t+1}(i+1,j) - v^{t+1}(i,j))}{\Delta y} = residual \tag{30}$$

where $residual$ indicates the extent of convergence. Although its ideal value is zero, a small number is acceptable for a numerical solution.

It is to be noted that Navier-Stokes equations (Equations (28) and (29)) are discretized in time domain t. However, the continuity equation (Equation (30)) has been discretized in time domain t+1. This is because the velocities in the new domain are calculated using Navier-Stokes discretized equations (Equations (28) and (29)) and checked in the discretized continuity equation (Equation (30)). The solution is iterated until the residual reduces to a small number.

Since the solution is calculated via Navier-Stokes discretized equations (Equations (28) and (29)), it always conforms with the law of conservation of momentum. However, it may not conform with the law of conservation of mass as defined by the discretized continuity equation (Equation (30)). In order to obtain convergence for both, laws of conservation of momentum and mass, corrections need to be applied to velocities and pressure variables, as shown in Equations (31) to (33),

$$u^{t+1}(i,j)_{corrected} = u^{t+1}(i,j) + du^{t+1}(i,j) \tag{31}$$

$$v^{t+1}(i,j)_{corrected} = v^{t+1}(i,j) + dv^{t+1}(i,j) \tag{32}$$

$$p^{t+1}(i,j)_{corrected} = p^{t+1}(i,j) + dp^{t+1}(i,j) \tag{33}$$

where $u^{t+1}(i,j)_{corrected}$, $v^{t+1}(i,j)_{corrected}$, and $p^{t+1}(i,j)_{corrected}$ are the corrected values of velocity u, velocity v and pressure p that satisfy both continuity and Navier-Stokes discretized equations (Equations (28) to (30)). $du^{t+1}(i,j)$, $dv^{t+1}(i,j)$, and $dp^{t+1}(i,j)$ are the differences between the corrected values and are calculated for velocity u, velocity v and pressure p, respectively.

Substituting the corrected values in the discretized Navier-Stokes equations (Equations (28) and (29)) results in Equations (34) and (35),

$$-du^{t+1}(i,j) = \Delta t\left(\frac{dp^{t+1}(i,j) - dp^{t+1}(i,j+1)}{\Delta x}\right) \tag{34}$$

$$-dv^{t+1}(i,j) = \Delta t\left(\frac{dp^{t\,+1}(i,j) - dp^{t+1}(i+1,j)}{\Delta y}\right) \tag{35}$$

Substituting Equations (34) and (35) in the discretized continuity equation (Equation (30)) results in Equation (36),

$$\Delta t\left(\begin{array}{c}\left(\frac{dp^{t+1}(i,j-1) - dp^{t+1}(i,j) - dp^{t+1}(i,j)dp^{t+1}(i,j+1)}{\Delta x^2}\right) + \\ \left(\frac{dp^{t+1}(i-1,j) - dp^{t+1}(i,j) - dp^{t+1}(i,j) + dp^{t+1}(i+1,j)}{\Delta y^2}\right)\end{array}\right) = residual \tag{36}$$

Equation (36) can be rearranged in Equation (37),

$$\begin{aligned}\alpha dp^{t+1}(i,j) + \beta dp^{t+1}(i+1,j) + \beta dp^{t+1}(i-1,j)\\ + \beta dp^{t+1}(i,j+1) + \beta dp^{t+1}(i,j-1) = -\frac{residual}{\Delta t}\end{aligned} \tag{37}$$

where α and β are given in Equations (38) and (39),

$$\alpha = 2\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right) \tag{38}$$

$$\beta = -\frac{1}{\Delta x^2} \tag{39}$$

Equation (37) can be rearranged, as shown in Equation (40),

$$\alpha\beta\begin{pmatrix}\alpha/\beta & \cdots & 1\\ \vdots & \ddots & \vdots\\ 1 & \cdots & \alpha/\beta\end{pmatrix}\begin{bmatrix}dP^{t+1}(1,1)\\ \vdots\\ dP^{t+1}(n,n)\end{bmatrix} = -\frac{residual}{\Delta t}\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix} \tag{40}$$

where α and β can be rearranged in the form of a sparse matrix.

The above equation allows solving for residuals in an implicit manner. The obtained pressure corrections can be substituted for velocity corrections (Equations (41) and (43)). To stabilize the solution, the $relaxation\_factor$ is introduced with a value between 0 to 1.

$$u^{t+1}(i,j)_{corrected} = u^{t+1}(i,j) + relaxation\_factor.du^{t+1}(i,j) \tag{41}$$

$$v^{t+1}(i,j)_{corrected} = v^{t+1}(i,j) + relaxation\_factor.dv^{t+1}(i,j) \tag{42}$$

$$p^{t+1}(i,j)_{corrected} = p^{t+1}(i,j) + relaxation\_factor.dp^{t+1}(i,j) \tag{43}$$

In addition, for stability and accuracy, it is vital to choose the correct timestep value. In this work, the Courant–Friedrichs–Lewy (CFL) condition [1, 21] is used to decide the timestep size. The CFL condition is given in Equation (44),

$$max(u,v).\Delta t \leq min(\Delta x, \Delta y) \tag{44}$$

The given method of solution is solved and post-processed in MATLAB® [22]. The flow chart of the code, which can be accessed from the Mathworks repository [23], is given in Fig. 4. The results are discussed in the next section.

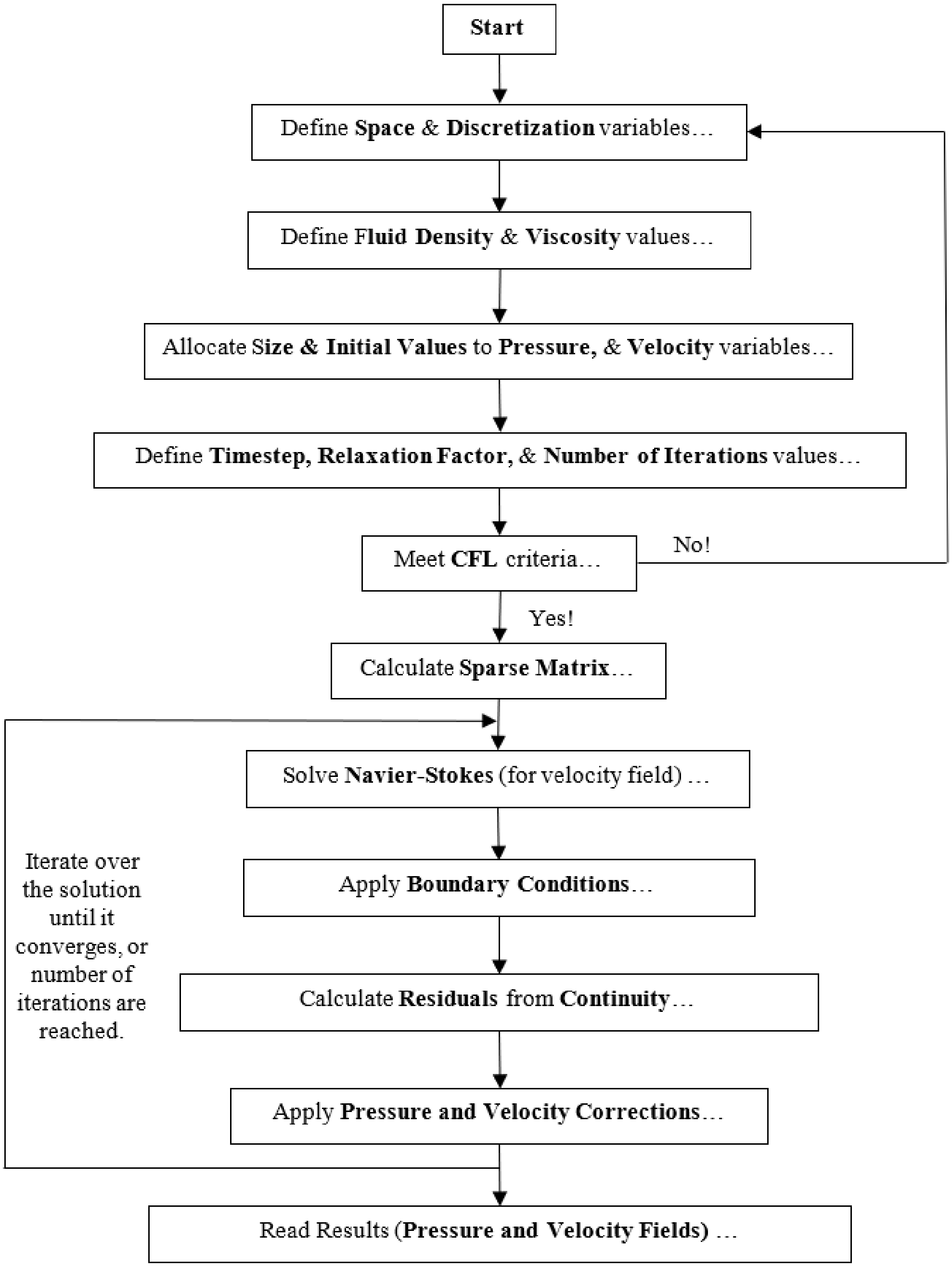

Start
Define Space & Discretization variables…
Define Fluid Density & Viscosity values…
Allocate Size & Initial Values to Pressure, & Velocity variables…
Define Timestep, Relaxation Factor, & Number of Iterations values…
Meet CFL criteria…
No!
Yes!
Calculate Sparse Matrix…
Solve Navier-Stokes (for velocity field) …
Iterate over the solution until it converges, or number of iterations are reached.
Apply Boundary Conditions…
Calculate Residuals from Continuity…
Apply Pressure and Velocity Corrections…
Read Results (Pressure and Velocity Fields) …

Figure 4. Flowchart of the code.

3. RESULTS AND DISCUSSION

The pressure and velocity fields are obtained in a 2-D domain, as shown in Fig. 5. The velocity vectors show that the flow was diverted at the corners of the square. Pressure fields demonstrate that the high pressures were developed in front of the square and low pressure behind the square.

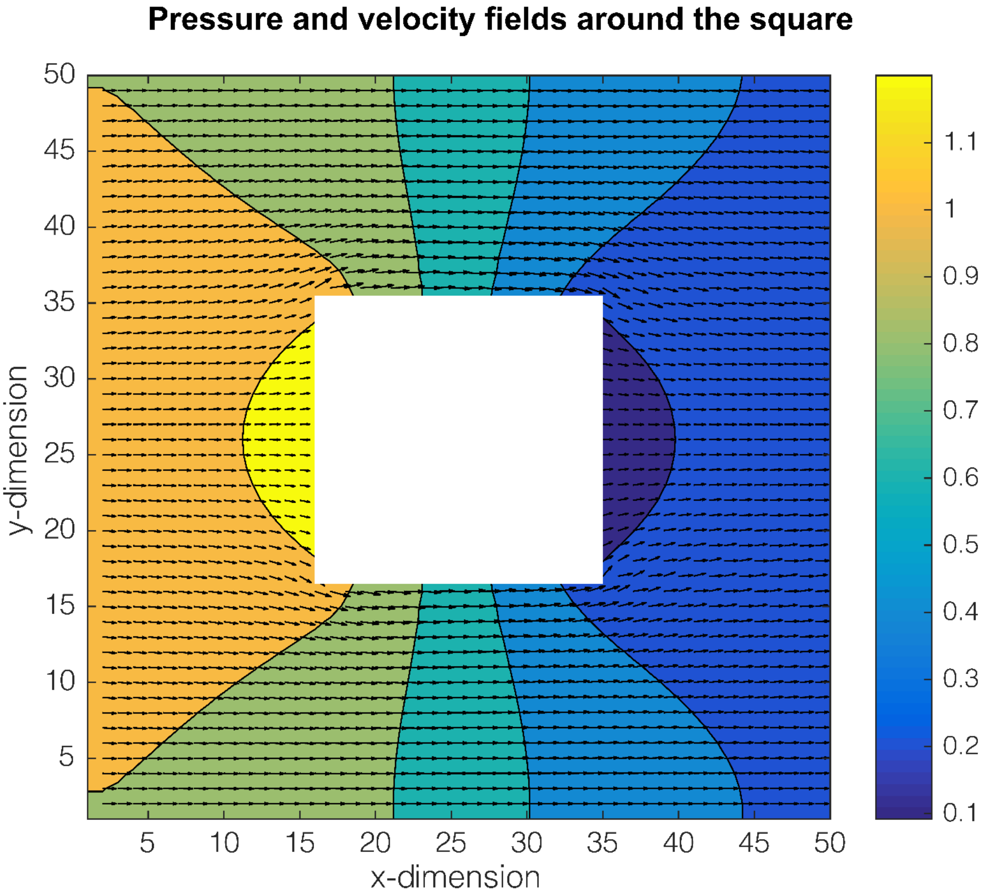


Figure 5. Pressure and velocity fields around a square in the 2-D domain.

In the given case, the maximum value of the residual value dropped below 10^{-4}. The maximum values of residuals from the continuity equation are plotted against the iteration number, as shown in Fig. 6.

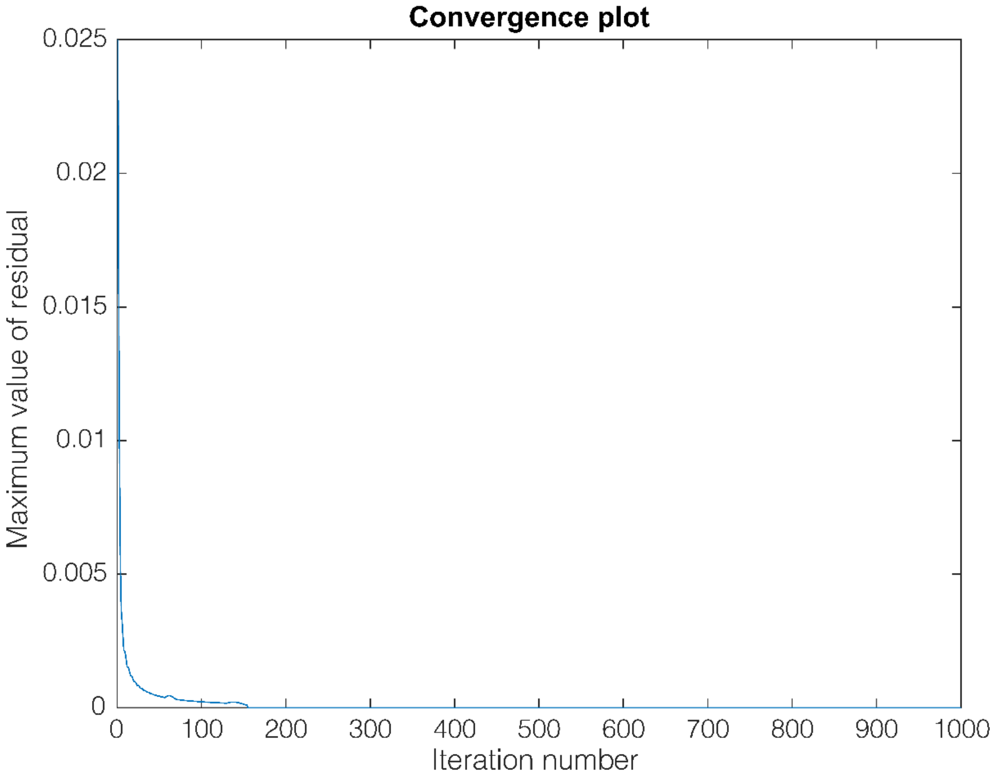


Figure 6. Convergence plot: the maximum value of the residual is plotted against the iteration number.

4. CONCLUSION

The presented study demonstrates that flow around a square in a 2-D domain can be successfully solved using the SIMPLE algorithm. The paper offers an explanation of the SIMPLE algorithm, detailing assumptions, degrees of freedom, discretization, as well as pressure and velocity corrections. The method of solution is given in the form of a flowchart and the code is implemented in MATLAB® and can be accessed from [23].

ACKNOWLEDGMENTS

The author would also like to acknowledge the support of Linda March from The Good English Company, UK, for proofreading this work.

REFERENCES

[1] S. Patankar. Numerical Heat Transfer and Fluid Flow: Taylor & Francis; 1980.

[2] H. Khawaja. CFD-DEM Simulation of Minimum Fluidisation Velocity in Two Phase Medium. The International Journal of Multiphysics. 2011;5(2):89-100.

[3] H. Khawaja. CFD-DEM and Experimental Study of Bubbling in a Fluidized Bed. The Journal of Computational Multiphase Flows. 2015;7(4):227-40.

[4] H. Khawaja, T. Rashid, O. Eiksund, E. Brodal, K. Edvardsen. Multiphysics Simulation of Infrared Signature of an Ice Cube. The International Journal of Multiphysics. 2016;10(3):291-302.

[5] J. Anderson. Computational Fluid Dynamics: McGraw-Hill Education; 1995.

[6] R. H. Pletcher, J. C. Tannehill, D. Anderson. Computational Fluid Mechanics and Heat Transfer, Third Edition: CRC Press; 2016.

[7] H. Khawaja. Application of a 2-D approximation technique for solving stress analyses problem in FEM. The International Journal of Multiphysics. 2015;9(4):317-24.

[8] H. Khawaja, S. Scott. CFD-DEM Simulation of Propagation of Sound Waves in Fluid Particles Fluidised Medium. The International Journal of Multiphysics. 2011;5(1):47-60.

[9] H. Khawaja, S. Scott, M. Virk, M. Moatamedi. Quantitative Analysis of Accuracy of Voidage Computations in CFD-DEM Simulations. The Journal of Computational Multiphase Flows. 2012;4(2):183-92.

[10] H. Khawaja, I. Raouf, K. Parvez, A. Scherer. Optimization of elstomeric micro-fluidic valve dimensions using non-linear finite element methods. The International Journal of Multiphysics. 2009;3(2):187-200.

[11] H. Khawaja, K. Parvez. Validation of normal and frictional contact models of spherical bodies by FEM analysis. The International Journal of Multiphysics. 2010;4(2):175-85.

[12] H. Khawaja, M. Moatamedi. Selection of High Performance Alloy for Gas Turbine Blade Using Multiphysics Analysis. The International Journal of Multiphysics. 2014;8(1):91-100.

[13] H. Khawaja, T. Bertelsen, R. Andreassen, M. Moatamedi. Study of CRFP Shell Structures under Dynamic Loading in Shock Tube Setup. Journal of Structures. 2014;2014(Article ID 487809):6.

[14] H. Khawaja, R. Messahel, B. Ewan, S. Mhamed, M. Moatamedi. Experimental and Numerical Study of Pressure in a Shock Tube. Journal of Pressure Vessel Technology. 2016;138(4):6.

[15] U. Mughal, H. Khawaja, M. Moatamedi. Finite element analysis of human femur bone. The International Journal of Multiphysics. 2015;9(2):101-8.

[16] M. Kim. Finite Element Methods with Programming and Ansys: LULU Press; 2013.

[17] T. Stolarski, Y. Nakasone, S. Yoshimoto. Engineering Analysis with ANSYS Software: Elsevier Science; 2011.

[18] Y. W. Kwon, H. Bang. The Finite Element Method Using MATLAB, Second Edition: CRC Press; 2000.

[19] H. D. Young, A. L. Ford, R. A. Freedman. University Physics: Pearson Education; 2008.

[20] H. Khawaja. Applicability extent of 2-D heat equation for numerical analysis of a multiphysics problem. AIP Conference Proceedings. 2017;1798(1):020075.

[21] R. Courant, K. Friedrichs, H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen. 1928;100(1):32-74.

[22] MATLAB®. Natick, Massachusetts: The MathWorks Inc.; 2016.

[23] H. Khawaja. Semi-Implicit Method for Pressure Linked Equations (SIMPLE) – solution in MATLAB®. https://se.mathworks.com/matlabcentral/fileexchange/66129-simple 2018.