Novel fully Implicit Collocated grid incompressible flow solver on unstructured meshes

Jobin Joseph Unnupatt\textsuperscript{1}, N.Munikrishna\textsuperscript{2}, N.Balakrishnan\textsuperscript{3}

1: Research scholar, CAd Lab, Department of Aerospace Engineering, Indian Institute of Science, Bangalore.
2: CTO, S & I Engineering Solutions Pvt. Ltd., Bangalore.
3: Associate Professor, Department of Aerospace Engineering, Indian Institute of Science, Bangalore. Email: nbalak@aero.iisc.ernet.in

Abstract.
This article presents a novel implicit cell center finite volume scheme for solving two-dimensional incompressible steady state flows. A collocated approach has been employed considering its use in conjunction with unstructured meshes. The method inspired by the methodologies for compressible flow solvers is evolved to naturally accommodate several modules from such solvers. The explicit fluxes are computed to second order accuracy using the method of reconstruction. An inconsistent linearization is adopted for the implicit part which employs SGS sub-iterations. At the heart of the implicit procedure is obtaining an auxiliary velocity field, the divergence of which results in a Poisson equation for the pressure field, ensuring a divergence free velocity field resulting from the implicit iteration. The efficacy of the proposed method is demonstrated for several test problems, such as the flow past a Circular Cylinder, NACA 0012 airfoil and lid driven cavity.

1. Introduction
The incompressible flow solvers can be broadly classified as staggered and collocated grid methods, based on the location the pressure and momentum are stored [1,2]. The staggered grid approach, for which the location where the pressure and momentum data are staggered, may not be suited for unstructured data based flow solvers allowing for the use of arbitrary polyhedral volumes. Particularly, when we have access to a compressible flow solver, with its typical modules for solution reconstruction[3], implicit update[4], time accuracy[5] and solution based grid adaption[6], it becomes imperative that most of these modules are adopted for the development of an incompressible flow solver, without resorting to major code changes. Towards this objective, while a preconditioned compressible flow solver can be an interesting option, often, in terms of usage, it gives a lot of flexibility to handle incompressibility in its own right. The present effort in the development of a fully implicit collocated grid incompressible flow solver is inspired by such a requirement.

2. Methodology
Consider the Navier Stokes equations as applied to the incompressible flows given by

\[
\frac{\partial U}{\partial t} + \frac{\partial}{\partial x} (f_c + P_1 - f_v) + \frac{\partial}{\partial y} (g_c + P_2 - g_v) = 0
\]

where \(U = [\rho u \quad \rho v]^T\) represents the vector of conserved variables, with \(\rho\) standing for the fluid density and \(u\) and \(v\) represent the \(x\) and \(y\) components of the fluid velocity.
The flux vectors are given as below:

\[
\begin{align*}
    f_c &= \left[ \begin{array}{c} \rho u^2 \\ \rho u v \\
    \end{array} \right], \\
    f_v &= \left[ \begin{array}{c} \rho \frac{\partial \xi}{\partial x} \\ \rho \frac{\partial \eta}{\partial x} \\
    \end{array} \right], \\
    g_c &= \left[ \begin{array}{c} \rho u v \\ \rho v^2 \\
    \end{array} \right], \\
    P_1 &= \left[ \begin{array}{c} p \\
    0 \\
    \end{array} \right], \\
    g_v &= \left[ \begin{array}{c} \rho \frac{\partial \xi}{\partial y} \\ \rho \frac{\partial \eta}{\partial y} \\
    \end{array} \right].
\end{align*}
\]

Defining \( \vec{F}_c = (f_c, g_c) \), \( \vec{P} = (P_1, P_2) \) and \( \vec{F}_v = (f_v, g_v) \), Eqn.(1) can be recast as,

\[
\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\vec{F}_c + \vec{P} - \vec{F}_v) = 0 \tag{2}
\]

For a finite volume \( i \) sharing its \( J^{th} \) face with its neighbour \( j \), the state update formula can be written as,

\[
\frac{\Delta t U_i}{\Delta t} + \frac{1}{\Omega_i} \sum_{J} \left( \vec{F}_c + \vec{P} - \vec{F}_v \right)_J \cdot \hat{n}_J \Delta S_J = 0 \tag{3}
\]

where \( \Delta t U_i = U_i^{n+1} - U_i^n \), \( \Omega_i \), \( \hat{n}_J \) and \( \Delta S_J \) represent the volume of the \( i^{th} \) cell, unit normal and area (or length in 2D) of the \( J^{th} \) interface respectively. Eqn.(3) can be further simplified as,

\[
\frac{\Delta t U_i}{\Delta t} + \frac{1}{\Omega_i} \sum_{J} (F_{c\perp} + P_{\perp} - F_{v\perp}) \Delta S_J = 0 \tag{4}
\]

with, \( F_{c\perp} = [\rho uu_{\perp}, \rho uu_{\perp}]^T \), \( P_{\perp} = [p_{nx}, p_{ny}]^T \), \( F_{v\perp} = [ \mu \frac{\partial u}{\partial n}, \mu \frac{\partial v}{\partial n}]^T \) and \( u_{\perp} = (\vec{u} \cdot \hat{n})_J \).

The second term in Eqn.(4), often referred to as residual, a discrete representation for the flux integral, is a function of \( \mathbf{U} \) and can be determined either at \( n^{th} \) time level for an explicit procedure or at \( (n+1)^{th} \) time level for an implicit procedure. Though the work primarily deals with the implicit procedure, explicit procedure is described to start with.

2.1. Explicit Procedure

One of the major challenges in the design of incompressible flow solvers is the lack of a governing equation for pressure update. Often the pressure field is recovered by manipulating the mass conservation equation. Hence, we describe an auxiliary velocity based procedure for recovering an equation for pressure update. Towards this, an equation for auxiliary velocity update is written by dropping the pressure term in Eqn.(4) as

\[
\frac{\Delta t \tilde{U}_i}{\Delta t} + \frac{1}{\Omega_i} \sum_{J} (F_{c\perp} - F_{v\perp})_J^n \Delta S_J = 0 \tag{5}
\]

where, \( \Delta t \tilde{U}_i = \tilde{U}_i - U_i^n \). Subtracting Eqn.(5) from Eqn.(4), we have,

\[
\frac{U_i^{n+1} - \tilde{U}_i}{\Delta t} = - \frac{1}{\Omega_i} \sum_{J} P_{\perp}^{n+1} \Delta S_J. \tag{6}
\]

The RHS of the above equation represents an approximation for the volume integral of the pressure gradient, as prescribed below:

\[
\int_{\Omega_i} \nabla p d\Omega = \int_{\Gamma_i} p d\vec{\Gamma} \cong \sum_{J} p_{\perp} \hat{n}_J \Delta S_J. \tag{7}
\]

Realizing that the finite volume state update formula returns cell averaged values, we have from Eqn.(6),

\[
\int_{\Omega_i} \left[ \frac{U_i^{n+1} - \tilde{U}_i}{\Delta t} + \nabla p \right] d\Omega = 0. \tag{8}
\]
It follows from Eqn.(8),
\[ \triangle t \nabla^2 p - \nabla \cdot \mathbf{U} = 0 \]
(9)
where, the fact that the velocity field in the \((n + 1)\)th time is divergence free has been invoked. A more practical way to obtain an iterative solution to Eqn.(8) is introducing a pseudo-time derivative as below:
\[ \frac{\partial p}{\partial \tau} = \triangle t \nabla^2 p - \nabla \cdot \mathbf{U} \]
(10)
Using Eqn.(10), knowing the auxiliary velocity field, the pressure field can be updated. The above pressure equation can be solved implicitly. Eqn.(8) also suggests the auxiliary velocity field can be corrected to obtain the velocity field at the \((n + 1)\)th time level as below:
\[ U_i^{n+1} = \mathbf{U}_i - \triangle t \nabla p_i. \]
(11)

It is important to note that in Eqn.(5), the convective fluxes are determined using an upwind derivative as below:
\[ A \]
respectively, with
\[ \text{The third and fourth terms in the RHS represent the convective and viscous flux linearization respectively, with } A_i^+ = a_i^+ I, \ A_i^- = a_i^- I \text{ and} \]
\[
\begin{align*}
a_i^+ &= \begin{cases} 
0, & \text{ for } Pe_i < -1 \\
0.25d(Pe_i + 1)^2, & \text{ for } -1 \leq Pe_i \leq 1 \\
Pe_i \cdot d, & \text{ for } 1 < Pe_i 
\end{cases} \\
a_j^- &= \begin{cases} 
0, & \text{ for } Pe_j < 1 \\
-0.25d(-Pe_j + 1)^2, & \text{ for } -1 \leq Pe_j \leq 1 \\
Pe_j \cdot d, & \text{ for } -1 > Pe_j 
\end{cases}
\end{align*}
\]

2.2. Implicit procedure
As remarked earlier, an implicit update demands that the residual is computed at \((n + 1)\)th time level and Eqn.(4) can be recast as below:
\[ \frac{\triangle t U_i}{\triangle t} + \frac{1}{\Omega_i} \sum_j \left[ F_{cL}^+ (U_L) + F_{cR}^- (U_R) \right] \triangle S_j = 0 \]
(13)
An inconsistent linearization of the above equation results in,
\[ \frac{\triangle t U_i}{\triangle t} = -\frac{1}{\Omega_i} \left( \sum_j (F_{cL} - F_{cR})\triangle S_j + \sum_j P_{n+1} \triangle S_j + \sum_j (A_i^+ \triangle t U_i + A_j^+ \triangle t U_j) \triangle S_j + \sum_j \nu \frac{\triangle t U_j - \triangle t U_i}{|r_{ij} \cdot n_j|} \triangle S_j \right) \]
(14)
The third and fourth terms in the RHS represent the convective and viscous flux linearization respectively, with \(A_i^+ = a_i^+ I, \ A_i^- = a_i^- I \) and
We employ an SGS iteration as described in the following pseudo code:

\[ T = r \cdot n \quad d = \frac{\nu}{r} \quad Pe_i = \frac{u_i r}{\nu} \quad Pe_j = \frac{u_j r}{\nu}. \]

The above linearization of the convective flux is continuously differentiable and known to significantly enhance the performance of the implicit solver. An update for the auxiliary velocity field may be obtained from Eqn.(14) simply by dropping the pressure terms in the RHS.

\[
\frac{\Delta t \tilde{U}_i}{\Delta t} = -\frac{1}{\Omega_i} \left[ \sum_j (F_{C,1} - F_{V,1})^n_j \Delta S_J + \sum_j \left( A_j^+ \Delta t U_i + A_j^- \Delta t U_i \right) \Delta S_J + \sum_j \nu \frac{\Delta t U_j - \Delta t U_i}{|r_j \cdot n_j|} \Delta S_J \right].
\]

(15)

Subtracting Eqn.(15) from Eqn.(14) results in,

\[
\frac{\Delta t U_i - \Delta t \tilde{U}_i}{\Delta t} = -\frac{1}{\Omega_i} \sum J P_{n+1}^+ \Delta S_J
\]

(16)

Noting that the velocity field at \((n+1)\)th time level is divergence free, the following Poisson equation for pressure may be obtained:

\[
\nabla \cdot \tilde{U} = \Delta t \nabla^2 p.
\]

(17)

The above equation is solved in pseudo-time implicitly akin to the explicit procedure. The pressure thus obtained may be substituted in Eqn.(14) for velocity update. Owing to the implicit nature of the equation, it is important to establish a robust iterative procedure to solve Eqn.(14). We employ an SGS iteration as described in the following pseudo code:

- **Step 1** Let \( \Delta t U_i^{(0)} = 0 \quad \forall \ i \)

- **Step 2**

\[
\frac{\Delta t \tilde{U}_i}{\Delta t} = -\frac{1}{\Omega_i} \left[ \sum_j (F_{C,1} - F_{V,1})^n_j \Delta S_J + \sum_j \left( A_j^+ \Delta t U_i^{(m-1)} + A_j^- \Delta t U_j^{(m-1)} \right) \Delta S_J + \sum_j \nu \frac{\Delta t U_j^{(m-1)} - \Delta t U_i^{(m-1)}}{|r_j \cdot n_j|} \Delta S_J \right]
\]

(18)

- **Step 3** For obtaining \( p^{(m)} \), solve

\[
\frac{\partial p}{\partial t} = \nabla \cdot \tilde{U} \quad \nabla \cdot \tilde{U}
\]

(19)

- **Step 4** Perform a forward and backward sweep of the SGS iteration, with \( \Delta t w_i^{(0)} = \Delta t U_i^{(m-1)} \quad \forall \ i \)

\[
\Rightarrow \text{For forward sweep, } i = 1 \text{ to } nc
\]

\[
\left[ I + \frac{\Delta t}{\Omega_i} \left( \sum_j A_j^+ \Delta S_J + \sum_j \nu \frac{\nu}{|r_j \cdot n_j|} \Delta S_J \right) \right] \Delta t w_i^{(k)} = -\frac{\Delta t}{\Omega_i} \left[ \sum_j (F_{C,1} - F_{V,1})^n_j \Delta S_J + \sum_j P_{+J}^{(m)} \Delta S_J + \sum_{j < i} \left( A_j^+ + \frac{\nu}{|r_j \cdot n_j|} \right) \Delta t w_j^{(k)} \Delta S_J + \sum_{j > i} \left( A_j^- + \frac{\nu}{|r_j \cdot n_j|} \right) \Delta t w_j^{(k-1)} \Delta S_J \right]
\]

(20)
For backward sweep, $i = n_c$ to 1

$$
[I + \frac{\Delta t}{\Omega_i} \left( \sum_j A^+_i \Delta S_j + \sum_j \frac{\nu}{|\vec{r}_{ji} \cdot \hat{n}_j|} \Delta S_j \right)] \Delta t w^{(k)}_i = -\frac{\Delta t}{\Omega_i} \left[ \sum_j (F_{c,\perp} - F_{c,\parallel}) \Delta S_j + \sum_j P^{(m)} \Delta S_j + \sum_{j>i} \left( A^-_j + \frac{\nu}{|\vec{r}_{ji} \cdot \hat{n}_j|} \right) \Delta t w^{(k)}_j \Delta S_j + \sum_{j<i} \left( A^-_j + \frac{\nu}{|\vec{r}_{ji} \cdot \hat{n}_j|} \right) \Delta t w^{(k-1)}_j \Delta S_j \right]
$$

(21)

$$
\Rightarrow \Delta t U^{(m)}_i = \Delta t w^{(k)}_i \forall i .
$$

- **Step 5** Repeat the steps till convergence

It is important to note that the above implicit iterations have been made possible primarily because $\Delta t \tilde{U}_i$ is allowed to lag by one sub iteration. This is of no consequence, assuming a convergence of $\Delta t U_i$. The number of SGS iterations pertaining to Eqns.[20] and [21] and total number of iterations for the pressure field are left to users choice. In the present work, one SGS iteration corresponding to one forward and one backward sweep and one pressure iteration are found to be optimum. The most important aspect of the present procedure is that velocity resulting from the implicit iterations is divergence free and therefore there is no need for any velocity correction step.

3. Results

In this section, we have presented the validation study of the incompressible solver developed with the proposed methodology. Laminar test cases are considered, the details of the test conditions and grid size are provided in Table.(1). The numerical results are compared with the experiment or standard reference values in Fig.(1) to Fig.(4) and Table(2). The results clearly establish the accuracy of the proposed solution methodology.

| Sl.no | Geometry         | Re  | $\alpha$ deg | Grid type               | Grid size (no of volumes) |
|-------|-----------------|-----|---------------|-------------------------|--------------------------|
| 1     | Circular cylinder | 40  | -             | O type structured       | 8000                     |
| 2     | NACA0012         | 10000 | 0            | hybrid unstructured     | 12164                    |
| 3     | Lid driven Cavity | 10000 | 0            | structured              | 16641                    |

Table 2. Flow over Cylinder: Comparison of drag coefficient ($C_D$)

| Re  | Present work | Lima et al [8] | Park et al [9] | Sucker et al [10] | Triton [11] |
|-----|--------------|-----------------|----------------|-------------------|-------------|
| 10  | 2.78         | 2.81            | 2.78           | 2.67              | 2.22        |
| 20  | 2.03         | 2.04            | 2.01           | 2.08              | 2.22        |
| 40  | 1.53         | 1.54            | 1.51           | 1.73              | 1.48        |

4. Conclusions

The preliminary results presented show lot of promise. The proposed method can be seamlessly integrated into any compressible solver. Efforts are on to further extend the method to turbulent and 3D flows.
Figure 1. Flow over Cylinder at $Re = 40$

Figure 2. Flow over Cylinder: Variation of bubble size with $Re$
Figure 3. Flow over NACA0012 at Re = 10000

- a. Mesh
- b. Cp distribution
- c. Sfc distribution
- d. Separation bubble

Figure 4. Lid driven cavity flow at Re = 10000

- a. X velocity
- b. Y velocity
- c. Vorticity contour
- d. Streamlines contour
5. References

[1] Gresho P M, 1991 *Annu. Rev. Fluid Mech.* **23** 413
[2] Patankar S V, *Numerical Heat Transfer and Fluid Flow* (Florida: CRC Press)
[3] Barth T J 1992 *Proc. Int. Conf. on Numerical Methods in Fluid Dynamics* **414** 240
[4] Shende N and Balakrishnan N 2004 *AIAA Journal* **42** 1863
[5] Jameson A 1991 *AIAA Paper* 1596
[6] Ganesh V, Nikhil V Shende and Balakrishnan N 2009 *Computers & Fluids* **38** 1799
[7] Jawahar P and Hemanth Kamath 2000 *J. Comp. Physics* **164** 165
[8] Lima E Silva A L F, Silveira-Neto and Damasceno J J R 2003 *J. Comp. Physics* **189** 351
[9] Park J, Kwon K and Choi H 1998 *KSME Int. J.* **12** 1200
[10] Sucker D and Brauer H 1975 *Warme Stoffübertragung* **8** 149
[11] Tritton D J 1959 *J. Fluid Mech.* **6** 547
[12] Batchelor G K *An Introduction to Fluid Dynamics* (Cambridge: Cambridge University Press)
[13] Ghia U, Ghia K N and Shin C T 1982 *J. Comput. Physics* **48** 387
[14] HiFUN [www.sandi.co.in](http://www.sandi.co.in)