INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **46**:383–413 (DOI: 10.1002/fld.763)

A multiblock operator-splitting algorithm for unsteady flows at all speeds in complex geometries

Siddharth Thakur[1,*,†,‡] and Jeffrey Wright[2,§]

[1] *Mechanical & Aerospace Engineering Bldg. MAE-A, University of Florida, Gainesville, FL 32611, U.S.A.*
[2] *Streamline Numerics, Inc., 3321 NW 13th Street, Suite A, Gainesville, FL 32609, U.S.A.*

SUMMARY

This paper describes a non-iterative operator-splitting algorithm for computing all-speed flows in complex geometries. A pressure-based algorithm is adopted as the base, in which pressure, instead of density, is a primary variable, thus allowing for a unified formulation for all Mach numbers. The focus is on adapting the method for (a) flows at all speeds, and (b) multiblock, non-orthogonal, body-fitted grids for very complex geometries. Key features of the formulation include special treatment of mass fluxes at control volume interfaces to avoid pressure–velocity decoupling for incompressible (low Mach number limit) flows and to provide robust pressure–velocity–density coupling for compressible (high-speed) flows. The method is shown to be robust for all Mach number regimes for both steady and unsteady flows; it is found to be stable for *CFL* numbers of order ten, allowing large time steps to be taken for steady flows. Enhancements to the method which allow for stable solutions to be obtained on non-orthogonal grids are also discussed. The method is found to be very reliable even in complex engineering applications such as unsteady rotor–stator interactions in turbulent, all-speed turbomachinery flows. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: PISO; unsteady; all-speed; operator-splitting

1. INTRODUCTION

Predictor–corrector operator-splitting methods based on the pressure implicit splitting of operators (PISO) algorithm first proposed by Issa [1] have been in use for unsteady flows using pressure-based solvers for quite some time now. Several variants of the PISO algorithm have appeared in the literature (e.g. see References [2–6]). The original pressure-based methods based on the semi-implicit method for pressure-linked equations (SIMPLE)

*Correspondence to: S. Thakur, Mechanical & Aerospace Engineering Bldg. MAE-A, University of Florida, Gainesville, FL 32611, U.S.A.
†E-mail: sst@mae.ufl.edu
‡Adjunct Associate Scientist.
§Senior Scientist.

Published online 17 August 2004

Received 24 November 2003
Revised 24 May 2004

algorithm [7] were designed for steady-state flows and solve the discretized governing equations sequentially in an iterative manner. The conventional extensions of steady-state algorithms to unsteady flows are also iterative in nature and, thus, computationally expensive. The advantages of methods based on the PISO algorithm include non-iterative solution of the implicitly discretized flow equations and avoidance of the use of flow-dependent relaxation factors. Though the PISO algorithm was formulated for both incompressible and compressible flows [1–3], its application to high-speed compressible flows, especially supersonic flows, has been very limited. Recently, Bressloff [6] has applied an operator-splitting method to high-speed compressible flows; however, the compressible flow applications shown in that work are limited to steady, inviscid (isenthalpic) flows in simple geometries. Moreover, Bressloff [6] reports that their method fails to converge for supersonic flows on relatively fine grids. There is little additional evidence of application of such methods to unsteady, viscous, all-speed flows in very complex geometries. In the present work, an operator-splitting algorithm is developed and implemented in an existing three-dimensional flow solver [8] employing multiblock, body-fitted (curvilinear), structured grids to solve steady and unsteady all-speed flows in complex geometries. The method developed in this paper is applied to several steady and unsteady, incompressible and compressible flows, including supersonic and hypersonic flows on refined grids, with success.

In conventional multiblock methods, an iterative procedure is employed (this is different from the iterative procedure for the sequential solution of the governing equations, as mentioned above) wherein each multiblock iteration consists of block-by-block solution of the discretized equations followed by an exchange of information at the block interfaces. This would negate the benefit gained by using a non-iterative PISO-based algorithm. To avoid this multiblock iteration, an efficient solution procedure is devised wherein the coefficient matrix is assembled for the entire domain and solved using an algebraic multigrid method which uses Gauss–Seidel method as the smoother. The predictor–corrector sequence of the proposed algorithm is similar to that of the original PISO scheme. However, the density–velocity–pressure coupling follows the compressible extension of the original SIMPLE algorithm for steady flows (e.g. References [9, 10]), instead of that proposed by Issa [1]. In this regard, the method presented here can be relatively easily implemented in existing SIMPLE-based steady-state flow solvers. It is found that although the original algorithm works well for simulation of incompressible, subsonic and supersonic flows, it is unstable for grids with skewed cells. To rectify this, enhancements to the original PISO algorithm are proposed in this paper and are demonstrated to be robust. The method is demonstrated to be robust for *CFL* numbers of the order of ten, thus allowing large time steps to be taken for steady flows.

It is the objective of the present paper to present a robust, high-resolution pressure-based algorithm for the efficient computation of steady and unsteady flows at all Mach numbers (ranging from incompressible to hypersonic) in complex three-dimensional geometries. A detailed derivation of the algorithm in body-fitted curvilinear co-ordinates is provided. Special emphasis is given to the pressure–velocity–density coupling for high-speed compressible flows. Enhancements to the method to improve efficiency for multi-block grids are discussed. Modifications necessary to enhance the robustness of the method for non-orthogonal, skewed grids are also highlighted. The method is demonstrated to be robust for steady and unsteady flows at all Mach numbers in complex geometries.

2. OVERVIEW OF THE BASIC COMPUTATIONAL ALGORITHM

2.1. Overall numerical approach

A pressure-based flow solver, named STREAM [8] is used as the basic platform for development of the operator splitting method discussed in this paper. It employs structured, body-fitted grids for computing compressible and incompressible, laminar and turbulent flows. For handling complex geometries, multiblock abutting grids with flux conservation at the block interfaces, are employed [11]. Various convection schemes including first-order upwind, second-order upwind, central difference and QUICK are available in the code [12]. For flows involving shocks, a TVD-based controlled variation scheme (CVS) is also implemented in the flow solver [13].

The flow solver is based on the SIMPLE algorithm [7]. It uses a control volume approach with a collocated arrangement for the velocity components and the scalar variables like pressure. Pressure–velocity decoupling is prevented by employing the momentum interpolation approach [14, 15]; this involves adding a fourth-order pressure dissipation term while estimating the mass flux at the control volume interfaces. The velocity components are computed from the respective momentum equations. The velocity and the pressure fields are corrected using a pressure correction equation. The correction procedure leads to a continuity-satisfying velocity field. The whole process is repeated until the desired convergence is reached. Relevant details of the algorithm can be found in Reference [16]. The details of the implementation of the basic steady state algorithm can be found in Reference [8].

2.2. Governing equations

The instantaneous governing equations at a point in space, written in Cartesian co-ordinates are as follows:

$$\textit{Continuity}: \quad \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{1}$$

$$\textit{Momentum}: \quad \frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{2}$$

$$\textit{Energy}: \quad \frac{\partial}{\partial t}(\rho h_0) + \frac{\partial}{\partial x_j}(\rho u_j h_0) = -\frac{\partial q_j}{\partial x_j} + \frac{\partial}{\partial x_j}(u_i \tau_{ij}) + \frac{\partial p}{\partial t} \tag{3}$$

An equation of state is also required to related pressure and density, which, for an ideal gas, is:

$$p = \rho R T \tag{4}$$

In the above equations, ρ is density, u_j is the velocity vector, p is pressure, T is temperature, R is the gas constant, and h_0 is the total (or stagnation) enthalpy given by

$$h_0 = h + \tfrac{1}{2} u_i u_i \tag{5}$$

For turbulent flows, all of the above variables are Favre-averaged [17]. Assuming a Newtonian fluid and employing the eddy viscosity hypothesis to model turbulence, the stress tensor in

the momentum equation is written as:

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

where μ_t is the eddy (turbulent) viscosity to be defined later. The heat flux vector in the energy equation is obtained from Fourier's law:

$$q_j = -\left(\kappa + \frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial x_j} \tag{7}$$

where δ_{ij} is the Kronecker delta and κ is the thermal conductivity; Pr_L is the laminar Prandtl number defined as:

$$Pr_L = \frac{C_p \mu}{\kappa} \tag{8}$$

where C_p is the specific heat of the fluid at constant pressure.

The transformation of Equations (1) and (2) to body-fitted curvilinear co-ordinates (ξ, η, ζ), shown in Figure 1(a), yields the following equations:

Continuity:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial \xi}(\rho U) + \frac{\partial}{\partial \eta}(\rho V) + \frac{\partial}{\partial \zeta}(\rho W) = 0 \tag{9}$$

u-, *v*- or *w*-momentum ($i = 1$, 2 or 3):

$$\begin{aligned}
&\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial \xi}(\rho U u_i) + \frac{\partial}{\partial \eta}(\rho V u_i) + \frac{\partial}{\partial \zeta}(\rho W u_i) \\
&\quad = -\left(f_{1i}\frac{\partial p}{\partial \xi} + f_{2i}\frac{\partial p}{\partial \eta} + f_{3i}\frac{\partial p}{\partial \zeta}\right) \\
&\qquad + \frac{\partial}{\partial \xi}\left[\frac{\Gamma}{J}\left(q_{11}\frac{\partial u_i}{\partial \xi} + q_{12}\frac{\partial u_i}{\partial \eta} + q_{13}\frac{\partial u_i}{\partial \zeta}\right)\right] \\
&\qquad + \frac{\partial}{\partial \eta}\left[\frac{\Gamma}{J}\left(q_{21}\frac{\partial u_i}{\partial \xi} + q_{22}\frac{\partial u_i}{\partial \eta} + q_{23}\frac{\partial u_i}{\partial \zeta}\right)\right] \\
&\qquad + \frac{\partial}{\partial \zeta}\left[\frac{\Gamma}{J}\left(q_{31}\frac{\partial u_i}{\partial \xi} + q_{32}\frac{\partial u_i}{\partial \eta} + q_{33}\frac{\partial u_i}{\partial \zeta}\right)\right]
\end{aligned} \tag{10}$$

where $\Gamma = \mu + \mu_t$. The metric terms in the above equations resulting from the co-ordinate transformation are given by the following:

$$f_{11} = y_\eta z_\zeta - z_\eta y_\zeta, \quad f_{12} = z_\eta x_\xi - x_\eta z_\xi, \quad f_{13} = x_\eta z_\zeta - y_\eta z_\zeta$$

$$f_{21} = z_\xi y_\zeta - y_\zeta z_\xi, \quad f_{22} = x_\xi z_\zeta - z_\xi x_\zeta, \quad f_{23} = y_\xi x_\zeta - x_\xi y_\zeta$$

$$f_{31} = y_\xi z_\eta - z_\xi y_\eta, \quad f_{32} = z_\xi x_\eta - x_\xi z_\eta, \quad f_{33} = x_\xi y_\eta - y_\xi x_\eta$$

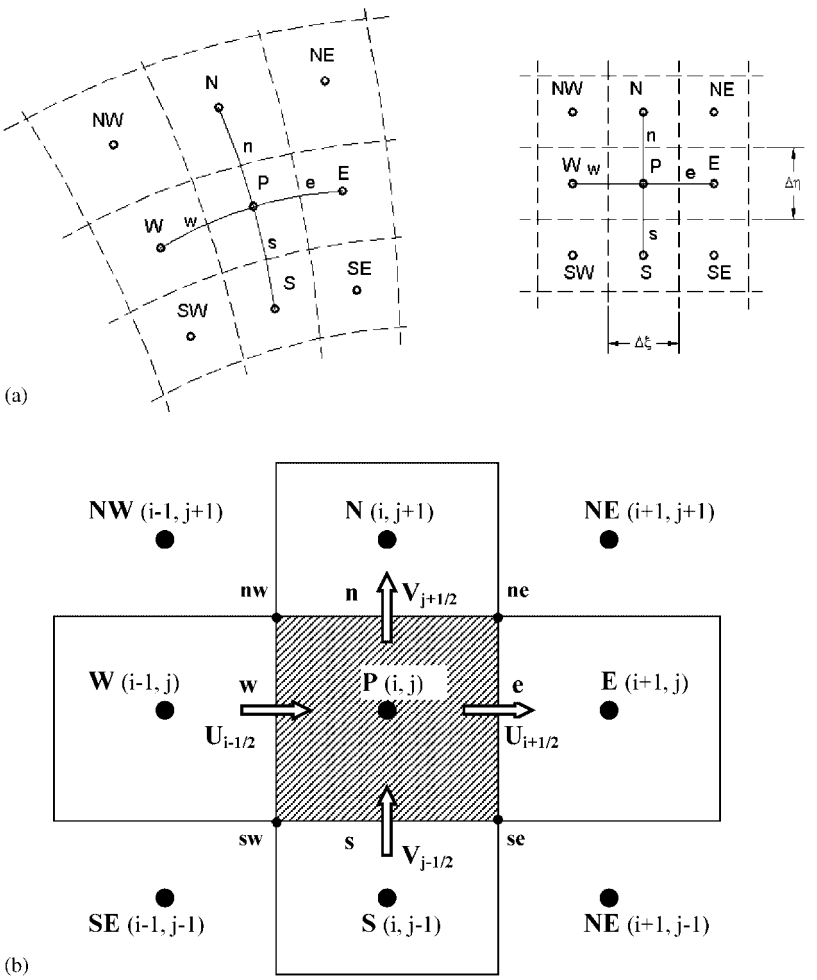


Figure 1. Notation for body-fitted, curvilinear grids: (a) Notation for a 2-D collocated grid: physical plane (left) and transformed plane (right); and (b) schematic of a control volume showing the face contravariant velocities.

$$
\begin{aligned}
q_{11} &= f_{11}^2 + f_{12}^2 + f_{13}^2, \quad q_{12} = q_{21} = f_{11}f_{21} + f_{12}f_{22} + f_{13}f_{23} \\
q_{22} &= f_{21}^2 + f_{22}^2 + f_{23}^2, \quad q_{13} = q_{31} = f_{11}f_{31} + f_{12}f_{32} + f_{13}f_{33} \\
q_{33} &= f_{31}^2 + f_{32}^2 + f_{33}^2, \quad q_{23} = q_{32} = f_{31}f_{21} + f_{32}f_{22} + f_{33}f_{23}
\end{aligned} \tag{11}
$$

and J is the Jacobian of the transformation which is also the physical volume of the computational control volume; it is given by

$$
J = x_\xi y_\eta z_\zeta + x_\eta y_\zeta z_\xi + x_\zeta y_\xi z_\eta - x_\zeta y_\eta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta \tag{12}
$$

In the above equations, U, V and W are contravariant velocities given by

$$\begin{aligned} U &= f_{11}u + f_{12}v + f_{13}w \\ V &= f_{21}u + f_{22}v + f_{23}w \\ W &= f_{31}u + f_{32}v + f_{33}w \end{aligned} \tag{13}$$

The energy equation can be written in a similar manner.

When the equations are solved in a frame of reference rotating with a steady angular velocity $\mathbf{\Omega}$, the following source terms corresponding to Coriolis and centrifugal forces need to be included in the momentum equation:

$$S = -2\mathbf{\Omega} \times \mathbf{U} - \mathbf{\Omega} \times (\mathbf{\Omega} \times \mathbf{R}) \tag{14}$$

In the SIMPLE family of algorithms, the continuity equation is replaced by a pressure correction equation, which is formulated by manipulating the continuity and momentum equations. The discretized equations are solved sequentially in an iterative manner. At each iteration, the computed pressure correction field is used to correct the pressure and velocity fields to enforce the continuity equation.

2.3. Turbulence model

For turbulence closure, the standard k–ε model [18] is employed. Eddy viscosity is estimated from the turbulent kinetic energy (k) and the rate of dissipation of turbulent kinetic energy (ε) by the following relationship:

$$\mu_t = \frac{C_\mu \rho k^2}{\varepsilon} \tag{15}$$

The k and ε are estimated by their own transport equations which can be written, in Cartesian co-ordinates, as the following:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P - \rho\varepsilon \tag{16}$$

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \frac{\partial}{\partial x_j}(\rho u_j \varepsilon) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + C_1 P\frac{\varepsilon}{k} - C_2\rho\frac{\varepsilon^2}{k} \tag{17}$$

where

$$P = \tau_{ij}\frac{\partial u_i}{\partial x_j} = \left\{\mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu_t\frac{\partial u_l}{\partial x_l}\delta_{ij}\right\}\cdot\frac{\partial u_i}{\partial x_j} \tag{18}$$

The transformation to curvilinear co-ordinates is similar to the momentum equations shown earlier. The constants in the above model have the following values:

$$C_\mu = 0.09, \quad C_1 = 1.44, \quad C_2 = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3$$

Standard wall functions are used to estimate the velocity, k and ε at the first node near a solid wall [18]. They are based on the assumption that the boundary layer structure near

any solid wall is similar to that of a flat plate boundary layer and that turbulence is in local equilibrium.

3. PISO-BASED ALGORITHM FOR COLLOCATED BODY-FITTED GRIDS

In this section, the discretized equations for the proposed operator-splitting algorithm are derived for compressible flows using curvilinear body-fitted co-ordinates along with a collocated arrangement of flow variables. For clarity and brevity, the derivation is presented for a two-dimensional situation. However, there is no loss of generality; the extension to three-dimensional flows is straightforward. The notation used for a two-dimensional collocated grid in curvilinear co-ordinates is shown in Figure 1(a). A representative control volume is shown in Figure 1(b). Point P is the control point under consideration and is referred to by the index (i, j). E, W, N and S are the immediate east, west, north and south neighbours, respectively, of point P. The faces of the control volume are labelled as e, w, n and s or, equivalently, $i+\frac{1}{2}$, $i-\frac{1}{2}$, $j+\frac{1}{2}$, $j-\frac{1}{2}$. The contravariant velocities are stored at the control volume interfaces as shown in Figure 1(b).

First consider the discretized equation governing u-momentum using the fully implicit backward-Euler time-stepping scheme. In curvilinear co-ordinates, this equation can be written for the control volume around point P as

$$\left(A_P + \frac{\rho^{n+1}J}{\Delta t}\right) u^{n+1} = H^n(u^{n+1}) + \frac{\rho^n u^n J}{\Delta t} - f_{11}(p_e - p_w)^n - f_{21}(p_n - p_s)^n \tag{19}$$

where A_P is the nodal coefficient involving only the spatial discretization. The term H^n contains all terms resulting from the discretization of convective and diffusive fluxes as well as any source terms (such as higher-order convective flux terms, viscous cross-derivative terms and physical source terms, if any) with the exception of the pressure gradient terms. Note that the superscript n denotes the old (or the present) time level and $n+1$ denotes the new time level being computed. The implicit solution of the above equation requires several iterations at every time step, which is very inefficient. The unsteady algorithm developed in the present work seeks to eliminate this iterative solution procedure by employing a series of predictor–corrector steps. The solution field at the predictor stage is denoted by the superscript *, that in the successive corrector steps by **, ***, etc. The solution obtained at the last corrector step is considered as the solution at the new time level $(n+1)$. In the following sections, we describe each of the predictor and corrector steps in detail.

3.1. Momentum predictor step

In the momentum predictor step, the u- and v-momentum equations are solved implicitly and can be written as follows:

$$\left(A_P + \frac{\rho^n J}{\Delta t}\right) u^* = H^n(u^*) + \frac{\rho^n u^n J}{\Delta t} - f_{11}(p_e - p_w)^n - f_{21}(p_n - p_s)^n \tag{20}$$

$$\left(A_P + \frac{\rho^n J}{\Delta t}\right) v^* = H^n(v^*) + \frac{\rho^n v^n J}{\Delta t} - f_{12}(p_e - p_w)^n - f_{22}(p_n - p_s)^n \tag{21}$$

Note that the above equations use the pressure field at the time level t^n. Also, it should be noted that we have used ρ^n on the left-hand side of the above equations, which results from the standard procedure of subtracting the discretized continuity equation from the momentum equations to yield diagonal dominance in the coefficient matrix [7].

3.2. First momentum corrector step

In the first corrector stage of the overall splitting process, a new velocity field, (u^{**} and v^{**}), is sought, along with a corresponding pressure field, p^*. Towards this, the velocity field obtained from the predictor step (u^* and v^*) is used to write the momentum equations for the first corrector step as follows:

$$\left(A_P + \frac{\rho^n J}{\Delta t}\right) u^{**} = H^n(u^*) + \frac{\rho^n u^n J}{\Delta t} - f_{11}(p_e - p_w)^* - f_{21}(p_n - p_s)^* \tag{22}$$

$$\left(A_P + \frac{\rho^n J}{\Delta t}\right) v^{**} = H^n(v^*) + \frac{\rho^n v^n J}{\Delta t} - f_{12}(p_e - p_w)^* - f_{22}(p_n - p_s)^* \tag{23}$$

In Equations (22)–(23), p^* is yet to be determined. To determine it, the above momentum equations and the continuity equation are used to formulate an equation governing pressure correction, p', which is defined as:

$$p' = p^* - p^n \tag{24}$$

First, we rewrite the momentum equations in terms of p'. Then, subtracting Equations (20) and (21) from Equations (22) and (23), respectively, we obtain:

$$u^{**} = u^* - \frac{1}{A_P^n}[f_{11}(p'_e - p'_w) - f_{21}(p'_n - p'_s)] \tag{25}$$

$$v^{**} = v^* - \frac{1}{A_P^n}[f_{12}(p'_e - p'_w) - f_{22}(p'_n - p'_s)] \tag{26}$$

where we have defined

$$A_P^n = A_P + \frac{\rho^n J}{\Delta t} \tag{27}$$

Equations (25) and (26) will be used to correct u^* and v^* to yield u^{**} and v^{**} once the p' field is obtained. To obtain an equation for p', we first need to formulate the interface contravariant velocities, which in turn require interface Cartesian velocity components. This will be described next.

3.2.1. Formulation of cell-face contravariant velocities. As is well known, the evaluation of cell-face contravariant velocities for non-staggered grids requires special attention. The so-called momentum interpolation approach, which adds a fourth-order pressure dissipation term

to avoid the pressure–velocity decoupling problem, is commonly employed [14, 15]. We will illustrate the formulation of the contravariant velocities, in the context of the present algorithm, along the i direction, i.e. U^*. Due to the finite volume approach being employed here, the contravariant velocities, multiplied by density, represent the mass fluxes at the faces of control volumes. Thus, for example, for a control volume located around the point P (index i) shown in Figure 1(b), the contravariant velocities are required at the east $(i + \frac{1}{2})$ and west $(i - \frac{1}{2})$ faces. Thus, in the algorithm described here, the contravariant velocities are only stored at the control volume interfaces.

In this section, we will describe the estimation of contravariant velocity on the east face, i.e. $U_{i+1/2}$. First, consider the Cartesian velocity components. We can rewrite the discretized momentum equations, Equations (20) and (21) as follows:

$$u_i^* = \frac{G_i^{u*}}{A_P^n} - \frac{f_{11}}{A_P^n}(p_e - p_w)^n \tag{28}$$

$$v_i^* = \frac{G_i^{v*}}{A_P^n} - \frac{f_{12}}{A_P^n}(p_e - p_w)^n \tag{29}$$

where

$$G_i^u = H(u^*) + \frac{\rho^n u^n J}{\Delta t} - f_{21}(p_n - p_s)^n \tag{30}$$

$$G_i^v = H(v^*) + \frac{\rho^n v^n J}{\Delta t} - f_{22}(p_n - p_s)^n \tag{31}$$

Similarly, for evaluating V^* (i.e. the contravariant velocity along the j direction), we write Equations (20) and (21) as:

$$u_j^* = \frac{G_j^{u*}}{A_P^n} - \frac{f_{21}}{A_P^n}(p_n - p_s)^n \tag{32}$$

$$v_j^* = \frac{G_j^{v*}}{A_P^n} - \frac{f_{22}}{A_P^n}(p_n - p_s)^n \tag{33}$$

It should be noted that the pressure cross-derivative terms are included in the definition of the G's in the above equations. Now, let us consider the east face $(i + \frac{1}{2})$ and write $u_{i+1/2}$ and $v_{i+1/2}$ in a form similar to u_i and v_i in Equations (28) and (29):

$$u_{i+1/2}^* = \frac{G_{i+1/2}^{u*}}{A_{P_{i+1/2}}^n} - \frac{f_{11}}{A_{P_{i+1/2}}^n}(p_{i+1} - p_i)^n \tag{34}$$

$$v_{i+1/2}^* = \frac{G_{i+1/2}^{v*}}{A_{P_{i+1/2}}^n} - \frac{f_{12}}{A_{P_{i+1/2}}^n}(p_{i+1} - p_i)^n \tag{35}$$

Note that the first terms on the right-hand side in Equations (34) and (35), respectively, are evaluated using Equations (28)–(29) instead of Equations (30)–(31), as follows:

$$\frac{G^{u*}_{i+1/2}}{A^n_{P_{i+1/2}}} = I_{i+1/2}\frac{G^{u*}_i}{A^n_{P_i}} + (1-I_{i+1/2})\frac{G^{u*}_{i+1}}{A^n_{P_i+1}}$$

$$= I_{i+1/2}u^*_i + (1-I_{i+1/2})u^*_{i+1} + I_{i+1/2}\left(\frac{f_{11}}{A^n_P}\right)_i (p_{i+1/2} - p_{i-1/2})$$

$$+ (1-I_{i+1/2})\left(\frac{f_{11}}{A^n_P}\right)_{i+1} (p_{i+3/2} - p_{i+1/2}) \tag{36}$$

$$\frac{G^{v*}_{i+1/2}}{A^n_{P_{i+1/2}}} = I_{i+1/2}\frac{G^{v*}_i}{A^n_{P_i}} + (1-I_{i+1/2})\frac{G^{v*}_{i+1}}{A^n_{P_{i+1}}}$$

$$= I_{i+1/2}v^*_i + (1-I_{i+1/2})v^*_{i+1} + I_{i+1/2}\left(\frac{f_{12}}{A_P}\right)_i (p_{i+1/2} - p_{i-1/2})$$

$$+ (1-I_{i+1/2})\left(\frac{f_{12}}{A_P}\right)_{i+1} (p_{i+3/2} - p_{i+1/2}) \tag{37}$$

where $I_{i+1/2}$ is the distance-weighted interpolation coefficient for evaluating a value at the $i+\frac{1}{2}$ interface using the nodal values straddling that interface. Now we can formulate the east-interface contravariant velocity as:

$$U^*_{i+1/2} = f_{11}u^*_{i+1/2} + f_{12}v^*_{i+1/2}$$

$$= f_{11}[I_{i+1/2}u^*_i + (1-I_{i+1/2})u^*_{i+1}] + f_{12}[I_{i+1/2}v^*_i + (1-I_{i+1/2})v^*_{i+1}] - F_{i+1/2}(p_{i+1} - p_i)^n$$

$$+(f^2_{11} + f^2_{12})\left[\frac{I_{i+1/2}}{A_{P_i}}(p^n_{i+1/2} - p^n_{i-1/2}) + \frac{(1-I_{i+1/2})}{A_{P_{i+1}}}(p^n_{i+3/2} - p^n_{i+1/2})\right] \tag{38}$$

where the coefficient $F_{i+1/2}$ is defined as

$$F_{i+1/2} = \left(\frac{f^2_{11} + f^2_{12}}{A^n_P}\right)_{i+1/2} = (f^2_{11} + f^2_{12}).\left(\frac{I_{i+1/2}}{A_{P_i}} + \frac{(1-I_{i+1/2})}{A_{P_{i+1}}}\right) \tag{39}$$

3.2.2. Formulation of equation governing p'. The next step is to formulate the equation for p'. Towards this end, let us first write the momentum equations for the $i+\frac{1}{2}$ face:

$$\textit{predictor}: \quad u^*_{i+1/2} = \frac{G^{u*}_{i+1/2}}{A^n_{P_{i+1/2}}} - \frac{f_{11}}{A^n_{P_{i+1/2}}}(p_{i+1} - p_i)^n \tag{40}$$

$$\textit{corrector}: \quad u^{**}_{i+1/2} = \frac{G^{u*}_{i+1/2}}{A^n_{P_{i+1/2}}} - \frac{f_{11}}{A^n_{P_{i+1/2}}}(p_{i+1} - p_i)^* \tag{41}$$

Now, subtracting Equation (40) from Equation (41), and doing the same for $v_{i+1/2}$, we get,

$$u^{**}_{i+1/2} = u^{*}_{i+1/2} - \frac{f_{11}}{A^{n}_{P_{i+1/2}}}(p'_{i+1} - p'_i) \tag{42}$$

$$v^{**}_{i+1/2} = v^{*}_{i+1/2} - \frac{f_{12}}{A^{n}_{P_{i+1/2}}}(p'_{i+1} - p'_i) \tag{43}$$

Note that we have absorbed the pressure cross-derivative terms in the definition of $G^{u*}_{i+1/2}$ (and also in $G^{v*}_{i+1/2}$) and thus they do not appear in the expressions for $u^{**}_{i+1/2}$ and $v^{**}_{i+1/2}$. This can lead to the pressure correction equation being unstable on skewed grids for which these cross-derivative terms may be significant, as will be discussed later.

Next, we multiply Equations (42) and (43) with f_{11} and f_{12}, respectively, and then add them to yield $U^{**}_{i+1/2}$, to get the following:

$$\begin{aligned} U^{**}_{i+1/2} &= U^{*}_{i+1/2} + U'_{i+1/2} \\ &= U^{*}_{i+1/2} - F_{i+1/2}(p'_{i+1} - p'_i) \end{aligned} \tag{44}$$

Similarly, at the $i - \frac{1}{2}$ interface, we can obtain $U^{**}_{i-1/2}$:

$$\begin{aligned} U^{**}_{i-1/2} &= U^{*}_{i-1/2} + U'_{i-1/2} \\ &= U^{*}_{i-1/2} - F_{i-1/2}(p'_i - p'_{i-1}) \end{aligned} \tag{45}$$

Now, we invoke the continuity equation for this stage. For illustration, we will temporarily use the continuity equation for a one-dimensional situation which involves mass fluxes at east and west faces only. It can be written as:

$$\frac{J}{\Delta t}(\rho^* - \rho^n) + [(\rho^* U^{**})_{i+1/2} - (\rho^* U^{**})_{i-1/2}] = 0 \tag{46}$$

The density ρ^* (which is an unknown so far) is expressed as

$$\rho^* = \rho^n + \rho' \tag{47}$$

Using the above equations, the continuity equation, Equation (46), can be written as follows:

$$\begin{aligned} &\frac{J}{\Delta t}(\rho^* - \rho^n) + (\rho^n + \rho')_{i+1/2}.U^{*}_{i+1/2} - (\rho^n + \rho')_{i-1/2}.U^{*}_{i-1/2} \\ &\quad -[(\rho^n + \rho')_{i+1/2}.F_{i+1/2}.(p'_{i+1} - p'_i) - (\rho^n + \rho')_{i-1/2}.F_{i-1/2}.(p'_i - p'_{i-1})] = 0 \end{aligned} \tag{48}$$

In Equation (48), the terms involving the product of ρ' and p' are neglected. Thus, the equation governing p' can be written as:

$$\begin{aligned} &\frac{\rho'_i}{\Delta t} J - (\rho^n F)_{i+1/2}.(p'_{i+1} - p'_i) + (\rho^n F)_{i-1/2}.(p'_i - p'_{i-1}) + \rho'_{i+1/2} U^{*}_{i+1/2} - \rho'_{i-1/2} U^{*}_{i-1/2} \\ &\quad = -[(\rho^n U^*)_{i+1/2} - (\rho^n U^*)_{i-1/2}] \end{aligned} \tag{49}$$

where the term involving density difference is treated implicitly by invoking the equation of state:

$$\rho' = \frac{p'}{RT} \tag{50}$$

Moreover, when evaluating $\rho' U^*$ at the control volume interface, an upwind scheme is employed, so that at the east $(i + \frac{1}{2})$ interface, for example, we have

$$\rho'_{i+1/2} U^*_{i+1/2} = \frac{p'_{i+1/2}}{RT} U^*_{i+1/2} = \begin{cases} \dfrac{p'_i}{RT} U^*_{i+1/2} & \text{if } U^*_{i+1/2} > 0 \\ \dfrac{p'_{i+1}}{RT} U^*_{i+1/2} & \text{if } U^*_{i+1/2} < 0 \end{cases} \tag{51}$$

The mass fluxes at the north and south faces are treated in a similar manner. The final discretized form of the two-dimensional equation governing the p' field can be written as:

$$A_P p'_P = A_E p'_E + A_W p'_W + A_N p'_N + A_S p'_S + b \tag{52}$$

where

$$A_E = (\rho^n F)_{i+1/2} + \frac{1}{RT} \max(-U_{i+1/2}, 0) \tag{53}$$

$$A_W = (\rho^n F)_{i-1/2} + \frac{1}{RT} \max(U_{i-1/2}, 0) \tag{54}$$

$$A_P = (\rho^n F)_{i+1/2} + (\rho^n F)_{i-1/2} + \frac{1}{RT} \max(U_{i+1/2}, 0) + \frac{1}{RT} \max(-U_{i-1/2}, 0) + \frac{J}{RT\Delta t} \tag{55}$$

$$b = -[(\rho^n U^*)_{i+1/2} - (\rho^n U^*)_{i-1/2}] \tag{56}$$

and the coefficients A_N and A_S have a form similar to A_E and A_W along with the corresponding contributions to the coefficient A_P.

3.2.3. Correction of pressure and velocity. Following the computation of the p' field, pressure field is corrected by:

$$p^* = p^n + p' \tag{57}$$

The nodal Cartesian velocities are corrected by using Equations (25) and (26). Contravariant velocities are corrected by using Equation (44) for $U^{**}_{i+1/2}$ and a similar equation for $V^{**}_{j+1/2}$.

3.3. Energy predictor step

The energy equation is solved next implicitly:

$$\left(B_P + \frac{\rho^n J}{\Delta t}\right) h_0^* = G^n(h_0^*) + \frac{\rho^n h_0^n J}{\Delta t} + \frac{(p^* - p^n)J}{\Delta t} \tag{58}$$

where h_0 is the total enthalpy. The coefficient B_P is analogous to the coefficient A_P of the momentum equations and the operator G^n is analogous to the operator H^n. Temperature (T^*) is extracted from h_0^* along with u^{**} and v^{**} using Equation (5).

3.4. Second momentum corrector step

The u-momentum equation for the second corrector stage is written as:

$$\left(A_P + \frac{\rho^n J}{\Delta t}\right) u^{***} = H^*(u^{**}) + \frac{\rho^n u^n J}{\Delta t} - f_{11}(p_e - p_w)^{**} - f_{21}(p_n - p_s)^{**} \tag{59}$$

where H^* is the updated operator using u^{**}. Note that, similar to Equation (22), we have used ρ^n in the term on the left-hand side of Equation (59). The pressure field p^{**} is yet to be determined. Let us define the pressure correction at the second stage as

$$p'' = p^{**} - p^* \tag{60}$$

Subtracting the first corrector equation, Equation (22), from Equation (59), we get,

$$A_P^n(u^{***} - u^{**}) = [H^*(u^{**}) - H^n(u^*)] - f_{11}(p''_e - p''_w) - f_{21}(p''_n - p''_s) \tag{61}$$

where A_P^n is given by Equation (27). From this we obtain

$$u^{***} = \frac{1}{A_P^n}\{u^{**} + [H^*(u^{**}) - H^n(u^*)] - f_{11}(p''_e - p''_w) - f_{21}(p''_n - p''_s)\} \tag{62}$$

Next, we derive the contravariant velocities at the control volume interfaces for the second corrector stage. First, let us rewrite Equations (59) and (22) as follows:

$$\textit{Second Corrector}: \quad u^{***} = \frac{1}{A_P^n}[\hat{u}^* - f_{11}(p_e - p_w)^{**}] \tag{63}$$

$$\textit{First Corrector}: \quad u^{**} = \frac{1}{A_P^n}[\hat{u}^n - f_{11}(p_e - p_w)^*] \tag{64}$$

where

$$\hat{u}^* = H^*(u^{**}) + \frac{\rho^n u^n J}{\Delta t} - f_{21}(p_n - p_s)^* \tag{65}$$

$$\hat{u}^n = H^n(u^*) + \frac{\rho^n u^n J}{\Delta t} - f_{21}(p_n - p_s)^n \tag{66}$$

Now, let us rewrite Equations (63) and (64) for the interface $i + \frac{1}{2}$ as follows:

$$\textit{Second Corrector}: \quad u^{***}_{i+1/2} = \frac{1}{A_P^n}[\hat{u}^*_{i+1/2} - f_{11}(p_{i+1} - p_i)^{**}] \tag{67}$$

$$\textit{First Corrector}: \quad u^{**}_{i+1/2} = \frac{1}{A_P^n}[\hat{u}^n_{i+1/2} - f_{11}(p_{i+1} - p_i)^*] \tag{68}$$

Subtracting Equation (68) from Equation (67) and rearranging, we get,

$$u^{***}_{i+1/2} = u^{**}_{i+1/2} + (\hat{u}^*_{i+1/2} - \hat{u}^n_{i+/12}) - f_{11}(p''_{i+1} - p''_i) \tag{69}$$

Similarly, we can write the expression for $v^{***}_{i+1/2}$ as

$$v^{***}_{i+1/2} = v^{**}_{i+1/2} + (\hat{v}^{*}_{i+1/2} - \hat{v}^{n}_{i+/12}) - f_{12}(p''_{i+1} - p''_{i}) \tag{70}$$

Then, multiplying Equations (69) and (70) with f_{11} and f_{12}, respectively, the contravariant velocity for the second corrector stage at the $i + \frac{1}{2}$ interface can be obtained as:

$$U^{***}_{i+1/2} = U^{**}_{i+1/2} + \hat{U}_{i+1/2} - F_{i+1/2}(p''_{i+1} - p''_{i}) \tag{71}$$

where

$$\hat{U}_{i+1/2} = \left(\frac{f_{11}}{A^{n}_{P}}\right)_{i+1/2}(\hat{u}^{*}_{i+1/2} - \hat{u}^{n}_{i+1/2}) + \left(\frac{f_{12}}{A^{n}_{P}}\right)_{i+1/2}(\hat{v}^{*}_{i+1/2} - \hat{v}^{n}_{i+1/2}) \tag{72}$$

and $F_{i+1/2}$ is defined in Equation (39). The contravariant velocity at the $i - \frac{1}{2}$ interface can be written in a similar manner.

Next, we invoke the continuity equation for the second corrector stage (again, we write it for a one-dimensional situation for illustration):

$$\frac{J}{\Delta t}(\rho^{**} - \rho^{n}) + [\rho^{**}U^{***}]^{i+12}_{i-1/2} = 0 \tag{73}$$

where $[\cdot]^{i+1/2}_{i-1/2} = (\cdot)_{i+1/2} - (\cdot)_{i-1/2}$. The density for the second stage, ρ^{**}, is expressed as

$$\rho^{**} = \rho^{*} + \rho'' \tag{74}$$

Using the above equations, the continuity equation, Equation (73), can be written as:

$$\begin{aligned}\frac{J}{\Delta t}(\rho^{**} - \rho^{n}) + [(\rho^{*} + \rho'')U^{**}]^{i+1/2}_{i-1/2} + [(\rho^{*} + \rho'')\hat{U}]^{i+1/2}_{i-1/2} \\ - (\rho^{*} + \rho'')_{i+1/2}F_{i+1/2}(p''_{i+1} - p''_{i}) + (\rho^{*} + \rho'')_{i-1/2}F_{i-1/2}(p''_{i} - p''_{i-1}) = 0\end{aligned} \tag{75}$$

In Equation (75), the terms involving the product of ρ'' and p'' are neglected. Thus, the equation governing p'' can be written as:

$$\begin{aligned}\frac{(\rho''_{i} + \rho'_{i})}{\Delta t}J - (\rho^{*}F)_{i+1/2}.(p''_{i+1} - p''_{i}) + (\rho^{*}F)_{i-1/2}.(p''_{i} - p''_{i-1}) + \rho''_{i+1/2}(U^{**}_{i+1/2} + \hat{U}_{i+1/2}) \\ -\rho''_{i-1/2}(U^{**}_{i-1/2} + \hat{U}_{i-1/2}) = -[\rho^{*}(U^{**} + \hat{U})]^{i+1/2}_{i-1/2}\end{aligned} \tag{76}$$

where the term involving density difference is treated implicitly by invoking the equation of state:

$$\rho'' = \frac{p''}{RT} \tag{77}$$

Moreover, the $\rho''U^{**}$ term at control volume interfaces is evaluated using an upwind scheme, similar to Equation (51).

The final discretized form of the two-dimensional equation governing the p'' field can be written as:

$$A_P p''_P = A_E p''_E + A_W p''_W + A_N p''_N + A_S p''_S + b \tag{78}$$

where

$$A_E = (\rho^* F)_{i+1/2} + \frac{1}{RT} \max[-(U^*_{i+1/2} + \hat{U}_{i+1/2}), 0] \tag{79}$$

$$A_W = (\rho^* F)_{i-1/2} + \frac{1}{RT} \max[(U^*_{i-1/2} + \hat{U}_{i-1/2}), 0] \tag{80}$$

$$A_P = (\rho^* F)_{i+1/2} + (\rho^* F)_{i-1/2} + \frac{1}{RT} \max[(U^*_{i+1/2} + \hat{U}_{i+1/2}), 0] + \frac{1}{RT} \max[-(U^*_{i-1/2} + \hat{U}_{i-1/2}), 0] + \frac{J}{RT\Delta t} \tag{81}$$

$$b = -[(\rho^* U^{**})_{i+1/2} - (\rho^* U^{**})_{i-1/2}] - [(\rho^* \hat{U})_{i+1/2} - (\rho^* \hat{U})_{i-1/2}] + \frac{\rho'}{\Delta t} J \tag{82}$$

and the coefficients A_N and A_S have a form similar to A_E and A_W along with the corresponding contributions to the coefficient A_P.

Following the computation of the p'' field, pressure field is corrected by:

$$p^{**} = p^* + p'' \tag{83}$$

Cartesian velocities are corrected by using Equation (62) for u^{***} and a similar equation for v^{***}. Contravariant velocities are corrected by using Equation (71) for $U^{***}_{i+1/2}$ and a similar equation for $V^{***}_{j+1/2}$.

3.5. Energy corrector step

Similar to the original PISO method described by Issa [1], this step involves the update of total enthalpy using the following explicit equation:

$$\left(B_P + \frac{\rho^{**} J}{\Delta t}\right) h_0^{**} = G^n(h_0^*) + \frac{\rho^n h_0^n J}{\Delta t} + \frac{(p^{**} - p^n)J}{\Delta t} + S \tag{84}$$

Subsequently, a new temperature field (T^{**}) is obtained from h_0^{**} along with u^{**} and v^{**}.

3.6. Summary of the basic algorithm (Method 0)

The algorithm described in the preceding section can be summarized, for each new time step, as a series of steps (for later reference, this is labelled Method 0) as follows:

1. Momentum Predictor
   - Compute u^* and v^* using Equations (20) and (21)
2. Velocity–Pressure–Density Corrector A
   - Compute contravariant velocities U^* and V^* using Equation (38)

- Compute p' using Equation (52)
- Update velocities using p': u^{**} and v^{**} from Equations (25)–(26); U^{**} and V^{**} from Equation (44)
- Update pressure and density: p^* from Equation (57) and ρ^* from equation of state

3. Energy Predictor
4. Velocity–Pressure–Density Corrector B
   - Compute $\hat{U}$ and $\hat{V}$ from Equation (72)
   - Compute p'' using Equation (78)
   - Using p'', update Cartesian velocities (u^{***} and v^{***}) using Equation (62) and contravariant velocities (U^{***} and V^{***}) using Equation (71)
   - Update pressure p^{**} from Equation (83)
   - ρ^{**} from equation of state
5. Energy Corrector

The variables u^{***}, v^{***}, U^{***}, V^{***}, p^{**}, ρ^{**}, h_0^{**} and T^{**} are considered the values at the new time level $(n+1)$ and we proceed to step 1 for the next time level. Note that if turbulence model equations are also being solved, the predictor and corrector steps for those equations immediately follow the energy predictor and corrector steps, respectively.

3.7. Enhanced algorithm for highly skewed grids

As will be demonstrated in Section 4, the basic algorithm summarized above (labelled as Method 0) is unstable for skewed meshes. This can be attributed to the approximation used in the derivation of the pressure correction where the cross-derivative p' terms are neglected. To achieve stability on highly non-orthogonal (or skewed) meshes, the non-orthogonal terms need to be accounted for in the formulation of the pressure correction equation. One of the methods involves the treatment of these non-orthogonal terms using a deferred correction approach. Such an approach is developed in the context of the present algorithm for body-fitted curvilinear grids. Another method, based on additional corrector steps, is also proposed to handle the non-orthogonal terms in the p' equation and is shown to be more robust. Both of these methods are presented next.

3.7.1. Deferred correction approach (Method 1). This treatment, labelled Method 1, is similar in spirit to the deferred correction approach discussed by Ferziger and Peric [19]; however, the implementation in the present algorithm is different. First, we rewrite the interface Cartesian velocity components, $u_{i+1/2}$ and $v_{i+1/2}$, by including the cross-derivative pressure terms, i.e. we replace Equations (34) and (35) with the following:

$$u^*_{i+1/2} = \frac{G'^{u*}_{i+1/2}}{A^n_{P_{i+1/2}}} - \frac{f_{11}}{A^n_{P_{i+1/2}}}(p_{i+1} - p_i)^n - \frac{f_{21}}{A^n_{P_{i+1/2}}}(p_{\mathrm{ne}} - p_{\mathrm{se}})^n \tag{85}$$

$$v^*_{i+1/2} = \frac{G'^{v*}_{i+1/2}}{A^n_{P_{i+1/2}}} - \frac{f_{12}}{A^n_{P_{i+1/2}}}(p_{i+1} - p_i)^n - \frac{f_{22}}{A^n_{P_{i+1/2}}}(p_{\mathrm{ne}} - p_{\mathrm{se}})^n \tag{86}$$

where the subscripts 'ne' and 'se' denote the north-east and the south-east corners of the control volume, as shown in Figure 1(b). Note that the terms $G'^{u*}_{i+1/2}$ and $G'^{v*}_{i+1/2}$ in Equations (85)

and (86) are evaluated similar to $G^{u*}_{i+1/2}$ and $G^{v*}_{i+1/2}$ in Equations (36) and (37), the difference being the extra terms resulting from the pressure cross-derivative terms (the expressions are not included here for brevity).

Next, we multiply $u^*_{i+1/2}$ and $v^*_{i+1/2}$ with f_{11} and f_{12}, respectively, and then add them to yield $U^*_{i+1/2}$. Following the development in Section 3.2.2, we then obtain $U'_{i+1/2}$:

$$\begin{aligned} U'_{i+1/2} &= U^{**}_{i+1/2} - U^*_{i+1/2} \\ &= -F_{i+1/2}(p'_{i+1} - p'_i) - \frac{q_{12}}{A^n_{P_{i+1/2}}}(p'_{\mathrm{ne}} - p'_{\mathrm{se}}) \end{aligned} \tag{87}$$

where q_{12} is defined in Equation (11). Similarly, we can obtain an expression for $U'_{i-1/2}$. Thus, the pressure correction equation has the same form as Equation (49) with the additional contribution from the pressure cross-derivative terms from Equation (87). These extra terms can be handled in an iterative manner. In the first step, we neglect these terms which yields the pressure correction equation, Equation (52), with the coefficients and the source terms given by Equations (53)–(56). In the second step, the error made in the first step due to the omission of the cross-terms is corrected by adding another correction, as follows:

$$\begin{aligned} U'_{i+1/2} + U'^{(2)}_{i+1/2} = &-F_{i+1/2}(p'_{i+1} - p'_i) - \frac{q_{12}}{A^n_{P_{i+1/2}}}(p'_{\mathrm{ne}} - p'_{\mathrm{se}}) \\ &- F_{i+1/2}(p'^{(2)}_{i+1} - p'^{(2)}_i) - \frac{q_{12}}{A^n_{P_{i+1/2}}}(p'^{(2)}_{\mathrm{ne}} - p'^{(2)}_{\mathrm{se}}) \end{aligned} \tag{88}$$

Then, the non-orthogonality terms are neglected in the second correction $p'^{(2)}$ but are retained in the first correction p', which leads to the following expression for the second mass flux correction:

$$U'^{(2)}_{i+1/2} = -F_{i+1/2}(p'^{(2)}_{i+1} - p'^{(2)}_i) - \frac{q_{12}}{A^n_{P_{i+1/2}}}(p'_{\mathrm{ne}} - p'_{\mathrm{se}}) \tag{89}$$

The last term in Equation (89) can be explicitly evaluated since p' has been previously computed. The equation for the second correction $p'^{(2)}$ has the same coefficients as Equations (53)–(56) with the additional non-orthogonal terms contributing to the source term b in Equation (56).

The same procedure can be employed for the pressure correction equation in the second corrector step (Corrector Step B) by solving an equation for p'' followed by an equation for $p''^{(2)}$. Such an approach was tested in the context of the present algorithm. Though it works well, it was found to significantly increase the computational expense while solving the pressure correction equation.

3.7.2. New approach (Method 2). An alternative approach, labelled Method 2, is proposed in the present study. This approach is based on the observation that the error in the pressure correction equation due to the omission of pressure cross-derivative terms emanates from the approximation made in evaluating the interface Cartesian velocities, namely, Equations (42) and (43), which resulted from the fact that the cross-derivative pressure gradient terms in the nodal Cartesian velocity expressions given by Equations (28) and (29) were absorbed in the

terms G^{u*} and G^{v*}, respectively. This carries over to the evaluation of the corrections to the interface contravariant velocities in Equations (44) and (45). In Method 2, steps 1–5 outlined for Method 0 are still used. However, while using the pressure correction p' computed in Step 2 (Corrector A) to correct the nodal Cartesian velocities, the cross-derivative terms are retained, as shown below:

$$u'_i = -\frac{f_{11}}{A_P^n}(p'_e - p'_w) - \frac{f_{21}}{A_P^n}(p'_n - p'_s) \tag{90}$$

$$v'_i = -\frac{f_{12}}{A_P^n}(p'_e - p'_w) - \frac{f_{22}}{A_P^n}(p'_n - p'_s) \tag{91}$$

The corrections to the interface contravariant velocity components are also treated in a similar manner. Once the corrected Cartesian and contravariant velocity fields are obtained, the pressure correction equation is solved again by using the updated mass fluxes. In this manner, the correction for non-orthogonality is included in the pressure correction in an indirect manner. The above procedure is repeated for the second corrector step (Velocity–Pressure–Density Corrector B).

To summarize, Method 2 involves two additional steps compared to Method 0, in addition to the modified form of the correction procedure for the nodal Cartesian and interfacial contravariant velocity components. Steps 1–5 in Method 2 are the same as in Method 0 above. The two additional steps are as shown below:

6. Velocity–Pressure–Density Corrector A
7. Velocity–Pressure–Density Corrector B

This approach has been found to be relatively less expensive and quite robust. In the present algorithm, the most time-consuming element is the solution of pressure correction equation. In Method 2, the p' equation needs to be solved two times per corrector step, i.e. typically a total of four times per time step. In Method 1, the p' equation is typically solved two or three times for each corrector step. Thus, Method 1 can be more expensive for grids with a high degree of skewness. This will be demonstrated in the results section for computations employing grids of varying degrees of skewness.

3.8. Solution of discretized equations for multiblock grids

In traditional multiblock methods, the discretized equations are solved separately in each block followed by an exchange of information among blocks during the iterative solution procedure. For steady solvers (or for unsteady solvers used to solve steady flows), the discretized equations need not be converged down to machine accuracy at each iteration level since each outer iteration merely serves as a small step towards the final steady solution. For unsteady problems, on the other hand, the equations must be solved down to machine accuracy at each time step. In the context of the operator splitting algorithm presented here, a decoupled block-by-block solution procedure would necessitate inter-block iterations within each time step. To avoid this, in the present method, the equations are assembled for all the blocks at the same time and then solved on a point-by-point basis. If the Gauss–Seidel method is used to solve the discretized equations, convergence is very slow, especially for the pressure correction equation. To accelerate the solution of the linearized systems of equations, an

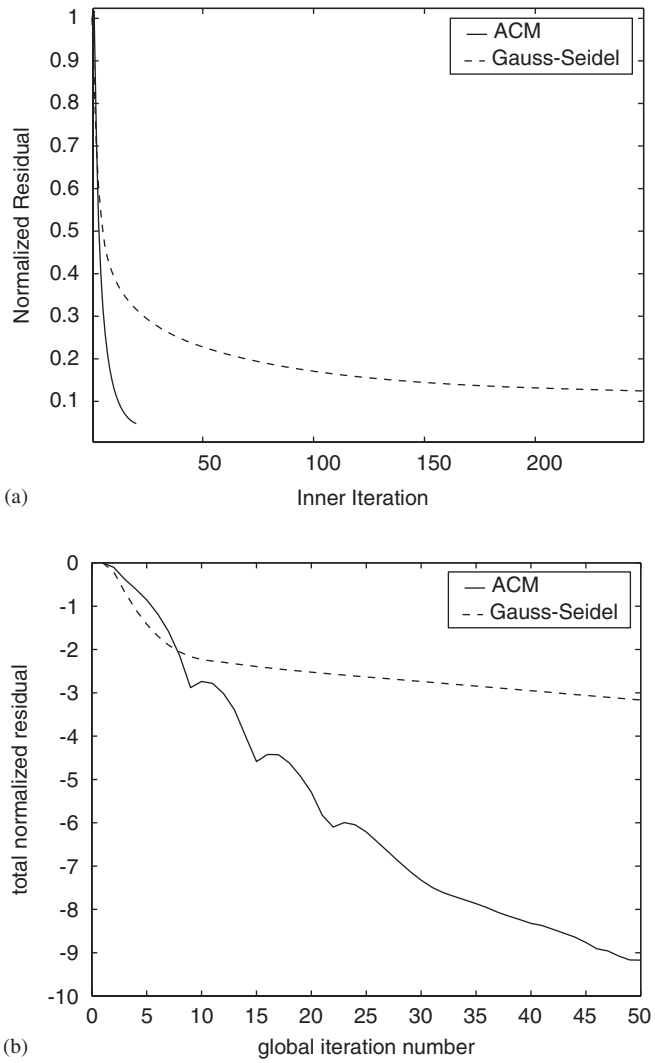


Figure 2. Comparison of global convergence with Gauss–Seidel and additive correction multigrid (ACM) methods. The geometry used is the 60° skewed cavity shown in Figure 3, using an 81×81 grid: (a) Initial-to-final residual ratio versus inner iterations for one outer (global) iteration; and (b) normalized total residual versus outer iterations.

algebraic multigrid method, based on the additive correction multigrid (ACM) method proposed by Hutchinson and Raithby [20] is employed. This allows for a convergence of the linearized equations down to machine accuracy in significantly smaller number of iterations. This is evident from the convergence history for a typical computation, shown in Figure 2. The case used is laminar flow in a skewed lid-driven cavity on an 81×81 grid, the results of which are presented in the next section. During the solution of the linearized matrix equation

(which we call the 'inner' iteration step), the Gauss–Seidel method is not able to drive the residuals down to machine accuracy. This is evident in Figure 2(a) which shows a plot of the initial-to-final residual ratio versus number of inner iterations during a typical global (outer) iteration. Consequently, the Gauss–Seidel method takes a larger number of outer iterations to converge. The ACM method, on the other hand, drives the residuals down rapidly to machine accuracy, i.e. the initial-to-final residual ratio approaches zero in a small number of inner iterations, as seen in Figure 2(b). This significantly accelerates the overall (outer) convergence of the solver, as seen in Figure 2(b).

4. RESULTS

In this section, we demonstrate the efficacy of the non-iterative operator-splitting method developed in this paper for all Mach number regimes. Results are presented for (a) incompressible flows in skewed lid-driven cavities, (b) steady and unsteady incompressible flows past a circular cylinder, (c) subsonic, transonic and supersonic flows past a circular bump, and (d) supersonic and hypersonic flows past a wedge. Finally, to demonstrate the method's applicability to all-speed flows in complex geometries, an unsteady compressible flow in a centrifugal compressor is also presented.

For time-accurate resolution of unsteady computations, the specified time step size should be of the same order of magnitude as the smallest characteristic time scale, δt_c, for convection and diffusion, which is given by

$$\delta t_c = \min\left(\frac{\delta L}{u^*}, \frac{\rho \delta L^2}{\Gamma}\right) \tag{92}$$

where δL is the local mesh size, u^* is the local characteristic speed, ρ is the fluid density and Γ is the diffusion coefficient. This heuristic criterion can also be stated in terms of dimensionless parameters based on the characteristic time scales and the time step employed for the computations. The parameter based on the convection time scale is the *CFL* (or Courant) number, which is defined, for incompressible flows, as

$$CFL = \frac{u^* \Delta t}{\delta L} \tag{93}$$

and for compressible flows as

$$CFL = \frac{(u^* + a)\Delta t}{\delta L} \tag{94}$$

where a is the speed of sound. For body-fitted grids using curvilinear co-ordinates, it is appropriate to use the local co-ordinate directions to estimate the *CFL* number. It can be written, for incompressible flows, as

$$CFL = \frac{\Delta t}{J} \max(U, V, W) \tag{95}$$

where J is the Jacobian defined in Equation (12) (which represents the cell volume), and U, V and W are the contravariant velocities. For compressible flows, the above defini-

tion gets modified by the inclusion of the local speed of sound, similar to Equation (94). The corresponding parameter for the diffusion time scale is given by,

$$d = \frac{\Gamma \Delta t}{\rho \delta L^2} \tag{96}$$

which can be written for curvilinear body-fitted grids as follows:

$$d = \frac{\Gamma \Delta t}{\rho J} \max\left(\frac{q_{11}}{J}, \frac{q_{22}}{J}, \frac{q_{33}}{J}\right) \tag{97}$$

Now the condition stated in Equation (92) can be re-stated in terms of the parameters *CFL* and d: for time-accurate computations, the larger of these two parameters should be less than one.

4.1. Lid-driven flows in skewed cavities

To demonstrate the robustness of the present method for skewed, non-orthogonal grids, lid-driven cavity flows with angles of 30° and 60° between the side wall of the cavity and the horizontal direction are solved. These geometries have been used by previous researchers, e.g. Reference [21]. Uniformly spaced grids with 21×21 nodes are used with a Reynolds Number of 100. The grids and the solutions obtained are shown in Figure 3. Method 0 is stable for the 60° cavity case but is not stable for the 30° cavity geometry for which the grid skewness is higher. Stable solutions can be obtained for the 30° cavity geometry using either Method 1 or 2. To test whether the stability of the method is dependent on grid density, the 30° cavity case is also solved using an 81×81 grid. The same behaviour is observed, i.e. Method 0 is unstable while Methods 1 and 2 are stable. However, it is found that as the grid skewness increases, Method 2 converges in fewer number of outer iterations than Method 1. For all the subsequent cases shown in the following, Method 2 is used as the default method.

4.2. Evolution of steady incompressible flow past a cylinder

The transient development to steady state of laminar, incompressible flow past an impulsively started circular cylinder in a quiescent fluid is used as the next test case to assess the present algorithm. The Reynolds number of the flow, based on the cylinder diameter is $Re = 40$. A single-block C-type grid consisting of 201×121 nodes, shown in Figure 4(a), is used to discretize the domain. A second-order upwind scheme is used for the convection terms in the momentum equation. Experimental data obtained by Coutanceau and Bouard [22] indicates that the flow develops to steady state after elapsed non-dimensional time equal to approximately 15. Since the (time-accurate) transient development of the flow is desired, a time step size equal to 0.02 is employed, which corresponds to a maximum *CFL* number of approximately 0.45 in the flowfield. The main characteristic of the flow field is a pair of counter-rotating recirculation regions behind the cylinder. The length of these recirculation regions grows from zero to a constant value when steady state is reached. Experimental data is used to compare the evolution of the recirculation region with time, as shown in Figure 4. The computed results match well with the measured data. To ascertain the range of *CFL* numbers for which the method produces stable solutions for steady flows, several computations were conducted using successively larger time steps. For this particular flow, the method was found to be stable for time step sizes up to 0.5, which is 25 times larger than the time

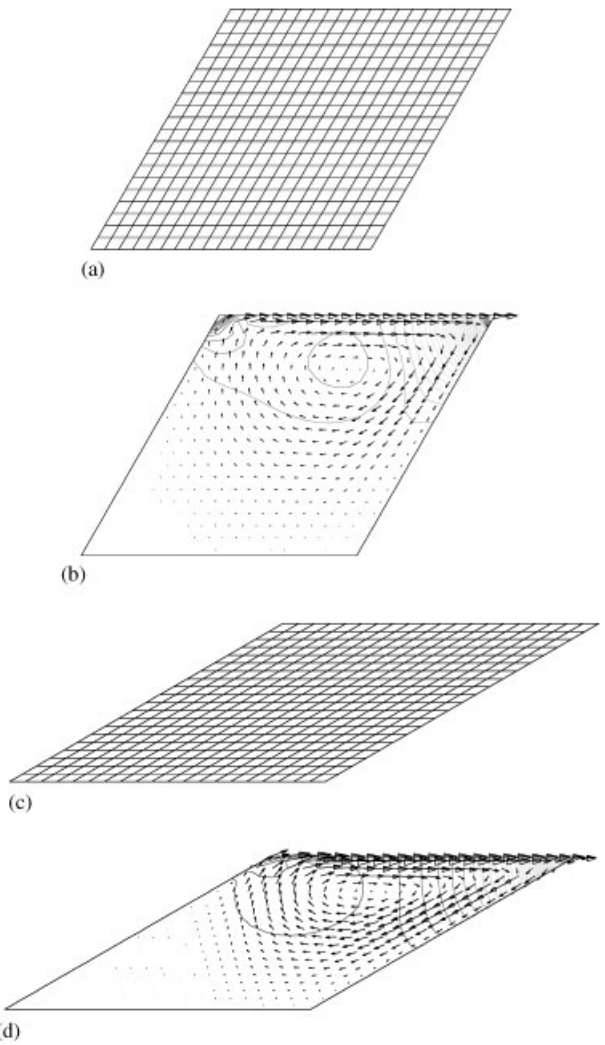


Figure 3. Illustration of the method for skewed grids (cavities with 30° and 60° skewness). Solution for the 30° cavity case can only be obtained with the enhanced algorithm (Method 2): (a) Grid for cavity with 60° skewness; (b) velocity vectors and pressure contours for cavity with 60° skewness; (c) grid for cavity with 30° skewness; and (d) velocity vectors and pressure contours for cavity with 30° skewness.

step used to obtain the time-accurate development of the present flow to steady state. The contour plot of *CFL* number for $\Delta t = 0.5$ is shown in Figure 4(d). It can be observed that the maximum *CFL* number for which the present method is stable for this flow is approximately 12. This demonstrates that the present method can be suitably applied to steady flows as well without the inconvenience of specifying relaxation factors in an arbitrary manner as is required for algorithms based on the iterative SIMPLE method.

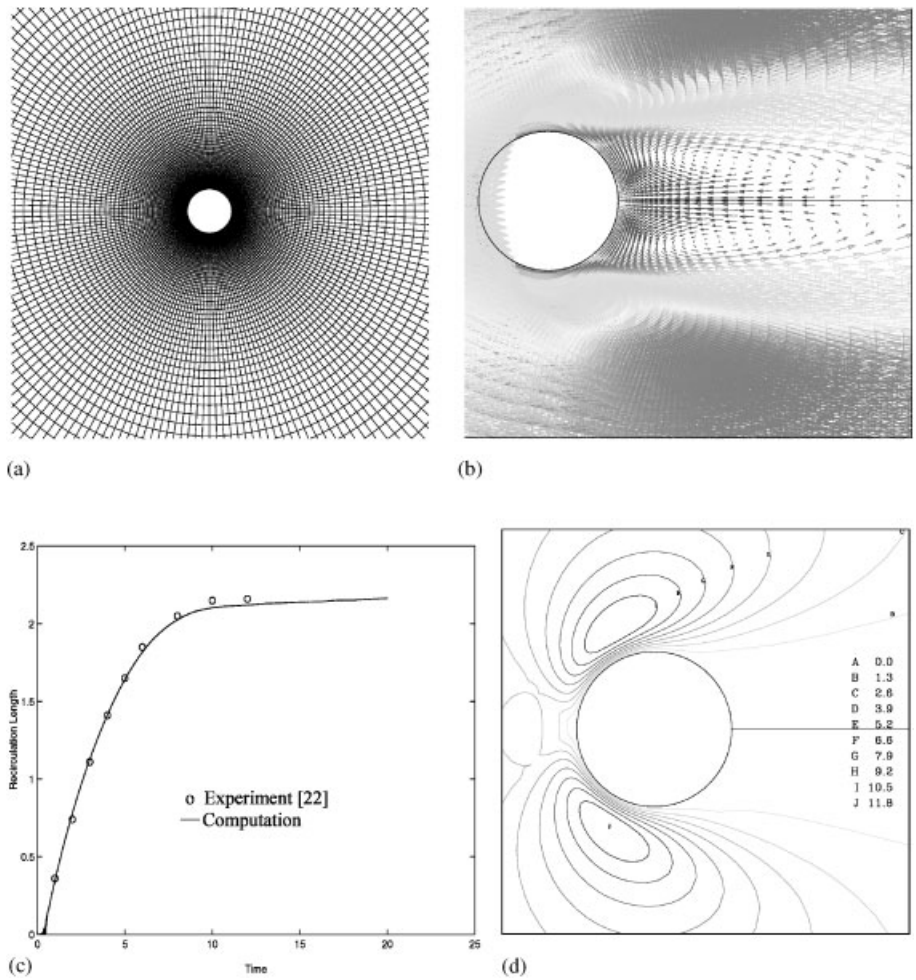
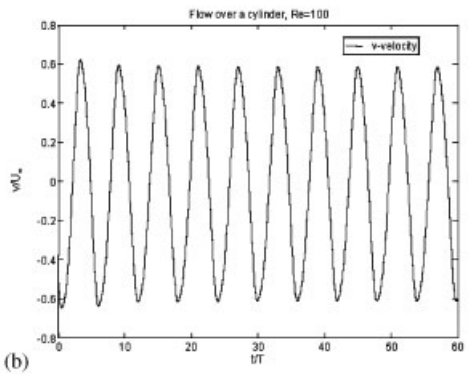


Figure 4. Transient development of laminar flow ($Re = 40$) past a circular cylinder to steady state: (a) 201×121 grid; (b) velocity vectors showing the recirculation region; (c) recirculation length versus time using $\Delta t = 0.02$; and (d) *CFL* number contours using $\Delta t = 0.5$.

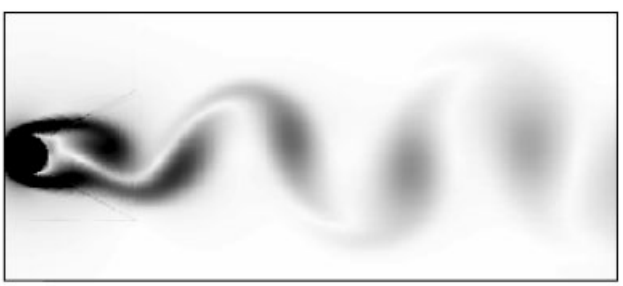
4.3. Unsteady incompressible flow past a circular cylinder

Although the history of the development of the recirculation regions behind the cylinder in the previous test case provides a validation of the algorithm with experimental data, the next case allows the validation for a truly unsteady flow. This case involves the flow past a circular cylinder at $Re = 100$ at which the flow is laminar and exhibits the well-known vortex shedding phenomenon. To demonstrate the performance of the algorithm for a multiblock grid, the domain is discretized using eight blocks, as shown in Figure 5(a). The grid consists of 100 000 nodes. Two time step sizes were used, namely, 0.002 and 0.001, which correspond to maximum *CFL* numbers of approximately 0.26 and 0.13, respectively. A perturbation is

(a)



(b)



(c)

Figure 5. Multiblock computation of unsteady flow past a circular cylinder ($Re = 100$) using $\Delta t = 0.002$ (max $CFL = 0.26$): (a) Computational domain showing the multiblock grid consisting of 8 blocks; (b) periodic oscillation of the cross-stream (v) component of velocity. Strouhal number: computed = 0.164; experiment [23] = 0.164; (c) spanwise vorticity: snapshot of the vortex shedding and the von Karman vortex street.

given to the initial flow field to quickly reach the periodic oscillatory flow pattern. On the left boundary, the steady-state inlet condition with constant horizontal velocity is adopted. On the right, the outlet condition employing zero gradient extrapolation of the velocity and specified pressure is invoked. Williamson [23] has performed experiments for this geometry with an open domain. Accordingly, on the top and bottom boundaries an outlet boundary condition is assigned which requires specification of pressure.

Figure 5(b) shows the periodic behaviour of the cross-stream (v) component of the velocity obtained by probing at a single point downstream of the cylinder. The v component of the

velocity alternates due to the vortex shedding behind the cylinder. Vortices form on the surface of the cylinder and shed downstream forming the well known von Karman vortex street as shown in Figure 5(c). The computed value of the Strouhal number (namely, 0.164) is essentially the same as the experimentally measured value (also 0.164).

4.4. Subsonic, transonic and supersonic flows over a bump

An inviscid flow in a channel with a bump is used to assess the algorithm for compressible flows over a range of Mach numbers, including subsonic, transonic and supersonic flows. These cases have been used as test cases by several researchers [24, 25] to test the accuracy and stability of their numerical methods. The chord length of the bump is the same as the channel height, and the channel length is three times its height. In all of the cases, the grid is symmetrical to the left and right of the middle of the bump. The thickness-to-chord ratio of the bump for the subsonic and transonic flows is 10% whereas that for the supersonic case is 4%. The grid is composed of three blocks: one upstream of the bump, one over the bump, and one downstream of the bump. The exact same grid distribution was used for all three flow cases. Grid dependence test were performed using systematically refined grids consisting of 51×21, 101×41, 201×81 and 301×151 points, respectively. For all the cases, the solutions obtained on the 201×81 and 301×151 grids were essentially the same. The enhanced algorithm, namely, Method 2, is used to march to steady state. A series of time step sizes were used. It is found that the largest time step size on the finest grid (301×151) for which the computations are stable are approximately 2.0×10^{-5}, 1.5×10^{-5} and 1.0×10^{-5} for the subsonic, transonic and supersonic cases, respectively. A similar *CFL* number limit is found to hold on all the grids. Consequently, larger time steps can be used on the coarser grids. For the subsonic and transonic cases the second-order upwind scheme [12] was used, whereas for the supersonic case the TVD-based controlled variation scheme [13], mentioned earlier, was employed.

For the subsonic case, the boundary conditions involve fixing the mass flux, total temperature and the flow angle at the inlet, and the static pressure at the outlet. The assigned boundary conditions yield a Mach number $M = 0.5$ at the inlet, for which the flow remains subsonic throughout the channel. The analytical solution for this case is symmetrical with no shock waves in the domain, which is also evident in the computed pressure contours plotted in Figure 6(a). The transonic case involves similar boundary conditions as the subsonic case such that the inlet Mach number is 0.675. The maximum Mach number immediately before the shock which forms on the second half of the bump is 1.43. The present computed solution, shown in Figure 6(b), corresponds to the reference solutions from the literature [24, 25]. For the supersonic case, the inlet Mach number is 1.65. All the flow variables are specified at the inlet. At the outlet, all the variables are extrapolated from the interior of the domain. Two oblique shocks are formed at the two corners of the bump. The shock from the leading edge of the bump reflects from the top wall and intersects with the shock from the trailing edge of the bump. The computed solution matches very well with the reference solutions from the literature [24, 25]. It was reported in an earlier study [6] that a PISO-based algorithm failed to yield converged solutions for the supersonic bump flow when very fine grids were used. However, in the present study, converged steady state solutions were obtained on all grids, the finest grid employed being 301×151. The plot of *CFL* number contours for the supersonic flow case on the finest grid are shown in

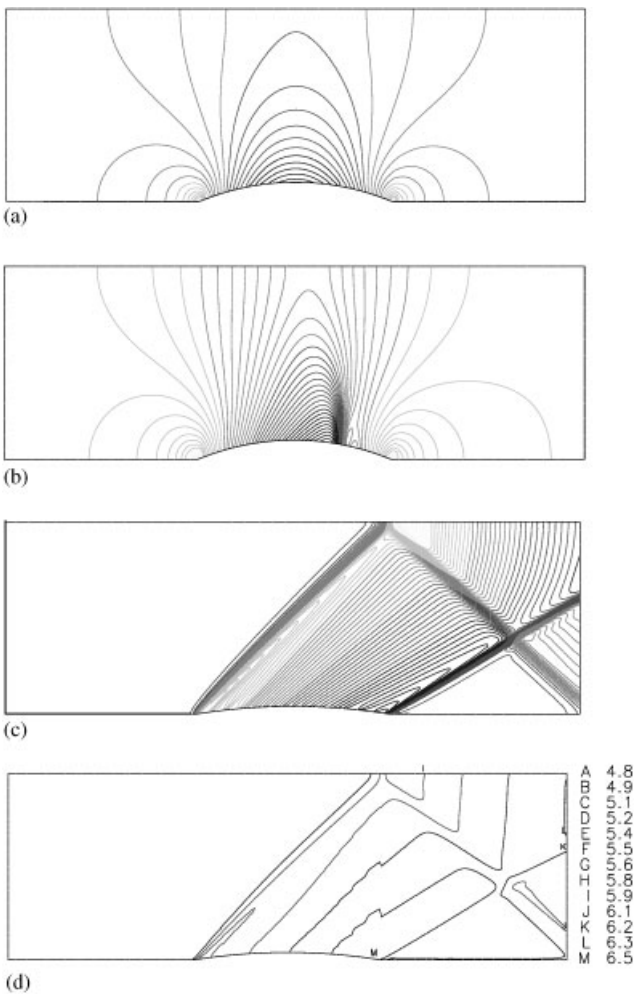


Figure 6. Pressure and *CFL* number contours for flow over a bump at different inlet Mach numbers ($\Delta t = 1.0 \times 10^{-5}$) on the finest grid, namely, 301×151: (a) Pressure contours for subsonic flow (inlet $M = 0.5$); (b) pressure contours for transonic flow (inlet $M = 0.675$); (c) pressure contours for supersonic flow (inlet $M = 1.65$); and (d) CFL number contours for supersonic flow (inlet $M = 1.65$).

Figure 6(d). The maximum *CFL* number for which the method is stable is approximately 6, which is about half of the maximum *CFL* number attained for the incompressible cylinder flow presented earlier. The reason for this needs further investigation.

4.5. Supersonic and hypersonic flows past a wedge

This case involves steady supersonic and hypersonic flows past a wedge placed in a channel. The lower half of the geometry is shown in Figure 7(a). The flow is two-dimensional, though

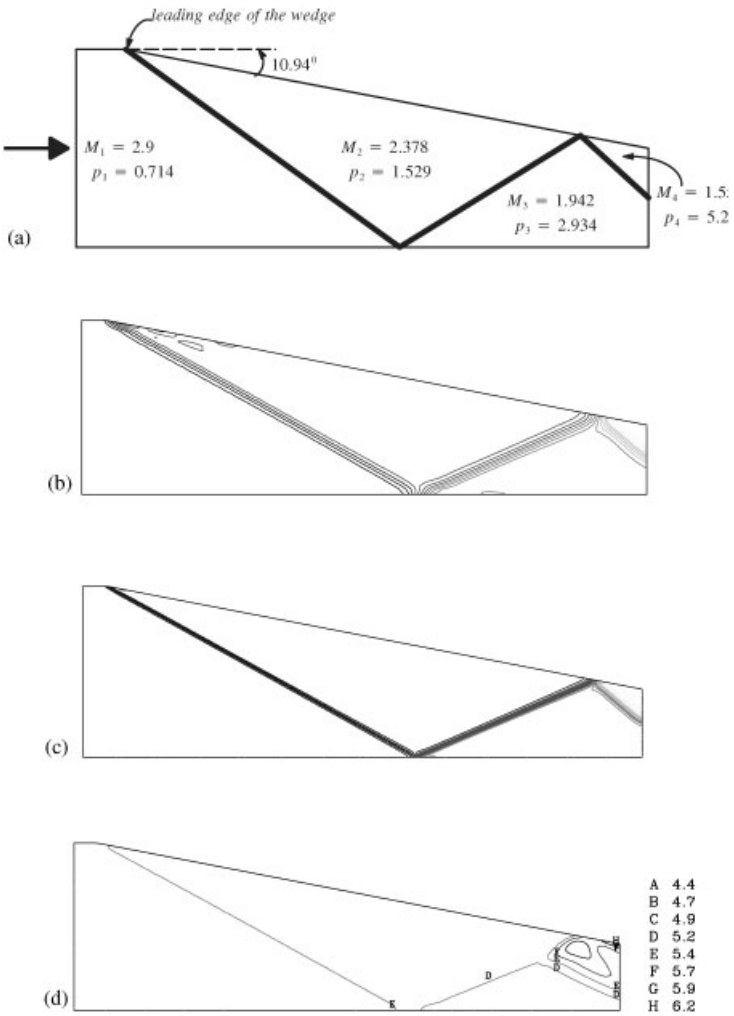


Figure 7. Analytical and computed supersonic flow past a wedge with inlet Mach number of 2.9: (a) Analytical solution of the flow; (b) pressure contours on 101×31 grid; (c) pressure contours on 201×61 grid; and (d) *CFL* number contours on 101×31 grid.

a three-dimensional grid is used for the computations. The flow enters the domain from the left and exits to the right. The top and bottom boundaries are assigned as slip walls. For the supersonic flow, the assigned boundary conditions yield a Mach number $M = 2.9$ at the inlet, for which the flow remains supersonic throughout the channel. The boundary conditions involve fixing the magnitude of all the variables at the inlet. The solution contains a shock wave generated at the leading edge of the wedge and subsequently reflected at the lower channel wall and then the upper (wedge) boundary. Figure 7 shows the computed pressure contours on grids with 101×31 and 201×61 nodes, which show essentially the same shock

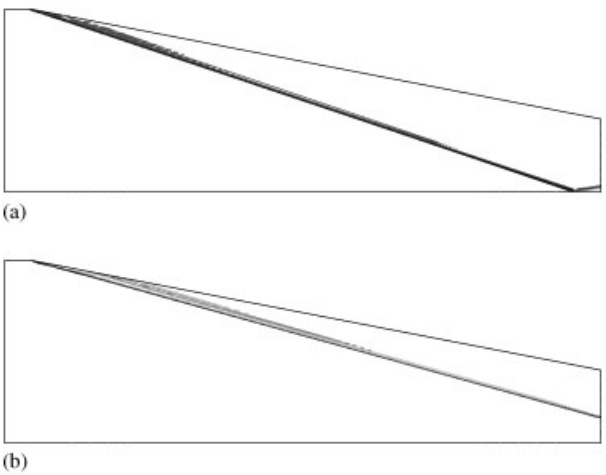


Figure 8. Computed hypersonic flows past a wedge on a 251×81 grid with $\Delta t = 1.0 \times 10^{-5}$: (a) Inlet Mach number = 6; and (b) inlet Mach number = 10.

pattern as the analytical solution shown in Figure 7(a). The maximum time step size that can be used on the 101×31 grid is 1.0×10^{-4} which corresponds to a maximum *CFL* number in the domain approximately equal to 6, as can be seen from the plot of *CFL* number contours in Figure 7(d).

To demonstrate the applicability of the present method to hypersonic Mach numbers, the wedge geometry is used to compute flows with Mach numbers equal to 6 and 10, respectively. At inlet Mach number equal to 6, the shock from the edge of the wedge is more oblique than that obtained at Mach number equal to 2.9 and reflects only from the bottom wall before exiting the computational domain, as shown in Figure 8(a). The shock obtained for inlet Mach number equal to 10 is even more oblique and exits the domain without reflecting at the bottom wall, as seen in Figure 8(b). These results indicate that the present method is both accurate and stable at high Mach numbers.

4.6. Sliding mesh turbomachinery rotor–stator computation

To demonstrate the robustness of the algorithm described in this paper for compressible flows in complex 3-D geometries, unsteady interaction between the stationary stator and the rotating blades of the rotor in a centrifugal compressor is considered. The original geometry of the compressor consists of 17 blades. Two stages of the compressor are modelled. The grid consists of 54 blocks with 4.2 million nodes. Three rotor–stator interfaces exist in this geometry, as shown in Figure 9: (1) interface between the inlet region and the first-stage impeller, (2) interface between the return channel and the inlet of the second-stage impeller, and (3) interface between the exit of the second-stage impeller and the diffuser. The blocks in the stator region of the domain are solved in the absolute frame of reference whereas those in the rotor region are solved in the rotating frame of reference. The flow information is exchanged at the rotor–stator interfaces by constructing ghost cells on either side of the

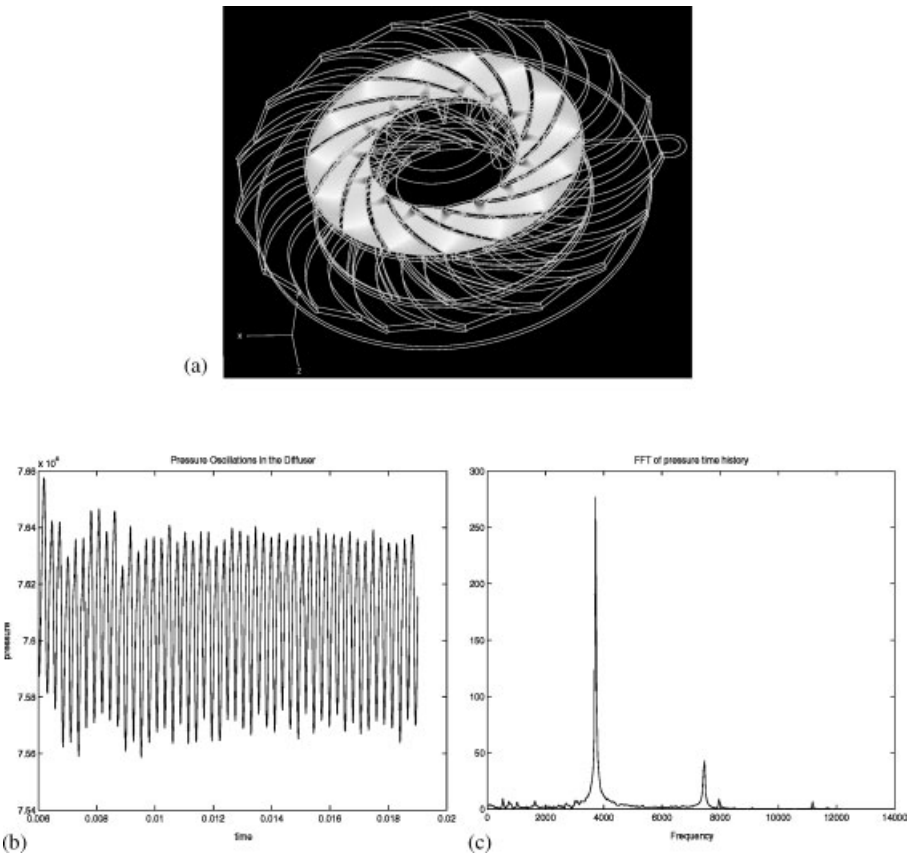


Figure 9. Illustration of sliding mesh computation of unsteady, turbulent flow in a two-stage centrifugal compressor. The time step used is 5.0×10^{-6} which corresponds to approximately 1° rotation of the rotor: (a) The multiblock grid and the transient pressure field in the first-stage impeller; (b) pressure fluctuation at diffuser inlet; and (c) FFT of pressure showing the harmonics.

interfaces and employing three-dimensional interpolations followed by an appropriate change of reference frame for the local velocity vectors. The details of the sliding mesh algorithm to model the moving rotor–stator interface are presented elsewhere [26]. The results from the fully transient sliding mesh computation are shown in Figure 9. A time step size equal to 5.0×10^{-6} is selected based on the criterion that the rotor mesh moves by 1° per time step. The transient pressure field in the blade passages in the first-stage impeller is shown in Figure 9(a). The fluctuation of pressure at one point at the inlet of the diffuser is shown in Figure 9(b). The FFT analysis of this pressure signal, shown in Figure 9(c), reveals the presence of the primary harmonic frequency—which corresponds to the rotational speed of the impeller—but also a strong secondary harmonic frequency. This computation illustrates that the present operator-splitting algorithm (Method 2) is very robust and efficient for engineering flows with complex physics in complex geometries.

5. CONCLUSION

In this paper, a non-iterative operator-splitting pressure-based algorithm for computing all-speed flows in complex geometries is described. The method is based on the PISO algorithm of Issa [1]. However, several modifications and enhancements to the original method have been made to adapt it for: (a) high-speed flows, and (b) multiblock, non-orthogonal, body-fitted grids for very complex geometries. The method is shown to be robust for all Mach number regimes. Enhancements to the method which allow for stable solutions to be obtained on highly skewed grids are also discussed. Since the method dispenses with iterations within each time step, it is suitable for unsteady flows. However, the method is demonstrated to be suitable for steady flows too since it is demonstrated to be stable for *CFL* numbers of the order of 10 for all Mach numbers, allowing large time steps to be taken for steady flows. The method is found to be very reliable even in complex engineering applications such as unsteady rotor–stator interactions in turbulent, all-speed turbomachinery flows.

ACKNOWLEDGEMENTS

The geometry for the last case presented in this paper was provided by Dr Lee Hill of the Dresser-Rand Company of Olean, New York. The assistance of Dr Inanc Sonecac in conducting the unsteady cylinder flow presented in this paper is also acknowledged. The authors are also grateful to Dr. Wei Shyy of the University of Florida for technical discussions. This work was supported in part by the URETI program funded by NASA.

REFERENCES

1. Issa RI. Solution of the implicitly discretised fluid flow equations by operator splitting. *Journal of Computational Physics* 1986; **62**:40–65.
2. Issa RI, Gosman AD, Watkins AP. The computation of compressible and incompressible recirculating flows by a non-iterative scheme. *Journal of Computational Physics* 1986; **62**:66–82.
3. Issa RI, Ahmadi-Befrui B, Beshay KR, Gosman AD. Solution of the implicitly discretised reacting flow equations by operator-splitting. *Journal of Computational Physics* 1991; **93**:388–410.
4. Oliveira PJ, Issa RI. An improved PISO algorithm for the computation of buoyancy driven flows. *Numerical Heat Transfer Part B* 2001; **40**:473–493.
5. Moukalled F, Darwish M. A unified formulation of the segregated class of algorithms for fluid flow at all speeds. *Numerical Heat Transfer Part B* 2000; **37**:103–139.
6. Bressloff NW. A parallel pressure implicit splitting of operators algorithm applied to flows at all speeds. *International Journal for Numerical Methods in Fluids* 2001; **36**:497–518.
7. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: Washington, DC, 1980.
8. Thakur S, Wright J. *STREAM*: *A Computational Fluid Dynamics and Heat Transfer Code for Complex Geometries. Part 1*: *Theory. Part 2*: *User's Guide*. Streamline Numerics, Inc.: Gainesville, FL, 1999.
9. Karki KC, Patankar SV. Pressure-based calculation procedure for viscous flows at all speeds in arbitrary configurations. *AIAA Journal* 1989; **27**:1167–1174.
10. Thakur S, Shyy W, Liou MS. Convection treatment and pressure splitting for sequential solution procedures. Part II: pressure-based algorithm. *Numerical Heat Transfer* 1996; **29**:29–42.
11. Thakur S, Shyy W, Udaykumar HS, Hill L. Multiblock interface treatments in a pressure-based solver. *Numerical Heat Transfer Part B* 1998; **33**:367–396.
12. Thakur S, Shyy W. Some implementational issues of convection schemes for finite volume formulations. *Numerical Heat Transfer Part B* 1993; **24**:31–35.
13. Shyy W, Thakur S. Development of a controlled variation scheme in a sequential solver for recirculating flows. Part I. Theory and formulation. *Numerical Heat Transfer Part B* 1994; **25**:245–272.
14. Rhie CL, Chow WL. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA Journal* 1983; **21**:1525–1532.
15. Majumdar S. Role of underelaxation in momentum interpolation for calculation of flow with nonstaggered grids. *Numerical Heat Transfer* 1988; **13**:125–132.

16. Shyy W, Thakur SS, Ouyang H, Liu J, Blosch E. *Computational Techniques for Complex Transport Phenomena*. Cambridge University Press: New York, 1987.
17. Favre A. Equations des Gaz Turbulents Compressibles. *Journal de Mecanique* 1965; **4**:361–390.
18. Jones WP, Launder BE. The prediction of laminarization with a two-equation model of turbulence. *International Journal of Heat Transfer* 1972; **15**:301–314.
19. Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics* (3rd edn). Springer: Berlin, 2002.
20. Hutchinson BR, Raithby GD. A multigrid method based on the additive correction strategy. *Numerical Heat Transfer* 1986; **9**:511–537.
21. Xu H, Zhang C. Study of the effect of the non-orthogonality for non-staggered grids—the results. *International Journal for Numerical Methods in Fluids* 1999; **29**:625–644.
22. Coutanceau M, Bouard R. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. *Journal of Fluid Mechanics* 1977; **79**:257.
23. Williamson CHK. Oblique and parallel mode of vortex shedding in the wake of as cylinder at low Reynolds number. *Journal of Fluid Mechanics* 1989; **206**:579–627.
24. Demirdzic I, Lilek Z, Peric M. A collocated finite volume method for predicting flows at all speeds. *International Journal for Numerical Methods in Fluids* 1993; **16**:1029–1050.
25. Issa RI, Javareshkian MH. Pressure-based compressible calculation method utilizing total variation diminishing schemes. *AIAA Journal* 1998; **36**:1652–1657.
26. Thakur S, Wright J. CFD predictions of turbomachinery flows using quasi-steady and unsteady models. *AIAA 2003-4133*, *33rd AIAA Fluid Dynamics Conference*, Orlando, FL, 2003.

