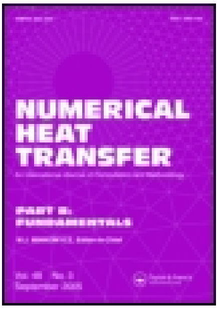


Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/unhb20

A UNIFIED FORMULATION OF THE SEGREGATED CLASS OF ALGORITHMS FOR FLUID FLOW AT ALL SPEEDS

F. Moukalled, M. Darwish

Published online: 29 Oct 2010.

Numerical Heat Transfer, Part B, 37:103–139, 2000
Copyright © 2000 Taylor & Francis
1040–7790 / 00 $12.00 + .00

A UNIFIED FORMULATION OF THE SEGREGATED CLASS OF ALGORITHMS FOR FLUID FLOW AT ALL SPEEDS

F. Moukalled and M. Darwish
Mechanical Engineering Department, Faculty of Engineering & Architecture,
American University of Beirut, P.O. Box 11-0236, Beirut, Lebanon

In this article, the segregated SIMPLE algorithm and its variants are reformulated, using a collocated variable approach, to predict fluid flow at all speeds. In the formulation, a unified, compact, and easy-to-understand notation is employed. The SIMPLE, SIMPLER, SIMPLEST, SIMPLEM, SIMPLEC, SIMPLEX, PRIME, and PISO algorithms that are scattered in the literature and appear to a nonversed computational fluid dynamics (CFD) user as being unrelated, are shown to share the same essence in their derivations and to be equally applicable for the simulation of incompressible and compressible flows. Moreover, the philosophies behind these algorithms in addition to their similarities and differences are explained.

INTRODUCTION

Over the last two decades important advances in computational fluid dynamics (CFD), pertaining to the development and maturity of solution algorithms, have been achieved. In this work, a solution algorithm, such as the SIMPLE [1, 2] algorithm, denotes the procedure used to resolve the coupling that arises in the solution of Navier-Stokes equations between velocity, density, and pressure. Many difficulties connected with these algorithms have been resolved and better insight gained. In specific, work has been directed toward settling a number of pending issues related to the choice of primitive variables (density-based versus pressure-based), the grid arrangement (staggered versus collocated arrangement), and the solution approach (semidirect versus segregated approach). Developments in these areas are briefly reviewed next. In the choice of primitive variables, density-based algorithms, in which the continuity equation acts as an equation for density while pressure is obtained from the energy (enthalpy) and state equations, have been used successfully in the simulation of highly compressible flows. For low Mach number flows, however, as small disturbances in density result in large variations in the pressure field, density-based algorithms become unstable and their convergence rate greatly diminishes. Moreover, despite the extension of this class of algorithms to predicting incompressible flows through the use of the so-called pseudo or artificial compressibility technique [3–5], the difficulties encountered in efficiently avoiding the stiff solution matrices generated by these methods have led

Received 1 April 1999; accepted 2 June 1999.

The financial support provided by the University Research Board of the American University of Beirut is gratefully acknowledged.

Address correspondence to Dr. F. Moukalled, Mechanical Engineering Department, American University of Beirut, P.O. Box 11-0236, Beirut, Lebanon. E-mail: memouk@aub.edu.lb

NOMENCLATURE

a_P^{ϕ}, a_E^{ϕ},	coefficients in the discretized equation for ϕ
$\|\|a, b\|\|$	the maximum of a and b
b_P^{ϕ}	source term in the discretized equation for ϕ
C_p	coefficient equal to 1 /RT
$\hat{\mathbf{d}}_f$	covariant unit vector (i.e., in the direction of $\mathbf{d}_f$)
$D[\phi]$	the D operator
$\tilde{D}[\phi]$	the modified D operator
$\mathbf{D}[\phi]$	the vector form of the D operator
$\tilde{\mathbf{D}}[\phi]$	the vector form of the modified D operator
F_f	convective flux at cell face f
f_f	interpolation factor
$H[\phi]$	the H operator
$\mathbf{H}[\phi]$	the vector form of the H operator
$\tilde{H}[\phi]$	the modified H operator
$\tilde{\mathbf{H}}[\phi]$	the vector form of the modified H operator
i	unit vector in the x direction
j	unit vector in the y direction
P	pressure
Q^{ϕ}	source term in the conservation equation for ϕ
R	gas constant
$\mathbf{R}_M$	momentum residual
$\hat{\mathbf{s}}_f$	contravariant unit vector (i.e., in the direction of $\mathbf{S}_f$)
$\mathbf{S}_f$	surface vector
T	temperature
t	time
u, v	velocity components in the x and y directions
U_f	interface flux velocity $(\mathbf{v}_f \cdot \mathbf{S}_f)$
V	cell volume
$\mathbf{v}$	velocity vector $(u\mathbf{i} + v\mathbf{j})$
x, y	Cartesian coordinates

α	underrelaxation factor
β	thermal expansion coefficient
δt	time step
γ	scaling factor
Γ^{ϕ}	diffusion coefficient for ϕ
$\Delta[\phi]$	the Δ operator
ϕ	dependent variable
ϕ_f	scalar value at cell face f
Φ	dissipation term in energy equation
$\mathbf{\kappa}_f$	space vector equal to $(\hat{\mathbf{n}}_f - \gamma\hat{\mathbf{d}}_f)S_f$
μ	viscosity
ρ	density

Subscripts

e, w, ...	refers to the east, west, ... faces of a control volume
E, W, ...	refers to the east, west, ... neighbors of the main grid point
f	refers to control-volume face f
F	refers to main grid point F
P	refers to the P grid point

Superscripts

C	refers to convection contribution
D	refers to diffusion contribution
(n)	refers to value from the previous iteration
sx	refers to SIMPLEX
x	refers to component in x direction
y	refers to component in y direction
ϕ	refers to dependent variable
*, **, ...	refers to first, second, ..., updated value at the current iteration
^{o}	refers to values from the previous time step
$'$	refers to correction field

a number of researchers [6–10] to work on extending the pressure-based algorithms, originally developed to solve incompressible flows, to this class of flows, thus encompassing the entire subsonic to hypersonic spectrum.

With pressure-based solution algorithms, two different approaches have been followed, which are denoted in the literature by the semidirect approach and the segregated approach, respectively. In the semidirect method, first proposed by Caretto et al. [11], the discretized forms of the momentum and continuity equations are linked together and the resulting system of equations is solved simultaneously. This technique guarantees a close connection between the velocity components and the pressure, which enhances the convergence rate and hence the efficiency of the method. However, the memory required to store the various coefficients at all grid nodes is often prohibitive, particularly for multidimensional and multiphase flows. This storage problem can be alleviated by subdividing the domain into parts and solving the system of equations over each subdomain separately. These so-called subdomain solvers, which degenerate to point solvers when the domain is confined to one volume, suffer a connectivity problem and require iterations among the various parts of the domain, leading to a drastic loss of the convergence rate [12–14]. In the second, much more popular segregated approach, the discretized forms of the various differential equations are solved separately, but over the whole domain. This has the advantage of requiring considerably less computer storage than the semidirect method in addition to providing the flexibility of easily solving additional partial differential equations (such as turbulence kinetic energy, turbulence dissipation rate, concentration of chemical species, etc.) when needed. In this technique, while the velocity components are obtained from the corresponding momentum equations, there is no apparent equation governing pressure. To derive a pressure or an equivalent pressure-correction equation, the discretized forms of the continuity and momentum equations are combined together. Moreover, for compressible flows, density is replaced by pressure through the equation of state. For an incompressible flow, this pressure or pressure-correction equation is elliptic, while for a compressible flow it is hyperbolic. Once the pressure (or pressure-correction) field is calculated, the velocity and density fields are updated to satisfy the continuity equation. Hence, in compressible flow simulations the pressure plays a dual role by acting on both the density and the velocity through the equation of state and momentum conservation, respectively, so that mass conservation is satisfied. Obviously, for low-speed flow, the pressure gradient required to drive the velocities through momentum conservation is of such magnitude that the density is not significantly affected and the flow can be considered nearly incompressible. In the hypersonic limit where variations in velocity become relatively small as compared to the velocity itself, the changes in pressure do affect density significantly. In fact, in this limit, the pressure can be viewed to act on density alone through the equation of state so that mass conservation is satisfied [6]. This dual role played by the pressure field explains why pressure-based algorithms have been applied with success to both incompressible and compressible flows.

The cell-centered or collocated variable scheme has been used quite successfully with density-based algorithms. With pressure-based algorithms, however, the use of this grid layout is not as straightforward. While the momentum equations link the velocity to the respective pressure gradients, the continuity equation, apparently having no direct link to pressure, acts as a constraint on the velocity field. Consequently, the convergence and stability of pressure-based algorithms depend largely on how the pressure gradients and velocities are evaluated in the continuity and momentum equations. Early attempts to use collocated variables in pressure-based algorithms failed to give converged solutions, producing spurious oscillations, and resulting in the so-called red–black checkerboard splitting of the pressure field [1]. This undesirable behavior stems from the linear interpolation practice used in evaluating the velocity at the control-volume faces in the continuity equation, which causes the velocity there to be related to the pressure difference between two alternating, rather than consecutive, nodes. The use of staggered grids, first proposed by Harlow and Welsh [15], in which pressure is stored at the control-volume center and velocities at the cell faces, removes the need for interpolation of pressure in the momentum equations and of velocity in the continuity equation. This is a considerable advantage and hence staggered grids became very popular and have been used with great success to solve a wide range of problems in Cartesian, polar, and also more general orthogonal coordinates. However, the primary disadvantage of the staggered grid arrangement is the greater geometric, and related mathematical, complexity associated with the use of different grid systems for the various variables, which becomes overwhelming in curvilinear coordinates. Another disadvantage of the staggered-grid approach is related to the choice of the contravariant, covariant, or even Cartesian velocity components at the control-volume faces [16]. A better solution to avoid the checkerboard splitting of the pressure field was proposed independently, at the beginning of the 1980s, by several workers through the use of a special interpolation procedure for evaluating the control-volume face velocities. Depending on the path followed, two collocated methods have been devised, which are denoted in the literature as the pressure-weighted interpolation method [17–20] (PWIM) and the momentum-weighted interpolation method [21–25] (MWIM), respectively. Both methods reconstruct the momentum equations at the cell faces by interpolating the coefficients at the cell centers.

It is apparent that while decisions on the above-mentioned points at one time seemed difficult, the direction followed by the CFD community (in connection with the finite-volume method, FVM) is now much more distinct. Although the semidirect versus segregated approach dichotomy is still not clearly resolved, especially with the increasing use of multigrid techniques [8, 26–29], it is now widely believed that collocated pressure-based solution algorithms for predicting fluid flow at all speeds are optimum in the use of resources. Moreover, the segregated approach when combined with multigrid acceleration techniques, at least for the near future, still represents the most efficient approach.

Within this context, a large number of solution algorithms [1, 2, 30–37] have been proposed. The first of these algorithms was the SIMPLE (Semi-Implicit Method for the Pressure-Linked Equation) algorithm of Patankar and Spalding [1, 2], developed in the late 1960s and beginning 1970s. Over the years, a number of modifications to the SIMPLE algorithm have been suggested with the aim of improving the robustness and/or convergence rate. The SIMPLER (SIMPLE Revised) algorithm of Patankar [1], proposed in 1981, the SIMPLEC (SIMPLE Consistent) algorithm of Van Doormaal and Raithby [30], proposed in 1984, the SIMPLEST algorithm (SIMPLE ShorTened) of Spalding [31], the SIMPLEX algorithm of Van Doormal and Raithby [32], the SIMPLEM algorithm (SIMPLE-Modified) of Acharya and Moukalled [33]. The PISO (Pressure Implicit with Split Operator) algorithm of Issa [34], and the PRIME (PRessure Implicit Momentum Explicit) algorithm of Maliska and Raithby [35] are the most important attempts to improve on the original solution algorithm. Sometimes minor changes were introduced to these algorithms, such as in the MSIMPLEC [36], MPISO [36], SIMPLESSEC [37], and SIMPLESSE [37] algorithms.

These SIMPLE-like algorithms, scattered in the literature using different notation and appearing to a nonversed CFD user as being unrelated are in fact very similar and share the same essence in their derivations. Thus, the main objective of this work is to derive these algorithms using a unified, compact, and easy-to-understand notation that can be expanded systematically to yield the coefficients of the pressure-correction equation, hence facilitating the implementation of these algorithms for new CFD users. In the process, the philosophies behind these algorithms in addition to their similarities and differences are explained. Moreover, because they were originally developed for incompressible-flow calculations, the majority of these algorithms has not yet been extended to simulate hypersonic flows (compressible), even though such extension has been successfully applied to the original SIMPLE solution algorithm [6, 8, 10, 17]. Therefore, a second objective of this work is to extend all the above-mentioned algorithms to solve for fluid flow at all speeds, and to demonstrate in so doing that the incompressibility limitation is not inherent to these algorithms.

In what follows the governing equations for compressible flow are presented and their discretization outlined so as to lay the ground for the derivation of the pressure and pressure-correction equations, which are developed using a unified notation and shown to be equivalent mathematically. Moreover, the use of a unified notation also helps in understanding the importance of the different influences affecting the convergence properties of the various algorithms. However, it should be stressed that the purpose of the article is not to compare the relative performance of the different SIMPLE-like algorithms; rather, the aim is to unify their formulation.

GOVERNING EQUATIONS

The equations governing the flow of a two-dimensional compressible fluid are the continuity equation, the momentum equations, and the energy equation. This set of nonlinear, coupled equations is solved for the unknowns ρ, $\mathbf{v}$, T, and P. In vector form, these equations may be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{1}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla P + \nabla \cdot (\mu \nabla \mathbf{v}) + \frac{1}{3}\nabla(\mu \nabla \cdot \mathbf{v}) \tag{2}$$

$$\frac{\partial (\rho T)}{\partial t} + \nabla \cdot (\rho \mathbf{v} T) = \frac{1}{c_p}\left\{\nabla \cdot (k \nabla T) + \beta T\left[\frac{\partial P}{\partial t} + \nabla \cdot (P\mathbf{v}) - P\nabla \cdot (\mathbf{v})\right] + \Phi + \dot{q}\right\} \tag{3}$$

where

$$\Phi = \mu\left\{2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 - \frac{2}{3}(\nabla \cdot \mathbf{v})^2\right\} \tag{4}$$

and β is the thermal expansion coefficient, which is equal to $1/T$ for an ideal fluid. In addition to the above differential equations, an auxiliary equation of state relating density to pressure and temperature $[\rho = f(P, T)]$ is needed. For an ideal fluid, this equation is given by

$$\rho = \frac{P}{RT} = C_p P \tag{5}$$

where R is the gas constant.

A review of the above differential equations reveals that they are similar in structure. If a typical representative variable is denoted by ϕ, the general differential equation may be written as

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho \mathbf{v} \phi) = \nabla \cdot (\Gamma^{\phi} \nabla \phi) + Q^{\phi} \tag{6}$$

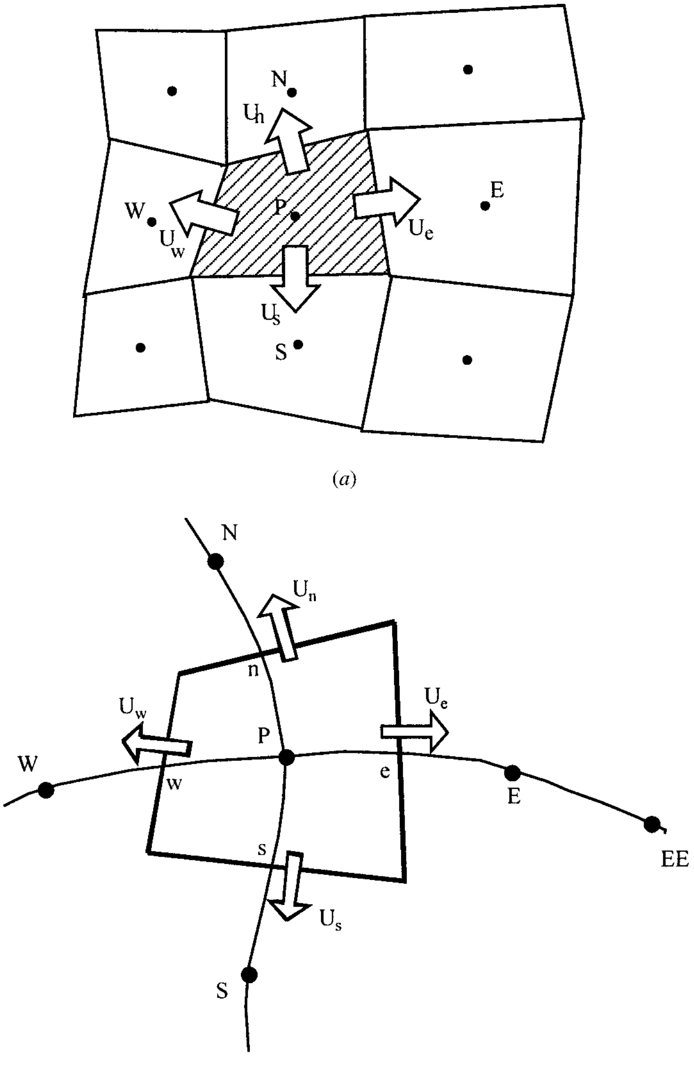
where the expressions for Γ^{ϕ} and Q^{ϕ} can be deduced from the parent equations. The four terms in the above equation describe successively unsteadiness, convection (or advection), diffusion, and generation/dissipation effects. In fact, all terms not explicitly accounted for in the first three terms are included in the catchall source term Q^{ϕ}.

FINITE-VOLUME DISCRETIZATION

The general transport equation [Eq. (6)] is discretized using the control-volume approach. In this method, the solution domain is divided into a number of control volumes, each associated with a main grid point P (Figure 1). The discretized form is obtained by first integrating the general transport equation over the control volume shown in Figure 1 and then reducing the integrated equation to an algebraic equation by expressing the variation in the dependent variable and its derivatives, using interpolation profiles, in terms of the grid point values. This results in the following algebraic equation linking the value of the dependent variable at the control-volume center to the neighboring dependent variable values:

$$a_{\mathrm{P}}^{\phi} \phi_{\mathrm{P}} = \sum_{\mathrm{NB(P)}} a_{\mathrm{NB}}^{\phi} \phi_{\mathrm{NB}} + b_{\mathrm{P}}^{\phi} \tag{7}$$

In the above equation, a_{NB}^{ϕ} are the coefficients multiplying the value of ϕ at the neighboring nodes NB(P) = (E, W, N, S) surrounding the central node P, a_{P}^{ϕ} is the coefficient of ϕ_{P}, and b_{P}^{ϕ} contains all terms that are not expressed through the nodal values of the dependent variable. An equation similar to Eq. (7) is obtained at every grid point in the domain and the collection of these equations forms a system of algebraic equations that is solved to obtain the ϕ field. Since the coefficients in these equations are in general dependent on ϕ, an iterative solution scheme should be employed for handling this nonlinearity. During the iterative process, it is often desirable to slow the changes, from iteration to iteration, in the



(*a*)

(*b*)

Figure 1. Control volume.

values of the dependent variable. This process is called underrelaxation. It is an important tool that prevents divergence of the iterative solution for strongly nonlinear problems, as is the case here. If ϕ_P^* and ϕ_P are the values from the previous and current iterations, respectively, then Eq. (7) can be written as

$$\phi_P = \phi_P^* + \underline{\left(\frac{\sum_{NB(P)} a_{NB}^{\phi} \phi_{NB} + b_P^{\phi}}{a_P^{\phi}} - \phi_P^* \right)} \tag{8}$$

In the above equation, the underlined term represents the change in ϕ_{P} produced by the current iteration. This change can be varied by introducing and underrelaxation factor α $(0 \leq \alpha \leq 1)$, so that

$$\phi_{\mathrm{P}} = \phi_{\mathrm{p}}^{*} + \underline{\alpha\left(\frac{\sum_{\mathrm{NB(P)}} a_{\mathrm{NB}}^{\phi}\phi_{\mathrm{NB}} + b_{\mathrm{P}}^{\phi}}{a_{\mathrm{P}}^{\phi}} - \phi_{\mathrm{P}}^{*}\right)} \tag{9}$$

or

$$\frac{a_{\mathrm{P}}^{\phi}}{\alpha}\phi_{\mathrm{P}} = \sum_{\mathrm{NB(P)}} a_{\mathrm{NB}}^{\phi}\phi_{\mathrm{NB}} + b_{\mathrm{P}}^{\phi} + (1-\alpha)\frac{a_{\mathrm{P}}^{\phi}}{\alpha}\phi_{\mathrm{P}}^{*} \tag{10}$$

At the state of convergence, ϕ_{P}^{*} and ϕ_{P} are equal and the original equation is satisfied. There are no general rules for choosing the optimum value of α, and a suitable value for a given problem is usually found from exploratory computations. Equation (10) can be rewritten in the form of Eq. (7) by redefining a_{P}^{ϕ} and b_{P}^{ϕ} as follows:

$$\left\{\begin{array}{l} a_{\mathrm{P}}^{\phi} \leftarrow \dfrac{a_{\mathrm{P}}^{\phi}}{\alpha} \\ b_{\mathrm{P}}^{\phi} \leftarrow b_{\mathrm{P}}^{\phi} + (1-\alpha)\dfrac{a_{\mathrm{P}}^{\phi}}{\alpha}\phi_{\mathrm{P}}^{*} \end{array}\right. \tag{11}$$

THE SEGREGATED APPROACH FOR SOLVING FLOW PROBLEMS

The need for a solution algorithm arises in the simulation of flow problems because a scalar equation does not exist for pressure. Rather, the pressure field acts indirectly on the velocity field to constraint it to satisfy the continuity equation. Hence, if a segregated approach is to be adopted, coupling between the u, v, ρ, and P primitive variables within the continuity and momentum equations will be required. Evidently, the whole set of equations could be solved directly (after linearization), since the number of equations equals the number of unknowns. However, the computational effort and storage requirements needed by such an approach are often prohibitive, particularly for multidimensional and multiphase flows. This has forced researchers to seek less expensive methods and resulted in the development of the various segregated solution algorithms.

The segregated approach is iterative in nature and involves a *predictor* and a *corrector* step. In the *predictor* step, the velocity field is calculated based on a guessed or estimated pressure field. In the *corrector* step, a pressure (or a pressure-correction) equation is derived and solved. Then, the variation in the pressure field is accounted for within the momentum equations by corrections to the velocity and density fields. Thus, the velocity, density, and pressure fields are driven, iteratively, to better satisfying the momentum and continuity equations simultaneously and convergence is achieved by repeatedly applying the above-described procedure.

Before deriving the pressure or pressure-correction equation, the discretized momentum equations are first written in the following notationally more suitable form:

$$\begin{aligned} a_{\mathrm{P}}^{u} u_{\mathrm{P}} &= \sum_{\mathrm{NB(P)}} a_{\mathrm{NB}}^{u} u_{\mathrm{NB}} + b_{\mathrm{P}}^{u} - V(\nabla P)_{\mathrm{P}} \cdot \mathbf{i} \\ a_{\mathrm{P}}^{v} v_{\mathrm{P}} &= \sum_{\mathrm{NB(P)}} a_{\mathrm{NB}}^{v} v_{\mathrm{NB}} + b_{\mathrm{P}}^{v} - V(\nabla P)_{\mathrm{P}} \cdot \mathbf{j} \end{aligned} \tag{12}$$

This form can be simplified to

$$\begin{Bmatrix} u_{\mathrm{P}} \\ v_{\mathrm{P}} \end{Bmatrix} - \begin{Bmatrix} H[u]_{\mathrm{P}} \\ H[v]_{\mathrm{P}} \end{Bmatrix} = -\begin{Bmatrix} D[u]_{\mathrm{P}} & 0 \\ 0 & D[v]_{\mathrm{P}} \end{Bmatrix} \begin{Bmatrix} (\nabla P)_{\mathrm{P}} \cdot \mathbf{i} \\ (\nabla P)_{\mathrm{P}} \cdot \mathbf{j} \end{Bmatrix} \tag{13}$$

where

$$(\nabla P)_{\mathrm{P}} = \frac{1}{V} \int_{V} \nabla P \, dV \qquad H[\phi]_{\mathrm{P}} = \frac{\Sigma_{\mathrm{NB(P)}} a_{\mathrm{NB}}^{\phi} \phi_{\mathrm{NB}} + b_{\mathrm{P}}^{\phi}}{a_{\mathrm{P}}^{\phi}} \qquad D[\phi]_{\mathrm{P}} = \frac{V}{a_{\mathrm{P}}^{\phi}} \tag{14}$$

In the above equations, V is the volume of cell P, and the subscripts e, w, n, and s refer to values at the east, west, north, and south faces of the control volume (Figure 1). Defining the vector forms of the above operators as

$$\begin{aligned} \mathbf{H}[\mathbf{v}]_{\mathrm{P}} &= \begin{bmatrix} H[u]_{\mathrm{P}} \\ H[v]_{\mathrm{P}} \end{bmatrix} \qquad \mathbf{D}_{\mathrm{P}} = \begin{bmatrix} D[u]_{\mathrm{P}} & 0 \\ 0 & D[v]_{\mathrm{P}} \end{bmatrix} \\ (\nabla P)_{\mathrm{P}} &= \begin{bmatrix} (\nabla P)_{\mathrm{P}} \cdot \mathbf{i} \\ (\nabla P)_{\mathrm{P}} \cdot \mathbf{j} \end{bmatrix} = \begin{bmatrix} (\nabla P)_{\mathrm{P}}^{x} \\ (\nabla P)_{\mathrm{P}}^{y} \end{bmatrix} \end{aligned} \tag{15}$$

the momentum equations in vector form become

$$\mathbf{v}_{\mathrm{P}} - \mathbf{H}[\mathbf{v}]_{\mathrm{P}} = -\mathbf{D}_{\mathrm{P}} (\nabla P)_{\mathrm{P}} \tag{16}$$

For later use, modified forms of the H and D operators are defined as follows:

$$\tilde{H}[\phi]_{\mathrm{P}} = \frac{\Sigma_{\mathrm{NB(P)}} a_{\mathrm{NB}}^{\phi} \phi_{\mathrm{NB}}}{a_{\mathrm{P}}^{\phi}} \qquad \tilde{\mathbf{H}}[\mathbf{v}]_{\mathrm{P}} = \begin{bmatrix} \tilde{H}[u]_{\mathrm{P}} \\ \tilde{H}[v]_{\mathrm{P}} \end{bmatrix} \qquad \tilde{\mathbf{D}}_{\mathrm{P}} = \frac{\mathbf{D}_{\mathrm{P}}}{1 - \tilde{\mathbf{H}}[1]_{\mathrm{P}}} \tag{17}$$

Since an equation for pressure will be derived by combining momentum and continuity, the discretized form of the continuity equation is needed and is obtained as (Figure 1)

$$\begin{gathered}\int_v \frac{\partial \rho}{\partial t}\, dV + \int_v \nabla \cdot (\rho \mathbf{v})\, dV = 0 \\ \frac{(\rho_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \oint_s (\rho \mathbf{v})\, dS = 0\end{gathered} \tag{18}$$

which can be written as

$$\begin{gathered}\frac{(\rho_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[\rho \mathbf{v} \cdot S]_{\mathrm{P}} = 0 \\ \frac{(\rho_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[\rho U]_{\mathrm{P}} = 0\end{gathered} \tag{19}$$

where

$$U_{\mathrm{f}} = \mathbf{v}_{\mathrm{f}} \cdot \mathbf{S}_{\mathrm{f}} \qquad \text{and} \qquad \Delta[\phi]_{\mathrm{P}} = \phi_{\mathrm{e}} + \phi_{\mathrm{w}} + \phi_{\mathrm{n}} + \phi_{\mathrm{s}} \tag{20}$$

For the calculation of the mass fluxes across the control-volume faces and for checking mass conservation, the values of the velocity components are needed there. In order to avoid oscillations, which may result if a simple linear interpolation is used, a special interpolation practice has to be employed. In this work, the MWIM is followed [23]. The basis for the interpolation procedure are the discretized momentum equations at the control-volume centers, as given by Eq. (16), where the pressure source term has been taken out of the Q^{ϕ} term and shown explicitly. To evaluate velocities at the control-volume face f, terms in Eq. (16) are selectively interpolated and evaluated at the f location to yield

$$\mathbf{v}_{\mathrm{f}} - \overline{\mathbf{H}}[\mathbf{v}]_{\mathrm{f}} = -\overline{\mathbf{D}}_{\mathrm{f}} (\nabla P)_{\mathrm{f}} \tag{21}$$

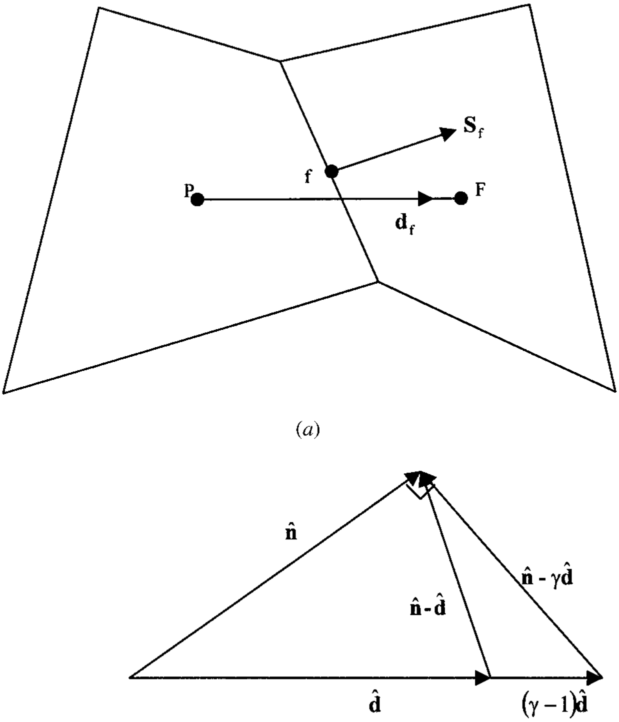
where the overbar denotes a linear interpolation between the two control volumes straddling the surface f. This equation, very similar to Eq. (16), may be viewed as a pseudo-momentum equation at the control-volume face. Since the mass fluxes are needed at the control-volume faces, rather than the velocity components, it is the value of U_{f} that should be calculated there. Thus,

$$U_{\mathrm{f}} = \mathbf{v}_{\mathrm{f}} \cdot \mathbf{S}_{\mathrm{f}} = \overline{\mathbf{H}}[\mathbf{v}]_{\mathrm{f}} \cdot \mathbf{S}_{\mathrm{f}} - \overline{\mathbf{D}}_{\mathrm{f}} (\nabla P)_{\mathrm{f}} \cdot \mathbf{S}_{\mathrm{f}} \tag{22}$$

$\overline{\mathbf{H}}[\mathbf{v}]_{\mathrm{f}}$ and $\overline{\mathbf{D}}_{\mathrm{f}}$ are evaluated using the following relations:

$$\begin{gathered}\overline{\mathbf{D}}[\mathbf{v}]_{\mathrm{f}} = f_{\mathrm{f}} \mathbf{D}[\mathbf{v}]_{\mathrm{P}} + (1 - f_{\mathrm{f}}) \mathbf{D}[\mathbf{v}]_{\mathrm{F}} \\ \overline{\mathbf{H}}[\mathbf{v}]_{\mathrm{f}} = f_{\mathrm{f}} \mathbf{H}[\mathbf{v}]_{\mathrm{P}} + (1 - f_{\mathrm{f}}) \mathbf{H}[\mathbf{v}]_{\mathrm{F}}\end{gathered} \tag{23}$$

where f_{f} denotes the interpolation factor between the main grid points P and F. Moreover, the term $(\nabla P)_{\mathrm{f}} \cdot \mathbf{S}_{\mathrm{f}}$ is discretized using the method described by Zwart


$\mathbf{S}_f$
f
P
F
$\mathbf{d}_f$
$\hat{\mathbf{n}}$
$\hat{\mathbf{n}} - \gamma\hat{\mathbf{d}}$
$\hat{\mathbf{n}} - \hat{\mathbf{d}}$
$\hat{\mathbf{d}}$
$(\gamma - 1)\hat{\mathbf{d}}$

(*a*)

(*b*)

Figure 2. Typical control-volume faces and geometric nomenclature.

et al. [38], according to which it is decomposed into

$$(\nabla P)_f \cdot \mathbf{S}_f = (\nabla P)_f \cdot \hat{\mathbf{n}}_f \mathbf{S}_f = \left[(\nabla P)_f \cdot \left(\gamma \hat{\mathbf{d}}\right)_f + \left(\overline{\nabla P}\right)_f \cdot \left(\hat{\mathbf{n}}_f - \left(\gamma \hat{\mathbf{d}}\right)_f\right)\right] S_f \tag{24}$$

where $(\overline{\nabla P})_f$ is the average of the adjacent cell pressure gradients $\hat{\mathbf{n}}_f$ and $\hat{\mathbf{d}}_f$ (Figure 2) are the contravariant (surface vector) and covariant (curvilinear coordinate) unit vectors, respectively, and γ is a scaling factor. This factor is chosen such that it is equal to 1 on orthogonal meshes in order for the method to collapse to classical stencils [38, 39]. With that constraint, the expression for γ on structured meshes is

$$\gamma_f = \frac{1}{\hat{\mathbf{n}}_f \cdot \hat{\mathbf{d}}_f} = \frac{S_f d_f}{\mathbf{S}_f \cdot \mathbf{d}_f} \tag{25}$$

defining the space vector $\boldsymbol{\kappa}_f$ as

$$\boldsymbol{\kappa}_f = \left[\hat{\mathbf{n}}_f - \left(\gamma \hat{\mathbf{d}}\right)_f\right] S_f = \kappa_f^x \mathbf{i} + \kappa_f^y \mathbf{j} \tag{26}$$

the expression for $(\nabla P)_f \cdot \mathbf{S}_f$ becomes

$$(\nabla P)_f \cdot \mathbf{S}_f = (\nabla P)_f \cdot (\hat{\mathbf{d}})_f \frac{S_f d_f}{\mathbf{S}_f \cdot \mathbf{d}_f} S_f + \left(\overline{\nabla P}\right)_f \cdot (\kappa_f^x \mathbf{i} + \kappa_f^y \mathbf{j}) \tag{27}$$

In this form, the term $(\nabla P)_f \cdot (\hat{\mathbf{d}})_f$ represents the pressure gradient in the direction of the coordinate line joining P and F (see Figure 2). Therefore, the above equation can be rewritten as

$$(\nabla P)_f \cdot \mathbf{S}_f = \frac{P_F - P_P}{d_f} \frac{S_f d_f}{\mathbf{S}_f \cdot \mathbf{d}_f} S_f + \left(\overline{\nabla P}\right)_f \cdot (\kappa_f^x \mathbf{i} + \kappa_f^y \mathbf{j}) \tag{28}$$

and upon simplifying, it reduces to

$$(\nabla P)_f \cdot \mathbf{S}_f = (P_F - P_P) \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_f} + \left(\overline{\nabla P}\right)_f \cdot (\kappa_f^x \mathbf{i} + \kappa_f^y \mathbf{j}) \tag{29}$$

Substituting the various terms in Eq. (22) by their equivalent expressions, the following is obtained:

$$\begin{aligned} U_f = {} & \left[\overline{H}[u]_f \quad \overline{H}[v]_f\right] \cdot \left[S_f^x \quad S_f^y\right] - \begin{bmatrix} \overline{D}[u]_f & 0 \\ 0 & \overline{D}[v]_f \end{bmatrix} \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_f} (P_F - P_P) \\ & - \begin{bmatrix} \overline{D}[u]_f & 0 \\ 0 & \overline{D}[v]_f \end{bmatrix} \left(\overline{\nabla P}\right)_f \cdot (\kappa_f^x \mathbf{i} + \kappa_f^y \mathbf{j}) \end{aligned} \tag{30}$$

or

$$\begin{aligned} U_f = {} & \left(\overline{H}[u]_f S_f^x + \overline{H}[v]_f S_f^y\right) - \begin{bmatrix} \overline{D}[u]_f & 0 \\ 0 & \overline{D}[v]_f \end{bmatrix} \begin{bmatrix} S_f^x \\ S_f^y \end{bmatrix} \\ & \cdot \left[S_f^x \quad S_f^y\right] \frac{1}{\mathbf{S}_f \cdot \mathbf{d}_f} (P_F - P_P) - \begin{bmatrix} \overline{D}[u]_f & 0 \\ 0 & \overline{D}[v]_f \end{bmatrix} \begin{bmatrix} \left(\overline{\nabla P}\right)_f^x \\ \left(\overline{\nabla P}\right)_f^y \end{bmatrix} \cdot \left[\kappa_f^x \quad \kappa_f^y\right] \end{aligned} \tag{31}$$

Upon expanding, the final form of the interface flux velocity is written as

$$\begin{aligned} U_f = {} & \left(\overline{H}[u]_f S_f^x + \overline{H}[v]_f S_f^y\right) - \frac{\overline{D}[u]_f (S_f^x)^2 + \overline{D}[v]_f (S_f^y)^2}{S_f^x d_f^x + S_f^y d_f^y} (P_F - P_P) \\ & - \left(\overline{D}[u]_f \left(\overline{\nabla P}\right)_f^x \kappa_f^x + \overline{D}[v]_f \left(\overline{\nabla P}\right)_f^y \kappa_f^y\right) \end{aligned} \tag{32}$$

Similar to $\overline{\mathbf{H}}[\mathbf{v}]_f$ and $\overline{\mathbf{D}}_f$, the $\left(\overline{\nabla P}\right)_f$ is calculated from

$$\left(\overline{\nabla P}\right)_f = f_f (\nabla P)_P + (1 - f_f)(\nabla P)_F \tag{33}$$

and the pressure gradient at the main grid point P (or F) is obtained from

$$\begin{aligned}\nabla P_{\mathrm{P}} &= \frac{1}{V}\int_{v}\nabla P\,dV = \frac{1}{V}\oint_{s} P\cdot d\mathbf{S} = \frac{1}{V}(P_{\mathrm{e}}\mathbf{S}_{\mathrm{e}} + P_{\mathrm{w}}\mathbf{S}_{\mathrm{w}} + P_{\mathrm{n}}\mathbf{S}_{\mathrm{n}} + P_{\mathrm{s}}\mathbf{S}_{\mathrm{s}})\\ &= \frac{1}{V}(P_{\mathrm{e}}S^{x}_{\mathrm{e}} + P_{\mathrm{w}}S^{x}_{\mathrm{w}} + P_{\mathrm{n}}S^{x}_{\mathrm{n}} + P_{\mathrm{s}}S^{x}_{\mathrm{s}})\mathbf{i}\\ &\quad + \frac{1}{V}(P_{\mathrm{e}}S^{y}_{\mathrm{e}} + P_{\mathrm{w}}S^{y}_{\mathrm{w}} + P_{\mathrm{n}}S^{y}_{\mathrm{n}} + P_{\mathrm{s}}S^{y}_{\mathrm{s}})\mathbf{j}\\ &= (\nabla P)^{x}_{\mathrm{P}}\mathbf{i} + (\nabla P)^{y}_{\mathrm{P}}\mathbf{j} \qquad (34)\end{aligned}$$

The cell face velocities are thus made dependent on the pressure difference across the face, which helps avoiding the checkerboard problem previously encountered in collocated variable algorithms (Patankar [1]).

The Generalized Pressure and Pressure-Correction Equations

As mentioned before, the convergence in the segregated approach is driven by the corrector stage where a pressure (or a pressure-correction) equation is solved. Therefore, the first step in developing a segregated solution algorithm is to derive such an equation. The key step in the derivation is to note that in the predictor stage a guessed or estimated pressure field from the previous iteration, denoted by $P^{(n)}$, is substituted into the momentum equations. The resulting velocity field, denoted by u^* and v^*, which now satisfies the momentum equations, in general, will not satisfy the continuity equation. Thus, a correction is needed in order to yield velocity and pressure fields that will satisfy both equations. Denoting the corrections with a superscript$'$, the corrected fields are written as

$$\begin{cases} P = P^{(n)} + P' \\ \mathbf{v} = \mathbf{v}^* + \mathbf{v}' \qquad (u = u^* + u',\, v = v^* + v') \\ \rho = \rho^{(n)} + \rho' \end{cases} \qquad (35)$$

where P', $\mathbf{v}'(u', v')$, and ρ' are the pressure, velocity, and density corrections, respectively. Thus, before the pressure is known, the velocities obtained from the solution of the momentum equations are actually u^* and v^* rather than u and v. Hence the equations solved in the predictor stage are

$$\mathbf{v}^*_{\mathrm{P}} - \mathbf{H}[\mathbf{v}^*]_{\mathrm{P}} = -\mathbf{D}_{\mathrm{P}}(\nabla P^{(n)})_{\mathrm{P}} \qquad (36)$$

while the final solution satisfies

$$\mathbf{v}_{\mathrm{P}} - \mathbf{H}[\mathbf{v}]_{\mathrm{P}} = -\mathbf{D}_{\mathrm{P}}(\nabla P)_{\mathrm{P}} \qquad (37)$$

Subtracting the two sets of equations (37) and (36) from each other yields the following equation involving the correction terms:

$$\mathbf{v}'_{\mathrm{P}} - \mathbf{H}[\mathbf{v}']_{\mathrm{P}} = -\mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \qquad (38)$$

The velocity and density fields should be corrected to satisfy mass conservation. Therefore, the new density and velocity fields, ρ and $\mathbf{v}$, will satisfy the continuity equation if

$$\frac{(\rho_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[\rho \mathbf{v} \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{39}$$

Linearizing the $(\rho \mathbf{v})$ term, one gets

$$(\rho^* + \rho')(\mathbf{v}^* + \mathbf{v}') = \rho^* \mathbf{v}^* + \rho^* \mathbf{v}' + \rho' \mathbf{v}^* + \rho' \mathbf{v}' \tag{40}$$

Alternatively, the $(\rho \mathbf{v})$ term can be written as

$$(\rho^* + \rho')(\mathbf{v}^* + \mathbf{v}') = \rho \mathbf{v}^* + \rho^* \mathbf{v} - \rho^* \mathbf{v}^* + \rho' \mathbf{v}' \tag{41}$$

Substitution of Eq. (41) into Eq. (39) gives

$$\frac{(\rho_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[(\rho \mathbf{v}^* + \rho^* \mathbf{v} - \rho^* \mathbf{v}^* + \rho' \mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{42}$$

or

$$\frac{V}{\delta t} \rho_{\mathrm{P}} + \Delta[(\rho \mathbf{v}^* + \rho^* \mathbf{v}) \cdot \mathbf{S}]_{\mathrm{P}} = \frac{V}{\delta t} \rho_{\mathrm{P}}^{o} + \Delta[(\rho^* \mathbf{v}^*) \cdot \mathbf{S}]_{\mathrm{P}} - \Delta[(\rho' \mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \tag{43}$$

Using Eq. (37), the above equation may be rewritten as

$$\begin{aligned} &\frac{V}{\delta t} \rho_{\mathrm{P}} + \Delta[U^* \rho]_{\mathrm{P}} + \Delta[\rho^* (\mathbf{H}[\mathbf{v}] - \mathbf{D}(\nabla P)) \cdot \mathbf{S}]_{\mathrm{P}} \\ &\quad = \frac{V}{\delta t} \rho_{\mathrm{P}}^{o} + \Delta[\rho^* U^*]_{\mathrm{P}} - \Delta[(\rho' \mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \end{aligned} \tag{44}$$

After rearranging, one gets

$$\begin{aligned} &\frac{V}{\delta t} \rho_{\mathrm{P}} + \Delta[U^* \rho]_{\mathrm{P}} - \Delta[\rho^* \mathbf{D}(\nabla P) \cdot \mathbf{S}]_{\mathrm{P}} \\ &\quad = -\Delta[\rho^* (\mathbf{H}[\mathbf{v}]) \cdot \mathbf{S}]_{\mathrm{P}} + \frac{V}{\delta t} \rho_{\mathrm{P}}^{o} + \Delta[\rho^* \mathbf{U}^*]_{\mathrm{P}} - \Delta[(\rho' \mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \end{aligned} \tag{45}$$

Substituting for density using the equation of state, the pressure equation becomes

$$\begin{aligned} &\frac{V C_p}{\delta t} P_{\mathrm{P}} + \Delta[C_p U^* P]_{\mathrm{P}} - \Delta[\rho^* \mathbf{D}(\nabla P) \cdot \mathbf{S}]_{\mathrm{P}} \\ &\quad = \frac{V}{\delta t} \rho_{\mathrm{P}}^{o} + \Delta[\rho^* U^*]_{\mathrm{P}} - \Delta[\rho^* (\mathbf{H}[\mathbf{v}^*]) \cdot \mathbf{S}]_{\mathrm{P}} \\ &\qquad - \Delta[\rho^* (\mathbf{H}[\mathbf{v}']) \cdot \mathbf{S}]_{\mathrm{P}} - \Delta[(\rho' \mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \end{aligned} \tag{46}$$

Alternatively, the above equation may be written for pressure correction instead of pressure by simply substituting for $(\rho\mathbf{v})$ in Eq. (39) using Eq. (40). By so doing, the continuity equation can be written as

$$\frac{(\rho_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[(\rho^{*}\mathbf{v}^{*} + \rho^{*}\mathbf{v}' + \rho'\mathbf{v}^{*} + \rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{47}$$

Rearranging, the following equation is obtained:

$$\frac{V}{\delta t}(\rho_{\mathrm{P}}^{*} + \rho_{\mathrm{P}}') + \Delta[(\rho'\mathbf{v}^{*} + \rho^{*}\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} = \frac{V}{\delta t}\rho_{\mathrm{P}}^{o} - \Delta[(\rho^{*}\mathbf{v}^{*}) \cdot \mathbf{S}]_{\mathrm{P}} - \Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \tag{48}$$

Using Eq. (38), the above equation becomes

$$\begin{aligned} &\frac{V}{\delta t}\rho_{\mathrm{P}}' + \Delta[U^{*}\rho' + \rho^{*}(\mathbf{H}[\mathbf{v}'] - \mathbf{D}(\nabla P')) \cdot \mathbf{S}]_{\mathrm{P}} \\ &\quad = -\frac{(\rho_{\mathrm{P}}^{*} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta[(\rho^{*}\mathbf{v}^{*}) \cdot \mathbf{S}]_{\mathrm{P}} - \Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \end{aligned} \tag{49}$$

Finally, substituting density correction by pressure correction, as obtained from the equation of state, the final form of the pressure-correction equation is

$$\begin{aligned} &\frac{VC_{p}}{\delta t}P_{\mathrm{P}}' + \Delta[C_{p}U^{*}P']_{\mathrm{P}} - \Delta[\rho^{*}\mathbf{D}(\nabla P') \cdot \mathbf{S}]_{\mathrm{P}} \\ &\quad = -\frac{(\rho_{\mathrm{P}}^{*} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta[\rho^{*}U^{*}]_{\mathrm{P}} - \Delta[\rho^{*}\mathbf{H}[\mathbf{v}'] \cdot \mathbf{S}]_{\mathrm{P}} \\ &\quad\quad - \Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \end{aligned} \tag{50}$$

From the above, the one-to-one correspondence between the pressure and pressure-correction equations is obvious, and as such, the two equations are interchangeable. Thus, algorithms in which the pressure field is obtained using a pressure equation (e.g., SIMPLER, PISO, SIMPLEM, . . .), may equally be calculated from a pressure-correction equation. Moreover, the second-order correction term $\rho'\mathbf{v}'$ is usually neglected. This does not affect either the convergence rate (i.e., it is considerably smaller than other terms) or the final solution, since at the state of convergence the correction fields vanish. For this reason, it is neglected in all subsequent derivations. Furthermore, if the $\mathbf{H}[\mathbf{v}']$ term in the above equation is retained, there will result a pressure-correction equation relating the pressure-correction value at a point to all values in the domain. Such an equation ensures that the corrected fields will satisfy the continuity and momentum equations, but is undesirable because it becomes intractable. To facilitate implementation and reduce cost, simplifying assumptions related to this term have been introduced. Depending on these assumptions, different algorithms are obtained.

THE SIMPLE ALGORITHM AND ITS VARIANTS

Having derived the necessary equation for the calculation of the pressure field (i.e., the pressure and/or pressure-correction equation), the SIMPLE [1, 2], SIMPLER [1], SIMPLEC [30], SIMPLEST [31], SIMPLEX [32], SIMPLEM [33], PISO [34], PRIME [35], SIMPLESSEC [37], and SIMPLESSE [37, 43] iterative algorithms that were developed to solve the coupled system of continuity and momentum equations can now be presented. In the derivations to follow, the superscripts *o* and *n* denote values from the previous time step and values from the previous iteration, respectively. Moreover, the superscripts *, **, ***, and **** represent the first, second, third, and fourth updated values at the current iteration, respectively.

The purpose in the various algorithms is to arrive at a velocity field that satisfies both the momentum and continuity equations. Moreover, as shown above, the pressure or pressure-correction equation in all algorithms is derived by combining the following momentum and continuity equations:

$$\mathbf{v}_\mathrm{P} = \mathbf{H}[\mathbf{v}]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P)_\mathrm{P} \tag{51}$$

$$\frac{(\rho_\mathrm{P} - \rho_\mathrm{P}^o)}{\delta t} V + \Delta[\rho \mathbf{v} \cdot \mathbf{S}]_\mathrm{P} = 0 \tag{52}$$

in addition to the equation of state given by

$$\rho = C_p P \tag{53}$$

One then may wonder about the differences among the various algorithms. The key answer is in the different approximations to the $\mathbf{H}[\mathbf{v}']$ terms and the various sequences of operations that take place.

The SIMPLE Algorithm

The SIMPLE algorithm of Patankar and Spalding [1, 2] has probably been the most widely used method in the last three decades. The method's success in handling difficult problems in heat transfer and fluid flow has led to its evolution into a whole family of related methods. The SIMPLE algorithm consists of two stages: a predicator stage and a corrector stage, as described below.

The SIMPLE algorithm: Symbolic form

Predictor:

$$\mathbf{v}_\mathrm{P}^* = \mathbf{H}[\mathbf{v}^*]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla \mathbf{P}^{(n)})_\mathrm{P} \tag{54}$$

Corrector:

$$(v', P', \rho')(\mathbf{v}^{**} = \mathbf{v}^* + \mathbf{v}', P^* = P^{(n)} + P', \rho^* = \rho^{(n)} + \rho') \tag{55}$$

$$\therefore \mathbf{v}_\mathrm{P}^{**} = \mathbf{H}[\mathbf{v}^{**}]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P^*)_\mathrm{P} = \mathbf{H}[\mathbf{v}^* + \mathbf{v}']_\mathrm{P} - \mathbf{D}_\mathrm{P}[\nabla(P^{(n)} + P')]_\mathrm{P} \tag{56}$$

$$\therefore \begin{cases} v_\mathrm{P}' = \mathbf{H}[\mathbf{v}']_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P')_\mathrm{P} \\ \rho' = C_p P' \end{cases} \tag{57}$$

Condition:

$$\frac{(\rho_{\mathrm{P}}^{*} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[\rho^{*}\mathbf{v}^{**} \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{58}$$

$$\therefore \frac{(\rho_{\mathrm{P}}^{(n)} + \rho_{\mathrm{P}}' - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[(\rho^{(n)} + \rho')(\mathbf{v}^{*} + \mathbf{H}[\mathbf{v}'] - \mathbf{D}(\nabla P')) \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{59}$$

$$\begin{aligned} \therefore \frac{VC_p}{\delta t} P_{\mathrm{P}}' + \Delta[C_p U^{*} P']_{\mathrm{P}} - \Delta[\rho^{(n)}\mathbf{D}(\nabla P') \cdot \mathbf{S}]_{\mathrm{P}} &= -\frac{(\rho_{\mathrm{P}}^{(n)} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta[\rho^{(n)} U^{*}]_{\mathrm{P}} \\ &\quad - \Delta[\rho^{(n)}\mathbf{H}[\mathbf{v}'] \cdot \mathbf{S}]_{\mathrm{P}} - \Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \end{aligned} \tag{60}$$

Approximation: Neglect: $\mathbf{H}[v']$, $\Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}}$.

$$\Rightarrow \mathbf{v}_{\mathrm{P}}' = -\mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \tag{61}$$

Approximate equation:

$$\begin{aligned} \Rightarrow \frac{VC_p}{\delta t} P_{\mathrm{P}}' + \Delta[C_p U^{*} P']_{\mathrm{P}} - \Delta[\rho^{(n)}\mathbf{D}(\nabla P') \cdot \mathbf{S}]_{\mathrm{P}} &= -\frac{(\rho_{\mathrm{P}}^{(n)} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta[\rho^{(n)} U^{*}]_{\mathrm{P}} \end{aligned} \tag{62}$$

Discussion. Since at the state of convergence the pressure- and velocity-correction fields are zero, the approximation introduced does not have any effect on the final solution. Rather, it affects the convergence behavior. Moreover, had this term been retained, the pressure correction at any point would have been related to the pressure correction at all grid points in the domain. The resulting exact pressure-correction equation would have had a full coefficient matrix, and its solution would have required an undesirable, more expensive, direction solution method. However, because the $\mathbf{H}[\mathbf{v}']$ terms are dropped, the predicted pressure-correction field is overestimated and the corrected velocity field no longer satisfies the momentum equations. Therefore, in order to avoid divergence, the pressure field is underrelaxed according to:

$$P = P^{*} + \alpha_p P' \tag{63}$$

which slows down the convergence rate of the iteration process (α_p being the underrelaxation factor for pressure).

A global SIMPLE iteration

Solve implicitly for u and v, using the old pressure and density fields.
Calculate the $\mathbf{D}$ field.

Solve the pressure-correction equation.
Correct u, v, P, and ρ.
Solve implicitly the energy equation and update the density field.
Return to the first step and iterate until convergence.

The SIMPLEC Algorithm

Experience has shown that the rate of convergence of the SIMPLE algorithm is greatly dependent on the proper choice of the underrelaxation factors for the velocity components and the pressure. The optimum values for these variables are usually problem dependent. The SIMPLEC algorithm of Van Dormaal and Raithby [30] was developed with the intention of alleviating the aforementioned problem through a better approximation to the $\mathbf{H}[\mathbf{v}']$ so as to eliminate the need for underrelaxing the pressure field. The steps involved in the SIMPLEC algorithm are outlined below.

The SIMPLEC algorithm: Symbolic form

Predictor:

$$\mathbf{v}_{\mathrm{P}}^{*} = \mathbf{H}[\mathbf{v}^{*}]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{(n)})_{\mathrm{P}} \tag{64}$$

Corrector:

$$(\mathbf{v}', P', \rho')(\mathbf{v}^{**} = \mathbf{v}^{*} + \mathbf{v}', P^{*} = P^{(n)} + P', \rho^{*} = \rho^{(n)} + \rho') \tag{65}$$

$$\therefore \mathbf{v}_{\mathrm{P}}^{**} = \mathbf{H}[\mathbf{v}^{**}]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{*})_{\mathrm{P}} = \mathbf{H}[\mathbf{v}^{*} + \mathbf{v}']_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}[\nabla(P^{(n)} + P')]_{\mathrm{P}} \tag{66}$$

$$\therefore \mathbf{v}_{\mathrm{P}}' = \mathbf{H}[\mathbf{v}']_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \tag{67}$$

Subtracting $\tilde{\mathbf{H}}[1]_{\mathrm{P}}\mathbf{v}_{\mathrm{P}}'$ from both sides, one gets

$$\mathbf{v}_{\mathrm{P}}' - \tilde{\mathbf{H}}[\mathbf{1}]_{\mathrm{P}}\mathbf{v}_{\mathrm{P}}' = \mathbf{H}[\mathbf{v}']_{\mathrm{P}} - \tilde{\mathbf{H}}[\mathbf{1}]_{\mathrm{P}}\mathbf{v}_{\mathrm{P}}' - \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \tag{68}$$

$$\therefore (1 - \tilde{\mathbf{H}}[\mathbf{1}]_{\mathrm{P}})\mathbf{v}_{\mathrm{P}}' = \mathbf{H}[\mathbf{v}' - v_{\mathrm{P}}']_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \tag{69}$$

$$\therefore \begin{cases} \mathbf{v}_{\mathrm{P}}' = \dfrac{\mathbf{H}[\mathbf{v}' - v_{\mathrm{P}}']_{\mathrm{P}}}{(1 - \tilde{\mathbf{H}}[\mathbf{1}]_{\mathrm{P}})} - \dfrac{\mathbf{D}_{\mathrm{P}}}{(1 - \tilde{\mathbf{H}}[\mathbf{1}]_{\mathrm{P}})}(\nabla P')_{\mathrm{P}} \\ \rho' = C_{p} P' \end{cases} \tag{70}$$

Condition:

$$\frac{(\rho_{\mathrm{P}}^{*} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[\rho^{*}\mathbf{v}^{**} \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{71}$$

$$\therefore \frac{(\rho_{\mathrm{P}}^{(n)} + \rho'_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta\left[(\rho^{(n)} + \rho')\left(v^* + \frac{\mathbf{H}[\mathbf{v}' - \mathbf{v}'_{\mathrm{P}}]}{1 - \tilde{\mathbf{H}}[1]} - \frac{\mathbf{D}}{1 - \tilde{\mathbf{H}}[1]}(\nabla \mathbf{P}')\right)\cdot \mathbf{S}\right]_{\mathrm{P}} = 0 \quad (72)$$

$$\therefore \frac{VC_p}{\delta t} P'_{\mathrm{P}} + \Delta[C_p U^* P']_{\mathrm{P}} + \Delta\left[\rho^{(n)}\left(-\frac{\mathbf{D}}{1 - \tilde{\mathbf{H}}[1]}(\nabla P')\right)\cdot \mathbf{S}\right]_{\mathrm{P}}$$

$$= -\frac{(\rho_{\mathrm{P}}^{(n)} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta[\rho^{(n)} U^*]_{\mathrm{P}} - \Delta\left[\rho^{(n)}\left(\frac{\mathbf{H}[\mathbf{v}' - \mathbf{v}'_{\mathrm{P}}]}{1 - \tilde{\mathbf{H}}[1]}\right)\cdot \mathbf{S}\right]_{\mathrm{P}} - \Delta[(\rho'\mathbf{v}')\cdot \mathbf{S}]_{\mathrm{P}} \quad (73)$$

Approximation: Neglect: $\mathbf{H}[\mathbf{v}' - \mathbf{v}'_{\mathrm{P}}]$, $\Delta[(\rho'\mathbf{v}')\cdot \mathbf{S}]_{\mathrm{P}}$.

$$\Rightarrow \mathbf{v}'_{\mathrm{P}} = -\tilde{\mathbf{D}}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \quad (74)$$

Approximate equation:

$$\Rightarrow \frac{VC_p}{\delta t} P'_{\mathrm{P}} + \Delta[C_p U^* P']_{\mathrm{P}} - \Delta\left[\rho^{(n)}\tilde{\mathbf{D}}(\nabla P')\cdot \mathbf{S}\right]_{\mathrm{P}} = -\frac{(\rho_{\mathrm{P}}^{(n)} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta[\rho^{(n)} U^*]_{\mathrm{P}} \quad (75)$$

Discussion. Due to a better approximation in SIMPLEC (i.e., neglecting $\mathbf{H}[\mathbf{v}' - \mathbf{v}'_{\mathrm{P}}]$ rather than $\mathbf{H}[\mathbf{v}']$), the relaxation of pressure becomes unnecessary and, as compared to SIMPLE, the resulting velocity corrections will satisfy the momentum equations better. Consequently, a higher rate of convergence is obtained. The sequence of steps being the same, the only difference between SIMPLE and SIMPLEC is in the definition of the coefficients in the pressure-correlation equation.

A global SIMPLEC iteration

Solve implicitly for u and v, using the old pressure and density fields.
Calculate the $\tilde{\mathbf{D}}$ field.
Solve the pressure-correction equation.
Correct u, v, P, and ρ.
Solve implicitly the energy equation and update the density field.
Return to the first step and iterate until convergence.

The PRIME Algorithm

In the PRIME algorithm [35], the momentum equations are solved explicitly. This explicit treatment of the momentum equations is justified by the small contribution to the convergence of the entire flow field by the iterative sweeps within each momentum equation. On the other hand, finding the correct solution for the pressure field represents the most important factor in the overall convergence. Based on this argument, the PRIME algorithm can be summarized as follows.

The PRIME algorithm: Symbolic form

Predictor:

$$\mathbf{v}_{\mathrm{P}}^{*} = \mathbf{H}\left[\mathbf{v}^{(n)}\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{(n)})_{\mathrm{P}}$$

Corrector:

$$(\mathbf{v}', P', \rho')(\mathbf{v}^{**} = \mathbf{v}^{*} + \mathbf{v}', P^{*} = P^{(n)} + P', \rho^{*} = \rho^{(n)} + \rho') \tag{76}$$

$$\therefore \mathbf{v}_{\mathrm{P}}^{**} = \mathbf{H}\left[\mathbf{v}^{**}\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{*})_{\mathrm{P}} = \mathbf{H}\left[\mathbf{v}^{*} + \mathbf{v}'\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}\left[\nabla(P^{(n)} + P')\right]_{\mathrm{P}} \tag{77}$$

$$\mathbf{v}_{\mathrm{P}}^{*} = \mathbf{H}\left[\mathbf{v}^{(n)}\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{(n)})_{\mathrm{P}} \tag{78}$$

$$\therefore \begin{cases} \mathbf{v}_{\mathrm{P}}' = \mathbf{H}\left[\mathbf{v}^{*} - \mathbf{v}^{(n)}\right]_{\mathrm{P}} + \mathbf{H}\left[\mathbf{v}'\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \\ \rho' = C_p P' \end{cases} \tag{79}$$

Condition:

$$\frac{(\rho_{\mathrm{P}}^{*} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta\left[\rho^{*}\mathbf{v}^{**} \cdot \mathbf{S}\right]_{\mathrm{P}} = 0 \tag{80}$$

$$\begin{aligned} \therefore \frac{(\rho_{\mathrm{P}}^{(n)} + \rho_{\mathrm{P}}' - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta\big[(\rho^{(n)} + \rho')(\mathbf{v}^{*} + \mathbf{H}\left[\mathbf{v}^{*} - \mathbf{v}^{(n)}\right] \\ + \mathbf{H}\left[\mathbf{v}'\right] - \mathbf{D}(\nabla P')) \cdot \mathbf{S}\big]_{\mathrm{P}} = 0 \end{aligned} \tag{81}$$

$$\begin{aligned} \therefore \frac{VC_p}{\delta t} P_{\mathrm{P}}' + \Delta\left[C_p U^{*} P'\right]_{\mathrm{P}} - \nabla\left[\rho^{(n)}\mathbf{D}(\nabla P') \cdot \mathbf{S}\right]_{\mathrm{P}} \\ = -\frac{(\rho_{\mathrm{P}}^{(n)} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta\left[\rho^{(n)} U^{*}\right]_{\mathrm{P}} \\ - \Delta\left[(\rho^{(n)})(\mathbf{H}\left[\mathbf{v}^{*} - \mathbf{v}^{(n)}\right] + \mathbf{H}\left[\mathbf{v}'\right]) \cdot \mathbf{S}\right]_{\mathrm{P}} - \Delta\left[(\rho'\mathbf{v}') \cdot \mathbf{S}\right]_{\mathrm{P}} \end{aligned} \tag{82}$$

Approximation: Neglect: $\mathbf{H}\left[\mathbf{v}^{*} - \mathbf{v}^{(n)}\right]_{\mathrm{P}}, \mathbf{H}\left[\mathbf{v}'\right]_{\mathrm{P}}, \Delta\left[(\rho'\mathbf{v}') \cdot \mathbf{S}\right]_{\mathrm{P}}$.

Approximate equation:

$$\begin{aligned} \Rightarrow \frac{VC_p}{\delta t} P_{\mathrm{P}}' + \Delta\left[C_p U^{*} P'\right]_{\mathrm{P}} - \Delta\left[\rho^{(n)}\mathbf{D}(\nabla P') \cdot \mathbf{S}\right]_{\mathrm{P}} \\ = -\frac{(\rho_{\mathrm{P}}^{(n)} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta\left[\rho^{(n)} U^{*}\right]_{\mathrm{P}} \end{aligned} \tag{83}$$

Discussion. The terms neglected in PRIME $(\mathbf{H}[\mathbf{v}']_{\mathrm{P}} + \mathbf{H}[\mathbf{v}^* - \mathbf{v}^{(n)}]_{\mathrm{P}})$ can become smaller than the term neglected in SIMPLE $(\mathbf{H}[\mathbf{v}']_{\mathrm{P}})$ if $\mathbf{H}[\mathbf{v}']_{\mathrm{P}}$ and $\mathbf{H}[\mathbf{v}^* - \mathbf{v}^{(n)}]_{\mathrm{P}}$ are of opposite signs. It is worth noting that $\mathbf{H}[\mathbf{v}']_{\mathrm{P}} = \mathbf{H}[\mathbf{v}^{**} - \mathbf{v}^*]_{\mathrm{P}}$ is a correction to satisfy continuity, while $\mathbf{H}[\mathbf{v}^* - \mathbf{v}^{(n)}]_{\mathrm{P}}$ is a correction to satisfy momentum. Usually the corrector added to satisfy momentum is opposite to that added to satisfy continuity and hence, the neglected term $(\mathbf{H}[\mathbf{v}']_{\mathrm{P}} + \mathbf{H}[\mathbf{v}^* - \mathbf{v}^{(n)}]_{\mathrm{P}})$ is smaller. Another way to look at these terms is to consider $\mathbf{v}^{**}$ as $\mathbf{v}^{(n+1)}$. Then, $\mathbf{H}[\mathbf{v}']_{\mathrm{P}}$ will be equal to $\mathbf{H}[\mathbf{v}^{(n+1)} - \mathbf{v}^*]_{\mathrm{P}}$ and will have a sign opposite to that of $\mathbf{H}[\mathbf{v}^* - \mathbf{v}^{(n)}]_{\mathrm{P}}$. Moreover, since the momentum equations are solved explicitly, no underrelaxation is required. This has the advantage of increasing the stability of the algorithm.

A global PRIME iteration

Solve explicitly for u and v, using the old pressure and density fields.
Calculate the $\mathbf{D}$ field.
Solve the pressure-correction equation.
Correct u, v, P, and ρ.
Solve implicitly the energy equation and update the density field.
Return to the first step and iterate until convergence.

The SIMPLEST Algorithm

It is well known that disturbances in a pure diffusion situation are propagated instantaneously in all directions, but their amplitude decays rapidly. This is equivalent to propagation of errors throughout the entire solution domain, in a single iteration by implicit solution methods. On the other hand, disturbances in a pure convection situation are propagated, in the flow direction, at a finite speed without any change in their magnitude. This is similar to propagation of error, from a particular point to the neighboring grid points, in a single iteration of explicit iterative methods.

Based on this physical argument, Spalding [31] developed the SIMPLEST method, in which the implicit treatment of the diffusion terms in the general algebraic equation is combined with an explicit treatment of the convection terms. For that purpose, the coefficients in the momentum equations are first separated into their diffusion and convection parts as

$$\mathbf{H}[\mathbf{v}]_{\mathrm{P}} = \mathbf{H}^{\mathrm{D}}[\mathbf{v}]_{\mathrm{P}} + \mathbf{H}^{\mathrm{C}}[\mathbf{v}]_{\mathrm{P}} \tag{84}$$

Thus,

$$\mathbf{v}_{\mathrm{P}} = \mathbf{H}^{\mathrm{D}}[\mathbf{v}]_{\mathrm{P}} + \mathbf{H}^{\mathrm{C}}[\mathbf{v}]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P)_{\mathrm{P}} \tag{85}$$

The remaining steps are as follows.

The SIMPLEST algorithm: Symbolic form

Predictor:

$$\mathbf{v}^*_{\mathrm{P}} = \mathbf{H}^{\mathrm{D}}[\mathbf{v}^*]_{\mathrm{P}} + \mathbf{H}^{\mathrm{C}}[\mathbf{v}^{(n)}]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{(n)})_{\mathrm{P}} \tag{86}$$

Corrector:

$$(\mathbf{v}', P', \rho')(\mathbf{v}^{**} = \mathbf{v}^* + \mathbf{v}', \mathbf{P}^* = \mathbf{P}^{(n)} + P', \rho^* = \rho^{(n)} + \rho') \tag{87}$$

$$\mathbf{v}_\mathrm{P}^{**} = \mathbf{H}^\mathrm{D}[\mathbf{v}^{**}]_\mathrm{P} + \mathbf{H}^\mathrm{C}[\mathbf{v}^{**}]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P^*)_\mathrm{P}$$

$$\therefore = \mathbf{H}^\mathrm{D}[\mathbf{v}^* + \mathbf{v}']_\mathrm{P} + \mathbf{H}^\mathrm{C}[\mathbf{v}^* + \mathbf{v}']_\mathrm{P} - \mathbf{D}_\mathrm{P}[\nabla(P^{(n)} + P')]_\mathrm{P}$$

$$= \mathbf{H}^\mathrm{D}[\mathbf{v}^*]_\mathrm{P} + \mathbf{H}^\mathrm{D}[\mathbf{v}']_\mathrm{P} + \mathbf{H}^\mathrm{C}[\mathbf{v}^*]_\mathrm{P} + \mathbf{H}^\mathrm{C}[\mathbf{v}']_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P^{(n)})_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P')_\mathrm{P} \tag{88}$$

$$\therefore \begin{cases} \mathbf{v}'_\mathrm{P} = \mathbf{H}[\mathbf{v}']_\mathrm{P} + \mathbf{H}^\mathrm{C}[\mathbf{v}^* - \mathbf{v}^{(n)}]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P')_\mathrm{P} \\ \rho' = C_p P' \end{cases} \tag{89}$$

Condition:

$$\frac{(\rho_\mathrm{P}^* - \rho_\mathrm{P}^o)}{\Delta t} V + \Delta[\rho^* \mathbf{v}^{**} \cdot \mathbf{S}]_\mathrm{P} = 0 \tag{90}$$

$$\therefore \frac{(\rho_\mathrm{P}^{(n)} + \rho'_\mathrm{P} - \rho_\mathrm{P}^o)}{\delta t} V + \Delta[(\rho^{(n)} + \rho')(\mathbf{v}^* + \mathbf{H}[\mathbf{v}']$$

$$+ \mathbf{H}^\mathrm{C}[\mathbf{v}^* - \mathbf{v}^{(n)}] - \mathbf{D}(\nabla P')) \cdot \mathbf{S}]_\mathrm{P} = 0 \tag{91}$$

$$\therefore \frac{VC_p}{\delta t} P'_\mathrm{P} + \Delta[C_p U^* P']_\mathrm{P} - \Delta[\rho^{(n)} \mathbf{D}(\nabla P') \cdot \mathbf{S}]_\mathrm{P}$$

$$= -\frac{(\rho_\mathrm{P}^{(n)} - \rho_\mathrm{P}^o)}{\delta t} V - \Delta[\rho^{(n)} U^*]_\mathrm{P}$$

$$- \Delta[\rho^{(n)}(\mathbf{H}[\mathbf{v}'] + \mathbf{H}^\mathrm{C}[\mathbf{v}^* - \mathbf{v}^{(n)}]) \cdot \mathbf{S}]_\mathrm{P} - \Delta[(\rho' \mathbf{v}') \cdot \mathbf{S}]_\mathrm{P} \tag{92}$$

Approximation: Neglect: $\mathbf{H}[\mathbf{v}']_\mathrm{P}$, $\mathbf{H}^\mathrm{C}[\mathbf{v}^* - \mathbf{v}^{(n)}]_\mathrm{P}$, $\Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_\mathrm{P}$.

$$\Rightarrow \mathbf{v}'_\mathrm{P} = -\mathbf{D}_\mathrm{P}(\nabla P')_\mathrm{P} \tag{93}$$

Approximate equation:

$$\Rightarrow \frac{VC_p}{\delta t} P'_\mathrm{P} + \Delta[C_p U^* P']_\mathrm{P} - \Delta[\rho^{(n)} \mathbf{D}(\nabla P') \cdot \mathbf{S}]_\mathrm{P}$$

$$= -\frac{(\rho_\mathrm{P}^{(n)} - \rho_\mathrm{P}^o)}{\delta t} V - \Delta[\rho^{(n)} U^*]_\mathrm{P} \tag{94}$$

Discussion. Similar to PRIME, the terms neglected in SIMPLEST ($\mathbf{H}[\mathbf{v}']_\mathrm{P} + \mathbf{H}^\mathrm{C}[\mathbf{v}^* - \mathbf{v}^{(n)}]_\mathrm{P}$) can become smaller than the term neglected in SIMPLE ($\mathbf{H}[\mathbf{v}']_\mathrm{P}$) if $\mathbf{H}[\mathbf{v}']_\mathrm{P}$ and $\mathbf{H}^\mathrm{C}[\mathbf{v}^* - \mathbf{v}^{(n)}]_\mathrm{P}$ are of opposite signs. As mentioned earlier, this should be the case since $\mathbf{H}[\mathbf{v}']_\mathrm{P} = \mathbf{H}[\mathbf{v}^{**} - \mathbf{v}^*]_\mathrm{P}$ is a correction to satisfy continuity, while $\mathbf{H}^\mathrm{C}[\mathbf{v}^* - \mathbf{v}^{(\mathrm{n})}]_\mathrm{P}$ is a correction to satisfy momentum, and these usually work in opposite directions.

A good understanding of the SIMPLEST algorithm is obtained by considering the following two limiting situations. The first is when the Reynolds number approaches zero, in which case convection becomes negligible, i.e.,

$$\mathbf{H}^{\mathrm{C}}[\mathbf{v}]_{\mathrm{P}} \to 0 \Rightarrow \mathbf{H}^{\mathrm{D}}[\mathbf{v}]_{\mathrm{P}} \to \mathbf{H}[\mathbf{v}]_{\mathrm{P}} \tag{95}$$

and the method becomes identical to SIMPLE. The second limiting situation is when the value of the Reynolds number is very large so that diffusion is negligible in comparison with convection, i.e.,

$$\mathbf{H}^{\mathrm{D}}[\mathbf{v}]_{\mathrm{P}} \to 0 \Rightarrow \mathbf{H}^{\mathrm{C}}[\mathbf{v}]_{\mathrm{P}} \to \mathbf{H}[\mathbf{v}]_{\mathrm{P}} \tag{96}$$

In this case, the momentum equations are solved, as in PRIME, via the explicit Jacobi method [40], using old information from the previous iteration and eliminating the need for any underrelaxation. Thus, the SIMPLEST algorithm can be seen to be a compromise between SIMPLE and PRIME.

A global SIMPLEST iteration

Solve for u and v, treating $H^{\mathrm{D}}[v]$ implicitly and $H^{\mathrm{c}}[v]$ explicitly.
Calculate the **D** field.
Solve the pressure-correction equation.
Correct u, v, P, and ρ.
Solve implicitly the energy equation and update the density field.
Return to the first step and iterate until convergence.

The SIMPLER Algorithm

Although the pressure-correction field correctly drives the velocity field toward satisfying the continuity equation, it provides poor approximations to the correct pressure field. The reason is that, once the velocity field is updated using the predicted pressure-correction field, it no longer satisfies the momentum equations due to the approximations made for the velocity-correction formulas. Realizing this, Patankar [1] suggested that the pressure-correction field be used only to correct the velocity field to make it satisfy the continuity equation, and that the pressure be calculated from another equation to match the velocities, so that the momentum equations are also satisfied. The resulting algorithm is denoted by SIMPLER [1] (SIMPLE-Revised) and, as originally presented, the additional equation involved P as its variable. However, since it was shown that the pressure and the pressure-correction equations are equivalent, it will be presented here based on pressure correction. The various steps in the algorithm are as outlined below.

The SIMPLER algorithm: Symbolic form

First predictor. No predictor stage. Only coefficients of the momentum equations are calculated.

First corrector

$$(\mathbf{v}', P', \rho')(\mathbf{v}^* = \mathbf{v}^{(n)} + \mathbf{v}', P^* = P^{(n)} + P', \rho^* = \rho^{(n)} + \rho') \tag{97}$$

$$\therefore \mathbf{v}_\mathrm{P}^* = \mathbf{H}[\mathbf{v}^*]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P^*)_\mathrm{P} \tag{98}$$

$$\mathbf{v}_\mathrm{P}^{(n)} = \mathbf{H}[\mathbf{v}^{(n)}]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P^{(n)})_\mathrm{P} \tag{99}$$

$$\therefore \begin{cases} \mathbf{v}_\mathrm{P}' = \mathbf{H}[\mathbf{v}']_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P')_\mathrm{P} \\ \rho' = C_p P' \end{cases} \tag{100}$$

Condition:

$$\frac{(\rho_\mathrm{P}^* - \rho_\mathrm{P}^o)}{\delta t} V + \Delta[\rho^* \mathbf{v}^* \cdot \mathbf{S}]_\mathrm{P} = 0 \tag{101}$$

$$\therefore \frac{(\rho_\mathrm{P}^{(n)} + \rho_\mathrm{P}' - \rho_\mathrm{P}^o)}{\delta t} V + \Delta[(\rho^{(n)} + \rho')(\mathbf{v}^{(n)} + \mathbf{H}[\mathbf{v}'] - \mathbf{D}(\nabla P')) \cdot \mathbf{S}]_\mathrm{P} = 0 \tag{102}$$

$$\begin{aligned} \therefore \frac{VC_p}{\delta t} P_\mathrm{P}' + \Delta\Big[C_p U^{(n)} P'\Big]_\mathrm{P} - \Delta[\rho^{(n)} \mathbf{D}(\nabla P') \cdot \mathbf{S}]_\mathrm{P} \\ = -\frac{(\rho_\mathrm{P}^{(n)} - \rho_\mathrm{P}^o)}{\delta t} V - \Delta[\rho^{(n)} U^{(n)}]_\mathrm{P} \\ - \Delta[\rho^{(n)} \mathbf{H}[\mathbf{v}'] \cdot \mathbf{S}]_\mathrm{P} - \Delta[(\rho' \mathbf{v}') \cdot \mathbf{S}]_\mathrm{P} \end{aligned} \tag{103}$$

Approximation: Neglect: $\mathbf{H}[\mathbf{v}']$, $\Delta[(\rho' \mathbf{v}') \cdot \mathbf{S}]_\mathrm{P}$.
Approximate equation:

$$\begin{aligned} \Rightarrow \frac{VC_p}{\delta t} P_\mathrm{P}' + \Delta\Big[C_p U^{(n)} P'\Big]_\mathrm{P} - \Delta[\rho^{(n)} \mathbf{D}(\nabla P') \cdot \mathbf{S}]_\mathrm{P} \\ = -\frac{(\rho_\mathrm{P}^{(n)} - \rho_\mathrm{P}^o)}{\delta t} V - \Delta[\rho^{(n)} U^{(n)}]_\mathrm{P} \end{aligned} \tag{104}$$

Apply correction to pressure field only.

Second predictor:

$$\mathbf{v}_\mathrm{P}^* = \mathbf{H}[\mathbf{v}^*]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P^*)_\mathrm{P} \tag{105}$$

Second corrector:

$$(v'', P'', \rho'')(\mathbf{v}^{**} = \mathbf{v}^* + \mathbf{v}'', P^{**} = P^* + P'', \rho^{**} = \rho^* + \rho'') \tag{106}$$

$$\therefore \mathbf{v}_P^{**} = \mathbf{H}[\mathbf{v}^{**}]_P - \mathbf{D}_P(\nabla P^{**})_P = \mathbf{H}[\mathbf{v}^* + \mathbf{v}'']_P - \mathbf{D}_P[\nabla(P^* + P'')]_P \tag{107}$$

$$\therefore \begin{cases} \mathbf{v}_P'' = \mathbf{H}[\mathbf{v}'']_P - \mathbf{D}_P(\nabla P'')_P \\ \rho'' = C_p P'' \end{cases} \tag{108}$$

Condition:

$$\frac{(\rho_P^{**} - \rho_P^o)}{\delta t} V + \Delta[\rho^{**}\mathbf{v}^{**} \cdot \mathbf{S}]_P = 0 \tag{109}$$

$$\therefore \frac{(\rho_P^* + \rho_P'' - \rho_P^o)}{\delta t} V + \Delta[(\rho^* + \rho'')(\mathbf{v}^* + \mathbf{H}[\mathbf{v}''] - \mathbf{D}(\nabla P'')) \cdot \mathbf{S}]_P = 0 \tag{110}$$

$$\therefore \frac{VC_p}{\delta t} P_P'' + \Delta[U^* C_p P'']_P - \Delta[\rho^* \mathbf{D}(\nabla P'') \cdot \mathbf{S}]_P$$

$$= -\frac{(\rho_P^* - \rho_P^o)}{\delta t} V - \Delta[\rho^* U^*]_P - \Delta[\rho^* \mathbf{H}[\mathbf{v}''] \cdot \mathbf{S}]_P - \Delta[(\rho''\mathbf{v}'') \cdot \mathbf{S}]_P \tag{111}$$

Approximation: Neglect: $\mathbf{H}[\mathbf{v}'']$, $\Delta[(\rho''\mathbf{v}'') \cdot \mathbf{S}]_P$.

$$\Rightarrow \mathbf{v}_P'' = -\mathbf{D}_P(\nabla P'')_P \tag{112}$$

Approximate equation:

$$\Rightarrow \frac{VC_p}{\delta t} P_P'' + \Delta[U^* C_p P'']_P - \Delta[\rho^* \mathbf{D}(\nabla P'') \cdot \mathbf{S}]_P = -\frac{(\rho_P^* - \rho_P^o)}{\delta t} V - \Delta[\rho^* U^*]_P \tag{113}$$

Do not correct pressure.

Discussion. In the first predictor-corrector step, the pressure field is computed using the old velocity field. This is somewhat similar to the PRIME method with the difference that the velocity field is neither explicitly evaluated during the predictor step nor corrected with the obtained pressure field. In the second predictor-corrector step, the second pressure-correction field, which is used to update only the velocity field, is computed using the new velocity field calculated at the current iteration. With the exception of ρ^* replacing $\rho^{(n)}$, this is equivalent to SIMPLE. As such, the SIMPLER algorithm can be viewed as a combination of an approximate form of PRIME to compute the pressure and SIMPLE to compute the velocities, thereby combining the best features of both methods.

A global SIMPLER iteration

Calculate the **D** field.

Solve the pressure-correction equation and update the pressure and density fields.

Solve implicitly for u and v using the new pressure and density fields.
Solve the pressure-correction equation using the new velocity field to obtain a new pressure-correction field.
Correct u and v using the new pressure-correction field.
Solve implicitly the energy equation and update the density field.
Return to the first step and iterate until convergence.

The PISO Algorithm

The PISO algorithm, as proposed by Issa [34], is a time-marching procedure in which, during each time step, there is a predictor step and one or more corrector steps. However, since the unsteady term and the underrelaxation practice have similar effects on the finite-difference equations [41], the procedure may equally be used in the context of an iterative formulation. The sequences of events in PISO are as follows.

The PISO Algorithm: Symbolic form

First predictor:

$$\mathbf{v}_{\mathrm{P}}^{*} = \mathbf{H}[\mathbf{v}^{*}]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{(n)})_{\mathrm{P}} \tag{114}$$

First corrector:

$$(\mathbf{v}', P', \rho')(\mathbf{v}^{**} = \mathbf{v}^{*} + \mathbf{v}', P^{*} = P^{(n)} + P', \rho^{*} = \rho^{(n)} + \rho') \tag{115}$$

$$\therefore \mathbf{v}_{\mathrm{P}}^{**} = \mathbf{H}[\mathbf{v}^{**}]_{\mathrm{P}} - \mathbf{D}_{\mathrm{p}}(\nabla P^{*})_{\mathrm{P}} = \mathbf{H}[\mathbf{v}^{*} + \mathbf{v}']_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}\lfloor \nabla(P^{(n)} + P')\lfloor_{\mathrm{P}} \tag{116}$$

$$\therefore \begin{cases} \mathbf{v}_{\mathrm{P}}' = \mathbf{H}[\mathbf{v}']_{\mathrm{P}} - D_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \\ \rho' = C_{p} P' \end{cases} \tag{117}$$

Condition:

$$\frac{(\rho_{\mathrm{P}}^{*} - \rho_{\mathrm{P}}^{\mathrm{o}})}{\delta t} V + \Delta[\rho^{*}\mathbf{v}^{**} \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{118}$$

$$\therefore \frac{(\rho_{\mathrm{P}}^{(n)} + \rho_{\mathrm{P}}' - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[(\rho^{(n)} + \rho')(\mathbf{v}^{*} + \mathbf{H}[\mathbf{v}'] - \mathbf{D}(\nabla P')) \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{119}$$

$$\therefore \frac{VC_P}{\delta t} P_{\mathrm{P}}' + \Delta[C_p U^{*} P']_{\mathrm{P}} - \Delta[\rho^{(n)}\mathbf{D}(\nabla P') \cdot \mathbf{S}]_{\mathrm{P}}$$

$$= -\frac{(\rho_{\mathrm{P}}^{n} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta[\rho^{(n)} U^{*}]_{\mathrm{P}} - \Delta[\rho^{(n)}\mathbf{H}[\mathbf{v}'] \cdot \mathbf{S}]_{\mathrm{P}} - \Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \tag{120}$$

Approximation: Neglect $\mathbf{H}[\mathbf{v}'], \Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}}$.

$$\Rightarrow \mathbf{v}_{\mathrm{P}}' = -\mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \tag{121}$$

Approximate equation:

$$\Rightarrow \frac{VC_p}{\delta t} P'_{\mathrm{P}} + \Delta\left[C_p U^* P'\right]_{\mathrm{P}} - \Delta\left[\rho^{(n)} \mathbf{D}(\nabla P') \cdot \mathbf{S}\right]_{\mathrm{P}}$$

$$= -\frac{(\rho_{\mathrm{P}}^{n} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta\left[\rho^{(n)} U^*\right]_{\mathrm{P}} \tag{122}$$

Second predictor:

$$\mathbf{v}_{\mathrm{P}}^{***} = \mathbf{H}^{**}\left[\mathbf{v}^{**}\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}{}^{**}(\nabla P^*)_{\mathrm{P}} \tag{123}$$

Second corrector:

$$(\mathbf{v}'', P'', \rho'')(\mathbf{v}^{****} = \mathbf{v}^{***} + \mathbf{v}'', P^{**} = P^* + P'', \rho^{**} = \rho^* + \rho'') \tag{124}$$

$$\therefore \mathbf{v}_{\mathrm{P}}^{****} = \mathbf{H}^{**}\left[\mathbf{v}^{****}\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}^{**} \lfloor \nabla(P^* + P'') \lfloor_{\mathrm{P}} \tag{125}$$

$$\mathbf{v}_{\mathrm{P}}^{***} = \mathbf{H}^{**}\left[\mathbf{v}^{**}\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}^{**}(\nabla P^*)_{\mathrm{P}} \tag{126}$$

$$\therefore \begin{cases} \mathbf{v}''_{\mathrm{P}} = \mathbf{H}^{**}\left[\mathbf{v}^{****} - \mathbf{v}^{**}\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}^{**}(\nabla P'')_{\mathrm{P}} \\ \quad = \mathbf{H}^{**}\left[\mathbf{v}^{***} - \mathbf{v}^{**} + v''\right]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}^{**}(\nabla P'')_{\mathrm{P}} \\ \rho'' = C_p P'' \end{cases} \tag{127}$$

Condition:

$$\therefore \frac{(\rho_{\mathrm{P}}^{**} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta\left[\rho^{**} \mathbf{v}^{****} \cdot \mathbf{S}\right]_{\mathrm{P}} = 0 \tag{128}$$

$$\therefore \frac{(\rho_{\mathrm{P}}^{*} + \rho''_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t} V$$

$$+ \Delta\left[(\rho^* + \rho'')(\mathbf{v}^{***} + \mathbf{H}^{**}\left[\mathbf{v}^{***} - \mathbf{v}^{**} + \mathbf{v}''\right] - \mathbf{D}_{\mathrm{P}}^{**}(\nabla P'')) \cdot \mathbf{S}\right]_{\mathrm{P}} = 0 \tag{129}$$

$$\therefore \frac{VC_p}{\delta t} P''_{\mathrm{P}} + \Delta\left[C_p U^{***} P''\right]_{\mathrm{P}} - \Delta\left[\rho^* \mathbf{D}_{\mathrm{P}}^{**}(\nabla P'') \cdot \mathbf{S}\right]_{\mathrm{P}}$$

$$= -\frac{(\rho_{\mathrm{P}}^{*} - \rho_{\mathrm{P}}^{o})}{\delta t} V - \Delta\left[\rho^* U^{***}\right]_{\mathrm{P}} - \Delta\left[\rho^* \mathbf{H}^{**}\left[\mathbf{v}^{***} - \mathbf{v}^{**} + \mathbf{v}''\right] \cdot \mathbf{S}\right]_{\mathrm{P}}$$

$$- \Delta\left[(\rho'' \mathbf{v}'') \cdot \mathbf{S}\right]_{\mathrm{P}} \tag{130}$$

Approximation: Neglect: $\mathbf{H}^{**}[\mathbf{v}^{***} - \mathbf{v}^{**} + \mathbf{v}'']_{\mathrm{P}}, \Delta[(\rho''\mathbf{v}'')\cdot\mathbf{S}]_{\mathrm{P}}$.

$$\Rightarrow \mathbf{v}''_{\mathrm{P}} = -\mathbf{D}^{**}_{\mathrm{P}}(\nabla P'')_{\mathrm{P}} \tag{131}$$

Approximate equation:

$$\begin{aligned} &\Rightarrow \frac{VC_p}{\delta t}P''_{\mathrm{P}} + \Delta[C_p U^{***}P'']_{\mathrm{P}} - \Delta[\rho^{*}\mathbf{D}^{**}_{\mathrm{P}}(\nabla P'')\cdot\mathbf{S}]_{\mathrm{P}} \\ &= -\frac{(\rho^{*}_{\mathrm{P}} - \rho^{o}_{\mathrm{P}})}{\delta t}V - \Delta[\rho^{*}U^{***}]_{\mathrm{P}} \end{aligned} \tag{132}$$

Discussion. The overall velocity correction is given as

$$\begin{aligned} \mathbf{v}^{****}_{\mathrm{P}} = \mathbf{v}^{***}_{\mathrm{P}} + \mathbf{v}''_{\mathrm{P}} &= \mathbf{H}^{**}[\mathbf{v}^{**}]_{\mathrm{P}} - \mathbf{D}^{**}_{\mathrm{P}}(\nabla P^{*})_{\mathrm{P}} + \mathbf{v}''_{\mathrm{P}} \\ &= \mathbf{H}^{**}[\mathbf{v}^{*} + \mathbf{v}']_{\mathrm{P}} - \mathbf{D}^{**}_{\mathrm{P}}(\nabla P^{*})_{\mathrm{P}} + \mathbf{v}''_{\mathrm{P}} \\ &= \mathbf{H}^{**}[\mathbf{v}^{*}] + \mathbf{H}^{**}[\mathbf{v}']_{\mathrm{P}} - \mathbf{D}^{**}_{\mathrm{P}}(\nabla P^{*})_{\mathrm{P}} + \mathbf{v}''_{\mathrm{P}} \\ &= \mathbf{H}^{**}[\mathbf{v}^{*}] - \mathbf{D}^{**}_{\mathrm{P}}(\nabla P^{*})_{\mathrm{P}} + \mathbf{H}^{**}[\mathbf{v}']_{\mathrm{P}} + \mathbf{v}''_{\mathrm{P}} \end{aligned} \tag{133}$$

If $\mathbf{v}^{*}$ is assumed to be nearly equal to $\mathbf{H}^{**}[\mathbf{v}^{*}] - \mathbf{D}^{**}_{\mathrm{P}}(\nabla P^{*})_{\mathrm{P}}$ then, the $\mathbf{H}[\mathbf{v}']$ term that was neglected in the first corrector stage is partially recovered as $\mathbf{H}^{**}[-\mathbf{D}(\nabla P')]_{\mathrm{P}}$ in the second corrector stage in addition to $\mathbf{v}''_{\mathrm{P}}$. Therefore, the second pressure-correction brings the velocity and pressure fields closer to satisfying both the momentum and continuity equations without the need to underrelax the pressure correction. Because of this fact, PISO does provide considerable savings in computational time when compared to SIMPLE. Moreover, by following the sequence of events, it can be easily seen that PISO may be considered to be a combination of one SIMPLE step and one PRIME step, hence combining the implicitness of the SIMPLE algorithm with the stability of the PRIME algorithm. Furthermore, improving the time accuracy of PISO through increasing the number of corrector iterations is akin to adding further PRIME steps.

Finally, it is common in the literature to read that PISO is essentially equivalent to SIMPLER (e.g., [42]). From the above presentation, it is obvious that the two algorithms are different. In SIMPLER the pressure is updated once after solving a pressure or a pressure-correction equation in which the coefficients are based on the old velocity field. In PISO the pressure field is updated twice using the latest available velocity fields. Consequently, the pressure and velocity fields at the end of a SIMPLER iteration are different from those obtained at the end of a PISO iteration, and users of SIMPLER and PISO should be aware of that.

A global PISO iteration

Solve implicitly for u and v, using the old pressure and density fields.
Calculate the $\mathbf{D}$ field.
Solve the pressure-correction equation.
Correct u, v, P, and ρ.

Solve implicitly the energy equation and update the density field.
Solve the momentum equations explicitly and calculate the **D** field.
Solve the pressure-correction equation.
Correct u, v, P, and ρ.
Return to the first step and iterate until convergence

The SIMPLEX Algorithm

In all SIMPLE-based methods, no care is taken to ensure that the rate of convergence will not degrade with grid refinement. This concern is addressed in SIMPLEX [32] by considering the influence of pressure differences outside but in the vicinity of the pressure difference local to the velocity. This is accomplished by using extrapolation to express all pressure differences in the domain in terms of the pressure difference local to the velocity; the X is appended to SIMPLE to indicate the use of extrapolation. The idea is based on the fact that the spatial distribution of pressure difference influence changes little with grid refinement. Therefore, if the pressure difference influence is restricted to a control volume, it would be appropriate to assume that, by extrapolation, the pressure difference at the main grid point adequately represents the pressure differences at the control-volume faces. Mathematically, the SIMPLEX algorithm can be described as follows.

The SIMPLEX Algorithm: SYMBOLIC form

Predictor:

$$\mathbf{v}_{\mathrm{P}}^{*} = \mathbf{H}[\mathbf{v}^{*}]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{(n)})_{\mathrm{P}} \tag{134}$$

Corrector:

$$(\mathbf{v}', P', \rho')(\mathbf{v}^{**} = \mathbf{v}^{*} + \mathbf{v}',\ P^{*} = P^{(n)} + P',\ \rho^{*} = \rho^{(n)} + \rho') \tag{135}$$

$$\therefore \mathbf{v}_{\mathrm{P}}^{**} = \mathbf{H}[\mathbf{v}^{**}]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P^{*})_{\mathrm{P}} = \mathbf{H}[\mathbf{v}^{*} + \mathbf{v}']_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}\lfloor \nabla(P^{(n)} + P')\lfloor_{\mathrm{P}} \tag{136}$$

$$\therefore \begin{cases} \mathbf{v}_{\mathrm{P}}' = \mathbf{H}[\mathbf{v}']_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \\ \rho' = C_p P' \end{cases} \tag{137}$$

Condition:

$$\frac{(\rho_{\mathrm{P}}^{*} - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[\rho^{*}\mathbf{v}^{**} \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{138}$$

$$\therefore \frac{(\rho_{\mathrm{P}}^{(n)} + \rho_{\mathrm{P}}' - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[(\rho^{(n)} + \rho')(\mathbf{v}^{*} + v') \cdot \mathbf{S}]_{\mathrm{P}} = 0 \tag{139}$$

$$\therefore \frac{(\rho_{\mathrm{P}}^{(n)} + \rho_{\mathrm{P}}' - \rho_{\mathrm{P}}^{o})}{\delta t} V + \Delta[\rho'\mathbf{v}^{*} \cdot \mathbf{S}]_{\mathrm{P}} + \Delta[\rho^{(n)} v' \cdot \mathbf{S}]_{\mathrm{P}} = -\Delta[\rho^{(n)}\mathbf{v}^{*} \cdot \mathbf{S}]_{\mathrm{P}} - \Delta[(\rho'\mathbf{v}') \cdot \mathbf{S}]_{\mathrm{P}} \tag{140}$$

Approximation: Neglect: $\Delta[(\rho'\mathbf{v}')\cdot\mathbf{S}]_{\mathrm{P}}$ and let

$$\mathbf{v}'_{\mathrm{P}} = \mathbf{H}[\mathbf{v}']_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} = -\mathbf{D}^{\mathrm{SX}}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \tag{141}$$

$$\Rightarrow -\mathbf{D}^{\mathrm{sx}}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} = \mathbf{H}[-\mathbf{D}^{\mathrm{sx}}(\nabla P')]_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \tag{142}$$

Assume that the pressure difference local to the velocity can be approximated to be representative of all pressure differences, i.e., $\mathbf{H}[-\mathbf{D}^{\mathrm{sx}}(\nabla \mathrm{P}')]_{\mathrm{P}} = (\nabla \mathrm{P}')_{\mathrm{P}}\mathbf{H}[-\mathbf{D}^{\mathrm{sx}}]_{\mathrm{P}}$. Thus,

$$\mathbf{D}^{\mathrm{sx}}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} = (\nabla P')_{\mathrm{P}}\mathbf{H}[\mathbf{D}^{\mathrm{sx}}]_{\mathrm{P}} + \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} \tag{143}$$

$$\Rightarrow \mathbf{D}^{\mathrm{sx}}_{\mathrm{P}} = \mathbf{H}[\mathbf{D}^{\mathrm{sx}}]_{\mathrm{P}} + \mathbf{D}_{\mathrm{P}} \tag{144}$$

Approximate equation:

$$\Rightarrow \frac{VC_p}{\delta t}P'_P + \Delta[C_p U^* P']_{\mathrm{P}} - \Delta[\rho^{(n)}\mathbf{D}^{\mathrm{sx}}(\nabla P')\cdot\mathbf{S}]_{\mathrm{P}}$$

$$= -\frac{(\rho^{(n)}_{\mathrm{P}} - \rho^{o}_{\mathrm{P}})}{\delta t}V - \Delta[\rho^{(n)}U^*]_{\mathrm{P}} \tag{145}$$

Discussion. The pressure-correction equation in SIMPLEX involves the $\mathbf{D}^{\mathrm{sx}}$ field, rather than the $\mathbf{D}$ field, which has to be computed by solving an additional system of equations [Eq. (144)] but using the coefficients of the momentum equations. Therefore, since the coefficients need not be recalculated, the added computational effort is not major. Moreover, as reported in [32], this additional cost is offset by a lower degradation in the rate of convergence with grid refinement as compared to other SIMPLE-like algorithms. Although SIMPLEX is not necessarily the most efficient algorithm on coarse grids, for sufficiently fine grids the study presented in [32] indicates that SIMPLEX is more efficient than SIMPLE, SIMPLER, and SIMPLEC.

A global SIMPLEX iteration

Solve implicitly for u and v, using the old pressure and density fields.
Calculate the $\mathbf{D}$ field.
Solve implicitly for the $\mathbf{D}^{\mathrm{sx}}$ field.
Solve the pressure-correction equation using this $\mathbf{D}^{\mathrm{sx}}$ field.
Correct u, v, P, and ρ.
Solve implicitly the energy equation and update the density field.
Return to the first step and iterate until convergence.

The SIMPLEM Algorithm

The SIMPLEM algorithm of Acharya and Moukalled [33] is composed of a predictor and a corrector stage. In the corrector stage, a pressure-correction field is obtained based on the old velocity field. This new pressure field is then used to predict a new velocity field. Mathematically, the various steps are as follows.

The SIMPLEM algorithm: Symbolic form

***First predictor*:**. No predictor stage. Only coefficients of the momentum equations are calculated.

***First corrector*:**

$$(\mathbf{v}', P', \rho')(\mathbf{v}^* = \mathbf{v}^{(n)} + \mathbf{v}', P^* = P^{(n)} + P', \rho^* = \rho^{(n)} + \rho') \tag{146}$$

$$\therefore \mathbf{v}_\mathrm{P}^* = \mathbf{H}[\mathbf{v}^*]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P^*)_\mathrm{P} \tag{147}$$

$$\mathbf{v}_\mathrm{P}^n = \mathbf{H}[\mathbf{v}^{(n)}]_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P^{(n)})_\mathrm{P} \tag{148}$$

$$\therefore \begin{cases} \mathbf{v}'_\mathrm{P} = \mathbf{H}[\mathbf{v}']_\mathrm{P} - \mathbf{D}_\mathrm{P}(\nabla P')_\mathrm{P} \\ \rho' = C_p P' \end{cases} \tag{149}$$

Condition:

$$\frac{(\rho_\mathrm{P}^* - \rho_\mathrm{P}^o)}{\delta t} V + \Delta[\rho^* \mathbf{v}^* \cdot S]_\mathrm{P} = 0 \tag{150}$$

$$\therefore \frac{(\rho_\mathrm{P}^{(n)} + \rho'_\mathrm{P} - \rho_\mathrm{P}^o)}{\delta t} V + \Delta[(\rho^{(n)} + \rho')(\mathbf{v}^{(n)} + \mathbf{H}[\mathbf{v}']_\mathrm{P} - \mathbf{D}(\nabla P')) \cdot \mathbf{S}]_\mathrm{P} = 0 \tag{151}$$

$$\therefore \frac{VC_p}{\delta t} P_\mathrm{P}{}' + \Delta\Big[C_p U^{(n)} P'\Big]_\mathrm{P} - \Delta[\rho^{(n)} \mathbf{D}(\nabla P') \cdot \mathbf{S}]_\mathrm{p}$$
$$= -\frac{(\rho_\mathrm{P}^{(n)} - \rho_\mathrm{P}^o)}{\delta t} V - \Delta[\rho^{(n)} U^n]_\mathrm{P} - \Delta[\rho^{(n)} \mathbf{H}[\mathbf{v}']_\mathrm{P} \cdot \mathbf{S}]_\mathrm{P} - \Delta[(\rho' \mathbf{v}') \cdot \mathbf{S}]_\mathrm{P} \tag{152}$$

Approximation: Neglect: $\mathbf{H}[\mathbf{v}']_\mathrm{P}, \Delta[(\rho'\mathbf{v}') \cdot_\mathbf{S}]_\mathrm{P}$.
Approximate equation:

$$\Rightarrow \frac{VC_p}{\delta t} P'_\mathrm{P} + \Delta\Big[C_p U^{(n)} P'\Big]_\mathrm{P} - \Delta[\rho^{(n)} \mathbf{D}(\nabla P') \cdot \mathbf{S}]_\mathrm{P}$$
$$= -\frac{(\rho_\mathrm{P}^{(n)} - \rho_\mathrm{P}^o)}{\delta t} V - \Delta[\rho^{(n)} U^{(n)}]_\mathrm{P} \tag{153}$$

***Second predictor*:**

$$\mathbf{v}_\mathrm{P}^{**} = \mathbf{H}^*[\mathbf{v}^{**}]_\mathrm{P} - \mathbf{D}_\mathrm{P}^*(\nabla P^*)_\mathrm{P} \tag{154}$$

***Second corrector*:**. No corrector stage.

***Discussion*.** As in SIMPLER, the pressure field in SIMPLEM is computed using the old velocity field. This is nearly equivalent to PRIME, which was shown to do a good job in correcting the pressure field. By so doing, however, the velocity corrections will be at a disadvantage. Therefore, in SIMPLEM the disadvantages and advantages of SIMPLE are interchanged.

A global SIMPLEM iteration

Calculate the **D** field based on values from the previous iteration.
Solve the pressure-correction equation.
Correct u, v, P, and ρ.
Calculate new **H** and **D** fields.
Solve implicitly for u and v using the new fields.
Solve implicitly the energy equation and update the density field.
Return to the first step and iterate until convergence.

ENHANCEMENT OF THE SIMPLE ALGORITHM AND ITS VARIANTS BY AN ADDITIONAL CORRECTION TERM

This method, proposed by Shaw and Sivaloganathan [43], takes into consideration that the momentum equations are not driven to full convergence at every iteration and uses the residual of the incompletely converged equations in the expression for the velocity corrections. Therefore, in deriving the pressure-correction equation, the velocity correction field [Eq. (38)] is modified to

$$\mathbf{v}'_{\mathrm{P}} = \mathbf{H}[\mathbf{v}']_{\mathrm{P}} - \mathbf{D}_{\mathrm{P}}(\nabla P')_{\mathrm{P}} - \frac{\mathbf{D}_{\mathrm{P}}}{V}\mathbf{R}_M \qquad (155)$$

where $\mathbf{R}_M$ denotes the momentum residual. As stated in [37], this method has a better theoretical smoothing rate, suitable for the high-frequency error reduction that is important in multigrid techniques. In [37, 43], the technique was applied to the SIMPLE and SIMPLEC algorithms to yield two modified versions denoted by SIMPLESSE and SIMPLESSEC, respectively. However, the approach is equally applicable to all previously described algorithms. Moreover, when using this technique with the SIMPLEC algorithm, the **H** and **D** operators should be replaced by their equivalent expressions as derived earlier.

THE EXPANDED FORM OF THE PRESSURE-CORRECTION EQUATION

It is obvious by now that the various simplified pressure-correction equations are similar and may be written as

$$\frac{VC_p}{\delta t}P'_{\mathrm{P}} + \Delta[C_p UP']_{\mathrm{P}} - \Delta[\rho\mathbf{D}(\nabla P')\cdot\mathbf{S}]_{\mathrm{P}} = -\frac{V(\rho_P - \rho^o_{\mathrm{P}})}{\delta t} - \Delta[\rho U]_{\mathrm{P}} \qquad (156)$$

where, depending on the algorithm used, U and ρ represent values from the previous iteration or from a previous corrector step. When discretizing this equation, careful attention should be paid to the second term on the left-hand side, which is similar to a convection term and for which any convective scheme may be used. Adopting the UPWIND scheme [1], the discretized form of the convection-like term $\Delta\lfloor C_pUP'\lfloor_{\mathrm{P}}$ is

$$
\begin{aligned}
&\Delta\lfloor C_pUP'\lfloor_{\mathrm{P}}\\
&\quad = (C_pUP')_{\mathrm{e}} + (C_pUP')_{\mathrm{w}} + (C_pUP')_{\mathrm{n}} + (C_pUP')_{s}\\
&\quad = (C_p)_e[\|U_e,0\|P'_{\mathrm{P}} - \|-U_{\mathrm{e}},0\|P'_{\mathrm{E}}] + (C_p)_{\mathrm{w}}\big[\|U_{\mathrm{w}},0\|P'_{\mathrm{p}} - \|-U_{\mathrm{w}},0\|P'_{\mathrm{W}}\big]\\
&\qquad +(C_p)_{\mathrm{n}}[\|U_{\mathrm{n}},0\|P'_{\mathrm{P}} - \|-U_{\mathrm{n}},0\|P'_{\mathrm{N}}] + (C_p)_{\mathrm{s}}[\|U_{\mathrm{s}},0\|P'_{\mathrm{P}} - \|-U_{\mathrm{s}},0\|P'_{\mathrm{S}}]
\end{aligned}
\tag{157}
$$

Rearranging, one obtains:

$$
\begin{aligned}
\Delta[C_pUP']_{\mathrm{P}} = &\big[(C_p)_{\mathrm{e}}\|U_{\mathrm{e}},0\| + (C_p)_{\mathrm{w}}\|U_{\mathrm{w}},0\| + (C_p)_{\mathrm{n}}\|U_{\mathrm{n}},0\| + (C_p)_{\mathrm{s}}\|U_{\mathrm{s}},0\|\big] P'_{\mathrm{P}}\\
&-(C_p)_{\mathrm{e}}\|-U_{\mathrm{e}},0\|P'_{\mathrm{E}} - (C_p)_{\mathrm{w}}\|-U_{\mathrm{w}},0\|P'_{\mathrm{W}}\\
&-(C_p)_{\mathrm{n}}\|-U_{\mathrm{n}},0\|P'_{\mathrm{N}} - (C_p)_{\mathrm{s}}\|-U_{\mathrm{s}},0\|P'_{\mathrm{S}}
\end{aligned}
\tag{158}
$$

The term $\Delta[\rho\mathbf{D}(\nabla P')\cdot\mathbf{S}]_{\mathrm{P}}$ is discretized following the same procedure as was used in discretizing the pressure gradient term needed in calculating the interface velocities. Its final form is given by

$$
\begin{aligned}
&\Delta[\rho\mathbf{D}(\nabla P')\cdot\mathbf{S}]_{\mathrm{P}}\\
&\quad = (\rho\mathbf{D}(\nabla P')\cdot\mathbf{S})_{\mathrm{e}} + (\rho\mathbf{D}(\nabla P')\cdot\mathbf{S})_{\mathrm{w}} + (\rho\mathbf{D}(\nabla P')\cdot\mathbf{S})_{\mathrm{n}} + (\rho\mathbf{D}(\nabla P')\cdot\mathbf{S})_{\mathrm{s}}\\
&\quad = \rho_{\mathrm{e}}\frac{\overline{D}[u]_{\mathrm{e}}(S^x_{\mathrm{e}})^2 + \overline{D}[v]_{\mathrm{e}}(S^y_{\mathrm{e}})^2}{\mathbf{S}_{\mathrm{e}}\cdot\mathbf{d}_{\mathrm{e}}}(P'_{\mathrm{E}} - P'_{\mathrm{P}})\\
&\qquad \underline{+\rho_{\mathrm{e}}\Big(\overline{D}[u]_{\mathrm{e}}\big(\overline{\nabla P'}\big)^x_{\mathrm{e}}\kappa^x_{\mathrm{e}} + \overline{D}[v]_{\mathrm{e}}\big(\overline{\nabla P'}\big)^y_{\mathrm{e}}\kappa^y_{\mathrm{e}}\Big)}\\
&\qquad\qquad +\rho_{\mathrm{w}}\frac{\overline{D}[u]_{\mathrm{w}}(S^x_{\mathrm{w}})^2 + \overline{D}[v]_{\mathrm{w}}(S^y_{\mathrm{w}})^2}{\mathbf{S}_{\mathrm{w}}\cdot\mathbf{d}_{\mathrm{w}}}(P'_{\mathrm{W}} - P'_{\mathrm{P}})\\
&\qquad \underline{+\rho_{\mathrm{w}}\Big(\overline{D}[u]_{\mathrm{w}}\big(\overline{\nabla P'}\big)^x_{\mathrm{w}}\kappa^x_{\mathrm{w}} + \overline{D}[v]_{\mathrm{w}}\big(\overline{\nabla P'}\big)^y_{\mathrm{w}}\kappa^y_{\mathrm{w}}\Big)}\\
&\qquad +\rho_{\mathrm{n}}\frac{\overline{D}[u]_{\mathrm{n}}(S^x_{\mathrm{n}})^2 + \overline{D}[v]_{\mathrm{n}}(S^y_{\mathrm{n}})^2}{\mathbf{S}_{\mathrm{n}}\cdot\mathbf{d}_{\mathrm{n}}}(P'_{\mathrm{N}} - P'_{\mathrm{P}})\\
&\qquad \underline{+\rho_{\mathrm{n}}\Big(\overline{D}[u]_{\mathrm{n}}\big(\overline{\nabla P'}\big)^x_{\mathrm{n}}\kappa^x_{\mathrm{n}} + \overline{D}[v]_{\mathrm{n}}\big(\overline{\nabla P'}\big)^y_{\mathrm{n}}\kappa^y_{\mathrm{n}}\Big)}\\
&\qquad +\rho_{\mathrm{s}}\frac{\overline{D}[u]_{\mathrm{s}}(S^x_{\mathrm{s}})^2 + \overline{D}[v]_{\mathrm{s}}(S^y_{\mathrm{s}})^2}{\mathbf{S}_{\mathrm{s}}\cdot\mathbf{d}_{\mathrm{s}}}(P'_{\mathrm{S}} - P'_{\mathrm{P}})\\
&\qquad \underline{+\rho_{\mathrm{s}}\Big(\overline{D}[u]_{\mathrm{s}}\big(\overline{\nabla P'}\big)^x_{\mathrm{s}}\kappa^x_{\mathrm{s}} + \overline{D}[v]_{\mathrm{s}}\big(\overline{\nabla P'}\big)^y_{\mathrm{s}}\kappa^y_{\mathrm{s}}\Big)}
\end{aligned}
\tag{159}
$$

where the underlined terms account for the nonorthogonal factors. They are usually neglected since their contribution is small in comparison with other terms and vanish when the grid is orthogonal. However, they could be accounted for by moving them to the right-hand side, adding them to the source term, and modifying the solver to explicitly update their values after a solver (not global) iteration. Neglecting these terms, Eq. (159) becomes

$$
\begin{aligned}
&\Delta[\rho\mathbf{D}(\nabla P')\cdot\mathbf{S}]_{\mathrm{P}}\\
&\quad=\rho_{\mathrm{e}}\frac{\overline{D}[u]_{\mathrm{e}}(S_{\mathrm{e}}^{x})^{2}+\overline{D}[v]_{\mathrm{e}}(S_{\mathrm{e}}^{y})^{2}}{S_{\mathrm{e}}^{x}d_{\mathrm{e}}^{x}+S_{\mathrm{e}}^{y}d_{\mathrm{e}}^{y}}(P'_{\mathrm{E}}-P'_{\mathrm{P}})\\
&\qquad+\rho_{\mathrm{w}}\frac{\overline{D}[u]_{\mathrm{w}}(S_{\mathrm{w}}^{x})^{2}+\overline{D}[v]_{\mathrm{w}}(S_{\mathrm{w}}^{y})^{2}}{S_{\mathrm{w}}^{x}d_{\mathrm{w}}^{x}+S_{\mathrm{w}}^{y}d_{\mathrm{w}}^{y}}(P'_{\mathrm{W}}-P'_{\mathrm{P}})\\
&\qquad+\rho_{\mathrm{n}}\frac{\overline{D}[u]_{\mathrm{n}}(S_{\mathrm{n}}^{x})^{2}+\overline{D}[v]_{\mathrm{n}}(S_{\mathrm{n}}^{y})^{2}}{S_{\mathrm{n}}^{x}d_{\mathrm{n}}^{x}+S_{\mathrm{n}}^{y}d_{\mathrm{n}}^{y}}(P'_{\mathrm{N}}-P'_{\mathrm{P}})\\
&\qquad+\rho_{\mathrm{s}}\frac{\overline{D}[u]_{\mathrm{s}}(S_{\mathrm{s}}^{x})^{2}+\overline{D}[v]_{\mathrm{s}}(S_{\mathrm{s}}^{y})^{2}}{S_{\mathrm{s}}^{x}d_{\mathrm{s}}^{x}+S_{\mathrm{s}}^{y}d_{\mathrm{s}}^{y}}(P'_{\mathrm{S}}-P'_{\mathrm{P}})\\
&\quad=\Gamma_{\mathrm{e}}^{P'}(P'_{\mathrm{E}}-P'_{\mathrm{P}})+\Gamma_{\mathrm{w}}^{P'}(P'_{\mathrm{W}}-P'_{\mathrm{P}})+\Gamma_{\mathrm{n}}^{P'}(P'_{\mathrm{N}}-P'_{\mathrm{P}})+\Gamma_{\mathrm{s}}^{P'}(P'_{\mathrm{S}}-P'_{\mathrm{P}})\\
&\quad=-(\Gamma_{\mathrm{e}}^{P'}+\Gamma_{\mathrm{w}}^{P'}+\Gamma_{\mathrm{n}}^{P'}+\Gamma_{\mathrm{s}}^{P'})P'_{\mathrm{P}}+\Gamma_{\mathrm{e}}^{P'}P'_{\mathrm{E}}+\Gamma_{\mathrm{w}}^{P'}P'_{\mathrm{W}}+\Gamma_{\mathrm{n}}^{P'}P'_{\mathrm{N}}+\Gamma_{\mathrm{s}}^{P'}P'_{\mathrm{S}}
\end{aligned}
\tag{160}
$$

Moreover, the discretized form of $\Delta[\rho U]_{\mathrm{P}}$ is given by

$$\Delta[\rho U]_{\mathrm{P}}=\rho_{\mathrm{e}}U_{\mathrm{e}}+\rho_{\mathrm{w}}U_{\mathrm{w}}+\rho_{\mathrm{n}}U_{\mathrm{n}}+\rho_{\mathrm{s}}U_{\mathrm{s}} \tag{161}$$

Substituting the various terms in Eq. (156) by their equivalent expressions as derived above, the final form of the pressure-correction equation is written as

$$a_{\mathrm{P}}^{P'}P'_{\mathrm{P}}=a_{\mathrm{E}}^{P'}P'_{\mathrm{E}}+a_{\mathrm{W}}^{P'}P'_{\mathrm{W}}+a_{\mathrm{N}}^{P'}P'_{\mathrm{N}}+a_{\mathrm{S}}^{P'}P'_{\mathrm{S}}+b_{\mathrm{P}}^{P'} \tag{162}$$

where

$$
\begin{aligned}
a_{\mathrm{E}}^{P'}&=\Gamma_{\mathrm{e}}^{P'}+(C_{p})_{\mathrm{e}}\|-U_{\mathrm{e}},0\|\\
a_{\mathrm{W}}^{P'}&=\Gamma_{\mathrm{w}}^{P'}+(C_{p})_{\mathrm{w}}\|-U_{\mathrm{w}},0\|\\
a_{\mathrm{N}}^{P'}&=\Gamma_{\mathrm{n}}^{P'}+(C_{p})_{\mathrm{n}}\|-U_{\mathrm{n}},0\|\\
a_{\mathrm{S}}^{P'}&=\Gamma_{\mathrm{s}}^{P'}+(C_{p})_{\mathrm{s}}\|-U_{\mathrm{s}},0\|
\end{aligned}
\tag{163}
$$

$$(a_{\mathrm{P}}^{P'})^{o} = \frac{V(C_p)_{\mathrm{P}}}{\delta t}$$

$$a_{\mathrm{P}}^{P'} = a_{\mathrm{E}}^{P'} + a_{\mathrm{W}}^{P'} + a_{\mathrm{N}}^{P'} + a_{\mathrm{S}}^{P'} + (a_{\mathrm{P}}^{p'})^{o}$$

$$+\left[(C_p)_{\mathrm{e}}U_{\mathrm{e}} + (C_p)_{\mathrm{w}}U_{\mathrm{w}} + (C_p)_{\mathrm{n}}U_{\mathrm{n}} + (C_p)_{\mathrm{s}}U_{\mathrm{s}}\right]$$

$$b_{\mathrm{P}}^{P'} = -\frac{(\rho_{\mathrm{P}} - \rho_{\mathrm{P}}^{o})}{\delta t}V - (\rho_{\mathrm{e}}U_{\mathrm{e}} + \rho_{\mathrm{w}}U_{\mathrm{w}} + \rho_{\mathrm{n}}U_{\mathrm{n}} + \rho_{\mathrm{s}}U_{\mathrm{s}}) \qquad (163) \; (Cont.)$$

CLOSING REMARKS

The SIMPLE algorithm and its variants were reformulated using a unified, compact, and easy-to-understand notation. The new notation allowed the essence shared in their derivations to be transparent. Moreover, the philosophies behind these algorithms and the differences among them were pointed out. Furthermore, the formulation of all these algorithms was extended to predict fluid flow at all speeds.

REFERENCES

1. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1981.
2. S. V. Patankar and D. B. Spalding, A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows, *Int. J. Heat Mass Transfer*, vol. 15, pp. 1787–1806, 1972.
3. A. J. Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, *J. Comput. Phys.*, vol. 2, pp. 12–26, 1967.
4. A. Rizzi and L. E. Eriksson, Computation of Inviscid Incompressible Flow with Rotation, *J. Fluid Mech.*, vol. 153, no. 3, pp. 275–312, 1985.
5. D. Choi and C. L. Merkle, Application of Time-Iterative Schemes to Incompressible Flow, *AIAA J.*, vol. 23, pp. 1518–1524, 1985.
6. K. C. Karki, A Calculation Procedure for Viscous Flows at All Speeds in Complex Geometries, Ph.D. thesis, University of Minnesota, June 1986.
7. W. Shyy and M. E. Braaten, Adaptive Grid Computation for Inviscid Compressible Flows Using a Pressure Correction Method, AIAA Paper 88-3566-CP, 1988.
8. C. M. Rhie. A Pressure Based Navier-Stokes Solver Using the Multigrid Method, AIAA Paper 86-0207, 1986.
9. F. S. Lien and M. A. Leschziner, A General Non-orthogonal Collocated Finite Volume Algorithm for Turbulent Flow at All Speeds Incorporating Second-Moment Turbulence-Transport Closure, Part 1: Computational Implementation, *Comput. Meth. Appl. Mech. Eng.*, vol. 114, pp. 123–148, 1994.
10. E. S. Politis and K. C. Giannakoglou, A Pressure-Based Algorithm for High-Speed Turbomachinery Flows, *Int. J. Numer. Meth. Fluids*, vol. 25, pp. 63–80, 1997.
11. L. S. Caretto, R. M. Curr, and D. B. Spalding, Two Numerical Methods for Three Dimensional Boundary Layers, *Meth. Appl. Mech. Eng.*, vol. 1, p. 39, 1972.
12. S. P. Vanka, Fully Coupled Calculation of Fluid Flows with Limited Use of Computer Storage, Argonne Nat. Lab. Rep. ANL-83-87, 1983.

13. S. P. Vanka, and G. K. Leaf, Fully Coupled Solution of Pressure Linked Fluid Flow Equations, Argonne Nat. Lab. Rep. ANL-83-73, 1983.
14. M. E. Braaten, Development and Evaluation of Iterative and Direct Methods for the Solution of the Equations Governing Recirculating Flows, Ph.D. thesis, University of Minnesota, 1985.
15. F. H. Harlow and J. E. Welch, Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, *Phys. Fluids*, vol. 8, pp. 2182–2189, 1965.
16. W. Rodi, S. Majumdar, and B. Schonung, Finite Volume Methods for Two-Dimensional Incompressible Flows with Complex Boundaries, *Comput. Meth. Appl. Mech. Eng*, vol. 75, pp. 369–392, 1989.
17. C. M. Rhie, A Numerical Study of the Flow Past an Isolated Airfoil with Separation, Ph.D. thesis, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, 1981.
18. C. M. Rhie and W. L. Chow, A Numerical Study of the Turbulent Flow Past an Isolated Airfoil with Trailing Edge Separation, *AIAA J.*, vol. 21, pp. 1525–1532, 1983.
19. M. Peric, R. Kessler, and G. Scheuerer, Comparison of Finite Volume Numerical Methods with Staggered and Collocated Grids, *Comput. Fluids*, vol. 16, no. 4, pp. 289–403, 1988.
20. S. K. Choi, H. Y. Nam, and M. Cho, A Calculation Procedure for Incompressible Flow in Complex Geometries Using Momentum Interpolation Method, *7th Int. Conf. on Numerical Methods in Laminar & Turbulent Flow*, vol. VII, part 2, pp. 1634–1644, 1991.
21. C. Hsu, A Curvilinear-Coordinate Method for Momentum, Heat and Mass Transfer in Domains of Irregular Geometry, Ph.D. thesis, Univeristy of Minnesota, 1981.
22. C. Prakash, A Finite Element Method for Predicting Flow through Ducts with Arbitrary Cross Sections, Ph.D. thesis, University of Minnesota, 1981.
23. M. Peric, A Finite Volume Method for the Prediction of Three Dimensional Fluid Flow in Complex Ducts, Ph.D. thesis, Imperial College, Mechanical Engineering Department, London, 1985.
24. H. Aksoy and C. J. Chen, Numerical Solution of Navier-Stokes Equations with Non-Staggered Grids Using the Finite Analytical Method, *Numer. Heat Transfer B*, vol. 21, pp. 287–306, 1992.
25. T. F. Miller and F. W. Schmidt, Use of a Pressure-Weighted Interpolation Method for the Solution of the Incompressible Navier-Stokes Equations on a Non-staggered Grid System, *Numer. Heat Transfer*, vol. 14, pp. 213–233, 1988.
26. F. V. Poussin, An Accelerated Relaxation Algorithm for Iterative Solution of Elliptic Equations, *SIAM. J. Numer. Anal.*, vol. 5, pp. 340–351, 1968.
27. A. Brandt, Multi-level Adaptive Solutions to Boundary-Value Problems, *Math. Comput.*, vol. 31, pp. 333–390, 1977.
28. A. Brandt, *Guide to Multigrid Development*, Springer-Verlag, 1982.
29. B. R. Hutchinson and G. D. Raithby, A Multigrid Based on the Additive Correction Strategy, *Num. Heat Transfer* vol. 9, pp. 511–537, 1986.
30. J. P. Van Doormaal and G. D. Raithby, Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numer. Heat Transfer*, vol. 7, pp. 147–163, 1984.
31. D. B. Spalding, Mathematical Modelling of Fluid Mechanics, Heat Transfer and Mass Transfer Processes, Mech. Eng. Dept., Rep. HTS/80/1, Imperial College of Science, Technology and Medicine, London, 1980.
32. J. P. Van Doormaal and G. D. Raithby, An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows, ASME Paper 85-HT-9, presented at the Natl. Heat Transfer Conf., Denver, CO, 4–7 August 1985.

33. S. Acharya and F. Moukalled, Improvements to Incompressible Flow Calculation on a Non-staggered Curvilinear Grid, *Numer. Heat Transfer B*, vol. 15, pp. 131–152, 1989.
34. R. I. Issa, Solution of the Implicit Discretized Fluid Flow Equations by Operator Splitting, Mechanical Engineering Rep. FS/82/15, Imperial College, London, 1982.
35. C. R. Maliska and G. D. Raithby, Calculating 3-D Fluid Flows Using Non-orthogonal Grid, *Proc. Third Int. Conf. On Numerical Methods in Laminar and Turbulent Flows*, Seattle, WA, pp. 656–666, 1983.
36. R. H. Yen and C. H. Liu, Enhancement of the SIMPLE Algorithm by an Additional Explicit Corrector Step, *Numer. Heat Transfer B*, vol. 24, pp. 127–141, 1993.
37. T. Gjesdal and M. E. H. Lossius, Comparison of Pressure Correction Smoothers for Multigrid Solution of Incompressible Flow, *Int. J. Numer. Meth. Fluids*, vol. 25, pp. 393–405, 1997.
38. P. J. Zwart, G. D. Raithby, and M. J. Raw, An integrated Space-Time Finite-Volume Method for Moving-Boundary Problems, *Numer. Heat Transfer B*, vol. 34, pp. 257–270, 1998.
39. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, pp. 217–222, Springer-Verlag, Berlin, 1996.
40. G. Dahlquist and A. Bjorck, *Numerical Methods*, Prentice Hall, Englewood Cliffs, NJ, 1974.
41. D. S. Jang, R. Jetli, and S. Acharya, Comparison of the PISO, SIMPLER and SIMPLEC Algorithms for the Treatment of the Pressure-Velocity Coupling in Steady Flow Problems, *Numer. Heat Transfer*, vol. 10, pp. 209–228, 1986.
42. M. E. Braaten, Development and Evaluation of Iterative and Direct Methods for the Solution of the Equations Governing Recirculating Flows, Ph.D thesis, University of Minnesota, 1985.
43. G. J. Shaw and S. Sivaloganathan, The SIMPLE Pressure-Correction Method as a Nonlinear Smoother, in S. F. McCormick (ed.), *Multigrid Methods*: *Theory*, *Applications*, *and Supercomputing*. LNPAM vol. 110, pp. 579–596, Marcel Dekker, New York, 1988.