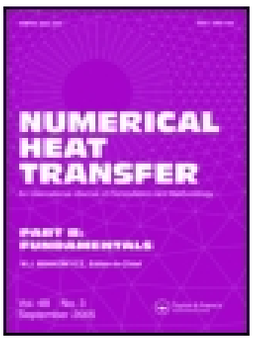


Numerical Heat Transfer, Part B: Fundamentals

ISSN: 1040-7790 (Print) 1521-0626 (Online) Journal homepage: http://www.tandfonline.com/loi/unhb20

PRESSURE-BASED ALGORITHMS FOR MULTIFLUID FLOW AT ALL SPEEDS—PART I: MASS CONSERVATION FORMULATION

F. Moukalled & M. Darwish

To cite this article: F. Moukalled & M. Darwish (2004) PRESSURE-BASED ALGORITHMS FOR MULTIFLUID FLOW AT ALL SPEEDS—PART I: MASS CONSERVATION FORMULATION, Numerical Heat Transfer, Part B: Fundamentals, 45:6, 495-522, DOI: 10.1080/10407790490430651

To link to this article: http://dx.doi.org/10.1080/10407790490430651

Published online: 17 Aug 2010.

Numerical Heat Transfer, Part B, 45: 495–522, 2004

ISSN: 1040-7790 print/1521-0626 online
DOI: 10.1080/10407790490430651

Taylor & Francis
Taylor & Francis Group

PRESSURE-BASED ALGORITHMS FOR MULTIFLUID FLOW AT ALL SPEEDS—PART I: MASS CONSERVATION FORMULATION

F. Moukalled and M. Darwish
American University of Beirut, Faculty of Engineering & Architecture, Mechanical Engineering Department, Riad El Solh, Beirut, Lebanon

In this article, seven segregated single-fluid, pressure-based algorithms are extended to predict multifluid flow at all speeds. The extended algorithms form part of the mass conservation-based algorithms (MCBA) group, in which the pressure-correction equation is derived from overall mass conservation. The performance and accuracy of these algorithms are assessed by solving a variety of two-dimensional two-phase flow problems in the subsonic, transsonic, and supersonic regimes. Solutions are generated for several grid densities using the single-grid (SG), the prolongation-grid (PG), and the full nonlinear multigrid (FMG) methods, and their effects on convergence behavior are studied. The main outcomes of this study are clear demonstrations of: (1) the capability of all MCBA algorithms to deal with multifluid flow situations; (2) the ability of the FMG method to tackle the added nonlinearity of multifluid flows; (3) and the capacity of the MCBA algorithms to predict multifluid flow at all speeds. Moreover, results indicate that the performances of SIMPLE, SIMPLEC, and SIMPLEX are very close. The PRIME algorithm is the most expensive, due to the explicit treatment of the fluidic momentum equations. The PISO algorithm is generally more expensive than SIMPLE. In terms of CPU effort, SIMPLEM stands between PRIME and SIMPLE. For all algorithms, use of the PG and FMG methods speeds up acceleration, with the FMG method being more efficient at accelerating the convergence rate, for the problems solved on the densest grid used, over the SG method, by a factor reaching a value as high as 6.55.

INTRODUCTION

Since its development over three decades ago, the SIMPLE (Semi-Implicit Methods for Pressure Linked Equations) algorithm of Patankar and Spalding [1, 2], originally derived for incompressible single-fluid flow, has witnessed a consistent and sustained research effort by many workers. In an attempt to improve its performance in simulating incompressible flows, several variations of this algorithm emerged (e.g. SIMPLEC [3], SIMPLEM [4], SIMPLEX [5], SIMPLEST [6], PISO [7], and PRIME [8] to cite a few), and their collection has been denoted by the computational fluid

Received 29 August 2003; accepted 6 November 2003.
Address correspondence to F. Moukalled, American University of Beirut, Faculty of Engineering & Architecture, Mechanical Engineering Department, P. O. Box 11–0236, Riad El Solh, Beirut 1107 2020, Lebanon. E-mail: memouk@aub.edu.lb

NOMENCLATURE

$A_P^{(k)}, \ldots$	coefficients in the discretized equation for $\phi^{(k)}$
$B_P^{(k)}$	source term in the discretized equation for $\phi^{(k)}$
$\mathbf{B}^{(k)}$	body force per unit volume of fluid/phase k
$C_\rho^{(k)}$	coefficient equals to $1/R^{(k)}T^{(k)}$
$\mathbf{D}_P^{(k)}[\phi^{(k)}]$	the matrix **D** operator
$H_P[\phi^{(k)}]$	the H operator
$\mathbf{H}_P[\mathbf{u}^{(k)}]$	the vector form of the H operator
$\mathbf{I}^{(k)}$	interphase momentum transfer
$\mathbf{J}_f^{(k)D}$	diffusion flux of $\phi^{(k)}$ across cell face f
$\mathbf{J}_f^{(k)C}$	convection flux of $\phi^{(k)}$ across cell face f
$\dot{M}^{(k)}$	mass source per unit volume
P	pressure
$\mathrm{Pr}^{(k)}$, $\mathrm{Pr}_t^{(k)}$	laminar and turbulent Prandtl number of fluid/phase k
$\dot{q}^{(k)}$	heat generated per unit volume of fluid/phase k
$Q^{(k)}$	general source term of fluid/phase k
$r^{(k)}$	volume fraction of fluid/phase k
$R^{(k)}$	gas constant for fluid/phase k
S_f	surface vector
t	time
$T^{(k)}$	temperature of fluid/phase k
$\mathbf{u}^{(k)}$	velocity vector of fluid/phase k
$\mathrm{u}^{(\mathrm{k})}$, $\mathrm{v}^{(\mathrm{k})}$	velocity components of fluid/phase k in x and y directions, respectively
$U_f^{(k)}$	interface flux velocity $\left(\mathbf{v}_f^{(k)} \cdot \mathbf{S}_f\right)$ of fluid/phase k
$\beta^{(k)}$	thermal expansion coefficient for phase/fluid k
$\Gamma^{(k)}$	diffusion coefficient of fluid/phase k
δt	time step
$\Delta_P\left[\phi^{(k)}\right]$	the Δ operator
$\mu^{(k)}, \mu_t^{(k)}$	laminar and turbulent viscosity of fluid/phase k
$\rho^{(k)}$	density of fluid/phase k
$\phi^{(k)}$	general scalar quantity associated with fluid/phase k
$\Phi^{(k)}$	dissipation term in energy equation of fluid/phase k
Ω	cell volume

Subscripts

E	refers to energy equation
k	refers to turbulent kinetic energy equation
nb	refers to the east, west, ... face of a control volume
NB	refers to the East, West, ... neighbors of the main grid point
M	refers to momentum equation
P	refers to the P grid point
ε	refers to turbulent eddy diffusivity equation

Superscripts

C	refers to convection contribution
D	refers to diffusion contribution
(k)	refers to fluid/phase k
$(k)^*$	refers to updated value at the current iteration
$(k)^\circ$	refers to values of fluid/phase k from the previous iteration
$(k)'$	refers to correction field of phase/fluid k
Old	refers to values from the previous time step

dynamics (CFD) community as the SIMPLE family. However, none of these algorithms could be singled out as being universally the best. Rather, their performance was shown to be problem and even grid dependent. To further enhance the performance of the pressure-based approach, CFD researchers sought other alternatives, among which the multigrid technique proved to be promising, with grid-independent convergence characteristics. Consequently, some algorithms in the SIMPLE family were implemented within multigrid strategies [9–12] and resulted in substantial acceleration in the convergence rate for incompressible flow [13], undoubtedly qualifying them as efficient smoothers in multigrid calculations [14].

Due to its advantages over the density-based technique, the pressure-based approach was the focus of further developments to extend its use for predicting fluid flow in the various Reynolds and Mach number regimes. These efforts culminated in several extended versions of the SIMPLE algorithm that are capable of solving

single-fluid flow problems at all speeds [15–21]. A unified formulation of the SIMPLE family of algorithms for the prediction of incompressible and compressible single-fluid flow at all speeds was presented by Moukalled and Darwish [22], while an assessment of their performance within a single grid, a prolongation grid, and a full multigrid methodology was recently carried out by Darwish et al. [23].

In parallel with developments in single-fluid flow algorithms, researchers have exploited the pressure-based method to predict incompressible multiphase flow (e.g., [24–29]). However, unlike the substantial effort spent on developing variations of the single-fluid SIMPLE algorithm, work on multifluid flow algorithms concentrated mostly on the SIMPLE, SIMPLEC, and to a lesser extent PISO, with no attention given to other algorithms, and developments were confined to incompressible applications. The first article dealing with pressure-based algorithms for the prediction of multifluid flow at all speeds was the one reported by Darwish et al. [30]. In their work, it was shown that a multifluid pressure-correction equation can be derived either from the overall mass conservation equation or from the geometric conservation equation. Depending on the route followed, the resulting multifluid algorithms were denoted either mass conservation-based algorithms (MCBA) or geometric conservation-based algorithms (GCBA), respectively.

Assessments of performance in one-dimensional space of the incompressible multifluid versions of the SIMPLE families were reported by Moukalled and Darwish [31, 32]. Moreover, Moukalled et al. [33] extended the MCBA-SIMPLE algorithm to predict multifluid flow at all speeds, while Darwish et al. [34] extended the work of Moukalled et al. [33] into a multigrid framework. With the exception of MCBA-SIMPLE, the MCBA and GCBA families have not been implemented or systematically tested for the prediction of multifluid flow at all speeds. It is the intention of this and a companion article [35] to perform these tasks and implement both families within a single-grid (SG), a prolongation-grid (PG), and a full multigrid (FMG) method.

In what follows, the governing equations for multifluid flow are introduced along with a brief description of the discretization procedure. Then the MCBA approach is outlined, the FMG and PG methods introduced, the capability of the MCBA algorithms to predict multifluid flow phenomena at all speeds demonstrated, and their performance characteristics (in terms of convergence history and speed) assessed. For that purpose, two incompressible, turbulent, two-phase flow problems and two compressible two-phase flow problems encompassing dilute and dense gas–solid and bubbly flows in the subsonic, transonic, and supersonic regimes are solved. In addition, the performance of these algorithms using a single grid, a prolongation grid, and a full multigrid method is compared.

THE GOVERNING EQUATIONS

The equations governing multifluid flows are the conservation laws of mass, momentum, and energy for each individual fluid, which can be written for the kth fluid/phase as

$$\frac{\partial\left(r^{(k)}\rho^{(k)}\right)}{\partial t}+\nabla\cdot\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\right)=r^{(k)}\dot{M}^{(k)} \tag{1}$$

$$\frac{\partial\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\right)}{\partial t}+\nabla\cdot\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\mathbf{u}^{(k)}\right)=\nabla\cdot\left[r^{(k)}(\mu^{(k)}+\mu_t^{(k)})\nabla\mathbf{u}^{(k)}\right]+r^{(k)}\left(-\nabla P+\mathbf{B}^{(k)}\right)+\mathbf{I}_M^{(k)} \tag{2}$$

$$\frac{\partial\left(r^{(k)}\rho^{(k)}T^{(k)}\right)}{\partial t}+\nabla\cdot\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}T^{(k)}\right)=\nabla\cdot\left[r^{(k)}\left(\frac{\mu^{(k)}}{\mathrm{Pr}^{(k)}}+\frac{\mu_t^{(k)}}{\mathrm{Pr}_t^{(k)}}\right)\nabla T^{(k)}\right]+\frac{r^{(k)}}{c_P^{(k)}}\times\left\{\beta^{(k)}T^{(k)}\left[\frac{\partial P}{\partial t}+\nabla\cdot\left(P\mathbf{u}^{(k)}\right)-P\nabla\cdot\left(\mathbf{u}^{(k)}\right)\right]+\Phi^{(k)}+\dot{q}^{(k)}\right\}+\frac{I_E^{(k)}}{c_P^{(k)}} \tag{3}$$

where the superscript (k) refers to the kth fluid and other parameters are as defined in the Nomenclature. Moreover, the mass sources on the right-hand side of Eq. (1) are often nonzero, as when one fluid is transformed to another fluid. However, summation over all fluids leads to the following "global mass conservation" equation, which is used in deriving the pressure-correction equation in the mass conservation formulation:

$$\sum_k\left(\frac{\partial\left(r^{(k)}\rho^{(k)}\right)}{\partial t}+\nabla\cdot\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\right)\right)=0 \tag{4}$$

For high Reynolds number multifluid flow, several models have been advertised for incorporating the effect of turbulence on interfacial mass, momentum, and energy transfer, which vary in complexity from simple algebraic models [36] to state-of-the-art Reynolds-stress models [37]. However, the widely used multiphase turbulence model, adopted in this work, is the two-equation k–ε model [38], in which the fluidic conservation equations governing the turbulence kinetic energy (k) and turbulence dissipation rate (ε) for the kth fluid are given by

$$\frac{\partial\left(r^{(k)}\rho^{(k)}k^{(k)}\right)}{\partial t}+\nabla\cdot\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}k^{(k)}\right)=\nabla\cdot\left(r^{(k)}\frac{\mu_t^{(k)}}{\sigma_k^{(k)}}\nabla k^{(k)}\right)+r^{(k)}\rho^{(k)}\left(G^{(k)}-\varepsilon^{(k)}\right)+I_k^{(k)} \tag{5}$$

$$\frac{\partial\left(r^{(k)}\rho^{(k)}\varepsilon^{(k)}\right)}{\partial t}+\nabla\cdot\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\varepsilon^{(k)}\right)=\nabla\cdot\left(r^{(k)}\frac{\mu_t^{(k)}}{\sigma_\varepsilon^{(k)}}\nabla\varepsilon^{(k)}\right)+r^{(k)}\rho^{(k)}\frac{\varepsilon^{(k)}}{k^{(k)}}\left(c_{1\varepsilon}G^{(k)}-c_{2\varepsilon}\varepsilon^{(k)}\right)+I_\varepsilon^{(k)} \tag{6}$$

where $I_k^{(k)}$ and $I_\varepsilon^{(k)}$ represent the interfacial turbulence terms. The turbulent viscosity is calculated as

$$\mu_t^{(k)} = C_\mu \rho^{(k)} \frac{\left(k^{(k)}\right)^2}{\varepsilon^{(k)}} \tag{7}$$

For two-phase flow, several extensions of the k–ε model that are based on calculating the turbulent viscosity by solving the k and ε equations for the carrier or continuous phase only have been proposed in the literature [39–45]. Due to the lack of better well-developed alternatives, this approach is adopted in this work.

A review of the above differential equations reveals that they are similar in structure. If a typical representative variable associated with phase (k) is denoted by $\phi^{(k)}$, the general fluidic differential equation may be written as

$$\frac{\partial\left(r^{(k)}\rho^{(k)}\phi^{(k)}\right)}{\partial t} + \nabla\cdot\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\phi^{(k)}\right) = \nabla\cdot\left(r^{(k)}\Gamma^{(k)}\nabla\phi^{(k)}\right) + r^{(k)}Q^{(k)} \tag{8}$$

where the expressions for $\Gamma^{(k)}$ and $Q^{(k)}$ can be deduced from the parent equations.

The above set of differential equations has to be solved in conjunction with constraints on certain variables represented by algebraic relations. These auxiliary relations include the equations of state, the geometric conservation equation, and the interfacial mass, momentum, energy, and turbulence energy transfers. Details regarding the auxiliary relations and closures used in this work were reported in [33, 34] and are not repeated here, for compactness.

DISCRETIZATION PROCEDURE

The first step consists of integrating the general conservation equation [Eq. (8)] over a finite cell (Figure 1a) of volume Ω to yield

$$\iint_\Omega \frac{\partial\left(r^{(k)}\rho^{(k)}\phi^{(k)}\right)}{\partial t}\,d\Omega + \iint_\Omega \nabla\cdot\left(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\phi^{(k)}\right)d\Omega = \iint_\Omega \nabla\cdot\left(r^{(k)}\Gamma^{(k)}\nabla\phi^{(k)}\right)d\Omega + \iint_\Omega r^{(k)}Q^{(k)}\,d\Omega \tag{9}$$

Using the divergence theorem to transform the volume integral into a surface integral and then replacing the surface integral by a summation of the fluxes over the sides of the control volume, Eq. (9) is transformed to

$$\frac{\partial\left(r^{(k)}\rho^{(k)}\phi^{(k)}\Omega\right)}{\partial t} + \sum_{\text{nb}}\left(\mathbf{J}_{\text{nb}}^{(k)D} + \mathbf{J}_{\text{nb}}^{(k)C}\right) = r^{(k)}Q^{(k)}\Omega \tag{10}$$

where $\mathbf{J}_{\text{nb}}^{(k)D}$ and $\mathbf{J}_{\text{nb}}^{(k)C}$ are the diffusive and convective fluxes, respectively. The discretization of the diffusion term is second-order accurate and follows the derivations presented in [46]. For the convective terms, the high-resolution (HR) SMART [47] scheme is employed, even for calculating interface densities [21], and applied within the context of the NVSF methodology [48]. Substituting the face values by their

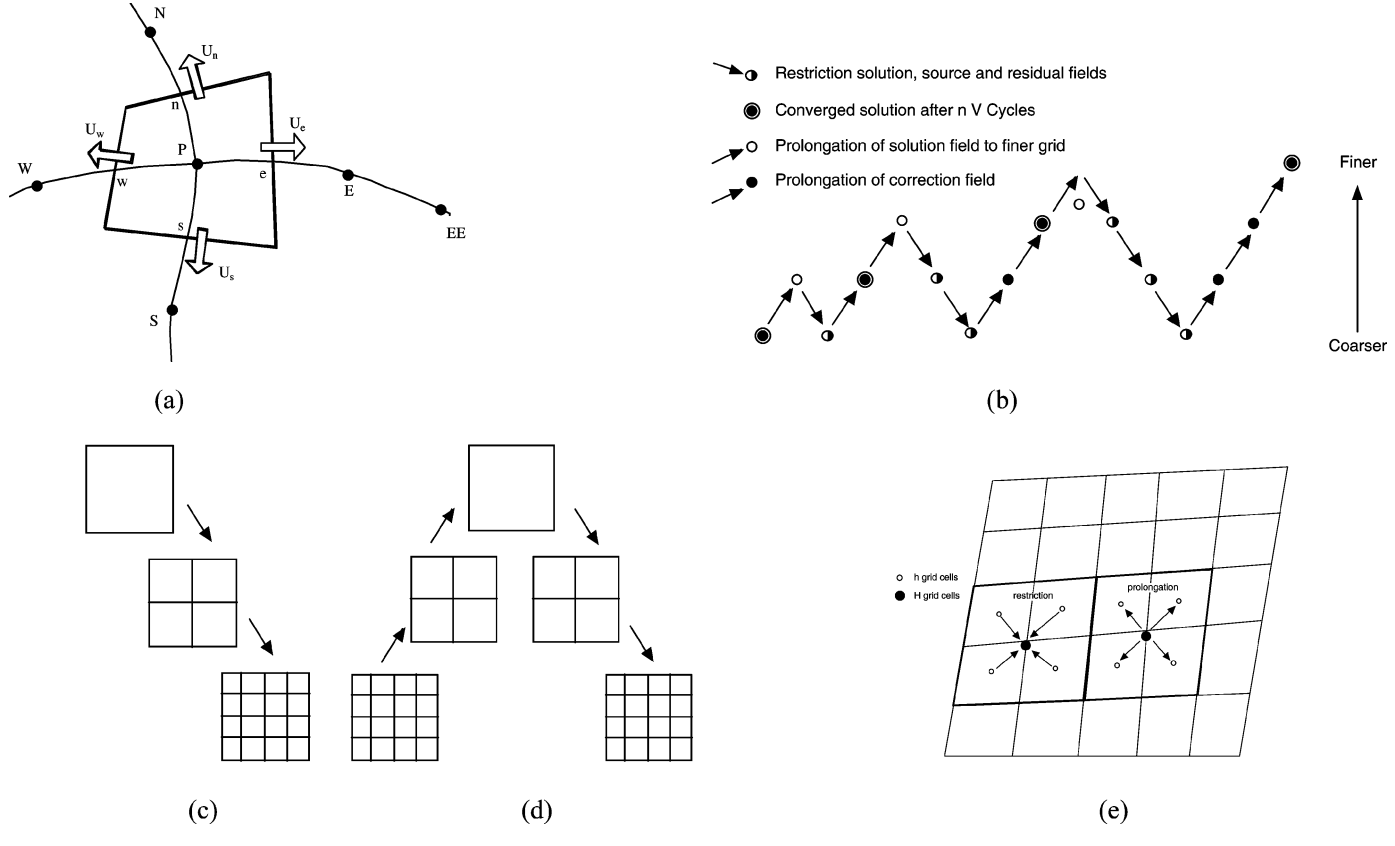


Figure 1. (*a*) Control volume. (*b*) Full multigrid V cycles. (*c*) The prolongation-only strategy. (*d*) The MG strategy. (*e*) Prolongation and restriction.

functional relationship relating to the node values of ϕ, Eq. (10) is transformed after some algebraic manipulations into the following discretized equation:

$$A_P^{(k)}\phi_P^{(k)} = \sum_{\mathrm{NB}} A_{\mathrm{NB}}^{(k)}\phi_{\mathrm{NB}}^{(k)} + B_P^{(k)} \tag{11}$$

where the coefficients $A_P^{(k)}$ and $A_{\mathrm{NB}}^{(k)}$ depend on the selected scheme and $B_P^{(k)}$ is the source term of the discretized equation . In compact form, the above equation can be written as

$$\phi^{(k)} = H_P\left[\phi^{(k)}\right] = \left(\sum_{\mathrm{NB}} A_{\mathrm{NB}}^{(k)}\phi_{\mathrm{NB}}^{(k)} + B_P^{(k)}\right) \Big/ A_P^{(k)} \tag{12}$$

In a similar manner, the discretized form of the momentum equation can be written as

$$\mathbf{u}_P^{(k)} = \mathbf{H}_P\left[\mathbf{u}^{(k)}\right] - r^{(k)}\mathbf{D}_P^{(k)}\nabla_P(P) \tag{13}$$

THE MCBA PRESSURE-CORRECTION EQUATION

The number of equations describing an n-fluid flow situation is: n fluidic momentum equations, n fluidic volume fraction (or mass conservation) equations, a geometric conservation equation, and for the case of a compressible flow an additional n auxiliary pressure–density relations. Moreover, the variables involved are the n fluidic velocity vectors, the n fluidic volume fractions, the pressure field, and for a compressible flow an additional n unknown fluidic density fields. In all MCBA algorithms, the n momentum equations are used to calculate the n velocity fields, $n-1$ volume fraction (mass conservation) equations are employed to calculate $n-1$ volume fraction fields, and the last volume fraction field is calculated utilizing the geometric conservation equation

$$r^{(n)} = 1 - \sum_{k \neq n} r^{(k)} \tag{14}$$

The remaining volume fraction equations can be used to calculate the shared pressure field, which does not have an explicit equation. However, instead of using this last volume fraction equation, the global mass conservation equation [Eq. (4)] is employed in combination with the fluidic momentum equations, to derive a pressure-correction equation. For that purpose, a guessed pressure field (P°) is substituted into the momentum equations. The resulting velocity fields, denoted by $\mathbf{u}^{(k)^*}$, which now satisfy the momentum equations, will not, in general, satisfy the mass conservation equations. Thus, corrections are needed in order to yield velocity, pressure, and density fields that satisfy both sets of equations. Denoting the corrections for pressure, velocity, and density by P', $\mathbf{u}^{(k)'}$, and $\rho^{(k)'}$, respectively, the corrected fields are written as

$$P = P^\circ + P' \qquad \mathbf{u}^{(k)} = \mathbf{u}^{(k)^*} + \mathbf{u}^{(k)'} \qquad \rho^{(k)} = \rho^{(k)^\circ} + \rho^{(k)'} \tag{15}$$

Hence the equations solved in the predictor stage are

$$\mathbf{u}_P^{(k)^*} = \mathbf{H}_P\left(\mathbf{u}^{(k)^*}\right) - r_P^{(k)\circ}\mathbf{D}_P^{(k)}\nabla_P P^\circ \tag{16}$$

While the final solutions satisfy

$$\mathbf{u}_P^{(k)} = \mathbf{H}_P\left(\mathbf{u}^{(k)}\right) - r_P^{(k)\circ}\mathbf{D}_P^{(k)}\nabla_P P \tag{17}$$

Subtracting the two equation sets [(16) and (17)] from each other yields the following equation involving the correction terms:

$$\mathbf{u}_P^{(k)'} = \mathbf{H}_P\left(\mathbf{u}^{(k)'}\right) - r_P^{(k)\circ}\mathbf{D}_P^{(k)}\nabla_P P' \tag{18}$$

Moreover, the new density and velocity fields, $\rho^{(k)}$ and $\mathbf{u}^{(k)}$, will satisfy the overall mass conservation equation if

$$\sum_k \left[\frac{\left(r_P^{(k)\circ}\rho_P^{(k)}\right) - \left(r_P^{(k)}\rho_P^{(k)}\right)^{\text{Old}}}{\delta t}\Omega_P + \Delta_P\left(r^{(k)\circ}\rho^{(k)}\mathbf{u}^{(k)}.\mathbf{S}\right)\right] = 0 \qquad \text{where}$$

$$\Delta_P(\Phi) = \sum_{\text{nb}} \Phi \tag{19}$$

Linearizing the $(\rho^{(k)}\mathbf{u}^{(k)})$ term, one gets

$$\left(\rho^{(k)^*} + \rho^{(k)'}\right)\left(\mathbf{u}^{(k)^*} + \mathbf{u}^{(k)'}\right) = \rho^{(k)^*}\mathbf{u}^{(k)^*} + \rho^{(k)^*}\mathbf{u}^{(k)'} + \rho^{(k)'}\mathbf{u}^{(k)^*} + \rho^{(k)'}\mathbf{u}^{(k)'} \tag{20}$$

Substituting Eqs. (20) and (18) into Equation (19), rearranging, and replacing density correction by pressure correction, the final form of the pressure-correction equation is written as

$$\sum_k \left\{\frac{\Omega_P}{\delta t} r_P^{(k)^\circ} C_\rho^{(k)} P'_P + \Delta_P\left(r^{(k)^\circ} U^{(k)^*} C_\rho^{(k)} P'\right) - \Delta_P\left[r^{(k)^\circ}\rho^{(k)^*}\left(r^{(k)^\circ}\mathbf{D}^{(k)}\nabla P'\right).\mathbf{S}\right]\right\}$$

$$= -\sum_k \begin{pmatrix} \frac{r_P^{(k)\circ}\rho_P^{(k)^*} - \left(r_P^{(k)}\rho_P^{(k)}\right)^{\text{Old}}}{\delta t}\Omega_P + \Delta_P\left(r^{(k)\circ}\rho^{(k)^*}U^{(k)^*}\right) \\ +\Delta_P\left[r^{(k)^\circ}\rho^{(k)^*}\left[\mathbf{H}\left(\mathbf{u}^{(k)'}\right)\right]\cdot\mathbf{S}\right] + \Delta_P\left(r^{(k)\circ}\rho^{(k)'}\mathbf{u}^{(k)'}.\mathbf{S}\right) \end{pmatrix} \tag{21}$$

The corrections applied to the velocity, pressure, and density fields are given by

$$\mathbf{u}_P^{(k)^*} = \mathbf{u}_P^{(k)\circ} - r_P^{(k)\circ}\mathbf{D}_P^{(k)}\nabla_P P' \qquad P^* = P^\circ + P' \qquad \rho^{(k)^*} = \rho^{(k)\circ} + C_\rho^{(k)} P' \tag{22}$$

If the $\mathbf{H}(\mathbf{u}^{(k)'})$ term in Eq. (21) is retained, there will result a pressure-correction equation relating the pressure correction value at a point to all values in the domain. To facilitate implementation and reduce cost, simplifying assumptions related to this term have been introduced. Depending on these assumptions, different algorithms are obtained. A summary of the various MCBA algorithms used in this work was accorded a full-length article, to which interested reader is referred [30].

The sequence of events in the MCBA is as follows:

1. Solve the fluidic momentum equations for velocities.
2. Solve the pressure-correction equation based on global mass conservation.
3. Correct velocities, densities, and pressure.
4. Solve the fluidic mass conservation equations for volume fractions.
5. Solve the fluidic energy equations.
6. Return to the first step and repeat until convergence.

THE NONLINEAR MULTIGRID STRATEGY

The above-described procedure applies to a single-grid method, for which the rate of convergence is high during the very first iterations but degrades thereafter, and the situation gets worse as the number of grid points increases. This behavior is attributed to the smoothing property of the algorithms [13], which are efficient in removing only the high-frequency components of the error. The low-frequency components, which have long wavelength compared to the grid spacing, are not properly resolved. This is clearly noticed when using any of the solvers on progressively denser grids, in which case the convergence rate decreases more rapidly on the finer grids. The performance of these algorithms can thus be substantially improved by combination with the multigrid method.

In the multigrid approach the high-frequency errors are initially eliminated on the finest grid, then, when the convergence rate degrades, the process is repeated on a coarser grid, where part of the low-frequency error components of the finer grid are transformed into high-frequency error components on the coarser grid, which can be efficiently removed. This step is recursively applied on coarser grids, and more of the error components are reduced. Results are then interpolated back from coarser to finer grids.

The V-cycle multigrid strategy (Figure 1*b*) adopted in this work starts at the coarsest level, where the solution is first computed; this solution is interpolated onto the next finer mesh, where it is used as initial guess. This stage is called the prolongation stage (see Figure 1*c*). Then iterations are performed on the fine mesh and the solution is transferred back to the coarser mesh by applying a restriction operator and adding forcing terms so that the truncation errors on both the fine and coarse grids are of the same order. After performing a number of iterations on the coarse mesh, the solution is transferred back to the finer mesh in the form of a correction, and a number of iterations are performed on the finer grid to smooth the fields. This process is continued until a converged solution on the fine mesh is obtained (see Figure 1*d*). Then the solution is extrapolated into a finer mesh and the process repeated until convergence is reached on the desired finest mesh.

In the restriction step (Figure 1*e*), the coarse-grid variables are computed from the fine-grid values as

$$\tilde{\phi}_C = \frac{1}{4} \sum_{i=1\ldots4} \left(\phi_{F_i} + \nabla \phi_{F_i} \cdot \mathbf{d}_{F_i C} \right) \tag{23}$$

while in the prolongation step (Figure 1*e*), the fine-grid corrections are computed from the coarse-grid corrections as

$$\phi'_{F_i} = \phi'_C + \nabla \phi'_C \cdot \mathbf{d}_{CF_i} \tag{24}$$

where

$$\phi'_C = \phi_C - \tilde{\phi}_C \tag{25}$$

The special character of the volume fraction and k–ε equations necessitates modification to the prolongation procedure. For the k–ε turbulence model, the remedy suggested by Cornelius et al. [49] is adopted. For the volume-fraction equations, the following treatment is applied: While extrapolating the volume-fraction field from the coarse to the fine grid, the prolongation operator may yield negative volume-fraction values or values that are greater than one. Such unphysical values are detrimental to the overall convergence rate and may even cause divergence. To circumvent this problem, a simple yet very effective treatment is adopted: Once the r values are extrapolated, a check is performed to make sure they are within bounds. If any of the r values is found to be unbounded, the r fluidic volume-fraction equation is solved starting with the interpolated values until all of the r values are within the set bounds. Typically, fewer than 10 iterations are needed. This treatment has been found to be very effective and to preserve the convergence acceleration rate.

In addition to the FMG strategy, the PG approach is also tested. This approach differs from the FMG method in that the solution moves in one direction from the coarse to the fine grids, with the initial guess on level $n+1$ obtained by interpolation from the converged solution on level n (Figure 1c). As such, the acceleration over the SG method obtained with this approach is an indication of the effect of initial guess on convergence.

RESULTS AND DISCUSSION

The performance of the various multifluid mass conservation-based algorithms is assessed in this section by presenting solutions to four two-dimensional, two-phase flow problems. The first two tests deal with incompressible turbulent flows, while the last two are concerned with compressible flows. As mentioned earlier, solutions are obtained using the prolongation-grid scheme (PG) and the full multigrid method (FMG) in addition to the usual single-grid (SG) approach. Results are presented in terms of the CPU time needed to converge the solution to a set level and of the convergence history. Moreover, solutions are generated for a number of grids in order to assess the performance of the various algorithms with increasing grid density. Results are compared against available experimental data and/or numerical/theoretical values. The residual of a variable ϕ at the end of an outer iteration is defined as

$$\mathrm{RES}_\phi^{(k)} = \sum_{c.v} \left| A_p \phi_p^{(k)} - \sum_{\substack{\text{all } p \\ \text{neighbors}}} A_{\mathrm{nb}} \phi_{\mathrm{nb}}^{(k)} - B_p^{(k)} \right| \tag{26}$$

For global mass conservation, the imbalance in mass is given by

$$\mathrm{RES}_C = \sum_k \sum_{c.v.} \left| \frac{\left(r_P^{(k)} \rho_P^{(k)}\right) - \left(r_P^{(k)} \rho_P^{(k)}\right)^{\mathrm{Old}}}{\delta t} \Omega - \Delta_P \left[r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S} \right] - r^{(k)} \dot{M}^{(k)} \right| \tag{27}$$

All residuals are normalized by their respective inlet fluxes. Computations are terminated when the maximum normalized residuals of all variables drop below a very small number ε_s. In all problems, the first phase represents the continuous phase [denoted by a superscript (c)], which must be fluid, and the second phase is the disperse phase [denoted by a superscript (d)], which may be solid or fluid. Unless otherwise specified, the HR SMART scheme is used in all computations reported in this study. For a given problem, all results are generated starting from the same initial guess. Moreover, the computational details of the various problems considered were discussed in [33, 34], to which the reader is referred. Thus, only a brief description of the physical phenomena is given here.

Problem 1: Turbulent Upward Bubbly Flow in a Pipe

In this problem, the radial phase distribution of turbulent upward air–water flow in a pipe is predicted. Many experimental and numerical studies addressing this problem have appeared in the literature [33, 34, 50–53]. The case considered here reproduces numerically the experimental data reported by Seriwaza et al. [50], for which the Reynolds number based on superficial liquid velocity and pipe diameter is 8×10^4, the inlet superficial gas and liquid velocities are 0.077 and 1.36 m/s, respectively, and the inlet void fraction is 5.36×10^{-2} with no slip between the incoming phases. Moreover, the bubble diameter is taken as 3 mm [53], while the fluid properties are set to $\rho^{(c)} = 1{,}000\ \text{kg/m}^3$, $\rho^{(d)} = 1.23\ \text{kg/m}^3$, and $\nu_l^{(c)} = 10^{-6}\ \text{m}^2/\text{s}$. Predicted radial profiles of the vertical liquid velocity and void fraction presented in Figure 2 using a grid of size 96×32 control volumes concur very well with measurements [50] and compare favorably with numerical profiles reported by Boisson and Malin [53].

To assess the merits of the various algorithms for such flows, calculations are performed on four different grids of sizes 12×4, 24×8, 48×16, and 96×32 control volumes. On each grid, solutions are generated using the SG, PG, and FMG strategies for all algorithms. Results are displayed in the form of (1) total mass residuals summed over both phases as a function of outer iterations, and (2) normalized CPU time (Table 1) needed for the maximum normalized residuals of all variables and for all phases to drop below $\varepsilon_s = 10^{-6}$. In order not to overload plots, only residuals over the densest grid using the SG, PG, and FMG methodologies are presented in Figure 3. Moreover, it should be clarified that in the case of PISO (Figure 3*a*) and PRIME (Figure 3*f*), the number of iterations needed to satisfy the convergence criteria is higher than the one presented (this will be reflected by their respective normalized CPU times displayed in Table 1), due to the slower convergence rate of the axial liquid momentum and turbulence kinetic energy equations; i.e., the convergence rate of these equations is slower than the overall mass conservation equation. Since the overall mass residuals are presented, only the iterations needed to decrease these residuals to the required level (10^{-6}) are displayed, even though at the state of convergence these residuals are much lower (of the order of 10^{-9}).

As can be seen from Figure 3, it is possible to converge the solution to the desired level with all algorithms. With the single-grid method, the SIMPLEM algorithm (Figure 3*d*) requires a considerably larger number of iterations in comparison with SIMPLE. Moreover, the performance of SIMPLE (Figure 3*b*),

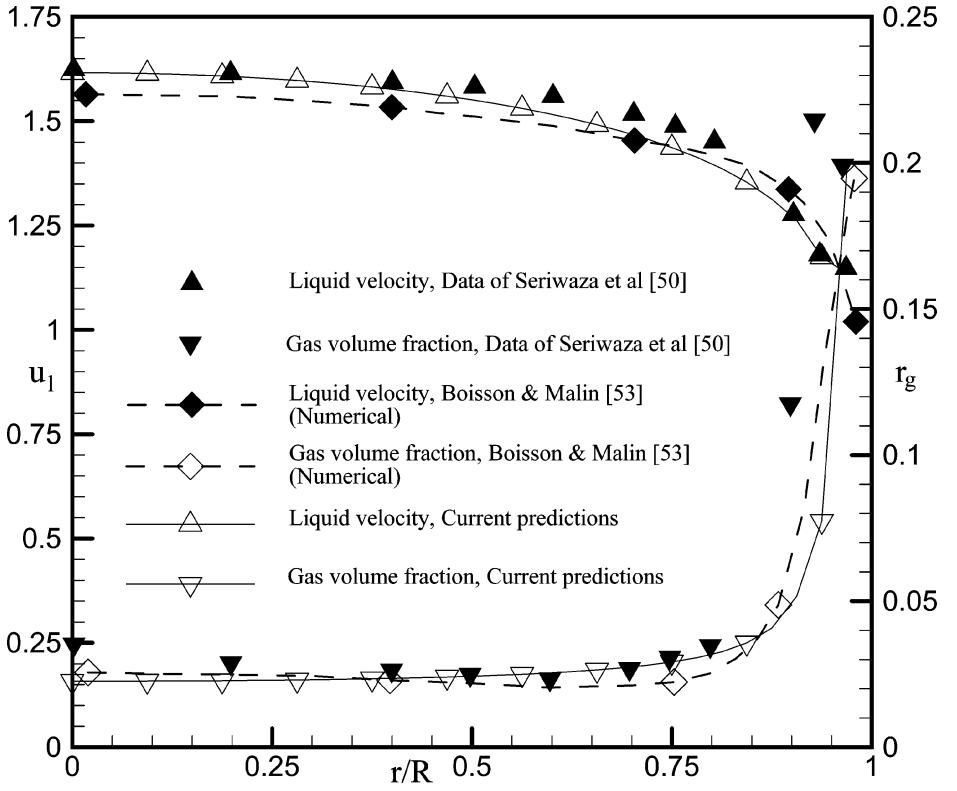


Figure 2. Comparison of fully developed liquid velocity and void fraction profiles for turbulent bubbly upward bubbly flow in a pipe.

Table 1. Normalized CPU times for turbulent bubbly flow in a pipe

		Algorithms						
Grid	Method	SIMPLE	SIMPLEC	SIMPLEX	SIMPLEST	SIMPLEM	PISO	PRIME
12×4 C.V.	SG	1.00	0.97	0.98	1.02	1.27	0.91	1.62
24×8 C.V.	SG	3.30	3.43	3.42	3.79	5.54	3.77	5.76
	PG (2 levels)	3.62	3.53	3.58	3.97	5.70	3.87	6.21
	SG/PG	0.91	0.97	0.95	0.96	0.97	0.97	0.93
	FMG (2 levels)	3.10	3.04	3.12	4.68	5.34	8.81	7.06
	SG/FMG	1.07	1.13	1.10	0.81	1.04	0.43	0.82
48×16 C.V.	SG	25.34	25.71	26.19	30.52	49.17	33.43	46.42
	PG (3 levels)	21.92	22.37	23.36	24.27	44.43	70.17	85.56
	SG/PG	1.16	1.15	1.12	1.26	1.11	0.48	0.54
	FMG (3 levels)	14.44	13.55	14.23	21.74	22.02	45.61	45.36
	SG/FMG	1.75	1.90	1.84	1.40	2.23	0.73	1.02
96×32 C.V.	SG	1231.38	1238.38	1296.28	1249.46	2038.94	1934.27	2059.89
	PG (4 levels)	978.29	978.84	1037.55	1002.88	1593.90	1730.55	1853.99
	SG/PG	1.26	1.27	1.25	1.25	1.28	1.12	1.11
	FMG (4 levels)	207.61	208.46	217.84	214.60	836.08	352.91	314.29
	SG/FMG	5.93	5.94	5.95	5.82	2.44	5.48	6.55

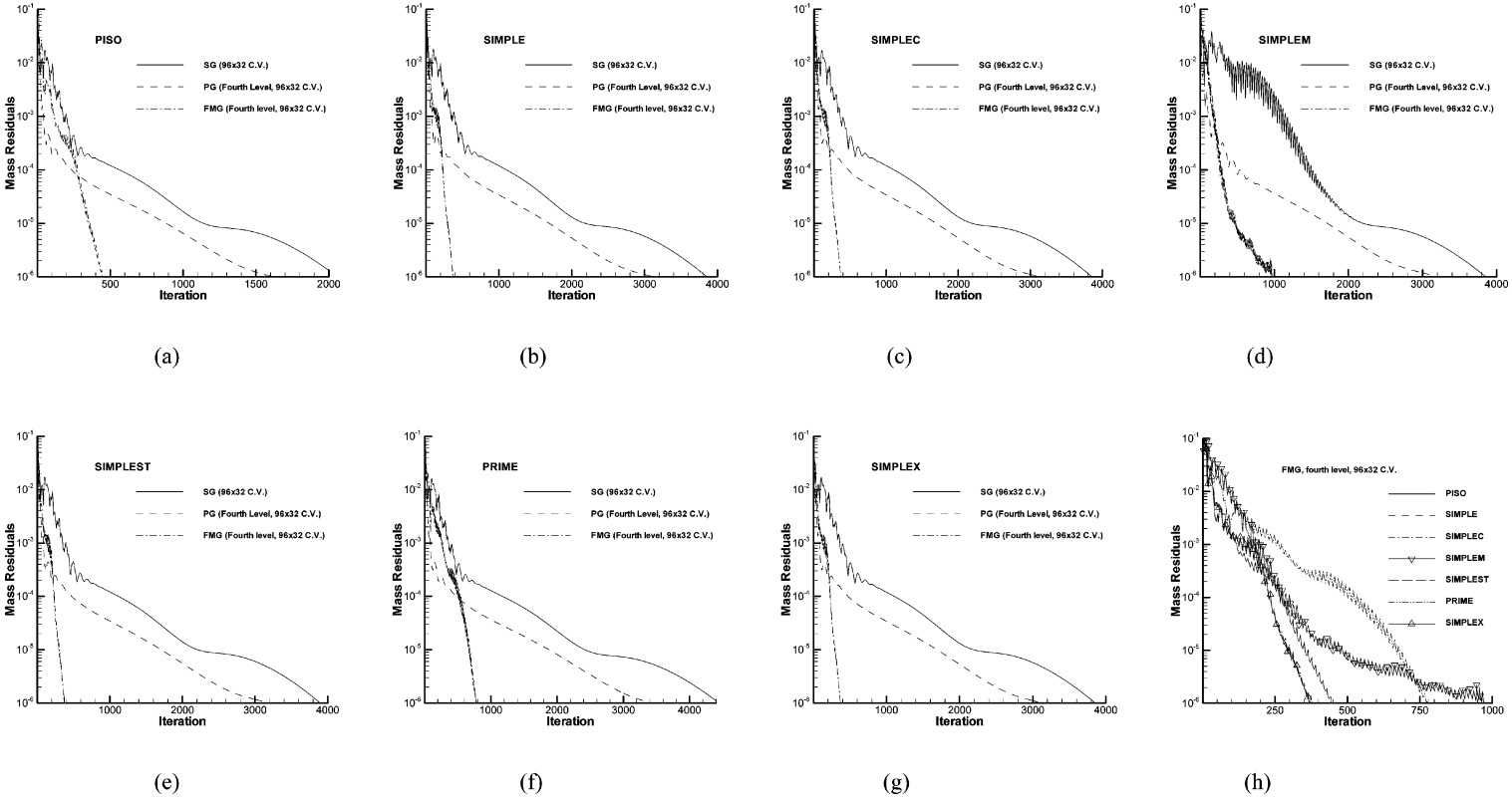


Figure 3. (*a*)–(*g*) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (*h*) convergence histories of the various algorithms on the finest mesh using the FMG method for turbulent upward bubbly flow in a pipe.

SIMPLEC (Figure 3*c*), SIMPLEST (Figure 3*e*), and SIMPLEX (Figure 3*g*) is very close. This is also demonstrated in Figure 3*h*, where the convergence histories using the FMG method are plotted for all algorithms. As shown, the plots for these algorithms are hard to distinguish. For all algorithms, the SG requires the largest number of iterations and the FMG the lowest number of iterations. The FMG method greatly reduces the number of iterations, whereas reduction with the PG method is not as significant. As shown in the plots, the rate of convergence of the SG and PG methods diminishes as the iterations proceed. This is in contrast with the convergence behavior of the FMG method, which seems to maintain, more or less, the same convergence rate (on the finest level) as the iterations progress.

In Table 1, the normalized CPU times required by the different algorithms to solve the problem on the various grids and with the various methodologies are presented. In addition, the ratio of the time needed by the SG method to the one needed by the PG and the FMG method are displayed. This allows for a direct quantitative assessment of their acceleration rate. As depicted, the CPU effort increases for all methods and algorithms with increasing the grid size. The CPU times of SIMPLE, SIMPLEC, SIMPLEX, and SIMPLEST are very close on all grids and for all methods, with no clear superiority of any algorithm over the others. PISO requires the least computational time on the coarsest grid used (12×4 c.v.); however, as the grid size increases, the picture is reversed and its performance degrades and becomes closer to that of PRIME and SIMPLEM. The use of the PG method on the relatively coarse grids increases the computational cost. The benefit of the PG method is realized on the relatively dense grids. On the densest grid, the savings in computational effort varies from 11% for PRIME to 28% for SIMPLEM, with an average acceleration rate for all algorithms of 22%. The use of the FMG method is highly beneficial and its virtues generally appear on all grids, with the rate of acceleration increasing with increasing grid density. On the finest grid, the rate of acceleration varies from 144% for SIMPLEM to 555% for PRIME, with an average acceleration rate for all algorithms of 444%. This average rate of acceleration is nearly 20 times the one obtained with the PG method. This indicates that the nonlinearity in multifluid flows is properly resolved using the nonlinear multigrid method.

Problem 2: Turbulent Air–Particle Flow in a Vertical Pipe

The problem consists of predicting the upward flow of a dilute gas–solid mixture in a vertical pipe [54–56]. For these simulations, the axisymmetric form of the gas and particulate transport equations are employed. The configuration selected is the one experimentally investigated by Tsuji et al. [54], for which the pipe diameter is 30.5 mm, the Reynolds number based on the pipe diameter is 3.3×10^4, the mean air inlet velocity is 15.6 m/s, and the particle diameter and density are 200 μm and 1,020 kg/m^3, respectively. In the computations, the mass-loading ratio at the inlet is considered to be 1, with no slip between the phases. Figure 4 shows the fully developed gas and particle mean axial velocity profiles generated using a grid of size 96×40 C.V. It is evident that there is generally very good agreement between the predicted and experimental data, with the gas velocity being slightly overpredicted and the particle velocity slightly underpredicted. Moreover, close to the wall, the

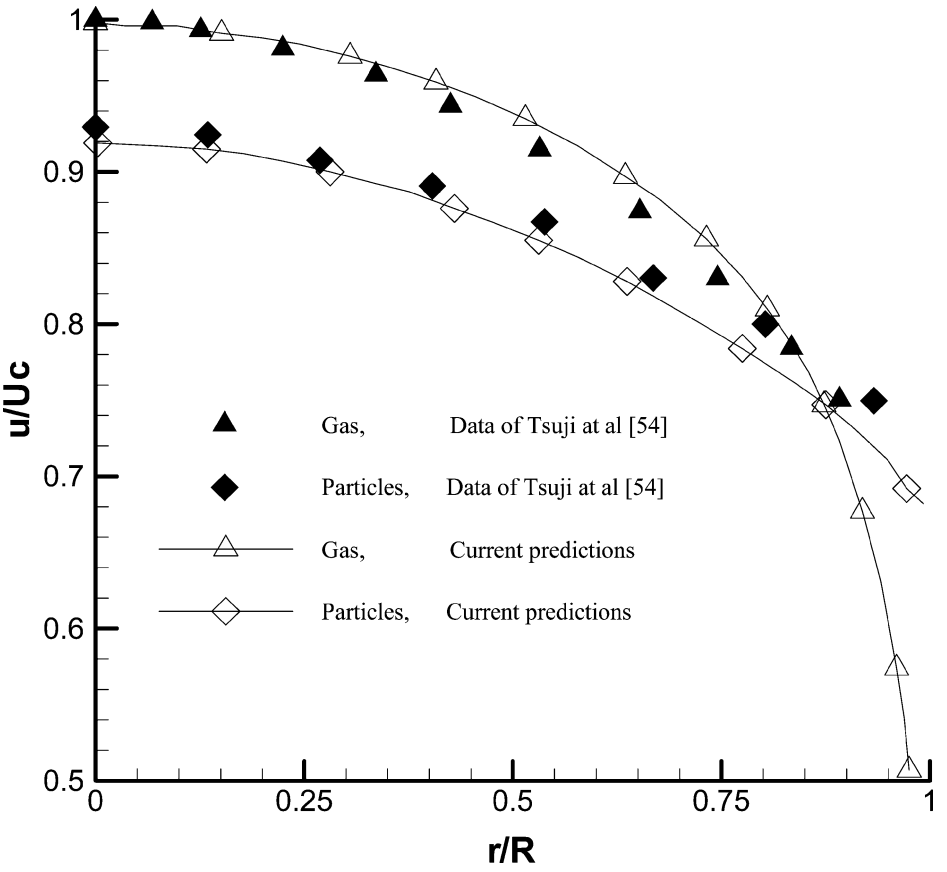


Figure 4. Comparison of fully developed gas and particle velocity profiles for turbulent air–particle flow in a pipe.

model predictions indicate that the particles have higher velocities than the gas, which is in accord with the experimental results of Tsuji et al. [54].

The problem is solved over four different grids of sizes 12×5, 24×10, 48×20, and 96×40 control volumes using the SIMPLE, SIMPLEC, SIMPLEX, and SIMPLEST multifluid algorithms and the SG, PG, and FMG solution methods. In solving the problem, it is noticed that the initial guess greatly affects the convergence history and time required to converge the solution to the desired level. Except when solving on the finest mesh with the SG method, the initial guess used for the velocity field is $u^{(c)} = u^{(d)} = 1\,\mathrm{m/s}$. The use of this initial value with the SG method on the finest mesh increased the CPU effort needed by 500% over the one needed when starting with an initial field of $u^{(c)} = u^{(d)} = 15.6\,\mathrm{m/s}$. To reduce cost, the latter initial guess is used. For this reason the mass residuals start from a somewhat lower value than expected. As in the previous problem, results are displayed in the form of (1) total mass residuals summed over both phases as a function of outer iterations (Figure 5), and (2) normalized CPU time (Table 2) needed for the maximum

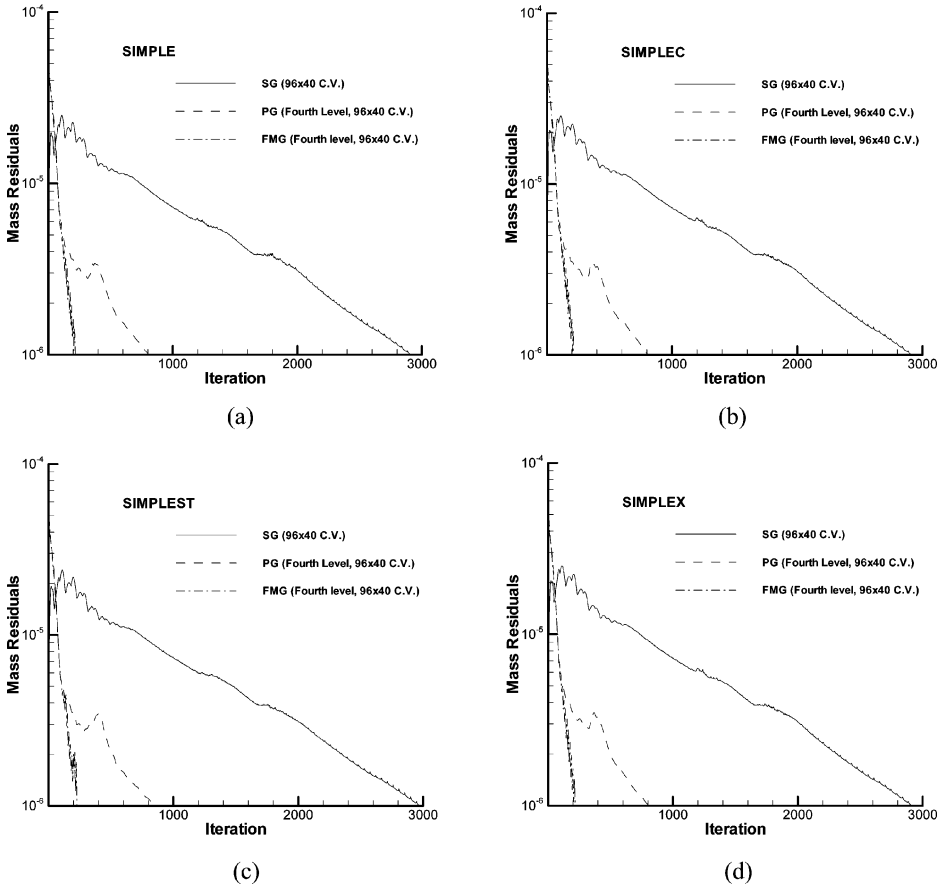


Figure 5. Convergence histories of (*a*) SIMPLE, (*b*) SIMPLEC, (*c*) SIMPLEST, and (*d*) SIMPLEX algorithms using the SG, PG, and FMG methods on the finest mesh for turbulent air–particle flow in a pipe.

normalized residuals of all variables and for all phases to drop below $\varepsilon_s = 10^{-6}$. In order not to overload plots, only residuals over the densest grid using the SG, PG, and FMG methodologies are presented. As shown in Figures 5(*a*)–5(*d*), the convergence histories and the number of iterations required by the various methods and algorithms are nearly identical. In addition, the PG and FMG methods greatly reduce the number of iterations, with the number needed by the PG method being around four times the number entailed by the FMG method. The slopes of the various curves displayed in Figure 5, which are clearly much higher for the FMG method, further reveal this.

The normalized CPU times consumed by the four algorithms to solve the problem on the various grids and with the various methodologies, which are presented in Table 2, indicate an increase with increasing the grid size. On the dense grids, the CPU time required by the four algorithms is very close, with no appreciable superiority of any one. Using the PG method on the finest mesh accelerates the

Table 2. Normalized CPU times for turbulent air–particle flow in a pipe

		Algorithms			
Grid	Method	SIMPLE	SIMPLEC	SIMPLEX	SIMPLEST
12×5 C.V.	SG	1.00	0.97	0.99	1.36
24×10 C.V.	SG	6.75	6.78	7.09	6.9
	PG (2 levels)	5.82	5.76	5.93	5.99
	SG/PG	1.16	1.18	1.20	1.15
	FMG (2 levels)	4.96	4.43	5.18	6.36
	SG/FMG	1.36	1.53	1.37	1.08
48×20 C.V.	SG	73.46	73.11	73.09	71.67
	PG (3 levels)	59.41	59.84	62.78	62.75
	SG/PG	1.24	1.22	1.16	1.14
	FMG (3 levels)	30.28	28.99	30.38	31.37
	SG/FMG	2.43	2.52	2.07	2.28
96×40 C.V.	SG	829.55	831.40	879.20	841.92
	PG (4 levels)	657.14	663.89	703.07	701.53
	SG/PG	1.26	1.25	1.25	1.20
	FMG (4 levels)	164.88	165.76	173.60	171.88
	SG/FMG	5.03	5.02	5.06	4.90

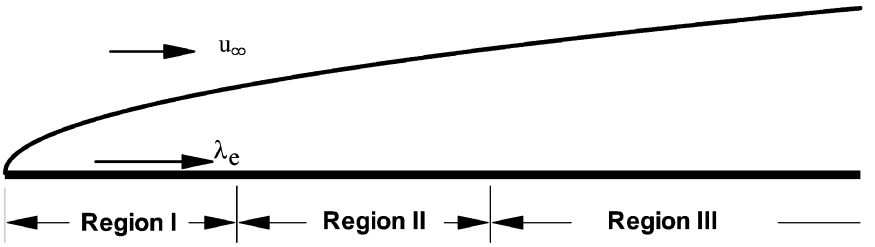
convergence by an average factor of 1.24, whereas the FMG does a much better job of accelerating the solution, by a factor of around 5.

Problem 3: Compressible Dusty Flow over a Flat Plate

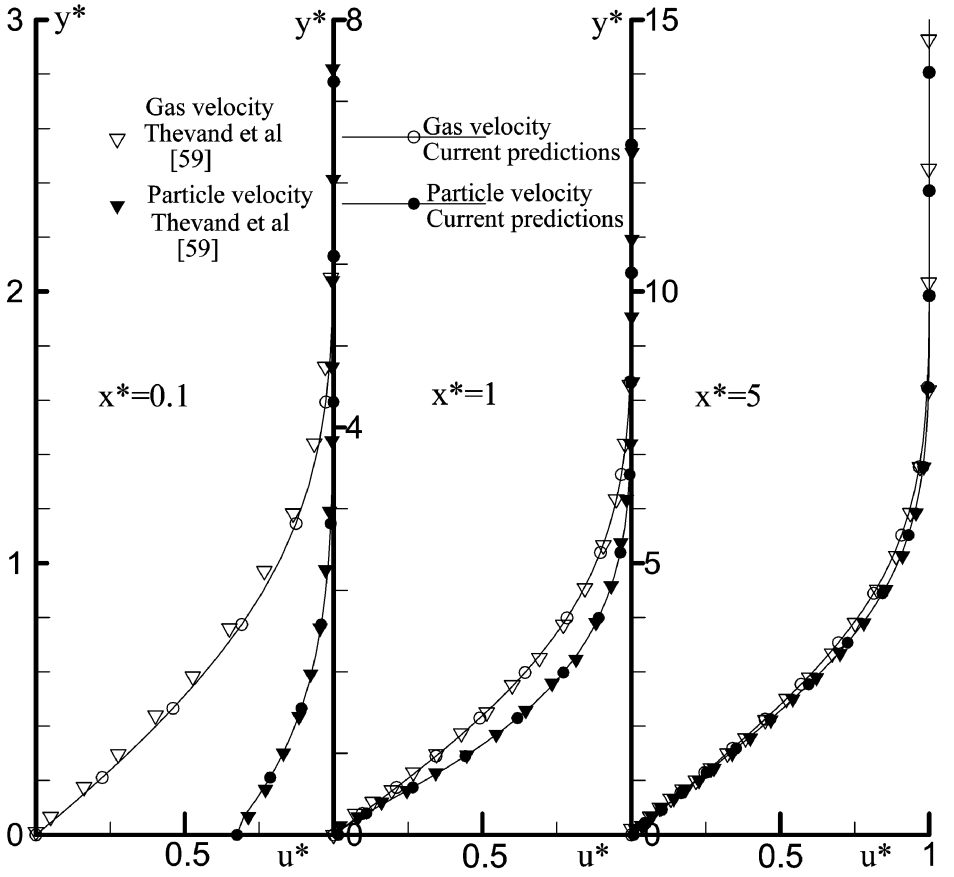
This problem has been used by several researchers [57–59] to check their computational methodologies and is used here for the same purpose. It is known that two-phase flow greatly changes the main features of the boundary layer over a flat plate. Typically, three different regions are defined in the two-phase boundary layer (Figure 6*a*), which can be distinguished by the relative velocity between the two phases: a large-slip region close to the leading edge, a moderate-slip region farther down, and a small-slip one far downstream. It has also been established [58] that the characteristic scale in this two-phase flow problem is the relaxation length λ_e, defined as

$$\lambda_e = \frac{2}{9} \frac{\rho^{(d)} r_p^2 u_\infty}{\mu^{(c)}} \tag{28}$$

where $\rho^{(d)}$ and r_p are, respectively, the density and radius of the particles, $\mu^{(c)}$ is the viscosity of the fluid, and u_∞ is the free-stream velocity. The three regions are defined according to the order of magnitude of the slip parameter $x^* = x/\lambda_e$. In the simulation, the particle diameter is chosen to be 10 μm, the particle Reynolds number is assumed to be equal to 10, and the material density is 1,766 kg/m^3. The Prandtl number is equal to 0.75. The south wall is treated as a no-slip wall boundary for the gas phase (both components of the gas velocity are set to zero) and a slip wall condition for the particles phase (the normal fluxes are set to zero). The gas and the particles enter the computational domain under thermal and dynamical equilibrium



(a)



(b)

Figure 6. The three different regions within the boundary layer of dusty flow over a flat plate. Comparison of fully developed gas and particle velocity profiles inside the boundary layer at different axial locations for dilute two-phase flow over a flat plate.

conditions. A mass load ratio of 1 between the particle phase and the gas phase is used.

Figure 6*b* shows the results for the steady flow obtained on a rectangular domain with a mesh of density 104×48 C.V. stretched in the y direction. The figure provides the development of gas and particle velocity profiles within the three regions mentioned earlier. In the near-leading-edge area ($x^* = 0.1$), the gas velocity is adjusted at the wall to obtain the no-slip condition as for the case of a pure gas boundary layer. The particles have no time to adjust to the local gas motion and there is a large velocity slip between the phases. In the transition region ($x^* = 1$), significant changes in the flow properties take place. The interaction between the phases cause the particles to slow down, while gas accelerate as apparent in the plots. In the far downstream region ($x^* = 5$), the particles have enough time to adjust to the state of the gas motion. The slip is very small and the solution tends to equilibrium. These results are in excellent agreement with numerical solutions reported by Thevand et al. [60], plotted in Figure 6*b*, which gives credibility to the proposed methodology.

The problem is then solved over four different grids of sizes, 13×6, 26×12, 52×24, and 104×48 control volumes, using all multifluid algorithms and the SG, PG, and FMG solution methods. As in the previous two problems, results are displayed in the form of total mass residuals (Figure 7) and normalized CPU time (Table 3) with $\varepsilon_s = 10^{-6}$. Only normalized residuals over the densest grid using the SG, PG, and FMG methodologies are presented.

As depicted in Figure 7*a*, PISO requires the least number of iterations for the reasons stated earlier, but this is not associated with the lowest computational effort (Table 3). Moreover, the convergence behavior of SIMPLE (Figure 7*b*), SIMPLEC (Figure 7*c*), SIMPLEM (Figure 7*d*), and SIMPLEX (Figure 7*g*) is very similar, entailing nearly the same number of iterations with all solution methods. In comparison, SIMPLEST (Figure 7*e*) consumes a larger number of iterations. The worst performance is for PRIME (Figure 7*f*), especially with the SG method, where it is seen to require over 8,000 iterations to reduce the total mass residuals to the desired level. The above can also be inferred from the plots displayed in Figure 7*h*. Again, the PG and FMG methods greatly reduce the number of iterations with the number needed by the PG method, for most algorithms, being around four times the number required by the FMG method.

The normalized CPU times for all cases considered are presented in Table 3. The general trend is similar to the previous cases, with the SG requiring the highest CPU effort and the FMG being the most efficient. The use of the PG method on the dense grid decreases, on average, the computational time by about 38%, whereas about 363% reduction in computational time is realized with the FMG method. Consistently, the CPU time required by the SIMPLE, SIMPLEC, and SIMPLEX algorithms is very close, with no appreciable superiority of any one of these algorithms over the others. Moreover, the virtues of the FMG method increase as the grid size increases (e.g., for SIMPLEC the acceleration rate increases from 1.65 with two levels, to 2.2 with three levels, up to 5.57 with four levels). On the dense grids, which are of interest, the most time-consuming algorithm is PRIME, followed by SIMPLEST and PISO. The best performance, however, is for SIMPLE, SIMPLEC, and SIMPLEX, with that of SIMPLEM being close.

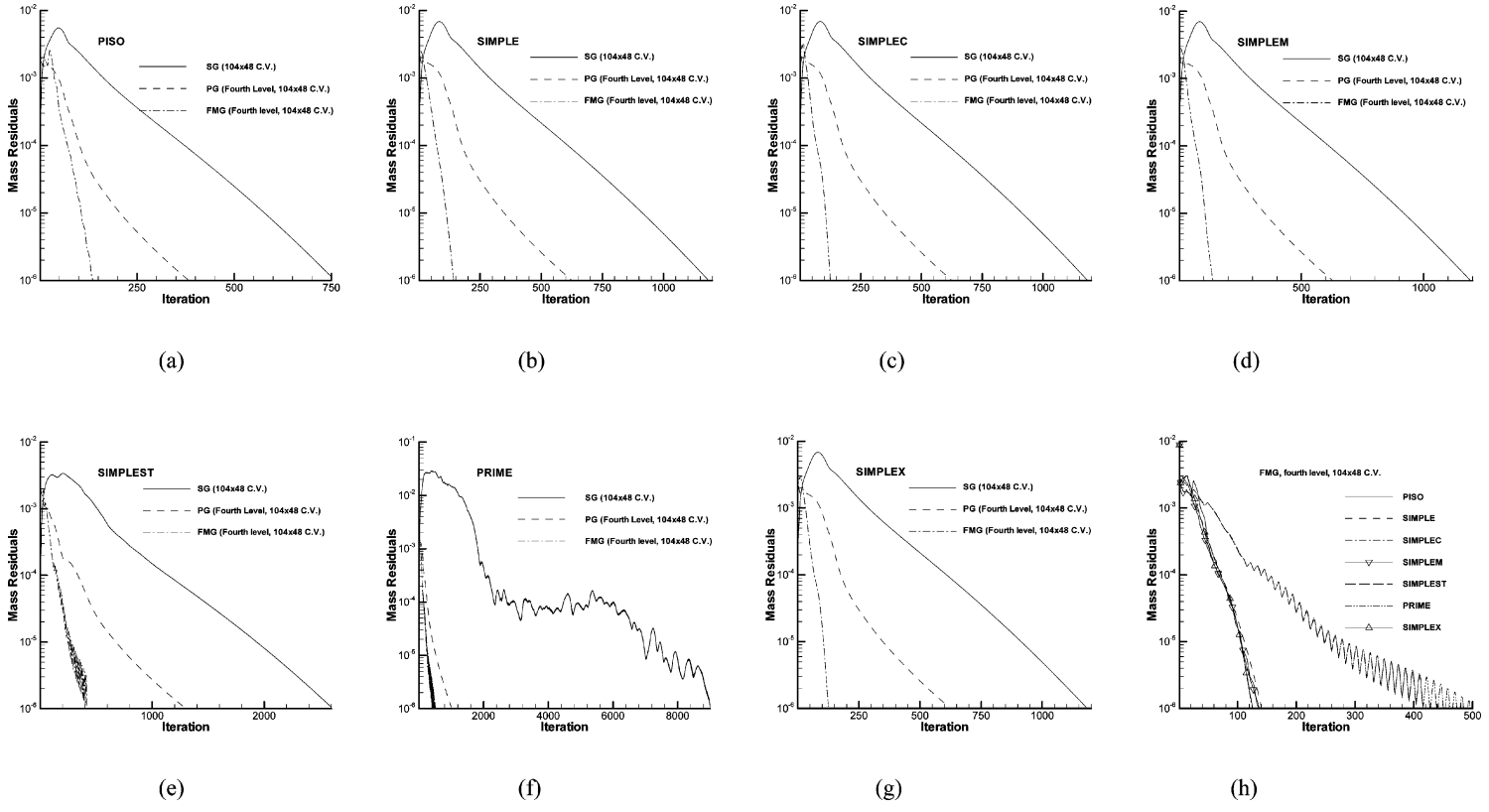


Figure 7. (*a*)–(*g*) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (*h*) convergence histories of the various algorithms on the finest mesh using the FMG method for dusty gas flow over a flat plate.

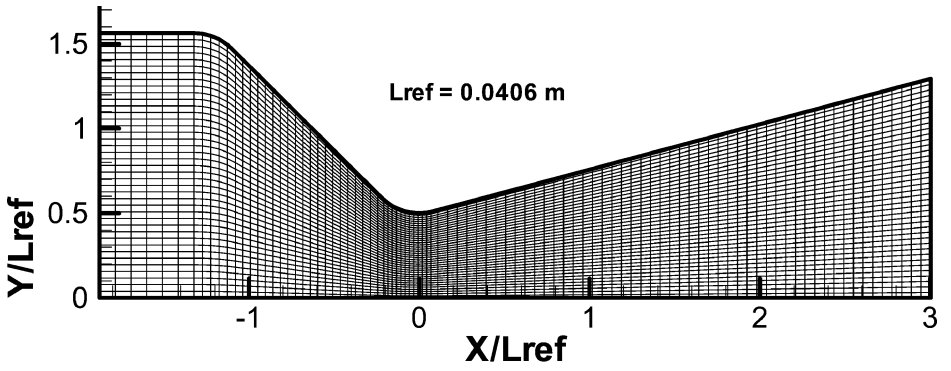
Table 3. Normalized CPU times for dusty flow over a flat plate

Grid	Method	Algorithms						
		SIMPLE	SIMPLEC	SIMPLEX	SIMPLEST	SIMPLEM	PISO	PRIME
13×6 C.V.	SG	1.00	0.96	0.99	1.06	1.43	0.74	1.12
26×12 C.V.	SG	4.06	4.38	4.55	5.31	5.44	3.84	5.75
	PG (2 levels)	3.77	3.87	3.73	5.59	4.49	3.02	5.47
	SG/PG	1.08	1.13	1.22	0.95	1.21	1.27	1.05
	FMG (2 levels)	2.80	2.66	2.69	9.42	3.77	3.31	7.98
	SG/FMG	1.45	1.65	1.69	0.56	1.44	1.16	0.72
52×24 C.V.	SG	30.61	30.69	32.76	43.41	39.45	38.08	62.78
	PG (3 levels)	27.22	27.41	29.74	37.92	33.53	35.25	54.02
	SG/PG	1.12	1.12	1.10	1.14	1.18	1.08	1.16
	FMG (3 levels)	13.52	13.97	14.34	39.07	17.15	23.41	39.17
	SG/FMG	2.26	2.20	2.28	1.11	2.30	1.63	1.60
104×48 C.V.	SG	373.17	375.29	386.01	715.44	423.80	495.72	1295.88
	PG (4 levels)	280.69	279.03	286.76	629.68	301.09	423.66	680.36
	SG/PG	1.33	1.34	1.35	1.14	1.41	1.17	1.90
	FMG (4 levels)	75.74	67.34	70.48	234.00	86.15	154.20	246.60
	SG/FMG	4.93	5.57	5.48	3.06	4.92	3.21	5.25

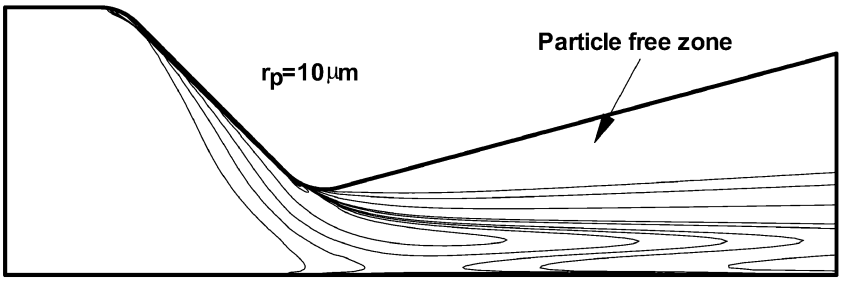
Problem 4: Inviscid Transsonic Dusty Flow in a Converging-Diverging Nozzle

The final test deals with dilute two-phase supersonic flow in an axisymmetric converging-diverging rocket nozzle. Several researchers have analyzed the problem, and data are available for comparison [60–65]. The flow is assumed to be inviscid and the single-phase results are used as an initial guess for solving the two-phase flow problem. The physical configuration (Figure 8*a*) and properties used are the ones described in [62]. The gas stagnation temperature and pressure at inlet to the nozzle are 555 K and $10.34 \times 10^5\,\mathrm{N/m^2}$, respectively. The specific heat for the gas and particles are $1.07 \times 10^3\,\mathrm{J/kg\,K}$ and $1.38 \times 10^3\,\mathrm{J/kg\,K}$, respectively, and the particle density is $4{,}004.62\,\mathrm{kg/m^3}$. With a zero inflow velocity angle, the fluid is accelerated from subsonic to supersonic speed in the nozzle. The inlet velocity and temperature of the particles are presumed to be the same as those of the gas phase. Results for a particle size of radius 10 μm with a mass fraction $\phi = 0.3$ are presented using a grid of size 188×80 C.V. Figure 8*b* shows the particle volume fraction contours, while Figure 8*c* displays the velocity distribution. For the particle size selected, a large particle-free zone appears due to the inability of these relatively heavy particles ($r_p = 10\,\mu\mathrm{m}$) to turn around the throat corner. These findings are in excellent agreement with published results reported by Chang et al. [62] and others using different methodologies. In addition, contours are similar to those reported by Chang et al. [62].

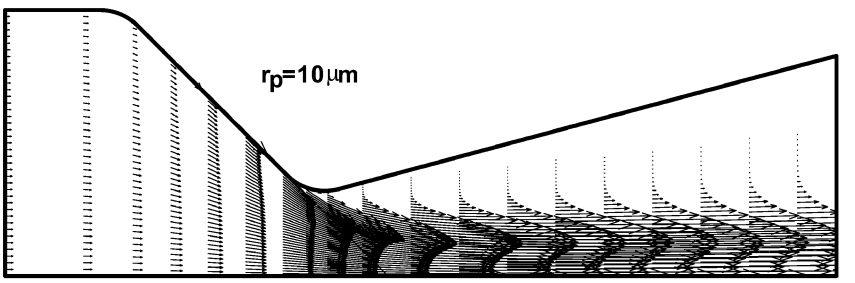
A quantitative comparison of current predictions with published experimental and numerical data is presented in Figure 9 through gas Mach number distributions along the wall (Figure 9*a*) and centerline (Figure 9*b*) of the nozzle for the one-phase and two-phase flow situations. As can be seen, the one-phase predictions fall on top of experimental data reported in [63–65]. Due to the unavailability of two-phase flow data, predictions are compared against the numerical results reported in [62].



(a)



(b)

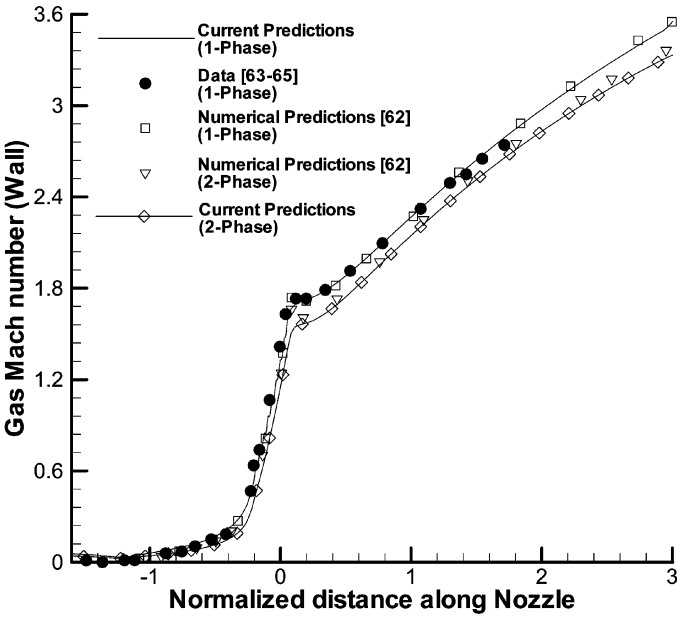


(c)

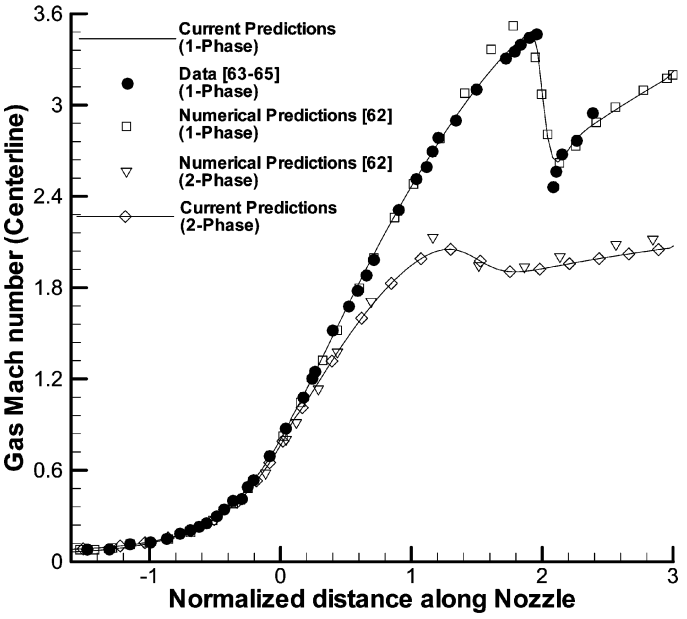
Figure 8. (*a*) Physical domain for the dusty gas flow in a converging-diverging nozzle: (*b*) Volume–fraction contours and (*c, d*) particle velocity vectors for dusty gas flow in a converging-diverging nozzle.

As displayed in Figures 9*a* and 9*b*, both solutions are in good agreement with each other, indicating once more the correctness of the calculation procedures.

As reported in [34], the use of the single-grid method requires a large number of iterations on the dense grid with heavy underrelaxation, and the performance of the



(a)



(b)

Figure 9. Comparison of one-phase and two-phase gas Mach number distributions along (*a*) wall and (*b*) centerline of the dusty flow in a converging-diverging nozzle problem.

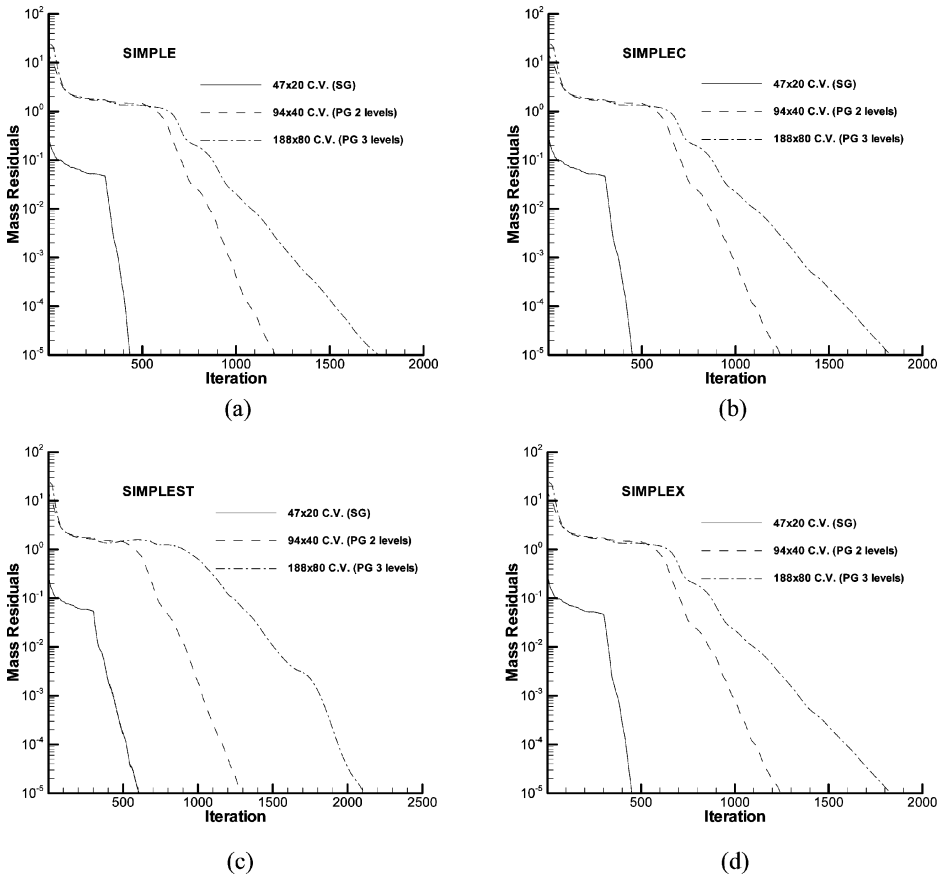


Figure 10. Convergence histories of (*a*) SIMPLE, (*b*) SIMPLEC, (*c*) SIMPLEST, and (*d*) SIMPLEX algorithms using the SG method for dusty gas flow in a converging-diverging nozzle.

FMG is marginally better than that of the PG method. Because of this, the relative performance of the SIMPLE, SIMPLEC, SIMPLEX, and SIMPLEST multifluid algorithms is compared using the PG method only over three different grids of sizes 47×20, 94×40, and 188×80 C.V. As before, results are displayed in the form of total mass residuals (Figure 10) and normalized CPU times (Table 4).

As shown in Figure 10, with the exception of SIMPLEST, which needs about 17% more iteration, all algorithms require almost the same number of iterations. As

Table 4. Normalized CPU times for dusty flow in a converging-diverging nozzle

		Algorithms			
Grid	Method	SIMPLE	SIMPLEC	SIMPLEX	SIMPLEST
47×20 C.V.	SG	1.00	1.05	1.13	1.47
94×40 C.V.	PG (2 levels)	12.57	12.97	14.06	13.62
188×80 C.V.	PG (3 levels)	86.64	90.61	98.06	98.53

expected, the number of iterations increases with increasing grid density. Moreover, the convergence histories of all algorithms are nearly identical. The normalized CPU times presented in Table 4 confirm these conclusions and reveal the close performance of SIMPLE and SIMPLEC. The normalized CPU times needed by SIMPLEX, which are close to SIMPLEST, are higher than those needed by SIMPLE and SIMPLEC due to the additional equations solved. Its performance could improve on denser grids.

CLOSING REMARKS

The implementation of seven MCBA algorithms for the simulation of multifluid flow at all speeds was accomplished. The algorithms were embedded within a nonlinear full multigrid strategy. A two-fluid k–ε model and several interphase models were also employed. Solving a variety of two-dimensional two-phase flow problems assessed the performance and accuracy of these algorithms. For each test problem, solutions were generated on a number of grid systems using the single-grid method (SG), the prolongation-grid method (PG), and the full nonlinear multigrid method (FMG). Results obtained demonstrated the capability of all algorithms to predict multifluid flow at all speeds and the ability of the FMG method to tackle the added nonlinearity of laminar and turbulent multifluid flows. The convergence history plots and CPU times presented indicated similar performance for SIMPLE, SIMPLEC, and SIMPLEX. The PISO, SIMPLEM, and SIMPLEST algorithms were generally more expensive than SIMPLE. In general, the PRIME algorithm was the most expensive to use. Moreover, the PG and FMG methods accelerated the convergence rate for all algorithms. The FMG method was found to be by far more efficient.

REFERENCES

1. 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.
2. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1981.
3. 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.
4. 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.
5. J. P. Van Doormaal and G. D. Raithby, An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows, ASME Paper 85-HT-9, Natl. Heat Transfer Conf., Denver, Co., August 4–7, 1985.
6. 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 Medecine, London, UK, 1980.
7. R. I. Issa, Solution of the Implicit Discretized Fluid Flow Equations by Operator Splitting, Mech. Eng. Rep. FS/82/15, Imperial College, London, UK, 1982.
8. 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, 1983, pp. 656–666.

9. C. Becker, J. H. Ferziger, M. Perıc, and G. Scheurer, Finite Volume Multigrid Solutions of the Two-Dimensional Incompressible Navier-Stokes Equations, *Notes on Numer. Fluid Mech.*, vol. 23, pp. 37–47, 1988.
10. K. M. Smith, W. K. Cope, and S. P.Vanka, Multigrid Procedure for Three Dimensional Flows on Non-orthogonal Collocated Grids, *Int. J. Numer. Meth. Fluids*, vol. 17, pp. 887–904, 1993.
11. F. S. Lien and M. A. Lechziner, Multigrid Acceleration for Recirculating Laminar and Turbulent Flow Computed on a Non-orthogonal, Collocated Finite Volume Scheme, *Comput. Meth. Appl. Mech. Eng.*, vol. 118, pp. 351–371, 1994.
12. S. Sivaloganathan and G. J. Shaw, An Efficient Non-linear Multigrid Procedure for the Incompressible Navier-Stokes Equations, *Int. J. Numer. Meth. Fluids*, vol. 8, pp. 417–440, 1988.
13. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, Heidelberg, 1996.
14. 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.
15. I. Demirdzic, Z. Lilek, and M. Peric, A Collocated Finite Volume Method for Predicting Flows at All Speeds, *Int. J. Numer. Meth. Fluids*, vol. 16, pp. 1029–1050, 1993.
16. C. H. Marchi, and C. R. Maliska, A Non-orthogonal Finite-Volume Method for the Solution of All Speed Flows Using Co-Located Variables, *Numer. Heat Transfer, B*, vol. 26, pp. 293–311, 1994.
17. F. S. Lien and M. A. Leschziner, A Pressure-Velocity Solution Strategy for Compressible Flow and Its Application to Shock/Boundary-Layer Interaction Using Second-Moment Turbulence Closure, *J. Fluids Eng.*, vol. 115, pp. 717–725, 1993.
18. M. A. 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.
19. K. H. Chen and R. H. Pletcher, Primitive Variable, Strongly Implicit Calculation Procedure for Viscous Flows at All Speeds, *AIAA J.*, vol. 29, no. 8, pp. 1241–1249, 1991.
20. M. Darbandi and G. E. Shneider, Momentum Variable Procedure for Solving Compressible and Incompressible Flows, *AIAA J.*, vol. 35, no. 12, pp. 1801–1805, 1997.
21. F. Moukalled and M. Darwish, A High-Resolution Pressure-Based Algorithm for Fluid Flow at All Speeds, *J. Comput. Phys.*, vol. 168, no., pp. 101–133, 2001.
22. F. Moukalled and M. Darwish, A Unified Formulation of the Segregated Class of Algorithms for Fluid Flow at All Speeds, *Numer. Heat Transfer B*, vol. 37, no. 1, pp. 103–139, 2000.
23. M. Darwish, D. Asmar, and F. Moukalled, A Comparative Assessment within a Multigrid Environment of Segregated Pressure-Based Algorithms for Fluid Flow at All Speeds, *Numer. Heat Transfer B*, 1994, vol. 45, no. 1, pp. 49–74, 2004.
24. D. B. Spalding, The Calculation of Free-Convection Phenomena in Gas-Liquid Mixtures, Rep. HTS/76/11 Mech. Eng., Imperial College, London, UK, 1976.
25. D. B. Spalding, Numerical Computation of Multi-phase Fluid Flow and Heat Transfer, in *Recent Advances in Numer. Meth. in Fluids*, vol. 1, C. Taylor and K. Morgan (eds.) pp. 139–167, 1980.
26. D. B. Spalding, A General Purpose Computer Program for Multi-dimensional, One and Two Phase Flow, Mechanical Engineering Department Rep. HTS/81/1, Imperial College, London, UK, 1981.
27. W. W. Rivard and M. D. Torrey, KFIX: A Program for Transient Two Dimensional Two Fluid Flow, Rep. Los Alamos Nuclear Regulatory Commission-6623, 1978.

28. A. A. Amsden and F. H. Harlow, KACHINA: An Eulerian Computer Program for Multi-field Flows, Rep. LA-NUREG-5680, 1975.
29. A. A. Amsden and F. H. Harlow, KTIFA Two-Fluid Computer Program for Down Comer Flow Dynamics, Rep. LA-NUREG-6994, 1977.
30. M. Darwish, F. Moukalled, and B. Sekar, A Unified Formulation of the Segregated Class of Algorithms for Multi-fluid Flow at All Speeds, *Numer. Heat Transfer B*, vol. 40, no. 2, pp. 99–137, 2001.
31. F. Moukalled and M. A. Darwish, Comparative Assessment of the Performance of Mass Conservation Based Algorithms for Incompressible Multi-phase Flows, *Numer. Heat Transfer B*, vol. 42, pp. 259–283, 2002.
32. F. Moukalled and M. Darwish, The Performance of Geometric Conservation Based Algorithms for Incompressible Multi-fluid Flow, *Numer. Heat Transfer B*, 2004, vol. 45, no. 4, pp. 343–368.
33. F. Moukalled, M. Darwish, and B. Sekar, A High Resolution Pressure-Based Algorithm for Multi-phase Flow at all Speeds, *J. Comput. Phys.*, vol. 190, pp. 550–571, 2003.
34. M. Darwish, F. Moukalled, and B. Sekar, A Robust Multi-grid Algorithm for Multifluid Flow at All Speeds, *Int. J. Numer. Meth. Fluids*, vol. 41, pp. 1221–1251, 2003.
35. F. Moukalled and M. Darwish, Pressure Based Algorithms for Multi-fluid Flow at All Speeds—Part II: Geometric Conservation Formulation, *Numer. Heat Transfer B*, vol. 45, pp. 523–540, 2004.
36. B. S. Baldwin and H. Lomax, Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows, AIAA Paper 78–257, 1978.
37. F. Sotiropoulos and V. C. Patel, Application of Reynolds-Stress Transport Models to Stern and Wake Flow, *J. Ship Res.*, vol. 39, p. 263, 1995.
38. D. Cokljat, V. A. Ivanov, F. J. Srasola, and S. A. Vasquez, Multiphase K-Epsilon Models for Unstructured Meshes, ASME 2000 Fluids Engineering Division Summer Meeting, Boston, MA, June 11–15, 2000.
39. F. Pourahmadi and J. A. C. Humphrey, Modeling Solid-Fluid Turbulent Flows with Application to Predicting Erosive Wear, *Int. J. Phys. Chem. Hydrodynamic*, vol. 4, pp. 191–219, 1983.
40. S. E. Elghobashi and T. W. Abou-Arab, A Two-Equation Turbulence Model for Two-Phase Flows, *Phys. Fluids*, vol. 26, no. 4, pp. 931–938, 1983.
41. C. P. Chen and P. E. Wood, Turbulence Closure Modeling of the Dilute Gas-Particle Axisymmetric Jet, *AICHE J.*, vol. 32, no. 1, pp. 163–166, 1986.
42. A. A. Mostafa and H. C. Mongia, On the Interaction of Particles and Turbulent Fluid Flow, *Int. J. Heat Mass Transfer*, vol. 31, pp. 2063–2075, 1988.
43. M. Lopez de Bertodano, S. J. Lee, R. T. Lahey, Jr., and D. A. Drew, The Prediction of Two-Phase Turbulence and Phase Distribution Phenomena Using a Reynolds Stress Model, *ASME J. Fluids Eng.*, vol. 112, pp. 107–113, 1990.
44. M. Lopez de Bertodano, R. T. Lahey, Jr., and O. C. Jones, Development of a k–ε Model for Bubbly Two-Phase Flow, *ASME J. Fluids Eng.*, vol. 116, pp. 128–134, 1994.
45. M. Lopez de Bertodano, R. T. Lahey, Jr., and O. C. Jones, Phase Distribution in Bubbly Two-Phase Flow in Vertical Ducts, *Int. J. Multiphase Flow*, vol. 20, no. 5, pp. 805–818, 1994.
46. 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.
47. P. H. Gaskell and A. K. C. Lau, Curvature Compensated Convective Transport: SMART, a New Boundedness Preserving Transport Algorithm, *Int. J. Numer. Meth. Fluids*, vol. 8, pp. 617–641, 1988.

48. M. S. Darwish and F. Moukalled, Normalized Variable and Space Formulation Methodology for High-Resolution Schemes, *Numer. Heat Transfer B*, vol. 26, pp. 79–96, 1994.
49. C. Cornelius, W. Volgmann, and H. Stoff, Calculation of Three-Dimensional Turbulent Flow with a Finite Volume Multigrid Method, *Int. J. Numer. Meth. Fluids*, vol. 31, pp. 703–720, 1999.
50. A. Serizawa, I. Kataoka, and I. Michiyoshi, Phase Distribution in Bubbly Flow, Data Set No. 24, *Proc. Second Int. Workshop on Two-Phase Flow Fundamentals*, Rensselaer Polytechnic Institute, Troy, NY, 1986.
51. V. E. Nakoryakov, O. N. Kashinsky, V. V. Randin, and L. S. Timkin, Gas-Liquid Bubbly Flow in Vertical Pipes, *J. of Fluids Eng.*, vol. 118, pp. 377–382, 1996.
52. R. T. Lahey, M. Lopez de Bertodano, and O. C. Jones, Phase Distribution in Complex Geometry Ducts, *Nuclear Eng. Design*, vol. 141, p. 177, 1993.
53. N. Boisson and M. R. Malin, Numerical Prediction of Two-Phase Flow in Bubble Columns, *Int. J. Numer. Meth. Fluids*, vol. 23, pp. 1289–1310, 1996.
54. Y. Tsuji, Y. Morikawa, and H. Shiomi, LDV Measurements of an Air-Solid Two-Phase Flow in a Vertival Pipe, *J. Fluid Mech.*, vol. 139, pp. 417–434, 1984.
55. A. Adeniji-Fashola and C. P. Chen, Modeling of Confined Turbulent Fluid-Particle Flows Using Eulerian and Lagrangian Schemes, *Int. J. Heat Mass Transfer*, vol. 33, pp. 691–701, 1990.
56. S. Naik and I. G. Bryden, Prediction of Turbulent Gas-Solids Flow in Curved Ducts Using the Eulerian-Lagrangian Method, *Int. J. Numer. Meth. Fluids*, vol. 31, pp. 579–600, 1999.
57. A. N. Osiptsov, Structure of the Laminar Boundary Layer of a Disperse Medium on a Flat Plate, *Fluid Dynam.*, vol. 15, pp. 512–517, 1980.
58. B. Y. Wang and I. I. Glass, Compressible Laminar Boundary Layer Flows of a Dusty Gas over a Semi-infinite Flat Plate, *J. Fluid Mech.*, vol. 186, pp. 223–241, 1988.
59. N. Thevand, E. Daniel, and J. C. Loraud, On High-Resolution Schemes for Solving Unsteady Compressible Two-Phase Dilute Viscous Flows, *Int. J. Numer. Meth. Fluids*, vol. 31, pp. 681–702, 1999.
60. I. S. Chang, One and Two-Phase Nozzle Flows, *AIAA J.*, vol. 18, pp. 1455–1461, 1980.
61. R. Ishii, Y. Umeda, and K. Kawasaki, Nozzle Flows of Gas-Particle Mixtures, *Phys. Fluids*, vol. 30, no. 3, pp. 752–760, 1987.
62. H. T. Chang, L. W. Hourng, and L. E. Chien, Application of Flux-Vector-Splitting Scheme to a Dilute Gas-Particle JPL Nozzle Flow, *Int. J. Numer. Meth. Fluids*, vol. 22, pp. 921–935, 1996.
63. L. H. Back and R. F. Cuffel, Detection of Oblique Shocks in a Conical Nozzle with a Circular-Arc Throat, *AIAA J.*, vol. 4, pp. 2219–2221, 1966.
64. L. H. Back, P. F. Massier, and R. F. Cuffel, Flow Phenomena and Convective Heat Transfer in a Conical Supersonic Nozzle, *J. Spacecraft*, vol. 4, pp. 1040–1047, 1967.
65. R. F. Cuffel, L. H. Back, and P. F. Massier, Transonic Flowfield in a Supersonic Nozzle with Small Throat Radius of Curvature, *AIAA J.*, vol. 7, pp. 1364–1366, 1969.