

The Performance of Geometric Conservation Based Algorithms for Incompressible Multi-Fluid Flow

F. Moukalled¹ and M. Darwish²
American University of Beirut,
Faculty of Engineering & Architecture,
Mechanical Engineering Department,
P.O.Box 11-0236
Riad El Solh, Beirut 1107 2020
Lebanon

Abstract

This paper deals with the implementation and testing of seven segregated pressure-based algorithms for the prediction of incompressible multi-fluid flow. These algorithms belong to the Geometric Conservation Based Algorithm (GCBA) group in which the pressure correction equation is derived from the constraint on volume fractions (i.e. sum of volume fractions equals 1). The pressure correction schemes in these algorithms are based on SIMPLE, SIMPLEC, SIMPLEX, SIMPLEM, SIMPLEST, PISO, and PRIME. The performance and accuracy of these algorithms are assessed by solving eight one-dimensional two-phase flow problems and comparing results with published data. The effects of grid size on convergence characteristics are analyzed by solving each problem over different grid sizes. Results clearly demonstrate the capability of all GCBA algorithms to predict a wide range of multi-fluid flow situations. Based on the convergence history plots and CPU-times obtained for the problems solved, the GCBA can be divided into two groups with the one composed of SIMPLEST and PRIME being generally less efficient than the second group to which the remaining algorithms belong.

¹ Author to whom all correspondence should be addressed; email: memouk@aub.edu.lb; ²email: Darwish@aub.edu.lb

Nomenclature

$A_p^{(k)}, \dots$ 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.

$\mathbf{D}_p^{(k)}$ the Matrix $\mathbf{D}$ operator.

$H_p[\phi^{(k)}]$ the H operator.

$\mathbf{H}_p[\mathbf{u}^{(k)}]$ the vector form of the H operator.

$\mathbf{I}^{(k)}$ inter-phase momentum transfer.

$M^{(k)}$ mass source per unit volume.

P pressure.

$Q^{(k)}$ general source term of fluid/phase k.

$r^{(k)}$ volume fraction of fluid/phase k.

$R_p^{(k)}$ coefficient equals $1 / A_p^{(k)}$

$\mathbf{S}_f$ surface vector.

t time.

$U_f^{(k)}$ interface flux velocity $(\mathbf{v}_f^{(k)} \cdot \mathbf{S}_f)$ of fluid/phase k.

$\mathbf{u}^{(k)}$ velocity vector of fluid/phase k.

$u^{(k)}, v^{(k)}, \dots$ velocity components of fluid/phase k.

x, y Cartesian coordinates.

Greek Symbols

$\rho^{(k)}$ density of fluid/phase k.

$\Gamma^{(k)}$ diffusion coefficient of fluid/phase k.

$\phi^{(k)}$ general scalar variable associated with fluid/phase k.

$\Delta_P[\phi^{(k)}]$ the Δ operator.

Ω cell volume.

δt time step.

Subscripts

e, w, . refers to the east, west, ... face of a control volume.

E, W, .. refers to the East, West, ... neighbors of the main grid point.

f refers to control volume face f.

P refers to the P grid point.

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.

$(k)^\bullet$ 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.

Introduction

Flows involving multiple distinct phases, known as multi-fluid flows, arise in many industrial applications [1,2] such as mixing tanks [3-5], bubbly flows [6-8], fluidized beds [9,10], hydro-transport [11], separators [12-14], and spray drying [15,16] to cite a few. Numerical simulation is arguably the principal path to a fundamental understanding of these flows, which may comprise any combination of gas, liquid, and solid phases. A concerted research effort has been undertaken over the last three decades to develop appropriate numerical techniques [17-26] capable of simulating such flows. Workers have rigorously enhanced the depiction of the governing equations [27-31] and exploited the extensive advances that have been achieved in single-fluid numerical techniques [32-39] to develop reliable multi-fluid numerical strategies [40,41].

The description of multi-fluid flow begins with the general principles governing the behavior of matter, namely conservation of mass, momentum, and energy. These principles can be expressed mathematically at every point in space by local and instantaneous field equations. Because of their numerical intractability at microscopic level, these equations are averaged in space and time. Several averaging strategies have been developed (Ishii [28] and Drew [42], Soo [43], Gidaspow [44]), which led basically to two different methods for describing the physical processes. In the first approach the n phases are averaged together to obtain averaged variables for an n -phase mixture, while in the second technique each phase is averaged separately giving rise to averaged variables for each of the n phases. The latter procedure, adopted here, yields the multi-fluid model, which is a bit more general and useful. Even the most simplified models for two-dimensional incompressible multi-fluid flows are complex, as they involve the solution of $3n$ (n the number of phases/fluids) coupled partial differential equations (averaged mass and momentum equations for each fluid) and numerous constitutive equations. Work on numerical techniques for the simulation of multi-fluid flows

within a segregated pressure-based approach was initiated independently, following two distinct tracks, by the CFD group at the Los Alamos Scientific Laboratory [20-21] and by the Spalding group at Imperial College [17-19,45]. In a recent article, Darwish et al. [40] have shown that these two approaches and the extensive developments that followed can be classified under what they denoted by the mass conservation and geometric conservation formulation, respectively. This designation was based upon the equation used in obtaining the pressure correction equation, which can be derived either by using the geometric conservation equation or the overall mass conservation equation. Depending on which equation is employed, the segregated pressure-based multi-fluid flow algorithms were classified respectively as either the Geometric Conservation Based family of Algorithms (GCBA) or the Mass Conservation Based family of Algorithms (MCBA). Moukalled and Darwish [41] implemented and tested the MCBA family and proved its capability to predict multi-fluid flow. On the other hand, the GCBA family has not yet been implemented nor tested.

The objective of the present work is to implement and test seven multi-phase algorithms from the GCBA family (with their pressure correction schemes based on SIMPLE [46], SIMPLEST [47], SIMPLEC [48], SIMPLEM [49], PISO [50], PRIME [51], and SIMPLEX [52]) and to assess their relative performance by solving a total of eight incompressible one-dimensional two-phase flow problems encompassing dilute and dense gas-solid flows in addition to bubbly flows on several grid sizes.

In what follows, the governing equations for incompressible multi-phase flows are first introduced, followed by a brief description of the discretization procedure. Then the capability of the GCBA algorithms to predict multi-fluid flow phenomena is demonstrated and their performance characteristics (in terms of convergence history and speed) assessed.

The Governing Equations

The equations governing incompressible multi-phase flows are the conservation laws of mass and momentum for each individual fluid, given by:

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

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

where the superscript (k) refers to the kth phase, $r^{(k)}$ the volume fraction ($\Omega^{(k)}/\Omega$), $\rho^{(k)}$ the fluidic density, $\mathbf{u}^{(k)}$ the velocity vector, P the pressure shared by all fluids/phases, $\mathbf{B}^{(k)}$ the body force per unit volume, $\mu^{(k)}$ the fluid viscosity, and $\mathbf{I}_M^{(k)}$ represents the interfacial forces per unit volume due to drag, virtual mass effects, lift, etc...

An adequate manipulation of equations (1) and (2) allows their representation in a unified equation of a general fluidic scalar variable $\phi^{(k)}$ as follows:

$$\frac{\partial(r^{(k)}\rho^{(k)}\phi^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\phi^{(k)}) = \nabla \cdot (r^{(k)}\Gamma^{(k)}\nabla\phi^{(k)}) + r^{(k)}Q^{(k)} \quad (3)$$

The above set of differential equations has to be solved in conjunction with constraints on certain variables represented by algebraic relations. For incompressible laminar multi-phase flow, these auxiliary relations include the geometric conservation equation $\left(\sum_k r^{(k)} = 1\right)$ and the interfacial mass and momentum transfers. In this work, only interfacial momentum transfer is of interest and its closure will be detailed later. Moreover, in order to present a closed mathematical model, initial and boundary conditions should supplement the above equations.

Discretization Procedure

The first step in the discretization process consists of integrating the general equation (Eq. (3)) over a differential control volume (Fig. 1(a)) to yield:

$$\begin{aligned}
\iint_{\Omega} \frac{\partial(r^{(k)} \rho^{(k)} \phi^{(k)})}{\partial t} d\Omega + \iint_{\Omega} \nabla \cdot (r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \phi^{(k)}) d\Omega \\
= \iint_{\Omega} \nabla \cdot (r^{(k)} \Gamma^{(k)} \nabla \phi^{(k)}) d\Omega + \iint_{\Omega} r^{(k)} Q^{(k)} d\Omega
\end{aligned} \tag{4}$$

where Ω is the volume of the control cell. Through the use of the divergence theorem, the convection and diffusion volume integrals are transformed into surface integrals as:

$$\begin{aligned}
\iint_{\Omega} \frac{\partial(r^{(k)} \rho^{(k)} \phi^{(k)})}{\partial t} d\Omega + \iint_{\partial\Omega} (r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \phi^{(k)}) \cdot d\mathbf{S} \\
= \iint_{\partial\Omega} (r^{(k)} \Gamma^{(k)} \nabla \phi^{(k)}) \cdot d\mathbf{S} + \iint_{\Omega} r^{(k)} Q^{(k)} d\Omega
\end{aligned} \tag{5}$$

By a trapezoidal-type approximation, the surface integral about the cell faces is set equal to a summation of fluxes at the cell faces centers, while the volume integral is evaluated at the cell center. Thus, Eq. (5) becomes:

$$\frac{\partial(r^{(k)} \rho^{(k)} \phi^{(k)})}{\partial t} \Omega + \sum_{nb} (r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S} \phi^{(k)} - r^{(k)} \Gamma^{(k)} \nabla \phi^{(k)} \cdot \mathbf{S}) = r^{(k)} Q^{(k)} \Omega \tag{6}$$

These fluxes are then related to the values at the cell centers and their neighboring nodes by using a suitable interpolation profile in a local coordinate direction. The profile of the diffusion term is second order accurate and follows the derivations presented in [53]. For the convective terms, the High Resolution SMART [38] scheme is employed and applied within the context of the NVSF methodology [34]. Substitution of the interpolation profiles into Eq. (6) gives the final form of the discretized equation as

$$A_p^{(k)} \phi_p^{(k)} = \sum_{NB} A_{NB}^{(k)} \phi_{NB}^{(k)} + B_p^{(k)} \tag{7}$$

where the coefficients $A_p^{(k)}$ and $A_{NB}^{(k)}$ 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

$$\phi^{(k)} = H_p[\phi^{(k)}] = \frac{\sum_{NB} A_{NB}^{(k)} \phi_{NB}^{(k)} + B_p^{(k)}}{A_p^{(k)}} \tag{8}$$

The discretization procedure for the momentum equation yields an algebraic equation of the form:

$$\mathbf{u}_p^{(k)} = \mathbf{H}_p[\mathbf{u}^{(k)}] - r^{(k)} \mathbf{D}_p^{(k)} \nabla_p(P) \quad (9)$$

Moreover, the phasic mass-conservation equation (Eq. (1)) can be viewed as a phasic volume fraction equation, which can either be written as:

$$r_p^{(k)} = H_p[r^{(k)}] \quad (10)$$

or as a fluidic continuity equation to be used in deriving the pressure correction equation:

$$\frac{(r_p^{(k)} \rho_p^{(k)}) - (r_p^{(k)} \rho_p^{(k)})^{Old}}{\delta t} \Omega + \Delta_p[r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S}] = r^{(k)} M^{(k)} \quad (11)$$

where the Δ operator represents the following operation:

$$\Delta_p[\Theta] = \sum_f \Theta_f \quad (12)$$

Geometric Conservation Based Algorithms (GCBA)

The numbers of equations describing an incompressible n-fluid flow situation are: n momentum equations, n volume fraction (or mass conservation) equations, and a geometric conservation equation. Moreover, the variables involved are the n velocity vectors, the n volume fractions, and the pressure field. It is clear that the n-velocity fields are associated with the n-momentum equations, i.e. the momentum equations can be used directly to calculate the velocity fields. The volume fractions could arguably be calculated from the volume fraction equations, which mean that the remaining equation, i.e. the geometric conservation equation (the volume fractions sum to 1), has to be used in deriving the pressure equation, or equivalently the pressure correction equation. This results in what was denoted [40] by the Geometric Conservation Based Algorithm (GCBA).

The GCBA uses the momentum equations for a first estimate of velocities. However, the volume fractions are calculated without enforcing the geometric conservation equation. Hence, the mass conservation equations of all fluids are used to calculate the volume fractions. As such, the pressure correction equation should be based on the geometric conservation equation and used to restore the imbalance of volume fractions. The errors in

the calculated volume fractions are expressed in terms of pressure correction(P'), which is also used to adjust the velocity and volume fraction fields. The sequence of events in the Geometric Conservation Based Algorithm (GCBA) is as follows:

- Solve the individual mass conservation equations for volume fractions.
- Solve the momentum equations for velocities.
- Solve the pressure correction equation.
- Correct velocity, volume fraction, and pressure fields.
- Return to the first step and repeat until convergence.

The Pressure Correction Equation

In order for the volume fraction fields to satisfy the compatibility equation and the velocity and pressure fields the continuity equations, a guess-and-correct scheme is adopted. Correction is obtained by solving a pressure correction equation, which is derived from the geometric conservation equation by noticing that initially the volume fraction fields denoted by $r^{(k)*}$, do not satisfy the compatibility equation and a discrepancy exists, i.e.

$$RESG_p = 1 - \sum_k r_p^{(k)*} \quad (13)$$

A change to $r^{(k)*}$ is sought that would restore the balance. The corrected r value, denoted by $r^{(k)}$ ($r^{(k)} = r^{(k)*} + r^{(k)'}$), is such that

$$\sum_k (r^{(k)'}) = \sum_k (r^{(k)}) - \sum_k (r^{(k)*}) = 1 - \sum_k (r^{(k)*}) = RESG_p \quad (14)$$

Correction to the volume fraction, $r^{(k)'}$, will be associated with a correction to the velocity and pressure fields, $\mathbf{u}^{(k)'}$ and P' respectively. Thus, the corrected fields are given as:

$$r^{(k)} = r^{(k)*} + r^{(k)'}, P = P^o + P', \mathbf{u}^{(k)} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'} \quad (15)$$

The discretized form of the corrected continuity equation of phase (k) can be written as

$$\frac{(r_p^{(k)*} + r_p^{(k)'})\rho_p^{(k)} - (r_p^{(k)}\rho_p^{(k)})^{Old}}{\delta t}\Omega_p + \Delta_p \left((r_p^{(k)*} + r_p^{(k)'})\rho_p^{(k)}(\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}) \cdot \mathbf{S} \right) = \mathcal{M}_p^{(k)}(r_p^{(k)*} + r_p^{(k)'})\Omega_p \quad (16)$$

Neglecting second order terms, its expanded form reduces to:

$$\frac{(r_p^{(k)'}\rho_p^{(k)})}{\delta t}\Omega_p + \Delta_p \left[(r_p^{(k)*}\rho_p^{(k)}\mathbf{u}^{(k)'} \cdot \mathbf{S} + \rho_p^{(k)}U^{(k)*}r_p^{(k)'}) \right] - \mathcal{M}_p^{(k)}r_p^{(k)'}\Omega_p = -\frac{(r_p^{(k)*}\rho_p^{(k)}) - (r_p^{(k)}\rho_p^{(k)})^{Old}}{\delta t}\Omega_p - \Delta_p \left[(r_p^{(k)*}\rho_p^{(k)}U^{(k)*}) \right] + \mathcal{M}_p^{(k)}r_p^{(k)*}\Omega_p \quad (17)$$

Writing $\mathbf{u}_p^{(k)'}$ as a function of P' , the correction momentum equations become

$$\mathbf{u}_p^{(k)'} = \mathbf{H}_p[\mathbf{u}^{(k)'}] - r_p^{(k)*}\mathbf{D}_p^{(k)}\nabla P' - r_p^{(k)'}\mathbf{D}_p^{(k)}\nabla P^o \quad (18)$$

Substituting Eq. (18) into Eq. (17), rearranging, and discretizing one gets

$$r_p^{(k)'} - H_p[r_p^{(k)'}] = -R_p^{(k)} \left(\Delta_p \left[r_p^{(k)*}\rho_p^{(k)} \begin{pmatrix} \mathbf{H}[\mathbf{u}^{(k)'}] - r_p^{(k)*}\mathbf{D}_p^{(k)}\nabla P' \\ -r_p^{(k)'}\mathbf{D}_p^{(k)}\nabla P' \end{pmatrix} \cdot \mathbf{S} \right] + \frac{(r_p^{(k)*}\rho_p^{(k)}) - (r_p^{(k)}\rho_p^{(k)})^{Old}}{\delta t}\Omega_p + \Delta_p \left[(r_p^{(k)*}\rho_p^{(k)}U^{(k)*}) \right] - \mathcal{M}_p^{(k)}r_p^{(k)*}\Omega_p \right) \quad (19)$$

where $R_p^{(k)} = 1/A_p^{(k)}$.

Neglecting volume fraction correction to neighboring cells, equation (19) reduces to:

$$r_p^{(k)'} = -R_p^{(k)} \left(\Delta_p \left[r_p^{(k)*}\rho_p^{(k)} \begin{pmatrix} \mathbf{H}[\mathbf{u}^{(k)'}] - r_p^{(k)*}\mathbf{D}_p^{(k)}\nabla P' \\ -r_p^{(k)'}\mathbf{D}_p^{(k)}\nabla P' \end{pmatrix} \cdot \mathbf{S} \right] + \frac{(r_p^{(k)*}\rho_p^{(k)}) - (r_p^{(k)}\rho_p^{(k)})^{Old}}{\delta t}\Omega_p + \Delta_p \left[(r_p^{(k)*}\rho_p^{(k)}U^{(k)*}) \right] - \mathcal{M}_p^{(k)}r_p^{(k)*}\Omega_p \right) \quad (20)$$

Substituting this equation into the geometric conservation equation, the pressure correction equation is obtained as

$$\sum_k \left\{ -R_p^{(k)} \left(\Delta_p \left[r_p^{(k)*}\rho_p^{(k)} \begin{pmatrix} \mathbf{H}[\mathbf{u}^{(k)'}] - r_p^{(k)*}\mathbf{D}_p^{(k)}\nabla P' - r_p^{(k)'}\mathbf{D}_p^{(k)}\nabla P' \end{pmatrix} \cdot \mathbf{S} \right] + \frac{(r_p^{(k)*}\rho_p^{(k)}) - (r_p^{(k)}\rho_p^{(k)})^{Old}}{\delta t}\Omega_p + \Delta_p \left[(r_p^{(k)*}\rho_p^{(k)}U^{(k)*}) \right] \right) \right\} = RESG_p \quad (21)$$

If the $\mathbf{H}[\mathbf{u}^{(k)'}]$ term in the above equation is retained, there would 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. These algorithms were detailed in [40] to which interested readers are referred. Following the calculation of the pressure correction field, $\mathbf{u}_p^{(k)'} and $r_p^{(k)'}$ are obtained using the following equations$

$$\begin{aligned} \mathbf{u}_p^{(k)'} &= -r^{(k)} \mathbf{D}_p^{(k)} \nabla_p (P') \\ r_p^{(k)'} &= -R_p^{(k)} \left(\Delta_p \left[\left(r^{(k)} \rho^{(k)} \mathbf{u}^{(k)'} \right) \cdot \mathbf{S} \right] \right) \end{aligned} \quad (22)$$

Results and Discussion

The performance of the various multi-fluid Geometric Conservation Based Algorithms is assessed in this section by solving eight one-dimensional two-phase flow problems. These problems can be broadly classified as: (i) horizontal particle transport, and (ii) vertical particle transport. Results are presented in terms of the CPU-time needed for the residuals to reach a set level and of the convergence history. Moreover, in order to assess the performance of the various algorithms with increasing grid density solutions are generated using four grid systems of sizes 20, 40, 80, and 160 control volumes. Results are compared against available numerical/theoretical values. The residual of a variable $\phi^{(k)}$ at the end of an outer iteration is defined as:

$$RES_{\phi}^{(k)} = \sum_{c.v.} \left| A_p \phi_p^{(k)} - \sum_{NB} A_{NB} \phi_{NB}^{(k)} - B_p^{(k)} \right| \quad (23)$$

For global mass conservation, the imbalance in mass is defined as:

$$RES_C = \sum_k \sum_{c.v.} \left| \frac{\left(r_p^{(k)} \rho_p^{(k)} \right) - \left(r_p^{(k)} \rho_p^{(k)} \right)^{Old}}{\delta t} \Omega - \Delta_p \left[r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S} \right] - r^{(k)} M^{(k)} \right| \quad (24)$$

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 , which is set to 10^{-8} . 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, since in iterative techniques different initial guesses might require different computational efforts.

Many runs were performed so as to set the control parameters of each algorithm near optimum values. The CPU times are reported in the form of charts and normalized by the time needed by the residuals of GCBA-SIMPLE to reach the set level on the coarsest grid.

Horizontal particle/bubble transport

The physical situation is illustrated in Fig. 1(b) with the gravitational acceleration set to zero. Air-particle and bubbly flows are modeled by correctly setting the densities of the continuous and disperse phases. Diffusion within both phases is neglected and drag, which is the sole driving force for the particle-air/bubble-water motion, is calculated as:

$$I_M^{(c)} = -I_M^{(d)} = \frac{3}{8} \frac{C_D}{r_p} r^{(d)} \rho^{(c)} V_{slip} (\mathbf{u}^{(d)} - \mathbf{u}^{(c)}) \quad (25)$$

$$V_{slip} = \|\mathbf{u}^{(d)} - \mathbf{u}^{(c)}\| \quad (26)$$

with the drag coefficient (C_D) assigned the value of 0.44. Since diffusion is neglected, the GCBA-SIMPLEST and GCBA-PRIME becomes identical and reference will be made to GCBA-SIMPLEST only. The task is to calculate the particle/bubble-velocity distribution as a function of position. If the flow field is extended far enough ($L=2m$), the particle/bubble and fluid phases are expected to approach an equilibrium velocity given by:

$$U_{equilibrium} = r_{inlet}^{(c)} \mathbf{u}_{inlet}^{(c)} + r_{inlet}^{(d)} \mathbf{u}_{inlet}^{(d)} \quad (27)$$

Problem 1: Dilute gas-solid flow

The first case considered is that of dilute gas-solid flow. The physical properties and boundary conditions of the two phases are: $\rho^{(d)} / \rho^{(c)} = 2000$, $r_p = 1 \text{ mm}$, $V_{inlet}^{(c)} = 5 \text{ m/s}$,

$V_{inlet}^{(d)} = 1 \text{ m/s}$, and $r_{inlet}^{(d)} = 10^{-5}$. Due to the dilute concentration of the particles, the free stream velocity is more or less unaffected by their presence, the equilibrium velocity is nearly equal to the inlet free stream velocity, and the variation of the particle velocity $u^{(d)}$ as a function of the position x and the properties of the two phases is given by [54]:

$$\text{Ln}\left[V_{inlet}^{(c)} - u^{(d)}\right] + \frac{V_{inlet}^{(c)}}{V_{inlet}^{(c)} - u^{(d)}} = \frac{3}{8} \frac{\rho^{(c)}}{\rho^{(d)}} \frac{C_D}{r_p} x + \text{Ln}\left[V_{inlet}^{(c)} - V_{inlet}^{(d)}\right] + \frac{V_{inlet}^{(c)}}{V_{inlet}^{(c)} - V_{inlet}^{(d)}} \quad (28)$$

The accuracy of the numerical procedure is demonstrated in Fig. 2(a) by comparing the predicted particle velocity distribution against the analytical solution given by Eq. (28). As shown, the two solutions are indistinguishable. The convergence histories of the various GCBA over the four grid networks used are displayed in Figs. 2(b)-2(h). For all algorithms, the required number of iterations increases as the grid size increases, with PISO (Fig. 2(b)) requiring the minimum and SIMPLEST/PRIME (Fig. 2(f)) the maximum number of iterations on all grids. This performance of SIMPLEST/PRIME is due to the explicit methodology followed in solving the momentum equations. The convergence histories of SIMPLE, SIMPLEC, SIMPLEM, and SIMPLEX (Figs. 2(c), 2(d), 2(e), and 2(g), respectively) are very similar with SIMPLEM (Fig. 2(e)) requiring the lowest number of iterations. The convergence paths of the various algorithms over a grid of size 80 C.V. are compared in Fig. 2(h) and the above observations are easily inferred from the figure.

Problem 2: Dense gas-solid flow

A dense gas-solid flow is simulated by setting the concentration of particles $(r_{inlet}^{(d)})$ to 10^{-2} while retaining all other boundary conditions and physical properties as for the previous test case. Even though the particle volume fraction is low, the ratio of disperse phase and continuous phase mass loadings is large $(r^{(d)}\rho^{(d)} / r^{(c)}\rho^{(c)} = 20)$. Thus the disperse phase carries most of the inertia of the mixture. Due to the small variation in the gas velocity (i.e. the equilibrium velocity is 4.96 m/s), it may be assumed constant and the variation in particle

velocity can be obtained again from Eq. (28). The comparison between the numerical and analytical particle velocity distributions displayed in Fig. 3(a) confirms once more the validity and accuracy of the numerical procedure. The convergence paths for all algorithms and over all grid systems used are displayed in Figs. 3(b)-3(h). In general, higher number of iterations is required to reach the desired level of convergence on a given grid as compared to the dilute case due to the increased importance of the inter-phase term. The general convergence trend is similar to that of the dilute problem with PISO requiring the minimum and SIMPLEST the maximum number of iterations. The SIMLEM algorithm (Fig. 3(e)) is seen to require a slightly lower number of iterations on the finest grid as compared to SIMPLE (Fig. 3(c)), SIMPLEC (Fig. 3(d)), and SIMPLEX (Fig. 3(c)). As depicted in Figs. 3(f) and 3(h), the performance of SIMPLEST/PRIME is poor in comparison with other algorithms for the same reasons stated above.

Problem 3: Dilute bubbly flow

Here a bubbly flow is considered by regarding the continuous phase as water and the disperse phase as air. Thus, for the same configuration displayed in Fig. 1(b), the disperse and continuous phase density ratio $(\rho^{(d)} / \rho^{(c)})$ is set to 10^{-3} , the bubble inlet volume fraction $(r_{inlet}^{(d)})$ is assigned the value 0.1, while values for other physical properties and inlet conditions are retained as for the previous cases. This is a strongly coupled problem and represents a good test for the numerical procedure and performance of the algorithms. Based on the inlet conditions, the equilibrium velocity for this strongly coupled problem (Eq. (27)) is 4.6 m/s and is realized in a distance too small to be correctly resolved by any of the grid networks used. As expected, the axial velocity profiles displayed in Fig. 4(a) indicate that both phases reach the equilibrium velocity of 4.6 m/s over a very short distance from the inlet section and remain constant afterward. The relative convergence characteristics of the various algorithms (Figs. 4(b)-4(h)) remain the same. However, most algorithms require larger number of

iterations as compared to the dilute gas solid flow case due to the stronger coupling between the phases. Consistently, the PISO (Fig. 4(b)) and SIMPLEST/PRIME (Fig. 4(f)) algorithms need the lowest and highest number of iterations, respectively. As in the previous two cases, the convergence attributes of SIMPLE (Fig. 4(c)), SIMPLEC (Fig. 4(d)), SIMPLEM, and SIMPLEX (Fig. 4(g)) are very similar with SIMPLEM consistently requiring a lower number of iterations. The large difference in performance between SIMPLEST/PRIME and the remaining algorithms is clearly demonstrated in Fig. 4(h).

Problem 4: Dense bubbly flow

A dense bubbly flow is simulated by setting the concentration of bubbles to $r_{inlet}^{(d)} = 0.5$ while holding the values of other physical properties and boundary conditions as in the previous case. Coalescence of bubbles which may occur with such high value of void fraction is not accounted for here. The analytical solution is the same as in the previous case with the equilibrium velocity, as computed from Eq. (27), being 3 m/s. As depicted in Fig. 5(a), the equilibrium velocity obtained numerically is exact. With the exception of requiring higher number of iterations to reach the desired level of convergence, the performance of the various algorithms (Figs. 5(b)-5(h)) vary relatively in a manner similar to what was previously discussed and deemed redundant to be repeated.

CPU time: Horizontal particle/bubble transport

As depicted in Fig. 6, the normalized CPU times required by the various algorithms increase with increasing grid density. For the dilute gas-solid problem (Fig. 6(a)), it is hard to see any noticeable difference in the CPU times of SIMPLE, SIMPLEM, and SIMPLEX. The SIMPLEC and PISO algorithms require slightly lower and higher computational efforts, respectively, as compared to SIMPLE. The worst performance is for SIMPLEST which degenerates to PRIME in the absence of diffusion and results in a fully explicit solution scheme. For the dense gas-solid flow (Fig. 6(b)), the computational times needed by

SIMPLE, SIMPLEC, SIMPLEM, and SIMPLEX are nearly identical. PISO, however, requires higher computational effort (50% more than SIMPLE on the finest meshes (80 and 160 C.V.)). The computational effort needed by SIMPLEST/PRIME is however the most extensive and is nearly 500% the one needed by SIMPLE on the finest mesh.

The normalized CPU time of SIMPLEST/PRIME for the bubbly flow problems (Figs. 6(c) and 6(d)) is lower than in the previous two problems due to a higher rate of increase in the time needed by other algorithms (the computational time of all algorithms has increased). The relative performance of the various algorithms is nearly as described earlier with the time required by of PISO, SIMPLE, SIMPLEC, and SIMPLEX being on average the same. The SIMPLEST/PRIME algorithm however, requires nearly 400% (Fig. 6(c)) and 150% (Fig. 6(d)) the time needed by SIMPLE on the finest mesh for the dilute and dense bubbly flow, respectively. The best performance for the dense bubbly flow problem is for SIMPLEM, which requires about 50% less effort on the finest mesh than SIMPLE.

Vertical particle/bubble transport

For the configuration shown in Fig. 1(b), the gravitational acceleration g is considered to be operating in the negative x -direction and assigned the constant value of 10 m/s^2 . The length of the computational domain is extended to $L=20\text{m}$ in order to allow the particle/bubble phase to reach its final settling velocity relative to the continuous phase, at which the gravitational force balances the drag force [54]. Unlike the horizontal transport problems, diffusion in the continuous phase is retained. Moreover, the inter-phase drag force is calculated using Eqs. (25) and (26) and the drag coefficient, C_D , is considered to be particle Reynolds number dependent and calculated as:

$$C_D = \frac{24}{\text{Re}_p} + 0.44 \quad \text{with} \quad \text{Re}_p = \frac{2r_p V_{slip}}{g^{(c)}} \quad (29)$$

Problem 5: Dilute gas-solid flow

The material properties and boundary conditions considered for this case are given by:

$$\rho^{(d)} / \rho^{(c)} = 1000, g^{(c)} = 10^{-5}, r_p = 1 \text{ mm} \quad (30)$$

$$V_{inlet}^{(c)} = 100 \text{ m/s}, V_{inlet}^{(d)} = 10 \text{ m/s}, r_{inlet}^{(d)} = 10^{-5} \quad (31)$$

The large velocity boundary condition is used to ensure that the solid phase does not exit the inlet. The predicted air and particle velocity distributions depicted in Fig. 7(a) are in excellent agreement with similar predictions reported in [45]. As shown in Figs. 7(b)-7(h), the mass residuals tend to slightly increase at the beginning of the iterative process, stagnate over a number of iterations (this number increases with increasing grid size), and then decrease rapidly to the desired level of convergence. This behavior is also true for the horizontal case discussed earlier and is attributed to the approximations introduced to the pressure correction equation especially with regard to neglecting second order correction terms, which may be important at the beginning of the iterative process. Once these neglected terms become unimportant, the rate of convergence increases significantly. Retaining these terms could have improved the convergence rate but this has not been considered in this work.

As depicted in Fig. (7), the number of iterations required for the solution to reach the desired level of convergence is very close to that needed in the similar horizontal transport case. This is equally true with regard to the relative performance of the various algorithms. The performance of SIMPLEST (Fig. 7(f)) and PRIME (Fig. 7(g)) is very close due to the fact that the implicitness introduced by the diffusion of the continuous phase does not seem to be that important. However, both require on the finest mesh almost 430% the number of iterations needed by SIMPLE. The number of outer iterations needed by SIMPLEX and SIMPLEC is very close to that of SIMPLE, while SIMPLEM entails lower number of iterations. Again, PISO requires the lowest number of iterations.

Problem 6: Dense gas-solid flows

The only difference between this case and the previous one is in the particles' volume fraction, which is set to $r_{inlet}^{(d)} = 10^{-2}$. Predicted air and particle velocity profiles are displayed in Fig. 8(a) while mass residuals are presented in Figs. 8(b)-8(h). Higher number of iterations is needed in comparison with the dilute case due to the higher mass-loading ratio. Besides that, the convergence behavior is similar to the previous cases with SIMPLEST (Fig. 8(f)) and PRIME (Fig. 8(g)) requiring the highest number of iterations and PISO (Fig. 8(b)) the lowest number of iterations. The number of iterations needed by SIMPLE, SIMPLEC, and SIMPLEX (compare Figs. 8(c), 8(d), and 8(h)) is very close. The SIMPLEM algorithm (Fig. 8(e)) needs on the finest mesh about 25% less iteration than SIMPLE (Fig. 8(c)).

Problem 7: Dilute bubbly flows

In this problem, the continuous phase is water and the disperse phase is air. With the exception of setting $\rho^{(d)} / \rho^{(c)}$ to 10^{-3} , $V_{inlet}^{(c)}$ and $V_{inlet}^{(d)}$ to 1, and $r_{inlet}^{(d)}$ to 0.1, other physical properties are the same as those considered earlier. This is a very difficult problem to get convergence to unless the proper under-relaxation is used. By starting with relatively high under-relaxation factors, the number of iterations needed with all algorithms was found to be very high. In order to get feasible solutions, the under-relaxation factors during the first 20 iterations were set to 0.05 and then increased to the desired values. This was found to greatly improve the convergence rate and to generate solutions with nearly the same computational effort as in the previous cases. In addition, this treatment has greatly improved the performance of SIMPLEST and PRIME and has decreased their required number of iterations to values similar to those needed by other algorithms (Figs. 9(b)-9(h)). In fact, SIMPLEST and PRIME are performing slightly better than SIMPLE for this particular problem. Overall, none of the algorithms shows an outstanding superiority in performance over others.

Problem 8: Dense bubbly flows

The results for this problem, which are obtained by setting the volume fraction at inlet to 0.5 for the physical situation, material properties, and boundary conditions used in the previous case, are presented in Fig. 10. It was possible to get feasible solutions (i.e. with reasonable computational time) only when under-relaxing by inertia (i.e. through the use of false time steps). For the results presented in Fig. 10, a time step (Δt) of value 10^{-4} s is used for the velocity field of the dispersed gas phase, $\Delta t=1$ s for the volume fractions, and $\Delta t=0.01$ s for the velocity field of the liquid phase for the grids of sizes 20 and 40 C.V. and $\Delta t=0.05$ for all variables and for both phases with the dense grids (i.e. 80 and 160 C.V.). The predicted liquid and gas velocity distributions, which are in excellent accord with published data, are depicted in Fig. 10(a). The trend of convergence seems to be different than what has been presented so far with the convergence histories of the various algorithms (Figs. 10(b)-10(h)) being similar and requiring nearly the same number of iterations. It is also noticed that the number of iterations needed on the finest mesh is smaller than the numbers needed on the grids of sizes 40 and 80 C.V. Nevertheless, it was possible to obtain solutions with all GCBA algorithms.

CPU time: Vertical particle/bubble transport

As in the horizontal configuration, the normalized CPU times for the vertical particle transport problems displayed in Fig. 11 increase with increasing grid density. For the dilute and dense gas-solid flow problems (Figs. 11(a) and 11(b)), the relative performance of the various algorithms is similar. For the dilute case (Fig. 11(a)), the efficiency of PRIME is slightly better than SIMPLEST (due to the use of an explicit algebraic-equation solver), both however are about four times more expensive than all other algorithms whose performance is very comparable (i.e. of the same order of magnitude) with SIMPLEM being the least expensive (7% less than SIMPLE on the finest mesh) and PISO the most expensive (9.5% more than SIMPLE on the finest mesh). The same is true for the dense gas-solid case (Fig.

11(b)) with the performance of SIMPLE, SIMPLEC, SIMPLEM, SIMPLEX, and PISO being closer.

For the vertical bubbly flows, a noticeable change in the normalized time chart (Figs. 11(c) and 11(d)) is depicted, with the performance of SIMPLEST and PRIME showing good improvements while the performance of the remaining algorithms deterioration. As depicted, the CPU times needed by the various algorithms are of the same order of magnitude with SIMPLEM being slightly more expensive.

By comparing the behavior of the various algorithms in all problems, it is clear that the performance of SIMPLE, SIMPLEC, SIMPLEM, SIMPLEX, and PISO is consistent and require, on average, similar computational effort. The performance of the SIMPLEST and PRIME algorithms was comparable to SIMPLE for upward bubbly flows only and were, in general, the most expensive to use on all grids and for all physical situations presented here. Most importantly however, is the fact that all these algorithms can be used to predict multi-phase (in this case two-phase) flow.

Closing Remarks

The implementation of seven GCBA algorithms for the simulation of incompressible multi-fluid flow was accomplished. The performance and accuracy of these algorithms were assessed by solving a variety of one-dimensional two-phase flow problems. For each test problem, solutions were generated on a number of grid systems. Results obtained demonstrated the capability of all algorithms to deal with multi-fluid flow situations. The convergence history plots and CPU-times presented, indicated similar performances for all algorithms with the exception of SIMPLEST and PRIME.

Acknowledgments

The financial support provided by the University Research Board at AUB is gratefully acknowledged.

References

1. Beckermann, C. and Viskanta, R., "Mathematical Modeling of Transport Phenomena during Alloy Solidification," *Applied Mechanics Reviews*, vol. 46, no. 1, pp.1-27, 1993.
2. Al Taweel, A.M. and Chen, C., "A Novel Mixer for the Effective Dispersion of Immiscible Liquids," *Trans. IChemE.*, vol. 74, Part A, pp. 445-450, 1996.
3. Lines, P.C., "Gas Liquid Mass Transfer Using Surface-Aeration in Stirred Vessels with Dual Impellers," *Chemical Engineering Research and Design*, vol. 78, pp. 342-347, 2000.
4. Forrester, S.E., Rielly, C.D., and Carpenter, K.J., "Gas-Inducing Impeller Design and Performance characteristics," *Chemical Engineering Science*, vol. 53, pp. 603-615, 1998.
5. Dohi, N., Matsuda, Y., Itano, N., Shimizu, K., Minekawa, K., and Kawase, Y., "Mixing Characteristics of in Slurry Stirred Tank Reactors with Multiple Impellers," *Chem Eng Commun*, vol. 171, pp. 211-229, 1999.
6. Celik, I. and Wang, Y.Z., "Numerical Simulation of liquid circulation in a bubbly column," *Numerical Methods in Multiphase Flows*, FED vol. 91, pp. 19, ASME, New York, 1990.
7. Huang, B., "Modelisation Numerique D'ecoulements Diphasique dans les Reacteur Chimiques," Ph.D. thesis, University Claude Bernard, Lyon, 1989.
8. Boisson, N. and Malin, M.R., "Numerical Prediction of Two-Phase Flow in Bubble Columns," *International Journal for Numerical Methods in Fluids*, vol. 23, pp. 1289-1310, 1996.
9. Witt, P.J. and Perry, J.H. "A Study in Multiphase Modeling of Fluidized Beds," *Computational Techniques and Applications: CTAC95*, 1995.

10. Bouillard, L.X., Lyczkowski, R.W., and Gidaspow, D., "Porosity Distribution in a Fluidized Bed with an Immersed Obstacle," *AIChE*, vol. 35, pp.908-922, 1989.
11. Ghani, A.G., Farid, M.M., Chen, X.D., and Richards, P., " Numerical Simulation of Biochemical Changes in Liquid Food During Sterilization of Viscous Liquid using Computational Fluid Dynamics (CFD)," 10th World Congress of Food Science and Technology ,Sydney, Australia, paper number P08/40, 1999.
12. Erdal, F. M., Shirazi, S. A., Mantilla, I., and Shoham, O., "Bubble Trajectory Analysis for Design of Gas Liquid Cylindrical Cyclone Separators," SPE Paper 49309, in Proceedings of the SPE Annual Technical Conference & Exhibition, Production Operations and Engineering/General, New Orleans, LA, 1998.
13. Gomez, L. E., Mohan, R. S., Shoham, O., and Kouba, G.E., "Enhanced Mechanistic Model and Field Application Design of Gas-Liquid Cylindrical Cyclone Separators," SPE Annual Technical Conference and Exhibition, New Orleans, LA, 1998.
14. Behrouzi, P. and Crane, R.I., "Experimental and Numerical Studies of Axial-entry Reverse-flow Cyclones," Proceedings of the 6th World Filtration Congress , Nagoya, Japan, pp 474-478, May 1993.
15. Rüger, M., Hohmann, S., Sommerfeld, M., and Kohnen, G., " Euler/Lagrange Calculations of Turbulent Sprays: The Effect of Droplet Collisions and Coalescence," *Atomization and Sprays*, Vol. 10, pp. 47-81, 2000.
16. Mundo, C., Sommerfeld, M., and Tropea, C., " On the Modeling of Liquid Sprays Impinging on Surfaces," *Atomization and Sprays*, Vol. 8, pp. 625-652, 1998.
17. Spalding, D.B., " The Calculation of Free-Convection Phenomena in Gas-Liquid Mixtures," Report HTS/76/11, Mechanical Engineering, Imperial College, London, 1976.
18. Spalding, D.B., " Numerical Computation of Multi-Phase Fluid Flow and Heat Transfer," in *Recent Advances in Numerical Methods in Fluids*, eds. Taylor C.,

- Morgan K., vol. 1, pp. 139-167, 1980.
19. Spalding, D.B., "A General Purpose Computer Program for Multi-Dimensional, One and Two Phase Flow," Report HTS/81/1, Mechanical Engineering, Imperial College, London, 1981.
 20. Rivard, W.W. and Torrey, M.D., "KFIX: A Program for Transient Two Dimensional Two Fluid Flow," Report LA-NUREG-6623, 1978.
 21. Amsden, A.A. and Harlow F.H., "KACHINA: An Eulerian Computer Program for Multifield Flows," Report LA-NUREG-5680, 1975.
 22. Amsden, A.A. and Harlow F.H., "KTIFA Two-Fluid Computer Program for Down Comer Flow Dynamics," Report LA-NUREG-6994, 1977.
 23. Ferziger, J.H. and Peric, M., *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin Heidelberg, 1996.
 24. Carver, M.B., "Numerical Computation of Phase Separation in Two Fluid Flow," *ASME Journal of Fluids Engineering*, vol. 106, pp. 147-153, 1984.
 25. Carver, M.B. and Salcudean, M., "Three Dimensional Numerical Modeling of Phase Distribution of Two-Fluid Flow in Elbows and Return Bends," *Numerical Heat Transfer*, Vol. 10, pp. 229-251, 1986.
 26. Gosman, A.D., Issa, R.I., Lekakou, C., Looney, M., and Politis, S., "Multidimensional Modeling of Turbulent Two-Phase Flows in Stirred Vessels," *AIChE Journal*, vol. 38, pp. 1853-1859, 1992.
 27. Drew, D.A., "Analytical Modeling of Multiphase Flows," *Boiling Heat Transfer*, Ed. Laye, Elsevier, pp. 31-84, 1992.
 28. Ishii, M., *Thermo-Fluid Dynamic theory of Two-Phase Flow*, Eyrolles, 1975.
 29. Lo, S.M., "Mathematical Basis of a Multi-Phase Flow Model," ARER Report 13432, Harwell Laboratory, United Kingdom, 1990.
 30. Kelly, J.M., "A Four-field Approach Towards Developing a Mechanistic Model for

- Multiphase flow," Batelle, Pacific Northwest National Laboratories, Report # PNL-SA-18878, 1993.
31. Gray, W.G. and Hassanizadeh, S.M., "Averaging Theorems and Averaged Equations for Transport of Interface Properties in Multiphase systems," *International Journal of Multiphase Flow*, Vol. 15, no. 1, pp. 81-95, 1989.
 32. Rhie, C.M., "A Pressure Based Navier-Stokes Solver Using the Multigrid Method," AIAA paper 86-0207, 1986.
 33. Mitchell, C.R. "Improved Reconstruction Schemes for the Navier-Stokes Equations on Unstructured Grids," AIAA paper 94-0642, 1994.
 34. Darwish, M.S. and Moukalled, F., "Normalized Variable and Space Formulation Methodology for High-Resolution Schemes," *Numerical Heat Transfer, Part B*, vol. 26, pp. 79-96, 1994.
 35. Darwish, M. and Moukalled, F., "An Efficient Very High-Resolution scheme Based on an Adaptive-Scheme Strategy," *Numerical Heat Transfer, Part B*, vol. 34, pp. 191-213, 1998.
 36. Demirdzic, I., Lilek, Z., and Peric, M., "A Collocated Finite Volume Method For Predicting Flows at All Speeds," *International Journal for Numerical Methods in Fluids*, vol. 16, pp. 1029-1050, 1993.
 37. Moukalled, F. and Darwish, M., "A High-Resolution Pressure-Based Algorithm for Fluid Flow at All Speeds," *Journal of Computational Physics*, vol. 168, no. 1, pp. 101-133, 2001.
 38. Gaskell, P.H. and Lau, A.K.C., "Curvature compensated Convective Transport: SMART, a New Boundedness Preserving Transport Algorithm," *International Journal for Numerical Methods in Fluids*, vol. 8, pp. 617-641, 1988.
 39. Leonard, B.P., "Locally Modified Quick Scheme for Highly Convective 2-D and 3-D Flows," Taylor, C. and Morgan, K. (eds.), *Numerical Methods in Laminar and*

- Turbulent Flows, Pineridge Press, Swansea, U.K., vol. 15, pp. 35-47, 1987.
40. Darwish, M., Moukalled, F., and, Sekar, B." A Unified Formulation of the Segregated Class of Algorithms for Multi-Fluid Flow at All Speeds," Numerical Heat Transfer, Part B, vol. 40, no. 2, pp. 99-137, 2001.
 41. Moukalled, F. and Darwish, M." A Comparative Assessment of the Performance of Mass Conservation Based Algorithms for Incompressible Multiphase Flows," Numerical Heat Transfer, Part B, vol. 42, pp. 259-283, 2002.
 42. Drew, D.A., "Mathematical Modeling of Two-Phase Flow," Ann. Rev. Fluid Mech., vol. 15, pp. 261-291, 1983.
 43. Soo, S.L., Multiphase Fluid Dynamics, Science Press, Beijing, 1990.
 44. Gidaspow, D., Multiphase flow and Fluidization, Academic Press, 1994.
 45. Baghdadi, A.H.A., " Numerical Modeling of Two-Phase Flow with Inter-Phase Slip," Ph.D. Thesis, Imperial College, University of London, 1979.
 46. Patankar, S.V. and Spalding, D.B., " A Calculation Procedure for Heat Mass and Momentum Transfer in Three Dimensional Parabolic Flows," International Journal of Heat and Mass Transfer, vol. 15, pp. 1787, 1972.
 47. Spalding, D.B., "A General Purpose Computer Program for Multi-Dimensional, One and Two Phase Flow," Report HTS/81/1, Mechanical Engineering, Imperial College, London, 1981.
 48. Van Doormaal, J. P. and Raithby, G. D., " Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows," Numerical Heat Transfer, vol. 7, pp. 147-163, 1984.
 49. Acharya, S. and Moukalled, F., " Improvements to Incompressible Flow Calculation on a Non-Staggered Curvilinear Grid," Numerical Heat Transfer, Part B, vol. 15, pp. 131-152, 1989.
 50. Issa, R.I., " Solution of the Implicit Discretized Fluid Flow Equations by Operator

- Splitting," Report FS/82/15, Mechanical Engineering, Imperial College, London, 1982.
51. Maliska, C.R. and Raithby, G.D., " Calculating 3-D fluid Flows Using Non-orthogonal Grid," Proceedings of the Third International Conference on Numerical Methods in Laminar and Turbulent Flows, Seattle, pp. 656-666, 1983.
 52. Van Doormaal, J. P. and Raithby, G. D." An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows," ASME Paper 85-HT-9, Presented at the National Heat Transfer Conference, Denver, Colorado, August 4-7, 1985.
 53. Zwart, P.J., Raithby, G.D., and Raw, M.J., " An integrated Space-Time Finite-Volume Method for Moving-Boundary Problems," Numerical Heat Transfer Part B, vol. 34, pp. 257-270, 1998.
 54. Morsi, S.A. and Alexander, A.J., " An investigation of Particle Trajectories in Two-Phase Flow System," Journal of Fluid Mechanics, vol. 55, part 2, pp. 193-208, 1972.

Figure Captions

- Fig. 1 (a) Control volume, (b) the prolongation only, and (c) FMG strategies, and (d) Physical domain for the gas-particle transport problem.
- Fig. 2 (a) Comparison between the analytical and numerical particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories on the 80 C.V. grid for the horizontal dilute gas-solid flow problem.
- Fig. 3 (a) gas and particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dense gas-solid flow problem.
- Fig. 4 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dilute bubbly flow problem.
- Fig. 5 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories on the 80 C.V. grid for the horizontal dense bubbly flow problem.
- Fig. 6 Normalized CPU-times for the horizontal (a) dilute gas-solid, (b) dense gas-solid, (c) dilute bubbly, and (d) dense bubbly flow problem.
- Fig. 7 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dilute gas-solid flow problem.
- Fig. 8 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense gas-solid flow problem.
- Fig. 9 (a) gas and particle velocity distributions; and (b)-(h) convergence histories on the different grid systems for the vertical dilute bubbly flow problem.
- Fig. 10 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense bubbly flow problem.
- Fig. 11 Normalized CPU-times for the vertical (a) dilute gas-solid, (b) dense gas-solid, (c) dilute bubbly, and (d) dense bubbly flow problem.

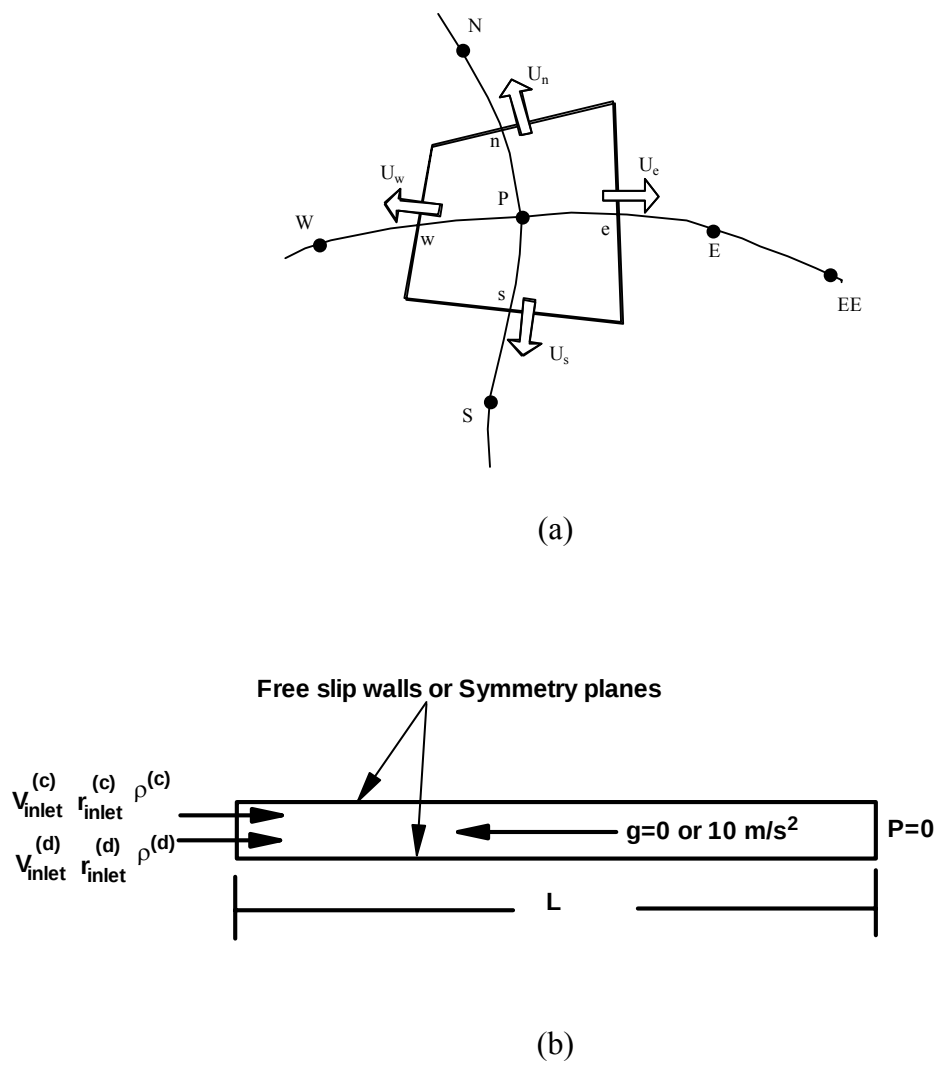
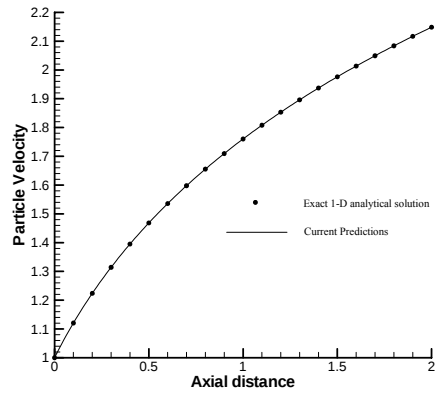
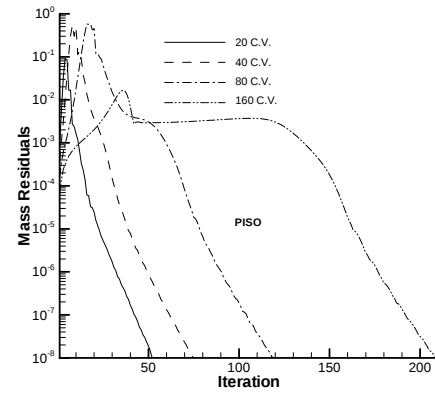


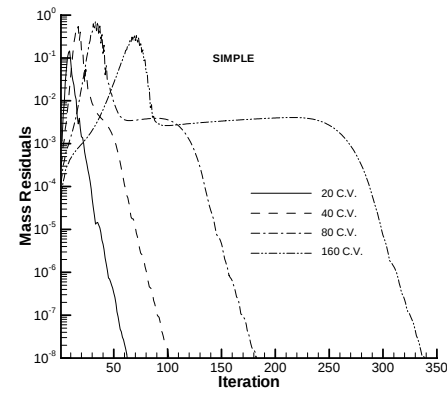
Fig. 1 (a) Control volume, (b) Physical domain for the gas-particle transport problem.



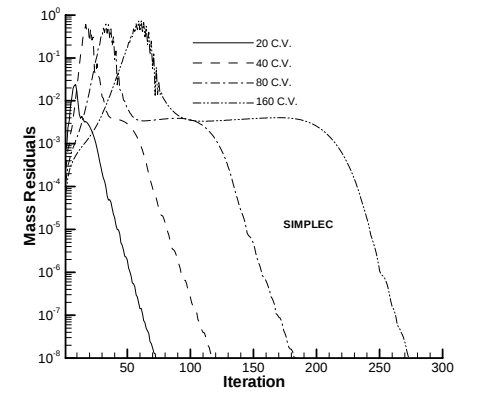
(a)



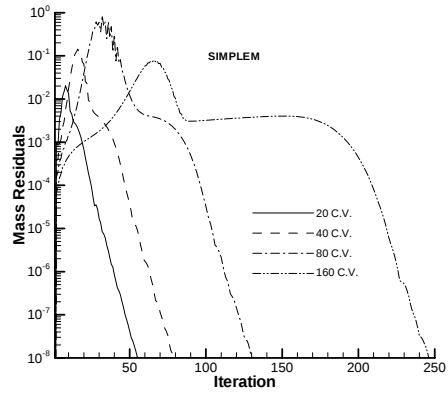
(b)



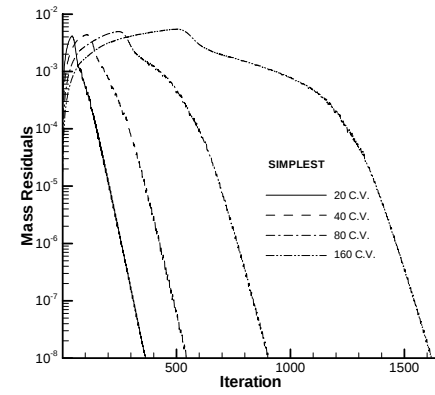
(c)



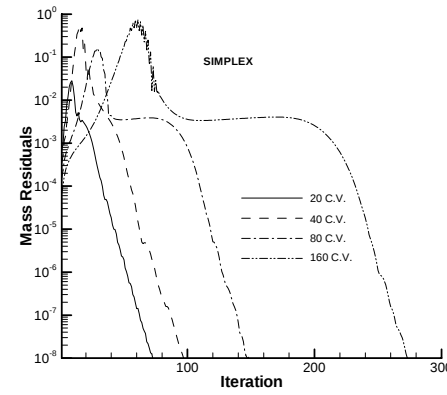
(d)



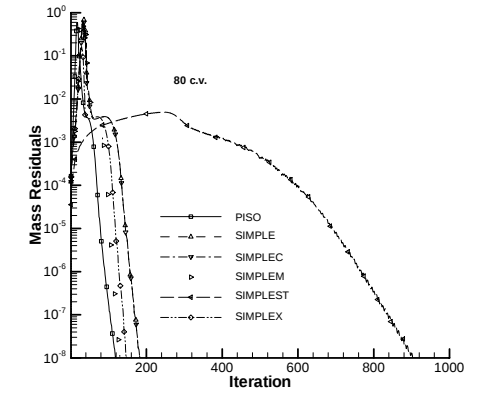
(e)



(f)



(g)



(h)

Fig. 2 (a) Comparison between the analytical and numerical particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories on the 80 C.V. grid for the horizontal dilute gas-solid flow problem.

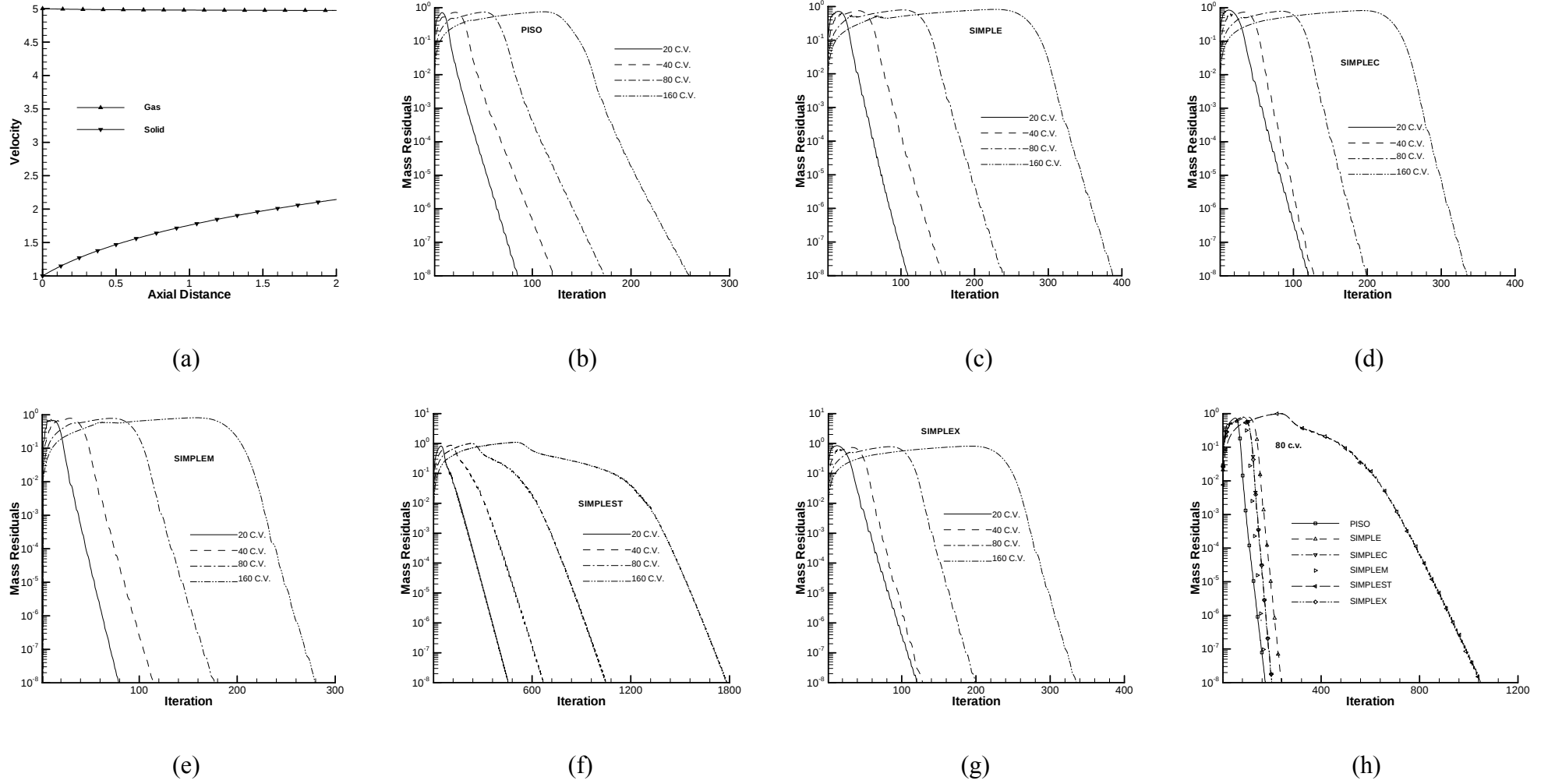


Fig. 3 (a) gas and particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dense gas-solid flow problem.

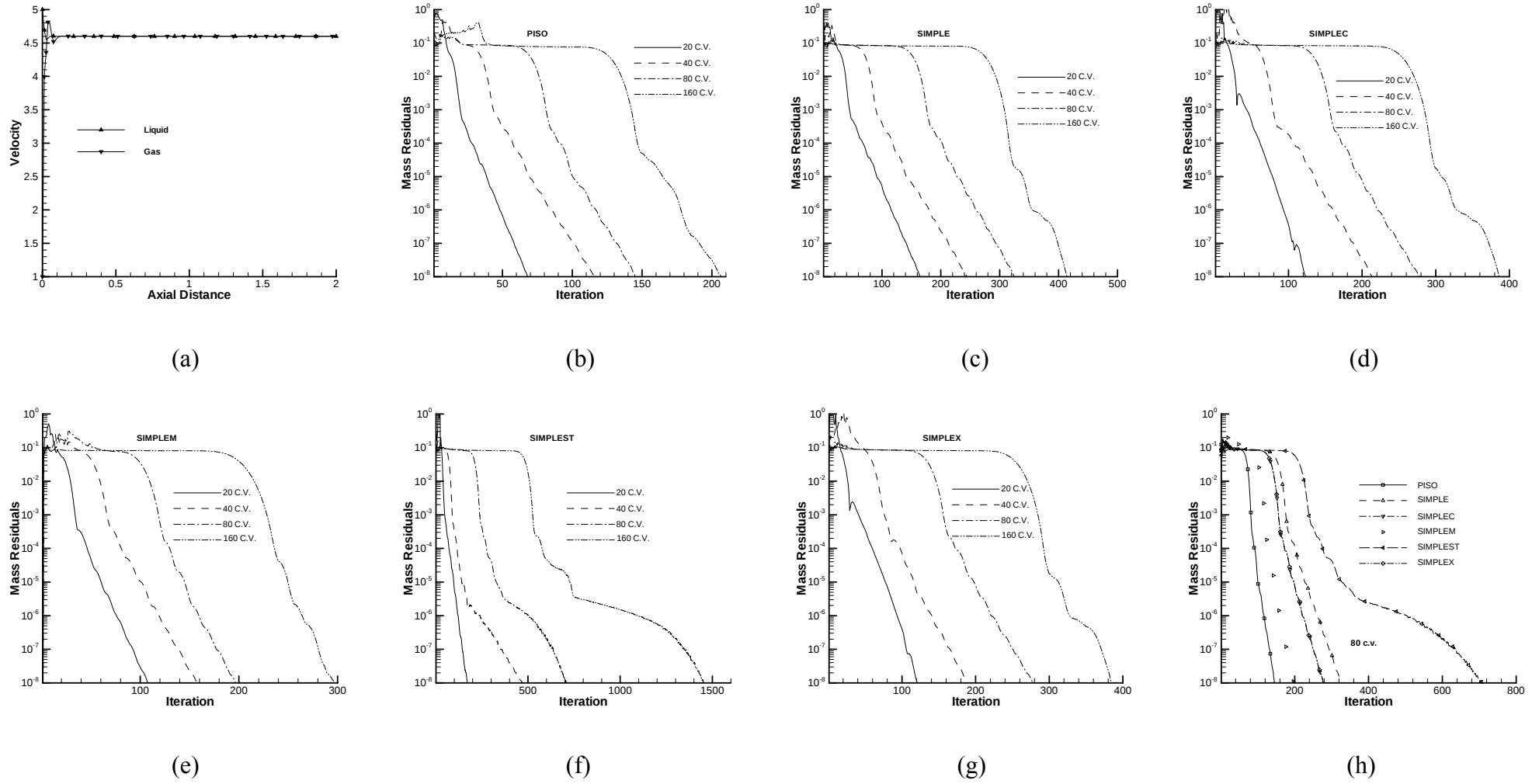
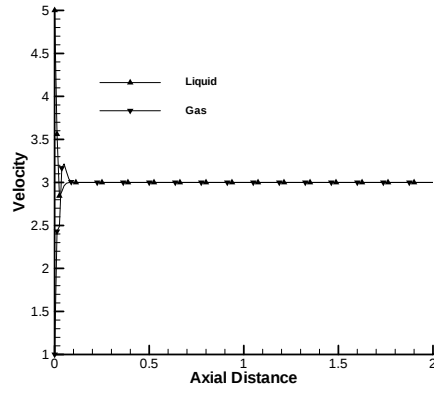
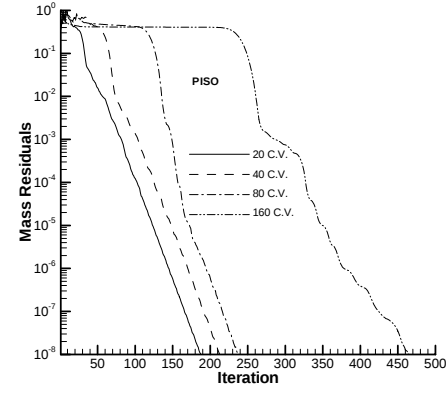


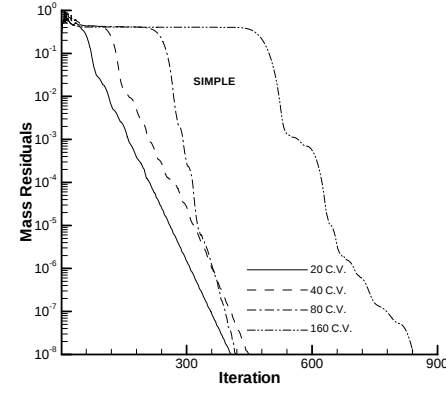
Fig. 4 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dilute bubbly flow problem.



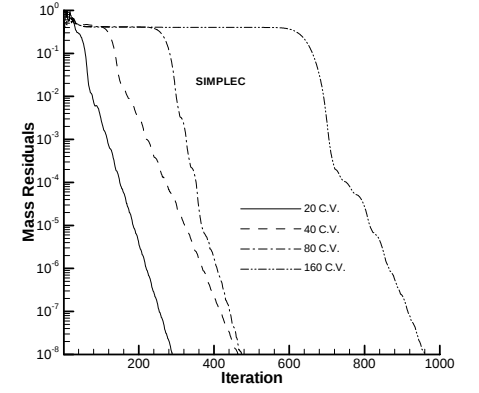
(a)



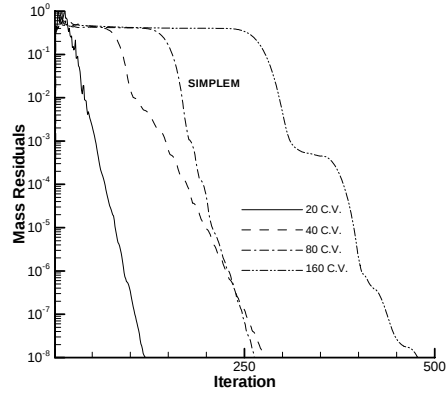
(b)



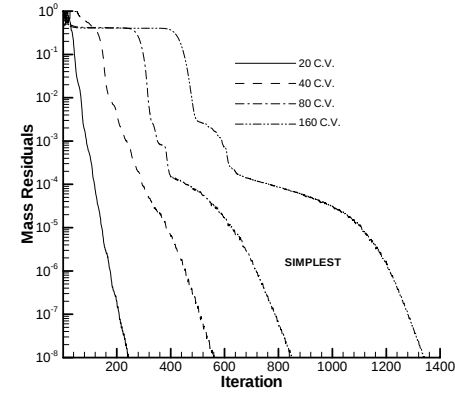
(c)



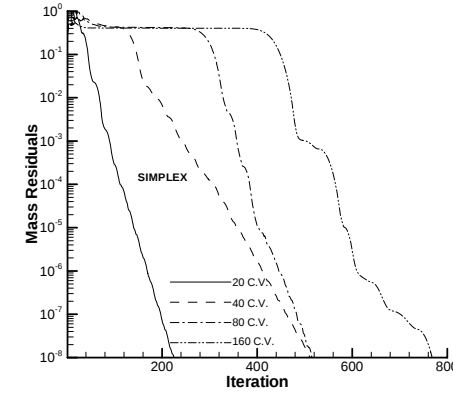
(d)



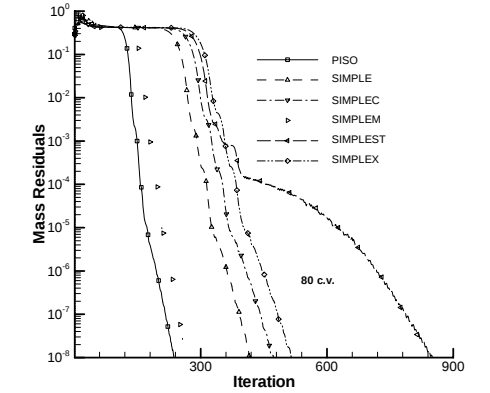
(e)



(f)

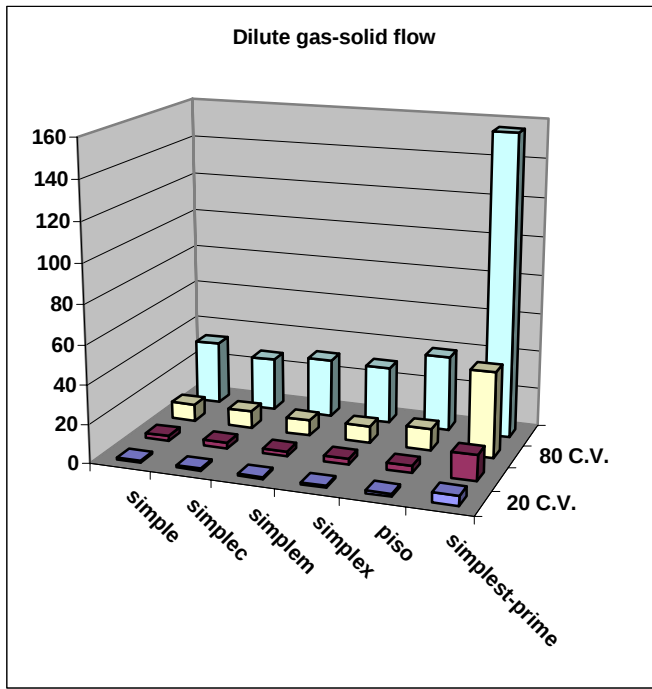


(g)

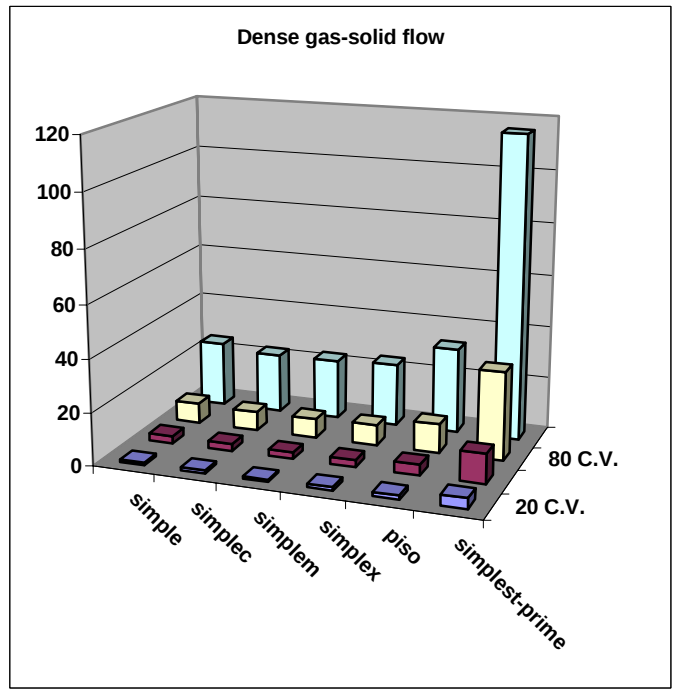


(h)

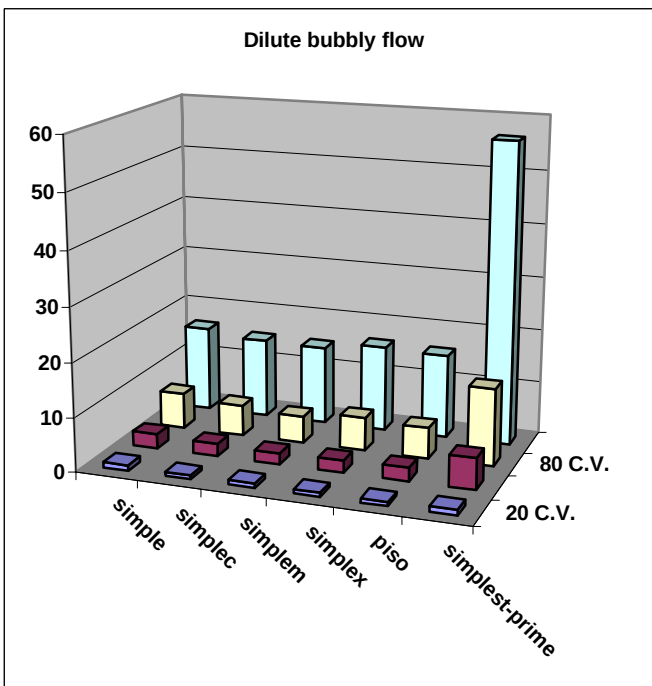
Fig. 5 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories on the 80 C.V. grid for the horizontal dense bubbly flow problem.



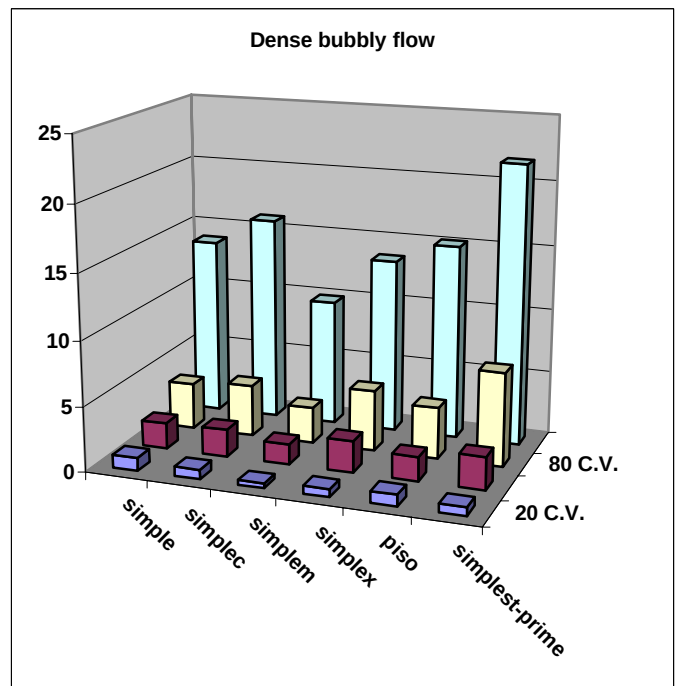
(a)



(b)

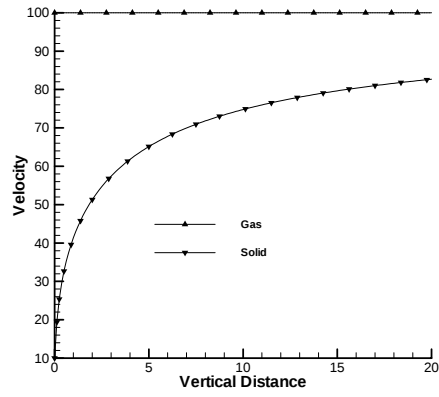


(c)

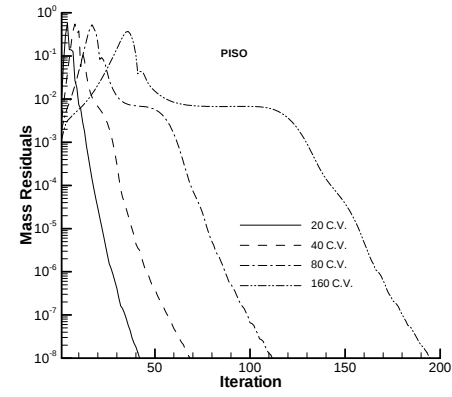


(d)

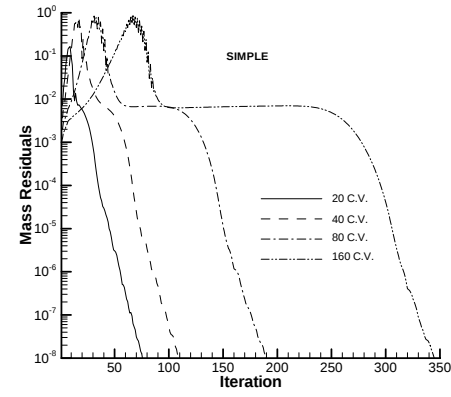
Fig. 6 Normalized CPU-times for the horizontal (a) dilute gas-solid, (b) dense gas-solid, (c) dilute bubbly, and (d) dense bubbly flow problem.



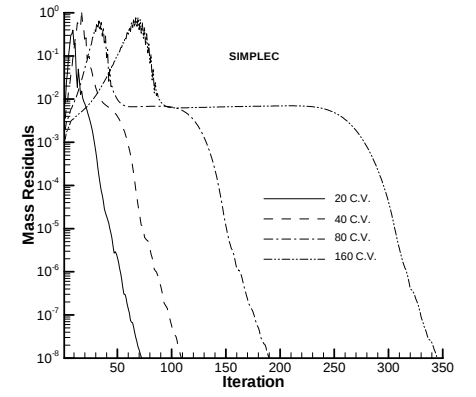
(a)



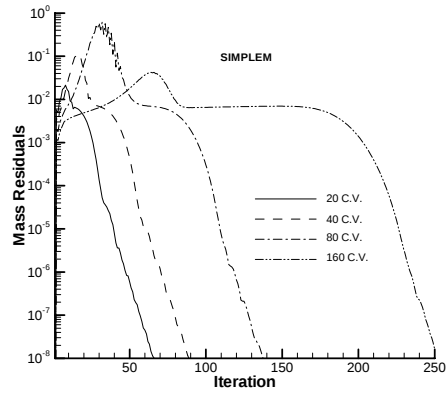
(b)



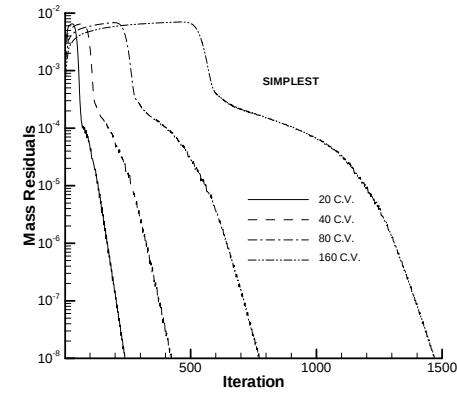
(c)



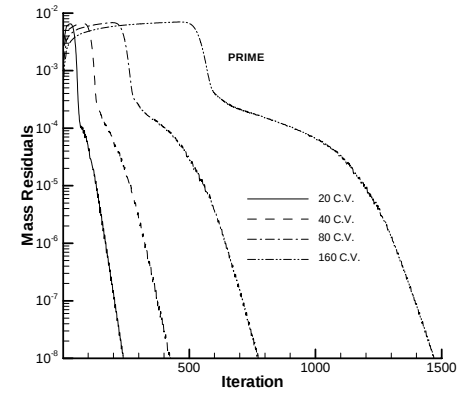
(d)



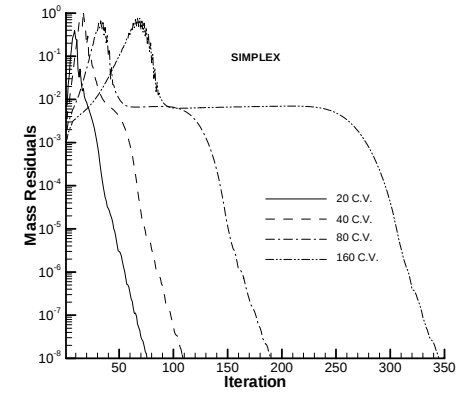
(e)



(f)



(g)



(h)

Fig. 7 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dilute gas-solid flow problem.

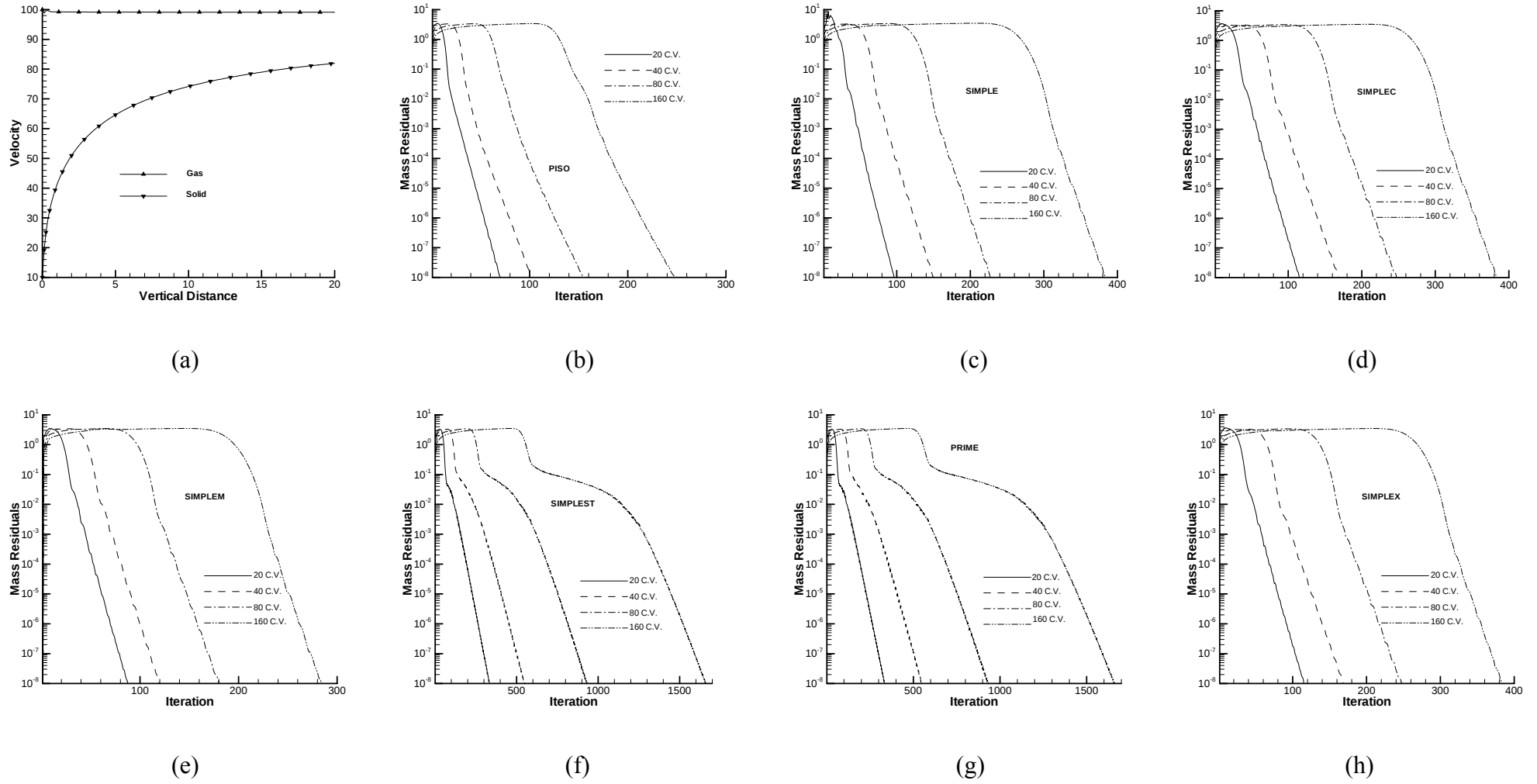
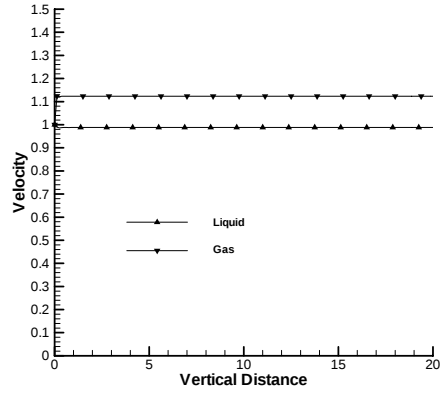
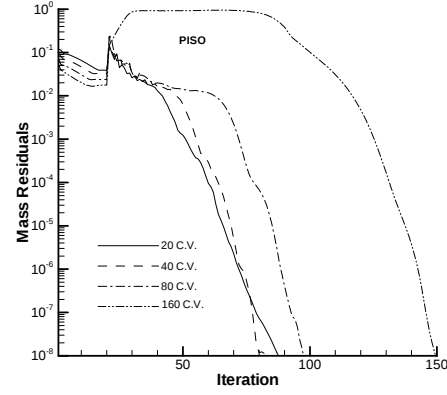


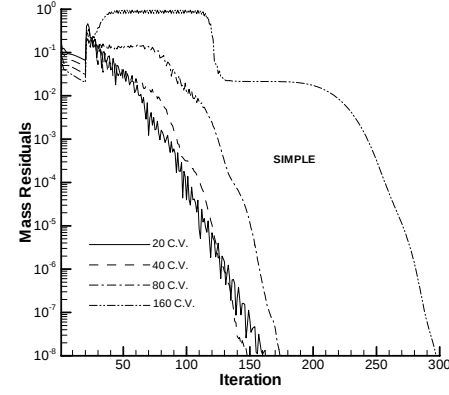
Fig. 8 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense gas-solid flow problem.



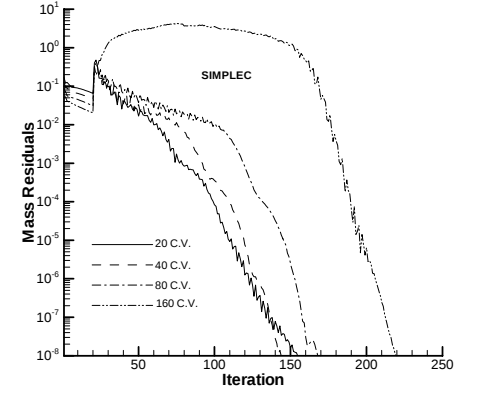
(a)



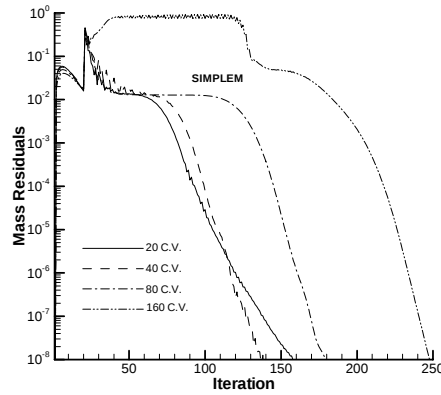
(b)



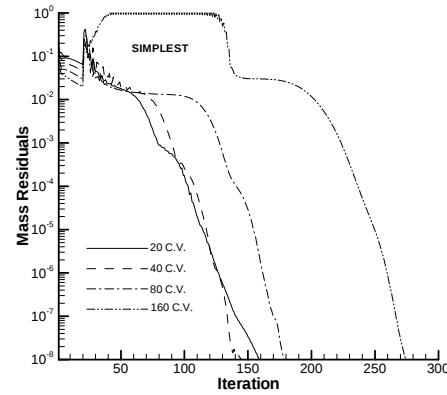
(c)



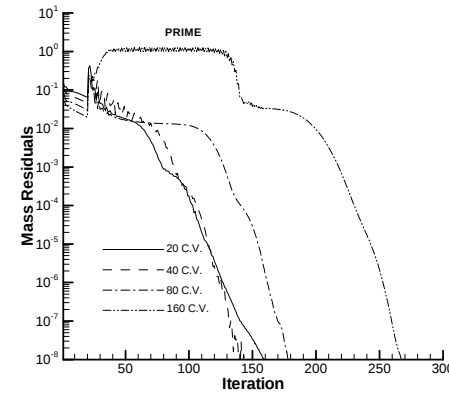
(d)



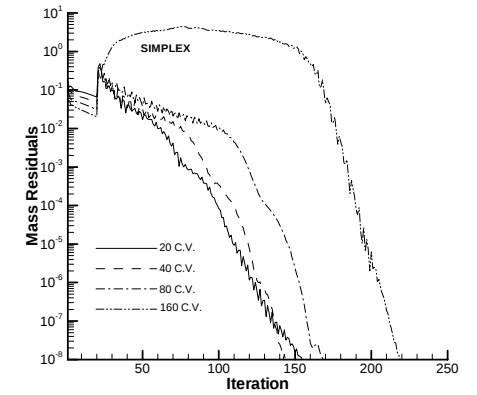
(e)



(f)

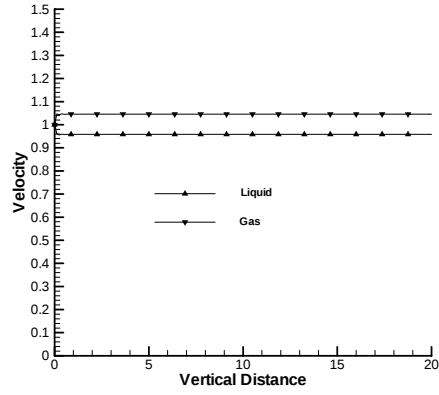


(g)

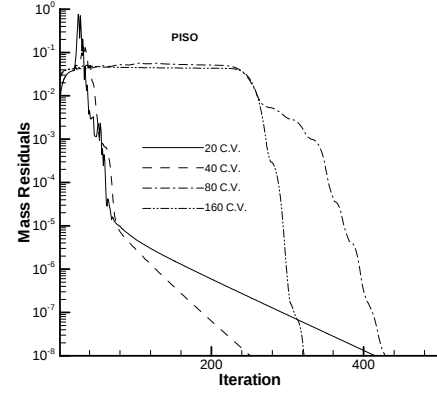


(h)

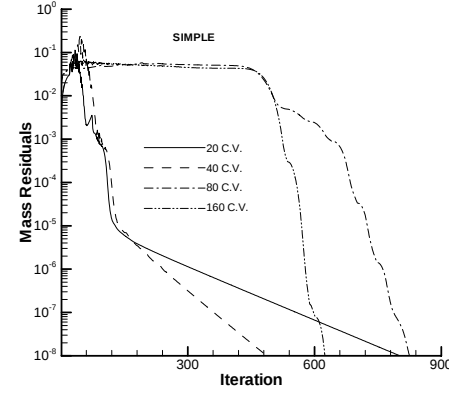
Fig. 9 (a) gas and particle velocity distributions; and (b)-(h) convergence histories on the different grid systems for the vertical dilute bubbly flow problem.



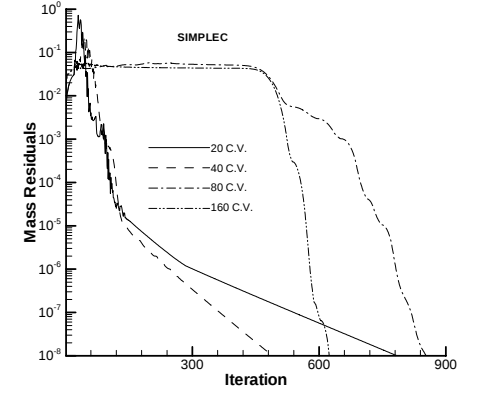
(a)



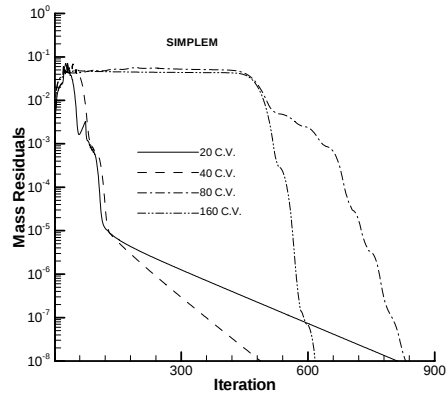
(b)



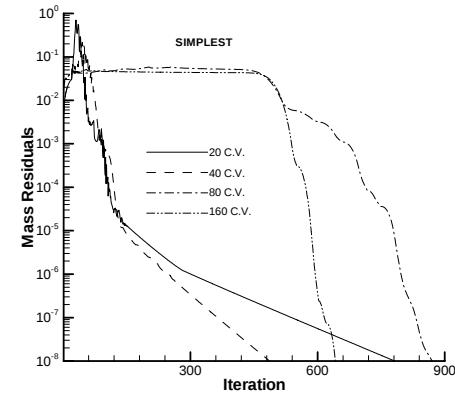
(c)



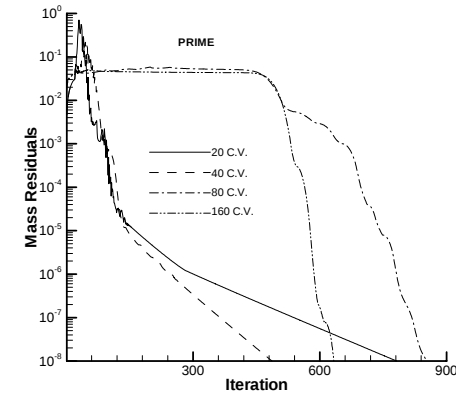
(d)



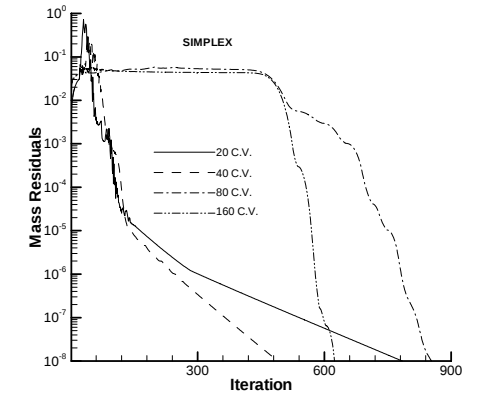
(e)



(f)

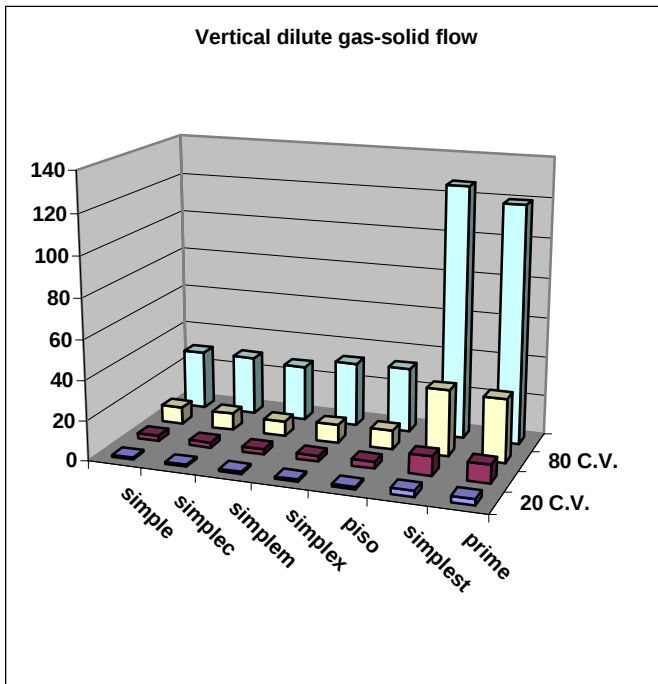


(g)

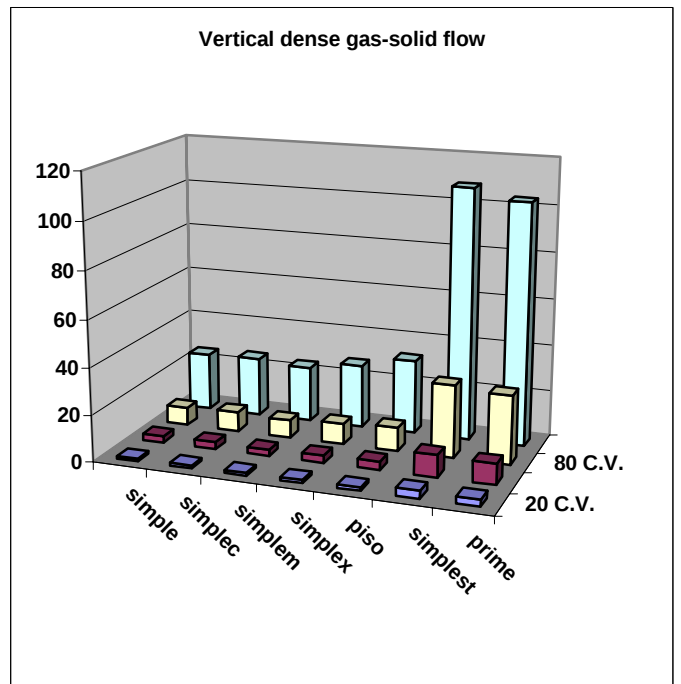


(h)

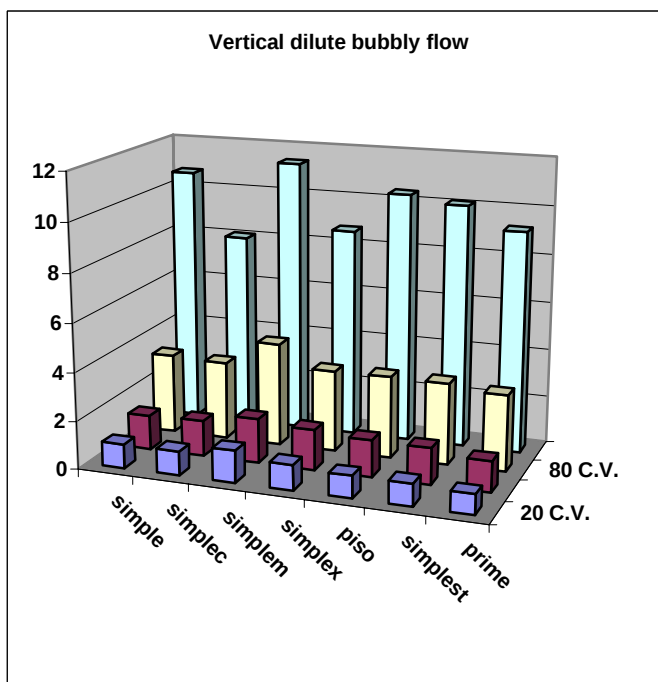
Fig. 10 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense bubbly flow problem.



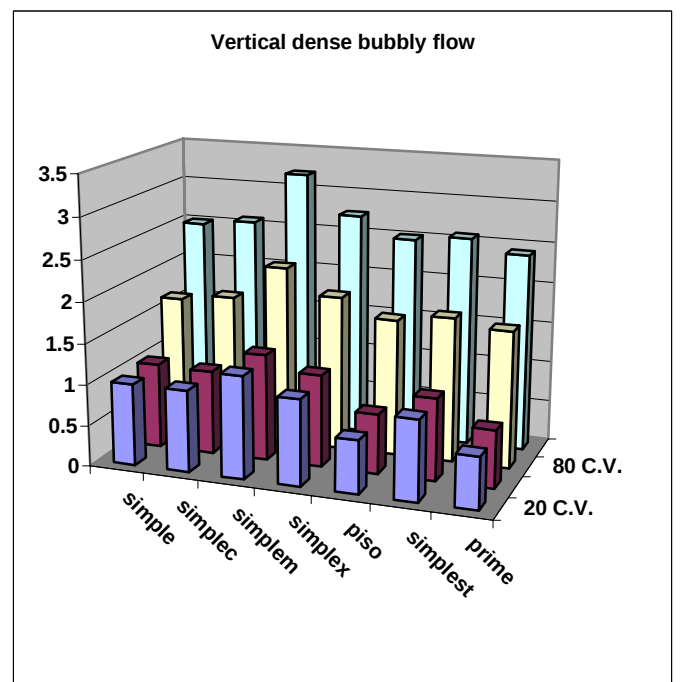
(a)



(b)



(c)



(d)

Fig. 11 Normalized CPU-times for the vertical (a) dilute gas-solid, (b) dense gas-solid, (c) dilute bubbly, and (d) dense bubbly flow problem.

