Received September 2002
Revised January 2003
Accepted January 2003

Multigrid computations with the immersed boundary technique for multiphase flows

Marianne Francois, Eray Uzgoren, Jelliffe Jackson and Wei Shyy

Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida, USA

Keywords *Heat transfer media, Liquid flow, Flow measurement*

Abstract *Multiphase flow computations involve coupled momentum, mass and energy transfer between moving and irregularly shaped boundaries, large property jumps between materials, and computational stiffness. In this study, we focus on the immersed boundary technique, which is a combined Eulerian-Lagrangian method, to investigate the performance improvement using the multigrid technique in the context of the projection method. The main emphasis is on the interplay between the multigrid computation and the effect of the density and viscosity ratios between phases. Two problems, namely, a rising bubble in a liquid medium and impact dynamics between a liquid drop and a solid surface are adopted. As the density ratio increases, the single grid computation becomes substantially more time-consuming; with the present problems, an increase of factor 10 in density ratio results in approximately a three-fold increase in CPU time. Overall, the multigrid technique speeds up the computation and furthermore, the impact of the density ratio on the CPU time required is substantially reduced. On the other hand, the impact of the viscosity ratio does not play a major role on the convergence rates.*

Nomenclature

C	= curve representing the interface
D	= diameter of computational cylinder
d	= diameter of the bubble/drop
d_p	= $2h$, twice the grid spacing
F_p	= surface tension force
Fr	= Froude number
g	= gravitational acceleration
H	= height of the computational cylinder
h	= grid spacing
$\hat{\mathbf{n}}$	= unit normal vector
p	= pressure
Re	= Reynolds number
$\mathbf{S}$	= source term in momentum equation
s	= arclength of interface
t	= time
$\mathbf{U}$	= face-centered velocity vector
$\mathbf{u}$	= cell-centered velocity vector
$\mathbf{V}_p$	= interface velocity vector
$\mathbf{x}$	= grid coordinate
$\mathbf{x}_k$	= marker coordinate

Greek symbols

Δ	= incremental difference
H	= Heaviside function

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 14 No. 1, 2003
pp. 98-115

0961-5539
DOI 10.1108/09615530410511658

Present address of Marianne Francois: Los Alamos National Laboratory, Los Alamos, New Mexico, USA.

The present work is supported by NASA and US Air Force.

δ = Dirac-delta function
κ = interface curvature
μ = dynamic viscosity
ρ = density
σ = surface tension
ϕ = any field variable
ω = interpolation weight

Superscripts
g = grid level
m = space dimension
n = time indices
$*$ = intermediate time level

Subscripts
1 = fluid 1
2 = fluid 2
e,w,n,s = face cell points in east, west, north and south respectively
cc = cell-centered
fc = face-centered
G = finest grid level

Introduction

Multiphase flow computations involve several challenging issues. For example, the momentum, mass and energy transfer between phases are coupled. When the interface moves, one needs to compute the domain shape and associated geometric information, such as curvature, normal and projected area/volume, as part of the solution, which adds nonlinearity to the problem, and can create difficulties in grid generation. Oftentimes, there are large property jumps across the interface, e.g. the density ratio between vapor and water under standard sea level conditions is around 1,000, which results in multiple time and length scales and computational stiffness. To deal with these issues, numerous numerical techniques have been developed, each with its own merits and difficulties. There are three categories: Lagrangian, Eulerian and combined Eulerian-Lagrangian methods, which are reviewed by Shyy *et al.* (1996, 2001).

In this study, we focus on the combined Eulerian-Lagrangian method. Specifically, the immersed boundary technique (Francois, 2002; Francois and Shyy, 2002; Peskin, 1977, 2002; Shyy *et al.*, 2001; Tryggvason *et al.*, 2001; Udaykumar *et al.*, 1997) is used. In particular, we investigate the performance improvement using the multigrid technique (Brandt, 1977; Briggs, 2000; Shyy, 1994) in the context of the projection method (Francois, 2002; Ye *et al.*, 1999). The present approach tracks the interface with the Lagrangian method using massless markers, while the field equation computations are carried out with the Eulerian method on fixed, Cartesian meshes.

In the fractional step method, the Poisson pressure equation is responsible for the majority of the computational cost. As detailed by Shyy (1994), the pressure equation, which is a diffusion-type for low speed flows, exhibits slower convergence rates than the convective-diffusive ones when employing iterative matrix solvers. Therefore, improvement on the solver of the Poisson equation can accelerate the overall performance of the immersed boundary method. In this work, we investigate the multigrid technique. Although the multigrid method is well established for many single-phase fluid flow problems (Hackbusch, 1980; Luchini and Dalascio, 1994; McCormick, 1987; Shyy, 1994), its application to moving boundary problems is not widely reported

(Udaykumar *et al.*, 2001). The inclusion of the immersed boundary in the flow field for two-phase flows changes the characteristics of the coefficient matrix. The property jump between phases also alters the convergence behavior. In this study, we assess the interplay between the multigrid computation, the moving boundary separating two fluids, and the effect of the property ratios between phases. Two problems, one involving a rising bubble in a liquid medium and the other a liquid drop impinging on a solid surface, serve as test cases to evaluate the multigrid performance. The density and viscosity are varied to offer ranges of property jumps between phases. The problems studied are axisymmetric, with constant properties within each phase.

Numerical approach

The governing equations of mass and momentum conservation for the unsteady, viscous, incompressible flow field (in each material) are given in equations (1) and (2), respectively.

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u}\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}} \nabla \cdot (\mu \nabla \mathbf{u}) + \frac{1}{\mathrm{We}} \int_{C(t)} \sigma \kappa \hat{\mathbf{n}} \delta(\mathbf{x} - \mathbf{x}_k)\, \mathrm{d}s + \frac{1}{\mathrm{Fr}} \rho g \tag{2}$$

The projection method (Chorin, 1968; Francois, 2002; Ye *et al.*, 1999) is used to solve the above equations. It consists of splitting the solution procedure into two distinct steps. In the first step, the momentum equation without the pressure term is solved for an intermediate solution of the velocity field as shown in equation (3), which is derived on a cell-centered, collocated grid arrangement, between the primary dependent variables and the mass and momentum fluxes, using the second order Adams-Bashforth scheme for the convection term and the Crank-Nicolson scheme for the viscous term (Ye *et al.*, 1999).

$$\frac{\rho \mathbf{u}^* - \rho \mathbf{u}^n}{\Delta t} + \frac{1}{2}[3\nabla \cdot \rho \mathbf{U}^n \mathbf{u}^n - \nabla \cdot \rho \mathbf{U}^{n-1} \mathbf{u}^{n-1}] = \frac{1}{2\mathrm{Re}}[\mu \nabla^2 u^n + \mu \nabla^2 u^*] + \mathbf{S} \tag{3}$$

where $\mathbf{U}$ is the face-centered velocity, $\mathbf{u}$ is the cell-centered velocity and $\mathbf{S}$ is the source term that contains the gravitational and surface tension body forces.

For the second step, the following Poisson equation for pressure is derived and solved.

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p^{n+1} \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{U}^* \tag{4}$$

Once the pressure is obtained, the velocity field values are corrected according to equations (5) and (6).

$$\mathbf{u}^{n+1} = \mathbf{u}^{*} - \Delta t\left(\frac{1}{\rho}\nabla p^{n+1}\right)_{\mathrm{cc}} \tag{5}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^{*} - \Delta t\left(\frac{1}{\rho}\nabla p^{n+1}\right)_{\mathrm{fc}} \tag{6}$$

In the immersed boundary method, the material properties are assigned with the aid of the discrete Heaviside step function, demonstrated in equations (7) and (8), so that the underlying solver does not encounter discontinuities across the interface.

$$\rho = \rho_2 H(\mathbf{x} - \mathbf{x}_k) + \rho_1(1 - H(\mathbf{x} - \mathbf{x}_k)) \tag{7}$$

$$\mu = \mu_2 H(\mathbf{x} - \mathbf{x}_k) + \mu_1(1 - H(\mathbf{x} - \mathbf{x}_k)) \tag{8}$$

where the discrete Heaviside step function H is defined as follows:

$$H(\mathbf{x} - \mathbf{x}_k) = \begin{cases} \prod_{m=1}^{\dim} \frac{1}{2}\left(1 + \frac{\mathbf{x}_m - (\mathbf{x}_m)_k}{d_{\mathrm{p}}} + \frac{1}{\pi}\sin\frac{\pi(\mathbf{x}_m - (\mathbf{x}_m)_k)}{d_{\mathrm{p}}}\right) & \text{if } |\mathbf{x} - \mathbf{x}_k| \le d_{\mathrm{p}} \\ 1 & \text{if } \mathbf{x} - \mathbf{x}_k > +d_{\mathrm{p}} \\ 0 & \text{if } \mathbf{x} - \mathbf{x}_k < -d_{\mathrm{p}} \end{cases} \tag{9}$$

where dim is the space dimension, $d_{\mathrm{p}} = 2h$ with h being the grid spacing, $\mathbf{x}$ is the grid coordinate, and $\mathbf{x}_k$ is the interfacial marker coordinate.

The interface force acting on the marker points is spread to the nearby grid points using the discrete Delta function, defined as follows

$$\delta(\mathbf{x} - \mathbf{x}_k) = \begin{cases} \prod_{m=1}^{\dim} \frac{1}{2d_{\mathrm{p}}}\left(1 + \cos\frac{\pi(\mathbf{x}_m - (\mathbf{x}_m)_k)}{d_{\mathrm{p}}}\right) & \text{if } |\mathbf{x} - \mathbf{x}_k| \le d_{\mathrm{p}} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The surface force is incorporated into the field equation in the form of the body force based on the following formula:

$$\mathbf{F}_{\mathrm{p}} = \sum_{k} \sigma \kappa_k \hat{\mathbf{n}}_k \delta(\mathbf{x} - \mathbf{x}_k) \Delta s_k \tag{11}$$

Including the above term as a source term in the momentum equation means that the surface effects, instead of being applicable at the zero thickness interface, now spreads over a circle of radius $2h$ around the cell center, as shown in Figure 1.

The interface velocity is obtained for each marker (Figure 1) with the help of the interfacial continuity condition and it is calculated by using equation (12).

$$\mathbf{V}_{\mathrm{p}} = \sum_{ij} \mathbf{u}_{ij} \delta(\mathbf{x} - \mathbf{x}_k) h^2 \tag{12}$$

Equations (3) and (4) are discretized using the finite-volume technique. The final discrete form of the advection-diffusion and pressure equations (equations (3) and (4), respectively) are cast into the following generalized form:

$$a_{\mathrm{p}} \phi_{\mathrm{p}} = a_{\mathrm{e}} \phi_{\mathrm{e}} + a_{\mathrm{w}} \phi_{\mathrm{w}} + a_{\mathrm{n}} \phi_{\mathrm{n}} + a_{\mathrm{s}} \phi_{\mathrm{s}} + b \tag{13}$$

which results in a five-point stencil. In matrix form, equation (13) reads:

$$[A]\,\{\phi\} = \{B\} \tag{14}$$

where $[A]$ is a pentadiagonal coefficient matrix, $\{B\}$ is the source vector and $\{\phi\}$ is the solution vector. Equation (14) is solved here iteratively using the line successive over relaxation (LSOR) method (Press *et al.*, 1992) that decomposes the system into two tri-diagonal matrices, which are then solved using the Thomas algorithm.

As already mentioned and discussed by Shyy (1994), for the discretized advection-diffusion equation, the convergence rate of equation (14) is fast such that the residual can be reduced to an acceptable level typically within a few iterations. However, the pressure Poisson equation has a slower convergence rate than the advection-diffusion equation. In fact, the convergence rate can be further reduced with the inclusion of the immersed body where the fluid properties, especially the density, change abruptly. The density jump within a few cells around the interface directly modifies the terms in the coefficient matrix of equation (14). Furthermore, equation (14) is observed to be sensitive to the density ratio of the fluids, as discussed in the next section.

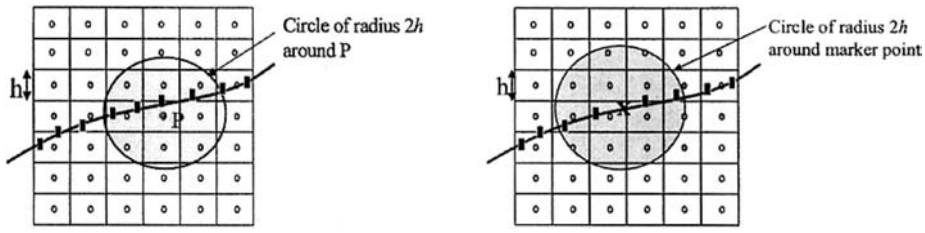


Figure 1.
Marker points considered for the estimation of the force at point P and grid points considered for the interface velocity around the marker X

The multigrid technique works on the principle that high wave number components decay faster than low wave number components. A component's wave number is considered high or low depending on the grid size. This dependence is such that low wave number components on a fine mesh behave like high wave number components on a coarse mesh as demonstrated by Ghia *et al.* (1982) and Shyy (1994). Therefore, treating the various wave number components on different grids makes it possible to accelerate the convergence rate. The multigrid procedure adopted in this work is based on a conventional W-cycle, similar to the procedure of Udaykumar *et al.* (2001), which is summarized below.

Consider a sequence of $g = 1,\ldots,G$ grids, where the grid spacing h_{g-1} on grid $g-1$ is twice that of the grid spacing on the grid level g. Equation (14) can be cast into the following form:

$$[A]_G\{\phi\}_G = \{B\}_G \tag{15}$$

where $[A]_G$ is the coefficient matrix, $\{\phi\}_G$ is the solution vector and G is the source vector on grid G, the finest grid level. The final converged solution is obtained on the finest grid, however all the computations are not carried out on this grid.

The first step is to carry out a few computations on the finest grid $g = G$, which results in:

$$[A]_g\{\phi\}_g - \{B\}_g = \{R\}_g \tag{16}$$

where $\{R\}_g$ is the residual vector. The residuals are then transferred to the next coarser grid via the restriction operator:

$$\{R\}_{g-1}^{g} = \sum_{M}(\{R\}_M)_g \tag{17}$$

where M denotes the four surrounding points on the fine grid. The following system is then solved on the coarser grid:

$$[A]_{g-1}\{\phi\}_{g-1} = -\{R\}_{g-1}^{g} \tag{18}$$

The solution vector is then transferred by the prolongation operator to the fine grid:

$$\{\phi\}_{g}^{g-1} = \sum_{M}\omega_M(\{\phi\}_M)_{g-1} \tag{19}$$

where M in this case denotes the four surrounding point on the coarse grid and ω_M denotes the interpolation weights. This process is scheduled and the final solution is obtained as

$$\{\phi\}_G = \{\phi\}_g + \{\phi\}_g^{g-1} \tag{20}$$

The multigrid technique, described earlier, is employed to solve the pressure Poisson equation (4). In the following section, we present a performance analysis of the multigrid method applied to two-phase flows: the first case being a rising bubble and the second an impinging droplet on a flat surface.

Results and discussion

Rising bubble in a viscous liquid

This study extends the work presented by Francois (2002) for the rising bubble in a viscous fluid. The same configuration is used for validation purposes and is shown in Figure 2.

A spherical bubble of diameter d, fluid 2, is placed in a close cylinder of diameter $D = 5d$ and height $H = 10d$, full of denser fluid, fluid 1. The buoyancy force drives the bubble to rise inside the cylinder. No-slip boundary condition is applied on the wall of the cylinder. The flow conditions, defined by the dimensionless parameters, determine the shape and location of the bubble.

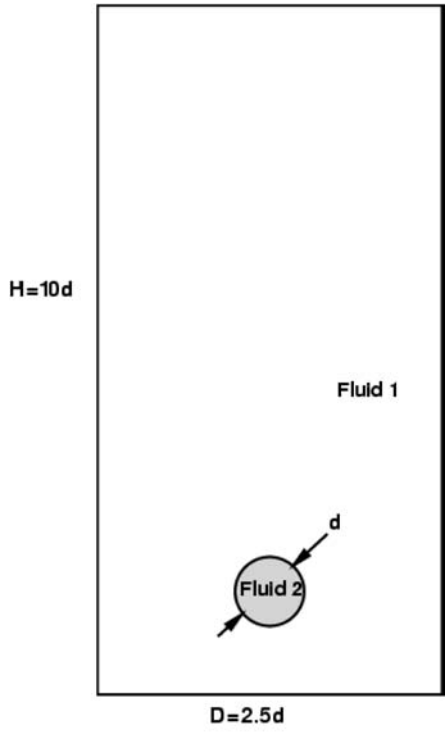


Figure 2.
Schematic of the computational setup for a single bubble rising by buoyancy

These dimensionless parameters, which include Reynolds (Re), Weber (We) and Froude (Fr) numbers, are based on the properties of fluid 1 as given in equations (21)-(23).

$$\mathrm{Re} = \frac{\rho_1 U d}{\mu_1} \tag{21}$$

$$\mathrm{We} = \frac{\rho_1 U^2 d}{\sigma} \tag{22}$$

$$\mathrm{Fr} = \frac{U^2}{gd} \tag{23}$$

where U is the characteristic speed, ρ_1 and μ_1 are the density and viscosity, respectively, of fluid 1, σ is the surface tension and g is the gravitational acceleration. The characteristic speed is calculated by using equation (24) and set to one for the cases considered.

$$U = \sqrt{gd} \tag{24}$$

In all cases, the computational domain consists of a 202×42 grid, distributed non-uniformly. Figures 3 and 4 show the typical bubble shapes in time for different density and viscosity ratios. The initial bubble starts to rise due to the effect of buoyancy in the cylinder and it eventually deforms to a steady-state shape. For a detail presentation and discussion of the physical results, we refer to Francois (2002).

Figure 5 shows the number of fine grid iterations required to reach a residual level of 1×10^{-6} at the very first time step, which requires the largest number of iterations to converge among all time steps since it starts to iterate from the initial conditions. One level represents the iteration history for a single grid computation. As demonstrated, the convergence rate improves dramatically when the level of multigrid is increased.

Figure 6 shows the convergence history at a later time instant, $t = 500\Delta t$ for a density ratio of ten. While the single grid computation requires substantially fewer fine grid iterations at the later stage than initially, the multigrid technique still exhibits noticeable improvement. With a higher ratio, 100, as shown in Figures 7 and 8, one can see that a larger number of iterations is required as the density ratio increases. The multigrid significantly accelerates the convergence rate for both density ratios.

The number of fine grid iterations required throughout the course of computation is shown in Figures 9-11 for density ratios of 10 and 100 for single grid, two and three level grids, respectively. The number of fine grid iterations required at each time step is averaged for each case. As the density ratio increases, the number of iterations required to reach the same residual level also increases.

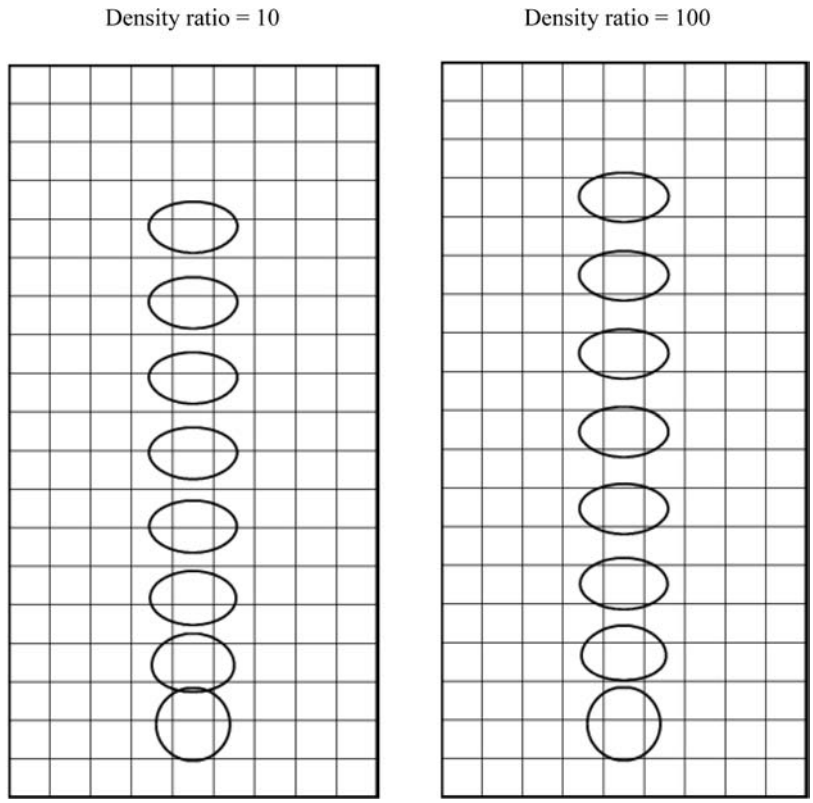


Figure 3.
Instantaneous bubble shapes at time $t = 0$, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5 for Re = 100, We = 4 and Fr = 1 for density ratio of 10, 100 at a fixed viscosity ratio of 1

Figure 12 summarizes the performance of the multigrid technique with the viscosity ratio of ten and three density ratios. The single grid computation with the density ratio of ten is used as the reference, whose CPU time is assigned as unity. Single- and multi-grid computations are based on the same time step size and with the identical number of time steps. The CPU time for all other cases is normalized by the density ratio of ten and computed with single grid. Clearly, as the density ratio increases, the single grid computation becomes substantially more time-consuming; for the present case, an increase of factor ten in density ratio results in approximately a three-fold increase in CPU time. Employing two and three level grids decreases the CPU time for each density ratio. Furthermore, the efficiency of the multigrid performance is more pronounced for larger density ratio.

Figures 13 and 14 present the effect of viscosity ratio on the performance of the multigrid technique. In Figure 13, the density ratio is held at ten while in Figure 14, it is held at 100. Again, the multigrid technique improves the convergence rate for all cases. It is worth noting that the effect of viscosity ratio

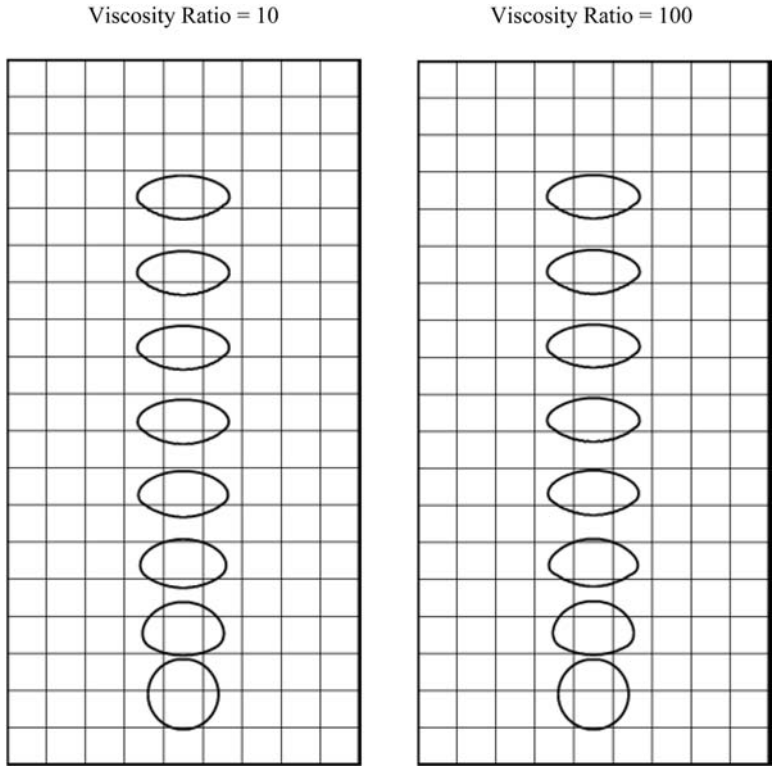


Figure 4.
Instantaneous bubble shapes at time $t = 0$, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5 for $\mathrm{Re} = 100$, $\mathrm{We} = 4$ and $\mathrm{Fr} = 1$ for viscosity ratio of 10, 100 at a fixed density ratio of ten

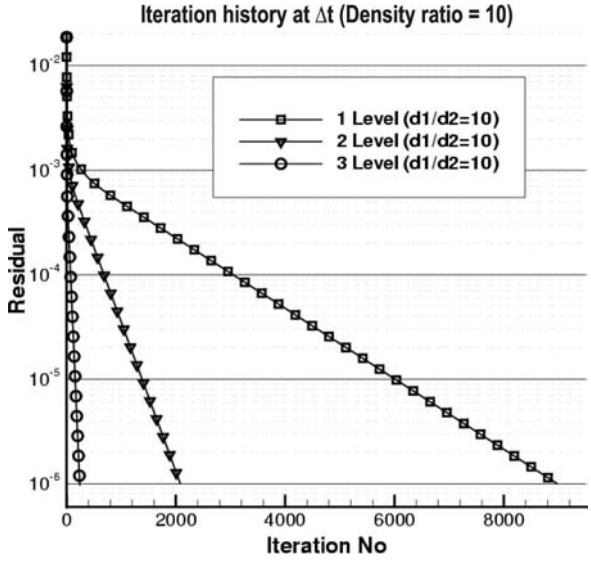


Figure 5.
Fine grid iteration history at $t = \Delta t$ for three levels at a fixed density ratio of ten for the rising bubble case

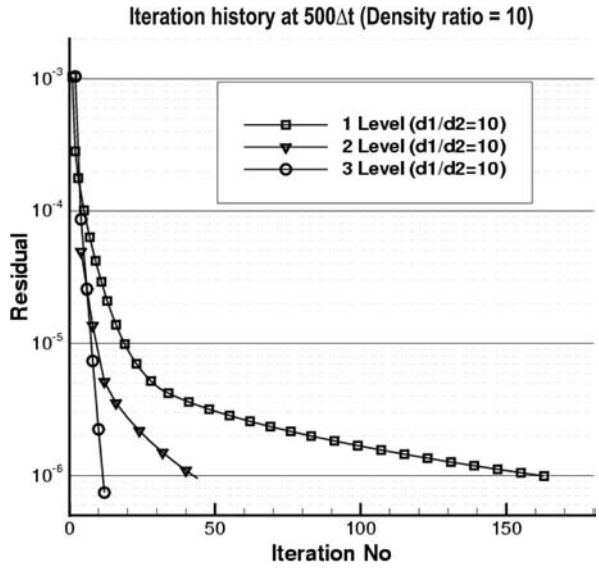


Figure 6.
Fine grid iteration history at $t = 500\Delta t$ for three levels at a fixed density ratio of ten for the rising bubble case

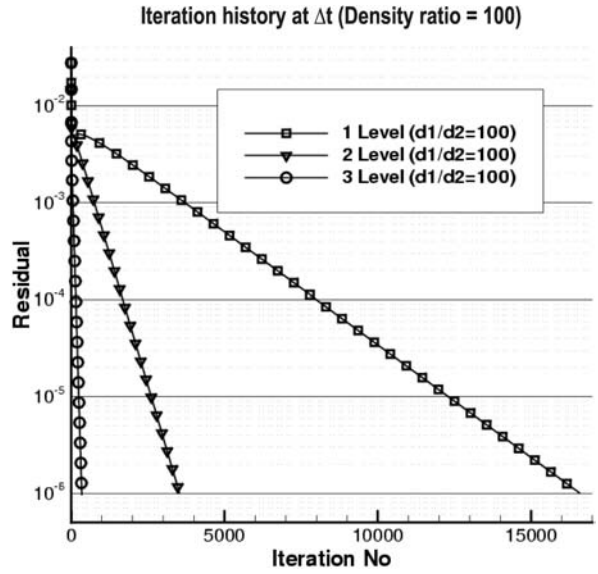


Figure 7.
Fine grid iteration history at $t = \Delta t$ for three levels at a fixed density ratio of 100 for the rising bubble case

on convergence rates is negligible. On the other hand, the multigrid improvement is greater for the higher density ratio cases.

Case II: impact of a droplet

The second problem considered is that of a liquid drop impinging on a flat, solid surface. The schematic is shown in Figure 15.

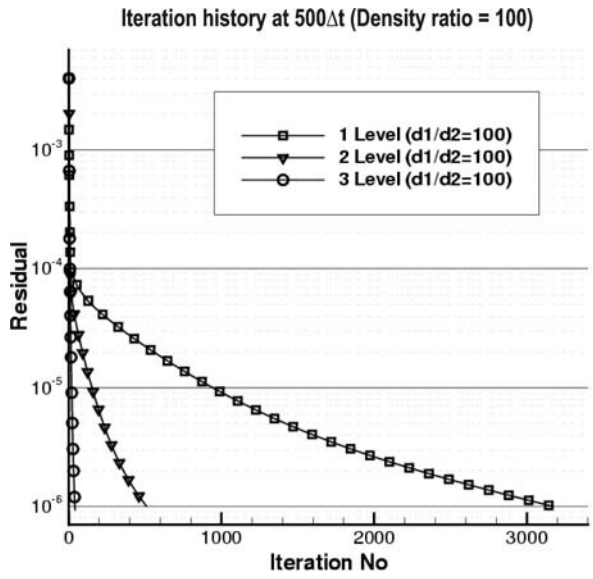


Figure 8.
Fine grid iteration history at $t = 500\Delta t$ for three levels at a fixed density ratio of 100 for the rising bubble case

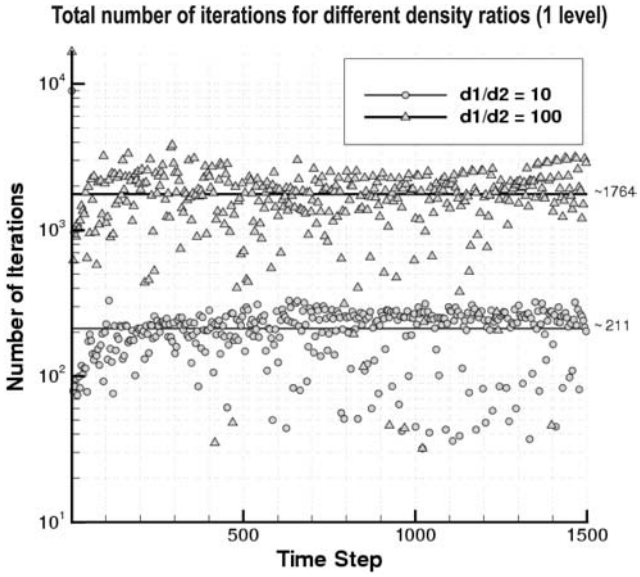


Figure 9.
Total number of fine grid iterations for density ratios of 10 and 100 for single grid computation for the rising bubble case

For this problem, in addition to the abovementioned dimensionless parameters (Re, We and Fr, based on fluid 2), the contact angle is another parameter influencing the dynamics. In this study, we extend the work of Francois (2002) and consider a case with Re = 100, We = 4, Fr = ∞ and a static contact angle of 60°. The droplet is initially a sphere and the impact velocity is set to be one.

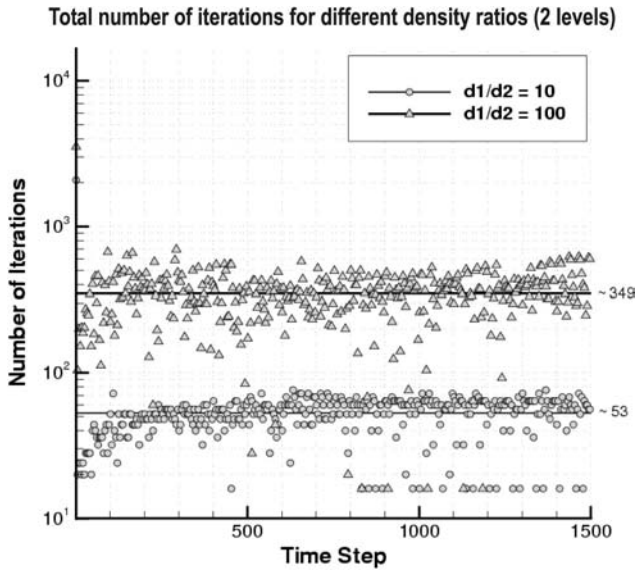


Figure 10.
Total number of fine grid iterations for density ratios of 10 and 100 with a two-level multigrid method for the rising bubble case

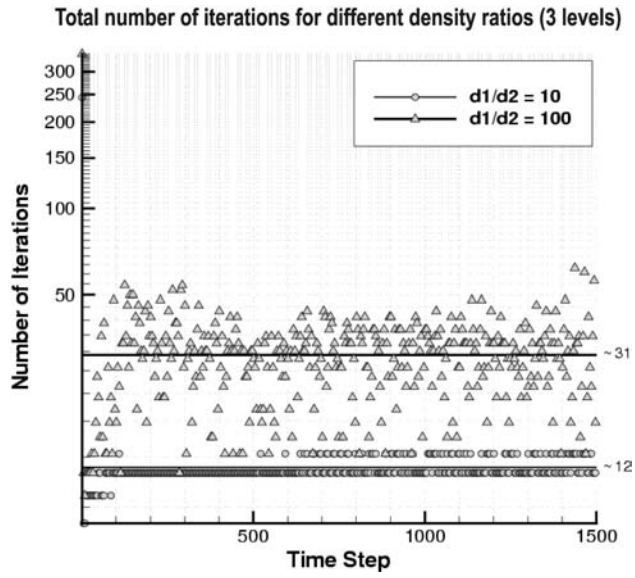
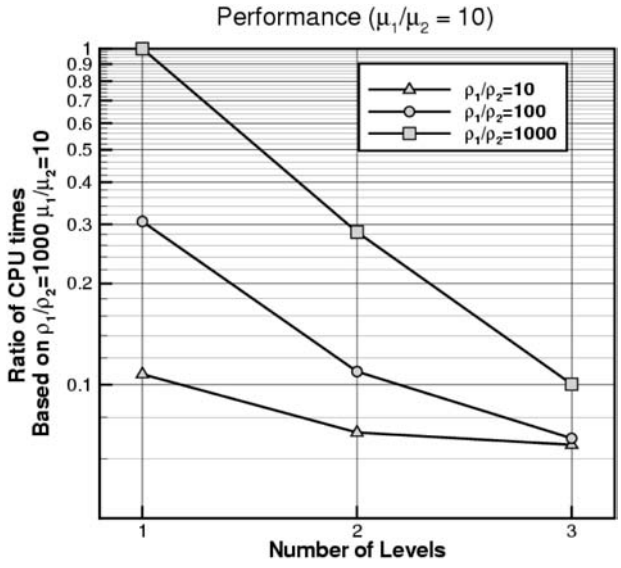


Figure 11.
Total number of fine grid iterations for density ratios of 10 and 100 with a three-level multigrid method for the rising bubble case

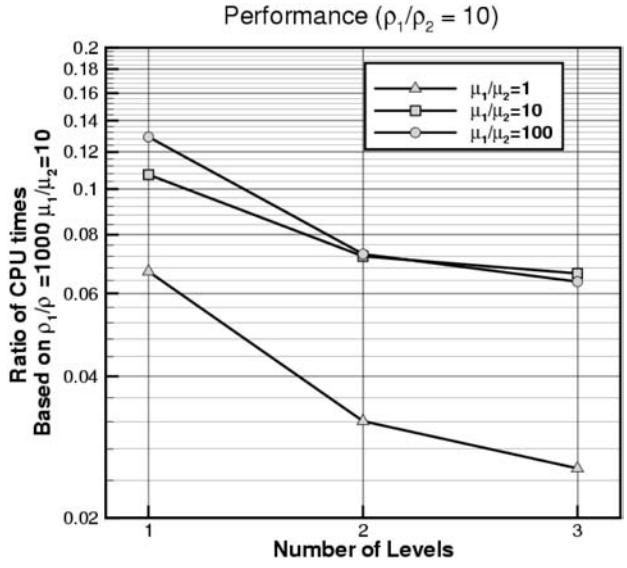
The domain is a cylinder with a diameter of $D = 5d$ and a height of $H = 5d$, and in all computations, the grid system is 152×82 for all cases, distributed non-uniformly.

Figure 16 shows snapshots from the simulation to illustrate the shape deformation of the droplet after the impact under the given conditions (Francois, 2002).



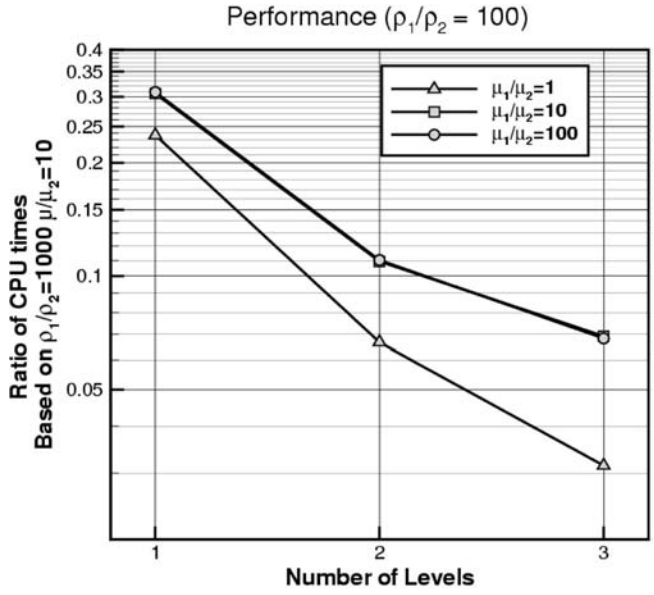
Note: The ratios are with respect to the slowest case (density ratio = 1000 and viscosity ratio = 10)

Figure 12.
Effect of viscosity on performance for the rising bubble case



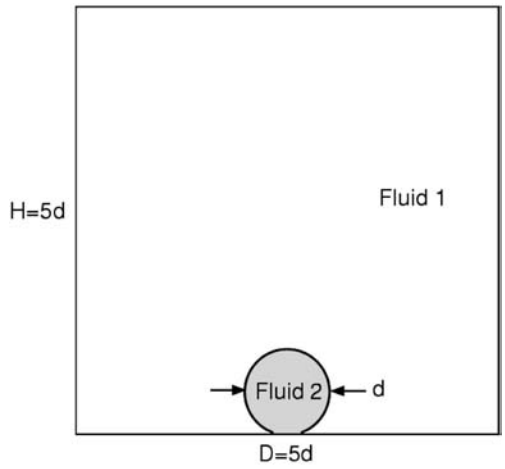
Note: The ratios are with respect to the slowest case (density ratio = 1000 and viscosity ratio = 10)

Figure 13.
Effect of viscosity on performance for rising bubble case



Note: The ratios are with respect to the slowest case (density ratio = 1000 and viscosity ratio = 10)

Figure 14.
Effect of viscosity on performance for rising bubble case



Note: The droplet is a sphere and initially placed on the central axis

Figure 15.
Schematic of the computational setup for the impact of a droplet

Figure 17 presents the CPU time summary for two density ratios, between the liquid drop and ambient gas, while fixing the viscosity ratio to be ten. Comparing Figures 12 and 17, the drop impact and rising bubble cases exhibit very similar behaviors in terms of the influence of the density ratio and contribution of the multigrid technique.

Summary and conclusion

In this study, we examine the interplay between the multigrid computation of the pressure Poisson equation with the focus on the moving boundary separating two fluids, and the effect of the density and viscosity ratios between phases. Two flow problems, one involving a rising bubble in a liquid medium

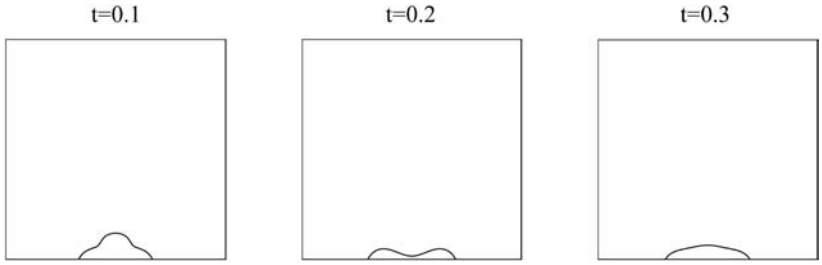
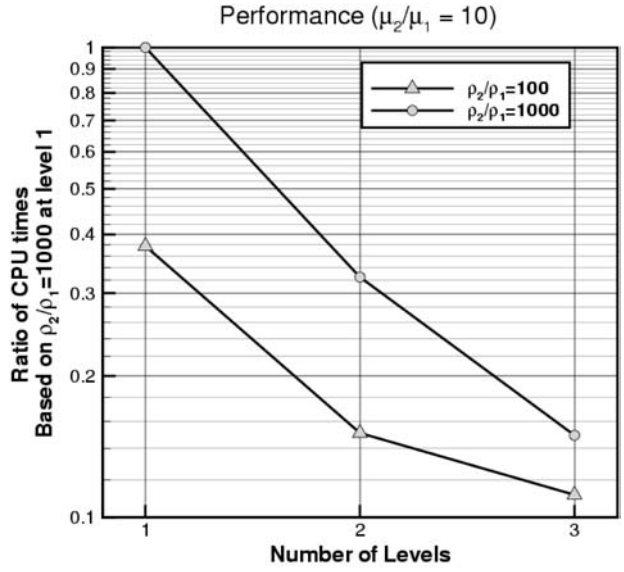


Figure 16.
Impact of a droplet for a density ratio of 100 (Re = 100, We = 4 and Fr = ∞ with a static contact angle of 60°



Note: The ratios are with respect to the slowest case (density ratio = 1000 and viscosity ratio = 10)

Figure 17.
Effect of density on performance for the drop impact case

and the other a liquid drop impinging on a solid surface, are adopted. It is demonstrated that as the density ratio increases, the single grid computation requires an enormous amount of CPU time because of the increase in stiffness of the system. For the grid size and flow problems considered, an increase of factor ten in density ratio results in approximately a three-fold increase in CPU time. The multigrid computation substantially improves convergence rate independent of the density ratio value. Comparing Figures 12 and 17, the impinging drop case and rising bubble cases exhibit similar behaviors in terms of the influence of the density ratio and contribution of the multigrid technique. Finally, the effect of viscosity ratio on the convergence rate of the pressure Poisson equation does not play a major role.

References

Brandt, A. (1977), "Multilevel adaptive solutions to boundary-value problems", *Mathematics of Computation*, Vol. 31, pp. 333-90.

Briggs, W. (2000), *A Multigrid Tutorial*, SIAM, Philadelphia.

Chorin, A.J. (1968), "Numerical solution of the Navier Stokes equations", *Mathematics of Computation*, Vol. 22, pp. 745-62.

Francois, M. (2002), "Computations of drop dynamics with heat transfer", PhD thesis, University of Florida.

Francois, M. and Shyy, W. (2002), "Numerical simulation of droplet dynamics with heat transfer", in Taine, J. (Ed.), *Proceedings of the 12th International Heat Transfer Conference*, 18-23 August, Elsevier, Amsterdam, Grenoble, France, pp. 401-6.

Ghia, U., Ghia, K.N. and Shin, C.T. (1982), "High resolution for incompressible flow using the Navier-Stokes equations and a multigrid Method", *Journal of Computational Physics*, Vol. 48, pp. 387-411.

Hackbusch, W. (1980), "Convergence of multigrid iterations applied to difference equations", *Mathematics of Computation*, Vol. 34, pp. 425-40.

Luchini, P. and Dalascio, A. (1994), "Multigrid pressure-correction techniques for the computation of quasi-incompressible internal flows", *International Journal for Numerical Methods in Fluids*, Vol. 18, pp. 489-507.

McCormick, S.F. (1987), *Multigrid Methods*, SIAM, Philadelphia.

Peskin, C.S. (1977), "Numerical analysis of blood flow in the heart", *Journal of Computational Physics*, Vol. 25, pp. 220-52.

Peskin, C.S. (2002), "The immersed boundary method", *Acta Numerica*, Cambridge University Press, Cambridge, UK, pp. 1-39.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992), *Numerical Recipes in Fortran, The Art of Scientific Computing*, 2nd ed., Cambridge University Press, London.

Shyy, W. (1994), *Computational Modeling for Fluid Flow and Interfacial Transport*, Elsevier, Amsterdam, The Netherlands.

Shyy, W., Francois, M., Udaykumar, H.S., N'Dri, N. and Tran-Son-Tay, R. (2001), "Moving boundaries in micro-scale biofluid dynamics", *Applied Mechanics Reviews*, Vol. 54, pp. 419-53.

Shyy, W., Udaykumar, H.S., Rao, M.M. and Smith, R.W. (1996), *Computational Fluid Dynamics with Moving Boundaries*, Taylor and Francis, Washington, DC.

Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S. and Jan, Y.J. (2001), "A front tracking method for the computations of multiphase flow", *Journal of Computational Physics*, Vol. 169, pp. 708-59.

Udaykumar, H.S., Kan, H.C., Shyy, W. and Tran-Son-Tay, R. (1997), "Multiphase dynamics in arbitrary geometries on fixed Cartesian grids", *Journal of Computational Physics*, Vol. 137, pp. 366-405.

Udaykumar, H.S., Mittal, R., Rampunggoon, P. and Khanna, K. (2001), "A sharp interface Cartesian grid method for simulating flows with complex moving boundaries", *Journal of Computational Physics*, Vol. 174, pp. 345-80.

Ye, T., Mittal, R. and Shyy, W. (1999), "An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries", *Journal of Computational Physics*, Vol. 156, pp. 209-40.