AIAA Journal
Vol. 43, No. 9, September 2005

Hybrid Compressible–Incompressible Numerical Method for Transient Drop-Gas Flows

Amrita R. Wadhwa[*] and John Abraham[†]
Purdue University, West Lafayette, Indiana 47907
and
Vinicio Magi[‡]
University of Basilicata, 3-85100 Potenza, Italy

A numerical method for drops in which the liquid phase is modeled as an incompressible fluid while the gas phase is modeled as a compressible fluid is presented. An unstructured grid that conforms to the deforming interface at all instants in time is employed. Finite volume discretization of the strong conservative form of the governing equations is solved with an implicit iterative procedure that simultaneously solves governing equations for the two phases and the boundary conditions at the interface. The numerical accuracy of the code is assessed by comparing the computed transient drag of a decelerating solid sphere with prior computed results published in the literature and comparing the computed behavior of an oscillating drop with the classical solution of the problem. The numerical method is assessed in axisymmetric computations of deforming and decelerating liquid drops. The transient drag of the drop is compared with computed results for a solid sphere.

Nomenclature

$\boldsymbol{A}$	=	normal Jacobian matrix
a	=	eigenvalues of normal Jacobian matrix
b	=	coefficients in Legendre polynomial fit for interface curvature
c	=	speed of sound
e	=	total specific energy
$\hat{e}$	=	unit vector along the coordinate direction
$\boldsymbol{F}_c$	=	column matrix of convective fluxes
$\boldsymbol{F}_v$	=	column matrix of viscous fluxes
$\boldsymbol{G}$	=	represents certain terms in Eq. (21)
$\underline{h}^0$	=	total specific enthalpy
$\vec{\vec{I}}$	=	identity tensor
K_{ij}	=	spring stiffness associated with the edge joining vertices i and j
k	=	thermal conductivity
l	=	normalized curve length along the interface
$\hat{n}$	=	unit normal vector
Pr	=	Prandtl number
p	=	pressure
Q	=	source term in grid movement
$\boldsymbol{q}$	=	heat-flux vector
R	=	gas constant
$\boldsymbol{R}$	=	right eigenvectors for normal Jacobian of the inviscid governing equations
r	=	radial coordinate
$\boldsymbol{r}$	=	position vector
r_j	=	distance from vertex to centroid of triangle j
S	=	surface area of control volume
T	=	temperature
t	=	time
t_c	=	convective time
t_v	=	viscous time
t^*	=	pseudotime
$\hat{t}$	=	unit tangential vector
$\tilde{u}$	=	specific internal energy
$\boldsymbol{V}$	=	fluid velocity vector
$\boldsymbol{V}_{\text{drop}}$	=	mass-averaged velocity of drop
$\boldsymbol{V}_{\text{int}}$	=	interface velocity vector
$\boldsymbol{V}_s$	=	velocity vector of the moving surface bounding a control volume
$\boldsymbol{V}_{\text{solid}}$	=	velocity of moving solid
v	=	fluid velocity component
$\boldsymbol{W}$	=	column matrix of unknown variables
z	=	axial coordinate
β	=	artificial compressibility factor
γ	=	ratio of specific heats
Δ	=	space or time discretized value of the variable succeeding it
δ	=	change in axial coordinate from one time step to another
ε	=	small positive parameter in total-variation-diminishing (TVD) scheme
θ	=	azimuthal coordinate
κ	=	curvature
$\kappa_{2\text{d}}$	=	curvature in two-dimensional plane
μ	=	dynamic viscosity
ν	=	volume of control volume
ρ	=	density
ρ^*	=	artificial density
$\underline{\sigma}$	=	surface tension
$\vec{\vec{\tau}}$	=	viscous stress tensor
Φ	=	column matrix of numerical antidiffusive fluxes
Ψ	=	entropy correction in TVD scheme
Ω	=	coefficient in Runge–Kutta scheme

Subscripts

g	=	gas phase
gas	=	freestream, that is, gas phase at the start of computation
l	=	liquid phase
r	=	radial coordinate direction
z	=	axial coordinate direction
θ	=	azimuthal coordinate direction

Superscript

n	=	time level

Received 2 September 2004; revision received 24 February 2005; accepted for publication 25 March 2005. Copyright © 2005 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/05 $10.00 in correspondence with the CCC.

[*] Ph.D. Student, Maurice J. Zucrow Laboratories, School of Mechanical Engineering.

[†] Professor, School of Mechanical Engineering, 500 Allison Road. Senior Member AIAA.

[‡] Professor, Department of Environmental Engineering and Physics.

I. Introduction

In liquid injection, the intact liquid core breaks up into drops within several diameters downstream of the orifice. These drops then undergo a variety of interactions among themselves and with the ambient gas. The interactions between the drops result in several outcomes as discussed in the literature.[1,2] Numerical studies of these drop-drop interactions have been reported,[3–6] but much remains to be done. The interactions with the ambient gas result in momentum transfer between the drop and the gas and in drop deformation and secondary breakup. These interactions are also not well understood. In fact, in multidimensional spray models[7,8] the momentum transfer between the drop and the gas is estimated by employing drag correlations, which have been developed for solid spheres in steady flows. Recent work by Kim et al.[9] has shown that these drag coefficients are inadequate in transient flows even for solid spheres. Furthermore, it has been reported in the literature that steady drag coefficients for deforming spheres are different from those for solid spheres.[10–12] There have not been many studies of transient drag coefficients for drops though quasi-steady drag has been reported.[13] The experimental work of Temkin and Mehta[14] indicates that the unsteady drag for decelerating flows is always larger than the steady drag for a solid sphere at the same Reynolds number, whereas in accelerating flows the unsteady drag is observed to be smaller than the corresponding steady drag of a solid at the same Reynolds number. To numerically study the transient drag for drops, a numerical method is developed in this work.

In the next few sections, the governing equations for the two fluids, the physical model for the interface, the space and time discretizations employed in the numerical method along with the grid movement strategies are discussed. The computed transient drag for a decelerating solid sphere is then compared with the prior work of Kim et al.[9] This is followed by a comparison of the analytical solution[15] and the computed results for an oscillating liquid drop. The initial results from an axisymmetric computation of a decelerating liquid drop are presented, and the transient drag is compared with the drag of a decelerating solid sphere. The paper closes with summary and conclusions.

II. Model

A. Governing Equations

The physical problem is that of a liquid drop surrounded by gas. The liquid drop is modeled as an incompressible fluid, while the gas phase is modeled as a compressible fluid. The two immiscible fluids are separated by an interface, which has constant surface tension. The governing equations employed are the standard Navier–Stokes equations for compressible fluid in the gas phase and for incompressible fluid in the liquid phase.[16]

B. Numerical Model

The introduction of a pseudotime t^* in the unsteady governing equations of the compressible flow leads to a reformulation of the system of equations such that a time-marching steady-state solver can be employed.[17] The pseudotime term has the same form as the physical time term. The converged solution in pseudotime represents the transient solution of the governing equations in physical time. Because physical time is fixed when marching in pseudotime, it is possible to employ standard convergence acceleration techniques like local time stepping for steady-state solution in pseudotime. This does not compromise the solution accuracy in physical time. Hence, the unsteady integral form of the governing equations for the compressible gas phase is

$$\frac{\partial}{\partial t^*}\iiint\limits_{\nu(t)} \boldsymbol{W}_g \,\mathrm{d}\nu + \frac{\partial}{\partial t}\iiint\limits_{\nu(t)} \boldsymbol{W}_g \,\mathrm{d}\nu + \iint\limits_{S(t)} (\boldsymbol{F}_c - \boldsymbol{F}_v)_g \bullet \hat{n}\,\mathrm{d}S = 0 \tag{1}$$

where

$$\boldsymbol{W}_g = \begin{pmatrix} \rho_g \\ \rho_g \boldsymbol{V}_g \\ \rho_g e_g \end{pmatrix}, \qquad (\boldsymbol{F}_c)_g = \boldsymbol{W}_g\big(\boldsymbol{V}_g - \boldsymbol{V}_s\big) + p_g \begin{pmatrix} 0 \\ \vec{\vec{I}} \\ \boldsymbol{V}_g \end{pmatrix}$$

$$(\boldsymbol{F}_v)_g = \begin{pmatrix} 0 \\ \vec{\vec{\tau}}_g \\ -\boldsymbol{q}_g + (\vec{\vec{\tau}}_g \bullet \boldsymbol{V}_g) \end{pmatrix}$$

t is the physical time, $S(t)$ is the surface area bounding the control volume $\nu(t)$ at time t, $\hat{n}$ is the unit normal vector pointing outward from the control volume $\nu(t)$, $\boldsymbol{V}_s$ is the velocity vector of the moving surface $S(t)$, ρ_g is the density, $\boldsymbol{V}_g$ is the velocity, e_g is the total specific energy, p_g is the pressure, and $\boldsymbol{q}_g$ is the heat-flux vector of the gas phase in the control volume. The heat-flux vector $\boldsymbol{q}_g$ and the total specific energy e_g are defined as

$$\boldsymbol{q}_g = -k_g \nabla T_g, \qquad e_g = \tilde{u}_g + (\boldsymbol{V}_g \bullet \boldsymbol{V}_g)/2 \tag{2}$$

where k_g is the thermal conductivity, T_g is the temperature, and $\tilde{u}_g$ is the specific internal energy of the gas phase. The set of conservation equations [Eq. (1)] is closed by employing the ideal-gas equation of state:

$$p_g = \rho_g R T_g \tag{3}$$

A direct extension of the time-marching methods for compressible flows to incompressible flows is hindered because of the lack of a time evolutionary term in the mass conservation equation. To overcome this problem, Chorin[18] developed an artificial compressibility method for steady incompressible flows. This involved modification of the continuity equation to include a pseudotime term that vanishes when steady-state solution is reached. The addition of this term results in a mixed set of hyperbolic-parabolic Navier–Stokes equations, which can be solved using a standard time-dependent approach. To obtain a time-accurate time-marching scheme for unsteady incompressible flows, the artificial compressibility approach can be extended to the momentum and energy conservation equations. This is achieved by modifying the governing equations to include an unsteady term in pseudotime.[19] A solution of the modified equations that is steady in pseudotime is identical to the instantaneous unsteady solution of the governing equations in physical time. With the objective of employing the same numerical method for both the compressible gas phase and the incompressible liquid phase, the artificial compressibility approach has been employed in this work for the incompressible liquid phase. Hence, the unsteady integral form of the governing equations employed for the incompressible liquid phase are

$$\frac{\partial}{\partial t^*}\iiint\limits_{\nu(t)} \boldsymbol{W}_l^* \,\mathrm{d}\nu + \frac{\partial}{\partial t}\iiint\limits_{\nu(t)} \boldsymbol{W}_l \,\mathrm{d}\nu + \iint\limits_{S(t)} (\boldsymbol{F}_c - \boldsymbol{F}_v)_l \bullet \hat{n}\,\mathrm{d}S = 0 \tag{4}$$

where

$$\boldsymbol{W}_l^* = \begin{pmatrix} \rho_l^* \\ \rho_l \boldsymbol{V}_l \\ \rho_l e_l \end{pmatrix}, \qquad \boldsymbol{W}_l = \rho_l \begin{pmatrix} 0 \\ \boldsymbol{V}_l \\ e_l \end{pmatrix}$$

$$(\boldsymbol{F}_c)_l = \rho_l \begin{pmatrix} \boldsymbol{V}_l \\ \boldsymbol{V}_l(\boldsymbol{V}_l - \boldsymbol{V}_s) \\ e_l(\boldsymbol{V}_l - \boldsymbol{V}_s) \end{pmatrix} + p_l \begin{pmatrix} 0 \\ \vec{\vec{I}} \\ \boldsymbol{V}_l \end{pmatrix}$$

$$(\boldsymbol{F}_v)_l = \begin{pmatrix} 0 \\ \vec{\vec{\tau}}_l \\ -\boldsymbol{q}_l + (\vec{\vec{\tau}}_l \bullet \boldsymbol{V}_l) \end{pmatrix}$$

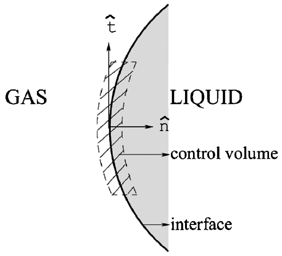


Fig. 1 Control volume employed to derive the interface conditions.

where ρ_l is the density of the liquid phase and is assumed to be constant because of the incompressible fluid assumption, $\boldsymbol{V}_l$ is the velocity, e_l is the total specific energy, p_l is the pressure, $\vec{\vec{\tau}}_l$ is the viscous stress tensor, and $\boldsymbol{q}_l$ is the heat-flux vector of the liquid phase in the control volume. The heat-flux vector $\boldsymbol{q}_l$ and the total specific energy e_l for the liquid phase are defined in a manner similar to that of the gas phase [Eq. (2)]. In the artificial compressibility method, the artificial density ρ_l^* is related to the pressure p_l by the artificial equation of state

$$p_l = \beta \rho_l^* \tag{5}$$

It is representative of an artificial speed of sound. The optimum value of β depends on the problem being solved.

C. Physical Modeling of Interface

Figure 1 shows the control volume employed in the derivation of the interface conditions. The assumptions made for the control volume analysis are as follows:

1) Thickness of control volume is negligible at the interface.

2) Mass within the interface is negligible.

3) Velocity at the interface is subjected to the no-slip condition.

These assumptions, when applied to the control volume considered in Fig. 1, result in the following transport equations.

Mass:

$$\rho_l[(\boldsymbol{V}_l - \boldsymbol{V}_{\text{int}}) \bullet \hat{n}] = \rho_g[(\boldsymbol{V}_g - \boldsymbol{V}_{\text{int}}) \bullet \hat{n}] \tag{6}$$

Because the two fluids on either side of the interface are assumed to be immiscible, the mass-transfer rate per unit area in Eq. (6) is set to zero. This results in the following equation for mass conservation at the interface:

$$\boldsymbol{V}_l = \boldsymbol{V}_g = \boldsymbol{V}_{\text{int}} \tag{7}$$

Momentum normal to the interface:

$$p_l - (\hat{n} \bullet \vec{\vec{\tau}}_l) \bullet \hat{n} = p_g - (\hat{n} \bullet \vec{\vec{\tau}}_g) \bullet \hat{n} + \sigma\kappa \tag{8}$$

Momentum tangential to the interface:

$$(\hat{n} \bullet \vec{\vec{\tau}}_l) \bullet \hat{t} = (\hat{n} \bullet \vec{\vec{\tau}}_g) \bullet \hat{t} \tag{9}$$

Energy:

$$\begin{aligned} p_l(\boldsymbol{V}_l \bullet \hat{n}) - \hat{n} \bullet (\vec{\vec{\tau}}_l \bullet \boldsymbol{V}_l) + \boldsymbol{q}_l \bullet \hat{n} &= p_g(\boldsymbol{V}_g \bullet \hat{n}) \\ &- \hat{n} \bullet (\vec{\vec{\tau}}_g \bullet \boldsymbol{V}_g) + \boldsymbol{q}_g \bullet \hat{n} + \sigma\kappa(\boldsymbol{V}_{\text{int}} \bullet \hat{n}) \end{aligned} \tag{10}$$

D. Space Discretization

An unstructured grid of triangular elements is generated separately for each phase with the interface as a common boundary. This allows the unstructured grid to conform to the irregular shape of the interface at all times. The grid generator employed in this work, delaundo,[20] creates triangular grids based on the frontal Delaunay method.

A finite volume approach with a cell-centered scheme is chosen to solve the governing equations of the two phases. Figure 2 shows a triangular control volume, which will be employed to illustrate the details of the numerical discretization. Also, for illustration, the governing equations of the gas phase, that is, Eq. (1), will be employed.

The integration of the governing equations for the triangular control volume in Fig. 2 yields

$$\frac{\Delta[(\boldsymbol{W}_g)_{j0}\Delta v_{j0}]}{\Delta t^*} + \frac{\Delta[(\boldsymbol{W}_g)_{j0}\Delta v_{j0}]}{\Delta t} + \sum_i (\boldsymbol{F}_c - \boldsymbol{F}_v)_i \bullet \hat{n}_i \Delta S_i = 0 \tag{11}$$

where Δv_{j0} denotes the volume of the triangular cell centered at $j0$ in Fig. 2. $(\boldsymbol{W}_g)_{j0}$ represents $\boldsymbol{W}_g$ estimated using the cell-centered values of $j0$. The summation is carried out over the three boundaries — $i1$, $i2$, and $i3$— of triangle $j0$. ΔS_i denotes the surface area of boundary i of $j0$, and $\hat{n}_i$ is the unit normal vector to boundary i, which is directed outward from the control volume $j0$. $(\boldsymbol{F}_c)_i$ and $(\boldsymbol{F}_v)_i$ denote the convective and viscous fluxes transported across the boundary i into the control volume $j0$.

For the estimation of $(\boldsymbol{F}_v)_i$, the values of $\boldsymbol{V}_g$, $\vec{\vec{\tau}}_g$, and $\boldsymbol{q}_g$ are required at the boundary surface S_i. These values are to be determined from the cell-centered values of the control volumes. $\boldsymbol{V}_g$ is defined by interpolating the velocities in triangle $j0$ and the neighboring triangle, which leads to a second-order-accurate solution. This method is also employed to define the fluid properties like dynamic viscosity and thermal conductivity, which are required in the estimation of $\vec{\vec{\tau}}_g$ and $\boldsymbol{q}_g$, respectively. The gradients of velocity and temperature in $\vec{\vec{\tau}}_g$ and $\boldsymbol{q}_g$ are defined by applying the Gauss divergence theorem to the two triangles, which are on either side of the boundary surface S_i. This method of numerical flux estimation leads to a second-order central difference approximation.

The convective fluxes are estimated by employing a total-variation-diminishing (TVD) scheme.[21] This scheme when applied to the boundary surface $i1$ results in the following:

$$(\boldsymbol{F}_c \bullet \hat{n})_{i1} = \tfrac{1}{2}\{[(\boldsymbol{F}_c)_{j0} + (\boldsymbol{F}_c)_{j1}] \bullet \hat{n}_{i1} + \boldsymbol{R}_{i1}\Phi_{i1}\} \tag{12}$$

where $(\boldsymbol{F}_c)_{j0}$ and $(\boldsymbol{F}_c)_{j1}$ represent $\boldsymbol{F}_c$ estimated using cell-centered variable values of triangle $j0$ and $j1$, respectively. It is the matrix of eigenvectors of $\boldsymbol{A}$ that diagonalizes $\boldsymbol{A}$, that is, $\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{R}$ is a diagonal matrix consisting of eigenvalues of $\boldsymbol{A}$. The normal Jacobian matrix for a boundary surface $i1$ is defined as

$$\boldsymbol{A}_{i1} = \left(\frac{\partial \boldsymbol{F}_c}{\partial \boldsymbol{W}_g} \bullet \hat{n}\right)_{i1} \tag{13}$$

Both the matrices, $\boldsymbol{A}$ and $\boldsymbol{R}$, are functions of space and time as a result of the nonlinear nature of the governing equations. For axisymmetric flows in cylindrical coordinates (r, θ, z),

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ v_z & -n_r & v_z + cn_z & v_z - cn_z & 0 \\ v_r & n_z & v_r + cn_r & v_r - cn_r & 0 \\ v_\theta & 0 & v_\theta & v_\theta & 1 \\ \left(v_r^2 + v_\theta^2 + v_z^2\right)/2 & v_r n_z - v_z n_r & h^0 + c(v_r n_r + v_z n_z) & h^0 - c(v_r n_r + v_z n_z) & v_\theta \end{bmatrix} \tag{14}$$

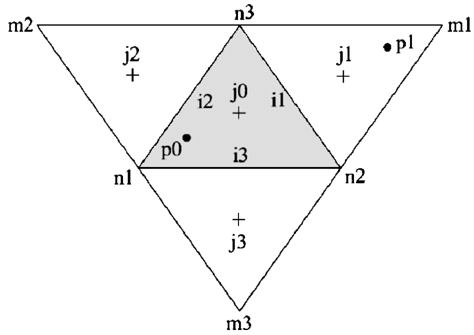


Fig. 2 Triangular control volume employed to illustrate the details of the numerical discretization.

where (v_r, v_θ, v_z) are the fluid velocity components along the (r, θ, z) coordinate directions respectively; $(n_r, 0, n_z)$ represents the unit normal vector $\hat{n}$ in the cylindrical coordinate system for an axisymmetric flow; and h^0 is the total specific enthalpy and is defined as $h^0 = e/p$.

Linearization is implemented by employing the local characteristic approach.[21] In this approach, the matrix $\boldsymbol{R}$ is evaluated using local variables of the computed flowfield. For the implementation of TVD scheme, $\boldsymbol{R}$ is to be estimated at the boundary surface S_{i1}. The computed local variables at the boundary are obtained by Roe averaging the cell-centered variables of the two triangles, which are present on either side of the boundary surface.

In Eq. (12), Φ_{i1} is the numerical antidiffusive flux that makes the scheme upwind-biased TVD and second-order accurate. Each lth element of the column matrix Φ_{i1} is

$$\Phi^l_{i1} = \tfrac{1}{2}\Psi\left(a^l_{i1}\right)\left(g^l_{j0} + g^l_{j1}\right) - \Psi\left(a^l_{i1} + \eta^l_{i1}\right)\alpha^l_{i1} \tag{15}$$

where

$$\Psi(x) = \begin{cases} |x|, & |x| \geq \varepsilon \\ (x^2 + \varepsilon^2)/2\varepsilon, & |x| < \varepsilon \end{cases}$$

$$\eta^l_{i1} = \frac{1}{2}\Psi\left(a^l_{i1}\right) \begin{cases} \left(g^l_{j1} - g^l_{j0}\right)/\alpha^l_{i1}, & \alpha^l_{i1} \neq 0 \\ 0, & \alpha^l_{i1} = 0 \end{cases} \tag{16}$$

$$g^l_{j0} = \text{minmod}\left(\alpha^l_{i1}, \alpha^l_{p0}\right)$$

$$g^l_{j1} = \text{minmod}\left(\alpha^l_{i1}, \alpha^l_{p1}\right)$$

where $\Psi(x)$ is an entropy correction to $|x|$. The larger the value of ε, the more numerically dissipative the scheme becomes. Here a_{i1} represents the eigenvalues of the normal Jacobian A_{i1}, defined as

$$\boldsymbol{A}_{i1} = \left(\frac{\partial(\boldsymbol{F}_c)_g}{\partial \boldsymbol{W}_g} \bullet \hat{n}\right)_{i1} \tag{17}$$

α^l_{i1}, α^l_{p0}, and α^l_{p1} represent the lth element, respectively, of the following column matrices:

$$\boldsymbol{\alpha}_{i1} = \boldsymbol{R}^{-1}_{i1}\left[(\boldsymbol{W}_g)_{j1} - (\boldsymbol{W}_g)_{j0}\right]$$

$$\boldsymbol{\alpha}_{p0} = \boldsymbol{R}^{-1}_{i1}\left[(\boldsymbol{W}_g)_{j0} - (\boldsymbol{W}_g)_{n1}\right]$$

$$\boldsymbol{\alpha}_{p1} = \boldsymbol{R}^{-1}_{i1}\left[(\boldsymbol{W}_g)_{m1} - (\boldsymbol{W}_g)_{j1}\right] \tag{18}$$

where $(\boldsymbol{W}_g)_k$ is estimated using values of variables at the spatial position k in the computational domain. The values of variables at a vertex are evaluated by mass averaging the cell-centered variables of triangles around the vertex. In Eq. (18), $p0$ and $p1$ denote the spatial positions shown in Fig. 2.

The scheme just outlined applies for both the cases with and without any grid movement. For grid movement, an additional term needs to be added to Eq. (12), which is estimated as follows:

$$(\boldsymbol{W}_g)_{i1}(\boldsymbol{V}_s \bullet \hat{n})_{i1} = \tfrac{1}{2}\left[(\boldsymbol{W}_g)_{j0} + (\boldsymbol{W}_g)_{j1}\right](\boldsymbol{V}_s \bullet \hat{n})_{i1} \tag{19}$$

The space discretization of the governing equations for the liquid phase is similar to that of the gas phase. To solve the governing equations of the two phases, the fluxes across the interface are to be estimated such that the interface conditions, that is, Eqs. (7–10), are satisfied. The implementation of these interface conditions include estimation of the viscous stresses and heat flux across the interface. For a control volume formulation, these are estimated at the centroids of the gas and liquid control volumes on either side of the interface and cannot be determined at the interface without a viscous flux model. In this work, the viscous fluxes are assumed to be continuous at the interface, that is, viscous stresses and heat flux for the two phases are assumed to be equal at the interface. The impact of this model on computed results is evaluated by showing that there is adequate agreement between the computed oscillation frequency of a liquid drop and its analytical solution (see Sec. III). The continuous viscous flux assumption implies that the change in normal momentum and energy fluxes across the interface is caused by a pressure differential caused by surface tension forces [Eqs. (8) and (10)]. As a result, the continuous viscous flux model can be employed for drop-gas flows where the pressure drag is dominant because the shearing effect of the gas phase on the interface cannot be captured. Because the viscous fluxes are assumed to be continuous at the interface, they are estimated using gas-phase variables. This approach is adopted so that the shear stress component of the transient drag computed for a liquid drop is consistent with the drag estimation method for a solid sphere.

The interface boundary condition, Eq. (7), states that the interface moves with fluid velocity, that is, $\boldsymbol{V}_s = \boldsymbol{V}_g = \boldsymbol{V}_l = \boldsymbol{V}_{\text{int}}$. This implies that the convective flux $\boldsymbol{F}_c$ at the interface is caused by pressure alone. Assuming boundary surface $i1$ in Fig. 2 represents an interface boundary element, the convective flux is estimated as follows:

$$(\boldsymbol{F}_c \bullet \hat{n})_{i1} = \frac{1}{2}[(\boldsymbol{F}_c)_{j0} + (\boldsymbol{F}_c)_{j1}] \bullet \hat{n}_{i1} \pm \frac{1}{2}\sigma\kappa \begin{pmatrix} 0 \\ \overset{\Rightarrow}{I} \\ \boldsymbol{V}_{\text{int}} \end{pmatrix}_{i1} \bullet \hat{n}_{i1} \tag{20}$$

which leads to a second-order central difference scheme at the interface boundary. In Eq. (20), $j0$ and $j1$ represent triangles on either side of the interface boundary element, one of which is in the gas phase and the other in the liquid phase. The flux caused by surface tension in Eq. (20) has a positive value for the control volume in the liquid phase and a negative value for the gas phase. This numerical treatment of the flux at the interface ensures that the jump in normal momentum flux is $\sigma\kappa$, and for the energy equation it is $\sigma\kappa(\boldsymbol{V}_{\text{int}} \bullet \hat{n})$.

The interface curvature κ in the axisymmetric computations is defined as follows:

$$\kappa = \kappa_{2\text{d}} - (\hat{n} \bullet \hat{e}_r)/|\boldsymbol{r} \bullet \hat{e}_r| \tag{21}$$

where $\kappa_{2\text{d}}$ is the curvature in the two-dimensional plane of Fig. 1, $\hat{e}_r$ is a unit vector along the radial coordinate direction, and $\boldsymbol{r}$ and $\hat{n}$ represent the position vector and unit normal vector respectively, both of which are defined at the center of the interface element. The direction of normal vector is illustrated in Fig. 1. Then, $\kappa_{2\text{d}}$ at the center of the interface element is estimated using Legendre polynomials. For example, in the cylindrical coordinate system (r, θ, z) the Legendre polynomials are fitted to the z coordinates of the interface vertices as follows:

$$z = b_1 + b_2 l + b_3[(3l^2 - 1)/2] + b_4[(5l^3 - 3l)/2] \tag{22}$$

where l represents the normalized curve length along the interface to the vertex whose coordinates are (r, z). The four polynomials on the right side of Eq. (22) are the first four Legendre polynomials, and so Eq. (22) is a third-degree polynomial fit for z. The coefficients b_1–b_4 are determined from the z coordinates of four interface nodes.

For example, the two vertices of an interface edge and the adjacent vertices of its neighboring interface edges constitute the four vertices for the polynomial fit. Similarly, Legendre polynomials can be fit to the r coordinates of the same four interface nodes. $\kappa_{2\mathrm{d}}$ is then defined as

$$\kappa_{2D} = \frac{\ddot{r}\,\dot{z} - \dot{r}\,\ddot{z}}{(\dot{r}^2 + \dot{z}^2)^{\frac{3}{2}}} \tag{23}$$

where $\dot{r}$ and $\dot{z}$ represent the first derivatives while $\ddot{r}$ and $\ddot{z}$ denote the second derivatives of r and z, respectively. These derivatives are estimated from the Legendre polynomials. In this method, coordinates of the two vertices of any interface edge are employed in three different sets of Legendre polynomial fits. These three polynomial fits result in three different values of $\kappa_{2\mathrm{d}}$, each of which is estimated by Eq. (23). An average value of these three values is employed for $\kappa_{2\mathrm{d}}$ in Eq. (21). This averaging method has been validated by Tryggvason et al.[22] The averaging method smoothes the discontinuous values of curvature that arise as a result of the numerical representation of the interface as a set of discrete boundary surfaces. Increasing computational grid resolution decreases the discontinuity in the curvature values and also improves the accuracy of the interface conditions implemented at the drop surface.

The boundaries of the computational domain consist of flow boundaries and an axisymmetry boundary. For inflow–outflow boundaries, the flow is assumed locally to be one dimensional along the direction normal to the boundary surface element. The fluxes at the boundary surfaces are estimated using variables that are obtained by solving a local one-dimensional inviscid (LODI) system of equations.[23] Nonreflective boundary conditions are implemented in LODI by specifying amplitudes of the incoming waves to be constant in time.[24] For inflow, the incoming Riemann invariant, entropy, and flow direction are imposed together with the outgoing Riemann invariant. In the case of outflow, the outgoing Riemann invariant is used along with the entropy and the tangential component of velocity. The incoming Riemann invariant for the outflow boundary is corrected such that the imposed boundary conditions are partially nonreflecting.[22] In the case of axisymmetry boundary, the gradients of pressure and temperature are enforced to be zero. The gradient of normal component of velocity is set to be zero, whereas the tangential component of velocity is subjected to the slip condition along the axis. For the solid-sphere computations, wall boundary condition is implemented for the sphere surface. At the wall, no-slip condition is enforced for velocity, pressure is obtained by assuming zero normal gradient, and an adiabatic condition is imposed for temperature.

E. Time Discretization

A second-order time-accurate implicit scheme is used for solving the governing equations of the two phases. For example, Eq. (11) is discretized at physical time level $n+1$ as follows:

$$\frac{\Delta[(\boldsymbol{W}_g)_{j0}\Delta\nu_{j0}]}{(\Delta t^*)_{j0}} + \left[\frac{3(\boldsymbol{W}_g)_{j0}^{n+1}\Delta\nu_{j0}^{n+1}}{2\Delta t^{n+1}} - \frac{4(\boldsymbol{W}_g)_{j0}^{n}\Delta\nu_{j0}^{n}}{2\Delta t^{n+1}} + \frac{(\boldsymbol{W}_g)_{j0}^{n-1}\Delta\nu_{j0}^{n-1}}{2\Delta t^{n+1}}\right] + \left[\sum_i (\boldsymbol{F}_c - \boldsymbol{F}_v)_i \bullet \hat{n}_i \Delta S_i\right]^{n+1} = 0 \tag{24}$$

where the superscripts on each term denote the physical time level at which the term is evaluated. Δt^{n+1} denotes the change in physical time from time level n to $n+1$. The solution at physical time level $n+1$ is obtained by marching Eq. (24) to a steady state in pseudotime. Thus, the unsteady flow calculation is carried out as a series of steady-state calculations in pseudotime. The pseudotime marching is carried out using a four-stage Runge–Kutta scheme as follows:

$$\left[1 + \Omega_m \frac{3(\Delta t^*)_{j0}}{2\Delta t}\right][(\boldsymbol{W}_g)_{j0}\Delta\nu_{j0}]^m = [(\boldsymbol{W}_g)_{j0}\Delta\nu_{j0}]^n - \Omega_m(\Delta t^*)_{j0}\boldsymbol{G}_{j0}^{m-1} \tag{25}$$

where $\Omega_m = 1/(5-m)$ represents the coefficients and m denotes the stage number of the Runge–Kutta scheme, that is, $m = 1, 2, 3, 4$. $(\Delta t^*)_{j0}$ is the local pseudotime step for the control volume $j0$. To increase stability, as shown by Venkatakrishnan and Mavriplis,[25] the term $3(\boldsymbol{W}_g)_{j0}^{n+1}\Delta\nu_{j0}^{n+1}/2\Delta t^{n+1}$ in Eq. (24) is treated implicitly at each stage of the Runge–Kutta scheme, that is, it is considered to be evaluated at stage m in Eq. (25). $\boldsymbol{G}_{j0}$ represents all of the terms in Eq. (24) other than $3(\boldsymbol{W}_g)_{j0}^{n+1}\Delta\nu_{j0}^{n+1}/2\Delta t^{n+1}$ and the pseudotime term. For convergence acceleration, the viscous fluxes in $\boldsymbol{G}_{j0}$ are estimated only during the first two stages of the Runge–Kutta scheme.

The local pseudotime step $(\Delta t^*)_{j0}$ for each control volume $j0$ is defined to be a constant fraction of the local physical time step $(\Delta t)_{j0}$. The value of the constant depends on the physical problem being solved. Both of these local time steps are evaluated at the beginning of each physical time step. The local physical time step is estimated as follows[26]:

$$(\Delta t)_{j0} = \mathrm{CFL}\left[\frac{(\Delta t_c)_{j0}(\Delta t_v)_{j0}}{(\Delta t_c)_{j0} + (\Delta t_v)_{j0}}\right] \tag{26}$$

where CFL is the Courant–Friedrichs–Lewy number. $(\Delta t_c)_{j0}$ and $(\Delta t_v)_{j0}$ denote the convective and viscous time-step limits and are estimated as follows:

$$(\Delta t_c)_{j0} = \frac{\Delta\nu_{j0}}{(\lambda_c)_{j0}}, \qquad (\Delta t_v)_{j0} = \frac{1}{4}\left[\frac{(\Delta\nu_{j0})^2}{(\lambda_v)_{j0}}\right] \tag{27}$$

where

$$(\lambda_c)_{j0} = \sum_i |(\boldsymbol{V}_{g\,\mathrm{or}\,l} - \boldsymbol{V}_s) \bullet \hat{n} + c|_i \Delta S_i$$

$$(\lambda_v)_{j0} = \frac{\gamma_g}{Pr_g}\sum_i \left(\frac{\mu_{g\,\mathrm{or}\,l}}{\rho_{g\,\mathrm{or}\,l}}\right)(\Delta S_i)^2$$

where γ_g is the ratio of specific heats and Pr_g is the Prandtl number of the gas phase. For liquid phase, $(\Delta t_v)_{j0}$ is the same expression except that it does not contain the ratio of (γ_g/Pr_g), and $\mu_{g\,\mathrm{or}\,l}$ represents the dynamic viscosity of the fluid, which could either be the gas phase g or the liquid phase l. Summations in the definition of $(\lambda_c)_{j0}$ and $(\lambda_v)_{j0}$ are defined for boundary surfaces of the control volume $j0$. The contribution of interface boundary surfaces to this summation is set to be zero. For the gas phase, speed of sound c is defined as $\sqrt{(\gamma_g R T_g)}$, where

$$T_g = [(\gamma_g - 1)/R]\big(\rho_g e_g - \tfrac{1}{2}\boldsymbol{V}_g \bullet \boldsymbol{V}_g\big) \tag{28}$$

For the liquid phase, c represents the artificial speed of sound in the artificial compressibility method and is defined as

$$c = \sqrt{[(\boldsymbol{V}_l - 0.5\boldsymbol{V}_s)\bullet\hat{n}]^2 + \beta} \tag{29}$$

F. Grid Movement Strategies

The vertices on the boundary of the computational domain are moved with a constant velocity in the axial direction at each time step. In the case of a solid sphere, this axial velocity is that of the moving sphere, whereas in the case of the liquid drop the axial velocity of center of mass of the moving drop is employed. The movement of the vertices on the interface is governed by Eq. (7). In Eq. (7), the fluid velocity $\boldsymbol{V}_{\mathrm{int}}$ for an interface vertex $n2$ in Fig. 2 can be defined by the following spatial interpolation method:

$$(\boldsymbol{V}_{\mathrm{int}})_{n2} = \left[\frac{\sum_j (1/r_j)(\boldsymbol{V}_g \text{ or } \boldsymbol{V}_l)_j}{\sum_j (1/r_j)}\right]_{n2} \tag{30}$$

where the summation is defined for all of the triangles j surrounding the vertex $n2$. The weighting function $1/rj$ is defined as the inverse of the distance from the vertex $n2$ to the centroid of triangle j. This choice of the weighting function ensures that for larger rj, the contribution of the gas velocity $\boldsymbol{V}_g$ or the liquid velocity $\boldsymbol{V}_l$ of triangle j to $(\boldsymbol{V}_{\mathrm{int}})_{n2}$ is small.

The tension spring analogy[25] has been employed in this work to specify the movement of vertices in the interior domain. The unstructured grid is considered to be analogous to a system of interconnected springs. In a two-dimensional grid, this system is constructed by representing each edge of every triangle by a tension spring. The spring stiffness is assumed to be inversely proportional to the square of the edge length. In this approach, the vertices on boundary and interface are moved as already discussed, and the instantaneous positions of interior vertices are prescribed by preserving the spring stiffness of the grid before deformation. This is achieved by solving a linear system of equations, each equation of which has the following form:

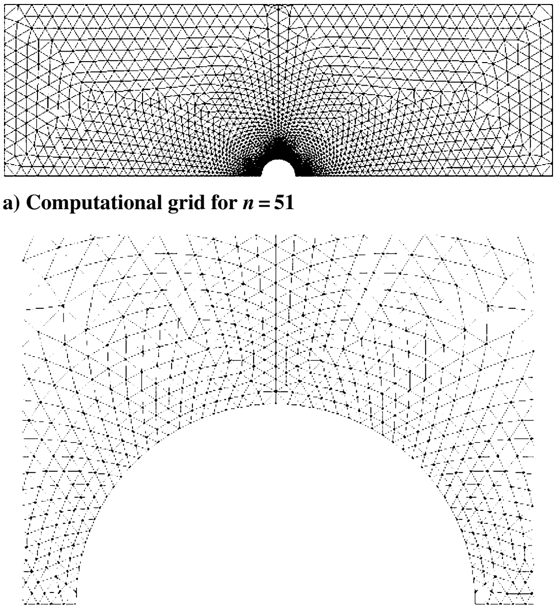
$$\sum_j K_{ij}(z_i - z_j) = Q_i \tag{31}$$

where j is summed over all vertices connected to vertex i. K_{ij} is the spring stiffness associated with the edge joining the vertices i and j. Q_i is a source term that preserves the initial grid in the absence of any interface movement. $K_{ij}(z_i - z_j)$ represents the tension spring force in each edge, which is proportional to the stretched length of the spring. For the unstructured grid, the stretched length would be analogous to the difference in values of one of the coordinates of the grid vertices. Equation (31) represents the linear system of equations obtained when the z coordinate represents the stretched length. Similar to Eq. (31), a linear system of equations is solved for the r coordinates of the vertices. These linear systems of equations in z and r coordinates are solved iteratively for new positions of the interior vertices. Initially, the displacements of the interior vertices are predicted according to a linear extrapolation given by

$$z_i^{n+1} = z_i^n + 2\delta_i^n - \delta_i^{n-1} \tag{32}$$

where $\delta_i^n = z_i^n - z_i^{n-1}$. These predicted values are corrected using a point Gauss–Seidel method until a prescribed convergence is obtained. The resulting values of $z_i^{n+1}$and r_i^{n+1} are then employed to define the vertex velocities, for example, radial component of the vertex velocity at time step $n+1$ is $(r_i^{n+1} - r_i^n)/\Delta t$. These vertex velocities further define the boundary surface velocities $\boldsymbol{V}_s$.

The tension spring analogy preserves connectivity of the grid. However, edges in the grid can cross, resulting in invalid triangulations. This crossing is corrected by checking grid quality at the end of each physical time step. Before grid lines cross, angles of triangles tend to become small. Thus, a prescribed check on the angles of each triangle ensures that the occurrence of grid line crossing is reduced. Furthermore, a check is always carried out for grid line crossing along with the check for angles of triangles. If the prescribed criteria are violated, the grid is regenerated, and a mass-averaged solution is interpolated from the old grid to the new grid. This grid movement strategy ensures that the grid quality is maintained in and around the vicinity of the deforming interface. For example, a triangle near the interface has two vertices on the interface, and one vertex is present either in the gas or liquid phase. The movement of the two interface vertices is defined by Eq. (30) while the single vertex is moved according to the spring tension analogy. These two methods of moving the vertices of a single triangle lead to poor grid quality, that is, the angles in the triangle become small. The prescribed check on the angles of the triangle ensures that the grid is regenerated before the numerical code encounters any convergence problems.



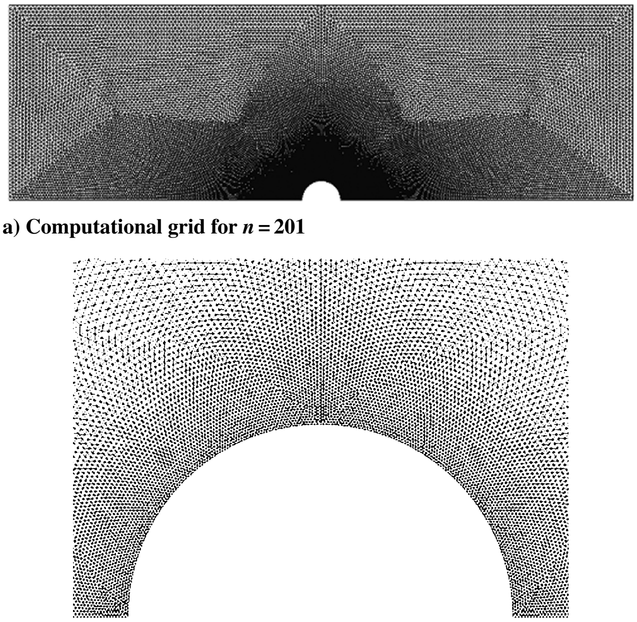
a) Computational grid for $n = 51$

b) Magnified view of the grid around the solid surface for $n = 51$

Fig. 3 Lowest-resolution computational grid for the solid sphere: $n = 51$.

For moving grids, the cell volumes are computed by solving the geometric conservation law (GCL)[27]:

$$\frac{\delta v}{\delta t} = \iint_{S(t)} \boldsymbol{V}_s \bullet \hat{n}\ \mathrm{d}S \tag{33}$$



a) Computational grid for $n = 201$

b) Magnified view of the grid around the solid surface for $n = 201$

Fig. 4 Highest-resolution computational grid for the solid sphere: $n = 201$.

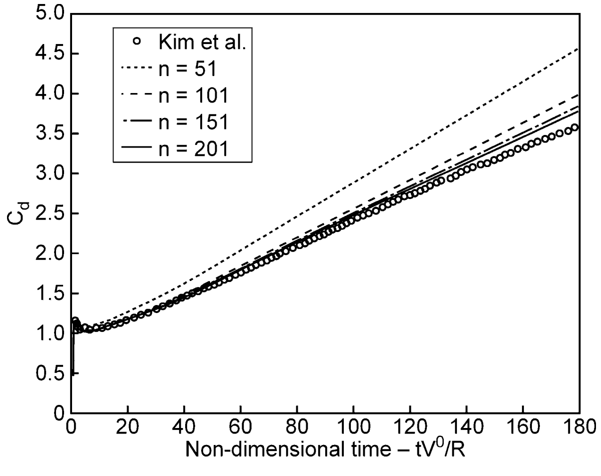
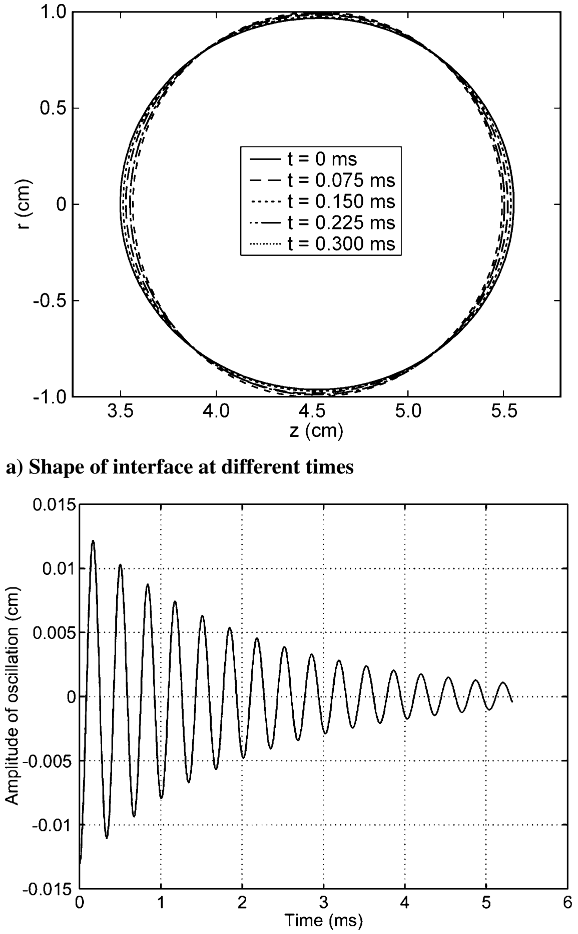


Fig. 5 Comparison of transient drag coefficient for a decelerating solid sphere with the computed data of Kim et al.[9]



a) Shape of interface at different times

b) Amplitude of oscillation as a function of time

Fig. 6 Computed results for an oscillating liquid drop of $\rho_l/\rho_g = 100$.

The GCL is solved numerically using the same scheme that is employed for integrating the governing equations of the two fluids. This numerical procedure is necessary to obtain a cell volume that is consistent with the grid movement.

III. Results and Discussion

The numerical formulation is valid for full three-dimensional simulations with a tetrahedral grid generator. These simulations would require massively parallel computers as the computational effort involved is about two orders of magnitude greater than the two-dimensional axisymmetric computations. In this work, the governing equations are solved in a cylindrical coordinate system (r, θ, z), and the flow is assumed to be axisymmetric. Comparisons of computed results with analytical solutions and prior numerical works are presented in this section along with preliminary results of transient drag coefficients.

Kim et al.[9] solved the Navier–Stokes equations for a solid sphere that moves with an initial velocity in a stagnant incompressible fluid. They presented the transient variation of the drag coefficient C_d for the decelerating solid sphere as a function of nondimensional time t^+, defined as $t^+ = (t\ V^0)/R$, where V^0 is the initial velocity of sphere and R is the sphere radius. C_d for the solid sphere is defined as

$$C_d = \frac{\iint_{S(t)} - p_g \hat{n} \bullet \hat{e}_z \, \mathrm{d}S + \iint_{S(t)} \hat{n} \bullet \vec{\vec{\tau}}_g \bullet \hat{e}_z \, \mathrm{d}S}{\frac{1}{2}\rho_{\text{gas}}(\boldsymbol{V}_{\text{gas}} - \boldsymbol{V}_{\text{solid}}) \bullet (\boldsymbol{V}_{\text{gas}} - \boldsymbol{V}_{\text{solid}})\pi R^2} \tag{34}$$

where $\hat{e}_z$ is a unit vector along the axial coordinate direction, $S(t)$ represents the moving solid surface of the sphere, and ρ_{gas} and $\boldsymbol{V}_{\text{gas}}$

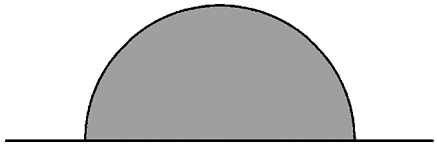
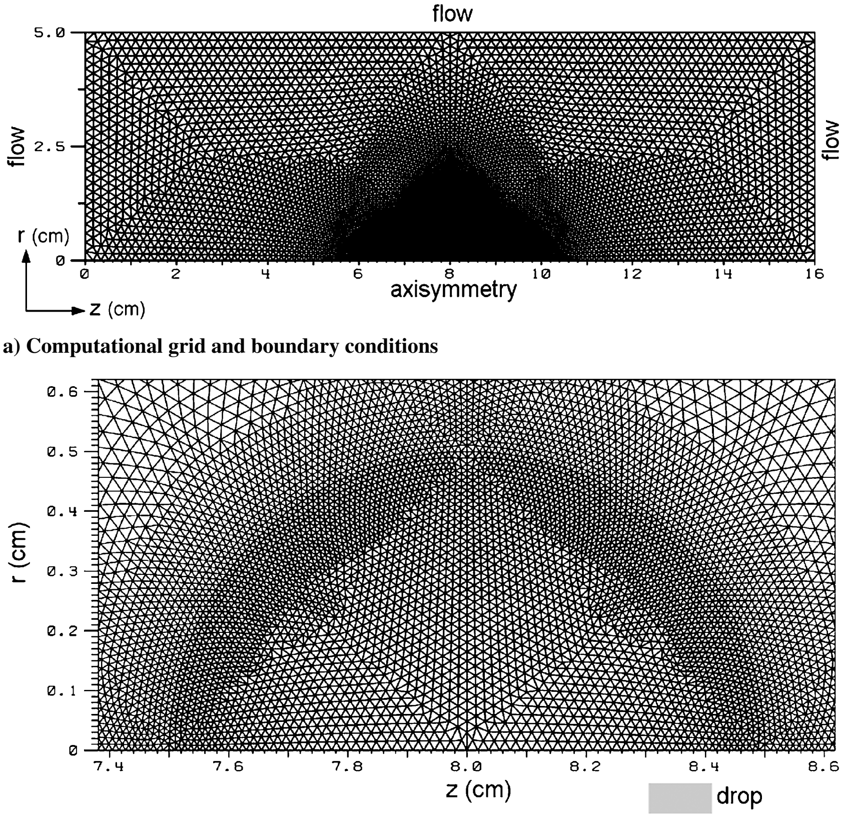


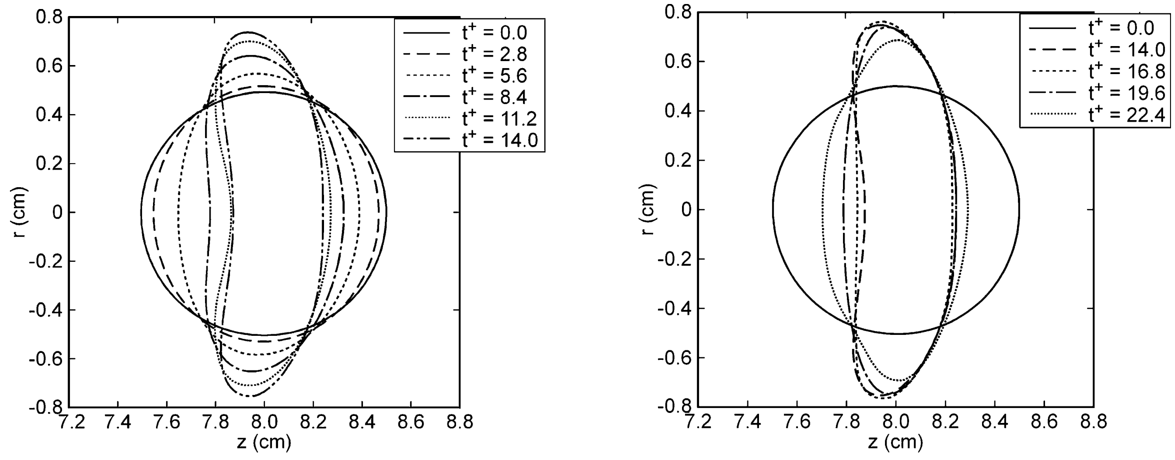
Fig. 8 Spherical liquid interface of the drop at the start of computation.



a) Computational grid and boundary conditions

b) Magnified view of the grid in and around the liquid drop

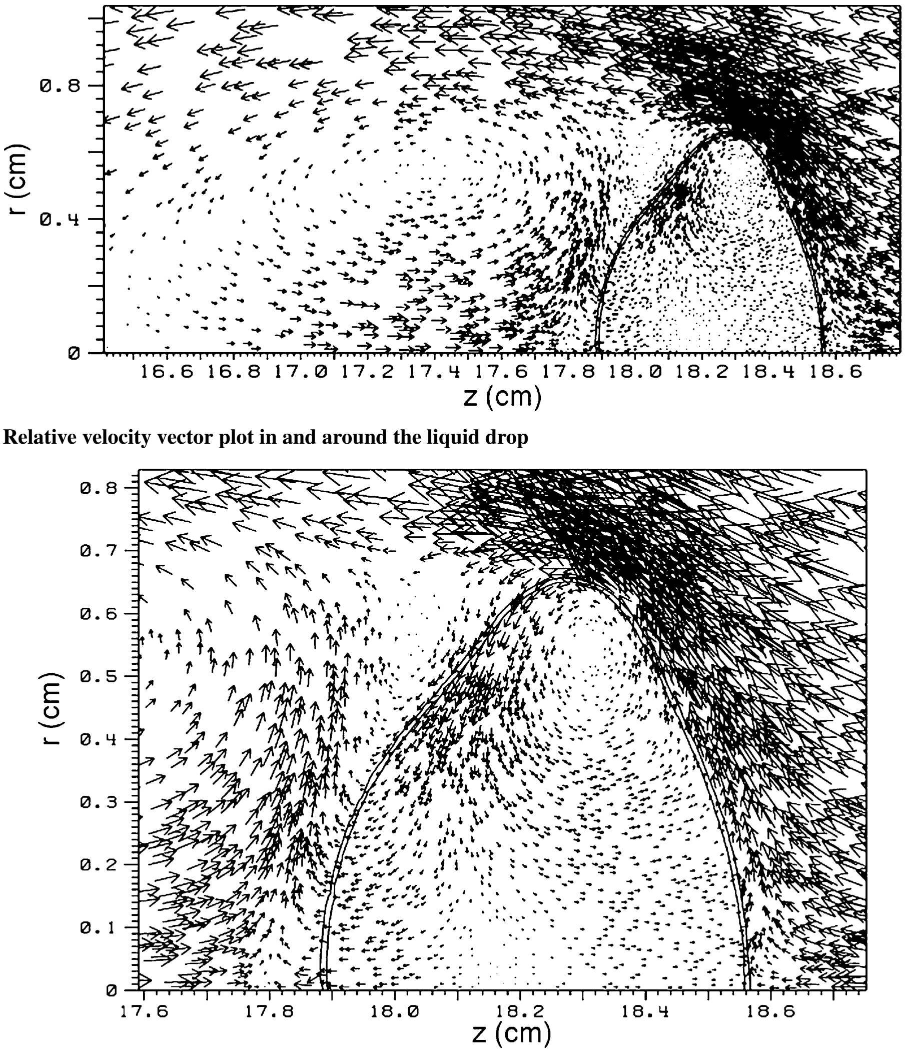
Fig. 7 Unstructured grid employed in liquid drop computations.



a) $t^+ \leq 14$

b) $t^+ \geq 14$

Fig. 9 Computed shapes of interface at different times for a decelerating liquid drop of $Oh_l = 0.01$.



a) Relative velocity vector plot in and around the liquid drop

b) Magnified view of the relative velocity vector plot

Fig. 10 Computed relative velocity vector plot for a decelerating liquid drop of $Oh_l = 0.01$ at $t^+ = 23.8$.

are the density and velocity of the freestream, that is, density and velocity of the gas phase at the start of computation. Computations were carried out for four different numerical resolutions. The number of vertices n employed to discretize the surface of the sphere in the lowest resolution case was 51. For higher-resolution computations, the number of vertices was increased in steps of 50% on all boundaries of the computational domain. The lowest- and highest-resolution computational grids, that is, $n = 51$ and 201, are shown in Figs. 3 and 4, respectively. The computed transient C_d for a compressible fluid with $\rho_{\text{solid}}/\rho_g = 5$ and initial $Re = 150$ is shown in Fig. 5 along with the results of Kim et al.[9] At a nondimensional time of 180, the computed C_d value for $n = 51$ is 27.3%, $n = 101$ is 9.9%, $n = 151$ is 5.6%, and $n = 201$ is 4.6%, greater than the results of Kim et al.[9] As the resolution is increased from $n = 51$ to 201, the computed drag shows better agreement with the data of Kim et al.[9] These results for solid spheres indicate

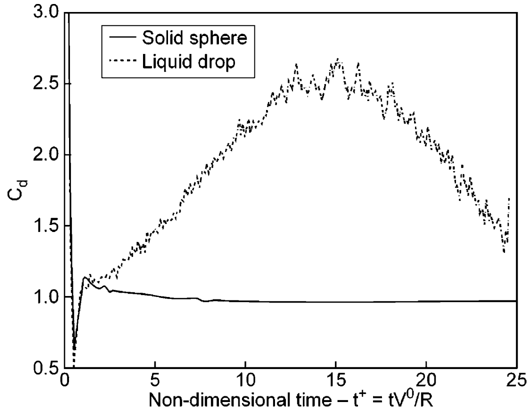


Fig. 11 Comparison of transient drag coefficient for a decelerating solid sphere and liquid drop of $Oh_l = 0.01$.

that the numerical code is capable of estimating the transient drag accurately.

A classical problem that is often employed to assess the accuracy of numerical codes for drop studies is the oscillating drop problem. An analytical solution for the small oscillations of a liquid drop about its spherical shape has been presented by Lamb.[15] The analytical frequency of oscillation ω is given as $\omega^2 = 24\sigma/R^3(3\rho_l + 2\rho_g)$, where R is the radius of the spherical drop, ρ_l is the liquid density, and ρ_g is the gas density. Computations were carried out to compare computed results with the analytical results. In the computation, $R = 1$ cm, $\rho_g = 1$ kg/m^3, $\sigma = 4700$ N/m, $\mu_g = 1.787 \times 10^{-5}$ Ns/m^2, $\rho_l/\rho_g = 100$, and $\mu_l/\mu_g = 35{,}000$. Initially both the air and liquid drop are at rest. Figure 6a shows the computed interface of the liquid drop at different times. It can be observed that the initially perturbed interface oscillates about a mean circular shape. Figure 6b shows the decaying amplitude of oscillations of the liquid drop. The computed time period of oscillation is 0.326 ms, which is 1.56% higher than the analytical value of 0.321 ms. For $\rho_l/\rho_g = 1000$, the computed time period is found to be 1.024 ms, which is 1.14% higher than the analytical value of 1.0125 ms.

Next, computations were carried out for a decelerating liquid drop, which was placed initially with a velocity of $V^0 = 70$ m/s in stagnant air at $\rho_g = 1.2$ kg/m^3 and $p_g = 1$ atm. Figure 7 shows the unstructured grid employed in the computations. The ratio of the liquid density to the ambient air density is 50, and the initial radius R is 0.5 cm, and Fig. 8 shows the spherical liquid interface of the drop at the start of computation. Based on the relative velocity of the drop and air, the initial $Re_g = 150$. The computation for the drop was carried out for an initial $We_g = \rho_g(\boldsymbol{V}_g - \boldsymbol{V}_l) \bullet (\boldsymbol{V}_g - \boldsymbol{V}_l)D/\sigma = 10$ and $Oh_l = \mu_l/\sqrt{(\rho_l \sigma D)} = 0.01$, where initially, $\boldsymbol{V}_g - \boldsymbol{V}_l = -\boldsymbol{V}^0$. $D = 2R$ is the diameter of the spherical drop at the start of the computation. The computed shape of the interface at different nondimensional times $t^+ = (tV^0)/R$, is shown in Fig. 6. The drop is moving in the positive z direction in an initially stagnant fluid. Figure 9a shows the deforming drop surface at several instants from $t^+ = 0$ until 14.0, and Fig. 9b shows the surface for $14.0 \leq t^+ \leq 22.4$. By $t^+ = 14.0$, the drop is observed to deform noticeably from its initial spherical shape. This deformation is a transient phenomenon, which is dependent on the transient velocity flowfield inside and outside the drop and the transient pressure field. For $t^+ \geq 14.0$, the drop shape tends to first reach an equilibrium and then retract back towards a spherical shape. To illustrate the complex nature of the flowfield, Fig. 10 shows relative velocity vector plots at $t^+ = 23.8$. The relative velocity is defined as $\boldsymbol{V}_{\text{rel}} = \boldsymbol{V} - \boldsymbol{V}_{\text{drop}}$. Figure 10b is a magnified view of the near-drop region in Fig. 10a. Two distinct vortices can be identified, one in the gas and another in the liquid at this instant. But the shape of the drop at any instant is a result of the history of transient velocities and pressure that the drop is subjected to internally and externally. Hence, a clear understanding of the behavior illustrated here would require that this transient flowfield be studied in great detail. This is done elsewhere.[28]

It is possible to derive global parameters from these computations. Consider the drag coefficient C_d, which is defined similar to Eq. (34) except that $\boldsymbol{V}_{\text{drop}}$ is employed instead of $\boldsymbol{V}_{\text{solid}}$ and $S(t)$ represents the deforming interface of the liquid drop. Figure 11 shows the computed transient drag coefficients for the liquid drop and a solid sphere with the same initial conditions as the drop. The C_d for the drop is observed to increase until it reaches a maximum value of 2.68 at about $t^+ \approx 15.0$, and then it decreases. The increase in drag is caused by drop deformation, as seen in Fig. 9a, which results in a flatter surface on the windward side of the drop until $t^+ = 14.0$. For $t^+ \geq 14.0$, the drop retracts from its deformed shape and tends toward its initial spherical shape. As the deformation decreases for $t^+ \geq 14.0$, the C_d also decreases.

IV. Summary

A hybrid compressible-incompressible numerical method, which employs a preconditioned form of the compressible flow equations, is developed for computing transient flows past drops at all Mach numbers. An artificial compressibility method is employed to solve the incompressible flow in the liquid phase. A finite volume numerical discretization is employed in the interface tracking method. Several problems are solved to illustrate the capabilities of the method. These include transient drag of a decelerating solid sphere, which is compared with other computed results published in the literature, comparisons of computed and analytical results of the time period of oscillations of an oscillating drop, and, as an illustration of its capability for computing moving drops, comparisons of the drag coefficients of a decelerating liquid drop with that of a solid sphere. The computed transient drag of the decelerating solid sphere shows an agreement within 4.6% with the prior numerical results. The oscillating drop frequency is computed within 1.6% accuracy of the analytical solution. An axisymmetric computation of a decelerating drop indicates that the transient drag coefficients are initially larger for the drop than for a corresponding solid sphere on account of drop deformation and the transient flowfield.

Acknowledgments

The authors thank the Purdue Research Foundation for partial support of this work. They also thank the Purdue University Computer Center for computer resources on the IBM SP-2 supercomputers.

References

[1] Qian, J., and Law, C. K., "Regimes of Coalescence and Separation in Droplet Collision," *Journal of Fluid Mechanics*, Vol. 331, 1997, pp. 59–80.

[2] Post, S. L., and Abraham, J., "Modeling the Outcome of Drop-Drop Collisions in Diesel Sprays," *International Journal of Multiphase Flow*, Vol. 28, No. 6, 2002, pp. 997–1019.

[3] Nobari, M. R. H., Jan, Y.–J., and Tryggvason, G., "Head-on Collision of Drops—A Numerical Investigation," *Physics of Fluids*, Vol. 8, No. 1, 1996, pp. 29–42.

[4] Nobari, M. R. H., and Tryggvason, G., "Numerical Simulations of Three-Dimensional Drop Collisions," *AIAA Journal*, Vol. 34, No. 4, 1996, pp. 750–755.

[5] Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S., and Zanetti, G., "Modeling Merging and Fragmentation in Multiphase Flows with SURFER," *Journal of Computational Physics*, Vol. 113, No. 1, 1994, pp. 134–147.

[6] Rieber, M., and Frohn, A., "Three-Dimensional Navier–Stokes Simulation of Binary Collisions Between Droplets of Equal Size," *Journal of Aerosol Science*, Vol. 26, Suppl. 1, 1995, pp. S929–S930.

[7] Magi, V., "A New 3-D Code for Flows, Sprays and Combustion in Reciprocating and Rotary Engines," Dept. of Mechanical and Aerospace Engineering, Princeton Univ., Rept. 1793, Princeton, NJ, Oct. 1987.

[8] Amsden, A. A., O'Rourke, P. J., and Butler, T. D., "KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays," Los Alamos National Lab., LANL TR LA-11560, Los Alamos, NM, May 1989.

[9] Kim, I., Elghobashi, S., and Sirignano, W. A., "On the Equation for Spherical-Particle Motion: Effect of Reynolds and Acceleration Numbers," *Journal of Fluid Mechanics*, Vol. 367, 1998, pp. 221–253.

[10] Clift, R., Grace, J. R., and Weber, M. E., *Bubbles, Drops and Particle*, Academic Press, San Diego, CA, 1978.

[11] Sirignano, W. A., *Fluid Dynamics and Transport of Droplets and Sprays*, Cambridge Univ. Press, Cambridge, England, U.K., 1999.

[12] Crowe, C., Sommerfeld, M., and Tsuji, Y., *Multi-Phase Flows with Droplets and Particles*, CRC Press, Boca Raton, FL, 1998.

[13] Helenbrook, B. T., and Edwards, C. F., "Quasi-Steady Deformation and Drag of Uncontaminated Liquid Drops," *International Journal of Multiphase Flow*, Vol. 28, No. 10, 2002, pp. 1631–1657.

[14] Temkin, S., and Mehta, H. K., "Droplet Drag in an Accelerating and Decelerating Flow," *Journal of Fluid Mechanics*, Vol. 116, 1982, pp. 297–313.

[15] Lamb, H., *Hydrodynamics*, Dover, New York, 1945.

[16] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Taylor and Francis, Washington, DC, 1997.

[17] Jameson, A., "Time-Dependent Calculations Using Multigrid, with Applications to Unsteady Flows past Airfoils and Wings," AIAA Paper 91-1596, June 1991.

[18] Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, No. 1, 1967, pp. 12–26.

[19] Gaitonde, A. L., "A Dual-Time Method for Two-Dimensional Unsteady Incompressible Flow Calculations," *International Journal for Numerical Methods in Engineering*, Vol. 41, No. 7, 1998, pp. 1153–1166.

[20] Müller, J.–D., "On Triangles and Flow," Ph.D. Dissertation, Dept. of Aerospace Engineering, Univ. of Michigan, Ann Arbor, MI, May 1996.

[21] Yee, H. C., "A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods," NASA TM 101088, Oct. 1989.

[22] Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., and Jan, Y.-J., "A Front-Tracking Method for the Computations of Multiphase Flow," *Journal of Computational Physics*, Vol. 169, No. 2, 2001, pp. 708–759.

[23] Poinsot, T. J., and Lele, S. K., "Boundary Conditions for Direct Simulations of Compressible Viscous Flows," *Journal of Computational Physics*, Vol. 101, No. 1, 1992, pp. 104–129.

[24] Thompson, K. W., "Time Dependent Boundary Conditions for Hyperbolic Systems," *Journal of Computational Physics*, Vol. 68, No. 1, 1987, pp. 1–24.

[25] Venkatakrishnan, V., and Mavriplis, D. J., "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids," AIAA Paper 95-1705, June 1995.

[26] Mavriplis, D. J., Jameson, A., and Martinelli, L., "Multigrid Solution of the Navier–Stokes Equations on Triangular Meshes," AIAA Paper 89-0120, Jan. 1989.

[27] Thomas, P. D., and Lombard, C. K., "Geometric Conservation Law and its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037.

[28] Wadhwa, A. R., "A Hybrid Compressible-Incompressible Numerical Method for Moving Drops," Ph.D. Dissertation, Dept. of Mechanical Engineering, Purdue Univ., West Lafayette, IN, Dec. 2004.

P. Givi
Associate Editor