Journal of Computational Physics **179,** 330–345 (2002)
doi:10.1006/jcph.2002.7065

An Efficient Method for Large-Scale Simulations of Bubbly Liquids

Sang-Yoon Kang*,[1] and Ashok S. Sangani†

*Department of Chemical Engineering, Korea Advanced Institute of Science and Technology, Taejeon 305-701, Korea; and †Department of Chemical Engineering and Materials Science, Syracuse University, Syracuse, New York 13244
E-mail: kangsa@mail.kaist.ac.kr

Received July 18, 2001; revised March 25, 2002

An efficient numerical method for simulating the motion of a large number of spherical bubbles through an inviscid liquid is described. The flow induced by each bubble is approximated as that due to a monopole and a dipole, and a rapid summation method based on dividing the system into groups of bubbles is used for evaluating interactions among bubbles. The computational time is shown to increase linearly with the number of bubbles. The method is shown to give accurate results for the effective properties of bubbly liquids.

Key Words: large-scale simulations; parallel computing; bubbly liquids; potential flow interactions; added mass coefficient.

1. INTRODUCTION

Bubbly liquids occur in many natural and industrial processes. Large-scale simulations of motion of bubbles can be used to develop closure relations for the average equations of motion of bubbly liquids and to understand the origin of macroscopic instabilities that develop in bubbly liquids. A case that has been examined analytically, numerically as well as theoretically, in detail over the last several years corresponds to high Reynolds and small Weber numbers defined by, respectively, $Re = R\rho v/\mu$ and $We = \rho R v^2/\sigma$, with R being the bubble radius, v the characteristic velocity of the bubbles, σ the interfacial tension, ρ the liquid density, and μ the viscosity. When We is small, the interfacial forces are strong enough to maintain approximately spherical bubbles, and, in the absence of surface-active impurities, the flow around bubbles at large Re can be approximated as a potential flow. Unlike the rigid particles, there is no boundary layer separation and the velocity correction in the boundary layer and wake behind the bubbles due to viscous effects is small compared

[1] Fax: 82-42-869-3910.

0021-9991/02 $35.00

with the potential flow [5]. Since the numerical solution of the Laplace equation for the velocity potential is much easier to obtain than the solution of the full Navier–Stokes equations, it is possible to make rapid analytical progress in developing equations of motion of bubbly liquids. In addition to the practical significance of bubbly liquids due to their occurance in many processes, bubbly liquids also serve as a model of suspensions in which the continuous-phase inertia is significant. As a consequence, a number of studies in the recent years have dealt with the subject of understanding the behavior of bubbly liquids in this ideal dual limit [1, 4, 6, 7, 9–12]. The case of flow induced by bubbles rising due to buoyancy forces acting on them is examined by Sangani and Didwania [6] and Smereka [9] and the case of bubble suspensions subjected to simple shear by Kang *et al*. [4]. A complete set of equations of motion for bubbly liquids together with the closure relations is given by Spelt and Sangani [10].

The above studies were primarily concerned with the behavior of nearly uniform bubbly liquids which could be numerically simulated with about 25–50 bubbles placed inside a unit cell of a periodic array, a commonly used model for simulating infinitely extended systems with only a finite number of particles or bubbles. The computational effort for the algorithms used in the numerical simulations cited above was $O(N^2)$, with N being the number of bubbles per unit cell. The periodicity suppresses any disturbances in the volume fraction of bubbles that may occur on a scale greater than the unit cell length. In some situations the inhomogeneity resulting from the instability of long wavelength disturbances in the volume fraction and the velocity of the bubbles may be important in understanding the macroscopic behavior of bubbly liquids, e.g., in understanding the mechanism for transition from a uniform bubbly liquid flow to a slug flow. For such situations it is necessary to simulate systems consisting of several thousand bubbles. Large-scale simulations are also needed for simulating the flow of bubbly liquid bounded by walls. The $O(N^2)$ algorithm used in the previous studies cannot be used for simulating such large systems. We describe here an algorithm that is suitable for simulating such systems. The algorithm is based on an $O(N)$ algorithm for Stokes and Laplace interactions described by Sangani and Mo [8]. The algorithm described by these investigators is quite general but somewhat complicated to understand. Some simplifications and modifications are made to make that algorithm suitable for simulating bubbly liquids. We describe this modified algorithm in detail here and evaluate the performance of the algorithm. Also, the aforementioned analytical work was mainly concerned with the case when the radius of the bubbles remains constant. We describe here the modifications that are required to simulate the case when the radius of the bubbles changes with time.

2. GOVERNING EQUATIONS AND THE METHOD OF MULTIPOLE EXPANSIONS

As mentioned in the Introduction, the flow induced by the bubbles at large Re and small We can be approximated as a potential flow. Thus the velocity $\mathbf{u}$ of the liquid at a point is related to the velocity potential φ by $\mathbf{u} = \nabla\varphi$, and, assuming that the liquid can be treated as an incompressible fluid, the continuity equation reduces to the Laplace equation for the velocity potential; i.e.,

$$\nabla^2\varphi = 0. \tag{1}$$

At small Weber numbers the bubbles remain spherical. We shall allow the pressure inside the bubble and hence the radius of the bubbles to vary depending on the ambient pressure.

When the ambient pressure is reduced the bubbles grow and during this growth phase the bubbles remain approximately spherical. Thus, the method to be presented here is most applicable either during the growth phase or for small Weber number flows with no change in the ambient pressure.

The kinematic condition at the surface of the bubble α ($\alpha = 1, 2, \ldots, N$) is given by

$$\mathbf{n} \cdot \nabla\varphi = \dot{R}^\alpha + \mathbf{n} \cdot \mathbf{v}^\alpha \quad \text{at} \quad |\mathbf{x} - \mathbf{x}^\alpha(t)| = R^\alpha, \tag{2}$$

where R^α, $\mathbf{x}^\alpha$, and $\mathbf{v}^\alpha$ are, respectively, the radius, position vector, and velocity of the bubble α at time t, $\mathbf{n}$ is the unit outward normal on the surface of the bubble, and the dot above a quantity implies its time derivative. The dynamic boundary condition is given by

$$p_g - p = 2\sigma/R, \tag{3}$$

where p_g is the pressure inside the gas bubble and p is pressure in the liquid. Since the density of the gas is typically much smaller than the density of the liquid, it can be shown, using the momentum equation for the gas phase, that the pressure inside the gas is independent of position and varies only with time. The pressure on the liquid side is related to the velocity potential by the Bernoulli's equation

$$p = p_0 - \rho\left[\frac{\partial\varphi}{\partial t} + \frac{1}{2}|\nabla\varphi|^2\right], \tag{4}$$

where p_0 is the ambient pressure. The pressure inside the gas depends on the equation of state for the gas and the temperature variations in the gas, and, in general, one must solve for the energy equation to determine the temperature distribution and hence the gas pressure. We may adopt a simple polytropic model and take

$$p_g = BR^{-3n}. \tag{5}$$

The ideal gas case with nearly constant temperature corresponds to $n = 1$. The ambient pressure p_0 must be prescribed as a function of time as a part of the statement of the problem.

The kinematic and dynamic boundary conditions can be expressed in the integral form by multiplying them with Y_{kl} and integrating over the surface of bubbles, with Y_{kl} being the surface spherical harmonics; i.e., $Y_{kl} = P_k^l(\mu)e^{il\phi}$. Here, P_k^l is the associated Legendre function, $\mu = \cos\theta$, and θ and ϕ are the spherical polar angles with $\mathbf{x}^\alpha$ as the center of the local spherical coordinate system and the x_1-axis as the polar axis. The kinematic condition is therefore expressed as

$$\int_{\mu=-1}^{1}\int_{\phi=0}^{2\pi} \nabla\varphi \cdot \mathbf{n} Y_{kl}(R^\alpha)^2\, d\mu\, d\phi = \int_{\mu=-1}^{1}\int_{\phi=0}^{2\pi} [\mathbf{v}^\alpha + \dot{R}^\alpha \mathbf{n}] \cdot \mathbf{n} Y_{kl}(R^\alpha)^2\, d\mu\, d\phi. \tag{6}$$

Next, using the identity

$$\begin{aligned}\frac{d}{dt}\int_{-1}^{1}\int_{0}^{2\pi} &\varphi(R^\alpha)^2 Y_{kl}(\mu,\phi)\, d\mu\, d\phi \\ &= \int_{-1}^{1}\int_{0}^{2\pi}\left[\frac{\partial\varphi}{\partial t} + 2\varphi\frac{\dot{R}^\alpha}{R^\alpha} + \nabla\varphi \cdot (\mathbf{v}^\alpha + \dot{R}^\alpha \mathbf{n})\right](R^\alpha)^2 Y_{kl}\, d\mu\, d\phi,\end{aligned} \tag{7}$$

the dynamic boundary condition can be expressed as

$$\frac{dI_{kl}}{dt} = \int_{-1}^{1}\int_{0}^{2\pi}\left[\frac{1}{\rho}\left\{p_0(t) - p_g(t) + \frac{2\sigma}{R^\alpha}\right\} + \nabla\varphi\cdot\left\{\mathbf{v}^\alpha + \dot{R}^\alpha\mathbf{n} - (1/2)\nabla\varphi\right\} + \frac{2\dot{R}^\alpha}{R^\alpha}\varphi\right](R^\alpha)^2 Y_{kl}\, d\mu\, d\phi. \tag{8}$$

Here, I_{kl} are the moments of velocity potential distribution on the surface of the bubble defined by

$$I_{kl} = \int_{-1}^{1}\int_{0}^{2\pi}\varphi Y_{kl}(R^\alpha)^2\, d\mu\, d\phi. \tag{9}$$

It should be noted that the dynamic boundary condition (3) is approximate since the bubbles are treated as spherical. Small deformations in the spherical shape will make significant contributions to (8) for $k > 1$ when the Weber number is small. Thus, (8) must be used only for $k \leq 1$. On the other hand, the small deformations affect the kinematic condition by an insignificant amount and therefore (7) can be used for all k.

The numerical scheme for determining the radius and the position of the bubbles as functions of time for small Weber number flows consists of (i) determining the velocity potential φ at time t given I_{kl} with $k \leq 1$ at time t and the kinematic condition (6) for $k > 1$; (ii) determining $\mathbf{v}^\alpha$ and $\dot{R}^\alpha$ once φ is determined using (6) for $k \leq 1$; and (iii) determining $\dot{I}^\alpha_{kl}$ for $k \leq 1$ and hence I^α_{kl} at time $t + \triangle t$ from (8). Here, $\triangle t$ is the time step for integrating the trajectories of the bubbles.

Simulations for smaller systems of bubbles by Sangani and Didwania [6] have shown that the method of multipole expansions is very efficient for determining the velocity potential. A sufficiently accurate solution can be obtained by neglecting the multipoles of higher order than a dipole. For the case of N bubbles placed within a unit cell of a periodic array the velocity potential is expressed in terms of fundamental periodic singular solutions of the Laplace equation [3] as

$$\varphi(\mathbf{x}) = \varphi_\infty(\mathbf{x}) + \sum_{\alpha=1}^{N}[M^\alpha + \mathbf{D}^\alpha\cdot\nabla]S_1(\mathbf{x} - \mathbf{x}^\alpha), \tag{10}$$

where M^α and $\mathbf{D}^\alpha$ are, respectively, the monopole and dipole induced by the presence of bubble α, and S_1 is the fundamental singular solution satisfying [3],

$$\nabla^2 S_1(\mathbf{x}) = 4\pi\left[\tau^{-1} - \sum_{\mathbf{r_L}}\delta(\mathbf{x} - \mathbf{x_L})\right], \tag{11}$$

where τ is the volume of the unit cell and $\mathbf{r_L}$ are the lattice points of the periodic array. For φ to satisfy the Laplace equation, φ_∞ must satisfy

$$\nabla^2\varphi_\infty + (4\pi/\tau)\sum_{\alpha=1}^{N} M^\alpha = 0. \tag{12}$$

The monopoles of the bubbles are related to the rate of change of bubble volume. When the sum of the volume changes of all the bubbles is nonzero, the unit cell volume must

increase since the liquid is incompressible. The shape and volume of the unit cell change to accommodate the volume changes in the bubbles and the container geometry. We consider the case where the bubbly liquid is restricted from growing along the x_2- or x_3-axis by the walls of the container. This case can be modeled with a periodically extended system in which only the unit cell height along the x_1-axis is allowed to vary. On solving (12) for φ_∞ we obtain

$$\varphi_\infty = \lambda x_1^2 + \mathbf{G}\cdot\mathbf{x}, \tag{13}$$

where

$$\lambda = -(2\pi/\tau)\sum_{\alpha=1}^{N} M^\alpha \tag{14}$$

and $\mathbf{G}$ is the backflow contribution chosen to be the same as in Sangani and Didwania [6] and Spelt and Sangani [10]; i.e.,

$$\mathbf{G} = -(4\pi/\tau)\sum_{\alpha=1}^{N} \mathbf{D}^\alpha. \tag{15}$$

It can be shown that the average velocity of the gas–liquid mixture with the above choice of $\mathbf{G}$ is given by

$$\langle \mathbf{u}\rangle = 2\lambda x_1 \mathbf{e}_1, \tag{16}$$

with $\mathbf{e}_1$ being the unit vector along the x_1-axis. The velocity potential can be expanded in a series of singular and regular solutions of the Laplace equation near each bubble as given by

$$\varphi(\mathbf{x}) = A^\alpha + \frac{M^\alpha}{r} + C_j^\alpha r_j - D_j^\alpha \frac{r_j}{r^3} + E_{jk}^\alpha (r_j r_k - \delta_{jk} r^2/3)/2 + \cdots, \tag{17}$$

where $\mathbf{r} = \mathbf{x} - \mathbf{x}^\alpha$, $E_{jk} = E_{kj}$, and $E_{kk} = 0$. Note that the terms singular at $r = 0$ in the above expansion arise from the singular part $1/r$ of $S_1(\mathbf{x} - \mathbf{x}^\alpha)$ [3]. The regular terms represent the contribution to the velocity potential at $\mathbf{x}^\alpha$ from the bubbles other than the bubble α. A^α and $\mathbf{C}^\alpha$ are related to the regular part of φ near $\mathbf{x} = \mathbf{x}^\alpha$ by

$$A^\alpha = \varphi^r(\mathbf{x}^\alpha), \quad C_j^\alpha = \frac{\partial \varphi}{\partial x_j}(\mathbf{x}^\alpha), \quad E_{jk}^\alpha = \frac{\partial^2 \varphi^r}{\partial x_j \partial x_k}(\mathbf{x}^\alpha). \tag{18}$$

Since we are only interested in $k \le 1$ terms in (9) it is convenient to use the vector forms of I_{kl}. Noting that Y_{1l} are related to the components of the unit normal vector on the spherical surface, we introduce the impulse of a bubble defined by

$$\mathbf{I}^\alpha = -\rho \int_{-1}^{1}\int_{0}^{2\pi} \varphi \mathbf{n} (R\alpha)^2 \, d\mu \, d\phi. \tag{19}$$

The total momentum of the liquid is related to the sum of the impulses of the bubbles [6]. Thus, the impulse of the bubble may be regarded as the virtual momentum of the bubble.

Now the use of the expansion given by (17) gives

$$\mathbf{I}^\alpha = -(4\pi\rho/3)[\mathbf{C}^\alpha (R^\alpha)^3 - \mathbf{D}^\alpha] \tag{20}$$

and

$$I_{00}^\alpha = 4\pi (R^\alpha)^2 [A^\alpha + M^\alpha / R^\alpha]. \tag{21}$$

The dynamic boundary condition (8) reduces to

$$\frac{dI_{00}^\alpha}{dt} = 2\pi \left[-3 \left(\frac{M^\alpha}{R^\alpha} \right)^2 - 4 \frac{A^\alpha M^\alpha}{R^\alpha} + \mathbf{C}^\alpha \cdot \mathbf{C}^\alpha (R^\alpha)^2 + 4 \frac{\mathbf{C}^\alpha \cdot \mathbf{D}^\alpha}{R^\alpha} + 2 \frac{(R^\alpha)^2}{\rho} \left(p_0 + \frac{2\sigma}{R^\alpha} - p_g \right) \right], \tag{22}$$

$$\frac{dI_j^\alpha}{dt} = -4\pi\rho \left[M^\alpha C_j^\alpha + E_{jk}^\alpha D_j^\alpha \right]. \tag{23}$$

A numerical scheme for simulating the motion of bubbles consists of determining the multipoles M and $\mathbf{D}$ for each bubble given I_{00} and I_j for all the bubbles at time t. Next, (22) and (23) are used to determine the time derivatives of I_{00} and I_j, and this, in turn, yields the estimates of I_{00} and I_j at time $t + \Delta t$. The position and the radius of the bubbles are updated using

$$\dot{\mathbf{x}}^\alpha = \mathbf{v}^\alpha = \mathbf{C}^\alpha + \frac{2\mathbf{D}^\alpha}{(R^\alpha)^3}, \tag{24}$$

$$\dot{R}^\alpha = -\frac{M^\alpha}{(R^\alpha)^2}. \tag{25}$$

Finally, the unit cell height along the x_1-axis is updated using

$$\dot{h}_v = -\frac{4\pi}{h^2} \sum_{\alpha=1}^{N} M^\alpha, \tag{26}$$

where h is the unit cell length along the x_2- and x_3-axes, with the unit cell being assumed to be oblong in the shape.

The constants A^α and $\mathbf{C}^\alpha$ are related to the monopole and dipole of the bubbles by

$$A^\alpha = \varphi_\infty(\mathbf{x}^\alpha) + \sum_{\gamma=1}^{N} [A^\gamma + \mathbf{D}^\gamma \cdot \nabla] S_1^r (\mathbf{x}^\alpha - \mathbf{x}^\gamma), \tag{27}$$

$$\mathbf{C}^\alpha = \nabla\varphi_\infty(\mathbf{x}^\alpha) + \sum_{\gamma=1}^{N} [A^\gamma + \mathbf{D}^\gamma \cdot \nabla] \nabla S_1^r (\mathbf{x}^\alpha - \mathbf{x}^\gamma), \tag{28}$$

where S_1^r equals S_1 for $\gamma \neq \alpha$ and equals the regular part of S_1, i.e., $S_1 - 1/r$, for $\gamma = \alpha$.

Substitution of the above equations into (20) and (21) leads to a set of $4N$ linear equations in the same number of unknowns. These equations can be written in the form $A_{ij} X_j = b_i$ ($i = 1, 2, \ldots, 4N$), where X_j is the vector consisting of monopoles and dipoles of the bubbles

and b_i is a vector consisting of I_{00} and $\mathbf{I}$ of the bubbles. For the case where the ambient pressure is constant, and the radius of the bubbles remains constant, the monopoles vanish and the above set can be reduced to a set of three equations per bubble by retaining only the equations for the impulses of the bubbles, as was done by Sangani and Didwania [6] and Spelt and Sangani [10]. The algorithm used by these investigators requires an $O(N^2)$ computational effort since the derivatives of S_1 must be evaluated for all $(N(N-1)/2)$ pairs of bubbles and since determining the solution of $4N$ linear equations of the form $A_{ij}X_j = b_i$, given the coefficients of the $4N \times 4N$ matrix A_{ij}, is at least an $O(N^2)$ effort (note that evaluation of the coefficients of the matrix A_{ij} itself is an $O(N^2)$ effort). The time required to evaluate the derivatives of S_1 is the computationally most expensive $O(N^2)$ step since it requires evaluating sums over real and reciprocal lattice space vectors [3].

3. THE $O(N)$ ALGORITHM

Large systems of equations are usually solved by iterative methods. Thus one must evaluate $A_{ij}X_j$ for an assumed value of X_j and use a suitable iterative method to generate subsequent guesses for X_j. The number of iterations required depends on the algorithm used and the nature of the matrix A_{ij}. In general, a smaller number of iterations are required when the eigenvalues of the matrix are widely separated. The matrix A_{ij} for bubble interaction calculations has this property and we therefore expect the total number of iterations to be small and relatively independent of N. Thus the total computational times scales roughly in proportion to the time it takes to evaluate $A_{ij}X_j$ for given X_j, and the main objective of the efficient algorithms is to devise efficient methods for computing $A_{ij}X_j$. This must be done without computing all the elements of A_{ij} first, since that would immediately result in an $O(N^2)$ algorithm. Note also that computing the elements of A_{ij} would take up $O(N^2)$ bytes of computer memory.

To develop an algorithm in which the total computational effort scales linearly with N, we must avoid evaluating the derivatives of S_1 for each pair of bubbles. The velocity field induced by the bubbles that are close to each other and at a relatively large distance from the bubble α can be more efficiently evaluated by grouping the bubbles together. To group such bubbles we use a Taylor series expansion,

$$\begin{aligned}
&\sum \left[M^\gamma + D_j^\gamma \frac{\partial}{\partial x_j} \right] S_1(\mathbf{x} - \mathbf{x}^\gamma) \\
&\quad = \sum \left[M^\gamma + \left(y_j M^\gamma + D_j^\gamma \right) \frac{\partial}{\partial x_j} + \left(\frac{y_j y_k}{2} M^\gamma + y_k^\gamma D_j^\gamma \right) \frac{\partial^2}{\partial x_j \partial x_k} + \cdots \right] S_1(\mathbf{x} - \mathbf{x}^c) \\
&\quad = \left[M^c + D_j^c \frac{\partial}{\partial x_j} + Q_{jk}^c \frac{\partial^2}{\partial x_j \partial x_k} + \cdots \right] S_1(\mathbf{x} - \mathbf{x}^c),
\end{aligned} \tag{29}$$

where the summation is over all the bubbles with their centers $\mathbf{x}^\gamma$ lying within a volume, say V, with its center at $\mathbf{x}^c$, $\mathbf{y}^\gamma = \mathbf{x}^c - \mathbf{x}^\gamma$ is the position vector of the center with respect to the bubble center $\mathbf{x}^\gamma$, and M^c, D_j^c, and Q_{jk}^c are, respectively, the equivalent monopole, dipole, and quadrupole representing the combined effect of all the bubbles lying within V. Note that these are related to the monopoles and dipoles of the bubbles through equations

$$M^c = \sum M^\gamma, \quad D_j^c = \sum \left(M^\gamma y_j + D_j^\gamma \right), \quad Q_{jk}^c = \sum \left(M^\gamma y_j y_k / 2 + D_j^\gamma y_k^\gamma \right). \tag{30}$$

In other words, the velocity potential induced by a group of bubbles can be expressed in terms of a series of multipoles at the center of the box containing the bubbles. This representation will be accurate with only a few terms in the series retained when it is used to evaluate the induced velocity potential at large distances from $\mathbf{x}^c$, i.e., for $\ell/|\mathbf{x} - \mathbf{x}^c| \ll 1$, with ℓ being the linear dimension characterizing V. To evaluate the velocity potential at $\mathbf{x}$ induced by the bubbles that are not far from $\mathbf{x}$, many more terms in the series will be necessary and it is therefore more appropriate to evaluate separately the velocity potential due to each such bubble.

To show that the grouping of the bubbles will lead to a total computational effort that is smaller than $O(N^2)$, let us consider first a simple case in which the unit cell is divided into M boxes with each box on average containing N/M bubbles. Computation of the multipole moments will require $O(N/M)$ operations per box or a total of $O(N)$ operations. Computation of the velocity field at the center of each box due to the moments at the center of other boxes will require $O(M^2)$ operations. Finally, computation of the velocity at the center of each bubble due to bubbles that are within a distance comparable to the box size will require $O(N^2/M)$ operations since there are roughly $O(N/M)$ bubbles close to each bubble, for which the grouping will not be efficient. Thus the total computational effort is $O(N) + O(M^2) + O(N^2/M)$. This has a minimum of $O(N^{4/3})$ for $M = O(N^{2/3})$. Thus, a straightforward grouping with each group containing an $O(N^{1/3})$ bubbles will result in an algorithm in which the computational effort will scale as $O(N^{4/3})$.

To obtain an even more efficient algorithm it is necessary to have groups of different sizes. The velocity potential near a bubble can then be evaluated from larger groups of bubbles that are further away from the bubble than the groups of bubbles that are close to it. This hierarchial grouping of particles was first used by Greengard and Rokhlin [2]. As mentioned in the introduction, we shall use the algorithm described by Sangani and Mo [8]. Accordingly, the total unit cell volume is first divided into eight equal-size boxes, which are referred to as the zeroth-level boxes. Each of these boxes is divided again into eight smaller boxes, resulting in a total of $8^2 = 64$ first-level boxes. These boxes are further subdivided into $8^3 = 512$ second-level boxes, and this procedure is repeated to a total of $M = 8^{m+1}$ finest, m-level boxes such that $M = O(N)$. In other words, each finest-level box contains an $O(1)$ number of bubbles. Thus $m + 1 = O(\log_8 N)$.

The $O(N)$ algorithm consists of three steps. The first step computes the multipole moments for each box, as represented by black small squares located at the center boxes in Fig. 1. Given X_j, which are related to the assumed value of monopoles and dipoles induced by the bubbles, the equivalent multipoles for the finest-level boxes are evaluated using (30). These multipoles in turn are used to evaluate the multipoles of the coarser, $(m-1)$-level, boxes. Each box at this level consists of eight finest-level boxes, whose multipoles have already been evaluated. The expressions for computing multipoles are similar to those given by (30), with the sum now over the eight "children" boxes instead of bubbles γ. This procedure of computing the multipole moments of larger boxes from the knowledge of multipoles of their eight "children" boxes is continued to the coarsest-level boxes. Figure 1 demonstrates this procedure graphically for the finest-level boxes in the two-dimensional case. The computational effort for evaluating the multipoles of the finest-level boxes is $O(N)$, and the effort for computing moments of coarser-level boxes decreases roughly by a factor of eight, with every level owing to a fewer number of boxes at the coarser levels. Thus, even though the total number of levels is $O(\log N)$, the total computational time for computing multipoles of all the boxes at all levels is $O(N)$. This step of the algorithm is

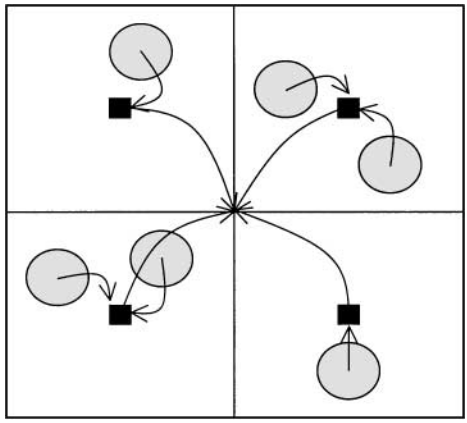

FIG. 1. Demonstration of the upward pass for the finest-level boxes in the two-dimensional case.

referred to as the *upward pass*, since we are computing multipoles of higher-level boxes from the lower-level boxes and the monopoles and dipoles of the bubbles.

The second step is referred to as the *downward pass*. In this we determine the velocity potential induced by the bubbles in all the boxes that are not in the immediate vicinity of a given box. Thus, we determine φ^r due to all but the 27 ($3 \times 3 \times 3$) boxes surrounding a box at a given level. The downward pass consists of determining the coefficients A, $\mathbf{C}$, $\mathbf{E}$, and so forth, that appear in the expansion of the regular part of φ (cf. (17)) for each box, since φ^r can always be determined, in principle, if its derivatives at some point are known. (Note that, as can be seen from (18), $\mathbf{C}$, $\mathbf{E}$, etc., are related to the derivatives of φ^r.) The coarsest-level boxes indicated by solid lines in Fig. 1 are all the nearest neighbors of each other and therefore the coefficients A, $\mathbf{C}$, $\mathbf{E}$, and so forth, of these boxes are taken to be zero. At the next level there are 64 boxes. For each box, therefore, $64 - 27 = 37$ boxes contribute to the regular expansion coefficients of a box. The boxes that are "children" of a nearest neighbor of a "parent" box are referred to as the equal-generation boxes. There are 37 equal-generation boxes for each first-level box and $6^3 - 3^3 = 189$ equal-generation boxes for each higher-level box. The gray colored boxes in Fig. 2a and 2b are equal-generation boxes corresponding to the hatched box. The coefficients in the expansion of φ^r around a center $\mathbf{x}^{c_1}$, i.e. $\otimes$, of a box in first level are evaluated by summing over all its equal-generation boxes, using formulas similar to (27) and (28). Thus, for example, the expression for evaluating the contribution from the equal-generation boxes is

$$C_i^{eq\to c_1} = \frac{\partial \varphi^r}{\partial x_{\mathbf{i}}}\left(\mathbf{x}^{c_1}\right) = \sum\nolimits_{eq}\left[M^{eq}\frac{\partial}{\partial x_i} + D_j^{eq}\frac{\partial^2}{\partial x_i \partial x_j} + Q_{jk}^{eq}\frac{\partial^3}{\partial x_i \partial x_j \partial x_k} + \cdots\right] S_1\left(\mathbf{x}^{c_1} - \mathbf{x}^{eq}\right), \quad (31)$$

where eq represents an equal-generation box corresponding to the small black squares in Fig. 2a. For second- and finer-level boxes the coefficients of the regular expansion has two contributions. As in the first-level boxes, there is a contribution from the 189 equal-generation boxes. The second contribution is from its parent, first-level box. The contribution from the equal-generation boxes can be calculated using formulas similar to (31). The other

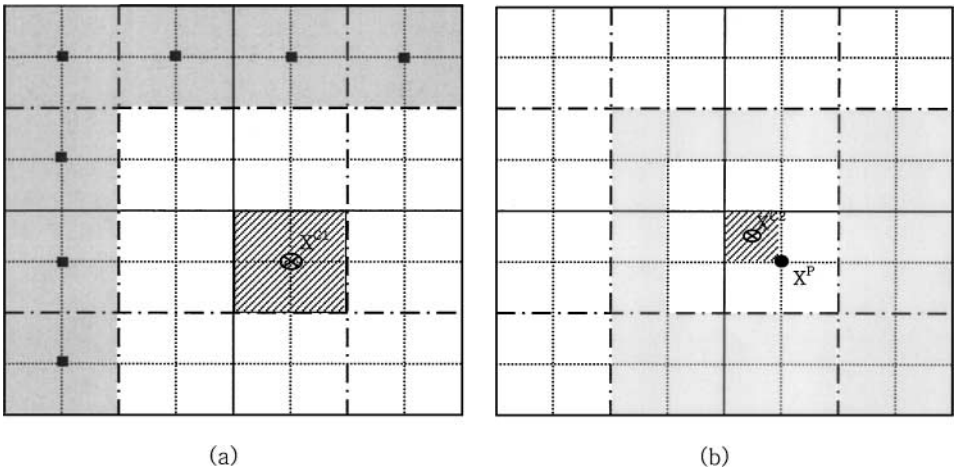

FIG. 2. Demonstrations of equal-generation boxes. The gray boxes are equal-generation boxes of hatched boxes. (a) Equal-generation boxes of first level; (b) equal-generation boxes of second level.

part, which is the contribution from the parent of a given box, is obtained by reexpanding the expansion for regular part of φ around the parent's center $\mathbf{x}^p$ to that around the center of the child's box. Here $\mathbf{x}^p$ is the same as $\mathbf{x}^{c_1}$ in Fig. 2a. Thus,

$$\mathbf{C}^{p\to c_2} = \nabla\varphi\left(\mathbf{x}^{c_2}\right) = \mathbf{C}^p + 2\mathbf{E}^p \cdot \left(\mathbf{x}^{c_2} - \mathbf{x}^p\right). \tag{32}$$

This procedure of computing the coefficients A, $\mathbf{C}$, $\mathbf{E}$, and so forth, at each box from the coefficients A, $\mathbf{C}$, $\mathbf{E}$, and so forth, of its parent box and multipoles M, $\mathbf{D}$, $\mathbf{Q}$, and so on, of its equal generation boxes is continued to the finest-level boxes. Finally, the coefficients in the expansion of the regular part of φ around each bubble are determined by reexpanding the field around the center of each finest-level boxes to that around the center of each bubble.

The above calculation determines the contribution to the regular coefficients from all the bubbles lying outside the 27 boxes immediately surrounding each bubble. The third step consists of adding the contribution from the bubbles in these 27 boxes. At this stage it is possible to evaluate $A_{ij}X_j$ for given X_j. Note that this amounts to evaluating A^α and $\mathbf{C}^\alpha$ for each bubble and determining I_{00}^α and $\mathbf{I}^\alpha$ using (20) and (21).

The above procedure must be repeated with a new guess for X_j if $A_{ij}X_j$ is not close enough to b_i. We used a subroutine DGMRES based on a generalized moment-residual method for determining the new guess.

Once the correct values of M, $\mathbf{D}$, A, and $\mathbf{C}$ for each bubble are determined, the velocity and the rate of change of radius of each bubble are evaluated using (24) and (25). Finally, the time derivatives of I_{00} and $\mathbf{I}$ are evaluated using (22) and (23) and this, together with a suitable integration routine, can be used to determine the values of I_{00} and $\mathbf{I}$ at the next time step.

4. PERFORMANCE EVALUATION

In this section we test the algorithm described in the previous section for the accuracy and computational efficiency. Although the scheme described above was restricted for the

sake of clarity to the bubbles represented by monopoles and dipoles, the computer program we developed allowed inclusion of higher order multipoles. In the results presented below N_s and N_{sp} represent, respectively, the multipole orders used for representing the flow induced by bubbles and by the group of bubbles. Thus, $N_s = 1$ corresponds to representing the bubbles by monopole and dipoles and $N_s = 2$ corresponds to including in addition the quadrupoles induced by the bubbles. Also we considered only the case of constant bubble radius, which has been examined by the $O(N^2)$ algorithm used by previous investigators. The accuracy of the $O(N^2)$ algorithm is solely determined by N_s. The calculations presented in Sangani and Didwania [6], Spelt and Sangani [10], and Kang *et al.* [4] all corresponded to using $N_s = 1$. As noted earlier, this gives reasonably accurate (typically within 10%) estimates for various properties of bubbly liquids. For the $O(N)$ algorithm there is an additional parameter, N_{sp}, that is the order of multipoles used either to represent the combined effect of a group of bubbles or in expanding the regular part of φ. The $O(N)$ algorithm should give the same result as that obtained with the $O(N^2)$ algorithm for a given N_s as $N_{sp} \to \infty$. The total computational effort scales as $N N_{sp}^4$, and it is therefore important from a practical point of view to assess the accuracy of the $O(N)$ algorithm when N_{sp} is relatively small. We aim for the total error arising from the trucation with finite N_s or N_{sp} to less than, say, 10%.

The accuracy and efficiency of the algorithm were evaluated by calculating various properties of bubbly liquids with a total of 1024 bubbles per unit cell. Since the computational effort for this number of bubbles by the $O(N^2)$ algorithm is excessive, we assign the positions of the bubbles by dividing the unit cell into 64 equal-size boxes, with each box containing 16 randomly placed bubbles. All 64 boxes are identical and therefore this arrangement of bubbles is the same as a random array of bubbles with $N = 16$. We used the $O(N^2)$ algorithm to calculate the average properties of the random array with $N = 16$ and compared it with the results obtained with $N = 1024$ bubbles, using the $O(N)$ algorithm. As mentioned above, the results should be identical in the limit $N_{sp} \to \infty$ for a fixed vallue of N_s in both cases. The results for configurations with $\phi = 0.1$ and 0.25 are shown in Table I. Here ϕ is the volume fraction of the bubbles. The results shown correspond to two cases. In the first case, the velocity of all the bubbles was taken to be along the x_1-axis and to

TABLE I
Computed Results with $N = 1024$ and 16 for Volume Fractions $\phi = 0.1$ and 0.25

N	ϕ	N_s	N_{sp}	C_k	f_k	C_a	f_a
1024	0.1	2	2	0.957	0.0206	1.261	0.0244
			3	0.959	0.0209	1.263	0.0246
			4	0.958	0.0210	1.264	0.0248
			5	0.959	0.0209	1.265	0.0247
16	0.1	1		0.949	0.0225	1.256	0.0252
		2		0.961	0.0230	1.266	0.0260
1024	0.25	2	2	1.051	0.0767	2.039	0.125
			3	1.048	0.0743	2.038	0.117
			4	1.050	0.0754	2.037	0.117
			5	1.050	0.0755	2.040	0.119
16	0.25	1		1.021	0.068	2.000	0.103
		2		1.067	0.074	2.040	0.111

have a unit magnitude, and the impulse of each bubble was determined. The added mass coefficient C_a was then computed using

$$C_a = \frac{2}{m}\frac{\langle I_1 \rangle}{\langle v_1 \rangle}, \tag{33}$$

with the average $\langle v_1 \rangle$ being, of course, unity. Here, $m = (4\pi/3)\rho R^3$. For this case we also determine the potential interaction force $\mathbf{F}_p \equiv \dot{\mathbf{I}}$ for each bubble. The sum of the the interaction force over all the bubbles must be identically zero [6] and therefore it is more meaningful to evaluate the variance in the potential interaction force. This is defined in terms of a nondimensional coefficient

$$f_a = \frac{R^2}{Nm^2\langle v_1 \rangle^4}\sum_{\alpha=1}^{N} \mathbf{F}_p^{\alpha} \cdot \mathbf{F}_p^{\alpha}. \tag{34}$$

As seen in Table I, the results for C_a and f_a obtained by the $O(N)$ and $O(N^2)$ are in excellent agreement with each other. We note that $N_s = N_{sp} = 2$ gives sufficiently accurate results and that it is unnecessary to evaluate higher order moments or coefficients.

Also shown in Table I are the results for a case in which all the bubbles are given isotropic velocity fluctuations. The average impulse and average velocity in this case are zero, although the total kinetic energy of the liquid per unit volume as given by

$$\langle \mathbf{I} \cdot \mathbf{v} \rangle = \frac{m}{2} C_k \langle \mathbf{v} \cdot \mathbf{v} \rangle \tag{35}$$

is not zero. Once again, the results for C_k obtained by the two algorithms are seen to be in excellent agreement with each other. Finally, analogous to (34), we can define the variance in the potential interaction in terms of a nondimensional coefficient f_k. Since the average velocity is zero in this case, we used $\langle v^2 \rangle$ to nondimensionalize this coefficient. The difference in f_k determined by the two algorithms is somewhat greater in this case (slightly more than 10%). Considering that the potential interaction forces play a relatively insignificant role in the dynamics of bubbly liquids subjected to isotropic velocity fluctuations, this difference will not be important in simulations of flows of bubbly liquids. (Note that the coefficient f_k is very small in magnitude.)

It is useful to give some idea regarding the time taken by different steps in the $O(N)$ algorithm. As mentioned earlier, each iteration consists of three steps: the upward pass, the downward pass, and the bubble–bubble interactions. The upward and downward passes require $O(NN_{sp}^4)$ operations, while the bubble–bubble step requires $O(NN_s^2N_{sp}^2)$ operations. With $N_s = N_{sp} = 2$, all the steps scale with N and N_s in the same way. However, the coefficient for the downward pass is much greater than that for the upward pass. This is so because, in general, one must sum over 189 equal-generation boxes for each box in the downward pass. As a result, the time needed for the upward pass is essentially negiligible in comparison with that for the downward pass. The bubble–bubble step requires summing over 27 boxes. Thus, the number of operations required by this step are roughly given by $27NPN_s^4$, with P being the number of bubbles. In other words, the time spent in the downward pass is about $6/P$ times the time spent in bubble–bubble calculations. The total computational time per bubble is minimum when P is in the range 2–4 and the computational time is governed equally by the downward pass and the bubble–bubble interaction step.

TABLE II
Number of Iterations as a Function of Error

Error	Number of iterations	C_a
0.1	2	1.982
0.001	3	2.103
0.00001	3	2.103

It was mentioned earlier that the matrix A_{ij} arising in the dynamics of bubbly liquids has the property of largely separated eigenvalues. To show that the convergence to the exact solution is very rapid by the iterative method used in the present study, we give results for the number of iterations in Table II. The results are given for a random configuration of bubbles with $N = 1024$. The velocity of all the bubbles was taken to be equal and along the x_1-axis. The initial guess for the dipoles for all the bubbles was taken to be the same as the velocity of the bubbles. The DGMRES subroutine computes an estimate of the error by computing a norm of $A_{ij}X_j - b_i$. This error estimate and C_a as functions of the number of iterations are shown in Table II. We see that the convergence is very rapid and that only a few iterations are required to solve the system of equations.

We now compare the computer time required by the two algorithms. The results are presented in Table III and Fig. 3. The times for the $O(N^2)$ algorithm listed in Table III are decomposed according to the time spent in two major steps: (i) the evaluation of the derivatives of S_1 for all the pairs of bubbles and the filling of the matrix A_{ij}; and (ii) the solution of the system of linear equations using a Gaussian elimination routine. As mentioned earlier, the time for solving the system of equations is negligible for the values of N examined in the table, even though the Gaussian elimiation is an $O(N^3)$ algorithm. The times for the $O(N)$ algorithms are given for the time required per iteration and for the

TABLE III
Computational Time in Seconds for Computing the Derivatives of S_1 and Filling in the Coefficients of Matrix A_{ij} (i), for Solving the Linear Equations by Gaussian Elimination (ii), and for the Downward Pass (iii) and the Total Time Needed for Solving the System of Equations (iv)

N	16	32	64	128	256
(i)	4.7	21.4	89.1	369	1464
(ii)	0.44E-2	0.017	0.093	0.60	4.31
Total	4.7	21.4	89.2	369	1469
N	512	1024	4096	8192	
(iii)	4.7	8.8	46.1	83.3	
(iv)	19.5	35.9	185	326	
Total	29.1	50.3	240	425	

Note. (i) and (ii) correspond to the $O(N^2)$ algorithm and (iii) and (iv) to the $O(N)$ algorithm.

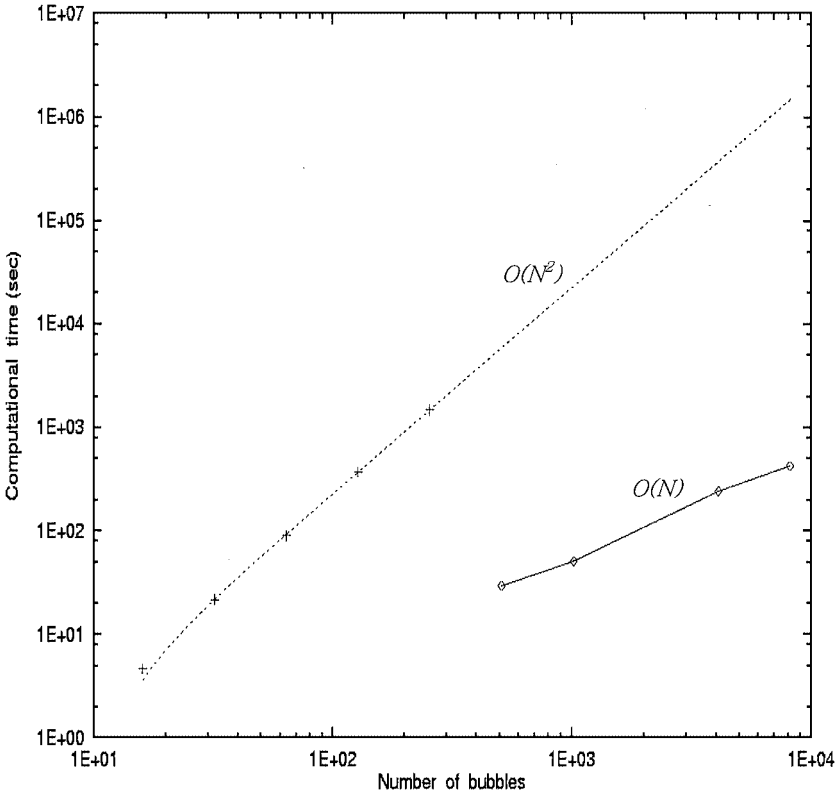

FIG. 3. Computational time in seconds required by the $O(N^2)$ and $O(N)$ algorithm.

total time required, which roughly equals the time taken per one iteration times the number of iterations. All the times shown in Table III correspond to the wall-clock time using a single processor on an IBM SP machine at Cornell Theory Center. Figure 3 compares the total time required by the $O(N)$ algorithm with the time required by the $O(N^2)$ algorithm for evaluating the derivatives of S_1 for all pairs of bubbles and for filling the matrix (i.e., the total time minus the time required for solving the system of equations). The results in Table III were extrapolated assuming an $O(N^2)$ dependence of the computational time. We see that the $O(N)$ algorithm takes about 200 times less time than the $O(N^2)$ algorithm for $N = 512$. For $N = 4096$, the savings is by a factor of 20,000!

The results presented above correspond to the single processor. Considerable wall-clock time savings can be gained with the use of multiple processors that are available on machines such as the IBM SP2. We used a masterworkers model. The unit cell was divided into a number of domains equal to the number of worker processors. The upward pass was performed by the master processor while the downward pass and bubble–bubble interaction calculations were done by the worker processors. Once the workers completed the calculations of A^{α} and $\mathbf{C}^{\alpha}$, the results were sent to the master processor. The DGMRES routine was written for a single processor and hence updating of the guesses for X_j was done by the master processor. Figure 4 shows the reduction in the wall-clock time achieved through the use of multiple processors:

$$\text{Speedup} = \frac{\text{Wall-clock time with certain number of processors}}{\text{Wall-clock time for a single processor}}. \tag{36}$$

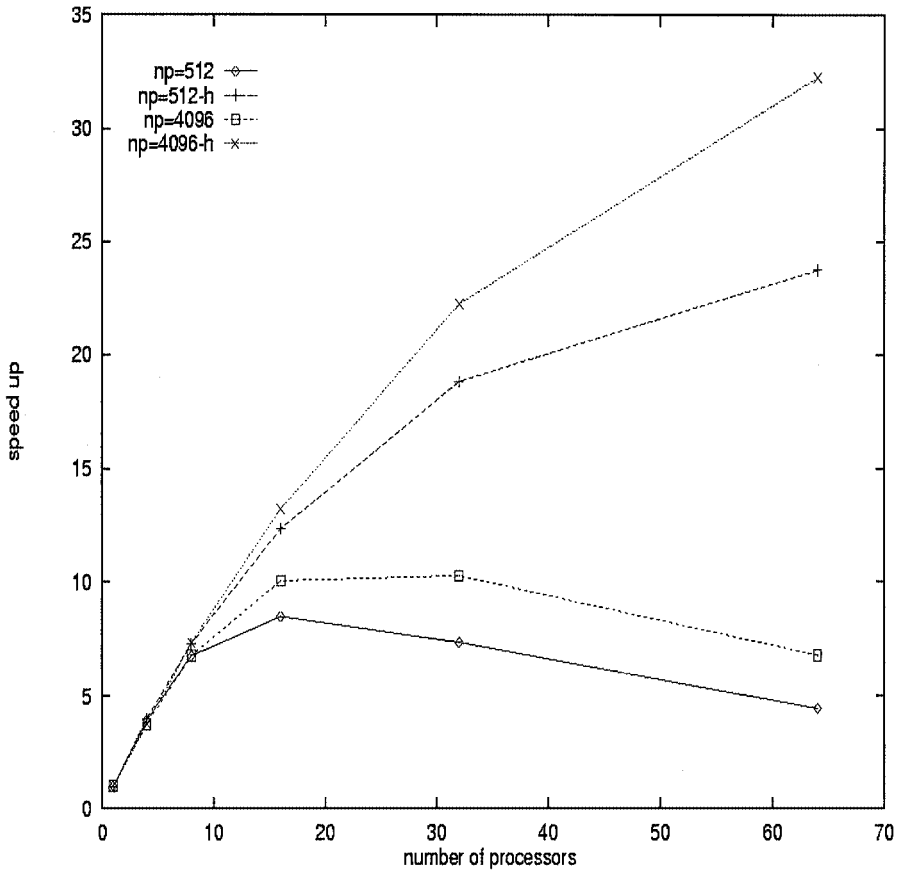

FIG. 4. Speedup as a function of number of processors.

As seen in Fig. 4, the speedup is very different depending on the communication algorithm, owing to the significant time needed for communication among processors, especially for computations using large numbers of processors. The diamonds and squares are obtained with no scheduling of communication. The algorithm for efficient communication scheduling among processors is thus necessary to improve speedup as the number of processors increases. The points represented by crosses in Fig. 4 employed the hypercube algorithm. When N is small it is not efficient to use many processors, since the time required for communicating between the master and the worker processors becomes more significant. However, we note that very significant savings, by a factor as great as 30, can be achieved with 64 processors for $N = 4096$. This suggests that by using 64 processors the velocity field and the potential interaction force for a system of 5000 bubbles can be determined in about 8 s of wall-clock time.

5. CONCLUSIONS

An efficient algorithm for computing Laplace interactions among many bubbles is described. The hierarchical grouping of bubbles is used for evaluating the field induced by a group of bubbles in a box. It is shown that the properties of bubbly liquids obtained from this

method are in excellent agreement with the ones obtained using the $O(N^2)$ algorithm. The $O(N)$ algorithm will make it possible to carry out dynamic simulations involving several thousand bubbles. Additionally, the computational time can be reduced by parallelizing the algorithm and using proper communication schedule.

ACKNOWLEDGMENT

S.-Y. Kang acknowledges the support by the BK21 program.

REFERENCES

1. A. Biesheuvel and L. van Wijngaarden, Two-phase flow equations for a dilute dispersion of gas bubbles in liquid, *J. Fluid Mech.* **148**, 301 (1984).
2. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* **73**, 325 (1987).
3. H. Hasimoto, On the periodic fundamental solutions of the Stokes equations and their application to viscous flow past a cubic array of spheres, *J. Fluid Mech.* **5**, 317 (1959).
4. S.-Y. Kang, A. S. Sangani, H.-K. Tsao, and D. L. Koch, Rheology of dense bubble suspension, *Phys. Fluids* **9**, 1540 (1997).
5. D. W. Moore, The boundary layer on a spherical gas bubble, *J. Fluid Mech.* **16**, 161 (1963).
6. A. S. Sangani and A. K. Didwania, Dynamic simulations of flows of bubbly liquids at large Reynolds numbers, *J. Fluid Mech.* **250**, 307 (1993).
7. A. S. Sangani and A. K. Didwania, Dispersed-phase stress tensor in flows of bubbly liquids at large Reynolds number, *J. Fluid Mech.* **248**, 27 (1993).
8. A. S. Sangani and G. Mo, An $O(N)$ algorithm for Stokes and Laplace interactions of particles, *Phys. Fluids* **8**, 1990 (1996).
9. P. Smereka, On the motion of bubbles in a periodic box, *J. Fluid Mech.* **254**, 79 (1993).
10. P. D. M. Spelt and A. S. Sangani, Properties and averaged equations for flows of bubbly liquids, *Appl. Sci. Res.* **58**, 337 (1998).
11. Y. Yurkovetsky and J. F. Brady, Statistical mechanics of bubbly liquids, *Phys. Fluids* **8**, 881 (1996).
12. D. Z. Zhang and A. Prosperetti, Averaged equations for inviscid disperse two-phase flow, *J. Fluid Mech.* **267**, 185 (1994).