Journal of Computational Physics **167,** 196–216 (2001)
doi:10.1006/jcph.2000.6667, available online at http://www.idealibrary.com on IDEAL®

Physalis: A New *o*(*N*) Method for the Numerical Simulation of Disperse Systems: Potential Flow of Spheres

A. Prosperetti[1] and H. N. Oğuz

Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, Maryland 21218

Received March 20, 2000; revised October 19, 2000

This paper presents a new approach to the direct numerical simulation of potential problems with many spherical internal boundaries, e.g., many spheres in potential flow. The basic idea is to use a local analytic representation valid near the particle and to match it to an external field calculated by a standard finite-difference (or finite-element) method. In this way the geometric complexity arising from the irregular relation between the particle boundary and the underlying mesh is avoided and fast solvers can be used. The results suggest that the computational effort increases less than proportionally to the number of particles and, additionally, that meshes that would be excessively coarse as measured in terms of particle radius in a conventional calculation can be used without significant loss of accuracy. In separate (if preliminary) work the same approach has been extended to the simulation of viscous flow about spheres and cylinders at finite Reynolds numbers. © 2001 Academic Press

Key Words: multiphase flow; potential flow; particle flow.

1. INTRODUCTION

The practical importance of disperse multiphase flows, coupled with great progress in computational hardware and software, has motivated a strong interest in the direct numerical simulation of particle flows. While a good deal of effort has been directed to low-Reynolds-number flows (see, e.g., Refs. [1–8]) and to the simulation of flows containing point-like suspended particles (e.g., Refs. [9–11]), several methods have also been developed for finite-size solid and fluid particles at finite Reynolds numbers. For example, Refs. [12–18] describe finite-element methods for the calculation of particle flows in two and three spatial dimensions; Ref. [19] describes an approach based on the so-called immersed boundary

[1] Also Faculty of Applied Physics and Twente Institute of Mechanics, University of Twente, AE 7500 Enschede, The Netherlands, and Burgerscentrum, The Netherlands.

0021-9991/01 $35.00

strategy [20], while Ref. [21] advocates solving the elasticity equations inside the particles at the same time the Navier–Stokes equations are solved in the fluid. In the recent methods described in Refs. [22, 23], suitably augmented equations are solved both inside and outside the particles, a strategy that has proven quite effective in the calculation of free surface flows (see, e.g., Refs. [24–27]). An alternative approach, the so-called CHIMERA method, consists of the use of a fixed global grid on which local grids, attached to each particle, move (see, e.g., Refs. [28, 29]).

An unquestionable strength of the methods mentioned so far is their adaptability to broad classes of solid and fluid particles, if at the cost of complexity and computational overhead. Here we present an alternative approach—less versatile because it can only be applied to particles with a simple shape, such as spheres and cylinders, but simpler and computationally efficient. This paper describes the method in its application to the simplest case, that of spheres in a potential field; some preliminary results for steady and unsteady Navier–Stokes flow around spheres and cylinders have been presented in [30].

As explained below, our numerical experience to date suggests that the computational load of the method grows less than proportionally to the number of particles N. This feature is common in methods (such as those of Refs. [24–27]) employing a fixed grid independent of the number of particles. On the other hand, for many of the existing methods (e.g., for potential [7, 31] or Stokes [7] flow), the computational effort increases at least linearly with N.

In some way, the method is similar in spirit to the CHIMERA approach but, unlike that method, it relies on an exact analytical solution in the neighborhood of each particle, rather than on a local numerical solution. One advantage is greater accuracy, as the interpolation between the fixed and moving grids necessary with the CHIMERA approach is avoided.

The basic idea underlying the present approach can be simply explained in the following terms. Enclose each particle by a surface S_Q (Fig. 1), and assume that the problem at hand is linear or can at least be approximately linearized in the region of space between the surface of the particle and S_Q. (How this can be achieved for the case of the Navier–Stokes equations is mentioned below.) Let the solution of the problem on S_Q be denoted by ϕ^Q. Then, by the linearity of the problem, there will be a linear operator $\mathcal{G}^Q$ connecting some quantity ψ related to ϕ (perhaps ϕ itself) evaluated on the particle surface to ϕ^Q,

$$\phi^Q = \mathcal{G}^Q \psi + g^Q, \tag{1}$$

where we have allowed for the possible presence of a known forcing term g^Q. Let S_P be

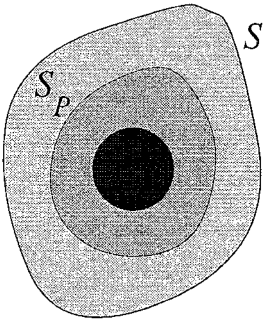

FIG. 1. Definition of surfaces S_Q and S_P enclosing a particle.

another surface enclosing the same particle and, for example, internal to S_Q (Fig. 1). For this surface we may similarly write

$$\phi^P = \mathcal{G}^P \psi + g^P. \tag{2}$$

By eliminating ψ between these two relations, we find a consistency relation between ϕ^Q and ϕ^P that we write as

$$\tilde{\phi}^P = \mathcal{G}^P (\mathcal{G}^Q)^{-1} \tilde{\phi}^Q, \tag{3}$$

where $(\mathcal{G}^Q)^{-1}$ may need to be understood as a generalized inverse and

$$\tilde{\phi} = \phi - g. \tag{4}$$

If the mathematical nature of the problem is such that the satisfaction of the consistency relation (3) ensures the correct solution in the region between the particle and S_Q, then the boundary conditions at the surface of the particles can simply be replaced by a corresponding set of relations of the type (3) to be imposed between the values of ϕ on the surfaces S_P and S_Q surrounding each particle. In practice, an obvious advantage of this approach is that the surfaces S_P, S_Q can be chosen to conform to a particular discretization of the computational domain so that the boundary condition on the particle surface—the satisfaction of which often requires irregular grids—can be simply transferred to the nodes of a regular grid.

An important conclusion of the well-posedness of an equation such as (3) stems very simply by noting that, if the original problem is well posed, $\mathcal{G}^P$ and $\mathcal{G}^Q$ depend continuously on the surfaces S_P and S_Q. Thus, if S_P is close to S_Q, the operator $\mathcal{G}^P(\mathcal{G}^Q)^{-1}$ would be close to the identity operator.

The previous description is very general and leaves many details—both mathematical and numerical—unexplained. In the rest of the paper we present a detailed exposition for the case of potential flow from which several valuable numerical features of the method will become apparent. In the last section we comment on how the approach can be extended to other problems. These extensions will be described in future articles.

We call our method Physalis.[2]

2. POTENTIAL FLOW

Consider the potential flow around a number of spherical particles or bubbles, not necessarily all with the same radius. In particular, the radius may depend on time, although, for simplicity, we do not explicitly consider this case here.

The velocity potential ϕ satisfies Laplace's equation

$$\nabla^2 \phi = 0, \tag{5}$$

subject, on the surface of each particle, to the condition

$$(\boldsymbol{\nabla}\phi - \mathbf{w}) \cdot \mathbf{n} = 0 \quad \text{on } |\mathbf{x} - \mathbf{y}| = a. \tag{6}$$

[2] According to Webster's Dictionary, this word denotes, a genus of herbs characterized by a bladdery calyx which encloses an edible fruit; also called ground cherry. In French the word also stands for "amour en cage."

Here $\mathbf{y}$ is the instantaneous position of the particle center, $\mathbf{w}$ the instantaneous particle velocity, $\mathbf{n}$ the outward unit normal, and a the particle radius.

The general solution of the problem in the neighborhood of the generic particle may be written down at once and is

$$\phi(\mathbf{x}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \left(r^{\ell} + \frac{\ell}{\ell+1} \frac{a^{\ell+1}}{r^{\ell+1}} \right) Y_{\ell}^{m}(\theta, \varphi) b_{\ell m} - \frac{a^3}{2r^3} \mathbf{w} \cdot (\mathbf{x} - \mathbf{y}), \tag{7}$$

where the position vector $\mathbf{x}$ is written as $\mathbf{x} = \mathbf{y} + \mathbf{r}$, with the vector $\mathbf{r}$ expressed in terms of spherical coordinates (r, θ, φ) centered at the center of the particle; the Y_n^m are spherical harmonics and, by varying the coefficients $b_{\ell m}$, the totality of possible potential flows around the sphere can be captured. In the absence of other boundaries, the series in (7) converges at all points $\mathbf{x}$ inside a sphere with radius equal to the distance between the centers of the given sphere and the sphere closest to it.

If (7) is evaluated at the points Q of a surface S_Q enveloping the sphere, we have a particular case of the general relation (1) in which ψ is an infinite-dimensional vector containing the coefficients b_{nm}, the operator $\mathcal{G}^Q$ is the summation with coefficients evaluated at the points of S_Q, and the function g^Q is the last term of (7) proportional to $\mathbf{w}$, also evaluated at the points of S_Q. The standard theory of spherical harmonic expansions (see e.g., Ref. [32]) ensures that, given ϕ^Q, all the coefficients b_{nm} can be uniquely calculated; for example, if S_Q is a sphere of radius R, we have

$$\left(R^{\ell} + \frac{\ell}{\ell+1} \frac{a^{\ell+1}}{R^{\ell+1}} \right) \left(Y_{\ell}^{m}, Y_{\ell}^{m} \right) b_{\ell m} = \left(Y_{\ell}^{m}, \phi + \frac{a^3}{2r^3} \mathbf{w} \cdot (\mathbf{x} - \mathbf{y}) \right), \tag{8}$$

where (,) denotes the scalar product over the unit sphere. We thus conclude that the operator $\mathcal{G}^Q$ is well defined. Analogous considerations apply to $\mathcal{G}^P$ and g^P when (7) is evaluated at the points of S_P.

The developments in this section have been explicitly written down for the case of the Neumann problem (6). It is obvious that a similar treatment can be followed for Dirichlet or mixed boundary conditions.

3. NUMERICAL FORMULATION

We now describe a numerical method for implementing the basic idea outlined at the end of Section 1. The particular implementation we describe is based on finite differences which, on regular domains, give rise to algebraic systems that can be solved noniteratively with fast solvers. On irregular domains it might be advantageous to use a finite-element discretization to which the present method can be adapted equally well. For brevity, however, we do not consider this possibility further.

The first step is to discretize the surfaces S_P and S_Q by approximating them by “cages” consisting of a finite number of points N_P and N_Q, respectively. In the finite-difference formulation that we use, these points will be taken to belong to the regular grid into which the computational domain is discretized. Figure 2 shows square (or cubic) cages of grid points, but there could be good reasons to use other shapes, e.g., in the Navier–Stokes case. Next, we truncate the summation in (7) to a maximum value L of the index ℓ, which has the effect of retaining only the first $L(L+2)$ coefficients $b_{\ell m}$. We assemble these remaining

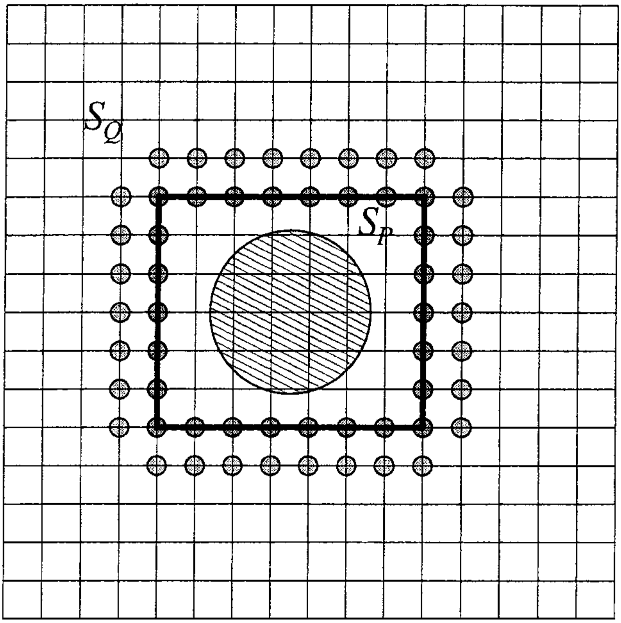

FIG. 2. Discretization of surfaces S_Q and S_P enclosing a particle into N_P and N_Q points.

coefficients in a vector $\mathbf{b} = \{b_k\}$, $k = 1, 2, \ldots, L(L+2)$, where the index k is shorthand for the pair of indices (ℓ, m). We assume that $N_Q \geq L(L+2)$.

We now introduce an ordering of the outer cage points, denoted by $Q_1, Q_2, \ldots, Q_{N_Q}$, and a corresponding ordering for the points $P_1, P_2, \ldots, P_{N_P}$ of the inner cage. With this discretization, the operator $\mathcal{G}^Q$ is approximated by the $N_Q \times L(L+2)$ matrix G^Q with elements

$$\mathsf{G}^Q_{hk} = \left(r_h^\ell + \frac{\ell}{\ell+1} \frac{a^{\ell+1}}{r_h^{\ell+1}} \right) Y_\ell^m(\theta_h, \varphi_h), \tag{9}$$

where $(r_h, \theta_h, \varphi_h)$ are the coordinates of the hth point of the outer cage with respect to the particle center and $k \equiv (\ell, m)$. Note that this matrix only depends on the relative position of the sphere center with respect to the points of the outer cage. In a similar way, the function g^Q has been discretized into an N_Q-dimensional vector $\mathbf{g}^Q$ with components

$$g_h^Q = -\frac{a^3}{2r_h^3} \mathbf{w} \cdot (\mathbf{x}_h - \mathbf{y}). \tag{10}$$

With this discretization Eq. (1) becomes

$$\phi^Q = \mathsf{G}^Q \mathbf{b} + \mathbf{g}^Q. \tag{11}$$

An effective way to find the generalized inverse of G^Q is to use the singular-value decomposition (see, e.g., Refs. [33, 34])

$$\mathsf{G}^Q = \mathsf{U}\mathsf{D}\mathsf{V}^T, \tag{12}$$

where D is a square $L(L+2) \times L(L+2)$ diagonal matrix, V is another $L(L+2) \times$

$L(L+2)$ square matrix, and U is $N_Q \times L(L+2)$; these matrices satisfy

$$\mathsf{U}^T\mathsf{U} = \mathsf{I}, \quad \mathsf{V}\mathsf{V}^T = \mathsf{I}, \tag{13}$$

where I denotes identity matrices of suitable dimension. By using these properties, we readily find

$$(\mathsf{G}^Q)^{-1} = \mathsf{V}\mathsf{D}^{-1}\mathsf{U}^T, \tag{14}$$

and we may write (3) in the form

$$\tilde{\phi}^P = \mathsf{B}\tilde{\phi}^Q, \tag{15}$$

where the $N_P \times N_Q$ matrix B is given by

$$\mathsf{B} = \mathsf{G}^P\mathsf{V}\mathsf{D}^{-1}\mathsf{U}^T, \tag{16}$$

and

$$\tilde{\phi} = \phi - \mathbf{g}. \tag{17}$$

Equation (15) is the discrete version of the consistency condition (3) and, as noted near the end of Section 1, the matrix (16) may be expected not to differ too much from the identity matrix and to be therefore well-conditioned. This expectation is borne out by our numerical experience.

Following the plan outlined in Section 1, it is now necessary to solve the Laplace equation on the domain imposing relations of the form (15) on the nodes that constitute the cages attached to each particle. For this purpose we let

$$\mathsf{L}\phi = 0 \tag{18}$$

be a finite-difference discretization of the Laplace equation (5). Then the conditions (3) are imposed iteratively solving

$$\mathsf{L}\phi^{j+1} = \mathsf{M}\mathbf{f}^j, \tag{19}$$

where the superscript denotes the iteration number, M is a projector on the P-nodes of the inner cages, and

$$\mathbf{f} = \mathsf{L}\phi + \beta C(\tilde{\phi}^P - \mathsf{B}\tilde{\phi}^Q), \tag{20}$$

where β is a relaxation factor (chosen to be around 0.1) and C the common diagonal element of L. As j increases, the solution of (19) tends to the solution of (18) at all nodes other than those belonging to the inner cage. For these latter nodes, at convergence, one has

$$\mathsf{L}\phi = \mathsf{L}\phi + \beta C(\tilde{\phi}^P - \mathsf{B}\tilde{\phi}^Q), \tag{21}$$

i.e., (15). Since the satisfaction of this relation also implies that of the Laplace equation, we thus conclude that the procedure generates a discrete approximation to the solution of the Laplace equation over the entire computational domain. For the solution of the system (19) we use a fast solver; the iterative procedure—which is in effect the way in which information is exchanged between the cages and the surrounding flow—is only needed to implement the constraint (15).

After convergence, the potential at the grid nodes outside the cages is given by the solution of (19), while, in the inner regions between each particle and the cage, it is represented by the spherical harmonic expansion (7) with coefficients that can be calculated from (12); by evaluating (7) at all the grid nodes in the inner region and on the cage, one then completes the calculation of the potential at all the grid nodes exterior to the particles.

As a final comment, it may be noted that the method contains in some sense a built-in accuracy check in that the very fact that convergence of the iterative process occurs ensures that the potential is accurately represented on **both cages** by the **same** set of coefficients. This feature implies that, within the error limits set for convergence, the local expansion and the outer solution match in the region between the two cages.

In the examples that we consider here the computational domain is regular and one can take advantage of the existence of fast Poisson solvers for such domains to considerably increase the efficiency of the computation. For these solvers to work effectively, it is necessary to solve (19) everywhere and, therefore, also inside the cages surrounding the particles. The solution of the Laplace equation generated in this way is simply the solution of a regular potential flow satisfying given boundary conditions on the cage boundary. Although not relevant for the problem at hand, this is a well-defined mathematical entity. If algorithms other than fast solvers are used, then the regions interior to the cages may of course be skipped.

4. OVERLAPPING CAGES

When two or more spheres are close, it may happen that the respective cages overlap (Fig. 3). We now describe a procedure for handling this situation. The details are tedious but straightforward and an abbreviated description will be sufficient.

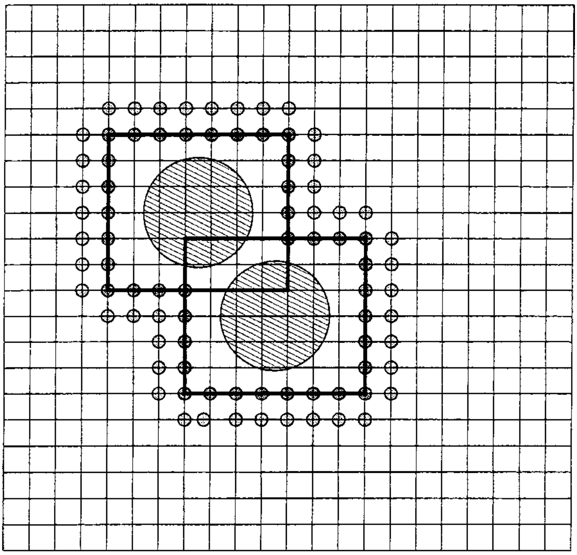

FIG. 3. Discretization of surfaces S_Q and S_P enclosing two particles.

Let there be p particles inside the surface (cage) S_Q. Then it can easily be proven, e.g., by a series expansion of Green's formula as shown in the Appendix, that inside the cage the potential may be represented as

$$\phi = \sum_{\alpha=1}^{p} \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{1}{r_\alpha^{\ell+1}} Y_\ell^m(\theta_\alpha, \varphi_\alpha) d_{\ell m}^\alpha + \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} r^\ell Y_\ell^m(\theta, \varphi) e_{\ell m}, \tag{22}$$

where $(r_\alpha, \theta_\alpha, \varphi_\alpha)$ are spherical coordinates centered at the center of the αth sphere and (r, θ, φ) are spherical coordinates centered, e.g., at the centroid of the volume bounded by S_Q. By discretizing the problem as before, we may rewrite this equation as

$$\phi = \mathsf{F}\mathbf{d} + \mathsf{H}\mathbf{e}, \tag{23}$$

where the vectors $\mathbf{d}$, $\mathbf{e}$ contain all the coefficients $\{d_{\ell m}^\alpha\}$, $\{e_{\ell m}^\alpha\}$ for all the particles, and the matrices F, H are analogous to G defined in (9).

In the neighborhood of each sphere the potential may be represented analogously to (7) used before:

$$\phi^{(\alpha)} = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} r_\alpha^\ell Y_\ell^m(\theta_\alpha, \varphi_\alpha) b_{\ell m}^\alpha + \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{1}{r_\alpha^{\ell+1}} Y_\ell^m(\theta_\alpha, \varphi_\alpha) c_{\ell m}^\alpha. \tag{24}$$

Of course, in the case of the Neumann problem (6), this expansion takes the form (7) but here, for the sake of generality, we do not commit ourselves to the Neumann problem. We merely note that the boundary condition on the surface of the spheres will lead to a linear relation of the type

$$\mathbf{d} = \mathsf{C}\mathbf{c} + \mathbf{f}, \tag{25}$$

where the vector $\mathbf{c}$ contains the coefficients $\{c_{\ell m}^\alpha\}$ and $\mathbf{f}$ will depend on the known quantities prescribed on the sphere, such as the translational velocity $\mathbf{w}$ of the previous section.

Our objective is to derive a relation analogous to (1) valid on the outer cage, after which the same procedure as before can be followed. For this purpose we start by noting that the problem of relating the coefficients $\{d_{\ell m}^\alpha\}$, $\{e_{\ell m}\}$ to the $\{b_{\ell m}^\alpha\}$, $\{c_{\ell m}^\alpha\}$ is solved in Ref. [32] (see also Ref. [35]) and leads to a relation of the form (see the Appendix)

$$\mathbf{c} = \mathsf{D}\mathbf{d} + \mathsf{E}\mathbf{e}, \tag{26}$$

where D, E are suitable (infinite) matrices. By using this relation to eliminate $\mathbf{c}$ from (25) and solving for $\mathbf{d}$ we find

$$\mathbf{d} = (\mathsf{I} - \mathsf{CD})^{-1}(\mathsf{CE}\mathbf{e} - \mathbf{f}), \tag{27}$$

which, when substituted into (23) evaluated at the points Q, leads to

$$\phi^Q = \mathsf{G}^Q\mathbf{e} + \mathbf{g}^Q, \tag{28}$$

with

$$\mathsf{G} = \mathsf{F}(\mathsf{I} - \mathsf{CD})^{-1}\mathsf{CE} + \mathsf{H}, \quad \mathbf{g} = -\mathsf{F}(\mathsf{I} - \mathsf{CD})^{-1}\mathbf{f}. \tag{29}$$

The relation (28) is analogous to (1) in that it connects the values of ϕ on the outer cage to the set of coefficients appearing in that part of ϕ that is regular inside the cage. From this point on, the same method described in the previous section can be followed for the numerical solution. Our numerical experience shows that the matrices introduced in this section are well behaved and the required inverses exist.

5. NUMERICAL DETAILS

The method is implemented in a box-shaped domain of dimensions X, Y, and Z uniformly discretized by means of $N_x + 1$, $N_y + 1$, and $N_z + 1$ grid points in the three orthogonal directions. The grid spacings are given by $\Delta x = X/N_x$, $\Delta y = Y/N_y$, and $\Delta z = Z/N_z$; in the numerical examples that follow we have varied X, Y, and Z while maintaining $\Delta x = \Delta y = \Delta z$.

We start with a set of N spherical particles, each with a velocity $\mathbf{w}_\alpha = (u_\alpha, v_\alpha, w_\alpha)$, $\alpha = 1, 2, \ldots, N$, centered at given positions $\mathbf{y}_\alpha = (x_\alpha, y_\alpha, z_\alpha)$. In principle the method can be applied to systems with different, possibly time-dependent particle sizes, but, for the examples considered in this paper, all particles have the same fixed radius a.

For each particle a “cage” must be constructed. For ease of coding and efficiency of computation in this paper we use cubic cages, but this choice is not essential. In particular, it proves undesirable for flow at finite Reynolds numbers for the reasons mentioned below (see also [30]). If $\mathbf{x}_\alpha$ is the grid point closest to the center of the αth particle and corresponding to indices $\mathbf{i}_\alpha = (i_\alpha, j_\alpha, k_\alpha)$, the cage is constituted by the grid points with indices $i_\alpha - \frac{1}{2}N_c \le i \le i_\alpha + \frac{1}{2}N_c$, and similarly for the other directions. The even integer N_c is large enough that the particle fits comfortably inside the cage; it ranges from 6 for low-resolution grids with $a/\Delta x = 1$ to 24 for larger cages or fine-resolution grids with $a/\Delta x = 4$.

The six faces of the cube define the inner cage S_P described in the previous section. The outer cage S_Q is constructed from the inner one by displacing the six faces outward by a certain number of grid spacings. This number is normally chosen to be 1, as in the case shown in Fig. 2, although we have also tested cases with 2 or more grid spacings without noticing any significant difference in the results. It is important to stress that the need to impose proper boundary conditions on the outer solution dictates that the inner cage form a complete (if discretized) closed surface. However, the outer cage is only used to determine the expansion coefficients by collocation according to Eq. (11), and, therefore, it only needs to have a sufficient number of grid points that this equation be solvable. Hence it is expected that “gaps,” or openings (e.g., at the corners), would not adversely affect the computation, which conforms to our numerical experience.

With P (inner) and Q (outer) points defined in this fashion the matrices G^P and G^Q can be constructed and combined to form the matrix B defined in (16). As noted before, for a given cage shape and size, these matrices depend only on the particle radius and the location of the particle center with respect to the grid points of the mesh. If, for example, all the particle centers happened to coincide with grid points, all the B-matrices for each particle would be identical. Of course, this situation would not normally occur, and, in principle, these matrices would need to be computed at each time step for each particle depending on the position of that particle with respect to the fixed grid. With a large number of particles this requirement can be an excessive burden in terms of both computation and memory. To alleviate this problem we use interpolation as follows. At the beginning of the calculation we compute the B-matrices for a certain number of positions of the particle center with respect to the grid, and then we effect a three-dimensional interpolation to approximate the

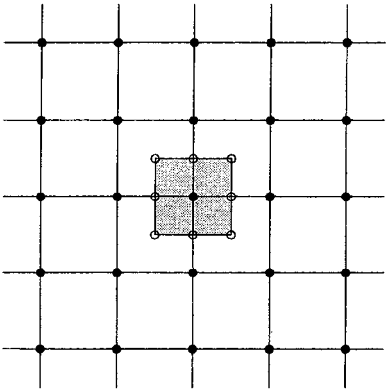

FIG. 4. The cage is centered at the center of the shaded region whenever the particle center is anywhere in this region. The open circles (and the analogous points in the third dimension) are the points corresponding to which a B-matrix is computed. The B-matrices for other particle positions are obtained by interpolation. In three dimensions the procedure requires the computations of a total of 27 B-matrices.

B-matrix appropriate for the actual position of each particle. If we define $\mathbf{r}_\alpha = \mathbf{y}_\alpha - \mathbf{x}_\alpha$, then it is clear that all the possible arrangements of the particle with respect to the grid correspond to the range $-\frac{1}{2}\Delta x < r_x < \frac{1}{2}\Delta x$, and similarly for the other two directions. Hence we pick N_{int} regularly spaced points in a cube of sides Δx, Δy, Δz centered at a grid point and calculate B-matrices for each one of these N_{int} positions (Fig. 4). With this information, a look-up table of B-matrices is constructed from which each element of the B-matrix corresponding to any particle position is obtained by linear interpolation. In the examples that follow this idea was implemented by taking $N_{int} = 27$.

When two particles are so close to each other that their inner cages touch or overlap, the code automatically switches to the scheme described in Section 4 and merges the two cages. The points of the original cages that lie inside the merged cage are removed and a new B-matrix is computed from (24) rather than from (11). In this case the number of possible geometric arrangements is too large for an interpolation scheme to be efficient and it is best to compute the matrices directly. For the cases described in this paper this difficulty has not been a major drawback since the number of overlapping cages was usually small.

The Poisson equation (19) has been solved by means of a fast solver from the package FISHPACK freely available from `http://netlib.org`.

6. ILLUSTRATIVE RESULTS

To validate the proposed new method we have made many comparisons between the results of the present method and those given by the multipole expansion method (see, e.g., Refs. [35, 36]), which is based essentially on the use of the representation (22) for the potential. In addition to ensuring the agreement of the two sets of results, it was important to demonstrate that the present new results are independent of the cage size. The comparison has also enabled us to assess the computational efficiency of Physalis with respect to the multipole method.

It should be noted that, while for Physalis generally we have used periodic boundary conditions at the boundary of the computational domain, the computational domain for the multipole method has been assumed to be unbounded. It is well known that, by using

Hasimoto's method (see, e.g., Refs. [36–39]), a multipole method can be constructed for a periodic system, but this approach leads to much lengthier computations and for simplicity we have not pursued it.

The first set of results that we discuss is for the simplest case of two spheres arranged along the z-axis and having a unit velocity in the direction of the positive z-axis (Figs. 5–7). The situation is evidently axisymmetric in this case and all coefficients $b_{\ell m}$ with $m \neq 0$ vanish. The inner cage is a cube with linear dimensions $3a$. The size of this cage is relatively large but was used in this example to better assess the performance of the overlapping cages algorithm and the switch from two-particle to single-particle cages. The computational domain consists of $10 \times 10 \times 30$ cells.

In Fig. 5 we show the coefficients b_{10} and b_{20} of the Legendre polynomials P_1 and P_2 of the inner expansion as functions of the distance d/a between the particle centers. These are the coefficients of the upstream sphere; those for the downstream sphere behave similarly. The lines are the multipole method results with $L = 4$ and $|m| \leq 4$. The black circles are

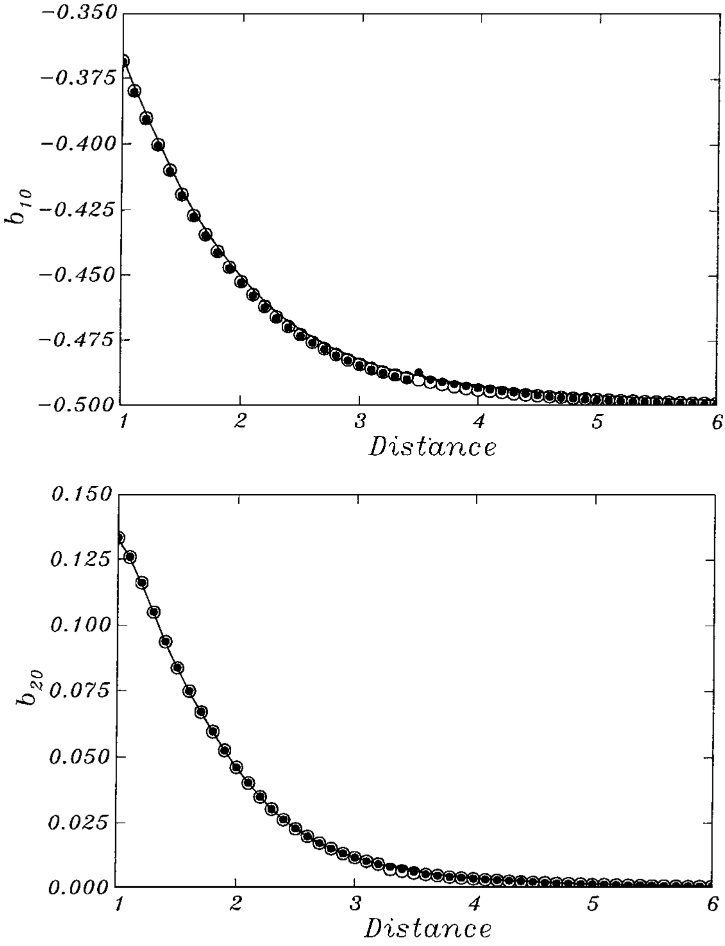

FIG. 5. The coefficients of P_1 and P_2 as a function of the distance between two particles moving in the same direction, as computed by the multipole expansion method in an unbounded domain (solid line) and the current method in a $10 \times 10 \times 30$ periodic Cartesian domain; open circles: $a/\Delta x = 4$; black circles: $a/\Delta x = 2$. The cage is a $(3a)^3$ cube.

the present result for a coarse grid with $a/\Delta x = 2$, while the open circles are for a finer grid with $a/\Delta x = 4$. It should be stressed that here the diameter of the particle relative to the mesh spacing is so large that, with a standard finite-difference method, the calculation would be grossly underresolved.

Figure 5 demonstrates an excellent agreement for b_{20} even for the case in which the grid spacing Δx is as large as half the particle radius (black circles). In this case, there are at most only 15 grid points inside or on the surface of the particle.

The comparison shown in Fig. 5 also demonstrates the insensitivity of the results to the switch from the combined to the individual cages. For both grids, overlapping ceases when the distance between the particles is greater than $3.5a$. The only evidence of this transition is a slight irregularity in the coarser-grid result for b_{10} at this distance. Here as well as in all the other cases that we have tested, this irregularity disappears with a smaller Δx.

The comparison shown in Fig. 6 is for the same situation, except that here, instead of calculating the matrix B exactly for each position of the spheres, we have used the

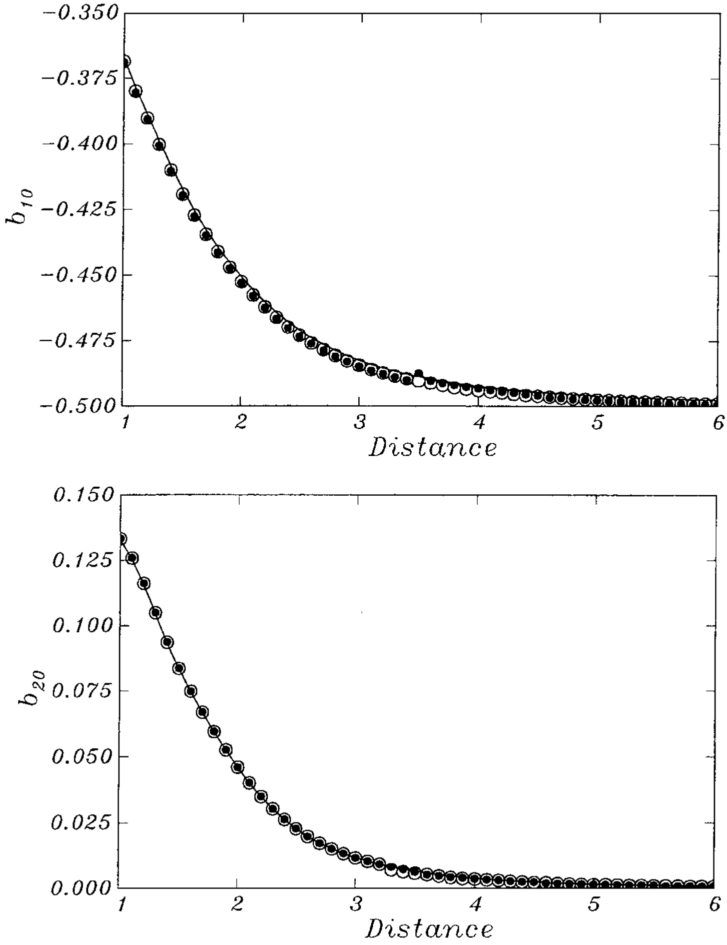

FIG. 6. The coefficients of P_1 and P_2 as a function of the distance between two particles of radius $a = 1$ moving in the same direction, as computed by the multipole expansion method in an unbounded domain (solid line) and the current method in a $10 \times 10 \times 30$ periodic Cartesian domain; open circles: $a/\Delta x = 4$; black circles: $a/\Delta x = 2$. The cage is a $(3a)^3$ cube. These results differ from those of Fig. 5 in that matrix interpolation was used.

interpolation scheme with 27 precomputed matrices mentioned in Section 5. It is evident from these plots that the performance of the method is not degraded by the approximation. The full significance of this conclusion can be better appreciated when a large number of particles—e.g., 1000 or more—are present in the system, as in this case its adoption affords a substantial memory saving without compromising accuracy.

It is interesting to note that a distance $d/a = 2$ corresponds to the two spheres touching. Nevertheless, as shown in both Figs. 5 and 6, we can generate accurate results even for $d/a < 2$, i.e., when the spheres are interpenetrating. This result is at first surprising, but it is mathematically very obvious and simply corresponds to the analytic continuation of the local expansions (7) inside the spheres, which is valid provided the distance between the centers is greater than the sphere radius. This property can be useful to approximate the shapes of other objects by "gluing together" several spheres.

Another interesting possibility suggested by this result is the use of cages contained **totally inside** the particles. We have run a limited number of tests of this idea and we have found it to work. The only negative is that, unless the mesh is sufficiently fine, there may not be enough nodes inside the particle to determine a sufficient number of coefficients $b_{\ell m}$. Nevertheless, this is a potentially useful option that would render the multiple-cage algorithm unnecessary and which we plan to pursue in future work.

Similar results for two spheres moving in opposite directions, again along the z-axis, are shown in Fig. 7. Our first calculations of this situation showed a somewhat greater discrepancy (about 0.3%) between the Physalis and the multipole results. When we changed the conditions imposed at the boundary of the computational domain from periodic to Dirichlet ($\phi = 0$), the error decreased substantially; it is this latter solution that is shown in Fig. 7. In this case the results are more sensitive to the grid spacing than before: the finer mesh with $a/\Delta x = 4$ gives results virtually indistinguishable from the multipole ones while those produced by the coarser grid with $a/\Delta x = 2$ are slightly worse. In this case it is not possible to obtain convergence when the two spheres interpenetrate.

We have carried out similar comparisons with different cages, different computational domain sizes, and a different number of coefficients (up to $L = 8$) retained in the local expansion (7). In all cases we have found comparable or better results.

Moving on to more complex situations, we now summarize the results of extensive simulations carried out with particle numbers ranging from 20 to 5000. The particle configurations were generated by inserting one sphere at a time in the computational domain. A triplet (x, y, z) of random numbers was obtained from a standard random number generator with equal probabilities in the three orthogonal directions. If the distance from the point (x, y, z) to the center of the nearest particle was greater than a specified minimum (taken as $2.2\,a$), a new sphere was placed at that location. Otherwise, a new triplet was obtained and the process was continued until all the particles were positioned. Since we only considered low-volume fractions, this simple method worked reliably and efficiently.

The calculations were carried out in a cubic domain with a Cartesian grid uniformly discretized with $128 \times 128 \times 128$ nodes. Periodicity conditions were imposed at the boundary. Each cage consisted of $4 \times 4 \times 4$ nodes per particle with a mesh spacing $a/\Delta x = 1$, i.e., two mesh lengths per particle diameter. All particles were assigned a unit velocity in the z direction. As before, for each configuration, the performance of Physalis is compared with that of the multipole method.

In Figs. 8 to 10 we show plots of the CPU time on a 600-MHz Pentium workstation with a main memory of 256 MB as a function of the number of particles for the two methods. The three figures differ in the number of terms kept in the local solution (7) around each particle:

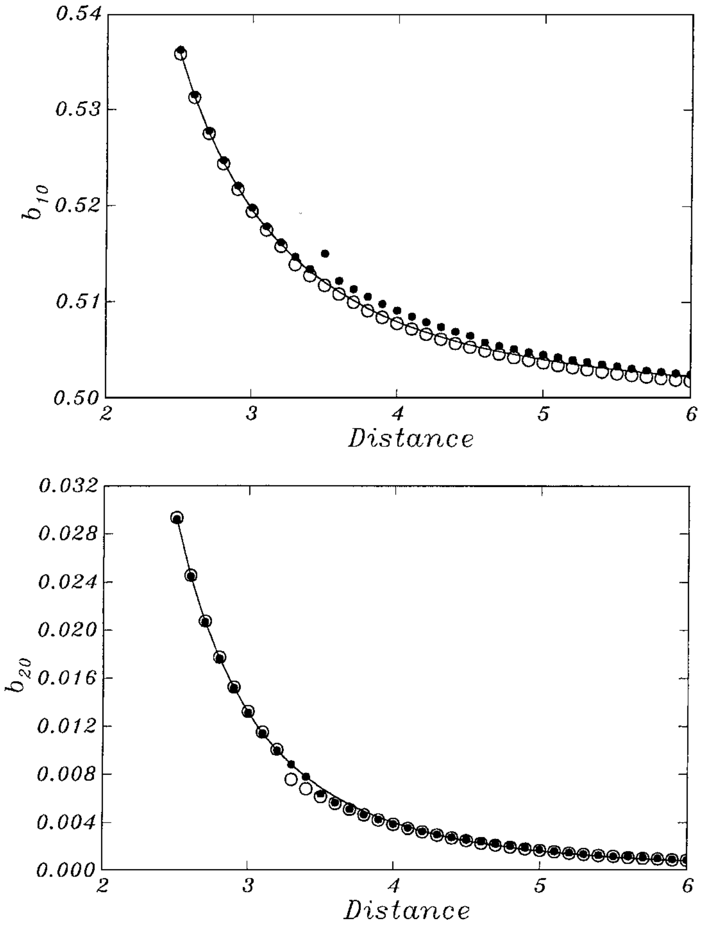

FIG. 7. The coefficients of P_1 and P_2 as a function of the distance between two particles moving in opposite directions, as computed by the multipole expansion method in an unbounded domain (solid line) and the current method in a $10 \times 10 \times 30$ Cartesian domain; open circles: $a/\Delta x = 4$; black circles: $a/\Delta x = 2$. The cage is a $(3a)^3$ cube.

Fig. 8 is for $L = 1$ (i.e., 3 coefficients), Fig. 9 for $L = 2$ (8 coefficients), and Fig. 10 for $L = 4$ (24 coefficients); in all cases the second index $|m|$ ranges from 0 up to L.

The figures show that, for a relatively small particle number, the multipole method is much faster; this feature is a consequence of the fixed overhead associated with a large discretized domain that can accommodate up to 5000 particles. An interesting feature of the comparison, however, is the large difference in the slope of the lines, which makes Physalis more efficient as the number of particles is increased above a certain level which decreases as the accuracy requirements (as dictated by the maximum order of multipoles retained) are increased. Thus, for $L = 1$, the CPU times are comparable for about 600 particles, but for $L = 4$, Physalis becomes advantageous for 100 particles.

With the multipole method, both the CPU time and the size of the calculation increase with the particle number and eventually the simulations fail due to lack of memory. Physalis, on the other hand, seems not to be affected as much and exhibits an increase in the computational effort that, in the range investigated, grows less than linearly with the number of particles. The factor most responsible for this favorable scaling (and in particular the memory saving)

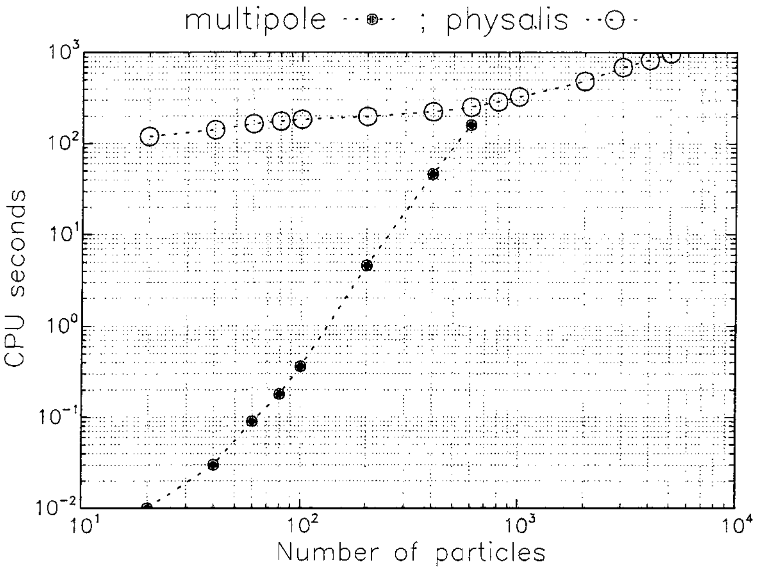

FIG. 8. Comparison of the CPU time required for the multipole expansion method and Physalis. Here $L = 1$, so that each particle is represented by 3 coefficients.

is the cage interpolation scheme described in Section 5. Another contributing factor is the use of a direct Poisson solver which, for a given grid, takes about the same CPU time per iteration regardless of the number of particles in the system. The CPU time is also seen to be weakly dependent on the order of the local expansion, in contrast to the multipole method. Finally, the results displayed in Figs. 8 to 10 show that the number of iterations is substantially unaffected when more terms in the local expansion (7) are retained. As the number of particles increases there is a proportional increase in the number of matrices B that need to be evaluated and a slowing of the rate of convergence of the iterative scheme (19).

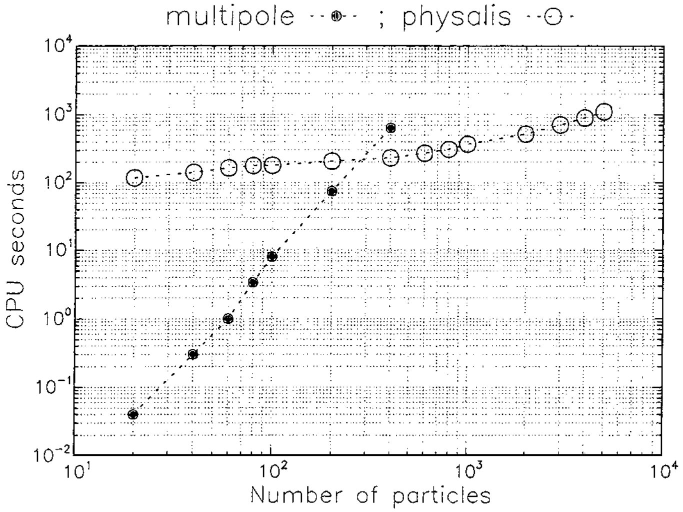

FIG. 9. Comparison of the CPU time required for the multipole expansion method and Physalis. Here $L = 2$, so that each particle is represented by 8 coefficients.

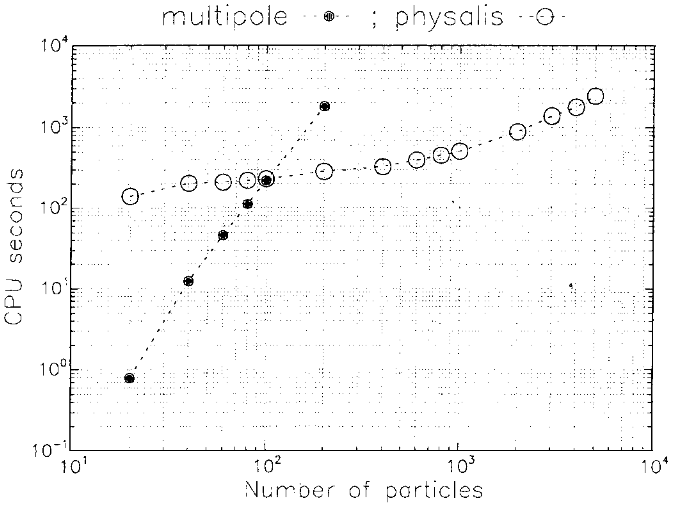

FIG. 10. Comparison of the CPU time required for the multipole expansion method and Physalis. Here $L = 4$, so that each particle is represented by 24 coefficients.

Having illustrated the computational performance of the new method, we now turn to a consideration of its accuracy in a practical situation with a large number of particles. For these calculations the Cartesian computational domain is reduced to a size of $(64a)^3$ with all the particles randomly distributed in an inner $(32a)^3$ cube. This arrangement minimizes the differences due to the boundary conditions (periodic vs unbounded) imposed on the two methods. We have discretized this domain with 128^3 grid points yielding a grid spacing of $a/\Delta x = 2$. We retain multipoles up to $L = 4$ in the local solution as well as in the multipole method expansion. As before, all particles have a unit velocity in the z-direction.

We have tested a large number of distributions and the results are available at `http://plesset.me.jhu.edu/physalis`. Here we only show comparisons of the quadrupole

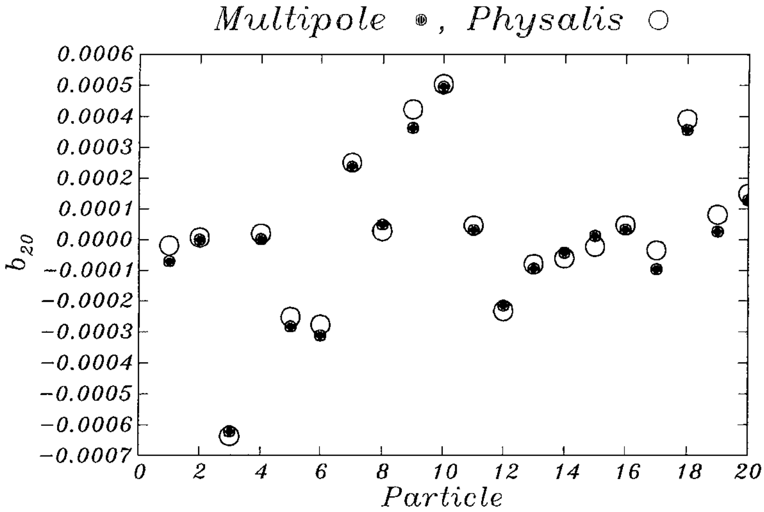

FIG. 11. Comparison of the coefficient b_{20} as computed by Physalis and the multipole method for a system of 20 particles.

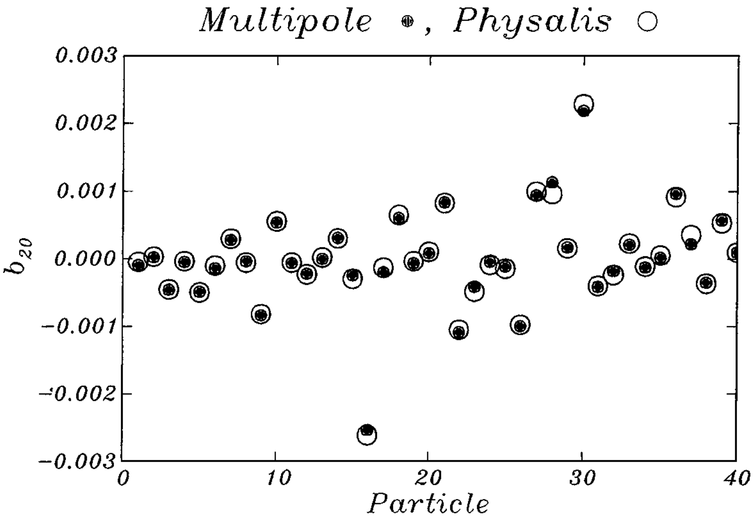

FIG. 12. Comparison of the coefficient b_{20} as computed by Physalis and the multipole method for a system of 40 particles.

coefficients b_{20} (cf. Eq. (7)) for simulations with 20, 40, and 60 particles, corresponding to particle volume fractions of 0.256, 0.512, and 0.768%. Figure 11 shows the coefficients b_{20} for each particle in a 20-particle system; the Physalis results are shown by the open circles and the multipole ones by the black circles. The agreement is very good for all the particles in the system. A similar comparison is shown in Fig. 12 for a 40-particle system. It is reassuring to observe that the results are equally good in this system where several dual cages are involved due to the higher particle density. Even more dual cages are necessary when the particle number is increased to 60, but the results are equally good (Fig. 13).

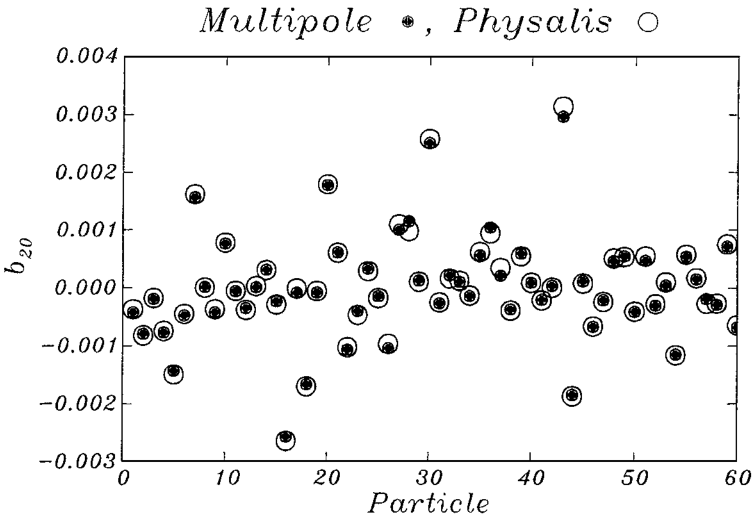

FIG. 13. Comparison of the coefficient b_{20} as computed by Physalis and the multipole method for a system of 60 particles.

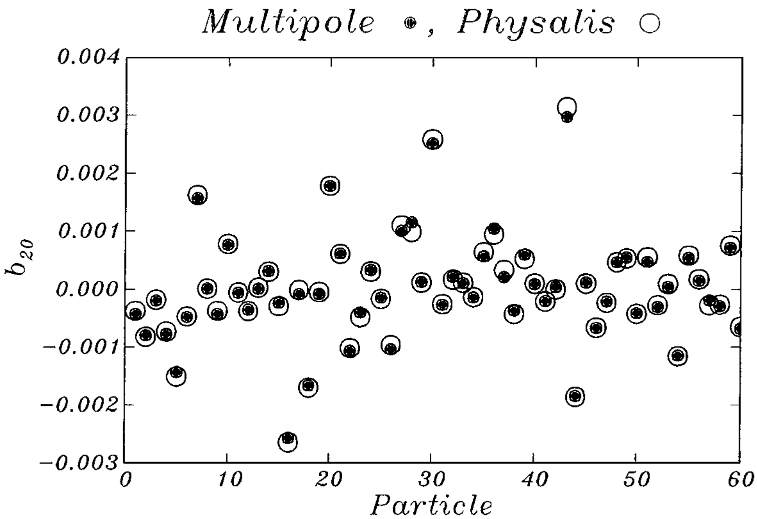

FIG. 14. Comparison of the coefficient b_{20} as computed by Physalis and the multipole method for a system of 60 particles; this figure differs from Fig. 13 in that here interpolation was used for the B-matrices.

The preceding results have been obtained by calculating the matrix B exactly for each particle. In Fig. 14 we show a similar comparison when the matrices are obtained by interpolation. The particle configuration and the discretization are identical to those of Fig. 13 and we have used 27 matrices (3 in each direction) in our matrix table. Once again, the comparison is excellent.

7. CONCLUSIONS AND OUTLOOK

We have presented in the simplest possible setting—potential flow—a new method for the direct numerical simulation of disperse flows of spheres; a similar approach can be followed for other particle shapes (e.g., cylinders), for which general exact local solutions can be written in analytic form. Further extensions to numerically generated local solutions may also be possible. The method can also be applied to other governing equations, such as the Helmholtz equation, the diffusion equation, and the equations of linear elasticity.

Preliminary results for the Navier–Stokes equations have been presented in Ref. [30] and further work will be described in future papers. The basic idea stems from the observation that, in the (noninertial) rest frame of each particle, the no-slip condition at the particle surface renders the Stokes equation (augmented by suitable apparent—or D'Alembert—forces) valid in the immediate neighborhood of the particle surface. To be sure, the larger the Reynolds number, the smaller the region where the Stokes equation gives an accurate description of the flow; nevertheless, for any finite Reynolds number, a finite such region can be identified. It is well known that the Stokes equation for a general flow near a sphere possesses an exact solution in terms of a spherical harmonic expansion [40, 41], and this local solution can be matched to the flow field at grid points surrounding the particle in the same way as was done here for potential flow. For this application the use of cubic cages is undesirable as it may extend the domain where the Stokes equation is assumed to hold to an excessive distance from the particle. In order to minimize this problem, we have used a cage made only of nodes about one mesh spacing away from the particle surface.

APPENDIX

The solution of the Laplace equation outside the spheres may be represented in terms of Green's function $|\mathbf{x}-\mathbf{x}'|^{-1}$ of the Laplace operator as

$$\phi'(\mathbf{x}) = \sum_{\alpha=1}^{N} \frac{1}{4\pi} \int_{S_\alpha} \left[\phi'(\mathbf{x}') \boldsymbol{\nabla}' \frac{1}{|\mathbf{x}-\mathbf{x}'|} - \frac{1}{|\mathbf{x}-\mathbf{x}'|} \boldsymbol{\nabla}' \phi' \right] \cdot \mathrm{n}' \, dS', \tag{A.1}$$

where $\phi' = \phi - \phi_\infty$, with ϕ_∞ a regular solution of the Laplace equation (representing, e.g., an imposed external flow) and the integrals are over each one of the surfaces S_α, $\alpha = 1, 2 \ldots, N$, of the N particles; the time dependence is inessential and has been suppressed. Upon setting $\mathbf{x}' = \mathbf{y}_\alpha + \mathbf{r}$, where $\mathbf{y}_\alpha$ is the position of the center of the αth particle, the Green's functions centered at each particle can be expanded in Taylor series of ascending powers of $a/|\mathbf{x}-\mathbf{y}_\alpha|$ to give rise to the well-known multipole expansion of the form

$$\phi(\mathbf{x}) = \phi_\infty(\mathbf{x}) + \sum_{\alpha=1}^{N} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{1}{|\mathbf{x}-\mathbf{y}_\alpha|^{n+1}} Y_n^m(\theta_\alpha, \varphi_\alpha) \, d_{nm}^\alpha. \tag{A.2}$$

Consider now the neighborhood of a generic particle that, for convenience, may be labeled as particle 1. In the region bounded by a sphere with radius equal to the distance between particle 1 and the particle closest to it, the potential may be represented in the form (24)

$$\phi = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} r_1^\ell Y_\ell^m(\theta_1, \varphi_1) c_{\ell m}^1 + \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \frac{1}{r_1^{\ell+1}} Y_\ell^m(\theta_1, \varphi_1) \, d_{\ell m}^1, \tag{A.3}$$

where $r_1 = |\mathbf{x}-\mathbf{y}_1|$, θ_1, φ_1 are spherical coordinates centered at the center of particle 1. The expansion coefficients $d_{\ell m}^1$ in this formula are the same as in (A.2), while the $c_{\ell m}^1$ are determined by writing.

$$\sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} |\mathbf{x}-\mathbf{y}_1|^\ell Y_\ell^m(\theta_1, \varphi_1) c_{\ell m}^1 = \phi_\infty + \sum_{\alpha=2}^{N} \sum_{\ell=2}^{\infty} \sum_{m=-\ell}^{\ell} \frac{1}{|\mathbf{x}-\mathbf{y}_\alpha|^{\ell+1}} Y_\ell^m(\theta_\alpha, \varphi_\alpha) \, d_{\ell m}^\alpha. \tag{A.4}$$

This equation can be solved for the $c_{\ell m}^1$ by taking a scalar product with the generic spherical harmonic Y_n^k to find, using orthogonality,

$$|\mathbf{x}-\mathbf{y}_1|^n \left(Y_n^k, Y_n^k\right) c_{nk}^1 = \left(Y_n^k, \phi_\infty\right) + \sum_{\alpha=2}^{N} \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \left(Y_n^k, \frac{1}{|\mathbf{x}-\mathbf{y}_\alpha|^{\ell+1}} Y_\ell^m \right) d_{\ell m}^\alpha. \tag{A.5}$$

The scalar products can be evaluated by using the results in Article 89 of Ref. [32]. The result is an infinite system of algebraic equations that can be suitably truncated and solved to express the $d_{\ell k}^\alpha$, $\alpha = 2, 3 \ldots,$ in terms of the $c_{\ell k}^1$. On the other hand, the $c_{\ell k}^1$'s can be expressed in terms of the $d_{\ell k}^1$ because of the boundary conditions at the surface of the spheres. Therefore one finds an algebraic system in which the unknowns are the $d_{\ell k}^\alpha$ and the forcing is provided by ϕ_∞ and, possibly, a term related to the boundary conditions on the particle surfaces, such as the particle velocities.

ACKNOWLEDGMENT

The support of this work by DOE under Grant DE-FG02–99ER14966 is gratefully acknowledged.

REFERENCES

1. J. F. Brady and G. Bossis, Stokesian dynamics. *Annu. Rev. Fluid Mech.* **20**, 111 (1988).
2. S. Weinbaum, P. Ganatos, and Z.-Y. Yan, Numerical multipole and boundary integral equation techniques in Stokes flow. *Annu. Rev. Fluid Mech.* **22**, 275 (1990).
3. B. Cichocki, B. U. Felderhof, K. Hinsen, E. Wajnryb, and J. Blawzdziewicz, Friction and mobility of many spheres in Stokes flow. *J. Chem. Phys.* **100**, 3780 (1994).
4. P. R. Nott and J. F. Brady, Pressure-driven flow of suspensions: Simulation and theory. *J. Fluid Mech.* **275**, 157 (1994).
5. C. C. Chang and R. L. Powell, The rheology of bimodal hard-sphere dispersions. *Phys. Fluids* **6**, 1628 (1994).
6. C. C. Chang and R. L. Powell, Self-diffusion of bimodal suspensions of hydrodynamically interacting spherical particles in shearing flow. *Phys. Fluids* **6**, 1628 (1994).
7. A. S. Sangani and G. B. Mo, An $O(N)$ for Stokes and Laplace interactions of spheres. *Phys. Fluids* **8**, 1990 (1996).
8. A. J. C. Ladd, Sedimentation of homogeneous suspensions of non-Brownian spheres. *Phys. Fluids* **9**, 491 (1997).
9. S. Elghobashi and G. C. Truesdell, On the two-way interaction between homogeneous turbulence and dispersed solid particles. I: Turbulence modification. *Phys. Fluids* **A5**, 1790 (1993).
10. D. E. Stock, Particle dispersion in flowing gases. *J. Fluids Eng.* **118**, 4 (1996).
11. Y. Pan and S. Banerjee, Numerical simulation of particle interactions with wall turbulence. *Phys. Fluids* **8**, 2733 (1996).
12. H. H. Hu, D. D. Joseph, and M. J. Crochet, Direct simulation of fluid-particle motions. *Theor. Comput. Fluid Dyn.* **3**, 285 (1992).
13. J. Feng, H. H. Hu, and D. D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid, Part 1. Sedimentation. *J. Fluid Mech.* **261**, 95 (1994).
14. J. Feng, H. H. Hu, and D. D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid, Part 2. Couette and Poiseuille flows. *J. Fluid Mech.* **277**, 271 (1994).
15. H. H. Hu, Motion of a circular cylinder in a viscous liquid between parallel plates. *Theor. Comput. Fluid Dyn.* **7**, 441 (1995).
16. H. H. Hu, Direct simulation of flows of solid–liquid mixtures. *Int. J. Multiphase Flow* **22**, 335 (1996).
17. A. A. Johnson and T. Tezduyar. Simulation of multiple spheres falling in a liquid-filled tube. *Comput. Meth. Appl. Mech. Eng.* **134**, 351 (1996).
18. A. A. Johnson and T. Tezduyar, 3-D simulation of fluid–particle interactions with the number of particles reaching 100. *Comput, Meth. Appl. Mech. Eng.* **145**, 301 (1997).
19. Y. Pan and S. Banerjee, Numerical investigation of the effect of large particles on wall turbulence. *Phys. Fluids* **9**, 3786 (1997).
20. A. L. Fogelson and C. S. Peskin, A fast numerical method for solving the three-dimensional Stokes equations in the presence of suspended particles. *J. Comput. Phys.* **79**, 50 (1988).
21. D. Sulsky and J. U. Brackbill, A numerical method for suspension flow. *J. Comput. Phys.* **96**, 339 (1991).
22. R. Glowinski, T. W. Pan, T. I. Hesla, and D. D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows. *Int. J. Multiphase Flow* **25**, 755 (1999).
23. N. A. Patankar, P. Singh, D. D. Joseph, R. Glowinski, and T.-W. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. *Int. J. Multiphase Flow* **26**, 1509 (2000).
24. S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.* **100**, 25 (1992).
25. S. O. Unverdi and G. Tryggvason, Computation of multi-fluid flows. *Physica D* **60**, 70 (1992).

26. A. Esmaeeli and G. Tryggvason, Direct numerical simulations of bubbly flows. Part 1. Low Reynolds numbers arrays. *J. Fluid Mech.* **314**, 315 (1998).

27. A. Esmaeeli and G. Tryggvason, Direct numerical simulations of bubbly flows. Part 2. Moderate Reynolds numbers arrays. *J. Fluid Mech.* **385**, 325 (1998).

28. H. Nirschl, H. A. Dwyer, and V. Denk, Three-dimensional calculations of the simple shear flow around a single particle between two moving walls. *J. Fluid Mech.* **283**, 273 (1995).

29. J. J. Chattot and Y. Wang, Improved treatment of intersecting bodies with the chimera method and validation with a simple and fast flow solver. *Comput. Fluids* **27**, 721 (1998).

30. E. Ory, H. N. Oğuz, and A. Prosperetti, PHYSALIS: A new numerical method for particle simulations, in *Proceedings of the A.S.M.E. FED Summer Meeting, New York, 2000*. A.S.M.E.

31. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations. *J. Comput. Phys.* **73**, 325 (1987).

32. E. W. Hobson, *The Theory of Spherical and Ellipsoidal Harmonics* (Cambridge Univ. Press, Cambridge, UK, 1931, reprinted by Chelsea, New York, 1965).

33. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. (Johns Hopkins Univ. Press, Baltimore, 1989).

34. W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery, *Numerical Recipes in FORTRAN*, 2nd ed. (Cambridge Univ. Press, Cambridge, UK, 1992).

35. M. Watanabe, *Topics in Bubbly Liquid Flows and Cavitation*, Ph.D. thesis (Johns Hopkins University, Baltimore, 1995).

36. A. S. Sangani and C. Yao, Bulk conductivity of composites with spherical inclusions. *J. Appl. Phys.* **63**, 1334 (1998).

37. 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).

38. A. S. Sangani and A. Acrivos, Slow flow through a periodic array of spheres. *Int. J. Multiphase Flow* **8**, 343 (1982).

39. A. S. Sangani, D. Z. Zhang, and A. Prosperetti, The added mass, Basset, and viscous drag coefficients in nondilute bubbly liquids undergoing small-amplitude oscillatory motion. *Phys. Fluids* **A3**, 2955 (1991).

40. H. Lamb, *Hydrodynamics*, 6th ed. (Cambridge Univ. Press, Cambridge, UK, 1932).

41. S. Kim and S. J. Karrila, *Microhydrodynamics* (Butterworth-Heinemann, Boston, 1991).