



A second-order method for three-dimensional particle simulation

Z. Zhang ^{a,*}, A. Prosperetti ^{a,b,1}

^a *Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218, USA*

^b *Faculty of Applied Science and IMPACT, University of Twente, AE 7500 Enschede, The Netherlands, and Burgerscentrum, The Netherlands*

Received 29 October 2004; received in revised form 10 April 2005; accepted 18 April 2005

Available online 24 June 2005

Abstract

This paper describes a numerical method for the direct numerical simulation of Navier–Stokes flows with one or more solid spheres. The particles may be fixed or mobile, and they may have different radii. The basic idea of the method stems from the observation that, due to the no-slip condition, in the reference frame of each particle, the velocity near the particle boundary is very small so that the Stokes equations constitute an excellent approximation to the full Navier–Stokes problem. The general analytic solution of the Stokes equations can then be used to “transfer” the no-slip condition from the particle surface to the adjacent grid nodes. 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 method is validated by a detailed comparison with spectral solutions for the flow past a sphere at Reynolds numbers of 50 and 100. The existence in these situations of a Stokes region near the particle is explicitly demonstrated. Other numerical experiments to show the performance of the code are also described. To illustrate the power and efficiency of the method, a simulation of decaying homogeneous turbulence in a cell containing 100 moveable spheres is described. As implemented here, the method can only be applied to simple body shapes such as spheres and cylinders. Extensions to more general situations are mentioned in the last section.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Flow around spheres; Particle flow; Disperse multiphase flow

* Corresponding author.

E-mail address: prosper@jhu.edu (A. Prosperetti).

¹ Tel.: +1 410 516 8534; fax: +1 410 516 7254.

1. Introduction

This paper describes the extension of our numerical method `PHYSALIS` for the simulation of particles suspended in a viscous fluid described in [1,2] to three spatial dimensions and second order time- and space-accuracy.

One of the main motivations for the development of this method is the desire to carry out simulations of turbulent disperse two-phase flows. This is an active research area in which much of the work to date has been based on point-particle models (see e.g. [3–9]). The limitations of these models are well known. Even leaving aside the difficulty of accurately parameterizing the fluid-particle forces, this approximation may only be justified for dilute systems and very small particles, such as are encountered in dusty gases, electrostatic separators, and others. In more concentrated systems, such as fluidized beds, solids transport, and others, however, the fact that the particles have a finite size plays a determinant role in the physics of the problem and cannot be neglected.

In the past decade several methods have been proposed to simulate in an efficient manner the flow around suspended particles. Examples are the early calculations of Joseph and collaborators (see e.g. [10,11]), the more recent work of this group (see e.g. [12–14]) and others (e.g. [15]) with a fixed finite-element mesh, the finite-element calculations of [16,17], and the `CHIMERA` method (see, e.g., [18,19]). These applications are a special case of problems with complex boundaries, which are currently a subject of active research in the computational fluid dynamics community. In particular, the immersed boundary method, originally developed by Peskin [20] (see also [21]), and later used e.g. in [22,23], has attracted considerable interest in the last few years giving rise to so-called Cartesian grid methods (see e.g. [24,25]) and other immersed boundary methods (see e.g. [26–32,71]). Methods based on lattice-Boltzmann formulations have also shown considerable promise and are being pursued by a number of investigators [33–36]. An interesting combination of the lattice-Boltzmann and immersed boundary approaches is presented in [37,38].

The method described in the present paper differs from those mentioned above. It is based on the observation that, because of the no-slip boundary condition on its surface, the flow in the immediate neighborhood of a particle differs little from a rigid-body motion and can be linearized about this rigid-body motion with a negligible error. It is shown that this specific structure of the flow manifests itself in certain non-local relations among the flow fields (velocity, pressure, vorticity) which can be captured by an analytic solution. Thus, rather than solving the problem with the particle in place, one can impose this relationship directly on the nodes of a fixed regular grid and effectively remove the particle. In this way, the actual boundary of the particle, with its usually complex relation to the underlying regular grid, can be replaced by a simpler boundary consisting of grid nodes. The non-local constraint on the flow fields is imposed by matching the local analytic solution, valid in the neighborhood of each particle, to a global solution valid elsewhere in the domain. The reconciliation of the two solutions is carried out iteratively.²

The global solution is calculated by a second-order finite-difference projection method, while the local solution is expressed as a spectrally convergent series. In this respect, our method is similar to that of [40].³ While in [40] use was made of an expansion in Chebyshev polynomials, a transformation of the dependent variables enables us to use Stokes flow expansions which are better suited for the simulation of spherical particles in which we are interested. It should be explicitly noted that our use of Stokes flow theory in no way restricts our method to low Reynolds number. Indeed, as shown in Section 4.2, our method gives very accurate results for steady flow past a sphere at Reynolds numbers of 50 and 100.

Unlike some other methods, our procedure does not require approximating the particle shape in a “staircase” way (see e.g. [35,41]) nor diffusing its boundary over one or more cells (see e.g. [38,42]). The no-slip

² The convergence of this procedure was studied in [39] for some model problems.

³ We are grateful to one of the referees for pointing out this interesting work to us.

condition at the particle surface is enforced exactly and, as the number of degrees of freedom per particle is increased, the error decreases spectrally. On the other hand, as presented here, the method is only applicable to spherical particles. As discussed in the last section, extensions are possible although spectral convergence may have to be abandoned.

Although very different in detail, PHYSALIS is in some sense related to the method of artificial boundary conditions of Givoli, Keller, and collaborators [43–46] and in particular to the Dirichlet-to-Neumann map. The reconciliation of the local and global solutions is also reminiscent of the situation encountered in the domain decomposition method (see e.g. [47–49]) and, in some implementations, reminiscent of the capacitance matrix method (see e.g. [50–52]). In many respects, these are more analogies than similarities, as the differences with all of these methods are quite substantial.

2. Reduction to the rest frame

The computational domain contains a viscous Newtonian fluid in which several (equal or unequal) spherical particles are suspended. In order to show how a local solution valid near each particle can be derived, we consider the flow in the neighborhood of a generic particle, having instantaneous translational and angular velocities $\mathbf{w}(t)$ and $\mathbf{\Omega}(t)$, respectively, and adopt a non-inertial reference frame in which the particle is at rest. If \mathbf{u} and \mathbf{U} denote the flow velocities in the particle rest frame and the original inertial frame, respectively, we then have

$$\mathbf{U} = \mathbf{u} + \mathbf{w} + \mathbf{\Omega} \times \mathbf{r}, \quad (1)$$

in which \mathbf{r} is the position relative to the particle center. The momentum equation in the rest frame takes the form

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + 2\mathbf{\Omega} \times \mathbf{u} \right] = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} - \rho [\dot{\mathbf{w}} + \dot{\mathbf{\Omega}} \times \mathbf{r} + \mathbf{\Omega} \times (\mathbf{\Omega} \times \mathbf{r})], \quad (2)$$

to be solved subject to the incompressibility constraint and to the boundary condition $\mathbf{u} = 0$ on the particle surface. In (2) ρ and μ are the fluid density and viscosity, p is the pressure, and \mathbf{g} the body force; dots denote Lagrangian time derivatives following the particle.

The change of variables

$$\mathbf{u} = \tilde{\mathbf{u}} + \frac{r^5 - a^5}{10\nu r^3} \dot{\mathbf{\Omega}} \times \mathbf{r}, \quad p = \tilde{p} + \frac{1}{2} \rho (\mathbf{\Omega} \times \mathbf{r})^2 + \rho (\mathbf{g} - \dot{\mathbf{w}}) \cdot \mathbf{r}, \quad (3)$$

in which $r = |\mathbf{r}|$ and ν is the kinematic viscosity, brings (2) into the form

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + 2\mathbf{\Omega} \times \mathbf{u} \right] = -\nabla \tilde{p} + \mu \nabla^2 \tilde{\mathbf{u}}, \quad (4)$$

with $\tilde{\mathbf{u}} = 0$ on the particle surface.

It will be noted that the left-hand side of (4) contains the original velocity in the particle rest-frame, \mathbf{u} , which equals zero on the particle surface. Therefore, by continuity, this quantity will be small near the particle and, therefore, there is a region adjacent to the particle where the left-hand side of (4) is small. Thus, locally, $(\tilde{\mathbf{u}}, \tilde{p})$ approximately satisfy

$$-\nabla \tilde{p} + \mu \nabla^2 \tilde{\mathbf{u}} = 0, \quad \nabla \cdot \tilde{\mathbf{u}} = 0, \quad (5)$$

i.e., the Stokes equations. Naturally, the extent of the spatial region where (5) are a good approximation to (4) becomes smaller and smaller as the Reynolds number increases but, for any finite Reynolds number, there is a non-vanishing region where (5) are applicable with but a small error.

The general solution of the Stokes equations (5) in the presence of a spherical boundary was given by Lamb ([53, p. 594]; see also [54, Chapter 4]) and may be written in the form

$$\begin{aligned} \tilde{\mathbf{u}} = & \frac{v}{a^2} \sum_{n=1}^{\infty} \frac{1}{(n+1)(2n+3)} \left[\frac{1}{2} (n+3) r^2 \nabla p_n - n \mathbf{r} p_n \right] + \frac{v}{a} \sum_{n=1}^{\infty} [a \nabla \phi_n + \nabla \times (\mathbf{r} \chi_n)] \\ & + \frac{v}{a^2} \sum_{n=1}^{\infty} \frac{1}{n(2n-1)} \left[-\frac{1}{2} (n-2) r^2 \nabla p_{-n-1} + (n+1) \mathbf{r} p_{-n-1} \right] + \frac{v}{a} \sum_{n=1}^{\infty} [a \nabla \phi_{-n-1} + \nabla \times (\mathbf{r} \chi_{-n-1})], \end{aligned} \quad (6)$$

$$\tilde{p} = \frac{\mu v}{a^2} \left[p_0 + \sum_{n=1}^{\infty} (p_n + p_{-n-1}) \right]. \quad (7)$$

The p_n, ϕ_n, χ_n are regular solid spherical harmonics of order n , while those carrying a negative index are singular harmonics of the same order. Thus, e.g.

$$p_n = \left(\frac{r}{a}\right)^n \sum_{m=0}^n [P_{nm} \cos m\varphi + \tilde{P}_{nm} \sin m\varphi] P_n^m(\cos \theta), \quad (8)$$

where P_{nm} and \tilde{P}_{nm} are dimensionless coefficients, P_n^m is an associated Legendre function, and r, θ , and φ are spherical coordinates centered at the particle center. The other harmonics ϕ_n, χ_n are written in a similar way with dimensionless coefficients $\Phi_{nm}, \tilde{\Phi}_{nm}, X_{nm}$, and \tilde{X}_{nm} , respectively. In (6) and (7) the regular harmonics represent the incident flow and the singular ones the disturbance induced by the particle.

It is readily shown that the form of the singular harmonics dictated by the condition of vanishing velocity on the particle surface is

$$p_{-n-1} = -\frac{1}{2} \frac{2n-1}{n+1} n [p_n + 2(2n+1)\phi_n] \left(\frac{a}{r}\right)^{2n+1}, \quad (9)$$

$$\phi_{-n-1} = -\frac{a^2}{4} \frac{n}{n+1} \left[\frac{2n+1}{2n+3} p_n + 2(2n-1)\phi_n \right] \left(\frac{a}{r}\right)^{2n+1}, \quad (10)$$

$$\chi_{-n-1} = -\left(\frac{a}{r}\right)^{2n+1} \chi_n. \quad (11)$$

Upon substitution into (6) and (7), these relations express the local fields near the particle in terms of the incident flow.

The vorticity field corresponding to (6) is

$$\tilde{\omega} = \frac{v}{a^2} \sum_{n=1}^{\infty} \left[\frac{1}{n+1} (\mathbf{r} \times \nabla p_n) - \frac{1}{n} (\mathbf{r} \times \nabla p_{-n-1}) + a \nabla \times \nabla [\mathbf{r} \times (\chi_n + \chi_{-n-1})] \right]. \quad (12)$$

Detailed formulae for the components of \mathbf{u} and ω are given in Appendix A.

The particle position \mathbf{y} and velocity \mathbf{w} are updated using

$$\frac{d\mathbf{y}}{dt} = \mathbf{w}, \quad m \frac{d\mathbf{w}}{dt} = \mathbf{F}_{hd} + \mathbf{F}_e, \quad (13)$$

where m is the particle mass, \mathbf{F}_{hd} the hydrodynamic force, and \mathbf{F}_e the external force. The particle angular velocity Ω is updated from

$$I \frac{d\Omega}{dt} = \mathbf{L}_{hd} + \mathbf{L}_e, \quad (14)$$

where I is the moment of inertia, \mathbf{L}_{hd} the hydrodynamic couple, and \mathbf{L}_e the external couple. From Ω one could readily calculate the change in the particle orientation. However, for spherical homogeneous

particles, this degree of freedom is uncoupled from the dynamics. In the original inertial frame, the hydrodynamic force and couple acting on the particle are given by

$$\mathbf{F}_{\text{hd}} = \rho v(\dot{\mathbf{w}} - \mathbf{g}) + \pi\mu v(P_{11} + 6\Phi_{11}, \tilde{P}_{11} + 6\tilde{\Phi}_{11}, P_{10} + 6\Phi_{10}), \quad (15)$$

$$\mathbf{L}_{\text{hd}} = \rho v a^2 \dot{\boldsymbol{\Omega}} + 8\pi\mu v a(X_{11}, \tilde{X}_{11}, X_{10}). \quad (16)$$

It should be stressed that (6) and (7) represent the *most general* Stokes flow compatible with the boundary conditions at the particle surface. Clearly, no assumptions or restrictions about this flow (in particular, about its behavior far from the particle) have been introduced. The plan of the calculation consists in determining the coefficients of the local solution in the neighborhood of each particle by matching the fields given by (1) and (3) with (6) and (7), with those numerically computed.

3. Implementation

In order to implement the matching between the local and global fields mentioned in the previous section, we cover the entire domain by a regular finite-difference grid without regard for the presence of the particles. Each particle is surrounded by a *cage* of cells straddling the body surface. An example is shown in Fig. 1 and a two-dimensional cut is shown for greater clarity in Fig. 2; the algorithm used to construct the cages is described later. We use a standard staggered grid arrangement, with pressure at cell centers (crosses in Fig. 2) and velocities at the midpoints of cell sides (arrows). The z -vorticity component at position $(i + 1/2, j + 1/2, k)$ is calculated from

$$(\omega_z)_{i+1/2, j+1/2, k} \simeq \frac{(u_y)_{i+1, j+1/2, k} - (u_y)_{i, j+1/2, k}}{\Delta x} - \frac{(u_x)_{i+1/2, j+1/2, k} - (u_x)_{i+1/2, j, k}}{\Delta y}, \quad (17)$$

where Δx and Δy are the mesh spacings, and, therefore, it resides at the midpoint of cell edges (\bullet , in Fig. 2).

The computational procedure can be summarized as follows. Suppose that a provisional estimate of velocity and pressure fields is available; this could be, for example, the velocity field at the previous time step or the previous iteration. From this estimate, provisional auxiliary fields \tilde{p} and $\tilde{\omega}$, defined in (3) can be calculated and the vorticity found by using (17). Then, after truncating the summations in Eqs. (A.2)–(A.8) for pressure, velocity, and vorticity to a finite number of terms N_c :

- (1) For each particle, let $1, 2, \dots, N_p$ be the pressure nodes and $1, 2, \dots, N_\omega$ the vorticity-component nodes of the cage. Match the provisional \tilde{p} and $\tilde{\omega}$ to the analytic expressions (7) and (12) at these nodes to generate a linear system for the coefficients P_{nm} , Φ_{nm} , X_{nm} , and their tilde counterparts. In principle, the maximum number of coefficients that can be determined in this way equals the number of cage nodes although, as explained below, it is necessary to use fewer coefficients. The linear system has therefore a rectangular matrix and is solved in a least square sense (see below).
- (2) By using the values of the coefficients determined at the previous step, compute from the analytic formulae (1), (3), and (6) the velocity at the velocity points $1, 2, \dots, N_v$ of the cage surrounding the particle.
- (3) For each particle, recalculate new position, velocity, angular velocity, and cage.
- (4) Solve the full Navier–Stokes equations on the finite-difference grid imposing the new velocity field as boundary condition on the velocity nodes of the cage of each particle.
- (5) Calculate the corresponding pressure and vorticity, return to step 1, and repeat until convergence.

As will be explained in the next section, in executing step 4, it is efficient to solve for the flow field over the entire grid, disregarding the presence of the particles. The velocity field outside the cages is the one that

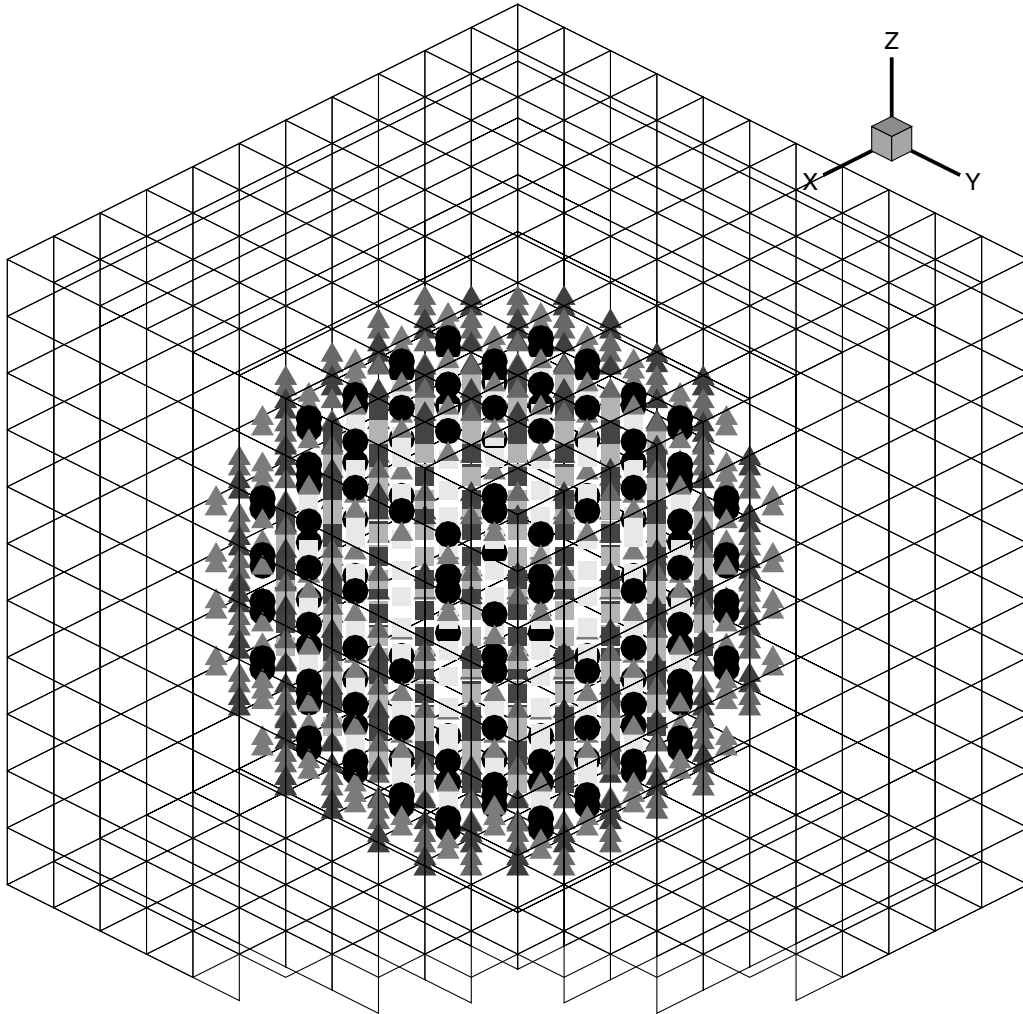


Fig. 1. Perspective view of a particle and associated cage. Black disks: pressure points; triangles: vorticity points; squares: velocity points.

is desired. The field inside the cages is the correct solution of another flow problem, in which the flow is driven by the imposed velocity on the cage nodes: this solution is not unphysical – it is simply irrelevant for the purposes of the calculation and can be disregarded. It should be stressed that the solution procedure is devised in such a way that, other than for satisfying a common velocity boundary conditions at the cage nodes, the solutions inside and outside the cages are completely uncoupled, so that any ‘contamination’ of the latter by the former is avoided. In particular there is no continuity or other relation satisfied by the stresses across the cage surface. The final flow field is given by the finite-difference solution outside the cages, and by the analytical representation (6) and (7) in the thin region between each particle and the surrounding cage.

We now turn to some details of the implementation; additional information can be found in [1].

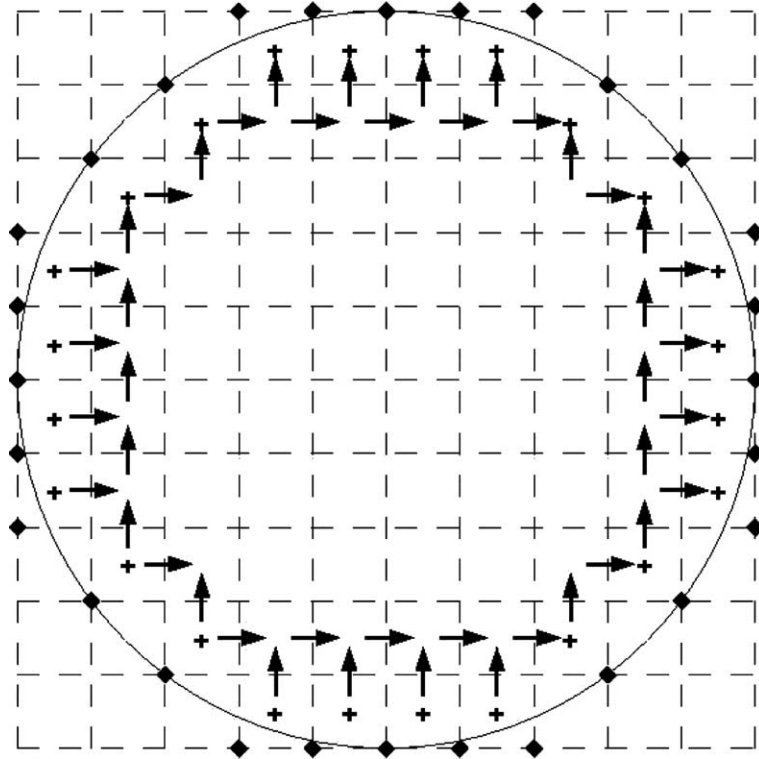


Fig. 2. A two-dimensional cut across the particle of the previous figure showing some nodes of a typical cage: +, pressure points; •, vorticity points; →, horizontal velocity points; ↑, vertical velocity points.

3.1. Flow solver

As explained before, the calculation progresses from time level t^n to $t^{n+1} = t^n + \Delta t$ by reconciling in an iterative fashion the local analytic solution for each particle with the finite-difference one. Let κ be the counter for these iterations.

For each iteration, the first step is executed by the second-order-accurate projection method described in [55] suitably modified for the present purposes as we now explain. A provisional estimate \mathbf{u}^* of the new velocity $\mathbf{u}^{\kappa+1}$ is calculated from

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{1}{\rho} \nabla_h p^{\kappa-1/2} = -(\mathbf{u} \cdot \nabla_h \mathbf{u})^{\kappa+1/2} + \frac{1}{2} \nu \nabla_h^2 (\mathbf{u}^n + \mathbf{u}^*), \tag{18}$$

where the convective term $(\mathbf{u} \cdot \nabla_h \mathbf{u})^{\kappa+1/2}$ is calculated by the second-order Adams–Bashforth method and the diffusive term $\nabla_h^2 (\mathbf{u}^n + \mathbf{u}^*)$ by the Crank–Nicholson method; ∇_h denotes the spatial discretization on the finite-difference grid with spacing h in all directions. In the examples described below, at the outer boundary of the computational domain we have Dirichlet, periodicity, or outflow boundary conditions. In solving (18), these conditions are imposed directly on \mathbf{u}^* . At the cage nodes, \mathbf{u}^* is set equal to the velocity obtained from the analytic representation of the solution calculated using the most recently updated values of the coefficients.

The pressure is given by

$$\nabla_h p^{\kappa+1/2} = \nabla_h p^{\kappa-1/2} + \nabla_h \phi^{\kappa+1} - \frac{1}{2} \nu \Delta t \nabla_h \nabla_h^2 \phi^{\kappa+1}, \tag{19}$$

where

$$\nabla_h^2 \phi^{\kappa+1} = \rho \frac{\nabla_h \cdot \mathbf{u}^*}{\Delta t} - \frac{1}{h} P_h \mathbf{n}_c \cdot \nabla_h \phi^{\kappa+1}. \quad (20)$$

Here, P_h is a projector which singles out the cage nodes [56,1,2], and \mathbf{n}_c is the unit normal directed outward from the cage. As an example (see [2]), some of the relevant terms of (20) would be

$$\frac{1}{h} \left(\frac{\phi_{i+1,j,k}^{\kappa+1} - \phi_{i,j,k}^{\kappa+1}}{h} - \frac{\phi_{i,j,k}^{\kappa+1} - \phi_{i-1,j,k}^{\kappa+1}}{h} \right) + \dots = \rho \frac{\nabla_h \cdot \mathbf{u}^*}{\Delta t} - \frac{\phi_{i,j,k}^{\kappa+1} - \phi_{i-1,j,k}^{\kappa+1}}{h^2} \quad (21)$$

in which the dots in the left-hand side stand for the remaining terms of the discretized Laplacian. At convergence, this deferred-correction approach effectively enforces a zero-normal-gradient condition on the cage nodes without the necessity of recognizing them as internal boundaries. This feature enables us to use fast Poisson solvers for (20). It may also be noted that, at convergence, $\phi_{i-1}^{\kappa+1}$ cancels on the two sides of (21) so that there is no coupling between the pressure fields inside and outside the particle. In view of this feature, how the values of $\phi^{\kappa+1}$ at the nodes internal to the particles are initialized is immaterial. Eq. (20) is implicit in $\phi^{\kappa+1}$ and is solved by iteration. Since these iterations are embedded in the outer iteration procedure which updates the coefficients of the analytic solution, it is not necessary to adopt a very strict criterion for their termination. At the external boundary of the computational domain we impose

$$\mathbf{n} \cdot \nabla_h \phi^{\kappa+1} = 0, \quad (22)$$

where \mathbf{n} is the outward unit normal. The final velocity at the end of the iteration is given by

$$\mathbf{u}^{\kappa+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla_h \phi^{\kappa+1}. \quad (23)$$

At the end of each iteration, the position \mathbf{y} , velocity \mathbf{w} , and angular velocity $\boldsymbol{\Omega}$ of each particle are updated by a second-order discretization of (13) and (14):

$$\mathbf{w}^{\kappa+1} = \mathbf{w}^n + \frac{\Delta t}{2m} (\mathbf{F}_{\text{hd}}^{\kappa+1} + \mathbf{F}_{\text{hd}}^n + \mathbf{F}_e^{n+1} + \mathbf{F}_e^n), \quad \mathbf{y}^{\kappa+1} = \mathbf{y}^n + \frac{\Delta t}{2} (\mathbf{w}^{\kappa+1} + \mathbf{w}^n), \quad (24)$$

$$\boldsymbol{\Omega}^{\kappa+1} = \boldsymbol{\Omega}^n + \frac{\Delta t}{2I} (\mathbf{L}_{\text{hd}}^{\kappa+1} + \mathbf{L}_{\text{hd}}^n + \mathbf{L}_e^{\kappa+1} + \mathbf{L}_e^n). \quad (25)$$

As described, the procedure contains two iterations, an inner one, Eq. (20), to solve the equation for ϕ , and an outer one to synchronize the expansion coefficients with the flow field. The first iteration can be eliminated by avoiding the use of a fast solver for the Poisson equation. The boundary condition $\mathbf{n} \cdot \nabla \phi$ would be imposed directly rather than iteratively as in (20), and the equation would then be solved only outside the particle cages rather than everywhere. The second iteration is required by the second-order time-accuracy of the scheme, like other second-order methods (see e.g. [32]). If one were to use a very small time step, one could advance the flow field, calculate the coefficients from p and ω as before, use these coefficients to generate new boundary conditions for \mathbf{u} at the cage nodes, advance the flow field by another step, and so forth. In this case, the time-accuracy would evidently decrease to first order, like other methods that do not use iterative procedures (see e.g. [26–29,31]).

3.2. Truncation and matching

Truncation of the expansions for the flow fields at order $n = N_c$ retains $(N_c + 1)^2$ coefficients (P_{nm}, \tilde{P}_{nm}), $N_c(N_c + 2)$ coefficients ($\Phi_{nm}, \tilde{\Phi}_{nm}$), and (X_{nm}, \tilde{X}_{nm}) for a total of $3N_c(N_c + 2) + 1$ coefficients.

As noted before, at each iteration, the coefficients $(P^{\kappa+1}, \Phi^{\kappa+1}, X^{\kappa+1})$ and their tilde counterparts are updated by matching the computed values of $p^{\kappa+1}$ (obtained from $p^{\kappa+1/2}$ and $p^{n-1/2}$ by extrapolation) and $\omega^{\kappa+1}$ to the Stokes analytic expressions (7) and (12) on the cage nodes. If h is the grid spacing and a the particle radius, there are about $4\pi a^2/h^2$ cage nodes for each scalar field (i.e. p and the three components of ω). Therefore, the matching operation gives rise to a linear system of approximately $4 \times (4\pi a^2/h^2)$ equations.

As already mentioned, in general it is not possible to retain these many coefficients. A first consideration is aliasing. One cannot include modes of such a high order that they cannot be represented on the finite-difference grid. A very crude estimate of this upper limit can be obtained by noting that the shortest wave that can be represented on a grid with a spacing h is of the order of $2h$. Over a length of the order of $2a$, one can fit a/h waves with this wavelength, which therefore puts an upper limit of this order on the maximum value of N_c . Our numerical experience, however, indicates that N_c should be taken smaller than this value. We are at present unable to give a precise guideline concerning this aspect of our method, for which we have to rely on the numerical experience described later.

In [57] it is suggested that a consistent truncation would retain N_c coefficients (P_{nm}, \tilde{P}_{nm}) , $(N_c - 2)$ $(\Phi_{nm}, \tilde{\Phi}_{nm})$, and $(N_c - 1)$ (X_{nm}, \tilde{X}_{nm}) . We tried this truncation without any appreciable change in the results.

The linear system for the coefficients has a rectangular matrix with many more rows than columns and is solved by the singular value decomposition algorithm, which is equivalent to a least-squares procedure when all singular values are retained as here (see e.g. [58,59]). Use of this algorithm also enables us to consider all three components of the vorticity, which would give rise to a nearly singular matrix if the number of unknowns were equal to the number of equations.

3.3. Cage

We now turn to an explanation of the procedure followed to generate the cages used in the matching of the coefficients with the finite-difference solution. As will be explained later, in some cases it may be convenient to center the cage at a point other than the center of the actual particle. We refer to the spherical volume centered at the cage center and equal to the particle volume as the “virtual” particle.

- (1) The grid node closest to the cage center is located and, around that node, a cubic box is built with sides equal to the number of grid nodes per diameter plus 4, so as to make sure that the particle plus a layer of cells sufficient to locate all the cage nodes is entirely contained in it.
- (2) All the cells contained in the cage box are examined in turn to test whether their center is inside the virtual particle. The cage pressure points are identified as the cell centers inside the virtual particle, which have at least one adjacent cell outside the virtual particle. Examples are the points marked by black disks in Fig. 3 such as A , B , C , and D .
- (3) At the same time as the pressure points are identified, we construct the *cage surface* (solid line in Fig. 3) as the union of all the faces of the pressure cells which are common to other cells entirely contained inside the virtual particle which are not pressure cells. This condition prevents a cell surface such as that separating points C and D in Fig. 3 from being counted as part of the cage interface.
- (4) In a staggered grid arrangement, the velocity components are defined at the cell face centers. The cage velocity components are those velocity nodes which are closest to the pressure nodes, and also closer to the cage center than the pressure nodes.
- (5) In the staggered grid arrangement, vorticity nodes are at the midpoint of the cell edges. For each pressure cell and each vorticity component, therefore, there are 4 possible node positions (Fig. 4(a)). The one is chosen which is farthest from the center of the virtual particle such as node A in Fig. 4(a). It is possible that, with this procedure, gaps remain in the vorticity-node cage. An example is shown in Fig. 4(b). In this case nodes such as the one marked by an asterisk in Fig. 4(b) are added.

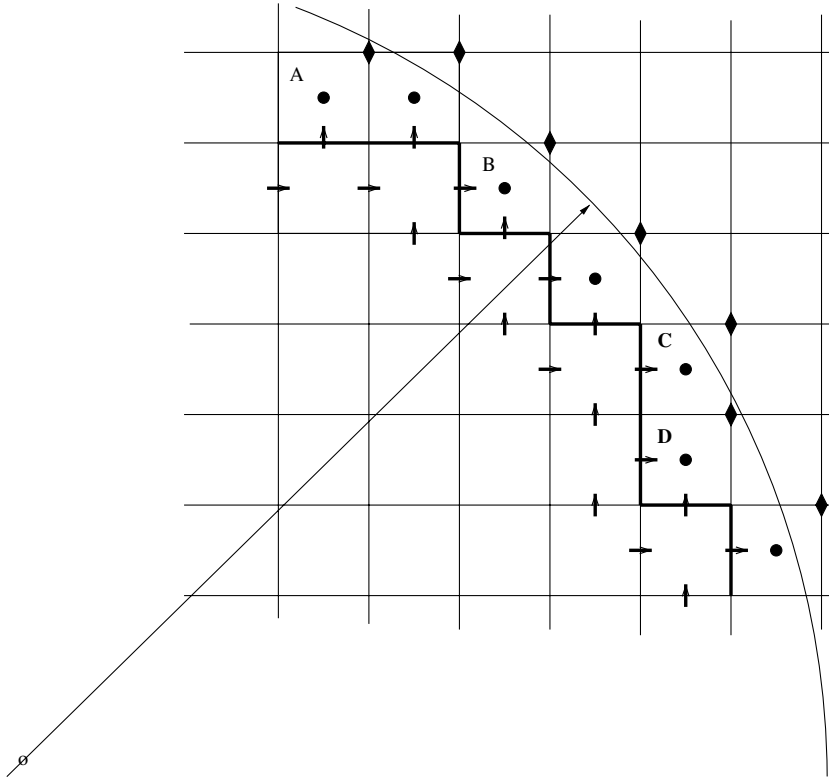


Fig. 3. Explanation of the cage construction procedure; see Section 3.3.

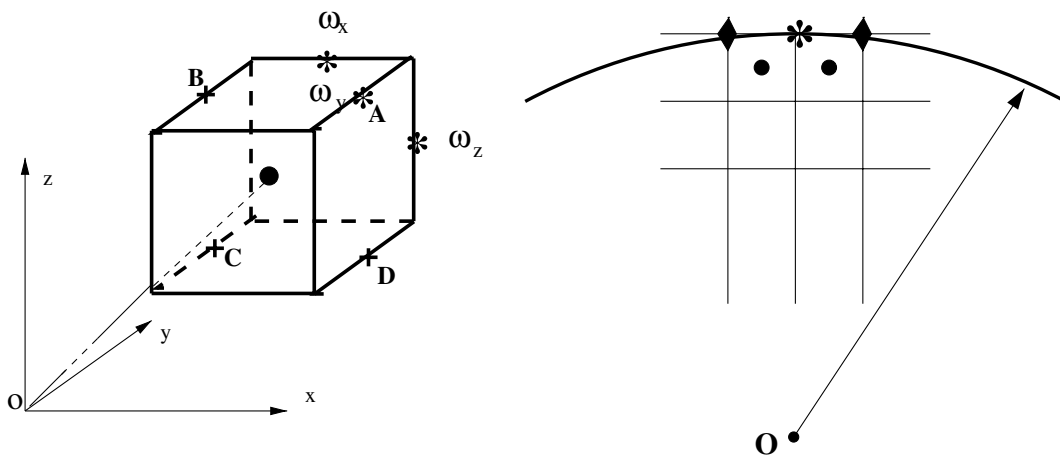


Fig. 4. (a) Left: A, B, C, D denote the four possible node positions for the y -component of vorticity. The one actually chosen is A , the node farthest from the center O of the virtual particle. (b) Right: Example of a gap (asterisk) that may remain in the vorticity cage constructed according to the procedure of Section 3.3. The node marked by the asterisk is added to the cage.

Many of the points used in the algorithm thus fall inside the particle, but this does not create any difficulty – practical or conceptual – as (6), (7), and (12) provide an analytic continuation of the flow fields inside the particle provided, of course, that $r > 0$. As a matter of fact, the possibility of placing the cage mostly inside the particle permits two particles to come in very near contact, or even overlap slightly.

If the particle moves, a new cage can be centered at each successive position of the particle center, or the same cage used as long as the particle remains in some neighborhood of the cage center, after which another cage is used, and so on. In this approach, one would only need to build a set of “standard cages” once at the beginning of the calculation. Evidently the second method is computationally more efficient, although it introduces a somewhat greater error as described later.

In this second approach, the lowest-order accuracy procedure is to use the same cage centered at the grid point (x_i, y_j, z_k) closest to the particle center as long as the latter is within a volume $x_i - \frac{1}{2}h \leq x \leq x_i + \frac{1}{2}h$, etc. The cage center will move to a different grid point only when the particle center moves out of this volume. These abrupt shifts in the cage position influence the smoothness of the time dependence of the hydrodynamic force and couple. This undesirable feature can be mitigated by using cages centered at several positions surrounding the grid point (x_i, y_j, z_k) . A natural choice is the nine points (x_i, y_j, z_k) and $(x_i \pm \frac{1}{2}h, y_j \pm \frac{1}{2}h, z_k \pm \frac{1}{2}h)$, for each one of which the previous procedure generates a different cage. The procedure can be further refined in an obvious way by including additional cages centered at intermediate points.

3.4. Spatial resolution

A source of error for the method is its reliance on an approximate solution in the fluid regions between the particle and the surrounding cage. This error can be reduced by refining the grid, which has the effect of putting the cage nodes closer to the particle surface. Another possibility (which we have not yet explored) would be to improve on the Stokes flow solution by approximating the solution of the full nonlinear equation (4) by a regular perturbation expansion. It may be noted, however, that in practice some control of the error is built into the procedure as convergence requires that all the flow fields be described by the *same* set of coefficients (P, Φ, X) . In the presence of strong nonlinear effects, which are not accounted for in the analytic solution, one would expect that this condition could not be satisfied and the iterative procedure would fail to converge. This expectation is borne out by the numerical evidence reported in Section 4.3 and also by exploratory calculations where we purposely put the cage nodes too far from the particle surface, and observed either an exceedingly slow convergence rate or convergence failure.

A rough idea of the grid size h necessary for a good numerical accuracy may be found by noting that the grid points should be well inside the boundary layer for the Stokes approximation to be valid. If the boundary layer thickness is estimated as a/\sqrt{Re} , where Re is the Reynolds number expressed in terms of a characteristic velocity and the sphere diameter, we thus have that ah should be sufficiently larger than \sqrt{Re} . This limit is not different from that applicable to a standard finite-difference calculation.

4. Validation

The method described before and the code that implements it have been subjected to a thorough series of validation tests.

4.1. Stokes flow

The most basic test of the method is whether a given exact Stokes solution can be reconstructed by the algorithm. To carry out this test we assign values to some of the coefficients appearing in the exact analytic

solution of the Stokes equations and use this solution to prescribe the velocity field on the boundary of the computational domain. We then study to what extent the prescribed coefficients can be recovered by the algorithm. In carrying out this test it is unnecessary to use a large domain and we choose a cube with a side of 4 times the sphere radius with the sphere centered at the center of the cube. Furthermore, if high-order coefficients are assigned as numbers of order 1, the velocity rapidly develops very large gradients as one moves away from the sphere, which would require a very fine discretization for an accurate finite-difference calculation. The use of a relatively small domain is suggested also by this consideration. To limit further the growth of the high-order modes, we also prescribe high-order coefficients as relatively small numbers.

Table 1 shows the results obtained with two discretizations, $alh = 4$ and 8 for A_{10} , C_{10} , and E_{10} assigned, in turn, equal to 1. Here and below, the coefficients not shown in the table and the entries left blank indicate numbers smaller than 0.001. For $alh = 4$ we can include in the reconstruction coefficients up to $N_c = 3$ (a total of 46). If we include also the $N_c = 4$ coefficients in the attempted reconstruction, the algorithm diverges. It appears that the finite-difference error contaminates the higher-order coefficients which in turn feed the error back to the boundary conditions at the cage, ultimately causing failure. This interpretation is supported by the fact that, if we go to higher resolution, $alh = 8$, the algorithm converges all the way up

Table 1
Recovery of assigned coefficients (column ‘Exact’) by the numerical method for different discretizations

	Exact	Numerical		
		$alh = 4$	$alh = 8$	
			$N_c = 3$	$N_c = 3$
A_{10}	1.0	0.995794	0.999576	0.999383
C_{10}	0.0	0.007844		
A_{30}	0.0	-0.001936		
A_{10}	0.0	-0.014179	-0.001356	-0.001882
C_{10}	1.0	1.0268	1.00085	1.00137
A_{30}	0.0	-0.008186	-0.00179	-0.001538
A_{50}	0.0			0.00329
E_{10}	1.0	1.00352	1.00011	1.00027

The coefficients not shown in the table and the entries left blank indicate numbers smaller than 0.001. For $alh = 4$ coefficients up to $N_c = 3$ (for a total of 46) can be included in the reconstruction. An attempt to include also the $N_c = 4$ coefficients causes the algorithm to diverge. With a higher resolution, $alh = 8$, the algorithm converges all the way up to $N_c = 8$ (a total of 241 coefficients).

Table 2
Recovery of assigned coefficients (column ‘Exact’) by the numerical method for $alh = 8$

	Exact	Numerical
A_{10}	0.50000	0.497288
C_{10}	0.50000	0.502856
E_{10}	0.50000	0.499959
A_{20}	0.12500	0.121622
C_{20}	0.12500	0.126076
E_{20}	0.12500	0.125050
A_{30}	0.03125	0.027951
C_{30}	0.03125	0.031877
E_{30}	0.03125	0.031340

Here, several coefficients are assigned at the same time; $N_c = 4$.

to $N_c = 8$ (a total of 241 coefficients). In order to compare with the $alh = 4$ results, for this higher-resolution calculation we also truncated at $N_c = 3$ finding an improvement over the $alh = 4$ results, but hardly any difference with the $N_c = 8$ ones, as shown in the next-to-the-last column in Table 1. The results are quite satisfactory and show convergence as the grid is refined. The results of a similar test for assigned coefficients of order (1,1), (2,0) and (3,0) are quite similar and are not shown.

The previous test was made with only one non-zero prescribed coefficient. Table 2 shows similar results when several coefficients are prescribed. For the reasons indicated earlier we prescribe smaller coefficients the higher their order. Here, we used a mesh with $alh = 8$ and we solved for coefficients up to order $N_c = 4$. Again, the results are quite good.

4.2. Uniform flow past a sphere

As a second test we consider the steady flow with a velocity w_0 along the z -axis past a fixed sphere at Reynolds numbers $Re = 2aw_0\nu = 50$ and 100 and compare our results with the spectral ones reported in [60,61] which we use as reference.

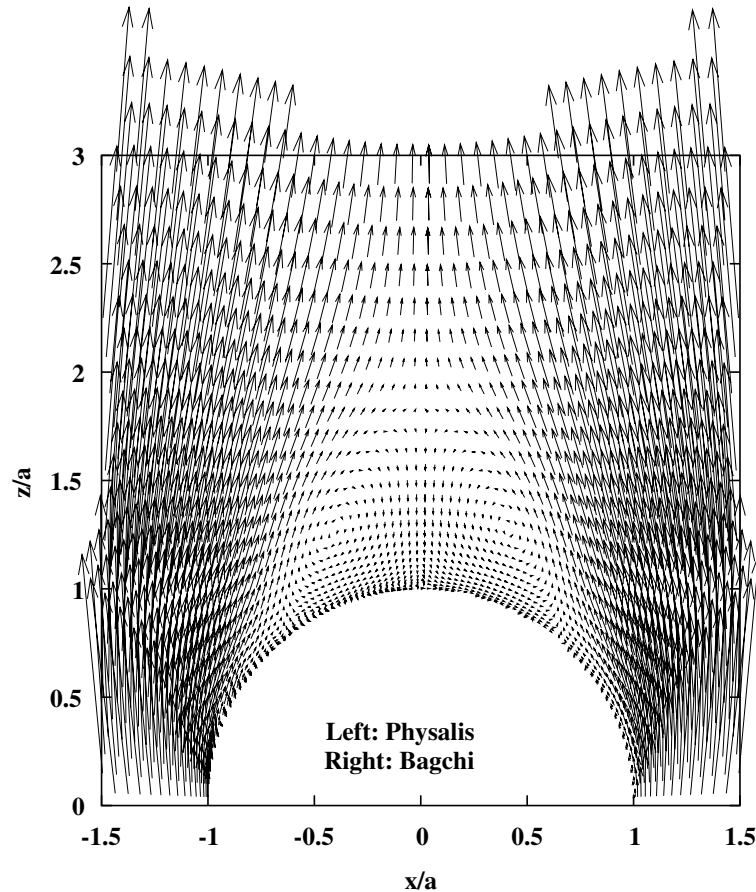


Fig. 5. The flow field in the near wake of a sphere in a uniform incident flow at $Re = 50$ as predicted by the present method (left) and by the spectral calculation of [60,61]. For the present calculation, $alh = 8$, $N_c = 3$ (46 coefficients). The arrows are placed at the collocation points of the spectral calculation shown in Fig. 7; the present results are linearly interpolated to these points from the grid nodes used in our calculation.

For $Re = 50$ we take a computational domain with a size of $20a \times 20a$ in the cross-stream direction and $40a$ in the flow direction. The sphere center is located on the line of intersection of the symmetry planes parallel to the flow velocity (z -axis) $10a$ downstream of the inflow boundary. Periodic boundary conditions are used on the domain boundaries parallel to the flow, a free outflow condition at the downstream boundary, and a prescribed constant velocity condition at the inflow boundary. We use a uniform grid with a spacing $alh = 8$ for a total of $161 \times 161 \times 321$ nodes. According to the estimate given in Section 3, for $Re = 50$, the number of nodes per particle radius should be larger than about 7 and, therefore, the present calculation is somewhat under resolved. It will be seen that nevertheless the results are good.

Our result for the drag coefficient, defined by

$$C_D = \frac{|\mathbf{F}_{hd}|}{\frac{1}{2}\pi a^2 \rho w_0^2} \quad (26)$$

is $C_D = 1.59, 1.58, 1.59, 1.59,$ and 1.58 for $N_c = 2, 3, 4, 5,$ and 6 , respectively, to be compared with the value 1.57 reported in [60,61]. The agreement is quite good, although the relative insensitivity to the order of truncation N_c indicates that C_D is not a very probing measure of the accuracy of the computation. It is therefore more interesting to compare details of the flow field.

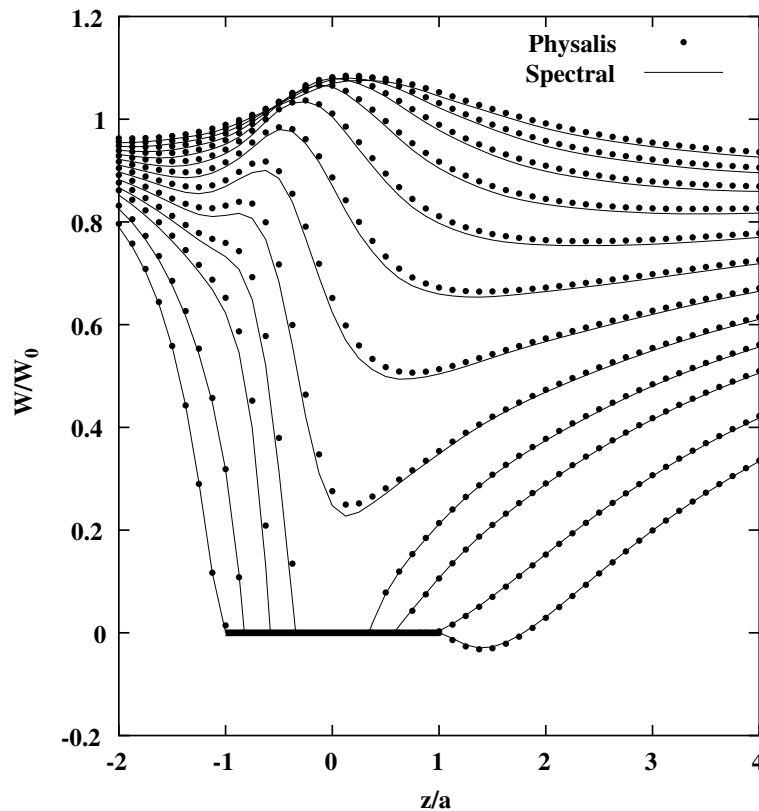


Fig. 6. The normalized velocity w/w_0 in the direction z of the incident flow as a function of z/a for different values of the cross-stream coordinate $x/a = 0.0884, 0.566, 0.815, 0.940, 1.064, 1.189, 1.314, 1.439, 1.564, 1.689, 1.814, 1.939$ for $Re = 50$; $alh = 8$, $N_c = 3$ (46 coefficients). The thick line corresponds to points inside the sphere. The solid lines are the spectral results and the points the present ones.

Fig. 5 shows a comparison of the present results with the spectral ones in the wake region of the sphere with $N_c = 3$ (46 coefficients). No appreciable differences are evident when the two results are juxtaposed in this way. A more detailed view is given in Fig. 6, where the solid lines are the spectral solution for w/w_0 , the z -velocity in the direction of the incident stream normalized by w_0 , the incident velocity, and the symbols the present results. The different lines correspond to $x/a =$ between 0.0884 and 1.938 where x is the cross-stream direction. Although some slight differences can be observed, the agreement is quite good. This result is particularly gratifying when the present grid (crosses) is compared with the collocation points used in the spectral solution (dots) as in Fig. 7. It may also be noted that the present results exhibited an excellent axial symmetry, which was not imposed at the outset. Allowing the particle to rotate resulted only in the exceedingly slow rotation rate $a\Omega/w_0 \sim 10^{-12}$.

Fig. 8 shows the normalized z -velocity w/w_0 at $r/a = 1.020, 1.080, 1.142, 1.222$ as given by the spectral calculation and as computed from the Stokes formulae of Appendix A with the values of the coefficients calculated from the present method with $N_c = 3$. It is seen that the spectral results clearly exhibit a Stokes flow region for $r/a \leq 1.142$ which is very accurately reconstructed in our method. A slight difference becomes apparent only for the outermost line shown, $r/a = 1.222$. In the application of our method, the cage nodes at which the Stokes solution is assumed to give a faithful representation of the flow field are located at slightly different distances r/a from the sphere center along the sphere surface. In the present case, all the cage points are located in the region $r/a \leq 1.077$. However, about 4% of the vorticity nodes used velocity points located between $r/a \leq 1.125$ and 1.24, which can be expected to introduce a small error. Since the system for the Stokes coefficients is greatly over-determined (over 2500 equations for 46 unknowns), these out-lying nodes can be eliminated from the calculation. We have found, however, that this procedure results in only very small differences in the final results.

The convergence of the present results as the order of truncation N_c is increased is illustrated in Figs. 9 and 10. The first figure is the normalized z -velocity w/w_0 at $r/a = 1.0802$ as calculated with different

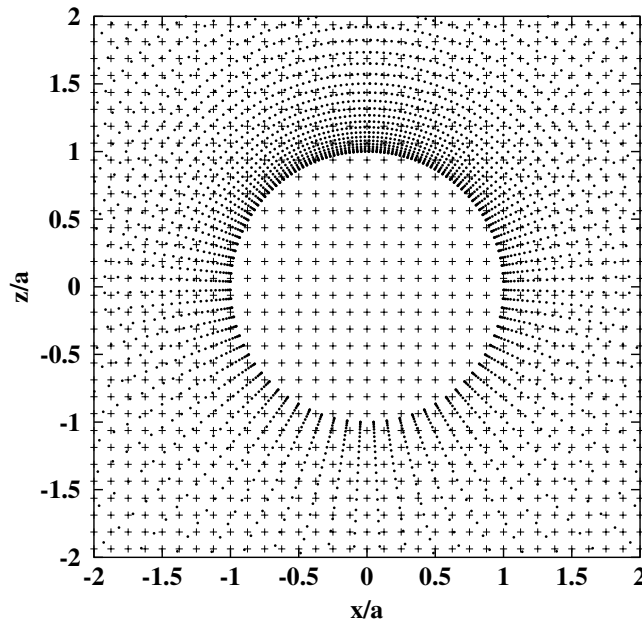


Fig. 7. Comparison between the nodes used in the present simulation for flow past a sphere with $Re = 50$ (crosses) and the collocation points used in the spectral solution of [60,61] (dots).

truncations, $N_c = 1, 2, 3,$ and $4,$ corresponding to $10, 25, 46,$ and 73 coefficients, respectively. Fig. 10 shows the same quantity in greater detail in the wake region. As N_c is increased from 1 to 3 one observes a clear convergence of the present results to the spectral ones. A divergence begins, however, for $N_c = 4$ and gets progressively worse for $N_c = 5$ and 6 (not shown). Exploratory calculations (which we do not describe in detail) suggest that, with a finer grid, the range of N_c where convergence is observed increases which permits the use of a larger N_c with a corresponding improvement of the quality of the solution. In this higher-Reynolds number case, before encountering divergence as N_c is increased, we find a range of N_c values where the solution converges, but its quality deteriorates. Thus, for a given resolution and a given Reynolds number, there is an optimum value of N_c .

The calculation was repeated *without increasing the resolution* for $Re = 100$. With $N_c = 2, 3, 4$ and 5 the drag coefficient was calculated to be $1.10, 1.09, 1.13, 1.09,$ respectively; the value reported in [60,61] is 1.09 . Fig. 11 is similar to Fig. 8 and compares, in the near-region, the spectral result for w/w_0 (lines) with the Stokes result evaluated with the coefficients calculated by the present method with $N_c = 3$. It is clear that a Stokes region still exists, although smaller than before, as expected. Since we use the same number of nodes as before, a larger fraction of them falls in a region where the Stokes flow approximation is not quite accurate. For example, for each vorticity component, there are 170 (24%), 282 (39%), and 650 (90%) nodes inside spheres with radii $r/a = 1.020, 1.056,$ and $1.109,$ respectively. Thus, the present results, cannot be

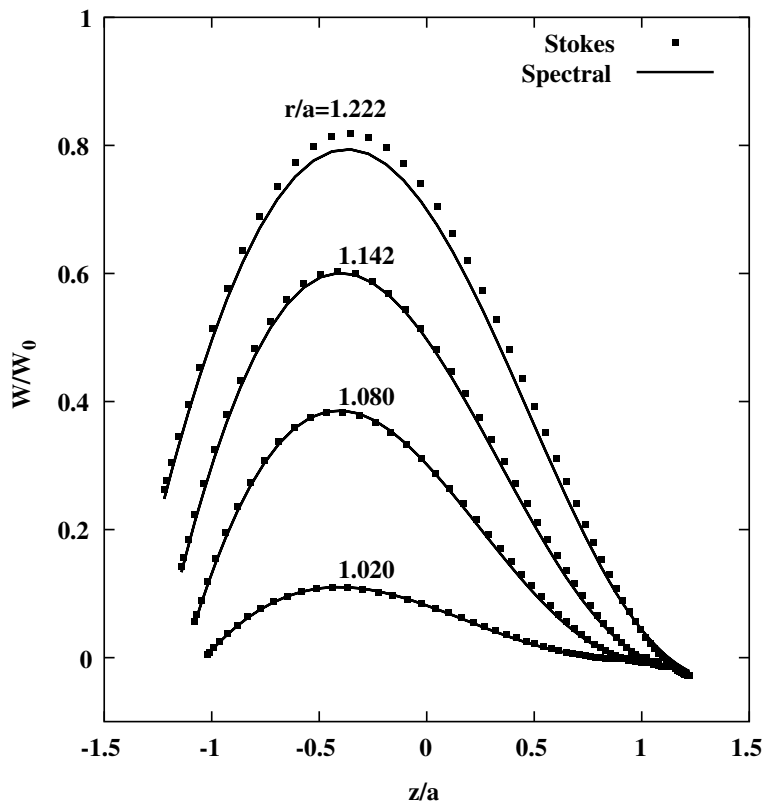


Fig. 8. This figure explicitly shows the existence of a Stokes region in the neighborhood of a sphere at $Re = 50$. The lines are the spectral results for the normalized z -velocity w/w_0 at $r/a = 1.020, 1.080, 1.142, 1.222$; the dots are the corresponding quantity as computed from the Stokes formulae of Appendix A with the values of the coefficients calculated from the present method with $N_c = 3$.

expected to be as accurate as those for $Re = 50$. Nevertheless, even if under-resolved, the method still gives quite acceptable results, not only for the drag coefficient, but also for the entire flow field. An example is shown in Fig. 12 which is analogous to Fig. 5 and compares the flow fields in the wake region. If one compares velocities in detail some differences emerge as shown in Fig. 13.

As explained before, the expansion coefficients are determined from an over-determined system by the SVD algorithm. One can test the quality of the solution by examining the relative error with which each equation of the linear system is satisfied by the calculated coefficients. Specifically, the right-hand side of the linear system for the coefficients consists of the finite-difference results for \tilde{p} and $\tilde{\omega}$. Once the system is solved, we have coefficients which enable us to re-calculate from the expressions in Appendix A the same quantities \tilde{p}_c and $\tilde{\omega}_c$. We define the relative error as $|\tilde{p} - \tilde{p}_c|/\tilde{p}$, and similarly for the vorticity. For $a/h = 8$, there are $N_p = 632$ pressure nodes and $N_\omega = 2166$ vorticity nodes. For $Re = 50$, at steady state, the pressure \tilde{p} exceeds 5% of the maximum value at 612 nodes and, of these, 72 and 40 nodes have errors greater than 10% and 20%, respectively. For the vorticity the corresponding results are 1112 nodes, with 24 and 4 exceeding 10% and 20%.

As expected, the error is much larger for $Re = 100$ due to the relatively coarser resolution used in the calculation which places many nodes out of the Stokes region. We have 604 nodes where the pressure exceeds 5% of the maximum and, of these, 76 and 52 have errors greater than 10% and 20%; for the vorticity, the corresponding values are 1096 nodes, with 276 and 104 exceeding 10% and 20%, respectively.

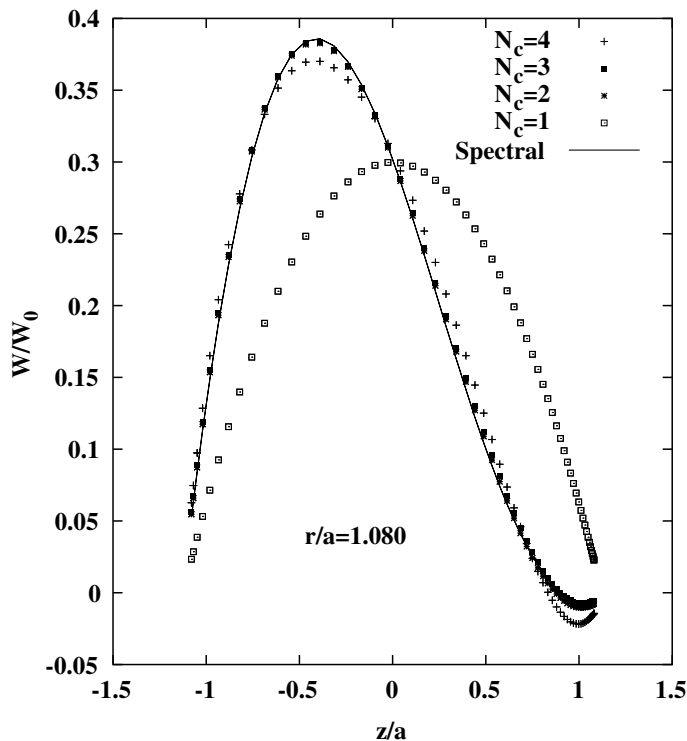


Fig. 9. Convergence of the present results as the order of truncation N_c is increased for flow past a sphere with $Re = 50$. The figure shows the normalized z -velocity w/w_0 at $r/a = 1.0802$ as calculated with different truncations, $N_c = 1, 2, 3$, and 4.

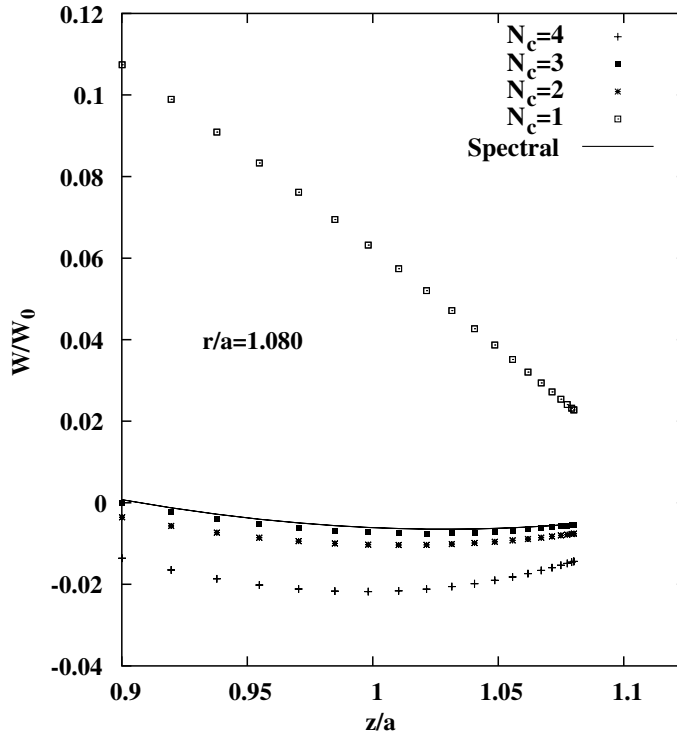


Fig. 10. Detail of the previous figure in the near-particle wake as the truncation order N_c is increased from 1 to 4. The figure shows the normalized z -velocity w/w_0 at $r/a = 1.0802$.

4.3. Flow past periodic sphere arrays

As a third test, we calculate the pressure-driven flow past an infinite simple cubic array of spheres. If the pressure gradient is written as

$$\nabla p = P\mathbf{e}_z + \nabla \hat{p}, \tag{27}$$

with \mathbf{e}_z the unit vector in the z -direction, then \hat{p} is periodic and the calculation can be carried out on a cubic cell of side $L = 4a$ with periodic boundary conditions and the particle positioned at the center of the cell. A simple momentum balance applied to the cell shows that

$$P = \frac{1}{1 - \beta} \frac{F_z}{L^3}, \tag{28}$$

where $\beta = \frac{4}{3}\pi a^3/L^3$, with a the particle radius, is the particle volume fraction in the cell and F_z the component of the total hydrodynamic force in the direction of the imposed pressure gradient. Monitoring the accuracy with which (28) is satisfied gives an indication of the accuracy of the computation.

We consider first the effect of N_c , the number of coefficients retained in the expansions. Two examples are shown in Table 3 where the values of $F_z/[(1 - \beta)PL^3]$ for $P_* \equiv a^3 P/\mu v = 10$ and 20 calculated with $a/h = 8$ are presented; the corresponding Reynolds numbers are 23.7 and 45. When N_c is increased from 1 to 9, convergence failure is observed for $N_c = 8$ or 9, as expected on the basis of the earlier results for the uniform incident flow calculation. The effect of N_c on the accuracy with which the pressure drop is calculated is very small provided N_c exceeds 3. The components of \mathbf{F} orthogonal to the applied pressure gradient are smaller

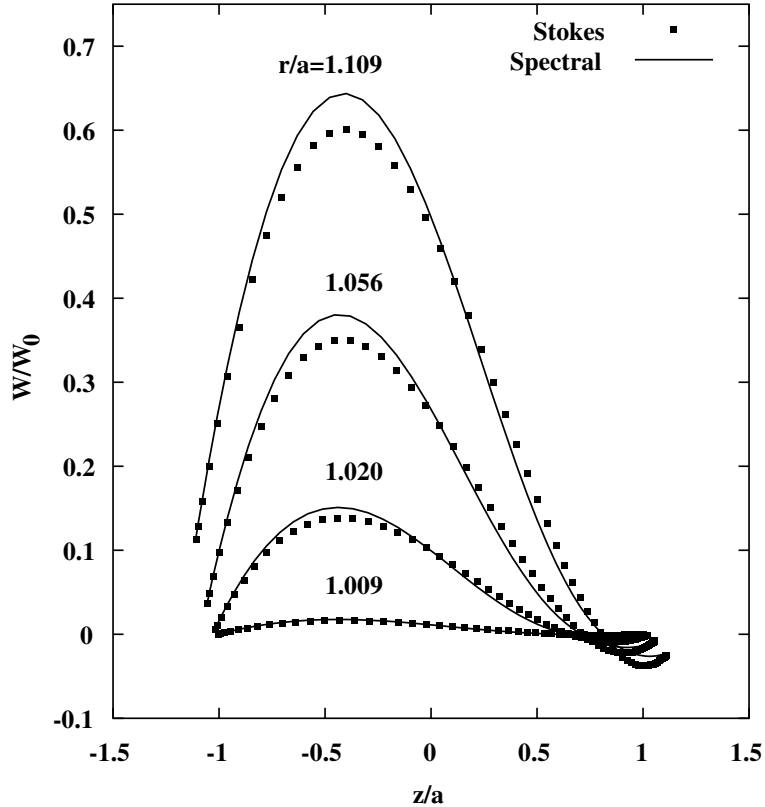


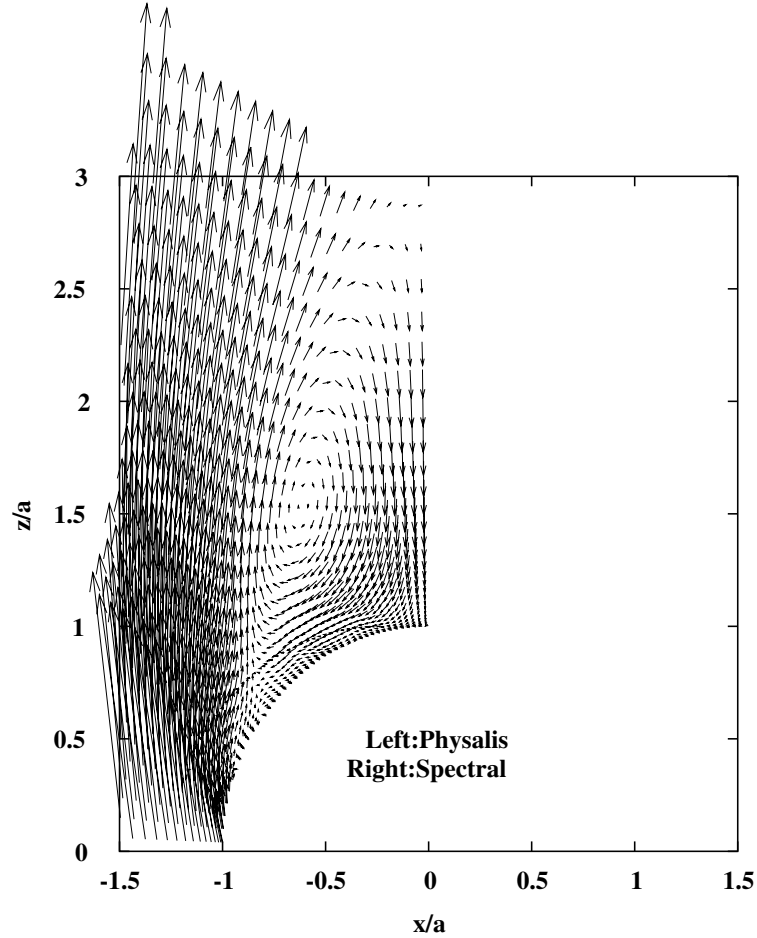
Fig. 11. Existence of a Stokes region in the neighborhood of a sphere at $Re = 100$. The lines are the spectral results for the normalized z -velocity w/w_0 at $r/a = 1.009, 1.020, 1.056,$ and 1.109 ; the dots are the corresponding quantity as computed from the Stokes formulae of Appendix A with the values of the coefficients calculated from the present method with $N_c = 3$.

than P_* by many orders of magnitude and are not given. With $P_* = 5$ ($Re \approx 12.4$) and $alh = 4$, convergence fails for $N_c = 4$. We have checked that these results are independent of which node of the computational domain is chosen to center the particle.

The effect of fixing both $alh = 8$ and $N_c = 4$ while varying the Reynolds number by varying the imposed pressure gradient is shown in Table 4. The results are comparable with the earlier ones.

Another parameter of interest, the effect of which is illustrated in Table 5, is the number of grid points per radius alh . For $Re \approx 12$, there is little improvement as alh is varied from 4 to 16. For $Re \approx 24$, the benefit is very noticeable and especially dramatic for $Re \approx 45$ where the iteration process for $alh = 4$ does not even converge. This result illustrates the comment made earlier in Section 3 that a check of the adequacy of the discretization is built into the method as convergence requires that all the flow fields be described by the same set of coefficients, which would only be true in the Stokes region.

In Section 3.3, we mentioned the possibility of using a set of “standard cages”, choosing the one centered closest to the particle center, in place of re-building a new cage for every new particle position. The present problem offers a good way to examine the error associated with this practice as, in theory, the result should be completely independent of particle position. Table 6 shows the error with which (28) is satisfied for various positions of the particle center relative to a cage centered at a grid node for $a^3 P/(\mu\nu) = 10$, $N_c = 4$, and $alh = 8$. The biggest error, slightly greater than 3%, is found when the cage center is farthest from the particle center. Here, the cage is centered at the center of the computational domain.



A similar test can be made moving the cage center relative to the particle, and moving both particle and cage. The results shown in [Table 7](#) are similar to the previous ones. It is interesting to note in these two tables that results corresponding to equal displacements of particle or cage in different directions give slightly different results given that the imposed pressure gradient breaks complete isotropy.

4.4. Falling particle

In all the previous examples the particle position was fixed. We now consider the case of a sphere freely falling from rest. The motion of the sphere is calculated from [\(24\)](#) and [\(25\)](#) with the hydrodynamic force and torque obtained from [\(15\)](#) and [\(16\)](#). For this calculation the computational domain is a cube with a side of $4a$; periodicity conditions are imposed in the z -direction, i.e. the direction of fall, while no-slip conditions are imposed on the vertical boundaries. For this calculation we used a single cage, the center of which was displaced to a neighboring node whenever the particle center left a cube of side h around

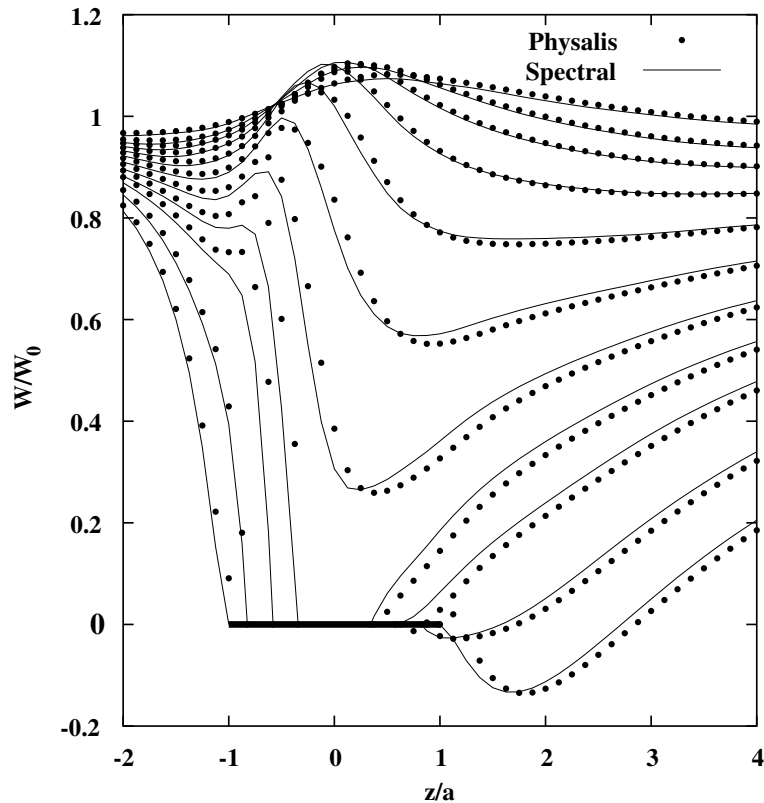


Fig. 13. The normalized velocity w/w_0 in the direction z of the incident flow as a function of z/a for different values of the cross-stream coordinate $x/a = 0.0884, 0.566, 0.815, 0.940, 1.064, 1.189, 1.314, 1.439, 1.564, 1.689, 1.939$ for $Re = 100$. The thick line corresponds to points inside the sphere. The solid lines are the spectral results and the points the present ones; $N_c = 3$, $alh = 8$.

the current cage center. The ratio of the particle to the fluid density is 2 and the Reynolds number based on the terminal velocity is about 22. The particle is released at $t = 0$ at the center of the domain. The time step is $v\Delta t/a^2 = 0.004$ and $N_c = 2$.

Table 3

Effect of increasing the number of coefficients in the expansions for the pressure-driven flow past a simple cubic array of spheres with $a^3 P/\mu v = 10$ and 20 ($R_c \approx 23.7$ and 45) and 8 cells per particle radius

N_c	$\frac{F_z}{(1-\beta)PL^3}$	
	$a^3 P/\mu v = 10$	$a^3 P/\mu v = 20$
1	1.0136	1.0190
2	1.0097	1.0110
3	0.99980	1.0002
4	0.99949	0.99830
5	0.99952	0.99944
6	0.99969	0.99594
8	1.0006	No convergence
9	No convergence	—

The exact result for the quantity shown is 1.

Table 4

Accuracy vs. Reynolds number for the pressure-driven flow past a cubic array of spheres with $N_c = 4$, and 8 grid nodes points per particle radius

P^*	Re	$\frac{F_z}{(1-\beta)PL^3}$
1	2.55	1.000
5	12.30	1.000
10	23.76	1.000
20	45.43	0.999
50	104.46	1.0041
100	–	No convergence

The exact value is 1.

Table 5

Effect of increasing the grid resolution for the pressure-driven flow past a cubic array of spheres for $\alpha^3 P/\mu\nu = 5, 10$, and 20

P_*	alh	$\frac{F_z}{(1-\beta)PL^3}$	N_c	R_c
5	4	0.99428	2	12.5556
5	8	1.0000	4	12.2978
5	16	0.99982	4	12.2810
10	4	0.97831	2	24.4429
10	8	0.99949	4	23.7575
10	16	0.99981	4	23.7044
20	4	No convergence	2	–
20	8	0.99428	4	45.4344
20	16	0.99996	4	45.2917

Table 6

Normalized hydrodynamic force on a particle the center of which is displaced away from the cage center for $\alpha^3 P/\mu\nu = 10$ ($Re \approx 23.7$), $N_c = 4$, and 8 grid nodes per particle radius

Particle position	$\frac{F_z}{(1-\beta)PL^3}$	$\frac{F_z}{(1-\beta)PL^3}$	$\frac{F_z}{(1-\beta)PL^3}$
0.0, 0.0, 0.0	0.0000	0.0000	0.9995
0.0, 0.0, $\frac{1}{2}h$	0.0000	0.0000	0.9946
$\frac{1}{3}h, 0.0, 0.0$	–0.01427	0.0000	0.9984
$\frac{1}{2}h, 0.0, \frac{1}{2}h$	–0.01668	0.0000	0.9918
$-\frac{1}{2}h, -\frac{1}{2}h, -\frac{1}{2}h$	0.01929	–0.01929	0.9889
$-\frac{1}{4}h, -\frac{1}{4}h, -\frac{1}{4}h$	–0.00690	–0.00690	0.9967
$\frac{1}{4}h, \frac{1}{4}h, \frac{1}{4}h$	0.00636	0.00636	1.0008
$\frac{1}{2}h, \frac{1}{2}h, \frac{1}{2}h$	0.01758	0.01758	1.0341

The exact results for the three columns are 0, 0, and 1, respectively.

A point to address when the particles move is that nodes which were inside the particle at some instant, may become cage or fluid nodes one or two time steps later. As shown in Section 3.1, our method generates the pressure field at time level $n + \frac{1}{2}$ rather than $n + 1$. In order to calculate the coefficients at a consistent time level, we estimate p^{n+1} at the cage nodes by linear extrapolation using $p^{n-1/2}$ and $p^{n+1/2}$. This procedure requires that $p^{n-1/2}$ be known at all cage points. Similarly, the Adams–Bashforth method for the

Table 7

Normalized hydrodynamic force on a particle the center of which is displaced away from the cage center for $a^3 P/\mu v = 10$ ($Re \approx 23.7$), $N_c = 4$, and 8 grid nodes per particle radius

Particle position	Cage position	$\frac{F_z}{(1-\beta)PL^3}$	$\frac{F_z}{(1-\beta)PL^3}$	$\frac{F_z}{(1-\beta)PL^3}$
0.0, 0.0, 0.0	0.0, 0.0, 0.0	0.0000	0.0000	0.9995
0.0, 0.0, 0.0	$\frac{1}{2}h, 0.0, 0.0$	-0.01719	0.0000	0.9992
0.0, 0.0, 0.0	$\frac{1}{2}h, 0.0, \frac{1}{2}h$	-0.02235	0.0000	0.9903
$\frac{1}{2}h, \frac{1}{2}h, \frac{1}{2}h$	h, h, h	-0.01929	-0.01929	0.9889
0.0, 0.0, 0.0	$\frac{1}{2}h, \frac{1}{2}h, \frac{1}{2}h$	-0.02751	-0.02751	0.9872
0.0, 0.0, $\frac{1}{2}h$	0.0, 0.0, $\frac{1}{2}h$	0.0000	0.0000	0.9987
$\frac{1}{2}h, 0.0, 0.0$	$\frac{1}{2}h, 0.0, 0.0$	-0.0000	0.0000	0.9999
$\frac{1}{2}h, 0.0, \frac{1}{2}h$	$\frac{1}{2}h, 0.0, \frac{1}{2}h$	-0.0000	0.0000	0.9989
$\frac{1}{2}h, \frac{1}{2}h, \frac{1}{2}h$	$\frac{1}{2}h, \frac{1}{2}h, \frac{1}{2}h$	0.0000	-0.0000	0.9995

Except for the first line, neither the particle nor the cage are simultaneously centered at a grid node. The exact results for the three columns are 0, 0, and 1, respectively.

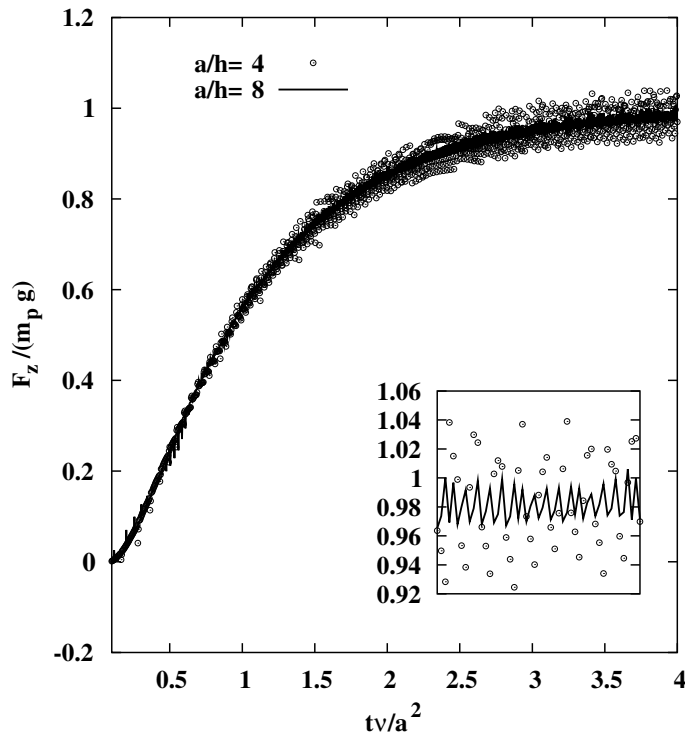


Fig. 14. Normalized z -component of the instantaneous hydrodynamic force on a falling particle vs. dimensionless time tv/a^2 for $N_c = 2$ and $a/h = 4$ (symbols) and 8 (line). The Reynolds number based on the terminal velocity is about 22.

convective terms requires the values of \mathbf{u}^{n-1} . To ensure that the correct information is available, at the end of each time step we use the analytic pressure solution to assign correct pressure and velocity values to a layer of nodes inside the particle adjacent to the current cage nodes.⁴

⁴ As long as the update of the particle variables is part of the iteration, this step has minor consequences. If, however, the particle variables are updated after the flow field, this proves to be a necessary step.

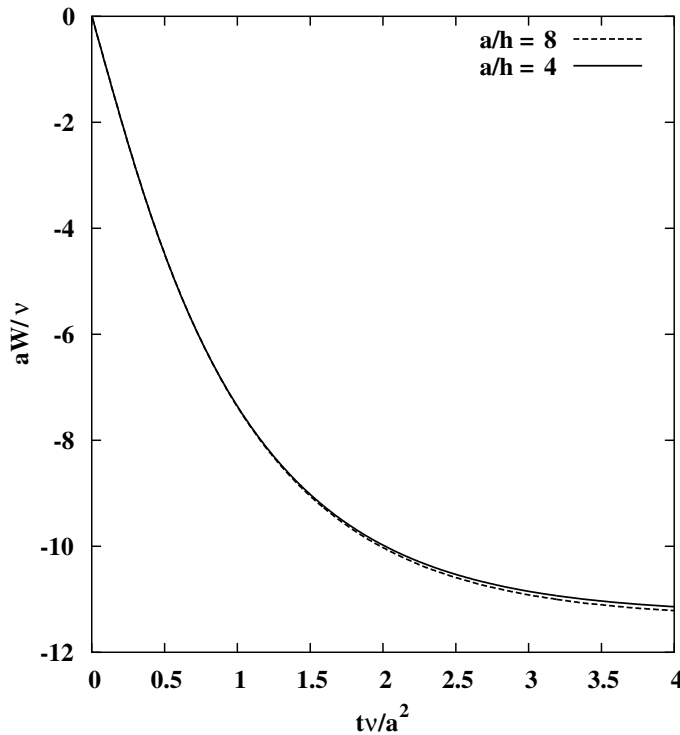


Fig. 15. Normalized z -component of the instantaneous velocity of a falling particle vs. dimensionless time vt/a^2 for $N_c = 2$ and $ah = 4$ and 8. The Reynolds number based on the terminal velocity is about 22.

Fig. 14 shows a graph of the z -component of the instantaneous hydrodynamic force on the particle, normalized by its weight mg , as a function of the dimensionless time vt/a^2 for $N_c = 2$ and $ah = 4$ and 8. As in all fixed-grid methods, the force is found to fluctuate somewhat as the sphere moves with respect to the underlying grid. The fluctuations are rather severe for the lowest resolution, $ah = 4$, but are greatly reduced for $ah = 8$. For all resolutions, the fluctuation amplitude at first increases as the velocity increases, and then saturates once the terminal velocity is approached. As may be expected from the absence of bias in the fluctuations, the velocity, shown in Fig. 15, is smooth and essentially the same for both resolutions.

It is interesting to note that the trajectory of the particle center keeps on the centerline of the computational domain to machine precision.

5. An application to turbulent disperse flow

As a final illustration of the capabilities of the code we consider 100 particles in a weak decaying turbulent field in a periodic cubic cell of side L .

To start the calculation, we use a spectral single-phase code to generate homogeneous forced isotropic turbulence [62]. The size of the domain used with this code, discretized with 32^3 cells, is $(16a)^3$ and the Taylor-microscale Reynolds number at steady state approximately 29. After doubling the grid size in each direction (for a total of 64^3 cells), we introduce in this turbulent flow, at random positions, 100 spheres avoiding overlap and attributing to each one an initial velocity equal to the average fluid velocity in the

volume it occupies. The sphere density ρ_p is taken as 1.02 times the fluid density, with a Stokes number of 2.6. The volume fraction occupied by the spheres is about 10%. Gravity is disregarded and the collisions modeled as elastic. At the initial time the particle radius is approximately 4.16 times the Kolmogorov scale and 0.389 times the Taylor microscale.

In order to be able to run the simulation with the memory available on a PC, we used a mesh spacing h (equal in all directions) such that ah was only equal to 4, which forced us to use a small number of coefficients, $N_c = 1$ (10 coefficients per particle). To improve efficiency, we also used the “standard cages” idea described in Section 3.3, with a set of $9^3 = 729$ different cages centered at 9^3 points in the neighborhood of a grid point. At the start of the simulation the particles were slightly displaced so that their center coincided with the closest center of a standard cage. When the calculation was terminated the Taylor Reynolds number was about 14.

The left panel in Fig. 16 shows the fluid kinetic energy at the start of the calculation $t = 0$; a layer of fluid has been removed to give a better sense of the volume occupied by the particles. A similar depiction of the fluid kinetic energy after 1 eddy turn-over time is shown in the right panel. Here, a gray scale is used with the magnitude of the kinetic energy decreasing from the dark areas bordered by a narrow very light band, to light, to dark again.

We are unaware of other simulations of this type with which to compare our results. Thus, for comparison, we used a point-particle model in spite of the fact that, at such a large volume fraction, its justification may be somewhat tenuous. The code used for this purpose was adapted from that described in [63]. It solves the Navier–Stokes momentum equation augmented by the effect of the particles:

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = -\nabla p + \mu \nabla^2 \mathbf{u} - \sum_{\alpha=1}^N \mathbf{f}^\alpha, \quad (29)$$

where \mathbf{f}^α is the total hydrodynamic force on the α th particle

$$m \frac{d\mathbf{w}}{dt} = \mathbf{f} = m \frac{(\mathbf{u} - \mathbf{w})}{\tau_p} + \rho v \frac{D\mathbf{u}}{Dt} + \frac{1}{2} \rho v \left[\frac{D\mathbf{u}}{Dt} - \frac{d\mathbf{w}}{dt} \right] + \frac{1}{2} \rho v (\mathbf{u} - \mathbf{w}) \times (\nabla \times \mathbf{u}), \quad (30)$$

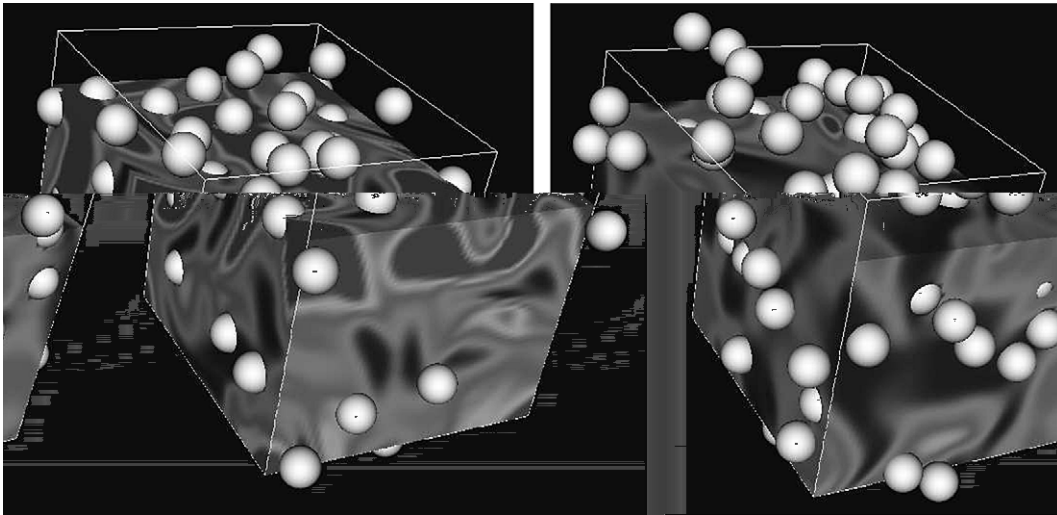


Fig. 16. Fluid kinetic energy at the start of the turbulence simulations (left) and after 1 eddy turn-over time (right); a layer of fluid has been removed to show some of the 100 suspended particles. The magnitude of the kinetic energy decreases from the dark areas bordered by a narrow, very light band, to light, to dark again.

where

$$\frac{1}{\tau_p} = \frac{3}{8a} C_D \frac{\rho}{\rho_p} |\mathbf{u} - \mathbf{w}|, \quad C_D = \frac{24}{Re_p} + \frac{6}{1 + \sqrt{Re_p}} + 0.4, \quad Re_p = \frac{2a|\mathbf{u} - \mathbf{w}|}{\nu}. \quad (31)$$

Here τ_p is the particle relaxation time, $v = \frac{4}{3}\pi a^3$ the particle volume, Re_p the particle Reynolds number, and C_D the drag coefficient; the Basset force is neglected. In Eq. (30) the force is interpolated to the grid nodes in the usual way.

The maximum value of Re_p in the course of the simulation was about 40. It is not obvious how to define a corresponding quantity for the extended-particle calculation because in this case there is no well-defined “undisturbed” fluid velocity \mathbf{u} . The analogous quantity $2a|\mathbf{w}|/\nu$ based on the particle velocity has a maximum value of about 80. However, since the density of the particle is only 2% greater than that of the fluid, it is likely that the fluid-particle relative velocity is smaller than $|\mathbf{w}|$, which shows that the previous estimate is most likely a very conservative upper limit.

Fig. 17 compares the turbulent kinetic energy, normalized by the initial value, vs. time, normalized by the eddy turn-over time, as obtained from our simulation (solid line) with the point-particle result (29), with and without two-way coupling, i.e., including (dotted line) or excluding (dashed line) the last term in this equation. As expected, two-way coupling increases the dissipation rate over that of the pure fluid, and the extended particles increase it further.

Fig. 18 shows the square of the mean particle displacement from the initial position vs. time as obtained from our simulation (solid line) with the point-particle result (29) with (dotted line) and without (dashed

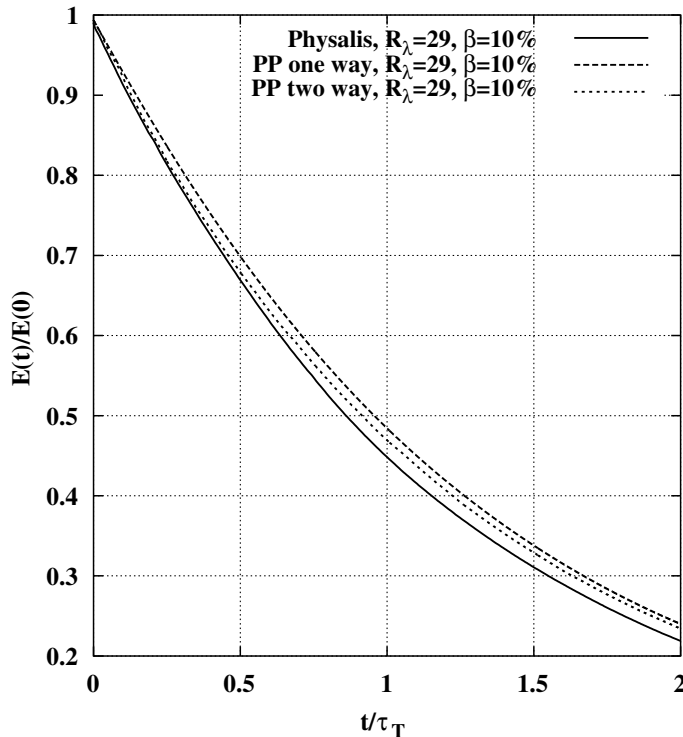


Fig. 17. Time dependence of the turbulent kinetic energy (normalized by the initial value) vs. time (normalized by the eddy turn-over time). Solid line: extended particles; dotted line: two-way coupling; dashed line: one-way coupling.

line) two-way coupling. The two simulations with a coupling between the particles and the fluid show the expected long-time behavior with the square of the mean particle displacement a linear function of time. The extended particles, however, diffuse slightly more slowly than the point particles, as expected.

A way to illustrate the effect of turbulence on the distribution of the particles is to study the structure factor (see e.g. [64, p. 7]; [65, p. 33]) integrated over the angular variables:

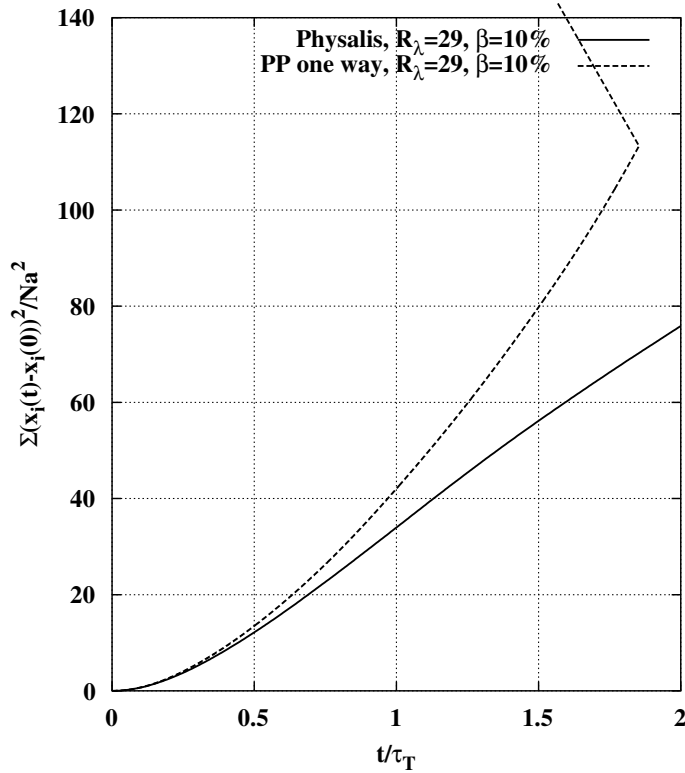
$$S(k) = 1 + \frac{1}{N} \sum_{\substack{\alpha, \beta=1 \\ \alpha \neq \beta}}^N \int d\Omega \cos \mathbf{k} \cdot (\mathbf{y}^\alpha - \mathbf{y}^\beta). \quad (32)$$

Here, \mathbf{y} denotes the position of the particle center and $\mathbf{k} = (2\pi/L)(i, j, k)$, with i, j, k integers between 0 and 8 (not all simultaneously 0) and $k = |\mathbf{k}|$. The structure factor is the Fourier transform of the two-particle distribution function

$$g(r) = 1 + \frac{L^3}{2\pi^2 N r} \int dk [S(k) - 1] k \sin kr \quad (33)$$

and, accordingly, it gives information on the mutual distance between the particles. Loosely speaking, one may say that maxima of $S(k)$ correspond to a higher probability of finding particle pairs separated by $2\pi/k$.

Fig. 19 shows $S(k)$ as a function of k/k_{\min} , with $k_{\min} = 2\pi/L$, after 2 eddy turn-over times for the extended particles and the point-particle model with and without two-way coupling; the thick dotted line is the hard-sphere distribution [66–68]. Results corresponding to the lower values of $kL/2\pi$ are strongly affected by the



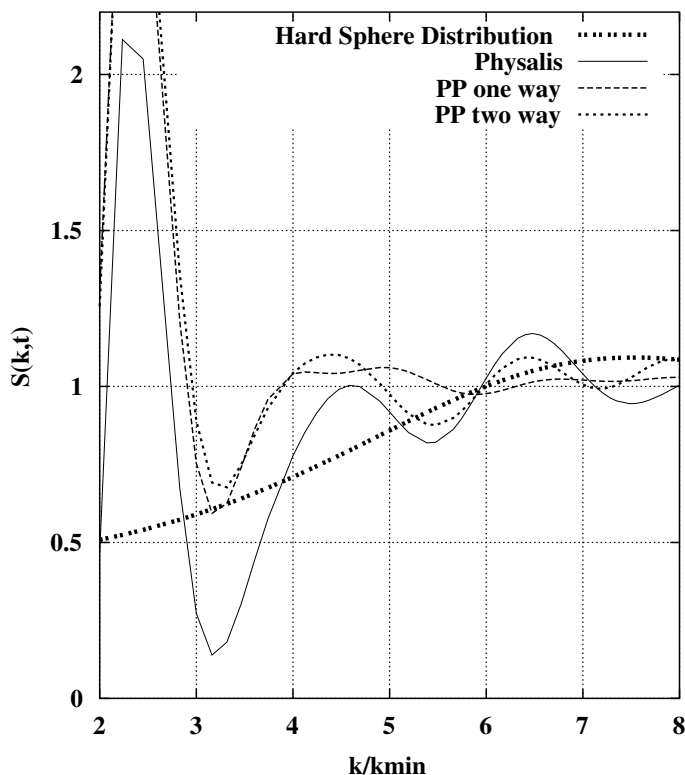


Fig. 19. Structure factor $S(k)$ as a function of k/k_{\min} , with $k_{\min} = 2\pi/L$, after 2 eddy turn-over times. Solid line: extended particles; dotted line: two-way coupling; dashed line: one-way coupling; thick dotted line: hard-sphere distribution. Touching particles correspond to $k/k_{\min} = 8$. The ordering of the maxima near $k/k_{\min} = 6.5$ suggests, in the mean, a tendency of the extended particles to cluster more than the point particles.

artificial periodicity imposed on the system; touching particles correspond to $k/k_{\min} = 8$. One notices that the maximum for $k/k_{\min} \approx 6.5$ is greater for the extended-particle simulation (solid line) than for the two-way coupled, point-particle model (dotted line), which suggests, in the mean, a tendency of the extended particles to cluster more than the point particles. For $k/k_{\min} \geq 4$, $S(k)$ for the one-way coupled point particles is close to 1 which, as shown by Eq. (33), implies a close-to-uniform distribution. The fact that this constant value is achieved for $k/k_{\min} \approx 4$ gives an idea of the range of k -values affected by the artificial periodicity.

In view of the limited accuracy of this calculation (the purpose of which is more to illustrate the applicability of the method than to obtain specific results) one may perhaps not want to make too much of these results. However, it may be noted that, even with an elastic collision, two extended particles would produce a relatively large viscous dissipation as they approach and push out the fluid that separates them.

6. Conclusions

We have described a computational method for the simulation of Navier–Stokes flow with suspended spherical particles. We have devoted a considerable effort to the validation of the method against independent accurate results and have found it to be efficient and accurate.

A useful feature of our approach is that the degrees of freedom attributed to each particle are used in a nearly-optimal way, as the summations in which the coefficients appear converge spectrally for smooth flow

fields. Another significant advantage is that the force and torque on each particle are found directly from the low-order coefficients as shown in (15) and (16). This avoids the difficulty encountered with some other methods which require high-order extrapolations to obtain the stress distribution on the particle.

The accuracy of the method can be increased arbitrarily by increasing the number of nodes with the effect, among others, of reducing the extent of the region where applicability of the Stokes equations is assumed. This procedure permits to increase the number of coefficients retained in the exact solution.

A limitation of our approach is the reliance on the existence of an exact solution of the Stokes equations for the particular body shape considered. Such solutions are available for cylinders and spheres. A possible way to deal with more general body shapes would be to use a boundary integral solution of the Stokes equations (see e.g. [69]) in the thin region between the body and the cage. Spectral convergence in this case would however not be automatic and would depend on the solution procedure of the integral equations. Another limitation stems from the explicit use of the no-slip condition to effect a local linearization about solid body motion. Unlike other methods capable of dealing with rigid and free complex boundaries, it does not appear possible to extend the present method to problems with free surfaces.

Ref. [70] presents some preliminary results obtained with a first-order time-accurate version of the present method implemented on a parallel computer. Some further validation tests are provided, and a simulation of the settling of 1024 spheres described.

Acknowledgments

We are greatly indebted to Drs. S. Balachandar and P. Bagchi for giving us their excellent results for spheres in a uniform flow. Drs. D. Lohse and I. Mazzitelli kindly gave us their point-particle code. We are also indebted with Dr. C. Meneveau and Mr. S. Chester for giving us their homogeneous turbulence code.

This study has been supported by NSF under Grant CTS-0210044 and DOE DE-FG02-99ER14966.

Appendix A. Detailed formulae for pressure, vorticity, and velocity

From (7) and (8) expressions for the Stokes pressure, velocity, and vorticity fields readily follow. We present these expressions in terms of dimensionless quantities defined by

$$s = \frac{r}{a}, \quad p^* = \frac{a^2}{\mu v} \tilde{p}, \quad \mathbf{u}^* = \frac{a}{v} \tilde{\mathbf{u}}, \quad \boldsymbol{\omega}^* = \frac{a^2}{v} \tilde{\boldsymbol{\omega}}. \quad (\text{A.1})$$

The pressure field is given by

$$p^* = p_0 + \sum_{n=0}^{\infty} \left\{ \left[s^n - \frac{n(2n-1)}{2(n+1)} s^{-n-1} \right] \sum_{m=0}^n [P_{nm} \cos(m\phi) + \tilde{P}_{nm} \sin(m\phi)] P_n^m(\cos \theta) - \frac{n(4n^2-1)}{n+1} s^{-n-1} \sum_{m=0}^n [\Phi_{nm} \cos(m\phi) + \tilde{\Phi}_{nm} \sin(m\phi)] P_n^m(\cos \theta) \right\}. \quad (\text{A.2})$$

The velocity components follow from (6) as:

$$u_r^* = \sum_{n=1}^{\infty} \left\{ \left[\frac{n}{2(2n+3)} s^{n+1} - \frac{n}{4} s^{-n} + \frac{n(2n+1)}{4(2n+3)} s^{-n-2} \right] \sum_{m=0}^n [P_{nm} \cos(m\phi) + \tilde{P}_{nm} \sin(m\phi)] P_n^m(\cos \theta) + \left[ns^{n-1} + \frac{n(2n+1)}{2} s^{-n} + \frac{n(2n-1)}{2} s^{-n-2} \right] \sum_{m=0}^n [\Phi_{nm} \cos(m\phi) + \tilde{\Phi}_{nm} \sin(m\phi)] P_n^m(\cos \theta) \right\}, \quad (\text{A.3})$$

$$\begin{aligned}
 u_\theta^* = & \sum_{n=1}^{\infty} \left\{ \left[\frac{n+3}{2(n+1)(2n+3)} s^{n+1} - \frac{n-2}{4(n+1)} s^{-n} - \frac{n(2n+1)}{4(n+1)(2n+3)} s^{-n-2} \right] \right. \\
 & \times \sum_{m=0}^n [P_{nm} \cos(m\phi) + \tilde{P}_{nm} \sin(m\phi)] \frac{dP_n^m(\cos \theta)}{d\theta} + \left[s^{n-1} + \frac{(n-2)(2n+1)}{2(n+1)} s^{-n} - \frac{n(2n-1)}{2(n+1)} s^{-n-2} \right] \\
 & \times \sum_{m=0}^n [\Phi_{nm} \cos(m\phi) + \tilde{\Phi}_{nm} \sin(m\phi)] \frac{dP_n^m(\cos \theta)}{d\theta} \\
 & \left. + (s^n - s^{-n-1}) \sum_{m=0}^n [-mX_{nm} \cos(m\phi) + mX_{nm} \sin(m\phi)] \frac{P_n^m(\cos \theta)}{\sin \theta} \right\}, \tag{A.4}
 \end{aligned}$$

$$\begin{aligned}
 u_\phi^* = & \sum_{n=1}^{\infty} \left\{ \left[\frac{n+3}{2(n+1)(2n+3)} s^{n+1} - \frac{n-2}{4(n+1)} s^{-n} - \frac{n(2n+1)}{4(n+1)(2n+3)} s^{-n-2} \right] \right. \\
 & \times \sum_{m=0}^n [-mP_{nm} \sin(m\phi) + m\tilde{P}_{nm} \cos(m\phi)] \frac{P_n^m(\cos \theta)}{\sin \theta} + \left[s^{n-1} + \frac{(n-2)(2n+1)}{2(n+1)} s^{-n} - \frac{n(2n-1)}{2(n+1)} s^{-n-2} \right] \\
 & \times \sum_{m=0}^n [-m\Phi_{nm} \sin(m\phi) + m\tilde{\Phi}_{nm} \cos(m\phi)] \frac{P_n^m(\cos \theta)}{\sin \theta} \\
 & \left. - (s^n - s^{-n-1}) \sum_{m=0}^n [X_{nm} \cos(m\phi) + \tilde{X}_{nm} \sin(m\phi)] \frac{dP_n^m(\cos \theta)}{d\theta} \right\}. \tag{A.5}
 \end{aligned}$$

The vorticity components are, from (12):

$$\omega_r^* = \sum_{n=1}^{\infty} \left\{ n(n+1)(s^{n-1} - s^{-n-2}) \sum_{m=0}^n [X_{nm} \cos(m\phi) + \tilde{X}_{nm} \sin(m\phi)] P_n^m(\cos \theta) \right\}, \tag{A.6}$$

$$\begin{aligned}
 \omega_\theta^* = & \sum_{n=1}^{\infty} \left\{ - \left[\frac{s^n}{n+1} + \frac{2n-1}{2(n+1)} s^{-n-1} \right] \sum_{m=0}^n [-mP_{nm} \sin(m\phi) + m\tilde{P}_{nm} \cos(m\phi)] \frac{P_n^m(\cos \theta)}{\sin \theta} \right. \\
 & - \frac{4n^2-1}{n+1} s^{-n-1} \sum_{m=0}^n \frac{1}{\sin \theta} [-m\Phi_{nm} \sin(m\phi) + m\tilde{\Phi}_{nm} \cos(m\phi)] \frac{P_n^m(\cos \theta)}{\sin \theta} \\
 & \left. + [(n+2)s^{n-1} + (n-1)s^{-n-2}] \sum_{m=0}^n [X_{nm} \cos(m\phi) + \tilde{X}_{nm} \sin(m\phi)] \frac{dP_n^m(\cos \theta)}{d\theta} \right\}, \tag{A.7}
 \end{aligned}$$

$$\begin{aligned}
 \omega_\phi^* = & \sum_{n=1}^{\infty} \left\{ \left[\frac{s^n}{n+1} + \frac{2n-1}{2(n+1)} s^{-n-1} \right] \sum_{m=0}^n [P_{nm} \cos(m\phi) + \tilde{P}_{nm} \sin(m\phi)] \frac{dP_n^m(\cos \theta)}{d\theta} \right. \\
 & + \frac{4n^2-1}{n+1} s^{-n-1} \sum_{m=0}^n [\Phi_{nm} \cos(m\phi) + \tilde{\Phi}_{nm} \sin(m\phi)] \frac{dP_n^m(\cos \theta)}{d\theta} \\
 & \left. + [(n+2)s^{n-1} + (n-1)s^{-n-2}] \sum_{m=0}^n [-mX_{nm} \sin(m\phi) + m\tilde{X}_{nm} \cos(m\phi)] P_n^m(\cos \theta) \right\}. \tag{A.8}
 \end{aligned}$$

References

- [1] S. Takagi, H. Oğuz, Z. Z. A. Prosperetti, PHYSALIS: A new method particle simulation. Part II: Two-dimensional Navier–Stokes flow around cylinders, *J. Comput. Phys.* 187 (2003) 371–390.
- [2] Z. Zhang, A. Prosperetti, A method for particle simulation, *J. Appl. Mech.* 70 (2003) 64–74.
- [3] S. Elghobashi, T. G.C., On the two-way interaction between homogeneous turbulence and dispersed solid particles. I: Turbulence modification, *Phys. Fluids A* 5 (1993) 1790–1801.
- [4] C. Crowe, T. Troutt, J. Chung, Numerical models for two-phase turbulent flows, *Ann. Rev. Fluid Mech.* 28 (1996) 11–41.
- [5] Y. Pan, S. Banerjee, Numerical simulation of particle interactions with wall turbulence, *Phys. Fluids* 8 (1996) 2733–2755.
- [6] S. Sundaram, L. Collins, A numerical study of the modulation of isotropic turbulence by suspended particles, *J. Fluid Mech.* 379 (1999) 105–143.
- [7] V. Armenio, U. Piomelli, V. Fiorotto, Effect of the subgrid scales on particle motion, *Phys. Fluids* 11 (1999) 3030–3042.
- [8] M. Boivin, O. Simonin, K. Squires, On the prediction of gas–solid flows with two-way coupling using large eddy, *Phys. Fluids* 12 (2000) 2080–2090.
- [9] Y. Yamamoto, M. Potthoff, T. Tanaka, T. Kajishima, Y. Tsuji, Large-eddy simulation of turbulent gas-particle flow in a vertical channel: effect of considering inter-particle collisions, *J. Fluid Mech.* 442 (2001) 303–334.
- [10] A. Fortes, D. Joseph, T. Lundgren, Nonlinear mechanics of fluidization of beds of spherical particles, *J. Fluid Mech.* 177 (1987) 467–483.
- [11] H. Hu, D. Joseph, M. Crochet, Direct simulation of fluid-particle motions, *Theor. Comput. Fluid Dynam.* 3 (1992) 285–306.
- [12] R. Glowinski, T. Pan, T. Hesla, D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, *Int. J. Multiphase Flow* 25 (1999) 755–794.
- [13] N. Patankar, P. Singh, D. Joseph, R. Glowinski, T.-W. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, *Int. J. Multiphase Flow* 26 (2000) 1509–1524.
- [14] P. Singh, T. Hesla, D. Joseph, Distributed Lagrangian multiplier method for particulate flows with collisions, *Int. J. Multiphase Flow* 29 (2001) 495–509.
- [15] S. Dong, D. Liu, M.R. Maxey, G.E. Karniadakis, Spectral distributed Lagrange multiplier method: algorithm and benchmark tests, *J. Comput. Phys.* 195 (2004) 695–717.
- [16] A. Johnson, T. T., Simulation of multiple spheres falling in a liquid-filled tube, *Comp. Meth. Appl. Mech. Eng.* 134 (1996) 351–373.
- [17] A. Johnson, T. T., 3-D simulation of fluid-particle interactions with the number of particles reaching 100, *Comp. Meth. Appl. Mech. Eng.* 145 (1997) 301–321.
- [18] J. Chattot, Y. Wang, Improved treatment of intersecting bodies with the Chimera method and validation with a simple and fast flow solver, *Comput. Fluids* 27 (1998) 721–740.
- [19] H. Nirschl, H. Dwyer, V. Denk, Three-dimensional calculations of the simple shear flow around a single particle between two moving walls, *J. Fluid Mech.* 283 (1995) 273–285.
- [20] C. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (1977) 220.
- [21] C. Peskin, The immersed boundary method, *Acta Numerica* 11 (2002) 479–517.
- [22] Y. Pan, S. Banerjee, Numerical investigation of the effect of large particles on wall turbulence, *Phys. Fluids* 9 (1997) 3786–3807.
- [23] S. Takiguchi, T. Kajishima, Y. Miyake, Numerical scheme to resolve the interaction between solid particles and fluid turbulence, *JSME Int. J. B* 42 (1999) 411–418.
- [24] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* 156 (1999) 209–240.
- [25] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, *J. Comput. Phys.* 174 (2001) 345–380.
- [26] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* 161 (2000) 35–60.
- [27] Z. Li, M.-C. Lai, The immersed interface method for Navier–Stokes equations with singular forces, *J. Comput. Phys.* 171 (2001) 822–842.
- [28] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, *J. Comput. Phys.* 171 (2001) 132–150.
- [29] A. Gilmanov, F. Sotiropoulos, E. Balaras, A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids, *J. Comput. Phys.* 191 (2003) 660–669.
- [30] Y.-H. Tseng, J. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2003) 593–623.
- [31] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, *Comput. Fluids* 33 (2004) 375–404.

- [32] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, *J. Comput. Phys.*, in press.
- [33] S. Chen, G. Doolen, Lattice Boltzmann method in fluid flows, *Annu. Rev. Fluid Mech.* 30 (1998) 329–364.
- [34] C. Aidun, Y. Lu, E.-J. Ding, Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation, *J. Fluid Mech.* 37 (1998) 287–311.
- [35] T. Inamuro, K. Maeba, F. Ogino, Flow between parallel walls containing the lines of neutrally buoyant circular cylinders, *Int. J. Multiphase Flow* 26 (2000) 1981–2004.
- [36] R.J. Hill, D.L. Koch, The transition from steady to weakly turbulent flow in a close-packed ordered array of spheres, *J. Fluid Mech.* 465 (2002) 59–97.
- [37] Z.-G. Feng, E.E. Michaelides, The immersed boundary-lattice Boltzmann method for solving fluid particles interaction problems, *J. Comput. Phys.* 195 (2004) 602–628.
- [38] Z.-G. Feng, E.E. Michaelides, *Proteus*: A direct forcing method in the simulations of particulate flows, *J. Comput. Phys.* 202 (2005) 20–51.
- [39] H. Huang, S. Takagi, PHYSALIS: A new method for particle flow simulation. Part III: Convergence analysis of two-dimensional flows, *J. Comput. Phys.* 189 (2003) 493–511.
- [40] W. Kalthoff, S. Schwarzer, H.J. Herrmann, Algorithm for the simulation of particle suspensions with inertia effects, *Phys. Rev. E* 56 (1997) 2234–2242.
- [41] N. Martys, D. Bentz, E. Garboczi, Computer simulation study of the effective viscosity in Brinkman’s equation, *Phys. Fluids* 6 (1994) 1434–1439.
- [42] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, *J. Comput. Phys.* 123 (1996) 450–465.
- [43] J. Keller, D. Givoli, Exact nonreflecting boundary conditions, *J. Comput. Phys.* 82 (1989) 172–192.
- [44] D. Givoli, *Numerical methods for problems in infinite domains*, Elsevier, Amsterdam, 1992.
- [45] D. Givoli, L. Rivkin, J. Keller, A finite-element method for domains with corners, *Int. J. Numer. Meth. Eng.* 35 (1992) 1329–1345.
- [46] D. Givoli, I. Patlashenko, J. Keller, High-order boundary conditions and finite elements for infinite domains, *Comput. Method Appl. M* 143 (1997) 13–39.
- [47] P. Bjorstad, O. Widlund, Iterative methods for the solution of elliptic problems on regions partitioned into substructures, *SIAM J. Numer. Anal.* 23 (1986) 1097–1120.
- [48] F. Nataf, F. Nier, Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains, *Numer. Math.* 75 (1997) 357–377.
- [49] A. Quarteroni, A. Valli, *Somain decomposition methods for partial differential equations*, Oxford Science Publishers, Oxford, 1999.
- [50] W. Proskurowski, O. Widlund, Numerical solution of Helmholtz’s equation by capacitance matrix method, *Math. Comput.* 135 (1976) 433–468.
- [51] W. Proskurowski, Numerical solution of eigenvalue problem of Laplace operator by a capacitance matrix method, *Computing* 20 (1978) 139–151.
- [52] C. Borgers, Domain embedding methods for the Stokes equations, *Numer. Math.* 57 (1990) 435–451.
- [53] H. Lamb, *Hydrodynamics*, sixth ed., Cambridge U.P., Cambridge, 1932.
- [54] S. Kim, S. Karrila, *Microhydrodynamics*, Butterworth-Heinemann, Boston, 1991.
- [55] D. Brown, R. Cortez, M. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 168 (2001) 464–499.
- [56] A. Prosperetti, H. Oğuz, PHYSALIS: A new $o(N)$ method for the numerical simulation of disperse flows of spheres. Part I: Potential flow, *J. Comput. Phys.* 167 (2001) 196–216.
- [57] G. Mo, A. Sangani, A method for computing Stokes flow interactions among spherical objects and its application to suspensions of drops and porous particles, *Phys. Fluids* 6 (1994) 1637–1652.
- [58] R. Kress, *Linear integral equations*, Springer, Berlin, 1989.
- [59] W. Press, W. Vetterling, S. Teukolsky, B. Flannery, *Numerical recipes in FORTRAN*, second ed., Cambridge U.P., Cambridge, 1992.
- [60] P. Bagchi, S. Balachandar, Direct numerical simulation of flow and heat transfer from a sphere in a uniform cross-flow, *J. Fluids Eng.* 123 (2001) 347–358.
- [61] P. Bagchi, Particle dynamics in inhomogeneous flows at moderate-to-high Reynolds number, Ph.D. Thesis, University of Illinois, Urbana, IL, 2002.
- [62] J.R. Mansfield, O.M. Knio, C. Meneveau, A dynamic LES scheme for the vorticity transport equation: Formulation and a priori tests, *J. Comput. Phys.* 145 (1998) 693–730.
- [63] M.I. M., D. Lohse, F. Toschi, On the relevance of the lift force in bubbly turbulence, *J. Fluid Mech.* 488 (2003) 283–313.
- [64] U. Balucani, M. Zoppi, *Dynamics of the liquid state*, Clarendon Press, Oxford, 1994.
- [65] P. Chaikin, T. Lubensky, *Principles of condensed matter physics*, Cambridge U.P., Cambridge, UK, 1995.

- [66] J. Percus, G. Yevick, Analysis of classical statistical mechanics by means of collective coordinates, *Phys. Rev.* 110 (1958) 1–13.
- [67] M.S. Wertheim, Exact solution of the Percus–Yevick integral equation for hard spheres, *Phys. Rev. Lett.* 10 (1963) 321–323.
- [68] E. Thiele, Equation of state for hard spheres, *J. Chem. Phys.* 39 (1963) 474–479.
- [69] C. Pozrikidis, *Boundary integral and singularity methods for linearized viscous flow*, Cambridge U.P., Cambridge, 1992.
- [70] Z. Zhang, A. Prosperetti, Sedimentation of 1024 particles, in: *Proceedings of the fluids engineering division summer meeting*, ASME, New York, 2005.
- [71] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flow, *J. Comput. Phys.* (2005), in press.