Improved procedure for the computation of Lamb's coefficients in the PHYSALIS method for particle simulation

K. Gudmundsson [a], A. Prosperetti [a,b,*]

[a] Faculty of Science and Technology and J.M. Burgers Centre for Fluid Dynamics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
[b] Department of Mechanical Engineering, Johns Hopkins University Baltimore, MD 21218, USA

ABSTRACT

The PHYSALIS method was designed for the simulation of flows with suspended spherical particles. It differs from standard immersed boundary methods due to the use of a local spectral representation of the solution in the neighborhood of each particle, which is used to bridge the gap between the particle surface and the underlying fixed Cartesian grid. This analytic solution involves coefficients which are determined by matching with the finite-difference solution farther away from the particle. In the original implementation of the method this step was executed by solving an over-determined linear system via the singular-value decomposition. Here a more efficient method to achieve the same end is described. The basic idea is to use scalar products of the finite-difference solution with spherical harmonic functions taken over a spherical surface concentric with the particle. The new approach is tested on a number of examples and is found to posses a comparable accuracy to the original one, but to be significantly faster and to require less memory. A novel test case that we describe demonstrates the accuracy with which the method conserves the fluid angular momentum in the case of a rotating particle.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Sediment transport, settling suspensions, fluidized beds and porous media are but a few examples of fluid flows with fixed or moving particles. As these examples show, such flows are frequently encountered and of great importance in many problems of scientific and technological relevance. From a computational perspective, they pose a major challenge as the particles constitute a very complex and frequently time-dependent internal boundary which needs to be adequately resolved for a *bona fide* simulation.

Until very recently the only practical approach was the Eulerian–Lagrangian point-particle model, in which the particles are assimilated to mathematical points contributing extra forces to the fluid, while their motion is found by integrating in time a dynamic equation in which the fluid forces are modeled rather than deduced from first principles. The last decade has seen the development of methods capable of dealing with the finite extent of the particles. The most successful such method so far is the one developed by Uhlmann [1,2], which has been also used by Lucci et al. [3].

An alternative approach, dubbed PHYSALIS, first described in [4,5] and, more fully, in [6], differs from standard immersed boundary methods in its reliance on an analytic spectral representation of the solution in the neighborhood of each particle which is coupled to a finite-difference numerical solution farther away; a brief description is provided in Section 2. The local analytic solution is used as a bridge between the particle boundary and the finite-difference solution to bypass the difficulty posed by the mis-match between the particle boundary and the underlying uniform Cartesian grid.

---

* Corresponding author at: Department of Mechanical Engineering, Johns Hopkins University Baltimore, MD 21218, USA.
  E-mail address: prosperetti@jhu.edu (A. Prosperetti).

The analytic solution contains undetermined coefficients which are found iteratively by matching the finite-difference solution with the local one. The significant improvement described in this paper concerns the calculation of these coefficients, which was carried out using the singular-value decomposition in the original implementation. We show how this step can be executed by taking suitable scalar products with a significant saving in execution time. The purpose of this paper is to describe the implementation of this new method and its verification.

Advantages of PHYSALIS are the exact satisfaction of the no-slip condition at the particle surface, the great simplification of the calculation of the hydrodynamic forces and couples, and the avoidance of the complex issues arising from the lack of geometrical conformity between the curved particle boundary and the underlying fixed Cartesian grid. Furthermore, the use of a local spectral representation of the solution permits one to describe the effect of each particle with fewer degrees of freedom than conventional finite-difference-based methods. As a consequence, the grid resolution can be kept relatively low without compromising the accuracy of the solution.

The method has been thoroughly validated in the original papers and it has proven its value in a number of applications: the sedimentation of 1024 particles [7], the turbulent flow around fixed particles [8,9], the flow induced by a rotating particle [10] and over a porous medium [11,12]. An interesting variant which uses artificial compressibility to generate a time-stepping pressure equation has been described and used by Perrin and Hu [13,14].

2. General framework

We briefly review here the PHYSALIS method for particle simulation, as originally introduced for cylindrical particles [4,5], and later extended to spherical particles [6].

The objective is to simulate, on a fixed Cartesian grid, incompressible fluid flow around spherical particles, while obeying the no-slip and no-penetration velocity boundary-conditions on the particle surface. To this end it is necessary to solve the incompressible Navier–Stokes equations,

$$\rho\left[\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U}\right] = -\nabla p + \mu \nabla^2 \mathbf{U} + \rho \mathbf{g}, \tag{1}$$

$$\nabla \cdot \mathbf{U} = 0, \tag{2}$$

where $\mathbf{U}$, ρ, p, and $\mathbf{g}$ respectively denote the velocity, density, pressure, and gravity fields.

As with other immersed-boundary methods, since the particles do not conform to the computational grid, the boundary conditions imposed by the particles must be transferred to the adjacent nodes. The PHYSALIS method enables this transferral to be done in a spectrally accurate manner.

To proceed, we transform the above equations to a frame in which the particle is at rest. (In the case of more than one particle this step is carried out for each particle.) The fluid velocity $\mathbf{u}$ in the particle rest-frame is related to $\mathbf{U}$, the velocity in the laboratory frame, by

$$\mathbf{U} = \mathbf{u} + \mathbf{w} + \boldsymbol{\Omega} \times \mathbf{x}, \tag{3}$$

where $\mathbf{w}$ and $\boldsymbol{\Omega}$ respectively represent the linear and angular velocities of the particle, and $\mathbf{x}$ is the position with respect to the particle center. The momentum equation in the particle rest frame is given by

$$\rho\left[\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + 2\boldsymbol{\Omega} \times \mathbf{u}\right] = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} - \rho\left[\dot{\mathbf{w}} + \dot{\boldsymbol{\Omega}} \times \mathbf{x} + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})\right], \tag{4}$$

and that of continuity is given by

$$\nabla \cdot \mathbf{u} = 0. \tag{5}$$

Now let

$$\mathbf{u}^{\Omega} = \frac{r^5 - a^5}{10\nu r^3} \dot{\boldsymbol{\Omega}} \times \mathbf{x} \quad \text{and} \quad p^{\Omega} = \frac{1}{2}\rho(\boldsymbol{\Omega} \times \mathbf{x})^2 - \rho(\dot{\mathbf{w}} - \mathbf{g}) \cdot \mathbf{x}, \tag{6}$$

where $r = |\mathbf{x}|$, and a represents the particle radius. Then, if we introduce the modified velocity and pressure fields

$$\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}^{\Omega} \quad \text{and} \quad \tilde{p} = p - p^{\Omega}, \tag{7}$$

we can write the rest-frame momentum and continuity equations respectively as

$$\rho\left[\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + 2\boldsymbol{\Omega} \times \mathbf{u}\right] = -\nabla \tilde{p} + \mu \nabla^2 \tilde{\mathbf{u}}, \quad \text{and} \tag{8}$$

$$\nabla \cdot \tilde{\mathbf{u}} = 0. \tag{9}$$

The terms within the square brackets still contain the original rest-frame velocity field and therefore they vanish at the particle surface. As a consequence, they will be small in its immediate vicinity. Thus, in this region, $\tilde{p}$ and $\tilde{\mathbf{u}}$ will approximately satisfy the Stokes equations, given by

$$-\nabla \tilde{p} + \mu \nabla^2 \tilde{\mathbf{u}} = 0 \quad \text{and} \quad \nabla \cdot \tilde{\mathbf{u}} = 0. \tag{10}$$

It should be stressed that the method is intended for application to finite-Reynolds-number flows. The procedure may be seen as a linearization of the Navier–Stokes equations about the solid-body motion which – at any Reynolds number – the fluid in contact with the particle surface executes by virtue of the no-slip condition.

Lamb [15] (see also Kim and Karrila [16]) developed an exact general solution of (10) in the case of a spherical boundary. In this solution the velocity field is represented as an expansion in terms of vector spherical harmonics (allied to the standard scalar spherical harmonics), which form a complete basis for square-integrable vector functions on the sphere [17]:

$$\tilde{\mathbf{u}} = \frac{v}{a^2} \sum_{n=-\infty}^{\infty} \frac{1}{(n+1)(2n+3)} \left[\frac{1}{2}(n+3)r^2 \nabla p_n - n\mathbf{x}p_n \right] + \frac{v}{a} \sum_{n=-\infty}^{\infty} \left[a\nabla\phi_n + \nabla \times (\mathbf{x}\chi_n) \right]. \tag{11}$$

The pressure $\tilde{p}$ is represented similarly as an expansion over scalar spherical harmonics:

$$\tilde{p} = \frac{\mu v}{a^2} \sum_{n=-\infty}^{\infty} p_n. \tag{12}$$

In these expressions the p_n, ϕ_n, and χ_n are solid harmonics of order n:

$$\begin{bmatrix} p_n \\ \phi_n \\ \chi_n \end{bmatrix} = \left(\frac{r}{a}\right)^n \sum_{m=-n}^{n} \begin{bmatrix} p_{nm} \\ \phi_{nm} \\ \chi_{nm} \end{bmatrix} Y_n^m(\theta, \varphi). \tag{13}$$

Here p_{nm}, ϕ_{nm}, χ_{nm} are undetermined dimensionless coefficients and the functions Y_n^m are normalized spherical harmonics shown in detail in (46) of the Appendix. The regular harmonics, corresponding to non-negative n, represent a general incident flow, while the singular harmonics (negative n) represent the disturbance field due to the particle. Although the coefficients $\{p_{nm}, \phi_{nm}, \chi_{nm}\}$ are in general complex, the relations $\{\overline{p}_{nm}, \overline{\phi}_{nm}, \overline{\chi}_{nm}\} = (-1)^m \{p_{n,-m}, \phi_{n,-m}, \chi_{n,-m}\}$ (in which the overline denotes the complex conjugate) obviate the need for dealing with complex numbers and the entire calculation can be carried out in the real domain as explained later.

The central observation at the root of the PHYSALIS method lies in the recognition that, in the immediate neighborhood of the particle, even at finite Reynolds numbers, the modified velocity and pressure fields $\tilde{p}$ and $\tilde{\mathbf{u}}$ must be describable as in (11) and (12) due to the smallness of inertia. Thus, provided the coefficients are known, (11) and (12) can be used to "transfer" the boundary conditions from the particle surface to a *cage* of adjacent grid nodes enclosing the particle. After this step, the computation is carried out only on the nodes of the fixed Cartesian grid and the complexity deriving from the geometric mismatch between the particle boundary and the grid is eliminated.

With this background, we can now describe the algorithm:

1. By matching a provisional finite-difference solution (e.g., that at the previous time step) to the explicit representations (11) and (12) and related ones (see below) at the cage nodes, find provisional values of the expansion coefficients;
2. Use these provisional values to generate new velocity boundary conditions at the cage nodes;
3. Solve again the Navier–Stokes equations on the finite-difference grid using these boundary conditions;
4. Repeat to convergence.

It should be noted that the coefficients appearing in (13) embody the physics of the interaction of the particle with the fluid and contain, therefore, important information. For example, the hydrodynamic force, $\mathbf{F}$, and couple, $\mathbf{L}$, acting on the particle can be expressed in terms of the $n = 1$ coefficients as [6]:

$$\mathbf{F} = \rho v(\dot{\mathbf{w}} - \mathbf{g}) + 6\pi\mu a\mathbf{\Phi} + \pi\mu a^3 \mathbf{P} \tag{14}$$

$$\mathbf{L} = 8v\mu\mathbf{X} + \rho a^2 v\dot{\mathbf{\Omega}}, \tag{15}$$

where $v = \frac{4}{3}\pi a^3$ is the volume of the particle, and we have defined $\mathbf{\Phi} = (\phi_{11}^r, \phi_{11}^i, \phi_{10}^r)$, $\mathbf{P} = (p_{11}^r, p_{11}^i, p_{10}^r)$, and $\mathbf{X} = (\chi_{11}^r, \chi_{11}^i, \chi_{10}^r)$, with superscripts r and i denoting the real and imaginary parts. Thus, unlike other immersed boundary methods, once the coefficients are known the force and couple are known as well with no need for additional calculations.

As discussed in the following section, the specific contribution of the present paper concerns the first step, that of deducing from a given velocity and pressure field the values of the coefficients.

3. Computing the expansion coefficients

Away from the particle(s), the Navier–Stokes Eq. (1) are solved on a staggered grid by a standard finite-difference method as described in [6]. This reference also explains in detail the manner in which the *cage* of Cartesian nodes enclosing the particle is constructed. Let $\mathbf{x}_K$, with $K = 1, 2, \ldots, N$ be the cage nodes. In the original algorithm, the coefficients are calculated by requiring that the pressure field given by (12) and the vorticity field $\tilde{\omega}$ obtained from (11) reproduce the finite-difference results. In other words, one forms a linear system involving the coefficients as unknowns, e.g.

$$\tilde{p} = \frac{\mu v}{a^2} \sum_{n=-\infty}^{\infty} \left(\frac{r_K}{a}\right)^n \sum_{m=-n}^{n} p_{nm} Y_n^m(\theta_K, \varphi_K) = p(\mathbf{x}_K) - p^{\Omega}(\mathbf{x}_K), \tag{16}$$

in which $\mathbf{x}_K = (r_K, \theta_K, \phi_K)$, the left-hand side is as given in (12) and contains the unknown coefficients, while the first term in the right-hand side is the finite-difference result at the cage nodes. The vorticity field is treated similarly. As explained before, the coefficients thus obtained are then substituted into (11) to generate velocity boundary conditions at the cage nodes.

In this way, four scalar equations are obtained for each cell comprising the cage. Taken together over the cage, this set of equations forms an overdetermined linear system for the unknown coefficients which is solved by means of the singular value decomposition (SVD) algorithm, i.e., in essence, in a least-squares manner. Although the method is stable, the typically large systems that result (e.g., a matrix of 2400×46 for a grid-resolution of 8 cells per radius, and with the analytical expressions truncated at $N_c = 3$) prompted us to search for a more efficient procedure.

An alternate and, perhaps, more natural way to obtain the coefficients relies on the orthogonality property of the spherical harmonics, for example

$$(Y_l^k, Y_n^m) \equiv \int_0^{\pi} \sin\theta d\theta \int_0^{2\pi} \overline{Y}_l^k Y_n^m d\varphi = \delta_{ln}\delta_{km}. \tag{17}$$

In the spirit of this approach, (16) is replaced by

$$(Y_n^m, \tilde{p}) = (Y_n^m, p - p^{\Omega}). \tag{18}$$

or, upon substituting (12) for $\tilde{p}$, using (17) and recalling that $Y_{-n-1}^m = Y_n^m$,

$$(Y_n^m, \tilde{p}) = \frac{\mu v}{a^2} (p_{nm} + p_{-n-1,m}) = (Y_n^m, p - p^{\Omega}). \tag{19}$$

It is shown in the Appendix that the coefficients of negative order can be expressed in terms of those of positive order to find

$$\frac{a^2}{\mu v}(Y_n^m, \tilde{p}) = \left[1 - \frac{n}{2}\frac{2n-1}{n+1}\left(\frac{a}{r}\right)^{2n+1}\right]\left(\frac{r}{a}\right)^n p_{nm} - \frac{n(2n-1)(2n+1)}{n+1}\left(\frac{a}{r}\right)^{n+1}\phi_{nm}. \tag{20}$$

This property of the spherical harmonics to single out individual coefficients is analogous to the Fourier transform and, as discussed in Section 4, there exists a "Fast Spherical Transform" (FST) [18], incorporating and scaling as the Fast Fourier Transform. As will be shown below, this procedure proves considerably faster than the SVD approach of [6], and comparably accurate.

The other coefficients can be determined by a similar procedure. There are different possible choices of field variables to use for this purpose but, whichever one may consider, ϕ_{nm} and p_{nm} appear coupled, while χ_{nm} appears separately.

In the derivations we will use two orthogonality properties of the vector spherical harmonics similar to (17), namely (see e.g. Ref. [17])

$$(r\nabla Y_l^k, r\nabla Y_n^m) = r^2 \int_0^{\pi} \sin\theta d\theta \int_0^{2\pi} \nabla\overline{Y}_l^k \cdot \nabla Y_n^m d\varphi = n(n+1)\delta_{ln}\delta_{km}, \tag{21}$$

$$(\mathbf{x} \times \nabla Y_l^k, \mathbf{x} \times \nabla Y_n^m) = \int_0^{\pi} \sin\theta d\theta \int_0^{2\pi} (\mathbf{x} \times \nabla\overline{Y}_l^k) \cdot (\mathbf{x} \times \nabla Y_n^m) d\varphi = n(n+1)\delta_{ln}\delta_{km}. \tag{22}$$

3.1. Computing ϕ_{nm} and p_{nm}

In general it appears desirable to choose pressure as one of the field variables used to calculate the coefficients p_{nm} and ϕ_{nm}, since this field is smoother than the velocity field and, additionally, does not vanish near the particle. Thus, (20) is one of the equations that we use to determine the coefficients.

There are three other scalar products which can be combined with the pressure to obtain a pair of equations suitable for the calculation of the coefficients p_{nm} and ϕ_{nm}: the radial and transversal components of velocity, $\tilde{u}_r = \mathbf{e}_r \cdot \tilde{\mathbf{u}}$ and $\tilde{\mathbf{u}}_{\parallel} = \tilde{\mathbf{u}} - \tilde{u}_r \mathbf{e}_r$, and the transversal component of vorticity $\tilde{\boldsymbol{\omega}}_{\parallel} = \tilde{\boldsymbol{\omega}} - \tilde{\omega}_r \mathbf{e}_r$. We will present each of these options below and, in Section 5, demonstrate their respective performance.

1. *Radial component of velocity*: we take the scalar product of Y_n^m and $\tilde{u}_r$, as found from (11), obtaining

$$\frac{a}{v}(Y_n^m, \tilde{u}_r) = \frac{n}{4}\left[\frac{2}{2n+3} + \left(\frac{2n+1}{2n+3}\frac{a^2}{r^2} - 1\right)\left(\frac{a}{r}\right)^{2n+1}\right]\left(\frac{r}{a}\right)^{n+1} p_{nm} + n\left[1 + \frac{1}{2}\left[2n-1 - (2n+1)\frac{r^2}{a^2}\right]\left(\frac{a}{r}\right)^{2n+1}\right]\left(\frac{r}{a}\right)^{n-1}\phi_{nm}, \tag{23}$$

where we have used (17) and the relations (49) and (50) of the Appendix to eliminate the coefficients with negative index. It may be shown that, near the particle, the terms in brackets in this expression are of the order of $[(r-a)/a]^2$.

2. *Transversal components of velocity*: By using (21) in (11) we find

$$\frac{a}{\nu}(r\nabla Y_n^m, \tilde{\mathbf{u}}_\parallel) = \frac{n}{4}\left\{\frac{2(n+3)}{2n+3} + \left[(n-2) - \frac{n(2n+1)}{2n+3}\frac{a^2}{r^2}\right]\left(\frac{a}{r}\right)^{2n+1}\right\}\left(\frac{r}{a}\right)^{n+1}p_{nm}$$
$$+ n\left[n+1+\frac{1}{2}\left((n-2)(2n+1)\frac{r^2}{a^2} - n(2n-1)\right)\left(\frac{a}{r}\right)^{2n+1}\right]\left(\frac{r}{a}\right)^{n-1}\phi_{nm}, \tag{24}$$

where again we have used the relations (49) and (50) to eliminate the coefficients with negative index. Near the particle, this expression is proportional to $(r-a)/a$.

3. *Transversal components of vorticity*: Upon taking the curl of the velocity given by (11) and projecting on the transversal direction we find

$$\frac{a^2}{\nu}\tilde{\boldsymbol{\omega}}_\parallel = \mathbf{e}_r \times \sum_{n=1}^\infty \frac{1}{n+1}\left\{\left[1+\frac{2n-1}{2}\left(\frac{a}{r}\right)^{2n+1}\right]\nabla' p_n + (2n-1)(2n+1)\left(\frac{a}{r}\right)^{2n+1}\nabla'\phi_n\right\}$$
$$+ \frac{a}{r}\sum_{n=1}^\infty\left[n+1+n\left(\frac{a}{r}\right)^{2n+1}\right]\nabla'\chi_n, \tag{25}$$

where for brevity we have defined the operator ∇' as, e.g.,

$$\nabla' p_n = \left(\frac{r}{a}\right)^n r \sum_{m=-n}^n p_{nm}\nabla Y_n^m(\theta,\varphi). \tag{26}$$

Upon taking the scalar product of this equation with $\mathbf{x} \times \nabla Y_l^k$ and using (22) we obtain

$$\frac{a^2}{\nu}(\mathbf{x}\times\nabla Y_n^m, \tilde{\boldsymbol{\omega}}_\parallel) = n\left[1+\frac{2n-1}{2}\left(\frac{a}{r}\right)^{2n+1}\right]\left(\frac{a}{r}\right)^n p_{nm} + n(2n-1)(2n+1)\left(\frac{a}{r}\right)^{n+1}\phi_{nm}, \tag{27}$$

where again (49) and (50) have been used. This result is clearly finite at the particle surface.

Having in hand Eq. (20), along with either (23), (24), or (27), we can now solve for the two unknowns, ϕ_{nm} and p_{nm}. The separation of the real and imaginary parts of these equations is immediate (see (35) and (36) below) and avoids the use of complex computation.

3.2. Computing χ_{nm}

Three scalar products can be formed that involve χ_{nm}:

1. *Transversal components of velocity*: projecting $\tilde{\mathbf{u}}_\parallel$ upon $\mathbf{x} \times \nabla Y_n^m$, rather than $r\nabla Y_n^m$, as done in Eq. (24) we obtain

$$\frac{a}{\nu}(\mathbf{x}\times\nabla Y_n^m, \tilde{\mathbf{u}}_\parallel) = -n(n+1)\left[1-\left(\frac{a}{r}\right)^{2n+1}\right]\left(\frac{r}{a}\right)^n\chi_{nm}, \tag{28}$$

where we have used Eq. (51) of the Appendix to eliminate $\chi_{-n-1,m}$.

2. *Radial component of vorticity*: after using Eq. (51), $\tilde{\omega}_r$ takes the form

$$\frac{a^2}{\nu}\tilde{\omega}_r = \frac{a}{r}\sum_{n=1}^\infty n(n+1)\left[1-\left(\frac{a}{r}\right)^{2n+1}\right]\chi_n. \tag{29}$$

Projecting this expression upon Y_n^m, we obtain

$$\frac{a^2}{\nu}(Y_n^m, \tilde{\omega}_r) = n(n+1)\left[1-\left(\frac{a}{r}\right)^{2n+1}\right]\left(\frac{r}{a}\right)^{n-1}\chi_{nm}, \tag{30}$$

3. *Transversal components of vorticity*: projecting $\tilde{\boldsymbol{\omega}}_\parallel$ upon $r\nabla Y_n^m$, rather than $\mathbf{x} \times \nabla Y_n^m$ as was the case in Eq. (27), we obtain

$$\frac{a^2}{\nu}(r\nabla Y_n^m, \tilde{\boldsymbol{\omega}}_\parallel) = n(n+1)\left[n+1+n\left(\frac{a}{r}\right)^{2n+1}\right]\left(\frac{r}{a}\right)^{n-1}\chi_{nm}. \tag{31}$$

Any one of relations (28), (30), and (31), may be used to obtain χ_{nm}.

4. Implementation

The second-order accurate discretization of the Navier–Stokes equations, along with the construction of the cage and the manner in which the particle boundary conditions are applied at the cage surface, are described in detail in [6]. We will therefore focus our discussion here on those aspects of the present methodology that diverge from the original version, namely, the calculation of the expansion coefficients.

The scalar products appearing in (18) and the following equations amount to integrations carried out over a spherical surface concentric with and enclosing the particle. In general, therefore, the quadrature nodes used for the evaluation of the integrals implied by the scalar products do not coincide with the finite-difference grid points. We thus require a procedure for approximating the values of ω, $\mathbf{u}$, and p at the quadrature nodes on the basis of those on the finite-difference grid. To this end we consider two separate approaches: interpolation, and local linearization. For the reasons explained below, we limit the latter to those scalar products involving $\tilde{p}$ and $\tilde{\mathbf{u}}_\parallel$ in relations (20), (24), and (28).

4.1. Interpolation

Since the spatial discretization used in PHYSALIS is second-order accurate, an interpolation of the same order or higher is appropriate. We experimented with cubic splines and quadratic Lagrange-polynomials, both as implemented in the GNU Scientific Library [19]. These proved comparably accurate, but the Lagrange interpolation is considerably more expedient and was hence adopted.

Given the point $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, \hat{z})$ at which the value of e.g. the pressure p is required, the interpolation proceeds as follows:

1. Let $\mathbf{x}_{i,j,k}$ be the cell containing $\hat{\mathbf{x}}$;
2. Centered on, and including this cell, construct a cube of 27 cells having coordinates $\mathbf{x}_{i+\alpha,j+\beta,k+\gamma}$, with $\alpha, \beta, \gamma = 0, \pm 1$, at which p is known;
3. Arbitrarily choosing to interpolate first on the y-z planes of the cube, we interpolate $p(x_i, y_j, z_k)$ in z for each y_j and x_i, and evaluate the results at $z = \hat{z}$, yielding $p(x_i, y_j, \hat{z})$;
4. Interpolate $p(x_i, y_j, \hat{z})$ in y for each x_j, and evaluate at $y = \hat{y}$, yielding $p(x_i, \hat{y}, \hat{z})$;
5. A final interpolation in x is performed and evaluated at $x = \hat{x}$, resulting in the desired approximation of $p(\hat{\mathbf{x}})$.

This three-dimensional, second-order accurate interpolation therefore requires 13 one-dimensional interpolations in total. Higher-order interpolations are clearly possible but the necessary effort and computational overhead do not seem warranted without a corresponding increase of the order of accuracy of the finite-difference discretization of the Navier–Stokes equations.

4.2. Local linearization

A considerably cheaper alternative is a linear interpolation of the fields. For pressure, we write

$$p(\hat{\mathbf{x}}) \approx p(\mathbf{x}_c) + \nabla p|_{\mathbf{x}_c} \cdot (\hat{\mathbf{x}} - \mathbf{x}_c), \tag{32}$$

where $\hat{\mathbf{x}}$ is defined as above, and $\mathbf{x}_c = \mathbf{x}_{i,j,k}$ is the cell-center which, given the staggered-grid formulation, is where p resides. The gradient ∇p is approximated with second-order accurate (at $\mathbf{x}_c$) central differences, using adjoining cells.

For velocity, the corresponding expression is

$$\mathbf{u}(\hat{\mathbf{x}}) \approx \mathbf{u}(\mathbf{x}_c) + \mathsf{J}(\mathbf{x}_c)(\hat{\mathbf{x}} - \mathbf{x}_c) \quad \text{where} \quad \mathsf{J}_{ij}(\mathbf{x}_c) = \left.\frac{\partial u_i}{\partial x_j}\right|_{\mathbf{x}_c}. \tag{33}$$

Since the velocity is defined at face centers, we approximate $\mathbf{u}(\mathbf{x}_c)$, to second-order accuracy, with the average across the cell. The gradients J_{ij} are calculated with central differences. The diagonal components $\mathsf{J}_{ii}(\mathbf{x}_c)$ reside at the cell-center. As the remaining components reside at edge-centers, we approximate them to second-order accuracy with the average across the cell.

The accuracy of linear interpolation is one order lower than that of the spatial discretization of the Navier–Stokes equations. This deficiency, however, is balanced by several significant advantages that make it very much worthy of exploration. Firstly, note that the estimate $\mathbf{u}(\hat{\mathbf{x}})$ is divergence-free to the same accuracy as the velocity field at the nodes, since

$$\nabla \cdot \mathbf{u}(\hat{\mathbf{x}}) = \mathsf{J}_{ii}(\mathbf{x}_c) = \left.\frac{\partial u_i}{\partial x_i}\right|_{\mathbf{x}_c}, \tag{34}$$

while the same is not necessarily true of the interpolation. Secondly, the operation count involved with the linearization is lower than for the interpolation. Lastly, the spatial extent of the region involved with the computation is reduced, as only the six adjacent, face-sharing cells need be considered. Given that the scalar product approach rests on the local validity of the Stokes equations, the ability to use only data very close to the particle is a very desirable feature of the local linearization, which might ultimately result in a better overall accuracy in spite of its low formal order.

The same approach applied to the vorticity would only give a zero-order accuracy. An effort to improve this low accuracy, on the other hand, would require a comparatively much larger set of cells. This is the reason why we did not pursue this method for the vorticity options.

With these procedures for sampling the flow field at the quadrature points on the sphere, the next task is to evaluate the scalar products. The present integrands (e.g., in Eq. (17)) are oscillatory and increasingly so with higher n. Particular care must therefore be exercised in distributing the quadrature points in order to maintain stability. Given the homogeneity of the azimuthal (ϕ) direction, a uniform grid $\phi_k = 2\pi k/2N$, is appropriate, where $k = 0, \ldots, 2N - 1$ and $2N$ is the number of

quadrature points in ϕ. The natural distribution in the polar (θ) direction, spanned by the associated Legendre polynomials (cf. Eqs. (46) and (47)), is $\theta_j = \pi(2j + 1)/4N$, where j has the same range as k.

The scalar product on the sphere can essentially be considered as the product of discrete Fourier and Legendre transforms in the azimuthal and polar directions, respectively. Using the quadrature points described above, the cost of a direct numerical evaluation of these transforms scales as $O(N^4)$. Faster alternatives exist, however. The Fast Fourier Transform can naturally be used in the azimuthal direction, and there additionally exists a Fast Legendre Transform, due to [18]. The combination of these fast algorithms reduces the cost scaling to $O(N^2 \log(N))$. The authors of [18] have released a C-implementation of their method, the *SpharmonicKit*,[1] which we embed in our code.

Lastly, we note that the Fast Spherical Transform of [18] includes only the spherical harmonics Y_n^m, and not gradients thereof. To compute the scalar products involving ∇Y_n^m (e.g., Eq. (28)), we express these in terms of the Y_n^m:

$$\frac{\partial Y_n^m}{\partial \phi} = im Y_n^m, \tag{35}$$

$$\frac{\partial Y_n^m}{\partial \theta} = \frac{m}{\tan(\theta)} Y_n^m + \frac{N_{n,m}}{N_{n,m+1}} e^{-i\phi} Y_n^{m+1}; \tag{36}$$

the normalization constants $N_{n,m}$ are given in (46) of the Appendix. It should be noted that, in spite of the explicit appearance of the imaginary unit, the actual computations are carried out in the real domain only.

5. Results

In this section we contrast the various aspects of the singular-value decomposition (SVD) and the scalar product (SP) version of PHYSALIS. In particular we examine relative speed and accuracy, and address the question of which SP is best suited.

5.1. Comparing the speed of the SVD and SP

As mentioned in Section 3, [6] compute the expansion coefficients by matching the vorticity and pressure, based respectively on the analytical expansions in Eqs. (11) and (12), to those from the finite-difference field, resulting in a linear system, $A\mathbf{c} = \mathbf{b}$. A is a rectangular matrix of size $L \times M$, where $L = N_\omega + N_p$ and $M = 3N_c(N_c + 2) + 1$, where N_c is the degree at which the summations in Eqs. (11) and (12) are truncated; $\mathbf{c}$ is the vector containing the unknown expansion coefficients; and $\mathbf{b}$ contains the values from the finite-difference field. N_ω and N_p respectively represent the number of vorticity and pressure nodes comprising the cage.

The commonly used SVD algorithm can be shown [20] to scale asymptotically as $O(LM^2 + M^3)$. In contrast, the cost of the FST scales as $O(N^2 \log(N))$, where $2N \times 2N$ is the number of quadrature points. If the flow variables $\bar{\omega}$, $\bar{\mathbf{u}}$, and $\bar{p}$ are *band-limited*, so that the summation index runs between 0 and B, then $N = B$ will suffice to calculate the coefficients without error (this is analogous to the Nyquist–Shannon sampling theorem [21]), assuming that the samples are error-free [18]. In the general case we do not know B and, therefore, to prevent aliasing, take $N = 3N_c$. In terms of N_c, the cost of the FST therefore scales similarly.

As shown in [6], $N_c = 4$ suffices to adequately describe flows with particle Reynolds numbers up to about 100, with a grid resolution of $a/h = 8$, where h is the grid spacing. As such, the asymptotic scalings above are not as relevant as empirical measurements of the cost associated with a single calculation of the coefficients. Such measurements are shown in Fig. 1, both as functions of N_c (with fixed a/h) and of the grid resolution a/h (with fixed N_c). The SP approach proves over two orders of magnitude faster than the SVD at all N_c shown. We also note that the cost of both the interpolation and the linearization, described in the previous section, is independent of the grid resolution as the number of quadrature points is only determined by N_c. This is apparent from Fig. 1, right, while the cost of the SVD grows strongly with a/h.

While the SP has been shown to be significantly faster than the SVD, there are some caveats of note. Firstly, the most expensive portion of the overall calculation is the solution of the Poisson equation for pressure. The speed-up in the calculation of the coefficients, while impressive, will therefore only affect the overall calculation fractionally, and not proportionally. Notwithstanding, the overall gains can be significant, particularly in the presence of many particles. Secondly, we note that the SP is only faster for moving particles. Since the cage of stationary particles is fixed, the SVD needs only to be computed once, and the resulting decomposition can be re-used, resulting in comparable speed with the SP. If the particle is moving, on the other hand, the cage moves with it (once the particle moves a distance of $h/2$ from the center of the cage, a new cage is constructed [6]), requiring a call to the SVD. Although this does not happen at every time step, the effect can be nonetheless significant as shown in Section 5.5.

5.2. Recovery of assigned coefficients in Stokes flow

The simplest test of the accuracy with which the coefficients are calculated can be carried out on a flow field generated analytically by using the expressions (11) and (12). We use these relations with arbitrarily chosen coefficients to assign the

---

[1] See http://www.cs.dartmouth.edu/~geelong/sphere/.
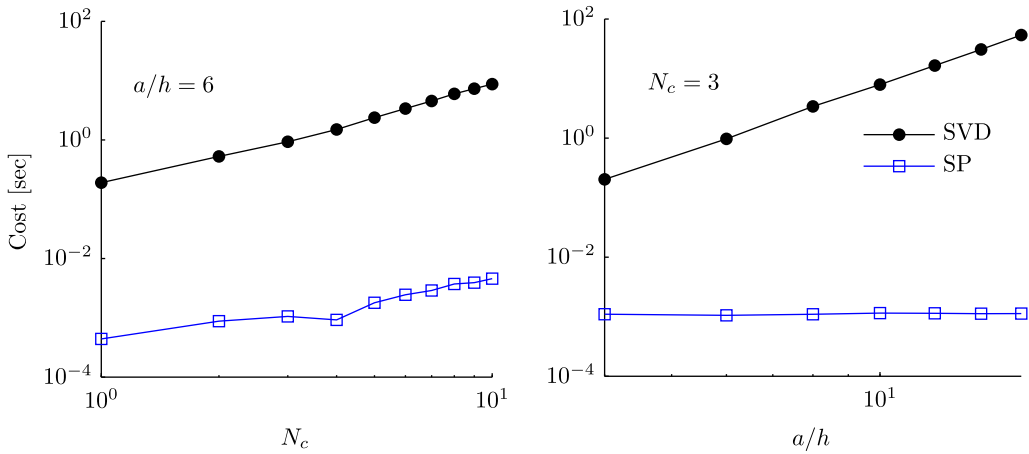
Fig. 1. Cost of coefficient recovery using the original SVD and the present SP methods, as a function of the truncation order N_c (left), and grid resolution a/h (right). The timings were measured empirically using an Intel 2.66 GHz CPU.

fields at the grid points. We choose $N_c = 4$ (for a total of $3N_c(N_c + 2) + 1 = 73$ coefficients assigned), and a spatial resolution of $a/h = 10$. The SPs are computed at $r/a = 1.5$. Given that the entire field in this case is Stokes, the solution is, in theory, insensitive to the choice of r/a. However, the further out one is from the surface of the particle, the finer the Cartesian grid becomes relative to the scale (r/a) on which the Y_n^m vary. Later we will demonstrate how the accuracy varies with r/a for flows in which the Stokes region has a finite extent.

To quantify the accuracy with which the assigned coefficients are recovered from the assigned fields, we define an error metric as

$$E_p(n) = \sqrt{\frac{1}{n} \sum_{m=-n}^{n} |p_{nm}^A - p_{nm}|^2}, \tag{37}$$

where p_{nm}^A and p_{nm} respectively denote the assigned and recovered values; E_ϕ and E_χ are defined in the same way.

Fig. 2 shows the results so obtained. Generally, the recovery by all methods is reasonable, with a maximum error level of $E_p = 6\%$, achieved by the SVD. In fact, all of the SPs prove more accurate than the SVD under the E_p error metric. In contrast the SVD incurs a larger error at higher values of n. This instability, presumably associated with the numerical implementation of the SVD, is particularly apparent in E_p, but can also be noted in E_ϕ and E_χ. In terms of E_p and E_ϕ, the SP involving the interpolated $\tilde{\mathbf{u}}_\parallel$ proves most accurate, followed by those involving $\tilde{u}_r$ and $\tilde{\omega}_\parallel$, and the linearization estimate of $\tilde{\mathbf{u}}_\parallel$ (recall, from Section 4, that linearization is only used with $\tilde{\mathbf{u}}_\parallel$ and $\tilde{p}$). The interpolated $\tilde{\mathbf{u}}_\parallel$ also performs best in terms of E_χ, but now followed by $\tilde{\omega}_\parallel$, $\tilde{\omega}_r$, and the linearized $\tilde{\mathbf{u}}_\parallel$.

Although the reasons for the differing performance, in terms E_p, E_ϕ, and E_χ, are somewhat unclear, we can make some observations. Firstly, we note that the interpolated $\tilde{\mathbf{u}}_\parallel$ is likely more accurate than $\tilde{u}_r$ due to its vector nature: the relevant SPs (Eqs. (24) and (28)) involve two velocity components, presumably resulting in some cancellation of error. While the same is true of $\tilde{\omega}_\parallel$, it should be remembered that this is a derived quantity and some error will be associated with the differentiation of the velocity field. The results involving the linearized approximations in Eqs. (32) and (33) are likely inferior to the interpolated ones due to first-order accuracy of the former. Stokes flow with $N_c = 4$ contains relatively rapid variations, which are better captured by the second-order method. This is evidenced by setting $N_c = 2$, at which degree the two methods produce nearly the same results (not shown). A further confirmation is provided by the results of the next sections where the two methods show comparable performance in smoother flows in which only the lowest order coefficients ($n \leqslant 2$ in the summations) prove significant.

5.3. Drag on a sphere in a triply-periodic array

In this section we examine the accuracy of the SVD and SPs in a pressure-driven flow through a triply-periodic array of spheres. As in [6], we write the pressure gradient as

$$\nabla p = P \mathbf{e}_z + \nabla \hat{p}, \tag{38}$$

in which $\hat{p}$ is periodic, $\mathbf{e}_z$ is the unit vector in the flow direction z, and P is a constant. By applying a momentum balance in the flow direction at steady state, it can be shown that the force $\mathbf{F} = F_z \mathbf{e}_z$ on the particle is given by

$$F_z = (1 - \beta) L^3 P, \tag{39}$$

where $\beta = \frac{4}{3} \pi a^3 / L^3$. We use a periodic domain of size $L = 4a$, with the particle fixed at the center. The flow is initially stationary, at which time a pressure gradient $P_* = a^3 P / \mu v = 10$ is applied. This accelerates the flow until it reaches a steady
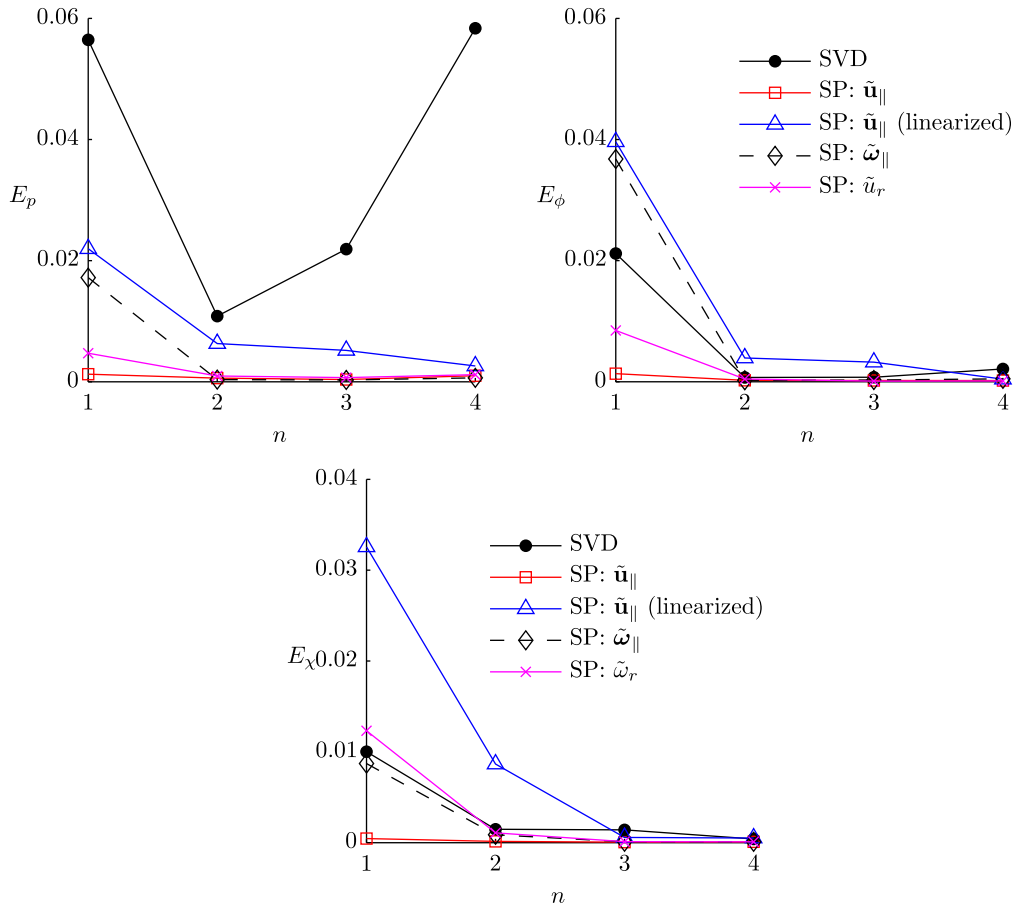
Fig. 2. Accuracy of the recovery of the coefficients p_{nm}, ϕ_{nm} and χ_{nm} from a Stokes flow constructed by assigning coefficients in Eqs. (11) and (12). All coefficients are set simultaneously and the error measured according to Eq. (37). Parameters: truncation order $N_c = 4$, mesh lengths per radius $a/h = 10$, and SPs taken at $r/a = 1.5$.

state where the drag force on the particle matches that applied by the pressure gradient, resulting in a Reynolds number of $Re = 2aU/v = 22.6$, where U is the mean flow rate. The accuracy of the force calculation in Eq. (14) can then be gauged by comparing with the exact result in Eq. (39).

Using a resolution of $a/h = 8$, we calculate the error $|1 - F_z/(1 - \beta)L^3 P|$ as a function of N_c, with the SPs being computed at $r/a = 1.25$. Fig. 3 show the results so obtained. Good results are obtained for all methods, with the error being less than 1% for $N_c \geqslant 3$, as was the case in [6]. Notable is the performance of SP involving $\bar{\omega}_\parallel$, suffering the lowest error out of all the methods, including the SVD. The two SPs involving $\bar{\mathbf{u}}_\parallel$ now prove comparably accurate, with the interpolated one only having a slight advantage. This is likely due to the smoothness of the flow field, as the $n = 1$ coefficients are dominant. Incidentally, these are the coefficients necessary for the force calculation as can be seen in Eq. (14). In contrast, the SP involving $\bar{u}_r$ suffers the largest error of all. While all velocities vanish at the surface of the particle, the radial component does so more rapidly than the transversal ones. This causes larger errors in the interpolation, as the signal-to-noise ratio is diminished. This explains why $\bar{u}_r$ proves less accurate here than in the previous section, where the SPs were computed at $r/a = 1.5$.

Since the SPs are computed at a given distance from the particle surface, this test case provides an opportunity to explore the extent of the Stokes region, at least for this flow configuration. As one moves closer to the surface, SPs involving velocity will at some point be increasingly in error as the velocity vanishes. The SPs involving vorticity and pressure will also suffer error in this region, as neither is continuous through the surface of the cage where the boundary conditions are applied. Thus, upon moving towards the cage, one will eventually interpolate through this discontinuity, resulting in error. The error will also grow as we move further from the particle and out of the Stokes region.

Fig. 4 shows error calculations analogous to those in Fig. 3, but now as functions of r/a, with N_c fixed at 3. The behavior of the different SPs with r/a is not uniform, with $\bar{\omega}_\parallel$ having a minimum in error at $r/a = 1.25$, while the minimum for those involving velocities occurs close to $r/a = 1.5$.

Lastly, we note that, as this test case is nearly axisymmetric in the vicinity of the particle, the coefficients χ_{nm} are very small and as such we cannot draw conclusions as to which SP serves best to compute this. This is addressed in the next subsection.
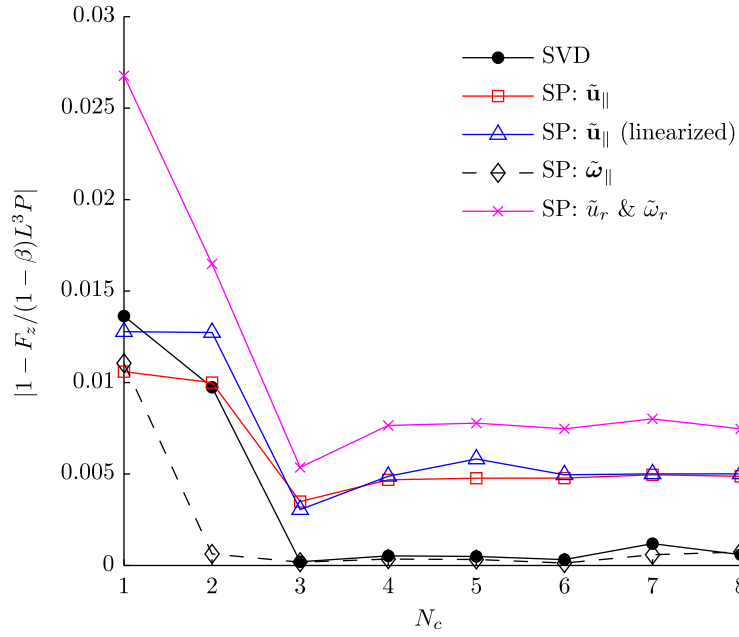
Fig. 3. Momentum balance on a single sphere in a triply-periodic array of spheres with an imposed pressure gradient compared with the analytical result in Eq. (39), as a function of the truncation order N_c. Parameters: cell size $L = 4a$, mesh lengths per radius $a/h = 8$, and SPs computed at $r/a = 1.25$.
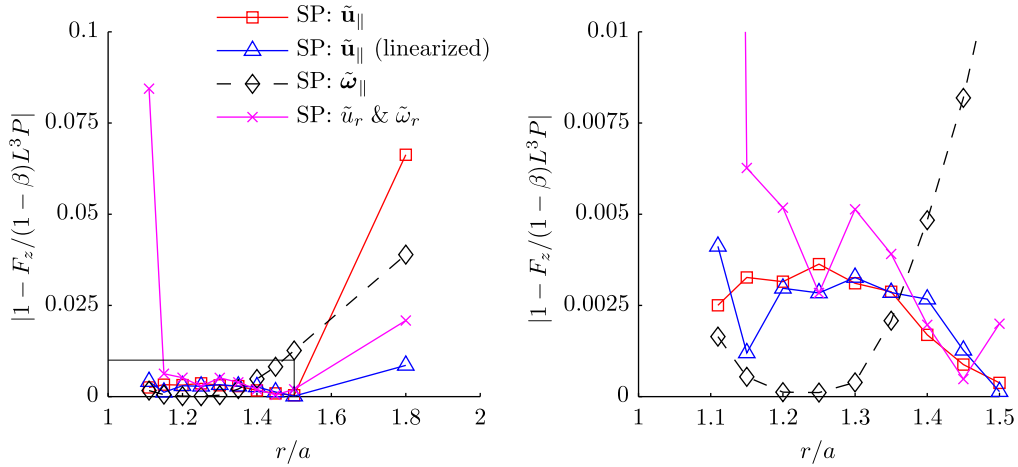
Fig. 4. Error computed as in Fig. 3, but as a function of the distance r/a at which the SPs are computed, with fixed truncation order $N_c = 3$. The figure on the right shows a close-up view of the rectangular region on the left.

5.4. Couple on a sphere in a triply-periodic array

In the previous section we presented results demonstrating the accuracy with which the coefficient calculation by means of the SVD and of different SPs conserves the momentum of flow driven by a pressure gradient. In this section we perform an analogous set of calculations, but this time involving the conservation of angular momentum. This is a stringent test of which few examples, if any, can be found in the literature.

A particle rotated at a constant rate $\boldsymbol{\Omega}$ will impart angular momentum to the surrounding fluid. In turn, in a finite or periodic domain, at steady state the couple at the surface of the particle must be balanced by the couple on the fluid on the outer boundary. The calculation proceeds as follows.

We take the cross-product of $\mathbf{x}$, the position vector relative to the particle center, with the momentum Eq. (1) at steady state and in the absence of gravity, and integrate over the domain

$$\rho \int_V \mathbf{x} \times \boldsymbol{\nabla} \cdot (\mathbf{U}\mathbf{U}) dV = \int_V \mathbf{x} \times (\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}) dV, \tag{40}$$

where

$$\sigma_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right). \tag{41}$$

By the symmetry of the stress tensor we have

$$\mathbf{x} \times (\mathbf{\nabla} \cdot \boldsymbol{\sigma})|_i = \epsilon_{ijk}x_j\frac{\partial \sigma_{kl}}{\partial x_l} = \frac{\partial}{\partial x_l}(\epsilon_{ijk}x_j\sigma_{kl}) - \epsilon_{ijk}\sigma_{kj} = \mathbf{\nabla} \cdot (\mathbf{x} \times \boldsymbol{\sigma})|_i. \tag{42}$$

An application of the divergence theorem transforms Eq. (40) to

$$\rho \int_S \mathbf{x} \times \mathbf{U}(\mathbf{U} \cdot \mathbf{n})dS = \int_S \mathbf{x} \times (\boldsymbol{\sigma} \cdot \mathbf{n})dS + \int_{S_p} \mathbf{x} \times (\boldsymbol{\sigma} \cdot \mathbf{n})dS, \tag{43}$$

where S is the surface of the domain and S_p that of the particle; $\mathbf{n}$ is the normal vector directed out of the fluid. We have dropped the integral over the particle surface in the left-hand side as the radial component of velocity vanishes there. For periodic boundary conditions, which we assume, the remaining integral in the left-hand-side also vanishes. Furthermore

$$\int_{S_p} \mathbf{x} \times (\boldsymbol{\sigma} \cdot \mathbf{n})dS = -\mathbf{L}_p \tag{44}$$

is just the hydrodynamic couple on the particle (with the minus sign due to the direction of the normal) and Eq. (43) therefore becomes

$$\mathbf{L}_p = \int_S \mathbf{x} \times (\boldsymbol{\sigma} \cdot \mathbf{n})dS \equiv \mathbf{L}_b. \tag{45}$$

We take $\mathbf{\Omega} = \Omega_z\mathbf{e}_z$ so that $\mathbf{L}_p = L_p\mathbf{e}_z$ and $\mathbf{L}_b = L_b\mathbf{e}_z$. We use a domain of size $4a$, a grid resolution of $a/h = 8$, and a rotation rate of $Re = a^2\Omega_z/v = 20$. We compute the error $|1 - L_b/L_p|$, as a function of N_c, similarly to the previous section. Fig. 5 shows the results. With the exception of $\tilde{\omega}_\parallel$, all SPs prove more accurate than the SVD for $N_c \geqslant 2$. At first glance, this seems at odds with the results of the previous section, where $\tilde{\omega}_\parallel$ was the most accurate SP. However, with $\tilde{\omega} = \Omega_z\mathbf{e}_z$, $\tilde{\omega}_\phi = 0$ and, therefore, $\tilde{\omega}_\parallel = \tilde{\omega}_\theta\mathbf{e}_\theta$. Hence, the second component of this SP is lost. Further, a significant portion of $\tilde{\omega}_\theta$ is associated with the solid-body rotation of the particle, which gets subtracted out in the frame of the particle, decreasing the signal-to-noise ratio in the calculation of $\tilde{\omega}_\theta$, which in turn causes higher error levels. In contrast, the SP associated with $\tilde{\omega}_r$ now performs best out of all candidates, which is also at odds with the results of the previous section. As in the previous section, the SPs involving the interpolated, and linearized, estimates of $\tilde{\mathbf{u}}_\parallel$ provide practically the same results.

We also compute the above error for fixed $N_c = 3$, but varying the r/a at which the SPs are computed. Fig. 6 shows the results so obtained. As was the case in Fig. 4, the error for different SPs does not behave uniformly in r/a. Generally, though, it seems that there is some leeway in the placement of r/a, with $1.1 \leqslant r/a \leqslant 1.5$ providing reasonable results.
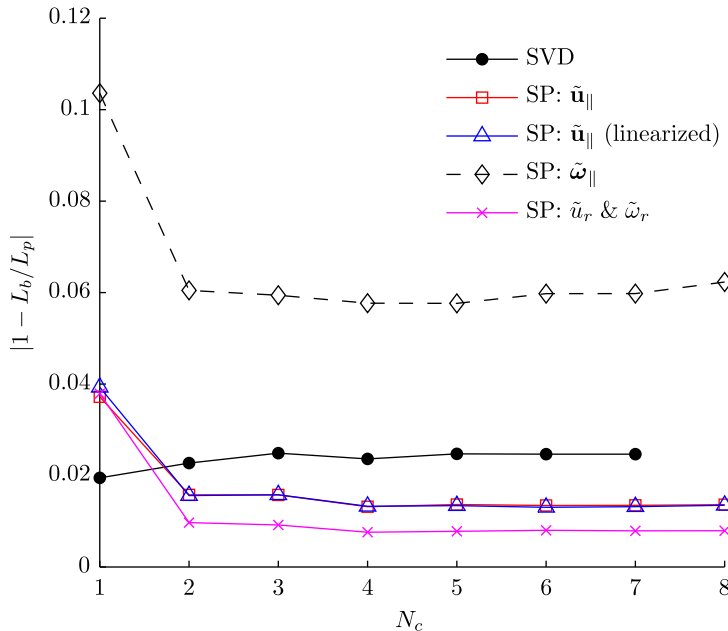
Fig. 5. Error in the calculation of the couple on a particle rotating at a constant velocity as a function of the truncation order N_c. The results for SVD at $N_c = 8$ are absent due to convergence failure. Parameters: $L = 4a$, $a/h = 8$, and SPs computed at $r/a = 1.25$.

The difference between the performance of the various SPs in this section and the previous section is instructive. Clearly, which SP will perform best depends on the flow conditions. If, for example, $\bar{\omega}_r$ is very small, it will not deliver a reliable estimate of the χ_{nm} (via Eq. (30)). On the whole, it seems that the combination of $\tilde{p}$ and $\tilde{\mathbf{u}}_\parallel$ provides consistently good estimates across widely varying flow conditions. One would further suggest using the linearization over the interpolation, given the simplicity of the former and its other advantages described in Section 4.

5.5. A falling particle

The test cases in the previous sections were all for non-translating particles. We now describe some tests on a freely falling particle in a parallelepipedal container of length $20a$ and cross section $8a \times 8a$. The boundary conditions in the lateral and vertical directions are periodic and no-slip, respectively. The ratio of the particle density to the fluid density is $\rho_p/\rho = 2$, the kinematic viscosity is $v = 0.2731$ m²/s, and the acceleration of gravity has the standard value $\mathbf{g} \cdot \mathbf{e}_z = -9.81$ m/s². The simulation is carried out with $a/h = 6$ and 8, respectively using $N_c = 2$ and 3. The time-step in all cases was $v\Delta t/a^2 = 0.005$, resulting in a CFL number of 0.45 at $a/h = 8$. In light of the previous results, we narrow our scope of SPs to those involving the linearization of $\tilde{\mathbf{u}}_\parallel$ and $\tilde{p}$, and those involving the quadratically interpolated $\bar{\omega}_\parallel$ and $\tilde{p}$.

The particle is released from rest at $t = 0$. If no provision is made, at $t = 0$ the weight of the particle would be suddenly imposed on the fluid, thus generating strong force oscillations. To avoid this artifact we have ramped up the value of the acceleration of gravity in an exponential manner over a time $0 \leqslant vt/a^2 \leqslant 0.1$.

Fig. 7 shows the evolution of the Reynolds number $Re = 2aw/v$. In an unbounded fluid the steady state Reynolds number predicted by the Schiller–Naumann correlation [22] is 25, but, due to the lateral constraint, here it is close to 22.5 in all cases. Essentially, the effect of the constraint can be viewed as a blockage effect due to the image particles. While the steady-state Reynolds number varies slightly between the different methods and at the two different resolutions, the largest difference is less than 3%.

Since the cage changes with time, for the reasons explained in Section 5.1, this is also a good test case to examine the CPU-time used by particle-related operations. Fig. 8 shows the cumulative time in seconds thus expended, as measured using an Intel 2.66 GHz CPU. As expected, both SPs provide a significant speed-up, particularly for $a/h = 8$. For $a/h = 6$, the SPs have cumulatively cost roughly 14 s at $vt/a^2 = 1$, while the SVD costs roughly 109 s, larger by a factor of 8. The cost associated with the SVD increases by a factor of 6 upon increasing the resolution to $a/h = 8$, while that of the SPs only increases by a factor of 2 (recall that $N_c = 2$ and 3 respectively for $a/h = 6$ and 8). This is consistent with the time measurements demonstrated in Fig. 1. The cumulative savings in the cost of particle-related operations would be proportionally increased in the presence of multiple particles. Lastly, the cost associated with the SP based on the linearized velocity and pressure is roughly 50% less than that based on the interpolated transversal vorticity and pressure.

It is generally the case with immersed-boundary methods that force oscillations are produced as the particle translates with respect to the underlying grid. It was shown in [6] that PHYSALIS is not immune from this problem, and it is therefore of interest to examine the performance of the SPs also in this respect.

Generally speaking, the origin of the force oscillations must be sought in the minor discontinuities that the calculated force undergoes as the particle moves. The same mechanism is responsible for the force oscillations encountered upon a
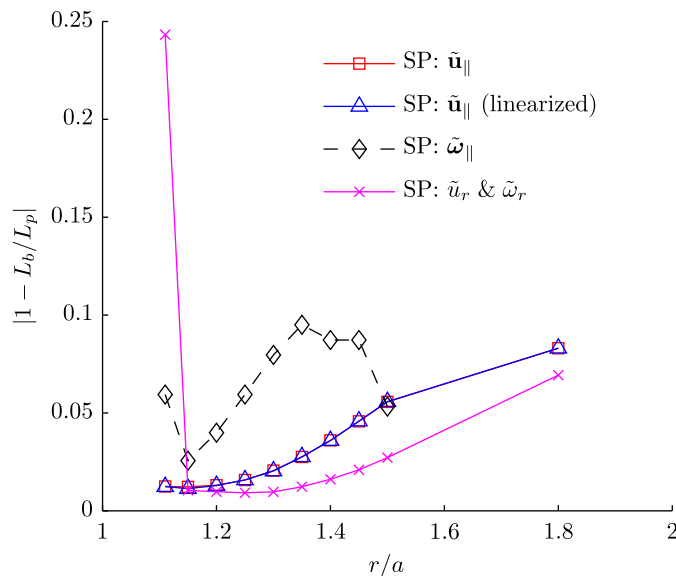
Fig. 6. Error in the calculation of the couple on a particle rotating at constant velocity, as a function of the r/a at which the SPs are computed. The results for $\bar{\omega}_\parallel$ at $r/a = 1.8$ are absent due to convergence failure. Parameters: $L = 4a$, $a/h = 8$, and $N_c = 3$.
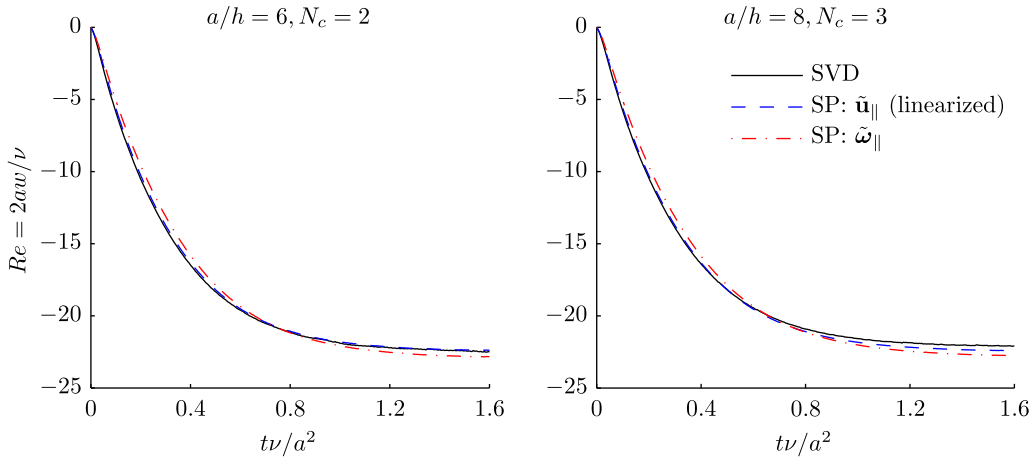
Fig. 7. Evolution of the particle Reynolds number $Re = 2aw/\nu$, for two levels of spatial resolution.
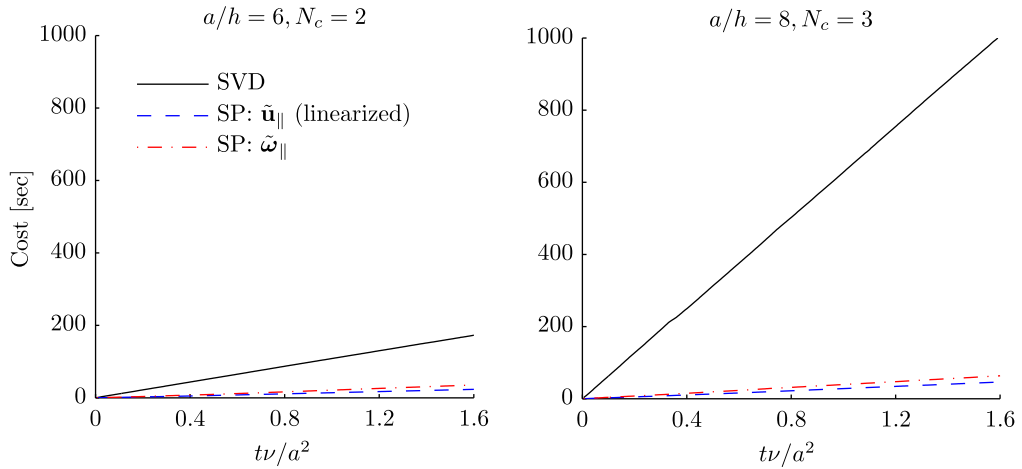
**Fig. 8.** Cumulative time spent on particle-related operations.

sudden release of the particle as mentioned before. A discontinuity in the hydrodynamic force on the particle, by the action-reaction principle, causes an impulsive force on the fluid, which responds with a pressure impulse. This is the reason why the artificial compressibility approach of [13,14] suffers from this difficulty much less than the original PHYSALIS.

In our method, at the end of each time-step, the position of the particle center is compared with that of the cage center and, if the difference is greater than $h/2$, the cage is displaced [6]. As a consequence, a different set of nodes is used to compute the coefficients at the next time step, and this causes an unavoidable small discontinuity in the force calculation even though the displacement of the cage is small enough that both the old and the new cages are entirely in the Stokes region. This mechanism leads one to expect that the force oscillations decrease with a refinement of the grid, as it was indeed observed in [6] and as will be shown presently. Another contributing factor may be the fact that the location at which the boundary conditions are applied on the fluid is different when the cage is moved. Furthermore, some of the grid points that were previously inside the cage become fluid points and need to be re-initialized as such, as described by [6]. However, this is likely not a significant source of force oscillations due to the iterations in the inner loop (steps 1–4 in Section 2).

Fig. 9 shows the time evolution of the normalized force on the particle for $a/h = 6$ and 8. It should be noted that, for clarity, the curves for the original SVD method and the SP with $\tilde{\omega}_\parallel$ have been translated by ± 0.1. The expected decrease of the amplitude of oscillations with increasing grid refinement is evident for all three methods, although the SVD curves show significant spikes even with the finer grid. The amplitude of the oscillations with both SP methods is smaller, does not present any spikes and exhibits a greater reduction on the finer grid. The greatest reduction is obtained by the SP with the linearly interpolated $\tilde{\mathbf{u}}_\parallel$ which thus is seen to perform well also from this point of view.

5.6. Taking the scalar products closer to the particle–surface

In the above sections, the SPs were taken over a sphere with a radius r with $r/a > 1 + ch/a$ with $1 \leqslant c \leqslant 4$, where a is the radius of the particle. While this does not raise concerns in single-particle simulations, it might under circumstances in
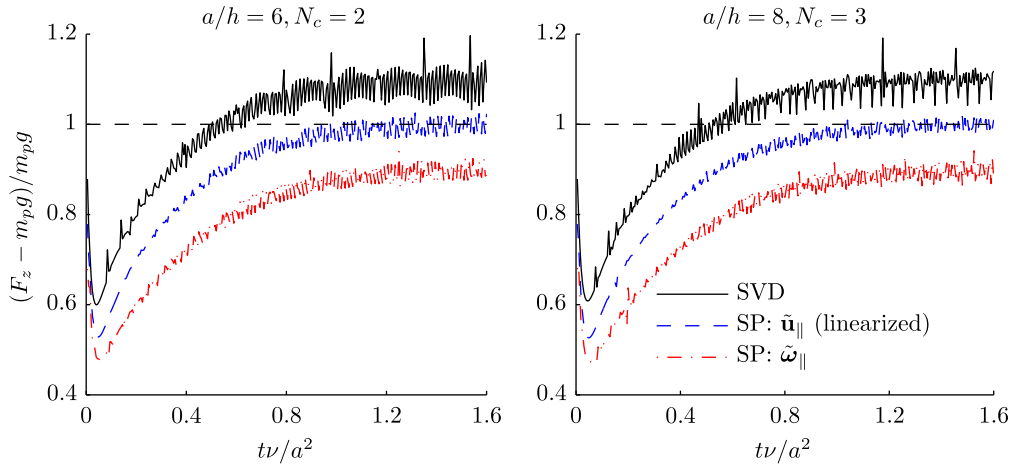
Fig. 9. Evolution of normalized hydrodynamic force in the direction of gravity, for a particle falling from rest between two plates, with periodic boundary conditions in the directions normal to gravity. For readability, the curves representing the SVD and the SP involving $\tilde{\omega}_\parallel$ are respectively shifted by $+0.1$ and -0.1.

which particles are expected to be in close proximity (such as for flow through a tightly-packed porous medium) and even to collide. In this section we show that $c \geqslant 1$ is not an intrinsic limitation of our method and that even $r/a = 1$ is permissible, allowing inter-particle contact.

As noted before, vorticity and pressure are not continuous through the cage. This is the reason why, for a given cage size R_c/a and grid resolution, there exists a lower limit on c below which the calculation of vorticity and pressure becomes in error. While this is the case for both the interpolation and local linearization, the latter is less affected due the smaller number of cells involved, thus allowing a smaller c before the cage is breached.

The constraint of operating outside the cage can be met either by using r/a greater than 1, as done before, or by decreasing R_c/a as we now show. The latter option is permissible since the analytic expressions (11) and (12) embody a smooth analytic continuation of the pressure and velocity fields inside the particle. Thus, the boundary conditions on the cage may be applied inside the particle as well. One feature to keep in mind, however, is the singularity at the particle enter which causes rapid variations of the fields inside the particle and a progressively increasing error as R_c decreases. For this reason, we limit ourselves to considering $R_c/a = 1$ and 0.9.

Fig. 10 shows the significant error that use of $\tilde{\omega}_\parallel$ incurs when $R_c/a = 1$ and $r/a = 1.05$. In this case calculation of the vorticity at the quadrature points requires information on the interior side of the particle surface across which the vorticity is discontinuous as noted before. However, by placing the cage at $R_c/a = 0.9$, the problem is eliminated and the SP involving $\tilde{\omega}_\parallel$ gives excellent results even with $r/a = 1$. Actually, in all the examples considered, except that of Section 5.4, this option gives
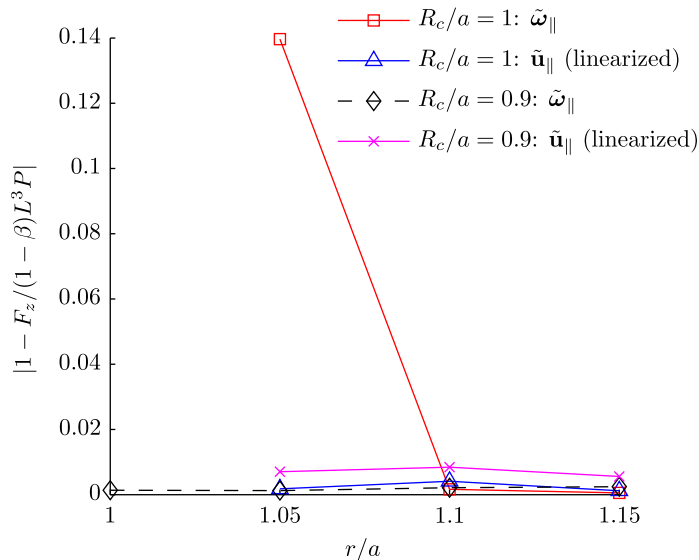
Fig. 10. Error for the same flow configuration as in Section 5.3, as a function of the r/a at which the SPs are taken, and the size of the cage, R_c/a. Note that the SP involving $\tilde{\omega}_\parallel$ can be taken right at the particle surface, provided that the cage is inside the particle.

excellent results and the associated computational overhead is only modestly larger than for the linear interpolation method. Fig. 10 shows that use of the scalar products involving the linearly interpolated transverse velocity also gives excellent results at a somewhat reduced computational cost.

6. Summary and conclusions

In this work we have presented a new approach to calculating the unknown expansion coefficients in Lamb's solution for the near-particle flow field in the PHYSALIS method. In the previous approach of [6], the coefficients were obtained via the SVD solution of a rectangular, relatively large linear system of equations formed by equating the analytical expansion, evaluated at the nodes of the finite-difference field around the particle, to the finite-difference field itself (cf. Eq. (16)). In the new approach we take advantage of the orthogonality of the set of vector harmonics by taking suitable scalar products thereof with values obtained from the finite-difference field.

We have described how the finite-difference field can be estimated on the spherical surface over which the scalar products are taken and further how the relevant integrals are computed efficiently via a combination of discrete Fourier and Legendre transforms on the sphere.

There are several options for which fields should be used to calculate the coefficients. Our results suggest that the a local linear interpolation method based on the pressure and the transverse velocity (Section 4.2) is efficient and accurate provided the scalar product can be computed on a surface with a radius exceeding that of the particle by 1–2 mesh lengths. If the cage is placed inside the particle, this method works well also when particles are very close or can collide (Fig. 10). Another good option in this latter case is the use of the transverse vorticity in addition to the pressure (Sections 3.1 and 3.2).

Via various examples involving stationary, rotating, and sedimenting particles, we have demonstrated that the new approach is more accurate, more stable in terms of force-oscillations, and significantly faster than the SVD approach of [6]. The speed-up is particularly significant at higher Reynolds numbers, as $h \sim Re^{-3/4}$. As the number of points comprising the cage scales as $Re^{3/2}$, the cost of the SVD scales as $Re^{3/2}N_c^4$, (cf. Section 5.1). While the same resolution is of course required when using the scalar product, the associated cost scales only as $N_c^2 \log N_c$. The speed-up reported in this paper are therefore likely to be further enhanced at higher Reynolds numbers.

Acknowledgments

Appendix A

The Y_n^m, in Eq. (13) and others, are the normalized orthogonal spherical harmonics:

$$Y_n^m(\theta, \phi) = N_n^m P_n^m(\cos\theta)e^{im\phi}, \quad N_n^m = \sqrt{\frac{2n+1}{4\pi}\frac{(n-m)!}{(n+m)!}}, \tag{46}$$

where θ and φ are respectively the polar and azimuthal angles, and $m = 0, \pm 1, \ldots, \pm n$. For $m \geqslant 0$, the associated Legendre polynomials P_n^m are given by

$$P_n^m(\cos\theta) = (-1)^m(\sin\theta)^m\frac{d^m}{d(\cos\theta)^m}P_n(\cos\theta), \tag{47}$$

while

$$P_n^{-m} = (-1)^m\frac{(n-m)!}{(n+m)!}P_n^m. \tag{48}$$

The constant N_n^m is included in order to render the Y_n^m orthonormal as shown in (17).

The regular and singular solid harmonics can be related through the no-slip boundary condition on the particle surface. The resulting expressions are

$$p_{-n-1} = -\frac{1}{2}\frac{2n-1}{n+1}n[p_n + 2(2n+1)\phi_n]\left(\frac{a}{r}\right)^{2n+1} \tag{49}$$

$$\phi_{-n-1} = -\frac{1}{4}\frac{n}{n+1}\left[\frac{2n+1}{2n+3}p_n + 2(2n-1)\phi_n\right]\left(\frac{a}{r}\right)^{2n+1}, \quad\text{and} \tag{50}$$

$$\chi_{-n-1} = -\left(\frac{a}{r}\right)^{2n+1}\chi_n. \tag{51}$$

References

[1] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476.

[2] M. Uhlmann, Interface-resolved direct numerical simulation of vertical particulate channel flow in the turbulent regime, Phys. Fluids 20 (5) (2008) 053305.
[3] F. Lucci, A. Ferrante, S. Elghobashi, Modulation of isotropic turbulence by particles of Taylor length-scale size, J. Fluid Mech. 650 (2010) 5–55.
[4] A. Prosperetti, H.N. Oguz, Physalis: a new o(N) method for the numerical simulation of disperse systems: potential flow of spheres, J. Comput. Phys. 167 (1) (2001) 196–216.
[5] S. Takagi, H. Oguz, Z. Zhang, A. Prosperetti, PHYSALIS: a new method for particle simulation – part II: two-dimensional Navier–Stokes flow around cylinders, J. Comput. Phys. 187 (2) (2003) 371–390.
[6] Z. Zhang, A. Prosperetti, A second-order method for three-dimensional particle simulation, J. Comput. Phys. 210 (2005) 292–324.
[7] Z. Zhang, L. Botto, A. Prosperetti, Microstructural effects in a fully-resolved simulation of 1024 sedimenting spheres, in: IUTAM Symposium on Computational Approaches to Multiphase Flow, 2006, pp. 1–10.
[8] A. Naso, A. Prosperetti, The interaction between a solid particle and a turbulent flow, New J. Phys. 12 (2010) 033040.
[9] L. Botto, A. Prosperetti, A fully resolved numerical simulation of turbulent flow past one or several particles, Phys. Fluids 24 (2012) 013303.
[10] Q. Liu, A. Prosperetti, Wall effects on a rotating sphere, J. Fluid Mech. 657 (2010) 1–21.
[11] Q. Zhang, A. Prosperetti, Pressure-driven flow in a two-dimensional channel with porous walls, J. Fluid Mech. 631 (2009) 1–21.
[12] Q. Liu, A. Prosperetti, Pressure-driven flow in a channel with porous walls, J. Fluid Mech. 679 (2011) 77–100.
[13] A. Perrin, H.H. Hu, An explicit finite-difference scheme for simulation of moving particles, J. Comput. Phys. 212 (1) (2006) 166–187.
[14] A. Perrin, H.H. Hu, An explicit finite difference scheme with spectral boundary conditions for particulate flows, J. Comput. Phys. 227 (20) (2008) 8776–8791.
[15] H. Lamb, Hydrodynamics, Cambridge University Press, 1932.
[16] S. Kim, S.J. Karrila, Microhydrodynamics, Principles and Selected Applications, Dover Publications, 2005.
[17] A. Prosperetti, Advanced Mathematics for Applications, Cambridge University Press, 2011.
[18] D. Healy, D.N. Rockmore, P. Kostelec, S. Moore, FFTs for the 2-sphere-improvements and variations, J. Fourier Anal. appl. 9 (4) (2003) 341–385.
[19] B. Gough (Ed.), GNU Scientific Library Reference Manual, third ed., Network Theory Ltd., 2009.
[20] C.L. Lawson, R.J. Hanson, Solving Least Squares Problems, Prentice Hall, 1974.
[21] C.E. Shannon, Communication in the presence of noise, Proc. IRE 37 (1) (1949) 10–21.
[22] L. Schiller, A. Naumann, Fundamental calculations in gravitational processing, Z. Ver Dtsch. Ing. 77 (1933) 318–320.