Journal of Computational Physics 309 (2016) 164–184

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Resolved-particle simulation by the Physalis method: Enhancements and new capabilities

Adam J. Sierakowski [a,*], Andrea Prosperetti [a,b]

[a] Department of Mechanical Engineering, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA
[b] 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

ARTICLE INFO

Article history:
Received 27 July 2015
Received in revised form 21 December 2015
Accepted 30 December 2015
Available online 6 January 2016

Keywords:
Resolved particle numerical simulation
Disperse multiphase flow
Physalis method
Spherical particle
Computational fluid dynamics

ABSTRACT

We present enhancements and new capabilities of the Physalis method for simulating disperse multiphase flows using particle-resolved simulation. The current work enhances the previous method by incorporating a new type of pressure-Poisson solver that couples with a new Physalis particle pressure boundary condition scheme and a new particle interior treatment to significantly improve overall numerical efficiency. Further, we implement a more efficient method of calculating the Physalis scalar products and incorporate short-range particle interaction models. We provide validation and benchmarking for the Physalis method against experiments of a sedimenting particle and of normal wall collisions. We conclude with an illustrative simulation of 2048 particles sedimenting in a duct. In the appendix, we present a complete and self-consistent description of the analytical development and numerical methods.

1. Introduction

Progress in algorithms and computing hardware has begun to make possible the resolved simulations of the interaction of many particles within a suspending fluid. The motivation for these studies is the many and diverse occurrences of disperse particle flows in nature and technology in the face of the still very incomplete understanding of many physical aspects of such flows. Though subject to limitations in domain size and particle number, particle-resolved simulations capture the detailed physics of particle–fluid and particle–particle interactions and offer the hope of shedding light on the micromechanics of disperse flows. This in turn may be expected to facilitate the development of reduced-order models, such as the two-fluid models [1], which are the only models able to describe realistic systems at a practical scale.

A landmark contribution to particle-resolved simulations was made by Uhlmann [2] who was able to simulate channel flow of a fluid with 4096 suspended particles by means of an immersed boundary method that he had developed earlier [3]. In the following years, the same method or some variation thereof was used by several researchers. Lucci et al. [4] presented results of the simulation of decaying homogeneous isotropic turbulence with up to 6400 particles. Kempe and Fröhlich [5] improved the satisfaction of the no-slip condition and the collision algorithm of Uhlmann [3]. Picano et al. [6] used a similar method to investigate 10,000 neutrally buoyant spheres in a channel flow. In the most recent simulations, Kidanemariam and Uhlmann [7] have been able to simulate more than 260,000 particles. Somewhat different versions of the immersed boundary method have been pursued by other researchers including Mehrabadi et al. [8], Yang and Stern [9], and

[*] Corresponding author.
E-mail address: sierakowski@jhu.edu (A.J. Sierakowski).

http://dx.doi.org/10.1016/j.jcp.2015.12.057

Ji et al. [10]. Different approaches have also been developed, such as the lattice Boltzmann method [see, e.g. [11,12]], the fictitious domain method [see, e.g., [13,14]] and its variants, the spectral element method [see, e.g. [15]], and the pressure boundary integral method of Simeonov and Calantoni [16].

In this paper we present recent enhancements to the Physalis method, which was developed in Zhang and Prosperetti [17] (henceforth Z&P) and Gudmundsson and Prosperetti [18] (henceforth G&P). Unlike immersed boundary methods, Physalis recognizes each particle as an internal boundary sidestepping the complex geometrical relationship between the particle surface and the fixed underlying discrete grid by means of a suitable analytical solution. The method is based on a spectral representation of the effect of the particles, which guarantees an accurate representation even with a relatively coarse discretization. In addition to the hydrodynamic force, the method produces highly accurate results for hydrodynamic couple, stresslet, and higher-order multipoles associated to each particle. An additional benefit is that these quantities are a byproduct of the solution method itself and do not require specific calculations. Furthermore, unlike most immersed boundary methods, the no-slip boundary condition on each particle is satisfied to analytical accuracy. The price to be paid for these advantages is that the method is iterative, requiring multiple solutions of the fluid mechanical problem at each time step, with the number of iterations depending generally on the complexity of the flow in the neighborhood of a particle. Given the rapid increase of the operation count for the numerical solution of the pressure-Poisson equation with the number of grid nodes, the present method may remain competitive with, or even superior to, immersed boundary methods in spite of its iterative nature because, as detailed in Z&P, the same accuracy can be reached with a significantly smaller number of grid nodes resolving a particle (typically 6 to 10 cells per radius for particle Reynolds numbers less than 150).

In Section 2, we present an overview of the updated Physalis method and detail significant improvements made to the algorithms presented in Z&P and G&P. Section 3 describes our enhancements to the flow solver. We discuss the addition of short-range particle interaction models in Section 4. Section 5 contains validation of the new methods and comparisons to the experiments of Mordant and Pinton [19] on a sedimenting sphere and Joseph et al. [20] on particle–wall collisions. We conclude with Section 6, which contains an example simulation of 2048 particles sedimenting in a duct. Throughout this manuscript, we focus on the improvements to Z&P and G&P, providing only the minimal amount of theory necessary to understand the relevance of the changes. For completeness, we include a detailed appendix containing a concise but complete description of the method.

2. Enhancements to the Physalis method

In brief, the Physalis method is characterized by the procedure used to transfer the no-slip boundary conditions satisfied at the surface of a spherical particle to an underlying fixed grid (see Section 2.1 and Appendix A). The geometry of each particle is not approximated: Physalis maintains a sharp interface and the no-slip boundary condition is satisfied to analytical accuracy. The procedure has many benefits:

1. The particle surface is defined implicitly by tracking only the particle's position with no Lagrangian surface tracking required;
2. The numerical error decreases exponentially with the number of degrees of freedom representing each particle (i.e., expansion coefficients, see Section 2.1), which permits the use of relatively coarse grids without compromising numerical accuracy;
3. The hydrodynamic force, couple, and higher-order multipoles are generated directly in the course of the solution procedure with no need for additional calculations.

2.1. The Physalis method

This version of the Physalis method extends the work presented in Z&P and G&P by introducing a number of algorithmic improvements. As with immersed boundary methods, the goal is to transfer the no-slip condition imposed by a particle onto the Cartesian computational grid. The following description provides a brief overview of the Physalis method. For a more detailed discussion, refer to the appendix.

The Physalis method is used to introduce particles into an incompressible Navier–Stokes flow solver, which solves the equations

$$\partial_t \mathbf{U} + (\mathbf{U}\cdot\nabla)\,\mathbf{U} = -\frac{1}{\rho_f}\nabla p + \nu\nabla^2\mathbf{U} + \mathbf{g}, \tag{1a}$$

$$\nabla\cdot\mathbf{U} = 0, \tag{1b}$$

for velocity $\mathbf{U}$ and pressure p; ρ_f is the fluid density, ν the kinematic viscosity, and $\mathbf{g}$ a body force per unit mass. By considering a reference frame attached to the particle, due to the no-slip condition on the surface, we may linearize the Navier–Stokes equations to the Stokes equations with increasing accuracy as the distance from the surface decreases. For flow about a sphere, in the sphere's rest frame, the Stokes equations admit a general analytic solution, which we call *Lamb's solution* [21,22], of the form

$$\tilde{\mathbf{u}}\,(r,\theta,\varphi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} A_{lm}\tilde{\mathbf{u}}_{lm}\,(r)\,Y_l^m\,(\theta,\varphi)\,, \tag{2a}$$

$$\tilde{p}\,(r,\theta,\varphi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} B_{lm}\tilde{p}_{lm}\,(r)\,Y_l^m\,(\theta,\varphi)\,, \tag{2b}$$

where the tilde quantities are related to (1) through a change of reference frame (see Appendix A.1). Here (r,θ,φ) are local spherical coordinates with origin at the particle center and Y_l^m are spherical harmonics. The time-dependent *Lamb's coefficients* A_{lm} and B_{lm} account for an arbitrary flow in the neighborhood of the particle and are simply related to the force, couple, and higher-order multipoles associated with the particle. The functions $\tilde{\mathbf{u}}_{lm}$ and $\tilde{p}_{lm}$ are simple algebraic functions of r known analytically. In particular, all the $\tilde{\mathbf{u}}_{lm}$ vanish at the particle surface and, as a consequence, (2a) satisfies analytically the no-slip condition at the particle surface irrespective of the order of truncation of the infinite series. Note that the velocity and pressure given by (2) continue analytically the solution of the Stokes equations inside the particles.

For each time step, the solution procedure is iterative and matches Lamb's solution valid near each particle to a global discretized solution. At each step of this sub-time-step iteration, Lamb's coefficients receive flow information from the current Navier–Stokes solution by means of scalar products described in Section 2.3 and Appendix A.3 (see also G&P). Equations (2) are then used to generate new boundary conditions on a set of nodes of the discrete grid near the surface of the particle constituting the so-called *cage*. A new Navier–Stokes solution is calculated and the process is repeated to convergence. Upon convergence, we update the linear and angular particle velocities, $\mathbf{w}$ and $\boldsymbol{\Omega}$, according to

$$m\frac{d\mathbf{w}}{dt} = \mathbf{F}_{\text{total}}, \qquad I\frac{d\boldsymbol{\Omega}}{dt} = \mathbf{L}_{\text{total}}, \tag{3}$$

where m and I are the particle mass and moment of inertia. Here, $\mathbf{F}_{\text{total}}$ and $\mathbf{L}_{\text{total}}$ are the hydrodynamic force and couple provided by Lamb's solution augmented by the collisional contributions and prescribed external forces, if any. The numerical procedure by which these equations are integrated is described in Appendix A.7.

We discuss only the important changes that we have made from previous embodiments of the algorithm, specifically:

1. Improving the particle cage generation;
2. Implementing a more efficient method of calculating the scalar products;
3. Developing a new treatment of the particle interior.

2.2. *Cage construction*

As mentioned above, a set of grid nodes called a cage is associated with each particle. The cage is generated in the following way (see Fig. 1):

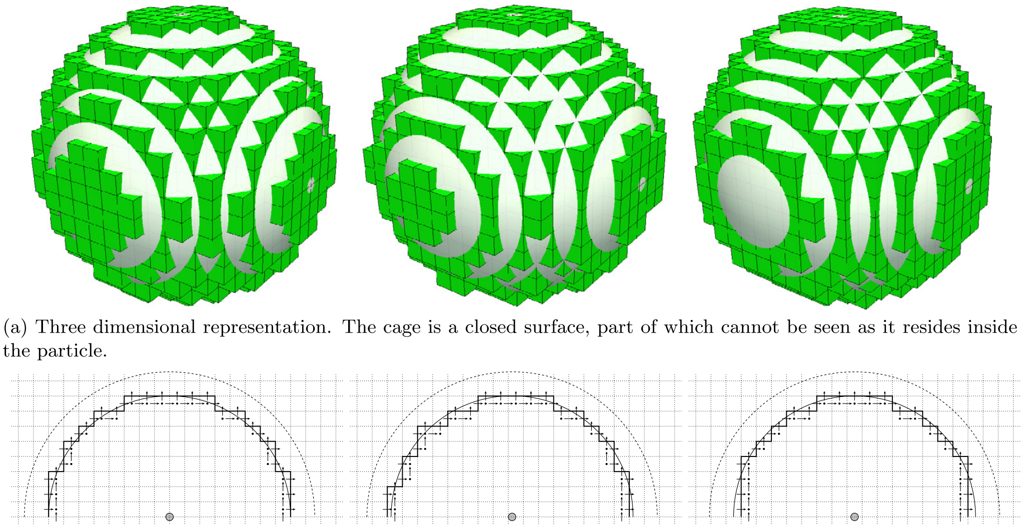
1. Mark all cells with cell centers inside the particle surface; the outermost shell of these cells constitutes the set of cage cells;
2. The centers of the cage cells are the locations at which Dirichlet conditions for the pressure-Poisson equation are imposed;
3. All faces of the cage cells with an outward normal with respect to the cage are part of the set of points where $\tilde{\mathbf{u}}$ is imposed;
4. The set of velocity nodes is completed by including the centers of all faces common to two neighboring cage cells.

The particle center may reside at an arbitrary location within a grid cell and the cage adapts its shape to its position. Fig. 1 illustrates the shape of the cage for three such particle positions: zero, $\frac{1}{4}$, and $\frac{1}{2}$ mesh lengths displaced from the cell corner. In reality, the cage shape changes even more smoothly than what is illustrated in these three discrete positions as particles typically move a small fraction of a cell in each time step. This construction carries with it the beneficial characteristic of significantly stabilizing particle collisions. In spite of the presence of cage nodes outside the particle, unlike the construction of Z&P, any interference between boundary conditions of two very close particles is inconsequential as explained in Section 4.

In contrast to Z&P, this cage generation procedure is so fast that it permits the generation of cages for all particles at each time step, providing a very smooth transition between cage shapes as particles move with respect to the Cartesian grid. The no-slip condition satisfied at the particle surface guarantees that the center of a particle moves a fraction of a cell by adaptively determining the time step size using the fluid velocity field to maintain a specified CFL condition number as discussed in Section 3.4. Further, as a margin of safety, the time step size calculation takes into account the fluid cells inside the particle surface, which are set equal to the solid body velocity of the particle as explained in Section 2.4. As a consequence, at each time step, only a small number of cells are added to or removed from the cage, which minimizes the unavoidable force fluctuations affecting all fixed-grid methods. Finally, the more smoothly transitioning cages assist Lamb's coefficient convergence.

2.3. *Lebedev quadrature scalar products*

As explained in Appendix A.3, the coefficients of (2) are calculated by taking scalar products of the type



(a) Three dimensional representation. The cage is a closed surface, part of which cannot be seen as it resides inside the particle.

(b) Two-dimensional planes through the centers of the Physalis cages in (a). Velocity and pressure boundary conditions are denoted by arrows and dots, respectively. The dashed line represents the integration surface used for the scalar products (4).

Fig. 1. Three examples of the Physalis particle cage that specifies the locations of the analytic boundary conditions. The particles are positioned with center points at displacements 0, $\Delta x/4$, and $\Delta x/2$ from a cell corner.

$$\left(Y_l^m, \tilde{p}\right) = \int_{-\pi}^{\pi} d\varphi \int_0^{\pi} \sin\theta \, d\theta \, \bar{Y}_l^m(\theta, \varphi) \, \tilde{p}(r_i, \theta, \varphi), \tag{4}$$

where the overbar denotes complex conjugation and r_i is the integration radius discussed below. We have found that the use of an integration scheme based on Lebedev quadrature [23] in place of the differential angular integration scheme used in G&P provides a more efficient algorithm for the evaluation of these integrals.

A Lebedev quadrature rule of order l integrates exactly spherical harmonics up to order l at the expense of function evaluations at a relatively small number of nodes on the unit sphere. In most of our simulations we use a maximum order of spherical harmonics $l = 3$ (and occasionally $l = 4$) and therefore, ideally, we would need an integration rule capable of integrating spherical harmonics of order 6 (or possibly 8). Lebedev rules are available for orders 5, 7, 11, and higher. We have chosen the seventh-order rule that requires only 26 quadrature nodes as opposed to 50 nodes for the 11th-order rule; it takes the form

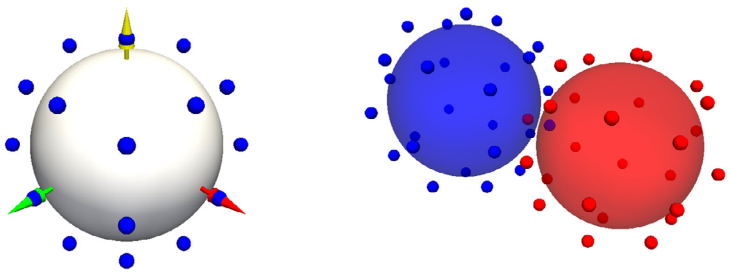
$$\frac{1}{4\pi} \int_{-\pi}^{\pi} d\varphi \int_0^{\pi} \sin\theta \, d\theta \, f(\theta, \varphi) \approx A_1 \sum_{i=1}^{6} f\left(a_i^1\right) + A_2 \sum_{i=1}^{12} f\left(a_i^2\right) + A_3 \sum_{i=1}^{8} f\left(a_i^3\right). \tag{5}$$

The Lebedev quadrature node weights $\{A_1, A_2, A_3\}$ and positions $\left\{a_i^1, a_i^2, a_i^3\right\}$ are reproduced in Table 1 and illustrated in Fig. 2(a). We interpolate the flow field values at the quadrature nodes from the Cartesian grid using trilinear interpolation as described in G&P.

Since Lamb's coefficients depend on the flow field in the neighborhood of the sphere, the surface of integration for the scalar products must be located at a radial location r_i outside the sphere, but still within the region of validity for the Stokes approximation. We refer to the work of G&P for an analysis of the accuracy of various choices of the ratio of r_i to the particle radius a; we typically choose $1.1 < r_i/a < 1.25$. We must take special care to treat the situation of particles sufficiently near to each other so that their Lebedev quadrature nodes interfere as illustrated in Fig. 2(b); we discuss the treatment of these cases in the following section.

2.4. Particle interior

Our solution procedure is inherently independent of the particle interiors as long as particles are sufficiently separated. However, to maintain this independence when a particle approaches another so that any of its Lebedev quadrature nodes fall inside the other particle, we should have recourse to a different integration scheme, which is undesirable from the perspective of code efficiency. To be able to continue using the Lebedev scheme, we set the interior velocity field of every particle equal to the solid body (linear and angular) motion of the particle and the pressure (essentially) equal to the mean pressure within the Stokes region near the particle (see Appendix A.2).



(a) The locations of the quadrature nodes used in the seventh-order Lebedev quadrature rule. The arrows represent the three Cartesian directions.

(b) An example of interfering Lebedev quadrature nodes, with the red node belonging to the red particle residing inside the blue particle (and likewise for the blue node).

Fig. 2. Illustrations of Physalis particles with their Lebedev quadrature scalar product nodes. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 1
Lebedev quadrature nodes and weights to be used in (5). We choose θ to be the polar angle.

N	a_i^1	(θ,φ)	N	a_i^2	(θ,φ)	N	a_i^3	(θ,φ)
1	a_1^1	$\left(\frac{\pi}{2},\frac{\pi}{2}\right)$	7	a_1^2	$\left(\frac{\pi}{2},\frac{\pi}{4}\right)$	19	a_1^3	$\left(\alpha_1,\frac{\pi}{4}\right)$
2	a_2^1	$\left(\frac{\pi}{2},\pi\right)$	8	a_2^2	$\left(\frac{\pi}{2},\frac{3\pi}{4}\right)$	20	a_2^3	$\left(\alpha_1,\frac{3\pi}{4}\right)$
3	a_3^1	$\left(\frac{\pi}{2},\frac{3\pi}{2}\right)$	9	a_3^2	$\left(\frac{\pi}{2},\frac{5\pi}{4}\right)$	21	a_3^3	$\left(\alpha_1,\frac{5\pi}{4}\right)$
4	a_4^1	$(0,0)$	10	a_4^2	$\left(\frac{\pi}{2},\frac{7\pi}{4}\right)$	22	a_4^3	$\left(\alpha_1,\frac{7\pi}{4}\right)$
5	a_5^1	$(\pi,0)$	11	a_5^2	$\left(\frac{\pi}{4},0\right)$	23	a_5^3	$\left(\alpha_2,\frac{\pi}{4}\right)$
6	a_6^1	$\left(\frac{\pi}{2},0\right)$	12	a_6^2	$\left(\frac{\pi}{4},\frac{\pi}{2}\right)$	24	a_6^3	$\left(\alpha_2,\frac{3\pi}{4}\right)$
			13	a_7^2	$\left(\frac{\pi}{4},\pi\right)$	25	a_7^3	$\left(\alpha_2,\frac{5\pi}{4}\right)$
			14	a_8^2	$\left(\frac{\pi}{4},\frac{3\pi}{2}\right)$	26	a_8^3	$\left(\alpha_2,\frac{7\pi}{4}\right)$
			15	a_9^2	$\left(\frac{3\pi}{4},0\right)$			
			16	a_{10}^2	$\left(\frac{3\pi}{4},\frac{\pi}{2}\right)$			
			17	a_{11}^2	$\left(\frac{3\pi}{4},\pi\right)$		$\alpha_1=\arccos\left(3^{-1/2}\right)$	
			18	a_{12}^2	$\left(\frac{3\pi}{4},\frac{3\pi}{2}\right)$		$\alpha_2=\arccos\left(-3^{-1/2}\right)$	
$A_1=0.598398600683775$			$A_2=0.478718880547015$			$A_3=0.403919055461543$		

This scheme provides the most natural and well-behaved treatment of the particle interference situation because, due to the no slip condition, it guarantees a smooth transition of velocity as a quadrature node traverses the surface of a neighboring particle. This procedure requires no special treatment of these interfering quadrature nodes as they simply interpolate an appropriate value under all particle configurations. The effect of any residual error is mitigated by the fact that the fraction of offending nodes is typically very small ($\sim 1/26$). We also apply this scheme to the case of a particle interacting with a no-slip domain boundary, where the velocity of the quadrature node is interpolated from the Dirichlet velocity specified on the boundary.

In past work of Z&P, internal flow was generated by the solution procedure because the process of solving the momentum equation (1a) everywhere in space simplified the numerical algorithm. We have found in the current work, however, that the internal velocities generated in this way can be large, thus unnecessarily limiting the time step. The present method overcomes this limitation.

3. Navier–Stokes flow solver

Physalis may be coupled to any Navier–Stokes equations solver, and we have chosen a fairly standard second-order finite difference fractional step method in a staggered arrangement on a uniform Cartesian grid [24–26]. The implementation can handle the standard set of boundary conditions on the outer boundary of the domain: inhomogeneous Dirichlet, homogeneous Neumann, and periodic. While the work presented in Z&P claimed to use fully second-order numerical schemes, we have found that the pressure-free formulation used to solve the Navier–Stokes equations achieved first-order accuracy only for pressure near boundaries and have improved the scheme to realize second-order accuracy for both pressure and velocity following the pressure-correction method of Brown et al. [26]. For some details on the numerical treatment of the flow solver, refer to Appendix A.4.

We have also made a number of changes to previous versions of the flow solver:

1. We take advantage of our analytic solution (2b) for pressure by applying it as a Dirichlet boundary condition at the cage nodes in the solution of the pressure-Poisson equation;
2. We have replaced the fast Poisson solver with a conjugate gradient algorithm;
3. We forgo the traditional implicit treatment of the diffusive term $\left(\nu\nabla^2\mathbf{U}\right)$ in (1a) in favor of more rapid explicit numerics;
4. We have adopted an adaptive time stepping scheme that chooses the largest stable time step size according to flow conditions.

3.1. Dirichlet pressure boundary conditions

The fractional step method that we adopt for solving the Navier–Stokes equations [26] requires the solution of a pressure-Poisson equation of the form

$$\nabla^2\phi = \frac{\rho_f}{\Delta t}\nabla\cdot\mathbf{U}^*, \tag{6}$$

where ϕ represents the pressure correction required to ensure that the solution to the momentum equation (1a) satisfies the continuity equation (1b), $\mathbf{U}^*$ is an intermediate velocity field (see Appendix A.4), and Δt is the discrete time step size. In Z&P, this equation was solved by imposing homogeneous Neumann conditions on the cage nodes as is usually done with the fractional step method. In the present work we have improved on this approximate procedure by imposing the analytically accurate pressure boundary condition (2b) provided by Lamb's solution.

In addition to providing a more physically accurate boundary condition than the zero-normal-gradient condition, this new procedure carries with it a number of desirable numerical consequences:

1. Lamb's solution converges in fewer iterations for the same overall accuracy;
2. Posing Dirichlet boundary conditions on (6) removes the solvability condition constraint that appears in the case of Poisson problems with homogeneous Neumann boundary conditions;
3. As more particles are added to a simulation, the Poisson solver converges more rapidly since the solution is specified at more points that are distributed throughout the domain. As a consequence, the condition number of the discrete Laplace operator is decreased because the low frequency modes are suppressed. Strongly inhomogeneous particle distributions may affect this property;
4. If a particle is positioned so that there exists a single grid cell jutting out from the rest of the cage, the zero-normal-gradient is applied to five out of six cell faces, which causes difficulties in the solution of the Poisson problem so much so that it may fail to converge; switching to Dirichlet boundary conditions removes this difficulty.

In general, utilizing the analytical representation (2b) provided by Lamb's solution improves the efficiency of Physalis by ensuring better consistency with the Navier–Stokes equations. For specific details regarding these boundary conditions, see Appendix A.4. In particular, as explained there, it is not precisely (2b) that we must apply as a boundary condition since (6) involves the pressure correction ϕ rather than the pressure itself.

3.2. Conjugate gradient solver

The most significant algorithmic change involves the pressure-Poisson solver. Previously, Z&P used a fast Fourier solver to find ϕ, but a fast solver cannot recognize internal boundary conditions. To overcome this difficulty, Z&P used an iterative deferred correction scheme for every sub-time-step (Lamb's coefficient) iteration. To achieve a complete decoupling between the interior and exterior of the particles, we have adopted a Jacobi-preconditioned conjugate gradient solver in place of the fast Fourier solver. As a consequence we also ensure that our Physalis boundary conditions (2) are more accurately satisfied.

3.3. Explicit numerics

Appendix A.4 describes the fractional step method used to solve (1). Here, we discuss only the rationale behind forgoing the more widely used implicit treatment of the diffusive term $\left(\nu\nabla^2\mathbf{U}\right)$ in favor of an explicit method.

Treating the diffusive term implicitly (e.g., using a Crank–Nicolson method) removes the diffusive stability requirement so that only the convective stability requirement (CFL condition) remains, resulting in an upper bound for the time step given by

$$\Delta t_I < \frac{\Delta x}{\max|u|},$$

written here in one dimension for simplicity; Δx is the discretization length. The stability limit on an explicit method is instead

$$\Delta t_E < \frac{1}{\dfrac{\max|u|}{\Delta x} + \dfrac{2\nu}{(\Delta x)^2}} = \frac{\Delta t_I}{1 + 2\mathrm{Re}_c^{-1}},$$

where

$$\mathrm{Re}_c = \frac{\max|u|\,\Delta x}{\nu} = \frac{\Delta x}{2a}\mathrm{Re}_p, \tag{7}$$

is the cell Reynolds number and $\mathrm{Re}_p = 2\max|u|a/\nu$ is the particle Reynolds number. If Re_c is of order one or greater, the difference between implicit and explicit bounds is small, which suggests that the explicit method becomes more efficient because it avoids the solution of a Helmholtz equation. In our simulation, typically $a/\Delta x \approx 8$ so that we find $\mathrm{Re}_c \approx 1$ when $\mathrm{Re}_p \approx 20$. We have verified that the expectation suggested by this argument is supported by numerical evidence for Re_p greater than this value.

Strictly, in terms of time step size, the explicit method becomes less competitive for smaller Re_p. However, this disadvantage is at least partially compensated by the increased rate of convergence of Lamb's coefficient iterations as the time step is reduced. Further, the explicit method avoids the splitting error of order $O\left[(\Delta t)^2/\mathrm{Re}\right]$ for velocity associated with the implicit fractional step method, ensuring accuracy in unsteady simulations at all nonvanishing Reynolds numbers. Other methods, designed for solving the Navier–Stokes equations at vanishing Re_p, may prove more efficient, but the explicit method adopted here guarantees both stable and accurate time advancement for any nonvanishing Reynolds number, limited only by particle resolution.

3.4. Adaptive time stepping

By adapting the time step size to the time-dependent flow conditions, we proceed through the simulation significantly more efficiently than if we were restricted to choosing the largest stable time step size for the entire duration of the simulation. At the end of every time step, we search the velocity field for the largest value in each of the three velocity components $\max|u_i|$, where the subscript i indicates $\{x, y, z\}$. We then select the next time step abiding by the convective (CFL) and diffusive stability criterion according to [25]:

$$\Delta t = \frac{C}{\sum_i \left[\dfrac{\max|u_i|}{\Delta x_i} + \dfrac{2\nu}{(\Delta x_i)^2}\right]}, \tag{8}$$

where $C < 1$ is a scaling constant similar to a CFL number and the Δx_i are the finite difference grid lengths in each direction, which, in principle, can be different though they are taken equal in all examples in this paper. In practice, we find that using $C \approx 0.5$ tends to best balance numerical accuracy and computational efficiency. As explained in Appendix A.4, Physalis uses a variable-time Adams–Bashforth method to step from time t^n to $t^{n+1} = t^n + \Delta t$.

Even though the flow solver by itself achieves second-order accuracy in time as expected and verified in Section 5.1, it should be recognized that its combination with the particles is not fully second-order accurate. Indeed, as shown in Appendix A.7, the force acting on the particle at time t^{n+1} is calculated using the newly-updated flow at t^{n+1} but maintaining the particle at the position that it occupied at time t^n. The same comment applies to the work reported in Z&P.

4. Particle interaction modeling

As particles approach each other, the number of fluid nodes separating them decreases until contact. When, as in Z&P, the cages were inside the particles, there was the possibility that fluid nodes requiring the solution of the Navier–Stokes equations remained in the gap between the two cages even with particles in contact. These geometries caused convergence difficulties for the pressure-Poisson problem. The current cage construction technique guarantees that any node requiring the solution of the Navier–Stokes equations lies in the fluid. This significantly stabilizes particle collisions. The cages of contacting particles may now abut but, due to continuity, they will carry very similar boundary conditions so that differences between these boundary conditions are inconsequential and such cases need not be handled in any special way.

When two particles α and β, with centers at $\mathbf{X}_\alpha$ and $\mathbf{X}_\beta$, approach each other, the gap between them

$$h = \left|\mathbf{X}_\alpha - \mathbf{X}_\beta\right| - a_\alpha - a_\beta, \tag{9}$$

shrinks to zero. Therefore, at some point, as h approaches Δx, the Navier–Stokes solution on the Cartesian grid fails to capture the physics of the interaction. Furthermore, the convergence rate of the series representing Lamb's solution deteriorates when h is much less than a [27]. For the interactions when $0 < h < \varepsilon$ we use a lubrication model and a soft-sphere contact model for $h < 0$. We add the forces derived from the lubrication model to those from Lamb's coefficients; the latter capture accurately the contribution of the fluid stress away from the contact region and, although the contribution of the contact region is incorrectly evaluated, the interaction force is dominant so that the resulting error is negligible. We have not explored the effects of the choice of ε in detail, but our experience shows that the results are little affected provided $\frac{1}{2}a < \varepsilon < a$. This work represents the first Physalis implementation to include a physically-accurate particle interaction model.

4.1. Short-range: $0 < h < \varepsilon$

When $0 < h < \varepsilon$, we supplement the force from Lamb's coefficients with a lubrication approximation for two approaching spheres. There are a few physical effects to consider:

1. The normal force $\mathbf{F}_n$ necessary to squeeze fluid from the gap between the two particles;
2. The tangential force $\mathbf{F}_t$ and couple $\mathbf{L}_b$ resulting from rotational motion;
3. The tangential force $\mathbf{F}_t$ and couple $\mathbf{L}_b$ resulting from tangential motion.

In terms of the unit vector connecting the particle centers

$$\hat{\mathbf{n}} = \frac{\mathbf{X}_\alpha - \mathbf{X}_\beta}{\left|\mathbf{X}_\alpha - \mathbf{X}_\beta\right|}, \tag{10}$$

we can write the normal and tangential components of the lubrication model force and couple to $O\left(\ln \varepsilon/h\right)$ accuracy as [16,22,28]

$$\mathbf{F}_n = -6\pi\mu a_\alpha \left[\frac{a_{\beta\alpha}^2}{\left(1 + a_{\beta\alpha}\right)^2} \left(\frac{a_\alpha}{h} - \frac{a_\alpha}{\varepsilon} \right) + \frac{a_{\beta\alpha}\left(1 + 7a_{\beta\alpha} + a_{\beta\alpha}^2\right)}{5\left(1 + a_{\beta\alpha}\right)^3} \ln\left(\frac{\varepsilon}{h}\right) \right] \left(\mathbf{w}_{\alpha\beta} \cdot \hat{\mathbf{n}}\right)\hat{\mathbf{n}}, \tag{11a}$$

$$\begin{aligned} \mathbf{F}_t = -6\pi\mu a_\alpha \Bigg[& \frac{4a_{\beta\alpha}\left(2 + a_{\beta\alpha} + 2a_{\beta\alpha}^2\right)}{15\left(1 + a_{\beta\alpha}\right)^3} \left[\mathbf{w}_{\alpha\beta} - \left(\mathbf{w}_{\alpha\beta} \cdot \hat{\mathbf{n}}\right)\hat{\mathbf{n}}\right] \\ & - \frac{2a_{\beta\alpha}^2}{15\left(1 + a_{\beta\alpha}\right)^2} \left(\boldsymbol{\Omega}_{\alpha\beta} + 4a_{\beta\alpha}^{-1}\boldsymbol{\Omega}_\alpha + 4a_{\beta\alpha}\boldsymbol{\Omega}_\beta\right) \times \hat{\mathbf{n}} \Bigg] \ln\left(\frac{\varepsilon}{h}\right), \end{aligned} \tag{11b}$$

$$\begin{aligned} \mathbf{L}_b = -8\pi\mu a_\alpha^2 \Bigg\{ & a_\alpha \frac{2a_{\beta\alpha}}{5\left(1 + a_{\beta\alpha}\right)} \left[\left(\boldsymbol{\Omega}_\alpha + \frac{1}{4}a_{\beta\alpha}\boldsymbol{\Omega}_\beta\right) - \left(\boldsymbol{\Omega}_\alpha + \frac{1}{4}a_{\beta\alpha}\boldsymbol{\Omega}_\beta\right) \cdot \hat{\mathbf{n}}\hat{\mathbf{n}} \right] \\ & + \frac{a_{\beta\alpha}\left(4 + a_{\beta\alpha}\right)}{10\left(1 + a_{\beta\alpha}\right)^2} \mathbf{w}_{\alpha\beta} \times \hat{\mathbf{n}} \Bigg\} \ln\left(\frac{\varepsilon}{h}\right), \end{aligned} \tag{11c}$$

where $a_{\beta\alpha} = a_\beta/a_\alpha$, $\mathbf{w}_{\alpha\beta} = \mathbf{w}_\alpha - \mathbf{w}_\beta$, and $\boldsymbol{\Omega}_{\alpha\beta} = \boldsymbol{\Omega}_\alpha - \boldsymbol{\Omega}_\beta$. The subtraction of the term a_α/ε in (11a) ensures that the influence of the short-range lubrication force smoothly increases from zero as h/ε decreases from one. All terms except the first one in (11a) are proportional to $\ln\left(\varepsilon/h\right)$ only. These higher-order terms are valuable in recovering the appropriate physical behavior for particles in close proximity by accounting for shear and rotational motion. We apply this theory also to the interaction of particles with a no-slip domain boundaries by letting $a_\beta \to \infty$.

4.2. Contact: $h < 0$

While the hydrodynamic forces and lubrication model together slow the velocity of an oncoming particle, sufficiently energetic interactions will still lead to contact because capturing the full effect of the repulsive force singularity would require infinitesimally small time steps. Therefore, we do expect to encounter cases in which $h < 0$. To deal with these situations we adopt a soft particle contact force model inspired by Tsuji et al. [29] and Barnocky and Davis [30] in which the normal component of the contact force is given by

$$\mathbf{F}_n = \left[-k_n\left(-h\right)^{3/2} - \eta_n \mathbf{w}_{\alpha\beta} \cdot \hat{\mathbf{n}}\right]\hat{\mathbf{n}}, \tag{12}$$

where k_n is the Hertz elastic contact constant for perfectly elastic spheres expressed in terms of Poisson's ratio σ and Young's modulus E by

$$k_n = \frac{4}{3}\left(\frac{1 - \sigma_\alpha^2}{E_\alpha} + \frac{1 - \sigma_\beta^2}{E_\beta}\right)^{-1}\left(\frac{a_\alpha a_\beta}{a_\alpha + a_\beta}\right)^{1/2}. \tag{13}$$

The damping term η_n is nonlinear in h and we choose its value by combining the work of Tsuji et al. [29] and Barnocky and Davis [30]. In Tsuji et al. [29],

$$\eta_n = \alpha\left(e\right)\sqrt{mk_n}\left(-h\right)^{1/4}, \tag{14}$$

where $m = m_\alpha m_\beta / \left(m_\alpha + m_\beta\right)$ is the reduced mass and α is directly related to the coefficient of restitution e, defined as the ratio of the normal particle velocity immediately after and before contact, via a nonlinear differential equation. We have found that the fit

$$\alpha\left(e\right) = 2.22 - 2.26e^{0.4}, \tag{15}$$

closely represents the solution of this differential equation. The proper value to be used for e in this relation must include the effect of the interstitial fluid. Experimental evidence [31,20] shows that the relationship between the dry and wet coefficients of restitution, e_{dry} and e_{wet}, depends on the Stokes number just before contact

$$\text{St}_0 = \frac{1}{9}\text{Re}_{p0}\frac{\rho_p}{\rho_f}, \tag{16}$$

with particle Reynolds number Re_{p0} based on velocity just before contact. This relationship as given by Barnocky and Davis [30] is

$$e_{\text{wet}} = e_{\text{dry}} + \frac{1 + e_{\text{dry}}}{\text{St}_0}\ln\frac{l_r}{h_0}, \tag{17}$$

where l_r is a measure of the particle surface roughness length and h_0 is the separation distance at the time step just before contact. The minimum value of e_{wet} allowed is zero. From the value of St_0 computed at the last time step before contact, we calculate e_{wet} using (17) and then α from (15). We then hold α constant in the damping coefficient (14) for the duration of the contact. We apply this theory to contact between particles and no-slip domain boundaries by letting $a_\beta \to \infty$. Further discussion of this method is the topic of a forthcoming publication currently in preparation.

5. Validation and benchmarking

We have implemented the methods described above in an open-source code designed for GPU-centric operation on CUDA-enabled GPUs [32], and we have found it to run more than 90 times faster than the legacy code of Z&P and G&P. In terms of accuracy, we find comparable results but with an increased convergence rate of Lamb's coefficient iterations and smaller particle force fluctuations.

In Section 5.1 we use a two-dimensional Taylor–Green vortex to prove that the background flow solver attains second-order accuracy. Section 5.2 returns to the work of G&P to illustrate the similarities and differences between the current and old methods in two key validation studies. In Section 5.3 we produce a comparison to the experimental work of Mordant and Pinton [19] to serve as a benchmark of Physalis in a more dynamic setting. Finally, Section 5.4 benchmarks the Physalis lubrication and contact models using the experiments of Joseph et al. [20].

5.1. Flow solver validation

To illustrate the second-order convergence of the flow solver, we consider the two-dimensional Taylor–Green vortex [33] that satisfies (1), namely,

$$u\left(x,y,t\right) = -\cos\frac{2\pi x}{L_x}\sin\frac{2\pi y}{L_y}e^{-2t/\tau}, \tag{18a}$$

$$v\left(x,y,t\right) = \sin\frac{2\pi x}{L_x}\cos\frac{2\pi y}{L_y}e^{-2t/\tau}, \tag{18b}$$

$$p\left(x,y,t\right) = -\frac{1}{4}\left(\cos\frac{4\pi x}{L_x} + \cos\frac{4\pi y}{L_y}\right)e^{-4t/\tau}, \tag{18c}$$

where $\tau = L_x L_y / \left(4\pi^2 \nu\right)$ on the domain $0 \leq x/L_x, y/L_y \leq 1$. Table 2 summarizes the results of the study of a comparison of the error and convergence rate at time $t/\tau = 1$ in the manner described in Brown et al. [26] using constant $C = 0.1$ for the determination of the time step using (8) for grids with resolution 32 × 32, 64 × 64, and 128 × 128. We confirm that the flow solver achieves second-order convergence in time and space as expected.

Table 2
Error and convergence rates for Taylor–Green vortex flow solver validation test.

Variable	L_1-error			Rate
	32×32	64×64	128×128	
u	3.199×10^{-4}	4.006×10^{-5}	5.013×10^{-6}	2.00
v	3.199×10^{-4}	4.006×10^{-5}	5.013×10^{-6}	2.00
p	1.018×10^{-6}	1.296×10^{-7}	1.629×10^{-8}	1.99

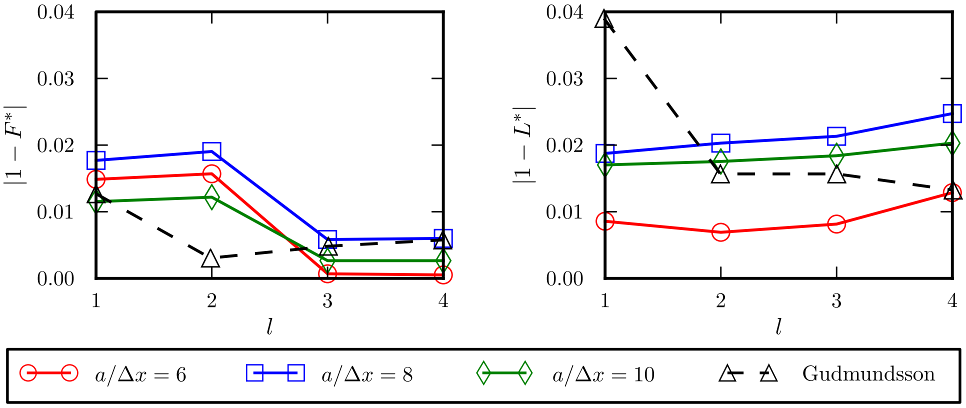


Fig. 3. Results of the momentum conservation tests in G&P with $r/a = 1.25$, $P^* = 10$, and $\mathrm{Re}_\Omega = 20$. The tests of G&P use $a/\Delta x = 8$.

5.2. Physalis validation

To validate the numerous changes in the present work, we reproduce two tests of G&P and compare the accuracy of the two methods. With the exception of the scalar product Lamb's coefficient calculations, the methods used in G&P mirror those of Z&P (in fact, they use the same code). By comparing against G&P, we effectively show the differences between the accuracy of the current work and its predecessors.

The first test considers the momentum conservation for the flow about a sphere in a cubic lattice by simulating a single particle located at the center of a triply-periodic domain. As explained in G&P, in the presence of a pressure gradient

$$\nabla p = P\hat{\mathbf{k}} + \nabla \hat{p}, \tag{19}$$

in which $\hat{p}$ is periodic and P is a constant, consideration of the momentum balance at steady state yields

$$F^* \equiv \frac{F_z}{(1-\beta)\, L^3 P} = 1, \tag{20}$$

where F_z is the force on the sphere, $\beta = \frac{4}{3}\pi a^3 L^{-3}$, and $L = 4a$ is the size of the domain. We reproduce a test in which we vary the truncation order l and particle resolution $a/\Delta x$ using precisely the same parameters as in G&P, namely, $P^* = a^3 P \mu^{-1} \nu^{-1} = 10$, and $r/a = 1.25$ and plot the results in Fig. 3.

The second test considers a similar situation, only now the sphere spins with a constant applied couple $L_{p,z}$. The work of G&P shows that, at steady state, the couple on the particle $L_{p,z}$ should equal the total hydrodynamic couple acting on the surface S of the simulation domain given by

$$L_{d,z} = \int_S d\mathbf{S}\, \mathbf{x} \times (\boldsymbol{\sigma} \cdot \mathbf{n})\,, \tag{21}$$

where $\sigma_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)$ is the stress tensor, so that the ratio $L^* \equiv L_{d,z}/L_{p,z}$ equals one. We plot the results in Fig. 3 using the same simulation parameters as above and a rotation rate $\mathrm{Re}_\Omega = a^2 \Omega_z \nu^{-1} = 20$. For both cases, the results of G&P were generated using $a/\Delta x = 8$.

Judging the results illustrated in Fig. 3, we can make some broad generalizations. For truncation of Lamb's series solutions up to $l = 1$ and $l = 2$, the current work exhibits approximately 50% larger error than G&P for the linear momentum conservation test, but consistently beats G&P by more than 10% for the angular momentum conservation test. For $l = 3$, however, with all $a/\Delta x$ tested for both linear and angular momentum conservation, the current work performs better than G&P. For $l = 4$, we see good results for both tests, which are essentially indistinguishable from those for $l = 3$.

The lack of monotonicity with increasing $a/\Delta x$ can be understood by noting that the position of the nodes used to interpolate the field values on the scalar-product integration surface will change with $a/\Delta x$. It may well happen that, for a

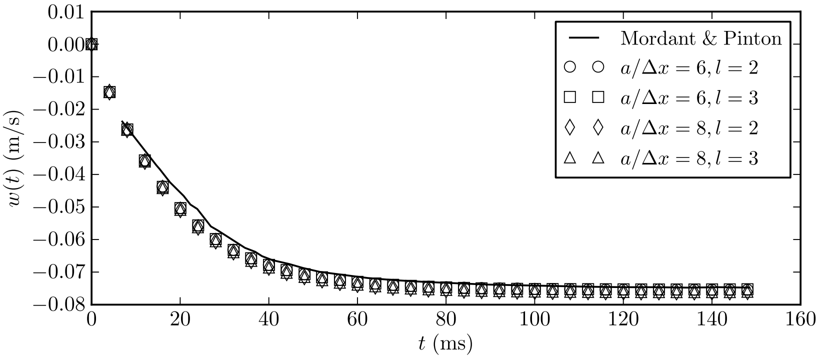


Fig. 4. Benchmarking Physalis using Mordant and Pinton [19], Case #1. Note that we have removed $t \lesssim 5$ ms in the experimental data due to a deficiency in the experimental method over this time period explained by Mordant and Pinton [19].

Table 3
The results of our benchmarking study for the sedimentation of a single particle in quiescent flow. The experimental data is from Mordant and Pinton [19], Case #1.

Symbol	$a/\Delta x$	l	V_1	% error
—	Experimental		0.0741	0.6
○ ○	6	2	0.0746	0.675
□ □	6	3	0.0745	0.580
◇ ◇	8	2	0.0759	2.43
△ △	8	3	0.0759	2.43

particular particle position, the mutual relationship between the integration surface and the nodes is more favorable for one $a/\Delta x$ than for others. This indeed appears to be the case for the three values of $a/\Delta x$ considered here. For both $a/\Delta x = 6$ and $a/\Delta x = 10$, the integration surface generally cuts through the center of the Cartesian grid cells while it aligns with the cell edges for $a/\Delta x = 8$, representing the two extreme possibilities. It appears that the former pair of $a/\Delta x$ may be more directly compared while the latter takes on a different character. These considerations illustrate the connection between the position of the quadrature points, which is itself dependent on the radius of the cage, and the underlying Cartesian grid. This is evidently a matter of some complexity, which must be left for future work.

For additional commentary regarding particle resolution, refer to Section 5.3, which shows that larger values of $a/\Delta x$ decrease force fluctuations encountered when particles move with respect to the grid. Also, as explained in Section 5.4, larger values of $a/\Delta x$ are also beneficial in the case of particle collisions.

5.3. Physalis benchmark: sedimentation

The work of Mordant and Pinton [19] presents high-quality data from experiments involving solid spheres of various sizes and densities sedimenting from rest in a quiescent fluid. Though much of the data is for high particle Reynolds numbers that would require very large grids, one experiment, for which $\mathrm{Re}_p = 41$, is an excellent benchmark against which to test the performance of Physalis for a moving particle.

Case #1 in Mordant and Pinton [19] provides data for velocity as a function of time for the sedimentation of a glass sphere of radius $a = 0.25$ mm and density $\rho_p = 2560$ kg/m^3 in water at 25 °C. Using an acoustic Doppler shift method, Mordant and Pinton [19] reports a terminal velocity $V_1 = 0.0741$ m/s ($\mathrm{Re}_p = 41$) and variance of 0.4 mm/s, which corresponds to an experimental uncertainty of 0.6%. The particle Galileo number is $\mathrm{Ga} = \nu^{-1}\left(\left|\rho_p/\rho_f - 1\right| 8a^3 g\right)^{1/2} = 49$.

To simulate this experiment we use a domain with cross section $20a \times 20a$ with no-slip walls and height $48a$ with Neumann boundary conditions on the top and bottom; the results with Neumann lateral walls were the same. We perform simulations using $r/a = 1.2$ for the integration surface and $C = 0.5$ for the CFL number, with combinations of $a/\Delta x$ and l as summarized in Table 3. A plot of velocity as a function of time for each validation test may be found in Fig. 4 (we have removed $t \lesssim 5$ ms in the experimental data due to a deficiency in the experimental method over this time period explained in Mordant and Pinton [19]).

Analyzing Fig. 4 in detail, we see that all simulations produce satisfactory results while consistently, if slightly, overestimating the falling velocity (in magnitude). For all simulations, the settling velocity in the time range of approximately $10 < t < 50$ differs from the experimental result up to nearly 10%, but then converges to within 2.5% accuracy once terminal velocity has been reached. The work of Yang and Stern [9] summarizes the results of several other numerical investigations all of which exhibit differences from the data similar in magnitude and direction as our own.

Table 3 quantifies the accuracy of the various simulations at terminal velocity where we can see that the higher resolutions produce an error at terminal velocity of 2.43% while the lower resolution simulations achieve results of 0.675% and

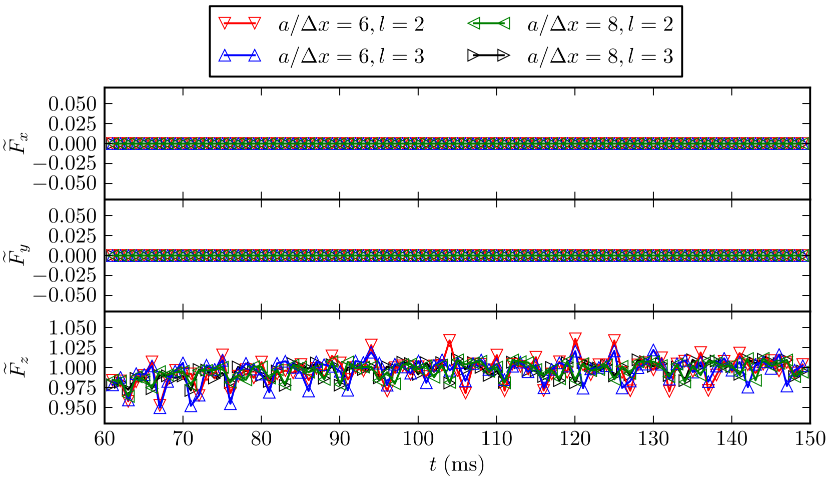


Fig. 5. Force fluctuations in the Physalis benchmarking results for Mordant and Pinton [19], Case #1.

0.580%, which are very near the uncertainty associated with the experiments (0.6%). Thus, as already found in Section 5.2, the simulations with lower resolution ($a/\Delta x = 6$) produce results better aligned with experiment than those with higher resolution ($a/\Delta x = 8$). However, in contrast to the results of Section 5.2, the differences between using truncation order $l = 2$ and $l = 3$ appear to be quite minor.

In addition to the settling velocity, we consider in Fig. 5 the three components of the hydrodynamic force on the particle normalized by the effective buoyancy force as:

$$\widetilde{\mathbf{F}} = \frac{\mathbf{F}}{\frac{4}{3}\pi a^3 \left(\rho_p - \rho_f\right) g}. \tag{22}$$

Here, we can see how the motion of the particle against the fixed background grid (and the changing shape of the cage) affect the force fluctuations. The plots of $\widetilde{F}_x$ and $\widetilde{F}_y$ show that the fluctuations remain very small ($O\left(10^{-14}\right)$) and that $\widetilde{F}_z$ decreases from about 4% of the net gravitational force for $a/\Delta x = 6$ to 1% for $a/\Delta x = 8$. The results shown in Z&P for a particle sedimenting in a duct at $\mathrm{Re}_p \approx 22$ exhibit force fluctuations on the order of 4% for $a/\Delta x = 8$. Even though the Reynolds numbers of the two simulations differ (a circumstance which might be expected to reduce the fluctuations as it enlarges the Stokes region), the comparison suggests that the force fluctuations in the present calculations are smaller, most likely because of the changes made to the cage generation described in Section 2.2, which enable the cage to change more smoothly as the particle moves.

5.4. Physalis benchmark: interactions

We use the work of Joseph et al. [20] to benchmark the lubrication and contact models described in Section 4. This experimental study explores particle–wall interactions by colliding spheres of various size and density into a wall in a pendulum arrangement, measuring the particle velocity before and after contact. The results are given in the form of a plot of wet coefficient of restitution e_{wet} as a function of Stokes number St_0 as defined in (16); we illustrate the experimental results along with our numerical results in Fig. 6.

To simulate these experiments, we accelerate the particle normally to the wall with a linear elastic force with stiffness k. We reproduce a subset of the experiment for a particle with $a = 1.5$ mm and material properties of glass: $\rho_p = 2540$ kg/m^3, $E = 0.6$ MPa, and $\sigma = 0.23$. We take the values $e_{\mathrm{dry}} = 0.98$ and $l_r/h_0 = 10^{-3}$ as reported in Joseph et al. [20]. The fluid is water at 25 °C. We use a domain of size $8a \times 8a \times 8a$ with a no-slip wall made of the same material as the particle, a Neumann condition on the opposing domain boundary, and periodic conditions elsewhere. The particle is released from a distance of 8.5 mm from the wall and we vary the driving spring stiffness k to change St_0. We use simulation parameters $r/a = 1.2$, $l = 3$, and $C = 0.5$ for all cases. Since particle velocity increases with St_0—and collision intensity along with it—we must resolve particles with a finer mesh (increased $a/\Delta x$) in order to capture the collisions for larger St_0. We found that too coarse a discretization, i.e. too small a value of $a/\Delta x$, terminates the simulation upon particle contact with the wall. We determined the minimum required particle resolution by progressively increasing $a/\Delta x$; we show the results together with Re_{p0} and coefficient of restitution e_{wet} in Table 4. To calculate St_0 and e_{wet} from each simulation, we follow the same procedures as Joseph et al. [20], which involve estimating the velocity u_i before and u_o after contact and calculating the ratio $e_{\mathrm{wet}} = -u_o/u_i$.

We compare the results in Fig. 6. Note that the value we have used for Young's modulus is five orders of magnitude smaller than the true value for glass because resolving the true stiffness requires Δt far smaller than may be reasonably simulated, and the work of Joseph et al. [20] concludes that varying E appears to have no effect on the results as suggested

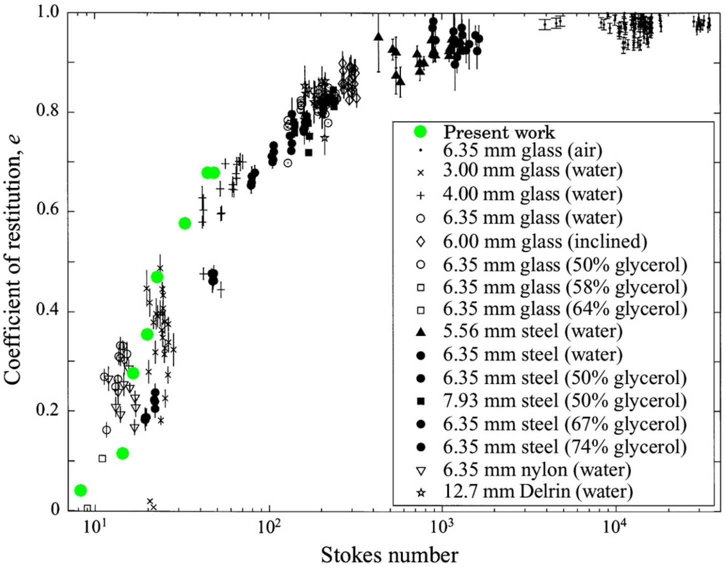


Fig. 6. Coefficient of restitution e as a function of St_0 for collisions of many different particles adapted from Joseph et al. [20] supplemented with the results of the current Physalis simulations.

Table 4
A summary of the particle–wall collision simulations, where k is the linear spring constant that drives the collision and $\mathrm{Re}_{p,\max}$ is the particle Reynolds number at maximum approach velocity. Indices zero and one denote conditions at the last time step prior to contact and at the first time step after contact, respectively. The last two columns show the good consistency between the *a priori* value of the restitution coefficient given by (17) and the value obtained from the numerical simulation.

$a/\Delta x$	k (N/m)	$\mathrm{Re}_{p,\max}$	Re_{p0}	St_0	Re_{p1}	e_{wet}	
						a priori	simulation
10	0.5	58.5	29.5	8.35	1.45	0	0.0491
12	1.0	88.3	54.6	15.5	6.33	0.0976	0.116
12	1.2	98.2	62.7	17.8	17.8	0.212	0.284
12	1.5	112	76.1	21.5	27.3	0.344	0.359
14	2.0	125	92.0	26.0	43.3	0.464	0.471
16	3.0	166	123	34.6	71.5	0.585	0.585
16	4.0	195	149	42.0	102	0.654	0.685
16	5.0	221	174	49.1	119	0.701	0.685

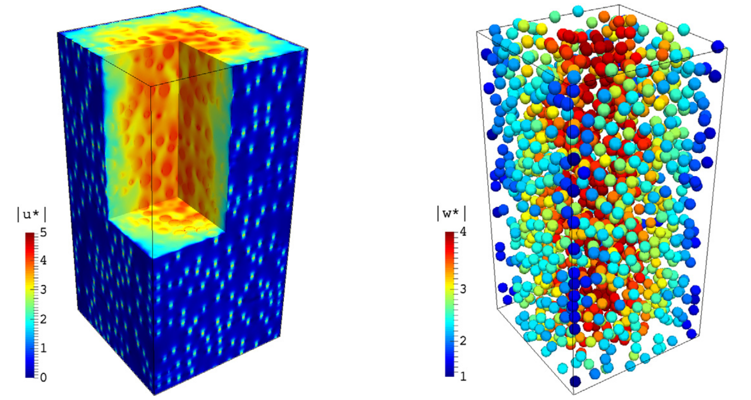
by the favorable comparison in Fig. 6. This value of E does, however, affect the duration of the contact as discussed in Kempe and Fröhlich [34]. From Fig. 6 we see that the lubrication and contact models described in Section 4 perform well. A more revealing comparison is provided by the last two columns of Table 4, which compare the *a priori* values of the restitution coefficient given by (17) with the results of the simulation. It is seen here that the model used for the damping coefficient η_n gives results consistent with the *a priori* value of the restitution coefficient.

6. 2048 particles sedimenting in a duct

To further illustrate the capabilities of Physalis, we briefly present a sizeable simulation involving 2048 particles sedimenting in a periodic duct with an average volume fraction of 20%. The particles and fluid use the properties of Section 5.3, which provides the single-particle terminal velocity $V_1 = 0.0741$ m/s and characteristic time $\tau_{95} = 55$ ms (see Mordant and Pinton [19]) with which to normalize the fluid and particle sedimentation velocities produced by the simulation as $u^* = u/V_1$ and $w^* = w/V_1$, respectively, as well as the time $t^* = t/\tau_{95}$. We use simulation parameters $a/\Delta x = 8$, $l = 3$, and CFL constant in (8) $C = 0.5$. The domain size is $28a \times 28a \times 56a$ with four no-slip lateral walls and periodic boundary conditions in the longest (gravity) direction. The particles have a Galileo number $\mathrm{Ga} = \nu^{-1}\left(\left|\rho_p/\rho_f - 1\right| 8a^3 g\right)^{1/2} = 49$.

Fig. 7 illustrates the magnitude of the normalized fluid and particle velocities, $|u^*|$ and $|w^*|$, respectively, at $t^* = 2.4$ and Fig. 8 shows $\langle j^* \rangle_z$, the volumetric flux averaged over the gravity direction, at $t^* = 3.5$. In Fig. 9, we plot the mean absolute (i.e. not relative to the fluid) velocity $\overline{|w^*|}$ plus/minus the standard deviation σ_{w^*}, and maximum/minimum particle velocities $|w^*|_{M/m}$ as functions of time. In Fig. 10 we plot the vertically-averaged particle volume fraction $\langle \beta \rangle_z$ over the cross section of the duct at times $t^* = 0$ and $t^* = 3.5$.

Even though, due to periodicity, the number of particles is effectively infinite, their velocity does not continue to increase because of the dissipation caused by the walls. However, we notice in Fig. 10 the accumulation of particles along the walls,



(a) Visualization of the fluid velocity. The particles are not shown for clarity.

(b) Visualization of the particle velocity. The fluid is not shown for clarity.

Fig. 7. 2048 particles sedimenting under gravity in a periodic duct at $t^* = t/\tau_{95} = 3.5$. Here, $|u^*|$ and $|w^*|$ are the fluid and the particle velocities normalized by the single-particle terminal velocity as determined in Section 5.3.

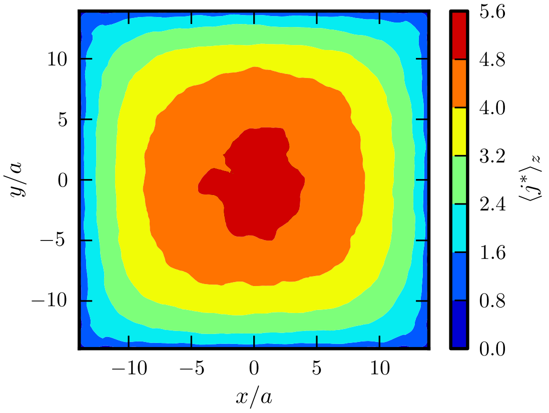


Fig. 8. The volumetric flux averaged over the gravitational direction at $t^* = 3.5$.

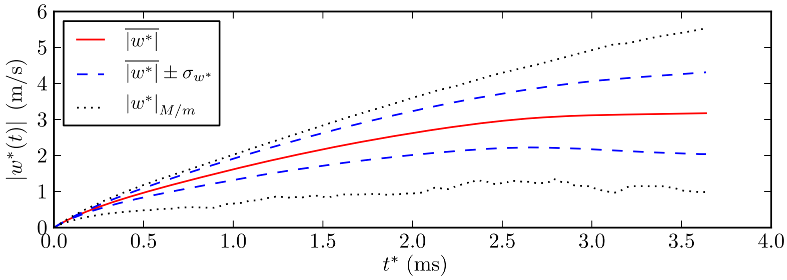
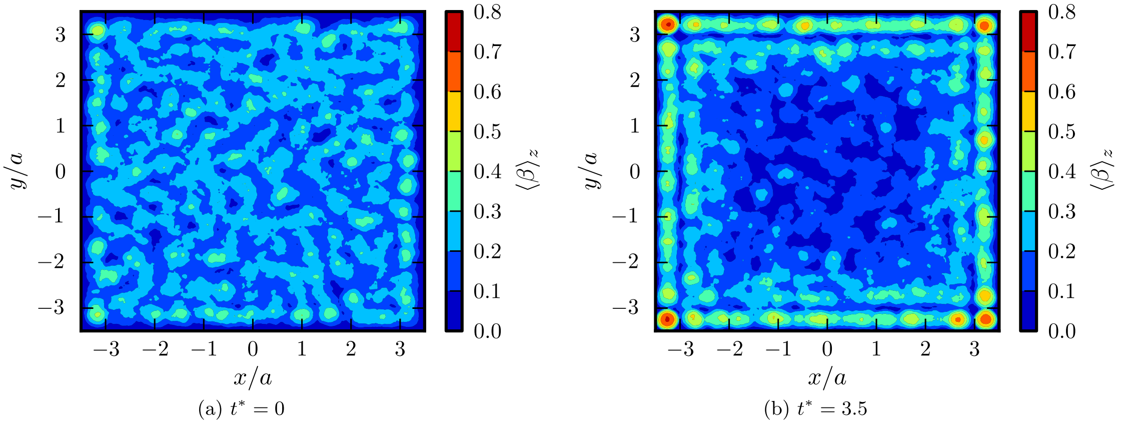


Fig. 9. The particle velocity $w^* = w/V_1$ over time with mean $\overline{|w^*|}$ plus/minus standard deviation σ_{w^*}, and maximum/minimum $|w^*|_{M/m}$.

which is reminiscent of phenomena shown in Shaffer and Gopalan [35]. This phenomenon is explained by the fact that, when a fast particle in the core of the duct collides with a slower-moving particle and pushes it toward the wall, the effective restitution coefficient is too weak to push the particle back into a region of significant flow velocity. It is somewhat surprising that the minimum velocity shown in Fig. 9, which is typically due to particles falling near the walls, is as large as the single-particle terminal velocity. This effect is due the strong shear caused by the flow in the core of the duct, where we observe particle velocities as large as five times the single-particle terminal velocity.

This simulation was carried out on a single Nvidia Tesla K40 GPU with an Intel Xeon E5-2650v2 CPU, and it required about four weeks of computation time for over 10,500 time steps averaging 12 Lamb's coefficient iterations per time step. By comparing the run time of the current implementation with that of the previous implementation, we find an improvement of nearly 90 times. An in-depth discussion of the numerical implementation—including a clear quantification of the relative



(a) $t^* = 0$

(b) $t^* = 3.5$

Fig. 10. Averages of particle volume fraction $\langle\beta\rangle_z$ over the gravitational direction.

efficiency of the new implementation—is the subject of [36]. We attempted this simulation also with a coarser discretization, $a/\Delta x = 6$, but at some point Lamb's coefficients failed to converge due to insufficiently resolved collisions.

7. Summary and conclusions

We have described several new developments introduced in the Physalis approach to the simulation of disperse particle flows, focusing on a number of improvements to the methods put forth in Z&P and G&P that together make the current method significantly more efficient than its predecessor. We validated the flow solver by itself using the two-dimensional Taylor–Green vortex to prove that we achieve second-order convergence in both velocity and pressure. We subsequently performed linear and angular momentum conservation tests to validate the numerous improvements to the Physalis method, showing favorable comparisons between the results of the current work and those of G&P. By validating the flow solver and Physalis method, we verified the accuracy of the methods even at relatively low particle resolutions.

We applied Physalis to generate two benchmarks that we compared against experimental data. The first one is the sedimentation of a single particle, for which we obtained results in agreement with those of several other numerical investigations [9], all of which exhibit similar small differences with experiment during the transient phase of the particle motion. The results, however, are within experimental error once terminal velocity is reached. In this test, we showed that the current Physalis methodology reduces by more than 75% the force fluctuations induced by the motion of the particle with respect to the background fixed discrete grid when compared to Z&P. The second benchmark concerns the collision of a particle with a solid wall for which we have found a very favorable comparison with the experimental results of Joseph et al. [20]. To conclude, we provided a brief example of the simulation of 2048 particles sedimenting in a periodic duct.

We include in the appendix a complete and self-sufficient document detailing the analytical background and numerical methods behind the current work. Additional details, as well as access to the open-source code, can be found at http://PhysalisCFD.org.

Acknowledgements

The first author sincerely thanks Dr. Claire VerHulst for her insights during the development of this work as well as Dr. Laura Lukassen, Yayun Wang, and Daniel Willen for their assistance in testing the code.

The present work has been supported by the United States National Science Foundation under grants CBET1258398 and CBET1335965. A.J.S. also acknowledges support from the Johns Hopkins University Modeling Complex Systems IGERT Fellowship Program, under National Science Foundation grant DGE0801471.

Appendix A. The Physalis method

This appendix contains several details omitted in the synthetic description of the method presented above in Sections 2 and 3. As Physalis has evolved in a substantial way since Z&P, we review the entire method here to provide a complete picture and to introduce a consistent notation.

A.1. Transformation to particle rest frame

The first step is to introduce the velocity $\mathbf{u}$ in the reference frame of a particle by writing

$$\mathbf{U} = \mathbf{u} + \mathbf{w} + \boldsymbol{\Omega} \times \mathbf{r}, \tag{A.1}$$

where $\mathbf{w}$ and $\boldsymbol{\Omega}$ are the particle linear and angular velocity, respectively, and $\mathbf{r}=(r,\theta,\varphi)$ is a position vector with respect to the center of the particle. Furthermore, we let

$$\mathbf{u}=\tilde{\mathbf{u}}+\frac{1}{10\nu}\left(\frac{r^5-a^5}{r^3}\right)\dot{\boldsymbol{\Omega}}\times\mathbf{r},\quad p=\tilde{p}+\frac{1}{2}\rho_f\left(\boldsymbol{\Omega}\times\mathbf{r}\right)^2+\rho_f\left(\mathbf{g}-\dot{\mathbf{w}}\right)\cdot\mathbf{r}. \tag{A.2}$$

It can be shown that, with these definitions, the Navier–Stokes equations become

$$\frac{\partial\mathbf{u}}{\partial t}+\left(\mathbf{u}\cdot\nabla\right)\mathbf{u}+2\boldsymbol{\Omega}\times\mathbf{u}=-\frac{1}{\rho_f}\nabla\tilde{p}+\nu\nabla^2\tilde{\mathbf{u}}, \tag{A.3}$$

$$\nabla\cdot\tilde{\mathbf{u}}=0. \tag{A.4}$$

Note that, while the left-hand side of (A.3) is in terms of the rest frame fluid velocity $\mathbf{u}$, the right-hand side is in terms of the modified fields $\tilde{\mathbf{u}}$ and $\tilde{p}$. Since, due to the no-slip condition, $\mathbf{u}$ vanishes on the particle surface, the left-hand side of (A.3), due to continuity, will be very small in the immediate neighborhood of the particle surface. Therefore, in a region near $r=a$, we may neglect the left-hand side of (A.3) and write

$$0=-\nabla\tilde{p}+\mu\nabla^2\tilde{\mathbf{u}}, \tag{A.5}$$

where $\mu=\rho_f\nu$ is the dynamic viscosity. Equations (A.5) and (A.4) are recognized as the Stokes equations. It is worth stressing that the region in which (A.5) is a good approximation to (A.3) decreases in size as particle Reynolds number increases, but that (A.5) is always a viable approximation—given sufficient resolution—for any finite Reynolds number. In this regard, the present method is no different from any other.

A.2. Lamb's solution for Stokes flow about a sphere

A solution to (A.5) about a sphere was given by Lamb in the late 19th century [21,22] in a spherical coordinate system. We split the velocity and pressure fields into the incident external flow (superscript e) and a particle-induced perturbation flow (superscript p)

$$\tilde{\mathbf{u}}=\mathbf{u}^e+\mathbf{u}^p,\qquad \tilde{p}=p^e+p^p. \tag{A.6}$$

Lamb's general solution for (A.5) is [21,22]

$$\mathbf{u}^e\left(r,\theta,\varphi\right)=\frac{\nu}{a^2}\sum_{l=0}^{\infty}\frac{1}{\left(l+1\right)\left(2l+3\right)}\left[\frac{1}{2}\left(l+3\right)r^2\nabla p_l-l\mathbf{r}p_l\right]$$
$$+\frac{\nu}{a}\sum_{l=0}^{\infty}\left[a\nabla\phi_l+\nabla\times\left(\mathbf{r}\chi_l\right)\right], \tag{A.7a}$$

$$\mathbf{u}^p\left(r,\theta,\varphi\right)=\frac{\nu}{a^2}\sum_{l=1}^{\infty}\frac{1}{l\left(2l-1\right)}\left[-\frac{1}{2}\left(l-1\right)r^2\nabla p_{-l-1}+\left(l+1\right)\mathbf{r}p_{-l-1}\right]$$
$$+\frac{\nu}{a}\sum_{l=1}^{\infty}\left[a\nabla\phi_{-l-1}+\nabla\times\left(\mathbf{r}\chi_{-l-1}\right)\right], \tag{A.7b}$$

$$p^e\left(r,\theta,\varphi\right)=\frac{\mu\nu}{a^2}\sum_{l=0}^{\infty}p_l, \tag{A.7c}$$

$$p^p\left(r,\theta,\varphi\right)=\frac{\mu\nu}{a^2}\sum_{l=0}^{\infty}p_{-l-1}, \tag{A.7d}$$

where the p_l, ϕ_l, and χ_l are solid harmonics of order l. For example,

$$p_l=\left(\frac{r}{a}\right)^l\sum_{m=-l}^{l}p_{lm}Y_l^m\left(\theta,\varphi\right)=\left(\frac{r}{a}\right)^l\left[p_{l0}^{\Re}N_l^0P_l^0+2\sum_{m=1}^{l}N_l^mP_l^m\left(p_{lm}^{\Re}\cos m\varphi-p_{lm}^{\Im}\sin m\varphi\right)\right], \tag{A.8}$$

where

$$Y_l^m\left(\theta,\varphi\right)=N_l^mP_l^me^{im\varphi};\qquad -l\le m\le l, \tag{A.9}$$

with

$$N_l^m = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}, \tag{A.10}$$

is a surface harmonic of order l.

The time-dependent Lamb's coefficients $p_{lm} = p_{lm}^{\Re} + ip_{lm}^{\Im}$ are complex-valued nondimensional constants that we have broken into real and imaginary parts in the last step of (A.8). The solid harmonics ϕ_l and χ_l are written similarly in terms of coefficients ϕ_{lm} and χ_{lm}. The harmonics of negative order appearing in (A.7b) and (A.7d) can be related to the positive harmonics by using the no-slip condition to find

$$p_{-l-1} = -\frac{2l-1}{2(l+1)}l\left[p_l + 2(2l+1)\phi_l\right]\left(\frac{a}{r}\right)^{2l+1}, \tag{A.11a}$$

$$\phi_{-l-1} = -\frac{a^2}{4}\frac{l}{l+1}\left[\frac{2l+1}{2l+3}p_l + 2(2l-1)\phi_l\right]\left(\frac{a}{r}\right)^{2l+1}, \tag{A.11b}$$

$$\chi_{-l-1} = \left(\frac{a}{r}\right)^{2l+1}\chi_l. \tag{A.11c}$$

The expansions (A.7) converge spectrally. Truncations of the summations with $l_{\max} = 2, 3$, and 4 requires a total of 25, 46, and 73 coefficients, respectively.

A.3. Determination of Lamb's coefficients

As explained in Section 2.1, at every Lamb's coefficient iteration, the coefficients are determined from the currently available Navier–Stokes solution. The various ways in which this objective can be achieved are discussed in G&P. In general, these calculations require the scalar product of spherical vector harmonics with the numerically determined $\tilde{\mathbf{u}}$ and $\tilde{p}$. The scalar products are defined as

$$(f, g) = \int_{-\pi}^{\pi} d\varphi \int_{0}^{\pi} \sin\theta\, d\theta\, \bar{f}(\theta,\varphi)\, g(\theta,\varphi), \tag{A.12}$$

where the overbar denotes complex conjugation. It is concluded in G&P that the following three scalar products are optimal for determining Lamb's coefficients:

$$\frac{a^2}{\nu\mu}\left(Y_l^m, \tilde{p}\right) = \left[1 - \frac{l(2l-1)}{2(l+1)}\left(\frac{a}{r}\right)^{2l+1}\right]\left(\frac{r}{a}\right)^{l} p_{lm} - \frac{l(2l-1)(2l+1)}{l+1}\left(\frac{a}{r}\right)^{l+1}\phi_{lm}, \tag{A.13a}$$

$$\begin{aligned}\frac{a}{\nu}\left(r\nabla Y_l^m, \tilde{\mathbf{u}}\right) = {} & \frac{l}{4}\left\{\frac{2(l+3)}{2l+3} + \left[l-2-\frac{l(2l+1)}{2l+3}\left(\frac{a}{r}\right)^2\right]\left(\frac{a}{r}\right)^{2l+1}\right\}\left(\frac{r}{a}\right)^{l+1} p_{lm} \\ & + l\left\{l+1+\frac{1}{2}\left[(l-2)(2l+1)\left(\frac{r}{a}\right)^2 - l(2l-1)\right]\left(\frac{a}{r}\right)^{2l+1}\right\}\left(\frac{r}{a}\right)^{l-1}\phi_{lm},\end{aligned} \tag{A.13b}$$

$$\frac{a}{\nu}\left(\mathbf{r}\times\nabla Y_l^m, \tilde{\mathbf{u}}\right) = l(l+1)\left[\left(\frac{a}{r}\right)^{2l+1} - 1\right]\left(\frac{r}{a}\right)^{l}\chi_{lm}. \tag{A.13c}$$

These are the scalar products that we have adopted in this work. More explicitly,

$$\left(Y_l^m, \tilde{p}\right) = \left(Y_l^m, \tilde{p}\right)^{\Re} - i\left(Y_l^m, \tilde{p}\right)^{\Im} = \int_{-\pi}^{\pi} d\varphi \int_{0}^{\pi} d\theta\, N_l^m P_l^m \tilde{p}(r,\theta,\varphi)\sin\theta\left[\cos(m\varphi) - i\sin(m\varphi)\right], \tag{A.14a}$$

$$\begin{aligned}\left(r\nabla Y_l^m, \tilde{\mathbf{u}}\right)^{\Re} = {} & \int_{-\pi}^{\pi} d\varphi \int_{0}^{\pi} d\theta\, N_l^m\left[(l-m+1)P_{l+1}^m - (l+1)\cos\theta P_l^m\right]\cos(m\varphi)\,\tilde{u}_\theta(r,\theta,\varphi) \\ & - \int_{-\pi}^{\pi} d\varphi \int_{0}^{\pi} d\theta\, m N_l^m P_l^m \sin(m\varphi)\,\tilde{u}_\varphi(r,\theta,\varphi),\end{aligned} \tag{A.14b}$$

$$\begin{aligned}\left(r\nabla Y_l^m, \tilde{\mathbf{u}}\right)^{\Im} = {} & -\int_{-\pi}^{\pi} d\varphi \int_{0}^{\pi} d\theta\, N_l^m\left[(l-m+1)P_{l+1}^m - (l+1)\cos\theta P_l^m\right]\sin(m\varphi)\,\tilde{u}_\theta(r,\theta,\varphi) \\ & - \int_{-\pi}^{\pi} d\varphi \int_{0}^{\pi} d\theta\, m N_l^m P_l^m \cos(m\varphi)\,\tilde{u}_\varphi(r,\theta,\varphi),\end{aligned} \tag{A.14c}$$

$$\left(\mathbf{r}\times\nabla Y_l^m,\tilde{\mathbf{u}}\right)^{\Re}=\int_{-\pi}^{\pi}d\varphi\int_0^{\pi}d\theta\, mN_l^m P_l^m\sin(m\varphi)\,\tilde{u}_\theta(r,\theta,\varphi)$$
$$+\int_{-\pi}^{\pi}d\varphi\int_0^{\pi}d\theta\, N_l^m\left[(l-m+1)\,P_{l+1}^m-(l+1)\cos\theta P_l^m\right]\cos(m\varphi)\,\tilde{u}_\varphi(r,\theta,\varphi)\,, \tag{A.14d}$$

$$\left(\mathbf{r}\times\nabla Y_l^m,\tilde{\mathbf{u}}\right)^{\Im}=\int_{-\pi}^{\pi}d\varphi\int_0^{\pi}d\theta\, mN_l^m P_l^m\cos(m\varphi)\,\tilde{u}_\theta(r,\theta,\varphi)$$
$$-\int_{-\pi}^{\pi}d\varphi\int_0^{\pi}d\theta\, N_l^m\left[(l-m+1)\,P_{l+1}^m-(l+1)\cos\theta P_l^m\right]\sin(m\varphi)\,\tilde{u}_\varphi(r,\theta,\varphi)\,. \tag{A.14e}$$

The effect of the choice of the radius r of the surface of integration has been studied in G&P, and on this basis we typically take $r/a\approx 1.2$ for the Reynolds numbers encountered in this study.

A.4. Analytic boundary condition application

At the κ-th step of Lamb's coefficient iterations, we solve the Navier–Stokes equations by the fractional step method of Brown et al. [26], namely

$$\frac{\mathbf{U}_\kappa^*-\mathbf{U}^n}{\Delta t}=-\left[(\mathbf{U}\cdot\nabla_h)\,\mathbf{U}\right]_\kappa^{n+1/2}-\frac{1}{\rho_f}\nabla_h p^{n-1/2}+\nu\nabla_h^2\mathbf{U}_\kappa^{n+1/2}+\mathbf{g}, \tag{A.15}$$

$$\mathbf{U}_{\kappa+1}^{n+1}=\mathbf{U}_\kappa^*-\frac{\Delta t}{\rho_f}\nabla_h\phi_\kappa^{n+1}, \tag{A.16}$$

$$p_\kappa^{n+1/2}=p^{n-1/2}+\phi_\kappa^{n+1}, \tag{A.17}$$

where the auxiliary field ϕ is determined from

$$\nabla^2\phi_\kappa^{n+1}=\frac{\rho_f}{\Delta t}\nabla\cdot\mathbf{U}_\kappa^*. \tag{A.18}$$

In (A.15) we treat both the convective and diffusive terms explicitly as discussed in Section 3.3 using the variable-time second-order Adams–Bashforth formulation with generic quantity $\mathbf{D}$

$$\mathbf{D}^{n+1/2}=\left(1+\frac{1}{2}\frac{\Delta t}{\Delta t_{-1}}\right)\mathbf{D}^n-\frac{1}{2}\frac{\Delta t}{\Delta t_{-1}}\mathbf{D}^{n-1}, \tag{A.19}$$

where $\Delta t_{-1}=t^n-t^{n-1}$ and $\Delta t=t^{n+1}-t^n$. As mentioned in Section 3.1, we solve (A.18) using Dirichlet boundary conditions on the cages found from

$$\phi_\kappa^{n+1}=p_\kappa^{n+1/2}-p^{n-1/2}, \tag{A.20}$$

in which $p_\kappa^{n+1/2}$ is given by (A.2) using (A.6) and the current Lamb's coefficients. We set $\mathbf{U}_\kappa^*=0$ at all velocity nodes inside the cage.

For a given set of Lamb's coefficients, the Stokes solution prescribes specific values of the flow field on the integration surface used for the scalar products. However, the finite-difference solution obtained using the boundary conditions obtained from the same set of Lamb's coefficients will be incompatible with the Stokes solution unless the coefficients are correct. This mismatch drives the iterative process.

At each step of the iteration, we calculate the quantity

$$\frac{\left|\psi_{lm}^{\kappa}-\psi_{lm}^{\kappa-1}\right|}{\left|\psi_{lm}^{\kappa}\right|}, \tag{A.21}$$

where ψ_{lm}^{κ} represents a generic Lamb's coefficient at iteration κ. The iterations are terminated when (A.21) become smaller than a prescribed tolerance, which we typically take as 10^{-2}. To avoid unnecessary iterations, (A.21) is calculated only for coefficients greater than 10^{-4} to 10^{-6} times the largest coefficient. In practice, the number of iterations required for convergence of Lamb's coefficients can vary from one to a few tens depending on many factors including the time step size, the amount of external forcing, the number of particles, and particle collisions. A typical number of sub-time-step iterations is approximately ten. Upon convergence, we set the velocity and pressure nodes inside the particles to reflect the particle

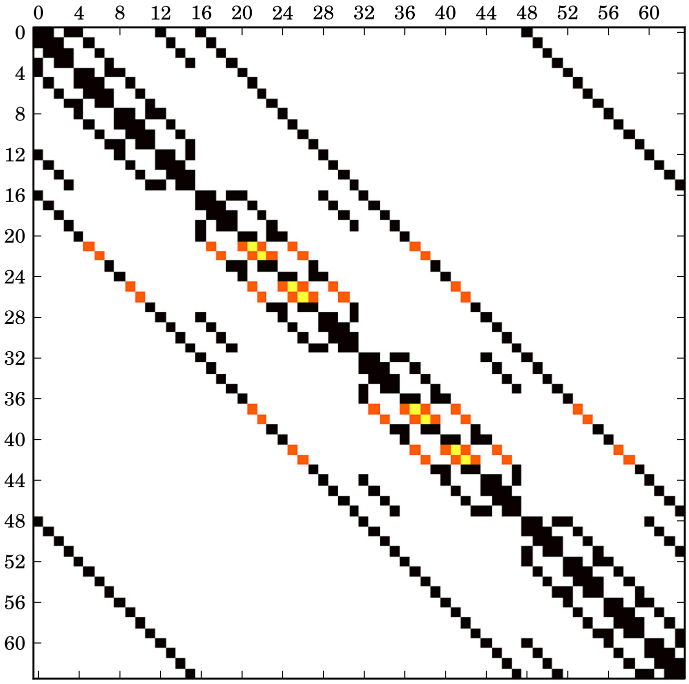


Fig. A.11. The decoupled pressure matrix for an example 4 × 4 × 4 triply-periodic domain with one (severely under-resolved) particle in the center. Nodes that are not white are initially nonzero according to the typical second-order finite difference stencil. When applying the particle boundary conditions, black nodes are unchanged, and yellow and red nodes are set equal to one and zero, respectively. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

solid body motion as explained in Section 2.4. Unlike previous versions of Physalis (Z&P), the current method no longer suffers from the Lamb's coefficient convergence instability that necessitated use of an underrelaxation scheme, although we occasionally use it in connection with very strong collisions.

A.5. Solution of the Poisson equation

Although the Poisson equation is not solved for the nodes inside the cages, removing such nodes from the linear system would destroy the diagonal structure of the matrix and therefore considerably lengthen the conjugate gradient iterations. To deal with this problem, we apply the analytical Dirichlet boundary condition (A.20) to the standard second-order finite difference matrix representation of (A.18) by setting the right-hand side of (A.18) equal to (A.20) at the designated cage nodes. We then edit the corresponding rows and columns in the Laplace operator matrix to be equal to one on the diagonal at the designated cage nodes and zero elsewhere. The resulting matrix structure is shown in Fig. A.11 for a 4 × 4 × 4 triply-periodic domain with one (severely under-resolved for illustration purposes) particle in the center. Here, the matrix elements corresponding to the nodes marked in yellow have a value of one while the red nodes on the same row are set to zero.

A.6. Hydrodynamic forces and couples

One of the biggest benefits of the Physalis method is that Lamb's coefficients determine the hydrodynamic forces, couples, etc. acting on the particles. Phrased another way, the process of determining the flow field also determines these forces, etc. without extra calculations. The hydrodynamic force $\mathbf{F}_h$ and couple $\mathbf{L}_h$ on the particle are expressed in terms of Lamb's coefficients by

$$\hat{\mathbf{F}}_h = \pi\mu v\,(2\boldsymbol{\Phi} - \mathbf{P}) + 2\pi\mu v\,(2\boldsymbol{\Phi} + \mathbf{P}) = \pi\mu v\,(6\boldsymbol{\Phi} + \mathbf{P})\,, \tag{A.22}$$

$$\hat{\mathbf{L}}_h = 8\pi\mu v a\boldsymbol{\chi}\,, \tag{A.23}$$

where $\mathbf{P} = -2N_1^1 p_{11}^{\Re}\hat{\mathbf{i}} + 2N_1^1 p_{11}^{\Im}\hat{\mathbf{j}} + N_1^0 p_{10}^{\Re}\hat{\mathbf{k}}$; $\boldsymbol{\Phi}$ and $\boldsymbol{\chi}$ are defined analogously. The separation into two terms of the force shown in (A.22) identifies the pressure and viscous contributions, respectively. These equations are written in the rest frame of the particle. The forces and couples in the inertial frame of the calculation are obtained by adding the contribution of the transformation (A.2) with the result

$$\mathbf{F}_h = \rho_f v\,(\dot{\mathbf{w}} - \mathbf{g}) + \hat{\mathbf{F}}_h, \qquad \mathbf{L}_h = \rho_f v a^2\dot{\boldsymbol{\Omega}} + \hat{\mathbf{L}}_h, \tag{A.24}$$

with $v = \frac{4}{3}\pi a^3$ the particle volume and where the dots denote differentiation with respect to time.

A.7. Particle translational and rotational motion

The particle acceleration $\dot{\mathbf{w}}$ is computed by adding together the hydrodynamic force $\mathbf{F}_h$, externally imposed forces $\mathbf{F}_e$ (beyond gravity, if any), particle interaction forces $\mathbf{F}_i$ (see Section 4), and applied gravitational force. The accelerations at the time $t^{n+1} = t^n + \Delta t$ and sub-time-step iteration number κ take the form

$$\dot{\mathbf{w}}_\kappa^{n+1} = \frac{1}{m}\left(\mathbf{F}_h^\kappa + \mathbf{F}_e^\kappa + \mathbf{F}_i^\kappa\right) + \mathbf{g}, \qquad \dot{\mathbf{\Omega}}_\kappa^{n+1} = \frac{1}{I}\left(\mathbf{L}_h^\kappa + \mathbf{L}_e^\kappa + \mathbf{L}_i^\kappa\right). \tag{A.25}$$

The counter κ appears in (A.25) because we update the particle linear and angular velocities at each iteration to improve particle interaction accuracy. Now, the particle position $\mathbf{X}$ and velocity $\mathbf{w}$ are given by the trapezoidal rule as

$$\mathbf{w}_\kappa^{n+1} = \mathbf{w}^n + \frac{1}{2}\left(\dot{\mathbf{w}}_\kappa^{n+1} + \dot{\mathbf{w}}^n\right)\Delta t, \qquad \mathbf{X}^{n+1} = \mathbf{X}^n + \frac{1}{2}\left(\mathbf{w}^{n+1} + \mathbf{w}^n\right)\Delta t. \tag{A.26}$$

The angular velocity is updated similarly:

$$\mathbf{\Omega}_\kappa^{n+1} = \mathbf{\Omega}^n + \frac{1}{2}\left(\dot{\mathbf{\Omega}}_\kappa^{n+1} + \dot{\mathbf{\Omega}}^n\right)\Delta t. \tag{A.27}$$

Even though the spherical symmetry decouples the particle orientation from the flow solver, we still track it by attaching an orthonormal basis $\{\mathbf{a}_x, \ \mathbf{a}_y, \ \mathbf{a}_z\}$ to the particle and integrating its motion using $\mathbf{\Omega}$ and quaternion rotations.

The equations change slightly if the body force $\mathbf{g}$ in the fluid momentum equation is incorporated in a modified pressure. In this case, the first of (A.24) becomes

$$\mathbf{F}_h = \rho_f v \dot{\mathbf{w}} + \hat{\mathbf{F}}_h, \tag{A.28}$$

and the first of (A.25) becomes

$$\dot{\mathbf{w}}_\kappa^{n+1} = \frac{1}{m}\left(\mathbf{F}_h^\kappa + \mathbf{F}_e^\kappa + \mathbf{F}_i^\kappa\right) + \frac{\rho_p - \rho_f}{\rho_p}\mathbf{g}. \tag{A.29}$$

It is well known that the added mass effects have an adverse influence on the stability of explicit methods for particles lighter than the fluid [37]. It can be deduced from Equation (5.14) of the stated reference that an explicit method becomes unstable for $\rho_f/\rho_p < C_{am}$ where C_{am} is the added mass coefficient (equal to one half for an isolated sphere in uniform potential flow). We have verified that this stability limit applies to the present algorithm in spite of its partial implicitness. We conclude, therefore, that the partially implicit character of our method is insufficient to stabilize the particle advancement. We retain this procedure because it proves more stable in the presence of collisions.

References

[1] R. Jackson, The Dynamics of Fluidized Particles, Cambridge University Press, Cambridge, 2000.

[2] M. Uhlmann, Interface-resolved direct numerical simulation of vertical particulate channel flow in the turbulent regime, Phys. Fluids 20 (5) (2008) 053305, http://dx.doi.org/10.1063/1.2912459.

[3] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476, http://dx.doi.org/10.1016/j.jcp.2005.03.017.

[4] F. Lucci, A. Ferrante, S. Elghobashi, Modulation of isotropic turbulence by particles of Taylor length-scale size, J. Fluid Mech. 650 (2010) 5–55, http://dx.doi.org/10.1017/S0022112009994022.

[5] T. Kempe, J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, J. Comput. Phys. 231 (9) (2012) 3663–3684, http://dx.doi.org/10.1016/j.jcp.2012.01.021.

[6] F. Picano, W.-P. Breugem, L. Brandt, Turbulent channel flow of dense suspensions of neutrally buoyant spheres, J. Fluid Mech. 764 (2015) 463–487, http://dx.doi.org/10.1017/jfm.2014.704.

[7] A.G. Kidanemariam, M. Uhlmann, Direct numerical simulation of pattern formation in subaqueous sediment, J. Fluid Mech. 750 (2014), http://dx.doi.org/10.1017/jfm.2014.284.

[8] M. Mehrabadi, S. Tenneti, R. Garg, S. Subramaniam, Pseudo-turbulent gas-phase velocity fluctuations in homogeneous gas–solid flow: fixed particle assemblies and freely evolving suspensions, J. Fluid Mech. 770 (2015) 210–246, http://dx.doi.org/10.1017/jfm.2015.146.

[9] J. Yang, F. Stern, A non-iterative direct forcing immersed boundary method for strongly-coupled fluid–solid interactions, J. Comput. Phys. 295 (2015) 779–804, http://dx.doi.org/10.1016/j.jcp.2015.04.040.

[10] C. Ji, A. Munjiza, E. Avital, D. Xu, J. Williams, Saltation of particles in turbulent channel flow, Phys. Rev. E 89 (2014) 052202, http://dx.doi.org/10.1103/PhysRevE.89.052202.

[11] J.J. Wylie, D.L. Koch, A.J.C. Ladd, Rheology of suspensions with high particle inertia and moderate fluid inertia, J. Fluid Mech. 480 (2003) 95–118, http://dx.doi.org/10.1017/S0022112002003531.

[12] S. Sarkar, M.A. van der Hoef, J.A.M. Kuipers, Fluid–particle interaction from lattice Boltzmann simulations for flow through polydisperse random arrays of spheres, Chem. Eng. Sci. 64 (11) (2009) 2683–2691, http://dx.doi.org/10.1016/j.ces.2009.02.045.

[13] S.V. Apte, M. Martin, N.A. Patankar, A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows, J. Comput. Phys. 228 (8) (2009) 2712–2738, http://dx.doi.org/10.1016/j.jcp.2008.11.034.

[14] A. Doostmohammadi, A.M. Ardekani, Suspension of solid particles in a density stratified fluid, Phys. Fluids 27 (2) (2015) 023302, http://dx.doi.org/10.1063/1.4907875.

[15] L. Zeng, S. Balachandar, F.M. Najjar, Wake response of a stationary finite-sized particle in a turbulent channel flow, Int. J. Multiph. Flow 36 (5) (2010) 406–422, http://dx.doi.org/10.1016/j.ijmultiphaseflow.2010.01.001.

[16] J.A. Simeonov, J. Calantoni, Modeling mechanical contact and lubrication in Direct Numerical Simulations of colliding particles, Int. J. Multiph. Flow 46 (2012) 38–53, http://dx.doi.org/10.1016/j.ijmultiphaseflow.2012.05.008.

[17] Z. Zhang, A. Prosperetti, A second-order method for three-dimensional particle simulation, J. Comput. Phys. 210 (1) (2005) 292–324, http://dx.doi.org/10.1016/j.jcp.2005.04.009.

[18] K. Gudmundsson, A. Prosperetti, Improved procedure for the computation of Lamb's coefficients in the physalis method for particle simulation, J. Comput. Phys. 234 (2013) 44–59, http://dx.doi.org/10.1016/j.jcp.2012.08.049.

[19] N. Mordant, J.F. Pinton, Velocity measurement of a settling sphere, Eur. Phys. J. B 18 (3) (2000) 343–352, http://dx.doi.org/10.1007/PL00011074.

[20] G.G. Joseph, R. Zenit, M.L. Hunt, A.M. Rosenwinkel, Particle–wall collisions in a viscous fluid, J. Fluid Mech. 433 (2001) 329–346, http://dx.doi.org/10.1017/S0022112001003470.

[21] H. Lamb, Hydrodynamics, Art. 336, 6th edn., Dover Publications, New York, NY, 1932.

[22] S. Kim, S.J. Karilla, Microhydrodynamics: Principles and Selected Applications, Sect. 4.2, Butterworth-Heinemann, Boston, MA, 1991.

[23] V.I. Lebedev, Quadratures on a sphere, USSR Comput. Math. Math. Phys. 16 (2) (1976) 10–24, http://dx.doi.org/10.1016/0041-5553(76)90100-2.

[24] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (2) (1985) 308–323, http://dx.doi.org/10.1016/0021-9991(85)90148-2.

[25] J.H. Ferziger, M. Perić, Computational Methods for Fluid Dynamics, 3rd edn., Springer, Berlin, 2002.

[26] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2) (2001) 464–499, http://dx.doi.org/10.1006/jcph.2001.6715.

[27] A.S. Sangani, G. Mo, Inclusion of lubrication forces in dynamic simulations, Phys. Fluids 6 (5) (1994) 1653–1662, http://dx.doi.org/10.1063/1.868228.

[28] D.J. Jeffrey, Low-Reynolds-number flow between converging spheres, Mathematika 29 (1982) 58–66, http://dx.doi.org/10.1112/S002557930001216X.

[29] Y. Tsuji, T. Tanaka, T. Ishida, Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe, Powder Technol. 71 (3) (1992) 239–250, http://dx.doi.org/10.1016/0032-5910(92)88030-L.

[30] G. Barnocky, R.H. Davis, Elastohydrodynamic collision and rebound of spheres: experimental verification, Phys. Fluids 31 (6) (1988) 1324–1329, http://dx.doi.org/10.1063/1.866725.

[31] C.T. Crowe, J.D. Schwarzkopf, M. Sommerfeld, Y. Tsuji, Multiphase Flows with Droplets and Particles, 2nd edn., CRC Press, Boca Raton, USA, 2012.

[32] A.J. Sierakowski, Bluebottle: GPU-centric fluid–particle direct numerical simulations using the Physalis method, http://PhysalisCFD.org, 2015, v2.0.2.

[33] A.J. Chorin, The numerical solution of the Navier–Stokes equations for an incompressible fluid, Bull. Am. Math. Soc. 73 (1967) 928–931, http://dx.doi.org/10.1090/S0002-9904-1967-11853-6.

[34] T. Kempe, J. Fröhlich, Collision modelling for the interface-resolved simulation of spherical particles in viscous fluids, J. Fluid Mech. 709 (2012) 445–489, http://dx.doi.org/10.1017/jfm.2012.343.

[35] F. Shaffer, B. Gopalan, The science and beauty of fluidization, APS Gallery of Fluid Motion entry number 102371.

[36] A.J. Sierakowski, GPU-centric resolved-particle disperse two-phase flow simulation using the Physalis method, Comput. Phys. Commun. (2016), in press.

[37] H. Hu, Computational Methods for Multiphase Flow, Chap. 5.2: Finite Element Methods for Particulate Flows: Fully Explicit Scheme and Its Stability, Cambridge University Press, Cambridge, United Kingdom, 2009, pp. 116–117.