Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2008) 7841–7888

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Algorithm for direct numerical simulation of emulsion flow through a granular material

Alexander Z. Zinchenko *, Robert H. Davis *

Department of Chemical and Biological Engineering, University of Colorado, Boulder, CO 80309–0424, USA

Received 30 November 2007; received in revised form 28 April 2008; accepted 1 May 2008
Available online 16 May 2008

Abstract

A multipole-accelerated 3D boundary-integral algorithm capable of modelling an emulsion flow through a granular material by direct multiparticle-multidrop simulations in a periodic box is developed and tested. The particles form a random arrangement at high volume fraction rigidly held in space (including the case of an equilibrium packing in mechanical contact). Deformable drops (with non-deformed diameter comparable with the particle size) squeeze between the particles under a specified average pressure gradient. The algorithm includes recent boundary-integral desingularization tools especially important for drop–solid and drop–drop interactions, the Hebeker representation for solid particle contributions, and unstructured surface triangulations with fixed topology. Multipole acceleration, with two levels of mesh node decomposition (entire drop/solid surfaces and "patches"), is a significant improvement over schemes used in previous, purely multidrop simulations; it remains efficient at very high resolutions (10^4–10^5 triangular elements per surface) and has no lower limitation on the number of particles or drops. Such resolutions are necessary in the problem to alleviate lubrication difficulties, especially for near-critical squeezing conditions, as well as using $\sim 10^4$ time steps and an iterative solution at each step, both for contrast and matching viscosities. Examples are shown for squeezing of 25–40 drops through an array of 9–14 solids, with the total volume fraction of 70% for particles and drops. The flow rates for the drop and continuous phases are calculated. Extensive convergence testing with respect to program parameters (triangulation, multipole truncation, etc.) is made.

Keywords: Emulsion; Granular material; Deformable drop; Multipole

1. Introduction

Squeezing of an emulsion of deformable drops through a granular material of solid particles is a problem of great relevance to many industrial applications (oil filtration through underground reservoirs, flow through fixed-bed catalytic reactors, etc.). Of particular interest are the pressure gradient-flow rate relationships for both the continuous and drop phases, and determining the conditions when the drop squeezing effectively

* Corresponding authors. Tel.: +1 303 492 0475; fax: +1 303 492 4341.
E-mail addresses: alexander.zinchenko@colorado.edu (A.Z. Zinchenko), robert.davis@colorado.edu (R.H. Davis).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.05.004

stops due to trapping in the pores by capillary forces. A phenomenological approach treating the emulsion as a single-phase fluid with effective properties is generally insufficient, since in most interesting cases drops are not small compared to the pore size. To describe the phenomenon from first principles, a method for rigorous, three-dimensional, large-scale microstructural dynamical simulations must be developed instead, which is the subject of the present paper. A granular material is modeled as a random arrangement of many solid spheres rigidly held in a periodic box at high volume fractions (including the most physically relevant case when the particles are mechanically supported by contact forces); the microscale Reynolds number in these applications is typically small and inertial effects are neglected. For a single-phase Stokes flow of a Newtonian liquid through such geometries, much work was done in the past, mostly by multipole methods. Sangani and Acrivos [1] and Zick and Homsy [2] used expansions based on the periodic Green function to consider the flow through periodic lattices. Ladd [3] and Mo and Sangani [4] developed multipole techniques to study the flow through random multiparticle systems. Chapman and Higdon [5] developed a different, multipole collocation tool with a free-space Green function to simulate an oscillatory Stokes flow through periodic lattices, including the case of strongly overlapping spheres.

When drops are present, an intricate geometry of the pore space and drop–solid lubrication make the problem far more complex. An axisymmetrical pressure-driven creeping motion of a single drop through capillary tubes with constriction was simulated in the past as a simplified prototype model, which captures some salient features of the emulsion squeezing. (See [6] for a list of references.) All these simulations were based on boundary-integral equations, of the first kind for tractions on solid boundaries, and could not succeed due to ill conditioning for the case when the drop squeezes with high resistance, even in the axisymmetric geometry. Instead of a boundary-integral method, Graham and Higdon [7,8] used a finite-element approach and fine adaptive domain meshing to study drop squeezing through a tight constriction. They were able to address the near-contact drop–wall interaction and estimate the critical forcing level necessary for squeezing, even with the full Navier–Stokes equations. Their solution technique, though, takes significant advantage of the axial symmetry.

Difficulties with the boundary-integral method to simulate drop motion through a tight constriction were addressed and largely overcome in our recent paper [6], where a 3D flow-driven single drop squeezing through a free-space cluster of two or three particles (spherical or spheroidal) was considered; this solution, although lacking periodic boundaries necessary for emulsion flow simulations, captures realistic granular material microstructure on a small scale. The emphasis in [6] was on the trapping mechanism and flow conditions close to critical. In this most challenging regime, a drop squeezes very slowly, if at all, with small drop–solid spacing, nearly coating the solid boundaries. Using the Hebeker [9] representation for each particle contribution as a proportional combination of single- and double-layer potentials and the reciprocal theorem for the drop interior leads to a well-posed system of Fredholm second-kind integral equations for the Hebeker density on the solids and the fluid velocity on the drop. Divergence of simple (Picard) iterations for this system is not an obstacle, since an alternative iterative scheme (GMRES) works well [6]. This relatively simple starting point is crucially complemented by suitable boundary-integral desingularization tools [6], especially the novel high-order near-singularity subtraction in the solid-to-drop double-layer contribution, to resolve lubrication and avoid drop–solid numerical overlapping, thus allowing simulations to succeed in the near-critical range. Still, we found it necessary to use typically a few thousand time steps for each run and $N_{\Delta} \sim 5000$–10,000 triangular boundary elements per surface for robustness/accuracy as the critical conditions for squeezing are approached; it is unlikely that high-order surface representation/integration rules could alleviate these requirements.

In a parallel recent study, Zhu et al. [10] offered a different boundary-integral formalism, when drops are in the interior of a domain V with prescribed fluid velocity on ∂V. They represent the velocity field as a double-layer contribution from the container plus single-layer contributions from the drops, which also leads to a system of second-kind integral equations. Unlike [6], though, the formulation [10] involves a hypersingular boundary-integral over the container, which we view as a difficulty in applying this method to drop squeezing through constrictions with significant resistance. A numerical example in [10] with a single drop squeezing through an axisymmetrical constricted tube (modeled as 3D) is for the conditions when the drop surface is away from the solid boundary and moves relatively easily through the throat.

Since we seek to extend the method and dynamical boundary-integral simulations [6] to large-scale multidrop-multiparticle problems with periodic boundaries, a standard boundary-integral coding, quadratically intensive in the total number of mesh-nodes, would be far too inefficient to succeed. Relevant here are previous works [11–13] on fast summation of Stokes interactions in 3D, with an asymptotically linear complexity. Sangani and Mo [11] developed the first hydrodynamical version of the traditional electrostatical FMM of Greengard and Rokhlin. Ying et al. [12] developed a kernel-independent version (applicable to Laplacian and Stokes interactions), which retains the logical scheme of FMM but is technically simpler and only slightly slower. Wang et al. [13] recently developed a parallel hydrodynamical version of the new FMM [14]. Of these works, only the code of Sangani and Mo [11] implemented periodic boundaries (which is an additional burden in terms of efficiency) and is significantly oriented on dispersed media simulations. By taking a few multipoles per particle (and adding lubrication analytically, when necessary), they studied sedimentation, effective viscosity and permeability of large random static configurations of solid spheres. Alternative approaches to include periodic boundaries within the framework of FMM are more involved, even for Laplace interactions [15].

In the present work, though, we do not follow the line of general FMM for multipole acceleration, but rather stick to the approach initiated in conductivity simulations by Zinchenko for 2D [16,17] and 3D [18,19] problems and developed further by Zinchenko and Davis [20–22] for 3D hydrodynamical simulations, all with periodic boundaries. Our scheme is logically simpler and has proven highly efficient, with little overheads, in practical dynamical simulations for purely multidrop systems, with several thousand time steps, several hundred drops and moderate resolution $N_{\triangle} \sim 10^3$ in emulsion sedimentation and rheology applications [20–22]. For the homoviscous case exploited most, when an iterative boundary-integral solution is not required, even over 1000 drops could be dynamically simulated [22]. Unlike in FMM, the scheme [20–22] does not rely on hierarchy of space decompositions by Cartesian grids, but uses instead natural grouping of mesh-nodes into interfacial surfaces. Another feature is a wide use of rotation-based techniques for multipole reexpansions and multipoles of high-order with "economical truncation," i.e., a broad spectrum of truncation bounds strongly dependent on mutual geometry of deformable drops for optimized performance. Truncations used in the algorithms [20–22] are typically based on plausible estimations of the behavior of multipole coefficients for the entire surfaces, which is smooth, at least for low indices. Had we chosen Cartesian cells for clustering the mesh-nodes, this behavior could be much less predictable, since a cell may contain fragments from different surfaces.

Compared to our previous purely multidrop simulations [20–22], the present problem of an emulsion flow through a granular material is literally on the next level of challenge, since it requires about 10,000 time steps and at least $N_{\triangle} \sim 10^4$ resolution on each surface, solid particles and drops together take almost the entire space, which further complicates simulations, and an iterative boundary-integral solution is necessary at each time step, even for the homoviscous case. These requirements severely limit the numbers of particles and drops that can be handled. The performance of the multipole acceleration schemes [20–22] having one level of mesh-node decomposition degrades for resolutions $N_{\triangle} \sim 10^4$ and higher, and these schemes were not designed for a small number of surfaces in a periodic cell, which limits their application in the present problem. On the other hand, free-space close interactions of two drops only with superhigh resolution $N_{\triangle} \sim 10^5$ were efficiently handled by a different, one level multipole acceleration scheme [23], with partitioning of each drop surface into a large number of continuous "patches." In the present work, we have combined the approaches [20–23] to arrive at a new multipole acceleration scheme with two levels of mesh-node decomposition, which remains highly efficient for $N_{\triangle} \sim 10^4 - 10^5$ resolutions and has no lower limitation on the number of surfaces in a periodic cell. The logic of this scheme is still less complex than that of FMM. Since the present problem or multidrop dynamical simulations [20–22] have been unchallenged by other, more general fast summation techniques, performance comparison is difficult to make; some available hydrodynamical implementations appear inefficient, judging by the benchmarks [13]. The goal of the present work, though, was not so in benchmarking, but in developing and testing the entire code capable of dynamical simulations for the emulsion flow through a granular material; accordingly, many issues arise, other than fast summation of interactions.

The plan of the paper is as follows. In Section 2, the boundary-integral formulation for many deformable drops squeezing under a constant average pressure gradient is derived, based on the periodic Green function. The drop phase flow rate, one of the main quantities of interest from the solution, is easily expressed through the interfacial velocity on drop surfaces. Less obvious, we also show how to reduce the calculation of the continuous phase flow rate from the solution to surface integrals. In Section 3, we discuss discretizations and adapt the boundary-integral desingularization tools from [6] to the present multidrop-multiparticle case; of particular importance here are the high-order near-singularity subtraction in the solid-to-drop double-layer contribution, and the variational technique for the double-layer contributions from drops; both are crucial to make the present simulations work. In Section 4.1, we discuss the general logic of our new and improved multipole-accelerated scheme for boundary-integral iterations, with a new partition of the periodic Green function into the near- and far-field parts and two levels of mesh-node decomposition (patches and entire surfaces). The next several subsections contain the details of multipole operations (a reader interested in the general structure only for the entire code can proceed to Section 5.2). Section 4.2 presents a new algorithm for generating Lamb's singular series for a patch, which we found to be simpler and considerably more economical than our initial scheme [20] for calculating singular moments. Moreover, the new approach leads, in a non-trivial fashion, to an even much faster routine for the special case of a patch on a spherical surface. In Section 4.3, we detail on merging Lamb's singular expansions for individual patches to form an expansion for the entire surface; this step is rotation-based and made fully optimal in the number of operations. In Section 4.4, far-field contributions to the boundary integrals resulting from interactions between remote images in the periodic Green function are discussed. Sections 4.5 and 4.6, complemented by Appendix A, elaborate on calculation of additional integrals appearing in our boundary-integral formulation, and on economical truncation of multipole expansions/reexpansions. In Section 5.1, we discuss the preiterative part of the boundary-integral calculations. Mesh control (Section 5.2), to retain quality of our unstructured drop surface triangulations with fixed topology in dynamical simulations, is based on passive mesh stabilization [21] combined with occasional active node redistribution through minimization of some new form of a "potential energy" function; relation to the mesh algorithm of Cristini et al. [24] is discussed. Section 5.3 is devoted to generating a start-up configuration of drops with high enough volume fraction between solid particles for dynamical simulation. Relevant here is the paper of Cunha and Loewenberg [25]. They were able to simulate expansion of a periodic (BCC) emulsion (one drop per cell, no solids) through the boundary-integral solution of Stokes equations for compressible fluids even to drop volume fractions of 0.98. They noted, however, that this procedure is very intensive computationally. In the present work, with many drops and particles and high resolution, we have found it even more prohibitive to reach a desired drop volume fraction through expansion in the course of boundary-integral simulations. Instead, Section 5.3 offers a new and simple, yet artificial "swelling technique" to prepare a start-up drop arrangement. Miscellaneous features of our algorithm are discussed in Section 5.4, including optimization for the homoviscous case and control of numerical drop–solid overlapping.

In Section 6.1, drop motion through a free-standing cluster of four spheres is simulated, to test the free-space version of the code and demonstrate a confining effect of solid boundaries on drop deformation. Solutions of Ladd [3], Mo and Sangani [4], and Chapman and Higdon [5] for a single-phase flow through random and periodic beds of spheres provide very useful tests for our code in Section 6.2. Finally, we present in Section 6.3 examples of the most demanding long-time multiparticle-multidrop simulations for contrast and matching viscosities, with convergence and performance analysis.

All timings below are for single-processor calculations on AMD PC, with Opteron 2.8 GHz CPU.

2. Boundary-integral formulation

Consider a three-dimensional flow of an emulsion of deformable drops through a granular material (Fig. 1). The granular material skeleton is modelled as a random arrangement of $\widehat{N}$ solid particles with surface centroids $\hat{\boldsymbol{x}}^c_1, \ldots \hat{\boldsymbol{x}}^c_{\widehat{N}}$ in a periodic cell $[0, L)^3$ and triply-periodic continuation into the whole space. The particles are assumed to be rigidly held, with the no-slip boundary conditions $\boldsymbol{u} = \boldsymbol{0}$ for the triply-periodic fluid velocity $\boldsymbol{u}$ on the particle surfaces $\widehat{S}_1, \ldots \widehat{S}_{\widehat{N}}$ and their images. The drop phase is formed by $\widetilde{N}$ deformable surfaces $\widetilde{S}_1, \ldots \widetilde{S}_{\widetilde{N}}$ with centroids $\tilde{\boldsymbol{x}}^c_1, \ldots \tilde{\boldsymbol{x}}^c_{\widetilde{N}} \in [0, L)^3$ and triply-periodic continuation. The drops are Newtonian, free from surfactants, have a constant interfacial tension σ and viscosity μ^{int} and are freely suspended in a Newtonian continuous phase with viscosity μ^e. The microscale Reynolds number is assumed small and so the Stokes equations apply. To make the solution unique for each configuration, some integral properties must be specified. By an analogy with drop motion through periodically constricted tubes [8], we can prescribe either a constant flow rate for the emulsion, or a constant-pressure gradient. Our ultimate goal is to handle

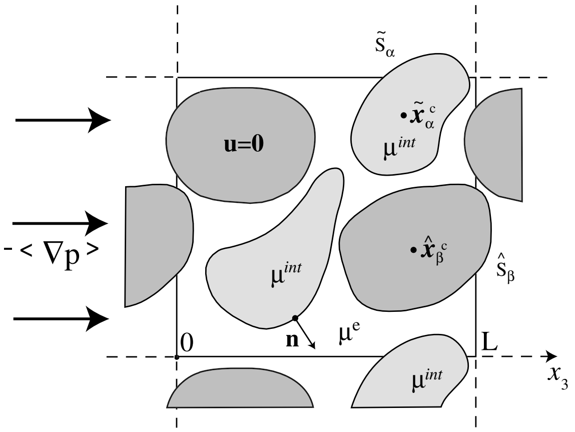


Fig. 1. Sketch for the flow of deformable drops (light shading) through a random array of fixed solid particles (dark shading) in a periodic box $[0, L)^3$ under a pressure gradient, not to scale.

large systems with $\widehat{N}$, $\widetilde{N} \gg 1$. For such systems, the difference between the two formulations is expected to disappear (Higdon, J.J.L., personal communication). The constant-pressure gradient formulation, though, provides some computational advantages (Section 5.4) and was chosen in the present work. The pressure is represented as $p(\boldsymbol{x}) = \langle \nabla p \rangle \cdot \boldsymbol{x}$ plus a periodic function, with a given average pressure gradient $\langle \nabla p \rangle$. Integrating the stress vector over the boundary of the periodic cell and using the divergence theorem represents the average pressure gradient condition in a convenient form

$$\sum_{\beta=1}^{\widehat{N}} \widehat{\boldsymbol{\mathcal{F}}}_\beta = -\langle \nabla p \rangle V, \tag{2.1}$$

where the LHS is the total hydrodynamical force acting on the solid surfaces $\widehat{S}_1, \ldots \widehat{S}_{\widehat{N}}$, and $V = L^3$ is the cell volume. Using the standard technique of energetic inequalities (e.g., [26]), it is easy to see that (2.1), indeed, makes the problem uniquely solvable for each configuration. The equations below are made non-dimensional using $U = |\langle \nabla p \rangle| \widehat{a}^2 / \mu^e$ and L as the velocity and length scales, respectively, where $\widehat{a}$ is the characteristic radius of solid particles; for simplicity, the average pressure gradient is along the negative x_3-axis.

A system of boundary-integral equations is facilitated through the use of Hasimoto's [27] periodic Green functions $\boldsymbol{G}^{(k)}(\boldsymbol{x})(k = 1, 2, 3)$ and corresponding stress tensors $\boldsymbol{\tau}^{(k)}(\boldsymbol{x})$. The vectors $\boldsymbol{G}^{(k)}(\boldsymbol{x})$ and the corresponding pressures $\mathcal{P}^{(k)}(\boldsymbol{x})$ satisfy

$$\nabla^2 \boldsymbol{G}^{(k)}(\boldsymbol{x}) - \nabla \mathcal{P}^{(k)}(\boldsymbol{x}) = \nabla \cdot \boldsymbol{\tau}^{(k)}(\boldsymbol{x}) = \sum_{\boldsymbol{m}} \delta(\boldsymbol{x} - \boldsymbol{m}) \boldsymbol{e}_k, \tag{2.2}$$

where the summation is over all lattice points $\boldsymbol{m} = (m_1, m_2, m_3)$ with integer m_1, m_2, m_3, and $\boldsymbol{e}_k$ are basis vectors. The additive constants in $\boldsymbol{G}^{(k)}$ are chosen so that the average of $\boldsymbol{G}^{(k)}$ over the periodic cell is zero.

Green's theorem and no-slip conditions on $\widehat{S}_\alpha$ give, for a point $\boldsymbol{y}$ lying outside the particles and drops,

$$u_k^e(\boldsymbol{y}) = \sum_{\beta=1}^{\widetilde{N}} \int_{\widetilde{S}_\beta} \left[\boldsymbol{G}^{(k)}(\boldsymbol{r}) \cdot \boldsymbol{T}_n^e(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x}) \cdot \boldsymbol{\tau}^{(k)}(\boldsymbol{r}) \cdot \boldsymbol{n}(\boldsymbol{x}) \right] \mathrm{d}S_x + \sum_{\beta=1}^{\widehat{N}} \int_{\widehat{S}_\beta} \boldsymbol{G}^{(k)}(\boldsymbol{r}) \cdot \boldsymbol{T}_n^e(\boldsymbol{x}) \mathrm{d}S_x + \langle u_k \rangle. \tag{2.3}$$

Here and henceforth, index e marks the values for the continuous phase, $\boldsymbol{T}_n = \boldsymbol{T} \cdot \boldsymbol{n}$ is the stress vector ($\boldsymbol{T}$ being the stress tensor), $\boldsymbol{n}$ is the outward normal to a surface, and $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}$. The additive constant $\langle \boldsymbol{u} \rangle$ (the average velocity over all phases, with $\boldsymbol{u} = \boldsymbol{0}$ inside $\widehat{S}_\alpha$) stems from the integration over the cell boundary (cf. with Eq. (2.4) of Zinchenko and Davis [20]). The flow inside the drops is excluded in a now standard way [28] through the reciprocal theorem,

$$\int_{\widetilde{S}_\beta} \left[\boldsymbol{G}^{(k)}(\boldsymbol{r}) \cdot \boldsymbol{T}_n^{\mathrm{int}}(\boldsymbol{x}) - \lambda \boldsymbol{u}(\boldsymbol{x}) \cdot \boldsymbol{\tau}^{(k)}(\boldsymbol{r}) \cdot \boldsymbol{n}(\boldsymbol{x}) \right] \mathrm{d}S_x = 0 \tag{2.4}$$

and stress jump conditions at an interface, to yield

$$u_k^e(\boldsymbol{y}) = F_k(\boldsymbol{y}) + (\lambda - 1)\sum_{\beta=1}^{\widetilde{N}} \int_{\widetilde{S}_\beta} \boldsymbol{u}(\boldsymbol{x})\cdot\boldsymbol{\tau}^{(k)}(\boldsymbol{r})\cdot\boldsymbol{n}(\boldsymbol{x})\mathrm{d}S_x + \sum_{\beta=1}^{\widehat{N}} \int_{\widehat{S}_\beta} \boldsymbol{G}^{(k)}(\boldsymbol{r})\cdot\boldsymbol{T}_n^e(\boldsymbol{x})\mathrm{d}S_x + \langle u_k\rangle, \tag{2.5}$$

with

$$\boldsymbol{F}(\boldsymbol{y}) = \frac{2\sigma}{|\langle\nabla p\rangle|\widehat{a}^2}\sum_{\beta=1}^{\widetilde{N}} \int_{\widetilde{S}_\beta} k(\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x})\cdot\boldsymbol{G}(\boldsymbol{r})\mathrm{d}S_x. \tag{2.6}$$

In Eqs. (2.4)–(2.6), index *int* relates to the drop phase, $\lambda = \mu^{int}/\mu^e$ is the viscosity ratio, the non-dimensional group $|\langle\nabla p\rangle|\widehat{a}^2/\sigma$ is related to the capillary number (Section 6), $k(\boldsymbol{x}) = (k_1 + k_2)/2$ is the mean surface curvature at $\boldsymbol{x} \in \widetilde{S}_\beta$, and $\boldsymbol{G} = (\boldsymbol{G}^{(1)}, \boldsymbol{G}^{(2)}, \boldsymbol{G}^{(3)})$ is the symmetric second-rank Green tensor.

It is inconvenient that the kernel $\boldsymbol{\tau}^{(k)}(\boldsymbol{r})$ is not periodic, but contains a linearly growing part from the pressure $\mathcal{P}^{(k)}(\boldsymbol{x})$, so we recast (2.5) in terms of the periodic kernel (cf. [20])

$$\breve{\boldsymbol{\tau}}^{(k)}(\boldsymbol{r}) = \boldsymbol{\tau}^{(k)}(\boldsymbol{r}) - r_k\boldsymbol{I}. \tag{2.7}$$

Note that $\breve{\boldsymbol{\tau}}^{(k)}$ has a constant, non-zero divergence at $\boldsymbol{r} \neq \boldsymbol{m}$ and the tensor $\breve{\boldsymbol{\tau}} = \left\{\breve{\tau}_{ij}^{(k)}\right\}$ is now symmetric in all three indices. With $\boldsymbol{\tau}^{(k)}$ replaced by $\breve{\boldsymbol{\tau}}^{(k)}$, Eq. (2.5) still holds, just $\langle\boldsymbol{u}\rangle$ is replaced by another, yet unknown additive constant which can be found, in principle, when the solution is substituted into (2.1). However, for any prescribed additive constant, the form (2.5) leads to a system of integral equations of the second-kind for $\boldsymbol{u}$ on $\widetilde{S}_\alpha$, but of the first kind for the fractions $\boldsymbol{T}_n^e$ on solids $\widehat{S}_\alpha$. Our experience for a single drop interacting with a finite, free-space cluster of particles [6] shows that an approach involving first kind integral equations fails to simulate 3D drop squeezing through tight constrictions due to ill-conditioning, and it must be also avoided in the present, more complex problem. Double-layer representation for the solid–particle contributions could not be used either, since it is range-deficient and cannot accommodate non-zero hydrodynamical forces and torques acting on $\widehat{S}_\beta$.

Instead, we notice that each integral in (2.5) over $\widehat{S}_\beta$ is a periodic Stokes flow outside $\widehat{S}_\beta$ and its images with a zero total flux through $\widehat{S}_\beta$, and it can be converted into a linear combination of a single- and double-layer potentials:

$$\int_{\widehat{S}_\beta} \boldsymbol{G}(\boldsymbol{r})\cdot\boldsymbol{T}_n^e(\boldsymbol{x})\mathrm{d}S_x = \int_{\widehat{S}_\beta} \boldsymbol{q}(\boldsymbol{x})\cdot[\eta\boldsymbol{G}(\boldsymbol{r}) + 2\breve{\boldsymbol{\tau}}(\boldsymbol{r})\cdot\boldsymbol{n}(\boldsymbol{x})]\mathrm{d}S_x, \tag{2.8}$$

given an arbitrary factor $\eta > 0$. Indeed, assuming temporarily that $\boldsymbol{T}_n^e(\boldsymbol{x})$ in the LHS is given, (2.8) gives a boundary-integral equation for $\boldsymbol{q}(\boldsymbol{x})$ on $\widehat{S}_\beta$ with a unique solution (the proof is analogous to that of Hebeker [9] for a free-space flow past a single solid body). Moreover, it follows from this equation for $\boldsymbol{q}(\boldsymbol{x})$ that the total flux of $\boldsymbol{q}(\boldsymbol{x})$ through $\widehat{S}_\beta$ is zero, a condition necessary for the RHS of (2.8) to be a Stokes flow, which proves the representation (2.8). In terms of the Hebeker density $\boldsymbol{q}(\boldsymbol{x})$, the force balance (2.1) takes the form

$$\eta\sum_{\beta=1}^{\widehat{N}} \int_{\widehat{S}_\beta} \boldsymbol{q}(\boldsymbol{x})\mathrm{d}S = \left(\frac{L}{\widehat{a}}\right)^2\boldsymbol{e}_3. \tag{2.9}$$

Let the prime denote the projection of a vector field on the subspace of rigid-body motions on a surface. Introducing the fluctuations $\boldsymbol{Q} = \boldsymbol{q} - \boldsymbol{q}'$ (on $\widehat{S}_\alpha$) and $\boldsymbol{Q} = \boldsymbol{u} - \boldsymbol{u}'$ (on $\widetilde{S}_\alpha$), the representation (2.5) can be written as

$$\boldsymbol{u}^e(\boldsymbol{y}) = \boldsymbol{u}_0(\boldsymbol{y}) + \boldsymbol{C}, \tag{2.10}$$

where

$$\begin{aligned}\boldsymbol{u}_0(\boldsymbol{y}) = \boldsymbol{F}(\boldsymbol{y}) &+ (\lambda - 1)\sum_{\beta=1}^{\widetilde{N}} \int_{\widetilde{S}_\beta} \boldsymbol{Q}(\boldsymbol{x})\cdot\breve{\boldsymbol{\tau}}(\boldsymbol{r})\cdot\boldsymbol{n}(\boldsymbol{x})\mathrm{d}S_x + \sum_{\beta=1}^{\widehat{N}} \int_{\widehat{S}_\beta} \boldsymbol{Q}(\boldsymbol{x})\cdot[\eta\boldsymbol{G}(\boldsymbol{r}) + 2\breve{\boldsymbol{\tau}}(\boldsymbol{r})\cdot\boldsymbol{n}(\boldsymbol{x})]\mathrm{d}S_x \\ &+ \sum_{\beta=1}^{\widehat{N}} \int_{\widehat{S}_\beta} \boldsymbol{q}'(\boldsymbol{x})\cdot[\eta\boldsymbol{G}(\boldsymbol{r}) + 2\breve{\boldsymbol{\tau}}(\boldsymbol{r})\cdot\boldsymbol{n}(\boldsymbol{x})]\mathrm{d}S_x\end{aligned} \tag{2.11}$$

and $\boldsymbol{C}$ is another additive constant. Taking the limit $\boldsymbol{y} \to \widetilde{S}_\alpha$ (or $\widehat{S}_\alpha$) gives the system of second-kind integral equations for $\boldsymbol{q}(\boldsymbol{x})$ and $\boldsymbol{u}(\boldsymbol{x})$. The spectral properties are improved (for $\lambda \ll 1$ or $\lambda \gg 1$, although slightly in the present calculations) when this system is recast in terms of the "deflated" velocity [29] $\boldsymbol{w} = \boldsymbol{u} - \kappa\boldsymbol{u}'$ (with $\kappa = (\lambda - 1)/(\lambda + 1)$ and $\boldsymbol{w} - \boldsymbol{w}' = \boldsymbol{u} - \boldsymbol{u}'$), so the equations to solve for $\boldsymbol{q}$ and $\boldsymbol{w}$ take the form

$$\boldsymbol{q}(\boldsymbol{y}) = (\boldsymbol{u}_0)_S(\boldsymbol{y}) + \boldsymbol{C}, \quad \boldsymbol{y} \in \widehat{S}_\alpha, \tag{2.12}$$
$$\boldsymbol{w}(\boldsymbol{y}) = \frac{2}{\lambda + 1}\left[(\boldsymbol{u}_0)_S(\boldsymbol{y}) + \boldsymbol{C}\right] + \frac{\kappa \boldsymbol{n}(\boldsymbol{y})}{\widetilde{S}_\alpha}\int_{\widetilde{S}_\alpha} \boldsymbol{w}\cdot\boldsymbol{n}\,\mathrm{d}S, \quad \boldsymbol{y} \in \widetilde{S}_\alpha,$$

where the subscript S denotes direct values on a surface. The constant $\boldsymbol{C}$ is expressed by substituting the first relation (2.12) into the force balance (2.9), and so the system (2.12) can be written in the operator form $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}$ for $\boldsymbol{X} = (\boldsymbol{q}, \boldsymbol{w})$. A traditional iterative method of "successive substitutions" into the RHS is divergent for this system, but successful solutions are obtained by minimal residual iterations (see Section 5.4 for more detail). Rigid-body projections $\boldsymbol{q}', \boldsymbol{w}'$ are easy to calculate, e.g., $\boldsymbol{q}' = \langle \boldsymbol{q}\rangle_\alpha + \boldsymbol{\Omega} \times (\boldsymbol{x} - \hat{\boldsymbol{x}}^c_\alpha)$ on $\widehat{S}_\alpha$, where $\langle\cdots\rangle_\alpha$ is the surface average over $\widehat{S}_\alpha$, and the vector $\boldsymbol{\Omega}$ is the solution of a 3×3 system [30]:

$$\left\{\int_{\widehat{S}_\alpha}\left[(\boldsymbol{x} - \hat{\boldsymbol{x}}^c_\alpha)^2\boldsymbol{I} - (\boldsymbol{x} - \hat{\boldsymbol{x}}^c_\alpha)(\boldsymbol{x} - \hat{\boldsymbol{x}}^c_\alpha)\right]\mathrm{d}S\right\}\boldsymbol{\Omega} = \int_{\widehat{S}_\alpha}(\boldsymbol{x} - \hat{\boldsymbol{x}}^c_\alpha) \times \boldsymbol{w}\mathrm{d}\boldsymbol{S}. \tag{2.13}$$

The interfacial velocity is recovered as $\boldsymbol{u} = \boldsymbol{w} + (\lambda - 1)\boldsymbol{w}'/2$. In the limit of fine surface discretizations, results are independent of $\eta > 0$, but in practice, values of $\eta \sim \hat{a}^{-1}$ are optimal for numerical solutions, as for finite clusters of particles [6].

The last term in (2.11) is calculated analytically, which is possible for spheres (Section 4.5) and other canonic solid particle shapes (e.g., spheroids, 3D ellipsoids, etc.). This present limitation on our algorithm still provides enough generality for a granular material with globular grains, whereas particles of more complex shapes would introduce too many parameters to make it a tractable study. We have found the splitting $\boldsymbol{q} = \boldsymbol{q}' + \boldsymbol{Q}$ to be very important in the case of high solid volume fractions (typical of a packed granular material): due to low permeability, $\boldsymbol{q}(\boldsymbol{x})$ takes on relatively large values, and subtracting the rigid-body component $\boldsymbol{q}'$ (most substantially, the $\langle\boldsymbol{q}\rangle$-part of it) greatly reduces the numerical error; otherwise, the convergence is poor and our dynamical simulations could not succeed. This issue did not arise for finite clusters [6].

The main quantities of interest from the solution of (2.9) and (2.12) are the non-dimensional, instantaneous drop phase ($\boldsymbol{U}_D$) and the continuous phase ($\boldsymbol{U}_C$) velocities, with subsequent time averaging. Calculation of $\boldsymbol{U}_D$ is simply reduced to surface integrals (cf. [6]):

$$\boldsymbol{U}_D = \frac{1}{c_d}\sum_{\alpha=1}^{\widetilde{N}}\int_{\widetilde{\mathcal{V}}_\alpha}\boldsymbol{u}\mathrm{d}V = \frac{1}{c_d}\sum_{\alpha=1}^{\widetilde{N}}\int_{\widetilde{S}_\alpha}(\boldsymbol{u}\cdot\boldsymbol{n})(\boldsymbol{x} - \tilde{\boldsymbol{x}}^c_\alpha)\mathrm{d}S, \tag{2.14}$$

where $\widetilde{\mathcal{V}}_\alpha$ is the domain bounded by $\widetilde{S}_\alpha$, and c_d is the drop volume fraction in the total space. For $\boldsymbol{U}_C$, which is the volume average of $\boldsymbol{u}$ over the continuous phase in the periodic cell $[0,1)^3$, the procedure is less obvious. We note that $\breve{\boldsymbol{\tau}}(\boldsymbol{r})$ is a periodic, odd function and, hence, it has a zero mean over any periodic cell, as does $\boldsymbol{G}(\boldsymbol{r})$. Consequently, the field (2.11) $\boldsymbol{u}_0(\boldsymbol{y})$, if formally continued into the whole space, has a zero mean over $[0,1)^3$, and, from (2.10),

$$\boldsymbol{U}_C = \boldsymbol{C} - \frac{1}{1 - c_s - c_d}\left[\sum_{\alpha=1}^{\widetilde{N}}\int_{\widetilde{\mathcal{V}}_\alpha}\boldsymbol{u}_0\mathrm{d}V + \sum_{\alpha=1}^{\widehat{N}}\int_{\widehat{\mathcal{V}}_\alpha}\boldsymbol{u}_0\mathrm{d}V\right], \tag{2.15}$$

where c_s is the solid volume fraction in the total space, and $\widehat{\mathcal{V}}_\alpha$ is the domain bounded by $\widehat{S}_\alpha$. It remains to transform (2.15) to surface integrals (cf. (2.14)) and relate the boundary values of $\boldsymbol{u}_0$ on the inner side of $\widetilde{S}_\alpha$ (or $\widehat{S}_\alpha$) to $(\boldsymbol{u}_0)_S$ through the jump properties of the double-layer potentials (2.11), to obtain:

$$\boldsymbol{U}_C = \boldsymbol{C} - \frac{1}{1 - c_s - c_d}\left\{\sum_{\alpha=1}^{\widetilde{N}}\int_{\widetilde{S}_\alpha}\left\{\left[(\boldsymbol{u}_0)_S + \frac{(\lambda - 1)}{2}\boldsymbol{Q}\right]\cdot\boldsymbol{n}\right\}(\boldsymbol{x} - \tilde{\boldsymbol{x}}^c_\alpha)\mathrm{d}S + \sum_{\alpha=1}^{\widehat{N}}\int_{\widehat{S}_\alpha}\left\{\left[(\boldsymbol{u}_0)_S + \boldsymbol{q}\right]\cdot\boldsymbol{n}\right\}(\boldsymbol{x} - \hat{\boldsymbol{x}}^c_\alpha)\mathrm{d}S\right\}. \tag{2.16}$$

Relation (2.16) must be used as written, with $(\boldsymbol{u}_0)_S$ taken from (2.11); we found that further analytical manipulations using (2.12) can bring simpler expressions, but with poor numerical convergence.

Instead of the Hebeker form, a Power-Miranda [31] representation could be used for the solid–particle contribution (2.8) as a double-layer plus additional Stokeslet and Rotlet contributions from the particle center to complete the range. Such an approach was tested for finite clusters [6] but the Hebeker form was found to be considerably more robust, when a drop squeezes with high resistance. We came to the same conclusion in the present calculations.

3. Discretization and desingularization

Each drop and particle surface is represented by an unstructured mesh of flat triangles with vertices $\boldsymbol{x}^j$ (called the mesh collocation nodes). All regular integrals (e.g., (2.13)–(2.16)) and boundary integrals (after sufficient desingularization) are calculated by the simplest second-order trapezoidal rule, with reassignment of triangle contributions to vertices [32]:

$$\int_S \varphi(\boldsymbol{x})\mathrm{d}S \approx \sum_{\boldsymbol{x}^j \in S} \varphi(\boldsymbol{x}^j)\triangle S_j, \tag{3.1}$$

where $\triangle S_j$ is 1/3 of the sum of flat triangle areas sharing node $\boldsymbol{x}^j$. The rule (3.1), however primitive it may appear, is very economical and has significant advantanges in the problems with close surface interactions (including those in [20–23] and the present problem). The solution in this case is not smooth enough and a large number of boundary elements is needed anyway, which discourages the use of much more expensive high-order integration schemes.

The singular and near-singular behavior of the integrands in (2.6), (2.11) at $\boldsymbol{x} - \boldsymbol{y} \approx \boldsymbol{m}$ stems from the free-space contributions

$$\boldsymbol{G}_0(\boldsymbol{\xi}) = -\frac{1}{8\pi}\left(\frac{\boldsymbol{I}}{\xi} + \frac{\boldsymbol{\xi\xi}}{\xi^3}\right), \quad \boldsymbol{\tau}_0(\boldsymbol{\xi}) = \frac{3}{4\pi}\frac{\boldsymbol{\xi\xi\xi}}{\xi^5} \tag{3.2}$$

to $\boldsymbol{G}$ and $\breve{\boldsymbol{\tau}}$, and it has to be alleviated for successful simulations. A set of desingularization tools found most suitable for a single drop squeezing through a finite free-space cluster [6] is used herein, but with necessary adaptations and optimizations for a multiparticle-multidrop system, as outlined below.

3.1. Drop-to-solid and drop-to-drop contributions

Let $\boldsymbol{x}^{i*}$ be the mesh-node on $\widetilde{S}_\beta$, which is closest to the periodic shift $\boldsymbol{y} - \boldsymbol{m}$ of mesh-node $\boldsymbol{y}$. Introducing the fluctuations $f(\boldsymbol{x}) = k(\boldsymbol{x}) - \langle k \rangle_\beta$ on $\widetilde{S}_\beta$, the single-layer integrals (2.6) are approximated as

$$\int_{\widetilde{S}_\beta} k(\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{G}(\boldsymbol{r})\mathrm{d}S_x \approx \sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} f(\boldsymbol{x}^j)\boldsymbol{n}(\boldsymbol{x}^j) \cdot \boldsymbol{G}(\boldsymbol{x}^j - \boldsymbol{y})\triangle S_j - \sum_{\boldsymbol{m}} \Theta_1 f^* \sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} \boldsymbol{G}_0(\boldsymbol{x}^j - \boldsymbol{y} + \boldsymbol{m}) \cdot \boldsymbol{n}(\boldsymbol{x}^j)\triangle S_j. \tag{3.3}$$

Here Θ_1 is a "barrier function," equal to 1 for $|\boldsymbol{y} - \boldsymbol{m} - \boldsymbol{x}^{i*}| < h_0$ and zero otherwise, where the threshold h_0 is normally set to $0.25\widehat{a}$; the undesirable effect of discontinuity in Θ_1 is alleviated by using the fluctuations $f(\boldsymbol{x})$ instead of $k(\boldsymbol{x})$. The value f^* of $f(\boldsymbol{x})$ at the *surface* point $\boldsymbol{x}_S^* \in \widetilde{S}_\beta$ nearest to $\boldsymbol{y} - \boldsymbol{m}$ is calculated by linear interpolation from the set of nodes $\mathcal{A}_{i*}$ directly connected to $\boldsymbol{x}^{i*}$:

$$f^* = f(\boldsymbol{x}^{i*}) + \sum_{j \in \mathcal{A}_{i*}} \boldsymbol{c}_{j,i*} \cdot (\boldsymbol{y} - \boldsymbol{m} - \boldsymbol{x}^{i*})[f(\boldsymbol{x}^j) - f(\boldsymbol{x}^{i*})], \tag{3.4}$$

where $\boldsymbol{c}_{j,i} = (c_{1,j,i}, c_{2,j,i}, c_{3,j,i})$ and the coefficients $c_{k,j,i}$ depend only on the surface geometry around $\boldsymbol{x}^i$ and are given by simple formulae (3.12) of Zinchenko and Davis [6]; f^* is used instead of a more straightforward $f(\boldsymbol{x}^{i*})$ to smooth the subtracted term. For $\boldsymbol{m} = \boldsymbol{0}$ and $\boldsymbol{y} \in \widetilde{S}_\beta$ ("self-interactions"), $\boldsymbol{x}^j = \boldsymbol{y}$ is excluded from the subtracted term (3.3) (see Section 4.1 for more detail). The subtracted term in (3.3) provides full desingularization, but disappears in the limit of fine triangulations. Owing to the barrier Θ_1, the subtracted terms (3.3) are in effect only for close interactions and are relatively inexpensive to calculate by the simplest, point-to-point summations.

For the double-layer contributions from drops (2.11) (omitting the factor $\lambda - 1$), a strongly singular ($\sim r^{-2}$) behavior of the integrands is alleviated by the variational technique of Zinchenko and Davis [21]:

$$\int_{\widetilde{S}_\beta} \boldsymbol{Q}(\boldsymbol{x}) \cdot \breve{\boldsymbol{\tau}}(\boldsymbol{r}) \cdot \boldsymbol{n}(\boldsymbol{x}) \mathrm{d}S_x \approx \sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} \boldsymbol{Q}(\boldsymbol{x}^j) \cdot \breve{\boldsymbol{\tau}}(\boldsymbol{x}^j - \boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{x}^j) \triangle S_j - \sum_{\boldsymbol{m}} \Theta_1 \boldsymbol{Q}^* \cdot \sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} \boldsymbol{\tau}_0(\boldsymbol{x}^j - \boldsymbol{y} + \boldsymbol{m}) \cdot \boldsymbol{n}(\boldsymbol{x}^j) \triangle S_j \tag{3.5}$$

(with the same barrier function Θ_1 as in (3.3)). The quantity $\boldsymbol{Q}^*$ is required to minimize the Euclidean norm of the free-space contribution to (3.5) from each $\boldsymbol{m}$ after subtraction, which yields, after some algebra [6,21]:

$$\boldsymbol{Q}^* \cdot \sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} \boldsymbol{\tau}_0(\boldsymbol{\xi}^j) \cdot \boldsymbol{n}(\boldsymbol{x}^j) \triangle S_j = \boldsymbol{D}^T \cdot \sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} \frac{\left[\boldsymbol{\xi}^j \cdot \boldsymbol{n}(\boldsymbol{x}^j) \triangle S_j\right]^2 \left[\boldsymbol{\xi}^j \cdot \boldsymbol{Q}(\boldsymbol{x}^j)\right] \boldsymbol{\xi}^j}{\left|\boldsymbol{\xi}^j\right|^8}, \tag{3.6}$$

where, for brevity, $\boldsymbol{\xi}^j = \boldsymbol{x}^j - \boldsymbol{y} + \boldsymbol{m}$, and the matrices

$$\boldsymbol{D} = \frac{3}{4\pi} \left[\sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} \frac{\left[\boldsymbol{\xi}^j \cdot \boldsymbol{n}(\boldsymbol{x}^j) \triangle S_j\right]^2 \boldsymbol{\xi}^j \boldsymbol{\xi}^j}{\left|\boldsymbol{\xi}^j\right|^8} \right]^{-1} \sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} \frac{\left[\boldsymbol{\xi}^j \cdot \boldsymbol{n}(\boldsymbol{x}^j)\right] \triangle S_j \boldsymbol{\xi}^j \boldsymbol{\xi}^j}{\left|\boldsymbol{\xi}^j\right|^5} \tag{3.7}$$

are precalculated (for $\Theta_1 \neq 0$) before the iterations (with compact storage). In the special case of "self-interactions" ($\boldsymbol{m} = \boldsymbol{0}$, $\boldsymbol{y} \in \widetilde{S}_\beta$, $\Theta_1 = 1$), (3.6) and (3.7) are ignored, and $\boldsymbol{Q}^*$ is set to $\boldsymbol{Q}(\boldsymbol{y})$, with $\frac{1}{2}\boldsymbol{Q}(\boldsymbol{y})$ added to (3.5), which is akin to the standard singularity subtraction in the free-space double-layer. Again, the barrier Θ_1 considerably limits the amount of work, and the corresponding operations in (3.5)–(3.7) are handled in the simplest, point-to-point manner. The technique (3.6) and (3.7) greatly improves the spectral properties of the discretized boundary-integral equations for close drop–drop and drop–solid interactions in concentrated systems at $\lambda \neq 1$, and it is one of the most essential elements of our algorithm. With a more obvious choice $\boldsymbol{Q}^* = \boldsymbol{Q}(\boldsymbol{x}^{i*})$ in (3.5), iterations become nonconvergent after a short simulation time, which was observed both for the emulsion shear flow [21] and in the present calculations, even with a moderate viscosity contrast; to make the choice $\boldsymbol{Q}^* = \boldsymbol{Q}(\boldsymbol{x}^{i*})$ work, much higher surface triangulations would be required. It can be noted, however, that the technique (3.5), (3.6), and (3.7) does not completely eliminate singularity of the integrands, only reduces it to $\mathrm{O}(\xi^{-1})$.

3.2. *Solid–solid contribution*

For each integral (2.11) with $\boldsymbol{Q}(\boldsymbol{x})$ over $\widehat{S}_\beta$, when the observation point $\boldsymbol{y} \in \widehat{S}_\alpha$, a simpler desingularization is used, and this integral is calculated as

$$\sum_{\boldsymbol{x}^j \in \widehat{S}_\beta} \boldsymbol{Q}(\boldsymbol{x}^j) \cdot [\eta \boldsymbol{G}(\boldsymbol{x}^j - \boldsymbol{y}) + 2\breve{\boldsymbol{\tau}}(\boldsymbol{x}^j - \boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{x}^j)] \triangle S_j + \sum_{\boldsymbol{m}} \Theta_2 \boldsymbol{Q}(\boldsymbol{x}^{i*}) \cdot \boldsymbol{\Pi}(\boldsymbol{y} - \boldsymbol{m}, \beta), \tag{3.8}$$

where, again, $\boldsymbol{x}^{i*}$ is the node on $\widehat{S}_\beta$, which is closest to $\boldsymbol{y} - \boldsymbol{m}$, and the symmetric second-rank tensor

$$\boldsymbol{\Pi}(\boldsymbol{y}, \beta) = - \sum_{\boldsymbol{x}^j \in \widehat{S}_\beta} [\eta \boldsymbol{G}_0(\boldsymbol{r}^j) + 2\boldsymbol{\tau}_0(\boldsymbol{r}^j) \cdot \boldsymbol{n}(\boldsymbol{x}^j)] \triangle S_j + \eta \int_{\widehat{S}_\beta} \boldsymbol{G}_0(\boldsymbol{r}) \mathrm{d}S_x, \tag{3.9}$$

with $\boldsymbol{r}^j = \boldsymbol{x}^j - \boldsymbol{y}$ and $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}$. The integral in (3.9) is handled analytically, which is possible for spheres (Section 5.1) and other canonic solid shapes [6]. The barrier Θ_2 is different from Θ_1, namely, for *all* nodes $\boldsymbol{y} \in \widehat{S}_\alpha$, $\Theta_2 = 1$ if the minimum node-to-node distance between $\widehat{S}_\alpha$ and the periodic image $\widehat{S}_\beta + \boldsymbol{m}$ of surface $\widehat{S}_\beta$ shifted by $\boldsymbol{m}$ is less than the threshold $h_0 (= 0.25\widehat{a})$, and $\Theta_2 = 0$ otherwise. For "self-interactions" (when $\alpha = \beta$ and $\boldsymbol{m} = \boldsymbol{0}$), $\boldsymbol{Q}(\boldsymbol{y})$ is added to (3.8) and $\boldsymbol{r}^j = \boldsymbol{0}$ is excluded from the summation (3.9). Compared to Θ_1, the barrier

Θ_2, as a function of $\boldsymbol{y}$, does not have discontinuity within a chosen observation surface $\widehat{S}_\alpha \ni \boldsymbol{y}$ (which we have found important for solid–solid contributions at high concentrations), but it increases the number of operations in (3.8). However, in the present case of stationary meshes on solid particles, the elements $\boldsymbol{\Pi}(\boldsymbol{y}-\boldsymbol{m},\beta)$ are time-independent and calculated only once (by the simplest point-to-point summations), so the additional term in (3.8) does not slow down dynamical simulations. The form (3.8), again, does not achieve complete desingularization (except for $\alpha=\beta$), but was sufficient, since the solid particles have no relative motion in the present problem, with no solid–solid lubrication.

3.3. Solid-to-drop contributions

Single-layer contributions from solids in (2.11), when $\boldsymbol{y}\in\widetilde{S}_\alpha$, are desingularized similar to (3.3), using the value $\boldsymbol{Q}^*$ of $\boldsymbol{Q}(\boldsymbol{x})$ at the surface point $\boldsymbol{x}_S^*\in\widehat{S}_\beta$ nearest to $\boldsymbol{y}-\boldsymbol{m}$, found by interpolation (cf. (3.4)). For double-layer contributions, however, a more elaborate approach is required, since drop–solid interactions (compared to drop–drop) are very lubrication-sensitive. Here, we follow our recent idea of *high-order* near-singularity subtraction [6] and construct a local linear approximation

$$\boldsymbol{Q}(\boldsymbol{x}^{i*})+\sum_{j\in\mathcal{A}_{i*}}\left[\boldsymbol{C}_{j,i*}\cdot(\boldsymbol{x}-\boldsymbol{x}^{i*})\right][\boldsymbol{Q}(\boldsymbol{x}^j)-\boldsymbol{Q}(\boldsymbol{x}^{i*})] \tag{3.10}$$

to $\boldsymbol{Q}(\boldsymbol{x})$ near $\boldsymbol{x}^{i*}$ on $\widehat{S}_\beta$ (with the same $\boldsymbol{C}$-coefficients as in (3.4), just calculated on the solid $\widehat{S}_\beta$), and subtract (3.10) from $\boldsymbol{Q}(\boldsymbol{x})$ in the double-layer terms to fully desingularize the integrands. For solid-to-drop contributions, most affected by numerical implementation, we employ a barrier Θ_3 analogous to Θ_2 with additional smoothing: for *all* nodes $\boldsymbol{y}\in\widehat{S}_\alpha$, $\Theta_3=1-\delta^2/h_0^2$, if the minimum distance δ between $\widetilde{S}_\alpha$ and $\widehat{S}_\beta+\boldsymbol{m}$ is less than the threshold $h_0(=0.25\widehat{a})$, and $\Theta_3=0$ otherwise; a more economical barrier Θ_1, or its smooth analogs, were not satisfactory for solid-to-drop contributions in the present problem. So, each integral (2.11) with $\boldsymbol{Q}(\boldsymbol{x})$ over a solid surface $\widehat{S}_\beta$, when $\boldsymbol{y}\in\widetilde{S}_\alpha$, is approximated as

$$\sum_{\boldsymbol{x}^j\in\widehat{S}_\beta}\boldsymbol{Q}(\boldsymbol{x}^j)\cdot[\eta\boldsymbol{G}(\boldsymbol{x}^j-\boldsymbol{y})+2\check{\boldsymbol{\tau}}(\boldsymbol{x}^j-\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{x}^j)]\triangle S_j$$
$$+\sum_{\boldsymbol{m}}\Theta_3\left\{\boldsymbol{Q}^*\cdot\boldsymbol{\Pi}(\boldsymbol{y}-\boldsymbol{m},\beta)+\sum_{j\in\mathcal{A}_{i*}}\boldsymbol{C}_{j,i*}\cdot\boldsymbol{\Gamma}(\boldsymbol{y}-\boldsymbol{m},\beta)\cdot[\boldsymbol{Q}(\boldsymbol{x}^j)-\boldsymbol{Q}(\boldsymbol{x}^{i*})]\right\}. \tag{3.11}$$

Here,

$$\boldsymbol{Q}^*=\boldsymbol{Q}(\boldsymbol{x}^{i*})+\sum_{j\in\mathcal{A}_{i*}}\left[\boldsymbol{C}_{j,i*}\cdot(\boldsymbol{y}-\boldsymbol{m}-\boldsymbol{x}^{i*})\right][\boldsymbol{Q}(\boldsymbol{x}^j)-\boldsymbol{Q}(\boldsymbol{x}^{i*})], \tag{3.12}$$

tensor $\boldsymbol{\Pi}$ has already appeared in (3.9), and an additional third-rank, fully symmetric tensor is

$$\boldsymbol{\Gamma}(\boldsymbol{y},\beta)=\frac{3}{2\pi}\left[-\sum_{\boldsymbol{x}^j\in\widehat{S}_\beta}\frac{[\boldsymbol{r}^j\cdot\boldsymbol{n}(\boldsymbol{x}^j)]\boldsymbol{r}^j\boldsymbol{r}^j\boldsymbol{r}^j}{|\boldsymbol{r}^j|^5}\triangle S_j+\int_{\widehat{S}_\beta}\frac{[\boldsymbol{r}\cdot\boldsymbol{n}(\boldsymbol{x})]\boldsymbol{rrr}}{|\boldsymbol{r}|^5}\mathrm{d}S_x\right]. \tag{3.13}$$

The added-back integral in (3.13) allows for analytical treatment for spheres (Section 5.1) and other canonic shapes [6]. The calculation of the subtraction tensors $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$ is taken out of iterations but, when done by the simplest point-to-point summations, would be too expensive (comparable in cost with the multipole-accelerated iterative solution). For this reason, we also employ multipole acceleration to calculate $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$, as described in Section 5.1.

The paper [6] gives reasons why we have to use, in the drop squeezing problem, different desingularization schemes for different types of interaction. In particular, high-order subtraction (3.11) could not be used for double-layer drop contributions because of the added-back integral in (3.13). On the other hand, the variational approach (3.5)–(3.7) is not powerful enough for handling solid-to-drop contributions, and (3.11)–(3.13) provide the only viable alternative.

4. Multipole acceleration of iterations

4.1. General scheme

Below we discuss economical calculation of the first sums in (3.5), (3.8) and (3.11). These operations, performed on each iteration, present the most computationally-extensive part, and using direct point-to-point summations to this end would be very prohibitive for all simulations of interest. Multipole acceleration tools developed for multidrop interactions with periodic boundaries [20–22] could be used herein. Those techniques, however, with one level of mesh-node decomposition and multipole expansions generated for individual surfaces (or the compact blocks that each elongated surface is divided into, Ref. [20]) were designed to be efficient for large systems of drops ($N \sim 10^2 - 10^3$) with moderate resolution ($N_\triangle \sim 10^3$ triangular boundary elements per surface). The efficiency of acceleration goes down for much higher resolutions and/or smaller systems. In fact, the partitioning of the Green function into the "near-field" and "far-field" parts in those algorithms requires the system size to be not too small (roughly, $N > 20 - 50$ at high concentrations); in particular, periodic systems ($N = 1$) cannot be handled. On the other hand, free-space close interactions of two-drops only with superhigh resolution ($N_\triangle \sim 10^5$, necessary for $Ca \ll 1$) were efficiently handled by a different multipole-accelerated code [23] with partitioning each drop surface into a large number of "patches."

In the present problem of emulsion squeezing through a granular material with high resistance, high surface resolution (at least $N_\triangle \sim 10^4$) is most important for successful simulations and severely limits the size ($\widehat{N}$, $\widetilde{N}$) of the systems that can be handled at present or in the near future. For this reason, we have combined the techniques of [20–23] to develop a new multipole-accelerated code with two levels of mesh-node decomposition, which is efficient at high surface resolution both for large and small systems, and has no lower limitation on the system size ($\widehat{N}$, $\widetilde{N}$). Besides, we have found considerable improvements to some relevant multipole tools of [20–23], especially for the important special case of spherical solid particles, and developed some new tools, as detailed below.

On the finest level of mesh decomposition, the collocation nodes $\boldsymbol{x}^j$ on each surface are grouped into a large number of non-overlapping sets called "patches" (Fig. 2). For each solid surface $\widehat{S}_\alpha$, a stationary, crude auxiliary mesh, almost uniform with $\widehat{M}_\triangle \gg 1$ triangles (and $\widehat{M}_\triangle/2 + 2$ vertices $\boldsymbol{z}^\gamma \in \widehat{S}_\alpha$) is first constructed, with associated Voronoi cells around each $\boldsymbol{z}^\gamma$; each cell is an intersection of typically six (rarely five) half-spaces. A patch $\mathcal{B}_\gamma$ simply consists of the nodes $\boldsymbol{x}^j$ of the main (almost uniform) mesh on $\widehat{S}_\alpha$, which are closer to $\boldsymbol{z}^\gamma$ than to any other crude mesh-node on $\widehat{S}_\alpha$. The same procedure is applied to each drop $\widetilde{S}_\alpha$ in the initial configuration when the drop is spherical, starting from a crude auxiliary mesh with $\widetilde{M}_\triangle \gg 1$ triangles and the initial nodes $\boldsymbol{x}^j \in \widetilde{S}_\alpha$ of the main mesh, both meshes being almost uniform. The initial partitioning of drop nodes $\boldsymbol{x}^j \in \widetilde{S}_\alpha$ into patches $\mathcal{B}_\gamma$ is kept unchanged in the swelling process (Section 5.3) and subsequent boundary-integral simulation, although drops deform and $\boldsymbol{x}^j$ move. These algorithms of grouping mesh-nodes into patches simplify

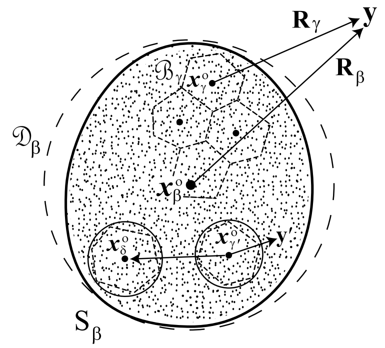


Fig. 2. Grouping of mesh-nodes (small size dots) into non-overlapping patches $\mathcal{B}_\gamma$. Lamb's singular series about the patch centers (e.g., $\boldsymbol{x}^0_\gamma$) are merged into cumulative singular series about the center $\boldsymbol{x}^0_\beta$ of the minimal shell $\mathcal{D}_\beta$ around S_β. For "sufficiently separated" patches $\mathcal{B}_{\gamma'}$ and $\mathcal{B}_\delta$, the $\mathcal{B}_{\gamma'} \to \mathcal{B}_\delta$ free-space contribution is handled by singular-to-regular reexpansion. Otherwise, for node $\boldsymbol{y}' \in S_\beta$, either Lamb's singular series about $\boldsymbol{x}^0_{\gamma'}$, or direct summation over $\mathcal{B}^0_{\gamma'}$ is used.

that of Ref. [23] (where partitioning was done dynamically at each time step). As in that paper, it is optimal to have 200–400 nodes $\boldsymbol{x}^j$ per patch for multipole acceleration. It would be relatively easy to adapt our stationary meshes on solids $\widehat{S}_\alpha$ to constriction regions, in which case the auxiliary mesh would also have to be non-uniform to keep the number of nodes per patch almost constant, for maximum efficiency of multipole acceleration. We have found adaptive meshes on solids, however, to be disadvantageous in the present problem, since they reduce the global accuracy.

A minimal spherical shell $\mathcal{D}_\gamma$ with center $\boldsymbol{x}^0_\gamma$ and radius d^0_γ is constructed around each $\mathcal{B}_\gamma$, both for solids $\widehat{S}_\alpha$ and (dynamically) for drops; a simple stochastic algorithm [20] provides sufficient accuracy for this operation. Using (approximately) minimal shells is also important for the efficiency of multipole acceleration. For the special case of spherical solids $\widehat{S}_\alpha$, the center $\boldsymbol{x}^0_\gamma$ of the shell $\mathcal{D}_\gamma$ around $\mathcal{B}_\gamma \subset \widehat{S}_\alpha$ is constrained to be on the surface $\widehat{S}_\alpha$, which increases the minimal radius d^0_γ only slightly but allows much faster generation of the multipole expansion (below) for the patch $\mathcal{B}_\gamma$ than by a general algorithm. Minimal spherical shells $\mathcal{D}_\alpha$ with centers $\boldsymbol{x}^0_\alpha$ and radii d^0_α are also constructed around each solid $\widehat{S}_\alpha$ and drop $\widetilde{S}_\alpha$ (for drops, we are not using the slicing [20] into compact blocks); $\mathcal{D}_\alpha = \widehat{S}_\alpha$ for a sphere $\widehat{S}_\alpha$. To avoid confusion, symbols γ, δ are reserved for patches and associated quantities, while α, β are used for the entire drop and solid surfaces; unless otherwise stated, S_α stands for $\widehat{S}_\alpha$ or $\widetilde{S}_\alpha$.

The free-space contribution of each patch $\mathcal{B}_\gamma$ to (3.5), (3.8) or (3.11), i.e.,

$$(\lambda-1)\sum_{\boldsymbol{x}^j\in\mathcal{B}_\gamma}\boldsymbol{Q}(\boldsymbol{x}^j)\cdot\boldsymbol{\tau}_0(\boldsymbol{x}^j-\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{x}^j)\triangle S_j \quad \text{or} \quad \sum_{\boldsymbol{x}^j\in\mathcal{B}_\gamma}\boldsymbol{Q}(\boldsymbol{x}^j)\cdot[\eta\boldsymbol{G}_0(\boldsymbol{x}^j-\boldsymbol{y})+2\boldsymbol{\tau}_0(\boldsymbol{x}^j-\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{x}^j)]\triangle S_j, \tag{4.1}$$

is expanded in Lamb's singular series:

$$\sum_{v=1}^{\infty}\left[\nabla\times\left(\boldsymbol{R}_\gamma\chi^{(\gamma)}_{-(v+1)}\right)+\nabla\Phi^{(\gamma)}_{-(v+1)}-\frac{(v-2)R_\gamma^2\nabla p^{(\gamma)}_{-(v+1)}}{2v(2v-1)}+\frac{(v+1)p^{(\gamma)}_{-(v+1)}\boldsymbol{R}_\gamma}{v(2v-1)}\right]+\nabla\Phi^{(\gamma)}_{-1}, \tag{4.2}$$

convergent for $|\boldsymbol{R}_\gamma| > d^0_\gamma$, where $\boldsymbol{R}_\gamma = \boldsymbol{y}-\boldsymbol{x}^0_\gamma$, and $p^{(\gamma)}_{-(v+1)}(\boldsymbol{R}_\gamma)$, $\Phi^{(\gamma)}_{-(v+1)}(\boldsymbol{R}_\gamma)$, $\chi^{(\gamma)}_{-(v+1)}(\boldsymbol{R}_\gamma)$ are solid spherical harmonics of order $-(v+1)$. These harmonics are generated to a sufficient order ($v \leqslant k_0$, with typically $k_0 = 25$–30) by the new algorithms of Section 4.2; these algorithms turn out to be considerably more economical than our initial scheme (Section 3.2 of Ref. [20]) for calculating singular moments. Generation of harmonics $p^{(\gamma)}_{-(v+1)}$, $\Phi^{(\gamma)}_{-(v+1)}$, $\chi^{(\gamma)}_{-(v+1)}$ is particularly fast for a patch $\mathcal{B}_\gamma$ on a spherical surface by a special technique.

The next step is to translate the expansions (4.2) for all $\mathcal{B}_\gamma \subset S_\beta$ to the center $\boldsymbol{x}^0_\beta$ of the shell $\mathcal{D}_\beta$ in order to obtain a cumulative Lamb's singular series

$$\sum_{v=1}^{\infty}\left[\nabla\times\left(\boldsymbol{R}_\beta\chi^{(\beta)}_{-(v+1)}\right)+\nabla\Phi^{(\beta)}_{-(v+1)}-\frac{(v-2)R_\beta^2\nabla p^{(\beta)}_{-(v+1)}}{2v(2v-1)}+\frac{(v+1)p^{(\beta)}_{-(v+1)}\boldsymbol{R}_\beta}{v(2v-1)}\right]+\nabla\Phi^{(\beta)}_{-1} \tag{4.3}$$

for

$$(\lambda-1)\sum_{\boldsymbol{x}^j\in\widetilde{S}_\beta}\boldsymbol{Q}(\boldsymbol{x}^j)\cdot\boldsymbol{\tau}_0(\boldsymbol{x}^j-\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{x}^j)\triangle S_j \quad \text{or} \quad \sum_{\boldsymbol{x}^j\in\widehat{S}_\beta}\boldsymbol{Q}(\boldsymbol{x}^j)\cdot[\eta\boldsymbol{G}_0(\boldsymbol{x}^j-\boldsymbol{y})+2\boldsymbol{\tau}_0(\boldsymbol{x}^j-\boldsymbol{y})\cdot\boldsymbol{n}(\boldsymbol{x}^j)]\triangle S_j, \tag{4.4}$$

convergent for $|\boldsymbol{R}_\beta| > d^0_\beta$, with $\boldsymbol{R}_\beta = \boldsymbol{y}-\boldsymbol{x}^0_\beta$ and $v \leqslant k_0$. This operation ("merging of singularities"), which is absent from our algorithms [20–23], is most efficiently done through rotational transformations of spherical harmonics and described in Section 4.3; the cost of translation for one patch $\mathcal{B}_\gamma$ becomes $\mathrm{O}(k_0^3)$.

The periodic Green function $\boldsymbol{G}$ and the corresponding stresslet $\breve{\boldsymbol{\tau}}$ are split into the "far-field" $(\boldsymbol{G}_1, \boldsymbol{\tau}_1)$ and "near-field" parts:

$$\boldsymbol{G}(\boldsymbol{r}) = \sum_{|\boldsymbol{m}|\leqslant m_0}\boldsymbol{G}_0(\boldsymbol{r}+\boldsymbol{m})+\boldsymbol{G}_1(\boldsymbol{r}) \quad , \quad \breve{\boldsymbol{\tau}}(\boldsymbol{r}) = \sum_{|\boldsymbol{m}|\leqslant m_0}\boldsymbol{\tau}_0(\boldsymbol{r}+\boldsymbol{m})+\boldsymbol{\tau}_1(\boldsymbol{r}), \tag{4.5}$$

with integer vectors $\boldsymbol{m}$. Algorithms [20–22] employed $m_0 = 0$ suitable (and optimal) for sufficiently large systems (roughly $N \geqslant \mathrm{O}(10^2)$). In the present work, we use $m_0 = 2$ to move singularities of $\boldsymbol{G}_1(\boldsymbol{r})$ and $\boldsymbol{\tau}_1(\boldsymbol{r})$ much farther away from the origin; $m_0 = 3$ would be beneficial for the smallest systems ($\widetilde{N}$, $\widehat{N} \sim 1$).

Now, to calculate the contribution of a single surface S_β to the sums (3.5), (3.8) or (3.11) for $\boldsymbol{y} \in S_\alpha$, the surface S_β is temporarily shifted by an integer vector $\boldsymbol{m}_{\alpha\beta}$ to a new position $S_\beta^{\min} = S_\beta + \boldsymbol{m}_{\alpha\beta}$ to minimize the center-to-center distance $\left\|\boldsymbol{x}_\beta^{0,\min} - \boldsymbol{x}_\alpha^0\right\|$ (with $\boldsymbol{x}_\beta^{0,\min} = \boldsymbol{x}_\beta^0 + \boldsymbol{m}_{\alpha\beta}$) between the corresponding shells; index min will supplement quantities related to $S_\beta^{\min}$. Using (4.5), the contribution of S_β can be written as

$$\sum_{|\boldsymbol{m}|\leqslant m_0} \sum_{\boldsymbol{x}^j \in S_\beta^{\min}+\boldsymbol{m}} [\ldots]_0 + \sum_{\boldsymbol{x}^j \in S_\beta^{\min}} [\ldots]_1, \tag{4.6}$$

where, for brevity, $[\ldots]_0$ is either $\boldsymbol{Q}(\boldsymbol{x}^j) \cdot \boldsymbol{\tau}_0(\boldsymbol{x}^j - \boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{x}^j)\triangle S_j$ or $\boldsymbol{Q}(\boldsymbol{x}^j) \cdot [\eta \boldsymbol{G}_0(\boldsymbol{x}^j - \boldsymbol{y}) + 2\boldsymbol{\tau}_0(\boldsymbol{x}^j - \boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{x}^j)]\triangle S_j$ (assuming periodic continuation for $\boldsymbol{Q}$, $\boldsymbol{n}$, $\triangle S$); analogous for $[\ldots]_1$, with $\boldsymbol{\tau}_1$, $\boldsymbol{G}_1$ instead of $\boldsymbol{\tau}_0$, $\boldsymbol{G}_0$. The far-field contribution to (4.6) is efficiently calculated by a special form of Taylor double series for $\boldsymbol{\tau}_1(\boldsymbol{x}^j - \boldsymbol{y})$ and/or $\boldsymbol{G}_1(\boldsymbol{x}^j - \boldsymbol{y})$ in powers of $\boldsymbol{x}^j - \boldsymbol{x}_\beta^{0,\min}$ and $\boldsymbol{y} - \boldsymbol{x}_\alpha^0$ (Section 4.4); for $m_0 \geqslant 2$, convergence is very fast, even for small systems, and just a few terms suffice.

If the shell $\mathcal{D}_\beta^{\min} + \boldsymbol{m}$ around $S_\beta^{\min} + \boldsymbol{m}$ and shell $\mathcal{D}_\alpha$ do not overlap, the (free-space) contribution of $S_\beta^{\min} + \boldsymbol{m}$ to (4.6) can be evaluated at $\boldsymbol{y} \in S_\alpha$ by first reexpanding the singular series (4.3) (written for $\boldsymbol{y} - \boldsymbol{x}_\beta^{0,\min} - \boldsymbol{m}$ instead of $\boldsymbol{R}_\beta$) into Lamb's regular series

$$\sum_{n=1}^{\infty} \left[\nabla \times (\boldsymbol{R}_\alpha \chi_n) + \nabla \Phi_n + \frac{(n+3)R_\alpha^2 \nabla p_n}{2(n+1)(2n+3)} - \frac{n p_n \boldsymbol{R}_\alpha}{(n+1)(2n+3)} \right], \tag{4.7}$$

where $\boldsymbol{R}_\alpha = \boldsymbol{y} - \boldsymbol{x}_\alpha^0$ and $p_n(\boldsymbol{R}_\alpha)$, $\Phi_n(\boldsymbol{R}_\alpha)$ and $\chi_n(\boldsymbol{R}_\alpha)$ are solid harmonics of order n. However, only drop images $S_\beta^{\min} + \boldsymbol{m}$ "sufficiently separated" from S_α are included in this operation, so that the shells $\mathcal{D}_\beta^{\min} + \boldsymbol{m}$ and $\mathcal{D}_\alpha$ have enough clearance for the reexpansion to converge within prescribed tolerance and threshold $(v, n \leqslant k_0)$. The free-space contributions of all such images $S_\beta^{\min} + \boldsymbol{m}$ to (4.6) are summed over all $\boldsymbol{m}$ and all β, using a fast, rotation-based reexpansion algorithm (Section 3.3 of Ref. [20]). The cumulative series (4.7) is then transformed to a more efficient form (see Eq. (3.19) of Ref. [20])

$$\frac{1}{2}\boldsymbol{R}_\alpha \sum_{n=1}^{\infty} p_n(\boldsymbol{R}_\alpha) + \sum_{n=0}^{\infty} \boldsymbol{h}_n(\boldsymbol{R}_\alpha), \tag{4.8}$$

where $\boldsymbol{h}_n(\boldsymbol{R}_\alpha)$ are solid harmonics of order n with vector coefficients, and calculated pointwise for all nodes $\boldsymbol{y} \in S_\alpha$.

Some free-space contributions in (4.6) are still not accounted for by the above scheme, in particular, "self-interactions" ($S_\alpha = S_\beta = S_\beta^{\min}$ and $\boldsymbol{m} = \boldsymbol{0}$). In this case, singular expansions (4.2) for all patches $\mathcal{B}_\gamma \subset S_\alpha$ "sufficiently separated" from a given patch $\mathcal{B}_\delta \subset S_\alpha$ are reexpanded into Lamb's regular forms and summed up, to form cumulative expansions of the type (4.7) and (4.8) for the patch $\mathcal{B}_\delta$ (with $\boldsymbol{R}_\delta = \boldsymbol{y} - \boldsymbol{x}_\delta^0$ in place of $\boldsymbol{R}_\alpha$) and calculate them pointwise for all mesh-nodes $\boldsymbol{y} \in \mathcal{B}_\delta$. If a patch $\mathcal{B}_\gamma$ is not "sufficiently separated" from $\mathcal{B}_\delta$, then the contribution from $\mathcal{B}_\gamma$ to $\boldsymbol{y}$ for $\boldsymbol{y} \in \mathcal{B}_\delta$ is calculated either by direct summation (4.1), or by Lamb's singular series (4.2) converted into a more efficient form (Eq. (3.21) of Ref. [20]):

$$\frac{1}{2}\boldsymbol{R}_\gamma \sum_{v=1}^{\infty} p_{-(v+1)}(\boldsymbol{R}_\gamma) + \sum_{v=0}^{\infty} \boldsymbol{h}_{-(v+1)}(\boldsymbol{R}_\gamma), \tag{4.9}$$

where $\boldsymbol{h}_{-(v+1)}$ are solid harmonics of order $-(v+1)$. The second strategy is chosen if (i) (4.9) is more efficient for point $\boldsymbol{y}$ and (ii) the truncation bound on v in (4.9) to reach necessary convergence does not exceed the threshold k_0. When $\mathcal{B}_\delta = \mathcal{B}_\gamma$, $\boldsymbol{x}^j = \boldsymbol{y}$ is excluded from direct summation.

Finally, the last remaining free-space contributions in (4.6), namely, when $S_\beta^{\min} + \boldsymbol{m}$ is different, but not "sufficiently separated" from S_α, are handled in a simpler manner (Fig. 3). Lamb's singular series (4.3) for the entire surface $S_\beta^{\min} + \boldsymbol{m}$, converted to the more efficient form (4.9) (with $\boldsymbol{y} - \boldsymbol{x}_\beta^{0,\min} - \boldsymbol{m}$ instead of $\boldsymbol{R}_\gamma$) is used for $\boldsymbol{y} \in S_\alpha$, if the truncation bound on v necessary for convergence does not exceed k_0. Otherwise, we consider the free-space contribution of $S_\beta^{\min} + \boldsymbol{m}$ to (4.6) as a sum from individual patches and employ for each patch either the multipole expansion (4.9) (only if the necessary truncation bound on v is within k_0) or direct summation (4.1), depending on which is more optimal.

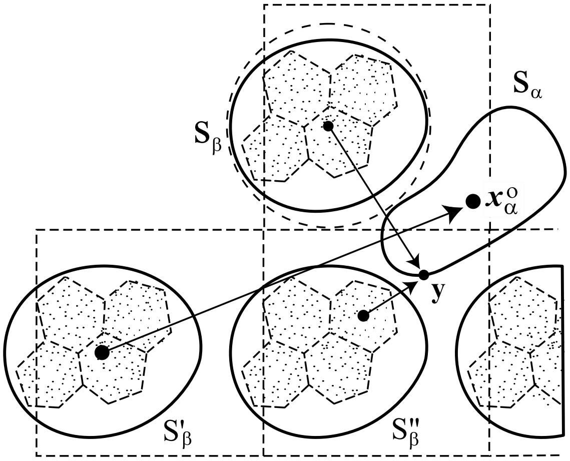


Fig. 3. Free-space interaction of periodic images (S_β, S'_β, S''_β, etc.) with another surface S_α. For S'_β "sufficiently separated" from S_α, the $S'_\beta \to S_\alpha$ contribution is handled by singular-to-regular center-to-center reexpansion. For node $\boldsymbol{y}$ "well outside" the shell $\mathcal{D}_\beta$ around S_β, Lamb's singular series about the shell center is used for $S_\beta \to \boldsymbol{y}$ contribution. The contribution $S''_\beta \to \boldsymbol{y}$ is handled as a sum from individual patches, with either Lamb's singular series or direct summation used for each patch.

A rather general description of our multipole acceleration scheme above is complemented in the following subsections by the details of multipole operations. In Section 5.2, we return to other, non-multipole features of the whole algorithm.

4.2. Generation of Lamb's singular series for a patch

To expand the patch contribution (4.1) in Lamb's singular form (4.2), in principle, the techniques of Ref. [20] (Section 3.2 therein) could be used based on rotational transformations of spherical harmonics. In the present work, however, we found a simpler and considerably more economical way, without the use of rotational transformations.

We start from a special form [20] of the Taylor series for a harmonic function:

$$f(\boldsymbol{x})=\sum_{\nu=0}^{\infty}\sum_{\mu=-\nu}^{\nu}\partial_{\nu,\mu}f(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}^0} Z_{\nu,\mu}\left(\boldsymbol{x}-\boldsymbol{x}^0\right),\quad \partial_{\nu,\mu}=(D_1-iD_2)^{\mu}D_3^{\nu-|\mu|}, \tag{4.10}$$

where $D_k=\partial/\partial x_k$ is the Cartesian partial derivative, $(D_1-iD_2)^{\mu}=(-1)^{\mu}(D_1+iD_2)^{-\mu}$ for $\mu<0$, $i=\sqrt{-1}$, and $Z_{\nu,\mu}$ are related to standard spherical harmonics $Y_{\nu,\mu}$:

$$\begin{aligned}
Z_{\nu,\mu}(\boldsymbol{r})&=\frac{2\pi^{1/2}r^{\nu}Y_{\nu,\mu}(\boldsymbol{r})}{[(2\nu+1)(\nu-\mu)!(\nu+\mu)!]^{1/2}},\\
Y_{\nu,\mu}(\boldsymbol{r})&=\left[\frac{(2\nu+1)(\nu-\mu)!}{4\pi(\nu+\mu)!}\right]^{1/2}P_{\nu}^{\mu}(\cos\theta)e^{i\mu\varphi}\qquad(\mu\geqslant 0),\\
Y_{\nu,\mu}(\boldsymbol{r})&=(-1)^{\mu}\bar{Y}_{\nu,-\mu}(\boldsymbol{r})\qquad(\mu<0)
\end{aligned}\tag{4.11}$$

for a vector $\boldsymbol{r}=(r\sin\theta\cos\varphi, r\sin\theta\sin\varphi, r\cos\theta)$, P_{ν}^{μ} is the associated Legendre function (in the definition of [33]) and the overbar stands for complex conjugation. Relation (4.10) can be shown to be equivalent to (27) of Sangani and Mo [11].

Let $\boldsymbol{\mathcal{P}}_0(\boldsymbol{\xi})=\left(\mathcal{P}_0^1,\mathcal{P}_0^2,\mathcal{P}_0^3\right)=\frac{1}{4\pi}\nabla\frac{1}{|\boldsymbol{\xi}|}$ be the vector of pressures for the free-space Green tensor (3.2) $\boldsymbol{G}_0(\boldsymbol{\xi})$. An auxiliary function

$$g_k^{\ell}(\boldsymbol{x},\boldsymbol{y})=G_0^{k\ell}(\boldsymbol{x}-\boldsymbol{y})-\frac{1}{2}\left(x-x_{\gamma}^{0}\right)_k\mathcal{P}_0^{\ell}(\boldsymbol{x}-\boldsymbol{y}) \tag{4.12}$$

is a harmonic function of $\boldsymbol{x}$, as well as $\boldsymbol{\mathcal{P}}_0(\boldsymbol{x}-\boldsymbol{y})$, so both can be expanded in the vicinity of $\boldsymbol{x}=\boldsymbol{x}_\gamma^0$ using (4.10). Substituting (4.12) for $\boldsymbol{x}=\boldsymbol{x}^j$ into (4.1) gives the single-layer free-space contribution in the form

$$\sum_{\boldsymbol{x}^j\in\mathcal{B}_\gamma} Q_k(\boldsymbol{x}^j)G_0^{k\ell}(\boldsymbol{x}^j-\boldsymbol{y})\triangle S_j=\sum_{v=0}^{\infty}\sum_{\mu=-v}^{v}\left[\sum_{\boldsymbol{x}^j\in\mathcal{B}_\gamma}\triangle S_j Z_{v,\mu}\left(\boldsymbol{x}^j-\boldsymbol{x}_\gamma^0\right)\boldsymbol{Q}(\boldsymbol{x}^j)\right]\cdot\partial_{v,\mu}\boldsymbol{g}^\ell(\boldsymbol{x},\boldsymbol{y})\Bigg|_{\boldsymbol{x}=\boldsymbol{x}_\gamma^0}$$
$$+\frac{1}{2}\sum_{v=0}^{\infty}\sum_{\mu=-v}^{v}\left[\sum_{\boldsymbol{x}^j\in\mathcal{B}_\gamma}\triangle S_j Z_{v,\mu}\left(\boldsymbol{x}^j-\boldsymbol{x}_\gamma^0\right)\boldsymbol{Q}(\boldsymbol{x}^j)\cdot\left(\boldsymbol{x}^j-\boldsymbol{x}_\gamma^0\right)\right]\partial_{v,\mu}\mathcal{P}_0^\ell(\boldsymbol{x}-\boldsymbol{y})\Bigg|_{\boldsymbol{x}=\boldsymbol{x}_\gamma^0}. \tag{4.13}$$

Here, the differential operations are with respect to $\boldsymbol{x}$, and $\boldsymbol{g}^\ell=\left(g_1^\ell,g_2^\ell,g_3^\ell\right)$. The pressure for the stresslet flow $\left(\tau_0^{ks1},\tau_0^{ks2},\tau_0^{ks3}\right)(\boldsymbol{\xi})$ is $\check{\mathcal{P}}_0^{ks}(\boldsymbol{\xi})=2\partial\mathcal{P}_0^k(\boldsymbol{\xi})/\partial\xi_s$. An auxiliary function

$$t_{sk}^\ell(\boldsymbol{x},\boldsymbol{y})=\tau_0^{\ell sk}(\boldsymbol{x}-\boldsymbol{y})-\frac{1}{4}\left[\left(x-x_\gamma^0\right)_k\check{\mathcal{P}}_0^{\ell s}(\boldsymbol{x}-\boldsymbol{y})+\left(x-x_\gamma^0\right)_s\check{\mathcal{P}}_0^{\ell k}(\boldsymbol{x}-\boldsymbol{y})\right] \tag{4.14}$$

is a harmonic function of $\boldsymbol{x}$, so is $\check{\mathcal{P}}_0^{ks}(\boldsymbol{x}-\boldsymbol{y})$, and both can be expanded near $\boldsymbol{x}=\boldsymbol{x}_\gamma^0$ using (4.10). Substituting (4.14) for $\boldsymbol{x}=\boldsymbol{x}^j$ into (4.1) yields the double-layer contribution

$$\sum_{\boldsymbol{x}^j\in\mathcal{B}_\gamma} Q_s(\boldsymbol{x}^j)\tau_0^{\ell sk}(\boldsymbol{x}^j-\boldsymbol{y})W_k(\boldsymbol{x}^j)=\sum_{v=0}^{\infty}\sum_{\mu=-v}^{v}\check{E}_{v,\mu,k,s}^{(\gamma)}\partial_{v,\mu}t_{sk}^\ell(\boldsymbol{x},\boldsymbol{y})\Bigg|_{\boldsymbol{x}=\boldsymbol{x}_\gamma^0}+\sum_{v=0}^{\infty}\sum_{\mu=-v}^{v}\check{\mathcal{D}}_{v,\mu,k}^{(\gamma)}\partial_{v,\mu}\check{\mathcal{P}}_0^{\ell k}(\boldsymbol{x},\boldsymbol{y})\Bigg|_{\boldsymbol{x}=\boldsymbol{x}_\gamma^0}. \tag{4.15}$$

The patch double-layer moments are defined as

$$\begin{aligned}
\check{E}_{v,\mu,k,s}^{(\gamma)}&=\sum_{\boldsymbol{x}^j\in\mathcal{B}_\gamma}W_{(s}Q_{k)}(\boldsymbol{x}^j)Z_{v,\mu}\left(\boldsymbol{x}^j-\boldsymbol{x}_\gamma^0\right),\\
\check{\mathcal{D}}_{v,\mu,k}^{(\gamma)}&=\frac{1}{4}\sum_{\boldsymbol{x}^j\in\mathcal{B}_\gamma}Z_{v,\mu}\left(\boldsymbol{x}^j-\boldsymbol{x}_\gamma^0\right)[Q_k(\boldsymbol{x}^j)\boldsymbol{W}(\boldsymbol{x}^j)+W_k(\boldsymbol{x}^j)\boldsymbol{Q}(\boldsymbol{x}^j)]\cdot\left(\boldsymbol{x}^j-\boldsymbol{x}_\gamma^0\right),
\end{aligned} \tag{4.16}$$

where $\boldsymbol{W}(\boldsymbol{x}^j)=(\lambda-1)\boldsymbol{n}(\boldsymbol{x}^j)\triangle S_j$ on a drop and $\boldsymbol{W}(\boldsymbol{x}^j)=2\boldsymbol{n}(\boldsymbol{x}^j)\triangle S_j$ on a solid surface, respectively, and $W_{(s}Q_{k)}=\frac{1}{2}(W_sQ_k+W_kQ_s)$ is the symmetrization in indices s and k.

The next step is to transform (4.15) to a form containing only $\partial_{v,\mu}\boldsymbol{g}^\ell$ and $\partial_{v,\mu}\mathcal{P}_0^\ell$. Since $\boldsymbol{\mathcal{P}}_0(\boldsymbol{\xi})$ is harmonic, the following simple algebra applies:

$$\begin{aligned}
&\partial_{v,\mu}\check{\mathcal{P}}_0^{\ell 1}=\left(\partial_{v+1,\mu+1}-\partial_{v+1,\mu-1}\right)\mathcal{P}_0^\ell,\quad \partial_{v,\mu}\check{\mathcal{P}}_0^{\ell 2}=i\left(\partial_{v+1,\mu+1}+\partial_{v+1,\mu-1}\right)\mathcal{P}_0^\ell,\\
&\partial_{v,\mu}\check{\mathcal{P}}_0^{\ell 3}=2\partial_{v+1,\mu}\mathcal{P}_0^\ell.
\end{aligned} \tag{4.17}$$

For t_{sk}^ℓ, we note that $t_{sk}^\ell(\boldsymbol{x},\boldsymbol{y})=D_sg_k^\ell(\boldsymbol{x},\boldsymbol{y})+D_kg_s^\ell(\boldsymbol{x},\boldsymbol{y})$, the derivatives taken with respect to $\boldsymbol{x}$, so $\partial_{v,\mu}t_{sk}^\ell$ can be expressed via $\partial_{v+1,\mu'}\boldsymbol{g}^\ell$ similar to (4.17). As a result, the RHS of (4.15) takes the form

$$\sum_{v}\sum_{\mu=-v}^{v}\left[\boldsymbol{E}_{v,\mu}^{(\gamma)}\cdot\partial_{v,\mu}\boldsymbol{g}^\ell(\boldsymbol{x},\boldsymbol{y})+\mathcal{D}_{v,\mu}^{(\gamma)}\partial_{v,\mu}\mathcal{P}_0^\ell(\boldsymbol{x}-\boldsymbol{y})\right]_{\boldsymbol{x}=\boldsymbol{x}_\gamma^0}, \tag{4.18}$$

with

$$\begin{aligned}
\boldsymbol{E}_{v,\mu}^{(\gamma)}&=\left(E_{v,\mu,1}^{(\gamma)},E_{v,\mu,2}^{(\gamma)},E_{v,\mu,3}^{(\gamma)}\right),\\
E_{v,\mu,k}^{(\gamma)}&=\check{E}_{v-1,\mu-1,k,1}^{(\gamma)}+i\check{E}_{v-1,\mu-1,k,2}^{(\gamma)}-\check{E}_{v-1,\mu+1,k,1}^{(\gamma)}+i\check{E}_{v-1,\mu+1,k,2}^{(\gamma)}+2\check{E}_{v-1,\mu,k,3}^{(\gamma)},\\
\mathcal{D}_{v,\mu}^{(\gamma)}&=\check{\mathcal{D}}_{v-1,\mu-1,1}^{(\gamma)}+i\check{\mathcal{D}}_{v-1,\mu-1,2}^{(\gamma)}-\check{\mathcal{D}}_{v-1,\mu+1,1}^{(\gamma)}+i\check{\mathcal{D}}_{v-1,\mu+1,2}^{(\gamma)}+2\check{\mathcal{D}}_{v-1,\mu,3}^{(\gamma)}
\end{aligned} \tag{4.19}$$

(assuming that $\check{\mathcal{D}}_{v,\mu,k}^{(\gamma)}$ and $\check{E}_{v,\mu,k,s}^{(\gamma)}$ are zero for $|\mu|>v$).

For a patch on a solid surface, single- (4.13) and double-layer (4.18) contributions are now combined. The problem is always reduced, therefore, to generating Lamb's singular series for (4.18). Using the explicit expressions for $\boldsymbol{g}^\ell$ and $\boldsymbol{\mathcal{P}}_0$, the vector form for (4.18) ($\ell=1,2,3$) can be written as

$$\boldsymbol{\Psi}(\boldsymbol{y}) = \sum_{\nu=0}^{\infty}\sum_{\mu=-\nu}^{\nu}\frac{(-1)^{\nu-1}}{8\pi}\left[\boldsymbol{E}^{(\gamma)}_{\nu,\mu} + \left(2D^{(\gamma)}_{\nu,\mu} - \boldsymbol{E}^{(\gamma)}_{\nu,\mu}\cdot\boldsymbol{R}_\gamma\right)\nabla^y\right]\partial^y_{\nu,\mu}\frac{1}{R_\gamma}. \tag{4.20}$$

Index y denotes differential operations with respect to $\boldsymbol{y} = \boldsymbol{x}^0_\gamma + \boldsymbol{R}_\gamma$. From Maxwell's relation [34],

$$\partial^y_{\nu,\mu}\frac{1}{R_\gamma} = (-1)^{\nu-\mu}\left[\frac{4\pi(\nu-\mu)!(\nu+\mu)!}{(2\nu+1)}\right]^{1/2}\frac{Y_{\nu,-\mu}(\boldsymbol{R}_\gamma)}{R_\gamma^{\nu+1}}. \tag{4.21}$$

It is now easy to expand the Stokes flow $\boldsymbol{\Psi}(\boldsymbol{y})$ in the form (4.2) and calculate the harmonics

$$\begin{aligned}
p^{(\gamma)}_{-(\nu+1)}(\boldsymbol{R}_\gamma) &= \sum_{m=-\nu}^{\nu} A^{(\gamma)}_{-(\nu+1),m}\left(\frac{d^0_\gamma}{R_\gamma}\right)^{\nu+1} Y_{\nu,m}(\boldsymbol{R}_\gamma),\\
\Phi^{(\gamma)}_{-(\nu+1)}(\boldsymbol{R}_\gamma) &= \sum_{m=-\nu}^{\nu} B^{(\gamma)}_{-(\nu+1),m}\left(\frac{d^0_\gamma}{R_\gamma}\right)^{\nu+1} Y_{\nu,m}(\boldsymbol{R}_\gamma),\\
\chi^{(\gamma)}_{-(\nu+1)}(\boldsymbol{R}_\gamma) &= \sum_{m=-\nu}^{\nu} C^{(\gamma)}_{-(\nu+1),m}\left(\frac{d^0_\gamma}{R_\gamma}\right)^{\nu+1} Y_{\nu,m}(\boldsymbol{R}_\gamma)
\end{aligned} \tag{4.22}$$

(assuming $p^{(\gamma)}_{-1} = \chi^{(\gamma)}_{-1} = 0$). Indeed, from (4.21),

$$\boldsymbol{\Psi}(\boldsymbol{y})\cdot\boldsymbol{R}_\gamma = \sum_{\nu=0}^{\infty}\sum_{\mu=-\nu}^{\nu}\frac{(-1)^{\nu-1}}{8\pi}\left[(\nu+2)\boldsymbol{E}^{(\gamma)}_{\nu,\mu}\cdot\boldsymbol{R}_\gamma - 2(\nu+1)D^{(\gamma)}_{\nu,\mu}\right]\partial^y_{\nu,\mu}\frac{1}{R_\gamma}. \tag{4.23}$$

Substituting (4.21) and standard recurrent relations

$$\begin{aligned}
(R_1 + iR_2)Y_{n,m}(\boldsymbol{R}) &= R\left[f^1_{n,m}Y^{m+1}_{n+1}(\boldsymbol{R}) + f^2_{n,m}Y^{m+1}_{n-1}(\boldsymbol{R})\right],\\
(R_1 - iR_2)Y_{n,m}(\boldsymbol{R}) &= R\left[f^3_{n,m}Y^{m-1}_{n+1}(\boldsymbol{R}) + f^4_{n,m}Y^{m-1}_{n-1}(\boldsymbol{R})\right],\\
R_3Y_{nm}(\boldsymbol{R}) &= R\left[f^5_{n,m}Y^{m}_{n+1}(\boldsymbol{R}) + f^6_{n,m}Y^{m}_{n-1}(\boldsymbol{R})\right],
\end{aligned} \tag{4.24}$$

with

$$\begin{aligned}
&f^1_{n,m} = \left[\frac{(n+m+1)(n+m+2)}{(2n+1)(2n+3)}\right]^{1/2}, \quad f^2_{n,m} = -\left[\frac{(n-m-1)(n-m)}{(2n+1)(2n-1)}\right]^{1/2},\\
&f^3_{n,m} = -f^1_{n,-m}, \quad f^4_{n,m} = -f^2_{n,-m},\\
&f^5_{n,m} = \left[\frac{(n-m+1)(n+m+1)}{(2n+1)(2n+3)}\right]^{1/2}, \quad f^6_{n,m} = \left[\frac{(n-m)(n+m)}{(2n-1)(2n+1)}\right]^{1/2}
\end{aligned} \tag{4.25}$$

into (4.24) and comparing the result with the general formula [35]

$$\boldsymbol{\Psi}(\boldsymbol{y})\cdot\boldsymbol{R}_\gamma = \sum_{\nu=0}^{\infty}\left[\frac{(\nu+1)}{2(2\nu-1)}R^2_\gamma p^{(\gamma)}_{-(\nu+1)} - (\nu+1)\Phi^{(\gamma)}_{-(\nu+1)}\right] \tag{4.26}$$

yields complex A- and B-coefficients in (4.22).

It easily follows from (4.20) that

$$[\nabla^y\times\boldsymbol{\Psi}(\boldsymbol{y})]\cdot\boldsymbol{R}_\gamma = \sum_{\nu=1}^{\infty}\sum_{\mu=-\nu}^{\nu}\frac{(-1)^{\nu-1}}{4\pi}\boldsymbol{E}^{(\gamma)}_{\nu,\mu}\cdot\left[\boldsymbol{R}_\gamma\times\nabla^y\partial^y_{\nu,\mu}\frac{1}{R_\gamma}\right]. \tag{4.27}$$

Substituting (4.21) and recurrent relations

$$\begin{aligned}
(R_2D_3 - R_3D_2)\frac{Y_{n,m}(\boldsymbol{R})}{R^{n+1}} &= -\frac{i}{2R^{n+1}}\left\{[(n-m)(n+m+1)]^{1/2}Y_{n,m+1}(\boldsymbol{R}) + [(n-m+1)(n+m)]^{1/2}Y_{n,m-1}(\boldsymbol{R})\right\},\\
(R_3D_1 - R_1D_3)\frac{Y_{n,m}(\boldsymbol{R})}{R^{n+1}} &= \frac{1}{2R^{n+1}}\left\{-[(n-m)(n+m+1)]^{1/2}Y_{n,m+1}(\boldsymbol{R}) + [(n-m+1)(n+m)]^{1/2}Y_{n,m-1}(\boldsymbol{R})\right\},\\
(R_1D_2 - R_2D_1)\frac{Y_{n,m}(\boldsymbol{R})}{R^{n+1}} &= \frac{im}{R^{n+1}}Y_{n,m}(\boldsymbol{R})
\end{aligned} \tag{4.28}$$

into (4.27) and comparing the result with the general formula [35]

$$[\nabla^y \times \boldsymbol{\Psi}(\boldsymbol{y})] \cdot \boldsymbol{R}_\gamma = \sum_{\nu=1}^{\infty} \nu(\nu+1)\chi^{(\gamma)}_{-(\nu+1)} \tag{4.29}$$

finally gives complex C-coefficients in (4.22). In actual programming, we prefer to reformulate this algorithm in terms of normalized solid harmonics $(|\boldsymbol{x}^j - \boldsymbol{x}^0_\gamma|/d^0_\gamma)^\nu Y_{\nu,\mu}(\boldsymbol{x}^j - \boldsymbol{x}^0_\gamma)$ instead of $Z_{\nu,\mu}(\boldsymbol{x}^j - \boldsymbol{x}^0_\gamma)$. This way, intermediate factorial operations (which could potentially overflow at very large indices ν, μ) are avoided. The computational cost of generating (4.22) to order $\nu = k_0$ for one patch is $\mathrm{O}(k_0^2|\mathcal{B}_\gamma|)$ (where $|\mathcal{B}_\gamma|$ is the number of mesh-nodes $\boldsymbol{x}^j$ in $\mathcal{B}_\gamma$) and comes from calculating the patch moments (4.13) and (4.16); transformation from (4.18) to Lamb's series (4.2) is only an $\mathrm{O}(k_0^2)$ operation. In the optimal form, the algorithm includes only $11|\mathcal{B}_\gamma|k_0^2$ and $15|\mathcal{B}_\gamma|k_0^2$ double precision multiplications for a patch on a drop and solid surface, respectively. In actual tests with $k_0 = 25$, this algorithm performed, respectively, about 2.5 and 2 times faster than our initial scheme (Section 3.2 of Ref. [20]) based on rotations. Besides, the new algorithm is considerably simpler and can be generalized for Green functions other than the free-space (3.2).

Moreover, this approach based on patch moments leads to a special, very fast scheme to generate Lamb's series (4.2) for a patch $\mathcal{B}_\gamma$ lying on a spherical (solid) surface S_β, if the expansion center $\boldsymbol{x}^0_\gamma$ is restricted to be on S_β. The key is the geometric identity

$$\left|\boldsymbol{x}^j - \boldsymbol{x}^0_\gamma\right|^2 = -2d^0_\beta\left(\boldsymbol{x}^j - \boldsymbol{x}^0_\gamma\right) \cdot \boldsymbol{n}\left(\boldsymbol{x}^0_\gamma\right) \tag{4.30}$$

for $\boldsymbol{x}^j \in \mathcal{B}_\gamma$, where d^0_β is the sphere radius. Let

$$\boldsymbol{\Omega}^{n,m,q}_{\nu,\mu} = \sum_{\boldsymbol{x}^j \in \mathcal{B}_\gamma} \left(\rho^j_1 + i\rho^j_2\right)^m \left(\rho^j_3\right)^{n-m} |\boldsymbol{\rho}^j|^{\nu+q} Y_{\nu,\mu}\left(\boldsymbol{x}^j - \boldsymbol{x}^0_\gamma\right) \triangle S_j \boldsymbol{Q}(\boldsymbol{x}^j), \tag{4.31}$$

with $\boldsymbol{\rho}^j = (\boldsymbol{x}^j - \boldsymbol{x}^0_\gamma)/d^0_\gamma$ and all indices being integer ($0 \leqslant m \leqslant n$, $|\mu| \leqslant \nu$). It is easy to see that the single-layer patch moments (4.13) are expressed via $\boldsymbol{\Omega}^{n,m,o}_{\nu,\pm\mu}$ with $n = 0$ and 1. So are the double-layer moments (4.16) $\breve{E}^{(\gamma)}_{\nu,\mu,k,s}$, since $\boldsymbol{n}(\boldsymbol{x}^j) = (d^0_\gamma/d^0_\beta)\boldsymbol{\rho}^j + \boldsymbol{n}(\boldsymbol{x}^0_\gamma)$. The moments $\breve{\mathcal{D}}^{(\gamma)}_{\nu,\mu,k}$ can be expressed in terms of $\boldsymbol{\Omega}^{n,m,o}_{\nu,\pm\mu}$ with $n = 1$ and 2. Let $\boldsymbol{\omega}^q_{\nu,\mu} = \boldsymbol{\Omega}^{o,o,q}_{\nu,\mu}$, for brevity. Using (4.24) and (4.30), a recurrent relation can be derived for $\boldsymbol{\omega}^q_{\nu,\mu}$:

$$\begin{aligned}\boldsymbol{\omega}^{q+2}_{\nu,\mu} = -\left(\frac{d^0_\beta}{d^0_\gamma}\right)\Big\{&\left[n_1\left(\boldsymbol{x}^0_\gamma\right) - in_2\left(\boldsymbol{x}^0_\gamma\right)\right]\left[f^1_{\nu,\mu}\boldsymbol{\omega}^q_{\nu+1,\mu+1} + f^2_{\nu,\mu}\boldsymbol{\omega}^q_{\nu-1,\mu+1}\right] \\ &+\left[n_1\left(\boldsymbol{x}^0_\gamma\right) + in_2\left(\boldsymbol{x}^0_\gamma\right)\right]\left[f^3_{\nu,\mu}\boldsymbol{\omega}^q_{\nu+1,\mu-1} + f^4_{\nu,\mu}\boldsymbol{\omega}^q_{\nu-1,\mu-1}\right] + 2n_3\left(\boldsymbol{x}^0_\gamma\right)\left[f^5_{\nu,\mu}\boldsymbol{\omega}^q_{\nu+1,\mu} + f^6_{\nu,\mu}\boldsymbol{\omega}^q_{\nu-1,\mu}\right]\Big\}\end{aligned} \tag{4.32}$$

(assuming $\boldsymbol{\omega}^q_{\nu,\mu} = 0$ for $|\mu| > \nu$). The values of $\boldsymbol{\omega}^0_{\nu,\mu}$ are calculated directly by patch summation (4.31) for $0 \leqslant \mu \leqslant \nu \leqslant k_0 + 1$ and then $\boldsymbol{\omega}^2_{\nu,\mu}$ and $\boldsymbol{\omega}^4_{\nu,\mu}$ are found successively from (4.32); continuation to negative μ is simply made by (4.11). Calculation of the necessary $\boldsymbol{\Omega}$-coefficients is performed by additional recurrent relations following from (4.24) and (4.31):

$$\begin{aligned}\boldsymbol{\Omega}^{n+1,m+1,q}_{\nu,\mu} &= f^1_{\nu,\mu}\boldsymbol{\Omega}^{n,m,q}_{\nu+1,\mu+1} + f^2_{\nu,\mu}\boldsymbol{\Omega}^{n,m,q+2}_{\nu-1,\mu+1}, \\ \boldsymbol{\Omega}^{n+1,m,q}_{\nu,\mu} &= f^5_{\nu,\mu}\boldsymbol{\Omega}^{n,m,q}_{\nu+1,\mu} + f^6_{\nu,\mu}\boldsymbol{\Omega}^{n,m,q+2}_{\nu-1,\mu}.\end{aligned} \tag{4.33}$$

Namely, Eq. (4.33) are first used with $n = m = 0, q = 2$, to determine $\boldsymbol{\Omega}^{n,m,2}_{\nu,\mu}$ for $n = 1$ and $m \leqslant n$, then with $n = 0$ and 1, $m \leqslant n$, and $q = 0$, to obtain the necessary $\boldsymbol{\Omega}^{n,m,0}_{\nu,\mu}$. Thus, instead of 13 scalar complex patch sums in (4.13) and (4.16) for every ν, μ, there are only three sums $(\omega^0_{\nu,\mu})_k$ $(k = 1, 2, 3)$ to calculate in the special case of a patch $\mathcal{B}_\gamma$ on a spherical solid surface; the total speedup in generating (4.22) is about three-fold (compared to our new algorithm for an arbitrary solid surface), with optimized recurrent calculation of solid harmonics for (4.13), (4.16), and (4.31).

4.3. Merging of singular expansions

Another element of our algorithm in Section 4.1 is the translation of Lamb's singular series (4.2) for a patch $\mathcal{B}_\gamma \subset S_\beta$ to the new expansion center $\boldsymbol{x}^0_\beta$. In principle, general formulae of Sangani and Mo [11] for translation of Stokes singularities of arbitrary order could be applied for this purpose. We use an alternative approach, however, based on rotational transformations of spherical harmonics, since it reduces the number of operations for one translation from O(k_0^4) to O(k_0^3) and also greatly simplifies the algebra (which is otherwise involved [11]). Mathematically, the idea is similar to that first used in conductivity [18,19] and then hydrodynamical multidrop [20–22] simulations when changing the expansion center (the difference, though, is that here we deal with singular-to-singular, not singular-to-regular reexpansions). Namely, a temporary "axial" basis (x_1', x_2', x_3') is introduced for patch $\mathcal{B}_\gamma$, with the x_3'-axis along the reexpansion vector $\boldsymbol{R}_{\beta\gamma} = \boldsymbol{x}_\gamma^0 - \boldsymbol{x}_\beta^0$ (Fig. 2). The A-, B- and C-coefficients in (4.22) are rotationally transformed to the new basis by Wigner functions (encountered in the quantum theory of angular momentum [36,37]), reexpansion to the new center $\boldsymbol{x}_\beta^0$ is made in the axial basis, and the new Lamb coefficients $A'^{(\beta)}_{-(v+1),m}$, $B'^{(\beta)}_{-(v+1),m}$, $C'^{(\beta)}_{-(v+1),m}$ are rotationally transformed back to the original basis (x_1, x_2, x_3). Each step of this three-step procedure (rotation + reexpansion + rotation) is O(k_0^3)-intensive. The back rotation is needed to sum up contributions from all patches $\mathcal{B}_\gamma \subset S_\beta$ and form the cumulative Lamb series (4.3).

The computationally efficient schemes for rotational transformation were discussed in detail [18–20], so we only need here to consider reexpansion from (4.2), (4.3) for one patch $\mathcal{B}_\gamma$ in the axial basis, assuming that the source Lamb coefficients $A^{(\gamma)}_{-(v+1),m}$, $B^{(\gamma)}_{-(v+1),m}$, $C^{(\gamma)}_{-(v+1),m}$ have already been rotationally transformed, and the new harmonics $p^{(\beta)}_{-(v+1)}$, $\Phi^{(\beta)}_{-(v+1)}$, $\chi^{(\beta)}_{-(v+1)}$ to be determined have the form

$$p^{(\beta)}_{-(v+1)}(\boldsymbol{R}_\beta) = \sum_{m=-v}^{v} A^{(\beta)}_{-(v+1),m} \left(\frac{d_\beta^0}{R_\beta}\right)^{v+1} Y_{v,m}(\boldsymbol{R}_\beta),$$

$$\Phi^{(\beta)}_{-(v+1)}(\boldsymbol{R}_\beta) = \sum_{m=-v}^{v} B^{(\beta)}_{-(v+1),m} \left(\frac{d_\beta^0}{R_\beta}\right)^{v+1} Y_{v,m}(\boldsymbol{R}_\beta),$$

$$\chi^{(\beta)}_{-(v+1)}(\boldsymbol{R}_\beta) = \sum_{m=-v}^{v} C^{(\beta)}_{-(v+1),m} \left(\frac{d_\beta^0}{R_\beta}\right)^{v+1} Y_{v,m}(\boldsymbol{R}_\beta) \tag{4.34}$$

in the axial basis (x_1', x_2', x_3'); primes are omitted in the derivations below using the (x_1', x_2', x_3') basis.

We start from the translation formula for negative-order solid harmonics, which follows from the generalized addition theorem (e.g., Ref. [38]) and greatly simplifies in the axial basis, with splitting in the azimuthal number m:

$$\left(\frac{d_\gamma^0}{R_\gamma}\right)^{v+1} Y_{v,m}(\boldsymbol{R}_\gamma) = \sum_{n=v}^{\infty} I^m_{v,n} \left(\frac{d_\beta^0}{\rho}\right)^{n+1} Y_{n,m}(\boldsymbol{\rho}), \tag{4.35}$$

where

$$I^m_{v,n} = \frac{1}{(n-v)!}\left[\frac{(2v+1)(n-m)!(n+m)!}{(2n+1)(v-m)!(v+m)!}\right]^{1/2} \frac{R_{\beta\gamma}^{n-v}\left(d_\gamma^0\right)^{v+1}}{\left(d_\beta^0\right)^{n+1}} \tag{4.36}$$

and $\boldsymbol{\rho} = \boldsymbol{R}_\beta = \boldsymbol{y} - \boldsymbol{x}_\beta^0$, for brevity. Coefficients $A^{(\beta)}_{-(v+1),m}$ are immediately derived from (4.35):

$$A^{(\beta)}_{-(n+1),m} = \sum_{v=|m|}^{n} I^m_{v,n} A^{(\gamma)}_{-(v+1),m}. \tag{4.37}$$

For calculating $B^{(\beta)}_{-(n+1),m}$, we dot the velocity (4.2) with $\boldsymbol{\rho}$, using $\boldsymbol{R}_\gamma = \boldsymbol{\rho} - \boldsymbol{R}_{\beta\gamma}$:

$$\sum_{n=0}^{\infty}\left[\frac{(n+1)}{2(2n-1)}\rho^2 p^{(\beta)}_{-(n+1)} - (n+1)\Phi^{(\beta)}_{-(n+1)}\right] = \sum_{v=0}^{\infty}\rho\frac{\partial\Phi^{(\gamma)}_{-(v+1)}}{\partial\rho} - R_{\beta\gamma}\sum_{v=1}^{\infty}\frac{\partial\chi^{(\gamma)}_{-(v+1)}}{\partial\varphi}$$
$$+\sum_{v=1}^{\infty}\left[\frac{(v+1)}{v(2v-1)}\left(\rho^2 - R_{\beta\gamma}\rho_3\right)p^{(\gamma)}_{-(v+1)} - \frac{(v-2)}{2v(2v-1)}\left(\rho^2 + R_{\beta\gamma}^2 - 2R_{\beta\gamma}\rho_3\right)\rho\frac{\partial p^{(\gamma)}_{-(v+1)}}{\partial\rho}\right], \tag{4.38}$$

where ρ_3 in the Cartesian component of $\boldsymbol{\rho}$ in the axial basis, and partial derivatives are taken in the spherical coordinates (ρ, θ, φ) associated with the axial Cartesian coordinate system centered at $\boldsymbol{x}_\beta^0$ (φ being the angle of positive rotation about $\boldsymbol{R}_{\beta\gamma}$). Expanding the harmonics (4.22) in terms of $\boldsymbol{\rho}$ by (4.35) and using the last recurrent relation (4.24) for $\rho_3 Y_{n,m}(\boldsymbol{\rho})$, one can derive from (4.38):

$$B^{(\beta)}_{-(n+1),m} = \frac{im}{(n+1)} R_{\beta\gamma} \sum_{v=|m|}^{n} I^m_{v,n} C^{(\gamma)}_{-(v+1),m} + \sum_{v=|m|}^{n} I^m_{v,n} B^{(\gamma)}_{-(v+1),m} + \frac{R_{\beta\gamma} d^0_\beta}{(n+1)} \left[\frac{(n-m+1)(n+m+1)}{(2n+1)(2n+3)}\right]^{1/2} \times \sum_{v=|m|}^{n+1} \frac{[(n+2)(v-2)+(v+1)]}{v(2v-1)} I^m_{v,n+1} A^{(\gamma)}_{-(v+1),m} - R^2_{\beta\gamma} \sum_{v=|m|}^{n} \frac{(v-2)}{2v(2v-1)} I^m_{v,n} A^{(\gamma)}_{-(v+1),m} \tag{4.39}$$

(omitting $v = 0$ for $m = 0$). To calculate $C^{(\beta)}_{-(n+1),m}$, we dot the curl of (4.2) with $\boldsymbol{\rho}$. The curl for Lamb's singular series is conveniently provided by the general formula from Sangani and Mo [11]:

$$\sum_{v=1}^{\infty} \left[-v \nabla \chi^{(\gamma)}_{-(v+1)} - \frac{1}{v} \boldsymbol{R}_\gamma \times \nabla p^{(\gamma)}_{-(v+1)} \right],$$

yielding

$$\sum_{n=1}^{\infty} n(n+1) \chi^{(\beta)}_{-(n+1)} = -\sum_{v=1}^{\infty} \left[\frac{1}{v} R_{\beta\gamma} \frac{\partial p^{(\gamma)}_{-(v+1)}}{\partial \varphi} + v\rho \frac{\partial \chi^{(\gamma)}_{-(v+1)}}{\partial \rho} \right], \tag{4.40}$$

where, again, the partial derivatives are taken in the spherical coordinates associated with the axial Cartesian coordinates centered at $\boldsymbol{x}^0_\beta$. Expanding the harmonics $p^{(\gamma)}_{-(v+1)}$ and $\chi^{(\gamma)}_{-(v+1)}$ in terms of $\boldsymbol{\rho}$ by (4.35), we arrive from (4.40) at

$$C^{(\beta)}_{-(n+1),m} = -\frac{imR_{\beta\gamma}}{n(n+1)} \sum_{v=|m|}^{n} \frac{1}{v} I^m_{v,n} A^{(\gamma)}_{-(v+1),m} + \sum_{v=|m|}^{n} \frac{v}{n} I^m_{v,n} C^{(\gamma)}_{-(v+1),m} \tag{4.41}$$

(omitting $v = 0$ for $m = 0$). Instead of seven sums over v in (4.37), (4.39) and (4.41), only four independent sums need to be calculated for each n and m, which can be seen from a recurrent relation

$$-I^m_{v,n} = \frac{R_{\beta\gamma}}{d^0_\beta} \left[\frac{(n-m)(n+m)(2n-1)}{2n+1}\right]^{1/2} \frac{I^m_{v,n-1}}{v} - \frac{n}{v} I^m_{v,n}. \tag{4.42}$$

Together with rotational transformations, our algorithm for reexpansion from (4.2) to (4.3) for one patch includes only $\approx 7.3k_0^3$ double precision multiplications and takes $1.0 \times 10^{-8} k_0^3$ seconds of CPU time.

4.4. Far-field calculations

The far-field contribution of surface $S^{\min}_\beta$, i.e., the last sum in (4.6), is calculated by a special form of the Taylor double series for $\boldsymbol{\tau}_1(\boldsymbol{x}^j - \boldsymbol{y})$ and/or $\boldsymbol{G}_1(\boldsymbol{x}^j - \boldsymbol{y})$ in powers of $\boldsymbol{x}^j - \boldsymbol{x}^{0,\min}_\beta$ and $\boldsymbol{y} - \boldsymbol{x}^0_\alpha = \boldsymbol{R}_\alpha$. The necessary formalism for this efficient, but somewhat cumbersome procedure has been already developed in our multidrop simulations [20–22] and only needs to be slightly adapted to the present problem. On every iteration, a sufficient number of double-layer far-field moments

$$\acute{D}^{(\beta)}_{v,\mu,k,s} = (-1)^{v-1} \sum_{\boldsymbol{x}^j \in S_\beta} W_{(s}(\boldsymbol{x}^j) Q_{k)}(\boldsymbol{x}^j) Z_{v,\mu}\left(\boldsymbol{x}^j - \boldsymbol{x}^0_\beta\right),$$
$$\acute{E}^{(\beta)}_{v,\mu,k,s,\ell} = (-1)^{v} \sum_{\boldsymbol{x}^j \in S_\beta} \left(x^j - x^0_\beta\right)_\ell W_{(s}(\boldsymbol{x}^j) Q_{k)}(\boldsymbol{x}^j) Z_{v,\mu}\left(\boldsymbol{x}^j - \boldsymbol{x}^0_\beta\right) \tag{4.43}$$

and (in case of a solid particle S_β) single-layer far-field moments

$$D^{(\beta)}_{v,\mu,k} = (-1)^{v} \sum_{\boldsymbol{x}^j \in S_\beta} Z_{v,\mu}\left(\boldsymbol{x}^j - \boldsymbol{x}^0_\beta\right) Q_k(\boldsymbol{x}^j) \triangle S_j,$$
$$E^{(\beta)}_{v,\mu,k,\ell} = (-1)^{v} \sum_{\boldsymbol{x}^j \in S_\beta} Z_{v,\mu}\left(\boldsymbol{x}^j - \boldsymbol{x}^0_\beta\right) Q_k(\boldsymbol{x}^j) \left(x^j - x^0_\beta\right)_\ell \triangle S_j \tag{4.44}$$

are precalculated for all S_β (cf. with Eqs. (3.15), (3.73) of Ref. [20]); these coefficients are not to be confused with *E*- and *D*-coefficients introduced in Section 4.2 for patches. For a spherical solid S_β (with $\boldsymbol{x}_\beta^0$ being the sphere center), only $\mathcal{D}_{v,\mu,k}^{(\beta)}$ are calculated by summation (4.44). Indeed, due to $\boldsymbol{W}(\boldsymbol{x}^j) = 2\boldsymbol{n}(\boldsymbol{x}^j)\triangle S_j$, $\boldsymbol{n}(\boldsymbol{x}^j) = (\boldsymbol{x}^j - \boldsymbol{x}_\beta^0)/d_\beta^0$ and recurrent relations (4.24), other moments $E_{v,\mu,k,s}^{(\beta)}$, $\acute{\mathcal{D}}_{v,\mu,k,s}^{(\beta)}$ and $\acute{E}_{v,\mu,k,s,\ell}^{(\beta)}$ are expressed through $\mathcal{D}_{v\pm1,\dots}^{(\beta)}$ and $\mathcal{D}_{v\pm2,\dots}^{(\beta)}$, $\mathcal{D}_{v,\dots}^{(\beta)}$, respectively. Using the moments (4.44), the single-layer part of the far-field contribution can be expressed in the coordinate form as

$$\begin{aligned}\sum_{\boldsymbol{x}^j\in S_\beta^{\min}} Q_k(\boldsymbol{x}^j)G_1^{k\ell}(\boldsymbol{x}^j-\boldsymbol{y})\triangle S_j = {} & \sum_{n=0}^{\infty}\sum_{m=-n}^{n} Z_{n,m}(\boldsymbol{R}_\alpha)\Bigg\{\sum_{v=0}^{\infty}\sum_{\mu=-v}^{v}\Bigg[D_{v,\mu,k}^{(\beta)}\partial_{n+v,m+\mu}\left[G_1^{k\ell}(\boldsymbol{r})-\frac{1}{2}\mathcal{P}_1^{(k)}(\boldsymbol{r})r_\ell\right]_{\boldsymbol{r}=\boldsymbol{R}_{\beta\alpha}} \\ & -\frac{1}{2}E_{v,\mu,k,\ell}^{(\beta)}\partial_{n+v,m+\mu}\mathcal{P}_1^{(k)}(\boldsymbol{R}_{\beta\alpha})+\frac{1}{2}(R_{\beta\alpha})_\ell D_{v,\mu,k}^{(\beta)}\partial_{n+v,m+\mu}\mathcal{P}_1^{(k)}(\boldsymbol{R}_{\beta\alpha})\Bigg]\Bigg\} \\ & +\frac{1}{2}(R_\alpha)_\ell\sum_{n=0}^{\infty}\sum_{m=-n}^{n}Z_{n,m}(\boldsymbol{R}_\alpha)\sum_{v=0}^{\infty}\sum_{\mu=-v}^{v}D_{v,\mu,k}^{(\beta)}\partial_{n+v,m+\mu}\mathcal{P}_1^{(k)}(\boldsymbol{R}_{\beta\alpha}),\end{aligned} \tag{4.45}$$

where $\boldsymbol{\mathcal{P}}_1(\boldsymbol{r}) = (\mathcal{P}_1^{(1)}, \mathcal{P}_1^{(2)}, \mathcal{P}_1^{(3)})$ is the vector of pressures for $\boldsymbol{G}_1(\boldsymbol{r})$, and $\boldsymbol{R}_{\beta\alpha} = \boldsymbol{x}_\alpha^0 - \boldsymbol{x}_\beta^{0,\min} \in (-1/2, 1/2)^3$. Upon appropriate transformation of moments $\acute{\mathcal{D}}_{v,\mu,k,s}^{(\beta)}$ and $\acute{E}_{v,\mu,k,s,\ell}^{(\beta)}$ (similar to (4.19)), the double-layer part of the far-field contribution takes a form analogous to (4.45) (see (A.2) of Ref. [22]) with the derivatives of $\boldsymbol{G}_1$ and $\boldsymbol{\mathcal{P}}_1$ only, and the two forms are combined for a solid surface S_β. The cumulative far-field contribution from all β in (4.6) is then combined with (4.8), before pointwise calculations are made for all $\boldsymbol{y} \in S_\alpha$.

Efficient Ewald-like forms for the periodic Green tensor $\boldsymbol{G}$ and the corresponding vector of pressures $\boldsymbol{\mathcal{P}}$ follow from Hasimoto [27]:

$$\begin{aligned}\boldsymbol{\mathcal{P}}(\boldsymbol{x}) &= -\boldsymbol{x} - \frac{1}{\pi^{3/2}}\sum_{\boldsymbol{m}}(\boldsymbol{x}-\boldsymbol{m})\int_{\pi^{1/2}}^{\infty}\mathrm{e}^{-t^2(\boldsymbol{x}-\boldsymbol{m})^2}t^2\mathrm{d}t - \frac{i}{2\pi}\sum_{\boldsymbol{m}\neq\boldsymbol{0}}\frac{\boldsymbol{m}\mathrm{e}^{-\pi\boldsymbol{m}^2-2\pi i\boldsymbol{m}\cdot\boldsymbol{x}}}{\boldsymbol{m}^2}, \\ \boldsymbol{G}(\boldsymbol{x}) &= \frac{\boldsymbol{I}}{4\pi} - \frac{1}{4\pi^{3/2}}\sum_{\boldsymbol{m}}\int_{\pi^{1/2}}^{\infty}\mathrm{e}^{-t^2(\boldsymbol{x}-\boldsymbol{m})^2}\left[\boldsymbol{I}+2t^2(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})\right]\mathrm{d}t \\ &\quad - \frac{1}{4\pi^2}\sum_{\boldsymbol{m}\neq\boldsymbol{0}}\left[\boldsymbol{I}-\pi\boldsymbol{m}\boldsymbol{m}\left(1+\frac{1}{\pi\boldsymbol{m}^2}\right)\right]\frac{\mathrm{e}^{-\pi\boldsymbol{m}^2-2\pi i\boldsymbol{m}\cdot\boldsymbol{x}}}{\boldsymbol{m}^2}.\end{aligned} \tag{4.46}$$

The far-field parts $\boldsymbol{G}_1$, $\boldsymbol{\mathcal{P}}_1$ are obtained from (4.46) by replacing the upper integration limit in the physical-space contributions by zero for $|\boldsymbol{m}| \leqslant m_0$ (which generalizeses Eq. (3.74) of Ref. [20] for arbitrary m_0). These expressions are employed to precalculate a large root table of $\partial_{v,\mu}[G_1^{k\ell}(\boldsymbol{r}) - \frac{1}{2}\mathcal{P}_1^{(k)}(\boldsymbol{r})r_\ell]$ and $\partial_{v,\mu}\mathcal{P}_1^{(k)}(\boldsymbol{r})$ for $\boldsymbol{r} = (n_1h, n_2h, n_3h)$, where the integers n_i are in the range $0 \leqslant n_1 \leqslant n_2 \leqslant N_T$, $0 \leqslant n_3 \leqslant N_T$, the table step is $h = 0.5/N_T$, and N_T is typically 20; when necessary, the tabulated values are extended to the entire box $[-1/2, 1/2]^3$ by symmetry properties. In dynamical simulations, the derivatives in (4.45) at $\boldsymbol{r} = \boldsymbol{R}_{\beta\alpha}$ are calculated by Taylor expansions (of typically third order) from the nearest point of the extended table (see (3.81) of Ref. [20]). With $m_0 \geqslant 2$, the far-field expansions are fast convergent even for small systems, since d_α^0, $d_\beta^0 \ll m_0 + 1$, and, on average, $\max(v, n) \leqslant$ 4-5 in (4.45) suffices for the dynamical simulations in Section 6, with the necessary order v of derivatives $\partial_{v,\mu}$ in the table never exceeding 11. For more demanding high-precision tests, larger $v \leqslant 16$ in the table were found to be sufficient. Calculation of the coefficients in the far-field part of the cumulative series (4.45) for all S_α is $\mathrm{O}(\max(v,n)^4(\widetilde{N}+\widehat{N})^2)$-intensive, but the numerical coefficient is small and independent of surface discretization. In the present problem, with necessarily high resolution, the total cost is instead greatly dominated by mesh-node operations, with approximately a linear scaling in the total number of nodes.

4.5. Calculation of additional integrals

For spherical solid particles $\widehat{S}_\beta$ (the only case considered in the numerical examples of Section 6), the simplest and most efficient way to incorporate the analytical calculation of additional integrals (2.11) (with $\boldsymbol{q}'(\boldsymbol{x}) = \langle\boldsymbol{q}\rangle_\beta + \boldsymbol{\Omega} \times (\boldsymbol{x} - \boldsymbol{x}_\beta^0)$ on $\widehat{S}_\beta$) into the scheme of Section 4.1 is through modification of multipole expansions. The root far-field moment (4.44) $\mathcal{D}^{(\beta)}_{v,\mu,k}$ is modified by simply adding a continuous analog, where the summation is replaced by integration, and $\boldsymbol{Q}$ replaced by $\boldsymbol{q}'$, which limits modifications (for $\mu \geqslant 0$) by

$$\begin{aligned}
&D^{(\beta)}_{0,0,k} \leftarrow D^{(\beta)}_{0,0,k} + 4\pi\left(d^0_\beta\right)^2 \langle q_k\rangle_\beta \\
&D^{(\beta)}_{1,0,1} \leftarrow D^{(\beta)}_{1,0,1} - \frac{4\pi}{3}\left(d^0_\beta\right)^4 \Omega_2, \quad D^{(\beta)}_{1,0,2} \leftarrow D^{(\beta)}_{1,0,2} + \frac{4\pi}{3}\left(d^0_\beta\right)^4 \Omega_1 \\
&D^{(\beta)}_{1,1,1} \leftarrow D^{(\beta)}_{1,1,1} + \frac{2\pi}{3}\left(d^0_\beta\right)^4 \Omega_3 i, \quad D^{(\beta)}_{1,1,2} \leftarrow D^{(\beta)}_{1,1,2} - \frac{2\pi}{3}\left(d^0_\beta\right)^4 \Omega_3 \\
&D^{(\beta)}_{1,1,3} \leftarrow D^{(\beta)}_{1,1,3} + \frac{2\pi}{3}\left(d^0_\beta\right)^4 (\Omega_2 - i\Omega_1).
\end{aligned} \tag{4.47}$$

The necessary changes are automatically propagated to other far-field moments $E^{(\beta)}_{v,\mu,k,\ell}$, $\acute{\mathcal{D}}^{(\beta)}_{v,\mu,k,s}$, $\acute{E}^{(\beta)}_{v,\mu,k,s,\ell}$ if they are expressed via modified $D^{(\beta)}_{v,\mu,k}$ (as discussed in Section 4.4). For free-space contributions, it can be shown that

$$\begin{aligned}
\int_{\widehat{S}_\beta} \boldsymbol{q}'(\boldsymbol{x}) \cdot [\eta \boldsymbol{G}_0(\boldsymbol{x}-\boldsymbol{y}) + 2\boldsymbol{\tau}_0(\boldsymbol{x}-\boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{x})]\mathrm{d}S_x = &-\frac{\eta\left(d^0_\beta\right)^2}{2}\left\{\left[1 + \frac{1}{3}\left(\frac{d^0_\beta}{R_\beta}\right)^2\right]\frac{\langle \boldsymbol{q}\rangle_\beta}{R_\beta}\right. \\
&\left.+\left[1 - \left(\frac{d^0_\beta}{R_\beta}\right)^2\right]\frac{\langle \boldsymbol{q}\rangle_\beta \cdot \boldsymbol{R}_\beta\boldsymbol{R}_\beta}{R^3_\beta}\right\} + \frac{\eta\left(d^0_\beta\right)^4}{3}\frac{\boldsymbol{R}_\beta \times \boldsymbol{\Omega}}{R^3_\beta} + I(\boldsymbol{y})\boldsymbol{q}'(\boldsymbol{y}), \quad \boldsymbol{R}_\beta = \boldsymbol{y} - \boldsymbol{x}^0_\beta,
\end{aligned} \tag{4.48}$$

where $I(\boldsymbol{y}) = 1$ for $\boldsymbol{y} \in \widehat{S}_\beta$ and $I(\boldsymbol{y}) = 0$ when $|\boldsymbol{R}_\beta| > d^0_\beta$. Representing the single-layer part of (4.48) as Lamb's singular series yields solid harmonics p_{-2}, Φ_{-2}, χ_{-2}, which must be added to (4.34) to modify the coefficients:

$$\begin{aligned}
&A^{(\beta)}_{-2,0} \leftarrow A^{(\beta)}_{-2,0} - \eta\left(\frac{4\pi}{3}\right)^{1/2}\langle q_3\rangle_\beta, \\
&A^{(\beta)}_{-2,1} \leftarrow A^{(\beta)}_{-2,1} - \eta\left(\frac{2\pi}{3}\right)^{1/2}\left(\langle q_1\rangle_\beta - i\langle q_2\rangle_\beta\right), \\
&B^{(\beta)}_{-2,0} \leftarrow B^{(\beta)}_{-2,0} - \frac{\eta\left(d^0_\beta\right)^2}{6}\left(\frac{4\pi}{3}\right)^{1/2}\langle q_3\rangle_\beta, \\
&B^{(\beta)}_{-2,1} \leftarrow B^{(\beta)}_{-2,1} - \frac{\eta\left(d^0_\beta\right)^2}{6}\left(\frac{2\pi}{3}\right)^{1/2}\left(\langle q_1\rangle_\beta - i\langle q_2\rangle_\beta\right), \\
&C^{(\beta)}_{-2,0} \leftarrow C^{(\beta)}_{-2,0} - \frac{\eta\left(d^0_\beta\right)^2}{3}\left(\frac{4\pi}{3}\right)^{1/2}\Omega_3, \\
&C^{(\beta)}_{-2,1} \leftarrow C^{(\beta)}_{-2,1} - \frac{\eta\left(d^0_\beta\right)^2}{3}\left(\frac{2\pi}{3}\right)^{1/2}(\Omega_1 - i\Omega_2).
\end{aligned} \tag{4.49}$$

With modified coefficients (4.47) and (4.49), the whole scheme of Section 4.1 now accounts for additional integrals (2.11) except for (a) self-interactions ($\widehat{S}_\alpha = \widehat{S}_\beta = \widehat{S}^{\min}_\beta$ and $\boldsymbol{m} = \boldsymbol{0}$) and (b) when the free-space contribution of $\widehat{S}^{\min}_\beta + \boldsymbol{m}$ to (4.6) is handled as a sum from individual patches. In these two cases, the term (4.48) is simply added directly, with $S^{\min}_\beta + \boldsymbol{m}$ and $\boldsymbol{x}^{0,\min}_\beta + \boldsymbol{m}$ replacing $\widehat{S}_\beta$ and $\boldsymbol{x}^0_\beta$, respectively.

4.6. Economical truncation of multipole expansions

An essential part of our algorithm in Section 4.1 is the "economical truncation" of multipole expansions/reexpansions depending on a single precision parameter, $\varepsilon \ll 1$. Although there is considerable freedom in constructing the truncation bounds, all efficient strategies must take into account that the convergence rate for Lamb's series (4.2) and (4.3) strongly depends on $d_\gamma^0/R_\gamma < 1$ or $d_\beta^0/R_\beta < 1$, respectively. Likewise, the rate of convergence of the far-field expansion (4.45) is mostly determined by the progression exponent $\max(d_\alpha^0, d_\beta^0)/\zeta_{\alpha\beta}$, where $\zeta_{\alpha\beta} = \min|\boldsymbol{R}_{\beta\alpha} - \boldsymbol{m}|$ (over $|\boldsymbol{m}| > m_0$) is the minimal center-to-center distance between the shell $\mathcal{D}_\alpha$ and the shells around the periodic images of $S_\beta^{\min}$ outside the near-field zone for S_α (see (4.5)). An *ad hoc* choice of uniform truncation bounds for all the expansions/reexpansions would greatly reduce the performance. On the other hand, rigorous majorants for multipole coefficients would not be most beneficial either, greatly overestimating the actual truncation errors. Our previous papers [20–22] on multidrop interactions offer a rational set of rules for the truncation bounds based on plausible arguments about the behavior of multipole coefficients. The same techniques are adapted to the present problem, as outlined in Appendix A. The differences from [20–22] are (i) a more general Green function partition (4.5) into the near-field and far-field parts and (ii) an additional set of truncation bounds associated with the patch expansions (4.2). Besides, surface partitioning into compact blocks through the slicing in Refs. [20–22] is disabled here, so blocks therein correspond to entire surfaces in the present work. This approach, although inevitably semi-empirical, leads to a truncation scheme depending on a single intuitive precision parameter, ε. This precision parameter is not a deviation from a non-multipole solution (by standard summations) in a rigorous sense, but does correlate with this deviation. As $\varepsilon \to 0$, all multipoles are eventually included (if unrestricted by the threshold k_0), which guarantees the convergence to the (much slower) non-multipole solution. Except for the initial moment $t = 0$, the truncation bounds are calculated on the first iteration only, since $\boldsymbol{Q}(\boldsymbol{x}^j)$ from the preceding time step provides a sufficient approximation for (A.2); the cost of truncation-bound calculations is negligible.

5. Additional features of the algorithm

5.1. Preiterative part of the boundary-integral calculations

Efficient calculation of the inhomogeneous term (2.6) $\boldsymbol{F}(\boldsymbol{y})$ also requires multipole acceleration. Note, however, that (i) only drop surfaces contribute to this term and (ii) $\boldsymbol{F}(\boldsymbol{y})$ is calculated outside the iterations. For this reason, we did not seek maximum efficiency of multipole acceleration for $\boldsymbol{F}(\boldsymbol{y})$ and used a simplification of the logical scheme of Section 4.1, without drop surface partitioning into patches (or blocks). Such a scheme almost parallels our initial algorithm [20] for drop–drop interactions, with two major exceptions. First, the Green function partition (4.5) into the near- and far-field parts herein is different. Second, an estimation of the behavior of the multipole coefficients (e.g., $a_k \sim C_\beta/(k+1)^3$ for small k) in the single-layer expansion for (3.3),

$$\sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} f(\boldsymbol{x}^j)\boldsymbol{n}(\boldsymbol{x}^j) \cdot \boldsymbol{G}(\boldsymbol{x}^j - \boldsymbol{y})\triangle S_j = \sum_{k=0}^{\infty} a_k \left(\frac{d_\beta^0}{R_\beta}\right)^{k+1}, \tag{5.1}$$

is done differently [21] through invariants of the tensor

$$\boldsymbol{T}_\beta = \sum_{\boldsymbol{x}^j \in \widetilde{S}_\beta} f(\boldsymbol{x}^j)\triangle S_j \boldsymbol{n}(\boldsymbol{x}^j)\left(\boldsymbol{x}^j - \boldsymbol{x}_\beta^0\right), \tag{5.2}$$

namely

$$C_\beta = \frac{1}{\pi\left(d_\beta^0\right)^2}\left[\left|\left|\boldsymbol{T}_\beta - \boldsymbol{T}_\beta^{\mathrm{T}}\right|\right| + \left|\mathrm{tr}\,\boldsymbol{T}_\beta\right| + 3\left|\left|\boldsymbol{T}_\beta\right|\right|\right], \tag{5.3}$$

which replaces (3.85) of Ref. [20] and was more successful in the present problem. In adapting the single-layer truncation scheme from that paper, the same parameters $e_{nf} = 1$ and $e_{ff} = 10$ were used.

For spherical solid particles $\widehat{S}_\beta$, the added-back integrals in (3.9) and (3.13) allow for simple analytical expressions [6]:

$$\begin{aligned}
&\int_{\widehat{S}_\beta} \boldsymbol{G}_0(\boldsymbol{r})\mathrm{d}S_x = -\frac{\widehat{a}^2}{2}\left[\left(1+\frac{\widehat{a}^2}{3R^2}\right)\frac{\boldsymbol{I}}{R}+\left(1-\frac{\widehat{a}^2}{R^2}\right)\frac{\boldsymbol{RR}}{R^3}\right],\\
&\int_{\widehat{S}_\beta}\frac{[\boldsymbol{r}\cdot\boldsymbol{n}(\boldsymbol{x})]r_k r_\ell r_m}{r^5}\mathrm{d}S_x = \frac{4\pi}{3}\left[\widehat{a}^3\left(\frac{\widehat{a}^2}{R^2}-1\right)\frac{R_kR_\ell R_m}{R^5}-\frac{\widehat{a}^5}{5R^5}(\delta_{k\ell}R_m+\delta_{km}R_\ell+\delta_{\ell m}R_k)\right],
\end{aligned} \tag{5.4}$$

where, for brevity, $\boldsymbol{r}=\boldsymbol{x}-\boldsymbol{y}$, $\boldsymbol{R}=\boldsymbol{R}_\beta=\boldsymbol{y}-\boldsymbol{x}^0_\beta$ and $\widehat{a}=d^0_\beta$. The discrete parts of $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$ in (3.9) and (3.13), although independent of the iterative solution, would still be too expensive to calculate by direct summations for all relevant combinations $(\boldsymbol{y},\boldsymbol{m},\beta)$ in drop–solid interactions, and so the use of multipole acceleration for these terms was quite essential. Still assuming the spherical shape for $\widehat{S}_\beta$ and substituting $\boldsymbol{r}^j=\widehat{a}\boldsymbol{n}(\boldsymbol{x}^j)-\boldsymbol{R}_\beta$ into (3.13), one can see that the calculation of the discrete parts of $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$ is reduced to calculation of twelve Stokes vector fields: (i) single- and double-layer parts (separately) of (4.4) with $\boldsymbol{Q}(\boldsymbol{x}^j)=\boldsymbol{e}_k$ $(k=1,2,3)$ and (ii) the double-layer part of (4.4) with $\boldsymbol{Q}(\boldsymbol{x}^j)=\widehat{a}n_k(\boldsymbol{x}^j)\boldsymbol{e}_\ell$ $(k\leqslant\ell)$. To expand each of these fields as a Lamb's singular series (4.3), we use the efficient technique of Section 4.2 (with $\mathcal{B}_\gamma$, $\boldsymbol{x}^0_\gamma$, $\boldsymbol{R}_\gamma$ and d^0_γ therein replaced by the quantities for the whole surface: $\widehat{S}_\beta$, $\boldsymbol{x}^0_\beta$, $\boldsymbol{R}_\beta$ and d^0_β, respectively). Specifically for the spherical shape, all necessary single- and double-layer moments (analogous to those in (4.13) and (4.16)) can be expressed in terms of

$$\sum_{\boldsymbol{x}^j\in\widehat{S}_\beta}\triangle S_j Y_{v,\mu}\left(\boldsymbol{x}^j-\boldsymbol{x}^0_\beta\right) \tag{5.5}$$

by recurrent relations (4.24). For stationary meshes on solids $\widehat{S}_\beta$, the coefficients (5.5) are time-independent, and they are precalculated before the entire simulation for all β and $0\leqslant\mu\leqslant v$ to a large order $v=k_0^{(S)}\approx 0.6\widehat{N}_\triangle^{1/2}$ (where $\widehat{N}_\triangle$ is the number of triangular elements on a solid surface). Lamb's series (4.3) is then generated dynamically (to save memory) for each of the 12 fields to order $v\approx k_0^{(S)}$ through (5.5) at a negligible cost, and converted to the more efficient form like (4.9). Pointwise calculations by these expansions are expedited by the symmetry of $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$. The truncation bound v^S on v for calculating the tensors $\boldsymbol{\Pi}(\boldsymbol{y}-\boldsymbol{m},\beta)$ and $\boldsymbol{\Gamma}(\boldsymbol{y}-\boldsymbol{m},\beta)$ in (3.11) by multipole expansions depends mostly on the clearance between $\boldsymbol{y}-\boldsymbol{m}$ and $\widehat{S}_\beta$. This bound is found semi-empirically from

$$\sum_{k=v^S+1}^{\infty}\breve{a}_k^{(\beta)}\left(\frac{d^0_\beta}{|\boldsymbol{y}-\boldsymbol{m}-\boldsymbol{x}^0_\beta|}\right)^{k+1}<0.2\varepsilon_1, \tag{5.6}$$

using the coefficients (A.1), to make the truncation errors for $\boldsymbol{\Pi}$- and $\boldsymbol{\Gamma}$-terms compatible with those for the first term in (3.11); the tolerance ε_1 is the same as in (A.3), only $\boldsymbol{R}_{\beta\alpha}$ in the second occurrence is replaced by $\boldsymbol{x}^0_\alpha-\boldsymbol{x}^0_\beta$. Once $v^S>k_0^{(S)}$, direct summations (3.9) and (3.13) are more economical and used for $\boldsymbol{\Pi}(\boldsymbol{y}-\boldsymbol{m},\beta)$ and $\boldsymbol{\Gamma}(\boldsymbol{y}-\boldsymbol{m},\beta)$ instead of the expansions. This simple combined scheme was found to accelerate the calculation of all necessary $\boldsymbol{\Pi}$- and $\boldsymbol{\Gamma}$-coefficients for solid-to-drop contributions almost three-fold in our runs; surface partition into patches is not advantageous in this part of the algorithm due to the complex, tensor character of $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$.

When the meshes on all solid spherical particles are identical, to within translation and possible scaling about the particle centers (in the case of polydispersity), the preiterative calculation of $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$ for solid-to-drop contributions can be further accelerated through tabulation. Indeed, in this case, $\boldsymbol{\Pi}(\boldsymbol{y},\beta)$ and $\boldsymbol{\Gamma}(\boldsymbol{y},\beta)$ are identical functions of $\boldsymbol{y}-\boldsymbol{x}^0_\beta$ for all β, to within similarity transformations. We used a large table outside a test unit solid sphere (150 points in the radial direction and 20480 auxiliary mesh triangles on each concentric sphere around the test one) to calculate the most time-consuming $\boldsymbol{\Gamma}(\boldsymbol{y},\beta)$ in the range $1.1<R_\beta/d^0_\beta<1.7$ by linear and quadratic interpolations in the radial and tangential directions, respectively, rather than using (3.13) dynamically. Outside the tabulation range and/or for $\boldsymbol{\Pi}(\boldsymbol{y},\beta)$, calculations proceed as described above. Such an approach additionally accelerates this part of the algorithm by about 1.5-fold, without any appreciable loss of accuracy, and makes sense primarily for matching viscosities $\lambda=1$, when the iterative part is relatively fast (Section 5.4); with even larger tables, $\boldsymbol{\Gamma}$ could be also handled by interpolations, except for $R_\beta/d^0_\beta\approx 1$. The tabulation approach, though, would be memory-prohibitive, if the meshes on solids $\widehat{S}_\beta$ were all different (in particular, adaptive to the random granular material geometry). We have found, however, adaptive meshes on solids to offer no advantage in the present calculations (Section 6).

5.2. Mesh control

A familiar difficulty in 3D boundary-integral or front-tracking calculations for deformable drops is dynamical surface-mesh degradation. Namely, if the mesh-nodes are advected simply with the interfacial (or normal) velocity, internode distances become highly irregular in a short simulation time, invalidating the mesh. One remedy, more traditional, is to restructure the mesh as needed, to maintain the quality of surface triangulation by nodes addition/subtraction/reconnection (Unverdi and Tryggvason [39], Tryggvason et al. [40]). The most elaborate version, widely used in drop breakup simulations, additionally incorporates adaptive mesh-node redistribution to minimize a potential, spring-like "mesh energy" (Cristini et al. [24]).

A quite different approach, "passive mesh stabilization" [6,20–22,30,41] is a family of methods to prevent mesh degradation not through topological changes but by adding a tangential field found globally for each surface to minimize a "kinetic energy" of disordered mesh motion. The same non-adaptive version as for a single drop squeezing through a finite cluster of particles [6] is employed herein. At any instant of time, the vertex velocities $\boldsymbol{V}_i = d\boldsymbol{x}^i/dt$ on $\widetilde{S}_\alpha$ to be used in the drop shape update are required to minimize

$$F = \sum_{\boldsymbol{x}_{ij}} \frac{1}{||\boldsymbol{x}_{ij}||^4}\left[\frac{\mathrm{d}}{\mathrm{d}t}||\boldsymbol{x}_{ij}||^2\right]^2 + 2\sum_{\triangle}\frac{1}{C_\triangle^2}\left(\frac{\mathrm{d}C_\triangle}{\mathrm{d}t}\right)^2 \tag{5.7}$$

under the constraints $\boldsymbol{V}_i \cdot \boldsymbol{n}(\boldsymbol{x}^i) = \boldsymbol{u}(\boldsymbol{x}^i)\cdot\boldsymbol{n}(\boldsymbol{x}^i)$, where the normal velocities in the RHS are provided by the boundary-integral solution. The summations in (5.7) are over all mesh edges $\boldsymbol{x}_{ij} = \boldsymbol{x}^j - \boldsymbol{x}^i$ (with $i < j$) between directly connected nodes on $\widetilde{S}_\alpha$, and over all mesh triangles $\triangle$ on $\widetilde{S}_\alpha$, and $C_\triangle = S_\triangle/(a^2+b^2+c^2)$ is the "compactness" of triangle $\triangle$ with area $S_\triangle$ and sides a, b, c. The first term (5.7) prevents the internode distances from becoming irregular, while the second term resists mesh triangle collapse. The form (5.7) is expressed as a quadratic function of the velocities $\boldsymbol{V}_i$ and minimized by conjugate-gradient iterations [30], until F stabilizes within a relative tolerance of 10^{-5}. The resulting computational cost of "passive mesh stabilization" is negligible compared to the rest of the code.

Compared to our other simulations [6,20–22], in the present problem for drops traveling through multiple tight constrictions between solids, the trend for mesh degradation is more severe due to close drop–solid interactions, even when the drops remain compact. For this reason, in the present problem, passive mesh stabilization alone could not provide sufficient control over the drop meshes for the entire simulation, and it was necessarily complemented by occasional "active" mesh-node redistribution (without topological changes to the mesh), to keep drop triangulations reasonably uniform. Let $\boldsymbol{x}_0^i$ be the mesh-node positions on $\widetilde{S}_\alpha$ prior to redistribution. For each $\boldsymbol{x}_0^i$, a best paraboloid [30] locally fitting $\widetilde{S}_\alpha$ around $\boldsymbol{x}_0^i$ is constructed. Iterations

$$\boldsymbol{x}^i_{\text{new}} = \boldsymbol{x}^i - \delta\frac{\partial E}{\partial \boldsymbol{x}^i} \tag{5.8}$$

are then organized to minimize the "potential mesh energy"

$$E = \sum_{\boldsymbol{x}_{ij}}\left(1 - \frac{\bar{\ell}^2}{||\boldsymbol{x}_{ij}||^2}\right)^q + c\sum_{\triangle}\frac{1}{C_\triangle^r}, \tag{5.9}$$

with empirical parameters $q = r = 20$, $c = 10^{-14}$–10^{-18}. Here, $\bar{\ell} = [4S_\alpha/(\widetilde{N}_\triangle\sqrt{3})]^{1/2}$ is the "target" value of a mesh edge $||\boldsymbol{x}_{ij}||$ for a hypothetical surface coverage by $\widetilde{N}_\triangle$ equilateral triangles. For very gradual mesh transition, the displacement δ is chosen as

$$0.001\min_i\left(\frac{\triangle x_i}{||\partial E/\partial \boldsymbol{x}^i||}\right), \tag{5.10}$$

where the minimum is over all mesh-nodes on the surface, and $\triangle x_i$ is the shortest distance from $\boldsymbol{x}^i$ to its neighbors. After each iteration (5.8), $\boldsymbol{x}^i_{\text{new}}$ is placed on the best paraboloid for the nearest node $\boldsymbol{x}_0^j \in \widetilde{S}_\alpha$ to continue the process; the positions $\boldsymbol{x}_0^i$ and the best paraboloids are not updated. Typically, from several thousand to O(10, 000) iterations are made; the maximum-to-minimum mesh edge ratio over the entire surface $\widetilde{S}_\alpha$ is usually minimized with a much smaller number of iterations, but additional iterations serve to improve the average compactness of mesh triangles. The first, spring-type term in (5.9) is highly sensitive to the "tensions"

$||\boldsymbol{x}_{ij}|| - \overline{\ell}$ and is akin to (although different from) the potential mesh energy of Cristini et al. [24]. Although there is some freedom in choosing a working combination (q, r, c), we have found it necessary, in the present problem, to always include the second term in (5.9), with a very small weight, to prevent mesh triangle collapse during minimizations (5.8) at the advanced stage of drop squeezing.

Mesh-node redistribution for a surface $\widetilde{S}_\alpha$ was made whenever (i) the maximum-to-minimum mesh edge ratio reached 5, or (ii) the minimum triangle compactness $C_\triangle$ was below 0.05, or (iii) the maximum curvature on $\widetilde{S}_\alpha$ reached (20–30)$\widetilde{a}^{-1}$ ($\widetilde{a}$ being the non-deformed drop radius); the first criterion was dominant. The curvature criterion, although necessary, was somewhat difficult to use, since it was unclear how to distinguish between numerical and physical high curvature developments, and it would crash the simulation in exclusive cases (probably, because node redistribution interferes with close drop–drop and drop–solid interactions). In such instances, operations (5.8) were simply delayed to a later moment. Owing to passive mesh stabilization greatly slowing down mesh degradation, node redistribution was needed extremely rarely, and for a few drops only. In a simulation of Section 6, with 25 drops and 10,000 time steps, the frequency of this operation (compared to the hypothetical, most unfavorable case, when node redistribution is done for *all* drops at *all* time steps) was about 5×10^{-5}. As an alternative, we attempted advecting mesh-nodes with the interfacial velocity $\boldsymbol{u}$ and using the node redistributions (5.8) as needed; the frequency of this operation was much higher (~ 0.005), and such a solution could proceed to small times only, with much smaller time steps required for stability and other difficulties. It is most beneficial, therefore, in the present problem to combine the two strategies, passive mesh stabilization and active node redistribution through minimization of a potential mesh energy.

The current version of our code does not incorporate topological mesh changes. It is perceived that the ability of drops traveling through tight pores of a *dense* granular material to stretch excessively and/or break (where the topological changes would be unavoidable) is severely limited by geometrical constraints; a numerical example in Section 6.1 confirms this viewpoint. The conditions for drop breakup in this problem, however, may require further analysis.

5.3. Swelling technique for generating a start-up configuration

In the present problem of an emulsion flow through a granular material, an appropriate model for solid particle arrangement would be a random packing in mechanical equilibrium (e.g., random “loose” or close packing [42]) or, at least, a very dense unconsolidated packing. A thermodynamical, or thermodynamic-like Monte-Carlo approach (e.g., [43–45]), with stochastic mixing and gradually swelling the particles, has been widely used in the literature to numerically simulate unconsolidated packings of spheres and approach (with fast densification rate) a random-close packing, which, mechanically, is the densest state that frictionless spheres can be randomly packed into. An alternative, very different algorithm [46] offers a mechanical picture [19] and can be extended for simulating “random-loose packing” of absolutely rigid spheres in mechanical equilibrium, as the opposite case of high interparticle friction. Having generated a particle arrangement, the next task is to add $\widetilde{N}$ drops with centroids in a periodic cell $[0, 1)^3$ and prescribed volumes $v_1, \ldots v_{\widetilde{N}}$, as a start-up configuration for dynamical boundary-integral (BI) simulation. Although the statistical steady state results (Section 6) are expected to be independent of the initial conditions, generating a start-up drop arrangement may present significant difficulties. One obvious way is to place drops first as spheres of tiny radius in the available space between the solid particles and then subject the drop phase to stochastic mixing, with gradual increase in the drop radii. In this manner, however, only low drop volume fractions $c_d = v_1 + \ldots v_{\widetilde{N}}$ can be achieved without drop–solid and drop–drop overlapping, For example, adding 40 drops to a random packing of 14 equisized solid spheres (at solid volume fraction $c_s = 0.5$), we could not go above $c_d = 0.031$ for monodisperse arrangements of spherical drops. It would be interesting to do BI simulations for much larger drop volume fractions, when squeezing meets high resistance and requires large drop deformation. Expanding the drops in the course of BI simulation, until the target volumes $v_1, \ldots v_{\widetilde{N}}$ are reached, was found to be computationally very prohibitive. The drop start-up volume fraction could be increased somewhat, if we stochastically mix *both* particles and spherical drops, with gradual swelling of their radii. In the example above, only $c_d = 0.155$ at $c_s = 0.5$ for monodisperse drop packings could be achieved through this procedure. More important, such a technique would be physically irrelevant; adequate simulation requires adding drops to the existing material microstructure.

We have found it very helpful to have a special geometrical algorithm for preparing a start-up configuration of smooth, deformed drops with a large enough volume fraction between solid particles. Once drops cannot be expanded anymore as spheres during stochastic mixing (see above), the mixing is stopped and replaced by continuous, simultaneous expansion of all drop surfaces $\widetilde{S}_\alpha$ with deformation, viewed as a "time-dependent" process, with artificial time τ. The normal "velocity" $\mathcal{D}_n(\boldsymbol{y})$ of the surface expansion at $\boldsymbol{y} \in \widetilde{S}_\alpha$ has a special form

$$\mathcal{D}_n(\boldsymbol{y}) = \frac{\delta^0(\boldsymbol{y}) - \delta^0_{\min}}{1 + c_{\exp}\left[k_1^2(\boldsymbol{y}) + k_2^2(\boldsymbol{y})\right]\delta^0(\boldsymbol{y})}. \tag{5.11}$$

Consider a ray $\mathcal{R}(\boldsymbol{y}) = \{\boldsymbol{x} : \boldsymbol{x} = \boldsymbol{y} + v\boldsymbol{n}(\boldsymbol{y}), v > 0 \text{ arbitrary}\}$, emanating from $\boldsymbol{y} \in \widetilde{S}_\alpha$ in the direction of the outward normal $\boldsymbol{n}(\boldsymbol{y})$ to $\widetilde{S}_\alpha$ (Fig. 4). The quantity $\delta^0(\boldsymbol{y})$ in (5.11) is simply the distance from $\boldsymbol{y}$ along $\mathcal{R}(\boldsymbol{y})$ to the nearest intersection point of $\mathcal{R}(\boldsymbol{y})$ with another surface (solid or drop) including periodic images, $\delta^0_{\min}$ is a prescribed small parameter so that the local surface expansion at $\boldsymbol{y}$ asymptotically slows down, as $\delta^0(\boldsymbol{y})$ reaches $\delta^0_{\min}$. The smooth invariant $k_1^2 + k_2^2$ is based on principal curvatures $k_1(\boldsymbol{y})$ and $k_2(\boldsymbol{y})$ of surface $\widetilde{S}_\alpha$ at $\boldsymbol{y}$, and $c_{\exp} > 0$ is another empirical parameter. There may be sharp variations of $\delta^0(\boldsymbol{y})$ along the surface $\widetilde{S}_\alpha$ (Fig. 4). The additional term in the denominator of (5.11) is designed to smooth $\mathcal{D}_n(\boldsymbol{y})$, makes it independent of $\delta^0(\boldsymbol{y})$ for large $\delta^0(\boldsymbol{y})$ and does not allow catastrophically high curvatures to develop in this swelling process; $c_{\exp} \neq 0$ is essential for the success of this technique.

In numerical implementation, the swelling process is applied to mesh nodes $\boldsymbol{y} = \boldsymbol{x}^i$. Intersections of $\mathcal{R}(\boldsymbol{y})$ with solids are handled analytically (which is the easiest for spheres, but can also be done for other shapes, such as ellipsoids); intersections with other drop surfaces are approximated as intersections with flat mesh triangles (Fig. 4). As in BI simulations (Section 6), the best paraboloid-spline method [20] is used to calculate normals $\boldsymbol{n}(\boldsymbol{y})$ and curvatures $k_1(\boldsymbol{y})$, $k_2(\boldsymbol{y})$. An Euler scheme $\boldsymbol{x}^i + \boldsymbol{V}_i\triangle\tau$ is employed to simultaneously update all the nodes $\boldsymbol{x}^i$, with the "velocities" $\boldsymbol{V}_i$ provided by passive mesh stabilization (5.7) under the constraints $\boldsymbol{V}_i \cdot \boldsymbol{n}(\boldsymbol{x}^i) = \mathcal{D}_n(\boldsymbol{x}^i)$; there is a Courant-like stability limitation on $\triangle\tau$, depending mostly on drop surface resolutions. Once a drop volume overshoots the target value v_α, the drop is scaled back slightly about the surface centroid $\tilde{\boldsymbol{x}}^c_\alpha$ to fit v_α and excluded from the swelling process, which greatly speeds up the final stage. Upon completion, $\delta^0_{\min}$ provides an estimation of the minimum drop–solid/drop–drop gap in the start-up configuration.

In the present applications of this swelling algorithm to monodisperse arrangements of particles and drops, we used $\delta^0_{\min} = 0.012\widehat{a}$ and $c_{\exp} = 2\widehat{a}$, where $\widehat{a}$ is the solid particle radius; stable steps $\triangle\tau$ were in the range of 0.005–0.01 for $\widetilde{N}_\triangle = 6000$–$8640$ triangular elements per drop. In the numerical example above with 40 drops and 14 solid spheres at $c_s = 0.5$, it took about 6000 steps $\triangle\tau$ to expand the drop phase from $c_d = 0.031$ to $c_d = 0.2$, requiring 8–15 h of CPU time, for $\widetilde{N} = 6000$ and 8640, respectively, with a simple way of calculating $\delta^0(\boldsymbol{y})$. An optimization of this stage would be hardly justified, since the boundary-integral simulations take longer (Section 6). Passive mesh stabilization is used in the swelling algorithm to keep drop meshing almost uniform. In the example above, the maximum-to-minimum mesh edge ratio over each surface $\widetilde{S}_\alpha$ remained

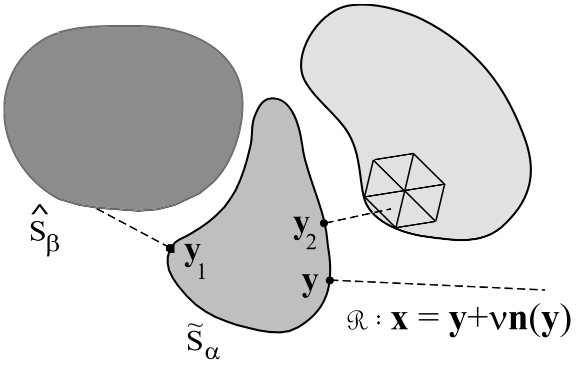

$\hat{S}_\beta$
y_2
y_1
y
$\mathcal{R} : \mathbf{x} = \mathbf{y} + \nu\mathbf{n}(\mathbf{y})$
$\widetilde{S}_\alpha$

Fig. 4. Sketch for the swelling algorithm to prepare a start-up configuration of drops between solid particles.

within 2.16 (and much closer to unity for most drops); in contrast, when the mesh-nodes were simply advected with the normal velocity (5.11), this ratio reached 17. Higher drop volume fractions (e.g., $c_d = 0.25$ at $c_s = 0.5$) could be also achieved through the swelling algorithm, but those were too difficult for boundary-integral simulations.

5.4. Miscellaneous

Traditional iterations by "successive substitutions" are divergent for the boundary-integral system (2.8), (2.12), but successful solutions were obtained in the present work by an alternative, generalized minimal residual iterative scheme, GMRES(k). Typically, $k = 4$ was sufficient and close to optimum in dynamical simulations. In exclusive cases, a larger k, up to 7, was needed for convergence, especially at start-up, due to close solid–solid contacts and the absence of a good initial approximation for $(\boldsymbol{q}, \boldsymbol{w})$. The code starts from $k = 4$ and automatically increases k by 1, if the maximum residual of the Eq. (2.12) over all mesh-nodes still exceeds a prescribed portion (10^{-4} in Section 6) of the r.m.s. value of the solution $(\boldsymbol{q}, \boldsymbol{w})$ over all surfaces after 25 iterations. The converged solution from two preceding time steps was linearly extrapolated to provide an initial approximation, except at start-up.

Unlike for pure drop systems, where the homoviscous case $\lambda = 1$ has been studied most extensively in the literature to avoid boundary-integral iterations, the present problem for $\lambda = 1$ still requires an iterative solution. However, the iterations for $\boldsymbol{q}(\boldsymbol{x})$ on solids $\widehat{S}_\alpha$ are now decoupled. After the Hebeker density $\boldsymbol{q}(\boldsymbol{x})$ is found by the iterative technique above, with the convergence control over solids $\widehat{S}_\alpha$ only (we used higher tolerance 10^{-5} in this special case), solid-to-drop contributions are calculated only once to get $\boldsymbol{w}(\boldsymbol{y})$. This provides a great boost in performance, if the number of drops $\widetilde{N}$ exceeds (or is comparable to) the number of solids $\widehat{N}$. The decoupling between $\boldsymbol{q}$ and $\boldsymbol{w}$, though, would not be possible had we chosen a constant flow rate instead of the constant-pressure gradient formulation (2.9).

Dynamical smoothing at each time step (Eq. (5.2) of Zinchenko and Davis [20]), together with occasional mesh-node redistribution (Subsection 5.2) helped to suppress artificial surface irregularities with abnormally high curvature; the smoothing parameter ε_S (Ref. [20]) was very small (10^{-5}) and found not to have any appreciable effect on the global quantities of interest. The drop shapes were rescaled at each time step about the drop centroids $\tilde{\boldsymbol{x}}_\alpha^c$ to keep the drop volumes constant. This common procedure is used to reduce the long-time cumulative error, with a negligible effect in the limit of fine triangulations. As in [6], an empirical rule provided a stable and economical time step (in dimensional form)

$$\triangle t = K_{\triangle t} \frac{\mu_e}{\sigma} \min_i \left\{ \frac{\triangle x_i}{\widetilde{a} \max\left[|k_1(\boldsymbol{x}^i)|, |k_2(\boldsymbol{x}^i)|\right]} \right\}, \tag{5.12}$$

where the minimum is taken over all mesh-nodes $\boldsymbol{x}^i$ on all drops, $\widetilde{a}$ is the non-deformed radius of the drop containing $\boldsymbol{x}^i$, and $\triangle x_i$ is the minimum distance from $\boldsymbol{x}^i$ to its neighbors on the same surface; an empirical parameter $K_{\triangle t}$ was set to ≈ 7.4 in all calculations. Although most calculations (Section 6) were done by the second-order Runge–Kutta time integration, we have found more recently that the simplest Euler scheme is quite adequate in the present problem; the results for the drop ($\boldsymbol{U}_D$) and the continuous ($\boldsymbol{U}_C$) phase velocities by the two schemes are barely distinguishable in the whole time range of simulations. The reason is that high local curvatures developing in the drop squeezing process necessitate a very small time step (5.12) for stability, which makes the time integration error far less than the triangulation effect.

If the problem was solved exactly, lubrication layers would prevent drop–solid and drop–drop contacts (in the absence of singular molecular forces [47,48]). For flow-induced squeezing of a single drop through a tight free-space cluster of two or three particles, it was possible to fully resolve lubrication and demonstrate that there is, indeed, a drop–solid spacing about 1–2% of the solid particle radius during squeezing (Zinchenko and Davis [6]). At subcritical conditions, when the drop is trapped in the constriction, a stationary gap (still of the order of $0.01\widehat{a}$) was observed in our simulations (ibid.); the existence of the stationary solution with a non-zero drop–solid spacing is due to the pumping mechanism of the external flow [49,50].

In the present simulations, though, for multiple drops traveling under a pressure gradient through a dense granular material with tight constrictions, it is much harder to resolve lubrication, and is not possible to completely avoid drop–solid overlappings, even with superhigh surface resolution (Section 6.3); if untreated, these overlappings quickly break down the simulation due to divergence of iterations. A simple technique was used to control drop–solid spacings. Let $\delta(\boldsymbol{y})$ be the distance from $\boldsymbol{y} \in \widetilde{S}_\alpha$ along the ray $\mathcal{R}(\boldsymbol{y})$ (Section 5.3) to the nearest intersection point of $\mathcal{R}(\boldsymbol{y})$ with a solid, and $\delta_{\min}$ (not to confuse with $\delta^0_{\min}$ from Section 5.3) be a small prescribed tolerance. On each time step, after the boundary-integral solution, the drop mesh-nodes $\boldsymbol{x}^i$ with $\delta(\boldsymbol{x}^i) < \delta_{\min}$ are simply moved in the direction opposite to $\boldsymbol{n}(\boldsymbol{x}^i)$ to make $\delta(\boldsymbol{x}^i) = \delta_{\min}$. This *ad hoc* technique (which worked substantially better than, say, an artificial repulsion with singular forces) is not a substitute for an accurate numerical solution and was only successful in our problem when lubrication is "almost resolved" and the trend for drop–solid overlapping is very weak. In our simulations with high resolution, the above geometrical correction was needed, at each time step, only for an extremely small portion of the total number of nodes on drops (Section 6.3). Even more important, in an appropriate range of small $\delta_{\min}$ and for conditions not too close to critical for squeezing to occur, the global long-time quantities of interest (the continuous and drop phase velocities) are practically insensitive to $\delta_{\min}$ (Section 6.3), which justifies the procedure for finite deformations, when lubrication layers are not highly localized.

Drop–drop overlappings also occurred in our simulations, but we have found it best to leave them untreated. The analysis in Section 6 shows that these overlappings involve only a very small portion of the total number of nodes on drops and, on the average, remain tiny throughout the simulation, disappearing as more triangles are used. The virtue of the variational approach (3.5)–(3.7) to near-singularity subtraction in the double-layer drop contributions is that it leaves boundary-integral iterations convergent for slight drop–drop overlapping and moderate λ, thus allowing simulations to succeed. Again, though, for small deformations not considered here, close drop–drop interactions would be far more difficult to handle (Zinchenko and Davis [23]), requiring resolution of small near-contact areas.

6. Numerical results

6.1. Drop squeezing through a cluster of four spheres

We first tested the free-space version of our multipole-accelerated code, when a single drop squeezes through a tight cluster of four spheres of radius $\widehat{a}$ rigidly held in an infinite domain of fluid. The drop is freely suspended in the flow $\boldsymbol{u}_\infty$, which is uniform away from the particles and normal to one of the cluster faces (Fig. 5); the drop non-deformed radius $\widetilde{a} = 0.5\widehat{a}$ is considerably larger than the inner radii of the constrictions. The physical formulation is the same as for two- and three particle constriction problems [6] but the new case, with the drop passing two successive constrictions and changing the direction of motion (Fig. 6) is much harder to simulate. As the drop squeezes through, it decelerates about 500 times in the middle of the cluster. High resolution (11.5 K triangular elements on the drop and 8.6 K elements on each solid surface) has allowed this simulation to succeed without the artificial parameter $\delta_{\min}$; the minimum drop–solid spacing reached $0.008\widehat{a}$ during squeezing. Due to the small cluster size and the lack of periodic boundaries, computational gains through multipole acceleration are not as dramatic in this case as for emulsion flow (Section 6.3), but they are still quite significant. For an instantaneous configuration in the middle of the squeezing process

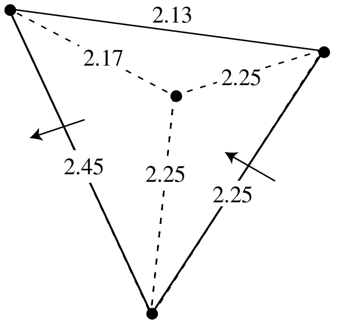


Fig. 5. Particle configuration for the simulation of drop squeezing through a cluster of four spheres. Numbers show center-to-center distances between the spheres, relative to the sphere radius. The inward arrow is the direction of $\boldsymbol{u}_\infty$ and also shows the face through which the drop enters the cluster, departing through the face shown by the outward arrow. The cluster orientation is the same as in Fig. 6.

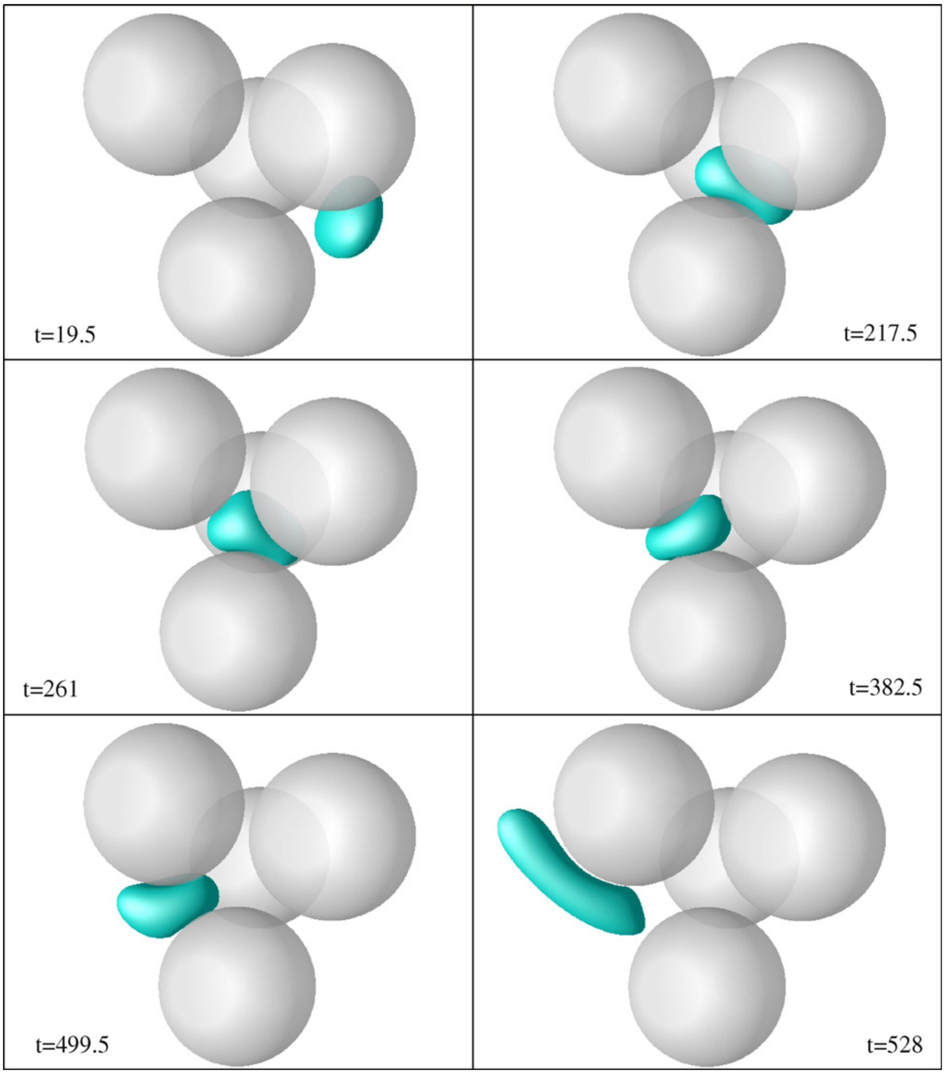


Fig. 6. Snapshots of a single drop squeezing between four fixed spheres in free-space, at $\tilde{a} = 0.5\hat{a}$, $\mu^e|\boldsymbol{u}_\infty|/\sigma = 2.5$, $\lambda = 4$; time is scaled with $\hat{a}/|\boldsymbol{u}_\infty|$. At $t = 0$, the drop was spherical, its center aiming at the center of the face shown by the inward arrow in Fig. 5 and at a distance of $3\hat{a}$ from that face.

($t = 308$), one boundary-integral iteration by the present code takes 2.9 s, which is four times faster than by the non-multipole code [6]; the whole simulation took 4500 second-order RK steps.

As the drop leaves the cluster (Fig. 6), it starts elongating extremely fast and breakup is imminent in the free-space (which the present mesh algorithm would not capture); the drop, however, remained compact during squeezing. This observation appears to confirm our viewpoint (Section 5.2) that excessive elongation and breakup of emulsion drops traveling through a dense granular material are severely limited by geometrical constraints imposed by the solid particles in a broad range of parameters, except when drops are small compared to the constriction size. With the added possibility of drop breakup, simulations of emulsion flow through a random granular material (Section 6.3) would probably be of formidable difficulty.

6.2. Single-phase flow through periodic and random beds of spheres

Accurate multipole expansion solutions have been obtained in the literature for the pressure-driven Stokes flow of a pure liquid (no drops) through periodic (Sangani and Acrivos [1], Zick and Homsy [2]) and random monodisperse (Ladd [3], Mo and Sangani [4]) arrays of fixed solid spheres. The results are usually represented by the ratio F/F_0 of the (average) hydrodynamic force acting on a bed particle to the Stokes drag $6\pi\mu_e\widehat{a}\langle\boldsymbol{u}\rangle$ on an isolated particle, where $\langle\boldsymbol{u}\rangle$ is the flow velocity averaged over the entire space (with the continuation $\boldsymbol{u}=0$ inside the solids). Our code is also applicable in the absence of drops to calculate the same quantity $F/F_0=2/[9c_s(1-c_s)(U_C)_3]$ via the z-component of the non-dimensional continuous phase velocity $\boldsymbol{U}_C$ (Section 2), which provided very useful tests. In Table 1, our results for dense simple cubic arrays are compared with those of Mo and Sangani [4]; at $c_s=0.45$ and 0.5236, F/F_0 can be also extracted from a more general multipole collocation solution of Chapman and Higdon [5], with a perfect match to Ref. [4]. As the number $\widehat{N}_\triangle$ of mesh triangles on a solid in our solution is increased from 3840 to 24,000, excellent agreement with prior results is achieved in the whole range of concentrations, up to the maximum packing $c_s=\pi/6$; for $c_s=0.5$, the deviation is only 0.24%.

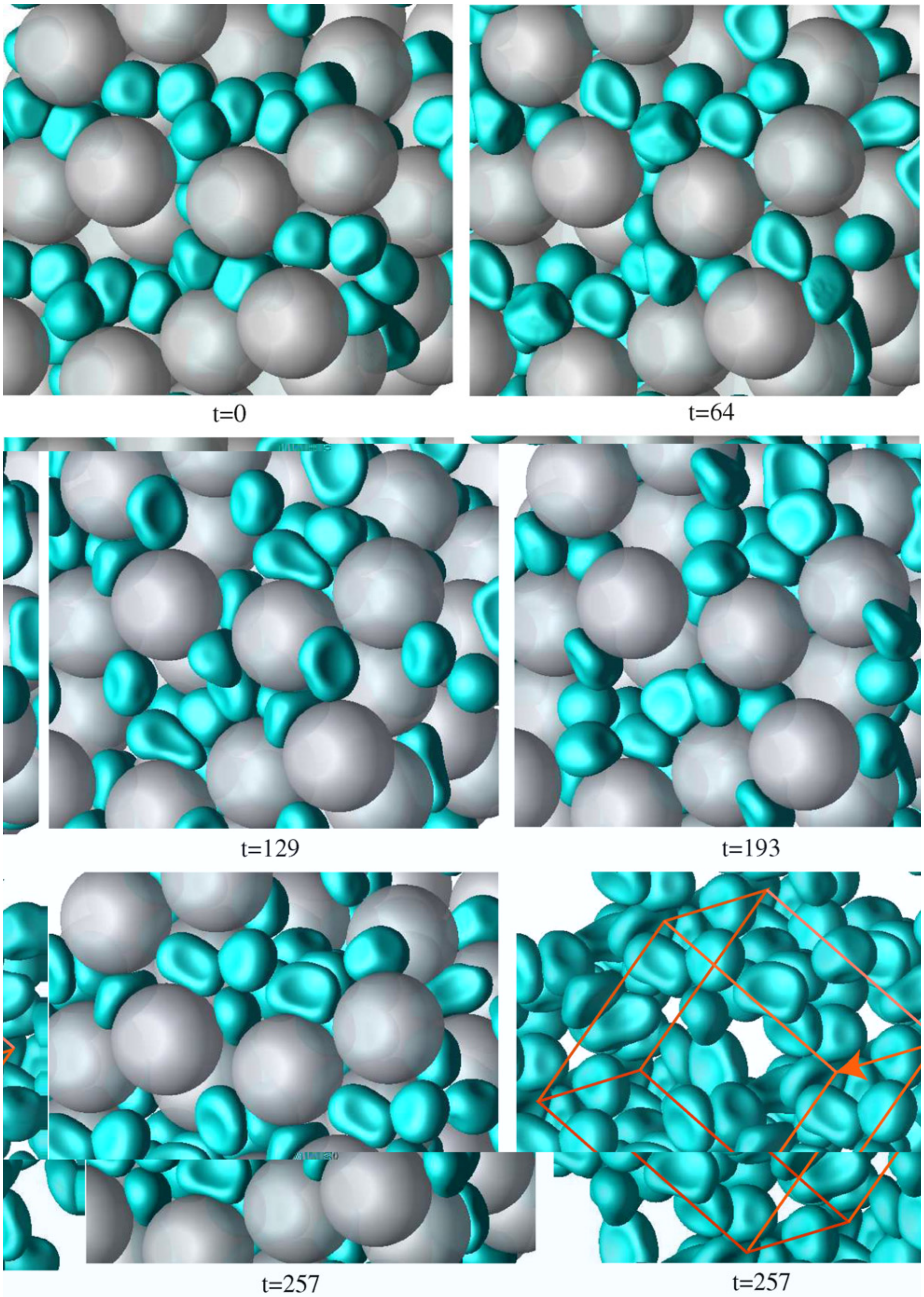
We also calculated F/F_0 for random arrays at $c_s=0.45$ by averaging over 30 configurations with $\widehat{N}=50$ solid spheres in a periodic box and $\widehat{N}_\triangle=11,520$. The only difference from the standard Monte-Carlo method [43] in preparing random configurations is that we did not allow spheres to come closer than $0.001\widehat{a}$ during stochastic mixing. This limitation is physically insignificant since there is no solid–solid lubrication in the problem, but it improves the iterative convergence. Our average result $\langle F/F_0\rangle=28.46\pm0.20$ (67% confidence level) is in excellent agreement with 28.6 for $\widehat{N}=16$ from Ladd [3] and Mo and Sangani [4], and with 28.2 for $\widehat{N}=108$ from Ladd [3]. Recalculation for several chosen configurations using higher resolution $\widehat{N}_\triangle=24,000$ suggests that our average value of 28.46 must be corrected by about +0.2 to eliminate the triangulation effect (in this test, a less optimal value of $\eta=1/\widehat{a}$ was used), which is still in very good agreement with the published results.

Previous solutions [1–5] for these permeability problems did not need particle meshing or a boundary-integral. Instead, the flow was represented by a sufficient (and modest) number of multipoles located at particle centers, taking advantage of spherical shapes. This approach results in a superior global rate of convergence for F/F_0 (compared to our boundary-integral method), and a natural question is whether it can be generalized for drop squeezing problems, with meshing and boundary-integrals applied to drop surfaces only. We explored this option for the problem of a single drop squeezing through a tight free-space cluster of three spheres (successfully handled by the boundary-integral tools [6]) but found it to not work. Even with multipoles of order up to 100 retained, our dynamical simulations crashed when the drop just entered the constriction; a strong trend for drop–solid overlapping could not be overcome even with the artificial geometrical barrier $\delta_{\min}$, leading to instability. It appears that the boundary-integral approach, with singularity distributions over a surface and subsequent integral desingularizations, provides much better control of the *local* error than when the singularities are placed in particle centers. In our code, multipoles are used only to (very substantially) accelerate the performance, but the boundary-integral is the starting point, with necessary meshing for both drops and particles.

The same observation explains why the Power-Miranda range completion for the double-layer (Section 2), with an additional Stokeslet and Rotlet placed in the particle center, is less robust in drop squeezing problems than is the Hebeker representation (used throughout this work) with force and torque effects distributed over the particle surface.

Table 1
Non-dimensional drag F/F_0 on a solid sphere for a single-phase flow through simple cubic arrays

c_s	Present code ($\eta=3/\widehat{a}$)			Mo and Sangani [4]
	$\widehat{N}_\triangle=8640$	11,520	24,000	
0.4	21.15	21.17	21.20	21.24
0.45	27.92	27.97	28.03	28.09
0.5	36.72	36.80	36.91	37.00
0.5231	41.56	41.67	41.83	–
0.5236	–	–	–	42.1



t=0

t=64

t=129

t=193

t=257

t=257

Fig. 7. Snapshots of the emulsion squeezing through a random loosely-packed array of solid spheres, with $Ca = 0.75$, $\lambda = 4$, $c_s = 0.5$, $c_d = 0.2$, $\widehat{N} = 9$, $\widetilde{N} = 25$, $\widehat{N}_\triangle = 8640$, $\widetilde{N}_\triangle = 6000$, $\widehat{M}_\triangle = \widetilde{M}_\triangle = 20$, $\varepsilon = 3.9 \times 10^{-7}$, $k_0 = 30$, $\eta = 1/\widehat{a}$, second-order RK scheme. Time is scaled with $\mu^e/(|\langle\nabla p\rangle|\widehat{a})$. In the last frame, the drop phase only is shown, together with the periodic box $[0,1)^3$ and the direction of $-\langle\nabla p\rangle$, common for all frames. Only the drops and solids with centroids in the extended box $[-1/2, 3/2]^3$ are shown.

6.3. Emulsion flow through a random granular material

Returning to the general problem formulated in Section 2, Fig. 7 represents snapshots of a monodisperse emulsion flow through a monodisperse granular material of spherical particles with $c_s = 0.5$ solid volume fraction, $c_d = 0.2$ overall drop volume fraction (or 40% of the void space between the particles), viscosity ratio $\lambda = 4$, capillary number $Ca = |\langle\nabla p\rangle|\widehat{a}\widetilde{a}/\sigma = 0.75$, $\widehat{N} = 9$ solids and $\widetilde{N} = 25$ drops with centroids in the periodic cell $[0,1)^3$, and high resolution ($\widehat{N}_\Delta = 8640$ and $\widetilde{N}_\Delta = 6000$ triangular elements on each solid and drop, respectively). In the last frame, the drop phase only is shown, together with the periodic box $[0,1)^3$ and the direction of $-\langle\nabla p\rangle$ (common for all frames of Fig. 7). The stationary solid phase was first generated as a "random-loose packing" of highly frictional particles with perfect contacts in mechanical equilibrium (against the confining pressure, if applied) at $c_s = 0.5098$. The code for such physically realistic particle arrangements is a generalization of the random-close packing algorithm of Zinchenko [46] and will be described elsewhere; for the purposes of the present paper, just the particle centers in $[0,1)^3$ are listed in Appendix B. The deviation of $c_s = 0.5098$ from the experimental value of 0.555 for random-loose packings [51] is primarily a statistical fluctuation for a small $\widehat{N} = 9$ system. The solid phase was then slightly diluted to $c_s = 0.5$ by shrinking the particle radius, to make gaps of $0.013\widehat{a}$ between former neighbors and alleviate iterative convergence difficulties for

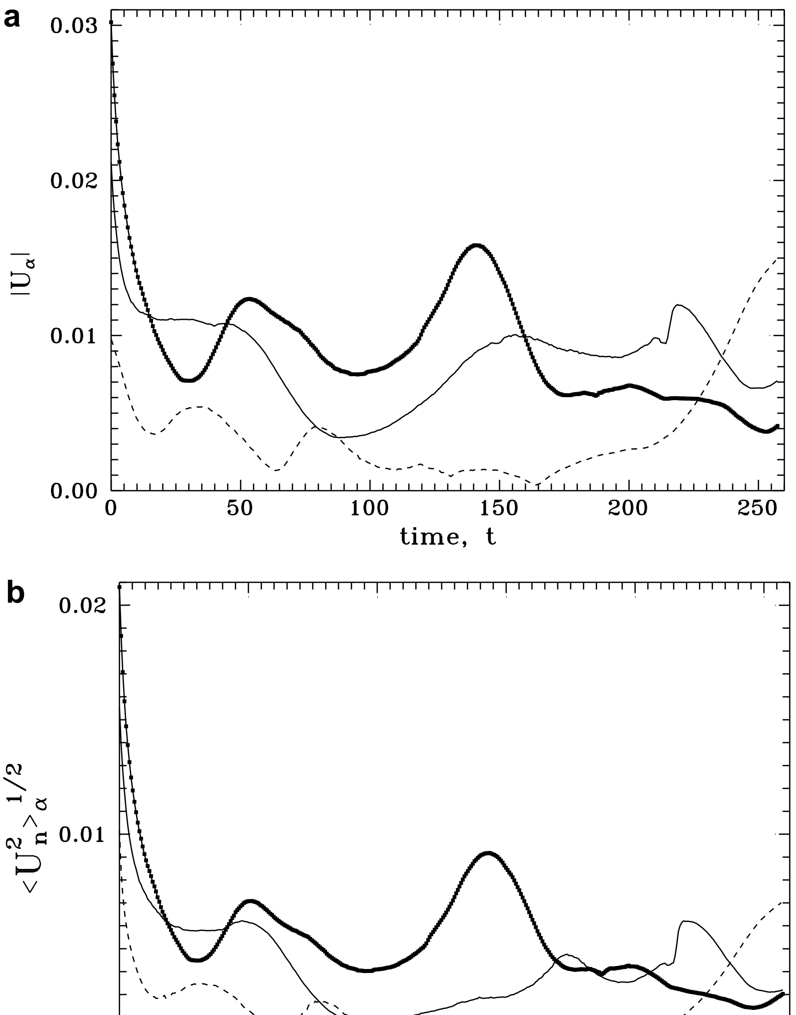
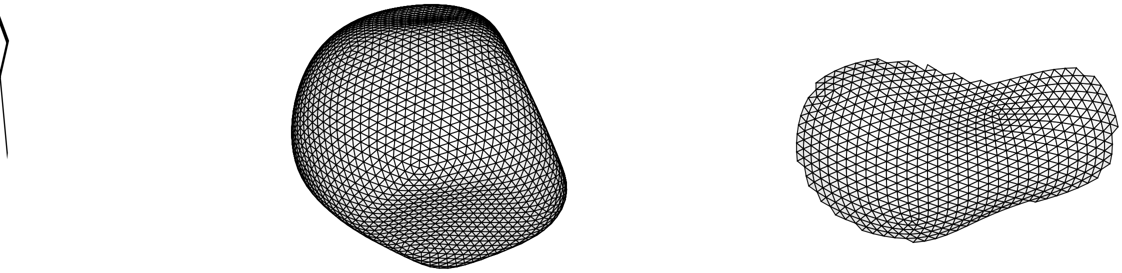


Fig. 8. The (a) individual drop velocities and (b) corresponding r.m.s values of the normal velocity for three chosen drops from the simulation shown in Fig. 7. The thin solid lines correspond to the drop shown in Fig. 9. The velocity scale is $|\langle\nabla p\rangle|\widehat{a}^2/\mu^e$.

boundary-integral simulations. The start-up configuration of the drop phase (Fig. 7 at $t = 0$) was prepared by the swelling technique of Section 5.3 from a dilute random arrangement of spherical drops with $c_d = 0.057$. Dimples on the drop surfaces visible in Fig. 7 result from close interaction with surrounding solids not included in the visualization. Fig. 8a and b presents the magnitude of an individual velocity $\boldsymbol{U}_\alpha$ (defined as the volume average of the fluid velocity $\boldsymbol{u}$ inside $\widetilde{S}_\alpha$) and the corresponding root-mean-square value $\langle(\boldsymbol{u}\cdot\boldsymbol{n})^2\rangle_\alpha^{1/2}$ of the normal velocity on $\widetilde{S}_\alpha$ for three chosen drops; $|\boldsymbol{U}_\alpha|$ and $\langle(\boldsymbol{u}\cdot\boldsymbol{n})^2\rangle_\alpha^{1/2}$ highly correlate. Zero r.m.s normal velocity would be characteristic of complete trapping in a constriction. Judging by Fig. 8, on the average, each drop passes a constriction 1–2 times; good mesh quality is retained for the entire simulation



(Fig. 9). The whole run took about 10,000 second-order Runge–Kutta steps, with typically 9–10 boundary-integral iterations on each half of the time step. A substantial number of iterations, even with the initial approximation extrapolated from the preceding time steps, indicates that our time steps are not far from the stability limit and could not be greatly increased.

Because of the large size and complexity of our multipole-accelerated code, it was important to provide an additional validation of our calculations. To this end, the run in Fig. 7 was interrupted at $t = 129$, to compare our inhomogeneous term $\boldsymbol{F}(\boldsymbol{x}^i)$ (Eq. (2.6)), interfacial velocity $\boldsymbol{u}(\boldsymbol{x}^i)$ and Hebeker density $\boldsymbol{q}(\boldsymbol{x}^i)$ (after one iteration, for simplicity) with the corresponding exact values $\boldsymbol{F}_{ex}(\boldsymbol{x}^i)$, $\boldsymbol{u}_{ex}(\boldsymbol{x}^i)$ and $\boldsymbol{q}_{ex}(\boldsymbol{x}^i)$ (for the given configuration and triangulation) calculated much simpler, using only direct summations in (3.3), (3.5), (3.8), (3.9), (3.11) and (3.13). In the non-multipole scheme, periodic kernels $\boldsymbol{G}$ and $\breve{\boldsymbol{\tau}}$ were calculated by shifting the argument to the box $[-\frac{1}{2}, \frac{1}{2})^3$ and interpolating the smooth functions $\boldsymbol{G} - \boldsymbol{G}_0$ and $\breve{\boldsymbol{\tau}} - \breve{\boldsymbol{\tau}}_0$ from tables, which is now a standard approach (following Loewenberg and Hinch [52]); using quadratic interpolations and a large table size ($121 \times 121 \times 121$ in $[-\frac{1}{2}, \frac{1}{2})^3$) allowed us to avoid any appreciable error in this test. For spherical particles, additional integrals (the last term in (2.11)) can be expressed via $\boldsymbol{G}(\boldsymbol{x}_\beta^c - \boldsymbol{y})$, $\nabla\boldsymbol{G}(\boldsymbol{x}_\beta^c - \boldsymbol{y})$ and $\nabla\boldsymbol{\mathcal{P}}(\boldsymbol{x}_\beta^c - \boldsymbol{y})$, and they were calculated as such in the non-multipole scheme (cf. with Section 4.5 for the multipole code) using high-order table interpolations. Following [20], three criteria are used to quantify the deviation between $\boldsymbol{F}$ and $\boldsymbol{F}_{ex}$:

$$\delta_1(\boldsymbol{F}, \boldsymbol{F}_{ex}) = \frac{1}{\langle \boldsymbol{F}_{ex}^2 \rangle^{1/2}} \max_{\alpha, \boldsymbol{x}^i \in S_\alpha} ||\boldsymbol{F}(\boldsymbol{x}^i) - \boldsymbol{F}_{ex}(\boldsymbol{x}^i)||,$$

$$\delta_2(\boldsymbol{F}, \boldsymbol{F}_{ex}) = \frac{1}{\widehat{N} + \widetilde{N}} \sum_\alpha \left\{ \frac{1}{\langle \boldsymbol{F}_{ex}^2 \rangle_\alpha} \max_{\boldsymbol{x}^i \in S_\alpha} [\boldsymbol{F}(\boldsymbol{x}^i) - \boldsymbol{F}_{ex}(\boldsymbol{x}^i)]^2 \right\}^{1/2},$$

$$\delta_3(\boldsymbol{F}, \boldsymbol{F}_{ex}) = \left[\langle (\boldsymbol{F} - \boldsymbol{F}_{ex})^2 \rangle / \langle \boldsymbol{F}_{ex}^2 \rangle \right]^{1/2}. \tag{6.1}$$

Here, the summation in α is over all surfaces (solid and drop); $\langle \ldots \rangle_\alpha$ and $\langle \ldots \rangle$ denote averaging over S_α and all surfaces, respectively; δ_1 provides pointwise convergence control, while δ_3 is the integral measure of convergence. The deviations $\delta_i(\boldsymbol{u}, \boldsymbol{u}_{ex})$ ($i = 1, 2, 3$) are defined the same way through $\boldsymbol{u}$ and $\boldsymbol{u}_{ex}$, using the drop surfaces only in (6.1) and replacing $\widehat{N} + \widetilde{N}$ by $\widetilde{N}$; analogous for $\delta_i(\boldsymbol{q}, \boldsymbol{q}_{ex})$, using solid surfaces only in (6.1) and replacing $\widehat{N} + \widetilde{N}$ by $\widehat{N}$.

Table 2 demonstrates that all $\delta_i(\boldsymbol{F}, \boldsymbol{F}_{ex})$, $\delta_i(\boldsymbol{u}, \boldsymbol{u}_{ex})$ and $\delta_i(\boldsymbol{q}, \boldsymbol{q}_{ex})$ tend to zero, as the precision parameter ε (Section 4.6) is tightened, which proves the convergence of our code to the direct summation code. The actual relative errors δ_i, though, should not be linked to this intuitive parameter ε (the latter, in particular, does not take the actual magnitude of the solution, with small non-dimensional velocities, into account). The CPU times for our code to calculate $\boldsymbol{F}$ and to perform one boundary-integral iteration are also given in Table 2;

Table 2
The convergence of the present solution $(\boldsymbol{F}, \boldsymbol{u}, \boldsymbol{q})$ to the standard summations solution $(\boldsymbol{F}_{ex}, \boldsymbol{u}_{ex}, \boldsymbol{q}_{ex})$, as $\varepsilon \to 0$, for the snapshot $t = 129$ of Fig. 7

$\varepsilon =$	3.9×10^{-5}	3.9×10^{-6}	3.9×10^{-7}	3.9×10^{-8}	3.9×10^{-9}
$\delta_1(\boldsymbol{F}, \boldsymbol{F}_{ex})$	1.3×10^{-2}	2.2×10^{-3}	2.5×10^{-4}	3.2×10^{-5}	3.2×10^{-6}
$\delta_2(\boldsymbol{F}, \boldsymbol{F}_{ex})$	4.6×10^{-3}	6.3×10^{-4}	8.2×10^{-5}	9.9×10^{-6}	1.2×10^{-6}
$\delta_3(\boldsymbol{F}, \boldsymbol{F}_{eex})$	6.9×10^{-4}	1.1×10^{-4}	1.5×10^{-5}	2.0×10^{-6}	2.6×10^{-7}
CPU time ($\boldsymbol{F}$)	15.0	17.0	19.0	22.5	26.5
$\delta_1(\boldsymbol{u}, \boldsymbol{u}_{ex})$	2.9×10^{-2}	4.8×10^{-3}	5.8×10^{-4}	6.6×10^{-5}	7.3×10^{-6}
$\delta_2(\boldsymbol{u}, \boldsymbol{u}_{ex})$	1.8×10^{-2}	2.6×10^{-3}	3.3×10^{-4}	4.2×10^{-5}	4.9×10^{-6}
$\delta_3(\boldsymbol{u}, \boldsymbol{u}_{ex})$	2.5×10^{-3}	3.9×10^{-4}	5.6×10^{-5}	7.4×10^{-6}	9.5×10^{-7}
$\delta_1(\boldsymbol{q}, \boldsymbol{q}_{ex})$	4.2×10^{-4}	1.0×10^{-4}	9.7×10^{-6}	1.1×10^{-6}	1.2×10^{-7}
$\delta_2(\boldsymbol{q}, \boldsymbol{q}_{ex})$	3.0×10^{-4}	4.6×10^{-5}	6.2×10^{-6}	7.9×10^{-7}	9.9×10^{-8}
$\delta_3(\boldsymbol{q}, \boldsymbol{q}_{ex})$	4.8×10^{-5}	7.3×10^{-6}	1.0×10^{-6}	1.4×10^{-7}	1.9×10^{-8}
CPU time (one iter.)	20.0	24.5	30.0	37.5	46.5

The CPU times are in seconds for the present solution.

these times are weakly sensitive to ε and are two-orders of magnitude smaller than those by the impractically slow direct summation code (29 min and 66 min for $\boldsymbol{F}$ and one iteration, respectively, even with the simplest, linear interpolation for $\boldsymbol{G} - \boldsymbol{G}_0$ and $\breve{\boldsymbol{\tau}} - \breve{\boldsymbol{\tau}}_0$). In particular, for $\varepsilon = 3.9 \times 10^{-7}$ used in dynamical simulation for Fig. 7, we achieve 95-fold and 137-fold computational gains for $\boldsymbol{F}$ and the iteration time, respectively. Our previous multipole-accelerated codes [20–22] developed for purely multidrop systems in a periodic box are most efficient for large ensembles ($\widetilde{N} \sim 10^2 - 10^3$ drops) with moderate resolution ($\widetilde{N}_\triangle \sim 1000$), but the performance of those codes significantly degrades or the codes are not applicable at all when the same total number of mesh-nodes are placed on a smaller ($\widetilde{N} \sim 10$) number of surfaces; likewise, increasing the drop volume fraction has a degrading effect, since costly direct summations for close interactions have to be used more often. For emulsion shear flow with $\lambda = 3$, $c_d = 0.55$, $\widetilde{N} = 100$, $\widetilde{N}_\triangle = 1500$ (a total of 75,200 mesh-nodes) and actual precisions $\delta_i(\boldsymbol{u}, \boldsymbol{u}_{ex})$ comparable with those in the present Table 2 at $\varepsilon = 3.9 \times 10^{-7}$, we observed an 87-fold gain in the iteration speed over the standard summation method (use Table 3 of Zinchenko and Davis [21] at $\varepsilon = 3 \times 10^{-4}a$), which translates into the "efficiency factor" of $87/75,200 = 0.00116$. In the present simulation with $\varepsilon = 3.9 \times 10^{-7}$, we observe a somewhat higher efficiency factor (0.00120) in the iteration speedup, and for much less favorable conditions with a larger number of nodes (114,000 total) on 34 surfaces only, and drops and solids together taking 70% of the total space. This efficiency is achieved owing to a more general near- far-field partition (4.5), improved multipole techniques, an additional level of mesh-node decomposition (into patches) and use of special geometrical properties of spherical solid shapes (Section 4). A noticeable part (5 s) of our one iteration time in Table 2 is taken by the variational drop-to-solid and drop-to-drop double-layer near-singularity subtractions (i.e., direct calculations of the RHS of (3.6)), but we have found the simulations to not succeed with simpler alternatives. For $\varepsilon = 3.9 \times 10^{-7}$, the total preiterative part takes about 64 s (19 s for $\boldsymbol{F}$, 24 s for $\boldsymbol{\Pi}$- and $\boldsymbol{\Gamma}$-tensors in the solid-to-drop near-singularity subtractions, and 20 s for precalculating the matrices, e.g., (3.7), used in the drop double-layer singularity subtractions). Postiterative passive mesh stabilization requires only 6 s. These timings, measured at $t = 129$, are typical of every half step in the simulation of Fig. 7.

The geometrical barrier (Section 5.4) $\delta_{\min} = 0.009\widehat{a}$ was used in this simulation to prevent drop–solid overlapping, but the corresponding correction was needed for an extremely small portion of the total number of drop mesh nodes (namely, for 17, 30, 10 and 23 nodes at $t = 64$, 129, 193, and 257, respectively, out of the 75,050 nodes in total), so the lubrication is "almost resolved." This numerical trend for overlapping, already small, is sharply reduced by stepping up to higher resolution ($\widehat{N}_\triangle = 11,520$, $\widetilde{N}_\triangle = 8640$): only 7, 30, 9 and 3 nodes out of the total ≈108,000 needed a correction at $t = 64$, 129, 193 and 244, respectively. It is important to stress that the high-order near-singularity subtraction in solid-to-drop double-layer contributions (Section 3) is the crucial element in making the drop–solid overlapping trend so low in the present problem. When the run in Fig. 7 was repeated with the c-terms in Eqs. (3.11) and (3.12) disabled (resorting them to the leading-order subtraction), 500–600 nodes required a correction at every half of the time step for all $t \geqslant 25$. Moreover, by $t = 112$, the run without the high-order subtraction lost much global accuracy (for the continuous and drop phase velocities) and became three times slower than the run in Fig. 7 (due to smaller time steps and/or larger number of boundary-integral iterations) and could not proceed any further. It is imperative, therefore, to use high-order subtraction in the present type of drop squeezing problems; the geometrical barrier $\delta_{\min}$ alone is not a sufficient remedy.

Since the drop volume fraction in the space available between solids (40%, corresponding to $c_d = 0.2$) is large enough, there are also close drop–drop interactions (Fig. 7) with a numerical trend for surface overlapping, but we found it best to leave this issue without a remedy. The degree of drop–drop overlapping, though,

Table 3
Computational expenses (in seconds) for the multipole-accelerated boundary-integral solution of the emulsion flow problem at $\lambda = 4$, $\widehat{N} = 9$, $\widetilde{N} = 25$, $t = 129$, $\varepsilon = 4 \times 10^{-7}$

$\widehat{N}_\triangle$	$\widetilde{N}_\triangle$	Total number of modes	Preiterative part (s)	One boundary-integral iteration (s)
8640	6000	113,948	64 (1900)	30 (4100)
11,520	8640	159,908	111 (3860)	49 (7800)

The corresponding CPU times (accurate to a few percent) for the direct summation algorithm are given in parentheses.

was systematically monitored in our calculations, which can be done rigorously on the approximation that each drop surface is identified with a collection of *flat* mesh triangles. The indicator function

$$I\left(\boldsymbol{y},\widetilde{S}_\beta\right)=\frac{1}{4\pi}\int_{\widetilde{S}_\beta}\frac{\boldsymbol{r}\cdot\boldsymbol{n}(\boldsymbol{x})}{r^3}\mathrm{d}S_{\boldsymbol{x}},\qquad \boldsymbol{r}=\boldsymbol{x}-\boldsymbol{y} \tag{6.2}$$

equals 0 and 1 for $\boldsymbol{y}$ outside and inside $\widetilde{S}_\beta$, respectively. For flat triangulation, I is calculated exactly, since the contribution of each mesh triangle $\triangle\in\widetilde{S}_\beta$ to the integral (6.2) is simply ± the area of the spherical triangle [33] formed by projecting $\triangle$ onto the unit sphere centered at $\boldsymbol{y}$. For each of $\widetilde{N}\times(\widetilde{N}_\triangle/2+2)$ mesh-nodes $\boldsymbol{y}$ on drops it is determined, through (6.2), whether $\boldsymbol{y}$ protrudes into another drop $\widetilde{S}_\beta$ (including periodic images) and, if it does, we calculate the degree of overlapping, δ_-, as the minimum of node-to-node, node-to-face and node-to-edge distances from $\boldsymbol{y}$ to the triangulated $\widetilde{S}_\beta$. This procedure is applied at selected time moments, and the efficiency is not an issue. Fig. 10 shows the dynamics of p_- (the fraction of protruding mesh-nodes) and the

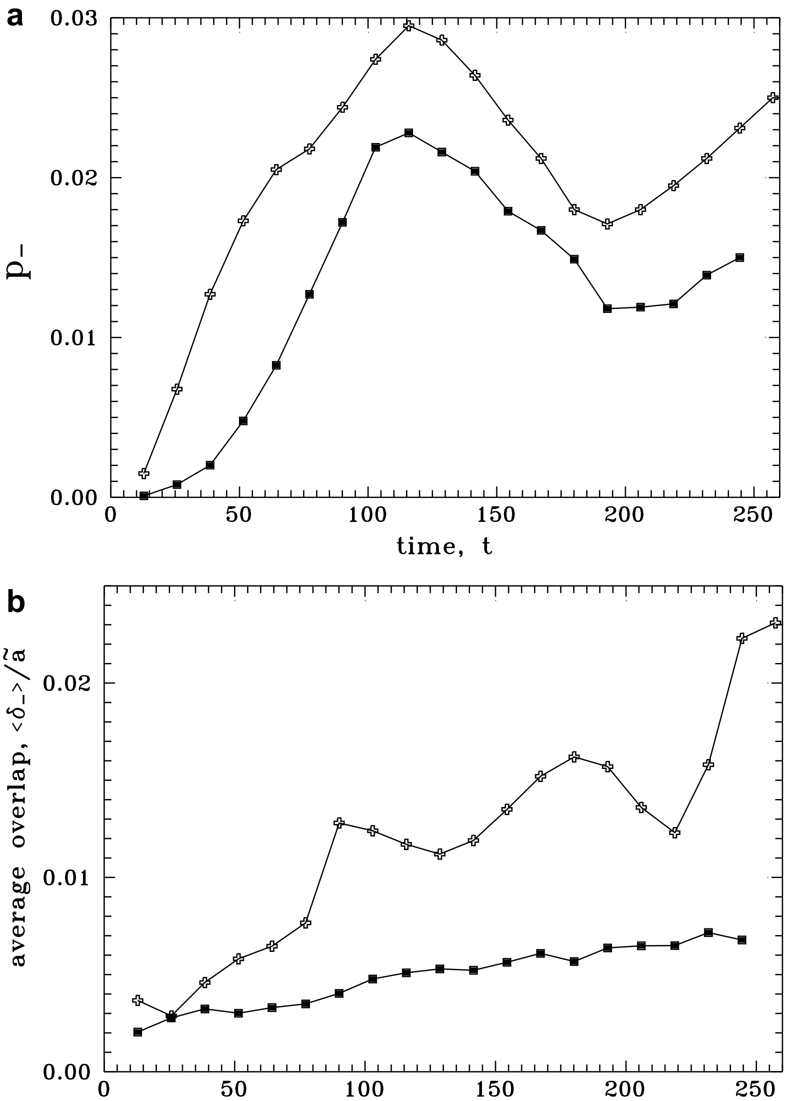


Fig. 10. The (a) portion of drop mesh-nodes protruding into another drop and (b) average overlap over the protruding nodes (relative to $\widetilde{a}$) for the simulation shown in Fig. 7 (crosses), and for a similar higher resolution simulation $\widehat{N}_\triangle=11,520$, $\widetilde{N}_\triangle=8640$ (squares).

average overlap $\langle\delta_-\rangle$ (i.e., δ_- averaged over protruding nodes only) relative to the drop non-deformed radius $\widetilde{a}$ for the simulation in Fig. 7 and a similar higher resolution simulation. For $\widehat{N}_\Delta = 8640$, $\widetilde{N}_\Delta = 6000$, there is only about 2% of protruding nodes, with the average overlap $\langle\delta_-\rangle \approx 0.015\widetilde{a}$ at the advanced stage of drop squeezing; for higher resolution ($\widehat{N}_\Delta = 11,520$, $\widetilde{N}_\Delta = 8640$), the average overlap decreases 2–2.5 times.

Fig. 11 shows the surface triangulation effects on the main quantities of interest – continuous (U_C) and drop phase (U_D) velocities (in the z-direction) for the simulation in Fig. 7, and for higher resolution $\widehat{N}_\Delta = 11,520$, $\widetilde{N}_\Delta = 8640$. The deviations between the two solutions for U_C remain modest in the entire simulation range (Fig. 11a). For U_D, the agreement is excellent up to $t = 116$ (Fig. 11b), with only minor deviations at larger times; the cruder solution (solid lines) is capable of capturing all the wiggles in $U_C(t)$ and $U_D(t)$ correctly, as the drops squeeze through the material. Due to stochastic nature of the process, the long-time behavior is inevitably sensitive to the details of numerical implementation; it is essential, though, that our solution retains accuracy in the time range practically sufficient for the statistically stationary regime to be

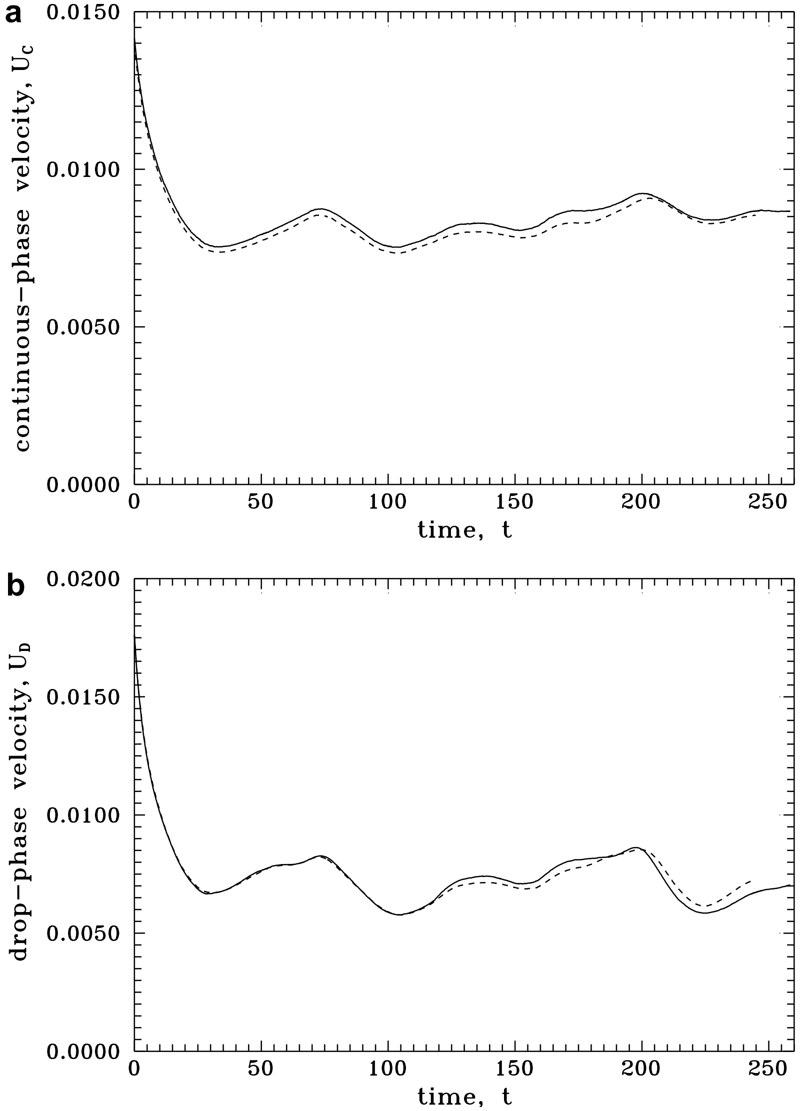


Fig. 11. The (a) continuous phase and (b) drop phase velocities in the z-direction for the simulation in Fig. 7 (solid lines) and for a similar higher resolution simulation $\widehat{N}_\Delta = 11,520$, $\widetilde{N}_\Delta = 8640$ (dashed lines).

established. For $\widehat{N}_\triangle = 11,520$, $\widetilde{N}_\triangle = 8640$, the preiterative part and one iteration in the boundary-integral solution take 111 and 49.5 s, respectively, at $t = 129$, with an approximately linear scaling in the total number of mesh-nodes (cf. Table 2 at $\varepsilon = 3.9 \times 10^{-7}$). Contrary to the expected scaling $\sim \widetilde{N}_\triangle^{-1/2}$, the average time step for $\widehat{N}_\triangle = 11,520$, $\widetilde{N}_\triangle = 8640$ is practically the same as for the lower resolution run in Fig. 7, since higher triangulations somewhat reduce the curvature factor $\max(k_1, k_2)$ in the time step criterion (5.12). Computational expenses of our code at $t = 129$ are summarized and compared to those by the direct summation algorithm in Table 3.

We have also studied the cumulative effects of the time integration scheme and the precision parameter ε on U_C and U_D in dynamical simulations with $\widehat{N}_\triangle = 8640$, $\widetilde{N} = 6000$ (Figs. 12 and 13); in these figures, small times $t \leqslant 15$ with no appreciable effects of numerical implementation are cut off. When the second-order RK scheme is replaced by the Euler scheme, the differences are virtually unnoticeable until $t = 135$ and do not exceed 2% and 1.5% for U_D and U_C, respectively, at larger times in the simulated range (Fig. 12); the triangulation effects are much more significant, especially for U_C (cf. Fig. 11). The cumulative effects of the precision parameter ε, when it is coarsened to $\varepsilon = 3.9 \times 10^{-6}$, are practically unnoticeable until $t = 125$, but larger than expected for subsequent evolution, reaching 3.7% and 2% for U_D and U_C, respectively (Fig. 13; cf. Table 2 for an instantaneous configuration). These differences justify the use of higher precisions $\varepsilon \sim 4 \times 10^{-7}$ in dynamical simulations; fortunately, the CPU time is only slightly sensitive to ε (Table 2).

For the homoviscous case $\lambda = 1$, the boundary-integral iterations are greatly optimized (Section 5.4), and larger systems can be handled. Figs. 14 and 15 present snapshots of our simulation with $Ca = 0.75$, $\widehat{N} = 14$ solids, $\widetilde{N} = 40$ drops, $\widehat{N}_\triangle = 11,520$ and $\widetilde{N}_\triangle = 8640$ elements on each solid and drop surface, respectively; in Fig. 14, only the initial and final configurations of both particles and drops are shown, while Fig. 15 demonstrates the motion of the drop phase. The solid particle arrangement at $c_s = 0.5$ was prepared (unlike in Fig. 7) by the standard Monte–Carlo method [43]; the coordinates of the particle centers are listed in Appendix B. Such a configuration, though, is lacking solid contact mechanical support and was chosen for test purposes only. For drops with $c_d = 0.2$, we used the swelling algorithm of Subsection 5.3 to prepare a start-up configuration. The whole simulation in Figs. 14 and 15 took about 8100 second-order RK steps, with typically 5–6 iterations on each half of a time step to find the Hebeker density $\boldsymbol{q}(\boldsymbol{x})$ from the decoupled boundary-integral equation (Subsection 5.4). In one boundary-integral solution at $t = 116$, the preiterative part takes 136 s, followed by the decoupled iterations for $\boldsymbol{q}(\boldsymbol{x})$ (each requiring 13 s); the postiterative part, to calculate $\boldsymbol{w}(\boldsymbol{y})$ from $\boldsymbol{q}(\boldsymbol{x})$, takes 35 s. These timings are typical for each half step in the simulation of Figs. 14 and 15.

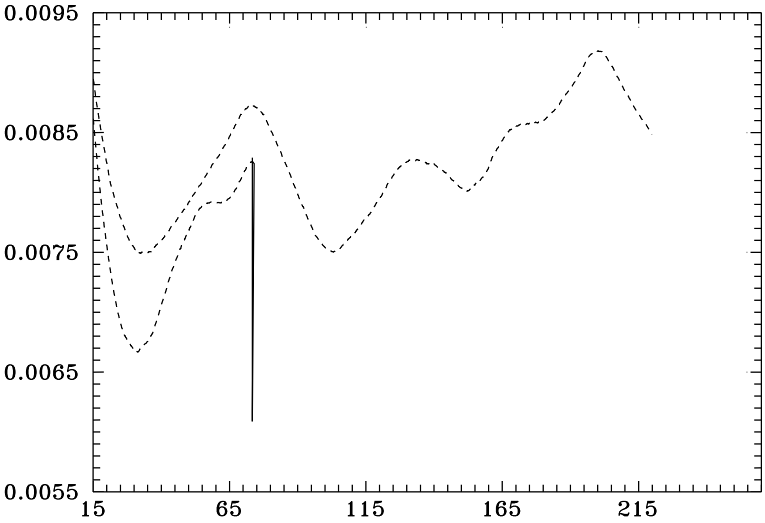


Fig. 7using the second-order RK scheme. Dashed lines: the same runby the Euler scheme.7878

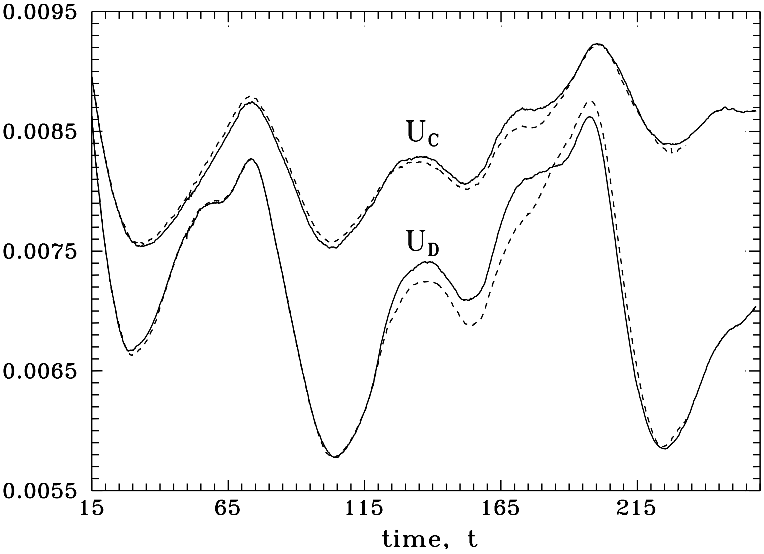


Fig. 13. The cumulative effect of the precision parameter on the continuous phase (the top two curves) and the drop phase (the bottom two curves) velocities. Solid lines: simulation in Fig. 7 with $\varepsilon = 3.9 \times 10^{-7}$. Dashed lines: the same run with $\varepsilon = 3.9 \times 10^{-6}$.

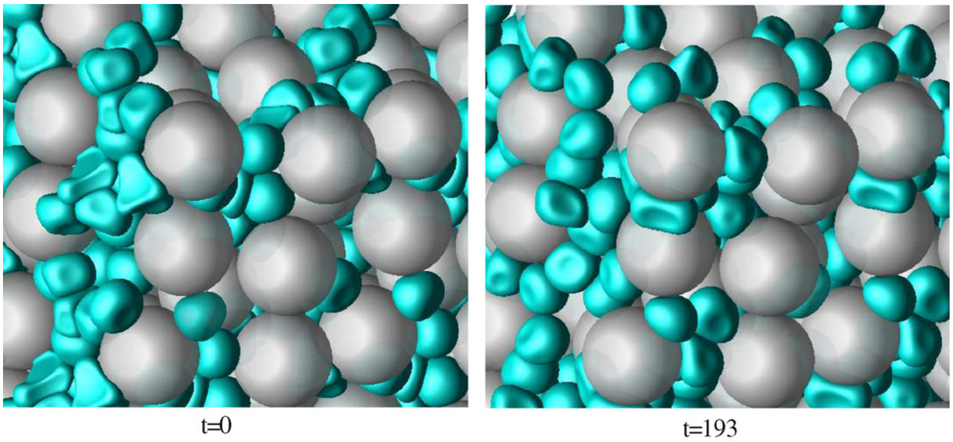


Fig. 14. The initial and final snapshots of the emulsion squeezing through a random array of solid spheres at $Ca = 0.75$, $\lambda = 1$, $c_s = 0.5$, $c_d = 0.2$, $\widehat{N} = 14$, $\widetilde{N} = 40$, $\widehat{N}_\Delta = 11,520$, $\widetilde{N} = 8640$, $\widehat{M}_\Delta = 60$, $\widetilde{M}_\Delta = 20$, $\varepsilon = 3.9 \times 10^{-7}$, $k_0 = 35$, $\eta = 1/\widehat{a}$, $\delta_{\min} = 0.009\widehat{a}$, second-order RK scheme.

Computational expenses for this, and a similar lower resolution ($\widehat{N}_\Delta = 8640$, $\widetilde{N}_\Delta = 6000$) run at $t = 116$ are summarized and compared to those by the direct summation algorithm in Table 4. Again, a non-multipole approach would not be capable of dynamical simulations in the present problem.

Due to reduced lubrication, it is even more difficult for $\lambda = 1$ to completely avoid drop–drop overlapping than for $\lambda = 4$. Again, however, the average overlap $\langle \delta_- \rangle / \widetilde{a}$ over drop mesh nodes protruding into another drop systematically reduces, albeit slowly, as more triangles are used (Table 5). Presumably, the present simulations would require much higher resolution to completely eliminate this artifact (in contrast, simpler problems of two-drop interactions in free-space at finite deformations need much less triangulations to avoid overlapping [30]). The global dynamical quantities of interest (the continuous and drop phase velocities),

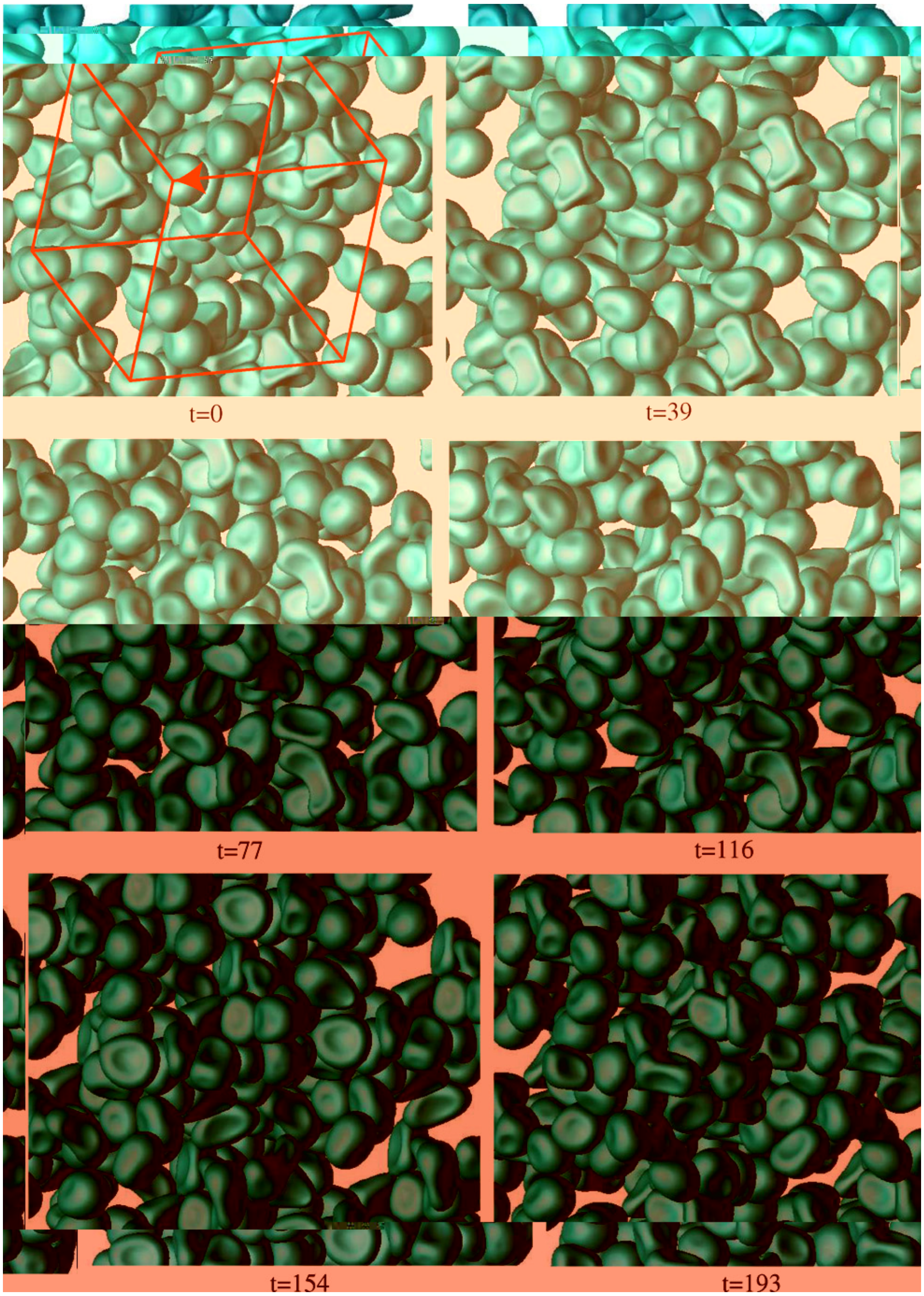


Fig. 15. Snapshots of the drop phase for the simulation in Fig. 14. The periodic box $[0,1)^3$ and the direction of $-\langle\nabla p\rangle$ (from right to left) common for all frames (and for Fig. 14) are also shown.

though, are well convergent with respect to triangulations (see below) and do not suffer appreciably from this weak trend for drop–drop overlapping. Aggressive mesh adaption to better resolve lubrication (which can be only achieved at the expense of *global* convergence and the time step) does not appear justified in the present problem. Finally, we note that our drop–drop overlapping criterion is based on the indicator function (6.2) *neglecting the curvature of the elements* and, hence, is approximate itself.

Table 4
Computational expenses (in seconds) for the multipole-accelerated boundary-integral solution of the emulsion flow problem at $\lambda = 1$, $\widehat{N} = 14\,\widetilde{N} = 40$, $t = 116$, $\varepsilon = 4 \times 10^{-7}$

$\widehat{N}_\triangle$	$\widetilde{N}_\triangle$	Preiterative part (s)	One Hebeker density iteration (s)	Postiterative part (s)
8640	6000	80 (5090)	10 (1840)	25 (8300)
11,520	8640	136 (10,200)	13 (3160)	35 (15,600)

The corresponding CPU times for the direct summation algorithm are given in parentheses. For details, see the text.

Table 5
The portion p_- of drop mesh-nodes protruding into another drop and the average overlap $\langle\delta_-\rangle/\widetilde{a}$ over the protruding nodes for the simulation shown in Figs. 14 and 15, and for a similar lower resolution simulation $\widehat{N}_\triangle = 8640$, $\widetilde{N} = 6000$

t	$\widehat{N}_\triangle = 8640$, $\widetilde{N}_\triangle = 6000$		$\widehat{N}_\triangle = 11,520$, $\widetilde{N}_\triangle = 8640$	
	p_-	$\langle\delta_-\rangle/\widetilde{a}$	p_-	$\langle\delta_-\rangle/\widetilde{a}$
12.9	0.000	0.002	0.000	0.003
25.7	0.022	0.003	0.010	0.002
38.6	0.045	0.007	0.031	0.004
51.5	0.063	0.011	0.052	0.007
64.3	0.071	0.016	0.062	0.011
77.2	0.072	0.022	0.065	0.015
90.1	0.073	0.027	0.066	0.020
102.9	0.075	0.031	0.066	0.022
115.8	0.081	0.032	0.070	0.023
128.7	0.086	0.035	0.076	0.024
141.5	0.087	0.041	0.074	0.029
154.4	0.089	0.045	0.074	0.032
167.3	0.088	0.050	0.073	0.035
180.1	0.084	0.055	0.067	0.041
193.0	0.084	0.060	0.063	0.048

In this simulation, we could not proceed beyond $t \approx 193$ due to difficulties with mesh-node redistribution (through (5.8)) for one drop, experiencing fast elongation. It remains to be seen in future work if an improved drop meshing algorithm would allow us to proceed further. The simulated time range, though, is sufficient for the squeezing process to attain a statistical steady state, both for $U_C(t)$ and $U_D(t)$. Fig. 16a and b presents the continuous and drop phase velocities for the simulation in Figs. 14 and 15 (solid lines), and for the same parameters with lower resolution $\widehat{N}_\triangle = 8640$, $\widetilde{N}_\triangle = 6000$ (dashed lines). The deviation between the two solutions for $U_C(t)$ is within 4% for the entire time range (Fig. 16a); presumably, accuracy could be improved with a more optimal value of $\eta = 3/\widehat{a}$ of the Hebeker parameter (cf. Table 1 for the single-phase flow). In contrast, for the drop phase velocity (Fig. 16b), the convergence is excellent for all $t \leqslant 100$ and remains good up to $t \approx 145$, faltering only at larger times – probably due to difficulties with drop mesh-node redistribution. The effect of switching to the first-order time integration (not shown) is even much smaller than in Fig. 12 making, again, the more economical Euler scheme quite adequate for the problem; in retrospect, we have found the same to be true for the simulations of a single drop squeezing through a finite cluster in Zinchenko and Davis [6].

All the simulations so far in this subsection used an artificial parameter $\delta_{\min} = 0.009\widehat{a}$ to prevent numerical drop–solid overlapping (Section 5.4), which may be questionable until the sensitivity of the solution to this parameter is explored. The simulation in Figs. 14 and 15 were repeated for $\delta_{\min} = 0.0045\widehat{a}$, with a remarkably small effect on the continuous and drop phase velocities in almost the whole time range (Fig. 17a and b); the maximum deviation between the two solutions is about 0.7% for U_C and 1% for U_D for t up to 170, with no appreciable differences at $t \leqslant 70$. In the simulation from Figs. 14 and 15 with $\delta_{\min} = 0.009\widehat{a}$, only 37 drop mesh-nodes (out of total $\approx 173,000$) needed the geometrical correction at $t = 116$; for the simulation with

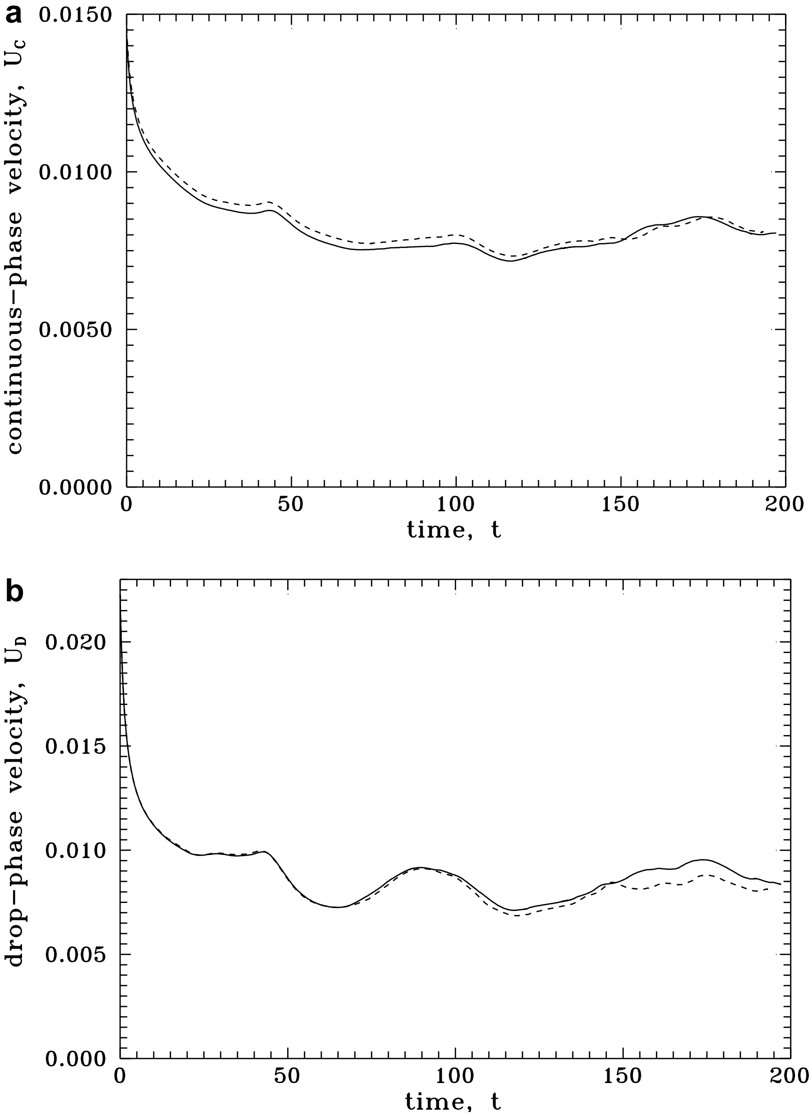


Fig. 16. The (a) continuous and (b) drop phase velocities (in the z-direction) for the simulation in Figs. 14 and 15 (solid lines) and for a similar lower resolution run $\widehat{N}_\Delta = 8640$, $\widetilde{N}_\Delta = 6000$ (dashed lines).

$\delta_{\min} = 0.0045\widehat{a}$, only 32 nodes required this correction at the same $t = 116$. We note that our high resolution runs "almost resolve" lubrication in the drop–solid near-contact areas; those very few nodes requiring the artificial correction do not have an appreciable effect on the overall squeezing process. The parameter $\delta_{\min}$ should not be set arbitrarily small for efficiency reasons, since too close unresolved drop–solid interactions can increase the local curvatures $k_1(\boldsymbol{x}^i)$, $k_2(\boldsymbol{x}^i)$ (especially when drops leave constrictions) and the total number of time steps by the stability criterion (5.12). For example, the solution for $\delta_{\min} = 0.0045\widehat{a}$ is 1.62 times more expensive than for $\delta_{\min} = 0.009\widehat{a}$. Besides, for the most challenging case of squeezing conditions close to critical not considered here, we expect the solution to be more sensitive to $\delta_{\min}$, and even higher triangulations would be required for adequate simulations.

Several physical trends in Figs. 11 and 16 are interesting to discuss. In a steady shear flow of an emulsion of freely suspended deformable drops, a statistical steady state is always attained (leaving aside breakup conditions at large Ca), which was simulated both for large and small systems [21,52]. In contrast, for sedimentation of many deformable drops [20,22], there is no steady state even away from breakup conditions and the drops

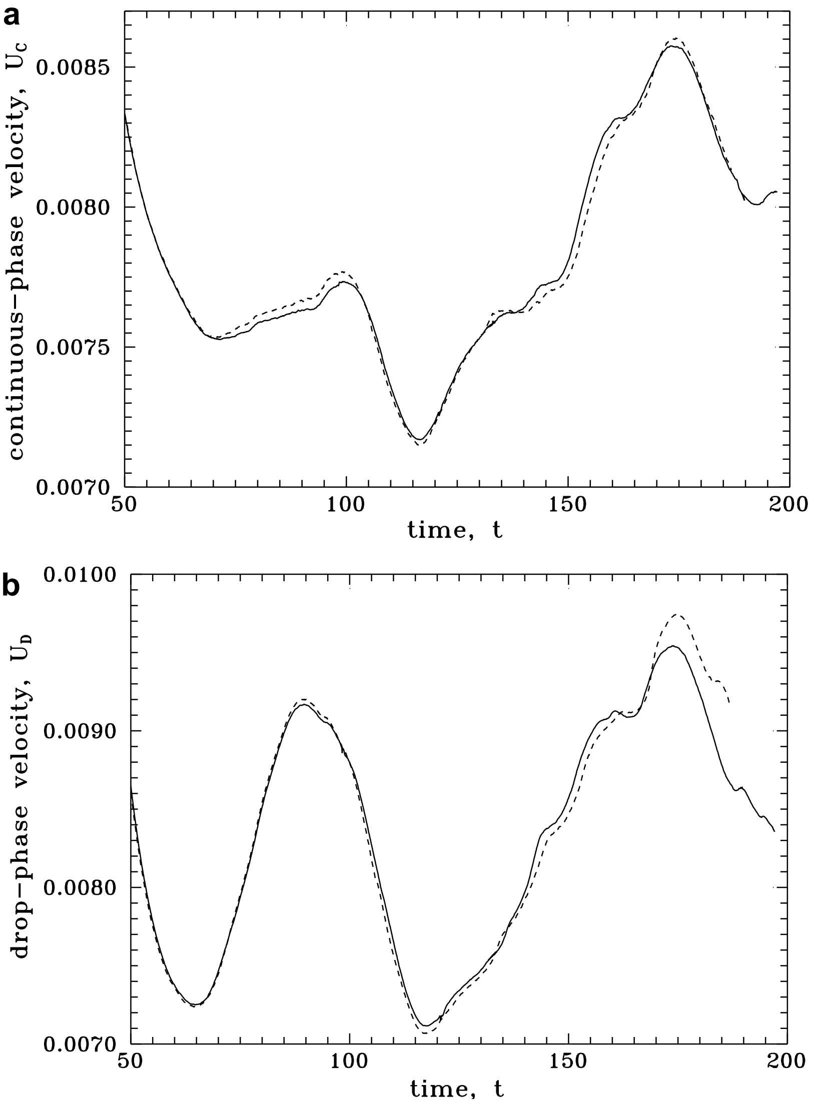


Fig. 17. The effect of the geometrical barrier $\delta_{\min}$ on the (a) continuous and (b) drop phase velocities. Solid lines: simulation in Figs. 14 and 15 with $\delta_{\min} = 0.009\widehat{a}$. Dashed lines: the same run with $\delta_{\min} = 0.0045\widehat{a}$.

cluster instead, which we attributed to the Koch–Shaqfeh [53] type of instability. Figs. 11 and 16 show that, for the pressure-driven flow of freely suspended drops through a granular material, there is no clustering and a well-defined statistical steady state is achieved. The approach to this state is surprisingly fast, after the drops pass the constrictions just 1–2 times, on the average. The long-time average velocities $\langle U_C \rangle$ and $\langle U_D \rangle$ are not very much different, but both are significantly smaller than the average continuous phase velocity for the motion of a pure liquid with viscosity μ^e through the same particle arrangement under the same pressure gradient $\langle \nabla p \rangle$ (2.9–3.2 times for the $\lambda = 4$ simulation, and 2.2–2.4 times for $\lambda = 1$). On the average, in the long-time regime, the drops move slower than the continuous phase in the $\lambda = 4$ simulation, but faster than the continuous phase for $\lambda = 1$. The latter case, although it looks unusual, finds an easy explanation: U_C includes volume averaging of the fluid velocity $\boldsymbol{u}$ outside the drops, with $\boldsymbol{u} \approx \boldsymbol{0}$ near the solid boundaries, and so $U_C < U_D$ is quite possible. A similar hydrodynamical phenomenon, when red cells move faster than the surrounding plasma, is known for blood flows in capillaries [54]. As the conditions approach critical for squeezing to occur in our problem, $U_D \rightarrow 0$ is expected, but at present this case would be extremely difficult to simulate for multidrop-multiparticle systems.

Despite dramatic gains over the standard boundary-integral method achieved in our code through multipole acceleration, overall the calculations are expensive, due to inevitable high surface resolution, the very large number of time steps, and an iterative solution at each step. For example, the simulation for the parameters in Fig. 7 even by the most economical Euler scheme took about 40 days on a single processor of AMD PC, with Opteron 2.8 GHz CPU. For the $\lambda = 1$ case in Figs. 14 and 15, the Euler scheme integration took about 23 days. Further progress will be made primarily by parallelizing the code.

7. Conclusions

A 3D algorithm has been implemented, for the first time, capable of modelling an emulsion flow through a dense random granular material by direct multidrop-multiparticle simulations in a periodic box at low Reynolds numbers. The algorithm incorporates recent boundary-integral desingularization tools developed for a single deformable drop squeezing through a finite cluster [6], periodic Green function, Hebeker representation for solid–particle contributions, unstructured surface triangulations with fixed topology, and multipole acceleration. Aside from dynamical simulations, our work also features a new “swelling” algorithm to prepare a start-up configuration of deformed drops with sufficiently high volume fraction between solid particles. Compared to our previous purely multidrop 3D simulations [20–22], the new problem is literally on the next level of difficulty because of strong, lubrication-sensitive drop–solid interactions and high total volume fraction of particles and drops. We found that the problem requires very high resolution (at least, $N_\Delta \sim 10^4$ boundary elements per surface), a large ($\sim 10^4$) number of time steps, and an iterative solution (~ 10 iterations) at each step, to successfully simulate squeezing of drops comparable in size with the particles, with the practical goal of the long-time pressure gradient-flow rate relationships for the continuous and drop phases. The multipole acceleration part, with two levels of mesh-node decomposition, is an extension and improvement of our previous codes [20–22]. It now has no lower limitation on the system size, and remains efficient for high triangulation ($N_\Delta \geqslant O(10^4)$) and high total volume fractions. A two-orders of magnitude gain through multipole acceleration over the standard boundary-integral coding has made the present simulations feasible. Also crucial in the present work is the use of two non-standard desingularization tools – a variational technique for drop double-layer integrals and high-order near-singularity subtraction in solid-to-drop double-layer contributions. The former improves the spectral properties of the boundary-integral equations, allowing us to avoid non-convergence of iterations, while the latter was found to greatly reduce the numerical trend for drop–solid overlapping. The current version of our code is operational for spherial solid particles (mono- or polydisperse), but an extension to other canonic shapes (spheroidal or ellipsoidal) would be straightforward.

Using this code, we simulated the squeezing of 25–40 drops through a random array of 9–14 solid spheres with $N_\Delta \approx 10^4$ at 50% solid and 20% drop volume fractions and viscosity contrasts $\lambda = 4$ and 1. Relaxation to a statistically steady flow regime is surprisingly fast, after drops pass the constrictions 1–2 times, on the average. For $\lambda = 4$ drops travel slower than the continuous phase, but the opposite was observed for $\lambda = 1$, on the average in the long-time regime. Variation of program parameters (N_Δ, etc.) and convergence tests prove the correctness and accuracy of our simulations in a long-time range, sufficient for the statistically steady regime to establish. In particular, there remains a very weak numerical trend for drop–solid overlapping, which we have found impossible to eliminate altogether by increasing triangulations and which had to be contained by an artificial geometrical barrier, but, most importantly, the global quantities of interest (continuous and drop phase velocities) were found to be virtually independent of this procedure.

The present research can be further developed in several ways. We have recently applied the present algorithm to study the special case of a periodic emulsion flowing through a packed cubic array of particles (one drop and one particle per cell), where many more results can be obtained, including critical conditions for squeezing to occur [55]. For random materials, when an emulsion squeezes with high resistance, it is still possible, due to geometry fluctuations, that some drops will experience very large deformations and even breakup, while most drops remain compact. Such situations would require a more versatile drop meshing algorithm, which presumably can be achieved by combining the present approach with topological mesh transformations [24,39,40,56–58]. As for larger multidrop-multiparticle simulations, they are still very demanding despite dramatic gains achieved in our code through multipole acceleration, and we will seek further progress by parallelizing the code.

Acknowledgement

This work was supported by the National Aeronautics and Space Administration and the Petroleum Research Fund of the American Chemical Society.

Appendix A. Details of economical truncation

For optimal values of the Hebeker parameter, η, single- and double-layer parts of the solid particle contributions are equally important, so we base, for simplicity, the truncation bounds on the behavior of Lamb's series (4.3) for the double-layer part, estimated as [20]

$$\sum_{k=1}^{\infty} \breve{a}_k^{(\beta)} \left(\frac{d_\beta^0}{R_\beta}\right)^{k+1}, \quad \breve{a}_k^{(\beta)} = \begin{cases} \breve{C}_\beta/k^2, & k<k_{cr} \\ \breve{C}_\beta k\Big/k_{cr}^3, & k \geqslant k_{cr}. \end{cases} \tag{A.1}$$

The low-k behavior of $\breve{a}_\beta^{(k)}$ models continuous distribution of stresslets over S_β (appropriate for fine triangulations), while the ultimate asymptotics ($k \to \infty$) reflects the singularity of the actual discrete distribution (4.4); the switch value k_{cr} is determined semi-empirically by surface discretization [20]. The factor $\breve{C}_\beta$ is estimated as [20]

$$\breve{C}_\beta = \frac{3}{4\pi\left(d_\beta^0\right)^2} \| \sum_{\boldsymbol{x}^j \in S_\beta} \boldsymbol{Q}(\boldsymbol{x}^j)\boldsymbol{w}(\boldsymbol{x}^j)\|, \tag{A.2}$$

with $\boldsymbol{W}(\boldsymbol{x}^j)$ defined in Subsection 4.2. The same relations (A.1) and (A.2) are used to estimate Lamb's series (4.3) for a drop S_β. For patch series (4.2), our estimations are analogous, with S_β and β replaced by $\mathcal{B}_\gamma$ and γ, respectively, in (A.1) and (A.2).

When the shells $\mathcal{D}_\beta^{\min} + \boldsymbol{m}$ and $\mathcal{D}_\alpha$ do not overlap, the truncation bounds $v_{nf}+1$ and $n_{nf}+1$ on v and n, respectively, in the reexpansion from (4.3) to (4.7) are found from

$$\sum_{n,k:n>n_{nf} \text{ or } k>v_{nf}} \breve{a}_k^{(\beta)} \frac{(k+n)!\left(d_\alpha^0\right)^n \left(d_\beta^0\right)^{k+1}}{k!n!\left|\boldsymbol{R}_{\beta\alpha}-\boldsymbol{m}\right|^{n+k+1}} < \varepsilon_1,$$

$$\varepsilon_1 = \left[\sum_\beta \sum_{|\boldsymbol{m}|\leqslant m_0} \frac{1}{\left|\boldsymbol{R}_{\beta\alpha}-\boldsymbol{m}\right|^4}\right]^{-1} \frac{\breve{e}_{nf}\varepsilon}{\left|\boldsymbol{R}_{\beta\alpha}-\boldsymbol{m}\right|^4}, \tag{A.3}$$

where $\breve{e}_{nf} = \mathrm{O}(1)$ is a numerical factor found empirically, and $\boldsymbol{m} = \boldsymbol{0}$ is excluded from the summation when $\alpha = \beta(\boldsymbol{R}_{\beta\alpha} = \boldsymbol{0})$. Relations (A.3) modify (3.87) and (3.88) of Ref. [20], but otherwise the technique for finding v_{nf} and n_{nf} (described by Eqs. (3.88)–(3.89) therein) is the same. The form (A.3) is designed to make the sum of the omitted near-field contributions from all β and $|\boldsymbol{m}| \leqslant m_0$ less than $\breve{e}_{nf}\varepsilon$.

If $|\boldsymbol{R}_{\beta\alpha} - \boldsymbol{m}| < d_\alpha^0 + d_\beta^0$, or $n_{nf} > k_0$, or $v_{nf} > k_0$, then $S_\beta^{\min} + \boldsymbol{m}$, by definition, is "not sufficiently separated" from S_α, and the reexpansion from (4.3) to (4.7) is not used. Instead (assuming $S_\alpha \neq S_\beta^{\min} + \boldsymbol{m}$), a truncation bound $v_{nf}^* + 1$ on v for direct calculation of Lamb's series (4.3) (with $\boldsymbol{R}_\beta \leftarrow \boldsymbol{y} - \boldsymbol{x}_\beta^{0,\min} - \boldsymbol{m}$) is found for every $\boldsymbol{y} \in S_\alpha$ outside the shell $\mathcal{D}_\beta^{\min} + \boldsymbol{m}$ from

$$\sum_{k=v_{nf}^*+1}^{\infty} \breve{a}_k^{(\beta)} \left(\frac{d_\beta^0}{R_\beta}\right)^{k+1} < \varepsilon_1. \tag{A.4}$$

In turn, if $v_{nf}^* > k_0$, or $|\boldsymbol{R}_\beta| < d_\beta^0$, cumulative Lamb's series (4.3) is not used. Instead, we consider contributions (4.1) from individual patches on $S_\beta^{\min} + \boldsymbol{m}$ and find a truncation bound $v_{nf}^{**} + 1$ on v for each of Lamb's series (4.2) (with $\boldsymbol{R}_\gamma \leftarrow \boldsymbol{y} - \boldsymbol{x}_\gamma^0 - \boldsymbol{m}_{\alpha\beta} - \boldsymbol{m}$, $|\boldsymbol{R}_\gamma| > d_\gamma^0$) from

$$\sum_{k=v_{nf}^{**}+1}^{\infty} \breve{a}_k^{(\gamma)} \left(\frac{d_\gamma^0}{R_\gamma}\right)^{k+1} < \left[\sum_\gamma \frac{1}{R_\gamma^2}\right]^{-1} \frac{\breve{e}_{nf}\varepsilon}{R_\gamma^2}, \tag{A.5}$$

the summation in the RHS being over all patches $\mathcal{B}_\gamma \subset S_\beta$. Further choice between using Lamb's series (4.2) (truncated at $v = v_{nf}^{**}$) or direct summation (4.1) (with $\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{m}_{\alpha\beta} - \boldsymbol{m}$) depends on the condition $v_{nf}^{**} \leqslant k_0$ and comparison of the number of operations in (4.9) and (4.1). For the special case of "self-interactions" ($S_\alpha = S_\beta = S_\beta^{\min}$, and $\boldsymbol{m} = \boldsymbol{0}$), our multipole operations also include reexpansion from a singular Lamb series (4.2) for a patch $\mathcal{B}_\gamma \subset S_\alpha$ to a regular form around $\mathcal{B}_\delta \subset S_\alpha$. The corresponding bounds v_{nf} and n_{nf} on v and n for each pair ($\mathcal{B}_\gamma$, $\mathcal{B}_\delta$) with $R_{\delta\gamma} = |\boldsymbol{x}_\gamma^0 - \boldsymbol{x}_\delta^0| > d_\gamma^0 + d_\beta^0$ are calculated from

$$\sum_{n,k:n>n_{nf} \text{ or } k>v_{nf}} \breve{a}_k^{(\gamma)} \frac{(k+n)!\left(d_\delta^0\right)^n \left(d_\gamma^0\right)^{k+1}}{R_{\delta\gamma}^{n+k+1}} < 10\varepsilon_2,$$

$$\varepsilon_2 = \left[\sum_\gamma \frac{1}{R_{\delta\gamma}^2}\right]^{-1} \frac{\breve{e}_{nf}\,\varepsilon}{R_{\delta\gamma}^2}. \tag{A.6}$$

The summation in (A.6) is over all patches $\mathcal{B}_\gamma \subset S_\alpha$, $\gamma \neq \delta$. If $n_{nf} > k_0$, or $v_{nf} > k_0$, or $R_{\delta\gamma} < d_\gamma^0 + d_\beta^0$, then, by definition, $\mathcal{B}_\gamma$ is not "sufficiently separated" from $\mathcal{B}_\delta$ and, instead of singular-to-regular reexpansion, we consider bounds $v_{nf}^{*} + 1$ on v for direct calculation of Lamb's series (4.2) at $\boldsymbol{y} \in \mathcal{B}_\delta$, $|\boldsymbol{R}_\gamma| > d_\gamma^0$, determined from

$$\sum_{k=v_{nf}^{**}+1}^{\infty} \breve{a}_k^{(\gamma)} \left(\frac{d_\gamma^0}{R_\gamma}\right)^{k+1} < \frac{\varepsilon_2}{2} \quad . \tag{A.7}$$

Once $v_{nf}^{**} > k_0$, or $|R_\gamma| < d_\gamma^0$, or (4.9) is not advantageous ($v_{nf}^{**} > 1.5|\mathcal{B}_\gamma|^{1/2}$), direct summation (4.1) is used instead.

For far-field expansion (4.45), fast convergent at $m_0 \geqslant 2$, careful construction of truncation bounds is less important. The minimum requirement is to always retain terms with $n + v \leqslant 2$, so that the truncation effect can be represented as an absolutely-convergent series, as if S_α were interacting in free-space with all periodic replicas $S_\beta^{\min} + \boldsymbol{m}$ of S_β in the far-field zone ($|\boldsymbol{m}| > m_0$). The far-field bounds on v, n and $n + v$ are constructed by adapting Eqs. (3.93)–(3.97) from Ref. [20] with $\boldsymbol{m} \neq \boldsymbol{0}$ replaced by $|\boldsymbol{m}| > m_0$; unlike in that paper, with the present approach far-field self-interaction ($\boldsymbol{R}_{\beta\alpha} = \boldsymbol{0}$) does not require a separate treatment. The parameters $\widetilde{e}_{ff}$ (Ref. [20]) for the far-field and $\breve{e}_{nf}$ (above) for near-field truncations were set to 10 and 0.2, respectively, in the present calculations. With $\breve{e}_{nf}$, $\widetilde{e}_{ff}$, k_0 and m_0 fixed, our truncation scheme for the iterative part depends on a single intuitive precision parameter, ε.

Appendix B. Solid particle configurations

For reproducibility purposes, below we list the solid–particle coordinates for the simulations of Section 6. The initial drop arrangements (prior to mixing and swelling) are not given, since they had a weak effect on the boundary-integral simulations.

See Appendix Tables 6 and 7

Table 6
The (x, y, z) coordinates of the particle centers in $[0, 1)^3$ for the simulation shown in Fig. 7

Particle number	x	y	z
1	0.3243874	0.7011747	0.4686981
2	0.0037587	0.5376121	0.8459543
3	0.8087956	0.9205165	0.3737008
4	0.0348363	0.3276157	0.2725578
5	0.4361551	0.5143805	0.0448221
6	0.2437784	0.1659034	0.6691010
7	0.1773149	0.9197166	0.0716289
8	0.6635986	0.8706258	0.8247929
9	0.6584930	0.3742273	0.5610833

Table 7
The (x, y, z) coordinates of the particle centers in $[0, 1)^3$ for the simulation shown in Figs. 14 and 15

Particle number	x	y	z
1	0.9494136	0.2078018	0.0926014
2	0.8056486	0.3315064	0.6724226
3	0.0615986	0.6598202	0.5259929
4	0.2566758	0.8913835	0.1755777
5	0.1306606	0.0926785	0.5496345
6	0.3959172	0.3806419	0.6850131
7	0.5416411	0.0828344	0.4292147
8	0.1190271	0.5788258	0.9378737
9	0.6070422	0.6600754	0.4329414
10	0.3894115	0.8368537	0.7521064
11	0.5386848	0.1437762	0.9916194
12	0.6538206	0.6043917	0.9808043
13	0.8513389	0.9239146	0.8112634
14	0.3216490	0.4072113	0.2634067

References

[1] A.S. Sangani, A. Acrivos, Slow flow through a periodic array of spheres, Int. J. Multiphase Flow 8 (1982) 343.

[2] A.A. Zick, G.M. Homsy, Stokes flow through periodic arrays of spheres, J. Fluid Mech. 115 (1982) 13.

[3] A.J.C. Ladd, Hydrodynamic transport coefficients of random dispersions of hard spheres, J. Chem. Phys. 93 (1990) 3484.

[4] G. Mo, A.S. Sangani, A method for computing Stokes flow interactions among spherical objects, Phys. Fluids 6 (1994) 1637.

[5] A.M. Chapman, J.J.L. Higdon, Oscillatory Stokes flow in periodic porous media, Phys. Fluids A 4 (1992) 2099–2116.

[6] A.Z. Zinchenko, R.H. Davis, A boundary-integral study of a drop squeezing through interparticle constrictions, J. Fluid Mech. 564 (2006) 227–266.

[7] D.R. Graham, J.J.L. Higdon, Oscillatory flow of droplets in capillary tubes. Part 1. Straight tubes, J. Fluid Mech. 425 (2000) 31–53.

[8] D.R. Graham, J.J.L. Higdon, Oscillatory flow of droplets in capillary tubes. Part 2. Constricted tubes, J. Fluid Mech. 425 (2000) 55–57.

[9] F.-K. Hebeker, Efficient boundary element methods for three-dimensional exterior viscous flow, Numer. Meth. PDE 2 (1986) 273–297.

[10] G. Zhu, A.A. Mammoli, H. Power, A 3-D indirect boundary element method for bounded creeping flow of drops, Eng. Anal. Boundary Elements 30 (2006) 856–868.

[11] A.S. Sangani, G. Mo, An O(N) algorithm for Stokes and Laplace interactions of particles, Phys. Fluids 8 (1996) 1990–2010.

[12] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, J. Comput. Phys. 196 (2004) 591–626.

[13] H. Wang, T. Lei, J. Li, J. Huang, Z. Yao, A parallel fast multipole accelerated integral equation scheme for 3D Stokes equations, Int. J. Numer. Meth. Eng. 70 (2007) 812–839.

[14] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, Acta Numer. 229 (1997).

[15] G.J. Rodin, J.R. Overfelt, Periodic conduction problems: the fast multipole method and convergence of integral equations and lattice sums, Proc. R. Soc. Lond. A 460 (2004) 2883–2902.

[16] A.Z. Zinchenko, Efficient numerical solution of the electrostatic problem for a randomly periodic fibrous composite dielectric, Sov. Phys. Tech. Phys. 34 (1989) 15.

[17] A.Z. Zinchenko, Effective dielectric constant of concentrated fiber composites studied by direct numerical simulation, Sov. Phys. Tech. Phys. 35 (1990) 1236.

[18] A.Z. Zinchenko, An efficient algorithm for calculating multiparticle thermal interaction in a concentrated dispersion of spheres, J. Comput. Phys. 111 (1994) 120.

[19] A.Z. Zinchenko, Effective conductivity of loaded granular materials by numerical simulation, Philos. Trans. R. Soc. Lond. A 356 (1998) 2953.

[20] A.Z. Zinchenko, R.H. Davis, An efficient algorithm for hydrodynamical interactions of many deformable drops, J. Comput. Phys. 157 (2000) 539–587.

[21] A.Z. Zinchenko, R.H. Davis, Shear flow of highly concentrated emulsions of deformable drops by numerical simulations, J. Fluid Mech. 455 (2002) 21–62.

[22] A.Z. Zinchenko, R.H. Davis, Large-scale simulations of concentrated emulsion flows, Phil. Trans. R. Soc. Lond. A 361 (2003) 813–845.

[23] A.Z. Zinchenko, R.H. Davis, A multipole-accelerated algorithm for close interaction of slightly deformable drops, J. Comput. Phys. 207 (2005) 695–735.

[24] V. Cristini, J. Blawzdziewicz, M. Loewenberg, An adaptive mesh algorithm for evolving surfaces: simulations of drop breakup and coalescence, J. Comput. Phys. 168 (2001) 445–463.

[25] F.R. Cunha, M. Loewenberg, A study of emulsion expansion by a boundary integral method, Mech. Res. Comm. 30 (2003) 639–649.

[26] O.A. Ladyzhenskaya, The Mathematical Theory of Viscous Incompressible Flow, 2nd ed., Gordon and Breach, 1969.

[27] H. Hasimoto, On the periodic fundamental solutions of the Stokes equations and their application to viscous flow past a cubic array of spheres, J. Fluid Mech. 5 (1959) 317.

[28] J.M. Rallison, A. Acrivos, A numerical study of the deformation and burst of a viscous drop in an extensional flow, J. Fluid Mech. 89 (1978) 191–200.

[29] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press, Cambridge, 1992.

[30] A.Z. Zinchenko, M.A. Rother, R.H. Davis, A novel-boundary integral algorithm for viscous interaction of deformable drops, Phys. Fluids 9 (1997) 1493–1511.

[31] H. Power, G. Miranda, Second kind integral equation formulation of Stokes flow past a particle of arbitrary shape, SIAM J. Appl. Maths 47 (1987) 689–698.

[32] J.M. Rallison, A numerical study of the deformation and burst of a viscous drop in general shear flows, J. Fluid Mech. 109 (1981) 465–482.

[33] G.A. Korn, T.M. Korn, Mathematical Handbook for Scientists and Engineers, McGraw-Hill, New York, 1968.

[34] E.W. Hobson, The Theory of Spherical and Ellipsoidal Harmonics, Chelsea, New York, 1955.

[35] J. Hapel, H. Brenner, Low Reynolds Number Hydrodynamics, Nijhoff, Dordrecht, 1973.

[36] A.F. Nikiforov, V.B. Uvarov, The Foundations of Special Function Theory, Nauka, Moscow, 1974, (in Russian).

[37] L.C. Biedenharn, J.D. Louck, Angular Momentum in Quantum Physics: Theory and Application, Addison-Wesley, Reading, MA, 1981.

[38] D.R. McKenzie, R.C. McPhedran, G.H. Derrick, The conductivity of lattices of spheres. II. The body-centred and face-centred cubic lattices, Proc. R. Soc. Lond. A 362 (1978) 211.

[39] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 1781.

[40] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, J.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.

[41] A.Z. Zinchenko, M.A. Rother, R.H. Davis, Cusping capture and breakup of interacting drops by a curvatureless boundary-integral algorithm, J. Fluid Mech. 391 (1999) 249–292.

[42] G.D. Scott, D.M. Kilgour, The density of random close packing of spheres, Br. J. Appl. Phys. Ser. 2 (1969) 863.

[43] N.A. Metropolis, A.W. Ronenbluth, M.N. Rosenbluth, A.N. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (1953) 1087.

[44] L.V. Woodcock, Glass transition in the hard-sphere model, J. Chem. Soc. Faraday Trans. II 72 (1976) 1667.

[45] S. Torquato, Random Heterogeneous Materials: Microstructure and Macroscopic Properties, Springer, New York, 2002.

[46] A.Z. Zinchenko, Algorithm for random close packing of spheres with periodic boundary condition, J. Comput. Phys. 114 (1994) 298–307.

[47] S.G. Yiantsios, R.H. Davis, On the buoyancy-driven motion of a drop towards a rigid surface or a deformable interface, J. Fluid Mech. 217 (1990) 547–573.

[48] S.G. Yiantsios, R.H. Davis, Close approach and deformation of two viscous drops due to gravity and van der Waals forces, J. Colloid Interface Sci. 144 (1991) 412–433.

[49] M. Nemer, Near-contact Motion of Liquid Drops in Emulsions and Foams, Ph.D. Thesis, Yale University, 2003.

[50] M.B. Nemer, X. Chen, D.H. Papadopoulos, J. Blawzdziewicz, M. Loewenberg, Hindered and enhanced coalescence of drops in Stokes flow, Phys. Rev. Lett. 92 (2004) 114501.

[51] G. Onoda, E.G. Liniger, Random loose packings of uniform spheres and the dilatancy onset, Phys. Rev. Lett. 64 (1990) 2727.

[52] M. Loewenberg, E.J. Hinch, Numerical simulation of a concentrated emulsion in shear flow, J. Fluid Mech. 321 (1996) 395–419.

[53] D.L. Koch, E.S.G. Shaqfeh, The instability of dispersion of sedimenting spheroids, J. Fluid Mech. 209 (1989) 521–542.

[54] T.W. Secomb, Mechanics of blood flow in the microcirculation, in: C.P. Ellington, T.J. Pedley (Eds.), Biological Fluid Dynamics, 1995, pp. 305–321.

[55] A.Z. Zinchenko, R.H. Davis, Squeezing of a periodic emulsion through a cubic lattice of spheres, Phys. Fluids 20 (2008) 040803.

[56] V. Cristini, G.S. Kassab, Computer modeling of red blood cell rheology in the microcirculation: a brief overview, Ann. Biomed. Eng. 33 (2005) 1724–1727.

[57] A. Anderson, X. Zheng, V. Cristini, Adaptive unstructured volume remeshing – I: The method, J. Comput. Phys. 208 (2005) 616–625.

[58] A. Anderson, X. Zheng, V. Cristini, Adaptive unstructured volume remeshing – II. Application to two- and three-dimensional level-set simulations of multiphase flow, J. Comput. Phys. 208 (2005) 626–650.