Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 208 (2005) 626–650

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Adaptive unstructured volume remeshing – II: Application to two- and three-dimensional level-set simulations of multiphase flow

Xiaoming Zheng [a,1], John Lowengrub [a,1], Anthony Anderson [b,2], Vittorio Cristini [c,*]

[a] *Department of Mathematics, University of California, Irvine, CA, USA*
[b] *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, USA*
[c] *Department of Biomedical Engineering, REC 204, University of California, Irvine, CA 92697-2715, USA*

Received 21 October 2004; received in revised form 28 February 2005; accepted 28 February 2005
Available online 22 April 2005

Abstract

In Part I [Adaptive unstructured volume remeshing – I: The method, J. Comput. Phys., in press], we presented an adaptive remeshing algorithm that automatically adjusts the size of the elements of meshes of unstructured triangles (2D) and unstructured tetrahedra (3D) with time and position in the computational domain in order to efficiently resolve the relevant physical scales. Here, we illustrate the performance of an implementation of the algorithm in finite-element/level-set simulations of deformable droplet and fluid–fluid interface interactions, breakup and coalescence in multiphase flows. The wide range of length scales characterizing the dynamics are accurately resolved as demonstrated by comparison to experiments and to theoretical and sharp-interface (boundary-integral) numerical results. The computational cost is found to be competitive even with respect to boundary-integral methods. For the first time using an interface-capturing (level-set) method we successfully simulate the inertia driven impact and rebound of a liquid droplet from a liquid interface and find agreement with recent experimental results.

[*] Corresponding author. Also at the Department of Mathematics. Formerly at the Department of Chemical Engineering and Materials Science and School of Mathematics, University of Minnesota, USA.
E-mail address: cristini@math.uci.edu (V. Cristini).

[1] Formerly at the School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA.

[2] Present address: Department of Applied Mathematics, Northwestern University, Evanston, IL 60208, USA.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.02.024

1. Introduction

In Part I [1], we presented a hybrid adaptive mesh algorithm using unstructured triangles in 2D and unstructured tetrahedra in 3D. The algorithm is a generalization to volume domains of the adaptive surface remeshing algorithm developed by Cristini et al. [11] in the context of deforming interfaces in two and three dimensions. Here, we present an implementation of the unstructured adaptive remeshing algorithm in finite-element/level-set simulations of droplet and fluid–fluid interface interactions and coalescence in multiphase flows. Finite element methods are best suited to handle complex domain geometries and unstructured computational meshes. We implemented an efficient MINI element Stokes solver [2] and a Navier–Stokes solver based on a new modified projection method, that uses a few iterations in the projection step to handle the nonlinear term and ensures divergence free flow. This method also ensures fast calculation of the fluid pressure by borrowing a scheme previously used only in finite-difference calculations [17]. To capture the evolving deformable fluid–fluid interfaces, we use a level-set function, that is, the signed distance to the interfaces. The level-set method automatically handles topological changes, such as coalescence and breakup. Another advantage is that it provides easy and accurate calculation of normal vectors to the interface, and thus accurate surface tension evaluation, which is crucial to the dynamics.

The level-set function also provides, in our implementation, the link between the finite-element/level-set method and the adaptive remeshing. The computational mesh is adapted, at every time step of simulation, based on a local length scale that is linearly proportional to the magnitude of the level-set function. Thus, the regions around interfaces are most resolved. We find that this provides a means to control the numerical error during a simulation. The wide range of length scales characterizing deforming fluid/fluid interfaces and thinning fluid films between drops in near-contact motion are efficiently and accurately resolved as demonstrated by comparison to experiments, to exact solutions and to sharp-interface (boundary-integral) results. In particular, simulations of the inertia driven impact and rebound of a liquid droplet from a liquid interface are presented and compared to recent experiments for the first time, to our knowledge, using an interface-capturing (level-set) method.

The finite-element/level-set method implementation on the adaptive, unstructured mesh is described in Section 2. Two- and three-dimensional numerical simulations of droplet and fluid interface breakup and coalescence are presented in Section 3 and are compared to experiments, exact solution and boundary-integral numerical results. Conclusions and future work are in Section 4.

2. Level-set models of fluid/fluid interfaces

We present a finite-element discretization of the Stokes and Navier–Stokes equations for deformable fluid–fluid interfaces using the level-set method to capture the evolving interfaces. We discuss the coupling of these algorithms with the adaptive mesh refinement scheme presented in Part I [1] and address efficiency of the coupled algorithms.

2.1. The Stokes/level-set equations

We begin with the Stokes equations which govern zero-Reynolds-number fluid dynamics. We consider a two-phase flow, e.g., dispersed droplets of one fluid in another matrix (ambient) fluid. The continuum surface stress (distribution) form of the equations is [6,20]

$$-\nabla\cdot(\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}}))+\nabla p=\nabla\cdot(\sigma\delta_{\Sigma}(\mathbf{I}-\mathbf{nn}))+\mathbf{F}, \tag{1}$$

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

where μ is the viscosity, $\mathbf{u}$ and p are the fluid velocity and pressure fields, σ is the surface tension between the two phases, δ_Σ is the surface delta function, $\mathbf{n}$ is the normal to the interface Σ, $\mathbf{I}$ is the identity matrix, $(\mathbf{nn})_{ij} = n_i n_j$ and $\mathbf{F}$ is a body force (e.g., gravity, $\mathbf{F} = \rho\mathbf{g}$). Note that since we use the continuum surface stress formulation, the curvature of Σ does not explicitly appear in Eq. (1).

The interface Σ is defined as the zero-set of the level-set function ϕ, i.e. $\Sigma(t) = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\}$. The interface moves with the fluid

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \tag{3}$$

and the geometric quantities are defined as $\mathbf{n} = \nabla\phi/|\nabla\phi|$ and $\delta_\Sigma = \delta(\phi)|\nabla\phi|$ where δ is the one-dimensional delta function. The material properties are defined as

$$\mu = \mu_1 \mathscr{H}(\phi) + \mu_2(1 - \mathscr{H}(\phi)) \quad \text{and} \quad \rho = \rho_1 \mathscr{H}(\phi) + \rho_2(1 - \mathscr{H}(\phi)), \tag{4}$$

where $\mathscr{H}$ is the Heaviside function and μ_1, μ_2, ρ_1, ρ_2 are the viscosities and densities of the components respectively. The initial data $\phi(\mathbf{x}, 0)$ is the signed distance from the point $\mathbf{x}$ to the interface Σ [28,26,6], thus $|\phi(\mathbf{x}, 0)| = d_\Sigma(\mathbf{x}, 0)$. The level-set function is reinitialized at every time step (see Section 2.1.2) to be the signed distance function to the evolving interface.

The boundary conditions for Eqs. (1) and (2) on the computational domain Ω (typically a square domain in 2D and a sphere in 3D) are as follows. We partition the computational domain boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_R$. On $\partial\Omega_D$, we impose a Dirichlet condition,

$$\mathbf{u}|_{\partial\Omega_D} = \mathbf{U}, \tag{5}$$

to prescribe a fully developed external flow $\mathbf{U}$ (e.g., shear, or no-slip: $\mathbf{U} = 0$). On $\partial\Omega_R$, free-slip (Robin-type) boundary conditions are used [19]:

$$\mathbf{u} \cdot \mathbf{n}|_{\partial\Omega_R} = 0, \quad (\nabla \times \mathbf{u}) \times \mathbf{n}|_{\partial\Omega_R} = 0, \tag{6}$$

where $\mathbf{n}$ is the normal to $\partial\Omega_R$. For the 2D drop-interface impact simulations presented in this paper, $\partial\Omega_D$ (with $\mathbf{U} = 0$) includes the top and bottom segments of $\partial\Omega$ and $\partial\Omega_R$ the two sides. For all other simulations presented, $\partial\Omega_D = \partial\Omega$.

2.1.1. The MINI element Stokes solver

To solve Eqs. (1) and (2) numerically, we use a mixed finite-element method together with our unstructured, adaptive triangulated mesh in 2D and tetrahedral mesh in 3D. The equations are recast in weak form for the unknowns $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times \mathbf{Q}_h$:

$$\begin{aligned} &\int_\Omega \mu(\nabla\mathbf{u}_h + (\nabla\mathbf{u}_h)^{\mathrm{T}}) : \nabla\mathbf{v}_h \, \mathrm{d}x - \int_\Omega p_h \nabla \cdot \mathbf{v}_h \, \mathrm{d}x = \langle \mathbf{f}, \mathbf{v}_h \rangle, \\ &\int_\Omega q_h \nabla \cdot \mathbf{u}_h \, \mathrm{d}x = 0 \end{aligned} \tag{7}$$

for all test functions $(\mathbf{v}_h, q_h) \in \mathbf{V}_h \times \mathbf{Q}_h$ (see below), where the notation $L : M = L_{ij}M_{ji}$, and the forcing term includes the singular surface stress and the body force integrals

$$\langle \mathbf{f}, \mathbf{v}_h \rangle = -\int_\Omega \sigma(\mathbf{I} - \mathbf{nn}) : \nabla\mathbf{v}_h \delta_\Sigma \, \mathrm{d}x + \int_\Omega \mathbf{F} \cdot \mathbf{v}_h \, \mathrm{d}x. \tag{8}$$

For completeness, a derivation is presented in Appendix A (see also [19]). We consider a partition $\mathscr{T}_h$ of Ω into triangle or tetrahedral elements, on which we choose the MINI element discretization [2]:

$$\mathbf{V}_h = \{\mathbf{v} \in (H^1(\Omega))^d \mid \mathbf{v} \in (P^1(K))^d \oplus (B(K))^d \; \forall K \in \mathscr{T}_h, \mathbf{v}|_{\partial\Omega_D} = 0, \mathbf{v} \cdot \mathbf{n}|_{\partial\Omega_R} = 0\},$$

$$\mathbf{Q}_h = \left\{ q \in L^2(\Omega) \,|\, q \in P^1(K), \int_\Omega q = 0 \right\},$$

where $d = 2, 3$ is the dimension, $P^1(K)$ is the set of piecewise linear functions on the element K and $B(K)$ is the bubble function [2]. In 2D, $B(K)$ is a subspace generated by $\lambda_1\lambda_2\lambda_3$, and in 3D, $B(K)$ is generated by $\lambda_1\lambda_2\lambda_3\lambda_4$, where the basis functions λ_i are barycentric coordinates in the triangles and tetrahedra respectively. The MINI element discretization involves the least number of unknowns of all mixed methods that use triangular/tetrahedral elements and satisfy the Brezzi–Babuska inf–sup (stability) condition [2,18]. For the sake of brevity, we have presented above the discretization for drop-interface impact simulations on a square 2D computational domain. For all other simulations presented in this paper, analogous discretizations can be straightforwardly derived by imposing the appropriate boundary conditions for velocity.

The integrals in Eqs. (7) and (8) are discretized using a 7-point Gaussian quadrature rule in 2D which is exact for polynomials of degree 5, and a 5-point rule in 3D which is exact for polynomials of degree 3, given in Table A.4 on page 343 and Table A.5 on page 344 of [5], respectively, where the values of the integrands at the Gaussian points are obtained either through interpolation of nodal values or through exact evaluation where appropriate.

To evaluate the surface stress integral in Eq. (8), the surface delta function $\delta_\Sigma = \delta(\phi)|\nabla\phi|$ is replaced by a smoothed version $\delta_\varepsilon(\phi)|\nabla\phi|$ where $\delta_\varepsilon(\phi) = \mathrm{d}\mathscr{H}_\varepsilon/\mathrm{d}\phi$ and $\mathscr{H}_\varepsilon$ is the smoothed Heaviside function [38],

$$\mathscr{H}_\varepsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right)\right) & \text{if } |\phi| \leqslant \varepsilon, \\ 1 & \text{if } \phi > \varepsilon, \end{cases} \tag{9}$$

where the parameter ε is taken to be 2–4 times the smallest prescribed mesh size h_1. The Heaviside function in Eq. (4) is also replaced by the smoothed version above. We note that other alternatives to the smoothed delta and Heaviside functions with possibly better convergence properties have recently been analyzed by Tornberg and Engquist [40] and Calhoun and Smereka [4] on structured Cartesian meshes.

It is possible to evaluate the surface stress integral in Eq. (8) without regularization by reducing the volume integral to an integral along the interface prescribed by the zero level-set in a mesh element. While this was found to be quite accurate, regularization is necessary to maintain stability through a topology transitions such as pinchoff and reconnection.

To evaluate the normal vector $\mathbf{n}$ to the interface Σ, we use a quadratic fit of the level-set function through each node and its nearest neighbors on the triangular/tetrahedral mesh.

We use the Uzawa method [18] to solve the discretized Stokes equations (1)–(2). The most time consuming part of the solution process is the inversion of the velocity Laplacian matrices. We use SSOR preconditioned CG method (SSORCG) [21] when the dimension of the matrix is small; we use the Incomplete Cholesky factorization preconditioned CG method (ICCG) [21] otherwise (e.g., in 3D). For these preconditioned sparse symmetric systems, the condition number is $\mathrm{O}(h_1^{-1})$ where h_1 is a measure of the smallest edge length [3,44]. It follows that the CPU cost for the Uzawa algorithm in 2D is $\mathrm{O}(N^{1.25})$ and in 3D is $\mathrm{O}(N^{1.17})$, where N is the total number of nodes [23].

Remark 1. Although we do not present a formal proof, numerical evidence strongly suggests that the elements K in the 2D and 3D adaptive, unstructured meshes [1] are quasi-regular. That is, $H_K/h_K \leqslant c$ where $c > 0$ is a constant independent of the mesh and $H_K = \min_{\mathscr{B}}\{\mathrm{diam}(\mathscr{B}) \,|\, \mathscr{B}$ is a ball containing $K\}$ and $h_K = \max_{\mathscr{B}}\{\mathrm{diam}(\mathscr{B}) \,|\, \mathscr{B}$ is a ball contained in $K\}$. Quasi-regular meshes are important for controlling the condition number of the stiffness matrices thus allowing efficient and accurate finite element solutions.

Remark 2. Although $\nabla \cdot \mathbf{u}$ should be zero analytically by the incompressible condition, this is not true numerically even in weak sense (see [19] for details). However, as the mesh size tends to zero, $\nabla \cdot \mathbf{u}$ converges to zero.

2.1.2. The discontinuous Galerkin method

The level-set advection equation (3) is reformulated as

$$\phi_t + \nabla \cdot (\mathbf{u}\phi) = \phi \nabla \cdot \mathbf{u}, \tag{10}$$

and is solved using the Runge–Kutta Discontinuous Galerkin method (RKDG) [8,9]. Here, we use piecewise linear elements $P^1(K)$ and a 2nd-order Runge–Kutta time integration scheme. The algorithm is formally 2nd-order accurate both in time and space, with CFL number 1/3, i.e.

$$\Delta t = \frac{1}{3} \min_{\text{node } i \in \Sigma_N} \frac{h^i_{\min}}{|\mathbf{u}|^i_{\max}}, \tag{11}$$

where $h^i_{\min}$ is the minimum adaptive mesh size around node i, and $|\mathbf{u}|^i_{\max}$ is the maximum speed at nodes in a neighborhood Σ_N of node i.

We use the local level-set [30]/narrow-band approach [7] so that Eq. (10) is solved only in the neighborhood of the interface Σ. Define Σ_1 (the 1-tube) as the closure of all the elements that are crossed by the interface and all the nodes, edges (and also faces in 3D) belonging to the these elements. For $N \geqslant 2$, the N-tube Σ_N is defined as the closure of Σ_{N-1} and its neighboring elements, as well as all the nodes and edges (and also faces in 3D) belonging to all these elements. That is, Σ_N is a narrow layer about the interface with thickness approximately equal to N triangles/tetrahedra. Then, the level-set equation (10) is solved in the N-tube Σ_N with $N = 3$ or 4.

2.1.3. Re-initialization

After every time step of evolution, we re-initialize the level-set function following [36] to ensure that the level-set function ϕ still describes the signed distance from the interface Σ. Accordingly, we integrate the first-order Hamilton–Jacobi equation

$$\begin{aligned} &\frac{\partial d}{\partial \tau} + \operatorname{sgn}(\phi)(|\nabla d| - 1) = 0, \\ &d(0) = \phi. \end{aligned} \tag{12}$$

After a few iterations in the artificial time τ, $d \approx d_\Sigma$, the distance function ($|\nabla d_\Sigma| = 1$), and we reset $\phi = d$. In a numerical implementation, we only need to re-initialize the level-set function in the narrow band Σ_N with $N = 3$ or 4. We integrate (12) using a first-order in space and second-order in time Explicit Positive Coefficient (EPC) method [39].

In order to overcome the numerical difficulties associated to the re-initialization of the level-set function: loss of mass conservation and artificial displacement of the interface, we treat the set Σ_1 of all triangles crossed by the interface (where ϕ changes sign) in a separate fashion. The level-set function in Σ_1 is reset to be

$$\phi = d_\Sigma + \Delta d,$$

where d_Σ is directly calculated in Σ_1, and the uniform displacement Δd is chosen to constrain the mass $\int_{\Sigma_1} H(d_\Sigma + \Delta d) = \int_{\Sigma_1} H(\phi)$, the mass before re-initialization. Note that since the mass is a monotonic function of the displacement Δd of the interface, it follows that Δd is unique. We use the secant method to find Δd. In our simulations, we found that $\Delta d \ll h_1$, the mesh size in Σ_1, thus ensuring that artificial displacements of the interface are negligible. The re-set level-set values in Σ_1 are taken as the boundary condition to Eq. (12) which is then solved in the remaining region $\Sigma_N \backslash \Sigma_1$.

2.1.4. Interpolation on the adapted mesh

Since the computational mesh is adapted in time, it is necessary to interpolate all the quantities from the old mesh to the new mesh. For simplicity, we use a pointwise interpolation (Fig. 1). That is, for each evaluation point P_{new} which is a node on the new mesh, we first locate the element K_{old}, in the old mesh (a triangle in the skematic of Fig. 1), such that $P_{\text{new}} \in K_{\text{old}}$. Then the value of a quantity w at P_{new} is obtained from the local representation for w on K_{old}. In the case of the level-set function, an additional step is performed since the function is discontinuous. The level-set function is first projected (in the L_2 sense) into the space of continuous functions (also piecewise linear) before interpolation is performed. To efficiently locate K_{old}, the following strategy is used. For the first node $P_{1,\text{new}}$, a one-dimensional search is performed to find the corresponding element $K_{1,\text{old}}$ where the search direction is obtained via a steepest descent type algorithm (always proceed in the direction of the most negative barycentric coordinate of $P_{1,\text{new}}$ with respect to the old elements). The next interpolation point $P_{2,\text{new}}$ is a neighboring node to $P_{1,\text{new}}$. The search for the corresponding $K_{2,\text{old}}$ starts from $K_{1,\text{old}}$ and also follows the steepest descent algorithm. This procedure is repeated until all the new nodes are located on the old mesh elements. The computational cost of the search algorithm scales linearly with N, the total number of nodes on the new mesh. Typically, only 2 or 3 search steps are needed for each new node.

2.2. Extension to the Navier–Stokes/level-set equations

The continuum surface stress (distribution) form of the Navier–Stokes equations for finite Reynolds number is [6,20]

$$\rho(\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u}) = -\nabla p + \nabla\cdot\left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}})\right) + \nabla\cdot(\sigma\delta_\Sigma(\mathbf{I} - \mathbf{nn})) + \mathbf{F}, \tag{13}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{14}$$

Herein, we consider the Boussinesq approximation for simplicity. That is, on the left-hand side of Eq. (13) we take $\rho = \rho_{\text{ambient}}$ while on the right-hand side, we take $\mathbf{F} = \Delta\rho\mathbf{g}$, where $\Delta\rho = \rho - \rho_{\text{ambient}}$ and ρ is obtained from Eq. (4). The boundary conditions are the same as used in the Stokes system.

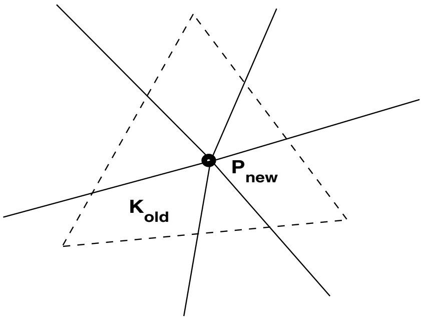


Fig. 1. Skematic (2D) illustrating the mapping relation between old and new meshes during interpolation.

2.2.1. A finite element projection method

To solve Eqs. (13) and (14), we use a finite-element projection method. The projection method, described below, requires significantly less computational cost to solve the Navier–Stokes equations compared to the Uzawa method used in the Stokes solver.

The projection scheme is based on an iterative solution of a Crank–Nicolson time discretization of Eqs. (13) and (14). Iteration is used to more accurately handle the nonlinear advection term and the incompressibility condition (14). For brevity, we present our scheme for the following simplified equations ($\rho_{\text{ambient}} = 1$):

$$\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}})] - \nabla p + \mathbf{f}(\phi), \tag{15}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{16}$$

where $\mathbf{f}(\phi)$ contains the surface tension and body forces. Let the superscript n denote time steps and k denote iterations within one time step. Then, the (spatially continuous) projection method is

$$\begin{aligned}\frac{\mathbf{u}^{*,k+1} - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^{n+1/2,k}\cdot\nabla)\mathbf{u}^{n+1/2,k} = {} & -\nabla p^{n,k} + \mathbf{f}(\phi^{n+1/2,k}) + \nabla\cdot\Big(\mu^{n+1/2,k}(\nabla\mathbf{u}^{*,k+1} + (\nabla\mathbf{u}^{*,k+1})^{\mathrm{T}})\Big)\Big/2 \\ & + \nabla\cdot\Big(\mu^{n+1/2,k}(\nabla\mathbf{u}^{n} + (\nabla\mathbf{u}^{n})^{\mathrm{T}})\Big)\Big/2,\end{aligned} \tag{17}$$

where $\mathbf{u}^{*,k+1}$ is an intermediate velocity field and $\mathbf{u}^{n+1/2,k} = (\mathbf{u}^{n+1,k}+\mathbf{u}^n)/2$ and $\mathbf{u}^{n+1,0} = \mathbf{u}^n$ (the initial guess for all variables $\mathbf{u}$, p, ϕ, ρ and μ is the value from the previous time step). Further, the viscosity $\mu^{n+1/2,k} = \mu(\phi^{n+1/2,k})$ and is obtained from Eq. (4) and

$$\phi^{n+1/2,k} = \frac{\phi^n + \phi^{n+1,k}}{2},$$

where

$$\phi^{n+1,k} = \phi^n - \Delta t\Big(\nabla\cdot\Big(\mathbf{u}^{n+1/2,k-1}\phi^{n+1/2,k-1}\Big) + \phi^{n+1/2,k-1}\nabla\cdot\mathbf{u}^{n+1/2,k-1}\Big), \tag{18}$$

which is the midpoint time integration rule applied to Eq. (10). The forcing term $\mathbf{f}(\phi^{n+1/2,k})$ is obtained analogously. The updated velocity iterate is obtained by solving

$$\mathbf{u}^{*,k+1} = \mathbf{u}^{n+1,k+1} + \Delta t\nabla q^{n+1,k+1}, \tag{19}$$

$$\nabla\cdot\mathbf{u}^{n+1,k+1} = 0, \tag{20}$$

which also yields the pressure increment $q^{n+1,k+1}$ (Neumann boundary conditions are applied for q). The pressure is then updated by

$$p^{n+1/2,k+1} = p^{n+1/2,k} + q^{n+1,k+1} - \Delta t\mu^{n+1/2,k}\nabla^2 q^{n+1,k+1}. \tag{21}$$

This form of the pressure update is inspired by the idea in [17] (Eq. 13) in that paper), in which an analogous update was used to obtain 2nd order convergence for pressure in a finite difference context with constant viscosity. In our case, however, there are interfaces across which the viscosity (and pressure) is discontinuous. As a consequence, we did not observe 2nd order convergence of pressure in the L_∞ norm (we do obtain convergence in the L_2 norm). Nevertheless, we find that this pressure update speeds up the convergence of the iteration method. Typically 4 or 5 iterations are enough to achieve convergence.

The initial values ($k = 0$) of the velocity and level-set iterates are given by $\mathbf{u}^{n+1/2,0} = \mathbf{u}^n$ and $\phi^{n+1/2,0} = \phi^n$ respectively. When $n = 0$, the initial guess for pressure p is zero. Thereafter, $p^{n,0}$ is set to be $p^{n-1/2}$, the pressure of last time step.

The spatial discretization of Eqs. (13) and (14) is performed by solving the weak form of the projection equations using vector-valued piecewise quadratic functions for velocity $P^2(K)$ and piecewise linear functions $P^1(K)$ for pressure (see [23] for details). This yields a more accurate solution of the equations than the MINI element method. Indeed, the overall scheme is 2nd order accurate for both the velocity and pressure in the L_2 norm. The weak formulation and treatment of boundary conditions analogous to that used in the Stokes system.

Finally, the discontinuous Galerkin method with piecewise linear functions $P^1(K)$ is used to discretize the level-set equation (18) in space.

Remark. Some authors [35] claimed that, since velocity and pressure are decoupled in the projection method, the inf–sup condition is not required and equal-order finite element spaces can be used. We find that while convergence is achieved in one-step projection method using equal-order finite element spaces, accuracy is reduced, especially for pressure, compared to a inf–sup-satisfying method. In particular, in the initialization step where pressure must be solved for, which is fundamentally a Stokes problem, the inf–sup condition is required for convergence.

2.3. Desired local length scale for adaptive mesh refinement

Numerical solution of the Stokes and Navier–Stokes level-set equations for interfacial flows require high resolution of the computational domain in a neighborhood of the interface. We thus seek a computational mesh of triangles (2D) and tetrahedra (3D) with a local element size prescribed by Eq. (7) in Part I [1], which resolves the mesh according to the distance to the interface. Therefore, we set:

$$L(\mathbf{x},t) = L_\Sigma = \min(h_0, h_1 + s|\phi(\mathbf{x},t)|) \tag{22}$$

since after reinitialization, the level-set function ϕ approximates a signed distance function to the interface Σ. In Eq. (22), h_1 is the imposed linear size of the computational elements in the sub-domain Σ_1 of elements that contain the interface Σ, and the slope s is set to smoothly increase the element size from h_1 to the uniform size h_0 away from Σ. This typically ensures accurate reinitialization and calculation of the normal vector.

To accurately account for the local curvature κ of the interface Σ (which is not needed by our finite element solvers) additional refinement would be necessary. Further, if multiple interfaces are in near contact, it is necessary to resolve the flow in the gap between the interfaces. In Stokes flows, these two length scales are sufficient to resolve accurately the flow-field. When inertia is present, high vorticity regions away from the interface may form and need to be resolved. In this case, the length scales associated to vorticity and/or the velocity gradients should also be directly resolved by the mesh. All these effects can be incorporated into our algorithm by taking the desired local length scale to be the minimum of the relevant local length scales

$$L(\mathbf{x},t) = \min(L_\Sigma, L_\kappa, L_{\text{gap}}, L_{\nabla\mathbf{u}}), \tag{23}$$

where $L_\kappa \sim 1/\kappa$, L_{gap} and $L_{\nabla\mathbf{u}} \sim |\mathbf{u}|/|\nabla\mathbf{u}|$ are the length scales associated with the interface curvature, gap-width between interfaces and velocity gradients respectively. The desired length scale L used by the adaption algorithm needs to be a smooth function of $\mathbf{x}$ and t [11]. For simplicity, in this paper we adapt the mesh only based on L_Σ. Note that using L_Σ alone we can in fact resolve all the local relevant scales by adjusting h_1 and s appropriately. Of course, this is not as efficient as implementing Eq. (23). The implementation of Eq. (23) is underway.

2.4. Efficiency of adaptive mesh refinement for interfacial flows

The most important advantage of adaptive mesh refinement is that the equations are solved using an optimal number of computational nodes. For example, suppose the local length scale to be resolved is

Table 1
Number of computational elements and CPU time for adaptive and non-adaptive finite-element and for boundary-integral simulations in 2D and 3D

	N(2D, 3D)	t_{CPU}(2D, 3D)
Non-adaptive FEM	$\mathrm{O}(h_1^{-2}), \mathrm{O}(h_1^{-3})$	$\mathrm{O}(h_1^{-7/2}), \mathrm{O}(h_1^{-9/2})$
Adaptive FEM	$\mathrm{O}(h_1^{-1}), \mathrm{O}(h_1^{-2})$	$\mathrm{O}(h_1^{-9/4}), \mathrm{O}(h_1^{-10/3})$
Boundary-integral	$\mathrm{O}(h_1^{-1}), \mathrm{O}(h_1^{-2})$	$\mathrm{O}(h_1^{-3}), \mathrm{O}(h_1^{-5})$

L_Σ. Then, the total number of computational nodes N required scales with the extent of interface in contrast to non-adaptive meshes where N scales with the extent of volume. In 2D, Σ is a line and in 3D a surface, thus, on the adaptive mesh $N = \mathrm{O}(h_1^{-1})$ in 2D and $N = \mathrm{O}(h_1^{-2})$ in 3D. In Table 1, the total number of computational elements and the CPU time scalings are summarized for 2D and 3D meshes using adaptive and non-adaptive finite elements (the entire volume is discretized) and comparing to those using boundary integrals (only the interface Σ is discretized).

Note that for the non-adaptive finite-element mesh the mesh size is uniform and equal to h_1. For the boundary-integral mesh, formula (22) does not apply and we have denoted with h_1 the linear size of the computational elements discretizing the interface. The CPU time associated to solving Eqs. (1) and (2) at each time step using the finite-element method described in Section 2.1 scales with the total number of computational elements with an exponent 5/4 in 2D and 7/6 in 3D; for the boundary-integral method, typically the solver scales $\mathrm{O}(N^2)$ (using direct summation). An implementation of the boundary-integral method using a fast summation technique would scale like $\mathrm{O}(N\log(N))$. The CFL-like constraint (11) implies that the CPU time associated to time stepping scales like h_1^{-1}. The total CPU times for simulations summarized in the above table follow from these considerations. These scalings demonstrate that the adaptive finite-element method is the most computationally efficient for a given resolution of the interface Σ.

Remark. Note that the $\mathrm{O}(N^{5/4})$ (2D) and $\mathrm{O}(N^{7/6})$ (3D) above scalings apply to the MINI element Stokes solver. The computational cost of the P2/P1 Navier–Stokes solver is likely to be higher. However, in the latter only few iterations are needed in the projection method, so the overall cost is acceptable.

3. Numerical results

3.1. Accuracy: steady drops in shear flow

To test our adaptive finite element/level-set numerical algorithm we simulate the Stokes flow (zero inertia) dynamics of 2D and 3D deformable drops in a steady shear flow. The dynamics depend on the drop-to-matrix viscosity ratio λ and the capillary number which measures the relative strength of the viscous stress and the surface tension stress: $Ca = \mu\dot{\gamma}a/\sigma$ where μ is matrix fluid viscosity, a is drop radius, σ is surface tension, and $\dot{\gamma}$ is the shear rate. For $Ca \leqslant Ca_c$, a critical value (that depends on the number of spatial dimensions), an initially undeformed drop deforms and rotates until it reaches a steady-state morphology. For $Ca > Ca_c$, the drop is unsteady and will deform continuously. Here, we compare our level-set/finite-element results to highly accurate boundary-integral simulations to examine the convergence of our numerical method. We focus on the subcritical case $Ca \leqslant Ca_c$.

The steady-state drops and corresponding adaptive meshes are shown in Fig. 2(a)–(c) together with the history of deformation (d) and a demonstration of numerical convergence (e). The initial drops are unit circle/sphere ($a = 1$), the viscosity ratio $\lambda = 1$, the capillary numbers $Ca = 0.5$ in 2D and $Ca = 0.3$ in 3D are sub-critical. In 2D, the equations are solved in a square domain. The drop-to-domain size ratio is

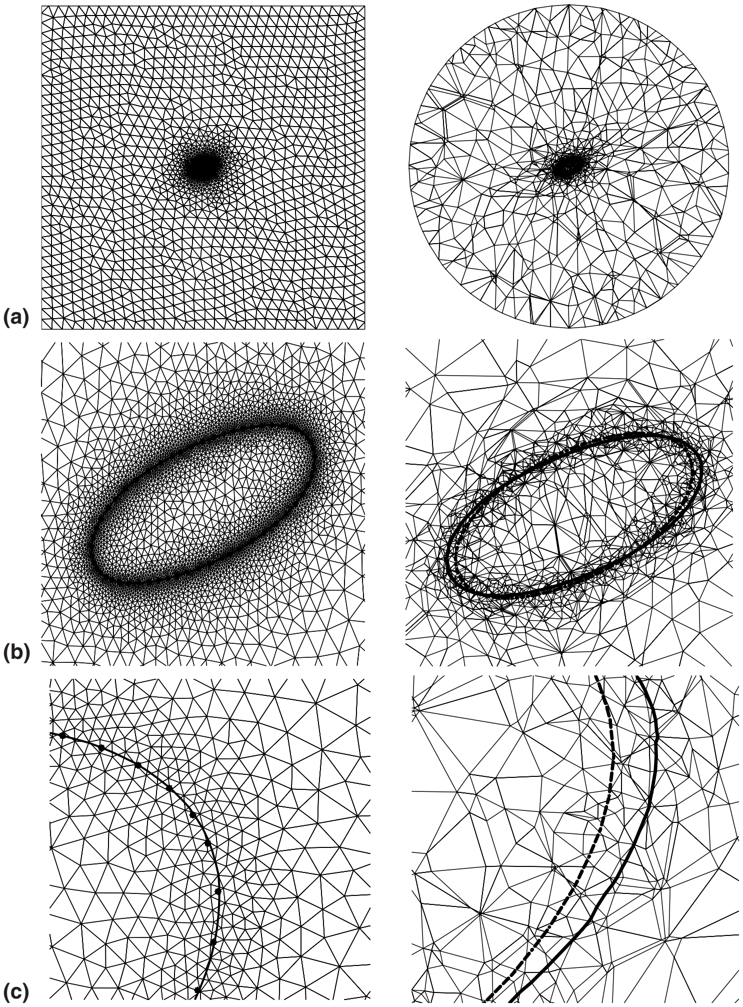


Fig. 2. Stationary drops in shear flow from adaptive level-set/finite-element simulations (viscosity ratio $\lambda = 1$). Left: 2D, square domain, $Ca = 0.5$, drop-to-domain size ratio 1/32, and remeshing parameters $h_0/a = 2$, $h_1/a = 0.04$, $s = 0.5$. Right: 3D, spherical domain, $Ca = 0.3$, size ratio 1/20, and $h_0/a = 3.5$, $h_1/a = 0.076$, $s = 0.4$. Full computational domains (a). Close-ups of the drops (b). Blow-up (c) by a factor h_0/h_1 demonstrating adaptivity and mesh quality invariance. Zero-level-set interface (solid curves). Boundary integral results in 2D (symbols) [42] and 3D (dashed) [11]. Drop lengths (d) as a function of strain $\dot{\gamma}t$ for 2D (left) and 3D (right) adaptive finite-element/level-set simulations compared to boundary-integral results (symbols). The discrepancies are within the prescribed error set by the mesh parameters h_1 and s. Convergence tests (e). Stationary drop length vs slope s for several values of h_1/a.

1/32 to minimize boundary effects. In 3D, a spherical mesh is used (drop-to-domain size ratio 1/20). At the boundary of the computational domain, the velocity is set to be the undisturbed linear shear flow. The mesh is adapted according to the local length scale in Eq. (22) with: $h_0/a = 2.0$, $h_1/a = 0.02$ and $s = 0.2$ in 2D; and with: $h_0/a = 3.5$, $h_1/a = 0.076$ and $s = 0.4$ in 3D. In Fig. 2(a)–(c), the plots on the left (2D) and on the right (3D) show the full computational domains (a) and close-ups of the steady drops (b,c). In 3D, a cross-section (drop mid-plane) of the adaptive mesh is shown together with the projection of the elements onto the plane of the page that crosses them. Also shown for comparison are the interface locations obtained using boundary-integral simulations in free-space using the algorithms from [42] in 2D (symbols) and from [11] in 3D

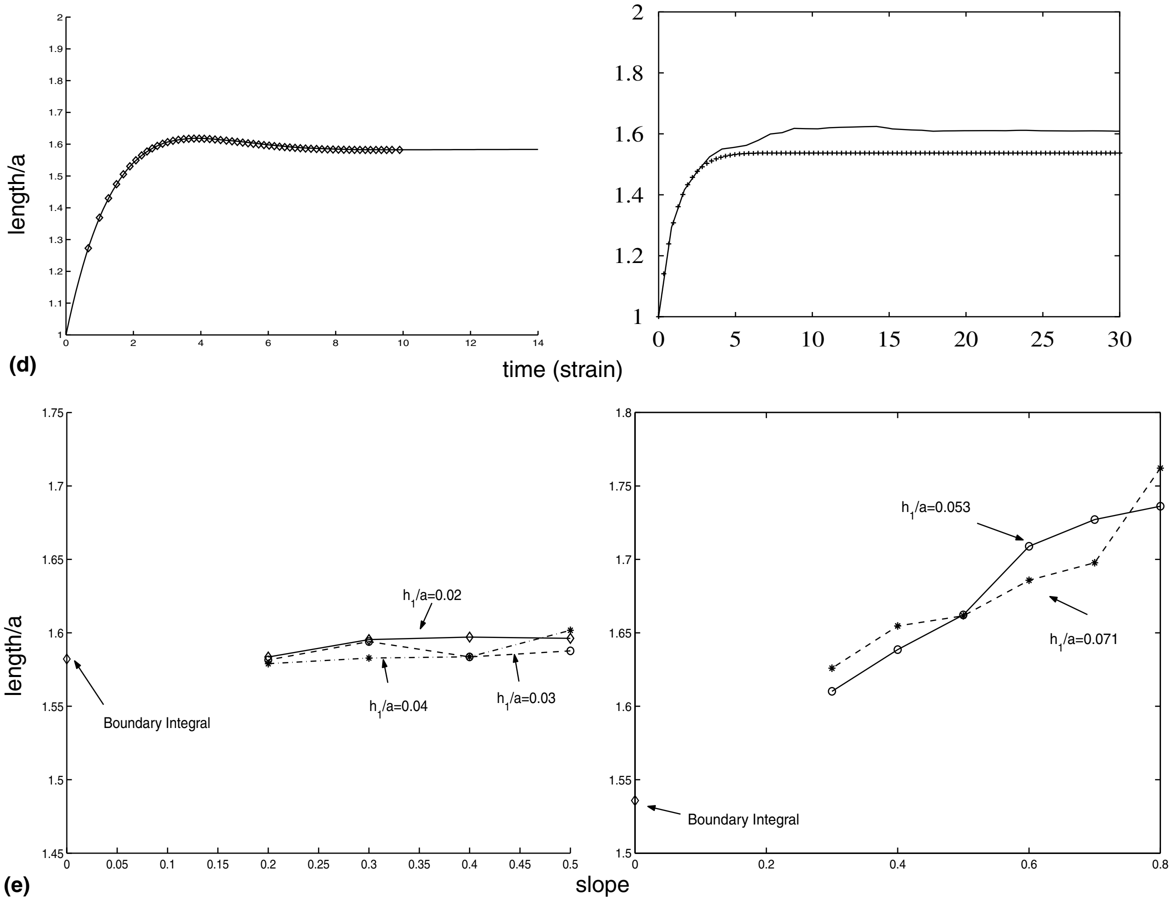


Fig. 2 (*continued*)

(dashed line). There is agreement between the level-set and boundary-integral simulations in 2D. The larger discrepancy in 3D is within the prescribed errors set by h_1 and s.

In Fig. 2(c), the computational mesh has been blown-up around the interface by the factor h_0/h_1. Observe that on this scale, the mesh elements around the interfaces are of approximately the same size and quality (i.e., nearly equilateral) as those elements at the domain boundaries in (a). This confirms that adaptivity and other desirable mesh properties are retained through dynamic spatial adaption [1]. At steady state there are $N = 5312$ and $N = 9869$ computational mesh nodes in 2D and 3D respectively. The 2D and 3D simulations take approximately 9 and 27 hours respectively to run on a Dell workstation (Xeon 2.4 GHz processor). For comparison, non-adaptive simulations with the same effective resolution would require $N = (32/0.02)^2 = 2.56 \times 10^6$ and $N = (20/0.076)^3 = 1.82 \times 10^7$ computational nodes in 2D and 3D respectively. According to the scalings discussed in Section 2.4 the non-adaptive simulations would require 2.3 (2D) and 37.3 (3D) years to complete.

In Fig. 2(d), the half-length of the drops is plotted as a function of time. The symbols denote the corresponding boundary integral results. There are a number of parameters that control the error. These include the domain size (in contrast the boundary-integral results correspond to an infinite domain), the minimum grid size h_1/a, the maximum grid size h_0/a and the slope s. Here, we present convergence with respect to h_1/a and s since these are found to dominate the error for the values of h_0 and domain size used. In Fig. 2(e), the stationary drop half-length is plotted as a function of slope s for several values of h_1 as labelled. The boundary-integral results are also reported. The effect of decreasing s is to increase the mesh density around the interface while leaving the smallest mesh size h_1 unchanged. Numerical convergence to the boundary-integral results is demonstrated. The slight oscillations of the convergence curves are likely a consequence of nonuniformity in the mesh refinement due to the tolerance values used during local mesh restructuring and dynamic node displacement [1].

3.2. Resolution of multiple scales: coalescence events in emulsions

Capturing the dynamics of deformable fluid–fluid interfaces in close hydrodynamic interaction is an extremely difficult challenge for any numerical method because of the critical role played by the very thin near contact lubrication zone between the interfaces. In this region, large pressures develop and resist coalescence sometimes preventing it altogether. A lack of numerical resolution results in inaccurate calculation of the lubrication pressure and in erroneous prediction of premature coalescence. To our knowledge, no continuum-based algorithm (e.g., level-set, volume-of-fluid, immersed-boundary, front-tracking) has been capable of accurately describing near-contact hydrodynamic interactions. Adaptive mesh refinement algorithms have been successfully used in 2D and 3D boundary-integral calculations [11].

3.2.1. Drop impact onto a fluid interface under gravity

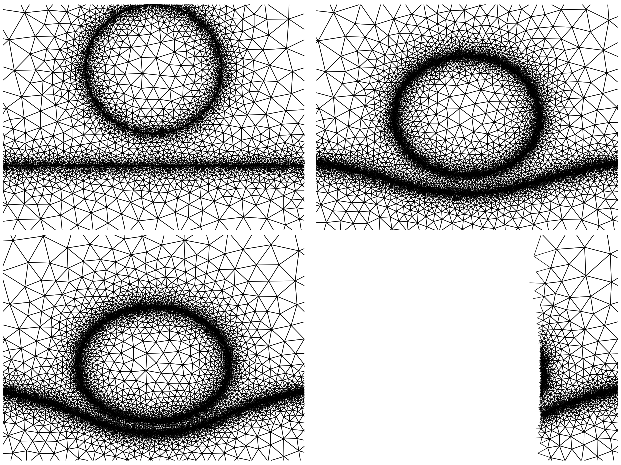
We present simulations of the impact of a drop onto a fluid interface under gravity. The fluid below the interface is the same as the drop fluid. Zero-Reynolds-number conditions are assumed and thus, the flow is characterized by the drop-to-matrix viscosity ratio λ and the Bond number which measures the relative strengths of the gravitational and surface tension forces: $Bo = \Delta\rho g a^2/\sigma$ where $\Delta\rho$ is the difference in densities of the drop and matrix fluids, and g is the acceleration of gravity.

Consider the lubrication flow in the thin gap between an approaching undeformed 2D drop and a planar fluid interface. At gap widths $w \ll a$, a singular pressure $p \sim F/(aw)^{1/2}$ per unit length arises in a near-contact region of extent $(aw)^{1/2}$ to resist the external force $F \sim \Delta\rho g a^2$ that pushes the drop onto the interface. Under the assumption of fully mobile undeformed interfaces [16] conservation of mass and momentum in the thin gap give respectively: $-\mathrm{d}w/\mathrm{d}t \cdot (aw)^{1/2} \sim uw$ and $pw \sim \mu u$, where u is the average velocity associated to the squeezing of fluid out of the gap between the approaching interfaces. By introducing the dimensionless variables $\bar{w} = w/a$, $\bar{t} = t/(\mu a/\sigma)$ (capillary relaxation time), we obtain

$$\frac{\mathrm{d}\bar{w}}{\mathrm{d}\bar{t}} \sim -Bo\bar{w}, \tag{24}$$

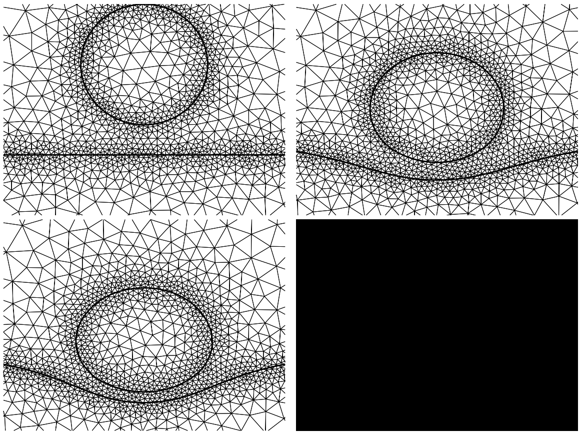
that is, the gap thins exponentially in time. Note that this 2D result differs from the finite coalescence time obtained for undeformed interfaces during 3D (axisymmetric) drainage [16]. In 2D, it is more difficult to squeeze fluid out of the gap thus, coalescence requires an infinite time.

Snapshots from an adaptive finite-element/level-set simulation are shown in Fig. 3(a) for parameters $\lambda = 1$ and $Bo = 0.8$. For the fluid parameters chosen, there is little deformation of the drop and interface. The mesh is refined according to Eq. (22) with parameters $h_0/a = 0.5$, $h_1/a = 0.0125$ and $s = 0.3$. During the dynamics, the mesh is continually refined/coarsened according to the motion of the interfaces such that the mesh tracks the interface throughout the evolution and the mesh size is small across the interfaces and smoothly increases away from the interfaces. Note that taking the desired mesh lengths to be related to the distance to the interface automatically ensures refinement of the near-contact region between the interfaces as far as the gapwidth $w > h_1$. In our simulations we found that as the minimum gap w between the interfaces decreases, numerical accuracy is lost when $w \approx 4h_1$ leading to erroneous prediction of coalescence



for $w < 4h_1$. It follows that the more refined the mesh, the smaller is the minimum gap that is accurately resolved.

The history of approach is reported in Fig. 3(c) corresponding to several numerical resolutions set by the mesh parameter h_1. For the simulation in Fig. 3(a) coalescence occurs around time $t = 55\mu a/\sigma$, and the minimum gap accurately resolved is $w/a \approx 0.05$. Taking a coarser mesh (see Fig. 3(b)), for example,



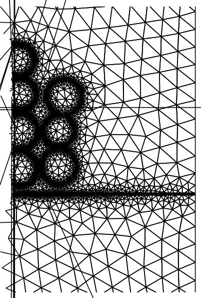
with $h_1/a = 0.05$ results in coalescence at the earlier time $t = 15\mu a/\sigma$ corresponding to a minimum accurate gap $w/a \approx 0.2$. The rapid drop-off in the curves in Fig. 3(c) indicates numerical coalescence that occurs at $w \approx 4h_1$ where the prescribed accuracy becomes inadequate to describe the flow in the narrow gap. The dashed straight-line marks exponential gap thinning according to Eq. (24). Our results show numerical convergence, as the mesh size $h_1 \rightarrow 0$, to the thinning rate and infinite coalescence time predicted by lubrication theory and demonstrate that mesh adaptivity allows the simulation error to be controlled [1].

3.2.2. Coalescence events in large systems of drops

We simulate 10 drops sedimenting onto a fluid interface under zero-Reynolds-number conditions and for viscosity ratio $\lambda = 1$. The simulation in Fig. 4(a) corresponds to a large Bond number $Bo = 80$. The drops have two sizes, corresponding to drop-to-domain size ratios either 0.1 or 0.09. The mesh is refined according to Eq. (22) with parameters $h_0/a = 1$, $h_1/a = 0.075$ (where a is the larger drop radius) and $s = 1$.

Because the surface tension force is weak compared to the gravitational force, the interfaces undergo large deformations during sedimentation. This leads to the complex morphologies visible in the figure. A cascade of drop/drop and drop/interface coalescence events results in the coalescence of all the dispersed phase with the interface. Note that the adaptive mesh follows the drop surfaces and interface through the complex dynamics and multiple coalescence events. As explained in the previous section, coalescence in finite time is a consequence of the loss of mesh resolution at gapwidths $w = \mathrm{O}(h_1)$. Further, the complex drop morphologies are consistent with recent experiments on drop/drop and drop/interface coalescence in dispersions at large Bo or low surface tension (Prof. E. Longmire, University of Minnesota, Personal communication), where similar "tree-like" structures of coalesced drops were observed. At later times $\overline{t} = t/(\mu a/\sigma) > 1$ (not shown), the coalesced interface retracts under capillarity and becomes flat.

In Fig. 4(b), a simulation of the same system under low Bond number conditions $Bo = 0.8$ is shown. When the Bond number is small, surface tension is important and thus drop deformation is smaller and the coalescence time scale is longer compared to that observed at high Bond number (a).



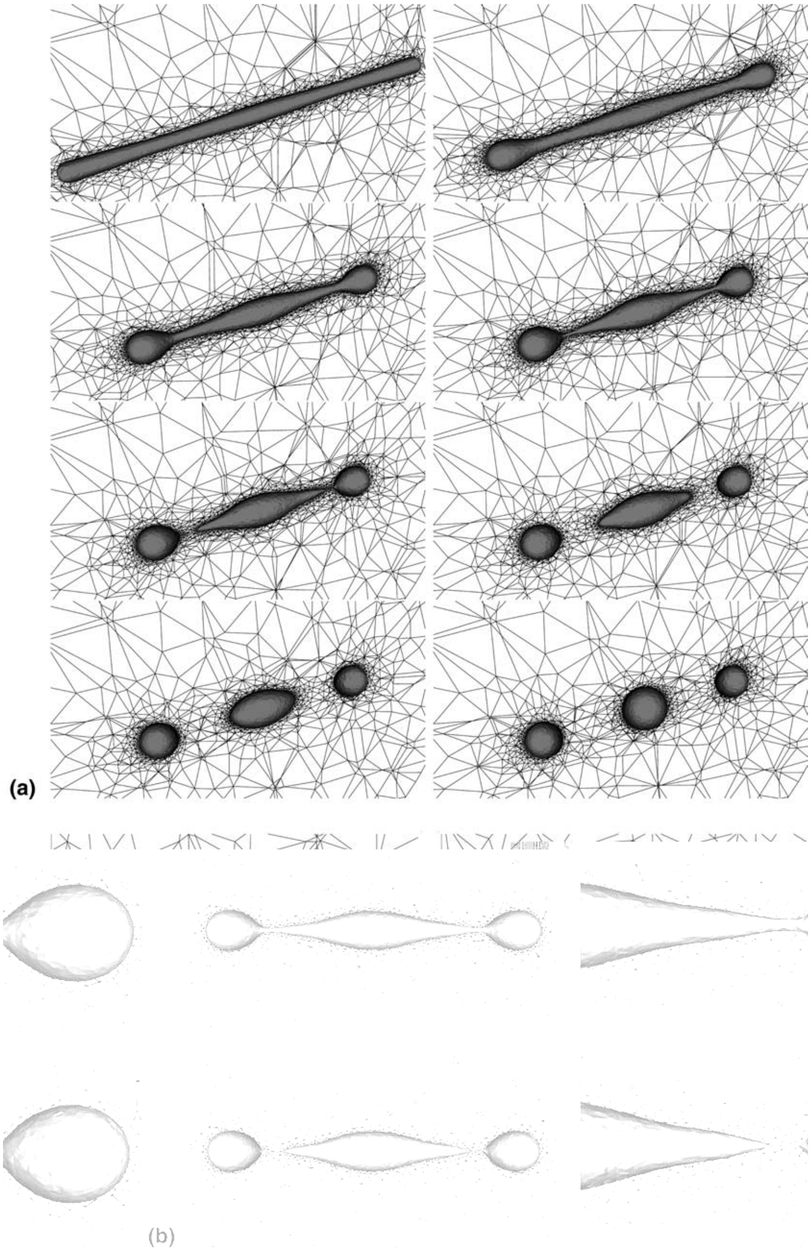


Fig. 5. Drop breakup under a Rayleigh instability in a quiescent matrix fluid. The drop surface and a slice of the 3D volume mesh (tetrahedral elements crossed by the slice) are visualized (a). The tetrahedra inside the drop are not visualized. Viscosity ratio $\lambda = 1$ and mesh parameters $h_0/a = 4.24$, $h_1/a = 0.0627$, $s = 0.4$. Times $\bar{t} = t/(\mu a/\sigma) = 0, 110.4, 188.9, 232.17, 240.5, 255.2, 271.9, 349.3$ (left to right, top to bottom). Blow-ups (b) of drop surface and computational mesh from (a) at times 232.2 (top) and 240.5 (bottom).

As the drop retracts, as shown in Fig. 5(a), bulbs form at the drop-tips and the drop necks down. At time $\bar{t} = t/(\mu a/\sigma) = 240.5$, the bulbs pinch off forming two satellite drops separated by an extended daughter drop. Surface tension then drives the newly produced drops to become spherical. As can be seen in the figure, and in Fig. 5(b), where a blow-up of the drop surface and volume mesh is shown, the mesh size is small near the interface and gradually increases away from the interface. Further, the mesh tracks the newly-created interfaces through the topology transition.

3.4. A 2D numerical investigation of the impact and rebound of a drop onto a fluid interface with inertia

We present simulations of a drop impacting onto a fluid interface at finite Reynolds number in 2D. The dynamics is governed by the nondimensional parameters: drop-to-matrix viscosity and density ratios and the Reynolds, Weber and Froude numbers. The Weber number $We = \rho_{\text{ambient}} U^2 a/\sigma$ measures the relative strength of the inertial and surface tension forces (U is the drop's terminal velocity). The Froude number $Fr = <\rho> U^2/(\Delta\rho g a)$, where $\langle\rho\rangle = (\rho_{\text{drop}} + \rho_{\text{ambient}})/2$ and $\Delta\rho = \rho_{\text{drop}} - \rho_{\text{ambient}}$, measures the relative strength of the inertial and gravitational forces. Note that the Bond number $Bo = We/Fr$.

In Fig. 6(a), the drop and interface are shown at a sequence of times during the evolution. The dimensionless parameters are matched with those in the experiment in [27]: $\lambda = 0.33$, $\rho_{\text{drop}}/\rho_{\text{ambient}} = 1.189$, $Re = 68$, $We = 7$ and $Fr = 1$. Initially, a circular drop with radius $a = 0.5$ is placed a distance $\bar{w} = w/a = 2.5$ away from a flat interface. The initial velocity is the divergence-free projection of the vector field $(0, -1)U$ inside the drop and equal to zero elsewhere in the domain. The computational domain is the square $[-D, D]^2$ with $D = 5$ so that the drop-to-domain size ratio $a/D = 0.1$. The adaptive mesh is refined according to Eq. (22) with mesh parameters: $s = 0.4$, $h_0/a = 2$; h_1 is decreased in time from $h_1/a = 0.02$ at time 0 to $h_1/a = 0.002$ at the final time shown. In the insets, snapshots from the experiment in [27] are also reproduced. Note that here we are comparing our 2D simulation with a fully 3D (nearly axisymmetric) experiment.

In contrast to viscosity-dominated low-Reynolds-number impact, the drop and interface deform significantly during the evolution. Due to the combination of inertial effects and surface tension, the drop shape oscillates as it falls. The initial impact between the drop and interface causes the drop to deform the interface downwards. The drop elongates in the vertical direction (time from impact: $\bar{t} - \bar{t}_i = 2.9$, where time is nondimensionalized as $\bar{t} = tU/a$). In fact, the upper surface of the drop falls below the original position of the flat interface, as also shown in Fig. 6(b). The drop then rebounds and both the drop and interface travel upwards before nearly coming to rest. The rebound is an inertial effect. Our 2D results are in qualitative agreement with the experiment although the numerically predicted drop deformations are somewhat larger. This may be due in part to the fact that in contrast to the experiment, in our simulation the initial separation between the drop and interface was not sufficiently large for the drop to reach a terminal shape. Preliminary axisymmetric calculations indicate that the Boussinesq approximation does not introduce any additional error.

In Fig. 6(b), the upper and lower drop surface positions are reported together with the interface position along the midline of approach as a function of time. The positions are measured relative to the initial position of the fluid interface and time is measured relative to the impact time, defined as the time at which the lower surface of the drop reaches the initial position of the fluid interface. The upper drop surface position decreases below the initial position of the flat interface and has two minima indicating that the upper surface of the drop has a small oscillation in time. The rebound of the drop and interface is seen by the fact that the positions of the lower drop surface and interface decrease to a minimum and increase thereafter. Observe that the distance between the lower drop surface and interface remains nearly constant in time after the minimum is achieved. The experiments demonstrated that after the rebound, the drop may rest on the interface for some time (even on the order of minutes) before coalescence occurs [27]. During the approach of the drop towards the interface (time from impact $\bar{t} - \bar{t}_i < 0$), the 2D terminal (impact) velocity is smaller than in 3D resulting in a delayed impact. Interestingly, after shifting the numerical results by a dimensionless time -1.1 to account for this delay, the results reported in the inset of Fig. 6(b) demonstrate

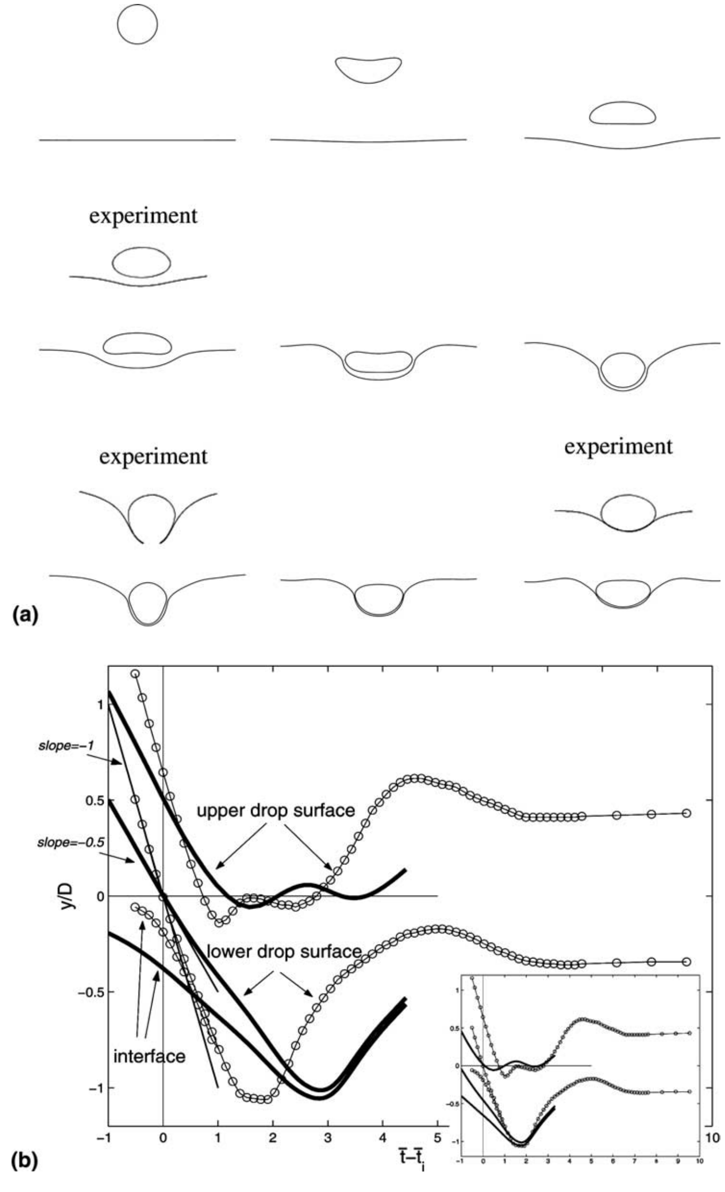


Fig. 6. Interface positions during impact of a drop onto an interface (a). Solid: 2D simulation; Insets: contours from experiments (3D) in [27]. Dimensionless parameters $Re = 68$, $We = 7$, $Fr = 1$, $\lambda = 0.33$, $\rho_{\text{drop}}/\rho_{\text{ambient}} = 1.189$. Mesh parameters: $h_0/a = 2$; h_1/a is continuously decreased during the simulation from 0.02 to 0.002. Left to right, top to bottom, times $\bar{t} - \bar{t}_i = (t - t_i)U/a = -4.7, -2.8, -0.8, 0, 1.2, 2.2, 2.9$ (1.8 in experiment), 3.7 and 4.3 (3.2 in experiment) measured from impact time. Time evolution of the upper and lower positions of the drop surface and of the position of the fluid interface (b). Solid: 2D simulation; symbols: experiment (3D). The 2D terminal (impact) velocity (labeled) is smaller than in 3D resulting in a delayed impact. Inset: the numerical results have been shifted by a dimensionless time −1.1 demonstrating that the interface near-contact dynamics in 2D and 3D are similar.

that the interface near-contact dynamics in 2D and 3D are similar. The numerical (2D) and experimental (3D) data reported in inset for the interfaces in near-contact (lower drop interface and fluid interface) virtually overlap during the interaction (experimental time from impact $1.3 < \bar{t} - \bar{t}_i < 2.3$). This result perhaps follows from the consideration that the drop-interface minimum separation occurs, in 3D, in a circular axisymmetric rim centered around the axis of approach and characterized by a circumference O(a), the undeformed drop radius, and by a rim thickness $\ll a$. Thus at this length scale the local near-contact geometry and flow are well approximated in our 2D simulations, where the minimum separation occurs in two infinitely long, symmetric sections of the drop and fluid interfaces of thickness $\ll a$ (e.g., see Fig. 7(c) bottom left, where the thickness of the minimum separation region is roughly the size of the box).

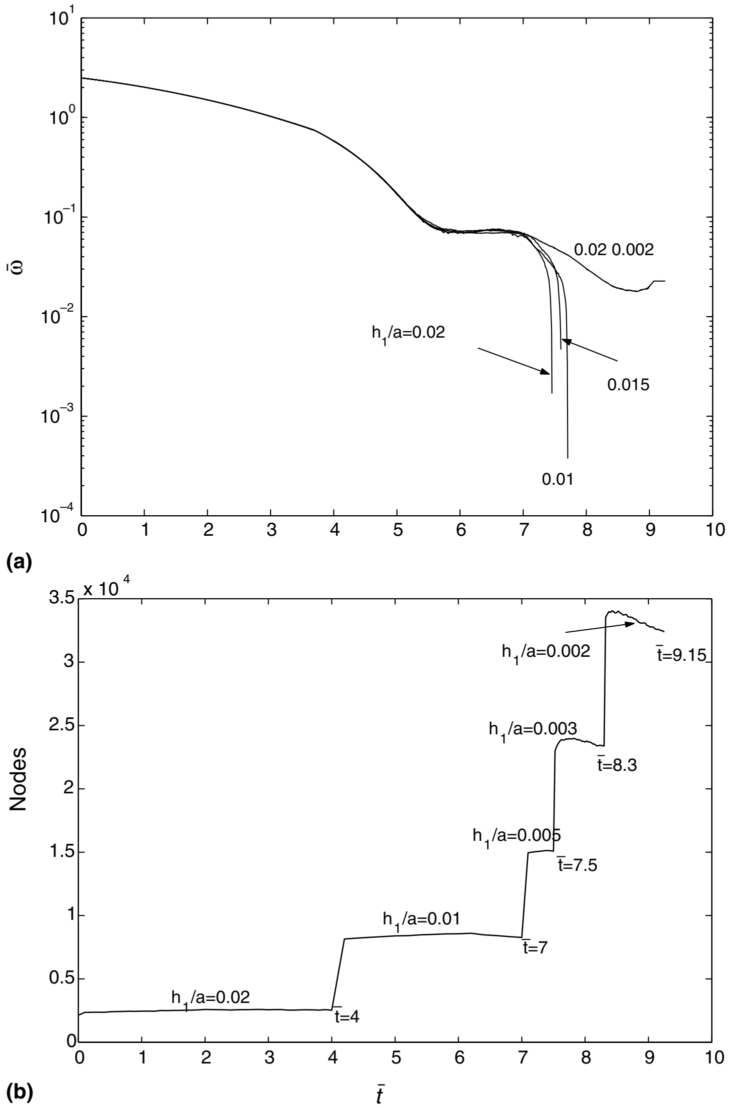


Fig. 7. Dimensionless gap width (minimum drop-interface separation) vs time (a) corresponding to different mesh resolutions set by the parameter h_1/a. Time evolution of the number of computational mesh nodes (b) for the simulation shown in Fig. 6 corresponding to $0.02 \geqslant h_1/a \geqslant 0.002$ and correctly predicting rebound and no coalescence. Simulations with lower resolution ($0.02 \geqslant h_1/a \geqslant 0.01$) erroneously predict coalescence. Computational mesh (c) from the simulation shown in Fig. 6 at time $\bar{t} = 9.15$. There are 32,601 nodes. Each of the first three figures has a boxed region that is magnified in the next figure (left to right, top to bottom). Note that the minimum drop-interface separation occurs in two symmetric locations centered around the plane of approach at a distance O(a) and of thickness $\ll a$ (bottom left).

As a test of refinement, in Fig. 7(a), we plot the minimum distance $\bar{w} = w/a$ between the drop and interface as a function of time for several resolutions corresponding to mesh parameters; h_1/a = 0.01, 0.015, 0.02 (each held fixed during the evolution) and for the case shown in Fig. 6 where h_1/a is varied from 0.02 to 0.002. The rapid drop off in the curves (a) indicates numerically induced coalescence due to insufficient resolution. In contrast, in the simulation where the parameter h_1/a is continuously refined, no coalescence is (correctly) predicted. Decreasing the parameter h_1/a in time during a simulation allows to resolve the lubrication flow in the near-contact region. Work is underway, following [11], to incorporate a gap-dependent length scale into the mesh density function to more efficiently resolve the near contact region.

Numerical accuracy depends critically on the resolution of the near-contact region. Here, rebound and no coalescence is correctly predicted from pure hydrodynamics. In Fig. 7(b), the total number of mesh

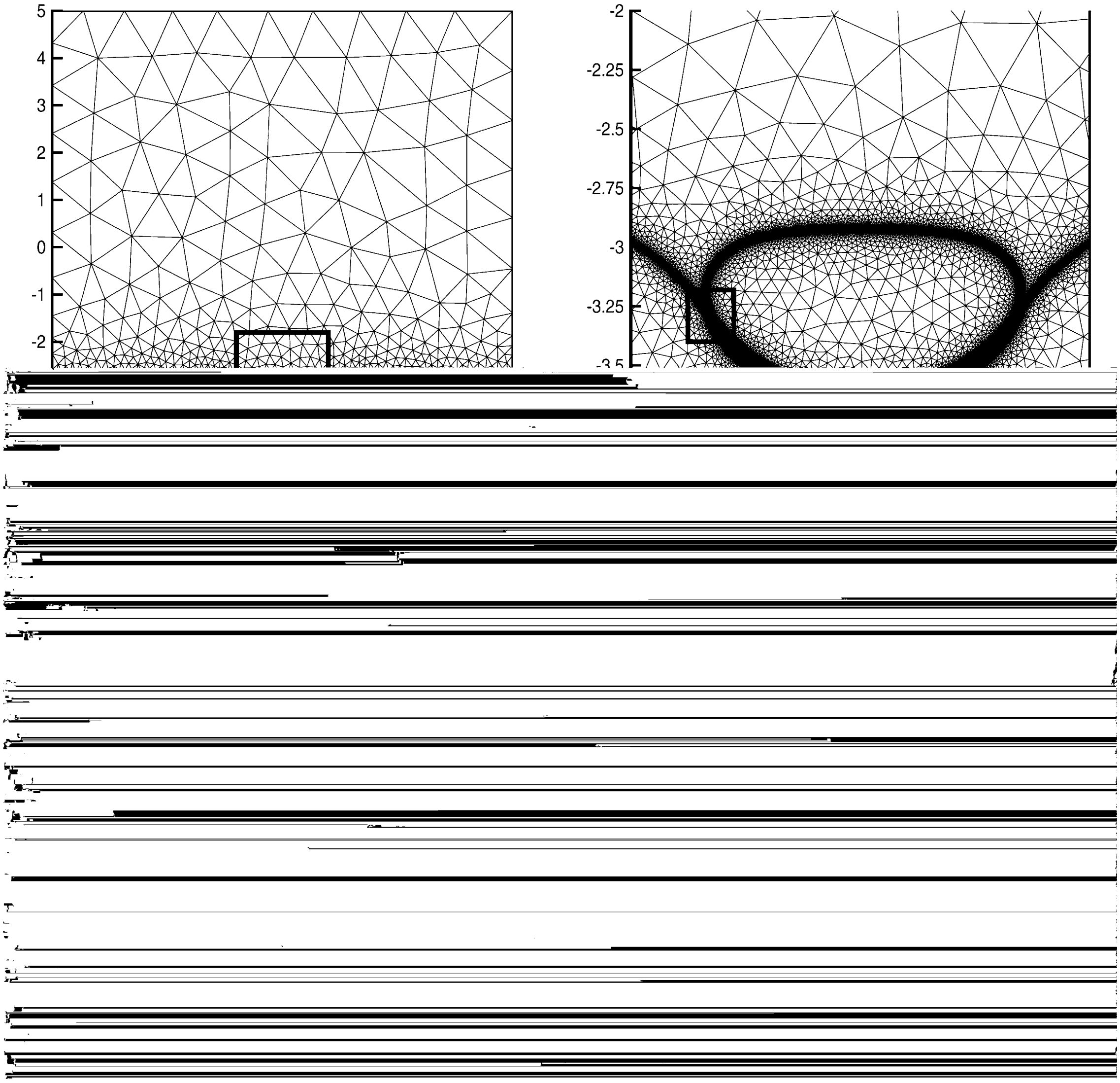


Fig. 7 (*continued*)

nodes is shown as a function of time throughout the simulation where h_1/a is continuously decreased from 0.02 to 0.002. The times at which h_1/a is decreased are indicated. The number of mesh points increases by a factor >10 throughout the simulation from $N \approx 2.5 \times 10^3$ at time 0 to $N \approx 3.26 \times 10^4$ at the final time.

In Fig. 7(c), the mesh is visualized at time $\bar{t} = 9.15$. The mesh is highly refined near the interface ($h_1/a = 0.002$). Each of the first 3 plots in (c) has a boxed region that is magnified in the next figure (top-to-bottom, left-to-right). The lower right plot corresponds to a magnification by a factor h_0/h_1. Note that the magnified triangles near the interface are approximately of the size and quality as the outer boundary triangles in the upper left figure confirming adaptivity according to Eq. (22) and that the quality of the mesh is maintained under refinement [1].

In Fig. 8, the streamlines are shown for the drop impact simulation from Fig. 6. Initially, two small vortices are located in the drop interior near the upper surface. As the drop continues to move downwards, the upper drop surface deforms and becomes concave. The vortices travel to the left and right. In addition, a strong wake region forms in the matrix fluid upstream from the drop. As the drop shape undergoes an oscillation, two additional vortices form near the drop's upper surface (time $\bar{t} - \bar{t}_i = -0.8$). Such additional vortices are present whenever the drop shape undergoes an oscillation (e.g., $\bar{t} - \bar{t}_i = 2.2$, 2.9 and 3.7) and the resulting flow inside the drop is quite complex. As the drop approaches the interface, the wake region intensifies and then dissipates once the drop begins to rebound. In addition, the vortices located at the drop-edges travel towards the lower drop tip and extend out of the drop across the near contact region and into the region below the interface (same fluid as the drop fluid). Finally, as the drop comes nearly to rest on the interface, the vortices near the upper drop surface move toward the drop center and dissipate. This evolution matches qualitatively that observed in the experiment [27]. Vortical structures off the interface are not

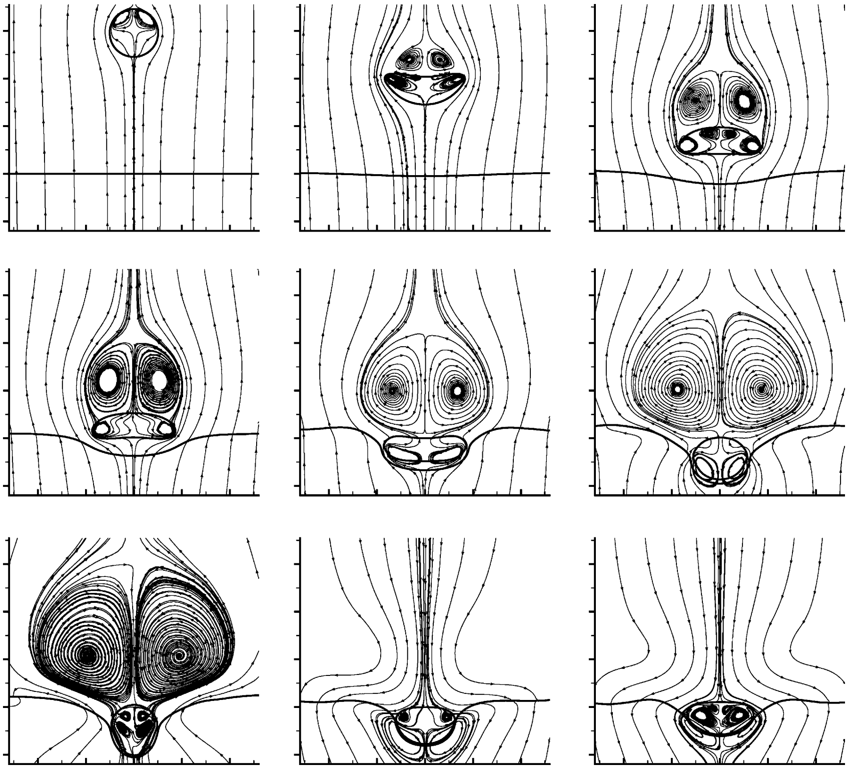


Fig. 8. Flow streamlines for the 2D simulation of drop-interface impact shown Fig. 6(a).

well resolved by the current choice of length scale L_Σ defined in Eq. (22) that ensures adequate resolution only near the interface Σ. Resolution of the wake region and other vortical features (particularly affecting inertial dynamics) can be achieved by using a local length scale associated with velocity gradients, e.g., $L_{\nabla\mathbf{u}} \sim |\mathbf{u}|/|\nabla\mathbf{u}|$, or $L_{\boldsymbol{\omega}} \sim |\mathbf{u}|/|\boldsymbol{\omega}|$ where $\omega = \nabla \times \mathbf{u}$ is the flow vorticity, in Eq. (23). This is currently under development.

4. Conclusions

We have illustrated the performance of an implementation of the adaptive remeshing algorithm developed in Part I [1], in finite-element/level-set simulations of deformable droplet and fluid–fluid interface interactions, breakup and coalescence in multiphase flows with and without inertia. We have compared our simulation results to experiments and to theoretical and sharp-interface (boundary-integral) numerical results, demonstrating that the wide range of length scales characterizing the dynamics are accurately resolved, while the computational cost is found to be competitive even with respect to boundary-integral methods. For the first time using an interface-capturing (level-set) method we successfully simulated the inertia driven impact and rebound of a liquid droplet from a liquid interface and found agreement with recent experimental results. This is made possible by the adaptivity of the computational mesh to the small scales characterizing the fluid motion and lubrication pressure associated to fluid interfaces in near contact.

In the simulations presented here for simplicity we have prescribed a length scale proportional to the distance from the interfaces, thus using the value of the level-set function to set the local size of the mesh elements. This approach is rigorously valid in Stokes flow simulations (zero inertia) where the flow features are completely determined by the instantaneous boundary. Our simulations with inertia have shown that additional flow features may develop in this case away from the interfaces, such as vortices in the wake past a sedimenting drop. We are developing a more complete length-scale function that accounts for this and other effects (e.g., concentration gradients when surfactants are present on the interfaces and in the bulk fluids). In particular, a parameter space investigation of the impact of a drop onto a fluid interface is underway. We have also applied our adaptive remeshing algorithm to multiphase flow simulations using a combined level-set/volume-of-fluid method [43].

Acknowledgments

V.C. and J.L. acknowledge funding from the National Science Foundation – Division of Mathematical Sciences. J.L. also acknowledges funding from the Department of Energy – Division of Basic Energy Sciences. Part of this work is contained in AA's undergraduate honors thesis (2004) at the University of Minnesota. A.A. acknowledges the Undergraduate Research Opportunity Program at U Minnesota for support. The authors also acknowledge the Minnesota Supercomputer Institute, and the Departments of Biomedical Engineering and Mathematics and the Network and Academic Computing Services, U C Irvine, for computing resources.

Appendix A. Derivation of the weak form formulation

Einstein summation convention is used. Also, $\partial_i = \partial/\partial x_i$. From (2), we have

$$-\partial_j(\mu(\partial_i u_j + \partial_j u_i)) + \partial_i p = f_i \quad i = 1, d, \tag{A.1}$$

where f_i, $i = 1, d$, is the force on the RHS of (2) and $d = 2, 3$ is number of spatial dimensions. Multiply (A.1) by a smooth function $\mathbf{v}$, integrate over Ω, using the divergence theorem, get

$$\int_\Omega \mu(\partial_i u_j + \partial_j u_i)\partial_j v_i - \int_\Omega p\partial_i v_i - \int_{\partial\Omega} \mu(\partial_i u_j + \partial_j u_i)n_j v_i + \int_{\partial\Omega} p v_i n_i = \langle f_i, v_i \rangle. \tag{A.2}$$

By using formula $(\mathbf{n}\cdot\nabla)\mathbf{u} = \mathbf{n}\cdot(\nabla\mathbf{u}) + (\nabla\times\mathbf{u})\times\mathbf{n}$, get

$$n_j\partial_j u_i v_i = n_j \partial_i u_j v_i + ((\nabla\times\mathbf{u})\times\mathbf{n})_i v_i. \tag{A.3}$$

Take $\mathbf{v} = 0$, $\mathbf{u} = 0$ on $\partial\Omega_D$, and $\mathbf{v}\cdot\mathbf{n} = 0$, $\mathbf{u}\cdot\mathbf{n} = 0$, and $(\nabla\times\mathbf{u})\times\mathbf{n} = \mathbf{0}$ on $\partial\Omega_R$, and $\partial\Omega_R$ flat and parallel to the Cartesian axes. It follows that $n_j\partial_i u_j v_i = 0$ on $\partial\Omega_R$ and the boundary integrals in (A.2) vanish. Thus we obtain the weak formulation:

$$\int_\Omega \mu(\partial_i u_j + \partial_j u_i)\partial_j v_i - \int_\Omega p\partial_i v_i = \langle f_i, v_i \rangle. \tag{A.4}$$

References

[1] A. Anderson, X. Zheng, V. Cristini, Adaptive unstructured volume remeshing – I: The method, J. Comput. Phys., in press, doi:10.1016/j.jcp.2005.02.023.

[2] D.N. Arnold, F. Brezzi, M. Fortin, A stable finite-element method for the stokes equations, Calcolo 21 (1984) 337.

[3] O. Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge, 1994.

[4] D. Calhoun, P. Smereka, The numerical approximation of a delta function, preprint, 2004.

[5] G.F. Carey, J.T. Oden, Finite Elements: Computational Aspects, III, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[6] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, J. Comput. Phys. 124 (1996) 449.

[7] D. Chopp, Computing minimal surfaces via level-set curvature flow, J. Comput. Phys. 101 (1993) 77.

[8] B. Cockburn, C.-W. Shu, The Runge–Kutta local projection discontinous Galerkin finite element method for conservation laws. iv: The multidimensional case, Math. Comput. 54 (1990) 545.

[9] B. Cockburn, C.-W. Shu, Runge–Kutta discontinous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173.

[10] V. Cristini, J. Blawzdziewicz, M. Loewenberg, Drop breakup in three-dimensional viscous flows, Phys. Fluids 10 (1998) 1781.

[11] 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.

[12] V. Cristini, J. Blawzdziewicz, M. Loewenberg, L. Collins, Breakup in stochastic stokes flows: sub-Kolmogorov drops in isotropic turbulence, J. Fluid Mech. 492 (2003) 231.

[13] V. Cristini, S. Guido, A. Alfani, J. Blawzdziewicz, M. Loewenberg, Drop breakup and fragment size distribution in shear flow, J. Rheol. 47 (2003) 1283.

[14] V. Cristini, C.W. Macosko, T. Jansseune, A note on transient stress calculation via numerical simulations, J. Non-Newtonian Fluid Mech. 105 (2002) 177.

[15] V. Cristini, Y.-C. Tan, Theory and numerical simulation of droplet dynamics in complex flows – a review, Lab Chip 4 (2004) 257.

[16] R.H. Davis, J.A. Schonberg, J.M. Rallison, The lubrication force between two viscous drops, Phys. Fluids A 1 (1989) 77.

[17] R. Cortez, D.L. Brown, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464.

[18] M. Fortin, F. Brezzi, Mixed and Hybrid Finite Elements, Springer-Verlag, New York, 1991.

[19] J.-L. Guermond, L. Quartapelle, On the approximation of the unsteady Navier–Stokes equations by finite element projection methods, Numer. Math. 80 (1998) 207.

[20] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423.

[21] W. Hackbusch, Iterative Solution of Large Sparse Systems of Equations, Springer-Verlag, New York, 1994.

[22] P.D. Anderson, I.B. Bazhlekov, H.E.H. Meijer, Nonsingular boundary integral method for deformable drops in viscous flows, Phys. Fluids 16 (2004) 1064.

[23] C. Johnson, Numerical Solution of Partial Differential Equations by the Finite Element Method, Cambridge University Press, Cambridge, 1987.

[24] D. Khismatullin, Y. Renardy, V. Cristini, Inertia-induced breakup of very viscous drops subjected to simple shear, Phys. Fluids 15 (2003) 1351.

[25] J. Li, Y.Y. Renardy, M. Renardy, Numerical simulation of breakup of a viscous drop in simple shear flow through a volume-of-fluid method, Phys. Fluids 12 (2000) 269.

[26] P. Smereka, M. Sussman, S. Osher, A levelset approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146.

[27] Z. Mohamed-Kassim, E.K. Longmire, Drop impact on a liquid–liquid interface, Phys. Fluids 15 (2003) 3263.

[28] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithm based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12.

[29] P.D. Patel, E.S.G. Shaqfeh, J.E. Butler, V. Cristini, J. Blawzdziewicz, M. Loewenberg, Drop breakup in the flow through fixed fiber beds: an experimental and computational investigation, Phys. Fluids 15 (2003) 1146.

[30] D.P. Peng, B. Merriman, H.-K. Zhao, S. Osher, A PDE-based fast local level-set method, J. Comput. Phys. 155 (1999) 410.

[31] Y. Renardy, V. Cristini, J. Li, Drop fragment distributions under shear with inertia, Int. J. Multiphase Flow 28 (2002) 1125.

[32] Y. Renardy, M. Renardy, Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys 183 (2002) 400.

[33] Y.Y. Renardy, V. Cristini, Effect of inertia on drop breakup under shear, Phys. Fluids 13 (2001) 7.

[34] Y.Y. Renardy, V. Cristini, Scalings for fragments produced from drop breakup in shear flow with inertia, Phys. Fluids 13 (2001) 2161.

[35] J. Shen, On error estimates of the projection methods for the Navier–Stokes equations: second-order schemes, Math. Comput. 65 (215) (1996) 1039–1065.

[36] M. Sussman, E. Fatemi, An efficient, interface preserving level set re-distancing algorithm and its application to interfacial incompressible fluid flow, SIAM J. Sci. Comput. 20 (1999) 1165.

[37] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, J. Comput. Phys. 30 (2000) 301.

[38] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146.

[39] J.A. Sethian, T.J. Barth, Numerical schemes for the Hamilton–Jacobi and level set equations on triangulated domains, J. Comput. Phys. 145 (1998) 1.

[40] A.-K. Tornberg, B. Engquist. Numerical approximations of singular source terms in differential equations. J. Comput. Phys., in press (online edition available).

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

[42] X.F. Li, C. Pozrikidis, Simple shear flow of suspensions of liquid drops, J. Fluid Mech. 320 (1996) 395.

[43] X. Yang, A.J. James, J. Lowengrub, X. Zheng, V. Cristini. An adaptive coupled level-set/volume-of-fluid interface tracking method for unstructured triangular grids, in review.

[44] X. Zheng. Adaptive algorithms for interface capturing methods, Ph.D. Thesis, University of California, Irvine (2005 expected).

[45] 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.