Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 219 (2006) 47–67
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Phase-field simulations of interfacial dynamics in viscoelastic fluids using finite elements with adaptive meshing

Pengtao Yue [a,b], Chunfeng Zhou [a], James J. Feng [a,b,*], Carl F. Ollivier-Gooch [c], Howard H. Hu [d]

[a] *Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z3*
[b] *Department of Mathematics, University of British Columbia, Vancouver, BC, Canada V6T 1Z2*
[c] *Department of Mechanical Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z4*
[d] *Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104, USA*

Received 30 October 2005; received in revised form 9 March 2006; accepted 16 March 2006
Available online 2 May 2006

Abstract

This paper describes a novel numerical algorithm for simulating interfacial dynamics of non-Newtonian fluids. The interface between two immiscible fluids is treated as a thin mixing layer across which physical properties vary steeply but continuously. The property and evolution of the interfacial layer is governed by a phase-field variable ϕ that obeys a Cahn–Hilliard type of convection-diffusion equation. This circumvents the task of directly tracking the interface, and produces the correct interfacial tension from the free energy stored in the mixing layer. Viscoelasticity and other types of constitutive equations can be incorporated easily into the variational phase-field framework. The greatest challenge of this approach is in resolving the sharp gradients at the interface. This is achieved by using an efficient adaptive meshing scheme governed by the phase-field variable. The finite-element scheme easily accommodates complex flow geometry and the adaptive meshing makes it possible to simulate large-scale two-phase systems of complex fluids. In two-dimensional and axisymmetric three-dimensional implementations, the numerical toolkit is applied here to drop deformation in shear and elongational flows, rise of drops and retraction of drops and torii. Some of these solutions serve as validation of the method and illustrate its key features, while others explore novel physics of viscoelasticity in the deformation and evolution of interfaces.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Non-Newtonian fluid mechanics; Two-phase flows; Drop deformation; Drop coalescence; Drop retraction; Interfacial tension; Diffuse-interface method; Moving boundary problem

1. Introduction

When two immiscible fluids are in contact, the interface often plays a central role in the fluid dynamics and rheology of the mixture system. The interface can be relatively well behaved and regular as in stratified

* Corresponding author.
E-mail address: jfeng@CHML.UBC.CA (J.J. Feng).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.03.016

two-phase flows or coextrusion of polymers, or highly irregular and transient as the internal phase boundaries in an emulsion, foam, or polymer blend. In general, the mathematical treatment of such moving boundary problems is complicated by the fact that the position of the interface is not known a priori. Rather it evolves according to the flow within both fluid components. In many industrial materials involving fluid mixtures, the components are often non-Newtonian complex fluids. The focus of this study is the interfacial dynamics in viscoelastic fluids.

A conceptually straightforward method for handling moving boundaries is to keep track of the motion of material points residing on the interface. Numerically, this may be realized by using a moving grid with grid points moving either according to the local fluid velocity or a mesh velocity [1,2]. This Lagrangian approach is often known as interface tracking [3]. Difficulties arise, however, when interfacial deformation becomes severe as to cause mesh entanglement. Typically a new mesh is laid down by remeshing and interpolation, with added cost and errors. An alternative to interface tracking is to compute the fluid flow of both components on a fixed Eulerian grid, with the interface being determined or reconstructed at each time step by using a scalar indicator function. Examples of this class of methods are the volume-of-fluid (VoF) method, the level-set method and the phase-field method.

The phase-field method stands out in its treatment of the interface as a physically diffuse thin layer. This differs from other fixed-grid methods where the interface is sharp conceptually but regularized numerically by spreading the interfacial force over a volume. Also known as the diffuse-interface method, the phase-field method dates back to van der Waals' great insight that the interface is in reality diffuse with a certain amount of mixing between any fluid pair. By introducing a phase-field parameter to distinguish the phases, one may construct a free energy – the mixing energy – that governs the dynamics of the diffuse interface. Interfacial tension, not surprisingly, can be identified with the mixing energy over the interfacial layer.

In recent years, several groups have applied the diffuse-interface method to two-phase flows [4–9]. In particular, Yue et al. [10] recognized the potential of this approach for treating microstructured fluids. The energy-based formalism of the theory makes it easy to incorporate complex rheology; the free energy of the microstructure can be directly added to the mixing energy to form the total free energy of the multi-phase system. Then a formal variational procedure will produce the proper constitutive equation for the fluid in addition to the evolution equation of the phase-field variable. Thus, the interfacial dynamics and non-Newtonian rheology are treated in a unified theoretical framework. Dissipative effects such as viscous stresses, of course, have to be accounted for separately, e.g., via the standard irreversible thermodynamic procedure [5].

Using a spectral discretization, Yue and coworkers have applied the phase-field method to simulate drop deformation, relaxation and collision in viscoelastic fluids and nematic liquid crystals [10–14]. These explorations have demonstrated the advantages of the model as well as its limitations, the most important of which is a stringent requirement on interfacial resolution [15]. In reality, the thickness of the mixing layer δ_{m} between nominally immiscible liquids is on the order of tens of nanometers. The great disparity between δ_{m} and the macroscopic flow scale D_{M} presents perhaps the most serious challenge for the diffuse-interface method. So far, this has been handled by using a thicker interface δ_n in the simulations and select the model parameters such that the proper interfacial tension is recovered [6,10]. Since the diffuse-interface model converges to the sharp-interface one in the limit of $\delta_n \to 0$, one should use as small a δ_n as is computationally affordable. Besides, δ_n also sets the lower limit on resolvable features of the system. For instance, a droplet smaller than δ_n essentially consists of interfaces only; its dimension and even its very existence are questionable.

Yue and coworkers found their spectral representation a handicap in this regard [15]. Its high accuracy derives from the requirements of a regular mesh, a simply shaped computational domain and spatial periodicity. Thus, to resolve δ_n at the interface entails a dense grid throughout the domain. This has led to large problem sizes even in two dimensions (2D). It has become clear that mesh adaptivity is essential for generalizing our algorithm to flow systems of larger scale, with complex and multiple interfaces and/or in three dimensions (3D). We also need the flexibility of an unstructured mesh to accommodate complex geometry for application to industrial flows.

This paper represents the first step in accomplishing these daunting goals. We have adopted a finite-element method which easily allows complex domains and unstructured mesh, and have developed an adaptive meshing scheme using the phase-field parameter as the criterion for dynamic refinement and coarsening of the grid. The objectives of this paper are: (a) to describe the new methodology, nicknamed AMPHI for *A*daptive *M*esh-

ClearCorrect Exhibit 1154, Page 2 of 21
ClearCorrect v. Align - IPR2025-00818

ing for Phase Field (ϕ); (b) to validate the theoretical and numerical models as well as the code by simulating benchmark problems; and (c) to use the unique capability of the simulation toolkit to explore how viscoelasticity impacts interfacial dynamics. The numerical results will be in 2D or axisymmetric 3D; extension to full 3D is underway and the results will be reported in a future communication.

2. Theoretical model

The diffuse-interface method as applied to two-phase flows has been described previously by a number of authors [4–6]. Yue et al. [10,12] have shown how the model can be extended to non-Newtonian fluids. In this section, we will briefly summarize the main ideas and list the governing equations, using the mixture of a Newtonian and an Oldroyd-B fluid as an example. The method accommodates other types of complex fluids such as nematic liquid crystals [12], but we will confine this paper to blends of a viscoelastic Oldroyd-B fluid with a Newtonian fluid.

The Newtonian and Oldroyd-B components are immiscible except in a very thin interfacial region, inside which the two components mix and store a mixing energy. An Oldroyd-B fluid is a dilute suspension of polymer chains, modeled as linear Hookean dumbbells, in a Newtonian solvent [16]. The total free energy of the mixture thus comprises two parts: mixing energy of the interface and elastic energy for the dumbbells. We introduce a phase-field variable ϕ such that the concentrations of the Oldroyd-B and Newtonian components are $(1+\phi)/2$ and $(1-\phi)/2$, respectively. For the mixing energy, we adopt the familiar Ginzburg–Landau form:

$$f_{\mathrm{mix}}(\phi,\nabla\phi) = \frac{1}{2}\lambda|\nabla\phi|^2 + \frac{\lambda}{4\epsilon^2}(\phi^2-1)^2, \tag{1}$$

where λ is the mixing energy density with the dimension of force, and ϵ is a capillary width that scales with the thickness of the diffuse interface. As $\epsilon \to 0$, the ratio λ/ϵ produces the interfacial tension σ in the classical sense [6,10]:

$$\sigma = \frac{2\sqrt{2}}{3}\frac{\lambda}{\epsilon}. \tag{2}$$

The evolution of ϕ is governed by the Cahn–Hilliard equation:

$$\frac{\partial\phi}{\partial t} + \boldsymbol{v}\cdot\nabla\phi = \gamma\nabla^2 G, \tag{3}$$

where $G = \frac{\delta\int f_{\mathrm{mix}}\,\mathrm{d}\Omega}{\delta\phi} = \lambda[-\nabla^2\phi + \frac{\phi(\phi^2-1)}{\epsilon^2}]$ is the chemical potential and γ is the mobility [10]. For an ensemble of dumbbells with an elastic constant H and a configuration distribution $\Psi(\boldsymbol{Q})$ about the connector vector $\boldsymbol{Q}$, the average energy is

$$f_{\mathrm{d}} = \int_{R^3}\left(kT\ln\Psi + \frac{1}{2}H\boldsymbol{Q}\cdot\boldsymbol{Q}\right)\Psi\,\mathrm{d}Q, \tag{4}$$

where k is the Boltzmann constant and T is the temperature, and the integration is over all possible configurations of $\boldsymbol{Q}$. Now the total free energy density of the two-phase system is:

$$f = f_{\mathrm{mix}} + \frac{1+\phi}{2}nf_{\mathrm{d}}, \tag{5}$$

where n is the number density of the dumbbells.

A variational procedure applied to the total free energy will yield the elastic stress tensor for the system. The stress tensor due to f_{mix} has been derived by Yue et al. [10], but in this paper we use the equivalent form of a body force:

$$\boldsymbol{B}_{\mathrm{mix}} = -\frac{\delta\int f_{\mathrm{mix}}\,\mathrm{d}\Omega}{\delta\boldsymbol{x}} = G\nabla\phi. \tag{6}$$

ClearCorrect Exhibit 1154, Page 3 of 21
ClearCorrect v. Align - IPR2025-00818

This form facilitates an energy law to be satisfied by the finite-element weak form; more details will be given in relation to Eq. (18). The elastic stress due to the dumbbell energy f_{d} turns out to be the Kramers expression for the polymer elastic stress tensor [11]:

$$\boldsymbol{\tau}_{\mathrm{d}} = -nkT\boldsymbol{I} + nH\langle \boldsymbol{QQ}\rangle, \tag{7}$$

where $\langle \boldsymbol{QQ}\rangle = \int_{R^3} \boldsymbol{QQ}\Psi \mathrm{d}Q$ and $\boldsymbol{I}$ is the identity tensor. $\boldsymbol{\tau}_{\mathrm{d}}$ obeys the Maxwell equation:

$$\boldsymbol{\tau}_{\mathrm{d}} + \lambda_{\mathrm{H}}\boldsymbol{\tau}_{\mathrm{d}(1)} = \mu_{\mathrm{p}}[\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^{\mathrm{T}}], \tag{8}$$

where the subscript $_{(1)}$ denotes the upper convected derivative, $\lambda_{\mathrm{H}} = \zeta/(4H)$ is the relaxation time, ζ being the friction coefficient between the dumbbell beads and the suspending solvent, and $\mu_{\mathrm{p}} = nkT\lambda_{\mathrm{H}}$ is the polymer viscosity. Adding the proper viscous stresses, we obtain the total stress tensor (excluding the part due to mixing energy):

$$\boldsymbol{\tau} = \left(\frac{1-\phi}{2}\mu_{\mathrm{n}} + \frac{1+\phi}{2}\mu_{\mathrm{s}}\right)[\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^{\mathrm{T}}] + \frac{1+\phi}{2}\boldsymbol{\tau}_{\mathrm{d}}, \tag{9}$$

where μ_{n} is the viscosity of Newtonian component and μ_{s} is the viscosity of the Newtonian solvent.

Thus the momentum equation can be written as:

$$\rho\left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{v}\right) = \nabla\cdot(-p\boldsymbol{I} + \boldsymbol{\tau}) + G\nabla\phi + \rho\boldsymbol{g}, \tag{10}$$

where ρ is the density of mixture: $\rho = \frac{1+\phi}{2}\rho_1 + \frac{1-\phi}{2}\rho_2$, ρ_1 and ρ_2 being the densities for the Oldroyd-B and Newtonian components, and $\boldsymbol{g}$ is the gravitational acceleration. Eqs. (3), (8), (9) and (10), along with the continuity equation $\nabla\cdot\boldsymbol{v} = 0$, form the governing equations for our two-phase system.

3. Numerical method

The current algorithm has two major ingredients: a finite-element flow solver and an adaptive meshing scheme. The former is based on a Navier–Stokes solver that Hu and coworkers [2,17] have used for simulating particle motion in Newtonian and viscoelastic fluids, while the latter is based on the mesh generator GRUMMP developed by Ollivier-Gooch and coworker [18]. Each ingredient has been generalized and adapted for the current purpose, and the two are then fused together.

3.1. Finite-element formulation

The discretization of the governing equations follows the standard Galerkin formalism [2]. However, the Cahn–Hilliard equation requires special attention. With C^0 elements, which are smooth within each element and continuous across their boundaries, one cannot represent spatial derivatives of higher order than 2. Thus the fourth-order Cahn–Hilliard equation has been decomposed into two second-order equations:

$$\frac{\partial\phi}{\partial t} + \boldsymbol{v}\cdot\nabla\phi = \frac{\gamma\lambda}{\epsilon^2}\Delta(\psi + s\phi), \tag{11}$$

$$\psi = -\epsilon^2\Delta\phi + (\phi^2 - 1 - s)\phi, \tag{12}$$

where s is a positive number that enhances the convergence of the iterative solution of the final linear system (see below). For all the calculations in this paper, $s = 0.5$ is used. The chemical potential G in Eq. (10) is now simply $G = \frac{\lambda}{\epsilon^2}(\psi + s\phi)$.

We seek the following weak solution: $(\boldsymbol{v}, p, \boldsymbol{\tau}_{\mathrm{d}}, \phi, \psi) \in \mathscr{U}\times\mathscr{P}\times\mathscr{T}\times\mathscr{S}\times\mathscr{S}$. For 2D flows, the solution spaces satisfy $\mathscr{U} \in H^1(\Omega)^2$, $\mathscr{P} \in L^2(\Omega)$, $\mathscr{T} \in L^2(\Omega)^3$ ($L^2(\Omega)^4$ for axisymmetric flow), and $\mathscr{S} \in H^1(\Omega)$. Using basis functions $(\tilde{\boldsymbol{v}}, \tilde{p}, \tilde{\tau}, \tilde{\phi}, \tilde{\psi})$, we write the following weak forms of the governing equations:

ClearCorrect Exhibit 1154, Page 4 of 21
ClearCorrect v. Align - IPR2025-00818

$$\int_\Omega \left\{ \left[\rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} - \boldsymbol{g} \right) - G \nabla \phi \right] \cdot \tilde{\boldsymbol{v}} + (-p\boldsymbol{I} + \boldsymbol{\tau}) : \nabla \tilde{\boldsymbol{v}} \right\} \mathrm{d}\Omega = 0, \tag{13}$$

$$\int_\Omega (\nabla \cdot \boldsymbol{v}) \tilde{p} \, \mathrm{d}\Omega = 0, \tag{14}$$

$$\int_\Omega \{ \boldsymbol{\tau}_\mathrm{d} + \lambda_\mathrm{H} \boldsymbol{\tau}_{\mathrm{d}(1)} - \mu_\mathrm{p} [\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^\mathrm{T}] \} : (\tilde{\boldsymbol{\tau}} + \alpha \boldsymbol{v} \cdot \nabla \tilde{\boldsymbol{\tau}}) \, \mathrm{d}\Omega = 0, \tag{15}$$

$$\int_\Omega \left[\left(\frac{\partial \phi}{\partial t} + \boldsymbol{v} \cdot \nabla \phi \right) \tilde{\phi} + \frac{\gamma \lambda}{\epsilon^2} \nabla (\psi + s\phi) \cdot \nabla \tilde{\phi} \right] \mathrm{d}\Omega = 0, \tag{16}$$

$$\int_\Omega \{ [\psi - (\phi^2 - 1 - s)\phi] \tilde{\psi} - \epsilon^2 \nabla \phi \cdot \nabla \tilde{\psi} \} \, \mathrm{d}\Omega = 0. \tag{17}$$

Note that we have adopted the streamline upwind Petrov–Galerkin scheme for the constitutive equations to improve stability [19]. The parameter $\alpha = \frac{h}{|\boldsymbol{v}|}$, h being the length scale of elements in the $\boldsymbol{v}$ direction.

One advantage of this weak formulation is that it satisfies an energy law [10,15]. If we take $\tilde{\boldsymbol{v}} = \boldsymbol{v}$ and $\tilde{\psi} = \frac{\lambda}{\epsilon^2}(\psi + s\phi)$ (noting that ϕ and ψ belong to the same function space $\mathscr{S}$), and add Eqs. (13) and (16) together, setting $\boldsymbol{\tau}_\mathrm{d} = 0$ for the moment, we arrive at an energy law that specifies the dissipation of the total energy of a Newtonian–Newtonian mixture:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \left(\frac{1}{2} \rho |\boldsymbol{v}|^2 + f_\mathrm{mix} \right) \mathrm{d}\Omega = - \int_\Omega \left[\left(\frac{1-\phi}{2} \mu_\mathrm{n} + \frac{1+\phi}{2} \mu_\mathrm{s} \right) \nabla \boldsymbol{v} : \nabla \boldsymbol{v} + \gamma |\nabla G|^2 \right] \mathrm{d}\Omega. \tag{18}$$

If the polymer stress is retained, a similar energy law, albeit in a much more complex form, can be written for a Newtonian–Oldroyd-B system that also includes the elastic energy of the dumbbells. Eq. (18) guarantees that the total energy of the system (excluding thermal energy) will decrease from internal dissipation. In general, such energy laws play an important role in the existence and convergence of finite-dimensional solutions to partial differential equations [10]. Note that the use of the body force $G\nabla\phi$ in Eq. (10), instead of the divergence of an interfacial stress tensor, eliminates the second-order derivatives which cannot be handled by C^0 elements. In treating similar problems for liquid-crystal flows, Liu and Walkington [20,21] employed Hermite elements and a mixed method to comply with an energy law.

In the derivation of the weak formulation, all surface integration has been neglected for simplicity, which implies that natural boundary conditions are satisfied. The boundary conditions can be thus summarized as:

$$\boldsymbol{v} = \boldsymbol{v}_\mathrm{g}, \ \text{on}(\partial\Omega)_\mathrm{u} \tag{19}$$

$$(-p\boldsymbol{I} + \boldsymbol{\tau}) \cdot \boldsymbol{n} = 0, \ \text{on}(\partial\Omega)_\tau \tag{20}$$

$$\boldsymbol{\tau}_\mathrm{d} = \boldsymbol{\tau}_\mathrm{in}, \ \text{on}(\partial\Omega)_\mathrm{in} \tag{21}$$

$$\nabla\phi \cdot \boldsymbol{n} = \nabla\psi \cdot \boldsymbol{n} = 0, \ \text{on } \partial\Omega \tag{22}$$

where $\partial\Omega = (\partial\Omega)_\mathrm{u} \cup (\partial\Omega)_\tau$ and $(\partial\Omega)_\mathrm{u} \cap (\partial\Omega)_\tau = \emptyset$, $(\partial\Omega)_\mathrm{in}$ is the inflow boundary, and $\boldsymbol{n}$ is the unit normal to the boundary. The zero-flux conditions in Eq. (22) help maintain volume conservation of the phases. This point will be revisited at the end of this section.

On an unstructured triangular mesh (see Section 3.2), we have used piecewise quadratic (P2) elements for $\boldsymbol{v}$, ϕ and ψ, and piecewise linear (P1) elements for p and $\boldsymbol{\tau}_\mathrm{d}$. After spatial discretization of Eqs. (13)–(17), the nonlinear algebraic system can be written in the following general form:

$$\boldsymbol{\Lambda} \cdot \left(\frac{\partial \boldsymbol{U}}{\partial t} \right)^{n+1} + \boldsymbol{F}(\boldsymbol{U}^{n+1}) = 0, \tag{23}$$

where $\boldsymbol{U}$ is the solution vector, known at time step n and unknown at the next step $n+1$, $\boldsymbol{\Lambda}$ is a diagonal matrix with 1 or 0 on the diagonal depending on whether the corresponding $\boldsymbol{U}$ component appears in a time derivative, and $\boldsymbol{F}(\boldsymbol{U})$ contains all the other terms. For temporal discretization, we have used the Crank–Nicholson scheme and the three-point backward difference scheme [2], and these second-order implicit schemes give nearly identical results. Eq. (23) is solved using an inexact Newton's method with backtracking [22] for enhanced convergence and stability. To save computational cost, the Jacobian matrix is updated once

ClearCorrect Exhibit 1154, Page 5 of 21
ClearCorrect v. Align - IPR2025-00818

in several iterations. Within each Newton iteration, the sparse linear system is solved by preconditioned Krylov methods such as the generalized minimum residual (GMRES) method and the biconjugate gradient stabilized (BCGSTAB) method. ILU(0) and ILUT preconditioners are found to be robust for the calculations in this paper. By design, our grid size varies greatly from the interface to the bulk. This gives rise to a highly ill-conditioned sparse matrix, which is treated by a scaling procedure whereby a diagonal scaling matrix is left-multiplied to the mass matrix prior to applying the aforementioned preconditioners. Thus, each row of the matrix is scaled by the inverse of the sum of the absolute values of the entries in that row before the linear system is sent to the preconditioned Krylov solvers.

The phase-field theory introduces three model parameters that do not have direct counterparts in classical fluid dynamics: λ, ϵ and γ. The capillary width ϵ will be chosen as small as possible to approximate the sharp-interface limit. This point will be amplified in Subsection 4.1. Once ϵ is chosen, the mixing energy density λ is determined, via Eq. (2), by the requirement to match a prescribed interfacial tension. The choice of the mobility γ is one of the subtleties of the diffuse-interface model. It determines the time scale of Cahn–Hilliard diffusion, which is not measurable in macroscopic experiments. Thus, γ has been chosen pragmatically, large enough for diffusion to maintain a more or less constant interfacial thickness, but small enough to avoid excessive damping [6,10].

The latter requirement is essential for conserving the volume of the drops. As is well known, the Cahn–Hilliard dynamics is conservative in the sense that $\int \phi \, d\Omega$ remains a constant in the absence of boundary fluxes [7]. However, as the interfacial profile evolves in a flow field, Cahn–Hilliard diffusion may shift the interface contour $\phi = 0$ and effectively change the volume of a drop. This effect can be minimized by using a sufficiently small γ. For the simulations reported in this paper, we have used γ values ranging from 10^{-5} to 10^{-2}. The volume change of drops is typically on the order of 1%.

3.2. Adaptive mesh generation

As mentioned in Section 1, the phase-field model allows one to use a fixed Eulerian mesh to capture moving internal boundaries. We need a mesh with dense grids covering the interfacial region and coarser grids in the bulk. As the interface moves out of the fine mesh, the mesh in front needs to be refined while that left behind needs to be coarsened. Such adaptive meshing is achieved by using a general-purpose mesh generator GRUMMP, which stands for *G*eneration and *R*efinement of *U*nstructured *M*ixed-Element *M*eshes in *P*arallel [18]. GRUMMP generates a mesh by using Delaunay refinement, and allows control of the internal grid size by using a scalar field. Our phase-field variable ϕ can easily be used to compute such a field. GRUMMP produces triangular elements in 2D and tetrahedral elements in 3D but the latter are not used in this work.

The basic schemes in GRUMMP follow the work of Ruppert [23] and Shewchuk [24], with several significant improvements in the areas of cell size and grading control [25] and meshing from curved boundaries [26]. Ruppert's scheme [23] begins with a constrained Delaunay triangulation. The mesh quality is improved through point insertion at the circumcenter of badly shaped cells having an angle smaller than a threshold $\theta_{\min}$. If a proposed new point encroaches on a boundary edge, that vertex is not inserted. Instead, the encroached boundary edge is bisected. This process is repeated until all cells are well-shaped. Ruppert [23] was able to prove that this algorithm always terminates successfully with minimum angle $\theta_{\min} = 20.7°$. Later variations on Ruppert's scheme [24,25] can provably attain angles of $\theta_{\min} = 25.7°$, and in practice, these schemes work even for larger $\theta_{\min}$ [24]. In the simulations reported here, we have used a minimum angle $\theta_{\min} = 30°$.

GRUMMP controls the spatial variation of grid size using a length scale L_S, which specifies the intended grid size at each location in the domain. It is computed from a "local feature size" l_s defined as the radius of the smallest circle centered at a point that intersects two disjoint parts of the domain boundary [23]. Thus, having l_s at an arbitrary point $\boldsymbol{p}$, one determines the local grid size by:

$$L_S(\boldsymbol{p}) = \min\left[\frac{l_s(\boldsymbol{p})}{R}, \min_{\text{Neighbors } \boldsymbol{q}_i}\left(L_S(\boldsymbol{q}_i) + \frac{|\boldsymbol{q}_i - \boldsymbol{p}|}{G}\right)\right], \tag{24}$$

where the inner minimum is taken among all pre-existing neighboring points $\boldsymbol{q}_i$, and R and G are user-supplied constant resolution and grading parameters. A larger R produces a finer local mesh and a larger G results in

ClearCorrect Exhibit 1154, Page 6 of 21
ClearCorrect v. Align - IPR2025-00818

slower increase in cell size over distance away from the local feature [25]. When the length scale at a vertex is changed, all of its neighbors are added to the set of vertices whose length scale must be checked; this ensures correct propagation of size and grading changes through the mesh. The value of L_S is stored at every vertex, and a cell is considered too large whenever its circumradius exceeds $\sqrt{2}/2$ times the average L_S of its vertices. Such cells are refined using the point-insertion or boundary-bisection algorithm described above for badly shaped triangles. In regions of the mesh where triangles are smaller than the local length scale requires, coarsening is applied. A subset of the mesh vertices is selected with the goal of retaining as many vertices as possible while ensuring that no two selected vertices are closer together than 0.9 times the average of their length scales. Then unwanted vertices are removed one at a time from the fine mesh [27].

In our work, the grid size distribution is dictated by the need to resolve thin interfaces. Since the phase-field variable ϕ is constant (± 1) in the bulk but varies steeply across the interface, we can impose a prescribed small grid size h_1 on the interface by making L_S depend on $|\nabla\phi|$ on every nodes:

$$L_S(x,y) = \frac{1}{|\nabla\phi|\frac{\sqrt{2}}{C} + \frac{1}{h_\infty}}, \tag{25}$$

where h_∞ is the mesh size in the bulk, and the constant C controls the mesh size in the interfacial region: $h_1 = L_S|_{\phi=0} \approx C \cdot \epsilon$, ϵ being the capillary width. In this paper, we have used C values between 0.5 and 1; $h_1 \leqslant \epsilon$ ensures that the thickness of the interface typically contains on the order of 10 grid points [10]. Furthermore, the length scale h_∞ can be set to differing values h_2 and h_3 in the two bulk fluids. This will allow, for example, the interior of a drop to be more finely resolved than the far field of the suspending fluid. Of course, the benchmarks h_1, h_2 and h_3 are guidelines that most of the elements satisfy approximately but not exactly. A grading factor $G \sim 5$ is found to produce generally satisfactory transitions among different regions of the mesh. Fig. 1 shows an example of the mesh inside a square containing an ellipse.

Note that the interface is enveloped in a ribbon of fine grids. Depending on the normal velocity of the interface, it will approach the boundary of the "fine-mesh ribbon" in a certain number of time steps, at which time GRUMMP is called to refine and coarsen the neighboring regions in the front and rear, respectively. This ribbon is taken to comprise three layers of the finest triangles. When the interface goes beyond the second triangle from the middle, remeshing is invoked. The solution of the last time step is then projected onto the new grid in a scheme described by Hu et al. [2]. The speed at which the interface traverses the fine-mesh ribbon also constrains the time step Δt. We require that the interface not advance more than a whole cell at one time step:

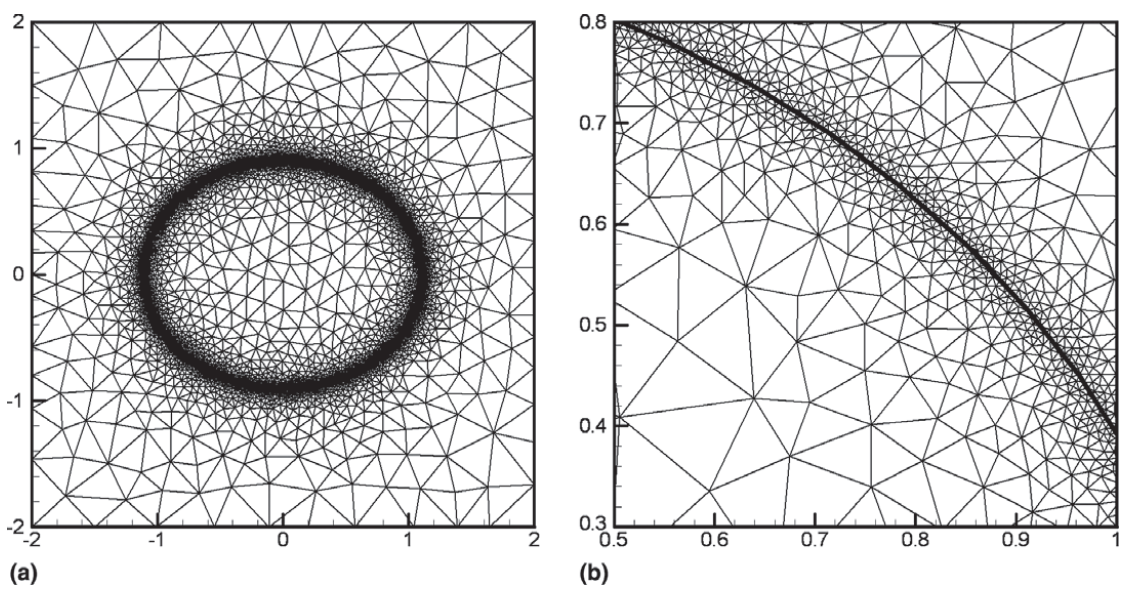

Fig. 1. (a) An unstructured triangular mesh generated by GRUMMP with interfacial refinement. The parameters are $G = 3$, outer boundary mesh size $h_2 = 0.5$, interior mesh size $h_3 = h_2/2$, and interfacial mesh size $h_1 = h_2/64$. (b) Magnified view of a portion of the interfacial region. The bold curve indicates the interface $\phi = 0$, which is centered in a belt-like region of refined triangles.

ClearCorrect Exhibit 1154, Page 7 of 21
ClearCorrect v. Align - IPR2025-00818

$$\Delta t \leqslant \Delta t_{\text{int}} = \min_{\text{All interfacial cells}} \frac{h}{|\boldsymbol{v} \cdot \boldsymbol{n}|}, \tag{26}$$

where $\boldsymbol{v}$ is the interfacial velocity, $\boldsymbol{n}$ is the unit normal to the interface and h is the cell size along $\boldsymbol{n}$.

Since GRUMMP was written in C++ while the finite-element solver is in Fortran, a special set of routines were written to bridge the two. At the beginning of the simulation, GRUMMP is called to generate the initial grid, which is passed onto the finite-element solver. When the updated interfacial position requires remeshing, the instruction is given to GRUMMP, which generates the new grid and passes it to the finite-element solver. This process repeats many times during a typical run, and the passing of information occurs through memory rather than disk I/O for higher efficiency.

Adaptive meshing algorithms have been developed by several groups in the past for computations using the boundary integral method [28], interface-fitting finite elements [2,29], and Eulerian methods with front-tracking, VoF or level sets [30–34]. To adaptively evolve an Eulerian grid, the typical approach is local refinement of cells that straddle the interface [30,31,33]. Some authors have added bridging elements to maintain mesh conformity [32,35]. Zheng et al. [34] impose a global control of the grid size distribution by requiring the grid size to vary linearly with distance from an interface. This scheme fits conveniently into their level-set framework. AMPHI adopts a similar global control; the phase-field variable ϕ provides a natural means for controlling the grid size relative to the location of the interface. Owing to the Eulerian nature of the grid, we need not account for the exact location of internal boundaries. Mesh refinement and coarsening occur relatively infrequently, on the order of tens of time steps for the simulations reported here. This contrasts, for example, the ALE strategy that tracks the interfacial grid points every time step [2].

Since this paper adopts a somewhat unusual theoretical framework and develops a new numerical toolkit, we will confine ourselves to 2D and axisymmetric 3D geometries. These simpler problems facilitate benchmarking the numerical code against published solutions and demonstrating the special features of the method. Besides, there are interesting applications, e.g. in micro-fluidics and biomedical engineering, that involve interfacial flows in axisymmetric geometry. It is worthwhile to develop a numerical tool suited for these. Both the GRUMMP mesh generator and the finite-element solver have been generalized to full 3D, and AMPHI will be upgraded to 3D in due course.

4. Numerical results and discussion

We will present numerical solutions of five problems: drop deformation in shear flows, drop deformation in uniaxial extension, rise of a drop in a quiescent fluid, retraction of an elongated drop, and retraction of a torus. The motion is driven by external flow, buoyancy and interfacial tension, respectively. For some of these problems, comparison with solutions in the literature serves to validate our theoretical and numerical models, and to demonstrate the capabilities and limitations of our tools. For others, we will also explore the role of viscoelasticity in the interfacial dynamics.

4.1. Shear-induced drop deformation

We use two-dimensional shear-induced drop deformation in Newtonian systems to illustrate the convergence of the diffuse-interface model as the capillary width ϵ decreases toward zero. This is an important issue since the diffuse-interface theory reduces to the classical sharp-interface theory as $\epsilon \to 0$, but prior simulations without adaptive meshing typically used relatively large ϵ values and suffered numerical errors due to excessive interfacial diffusion [10,15]. The computation setup is the same as that in Yue et al. [10], with a rectangular domain of size $8\pi a \times 8a$, $a = 0.25$ being the radius of the undeformed drop. The drop is located at the center of the domain. The upper and lower walls are moving at equal speed and in opposite directions. The only difference from Yue et al. [10] is that the undisturbed linear velocity profiles are imposed on the left and right boundaries, while periodic condition is used in [10]. The grid size is controlled by $h_1 = 0.5\epsilon$ at the interface (except for $h_1 = 0.0015$ when $\epsilon = 0.0025$), $h_2 = 0.1$ inside the drop and $h_3 = 0.2$ outside the drop. The drop deformation parameter is defined as $D = (L - B)/(L + B)$, L and B being the longest and shortest lengths from

ClearCorrect Exhibit 1154, Page 8 of 21
ClearCorrect v. Align - IPR2025-00818

the center to the drop interface. The capillary number is defined as $Ca = \mu_m \dot{\gamma} a/\sigma$, where μ_m is the viscosity of the matrix, and $\dot{\gamma}$ is the undisturbed shear rate.

Fig. 2 shows the variation of the drop deformation D with time for several values of ϵ, where t has been made dimensionless by $\dot{\gamma}$. In this case $Ca = 0.1$, and the drop and matrix phases have the same viscosity. For $\epsilon = 0.01$, the deformation first increases to a maximum, and then declines slightly in time. As discussed by Yue et al. [10], this sag is the result of the Cahn–Hilliard diffusion within an interface that is artificially too thick. If we take the interfacial thickness δ_n to be 7.5ϵ [10], then δ_n/a is as large as 0.3. For a real drop of radius $a = 10\ \mu$m and an interfacial thickness δ_m of tens of nanometers, the ratio δ_m/a is on the order of 0.001. Interestingly, as we decrease ϵ from 0.01 to 0.0025, the deformation curve $D \sim t$ evidently converges to a limiting curve, which corresponds to $\epsilon = 0$ and thus the sharp-interface limit. Take the time $t = 4$ for example, $D = 1.061$, 1.085, 1.087 for $\epsilon = 0.01$, 0.005, 0.0025, respectively. Thus for the current drop size $a = 0.25$, $\epsilon = 0.005$ is adequate although δ_n/a is still some two orders of magnitude greater than realistic values. All simulations in the rest of the paper have used $\epsilon \leqslant 0.02$. A curve from our previous calculation using a spectral method [10] is also shown for comparison. Despite the large $\epsilon = 0.01$ used, the spectral curve agrees well with the finite-element results at smaller ϵ, and shows much less sagging than the finite-element curve at $\epsilon = 0.01$. This is because in the previous calculations the interfacial profiles have been carefully relaxed before the shear starts. The adaptive finite-element mesh allows us to use smaller ϵ values to simulate thinner interfaces as to obviate the need for such pre-relaxation.

Table 1 shows how the current AMPHI code outperforms the previous spectral code. Using Fourier–Chebyshev spectral discretization, we have to lay down a large grid of 1024×512 to resolve the interface with $\epsilon = 0.01$. Thanks to adaptive meshing, only 9084 cells are needed in AMPHI for the same ϵ. Furthermore, the spectral code calculates the non-linear terms explicitly and the time marching is subject to a CFL constraint on the time step; $\Delta t = 0.002$ is used in the tabulated case. On the other hand, AMPHI is fully implicit and the

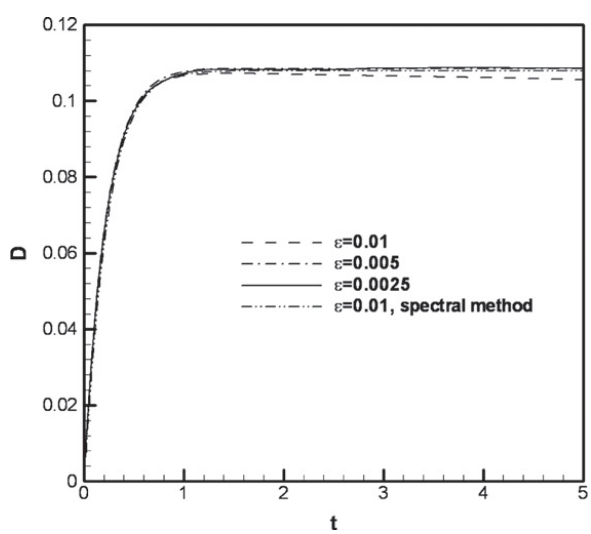

Fig. 2. Shear-induced deformation of a Newtonian drop in a Newtonian suspending fluid. The drop-to-matrix viscosity ratio $\beta = 1$, and $Ca = 0.1$.

Table 1
Comparison of grid numbers and computational (wall-clock) times between AMPHI and the spectral code of Yue et al. [10]

Case	ϵ	Grid number	Computational time till $t = 5$ (min)
Spectral method	0.01	1024×512	180
AMPHI	0.01	9084 cells	20
AMPHI	0.005	17,636 cells	66
AMPHI	0.0025	29,169 cells	153

ClearCorrect Exhibit 1154, Page 9 of 21
ClearCorrect v. Align - IPR2025-00818

main concern for the time step is $\Delta t \leqslant \Delta t_{\text{int}}$ (see Eq. 26). Because the flow is transient, we have used in this case a very (maybe overly) conservative time step to guarantee accurate temporal resolution:

$$\Delta t = \min(0.25\Delta t_{\text{int}}, 0.02). \tag{27}$$

The overall performance is illustrated by the computational time on a single 3.0 GHz Xeon processor. Although the spectral code utilizes FFT, it still takes much longer than AMPHI for the same ϵ. The speedup affords AMPHI the freedom to simulate thinner interfaces with better resolution.

4.2. Drop deformation in elongational flow

The transient deformation of a drop after startup of a uniaxial elongation has been studied in detail by Hooper et al. [36] using a moving-grid finite-element scheme. By comparison with their results, we seek to verify convergence with respect to the interfacial mesh size h_1, and to validate our treatment of unequal viscosity in the two components and the viscoelastic stress in an Oldroyd-B component. The computational domain is a square box as shown in Fig. 3, with $H = W = 10R_0$, R_0 being the radius of the initial spherical drop. The geometry is axisymmetric with respect to the y axis. The flow is inertialess and the initial polymer stress is zero everywhere. At $t = 0$, the drop is subject to an uniaxial extensional flow $(u, v) = (-0.5\dot{\epsilon}x, \dot{\epsilon}y)$, $\dot{\epsilon}$ being the extension rate. The capillary number is defined as $Ca = \mu_{\text{m}}\dot{\epsilon}R_0/\sigma$, and the viscosity ratio is $\beta = \mu_{\text{d}}/\mu_{\text{m}}$, μ_{d} and μ_{m} being the steady shear viscosity of the drop and matrix fluid, respectively. For the Oldroyd-B fluid, we define the Deborah number as $De = \lambda_H\dot{\epsilon}$. The solvent and polymer each contribute half of the total shear viscosity: $\mu_{\text{s}} = \mu_{\text{p}}$. All the parameters have been made dimensionless by R_0, $\dot{\epsilon}$ and μ_{m}. The following physical and mesh parameters are used: $Ca = 0.1$, $\epsilon = 0.01$, mesh size inside the drop $h_2 = 0.2$, and mesh size outside the drop $h_3 = 0.5$. Several values for the interfacial mesh size h_1 will be tested.

As in Hooper et al. [36], we compute only one quadrant of the meridian plane, with undisturbed velocities imposed on the outer boundaries and proper symmetry conditions on the axes of axisymmetry (y-axis) and symmetry (x-axis). When the matrix phase is an Oldroyd-B fluid, we impose the following boundary conditions for the extra stresses at the inflow boundary $(\partial\Omega)_{\text{in}}$:

$$\begin{cases} \tau_{\text{d}xx} = -\frac{\mu_{\text{p}}\dot{\epsilon}}{1+De}\left\{1 - \exp\left[-\frac{t}{\lambda_{\text{H}}}(1+De)\right]\right\}, \\ \tau_{\text{d}yy} = \frac{2\mu_{\text{p}}\dot{\epsilon}}{1-2De}\left\{1 - \exp\left[-\frac{t}{\lambda_{\text{H}}}(1-2De)\right]\right\}, \\ \tau_{\text{d}\theta\theta} = -\frac{\mu_{\text{p}}\dot{\epsilon}}{1+De}\left\{1 - \exp\left[-\frac{t}{\lambda_H}(1+De)\right]\right\}, \\ \tau_{\text{d}xy} = 0, \end{cases} \tag{28}$$

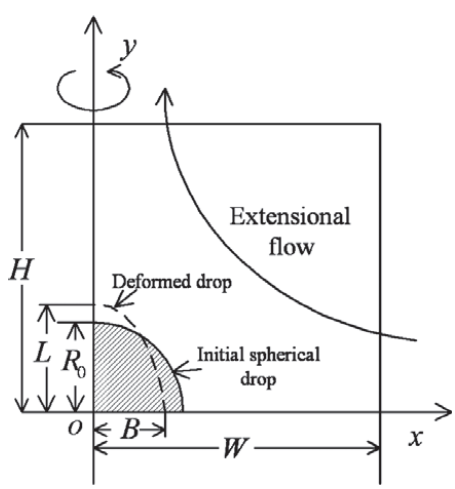

Fig. 3. Schematic of the computational domain for transient drop deformation after abrupt inception of a uniaxial extensional flow. The flow is axisymmetric with respect to the y axis and the domain lies in the meridional plane. The drop size is exaggerated; the actual domain size has $H = W = 10R_0$.

where θ is the azimuthal coordinate. These are the transient stresses after startup of uniaxial elongation in the absence of the drop [37].

Fig. 4 plots the deformation of a Newtonian drop in a Newtonian matrix for three viscosity ratios $\beta = 0.5$, 1 and 2. Fig. 4(a) shows excellent agreement between our diffuse-interface calculations and the sharp-interface calculations of Hooper et al. [36]. The absolute discrepancy in L/R_0 at $t = 4$ is less than *0.002* for all the three viscosity ratios. In previous simulations [10,11], we have mostly assumed $\mu_n = \mu_s$ such that the viscous stress in Eq. (9) is independent of ϕ. This reduces the degree of coupling between the Cahn–Hilliard and momentum equations and facilitates convergence. Non-equal viscosities present a difficulty for the spectral method. The finite-element results in Fig. 4 demonstrate the capability of AMPHI in handling a range of viscosity ratios. Fig. 4(*b*) examines the convergence of the result with respect to the interfacial grid size h_1. When h_1 decreases from ϵ to 0.5ϵ, the trend of convergence is evident. All subsequent calculations of this subsection use $h_1 = 0.5\epsilon$.

For four Deborah numbers $De = 0.1$, 0.2, 1 and 2, Fig. 5 shows the deformation curves of Newtonian drops in Oldroyd-B matrices (N/O) and Oldroyd-B drops in Newtonian matrices (O/N). The curve of a Newtonian

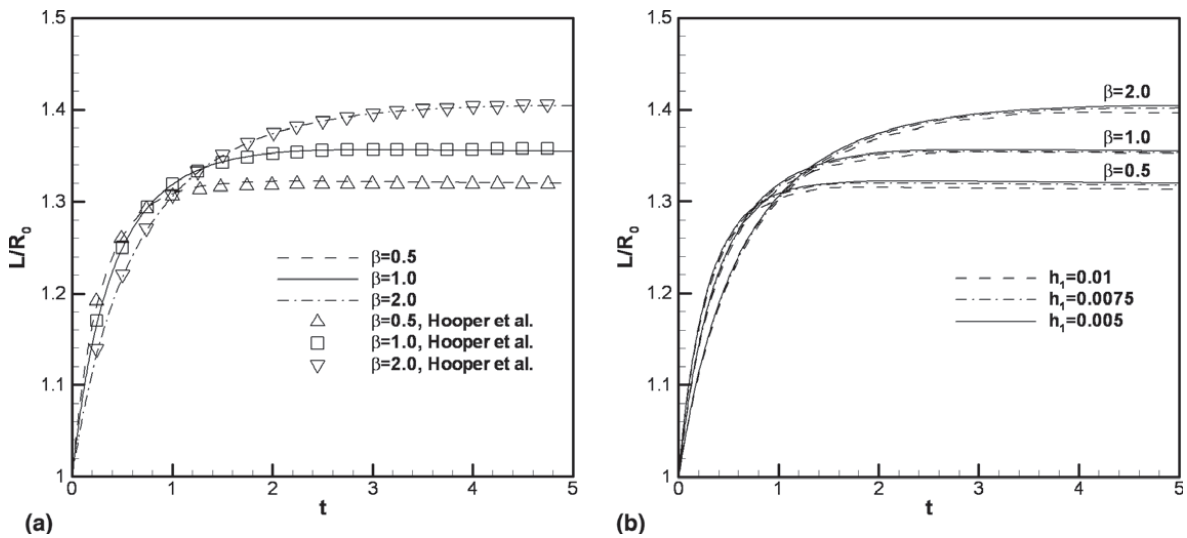

Fig. 4. Transient deformation of a Newtonian drop in a Newtonian matrix. (a) Comparison of our results with those of Hooper et al. [36] for three viscosity ratios. (b) Convergence of the solutions with respect to mesh size h_1.

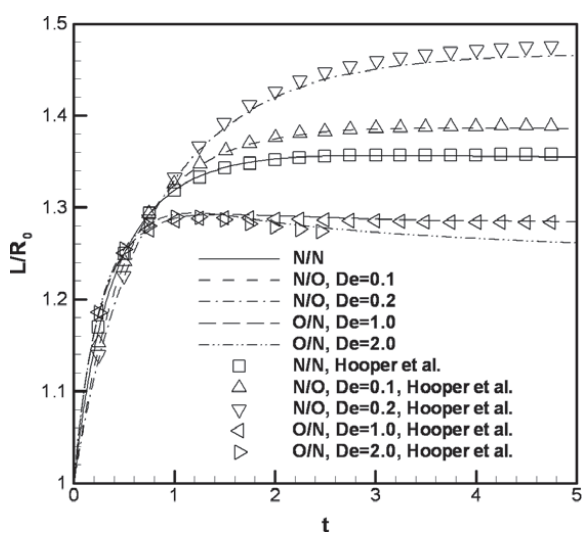

Fig. 5. Viscoelastic effects on drop deformation in elongation flow. N/O refers to a Newtonian drop in an Oldroyd-B matrix, and O/N is the inverse. Smaller Deborah numbers are tested for N/O since the Oldroyd-B matrix develops infinite elongational stresses as $De \to 0.5$.

drop deforming in a Newtonian matrix (N/N) is also provided as a baseline. The viscosity ratio is $\beta = 1$. For the N/O cases, viscoelasticity in the matrix enhances drop deformation, and the drop deforms more with increasing Deborah number. This is in agreement with prior calculations and measurements, and forms an interesting contrast with the trend in shear flows [13]. At $t = 4$, our results $L/R_0 = 1.386$ and 1.462 for $De = 0.1$ and 0.2 are very close to $L/R_0 = 1.388$ and 1.471 obtained by Hooper et al. [36].

For the O/N cases, drop deformation is inhibited by viscoelasticity in the dispersed phase. As in Hooper et al. [36], we observe an overshoot in the growth of L/R_0 in time, and its magnitude increases with De. Quantitatively, the agreement between the two studies is very close. For $De = 1$, $L/R_0 = 1.286$ at $t = 4$ in our calculation, while the corresponding result in [36] is 1.284. For $De = 2$, our result again agrees well with that of Hooper et al. [36] for $t < 2.5$ (their calculation fails at $t = 2.5$). At $t = 2$, for example, our $L/R_0 = 1.284$ is only slightly larger than their 1.279.

4.3. Rising drops

These simulations are motivated by a wealth of experimental data on the rise of Newtonian drops in a Newtonian matrix. Of the five problems studied in this paper, this is the only one in which inertia plays a major role.

Based on experimental data on the motion of liquid drops and air bubbles in liquids, Grace et al. [38,39] have constructed a diagram (Fig. 2 in both references) that describes how the drop or bubble behavior depends on material parameters. We will concern ourselves here only with the steady-state regimes at relatively low rise velocity. Thus, the terminal Reynolds number Re can be examined for ranges of the Eötvos number Eo (or Bond number) and the Morton number M. These dimensionless groups are defined as

$$Re = \frac{\rho_c d_e U}{\mu_c}, \tag{29}$$

$$Eo = \frac{g d_e^2 \Delta\rho}{\sigma}, \tag{30}$$

$$M = \frac{g \mu_c^4 \Delta\rho}{\rho_c^2 \sigma^3}, \tag{31}$$

where $\Delta\rho = |\rho_c - \rho_d|$, U is the terminal velocity of the rising drop, d_e is the effective diameter of the drop, g is the gravitational acceleration, and the subscripts c and d refer to the matrix (continuous phase) and the drop (dispersed phase) respectively. The system has two other dimensionless groups: the density ratio ρ_d/ρ_c and viscosity ratios μ_d/μ_c between the two phases. Experimental data show that the former need not be included as long as $\Delta\rho$ is. The drop viscosity μ_d has little effect on the terminal velocity since contaminants in real systems tend to immobilize the interface and suppress internal flow in any event [38].

If we choose ρ_c, $g/2$ and d_e as the dimensionally independent parameters, all other parameters can be made dimensionless: $\rho_d = 0.5$, $\epsilon = 0.01$, $h_1 = 0.0075$, $h_2 = 0.2$, $h_3 = 0.5$. The dimensionless viscosity and interfacial tension are given in Table 2 and the viscosity ratio is fixed: $\beta = \mu_d/\mu_c = 1$. Assuming axisymmetry, we compute half of the meridian plane in a rectangular domain of width 4 in the radial direction and length 15 along the axis of symmetry. The drop center is initially at $(0, 2)$. All the boundaries are non-slip walls except the axis of axisymmetry.

Table 2
Five simulated cases of the rising drop

Case	σ	μ_c	U	Eo	M	Re	Re_G	Drop shape
A	0.1	5.62×10^{-1}	0.091	10	100	0.16	0.17	Spherical
B	0.1	5.62×10^{-2}	0.567	10	0.01	10	11	Oblate
C	0.01	5.62×10^{-1}	0.087	100	10^5	0.15	0.17	Spherical
D	0.01	1.78×10^{-1}	0.251	100	10^3	1.4	1.6	Ellipsoidal cap, dimpled
E	0.01	3.16×10^{-2}	0.456	100	1	14	19	Ellipsoidal cap, skirted

Re is the computed terminal Reynolds number, and Re_G is experimental data extracted from Fig. 2 in [38].

ClearCorrect Exhibit 1154, Page 12 of 21
ClearCorrect v. Align - IPR2025-00818

We have simulated five cases corresponding to steady regimes in the Grace diagram with different final drop shapes [38,39]. The parameters and results, in terms of the terminal *Re* and drop shape, are tabulated in Table 2. The column under Re_G lists terminal Reynolds numbers read off the Grace diagram. The agreement between the computed *Re* and measured Re_G is quite good. The fact that *Re* is lower than Re_G is probably due to wall effects. Our computational domain is 15 drop diameters long with closed top and bottom, while the experimental chambers are much larger and the suspending liquid is considered infinite [39]. Another source of uncertainty is the drop viscosity μ_d. Grace argued that the effect of drop viscosity has been superseded in the experiments by that of surfactants [38], and did not give μ_d values. We have taken $\beta = \mu_d/\mu_c = 1$. In the experiment, the skirted cap tends to have trailing fringes, owing probably to a low μ_d, which are absent in our simulation (cf. Fig. 6d). This may account for the relatively large discrepancy between *Re* and Re_G for case E.

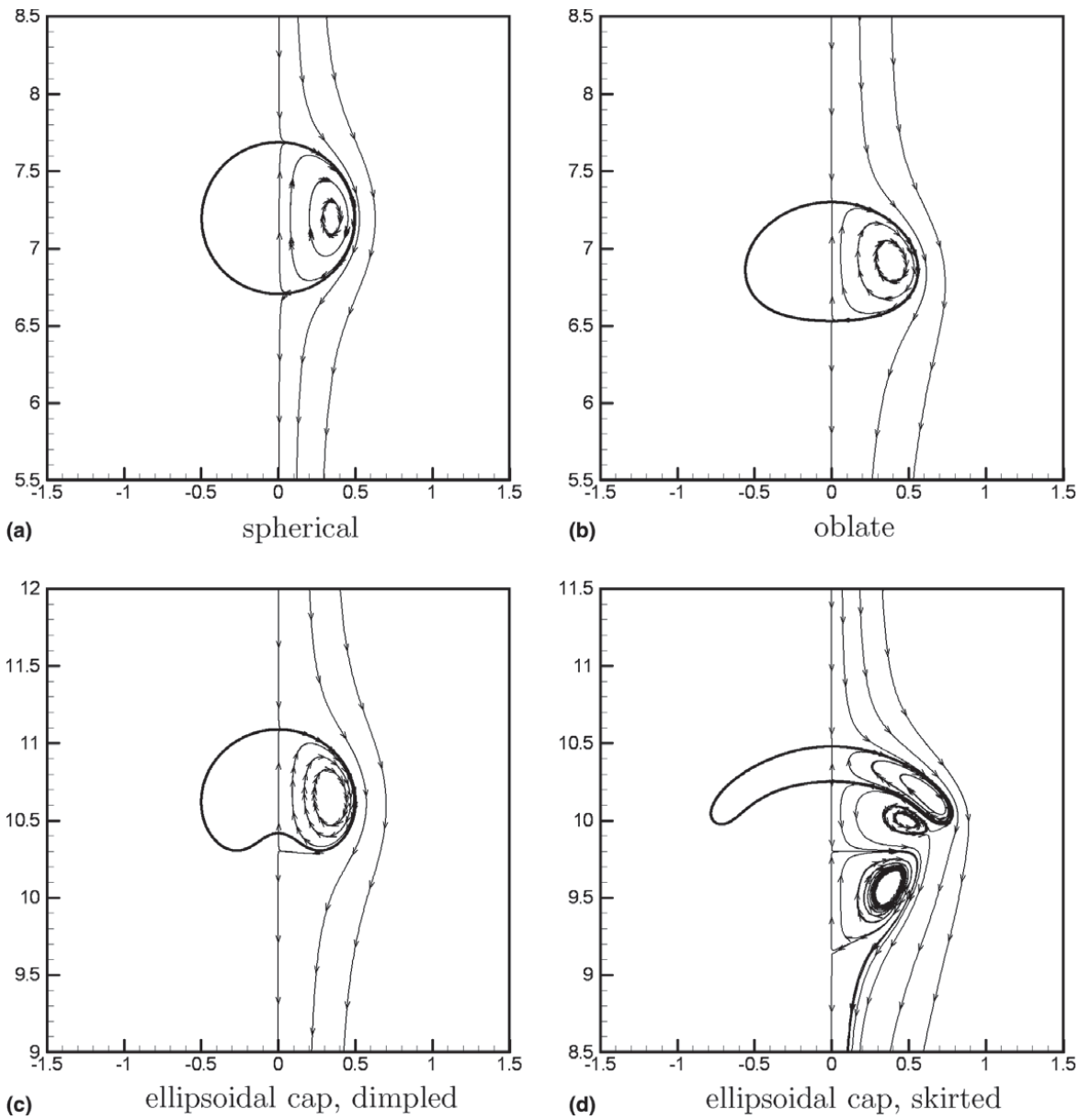

Fig. 6. Computed drop shapes and the flow field in the final steady state. The streamlines are in a reference frame attached to the center of mass of the drop. A mirror image has been added to show the full shape of the drops. (a), (b), (c) and (d) correspond to case A, B, D and E in Table 2, respectively.

Fig. 6 shows the steady-state drop shapes and the flow field in the different regimes. The drop shapes are in visual agreement with the Grace diagram. The streamlines cannot be compared with quantitative experimental data since none is available. Note that for the dimpled ellipsoidal cap (c), there is apparently a normal velocity at the center of the dimple. This is a numerical artifact due to Cahn–Hilliard diffusion at interfacial locations of high curvature. The magnitude of this velocity is much smaller than the rise velocity. Because of the concavity of the underside of the skirted cap in (d), flow separation occurs in the wake and produces two counter-rotating vortex rings.

We have also calculated the rise of a Newtonian drop in an Oldroyd-B matrix. The steady-state drop shape is shown in Fig. 7 for a polymer relaxation time $\lambda_H = 5$. The drop's terminal velocity is $U = 0.086$, resulting in $Re = 0.148$ and a Deborah number of $De = U\lambda_H/d_e = 0.43$. The most interesting feature is the almond shape of the drop, distinct from those in Fig. 6. This is evidently due to the large extensional stress that develops in the wake of the drop, as indicated by the stress contours. For the same reason, the rise velocity is smaller than the corresponding Newtonian case A. With increasing λ_H, our simulations show that the polymer stress pulling the drop downward becomes stronger. The rise velocity decreases further, and the tail of the drop becomes more pointed. However, the tip never becomes a true cusp because of the Cahn–Hilliard diffusion in the interfacial layer, in interesting contrast to level-set calculations of Pillapakkam and Singh [40].

The rise of gas bubbles has been simulated by a number of groups using finite-difference [41], level set [42], and VoF [43,44] methods. These simulations have employed a free-surface description or very high viscosity and density ratios (up to 1000) to mimic the gas-liquid flow. Such large ratios would present a difficulty to our diffuse-interface formulation because a slight shift of the ϕ field, say due to interfacial diffusion, implies large errors in the averaged density and viscosity in the mixing layer. Thus, we have concentrated on modest density differences that correspond to the rising-drop experiments summarized in the Grace diagram.

4.4. Drop retraction

Consider a spheroidal drop that retracts toward a sphere under interfacial tension in a quiescent matrix fluid. This process has received much attention as an experimental procedure for measuring the interfacial tension between the two phases [45–47]. If both components are Newtonian, Maffettone and Minale [48] proposed a phenomenological model that describes the retraction:

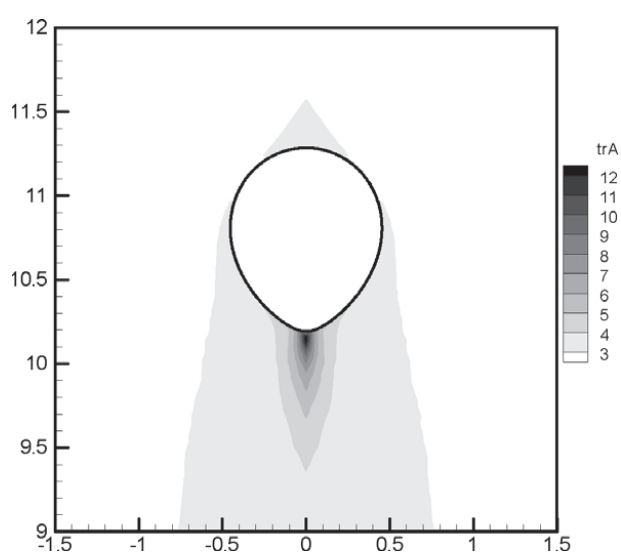

Fig. 7. Steady state shape of a Newtonian drop rising in an Oldroyd-B matrix. The polymer stress in the matrix is depicted by contours of tr($\boldsymbol{A}$), where $\boldsymbol{A} = \frac{\lambda_H}{\mu_p}\boldsymbol{\tau}_d + \boldsymbol{I}$ is the polymer configuration tensor. The steady-state Deborah number $De = 0.43$, $\mu_p = \mu_s = 0.5\mu_c$, and all other material parameters are the same as case A in Table 2.

$$L^2 - B^2 = (L^2 - B^2)_{t=0} \exp\left[-\frac{\sigma}{\mu_m R_0} f(\beta) t\right], \tag{32}$$

where R_0 is the equilibrium drop radius, L and B are the half-length and half-width of the drop (see Fig. 3), μ_m is the matrix viscosity, $f(\beta) = \frac{40(\beta+1)}{(2\beta+3)(19\beta+16)}$, β being the drop-to-matrix viscosity ratio. Thus the time evolution of $L^2 - B^2$ satisfies an exponential law, and from the slope of $\ln(L^2 - B^2)$ one can back out the interfacial tension σ.

As a validation of our code, we first compare simulations of a Newtonian drop in a Newtonian matrix (N/N) with Eq. (32). For the rest of this subsection, all parameters are made dimensionless with respect to R_0, σ and μ_m. The axisymmetric computational domain is as in Fig. 3 with $H = W = 20R_0$. A zero-stress condition (Eq. 20) is imposed at the outer boundaries. Initially, the velocity is zero everywhere, and the initial shape of the drop is a spheroid with $L_0 = 1.3104$, $B_0 = 0.8736$ and a deformation parameter $D_0 = 0.2$. Other parameters are: $\epsilon = 0.005$, $h_1 = 0.004$, $h_2 = 0.2$, $h_3 = 0.4$.

Fig. 8 gives the N/N retraction curves for three viscosity ratios $\beta = 0.2$, 1, and 5. All three curves are straight lines, exhibiting an exponential retraction. If we use their local slope to calculate the apparent interfacial tension σ_{ap} from Eq. (32), the result hovers around 0.92 in all three cases. The scaling is such that σ_{ap} is unity if the retraction obeys Eq. (32) exactly. Thus, the numerically calculated σ_{ap} is slightly smaller than that predicted by the phenomenological theory. This discrepancy does not change when we refine the grid and vary the capillary width ϵ and domain size. Thus, it is not a numerical error. Velankar et al. [47] simulated the retraction of a surfactant-laden drop using a finite-element/front-tracking scheme, and calculated σ_{ap} by fitting a phenomenological model similar to Eq. (32). For zero surfactant concentration (see Fig. 8 of [47]), their calculated apparent interfacial tension is also about 90% of the theoretical one. Thus the 10% difference is probably due to inherent errors in the phenomenological model.

Yue et al. [11] have previously calculated drop retraction in two dimensions, and explored the effects of viscoelasticity on drop retraction. Using the new AMPHI code, we have revisited this problem by comparing 2D and axisymmetric 3D results for a Newtonian drop retracting in a Newtonian matrix (N/N), a Newtonian drop in an Oldroyd-B matrix (N/O) and an Oldroyd-B drop in a Newtonian matrix (O/N). The Deborah number is defined using the capillary time: $De = \lambda_H \sigma/(\mu_m R_0)$. The polymer stress tensor is set to zero initially.

In 2D, the results in Fig. 9(a) essentially reproduce those of Yue et al. [11] obtained by a spectral method. The 3D retraction curves in Fig. 9(b) turn out to be very similar to the 2D ones. Although the retraction is somewhat slower in 3D, the viscoelastic effects are qualitatively the same and have confirmed those noted by Yue et al. [11]. First, N/O and O/N differ very little for the same De, and viscoelasticity modifies drop retraction in practically the same way whether it is in the drop or the matrix component. Second, with viscoelasticity the rate of retraction is initially faster than for N/N, but slows down considerably in longer times.

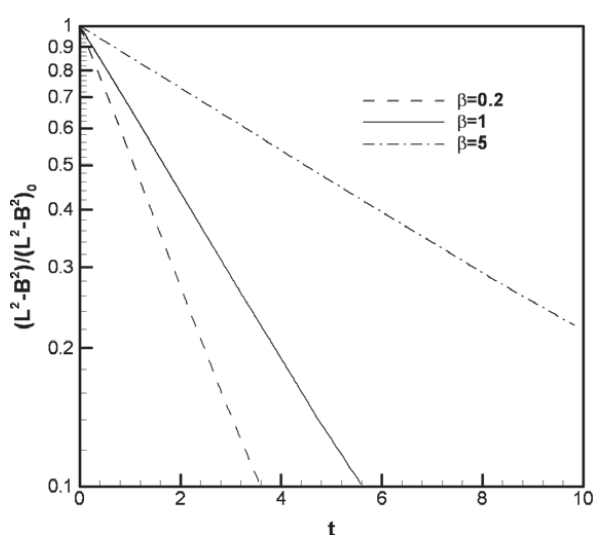

Fig. 8. Effect of the viscosity ratio on the retraction of a Newtonian drop in a Newtonian matrix (N/N).

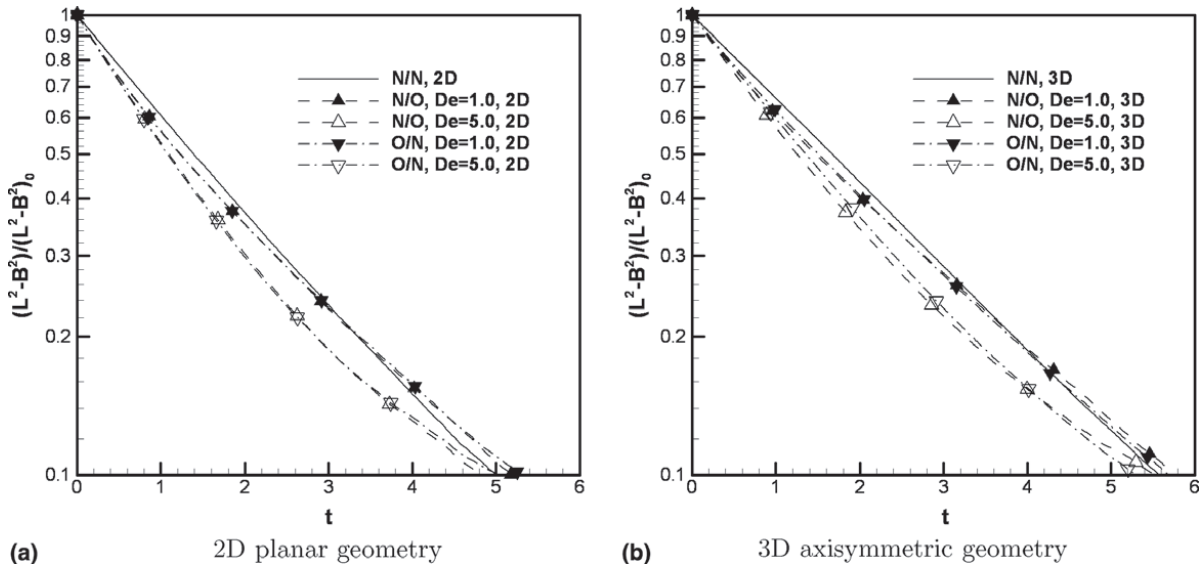

Fig. 9. Drop retraction in viscoelastic fluids. The computation domain is as in Fig. 3 with $W = H = 10$. The viscosity ratio is $\beta = 1$ with $\mu_s = \mu_p = \mu_n/2$ for the Oldroyd-B component. All other parameters are the same as in the N/N simulations in Fig. 8.

4.5. Contraction of a torus

The last set of results concern the contraction of a torus in a quiescent matrix fluid driven by interfacial tension. This problem not only highlights the capability of AMPHI in coping with topological changes, but also affords us the opportunity to study the effect of viscoelastic normal stresses on filament drainage and interfacial coalescence. The computational domain is shown in Fig. 10. All parameters are scaled by the effective radius of the torus R_0, based on a sphere of the same volume, the interfacial tension σ, and the matrix viscosity μ_m. Then the various lengths in the diagram are: $H = 8$, $W = 6$, $a = 0.4607$, and $d = 1$. The capillary width is $\epsilon = 0.01$, and the mesh sizes are $h_1 = 0.0075$, $h_2 = 0.2$ and $h_3 = 0.4$. The total viscosity inside and outside the torus are both set to 1. For Oldroyd-B fluids we set $\mu_p = \mu_s$, $\lambda_H = 1$ so the Deborah number $De = \lambda_H\sigma/(R_0\mu_m) = 1$. Inertia is neglected. The y-axis is the axis of axisymmetry, and no-slip boundary conditions are imposed on the other three boundaries.

Fig. 11 shows snap shots of the interface of a Newtonian torus contracting in a Newtonian matrix. Note first that the toroidal interface has curvature not only in the x–y plane shown here, but also in the third (azimuthal) dimension. The contraction of the torus thus reduces its surface area even as the cross-sectional area increases. As the inner side closes in onto itself, the cross-section takes on a kidney shape (b). The filament of matrix fluid is being squeezed out along the y-axis, producing a high pressure at the center of the torus and the

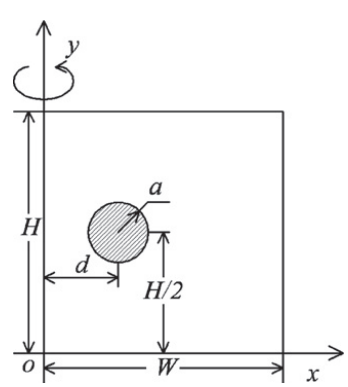

Fig. 10. Initial configuration of a torus. Because of axisymmetry with respect to the y-axis, only a single circular cross-section is shown.

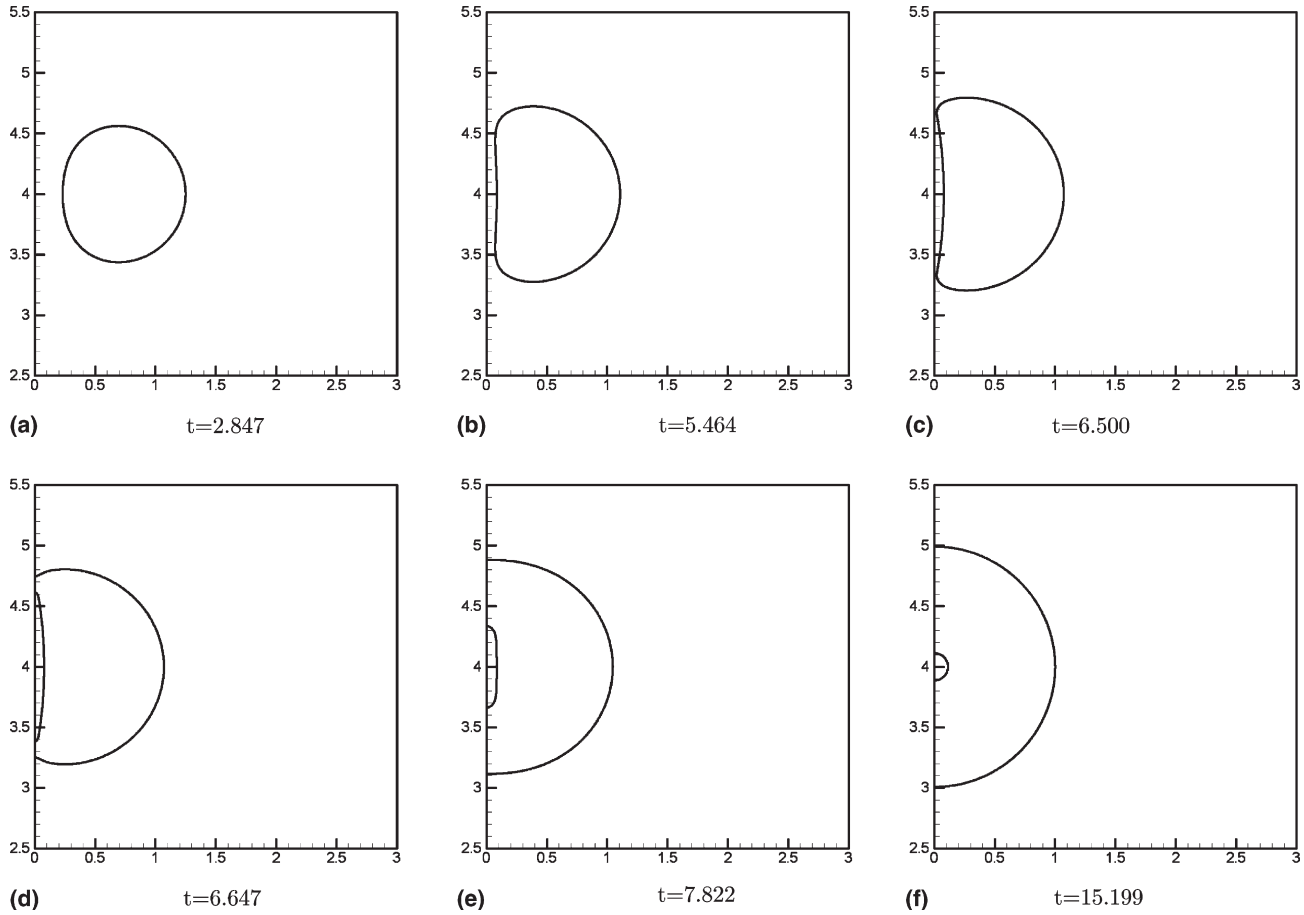

Fig. 11. Snapshots of a Newtonian torus contracting in a Newtonian matrix. Time t is scaled by the capillary time $\mu_m R_0/\sigma$.

concavity of the inner surface. The concavity grows more prominent in time, and the matrix fluid filament develops two thin necks (c). The high curvature around the necks produces a high capillary pressure as shown in Fig. 12, which hinders further drainage of the fluid filament. As explained by Yue et al. [11], a short-range

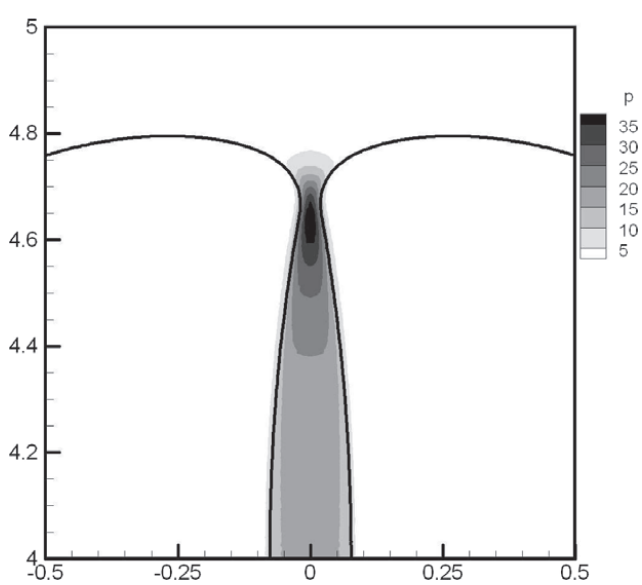

Fig. 12. Pressure distribution at $t = 6.500$ for plot (c) in Fig. 11.

force akin to van der Waals force kicks in between the approaching interfaces, causing the necks to pinch off. A spindle of the matrix fluid is trapped inside the torus fluid (d). It then contracts into a spherical drop enclosed by a spherical shell of the torus fluid (f). The above scenario bears a remarkable similarity to the interfacial evolution during head-on collision and coalescence between two drops [10]. A major difference is that for the torus, the collision of the inner surface onto itself is driven mostly by curvature in the third dimension.

If the matrix fluid is viscoelastic, the contraction of a Newtonian torus proceeds more slowly (Fig. 13). This is readily explained by the development of strong polymer stresses in the matrix filament as it is elongated (Fig. 14). Roughly speaking, the thinning of the thread is driven by the capillary stress $p_c = \sigma/r$, where r is the radius of the thread. We may approximately balance p_c against the polymer tensile stress plus the viscous stress due to elongation. Thus, the very high tensile stress τ_{dyy} near the shrinking neck of the filament (Fig. 14) reduces the extensional rate and prolongs the drainage process. Again, we see an analogy to the film drainage and rupture process when two drops collide in an Oldroyd-B fluid [11].

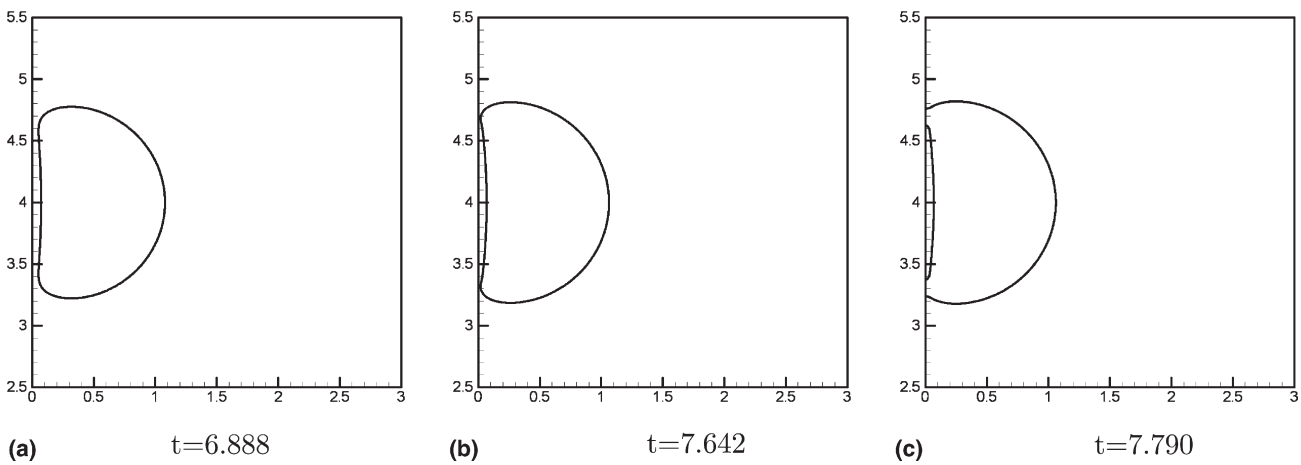

Fig. 13. Retraction of a torus of Newtonian fluid in an Oldroyd-B matrix.

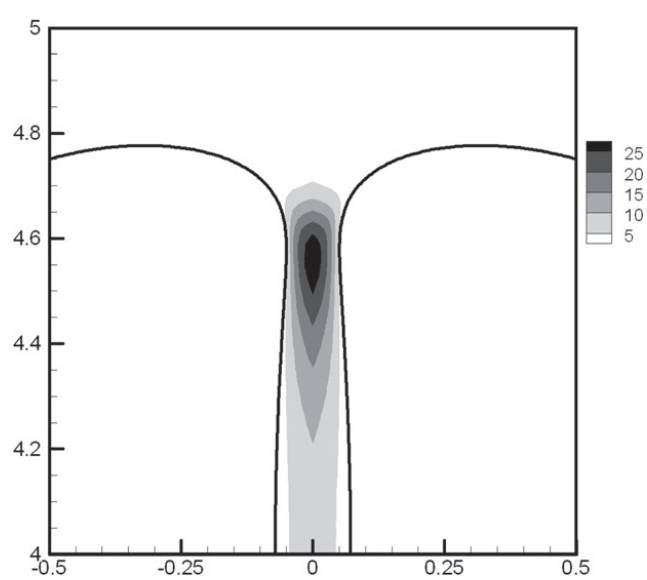

Fig. 14. Contours of the polymer tensile stress τ_{dyy} in the thinning filament of Oldroyd-B matrix. $t = 6.888$ corresponding to plot (a) in Fig. 13.

We have also calculated the contraction of an Oldroyd-B torus in a Newtonian matrix. The process is not much different from the Newtonian-in-Newtonian case because the fluid inside the thread undergoes elongation while the fluid in the torus experiences relatively little straining. Therefore, the rheology of the latter does not have a major role. This forms an interesting contrast to drop retraction discussed in the last subsection; there the viscoelastic effect is equally important in both components.

5. Conclusion

This paper describes AMPHI, a diffuse-interface algorithm for simulating interfacial dynamics in complex fluids, and presents its applications in simulating several flow problems. The numerical code has two major elements: an efficient finite-element solver originally developed for Navier–Stokes systems, and an adaptive meshing scheme using a global scalar field to control the grid size. The phase-field variable ϕ avoids the need to track the moving and deforming interface, and allows discretization and numerical solution within an Eulerian framework. Moreover, its gradient $|\nabla\phi|$ supplies a convenient criterion for adaptive meshing so as to maintain the densest grid over the moving interface. The results of this study may be summarized as follows:

(i) AMPHI represents a major improvement over our previous spectral algorithms developed for similar purposes. Adaptive meshing results in considerable savings in grid numbers, thus making it possible to simulate thinner interfaces with better resolution. This also opens up opportunities for tackling larger-scale flow problems, possibly in 3D. The finite-element method easily accommodates complex flow geometry and various boundary conditions.
(ii) The theoretical model and numerical schemes have been validated by comparing with our own prior results and results in the literature. The fully implicit scheme for coupled Cahn–Hilliard and flow equations allows large time steps and accurate calculation of large interfacial tension. Convergence with interfacial thickness and mesh size has been confirmed, and accuracy of the method has been explored with respect to the model parameters.
(iii) Application of AMPHI to simulate non-trivial flow problems has demonstrated its capability in probing interfacial dynamics in complex fluids, and led to physical insights about the interplay between viscoelasticity in the bulk and deformation and rupture of the interface.

We must also point out the limitations of diffuse-interface methods. The physical model is not easily generalized to accommodate more than two fluid species. Numerically, as is well known, the greatest challenge is the need to resolve thin interfacial layers. Numerical constraints arise from the fundamental issue of using an interfacial thickness that is necessarily much larger than for real fluid interfaces. For instance, convergence becomes difficult for large density or viscosity ratios ($\gtrsim 100$) because of errors caused by slight shifts in the ϕ profile. The progress achieved with AMPHI has suggested adaptive meshing as a promising answer to this challenge. The results reported here are limited by two-dimensionality. However, both the finite- element solver and the adaptive mesh generator have been upgraded to 3D, and we are developing a fully 3D version of AMPHI.

Acknowledgments

Acknowledgment is made to the Donors of The Petroleum Research Fund, administered by the American Chemical Society, for partial support of this research. J.J.F. was also supported by the NSERC, the Canada Foundation for Innovation, the Canada Research Chairs program and the NNSF of China (No. 20490220). C.F.O.-G. acknowledges support by the NSERC, and H.H.H. support by the NSF. The simulations were performed on computers at the Institute of Applied Mathematics, University of British Columbia and WestGrid.

References

[1] W.Z. Huang, Y.H. Ren, R.D. Russell, Moving mesh methods based on moving mesh partial-differential equations, J. Comput. Phys. 113 (1994) 279–290.

ClearCorrect Exhibit 1154, Page 19 of 21
ClearCorrect v. Align - IPR2025-00818

[2] H.H. Hu, N.A. Patankar, M.Y. Zhu, Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian technique, J. Comput. Phys. 169 (2001) 427–462.
[3] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, Ann. Rev. Fluid Mech. 35 (2003) 341–372.
[4] D.M. Anderson, G.B. McFadden, A.A. Wheeler, Diffuse-interface methods in fluid mechanics, Ann. Rev. Fluid Mech. 30 (1998) 139–165.
[5] J. Lowengrub, L. Truskinovsky, Quasi-incompressible Cahn–Hilliard fluids and topological transitions, Proc. Roy. Soc. Lond. A 454 (1998) 2617–2654.
[6] D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modelling, J. Comput. Phys. 155 (1999) 96–127.
[7] C. Liu, J. Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method, Physica D 179 (2003) 211–228.
[8] V.E. Badalassi, H.D. Ceniceros, S. Banerjee, Computation of multiphase systems with phase field model, J. Comput. Phys. 190 (2003) 371–397.
[9] B.J. Keestra, P.C.J. van Puyvelde, P.D. Anderson, H.E.H. Meijer, Diffuse interface modeling of the morphology and rheology of immiscible polymer blends, Phys. Fluids 15 (2003) 2567–2575.
[10] P. Yue, J.J. Feng, C. Liu, J. Shen, A diffuse-interface method for simulating two-phase flows of complex fluids, J. Fluid Mech. 515 (2004) 293–317.
[11] P. Yue, J.J. Feng, C. Liu, J. Shen, Diffuse-interface simulations of drop coalescence and retraction in viscoelastic fluids, J. Non-Newtonian Fluid Mech. 129 (2005) 163–176.
[12] P. Yue, J.J. Feng, C. Liu, J. Shen, Interfacial force and Marangoni flow on a nematic drop retracting in an isotropic fluid, J. Colloid Interface Sci. 290 (2005) 281–288.
[13] P. Yue, J.J. Feng, C. Liu, J. Shen, Viscoelastic effects on drop deformation in steady shear, J. Fluid Mech. 540 (2005) 427–437.
[14] P. Yue, J.J. Feng, C. Liu, J. Shen, Transient drop deformation upon startup of shear in viscoelastic fluids, Phys. Fluids 17 (2005) 123101.
[15] J.J. Feng, C. Liu, J. Shen, P. Yue, An energetic variational formulation with phase field methods for interfacial dynamics of complex fluids: advantages and challenges, in: M.-C.T. Calderer, E. Terentjev (Eds.), Modeling of Soft Matter, Springer, New York, 2005, pp. 1–26.
[16] R.B. Bird, D.F. Curtiss, R.C. Armstrong, O. Hassager, Dynamics of Polymeric Liquids, Vol. 2. Kinetic Theory, Wiley, New York, 1987.
[17] H. Gan, J.J. Feng, H.H. Hu, Simulation of the sedimentation of melting solid particles, Int. J. Multiphase Flow 29 (2003) 751–769.
[18] L.A. Freitag, C.F. Ollivier-Gooch, Tetrahedral mesh improvement using swapping and smoothing, Int. J. Numer. Methods Eng. 40 (1997) 3979–4002.
[19] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comp. Methods Appl. Mech. Eng. 32 (1982) 199–259.
[20] C. Liu, N.J. Walkington, Approximation of liquid crystal flows, SIAM J. Numer. Anal. 37 (2000) 725–741.
[21] C. Liu, N.J. Walkington, Mixed methods for the approximation of liquid crystal flows, Math. Modeling Numer. Anal. 36 (2002) 205–222.
[22] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM J. Sci. Comput. 17 (1996) 16–32.
[23] J. Rupert, A Delaunay refinement algorithm for quality 2-dimensional mesh generation, J. Algorithm 18 (1995) 548–585.
[24] J.R. Shewchuk, Delaunay Refinement Mesh Generation, PhD thesis, Carnegie Mellon University, 1997.
[25] C.F. Ollivier-Gooch, C. Boivin, Guaranteed-quality simplical mesh generation with cell size and grading control, Eng. Comput. 17 (2001) 269–286.
[26] C. Boivin, C.F. Ollivier-Gooch, Guaranteed-quality triangular mesh generation for domains with curved boundaries, Int. J. Numer. Methods Eng. 55 (2002) 1185–1213.
[27] C.F. Ollivier-Gooch, Coarsening unstructured meshes by edge contraction, Int. J. Numer. Methods Eng. 57 (2003) 391–414.
[28] 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.
[29] E.D. Wilkes, S.D. Phillips, O.A. Basaran, Computational and experimental analysis of dynamics of drop formation, Phys. Fluids 11 (1999) 3577–3598.
[30] G. Agresar, J.J. Linderman, G. Trggvason, K.G. Powell, An adaptive, Cartesian, front-tracking method for the motion, deformation and adhesion of circulating cells, J. Comput. Phys. 143 (1998) 346–380.
[31] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, An adaptive level set approach for incompressible two-phase flows, J. Comput. Phys. 148 (1999) 81–124.
[32] I. Ginzburg, G. Wittum, Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants, J. Comput. Phys. 166 (2001) 302–335.
[33] T. Chen, P.D. Minev, K. Nandakumar, A projection scheme for incompressible multiphase flow using adaptive eulerian grid, Int. J. Numer. Methods Fluids 45 (2004) 1–19.
[34] X. Zheng, J. Lowengrub, A. Anderson, 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.
[35] N. Provatas, N. Goldenfeld, J. Dantzig, Adaptive mesh refinement computation of solidification microstructures using dynamic data structures, J. Comput. Phys. 148 (1999) 265–290.
[36] R.W. Hooper, V.F. de Almeida, C.W. Macosko, J.J. Derby, Transient polymeric drop extension and retraction in uniaxial extensional flows, J. Non-Newtonian Fluid Mech. 98 (2001) 141–168.

ClearCorrect Exhibit 1154, Page 20 of 21
ClearCorrect v. Align - IPR2025-00818

[37] R.B. Bird, R.C. Armstrong, O. Hassager, Dynamics of Polymeric Liquids, Vol. 1. Fluid Mechanics, Wiley, New York, 1987.
[38] J.R. Grace, T. Wairegi, T.H. Nguyen, Shapes and velocities of single drops and bubbles moving freely through immiscible liquids, Trans. Inst. Chem. Eng. 54 (1976) 167–173.
[39] J.R. Grace, Hydrodynamics of liquid drops in immiscible liquids, in: N.P. Cheremisinoff, R. Gupta (Eds.), Handbook of Fluids in Motion, Ann Arbor Science, 1983, pp. 1003–1025 (Chapter 38).
[40] S.B. Pillapakkam, P. Singh, A level-set method for computing solutions to viscoelastic two-phase flow, J. Comput. Phys. 174 (2001) 552–578.
[41] G. Ryskin, L.G. Leal, Numerical solution of free-boundary problems in fluid mechanics. Part 2. Buoyancy-driven motion of a gas bubble through a quiescent liquid, J. Fluid Mech. 148 (1984) 29–35.
[42] M. Sussman, P. Smereka, Axisymmetric free boundary problems, J. Fluid Mech. 341 (1997) 269–294.
[43] 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) 423–456.
[44] M. van Sint Annaland, N.G. Deen, J.A.M. Kuipers, Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method, Chem. Eng. Sci. 60 (2005) 2999–3011.
[45] H. Mo, C. Zhou, W. Yu, A new method to determine interfacial tension from the retraction of ellipsoidal drops, J. Non-Newtonian Fluid Mech. 91 (2000) 221–232.
[46] Y. Son, K.B. Migler, Interfacial tension measurement between immiscible polymers: Improved deformed drop retraction method, Polymer 43 (2002) 3001–3006.
[47] S. Velankar, H. Zhou, H.K. Jeon, C.W. Macosko, CFD evaluation of drop retraction methods for the measurement of interfacial tension of surfactant-laden drops, J. Colloid Interface Sci. 272 (2004) 172–185.
[48] P.L. Maffettone, M. Minale, Equation of change for ellipsoidal drops in viscous flow, J. Non-Newtonian Fluid Mech. 78 (1998) 227–241.