Insights into sub-Kolmogorov-scale droplet breakup via boundary element simulations

Nicholas Morse[a]

[a]*Research Center Pharmaceutical Engineering GmbH, Inffeldgasse 13, 8010 Graz, Austria*

Abstract

The statistics of sub-Kolmogorov-scale droplet breakup are investigated at a higher Taylor Reynolds number (Re_{λ_T}) than similar work at low Re_{λ_T} (Cristini et al., J. Fluid Mech., vol. 492, 2003) to elucidate intermittency and neck pinch off behaviors. To this end, a boundary element method (BEM) is developed to enable simulations of the stochastic Stokes flow about ensembles of droplets along individual trajectories in homogeneous and isotropic turbulence (HIT), made possible through adaptive mesh refinement and fast multipole acceleration. Droplet deformation statistics, near-breakup behavior, and neck thinning statistics at $Re_{\lambda_T} \simeq 310$ are presented. Results highlight the effect of the HIT's strain rate intermittency on the droplet deformation statistics, and investigations of subcritical and critical neck thinning events provide insights into the critical disturbances leading to breakup and transition of the neck contraction to the established viscous pinch-off regime. Both the marked influence of intermittency on the droplet deformation statistics and local strain rate effects very close into the terminal pinch-off regime emphasize the multiscale nature of the problem, even for droplets in the idealized Stokes regime.

Keywords: droplet breakup, boundary element method, intermittency

---

1. Introduction

The deformation and breakup of droplets in turbulent flows is relevant for a variety of engineering applications, ranging from the food and petroleum industry to nanoemulsion/nanoparticle formation in the pharmaceutical industry. Ni [1] gave a detailed review of the deformation and breakup of drops and bubbles in turbulent flows, while Elghobashi [2] provided a review of direct numerical simulation (DNS) applied to such problems. Both reviews classify two regimes for droplet deformation: one in which the droplet diameter, d, is larger than the Kolmogorov dissipation scale of the surrounding turbulence ($d > \eta$), and the other where the diameter is smaller ($d < \eta$). The Kolmogorov length (η) and time (τ_η) scales are

$$\eta = \left(\frac{\nu_c^3}{\langle \varepsilon \rangle} \right)^{1/4} \tag{1}$$

$$\tau_\eta = \dot{\gamma}_\eta^{-1} = \left(\frac{\nu_c}{\langle \varepsilon \rangle} \right)^{1/2} \tag{2}$$

where $\nu_c = \mu_c/\rho_c$ is the kinematic viscosity of the carrier fluid, $\langle \varepsilon \rangle = \nu_c \langle \boldsymbol{\nabla u} : \boldsymbol{\nabla u} \rangle$ is the rate of turbulent energy dissipation, and the sub-Kolmogorov shear rate ($\dot{\gamma}_\eta$) is the inverse of τ_η. Note that the Reynolds number based on the Kolmogorov scales is unity.

For $d > \eta$, inertial forces dominate the droplet deformation, following the Kolmogorov-Hinze framework [3, 4]. The droplet breakup is therefore governed by the Weber number, $We = \rho_c C_2 (\langle \varepsilon \rangle d)^{2/3} d/\sigma$, where σ is the surface tension and C_2 is the Kolmogorov constant for homogeneous isotropic turbulence (HIT) [1]. A wealth of recent work has leveraged numerical experiments to answer previously unsolved questions about droplets near the Kolmogorov-Hinze (K-H) scale ($d_{KH} \sim \rho_c^{-3/5} \sigma^{3/5} \langle \varepsilon \rangle^{-2/5}$) and their connection with extreme local dissipation events by performing direct numerical simulation (DNS) of HIT laden with droplets [5, 6, 7, 8, 9]. Vela-Martin and Avila [5] performed ensembles of droplet breakup simulations for $d \gg \eta$ in resolved HIT and demonstrated a non-vanishing breakup rate below the K-H scale, which may be difficult to capture experimentally or computationally because of the long timescales required to observe droplets formed at low breakup rates. They postulated that breakup below the K-H scale is due to the small-scale extreme events induced by the surrounding turbulence intermittency. On the other hand, Crialesi-Esposito et al. [8] suggested that the formation of sub-K-H droplets is due to capillary effects when large droplets rupture, and the intermittent high local dissipation is linked to capillary vorticity production during breakup, rather than the other way around. This conclusion was established by way of a numerical experiment in which high vorticity regions were selectively penalized in the dispersed and carrier fluids. Several publications [6, 7, 9] have demonstrated that the dispersed phase affects the small-scale turbulent fluctuations, and recently Saeedipour and Schneiderbauer [9] revealed a strong link between the enstrophy generation due to vortex stretching and that due to surface tension, which may serve as a physical basis for the K-H scale.

As opposed to these studies, the present work focuses on the viscous-dominated regime ($d < \eta$), which is relevant to applications such as nanoemulsion formation, aerosols, and atmospheric clouds [1, 10]. Both Boxall et al. [11] and Gupta et al. [12] have produced experimental evidence for sub-Kolmogorov droplet scaling in emulsion formation in viscous carrier fluids. For $d < \eta$, the deformation of the droplet is driven by viscous forces, and the relevant non-dimensional parameter is the capillary number: $Ca = \sqrt{\mu_c \rho_c \langle \varepsilon \rangle} a/\sigma = \mu_c a \dot{\gamma}_\eta/\sigma$, commonly defined using the undeformed droplet radius, $a = d/2$. Cristini et al. [13] estimated the droplet radius at which viscous breakup was likely to scale as $a/\eta \sim \mu_c^{-5/4} \sigma \rho_c^{1/4} \langle \varepsilon \rangle^{-1/4}$, suggesting that a combination of low surface tension, high carrier fluid viscosity, and significant energy input is required to achieve sub-Kolmogorov droplet breakup. As a result, in many cases with low viscosity carrier fluids and high surface tension (e.g. water droplets in air), the resulting Ca is low enough that the deformation of sub-Kolmogorov droplets can be ignored, and simulations often assume a non-deformed droplet surface [2].

In cases where the droplet deformation is significant, computational approaches can be divided into two categories. For relatively small droplet deformations ($Ca \approx 0.1$), numerous computational works [14, 15, 16, 17] have applied the Maffettone-Minale (MM) model for droplet deformation, which assumes that the droplet shape can be described by the evolution of a triaxial ellipsoid [18], and there have even been efforts to predict viscous-range velocity gradients from large-eddy simulations as input to the MM model [19]. The original MM model coefficients were determined based on perturbation theory at small deformations ($Ca \ll 1$) and therefore lose accuracy for $Ca \gtrsim 0.1$, although extensions of the MM model parameters can extend the model into the finite Ca regime [20].

Despite these model limitations, relatively few studies have numerically resolved the deformation of sub-Kolmogorov droplets at sufficient Ca to result in breakup. Cristini et al. [13] were the first to conduct such a study by recording strain rates of massless tracer particles in pseudo-spectral DNS of HIT and calculating the resulting droplet deformation history using boundary element method (BEM) simulations. The key assumptions enabling this one-way coupling of the surrounding turbulent flow to the droplet deformation were: (i) the droplets were in a dilute suspension, (ii) a similar density of the dispersed and continuous phases ($\rho_d \approx \rho_c$), and (iii) the droplet size being much smaller than the Kolmogorov scale ($d \ll \eta$). The result is that both the droplet Reynolds number and Stokes number are much smaller than unity: $Re_d = d^2/(\tau_\eta \nu_c) = (d/\eta)^2 \ll 1$ and $St_d = \rho_d d^2/(18\mu_c\tau_\eta) = (\rho_d/\rho_c)(Re_d/18) \ll 1$. These assumptions enable the assumption of Stokes flow around the droplet, which is subject to a time-varying linear velocity gradient tensor imposed by the surrounding HIT. Cristini et al. [13] considered a Taylor Reynolds number of $Re_{\lambda_T} = 54$ and focused on a drop-to-carrier fluid viscosity ratio of $\lambda = \mu_d/\mu_c = 1$, although there was a limited discussion of the cases $\lambda = 0.1$ and 5. They reached several conclusions through a statistical analysis of droplet breakup events. First, they demonstrated that droplet deformation, rather than rotation, is the dominant contribution to droplet reorientation (aside from extreme deformations prior to breakup). Secondly, they showed from isolated trajectories that droplet breakup is dependent on the local flow history, rather than the instantaneous strain rate alone, since the strain history determines if elongated droplets will form a critical neck radius to induce pinching or relax back to a spherical shape. Finally, they pointed to the role of rare (intermittent) events as a driver for the fluctuations required for droplet breakup, and point to the extension of their study to higher Re_{λ_T}.

More recent studies [21, 22] have returned to the topic of resolving sub-Kolmogorov droplet deformation using lattice Boltzmann method (LBM) simulations. Milan et al. [21] approached the problem using LBM coupled with a diffuse interface model, where the droplet is immersed in a three-dimensional grid. Through a set of simulations at varying capillary numbers, Milan et al. [21] demonstrated a lower critical capillary number than predicted by the MM model. Taglienti et al. [22] recently developed an alternative LBM for the same problem by coupling LBM with an immersed boundary method (IBM) for the droplet interface. They considered trajectories taken from the publically-available *TURB-Lagr* database, which contains histories of particle position, velocity, acceleration, and flow velocity gradients from an ensemble of 327680 tracers advected for a total time $T \simeq 195\tau_\eta$ in pseudo-spectral DNS of HIT at $Re_{\lambda_T} \simeq 310$ [23]. Through an ensemble of simulations at $\lambda = 1$ and $Ca = 0.05$–0.25 at Reynolds numbers from the Stokes regime to above unity, they compared statistics of the maximum elongation eigenvalue and orientation parameter to the linear theory of Rallison and Acrivos [24] and Rallison [25]. However, they did not consider large enough Ca to result in breakup events.

In the present study, we seek to expand the insights of Cristini et al. [13] on sub-Kolmogorov droplet deformation by considering droplet breakup and neck-thinning behavior at a larger Taylor Reynolds number than the low Re_{λ_T} regime [26, 27] considered by Cristini et al. [13]. A study of droplet deformation at this higher Re_{λ_T} is of interest since the viscous stresses acting on the droplet are directly coupled to fluctuations in the instantaneous dissipation rate and to turbulent intermittency. Intermittency in the local dissipation rate also raises the possibility of droplet breakup during extreme events in regimes where Ca based on $\langle \varepsilon \rangle$ (rather than $\varepsilon(t)$) would not suggest breakup, as suggested by work on larger K-H droplets [5, 9]. Notably, existing scaling laws for sub-Kolmogorov droplets are derived from Kolmogorov scaling based on $\langle \varepsilon \rangle$ [28, 11, 12], without explicitly considering the role of intermittency. These factors motivate the present study, in which the deformation, necking, and breakup of sub-Kolmogorov droplets in HIT are explored using fully interface-resolved simulations in the higher Re_{λ_T} regime.

In contrast to fully resolved two-phase DNS studies, where both the turbulent flow and droplet interfaces are simultaneously resolved [5, 6, 7, 8, 9], the present study follows a one-way coupled approach, as in Cristini et al. [13] and Taglienti et al. [20]. Specifically, we make use of the *TURB-Lagr* database of time-resolved Lagrangian velocity gradient tensor histories along Langrangian tracer trajectories from DNS of HIT at $Re_{\lambda_T} \simeq 310$ [14]. On the droplet scale, we impose these velocity gradient tensor histories as prescribed, time-varying linear flows to individual sub-Kolmogorov droplets, which are numerically resolved using an in-house BEM. Considering an ensemble of droplets in HIT at $\lambda = 1$ and $Ca = 0.3$, we present statistics of droplet deformation and strain rates, with a focus on intermittent events near breakup and statistics of both subcritical and critical droplet neck formation.

The remainder of the paper is organized as follows: first, in section 2, the simulation framework is outlined, along with a brief description of how the *TURB-Lagr* database is used. In §3, we outline the BEM along with its numerical implementation. This is followed by a discussion of the AMR scheme for the droplet surface mesh in §4. The results for ensembles of droplets in HIT are presented in §5, after which conclusions are summarized in §6 to conclude the paper.

2. Simulation framework and use of the *TURB-Lagr* HIT database

The present study makes use of the publicly-available *TURB-Lagr* database [23], which contains an ensemble of Lagrangian tracer data obtained from DNS of incompressible HIT at $Re_{\lambda_T} \simeq 310$. These simulations were performed using a fully dealiased pseudo-spectral method in a triply periodic cubic domain with 1024^3 grid points. The database includes the trajectories of 327680 massless tracer particles randomly placed throughout the domain, each tracked for a duration of approximately $195\tau_\eta$. For each tracer, histories of the tracer position, velocity, acceleration, and velocity gradient tensor of the surrounding flow are recorded with a time resolution of approximately $0.1\tau_\eta$. For further details, the reader is referred to Biferale et al. [23].

For the purposes of the present study, we use only the velocity gradient tensor extracted from the *TURB-Lagr* database to prescribe the imposed time-varying background flow acting on the droplet. Specifically, the droplet experiences a time-varying linear velocity field specified by

$$\boldsymbol{u}^\infty(\boldsymbol{x}, t) = \boldsymbol{E}^\infty(t) \cdot \boldsymbol{x}(t), \tag{3}$$

where $\boldsymbol{E}^\infty(t)$ is the velocity gradient tensor from the HIT database at time t, and $\boldsymbol{x}(t)$ describes the position of the droplet surface relative to the droplet's surface centroid, which is fixed at the origin in the frame of reference of the droplet. A depiction of this simulation framework is shown in figure 1. In this one-way coupled approach, we investigate the response of the droplets to the imposed strain and rotation rate tensors from individual HIT trajectories. In the following sections, the numerical method for the droplet-scale BEM simulations is described.

3. Numerical method

We now detail the governing equations and boundary conditions governing flow on the droplet scale, basics of the BEM, and our numerical implementation.

3.1. Governing equations and boundary conditions

On the droplet scale ($a \ll \eta$), the flow both in and outside the droplet is governed by the Stokes equations:

$$\boldsymbol{\nabla} \cdot \boldsymbol{\Pi} = 0, \quad \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{4}$$

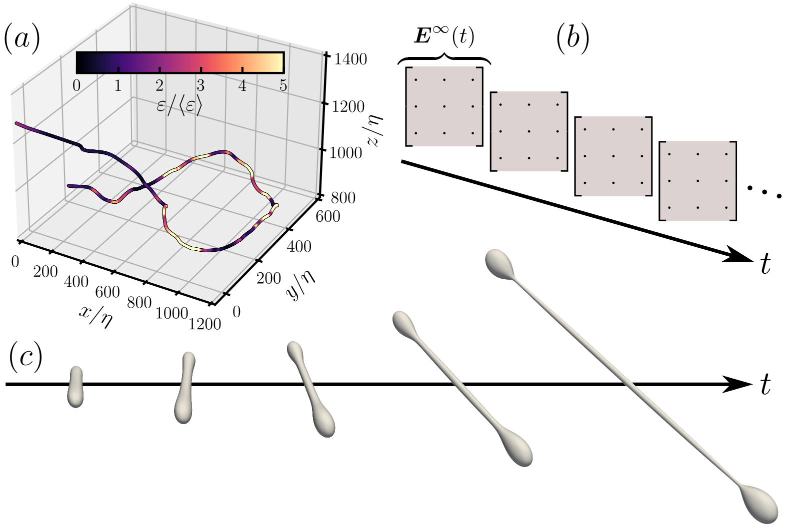


Figure 1: Depiction of the BEM simulation framework. A sample trajectory from the *TURB-Lagr* database Biferale et al. [23], colored by the instantaneous dissipation rate (a). Unsteady velocity gradient tensor history, $\boldsymbol{E}^{\infty}(t)$, along the trajectory from the database (b). A time history of the components $E_{ij}^{\infty}(t)$ for a sample trajectory is shown later in figure 5(a). BEM simulation of droplet deformation subject to $\boldsymbol{E}^{\infty}(t)$ along a sample trajectory (c).

where the stress tensor is defined as

$$\boldsymbol{\Pi} = -p\boldsymbol{I} + \mu\left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^{\top}\right), \tag{5}$$

with the dynamic viscosity taking the value μ_d within the droplet and μ_c outside the droplet.

For a drop immersed in a fluid with an imposed ambient velocity at infinity of $\boldsymbol{u}^{\infty}$, the far-field boundary condition can be written as $\boldsymbol{u} \to \boldsymbol{u}^{\infty}$ as $\|\boldsymbol{x}\| \to \infty$. In the present work, we consider droplets with their surface centroid fixed at the origin, such that $\boldsymbol{x}$ represents the vector from the surface centroid to a point on the droplet surface, and $\boldsymbol{u}^{\infty}(\boldsymbol{x}, t)$ is interpolated from the HIT database, as described in §2. On the droplet surface S, the flow velocity is continuous between the droplet and carrier fluids, such that the boundary condition reads as $\boldsymbol{u}^{(c)} = \boldsymbol{u}^{(d)}$ for $x \in S$. Finally, the stress balance equation to counteract the discontinuity in the interfacial tension force ($\Delta\boldsymbol{f}$) becomes

$$\Delta\boldsymbol{f} = \boldsymbol{f}^{(c)} - \boldsymbol{f}^{(d)} = \left(\boldsymbol{\Pi}^{(c)} - \boldsymbol{\Pi}^{(d)}\right) \cdot \boldsymbol{n} = \sigma \underbrace{(\boldsymbol{\nabla}_s \cdot \boldsymbol{n})}_{-2H} \boldsymbol{n} - \boldsymbol{\nabla}_s \sigma \tag{6}$$

where σ is the surface tension, H is the mean curvature of the surface, and the surface tangential gradient operator is defined by $\boldsymbol{\nabla}_s = (\boldsymbol{I} - \boldsymbol{nn}) \cdot \boldsymbol{\nabla} = \boldsymbol{\nabla} - \boldsymbol{n}\frac{\partial}{\partial n}$. For the present case, we do not consider surfactant effects, resulting in $\Delta\boldsymbol{f} = \Delta f^{\perp}\boldsymbol{n} = -2H\sigma\boldsymbol{n}$.

3.2. Boundary element method

The original BEM for Stokes flow for $\lambda \neq 0$ was derived by Rallison and Acrivos [24], and the method has been subsequently used to study droplet breakup in simple flows [29, 30], Marangoni effects due to surfactants [31, 32, 33, 34, 35], highly concentrated emulsions [36, 37, 38, 39, 40], and droplet flow through constrictions or packed beds [41, 42, 43]; a comprehensive overview of BEM theory, numerics, and applications is given by Pozrikidis [44]. The essence of the BEM is to reformulate the Stokes flow problem as boundary integrals on the fluid boundaries, expressed in terms of the Green's functions (the Stokeslet and Stresslet), which are given by

$$G_{ij}(\boldsymbol{r}) = \frac{\delta_{ij}}{\|\boldsymbol{r}\|} + \frac{r_i r_j}{\|\boldsymbol{r}\|^3}, \quad T_{ijk}(\boldsymbol{r}) = -6\frac{r_i r_j r_k}{\|\boldsymbol{r}\|^5}, \tag{7}$$

where $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}_0$ in which $\boldsymbol{x}$ is the observation (field) point, and $\boldsymbol{x}_0$ is the source point (pole). By employing the Lorentz reciprocal identity and taking $\boldsymbol{x}, \boldsymbol{x}_0 \in S$, Pozrikidis [44] details how the derivation of Rallison and Acrivos [24] produces a Fredholm integral equation of the second kind for the interfacial velocity $\boldsymbol{u}$:

$$\begin{aligned} u_j(\boldsymbol{x}) = &\frac{2}{\lambda+1}\left[u_j^{\infty}(\boldsymbol{x}) - \frac{1}{8\pi\mu_c}\int_S \Delta f_i(\boldsymbol{x}_0) G_{ij}(\boldsymbol{x}-\boldsymbol{x}_0) dS(\boldsymbol{x}_0)\right] \\ &+\frac{\lambda-1}{\lambda+1}\frac{1}{4\pi}\int_S u_i(\boldsymbol{x}_0) T_{ijk}(\boldsymbol{x}-\boldsymbol{x}_0) n_k(\boldsymbol{x}_0) dS(\boldsymbol{x}_0). \end{aligned} \tag{8}$$

The formulation of equation 8 in terms of the Stokeslet and Stresslet allows the flow around the droplet to be expressed only in terms of the boundary values on the droplet interface. Numerically, this enables the Stokes flow equations to be solved using a discretization of the droplet surface rather than the three-dimensional fluid domain as in the LBM [21, 22]. For $\lambda = 1$, equation 8 reduces to a particularly simple form through elimination of the second surface integral, named the double layer potential (DLP). This leaves only the single layer potential (SLP) integral and the applied external flow. For any value of λ, equation 8 can be solved for the interfacial velocity $\boldsymbol{u}(\boldsymbol{x})$ to evolve the shape of the droplet.

3.3. Numerical implementation

In discrete form, the surface integrals in equation 8 are discretized into surface elements and associated quadrature points as

$$\begin{aligned} u_j(\boldsymbol{x}) = &\frac{2}{\lambda+1}\left[u_j^{\infty}(\boldsymbol{x}) - \frac{1}{8\pi\mu_c}\sum_{ielm=1}^{N_\Delta}\sum_{qpt=1}^{K_\Delta}\left[\Delta f^{\perp}(\boldsymbol{x}_0) - \Delta f^{\perp}(\boldsymbol{x})\right] n_i(\boldsymbol{x}_0) G_{ij}(\boldsymbol{x}-\boldsymbol{x}_0) dS(\boldsymbol{x}_0)\right] \\ &+\frac{\lambda-1}{\lambda+1}\left[\frac{1}{4\pi}\sum_{ielm=1}^{N_\Delta}\sum_{qpt=1}^{K_\Delta}\left[u_i(\boldsymbol{x}_0) - u_i(\boldsymbol{x})\right] T_{ijk}(\boldsymbol{x}-\boldsymbol{x}_0) n_k(\boldsymbol{x}_0) dS(\boldsymbol{x}_0) + u_j(\boldsymbol{x})\right]. \end{aligned} \tag{9}$$

where we have made use of the singularity subtractions definitions of Pozrikidis [44] to analytically eliminate the singularities in $G_{ij}(\boldsymbol{r})$ and $T_{ijk}(\boldsymbol{r})$ as $\boldsymbol{r} \to 0$. In the present method, we discretize the droplet surface using an unstructured surface grid of curved six-node Lagrange triangular elements, which is dynamically adjusted in time following an AMR scheme (§4) to resolve the deformation of the droplet surface. The integrals in equation 8 are performed using a three-point Gauss-Legendre quadrature [45] where the values at the element nodes are interpolated on the element using the Lagrange basis functions. In contrast to the method outlined by Pozrikidis [44], we do not modify the shape functions to account for the displacement of edge nodes away from their ideal positions at the centerpoints between corner nodes; instead, we regularize their positions on each time step through a mesh smoothing scheme (§4.1.2). The surface normal vector ($\boldsymbol{n}$) and mean and Gaussian curvatures (H and K) on the element, which are critical for the accuracy of the BEM, are evaluated using identities based on the fundamental forms of the curved element surfaces. Appendix A provides the details of the discretization, along with the order of accuracy of the present method compared to existing methods.

To update the droplet shape using the velocities calculated from equation 8, the positions of the surface nodes on the droplet surface are advanced in time with a second-order Runge-Kutta (RK2) method applying equation 9 evaluated at the nodes. To limit the cost of the simulations, the time step is dynamically adjusted according to the formula suggested by Zinchenko and Davis [41], which reads as

$$\Delta t = F \frac{\mu_c}{\sigma d} \min_{ipt=1,\ldots,N_p} \left[\left(\kappa_{\max}^{(ipt)} \right)^{-1} \min_{jpt=1,\ldots,N_{nbrs}^{(ipt)}} \left(\| \boldsymbol{x}^{(ipt)} - \boldsymbol{x}^{(jpt)} \| \right) \right], \tag{10}$$

where $\min\left(\|\boldsymbol{x} - \boldsymbol{x}_{nbr}\|\right)$ is the minimum distance between a given node (ipt) and any of its neighbors (jpt), $\kappa_{\max} = \max\left(\kappa_{1,2}\right) = \max\left(H \pm \sqrt{H^2 - K}\right)$ is the maximum of the principal surface curvatures, and F is a numerical factor of order one. This allows the time step to adjust dynamically as the droplet deforms and the mesh elements become finer. After the node positions have been advanced on each time step, the surface grid is adjusted according to the algorithm outlined in §4, and the droplet volume and centroid are rescaled (by $\lesssim 10^{-5}$) to preserve their initial values.

3.3.1. Treatment of non-unity viscosity ratios

For non-unity viscosity ratios ($\lambda \neq 1$), equation 9 results in a dense linear system for $\boldsymbol{u}(\boldsymbol{x})$, which appears on the right-hand side in the DLP integral. We solve this matrix equation using a matrix-free restarted generalized minimal residual method (GMRES(k)) algorithm [46], which Zinchenko and Davis [41] found effective for the Stokes flow BEM. The matrix-free GMRES(k) formulation is especially valuable since it avoids the memory-prohibitive storage of the fully-dense matrix associated with the DLP integral. To improve the convergence of the iterative solver, we also use Wielandt eigenvalue deflation to eliminate the $\lambda \to 0$ and $\lambda \to \infty$ solutions from the spectrum of the integral operator [44, 47].

3.4. Fast multipole method acceleration

The update of the droplet surface velocities using equation 9 requires $O(N_\Delta^2)$ work due to the SLP and DLP integrals of the Stokeslet and Stresslet Green's functions across every source point ($\boldsymbol{x}_0$) for each target ($\boldsymbol{x}$). For $\lambda = 1$, each RK2 time step requires two of these evaluations, while the cost is even higher for $\lambda \neq 1$ due to the dense matrix-vector multiplication (DLP integration) at each GMRES(k) iteration. This $O(N_\Delta^2)$ cost quickly becomes prohibitively expensive as the grid size becomes large.

The fast multipole method (FMM) was introduced by Greengard and Rokhlin [48] for rapidly evaluating pairwise potentials between M source points and N target points, which traditionally requires $O(NM)$ work. The FMM reduces the cost to $O((N + M)\log^{3/2}(1/\epsilon_{\text{tol}}))$ for a given tolerance ϵ_{tol} by expanding the Green's functions using a multipole expansion to compress groups of sources across a hierarchy of length scales. We make use of the Laplace kernels in the Flatiron Institute's *FMM3D* software library [49] to evaluate the singularity-subtracted sums in equation 9.

Figure 2(a) shows the timings for the direct calculation and FMM evaluation (at $\epsilon_{\text{tol}} = 10^{-5}$) of a single SLP and DLP integral evaluations across a range of grid sizes, while figure 2(b) shows the relative speed-up offered by the FMM. The directly-evaluated SLP and DLP integrals obey the expected $O(N_\Delta^2)$ scaling across grid sizes, with a consistently faster DLP evaluation due to the lower number of floating-point operations required to calculate the integrand compared to the SLP. The FMM integral evaluations obey an $O(N_\Delta^1)$ scaling for $N_\Delta \gtrsim 1000$, with the nonlinear scaling below this grid size being due to the overhead cost of setting up the FMM data structures. As shown in figure 2(b), the FMM outperforms direct evaluation of the SLP and DLP integrals for grid sizes $N_\Delta \gtrsim 300$. To exploit this crossover, the BEM switches to the FMM for $N_\Delta > 300$, though simulation values typically exceed this threshold.

The FMM DLP evaluation is significantly more expensive than the FMM SLP due to the added cost of computing the Hessian, which reduces the relative speed-up of the FMM DLP. This poses a challenge, as the DLP integral must be evaluated in every GMRES(k) iteration. However, since the Hessian contributions are independent of the solution variable $\boldsymbol{u}(\boldsymbol{x})$, we

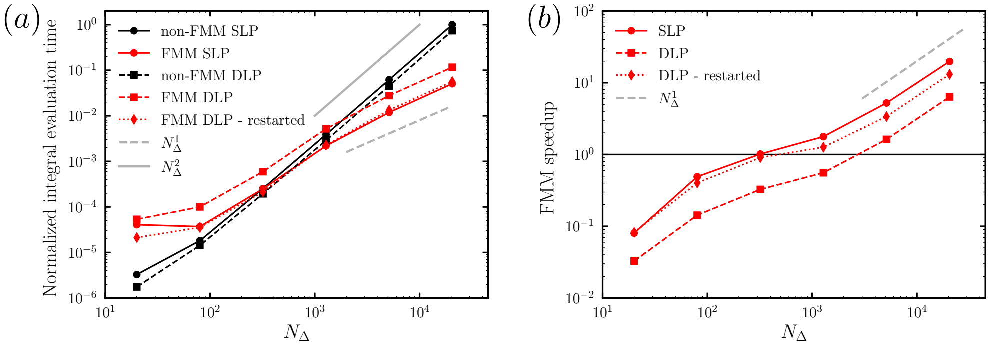


Figure 2: Comparison between non-FMM and FMM evaluations of the SLP and DLP integrals. Panel (*a*) shows the evaluation time for both integrals for each method, along with linear and quadratic scaling lines. In panel (*b*), the speed-up of the FMM over the standard non-FMM integral evaluation is shown.

compute them only during the first iteration. In subsequent iterations, only potential and gradient evaluations are required at the targets, with fewer sources, reducing the cost to a level comparable to the FMM SLP evaluation (see dotted red lines in figure 2).

4. Adaptive mesh refinement algorithm

Aside from the cost of evaluating the surface integrals, a primary challenge of using the BEM to track droplet deformation is the requirement that the surface grid deform to track the evolving shape of the droplet interface. This is especially important during pinch-off events as the droplets break up, during which the local radius of the droplet tends towards zero. To resolve the changing geometry of the interface, we employ an AMR algorithm similar in concept to that of Cristini et al. [50], although the present method uses some alternative techniques from graphics/computer aided design software and is the first to our knowledge to be applied to Lagrange triangle elements. An overview of the present AMR method is given here, with details of each algorithm left to Appendix B.

The current algorithm generally follows that outlined by Botsch et al. [51], as depicted in algorithm 1, although some modifications have been made for the six-node elements and overall robustness. The AMR algorithm is performed after the update of the node positions on each time step, and consists of smoothing, refinement, coarsening, and cell quality operations.

Algorithm 1 Overview of the AMR algorithm.

```
Calculate $L_\Delta$ at each point                ▷ Equation 11
Laplacian mesh smoothing                          ▷ Algorithm 2
for $iter = 1$ to $N_{max}$ do
    Refine grid                                   ▷ Algorithm 4
    Coarsen grid                                  ▷ Algorithm 5
    Flip edges                                    ▷ Algorithm 6
    if no topology changes then
        return
    end if
    Laplacian mesh smoothing                      ▷ Algorithm 2
end for
```

These operations are based on a target length scale L_Δ defined based on the formula of Cristini et al. [50] as

$$L_\Delta = R_\Delta \sqrt{\frac{2}{\kappa_1^2 + \kappa_2^2}} = R_\Delta (2H^2 + K)^{-1/2}, \tag{11}$$

where R_Δ is the desired grid resolution in radians based on the above estimate of the local surface radius. This definition of L_Δ ensures that the grid resolution adapts to the local curvatures of the surface. After L_Δ is calculated at each point, the values are limited such that the maximum growth rate does not exceed 1.3.

4.1. Smoothing operations

After calculating L_Δ, the first step in algorithm 1 is to apply smoothing to the surface grid to maintain the cell quality of nearly equilateral elements. This consists of Laplacian mesh smoothing and edge node adjustment steps, detailed as follows.

4.1.1. Laplacian mesh smoothing algorithm

The Laplacian smoothing algorithm (algorithm 2) repositions the corner nodes of the elements using weights proportional to the neighbor element areas and target length scales, similar to Dunyach et al. [52]. This weighting is used to construct the coordinates of a target point using the adjacent elements using

$$\boldsymbol{x}_t = \frac{\sum\limits_{jelm \in \Delta_{adj}} w^{(jelm)} \boldsymbol{b}^{(jelm)}}{\sum\limits_{jelm \in \Delta_{adj}} w^{(jelm)}}, \tag{12}$$

where $\boldsymbol{b}^{(jelm)}$ are the barycenters of the adjacent elements and the weight for each element is chosen as

$$w^{(jelm)} = S_{\Delta}^{(jelm)} \left(\frac{1}{6} \sum_{jpt=1}^{6} L_{\Delta}^{(jelm,jpt)} \right)^{-2}, \tag{13}$$

where $S_{\Delta}^{(jelm)}$ and $jpt = 1, \ldots, 6$ are the area and nodes of element $jelm$. The choice of this weight was found to reduce the required iterations in algorithm 1 compared to that suggested by Dunyach et al. [52] by properly retaining refined elements in regions of high local curvature and avoiding the diffusive 'flow' of points into coarser regions.

After calculating the target point in equation 12, $\boldsymbol{x}_t$ is projected onto the tangent plane using the coordinates and normal ($\boldsymbol{x}$ and $\boldsymbol{n}$) of the original node as $\boldsymbol{x}_t^{\perp} = \boldsymbol{x}_t + \boldsymbol{n}\,[\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{x_t})]$. Then the original point is updated using $\boldsymbol{x}_{new} = (1 - \zeta_s)\boldsymbol{x} + \zeta_s \boldsymbol{x}_t^{\perp}$ with a smoothing parameter $0 < \zeta_s \leq 1$. This smoothing operation is performed for up to 10 iterations, or until the maximum point displacement falls below a given threshold. After this point, the updated corner nodes are projected back onto the existing grid surface, where the other solution variables are interpolated to the new point. This ensures that the mesh smoothing operation preserves the shape of the droplet surface, as opposed to most (non-feature-preserving) Laplacian mesh smoothing algorithms, which shrink or lose details of the surface [51].

4.1.2. Edge node adjustment scheme

After the corner nodes have been adjusted using the Laplacian smoothing scheme, the position of the edge nodes needs to be adjusted to lie between the adjacent corner nodes. This is also necessary to preserve the interpolation accuracy of the Lagrange basis functions to avoid the corrective terms suggested by Pozrikidis [44]. The adjustment scheme (algorithm 3) uses a spring-damper analogy, whereby an artificial point slides along the surface of the adjacent elements to minimize the summed distance between the point and the adjacent corner nodes. The effect of the algorithm can be explained physically as a pseudo surface tension between the edge node and the adjacent corner nodes, and was found to significantly improve cell qualities at high droplet deformations.

4.2. Grid topology operations

Following the mesh smoothing, a number of mesh restructuring operations are performed to reach the target cell size distribution and maintain the cell

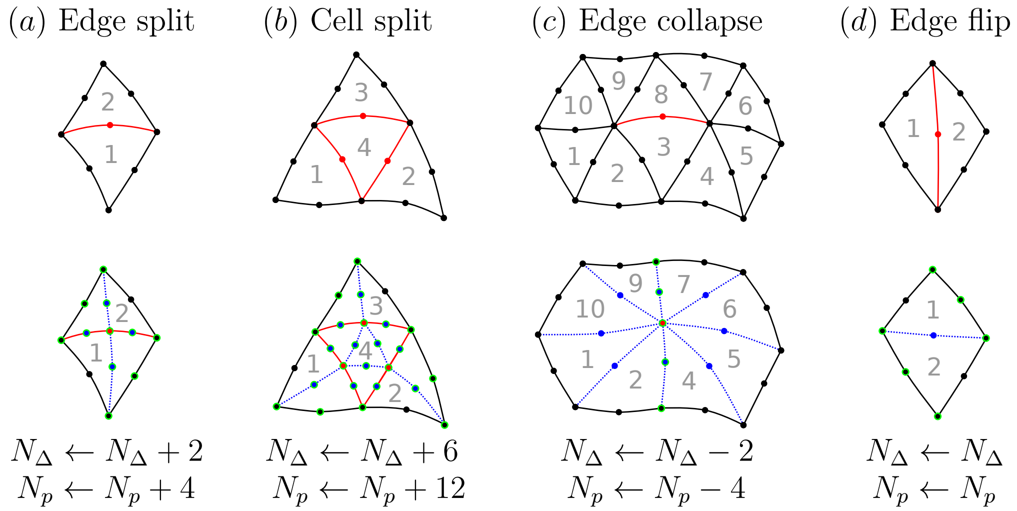


Figure 3: Depiction of the edge split (a), cell split (b), edge collapse (c), and edge flip (d) operations in the AMR algorithm. Points and edges to be operated on are shown in red, new points/edges are shown in blue, and green highlights indicate points requiring new $pt \rightarrow elm$ connectivity. The original element numbers are labelled before and after the grid operations, and the updated numbers of elements (N_Δ) and points (N_p) are shown below each operation.

qualities. Figure 3 depicts the individual topology operations of cell/edge refinement (a, b), coarsening (c), and cell quality adjustments through edge flipping (d). If no topology changes are required, the AMR loop in algorithm 1 is terminated and the method moves to the next physical time step.

4.2.1. Grid refinement algorithm

In the grid refinement step (algorithm 4), the code visits each edge of the mesh and marks the edge if its length exceeds $4L_\Delta/3$, an optimal criterion defined by Botsch et al. [51]. Elements having multiple marked edges are subject to cell splitting (figure 3(b)), while those with only one marked edge are split through that edge (figure 3(a)). The choice to perform cell splitting instead of purely relying on edge splitting [51, 52] improves the grid quality by producing more closely equilateral elements before smoothing. For each newly created point, the surface variables are interpolated and the relevant element and point connectivity must be updated.

4.2.2. Grid coarsening algorithm

It is also important to remove cells in regions of low surface curvature to limit the computational cost of the BEM. In this case, a coarsening step (algorithm 5) is performed on each edge of the grid to check if the edge is shorter than the optimal edge length criterion $4L_\Delta/5$ [51]. If so, the edge is collapsed, as shown in figure 3(*c*), and the associated connectivity is updated. The surrounding edge nodes are then repositioned according to the algorithm described in §4.1.2.

4.2.3. Cell quality adjustment

After the refinement and coarsening steps, the quality of the cells is improved by flipping edges to produce more equilateral elements, as shown in figure 3(*d*). Botsch et al. [51] suggest flipping a given edge if doing so would decrease the deviation of the valences of the surrounding corner nodes from the desired valence (six) of a two-dimensional lattice of equilateral triangles. However, we found that this method occasionally produced undesirable cell qualities and even invalid cells. Instead, we use the Delaunay triangulation method of Cristini et al. [50] (algorithm 6) to flip the cell if the edge-opposite node-to-node distance is smaller than the sum of the projected distances from the edge-opposite nodes to the cell circumcenters. This does not result in a change to the number of elements or points, but the flipped edge node is repositioned, and the local connectivity is updated.

4.3. Leveque deformation test case for the AMR scheme

To evaluate the AMR, we employ the popular droplet deformation test case introduced by Leveque [53]: a sphere of radius $a = 0.15$ at an initial position $(0.35, 0.35, 0.35)$ is advected over a period $T = 3$ by a velocity field

$$\boldsymbol{u} = \begin{bmatrix} 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{bmatrix} \cos\left(\frac{\pi}{T}t\right). \tag{14}$$

The droplet position is advected in time by disabling the evaluation of the BEM equations and directly advecting the droplet surface using equation 14 with RK2 time stepping. The surface of the droplet is tracked with a target grid resolution of $R_\Delta = 0.2$ radians.

Figure 4(*a*) shows the evolution of the surface, demonstrating the adaptive refinement of the highly-curved edges of the deformed droplet and the coarsening as the droplet returns to a sphere. We note by visual inspection that the shape of the droplet properly returns to a sphere after one period. To quantitatively demonstrate the feature-preserving behavior of the AMR algorithm, the post-time step volume renormalization used in the BEM is disabled, and the evolution of the droplet volume is recorded. As shown in figure 4(*b*), the total change in volume over the simulation is less than 0.2% of the initial volume. This justifies the $\lesssim 10^{-5}$ volume renormalization used in the actual droplet simulations.

Figure 4(*c*) shows the evolution of the grid size (N_Δ and N_p) as the mesh adapts and coarsens to resolve the evolving curvature of the surface. Since equation 14 defines a time-reversible velocity field over the period T (as evidenced by the droplet shape in figure 4(*a*)), the grid size should ideally also be symmetric in time. Figure 4(*c*) displays nearly symmetric profiles of N_Δ and N_p about $T/2$, confirming that the refinement and coarsening operations properly compliment each other.

Figure 4(*d*) displays this remeshing time plotted against N_Δ, where the points are colored by the number of iterations of the AMR algorithm. Note that the maximum number of iterations was limited to $N_{iter} = 10$. As demonstrated in the figure, the computational cost of the AMR scales as $O(N_\Delta^1)$, equivalent to that of the FMM-accelerated BEM. Aside from the scaling of the method, the overall cost can be reduced by limiting the number of required AMR iterations to reach an optimal grid. The times in which fewer grid restructuring operations are required are in parts of the deformation period with smaller surface distortions per time step (the peak and trough of the cosine wave, corresponding to the minimum and maximum grid sizes). This hints at the interplay between the choice of time step size and the cost of the AMR. We note that the actual simulations of droplet deformation have much less surface distortion per time step than this test case, and typically the cost of AMR for the simulations is around a quarter of the total simulation time (see §5.1). However, the dependence of the AMR cost on the number of iterations highlights the importance of properly choosing the weighting of the mesh smoothing algorithm, which was selected in equation 13 to reduce the number of topological AMR operations.

4.4. Definition of a droplet breakup criterion

Finally, to enable efficient simulation of droplet dynamics up to the point of breakup, we devise a criterion to determine when the droplets have entered the pinch-off regime such that the simulations can be terminated immediately

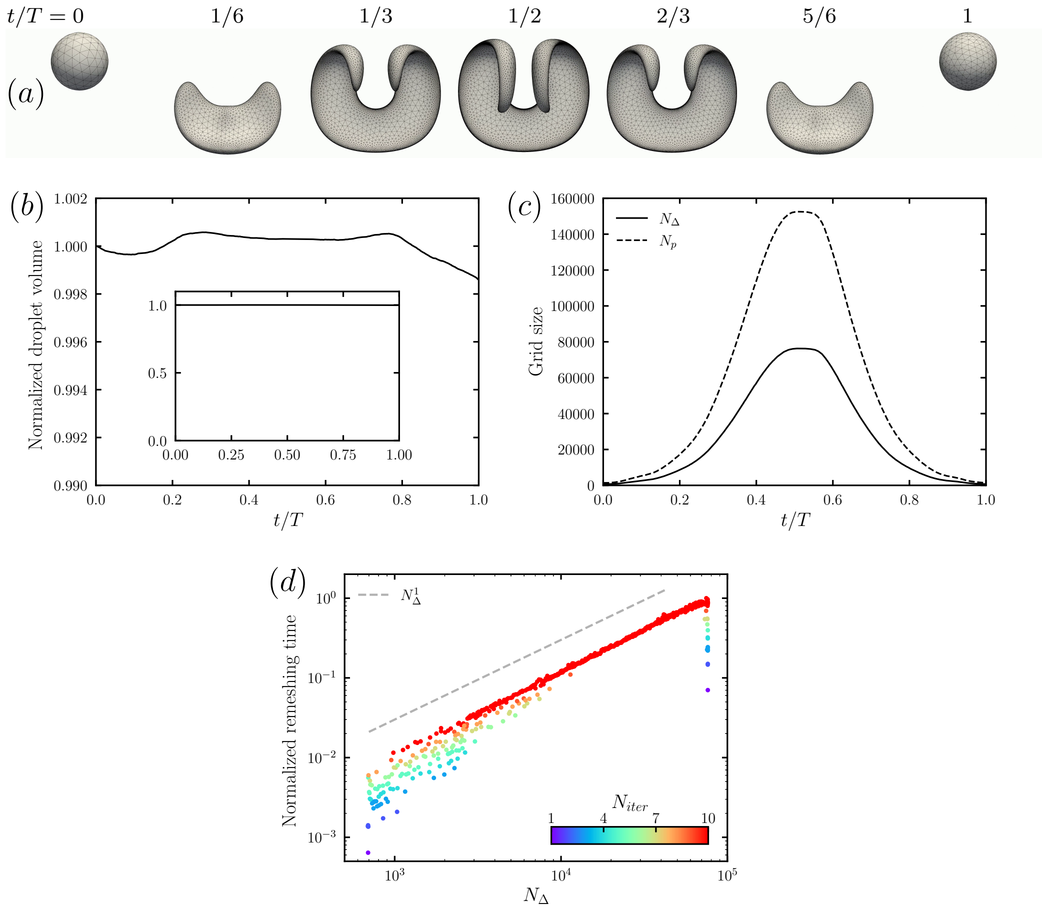


Figure 4: The droplet deformation numerical test case introduced by Leveque [53]: the initially spherical droplet is deformed by the unsteady velocity field given by equation 14, producing the deformation history shown in panel (a). Panel (b) shows the change in the droplet volume over one cycle of deformation, while panel (c) displays the change in the grid size across the deformation in terms of number of elements (N_Δ) and points (N_p). Panel (d) diplays the normalized remeshing time to demonstrate $O(N_\Delta^1)$ cost of the mesh restructuring. Points are colored by the number of remeshing cycles per timestep.

prior to breakup. We choose not to simulate the actual breakup of droplets, as our focus is on the dynamics leading to breakup rather than daughter droplet evolution, and breakup modeling would require complex mesh splitting and surface reconstruction within the BEM framework [50, 42]. In order to detect the onset of breakup in the pinching regime, we employ an algorithm that checks three criteria at each grid point:

1. Whether the minimum local principal curvature of the surface has become concave ($\min(\kappa_{1,2}) < 0$), such that the drop has a neck.
2. Whether the maximum local principal curvature corresponds to a radius below a specified breakup radius threshold: $\max(\kappa_{1,2})^{-1} < r_{breakup} = a/20$ (similar to Zinchenko and Davis [42]). While a general choice for this threshold likely depends on Ca, the present value is justified in §5.7.
3. Whether the local surface-normal velocity points into the drop, similar to the contraction parameter defined by Kim and Moin [54], although we note that the rigid-body component of the droplet surface velocity must be subtracted during this computation, since the droplet may be spinning or translating during breakup. The criterion is therefore expressed as $(\boldsymbol{u} - \langle \boldsymbol{V} \rangle - \langle \boldsymbol{\Omega} \rangle \times \boldsymbol{x}) \cdot \boldsymbol{n} < 0$.

If these three criteria are met at a point, the algorithm goes on to check if these statements hold for all points around the neck circumference by constructing the plane associated with $\max(\kappa_{1,2})$ using the cross product of its principal direction and the surface normal. Each element intersecting this plane is identified, and the above criteria are evaluated for its node closest to the plane.

5. Numerical experiments

Having described the use of the *TURB-Lagr* HIT database [23] and the present BEM algorithm, we move to the physical problem at hand. Namely, we consider the deformation of droplets along the trajectories in HIT ($Re_{\lambda_T} \simeq 310$) at $\lambda = 1$ and a capillary number of

$$Ca = \frac{\mu_c a \dot{\gamma}_\eta}{\sigma} = \frac{\mu_c a}{\sigma \tau_\eta} = 0.3. \tag{15}$$

In the simulations, each droplet is initialized as a sphere of radius a at $t = 0$, which is allowed to resolve in response to $\boldsymbol{E}^\infty(t)$ from the HIT trajectory.

Note that this initial condition results in some transient period after this initial condition during which the droplet is out of equilibrium with the flow.

Before conducting the numerical experiments, we first determine the grid resolution at which the results converge and validate the BEM through comparison to the MM model across a range of λ at lower Ca.

5.1. Grid convergence

To select a grid resolution of suitable accuracy at minimal computational cost, we conduct droplet deformation simulations at grid resolutions from $R_\Delta = 0.1$ to 0.4 along a single trajectory from the *TURB-Lagr* database with a velocity gradient tensor history shown in figure 5(a). The point where droplet breakup is detected is shown in the figure, along with the shape of the droplet at this point. Figure 5(b) shows the time history of the maximum radius of the droplet surface over time, defined as the maximum distance of any surface node from the droplet's surface centroid ($r_{max} = \max(\|\boldsymbol{x}\|)$). The time histories of r_{max} for the different resolutions lie nearly on top of each other. The differences between the simulations immediately before the breakup detection are highlighted in the inset, where a shaded area is plotted to show a ±1 percent band around the result at the finest grid resolution. Clearly the maximum extension of the droplet and the breakup time converge with the grid resolution. Figure 5(c) shows the time history of N_Δ throughout the simulations. Here we remark that $N_\Delta \propto R_\Delta^{-2}$, such that doubling R_Δ quadruples the total computational cost. To limit the cost of the simulations, we select a grid resolution of $R_\Delta = 0.25$, which was matched the results for $R_\Delta = 0.1$ within nearly 1% in figure 5(b).

Finally, we evaluate the relative cost of the AMR operations compared to the cost of the solver in figure 5(d). For the selected R_Δ, the cumulative AMR time falls below 20 percent of the cumulative solver time by the end of the simulations, indicating that the cost of the AMR is relatively small. The only point at which the AMR may take longer than the solver time is at the very beginning of the simulations, which contributes little to the overall cost.

5.2. Comparison to the Maffettone-Minale model

To validate the accuracy of the BEM solution, we compare its results to a linear model of droplet deformation at low Ca, as done by Taglienti et al. [22]. In our case, we compare the history of the maximum droplet surface radius and droplet orientation parameter to the MM model [18] for the

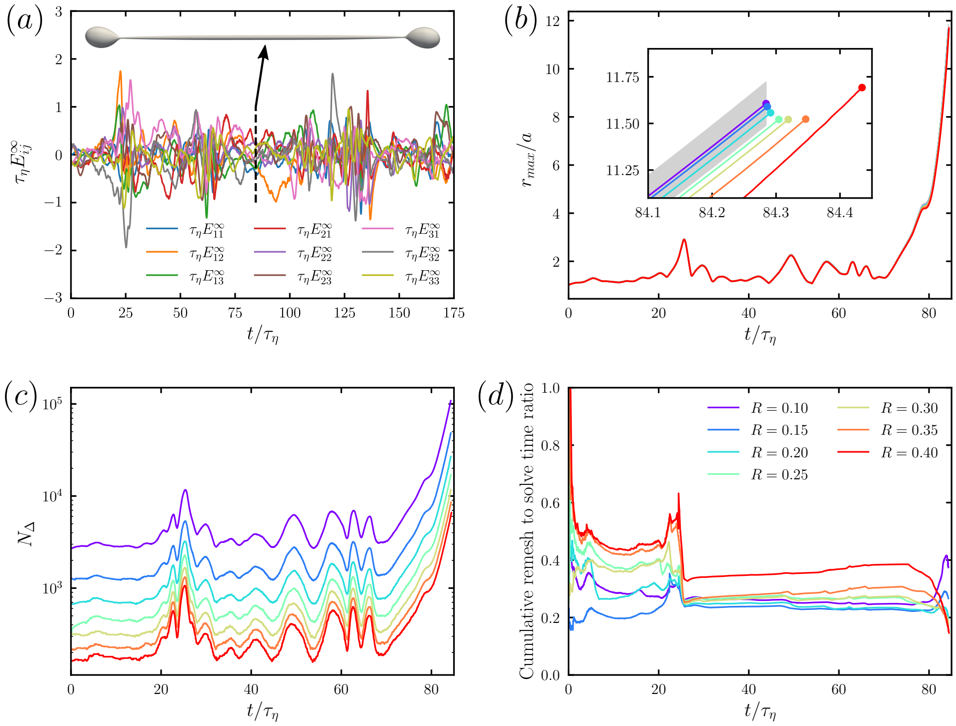


Figure 5: Results of the grid refinement study for resolutions $R_\Delta = 0.1$ to 0.4 (cool to warm colors). The velocity gradient tensor history is shown in panel (a), along with the droplet shape at the point of breakup. The maximum radius of the droplet surface versus time is shown in panel (b), with the inset showing a zoomed view of the point where breakup is detected, where the shaded region represents a ±1 percent range from the finest resolution. Panel (c) shows the number of elements for each case, and the ratio of the cumulative remeshing to solve time is shown in panel (d).

trajectory shown in figure 5(a). The droplet orientation parameter [13, 22] is a measure of the alignment of the droplet shape with the strain rate tensor $\boldsymbol{S}^\infty = \frac{1}{2}\left(\boldsymbol{E}^\infty + (\boldsymbol{E}^\infty)^\top\right)$, and is defined as

$$\beta = \frac{\hat{\boldsymbol{v}}_{max} \cdot \boldsymbol{S}^\infty \cdot \hat{\boldsymbol{v}}_{max}}{\left(\boldsymbol{S}^\infty : \boldsymbol{S}^\infty\right)^{1/2}}, \tag{16}$$

where $\hat{\boldsymbol{v}}_{max}$ is the unit vector in the direction of maximum deformation (r_{max}).

We consider a capillary number of $Ca = 0.05$ and viscosity ratios of $\lambda = 0.1$, 1, and 10 to confirm the accuracy of the SLP and DLP across two orders of magnitude in λ. The capillary number was chosen such that the MM model provides an accurate baseline of comparison. In figure 6, r_{max} and β from the simulations are compared to the MM model for each λ. In each case, the agreement of the simulation results with the MM model is excellent, confirming the accuracy of the BEM for such droplet deformations in turbulent flow histories.

5.3. Droplet statistics

Having established the grid resolution and accuracy of the BEM, simulations of droplet deformation at $Ca = 0.3$ and $\lambda = 1$ are conducted for 191 different trajectories from the *TURB-Lagr* database, of which 96 resulted in breakup. Figure 7 shows cumulative distribution functions (CDFs) of the maximum droplet length ($l = 2r_{\max}$) and orientation parameter across all simulations, compared to the results of Cristini et al. [13] at $Re_{\lambda_T} = 54$. The CDF of droplet length of the present simulations exhibits a similar median value of 2.51 as that of Cristini et al. [13] (2.51), while the median value of β (0.53) is slightly higher in the present ensemble than that from Cristini et al. [13] (0.52). Both variables have a broader distribution with longer tails than those of Cristini et al. [13], indicating an effect related to the increased intermittency of the fluctuating strain rate at higher Re_{λ_T} [26], which is different from the effect of Ca alone. These effects at the same $Ca = 0.3$ of Cristini et al. [13] indicate that the nominal Ca based on $\dot{\gamma}_\eta = \tau_\eta^{-1}$ is insufficient to characterize the droplet deformation behavior across Re_{λ_T}.

To investigate the correlation of l and β, figure 8 presents the joint probability density function (PDF) of these two variables. The logarithmic scale of the contours emphasizes that a vast majority of the droplet lifetime is spent near the median values of l and β. Increasing l is associated with a decrease

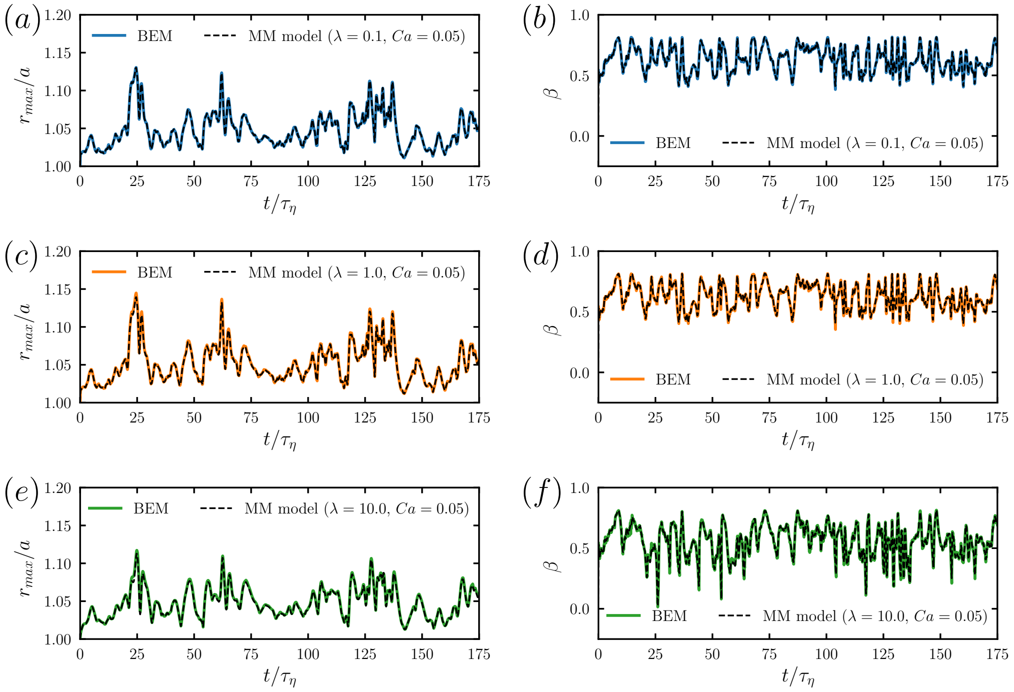


Figure 6: Comparison of the maximum droplet surface radius (a, c, e) and the droplet orientation parameter (b, d, f) from simulations compared to predictions of the MM model for the velocity gradient tensor history shown in figure 5(a) at $Ca = 0.05$. Results are shown for viscosity ratios $\lambda = 0.1$ (a, b), 1 (c, d), and 10 (e, f).

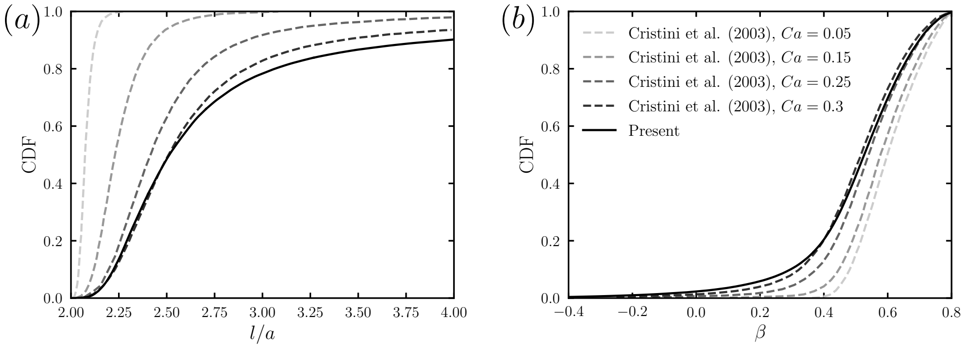


Figure 7: Cumulative distribution functions of the drop length (a) and orientation parameter (b) for the present simulations ($Ca = 0.3$, $Re_{\lambda_T} \simeq 310$) compared to the results of Cristini et al. [13] ($Ca = 0.05$–0.3, $Re_{\lambda_T} = 54$).

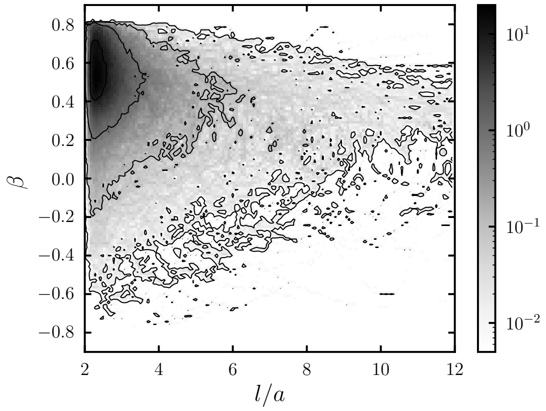


Figure 8: Joint probability density function of l/a and β for the present simulations.

of β, although there are also fewer events where $\beta < 0$. During breakup events, l/a is generally past the horizontal extent of the figure, emphasizing the rarity of such events. To investigate these breakup events in detail, ensemble-averaged statistics near breakup are presented in the next section.

5.4. Near-breakup behavior

We consider the behavior of the droplet close to breakup by examining ensemble-averaged histories of l and β within $50\tau_\eta$ of breakup, which are shown in figure 9(a) and 9(b), respectively. From figure 9(a), we see that the start of the final droplet deformation occurs at approximately $15\tau_\eta$ from the breakup, a similar result to that reported by Cristini et al. [13]. However, the average final droplet length attained at breakup in the present case is around double that of Cristini et al. [13] at $Re_{\lambda_T} = 54$, likely affecting the size of the daughter droplets after breakup. This effect is likely due to the intensity of extreme fluctuating strain rate events at the present $Re_{\lambda_T} \simeq 310$. Figure 9($b$) shows that while the average β at breakup is similar to Cristini et al. [13], the values further away from breakup are lower than those reported in their work, despite the fact that the median β is higher in the present study (figure 7(b)). This difference may be due to the long tail of the β distribution in the current case, or related to a smaller ensemble average in Cristini et al. [13].

Since viscous droplet breakup is a direct consequence of neck formation and pinch-off, we also present ensemble-averaged statistics of the neck radius of the droplet approaching pinch-off in figure 9(c). The neck radius is defined, following the first two criteria in §4.4, as the maximum local principal curvature over all mesh points where the minimum local principal curvature is negative (concave). Note that since a neck is not necessarily present at all times, the number of samples in the $\langle r_{\text{neck}} \rangle$ ensemble decreases away from the breakup point. Therefore, we present two sets of statistics in figure 9(c) along with their respective sample counts: first the ensemble-averaged across all necks, and secondly, the ensemble-average for only the last neck leading to breakup. The number of samples for the "last neck" ensemble decreases rapidly at around $10\tau_\eta$ from breakup, providing evidence that for almost all breakup events, neck formation is triggered by the extension shown in figure 9(a) rather than existing before the onset of the final breakup extension. Note that the low sample count away from the breakup point, together with the definition of $\langle r_{\text{neck}} \rangle$, causes spurious peaks of this data, since before the pinch-off regime the location of the minimum neck radius can shift on the mesh as the AMR algorithm dynamically adds and removes mesh nodes. The ensemble average across all necks demonstrates that there are intermittent subcritical necks in the droplet shape with a mean value of around $\langle r_{\text{neck}} \rangle / a \approx 0.6$ that are not associated with the breakup event. Note that the falling number of samples at larger $(t_e - t)$ in the ensemble of all necks is influenced by the lack of data points for breakup events which occur less than $50\tau_\eta$ from the start of the simulation, rather than suggesting a correlation between earlier neck formation and the final breakup event.

To investigate the stresses acting on the droplets before breakup, the ensemble-averaged strain rate eigenvalues, $\langle \Lambda_k \rangle$, are plotted in figure 9(d), along with the component of the strain rate tensor acting in the direction of the droplet extension (the numerator of β in equation 16). We find that $\langle \Lambda_1 \rangle$ begins to peak near the same point where $\langle l \rangle$ begins to diverge in figure 9(a). This is accompanied by an even stronger negative peak of $\langle \Lambda_3 \rangle$. Even though the maximum strain rate peaks strongly, the misalignment of the droplet with the mean flow (decreasing $\langle \beta \rangle$) causes the actual extensional strain rate along the droplet's long axis to decrease within $10\tau_\eta$ of breakup.

5.5. Strain rate intermittency and rare events

To investigate the source of this strong strain rate fluctuation before breakup, figure 10 shows the probability density functions (PDFs) of Λ_k from the *TURB-Lagr* database plotted as solid lines in semilogarithmic (figure 10(a)) and linear scales (figure 10(b)). The same PDFs conditioned on

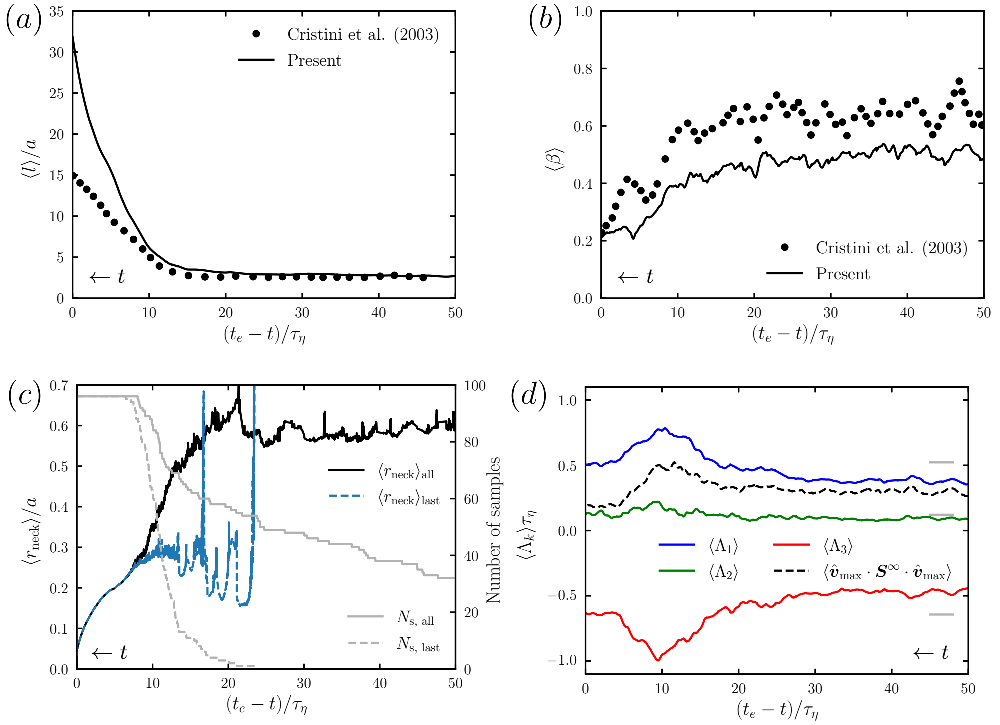


Figure 9: Ensemble-averaged drop length (a), orientation parameter (b), neck radius (c), and strain rate eigenvalues (d) near breakup for the present simulations ($Ca = 0.3$, $Re_{\lambda_T} \simeq 310$) compared to the results of Cristini et al. [13] ($Ca = 0.3$, $Re_{\lambda_T} = 54$). Horizontal gray lines in panel (d) correspond to the mean values of $\langle \Lambda_k \rangle$.

the existence of a neck on the droplet (condition 1 from §4.4) are plotted with dashed lines in the same figure. The roughness of the tails of the PDFs are due to the limited number of samples of the present ensemble. As evident through comparison of the PDFs, extreme events from the tails of the distributions are contributors to the most severe deformations and breakup of the droplets, and the conditional PDFs indicate even stronger non-Gaussian behavior than the baseline PDF. As opposed to the simulations of Crialesi-Esposito et al. [8], the present one-way coupled simulations offer no mechanism to generate vorticity in the surrounding HIT through capillary action, and the droplet breakup is a result of the HIT intermittency, similar to the mechanism proposed by Vela-Martin and Avila [5] for larger K-H droplets.

As a result of the droplet sensitivity to rare events and the intermittency of the small-scale turbulence, the local instantaneous dissipation rate, ε, that triggers the droplet breakup tends to be larger than the mean dissipation $\langle \varepsilon \rangle$. Therefore, the local Kolmogorov scales decrease, and $\dot{\gamma}_\eta$ and the effective capillary number increase. While the mean capillary number, Ca, describes mean behaviors of the system, breakup events are associated with the extreme fluctuations in ε, suggesting that the critical capillary number for breakup has some relation $Ca_{cr} \sim Ca Re_{\lambda_T}^{\beta}$. As argued by Vela-Martin and Avila [5], observations of droplets for this Ca_{cr} may only be possible after long times due to the rarity of the requisite intermittent events. It is also well known that dissipation quantities (and resulting strain rates) become increasingly non-Gaussian at higher Re_{λ_T} [26].

5.6. Pinch-off behavior

Finally, we examine the pinch-off behavior of the droplets as they approach breakup. In all trajectories resulting in breakup, the droplets form a symmetric double neck after the strong extension of the droplet, where the symmetric shape is due to the imposition of Stokes flow on the droplet scale (a typical double-neck shape is shown in figure 5(*a*)). These terminal droplet shapes are consistent with those reported in Cristini et al. [13] and Milan et al. [21]. The result is that both ends of the droplet undergo pinch-off, resulting in two daughter droplets plus a long, nearly cylindrical central filament, which may undergo a Rayleigh-Plateau style instability under relaxation.

Figure 11(*a*) shows the time history of the minimum neck radius (r_{neck}) approaching breakup for each trajectory, along with the ensemble average shown as a dashed line. Clearly the histories of r_{neck} are strongly dependent

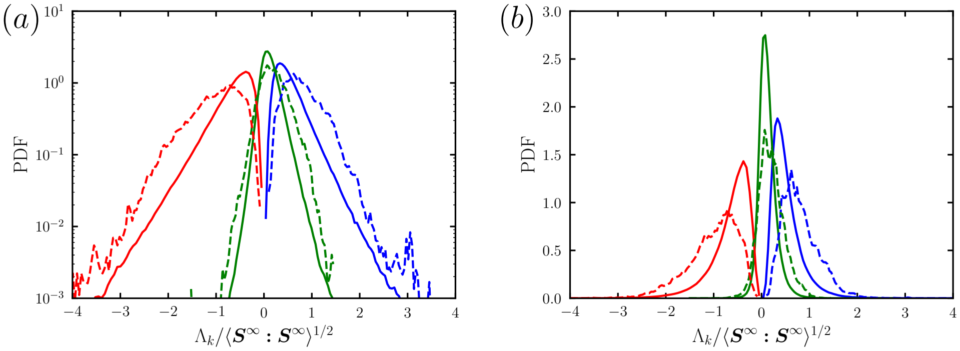


Figure 10: Probability density functions in semilogarithmic (a) and linear (b) scales of the largest (Λ_1, blue), middle (Λ_2, green), and smallest (Λ_3, red) eigenvalues of the fluctuating strain rate tensor from the *TURB-Lagr* database [23] (solid lines). The PDF of the same eigenvalues for the present simulation ensemble conditioned on the existence of a neck on the droplet are also shown with dashed lines.

on the local history of the droplet shape and strain field. In many cases, the neck does not experience a monotonic contraction, exhibiting multiple local minima/maxima as it approaches breakup. Often a neck may develop but recover without inducing breakup. This emphasizes the importance of choosing an appropriate value of r_{neck} to terminate the computations such that local minima are not misidentified as breakup events, which is discussed in §5.7.

To investigate the ultimate regime of the pinch off, we extend the simulation for the trajectory from figure 5(a) by over $0.5\tau_\eta$ and plot the resulting minimum neck radius in figure 11(b). As shown in the figure, the scaling of r_{neck} transitions from a nonlinear scaling to a linear scaling of $r_{\text{neck}} \sim (t_b - t)^1$ before breakup, as expected in the Stokes regime [55]. The ultimate breakup time in this relation, t_b, was calculated based on a fit of r_{neck} in the linear regime. While the neck does eventually transition to this linear scaling, it is notable that this scaling only exists for a tiny fraction of the overall contraction history of the neck towards pinch-off, as evidenced by the same black line plotted in figure 11(a). The insets of figure 11(b) illustrate the mesh refinement around the miniscule neck of the droplet, emphasizing the effectiveness of the AMR algorithm and the decision to terminate the simulation before the computationally-expensive linear regime.

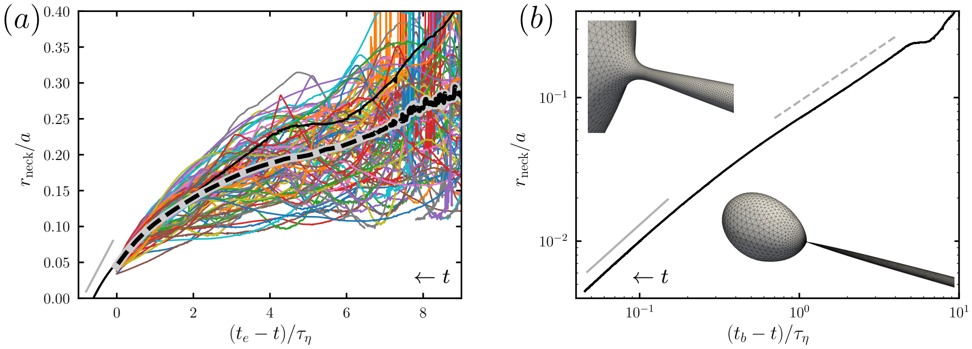


Figure 11: Necking behavior of the droplets before breakup. The neck radius history for all trajectories is shown in linear scale in panel (a), along with the average neck radius history in the dashed black and gray line. Note that the abscissa in this case is based on the simulation termination time, t_e, where the neck radius falls below $a/20$. Panel (b) shows the neck radius versus the time to breakup for the trajectory shown in figure 5 in logarithmic scale, where the breakup time (t_b) has been calculated based on a fit to the data in the linear regime. Scalings of $(t_b - t)^1$ and $(t_b - t)^{3/4}$ are shown in solid and dashed gray lines, respectively. The insets show the shape of one end of the droplet in the linear regime and a zoomed view of the neck.

5.7. *Subcritical neck formation*

Given that neck formation is a requirement for pinch-off and subsequent droplet breakup, we investigate the statistics of necking events that are not associated with breakup as a continuation of the discussion associated with figure 9(c). In these "subcritical" events, a neck is formed on the droplet but does not enter the final pinch off regime. Instead, r_{neck} reaches a minimum value and subsequently grows, avoiding breakup.

To properly discern breakup and non-breakup events, it is important to establish that the $a/20$ threshold for the neck radius used in §4.4 does not misidentify subcritical necks as breakup events, as we desire that all $r_{\mathrm{neck}} < a/20$ result in breakup. To establish the sufficiency of this $a/20$ threshold, figure 12 plots a histogram of local minima of the neck radius across the ensemble, excluding breakup events. The results in figure 12 demonstrate that the chosen breakup threshold of $a/20$ (vertical dashed line) is below the left-hand tail of the distribution, where the number of droplet necks which undergo recovery falls off steeply. The threshold is also not so far to the left of the distribution as to unduly increase the computational cost.

Using the ensemble of subcritical neck minima from figure 12, the ensemble-

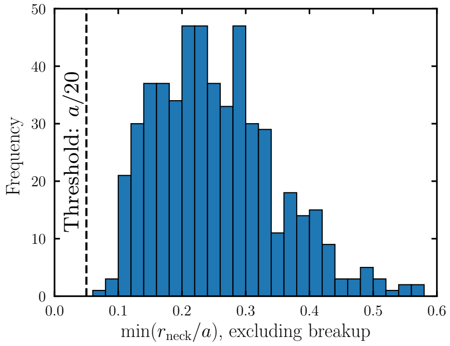


Figure 12: Histogram of occurrences of local neck minima for all trajectories, excluding final breakup events. The vertical dashed line denotes the numerical threshold for droplet breakup.

averaged statistics of $\langle \Lambda_k \rangle$, $\langle r_{\mathrm{neck}} \rangle$, and $\langle l \rangle$ are presented in figure 13, centered about the time, $t_{\min}$, at which r_{neck} reaches a local minimum. Figure 13(a) demonstrate that these necking events are associated with an elevated maximum strain rate eigenvalue $\langle \Lambda_1 \rangle$ for around $10\tau_\eta$ preceding $t_{\min}$, and a local spike in the eigenvalues at only $0.5\tau_\eta$ from $t_{\min}$. The peaks in $\langle \Lambda_k \rangle$ quickly drop over a timescale of τ_η centered around $t_{\min}$, before decaying to their average values by around $6\tau_\eta$ after $t_{\min}$. One notable result is that as opposed to figure 9(d), the droplet axis projected strain rate $\langle \hat{\boldsymbol{v}}_{\max} \cdot \boldsymbol{S}^\infty \cdot \hat{\boldsymbol{v}}_{\max} \rangle$ does not actually achieve a maximum at the point where $\langle \Lambda_1 \rangle$ peaks, and actually remains suppressed at $t_{\min} + 10\tau_\eta$ compared to its value at $t_{\min} - 10\tau_\eta$.

Figure 13b shows that $\langle l \rangle$ increases in response to the elevated $\langle \Lambda_1 \rangle$ throughout the ensemble, while also exhibiting a local maxima just past $t_{\min}$ in response to the locally-elevated strain rate. $\langle r_{\mathrm{neck}} \rangle$ steadily decreases before $t_{\min}$ until the local maxima in $\langle \Lambda_k \rangle$ just before $t_{\min}$, at which point $\langle r_{\mathrm{neck}} \rangle$ rapidly decreases. However, this minimum of $\langle r_{\mathrm{neck}} \rangle$ is short-lived, having a lifetime of only $2.5\tau_\eta$. $\langle r_{\mathrm{neck}} \rangle$ rapidly increases past $t_{\min}$ in response to the local contraction of $\langle l \rangle$. After the local minimum of $\langle l \rangle$ at approximately $2\tau_\eta$ from $t_{\min}$, $\langle r_{\mathrm{neck}} \rangle$ actually continues to slowly increase despite $\langle l \rangle$ continuing to rise, perhaps due to the misalignment of the droplet axis with the flow. This could also be an artifact of averaging over some cases where the neck fully recovers and others where the droplet goes on to break up. By comparing the time history of subcritical droplet necking events to those resulting in breakup in figure 9, we elucidate some clues about which types

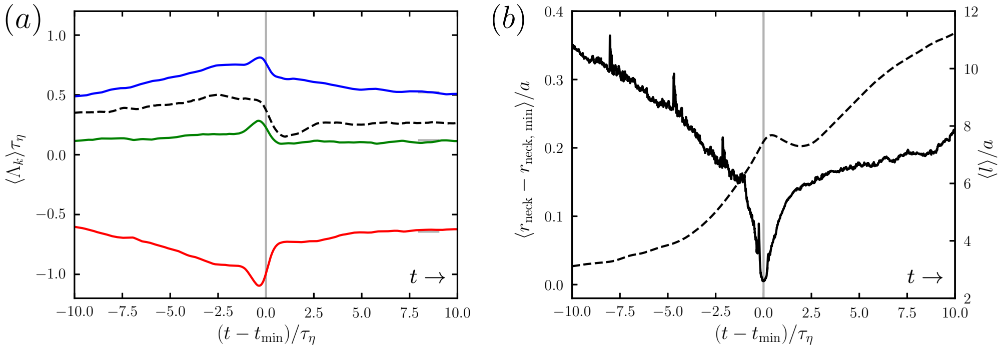


Figure 13: Ensemble-averaged statistics of subcritical necks centered about local neck minima. Panel (*a*) shows the ensemble-averaged strain rate eigenvalues quantities, as plotted in figure 9(*d*), while panel (*b*) shows the ensemble-averaged droplet length (dashed line) and neck radius (solid line).

of disturbances and droplet shapes are most susceptible to result in breakup. Specifically, while both breakup events (figure 9(*d*)) and subcritical events (figure 13(*a*)) exhibit an elevated $\langle \Lambda_1 \rangle$, subcritical events are associated with a peak of $\langle \Lambda_1 \rangle$ with a timescale of $1\tau_\eta$, during which the droplet axis misaligns with respect to the strain field, inducing a contraction of the droplet length and recovery of the neck. Given the rotational misalignment of the droplet, it would be interesting in future studies to also investigate the effect of the rotation rate tensor and associated quantities (e.g. enstrophy; vortex stretching/tilting), especially given the recently-established connection between larger K-H droplets and enstrophy generation [9].

6. Conclusions

This work presented a BEM framework capable of resolving droplet breakup across a range of capillary numbers (Ca) and viscosity ratios (λ), and provided detailed pinch-off and droplet statistics at a higher Reynolds number, $Re_{\lambda_T} \simeq 310$, than the work of Cristini et al. [13], which was confined to low-Re_{λ_T} [27, 26]. Distributions of the droplet length and orientation were found to exhibit longer tails in the present simulations than under the same conditions at lower Re_{λ_T}, echoing the non-Gaussian behavior of strain rates under extreme dissipation events at higher Re_{λ_T} [26]. As in Cristini et al. [13], the droplets undergo an elongational strain near breakup, followed by a misalignment with the externally-imposed flow. However, the average ultimate droplet length in the present simulations is approximately two times higher than at lower Re_{λ_T}, presumably affecting the daughter droplet sizes produced by breakup even at the same Ca and λ. This highlights the multiscale nature of sub-Kolmogorov-scale droplet breakup, even in the ideal assumption of local Stokes flow. This is a natural consequence of the droplet breakup dependence on extreme strain rate events, which link directly to the intermittency of the dissipation rate, echoing the conclusions of Vela-Martin and Avila [5] for larger droplets.

An investigation of the pinch-off behavior of the droplet necks was conducted to reveal a non-monotonic, non-global contraction of the neck radius near breakup. An extended simulation of a trajectory past the numerical pinch-off threshold demonstrated that the shift into the linear regime expected from similarity solutions of viscous neck pinch-off occurs only in the final stages of the neck contraction. Statistics centered about the subcritical minimum neck radii demonstrated that subcritical (non-breakup-inducing) events are characterized by a fluctuation in the strain rates of timescale τ_η superposed on a broader period of elevated strain. This local strain rate maximum induces a misalignment of the droplet with the strain field, inducing a local extension and contraction of the droplet length and recovery of the neck radius, avoiding breakup.

As a straight-forward extension of the present work, it would be valuable to investigate behaviors across the (Ca, λ, Re_{λ_T}) parameter space, especially to investigate the Re_{λ_T}-dependence droplet breakup statistics and daughter droplet sizes. This work could also draw upon the findings of Saeedipour and Schneiderbauer [9] to investigate how their vortex stretching/capillary enstrophy balance extends to the sub-Kolmogorov regime. Further work could also include droplet behavior with insoluble surfactants [33, 34, 35], or trajectories in more complex flows [15, 16]. In particular, studying realistic emulsification devices using a CFD-BEM framework could allow droplet breakup statistics to be connected to specific regions of the mixer.

Finally, considering the multiscale nature of the problem, it would be valuable to investigate more extreme regimes of droplet dynamics, focusing on either inter-scale or nanoscale effects. First, it would be interesting to consider droplets near the Kolmogorov scale ($d \sim \eta$) instead of the vanishing limit $d \ll \eta$. Even for cases where $d < \eta$, droplet elongations may become long enough that there are interactions with the Kolmogorov-scale eddies such that the assumption of Stokes flow and a locally linear velocity gradient are no longer valid, such that the present BEM framework would no longer be appropriate. This is especially evident in the present results, where droplet elongations are larger than at lower Taylor Reynolds numbers [13]. Secondly, as droplets approach the nanoscale, one could investigate low-surface-tension droplet breakup with fluctuating hydrodynamic effects, which have been studied in relation to the Rayleigh-Plateau instability at the nanoscale [56, 57, 58].

Acknowledgements

The author thanks Dr. Stefan Zitz, Dr. Johan Remmelgas, and Professor Johannes Khinast for their feedback on the numerical method and analysis. This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953110. The Research Center Pharmaceutical Engineering (RCPE) is funded within the framework of COMET–Competence Centers for Excellent Technologies by BMK, BMAW, Land Steiermark, and SFG. The COMET program is managed by the FFG.

Appendix A. Details of the BEM discretization

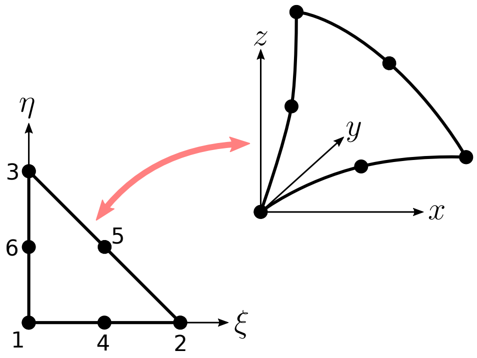
The following details the discretizaiton and convergence properties of the present BEM algorithm. The integrals in equation 8 are performed using a three-point Gauss-Legendre quadrature [45] based on a mapping of the element from three-dimensional space (x, y, z) to the unit right triangle (ξ, η), as shown in figure A.14. The values at the quadrature points are interpolated from the solution variables at the six edge and corner nodes using

$$\phi(\xi, \eta) = \sum_{i=1}^{6} \phi_i N_i(\xi, \eta), \tag{A.1}$$

where the quadratic shape functions for the Lagrange triangle are those listed in figure A.14.

In equation 9, the differential area associated with the quadrature point is calculated as

$$dS(\boldsymbol{x}_0) = w_{quad} h_s d\xi d\eta = w_{quad} \left\| \frac{\partial \boldsymbol{x}_0}{\partial \xi} \times \frac{\partial \boldsymbol{x}_0}{\partial \eta} \right\| d\xi d\eta, \tag{A.2}$$



$$N_1 = (1-\xi-\eta)(1-2\xi-2\eta)$$
$$N_2 = \xi(2\xi-1)$$
$$N_3 = \eta(2\eta-1)$$
$$N_4 = 4\xi(1-\xi-\eta)$$
$$N_5 = 4\xi\eta$$
$$N_6 = 4\eta(1-\xi-\eta)$$

Figure A.14: The six-node Lagrange triangle element used in the present BEM algorithm in (x, y, z) space and mapped to the unit right triangle (ξ, η). The shape functions (N_i) describing the element are also listed.

where w_{quad} is the Dunavant weight for the quadrature point and h_s is the square root of the determinant of the metric tensor of the surface. Figure A.15 shows the error associated with the integration of a specified scalar field on the surface of an ellipsoid with semi-major axis lengths of $(1, 0.5, 0.3)$ at different grid sizes. The grids were discretized by subdividing an icosahedron grid on a sphere and scaling it to the ellipsoidal shape. We choose the number of quadrature points as $K_\Delta = 3$ to match the second order interpolation accuracy of the Lagrange triangle elements, and as expected, figure A.15(b) shows that the integration error converges as N_Δ^{-2}.

The relevant ξ- and η-derivatives on the surface are evaluated following equation A.1 as

$$\frac{\partial \phi}{\partial \xi}(\xi, \eta) = \sum_{i=1}^{6} \phi_i \frac{\partial N_i}{\partial \xi}(\xi, \eta), \qquad \frac{\partial \phi}{\partial \eta}(\xi, \eta) = \sum_{i=1}^{6} \phi_i \frac{\partial N_i}{\partial \eta}(\xi, \eta). \tag{A.3}$$

The surface normal vector calculation follows as the cross product of the surface tangent vectors:

$$\boldsymbol{n}(\boldsymbol{x}_0) = \frac{1}{h_s}\left(\frac{\partial \boldsymbol{x}_0}{\partial \xi} \times \frac{\partial \boldsymbol{x}_0}{\partial \eta}\right). \tag{A.4}$$

Figure A.16 shows the convergence of the mean and maximum surface normal error with grid resolution, again for an ellipsoid with semi-major axis lengths $(1, 0.5, 0.3)$. We find in figure A.16(a) that the mean error reduces with the number of elements (N_Δ) with second order as N_Δ^{-2}, while the maximum

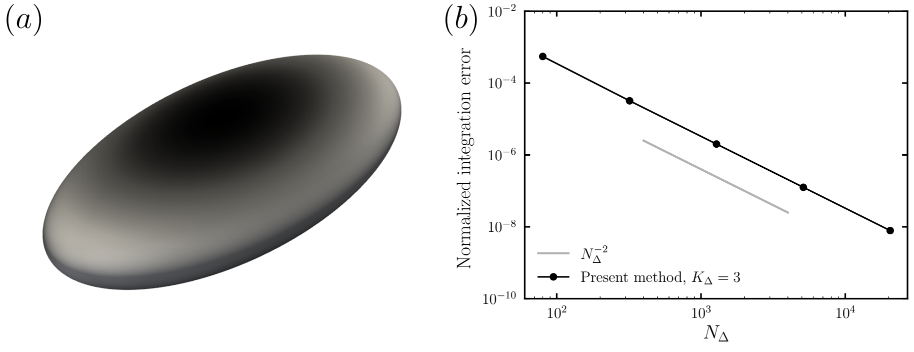


Figure A.15: The scalar field $\phi = \sqrt{\frac{x^2}{a^4} + \frac{y^2}{b^4} + \frac{z^2}{c^4}}$ on the surface of an ellipsoid (a) with semi-major axis lengths $(a, b, c) = (1, 0.5, 0.3)$ and the normalized error (b) of the surface integral calculation $\int_S \phi dS \approx \sum_{elem=1}^{N_\Delta} \sum_{qpt=1}^{K_\Delta} \phi dS$ of this scalar versus grid size (N_Δ).

error in figure A.16(b) decreases with an order between N_Δ^{-1} and N_Δ^{-2}. We also compare the convergence to the BEM of Zinchenko and Davis [59], which is based on a discretization of three-node (flat) triangles, requiring that the surface normal be calculated based on a local fit of a paraboloid-spline to the local region of nodes. We find that our method converges at a similar rate for both the mean and maximum errors, but with significantly higher accuracy (around two orders of magnitude) across all grid sizes.

In addition to the surface normals, of particular importance is the calculation of the local mean surface curvature of the droplet, since it plays an important role in the droplet evolution due to surface tension effects. The curved Laplace triangular elements lend themselves naturally to the calculation of the surface mean curvature (H) and Gaussian curvature (K), which are evaluated based on the first and second fundamental forms of the curved surface as

$$H = \frac{EN - 2FM + GL}{2(EG - F^2)}, \qquad K = \frac{LN - M^2}{EG - F^2}, \tag{A.5}$$

where

$$\begin{aligned} E &= \frac{\partial \boldsymbol{x}_0}{\partial \xi} \cdot \frac{\partial \boldsymbol{x}_0}{\partial \xi}, \qquad F = \frac{\partial \boldsymbol{x}_0}{\partial \xi} \cdot \frac{\partial \boldsymbol{x}_0}{\partial \eta}, \qquad G = \frac{\partial \boldsymbol{x}_0}{\partial \eta} \cdot \frac{\partial \boldsymbol{x}_0}{\partial \eta}, \\ L &= \frac{\partial \boldsymbol{x}_0}{\partial \xi} \cdot \frac{\partial \boldsymbol{n}}{\partial \xi}, \qquad M = \frac{\partial \boldsymbol{x}_0}{\partial \xi} \cdot \frac{\partial \boldsymbol{n}}{\partial \eta}, \qquad N = \frac{\partial \boldsymbol{x}_0}{\partial \eta} \cdot \frac{\partial \boldsymbol{n}}{\partial \eta}. \end{aligned} \tag{A.6}$$

As was performed for the surface normal calculation, we also examine the

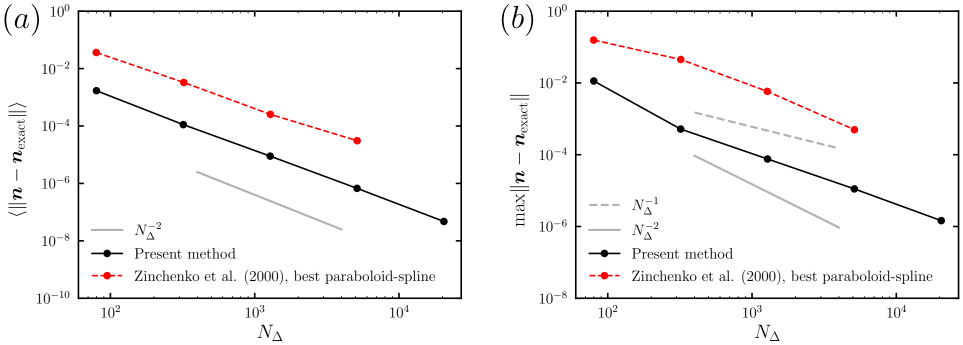


Figure A.16: Mean (*a*) and maximum (*b*) errors in the calculation of surface normal vectors for an ellipsoid with semi-major axis lengths $(1, 0.5, 0.3)$ compared to the results of Zinchenko and Davis [59].

order of accuracy of the mean curvature calculation compared to the results of Zinchenko et al. [47]. In this case, we consider a spheroid with semi-major axis lengths of $(1, 0.4, 0.4)$, again discretized by successive subdivision of an icosahedron grid. Figure A.17 shows that both the mean and maximum curvature errors for the present method converge as N_Δ^{-1}. Comparing to the contour integral and paraboloid fitting curvature calculation methods of Zinchenko et al. [47], we find that the present method has a similar first order accuracy, but results in a lower average error versus Zinchenko et al. [47]. In terms of the maximum error, their results show similar error magnitudes to the present method, perhaps due to the locally-fitted curvature calculation of their method versus our truly local curvature calculation.

Appendix B. AMR component algorithms

The Laplacian mesh smoothing, edge node adjustment, grid refinement, grid coarsening, and edge flipping algorithms are outlined in algorithms 2, 3, 4, 5, and 6, respectively. As a general comment on notation, $ielm$ and ipt correspond to global element and node numbers, while jpt and $jelm$ correspond to local numbering.

References

[1] R. Ni, Deformation and breakup of bubbles and drops in turbulence, Annual Review of Fluid Mechanics 56 (2024) 319–347.

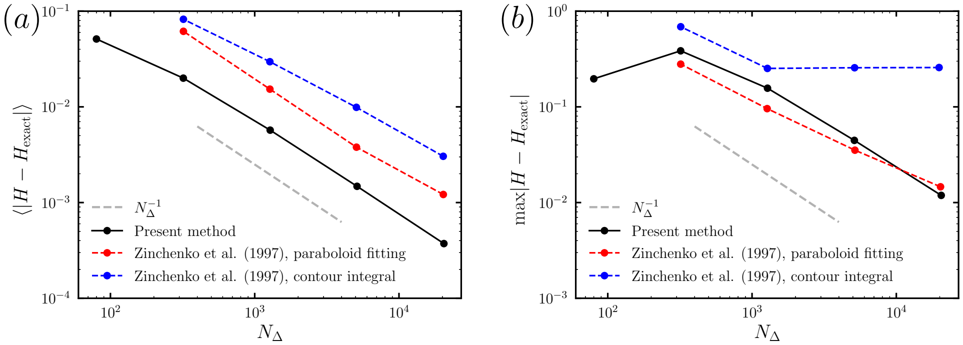


Figure A.17: Mean (a) and maximum (b) errors in the surface mean curvature calculation for a spheroid with semi-major axis lengths $(1, 0.4, 0.4)$ compared to the results of Zinchenko et al. [47].

[2] S. Elghobashi, Direct numerical simulation of turbulent flows laden with droplets or bubbles, Annual Review of Fluid Mechanics 51 (2019) 217–244.

[3] A. Kolmogorov, On the breakage of drops in a turbulent flow, in: Dokl. Akad. Navk. SSSR, volume 66, 1949, pp. 525–528.

[4] J. O. Hinze, Fundamentals of the hydrodynamic mechanism of splitting in dispersion processes, AIChE Journal 1 (1955) 289–295.

[5] A. Vela-Martin, M. Avila, Memoryless drop breakup in turbulence, Science Advances 8 (2022) eabp9561.

[6] M. Crialesi-Esposito, M. E. Rosti, S. Chibbaro, L. Brandt, Modulation of homogeneous and isotropic turbulence in emulsions, Journal of Fluid Mechanics 940 (2022) A19.

[7] M. Crialesi-Esposito, S. Chibbaro, L. Brandt, The interaction of droplet dynamics and turbulence cascade, Communications Physics 6 (2023) 5.

[8] M. Crialesi-Esposito, G. Boffetta, L. Brandt, S. Chibbaro, S. Musacchio, How small droplets form in turbulent multiphase flows, Physical Review Fluids 9 (2024) L072301.

[9] M. Saeedipour, S. Schneiderbauer, Enstrophy transport rates determine the kolmogorov-hinze scale in turbulent fragmentation of droplets, Phys. Rev. Fluids 10 (2025) 044301.

[10] A. Gupta, H. B. Eral, T. A. Hatton, P. S. Doyle, Nanoemulsions: formation, properties and applications, Soft Matter 12 (2016) 2826–2841.

[11] J. A. Boxall, C. A. Koh, E. D. Sloan, A. K. Sum, D. T. Wu, Droplet size scaling of water-in-oil emulsions under turbulent flow, Langmuir 28 (2012) 104–110. PMID: 22047095.

[12] A. Gupta, H. B. Eral, T. A. Hatton, P. S. Doyle, Controlling and predicting droplet size of nanoemulsions: scaling relations with experimental validation, Soft Matter 12 (2016) 1452–1458.

[13] V. Cristini, J. Bławzdziewicz, M. Loewenberg, L. R. Collins, Breakup in stochastic Stokes flows: sub-Kolmogorov drops in isotropic turbulence, Journal of Fluid Mechanics 492 (2003) 231–250.

[14] L. Biferale, C. Meneveau, R. Verzicco, Deformation statistics of sub-Kolmogorov-scale ellipsoidal neutrally buoyant drops in isotropic turbulence, Journal of fluid mechanics 754 (2014) 184–207.

[15] V. Spandan, D. Lohse, R. Verzicco, Deformation and orientation statistics of neutrally buoyant sub-Kolmogorov ellipsoidal droplets in turbulent Taylor-Couette flow, Journal of fluid mechanics 809 (2016) 480–501.

[16] V. Spandan, R. Verzicco, D. Lohse, Deformable ellipsoidal bubbles in Taylor-Couette flow with enhanced Euler-Lagrangian tracking, Physical Review Fluids 2 (2017) 104304.

[17] S. S. Ray, D. Vincenzi, Droplets in isotropic turbulence: deformation and breakup statistics, Journal of Fluid Mechanics 852 (2018) 313–328.

[18] P. Maffettone, M. Minale, Equation of change for ellipsoidal drops in viscous flow, Journal of Non-Newtonian Fluid Mechanics 78 (1998) 227–241.

[19] P. L. Johnson, C. Meneveau, Predicting viscous-range velocity gradient dynamics in large-eddy simulations of turbulence, Journal of Fluid Mechanics 837 (2018) 80–114.

[20] D. Taglienti, F. Guglietta, M. Sbragaglia, Reduced model for droplet dynamics in shear flows at finite capillary numbers, Physical Review Fluids 8 (2023) 013603.

[21] F. Milan, L. Biferale, M. Sbragaglia, F. Toschi, Sub-Kolmogorov droplet dynamics in isotropic turbulence using a multiscale lattice Boltzmann scheme, Journal of Computational Science 45 (2020) 101178.

[22] D. Taglienti, F. Guglietta, M. Sbragaglia, Droplet dynamics in homogeneous isotropic turbulence with the immersed boundary–lattice Boltzmann method, Physical Review E 110 (2024) 015302.

[23] L. Biferale, F. Bonaccorso, M. Buzzicotti, C. Calascibetta, TURB-Lagr. A database of 3d Lagrangian trajectories in homogeneous and isotropic turbulence, 2024. `arXiv:2303.08662`.

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

[25] J. M. Rallison, Note on the time-dependent deformation of a viscous drop which is almost spherical, Journal of Fluid Mechanics 98 (1980) 625–633.

[26] K. R. Sreenivasan, R. A. Antonia, The phenomenology of small-scale turbulence, Annual review of fluid mechanics 29 (1997) 435–472.

[27] T. Ishihara, T. Gotoh, Y. Kaneda, Study of high-Reynolds number isotropic turbulence by direct numerical simulation, Annual Review of Fluid Mechanics 41 (2009) 165–180.

[28] R. Shinnar, On the behaviour of liquid dispersions in mixing vessels, Journal of Fluid Mechanics 10 (1961) 259–275.

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

[30] H. A. Stone, L. G. Leal, The influence of initial deformation on drop breakup in subcritical time-dependent flows at low Reynolds numbers, Journal of Fluid Mechanics 206 (1989) 223–263.

[31] H. A. Stone, L. G. Leal, The effects of surfactants on drop deformation and breakup, Journal of Fluid Mechanics 220 (1990) 161–186.

[32] W. Milliken, H. A. Stone, L. Leal, The effect of surfactant on the transient motion of Newtonian drops, Physics of Fluids A: Fluid Dynamics 5 (1993) 69–79.

[33] A. Z. Zinchenko, R. H. Davis, General rheology of highly concentrated emulsions with insoluble surfactant, Journal of Fluid Mechanics 816 (2017) 661–704.

[34] J. R. Gissinger, A. Z. Zinchenko, R. H. Davis, Drops with insoluble surfactant squeezing through interparticle constrictions, Journal of Fluid Mechanics 878 (2019) 324–355.

[35] A. Z. Zinchenko, J. R. Gissinger, R. H. Davis, Flow of a concentrated emulsion with surfactant through a periodic porous medium, Journal of Fluid Mechanics 953 (2022) A21.

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

[37] A. Z. Zinchenko, R. H. Davis, Shear flow of highly concentrated emulsions of deformable drops by numerical simulations, Journal of Fluid Mechanics 455 (2002) 21–61.

[38] A. Z. Zinchenko, R. H. Davis, Large-scale simulations of concentrated emulsion flows, Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 361 (2003) 813–845.

[39] A. Z. Zinchenko, R. H. Davis, Extensional and shear flows, and general rheology of concentrated emulsions of deformable drops, Journal of Fluid Mechanics 779 (2015) 197–244.

[40] A. Z. Zinchenko, R. H. Davis, Algorithm for flow of highly-concentrated emulsions through a narrow constriction, Journal of Computational Physics 438 (2021) 110363.

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

[42] A. Z. Zinchenko, R. H. Davis, Emulsion flow through a packed bed with multiple drop breakup, Journal of Fluid Mechanics 725 (2013) 611–663.

[43] J. R. Gissinger, A. Z. Zinchenko, R. H. Davis, Drop squeezing between arbitrary smooth obstacles, Journal of Fluid Mechanics 908 (2021) A33.

[44] C. Pozrikidis, Boundary integral and singularity methods for linearized viscous flow, Cambridge University Press, 1992.

[45] D. A. Dunavant, High degree efficient symmetrical Gaussian quadrature rules for the triangle, International Journal for Numerical Methods in Engineering 21 (1985) 1129–1148.

[46] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (1986) 856–869.

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

[48] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, Journal of Computational Physics 73 (1987) 325–348.

[49] L. Greengard, Z. Gimbutas, FMM3D: Fast multipole methods in three dimensions, `https://github.com/flatironinstitute/FMM3D`, 2010–2012.

[50] V. Cristini, J. Bławzdziewicz, M. Loewenberg, An adaptive mesh algorithm for evolving surfaces: Simulations of drop breakup and coalescence, Journal of Computational Physics 168 (2001) 445–463.

[51] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, B. Levy, Polygon mesh processing, AK Peters, 2010.

[52] M. Dunyach, D. Vanderhaeghe, L. Barthe, M. Botsch, Adaptive Remeshing for Real-Time Mesh Deformation, in: M.-A. Otaduy, O. Sorkine (Eds.), Eurographics 2013 - Short Papers, The Eurographics Association, 2013.

[53] R. J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, SIAM Journal on Numerical Analysis 33 (1996) 627–665.

[54] D. Kim, P. Moin, Subgrid-scale capillary breakup model for liquid jet atomization, Combustion Science and Technology 192 (2020) 1334–1357.

[55] J. Eggers, E. Villermaux, Physics of liquid jets, Reports on progress in physics 71 (2008) 036601.

[56] C. Zhao, J. E. Sprittles, D. A. Lockerby, Revisiting the Rayleigh–Plateau instability for the nanoscale, Journal of Fluid Mechanics 861 (2019) R3.

[57] C. Zhao, D. A. Lockerby, J. E. Sprittles, Dynamics of liquid nanothreads: Fluctuation-driven instability and rupture, Physical Review Fluids 5 (2020) 044201.

[58] B. Barker, J. B. Bell, A. L. Garcia, Fluctuating hydrodynamics and the Rayleigh–Plateau instability, Proceedings of the National Academy of Sciences 120 (2023) e2306088120.

[59] A. Z. Zinchenko, R. H. Davis, An efficient algorithm for hydrodynamical interaction of many deformable drops, Journal of Computational Physics 157 (2000) 539–587.

Algorithm 2 The Laplacian mesh smoothing algorithm.

```
Initialize $\boldsymbol{x}_{new} \leftarrow \boldsymbol{x}$ for each point
for $iter = 1$ to $N_{smooth}$ do                              ▷ Laplacian smoothing of corner nodes
    for $ipt = 1$ to $N_p$ do
        if $ipt$ is a corner point then
            Initialize $\boldsymbol{x}_t \leftarrow 0$, $w_{sum} \leftarrow 0$
            for neighbor elements $jelm$ do
                Calculate element area $S_\Delta$
                Calculate element weight $w \leftarrow S_\Delta \left( \frac{1}{6} \sum_{jpt=1}^{6} L_\Delta^{(jpt)} \right)^{-2}$
                Calculate element barycenter $\boldsymbol{x}_b \leftarrow \frac{1}{6} \sum_{jpt=1}^{6} \boldsymbol{x}^{(jpt)}$
                $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + w\boldsymbol{x}_b$
                $w_{sum} \leftarrow w_{sum} + w$
            end for
            $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t / w_{sum}$
            $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + \boldsymbol{n}^{(ipt)} \left[ \boldsymbol{n}^{(ipt)} \cdot (\boldsymbol{x}^{(ipt)} - \boldsymbol{x}_t) \right]$      ▷ Project $\boldsymbol{x}_t$ onto tangent plane
            $\boldsymbol{x}_{new}^{(ipt)} \leftarrow (1 - \lambda_s)\boldsymbol{x}_{new}^{(ipt)} + \lambda_s \boldsymbol{x}_t$      ▷ Smoothing coefficient $0 < \lambda_s \leq 1$
        end if
    end for
    if $\max(\|\boldsymbol{x}_{new} - \boldsymbol{x}\|) <$ threshold then
        exit
    end if
end for
Project new corner nodes $\boldsymbol{x}_{new}$ onto existing grid surface and interpolate variables
for $ipt = 1$ to $N_p$ do                                      ▷ Adjust edge node positions
    if $ipt$ is an edge node then
        Adjust $\boldsymbol{x}^{(ipt)}$ to lie between neighboring corner nodes       ▷ See algorithm 3
    end if
end for
```

Algorithm 3 The edge node adjustment algorithm.

```
Given: Edge point ipt
Initialize $\boldsymbol{x}_{new} \leftarrow \boldsymbol{x}^{(ipt)}$, $\boldsymbol{V} \leftarrow 0$
Calculate $\boldsymbol{x}_{ref}$ as the average of neighboring two corner node coordinates
Determine which neighbor element jelm that ipt should be adjusted on
Determine initial local coordinates $(\xi, \eta)$ of ipt on jelm
repeat
    $\boldsymbol{F} \leftarrow \boldsymbol{x}_{ref} - \boldsymbol{x}_{new}$
    Calculate normal vector $\boldsymbol{n}$ at $(\xi, \eta)$ on jelm
    $\boldsymbol{F} \leftarrow \boldsymbol{F} - \boldsymbol{n}(\boldsymbol{F} \cdot \boldsymbol{n})$        ▷ Remove normal component
    $\boldsymbol{V} \leftarrow \boldsymbol{V} + dt(\boldsymbol{F} - 2\boldsymbol{V})$        ▷ where $dt = 0.2\pi$
    Calculate $\frac{\partial \xi}{\partial \boldsymbol{x}}$ and $\frac{\partial \eta}{\partial \boldsymbol{x}}$ at $(\xi, \eta)$ on jelm
    $\xi \leftarrow \xi + dt\left(\boldsymbol{V} \cdot \frac{\partial \xi}{\partial \boldsymbol{x}}\right)$
    $\eta \leftarrow \eta + dt\left(\boldsymbol{V} \cdot \frac{\partial \eta}{\partial \boldsymbol{x}}\right)$
    Interpolate $\boldsymbol{x}_{new}$ coordinates at $(\xi, \eta)$
until $\|\boldsymbol{F}\| <$ tolerance
Interpolate variables at $(\xi, \eta)$
$\boldsymbol{x}^{(ipt)} \leftarrow \boldsymbol{x}_{new}$
```

Algorithm 4 The grid refinement algorithm (both cell and edge splitting).

```
for $ielm = 1$ to $N_\Delta$ do
    for edges $jedge = 1$ to 3 on $ielm$ do
        if edge length $> \frac{4}{3} L_\Delta$ then    ▷ Optimal criterion defined by Botsch et al. [51]
            Record edge $jedge$ to split
        end if
        if more than one edge recorded then
            Record element $ielm$ to split
            exit
        end if
    end for
end for
if no elements to refine then
    return
end if
repeat    ▷ Ensure regions of cell splits are surrounded by edge splits
    for $ielm = 1$ to $N_\Delta$ do
        if $ielm$ is not marked for cell split then
            for edges $jedge = 1$ to 3 of $ielm$ do
                if neighbor element is marked for cell split then
                    if $jedge$ not marked to split then
                        Record edge $jedge$ to split
                    else
                        Record element $ielm$ to split
                    end if
                end if
            end for
        end if
    end for
until no new splits are recorded
Record $N_\Delta^{old} \leftarrow N_\Delta$
Allocate new grid storage using: $N_\Delta \leftarrow N_\Delta + 3(\text{\# cell splits}) + (\text{\# edge splits})$
                                 $N_p \leftarrow N_p + 6(\text{\# cell splits}) + 2(\text{\# edge splits})$
for $ielm = 1$ to $N_\Delta^{old}$ do
    if $ielm$ is marked for cell split then
        for edges $jedge = 1$ to 3 on $ielm$ do
            if edge points haven't already been created then
                Create 2 new edge points and interpolate variables
            else
                Record global indices of the 2 existing edge points
            end if
        end for
        Create 3 new interior points and interpolate variables
        Create 3 new elements and record $elm \rightarrow pt$ connectivity
```

```
    else if $ielm$ is marked for edge split then
        if edge points haven't already been created then
            Create 2 new edge points and interpolate variables
        else
            Record global indices of the 2 existing edge points
        end if
        Create 1 new interior point and interpolate variables
        Create 1 new element and record $elm \rightarrow pt$ connectivity
    end if
end for
Update all $pt \rightarrow elm$ and $elm \rightarrow elm$ connectivity
```

Algorithm 5 The grid coarsening algorithm.

```
$ipt \leftarrow 0$
repeat
    $ipt \leftarrow ipt + 1$
    if $ipt$ is an edge point then
        if edge length $< \frac{4}{5} L_\Delta$ then        ▷ Optimal criterion defined by Botsch et al. [51]
            Record elements to remove
            for attached elements $jelm = 1$ to 2 do
                if valence of opposite corner node $\leq 4$ then
                    cycle outer
                end if
                Record points to combine
            end for
            Record points to remove
            for both neighboring corner nodes $jpt$ of $ipt$ do
                for all surrounding elements $ielm$ of $jpt$ do
                    Add $ielm$ to list of surrounding elements
                end for
                for all surrounding corner points $ipt2$ of $jpt$ do
                    if $\|\boldsymbol{x}^{(ipt2)} - \boldsymbol{x}^{(ipt)}\| > \frac{4}{3} L_\Delta$ then
                        cycle outer
                    end if
                    Add point to list of surrounding corner points
                end for
                for all surrounding edge points of $jpt$ do
                    Add point to list of surrounding edge points
                end for
            end for
            for all surrouding edge points do
                Adjust position of edge point                        ▷ See algorithm 3
                if adjusted edge point creates an invalid cell then
                    cycle outer
                end if
                Interpolate variables at edge point
            end for
            Update local $elm \rightarrow pt$, $pt \rightarrow elm$, and $elm \rightarrow elm$ connectivity
            Deallocate grid storage for removed points and cells: $N_\Delta \leftarrow N_\Delta - 2$
                                                                  $N_p \leftarrow N_p - 4$
            $ipt \leftarrow ipt - \sum((\text{removed points}) \leq ipt)$
        end if
    end if
until $ipt \geq N_p$
```

Algorithm 6 The edge flipping algorithm.

```
for $ipt = 1$ to $N_p$ do
    if $ipt$ is an edge point then
        for attached cells $jelm = 1$ to 2 do
            Record coordinates $\boldsymbol{x}_{opp}^{(jelm)}$ of opposite corner node to edge
            Record element circumcenter $\boldsymbol{x}_{cc}^{(jelm)}$
        end for
        $\boldsymbol{v}_{opp} \leftarrow \boldsymbol{x}_{opp}^{(2)} - \boldsymbol{x}_{opp}^{(1)}$
        $\boldsymbol{x}_t^{(ipt)} \leftarrow \dfrac{\boldsymbol{x}_{opp}^{(1)} + \boldsymbol{x}_{opp}^{(2)}}{2}$
        if $\|\boldsymbol{v}_{opp}\| < \left|\left(\boldsymbol{x}_{cc}^{(1)} - \boldsymbol{x}_{opp}^{(1)}\right) \cdot \boldsymbol{v}_{opp}\right| + \left|\left(\boldsymbol{x}_{cc}^{(2)} - \boldsymbol{x}_{opp}^{(2)}\right) \cdot \boldsymbol{v}_{opp}\right|$ then
            if node position $\boldsymbol{x}_t^{(ipt)}$ would produce an invalid element then
                cycle outer
            end if
            Project $\boldsymbol{x}_t^{(ipt)}$ to existing grid surface                ▷ See algorithm 3
            Interpolate variables at new point
            Update local $elm \to pt$, $pt \to elm$, and $elm \to elm$ connectivity
        end if
    end if
end for
```