Journal of Computational Physics 386 (2019) 218–247

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Simulation and validation of surfactant-laden drops in two-dimensional Stokes flow

Sara Pålsson [a,*], Michael Siegel [b], Anna-Karin Tornberg [a]

[a] KTH Mathematics, Linné Flow Centre, 100 44, Stockholm, Sweden
[b] Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102, United States of America

ARTICLE INFO

Article history:
Received 27 June 2018
Received in revised form 12 November 2018
Accepted 12 December 2018
Available online 22 February 2019

Keywords:
Insoluble surfactants
Stokes flow
Validation
Integral equations
Two-phase flow
Drop deformation
Special quadrature

ABSTRACT

Performing highly accurate simulations of droplet systems is a challenging problem. This is primarily due to the interface dynamics which is complicated further by the addition of surfactants. This paper presents a boundary integral method for computing the evolution of surfactant-covered droplets in 2D Stokes flow. The method has spectral accuracy in space and the adaptive time-stepping scheme allows for control of the temporal errors. Previously available semi-analytical solutions (based on conformal-mapping techniques) are extended to include surfactants, and a set of algorithms is introduced to detail their evaluation. These semi-analytical solutions are used to validate and assess the accuracy of the boundary integral method, and it is demonstrated that the presented method maintains its high accuracy even when droplets are in close proximity.

1. Introduction

Microfluidics is the study of fluids at the microscopic scale. It is a field of study with a range of applications [42], such as: drug delivery, diagnostic chips and microreactors. Droplets in real-world microfluidic systems have a large surface-area to volume ratio, which makes interfacial forces important; a good review is provided by Rallison [35]. In two dimensions, this translates to a large interface-length to area ratio. Surfactants locally influence the surface tension of a droplet interface. This may create non-uniformity in the surface tension which in turn generates a stress opposed to the flow, known as the Marangoni stress. Thus, the addition of surfactants can strongly influence the behaviour of the system [1,7].

Stone and Leal [40] studied how surfactants change the drop deformation and breakup using a numerical boundary integral method. They found that for the same strain rate, surfactant-covered drops would reach a more deformed steady state than their clean counterparts. The effect of viscosity-ratio on deformation was investigated for surfactant-covered droplets in an extension of this work [26]. It was shown that for cases with near-zero diffusion of surfactants along the interface of the droplet, the steady state deformation was independent of viscosity-ratio.

A microfluidic system is typically one in which viscous forces are dominant. Microfluidic multiphase flows can therefore be accurately modelled using Stokes equations. With the addition of surfactants, it is also necessary to couple the Stokes equations with a convection-diffusion equation on the droplet interfaces. A discontinuity in the normal stress across the droplet interface is created by the surface tension forces. This represents a major challenge for the simulation of these flow types. One way to avoid this difficulty is to ensure that the computational grid is always coincident with the interfaces.

* Corresponding author.
E-mail address: sarapal@kth.se (S. Pålsson).

https://doi.org/10.1016/j.jcp.2018.12.044
0021-9991/

However, this becomes very expensive when droplets deform or advect as it requires the domain to be remeshed at every time-step. The most common approach for multiphase flow simulation is instead to utilise some form of interface tracking. This involves solving the Stokes equations on a fixed computational grid in the entire domain. The locations of the droplet interfaces are computed separately using either an explicit boundary discretisation or an implicit representation; such as a level-set function or volume-of-fluid method. Some examples of methods with surfactants are those by Khatri and Tornberg [15,16] and Muradoglu and Tryggvason [27], who considered the Navier-Stokes equations and surfactants using a front-tracking scheme and finite differences in 2D and 3D respectively. Another approach is to use a diffuse interface method as Teigen et al. [43]. Furthermore, recent work has been done to extend these kinds of methods also to soluble surfactants, as is done in both 2D and 3D in [6,36,45].

Accurate treatment of the discontinuity in normal stress remains a significant challenge for all interface tracking methods when the interfaces do not align with the computational grid. The most common approach to handle this has been to regularise the surface tension forces, such as introduced in e.g. Brackbill et al. [4] or in the immersed boundary method by Peskin [30]. However, regularisation limits the accuracy of the method to first order near the interfaces. Another approach is to impose the jump conditions in the normal stress directly. LeVeque and Li [24] obtained second order accuracy using an immersed interface method based on finite differences for clean drops, however they required both the drops and the bulk to have the same viscosity. This constraint still remained as Xu et al. [44] extended the method in [24] to simulate surfactant-covered droplets in Stokes flow. Care has to be taken when imposing the jump condition to avoid a time step constraint that is dependent on how the interfaces cut the underlying grid. This is thoroughly discussed for finite element methods in Hansbo et al. [12], where they manage to design a second order method and avoid such a constraint by adding suitable stabilising terms.

Reformulating the Stokes equations in integral form on the droplet interfaces avoids the aforementioned issues. This reduces the dimension of the problem as only the interfaces have to be discretised and gives an explicit representation of the interface which does not have to be coupled to an underlying grid. Furthermore, it naturally handles the discontinuity in normal stress. A review of boundary integral methods for Stokes flow is given in [31]. For drops, papers containing boundary integral methods including surfactants in 3D are among others [3,32,38]. In two dimensions, boundary integral methods have been used for e.g. vesicles as by Marple et al. [25] who simulated vesicle suspensions in confined flows and Quaife and Biros [34] who focused on developing an adaptive time-stepping scheme using an implicit spectral deferred correction method. This was extended by Bystricky et al. [5] for rigid body suspensions. Kropinski and Lushi [22] simulated surfactant-covered bubbles with a boundary integral method and used a spectral method to compute the surfactant concentration. Xu et al. [46] extended this work to include a model for soluble surfactants in the no-diffusion limit in the bulk. However, the numerical solution of the boundary integral equations has its own challenges. Primarily, the near-singular behaviour of the discretised integrals when evaluating interfaces in close proximity. In 2D, Ojala and Tornberg [28] used a special quadrature scheme to ensure high accuracy also for close to touching clean drops. Other options for closely interacting drops includes quadrature by expansion (QBX) by Klöckner et al. [20], which can be extended to 3D, and was done so efficiently for the special case of solid spheroids in [17].

This paper proposes an efficient and accurate boundary integral method for the simulation of deforming bubbles and droplets in two dimensional Stokes flow; for both clean and surfactant-covered droplets. The method is an extension of that proposed by Kropinski and Lushi [22] and uses the special quadrature method in [28] to enable accurate simulations of close drop-drop interactions. To match the spectral accuracy in space, an adaptive time-integration scheme is employed and also described herein.

This paper also aims to validate the proposed numerical method using conformal mapping theory following the approaches in [8,41]. For single surfactant-covered bubbles in an extensional flow at steady state, exact solutions are computed by Siegel [37]. Furthermore, Crowdy et al. [8] used conformal mapping theory to semi-analytically compute the deformation of a pair of clean bubbles in an extensional flow. With this approach, a conformal mapping of the interface is considered and the time-dependent problem of deforming bubbles is rewritten into a system of ODEs for the conformal mapping parameters, whilst the flow field can be described analytically. In this paper, this approach has been extended to surfactant-covered bubbles. Together, these solutions enable a precise and quantitative validation of the numerical method proposed by this paper. A comprehensive evaluation of the method is undertaken, examining temporal development, droplets in close proximity and the distribution of surfactants.

Additionally, the errors introduced when evaluating the near-singular integrals numerically using a standard Gauss-Legendre quadrature are estimated. This work extends that of af Klinteberg and Tornberg [18,19] where estimates based on contour integrals were derived for Laplace's and Helmholtz equations. The estimates are in excellent agreement with the observed numerical errors. Details of the estimates and their derivation can be found in Appendix A.

The paper is organised as follows: in §2 the model, governing equations and boundary integral formulation are introduced. §3 explains the numerical solution of the boundary integral equation, the special quadrature method and provides results for the error estimates. The complete numerical method for the coupled problem is explained in §4 and §5 details the methods used for validation. In §6 numerical results for drop simulation and validation are presented.

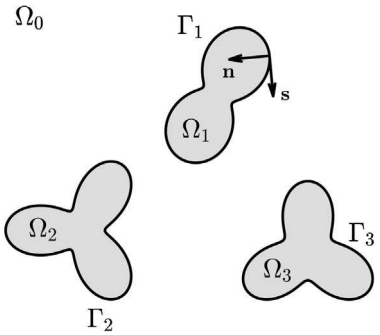


Fig. 1. Example of a domain consisting of three droplets.

2. Problem formulation

2.1. Governing equations

On the micro scale, the fluid velocity $\mathbf{u}$ and the pressure p of a flow can be computed by solving the Stokes equations. Here, an infinite expanse of fluid, Ω_0, with viscosity μ_0 is considered. The fluid contains n droplets, denoted by Ω_k, for $k = 1, \ldots, n$, each with viscosity μ_k. The interface between the droplet Ω_k and the bulk fluid Ω_0 is denoted by Γ_k, see Fig. 1 for an example domain. The Stokes equations take the form

$$\begin{aligned} \mu_0 \Delta \mathbf{u}_0 &= \nabla p_0, \text{ for } \mathbf{x} \in \Omega_0, \\ \mu_k \Delta \mathbf{u}_k &= \nabla p_k, \text{ for } \mathbf{x} \in \Omega_k,\ k = 1, \ldots, n. \end{aligned} \tag{1}$$

For droplets, $\mu_k \neq 0$ and the velocity is continuous over the interfaces. Also, the solution fulfils the normal stress balance over the interfaces,

$$-(p_0 - p_k)\,\mathbf{n} + 2\,(\mu_0 \mathbf{e}_0 - \mu_k \mathbf{e}_k) \cdot \mathbf{n} = -\sigma \kappa \mathbf{n} + \nabla_s \sigma \text{ on } \Gamma, \tag{2}$$

where $\Gamma = \bigcup_{k=1}^{n} \Gamma_k$. Here, $\mathbf{n}$ is the inward facing normal, σ the surface tension coefficient, $\kappa = \nabla_s \cdot \mathbf{n}$ the curvature and $\mathbf{e}$ the rate of strain tensor. Note that this curvature is negative for a circular drop. The term $\nabla_s \sigma$, where ∇_s is the surface gradient operator, gives the tangential stress (Marangoni force) which is the result of a non-uniform surface tension. The interfaces are discretised clockwise by $s \in [0, L_k(t)]$, where $L_k(t)$ is the length of the interface Γ_k at time t. This condition states that the jump in normal stress over an interface is proportional to the curvature and the Marangoni force. Furthermore, the kinematic interface condition states that the normal velocity of the interface, $\frac{d\mathbf{x}}{dt} \cdot \mathbf{n}$, is equal to the normal fluid velocity,

$$\frac{d\mathbf{x}}{dt} \cdot \mathbf{n} = \mathbf{u} \cdot \mathbf{n}. \tag{3}$$

The limit when $\mu_k = 0$ corresponds to the study of inviscid bubbles. The continuity condition of velocity across the interface can then be disregarded, as there is no velocity inside the bubbles. The kinematic condition (3) is still necessary, as is the normal stress balance (2) modified to contain only pressure and strain tensor of the bulk fluid.

As the fluid domain Ω_0 is unbounded, an additional condition at infinity is needed,

$$\mathbf{u} \to \mathbf{u}_\infty, \text{ as } |\mathbf{x}| \to \infty.$$

Here it is typical to impose a linear far-field flow, where

$$\mathbf{u}_\infty = \begin{pmatrix} Q_0 & B_0 + \frac{G_0}{2} \\ B_0 - \frac{G_0}{2} & -Q_0 \end{pmatrix} \cdot \mathbf{x}. \tag{4}$$

The two imposed linear flows considered in this paper are: *extensional flow*, where $B_0 = G_0 = 0$ and *shear flow*, where $Q_0 = 0$ and $G_0 = 2B_0$.

The surfactants considered here are insoluble, i.e. they exist only on the interfaces of the droplets. Their concentration, $\rho(s, t)$, is described by the convection-diffusion equation [39]

$$\frac{D\rho}{Dt} + \rho\,(\nabla_s \cdot \mathbf{u}) = D_\Gamma \nabla_s^2 \rho, \text{ for } \mathbf{x} \in \Gamma. \tag{5}$$

Here, D_Γ is the diffusion coefficient along the interface and $\frac{D}{Dt}$ the material derivative. As the surfactants are insoluble, their mass on each interface is conserved,

$$\frac{d}{dt}\int_{\Gamma_k(t)} \rho(t)dS = 0, \quad k = 1, \ldots, n. \tag{6}$$

The surfactant concentration and the surface tension at an interface are coupled through an equation of state. Two common equations of states are the Langmuir equation of state,

$$\sigma(\rho) = \sigma_0 + RT\log(\rho_\infty - \rho) \tag{7}$$

and the linearised equation of state

$$\sigma(\rho) = \sigma_0 - RT\rho, \tag{8}$$

as described in [29]. Here, R is the universal gas constant, T the temperature and σ_0 the surface tension coefficient of a clean interface. Furthermore, ρ_∞ is the maximum monolayer packing concentration of surfactants on $\Gamma(t)$. Generally, the Langmuir equation of state is considered more accurate. The linear equation of state is mostly used for problems with low saturation levels, but is also commonly used in many validation cases, which is the reason it is included here. To switch between the two is trivial.

2.2. Nondimensionalisation

The above equations are nondimensionalised using a characteristic length r_0, defined as the radius of an initial droplet. The characteristic velocity is defined as the ratio $\frac{\sigma_0}{\mu_0}$, where μ_0 is the viscosity of the bulk and σ_0 the surface tension coefficient of a clean interface. Furthermore the characteristic pressure is defined as $\frac{\sigma_0}{r_0}$. For the remainder of this paper, all physical quantities are considered in their non-dimensional form.

Rewriting the equations, (1) becomes

$$\begin{aligned} \Delta\mathbf{u}_0 &= \nabla p_0, \text{ for } \mathbf{x} \in \Omega_0, \\ \lambda_k \Delta\mathbf{u}_k &= \nabla p_k, \text{ for } \mathbf{x} \in \Omega_k,\ k = 1, \ldots, n, \end{aligned} \tag{9}$$

and (2)

$$-(p_0 - p_k)\mathbf{n} + 2(\mathbf{e}_0 - \lambda_k \mathbf{e}_k)\cdot\mathbf{n} = -\sigma\kappa\mathbf{n} + \nabla_s\sigma, \text{ on } \Gamma. \tag{10}$$

Here, $\lambda_k = \frac{\mu_k}{\mu_0}$ is the viscosity ratio between the droplet k and the bulk fluid. Also the imposed far-field flow is nondimensionalised, such that

$$(Q,\ G,\ B) = \frac{r_0\mu_0}{\sigma_0}(Q_0,\ G_0,\ B_0).$$

The nondimensionalisation of the surfactant concentration depends on which equation of state is used:

Case 1. In the case of the Langmuir equation of state, it is natural to nondimensionalise ρ with the maximum monolayer packing concentration ρ_∞, and (7) reads

$$\sigma = 1 + E\log(1 - \rho), \tag{11}$$

where $E = \frac{RT\rho_\infty}{\sigma_0}$ is the so-called elasticity number.

Case 2. When considering the linearised equation of state, it is instead common to nondimensionalise with the initial surfactant concentration, ρ_0. Then, (8) becomes

$$\sigma = 1 - E\rho, \tag{12}$$

with $E = \frac{RT\rho_0}{\sigma_0}$.

In both cases, (5) becomes

$$\frac{D\rho}{Dt} + \rho(\nabla_s\cdot\mathbf{u}) = \frac{1}{Pe_\Gamma}\nabla_s^2\rho, \text{ for } \mathbf{x} \in \Gamma, \tag{13}$$

where $Pe_\Gamma = \frac{\mu_0 r_0}{\sigma_0 D_\Gamma}$ is the Peclet number.

In summary, to evolve the deforming interfaces of the surfactant-covered droplets in time, the following coupled system needs to be solved for $\mathbf{x} \in \Gamma(t)$:

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{u}(\mathbf{x}, \rho, t), \\ \frac{D\rho}{Dt} &= -\rho\,(\nabla_s \cdot \mathbf{u}) + \frac{1}{Pe_\Gamma}\nabla_s^2\,\rho. \end{aligned} \tag{14}$$

The solution to this system will be explained in steps. First, the boundary integral equation (BIE) formulation for $\mathbf{u}$ is given in §2.3. In §3 the numerical method that is used to accurately solve the BIE is explained. In §4 the numerical procedure for the solution of the surfactant concentration, the coupling of the system and its evolution in time is described.

2.3. Boundary integral formulation

In two dimensions, it is convenient to regard this problem in the complex plane. The spatial variable $\mathbf{x}$ then corresponds to the complex variable $z = x + iy$. On the interface, z is regarded as a function of the parameter α and time, $z(\alpha, t)$, where $\alpha \in [0, 2\pi]$. The two parametrisations s and α are linked through $\alpha = 2\pi s/L_k(t)$ for each interface $\Gamma_k(t)$, where s was introduced below (2).

The integral equation used to compute the velocity on each interface is obtained through the Sherman-Lauricella formulation as described in [21]. This approach stems from the fact that the solution to the Stokes equations in two dimensions, $u(z) = u_1(x, y) + iu_2(x, y)$, can be written as the solution to the biharmonic equation. This is done by introducing a stream function $W(x, y)$ where $u_1 = W_y$ and $u_2 = W_x$. One can then replace the Stokes equations by the biharmonic equation $\Delta^2 W = 0$. By Goursat's formula, $W(x, y)$ has the representation

$$W(x, y) = \Re\left\{\overline{z} f(z) + h'(z)\right\},$$

where $f(z)$ and $h(z)$ are analytic functions on Ω_0. The functions f and h are known as Goursat functions, and all physical quantities of the problem, such as velocity, pressure and vorticity, can be expressed as combinations of them [23]. Using the Sherman-Lauricella formulation as in [21], the Goursat functions are written as Cauchy integrals of a complex-valued density $\mu(z)$. By substituting velocity and pressure in the interface condition (10) by their representations of Goursat functions, a Fredholm equation of the second kind is obtained to solve for $\mu(z)$ on the interfaces $\Gamma(t)$,

$$\begin{aligned} &\mu(z) + \beta(z)\mathcal{T}(z, \mu) + \beta(z)\int_{\Gamma(t)} \mu(\tau)|d\tau| \\ &\quad = -\sigma(\alpha, t)\frac{\gamma(z)}{2}\frac{\partial z}{\partial \alpha} - \beta(z)\,(B - iQ)\,\overline{z}, \quad z \in \Gamma(t), \end{aligned} \tag{15}$$

where $\mathcal{T}(z, \mu)$ is the complex variable formulation of the stresslet

$$\mathcal{T}(z, \mu) = \frac{1}{\pi}\int_{\Gamma(t)} \mu(\tau)\Im\left(\frac{d\tau}{\tau - z}\right) + \frac{1}{\pi}\int_{\Gamma(t)} \overline{\mu(\tau)}\frac{\Im\left(d\tau\,(\overline{\tau} - \overline{z})\right)}{(\overline{\tau} - \overline{z})^2},$$

see [21] for details. The third term in (15) evaluates to zero as a result of the area conservation of the droplets, and is used to remove rank deficiency in the case of inviscid bubbles. Here, notation as in [28] is used, where $\beta(z) := \frac{1-\lambda_k}{1+\lambda_k}$ and $\gamma(z) := \frac{1}{1+\lambda_k}$ for $z \in \Gamma_k(t)$. In the last term on the right hand side, B and Q are part of the imposed far-field flow in (4). Furthermore, $\sigma(\alpha, t) := \sigma(\rho(\alpha, t))$ is the surface tension coefficient obtained from the surfactant concentration through either equation of state, (11) or (12). Note that the expression in (10) has been integrated once, thus no differentiation of $\sigma(\alpha, t)$ is needed.

Once $\mu(z)$ is obtained for all $z \in \Gamma(t)$, the velocity can be evaluated for $z \in \Omega_0 \cup \Gamma \cup \Omega_k$, as

$$\begin{aligned} u(z) = u_1(z) + iu_2(z) &= -\frac{1}{\pi}\int_{\Gamma(t)} \mu(\tau)\Re\left(\frac{d\tau}{\tau - z}\right) \\ &\quad - \frac{1}{\pi i}\int_{\Gamma(t)} \overline{\mu(\tau)}\frac{\Im\left(d\tau\,(\overline{\tau} - \overline{z})\right)}{(\overline{\tau} - \overline{z})^2} + (Q + iB)\overline{z} - \frac{iG}{2}z, \end{aligned} \tag{16}$$

where the last two terms on the right hand side represent the far-field velocity $\mathbf{u}_\infty$ in complex form. Note that the first integral in (16) is singular and interpreted in a principal value sense.

3. Solving the BIE numerically

Here, a summary of the numerical method to solve the BIE is given. More details are available in [28]. The solution procedure described is valid for any instance of time $\bar{t}$.

To compute the integrals (15) and (16) accurately, a high order discretisation of $\Gamma(\bar{t})$ is needed. Here, an explicit representation of the interfaces is used; $z_k(\alpha)$, for droplets $k = 1, \dots, n$, where $\alpha \in [0, 2\pi)$. On the interface k, each interface is split into panels and N_k discretisation points are placed on a composite 16-point Gauss-Legendre grid. The total number of discretisation points becomes $N = \sum_{k=1}^{n} N_k$.

3.1. Solve for $\mu(z)$

To compute $\mu_i \approx \mu(z_i)$, where z_i are the Gauss-Legendre points on the interfaces at time $\bar{t}$, a Nyström method is used. The discretised version of (15) is

$$\begin{aligned} \mu_i + \frac{\beta_i}{\pi} \sum_{j=1}^{N} \mu_j M_{ij}^{(1)} + \frac{\beta_i}{\pi} \sum_{j=1}^{N} \overline{\mu_j} M_{ij}^{(2)} + \beta_i \sum_{j=1}^{N} \mu_j w_j |z_j'| \\ = -\sigma_i \frac{\gamma_i}{2} \frac{z_i'}{|z_i'|} - \beta_i (B - iQ) \overline{z_i}, \ i = 1, \dots, N, \end{aligned} \tag{17}$$

where $z_i' = \frac{\partial z}{\partial \alpha}\big|_i$. Here, w_j are the Gauss-Legendre weights associated with z_j. Also, $\sigma_i = \sigma(\alpha_i)$, $\beta_i = \beta(z_i)$ and $\gamma_i = \gamma(z_i)$. Moreover,

$$M_{ij}^{(1)} = w_j \Im \left\{ \frac{z_j'}{z_j - z_i} \right\}, \quad M_{ij}^{(2)} = w_j \frac{\Im \left\{ z_j' (\overline{z_j} - \overline{z_i}) \right\}}{(\overline{z_j} - \overline{z_i})^2},$$

for all $j \neq i$. The limits when $j = i$ are finite and given by

$$M_{ii}^{(1)} = w_i \Im \left\{ \frac{z_i''}{2 z_i'} \right\}, \quad M_{ii}^{(2)} = w_i \frac{\Im \left\{ z_i'' \overline{z_i'} \right\}}{2 (\overline{z_i'})^2}.$$

The derivatives of z_i are computed with Fourier differentiation on a uniform grid and evaluated on the Gauss-Legendre grid.

Solving (17) to obtain μ_i, $i = 1, \dots, N$, boils down to solving a system $A\boldsymbol{\mu} = \mathbf{b}$, where $\boldsymbol{\mu} = (\mu_1 \dots \mu_N)^T$, $\mathbf{b}$ represents the right hand side in (17) and $A\boldsymbol{\mu}$ the expression on the left hand side. The matrix A is dense, but as (15) is a Fredholm equation of the second kind, the condition number of A is typically small and does not increase with grid refinement. Due to the spectral properties of the discretised system, there is no need for a preconditioner when solving (17) with an iterative algorithm such as `GMRES`. Furthermore, the matrix-vector multiplications are sped up using the fast multipole method, `FMM` [10].

3.2. Compute $u(z)$

Once μ_i is obtained for all discretisation points z_i, the velocity $u_i \approx u(z_i)$ can be computed through (16). To handle the singular integral, singularity subtraction is used; when evaluating u_i the integral

$$\frac{1}{\pi} \int_{\Gamma} \mu_i \Re \left\{ \frac{d\tau}{\tau - z_i} \right\}$$

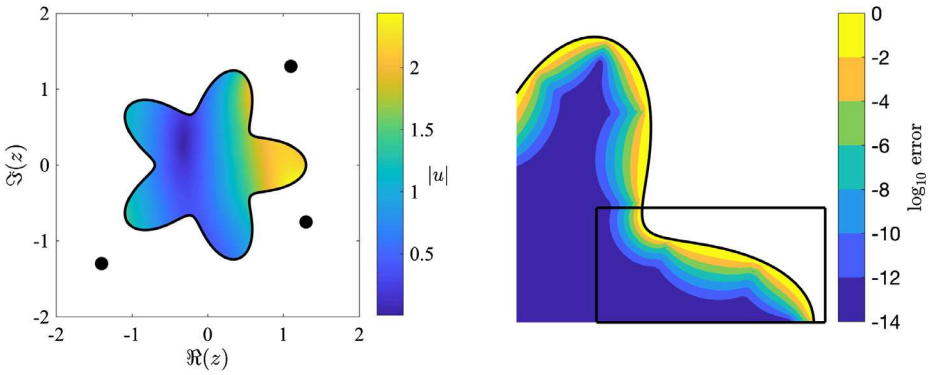
is added and subtracted from the expression in (16). Using calculus of residues this removes the singularity and leads to the discretised expression

$$\begin{aligned} u_i = -\frac{w_i}{\pi} \mu_i' - \frac{1}{\pi} \sum_{j=1, i \neq j}^{N} (\mu_j - \mu_i) w_j \Re \left\{ \frac{z_j'}{z_j - z_i} \right\} \\ - \frac{1}{i\pi} \sum_{j=1}^{N} \overline{\mu_j} M_{ij}^{(2)} + (Q + iB) \overline{z_i} - \frac{iG}{2} z_i, \end{aligned} \tag{18}$$

where $M_{ij}^{(2)}$ is defined as previously.

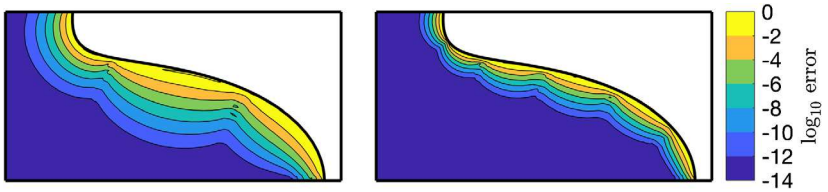
3.3. Error estimates

Regarding the equations (15) and (16), they both contain terms $(z_j - z_i)^{-1}$. For $M^{(1)}$ and $M^{(2)}$ the limits when $i = j$ are well-defined, and for the term $\Re \left\{ \frac{z_j'}{z_j - z_i} \right\}$ singularity subtraction is used, see (18). However, when droplets get close to each



(a) Domain Ω, source points and solution $u(z)$. (b) Error in the first quadrant of domain.

Fig. 2. Example of an analytical solution $u(z)$ and the error to (A.2) given sources (x_1, f_1), (x_2, f_2) and (x_3, f_3) (black markers) in a domain Ω. The solution is computed with a composite 16-point Gauss-Legendre quadrature with 25 panels. The black box shows the region displayed in Fig. 3.



(a) 25 panels (b) 50 panels

Fig. 3. Close up of domain marked in Fig. 2(b). Computed error estimates with different number of Gauss-Legendre panels in black for error levels 10^{-p}, where $p = 14, 12, \ldots, 2, 0$. Measured errors in colour. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

other, i.e. when $\|z_i - z_j\| \ll 1$ for points z_i and z_j on different interfaces, the integrals are said to become near-singular. Analytically this is not an issue, but numerically large errors are introduced as the integrand gets harder to resolve.

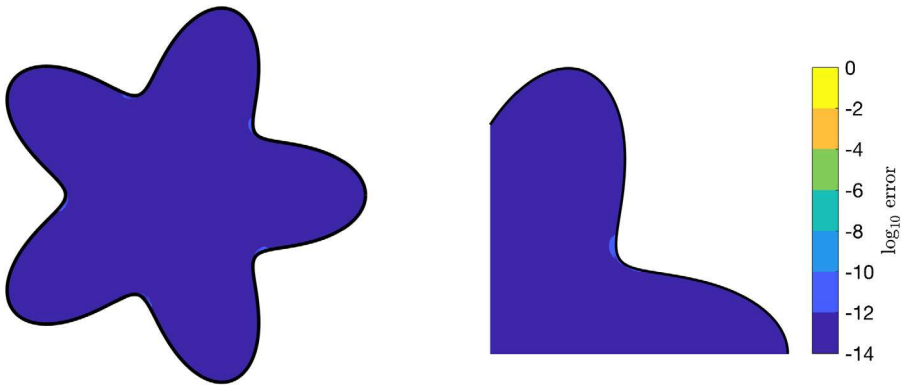
To demonstrate how these errors grow, the Stokes equations are solved in a fixed domain with Dirichlet boundary conditions. This problem contains the same near-singular behaviour as (16). Given any analytical solution to the Stokes equations (for example, that generated by a point source forcing as shown in Fig. 2(a)), the Dirichlet velocity data can be evaluated at the boundary and used as boundary conditions for the numerical method. The boundary integral formulation and exact problem setting is described in Appendix A, the domain and the solution $u(z)$ are shown in Fig. 2(a).

The relative error of $u(z)$ compared to the exact solution, using 25 Gauss-Legendre panels, is shown in Fig. 2(b). Here only the first quadrant is shown, as the error behaves identical in the other three. It is clear that at an evaluation point z_0 close to the boundary, the errors are large and other treatments of the integrals are needed for these cases. In this paper a special quadrature is used, which is explained in §3.4. At what distance from a panel the errors become large depends on the refinement of the discretisation. In order to know when special treatment is needed, the quadrature errors of such near-singular integrals can be estimated. This was originally done by af Klinteberg and Tornberg, where estimates for the quadrature errors for Laplace's and Helmholtz equations were derived in [18,19]. Using the same approach based on contour integration and calculus of residues for integrals of the type appearing in (16), the errors when computing $u(z)$ can be estimated also for the Stokes equations. Details of this can be found in Appendix A. The estimates follow the error levels remarkably well, as is shown in Fig. 3. Both the errors and the estimates for 25 and 50 panels are shown, and it is clear that refining the interface discretisation makes the region of large errors narrower, but will not eliminate it.

3.4. Special quadrature

To improve the accuracy of the computation of $\mu(z, \bar{t})$ and $u(z, \bar{t})$ for a z near any drop interface the special quadrature method of [28] will be employed. It was originally introduced by Helsing and Ojala [13] for Laplace's equation and extended to Stokes equations in [28]. This is a local method which regards point-panel pairs. For each evaluation point the error when using a standard Gauss-Legendre quadrature is estimated on each panel. If it is too large over a specific panel, Γ_i, the integral over that panel will instead be treated semi-analytically. An overview of the special quadrature can be found in Appendix B.

Solving the same problem for the Stokes equations in a domain with non-deforming boundaries as in §3.3, the $u(z)$ obtained by standard quadrature is corrected in the regions where the error is large. The reduction of error when using the special quadrature can be seen in Fig. 4. This error should be compared to that in Fig. 2(b), where standard 16-point



(a) Error in whole domain. (b) Error in first quadrant.

Fig. 4. Logarithm of the pointwise error when solving the Stokes equations with the special quadrature. The special quadrature is used for points close to the boundary, cf. Fig. 2(b) for a comparison of errors. The solution is computed with 50 Gauss-Legendre panels.

composite Gauss-Legendre quadrature is used. It is clear that errors can be kept at a very low level of order 10^{-10} or less also for evaluation points close to the interface.

4. Numerical method in time

Several steps are needed to simulate the deformation of surfactant-covered droplets through the system in (14). Here, a method of lines approach is used, where the discretisation in space generates a system of ODEs to solve in time. This system can then be solved using a numerical method for initial value problems using a time-integration scheme as described in §4.5.

To evolve the interfaces in time, as well as to compute the surfactant concentration, an equidistant spatial discretisation is used. However, to compute the integrals in the BIE formulation described in §3, it is beneficial to use a composite Gauss-Legendre discretisation. Thus, a hybrid method using both discretisations is employed. The interface of each droplet is discretised with N_k points. For the uniform discretisation, these points are placed equispaced in arc length. The quadrature weights associated with this discretisation are the standard trapezoidal weights. For the composite Gauss-Legendre discretisation, the interface is divided into $N_k/16$ panels of equal length, each discretised with a 16-point Gauss-Legendre quadrature rule. To go between the two discretisations, the points on each panel are interpolated.

Here follows a short overview of the steps needed at each instance in time, all of which will be explained in more detail further on in §4.1-4.4. In order to evolve the droplets in time, the interface velocities are obtained through the following steps:

1. Upsample the uniform discretisation and interpolate to the composite 16-point Gauss-Legendre grid.
2. Compute the fluid velocity at the boundaries accurately by solving the BIE and applying the special quadrature where needed, see §3.
3. Interpolate the velocity back to the uniform grid and downsample.
4. Compute the appropriate tangential velocity to keep the discretisation points equidistant in arc length.

The velocity is then fed into a time-integration scheme for propagation of the droplet interfaces in time. Each time the fluid velocity is computed, the surface tension coefficient is needed. Since this is dependent on the surfactant concentration through (11) or (12), also the surfactant concentration needs to be propagated in time through (13), see §4.4.

4.1. Interpolation to and from the composite Gauss-Legendre grid

First, upsampling is needed for stability and is achieved by zero-padding the Fourier coefficients of the equidistant discretisation with a factor two. Then, the interpolation to the composite Gauss-Legendre grid is done via a non-uniform `FFT` (`nuFFT`), see Greengard and Lee [9] for details. Once the velocity and the special quadrature corrections have been computed, a 16-point Gauss-Legendre interpolation on each panel is used to go back to the uniform discretisation. Finally, the velocity vector is downsampled and a Krasny filter is applied to the velocity vector, where all Fourier modes below the absolute value of 10^{-12} are set to zero. See [28] for more details.

4.2. Modifying the tangential velocity

With the velocity at the interfaces represented on the equidistant discretisation, denoted by u, a new modified velocity for the discretisation points is computed, $\frac{d\mathbf{x}}{dt}$. The reason for this is to avoid clustering of the discretisation points as the

interface evolves in time, an idea which was originally introduced by Hou et al. [14] and then used to simulate bubbles by Kropinski [21]. The idea is to decompose the velocity $u(z(s))$ into its normal and tangential components,

$$u(z(s)) = [u_n(s) + iu_t(s)]n(s),$$

where $n(s)$ is the complex representation of the normal, $s \in [0, L_k(t)]$ and u_n and u_s are real valued. The modified velocity is then defined as

$$\frac{d\mathbf{x}}{dt}(s) = \left[u_n(s) + i\tilde{u}_t(s)\right]n(s). \tag{19}$$

The new tangential velocity $\tilde{u}_t$ can be chosen to dynamically preserve s equidistant in arc length. It can be derived as (see [21])

$$\tilde{u}_t(s) = \frac{s}{2\pi}\int_0^{2\pi} \Im\left\{\frac{z''(q)}{z'(q)}\right\} u_n(q)dq - \int_0^{s} \Im\left\{\frac{z''(q)}{z'(q)}\right\} u_n(q)dq,$$

which simplifies to an antiderivative taken with an FFT. The notation of this tangential velocity is adapted from [28]. Note that the modified velocity $\frac{d\mathbf{x}}{dt}$ still fulfils the kinematic condition (3).

4.3. Spatial adaptivity

Using the modified tangential velocity as explained above keeps the interface discretisation points equidistant in arc length over time, i.e. with equal spacing Δs. As the droplets deform, the spacing Δs might increase or decrease. To keep the spatial resolution constant, a spatial adaptivity is implemented where more discretisation points are added to the interfaces if Δs grows more than a set threshold. Likewise, in the case when Δs becomes too small, discretisation points are removed. These operations are both performed on the equidistant discretisation.

4.4. Solving the surfactant equation

The material derivative in (13) can be expanded utilising the interface velocity $u(z(s))$ and the velocity of the discretisation points $\frac{d\mathbf{x}}{dt}(s)$, $s \in [0, L_k(t)]$). The convection-diffusion equation for the surfactant concentration becomes

$$\frac{\partial \rho}{\partial t} + \left(\mathbf{u} - \frac{d\mathbf{x}}{dt}\right)\cdot \nabla_s \rho + \rho \nabla_s \cdot \mathbf{u} = \frac{1}{Pe_\Gamma}\nabla_s^2 \rho, \ \mathbf{x} \in \Gamma. \tag{20}$$

Using that the normal components of $\frac{d\mathbf{x}}{dt}$ and u are the same and that the surface gradient operator can be written as $\nabla_s(\cdot) = \mathbf{s}\frac{\partial}{\partial s}(\cdot)$, (20) can for each droplet k be rewritten as

$$\frac{\partial \rho}{\partial t} = \tilde{u}_t \frac{\partial \rho}{\partial s} - \frac{\partial(\rho u_t)}{\partial s} - \rho u_n \kappa + \frac{1}{Pe_\Gamma}\frac{\partial^2 \rho}{\partial s^2}, \ s \in [0, L_k(t)]. \tag{21}$$

Here, the first two terms in the right hand side correspond to advection, the third to stretching and the last term represents diffusion along the interface. To reformulate (21) in terms of $\alpha \in [0, 2\pi)$, the equal-arc length parameter $s_\alpha^k(t)$ is defined as

$$s_\alpha^k(\alpha, t) = \frac{1}{2\pi}L_k(t), \tag{22}$$

see [22]. The convection-diffusion equation becomes

$$\frac{\partial \rho}{\partial t} = \frac{\tilde{u}_t}{s_\alpha^k(t)}\frac{\partial \rho}{\partial \alpha} - \frac{1}{s_\alpha^k(t)}\frac{\partial(\rho u_t)}{\partial \alpha} - \rho u_n \kappa + \frac{1}{Pe_\Gamma\, s_\alpha^k(t)^2}\frac{\partial^2 \rho}{\partial \alpha^2}, \ \alpha \in [0, 2\pi), \tag{23}$$

for all droplets $k = 1, \ldots, n$.

Using the method of lines approach, the right hand side in (23) is computed whilst keeping the time t continuous. Define $\rho_i^k(t) \approx \rho(\alpha_i, t)$, where $\alpha_i \in [0, 2\pi)$ are the parameter discretisation points for the equidistant discretisation on the interface, $z_i^k = z(\alpha_i, t)$, for the droplet k. For each droplet, the PDE can easily be reduced to a system of ODEs for the Fourier coefficients of $\rho_i^k(t)$, $i = 1, \ldots, N_k$, obtained by FFTs. In the following only the case of one droplet will be regarded, $k = n = 1$, but the method extends trivially to multiple droplets.

It is advantageous to treat the convection and stretching terms explicitly, whilst the diffusion term should be treated implicitly to avoid stiffness. Thus, (23) becomes

$$\frac{\partial \rho_i}{\partial t} = f_E(z_i, \rho_i, t) + f_I(z_i, \rho_i, t), \ i = 1, \ldots, N,$$

Table 1
Butcher tableau, adaptive method using explicit midpoint and explicit Euler for interface position z.

0		
1/2	1/2	
	0	1
	1	0

where the terms have been collected for explicit and implicit treatment respectively,

$$f_E(z_i,\rho_i,t) = \frac{\tilde{u}_t}{s_\alpha^k(t)}\frac{\partial\rho}{\partial\alpha} - \frac{1}{s_\alpha^k(t)}\frac{\partial(\rho u_t)}{\partial\alpha} - \rho u_n\kappa,$$
$$f_I(z_i,\rho_i,t) = \frac{1}{Pe_\Gamma\, s_\alpha^k(t)^2}\frac{\partial^2\rho}{\partial\alpha^2}.$$

The corresponding system of ODEs in Fourier space is

$$\frac{\partial\widehat{\rho}_j}{\partial t} = (\widehat{f_E})_j + (\widehat{f_I})_j,\ j = -\frac{M}{2},\dots,\frac{M}{2}-1,$$

where $\widehat{(\cdot)}_j$ denotes the Fourier coefficient of mode j. For a droplet in two dimensions, the interface discretisation is one dimensional. In one dimension, the implicit term, $\widehat{(f_I)}_j$, is trivial to compute as the second derivative corresponds to a multiplication of $-j^2$ for each Fourier mode. The parts in $\widehat{(f_E)}_j$ are computed pseudo-spectrally. To avoid aliasing the discretisations on all interfaces are upsampled by factor $\frac{3}{2}$, i.e. $M = \frac{3}{2}N$ and a Krasny filter is applied, setting all Fourier modes below the absolute value of 10^{-12} to zero.

4.5. Adaptive time-stepping

Including the modified tangential velocity and the pseudo-spectral method for the surfactant concentration, the system (14) that needs to be evolved in time can be rewritten as

$$\begin{aligned}\left.\frac{dz}{dt}\right|_l &= \left(u_n + i\tilde{u}_t\right)_l n_l = g(z_l,\sigma_l),\ l = 1,\dots,N\\ \frac{\partial\widehat{\rho}_j}{\partial t} &= (\widehat{f_E})_j + (\widehat{f_I})_j,\ j = -\frac{M}{2},\dots,\frac{M}{2}-1,\end{aligned} \tag{24}$$

for each droplet. The system is coupled, as the surface tension coefficient σ_i depends on ρ_i through some equation of state, e.g. (11) or (12).

As mentioned in §4.4, the diffusion term in (23) needs to be treated implicitly for stability whilst the other terms in the right hand side preferably are treated explicitly. The evolution equation for the interface should be treated explicitly. Furthermore, adaptivity in time is required in order to handle multiple droplets getting very close to each other.

A range of potential time-stepping schemes for this problem were studied in [33]. Unlike the case for vesicles [34], for deforming droplets the time step has a restriction of order one, which enables larger freedom when selecting a suitable time-integration scheme. The design criteria was an accurate but efficient scheme, which would keep the number of solutions of the Stokes equations at a minimum as it is the most costly part of the method. The conclusion was that a combination of an explicit midpoint method for z together with an implicit-explicit Runge-Kutta of order two for ρ together with adaptivity was advantageous to use. The authors observed that the adaptive time step should be adjusted considering the errors both in interface position z and surfactant concentration ρ. An explanation of each part of the scheme follows:

Interface position. The explicit method used to solve

$$\frac{dz}{dt} = g(z,\sigma),$$

is the standard Runge-Kutta explicit midpoint rule. At each time step, the local error in z^{n+1}, r_z^{n+1}, is computed by comparing z^{n+1} with that of a lower order method, $\tilde{z}^{n+1}$. The scheme in full is shown in the Butcher tableau in Table 1. The local error for the time step t_n to t_{n+1} is computed as

$$r_z^{n+1} = \frac{\|z^{n+1} - \tilde{z}^{n+1}\|_\infty}{\|z^{n+1}\|_\infty}.$$

Table 2
Butcher tableau, IMEX2 scheme for ρ.

0			0		
1/2		1/2	1/2	1/2	
	0	1		0	1

Surfactant concentration. The implicit-explicit Runge-Kutta of order two used here (IMEX2) is described by its Butcher tableau in Table 2 and explained in detail in [2]. Adaptivity in ρ can be achieved in two possible ways, either by comparing to a lower order IMEX scheme or by using the concept of mass conservation from (6). Here, adaptivity based on mass conservation is considered. The time-discretised version of the mass conservation reads

$$\int_{\Gamma^{n+1}} \rho^{n+1} dS - \int_{\Gamma^n} \rho^n dS = 0,$$

where the integrals can be computed with spectral accuracy using the trapezoidal rule. To measure the local error in ρ of the time step from t_n to t_{n+1}, regard

$$r_\rho^{n+1} = \frac{\| \int_{\Gamma^{n+1}} \rho^{n+1} dS - \int_{\Gamma^n} \rho^n dS \|_\infty}{\| \int_{\Gamma^n} \rho^n dS \|_\infty}.$$

Coupled method. The Runge-Kutta stages of the methods for z and ρ are the same. This means that at every stage, i.e. at t_n and $t_{n+1/2}$, information between the equations can be exchanged. This coupling, together with the two second order methods, give a scheme which is second order in total. However, the most important feature is the combined adaptivity. The total local error of a time step is computed as the maximum of the local errors in z and ρ,

$$r^{n+1} = \max\left(r_z^{n+1}, r_\rho^{n+1}\right).$$

As is typical in adaptive time-stepping methods, if the local error $r^{n+1} > tol$ for some tolerance tol for a time step of size dt_{old}^n, the time step is retaken with updated size. To meet the tolerance, the new time step to go from t_n to t_{n+1}, denoted dt_{new}^n, needs to fulfil

$$\left(\frac{dt_{new}^n}{dt_{old}^n}\right)^2 \approx \frac{tol}{r^{n+1}}.$$

This gives a new time step of

$$dt_{new}^n = dt_{old}^n \left(0.9 \frac{tol}{r^{n+1}}\right)^{\frac{1}{2}},$$

where 0.9 is a safety factor, [11]. The expected second order convergence rate is confirmed in [33].

5. Validations

To validate the coupled method described in §4, the conformal-mapping techniques of Crowdy et al. [8] and Siegel [37] are used to compute the evolution of surfactant-covered bubbles in an extensional flow for specific cases. The aim of the validation is to test the coupling of the solution for interface position and surfactant concentration as well as the dynamics over time to high accuracy. With this aim, three cases for validation have been selected:

1. the exact solutions of surfactant-covered bubbles and droplets in extensional flow at steady state
2. semi-analytic solutions for a pair of clean, deforming bubbles in extensional flow
3. semi-analytic solutions for a pair of surfactant-covered bubbles deforming in extensional flow.

The first case uses an analytical solution by Siegel and tests the dynamics of surfactant-covered bubbles and droplets over a long time. The second method was introduced by Crowdy et al. and regards a pair of clean bubbles in extensional flow. This case tests especially the special quadrature of §3.4 when the two bubbles are pushed close to each other. The third case builds on the second, with the addition of insoluble surfactants. This tests the time-dependent dynamics of the whole method put together.

In this section, an overview of how to compute the validation results for all three cases is described. Note that case two and three are both described in §5.2, where the former simplifies to no surfactants ρ and a constant surface tension σ.

5.1. Surfactant-covered bubble or drop in steady state

For single surfactant-covered bubbles, analytical solutions to the interface position z and the surfactant concentration ρ exist in certain cases. The case regarded here is that of a bubble in an extensional flow, $\mathbf{u}_\infty = Q(x, -y)$, at *steady state*, for which exact solutions were originally obtained by Siegel [37]. A steady state is defined as the time T when $\mathbf{u} \cdot \mathbf{n} = 0$. These solutions exist for bubbles deforming from an initially circular shape with a uniform surfactant concentration and where there is no diffusion of surfactants along the interface, i.e. $Pe_\Gamma = \infty$. To couple the surfactant concentration to the surface tension, the linear equation of state (12) is used. For details and derivations on how to obtain these solutions, the reader is referred to the original reference. Here, it will be explained how to compute these solutions to use as a validation case. Given a deformation at steady state, i.e. z at time T, these solutions are used to analytically compute ρ at time T as well as the Capillary number Q that is needed to obtain the set deformation at the specified steady state. Using a numerical method to simulate the deformation and the surfactant evolution under this Capillary number, the solution at the steady state can then be compared to the analytical solution to determine the accuracy.

To describe the bubble interface, a conformal mapping is used,

$$z(\zeta, t) = \frac{a(t)}{\zeta} + b(t)\zeta, \tag{25}$$

where $a(t) < 0$ and $b(t)$ is real. Furthermore, the condition of constant area gives $a^2 - b^2 = 1$. At the bubble interface, $|\zeta| = 1$, i.e. $\zeta = e^{i\nu}$ for $\nu \in [0, 2\pi]$. At steady state, it is required that

$$Q = -\frac{Ab}{(1+b^2)^{1/2}}, \tag{26}$$

where

$$A = \frac{\int_0^{2\pi} B(b,\nu)^{1/2} d\nu - 2\pi E}{\int_0^{2\pi} B(b,\nu) d\nu}, \tag{27}$$

with E the elasticity number in (12), and

$$B(b,\nu) = 1 + 2b^2 - 2(1+b^2)^{1/2} b\cos(2\nu). \tag{28}$$

The deformation of a bubble is measured as

$$D = \frac{R_{max} - R_{min}}{R_{max} + R_{min}}, \tag{29}$$

where R_{max}, R_{min} are the maximum and minimum radial distance of a point on the interface to the centre respectively.

To compute the surfactant concentration at steady state, (13) is used. At the steady state, this can be simplified to

$$\rho \Re\left[\frac{u(z) + \overline{z}_\nu}{|z_\nu|}\right] = \frac{1}{Pe_\Gamma}\left(\frac{\rho_\nu}{|z_\nu|}\right) = 0, \tag{30}$$

where the second equality comes from having no diffusion, $Pe_\Gamma = \infty$. This means, that on any part of the interface at the steady state, either $\rho = 0$ or $u(z) = 0$ (no slip condition). Assuming the extensional flow is sufficiently low such that the concentration of surfactants is non-zero everywhere, it fulfils

$$\rho(\nu) = \frac{1 - A|z_\nu|}{E}. \tag{31}$$

An algorithm to compute the steady state surfactant concentration $\rho(\nu)$ and Capillary number Q given the steady state deformation can be found in Algorithm 1. The deformation at steady state is determined through the conformal mapping parameters a and b in (25) which are given as input to the algorithm. How the deformation D depends on the Capillary number Q for different elasticity numbers is shown in Fig. 9.

To validate the results of a numerical method against these exact solutions, the results at steady state are compared when simulating an initially circular bubble with uniform surfactant concentration $\rho_0 = 1$ in an extensional flow with Capillary number Q and $Pe_\Gamma = \infty$, i.e. no diffusion of surfactants.

As was discussed by Milliken et al. in [26], in the case of no diffusion along the interface the viscosity ratio does not affect the steady state deformation. When $Pe_\Gamma = \infty$, the interfacial velocity is dominated by the high Marangoni stresses and the effect of viscosity ratio is therefore negligible. Thus, the results above can be used to validate the steady state deformation and surfactant concentration also for drops, where $\lambda \neq 0$.

Algorithm 1: Steady state computation.

Input : Conformal mapping parameters a and b at steady state.
Output: Deformation D. Capillary number Q for which this deformation is obtained at steady state. Interface position $z(\nu_j)$ and surfactant concentration $\rho(\nu_j)$ at steady state, for $\nu_j = \frac{2\pi j}{M}$, $j \in [0, M-1]$.

1 Discretise $\nu_j = \dfrac{2\pi j}{M}$, $j \in [0, M-1]$.
2 Compute $z(\nu_j)$ through (25).
3 Find R_{max}, R_{min} and compute deformation D in (29).
4 Compute $B(b, \nu_j)$ in (28).
5 Compute A in (27) using trapezoidal rule.
6 Compute $\rho(\nu_j)$ with (31).
7 Compute Q from (26).

5.2. A pair of bubbles in extensional flow

This method of using complex-variable formulations to semi-analytically compute the deformation of a pair of bubbles in an extensional flow was introduced by Crowdy et al. [8]. There, the specific problem setting was a pair of bubbles, $\lambda_k = 0$ for $k = 1, 2$, without surfactants. Here, the method is extended to also include surfactant-covered bubbles, using a linear equation of state (12). The imposed far-field flow is the extensional flow $\mathbf{u}_\infty = Q(x, -y)$, where Q is the Capillary number as previously defined. This method considers the special case when two bubbles are reflectionally symmetric about both x- and y-axes, see Fig. 11 (left).

The method is based on the reformulation of the Stokes equations using complex-variable methods. The bubble boundaries can then be parametrised in terms of a conformal mapping. The numerical algorithm boils down to solving a system of ODEs for the time evolution of the conformal mapping parameters. At each instance of time, the flow field is computed on the unit disc in the conformal-mapping space, using Laurent series.

For a detailed explanation of each of these steps, the reader is referred to [8]. Here, only the numerical approach will be explained in detail. The equations follow those of [8], with two modifications: firstly, the nondimensionalisation has been adjusted to allow for a non-uniform surface tension, σ, along the interface. This is in order to make the extension to surfactant-covered bubbles. Secondly, the convection-diffusion equation for surfactants (13) is coupled to the problem. When considering the original case of clean bubbles, the surface tension is set to be constant and (13) does not need to be solved.

The fluid domain and bubble interfaces are described by the conformal mapping,

$$z(\zeta, t) = \frac{b(t)}{\zeta - \sqrt{\phi(t)}} + \hat{z}(\zeta, t), \tag{32}$$

where

$$\hat{z}(\zeta, t) = \sum_{n=-\infty}^{\infty} a_n(t)\zeta^n. \tag{33}$$

From the symmetry of the problem, it holds that

$$z(\rho/\zeta, t) = -z(\zeta, t),$$

which gives the relations (using (32))

$$a_0(t) = \frac{b(t)}{2\sqrt{\phi(t)}}, \quad a_{-n}(t) = -\phi(t)^n a_n(t),\ n \geq 1, \tag{34}$$

where $0 < \phi(t) < 1$ and $b(t)$ is real. On the bubble interfaces, $\zeta = e^{i\nu}$, for $\nu \in [0, 2\pi]$. In order to also consider the surface tension in the ζ-plane, the composite functions $\sigma(\zeta, t) = \sigma(z(\zeta, t), t)$ for the surface tension and $\rho(\zeta, t) = \rho(z(\zeta, t), t)$ for the surfactant concentration are defined.

In the numerical implementation, all Laurent series expansions are truncated to only include terms $\{\zeta^n, n = -N_V, \ldots, N_V\}$. When zero-padding is needed to avoid aliasing, a factor two is used: giving a new truncation limit $M_V = 2N_V$. The upper bubble interface is discretised by $\zeta_j = e^{i\nu_j}$, for $j = 1, \ldots, 2M_V + 1$, where ν_j is equidistant in $[0, 2\pi)$ with $2M_V + 1$ discretisation points.

5.2.1. Computing the flow field

At a given time $\bar{t}$, the interface position of bubbles, $z(\zeta, \bar{t})$, its conformal mapping parameters $b(\bar{t}) =: b$, $\phi(\bar{t}) =: \phi$, $\{a_n(\bar{t}), n = -N_V, \ldots, N_V\}$ and $\sigma(\zeta, \bar{t})$ are known.

In [8], it is shown that the flow field at ζ at time $\bar{t}$ can be described by functions $F(\zeta,\bar{t})$, $G(\zeta,\bar{t})$ and $C(\bar{t})$. On the upper bubble interface, $|\zeta|=1$, these functions are determined by solving the equation

$$A(\zeta)F_0 + iF(\zeta,\bar{t}) + P(\zeta) - iG(\zeta^{-1},\bar{t}) - iC(\bar{t}) = E(\zeta), \tag{35}$$

where

$$\begin{cases} A(\zeta) = 2\sqrt{\phi}\left(\dfrac{i}{\zeta-\sqrt{\phi}} - \dfrac{iz(\zeta,\bar{t})}{z_\zeta(\zeta^{-1},\bar{t})\left(\zeta^{-1}-\sqrt{\phi}\right)^2}\right), \\ E(\zeta) = \sigma(\zeta,\bar{t})\dfrac{\zeta z_\zeta(\zeta,\bar{t})}{2|z_\zeta(\zeta,\bar{t})|} + \dfrac{iQb}{\zeta^{-1}-\sqrt{\phi}}, \\ P(\zeta) = B(\zeta)F_\zeta(\zeta^{-1},\bar{t}), \ \text{ where } \ B(\zeta) = \dfrac{iz(\zeta,\bar{t})}{z_\zeta(\zeta^{-1},\bar{t})}. \end{cases} \tag{36}$$

Both $F(\zeta,\bar{t})$ and $G(\zeta,\bar{t})$ can be expanded into Laurent series,

$$F(\zeta,\bar{t}) = \sum_{n=-\infty}^{\infty} F_n\zeta^n, \quad G(\zeta,\bar{t}) = \sum_{n=-\infty}^{\infty} G_n\zeta^n,$$

for real coefficients $\{F_n\}$ and $\{G_n\}$.

Expanding (35) into a Laurent series on both sides of the equality sign, and using the orthogonality of $\zeta = e^{i\nu}$, gives $2N_V+1$ equations:

$$A_nF_0 + iF_n + P_n - iG_{-n} - i\delta_nC(\bar{t}) = E_n, \quad n\in[-N_V,N_V], \tag{37}$$

where $\delta_n := 1$ for $n=0$ and 0 otherwise. To compute the flow field at a given time $\bar{t}$, the Laurent coefficients $\{F_n,\ \forall n\in[-N_V,N_V]\}$ and $\{G_n,\ \forall n\in[-N_V,N_V]\}$ need to be determined together with $C(\bar{t})$. The coefficients $\{F_n\}$, $\{G_n\}$ obey

$$F_{-n} = -\phi^nF_n,\ n\geq 1, \text{ and} \tag{38}$$

$$G_{-n} = -\phi^nG_n,\ n\geq 1,\ G_0 = \frac{Qb}{2\sqrt{\phi}}, \tag{39}$$

thus the unknowns are reduced to $\{F_n,\ \forall n\in[0,N_V]\}$, $\{G_n,\ \forall n\in[1,N_V]\}$ and $C(\bar{t})$. Note that since Q is known, G_0 is known for time $\bar{t}$. Therefore, in (37), G_0 should be moved to the right hand side for $n=0$.

The number of unknowns in the problem is thus $2N_V+2$, and the vector of unknowns is

$$\mathbf{x} = \begin{pmatrix} F_0 & F_1 & \dots & F_n & G_1 & \dots & G_n & C(\bar{t}) \end{pmatrix}^T. \tag{40}$$

From (37), $2N_V+1$ equations are obtained. The system is completed with the equation

$$G_\zeta\left(\sqrt{\phi},\bar{t}\right) = \sum_{n=1}^{N_V} 2n\phi^{\frac{n-1}{2}}G_n = Q\,\hat{z}_\zeta\left(\sqrt{\phi},\bar{t}\right), \tag{41}$$

which originates from the presence of an irrotational extensional flow at infinity. As typically $\phi<1$, the expressions above need to be computed using the relations between a_n and a_{-n} (34) as well as between G_n and G_{-n} (39) for stability.

Put together, to compute $\mathbf{x}$ in (40) it is necessary to solve a system $M\mathbf{x}=\mathbf{b}$ of size $2N_V+2$, where $M\mathbf{x}$ corresponds to the left hand side in (37) together with (41), and

$$\mathbf{b} = \begin{pmatrix} E_{-N_V} & \dots & E_{-1} & E_0+iG_0 & E_1 & E_{N_V} & Q\,\hat{z}_\zeta(\sqrt{\phi},\bar{t}) \end{pmatrix}^T. \tag{42}$$

The algorithm to compute $\mathbf{x}$ is described in Algorithm 2 with $M_V = 2N_V$. Despite the fact that the system matrix M has a reasonable condition number at around 10^5 the system $M\mathbf{x}=\mathbf{b}$ is sensitive, especially to the value of ϕ. The authors note that not all conventional approaches to solve the linear system will give a satisfying solution. It has been found numerically, however, that treating the matrix as sparse helps with stability.

5.2.2. Evolving the conformal mapping parameters in time

In order to evolve the deforming bubble interfaces, $z(\zeta,t)$, $|\zeta|=1$, the conformal mapping parameters $b(t)$, $\phi(t)$ and $\{a_n(t)\}_{n=-N_V}^{N_V}$ are integrated in time.

Discretising in time, at time t_k define $b^k\approx b(t_k)$, $\phi^k\approx\phi(t_k)$, $\{a_n^k\}\approx\{a_n(t_k)\}$ etc. The differential equations to evolve the bubble interfaces are

Algorithm 2: Compute flow field functions: F, G and $C(\bar{t})$.

Input : Conformal mapping parameters: b, ϕ and $\{a_n(\bar{t}), n \in [-N_V, N_V]\}$. Surface tension $\sigma(\zeta_j, \bar{t})$, $j = 1, \dots 2M_V + 1$.
Output: Coefficients $\{F_n, n \in [0, N_V]\}$ and $\{G_n, n \in [1, N_V]\}$, $C(\bar{t})$.

1 Zero-pad $\{a_n\}_{n=-N_V}^{N_V}$ to $\{a_m\}_{m=-M_V}^{M_V}$.
2 Compute $z(\zeta_j, \bar{t})$ through (32). Similarly, compute $z(\zeta_j^{-1}, \bar{t})$, $z_\zeta(\zeta_j, \bar{t})$ and $z_\zeta(\zeta_j^{-1}, \bar{t})$.
3 Compute $A(\zeta_j)$, $B(\zeta_j)$ and $E(\zeta_j)$, $\forall j \in [1, 2M_V + 1]$, in (36).
4 Compute Laurent series coefficients for $A(\zeta)$, $B(\zeta)$ and $E(\zeta)$ using FFTs.
5 Truncate Laurent series $\{A_m\}_{-M_V}^{M_V}$, $\{E_m\}_{-M_V}^{M_V}$ to include only $2N_V + 1$ terms.
6 Create right hand side $\mathbf{b}$ to solve $M\mathbf{x} = \mathbf{b}$, constructed according to (42).
7 Create system matrix M:
The first $2N_V + 1$ rows correspond to (37). To find the terms in P_n, the following expression is used

$$P_n = \sum_{k=-N_V}^{N_V} B_{n+k-1} k F_k, \ \forall n \in [-N_V, N_V]. \tag{43}$$

For all terms, the relations (38) and (39) are used.
The last row of the matrix corresponds to (41).
8 Solve $M\mathbf{x} = \mathbf{b}$.
9 Define function coefficients: $\{F_n\}_0^{N_V} = (x_1, \dots, x_{N_V+1})$, $\{G_n\}_1^{N_V} = (x_{N_V+2}, \dots, x_{2N_V+1})$ and $C(\bar{t}) = x_{2N_V+2}$.

$$\begin{aligned} \frac{d\hat{z}}{dt}(\zeta, t) =& \zeta \hat{z}_\zeta(\zeta, t) - \frac{b(t)\zeta I(\zeta, t) - b(t)\sqrt{\phi(t)} I(\sqrt{\phi(t)}, t)}{(\zeta - \sqrt{\phi(t)})^2} - 2F(\zeta, t) \\ &\dots + \frac{b(t) I(\sqrt{\phi(t)}, t) + b(t)\sqrt{\phi(t)} I_\zeta(\sqrt{\phi(t)}, t)}{\zeta - \sqrt{\phi(t)}} := f(\zeta, t), \end{aligned} \tag{44}$$

$$\frac{d\phi}{dt} = -2\phi(t) I(\sqrt{\phi(t)}, t) =: g(t), \tag{45}$$

where $I(\zeta, t_k)$ is known and $\hat{z}$ is defined in (33). Regarding (44) and taking the Laurent series expansion of both sides, this can be written as a system of ODEs, as

$$\frac{d}{dt} a_n = f_n(t), \ n \in [-N_V, N_V], \tag{46}$$

where $\{f_n(t)\}$ are the Laurent series coefficients of $f(\zeta, t)$ above. The function $I(\zeta, t)$ is obtained through the kinematic condition (3) by first computing

$$D(\zeta) = \frac{1}{2|z_\zeta|} + \Re\left[\frac{iC}{\zeta z_\zeta}\right], \tag{47}$$

and taking a Laurent series expansion $D(\zeta) = \sum_{n=-\infty}^{\infty} D_n \zeta^n$. The coefficients of I are obtained through

$$\begin{cases} I_n = 2\left(\dfrac{1 + \phi^n}{1 - \phi^{2n}}\right) D_n, \ n \geq 1 \\ I_0 = D_0 \\ I_{-n} = -\left(\dfrac{2\phi^n}{1 - \phi^n}\right) D_n, \ n \geq 1. \end{cases} \tag{48}$$

The last conformal mapping parameter, $b(t)$, can be decided through the constant area constraint, i.e.

$$\mathcal{A}(t) = -\frac{1}{2i} \oint_{|\zeta|=1} \overline{z}(\zeta^{-1}, t) z_\zeta(\zeta, t) d\zeta = \pi. \tag{49}$$

Each bubble will have an area of π due to the non-dimensionalisation. Expanding (49) the following expression for a is obtained:

$$\begin{aligned} b^2 \oint_{|\zeta|=1} \frac{d\zeta}{(\zeta^{-1} - \sqrt{\phi})(\zeta - \sqrt{\phi})^2} + b \oint_{|\zeta|=1} \frac{\hat{z}(\zeta^{-1}, t)}{(\zeta - \sqrt{\phi})^2} - \frac{\hat{z}_\zeta(\zeta, t)}{(\zeta^{-1} - \sqrt{\phi})} d\zeta \\ - \oint_{|\zeta|=1} \hat{z}(\zeta^{-1}, t) \hat{z}_\zeta(\zeta, t) d\zeta = 2i\pi. \end{aligned} \tag{50}$$

How to compute the $f(\zeta, t)$ and $g(t)$ in (44) and (45) is described in Algorithm 3.

Algorithm 3: Computation of RHS in (44), (45).

Input : Conformal mapping parameters at time t^k: b^k, ϕ^k, $\{a_n^k, n \in [-N_V, N_V]\}$. Flow functions and surface tension at time t^k: $C(t_k)$, $\{F_n^k\}_{n=0}^{N_V}$, $\{G_n^k\}_{n=1}^{N_V}$ and σ_j^k, $j = 1, \ldots, 2N_V + 1$.

Output: $\{f_n(t_k)\}_{n=-N_V}^{N_V}$, $g(t_k)$.

1 Zero-pad $\{a_n^k\}_{n=-N_V}^{N_V}$ to $\{a_m^k\}_{m=-M_V}^{M_V}$.
2 Compute $z_\zeta(\zeta_j, t_k)$, $\hat{z}_\zeta(\zeta_j, t_k)$ and $\hat{z}(\zeta^{-1}, t_k)$ through (32), $\forall j \in [1, 2M_V + 1]$.
3 Compute $D(\zeta_j)$ in (47), $\forall j \in [1, 2M_V + 1]$.
4 Compute $\{d_m\}_{m=-M_V}^{M_V}$ through FFT of $D(\zeta_j)$.
5 Compute $\{I_m\}_{m=-M_V}^{M_V}$ from $\{d_m\}_{-M_V}^{M_V}$ as in (48).
6 Using $\{I_m\}$, compute $I(\zeta_j, t_k)$, $I(\sqrt{\phi}, t_k)$ and $I_\zeta(\sqrt{\phi}, t)$.
7 Compute f according to (44), compute Laurent series coefficients $\{f_m\}_{m=-M_V}^{M_V}$ by FFT and truncate to $2N_V + 1$ terms, $\{f_n\}_{n=-N_V}^{N_V}$.
8 Compute g through (45).

5.2.3. Computing the surfactant concentration

On the bubble boundaries, $\zeta = e^{i\nu}$, the convection-diffusion equation for the surfactant concentration (13) can be written in terms of $\nu \in [0, 2\pi)$ as

$$\left.\frac{\partial \rho}{\partial t}\right|_\nu - \Re\left(\frac{\rho_\nu}{z_\nu} z_t\right) + \frac{1}{|z_\nu|}\frac{\partial}{\partial \nu}\Re\left(P(\nu,t)\right) - \frac{1}{|z_\nu|}\Im\left(\frac{z_{\nu\nu}}{z_\nu}\right)\Im\left(P(\nu,t)\right) - \frac{1}{|z_\nu|}\frac{1}{Pe_\Gamma}\frac{\partial}{\partial \nu}\left(\frac{\rho_\nu}{|z_\nu|}\right) = 0, \tag{51}$$

where

$$P(\nu,t) = \frac{u(\zeta,t)\overline{z}_\nu \rho(t)}{|z_\nu|}. \tag{52}$$

The flow velocity $u(\zeta,t) = u(z(\zeta,t),t)$ is obtained from the normal stress balance (10),

$$u(\zeta,t) = \frac{\sigma(\zeta,t)}{2}\frac{\zeta z_\zeta}{|z_\zeta|} + iC(t) - \frac{4i\sqrt{\phi}F_0}{\zeta - \sqrt{\phi}} - 2iF(\zeta,t), \tag{53}$$

where $F(\zeta,t)$ and F_0 as defined previously. Furthermore, z_t is obtained from the kinematic condition (3)

$$z_t = \zeta z_\zeta I(\zeta,t) - -\frac{4i\sqrt{\phi}F_0}{\zeta - \sqrt{\phi}} - 2iF(\zeta,t), \tag{54}$$

for $I(\zeta,t)$ as defined in (48).

As in §4.4, the convection and diffusion parts of (51) need to be treated explicitly and implicitly respectively. The equation becomes

$$\left.\frac{\partial \rho}{\partial t}\right|_\nu = f_{exp}(\nu,\rho,t) + f_{imp}(\nu,\rho,t), \tag{55}$$

where

$$f_{exp}(\nu,\rho,t) = \Re\left(\frac{\rho_\nu}{z_\nu} z_t\right) - \frac{1}{|z_\nu|}\frac{\partial}{\partial \nu}\Re\left(P(\nu,t)\right) + \frac{1}{|z_\nu|}\Im\left(\frac{z_{\nu\nu}}{z_\nu}\right)\Im\left(P(\nu,t)\right), \tag{56}$$

and

$$f_{imp}(\nu,\rho,t) = \frac{1}{|z_\nu|}\frac{1}{Pe_\Gamma}\frac{\partial}{\partial \nu}\left(\frac{\rho_\nu}{|z_\nu|}\right). \tag{57}$$

How to compute f_{exp} is shown in Algorithm 4. The function f_{imp} is computed in a similar way. In the time-integration scheme, the solution is found through solving a system using gmres.

5.2.4. Time-integration of the system

To evolve the system with the conformal mapping parameters and the surfactant concentration a suitable time-integration method is needed. In this paper, the second order adaptive method in §4.5 is used. For simplicity, in Algorithm 5 a first order method with fixed time step is employed to demonstrate the method. In the case of clean drops, $\sigma(\zeta,t)$ is set to be constant and the algorithm skips steps 9 to 12. At each time step, a Krasny filter is applied where all Fourier modes smaller than 10^{-12} are set to zero.

Algorithm 4: Computation of f_{exp} in (56).

Input : At time t_k: conformal mapping parameters; b^k, ϕ^k, $\{a_n^k, n \in [-N_V, N_V]\}$, flow field; $\{F_n^k\}_0^{N_V}$, $\{G_n^k\}_1^{N_V}$ and $C(t_k)$, and surfactant concentration, ρ_j^k, $j \in [0, 2N_V + 1]$.
Output: $f_{exp,j}(t_k)$

Zero-pad $\{a_n^k\}_{n=-N_V}^{N_V}$ to $\{a_m^k\}_{m=-M_V}^{M_V}$.
Using $\{a_m^k\}$, compute $z^k(\zeta_j)$, $z_v^k(\zeta_j)$ and $z_{vv}^k(\zeta_j)$ through (32).
Zero-pad Fourier coefficients of ρ_j^k to $\{\hat{\rho}_m^k\}_{m=-M_V}^{M_V}$ and compute $\rho_v^k(\zeta_j)$ from coefficients.
Compute $F(\zeta_j, t_k)$ from $\{F_n^k\}_0^{N_V}$.
Compute $u(\zeta_j, t_k)$ from (53).
Compute $z_t(\zeta_j, t_k)$ from (54).
Compute $P(v_j, t_k)$ through (52). Derivative $\frac{\partial}{\partial v}\Re(P(v,t))$ computed via `FFT`.
Compute $f_{exp}(v_j, \rho_j^k, t_k)$ from (56).

Algorithm 5: First order method for time-integration of surfactant-covered bubbles using fixed time step.

Input : Conformal mapping parameters: b^0, ϕ^0, $\{a_n^0, n \in [-N_V, N_V]\}$. Initial surfactant concentration ρ_j^0 and surface tension σ_j^0, $j = 1, \ldots, 2N_V + 1$. Time interval $[t_0, t_s]$, time step dt.
Output: Conformal mapping parameters at time t_s: $b(t_s)$, $\phi(t_s)$, $\{a_n(t_s), n \in [-N_V, N_V]\}$. Surfactant concentration at time t_s: $\rho(\zeta_j, t_s)$, $j = 1, \ldots, 2N_V + 1$.

$N_t = \dfrac{t_s - t_0}{dt}$
for $k \in [0, N_t - 1]$ **do**
  Compute flow field functions at time t_k:
  $\left[\{F_n^k\}_{n=0}^{N_V}, \{G_n^k\}_{n=0}^{N_V}, C^k\right] =$ `algorithm2`$\left(b^k, \{a_n^k\}_{-N_V}^{N_V}, \phi^k, \sigma_j^k\right)$.
  Compute RHS in (44) and (45):
  $[\{f_n\}_{n=-N_V}^{N_V}, g] =$ `algorithm3`$\left(b^k, \phi^k, \{a_n^k\}_n, \sigma_j^k, \{F_n^k\}_n, \{G_n^k\}_n, C^k\right)$
  Update $\{a_n\}$ in discretised version of (46):
  $a_n^{k+1} = a_n^k + dt \cdot f_n$, $n \in [1, N]$.
  Update ϕ in discretised version of (45):
  $\phi^{k+1} = \phi^k + dt \cdot g$.
  Compute b^{k+1} through (50).
  Compute a_n^{k+1} for $n \leq 0$ through (34).
  Steps $9 - 12$ *are skipped in case of clean bubbles:*
  Compute $f_{exp}(t_k)$ in (56), $\forall j \in [1, 2M_V + 1]$:
  $f_{exp,j}^k =$ `algorithm4`$\left(b^k, \phi^k, \{a_n^k\}, \{F_n^k\}, \{G_n^k\}, C(t_k), \rho_j^k\right)$.
  Compute ρ_j^{k+1} from $\rho_j^{k+1} = \rho_j^k + dt f_{exp,j}^k + dt f_{imp,j}^{k+1}$ with `gmres`, f_{imp} as in (57).
  Truncate zero-padding from ρ_j^{k+1}.
  Update surface tension coefficient σ_j^{k+1} through equation of state.
  Update time $t_{k+1} = t_k + dt$.
end

6. Results

In this section the numerical method suggested in this paper is tested against the validation methods described in §5. Each test case is described in detail below. The numerical method described in §4 will be denoted the BIE method for the remainder of the paper.

6.1. The influence of surfactants

First, the influence of the surfactant concentration on the deformation of a pair of droplets is presented. A pair of initially circular droplets, with viscosity ratio $\lambda = 0.2$, are placed on the imaginary axis with a minimum distance of 0.2. Under the influence of an extensional flow with Capillary number $Q = 0.2$ these drops will be pushed towards each other. Their configuration at time $t = 25$ can be seen in Fig. 5 (left). The presence of surfactants affects the deformation such that the minimum distance between the drops is larger, as can be seen to the right in Fig. 5. Also, the drops will become more elongated and narrower. For both simulations, each drop was initially discretised with 320 points. At the final time $t = 25$ this has been increased to 416 and 448 points by the spatial adaptivity, for the clean and surfactant-covered cases respectively. This corresponds to $\Delta s \approx 0.003$. The linear equation of state was used, and $E = 0.2$, $\mathrm{Pe}_\Gamma = 10$. A higher elasticity number E in (12) means that the surface tension coefficient will be more affected by the non-uniform surfactant concentration and

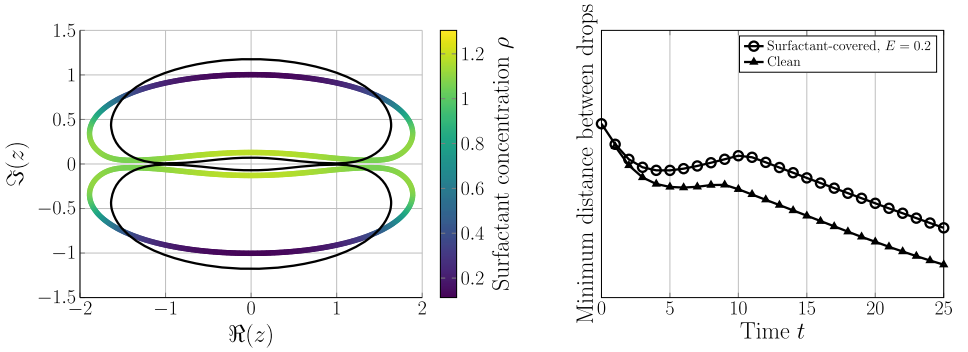


Fig. 5. Left: drop configuration at time $t = 25$ under the influence of an extensional flow with $Q = 0.2$. The coloured contours represent drops with surfactants present and elasticity number $E = 0.2$, the black contours represent those without, i.e. $E = 0$. Right: the minimum distance between the drops versus time, for both the clean and the surfactant-covered drops.

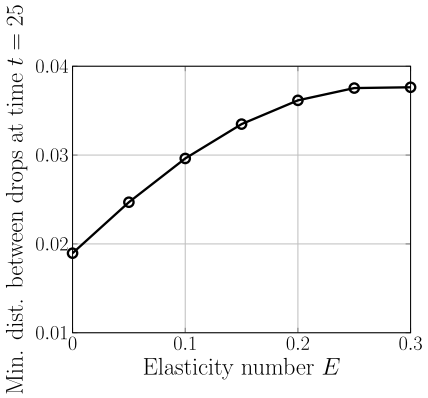


Fig. 6. How minimum distance between drops at time $t = 25$ changes with elasticity number E. The case $E = 0$ corresponds to the clean case.

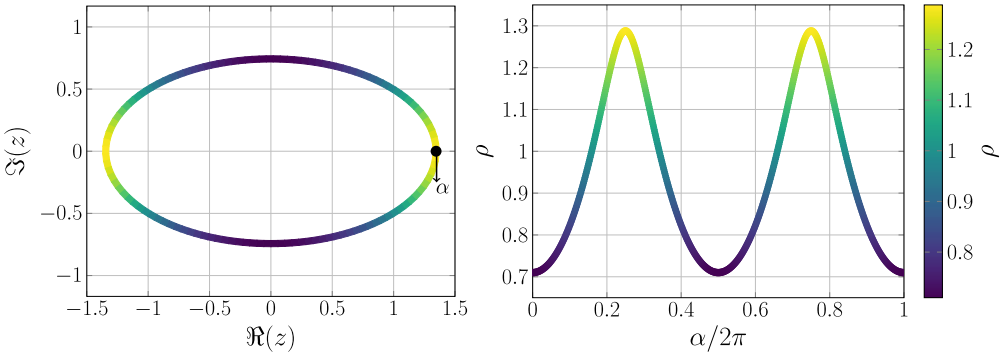


Fig. 7. Interface (left) and surfactant concentration (right) at steady state for Capillary number $Q = 0.14$. The simulation was run with elasticity number $E = 0.5$, initial surfactant concentration $\rho_0 = 1$, $Pe_\Gamma = \infty$ and viscosity ratio $\lambda = 0$. The steady state according to the definition in (58) was reached at time $T = 46.35$.

thus deform more. Also, the distance between the drops will be larger for larger values of E, as can be seen in Fig. 6. Note that the case of $E = 0$ corresponds to the surfactant-free (clean) case.

6.2. Surfactant-covered bubble in steady state

Using the exact solutions of §5.1, the bubble deformation and surfactant concentration at steady state is tested. Given a Capillary number Q, the steady state interface position $z(\zeta, T)$ and the surfactant concentration $\rho(\zeta, T)$ are known, where T is the time of steady state and $\zeta = e^{i\nu}$ for $\nu \in [0, 2\pi)$. An example of the interface position and the surfactant concentration at a steady state is shown in Fig. 7, for the steady state corresponding to $Q = 0.14$. In a simulation, a bubble is said to have reached steady state when the normal velocity, $\mathbf{u} \cdot \mathbf{n} = 0$. Practically, here the definition of steady state is that

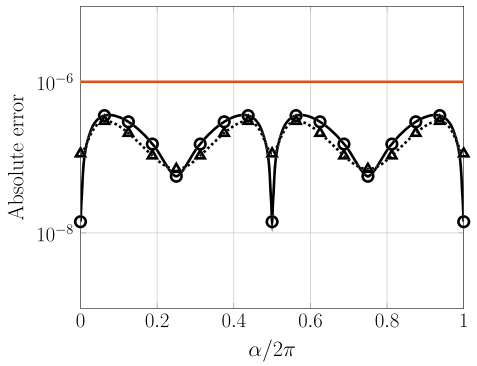


Fig. 8. Measured point-wise error in z (marker ∘) and ρ (marker △) vs. α at steady state, for the same parameters as in Fig. 7, with grid spacing $\Delta s = 0.0016$. Red line corresponds to the time-stepping tolerance.

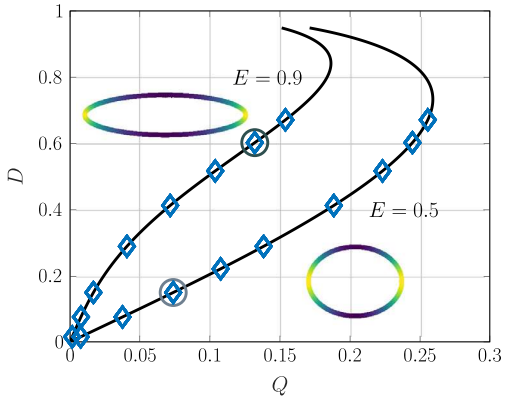


Fig. 9. Steady state deformation, D, vs. Capillary number Q at different elasticity numbers E for inviscid bubbles, $\lambda = 0$. Black lines show validation solutions computed through the method in §5.1, diamonds show numerical results from the BIE method and the circles indicate the selected simulations for which the bubble shape and surfactant concentration are shown.

$$|\mathbf{u} \cdot \mathbf{n}| \leq 10^{-8}. \tag{58}$$

It should be noted that for large Capillary numbers, Q, or high viscosity ratios, λ, this threshold is slightly too large and should be decreased since the deformation is very slow.

The analytical solutions from Algorithm 1 are denoted by $z^V(\zeta, T)$ and $\rho^V(\zeta, T)$ for the interface position and the surfactant concentration respectively. Defining the obtained solutions from the numerical method as $z(\alpha, T)$ and $\rho(\alpha, T)$, a comparison between the numerical results and the validation is obtained by studying the point-wise differences in $\alpha \in [0, 2\pi)$. Two errors will be studied: in position, $e_z = |z - z^V|$ and surfactant concentration ρ, $e_\rho = |\rho - \rho^V|$. In order to compare point-wise in $\alpha \in [0, 2\pi)$, a mapping from ν to the equal-arc length measure α^V is needed. This is defined as

$$\alpha^V(\nu) = \frac{S_\nu 2\pi}{L}, \tag{59}$$

where $S_\nu = \int_0^\nu |z_\nu^V| d\nu$. The solutions z^V and ρ^V are computed at non-equidistant discrete points $\alpha_i^V = \alpha^V(\nu_i)$, for $\nu_i \in [0, 2\pi)$, $i = 1, \ldots, N_V$ for some N_V. Finally, to compare the results, $z(\alpha, T)$ and $\rho(\alpha, T)$ need to be interpolated to α_i^V. This is done using their Fourier expansions.

In Fig. 8, the point-wise absolute error between the analytical solution and the BIE simulation can be seen for the case shown in Fig. 7. The spatial discretisation is adaptive, keeping the spatial distance in arc length similar at all times; here approximately $\Delta s = 0.008$. The simulation was run with time-step tolerance 10^{-6} and the errors are all below the set tolerance.

Using the method in §5.1, one can obtain a graph for how the deformation D depends on the Capillary number Q. Such a graph for two different elasticity numbers, $E = 0.5$ and $E = 0.9$, is shown in Fig. 9. The black lines represent the validation solutions computed through the method in §5.1 and the blue diamonds show the results from simulations using the BIE method of this paper. All simulations have been run from the same initial setting as the previous case, with time-step tolerance 10^{-6}. The simulations differ only in the choice of Capillary number Q. Examples of bubble deformation and surfactant concentration are also shown, corresponding to $Q = 0.07$ and $Q = 0.13$, marked with circles in the figure.

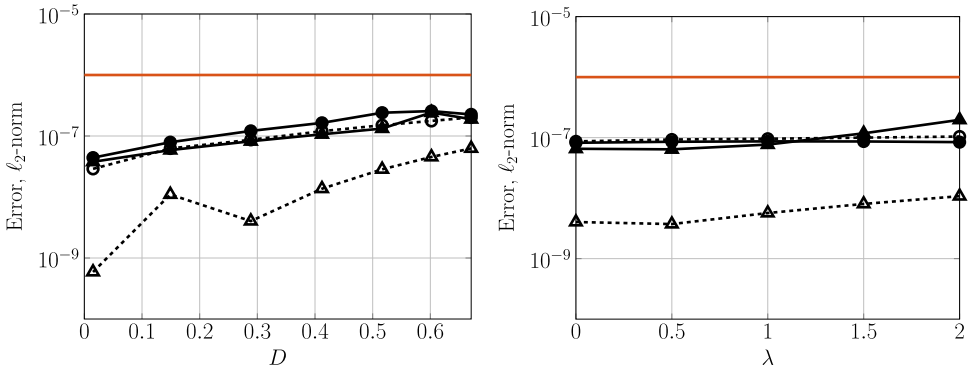


Fig. 10. ℓ_2 error in z and ρ vs. (left) deformation D for cases in Fig. 9 and (right) vs. viscosity ratio λ for deformation $D = 0.29$. Solid and dashed lines represent $E = 0.5$ and $E = 0.9$ respectively. Dots and triangles represent e_z and e_ρ respectively. Red solid line shows time-step tolerance 10^{-6}.

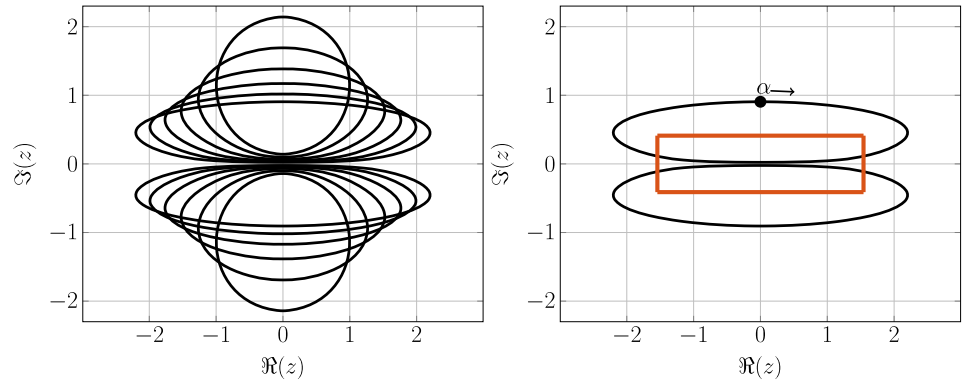


Fig. 11. Clean bubbles deforming in an extensional flow. Left: interface position from time $t = 0$ to $t = 1.5$, at $dt = 0.3$ intervals, computed with BIE method. Right: final interface configuration at time $t = 1.5$. Red box: cut-off region for method comparisons.

To quantify the results in Fig. 9, the ℓ_2-error for each deformation D is plotted in Fig. 10 (left). For each diamond in Fig. 9, the simulation and the validation data is compared and plotted vs. deformation D. Note that both cases $E = 0.5$ and $E = 0.9$ are shown (solid and dashed lines respectively), and that they for the same D were obtained by different Q. It is shown that both errors (e_z and e_ρ) stay below the time-step tolerance.

As mentioned in §5.1, the viscosity ratio does not affect the steady state deformation in the case of $Pe_\Gamma = \infty$. The results of this paper agree with this observation, as is shown in Fig. 10 (right). There, droplets of different viscosity ratios ranging from $\lambda = 0$ (corresponding to an inviscid bubble) to $\lambda = 2$ were deformed under Capillary number $Q = 0.14$. As can be seen in the figure, both the interface position and the surfactant concentration coincide with that for bubbles up to time-step tolerance.

6.3. A pair of clean bubbles in extensional flow

Using the approach described in §5.2, the semi-analytical solutions for a pair of bubbles deforming in an extensional flow are computed and compared against the boundary integral method of this paper.

The following case has been selected as it pushes the bubbles close to each other, thus providing an excellent test case for the special quadrature. The bubbles are initially circular with radius one, centred around $\pm 1.419i$ which corresponds to an initial $\phi(0) = 0.35$. The bubbles are clean, i.e. there are no surfactants present in this problem, and they deform under an extensional flow with Capillary number $Q = 0.5$ until time $t = 1.5$. At the final time, the minimum distance between the bubbles is 0.04. In Fig. 11 the movement of the bubbles over time and the final deformation are shown (left and right respectively).

The interface position obtained from the BIE method is denoted $z(\alpha, t)$, where $\alpha \in [0, 2\pi)$. In Fig. 12 (right) the discretisation along the interface is shown. Note that it is uniform around the interface. In contrast, the discretisation along the interface for the validation method is not uniform, see Fig. 12 (left). There, the points are clustered where the distance between the bubbles is the smallest. When comparing the results of the two methods, only the part of the interfaces between $\alpha = \frac{2\pi}{3}$ and $\alpha = \frac{4\pi}{3}$ will be considered. This corresponds to the red box in Fig. 11 (right). This cut-off is made because the discretisation of the validation method is very coarse outside of it, with large spatial errors. Also, since it is the

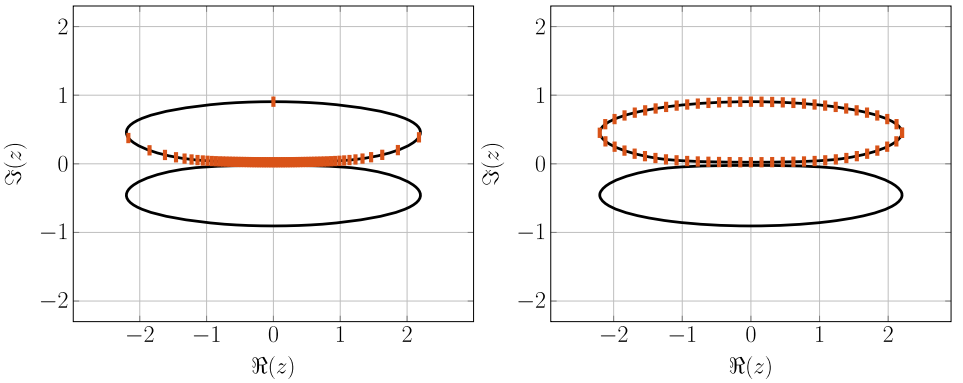


Fig. 12. Interface at time $t = 1.5$ for clean bubbles. The red markers to the left show every 16th point of the validation discretisation for $N_V = 1025$. To the right, the red markers show every 16th point of the uniform discretisation for the BIE method, with $N_k = 960$, $k = 1, 2$.

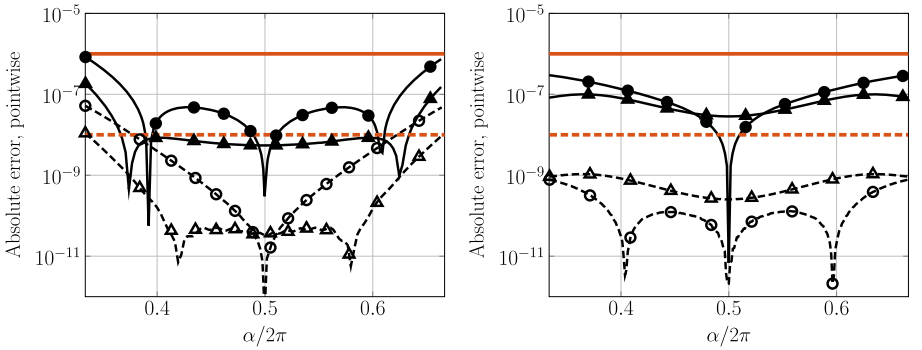


Fig. 13. Pointwise relative difference in x- and y-coordinates for validation method (left) and BIE method (right) against reference solution. Solid lines/markers shows time-stepping tolerance 10^{-6}, dashed for tolerance 10^{-8}. BIE method computed with 576 and 800 points per bubble for tol_1 and tol_2 respectively. Validation method computed with 2049 and 9001 points respectively. Error in x denoted by ∘ and in y by △.

area of near-interaction between the bubbles which is of greatest interest, as it is there the boundary integral method will potentially struggle the most, it is sufficient to consider only this part.

To investigate how the two methods behave, they are compared against a reference solution, $z_{ref} = x_{ref} + iy_{ref}$. This solution is computed with the boundary integral method using 600 Gauss-Legendre panels, i.e. $N_k = 9600$, $k = 1, 2$, and a time-step tolerance of $tol = 10^{-10}$. To compare the non-uniform discretisation of the validation method with the uniform discretisation of the BIE method, the map as in (59) is employed and the points are interpolated using a non-uniform FFT [9]. Errors in both x- and y-coordinates will be considered and denoted $e_x = |x - x_{ref}|$ and $e_y = |y - y_{ref}|$ respectively, the combined error in z is denoted $e_z = |e_x + ie_y|$.

In Fig. 13 the point-wise absolute errors are shown in the region of interest for the validation method and the BIE method. The results have been computed using two time-step tolerances: $tol_1 = 10^{-6}$ (solid lines) and $tol_2 = 10^{-8}$ (dashed lines). For the validation method, $N_V = 2049$ and $N_V = 9001$ points were used for tol_1 and tol_2 respectively, and for the BIE method $N_k = 576$, $k = 1, 2$, was used for tol_1 and $N_k = 800$ for tol_2. It is shown that both methods have errors below the set tolerance tol_1, but that the validation method does not quite achieve the tolerance 10^{-8}. This is due to the increase in $\Delta\alpha$ towards the end of the cut-off, which causes some of the points to have a larger error than the set tolerance.

To quantify the behaviour of the two methods, the absolute errors in max-norm for different discretisations of the bubbles are plotted. Note that the number of discretisation points is constant for both methods, i.e. it does not increase over time as the bubbles get more stretched out. This especially means that the spatial adaptivity of the BIE method was turned off throughout the simulation. In Fig. 14 the errors vs number of discretisation points N are shown. It is clear that the BIE method (Fig. 14 (right)) agrees with the reference solution to within time-step tolerance for both tol_1 and tol_2 when the spatial resolution is high enough. For tol_1, 36 panels, i.e. $N_k = 576$ is sufficient to reach the set tolerance. For tol_2, 46 panels, i.e. $N_k = 736$, are needed. For the validation method the spatial errors dominate the solution and it is first at around $N_V = 2000$ points that the method converges to the tolerance 10^{-6}. To achieve also tol_2, the spatial discretisation needs to be increased further.

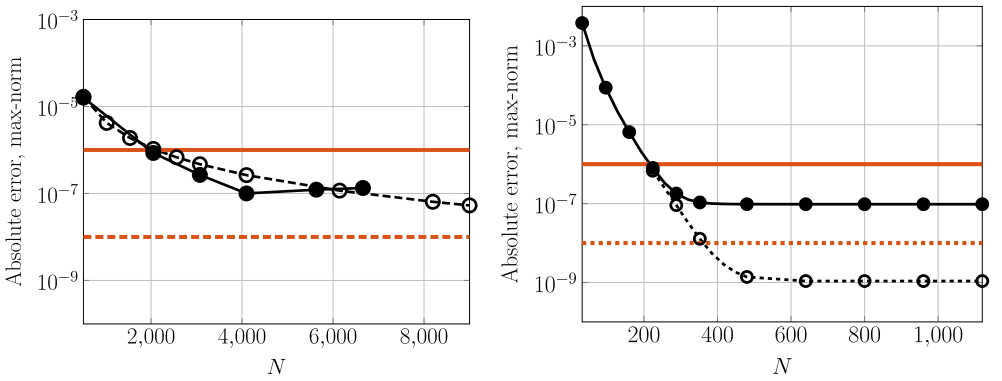


Fig. 14. Error as a function of number of discretisation points for validation method (left) and BIE method (right). Solid lines/markers show results for time-stepping tolerance 10^{-6} and dashed lines tolerance 10^{-8}. The spatial adaptivity of the BIE method was turned off to keep N constant throughout the simulation.

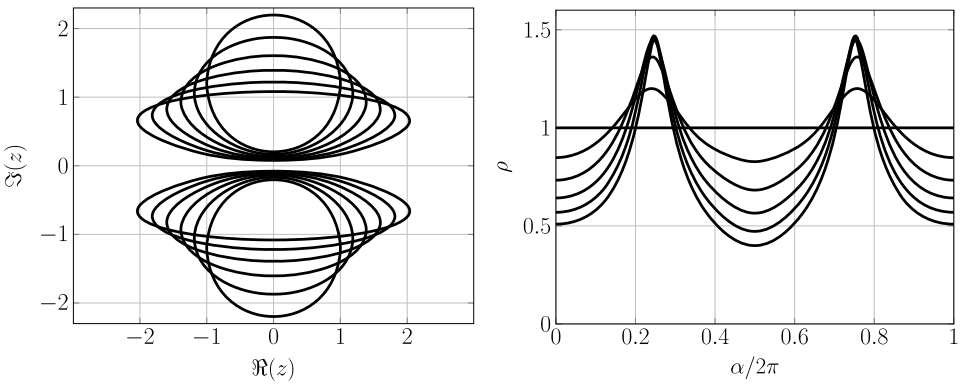


Fig. 15. Pair of surfactant-covered bubbles deforming over time. Left: interface position from time $t = 0$ to $t = 1$, at $dt = 0.2$ intervals, computed with BIE method. Right: surfactant concentration at corresponding times.

6.4. A pair of surfactant-covered bubbles in extensional flow

Similarly to a pair of clean bubbles, the semi-analytical solutions in §5.2 for a pair of surfactant-covered bubbles is computed and compared against the BIE method of this paper.

This case provides a test case for the coupling of the interface and the surfactant concentration, as well as for both the diffusion and the convection of the surfactants along the interface. The bubbles are initially circular, covered with a uniform surfactant concentration $\rho_0 = 1$ and centred around $\pm 1.201i$ which corresponds to $\Phi(0) = 0.2875$. In this problem, the elasticity number $E = 0.5$, Peclet number $Pe_\Gamma = 10$ and the bubbles are placed in an extensional flow with Capillary number $Q = 0.5$. The bubbles will deform until time $t = 1$, and their change over time can be seen in Fig. 15 (left) together with the change of surfactant concentration over time (right). The interface position and surfactant concentration at the final time $t = 1$ can be seen in Fig. 16. At time $t = 1$ the minimum distance between the bubbles is 0.16. Similarly to the clean case, only the region where the bubbles are close to each other will be studied, corresponding to the region between $\alpha = \frac{2\pi}{3}$ and $\alpha = \frac{4\pi}{3}$, shown in black in Fig. 16 (left).

To compare the two methods, a reference solution computed with the BIE method is used. It is computed with a time-step tolerance of 10^{-10} and $\Delta s \approx 6.5 \cdot 10^{-4}$. In Fig. 17 the point-wise absolute error between the two methods and the reference solution is shown, for two time-step tolerances: $tol_1 = 10^{-6}$ (solid lines) and $tol_2 = 10^{-8}$ (dashed lines). The errors considered are those in x- and y-coordinates, e_x and e_y, as well as those in surfactant concentration e_ρ. The BIE method was computed with 576 discretisation points per bubble for tol_1 and 800 for tol_2 and all errors stay within time-step tolerance. Note that the spatial adaptivity of the BIE method was turned off for these simulations. In contrast, for the validation method $N_V = 2049$ points was used for tolerance 10^{-6} and $N_V = 9001$ points for 10^{-8}. Again, for this discretisation the method does not meet the stricter tolerance, due to the spatial accuracy towards the end of the cut-off box. It is clear, however, that given enough spatial resolution the errors would come down also to tol_2. In Fig. 18 the convergence of the two methods is studied. To reach the set tolerances with the BIE method, 36 and 50 panels per bubble are needed for tol_1 and tol_2 respectively. In comparison, the validation method needs approximately 2049 points for tol_1 and more than 8192 to reach tol_2 due to spatial resolution.

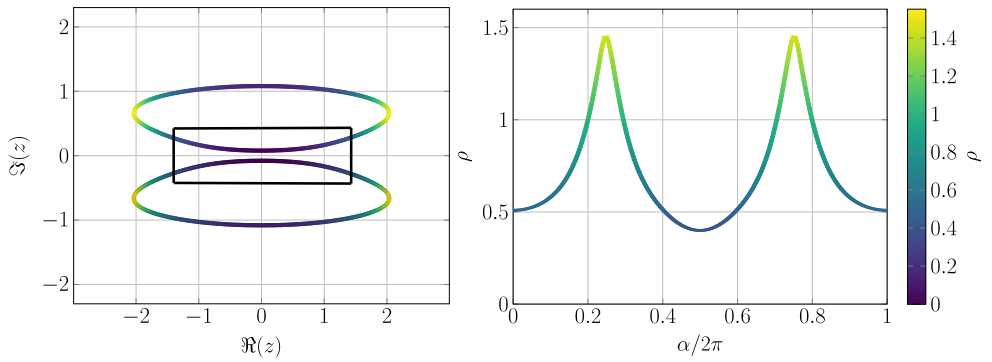


Fig. 16. Surfactant-covered bubbles at time $t = 1$. Left: interface position at final time. Cut-off region shown in black box. Right: surfactant concentration at final time.

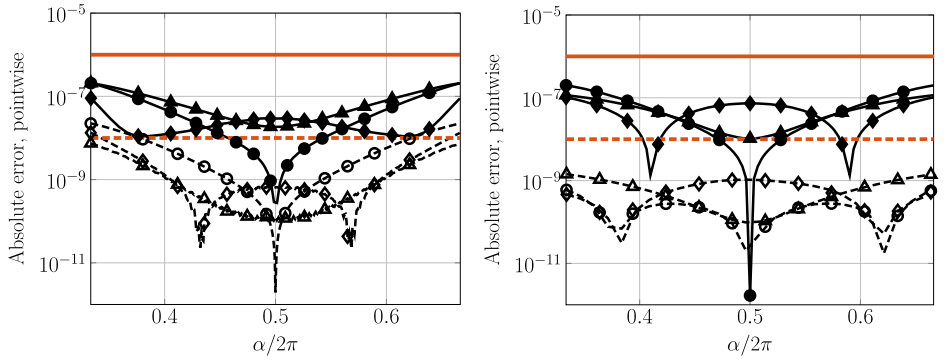


Fig. 17. Point-wise absolute difference in x- and y-coordinates and surfactant concentration for validation method (left) and BIE method (right) against reference solution. Solid lines/markers show time-step tolerance 10^{-6}, dashed for tolerance 10^{-8}. BIE method computed with 576 and 800 points per bubble respectively. Validation method computed with 2049 and 8193 points respectively. Markers: ○, △ and ◇ corresponds to e_x, e_y and e_ρ respectively.

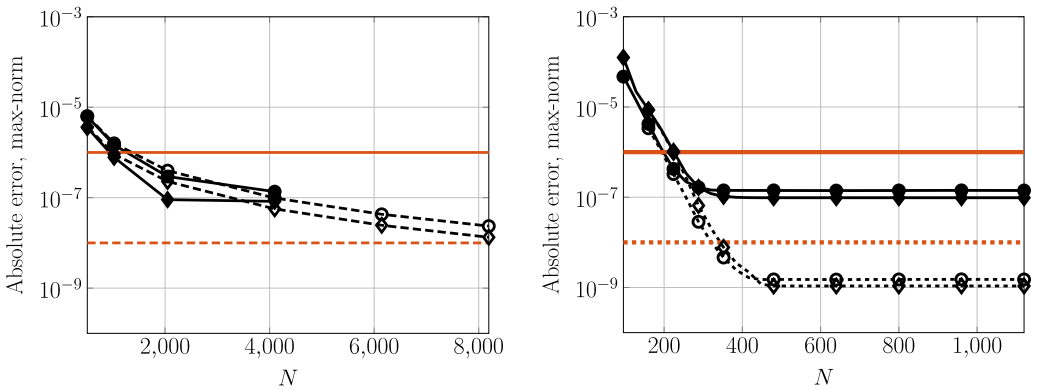


Fig. 18. Error as a function of number of discretisation points for validation method (left) and BIE method (right). Solid lines/markers show results for time-step tolerance 10^{-6} and dashed lines tolerance 10^{-8}. The spatial adaptivity of the BIE method was turned off to keep N constant throughout the simulation. Markers: ○ and ◇ corresponds to e_z and e_ρ respectively.

6.5. The Swiss roll

To further assess the BIE method, a more complicated simulation is set up to test the robustness of the method. The setup is inspired by that in [28]. The initial drop configuration is shown in Fig. 19. No far-field flow is imposed, instead the drops will be allowed to deform until circular under surface tension. A drop is deemed circular when

$$\left| 1 - \frac{\max\left(|z - c|\right)}{\text{mean}(|z - c|)} \right| < 10^{-4},$$

where c is the centre of each drop and z its interface discretisation.

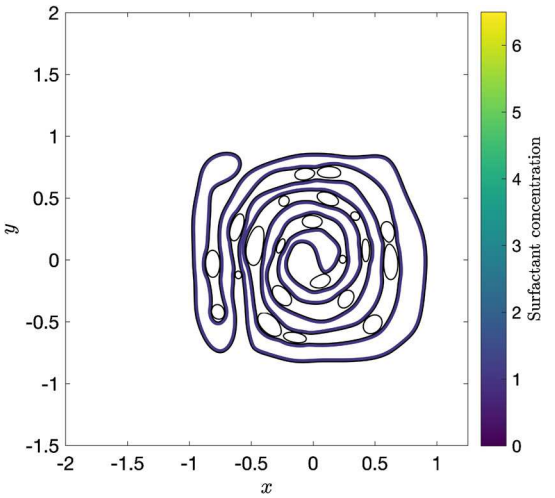


Fig. 19. Initial drop configuration for the swiss roll simulation.

The largest drop (i.e. the roll) will be covered with surfactants with an initial uniform concentration $\rho_0 = 1$. Other parameters are $Pe = 10$ and $E = 0.1$. The authors note that the standard nondimensionalisation of length by initial radius is not appropriate in this case; instead the characteristic length of half of the length of a square box containing the initial drop configuration is used. Initially, the roll is covered with 375 panels and the ellipses 37 panels each. For this simulation $\lambda_k = 1$ for all drops k is considered, in order to make the simulation run in a reasonable time on a standard workstation. The simulation has also been run with 50% more points without any visible difference. Comparing the final circle radius and centre point between the two discretisations, they differ on the level of the circular tolerance imposed above. The simulation takes approximately 10^5 time steps.

How the drops deform in time is shown in Fig. 20. The drops reach their circular form at time $t = 70$. The minimal distance between drops measured during the simulation is $9 \cdot 10^{-5}$. This can be compared to a corresponding simulation without surfactants, where the drops are circular at time $t = 32$, with minimal distance $3 \cdot 10^{-4}$.

7. Conclusions

This paper presents a highly accurate boundary integral method to simulate deforming droplets in Stokes flow. The method can be used for both droplets and inviscid bubbles, both which may be covered by insoluble surfactants. The boundary integral method is coupled with a pseudo-spectral method for the surfactant concentration, and together they are spectrally accurate in space.

The errors introduced by the numerical evaluation of the integrals in the boundary integral equation when the droplets get close to each other are accurately estimated using contour integrals. These errors are removed using a special quadrature, which enables the method to simulate close drop-drop interactions very accurately.

Given a sufficient spatial resolution, the accuracy of the method is limited only by the set tolerance of the adaptive time-stepping scheme. To the knowledge of the authors, this method is more accurate than other methods to simulate Stokes flow in 2D currently available.

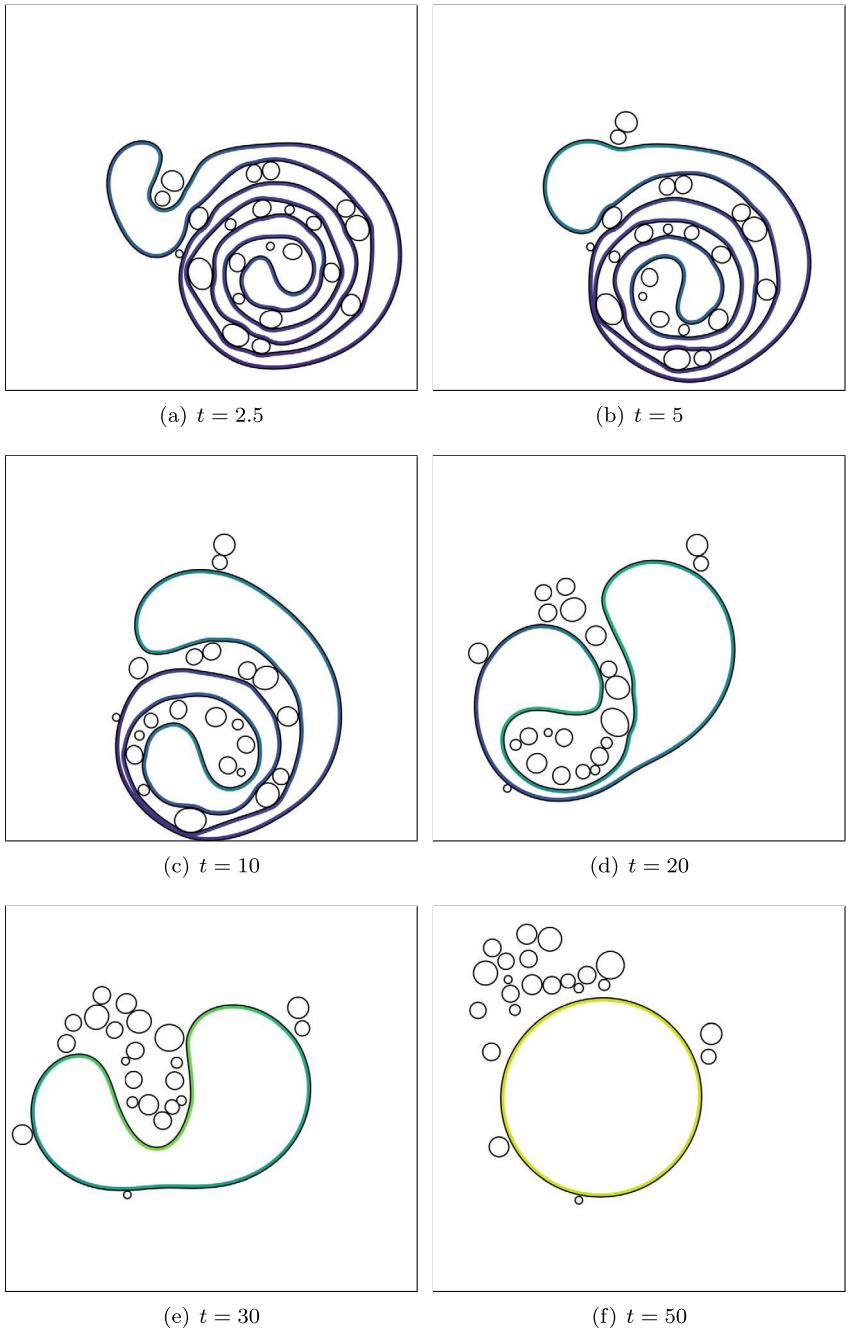
By presenting a set of easily accessible algorithms based on exact and semi-analytical solutions, the hope of the authors is that this will set a standard of validation for any proposed new method for surfactant-laden drops in 2D.

Acknowledgements

The authors gratefully acknowledge the support by the Göran Gustafsson Foundation for Research in Natural Sciences and Medicine (S.P. and A.K.T) as well as the support from the National Science Foundation grant DMS-1412789 (M.S.) and by the Knut and Alice Wallenberg Foundation under grant no. 2014-0338 (M.S. and A.K.T).

Appendix A. Error estimates

As discussed in §3.3, the integrals in equations (15) and (16) both contain terms $(z_j - z_i)^{-1}$, which is problematic when $\|z_i - z_j\| \ll 1$, $i \neq j$. The integrals are then said to become near-singular, which is a numerical problem where large errors are introduced through under-resolving the integrand. To see how the errors grow as an evaluation point $z_0 \in \Omega$ is approaching an interface Γ, the quadrature errors of such near-singular integrals can be approximated [18,19].



(a) $t = 2.5$ (b) $t = 5$

(c) $t = 10$ (d) $t = 20$

(e) $t = 30$ (f) $t = 50$

Fig. 20. Deformation over time for the swiss roll domain. Surfactant concentration in colour of the same scale as in Fig. 19.

In the following, the error of the near-singular integrals when solving Stokes equations in a domain with non-deforming boundaries, Ω, is considered. Similarly to the BIE formulation in §2.3, a complex-valued density is introduced and solved for on the boundary Γ through

$$\mu(z) + \frac{1}{\pi}\int_{\Gamma} \mu(\tau)\Im\left\{\frac{d\tau}{\tau - z}\right\} - \frac{1}{\pi}\int_{\gamma} \overline{\mu(\tau)}\frac{\Im\left\{d\tau\left(\overline{\tau} - \overline{z}\right)\right\}}{\left(\overline{\tau} - \overline{z}\right)^2} = f(z),\ z \in \Gamma, \tag{A.1}$$

where $f(z)$ is the prescribed boundary condition of Γ. Setting the right hand side $f(z)$, $z \in \Gamma$, as a sum of Stokeslets for point sources surrounding Ω, the analytical solution $u(z) = f(z)$ for all $z \in \Omega$. For a given set of point sources: $f_1 = 4\pi + 4i\pi$ located at $x_1 = 1.1 + 1.3i$, $f_2 = \pi - 2i\pi$ at $x_2 = -1.4 - 1.3i$ and $f_3 = -0.5\pi + 3.5i\pi$ at $x_3 = 1.3 - 0.75i$, the solution is

given in Fig. 2(a) (right). The domain and the location of the point sources can be seen in Fig. 2(a) (left). The solution $u(z_0)$ for $z_0 \in \Omega$ is computed through

$$u(z_0) = -\frac{i}{\pi}\int_\Gamma \mu(\tau)\Im\left(\frac{d\tau}{\tau - z_0}\right) + \frac{i}{\pi}\int_\Gamma \overline{\mu(\tau)}\frac{\Im\left\{d\tau(\overline{\tau} - \overline{z_0})\right\}}{(\overline{\tau} - \overline{z_0})^2}. \tag{A.2}$$

Comparing (A.2) to (16), it is clear that the only difference in the integrals between the two is the imaginary part in the first integral in (A.2). This should not affect the near-singular behaviour of the integral. In the following (A.1) is solved to high accuracy, and the errors of the solution will come from solely the evaluation of (A.2).

Discretising the integrals in (A.2) with a composite 16-point Gauss-Legendre quadrature and computing $u(z)$ for $z \in \Omega$, the error compared to the analytical solution is shown in Fig. 2(b). Far away from the boundary Γ, the error is very small but as z_0 approaches Γ the error increases. In the region where the errors are large, special treatment is needed for an accurate solution; the method for this is explained in §3.4. In order to decide where such special treatment is needed, it is important to know how the errors behave. For Laplace's and Helmholtz equations, estimates of the quadrature error has been derived by af Klinteberg and Tornberg [18,19], based on contour integration and calculus of residues. This approach will be followed here.

Regarding the integral expression in (A.2), the error at a point $z_0 \in \Omega$ can be seen as the sum of the errors when integrating over each panel Γ_i:

$$e(z_0) = \sum_{i=1}^{N_{panels}} e_i(z_0).$$

The errors will be estimated on a point-panel basis, where the error $e_i(z_0)$ is estimated by $R_i(z_0)$. For convenience only one panel Γ_i will be regarded in the following derivation.

In [18,19] estimates are shown for integrals of the type

$$I = \int_{-1}^{1} f(\xi)d\xi, \tag{A.3}$$

where the meromorphic function $f(\xi)$ is analytic on $[-1, 1]$ and has a pole $\xi_0 \in \mathbb{C}$ of order $m+1$. To consider the integrals in (A.2), the panel Γ_i is mapped to the real line $[-1, 1]$ by a numerically constructed mapping $\eta(\xi)$. Also, the point $\xi_0 \in \mathbb{C}$ that corresponds to each evaluation point $z_0 \in \Omega$ needs to be found. This procedure is explained in [19].

To compute the estimates, first the integrals are rewritten on the form of (A.3). For the first integral in (A.2), this is straightforward:

$$\begin{aligned} J_1(z_0) := \int_{\Gamma_i} \mu(\tau)\Im\left(\frac{d\tau}{\tau - z_0}\right) &= \Im\left(\int_{-1}^{1} \mu(\xi)\frac{\eta'(\xi)}{\eta(\xi) - \eta(\xi_0)}d\xi\right) \\ &=: \Im\left(\int_{-1}^{1} f_1(\xi)d\xi\right). \end{aligned} \tag{A.4}$$

The second integral is reformulated as follows

$$\int_\Gamma \overline{\mu(\tau)}\frac{\Im\left\{(\overline{\tau} - \overline{z_0})d\tau\right\}}{(\overline{\tau} - \overline{z_0})^2} = \frac{i}{2}\overline{\int_\Gamma \frac{\mu(\tau)\overline{n}_\tau^2 d\tau}{\tau - z_0}} + \frac{i}{2}\int_\Gamma \frac{\mu(\tau)(\overline{\tau} - \overline{z_0})}{(\tau - z_0)^2}, \tag{A.5}$$

where n_τ is the (outward) normal at point τ. Then,

$$\begin{aligned} J_2(z_0) := \int_\Gamma \frac{\mu(\tau)\overline{n}_\tau^2 d\tau}{\tau - z_0} &= \int_{-1}^{1} \frac{\mu(\xi)}{\eta(\xi) - \eta(\xi_0)}\left(\overline{\frac{i\eta'(\xi)}{|\eta'(\xi)|}}\right)^2 \eta'(\xi)d\xi \\ &=: \int_{-1}^{1} f_2(\xi)d\xi, \end{aligned} \tag{A.6}$$

and

$$J_3(z_0) := \int_\Gamma \frac{\mu(\tau)(\overline{\tau} - \overline{z_0})}{(\tau - z_0)^2} = \int_{-1}^{1} \frac{\mu(\xi)(\overline{\eta(\xi)} - \overline{\eta(\xi_0)})}{(\eta(\xi) - \eta(\xi_0))^2} \eta'(\xi) d\xi =: \int_{-1}^{1} f_3(\xi) d\xi. \tag{A.7}$$

The whole integral over Γ_i can be written as

$$I_i(z_0) = -\frac{i}{\pi} J_1(z_0) + \frac{1}{2\pi} \overline{J_2(z_0)} + \frac{1}{2\pi} \overline{J_3(z_0)}. \tag{A.8}$$

For each integral, error can be estimated as $R^1[\xi_0]$, $R^2[\xi_0]$ and $R^3[\xi_0]$ respectively. In total, the error of the sum of the integrals over the panel Γ_i can then be estimated as the absolute value of

$$R_i(z_0) = \left| -\frac{i}{\pi} R[f_1] + \frac{1}{2\pi} \overline{R[f_2]} + \frac{1}{2\pi} \overline{R[f_3]} \right|. \tag{A.9}$$

In the following, how to obtain each estimate $R[f_1]$, $R[f_2]$ and $R[f_3]$ will be explained.

af Klinteberg and Tornberg showed that for the integral (A.3) the quadrature error $R_n[f]$ can be approximated by

$$R_n[f] \approx -\frac{1}{m!} \lim_{\xi \to \xi_0} \frac{d^m}{d\xi^m} \left((\xi - \xi_0)^{m+1} f(\xi) k_n(\xi) \right). \tag{A.10}$$

Here, $k_n(\xi)$ is the characteristic remainder function, approximated as

$$k_n(\xi) \approx \frac{2\pi}{(\xi + \sqrt{\xi^2 - 1})^{2n+1}}, \tag{A.11}$$

where n is the order of the Gauss-Legendre quadrature rule used to approximate I, in this paper $n = 16$. They also showed that the derivatives of $k_n(\xi)$ can be approximated as

$$k_n^{(m)}(\xi) \approx k_n(\xi) \left(-\frac{2n+1}{\sqrt{\xi^2 - 1}} \right)^m, \tag{A.12}$$

for small m, and that

$$\lim_{\xi \to \xi_0} \frac{(\xi - \xi_0)^{m+1}}{(\eta(\xi) - \eta(\xi_0))^{m+1}} = \frac{1}{(\eta'(\xi_0))^{m+1}}.$$

For J_1, f_1 is a meromorphic function with a pole at ξ_0 of order one, and the estimate is defined by (A.10) as

$$\begin{aligned} R[f_1] &= \Im \left(\lim_{\xi \to \xi_0} (\xi - \xi_0) f_1(\xi) k_n(\xi) \right) \\ &\approx \Im \left(k_n(\xi_0) \mu(\xi_0) \right), \end{aligned}$$

where (A.12) is used and $k_n(\xi_0)$ can be approximated as in (A.11). Instead of finding the value of $\mu(\xi)$ at the parameter ξ_0, it is sufficient to consider the max-norm of μ on the panel Γ_i: $\|\mu^i\|_\infty$. Thus the error estimate for $J_1(z_0)$ becomes

$$R[f_1] \approx \Im \left(k_n(\xi_0) \|\mu^i\|_\infty \right). \tag{A.13}$$

Similarly, for the second integral J_2 the function f_2 is meromorphic with a pole of order one at ξ_0. The estimate reads

$$\begin{aligned} R[f_2] &= \lim_{\xi \to \xi_0} (\xi - \xi_0) f_2(\xi) k_n(\xi) \\ &\approx k_n(\xi_0) \mu(\xi_0) \overline{n(\xi_0)}^2 \\ &\approx k_n(\xi_0) \|\mu^i\|_\infty \overline{n(\xi_0)}^2. \end{aligned} \tag{A.14}$$

Again, (A.12) is used. Note that also the value of the normal n needs to be found at ξ_0 through the mapping $\eta(\xi)$.

Finally, regard the integral J_3. Here, f_3 is a meromorphic function with a pole of order two at ξ_0. Following (A.10), the estimate becomes

$$R[f_3] = \lim_{\xi \to \xi_0} \frac{d}{d\xi} \left((\xi - \xi_0)^2 f_3(\xi) k_n(\xi) \right).$$

Assuming a good approximation can be obtained by considering only the term with the derivative of k_n, see [19], this can be simplified into

$$\begin{aligned} R[f_3] &\approx k_n'(\xi_0)\frac{1}{\eta'(\xi_0)}\left(\overline{\eta(\overline{\xi_0})}-\overline{\eta(\xi_0)}\right)\mu(\xi_0) \\ &\approx k_n'(\xi_0)\frac{1}{\eta'(\xi_0)}\left(\overline{\eta(\overline{\xi_0})}-\overline{\eta(\xi_0)}\right)\|\mu\|_\infty. \end{aligned} \tag{A.15}$$

Together, the complete error estimate for the integrals in (A.2) over a panel Γ_i can be written as (A.9). This error estimate is shown in Fig. 3, where the estimate of errors of order 10^{-p} is marked with black level curves for $p=-14,\dots,0$ with increment 2.

Appendix B. Special quadrature

An overview of the special quadrature follows here, for more details see [28].

The special quadrature is employed on integrals of two types:

$$I_1(z)=\int_{\Gamma_i}\frac{f(\tau)d\tau}{\tau-z},$$

$$I_2(z)=\int_{\Gamma_i}\frac{f(\tau)d\tau}{(\tau-z)^2}.$$

The function $f(\tau)$ is interpolated on the panel Γ_i, as $f(\tau)\approx\sum_{j=0}^{15}c_j\tau^j$, where the coefficients c_j are computed through solving a Vandermonde system. For stability, the panel is transformed to have endpoints at -1 and 1. Using the above interpolation, the integrals I_1 and I_2 can be rewritten as

$$I_1(z)\approx\sum_{j=0}^{15}c_j\int_{-1}^{1}\frac{\tau^j d\tau}{\tau-z_0}=\sum_{j=0}^{15}c_j p_j, \tag{B.1}$$

$$I_2(z)\approx\alpha\sum_{j=0}^{15}c_j\int_{-1}^{1}\frac{\tau^j d\tau}{(\tau-z_0)^2}=\alpha\sum_{j=0}^{15}c_j q_j. \tag{B.2}$$

Here z_0 is the target point z under the same transformation as the one applied to the panel and $\alpha=\frac{2}{\tau_e-\tau_s}$, where τ_s, τ_e are the start and end points of the untransformed panel respectively. The numbers p_j and q_j can computed through recursion, where

$$\begin{cases} p_0 &= \int_{-1}^{1}\frac{d\tau}{\tau-z_0}=\log(1-z_0)-\log(-1-z_0), \\ p_j &= z_0 p_{j-1}+\frac{1-(-1)^j}{j},\ j=1,\dots,15, \end{cases} \tag{B.3}$$

and

$$\begin{cases} q_0 &= \int_{-1}^{1}\frac{d\tau}{(\tau-z_0)^2}=-\frac{1}{1+z_0}-\frac{1}{1-z_0}, \\ q_j &= z_0 q_{j-1}+p_j,\ j=1,\dots,15. \end{cases} \tag{B.4}$$

Note that if z_0 is within the contour created by the real axis from -1 to 1 and the transformed panel Γ_i, a residue of $2\pi i$ must be added or subtracted from p_0 depending on if z_0 has a positive or negative imaginary part respectively.

To apply the special quadrature on the integrals in (15) and (16), they need to be rewritten into the forms of I_1 and I_2. The second integral can be reformulated as in (A.5), i.e.

$$\int_\Gamma\overline{\mu(\tau)}\frac{\Im\left\{(\overline{\tau}-\overline{z_0})d\tau\right\}}{(\overline{\tau}-\overline{z_0})^2}=\frac{i}{2}\int_\Gamma\overline{\frac{\mu(\tau)\bar{n}_\tau^2 d\tau}{\tau-z_0}}+\frac{i}{2}\int_\Gamma\overline{\frac{\mu(\tau)(\overline{\tau}-\overline{z_0})}{(\tau-z_0)^2}}.$$

All integrals are then on the required form, with $f_1(\tau)=\mu(\tau)\tau'$, $f_2(\tau)=\mu(\tau)\bar{n}_\tau^2\tau'$ and $f_3(\tau)=\mu(\tau)(\overline{\tau}-\overline{z})\tau'$ respectively. The improved error computed with 50 Gauss-Legendre panels can be seen in Fig. 4. Using the special quadrature it is possible to evaluate $u(z)$ for points close to the boundary, with errors below 10^{-10}.

References

[1] S.L. Anna, Droplets and bubbles in microfluidic devices, Annu. Rev. Fluid Mech. 48 (1) (2016) 285–309, https://doi.org/10.1146/annurev-fluid-122414-034425.

[2] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (2–3) (1997) 151–167, https://doi.org/10.1016/S0168-9274(97)00056-1.

[3] I.B. Bazhlekov, P.D. Anderson, H.E.H. Meijer, Numerical investigation of the effect of insoluble surfactants on drop deformation and breakup in simple shear flow, J. Colloid Interface Sci. 298 (1) (2006) 369–394, https://doi.org/10.1016/j.jcis.2005.12.017.

[4] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (2) (1992) 335–354, https://doi.org/10.1016/0021-9991(92)90240-Y.

[5] L. Bystricky, S. Shanbhag, B.D. Quaife, Stable and contact-free time stepping for dense rigid particle suspensions, preprint, arXiv:1804.11012.

[6] C. Cleret de Langavant, A. Guittet, M. Theillard, F. Temprano-Coleto, F. Gibou, Level-set simulations of soluble surfactant driven flows, J. Comput. Phys. 348 (2017) 271–297, https://doi.org/10.1016/j.jcp.2017.07.003.

[7] V. Cristini, Y.-C. Tan, Theory and numerical simulation of droplet dynamics in complex flows a review, Lab Chip 4 (4) (2004) 257–264, https://doi.org/10.1039/B403226H.

[8] D. Crowdy, S. Tanveer, G.L. Vasconcelos, On a pair of interacting bubbles in planar Stokes flow, J. Fluid Mech. 541 (2005) 231, https://doi.org/10.1017/S0022112005005999.

[9] L. Greengard, J.-Y. Lee, Accelerating the nonuniform fast Fourier transform, SIAM Rev. 46 (3) (2004) 443–454, https://doi.org/10.1137/S003614450343200X.

[10] L. Greengard, V. Rokhlin, A new version of the Fast Multipole Method for the Laplace equation in three dimensions, Acta Numer. 6 (1997) 229, https://doi.org/10.1017/S0962492900002725.

[11] E. Hairer, S.P. Norsett, G. Wanner, Solving Ordinary Differential Equations. I: Nonstiff Problems, second edition, Springer, Berlin, ISBN 978-3-540-56670-0, Jan. 1993.

[12] P. Hansbo, M.G. Larson, S. Zahedi, A cut finite element method for a Stokes interface problem, Appl. Numer. Math. 85 (2014) 90–114, https://doi.org/10.1016/j.apnum.2014.06.009.

[13] J. Helsing, R. Ojala, On the evaluation of layer potentials close to their sources, J. Comput. Phys. 227 (5) (2008) 2899–2921, https://doi.org/10.1016/j.jcp.2007.11.024.

[14] T.Y. Hou, J.S. Lowengrub, M.J. Shelley, Removing the stiffness from interfacial flows with surface tension, J. Comput. Phys. (114) (1994) 312–338.

[15] S. Khatri, A.-K. Tornberg, A numerical method for two phase flows with insoluble surfactants, Comput. Fluids 49 (1) (2011) 150–165, https://doi.org/10.1016/j.compfluid.2011.05.008.

[16] S. Khatri, A.-K. Tornberg, An embedded boundary method for soluble surfactants with interface tracking for two-phase flows, J. Comput. Phys. 256 (2014) 768–790, https://doi.org/10.1016/j.jcp.2013.09.019.

[17] L. af Klinteberg, A.-K. Tornberg, A fast integral equation method for solid particles in viscous flow using quadrature by expansion, J. Comput. Phys. 326 (2016) 420–445, https://doi.org/10.1016/j.jcp.2016.09.006.

[18] L. af Klinteberg, A.-K. Tornberg, Error estimation for quadrature by expansion in layer potential evaluation, Adv. Comput. Math. 43 (1) (2017) 195–234, https://doi.org/10.1007/s10444-016-9484-x.

[19] L. af Klinteberg, A.-K. Tornberg, Adaptive quadrature by expansion for layer potential evaluation in two dimensions, SIAM J. Sci. Comput. 40 (3) (2018) A1225–A1249, https://doi.org/10.1137/17M1121615.

[20] A. Klöckner, A. Barnett, L. Greengard, M. O'Neil, Quadrature by expansion: a new method for the evaluation of layer potentials, J. Comput. Phys. 252 (2013) 332–349, https://doi.org/10.1016/j.jcp.2013.06.027.

[21] M.C.A. Kropinski, An efficient numerical method for studying interfacial motion in two-dimensional creeping flows, J. Comput. Phys. 171 (2) (2001) 479–508, https://doi.org/10.1006/jcph.2001.6787.

[22] M.C.A. Kropinski, E. Lushi, Efficient numerical methods for multiple surfactant-coated bubbles in a two-dimensional Stokes flow, J. Comput. Phys. 230 (12) (2011) 4466–4487, https://doi.org/10.1016/j.jcp.2011.02.019.

[23] W.E. Langlois, M.O. Deville, Slow Viscous Flow, second edition, Springer, ISBN 9783319038346, 2014.

[24] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (4) (1994) 1019–1044, https://doi.org/10.1137/0731054.

[25] G.R. Marple, A. Barnett, A. Gillman, S.K. Veerapaneni, A fast algorithm for simulating multiphase flows through periodic geometries of arbitrary shape, SIAM J. Sci. Comput. 38 (5) (2016) B740–B772, https://doi.org/10.1137/15M1043066.

[26] W.J. Milliken, H.A. Stone, L.G. Leal, The effect of surfactant on the transient motion of Newtonian drops, Phys. Fluids A, Fluid Dyn. 5 (1) (1993) 69–79, https://doi.org/10.1063/1.858790.

[27] M. Muradoglu, G. Tryggvason, Simulations of soluble surfactants in 3D multiphase flow, J. Comput. Phys. 274 (2014) 737–757, https://doi.org/10.1016/j.jcp.2014.06.024.

[28] R. Ojala, A.-K. Tornberg, An accurate integral equation method for simulating multi-phase Stokes flow, J. Comput. Phys. 298 (2015) 145–160, https://doi.org/10.1016/j.jcp.2015.06.002.

[29] Y. Pawar, K.J. Stebe, Marangoni effects on drop deformation in an extensional flow: the role of surfactant physical chemistry. I. Insoluble surfactants, Phys. Fluids 8 (7) (1996) 1738–1751, https://doi.org/10.1063/1.868958.

[30] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (Nov. 1977) 220–252, https://doi.org/10.1016/0021-9991(77)90100-0, ISSN 00219991.

[31] C. Pozrikidis, Interfacial dynamics for Stokes flow, J. Comput. Phys. 169 (2) (2001) 250–301, https://doi.org/10.1006/jcph.2000.6582.

[32] C. Pozrikidis, X. Li, The effect of surfactants on drop deformation and on the rheology of dilute emulsions in Stokes flow, J. Fluid Mech. 341 (1997) 165–194.

[33] S. Pålsson, C. Sorgentone, A.-K. Tornberg, Adaptive time-stepping for surfactant-laden drops, in: David J. Chappel (Ed.), On Boundary Integral Method (UKBIM11), Nottingham Trent University: Publications, Nottingham, 2017, pp. 161–170.

[34] B. Quaife, G. Biros, Adaptive time stepping for vesicle suspensions, J. Comput. Phys. 306 (Feb. 2016) 478–499, https://doi.org/10.1016/j.jcp.2015.11.050.

[35] J.M. Rallison, The deformation of small viscous drops and bubbles in shear flows, Annu. Rev. Fluid Mech. 16 (1) (1984) 45–66, https://doi.org/10.1146/annurev.fl.16.010184.000401.

[36] S. Shin, J. Chergui, D. Juric, L. Kahouadji, O.K. Matar, R.V. Craster, A hybrid interface tracking – level set technique for multiphase flow with soluble surfactant, J. Comput. Phys. 359 (2018) 409–435, https://doi.org/10.1016/j.jcp.2018.01.010.

[37] M. Siegel, Influence of surfactant on rounded and pointed bubbles in two-dimensional Stokes flow, SIAM J. Appl. Math. 59 (6) (1999) 1998–2027, https://doi.org/10.1137/S0036139997327435.

[38] C. Sorgentone, A.-K. Tornberg, A highly accurate boundary integral equation method for surfactant-laden drops in 3D, J. Comput. Phys. 360 (2018) 167–191, https://doi.org/10.1016/j.jcp.2018.01.033.

[39] H.A. Stone, A simple derivation of the time–dependent convective–diffusion equation for surfactant transport along a deforming interface, Phys. Fluids A, Fluid Dyn. 2 (1) (1990) 111–112, https://doi.org/10.1063/1.857686.

[40] H.A. Stone, L.G. Leal, The effects of surfactants on drop deformation and breakup, J. Fluid Mech. 220 (1990) 161, https://doi.org/10.1017/S0022112090003226.
[41] S. Tanveer, G.L. Vasconcelos, Time-evolving bubbles in two-dimensional Stokes flow, J. Fluid Mech. 301 (1995) 325, https://doi.org/10.1017/S0022112095003910.
[42] S.-Y. Teh, R. Lin, L.-H. Hung, A.P. Lee, Droplet microfluidics, Lab Chip 8 (2) (2008) 198, https://doi.org/10.1039/b715524g.
[43] K.E. Teigen, P. Song, J. Lowengrub, A. Voigt, A diffuse-interface method for two-phase flows with soluble surfactants, J. Comput. Phys. 230 (2) (2011) 375–393, https://doi.org/10.1016/j.jcp.2010.09.020.
[44] J.-J. Xu, Z. Li, J. Lowengrub, H. Zhao, A level set method for interfacial flows with surfactants, J. Comput. Phys. 212 (2) (2006) 590–616.
[45] J.-J. Xu, W. Shi, M.-C. Lai, A level-set method for two-phase flows with soluble surfactant, J. Comput. Phys. 353 (Jan. 2018) 336–355, https://doi.org/10.1016/j.jcp.2017.10.019.
[46] K. Xu, M.R. Booty, M. Siegel, Analytical and computational methods for two-phase flow with soluble surfactant, SIAM J. Appl. Math. 73 (1) (2013) 523–548, https://doi.org/10.1137/120881944.