Journal of Computational Physics 352 (2018) 615–636

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Direct numerical simulation of variable surface tension flows using a Volume-of-Fluid method

Ivana Seric, Shahriar Afkhami *, Lou Kondic

Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ, USA

ARTICLE INFO

Article history:
Received 21 February 2017
Received in revised form 1 October 2017
Accepted 5 October 2017

Keywords:
Direct Numerical Simulation (DNS)
Surface tension
Surface gradient
Marangoni
Volume-of-Fluid (VOF) method
Height function method

ABSTRACT

We develop a general methodology for the inclusion of a variable surface tension coefficient into a Volume-of-Fluid based Navier–Stokes solver. This new numerical model provides a robust and accurate method for computing the surface gradients directly by finding the tangent directions on the interface using height functions. The implementation is applicable to both temperature and concentration dependent surface tension coefficient, along with the setups involving a large jump in the temperature between the fluid and its surrounding, as well as the situations where the concentration should be strictly confined to the fluid domain, such as the mixing of fluids with different surface tension coefficients. We demonstrate the applicability of our method to the thermocapillary migration of bubbles and the coalescence of drops characterized by a different surface tension coefficient.

1. Introduction

Flows induced by the spatial variations in the surface tension coefficient, also known as the Marangoni effect [1], can be caused by surfactants, temperature or concentration gradients, or a combination of these effects. Understanding these flows is important since they are relevant in microfluidics [2], heat pipe flows [3], motion of drops or bubbles in materials processing applications that include heating or cooling [4], evolution of metal films of nanoscale thickness melted by laser pulses [5,6], and in a variety of other thin film flows, see [7,8] for reviews.

Numerical methods for studying variable surface tension flows include front tracking [9], level set [10], diffuse interface [11], marker particle [12,13], immersed boundary [14], boundary integral [15], interface-interaction [16], and Volume-of-Fluid (VOF) [17–19] methods. The VOF method is efficient and robust for tracking the topologically complex evolving interfaces. The improvements in the recent years in the computation of the surface tension force have empowered the VOF method to become a widespread method for modeling interfacial flows [20,21]. However, an accurate implementation of the variable surface tension coefficient in the VOF formulation is still lacking a general treatment.

A challenge of including the variable surface tension effects into the VOF method is that the surface tension coefficient is not known exactly at the interface — only the value averaged over a computational cell containing the interface is known. To obtain the surface tension coefficient at the interface, an approximation from the values near the interface, usually calculated at the center of each adjacent computational cell, is necessary. As we outline below, the approximation of the

* Corresponding author.
E-mail address: shahriar.afkhami@njit.edu (S. Afkhami).

https://doi.org/10.1016/j.jcp.2017.10.008
0021-9991/

interface values has been carried out in the literature differently, depending on the physics of the problem studied. An additional major issue concerns computing the surface gradients of the surface tension coefficient.

In Alexeev et al. [22] and Ma and Bothe [19], the VOF method is used to study flows involving a temperature dependent surface tension coefficient. The implementation in Alexeev et al. [22] solves the heat equation in fluids on both sides of the interface, and then imposes the continuity of the temperature and flux at the interface, and the conservation of energy in the cell containing the interface to approximate the temperature in the fluid and air in the cell. These temperature values are then used to calculate surface gradients of the temperature from nearby cells that are not cut by the interface; these gradients are then exponentially extrapolated to the interface. In the work by Ma and Bothe [19], the temperature at the interface is approximated from the temperatures in the liquid and the gas by imposing the continuity of the heat flux at the interface. The surface gradients of the temperature are approximated by computing the derivatives in each coordinate direction using finite differences, and then projecting them onto the tangential direction. If the interface is not contained in all cells of the finite difference stencil, then one sided differences are used. Hence, this method requires the temperature solution on both sides of the interface and therefore cannot be used for the setups involving a large difference in thermal conductivity of the two fluids, since the fluids may have a large difference in the temperature. Furthermore, both of these methods are not applicable to setups where the surface tension only depends on the concentration, such as mixing of miscible liquids with a different surface tension coefficient. In the work by James and Lowengrub [18], the VOF method is used to study the flows induced by the surfactant concentration gradient. In their method, the concentration values at the interface are obtained by imposing the condition that the average concentration at the interface is equal to the average concentration in the cell containing the interface. Then, the surface gradients are computed using the cell-center interfacial concentration in the two adjacent cells.

Here, we develop a method that can be applied to both temperature and concentration dependent surface tension coefficient, with the surface gradients computed using the cell-center values in the interfacial cells only. We note that comparing to the methods by Alexeev et al. [22] and Ma and Bothe [19], our approach is simpler, while retaining the accuracy, in that it uses interfacial cells only to compute the temperature at the interface, while these existing methods use interpolation around the interface for liquid and air temperature. We find the tangential gradients directly by computing the tangent directions on the interface using height functions [23]. We therefore note that our numerical method is consistent with underlying height-function method used for computing the surface tension force in the original method in [23].

Another advantage of our numerical method is its ability to naturally and automatically handle the surface tension variation that may result from either thermal or concentration gradients. Our method can therefore seamlessly be applied to the setups involving the concentration confined to the fluid domain, e.g. mixing of liquids with a different surface tension coefficients, as well as the configurations involving a large jump of the temperature between the liquid and the surrounding.

Our numerical method is implemented using Gerris: an open source adaptive Navier–Stokes solver [24,23]. The current version includes Continuum Surface Force (CSF) [25] implementation of the surface tension force with height function algorithm for computing the interfacial normal and curvature [23]. Here, we present a method for extending this formulation to include a variable surface tension coefficient, allowing to consider the surface force in the direction tangential to the interface. As far as we are aware, this is the first implementation of the variable surface tension combined with the accurate implementation of the CSF method, such that the curvature and interface normals are computed using generalized height functions [23]. In addition, we demonstrate that our new numerical method results in second order accurate computation of the surface tension gradients in the tangential direction which is a key element for describing the variable surface tension problems accurately. Such a result has not yet been reported in the literature. With the above mentioned improvements, our extension is a step closer to cover all aspects of the variable surface tension flows; the remaining one is the implementation of the surfactant transport and the surface tension gradients due to the presence of the surfactants. This will be a topic of our future work.

The rest of this paper is organized as follows: Section 2 gives an overview of the VOF method, including the CSF method for the computation of the surface tension; Section 3 describes in detail the implementation of the variable surface tension coefficient in two and three dimensions; and Section 4 illustrates the performance of our method for various test cases, including temperature and concentration dependent surface tension.

2. Governing equations

We consider an incompressible two-phase flow described by the Navier–Stokes equations

$$\rho(\partial_t \mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \nabla\cdot(2\mu D) + \mathbf{F}, \tag{1}$$

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

and the advection of the phase-dependent density $\rho(\chi)$

$$\partial_t \rho + (\mathbf{u}\cdot\nabla)\rho = 0, \tag{3}$$

where $\mathbf{u} = (u, v, w)$ is the fluid velocity, p is the pressure, $\rho(\chi) = \chi\rho_1 + (1-\chi)\rho_2$ and $\mu(\chi) = \chi\mu_1 + (1-\chi)\mu_2$ are the phase dependent density and viscosity respectively, and D is the rate of deformation tensor $D = \left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)/2$. Subscripts 1 and 2 correspond to the fluids 1 and 2, respectively (see Fig. 1). Here, χ is the characteristic function, such that $\chi = 1$ in

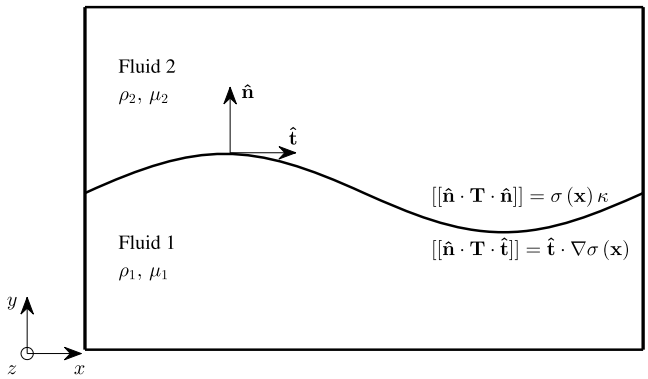

Fig. 1. The schematic of a system with two immiscible fluids and the corresponding boundary conditions.

the fluid 1, and $\chi = 0$ in the fluid 2. Note that any body force can be included in $\mathbf{F}$. The characteristic function is advected with the flow, thus

$$\partial_t \chi + (\mathbf{u}\cdot\nabla)\chi = 0. \tag{4}$$

Note that solving equation (4) is equivalent to solving equation (3).

The presence of an interface gives rise to the stress boundary conditions, see Fig. 1. The normal stress boundary condition at the interface defines the stress jump [26,27]

$$[\![\hat{\mathbf{n}}\cdot\mathbf{T}\cdot\hat{\mathbf{n}}]\!] = \sigma(\mathbf{x})\kappa, \tag{5}$$

where $\mathbf{T} = -p\mathbf{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$ is the total stress tensor, $\sigma(\mathbf{x})$ is the surface tension coefficient, κ is the curvature of the interface, and $\hat{\mathbf{n}}$ is the unit normal at the interface pointing out of the fluid 1. The variation of the surface tension coefficient results in the tangential stress jump at the interface

$$[\![\hat{\mathbf{n}}\cdot\mathbf{T}\cdot\hat{\mathbf{t}}]\!] = \hat{\mathbf{t}}\cdot\nabla\sigma(\mathbf{x}), \tag{6}$$

which drives the flow from the regions of a low surface tension coefficient to the ones with a high surface tension coefficient. Here, $\hat{\mathbf{t}}$ is the unit tangent vector in two dimensions (2D); in three dimensions (3D) there are two linearly independent unit tangent vectors. Using the Continuum Surface Force (CSF) method [25], the forces resulting from the normal and tangential stress jump at the interface can be included in the body force $\mathbf{F} = \mathbf{F}_{sn} + \mathbf{F}_{st}$, defined as

$$\mathbf{F}_{sn} = \sigma(\mathbf{x})\kappa\delta_s\hat{\mathbf{n}}, \tag{7}$$

and

$$\mathbf{F}_{st} = \nabla_s\sigma(\mathbf{x})\delta_s, \tag{8}$$

where δ_s is the Dirac delta function centered at the interface, $\delta_s\hat{\mathbf{n}} = \nabla\chi$, and ∇_s is the surface gradient. Substituting equations (7) and (8) in the momentum equation (1) gives

$$\rho(\partial_t\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \nabla\cdot(2\mu D) + \sigma(\mathbf{x})\kappa\delta_s\hat{\mathbf{n}} + \nabla_s\sigma(\mathbf{x})\delta_s. \tag{9}$$

We define the nondimensional variables, denoted with a superscript "*", as

$$x^* = \frac{x}{a},\quad t^* = \frac{t}{t_r},\quad \mathbf{u}^* = \frac{\mathbf{u}}{U_0},\quad p^* = \frac{p}{p_0},$$
$$\rho^*(\chi) = \frac{\rho(\chi)}{\rho_1},\quad \mu^*(\chi) = \frac{\mu(\chi)}{\mu_1},\quad \sigma^* = \frac{\sigma}{\sigma_0},$$

where the scales a, p_0, t_c, U_0 and σ_0 are chosen based on the problem studied. Hence the dimensionless equation (9) is

$$\mathrm{Re}\rho^*(\partial_t^*\mathbf{u}^* + \mathbf{u}^*\cdot\nabla^*\mathbf{u}^*) = -\nabla^* p^* + \nabla^*\cdot\left(2\mu^* D^*\right) + \mathrm{Ca}^{-1}\sigma^*\kappa^*\delta_s^*\hat{\mathbf{n}} + \frac{\sigma_0}{U_0\mu_1}\nabla_s^*\sigma^*\delta_s^*\hat{\mathbf{t}}, \tag{10}$$

where Re and Ca are the Reynolds and Capillary numbers respectively, defined as

$$\mathrm{Re} = \frac{\rho_1 U_0 a}{\mu_1},\quad \mathrm{Ca} = \frac{U_0\mu_1}{\sigma_0}. \tag{11}$$

The surface tension coefficient is a function of temperature, T, or concentration, C, which satisfy the advection diffusion equation

$$\rho(\chi) C_p(\chi) (\partial_t T + (\mathbf{u}\cdot\nabla)T) = \nabla\cdot(k(\chi)\nabla T), \tag{12}$$

$$\partial_t C + (\mathbf{u}\cdot\nabla)C = \nabla\cdot(\alpha(\chi)\nabla C), \tag{13}$$

where $C_p(\chi)$, $k(\chi)$ and $\alpha(\chi)$ are the phase dependent heat capacity, conductivity and diffusivity, respectively. We note that since our method does not depend on whether we consider temperature or concentration gradients, we will use them interchangeably in the remaining part of the paper.

Along with the scales given above, equations (12) and (13) are nondimensionalized using the following scales

$$k^*(\chi) = \frac{k(\chi)}{k_1}, \quad C_p^*(\chi) = \frac{C_p(\chi)}{C_{p_1}}, \quad T^* = \frac{T}{T_0}, \quad \alpha^*(\chi) = \frac{\alpha(\chi)}{\alpha_1}, \tag{14}$$

where T_0 is chosen based on the physics of the system. Hence the dimensionless equations (12) and (13) are

$$\mathrm{Ma}\rho^* C_p^* \left(\partial_t^* T^* + (\mathbf{u}^*\cdot\nabla^*)T^*\right) = \nabla^*\cdot\left(k^*\nabla^* T^*\right), \tag{15}$$

$$\mathrm{Ma}\left(\partial_t^* C^* + (\mathbf{u}^*\cdot\nabla^*)C^*\right) = \nabla^*\cdot\left(\alpha^*\nabla^* C^*\right), \tag{16}$$

where Ma is the Marangoni number defined as

$$\mathrm{Ma} = \frac{U_0 a}{\alpha_1}. \tag{17}$$

The diffusivity, α_1, in the heat equation is $\alpha_1 = k_1/(\rho_1 C_{p_1})$. Surface tension can have a linear or nonlinear dependence on the temperature or concentration. In many applications the surface tension coefficient depends on the temperature linearly, i.e.

$$\sigma = \sigma_0 + \sigma_T (T - T_R), \tag{18}$$

where σ_0 is the surface tension coefficient at a reference temperature T_R, and σ_T is a constant. Then, we can write

$$\nabla_s\sigma = \sigma_T \nabla_s T, \tag{19}$$

and compute $\nabla_s T$ in the same manner as $\nabla_s\sigma$. Using the scales given above, the dimensionless equation (18) is

$$\sigma^* = 1 + \frac{\sigma_T T_0}{\sigma_0}\left(T^* - T_R^*\right). \tag{20}$$

In the following section, we describe a method for computing $\nabla_s\sigma$ in general, regardless of the dependence on the temperature or concentration.

3. Numerical method

The proposed numerical method is implemented into Gerris, which numerically solves equations (1) to (3) using the VOF interface tracking method with an implicit treatment of the viscous forces [23,24,28]. The CSF method is used for the implementation of the surface tension force with curvatures computed using the height function method [23,29]. The Gerris code uses octree (3D) and quadtree (2D) grids, allowing to adaptively refine the grid in the immediate neighborhood of the interface. While we describe our implementation of the variable surface tension coefficient for uniform meshes, the extension to adaptively refined meshes is straightforward, following the implementation details described by Popinet [23,24].

The surface gradient of any scalar field Q is defined as the projection of the gradient onto the surface, i.e.

$$\nabla_s Q = \nabla Q - \hat{\mathbf{n}}\left(\hat{\mathbf{n}}\cdot\nabla Q\right) \tag{21}$$

where $\hat{\mathbf{n}}$ is the unit normal vector at the surface. However, this definition of the surface gradient can result in inaccuracies when implemented in the VOF method for general variable surface tension coefficient for two reasons. First, the discontinuities of the material properties across the interface can result in Q having a large jump across the interface: for example, in the case of surface tension coefficient dependence on the temperature where the fluids on each side of the interface have a large difference in the conductivity. The second reason is that, in general, surface tension coefficient can depend on the concentration: for example, in the case of the mixing of two liquids with different surface tension coefficients, or in the case of surface tension coefficient dependent on the surfactant concentration.

Here, we propose a numerical method for implementing a general variable surface tension coefficient. We compute the surface gradient as

$$\mathbf{F}_{st} = \frac{\partial\sigma}{\partial s_1}\delta_s\hat{\mathbf{t}}_1 + \frac{\partial\sigma}{\partial s_2}\delta_s\hat{\mathbf{t}}_2, \tag{22}$$

where $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$ are the unit tangent vectors at the interface, pointing in the s_1 and s_2 directions, respectively. In our approach, we first define the surface tension coefficient values at the interface, then compute the derivatives of σ along the

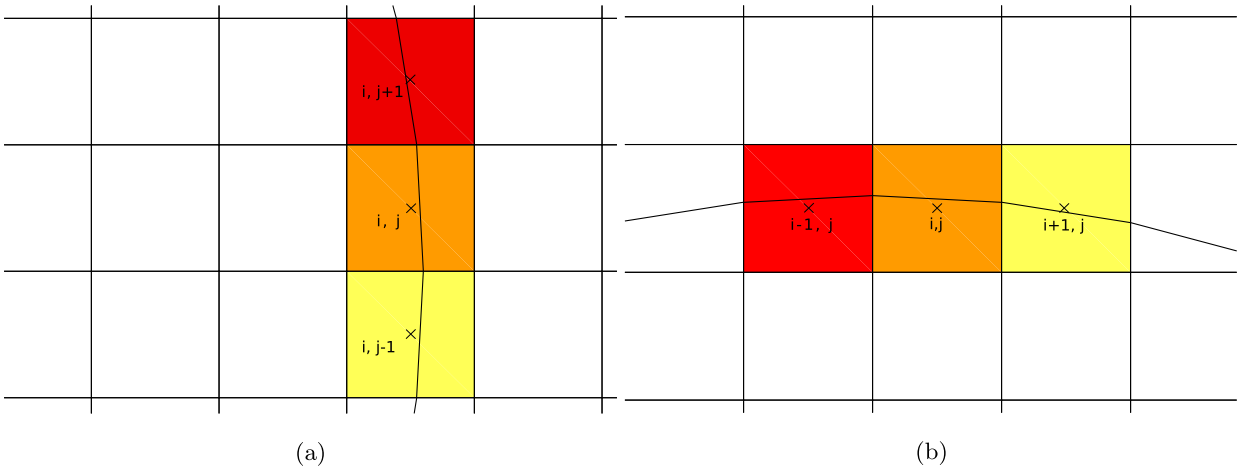

Fig. 2. An example of the interface orientation, where columns in the x (a) and the y (b) direction for computing $\tilde{\sigma}^c(\mathbf{x})$ contain one interfacial cell. Each color shows a different column in interfacial cells $\mathcal{C}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

interface, and finally project the derivatives onto the tangent space defined by $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$. In the following sections we present the details of the implementation. In Section 3.1, we show how to approximate the value of the surface tension coefficient on the interface using the cell-center values. Then in Section 3.2, we show how $\partial\sigma/\partial s_d$, for $d = 1, 2$, are evaluated, along with the choice of the tangent vectors and addition of the tangential surface force using the CSF method. This is done first for 2D is Section 3.2.1, and then for 3D in Section 3.2.2.

3.1. Approximation of the interfacial values of the surface tension coefficient

The algorithm for implementing $\nabla_s\sigma(\mathbf{x})$ in the VOF method starts with the approximation of the interfacial values of the surface tension in each cell containing an interface segment. More precisely, we use the idea of constructing the columns of cells inspired by the computation of interfacial curvature and normals using height functions [23] (see A.1). We should note however that to obtain a consistent height-function approximation, the interface must be sufficiently resolved. Clearly, we cannot expect reasonable accuracy for the computation of the surface tension gradient in the tangential direction at very low resolutions, such that the interface becomes very poorly resolved. In such cases, one should resort to interpolation methods described by Popinet [23]. In Section 4.2 we discuss via examples the spatial resolution required for application of a consistent height-function approximation.

Let $\sigma(\mathcal{C})$ be the surface tension coefficient evaluated from the temperature or concentration at the center of all interfacial cells $\mathcal{C}$, with the volume fraction $\chi(\mathcal{C})$. The surface tension coefficient in each column, denoted by $\tilde{\sigma}^c(\mathbf{x})$, is defined so that it has only one value in each column, regardless of how many interfacial cells are contained in that column. For illustration, Fig. 2 shows columns that contain only one interfacial cell, and Fig. 3 shows columns that contain more than one interfacial cell, where the same color denotes cells in the same column. The superscript, $c = x, y, z$, represents the column direction. For simplicity, here we show examples of the implementation in 2D, however, the algorithm extends trivially to 3D.

For columns with only one interfacial cell (see Fig. 2(a) and (b) for the columns in the x and y direction, respectively), the surface tension of the interfacial cells, $\tilde{\sigma}^c$, is equal to the surface tension $\sigma(\mathcal{C})$ in the same cells. If there is more than one interfacial cell in the column, then $\tilde{\sigma}^c$ is approximated by the volume weighted average of the $\sigma(\mathcal{C})$ values. In Fig. 3, the cells labeled with the cell indices will be used for computing $\tilde{\sigma}^c$ for columns in the x and y directions — Figs. 3(a) and (b) respectively. For example, in Fig. 3(a), the $\tilde{\sigma}^x$ is computed using the columns in the x direction, and the value of $\tilde{\sigma}^x$ in the column containing cell $\mathcal{C}_{i,j}$, denoted $\tilde{\sigma}^x_j$, is

$$\tilde{\sigma}^x_j = \frac{\chi_{i,j}\sigma_{i,j} + \chi_{i+1,j}\sigma_{i+1,j}}{\sum \chi_i}. \tag{23}$$

Note that the cells in the same column, in this particular example cells $\mathcal{C}_{i,j}$ and $\mathcal{C}_{i+1,j}$, have the same value of $\tilde{\sigma}^x$. For the columns in the y direction, as in Fig. 3(b), $\tilde{\sigma}^y$ in the column containing cell $\mathcal{C}_{i,j}$, denoted $\tilde{\sigma}^y_i$, is computed as

$$\tilde{\sigma}^y_i = \frac{\chi_{i,j}\sigma_{i,j} + \chi_{i,j+1}\sigma_{i,j+1}}{\sum \chi_i}. \tag{24}$$

Again, the cells in the same column, in this case $\mathcal{C}_{i,j}$ and $\mathcal{C}_{i,j+1}$ have the same value of $\tilde{\sigma}^y$.

In our implementation, we first define $\tilde{\sigma}^c$ for all c in all interfacial cells. For certain interface orientations, it is possible to define $\tilde{\sigma}^c$ for columns in more than one direction, e.g. the interface in Fig. 3. However, this is not always the case, e.g. in

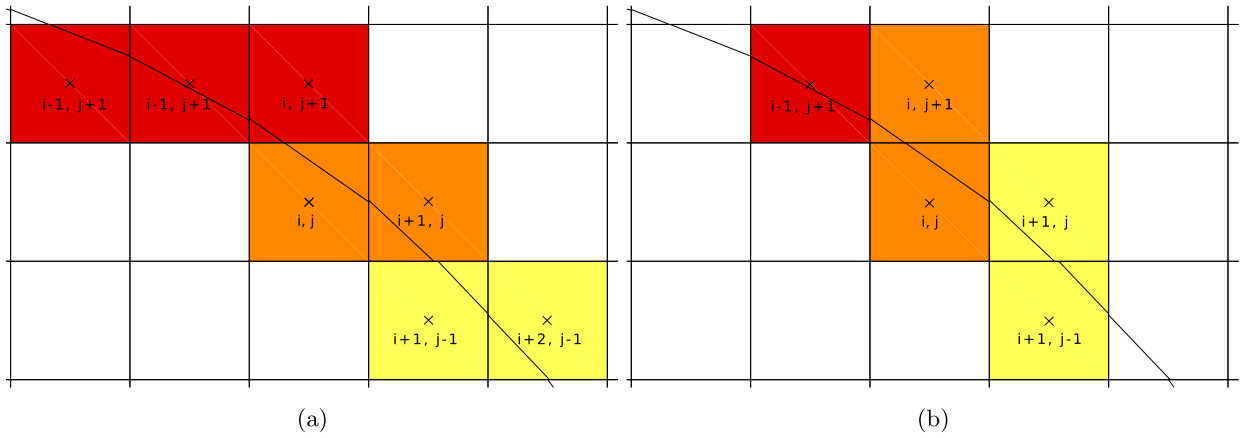

Fig. 3. An example of the interface orientation where columns in the x (a) and y (b) direction for computing $\tilde{\sigma}^c$ contain more than one interfacial cell. The cells with the same color belong to the same column. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 2(a) we can only compute $\tilde{\sigma}^x$, and in Fig. 2(b) we can only compute $\tilde{\sigma}^y$. For the former case, in the following sections we describe how the direction of the columns is chosen along with the discussion of the computation of the surface forces.

3.2. The computation of the surface forces

The next step in the variable surface force implementation is the evaluation of the derivatives along the interface, $\partial\sigma/\partial s_d$ in equation (22). In 2D, we only need to compute the derivative in one direction, since the basis for a tangent line consists of only one vector. However, in 3D, we need two tangent vectors to form a basis for the tangent space, hence we need to evaluate the derivative in two directions. We now discuss the implementation of the method for 2D and 3D.

3.2.1. The surface force in 2D

In 2D, equation (22) simplifies to

$$\mathbf{F}_{st} = \frac{\partial\sigma}{\partial s^c}\delta_s\hat{\mathbf{t}}, \tag{25}$$

since we only have one tangential direction. We remind the reader that in this case, $c = x, y$. The derivative of the surface tension coefficient along the interface, $\partial\sigma/\partial s^c$, is approximated by the derivative of the interfacial value, $\tilde{\sigma}^c$ in the column which is formed in the direction c. The choice of the direction, c, is based on the interface orientation: c is chosen to be the same as the largest component of the normal vector to the interface. The same choice is made for computing curvature and the interface normal using the height functions [23].

In each interfacial cell, we compute the derivative along the interface using the center difference, i.e. the finite difference of the $\tilde{\sigma}^c$ in the two neighboring columns. For example, in Fig. 2(a) and 3(a), the derivative is computed with respect to the y direction, as

$$\left(\frac{\partial\sigma}{\partial s^x}\right)_{i,j} = \frac{\tilde{\sigma}^x_{j+1} - \tilde{\sigma}^x_{j-1}}{\mathrm{d}s}. \tag{26}$$

As a reminder, $\tilde{\sigma}^x_j$ is the interfacial value of the surface tension in the column j constructed in the x direction. The arc length, $\mathrm{d}s$, is computed from the height function in the same direction as $\partial\sigma/\partial s^c$. For the example given in equation (26), the arc length is

$$\mathrm{d}s = 2\Delta\sqrt{1 + h_y}, \tag{27}$$

where h_y is the derivative of the height function with respect to the y direction (see A.1) and Δ is the cell size.

The next part of the surface gradient implementation is the choice of the tangent vector, $\hat{\mathbf{t}}$, which is computed so that it satisfies $\hat{\mathbf{t}} \cdot \hat{\mathbf{n}} = 0$, where $\hat{\mathbf{n}}$ is found using the Mixed Young's Center (MYC) method by Aulisa et al. [30]. We use the MYC method mainly due to its computational efficiency when compared to the height function method for computing the normals. This advantage is particularly noticeable in 3D. The direction of $\hat{\mathbf{t}}$ depends on the direction used for computing $\partial\sigma/\partial s^c$: $\hat{\mathbf{t}}$ points in the direction of the positive component orthogonal to the c direction. For example, $\hat{\mathbf{t}}$ points in the positive x direction if we construct columns in the y direction.

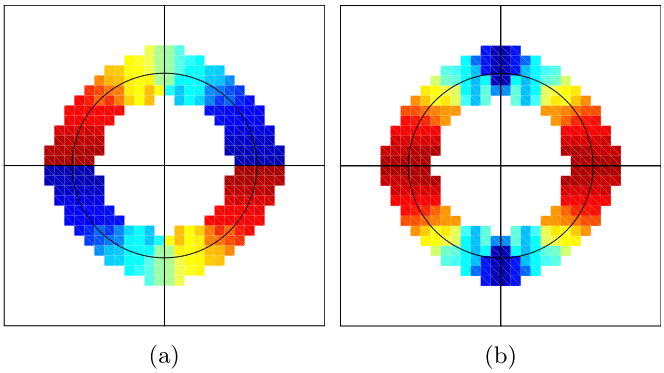

Fig. 4. The x (a) and y (b) component of **G**, with values around the interfacial cells defined by averaging the neighboring cells.

We have defined $\partial\sigma/\partial s^c$ in the interfacial cells. However, in the CSF method [25], we need to know $\partial\sigma/\partial s^c$ in the cells around the interface, i.e. in all the cells where $\delta_s = ||\nabla\chi||_2$ is nonzero. Consider an intermediate value of the surface force, **G**, which includes the correct sign of the surface force, but excludes the magnitude of the tangential vector, i.e.

$$G_x = \frac{\partial\sigma}{\partial s^c}\mathrm{sgn}(t_x), \tag{28}$$

$$G_y = \frac{\partial\sigma}{\partial s^c}\mathrm{sgn}(t_y). \tag{29}$$

The intermediate force **G** can be defined in the cells around the interface by using the same approach as for defining the curvature in the cells around the interface [28], i.e. the values in the cells neighboring the interfacial cells are defined by averaging the values in the direct neighbors that already have the curvature value defined. This procedure is repeated twice, insuring that the curvature values for the corner neighbors to the interfacial cells are defined as well. The x and y components of **G** in the cells around the interface are subsequently used to compute the surface force, defined in equation (25), as

$$F_x = G_x|t_x|\,\delta_s, \tag{30}$$

$$F_y = G_y|t_y|\,\delta_s. \tag{31}$$

Fig. 4 shows the result of this procedure for the example of the surface gradient where we impose a positive uniform gradient of the surface tension in the y direction.

3.2.2. Surface force in 3D

The implementation of the surface gradient in 3D extends the 2D implementation by considering the second tangential direction as stated in equation (22). Equivalently as in 2D, we first define the column values $\tilde{\sigma}^c$ of the surface tension coefficient σ. This part of the algorithm is identical to the 2D part, with the addition of one more direction. After the column values, $\tilde{\sigma}^c$, are defined, we compute the gradients along the two components orthogonal to the columns: for example, if the columns are constructed in the z direction, see Fig. 5, then the derivatives along the interface are computed in the x and y directions as

$$\begin{aligned}\left(\frac{\partial\sigma}{\partial s_1^z}\right)_{i,j} &= \frac{\tilde{\sigma}^z_{i+1,j} - \tilde{\sigma}^z_{i-1,j}}{2\Delta\sqrt{1+h_x^2}},\\ \left(\frac{\partial\sigma}{\partial s_2^z}\right)_{i,j} &= \frac{\tilde{\sigma}^z_{i,j+1} - \tilde{\sigma}^z_{i,j-1}}{2\Delta\sqrt{1+h_y^2}}.\end{aligned} \tag{32}$$

As previously discussed in 2D, the direction, c, in which the columns are constructed, is chosen based on the interface orientation, where c is the same as the direction of the largest component of the interface normal vector.

Next part of the surface gradient computation is the choice of the tangent vectors, $\hat{\mathbf{t}}_d$, which are computed so that they satisfy $\hat{\mathbf{t}}_d \cdot \hat{\mathbf{n}} = 0$. Among all the possibilities for $\hat{\mathbf{t}}_d$, we choose the two whose projections onto the coordinate plane, defined by all points with c coordinate equal to zero, are parallel to the axes. Fig. 5 illustrates this procedure by an example where the columns are constructed in the z direction and the projections of the tangent vectors $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$ onto the x-y plane are parallel to the x and y axes and denoted by $\hat{\mathbf{t}}_{1,xy}$ and $\hat{\mathbf{t}}_{2,xy}$, respectively. In this particular example, the tangent vectors will be of the form

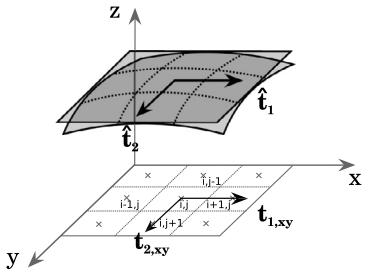

Fig. 5. A stencil used for computing surface gradient in the column containing the cell $\mathcal{C}_{i,j}$, with a tangent plane defined by the vectors $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$. Vectors $\hat{\mathbf{t}}_{1,xy}$ and $\hat{\mathbf{t}}_{2,xy}$ are the projections of $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$ onto the xy-plane respectively.

$$\hat{\mathbf{t}}_1 = (t_{1x}, 0, t_{1z}), \tag{33}$$

$$\hat{\mathbf{t}}_2 = (0, t_{2y}, t_{2z}). \tag{34}$$

The signs of the components of the tangential vectors are chosen so that their projections onto the coordinate plane point in the positive direction of the coordinate axes (see e.g. Fig. 5).

Finally, we compute the surface force, $\mathbf{F}_{st} = (F_x, F_y, F_z)$. In the case such that the columns are constructed in the z direction, the components of $\mathbf{F}_{st}$ are

$$F_x = \frac{\partial \sigma}{\partial s_1^z} t_{1x}, \tag{35}$$

$$F_y = \frac{\partial \sigma}{\partial s_2^z} t_{2y}, \tag{36}$$

$$F_z = \frac{\partial \sigma}{\partial s_1^z} t_{1z} + \frac{\partial \sigma}{\partial s_2^z} t_{2z}. \tag{37}$$

Similarly as in the 2D case, in order to use the CSF formulation, the components of the tangential force need to be defined in the cells around the interface. This is done equivalently as in 2D, using the neighbor averaging procedure, see Section 3.2.1. However, in 3D, there is one extra step due to one of the components containing an addition of two terms, e.g. as in equation (37). In order to illustrate this, consider the general form of the x component of the tangential force

$$F_x = \begin{cases} (\partial\sigma/\partial s_1^x) t_{1x} + (\partial\sigma/\partial s_2^x) t_{2x} & \text{if } c = x, \\ (\partial\sigma/\partial s_1^y) t_{1x} & \text{if } c = y, \\ (\partial\sigma/\partial s_1^z) t_{1x} & \text{if } c = z. \end{cases} \tag{38}$$

Similarly as in 2D, the differences in the sign in the derivatives, $\partial\sigma/\partial s_d^c$, may arise from the choice of the column directions. We proceed by defining the intermediate value of the surface force, $\mathbf{G}$. The components of $\mathbf{G}$ are computed equivalently as in 2D, except for the c component which is defined as

$$G_c = \frac{(\partial\sigma/\partial s_1^c) t_{1c} + (\partial\sigma/\partial s_2^c) t_{2c}}{\sqrt{t_{1c}^2 + t_{2c}^2}}, \tag{39}$$

where c is the direction of the columns. Now we can carry out the averaging procedure for each component of $\mathbf{G}$. Finally, the component of the force in the direction c is

$$F_c = G_c |t_{1c}| \delta_s + G_c |t_{2c}| \delta_s. \tag{40}$$

The other components are computed equivalently as in the 2D case.

4. Results

4.1. The surface gradient computation

We first present the validation of our methodology for computing the surface gradient in 2D geometry where we can compute the gradient exactly. The simplest geometry that we consider is a flat perturbed interface, i.e. let the interface be a function of x as

$$h(x) = h_0 + \varepsilon \cos(2\pi x). \tag{41}$$

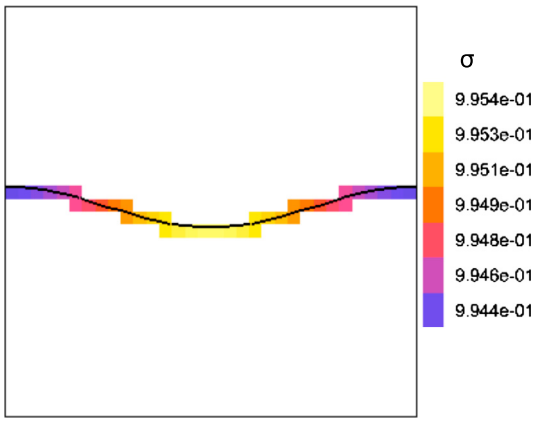

Fig. 6. The setup of the perturbed interface with a surface tension coefficient dependent on the interface profile $h(x) = h_0 + \varepsilon \cos(2\pi x)$. The color represents the surface tension coefficient at the interface, with dark red and dark blue being the maximum and minimum values respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Let the surface tension coefficient be a function of the interface position as

$$\sigma(h) = 1 + \sigma_h h(x), \tag{42}$$

Fig. 6 shows the interface profile and surface tension coefficient at the interface, for $h_0 = 0.5$, $\varepsilon = 0.05$, and $\sigma_h = -10^{-2}$ in a computational domain of 1×1, with the symmetry boundary conditions imposed on all sides. In this case, apart from using the definition of the surface gradient given in equation (21), we can also compute the exact surface gradient using the chain rule as

$$\nabla_s \sigma = \frac{\sigma_h h_x}{\sqrt{1 + h_x^2}} \hat{\mathbf{t}}, \tag{43}$$

where the unit tangent vector, $\hat{\mathbf{t}}$, is defined to point in the positive x direction as

$$\hat{\mathbf{t}} = (1, -h_x)/\sqrt{1 + h_x^2}. \tag{44}$$

Note that the numerator in equation (43), $\sigma_h h_x$, is equivalent to the numerator of equation (26); hence, we can compare their computed values to the exact ones directly. We present the errors associated with computing $\sigma_h h_x$ and h_x, separately, as well as each component of the surface gradient in equation (43). We test the convergence as a function of the mesh size, Δ, using L_1 and L_∞ norms to define E_1 and E_∞ errors respectively as

$$E_1(f) = \frac{\sum_j^N |f_{\text{approx}} - f_{\text{exact}}|}{N}, \tag{45}$$

$$E_\infty(f) = \max |f_{\text{approx}} - f_{\text{exact}}|, \tag{46}$$

where the summation is over all interfacial cells and N is the number of interfacial cells. The interface position in each cell can influence the errors obtained in constructing the columns for the computation of both surface gradients, $\partial\sigma/\partial s^y$, and the derivative of the height function, h_x. To avoid this error bias, we average the errors from 100 simulations where h_0 was modified to $\tilde{h_0} = h_0 + h_r$, where h_r is a random number in the interval $[0, \Delta]$ with uniform distribution.

We initialize the surface tension coefficient, σ, directly as a function of x, i.e. $\sigma(h) = 1 + \sigma_h(\tilde{h_0} + \varepsilon \cos(2\pi x))$. Fig. 7 shows the convergence of the computed $\sigma_h h_x$ as a function of the mesh refinement. As shown, the order of convergence is 2 for both E_1 and E_∞ errors. In this test case, the interfacial value of the surface tension coefficient $\tilde{\sigma}^y$ is computed in the y direction for all cells. Fig. 7 also shows the order of convergence of h_x, computed using height functions (see A.1), that is also 2 for E_1 and E_∞ errors. We note that in order to eliminate errors in initializing the volume fractions, we prescribe the initial shape with a more refined mesh.

Next we investigate the accuracy of the computed surface gradient

$$\nabla_s \sigma(\mathbf{x}) = (\nabla_s \sigma)_x \hat{i} + (\nabla_s \sigma)_y \hat{j},$$

where $\left((\nabla_s \sigma)_x, (\nabla_s \sigma)_y\right) = \left(G_x|t_x|, G_y|t_y|\right)$. Fig. 8 compares the x and y components of the surface gradient with the exact solution. As shown, E_1 and E_∞ errors for both x and y components converge with second order accuracy.

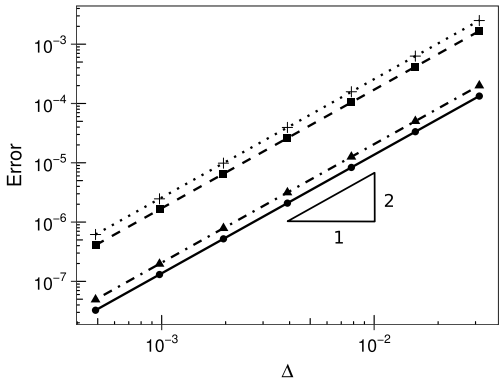

Fig. 7. The computed errors for $E_1(\sigma_h h_x)$ (•), $E_\infty(\sigma_h h_x)$ (▲), $E_1(h_x)$ (■), and $E_\infty(h_x)$ (+). The order of convergence is 2 for $\sigma_h h_x$ for both E_1 (—) and E_∞ (- · -) errors and for h_x for both E_1 (- -) and E_∞ (· · ·) errors. The symbols represent the errors from the computations and the lines show the linear fits.

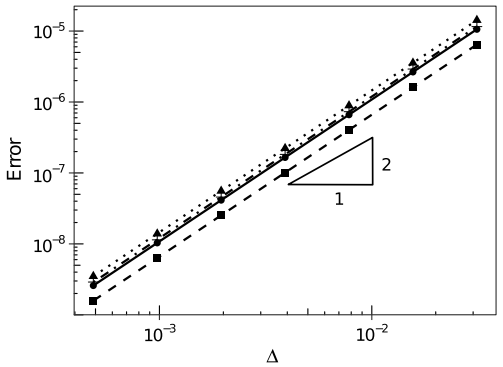

Fig. 8. The computed errors for $E_1((\nabla_s\sigma)_x)$ (•), $E_\infty((\nabla_s\sigma)_x)$ (▲), $E_1((\nabla_s\sigma)_y)$ (■), and $E_\infty((\nabla_s\sigma)_y)$ (+). The order of convergence is 2 for $(\nabla_s\sigma)_x$ for both E_1 (—) and E_∞ (· · ·) errors and for $(\nabla_s\sigma)_y$ for both E_1 (- -) and E_∞ (- · -) errors. The symbols represent the errors from the computations and the lines show the linear fits.

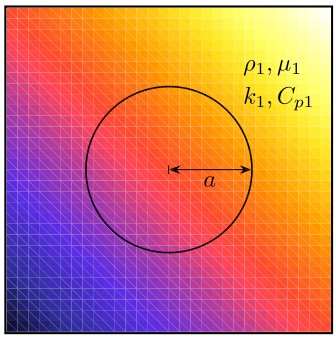

Fig. 9. The initial setup of a circular drop with an imposed uniform temperature gradient. The color shows the temperature with dark blue and dark red being the minimum and maximum values respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Next we test the convergence for a more general interfacial geometry where the interfacial values of $\tilde{\sigma}^c$ are computed using columns in both x and y directions (see Section 3). We consider a circle of radius $a = 0.25$ positioned at $(0.5, 0.5)$ in a 1×1 domain with an imposed temperature distribution

$$T(x, y) = \Delta T\,(x + y)\,, \tag{47}$$

where ΔT is a constant. We assume that the thermal diffusivity is equal for the fluid inside and outside of the circle, i.e. $k_1 = k_2$, $C_{p1} = C_{p2}$, $\rho_1 = \rho_2$, $\mu_1 = \mu_2$, where the subscripts 1 and 2 denote surrounding and the fluid inside of the drop, respectively. Fig. 9 shows the setup with color representing the temperature field. Here we choose $\Delta T = 0.1$, $k_1 = 1$, $C_{p1} = 1$ and $\rho_1 = 1$.

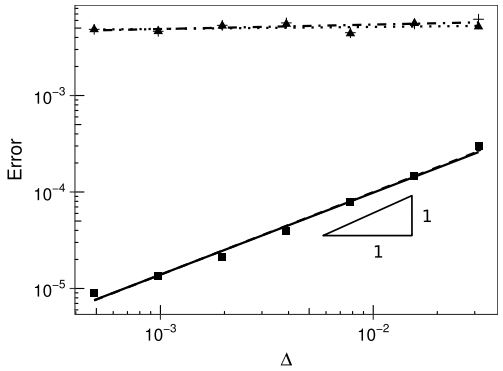

Fig. 10. The computed error for $E_1((\nabla_s\sigma)_x)$ (•), $E_\infty((\nabla_s\sigma)_x)$ (▲), $E_1((\nabla_s\sigma)_y)$ (■), and $E_\infty((\nabla_s\sigma)_y)$ (+). The order of convergence for $(\nabla_s\sigma)_x$ is 0.85 and 0.021 E_1 (—) and E_∞ (· · ·) errors respectively, and the order of convergence for $(\nabla_s\sigma)_y$ is 0.856 and 0.046 for E_1 (- -) and E_∞ (- · -) errors, respectively. The symbols represent the errors from the computations, and the lines show the linear fit of those points.

For simplicity, we assume that the surface tension coefficient is a linear function of temperature, i.e. $\sigma(T) = 1 + \sigma_T T$, where we let $\sigma_T = -0.1$. We set the velocity to zero, and knowing that the interface is exactly circular, we can compute exact surface gradient from the definition

$$\nabla_s\sigma = \nabla\sigma - \hat{\mathbf{n}}\left(\hat{\mathbf{n}}\cdot\nabla\sigma\right), \tag{48}$$

$$= \frac{\sigma_T\Delta T}{a^2}\left\langle a^2 - x^2 \mp x\sqrt{a^2-x^2}, x^2 \mp x\sqrt{a^2-x^2}\right\rangle, \tag{49}$$

$$= \frac{\sigma_T\Delta T}{a^2}\left\langle y^2 \mp y\sqrt{a^2-y^2}, a^2 - y^2 \mp y\sqrt{a^2-y^2}\right\rangle. \tag{50}$$

Equations (49) and (50) give the surface gradient as a function of x and y, respectively.

We initialize the temperature following two approaches, and discuss their performance. The first approach is to define the interface as a function of x and y depending on the more favorable interface orientation as follows

$$T(x,y) = \begin{cases} \Delta T\left(x \pm \sqrt{a^2-x^2}\right) & \text{if } |x| < |y|, \\ \Delta T\left(y \pm \sqrt{a^2-y^2}\right) & \text{otherwise,} \end{cases} \tag{51}$$

where x and y are coordinates of the cell centers. Second approach is to use positions of the centroid of the interface contained in each cell to initialize the temperature by equation (47). We show below that the second approach leads to more accurate results.

We compare the computed surface gradient with the exact solution by considering E_1 and E_∞ errors defined in equations (45) and (46), respectively. Similarly as in the previous example, in order to eliminate the dependence of the errors on the interface position in the cell, the center of the drop is positioned randomly in the interval $[0,\Delta]\times[0,\Delta]$, and the errors are averaged over 100 random realizations. Fig. 10 shows the convergence to the exact solution for the x and y components of the gradient. The convergence of E_1 error is 0.85 and 0.86 for the x and y components, respectively. The slow convergence of E_∞ error is due to the errors in initializing the temperature at the lines $|x| = |y|$ from equation (51), demonstrated later.

In order to reduce the influence of the initialization of T on the convergence, we also compute the convergence of L_1 norm of each component of the surface gradient

$$L_1(f) = \frac{1}{N}\sum_i^N |f_i|, \tag{52}$$

where f_i is the x or y component of the surface gradient, and N is the number of interfacial points. Fig. 11 shows L_1 norm for the x and y components of the surface gradient and the order of convergence of the temperature $T(x,y)$ in the interfacial points as a function of the mesh size, Δ. We find the order of convergence of the x and y components of the surface gradient to be 1.2 and 1.4. The order of convergence of $T(x,y)$ along the interface is 1.2. This indicates that the order of convergence of the surface gradient is limited by the order of convergence of the initial temperature at the interface.

Fig. 12(a) shows the distribution of errors at the circular interface for one random realization. The largest errors appear around the lines $|x| = |y|$. Based on this we conclude that the lack of convergence of E_∞ error is caused by the initialization of the temperature which changes the dependence on x or y variable at the lines $|x| = |y|$.

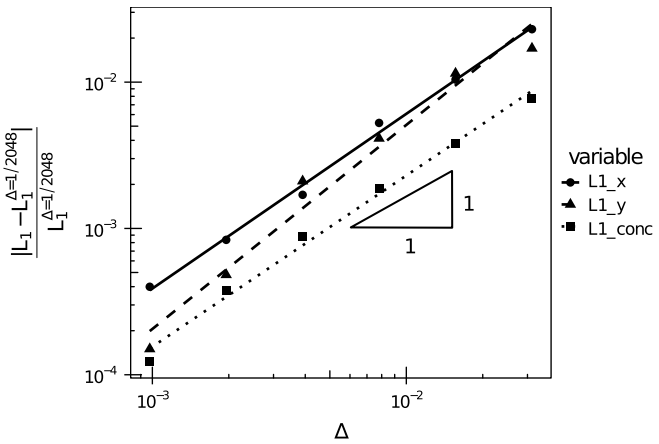

Fig. 11. The computed errors for $L_1((\nabla_s\sigma)_x)$ (•), $L_1((\nabla_s\sigma)_y)$ (▲), and $L_1(T)$ (■). The order of convergence for $(\nabla_s\sigma)_x$ and $(\nabla_s\sigma)_y$ is 1.2 (—) and 1.4 (- -), respectively. The order of convergence of $T(x, y)$ at the interface is 1.2 (· · ·). The symbols represent the errors from the computations and the lines show the linear fit of those points.

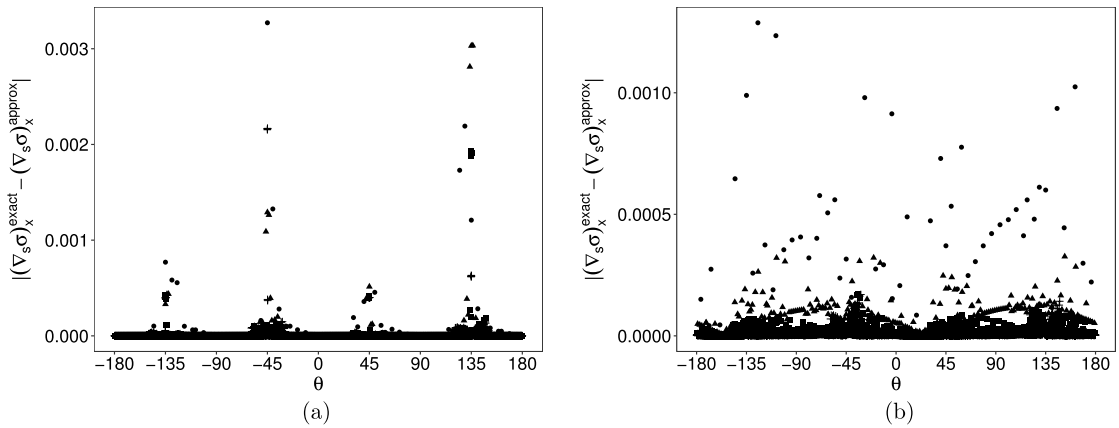

Fig. 12. Errors of the x component of the surface gradient at the interfacial cells for $\Delta = a/8$ (•), $a/32$ (▲), $a/128$ (■), and $a/512$ (+). Initializing the temperature using (a) equation (51) and (b) equation (53). θ is defined to be zero at the positive x axis and increasing counterclockwise.

In order to initialize the temperature more accurately at the interface we use the centroid of the interface segment contained in each cell, (x_c, y_c). Then the initial temperature is given by

$$T(x, y) = \Delta T\,(x_c + y_c)\,. \tag{53}$$

This reduces the errors from initializing the temperature at the lines $|x| = |y|$ compared to using equation (51). Here, we also explore a different way of approximating interfacial temperature, and use surface area weighted average instead of volume fraction weighted average (see Section 3). The volume weighted average gives the temperature at the center of the mass of the fluid phase in the column, whereas the surface area weighted average gives the temperature at the center of the interface in the column. Hence, the latter is consistent with the initialization of the temperature using equation (53).

Fig. 12(b) shows the errors of the x component of the surface gradient at the interfacial cells if the temperature is initialized using equation (53). The errors are still the largest around $|x| = |y|$, however those are the usual "weak" spots of the height function construction. Fig. 13 shows the improvement in the convergence to the exact solution using L_1 and L_∞ norm for the x and y components of the surface gradient as a function of mesh refinement. The order of convergence for L_1 norm is 0.94 and 0.89 for the x and y components of the surface gradient, respectively, and the order of convergence for L_∞ norm is 0.63 and 0.57 for the x and y components of the surface gradient, respectively. Hence, the second approach of initializing the temperature (using equation (53)) improves the convergence of the L_∞ norm significantly.

4.2. Drop migration

We further test our numerical implementation using a classical problem of the thermocapillary drop migration (see the reviews [4,31]). A drop or a bubble placed in a fluid with an imposed temperature gradient moves due to the variation in the surface tension coefficient as a function of temperature.

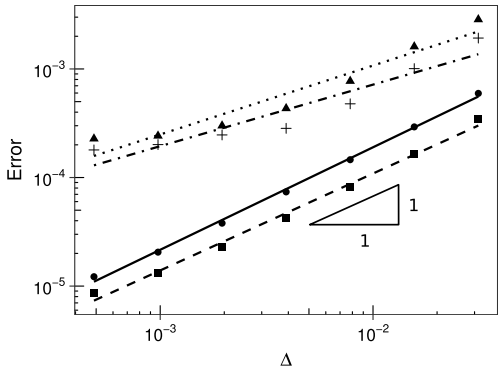

Fig. 13. The computed errors for $E_1((\nabla_s\sigma)_x)$ (•), $E_\infty((\nabla_s\sigma)_x)$ (▲), $E_1((\nabla_s\sigma)_y)$ (■), and $E_\infty((\nabla_s\sigma)_y)$ (+). The order of convergence for $(\nabla_s\sigma)_x$ is 0.94 (—) and 0.63 (- -) for E_1 and E_∞ errors, respectively, and the order of convergence for $(\nabla_s\sigma)_y$ is 0.89 (···) and 0.57 (- · -), respectively. The symbols represent the errors from the computations, and the lines show the linear fit of those points.

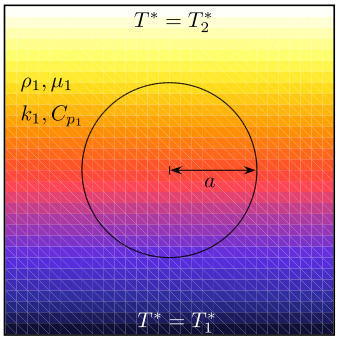

Fig. 14. The initial setup of the drop migration problem. The color represents the linear temperature distribution with imposed temperatures T_1^* and T_2^* at the horizontal boundaries. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Several authors use this problem for benchmarking their numerical algorithms for a temperature dependent surface tension coefficient [19,32,33]. We show the comparison of our numerical results with the available work in the literature. We also show the comparison with the analytical solution of the drop terminal velocity by Young et al. [34]. Young et al. [34] show that the nondimensional velocity of the drop in an unbounded domain for an axisymmetric geometry in the limit of small Ma and Ca numbers can be approximated as

$$v_{ygb}^* = \frac{\mu_1}{\sigma_T a \Delta T} \frac{2}{(2+k_r)(2+3\mu_r)}, \tag{54}$$

where μ_1 is the viscosity of the surrounding fluid, σ_T is the (constant) gradient of the surface tension coefficient with respect to the temperature, a is the drop/bubble radius, ΔT is the imposed temperature gradient, and k_r and μ_r are the thermal conductivity and viscosity ratios, respectively, for the drop/bubble compared to the surrounding fluid.

In what follows, we choose a combination of Re, Ma, and Ca numbers to compare our results with those in [19,32,33]. Also, we consider small values of Re, Ca, and Ma when comparing with the theoretical prediction above (equation (54)), which is valid in the limit of Re, Ma $\to 0$.

Fig. 14 shows the considered setup: a drop or a bubble of radius a is placed in an ambient fluid, with a linear temperature gradient imposed in the y direction. The temperature at the top and the bottom boundaries is set to constant values and a zero heat flux boundary condition is imposed at the left and right boundaries. The boundary conditions for the flow are no-slip and no penetration at the top and bottom boundaries and symmetry at the left and right boundaries.

We solve equations (10) and (15) and consider the following scales

$$p_0 = \frac{\mu_1 U_0}{a}, \quad t_r = \frac{a}{U_0}, \quad U_0 = \frac{\sigma_T a \Delta T}{\mu_1} \quad T_0 = a\Delta T,$$

where the subscript 1 denotes the properties of the ambient fluid. The surface tension at the interface between the drop and the ambient fluid is assumed to depend linearly on the temperature as given by equation (20), which, rescaled using the scales above, yields

$$\sigma^* = 1 + \mathrm{Ca}\left(T^* - T_R^*\right). \tag{55}$$

Next we present the comparison of our results with the available studies in the literature.

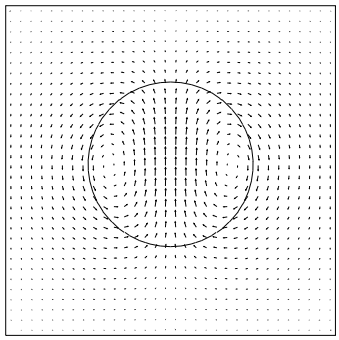

Fig. 15. The velocity field in the drop and the surrounding fluid.

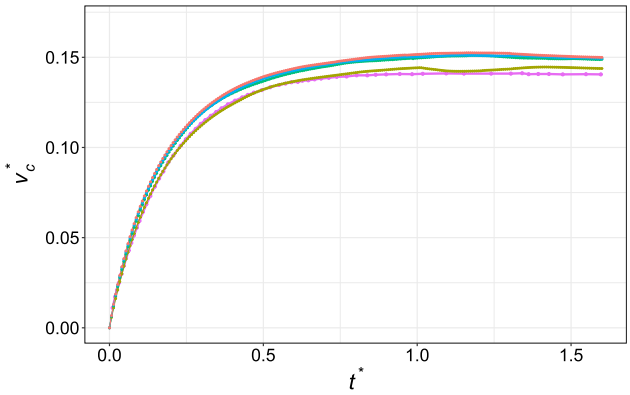

Fig. 16. The drop migration velocity for $\mathrm{Re} = \mathrm{Ma} = 0.72$ and $\mathrm{Ca} = 0.0576$ for $\Delta = 1/32$ (–), 1/64 (–), 1/128 (–), and 1/256 (–) compared with the result given in [19] (–) – with the resolution corresponding to $\Delta = 1/32$ in our results – for 2D simulations. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

We start by comparing our results with the ones by Ma and Bothe [19]. The material properties are $\rho_1 = 500 \frac{\mathrm{kg}}{\mathrm{m}^3}$, $\mu_1 = 0.024$ Pa·s, $\sigma_0 = 10^{-2}$ $\frac{\mathrm{N}}{\mathrm{m}}$, $\sigma_T = 2 \times 10^{-3}$ $\frac{\mathrm{N}}{\mathrm{m\,K}}$, $k_1 = 2.4 \times 10^{-6}$ $\frac{\mathrm{W}}{\mathrm{m\,K}}$, $C_{p_1} = 10^{-4}$ $\frac{\mathrm{J}}{\mathrm{kg\,K}}$, $\Delta T = 200$ $\frac{\mathrm{K}}{\mathrm{m}}$, $T_2 = 290$ K, $a = 1.44 \times 10^{-3}$ m. The ratio of the material properties between the ambient fluid and the drop is 2. These physical properties give the nondimensional parameters $\mathrm{Re} = \mathrm{Ma} = 0.72$, $\mathrm{Ca} = 0.0576$, and the velocity scale $U_0 = 0.024$ $\frac{\mathrm{m}}{\mathrm{s}}$. Fig. 15 shows the computed velocity field in the drop and the surrounding fluid. The gradient of the surface tension coefficient drives the flow from low surface tension coefficient region (top) to high surface tension coefficient region (bottom). This creates the flow inside the drop and as a result the drop moves in the positive y direction. The drop velocity is computed using the following definition of the centroid velocity

$$v_c^* = \frac{\sum_{i,j} v_{i,j}^* \chi_{i,j} (\Delta_{i,j}^*)^2}{\sum_{i,j} \chi_{ij} (\Delta_{i,j}^*)^2}$$

where $v_{i,j}^*$ is the y component of the cell-center velocity.

Fig. 16 shows the computed velocity of the drop compared to the results in Ma and Bothe [19]. In this test case, the computational domain is a square box with a side length equal to four times the drop radius; the drop is initially placed at the center of the domain. We choose this domain size to replicate as closely as possible the simulations in [19]. As shown, a good agreement is obtained with the result in [19], corresponding to $\Delta = 1/32$ in our results. We also note some decay in the migration velocity for late times. This decay also exists in the simulation of Ma and Bothe [19], although it is less pronounced. Additional simulations (not shown for brevity) confirm that this slight decay is the domain size effect: as the drop approaches the top boundary, the migration velocity decreases. For simulations carried out in larger domains, a decay is not present. On a different note, we point out that for a mesh resolution below $\Delta = 1/32$, i.e. 8 grid points per drop radius, a consistent height-function approximation is no longer possible and we cannot expect accurate solutions at such low resolutions.

Next, we carry out another comparison for a smaller value of Re and Ca numbers and when $\mathrm{Ma} = 0$; we choose $\mathrm{Re} = \mathrm{Ca} = 0.066$ in accordance with the results presented in [33] for the VOF method. The computational box is a rectangle of size $10a \times 15a$. The grid size is $\Delta = 5a/512$. The density of the ambient fluid is set to $\rho_1^* = 0.2$, and the viscosity is $\mu_1^* = 0.1$.

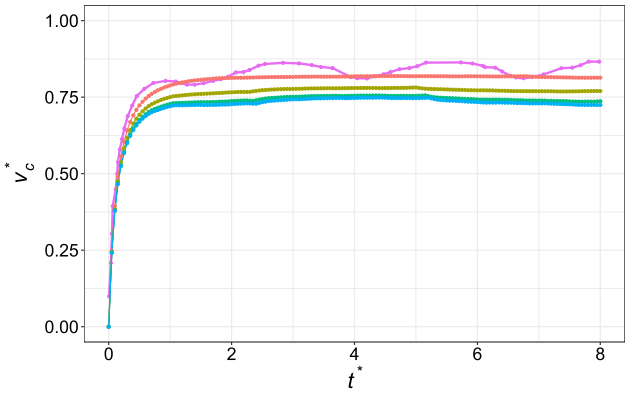

Fig. 17. The convergence of the migration velocity as a function of the time step for $\Delta t = 10^{-4}$ (-), 5×10^{-6} (-), 10^{-5} (-), and 5×10^{-6} (-) compared with the results in [33] (-) for 2D simulation, for $\mathrm{Re} = \mathrm{Ca} = 0.066$ and $\mathrm{Ma} = 0$. The velocity is rescaled by v^*_{ygb}. The grid size is $\Delta = 5a/512$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

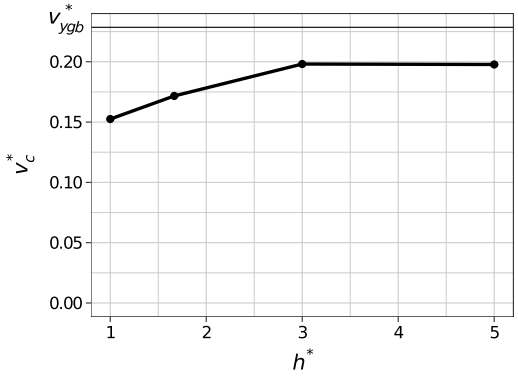

Fig. 18. The convergence of the terminal velocity with an increased distance from the wall, h^*, for $\mathrm{Re} = \mathrm{Ma} = 2.5 \times 10^{-3}$ and $\mathrm{Ca} = 1.25 \times 10^{-3}$ for 2D simulations. The grid size is $\Delta = a/32$, except for $h^* = 1.666$ case where $\Delta = a/32$.

The ratio of the physical properties of the drop to the ambient fluid is set to 1. The surface tension coefficient is $\sigma_0^* = 0.1$ at the reference temperature $T_R^* = 0$, with $\sigma_T^* = -0.1$. The temperature gradient is set to $\Delta T^* = 0.1\bar{3}$, which is fixed by setting $T_1^* = 0$ and $T_2^* = 1$. The drop is initially centered horizontally at a distance $3a$ from the bottom wall. Fig. 17 shows the comparison of our method with the results in [33], along with the temporal convergence of our method. Compared to the results by Herrmann et al. [33], our results do not exhibit oscillations, which agrees with the asymptotic solution of constant rise velocity. Another difference is that our terminal velocity converges to a smaller value with decreasing time step. However, the timestep used in the results of Herrmann et al. [33] is not specified in their paper.

We also test the convergence to the analytical solution obtained in the limit of Ma and Re approaching zero and in the unbounded domain, where the terminal velocity approaches v^*_{ygb} value given in equation (54). Fig. 18 shows the terminal velocity of a droplet for a 2D simulation as a function of a distance from the wall for $\mathrm{Re} = \mathrm{Ma} = 2.5 \times 10^{-3}$ and $\mathrm{Ca} = 1.25 \times 10^{-3}$. The grid size here is $\Delta = a/32$, except for $h^* = 1.666$ case where $\Delta = a/32$. Note that the Ca number is chosen small enough to prevent droplet oscillations, and it does not affect the migration velocity. The terminal velocity converges to a value lower than v^*_{ygb} due to the fact that we simulate the planar 2D geometry with a finite size computational domain, while the theoretical terminal velocity is calculated for an axisymmetric spherical drop under the assumption of infinite domain size and no drop deformation. We next show that our 3D result in fact converges to this analytical solution.

We perform similar tests for the 3D simulations. Fig. 19 shows the migration velocity for $\mathrm{Re} = \mathrm{Ma} = 0.72$ and $\mathrm{Ca} = 0.0576$. The parameters and the domain size are equivalent to the simulation results shown in Fig. 16. We note again that for a mesh resolution below $\Delta = 1/64$, i.e. 16 grid points per drop radius, a consistent height-function approximation is no longer possible, and we cannot expect accurate solutions at such low resolutions; such a result for $\Delta = 1/64$ (-) is shown in Fig. 19. (A consistent construction of height functions fails at a higher resolution in 3D than in 2D, see e.g. [23] and references therein.) Also as in 2D, we note a decay in the migration velocity at later times (domain size effect). The results also show that the oscillations in the computed velocity decay with mesh refinement and the terminal velocity converges to a higher value compared to the 2D case. However, this value is still smaller than v^*_{ygb} due to the small domain size and relatively large Re and Ma numbers. Fig. 20 shows the terminal velocity of a droplet for a 3D simulation as a function of a distance from the wall for $\mathrm{Re} = \mathrm{Ma} = 2.5 \times 10^{-3}$ and $\mathrm{Ca} = 10^{-3}$. As shown, the terminal velocity approaches the analytical

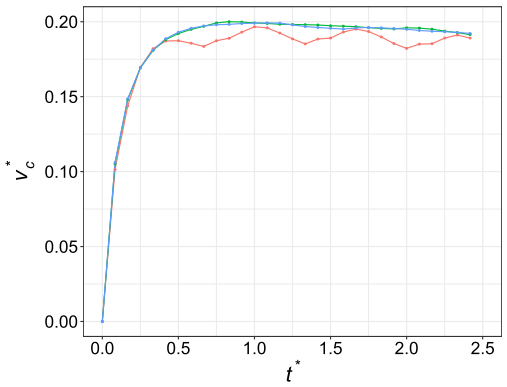

Fig. 19. The convergence of the migration velocity in a 3D simulation with mesh refinement for $\Delta = 1/32$ (-), $1/64$ (-), and $1/128$ (-); $\mathrm{Re} = \mathrm{Ma} = 0.72$ and $\mathrm{Ca} = 0.0576$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

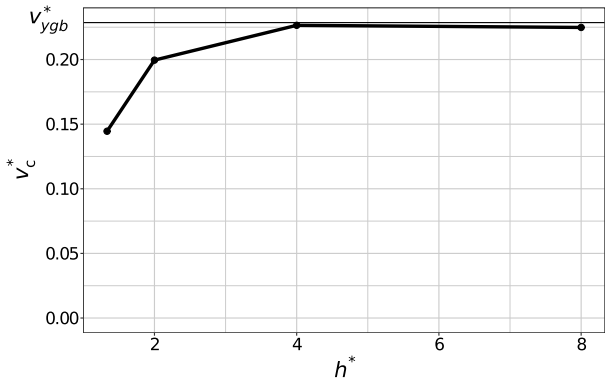

Fig. 20. The convergence of the terminal velocity with increased distance from the wall, h^*, for $\mathrm{Re} = \mathrm{Ma} = 2.5 \times 10^{-3}$ and $\mathrm{Ca} = 10^{-3}$ for 3D simulations. The grid size is $\Delta = a/16$, except for $h^* = 1.666$ case where $\Delta = a/24$.

value v_{ygb}^*. Recall that v_{ygb}^* is calculated for an axisymmetric spherical drop under the assumption of infinite domain size and no drop deformation. The grid size here is $\Delta = a/16$, except for $h^* = 1.666$ case where $\Delta = a/24$.

In this section we have shown the comparison of our method with existing literature and with a limiting analytical solution. Our method shows the convergence to the analytical value of the terminal velocity. The trend of the obtained solutions under grid refinement, as well as the time needed to reach the terminal velocity are consistent with the previously reported results.

4.3. Coalescence and non-coalescence of sessile drops

Next we demonstrate the performance of our numerical methods through an example of the coalescence behavior of sessile drops with different surface tension coefficient. We model the experiments of the coalescence of two droplets with different alcohol concentrations by Karpitschka et al. [35–37]. In their experimental study, they show three coalescence regimes depending on the difference in the surface tension coefficient between the two droplets: immediate coalescence, delayed coalescence, and non-coalescence. They identify a key parameter that governs the transition between the delayed and non-coalescence regimes: the specific Marangoni number $\mathrm{M} = 3\Delta\sigma/(2\bar{\sigma}\theta^2)$ [37], where $\Delta\sigma$ is the difference in the surface tension coefficient between the two drops and $\bar{\sigma}$ is the average of the surface tension coefficient of two drops. They determine a threshold Marangoni number $\mathrm{M}_t \approx 2 \pm 0.2$ experimentally for the transition between the delayed coalescence and non-coalescence regimes.

Here we show that our numerical simulations also reveal the three regimes in agreement with the experimental observations in [35,36]. From the numerical simulation point of view, this problem involves a level of difficulty: unlike temperature, the concentration should remain strictly confined to the liquid phase and should not leak out to the ambient phase. To avoid this difficulty, we combine our variable surface tension coefficient methodology with the numerical technique already implemented in the original version of GERRIS [28] which prevents the concentration from leaking out of the liquid domain into the ambient phase [38].

We model the 2D problem since the dominant flow dynamics in the problem is in the region connecting the two droplets, where the gradient of the surface tension coefficient is the strongest, and in this region we can ignore the out of plane curvatures. Initially, the drops have the shape of a circular segment with the base radius R_b and a contact angle θ,

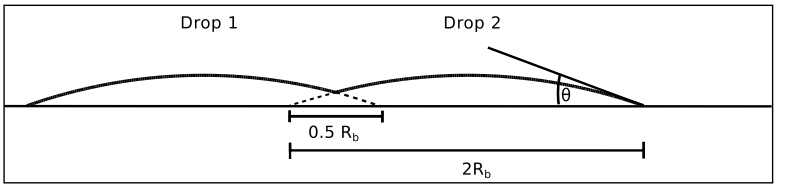

Fig. 21. A schematic of the drop coalescence problem.

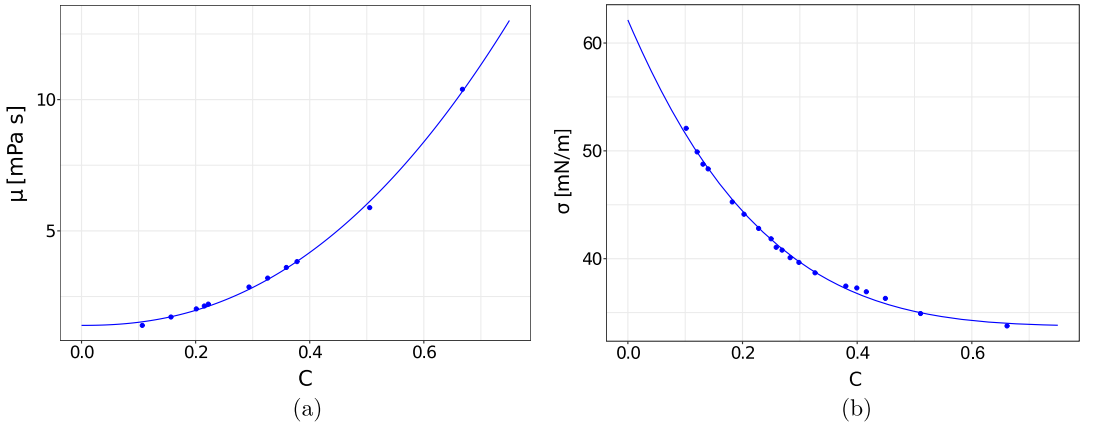

Fig. 22. Viscosity (a) and surface tension coefficient (b) of the mixture of 1,2-Butanediol and water as a function of alcohol concentration. Points represent the data from Karpitschka and Riegler [35], and the lines show the fit of the points.

and are connected by an overlap of $0.25R_b$ (see Fig. 21). The drops have an equal base radius R_b and we assume that their densities are equal. The viscosity depends on the alcohol concentration C, where we use a nonlinear fit to the data given in [35] of the form

$$\mu\,(C) = \mu_1 + a_\mu\,(\mu_2 - \mu_1)\,(1 - C)^{n_\mu}\,, \tag{56}$$

shown in Fig. 22(a). The drops are composed of the mixture of the 1,2-Butanediol and water, but they differ in the concentrations of alcohol. Fig. 22(b) shows the dependence of the surface tension coefficient on the concentration of 1,2-Butanediol in water. Similarly as for the viscosity, we fit this data to a function of the form

$$\sigma\,(C) = \sigma_1 + a_\sigma\,(\sigma_2 - \sigma_1)\,(1 - C)^{n_\sigma}\,. \tag{57}$$

The parameters a and n are determined from the fit.

We first show a simulation of two drops, with an equal surface tension coefficients. We consider the case where the concentration of alcohol is 45%, and the base radii of the circular segments are both $R_b = 3$ mm. Along with a no-slip boundary condition at the substrate, we also impose a $\theta = 15^\circ$ contact angle. For the contact angle implementation in GERRIS and related numerical discussion the reader is referred to [29,39]. Fig. 23 shows the evolution of the interface at different times. The droplets coalesce immediately, fully merge after 0.1 s, and assume an equilibrium shape of one large circular segment at a later time. The color represents the concentration of alcohol, which is contained inside of the fluid and zero in the surrounding.

Next we examine the case where $\mathrm{M} \approx 1.2 < \mathrm{M}_t$. Fig. 24 shows the simulations of this intermediate regime where droplets coalescence is delayed. Here, we set drop 1 to 45% and drop 2 to 35% of alcohol. The connected drops move toward higher surface tension coefficient due to the Marangoni induced flow until the concentrations are mixed, resulting in a smaller gradient in the surface tension coefficient. Fig. 25 shows the closeup images of the neck region between the two drops corresponding to the three panels in the middle shown in Fig. 24. In this figure, we show the flow mixing dynamics which leads to the decrease of the difference in the surface tension coefficient in the neck region, resulting in a consequent full coalescence of the two drops.

Next we consider a case in the non-coalescence regime. We set the drop 1 to 45% and the drop 2 to 33% of alcohol. Fig. 26 shows the simulation results for $\mathrm{M} \approx 1.8 \approx \mathrm{M}_t$. In this case, the Marangoni induced flow initially pushes the fluid from the drop 1 towards the drop 2. However, this results in the thinning of the connecting neck between the drops (at $t = 1$ s), and the fluid cannot pass from drop 1 to drop 2 anymore. Fig. 27 shows the closeup images of the neck region between the two drops corresponding to the middle three panels shown in Fig. 26. Compared to the previous case where the droplets coalescence is delayed ($\mathrm{M} \approx 1.2$), the behavior of the mixing of the fluids in the neck region is prevented by the thinning of the neck. Hence these droplets do not coalesce, but instead they move together with a constant velocity u_d on the substrate in the direction of the higher surface gradient.

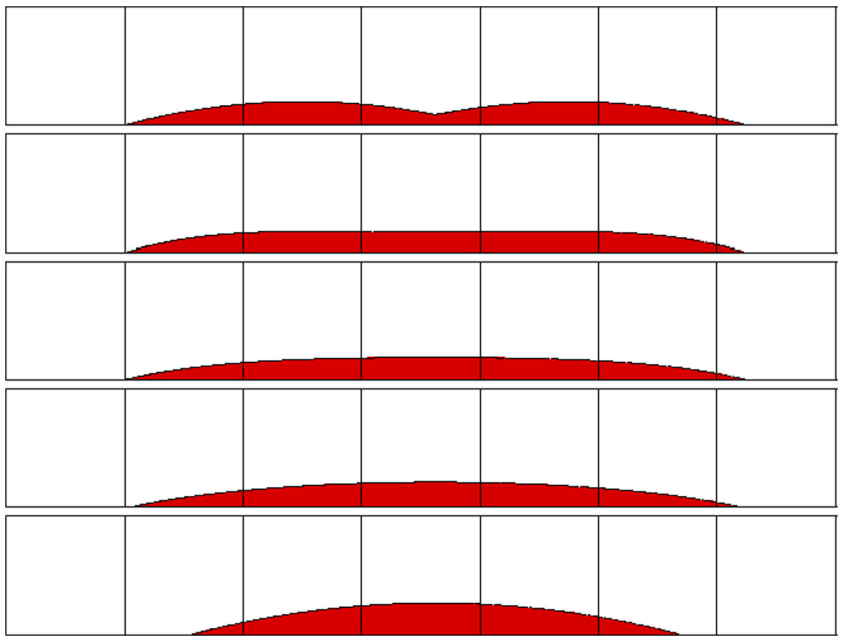

Fig. 23. The evolution of two drops with an equal alcohol concentration, i.e. no difference in the surface tension coefficient, at times $t = 0$ s, $t = 0.02$ s, $t = 0.04$ s, $t = 0.1$ s and $t = 1$ s from the top to the bottom. The color shows the concentration of alcohol. Each box is equivalent to 2 mm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

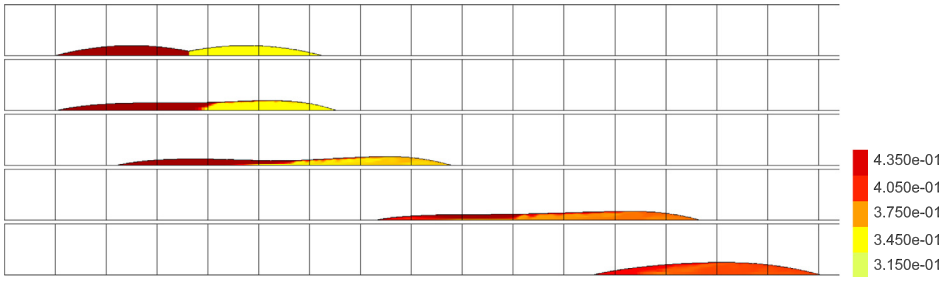

Fig. 24. The evolution of two drops with small difference in the surface tension coefficient at times $t = 0$ s, $t = 0.1$ s, $t = 1$ s, $t = 5$ s, and $t = 10$ s from top to bottom. The color shows the concentration of alcohol. Each box is equivalent to 2 mm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

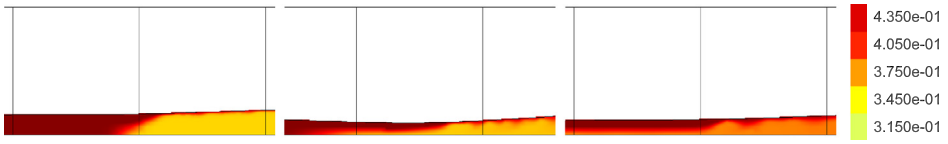

Fig. 25. The closeup of the neck region between the two drops shown in Fig. 24, at times $t = 0.1$ s, $t = 1$ s, and $t = 5$ s from left to right. The color shows the concentration of alcohol. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

This quasi-steady behavior is also observed in the experiments by Karpitschka and Riegler [36]. Fig. 28(a) shows the velocity of the points at the interface after the quasi-steady state is reached as a function of the distance from the bridge region, D_b. The points to the left of the bridge region have a velocity $\approx 2u_d$ (solid line). At the bridge region the interface is close to the solid substrate and the velocity becomes close to zero due to the no-slip boundary condition. In the region close to the bridge in the drop 2, the velocity has a jump and reaches the maximum value due to the Marangoni effect resulting from a high gradient of the surface tension coefficient at the neck region. Away from the bridge, the velocity is again comparable to u_d. This behavior is in qualitative agreement with the experimental observation by Karpitschka and Riegler [36,37]. To provide more insight into the flow through the neck region, in Fig. 28(b) we present the alcohol concentration at the interface as a function of the distance from the bridge region, D_b. As shown, a localized and steady state gradient of the surface tension coefficient is established through the neck region. This Marangoni effect can counteract

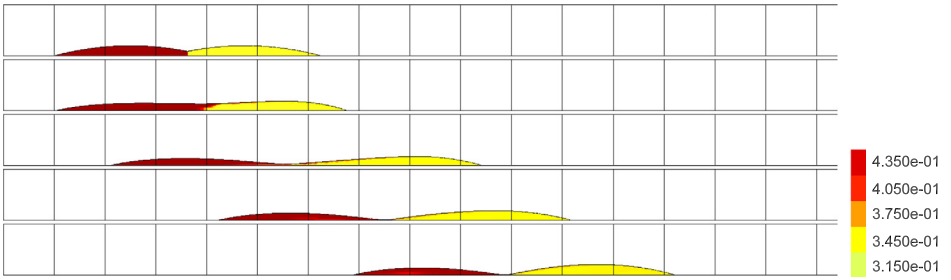

Fig. 26. Non-coalescence of drops at times $t = 0$ s, $t = 0.1$ s, $t = 1$ s, $t = 2$ s, and $t = 6$ s from the top to the bottom. The color shows the concentration of alcohol. Each box is equivalent to 2 mm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

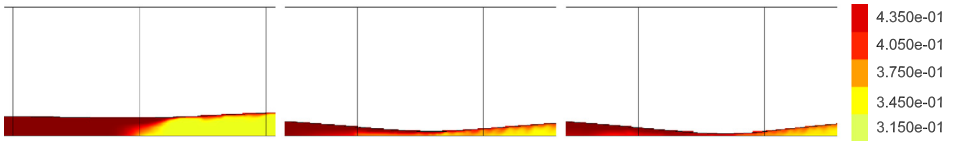

Fig. 27. Closeup of the neck region between the two drops shown in Fig. 26, at times $t = 0.1$ s, $t = 1$ s, and $t = 2$ s from left to right. The color shows the concentration of alcohol. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

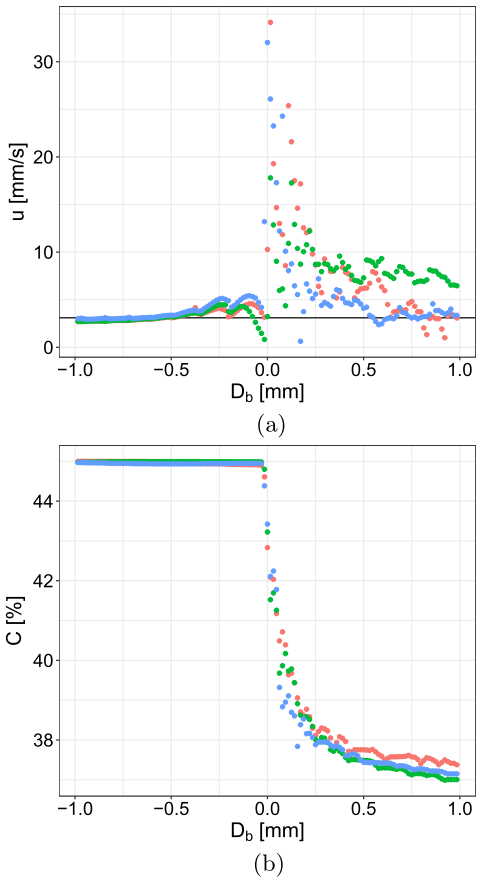

Fig. 28. The x component of the velocity field at the interfacial points (a) and the concentration of alcohol at the interfacial points (b) for the non-coalescent drops for the example shown in Fig. 26. The color shows different times, with red, green and blue being $t = 4$ s, $t = 5$ s, and $t = 6$ s, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the capillary effect that would otherwise result in the coalescence and can therefore sustain the non-coalescence and the movement of drops temporarily.

5. Conclusions

We have developed a new numerical methodology for including the variable surface tension coefficient in a VOF based Navier–Stokes solver. The method handles both temperature or concentration dependent variations in the surface tension coefficient. We employ a height function inspired formulation to compute the surface gradients and the resulting stresses at the interface (Marangoni forces) in a more general numerical framework. We show the robustness and accuracy of our developed method by studying the convergence of the computation of the surface gradient for multiple geometries and the convergence of the terminal velocity for the classical problem of the drop migration with an imposed constant temperature gradient. The drop migration simulation results are in agreement with the available theoretical and numerical results. We also show that our method produces results consistent with experimental data in the case of concentration dependent surface tension coefficient. Our numerical implementation extends to the adaptively refined meshes which improves the computational efficiency for Marangoni induced flows that require a high resolution around the interface.

The presented approach represents a first attempt for implementing a general variable surface tension coefficient in the VOF method. As presented here, our method can subsequently be used directly for the surface tension coefficient dependent on the surfactant concentration. This includes implementing the solution to the surfactant transport equation for soluble and insoluble surfactants. Our methodology can provide the tools for developing a more robust and accurate numerical simulations for two-phase flows with surfactants. The surfactant flows have many applications, e.g. in the chemical industry, pharmaceuticals and technology [40], and their understanding will have a far reaching effects in many areas.

The numerical verifications and validations with available literature demonstrate the efficiency and applicability of our methodology. Our numerical approach is implemented in an adaptive mesh refinement framework, which now makes the detailed numerical simulations that incorporate the effects of the tangential (Marangoni) stresses feasible. This is particularly relevant for a number of flow problems where the Marangoni effect may play a crucial role, such as the evolution of thin films on nanoscale, where Marangoni effects may result either from the concentration gradients (mixture of two fluids) or thermal gradients due to an internal or external sources. Our future research will continue in this direction.

Acknowledgements

This work was partially supported by the NSF grants No. DMS-1320037 (S. A.) and No. CBET-1604351 (L. K., S. A.).

Appendix A

A.1. Height functions

An accurate computation of the interface normals and curvature in the VOF method can be achieved using the height function method [41]. In this method, height functions are defined in each interfacial cell as a sum of the volume fractions in the fluid column constructed of the interfacial cells in the vertical or horizontal orientation. Fig. 29 shows the columns with a vertical orientation used for calculating the height functions implemented in Gerris solver [23]; the solid lines show the stencil size required to calculate the curvature in the cell marked by the bold lines. The height function for the column i is

$$h_i = \sum_k \chi_{i,k}, \tag{A.1}$$

where the index k includes all interfacial cells in the column. The number of cells used for construction of the columns is optimized based on each column, i.e. there is no fixed stencil size, as illustrated in Fig. 29. Fig. 29(a) shows a symmetric stencil where only three cells are required for calculating the height function in each column. Fig. 29(b) shows an asymmetric stencil where each column requires including a different number of cells.

In 2D, the height function can be calculated in either x or y direction, depending on the orientation of the interface. Here, we will present the discretization for the height function for the columns in the y direction (as in Fig. 29). For the heights collected in the x direction the equations are equivalent with derivatives of the height function with respect to x replaced by the derivatives with respect to y. The curvature of the interface is calculated from the height functions as

$$\kappa = \frac{h_{xx}}{\left(1 + h_x^2\right)^{3/2}}, \tag{A.2}$$

where the derivatives of the height functions in equation (A.2) are calculated using a second order central difference

$$h_x = \frac{h_{i+1} - h_{i-1}}{2\Delta}, \tag{A.3}$$

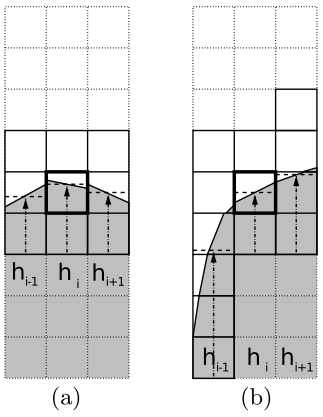

Fig. 29. A height function computation with compact stencil constructed independently for each column. (a) A stencil for a slightly curved interface. (b) An asymmetric stencil, where each column requires different number of cells. From [23].

$$h_{xx} = \frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta^2}, \tag{A.4}$$

where Δ is the cell size. In 3D, e.g. if the columns are computed in z direction, the curvature is

$$\kappa = \frac{h_{xx} + h_{yy} + h_{xx}h_y^2 + h_{yy}h_x^2 - 2h_{xy}h_x h_y}{\left(1 + h_x^2 + h_y^2\right)^{3/2}}. \tag{A.5}$$

A.2. Algorithm

Algorithm 1 shows the pseudocode for approximating the interfacial value of the surface tension coefficient, $\tilde{\sigma}$, in each column.

for *Each component* $c \in \{x, y, z\} \equiv \{0, 1, 2\}$ **do**
  for *All interfacial cells* $\mathcal{C}$ **do**
    Set $V \leftarrow \chi(\mathcal{C})$, $\tilde{\sigma}^c \leftarrow V\sigma(\mathcal{C})$;
    for *Each direction* $d \in \{2c, 2c+1\}$ **do**
      Set $\mathcal{N} \leftarrow$ neighboring cell of $\mathcal{C}$ in direction d;
      while $0 < \chi(\mathcal{N}) < 1$ **do**
        Set $v_d \leftarrow \chi(\mathcal{N})$, $\tilde{\sigma}^c \leftarrow \tilde{\sigma}^c + v_d\sigma(\mathcal{N})$, $V \leftarrow V + v_d$;
        Set $\mathcal{N} \leftarrow$ neighboring cell of $\mathcal{N}$ in direction d;
      end
    end
    if $\{v_{2c}, v_{2c+1}\} \supset \{0, 1\}$ **then**
      Set $\tilde{\sigma}^c(\mathcal{C}) \leftarrow \tilde{\sigma}^c / V$;
    else
      Return inconsistent $\tilde{\sigma}^c(\mathcal{C})$;
    end
  end
end

Algorithm 1: Defining interfacial values $\tilde{\sigma}$ for each column direction.

References

[1] L.E. Scriven, C.V. Sternling, The Marangoni effects, Nature 187 (1960) 186–188.
[2] R.H. Farahi, A. Passian, T.L. Ferrell, T. Thundat, Microfluidic manipulation via Marangoni forces, Appl. Phys. Lett. 85 (2004) 4237–4239.
[3] A. Kundan, J.L. Plawsky, P.C. Wayner, D.F. Chao, R.J. Sicker, B.J. Motil, T. Lorik, L. Chestney, J. Eustace, J. Zoldak, Thermocapillary phenomena and performance limitations of a wickless heat pipe in microgravity, Phys. Rev. Lett. 114 (2015) 146105.
[4] R.S. Subramanian, R. Balasubramaniam, G. Wozniak, Fluid mechanics of bubbles and drops, in: R. Monti (Ed.), Physics of Fluids in Microgravity, Taylor and Francis, 2002, pp. 149–177.
[5] J. Trice, D. Thomas, C. Favazza, R. Sureshkumar, R. Kalyanaraman, Novel self-organization mechanism in ultrathin liquid films: theory and experiment, Phys. Rev. Lett. 101 (2008) 017802.
[6] N. Dong, L. Kondic, Instability of nanometric fluid films on a thermally conductive substrate, Phys. Rev. Fluids 1 (2016) 063901.
[7] S. Davis, Thermocapillary instabilities, Annu. Rev. Fluid Mech. 19 (1987) 403–435.
[8] R. Craster, O. Matar, Dynamics and stability of thin liquid films, Rev. Mod. Phys. 81 (2009) 1131–1198.
[9] M. Muradoglu, G. Tryggvason, Simulations of soluble surfactants in 3D multiphase flow, J. Comput. Phys. 274 (2014) 737–757.

[10] J.-J. Xu, Z. Li, J. Lowengrub, H. Zhao, A level-set method for interfacial flows with surfactant, J. Comput. Phys. 212 (2006) 590–616.
[11] K.E. Teigen, P. Song, J. Lowengrub, A. Voigt, A diffuse-interface method for two-phase flows with soluble surfactants, J. Comput. Phys. 230 (2011) 375–393.
[12] F. Blanchette, L. Messio, J.W.M. Bush, The influence of surface tension gradients on drop coalescence, Phys. Fluids 21 (2009) 072107.
[13] F. Blanchette, A.M. Shapiro, Drops settling in sharp stratification with and without Marangoni effects, Phys. Fluids 24 (2012) 042104.
[14] M.-C. Lai, Y.-H. Tseng, H. Huang, An immersed boundary method for interfacial flows with insoluble surfactant, J. Comput. Phys. 227 (2008) 7279–7293.
[15] M.R. Booty, M. Siegel, A hybrid numerical method for interfacial fluid flow with soluble surfactant, J. Comput. Phys. 229 (2010) 3864–3883.
[16] F.S. Schranner, N.A. Adams, A conservative interface-interaction model with insoluble surfactant, J. Comput. Phys. 327 (2016) 653–677.
[17] M.A. Drumright-Clarke, Y. Renardy, The effect of insoluble surfactant at dilute concentration on drop breakup under shear with inertia, Phys. Fluids 16 (2004) 14–21.
[18] A.J. James, J. Lowengrub, A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant, J. Comput. Phys. 201 (2004) 685–722.
[19] C. Ma, D. Bothe, Direct numerical simulation of thermocapillary flow based on the Volume of Fluid method, Int. J. Multiph. Flow 37 (2011) 1045–1058.
[20] M.M. Francois, B.K. Swartz, Interface curvature via volume fractions, heights, and mean values on nonuniform rectangular grids, J. Comput. Phys. 229 (2010) 527–540.
[21] J. López, J. Hernández, On reducing interface curvature computation errors in the height function technique, J. Comput. Phys. 229 (2010) 4855–4868.
[22] A. Alexeev, T. Gambaryan-Roisman, P. Stephan, Marangoni convection and heat transfer in thin liquid films on heated walls with topography: experiments and numerical study, Phys. Fluids 17 (2005) 062106.
[23] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, J. Comput. Phys. 228 (2009) 5838–5866.
[24] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries, J. Comput. Phys. 190 (2003) 572–600.
[25] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[26] L.D. Landau, E.M. Lifshitz, Fluid Mechanics, 2nd, vol. 6, Pergamon Press, Oxford, 1987.
[27] V.G. Levich, V.S. Krylov, Surface-tension-driven phenomena, Annu. Rev. Fluid Mech. 1 (1969) 293–316.
[28] S. Popinet, Gerris flow solver, http://gfs.sourceforge.net/wiki/index.php, 1999.
[29] S. Afkhami, M. Bussmann, Height functions for applying contact angles to 2D VOF simulations, Int. J. Numer. Methods Fluids 57 (2008) 453–472.
[30] E. Aulisa, S. Manservisi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry, J. Comput. Phys. 225 (2007) 2301–2319.
[31] G. Wozniak, J. Siekmann, J. Srulijes, Thermocapillary bubble and drop dynamics under reduced gravity-survey and prospects, Z. Flugwiss. Weltraumforsch. 12 (1988) 137–144.
[32] S. Nas, G. Tryggvason, Thermocapillary interaction of two bubbles or drops, Int. J. Multiph. Flow 29 (2003) 1117–1135.
[33] M. Herrmann, J.M. Lopez, P. Brady, M. Raessi, Thermocapillary motion of deformable drops and bubbles, in: Proceedings of the Summer Program, Stanford University, Center for Turbulence Research, 2008, pp. 155–170.
[34] N.O. Young, J.S. Goldstein, M.J. Block, The motion of bubbles in a vertical temperature gradient, J. Fluid Mech. 6 (1959) 350–356.
[35] S. Karpitschka, H. Riegler, Quantitative experimental study on the transition between fast and delayed coalescence of sessile droplets with different but completely miscible liquids, Langmuir 26 (2010) 11823–11829.
[36] S. Karpitschka, H. Riegler, Noncoalescence of sessile drops from different but miscible liquids: hydrodynamic analysis of the twin drop contour as a self-stabilizing traveling wave, Phys. Rev. Lett. 109 (2012) 066103.
[37] S. Karpitschka, H. Riegler, Sharp transition between coalescence and non-coalescence of sessile drops, J. Fluid Mech. 743 (2014) R1.
[38] J.M. Lopez-Herrera, A.M. Ganan-Calvo, S. Popinet, M.A. Herrada, A VOF numerical study on the electrokinetic effects in the breakup of electrified jets, Int. J. Multiph. Flow 71 (2015) 14–22.
[39] S. Afkhami, S. Zaleski, M. Bussmann, A mesh-dependent model for applying dynamic contact angles to VOF simulations, J. Comput. Phys. 228 (2009) 5370–5389.
[40] M.J. Rosen, J.T. Kunjappu, Surfactants and Interfacial Phenomena, John Wiley & Sons, 2012.
[41] S.J. Cummins, M.M. Francois, D.B. Kothe, Estimating curvature from volume fractions, Comput. Struct. 83 (2005) 425–434.