Computer Physics Communications 221 (2017) 102–108

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A new conservative vector-valued Allen–Cahn equation and its fast numerical method

Junseok Kim [a], Hyun Geun Lee [b,*]

[a] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea
[b] Department of Mathematics, Kwangwoon University, Seoul 01897, Republic of Korea

ARTICLE INFO

Article history:
Received 22 July 2016
Received in revised form 29 July 2017
Accepted 3 August 2017
Available online 24 August 2017

Keywords:
Vector-valued Allen–Cahn equation
Mass conservation
Operator splitting
Linear multigrid

ABSTRACT

The scalar Allen–Cahn (AC) equation does not conserve the total mass, and its conservative forms have been studied analytically and numerically. Compared to the conservative scalar AC equations, a conservative form of the vector-valued AC equation is less studied. In this study, we introduce a new conservative vector-valued AC equation that conserves total mass and keeps the bulk phase values (away from the interfacial transition region) close to local minima. To solve the equation, we propose a fast numerical method that is based on the operator splitting method. In the proposed method, we split the equation into three subequations, and each subequation is solved in a component-wise manner. As a result, the conservative vector-valued AC equation is solved quickly, and the average CPU time is nearly linear with respect to the number of components. Numerical experiments with three and more components are presented to demonstrate the usefulness of the proposed method.

1. Introduction

The scalar Allen–Cahn (AC) equation was introduced as a mathematical model for antiphase domain coarsening in a binary mixture [1]:

$$\frac{\partial c(\mathbf{x},t)}{\partial t} = -M\left(\frac{F'(c(\mathbf{x},t))}{\epsilon^2} - \Delta c(\mathbf{x},t)\right), \quad \mathbf{x}\in\Omega,\ 0<t\le T, \tag{1}$$

where Ω is a domain in $\mathbb{R}^d$ ($d = 1, 2, 3$). Let $c(\mathbf{x}, t) = m_\alpha/(m_\alpha + m_\beta)$ be the fraction of one of the concentrations of the two components in an alloy, where m_α and m_β are the masses of phases α and β. $M > 0$ is a mobility (we take $M = 1$ for convenience), $F(c) = 0.25c^2(1-c)^2$ is the Helmholtz free-energy density for c, and $\epsilon > 0$ is the gradient energy coefficient.

The scalar AC equation and its modified forms have been applied to several important problems including crystal growth [2–4], grain growth [5–9], image segmentation and image inpainting [10–12], motion by mean curvature [13–19], phase transitions [1,20], and two-phase fluid flows [21].

Generally, the scalar AC equation (1) does not conserve total mass, Rubinstein and Sternberg [22] added a Lagrange multiplier $\beta(t)$ to Eq. (1) to enforce the conservation of mass:

$$\frac{\partial c(\mathbf{x},t)}{\partial t} = -\frac{f(c(\mathbf{x},t))}{\epsilon^2} + \Delta c(\mathbf{x},t) + \beta(t), \tag{2}$$

where $f(c) = F'(c) = c(c-0.5)(c-1)$ and $\beta(t) = \int_\Omega f(c(\mathbf{x},t))d\mathbf{x}/(\epsilon^2\int_\Omega d\mathbf{x})$. This equation has been extensively studied both analytically and numerically [21,23–31]. However, it has a drawback of not keeping the bulk phase values (away from the interfacial transition region) close to zero or one. This is because the Lagrange multiplier is only dependent on time t. Recently, Brassel and Bretin [32] introduced the following conservative scalar AC equation and demonstrated that it exhibits better mass-preserving properties than Eq. (2):

$$\frac{\partial c(\mathbf{x},t)}{\partial t} = -\frac{f(c(\mathbf{x},t))}{\epsilon^2} + \Delta c(\mathbf{x},t) + \beta(t)\sqrt{F(c(\mathbf{x},t))}, \tag{3}$$

where $\beta(t) = \int_\Omega f(c(\mathbf{x},t))d\mathbf{x}/(\epsilon^2\int_\Omega \sqrt{F(c(\mathbf{x},t))}d\mathbf{x})$. Kim et al. [33] proposed a first-order practically unconditionally stable hybrid scheme for solving Eq. (3). Lee [34] proposed first-, second-, third-order and mass-conserving methods for solving Eq. (3).

Compared to many works [21–34] on two components (2) and (3), a conservative form of the vector-valued AC equation (with $N > 2$ components) is less studied [35,36]. Therefore, the main goal of this study is to introduce a new conservative vector-valued AC equation that conserves total mass and keeps the bulk phase values close to zero or one. To solve the equation, we propose a fast numerical method that is based on the operator splitting method [37–40]. In the proposed method, we split the equation into three

* Corresponding author.
E-mail address: leeh1@kw.ac.kr (H.G. Lee).
URL: http://math.korea.ac.kr/~leeh (H.G. Lee).

http://dx.doi.org/10.1016/j.cpc.2017.08.006
0010-4655/

subequations, and each subequation is solved in a component-wise manner. As a result, the conservative vector-valued AC equation is solved quickly, and the average CPU time is nearly linear with respect to the number of components.

This paper is organized as follows. In Section 2, we introduce a new conservative vector-valued AC equation. In Section 3, we propose a fast numerical method for solving the equation. Numerical experiments with three and more components are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Conservative vector-valued Allen–Cahn equation

We consider the evolution of multi-component systems on a polygonal (polyhedral) domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. Let $\mathbf{c} = (c_1, \ldots, c_N)$ be a vector-valued phase-field. The components $\{c_k\}_{k=1}^N$ represent mole fractions of different components in the system. Clearly the total mole fractions must sum to 1,

$$c_1 + \cdots + c_N = 1, \tag{4}$$

and thus the admissible states belong to the Gibbs N-simplex

$$G := \left\{ \mathbf{c} \in \mathbb{R}^N \;\middle|\; \sum_{k=1}^{N} c_k = 1,\ 0 \le c_k \le 1 \right\}.$$

Without loss of generality, it is assumed that the free energy is expressed as follows:

$$\mathcal{E}(\mathbf{c}) := \int_\Omega \left(\frac{F(\mathbf{c})}{\epsilon^2} + \frac{1}{2} \sum_{k=1}^{N} |\nabla c_k|^2 \right) d\mathbf{x},$$

where $F(\mathbf{c}) = \sum_{k=1}^N F(c_k)$. The vector-valued AC equation [41–45] is a gradient flow for $\mathcal{E}(\mathbf{c})$ in the L^2 inner product under the additional constraint (4), which has to hold everywhere at any time. In order to ensure Eq. (4), we use a Lagrange multiplier $\alpha(\mathbf{c})\mathbf{c}/\epsilon^2$ [41,45–56]. The vector-valued AC equation is

$$\frac{\partial \mathbf{c}(\mathbf{x}, t)}{\partial t} = -\frac{\mathbf{f}(\mathbf{c}(\mathbf{x}, t))}{\epsilon^2} + \Delta \mathbf{c}(\mathbf{x}, t) + \frac{\alpha(\mathbf{c}(\mathbf{x}, t))\mathbf{c}(\mathbf{x}, t)}{\epsilon^2}, \tag{5}$$

where $\mathbf{f}(\mathbf{c}) = (f(c_1), \ldots, f(c_N))$ and $\alpha(\mathbf{c}) = \sum_{k=1}^N f(c_k)$. The natural boundary condition for Eq. (5) is the zero Neumann boundary condition:

$$\nabla c_k \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega,$$

where $\mathbf{n}$ is the unit normal vector to $\partial\Omega$. Eq. (5) does not conserve the total mass of each c_k: for $k = 1, \ldots, N$,

$$\begin{aligned} \frac{d}{dt} \int_\Omega c_k \, d\mathbf{x} &= \int_\Omega \frac{\partial c_k}{\partial t} \, d\mathbf{x} = \int_\Omega \left(-\frac{f(c_k)}{\epsilon^2} + \Delta c_k + \frac{\alpha(\mathbf{c}) c_k}{\epsilon^2} \right) d\mathbf{x} \\ &= -\int_\Omega \frac{f(c_k) - \alpha(\mathbf{c}) c_k}{\epsilon^2} \, d\mathbf{x} + \int_{\partial\Omega} \nabla c_k \cdot \mathbf{n} \, ds \\ &= -\int_\Omega \frac{f(c_k) - \alpha(\mathbf{c}) c_k}{\epsilon^2} \, d\mathbf{x} \end{aligned}$$

which is generally non-zero. We here introduce a new conservative vector-valued AC equation:

$$\begin{aligned} \frac{\partial \mathbf{c}(\mathbf{x}, t)}{\partial t} = &-\frac{\mathbf{f}(\mathbf{c}(\mathbf{x}, t))}{\epsilon^2} + \Delta \mathbf{c}(\mathbf{x}, t) + \frac{\alpha(\mathbf{c}(\mathbf{x}, t))\mathbf{c}(\mathbf{x}, t)}{\epsilon^2} \\ &+ G(\mathbf{c}(\mathbf{x}, t))\boldsymbol{\beta}(t), \end{aligned} \tag{6}$$

where $G(\mathbf{c})\boldsymbol{\beta}(t)$ is a space–time dependent Lagrange multiplier, $G(\mathbf{c}) = \sum_{k=1}^N \sqrt{F(c_k)}$, $\boldsymbol{\beta}(t) = (\beta_1(t), \ldots, \beta_N(t))$, and $\beta_k(t) = \int_\Omega (f(c_k) - \alpha(\mathbf{c}) c_k) d\mathbf{x} / (\epsilon^2 \int_\Omega G(\mathbf{c}) d\mathbf{x})$. Then, the total mass of each c_k is conserved: for $k = 1, \ldots, N$,

$$\begin{aligned} \frac{d}{dt} \int_\Omega c_k \, d\mathbf{x} &= \int_\Omega \frac{\partial c_k}{\partial t} \, d\mathbf{x} \\ &= \int_\Omega \left(-\frac{f(c_k)}{\epsilon^2} + \Delta c_k + \frac{\alpha(\mathbf{c}) c_k}{\epsilon^2} + G(\mathbf{c})\beta_k(t) \right) d\mathbf{x} \\ &= -\int_\Omega \frac{f(c_k) - \alpha(\mathbf{c}) c_k}{\epsilon^2} \, d\mathbf{x} + \int_{\partial\Omega} \nabla c_k \cdot \mathbf{n} \, ds \\ &\quad + \beta_k(t) \int_\Omega G(\mathbf{c}) \, d\mathbf{x} \\ &= -\int_\Omega \frac{f(c_k) - \alpha(\mathbf{c}) c_k}{\epsilon^2} \, d\mathbf{x} + \beta_k(t) \int_\Omega G(\mathbf{c}) \, d\mathbf{x} = 0. \end{aligned}$$

3. Numerical method

In this section, we propose a fast numerical method for solving the conservative vector-valued AC equation (6). For simplicity and clarity of exposition, we shall discretize Eq. (6) in a two-dimensional space. One- and three-dimensional discretizations are analogously defined. Let a computational domain be uniformly partitioned with spacing h. The cell center is located at $(x_i, y_j) = ((i-0.5)h, (j-0.5)h)$ for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$. N_x and N_y are the numbers of cells in the x- and y-directions, respectively. Let $\mathbf{c}_{ij}^n$ be an approximation of $\mathbf{c}(x_i, y_j, n\Delta t)$, where Δt is the time step. In this study, we adopt an operator splitting method in which we numerically solve the original problem (6) by successively solving a sequence of simpler problems:

$$\frac{\partial \mathbf{c}}{\partial t} = \Delta \mathbf{c} + \frac{\alpha(\mathbf{c})\mathbf{c}}{\epsilon^2}, \tag{7}$$

$$\frac{\partial \mathbf{c}}{\partial t} = -\frac{\mathbf{f}(\mathbf{c})}{\epsilon^2}, \tag{8}$$

$$\frac{\partial \mathbf{c}}{\partial t} = G(\mathbf{c})\boldsymbol{\beta}. \tag{9}$$

Note that we only need to solve Eqs. (7)–(9) with $c_1, \ldots, c_{N-1}$, because $c_N = 1 - \sum_{k=1}^{N-1} c_k$. First, we solve Eq. (7) by applying a semi-implicit method: for $k = 1, \ldots, N-1$,

$$\frac{{c_{k,ij}}^{n+1,1} - {c_{k,ij}}^{n}}{\Delta t} = \Delta_d {c_{k,ij}}^{n+1,1} + \frac{\alpha(\mathbf{c}_{ij}^n) {c_{k,ij}}^{n}}{\epsilon^2}, \tag{10}$$

where Δ_d is the discrete Laplacian operator by using a standard five-point stencil. The resulting implicit discrete system of Eq. (10) is solved by using a fast solver such as a linear multigrid method [57–60]. Subsequently, Eq. (8) is analytically solved by using the method of separation of variables [61], and the solution is given as follows: for $k = 1, \ldots, N-1$,

$${c_{k,ij}}^{n+1,2} = 0.5 + \frac{{c_{k,ij}}^{n+1,1} - 0.5}{\sqrt{e^{\frac{-\Delta t}{2\epsilon^2}} + ({2c_{k,ij}}^{n+1,1} - 1)^2 \left(1 - e^{\frac{-\Delta t}{2\epsilon^2}}\right)}}. \tag{11}$$

Finally, we discretize Eq. (9) as follows: for $k = 1, \ldots, N-1$,

$$\frac{{c_{k,ij}}^{n+1} - {c_{k,ij}}^{n+1,2}}{\Delta t} = G(\mathbf{c}_{ij}^{n+1,2}){\beta_k}^{n+1,2}. \tag{12}$$

Here, we take ${\beta_k}^{n+1,2} = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} ({c_{k,ij}}^{n} - {c_{k,ij}}^{n+1,2}) / [\Delta t \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} G(\mathbf{c}_{ij}^{n+1,2})]$ to ensure the mass conservation for each c_k:

$$\begin{aligned} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} {c_{k,ij}}^{n+1} &= \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} {c_{k,ij}}^{n+1,2} + \Delta t {\beta_k}^{n+1,2} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} G(\mathbf{c}_{ij}^{n+1,2}) \\ &= \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} {c_{k,ij}}^{n+1,2} \end{aligned}$$

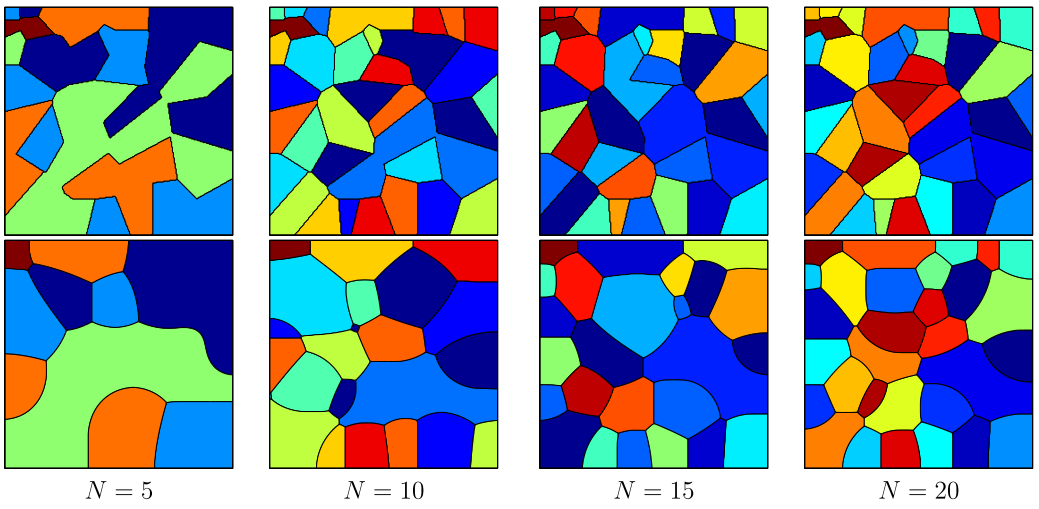

Fig. 1. Evolution of the phase-field $\mathbf{c}$ with 5, 10, 15, and 20 order parameters. The top and bottom rows correspond to $t = 0$ and $t = 0.0352$, respectively.

$$+\frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left(c_{k,ij}^{\,n}-c_{k,ij}^{\,n+1,2}\right)}{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}G(\mathbf{c}_{ij}^{n+1,2})}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}G(\mathbf{c}_{ij}^{n+1,2})$$
$$=\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}c_{k,ij}^{\,n}.$$

Before closing this section, we remark that $\alpha(\mathbf{c})$ in Eq. (10) and $G(\mathbf{c})$ in Eq. (12) are treated explicitly, and thus there is no relation between the solutions c_k at an implicit time level in Eqs. (10) and (12). Therefore, Eqs. (7)–(9) are solved in a decoupling way, i.e.,

Step 1. calculate $\alpha(\mathbf{c}^n)$ from a given $\mathbf{c}^n$.
Step 2. for $k = 1, \ldots, N-1$
update $c_k^{n+1,1}$ by using Eq. (10) and $\alpha(\mathbf{c}^n)$
update $c_k^{n+1,2}$ by using Eq. (11)
end; then we have $\mathbf{c}^{n+1,2}$.
Step 3. calculate $G(\mathbf{c}^{n+1,2})$ from $\mathbf{c}^{n+1,2}$ obtained in *Step 2*.
Step 4. for $k = 1, \ldots, N-1$
update c_k^{n+1} by using Eq. (12) and $G(\mathbf{c}^{n+1,2})$
end; then we have $\mathbf{c}^{n+1}$.

4. Numerical experiments

4.1. Mass conservation for a large system

To verify that the proposed method can be applied to a large system, we perform simulations with 5, 10, 15, and 20 order parameters ($N = 5, 10, 15$, and 20). For each simulation, the initial conditions are randomly chosen patches on $\Omega = [0, 1]\times[0, 1]$, and $\epsilon = 0.0019$, $h = 1/256$, and $\Delta t = \epsilon^2$ are used. Simulations are run for 10,000 time steps and performed on Intel Core i5-3470 CPU @ 3.20 GHz processor and 4 GB RAM. Fig. 1 shows the evolution of the phase-field $\mathbf{c}$. The evolution of $\sum_{k=1}^{N}\left(\int_\Omega (c_k(\mathbf{x},t)-c_k(\mathbf{x},0))d\mathbf{x}\right)^2$ with $N = 5$, 10, 15, and 20 is shown in Fig. 2. It is observed that the total masses of all c_k are conserved for a large system. Table 1 lists the average CPU time (in seconds) over the 10,000 time steps for $N = 5$, 10, 15, and 20. The results in Table 1 suggest that the average CPU time is nearly linear with respect to the number of components.

4.2. Effect of the space–time dependent Lagrange multiplier $G(\mathbf{c}(\mathbf{x}, t))\boldsymbol{\beta}(t)$

To conserve total mass and keep the bulk phase values close to zero or one, we introduce the space–time dependent Lagrange multiplier $G(\mathbf{c}(\mathbf{x}, t))\boldsymbol{\beta}(t)$ in Eq. (6). To examine the effect of

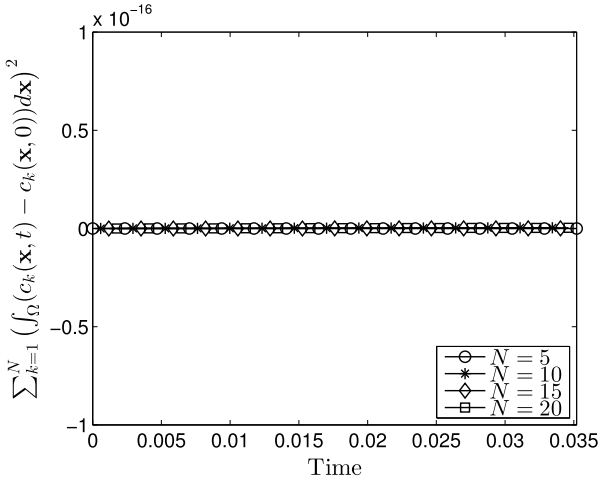

Fig. 2. Evolution of $\sum_{k=1}^{N}\left(\int_\Omega (c_k(\mathbf{x},t)-c_k(\mathbf{x},0))d\mathbf{x}\right)^2$ with $N = 5$, 10, 15, and 20 for $0 \le t \le 0.0352$.

$G(\mathbf{c}(\mathbf{x}, t))\boldsymbol{\beta}(t)$, we compare Eq. (6) to

$$\frac{\partial \mathbf{c}(\mathbf{x},t)}{\partial t} = -\frac{\mathbf{f}(\mathbf{c}(\mathbf{x},t))}{\epsilon^2} + \Delta\mathbf{c}(\mathbf{x},t) + \frac{\alpha(\mathbf{c}(\mathbf{x},t))\mathbf{c}(\mathbf{x},t)}{\epsilon^2} + \bar{\boldsymbol{\beta}}(t), \quad (13)$$

where $\bar{\boldsymbol{\beta}}(t) = (\bar{\beta}_1(t), \ldots, \bar{\beta}_N(t))$ and $\bar{\beta}_k(t) = \int_\Omega (f(c_k) - \alpha(\mathbf{c})c_k)d\mathbf{x}/(\epsilon^2\int_\Omega d\mathbf{x})$. For $N = 3$, the initial conditions are

$$c_1(x,y,0) = \begin{cases} 1 & \text{if } 0.32 \le x, y \le 0.68 \\ 0 & \text{otherwise} \end{cases},$$

$$c_2(x,y,0) = \begin{cases} 1 & \text{if } 1.38 \le x \le 1.62,\ 0.38 \le y \le 0.62 \\ 0 & \text{otherwise} \end{cases}$$

on $\Omega = [0, 2] \times [0, 1]$ (see Fig. 3(a)). Here, we use $\epsilon = 0.0056$, $h = 1/128$, and $\Delta t = h^2$, and define the steady state as the state when the discrete l_2-norm of the difference between $\mathbf{c}^{n+1}$ and $\mathbf{c}^n$ becomes less than 10^{-6}. Fig. 3(b) and (c) show the evolution of the numerical solutions of Eqs. (13) and (6), respectively. Note that the initial square shape converges to a circle while conserving the mass. When the initial feature is sufficiently large (in the case of c_1), both models (6) and (13) give a circular steady state. It should be noted that the order parameter c_1 in the bulk phases is -0.0060 or 0.9907 for Eq. (13) but is 0 or 1 for Eq. (6). In the case of model (13), the mass loss is globally corrected by using

Table 1
Average CPU times (s) for $N = 5$, 10, 15, and 20.

N	5	10	15	20	aN^p
Average CPU time	0.2044	0.4358	0.6761	0.9162	$\approx 0.0358N^{1.0836}$

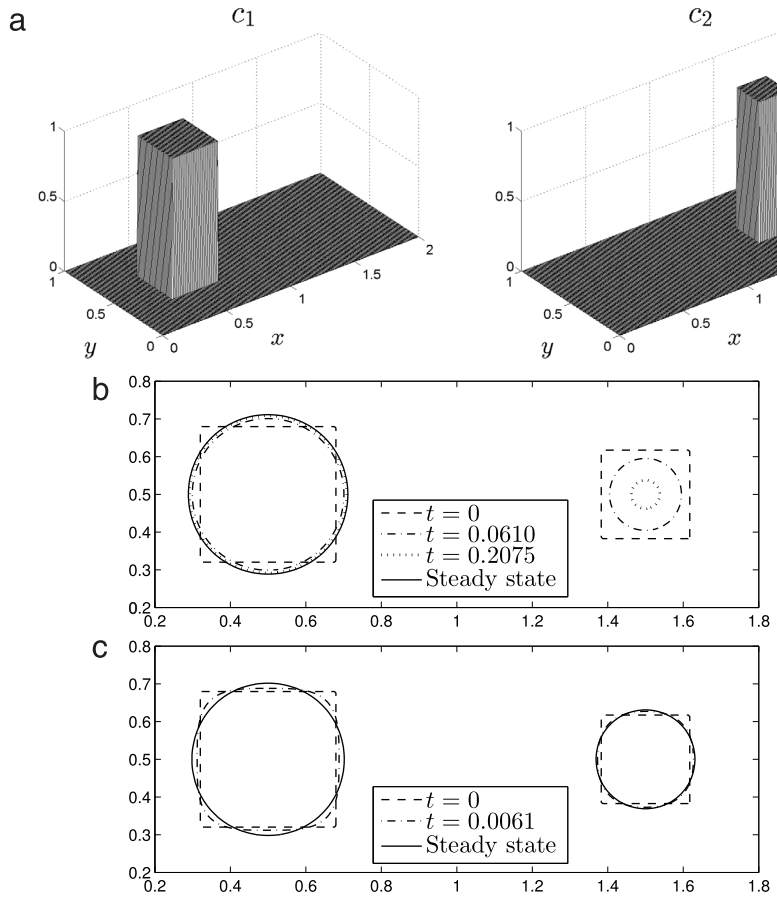

Fig. 3. (a) Initial conditions of c_1 and c_2. (b) and (c) show the evolution of the numerical solutions of Eqs. (13) and (6), respectively.

the time dependent Lagrange multiplier $\bar{\boldsymbol{\beta}}(t)$, and thus the order parameter slightly shifts from its expected values 0 and 1 in the bulk phases. However, in the case of model (6), the order parameter has 0 or 1 in the bulk phases since the mass loss is corrected in the interfacial region by using the space–time dependent Lagrange multiplier $G(\mathbf{c}(\mathbf{x}, t))\boldsymbol{\beta}(t)$. When the initial geometry is small (in the case of c_2), two models give different results. In the case of model (13), the square first evolves to a circle and then vanishes ($c_2 = 0.0275$ for the whole domain at the steady state and thus a 0.5 level set of c_2 cannot be taken to represent an interface). On the other hand, for Eq. (6), the circle remains as shown in Fig. 3(c).

4.3. Mass-conserved motion by curvature

We consider the following problem [22]: given M circles with radii $r_1, \ldots, r_M$ that do not intersect, the sharp interface problem for mass-conserved motion by curvature results in the following system of ordinary differential equations:

$$\frac{dr_i}{dt} = -\frac{1}{r_i} + \frac{M}{\sum_{m=1}^{M} r_m} \qquad \text{for } i = 1, \ldots, M, \tag{14}$$

together with the condition of mass-conservation $0 = \frac{1}{2}\frac{d}{dt}(\sum_{m=1}^{M} r_m^2)$, where the initial radii $r_1(0), \ldots, r_M(0)$ are known. In this system, larger circles grow at the expense of smaller ones and thus smaller ones eventually disappear and the fattest survives. For $N = 3$, we use this problem, where the component c_1 occupies three circles, the component c_2 occupies the other three circles, and the component c_3 is present outside the six circles (see Fig. 4). The initial conditions on $\Omega = [0, 3] \times [0, 2]$ are

$$\begin{aligned} c_1(x, y, 0) = {} & d(x, y; 0.7, 1.55, r_1(0)) + d(x, y; 1.5, 1.55, r_2(0)) \\ & + d(x, y; 2.3, 1.55, r_3(0)), \\ c_2(x, y, 0) = {} & d(x, y; 0.7, 0.7, \tilde{r}_1(0)) + d(x, y; 1.5, 0.7, \tilde{r}_2(0)) \\ & + d(x, y; 2.3, 0.7, \tilde{r}_3(0)). \end{aligned}$$

Here, $d(x, y; a, b, r) := \frac{1}{2}\left(1 + \tanh\left(\frac{r - \sqrt{(x-a)^2 + (y-b)^2}}{2\sqrt{2}\epsilon}\right)\right)$, $r_1(0) = 0.2$, $r_2(0) = 0.3$, $r_3(0) = 0.22$, $\tilde{r}_1(0) = 0.4$, $\tilde{r}_2(0) = 0.25$, and $\tilde{r}_3(0) = 0.27$, and we use $\epsilon = 0.0056$, $h = 1/128$, and $\Delta t = 10^{-2}h^2$. For the reference solution of r_1, r_2, r_3, $\tilde{r}_1$, $\tilde{r}_2$, and $\tilde{r}_3$, we numerically solve the ordinary differential equations using the fourth order Runge–Kutta method. Figs. 5 and 6 show the approximate and reference solutions of r_1, r_2, r_3, $\tilde{r}_1$, $\tilde{r}_2$, and $\tilde{r}_3$ using models (13) and (6), respectively. When the radius is sufficiently large, both models (6) and (13) accurately predict the evolution of the radius. However, when the radius is small (especially r_1), the circle of model (13) disappears faster than the time predicted by the system (14). On the other hand, even for r_1, the numerically calculated radius of model (6) agrees well with the corresponding reference radius.

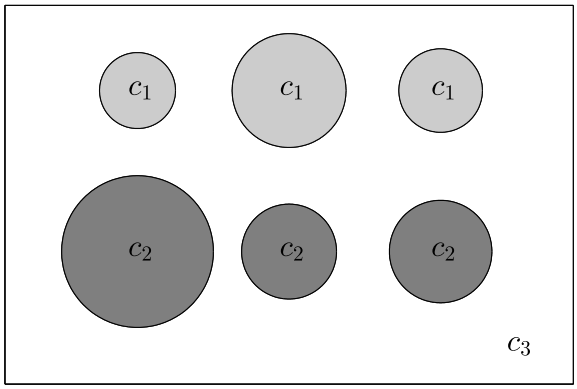

Fig. 4. Initial configuration for the conservative vector-valued AC equation ($N = 3$).

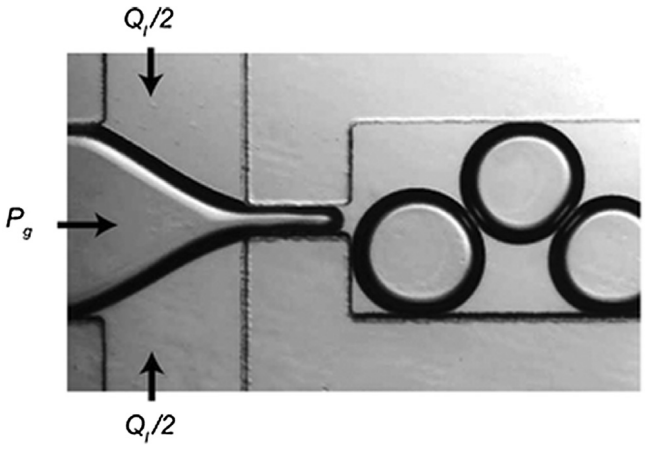

Fig. 7. Droplets in a microchannel. *Source:* Reprinted from Marmottant and Raven [62] with permission from the Royal Society of Chemistry.

4.4. Droplets in a microchannel

Finally, we consider a train of droplets in a microchannel (see Fig. 7) as an application of the conservative vector-valued AC equation.

If we use the modified Navier–Stokes equations and the convective conservative scalar AC equation to simulate the phenomenon in Fig. 7,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \Delta \mathbf{u} + \frac{1}{We} \mathbf{sf}(c), \tag{15}$$

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

$$\frac{\partial c}{\partial t} + \nabla \cdot (c\mathbf{u}) = \frac{1}{Pe} \left(-\frac{f(c)}{\epsilon^2} + \Delta c + \beta \sqrt{F(c)} \right), \tag{17}$$

then we obtain the results in Fig. 8 unless an extremely fine grid is used.

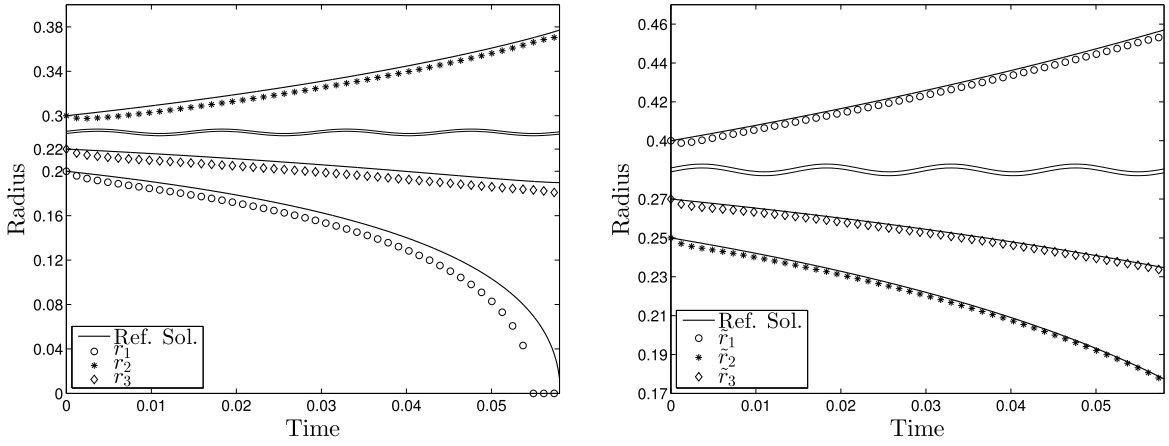

Fig. 5. Evolution of the approximate radii $r_1, r_2, r_3, \tilde{r}_1, \tilde{r}_2$, and $\tilde{r}_3$ using the model (13). Solid lines represent the corresponding reference solutions.

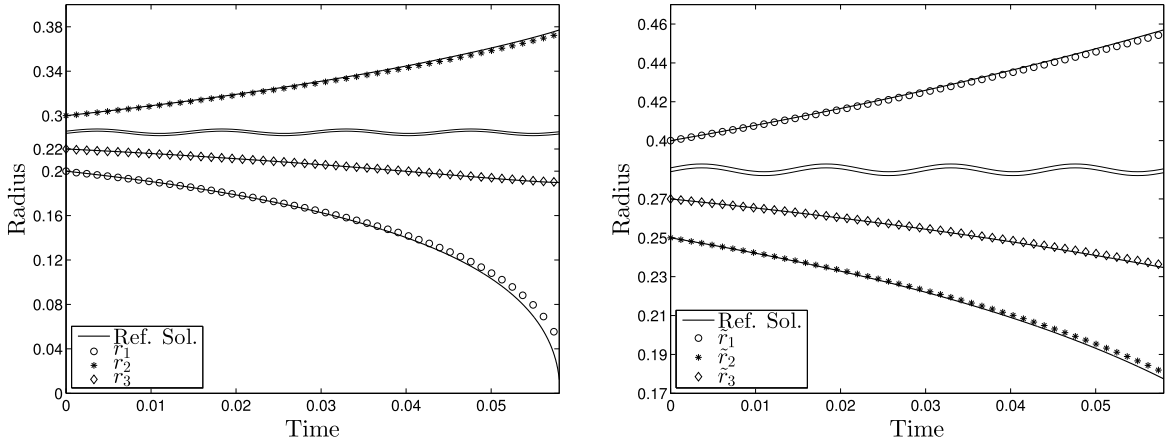

Fig. 6. Evolution of the approximate radii $r_1, r_2, r_3, \tilde{r}_1, \tilde{r}_2$, and $\tilde{r}_3$ using the model (6). Solid lines represent the corresponding reference solutions.

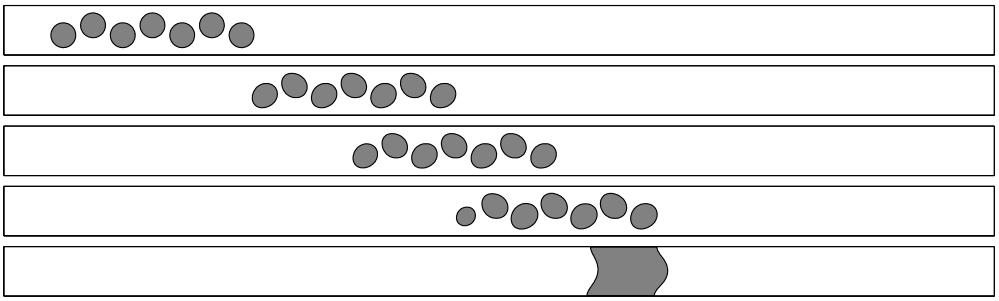

Fig. 8. Evolution of droplets in a microchannel obtained by solving Eqs. (15)–(17). Times are $t = 0$, 0.4883, 0.7324, 0.9766, and 1.2207 (from top to bottom).

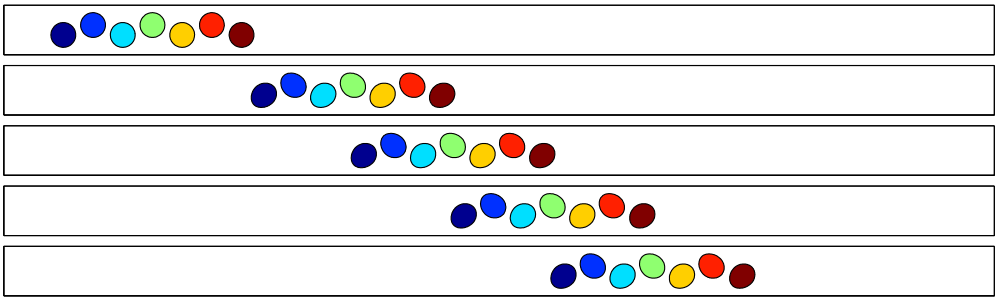

Fig. 9. Evolution of droplets in a microchannel obtained by solving Eqs. (18)–(20). Times are $t = 0$, 0.4883, 0.7324, 0.9766, and 1.2207 (from top to bottom).

Here, we assume that the fluids are density- and viscosity-matched and the gravity is neglected. $\mathbf{u}$ is the velocity, p is the pressure, $\mathbf{sf}(c) = -6\sqrt{2}\epsilon\nabla\cdot(\nabla c/|\nabla c|)|\nabla c|\nabla c$ is the surface tension force, ϵ is redefined according to the scaling, and $\beta = \int_\Omega f(c)d\mathbf{x}/(\epsilon^2\int_\Omega\sqrt{F(c)}d\mathbf{x})$. The dimensionless parameters are the Reynolds number, $Re = \rho_c U_c L_c/\eta_c$, the Weber number, $We = \rho_c U_c^2 L_c/\sigma$, and the Péclet number, $Pe = U_c L_c/M_c$, where L_c, U_c, ρ_c, η_c, and M_c are the characteristic length, velocity, density, viscosity, and mobility, respectively, and σ is the surface tension coefficient. Eqs. (15) and (16) are solved by using Chorin's projection method [63], and Eq. (17) is solved by using an operator splitting method.

In Fig. 8, we used the following initial condition and velocity:

$$\begin{aligned} c(x,y,0) &= \sum_{k=1}^{7} d(x,y;0.6(k+1),0.5 \\ &\quad + 0.1(2 \bmod (k+1,2)-1),0.25), \\ u(x,y,0) &= 40(y-y^2), \\ v(x,y,0) &= 0 \qquad \text{on } \Omega = [0,20]\times[0,1]. \end{aligned}$$

Furthermore, the nonreflecting outlet boundary condition [64] for Eqs. (15) and (16), the zero Neumann boundary condition for Eq. (17), and the parameters $\epsilon = 0.0075$, $h = 1/64$, $\Delta t = 0.1h^2$, $Re = 100$, $We = 10$, $Pe = 1$ were used.

To avoid the merging of droplets, we consider the modified Navier–Stokes equations and the convective conservative vector-valued AC equation: for $k = 1,\ldots,N$,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \frac{1}{Re}\Delta\mathbf{u} + \mathbf{SF}(\mathbf{c}), \tag{18}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{19}$$

$$\frac{\partial c_k}{\partial t} + \nabla\cdot(c_k\mathbf{u}) = \frac{1}{Pe}\left(-\frac{f(c_k)}{\epsilon^2} + \Delta c_k + \frac{\alpha(\mathbf{c})c_k}{\epsilon^2} + G(\mathbf{c})\beta_k\right), \tag{20}$$

where $\mathbf{SF}(\mathbf{c}) = \sum_{i=1}^{N-1}\left(\sum_{j=i+1}^{N} 0.5[\mathbf{sf}(c_i)+\mathbf{sf}(c_j)]\delta(c_i,c_j)/We_{ij}\right)$ is the surface tension force, $\delta(c_i,c_j) = 5c_ic_j$, $We_{ij} = \rho_c U_c^2 L_c/\sigma_{ij}$ is the Weber number of fluids i and j, and σ_{ij} is the physical surface tension coefficient between fluids i and j. We solve Eq. (20) in a decoupling way by applying the proposed method.

Fig. 9 shows the evolution of the phase-field $\mathbf{c}$ with the initial conditions $c_k(x,y,0) = d(x,y;0.6(k+1),0.5+0.1(2 \bmod (k+1,2)-1),0.25)$ for $k = 1,\ldots,7$ and $We_{ij} = 10$. The remaining conditions and parameters are the same as those used in Fig. 8. The droplets move from left to right in the microchannel by flow without merging.

5. Conclusions

In this study, we introduced a new conservative vector-valued AC equation and proposed its fast numerical method. We observed that the conservative vector-valued AC equation conserves total mass and keeps the bulk phase values close to zero or one. Additionally, the numerical method quickly solves the equation, and specifically, the average CPU time is nearly linear with respect to the number of components. Numerical experiments demonstrate that the new conservative vector-valued AC equation and its fast numerical method are capable of handling a large volume-conserved system.

Acknowledgments

The authors thank the reviewers for the constructive and helpful comments on the revision of this article. The first author (J.S. Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2014R1A2A2A01003683). The corresponding author (H.G. Lee) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2017R1D1A1B03034619).

References

[1] S.M. Allen, J.W. Cahn, Acta Metall. 27 (6) (1979) 1085–1095.
[2] A.A. Wheeler, W.J. Boettinger, G.B. McFadden, Phys. Rev. A 45 (10) (1992) 7424.
[3] M. Cheng, J.A. Warren, J. Comput. Phys. 227 (12) (2008) 6241–6248.
[4] Y. Li, H.G. Lee, J. Kim, J. Cryst. Growth 321 (1) (2011) 176–182.

[5] L.-Q. Chen, W. Yang, Phys. Rev. B 50 (21) (1994) 15752.
[6] I. Steinbach, F. Pezzolla, B. Nestler, M. Seeßelberg, R. Prieler, G.J. Schmitz, J.L. Rezende, Physica D 94 (3) (1996) 135–147.
[7] D. Fan, C. Geng, L.-Q. Chen, Acta Mater. 45 (3) (1997) 1115–1126.
[8] M.T. Lusk, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, Vol. 455, No. 1982, The Royal Society, 1999, pp. 677–700.
[9] R. Kobayashi, J.A. Warren, W.C. Carter, Physica D 140 (1) (2000) 141–150.
[10] M. Beneš, V. Chalupecký, K. Mikula, Appl. Numer. Math. 51 (2–3) (2004) 187–205.
[11] J.A. Dobrosotskaya, A.L. Bertozzi, IEEE Trans. Image Process. 17 (5) (2008) 657–663.
[12] Y. Li, J. Kim, Comput. Math. Appl. 62 (2) (2011) 737–745.
[13] L.C. Evans, H.M. Soner, P.E. Souganidis, Comm. Pure Appl. Math. 45 (9) (1992) 1097–1123.
[14] T. Ilmanen, et al., J. Differential Geom 38 (2) (1993) 417–461.
[15] M. Katsoulakis, G.T. Kossioris, F. Reitich, J. Geom. Anal. 5 (2) (1995) 255–279.
[16] M. Beneš, K. Mikula, Acta Math. Univ. Comenian. 67 (1) (1998) 17–42.
[17] W. Bao, Comput. Math. Appl. 46 (8–9) (2003) 1211–1228.
[18] X. Feng, A. Prohl, Numer. Math. 94 (1) (2003) 33–65.
[19] T. Ohtsuka, Asymptot. Anal. 56 (2) (2008) 87–123.
[20] S. Zhai, X. Feng, Y. He, Comput. Phys. Comm. 185 (10) (2014) 2449–2455.
[21] X. Yang, J.J. Feng, C. Liu, J. Shen, J. Comput. Phys. 218 (1) (2006) 417–428.
[22] J. Rubinstein, P. Sternberg, IMA J. Appl. Math. 48 (3) (1992) 249–264.
[23] M.J. Ward, SIAM J. Appl. Math. 56 (5) (1996) 1247–1279.
[24] L. Bronsard, B. Stoth, SIAM J. Math. Anal. 28 (4) (1997) 769–807.
[25] D. Stafford, M.J. Ward, B. Wetton, European J. Appl. Math. 12 (1) (2001) 1–24.
[26] M. Conti, B. Meerson, A. Peleg, P.V. Sasorov, Phys. Rev. E 65 (4) (2002) 046117.
[27] Z. Zhang, H. Tang, Comput. & Fluids 36 (8) (2007) 1307–1318.
[28] J. Shen, X. Yang, J. Comput. Phys. 228 (8) (2009) 2978–2992.
[29] X. Chen, D. Hilhorst, E. Logak, Interfaces Free Bound. 12 (4) (2010) 527–549.
[30] J. Shen, X. Yang, SIAM J. Sci. Comput. 32 (3) (2010) 1159–1179.
[31] M. Beneš, S. Yazaki, M. Kimura, Math. Bohem. 136 (4) (2011) 429–437.
[32] M. Brassel, E. Bretin, Math. Methods Appl. Sci. 34 (10) (2011) 1157–1180.
[33] J. Kim, S. Lee, Y. Choi, Internat. J. Engrg. Sci. 84 (2014) 11–17.
[34] H.G. Lee, Comput. Math. Appl. 72 (3) (2016) 620–631.
[35] H. Garcke, B. Nestler, B. Stinner, F. Wendler, Math. Models Methods Appl. Sci. 18 (08) (2008) 1347–1381.
[36] L. Blank, H. Garcke, L. Sarbu, V. Styles, IMA J. Numer. Anal. 33 (4) (2013) 1126–1155.
[37] G. Strang, SIAM J. Numer. Anal. 5 (3) (1968) 506–517.
[38] D. Goldman, T.J. Kaper, SIAM J. Numer. Anal. 33 (1) (1996) 349–367.
[39] Y. Li, H.G. Lee, D. Jeong, J. Kim, Comput. Math. Appl. 60 (6) (2010) 1591–1606.
[40] H.G. Lee, J.-Y. Lee, Comput. Math. Appl. 68 (3) (2014) 174–184.
[41] H. Garcke, B. Nestler, B. Stoth, Physica D 115 (1) (1998) 87–108.
[42] H. Garcke, V. Styles, Interfaces Free Bound. 6 (3) (2004) 271–294.
[43] R. Kornhuber, R. Krause, Comput. Vis. Sci. 9 (2) (2006) 103–116.
[44] D.A. Kay, A. Tomasi, IEEE Trans. Image Process. 18 (10) (2009) 2330–2339.
[45] H.G. Lee, J. Kim, Comput. Phys. Comm. 183 (10) (2012) 2107–2115.
[46] B. Nestler, A. Wheeler, Physica D 138 (1) (2000) 114–133.
[47] B. Nestler, A.A. Wheeler, L. Ratke, C. Stöcker, Physica D 141 (1) (2000) 133–154.
[48] J.R. Green, A Comparison of Multiphase Models and Techniques (Ph.D. thesis), University of Leeds, 2007.
[49] J. Kim, Comput. Methods Appl. Mech. Engrg. 196 (45) (2007) 4779–4788.
[50] J. Kim, Comput. Methods Appl. Mech. Engrg. 198 (37) (2009) 3105–3112.
[51] L. Vanherpe, F. Wendler, B. Nestler, S. Vandewalle, Math. Comput. Simulation 80 (7) (2010) 1438–1448.
[52] H.G. Lee, J.-W. Choi, J. Kim, Physica A 391 (4) (2012) 1009–1019.
[53] H.G. Lee, J. Kim, Eur. J. Mech. B Fluids 42 (2013) 37–46.
[54] H.G. Lee, J. Kim, Eur. J. Mech. B Fluids 49 (2015) 77–88.
[55] H.G. Lee, J. Kim, Physica A 423 (2015) 33–50.
[56] H.G. Lee, J. Kim, Physica A 387 (19) (2008) 4787–4799.
[57] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, SIAM, 2000.
[58] U. Trottenberg, C.W. Oosterlee, A. Schüller, Multigrid, Academic press, 2000.
[59] P. Bauer, V. Klement, T. Oberhuber, V. Žabka, Comput. Phys. Comm. 200 (2016) 50–56.
[60] Y. Li, H.G. Lee, B. Xia, J. Kim, Comput. Phys. Comm. 200 (2016) 108–116.
[61] A. Stuart, A.R. Humphries, Dynamical Systems and Numerical Analysis, Vol. 2, Cambridge University Press, 1998.
[62] P. Marmottant, J.-P. Raven, Soft Matter 5 (18) (2009) 3385–3388.
[63] A.J. Chorin, J. Comput. Phys. 135 (2) (1997) 118–125.
[64] G. Jin, M. Braza, J. Comput. Phys. 107 (2) (1993) 239–253.