

# Time integration for diffuse interface models for two-phase flow



Sebastian Aland

Institut für wissenschaftliches Rechnen, TU Dresden, 01062 Dresden, Germany

## ARTICLE INFO

### Article history:

Received 10 June 2013

Received in revised form 21 November 2013

Accepted 29 December 2013

Available online 7 January 2014

### Keywords:

Time integration

Diffuse interface model

Dominant surface tension

Time stability

CFL condition

Navier–Stokes

Cahn–Hilliard

Linearization

## ABSTRACT

We propose a variant of the  $\theta$ -scheme for diffuse interface models for two-phase flow, together with three new linearization techniques for the surface tension. These involve either additional stabilizing force terms, or a fully implicit coupling of the Navier–Stokes and Cahn–Hilliard equation.

In the common case that the equations for interface and flow are coupled explicitly, we find a time step restriction which is very different to other two-phase flow models and in particular is independent of the grid size. We also show that the proposed stabilization techniques can lift this time step restriction.

Even more pronounced is the performance of the proposed fully implicit scheme which is stable for arbitrarily large time steps. We demonstrate in a Taylor–flow application that this superior coupling between flow and interface equation can decrease the computation time by several orders of magnitude.

© 2014 Elsevier Inc. All rights reserved.

## 1. Introduction

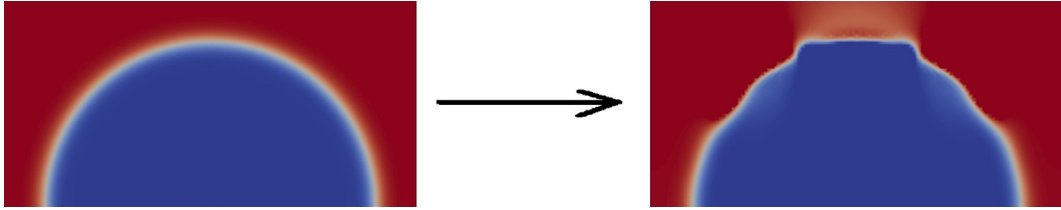
The numerical simulation of two-phase flows has reached some importance in microfluidic applications. In the last decade, diffuse interface (or phase-field) models have become a valuable alternative to the more established sharp interface methods (e.g. Level-Set, Arbitrary Lagrangian–Eulerian, Volume-Of-Fluid). The advantages of diffuse interface methods include the possibility to easily handle moving contact lines and topological transitions as well as the fact that they do not require any reinitialization or convection stabilization. The corresponding equations involve a Navier–Stokes (NS) equation coupled to a convective Cahn–Hilliard (CH) equation. A lot of efficient spacial discretization techniques and solvers for these equations have been proposed (e.g. [20]). However, not much work has been done on time integration strategies and efficient coupling between the NS and the CH equation, which we will address in this paper.

But at first, let us introduce the diffuse interface method more carefully. The method was originally developed to model solid–liquid phase transitions, see e.g. [6,14,26]. The interface thereby is represented as a thin layer of finite thickness and an auxiliary function, the so-called phase field, is used to indicate the phases. The phase field function varies smoothly between distinct values in both phases and the interface can be associated with an intermediate level set of the phase field function. Diffuse interface approaches for mixtures of two immiscible, incompressible fluids lead to the NS–CH equations and have been considered by several authors, see e.g. [19,15,20,11]. The simplest model reads:

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

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

E-mail address: [sebastian.aland@tu-dresden.de](mailto:sebastian.aland@tu-dresden.de).



**Fig. 1.** Evolution of a semi-circular bubble under the diffuse interface model with too large time steps. Left: initial shape; Right: Evolved shape after ten time steps.

$$\partial_t c + \mathbf{u} \cdot \nabla c = \nabla \cdot (M(c) \nabla \mu), \quad (3)$$

$$\mu = \tilde{\sigma} \epsilon^{-1} W'(c) - \tilde{\sigma} \epsilon \Delta c, \quad (4)$$

in the domain  $\Omega$ . Here  $\mathbf{u}$ ,  $p$ ,  $c$  and  $\mu$  are the velocity, pressure, phase field variable and chemical potential, respectively. The function  $W(c)$  is a double well potential, here we use  $W = 1/4(c^2 - 1)^2$  which ensures that  $c \approx \pm 1$  in two fluid phases, respectively.

The function  $M(c)$  is a mobility,  $\epsilon$  defines a length scale over which the interface is smeared out. In general for the diffuse interface fluid method, it is desirable to keep  $M$  small such that the phase field function is primarily moved by advection. At the same time the mobility needs to be big enough to ensure that the interface profile stays accurately modeled and the interface thickness is approximately constant. Furthermore,  $\mathbf{D}(\mathbf{u}) = \nabla \mathbf{u} + \nabla \mathbf{u}^T$  is the strain tensor,  $\rho(c)$ ,  $\nu(c)$  and  $\mathbf{F}$  are the (phase dependent) density, viscosity and body force. The parameter  $\tilde{\sigma}$  is a scaled surface tension which is related to the physical surface tension by  $\tilde{\sigma} = \sigma \frac{3}{2\sqrt{2}}$ . There are efficient solvers available to discretize and solve Eqs. (1)–(4) in space (see e.g. [20]).

Surface tension is a major component of all multiphase fluid models and hence various spatial discretizations of the surface tension force for diffuse–interface models have been proposed (e.g. [22]). The surface tension force  $\mu \nabla c$  introduces a strong coupling between the NS equation providing the flow field and the CH equation evolving the phase field. This is very similar to sharp interface models for two-phase flow where the same interface-to-flow coupling introduces a severe time step restriction of the form [7,9]:

$$\tau < C \rho^{\frac{1}{2}} h^{\frac{3}{2}} \sigma^{-\frac{1}{2}}. \quad (5)$$

Here,  $\tau$  is the maximum time step size,  $\rho$  the average density of both fluids and  $h$  the grid size. The above CFL-like restriction is particularly strong for large effective surface tensions, e.g. when small physical length scales are considered. It is usually assumed that this restriction also holds for diffuse interface models (e.g. in [21]). In Section 6.1 we will show that this assumption is wrong.

However, also for diffuse interface models there is some time step restriction which can make computations extremely costly, even in cases when the interface is supposed to hardly move. Fig. 1 shows such a case of a perfectly circular interface, which is almost stationary. However, if too big time steps are chosen, even such an equilibrated surface will start to wobble and finally break up. In sharp interface models, there are techniques to overcome such time step restrictions [17]. To the best of the authors' knowledge there is no such technique available for diffuse interface models yet. We will develop techniques to improve the coupling between the NS and the CH equations, which will turn out to lift the time step restrictions significantly.

Apart from increasing the computational performance, there is a second reason to develop better time integration schemes for diffuse interface models. The simple time discretization schemes available often imply the need to stabilize the system by choosing a relatively high CH mobility. But this high artificial diffusion perturbs the simulation results since matched asymptotic analysis shows the convergence of diffuse–interface methods toward the sharp interface equations only for small CH mobility [1]. Therefore better time integration strategies would not only speed-up the simulations but also allow to take smaller (more physical) CH mobility and thus improve the accuracy of diffuse–interface methods.

The structure of the remaining paper is as follows. Sections 2 and 3 will introduce a simple variant of the  $\theta$  scheme as well as a block Gauss–Seidel coupling strategy. The main attention is given to Section 4 where some new improved coupling techniques for diffuse interface models are presented. The solution of the resulting systems is discussed in Section 5. In Section 6 we perform numerical tests. In particular a CFL-condition for diffuse interface methods is numerically derived and it is shown that the new proposed coupling methods can, for some problems, result in an extreme gain of performance. Finally, conclusions are drawn in Section 7.

## 2. Time discretization: a variant of the $\theta$ -scheme

In this section we adopt the well-known  $\theta$ -scheme for the time discretization of the NS–CH equations. Let the time interval  $[0, T]$  be divided in  $N$  subintervals of size  $\tau^n$ ,  $n = 1 \dots N$ . We define the discrete time derivative of a (solution) variable  $v$  to be  $d_t v^{n+1} := (v^{n+1} - v^n) / \tau^n$ , where the upper index denotes the time step number. For a shorter notation we introduce

$$\mathbf{g}(\mathbf{u}, c, \mu) := -\rho(c)\mathbf{u} \cdot \nabla \mathbf{u} + \nabla \cdot (\nu(c)\mathbf{D}(\mathbf{u})) + \mathbf{F}(c) + \mu \nabla c, \quad (6)$$

$$f(\mathbf{u}, c, \mu) := -\mathbf{u} \cdot \nabla c + \nabla \cdot (M(c)\nabla \mu). \quad (7)$$

For a constant  $\theta \in [0, 1]$  we propose the following variant of the  $\theta$ -scheme:

$$\rho^{n+\frac{1}{2}} \partial_t \mathbf{u}^{n+1} + \nabla p^{n+1} = \theta \mathbf{g}(\mathbf{u}^{n+1}, c^{n+1}, \mu^{n+1}) + (1-\theta) \mathbf{g}(\mathbf{u}^n, c^n, \mu^n), \quad (8)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (9)$$

$$\partial_t c^{n+1} = \theta f(\mathbf{u}^{n+1}, c^{n+1}, \mu^{n+1}) + (1-\theta) f(\mathbf{u}^n, c^n, \mu^n), \quad (10)$$

$$\mu^{n+1} = \tilde{\sigma} \epsilon^{-1} W'(c^{n+1}) - \tilde{\sigma} \epsilon \Delta c^{n+1}, \quad (11)$$

where  $\rho^{n+\frac{1}{2}} = (\rho(c^{n+1}) + \rho(c^n))/2$  denotes an approximation to the intermediate density. Note that the system (8)–(11) differs from the standard  $\theta$ -scheme by the use of  $\rho^{n+\frac{1}{2}}$ . To derive a standard  $\theta$ -scheme one would have to divide the momentum equation by  $\rho(c)$  to isolate the time derivative of  $\mathbf{u}$ . Then the  $\theta$ -scheme can be applied to the system  $\partial_t \mathbf{u} = [g(\mathbf{u}, c, \mu) - \nabla p]/\rho(c)$ . This leads to more complicated equations requiring much more implementation effort. The above variant circumvents this problem by using  $\rho^{n+\frac{1}{2}}$ . One easily verifies that the method has the same consistency order and stability properties as the standard  $\theta$ -scheme. To be more precise, the method is of second order if  $\theta = 0.5$  and of first order if  $\theta \neq 0.5$ . The most important cases are  $\theta = 1$  (backward Euler) and  $\theta = 0.5$  (Crank–Nicolson). Both are  $A$ -stable and are often used to discretize two-phase flow problems in practice. We will also restrict our numerical experiments to these two cases. The biggest disadvantage of the Crank–Nicolson scheme is that it has no smoothing property, i.e. it does not smooth high frequencies. Therefore it is in some cases appropriate to use the optimally smoothing backward Euler scheme, although it has lower order.

### 3. Linearization and coupling

The first question that arises when looking at Eqs. (8)–(11) is: How to couple the NS and CH equations. The nonlinear coupling between  $\mathbf{u}^{n+1}$ ,  $c^{n+1}$  and  $\mu^{n+1}$  can be treated by several decoupling strategies. One usually applies an iterative strategy where each time step requires multiple solves of the governing equations until the approximation error in the solution variables is sufficiently small. To our knowledge, only sequential coupling of the NS and CH equations has been considered in the literature. Hence, both equations are solved separately, using the solution of the other equation explicitly from a previous computation.

#### 3.1. Block Gauss–Seidel decoupling

The simplest decoupling strategy is the block Gauss–Seidel method. This method is widely used, e.g. in [11,16]. We use subscript indices to denote the variables of the sub-iteration while superscripts still denote the time step number. Then the block Gauss–Seidel strategy applied in every time step reads:

1. Initialize the sub-iteration with the values from the last time step:

$$\mathbf{u}_0 = \mathbf{u}^n, \quad c_0 = c^n, \quad \mu_0 = \mu^n \quad (12)$$

2. for  $k = 0, 1, \dots$

- Solve the NS equation to get  $\mathbf{u}_{k+1}$  and  $p_{k+1}$ :

$$\rho^{n+\frac{1}{2}} \partial_t \mathbf{u}_{k+1} + \nabla p_{k+1} = \theta \mathbf{g}(\mathbf{u}_{k+1}, c_k, \mu_k) + (1-\theta) \mathbf{g}(\mathbf{u}^n, c^n, \mu^n), \quad (13)$$

$$\nabla \cdot \mathbf{u}_{k+1} = 0, \quad (14)$$

where  $\rho^{n+\frac{1}{2}}$  is approximated by  $(\rho(c_k) + \rho(c^n))/2$ .

- Solve the CH equation to get  $c_{k+1}$  and  $\mu_{k+1}$ :

$$\partial_t c_{k+1} = \theta f(\mathbf{u}_{k+1}, c_{k+1}, \mu_{k+1}) + (1-\theta) f(\mathbf{u}^n, c^n, \mu^n), \quad (15)$$

$$\mu_{k+1} = \tilde{\sigma} \epsilon^{-1} W'(c_{k+1}) - \tilde{\sigma} \epsilon \Delta c_{k+1}, \quad (16)$$

- proceed to the next  $k$

3. stop the iterative process when a given tolerance is reached, e.g.  $\|c_{k+1} - c_k\| < \text{tol}$ , and set the variables at the new time step to

$$\mathbf{u}^{n+1} = \mathbf{u}_{k+1}, \quad c^{n+1} = c_{k+1}, \quad \mu^{n+1} = \mu_{k+1} \quad (17)$$

Note that in Eq. (13) only previously calculated instances of the CH variables  $c$  and  $\mu$  appear. One can interpret the Gauss–Seidel strategy as a fix point iteration where the fix point operator consists of one solve of the NS equation and one solve of the CH equation. The fix point iteration consists in applying this operator on the iterates  $(\mathbf{u}_k, c_k, \mu_k)$  multiple times until convergence is reached. Note, that convergence can be very slow due to the high nonlinearity of the operators. In fact a contraction is only assured for very small time steps and divergence may occur if the time step is chosen too large.

### 3.2. Linearization

There are three remaining nonlinear terms:  $\mathbf{u}_{k+1} \cdot \nabla \mathbf{u}_{k+1}$  in Eq. (13),  $W'(c_{k+1})$  in Eq. (16) and  $\nabla \cdot (M(c_{k+1}) \nabla \mu_{k+1})$  in Eq. (15). Using a Taylor series expansion gives the second order linear approximations:

$$W'(c_{k+1}) \approx W'(c_k) + W''(c_k)(c_{k+1} - c_k), \quad (18)$$

$$\mathbf{u}_{k+1} \cdot \nabla \mathbf{u}_{k+1} \approx \mathbf{u}_k \cdot \nabla \mathbf{u}_{k+1} + \mathbf{u}_{k+1} \cdot \nabla \mathbf{u}_k - \mathbf{u}_k \cdot \nabla \mathbf{u}_k, \quad (19)$$

$$\nabla \cdot (M(c_{k+1}) \nabla \mu_{k+1}) \approx \nabla \cdot (M(c_k) \nabla \mu_{k+1}) + \nabla \cdot (M'(c_k)(c_{k+1} - c_k) \nabla \mu_k). \quad (20)$$

If small length scales are considered, as in most applications of diffuse interface models, convection does not dominate the NS equation. Hence, a simpler linearization than Eq. (19) is sufficient. Here we skip the last two terms on the RHS of Eq. (19). Note that this does not affect the accuracy of the method and would at most slow down the convergence if the problem was dominated by convection. Similarly, the last term of Eq. (20) can be omitted since it has very little effect on the convergence speed, at least in all of our applications.

### 3.3. Special case: semi-implicit Euler

A special case of the linearized  $\theta$ -scheme is when  $\theta = 1$  and only one iteration of Eqs. (13)–(16) is performed. The resulting method can be written as the following semi-implicit Euler scheme:

$$\rho(c^n) \partial_t \mathbf{u}^{n+1} + \nabla p^{n+1} = -\rho(c^n) \mathbf{u}^n \cdot \nabla \mathbf{u}^{n+1} + \nabla \cdot (v(c^n) \mathbf{D}(\mathbf{u}^{n+1})) + \mathbf{F}(c^n) + \mu^n \nabla c^n, \quad (21)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (22)$$

$$\partial_t c^{n+1} = -\mathbf{u}^{n+1} \cdot \nabla c^{n+1} + \nabla \cdot (M(c^n) \nabla \mu^{n+1}), \quad (23)$$

$$\mu^{n+1} = \bar{\sigma} \epsilon^{-1} W'(c^{n+1}) - \bar{\sigma} \epsilon \Delta c^{n+1}, \quad (24)$$

where  $W'(c^{n+1})$  is again linearized by Eq. (18). In each time step Eqs. (21)–(22) and Eqs. (23)–(24) can be solve sequentially. This semi-implicit Euler scheme is the simplest time-stepping strategy. It is of first order accuracy but may still give good results if sufficiently small time steps are used. The scheme has been used by many authors [20,3] including a benchmark comparison of diffuse interface with level-set and VOF methods which showed good agreement [4].

### 3.4. Defect correction scheme

It is sometimes useful to apply a defect correction scheme when solving the fix-point iteration in Eqs. (13)–(16). Therefore, the solution variable is split into its previous value plus an update value (here denoted with a star):  $u_{k+1} = u_k + u^*$ . Now, the linear system is only solved for the update variable. Hence, solving a system of the form  $Au_{k+1} = b$  is replaced by solving  $Au^* = b - Au^k$ . Left and right hand side of the latter equation are of the order of the defect  $u^*$ , which minimizes errors in computer arithmetic. Another advantage of this approach is that iterative solvers sometimes perform better, i.e. need less iterations, when looking for solutions of the defect corrected scheme. To get a better approximation  $u_{k+1}$ , the update equation may be altered to  $u_{k+1} = u_k + \omega u^*$ , where the step length  $\omega$  can be found by a line search strategy. The resulting Richardson type method is described in detail in [27].

Combining a defect correction scheme with the above block Gauss–Seidel splitting and the linearizations (18)–(20) is equal to applying one step of a Newton iteration alternately to each of the two subsystems (NS and CH).

## 4. Advanced linearization techniques

We will now develop techniques to improve the coupling between the NS and the CH equations, which will turn out to lift the time step restrictions significantly. In an iterative scheme, like the block Gauss–Seidel coupling, this is equivalent to linearizing the surface tension force more efficiently. We will start by coupling the NS and CH equation really implicitly by assembling both equations in one large system.

#### 4.1. Method 1: fully-coupled scheme

The reason for the instability of the NSCH system for larger time steps or high surface tensions is the explicit coupling of the NS and CH equations. To be more precise it is the chemical potential that may oscillate (i.e. alters its sign) in every time step. Consequently, the best way to stabilize the system is by taking the chemical potential from the new time step, i.e.  $\mu_{k+1} \nabla c_k$  instead of  $\mu_k \nabla c_k$  in the NS equation, while still advecting the phase field with the new velocity  $\mathbf{u}_{k+1}$ . This can be done by assembling both, the NS and CH equations, in one large system. Hence, step 2 of the block Gauss–Seidel iteration is replaced by

2. for  $k = 0, 1, \dots$

- Solve the coupled NS–CH equation

$$\rho^{n+\frac{1}{2}} \partial_t \mathbf{u}_{k+1} + \nabla p_{k+1} = \theta \mathbf{g}(\mathbf{u}_{k+1}, c_k, \mu_{k+1}) + (1 - \theta) \mathbf{g}(\mathbf{u}^n, c^n, \mu^n), \quad (25)$$

$$\nabla \cdot \mathbf{u}_{k+1} = 0, \quad (26)$$

$$\partial_t c_{k+1} = \theta f(\mathbf{u}_{k+1}, c_k, \mu_{k+1}) + (1 - \theta) f(\mathbf{u}^n, c^n, \mu^n), \quad (27)$$

$$\mu_{k+1} = \tilde{\sigma} \epsilon^{-1} W'(c_{k+1}) - \tilde{\sigma} \epsilon \Delta c_{k+1}, \quad (28)$$

- proceed to the next  $k$

The slight difference to the block Gauss–Seidel iteration from Section 3.1 is the usage of  $c_k$  in Eq. (27) and  $\mu_{k+1}$  in Eq. (25). We will see in Section 6 that this has a strong stabilizing effect, may speed up convergence of the fix-point scheme and allows larger time steps.

We also tested other variants of assembling NS and CH in one system, for instance using the surface tension force  $\mu_k \nabla c_{k+1}$ . We found all other variants not to give any improvements in stability. The key point here is really the use of the new curvature (contained in  $\mu_{k+1}$ ). This is a big advantage of diffuse interface models compared to other two-phase flow models, because the curvature ( $\mu$ ) here is a solution variable and can therefore be implicitly coupled to the NS equation.

The same form of coupling which uses  $\mu_{k+1} \nabla c_k$  in the momentum equation and  $\mathbf{u}_{k+1} \cdot \nabla c_k$  in the CH equation, is often chosen to show energy estimates of the NS–CH system (e.g. [5]).

#### 4.2. Method 2: linearization of the chemical potential

In this section we present an alternative way which avoids solving NS and CH in one system. We will derive a stabilizing term which can easily be added to the NS equation. From the previous section we know that it is desirable to replace the surface tension force  $\theta \mu_k \nabla c_k$  occurring in Eq. (13) by the more implicit version

$$F_{st} = \theta \mu_{k+1} \nabla c_k. \quad (29)$$

Instead of using  $\mu_k$  which is a first order approximation of the new chemical potential  $\mu_{k+1}$ , we will now derive a second order approximation. The idea is to predict  $\mu_{k+1}$  from the available variables  $c_k$  and  $\mathbf{u}_{k+1}$ . To start, we use that the surface tension force  $F_{st}$  can also be written as [15]:

$$F_{st} = -\theta \tilde{\sigma} \epsilon \nabla \cdot (\nabla c_{k+1} \otimes \nabla c_k), \quad (30)$$

where the difference to Eq. (29) has the form of a gradient term and can hence be absorbed into the pressure (by redefining the pressure). Next, let us assume that the movement of  $c$  is primarily driven by advection, i.e. the influence of the chemical potential in Eq. (15) is neglected. Subtracting Eq. (15) of step  $k+1$  from the same equation in step  $k$  gives

$$c_{k+1} = c_k - \tau \theta (\mathbf{u}_{k+1} \nabla c_{k+1} - \mathbf{u}_k \nabla c_k) \quad (31)$$

$$\approx c_k - \tau \theta (\mathbf{u}_{k+1} - \mathbf{u}_k) \nabla c_k. \quad (32)$$

Now, we may insert this expression in Eq. (30) and get

$$F_{st} \approx -\theta \tilde{\sigma} \epsilon \nabla \cdot (\nabla c_k \otimes \nabla c_k) + \tau \theta^2 \tilde{\sigma} \epsilon \nabla \cdot (\nabla ((\mathbf{u}_{k+1} - \mathbf{u}_k) \nabla c_k) \otimes \nabla c_k). \quad (33)$$

The first term in Eq. (33) can be transformed back in  $\theta \mu_k \nabla c_k$  (again by redefining the pressure) which gives

$$F_{st} \approx \theta \mu_k \nabla c_k + \tau \theta^2 \tilde{\sigma} \epsilon \nabla \cdot (\nabla (\mathbf{u}_{k+1} - \mathbf{u}_k) \cdot \nabla c_k \otimes \nabla c_k) + \tau \theta^2 \tilde{\sigma} \epsilon \nabla \cdot ((\nabla \nabla c_k \cdot (\mathbf{u}_{k+1} - \mathbf{u}_k)) \otimes \nabla c_k). \quad (34)$$

The last term in Eq. (34) turned out to be very small in all our simulations and did not influence neither accuracy nor convergence speed. We therefore omit it and get the stabilizing term

$$S_1 := \tau \theta^2 \tilde{\sigma} \in \nabla \cdot (\nabla(\mathbf{u}_{k+1} - \mathbf{u}_k) \cdot \nabla c_k \otimes \nabla c_k) \tag{35}$$

which should be added to the RHS of the NS equation (13). Note, that  $S_1$  is a kind of Laplacian of the velocity field and can therefore be expected to have a stabilizing effect which will be confirmed in Section 6. Adding  $S_1$  to Eq. (13) does not affect the accuracy of the method since  $S_1$  vanishes when the fix-point iteration converges. Also note, that the derivation assumed that the phase field  $c$  is only advected, which corresponds to vanishing mobility. For larger mobilities  $S_1$  should be scaled with an (unknown) factor  $\omega \in [0, 1]$ . Here, we use  $\omega = 0.2$  which turned out to speed up the convergence of the fix-point method significantly.

### 4.3. Method 3: stabilizing surface Laplacian

The introduction of a stabilizing Laplacian of the velocity field in the previous section reminds of a very popular technique used in level-set methods first introduced by Dziuk [13]. The idea is to use that  $\kappa \mathbf{n} = -\Delta_\Gamma \text{id}_\Gamma$ , where  $\kappa = \nabla \cdot \mathbf{n}$  is the mean curvature,  $\mathbf{n}$  the normal and  $\Delta_\Gamma \text{id}_\Gamma$  the Laplace–Beltrami of the identity mapping. Hence, the implicit part of the surface tension force can be expressed as

$$F_{st} = -\sigma \theta \delta_k \Delta_{\Gamma_k} \text{id}_{\Gamma_k}, \tag{36}$$

with some surface Delta function  $\delta_k$ . To get a more implicit version of Eq. (36), one can replace  $\text{id}_{\Gamma_k}$  by  $\text{id}_{\Gamma_{k+1}}$ . The latter can be approximated by

$$\text{id}_{\Gamma_{k+1}} \approx x \rightarrow x + \theta \tau (\mathbf{u}_{k+1} - \mathbf{u}_k)(x), \tag{37}$$

which gives the surface tension force:

$$F_{st} \approx -\sigma \theta \delta_k \Delta_{\Gamma_k} \text{id}_{\Gamma_k} - \sigma \theta^2 \delta_k \tau \Delta_{\Gamma_k} (\mathbf{u}_{k+1} - \mathbf{u}_k) \tag{38}$$

$$= -\sigma \theta \delta_k \Delta_{\Gamma_k} \text{id}_{\Gamma_k} - \sigma \theta^2 \tau \nabla_{\Gamma_k} \cdot (\delta_k \nabla_{\Gamma_k} (\mathbf{u}_{k+1} - \mathbf{u}_k)). \tag{39}$$

The first term on the RHS corresponds to the surface tension force  $\theta \mu_k \nabla c_k$  which is already included in our NS equation (13). Consequently, the second term on the RHS of (39) is an additional stabilizing term which enters the NS equation. A diffuse interface version is given by

$$S_2 := -\sigma \theta^2 \tau \nabla \cdot (|\nabla c_k| P \nabla (\mathbf{u}_{k+1} - \mathbf{u}_k)) \tag{40}$$

where  $P = (I - \frac{\nabla c_k \otimes \nabla c_k}{|\nabla c_k|^2})$  is the surface projection. Analogously to the derivation in the previous section, we assumed here (in Eq. (37)) that the interface is only advected. Hence, the derivation only holds in the limit of vanishing mobility. For larger mobility we therefore scale  $S_2$  in the same way as  $S_1$  with a parameter  $\omega \in [0, 1]$ . Note, that adding  $S_2$  does not affect the accuracy of the method, since it vanishes when the fix point iteration converges (and  $\mathbf{u}_{k+1} \approx \mathbf{u}_k$ ).

## 5. Space discretization and solvers

For the numerical solution of the partial differential equations we adapt existing algorithms for the NS–CH equation, e.g. [29,12]. We use the finite element toolbox AMDiS [28] for discretization with  $P^2$  elements for  $\mathbf{u}$ ,  $c$  and  $\mu$  and  $P^1$  elements for the pressure  $p$ . In 2D we solve the resulting linear system of equations with UMFPACK [10]. For the larger systems in 3D we have to use preconditioned iterative solvers. Efficient preconditioners are available for the individual systems, that is when NS and CH are solved separately. We apply the  $F_p$  preconditioner [20] with FGMRES iteration to solve the NS system. For the CH system we use the preconditioner proposed in [8] also with FGMRES iteration.

It remains the case of the fully coupled NS–CH system, Eqs. (25)–(28), which has not been considered in the literature so far. From the discretization of Eqs. (25)–(28) we obtain a system of the form

$$\begin{bmatrix} A_{NS} & M_c \\ N_c & A_{CH} \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} = \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix}, \tag{41}$$

where  $\vec{x}$  contains the degrees of freedom of  $\mathbf{u}$  and  $p$ , while  $\vec{y}$  contains the degrees of freedom of  $c$  and  $\mu$ .  $M_c$  and  $N_c$  denote the coupling terms between the NS and the CH system. Our (simple) approach to solve this system is to combine the two preconditioners for the CH and NS systems. Let  $P_{NS}$  the  $F_p$  preconditioner for NS and  $P_{CH}$  the CH preconditioner. For the coupled system we use the matrix

$$P = \begin{bmatrix} P_{NS} & M_c \\ & \frac{1}{\alpha} P_{CH} \end{bmatrix} \tag{42}$$

as a right preconditioner, with a scaling factor  $\alpha$ . Hence, Eq. (41) is replaced by solving the two systems

$$\begin{bmatrix} A_{NS} & M_c \\ N_c & A_{CH} \end{bmatrix} \begin{bmatrix} P_{NS}^{-1} & -\alpha P_{NS}^{-1} M_c P_{CH}^{-1} \\ 0 & \alpha P_{CH}^{-1} \end{bmatrix} \begin{bmatrix} \vec{v} \\ \vec{w} \end{bmatrix} = \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix} \tag{43}$$

**Table 1**  
Parameters used for stability estimations.

$\sigma$	$10^2$	$10^3$	$10^4$	$10^5$
$M$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-7}$
$\epsilon$	0.08	0.04	0.02	0.01
$h$	$2\epsilon$	$\epsilon$	$0.5\epsilon$	
$\rho$	$10^{-1}$	$10^0$	$10^1$	$10^2$

and

$$\begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} = \begin{bmatrix} P_{\text{NS}}^{-1} & -\alpha P_{\text{NS}}^{-1} M_c P_{\text{CH}}^{-1} \\ \mathbf{0} & \alpha P_{\text{CH}}^{-1} \end{bmatrix} \begin{bmatrix} \vec{v} \\ \vec{w} \end{bmatrix}. \quad (44)$$

A rigorous analysis of the matrix properties is still missing. However, in our numerical tests, this preconditioned system can be solved by an FGMRES iteration with  $\alpha = 0.1$ .

## 6. Numerical tests

We now validate the proposed linearization schemes on different test scenarios. First, we assess the numerical stability of the time integration schemes (Section 6.1). Then we conduct a benchmark comparison of the different linearization methods (Section 6.2). At last we present an application to a Taylor-Flow simulation which clearly shows the superior performance of the proposed methods. Throughout this section we use an iteration tolerance of  $\text{tol} = 10^{-10}$  (see Section 3.1). Furthermore, we terminate the block Gauss-Seidel scheme if no convergence is reached after 100 iterations. As stabilization constant we use  $\omega = 0.2$  (see Section 4).

### 6.1. Stability investigations

In this section, we assess the time step stability of the proposed schemes. Our goal is to find the maximum time step size at which a given two-phase flow configuration is solvable. We use configurations with different surface tensions  $\sigma$ , mobilities  $M$ , grid sizes  $h$  and interface thicknesses  $\epsilon$ . In the end we want to find an estimate to predict the maximum time step size from these parameters. As mentioned in Section 1 for other two-phase flow methods (e.g. Level-Set, ALE) the estimate gives the CFL-like condition

$$\tau_{\text{max}} < C \rho^{\frac{1}{2}} h^{\frac{3}{2}} \sigma^{-\frac{1}{2}} \quad (45)$$

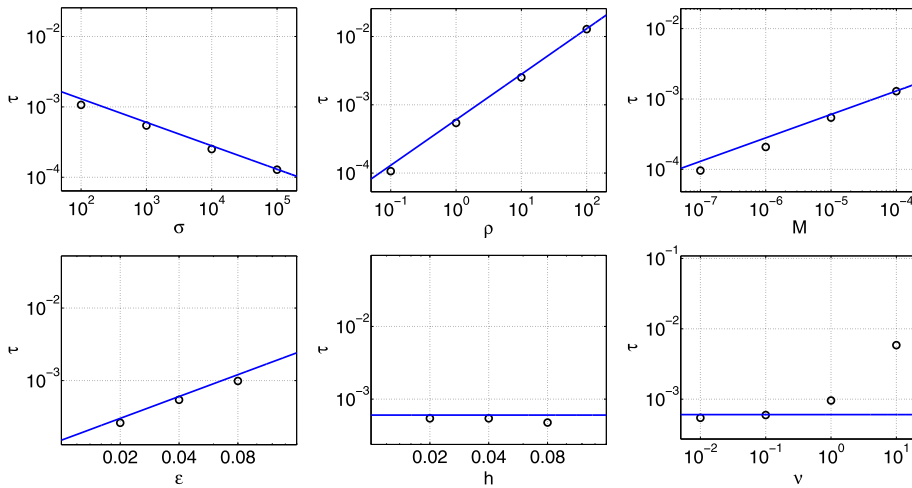
with some non-dimensional constant  $C$ . This restriction can be derived from the consideration of capillary waves [9] in the limit of small viscosity. To resolve such waves the time step size should suffice  $\tau_{\text{max}} < h/2V$ , with  $V = \sqrt{\sigma k/2\rho}$  being the capillary wave phase velocity. Here  $k$  is the wavenumber which can for sharp interface models become as big as  $\pi/h$ . For diffuse interface models this derivation fails since, the smallest wavenumber should depend on  $\epsilon$  rather than  $h$ . Furthermore the CH diffusion smooths the phase field function which does not allow the simple ansatz  $\tau_{\text{max}} < h/2V$ . However, a common assumption is that the time step restriction (45) also holds for diffuse interface methods which we will contradict in the following.

We use numerical testing to assess the time step stability, since analytical investigations on the coupled NS-CH system are extremely complicated. We restrict the numerical tests to the Crank-Nicolson scheme (Eqs. (12)–(17) with  $\theta = 0.5$ ).<sup>1</sup> As initial condition we use  $\mathbf{u} = 0$  and a phase field given by  $c = \tanh((y - 0.5)/\sqrt{2}\epsilon)$  in the domain  $\Omega = [0, 1]^2$  which represents a flat horizontal interface. Since the curvature is zero this initial condition corresponds to a stationary state. To trigger an instability, a different random number  $\in [-0.001, 0.001]$  is added to each grid point of the phase field. Hence, the fix point iteration will not converge for too large time steps. We then try to solve a single time step of the system with these initial conditions. We do this multiple times for varied time step sizes. We start with very small time steps which assure convergence. As long as the system can be solved we increase the time step size (by a factor of 1.1) and start again. At some point the system will not be solvable and we denote the corresponding time step as the maximum time step size  $\tau_{\text{max}}$  for this configuration.

We repeat this procedure for various numerical parameters shown in Table 1. We use equal densities in both phases and choose  $\nu = 0.01$  to make sure that viscosity does not play a dominant role.<sup>2</sup> Varying all other parameters ( $\sigma, \rho, \epsilon, h, M$ ) independently gives a total number of  $4 \cdot 4 \cdot 4 \cdot 3 \cdot 4 = 768$  test cases and we denote the corresponding parameters for some test case  $i \in \{1, \dots, 768\}$  by subscripts:  $\sigma_i, \rho_i, \epsilon_i, h_i, M_i$ . We use the explicit scheme (Eqs. (12)–(17) with  $\theta = 0.5$ ) and monitor for each test case the maximally possible time step  $\tau_{\text{max},i}$ . Our goal is to obtain a relationship between the test parameters of case  $i$  and  $\tau_{\text{max},i}$ . Assuming a multiplicative relationship with unknown exponents gives the nonlinear least-squares problem:

<sup>1</sup> For  $\theta = 1$ , the obtained maximum time step would be need to be divided by 2.

<sup>2</sup> Analogously to the derivation of (45), see e.g. [9].



**Fig. 2.** Log-log plots comparing the maximum time step size with the CFL-condition (48). Either one of the variables  $\rho$ ,  $\sigma$ ,  $\epsilon$ ,  $M$ ,  $h$ ,  $\nu$  is varied while keeping the other variables fixed. Results are obtained with a Crank–Nicolson scheme (Eqs. (12)–(17) with  $\theta = 0.5$ ).

$$\min_{\alpha} \sum_{i=1}^{768} \left| \log \left( \frac{\alpha_1 h_i^{\alpha_2} \epsilon_i^{\alpha_3} \sigma_i^{\alpha_4} M_i^{\alpha_5} \rho_i^{\alpha_6}}{\tau_{\max,i}} \right) \right|^2 \tag{46}$$

where  $\alpha$  contains the searched unknowns,  $\alpha = (\alpha_1, \dots, \alpha_6)$ . We use the routine `LSURVEFIT` in MatLab to solve (46) and obtain  $\alpha = (7.603, 0.001, 0.916, -0.326, 0.374, 0.673)$  which is equivalent to the time step restriction

$$\tau_{\max} < 7.603 h^{0.001} \epsilon^{0.916} \sigma^{-0.326} M^{0.374} \rho^{0.673}. \tag{47}$$

The calculation of this time step restriction involved some rounding, in particular the maximum time steps may be over estimated by up to 10%, since we increased them stepwise by 10%. These errors limit the precision of (47) and justifies to round the obtained values. In this way, we get the following CFL-like time step restriction

$$\tau_{\max} < 7.0 \epsilon \rho^{2/3} M^{1/3} \sigma^{-1/3}. \tag{48}$$

In strong contrast to (45) this time step restriction is independent of  $h$ . It is hard to come up with a theoretical justification of this expression, due to the complicated interplay of diffusion and advection in the NS–CH model. However, the smallest capillary wavelength in a NS–CH model can be expected to be proportional to  $\epsilon$  which explains the occurrence of  $\epsilon$  on the RHS. Also, it seems reasonable that the CH mobility  $M$  enters the RHS with a positive exponent since higher  $M$  results in additional smoothing. A further justification is given by looking at the physical units of Eq. (48). Using that  $M$  is measured in  $m^3 \text{ s/kg}$  we obtain that both sides of (48) are measured in s (seconds).

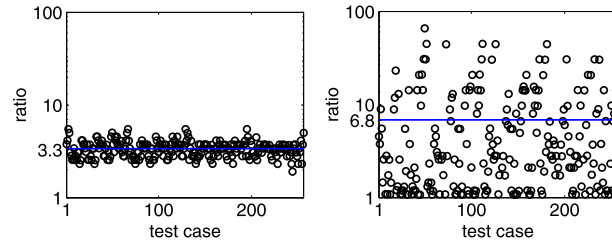
Fig. 2 shows the experimental maximum time step size compared to the CFL-condition (48). Thereby, we vary one of the six variables ( $\rho, \sigma, \epsilon, M, h, \nu$ ) while keeping the other variables fixed at  $\rho = 1.0$ ,  $\sigma = 10^3$ ,  $M = 10^{-5}$ ,  $\epsilon = 0.04$ ,  $h = 0.04$ ,  $\nu = 0.01$ . One can see an excellent agreement of the numerical data with the time step restriction curve. But also the limit of the derived CFL condition becomes apparent by looking at the case of varied viscosity  $\nu$  in Fig. 2. According to our derivation, the time step restriction only holds in the limit of small viscosities (here  $\text{Re} \lesssim 1$ ). Note, that this coincides with the famous CFL condition (45) which was also derived for the small viscosity case [9].

We also assess the time step stability of the advanced linearization schemes proposed in Section 4. In particular the fully coupled scheme (Eqs. (25)–(28)) shows a superior performance. For all test cases the scheme converges in at most 10 iterations independently of the time step. Hence, for a configuration close to the stationary state the fully coupled scheme allows arbitrary large time steps. This outstanding property will be further exploited in Section 6.3.

But also the two schemes employing the stabilizing terms  $S_1$  and  $S_2$  (Eqs. (35) and (40), resp.) perform very well. We divide the maximum time step of the stabilized schemes by the maximum time step of the simple explicit scheme. Fig. 3 shows this ratio for the 256 test cases with  $h = 2\epsilon$ .

Using  $S_1$  increases the maximum time step by a factor of 1.9–5.5 (average: 3.3). Much more diverse ratios are obtained when using  $S_2$ , which increases the maximum time step by a factor between 1.0 and 65.9 (average: 6.8). Note, that the stabilizing effect of  $S_1$  and  $S_2$  depends on the factor  $\omega$ . For the ease of comparison, we set  $\omega = 0.2$  here, but adjusting it manually to the used mobility would allow even higher time steps.





**Fig. 3.** The ratio of maximum time steps (stabilized scheme S1 (left) and S2 (right) divided by explicit scheme) for the different test cases. The straight line shows the average ratio. The vertical axis uses log scaling.

**Table 2**

Bubble position, errors and rate of convergence (ROC) for different time step sizes  $\tau$ . The results confirm first order convergence for  $\theta = 1.0$  (a) and second order convergence for  $\theta = 0.5$  (b).

$\tau$	position	error	ROC
0.100	0.51280317	3.69E-03	
0.040	0.51064693	1.53E-03	0.96
0.020	0.50989193	7.74E-04	0.98
0.010	0.50950797	3.90E-04	0.99
0.005	0.50931372	1.96E-04	0.99

(a)  $\theta = 1.0$

$\tau$	position	error	ROC
0.100	0.50904344	7.45E-05	
0.040	0.50910657	1.14E-05	2.05
0.020	0.50911533	2.66E-06	2.10
0.010	0.50911762	3.70E-07	2.84
0.005	0.50911799	0.00E-00	–

(b)  $\theta = 0.5$

## 6.2. Benchmark problem

We use the test setup from the two-phase flow benchmark of Hysing et al. [18]. It considers a single bubble rising in a liquid column for a period of 3.0 time units. The benchmark scenario has also been studied with a diffuse interface model [4].

We start with the same configuration as test case 1 in [4] with  $\epsilon = 0.02$ ,  $M = 2 \cdot 10^{-5}$ ,  $\rho_1 = 1000$ ,  $\rho_2 = 100$ ,  $\nu_1 = 10$ ,  $\nu_2 = 1$ ,  $\sigma = 24.5$ . First, let us confirm the accuracy of the  $\theta$ -scheme. To make the comparison computationally cheaper, we restrict the following studies to the time interval  $[0, 0.2]$ . We solve the system with different time step sizes for  $\theta = 1.0$  and  $\theta = 0.5$ . Table 2 shows the final bubble position for all of these cases. We assume that the smallest time step together with  $\theta = 0.5$  gives the most accurate result and take this as a reference value to compute the errors of the other test cases. The order of convergence (ROC) shows clearly first order convergence for  $\theta = 1.0$  and second order convergence for  $\theta = 0.5$  (Table 2).

So far, it did not matter whether we use the standard block Gauss–Seidel coupling, the fully coupled NS–CH system, or any of the introduced stabilizing terms  $S_1$ ,  $S_2$ . All of these methods give the same computational result as long as their inner fix-point iteration converges. However, the number of needed iterations and the time for each iteration may vary among the methods. Therefore, we will next analyze the performance of the four different solution methods:

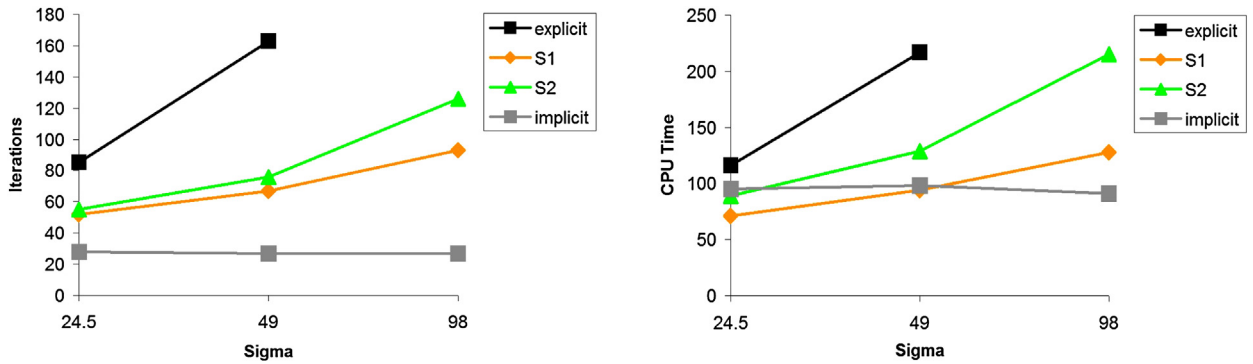
- *explicit*: standard block Gauss–Seidel coupling (see Sections 3.1–3.2)
- *S1*: with additional stabilization term  $S_1$  from Eq. (35)
- *S2*: with additional stabilization term  $S_2$  from Eq. (40)
- *implicit*: fully coupled NS–CH system (Eqs. (25)–(28))

We use the same configuration as before, with  $\epsilon = 0.005$ ,  $M = 5 \cdot 10^{-6}$ . We will use a second order Crank–Nicolson time stepping ( $\theta = 0.5$ ) with  $\tau = 0.02$  which gives comparable time discretization errors as in the original benchmark paper [4].

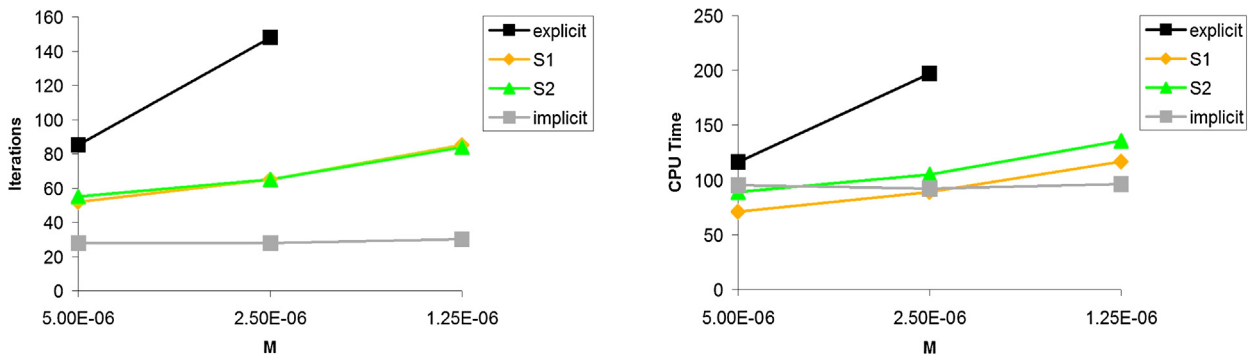
Table 3 shows the total number of iterations and CPU time in seconds for the different simulations. One can see that the fully coupled system needs by far the least iterations followed by the two stabilized schemes *S1*, *S2*. The number of iterations for the *explicit* method is more than three times higher than for the *implicit* system. However, these differences in iterations are not directly reflected in the CPU timings. Apparently one solution of the *implicit* system is almost three times as expensive as solving the *explicit* system, reducing the advantage of the *implicit* system significantly.

**Table 3**  
Performance of different solution methods.

	explicit	S1	S2	implicit
Iterations	85	52	55	28
CPU time (s)	116	71	89	95



**Fig. 4.** Number of iterations and CPU times for different surface tensions  $\sigma$ . While the explicit scheme does not even converge for large surface tensions, the implicit scheme seems unaffected.

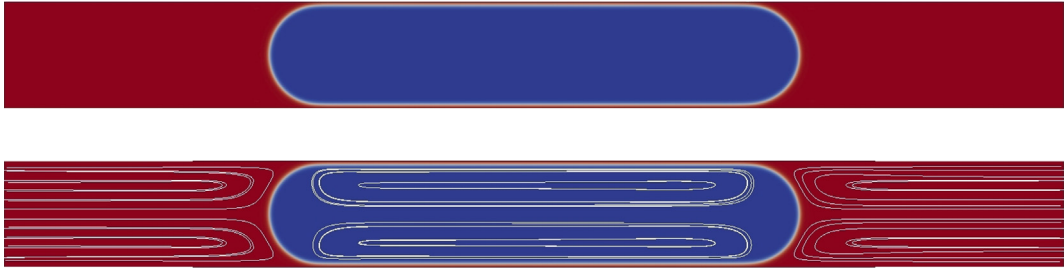


**Fig. 5.** Number of iterations and CPU times for different mobilities  $M$ . While the explicit scheme does not even converge for low mobility, the implicit scheme seems unaffected.

Also the stabilized systems  $S1$ ,  $S2$  reduce the number of iterations compared to the *explicit* scheme, while the resulting linear systems can be solved as fast as the *explicit* system. Consequently, the methods  $S1$  and  $S2$  perform best in terms of CPU time. Method  $S1$  is the fastest.

In a next comparison we include higher surface tensions  $\sigma$  and lower mobilities  $M$ , since we expect our improved methods to be particularly fast in these cases. Fig. 4 shows the number of iterations and CPU times for surface tensions increased by a factor of 2 and 4. The number of iterations and CPU time increases for the *explicit* method with increasing  $\sigma$ . For  $\sigma = 98.0$  the fix point iteration does not converge anymore. Also the stabilized methods  $S1$ ,  $S2$  become slower for increased  $\sigma$  but they still converge. Remarkably, the *implicit* method is not affected by the increase in  $\sigma$ , neither the number of iterations nor the CPU time changes significantly. A very similar picture can be seen when the mobility is varied. In Fig. 5 the mobility  $M$  is decreased by a factor of 2 and 4. Again the *explicit* method gets very slow and does not even converge for the smallest  $M$ , whereas the *implicit* method remains almost unaffected.

**A note on spurious velocities:** Spurious currents, i.e. artificial velocity fields that persist in an equilibrium situation of a circular droplet are an ongoing issue in simulations of two-phase flows. We have tested our schemes for such currents by disabling the gravity force in the benchmark scenario. We found spurious currents to occur here as well, with a velocity magnitude independent of the chosen scheme. This is not surprising since these currents are a steady state phenomenon, and should not be affected by the time stepping scheme. However, for the *explicit* scheme such currents may cause problems, since they prevent the use of large time steps even if the system is close to the equilibrium. The schemes  $S1$  and  $S2$  allow larger time steps and the *implicit* scheme lifts this time step restriction completely and allows the use of arbitrary large time steps if the system is in equilibrium. We will exploit this property more in detail in the following application to Taylor flow.



**Fig. 6.** Initial shape (top) and final shape (bottom) with flow streamlines of the Taylor bubble. The bubble shape only changed slightly during the time evolution due to high surface tension pressing the bubble against the channel wall.

### 6.3. Application to Taylor-flow

We consider a Taylor flow simulation to further demonstrate the efficiency of the proposed linearization techniques. Taylor flow is the flow of a single elongated bubble through a narrow channel. In narrow channels, the bubble assumes a quasi-stationary state, i.e. a fixed shape which is only advected in the direction of the channel. Many technical applications involve Taylor flow, e.g. catalytic converters [30], monolith reactors [23], or microfluidic channels [25]. In these applications, bubbles of identical size, shape, and distance to each other are typically required. Thus, the hydrodynamics that lead to a perfect quasi-stationary bubble are of interest in these research areas. This makes Taylor flow also interesting as a general benchmark for two phase flow methods and indeed such a benchmark has been recently defined in 2D [2] as well as in 3D [24].

#### 6.3.1. 2D Taylor-flow

As in the above-mentioned benchmarks, we want to compute the quasi-stationary state of a Taylor bubble driven through the channel by a pressure difference between inlet and outlet. An implicit Euler scheme is expected to converge fast to the stationary solution due to its high numerical dissipativity. Therefore, we use the scheme proposed in Section 3.3 (i.e.  $\theta = 1$  and only one sub-iteration) in four different versions:

- *explicit*: no stabilization (exactly as in Section 3.3)
- *S1*: with additional stabilization term  $S_1$  from Eq. (35)
- *S2*: with additional stabilization term  $S_2$  from Eq. (40)
- *implicit*: fully coupled NS-CH system (Eqs. (25)–(28) also with  $\theta = 1$  and only one sub-iteration)

As in [2] we use a moving frame of reference. Therefore we calculate the bubble velocity  $\mathbf{u}_b$  by  $\mathbf{u}_b = (\int_{\Omega} (1-c)\mathbf{u}) / (\int_{\Omega} 1-c)$  and replace the velocity of the convective terms in Eqs. (21) and (23) by  $(\mathbf{u} - \mathbf{u}_b)$ . Hence the frame of reference moves with the bubble and the quasi-stationary state becomes really-stationary. Similar to [2] we use the parameters  $\nu_1 = \nu_2 = 10.0$ ,  $\rho_1 = \rho_2 = 1.0$ ,  $\sigma = 5000$ ,  $M = 3.0 \cdot 10^{-6}$ ,  $\epsilon = 0.015$ ,  $\omega = 1.0$ .

To demonstrate the efficiency of the proposed linearization schemes, we use a simple adaptive time stepping. Our goal is to control the CFL number which can be accomplished by choosing something like  $\tau = 0.5h / \max(|\mathbf{u} - \mathbf{u}_b|)$ . Since the usage of  $\mathbf{u} - \mathbf{u}_b$  would not include movement due to CH diffusion we replace this term by the phase field velocity  $\partial_t c / |\nabla c|$  and end up with the following time step selection:

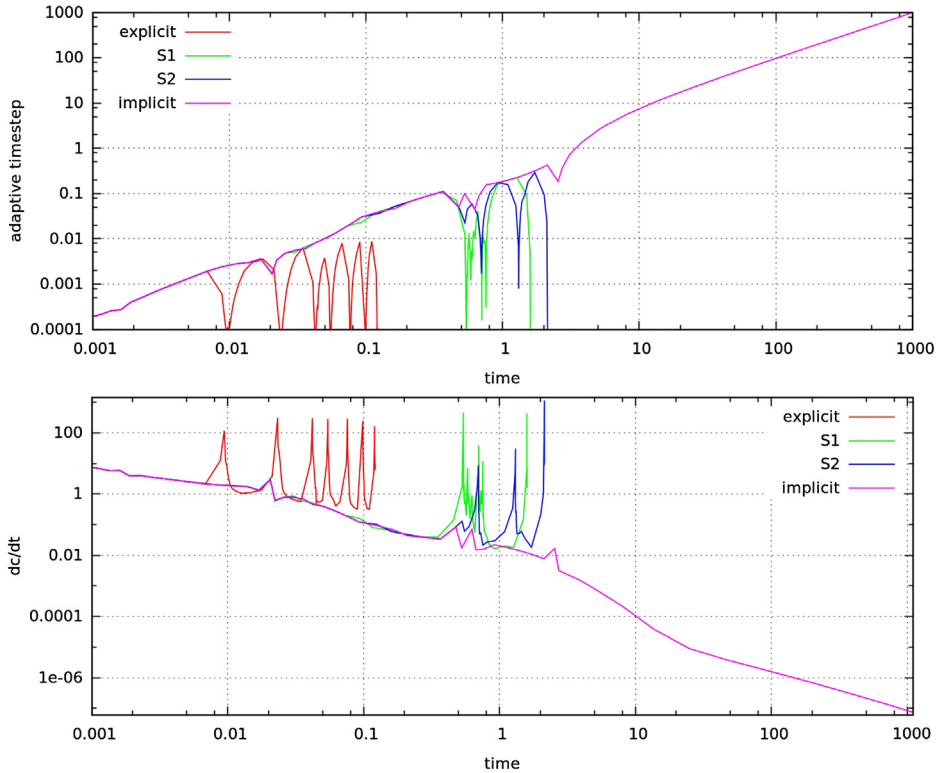
$$\tau^n = \frac{1}{2} \tau^{n-1} \cdot \max \frac{h |\nabla c^n|}{|c^n - c^{n-1}|}, \quad (49)$$

where we choose the first time step,  $\tau^0 = 10^{-4}$ .

The initial condition for the bubble is as in [2]: A rod-like bubble of total length 5 is placed in the middle of a channel  $\Omega = [0, 10] \times [0, 1]$ , see Fig. 6. The pressure difference between channel inflow and outflow is iteratively adjusted to give a bubble velocity  $\mathbf{u}_b = 1$ . Though it is hard to see differences between the initial and final bubble shape with the naked eye, there is some significant bubble deformation going on. In particular, the width of the thin liquid film between bubble and wall changes during the time evolution.

Fig. 7 shows the adaptive time steps (top) and the change in  $c$  (bottom) for the four methods. Let us first focus on the *explicit* scheme. Although the bubble is close to its stationary shape, the *explicit* scheme becomes unstable for large time steps. Once  $\tau$  is larger than approximately  $10^{-3}$  the interface starts to oscillate ( $\|\partial_t c\|_{L^1}$  increases), which results in a rapid decrease of the time steps. After some calculations with small time steps the bubble stabilizes and the time steps increase and the whole process starts over again. This results in an up and down of time steps which renders the explicit scheme almost useless. To reach the stationary state one would have to limit  $\tau$  from above to approximately  $10^{-3}$ , which would require a hundred thousand of total time steps to reach a sufficiently stationary state (say at  $t = 100$ ).

A similar instability occurs for the schemes *S1* and *S2* but at much larger time steps,  $\tau \geq 10^{-1}$ . Hence adding either of the terms  $S_1$  and  $S_2$  allows to increase the time steps almost by two orders of magnitude. To reach the stationary state one



**Fig. 7.** Comparison the Taylor-flow simulation with the four methods. **Top:** The adaptively chosen time step over time. **Bottom:** The change in the phase field  $\|\partial_t c\|_{L^1}$  over time. The stationary state is reached when  $\|\partial_t c\|_{L^1}$  is sufficiently small. The explicit scheme becomes unstable already for time steps larger than  $10^{-3}$ . Only the implicit scheme is stable for arbitrarily large time steps and approaches the stationary state.

would have to limit  $\tau$  from above to approximately  $10^{-1}$ , which would require about 1000 total time steps to reach the stationary state. Note that the computational costs per time step are almost equal for the schemes *explicit*, *S1* and *S2*: 3.15 s, 3.41 s and 3.56 s, respectively. Very impressive is the performance of the *implicit* scheme which is stable for arbitrarily large time steps. Thus, while the bubble approaches the stationary state, larger and larger time steps are possible. Hence, we obtain the stationary state in just 64 total time steps. The average cost per time step is 8.52 s which gives a computation time of approximately 9 min for the complete simulation. Hence the *implicit* scheme outperforms the other schemes by far.

### 6.3.2. 3D Taylor-flow

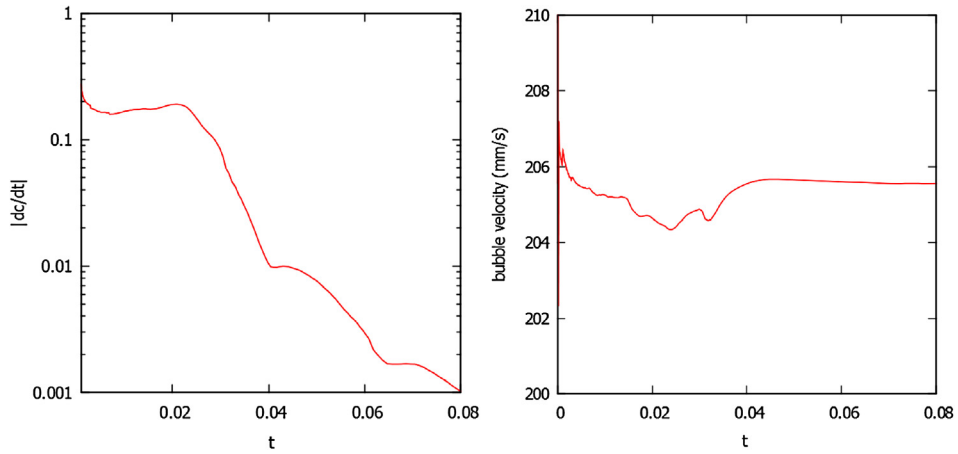
Encouraged by the good results of the fully coupled time discretization in 2D, we venture to try a 3D Taylor flow example. We use the test setup from the 3D Taylor flow benchmark in [24]. Thereby a gas bubble of volume  $17.5 \text{ mm}^3$  is placed in a domain of size  $1.98 \text{ mm} \times 11.88 \text{ mm} \times 1.98 \text{ mm}$ . Exploiting the symmetry in  $x$ - and  $z$ -direction, we restrict the calculations to a quarter of this domain. Further parameters are in the liquid phase:  $\rho_L = 1195.6 \text{ kg/m}^3$ ,  $\nu_L = 28.54 \cdot 10^{-3} \text{ kg/ms}$ , in the gas bubble:  $\rho_G = 11.95 \text{ kg/m}^3$ ,  $\nu_L = 28.54 \cdot 10^{-5} \text{ kg/ms}$ , the surface tension is  $\sigma = 66.69 \cdot 10^{-3} \text{ kg/s}^2$  and the desired final bubble velocity is  $205.57 \text{ mm/s}$ . For the diffuse interface model we use  $M = 0.9 \cdot 10^{-9} \text{ m}^3\text{s/kg}$  and  $\epsilon = 0.03 \text{ mm}$ .

Again, we are looking for a stationary state. Hence, we use the fully coupled NS-CH system (Eqs. (25)–(28)) with  $\theta = 1$  and only one sub-iteration. Starting with  $\tau^0 = 2.0 \cdot 10^{-5} \text{ s}$ , we double the time step size every 50 time steps. A grid size of  $h = 0.087 \text{ mm}$  at the interface gives around 1 million total degrees of freedom. We use an MPI based parallelization with 44 cores and the preconditioned FGMRES iteration described in Section 5. One time step is solved in approximately 1 minute. Fig. 8 shows the time evolution of  $\|\partial_t c\|_{L^1}$  and the bubble velocity  $\mathbf{u}_b$ . The change in  $c$  decays exponentially. We assume that we are sufficiently close to the stationary solution if  $\|\partial_t c\|_{L^1} < 10^{-3}$ .

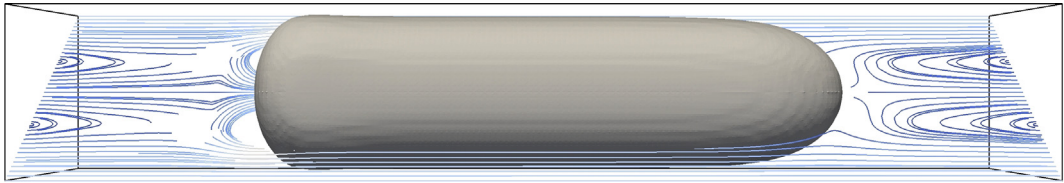
The final bubble is depicted in Fig. 9 together with the streamlines showing a recirculating flow pattern at the front and the rear of the bubble. To compare the bubble shape with the reference solution from [24], we cut the phase field vertically through the middle of the domain. The zero level set of the phase field along this slice is compared to the reference solution from DROPS [24] in Fig. 10 and shows a quite satisfactory agreement.

## 7. Conclusions

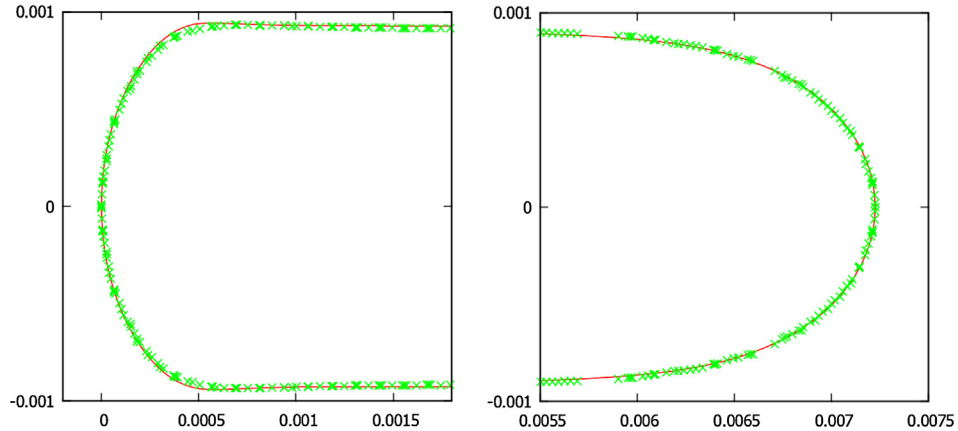
In this paper, we addressed time integration strategies for the diffuse interface model for two-phase flows. We proposed a variant of the  $\theta$ -scheme together with three new linearization techniques for the surface tension. These involve either additional stabilizing terms ( $S_1$  or  $S_2$ ), or a fully implicit coupling of the NS and CH equation.



**Fig. 8.** Time evolution of  $\|\partial_t c\|_{L^1}$  (left) and bubble velocity  $u_b$  (right). The change in  $c$  decays exponentially (left).



**Fig. 9.** Quasi-stationary state of the Taylor bubble moving from the left to the right. The streamlines show a recirculating flow pattern at the front and the rear of the bubble.



**Fig. 10.** Final bubble shape at the rear (left) and the front (right) of the bubble. The line shows the reference solution from DROPS [24]. The crosses mark our diffuse interface solution. Both solutions are aligned at the ends.

As in all two phase flow methods the coupling between the flow and the interface equation plays a crucial role and limits the stability and the range of applicable time steps significantly. This is particularly true if high surface tensions or small length scales are considered. In the common case that interface and flow equation are coupled explicitly, we could show a time step restriction of the form

$$\tau < C\epsilon\sigma^{-1/3}M^{1/3}\rho^{2/3} \quad (50)$$

which is very different to other two-phase flow models and in particular is independent of the grid size. As in other two-phase flow models, this restriction can make computations extremely costly. Even in cases when the interface is almost stationary, too large time steps will lead to oscillations and finally destruction of the interface.

We also showed that the proposed stabilization techniques could lift the above time step restriction. The simple stabilizing terms  $S_1$  and  $S_2$  may allow an increase in time step size of about two orders of magnitude. If a fix point or Newton sub-iteration is used in each time step, these terms can reduce the number of iterations significantly, while not affecting the accuracy.

Very impressive is the performance of the fully implicit scheme which is stable for arbitrarily large time steps. We demonstrate in a Taylor flow application that this superior coupling between flow and interface equation can decrease computation time of diffuse interface models by several orders of magnitude.

Apart from increasing the computational performance, the improved time integration schemes allow to choose lower CH mobility, which may come closer to physically correct values. Hence, the mean curvature flow included in the CH equation is suppressed which may lead to more accurate computational results.

## Acknowledgements

We acknowledge support from the German Science Foundation through grant SPP-1506 (AL 1705/1-1) and support of computing time at JSC at FZ Jülich.

## References

- [1] Helmut Abels, Harald Garcke, Günther Grün, Thermodynamically consistent, frame indifferent diffuse interface models for incompressible two-phase flows with different densities, *Math. Models Methods Appl. Sci.* 22 (03) (2012).
- [2] S. Aland, A. Hahn, F. Klingbeil, M. Weismann, S. Weller, Quantitative comparison of Taylor flow simulations based on sharp- and diffuse-interface models, *Int. J. Numer. Methods Fluids* 73 (2013) 344–361.
- [3] S. Aland, J. Lowengrub, A. Voigt, A continuum model of colloid-stabilized interfaces, *Phys. Fluids* 23 (6) (2011) 062103.
- [4] S. Aland, A. Voigt, Benchmark computations of diffuse interface models for two-dimensional bubble dynamics, *Int. J. Numer. Methods Fluids* 69 (3) (2012) 747–761.
- [5] Sebastian Aland, Chen Feng, An efficient and energy stable scheme for a phase-field model for the moving contact line problem, *SIAM Comput. Methods Sci. Eng.* (2014), submitted for publication.
- [6] D.M. Anderson, G.B. McFadden, A.A. Wheeler, Diffuse interface methods in fluid mechanics, *Annu. Rev. Fluid Mech.* 30 (1998) 139–165.
- [7] V.E. Badalassi, H.D. Ceniceros, Sanjoy Banerjee, Computation of multiphase systems with phase field models, *J. Comput. Phys.* 190 (2) (2003) 371–397.
- [8] Petia Boyanova, Minh Do-Quang, Maya Neytcheva, Efficient preconditioners for large scale binary Cahn–Hilliard models, *Comput. Methods Appl. Math.* 12 (1) (2012) 1–22.
- [9] J.U. Brackbill, Douglas B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (2) (1992) 335–354.
- [10] Timothy A. Davis, Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method, *ACM Trans. Math. Softw.* 30 (2) (June 2004) 196–199.
- [11] H. Ding, P.D.M. Spelt, C. Shu, Diffuse interface model for incompressible two-phase flows with large density ratios, *J. Comput. Phys.* (2007) 2078–2095.
- [12] Minh Do-Quang, Gustav Amberg, The splash of a solid sphere impacting on a liquid surface: Numerical simulation of the influence of wetting, *Phys. Fluids* 21 (2009).
- [13] Gerhard Dziuk, An algorithm for evolutionary surfaces, *Numer. Math.* 58 (1) (1990) 603–611.
- [14] H. Emmerich, Advances of and by phase-field modeling in condensed-matter physics, *Adv. Phys.* 57 (2008) 1–87.
- [15] X. Feng, Fully discrete finite element approximations of the Navier–Stokes–Cahn–Hilliard diffuse interface model for two-phase fluid flows, *SIAM J. Numer. Anal.* 44 (2006) 1049–1072.
- [16] Günther Grün, Fabian Klingbeil, Two-phase flow with mass density contrast: stable schemes for a thermodynamic consistent and frame-indifferent diffuse–interface model, arXiv:1210.5088, 2012.
- [17] S. Hysing, A new implicit surface tension implementation for interfacial flows, *Int. J. Numer. Methods Fluids* 51 (6) (2006) 659–672.
- [18] S. Hysing, S. Turek, D. Kuzmin, N. Parlini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, *Int. J. Numer. Methods Fluids* 60 (2009) 1259–1288.
- [19] D. Jaqmin, Calculation of two-phase Navier–Stokes flows using phase-field modelling, *J. Comput. Phys.* 155 (1999) 96–127.
- [20] D. Kay, R. Welford, Efficient numerical solution of Cahn–Hilliard–Navier–Stokes fluids in 2D, *SIAM J. Sci. Comput.* 29 (2007) 2241–2257.
- [21] J. Kim, J. Lowengrub, Phase field modeling and simulation of three-phase flows, *Interfaces Free Bound.* 7 (2005) 435–466.
- [22] Junseok Kim, A continuous surface tension force formulation for diffuse–interface models, *J. Comput. Phys.* 204 (2) (2005) 784–804.
- [23] Michiel T. Kreutzer, Freek Kapteijn, Jacob A. Moulijn, Johan J. Heiszwolf, Multiphase monolith reactors: chemical reaction engineering of segmented flow in microchannels, *Chem. Eng. Sci.* 60 (22) (2005) 5895–5916.
- [24] H. Marschall, S. Boden, C. Lehrenfeld, C. Falconi, U. Hampel, A. Reusken, M. Wörner, D. Bothe, Validation of interface capturing and tracking techniques with different surface tension treatments against a Taylor bubble benchmark problem, *Comput. Fluids* (2014), submitted for publication.
- [25] M. Muradoglu, A. Günther, H.A. Stone, A computational study of axial dispersion in segmented gas–liquid flow, *Phys. Fluids* 19 (7) (2007).
- [26] I. Singer-Loginova, H. Singer, The phase field technique for modeling multiphase materials, *Rep. Prog. Phys.* 71 (2008) 106501.
- [27] Stefan Turek, Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach, vol. 6, Springer Verlag, 1999.
- [28] S. Vey, A. Voigt, Amdis: adaptive multidimensional simulations, *Comput. Vis. Sci.* 10 (1) (March 2007) 57–67.
- [29] W. Villanueva, G. Amberg, Some generic capillary-driven flows, *Int. J. Multiph. Flow* 32 (9) (September 2006) 1072–1086.
- [30] J.L. Williams, Cheminform abstract: monolith structures, materials, properties and uses, *ChemInform* 33 (11) (2002).