A subspace projection method for the implementation of interface conditions in a two-phase flow problem

von

K. Bäumler & E. Bänsch

No. 355 2012
A subspace projection method for the implementation of interface conditions in a two–phase flow problem

Kathrin Bäumler*a,†, Eberhard Bänsch*a

aDepartment of Mathematics, Friedrich-Alexander-University Erlangen-Nuremberg, Haberstr. 2, 91058 Erlangen, Germany

Abstract

We present a mesh moving method within a finite element context, where the interface conditions of the two–phase flow problem are conveniently included in suitable chosen subspaces of the general trial and testfunction spaces. A weak formulation of the two–phase flow problem including species transport is derived, where the problem specific function spaces are replaced by appropriate projections operating on standard function spaces. The transfer of the variational formulation to a finite element method is straightforward and results in one set of equations for both fluidic phases. This subspace projection method is applicable to a wide range of interfacial conditions for multiphase flow problems. The method is validated for single drop flow problems including species transfer. Furthermore, an application to the simulation of stagnant caps is presented.

Keywords: Finite elements, two–phase flow, species transport, stagnant cap

1. Introduction

Flow problems including a sharp interface arise in many physical problems. Let us mention slug flow, Taylor bubbles, and dispersed liquid–liquid systems just to name a few examples. Over the last years the interest in two–phase flow systems has extended to the investigation of systems with species transport from one phase to another and to systems with impurities, occurring in the form of surfactants.

Single drop flow being a small scale detail of dispersed systems is a complex flow problem, especially when deformation of the interface, species transport or the occurrence of surfactants are considered for direct numerical simulations. Single drop flow is of practical interest in engineering applications, where large scale closures of single drop flow simulations may be used to predict exposure times and mass transfer rates. Due to the practical interest in understanding single drop flow, experimental data is available for various test systems. The availability of experimental results in terms of terminal velocities, overall mass transfer and

*Corresponding author

Email addresses: baeumler@am.uni-erlangen.de (Kathrin Bäumler ), baensch@am.uni-erlangen.de (Eberhard Bänsch)

Preprint submitted to Elsevier August 30, 2012
shape deformation permits a validation of computational results. Once a numerical method has been proven to provide reliable simulations for these systems, it can be used as a powerful tool for prognostic or diagnostic purposes for new physical systems.

In the last years the interest in the simulation of systems with impurities has increased. Impurities usually occur in form of surfactants, and are of importance in industrial two-phase flows like liquid–liquid extraction processes because of their ubiquity in any practical situation. Various models for the simulation of surfactants are available (for an overview we refer to [1]). Many of them include an advection diffusion equation for the surfactant, which – if complicated adsorption dynamics are taken into account – additionally has to be solved on the interface.

A possible approach to simulate the effects of surfactants on mass transfer rates and terminal velocities of drops without solving an extra equation for surfactant transport consists in the simulation of a stagnant cap, as done by Dani et al. [2] for spherically shaped drops. A theoretical approach to stagnant cap regions is presented by Palaparthi et al. in [3].

The key issue in the simulation of two–phase flow problems is the numerical representation and model for the interface. One very popular class of methods are front capturing methods, where the interface is obtained implicitly as the zero–level–set of a distance function to the interface (for level set methods see [4, 5]) or where the distribution of the fluid phases is given in terms of a volume fraction function [6, 7] (VOF). In either case, both fluid phases are usually solved on a fixed grid, and topological changes of the phases can be tracked by these methods. Advanced methods have been developed to improve the handling of surface tension and to diminish the numerical error due to smearing of fluid parameters across the interface or to capture discontinuous functions (pressure, solute concentration).

Another popular class of methods are front tracking methods [8, 9]. Front tracking methods follow the propagation of the interface explicitly. Techniques to map surface tension and to represent discontinuous functions on the fixed grid are necessary.

Simulation of buoyancy driven bubbles with and without species transport with the VOF method have been performed in [10, 11]. The level set method has been applied to the simulation of deformable liquid drops rising in ambient liquid by Bertakis et al. [12]. Wang [13] and Yang [14] used the level set method to simulate species transport with the level set method. Smolianski [15] applied the level set method to the simulation of gas bubbles.

Front tracking methods have been applied to the simulation of fluid drops by Tryggvason [8], Aulisa et al. [16], and by Darmiana et al. [17].

The above mentioned list of references is by far not complete, many similar approaches have been developed.

The method presented in this paper belongs to a class of methods called mesh moving methods or sharp interface methods. In mesh moving methods the grid is fitted to the boundary such that a subset of edges (in 2d) or faces (in 3d) of the triangulation coincides with the interface. This fitting of the mesh leads to a desirable high accuracy in resolving interfacial phenomena, especially surface tension effects, and to a generally simpler implementation of interface conditions. A downside of these mesh moving methods with a fitted interface is their applicability to problems with moderate interfacial movement only. Prob-
lems with topological changes cannot be depicted with such a method in general. However, if the considered problem allows the use of a mesh moving method, the benefits of the high accuracy may be exploited.

The mesh moving method has been combined with a finite difference solver to simulate bubbles by Ryskin et al. in [18] in form of staggered iterations of each phase separately, until convergence of the interfacial data is achieved. Numerical simulations of buoyancy driven droplets using a mesh moving method in a finite element context are presented by the authors et al. in [19] to validate drop rise velocities and drop deformation in standard test systems.

The central point of the method presented in this paper is a reformulation of the weak two-phase flow problem, where the interface conditions are conveniently included in suitably chosen subspaces. This formulation leads to a combined treatment of both phases in one system of equations. Since a co-moving frame of reference is used (see Section 2), only moderate mesh distortion has to be expected for single drop simulations and a sharp interface representation may be used. The advantage of the mesh moving method lies in a substantial increase of accuracies for the same number of unknowns. On the discrete level this formulation on appropriate subspaces can be implemented with discrete projections working on the interface only. For this reason we call the method subspace projection method.

Moreover, the subspace projection method is a versatile method for the implementation of interface conditions in general two-phase flows on the discrete level. In this article we give a detailed description of the method and show its applicability to systems with interface conditions where the weak formulation of the problem can be written as described in Section 3.

The rest of this paper is organized as follows. In Section 2 we give a detailed model (in strong form) of a single drop flow problem, with interesting special cases of rigid drops, spherically shaped drops and drops with a partly immobile interface (stagnant cap). Due to suitably chosen subspaces, interface conditions for the flow field, as well as pressure and species discontinuity are inherited to the solutions of the combined set of equations. The subspaces suitable for trial and test functions for the given flow and species transport problem are described as projections of general function spaces. This approach is described in detail, starting with a weak formulation in Section 3. The weak formulation is then transferred to the finite element context - the subspace projection method - in Section 4.

We discuss numerical issues concerning the results obtained by the method (parasitic currents, wall effect, resolution of boundary layers) in Section 5.

To illustrate the versatile range of applications we present simulations of spherically shaped drops with stagnant caps in Section 6. The effects of the immobile interface include reduction of solute transfer rates [20, 21] and terminal rise velocities [22, 23].

2. Mathematical model for two-phase flow

The mathematical model to describe the system of a two-phase flow consisting of a (possibly) deformable liquid drop (hereafter called disperse phase) suspended in a continuous second liquid phase with additional diffusive–convective transport of a (chemically inert)
substance consists of the hydrodynamic equations in either phase, a transport equation for the solutant and a set of boundary conditions at the interface of the two liquid phases. The corresponding equations are discussed in [24, 25] and others. They are presented in a two–phase non–dimensional form in the following sections.

A sketch of the geometry and the corresponding nomenclature is given in Figure 1.

One of the main features of our method consists of a co–moving frame of reference, as we transform the governing equations to an accelerated coordinate system moving with the center of gravity of the drop. This approach has several significant advantages for the numerical simulation. Most important, a sharp interface method/mesh moving method may be used, where the interface coincides with inner element boundaries, since mesh distortion is significantly reduced. Combined with the piecewise quadratic representation of the interface and the variational formulation of the curvature, see below, this approach is very well suited for handling interfacial phenomena.

The acceleration of the corresponding coordinate system is time–dependent but spatially constant and is taken into account by an additional volume force and according inlet–boundary conditions in the momentum equation of the fluid flow, see [26].

2.1. Fluid mechanics

In the bulk of each phase we consider Newtonian, incompressible, isothermal fluids. We assume the fluid parameters density and viscosity to possibly differ for the two phases, however constant within each phase.

Distinguishing the unknown velocity $\mathbf{u}$ and pressure $p$ as well as the parameters by a subscript $j$, $j = c$ or $j = d$ for the continuous and disperse phase, respectively, the hydrodynamic equations take the non–dimensional form:
\[
\begin{align*}
\Lambda_j \partial_t \mathbf{u}_j + \Lambda_j (\mathbf{u}_j \cdot \nabla) \mathbf{u}_j - \frac{1}{Re_j} \Delta \mathbf{u}_j + \nabla p_j &= \Lambda_j f_j \quad \text{in } \Omega_j, \quad (1a) \\
\nabla \cdot \mathbf{u}_j &= 0 \quad \text{in } \Omega_j. \quad (1b)
\end{align*}
\]

Here, \( \Lambda_j := \frac{\rho_j}{\rho_c} \) is defined as the ratio of the densities, \( f_j \) is the phase depending right hand side, consisting of the gravitational force density and an additional term due to the acceleration of the coordinate system. The dimensionless Reynolds number \( Re_j := \frac{\mu_j}{UL \rho_c} \) is defined via the density of the continuous phase, a characteristic velocity scale \( U \), and a characteristic length scale \( L \), in the case of the drop the drop diameter \( d_p \). The benefit of these special definitions lie in the unaltered shape of the interfacial conditions. For a more detailed discussion of the 2-phase fluid model we refer to [26]. The interface conditions are as follows:

\[
\begin{align*}
[f] &= 0 \quad \text{on } \Gamma, \quad (2a) \\
[\sigma \nu] &= \frac{1}{W\kappa} \nu \quad \text{on } \Gamma, \quad (2b) \\
V_\Gamma &= (\mathbf{u} \cdot \nu) \nu \quad \text{on } \Gamma, \quad (2c)
\end{align*}
\]

where \( \sigma_j = \frac{1}{Re_j} D(\mathbf{u}) - p \mathbf{I} = \frac{1}{Re_j} (\nabla \mathbf{u} + \nabla \mathbf{u}^T) - p \mathbf{I} \) denotes the stress tensor, \( \nu \) the normal vector on \( \Gamma \), pointing from \( \Omega_c \) to \( \Omega_d \); \( \kappa \) is the sum of the principal curvatures, with the convention \( \kappa > 0 \) if \( \Omega_d \) is convex. \( V_\Gamma \) denotes the velocity of \( \Gamma \) in direction of \( \nu \) and \( We = \frac{\rho_c LU^2}{\gamma_0} \) the Weber number, with the constant surface tension \( \gamma_0 \).

The brackets \([\cdot]\) stand for a jump of a quantity, say \( f \), across \( \Gamma \). For a point \( x_0 \in \Gamma \) we use the definition:

\[
[f](x_0) := \lim_{x \to x_0} \begin{cases} f_c(x) & x \in \Omega_c \\ f_d(x) & x \in \Omega_d \end{cases}
\]

Eq. (2a) states continuity of the velocity across the interface (the no-slip condition), (2b) characterizes the jump of stresses at the interface, or – in other words– the equilibrium of surface tension force and the forces exerted by the bulk fluid on the interface. Eq. (2c) is called kinematic condition, expressing that \( \Gamma \) is a material interface, moving according to the normal component of the velocity.

The above system has to be complemented by suitable initial conditions and boundary conditions on \( \partial \Omega \).

#### 2.2. Species transport

The time–depending transport of a solutant by means of diffusion and convection is described in non–dimensional form by

\[
\partial_t c_j + \nabla \cdot (c_j \mathbf{u}_j) - \frac{1}{Pe_j} \Delta c_j = 0 \quad \text{in } \Omega_j, \quad (3)
\]
with the non-dimensional solutant concentration \( c \) and the Peclet numbers \( Pe_j := \frac{UL_d}{D_j} \). \( D_j \) is the diffusion coefficient. The interface conditions for the concentration read:

\[
\begin{align*}
\left[ \frac{1}{Pe} \nabla c \cdot \nu \right] &= 0 \quad \text{on} \ \Gamma, \\
(c) &= \frac{1}{H} c_d \quad \text{on} \ \Gamma.
\end{align*}
\]

Eq. (4a) states that the flux is continuous across the interface. Thus in this model, no solutant is absorbed on the interface. Eq. (4b) is called Henry’s law, \([25]\), with the partition coefficient \( H \). (4b) is a modeling assumption stating that the concentration is in thermodynamic equilibrium at the interface. Moreover, for simplicity, we assume that the partition coefficient \( H \) does not depend on \( c \) but is rather constant in space and time.

Again, system (3) – (4) has to be closed by initial conditions and boundary conditions at the outer boundary.

In the rest of the section we discuss special cases. The special cases of rigid drops and spherically shaped drops might be of particular interest as simplified models neglecting drop deformation. Although the following models have to be regarded as approximations to the full system, they are easier to handle mathematically and computationally and already provide some insights into the complex phenomena of the system. The special case of drops with a stagnant cap can be applied to spherically shaped drops and gives some insight into the effects of surfactants on drop dynamics.

2.3. Special case of a rigid “drop”

Considering the “drop” as a rigid body allows for some significant simplifications of the governing equations. Most importantly the fluid dynamics need to be solved only in the continuous phase, since rigid “drop” implies zero velocity inside the droplet \( (\mathbf{u}_d = 0) \).

The equations for the concentration remain unaltered except for the omission of the convection–term within the disperse phase. The interface conditions for fluid dynamics and species transport now read:

\[
\begin{align*}
\mathbf{u}_c &= 0 \quad \text{on} \ \Gamma, \\
V_T &= 0 \quad \text{on} \ \Gamma, \\
\left[ \frac{1}{Pe} \nabla c \cdot \nu \right] &= 0 \quad \text{on} \ \Gamma, \\
(c) &= \frac{1}{H} c_d \quad \text{on} \ \Gamma.
\end{align*}
\]

2.4. Special case of a spherical, non–deformable drop

In contrast to the above case of a rigid drop, this special case considers fluid motion within the drop, however, assumes the drop to be non–deformable and of spherical shape. Even though this case is an artificial model, in real world applications drops are observed to be of spherical shape up to a certain diameter \([27]\). This is due to the strong capillary forces for small diameters. Thus, for one, the spherical case is a good model for drops of
small diameters, and furthermore allows for insight into the complex interfacial phenomena
for larger drop diameters, when compared to the deformable setting.

As mentioned above, drop deformations are not considered in this spherical model, thus
only tangential velocities arise at the interface.

Eqs. (1), (3) and (4) remain valid, deviations to the full model occur in the set of interface
conditions (2). For the case of a spherical droplet they now take the form:

\[
\begin{align*}
[u] \cdot \tau &= 0 \quad \text{on } \Gamma, \\
[u_e \cdot \nu] &= 0 \quad \text{on } \Gamma, \\
[\tau \cdot \sigma \nu] &= 0 \quad \text{on } \Gamma, \\
V_\Gamma &= 0 \quad \text{on } \Gamma
\end{align*}
\]  

(6a), (6b), (6c), (6d)

for any tangential vector \( \tau \). For further reference see [25] and [26].

2.5. Stagnant cap

The essential characteristic of surfactant molecules, their hydrophobic and hydrophilic
parts, cause the surfactants to preferably occupy the interface in two-phase flow systems.
Once allocating the interface of rising drops, the typical flow drifts the surfactants to the rear
area. The occupation of the interface leads to an immobility, as the flow field is oppressed
in tangential directions on the interface. This immobile area of a drop is called \textit{stagnant cap}
(Figure 2). ([2, 3]).

To obtain a simple model for the simulation of spherically shaped drops with stagnant
caps, in addition to the no-slip condition, we assume that in the stagnant cap area no
tangential velocities arise. We define the stagnant cap area \( \Gamma_s \subset \Gamma \) of the interface with
stagnant cap angle \( \alpha \in [0, \pi] \) as

\[
\Gamma_s := \left\{ x \in \Gamma : \arccos \left( \frac{-y}{|x|} \right) \leq \alpha \right\}.
\]  

(7)
Recall that the barycenter of the drop is fixed to the center of origin of the coordinate system. The model of a spherical drop with a stagnant cap angle \( 0 \leq \alpha \leq \pi \) consists of the above given model augmented by the condition

\[
\mathbf{u}_c \cdot \tau = 0 = \mathbf{u}_d \cdot \tau \quad \text{on } \Gamma_S. \tag{8}
\]

Thus for spherical droplets, the velocity in the stagnant cap area fulfills \( \mathbf{u}_c \cdot \nu = 0 = \mathbf{u}_d \cdot \nu \) (Eq. (6b)) in addition to \( \mathbf{u}_c \cdot \tau = 0 = \mathbf{u}_d \cdot \tau \) (Eq. (8)), resulting in the no-slip condition \( \mathbf{u}_c = 0 = \mathbf{u}_d \) on \( \Gamma_S \).

3. Functions spaces and weak formulation

One of the main difficulties in solving (3), (4) consists in how to deal with the jump of \( c \), given by Henry’s law (4b) or with the discontinuity of the pressure (2b) across the interface. An approximation of the pressure by continuous functions leads to undesirable spurious velocities, see [28, 29]. The presented subspace projection method handles functions with discontinuities across the interface by working with suitable subspaces that inherit the compliance of interface conditions.

The first part of this section introduces the appropriate function spaces followed by the weak formulation for the example of the scalar convection diffusion equation for two–phase species transport and the Navier–Stokes equations.

3.1. Function spaces

To start with, we introduce the function spaces needed to derive a weak formulation of our system and to develop a finite element discretization.

Fixing one time instant, for the moment being we consider \( \Omega_c \) and \( \Omega_d \) to be given. Then define \( \mathcal{V} := \mathcal{H}^1(\Omega_d) \times \mathcal{H}^1(\Omega_c) \) the “product space” of Sobolev functions defined locally on each phase, where no kind of continuity across \( \Gamma \) is imposed. Let \( \mathcal{W} := \mathcal{L}^*(\Omega_d) \times \mathcal{L}^*(\Omega_c) \) be the appropriate pressure space, for \( \mathcal{L}_2^*(\Omega) := \{ f \in \mathcal{L}_2(\Omega) : \int_\Omega f \, dx = 0 \} \).

Furthermore, for piecewise given functions \( f \in \mathcal{V} \) define the broken inner product by

\[
(g, h)_{\Omega} := (g_c, h_c)_{\mathcal{L}^2(\Omega_c)} + (g_d, h_d)_{\mathcal{L}^2(\Omega_d)}. \tag{9}
\]

For any \( \lambda \in \mathbb{R} \)

\[
\mathcal{V}_\lambda := \{ f = (f_d, f_c) \in \mathcal{H}^1(\Omega_d) \times \mathcal{H}^1(\Omega_c) : f_d = \lambda f_c \text{ on } \Gamma \} \tag{10}
\]
defines the subspace of \( \mathcal{V} \) consisting of functions that have a prescribed jump of \( \lambda \) across the interface (in a weak sense on the respective trace spaces). Note that \( \mathcal{V}_1 \) is the space of functions continuous across \( \Gamma \) and that all \( f \in \mathcal{V}_H \) automatically fulfill the Henry law (4b).

A useful subspace of \( (\mathcal{V}_1(\Omega))^d \) is

\[
\mathcal{V}_\nu := \{ f \in (\mathcal{V}_1(\Omega))^d : f_d \cdot \nu = 0 = f_c \cdot \nu \text{ on } \Gamma \}, \tag{11}
\]
with the unit normal \( \mathbf{v} \) on \( \Gamma \). \( \mathcal{V}_\mathbf{v} \) is the space of functions with vanishing normal traces. This subspace will be utilized when modelling spherical drops.

\( \Omega_c = \Omega_c(t) \) and \( \Omega_d = \Omega_d(t) \) depend on time in general, since the interface is moving in time and thus \( \mathcal{V} = \mathcal{V}(t) \) and \( \mathcal{V}_H = \mathcal{V}_H(t) \) are time–dependent.

3.2. Weak formulation of the advection–diffusion equation

Without precisely specifying the correct parabolic space and also assuming no–flux conditions at the outer boundaries (other cases can be treated likewise) for the ease of presentation, consider the following variational formulation:

Find \( c \) with \( c(t, \cdot) \in \mathcal{V}_H(t) \) for all time instants \( t > 0 \) such that

\[
(\partial_t c + \nabla \cdot (uc), \phi)_{\Omega} + \frac{1}{Pe}(\nabla c, \nabla \phi)_{\Omega} = 0 \quad \text{for all } \phi \in \mathcal{V}_1(t). \quad (12)
\]

Note that in the above formulation \( c \) is discontinuous across \( \Gamma \) for \( H \neq 1 \), whereas the test functions are taken from the space \( \mathcal{V}_1 \) of functions being continuous across \( \Gamma \). Also note that the inner products are defined in a broken sense, i.e. piecewise for \( \Omega_c \) and \( \Omega_d \), as given in definition (9). The Peclet number - and from now on all constants - are defined piecewise over the domains \( \Omega_c \) and \( \Omega_d \).

The next lemma states that the above weak formulation is equivalent to the strong form (3), (4) (assuming sufficient regularity). The crucial point here is that the weak formulation contains both conditions (4a) and (4b).

**Lemma 1.** Let \( c \) be a classical, piecewise smooth solution of (3), (4). Then \( c \) is also a weak solution of (12). On the other hand, if \( c \) is a weak solution of (12) and additionally \( c \) is piecewise smooth, then \( c \) is a classical solution of (3) fulfilling the interface conditions (4).

**Proof**

\( \Rightarrow \): Let \( c \) be a classical, piecewise smooth solution and \( \phi \in \mathcal{V}_1 \) a testfunction. Then \( c \in \mathcal{V}_H \) and

\[
(\partial_t c + \nabla \cdot (uc), \phi)_{\Omega} - \left( \frac{1}{Pe} \Delta c, \phi \right)_{\Omega} = 0.
\]

Since we assume homogeneous Neumann conditions on \( \partial \Omega \), integration by parts yields:

\[
(\partial_t c + \nabla \cdot (uc), \phi)_{\Omega} + \left( \frac{1}{Pe} \nabla c, \nabla \phi \right)_{\Omega} - \int_{\Gamma} \frac{1}{Pe} \mathbf{v} \cdot \nabla c \phi \right] = 0.
\]

The last term equals zero because of \( \phi \in \mathcal{V}_1 \) and (4a).

\( \Leftarrow \): Let \( c \) be a piecewise smooth weak solution of (12). Since \( c \in \mathcal{V}_H \), (4b) is fulfilled. As above, integrating (12) by parts yields:

\[
(\partial_t c + \nabla \cdot (uc), \phi)_{\Omega} - \left( \frac{1}{Pe} \Delta c, \phi \right)_{\Omega} + \int_{\Gamma} \frac{1}{Pe} \mathbf{v} \cdot \nabla c \phi \right] = 0.
\]
Now, as usual, first consider testfunctions \( \phi \) with \( \text{supp}(\phi) \subset \Omega_j, \ j = c,d \). Using the fundamental lemma of calculus of variations it is concluded that \( c \) solves Eq. (3) in a pointwise sense in the interior of each domain \( \Omega_c \) and \( \Omega_d \). Then by considering any testfunction \( \phi \in V_1 \), and observing that \( c \) solves (3), one deduces that the flux condition (4a) holds. Thus the proof is completed.

Now let \( \mathcal{P}_\lambda : V \to V_\lambda \) be a projection from the general space \( V \) onto \( V_\lambda \). Then (12) is clearly equivalent to: find \( \tilde{c} \) with \( \tilde{c}(t, \cdot) \in V \) for all \( t > 0 \) such that

\[
(\partial_t \mathcal{P}_H \tilde{c}, \mathcal{P}_1 v)_\Omega + (\nabla \cdot (u \mathcal{P}_H \tilde{c}), \mathcal{P}_1 v)_\Omega + \frac{1}{Pr} (\nabla \mathcal{P}_H \tilde{c}, \nabla \mathcal{P}_1 v)_\Omega = 0 \quad \text{for all} \ v \in V. \tag{13}
\]

Then set \( c := \mathcal{P}_H \tilde{c} \).

3.2.1. Weak formulation of the Navier–Stokes equations

We give a variational formulation of Eqs. (1), where the free boundary condition (2b) is transformed to a boundary integral part of the bilinear forms. This approach was developed in [30] for a one–phase problem with a free boundary. Applying it to interface condition (2b) yields:

\[
-\int_{\Gamma} \phi \cdot \sigma \nu = -\int_{\Gamma} \phi \cdot [\sigma \nu] = -\frac{1}{We} \int_{\Gamma} \kappa \nu \cdot \phi = -\int_{\Gamma} \nabla x \cdot \nabla \phi,
\]

where \( \nabla \) is the tangential gradient on \( \Gamma \) and \( x \) the position vector.

Analogously to the derivation of the convection–diffusion equation, we obtain the complete variational formulation of the two–phase Navier–Stokes equations with a free boundary and with the above given projection. Recall (see Eqs. (1)) that \( \Lambda = \frac{\rho_i}{\rho_c} \) is the phase–wise defined ratio of fluid densities, and \( f_j \) is the acceleration acting on the fluid.

Find \( (\tilde{u}, p) \) with \( \tilde{u}(t, \cdot) \in (V(t))^d, p(t, \cdot) \in W(t) \) for all time instants \( t > 0 \) such that

\[
\Lambda(\partial_t \mathcal{P}_1 \tilde{u}, \mathcal{P}_1 \phi)_{\Omega} + (\mathcal{P}_1 \tilde{u} \cdot \nabla \mathcal{P}_1 \tilde{u}, \mathcal{P}_1 \phi)_{\Omega} + \frac{1}{2Re} \int_{\Omega} D(\mathcal{P}_1 \tilde{u}) : D(\mathcal{P}_1 \phi) - (p, \nabla \cdot \mathcal{P}_1 \phi)_{\Omega} + \frac{1}{We} \int_{\Gamma} \nabla x \cdot \nabla \phi = \Lambda(f, \mathcal{P}_1 \phi)_{\Omega} \quad \forall \phi \in V^d, \tag{14a}
\]

\[
(\nabla \cdot \mathcal{P}_1 \tilde{u}, \psi)_{\Omega} = 0 \quad \forall \psi \in W. \tag{14b}
\]

Then set \( u = \mathcal{P}_1 \tilde{u} \). Again, this variational formulation, under sufficient smoothness conditions on \( u \) and \( p \) is equivalent to the classical form (1) with boundary conditions (2).
4. Finite element discretization and the subspace projection method (SPM)

Based on the weak formulation (13) or (14), a finite element discretization in connection with a sharp interface method is now straightforward. Given a geometry $\Omega$, we assume a triangulation $\mathcal{T}$ of $\Omega$ respecting the interface $\Gamma$ in a sense that no edge/face of $\mathcal{T}$ (in 2d/3d respectively) intersects $\Gamma$. The next step is to define the discrete counterparts of the function spaces defined in Section 3.1. To enable the representation of discontinuous functions across the interface, the nodes at the interface $\Gamma$ are doubled, see Fig. 3.

4.1. Discrete counterpart of $\mathcal{V}$

The node doubling results in two triangulations $\mathcal{T}_j$, $j = c$ and $j = d$ where the respective interface nodes have the same coordinates. Now define the discrete finite element spaces $\mathcal{V}_{h,c}$ on $\Omega_c$ and $\mathcal{V}_{h,d}$ on $\Omega_d$, respectively, consisting of Lagrange basis functions of order $k \in \mathbb{N}$:

$$\mathcal{V}_{h,\Omega_d} := \{ \varphi \in C^0(\Omega_d) : \varphi|_T \in P_k \text{ for all } T \in \mathcal{T}_d \} = \text{span}\{ \varphi_i : i \in I_d \},$$

$$\mathcal{V}_{h,\Omega_c} := \{ \varphi \in C^0(\Omega_c) : \varphi|_T \in P_k \text{ for all } T \in \mathcal{T}_c \} = \text{span}\{ \varphi_i : i \in I_c \}.$$

$I_d$ and $I_c$ denote the respective index sets of Lagrange nodes of the dispersive and continuous phases. We require $I_c \cap I_d = \emptyset$, describing the fact that no (logical) node belongs to both phases.

Note that the supports of all Lagrange basis functions are always completely contained in one phase: $\text{supp}(\varphi_i) \subset \Omega_d$ for all $i \in I_d$, and $\text{supp}(\varphi_i) \subset \Omega_c$ for all $i \in I_c$.

The interface nodes from one phase, say the continuous phase, are called primary nodes, while the corresponding interface nodes from the other phase are the secondary nodes. Thus there is a one-to-one correspondence between secondary and primary nodes. Let $I_P$ and $I_S$ be the respective index sets of primary and secondary nodes with $I_P \subsetneq I_c$ and $I_S \subsetneq I_d$.

Then we define a mapping between the index sets:
\[ s : \quad I_c \cup I_d \rightarrow I_c \cup \{0\} \quad (15) \]

\[ i \mapsto s(i) = \begin{cases} 0 & \text{if } i \not\in I_S; \\ p = \text{matching primary node number} & \text{if } i \in I_S. \end{cases} \]

Note that the mapping \( s \) does not depend on time, since it operates on nodal index sets, and not on the associated coordinates of the vertices. For an illustration see Figure 3.

Let \( \mathcal{V}_h \subset \mathcal{V} \) be the finite element subspace on the domain \( \Omega = \Omega_c \cup \Omega_d \):

\[ \mathcal{V}_h := \{ \phi = (\varphi_d, \varphi_c) \in \mathcal{V}_{h,\Omega_d} \times \mathcal{V}_{h,\Omega_c} \} = \text{span}\{\phi_i, \phi_j : \phi_i := (\varphi_i, 0), \phi_j := (0, \varphi_j) \text{ with } i \in I_d, j \in I_c\}. \quad (16) \]

And accordingly define \( \mathcal{V}_{h,\lambda} \subset \mathcal{V}_h \):

\[ \mathcal{V}_{h,\lambda} := \{ \varphi = (\varphi_d, \varphi_c) \in \mathcal{V}_h : \varphi_d|_\Gamma = \lambda \varphi_c|_\Gamma \} . \]

With the definition of \( \phi \) in (16) and defining \( \varphi_0 := 0 \) and \( I := I_c \cup I_d \setminus I_S \), we can write

\[ \mathcal{V}_{h,\lambda} = \text{span}\{\phi_i + \lambda \phi_{s(i)} : i \in I\} . \]

From a computational point of view, working with the spaces \( \mathcal{V}_{h,\lambda} \) is not convenient. Instead, let \( \mathcal{P}_\lambda \) be a projection, mapping \( \mathcal{V}_h \) onto \( \mathcal{V}_{h,\lambda} \):

\[ \mathcal{P}_\lambda : \mathcal{V}_h \rightarrow \mathcal{V}_{h,\lambda} . \]

Each function of the finite element subspaces can now be represented by its nodal values, and we identify the coefficient vector \( F \in \mathbb{R}^N, \quad N := |I_c \cup I_d|, \) with the finite element function

\[ f = \sum_{i \in I_c \cup I_d} F_i \phi_i. \]

The finite element function \( f \) and the vector \( F \) of nodal values are linked via the mapping \( J \):

\[ F = J(f), \quad f = \sum_{i \in I_c \cup I_d} F_i \phi_i. \]

The matrix representation \( P_\lambda \) of \( \mathcal{P}_\lambda \) is given by

\[ P_\lambda : \quad \mathbb{R}^N \rightarrow \mathbb{R}^N, \quad P_\lambda(J(f)) = J(\mathcal{P}_\lambda f) \quad \text{for } f \in \mathcal{V}_h. \quad (17) \]

Since \( P_\lambda \) operates on nodal values (whose numberings do not change over time), the time–dependency of the matrix representation of the discrete projections is safely omitted.
4.2. Discretization of the advection–diffusion equation

Choosing discrete time instants $0 = t_0 < t_1 < t_2 < \ldots$ and corresponding time step sizes $\tau_n := t_{n+1} - t_n$, we discretize the advection–diffusion equation by the backward Euler method in time and the above introduced finite elements in space. Similar to the weak form \(^{(12)}\) of the equation, the fully discrete system is defined by: find $\tilde{c} \in \mathcal{V}_h$ and $c^{n+1} = P_H \tilde{c}$ such that

\[
a(P_H \tilde{c}, P_1 \phi) := (P_H \tilde{c}, P_1 \phi)_\Omega + n(u; P_H \tilde{c}, P_1 \phi)_h + \frac{\tau_n}{Pe} (\nabla P_H \tilde{c}, \nabla P_1 \phi)_\Omega = (c^n, P_1 \phi)_h \tag{18}
\]

for all $\phi \in \mathcal{V}_h$. Here, $n(\cdot; \cdot, \cdot)_h$ stands for the convective term. Since the problem is rather convection dominated, we use a Streamline Upwind Petrov Galerkin method (SUPG) \(^{31}\) in order to stabilize the problem (for possible choices of SUPG–parameters see \(^{32,33}\)).

The bilinear form $a(\cdot, \cdot)$ induces an operator $A : \mathcal{V}_h \to \mathcal{V}_h$ by

\[
a(v, w) = (Av, w)_\Omega \quad \text{for all } v, w \in \mathcal{V}_h,
\]

so that \((18)\) is equivalent to

\[
(P_1^* A P_H \tilde{c}, \phi)_\Omega = (c^n, P_1 \phi)_h \quad \text{for all } \phi \in \mathcal{V}_h, \tag{19}
\]

with $P_1^*$ the adjoint of $P_1$ with respect to the $L^2$ inner product.

For a matrix form of Eqs. \((18)\) or \((19)\) let $M, N(u)$ and $K$ denote the respective mass matrix, advection matrix and stiffness matrix with respect to the nodal basis in $\mathcal{V}_h$, for example

\[
K_{i,j} = \frac{1}{Pe} (\nabla \phi_j, \nabla \phi_i)_\Omega \quad i, j = 1, \ldots, I_c \cup I_d.
\]

Then \((18)\) is equivalent to

\[
\bar{A} \bar{C} := (P_1)^T A P_H \tilde{C} := (P_1)^T (M + \tau_n N(u) + \tau_n K) P_H \tilde{C} = (P_1)^T F, \tag{20}
\]

where $(P_1)^T$ is the transpose of $P_1$ with respect to the standard Euclidean inner product and $F$ is the right hand side.

Eq. \((18)\) or equivalently \((19)\) or \((20)\) is what we call the subspace projection method (SPM).

4.3. Implementation of the subspace projection method

So far the projection $P_\lambda$ has not yet been specified. In our implementation we use the following projection, given in nodal form by the pointwise formula

\[
(P_\lambda F)_i := \begin{cases} \lambda F_{s(i)} & \text{if } i \in I_s, \\ F_i & \text{else} \end{cases} \tag{21}
\]

with $s(i)$ defined in \((15)\).
Correspondingly, the transpose $P^T_\lambda$ of $P_\lambda$ is given by

$$(P^T_\lambda F)_i := \begin{cases} F_i + \lambda F_{s^{-1}(i)} & \text{if } i \in I_P, \\ 0 & \text{if } i \in I_S, \\ F_i & \text{else}. \end{cases}$$ (22)

Thus, in order to solve (20), the system matrix $A := M + \tau_n N(u) + \tau_n K$ is constructed in the full, convenient finite element space $V_h$. Using an iterative solver like for instance GMRes, the evaluation of the operator $(P_1^T A P_H)$ is then realized by pre- and post-multiplying the iterates by $P_H$ and $P_1^T$, respectively. To this end, the projection matrices are not built explicitly, but rather the projections are realized by simple routines. For example projection (21) reads:

```fortran
subroutine project(lambda,F)
  do i=1,N
    if(i secondary node ) then
      F(i) = lambda*F(primary(i))
    endif
  end do
end subroutine
```

For each of the conditions (2a), (2b), (4a), and (4b) the projections can be explicitly given similar to the above routines. For the different settings (i.e. for rigid, spherical, or deformable sphere) the specific interface conditions can be modularly turned on or off via different projections routines, and the code remains the same for all special cases of Section 2, except for the respective projection routines.

4.4. Solvability of the $P^T_1 A P_H$ system

Since $\tilde{A}$ has a large kernel and because of the nonsymmetry of (18) with respect to the projections, the solvability of the system is a priori not completely obvious. We answer this question in the following.

**Lemma 2.** Let the projection $P_\lambda$ be given by (21) and (17) and $\phi_i$ as in definition (16). It holds

$$\ker(P_\lambda) = \text{span}\{\phi_i : i \in I_S\}, \quad \ker(P^*_\lambda) = \text{span}\{\lambda \phi_i - \phi_{s^{-1}(i)} : i \in I_P\}.$$ 

Proof: clear from (21), (22).

**Proposition 3.** Solvability of (18)

Let $H > 0$ and the bilinear form $a(\cdot, \cdot)$ be coercive on $V_h$ (which is at least the case for sufficiently small $\tau_n$). Then (18) or equivalently (19) or (20) admits a solution $\tilde{c}$. Any further solution $\tilde{c}$ is given by $\tilde{c} = \tilde{c} + y$, where $y \in \ker(P_H) = \text{span}\{\phi_i : i \in I_S\}$. Moreover, $c^{n+1} = P_H \tilde{c}$ is uniquely determined.
Proof: let \( c \in V_h \) be such that \( a(P_H c, P_1 \phi) = 0 \) for all \( \phi \in V_h \) or equivalently
\[
P_1^* A P_H c = 0.
\]

First we show that
\[
\ker(P_1^* A P_H) = \ker(P_H).
\]

Let \( \omega := P_H c, \omega = \sum_i \omega_i \phi_i \). Define \( \tilde{\omega} = \sum_i \tilde{\omega}_i \phi_i \) by
\[
\tilde{\omega}_i := \begin{cases} 
\sqrt{H} \omega_i & i \in I_c, \\
\omega_i & i \in I_d.
\end{cases}
\]

Since \( \omega \in V_{h,H} \), it is clear that \( \tilde{\omega} \in V_{h,\sqrt{H}} \). For nodes \( x_i \) not at the interface, one readily sees
\[
a(\tilde{\omega}, \phi_i) = (A\tilde{\omega}, \phi_i)_\Omega = 0.
\]

On the other hand, for nodes \( x_i \) lying on the interface, i.e. \( i \in I_S \) or \( i \in I_P \) one computes
\[
(A\tilde{\omega}, \phi_i)_\Omega = (A\omega, \phi_i)_\Omega \quad i \in I_P, \\
(A\tilde{\omega}, \phi_i)_\Omega = (A\omega, \phi_i)_\Omega \quad i \in I_S.
\]

This together with the fact that \( A\omega \in \ker(P_1^*) \) and from Lemma 2 it follows that \( A\tilde{\omega} \in \ker(P_{\sqrt{H}}^*) \) and in particular
\[
0 = (P_{\sqrt{H}}^* A\tilde{\omega}, \tilde{\omega}) = (A\tilde{\omega}, P_{\sqrt{H}} \tilde{\omega}) = a(\tilde{\omega}, P_{\sqrt{H}} \tilde{\omega}) = a(\tilde{\omega}, \tilde{\omega}).
\]

From the coercivity of \( a(\cdot, \cdot) \) we conclude \( \tilde{\omega} = 0 \) and then also \( \omega = 0 \), which means \( c \in \ker(P_H) \) and therefore
\[
\ker(P_1^* A P_H) = \ker(P_H).
\]

Likewise one shows
\[
\ker(P_H^* A^* P_1) = \ker(P_1).
\]

Now, \( \text{rg}(P_1^* A P_H) = \ker(P_H^* A^* P_1) = \ker(P_1) = \text{rg}(P_1^*) \). This means, (18) admits a solution. The rest of the lemma is clear.

\[\square\]

4.5. Solving the fluid dynamics equations

Discretizing Eqs. (1) and (2), the free boundary conditions (2) cause several problems, in particular the treatment of the curvature terms as pointed out in Section 3.2.1. The variational formulation presented there is a powerful tool for the accurate discretization of the surface forces.

Additionally the fluid dynamic equations are stabilized by the Grad-Div stabilizing method ([34, 35]), where the bilinear form is augmented by the term \( C(\nabla \cdot u, \nabla \cdot \phi) \) with \( C \) a constant of order 1. Adding this consistent term leads to a reduction of the pointwise error of \( \nabla \cdot u \), i.e. the local mass balance, which is fundamental for the reliable solution of the species transport equations.
Time discretization. To evolve in time we proceed as follows: given the values \( \Omega^n, u^n, p^n \) at the discrete time instant \( t_n \), we compute

A: \( u^{n+1}, p^{n+1} \) by solving (1) with boundary conditions (2a) and (2b) in \( \Omega^n \) with the \textit{subspace projection method},

B: \( \Gamma^{n+1} \) by a discretized form of (2c): \( \Gamma^{n+1} := \Gamma^n + (t_{n+1} - t_n) u^{n+1} \),

C: update of the domain by an \textit{extension} of \( \Gamma^{n+1} \) into the interior, resulting in \( \Omega^{n+1} \).

In A the boundary condition (2b) is incorporated via the variational formulation according to (14).

For the extension of \( \Gamma^{n+1} \) into the interior in C a smoothing of the triangulation by a discrete Poisson equation is used. Details on how to obtain an extension to \( \Gamma^{n+1} \) are given in [36, 37].

The computation of \( u^{n+1}, p^{n+1} \) is based on the axisymmetric version of the method and implementation described in [37]. The underlying code uses the fractional step \( \theta \)–scheme in a variant as operator splitting, which decouples two major numerical difficulties, the solenoidal condition and the nonlinearity, see [38]. The axisymmetric code solves for the unknowns \( (v_s, v_z) \) and \( p \), where \( v_s := \frac{v_r}{r} \). This scaling resolves the singularities of the operators at the axis of symmetry \( (r = 0) \), and may be interpreted as a proper variational formulation of the axisymmetric Navier-Stokes equation in appropriately \( r \)–weighted Sobolev spaces, see also [39].

Spatial discretization. To discretize in space the Taylor-Hood element, i.e. piecewise quadratic, globally continuous elements for \( u \) and piecewise linear, globally continuous (except for the jump at the interface) elements for \( p \), are used on triangular grids.

4.6. Overall procedure for rising drop

For the implementation of the rising drop, we formulate the Navier–Stokes equations in an accelerated coordinate system, where the center of origin is fixed to the barycenter of the drop (see [26, 19]). Then the solution of the two–phase problem consists of the following steps: given the values \( \Omega^n, u^n, p^n, c^n, a^n \) at time instant \( t_n \), where \( a^n \) is the acceleration of the reference frame, we compute

**Step 1:** \( u^{n+1}, p^{n+1}, \Omega^{n+1} \) as described above by steps A–C,

**Step 2:** \( a^{n+1} \), the acceleration of the coordinate system as described in [26, 19],

**Step 3:** \( c^{n+1} \) by the described \textit{subspace projection method}.

When implementing the resulting finite element discretization, only minor changes to a standard implementation are required. Besides the \textit{node doubling}, outlined in Section 4.1, only a modification of the \textit{matrix vector multiplication} routines for iterative solvers is required (Section 4.3).
Remark 4 (Effect of the SPM on the iterative solver routines). The system matrix $A$ has a block structure, one block for each fluid phase. In the case of the Navier–Stokes equations, the blocks are weighted with the phase dependent density quotient $\Lambda_j = \frac{\rho_j}{\rho_c}$ (see Eq. (1)). The projection routines incorporate the boundary conditions, resulting in a system matrix $\tilde{A} = P_1^* A P_H$ with a nontrivial kernel. In our experience the iterative GMRES - solver is well capable to deal with the nontrivial kernel.

5. Validation and some computational results

This section is devoted to validate our method with respect to grid effects as well as comparison to data from literature. The method has been implemented in the academic code NAVIER [30, 40, 19].

5.1. Grid and domain dependencies

To validate the discretization with respect to grid dependency and thus also optimize the computational effort by choosing an appropriate grid, we present some domain and mesh dependency tests.

5.1.1. Wall effects

The first test is devoted to so called wall effects, i.e. the effect of the size of the (finite) computational domain.

To this end we consider a 4 mm deformable drop (Eqs. (1) and (2)) in the ternary system water/toluene/acetone. Parameters for this setting are found in Table 1. The simulations start from rest and run until the drop reaches its terminal velocity.

<table>
<thead>
<tr>
<th></th>
<th>water dispersed phase</th>
<th>cyclohexanol continuous phase</th>
<th>acetic acid transferring solute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density $\rho \left[ \frac{kg}{m^3} \right]$</td>
<td>993</td>
<td>954</td>
<td></td>
</tr>
<tr>
<td>Viscosity $\mu \left[ Pas \right]$</td>
<td>9.750E-04</td>
<td>1.635E-02</td>
<td></td>
</tr>
<tr>
<td>Diffusion coefficient $D \left[ \frac{m^2}{s} \right]$</td>
<td>1.24E-09</td>
<td>1.24E-09</td>
<td></td>
</tr>
<tr>
<td>surface tension $\gamma \left[ \frac{N}{m} \right]$</td>
<td></td>
<td></td>
<td>3.4E-3</td>
</tr>
<tr>
<td>Henry coefficient $H \left[ - \right]$</td>
<td></td>
<td></td>
<td>0.4</td>
</tr>
</tbody>
</table>

Table 1: Parameters for numerical simulation.

To identify wall effects, we compare the terminal velocity of the drop on different computational domains, with diameter $G \in \{4, 8, 12, 16\}$ (see Figure 4). The results are given in Table 2.
$G \in \{4, 8, 12, 16\}$

Figure 4: Geometry of the computational domain.

<table>
<thead>
<tr>
<th>Domain diameter</th>
<th>Terminal velocity $\frac{m}{s}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G = 4$</td>
<td>0.16718</td>
</tr>
<tr>
<td>$G = 8$</td>
<td>0.17334</td>
</tr>
<tr>
<td>$G = 12$</td>
<td>0.17398</td>
</tr>
<tr>
<td>$G = 16$</td>
<td>0.17400</td>
</tr>
</tbody>
</table>

Table 2: Rise velocity of a 4mm toluene droplet for different domain diameters. Parameters are set according to Table 1.

For the smallest domain ($G = 4$) the terminal velocity $u_{t,G=4}$ deviates 4% compared to the terminal velocity of the largest domain $u_{t,G=16}$, measured as the relative error. The domain with $G = 12$ deviates less than 0.01% to $G = 16$. We conclude that using a computational domain with $G = 12$, wall effects can be neglected when considering terminal velocities.

5.1.2. Boundary layers

Next, also boundary layers need to be considered when choosing a suitable grid for a two-phase flow application. In the problem from Section 2 boundary layers occur at the interface of the drop, and close to the axis of symmetry in downstream direction. The widths of these boundary layers depend on the Reynolds and Peclet numbers and in many applications a resolution of the boundary layers is not possible, due to computational limitations.
Figure 5: Detail of grids with different interface resolutions.

For the fluid dynamical investigation we choose a setting with \( Re \approx 700 \) for \( d_p = 4 \text{ mm} \), which corresponds to the parameters of the water/toluene two–phase flow, given in Table 1.

Comparison of the terminal velocities for different interface resolutions reveals a deviation of 0.15\% between mesh size (at the interface) \( h_2 = 0.013 \), and mesh size \( h_3 = 0.007 \) (the results are given in Table 3, details of the corresponding grids are displayed in Figure 5.) The macro triangulation has been generated using the software “Triangle” (described in [41]), and was a–priori refined.

<table>
<thead>
<tr>
<th></th>
<th>( h_1 = 0.026 )</th>
<th>( h_2 = 0.013 )</th>
<th>( h_3 = 0.007 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal velocity ( \frac{m}{s} )</td>
<td>0.1739</td>
<td>0.1767</td>
<td>0.1769</td>
</tr>
<tr>
<td>CPU-time [s]</td>
<td>( 0.126E + 06 )</td>
<td>( 0.164E + 06 )</td>
<td>( 0.100E + 07 )</td>
</tr>
</tbody>
</table>

Table 3: Rise velocity of a 4\text{mm} toluene droplet for different refinement levels and \( G = 12 \). Parameters are set according to Table 1.

As a more sensitive quantity, the interfacial velocity is visualized in Figure 6. The results reveal a relative error of 9.4\% between refinement levels 1 and 3, and a relative error of 3.6\% comparing refinement levels 2 and 3.

To approve the grid refinement, we investigate the mass transfer rate in terms of the dimensionless mean Sherwood number \( Sh := \int \frac{\partial \rho_c}{r} \). The equations are (1), (2) together with (3) and (4).

Results for water/acetic–acid/cyclohexanol setting and a 2.5 \text{mm} droplet with \( Re \approx 30 \) and \( Pe \approx 10^4 \) are given in Table 4. Between refinement levels 1 and 2 the mean Sherwood
Figure 6: Magnitude of velocity $|\mathbf{u}|$ at the interface at steady state of a 4mm toluene drop in water for grids with different interface resolutions. The azimuthal angle $\theta = 0$ corresponds to the south pole of the drop. Refinement level 1 corresponds to $h = 0.026$, refinement level 2 corresponds to $h = 0.013$ and refinement level 3 corresponds to $h = 0.007$. Parameters are set according to Table 1.

number nearly doubles. This corresponds to the estimate of the characteristic boundary layer width $\delta$ at the interface $\delta \approx \sqrt{\frac{1}{Pe}} \approx 0.01$. Refinement level 2 deviates only about 1.1% to refinement level 3.

<table>
<thead>
<tr>
<th></th>
<th>$h_1 = 0.026$</th>
<th>$h_2 = 0.013$</th>
<th>$h_3 = 0.007$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean Sherwood number at $T = 1.5$ s [-]</td>
<td>19.16</td>
<td>45.27</td>
<td>45.78</td>
</tr>
<tr>
<td>mean concentration at $T = 1.5$ s [-]</td>
<td>0.848917508</td>
<td>0.8675685072</td>
<td>0.86760318546</td>
</tr>
</tbody>
</table>

Table 4: Mass transfer of 2.5 mm droplet in the ternary system water/acetic–acid/ cyclohexanol for different refinement levels of the triangulation. Parameters are set according to Table 1.

As a validation for the mean concentration in the drop, the simulation result is compared to analytical results by Calderbank & Korchinski [42], Kronig & Brink [43] and Brauer [44] and experimental data and Star-CD simulation from Schulze [45] in Figure 7.

5.1.3. Spurious oscillations

In the numerical simulation of two–phase flow problems the jump of the pressure at the interface, if not depicted in the discrete solution, and the inexact numerical representation of the interface may lead to so called spurious velocities, see [28, 29, 46]. In the finite element
or finite volume context, the *ghost fluid* method [47, 48], or the extended finite element method (XFEM) [28, 49, 50, 51] have been developed to overcome this problem. In the XFEM method, the grid is not fitted to the interface in general. The finite-element space is enriched by functions with a discontinuity across the interface. To enforce the interface conditions a penalty method, Lagrange multipliers or Nitsche’s method are used in general (for an overview see [50]).

In contrast, one may consider our approach as two distinct finite element spaces for each bulk phase, which are connected via the incorporation of interface conditions by the subspace projection method. Discontinuous functions are naturally included in the projected finite element space and the solution fulfills the prescribed jump conditions at the interface exactly. Due to the sharp interface method, a reconstruction of the interface is not necessary, as the interface is at hand as isoparametric $P_2$-element edges. These two features diminish spurious oscillations considerably.

Figure 8 shows the result for three different grids for steady state and zero gravity in the toluene/water system, where the exact solution of the Young-Laplace equation is given by $u = 0$, and $[p] = \frac{1}{W_e} \kappa$. The initial grid is chosen very coarse with the interface consisting of 10 edges (21 nodes) only. Evaluation at a fixed time, after 100 time steps, shows spurious velocities in the order of magnitude of only $10^{-4}[\text{m/s}]$ on the coarsest grid with mesh size (at the interface) $\approx 0.157$. These parasitic currents further decrease down to $10^{-6}[\text{m/s}]$ on the grid twice refined with mesh size $\approx 0.039$.

6. Simulation of a stagnant cap angle with the SPM

As mentioned above, the subspace projection method is a general framework for the implementation of interface (or other) conditions in the finite element context, where the
Figure 8: Spurious velocities for a 3 mm toluene drop in water. Right part displays the magnitude of the velocity in [m/s], left part displays the grid. Parameters are set according to Table 5.

solution lies in a subspace of the general finite element space. In Section 3 we presented the subspace projection method in the context of a two–phase flow problem for the implementation of interface conditions of the Navier–Stokes equations and species transport equation. Here, we present the application of the SPM for modeling a stagnant cap, which may arise when surfactants are present, see eqs. (1), 6 and (8).

6.1. Simulation of stagnant cap

Figure 9 displays the stationary flow field of a spherical 4 mm toluene drop rising in water for different stagnant cap angles $\alpha$. The stagnant cap angle $\alpha = 0$ corresponds to the case $\Gamma_s = \emptyset$ and $\alpha = \pi$ corresponds to $\Gamma_s = \Gamma$. The parameters for the simulations are given in Table 5.

<table>
<thead>
<tr>
<th>sodium dodecyl sulphate (SDS) surfactant</th>
<th>toluene dispersed phase</th>
<th>water continuous phase</th>
<th>artificial substance transferring solute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density $\varrho [kg/m^3]$</td>
<td>862.3</td>
<td>997.02</td>
<td></td>
</tr>
<tr>
<td>Viscosity $\mu [Pas]$</td>
<td>5.52e-4</td>
<td>8.903e-4</td>
<td></td>
</tr>
<tr>
<td>Diffusion coefficient $D [m^2/s]$</td>
<td>2.90e-7</td>
<td>1.25e-7</td>
<td></td>
</tr>
<tr>
<td>Henry coefficient $H [-]$</td>
<td></td>
<td></td>
<td>0.63</td>
</tr>
</tbody>
</table>

Table 5: Parameters. The surface tension has been set according to Wegener [52].

In the case of a 4 mm drop (Figure 9) an additional rear flow vortex appears for $\alpha \geq 0.75$. Its size increases with increasing stagnant cap angle $\alpha$. Additional vortices appear both inside the drop and in the rear flow area.

For stagnant cap angles $\alpha > 2$, the magnitude of the velocity observed inside the drop is significantly smaller than the characteristic velocity of the drop.
In Figure 9 the terminal rise velocity of a 4 mm toluene drop in water is displayed. It decreases for increasing stagnant cap angles.

Species transport has been simulated for an artificial substance with parameters given in Table 5. For increasing stagnant cap angles the solute concentration is decelerated, as observed in Figure 11, where at a fixed time the solute is plotted inside the droplet. The temporal development of the mean solute concentration inside the drop is displayed in Figure
Figure 10: Terminal velocity of 4 mm toluene drop in water for varying stagnant cap angles. The parameters are set according to Table 5.

(a) $\alpha = 0$
(b) $\alpha = 1$
(c) $\alpha = 1.5$
(d) $\alpha = 2$
(e) $\alpha = 2.5$
(f) $\alpha = 3$

Figure 11: Concentration of solutant in drop with stagnant cap angle $\alpha$ at time $T = 0.16$ s for a 4 mm drop. The parameters for the simulations are given in Table 5.
As expected the mean concentration decreases faster for smaller stagnant cap angles.

Figure 12: Time evolution of mean concentration in drop for varying stagnant cap angles for 4 mm drop. The parameters are set according to Table 5.

Conclusion

We presented a sharp interface finite element method for the simulation of two-phase incompressible flows with a transport equation for a soluted species. The main features of the method comprise

• An isoparametric \( P_2 \) front tracking method to capture the free surface;
• a co-moving frame of reference, minimizing mesh distortion effects;
• a variational formulation of the curvature term;
• a subspace projection method to deal with discontinuities at the interface, and to incorporate the interface conditions.

The subspace projection method has been proven to be a versatile tool for the implementation of interfacial conditions in general two-phase flow problems including interfacial conditions for species transport as well as interfacial conditions for the fluid field. Different physical models (spherical drop, deformable drop, stagnant cap) may be simulated with one code and a simple switch between different subspaces - or in the implementation - different projection routines. Thorough validation experiments for a dispersed single drop rising in a continuous surrounding phase showed good agreements with data from literature, both experimental as well as numerical results.
Acknowledgement

The authors like to acknowledge the German Research Foundation (DFG) for financial support under project GZ BA 1727/6.


28


[41] J. R. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in: M. C. Lin, D. Manocha (Eds.), Applied Computational Geometry:


[52] M. Wegener, A. R. Paschedag, The effect of soluble anionic surfactants on rise velocity and mass transfer at single droplets in systems with Marangoni instabilities, J. Colloid Interface Sci. – (2011) –. Accepted for publication. 22
M. Prechtel, G. Leugering, P. Steinmann, M. Stingl: Towards optimization of crack resistance of composite materials by adjustment of fiber shapes

A. M. Khludnev, G. Leugering: Optimal control of cracks in elastic bodies with thin rigid inclusions

M. Prechtel, P. Leiva Ronda, R. Janisch, A. Hartmaier, G. Leugering, P. Steinmann, M. Stingl: Simulation of fracture in heterogeneous elastic materials with cohesive zone models

E. Marchand: Combined Deterministic-Stochastic Sensitivity Analysis; Application to Uncertainty Analysis.

G. Eichfelder, T.X.D. Ha: Optimality conditions for vector optimization problems with variable ordering structures


M. Kaiser, K. Klarmroth, A. Thekale: Test examples for nonlinear feasibility problems with expensive functions

J. Jahn, T.X.D. Ha: New Order Relations in Set Optimization

G. Eichfelder, J. Povh: On reformulations of nonconvex quadratic programs over convex cones by set-semidefinite constraints

T.X.D. Ha and J. Jahn: Properties of Bishop-Phelps Cones

N. Ray, Ch. Eck, A. Muntean, P. Knabner: Variable Choices of Scaling in the Homogenization of a Nernst-Planck-Poisson Problem

A. Muntean, T. L. Van Noorden: Corrector estimates for the homogenization of a locally-periodic medium with areas of low and high diffusivity


F. Brunner, F. A. Radu, M. Bause, P. Knabner: Optimal order convergence of a modified $BDM_1$ mixed finite element scheme for reactive transport in porous media

G. Eichfelder: Cone-valued maps in optimization

G. Eichfelder, J. Povh: On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets

S. Kräutle: Existence of global solutions of multicomponent reactive transport problems with mass action kinetics in porous media


J. Hoffmann, S. Kräutle, P. Knabner: A general reduction scheme for reactive transport in porous media

S. Kräutle: Existence of solutions of reactive transport problems with mass action kinetics and species-dependent diffusion on large time intervals

K. Bäumler, E. Bänisch: A subspace projection method for the implementation of interface conditions in a two-phase flow problem