Characteristic Cut Finite Element Methods for Convection-Diffusion Problems on Time Dependent Surfaces

Peter Hansbo[a], Mats G. Larson[b], Sara Zahedi[c]

[a] *Department of Mechanical Engineering, Jönköping University, SE-551 11 Jönköping, Sweden*
[b] *Department of Mathematics and Mathematical Statistics, Umeå University, SE–901 87 Umeå, Sweden*
[c] *Department of Information Technology, Uppsala University, Box 337, SE–751 05 Uppsala, Sweden.*

Abstract

We develop a finite element method for convection diffusion problems on a given time dependent surface, for instance modeling the evolution of a surfactant. The method is based on a characteristic-Galerkin formulation combined with a piecewise linear cut finite element method in space. The cut finite element method is constructed by embedding the surface in a background grid and then using the restriction to the surface of a finite element space defined on the background grid. The surface is allowed to cut through the background grid in an arbitrary fashion. To ensure well posedness of the resulting algebraic systems of equations, independent of the position of the surface in the background grid, we add a consistent stabilization term. We prove error estimates and present confirming numerical results.

Keywords: cut finite element method, surfactants, PDEs on surfaces, characteristic Galerkin method

1. Introduction

Surfactants are important because of their ability to reduce the surface tension and are for example used in detergents, oil recovery, and in treatment of lung diseases [1]. In order to perform realistic simulations, development of numerical methods that accurately and efficiently solve the partial differential equation (PDE) for the evolution of the concentration of surfactants on moving and deforming interfaces is necessary. The effort put in to this reserach area has therefore been extensive.

In order to solve for the concentration of surfactants on a moving interface there is a need to represent the interface. Techniques to represent the interface can be roughly divided into two classes: 1) explicit representation, e.g. by marker particles, and 2) implicit representation, e.g. by the level set of a higher dimensional function. In general, strategies for solving quantities on evolving surfaces are developed on the basis of the interface representation technique. Therefore, existing methods are usually tightly coupled to the interface representation, and several methods have been proposed both based on explicit [2–4] and implicit [5–10] representations. Implicit representation techniques have the benefit that it is usually not more complicated to solve the equations in three space dimensions compared to two space dimensions. However, the quantity on the interface is often extended off the interface and given by the solution of a PDE on a higher dimension and precautions have therefore to be taken in order to ensure good conservation of the surfactant mass.

Recently, a finite element method for the Laplace–Beltrami operator, where the surface is embedded into a three dimensional background grid and the finite element space on the surface

Email addresses: Peter.Hansbo@jth.hj.se (Peter Hansbo), mats.larson@math.umu.se (Mats G. Larson), sara.zahedi@it.uu.se (Sara Zahedi)

is the restriction of a finite element space defined on the background grid, was proposed in [11]. We call this type of method cut finite elements since the surface cuts through the background grid without restrictions. The main advantage of this approach is that it is very convenient when solving problems that involve coupled physical phenomena in the bulk domain and on surfaces or interfaces since the same finite element spaces can be used for all problems. Furthermore, it can be used with both explicit and implicit surface representations. In case of an implicit surface representation, the same background grid may be used both for the surface representation and for the problem on the surface. A drawback of this type of methods is that the stiffness matrix may become arbitrarily ill conditioned depending on the position of the surface in the background mesh. In the case of the Laplace–Beltrami operator the ill conditioning can be handled using a scaling, see [12]. Another alternative, which may be more suitable for more complex problems, is to add stabilization terms, see [13] and [14].

In this paper we extend the cut finite element method to a time dependent convection diffusion equation on a given evolving surface governing the evolution of the surfactant concentration on a moving surface [15, 16]. The strategy is to combine the characteristic Galerkin scheme [17] with a stabilized version of the cut finite element method. The method is stable and yields a linear algebraic system of equations with bounded condition number independently on how the interface cuts the background grid. We use linear finite elements in space and a second order characteristic Galerkin scheme, which results in first order convergence in the L_2 norm when the timestep k is proportional to the spatial mesh size h. In addition, the total mass of surfactant can be accurately conserved using a Lagrange multiplier. The proposed finite element method is straightforward and works both with explicit and implicit interface representation techniques.

The remainder of the paper is organized as follows. In Section 2 we formulate the convection diffusion equation. In Section 3 we present the finite element method and discuss the accuracy of the method. Finally, in Section 4 we show numerical examples in two space dimensions.

2. The Continuous Problem

2.1. The Surface

Let $I = [0, T]$ be a time interval and let $\Sigma(t)$ be a smooth closed $d-1$ dimensional surface embedded in $\mathbf{R}^d$, with $d = 2$ or 3, which evolves smoothly in time without any self intersection.

Let $\Omega(t)$ be a neighborhood of $\Sigma(t)$ and let $\boldsymbol{p}(t, \boldsymbol{x}) : I \times \Omega(t) \to \Sigma(t)$ be the closest point mapping

$$\boldsymbol{p}(t, \boldsymbol{x}) = \text{argmin}_{\boldsymbol{y} \in \Sigma(t)} |\boldsymbol{y} - \boldsymbol{x}| \tag{2.1}$$

i.e. the point $\boldsymbol{y}$ on $\Sigma(t)$ that minimizes the Euclidian distance $|\boldsymbol{y} - \boldsymbol{x}|$ to $\boldsymbol{x}$. We note that for $\Omega(t)$ small enough this mapping is well defined. We define the signed distance function as $\rho(t, \boldsymbol{x}) = |\boldsymbol{x} - \boldsymbol{p}(t, \boldsymbol{x})|$ for $\boldsymbol{x}$ on the outside of the surface and $\rho(t, \boldsymbol{x}) = -|\boldsymbol{x} - \boldsymbol{p}(t, \boldsymbol{x})|$ for $\boldsymbol{x}$ in the inside of the surface. The exterior unit normal $\boldsymbol{n} = \boldsymbol{n}(t, \boldsymbol{x})$ is the spatial gradient of the signed distance function, $\boldsymbol{n}(t, \boldsymbol{x}) = \nabla \rho(t, \boldsymbol{x})$, for $\boldsymbol{x} \in \Sigma(t)$.

2.2. Surface Gradient and Divergence

We shall define the surface gradient as the projection onto the tangent plane of the $\mathbf{R}^d$ gradient of an extension of the function from the surface to a neighborhood of the surface. Using the closest point mapping we can extend a function $v : \Sigma(t) \to \mathbf{R}$ from $\Sigma(t)$ to $\Omega(t)$ by defining $v(\boldsymbol{x}) = v \circ \boldsymbol{p}(x)$. The tangent gradient ∇_Σ on $\Sigma(t)$ of a scalar valued function v is then defined as follows

$$\nabla_\Sigma v = \boldsymbol{P} \nabla v \tag{2.2}$$

where ∇ denotes the usual $\mathbf{R}^d$ gradient and $\boldsymbol{P} = \boldsymbol{P}(\boldsymbol{x})$ is the projection of $\mathbf{R}^d$ onto the tangent plane of $\Sigma(t)$ at $\boldsymbol{x} \in \Sigma(t)$, defined by

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n} \tag{2.3}$$

where the tensor product is defined by $(\boldsymbol{a} \otimes \boldsymbol{b})_{ij} = a_i b_j$ for any two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The tangent gradient $\mathrm{D}_\Sigma \boldsymbol{v}$ and divergence $\mathrm{div}_\Sigma \boldsymbol{v}$ on $\Sigma(t)$ of a vector valued function $\boldsymbol{v}$ is defined by

$$\mathrm{D}_\Sigma \boldsymbol{v} = \boldsymbol{v} \otimes \nabla_\Sigma, \qquad \mathrm{div}_\Sigma \boldsymbol{v} = \mathrm{tr}(\mathrm{D}_\Sigma \boldsymbol{v}) \tag{2.4}$$

where $(\boldsymbol{v} \otimes \nabla_\Sigma)_{ij} = (\nabla_\Sigma)_j v_i$.

We let $L^2(\Sigma(t))$ denote the space of square integrable functions $v : \Sigma(t) \to \mathbf{R}$ with scalar product $(v, w)_{\Sigma(t)} = \int_{\Sigma(t)} vw$ and norm $\|v\|^2_{\Sigma(t)} = \int_{\Sigma(t)} v^2$. Further, we let $H^m(\Sigma(t)), m = 1, 2$, be the Sobolev spaces with norm $\|v\|^2_{m,\Sigma(t)} = \sum_{k=0}^m \|(\otimes_{l=0}^k \mathrm{D}_\Sigma)v\|^2_{\Sigma(t)}$.

2.3. Governing Equations

We assume that there is a smooth vector field $\boldsymbol{\beta} : I \times \Sigma \to \mathbf{R}^d$ such that the associated characteristics $\boldsymbol{X}(t) = \boldsymbol{X}(t; s, \boldsymbol{x}_s)$, defined as solutions to the ordinary differential equation

$$\partial_t \boldsymbol{X} = \boldsymbol{\beta}(t, \boldsymbol{X}) \quad t \in I, \qquad \boldsymbol{X}(s) = \boldsymbol{x}_s \tag{2.5}$$

with $0 \le s, t \le T$, satisfy

$$\boldsymbol{X}(t; s, \boldsymbol{x}_s) \in \Sigma(t) \quad \forall t \in I, \ \forall \boldsymbol{x}_s \in \Sigma(s) \tag{2.6}$$

In other words, the characteristics are curves on the spacetime manifold Σ. From smoothness it then follows that $\Sigma(t) = \boldsymbol{X}(t; s, \Sigma(s)) = \cup_{\boldsymbol{x}_s \in \Sigma(s)} \boldsymbol{X}(t; s, \boldsymbol{x}_s)$.

We now consider the following time dependent convection diffusion problem: find $u : I \times \Sigma \to \mathbf{R}$ such that

$$\partial_t u + \boldsymbol{\beta} \cdot \nabla u + (\mathrm{div}_\Sigma \boldsymbol{\beta}) u - \mathrm{div}_\Sigma \left(\epsilon \nabla_\Sigma u\right) = f \quad \text{on } I \times \Sigma(t) \tag{2.7}$$

with initial condition $u(0) = u_0$ on $\Sigma(0)$. Here $f \in L^2(I \times \Sigma)$ is a given function and ϵ is a given positive constant. This problem models the evolution of a surfactant on the given time dependent surface Σ, see for example [15, 18].

2.4. Weak Form

The weak form of (2.7) takes the form: find $u : I \to H^1(\Sigma(t))$ with $u(0) = u_0$ such that

$$(\partial_t u + \boldsymbol{\beta} \cdot \nabla u, v)_{\Sigma(t)} + a(u, v) = (f, v)_{\Sigma(t)} \quad \forall t \in I, \ v \in H^1(\Sigma(t)) \tag{2.8}$$

where we introduced the bilinear form

$$a(u, v) = ((\mathrm{div}_\Sigma \boldsymbol{\beta}) u, v)_{\Sigma(t)} + (\epsilon \nabla_\Sigma u, \nabla_\Sigma v)_{\Sigma(t)} \tag{2.9}$$

3. The Characteristic Cut Finite Element Method

We formulate a finite element method by combining the characteristic Galerkin method [17] for solving the convection diffusion equation (2.7) on a moving surface $\Sigma(t)$ with a cut finite element method in space. The characteristic Galerkin method is based on splitting the problem as follows:

- Step 1: Transport the solution from $\Sigma(t_{n-1})$ to $\Sigma(t_n)$ using an approximation of the characteristic equation.
- Step 2: Solve for the solution on $\Sigma(t_n)$ using the transported solution as initial data

In the following subsections we present discretizations in time and space for an exact surface representation and then we comment on a particular discrete surface representation based on a piecewise linear level set function. Finally, we show that the method is of first order.

3.1. Discretization in Time

Let $0 = t_0 < t_1 < \cdots < t_N = T$ be a partition of the time interval into time steps of length $k_n = t_n - t_{n-1}$ for $n = 1, 2, \ldots, N$.

For each time step we define the transport operator $T^n : L^2(\Sigma^{n-1}) \to L^2(\Sigma^n)$ by

$$T^n u^{n-1}(\boldsymbol{x}) = u^{n-1} \circ \boldsymbol{X}(t_{n-1}; t_n, \boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Sigma^n \tag{3.1}$$

where $\boldsymbol{X}(t; s, \boldsymbol{x}_s)$ is defined in (2.5) and we introduced the notation $\Sigma^n = \Sigma(t_n), n = 0, 1, \ldots, N$.

The characteristic discretization of (2.8) in time takes the following form: given $u_0 = u(0)$, we define the approximation $u^n \in H^1(\Sigma^n)$ of $u(t_n)$ for $n = 1, 2, \ldots, N$, by

$$(u^n - T^n u^{n-1}, v)_{\Sigma^n} + k_n a_n(u^n, v) = k_n (f, v)_{\Sigma^n} \quad \forall v \in H^1(\Sigma^n) \tag{3.2}$$

where

$$a_n(u, v) = ((\mathrm{div}_\Sigma \boldsymbol{\beta}) u, v)_{\Sigma^n} + (\epsilon \nabla_\Sigma u, \nabla_\Sigma v)_{\Sigma^n} \tag{3.3}$$

In practice, we compute an approximation T_h^n of T^n as follows

$$T_h^n u^{n-1}(\boldsymbol{x}) = u^{n-1} \circ \boldsymbol{X}_h^n(t_{n-1}; t_n, \boldsymbol{x}) \tag{3.4}$$

where $\boldsymbol{X}_h^n(t_{n-1}; t_n, \boldsymbol{x})$ is defined by

$$\boldsymbol{X}_h^n(t_{n-1}; t_n, \boldsymbol{x}) = \boldsymbol{p}(t_{n-1}, \boldsymbol{x} - k_n \boldsymbol{\beta}(t_{n-1} + \frac{k_n}{2}, \boldsymbol{x}_m)) \quad \boldsymbol{x}_m = \boldsymbol{x} - \frac{k_n}{2} \boldsymbol{\beta}(t_n, \boldsymbol{x}) \tag{3.5}$$

and $\boldsymbol{p}(t, \cdot)$ is the closest point mapping defined in (2.1). We note that the approximation is of second order

$$|\boldsymbol{X}(t_{n-1}; t_n, \boldsymbol{x}) - \boldsymbol{X}_h^n(t_{n-1}; t_n, \boldsymbol{x})| \leq C k_n^2, \quad \forall x \in \Sigma^n, n = 1, 2, \ldots, N \tag{3.6}$$

3.2. Discretization in Space

Let $\mathcal{K}_h = \{K\}$ be a partition of a domain Ω_I in $\mathbf{R}^d$ completely containing $\cup_{t \in I} \Sigma(t)$ into shape regular triangles for $d = 2$ and tetrahedra for $d = 3$ of diameter h_K. Let $\mathcal{F}_h$ denote the set of faces belonging to elements in $\mathcal{K}_h$. For each $n = 0, 1, \ldots, N$ we define the set of elements $\mathcal{K}_h^n$ that intersect Σ^n by

$$\mathcal{K}_h^n = \{K \in \mathcal{K}_h : \overline{K} \cap \Sigma^n \neq \phi\} \tag{3.7}$$

Let $\Omega_h^n = \cup_{K \in \mathcal{K}_h^n} K$ be the union of all elements in $\mathcal{K}_h^n$ and denote the set of interior faces in $\mathcal{K}_h^n$, i.e. faces shared by two elements K^+ and K^- in $\mathcal{K}_h^n$, by

$$\mathcal{F}_h^n = \{F \in \mathcal{F}_h : F = K^+ \cap K^-, K^+, K^- \in \mathcal{K}_h^n\} \tag{3.8}$$

See Fig. 1 for an illustration of the set $\mathcal{F}_h^n$ and the domain Ω_h^n.

We let $\mathcal{V}_h$ be the space of continuous piecewise linear polynomials defined on $\mathcal{K}_h$ and each timestep we define the finite element space

$$\mathcal{V}_h^n = \mathcal{V}_h|_{\Omega_h^n}, \quad n = 0, 1, \ldots, N \tag{3.9}$$

i.e. the space of restrictions to Ω_h^n of functions in $\mathcal{V}_h$.

The finite element method reads: given $u_h^0 \in \mathcal{V}_h^0$ find $u_h^n \in \mathcal{V}_h^n$, for $n = 1, 2, \ldots, N$, such that

$$\left(u_h^n - T_h^n u_h^{n-1}, v\right)_{\Sigma^n} + k_n a_n(u_h^n, v) + j_n(u_h^n, v) = k_n (f, v)_{\Sigma^n} \quad \forall v \in \mathcal{V}_h^n \tag{3.10}$$

Here we have added the stabilization term

$$j_n(u, v) = \sum_{F \in \mathcal{F}_h^n} (c_1 k_n \epsilon + c_2 (1 + k_n |\mathrm{div}_\Sigma \boldsymbol{\beta}|) h_F^2) \left([\![\boldsymbol{n}_F \cdot \nabla u]\!]_F, [\![\boldsymbol{n}_F \cdot \nabla v]\!]_F \right)_F \tag{3.11}$$

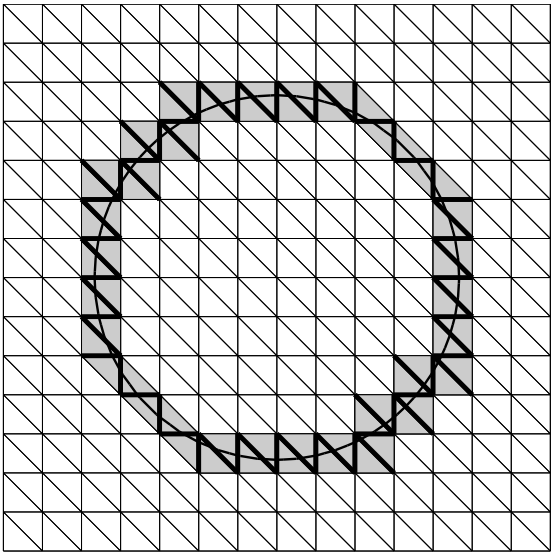

Figure 1: Illustration of the interface Σ, the domain Ω_h^n, and the edges in $\mathcal{F}_h^n$ in 2D. A triangulation of a domain Ω_I completely containing the circle Σ is shown. The domain Ω_h^n consists of all the shaded triangles. The edges in $\mathcal{F}_h^n$ are marked with a thick line.

where c_1 and c_2 are parameters, $h_F = (h_{K^+} + h_{K^-})/2$ is the average of the sizes of the elements K^+ and K^- sharing the face F, and

$$[\![v]\!]_F = v^+ - v^- \tag{3.12}$$

with $v^\pm(\boldsymbol{x}) = \lim_{t\to 0^+} v(\boldsymbol{x} \mp t\boldsymbol{n}_F)$ for $\boldsymbol{x} \in F$ and $\boldsymbol{n}_F$ a fixed unit normal to F, denotes the jump in a discontinuous function v across F. The stabilization term is added to control the condition number of the resulting algebraic system of equations. Without stabilization the linear system of equations may become arbitrarily ill conditioned depending on the position of the surface relative to the mesh. We refer to [13] and [14] for estimates of the condition number for various stabilization terms.

3.3. Discretization of the Surface

The surface $\Sigma(t)$ is in general not available exactly, instead we have to use some kind of discrete representation Σ_h of Σ. Our method is independent of the particular type of representation of the surface. To formulate a completely discrete method we need to compute the exterior unit normal $\boldsymbol{n}_h$ to the discrete surface, the nearest point mapping

$$\boldsymbol{p}_h(t, \boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{y}\in\Sigma_h(t)} |\boldsymbol{y} - \boldsymbol{x}| \tag{3.13}$$

onto the discrete surface $\Sigma_h(t)$, and we also need a quadrature rule to compute the integrals over Σ_h appearing in the method.

A common choice in the case of evolving surfaces is to use a level set approach [19, 20], for instance a discrete level set function may be constructed by defining $\rho_h = \pi_h\rho$ where ρ is the distance function, see Section 2.1, and $\pi_h\rho$ is the piecewise linear Lagrange interpolant of ρ. We note that the zero level set Σ_h of ρ_h is piecewise linear where the facets may be triangles or quadrilaterals. We define the discrete normal $\boldsymbol{n}_h = \nabla\rho_h$ and the discrete closest point mapping is given by $\boldsymbol{p}_h(t, \boldsymbol{x}) = \boldsymbol{x} - \rho_h(\boldsymbol{x})\boldsymbol{n}$.

Another alternative, not used in this paper, is a parametric representation where we have access to a local or piecewise parametrization, $\boldsymbol{x}(t, \xi, \eta)$ where ξ and η are the parameters, of the surface. In this case the normal and the nearest point mapping can be computed using the parametrization.

3.4. Error Estimates

Let us, for simplicity, only consider the pure transport problem: find $u : I \times \Sigma \to \mathbf{R}$ such that

$$\partial_t u + \boldsymbol{\beta} \cdot \nabla u + (\mathrm{div}_\Sigma \boldsymbol{\beta}) u = f \quad \text{on } I \times \Sigma(t) \tag{3.14}$$

with intial conditions $u(0) = u_0$ on $\Sigma(0)$. In this case the exact solution is given by

$$u(t, \boldsymbol{x}) = u(t_{n-1}, \boldsymbol{X}(t_{n-1}; t, \boldsymbol{x}))e^{-B(t;t,\boldsymbol{x})} + \int_{t_{n-1}}^{t} e^{(B(s;t,\boldsymbol{x})-B(t;t,\boldsymbol{x}))} f(s, \boldsymbol{X}(s; t, \boldsymbol{x}))ds \tag{3.15}$$

where $B(s) = B(s; t, \boldsymbol{x}) = \int_{t_{n-1}}^{s} \mathrm{div}_\Sigma \boldsymbol{\beta}(\boldsymbol{X}(\tau; t, \boldsymbol{x}))d\tau$. Furthermore, let us consider the numerical method (3.10) on the exact geometry Σ. The method then takes the form: given u_h^0 find u_h^n for $n = 1, 2, \ldots, N$ such that

$$(u_h^n, v)_{\Sigma^n} + k_n((\mathrm{div}_\Sigma \boldsymbol{\beta}) u_h^n, v)_{\Sigma^n} + j_n(u_h^n, v) = (T_h^n u_h^{n-1} + k_n f(t_n), v)_{\Sigma^n} \quad \forall v \in \mathcal{V}_h^n \tag{3.16}$$

We shall estimate the error in the stabilized L^2 norm

$$|||v|||_{\Sigma^n}^2 = \|v\|_{\Sigma^n}^2 + j_n(v, v) \tag{3.17}$$

Theorem 3.1. *Let u be the solution to (3.14) and extend it from $\Sigma(t)$ by $u(\boldsymbol{x}) = u \circ \boldsymbol{p}(x)$ and let $u_h^n, n = 0, 1, \ldots, N$ be defined by (3.16). If there is a constant c such that $\max_{1 \le n \le N} k_n \| div_\Sigma \boldsymbol{\beta} \|_{L^\infty(\Sigma^n)} \le c < 1$. Then the following error estimate holds*

$$|||u(T) - u_h^N|||_{\Sigma(T)} \le CN(k^2 + h^2)\Big(C + \max_{1 \le n \le N}(\|u(t_{n-1})\|_{\Sigma^{n-1}} + \|u(t_n)\|_{2,\Sigma^n})\Big) \tag{3.18}$$

where $k = \max_{1 \le n \le N} k_n$ and $h = \max_{K \in \mathcal{K}_h} h_K$.

Remark 3.1. *We note that for a uniform timestep, $N = T/k$ and we have $CN(k^2 + h^2) = CT(k + h^2/k)$ and thus for $k = Ch$, Theorem 3.1 gives the first order error estimate*

$$|||u(T) - u_h^N|||_{\Sigma(T)} \le Ch \tag{3.19}$$

Proof. We first split the error as follows

$$|||u(t_n) - u_h^n|||_{\Sigma^n}^2 = |||u(t_n) - P_n u(t_n)|||_{\Sigma^n}^2 + |||P_n u(t_n) - u_h^n|||_{\Sigma^n}^2 = I + II \tag{3.20}$$

where $P_n : L^2(\Sigma^n) \to \mathcal{V}_h^n$ is the stabilized L^2 projection defined by

$$(P_n u, v)_{\Sigma^n} + j_n(P_n u, v) = (u, v)_{\Sigma^n} \quad \forall v \in \mathcal{V}_h^n \tag{3.21}$$

Term $\boldsymbol{I}$. Using equation (3.21) and Cauchy-Schwarz inequality we have

$$\begin{aligned} I = |||u - P_n u|||_{\Sigma^n}^2 &= (u - P_n u, u)_{\Sigma^n} + (P_n u - u, P_n u)_{\Sigma^n} + j_n(P_n u, P_n u) = \\ &= (u - P_n u, u - \pi_h^n u)_{\Sigma^n} + (u - P_n u, \pi_h^n u)_{\Sigma^n} \le \\ &\le \|u - P_n u\|_{\Sigma^n} \|u - \pi_h^n u\|_{\Sigma^n} + j_n(P_n u, P_n u)^{1/2} j_n(\pi_h^n u, \pi_h^n u)^{1/2} \\ &\le 2|||u - P_n u|||_{\Sigma^n} |||u - \pi_h^n u|||_{\Sigma^n} \end{aligned} \tag{3.22}$$

where the interpolation operator $\pi_h^n : L^2(\Omega_h^n) \to \mathcal{V}_h^n$ is the usual Scott-Zhang interpolation operator, see [21]. For $u \in H^2(\Omega_h^n)$, using a trace inequality see [22] and the approximation property of the interpolation operator, we obtain

$$\begin{aligned} |||u - \pi_h^n u|||_{\Sigma^n}^2 &= \sum_{K \in \mathcal{K}_{\Sigma^n}} \|u - \pi_h^n u\|_{\Sigma \cap K}^2 + \sum_{F \in \mathcal{F}_h^n} Ch^2 \| [\![\boldsymbol{n}_F \cdot \nabla (u - \pi_h^n u)]\!]_F \|_F^2 \\ &\le \sum_{K \in \mathcal{K}_{\Sigma^n}} C_1 h^{-1} \|u - \pi_h^n u\|_K^2 + C_2 h \|u - \pi_h^n u\|_{1,K}^2 + C_3 h^3 \|u - \pi_h^n u\|_{2,K}^2 \\ &\le Ch^3 \|u\|_{2,\Omega_h^n}^2 \le Ch^4 \|u\|_{2,\Sigma^n}^2 \end{aligned} \tag{3.23}$$

which concludes the estimate of Term I.

Term $\boldsymbol{II}$. Denoting $P_n u(t_n) - u_h^n$ by e_h^n and using equation (3.21) we first note that

$$\begin{aligned} II &= |||e_h^n|||_{\Sigma^n}^2 = (e_h^n, e_h^n)_{\Sigma^n} + j_n(e_h^n, e_h^n) \\ &= (P_n u - u_h^n, e_h^n)_{\Sigma^n} + j_n(P_n u - u_h^n, e_h^n) = (u(t_n) - u_h^n, e_h^n)_{\Sigma^n} - j_n(u_h^n, e_h^n) \end{aligned} \tag{3.24}$$

Next using the solution formula (3.15) and the definition of the numerical method (3.16) we derive the following representation formula

$$\begin{aligned} II &= (u(t_n), e_h^n)_{\Sigma^n} - (u_h^n, e_h^n)_{\Sigma^n} - j_n(u_h^n, e_h^n) \\ &= (T^n u(t_{n-1}) e^{-B(t_n)}, e_h^n)_{\Sigma^n} + \int_{t_{n-1}}^{t_n} (e^{(B(s)-B(t_n))} f(s), e_h^n)_{\Sigma^n} ds \\ &\quad - (T_h^n u_h^{n-1} + k_n f(t_n), e_h^n)_{\Sigma^n} + k_n((\mathrm{div}_\Sigma \boldsymbol{\beta}) u_h^n, e_h^n)_{\Sigma^n} \\ &= (T^n u(t_{n-1})(e^{-B(t_n)} - 1 + k_n(\mathrm{div}_\Sigma \boldsymbol{\beta}), e_h^n)_{\Sigma^n} \\ &\quad + (T^n u(t_{n-1}) - T_h^n u_h^{n-1}, v)_{\Sigma^n} \\ &\quad + k_n((\mathrm{div}_\Sigma \boldsymbol{\beta})(u_h^n - T^n u(t_{n-1})), e_h^n)_{\Sigma^n} \\ &\quad + \int_{t_{n-1}}^{t_n} (e^{(B(s)-B(t_n))} f(s), e_h^n)_{\Sigma^n} ds - (k_n f(t_n), e_h^n)_{\Sigma^n} \\ &= II_1 + II_2 + II_3 + II_4 \end{aligned} \tag{3.25}$$

We now proceed with estimates of these contributions.
Term II_1. Using Cauchy-Schwarz inequality and Taylors formula we get the estimate

$$\begin{aligned} II_1 &\leq \|T^n u(t_{n-1})(e^{-B(t_n)} - 1 + k_n(\mathrm{div}_\Sigma \boldsymbol{\beta}))\|_{\Sigma^n} \|e_h^n\|_{\Sigma^n} \\ &\leq C k_n^2 \|T^n\| \|u(t_{n-1})\|_{\Sigma^{n-1}} \|e_h^n\|_{\Sigma^n} \end{aligned} \tag{3.26}$$

where $\|T^n\| = \sup_{v \in L^2(\Sigma^{n-1})} \|T^n v\|_{L^2(\Sigma_n)} / \|v\|_{L^2(\Sigma^{n-1})}$.
Term II_2. Using the fact that we are using a second order approximation T_h^n of T^n we obtain

$$\begin{aligned} II_2 &= (T^n u(t_{n-1}) - T_h^n u_h^{n-1}, e_h^n)_{\Sigma^n} \\ &\leq ((T^n - T_h^n) u(t_{n-1}), e_h^n)_{\Sigma^n} + (T_h^n (u(t_{n-1}) - u_h^{n-1}), e_h^n)_{\Sigma^n} \\ &\leq C k_n^2 \|u(t_{n-1})\|_{\Sigma^{n-1}} \|e_h^n\|_{\Sigma^n} + \|T_h^n\| \|(u(t_{n-1}) - u_h^{n-1})\|_{\Sigma^{n-1}} \|e_h^n\|_{\Sigma^n} \end{aligned} \tag{3.27}$$

Term II_3. Using Cauchy-Schwarz inequality and Taylors formula we get the estimate

$$\begin{aligned} II_3 &= k_n((\mathrm{div}_\Sigma \boldsymbol{\beta})(u_h^n - T^n u(t_{n-1})), e_h^n)_{\Sigma^n} \\ &= k_n((\mathrm{div}_\Sigma \boldsymbol{\beta})(u_h^n - u(t_n)), e_h^n)_{\Sigma^n} \\ &\quad + k_n((\mathrm{div}_\Sigma \boldsymbol{\beta})(u(t_n) - T^n u(t_{n-1})), e_h^n)_{\Sigma^n} \\ &\leq k_n \|\mathrm{div}_\Sigma \boldsymbol{\beta}\|_{L^\infty(\Sigma^n)} \|u_h^n - u(t_n)\|_{\Sigma^n} \|e_h^n\|_{\Sigma^n} \\ &\quad + C k_n^2 \|u(t_{n-1})\|_{\Sigma^{n-1}} \|e_h^n\|_{\Sigma^n} \end{aligned} \tag{3.28}$$

Term II_4. Using Cauchy-Schwarz inequality and Taylors formula we get the estimate

$$II_4 = \int_{t_{n-1}}^{t_n} (e^{(B(s)-B(t_n))} f(s) - f(t_n), e_h^n)_{\Sigma^n} ds \leq C k_n^2 \|e_h^n\|_{\Sigma^n} \tag{3.29}$$

Collecting the estimates of I–II we obtain

$$\begin{aligned} (1 - k_n \|\mathrm{div}_\Sigma \boldsymbol{\beta}\|_{L^\infty(\Sigma^n)}) |||u(t_n) - u_h^n|||_{\Sigma^n} &\leq C h^2 \|u(t_n)\|_{2,\Sigma^n} + C k_n^2 \\ &\quad + k_n^2 (C\|T^n\| + C) \|u(t_{n-1})\|_{\Sigma^{n-1}} + C \|T_h^n\| \|u(t_{n-1}) - u_h^{n-1}\|_{\Sigma^{n-1}} \end{aligned} \tag{3.30}$$

for $n = 1, 2, \ldots, N$. Iterating this estimate we arrive at the desired result

$$
\begin{aligned}
|||u(t_N) - u_h^n|||_{\Sigma^n} &\leq \sum_{n=1}^{N} Ck_n^2(C + \|u(t_{n-1})\|_{\Sigma^{n-1}}) + Ch^2\|u(t_n)\|_{2,\Sigma^n} \\
&\leq CN(k^2 + h^2)\Big(C + \max_{1\leq n\leq N}(\|u(t_{n-1})\|_{\Sigma^{n-1}} + \|u(t_n)\|_{2,\Sigma^n})\Big) \qquad (3.31)
\end{aligned}
$$

□

4. Numerical Examples

We consider three examples where we solve for the concentration u of surfactants on moving interfaces. We expect the method (3.10) to be of first order, see Theorem 3.1 and Remark 3.1, and that the condition number of the algebraic system of equations is stable independently on how the interface cuts the elements. We use a uniform underlying mesh $\mathcal{K}_h$ consisting of triangles of size h with a constant time step of the form $k = Ch$. The stabilization constants c_1 and c_2 in the edge stabilization j_n are 10 in all the computations.

4.1. Time Dependent Surface Representation

In our numerical examples we use a level set representation of the geometry. We approximate the level set function ρ by $\rho_h \in \mathcal{V}_{h/2}$ where $\mathcal{V}_{h/2}$ is the space of piecewise linear continuous functions defined on the mesh $\mathcal{K}_{h/2}$ obtained by refining $\mathcal{K}_h$ uniformly once. $\Sigma^h(t)$ is the zero level set of $\rho_h(t)$. In the first two examples the level set function is given on closed form and ρ_h is obtained by nodal interpolation in $\mathcal{V}_{h/2}$, while in the third example we will have to solve a partial differential equation for the level set function.

Given a vector field $\boldsymbol{\beta}$ the evolution of the surface $\Sigma(t)$ is governed by the following problem for the level set function: find $\rho : I \times \Omega \to \mathbf{R}$ such that

$$
\rho_t + \boldsymbol{\beta} \cdot \nabla\rho = 0, \quad \rho(0) = \rho_0 \qquad (4.1)
$$

Using Crank-Nicholson in time and piecewise linear continuous finite elements with streamline diffusion stabilization in space we obtain the method: find $\rho_h^n \in \mathcal{V}_{h/2}$ such that, for $n = 1, 2, \ldots, N$,

$$
\begin{aligned}
&(\frac{\rho_h^n}{k_n}, v^n)_\Omega + \theta(\boldsymbol{\beta}^n \cdot \nabla\rho_h^n, v^n)_\Omega + (\frac{\rho_h^n}{k_n} + \theta\boldsymbol{\beta}^n \cdot \nabla\rho_h^n - g^{n-1}, \tau_{SD}\boldsymbol{\beta}^n \cdot \nabla v^n)_\Omega \quad \forall v_h^n \in \mathcal{V}_{h/2} \\
&g^{n-1} = \frac{\rho_h^{n-1}}{k_n} - (1-\theta)\boldsymbol{\beta}^{n-1} \cdot \nabla\rho_h^{n-1}, \quad \theta = \frac{1}{2} \qquad (4.2)
\end{aligned}
$$

where the streamline diffusion parameter τ_{SD} is $\tau_{SD} = 2(k_n^{-2} + |\boldsymbol{\beta}|^2 h^{-2})^{-1/2}$. To keep the level set function a signed distance function the reinitialization equation, equation (15) in [23], can be solved, which can be discretized in the same way as we did with the advection equation, see equation (4.2).

4.2. Example 1: Moving interface

We consider the problem given in Example 2 of [10]. Initially the interface Σ is a circle centered at the origin with radius $r_0 = 2$ and $\boldsymbol{\beta} = (1, 0)$. The initial concentration $u = y/r_0 + 2$ and $\epsilon = 1$. The time step $k = h/4$ as in [10]. The interface $\Sigma(t)$ and the concentration u on the moving interface at times $t = 0, 2, 4$ for the mesh size $h = 0.2$ are shown in Fig 2. In Fig. 3 we show the relative change of the total mass of surfactant

$$
\frac{\int_{\Sigma(t)} u(t) - \int_{\Sigma(0)} u(0)}{\int_{\Sigma(0)} u(0)} \qquad (4.3)
$$

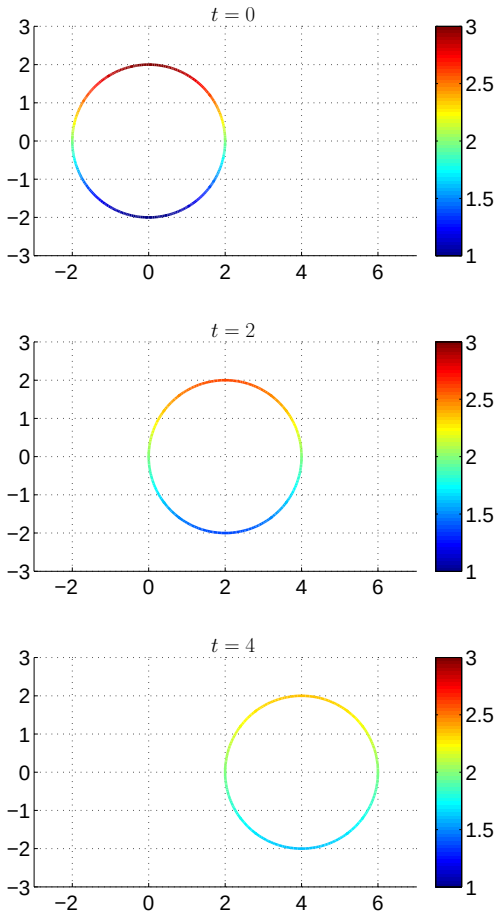

Figure 2: Position of the interface and the surfactant concentration on the moving interface at time t=0, 2, 4.

and the spectral condition number of the equation system as function of time for $h = 0.2$. In this example, we have not prescribed the total surfactant mass using a Lagrange multiplier but since the velocity field is constant we observe very good conservation of the total surfactant. We also see that the stabilization term ensures that the condition number of the resulting equation system is bounded as the interface evolves. Without stabilization the condition number sometimes becomes extremely large depending on how the interface cuts the mesh.

We compare our solution with the analytical solution

$$u(x, y, t) = e^{-t/4} \frac{y}{\sqrt{(x-t)^2 + y^2}} + 2 \tag{4.4}$$

at time $t = 2$. The error measured in the L^2 norm and the H^1 norm for different mesh sizes h is shown in Fig. 4. We also show the maximum error over all endpoints of the linear segments approximating the interface. We observe the expected first order convergence. The errors can be compared with Table II in [10]. Our two coarsest meshes correspond to the two finest meshes in [10]. For the mesh sizes used in Table II in [10] we observe smaller errors.

4.3. Example 2: Rigid body rotation

Here we consider the same rigid body rotation as in [24]. The initial interface is a circle centered at $(0.5, 0.22)$ with radius 0.17 and the velocity field $\boldsymbol{\beta} = (\frac{2\pi}{3}(y - 0.5), \frac{2\pi}{3}(x - 0.5))$. In this example we set $\epsilon = 0$ and hence the concentration is only advected by the velocity field. The time step

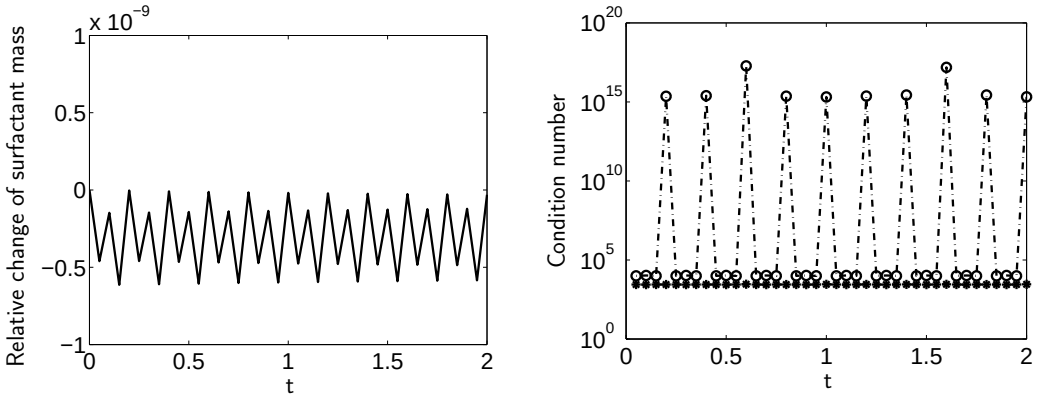

Figure 3: Left: The relative change of the total surfactant mass, given by (4.3), versus time using the proposed method. Right: The spectral condition number as function of time. Stars represent the condition number using the proposed method and circles represent the condition number without the stabilization term j_n. The mesh size $h = 0.2$.

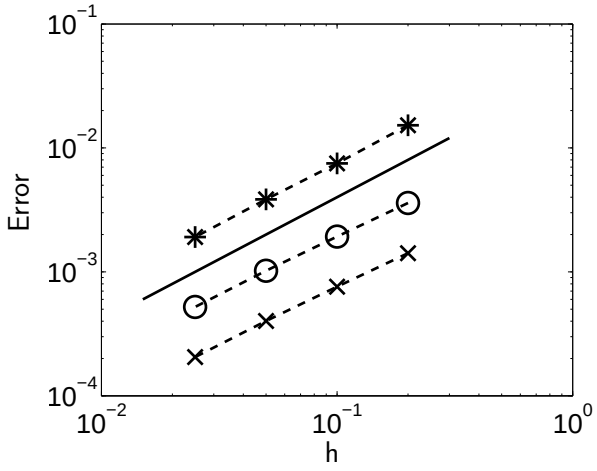

Figure 4: The error in the surfactant concentration at time $t = 2$ for different mesh sizes h. Circles represent the error measured in the L^2 norm. Stars represent the error measured in the H^1 norm. Crosses represent the maximum error over endpoints of the linear segments approximating the interface.

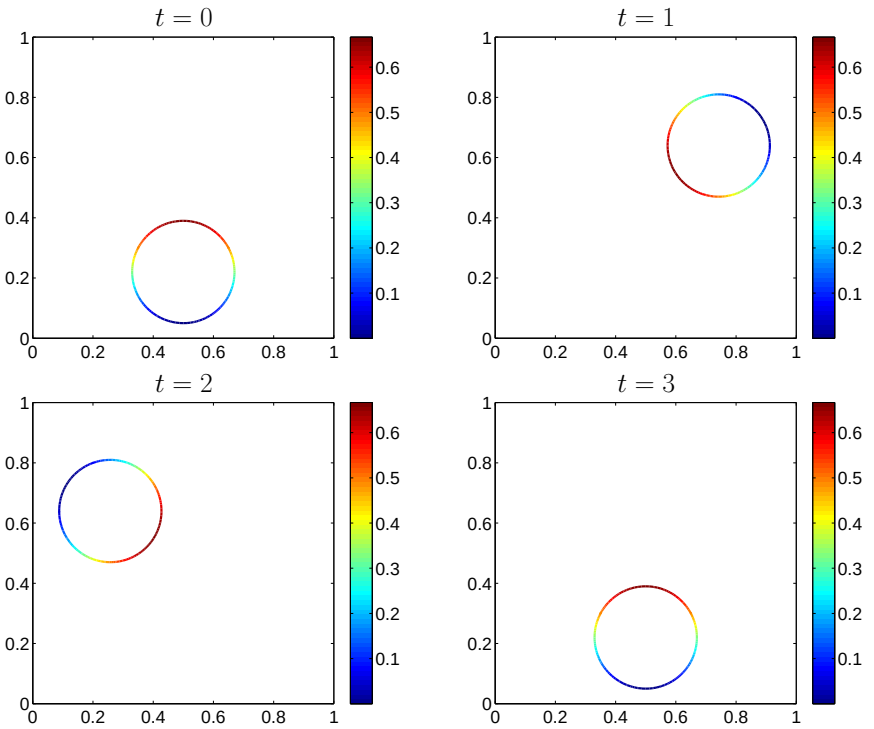

Figure 5: Position of the interface and the surfactant concentration on the moving interface at time t=0, 1, 2, 3. The mesh size $h = 0.025$.

$k = h/2$ as in [24]. For the coarsest mesh with $h = 0.025$ we show the surfactant concentration at times $t = 0, 1, 2, 3$ in Fig. 5 and the relative change of the total mass of surfactant in time in Fig. 6. We observe a mass loss of almost 0.016% when $h = 0.025$ without prescribing the exact surfactant mass. In [24] the mass loss is almost the same (Figure 3.13 in [24]) but for five times smaller mesh size ($h = 0.005$). We can easily improve the error in the surfactant mass in our method by prescribing the exact mass with a Lagrange multiplier, see Fig. 7.

We report the convergence in the L^2 norm but also in the L^1 norm in order to compare with the results in [24]. We observe slightly better convergence than the expected first order convergence. For the mesh sizes shown in Figure 3.13 of [24] (for $Pe_\Gamma = \infty$ corresponding to $\epsilon = 0$) we obtain smaller errors. However, the method in [24] is second order.

In Fig. 8 we also show the spectral condition number for different mesh sizes h. The numerical results indicate that the condition number grows as $\mathcal{O}(h^{-2})$.

4.4. Example 3: Deforming interface

We consider the problem given in Example 6 of [10]. Initially the interface Σ is a circle centered at the origin with radius $r_0 = 1$ and the velocity field $\boldsymbol{\beta} = (\frac{(y+2)^2}{3}, 0)$. The initial surfactant concentration $u = y/r_0 + 2$. The computational domain is $[-2\ 6.4] \times [-2\ 2]$ and we use 148 gridpoints in the x-direction which yields a meshsize $h \approx 0.06$. The time step $k = h/8$ as in [10]. The surfactant concentration on the moving interface at times $t = 0, 1, 2$ is shown in Fig 9. In Fig. 10 we show the relative change of the area enclosed by the interface and the relative error in the total surfactant mass versus time. We have prescribed the total surfactant mass using a Lagrange multiplier. We observe a change in the area by less than 0.005% and the relative error in the total surfactant mass (with respect to the exact total mass, $4\pi r_0$) is around 0.003% for $h \approx 0.06$. Fig. 8 and 9 in [10] show a surfactant mass loss of 4-5 % and a change in the area by 1% for $h = 0.04$.

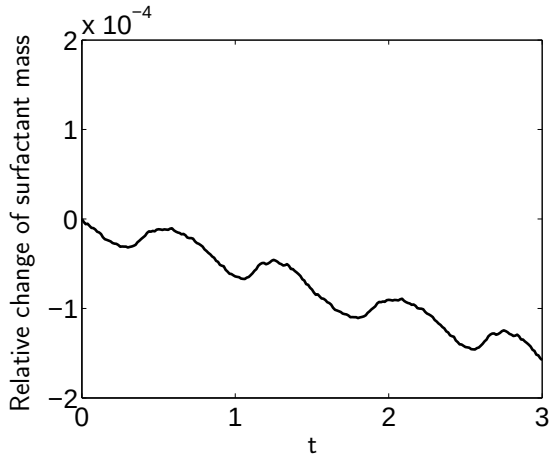

Figure 6: The relative change of the total surfactant mass, given by (4.3), versus time. The exact surfactant mass ($\frac{2}{3}\pi r_0$) is not prescribed. The mesh size $h = 0.025$.

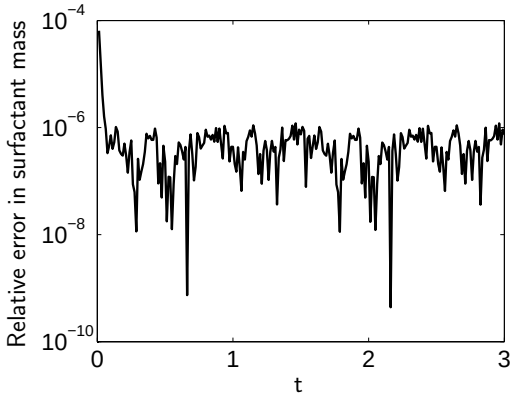

Figure 7: The relative error of the total surfactant mass, $\frac{\left|\int_{\Sigma(t)} u(t) - \frac{2}{3}\pi r_0\right|}{\frac{2}{3}\pi r_0}$, versus time when the condition $\int_{\Sigma(t)} u(t) = \frac{2}{3}\pi r_0$ is prescribed using a Lagrange multiplier. The mesh size $h = 0.025$.

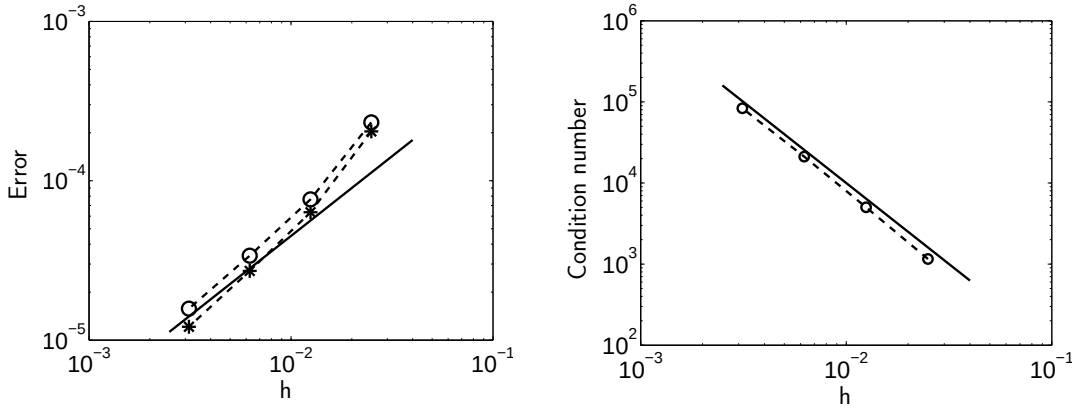

Figure 8: Left: The error in the surfactant concentration at time $t = 3$ for different mesh sizes h. Circles represent the error measured in the L^2 norm. Stars represent the error measured in the L^1 norm. The solid line is proportional to h^1. Right: Circles represent the spectral condition number at $t = 3$ for different mesh sizes h. The solid line is $y = h^{-2}$.

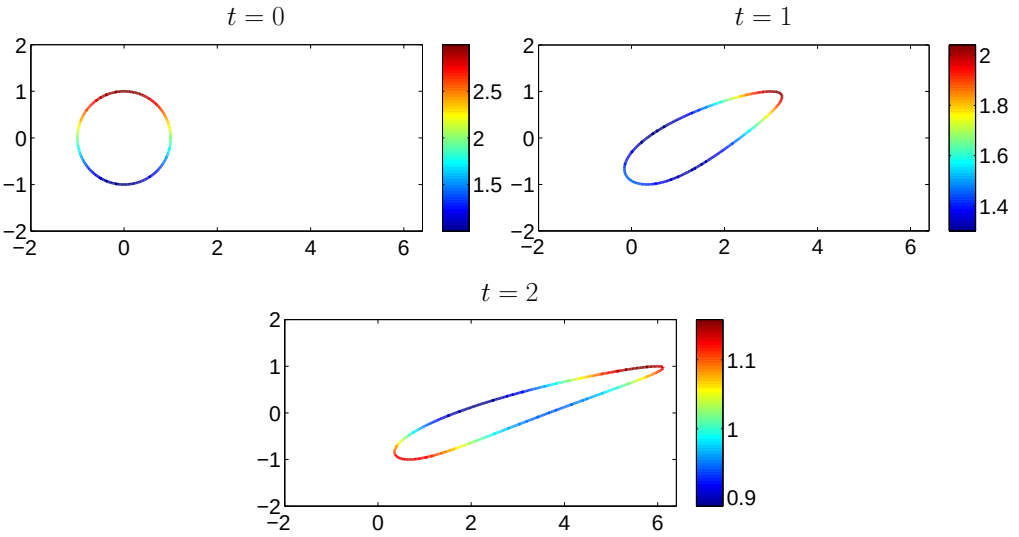

Figure 9: Position of the interface and the surfactant concentration on the moving interface at time t=0, 1, 2.

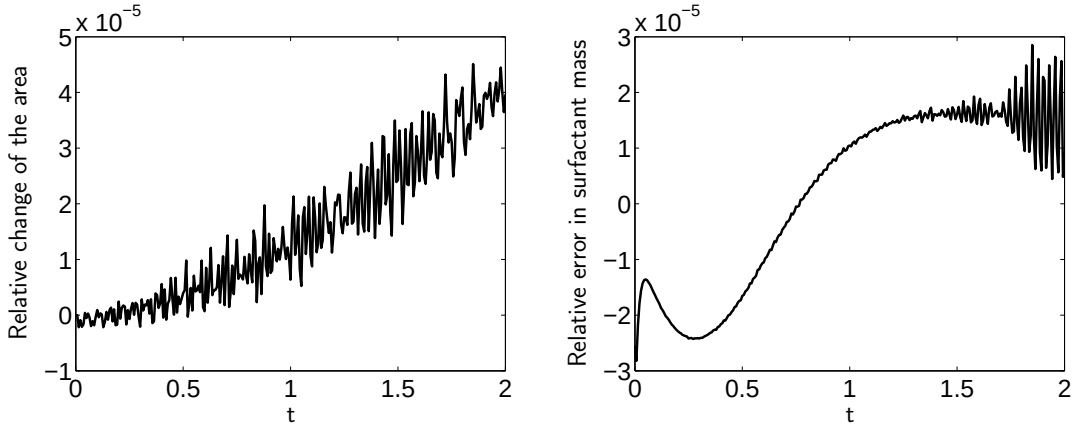

Figure 10: Left: The relative change of the area enclosed by the interface versus time. Right: The relative error in the total surfactant mass, $\frac{\int_{\Sigma(t)} u - 4\pi r_0}{4\pi r_0}$, versus time when the condition $\int_{\Sigma(t)} u = 4\pi r_0$ is prescribed using a Lagrange multiplier.

References

[1] L. L. Schramm, E. N. Stasiuk, D. G. Marangoni, 2 surfactants and their applications, Annu. Rep. Prog. Chem., Sect. C: Phys. Chem. 99 (2003) 3–48.

[2] S. Khatri, A.-K. Tornberg, A numerical method for two phase flows with insoluble surfactants, Computers & Fluids 49 (1) (2011) 150 – 165.

[3] M.-C. Lai, Y.-H. Tseng, H. Huang, An immersed boundary method for interfacial flows with insoluble surfactant, Journal of Computational Physics 227 (15) (2008) 7279 – 7293.

[4] S. Yon, C. Pozrikidis, A finite-volume/boundary-element method for flow past interfaces in the presence of surfactants, with application to shear flow past a viscous drop, Computers & Fluids 27 (8) (1998) 879 – 902.

[5] D. Adalsteinsson, J. A. Sethian, Transport and diffusion of material quantities on propagating interfaces via level set methods, Journal of Computational Physics 185 (1) (2003) 271 – 288.

[6] G. Dziuk, C. Elliott, An eulerian approach to transport and diffusion on evolving implicit surfaces, Computing and Visualization in Science 13 (2010) 17–28.

[7] C. M. Elliott, B. Stinner, V. Styles, R. Welford, Numerical computation of advection and diffusion on evolving diffuse interfaces, IMA Journal of Numerical Analysis 31 (3) (2011) 786–812.

[8] A. J. James, J. Lowengrub, A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant, Journal of Computational Physics 201 (2) (2004) 685 – 722.

[9] J.-J. Xu, Y. Yang, J. Lowengrub, A level-set continuum method for two-phase flows with insoluble surfactant, Journal of Computational Physics 231 (17) (2012) 5897 – 5909.

[10] J.-J. Xu, H.-K. Zhao, An Eulerian formulation for solving partial differential equations along a moving interface, J. Sci. Comput. 19 (1-3) (2003) 573–594, special issue in honor of the sixtieth birthday of Stanley Osher. doi:10.1023/A:1025336916176.
URL `http://dx.doi.org/10.1023/A:1025336916176`

[11] M. A. Olshanskii, A. Reusken, J. Grande, A finite element method for elliptic equations on surfaces, SIAM J. Numer. Anal. 47 (2009) 3339 – 3358.

[12] M. A. Olshanskii, A. Reusken, A finite element method for surface PDEs: matrix properties, Numer. Math. 114 (3) (2010) 491–520. doi:10.1007/s00211-009-0260-4.
URL `http://dx.doi.org/10.1007/s00211-009-0260-4`

[13] E. Burman, P. Hansbo, M. G. Larson, Stabilized cut finite elements for the laplace beltrami and helmholtz problems on surfaces, Tech. rep., UmeåUniversity (2013).

[14] P. Hansbo, A. Johansson, M. G. Larson, High order stabilized cut finite elements for the laplace beltrami and heat equation on surfaces, Tech. rep., UmeåUniversity (2013).

[15] H. A. Stone, A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface, Physics of Fluids A: Fluid Dynamics 2 (1) (1990) 111–112.

[16] H. Wong, D. Rumschitzki, C. Maldarelli, On the surfactant mass balance at a deforming fluid interface, Physics of Fluids 8 (11) (1996) 3203–3204.

[17] O. Pironneau, J. Liou, T. Tezduyar, Characteristic-galerkin and galerkin/least-squares space-time formulations for the advection-diffusion equation with time-dependent domains, Computer Methods in Applied Mechanics and Engineering 100 (1) (1992) 117 – 141.

[18] P. Cermelli, E. Fried, M. E. Gurtin, Transport relations for surface integrals arising in the formulation of balance laws for evolving fluid interfaces, J. Fluid Mech. 544 (2005) 339–351.

[19] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12 – 49.

[20] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annual Review of Fluid Mechanics 35 (1) (2003) 341–372.

[21] L. R. Scott, S. Zhang, Finite element interpolation of nonsmooth functions satisfying boundary conditions, Math. Comp. 54 (190) (1990) 483–493. doi:10.2307/2008497.
URL http://dx.doi.org/10.2307/2008497

[22] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Comput. Methods Appl. Mech. Engrg. 191 (47-48) (2002) 5537–5552. doi:10.1016/S0045-7825(02)00524-8.
URL http://dx.doi.org/10.1016/S0045-7825(02)00524-8

[23] M. Sussman, E. Fatemi, An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, SIAM Journal on Scientific Computing 20 (4) (1999) 1165–1191.

[24] S. Khatri, A Numerical Method for Two Phase Flows with Insoluble and Soluble Surfactants, Doctorate Thesis, New York University, Courant Institute of Mathematical Sciences, 2009.