

Eulerian Finite Element Methods for Parabolic Equations on Moving Surfaces

Jörg Grande*

Bericht Nr. 360

April 2013

Key words: finite elements, evolving surface, parabolic PDE,
space–time, two–phase flow, surfactant, computations

AMS Subject Classifications: 65M60, 35R01, 35K15, 76T99

**Institut für Geometrie und Praktische Mathematik
RWTH Aachen**

Templergraben 55, D–52056 Aachen (Germany)

* Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Templergraben 55,
D–52056 Aachen, Germany (grande@igpm.rwth-aachen.de).

EULERIAN FINITE ELEMENT METHODS FOR PARABOLIC EQUATIONS ON MOVING SURFACES

JÖRG GRANDE*

Abstract. Three new Eulerian finite element methods for parabolic PDEs on a moving surface $\Gamma(t)$ are presented and compared in numerical experiments. These are space-time Galerkin methods, which are derived from a weak formulation in space and time. The trial- and test-spaces contain the traces on the space-time manifold of an outer prismatic finite element space. The numerical experiments show that two of the methods converge with second order with respect to both, the time step size and the spatial mesh width.

Key words. finite elements, evolving surface, parabolic PDE, space-time, two-phase flow, surfactant, computations

AMS subject classifications. 65M60, 35R01, 35K15, 76T99

1. Introduction. In this paper, three new Eulerian finite element Galerkin methods for second-order parabolic partial differential equations (PDEs) on an evolving smooth hypersurface $\Gamma(t) \subset \mathbb{R}^3$, $t > 0$, are proposed. The idea from [12] to use traces of ‘outer’ finite element spaces is generalized to this setting in a space-time approach. Numerical experiments suggest that two of the methods converge with a rate that is second order in the spatial mesh width h_x and time step size h_t , if the error is measured in the $L_t^\infty L_x^2$ -norm on the space-time manifold defined below. To our knowledge, these are the first Eulerian finite element methods, which are second order accurate in space and time for parabolic PDEs on evolving surfaces.

An application, which involves such a PDE, is two-phase flow with a surface active agent (surfactant). The surface-concentration u of the surfactant on the phase interface $\Gamma(t)$ determines the properties of $\Gamma(t)$, for example the surface tension. The PDE, which describes the evolution of u , is introduced in Section 1.1. Some choices in the design of the new methods are based on the application to two-phase flow. This will be explained further in Section 10. Additional details on two-phase flow can be found in [8].

1.1. A model problem. The discretization methods that we treat in this paper are applicable to parabolic PDEs on evolving surfaces. In the derivation of the methods, we use a model that we introduce now.

The hypersurface Γ evolves according to the flow of the vector-field \mathbf{w} , called the wind. This means, that any point $x(t) \in \Gamma(t)$ follows the trajectory $\frac{\partial x}{\partial t} = \mathbf{w}(x(t), t)$, $t \in (0, T]$. The surfactant-concentration u obeys the PDE

$$\begin{cases} \bar{\mathbf{w}}^T \bar{D}u + D_\Gamma \cdot \mathbf{w} u - D_\Gamma \cdot (\alpha D_\Gamma u) = f & \text{on } \Gamma(t), t \in (0, T], \\ u(\cdot, 0) = u_0 & \text{on } \Gamma(0). \end{cases} \quad (1.1)$$

We use the space-time wind $\bar{\mathbf{w}} = (w_1, w_2, w_3, 1)^T$ and the space-time gradient $\bar{D} = (D_1, D_2, D_3, D_t)^T$. Then, $\bar{\mathbf{w}}^T \bar{D} = D_t + \mathbf{w}^T D$ is the material derivative. The operator D_Γ is the tangential gradient on $\Gamma(t)$. It is intrinsically defined on Γ . For functions, which are defined in a neighborhood of Γ , it is equivalently given by $\mathbf{P}D = (\mathbf{I} - \mathbf{n}\mathbf{n}^T)D$, where \mathbf{n} denotes the unit-length normal-field on Γ .

*Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, D-52056 Aachen, Germany (grande@igpm.rwth-aachen.de).

The operator D_Γ depends on t and the position x on $\Gamma(t)$. To simplify the notation, these dependencies are usually not made explicit.

Equation (1.1) is the differential version of the conservation law for u on Γ with the Fickian ansatz $F = -\alpha D_\Gamma u$ for the flux. A concise derivation can be found in [16].

1.2. Overview of the new methods. The numerical methods that we introduce in this paper are based on a weak formulation in *space and time*, cf. (3.2) in Section 3. An advantage of this novel approach for surface PDEs is that the material derivative can be applied exactly to the finite element trial functions. Other methods must resort to finite differences, e. g. to discretize the time derivative.

The formulation uses the space-time manifold \mathcal{G} , but no extension of the PDE to Ω . This avoids most of the problems mentioned in [6]. The numerical methods are Galerkin methods. The trial- and the test-spaces are (subspaces of) the trace-spaces of the standard prismatic finite element spaces on successive time slabs. The use of trace spaces in Galerkin methods was introduced in [12] for a Poisson-equation on a stationary manifold Γ . Another novelty of our paper is the extension of this idea to a parabolic equation on a non-stationary manifold using a space-time representation of $\Gamma(t)$.

Furthermore, no explicit surface mesh $\Gamma_h(t)$ is required. Only local approximations (per tetrahedron) are required for quadrature, cf. Section 9. This is an advantage over Lagrangian methods, which employ a surface mesh.

REMARK 1.1. *It might seem computationally inefficient, that a PDE on a 2-dimensional surface requires computations on the 4-dimensional space-time. This is not the case: The slabs of the space-time cylinder $\Theta = \Omega \times (0, T]$ are considered one after another as in a time-stepping scheme. Moreover, we use an adaptive strategy for the tetrahedral spatial mesh in a single time step; only the vicinity of $\Gamma(t)$, $t \in (t_i, t_{i+1}]$, is refined. The numerical experiments in Section 10 (Table 10.2) show that all three methods scale like methods for a PDE on a 2-dimensional domain.*

The first method, discussed in Section 5, uses a discontinuous Galerkin approach to enforce the initial values in each time step. This requires the addition of a (consistent) jump-term to the continuous problem. The method is a Ritz-Galerkin method, that is, it uses the same space as trial- and test-space.

The second method, discussed in Section 6, is a Petrov-Galerkin method for the unmodified continuous problem. The initial values are interpolated exactly, which requires some care on adaptive meshes. The initial values are eliminated from the trial-space. The test-space is reduced to obtain a square system of linear equations.

Both methods require the evaluation of integrals over space-time tetrahedra – that is, over tetrahedra which are embedded in \mathbb{R}^4 , cf. Subsection 8.2. This requires some implementation effort. The third method, discussed in Section 7, is a variant of the second one. It avoids the space-time tetrahedra by using the trapezoidal-rule to approximate the integral over time in (3.2), cf. Subsection 9.1. To avoid some degeneracies, the trial- and the test-space have to be reduced further compared to the second method. In the numerical experiments, this last method only works well on a stationary manifold.

1.3. Related literature. Known methods for PDEs on non-stationary manifolds can be classified by the frame of reference, in which the discretization is performed. Lagrangian methods employ a frame of reference which moves (approximately) with the flow-field of \mathbf{w} ; Eulerian methods use a fixed frame of reference. An

additional classification is possible via the discretization method for space and time, e.g. finite elements, finite volumes, finite differences.

Lagrangian methods are treated in [2, 4, 3, 10, 11, 14]. In [2, 4, 3], linear finite elements are used on a piecewise linear surface mesh. In [10] a finite volume method is introduced on such a mesh, and in [11], a finite difference method is presented. The method in [14] requires an explicit parameterization of the surface, employing isoparametric quadratic finite elements.

Eulerian methods are treated in [19, 18, 15, 9, 1]. The surface-PDE is extended to a degenerate parabolic PDE on a narrow band around the surface. In [19, 18, 15], the extended PDE is solved with finite differences on a simple Cartesian grid. A volume-of-fluid method is used in [9]. In [1], a finite element method is combined with an implicit Euler scheme for time discretization.

The recent preprint [13] is concerned with the well-posedness of the weak formulation in space and time of (1.1). Some numerical results with a variant of our first method are presented.

1.4. Outline of the paper. Basic notation is introduced in Section 2. The weak formulation in *space and time* of (1.1), cf. (3.2), is presented in Section 3.

Section 4 contains the definition of the finite element spaces needed subsequently; the ‘outer’ spaces and the trace spaces are defined. The three finite element methods are defined in the Sections 5, 6, and 7. In Section 8, explicit local geometric approximations of $\Gamma(t)$ and of the space-time manifold are introduced. These are needed for the local quadrature schemes in Section 9.

Section 10 contains three experiments, in which the three methods are compared. We report the convergence to the exact solutions with respect to h_x and h_t in two natural norms. Moreover, the computational costs of the three methods are compared.

2. Basic notation. The time-interval $I = (0, T]$ and the fixed outer domain $\Omega \subset \mathbb{R}^3$ define the *space-time-cylinder* $\Theta = \Omega \times I \subset \mathbb{R}^{3+1}$. The family $(\Gamma(t))_{t \in I}$ of 2-dimensional sub-manifolds of Ω defines

$$\mathcal{G} = \bigcup_{t \in I} \Gamma(t) \times \{t\},$$

the *space-time representation* of the non-stationary manifold $\Gamma(t)$. It is assumed that \mathcal{G} is at least C^2 -embedded into Θ as Riemannian sub-manifold. Also, $\partial\Gamma(t) = \emptyset$ is assumed for all $t \in I$. The latter assumption is not essential; it simplifies the presentation and the implementation of the methods.

For the weak formulation of (1.1), notation for integrals over Γ and \mathcal{G} is required. Let \mathcal{H}^d denote the d -dimensional Hausdorff-measure. For the hypersurface \mathcal{G} , the integral with respect to the (hyper-) surface measure $d\sigma = d\mathcal{H}^3$ and the iterated integral over space and time are related by

$$\int_{\mathcal{G}} f d\sigma = \int_I \int_{\Gamma(t)} Jf d\mathcal{H}^2 dt \quad (2.1)$$

with $J(x, t) = (1 + (\mathbf{n}(x, t) \cdot \mathbf{w}(x, t))^2)^{1/2}$. Both, the integral on \mathcal{G} with respect to $d\sigma$ and the iterated integral without J , can be used to define L^p -norms. Due to $1 \leq |J| \leq 1 + \|\mathbf{w}\|_{L^\infty(\mathcal{G})}$, these are equivalent up to the dependence on \mathbf{w} . For the weak formulation of (1.1), the iterated integrals are a natural choice, cf. Remark 3.1. A robust numerical approximation is easier for the surface measure $d\sigma$, cf. Section 9.

Hence, notation for both approaches is introduced. For a weighted L^2 -inner-product with respect to the Hausdorff-measure \mathcal{H}^d on the set S , the standard notation

$$(f, g)_{S; \omega} = \int_S \omega f g d\mathcal{H}^d$$

is used, where $d = 3$ for subsets of \mathcal{G} and $d = 2$ for subsets of $\Gamma(t)$. For a weighted L^2 -inner-product with respect to the iterated integral, the notation

$$(f, g)_{S; \omega}^{t,x} = \int_{\mathbb{R}} \int_{\{x|(x,t) \in S\}} \omega f g d\mathcal{H}^{d-1} dt$$

is used. In both cases, a trivial weight $\omega \equiv 1$ is omitted. Also, if the set S is clear from the context, it may be omitted. As an example, equation (2.1) can be written as $(f, 1)_{\mathcal{G}} = (f, J)_{\mathcal{G}}^{t,x} = (f, 1)_{\mathcal{G}; J}^{t,x}$.

The Sobolev-space $H^1(\Gamma(t))$ contains all \mathcal{H}^2 -measurable functions with weak tangential derivatives and finite norm given by $\|u\|_{H^1(\Gamma(t))}^2 = \|u\|_{\Gamma(t)}^2 + \|D_{\Gamma}u\|_{\Gamma(t)}^2$, where the standard notation $\|u\|_{\Gamma(t)}^2 = (u, u)_{\Gamma(t)}$ for the L^2 -norm on $\Gamma(t)$ is used on the right-hand side. Likewise, $H^1(\mathcal{G}) \subset L^2(\mathcal{G})$ is the Sobolev-space containing all functions on \mathcal{G} with weak derivatives. The Hilbert-space $L_t^2 H_x^1(\mathcal{G})$ is the set of all \mathcal{H}^3 -measurable functions on \mathcal{G} with finite $\int_I \|u\|_{H^1(\Gamma(t))}^2 dt$; the space $L_t^2 H_x^{-1}(\mathcal{G})$ denotes its dual. Finally, the norm $L_t^\infty L_x^2(\mathcal{G})$ -norm is defined by $\text{ess sup}_{t \in I} \|u\|_{\Gamma(t)}$.

3. Weak form of the transport equation. We derive a weak formulation in space and time of (1.1). This allows us to evaluate the material derivative exactly in the numerical methods. Equation (1.1) is multiplied by a test-function $v \in L_t^2 H_x^1(\mathcal{G}) =: X$ and integrated over $\Gamma(t)$. Partial integration on $\Gamma(t)$ yields

$$- \int_{\Gamma(t)} D_{\Gamma}(\alpha D_{\Gamma}u)v = \int_{\Gamma(t)} \alpha D_{\Gamma}u D_{\Gamma}v =: a_t(u, v).$$

Let \tilde{X} be a suitable subspace of X ; we comment on the choice below. We arrive at a first formulation, which is weak in space and strong in time: Find $u \in \tilde{X}$, such that

$$\begin{cases} c_t(u, v) + a_t(u, v) = (f, v)_{\Gamma} & \text{for all } v \in X, \text{ for a. e. } t \in I, \\ u(x, 0) = u_0(x) & \text{for all } x \in \Gamma(0), \end{cases} \quad (3.1)$$

where $c_t(u, v) = (\bar{\mathbf{w}}^T \bar{D}u, v)_{\Gamma(t)} + (u, v)_{\Gamma(t); D_{\Gamma} \cdot \mathbf{w}}$. Integration with respect to time yields the weak form of (1.1) in space and time: Find $u \in \tilde{X}$, such that

$$\begin{cases} c(u, v) + a(u, v) = (f, v)_{\mathcal{G}} & \text{for all } v \in X, \\ u(x, 0) = u_0(x) & \text{for all } x \in \Gamma(0), \end{cases} \quad (3.2)$$

where $c(u, v) = \int_I c_t(u, v)$ and $a(u, v) = \int_I a_t(u, v)$.

REMARK 3.1. *In Problem (3.2), integration is performed over space first and then over time. This makes it easy to apply partial integration in space. At the expense of a factor of J^{-1} , one can use (2.1) to restate Problem (3.2) with (\cdot, \cdot) instead of $(\cdot, \cdot)^{t,x}$. We use this fact in the implementation of two of the proposed new methods, cf. Section 9.*

A different weak formulation is obtained, if one multiplies (1.1) by v and integrates over \mathcal{G} . Then, by (2.1), one can still perform partial integration with respect to space, which yields

$$\int_{\mathcal{G}} D_{\Gamma}(\alpha D_{\Gamma}u)v = \int_{\mathcal{G}} \alpha D_{\Gamma}u D_{\Gamma}v + \int_I \int_{\Gamma(t)} \alpha v D_{\Gamma}u \cdot D_{\Gamma}J.$$

The right-most integral is an additional advective term acting on u compared to (3.2). This term does not vanish in general – an elementary computation shows $D_{\Gamma}J = J^{-1}(\mathbf{H}\mathbf{w} + (D_{\Gamma}\mathbf{w})\mathbf{n})$ with $\mathbf{H}_{ij} = D_j\mathbf{n}_i$ and $(D_{\Gamma}\mathbf{w})_{ij} = (D_{\Gamma})_j\mathbf{w}_i$. For purely normal wind $\mathbf{w} = \beta\mathbf{n}$, this simplifies to $D_{\Gamma}J = J^{-1}\beta D_{\Gamma}\beta$, which also does not vanish for general $\beta(x, t)$.

From [2], the well-posedness of Problem (3.1) is known, if we choose $\tilde{X} = H^1(\mathcal{G})$: Let $u_0 \in H^1(\Gamma(0))$ and $f \equiv 0$. Then, Problem (3.1) has a unique solution in $H^1(\mathcal{G})$ with the stability estimates

$$\begin{aligned} \|u\|_{L_t^\infty L_x^2(\mathcal{G})}^2 + \int_I \|D_{\Gamma}u\|_{\Gamma(t)}^2 &\leq c \|u_0\|_{\Gamma(0)}^2, \\ \int_I \|\tilde{\mathbf{w}}^T \bar{D}u\|_{\Gamma(t)}^2 + \|D_{\Gamma}u\|_{L_t^\infty L_x^2(\mathcal{G})}^2 &\leq c \|u_0\|_{H^1(\Gamma(0))}^2. \end{aligned} \quad (3.3)$$

Note, that the combination of the inequalities gives a bound for $\|u\|_{H^1(\mathcal{G})}$.

Clearly, any solution of Problem (3.1) solves Problem (3.2). For $\tilde{X} = H^1(\mathcal{G})$, in [8] the uniqueness of solutions of Problem (3.2) is shown.

REMARK 3.2. *In analogy to parabolic PDEs on standard domains, the choice $\tilde{X} = \{u \in X \mid \tilde{\mathbf{w}}^T \bar{D}u \in L_t^2 H_x^{-1}(\mathcal{G})\}$ seems natural. It is simple to formally derive typical stability estimates, e. g. (3.3), for Problem (3.2). Rigorous stability estimates in an Eulerian setting are derived in the preprint [13]. The solutions of the numerical examples in this preprint are obtained with a variant of the first method introduced here, based on our implementation in the finite element software DROPS, [7].*

4. Finite elements. For the purpose of discretization, the time interval I is partitioned into N sub-intervals as

$$0 = t_1 < t_2 < t_3 < \dots < t_{N+1} = T, \quad I^i = (t_i, t_{i+1}] \quad \text{for } i \in \{1, \dots, N\}.$$

The mesh width in time is denoted as $h_t = t_{i+1} - t_i$, where the dependence on the index i is suppressed for simpler notation. This partitions the space-time Θ and the space-time manifold \mathcal{G} into N space-time slabs,

$$\Theta^i = \Omega \times I^i, \quad \mathcal{G}^i = \mathcal{G} \cap \Theta^i, \quad i \in \{1, \dots, N\}.$$

A triangulation \mathcal{T}^i of the domain Ω is assumed at every discrete time t_i , $i \in \{1, \dots, N+1\}$. The slab Θ^i is partitioned into orthogonal prisms $T \times I^i$, where $T \in \mathcal{T}^i$. In general, \mathcal{T}^i is not aligned to Γ . The triangulations may be different for different i , for instance to coarsen regions of Ω , which are sufficiently far from $\Gamma(t)$. The following mild restriction on the \mathcal{T}^i is needed in Section 6.

ASSUMPTION 4.1. *For any $i \in \{1, \dots, N\}$ and $T \in \mathcal{T}^i$, if $\overline{T \times I^i} \cap \overline{\mathcal{G}^i} \neq \emptyset$ then $T \in \mathcal{T}^{i+1}$ or a refinement of T is in \mathcal{T}^{i+1} .*

Essentially, this means that the mesh is not coarsened in the part of Θ that contains the evolving surface. Coarsening is permitted everywhere else; this allows for an adaptive refinement strategy, which uses a small mesh width only in a small neighborhood of Γ .

4.1. The outer finite element space. First, let $Y_h^i = \{f \in C(\Omega) \mid f|_T \in \mathcal{P}^1, T \in \mathcal{T}^i\}$ be the space of continuous, piecewise linear finite elements on \mathcal{T}^i , $i \in \{1, \dots, N+1\}$. Let \mathcal{V}^i be the set of vertices of \mathcal{T}^i . Then, the nodal basis for Y_h^i is

$$\mathcal{B}_Y^i = \{b_j \in Y_h^i \mid b_j(v_k) = \delta_{j,k}, v_k \in \mathcal{V}^i, j = 1, \dots, |\mathcal{V}^i|\}.$$

Let I_{nodal}^i be the nodal interpolation operator of Y_h^i . The prismatic finite elements on the time slabs Θ^i of the outer domain are defined as,

$$X_h^i = \{(f + tg)1_{I^i} \mid f, g \in Y_h^i\},$$

where 1_{I^i} is the characteristic function of the time-interval I^i . We need the nodal basis of X_h^i . Let

$$\tau_0^i(t) = h_t^{-1}(t_{i+1} - t)1_{I^i}, \quad \tau_1^i(t) = h_t^{-1}(t - t_i)1_{I^i}, \quad i \in \{1, \dots, N\}.$$

Then, a basis for X_h^i is given by

$$\mathcal{B}_X^i = \tau_0^i \mathcal{B}_Y^i \cup \tau_1^i \mathcal{B}_Y^i, \quad i \in \{1, \dots, N\}.$$

For a single function τ and a set of functions M , we use the notation $\tau M = \{\tau m \mid m \in M\}$ for the set of products.

4.2. The trace spaces. In the finite element methods introduced in the following sections, the traces on \mathcal{G} and $\Gamma(t)$ of $u \in X_h^i$ are used to define the trial- and test-spaces.

Many functions in X_h^i are identically zero on \mathcal{G}^i . Consequently, they do not play any role in the discretization. To reduce the degeneracy of the trace-spaces, we introduce a subspace of X_h^i by using only those basis-functions in X_h^i which are nonzero on some part of \mathcal{G} . In particular, the basis functions are important which are nonzero on some $\Gamma(t_i)$, $i \in \{1, \dots, N+1\}$. We make this precise now.

We denote by tr_B the trace operator mapping $C(A, \mathbb{R}) \rightarrow C(B, \mathbb{R})$ for sets $B \subseteq A$. For a set of functions M , $\text{tr}_B M := \{\text{tr}_B f \mid f \in M\}$ is the image of M under tr_B . For $i \in \{1, \dots, N\}$, let

$$\mathcal{B}^i = \{b \in \mathcal{B}_Y^i \mid (b, b)_{\mathcal{G}^i} > 0\}, \quad (4.1)$$

$$\mathcal{C}_0^i = \{b \in \mathcal{B}_Y^i \mid (b, b)_{\Gamma(t_i)} > 0\}, \quad \mathcal{C}_1^i = \{b \in \mathcal{B}_Y^i \mid (b, b)_{\Gamma(t_{i+1})} > 0\}. \quad (4.2)$$

These sets contain the nodal basis functions on \mathcal{T}^i which are not identically zero on \mathcal{G}^i , respectively, on $\Gamma(t_i)$ and on $\Gamma(t_{i+1})$. An example is shown in Figure 4.1.

The set \mathcal{B}^i of spatial nodal basis functions is the smallest subset of \mathcal{B}_Y^i which generates all traces on \mathcal{G}^i . By Lemma 4.2 below, it is in one-to-two correspondence to the smallest subset of \mathcal{B}_X^i with non-vanishing traces on \mathcal{G}^i . Given $u \in X_h^i$, one obtains a function $\tilde{u} \in \text{span}(\tau_0^i \mathcal{B}^i \cup \tau_1^i \mathcal{B}^i)$ with $\text{tr}_{\mathcal{G}^i} \tilde{u} = \text{tr}_{\mathcal{G}^i} u$ by setting all coefficients from $\mathcal{B}_Y^i \setminus \mathcal{B}^i$ to zero. Likewise, $u \in Y_h^i$ implies that functions exist in $\text{span } \mathcal{C}_0^i$, $\text{span } \mathcal{C}_1^i$ with the same trace on $\Gamma(t_i)$, respectively on $\Gamma(t_{i+1})$.

LEMMA 4.2. *Let $i \in \{1, \dots, N\}$ be arbitrary.*

1. *For $b \in \mathcal{B}_Y^i$ the equivalences $b \in \mathcal{B}^i \iff (\tau_0^i b, \tau_0^i b)_{\mathcal{G}^i} > 0 \iff (\tau_1^i b, \tau_1^i b)_{\mathcal{G}^i} > 0$ hold.*
2. *The inclusion $\mathcal{C}_0^i \cup \mathcal{C}_1^i \subseteq \mathcal{B}^i$ holds.*
3. *Under Assumption 4.1, $u \in \text{span } \mathcal{B}^i$ implies $\text{tr}_{\Gamma(t_{i+1})} u = \text{tr}_{\Gamma(t_{i+1})} I_{\text{nodal}}^{i+1} u$.*

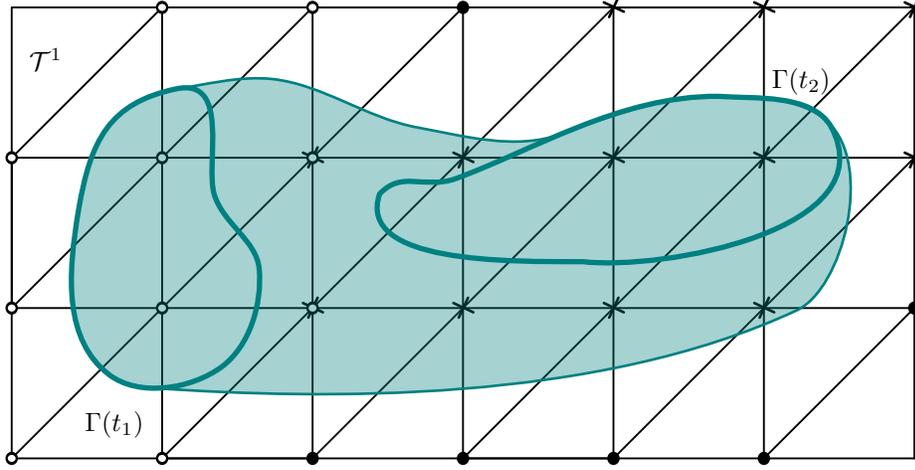


FIG. 4.1. The 1D evolving surface $\Gamma(t)$ moves over the shaded area in $(t_1, t_2]$. The vertices corresponding to the nodal basis \mathcal{C}_0^1 are marked with \circ , and the vertices corresponding to \mathcal{C}_1^1 are marked with \star ; the vertices of the additional basis functions in \mathcal{B}^1 are marked with \bullet .

The point of claim three is that u is defined with spatial finite elements on \mathcal{T}^i , whereas I_{nodal}^{i+1} is the interpolation on \mathcal{T}^{i+1} . The claim will be used in Section 6 to construct a continuous discrete solution in the presence of mesh changes.

Proof. The first claim: For $b \in \mathcal{B}^i$, one has from the mean value theorem that $(\tau_0^i b, \tau_0^i b)_{\mathcal{G}^i} = \tau_0^i(s)^2 (b, b)_{\mathcal{G}^i}$ with s in the open interval (t_i, t_{i+1}) . Thus, $\tau_0^i(s)^2$ is positive and the first equivalence is proved. Analogously for $\tau_1^i b$.

The second claim: For any continuous function $b(x)$ on Ω , the expression $(b, b)_{\Gamma(t)}$ is a continuous function of t . Thus, for any $b \in \mathcal{C}_0^i$, $(b, b)_{\Gamma(t)} > 0$ for all t in an open interval (t_i, s) with $s > t_i$. Analogously for \mathcal{C}_1^i .

The third claim: For any $u \in \text{span } \mathcal{B}^i$, let $u_{i+1} = I_{\text{nodal}}^{i+1} u \in \text{span } \mathcal{B}_Y^{i+1}$ be its nodal interpolation. Further, choose any $T \in \mathcal{T}^i$ with $\overline{T} \times \overline{T} \cap \overline{\mathcal{G}^i} \neq \emptyset$. By Assumption 4.1, either $T \in \mathcal{T}^{i+1}$ or there are $T_1, \dots, T_n \in \mathcal{T}^{i+1}$ which refine T . In both cases, $u_{i+1}|_T = u|_T$ because $u|_T$ is a linear function. \square

Let P be the $L^2(\Gamma(0))$ -projector on $\text{span } \mathcal{C}_0^1$, that is, for arbitrary $f \in L^2(\Gamma(0))$

$$(Pf - f, v)_{\Gamma(0)} = 0 \quad \text{for all } v \in \mathcal{C}_0^1.$$

REMARK 4.3. Under special geometric conditions on $\Gamma(t_i)$ and \mathcal{T}^i , $\text{tr}_{\Gamma(t_i)} \mathcal{C}_0^i$ can be linearly dependent, although \mathcal{C}_0^i is always linearly independent on Ω . A 2-dimensional example is depicted in Figure 4.2. This would only affect the solvers of the resulting system of equations, not the discrete solution on Γ . We have not yet met this case in numerical simulations with the approximation scheme for Γ and \mathcal{G} described in Section 8.

5. Method I – cGdG. The first method is a Galerkin method with a trial space of spatially continuous functions, which may be discontinuous at the times t_i . Similar to [5], we abbreviate such a continuous-in-space, discontinuous-in-time Galerkin discretization as ‘cGdG’. The time-variable in Problem (3.2) is discretized with a standard discontinuous Galerkin formulation for parabolic problems, cf. [5], but on an evolving surface – the initial values are enforced via an additional term in

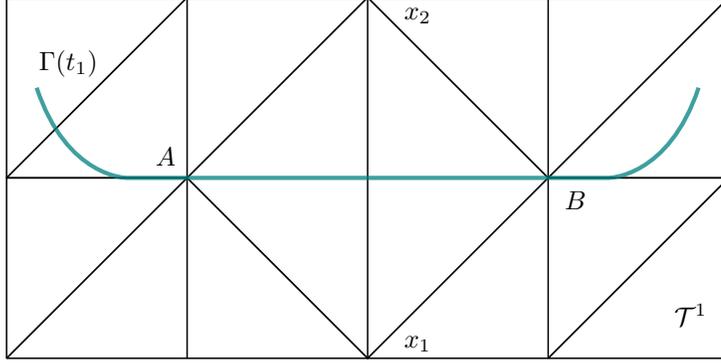


FIG. 4.2. The outer piecewise linear nodal basis functions b_1, b_2 for the vertices x_1, x_2 are linearly dependent when restricted to the (smooth) surface $\Gamma(t_1)$: $b_2 - b_1 \equiv 0$ on $\Gamma(t_1)$. There are no edges of \mathcal{T}^1 between A and B .

the PDE. This allows one to choose the same finite-dimensional space as trial- and test-space. The problem will be posed in a time-marching fashion.

We come to the details. Both, for the trial- and the test-space, the trace of X_h^i is used,

$$X_{\text{dG}}^i = \text{tr}_{\mathcal{G}} X_h^i, \quad i \in \{1, \dots, N\}.$$

Using the full trace-space X_{dG}^i as trial-space ensures good approximation properties.

The initial data for the cGdG-problem in time step i are denoted by u_0^i . The solution of the cGdG-problem in time step i is denoted by $u_h^i \in X_{\text{dG}}^i$, and the global discrete solution is $u_h = \sum_{i=1}^N u_h^i$. The initial data are defined inductively by

$$u_0^i = \begin{cases} Pu_0, & i = 1, \\ u_h^{i-1}(\cdot, t_i), & i = 2, \dots, N. \end{cases}$$

Opposite to the cGcG-method in Section 6, the boundary conditions are enforced in an L^2 -weak sense. Therefore, we do not need Assumption 4.1. Instead, we need notation for the limit of finite element functions, which are discontinuous in time: For an arbitrary point $(x, t) \in \bar{\mathcal{G}}$, let $((x_n^+, s_n^+))_n \subseteq \mathcal{G}$ be a sequence converging to (x, t) with $s_n^+ > s$. Then, the limit from above of a function f on \mathcal{G} is

$$f^+(x, t) = \lim_{n \rightarrow \infty} f(x_n^+, s_n^+).$$

The jump of f is $\llbracket f \rrbracket(x, t) = f(x, t)^+ - f(x, t)$. If $(x, t) \notin \mathcal{G}$, let $\llbracket f \rrbracket(x, t) = f^+(x, t)$.

The cGdG-problem for time step $i \in \{1, \dots, N\}$ reads: Find $u_h^i \in X_{\text{dG}}^i$ such that

$$c(u_h^i, v) + a(u_h^i, v) + \left(u_h^{i,+}(\cdot, t_i), v(\cdot, t_i) \right)_{\Gamma(t_i)} = (f, v)_{\bar{\mathcal{G}}}^{t,x} + (u_0^i, v^+(\cdot, t_i))_{\Gamma(t_i)} \quad \text{for all } v \in X_{\text{dG}}^i. \quad (5.1)$$

REMARK 5.1. In the implementation of the method, a frame for X_{dG}^i is required, which is given by

$$\mathcal{B}_{\text{dG}}^i = \tau_0^i \mathcal{B}^i \cup \tau_1^i \mathcal{B}^i, \quad i \in \{1, \dots, N\}.$$

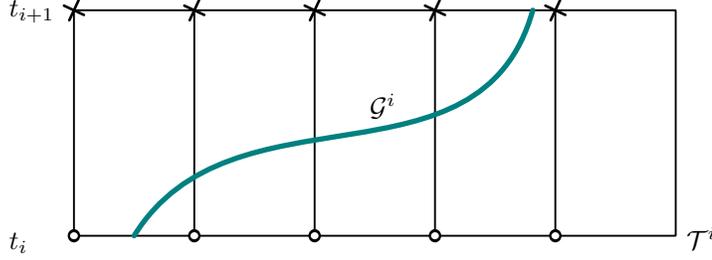


FIG. 5.1. The nodes of the outer prismatic finite element basis functions involved in the definition of $\mathcal{B}_{\text{dG}}^i$; \circ indicates $\tau_0^i b$, and \star indicates $\tau_1^i b$ for the spatial basis function b .

The nodes of the corresponding outer prismatic basis functions are visualized for a 1D space-time manifold in Figure 5.1. All integrals over $\Gamma(t_i)$ and \mathcal{G}^i are approximated as described in Section 9. This includes the integrals in the definition of $\mathcal{B}_{\text{dG}}^i$.

An equivalent formulation on \mathcal{G} is: Find $u_h \in X_{\text{dG}} = \bigoplus_{i=1}^N X_{\text{dG}}^i$ with

$$c(u_h, v) + a(u_h, v) + \sum_{i=1}^N (\llbracket u_h \rrbracket(\cdot, t_i), v^+(\cdot, t_i))_{\Gamma(t_i)} = (f, v)_{\mathcal{G}}^{t,x} + (u_0, v^+(\cdot, t_1))_{\Gamma(0)}$$

for all $v \in X_{\text{dG}}$.

6. Method II – cGcG. For this method, Problem (3.2) is discretized with different finite-dimensional trial- and test-spaces. The trial-space will be X_{dG}^i in time step i as in the cGdG-method. In contrast to the cGdG-method, a discrete solution is computed which is continuous in space and time. Due to the continuity of the discrete solution in both variables, the resulting Galerkin method is called cGcG-method in the sequel.

To obtain continuity in time, the initial values are interpolated with the help of the functions in \mathcal{C}_0^i from (4.2). These are all nodal basis-functions which are not identically zero on $\Gamma(t_i)$, cf. Figure 4.1. More precisely, the functions in $\tau_0^i \mathcal{C}_0^i$ are used for the interpolation. Consequently, only the coefficients of the remaining nodal basis-functions in \mathcal{B}^i must be determined by testing (3.2). Compared to the cGdG-method, this leads to a discrete problem with smaller dimension in each time step. Like the cGdG-method, the cGcG-method is a time-stepping method.

We come to the details. The trial-space is $X_{\text{trial}}^i = \text{tr}_{\mathcal{G}} X_h^i$. The generating system $\mathcal{B}_{\text{trial}}^i$ in time step $i \in \{1, \dots, N\}$ can be partitioned as follows:

$$\mathcal{B}_{\text{trial}}^i = \tau_0^i \mathcal{B}^i \cup \tau_1^i \mathcal{B}^i = \mathcal{B}_{\text{ini}}^i \cup \tau_0^i (\mathcal{B}^i \setminus \mathcal{C}_0^i) \cup \tau_1^i \mathcal{B}^i, \quad \mathcal{B}_{\text{ini}}^i = \tau_0^i \mathcal{C}_0^i. \quad (6.1)$$

The partitioning can be visualized in Figure 5.1. The leftmost two nodes marked with \circ belong to $\mathcal{B}_{\text{ini}}^i$; the other nodes with \circ belong to the middle set of the partition, and the nodes marked with \star correspond to $\tau_1^i \mathcal{B}^i$. The second equality in (6.1) follows from $\mathcal{C}_0^i \subseteq \mathcal{B}^i$ in Lemma 4.2. As noted above, $\text{tr}_{\mathcal{G}}(\text{span } \mathcal{B}_{\text{trial}}^i) = \text{tr}_{\mathcal{G}} X_h^i$. The functions in $\mathcal{B}_{\text{ini}}^i$ are precisely those of $\mathcal{B}_{\text{trial}}^i$ which are not identically zero on $\Gamma(t_i)$.

A generating system for test-space is defined by

$$\mathcal{B}_{\text{test}}^i = \tau_0^i (\mathcal{B}^i \setminus \mathcal{C}_0^i) \cup \mathcal{B}^i.$$

The nodes of the outer basis functions corresponding to the members of $\mathcal{B}_{\text{test}}^i$ are shown in Figure 6.1. The test-space contains the functions generated by \mathcal{B}^i which

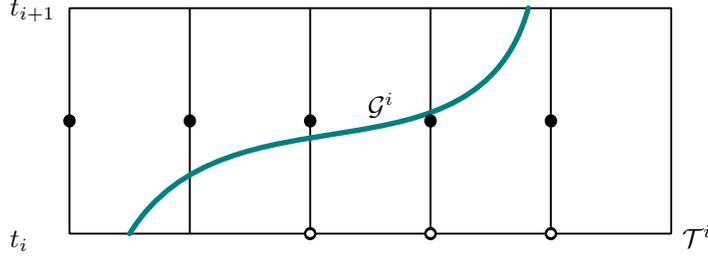


FIG. 6.1. The nodes of the outer prismatic finite element basis functions involved in the definition of \mathcal{B}_{test}^i ; \circ indicates $\tau_0^i b$, and \bullet indicates $1_{T^i} b$ for the spatial basis function b .

are constant in time. This is motivated by superconvergence theory for Runge-Kutta schemes for ODEs. It is also consistent with the well-known cG(1)-time-discretization for parabolic PDEs on stationary domains, cf. [5].

REMARK 6.1. *We comment on the choice of the test-functions. On a stationary surface, there holds $\mathcal{C}_0^i = \mathcal{B}^i$, so $\mathcal{B}_{trial}^i = \mathcal{B}_{ini}^i \cup \tau_1^i \mathcal{B}^i$ and $\mathcal{B}_{test}^i = \mathcal{B}^i$. As the degrees of freedom from \mathcal{B}_{ini}^i are eliminated with the help of the initial data, this yields an equal number of degrees of freedom remaining in \mathcal{B}_{trial}^i and in \mathcal{B}_{test}^i . Precisely this choice is made in the cG(1)-time-discretization, cf. [5].*

For non-stationary $\Gamma(t)$, \mathcal{B}_{trial}^i contains the additional functions $\tau_0^i(\mathcal{B}^i \setminus \mathcal{C}_0^i)$. We enlarge the set of test-functions by the same set. As for a stationary domain, the sets $\mathcal{B}_{trial}^i \setminus \mathcal{B}_{ini}^i$ and \mathcal{B}_{test}^i have the same cardinality.

The initial data for the cGcG-problem in time step i are denoted by $u_0^i \in \text{span } \mathcal{C}_0^i$. The discrete solution in time step i is denoted by $u_h^i \in \text{span } \mathcal{B}_{trial}^i$, and the global discrete solution is $u_h = \sum_{i=1}^N u_h^i$. The initial data are inductively defined as

$$u_0^i = \begin{cases} Pu_0, & i = 1, \\ I_{nodal}^i u_h^{i-1}(\cdot, t_i), & i = 2, \dots, N. \end{cases} \quad (6.2)$$

Note, that in the presence of mesh-coarsening u_h is generally not a continuous function at the t_i . But under Assumption 4.1 and by Lemma 4.2, its trace is continuous,

$$\text{tr}_{\mathcal{G}} u_h \in C(\mathcal{G}, \mathbb{R}).$$

The cGcG-problem for time step $i \in \{1, \dots, N\}$ reads: Find $u_h^i = \tau_0^i u_0^i + \tilde{u}_h^i$, $\tilde{u}_h^i \in \text{span}(\mathcal{B}_{trial}^i \setminus \mathcal{B}_{ini}^i)$, such that

$$c(u_h^i, v) + a(u_h^i, v) = (f, v)_{\mathcal{G}^i}^{t,x} \quad \text{for all } v \in \text{span } \mathcal{B}_{test}^i. \quad (6.3)$$

An equivalent formulation on the all of \mathcal{G} is as follows: Find $u_h = \tau_0^1 Pu_0 + \tilde{u}_h$, $\tilde{u}_h \in \text{span}((\cup_{i=1}^N \mathcal{B}_{trial}^i) \setminus \mathcal{B}_{ini}^1)$ with $\text{tr}_{\mathcal{G}} u_h \in C(\mathcal{G}, \mathbb{R})$ and

$$c(u_h, v) + a(u_h, v) = (f, v)_{\mathcal{G}}^{t,x} \quad \text{for all } v \in \text{span}(\cup_{i=1}^N \mathcal{B}_{test}^i).$$

In the implementation of the method, all integrals over \mathcal{G}^i are approximated as described in Subsection 8.2. This includes the integrals in the definition of \mathcal{B}_{trial}^i and \mathcal{B}_{test}^i .

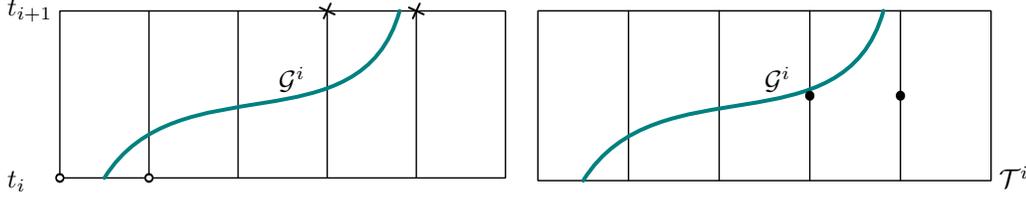


FIG. 7.1. Left: The nodes of the outer prismatic finite element basis functions involved in the definition of $\tilde{\mathcal{B}}_{\text{trial}}^i$. Right: The nodes involved in the definition of $\tilde{\mathcal{B}}_{\text{test}}^i$. The circle \circ indicates $\tau_0^i b$; \star indicates $\tau_1^i b$, and \bullet indicates $1_{\Gamma^i} b$ for the spatial basis function b .

7. Method III – cGcG*. This method is a simplification of the cGcG-method. The intention is to avoid a full space-time approach, which requires considerable implementation effort. Instead, the trapezoidal rule is used to approximate the integral over time, cf. Subsection 9.1. This makes it necessary to reduce the trial- and test-space of the cGdG-method to exclude functions which are zero on $\Gamma(t_i)$ and on $\Gamma(t_{i+1})$, but not on \mathcal{G}^i .

The generating system $\tilde{\mathcal{B}}_{\text{trial}}^i$ in time step $i \in \{1, \dots, N\}$ is defined as

$$\tilde{\mathcal{B}}_{\text{trial}}^i = \mathcal{B}_{\text{ini}}^i \cup \tau_1^i \mathcal{C}_1^i, \quad \mathcal{B}_{\text{ini}}^i = \tau_0^i \mathcal{C}_0^i.$$

The generating system for test-space is defined as

$$\tilde{\mathcal{B}}_{\text{test}}^i = \mathcal{C}_1^i.$$

The nodes of the outer basis functions corresponding to the members of $\tilde{\mathcal{B}}_{\text{trial}}^i$ and $\tilde{\mathcal{B}}_{\text{test}}^i$ are shown in Figure 7.1. The initial value is computed as in (6.2). The cGcG-problem for time step $i \in \{1, \dots, N\}$ reads: Find $u_h^i = \tau_0^i u_0^i + \tilde{u}_h^i$, $\tilde{u}_h^i \in \text{span}(\tau_1^i \mathcal{C}_1^i)$, such that

$$c(u_h^i, v) + a(u_h^i, v) = (f, v)_{\mathcal{G}^i}^{t,x} \quad \text{for all } v \in \text{span} \tilde{\mathcal{B}}_{\text{test}}^i. \quad (7.1)$$

An equivalent formulation on all of \mathcal{G} can be stated as: Find $u_h = \tau_0^1 P u_0 + \tilde{u}_h$, $\tilde{u}_h \in \text{span}((\cup_{i=1}^N \tilde{\mathcal{B}}_{\text{trial}}^i) \setminus \mathcal{B}_{\text{ini}}^1)$ with $\text{tr}_{\mathcal{G}} u_h \in C(\mathcal{G}, \mathbb{R})$ and

$$c(u_h, v) + a(u_h, v) = (f, v)_{\mathcal{G}}^{t,x} \quad \text{for all } v \in \text{span}(\cup_{i=1}^N \tilde{\mathcal{B}}_{\text{test}}^i).$$

In the implementation of the method, all integrals over \mathcal{G}^i are approximated as described in Subsection 9.1. This includes the integrals in the definition of $\tilde{\mathcal{B}}_{\text{trial}}^i$ and $\tilde{\mathcal{B}}_{\text{test}}^i$.

8. Approximation of the hypersurfaces. The evolving surface $\Gamma(t)$ is described implicitly as the zero-level of the time-dependent level set function $\varphi(x, t)$,

$$\Gamma(t) = \{x \in \Omega \mid \varphi(x, t) = 0\}.$$

In the application to two-phase flow, Γ evolves with the flow of the fluid. The motion of Γ is described via the time evolution of φ given by the level set equation, cf. [8]. In the experiments in Section 10, appropriate time-dependent level set functions are prescribed.

To implement the three new finite element methods, integrals over the 2-dimensional manifolds $\Gamma(t_i)$ and over the 3-dimensional space-time manifolds \mathcal{G}^i must be

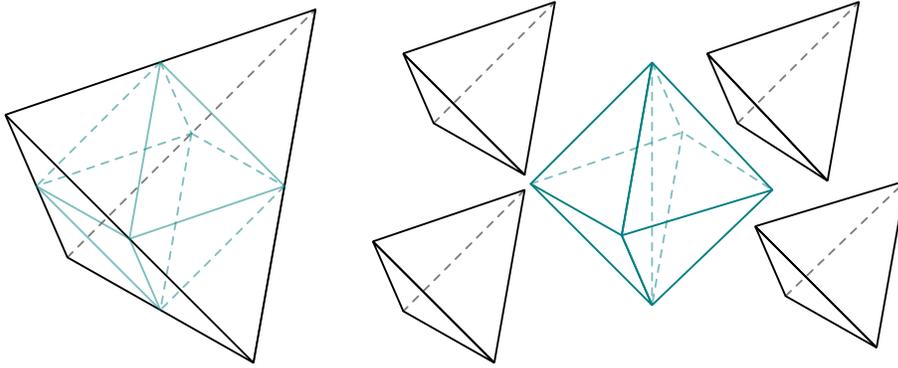


FIG. 8.1. The regular refinement of a tetrahedron is composed of 8 smaller tetrahedra.

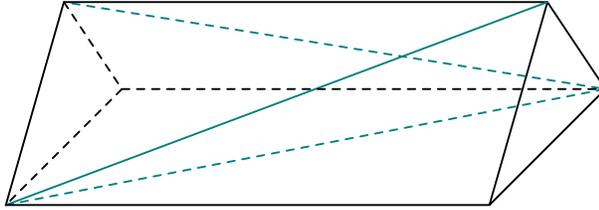


FIG. 8.2. A prism with triangular base can be tetrahedralized with 3 diagonals.

computed. Therefore, a geometric approximation Γ_h of $\Gamma(t_i)$ is required; likewise, a geometric approximation \mathcal{G}_h of \mathcal{G}^i is required. The approximations are computed locally, that is, as an approximation within a single tetrahedron $T \in \mathcal{T}^i$ or within a single space-time prism $T \times I^i$. The surface Γ_h will be composed of triangles, and \mathcal{G}_h will be composed of space-time tetrahedra.

REMARK 8.1. *In general, the approximation of the geometry makes it necessary to extend data, which are defined only on $\Gamma(t)$. For the numerical experiments in Section 10, all problem data are given by smooth functions on Ω . We do not perform an extension there.*

8.1. Approximation of Γ . We describe the construction of Γ_h within a single tetrahedron T . First, T is regularly refined into 8 tetrahedra T_1, \dots, T_8 , cf. Figure 8.1. This improves the approximation on very coarse triangulations. On each T_j , the unique linear interpolant $l_j(x)$ of the level set function φ is computed. The zero-level of l_j is denoted as Z_j . If Z_j is not degenerate, it is either a triangle or a (planar) quadrilateral. In the latter case, a diagonal is inserted. The triangles composing Z_1, \dots, Z_8 are the local approximation of Γ within T .

8.2. Approximation of \mathcal{G} . We describe the construction of \mathcal{G}_h within a single space-time prism $T \times I^i$. As in Section 8.1, T is regularly refined into 8 tetrahedra T_1, \dots, T_8 . Each of the resulting space-time prisms $T_j \times I^i$ is partitioned into 4 pentatopes by inserting adequate diagonals – Figure 8.2 shows the corresponding partitioning of a prism with triangular base in \mathbb{R}^3 . On each pentatope P , the unique linear interpolant $l_P(x, t)$ of the level set function φ is computed. The zero-level of l_P is denoted as Z_P . If Z_P is not degenerate, it is a 3-dimensional convex polytope, which is partitioned into tetrahedra. We omit the technical details. The tetrahedra

composing the sets Z_P are the local approximation of \mathcal{G} within $T \times I^i$.

9. Quadrature. Let S be a triangle defined in Section 8.1 or a tetrahedron defined in Section 8.2. On the simplex S , a quadrature-rule Q_S from [17], which is exact for polynomials up to degree 5, is employed to approximate the integral \int_S . The rule for triangles uses 7 points, the rule for tetrahedra uses 15 points. All quadrature weights are positive. The iterated integrals occurring in (3.2) and its discretizations are approximated using (2.1),

$$\int_{I^i} \int_{\Gamma(t)} f \approx \sum_{S \subseteq \mathcal{G}_h \cap I^i} Q_S(J^{-1}f).$$

REMARK 9.1. *Quadrature is performed element by element. First, the local approximation $\Gamma_h \cap T$, respectively $\mathcal{G}_h \cap (T \times I^i)$, for a single $T \in \mathcal{T}^i$ is computed. Then, all required integrals are evaluated locally. Hence, a memory-intensive global representation of Γ_h and, more important, of \mathcal{G}_h is avoided. Moreover, only one pass over \mathcal{T}^i is needed for the discretization in each time step.*

To further accelerate the discretization, the tetrahedra are treated in parallel in DROPS. A coloring of \mathcal{T}^i is employed to ensure that the updates from different processes to the stiffness matrices and load vectors do not interfere with each other.

9.1. Direct approximation of the iterated integral. For the cGcG* -method, we directly approximate the iterated integral $\int_{I^i} \int_{\Gamma_h} f$ without an intermediate geometric approximation of \mathcal{G} . For the time integral, the trapezoidal rule is used, for the integral over Γ quadrature over Γ_h is used,

$$\begin{aligned} \int_{I^i} \int_{\Gamma} f &\approx \frac{h_t}{2} \left(\int_{\Gamma_h(t_i)} f + \int_{\Gamma_h(t_{i+1})} f \right) \\ &\approx Q(f) = \frac{h_t}{2} \left(\sum_{S \subseteq \Gamma_h(t_i)} Q_S f + \sum_{S \subseteq \Gamma_h(t_{i+1})} Q_S f \right). \end{aligned}$$

10. Numerical experiments. Three experiments with increasing complexity are conducted: We solve the heat equation on a stationary sphere, the parabolic equation (1.1) with tangential wind on a stationary torus, and the same equation on a deforming ellipsoid. In all cases, the exact solution is known and used to determine the discretization errors.

We implement the three new finite element methods in the flow-solver DROPS, [7]. Some of the important available components are

- a high-quality tetrahedral 3D mesh with adaptive refinement close to $\Gamma(t)$,
- a well-tested finite element implementation on the outer 3D domain Ω ,
- a level set representation of $\Gamma(t)$ as $\{x \in \Omega \mid \varphi(x, t) = 0\}$.

Explicit 2-dimensional meshes embedded in \mathbb{R}^3 and methods to retain the quality of such meshes as they move through Ω are not available.

The linear systems are solved with an iterative method with a stopping criterion such that the initial residual is reduced by 1e-8 in the l^2 -norm.

The outer domain is the cube $\Omega = (-1.5, 1.5)^3$, which is subdivided into 3^3 cubes with edge-length 1. Each cube is further divided into six tetrahedra. Then, adaptive refinement is applied in the proximity of $\Gamma(0)$ to reach the mesh width h_x reported in

the tables below. All tetrahedra which are intersected by $\Gamma(t)$ have bounded interior angles because the applied refinement method is stable. For test case three, adaptive refinement is applied in a sufficiently large neighborhood of $\Gamma(0)$ such that $\Gamma(t)$ does not leave the region with the mesh width h_x . Effectively, the evolving surface is always embedded in a uniform triangulation of mesh width h_x , but the memory-footprint is *much smaller* due to the adaptive strategy.

Two error-measures are reported, which correspond to standard norms in the analysis of parabolic PDEs. The discrete $L_t^\infty L_x^2$ -norm: The $L_x^2(\Gamma(t_i))$ -norm is computed on the approximate interfaces $\Gamma_h(t_i)$ with the quadrature rule described in Section 9. The known solutions in the three test cases are defined by smooth functions on Ω , which are evaluated on Γ_h . The maximum of the spatial errors is taken over all discrete time-points t_i in the simulation.

The discrete $L_t^2 H_x^1$ -norm: The $H_x^1(\Gamma(t_i))$ -norm is computed on Γ_h as follows. The smooth known solution u is interpolated on the outer triangulation with standard linear finite elements yielding $u_{I,i}$. Then, $(u_h - u_{I,i})^T A_h (u_h - u_{I,i})$ is computed for each t_i , where A_h is the discrete Laplace-Beltrami operator on $\Gamma_h(t_i)$. The integration with respect to time is approximated with the trapezoidal rule from these data.

The error plots for the different methods within one test case all have the same ranges on the axes to simplify their visual comparison. The left-hand plots show the error as a function of h_t for several fixed values of h_x ; in the right-hand plots, the roles of h_t and h_x are reversed. Moreover, the numerical parameter in the legend has a quantitative meaning; it is the negative logarithm to base 2 of the fixed parameter. As an example, in the left plot of Figure 10.1, the numbers of the six curves are $-\log_2 h_x$ corresponding to $h_x \in \{2^{-1}, \dots, 2^{-6}\}$.

10.1. Heat equation on a stationary sphere. In the first experiment, the surface is $\Gamma(t) = \Gamma(0) = S_2$ with $0 < t \leq T = 0.5$. Then, $u(x, t) = \exp(-6t)x_1x_2$ solves

$$\begin{cases} \partial_t u - \Delta_\Gamma u = 0 & \text{on } S_2, 0 < t \leq T, \\ u(\cdot, 0) = x_1x_2 & \text{on } S_2. \end{cases}$$

Figure 10.1 shows the $L_t^\infty L_x^2$ -errors for the cGdG-method. The left plot shows the errors for several fixed values of h_x as h_t varies. One can see convergence against the semidiscrete solutions. In the right plot, the roles of h_x and h_t are reversed: h_x varies for several fixed values of h_t . Again, convergence against the semidiscrete solutions can be observed, at least for the larger time step sizes. With respect to time, we observe order 2.6, with respect to space order 2.

Figure 10.2 shows the $L_t^\infty L_x^2$ -errors for the cGcG-method. The plots have the same layout as those for the cGdG-method in Figure 10.1. With respect to time, we observe order 2.1, with respect to space order 2. For $h_x = h_t$, the absolute size of the error is comparable to that of the cGdG-method.

The $L_t^\infty L_x^2$ -errors of the cGcG*-method agree with that of the cGcG-method up to solver accuracy for this test case. This is expected: As Γ is stationary, the finite element spaces of the two methods are identical. Further, all discrete operators are constant in time, which makes the trapezoidal rule in the cGcG*-method an exact integrator.

The $L_t^2 H_x^1$ -errors of the three methods are shown in Table 10.1 for $h_t = h_x$. As before, the errors of the cGcG- and the cGcG*-method are identical. The methods

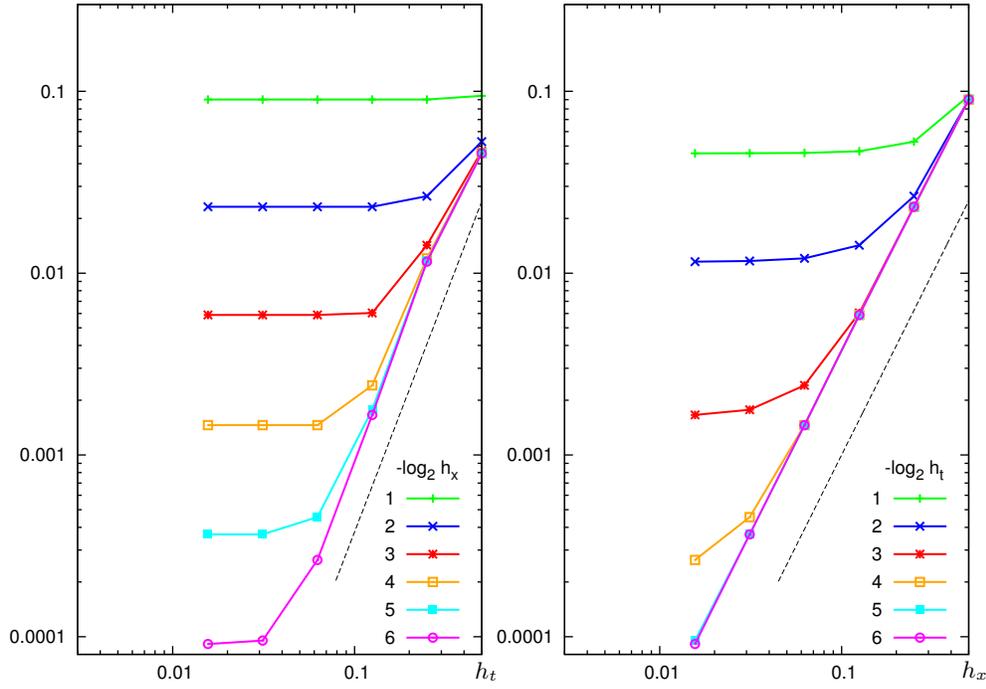


FIG. 10.1. Test case 1: $L_t^\infty L_x^2$ -error for the cGdG-method; left: $h_x = \text{const.}$, right: $h_t = \text{const.}$. The dotted lines have a slope of 2.6 and 2.

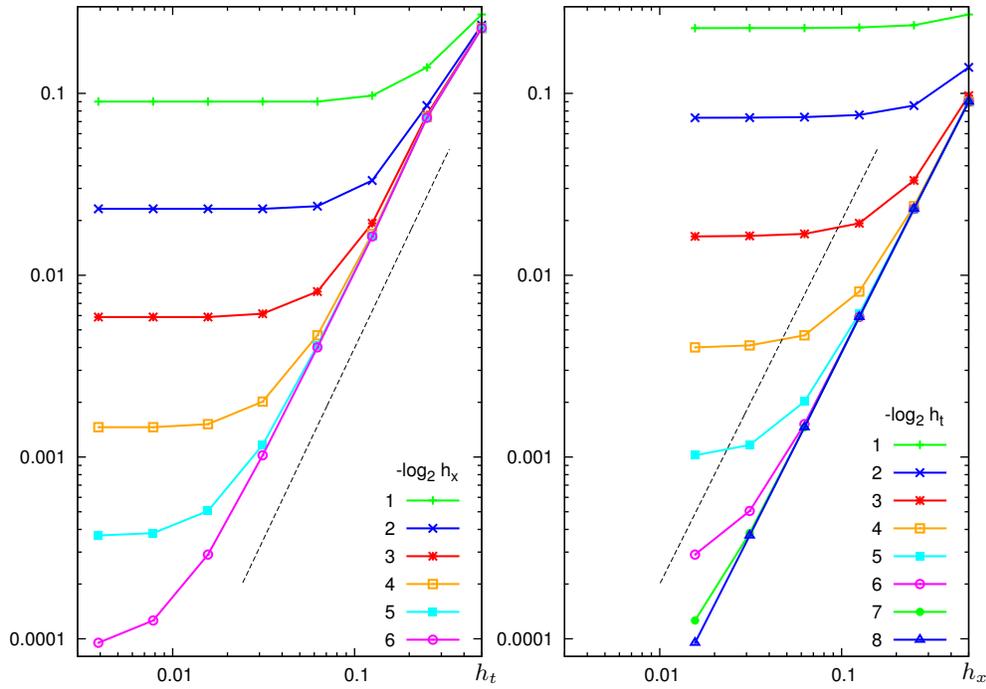


FIG. 10.2. Test case 1: $L_t^\infty L_x^2$ -error for the cGcG-method; left: $h_x = \text{const.}$, right: $h_t = \text{const.}$. The dotted lines have a slope of 2.1 and 2.

$h_t = h_x$	0.5	0.25	0.125	0.0625	0.03125	0.015625
cGdG	0.081365	0.0451713	0.0307505	0.0188471	0.01037	0.00545458
		0.849	0.555	0.706	0.862	0.927
cGcG	0.448733	0.239563	0.118549	0.0594443	0.0296327	0.0148497
		0.906	1.02	0.996	1.00	0.997
cGcG*	0.448733	0.239563	0.118549	0.0594443	0.0296327	0.0148497
		0.906	1.02	0.996	1.00	0.997

TABLE 10.1

Test case 1: $L_t^2 H_x^1$ -error and estimated order of convergence (e.o.c.).

$h_t = h_x$	0.5	0.25	0.125	0.0625	0.03125	0.015625	
cGdG	n_{\min}	38	73	167	467	997	2000*
	n_{\max}	38	73	168	470	1014	2000*
	t_{steps}	0.033	0.292	2.829	33.95	358.8	5964
	t_{disc}	0.032	0.278	2.521	21.35	153.9	1385
	M_{\max}	9.20	31.5	141	568	2220	8870
cGcG	n_{\min}	23	46	83	209	534	1514
	n_{\max}	23	48	85	249	605	1972
	t_{steps}	0.029	0.242	2.208	21.67	213.1	3094
	t_{disc}	0.029	0.238	2.135	19.19	159.2	1319
	M_{\max}	8.35	31.5	141	568	2220	8870
cGcG*	n_{\min}	23	46	83	209	534	1514
	n_{\max}	23	48	85	249	605	1973
	t_{steps}	0.014	0.054	0.798	6.221	99.55	2109
	t_{disc}	0.014	0.051	0.651	4.110	37.54	282.3
	M_{\max}	6.68	31.5	141	568	2220	8870

TABLE 10.2

Test case 1: Performance of the methods in the simulations with $h_x = h_t$. (* maximum iteration number; the reduction of the residual is $3.9e-8$ instead of $1e-8$.)

show linear convergence. The error of the cGdG-method is about four times smaller than that of the other two methods, however, the estimated order of convergence (e.o.c.) is only about 0.78.

Some performance-data are presented for the three methods in Table 10.2. We only consider the case $h_x = h_t$ to conserve space as the data are representative for all test cases. The sub-rows of Table 10.2 contain the following pieces of information:

- n_{\min} , n_{\max} – minimum/maximum number of iterations of the linear solver during the simulation,
- t_{steps} – the overall time (seconds) spent in the time-stepping-loop,
- t_{disc} – the part of t_{steps} (seconds) spent for discretization,
- M_{\max} – the maximum memory usage (megabytes) during the simulation.

The dimension of the linear system of equations is displayed in Table 10.5 for test case 3, which is more interesting in this regard as Γ deforms.

For all methods, the number of solver iterations grows with a factor between 2 and 3, if $h_x = h_t$ is halved. This might indicate that the condition number of the

linear system scales like h^{-1} in these cases. As we did not employ an optimal solver like a multigrid-solver, the condition number might still scale more favorably. For the same reason, the iteration numbers should only be used to compare the three methods with each other. The linear system of the cGdG-method seems to be slightly harder than that of the other two methods.

The simulation time is dominated by the discretization of the PDE for large values of h_x . For small values, the time to solve the linear systems starts to dominate, which may again be different for an optimal solver. As the dimensions of the linear system of the cGdG-method are twice as big as that of the other two methods, solving takes most of the time for the cGdG-method. The cGcG*-method is three to four times faster.

Peak memory consumption is similar for all methods. One can observe a fourfold increase, when h_x is halved, which is typical for two-dimensional problems. This demonstrates the effectivity of the adaptive refinement strategy close to Γ .

10.2. Rotating torus. The surfactant-transport-equation (1.1) is solved on a stationary torus defined as the zero-level of

$$\varphi(x) = ((R - |(x_1, x_2)^T|^2 + x_3^2)^{1/2} - r \quad \text{with } R = 0.7, r = 0.5.$$

The wind-field is tangential to the torus and shears depending on x_3 ,

$$\mathbf{w}(x) = (-x_2\omega(x), x_1\omega(x), 0)^T \quad \text{with } \omega(x) = 1 + x_3.$$

The maximum magnitude of \mathbf{w} on Γ is 1.2. The diffusion is set to $\alpha = 1$. The initial condition and the right-hand side f are computed, such that $u(x, t) = e^{-t}q_1q_2$ solves (1.1) for $0 < t \leq T = 1$, where

$$q(x, t) = \begin{pmatrix} \cos(\omega(x)t) & \sin(\omega(x)t) & \cdot \\ -\sin(\omega(x)t) & \cos(\omega(x)t) & \cdot \\ \cdot & \cdot & 1 \end{pmatrix} x.$$

Figure 10.3 shows the $L_t^\infty L_x^2$ -errors for the cGdG-method. The behavior is very similar to that for test case 1. The left plot shows the error for several fixed values of h_x as h_t varies. One can see convergence against the semidiscrete solution. In the right plot, the roles of h_x and h_t are reversed: h_x varies for several fixed values of h_t . Again, convergence against semidiscrete solutions can be observed. With respect to time, we observe order 2.7, with respect to space order 2.

Figure 10.4 shows the $L_t^\infty L_x^2$ -errors for the cGcG-method. The plots have the same layout as those for the cGdG-method in Figure 10.3. The behavior of the method is similar to that for test case 1. With respect to time, we observe order 2.1, with respect to space order 2. For $h_x = h_t$, the absolute size of the error is comparable to that of the cGdG-method.

Figure 10.5 shows the $L_t^\infty L_x^2$ -errors for the cGcG*-method. The plots have the same layout as those for the cGdG-method in Figure 10.3. Small differences between this method and the cGcG-method can be observed for the coarse discretizations. For these, the cGcG*-method produces marginally smaller errors. The methods use the same trial- and test-space, but the quadrature-schemes now yield different results. With respect to time, we observe order 2.1, with respect to space order 2. For $h_x = h_t$, the absolute size of the error is comparable to that of the cGdG-method.

The $L_t^2 H_x^1$ -errors of the three methods are shown in Table 10.3 for $h_t = h_x$. All three methods show comparable absolute errors. The mean e.o.c. for the cGdG-method is 0.93; for the cGcG-method it is 0.97, and for the cGcG*-method it is 0.96.

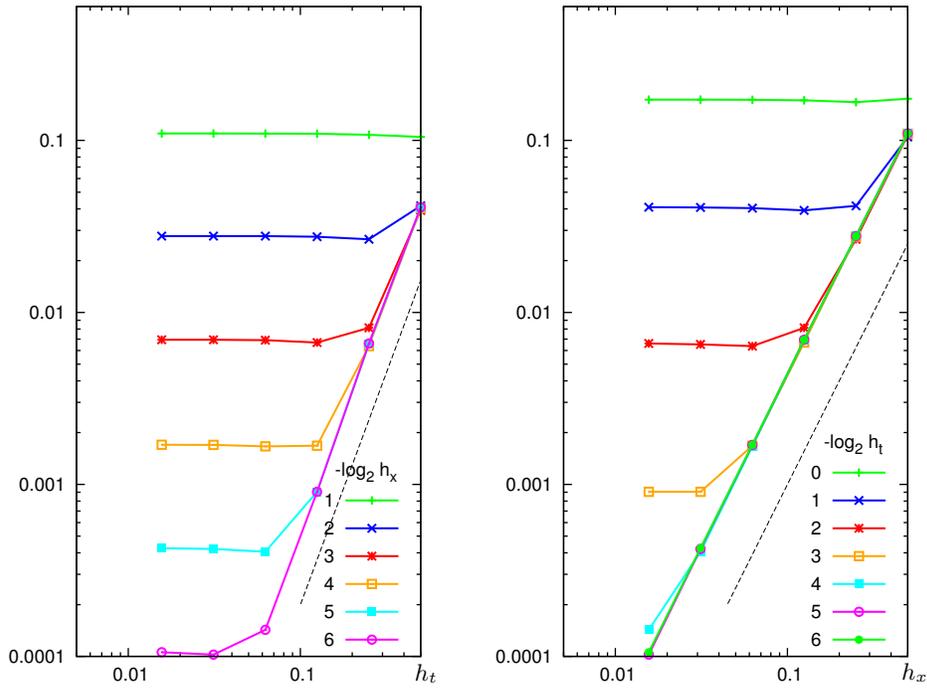


FIG. 10.3. Test case 2: $L_t^\infty L_x^2$ -error for the cGdG-method; left: $h_x = \text{const.}$, right: $h_t = \text{const.}$. The dotted lines have a slope of 2.7 and 2.

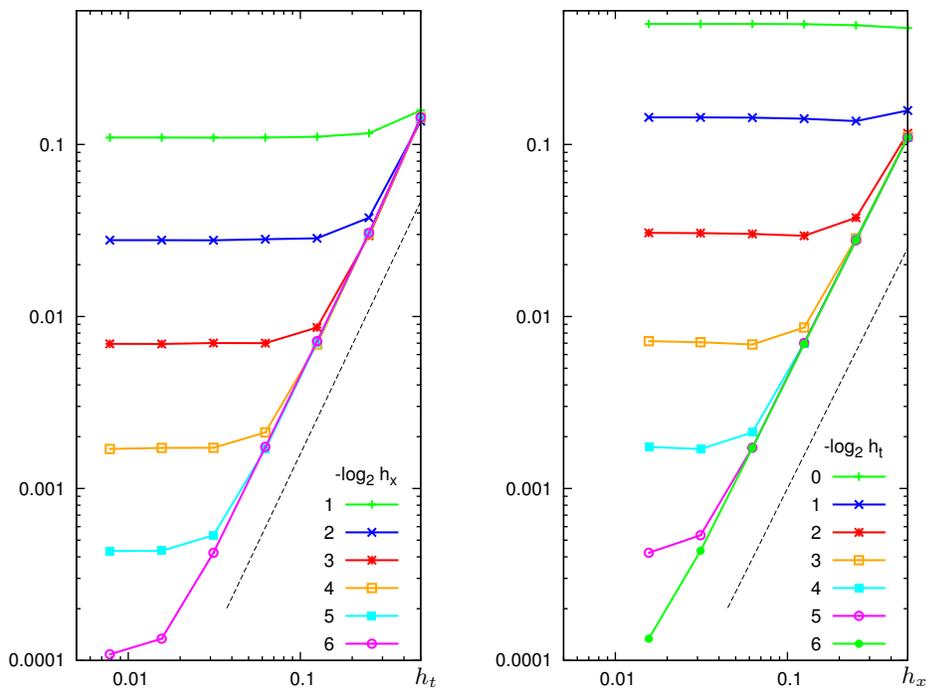


FIG. 10.4. Test case 2: $L_t^\infty L_x^2$ -error for the cGcG-method; left: $h_x = \text{const.}$, right: $h_t = \text{const.}$. The dotted lines have a slope of 2.1 and 2.

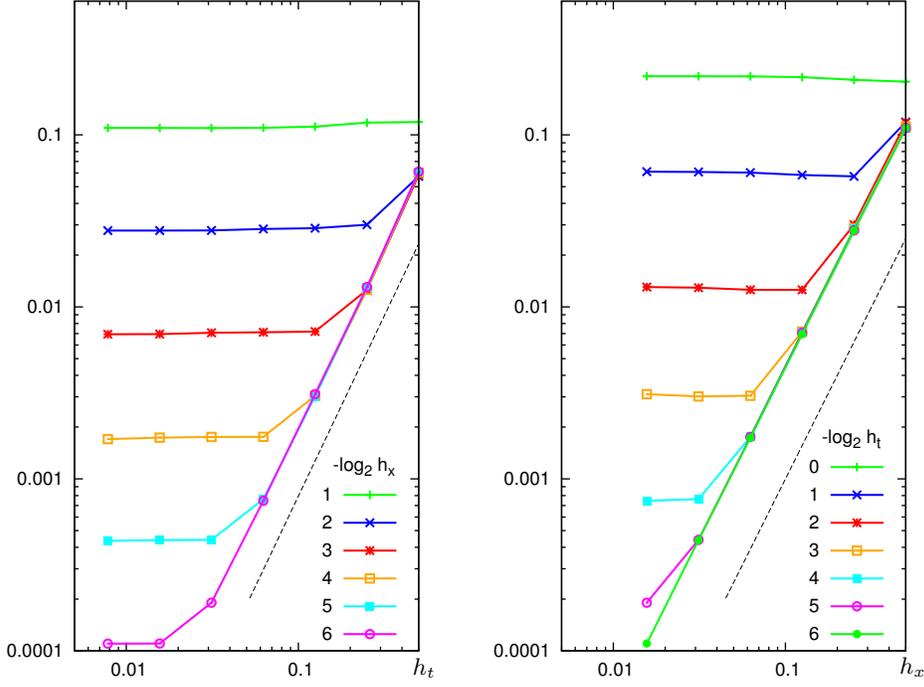


FIG. 10.5. Test case 2: $L_t^\infty L_x^2$ -error for the $cGcG^*$ -method; left: $h_x = \text{const.}$, right: $h_t = \text{const.}$ The dotted lines have a slope of 2.1 and 2.

$h_t = h_x$	0.5	0.25	0.125	0.0625	0.03125	0.015625
cGdG	0.41649	0.236876	0.123792	0.0638836	0.0326463	0.0165697
		0.814	0.936	0.954	0.969	0.978
cGcG	0.725294	0.386686	0.195397	0.0982215	0.049696	0.0250853
		0.907	0.985	0.992	0.983	0.986
cGcG*	0.688789	0.384255	0.195142	0.0981903	0.0496922	0.0250849
		0.841	0.978	0.991	0.983	0.986

TABLE 10.3

Test case 2: $L_t^2 H_x^1$ -error and estimated order of convergence (e.o.c.).

10.3. Sphere in a shear flow. The surfactant-transport-equation (1.1) is solved on a moving and deforming ellipsoid defined as the zero-level of

$$\varphi(x, t) = \left(\frac{x_1}{1 + 0.25 \sin(t)} \right)^2 + x_2^2 + x_3^2 - 1.$$

This last example is significantly different from the previous two because of the motion of the surface itself. The time-evolution of the domain, on which the PDE is posed, is the main motivation for the development of the methods in this paper.

The ellipsoid starts as a sphere, is dilated along the x_1 -axis and then contracted along the same axis. The wind-field is not tangential to $\Gamma(t)$,

$$\mathbf{w}(x, t) = 0.25 \frac{\cos(t)}{1 + 0.25 \sin(t)} x_1 (1, 0, 0)^T.$$

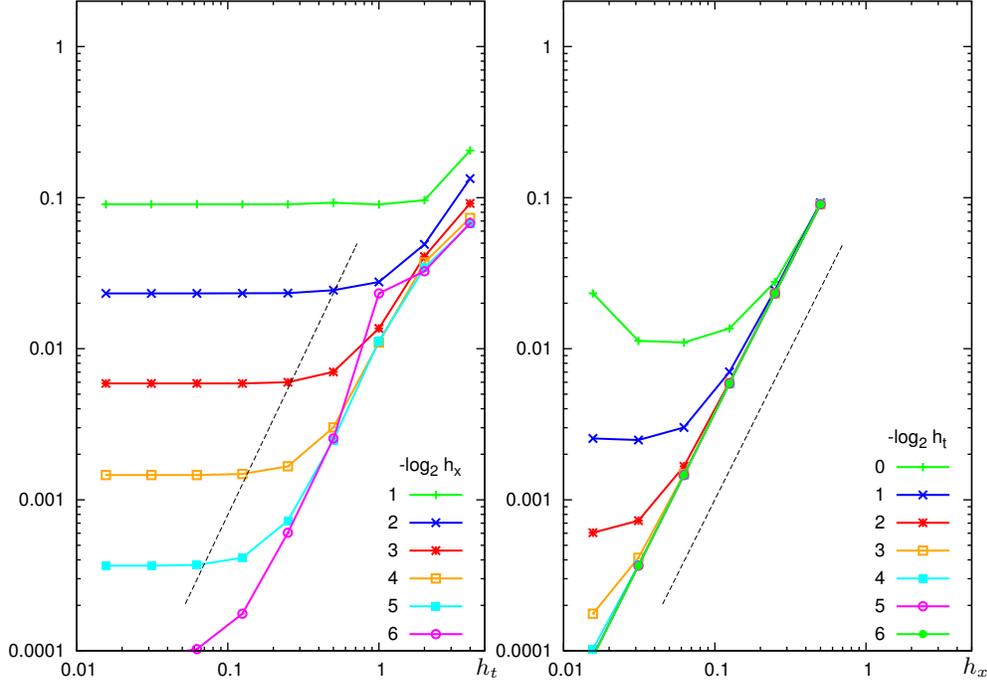


FIG. 10.6. Test case 3: $L_t^\infty L_x^2$ -error for the cGdG-method; left: $h_x = \text{const.}$, right: $h_t = \text{const.}$ The dotted lines have a slope of 2.1 and 2.

The diffusion is set to $\alpha = 1$. The initial condition and the right-hand side f are computed, such that $u(x, t) = e^{-t}x_1x_2$ solves (1.1) for $0 < t \leq T = 4$. As the time interval is longer than in the previous test cases, the data points with $h_x = h_t$ in the figures presented below lie in a different position compared to the test cases 1 and 2.

REMARK 10.1. For test case 3, two variants of the cGdG-method and the cGcG-method are implemented, which differ in the treatment of the material derivative and the term $D_\Gamma \cdot \mathbf{w}$. In variant one, the latter term must be discretized explicitly. This is Problem (6.3), respectively (5.1). In variant two, Reynold's theorem is applied to the material derivative. The term containing $D_\Gamma \cdot \mathbf{w}$ vanishes, but an integral over $\Gamma(t_i)$ appears, which is simpler to evaluate, both in terms of implementation-complexity and -speed. Therefore, the discretization is slightly faster.

However, the linear solver consistently needs more iterations, which increases the overall simulation-time. The accuracy of both methods differs only marginally. Consequently, we present no numbers for the variant two of both methods.

Figure 10.6 shows the $L_t^\infty L_x^2$ -errors for the cGdG-method. The left plot shows the error for several fixed values of h_x as h_t varies. One can see convergence against the semidiscrete solution. In the right plot, the roles of h_x and h_t are reversed: h_x varies for several fixed values of h_t . Again, convergence against semidiscrete solutions can be seen. With respect to time, we observe order 2.1, which is less than in the previous test cases. With respect to space, we observe order 2.

Figure 10.7 shows the $L_t^\infty L_x^2$ -errors for the cGcG-method. The plots have the same layout as those for the cGdG-method in Figure 10.6. With respect to time, we observe order 1.7, with respect to space order 2. For $h_x = h_t$, the absolute size of the error is comparable to that of the cGdG-method. Moreover, the e.o.c. in this case is

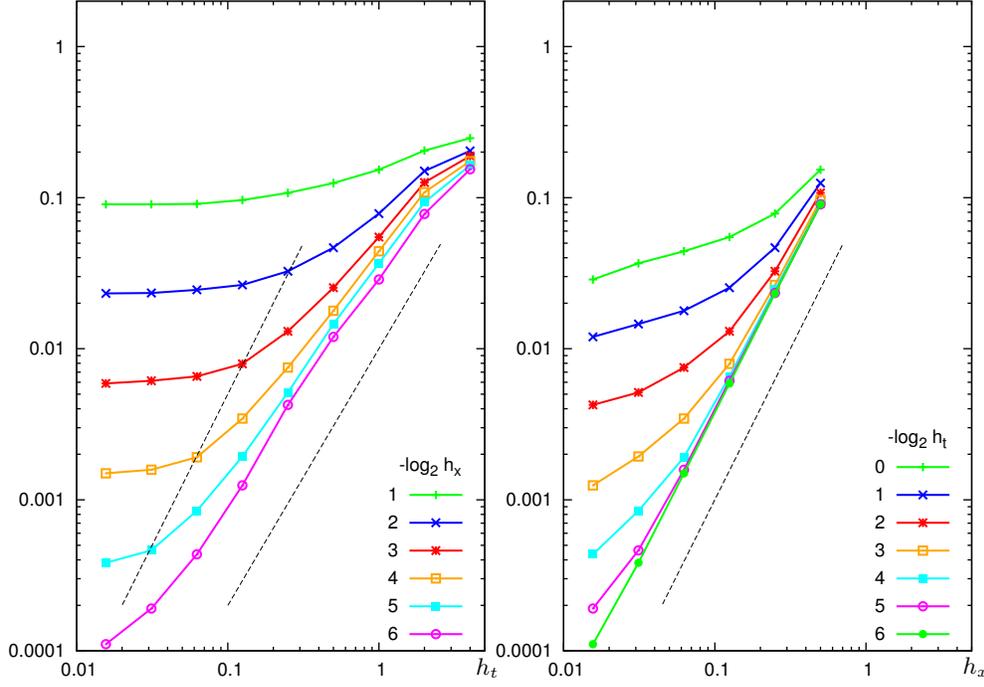


FIG. 10.7. Test case 3: $L_t^\infty L_x^2$ -error for the cGcG-method; left: $h_x = \text{const.}$, right: $h_t = \text{const.}$ The dotted lines have a slope of 2 and 1.7 on the left and of 2.1 on the right.

$h_t = h_x$	0.5	0.25	0.125	0.0625	0.03125	0.015625
cGdG	0.574872	0.303749	0.157731	0.0803239	0.0404533	0.020306
		0.920	0.945	0.974	0.990	0.994
cGcG	1.12954	0.550169	0.249556	0.112138	0.051407	0.0242118
		1.04	1.14	1.15	1.13	1.09
cGcG*	1.1038	0.721527	2.83051	0.83297	8.88659	6.15232
		0.613	-1.97	1.77	-3.42	0.531

TABLE 10.4

Test case 3: $L_t^2 H_x^1$ -error and estimated order of convergence (e.o.c.).

2.0, so the method is clearly of second order for $h_x = h_t$.

Figure 10.8 shows the $L_t^\infty L_x^2$ -errors for the cGcG*-method. The plots have the same layout as those for the cGdG-method in Figure 10.6. Convergence does not take place, neither in h_t nor in h_x . The clipped curve on the left reaches the value 44; the clipped values on the right are 8.6, 6.2 and 44. The method fails for this test case with a moving surface.

The $L_t^2 H_x^1$ -errors of the three methods are shown in Table 10.4 for $h_t = h_x$. The cGdG- and the cGcG-method show comparable absolute errors. The mean e.o.c. for the cGdG-method is 0.96, for the cGcG-method it is 1.1. For the cGcG*-method, there is no clear trend – the error even increases for small values of h_x, h_t .

REMARK 10.2. For standard parabolic PDEs, the discontinuous Galerkin method with discontinuous linear finite elements in time is known to have order 3 at the

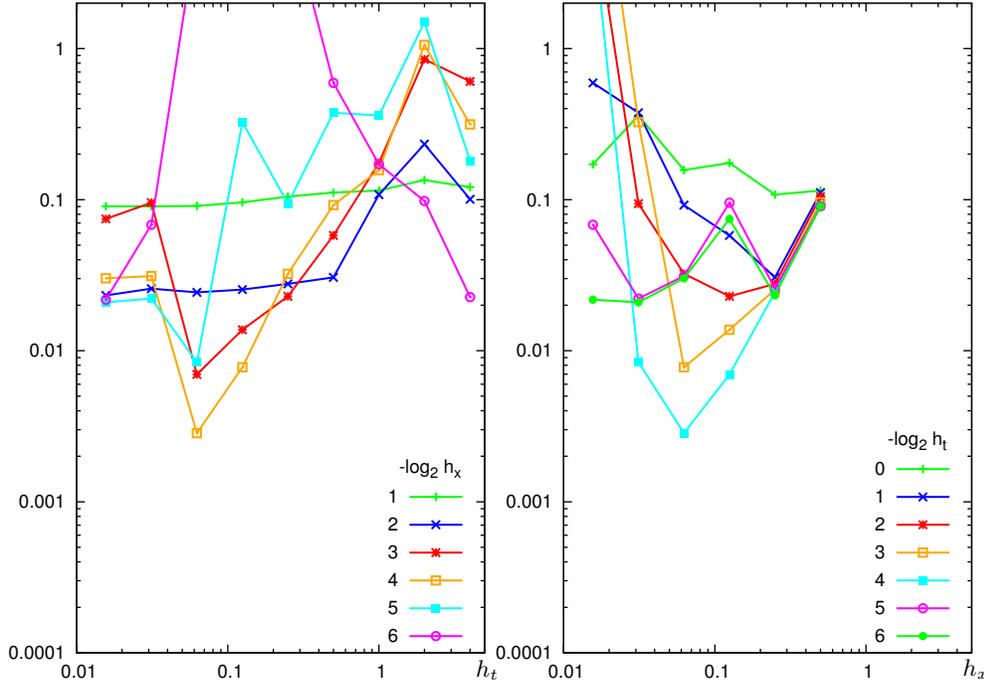


FIG. 10.8. Test case 3: $L_t^\infty L_x^2$ -error for the $cGcG^*$ -method; left: $h_x = \text{const.}$, right: $h_t = \text{const.}$

$h_t = h_x$		0.5	0.25	0.125	0.0625	0.03125	0.015625
cGdG	N_{\min}	200	864	3488	13728	55024	218784
	N_{\max}	272	1136	4440	17856	71256	285784
cGcG	N_{\min}	100	440	1788	7000	28228	111996
	N_{\max}	156	604	2324	9204	36152	144548
cGcG*	N_{\min}	100	412	1680	6620	26604	106596
	N_{\max}	136	548	2176	8808	35492	142188

TABLE 10.5

Test case 3: Minimum and maximum dimension of the system of linear equations in the simulations with $h_x = h_t$.

discrete time points t_i . This superconvergence property seems to hold in test case 1 and 2, where the interface is stationary. With the moving interface in test case 3, only order 2 in h_t is observed.

In this test case, the dimension of the system of linear equations varies as Γ moves and deforms. The minimum and maximum size are shown in Table 10.5. As expected, the dimension for the cGdG-method is roughly twice that of the cGcG-method because the initial values are eliminated in the latter. For the cGcG*-method, the dimension is a few per cent less than that of the cGcG-method. Generally, the dimension quadruples, if h_x is halved, which reflects the 2-dimensional character of the discrete problems in each time step.

11. Conclusion. We have introduced three new Eulerian finite element Galerkin methods for parabolic PDEs on moving surfaces and have examined their convergence properties in numerical experiments. The cGdG- and the cGcG-method seem to be the first Eulerian finite element methods for parabolic surface PDEs which are second order accurate in space and time.

Acknowledgments. I thank Arnold Reusken for reading and discussing the initial versions of this paper; this led to an improved presentation.

REFERENCES

- [1] G. DZIUK AND C.M. ELLIOTT, *An Eulerian approach to transport and diffusion on evolving implicit surfaces*, Computing and Visualization in Science, 13 (2010), pp. 17–28.
- [2] G. DZIUK AND C. M. ELLIOTT, *Finite elements on evolving surfaces*, IMA J. Numer. Anal., 27 (2007), pp. 262–292.
- [3] GERHARD DZIUK AND CHARLES M. ELLIOTT, *L^2 -estimates for the evolving surface finite element method*, Math. Comp., 82 (2013), pp. 1–24.
- [4] G. DZIUK, CH. LUBICH, AND D. MANSOUR, *Runge-Kutta time discretization of parabolic differential equations on evolving surfaces*, IMA J. Numer. Anal., 32 (2011), pp. 394–416.
- [5] K. ERIKSSON, D. ESTEP, P. HANSBO, AND C. JOHNSON, *Computational differential equations*, Cambridge University Press, Cambridge, 1996.
- [6] JOHN GREER, *An improvement of a recent Eulerian method for solving PDEs on general geometries*, Journal of Scientific Computing, 29 (2006), pp. 321–352. 10.1007/s10915-005-9012-5.
- [7] S. GROSS, J. PETERS, V. REICHEL, AND A. REUSKEN, *The DROPS package for numerical simulations of incompressible flows using parallel adaptive multigrid techniques*, IGPM-Report 211, IGPM, RWTH Aachen University, 2002.
- [8] SVEN GROSS AND ARNOLD REUSKEN, *Numerical methods for two-phase incompressible flows*, vol. 40 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2011.
- [9] ASHLEY J. JAMES AND JOHN LOWENGRUB, *A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant*, Journal of Computational Physics, 201 (2004), pp. 685 – 722.
- [10] MARTIN LENZ, SIMPLICE FIRMIN NEMADJIEU, AND MARTIN RUMPF, *A convergent finite volume scheme for diffusion on evolving surfaces*, SIAM J. Numer. Anal., 49 (2011), pp. 15–37.
- [11] METIN MURADOGLU AND GRETAR TRYGGVASON, *A front-tracking method for computation of interfacial flows with soluble surfactants*, Journal of Computational Physics, 227 (2008), pp. 2238 – 2262.
- [12] M. OLSHANSKII, A. REUSKEN, AND J. GRANDE, *A finite element method for elliptic equations on surfaces*, SIAM Journal on Numerical Analysis, 47 (2009), pp. 3339–3358.
- [13] MAXIM A. OLSHANSKII, ARNOLD REUSKEN, AND XIANMIN XU, *A Eulerian space-time finite element method for diffusion problems on evolving surfaces*, Preprint 362, IGPM, RWTH Aachen University, 2013.
- [14] C. POZRIKIDIS, *A finite-element method for interfacial surfactant transport, with application to the flow-induced deformation of a viscous drop*, J. Engrg. Math., 49 (2004), pp. 163–180.
- [15] STEVEN J. RUUTH AND BARRY MERRIMAN, *A simple embedding method for solving partial differential equations on surfaces*, Journal of Computational Physics, 227 (2008), pp. 1943 – 1961.
- [16] 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 (1990), pp. 111–112.
- [17] A. H. STROUD, *Approximate calculation of multiple integrals*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1971. Prentice-Hall Series in Automatic Computation.
- [18] JIAN-JUN XU, ZHILIN LI, JOHN LOWENGRUB, AND HONGKAI ZHAO, *A level-set method for interfacial flows with surfactant*, J. Comput. Phys., 212 (2006), pp. 590–616.
- [19] JIAN-JUN XU AND HONG-KAI ZHAO, *A Eulerian formulation for solving partial differential equations along a moving interface*, J. Sci. Comput., 19 (2003), pp. 573–594. Special issue in honor of the sixtieth birthday of Stanley Osher.