SIAM J. SCI. COMPUT.
Vol. 39, No. 4, pp. A1301–A1319

A TRACE FINITE ELEMENT METHOD FOR PDES ON EVOLVING SURFACES*

MAXIM A. OLSHANSKII† AND XIANMIN XU‡

Abstract. In this paper, we propose an approach for solving PDEs on evolving surfaces using a combination of the trace finite element method and a fast marching method. The numerical approach is based on the Eulerian description of the surface problem and employs a time-independent background mesh that is not fitted to the surface. The surface and its evolution may be given implicitly, for example, by the level set method. Extension of the PDE off the surface is *not* required. The method introduced in this paper naturally allows a surface to undergo topological changes and experience local geometric singularities. In the simplest setting, the numerical method is second-order accurate in space and time. Higher-order variants are feasible but not studied in this paper. We show results of several numerical experiments that demonstrate the convergence properties of the method and its ability to handle the case of the surface with topological changes.

Key words. surface PDEs, evolving surfaces, TraceFEM, level set method

AMS subject classifications. 65M60, 58J32

DOI. 10.1137/16M1099388

1. Introduction.

Partial differential equations on evolving surfaces arise in a number of mathematical models in natural sciences and engineering. Well-known examples include the diffusion and transport of surfactants along interfaces in multiphase fluids [30, 37, 50], diffusion-induced grain boundary motion [10, 36], and lipid interactions in moving cell membranes [23, 38]. Thus, recently there has been a significant interest in developing and analyzing numerical methods for PDEs on time-dependent surfaces. While all of finite difference, finite volumes, and finite element methods have been considered in the literature for numerical solution of PDEs on manifolds, in this work we focus on finite element methods.

The choice of a numerical approach for solving a PDE on evolving surface $\Gamma(t)$ largely depends on which of Lagrangian or Euclidian frameworks is used to set up the problem and describe the surface evolution. In [19, 21, 24], Elliott and coworkers developed and analyzed a finite element method (FEM) for computing transport and diffusion on a surface that is based on a Lagrangian tracking of the surface evolution. Some recent developments of the finite element methods for surface PDEs based on the Lagrangian description can be found, e.g., in [3, 5, 22, 35, 49]. If a surface undergoes strong deformations or topological changes or it is defined implicitly, e.g., as the zero level of a level set function, then numerical methods based on the Lagrangian approach have certain disadvantages. Methods using an Eulerian approach were developed in [1, 4, 51, 52], based on an extension of the surface PDE into a bulk domain that contains the surface. Although in the original papers finite differences were used, the approach is also suitable for finite element methods; see, e.g., [6]. A related technique

*Submitted to the journal's Methods and Algorithms for Scientific Computing section October 18, 2016; accepted for publication (in revised form) March 3, 2017; published electronically July 27, 2017.
http://www.siam.org/journals/sisc/39-4/M109938.html
Funding: Partially supported by NSF through the Division of Mathematical Sciences grants 1522252 and 1717516.
†Department of Mathematics, University of Houston, Houston, TX 77204-3008 (molshan@math.uh.edu, http://www.math.uh.edu/~molshan/).
‡LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, NCMIS, AMSS, Chinese Academy of Sciences, Beijing 100190, China (xmxu@lsec.cc.ac.cn).

is the closest point method in [45], where the closest point representation of the surface and differential operators is used in an ambient space to allow a standard Cartesian finite difference discretization method.

In the present paper, we develop yet another finite element method for solving a PDE on a time-dependent surface $\Gamma(t)$. The surface is embedded in a bulk computational domain. We assume a sharp representation of the surface rather than a diffusive interface approach typical for the phase-field models of interfacial problems. The level set method [48] is suitable for the purposes of this paper and will be used here to recover the evolution of the surface. We are interested in a surface FEM known in the literature as the trace or cut FEM. The trace finite element method uses the restrictions (traces) of a function from the background time-independent finite element space on the reconstructed discrete surface. This does not involve any mesh fitting toward the surface or an extension of the PDE.

The trace FEM method was originally introduced for elliptic PDEs on stationary surfaces in [41]. Further, the analysis and several extensions of the method were developed in the series of publications. This includes higher-order, stabilized, discontinuous Galerkin and adaptive variants of the method as well as applications to the surface–bulk coupled transport–diffusion problem, two-phase fluids with soluble surfactants, and coupled bulk-membrane elasticity problems; see, e.g., [7, 8, 9, 11, 13, 16, 28, 29, 34, 39, 44, 46]. There have been several successful attempts to extend the method to time-dependent surfaces. In [15], the trace FEM was combined with the narrowband unfitted FEM from [14] to devise an unfitted finite element method for parabolic equations on evolving surfaces. The resulting method preserves mass in the case of an advection-diffusion conservation law. The method based on a characteristic-Galerkin formulation combined with the trace FEM in space was proposed in [31]. Thanks to the semi-Lagrangian treatment of the material derivative (numerical integration back in time along characteristics), this variant of the method does not require stabilization for the dominating advection. The first-order convergence of the characteristic–trace FEM was demonstrated by a rigorous a priori error analysis and in numerical experiments. Another direction was taken in [43], where a space–time weak formulation of the surface problem was introduced. Based on this weak formulation, space–time variants of the trace FEM for PDEs on evolving surfaces were proposed in that paper and in [25]. The method from [43] employs discontinuous piecewise linear in time–continuous piecewise linear in space finite elements. In [40], the first-order convergence in space and time of the method in an energy norm and second-order convergence in a weaker norm was proved. In [25], the author experimented with both continuous and discontinuous in time piecewise linear finite elements.

In the space–time trace FEM, the trial and test finite element spaces consist of traces of standard volumetric elements on a space–time manifold resulting from the evolution of a surface. The implementation requires the numerical integration over the tetrahedral reconstruction of the 3D manifold embedded in the $\mathbb{R}^4$ ambient space. An efficient algorithm for such numerical reconstruction was suggested in [25] and implemented in the DROPS finite element package [17]. In [32], a stabilized version of the space–time trace FEM for coupled bulk-surface problems was implemented using Gauss-Lobatto quadrature rules in time. In this implementation, one does not reconstruct the 3D space–time manifold but instead needs the 2D surface approximations in the quadrature nodes. The numerical experience with space–time trace FEM based on quadrature rules in time is mixed. The authors of [32] reported a second-order convergence of the method for a number of 2D tests (in this case a 1D PDE is posed on an evolving curve), while in [25] one finds an example of a 2D smoothly deforming surface

when the space–time method based on the trapezoidal quadrature rule fails to deliver convergent results. The error analysis of such simplified versions is an open question.

Although the space–time framework is natural for the development of unfitted FEMs for PDEs on evolving surfaces, the implementation of such methods is not straightforward, especially if a higher-order method is desired. In this paper, we propose a variant of the trace FEM for time-dependent surfaces that uses simple finite difference approximations of *time* derivatives. It avoids any reconstruction of the surface–time manifold; it also avoids finding surface approximations at quadrature nodes. Instead, the method requires arbitrary (but smooth in a sense clarified later) extension of the numerical solution off the surface to a narrow strip around the surface. We stress that in the present method one does not extend either problem data or differential operators to a surface neighborhood as in the methods based on PDEs extension. At a given time node t_n, the degrees of freedom in the narrow strip (except those belonging to tetrahedra cut by the surface $\Gamma(t_n)$) do not contribute to algebraic systems but are only used to store the solution values from several previous time steps. In numerical examples, we use the BDF2 scheme for time integration, and so the narrow band degrees of freedom store the finite element solution for $t = t_{n-1}$ and $t = t_{n-2}$. To find a suitable extension, we apply a variant of the fast marching method (FMM); see, e.g., [30, 47]. At each time step, the trace FEM for a PDE on a steady surface $\Gamma(t_n)$ and the FMM are used in a modular way, which makes the implementation straightforward in a standard or legacy finite element software. For P1 background finite elements and BDF2 time-stepping scheme, numerical experiments show that the method is second-order accurate (assuming $\Delta t \simeq h$) and has no stability restrictions on the time step. We remark that the numerical methodology naturally extends to the surface-bulk coupled problems with propagating interfaces. However, in this paper we concentrate on the case when surface processes are decoupled from processes in the bulk.

The remainder of the paper is organized as follows. In section 2, we present the PDE model on an evolving surface and review some properties of the model. Section 3 introduces our variant of the trace FEM, which avoids space–time elements. Here we discuss implementation details. Section 4 collects the results for a series of numerical experiments. The experiments aim to access the accuracy of the method as well as the ability to solve PDEs along a surface undergoing topological changes. For the latter purpose, we consider the example of the diffusion of a surfactant on a surface of two colliding droplets.

2. Mathematical formulation. Consider a surface $\Gamma(t)$ passively advected by a smooth velocity field $\mathbf{w} = \mathbf{w}(\mathbf{x}, t)$; i.e., the normal velocity of $\Gamma(t)$ is given by $\mathbf{w} \cdot \mathbf{n}$, with $\mathbf{n}$ the unit normal on $\Gamma(t)$. We assume that for all $t \in [0, T]$, $\Gamma(t)$ is a smooth hypersurface that is closed ($\partial\Gamma = \emptyset$), connected, oriented, and contained in a fixed domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$. In the remainder, we consider $d = 3$, but all results have analogs for the case $d = 2$.

As an example of the surface PDE, consider the transport–diffusion equation modeling the conservation of a scalar quantity u with a diffusive flux on $\Gamma(t)$ (cf. [33]):

$$\dot{u} + (\mathrm{div}_\Gamma \mathbf{w})u - \nu \Delta_\Gamma u = 0 \quad \text{on} \quad \Gamma(t), \quad t \in (0, T], \tag{1}$$

with initial condition $u(\mathbf{x}, 0) = u_0(\mathbf{x})$ for $\mathbf{x} \in \Gamma_0 := \Gamma(0)$. Here $\dot{u}$ denotes the advective material derivative, $\mathrm{div}_\Gamma := \mathrm{tr}\left((I - \mathbf{n}\mathbf{n}^T)\nabla\right)$ is the surface divergence, Δ_Γ is the Laplace–Beltrami operator, and $\nu > 0$ is the constant diffusion coefficient. The well-posedness of suitable weak formulations of (1) has been proved in [19, 43, 2].

The equation (1) can be written in several equivalent forms; see [20]. In particular, for any smooth extension of u from the space–time manifold

$$\mathcal{G} := \bigcup_{t\in(0,T)} \Gamma(t)\times\{t\}, \quad \mathcal{G}\subset\mathbb{R}^4,$$

to a neighborhood of $\mathcal{G}$, one can expand the material derivative $\dot{u} = \frac{\partial u}{\partial t} + \mathbf{w}\cdot\nabla u$. Note that the identity holds independently of a smooth extension of u off the surface.

Assume further that the surface is defined implicitly as the zero level of the smooth level set function ϕ on $\Omega\times(0,T)$:

$$\Gamma(t) = \{\mathbf{x}\in\mathbb{R}^3 \,:\, \phi(t,\mathbf{x}) = 0\},$$

such that $|\nabla\phi|\geq c_0 > 0$ in $\mathcal{O}(\mathcal{G})$, a neighborhood of $\mathcal{G}$. One can consider an extension u^e in $\mathcal{O}(\mathcal{G})$ such that $u^e = u$ on $\mathcal{G}$ and $\nabla u^e\cdot\nabla\phi = 0$ in $\mathcal{O}(\mathcal{G})$. Note that u^e is smooth once ϕ and u are both smooth. Below we use the same notation u for the solution of the surface PDE (1) and its extension. We have the equivalent formulation,

$$\left\{\begin{array}{ll} \dfrac{\partial u}{\partial t} + \mathbf{w}\cdot\nabla u + (\mathrm{div}_\Gamma\mathbf{w})u - \nu\Delta_\Gamma u = 0 & \text{on } \Gamma(t), \\ \nabla u\cdot\nabla\phi = 0 & \text{in } \mathcal{O}(\Gamma(t)), \end{array}\right. \quad t\in(0,T]. \tag{2}$$

If ϕ is the signed distance function, the second equation in (2) defines the normal extension of u; i.e., the solution u stays constant in normal directions to $\Gamma(t)$. Otherwise, $\nabla u\cdot\nabla\phi = 0$ defines an extension, which is not necessarily the normal extension. In fact, any extension is suitable for our purposes if u is smooth function in a neighborhood of $\mathcal{G}$. We shall make an exception is section 3.2, where error analysis is reviewed and we need the normal extension to formulate certain estimates.

In the next section, we devise the trace FEM based on the formulation (2).

3. The finite element method.

We first collect some preliminaries and recall the trace FEM from [41] for the elliptic equations on stationary surfaces and some of its properties. Further, in section 3.3 we apply this method on each time step of a numerical algorithm for the transient problem (2).

3.1. Background mesh and induced surface triangulations.

Consider a tetrahedral subdivision $\mathcal{T}_h$ of the bulk computational domain Ω. We assume that the triangulation $\mathcal{T}_h$ is regular (no hanging nodes). For each tetrahedron $S\in\mathcal{T}_h$, let h_S denote its diameter and define the global parameter of the triangulation by $h = \max_S h_S$. We assume that $\mathcal{T}_h$ is shape regular; i.e., there exists $\kappa > 0$ such that for every $S\in\mathcal{T}_h$, the radius ρ_S of its inscribed sphere satisfies

$$\rho_S > h_S/\kappa. \tag{3}$$

For each time $t\in[0,T]$, denote by $\Gamma_h(t)$ a polygonal approximation of $\Gamma(t)$. We assume that $\Gamma_h(t)$ is a $C^{0,1}$ surface without boundary and that $\Gamma_h(t)$ can be partitioned in planar triangular segments:

$$\Gamma_h(t) = \bigcup_{T\in\mathcal{F}_h(t)} T, \tag{4}$$

where $\mathcal{F}_h(t)$ is the set of all triangular segments T. We assume that for any $T\in\mathcal{F}_h(t)$, there is only *one* tetrahedron $S_T\in\mathcal{T}_h$ such that $T\subset S_T$ (if T lies on a face shared by two tetrahedra, any of these two tetrahedra can be chosen as S_T).

For the level set description of $\Gamma(t)$, the polygonal surface $\Gamma_h(t)$ is defined by the finite element level set function as follows. Consider a continuous function $\phi_h(t,\mathbf{x})$

such that for any $t \in [0,T]$, the function ϕ_h is piecewise linear with respect to the triangulation $\mathcal{T}_h$. Its zero level set defines $\Gamma_h(t)$:

$$\Gamma_h(t) := \{\mathbf{x} \in \Omega \,:\, \phi_h(t,\mathbf{x}) = 0\}. \tag{5}$$

We assume that $\Gamma_h(t)$ is an approximation to $\Gamma(t)$. This is a reasonable assumption if either ϕ_h is an interpolant to the known ϕ or one finds ϕ_h as the solution to a discrete level set equation. In the latter case, one may have no direct knowledge of $\Gamma(t)$. Other interface capturing techniques, such as the volume of fluid method, also can be used subject to a postprocessing step to recover Γ_h.

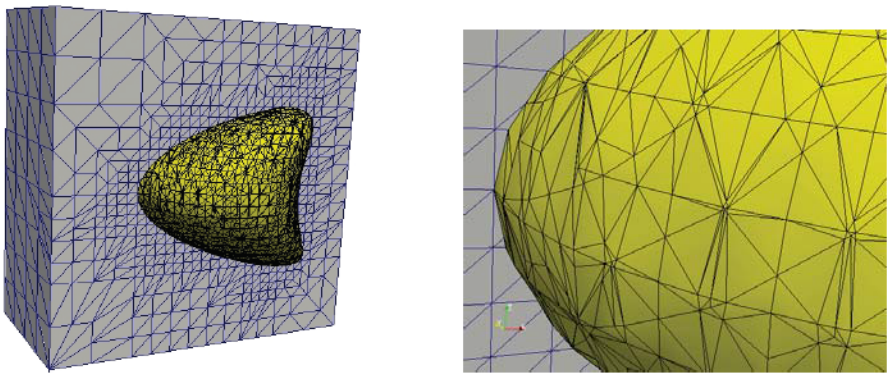
The intersection of $\Gamma_h(t)$ defined in (5) with any tetrahedron in $\mathcal{T}_h$ is either a triangle or a quadrilateral. If the intersection is a quadrilateral, we divide it into two triangles. This construction of $\Gamma_h(t)$ satisfies the assumptions made above. The bulk triangulation $\mathcal{T}_h$ consisting of tetrahedra and the induced surface triangulation are illustrated in Figure 1. There are no restrictions on how $\Gamma_h(t)$ cuts through the background mesh, and so for any fixed time instance t, the resulting triangulation $\mathcal{F}_h(t)$ is *not* necessarily regular. The elements from $\mathcal{F}_h(t)$ may have very small internal angles, and the size of neighboring triangles can vary strongly (cf. Figure 1 (right)). Thus, $\Gamma_h(t)$ is not a regular triangulation of $\Gamma(t)$. Two interesting properties of the induced surface triangulations are known in the literature [16, 42]: (i) If the background triangulation $\mathcal{T}_h$ satisfies the minimal angle condition (3), then the surface triangulation satisfies *maximum* angle condition, and (ii) any element from $\mathcal{F}_h(t)$ shares at least one vertex with a full size shape regular triangle from $\mathcal{F}_h(t)$. The trace finite element method does not exploit these properties directly, but they are still useful if one is interested in understanding the performance of the method.

We note that the surface triangulations $\mathcal{F}_h(t)$ will be used only to perform numerical integration in the finite element method below, while approximation properties of the method, as we shall see, depend on the volumetric tetrahedral mesh.

3.2. The trace FEM: Steady surface. To review the idea of the trace FEM, assume for a moment the stationary transport–diffusion problem on a steady closed smooth surface Γ:

$$\alpha u + \mathbf{w} \cdot \nabla u + (\mathrm{div}_\Gamma \mathbf{w})u - \nu \Delta_\Gamma u = f \quad \text{on } \Gamma. \tag{6}$$

Here, we assume $\alpha > 0$ and $\mathbf{w} \cdot \mathbf{n} = 0$. Integration by parts over Γ gives the weak formulation of (6): Find $u \in H^1(\Gamma)$ such that



(a) (b)

FIG. 1. *Left: Cut of the background and induced surface meshes for* $\Gamma_h(0)$ *from Experiment* 4 *in section* 4. *Right: The zoom-in of the surface mesh.*

$$\int_\Gamma (\alpha\, uv + \nu \nabla_\Gamma u \cdot \nabla_\Gamma v + (\mathbf{w}\cdot\nabla u)v + (\mathrm{div}_\Gamma \mathbf{w})uv)\, \mathrm{d}s = \int_\Gamma f v \,\mathrm{d}s \tag{7}$$

for all $v \in H^1(\Gamma)$. In the trace FEM, one substitutes Γ with Γ_h in (7) (Γ_h is constructed as in section 3.1) and instead of $H^1(\Gamma)$ considers the space of traces on Γ_h of all functions from the background ambient finite element space. This can be formally defined as follows.

Consider the volumetric finite element space of all piecewise linear continuous functions with respect to the bulk triangulation $\mathcal{T}_h$:

$$V_h := \{v_h \in C(\Omega) \mid v|_S \in P_1 \quad \forall\, S \in \mathcal{T}_h\}. \tag{8}$$

V_h induces the following space on Γ_h:

$$V_h^\Gamma := \{\psi_h \in C(\Gamma_h) \mid \exists\, v_h \in V_h \text{ such that } \ \psi_h = v_h \text{ on } \Gamma_h\}.$$

Given the surface finite element space V_h^Γ, the finite element discretization of (6) reads as follows: Find $u_h \in V_h^\Gamma$ such that

$$\int_{\Gamma_h} (\alpha\, u_h v_h + \nu \nabla_{\Gamma_h} u_h \cdot \nabla_{\Gamma_h} v_h + (\mathbf{w}\cdot\nabla u_h)v_h + (\mathrm{div}_{\Gamma_h} \mathbf{w})u_h v_h)\, \mathrm{d}s_h = \int_{\Gamma_h} f_h v_h \,\mathrm{d}s_h \tag{9}$$

for all $v_h \in V_h^\Gamma$. Here, f_h is an approximation of the problem source term on Γ_h.

From here and up to the end of this section, f^e denotes a *normal* extension of a quantity f from Γ. For a smooth closed surface, f^e is well defined in a neighborhood $\mathcal{O}(\Gamma)$. Assume that Γ_h approximates Γ in the following sense: It holds $\Gamma_h \subset \mathcal{O}(\Gamma)$ and

$$\|\mathbf{x} - \mathbf{p}(\mathbf{x})\|_{L^\infty(\Gamma_h)} + h\|\mathbf{n}^e - \mathbf{n}_h\|_{L^\infty(\Gamma_h)} \le c\, h^2, \tag{10}$$

where $\mathbf{n}_h$ is an external normal vector on Γ_h and $\mathbf{p}(\mathbf{x}) \in \Gamma$ is the closest surface point for $\mathbf{x}$. Given (10), the trace FEM is second-order accurate in the L^2 surface norm and first-order accurate in H^1 surface norm [41, 44]: For solutions of (6) and (9), it holds

$$\|u^e - u_h\|_{L^2(\Gamma_h)} + h\|\nabla_{\Gamma_h}(u^e - u_h)\|_{L^2(\Gamma_h)} \le c\, h^2,$$

with a constant c dependent only on the shape regularity of $\mathcal{T}_h$ and *independent of how the surface* Γ_h *cuts through the background mesh.* This robustness property is extremely useful for extending the method to time-dependent surfaces. It allows keeping the same background mesh while the surface evolves through the bulk domain, avoiding unnecessary mesh fitting and mesh reconstruction.

Before we consider the time-dependent case, a few important properties of the method should be mentioned. First, the authors of [15] noted that one can use the full gradient instead of the tangential gradient in the diffusion term in (9). This leads to the following FEM formulation: Find $u_h \in V_h$ such that

$$\int_{\Gamma_h} (\alpha\, u_h v_h + \nu \nabla u_h \cdot \nabla v_h + (\mathbf{w}\cdot\nabla u_h)v_h + (\mathrm{div}_{\Gamma_h} \mathbf{w})u_h v_h)\, \mathrm{d}s_h = \int_{\Gamma_h} f_h v_h \,\mathrm{d}s_h \tag{11}$$

for all $v_h \in V_h$. The rationality behind the modification is clear from the following observation. For the normal extension u^e of the solution u, we have $\nabla_\Gamma u = \nabla u^e$, and

so u^e satisfies the integral equality (7) with surface gradients (in the diffusion term) replaced by full gradients and for arbitrary smooth function v on Ω. Therefore, by solving (11), we recover u_h, which approximates the PDE solution u on the triangulated surface Γ_h *and* its normal extension u^e in the strip consisting of all tetrahedra cut by the surface Γ_h.

The formulation (11) uses the bulk finite element space V_h instead of the surface finite element space V_h^Γ in (9). However, the practical implementation of both methods uses the same frame of all bulk finite element nodal basis functions $\phi_i \in V_h$ such that $\mathrm{supp}(\phi_i) \cap \Gamma_h \neq \emptyset$. Hence, the active degrees of freedom in both methods are the same. The stiffness matrices are, however, different. For the case of the Laplace-Beltrami problem and a regular quasi-uniform tetrahedral grid, the studies in [15, 46] show that the conditioning of the (diagonally scaled) stiffness matrix of the method (11) improves over the conditioning of the matrix for (9), at the expense of a slight deterioration of the accuracy. Further in this paper, we shall use the full gradient version of the trace FEM.

From the formulations (11) or (9), we see that only those degrees of freedom of the background finite element space V_h are active (enter the system of algebraic equations) that are tailored to the tetrahedra cut by Γ_h. This provides us with a method of optimal computational complexity, which is not always the case for the methods based on an extension of surface PDE to the bulk domain. Due to the possible small cuts of bulk tetrahedra (cf. Figure 1), the resulting stiffness matrices can be poor conditioned. The simple diagonal rescaling of the matrices significantly improves the conditioning and eliminates outliers in the spectrum; see [39, 46]. Therefore, Krylov subspace iterative methods applied to the rescaled matrices are very efficient in solving the algebraic systems. Since the resulting matrices are sparse and resemble discretizations of 2D PDEs, using an optimized direct solver is also a suitable option.

3.3. The trace FEM: Evolving surface. For the evolving surface case, we extend the approach in such a way that the trace FEM (11) is applied on each time step for the recovered surface $\Gamma_h^n \approx \Gamma(t_n)$. Here and further, $\{t_n\}$, with $0 = t_0 < \cdots < t_n < \cdots < t_N = T$, is the temporal mesh, and u^n approximates $u(t_n)$. As before, V_h is a *time-independent* bulk finite element space with respect to the given background triangulation $\mathcal{T}_h$.

Assume that a smooth extension $u^e(\mathbf{x}, t)$ is available in $\mathcal{O}(\mathcal{G})$ and that

$$\Gamma(t_n) \subset \{\mathbf{x} \in \Omega \,:\, (\mathbf{x}, t_{n-1}) \in \mathcal{O}(\mathcal{G})\}\,. \tag{12}$$

In this case, one may discretize (2) in time using, for example, the implicit Euler method:

$$\frac{u^n - u^e(t_{n-1})}{\Delta t} + \mathbf{w}^n \cdot \nabla u^n + (\mathrm{div}_\Gamma \mathbf{w}^n) u^n - \nu \Delta_\Gamma u^n = 0 \quad \text{on } \Gamma(t_n), \tag{13}$$

$\Delta t = t_n - t_{n-1}$. Now we apply the trace FEM to solve (13) numerically. The trace FEM is a natural choice here since $\Gamma(t_n)$ is not fitted by the mesh. We look for $u_h^n \in V_h$ solving

$$\begin{aligned} &\int_{\Gamma_h^n} \left(\frac{1}{\Delta t} u_h^n v_h + (\mathbf{w}^n \cdot \nabla u_h^n) v_h + (\mathrm{div}_{\Gamma_h} \mathbf{w}^n) u_h^n v_h \right) \mathrm{d}s_h + \nu \int_{\Gamma_h^n} \nabla u_h^n \cdot \nabla v_h \, \mathrm{d}s_h \\ &\qquad = \int_{\Gamma_h^n} \frac{1}{\Delta t} u_h^{e,n-1} v_h \, \mathrm{d}s_h \end{aligned} \tag{14}$$

for all $v_h \in V_h$. Here, $u_h^{e,n-1}$ is a suitable extension of u_h^{n-1} from Γ_h^{n-1} to the surface neighborhood, $\mathcal{O}(\Gamma_h^{n-1})$. Condition (12) yields to the condition

$$\Gamma_h^n \subset \mathcal{O}(\Gamma_h^{n-1}). \tag{15}$$

Note that (15) is not a Courant condition on Δt but rather a condition on a width of a strip surrounding the surface, where the extension of the finite element solution is performed. Over one time step, a material point on the surface can travel a distance not exceeding $\|\mathbf{w}\|_{L^\infty}\Delta t$. Therefore, it is safe to extend the solution to all tetrahedra intersecting the strip of the width $2\|\mathbf{w}\|_{L^\infty}\Delta t$ surrounding the surface. Hence, we consider all tetrahedra having at least one vertex closer than $\|\mathbf{w}\|_{L^\infty}\Delta t$ to the surface: Define

$$\widetilde{\mathcal{S}}(\Gamma_h^n) := \{S \in \mathcal{T}_h \ : \ \exists\, x \in \mathcal{N}(S), \ \text{ s.t. } \ \operatorname{dist}(\mathbf{x}, \Gamma_h^n) < L\|\mathbf{w}\|_{L^\infty}\Delta t\}\,, \quad L = 1, \tag{16}$$

where $\mathcal{N}(S)$ is the set of all nodes for $S \in \mathcal{T}_h$. The criterion in (16) can be refined by exploiting the local information about $\mathbf{w}$ or about $\mathbf{n}\cdot\mathbf{w}$.

After we determine the numerical extension procedure, $u_h^k \to u_h^{e,k}$, the identity (14) defines the fully discrete numerical method.

In general, to find a suitable extension, one can consider a numerical solver for hyperbolic systems and apply it to the second equation in (2). For example, one can use a finite element method to solve the problem

$$\frac{\partial u^e}{\partial t'} + \nabla u^e \cdot \nabla \phi(t^k) = 0, \quad \text{such that } \ u^e = u_h^k \ \text{ on } \Gamma_h^k,$$

with the auxiliary time t', and let $u_h^{k,e} := \lim\limits_{t'\to\infty} u^e(t')$. Another technique to compute extensions (also used for the reinitialization of the signed distance function in the level set method) is the FMM [47]. We find the FMM technique convenient and fast for building suitable extensions in narrow bands of tetrahedra containing Γ_h. We give the details of the FMM in the next section.

We need one further notation. Denote by $\mathcal{S}(\Gamma_h^k)$ a strip of all tetrahedra cut by Γ_h^k:

$$\mathcal{S}(\Gamma_h^k) = \bigcup_{S\in\mathcal{T}_\Gamma^k} \overline{S}, \quad \text{with } \ \mathcal{T}_\Gamma^k := \{S \in \mathcal{T}_h : S \cap \Gamma_h^k \neq \emptyset\}.$$

We want to exploit the fact that the trace finite element method provides us with the normal extension in $\mathcal{S}(\Gamma_h^k)$ "for free" since the solution u_h^n of (14) approximately satisfies $\frac{\partial u_h^k}{\partial \mathbf{n}} = 0$ in $\mathcal{S}(\Gamma_h^k)$ by the property of the full gradient FEM formulation.

For given $u_h^{e,n-1}$ and $\phi_h(t_n)$, one time step of the algorithm now reads as follows:

1. Solve (14) for $u_h^n \in V_h$;
2. Apply the FMM to find $u_h^{e,n}$ in $\widetilde{\mathcal{S}}(\Gamma_h^n)\setminus\mathcal{S}(\Gamma_h^n)$ such that $u_h^{e,n} = u_h^n$ on $\partial\mathcal{S}(\Gamma_h^n)$.

If the motion of the surface is coupled to the solution of the surface PDE (the examples include two-phase flows with surfactant or some models of tumor growth [30, 12]), then a method to find an evolution of ϕ_h has to be added, while finding $u_h^{e,n}$ can be combined with a reinitialization of ϕ_h in the FMM.

A particular advantage of the present variant of the trace FEM for evolving domains is that the accuracy order in time can be easily increased using standard finite differences. In numerical experiments, we use the BDF2 scheme: The first term in (13) is replaced by

$$\frac{3u^n - 4u^e(t_{n-1}) + u^e(t_{n-2})}{2\Delta t},$$

and we set $L = 2$ in (16); the corresponding modifications in (14) are obvious. Furthermore, one may increase the accuracy order in space by using higher-order background

finite elements and a higher-order surface reconstruction; see [28, 34] for practical higher-order variants of trace FEM on stationary surfaces. In the framework of this paper, the use of these higher-order methods is decoupled from the numerical integration in time.

3.4. Extension by FMM. The FMM is a well-known technique to compute an approximate distance function to an interface embedded in a computational domain. Here, we build on the variant of the FMM from section 7.4.1 of [30] to compute finite element function extensions in a strip of tetrahedra. We need some further notations. For a vertex $\mathbf{x}$ of the background triangulation $\mathcal{T}_h$, $\mathcal{S}(\mathbf{x})$ denotes the union of all tetrahedra sharing $\mathbf{x}$. We fix t_n and let $\Gamma_h = \Gamma_h^n$, $\mathcal{S}(\Gamma_h) = \mathcal{S}(\Gamma_h^n)$. Note that we do not necessarily have a priori information of $\widetilde{\mathcal{S}}(\Gamma_h)$ since the distance function may not be available. Finding the narrow band for the extension is a part of the FMM below. We need the set of vertices from tetrahedra cut by the mesh:

$$\mathcal{N}_\Gamma = \{\mathbf{x} \in \mathbb{R}^3 \,:\, \mathbf{x} \in \mathcal{N}(S) \ \text{ for some } S \in \mathcal{S}(\Gamma_h)\}.$$

Assume that $u_h = u_h^n \in V_h$ solves (14) and we are interested in computing u_h^e in $\widetilde{\mathcal{S}}(\Gamma_h)$. The FMM is based on a greedy grid traversal technique and consists of two phases.

Initialization phase. In the tetrahedra cut by Γ_h, the full gradient trace FEM provides us with the normal extension. Hence, we set

$$u_h^e(\mathbf{x}) = u_h(\mathbf{x}) \quad \text{for } \mathbf{x} \in \mathcal{N}_\Gamma.$$

For the next step of FMM, we also need a distance function $d(\mathbf{x})$ for all $x \in \mathcal{N}_\Gamma$. For any $S_T \in \mathcal{S}(\Gamma_h)$, we know that $T = S_T \cap \Gamma_h$ is a triangle or quadrilateral with vertices $\{\mathbf{y}_j\}$, $j = 1, \dots, J$, where $J = 3$ or $J = 4$. Denote by $\mathbb{P}_T$ the plane containing T and by $P_h\mathbf{x}$ the projection of $\mathbf{x}$ on $\mathbb{P}_T$. Then, for each $\mathbf{x} \in \mathcal{N}(T)$, we define

$$d_T(\mathbf{x}) := \begin{cases} |\mathbf{x} - P_h\mathbf{x}| & \text{if } P_h\mathbf{x} \in T, \\ \min\limits_{1 \le j \le J} |\mathbf{x} - \mathbf{y}_j| & \text{otherwise.} \end{cases} \tag{17}$$

After we loop over all $S \in \mathcal{S}(\Gamma_h)$, the value $d(\mathbf{x})$ in each $\mathbf{x} \in \mathcal{N}_\Gamma$ is given by

$$d(\mathbf{x}) = \min_{S_T \in \mathcal{S}(\mathbf{x})} d_T(\mathbf{x}). \tag{18}$$

Extension phase. During this phase, we determine both $d(\mathbf{x})$ and $u_h^e(\mathbf{x})$ for $\mathbf{x} \in \mathcal{N} \setminus \mathcal{N}_\Gamma$. To this end, the set $\mathcal{N}$ of all vertices from $\mathcal{T}_h$ is divided into three subsets. A finished set $\mathcal{N}_f$ contains all vertices where d and u_h^e have already been defined. We initialize $\mathcal{N}_f = \mathcal{N}_\Gamma$. Initially the active set $\mathcal{N}_a$ contains all the vertices, which has a neighbor in $\mathcal{N}_f$:

$$\begin{aligned} \mathcal{N}_a &= \{\mathbf{x} \in \mathcal{N} \setminus \mathcal{N}_f \,:\, \mathcal{N}(\mathcal{S}(\mathbf{x})) \cap \mathcal{N}_f \neq \emptyset\}, \\ \mathcal{N}_u &= \mathcal{N} \setminus (\mathcal{N}_f \cup \mathcal{N}_a). \end{aligned}$$

The active set is updated during the FMM, and the method stops once $\mathcal{N}_a$ is empty.

For all $\mathbf{x} \in \mathcal{N}_a$, the FMM iteratively computes auxiliary distance function and extension function values $\tilde{d}(\mathbf{x})$ and $\tilde{u}_h^e(\mathbf{x})$, which become final values $d(\mathbf{x})$ and $u_h^e(\mathbf{x})$ once $\mathbf{x}$ leaves $\mathcal{N}_a$ and joins $\mathcal{N}_f$. The procedure is as follows: For $\mathbf{x} \in \mathcal{N}_a$, we consider all $S \in \mathcal{S}(\mathbf{x})$ such that $\mathcal{N}(S) \cap \mathcal{N}_f \neq \emptyset$. If $\mathcal{N}(S) \cap \mathcal{N}_f$ contains only one vertex $\mathbf{y}$, we set

$$\tilde{d}_S(\mathbf{x}) = d(\mathbf{y}) + |\mathbf{x} - \mathbf{y}|, \quad \tilde{u}_{h,S}(\mathbf{x}) = u_h^e(\mathbf{y}).$$

If $\mathcal{N}(S)\cap\mathcal{N}_f$ contains two or three vertices $\{\mathbf{y}_j\}$, $1\le j\le J$, $J=2$ or 3, then we compute

$$\tilde{d}_S(\mathbf{x})=\begin{cases} d(P_h\mathbf{x})+|\mathbf{x}-P_h\mathbf{x}|, & \text{if } P_h\mathbf{x}\in S,\\ d(\mathbf{y}_{min})+|\mathbf{x}-\mathbf{y}_{min}|, & \text{otherwise,}\end{cases}$$
$$\tilde{u}_{h,S}(\mathbf{x})=\begin{cases} u_h^e(P_h\mathbf{x}), & \text{if } P_h\mathbf{x}\in S,\\ u_h^e(\mathbf{y}_{min}), & \text{otherwise,}\end{cases}$$

where $\mathbf{y}_{min}=\operatorname{argmin}_{1\le j\le m}(d(\mathbf{y}_j)+|\mathbf{x}-\mathbf{y}_j|)$ and $P_h\mathbf{x}$ is the orthogonal projection of $\mathbf{x}$ on the line passing through $\{\mathbf{y}_j\}$ (if $J=2$) or the plane containing $\{\mathbf{y}_j\}$ (if $J=3$). The value of $d(P_h\mathbf{x})$ is computed as the linear interpolation of the known values $d(\mathbf{y}_j)$. Now we set

$$\begin{aligned}\tilde{d}(\mathbf{x})&=\tilde{d}_{S_{min}}(\mathbf{x}),\\ \tilde{u}_h(\mathbf{x})&=\tilde{u}_{h,S_{min}}(\mathbf{x}),\end{aligned}\quad \text{for } S_{min}=\operatorname{argmin}\{\tilde{d}_S(\mathbf{x}) \ : \ S\in\mathcal{S}(\mathbf{x}) \text{ and } \mathcal{N}(S)\cap\mathcal{N}_f\neq\emptyset\}.$$

We determine such vertex $\mathbf{x}_{min}\in\mathcal{N}_a$ that $d(\mathbf{x}_{min})=\min_{\mathbf{x}\in\mathcal{N}_a}\tilde{d}(\mathbf{x})$ and set

$$d(\mathbf{x}_{min})=\tilde{d}(\mathbf{x}_{min}),\quad u_h^e(\mathbf{x}_{min})=\tilde{u}_h(\mathbf{x}_{min}).$$

The vertex $\mathbf{x}_{min}$ is now moved from the active set $\mathcal{N}_a$ to the finalized set $\mathcal{N}_f$. Based on the value of $d(\mathbf{x}_{min})$, one checks if any tetrahedron from $\mathcal{S}(\mathbf{x}_{min})$ may belong to $\widetilde{\mathcal{S}}(\Gamma_h)$ strip. If such $S\in\widetilde{\mathcal{S}}(\Gamma_h)$ exists, then $\mathcal{N}_a$ is updated by vertices from $\mathcal{N}_u$ connected with $\mathbf{x}_{\min}$. Otherwise, no new vertices are added to $\mathcal{N}_a$. In our implementation, we use the simple criterion: If it holds

$$d(\mathbf{x}_{min})>h+L|\mathbf{w}|_\infty\Delta t,$$

then we do not update $\mathcal{N}_a$ with new vertices from $\mathcal{N}_u$.

4. Numerical examples. This section collects the results of several numerical experiments for a number of problems posed on evolving surfaces. The results demonstrate the accuracy of the trace FEM, its stability with respect to the variation of discretization parameters, and the ability to handle the case when the transport–diffusion PDE is solved on a surface undergoing topological changes.

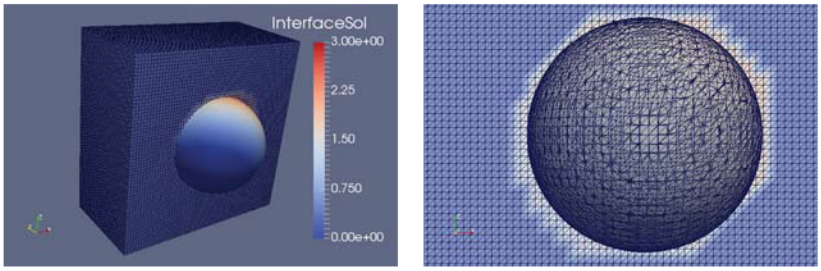
All implementations are done in the finite element package DROPS [17]. The background finite element space V_h consists of piecewise linear continuous finite elements. The BDF2 scheme is applied to approximate the time derivative. At each time step, we assemble the stiffness matrix and the right-hand side by numerical integration over the discrete surfaces Γ_h^n. A Gaussian quadrature of degree 5 is used for the numerical integration on each $K\in\mathcal{F}_h$. The same method is used to evaluate the finite element error. All linear algebraic systems are solved using the GMRES iterative method with the Gauss–Seidel preconditioner to a relative tolerance of 10^{-6}.

The first series of experiments verifies the formal accuracy order of the method for the examples with known analytical solutions.

Experiment 1. We consider the transport–diffusion equation (1) on the unit sphere $\Gamma(t)$ moving with the constant velocity $\mathbf{w}=(0.2,0,0)$. The initial data are given by

$$\Gamma(0):=\{\mathbf{x}\in\mathbb{R}^3 \ : \ |\mathbf{x}|=1\},\quad u|_{t=0}=1+x_1+x_2+x_3.$$

One easily checks that the exact solution is given by $u(\mathbf{x},t)=1+(x_1+x_2+x_3-0.2t)\exp(-2t)$. In this and the next two experiments, we set $T=1$.



(a) (b)

FIG. 2. *The cut of the background mesh and a part of the surface mesh for $t_n = 1$. Colors illustrate the solution and its extension.*

The computational domain is $\Omega = [-2,2]^3$. We divide Ω into tetrahedra as follows: First, we apply the uniform tessellation of Ω into cubes with side length h. Further, the Kuhn subdivision of each cube into six tetrahedra is applied. This results in the shape regular background triangulation $\mathcal{T}_h$. The finite element level set function $\phi_h(\mathbf{x},t)$ is the nodal Lagrangian P1 interpolant for the signed distance function of $\Gamma(t)$ and

$$\Gamma_h^n = \{\mathbf{x} \in \mathbb{R}^3 \ : \ \phi_h(\mathbf{x},t_n) = 0\}.$$

The temporal grid is uniform, $t_n = n\Delta t$. We note that in all experiments, we apply the FMM to find both distances to Γ_h^n and u^e, so we never explore the explicit knowledge of the distance function for $\Gamma(t)$.

Figure 2 shows the cut of the background mesh and the surface mesh colored by the computed solution at time $t = 1$. We are interested in the $L^2(H^1)$ and $L^2(L^2)$ surface norms for the error. We compute them using the trapezoidal quadrature rule in time:

$$\begin{aligned} err_{L^2(H^1)} = \Big\{ &\frac{\Delta t}{2}\|\nabla_{\Gamma_h}(u^e - \pi_h u)\|^2_{L^2(\Gamma_h^0)} + \sum_{i=1}^{N-1} \Delta t \|\nabla_{\Gamma_h}(u^e - u_h)\|^2_{L^2(\Gamma_h^i)} \\ &+ \frac{\Delta t}{2}\|\nabla_{\Gamma_h}(u^e - u_h)\|^2_{L^2(\Gamma_h^n)} \Big\}^{1/2}, \\ err_{L^2(L^2)} = \Big\{ &\frac{\Delta t}{2}\|(u^e - \pi_h u)\|^2_{L^2(\Gamma_h^0)} + \sum_{i=1}^{N-1} \Delta t \|(u^e - u_h)\|^2_{L^2(\Gamma_h^i)} \\ &+ \frac{\Delta t}{2}\|(u^e - u_h)\|^2_{L^2(\Gamma_h^N)} \Big\}^{1/2}. \end{aligned}$$

Tables 1 and 2 present the error norms for the Experiment 1 with various time steps Δt and mesh sizes h. If one refines both Δt and h, the first order of convergence in the surface $L^2(H^1)$-norm and the second order in the surface $L^2(L^2)$-norm are clearly seen. For the case of large Δt and small h, the FMM extension strip $\widetilde{S}(\Gamma_h) \setminus S(\Gamma_h)$ becomes wider in terms of characteristic mesh size h, and the accuracy of the method diminishes. This numerical phenomenon can be noted in the top rows of Tables 1 and 2, where the error increases as the mesh size becomes smaller. We expect that the situation improves if one applies more accurate extension methods in $\widetilde{S}(\Gamma_h) \setminus S(\Gamma_h)$. One candidate would be the normal derivative volume stabilization method from [26] extended to all tetrahedra in $\widetilde{S}(\Gamma_h)$.

TABLE 1
The $L^2(H^1)$-norm of the error in Experiment 1.

	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$
$\Delta t = 1/8$	**0.96365**	0.835346	1.221340	2.586520
$\Delta t = 1/16$	0.963654	**0.74794**	0.423799	0.653380
$\Delta t = 1/32$	0.954179	0.759253	**0.37954**	0.225399
$\Delta t = 1/64$	0.953155	0.766650	0.381567	**0.19143**

TABLE 2
The $L^2(L^2)$-norm of the error in Experiment 1.

	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$
$\Delta t = 1/8$	**0.39351**	0.192592	0.319912	0.691862
$\Delta t = 1/16$	0.435067	**0.16268**	0.057801	0.107322
$\Delta t = 1/32$	0.445765	0.172543	**0.04013**	0.018707
$\Delta t = 1/64$	0.448433	0.175145	0.041875	**0.01040**

TABLE 3
Averaged CPU times per each time step of the method in Experiment 1.

	Active d.o.f.	Extra d.o.f.	T_{assemb}	T_{solve}	T_{ext}
$h_0 = 1/2,\ \Delta t_0 = 1/8$	31	8	0.0038	0.0004	0.0012
$h = h_0/2,\ \Delta t = \Delta t_0/2$	104	24	0.0160	0.0021	0.0041
$h = h_0/4,\ \Delta t = \Delta t_0/4$	452	63	0.0708	0.0087	0.0195
$h = h_0/8,\ \Delta t = \Delta t_0/8$	1880	170	0.3814	0.0374	0.0906

Table 3 shows the breakdown of the computational costs of the method into the averaged CPU times for assembling stiffness matrices, solving resulting linear algebraic systems, and performing the extension to $\widetilde{S}(\Gamma_h) \setminus S(\Gamma_h)$ by FMM. Since the surface evolves, all the statistics slightly vary in time, and so the table shows averaged numbers per one time step. "Active d.o.f." is the dimension of the linear algebraic system, i.e., the number of bulk finite element nodal values tailored to tetrahedra from $S(\Gamma_h)$. "Extra d.o.f." is the number of mesh nodes in $\widetilde{S}(\Gamma_h) \setminus \overline{S(\Gamma_h)}$; these are all nodes where extension is computed by FMM. The averaged CPU times demonstrate optimal or close to the optimal scaling with respect to the number of degrees of freedom. As common for a finite element method, the most time consuming part is the assembling of the stiffness matrices. The costs of FMM are modest compared to the assembling time, and T_{solve} indicates that using a preconditioned Krylov subspace method is the efficient approach to solve linear algebraic systems (no extra stabilizing terms were added to the FE formulation for improving its algebraic properties).

Experiment 2. The setup of this experiment is similar to the previous one. The transport velocity is given by $\mathbf{w} = (-2\pi x_2, 2\pi x_1, 0)$. Initially, the sphere is set off the center of the domain: The initial data are given by

$$\Gamma(0) := \{\mathbf{x} \in \mathbb{R}^3 \ : \ |\mathbf{x} - \mathbf{x}_0| = 1\}, \quad u|_{t=0} = 1 + (x_1 - 0.5) + x_2 + x_3,$$

with $\mathbf{x}_0 = (0.5, 0, 0)$. Now $\mathbf{w}$ revolves the sphere around the center of domain without changing its shape. One checks that the exact solution to (1) is given by

$$u(\mathbf{x}, t) = (x_1(\cos(2\pi t) - \sin(2\pi t)) + x_2(\cos(2\pi t) + \sin(2\pi t)) + x_3 + 0.5)\exp(-2t).$$

Tables 4 and 5 show the error norms for Experiment 2 with various time steps Δt and mesh sizes h. If one refines both Δt and h, the first order of convergence in the surface $L^2(H^1)$-norm and the second order in the surface $L^2(L^2)$-norm are again

TABLE 4
The $L^2(H^1)$-norm of the error in Experiment 2.

	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$
$\Delta t = 1/32$	0.978459	1.931081	3.740840	4.048480
$\Delta t = 1/64$	**0.90425**	0.690963	0.813820	1.066030
$\Delta t = 1/128$	0.901234	**0.64014**	0.348516	0.300654
$\Delta t = 1/256$	0.901443	0.640055	**0.32352**	0.171101
$\Delta t = 1/512$	0.901631	0.641018	0.323199	**0.16286**

TABLE 5
The $L^2(L^2)$-norm of the error in Experiment 2.

	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$
$\Delta t = 1/32$	0.294122	0.548567	0.975485	0.958447
$\Delta t = 1/64$	**0.27244**	0.120061	0.152520	0.175920
$\Delta t = 1/128$	0.279106	**0.10451**	0.034962	0.037085
$\Delta t = 1/256$	0.279744	0.105975	**0.02699**	0.010692
$\Delta t = 1/512$	0.279811	0.106116	0.026444	**0.00736**

observed. Note that the transport velocity $\|\mathbf{w}\|_\infty \approx 9.42$ in this experiment scales differently compared to Experiment 1. Therefore, we consider smaller Δt to obtain meaningful results.

Experiment 3. In this experiment, we consider a shrinking sphere and solve (1) with a source term on the right-hand side. The bulk velocity field is given by $\mathbf{w} = -\frac{1}{2}e^{-t/2}\mathbf{n}$, where $\mathbf{n}$ is the unit outward normal on $\Gamma(t)$. $\Gamma(0)$ is the unit sphere. One computes $\mathrm{div}_\Gamma \mathbf{w} = -1$. The prescribed analytical solution $u(\mathbf{x}, t) = (1 + x_1x_2x_3)e^t$ solves (1) with the right-hand side $f(\mathbf{x}, t) = (-1.5e^t + 12e^{2t})x_1x_2x_3$.

Tables 6 and 7 show the error norms for various time steps Δt and mesh sizes h. If one refines both Δt and h, the first order of convergence in the surface $L^2(H^1)$-norm and the second order in the surface $L^2(L^2)$-norm are observed for the example of the shrinking sphere.

Experiment 4. In this example, we consider a surface transport–diffusion problem as in (1) on a more complex moving manifold. The initial manifold and concentration are given (as in [18]) by $\Gamma(0) = \{\, \mathbf{x} \in \mathbb{R}^3 \;:\; (x_1 - x_3^2)^2 + x_2^2 + x_3^2 = 1 \,\}$, $u_0(\mathbf{x}) = 1 + x_1x_2x_3$. The velocity field that transports the surface is

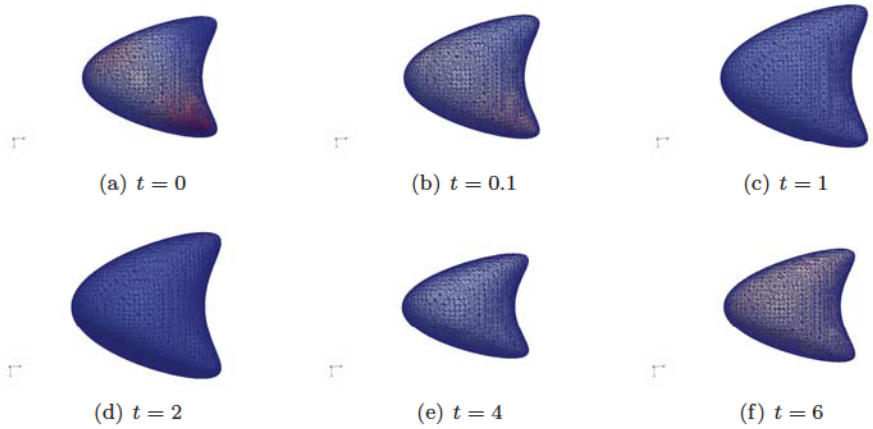
$$\mathbf{w}(\mathbf{x}, t) = \big(0.1x_1 \cos(t), 0.2x_2 \sin(t), 0.2x_3 \cos(t)\big)^T.$$

TABLE 6
The $L^2(H^1)$-norm of the error in Experiment 3.

	$h = 1/4$	$h = 1/8$	$h = 1/16$	$h = 1/32$
$\Delta t = 1/16$	**0.48893**	0.311146	0.170104	0.088521
$\Delta t = 1/32$	0.481896	**0.30859**	0.168635	0.087013
$\Delta t = 1/64$	0.478675	0.307416	**0.16801**	0.086513
$\Delta t = 1/128$	0.477226	0.306872	0.167747	**0.08634**

TABLE 7
The $L^2(L^2)$-norm of the error in Experiment 3.

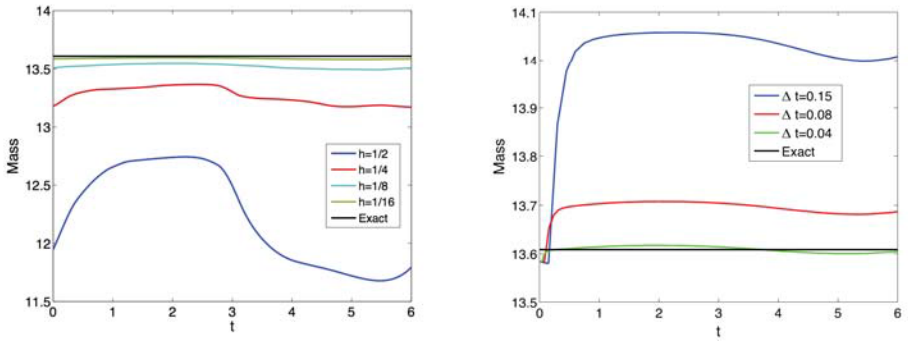
	$h = 1/4$	$h = 1/8$	$h = 1/16$	$h = 1/32$
$\Delta t = 1/16$	**0.12237**	0.0468812	0.0224705	0.0178009
$\Delta t = 1/32$	0.116763	**0.040745**	0.0130912	0.0060213
$\Delta t = 1/64$	0.115589	0.0396511	**0.011517**	0.0035199
$\Delta t = 1/128$	0.115336	0.0394023	0.0112094	**0.003038**



(a) $t = 0$ (b) $t = 0.1$ (c) $t = 1$

(d) $t = 2$ (e) $t = 4$ (f) $t = 6$

FIG. 3. *Snapshots of the surface, surface mesh, and the computed solution from Experiment* 4.



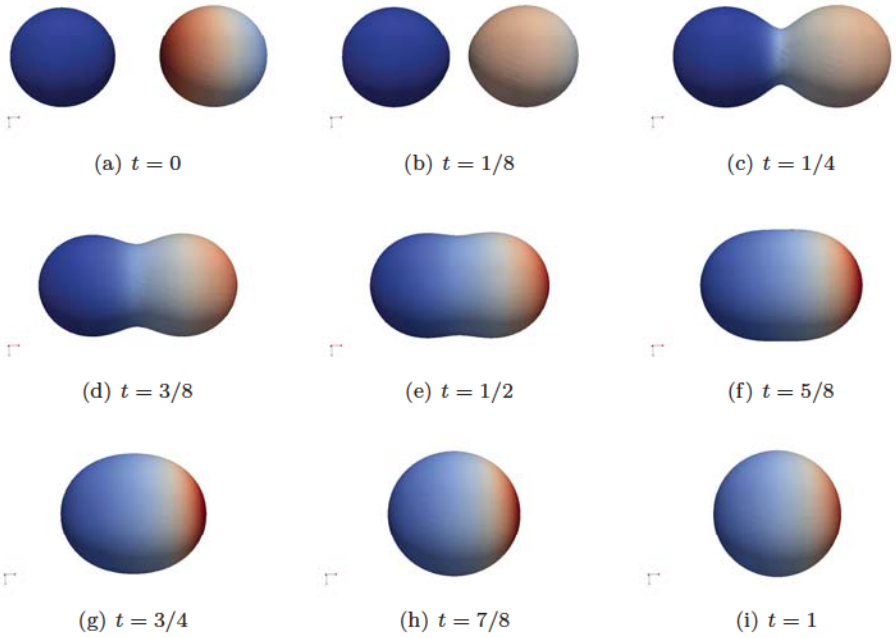
(a) (b)

FIG. 4. *Total mass evolution for the finite element solution in Experiment* 4.

We compute the problem until $T = 6$. In this example, the total mass $M(t) = \int_{\Gamma(t)} u(\cdot, t)\, \mathrm{d}s$ is conserved and equal to $M(0) = |\Gamma(0)| \approx 13.6083$. We check how well the discrete quantity $M_h(t) = \int_{\Gamma_h(t)} u_h(\cdot, t)\, \mathrm{d}s$ is conserved. In Figure 4 (left), we plot $M_h(t)$ for different mesh sizes h and a fixed time step $\Delta t = 0.01$. The error in the total mass at $t = T$ is equal to 1.8142, 0.4375, 0.1006, and 0.0239 (for mesh sizes as in Figure 4 (left)). In Figure 4 (right), we plot $M_h(t)$ for different time steps Δt and a fixed mesh size $h = 1/16$. The error in the total mass at $t = T$ is equal 0.3996, 0.0785, and 0.0038 (for time steps as in Figure 4 (right)). The error in the mass conservation is consistent with the expected second-order accuracy in time and space.

If one is interested in the exact mass conservation on the discrete level, then one may enforce $M_h(t_n) = M_h(0)$ as a side constrain in the finite element formulation (14) with the help of the scalar Lagrange multiplier; see [32]. Here we used the error reduction in total mass as an *indicator* of the method convergence order for the case when the exact solution is not available.

Experiment 5. In this test problem from [27], one solves the transport–diffusion equation (1) on an evolving surface $\Gamma(t)$, which undergoes a change of topology and



(a) $t = 0$ (b) $t = 1/8$ (c) $t = 1/4$

(d) $t = 3/8$ (e) $t = 1/2$ (f) $t = 5/8$

(g) $t = 3/4$ (h) $t = 7/8$ (i) $t = 1$

FIG. 5. *Snapshots of discrete solution in Experiment* 5 *with* $h = 1/16$, $\Delta t = 1/128$.

experiences a local singularity. The computational domain is $\Omega = (-3, 3) \times (-2, 2)^2$, $t \in [0, 1]$. The evolving surface is the zero level of the level set function ϕ defined as

$$\phi(\mathbf{x}, t) = 1 - \frac{1}{\|\mathbf{x} - c_+(t)\|^3} - \frac{1}{\|\mathbf{x} - c_-(t)\|^3},$$

with $c_\pm(t) = \pm\frac{3}{2}(t - 1, 0, 0)^T$, $t \in [0, 1]$. For $t = 0$ and $\mathbf{x} \in B(c_+(0); 1)$, one has $\|\mathbf{x} - c_+(0)\|^{-3} = 1$ and $\|\mathbf{x} - c_-(0)\|^{-3} \ll 1$. For $t = 0$ and $\mathbf{x} \in B(c_-(0); 1)$, one has $\|\mathbf{x} - c_+(0)\|^{-3} \ll 1$ and $\|\mathbf{x} - c_-(0)\|^{-3} = 1$. Hence, the initial configuration $\Gamma(0)$ is close to two balls of radius 1, centered at $\pm(1.5, 0, 0)^T$. For $t = 1$, the surface $\Gamma(1)$ is the ball around 0 with radius $2^{1/3}$. For $t > 0$, the two spheres approach each other until time $\tilde{t} = 1 - \frac{2}{3}2^{1/3} \approx 0.160$, when they touch at the origin. For $t \in (\tilde{t}, 1]$, the surface $\Gamma(t)$ is simply connected and deforms into the sphere $\Gamma(1)$.

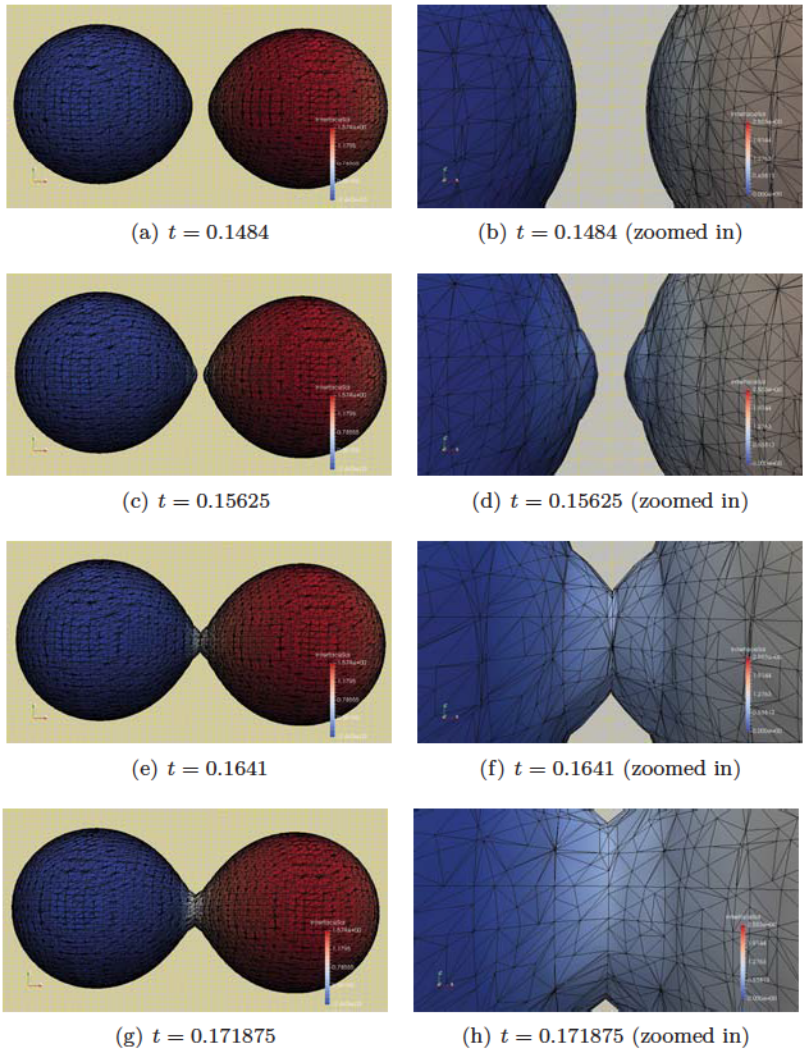
In the vicinity of $\Gamma(t)$, the gradient $\nabla\phi$ and the time derivative $\partial_t\phi$ are well defined and given by simple algebraic expressions. The normal velocity field, which transports $\Gamma(t)$, can be computed (cf. [27]) to be

$$\mathbf{w} = -\frac{\partial_t\phi}{|\nabla\phi|^2}\nabla\phi.$$

The initial value of u is given by

$$u_0(\mathbf{x}) = \begin{cases} 3 - x_1 & \text{for } x_1 \geq 0; \\ 0 & \text{otherwise.} \end{cases}$$

In Figure 5, we show a few snapshots of the surface and the computed surface solution on for the background tetrahedral mesh with $h = 1/16$ and $\Delta t = 1/128$. The



(a) $t = 0.1484$ (b) $t = 0.1484$ (zoomed in)

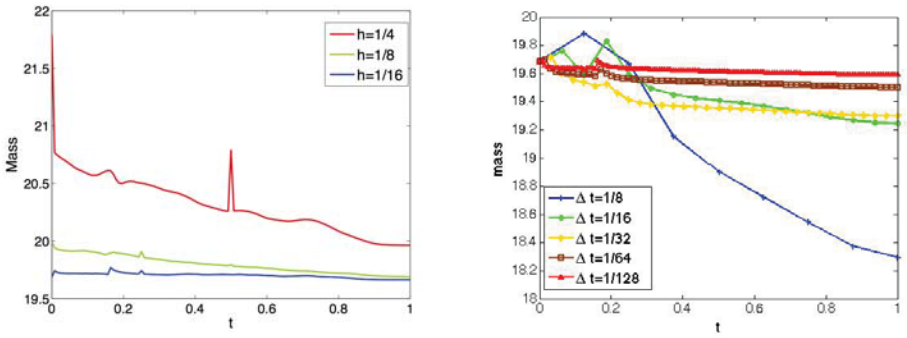
(c) $t = 0.15625$ (d) $t = 0.15625$ (zoomed in)

(e) $t = 0.1641$ (f) $t = 0.1641$ (zoomed in)

(g) $t = 0.171875$ (h) $t = 0.171875$ (zoomed in)

FIG. 6. *The computed solution and surface meshes close to the time of collision Example* 5.

surfaces Γ_h^n close to the time of collision are illustrated in Figure 6. The suggested variant of the trace FEM handles the geometrical singularity without any difficulty. It is clear that the computed extension u^e in this experiment is not smooth in a neighborhood of the singularity, and so formal analysis of the consistency of the method is not directly applicable to this case. However, the closest point extension is well defined, and numerical results suggest that this is sufficient for the method to be stable. Similar to the previous example, we compute the total discrete mass $M_h(t)$ on Γ_h^n. This can be used as a measure of accuracy. The evolution of $M_h(t)$ for varying h and Δt is shown in Figure 7. The convergence of the quantity is obvious. Finally, we note that in this experiment, we observed the stable numerical behavior of the method for any combinations of the mesh size and time step we tested, including $\Delta t = \frac{1}{8}$, $h = \frac{1}{16}$, and $\Delta t = \frac{1}{128}$, $h = \frac{1}{4}$.



(a) (b)

FIG. 7. *Total mass evolution for the finite element solution in Experiment* 5.

5. Conclusions. We studied a new fully Eulerian unfitted finite element method for solving PDEs posed on evolving surfaces. The method combines three computational techniques in a modular way: a finite difference approximation in time, a finite element method on a stationary surface, and an extension of finite element functions from a surface to a neighborhood of the surface. All three computational techniques have been intensively studied in the literature, and so well-established methods can be used. In this paper, we used the trace piecewise linear finite element method—the higher-order variants of this method are also available in the literature [26]—for the spatial discretization and a variant of the FMM to construct suitable extension. We observed stable and second-order-accurate numerical results in a several experiments, including one experiment with two colliding spheres. The rigorous error analysis of the method is a subject of further research.

REFERENCES

[1] D. ADALSTEINSSON AND J. A. SETHIAN, *Transport and diffusion of material quantities on propagating interfaces via level set methods*, J. Comput. Phys., 185 (2003), pp. 271–288.

[2] A. ALPHONSE, C. M. ELLIOTT, AND B. STINNER, *On some linear parabolic PDEs on moving hypersurfaces*, Interfaces Free Bound., 17 (2015), pp. 157–187.

[3] J. W. BARRETT, H. GARCKE, AND R. NÜRNBERG, *On the stable numerical approximation of two-phase flow with insoluble surfactant*, ESAIM: Math. Model. Numer. Anal., 49 (2015), pp. 421–458.

[4] M. BERTALMIO, L.-T. CHENG, S. OSHER, AND G. SAPIRO, *Variational problems and partial differential equations on implicit surfaces*, J. Comput. Phys., 174 (2001), pp. 759–780.

[5] T. BRETSCHNEIDER, C.-J. DU, C. M. ELLIOTT, T. RANNER, AND B. STINNER, *Solving Reaction-Diffusion Equations on Evolving Surfaces Defined by Biological Image Data*, http://arxiv.org/abs/1606.05093, 2016.

[6] M. BURGER, *Finite element approximation of elliptic partial differential equations on implicit surfaces*, Comput. Vis. Sci., 12 (2009), pp. 87–100.

[7] E. BURMAN, P. HANSBO, AND M. G. LARSON, *A stabilized cut finite element method for partial differential equations on surfaces: The Laplace–Beltrami operator*, Comput. Methods Appl. Mech. Engrg., 285 (2015), pp. 188–207.

[8] E. BURMAN, P. HANSBO, M. G. LARSON, AND A. MASSING, *A cut discontinuous Galerkin method for the Laplace–Beltrami operator*, IMA J. Numer. Anal., 37 (2017), pp. 138–169.

[9] E. BURMAN, P. HANSBO, M. G. LARSON, A. MASSING, AND S. ZAHEDI, *Full gradient stabilized cut finite element methods for surface partial differential equations*, Comput. Methods Appl. Mech. Engrg., 310 (2016), pp. 278–296.

[10] J. W. Cahn, P. Fife, and O. Penrose, *A phase field model for diffusion induced grain boundary motion*, Acta Mat., 45 (1997), pp. 4397–4413.
[11] M. Cenanovic, P. Hansbo, and M. G. Larson, *Cut finite element modeling of linear membranes*, Comput. Methods Appl. Mech. Engrg., 310 (2016), pp. 98–111.
[12] M. A. Chaplain, M. Ganesh, and I. G. Graham, *Spatio-temporal pattern formation on spherical surfaces: Numerical simulation and application to solid tumour growth*, J. Math. Biol., 42 (2001), pp. 387–423.
[13] A. Y. Chernyshenko and M. A. Olshanskii, *An adaptive octree finite element method for PDEs posed on surfaces*, Comput. Methods Appl. Mech. Engrg., 291 (2015), pp. 146–172.
[14] K. Deckelnick, G. Dziuk, C. M. Elliott, and C.-J. Heine, *An h-narrow band finite-element method for elliptic equations on implicit surfaces*, IMA J. Numer. Anal., (2009), p. drn049.
[15] K. Deckelnick, C. M. Elliott, and T. Ranner, *Unfitted finite element methods using bulk meshes for surface partial differential equations*, SIAM J. Numer. Anal., 52 (2014), pp. 2137–2162.
[16] A. Demlow and M. A. Olshanskii, *An adaptive surface finite element method based on volume meshes*, SIAM J. Numer. Anal., 50 (2012), pp. 1624–1647.
[17] *DROPS Package*, http://www.igpm.rwth-aachen.de/DROPS/.
[18] G. Dziuk, *Finite Elements for the Beltrami Operator on Arbitrary Surfaces*, in Partial Differential Equations and Calculus of Variations, Lecture Notes in Mathematics 1357, S. Hildebrandt and R. Leis, eds., Springer, Berlin, 1988, pp. 142–155.
[19] G. Dziuk and C. M. Elliott, *Finite elements on evolving surfaces*, IMA J. Numer. Anal., 27 (2007), pp. 262–292.
[20] G. Dziuk and C. M. Elliott, *Finite element methods for surface PDEs*, Acta Numer., 22 (2013), pp. 289–396.
[21] G. Dziuk and C. M. Elliott, *L^2-estimates for the evolving surface finite element method*, Math. Comp., 82 (2013), pp. 1–24.
[22] C. M. Elliott and T. Ranner, *Evolving surface finite element method for the Cahn–Hilliard equation*, Numer. Math., 129 (2015), pp. 483–534.
[23] C. M. Elliott and B. Stinner, *Modeling and computation of two phase geometric biomembranes using surface finite elements*, J. Comput. Phys., 226 (2007), pp. 1271–1290.
[24] C. M. Elliott and C. Venkataraman, *Error analysis for an ALE evolving surface finite element method*, Numer. Methods Partial Differential Equations, 31 (2015), pp. 459–499.
[25] J. Grande, *Eulerian finite element methods for parabolic equations on moving surfaces*, SIAM J. Sci. Comput., 36 (2014), pp. B248–B271.
[26] J. Grande, C. Lehrenfeld, and A. Reusken, *Analysis of a High Order Trace Finite Element Method for PDEs on Level Set Surfaces*, http://arxiv.org/abs/1611.01100, 2016.
[27] J. Grande, M. A. Olshanskii, and A. Reusken, *A space-time FEM for PDEs on evolving surfaces*, in Proceedings of 11th World Congress on Computational Mechanics, E. Onate, J. Oliver, and A. Huerta, eds., Eccomas, IGPM report 386, RWTH, Aachen, 2014.
[28] J. Grande and A. Reusken, *A higher order finite element method for partial differential equations on surfaces*, SIAM J. Numer. Anal., 54 (2016), pp. 388–414.
[29] S. Gross, M. A. Olshanskii, and A. Reusken, *A trace finite element method for a class of coupled bulk-interface transport problems*, ESAIM: Math. Model. Numer. Anal., 49 (2015), pp. 1303–1330.
[30] S. Gross and A. Reusken, *Numerical Methods for Two-Phase Incompressible Flows*, Springer, Berlin, 2011.
[31] P. Hansbo, M. G. Larson, and S. Zahedi, *Characteristic cut finite element methods for convection–diffusion problems on time dependent surfaces*, Comput. Methods Appl. Mech. Engrg., 293 (2015), pp. 431–461.
[32] P. Hansbo, M. G. Larson, and S. Zahedi, *A cut finite element method for coupled bulk–surface problems on time–dependent domains*, Comput. Methods Appl. Mech. Engrg., 307 (2016), pp. 96–116.
[33] A. James and J. Lowengrub, *A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant*, J. Comput. Phys., 201 (2004), pp. 685–722.
[34] C. Lehrenfeld, *High order unfitted finite element methods on level set domains using isoparametric mappings*, Comput. Methods Appl. Mech. Engrg., 300 (2016), pp. 716–733.
[35] G. MacDonald, J. Mackenzie, M. Nolan, and R. Insall, *A computational method for the coupled solution of reaction–diffusion equations on evolving domains and manifolds: Application to a model of cell migration and chemotaxis*, J. Comput. Phys., 309 (2016), pp. 207–226.
[36] U. F. Mayer and G. Simonnett, *Classical solutions for diffusion induced grain boundary motion*, J. Math. Anal., 234 (1999), pp. 660–674.

[37] W. Milliken, H. Stone, and L. Leal, *The effect of surfactant on transient motion of Newtonian drops*, Phys. Fluids A, 5 (1993), pp. 69–79.
[38] I. L. Novak, F. Gao, Y.-S. Choi, D. Resasco, J. C. Schaff, and B. Slepchenko, *Diffusion on a curved surface coupled to diffusion in the volume: Application to cell biology*, J. Comput. Phys., 229 (2010), pp. 6585–6612.
[39] M. A. Olshanskii and A. Reusken, *A finite element method for surface PDEs: Matrix properties*, Numer. Math., 114 (2010), pp. 491–520.
[40] M. A. Olshanskii and A. Reusken, *Error analysis of a space–time finite element method for solving PDEs on evolving surfaces*, SIAM J. Numer. Anal., 52 (2014), pp. 2092–2120.
[41] M. A. Olshanskii, A. Reusken, and J. Grande, *A finite element method for elliptic equations on surfaces*, SIAM J. Numer. Anal., 47 (2009), pp. 3339–3358.
[42] M. A. Olshanskii, A. Reusken, and X. Xu, *On surface meshes induced by level set functions*, Comput. Vis. Sci., 15 (2012), pp. 53–60.
[43] M. A. Olshanskii, A. Reusken, and X. Xu, *An Eulerian space–time finite element method for diffusion problems on evolving surfaces*, SIAM J. Numer. Anal., 52 (2014), pp. 1354–1377.
[44] M. A. Olshanskii, A. Reusken, and X. Xu, *A stabilized finite element method for advection-diffusion equations on surfaces*, IMA J. Numer. Anal., 34 (2014), pp. 732–758.
[45] A. Petras and S. Ruuth, *PDEs on moving surfaces via the closest point method and a modified grid based particle method*, J. Comput. Phys., 312 (2016), pp. 139–156.
[46] A. Reusken, *Analysis of trace finite element methods for surface partial differential equations*, IMA J. Numer. Anal., 35 (2015), pp. 1568–1590.
[47] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA, 93 (1996), pp. 1591–1595.
[48] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, 1999.
[49] A. Sokolov, R. Ali, and S. Turek, *An AFC-stabilized implicit finite element method for partial differential equations on evolving-in-time surfaces*, J. Comput. Appl. Math., 289 (2015), pp. 101–115.
[50] H. Stone, *A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface*, Phys. Fluids A, 2 (1990), pp. 111–112.
[51] J.-J. Xu, Z. Li, J. Lowengrub, and H. Zhao, *A level-set method for interfacial flows with surfactant*, J. Comput. Phys., 212 (2006), pp. 590–616.
[52] J.-J. Xu and H.-K. Zhao, *An Eulerian formulation for solving partial differential equations along a moving interface*, J. Sci. Comput., 19 (2003), pp. 573–594.