

Preprint Series

A high order FEM with exact geometry description for the Laplacian on implicitly defined surfaces

Michael Gfrerer^{1,2}, Martin Schanz¹

¹*Institute of Applied Mechanics, Graz University of Technology*

²*Felix Klein Center for Mathematics, University of Kaiserslautern*

Published in *International Journal for Numerical Methods in Engineering*, 114, 1163–1178, 2018

DOI: 10.1002/nme.5779

Latest revision: 17.1.2018

Abstract

In this paper, a high order finite element method for partial differential equations on smooth surfaces is proposed. The surface is defined as the intersection of a rectangular cuboid and an implicitly defined surface. Therefore, the surface of interest may not be closed. The main novel contribution in this work is the incorporation of an exact geometry description of surfaces with boundary into the finite element method. To this end, a piecewise planar triangulation is mapped onto the surface of interest by making use of the implicit surface definition. The mapping uses predefined search directions and can, therefore, be tailored to consider boundaries. High order hierarchical shape functions are utilized for the field approximation. They are defined on a reference triangle in the usual way. The proposed method is easy to implement and bypasses the need of a high order geometry description. Furthermore, due to the exact geometry the imposition of Dirichlet boundary conditions, source terms and mesh refinement are easy to carry out.

1 Introduction

The numerical solution of partial differential equations on surfaces by means of the finite element method has been attracting intensive research since the emerging of finite element analysis. In the beginning, the main motivation was given by structural mechanics problems, see *e.g.* [1]. Today, besides membrane [2] and shell problems [3], surface problems are of interest in many branches in science and engineering, *e.g.* fluid flow on surfaces [4, 5], in the context of two-phase flow [6], and geometric flow problems [7]. We refer to [8] and references therein for applications in image processing, computer graphics, and pattern formation.

In the present paper, we focus on methods for problems on implicitly defined surfaces. We refer to [9] for an overview of different methods. The first surface finite element method for such problems was proposed in [10]. A piecewise linear discretization on a triangulation was used to solve the Laplace-Beltrami equation. This resulted in a second order accurate method. In order to compare functions on the triangulation and on the exact surface, the lift based on the closest point projection was introduced for the theoretical analysis. In [11], this lift was utilized in order to obtain an adaptive method. The extension towards an arbitrary order method was presented in [12] for closed smooth surfaces. In coupled bulk-surface problems, like transport and diffusion of surfactants in two phase flows and cellular dynamics in cell motility, only a finite element approximation of the level-set function defining the interface surface might be available. This setting was treated for closed stationary surfaces in [13]. To this end, the closest point projection was adapted making use of a continuous quasi-normal field instead of the possibly rough exact normals. The resulting method is of higher order if the original surface is smooth. In [14], a high order method for surfaces with boundaries defined by multiple level set functions was proposed. This method relies on the high order accurate surface approximation, which is constructed using a background mesh and directional mapping of Lagrange elements. The publications so far are concerned with stationary surfaces. In contrast to this, surface finite elements for evolving surfaces are treated in [15, 16, 17, 18]. The coupling of a surface and a bulk problem was investigated in [19].

Another class of methods for surface problems are the Eulerian methods [8]. Such methods rely on the extension of the problem to the bulk volume. Thus, the problem to solve is posed in a domain with one dimension higher than the original problem. This approach circumvents the explicit meshing of the surface. In [8, 20, 21] finite difference methods are proposed for the solution. Eulerian finite element methods are first described in [22] and [23]. In [24], the bulk problem was restricted to a narrow band.

A third class of methods for surface problems is labeled TraceFEM [25, 26] or CutFEM [27, 28]. The main idea of this method is to use the finite element spaces defined on a volume triangulation. However, the surface problem is not extended to a bulk problem. Instead, the trace of the bulk discretization on the (approximated) surface is used. Due to this the shape functions on the surface may not be linearly independent. Thus, stabilization techniques are necessary, cf. [26]. An adaptive method has been published in [29], whereas the extension to evolving surfaces is given in [30].

Besides the solution of surface problems, the integration over implicitly defined surfaces and volumes is of interest in the context of embedded/fictitious domain methods like the Finite Cell Method [31, 32, 33], CutFEM [34], immersed boundary methods [35], and immersed finite ele-

ment methods [36, 37]. In such methods, the boundary of the problem domain is not explicitly meshed. Therefore, the main difficulty is the integration of the "cut elements" and the imposition of boundary conditions. To face the integration issue, several approaches with optimal computational complexity has been proposed (see [38] for a comparison). One technique relies on a high order re-meshing, which was developed in a series of papers [39, 40, 41]. The integration of the cut elements and its boundary has been done by means of moment fitting in [42]. Furthermore, in [43] high order integration based on height functions over hyper-planes is developed. This last approach has delivered input to our developments.

In the present work, we propose a surface finite element method for smooth stationary surfaces defined as the intersection of the zero level of an exact level-set function and a rectangular cuboid. In contrast to high order surface finite element methods relying on high order accurate geometry approximations, a feature of our method is the parametrization of the exact surface over a coarse base triangulation. An exact geometry method making use of the closest point projection was proposed in [12]. Following this idea, hybridizable discontinuous Galerkin methods are defined and analyzed in [44]. Unfortunately, the mapping is only known explicitly if the signed distance function is known explicitly. This is the case only for special shapes. In many situations, only a level set function is known. Then the signed distance function has to be approximated numerically, which is a non-trivial task, cf. [45]. Furthermore, the mapping is in general only exact for closed surfaces. In the case of surfaces with boundaries a geometric error arises. Following the idea introduced in [46] and applied to a TraceFEM for surface problems in [47], we make use of a directional mapping. Therefore, we are able to map every point on the triangulated surface with a specific search direction. However, in the cited publications the search directions coincide with (some approximation of) the gradient of the level set function. Releasing this condition allows us to tailor the search directions such that surfaces with boundaries can be treated exactly. Moreover, we avoid the isoparametric approximation of the geometry mapping. Nevertheless, the implementation of the proposed method remains simple. Due to the exact geometry mapping, our method has a number of interesting features. The imposition of Dirichlet boundary conditions can be done in a strong form without extensions or corrections. Furthermore, the exact source terms defined on the exact surface can be used without extension. Moreover, mesh refinement can be performed by splitting of the existing elements without the need of adapting the new triangulation to the exact surface. Nevertheless, as usual in the finite element method, we require numerical quadrature for the integration over the surface. Due to the exact geometry standard quadrature rules can be applied, yielding a super-algebraic convergent surface integration. Hence, this approach can be used for the numerical integration in other types of methods.

The remainder of the paper is organized in the following way. In Section 2, we introduce the problem setting and present the variational formulation in the case of a parametrized surface. Section 3 deals with the basic ingredients of the finite element method. In Section 4, we present the exact parametrization of the implicitly defined surface. Numerical examples will be presented in Section 5. Finally, the paper will be concluded in Section 6.

2 Preliminaries

Let $B \subseteq \mathbb{R}^3$ be a rectangular cuboid. Let the smooth surface Ω be defined as the zero level-set of a given function $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$

$$\Omega = \{\mathbf{x} \in B \mid \phi(\mathbf{x}) = 0\}. \quad (1)$$

We are interested in solving the following boundary value problem on the surface $\Omega \subset B$ with boundary σ : Find the function u defined on Ω such that

$$\begin{aligned} u - \Delta_S u &= g_\Omega, \\ u(\mathbf{x}) &= g_D(\mathbf{x}) \quad \text{for } \mathbf{x} \in \sigma_D, \\ \nabla_S u(\mathbf{x}) \cdot \mathbf{n}_S &= g_N(\mathbf{x}) \quad \text{for } \mathbf{x} \in \sigma_N. \end{aligned} \quad (2)$$

Here, Δ_S is the Laplace-Beltrami operator on Ω and ∇_S the tangential gradient. The outer unit normal vector \mathbf{n}_S of σ lies in the tangent space of Ω , whereas the surface normal vector \mathbf{n} is orthogonal on the tangent plane. The geometric setting of the problem is illustrated in Figure 1. The given function g_Ω is the source term, g_D is the given Dirichlet datum on the Dirichlet boundary σ_D and g_N is the given Neumann datum on the Neumann boundary σ_N . Furthermore, we require $\sigma_D \cap \sigma_N = \{\emptyset\}$ and $\sigma_D \cup \sigma_N = \sigma$.

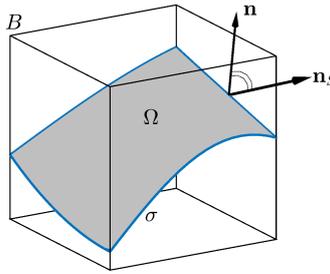


Figure 1: Problem setting

We recall the basic relations in the case of a parametrized surface. Let

$$\begin{aligned} \mathbf{g} : P &\rightarrow \Omega \\ (\theta^1, \theta^2) &\mapsto \mathbf{x} \end{aligned} \quad (3)$$

be a regular parametrization of Ω . Then the tangent plane to the surface is spanned by the two linear independent base vectors

$$\mathbf{g}_\alpha = \mathbf{g}_{,\alpha} := \frac{\partial \mathbf{g}}{\partial \theta^\alpha}. \quad (4)$$

Here, and in the following the Greek letters α and β take the values 1 and 2. The coefficients of the first fundamental form are given by the scalar products of the base vectors

$$g_{\alpha\beta} = \mathbf{g}_\alpha \cdot \mathbf{g}_\beta. \quad (5)$$

The dual basis is obtained by

$$\mathbf{g}^\alpha = g^{\alpha\beta} \mathbf{g}_\beta, \quad (6)$$

where $[g^{\alpha\beta}]$ is the inverse of the matrix $[g_{\alpha\beta}]$. Here, and in the following, Einstein summation convention applies. Whenever an index occurs once in an upper position and in a lower position we sum over this index. With the definitions above, we are able to define the surface area as integrals over Ω and P as

$$A = \int_{\Omega} d\mathbf{x} = \int_P \sqrt{\det([g_{\alpha\beta}])} d\Theta, \quad (7)$$

where $d\mathbf{x}$ and $d\Theta$ are the surface measures on Ω and P , respectively. Furthermore, the surface Laplacian $\Delta_S u$ on the surface Ω is defined as

$$\Delta_S u = \nabla_S \cdot \nabla_S u. \quad (8)$$

Defining $\hat{u}(\theta^1, \theta^2) = u(\mathbf{g}(\theta^1, \theta^2))$, the surface gradient is given in local coordinates by [48]

$$\nabla_S u = \hat{u}_{,\alpha} \mathbf{g}^\alpha. \quad (9)$$

Next, we state the weak form of problem (2). To this end, we define $V_g = \{u \in H^1(\Omega) \mid u = g_D \text{ on } \sigma_D\}$ and $V_0 = \{v \in H^1(\Omega) \mid v = 0 \text{ on } \sigma_D\}$. Find $u \in V_g$ such that

$$\int_{\Omega} u v d\mathbf{x} + \int_{\Omega} \nabla_S u \cdot \nabla_S v d\mathbf{x} = \int_{\Omega} g_{\Omega} v d\mathbf{x} + \int_{\sigma_N} g_N v ds_{\mathbf{x}}, \quad \forall v \in V_0. \quad (10)$$

Here, $ds_{\mathbf{x}}$ is the line measure on σ . In local coordinates, the integrals can be evaluated according to

$$\begin{aligned} \int_{\Omega} u v d\mathbf{x} &= \int_P \hat{u} \hat{v} \sqrt{\det([g_{\alpha\beta}])} d\Theta, \\ \int_{\Omega} \nabla_S u \cdot \nabla_S v d\mathbf{x} &= \int_P \hat{u}_{,\alpha} \hat{v}_{,\beta} g^{\alpha\beta} \sqrt{\det([g_{\alpha\beta}])} d\Theta, \\ \int_{\Omega} g_{\Omega} v d\mathbf{x} &= \int_P g_{\Omega}(\mathbf{g}(\theta^1, \theta^2)) \hat{v} \sqrt{\det([g_{\alpha\beta}])} d\Theta, \\ \int_{\sigma_N} g_N v ds_{\mathbf{x}} &= \int_{\mathbf{g}^{-1}(\sigma_N)} g_N(\mathbf{g}(\theta^1, \theta^2)) \hat{v} \sqrt{g_{\theta^1} \cdot g_{\theta^1}} d\theta. \end{aligned} \quad (11)$$

Here, g_{θ} is the tangent vector to the boundary curve σ_N and $d\theta$ the respective line measure. In the following, we construct an exact element-wise parametrization of Ω and make the first order derivatives available. Thus, the point-wise evaluation of the integrands in (11) is possible. Within the FEM, we evaluate the integrals numerically by means of a quadrature rule. Thus, a quadrature error is introduced.

3 Finite Element Method

In this section, we briefly recall the basic principles of the FEM. Here, the first step to obtain a finite element discretization is a geometric partitioning of the problem domain Ω with a mesh \mathcal{T} . The mesh is the union of geometric elements τ_e . We denote the union of all τ_e by Ω_h . In

the present paper, we consider triangular τ_e only. In many situations \mathcal{T} is only an approximation of Ω . This is particularly true in the case of curved surfaces Ω and flat triangles τ_e . In order to eliminate the geometric error we will introduce the mapping $a : \Omega_h \rightarrow \Omega$. This mapping will be examined in Section 4.

For the construction of shape functions, the reference element $\tau^R = \{(\xi, \eta) \in [0, 1]^2 \mid \eta + \xi \leq 1\}$ is introduced. We define the mapping Φ_e such that each geometric element τ_e is the image of

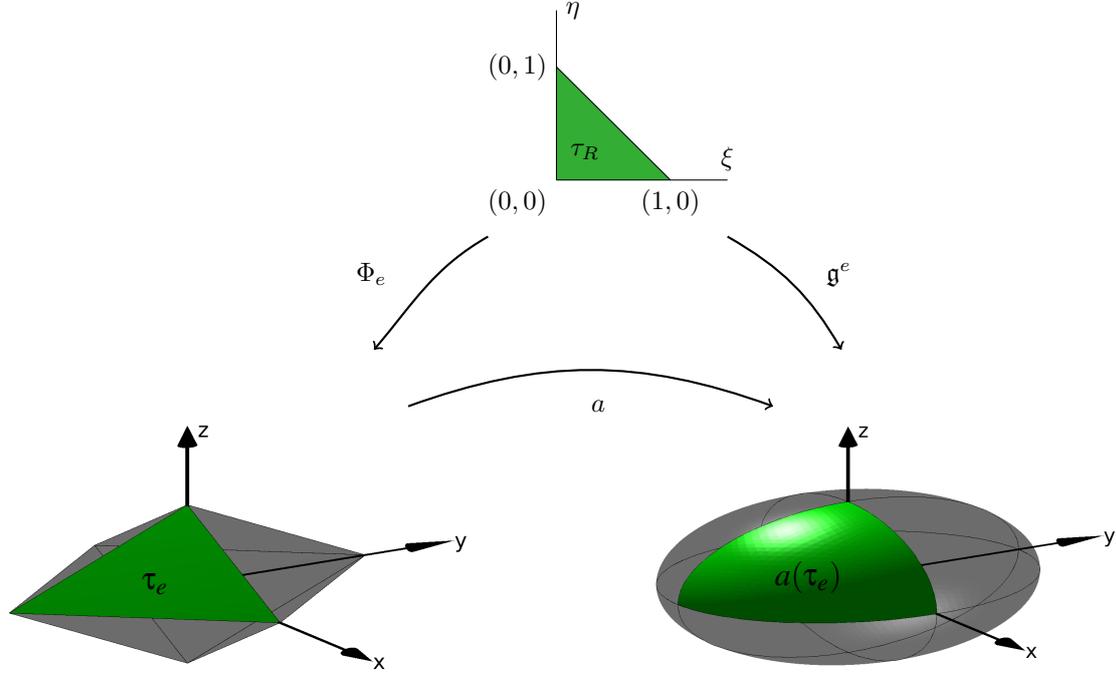


Figure 2: Mappings Φ_e , a , and $\mathfrak{g}_e = a \circ \Phi_e$ between the reference triangle τ^R , the element $\tau_e \in \mathcal{T}$, and $a(\tau_e)$

the reference element

$$\begin{aligned} \Phi_e : \tau^R &\rightarrow \tau_e \\ (\xi, \eta) &\mapsto \mathbf{x} = \sum_{i=1}^3 \lambda_i(\xi, \eta) \mathbf{x}_i^e, \end{aligned} \quad (12)$$

with

$$\begin{aligned} \lambda_1 &= 1 - \xi - \eta, \\ \lambda_2 &= \xi, \\ \lambda_3 &= \eta, \end{aligned} \quad (13)$$

and \mathbf{x}_i^e are coordinates of the vertexes of τ_e . The different mappings are illustrated in Figure 2. In order to discretize a function, we use hierarchical H^1 conforming shape functions defined on the reference triangle. We follow the developments and notations in [49]. The Legendre polynomials are defined for $x \in [-1, 1]$ by the three-term recurrence relation

$$(n+1)\ell_{n+1}(x) = (2n+1)\ell_n(x) - n\ell_{n-1}(x) \quad \text{for } n \geq 1, \quad (14)$$

with

$$\begin{aligned}\ell_0(x) &= 1, \\ \ell_1(x) &= x.\end{aligned}\tag{15}$$

Furthermore, the integrated Legendre polynomials are

$$L_n(x) = \int_{-1}^x \ell_{n-1}(y) \, dy.\tag{16}$$

The scaled integrated Legendre polynomials are defined as

$$L_n^S(x, t) = t^n L_n\left(\frac{x}{t}\right) \quad \text{for } x \in [-t, t], \, t \in (0, 1].\tag{17}$$

Then the shape functions of the Vertex-Edge-Cell base space of polynomial order p can be given. The vertex-based shape functions $1 \leq i \leq 3$ are

$$\phi_i^V = \lambda_i.\tag{18}$$

The edge-based shape functions $2 \leq i+1 \leq p$ for the oriented edges E_m from vertices e_1 to e_2 are

$$\phi_i^{E_m} = L_{i+2}^S(\lambda_{e_2} - \lambda_{e_1}, \lambda_{e_1} + \lambda_{e_2}).\tag{19}$$

The cell-based shape functions $1 \leq i+j \leq p-3$ are

$$\phi_{ij}^C = L_{i+2}^S(\lambda_2 - \lambda_1, \lambda_1 + \lambda_2) \lambda_3 \ell_j(2\lambda_3 - 1).\tag{20}$$

These element shape functions are pieced together to FEM basis functions in the usual way by establishing a connection between local and global degrees of freedom. In the present paper, we restrict us to a uniform polynomial order p . The space spanned by the finite element functions is denoted by V_h .

Due to the exact geometry, we are able to impose Dirichlet boundary conditions in a strong form without extension of the given data (see *e.g.* [50] for considerations using an extension). Unlike isoparametric elements, we do not have the concept of a nodal basis here. Therefore, we impose Dirichlet boundary conditions by a L_2 projection. To this end, we first solve for $u_{h,\sigma} \in V_{h,\sigma} = \{u_h \in V_h \mid u_h(\mathbf{x}) \neq 0 \text{ for } \mathbf{x} \in \sigma_D\}$ such that

$$\int_{\sigma_D} u_{h,\sigma} v_\sigma \, ds_{\mathbf{x}} = \int_{\sigma_D} g_D v_\sigma \, ds_{\mathbf{x}}\tag{21}$$

holds for all $v_\sigma \in V_{h,\sigma}$. Hence, the discrete problem reads: Find $u_h \in V_{h,g} = \{w_h \in V_h \mid w_h(x) = u_{h,\sigma}(x) \text{ for } x \in \sigma_D\}$ such that

$$\int_{\Omega} u_h v_h \, d\mathbf{x} + \int_{\Omega} \nabla_S u_h \cdot \nabla_S v_h \, d\mathbf{x} = \int_{\Omega} g_\Omega v_h \, d\mathbf{x} + \int_{\sigma_N} g_N v_h \, ds_{\mathbf{x}},\tag{22}$$

holds for all $v_h \in V_{h,0} = \{w_h \in V_h \mid w_h(x) = 0 \text{ for } x \in \sigma_D\}$.

4 Exact parametrization of the implicit surface

In this section, we extend the standard geometry mapping (12) such that the exact geometry is preserved, *i.e.* we map the discrete surface Ω_h to the exact surface Ω .

We briefly review the mapping based on the closest point projection. In order to compare a function v_h defined on Ω_h to a function defined on Ω , we introduce the lifted function v_h^l as in [10]

$$v_h^l(\mathbf{x} - d(\mathbf{x})\mathbf{n}(\mathbf{x})) = v_h(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega_h. \quad (23)$$

Here, d is the signed distance function. It holds $|d(\mathbf{x})| = \text{dist}(\mathbf{x}, \Omega)$. This is valid in a strip $U = \{\mathbf{x} \in \mathbb{R}^3 \mid \text{dist}(\mathbf{x}, \Omega) < \delta\}$ about Ω , where δ is bounded by the curvature. In (23), the mapping

$$\begin{aligned} \tilde{a} : \Omega_h &\rightarrow \Omega \\ \mathbf{x} &\mapsto \tilde{a}(\mathbf{x}) = \mathbf{x} - d(\mathbf{x})\mathbf{n}(\mathbf{x}). \end{aligned} \quad (24)$$

is used. This mapping has been used in [12] to propose a method based on the exact geometry description for closed surfaces. However, the use of (24) has two drawbacks. Firstly, the numerical realization is difficult in cases where the signed distance function d is not explicitly known. Secondly, for surfaces with boundaries, the mapped surface does not have to agree with the exact surface. Thus, the exact geometry cannot be preserved in general. Such a situation is illustrated in Figure 3. The level-set function $\phi(\mathbf{x}) = \|\mathbf{x}\| - 1$ defines a sphere in \mathbb{R}^3 . However,

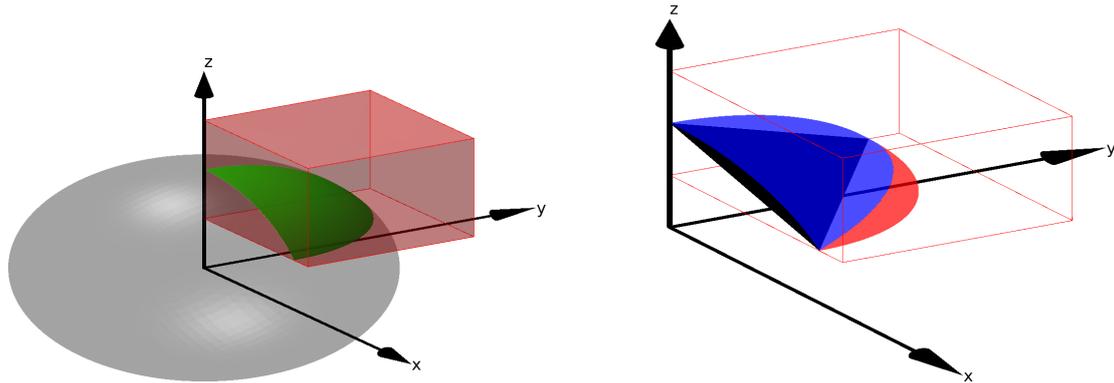


Figure 3: Limitations of the mapping \tilde{a}

only that part lying inside $B = [0, 1] \times [0, 1] \times [0.5, 1.5]$ is the considered surface (see the left representation of Figure 3). Let \mathcal{T} consist of the single black triangle, which is displayed in the right representation. The vertices are $(\sqrt{3}/4, 0, 1/2)$, $(0, \sqrt{3}/4, 1/2)$, and $(0, 0, 1)$. Thus, they are on the exact surface. The application of the mapping (24) to the triangle yields the blue surface $\tilde{a}(\Omega_h)$, which is depicted in the right of Figure 3). The red surface is the difference between the mapped triangle $\tilde{a}(\Omega_h)$ and the exact surface Ω . Thus, in the given example $\tilde{a}(\Omega_h) \neq \Omega$.

To overcome such issues arising with the mapping based on the closest point projection we use the mapping

$$\begin{aligned} a : \Omega_h &\rightarrow \Omega \\ x &\mapsto a(\mathbf{x}) = \mathbf{x} + r(\mathbf{x})\mathbf{s}(\mathbf{x}). \end{aligned} \quad (25)$$

Here, $\mathbf{s}(\mathbf{x})$ are predefined search directions and $r(\mathbf{x})$ is the distance of \mathbf{x} to the surface with respect to the respective search direction. Such a mapping has been considered in [46] and in [14]. However, [46] is concerned with interface problems where the level-set function describing the interface is given as a finite element function. There, the bulk problem is discretized by an unfitted background mesh. In order to increase the resolution of the interface, the mesh is transformed. In [14], surfaces described by multiple exact level-set functions are considered. However, only a high order accurate approximation of Ω is constructed making use of surface elements of Lagrange type. The elements and the search directions are induced by a background mesh.

In the present approach, no background mesh is necessary. The requirement is that a valid base triangulation is available, *i.e.* (25) is a regular map. Nevertheless, we remark that a background mesh can be a useful tool for the construction of a valid initial mesh. Furthermore, a necessary requirement for the present approach is that the vertices on the boundary of \mathcal{T} have to lie on the boundary of the bounding box B . We specify the search directions in (25) as follows. Let \mathcal{V} denote the set of all vertices of \mathcal{T} . We set

$$\tilde{\mathbf{s}}_v(\mathbf{x}) = \frac{\nabla\phi(\mathbf{x})}{\|\nabla\phi(\mathbf{x})\|} \quad \text{for } \mathbf{x} \in \mathcal{V}, \quad (26)$$

where $\nabla\phi$ is the usual gradient of ϕ in \mathbb{R}^3 . To preserve the exact geometry, we apply a modification at the vertices on the boundary of B . Thus, we set

$$\mathbf{s}_v(\mathbf{x}) = \begin{cases} \tilde{\mathbf{s}}_v(\mathbf{x}) - (\tilde{\mathbf{s}}_v(\mathbf{x}) \cdot \mathbf{n}_{\partial B}(\mathbf{x})) \mathbf{n}_{\partial B}(\mathbf{x}) & \text{for } \mathbf{x} \in \mathcal{V} \cap \partial B \\ \tilde{\mathbf{s}}_v(\mathbf{x}) & \text{else} \end{cases}, \quad (27)$$

where $\mathbf{n}_{\partial B}$ are the normal vectors to ∂B . Then the search direction field $\mathbf{s}(\mathbf{x})$ defined on Ω_h is obtained by linear interpolation of $\mathbf{s}_v(\mathbf{x})$.

The mapping (25) requires the solution of a non-linear root finding problem. This is numerically realized by means of the Newton iteration

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{s}(\mathbf{x}^k) \frac{\phi(\mathbf{x}^k)}{\nabla\phi(\mathbf{x}^k) \cdot \mathbf{s}(\mathbf{x}^k)}. \quad (28)$$

Thus, every point in the reference element can be mapped onto the exact geometry. In our experience, the Newton scheme (28) converges within a few iterations (at most seven iterations in the numerical examples given in Section 5). We enhance the standard FEM element mapping by the mapping (25). Thus, we obtain an element-wise parametrization of Ω by

$$\begin{aligned} \mathbf{g}^e : \tau^R &\rightarrow \Omega \\ (\theta^1, \theta^2) &\mapsto a(\Phi_e(\theta^1, \theta^2)). \end{aligned} \quad (29)$$

The evaluation of the integrals in (11) requires $\mathbf{x} = \mathbf{g}^e(\theta^1, \theta^2)$ and the base vectors. All other geometric quantities can be computed from the base vectors. In the case of (24), they can be calculated from the first and second order derivatives of d and the first order derivatives of Φ^e , cf. [11], [12]. Here, we are able to compute the base vectors based on the knowledge of $\nabla\phi$

only, once we have solved $\mathbf{x} = \mathbf{g}^e(\theta^1, \theta^2)$, *i.e.* we know the distance r . We are interested in the base vectors given by

$$\mathbf{g}_\alpha = \mathbf{g}_{,\alpha}^e = \Phi_{,\alpha}^e + r_{,\alpha} \mathbf{s} + r \mathbf{s}_{,\alpha}. \quad (30)$$

Since Φ^e and \mathbf{s} are linear finite element functions, their derivatives are available in finite element codes. However, $r_{,\alpha}$ is not readily available. In order to compute $r_{,\alpha}$ we note that for a point $\mathbf{x} \in \Omega$ the surface normal vector is given by

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla\phi(\mathbf{x})}{\|\nabla\phi(\mathbf{x})\|}. \quad (31)$$

The scalar product of (30) with the normal vector yields

$$\Phi_{,\alpha}^e \cdot \mathbf{n} + r_{,\alpha} \mathbf{s} \cdot \mathbf{n} + r \mathbf{s}_{,\alpha} \cdot \mathbf{n} = 0. \quad (32)$$

Thus, we are able to obtain the derivatives of the distance r by

$$r_{,\alpha} = -\frac{\Phi_{,\alpha}^e \cdot \mathbf{n} + r \mathbf{s}_{,\alpha} \cdot \mathbf{n}}{\mathbf{s} \cdot \mathbf{n}}. \quad (33)$$

This result can be inserted in (30) to obtain the computable expression

$$\mathbf{g}_\alpha = \Phi_{,\alpha}^e - \frac{\Phi_{,\alpha}^e \cdot \mathbf{n} + r \mathbf{s}_{,\alpha} \cdot \mathbf{n}}{\mathbf{s} \cdot \mathbf{n}} \mathbf{s} + r \mathbf{s}_{,\alpha}. \quad (34)$$

With (34) the base vectors and, therefore, all other geometric quantities necessary for the implementation are available.

For the success of the method, it is necessary to have a valid base triangulation. For such a triangulation we have the requirement $\sqrt{\det([g_{\alpha\beta}])} > 0$. We remark that $\sqrt{\det([g_{\alpha\beta}])}$ is non-negative by construction and has to be evaluated at the quadrature points. Thus, finding a point where $\sqrt{\det([g_{\alpha\beta}])} = 0$ is difficult. However, here we have $\sqrt{\det([g_{\alpha\beta}])} = \|\tilde{\mathbf{n}}\|$ where $\tilde{\mathbf{n}} = \mathbf{g}_1 \times \mathbf{g}_2$. Thus, we are able to evaluate the requirement $\tilde{\mathbf{n}} \cdot \mathbf{n} > 0$ instead. Furthermore, from (34) we observe the requirement $\mathbf{s} \cdot \mathbf{n} > 0$. These two conditions can easily be checked at a number of points in each triangle. If any is violated the base triangulation has to be refined in that region.

We end this section with a comment on solution refinement. In order to estimate the error of the numerical solution, refinements are of interest. In standard FEM approaches, for solution refinement, one needs a procedure how to refine the mesh in order that it yields a better geometry approximation. In contrast to this, in the present method, it is only necessary to have one valid base triangulation \mathcal{T} . Here, a valid mesh is a mesh for which (29) provides a piecewise regular parametrization of the exact surface. Thus, for a valid mesh, the geometry error vanishes. Therefore, no adaption of the refined mesh obtained by element splitting is necessary for the present approach. This means that although the triangulation \mathcal{T} is refined the base surface Ω_h is fixed during refinement. In this setting, it is desirable to keep the mapping (25) fixed. To this end, we define the search directions on the coarsest triangulation and interpolate them on the finer levels. This is easily possible since we define the search directions as linear finite element functions.

5 Numerical results

In this section, we verify and investigate the performance of the present approach considering numerical examples. We verify the developed FEM making use of the method of manufactured solutions (MMS). To this end, we define the surface Laplacian by means of tangential calculus. The surface gradient of a scalar function $v(x)$ defined on \mathbb{R}^3 is defined by

$$\nabla_S v(\mathbf{x}) = \nabla v(\mathbf{x}) - (\nabla v(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})) \mathbf{n}(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega. \quad (35)$$

Let $(\nabla_S v)_i = \frac{\partial v}{\partial x_i} - (\nabla_S v \cdot \mathbf{n}) \mathbf{n}_i$ be a component of $\nabla_S v$. Then the surface divergence of a vector field \mathbf{w} with component w_i is defined as

$$\nabla_S \cdot \mathbf{w} = \sum_{i=1}^3 (\nabla_S w_i)_i. \quad (36)$$

Following the MMS concept, we prescribe a solution u^M defined on \mathbb{R}^3 . Then the source term is given with

$$g_\Omega = u^M - \nabla_S \cdot \nabla_S u^M. \quad (37)$$

The boundary data functions are

$$\begin{aligned} g_D(\mathbf{x}) &= u^M(\mathbf{x}) & \text{for } \mathbf{x} \in \sigma_D, \\ g_N(\mathbf{x}) &= \nabla_S u^M(\mathbf{x}) \cdot \mathbf{n}_S & \text{for } \mathbf{x} \in \sigma_N. \end{aligned} \quad (38)$$

In the following examples, we investigate the error introduced by the numerical quadrature and the field approximation. To this end, we define the relative error in area

$$e_A = \frac{|A - A_h|}{A} \quad (39)$$

and the relative error in the field approximation

$$e_u = \sqrt{\frac{\int_\Omega (u^M - u_h)^2 \, d\mathbf{x}}{\int_\Omega (u^M)^2 \, d\mathbf{x}}}. \quad (40)$$

5.1 Spherical surface

In this first example, we consider the spherical surface Ω given by

$$\phi(x, y, z) = 1 - x^2 - y^2 - z^2 \quad (41)$$

and $B = [-2, 2] \times [-2, 2] \times [-0.2, 0.8]$. We have constructed a base triangulation \mathcal{T}_0 , which is only loosely in relation to Ω . The \mathcal{T}_0 consists only of six vertexes and six elements, see Figure 4. None of the vertexes is on the exact surface. First, we study the quadrature error. In order to assess the quadrature error, we compute (39) for three different refinement variants. The results are plotted in Figure 5a. In the variant *fixed elements*, we keep \mathcal{T}_0 fixed and raise the number of quadrature points and therefore the degree of exactness of the quadrature rule. In the *variants A*

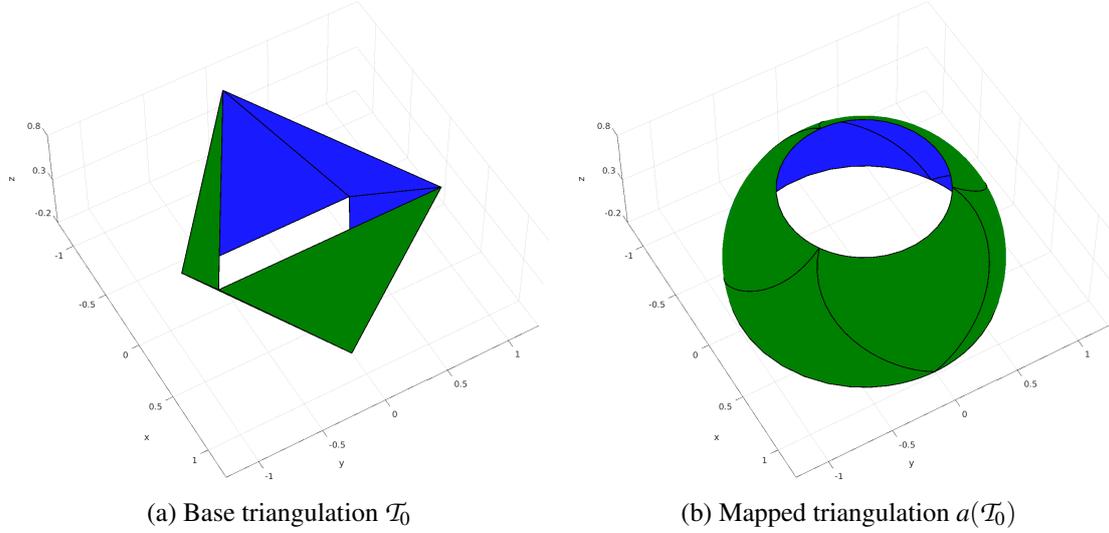


Figure 4: Triangulations for the spherical surface

and B , we uniformly refine the base triangulation \mathcal{T}_0 several times and use only one quadrature point per element. In *variant A*, we map the mesh after each refinement such that all vertices are on the exact surface. In *variant B*, we refine the mesh and leave the newly introduced vertices unchanged as outlined at the end of Section 4. Thus, in this case, Ω_h remains the same for all refinements. In Figure 5a, we see the superior convergence of the variant *fixed elements*. For the other two variants, we can identify a linear convergence behavior. Surprisingly, the error in *variant B* is slightly lower than in *variant A*.

Next, we study the convergence of the FEM under uniform mesh refinement. We do not move the nodes after refinement (according to *variant B*). Thus, the same Ω_h is used in all calculations. As the sought solution we set

$$u^M = x \sin(z). \quad (42)$$

We define $\sigma_N = \{(x, y, z) \in \sigma \mid z < 0\}$ and $\sigma_D = \sigma \cap \sigma_N$. Following the MMS, we derive the source term and the boundary conditions such that (42) is the solution of the considered boundary value problem. The results of the convergence study are presented in Figure 5b. We observe optimal convergence rates, *i.e.* $O(h^{p+1})$ -convergence where h is a characteristic element length.

5.2 Deformed torus

This example is inspired from the flower-shape example in [46, 51]. However, here we consider a 3D surface of a deformed torus. The level-set function of the surface is

$$\begin{aligned} \phi(x, y, z) &= \left(x^2 + y^2 + z^2 + R(x/y)^2 - r^2 \right)^2 - 4R(x/y)^2(x^2 + y^2), \\ R(\tau) &= 1 + r_0 \sin(6\tau), \quad r = 0.2, \quad r_0 = 0.1. \end{aligned} \quad (43)$$

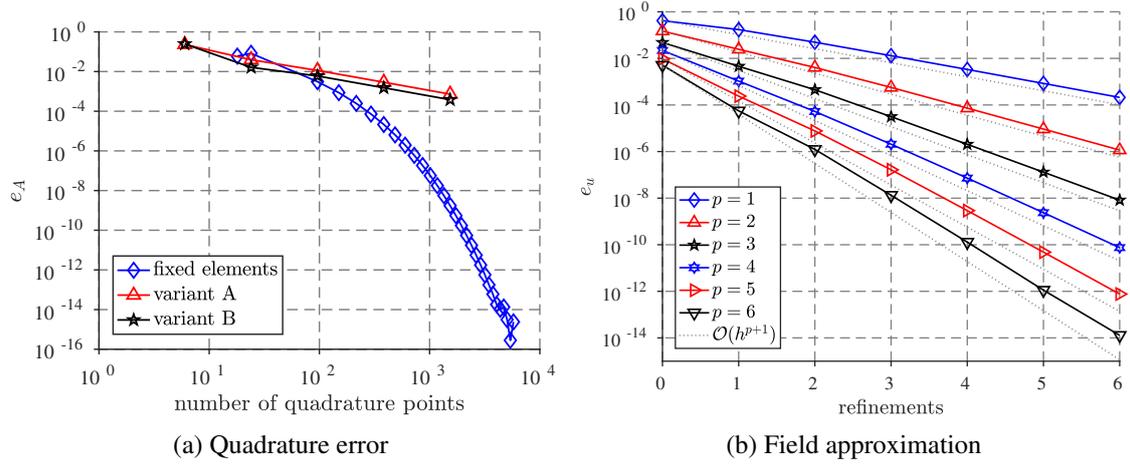


Figure 5: Convergence results for the spherical surface

We consider the zero level surface in $B = [-1.5, 1.5] \times [-1.5, 1.5] \times [0, 0.5]$. This surface is also parametrically defined by

$$\begin{aligned} \Omega &= \{(x, y, z) \mid x = (R(\theta^1) + r \cos(\theta^2)) \cos(\theta^1), \\ & \quad y = (R(\theta^1) + r \cos(\theta^2)) \sin(\theta^1), \\ & \quad z = r \sin(\theta^2), \theta^1 \in [0, \pi], \theta^2 \in [0, 2\pi]\}. \end{aligned} \quad (44)$$

We notice that for $r_0 = 0$ a torus is obtained. The base triangulation (see Figure 6a) is obtained in three steps. First, we apply the Matlab-function *isosurface* to a structured $10 \times 10 \times 5$ grid to obtain a triangulation of the surface. This function implements a variant of the Marching Cubes Algorithm [52]. In the second step, we map all vertices onto the exact surface. In the final step, we eliminate small triangles using the Matlab-function *reducepatch* and obtain the base triangulation. The mapped base triangulation is displayed in Figure 6b.

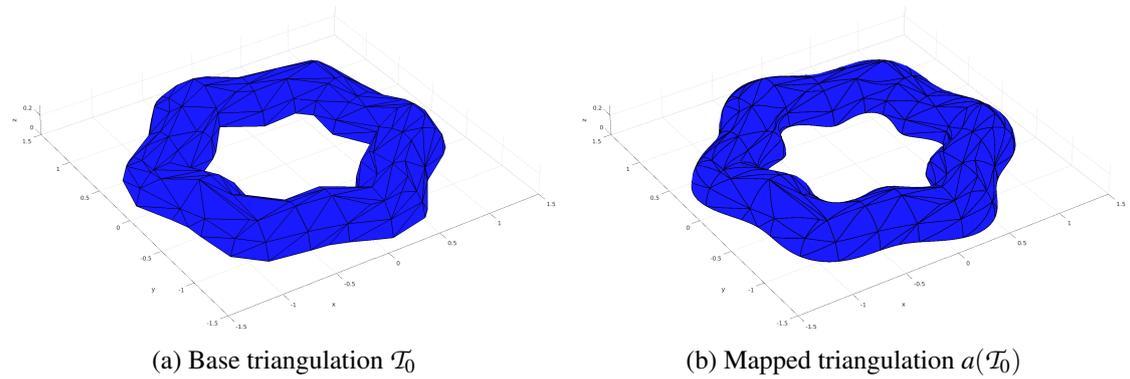


Figure 6: Triangulations for the deformed torus example

We study the convergence of e_A . To this end, we compute the area with different quadrature

rules with increasing degree of exactness. The super-algebraic convergence can be observed in Figure 7a.

For the investigation of the convergence of the field approximation, we consider the manufactured solution

$$u^M = z^2 \cos(2\pi x) \cos(2\pi y). \quad (45)$$

We take the whole boundary as Dirichlet boundary. Again, we perform uniform refinement of the triangulation without subsequent node manipulation. We observe optimal convergence rates in Figure 7b.

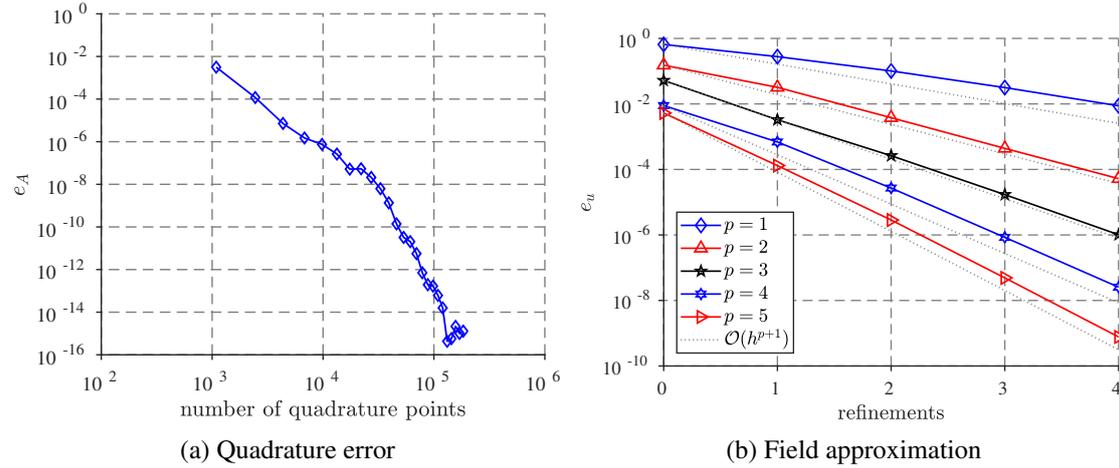


Figure 7: Convergence results for the deformed torus example

5.3 Gyroid

In this example, we consider a complexly shaped surface. The surface of interest is given by the zero level of the level-set function

$$\phi(x, y, z) = \sin(x) \cos(y) + \sin(y) \cos(z) + \sin(z) \cos(x) \quad (46)$$

and $B = [-1, 1]^3$. The exact surface and the base triangulation \mathcal{T}_0 with 240 elements are depicted in Figure 8. The base triangulation was obtained by making use of the Matlab-function *isosurface* applied to a structured $5 \times 5 \times 5$ grid. Since we are not aware of a parametric description of the gyroid, we study only the convergence of the field approximation in this example. Therefore, we prescribe the solution as

$$u^M = x^2 \sin(y) \exp(z) \quad (47)$$

and consider the case $\sigma_D = \sigma$. In Figure 9, the results of a convergence study are presented. We see the optimal convergence rates. This constitutes the validity of the proposed method, even in geometrically complex examples.

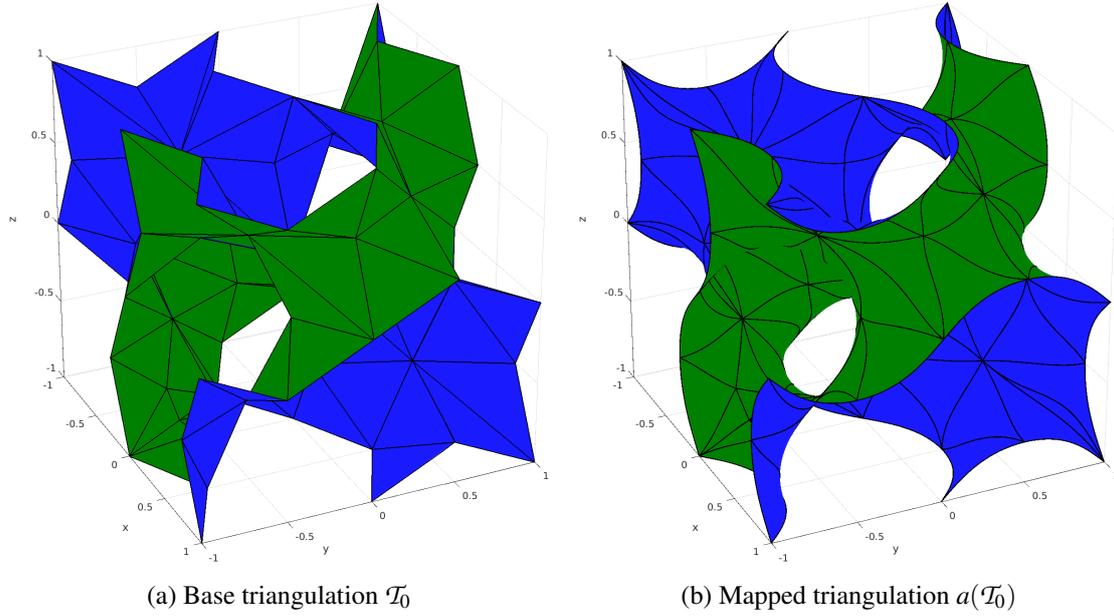


Figure 8: Triangulations for the gyroid example

5.4 Pot problem

In this example, we apply the present method for the analysis of an empty pot. The geometry of the pot is defined by the combination of three level-set functions. In order to realize boolean operations on implicitly given domains, the max-function can be utilized. However, then the resulting surface is not smooth in general. We follow [53] to obtain a smooth final surface. To this end, we have used the exterior blending with the parameters $\delta = 0.75$ and $\varepsilon = 0.0025$. This blending builds upon a smooth approximation of the max-function. The individual level-set functions are

$$\begin{aligned}
 \phi_1 &= 0.8 + \frac{z^2}{10} - x^2 - y^2 \\
 \phi_2 &= \left(R - \sqrt{(x-1)^2 + y^2} \right)^2 + (z-0.8)^2 - r^2 \\
 \phi_3 &= \left(R - \sqrt{(x+1)^2 + y^2} \right)^2 + (z-0.8)^2 - r^2
 \end{aligned} \tag{48}$$

with $R = 0.6$ and $r = 0.15$. Instead of modeling the bottom of the pot, we apply a Dirichlet boundary condition on the bottom curve, *i.e.* $u = 1$. The base triangulation and the mapped base triangulation are depicted in Figure 10. We have obtained the base triangulation in four steps. In the first step, we applied the Matlab-function *isosurface* on a $30 \times 20 \times 20$ structured grid. In order to capture the topology of the problem, we had to use such a fine grid. In the second step, we mapped all vertices on the surface. In the third step, we applied the Matlab-function *reducepatch* in order to keep the mesh as small as possible. However, the resulting mesh of these first three steps is not valid since a negative determinate occurs in one element. Thus, the fourth

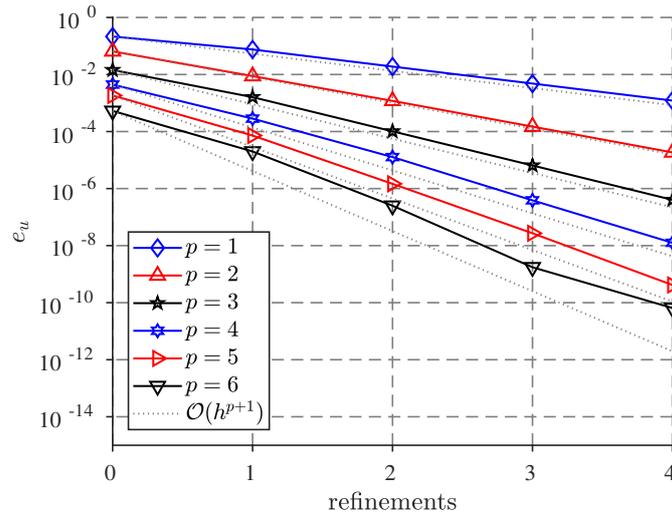


Figure 9: Convergence of the field approximation in the gyroid example

step consists of an adaptive refinement of the invalid triangle. In this example we keep the mesh fixed and consider the p -version of the FEM. The advantage of the present method is that once the valid base triangulation is set up no care about geometry approximation has to be taken.

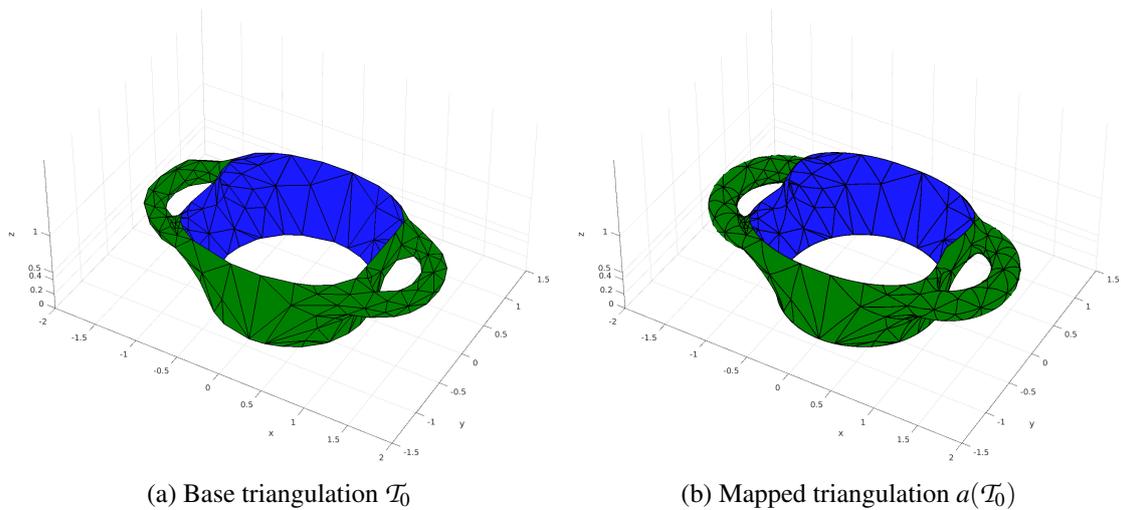


Figure 10: Triangulations for the pot problem

We study the convergence of the method in this example. For this purpose three meshes are used, which are the mesh given in Figure 10 and two uniform refinements. Since the resulting level-set function is highly complex, we were not able to compute a source term by means of the MMS in a reasonable time. Hence, the error in the integral of the solution $\tilde{u} = \int_{\Omega} u \, dx$ is compared with a reference value $\tilde{u} = 3.4718104451902$, the solution of the finest mesh and $p = 18$. The error plot can be found in Figure 11. Concerning the numerical effort the number

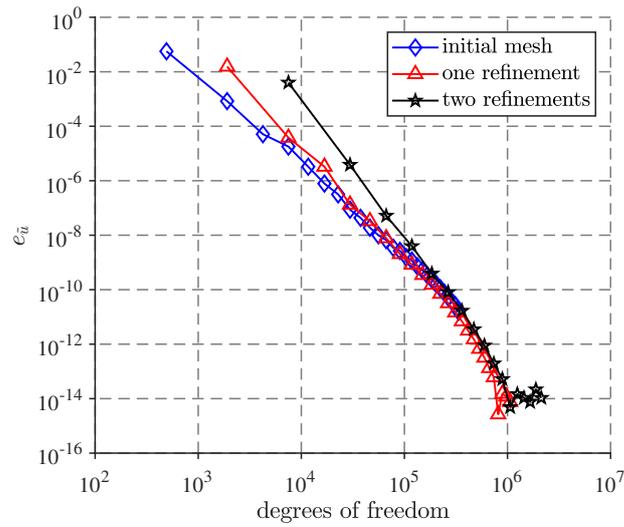


Figure 11: Convergence of the field approximation in the pot model

of dofs to obtain a distinct error level is considered. Up to an error level of 10^{-8} the initial mesh is the most efficient but to obtain lower error levels the first refinement shows the best efficiency. This shows that increasing the order of the shape functions is more efficient than a mesh refinement in this example.

In Figure 12, the solutions for $p = 1, 2, 3, 4$ and the initial mesh are shown. In the left column the solution u is given. We observe already for the linear method only minor deviations from the other solutions. Moreover, the solutions for $p = 2, 3, 4$ agree visually. This is in contrast to the right column where $\|\nabla_{\Omega} u\|$ the norm of the flux vector is shown. Here, the linear method yields only piecewise constant values and thus only a coarse approximation. The quality of the flux solution is considerably improved for $p = 2$. Nevertheless, in regions of high curvature, a deviation from the solutions obtained with the higher order methods is present. Finally, we do not observe visual differences between the solutions of $p = 3$ and $p = 4$.

6 Conclusion

A high order finite element method for surface problems has been presented. The main feature of the method is that an approximating triangulation is mapped onto the exact geometry of the considered problem. Thus, the method introduces no geometry error. This enables the construction of the high order FEM following standard concepts. The element shape functions are constructed on the reference triangle and pieced together to continuous shape functions in the usual way. Due to the exact geometry, refinements are easier compared to methods based on an approximative geometry handling. We only have to perform element subdivision without any adaption of the refined mesh to the exact surface. The only requirement is to have one valid initial surface mesh.

We have shown the capabilities of the method in four examples. In order to assess the con-

vergence behavior the method of manufactured solutions has been utilized. We investigated the convergence at a spherical surface, a deformed torus, and a gyroid surface. In all numerical experiments, we observe optimal convergence rates in the asymptotic range. Finally, we applied the method to a pot model where the geometry is given by the smooth combination of three level-set functions.

We conclude that the method works well in geometrically challenging examples provided that a valid base triangulation is available. We discussed conditions for such a mesh. In the present paper, we checked them at a number of points in each triangle and used local refinements if violated. In future work, the conditions should be checked by a more elaborate approach.

Like in all finite element methods, the present method requires integration over the problem domain. Due to the exact geometry, no refinement of the geometry representation is necessary once a appropriate base triangulation is available. Thus, the integration can be numerically realized with exponential convergence using standard quadrature rules. As outlined in the introduction surface integration is of interest in embedded/fictitious domain methods and the Trace-FEM/CutFEM, where the surface lies within cubes and tetrahedrons. We believe that the present bounding box setting can also be readily applied to tetrahedrons and the developed mapping techniques can also be used in those problems.

In the pot model example, the method has been applied to a smooth geometry defined by three level-set functions. In future work, it is planned to extend the present method to non-smooth Constructive Solid Geometry models. There, the challenge is the exact treatment of corners and edges, which occur not on the boundary.

References

- [1] M. Bischoff, K.-U. Bletzinger, W. Wall, E. Ramm, Models and finite elements for thin-walled structures, in: E. Stein, R. de Borst, T. Hughes (Eds.), *Encyclopedia of Computational Mechanics*, Vol. 2, Wiley Online Library, 2004, Ch. 3, pp. 59–137.
- [2] P. Hansbo, M. Larson, F. Larsson, Tangential differential calculus and the finite element modeling of a large deformation elastic membrane problem, *Comput. Mech.* 56 (1) (2015) 87–95.
- [3] D. Chapelle, K.-J. Bathe, *The finite element analysis of shells*, Springer Science & Business Media, 2010.
- [4] I. Nitschke, A. Voigt, J. Wensch, A finite element approach to incompressible two-phase flow on manifolds, *J. Fluid Mech.* 708 (2012) 418–438.
- [5] S. Reuther, A. Voigt, The interplay of curvature and vortices in flow on curved surfaces, *Multiscale Model. Simul.* 13 (2) (2015) 632–643.
- [6] S. Gross, A. Reusken, *Numerical methods for two-phase incompressible flows*, Vol. 40, Springer Science & Business Media, 2011.
- [7] K. Deckelnick, G. Dziuk, C. Elliott, Computation of geometric partial differential equations and mean curvature flow, *Acta Numer.* 14 (2005) 139–232.

- [8] M. Bertalmío, L.-T. Cheng, S. Osher, G. Sapiro, Variational problems and partial differential equations on implicit surfaces, *J. Comput. Phys.* 174 (2) (2001) 759–780.
- [9] G. Dziuk, C. Elliott, Finite element methods for surface PDEs, *Acta Numer.* 22 (2013) 289–396.
- [10] G. Dziuk, Finite elements for the Beltrami operator on arbitrary surfaces, in: S. Hildebrandt, R. Leis (Eds.), *Partial Differential Equations and Calculus of Variations*, Vol. 1357, Springer, 1988, pp. 142–155.
- [11] A. Demlow, G. Dziuk, An adaptive finite element method for the Laplace-Beltrami operator on implicitly defined surfaces, *SIAM J. Numer. Anal.* 45 (1) (2007) 421–442.
- [12] A. Demlow, Higher-order finite element methods and pointwise error estimates for elliptic problems on surfaces, *SIAM J. Numer. Anal.* 47 (2) (2009) 805–827.
- [13] J. Grande, A. Reusken, A higher order finite element method for partial differential equations on surfaces, *SIAM J. Numer. Anal.* 54 (1) (2016) 388–414.
- [14] T.-P. Fries, D. Schöllhammer, Higher-order meshing of implicit geometries part II: Approximations on manifolds, *Comput. Methods in Appl. Mech. Eng.* 326 (2017) 270–297.
- [15] C. Eilks, C. Elliott, Numerical simulation of dealloying by surface dissolution via the evolving surface finite element method, *J. Comput. Phys.* 227 (23) (2008) 9727–9741.
- [16] G. Dziuk, C. Elliott, L^2 -estimates for the evolving surface finite element method, *Math. Comput.* 82 (281) (2013) 1–24.
- [17] B. Kovács, B. Li, C. Lubich, C. Guerra, Convergence of finite elements on a solution-driven evolving surface, arXiv preprint arXiv:1607.07170.
- [18] B. Kovács, High-order evolving surface finite element method for parabolic problems on evolving surfaces, *IMA J. of Numer. Anal.* (2017) 1–30.
- [19] C. Elliott, T. Ranner, Finite element analysis for a coupled bulk–surface partial differential equation, *IMA J. of Numer. Anal.* 33 (2) (2013) 377–402.
- [20] J. Greer, A. Bertozzi, G. Sapiro, Fourth order partial differential equations on general geometries, *J. Comput. Phys.* 216 (1) (2006) 216–246.
- [21] J. Greer, An improvement of a recent eulerian method for solving pdes on general geometries, *J. Sci. Comput.* 29 (3) (2006) 321–352.
- [22] M. Burger, Finite element approximation of elliptic partial differential equations on implicit surfaces, *Comput. Vis. Sci.* 12 (3) (2009) 87–100.
- [23] G. Dziuk, C. Elliott, Eulerian finite element method for parabolic PDEs on implicit surfaces, *Interface. Free. Bound.* 10 (1) (2008) 119–138.

- [24] K. Deckelnick, G. Dziuk, C. Elliott, C.-J. Heine, An h-narrow band finite-element method for elliptic equations on implicit surfaces, *IMA J. of Numer. Anal.* 30 (2009) 351–376.
- [25] M. Olshanskii, A. Reusken, J. Grande, A finite element method for elliptic equations on surfaces, *SIAM J. Numer. Anal.* 47 (5) (2009) 3339–3358.
- [26] M. Olshanskii, A. Reusken, Trace finite element methods for PDEs on surfaces, arXiv preprint arXiv:1612.00054.
- [27] E. Burman, P. Hansbo, M. Larson, A stabilized cut finite element method for partial differential equations on surfaces: The Laplace–Beltrami operator, *Comput. Methods in Appl. Mech. Eng.* 285 (2015) 188–207.
- [28] E. Burman, P. Hansbo, M. Larson, A. Massing, Cut finite element methods for partial differential equations on embedded manifolds of arbitrary codimensions, arXiv preprint arXiv:1610.01660.
- [29] A. Demlow, M. Olshanskii, An adaptive surface finite element method based on volume meshes, *SIAM J. Numer. Anal.* 50 (3) (2012) 1624–1647.
- [30] M. Olshanskii, A. Reusken, X. Xu, An Eulerian space-time finite element method for diffusion problems on evolving surfaces, *SIAM J. Numer. Anal.* 52 (3) (2014) 1354–1377.
- [31] J. Parvizian, A. Düster, E. Rank, Finite cell method, *Comput. Mech.* 41 (1) (2007) 121–133.
- [32] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, *Comput. Methods in Appl. Mech. Eng.* 197 (45) (2008) 3768–3782.
- [33] D. Schillinger, M. Ruess, The Finite Cell Method: A review in the context of higher-order structural analysis of CAD and image-based geometric models, *Arch. Comput. Meth. Eng.* 22 (3) (2015) 391–455.
- [34] E. Burman, S. C. P. Hansbo, M. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, *Int. J. Numer. Methods Eng.* 104 (7) (2015) 472–501.
- [35] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261.
- [36] T. Rüberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows, *Comput. Methods in Appl. Mech. Eng.* 209 (2012) 266–283.
- [37] T. Rüberg, F. Cirak, J. Aznar, An unstructured immersed finite element method for nonlinear solid mechanics, *Adv. Model. Simul. Eng. Sci.* 3 (1) (2016) 22. doi:<https://doi.org/10.1186/s40323-016-0077-5>.
- [38] M. Olshanskii, D. Safin, Numerical integration over implicitly defined domains for higher order unfitted finite element methods, *Lobachevskii Journal of Mathematics* 37 (5) (2016) 582–596.

- [39] T.-P. Fries, S. Omerović, Higher-order accurate integration of implicit geometries, *Int. J. Numer. Methods Eng.* 106 (5) (2015) 323–371.
- [40] S. Omerović, T.-P. Fries, Conformal higher-order remeshing schemes for implicitly defined interface problems, *Int. J. Numer. Methods Eng.* 109 (6) (2016) 763–789.
- [41] T.-P. Fries, S. Omerović, D. Schöllhammer, J. Steidl, Higher-order meshing of implicit geometries part I: Integration and interpolation in cut elements, *Comput. Methods in Appl. Mech. Eng.* 313 (2017) 759–784.
- [42] B. Müller, F. Kummer, M. Oberlack, Highly accurate surface and volume integration on implicit domains by means of moment-fitting, *Int. J. Numer. Methods Eng.* 96 (8) (2013) 512–528.
- [43] R. Saye, High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles, *SIAM J. Sci. Comput.* 37 (2) (2015) A993–A1019.
- [44] B. Cockburn, A. Demlow, Hybridizable discontinuous Galerkin and mixed finite element methods for elliptic problems on surfaces, *Math. Comput.* 85 (2016) 2609–2638.
- [45] J. Grande, Analysis of highly accurate finite element based algorithms for computing distances to level sets, *SIAM J. Numer. Anal.* 55 (1) (2017) 376–399.
- [46] C. Lehrenfeld, High order unfitted finite element methods on level set domains using isoparametric mappings, *Comput. Methods in Appl. Mech. Eng.* 300 (2016) 716–733.
- [47] J. Grande, C. Lehrenfeld, A. Reusken, Analysis of a high order trace finite element method for PDEs on level set surfaces, arXiv preprint arXiv:1611.01100.
- [48] S. Rosenberg, The Laplacian on a Riemannian manifold: An introduction to analysis on manifolds, no. 31 in *London Mathematical Society*, Cambridge University Press, 1997.
- [49] J. Schöberl, S. Zaglmayr, High order Nédélec elements with local complete sequence properties, *Compel.* 24 (2) (2005) 374–384.
- [50] E. Burman, P. Hansbo, M. Larson, A cut finite element method with boundary value correction, *Math. Comput.* doi:<https://doi.org/10.1090/mcom/3240>.
- [51] Z. Li, A fast iterative algorithm for elliptic interface problems, *SIAM J. Numer. Anal.* 35 (1) (1998) 230–254.
- [52] W. Lorensen, H. Cline, Marching Cubes: A high resolution 3D surface construction algorithm, in: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87*, ACM, New York, NY, USA, 1987, pp. 163–169.
- [53] Q. Li, *Blending implicit shapes using smooth unit step functions*, WSCG Short Communication Papers Proceedings, UNION Agency, Science Press, Plzen, Czech Republic, 2004, pp. 297–304.

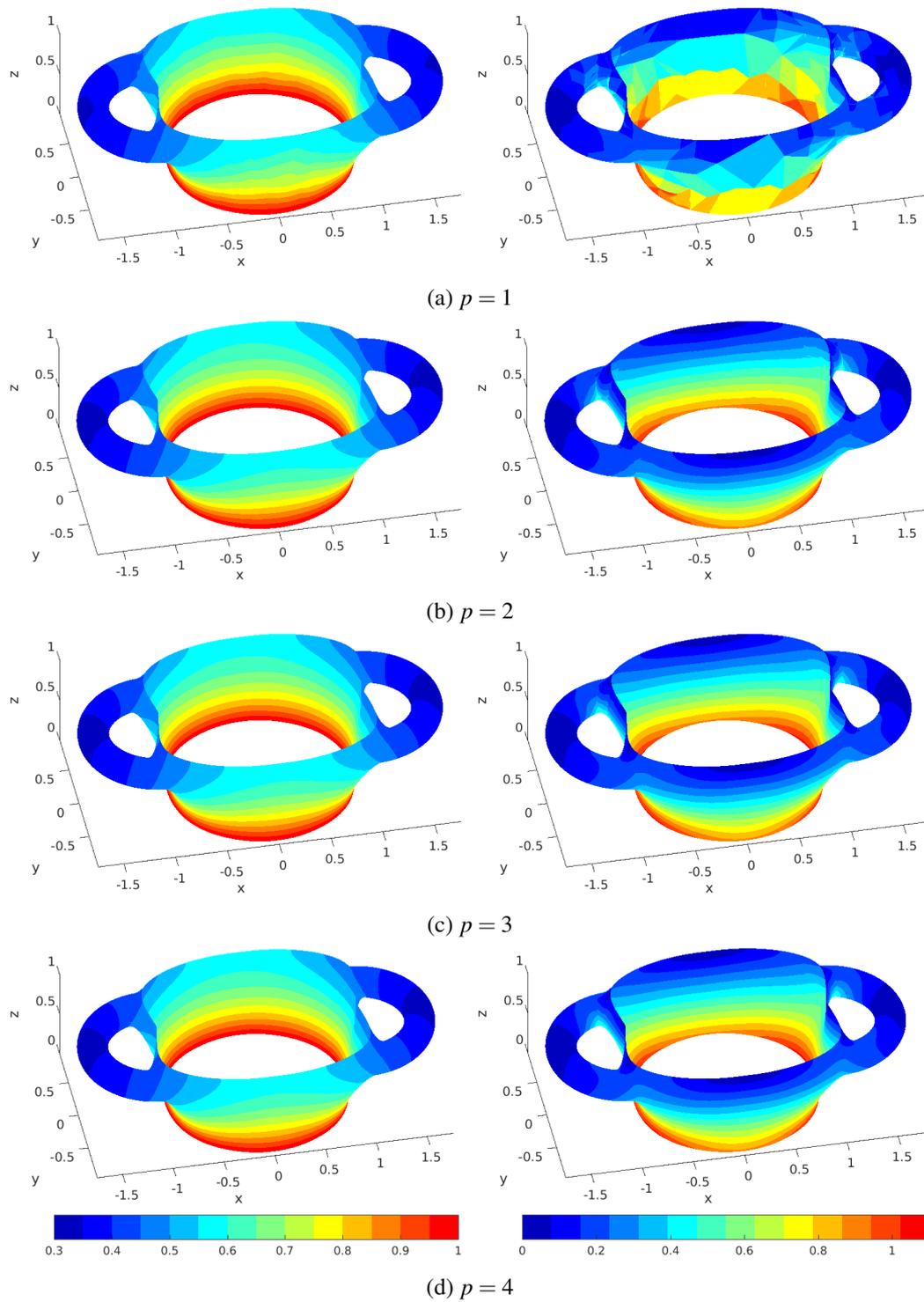


Figure 12: Solutions of the pot problem: u (left column), $||\nabla_{\Omega} u||$ (right column)