HIGHER-ORDER ACCURATE MESHING OF NON-SMOOTH IMPLICITLY DEFINED MANIFOLDS

Jakob W. Stanford[1] Thomas-Peter Fries[2]

[1,2] *Institute of Structural Analysis, Graz University of Technology, Graz, Austria.*
{stanford|fries}@tugraz.at

ABSTRACT

A higher-order accurate meshing algorithm for non-smooth surfaces defined via Boolean set operations from smooth surfaces is presented. Input data are a set of level-set functions. This geometry definition allows an easy detection of non-smooth features such as corners and edges. Furthermore, edges can be treated as intersection curves. At first, a linear reconstruction of the surface is subsequently generated with octree decomposition, corner- and edge-detection and surface meshing. Based on this, first the edges and then the interior nodes of higher-order Lagrangian elements are projected onto the implicit surface. Various projection methods are studied for nodes of edge elements and surface elements. Special attention is paid to the accurate meshing of tangential intersection curves. Optimal convergence properties for approximation problems are confirmed in numerical studies.

Keywords: higher-order finite elements, curved surface meshing, implicit surface, intersection problems, element lifting

1. INTRODUCTION

For finite element analyses, the discretization of the computational domain by means of a mesh is an essential prerequisite that can vastly influence the quality of the results. For analysis with higher-order methods such as the p-FEM, the mesh quality plays an even more critical role [1, 2, 3, 4]. It is not only important to accurately consider boundary features such as corners or edges. Also the placement of the inner- or higher-order element nodes is crucial for achieving the expected rates of convergence. For geometries created within a solid modeling environment, curved meshes for such methods are often created based on their parametric boundary representation. However, numerous challenges are encountered for geometries where trimming and ill-parametrization are involved, which frequently is the case in practical applications. Therefore, a promising alternative is to create such meshes based on the implicit description of the *solid* model, rather than just its boundary. This implies that boundary features such as corners, edges and surfaces are to be reconstructed in that order (specifically) for the purpose of mesh generation.

This contribution concentrates on the automated creation of surface meshes and edge meshes suitable for higher-order accurate finite element methods such as the p-FEM, for the case that geometry definition is available through a purely implicit representation. More specifically, the geometry description consists of multiple solid bodies implicitly described through analytical functions. While the boundary of each body itself must be smooth, their combination through Boolean operations—as usual in the concept of constructive solid geometry (CSG) [5]—may contain C^0 and C^1 continuous edges, hence, leading to non-smooth surfaces. The inclusion of C^1 continuous edges is necessary: They are commonly found in typical engineering geometries, where surface segments of different curvature meet along a line, for example in pipe bends or car bodies. Ignoring them, or not considering them accurately enough, introduces a geometry discretization error dominating and diluting the

convergence behavior of the finite element method as shown in [6]. Their accurate discretization has already been covered in [7].

1.1 Previous Work

Most commonly, higher-order surface meshes for geometries defined within a solid modeling software are created by using the parametric mapping provided by the boundary-representation (B-rep, usually described through NURBSs) of the geometry to map a mesh from some reference domain onto the surface patch. See [8] or text books such as [9, 10] for general B-rep meshing and e.g., [11, 12, 13, 14, 15] and the references cited therein for (curved) higher-order meshing.

An alternative to using the parametric B-rep of a solid model for mesh generation is to use the implicit description. Basically, this amounts to the task of triangulating implicit surfaces. For this task, existing methods are split according to [16] into *spatial partitioning methods*, such as the marching cubes algorithm [17], that discretize the ambient space to then intersect this discretization with the implicit surface and into *continuation methods*. These start by finding a point on the surface (or edge) and then continuously add new points to the triangulation. The marching triangles algorithm [18] or advancing front method [19, 20] belong to this category. See also [21] for a review on surface triangulation methods.

These methods mentioned, however, do not explicitly consider non-smooth surfaces, i.e., those containing corners, edges and transition regions. Additional work to consider sharp features include for example [22, 23]. These, however, do not make use of the multi-surface definition assumed in this paper, and it is not clear how accurate these methods can resolve C^1 continuous edges. When considering this special geometry definition, sharp features such as edges or corners can be defined as intersection of the smooth sub-surfaces. Their detection leads to a surface-surface intersection (SSI) problem, which has also been widely studied in literature [24, 25]. Works such as [26, 27, 28] consider the intersection of two implicitly defined surfaces and [29] even considers tangential intersections resulting in C^1 continuous edges.

So far, only linear geometry discretization was discussed. For higher-order mesh generation, approaches may be divided into *direct methods* that use the parametrization of a solid model to *directly* create a curved mesh. *A posteriori methods*, start with a linear mesh and lift inner nodes into the curved surface. For an overview see [30] and the references therein. Most research in that field assumes the geometry description to be available as B-rep. Regarding the curved meshing of implicitly defined geometries, the series of works [31, 32, 33, 34, 6] deals with the generation of curved higher-order elements based on a purely implicit description. Therein, a special issue is how to place inner nodes such that a smooth isoparametric mapping is implied and that optimal rates of convergence are achieved.

1.2 The Contributions of this Publication

The novelty of this publication is that it presents a way to create a curved mesh of non-smooth implicitly defined geometries that is usable for higher-order methods such as the p-FEM in a sense that the created meshes allow for optimal rates of convergence.

The presented approach is a two-staged bottom-up approach, that first discretizes corners, edges and surfaces in that order, and then in a second stage subsequently creates higher-order meshes for edges and surfaces. On the first glance it might seem as if that could also be achieved by employing any of the existing algorithms for implicit surfaces with simply projecting the higher-order nodes onto the surface a posteriori. But already the linear discretization needs to be created with the goal of higher-order accuracy in mind. Furthermore, not all methods of projecting element nodes onto the surface yield optimal rates of convergence. Therefore this paper studies several options to lift inner nodes onto the surface to achieve optimal rates of convergence.

The proposed method builds upon a linear—but accurate—discretization of the solid domain. As detailed in Sec. 3, this is achieved by an octree decomposition to find corners and points on edges. With a continuation method edges—defined as intersection curves—are traced and finally the surface is triangulated using a combination of marching cubes and advancing front method. This linear mesh is stored in a data structure where surface elements are defined in terms of edges and edges in terms of nodes. With this, edges and elements (faces) may be lifted separately as shown in Sec. 4. Sec. 2 of this paper introduces the notation used and Sec. 5 compares the discussed lifting methods and applies the procedure to some boundary value problem on different geometries.

2. PRELIMINARIES

Behind the manifolds (surfaces or edges) to be meshed lays a solid domain Ω,

$$\Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^3 \;\middle|\; \phi_\Omega(\boldsymbol{x}) < 0 \right\} , \tag{1}$$

where $\phi_\Omega(\boldsymbol{x}) : \mathbb{B}_\Omega \subset \mathbb{R}^3 \to \mathbb{R}$ is a level-set function and $\mathbb{B}_\Omega$ its respective bounding box such that $\Omega \subset \mathbb{B}_\Omega$. The

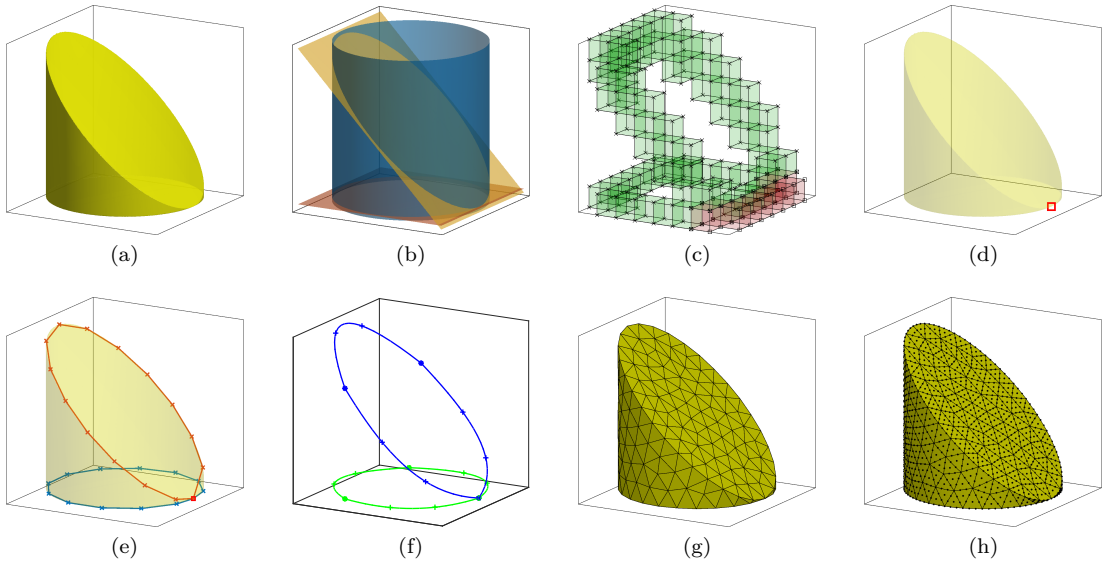


Figure 1: Overview of the process: (a) solid model under consideration, (b) constitutive surfaces, (c) octree decomposition (green cells: 2 intersections, red: ≥ 3 intersections, other cells not shown), (d) detected corners, (e) detected corners and edges, (f) higher-order edge mesh, (g) linear surface mesh, (h) curved surface mesh.

boundary of Ω is defined with the same concept as

$$\Sigma_\Omega = \partial\Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^3 \;\middle|\; \phi_\Omega(\boldsymbol{x}) = 0 \right\} . \tag{2}$$

We refer to Σ_Ω as the *final surface* to be meshed. Per the following definitions, it is a closed 2-manifold. There exist also a number of analytic C^∞-smooth level-set functions $\phi_i(\boldsymbol{x}) : \mathbb{B}_i \subset \mathbb{R}^3 \to \mathbb{R}$ to define *smooth surfaces* (or simply *surfaces*)

$$\Sigma_i = \left\{ \boldsymbol{x} \in \mathbb{R}^3 \;\middle|\; \phi_i(\boldsymbol{x}) = 0 \right\} , \tag{3}$$

where $\mathbb{B}_i$ are again the corresponding bounding boxes to each $\phi_i(\boldsymbol{x})$. Herein, it is assumed that each $\phi_i(\boldsymbol{x})$ describes a distinct surface, that is $\Sigma_i \neq \Sigma_j \;\forall\, i, j,\; i \neq j$. Each Σ_i is also equipped with a surface normal vector

$$\boldsymbol{n}_i(\boldsymbol{x}) = \frac{\nabla\phi_i(\boldsymbol{x})}{\|\nabla\phi_i(\boldsymbol{x})\|} . \tag{4}$$

Similar to the concept of constructive solid geometry, these level-set functions and surfaces together make up the final domain or surface, such that

$$\partial\Omega = \bigcup_i \Sigma_i\big|_{\partial\Omega} \qquad \text{with } \Sigma_i\big|_{\partial\Omega} = \Sigma_i \cap \partial\Omega , \tag{5}$$

where $\Sigma_i\big|_{\partial\Omega}$ are denoted as the *surface segments*. Also note that in practice, $\phi_\Omega(\boldsymbol{x})$ is created from $\phi_i(\boldsymbol{x})$ through Boolean operations. If it is desired to have $\phi_\Omega(\boldsymbol{x})$ available as an analytic function, R-functions [35] may be used to implement the Boolean operations. The intersection Γ_{ij} describes the point set

$$\Gamma_{ij} = \Sigma_i \cap \Sigma_j \cap \Sigma_\Omega , \tag{6}$$

which can contain regular and tangential intersection curves forming C^0 and C^1 continuous edges respectively. It may also contain *branching points* of intersection curves, which are characterized by

$$\|\nabla\phi_i(\boldsymbol{x}) \times \nabla\phi_j(\boldsymbol{x})\| = 0 . \tag{7}$$

Γ_{ij} can however not contain surfaces, due to the limitations on $\phi_i(\boldsymbol{x})$ set above. Additionally, as per [36, Theorem 3], it is a closed, smooth space curve. *Regular corners* are described by the point set

$$\Gamma_{ijk} = \Sigma_i \cap \Sigma_j \cap \Sigma_k \cap \Sigma_\Omega . \tag{8}$$

Together with branching points, these form the sets of *corners* and are denoted by $\partial\Gamma_c$. To facilitate tracing of edges, also *edge segments* Γ_e are defined. These go from corner to corner. Furthermore, edge segments exclude surfaces intersections that do not form an edge on Σ_Ω. Because Σ_i are smooth and branching points are included in the definition of $\partial\Gamma_c$, an edge segment is a smooth space curve, free from branching points and therefore easy to trace with a continuation method. An edge-segment is C^1 continuous or tangential if the tangent vector

$$\boldsymbol{t}_{ij}(\boldsymbol{x}) = \boldsymbol{n}_i(\boldsymbol{x}) \times \boldsymbol{n}_j(\boldsymbol{x}) \tag{9}$$

vanishes, that is $\|\boldsymbol{t}_{ij}\| = 0$. In such a case, $\boldsymbol{t}_{ij}$ points in the direction where the normal curvatures of both surfaces are equal.

Regarding the finite element technology used in this paper, Lagrangian elements of order p with equidistantly spaced element-nodes $\boldsymbol{x}_\ell$ are used. With the

help of the isoparametric mapping

$$\chi(\boldsymbol{r}) \;:\; \Omega_\tau^{\text{ref}} \subset \mathbb{R}^d \;\rightarrow\; \Omega_\tau \subset \mathbb{R}^3 \qquad d = \{1,2\}\,, \quad (10)$$

the Jacobian (in this case the Gram's determinant) is defined as

$$|\mathbf{J}(\boldsymbol{r})| = \sqrt{\det((\nabla\chi(\boldsymbol{r}))^\intercal \cdot \nabla\chi(\boldsymbol{r}))}\,. \qquad (11)$$

3. LINEAR RECONSTRUCTION

The linear discretization lays the foundation for the eventual lifting process and it relies heavily on the concept of geometry definition laid out in Sec. 2.

3.1 Corner Detection and Edge Tracing

With octree decomposition of the bounding box $\mathbb{B}_\Omega$ and parallel intersection testing with respect to each Σ_i, roots for potential intersection curves, branching points and regular corners are *isolated*. The roots are then *located* using (6), (7) and (8) respectively. These equations can be solved with a standard Newton-Raphson scheme. With the multi-level-set concept it is clear that only cells intersected by three or more Σ_i may contain regular corners and the same can be said for branching points and intersection curves with two or more Σ_i-intersections. Hence, only a small subset of cells needs to be checked for roots. Edge segments are then traced starting on respective points using a predictor-corrector-like scheme: First, a start value $\boldsymbol{x}^0_{E,e+1}$ near the next edge-point is created based on the tangent vector of the intersection curve:

$$\boldsymbol{x}^0_{E,e+1} = \boldsymbol{x}_{E,e} + \boldsymbol{t}_{ij}(\boldsymbol{x}_{E,e})\; h_{tr}. \qquad (12)$$

This start value is then projected onto the intersection curve by solving the system of equations:

$$\begin{aligned} \phi_i(\boldsymbol{x}_{E,e+1}) &= 0 \\ \phi_j(\boldsymbol{x}_{E,e+1}) &= 0 \qquad \text{with } \phi_\Omega(\boldsymbol{x}_c) = 0\,, \\ \phi_{\circ,e}(\boldsymbol{x}_{E,e+1}) &= 0 \end{aligned} \qquad (13)$$

where zero-level set of $\phi_{\circ,e}$ describes a sphere, centered at $\boldsymbol{x}_{E,e}$ with radius h_{tr}, which is the predefined step-length. In practice, h_{tr} is set to a value smaller than the minimal distance between two branching points. If no solution to (13) is found, h_{tr} may recursively be halved. Tracing terminates if $\boldsymbol{x}_{E,e+1}$ is closer than $h_{tr}/2$ to a corner or to the start point of the initial point ($\boldsymbol{x}_{E,0}$) on the edge. (13) is solved with the classical Newton-Raphson method using $\boldsymbol{x}^0_{E,e+1}$ as initial guess.

In case a tangential intersection curve is being traced, the approach to solve (13) with a Newton-Raphson scheme would fail or would produce inaccurate results at least. Therefore it is suggested to minimize the objective function

$$F(\boldsymbol{x}) = \phi_i(\boldsymbol{x})^2 + \phi_j(\boldsymbol{x})^2 + \phi_{\circ,e}(\boldsymbol{x})^2 + \tau_{ij}(\boldsymbol{x}) \qquad (14)$$

with the tangentiality constraint

$$\tau_{ij}(\boldsymbol{x}) = \|\nabla\phi_i(\boldsymbol{x}) \times \nabla\phi_j(\boldsymbol{x})\|^2\;. \qquad (15)$$

instead. The tangentiality constraint ensures an accurate solution by specifically enforcing tangentiality. Note that this constraint may also be used to quantify the tangentiality of a given intersection curve. Because τ_{ij} is simply a term added to the objective function, it is technically still an *unconstrained* optimization problem that is solved with the gradient descent method using the step length described in [37].

3.2 Linear Surface Meshing

In a nutshell, surface segments $\Sigma_i\big|_{\partial\Omega}$ are discretized with the following steps: 1) Discretization from a marching cubes scheme. This triangulates the whole Σ_i, i.e., also segments not part of Σ_Ω. 2) Therefore, simplices not completely part of $\Sigma_i\big|_{\partial\Omega}$ are removed. 3) The resulting triangulation is linked with the edge-polygons by an advancing front algorithm, extended for direct meshing of surfaces similar to [38] or [39]. 4) To avoid elements with undesired inner angles, the mesh is smoothed using an extension of distmesh [40] for curved surfaces. At the end of this process, a linear surface mesh Σ_i^h with specified edge-length h is available for linear FEM analyses or for subsequent higher-order meshing.

The combination of advancing front method and marching cubes is needed in order to accommodate for closed surfaces (where there exists no boundary that can serve as initial front) and bounded surfaces (where the edges need to be considered explicitly).

4. HIGHER-ORDER MESHING

4.1 Higher-Order Edge Meshing

For lifting a linear edge mesh Γ_e^h to a curved, higher-order mesh $\Gamma_e^{h,p}$, multiple alternatives may be considered. The first one is to distribute the required amount of higher-order mesh nodes along Γ_e equidistantly and project those nodes onto Γ_{ij}. This yields a curved mesh with approximately equidistant nodal spacing. By using the resulting curved mesh, one can iteratively improve the nodal positions, which leads to a variational approach: To equidistantly place nodes on a curved mesh, an auxiliary boundary value problem (BVP)—specifically the Laplace Beltrami equation (as later defined in (22))—is solved using the finite element method. The idea is to use the solution

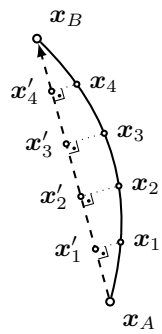


Figure 2: Illustration of equidistant node projection.

of the BVP as arc-length parametrization (approximate and scaled by a constant factor) to place the higher-order mesh nodes. For that, on both ends of the edge Dirichlet boundary conditions (the values 0 and 1 respectively) are set and the BVP is solved. The solution

$$u^h(\boldsymbol{x}) \;:\; \Gamma_e^h \;\longrightarrow\; [0\ 1]$$

is then used to place nodes $\hat{\boldsymbol{x}}$ at desired positions $\hat{u}$ in the arch-length domain by finding $\hat{\boldsymbol{x}}$ such that $u^h(\hat{\boldsymbol{x}}) = \hat{u}$. The node positions obtained this way, lay on the mesh Γ_e^h used in the BVP and hence need to be projected onto Γ_e again.

An arguably simpler method is to element-wise lift inner nodes onto Γ_{ij} such that their normal projection onto the linear connection between the end nodes is equidistant, as illustrated in Fig. 2. For this, only the following system of equations is solved for $\boldsymbol{x}_e$:

$$\begin{aligned} \phi_i(\boldsymbol{x}_e) &= 0 \\ \phi_j(\boldsymbol{x}_e) &= 0\,, \\ \phi_\perp(\boldsymbol{x}_e', \boldsymbol{v}_{AB}, \boldsymbol{x}_e) &= 0 \end{aligned} \tag{16}$$

where the zero iso-line of $\phi_\perp(\boldsymbol{x}_e', \boldsymbol{v}_{AB}, \boldsymbol{x}_e)$ defines a plane normal to the vector $\boldsymbol{v}_{AB} = \boldsymbol{x}_B - \boldsymbol{x}_A$ and going through $\boldsymbol{x}_e'$, which is the normal projection of $\boldsymbol{x}_e$ onto $\overline{AB}$. See Fig. 2 for a sketch.

4.2 Higher-Order Surface Meshing

The remaining task is to lift an existing linear surface mesh to a specified order p. Essentially, this is accomplished by projecting additional mesh-nodes onto the surface. While this sounds trivial, multiple variants exist and not all lead to optimal results. Specifically, there are choices to be made regarding a starting point and the projection method to be used. It is assumed that the linear surface mesh is represented through a hierarchical data structure where elements are defined in terms of element-edges and element-edges in terms of nodes:

$$\text{Element} \rightarrow \text{Edge} \rightarrow \text{Node}$$

This way, element-edges and elements can be lifted separately. This is important, because edges of the original edge-mesh, from hereon denoted *boundary edges*, require a different lifting than the other element-edges, herein denoted *interior edges*. The following discussion of edge-lifting hence only concerns interior edges.

The task of node lifting is to determine the coordinates of a higher-order element node $\boldsymbol{x}_\ell$ in the physical domain. This mapping is split into two stages:

$$\boldsymbol{r}_\ell \;\xrightarrow{\chi(\boldsymbol{r})}\; \boldsymbol{x}_\ell^{(0)} \;\xrightarrow{\mathcal{P}(\boldsymbol{x})}\; \boldsymbol{x}_\ell\,, \tag{17}$$

as depicted in Fig. 3. In the first stage, $\chi(\boldsymbol{r})$ maps the node from the reference domain of the given higher-order element onto some intermediary element in $\mathbb{R}^3$. This provides the start value $\boldsymbol{x}_\ell^{(0)}$ for the projection operator $\mathcal{P}(\boldsymbol{x})$, which lifts the nodes to their final location on the surface. The intermediary element in the simplest case can be a linear element yielding the mapping $\chi^{\text{lin}}(\boldsymbol{r})$. For surface-elements, it can also be an element that is curved based on its already lifted edges, as created with transfinite interpolation [41, 42] denoted by $\chi^{\text{bldg}}(\boldsymbol{r})$ in Fig. 3. Hence, the kind of intermediate element determines the location of the start value of the subsequent projection.

For the projection operator $\mathcal{P}(\boldsymbol{x})$ there are two alternatives: One is a line-search method: Find α such that

$$f(\alpha) := \phi_i(\boldsymbol{x}_\ell^{(0)} + \alpha\ \boldsymbol{v}) = 0\,, \tag{18}$$

where the vector $\boldsymbol{v}$ is the search direction and α a scaling parameter, which is to be determined by a univariate Newton-Raphson scheme. The final node position is then $\boldsymbol{x}_\ell = \boldsymbol{x}_\ell^{(0)} + \alpha\ \boldsymbol{v}$. For the search direction $\boldsymbol{v}$, possibilities are:

1. The surface gradient at the starting point: $\boldsymbol{v} = \nabla\phi_i(\boldsymbol{x}_\ell^{(0)})$, or

2. the surface normal of the intermediate element $\boldsymbol{v} = \boldsymbol{n}_\tau$, or, for edges, the average of the normal vectors of the adjacent elements $\boldsymbol{v} = \boldsymbol{n}_{\tau,1} + \boldsymbol{n}_{\tau,2}$.

The other projection method considered here would be to change the search direction $\boldsymbol{v}$ in each iteration. This leads to an iteration scheme like

$$\boldsymbol{x}_\ell^{(k+1)} = \boldsymbol{x}_\ell^{(k)} - \phi_i(\boldsymbol{x}_\ell^{(k)}) \frac{\nabla\phi_i(\boldsymbol{x}_\ell^{(k)})}{\left\|\nabla\phi_i(\boldsymbol{x}_\ell^{(k)})\right\|^2}\,. \tag{19}$$

In this case, the gradient of $\phi_i(\boldsymbol{x})$ at each iteration step can be interpreted as search direction.

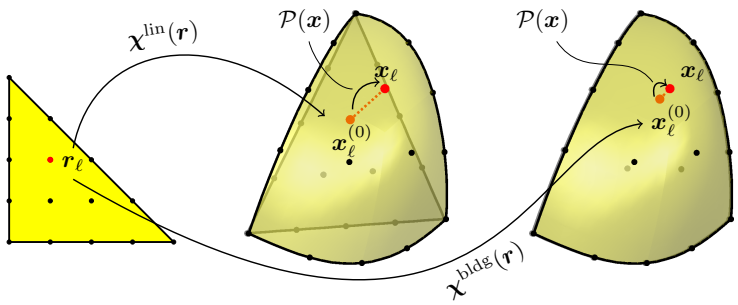


Figure 3: Two-staged mapping procedure for the surface mesh with linear intermediate element (gray, middle figure) and blended intermediate element (right figure).

5. NUMERICAL STUDIES

5.1 Study of Edge Lifting Variants

In this study, the effect of the two discussed possibilities to create higher-order edge meshes on the mesh quality is assessed. For this purpose, a convergence study is run on the geometry shown in Fig. 4a and Fig. 4d. The study was done only on the upper edge.

Referring to the alternatives discussed in Sec. 4.1, the variants studied are as follows:

$V_{E,1}$ lift with equidistant projection,

$V_{E,2}$ mesh with variational approach.

In the study, the error in the approximated edge-length

$$\varepsilon_{L,e} = \left| \frac{L_e^h - L_e^{ref}}{L_e^{ref}} \right| \tag{20}$$

is measured (here, $|\cdot|$ denotes the absolute value of a scalar). Also, the ratio of the Jacobi determinant (as defined in (11)) within an element is computed, which is hence an element-wise measure, as proposed, e.g., in [2]. As mesh-wide measure, the worst of these ratios is taken:

$$\rho_J = \max \frac{\min |\mathbf{J}(\boldsymbol{r})|}{\max |\mathbf{J}(\boldsymbol{r})|} \qquad \forall \boldsymbol{r} \in \Omega_\tau^{\mathrm{ref}} . \tag{21}$$

The possible values range from 1 (best) to 0 (worst). In the study, $|\mathbf{J}(\boldsymbol{r})|$ was evaluated on fine sub-intervals in each element.

The results in Fig. 4 show slightly better convergence behavior with a lower constant for $\varepsilon_{L,e}$ in $V_{E,2}$. Also for ρ_J, $V_{E,2}$ yields better results: Here, the rate of convergence is also dependent on p, whereas in $V_{E,1}$, it seems to only depend on the element size h. As result, $V_{E,2}$ is chosen as edge-lifting variant for the following studies.

5.2 Study of Surface Lifting Variants

The various options on lifting surface elements shall be studied on a full sphere (denoted G_2) and a hollow sphere intersected by two cylinders (denoted G_3) as originally shown in [43]. See Fig. 5 for the coarsest meshes for each geometry.

The options discussed in Sec. 4.2 are summarized and assigned the following variants: For the mapping onto the intermediate element resulting in the start value $\boldsymbol{x}_\ell^{(0)}$

$V_{\chi,1}$ linear element using $\chi^{\mathrm{lin}}(\boldsymbol{r})$,

$V_{\chi,2}$ transfinite interpolation from edges with $\chi^{\mathrm{bldg}}(\boldsymbol{r})$,

and for the choice of search direction

$V_{\boldsymbol{v},1}$ line search with $\boldsymbol{v} = \boldsymbol{n}_\tau$ for surface elements, or $\boldsymbol{v} = \boldsymbol{n}_{\tau,1} + \boldsymbol{n}_{\tau,2}$ or for edges,

$V_{\boldsymbol{v},2}$ line search with $\boldsymbol{v} = \nabla\phi_i(\boldsymbol{x}_\ell^{(0)})$,

$V_{\boldsymbol{v},3}$ gradient-descent method with variable search direction.

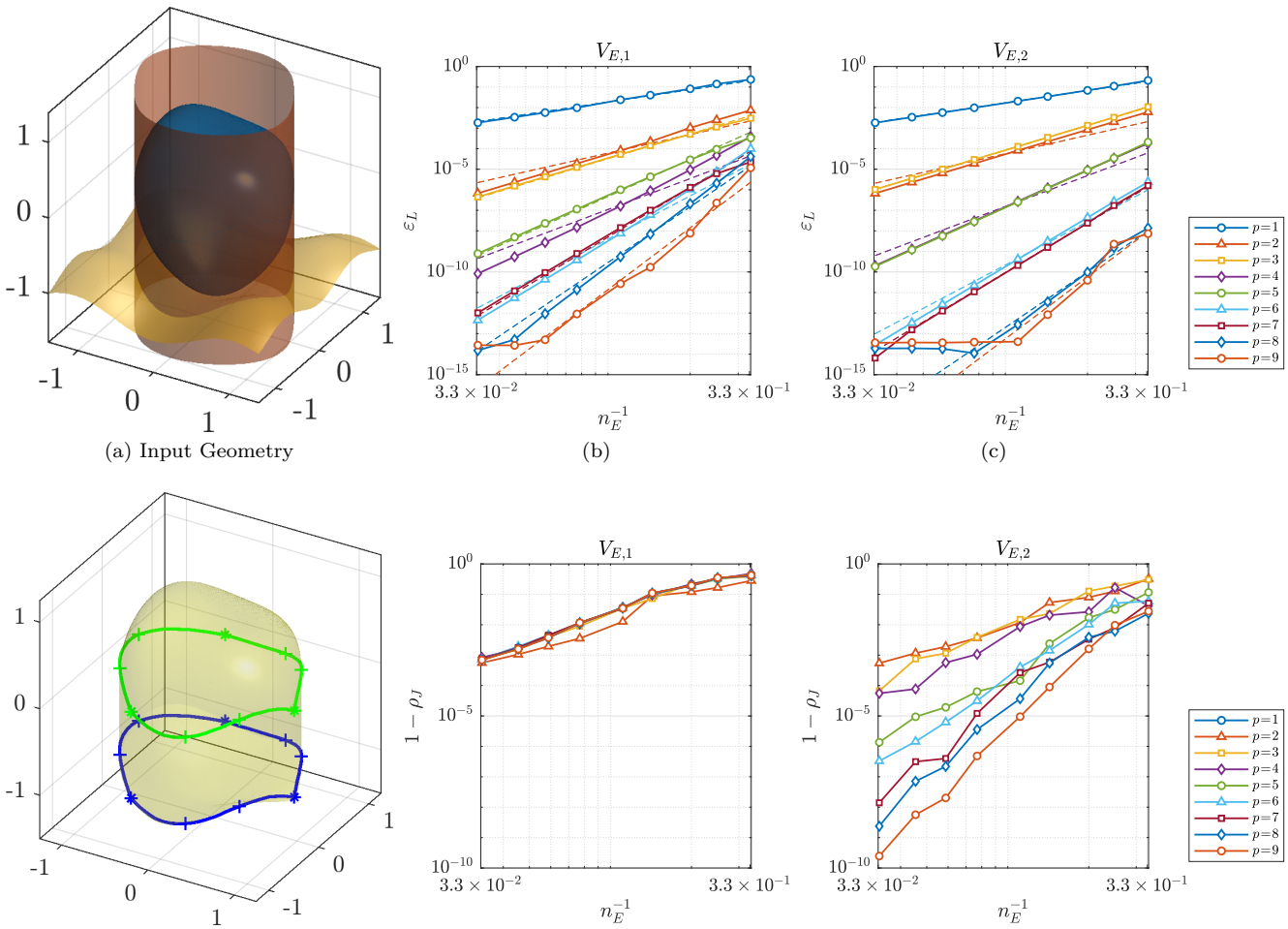
For each possible combination of the variants above, an *hp* convergence study is conducted. As boundary value problem the Laplace-Beltrami problem, as described in [44, 45], was considered. Find $u : \Gamma \mapsto \mathbb{R}$ such that:

$$\begin{aligned} -\Delta_\Gamma u &= f && \text{on } \Omega \\ u &= g_D && \text{on } \partial\Omega_D , \end{aligned} \tag{22}$$

where Δ_Γ is the surface Laplace-operator. In this case $\Omega = \bigcup_i \Sigma_i\big|_{\partial\Omega}$, $\partial\Omega_D = \bigcup_e \Gamma_e$ and

$$g_D = u^{ex} = \sin\left(2x + {}^3\!/_2 y + {}^7\!/_{10} z\right) .$$

The source term f as chosen such that $u = u^{ex}$ on Ω as done in the method of manufactured solutions.



(a) Input Geometry (b) (c)

(d) Edge Meshes (the upper one was considered in the study) (e) (f)

Figure 4: Results for the Edge Lifting Variants: Error in approximated edge lengths ε_L for both variants (b, c). Jacobian ratio ρ_J for both variants (e,f).

The closed form of f is not given here, because the resulting expression is rather lengthy, as its computation involves the surface Laplace-operator, which in turn involves the surfaces normal vector field of $\phi_\Omega(\boldsymbol{x})$.

The continuous weak form of (22) is: Find $u \in H^1(\Omega)$ such that:

$$\int_\Omega \nabla_\Gamma u \cdot \nabla_\Gamma v \, \mathrm{d}x = \int_\Omega f \, v \, \mathrm{d}x \quad \forall v \in H_0^1(\partial\Omega_D). \quad (23)$$

where ∇_Γ is the surface gradient operator, $H^1(\Omega)$ is the usual Sobolev space containing the all square integrable functions with first derivatives in Ω and $H_0^1(\partial\Omega_D) = \{u \in H^1 \mid u\big|_{\partial\Omega_D} = 0\}$. The weak form (23) is discretized with a standard continuous Bubnov-Galerkin method. The resulting linear system of equations is solved using a direct solver for sparse, symmetric systems.

The error of the residual of the strong form of the BVP

$$\varepsilon_{SF}^2 = \sum_\ell \left\| \Delta_\Gamma u + f^2 \right\|_{L_2(\Omega_\ell)}, \quad (24)$$

serves as measure for the quality of the approximated solution.

To assess the regularity of the iso-parametric mapping, again ρ_J as defined in (21) is taken.

The results for ε_{SF} in Fig. 6 attest optimal rates of convergence for $V_{\chi,2}$. For $V_{\chi,1}$ only elements of $p=2$ converge as expected. Since this is also the highest possible order for which standard Lagrangian elements to not have an inner node, it demonstrates the importance of choosing a suitable start value—achieved through a curved intermediate element—for surface projection. For the search direction, there is no clear conclusion from the error plots, especially for G_3, where the error levels seem to be almost identical across the variants. A noteworthy exception is $V_{\boldsymbol{v},3}$ for G_2. The sphere G_2 is smooth and hence is free from boundary edges, which would be lifted with a different method. From that it can be concluded that closed, smooth surfaces can also be lifted with this method.

5.3 Boundary Value Problem on Surface Mesh

In this example, the geometries shown in Fig. 8 are considered for an *hp* convergence study. The geometry G_{18} is considered in [46] as test case due to the branching point created by the intersection of the sphere with the cylinder. G_{24} features some curved, C^1 cont. edges and G_{31} shows a typical engineering geometry. Again, the Laplace-Beltrami problem (22) is solved. In addition to ε_{SF} also the relative Jacobian

$$|\mathbf{J}(\boldsymbol{r})|^{\mathrm{rel}} = \frac{|\mathbf{J}(\boldsymbol{r})|}{2\int_{\Omega_\tau} 1 \, \mathrm{d}\boldsymbol{x}} \quad (25)$$

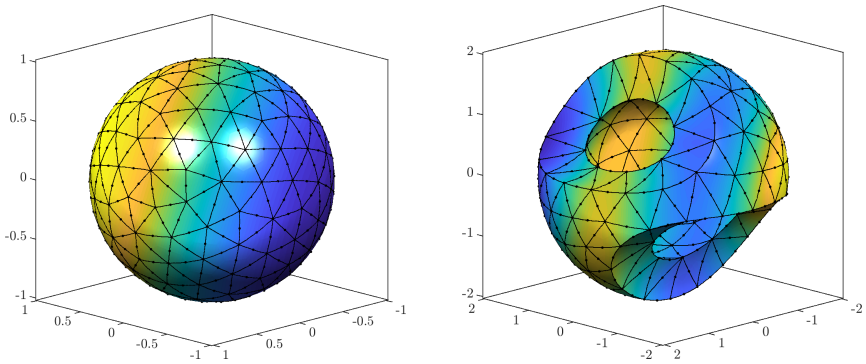
is observed. Values range from 0 to ∞ with 1 being the optimal value indicating a quasi-affine iso-parametric mapping.

The results of the study are collected in Fig. 10. In a nutshell, ε_{SF} reaches optimal rates of convergence for all tested geometries after some pre-asymptotic behavior. In Fig. 9c, the relative Jacobian is plotted for a specific mesh used in the study. It can be seen that the Jacobian varies especially in elements that are part of boundary edges. This behavior might be reduced by optimizing interior edges, for example by Winslow-based smoothing [47, 48] extended to manifolds.

6. CONCLUSION AND OUTLOOK

It was shown how curved, higher-order accurate meshes for non-smooth surfaces can be created. The non-smooth surfaces are the result of set-theoretic operations on smooth surfaces that are defined implicitly with algebraic equations. Special focus was paid to provide meshes that allow optimal rates of convergence in the p-FEM, even for C^1 continuous edges. By defining the final geometry via multiple smooth implicit surfaces, the detection of non-smooth surfaces features (such as corners and edges) is significantly simplified: In detail, the presented method starts with an octree decomposition of the embedding space for root isolation. Then, by root-finding, corners and points on intersection curves are found, which are then traced to arrive at a polygonized version of the edge. Smooth surfaces are first being discretized using marching cubes, then reduced to the faces part of the final surface and then finally, together with the edge mesh, connected using the advancing front method. The result is a linear surface mesh that is lifted by projecting element nodes onto the surface. Different variants for the surface meshing procedure are studied. Numerical examples show optimal rates of convergence for boundary value problems on different geometries.

The presented work can serve as basis for volume meshing of implicit geometries where the constructed higher-order surface meshes as described herein are used as a starting point. It may also be extended to incorporate variational methods to smooth surface meshes.



(a) G_2 (b) G_3

Figure 5: Coarsest meshes and solution to the Laplace-Beltrami problem (with $p=2$) used in studies.

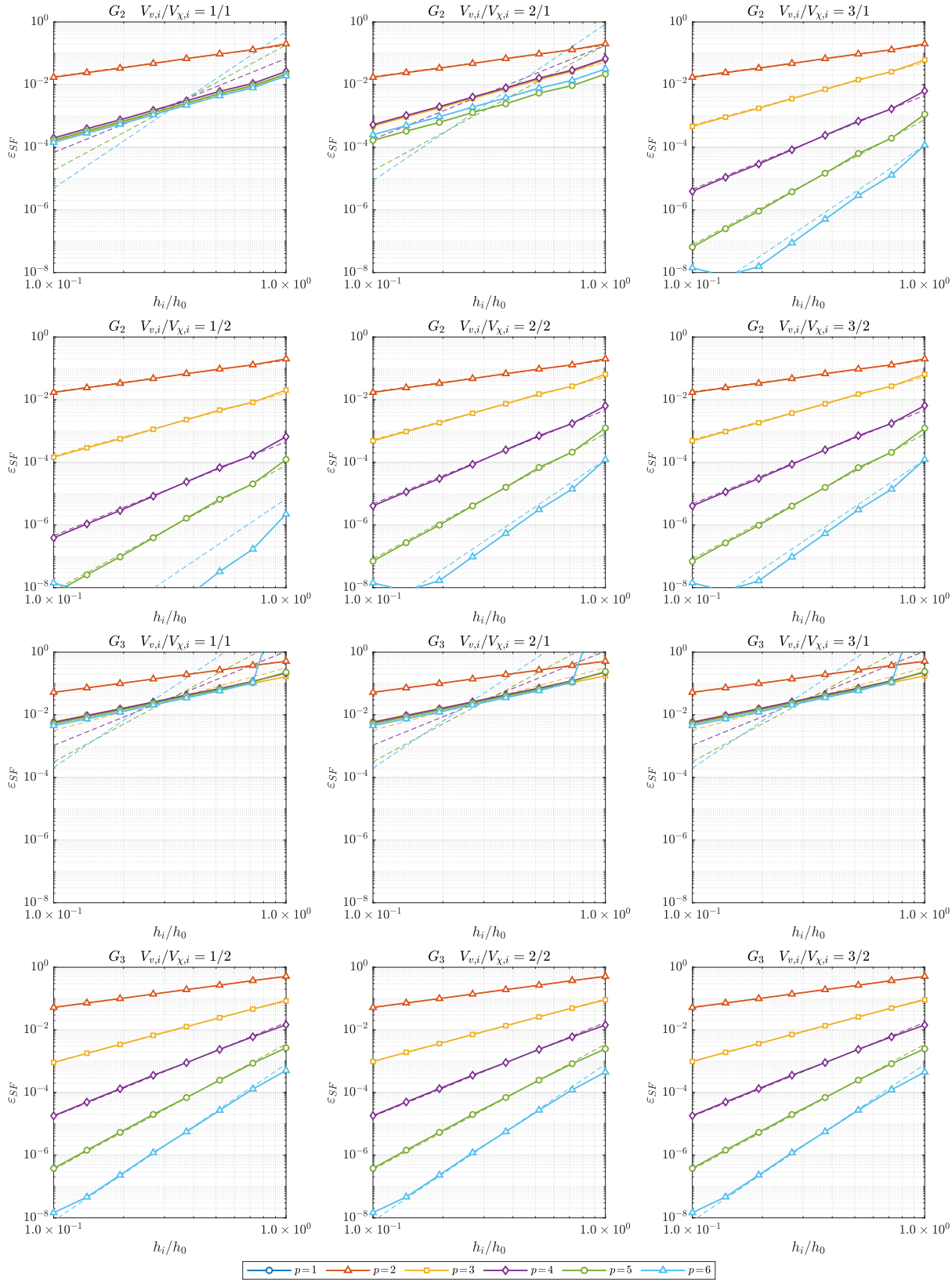


Figure 6: Residual error ε_{SF} for all surface lifting variants on both geometries.

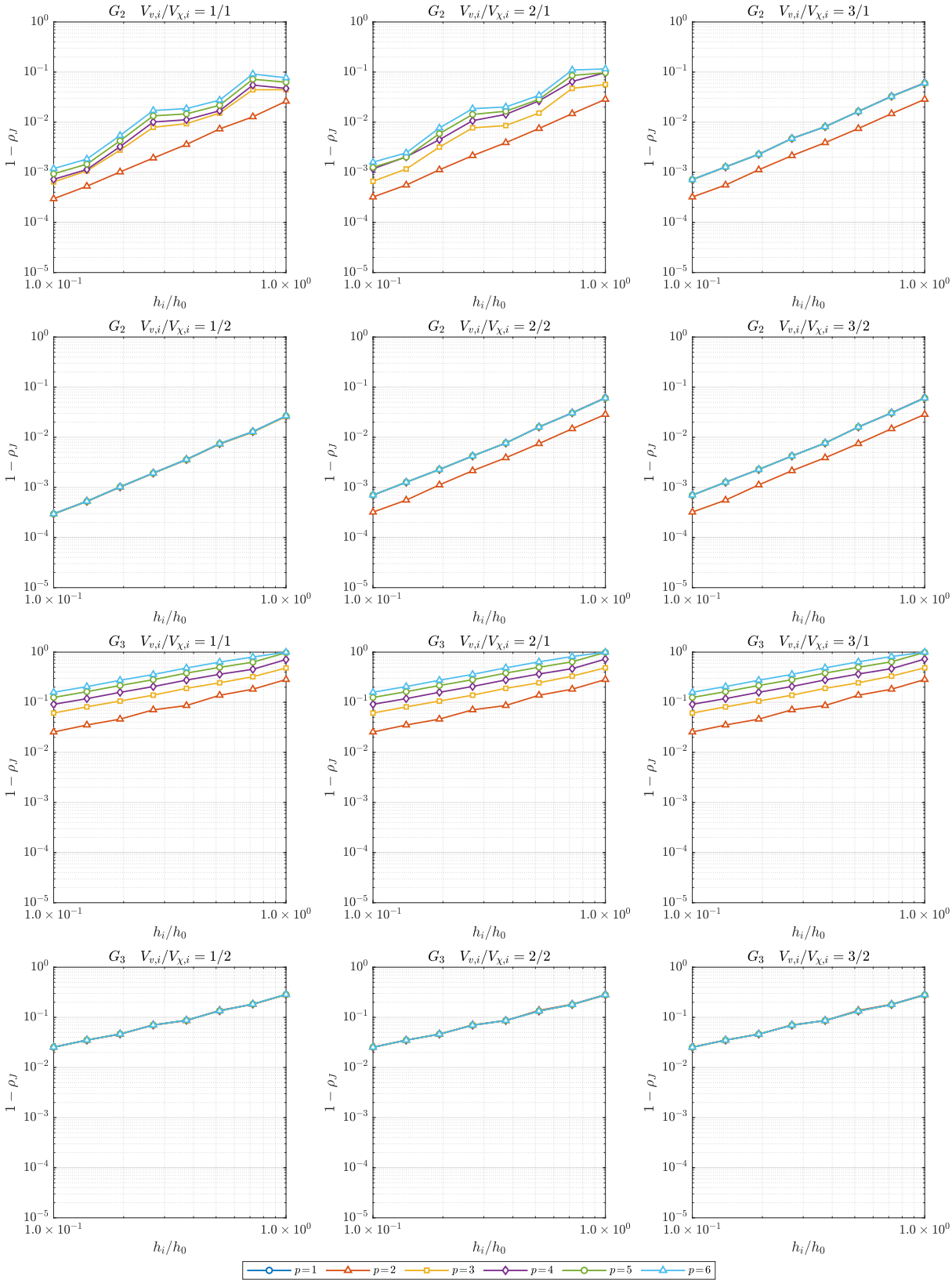
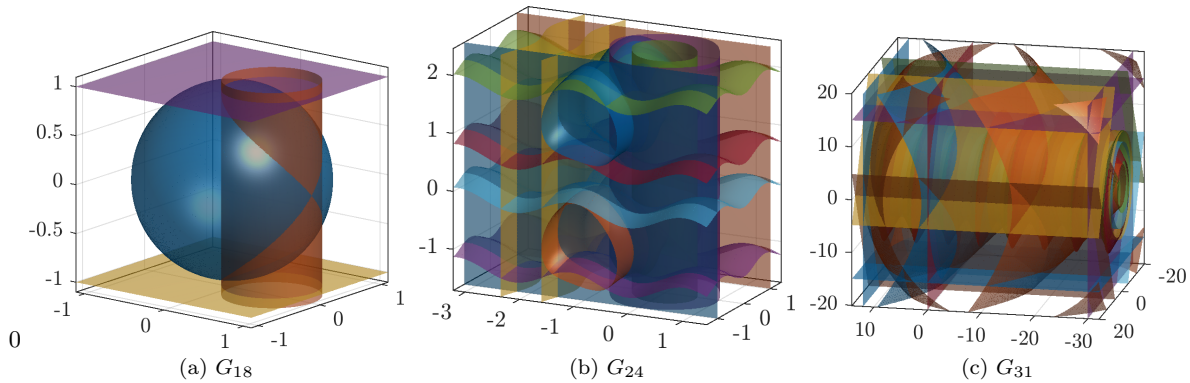
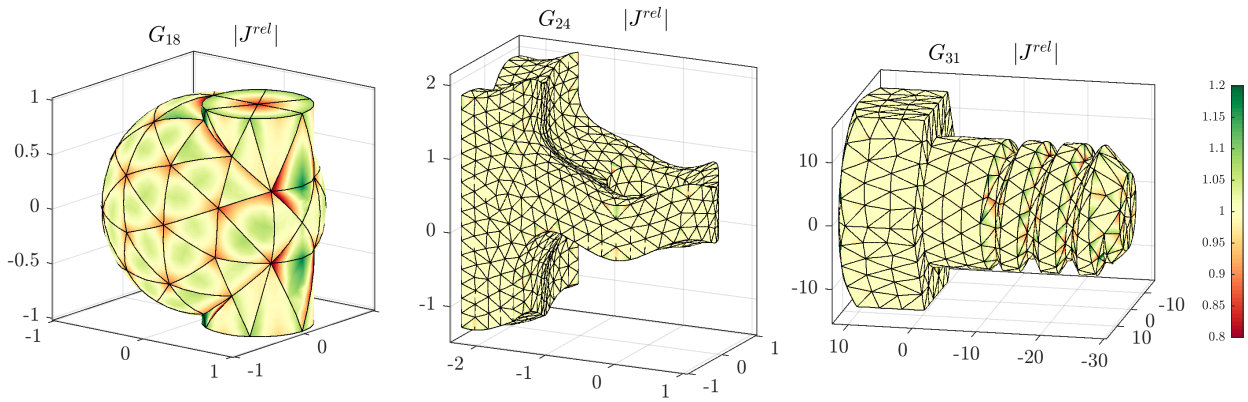


Figure 7: Jacobian ratios ρ_J for all surface lifting variants on both geometries.



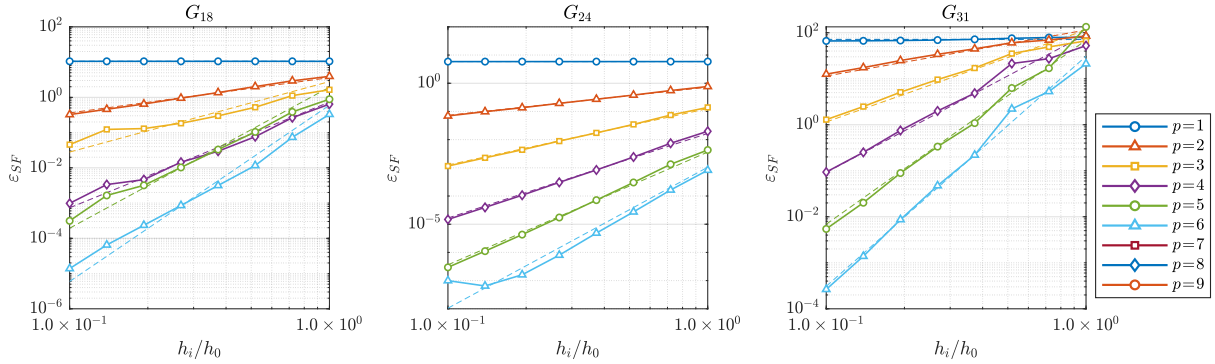
(a) G_{18} (b) G_{24} (c) G_{31}

Figure 8: Input Geometries for the problem considered in Sec. 5.3.



(a) (b) (c)

Figure 9: Approximated solution, residual error and scaled Jacobian for coarsest mesh with $p=6$ for the problem considered in Sec. 5.3.



(a) Condition Number κ (b) Residual Error ε_{SF} (c) Jacobian Ratio ρ_J

Figure 10: Results for the convergence study considered in Sec. 5.3.

References

[1] Dey S., Shephard M.S., Flaherty J.E. "Geometry representation issues associated with p-version finite element computations." *Comp. Methods Appl. Mech. Engrg.*, vol. 150, no. 1, 39–55, 1997

[2] Dey S., O'bara R.M., Shephard M.S. "Curvilinear Mesh Generation in 3D." *Proceedings of the 8th International Meshing Roundtable.* 1999

[3] Solin P., Segeth K., Dolezel I. *Higher-Order Finite Element Methods (Studies in Advanced Mathematics).* CRC Press, Boca Raton, FL, 2003

[4] Botti L. "Influence of Reference-to-Physical Frame Mappings on Approximation Properties of Discontinuous Piecewise Polynomial Spaces." *Journal of Scientific Computing*, vol. 52, no. 3, 675–703, 2011

[5] Requicha A.A. "Representations of rigid solid objects." *Computer Aided Design Modelling, Systems Engineering, CAD-Systems*, pp. 1–78. Springer, Berlin, 1980

[6] Stanford J., Fries T. "A higher-order conformal decomposition finite element method for plane B-rep geometries." *Comp. & Struct.*, 2018

[7] Stanford J., Fries T. "Higher-order accurate meshing of implicitly defined tangential and transversal intersection curves." *Lect. Notes in Comp. Sci. and Engrg.* 2018. (accepted)

[8] Sherwin S.J., Peiró J. "Mesh generation in curvilinear domains using high-order elements." *Internat. J. Numer. Methods Engrg.*, vol. 53, no. 1, 207–223, 2002

[9] Frey P., George P.L. *Mesh Generation: Application to Finite Elements.* John Wiley & Sons, Chichester, 2008

[10] Lo D.S. *Finite Element Mesh Generation.* CRC Press, Boca Raton, FL, 2014

[11] Gargallo-Peiró A., Roca X., Sarrate J. "A surface mesh smoothing and untangling method independent of the CAD parameterization." *Comput. Mech.*, vol. 53, no. 4, 587–609, 2013

[12] Geuzaine C., Remacle J.F. "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities." *Internat. J. Numer. Methods Engrg.*, vol. 79, no. 11, 1309–1331, 2009

[13] Ruiz-Gironés E., Sarrate J., Roca X. "Generation of Curved High-order Meshes with Optimal Quality and Geometric Accuracy." *Procedia Engineering*, vol. 163, 315–327, 2016

[14] Roca X., Gargallo-Peiró A., Sarrate J. "Defining quality measures for high-order planar triangles and curved mesh generation." *Proceedings of the 20th International Meshing Roundtable.* 2011

[15] Sevilla R., Fernández-Méndez S., Huerta A. "NURBS-enhanced finite element method (NEFEM)." *Internat. J. Numer. Methods Engrg.*, vol. 76, no. 1, 56–83, 2008

[16] Gomes A. *Implicit curves and surfaces : mathematics, data structures and algorithms.* Springer, Dordrecht New York, 2009

[17] Lorensen W.E., Cline H.E. "Marching cubes: A high resolution 3D surface construction algorithm." *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, 163–169, 1987

[18] Hartmann E. "A marching method for the triangulation of surfaces." *The Visual Computer*, vol. 14, no. 3, 95–108, 1998

[19] Lo S. "A new mesh generation scheme for arbitrary planar domains." *International Journal for Numerical Methods in Engineering*, vol. 21, no. 8, 1403–1426, 1985

[20] Löhner R., Parikh P. "Generation of three-dimensional unstructured grids by the advancing-front method." *Int. J. Numer. Methods Fluids*, vol. 8, no. 10, 1135–1149, 1988

[21] de Araújo B.R., Lopes D.S., Jepp P., Jorge J.A., Wyvill B. "A Survey on Implicit Surface Polygonization." *ACM Comput. Surv.*, vol. 47, no. 4, 60:1–60:39, 2015

[22] Belokrys-Fedotov A.I., Garanzha V.A., Kudryavtseva L.N. "Generation of Delaunay meshes in implicit domains with edge sharpening." *Comput. Math. Math. Phys.*, vol. 56, no. 11, 1901–1918, 2016

[23] Ohtake Y., Belyaev A.G. "Dual/primal mesh optimization for polygonized implicit surfaces." *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pp. 171–178. ACM, 2002

[24] Patrikalakis N.M. "Interrogation of Surface Intersections." *Geometry Processing for Design and Manufacturing*, pp. 161–185. Society for Industrial and Applied Mathematics, 1992

[25] Patrikalakis N. "Surface-to-surface intersections." *IEEE Computer Graphics and Applications*, vol. 13, no. 1, 89–95, 1993

[26] Abhyankar S.S., Bajaj C.J. "Automatic parameterization of rational curves and surfaces IV: algebraic space curves." *ACM Transactions on Graphics*, vol. 8, no. 4, 325–334, 1989

[27] Asteasu C. "Intersection of arbitrary surfaces." *Computer-Aided Des.*, vol. 20, no. 9, 533–538, 1988

[28] Bajaj C.L., Hoffmann C.M., Lynch R.E., Hopcroft J.E.H. "Tracing surface intersections." *Comput. Aided Geom. Des.*, vol. 5, no. 4, 285–307, 1988

[29] Düldül B.U., Düldül M. "Can we find Willmore-like method for the tangential intersection problems?" *J. Comput. Appl. Math.*, vol. 302, 301–311, 2016

[30] Turner M. *High-Order Mesh Generation For CFD Solvers.* Ph.D. thesis, 2017

[31] Fries T., Omerović S. "Higher-order accurate integration of implicit geometries." *Internat. J. Numer. Methods Engrg.*, vol. 106, no. 5, 323–371, 2016

[32] Fries T.P., Omerović S., Schöllhammer D., Steidl J. "Higher-order meshing of implicit geometries - Part I: Integration and interpolation in cut elements." *Comp. Methods Appl. Mech. Engrg.*, vol. 313, 759–784, 2017

[33] Fries T., Schöllhammer D. "Higher-order meshing of implicit geometries, Part II: Approximations on manifolds." *Comp. Methods Appl. Mech. Engrg.*, vol. 326, 270–297, 2017

[34] Fries T.P. "Higher-order conformal decomposition FEM (CDFEM)." *Comp. Methods Appl. Mech. Engrg.*, vol. 328, 75–98, 2018

[35] Pasko A., Adzhiev V., Sourin A., Savchenko V. "Function representation in geometric modeling: concepts, implementation and applications." *The Visual Computer*, vol. 11, no. 8, 429–446, 1995

[36] Hu C.Y., Maekawa T., Patrikalakis N.M., Ye X. "Robust interval algorithm for surface intersections." *Computer-Aided Design*, vol. 29, no. 9, 617–627, 1997

[37] Barzilai J., Borwein J.M. "Two-Point Step Size Gradient Methods." *IMA J. Numer. Anal.*, vol. 8, no. 1, 141–148, 1988

[38] Nakahashi K., Sharov D. "Direct surface triangulation using the advancing front method." *AIAA paper*, pp. 95–1686, 1995

[39] Lan T., Lo S. "Finite element mesh generation over analytical curved surfaces." *Computers & Structures*, vol. 59, no. 2, 301–309, 1996

[40] Persson P.O., Strang G. "A Simple Mesh Generator in MATLAB." *SIAM Review*, vol. 46, no. 2, 329–345, 2004

[41] Gordon W.J., Hall C.A. "Construction of curvilinear co-ordinate systems and applications to mesh generation." *Internat. J. Numer. Methods Engrg.*, vol. 7, no. 4, 461–477, 1973

[42] Gordon W.J., Hall C.A. "Transfinite element methods: Blending-function interpolation over arbitrary curved element domains." *Numerische Mathematik*, vol. 21, no. 2, 109–129, 1973

[43] Schöberl J. "NETGEN An advancing front 2D/3D-mesh generator based on abstract rules." *Computing and Visualization in Science*, vol. 1, no. 1, 41–52, 1997

[44] Burman E., Hansbo P., Larson M.G., Massing A. "Cut Finite Element Methods for Partial Differential Equations on Embedded Manifolds of Arbitrary Codimensions." *ESAIM: Math. Model Numer. Anal.*, 2018

[45] Dziuk G. "Finite elements for the Beltrami operator on arbitrary surfaces." S. Hildebrandt, R. Leis, editors, *Partial Differential Equations and Calculus of Variations*, vol. 1357 of *Lecture Notes in Mathematics*, pp. 142–155. Springer, Berlin, 1988

[46] Patrikalakis N.M., Maekawa T. *Shape Interrogation for Computer Aided Design and Manufacturing.* Springer, Berlin, 2002

[47] Knupp P.M. "Winslow Smoothing on Two-Dimensional Unstructured Meshes." *Engineering with Computers*, vol. 15, no. 3, 263–268, 1999

[48] Fortunato M., Persson P.O. "High-order unstructured curved mesh generation using the Winslow equations." *J. Comput. Phys.*, vol. 307, 1–14, 2016