NFN - Nationales Forschungsnetzwerk

Geometry + Simulation

http://www.gs.jku.at

Numerical integration on trimmed three-dimensional domains

Felix Scholz, Bert Jüttler

G+S Report No. 85

May 2019

Der Wissenschaftsfonds.

JOHANNES KEPLER UNIVERSITY LINZ

Numerical integration on trimmed three-dimensional domains

Felix Scholz[a], Bert Jüttler[a,b]

[a] *Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Linz, Austria*
[b] *Institute of Applied Geometry, Johannes Kepler University Linz, Austria*

Abstract

We present a novel technique for the numerical integration of trivariate functions on trimmed domains. Our approach combines a linear approximation of the trimming surface with a correction term. The latter term makes it possible to achieve a cubic convergence rate, which is one order higher than the rate obtained by using the linear approximation only. We also present numerical experiments that demonstrate the method's potential for applications in isogeometric analysis.

Keywords: Trimming, isogeometric analysis, numerical integration, quadrature

1. Introduction

The flexibility of NURBS surface representations in computer-aided design is greatly enhanced by the use of trimming. More precisely, only a part of the parameter domain is active, while a region defined by a trimming curve is discarded. For example, the trimming curve can be the result of a surface-surface intersection.

Isogeometric analysis [1, 2] is based on the use of bi- and trivariate NURBS parameterizations of two- and three-dimensional computational domains. Again, the use of trimming operations increases the flexibility of these representations of the geometry. According to the isogeometric paradigm, the same discrete space is employed for the analysis (i.e. the numerical solution of partial differential equations) as for the representation of the geometry, thus unifying the fields of computer-aided design and numerical simulation.

Trimming also plays a role in the context of unfitted finite element methods [3], such as the finite cell method [4]. In this class of methods, the solution is approximated using a background finite element mesh that is intersected with the underlying geometry.

In both situations, efficient and accurate quadrature rules are needed to perform numerical integration on the trimmed elements of the discretization spaces. They are required in order to assemble the Galerkin matrices as well as the right-hand sides (load vectors) of the discrete systems.

Several approaches to numerical integration on trimmed domains have been considered. Most of this earlier work is limited to the two-dimensional case, see [5] and the references therein. We discuss a number of related results, focusing on the trivariate case:

A common approach for numerical quadrature on trimmed domains is to subdivide the trimmed elements into simpler objects, which is sometimes called untrimming, and to find local approximations that can be dealt with by high-order quadrature rules. The accuracy of the overall procedure is then governed by the geometric approximation error.

In the volumetric case, this has been described in [6], where the trimming surface is approximated by piece-wise polynomials, and in [7], using a marching cube-type approach. Tetrahedral elements have also been used for the discretization of partial differential equations on trimmed volumes [8]. In [9], the trimmed patch is subdivided into a number of singular tensor-product B-Spline patches whose boundaries approximate the trimming surface.

Email addresses: felix.scholz@ricam.oeaw.ac.at (Felix Scholz), bert.juettler@jku.at (Bert Jüttler)

Clearly, the success of these approaches depends on the ability to compute surface-surface intersections, which is well-known to be a delicate problem [10]. Among other approaches, the use of interval arithmetics has been proposed to achieve robustness in degenerate or nearly-degenerate situations [11].

In the context of the finite cell method, the smart octree method for quadrature on volumetric domains has been introduced in [12]. It is based on an extended adaptive subdivision of the elements into octants. More precisely, the octant nodes are moved to the trimming surface, in order to increase the accuracy of the result.

An interesting approach has been presented in [13], where the authors show how to construct a specific quadrature rule for each trimmed element with the help of non-linear optimization. As a prerequisite, the method evaluates the integrals of polynomials on a piece-wise linear approximation to the trimmed domain using a subdivision into convex subdomains, and this determines the overall accuracy. The numerical experiments demonstrate that the method is suitable for two-dimensional problems.

Finally, we note that several techniques exist that can help to avoid trimming. Besides untrimming, these include discontinuous Galerkin methods on overlapping patches [14], Schwarz-type domain decomposition methods [15, 16].

The present paper builds on results from [17], where we presented a method for the quadrature on trimmed two-dimensional domains with an implicitly defined trimming function. The trimming function is locally approximated by a linear function whose level-set is used for an initial approximation of the integral. It is shown that the approximation order of the resulting quadrature rule can be increased by adding an error correction term based on a Taylor expansion.

We generalize this idea to the case of trimmed volumetric domains. Again, we assume that the trimming surface is given implicitly as a level set of a function, for example as the result of a Boolean operation on two domains. For parametric trimming surfaces, exact or approximate implicitization [18, 19] has to be applied as a preprocessing step.

The trivariate trimming function is approximated by a linear function using constrained least squares optimization, thereby ensuring a consistent topology. The resulting quadratic order of convergence of the quadrature rule is then increased by adding a simple error correction term. Since this error correction term only consists of a bivariate integral, it preserves the overall computational complexity, making the method very efficient.

We apply our method to the problems of L^2-projection and isogeometric analysis on trimmed volumes. In addition to the numerical quadrature, another challenge lies in the stability of the trimmed basis functions. If one uses the standard B-Spline basis, restricted to the trimmed domain, the supports of the basis function may be arbitrarily small, resulting in a loss of stability. This instability can be addressed using modified bases such as extended B-Splines [20, 21] or immersed B-Splines [22]. Other approaches are the addition of a stabilization term (ghost penalty) to the bilinear form [23] and the omission of basis functions with sufficiently small support [24]. In our numerical examples, we implemented the extended B-Spline basis, which recovers stability by eliminating unstable functions and adding them to a number of stable ones with appropriate weights, such that the approximation power stays the same.

The paper is organized as follows: We first state the considered problem in Section 2. Then we describe the reduction to base cases that our method can handle in Section 3. We introduce the linearized trimmed quadrature rule in Section 4 and its extension using the first order correction term in Section 5. Finally, we present numerical examples for the quadrature and its application to L^2-projection and isogeometric analysis in Section 6.

2. Problem

Given smooth trivariate functions f and τ and an axis-aligned box $B \subset \mathbb{R}^3$, we develop a method for the numerical evaluation of the integral

$$I_\tau f = \int_{B_\tau} f(x, y, z) \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x$$

over the part of B where τ, which is called the *trimming function*, takes positive values,

$$B_\tau = \{(x, y, z) \in B : \tau(x, y, z) > 0\}.$$

More precisely, we approximate the exact integral by using a weighted quadrature rule

$$Q_\tau f = \sum_i w_i f(x_i, y_i, z_i) \approx I_\tau f$$

with quadrature nodes (x_i, y_i, z_i) and weights w_i.

Our approach generalizes earlier work on the two-dimensional case [17]. It consists of three main steps:

1. We subdivide B into smaller boxes until the sign distribution of the trimming function τ at the boxes' vertices is an instance of one of seven base cases.
2. We approximate the trimming function τ by a linear polynomial σ and perform an approximate evaluation of the integral

$$I_\sigma f = \int_{B_\sigma} f(x, y, z) \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x$$

   by Gauss quadrature.
3. We improve this initial approximation to $I_\tau f$ by adding an error correction term that is derived from the Taylor expansion of the linear interpolant $I_{\sigma+u(\tau-\sigma)} f$ at $u = 0$. This results in a bivariate integral, which is again evaluated via Gauss quadrature.

Summing up, the quadrature rule Q_τ consists of two types of quadrature nodes and weights in each cell: the nodes and weights of the trivariate Gauss quadrature performed in the second step and the nodes and weights of the bivariate Gauss quadrature needed to evaluate the error correction term in the third step.

The next three sections discuss the three steps in more detail.

3. Subdivision and base cases

Our method is designed to be able to treat a number of base cases, which are characterized by the sign distribution of the trimming function on the vertices of B. Intuitively, the sign distribution captures how the trimming surface cuts the box B. In addition, we only treat boxes, whose maximum edge length does not exceed a user-defined constant h, in order to maintain the accuracy of the numerical integration.

If we distinguish between positive and non-positive signs of τ at the vertices of B, we arrive at $2^8 = 256$ possible cases. However, this number can be reduced significantly by considering the 48 isometries of the cube (i.e. the rotations and reflections of the cell) and possibly inverting the signs of τ at the vertices. This results in the 15 cube configurations of the marching cube algorithm.

Five of these configurations, which are shown in Figure 1, correspond to a single component of the trimming surface within B. More precisely, if the trimming surface has a single component within B, then the sign distribution belongs to one of these configurations. These are the base cases of our method, which are handled directly. In addition we also handle the case of only positive signs, which corresponds to an untrimmed box $B = B_\tau$. Clearly, this case does not require any particular treatment, simple Gauss quadrature is sufficient.

The remaining nine configurations are dealt with by splitting the box B into eight sub-boxes. The splitting is applied recursively until each generated box is an instance of one of the six base cases. Similarly, we split the box B into eight sub-boxes if the maximum edge length exceeds h. The total number of generated boxes satisfies $N_h = O(\frac{1}{h^3})$ if the trimming surface is regular within B.

It should be noted that the trimming surface may possess several components within B even if the sign distribution belongs to one of the base cases. As we shall see later, our technique gives correct results also in these situations.

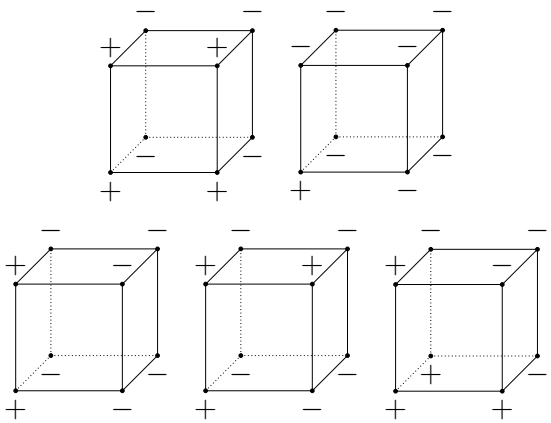

Figure 1: The base cases defined up to rotations, reflections and sign inversion.

4. Linearized trimmed quadrature rule

Consider a quadrature cell

$$B = [x_0, x_1] \times [y_0, y_1] \times [z_0, z_1].$$

We approximate the trimming function τ by a linear function

$$\sigma(x, y, z) = \sigma_1 x + \sigma_2 y + \sigma_3 z + \sigma_4 \tag{1}$$

and apply Gauss quadrature to the integral of f over

$$B_\sigma = \{(x, y, z) \in \Omega : \sigma(x, y, z) > 0\}.$$

The approximation ensures that the signs of σ on the vertices of B agree with the signs of τ. While in the two-dimensional case such an approximation can be obtained quite easily, we need to solve a constrained optimization problem in the three-dimensional case. This is described in the next section.

Since the zero-level set of the linear function σ is a plane, its intersection with B is a polygon. Hence, B_σ is a polyhedron. In order to get quadrature points and weights for B_σ we need to find a multi-patch parameterization over the reference domain $[0, 1]^3$. This is described in section 4.2.

4.1. Constrained least squares approximation

The approximation of the trimming function τ by a linear function σ of the form (1) with the additional condition that the plane $\sigma(x, y, z) = 0$ intersects B in the same edges as τ leads to a constrained least squares problem:

$$\begin{aligned} &\min_{\sigma \in \mathbb{R}^4} \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{k=0}^{1} (\sigma_1 x_i + \sigma_2 y_j + \sigma_3 z_k + \sigma_4 - \tau_{ijk})^2 \\ &\text{subject to } S\sigma \geq 0. \end{aligned} \tag{2}$$

Here, τ_{ijk} is the value of τ on the vertex (x_i, y_j, z_k) of B.

The matrix S represents the conditions on the intersections of the plane with the edges of B. Since we allow for the zero-level set of σ to go through the vertices, the sign of $\sigma(x, y, z)$ on each vertex of B needs to be either the same as the sign of τ_{ijk} or zero. This gives the eight inequalities

$$\begin{cases} \sigma_1 x_i + \sigma_2 y_j + \sigma_3 z_j + \sigma_4 \geq 0 & \text{for } \tau_{ijk} \geq 0 \\ -\sigma_1 x_i - \sigma_2 y_j - \sigma_3 z_j - \sigma_4 \geq 0 & \text{for } \tau_{ijk} \leq 0. \end{cases}$$

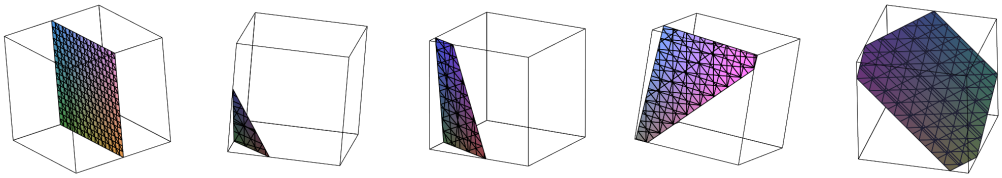

Figure 2: Example level sets for the five base cases.

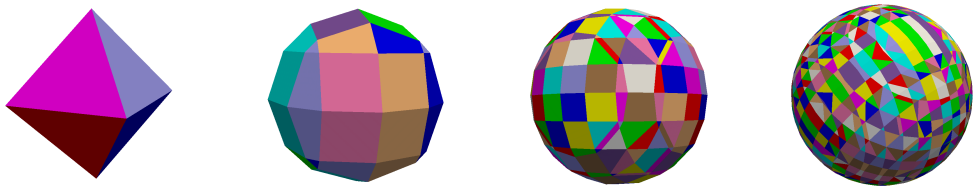

Figure 3: Several steps of refinement of the linear approximation to the ball.

Depending on the sign distribution of τ on B this number can be further reduced by omitting redundant conditions. For example, the matrix S for the second base case in Figure 1 (the tetrahedral case) equals

$$S_{\text{tet}} = \begin{pmatrix} x_0 & y_0 & z_0 & 1 \\ -x_0 & -y_0 & -z_1 & -1 \\ -x_0 & -y_1 & -z_0 & -1 \\ -x_1 & -y_0 & -z_0 & -1 \end{pmatrix}.$$

The hexagonal base case (the fifth in Figure 1), has the maximum number of 8 constraints.

In order to solve this optimization problem computationally, we formulate it as a quadratic programming problem

$$\begin{aligned} &\min_{\sigma} \frac{1}{2}\sigma^{\mathsf{T}} A\sigma - b^{\mathsf{T}}\sigma \\ &\text{subject to } S\sigma \geq 0, \end{aligned} \tag{3}$$

where A and b are defined by the objective function (2). Several algorithms exist for solving quadratic programming problems. In our implementation we use the one presented in [25]. Since the size of the optimization problem is bounded, the time needed to compute the solution is bounded by a constant, independent of the specific configuration.

Figure 2 shows examples for the intersections of B with the zero-level sets of linear functions obtained by solving problem (3) for all five base cases, Figure 3 shows the approximation of a ball (which is defined by a trimming function) obtained by considering uniformly sized boxes with decreasing diameter.

4.2. Parameterization of the approximate domain

In order to find the quadrature nodes, we need to define a parameterization of the polyhedral domain $B_\sigma = \{(x, y, z) \in B : \sigma(x, y, z) \geq 0\}$. In the first, second and third base case in Figure 1, the resulting polyhedron is a cuboid, a tetrahedron or a prism[1], respectively. A parameterization can be obtained by a single, trilinear parameterization, which is degenerate for the tetrahedron

[1] More precisely, it is a pentahedron, which is topologically equivalent to a triangular prism.

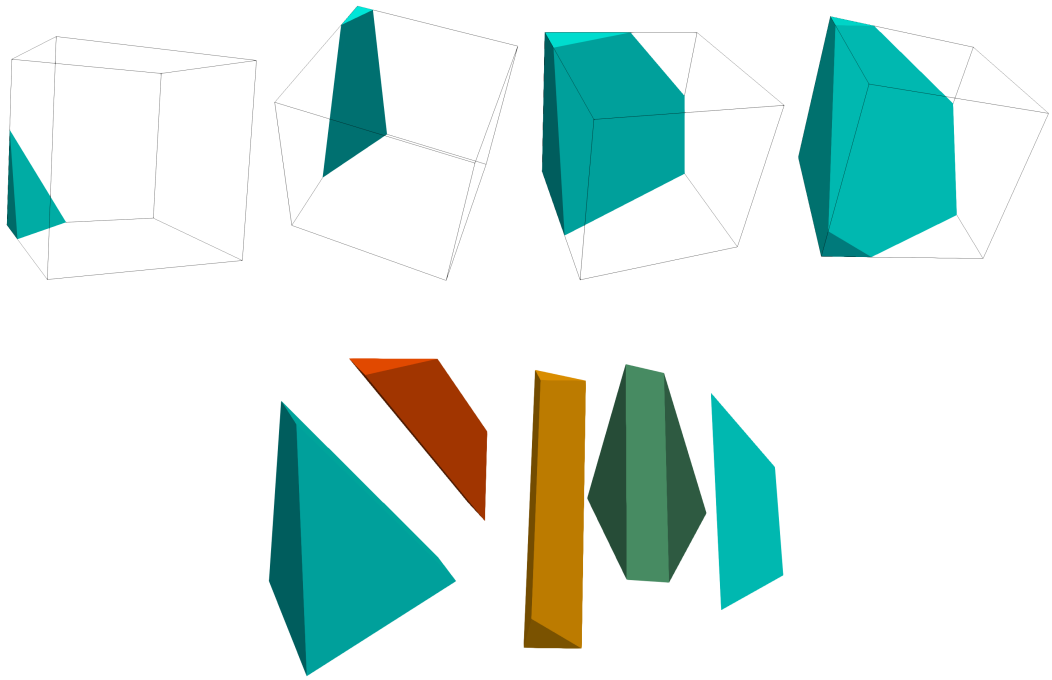

Figure 4: Parameterization of the linearized domain: Base cases 2-5 (top row) and the subdivision into trilinear patches for cases 4 and 5 (bottom row).

and the prism. For the fourth case, we create a multi-patch parameterization that consists of two degenerate trilinear patches representing prisms. In the fifth base case, we obtain three trilinear patches (two prisms and a four-sided pyramid). Instances of base cases 2-5 and the splitting into trilinear patches for cases 3 and 4 are shown in Figure 4.

If B is an instance of a rotation or a reflection of one of the base cases, the parameterization is simply composed with the respective transformation matrix. Whenever the box is in an inverted base case, i.e. one of the base cases with all signs flipped, we first compute the integral over the entire box B and then subtract the integral defined by the base case.

4.3. Numerical integration

We use a $q \times q \times q$-point Gauss rule for the approximation of the integral

$$I_\sigma f = \int_{B_\sigma} f(x, y, z) \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x. \tag{4}$$

This will be denoted as *Linearized Trimmed (LT)* quadrature rule.

If B is an instance of a base case without sign inversion, we obtain

$$\mathrm{LT}[B, \tau] f = \sum_{i=1}^{\#\text{patches}} \underbrace{\sum_{j=1}^{q^3} w_j^i f(x_j^i, y_j^i, z_j^i)}_{(\star)}, \tag{5}$$

where the nodes (x_j^i, y_j^i, z_j^i) are the images of the tensor-product Gauss nodes (ξ_j, η_j, ζ_j) for the corresponding patch G^i of the parameterization of B_σ and

$$w_j^i = w_j^{\text{Gauss}} |\det \nabla G^i(\xi_j, \eta_j, \zeta_j)|$$

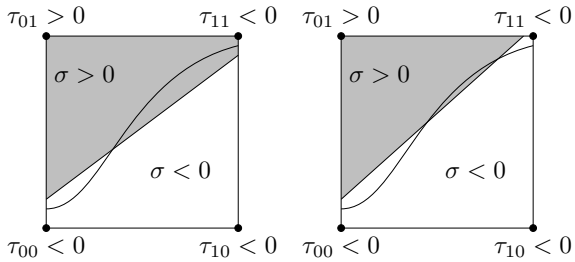

Figure 5: Two-dimensional visualization of the continuity of F. For the approximation on the left-hand-side, the function $F(u)$ is smooth. For the approximation on the right-hand-side, it is only C^0-smooth, because the boundary of the integration domain B_{η_u} crosses the north-east corner.

with the tensor-product Gauss weights w_j^{Gauss}. Otherwise, if B is an inversion of an instance of a base case (i.e. the signs of τ at the cell vertices are flipped), then the signs of the weights w_j^i are flipped and the sum ($\star$) that corresponds to the trilinear parameterization of the entire box is added to (5).

5. Corrected linearized trimmed quadrature

In the final step of our method, we add an error correction term to the cell-wise LT quadrature rule in order to alleviate the error caused by the linear approximation of the integration domain.

5.1. The first order correction term

First, we define the linear interpolant between σ and τ for $u \in [0, 1]$:

$$\eta_u(x, y, z) = \sigma(x, y, z) + u(\sigma(x, y, z) - \tau(x, y, z)). \tag{6}$$

We then consider the integral over the intermediate integration domains

$$B_{\eta_u} = \{(x, y, z) \in B : \eta_u(x, y, z) > 0\}$$

as a function of a parameter $u \in [0, 1]$, i.e.

$$F(u) = \int_{B_{\eta_u}} f(x, y, z) \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x.$$

Since σ and τ share the sign distribution at the box' vertices, this function is expected to be C^2-smooth for $u \in [0, 1]$. More precisely, it is C^2-smooth if the level sets of η_u intersect the boundary of the box transversally and do not cross any vertex or edge as u varies from 0 to 1. A two-dimensional illustration of this is shown in Figure 5.

By construction, $F(0) = I_\sigma f$ and $F(1) = I_\tau f$. In order to approximate $F(1)$, we perform a Taylor expansion of the function F around $u = 0$ and evaluate it at $u = 1$:

$$F(1) \approx F(0) + F'(0). \tag{7}$$

We call $F'(0)$ the *first order error correction term*. The right-hand side, where we use Gaussian quadrature to evaluate both contributions, defines the *Corrected Linearized Trimmed* (CLT) quadrature rule.

5.2. Evaluation of the correction term

The first order error correction term admits a simple analytic representation.

Lemma 1. *The first order error correction term is equal to*

$$F'(0) = -\int_{\{\sigma=0\}\cap B} f\frac{\tau}{\|\nabla\sigma\|}\,\mathrm{d}s. \tag{8}$$

Proof. We evaluate the derivative of the integral $F(u)$ with respect to the parameter u with the help of Reynold's transport theorem and obtain

$$F'(u) = \int_{\{\eta_u=0\}\cap B} f v_u\,\mathrm{d}s,$$

where the normal velocity v_u is given by

$$v_u = -\frac{\frac{\mathrm{d}}{\mathrm{d}u}\eta_u}{\|\nabla\eta_u\|}.$$

From the definition (6) of η_u we infer

$$\frac{\partial\eta_u}{\partial u} = \tau - \sigma$$

and

$$\nabla\eta_u = \nabla\sigma + u(\nabla\tau - \nabla\sigma).$$

Setting $u = 0$ and using that σ vanishes on the zero-level we arrive at the representation (8). □

For the numerical evaluation of the bivariate integral (8), we consider the subdivision of the integration domain $\{\sigma = 0\} \cap B$, which is defined by the patches covering the polyhedral domain B_σ, see Section 4.2. This subdivision consists of at most two triangular and quadrilateral facets, which admit bilinear parameterizations. We use a $q \times q$ point Gauss rule to evaluate the contributions of the facets to the overall integral. Consequently, the CLT quadrature rule takes the form

$$\mathrm{CLT}[B,\tau]f = \mathrm{LT}[B,\tau]f + \sum_{i=1}^{\#\text{surface patches}} \sum_{j=1}^{q^2} v_j^i f(x_j^i, y_j^i, z_j^i),$$

where (x_j^i, y_j^i, z_j^i) are the Gauss nodes mapped to the i-th planar patch using the parameterization H^i and

$$v_j^i = \frac{v_j^{\mathrm{Gauss}}\tau(x_j^i, y_j^i, z_j^i)\sqrt{\det \nabla H^{i^T}(\xi_j,\eta_j)\nabla H^i(\xi_j,\eta_j)}}{\|\nabla\sigma(x_j^i, y_j^i, z_j^i)\|_2} \tag{9}$$

with the (2D) tensor-product Gauss weights v_j^{Gauss}.

In the experiments we will observe that the optimal number of Gauss nodes in each direction when using the first order error correction term is $q = 2$. This is expected, since otherwise the overall approximation order is dominated by the linear approximation and the correction term.

Higher order correction terms, which are beyond the scope of the present paper, involve edge integrals and point evaluations. Clearly, higher order Gauss rules are needed to achieve higher overall accuracy.

5.3. Computational complexity

Next, we consider the computational complexity of both LT and CLT. We assume that each evaluation of f and τ takes constant time.

Theorem 1. *LT requires $O(q^3)$ function evaluations.*

Proof. The constrained least squares approximation is performed in constant time and thus the complexity for finding the approximations σ in all quadrature cells is $O(1)$. Thus, the complexity is dominated by the trivariate quadrature in (4) which is performed up to three times per element. Since we use a $q \times q \times q$-point Gauss rule, the overall complexity is $O(q^3)$. □

Theorem 2. *The computational complexity of CLT is the same as the computational complexity of LT.*

Proof. The first order correction term (8) consists of a bivariate integral which is approximated using the same amount of Gauss nodes per direction as for the trivariate integral (4). Since the number of function evaluations in the computation of the weights (9) is constant in h, the overall complexity is governed by the trivariate integral in LT and the complexities are asymptotically the same. □

The overall effort of the compound quadrature rule (which sums up the contributions of the N_h sub-boxes, see Section 3) thus evaluates to $O(N_h q^3)$.

6. Numerical examples

We implemented the method using the open source C++ library G+Smo [26]. For solving the quadratic programming problem (3) we use QuadProg++ [27].

6.1. Convergence of the quadrature

In this section, we analyze the behavior of our method with respect to h-refinement. This means that we first subdivide a given geometry uniformly into boxes of size h and then apply different quadrature rules to them.

Ellipsoid example. In our first numerical example we use our method to compute the volume of the region enclosed by an ellipsoid given implicitly by

$$\tau(x, y, z) = -\frac{(x-0.5)^2}{a^2} - \frac{(y-0.5)^2}{b^2} - \frac{(z-0.5)^2}{c^2} + 1.$$

Its piece-wise linear approximation after five steps of refinement is shown in Figure 6. We compare the convergence of LT and CLT when using two Gauss nodes per direction in both the trivariate integral in LT and the bivariate integral in the first order error correction term for CLT. Additionally, we compare the two methods with simply using Gauss quadrature for all cells that lie completely inside the integration domain. This method will be denoted as the inner cell (IC) method.

In the plot shown in Figure 7 we observe that the use of error correction increases the order of convergence by one, resulting in cubic convergence. On the other hand, simply omitting all trimmed cells gives only linear convergence.

Section 5.3 showed that the computational complexity of both LT and CLT is linear with respect to the number of quadrature cells. This is the same complexity as for the usual Gauss quadrature over untrimmed cells. Figure 8 shows the computation times needed by our implementation of LT, CLT and IC for various values of the mesh size h. We observe asymptotically equivalent computation times of all three methods. The additional computational effort for CLT, which is required for the advanced treatment of the trimmed cells, is more than justified by the increased accuracy.

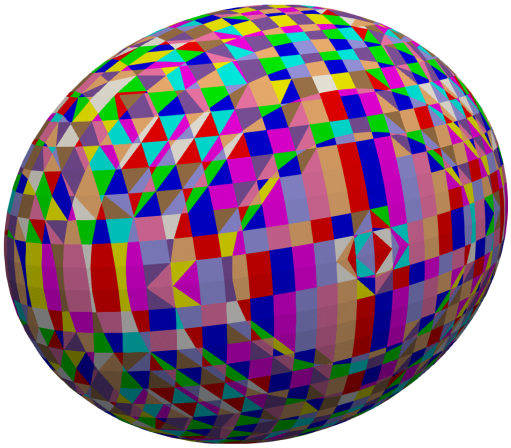

Figure 6: Piece-wise linear approximation of the ellipsoid after five steps of refinement

Torus example. In our next example, we compute the volume of a torus given implicitly by the trimming function

$$\tau(x,y,z) = -((x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2 + R - r)^2 + 4R((x-0.5)^2 + (y-0.5)^2).$$

Figure 9 shows the result of the approximation of τ by linear functions in each cell after five steps of refinement. As in the previous example, we employ a $2\times2\times2$-point Gauss rule for the trivariate quadrature in LT and a 2×2-point Gauss rule for the bivariate quadrature in the correction term for CLT. Figure 10 depicts the quadrature error for various values of h for LT, CLT, and IC. This example confirms that the previous observations also apply to non-convex (and hence more complicated) domains.

6.2. L^2 fitting and isogeometric analysis

We apply our method to the problems of L^2-projection onto the trimmed B-Spline basis and Galerkin projection of the Poisson problem on a trimmed domain $\Omega_\tau \subset \mathbb{R}^3$, where the trimming surface is given implicitly by a function $\tau : \Omega \to \mathbb{R}$.

In addition to the numerical quadrature, another important challenge in the numerical treatment of trimmed domains is the stability of the basis. If we used the trimmed B-Spline basis

$$\tilde{B}_i(x,y,z) = \begin{cases} \hat{B}_i(x,y,z) & \text{if } \tau(x,y,z) > 0 \\ 0 & \text{else} \end{cases} \tag{10}$$

for isogeometric analysis or for L^2 projection, then the supports $\operatorname{supp}\tilde{B}_i \cap \Omega_\tau$ might be arbitrarily small, depending on how the trimming surface cuts each element of the discretization. In order to address the loss of stability caused by this fact, we use the extended B-Spline basis [20].

Basis functions whose support is smaller than a given threshold (e.g. less than an entire element) are eliminated and added to $(p+1)^d$ stable basis functions with appropriate weights, thereby maintaining the property of polynomial reproduction. This is usually implemented as a post-processing step of the full system matrix. More precisely, The stable system matrix $\hat{A}$ and right-hand-side $\hat{b}$ are given by

$$\hat{A} = EAE^T, \qquad \hat{b} = Eb,$$

where the rectangular matrix E contains the extension weights (for details see [20]) and A and b are the system matrix and right-hand-side related to the trimmed B-splines (10).

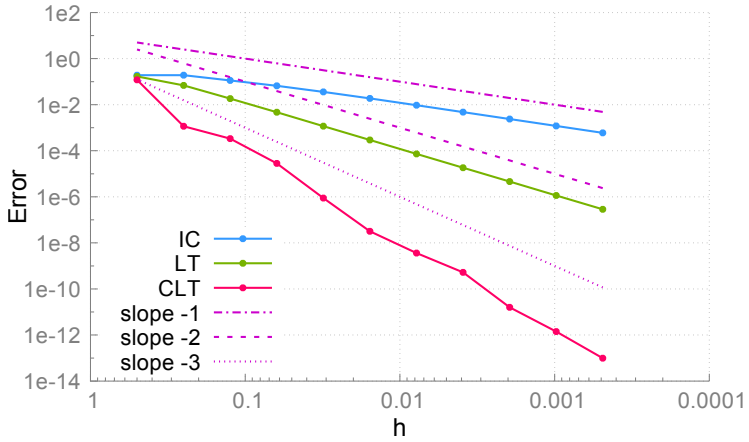

Figure 7: Approximation of the volume enclosed by an ellipsoid using the LT and CLT.

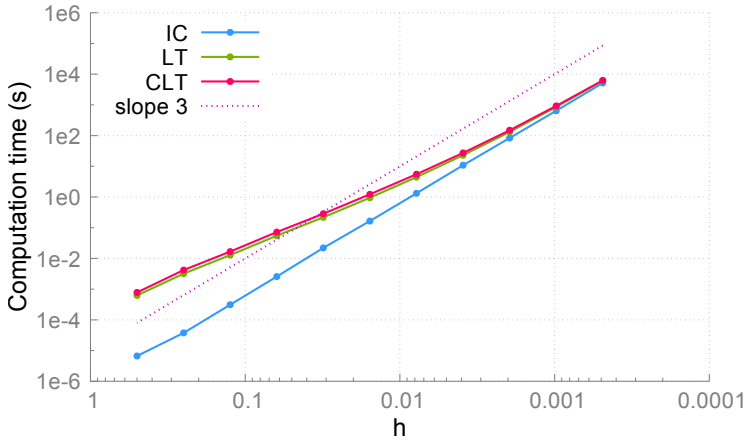

Figure 8: Computation times for computing the volume enclosed by the ellipsoid

Example: L^2-projection. We apply our quadrature method to the assembly of the mass matrix in order to perform an L^2-projection of the function

$$u(x, y, z) = \sin(\pi xyz) + \cos(2\pi(y + z)) \tag{11}$$

into the space of trimmed splines on the unit cube trimmed with a ball of radius $r = 0.23$ around one of the vertices. Its trimming function is given by

$$\tau(x, y, z) = x^2 + y^2 + z^2 - 0.23^2.$$

The solution is shown in Figure 11.

We use extended B-splines in order to arrive at a stable discretization. Figure 12 shows the h-dependence of the L^2-error for several polynomial degrees p. We achieve the optimal rates of convergence.

However, it should be noted that one obtains similar results when using LT or even IC. Indeed, the resulting L^2 approximation is not very sensitive with respect to perturbations of the domain boundary.

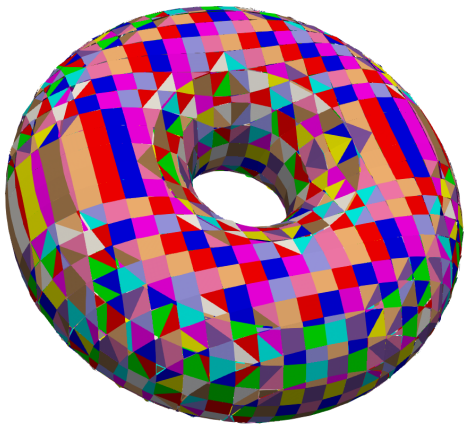

Figure 9: Linear approximation of the torus after some steps of refinement

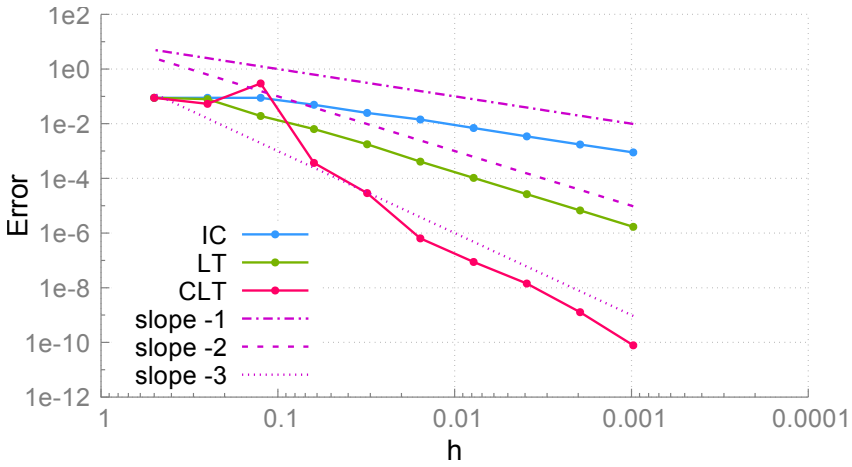

Figure 10: Approximation of the volume of the torus using the LT and CLT.

Example: Poisson equation on cube with spherical cavity. Next, we use our quadrature rules to assemble the stiffness matrix in order to solve the Poisson equation

$$\begin{cases} \Delta u &= f \text{ in } \Omega_\tau, \\ \frac{\partial u}{\partial \nu} &= h \text{ on } \partial\Omega_\tau^{\text{Neumann}}, \\ u &= g \text{ on } \partial\Omega_\tau^{\text{Dirichlet}} \end{cases} \tag{12}$$

using Galerkin projection, based on extended B-splines.

We use Gauss quadrature with $(p+1)^3$ nodes for the element-wise quadrature on the untrimmed elements and the linear approximations to the trimmed elements. We also use $(p+1)^2$ Gauss nodes for the bivariate integral in the error correction term for CLT.

The domain is the unit cube, which has been trimmed by a ball of radius $r = 0.23$ around the center, thus defining the trimming function

$$\tau(x, y, z) = (x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 - 0.23^2.$$

We impose Dirichlet boundary conditions on the boundary of the cube and homogeneous Neumann boundary conditions on the spherical trimming surface. As a manufactured solution fulfilling these

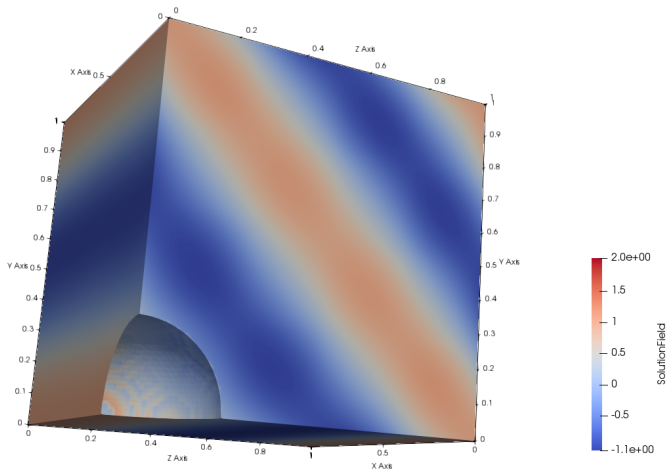

Figure 11: Exact solution (11) on the trimmed cube.

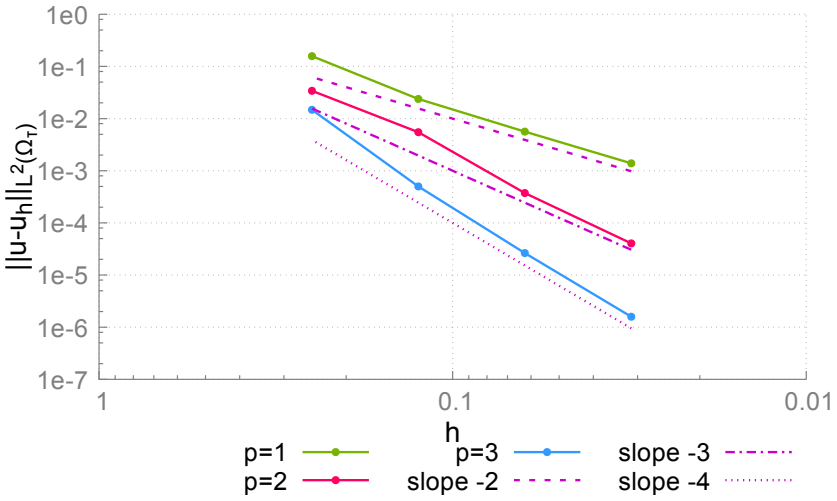

Figure 12: L^2-projection, stabilized using extended B-splines

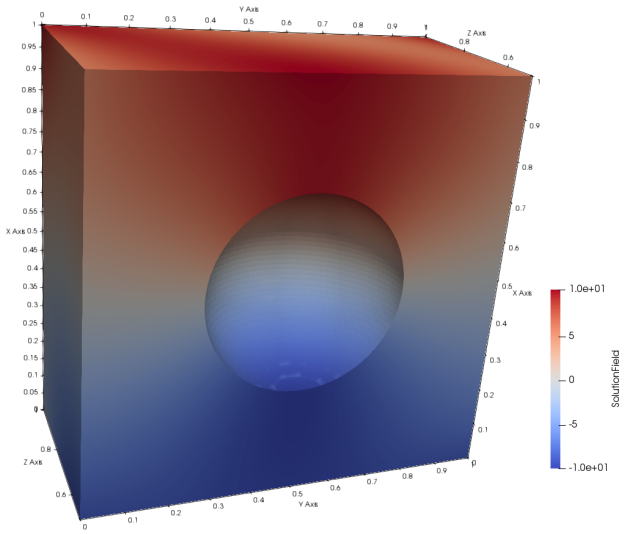

Figure 13: The example solution (13) on the trimmed domain clipped with a plane through the origin (only for the visualization).

boundary conditions we choose

$$u(x,y,z) = \frac{10(x-\frac{1}{2})}{\sqrt{(x-\frac{1}{2})^2+(y-\frac{1}{2})^2+(z-\frac{1}{2})^2}} + (\tau(x,y,z))^2(8\cos(2\pi(y+z)) + 5\sin(2\pi xy)). \tag{13}$$

The first term of u is constant on the lines through the center of the domain, thus fulfilling homogeneous Neumann boundary conditions on every sphere. The second term as well as its normal derivative vanishes on the trimming surface, which is given as the level set $\tau = 0$. Figure 13 shows the manufactured solution on the domain $\Omega\tau$ which was clipped by a plane through its center for the visualization.

First we consider the error measured in the H^1-seminorm, see Fig. 14, which can be analyzed with the help of Strang's lemma [28]. It is bounded by the sum of the approximation error (caused by the discretization) and the consistency error (due to the numerical integration).

For quadratic spline discretizations ($p = 2$), the approximation error (with respect to the H^1-seminorm) converges with order two, and hence it suffices to use a quadrature method that provides this level of accuracy. Nevertheless, the CLT provides a small improvement, compared to LT.

For cubic spline discretizations ($p = 3$), the approximation error (with respect to the H^1-seminorm) converges with order three, and hence one needs to employ quadrature method that provides the same accuracy. The plot clearly demonstrates that CLT maintains the overall accuracy, while LT does not.

Second we consider the error measured in the L^2-norm, see Fig. 15. For quadratic spline discretizations ($p = 2$), we obtain the full rate of convergence (order 3) when using CLT. However, the use of LT does not provide the full rate, even though it did suffice for optimal (quadratic) H^1 seminorm convergence.

The situation is different for cubic splines ($p = 3$): Here, CLT gives only a (sub-optimal) cubic convergence rate, and an even lower accuracy is obtained when using LT. We conclude that the accuracy of the quadrature does not suffice to extend the optimal rate of convergence with respect to the H^1 seminorm (which was guaranteed by Strang's lemma and demonstrated in Fig. 14) to the L^2 norm. A similar phenomenon was observed in [29], where a higher accuracy of the

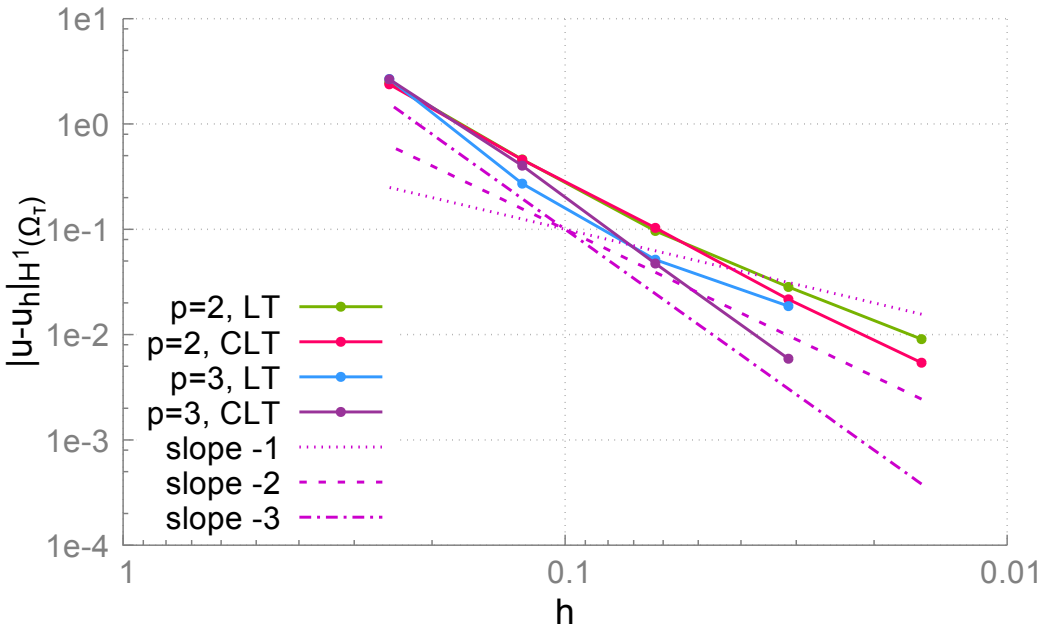

Figure 14: h-dependence of the error of the Galerkin approximation in the H^1-seminorm.

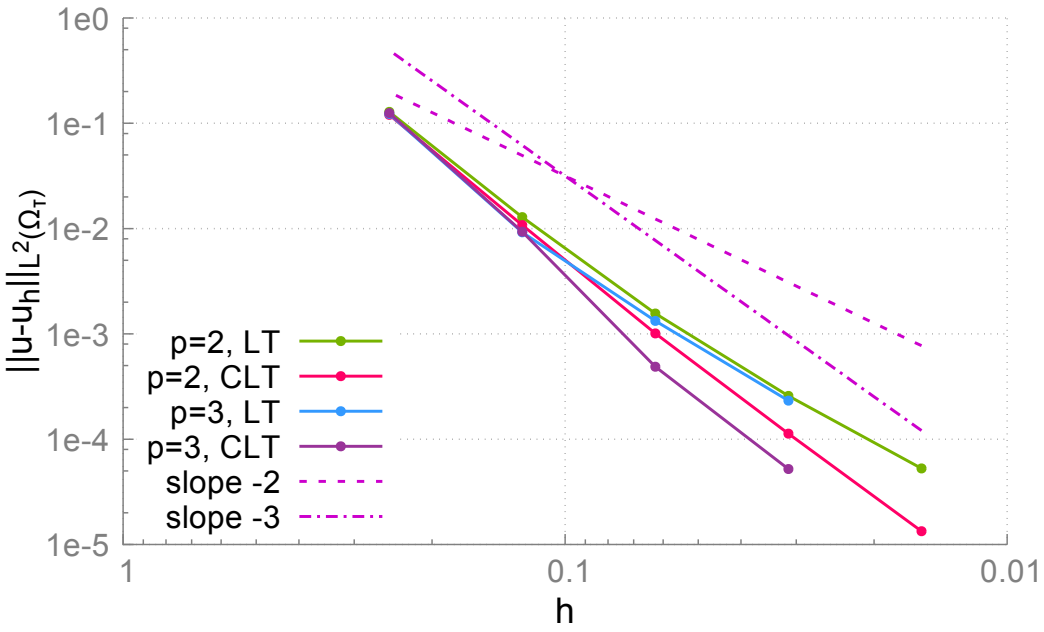

Figure 15: h-dependence of the error of the Galerkin approximation in the L^2-norm.

quadrature was required for even degree splines in order to achieve the optimal rate of convergence with respect to the L^2 norm.

We summarize the experimentally observed rates of convergence with respect to the H^1 seminorm and the L^2 norm (separated by /) in the following table:

degree	LT	CLT
$p = 2$	$\mathbf{2}/2$	$\mathbf{2}/\mathbf{3}$
$p = 3$	$2/2$	$\mathbf{3}/3$

Optimal rates are shown in bold. We conclude that CLT is ideally suited for quadratic discretizations, and beneficial for cubic ones.

In Figure 16, we see that, as is expected, the error appears predominantly on the trimming surface, which is where most of the consistency error occurs. However, for linear and quadratic spline discretizations, the error is more evenly distributed than for the cubic discretization. Again, this indicates that the accuracy of the trimmed quadrature – which is used near the trimming surface only – is still insufficient to obtain the optimal rate of convergence for cubic spline discretizations.

7. Conclusion and future work

We presented an efficient and stable method for the numerical quadrature on trimmed volumes and its application to isogeometric analysis with homogeneous Neumann boundary condition on the trimming surface. Dirichlet boundary conditions on trimming surfaces are usually imposed weakly, for example using Nitsche's method. To this end, an integral term over the boundary is added to the bilinear form. Moreover, boundary integrals appear naturally when imposing non-homogeneous Neumann boundary conditions. This makes it necessary to perform numerical quadrature on the trimming surface. An open task is therefore to apply the presented method to integration on level-set surfaces.

In order to apply the method to higher-order discretizations, it is natural to add further terms to the Taylor expansion (7). Our conjecture is that each term results in an additional order of convergence of the quadrature error.

Acknowledgment. The authors gratefully acknowledge the support provided by the Austrian Science Fund (FWF) through project NFN S11708, and by the ERC Advanced Grant "CHANGE", GA no. 694515.

References

[1] J. A. Cotrell, T. Hughes, Y. Bazilevs, Isogeometric Analysis, Toward Integration of CAD and FEA, John Wiley and Sons, 2009.

[2] A. Buffa, G. Sangalli (Eds.), IsoGeometric Analysis: A New Paradigm in the Numerical Approximation of PDEs, Springer International Publishing, 2016.

[3] S. P. A. Bordas, E. N. Burman, M. G. Larson, M. A. Olshanskii (Eds.), Geometrically Unfitted Finite Element Methods and Applications, Lecture Notes in Computational Science and Engineering, Springer International Publishing, 2017.

[4] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, A. Düster, Geometric modeling, isogeometric analysis and the finite cell method, Computer Methods in Applied Mechanics and Engineering 249-252 (2012) 104–115.

[5] B. Marussig, T. Hughes, A review of trimming in isogeometric analysis: Challenges, data exchange and simulation aspects, Archives of Computational Methods in Engineering (2017) 1–69.

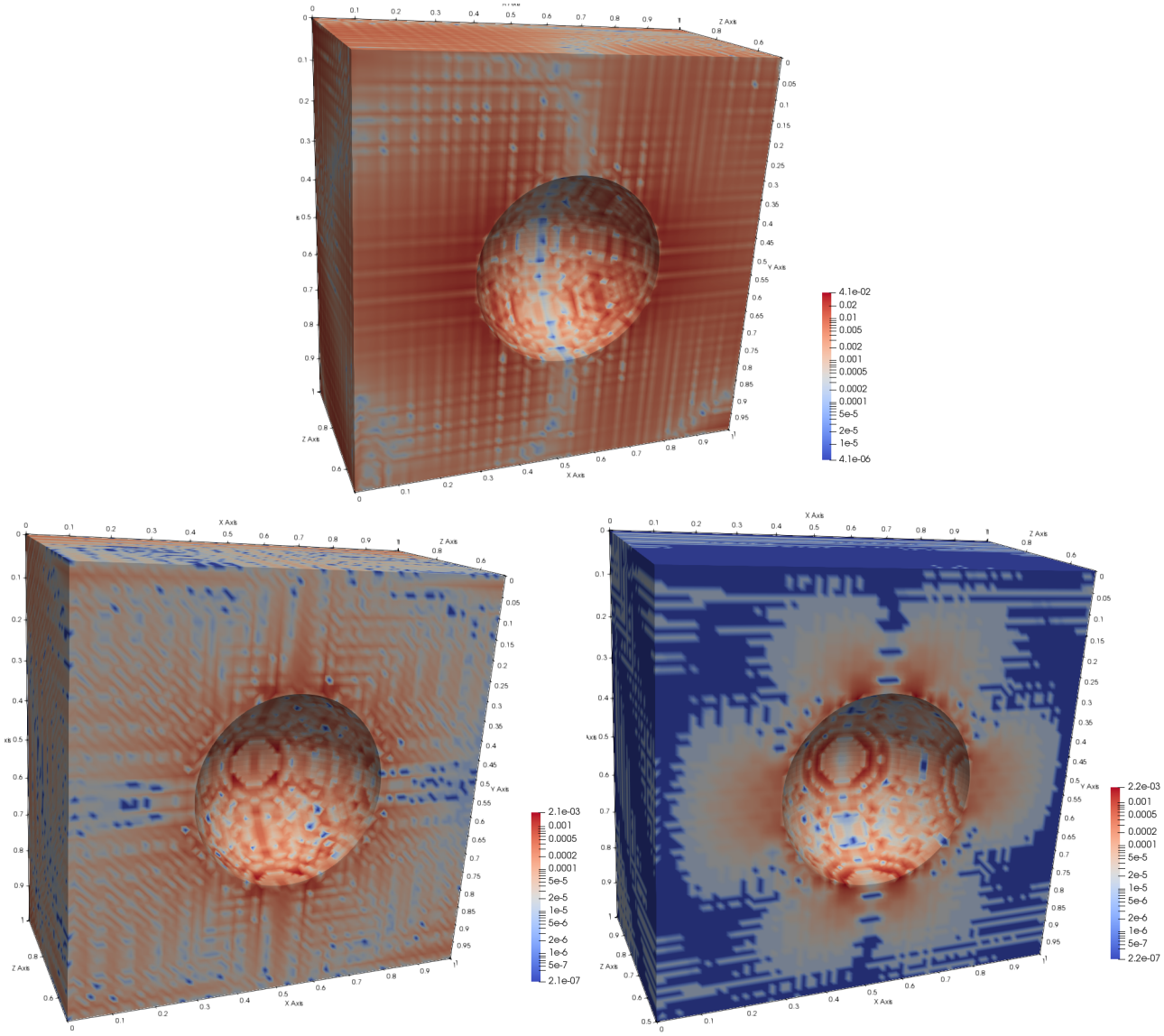

Figure 16: The absolute error $|u(x,y,z) - u_h(x,y,z)|$ for $p = 1$ (top), $p = 2$ (left) and $p = 3$ (right) after five steps of refinement using CLT for the assembly. Note that the scales are different!

[6] P. Antolin, A. Buffa, M. Martinelli, Isogeometric Analysis on V-reps: first results (2019). arXiv:1903.03362.

[7] C.-K. Im, S.-K. Youn, The generation of 3d trimmed elements for nurbs-based isogeometric analysis, International Journal of Computational Methods 15 (7).

[8] S. Xia, X. Qian, Isogeometric analysis with Bézier tetrahedra, Computer Methods in Applied Mechanics and Engineering 316 (2017) 782 – 816.

[9] F. Massarwi, P. Antolin, G. Elber, Volumetric untrimming: Precise decomposition of trimmed trivariates into tensor products, Computer Aided Geometric Design 71 (2019) 1 – 15.

[10] N. Patrikalakis, T. Maekawa, Shape interrogation for computer aided design and manufacturing, 2010.

[11] C.-Y. Hu, T. Maekawa, N. Patrikalakis, X. Ye, Robust interval algorithm for surface intersections, CAD Computer Aided Design 29 (9) (1997) 617–627.

[12] L. Kudela, N. Zander, S. Kollmannsberger, E. Rank, Smart octrees: Accurately integrating discontinuous functions in 3d, Computer Methods in Applied Mechanics and Engineering 306 (2016) 406 – 426.

[13] A. Nagy, D. Benson, On the numerical integration of trimmed isogeometric elements, Computer Methods in Applied Mechanics and Engineering 284 (2015) 165–185.

[14] H. Zhang, R. Mo, N. Wan, An IGA discontinuous Galerkin method on the union of overlapped patches, Computer Methods in Applied Mechanics and Engineering 326 (2017) 446–480.

[15] S. Kargaran, B. Jüttler, S. K. Kleiss, A. Mantzaflaris, T. Takacs, Overlapping multi-patch structures in isogeometric analysis, NFN Technical Report 82 (2019) 1–34.

[16] M. Bercovier, I. Soloveichik, Overlapping non matching meshes domain decomposition method in isogeometric analysis (2015). arXiv:1502.03756.

[17] F. Scholz, A. Mantzaflaris, B. Jüttler, First order error correction for trimmed quadrature in isogeometric analysis, in: Advanced Finite Element Methods with Applications, Lecture Notes in Computational Science and Engineering, Springer International Publishing, 2019, to appear, also available at `http://www.gs.jku.at` as NFN Report no. 84.

[18] D. A. Cox, T. W. Sederberg, F. Chen, The moving line ideal basis of planar rational curves, Computer Aided Geometric Design 15 (8) (1998) 803 – 827.

[19] G. Taubin, Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (11) (1991) 1115–1138.

[20] K. Höllig, U. Reif, J. Wipper, Weighted extended B-spline approximation of Dirichlet problems, SIAM Journal on Numerical Analysis 39 (2) (2001) 442–462.

[21] B. Marussig, J. Zechner, G. Beer, T. Fries, Stable isogeometric analysis of trimmed geometries, Computer Methods in Applied Mechanics and Engineering 316 (2017) 497–521.

[22] R. Sanches, P. Bornemann, F. Cirak, Immersed b-spline (i-spline) finite element method for geometrically complex domains, Computer Methods in Applied Mechanics and Engineering 200 (13) (2011) 1432 – 1445.

[23] E. Burman, Ghost penalty, Comptes Rendus Mathematique 348 (21) (2010) 1217 – 1220.

[24] D. Elfverson, M. G. Larson, K. Larsson, CutIGA with basis function removal, Advanced Modeling and Simulation in Engineering Sciences 5 (1) (2018) 6.

[25] D. Goldfarb, A. Idnani, A numerically stable dual method for solving strictly convex quadratic programs, Mathematical Programming 27 (1983) 1–33.

[26] A. Mantzaflaris, F. Scholz, others (see website), G+Smo (Geometry plus Simulation modules) v0.8.1, http://gs.jku.at/gismo (2019).

[27] L. Di Gaspero, Quadprog++, https://github.com/liuq/QuadProgpp.

[28] G. Strang, Approximation in the finite element method, Numerische Mathematik 19 (1972) 81–98.

[29] A. Mantzaflaris, B. Jüttler, Integration by interpolation and look-up for Galerkin-based isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 284 (2015) 373–400.