Highlights

A data-driven approach to cut-cell quadrature using spline interpolation

Michał Ł. Mika, René R. Hiemstra, Stein K.F. Stoter, Dominik Schillinger

- We present a data-driven approach to tensor-product quadrature rules.
- It enables higher-order accurate integration of cut elements based on implicitly defined domains.
- The quadrature rules are compatible with existing cut finite element techniques.
- We demonstrate the effectiveness of the approach via a series of embedded domain benchmarks.

A data-driven approach to cut-cell quadrature using spline interpolation

Michał Ł. Mika[a,*], René R. Hiemstra[b], Stein K.F. Stoter[b] and Dominik Schillinger[a]

[a] *Institute for Mechanics, Technical University of Darmstadt, Franziska-Braun-Straße 7, Darmstadt, 64287, Germany*

[b] *Department of Mechanical Engineering, Eindhoven University of Technology, Groene Loper 15, Eindhoven, 5612 AE, The Netherlands*

ARTICLE INFO

Keywords:
cut finite element methods, immersed boundary, cut-cell integration, spline interpolation, data-driven quadrature, tensor-product splines

ABSTRACT

This paper presents a data-driven approach to develop higher-order accurate tensor-product-stencil quadrature rules for implicitly defined two-dimensional domains. We construct a three-dimensional configuration space of possible domain cuts using a signed distance representation based on circular arcs, defined by their radius and center, and exploit symmetry to simplify its three-dimensional domain. The configuration space, being piecewise smooth, is carefully partitioned into smooth regions, enabling three-dimensional tensor-product spline interpolation to approximate quadrature data sampled from an established implicit domain quadrature technique. The resulting quadrature rules are simple to apply, highly accurate, efficient, and can be used as a black-box solution. We demonstrate compatibility with existing cut finite element techniques and illustrate their application through numerical examples.

1. Introduction

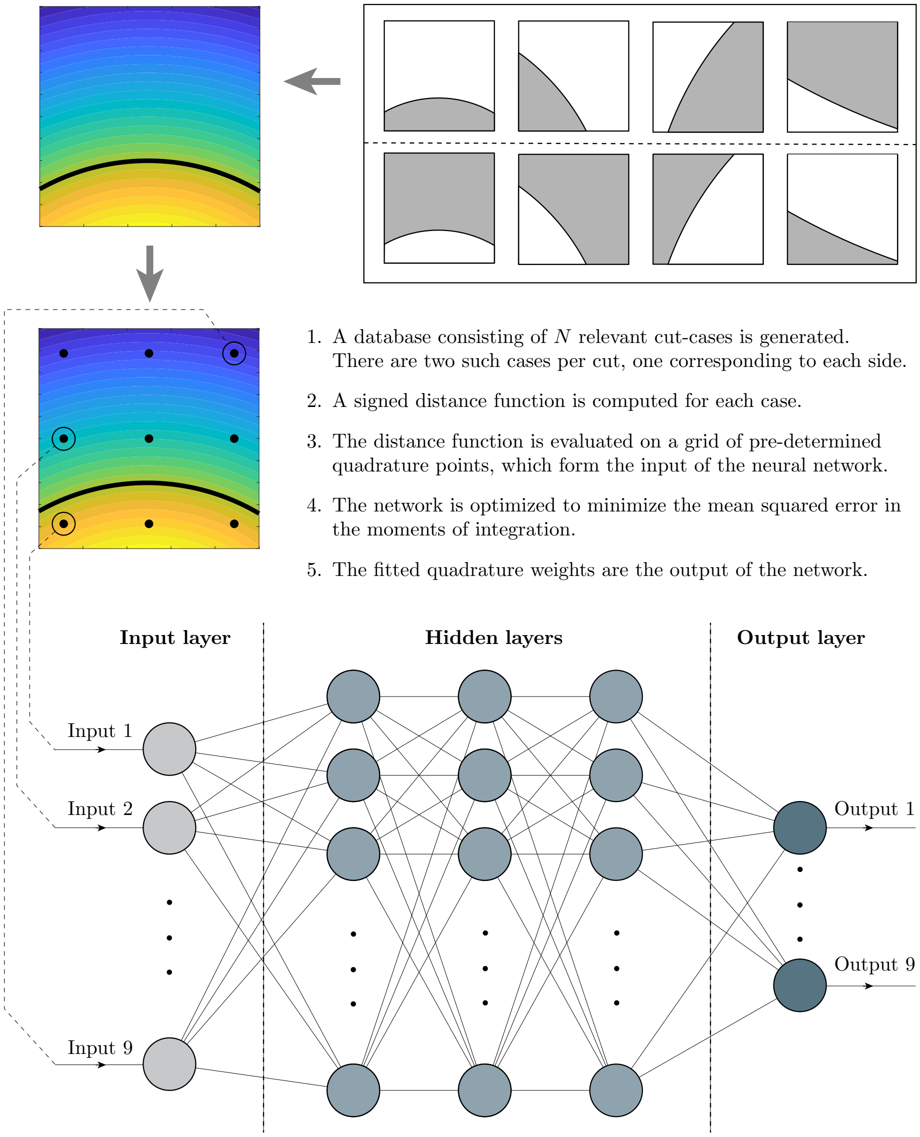
The finite element method is well-established for solving partial differential equations. However, generating boundary-fitted meshes may take up to 90% of analysis time in commercial applications [18], which has motivated the development of isogeometric analysis and immersed boundary finite element methods. Cut finite element methods – also known as embedded or fictitious domain, immersed boundary, or unfitted finite element methods – approximate solutions on non-boundary-fitted meshes [13, 51, 57]. They eliminate labor-intensive mesh generation, but introduce new challenges. One main challenge is numerical quadrature of weak-form integrals over cut cells, where domain discontinuities preclude standard quadrature techniques [11, 23, 24, 26, 27, 33, 36, 37, 45, 48, 54, 59, 62]. Other main challenges are the treatment of small cut cells that can cause ill-conditioning and discretization instabilities [4, 12, 14, 15, 17, 21, 22, 32, 42, 65], and enforcing boundary and interface conditions on immersed boundaries [1, 20, 25, 46, 47, 52, 53, 56]. Cut finite element methods have been successfully applied to handle very large deformations [58] and contact [6, 35, 50], topology optimization [49, 64], tackle trimming in isogeometric analysis [8, 29, 39, 60], and have been implemented efficiently and robustly for large-scale computations [3, 5, 34, 41, 43, 55]. Comprehensive reviews of cut finite element methods can be found for example in [13, 16, 57].

In this work, we propose a novel data-driven quadrature approach for immersed boundary methods, assuming that smooth, curved geometries can be approximated up to second-order accuracy by circular arcs on sufficiently fine meshes. We construct a three-dimensional configuration space of possible cuts on a unit square reference cell characterized by the cell's centroid $\mathbf{C}$. We use a signed distance representation of circular arcs defined by a radius $R \in [1, R_{\max}]$ to describe the cuts. By exploiting symmetry, we restrict the angular position to $\Theta \in [0, \pi/4]$ and the distance from the centroid to the cut to $Z \in [R - d/2, R + d/2]$, where d is the cell's diagonal length. The configuration space $R \times \Theta \times Z$ is piecewise smooth, with C^1 continuity along certain cross-sections. We carefully partition it into smooth regions, enabling three-dimensional tensor-product spline interpolation of sampled quadrature data computed using Algoim [54] which in turn enables construction of accurate moment-fitted quadrature rules [11, 28, 33, 38, 40, 45, 61]. By leveraging symmetry, the resulting spline interpolants of the moments enable high precision black-box evaluation of integrals over cut cells.

[*] Corresponding author

michal.mika@tu-darmstadt.de (M.Ł. Mika); r.r.hiemstra@tue.nl (R.R. Hiemstra); k.f.s.stoter@tue.nl (S.K.F. Stoter); dominik.schillinger@tu-darmstadt.de (D. Schillinger)

ORCID(s): 0000-0003-3222-9179 (M.Ł. Mika); 0000-0002-0179-9606 (R.R. Hiemstra); 0000-0002-0179-9606 (S.K.F. Stoter); 0000-0002-9068-6311 (D. Schillinger)



1. A database consisting of N relevant cut-cases is generated. There are two such cases per cut, one corresponding to each side.
2. A signed distance function is computed for each case.
3. The distance function is evaluated on a grid of pre-determined quadrature points, which form the input of the neural network.
4. The network is optimized to minimize the mean squared error in the moments of integration.
5. The fitted quadrature weights are the output of the network.

Figure 1: Schematic of our initial neural network approach for cut-cell quadrature [30]. The accuracy of the neural network approach is limited by (1) the convergence of stochastic gradient descent, typically $\mathcal{O}(10^{-8})$–$\mathcal{O}(10^{-6})$ in mean squared error, and (2) the need to account for the smoothness of the configuration space.

This spline-based interpolation approach was motivated by our early research into learning techniques for cut-cell integration, beginning in 2021 [30]. We initially employed small neural networks with 3-5 layers, as illustrated in Figure 1, where a database of N relevant cut cases (two per cut, one for each side) was generated, signed distance functions computed, and evaluated on a grid of pre-determined quadrature points (e.g., 9 points) to form network inputs. A feed-forward neural network with hidden layers was optimized to minimize the mean squared error in the integration moments, outputting fitted quadrature weights (e.g., 9 weights). However, we could not achieve the required accuracy in cut-cell integration due to two primary limitations: (1) stochastic gradient descent typically achieves a maximum accuracy of $\mathcal{O}(10^{-8})$ to $\mathcal{O}(10^{-6})$ in the mean squared error loss function, resulting in quadrature errors of $\mathcal{O}(10^{-4})$ to $\mathcal{O}(10^{-3})$; (2) attaining high accuracy necessitates carefully accounting for the smoothness of the configuration space of cuts, as illustrated in Figure 2. Similar limitations have been observed in recent work on neurally integrated finite elements for evolving domains [19], reporting integration errors of $\mathcal{O}(10^{-3})$. Further experiments with residual-based multi-stage neural networks [63] reduced the training error by two orders of magnitude, but led to overfitting and

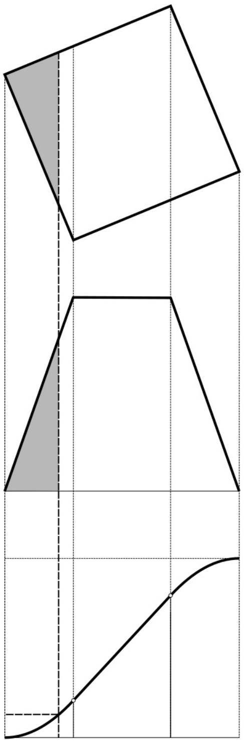


Figure 2: The integrand of the constant function over the square is a C^1 piecewise smooth quadratic function.

poor generalization to test data. In contrast to these observations, our data-driven approach presented in the following provides orders-of-magnitude greater accuracy, which is crucial for practical applicability in finite element methods. It is based on a simpler spline interpolation methodology, which explicitly accounts for reductions in continuity and can yield accuracies in the order of $\mathcal{O}(10^{-12})$ to $\mathcal{O}(10^{-8})$.

The remainder of this paper is organized as follows. Section 2 provides some background on implicit geometry modeling and moment-fitted quadrature. Section 3 discusses the parameterization of the configuration space of cuts and the generation of moment-fitting data for quadrilateral elements in two dimensions. Section 4 details the spline interpolation of the moment-fitting data. Section 5 presents numerical experiments displaying stability and accuracy for two-dimensional benchmark problems.

2. Moment-fitted quadrature on implicit domains

In this section, we review implicit geometry representation based on a signed distance function. We then summarize some relevant background on numerical quadrature that is used in the sequel of this paper. Given a target space, we introduce the equations that are needed for exact quadrature. Application of these equations leads to moment-fitted quadrature rules. The concepts introduced in this section are later used in the design of the spline interpolant.

2.1. Implicit geometry description using a signed distance function

An implicit domain is defined as a set of points in Euclidean space that satisfies an equation $f(\boldsymbol{x}) = 0$ with $\boldsymbol{x} \in \mathbb{R}^d$. Because the equation is never explicitly solved for $\boldsymbol{x}$, the representation is called implicit. If the function is represented in terms of polynomials in $x_k,\ k = 1, ..., d$, then the surface is called algebraic. In this work, we consider a level-set representation of geometry based on the signed distance function. However, we note that the proposed method does not necessarily depend on the distance property of signed distance functions.

Let $\Omega \in \mathbb{R}^d$ $(d = 2, 3)$ denote the physical domain with boundary $\partial\Omega$. A signed distance function is defined as

$$\psi(\boldsymbol{x}) = \begin{cases} -\mathrm{dist}(\boldsymbol{x}, \partial\Omega) & \text{for } \boldsymbol{x} \in \Omega \\ +\mathrm{dist}(\boldsymbol{x}, \partial\Omega) & \text{for } \boldsymbol{x} \notin \Omega \end{cases} \tag{1}$$

where $\mathrm{dist}(\boldsymbol{x}, \partial\Omega)$ denotes the shortest Euclidean distance between a point $\boldsymbol{x} \in \mathbb{R}^d$ and the boundary $\partial\Omega$. The boundary $\partial\Omega$ corresponds to the zero contour of $\psi(\cdot)$. The signed distance function is negative when $\boldsymbol{x}$ is inside of Ω and positive when it is outside of Ω.

2.2. Moment-fitted quadrature

By means of *moment fitting* it is possible to attain higher-order accurate tensor product quadrature rules for implicitly defined domains Ω [45]. Typically, these moment-fitted quadrature rules attain high accuracy with relatively few quadrature points. In the following, we briefly review numerical quadrature and discuss the conditions for *exact integration of moments* in a specified *target space*.

Numerical quadrature

A quadrature rule is a numerical approximation to a definite integral of a function $\phi(\boldsymbol{x})$, usually in terms of a weighted linear combination of function evaluations at specified points in the domain of integration. The rule thus corresponds to a set of quadrature points and weights, $(\boldsymbol{x}_j, w_j),\ j = 1, \ldots, m$, such that the following quadrature formula,

$$\sum_{j=1}^{m} w_j \phi(\boldsymbol{x}_j) \approx \int_{\Omega} \phi(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}, \tag{2}$$

is sufficiently accurate for all functions ϕ in some predetermined function space $\mathcal{V}$.

Conditions for exact integration of moments

Typically, the formula is required to be exact for an n-dimensional linear subspace $\mathcal{V}^n \subset \mathcal{V}$, called the target space for integration. If $\phi_\alpha,\ \alpha = 1, \ldots, n$, is a basis for the target space $\mathcal{V}^n$, this leads to the following *exactness* conditions,

$$\underbrace{\begin{pmatrix} \phi_1(\boldsymbol{x}_1) & \dots & \phi_1(\boldsymbol{x}_m) \\ \vdots & \ddots & \vdots \\ \phi_n(\boldsymbol{x}_1) & \dots & \phi_n(\boldsymbol{x}_m) \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix}}_{\mathbf{w}} = \underbrace{\begin{pmatrix} \int_\Omega \phi_1\, \mathrm{d}\boldsymbol{x} \\ \vdots \\ \int_\Omega \phi_n\, \mathrm{d}\boldsymbol{x} \end{pmatrix}}_{\mathbf{y}}. \tag{3}$$

Here, the matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ is formed by evaluating the basis at the quadrature points, the vector $\mathbf{w} \in \mathbb{R}^m$ collects the quadrature weights, and $\mathbf{y} \in \mathbb{R}^n$ collects the moments. An important observation is that the quadrature formula has $2m$ free parameters, m weights and m points, so we can hope to integrate exactly all functions in a target space of dimension $n = 2m$. For example, an m-point Gauss-Legendre rule integrates all polynomials x^k, $k = 0, 1, \ldots, 2m - 1$, exactly, and is optimal in the sense that there exists no other rule that exactly integrates all polynomials with fewer quadrature points. For general target spaces $\mathcal{V}^n$, such optimal rules are called generalized Gaussian quadratures [40]. The equations in (3) are highly non-linear in the location of the points $\boldsymbol{x}_i$, which makes simultaneous optimization of the points and weights a challenging problem [9]. Of specific relevance to this paper is the work in [45], where the above exactness conditions are used in the context of surface and volume integration on implicitly defined domains.

Remark 1. The moments in Equation (3) can be evaluated to high accuracy using higher-order accurate quadrature rules generated by Algoim [54]. The method converts a level-set function that describes the integration domain into an implicitly defined height function. It applies one-dimensional Gaussian quadrature in a suitable height function direction estimated by one-dimensional root finding, which leaves an integral in one fewer spatial dimensions. Repeating this procedure recursively down to the one-dimensional case yields higher-order accurate quadrature rules with strictly positive weights. On the left of Figure 3, we show three cut cells with quadrature rules obtained via Algoim. We note that our approach does not depend on Algoim, but any sufficiently accurate quadrature method can be used to evaluate the moments.

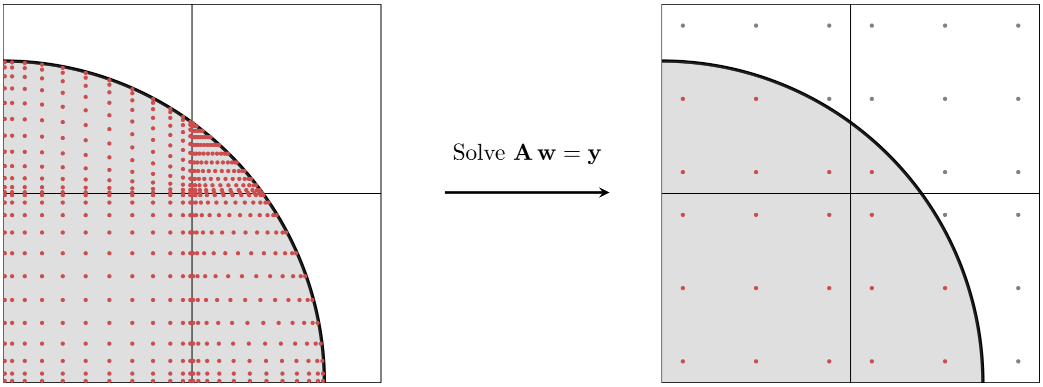


Figure 3: Moment fitting based on an accurate boundary-fitted quadrature rule.

Stencil of quadrature points

The above conditions for exact integration simplify considerably when the points are fixed a priori and only the weights are optimized. In this case the equations describe a matrix problem, $\mathbf{A}\mathbf{w} = \mathbf{y}$, that is linear in the weights $\mathbf{w}$. The conditions for existence and uniqueness are then simple. A solution exists if $\mathbf{A}$ has full row-rank, which means that $\mathbf{A}\mathbf{A}^T$ is invertible. If $\mathbf{A}$ itself is invertible then the solution is also unique.

In this work we consider the special case where the number of quadrature points m is equal to the dimension of the basis for the target space $\mathcal{V}^n$. Because $\left\{\phi_\alpha(\boldsymbol{x}),\ \alpha \in 1, \dots, n\right\}$ is a basis for $\mathcal{V}^n$, it follows that a unique set of weights exist when the n quadrature points $\left\{\boldsymbol{x}_j,\ j = 1, \dots, n\right\}$ are unique. Matrix $\mathbf{A}$ is then invertible and the unique set of weights can be computed from (3). In this work, we fix the stencil of quadrature points to coincide with a tensor product grid of Gauss-Legendre nodes. An example of a quarter circle embedded in a 2 × 2 cell mesh is shown in Figure 3. The integration points inside Ω are shown in red and outside of Ω in gray. The quadrature weights take values in $[-1, 1]$, see Remark 3.

Target space for quadrature

Let $b_{k,p}(x) := \binom{p}{k}\ x^k\ \ (1-x)^{p-k}$ denote the k'th univariate Bernstein polynomial of polynomial degree p. The target space $\mathcal{V}^n$, investigated in this paper, is the space

$$\mathcal{V}^n := \text{span}\left(\phi_\alpha(\boldsymbol{x}),\ \alpha \in 1, \dots, n\right) \tag{4}$$

of tensor product polynomials, $\phi_\alpha(\boldsymbol{x}) = b_{k,p}(x_1) \cdot b_{l,p}(x_2)$, $\alpha = (p+1) \cdot l + k + 1$. Its dimension is $n = (p+1)^2$.

Remark 2. Bernstein polynomials are strictly positive. This means that integrals of Bernstein polynomials are always positive and are not affected by numerical instability. For this reason, we use Bernstein polynomials in all computations.

Remark 3. Moment-fitted quadrature rules described in Section 2 are known to generate negative quadrature weights. While negative quadrature weights are typically not an issue when solving linear problems, in nonlinear problems they can lead to an instability of the Newton solver [11, 28, 38]. For nonlinear problems we recommend to replace Equation (3) by a nonlinear least squares problem that enforces positiveness of the quadrature weights [28, 38].

3. Parametrization of cut cells in two dimensions

For this approach, we restrict ourselves to two-dimensional cells and introduce a parametrization capable of representing a large configuration space of cut cells. This parametrization is subsequently used to generate target datasets for spline interpolation and testing the corresponding cut-cell integration accuracy. We will also explore the smoothness of the parameter space and introduce a corresponding partitioning strategy for spline-based interpolation.

3.1. Parametrization of circular cuts

We assume that we can approximate all cut contours via a circular arc. Therefore, our goal is to parametrize cut cells for which the cut contour is described by a circle with radius R. To simplify the geometric description, we assume that the cut contour is the zero contour of the signed distance function $\psi(\boldsymbol{x}) = \|\boldsymbol{x}\| - R$. Thus, all cuts are described by a circle located at the origin of the coordinate system.

Since the circular cut is fixed at the origin, we place the cells relative to the cut contour. Let us consider a square cell with edge size h, which is aligned with the x- and y-axis. The cell's centroid is denoted by $\boldsymbol{C}$. The ray from the origin through the centroid is denoted by $\boldsymbol{r}$. We introduce a radial position $z \in \mathbb{R}^+$ and an angular position $\theta \in [0, \pi/4]$ to characterize the position of a cell relative to the circular cut. The radial position $z = \|\boldsymbol{C}\|$ is the distance between the origin and the cell centroid along the ray $\boldsymbol{r}$. The angular position θ denotes the minimal angle between the ray $\boldsymbol{r}$ and the x-axis. Due to symmetry considerations it is sufficient to limit the angular position θ to the range $[0, \pi/4]$. Any cut cell for $\theta \in (\pi/4, 2\pi)$ can be transformed back to an equivalent cut-cell in the parameter range $[0, \pi/4]$.

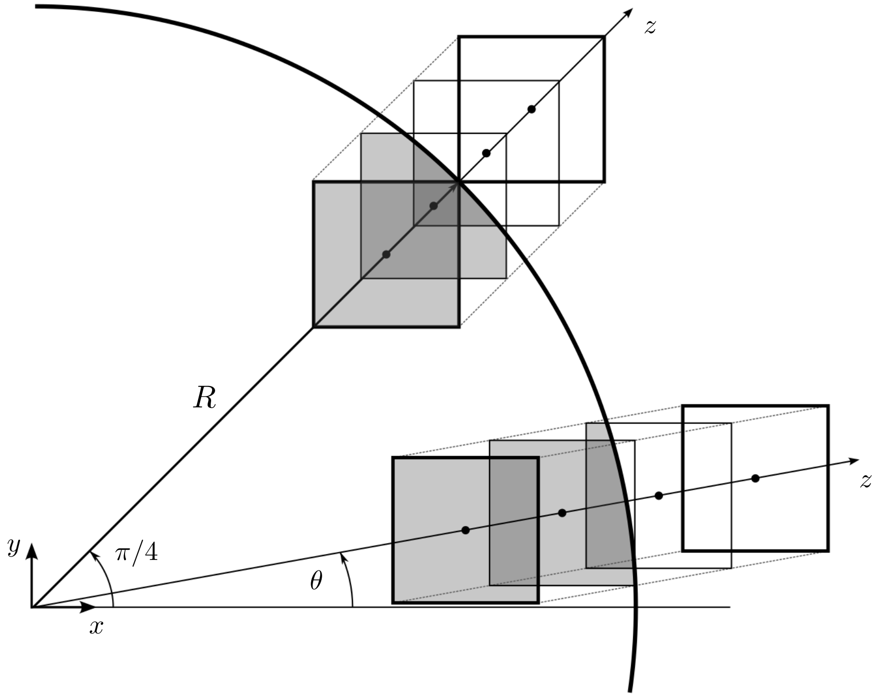


Figure 4: The configuration space of circular cuts is spanned by varying radius $R > 0$, varying angular position $0 \leq \theta \leq \pi/4$ and varying radial position, $z \geq 0$, of the cut cell with respect to the circle.

Given the circular cut parametrization and the cell diameter $d = h\sqrt{2}$, it is easy to see that for a fixed radius R and $z \geq R + d/2$ there is no angular position $\theta \in [0, \pi/4]$ for which the cell is cut. The cell is fully outside the circle except for the edge case $\theta = \pi/4$ and $z = R + d/2$, in which the lower-left cell corner touches the cut contour. Similarly, for a fixed radius R and $z \leq R - d/2$ the cell is fully inside the circle except for the edge case $\theta = \pi/4$ and $z = R - d/2$, in which the upper-right cell corner touches the cut contour. Figure 4 shows the configuration space of such circular cuts and the relative positioning of the cells. The edge cases described above can be identified at the angular position $\theta = \pi/4$.

3.2. Exploration of parameter space smoothness

Each cut cell represented by a unit square corresponds to a distinct set of parameters (R, z, θ). By varying the radius of the circle and the position of the cell we can span a large configuration space of relevant cut cases. Our goal now is to generate data which can serve as the basis for tensor-product spline interpolation. In particular, for a certain target space for the quadrature, we want to fit a tensor-product spline which maps (R, z, θ) to the moments in Equation (3). To achieve optimal approximation rates, the quadrature rules must be sampled from a smooth response surface. As we will demonstrate, particular attention must be paid to regions of the parameter space where the circle passes a corner of the cell or where the number of intersections with the cell boundary changes.

Figure 5 illustrates a circular cut of radius R through four square cells with edge size $h = 2R/3$. The angular position is fixed at $\theta = \pi/8$ and the ray through the centroids is now sketched horizontally. The first graph plots the definite integral of a constant function in the region inside the circle, i.e. the cut-cell area A as a function of z. The second graph depicts the corresponding integrand. A key observation is that the integral of a constant function over the square cut cell is a spline that is C^1 continuous at those locations where the z-coordinate passes a corner of the cell, or the number of intersections with the cell boundary changes. More generally, integrals of smooth functions over square cut cells are at least C^1 continuous as a function of z, θ and R. This implies that globally optimal higher-order convergence of tensor-product spline approximations can only be achieved when the parameter space is appropriately partitioned along the continuity transition loci.

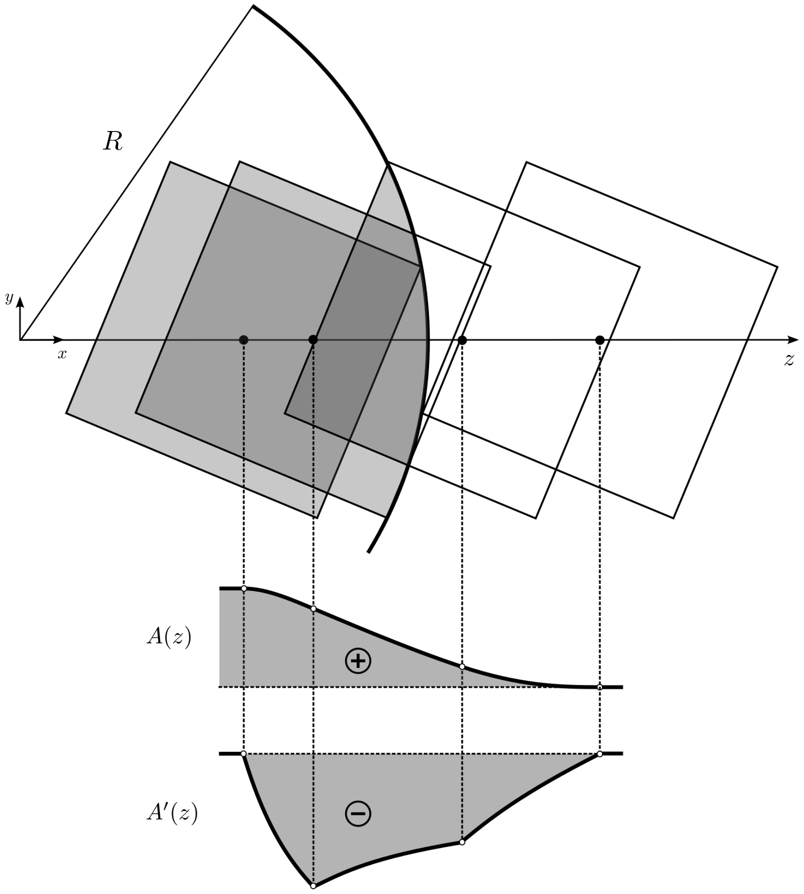


Figure 5: The integral $A(z)$ of a constant function over the square cell depicted here is a spline that is C^1 continuous at those locations where the z-coordinate passes a corner of the square element, as clearly visible in the derivative $A'(z) = \mathrm{d}A(z)/\mathrm{d}z$. More generally, integrals of smooth functions over the square cell are at least C^1 continuous functions of z, θ and R.

In order to find all continuity transition loci, we assume $R > h$ and explore the parameter space in a systematic way. The premise we follow while partitioning the parameter space is that the whole parameter range $R-d/2 \leq z \leq R+d/2$ and $0 \leq \theta \leq \pi/4$ must be covered. The assumption $R > h$ allows us to identify 15 distinct smooth cases, where no corner is passed by the circular cut and the number of intersections with the cell boundary is constant. Without the assumption $R > h$, the number of cases would be larger and their geometric description more involved. Moreover, a radius $R \leq h$ implies geometric features that are of the order of the element size, which in turn implies that mesh refinement could be required for accuracy.

3.3. Partitioning of the parameter space

We have already established that for $z = R - d/2$ and $\theta = \pi/4$ the cell is fully inside the circle and the upper-right corner touches the circular cut. Since for $z \leq R - d/2$ there is no θ for which the cell is cut, we can identify this limit case as one of the C^1 transition loci and define

$$z_9 = R - \frac{d}{2}. \tag{5}$$

For $z > z_9$ we observe that the angular position θ at which the upper-right corner touches the circular cut is smaller than $\pi/4$. We find the exact angular position from the condition that the norm of the vector from the origin to the upper-right cell corner must be equal to R,

$$\theta_8 = \arcsin\left(\frac{R^2 - z^2 - h^2/2}{zh\sqrt{2}}\right) - \frac{\pi}{4}. \tag{6}$$

This bounds the angular position to the range $\theta_8 \leq \theta \leq \pi/4$. The cuts in this range represent cut cells, whose cut is located at the upper-right corner region of the cell. These cuts intersect only the top and the right cell edge and do not pass any cell corners in the identified θ range. With an increasing radial position z, the angular position θ tends towards zero. At $\theta = 0$, the bottom-right corner comes into contact with the circular cut. Any further increase of the radial position z implies that the bottom-right corner passes the cut contour. Thus, this locus is one of the C^1 transition loci. From Equation (6) we find the upper bound on the radial position z,

$$z_8 = \sqrt{R^2 - \frac{h^2}{4}} - \frac{h}{2}. \tag{7}$$

With the radial position bounded in $z_9 \leq z < z_8$ and the angular position bounded in $\theta_8 \leq \theta \leq \pi/4$, we have successfully identified a smooth parameter partition which covers cases where the upper-right corner region is cut by the circular cut contour as shown in Figure 6.

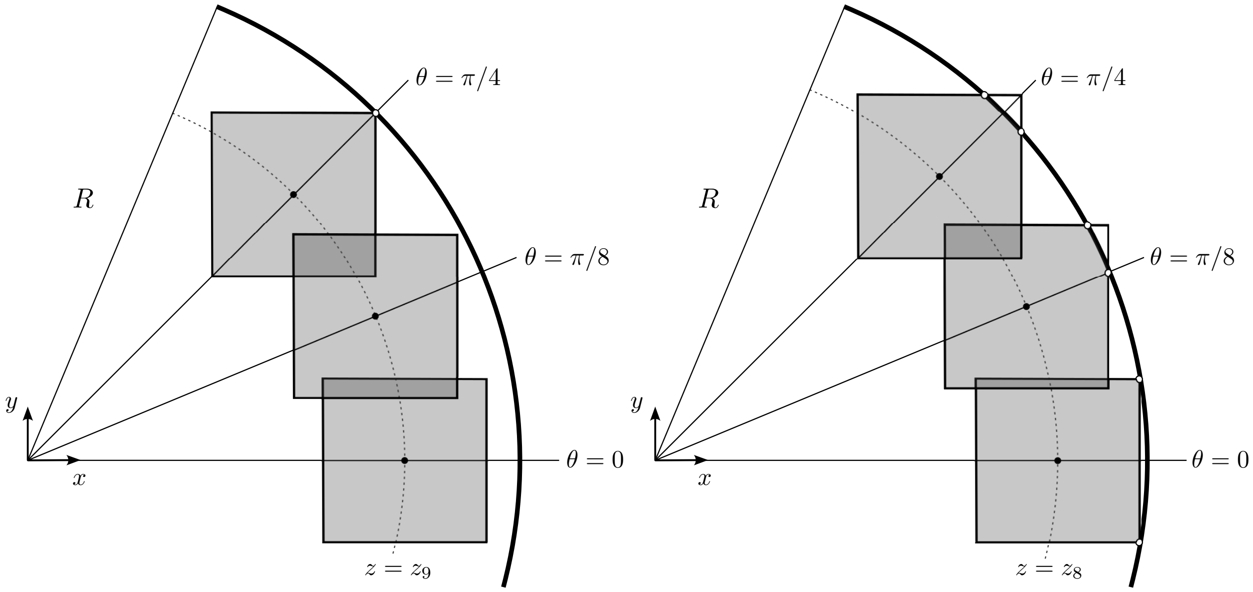


Figure 6: Sketch of the cut case with bounds $z_9 \leq z < z_8$ and $\theta_8 \leq \theta \leq \pi/4$.

Next, we analyze the next closest cut case. As shown in Figure 6 for the set of parameters $\theta = 0$ and $z = z_8$, the bottom-right and top-right corners are in contact with the cut contour. By increasing the radial position z we observe that both corner regions on the right side of the cell are cut by the circle. The radial position z can be increased only until the right edge comes into contact with the cut contour. Any further increase would change the number of intersections of the cut contour with the cell boundary. From the condition for the intersection of the right cell edge with the cut contour we find

$$z_7 = R - \frac{h}{2}. \tag{8}$$

This bounds the radial position to the range $z_8 \leq z < z_7$. Now, by increasing the angular position θ we see that the region cut-off at the bottom-right corner reduces in size and the corner itself approaches the cut contour. From the condition that the norm of the vector from the origin to the bottom-right cell corner must be equal to R, we find

$$\theta_6 = \arccos\left(\frac{R^2 - z^2 - h^2/2}{zh\sqrt{2}}\right) - \frac{\pi}{4}, \tag{9}$$

which bounds the angular position to the range $0 \leq \theta < \theta_6$.

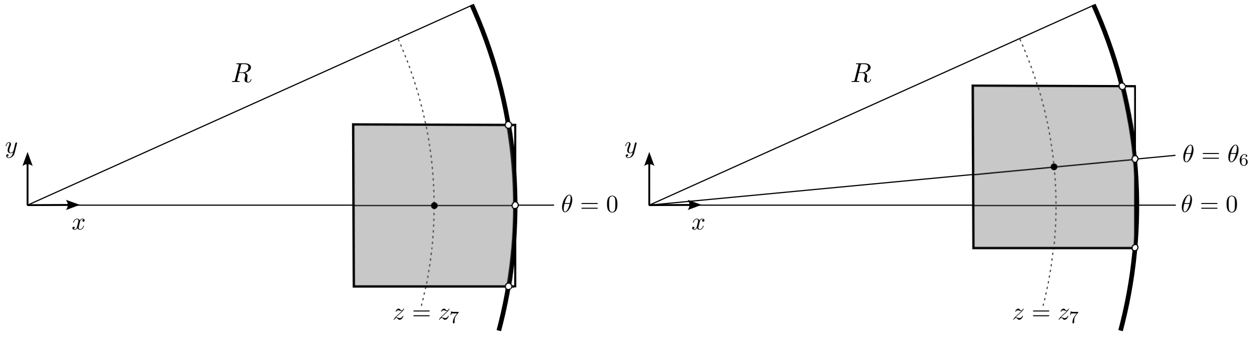


Figure 7: Sketch of the cut case with bounds $z_8 \leq z < z_7$ and $0 \leq \theta < \theta_6$.

Remark 4. It should be noted that both θ_6 and θ_8 are functions of the radial position z. Moreover, for the radial position $z = z_8$ we find that the angular positions θ_6 and θ_8 are equal to zero and $\pi/4$, respectively. Thus, for $z = z_8$ the angular position range degenerates in both cut cases to a point. This behavior does not pose an issue in the method. Nonetheless, it is a subtlety that will have important implications in Section 4.

Any square cell with edge size h that is intersected by a circle of radius $R > h$ can be assigned exactly one of the 15 smooth cut cases. The parameter domains associated with the individual cases are mutually disjoint. Hence, for any given combination of parameters z, θ and R there exists a unique case to which the cut cell belongs. For completeness we summarize all 15 smooth cut cases in Appendix A and show the partitioned parameter space in Figure 8. The cases illustrated above are case 15 and 13, respectively. We refer the reader to an interactive `GeoGebra` applet that visualizes all 15 configurations. The `GeoGebra` files are also included in our repository.

3.4. Normalization of cut-case parameters

We now have a parametrization of the configuration space of circular cuts as well as an appropriate partitioning of that parameter space for a fixed radius R. In each cut case defined by a partition, integrals are smooth functions of the radial position z and the angular position θ. In fact, integrals of smooth functions over the 15 cut cases in Appendix A are smooth functions not only in terms of z and θ but also in terms of the radius R of the circle. Since the geometry of the cut case is constrained by bounds on the radial position z and the angular position θ, which both depend on R, neither corners nor additional edges are intersected by varying R. For a fixed cut case the parameter R controls merely the curvature of the cut contour. In a sense, the bounds on z and θ characterize the limit shapes of the cut.

This observation suggests to normalize the ranges for z and θ in each cut case to an interval $[0, 1]$ and thus to use the set of parameters $\left\{(R, \hat{z}, \hat{\theta}) \,:\, R \in (1, \infty],\ \hat{z} \in [0, 1],\ \hat{\theta} \in [0, 1]\right\}$ to characterize each of the 15 cut cases numbered by $t = 1, \ldots, 15$. Conveniently, the normalization of the parameters z and θ simplifies the interpolation domain in each cut case to $[R_{\min}, R_{\max}] \times [0, 1] \times [0, 1]$. The parameters can be interpreted as follows. The cut case number t characterizes the type of the cut. The radius R controls the curvature. The normalized radial position $\hat{z}$ and the normalized angular

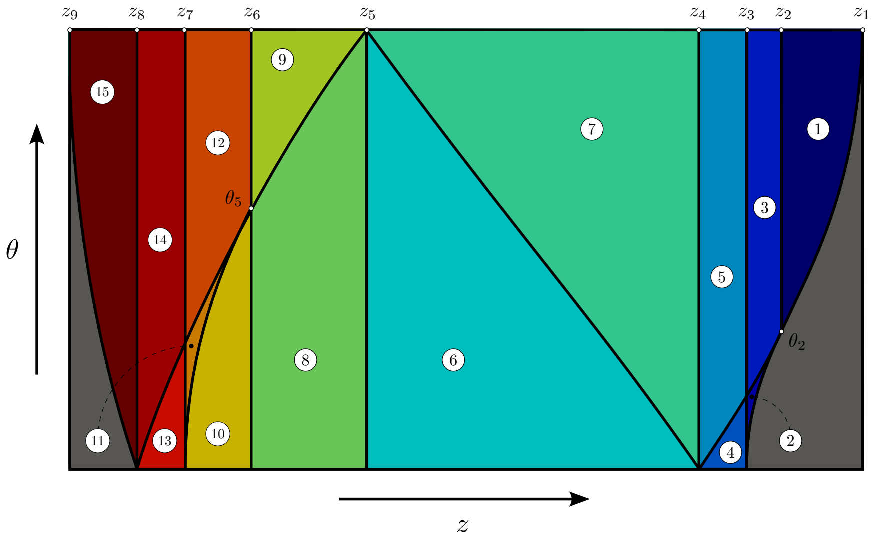


Figure 8: Partitions of the parameter space (z, θ) for unit square cells cut by a circle of radius $R = 1.5$. The colored regions labeled by a number correspond to smooth cut cases in $z_9 \leq z \leq z_1$ and $0 \leq \theta \leq \pi/4$. The gray regions correspond to cells which are either completely inside or completely outside of the circle. We annotate characteristic values for z and θ that depend only on the radius R and the cell width h.

position $\hat{\theta}$ localize the contour within the geometric constraints. Note that on the one hand, the tuple $(R, \hat{z}, \hat{\theta}, t)$ can always be associated with a unique (R, z, θ). On the other hand, the tuple (R, z, θ) may not always be uniquely expressed by $(R, \hat{z}, \hat{\theta}, t)$. For example, in cases where the angular position range degenerates to a point, the normalized parameter $\hat{\theta}$ becomes a free parameter that maps to a constant θ, see Remark 4. In these special cases, some cut configuration will be sampled multiple times. In practice, this is not an issue as the response surface is constant for $\hat{\theta} \in [0, 1]$.

4. Data-driven quadrature using spline interpolation

Our goal now is to design a data-driven approach that can generate a set of quadrature points and the corresponding weights given a level-set description of the cut contour. We use tensor-product B-splines to interpolate a given dataset of response surfaces of the moments in Equation (3) for all 15 smooth cut cases. The dataset is generated in an offline stage using Algoim [54]. In the online stage, given the spline approximation of the response surface of the moments and a circular approximation of the cut contour on a given cell, we can then evaluate the spline and obtain the moments we require in order to solve for the quadrature weights in Equation (3). By that we circumvent integration of the moments using expensive quadrature schemes. For cut-cell quadrature in nonlinear problems, we could directly interpolate the response surface of the quadrature weights and thus shift the solution of the nonlinear least squares problem to the offline stage, see Remark 3. In this work, we consider only linear problems and solve the linear problem in Equation (3) in the online stage.

4.1. Overview of the data-driven quadrature pipeline

Figure 9 shows the procedure of obtaining data-driven patch quadrature rules with Cartesian-product quadrature points. Cartesian-product quadrature points are suitable for formation and assembly via sum factorization on tensor-product spaces [2, 10, 31]. Given a level-set description of the cut as well as the cell domain, we first find a circle that best approximates the cut in the least squares sense. Here, we can reuse expensive level-set evaluations from the identification of cut cells. The cut configuration is then mapped back to the parametrization of circular cuts introduced in Section 3 by means of translation and scaling operations. Additionally, if the cut falls outside of the interpolation range of the angular positions $\theta \in [0, \pi/4]$, we find the corresponding symmetry transformation to an equivalent cut in the interpolation range. For the identified cut configuration, we find the corresponding cut case and normalize the radial and angular positions. We then evaluate the moments for the target space for the quadrature and map these integrals forward to the physical cell. Depending on the phase to be integrated, we may have to compute the complement of the moments integral on the cell domain which can be achieved via standard Gauss rules. Having the required moments, we can solve Equation (3) for the corresponding quadrature weights. If the cell is not cut and fully inside the cut contour, we return the standard Gauss rule for that cell. If it is fully outside, we set the weights to zero.

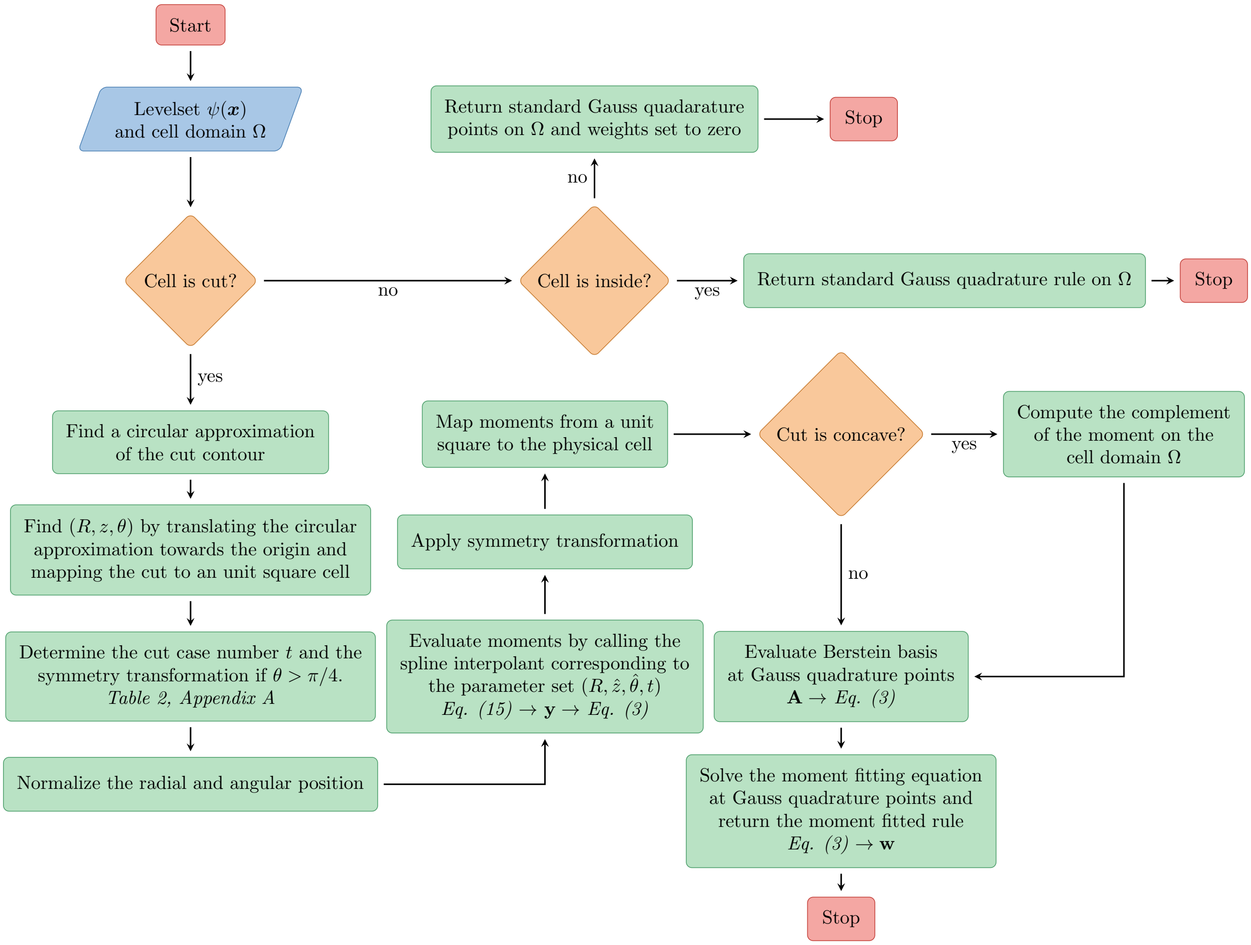

Start
Levelset $\psi(\boldsymbol{x})$ and cell domain Ω
Return standard Gauss quadrature points on Ω and weights set to zero
Stop
no
Cell is cut?
no
Cell is inside?
yes
Return standard Gauss quadrature rule on Ω
Stop
yes
Find a circular approximation of the cut contour
Map moments from a unit square to the physical cell
Cut is concave?
yes
Compute the complement of the moment on the cell domain Ω
Find (R, z, θ) by translating the circular approximation towards the origin and mapping the cut to an unit square cell
Apply symmetry transformation
no
Determine the cut case number t and the symmetry transformation if $\theta > \pi/4$. *Table 2, Appendix A*
Evaluate moments by calling the spline interpolant corresponding to the parameter set $(R, \hat{z}, \hat{\theta}, t)$ *Eq. (15)* $\rightarrow \mathbf{y} \rightarrow$ *Eq. (3)*
Evaluate Berstein basis at Gauss quadrature points $\mathbf{A} \rightarrow$ *Eq. (3)*
Normalize the radial and angular position
Solve the moment fitting equation at Gauss quadrature points and return the moment fitted rule *Eq. (3)* $\rightarrow \mathbf{w}$
Stop

Figure 9: Flowchart summarizing the generation of cut-cell quadrature rules with our data-driven approach for a cell patch.

4.2. Tensor-product B-spline interpolation of the response surface

Our target space for the quadrature is spanned by q'th degree tensor-product Bernstein polynomials $\phi_\alpha(\boldsymbol{x})$, $\alpha = 1, \dots, (q+1)^2$. We consider a parametrized integration domain Ω_c that represents the region of a reference cut cell that is inside a circular disk. For a fixed cut case t, the integration domain is uniquely determined by $\boldsymbol{\xi} = (R, \hat{z}, \hat{\theta})$. The moments of $\phi_\alpha(\boldsymbol{x})$ on the integration domain Ω_c are denoted by

$$m_\alpha(\boldsymbol{\xi}) = \int_{\Omega_c} \phi_\alpha \,\mathrm{d}\boldsymbol{x}. \tag{10}$$

We use tensor-product B-splines to interpolate the response surface of all $(q+1)^2$ moments in terms of $\boldsymbol{\xi}$.

An univariate B-spline basis function $B_{i,p}(\xi)$, $\xi \in \mathbb{R}$, is a piecewise polynomial of degree p characterized by a non-decreasing knot vector $\left\{\xi_1, \xi_2, \dots, \xi_{n+p+1}\right\}$ where $\xi_1 \leq \xi_2 \leq \dots \leq \xi_{n+p+1}$ and $\xi_j \in \mathbb{R}$. The knots do not need to be unique. A repeated knot with multiplicity r reduces the smoothness of the basis at that particular knot to C^{p-r}. In the case of open knot vectors, the first and last knot have multiplicity $p+1$ such that the basis becomes interpolatory at those knots. In this work we consider open knot vectors and B-splines of maximum regularity [7].

In the trivariate setting, the basis functions in terms of $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$ are constructed as a tensor-product of the univariate basis functions,

$$B_{\mathrm{i}}(\boldsymbol{\xi}) = \prod_{k=1}^{d} B_{i_k,p_k}(\xi_k). \tag{11}$$

Here, the multi-index i in the index set $\mathcal{I} = \left\{(i_1, \dots, i_d) \ : \ 1 \le i_k \le n_k, k = 1, 2, 3\right\}$ is used to denote the multivariate basis functions. The polynomial degree of the interpolation space is set to q^2 in each parametric direction. Given a set of interpolation sites $\left\{\boldsymbol{\xi}_{\mathrm{i}}\right\}_{\mathrm{i}\in\mathcal{I}}$, the interpolation problem is stated as

$$m_\alpha(\boldsymbol{\xi}_j) = \sum_{\mathrm{i}\in\mathcal{I}} c_{\mathrm{i},\alpha} B_{\mathrm{i}}(\boldsymbol{\xi}_{\mathrm{j}}) \tag{12}$$

where $c_{\mathrm{i},\alpha}$ are the control points corresponding to the moment $m_\alpha(\boldsymbol{\xi})$. We choose the interpolation sites to be a Cartesian product of the univariate Greville abscissae [7] which in the k'th parametric direction is given by

$$\tau_{i_k} = \frac{\xi_{i_k+1} + \xi_{i_k+2} + \cdots + \xi_{i_k+p_k}}{p_k}, \qquad i_k = 1, \dots, n_k. \tag{13}$$

Taking advantage of the separable nature of tensor-product B-splines evaluated at Cartesian product interpolation sites, we can write the interpolation problem in matrix form, where the interpolation matrix has a Kronecker product structure. We denote the small, univariate interpolation matrices in the k'th dimension by $[\mathsf{M}_k]_{i_k j_k} = B_{j_k,p_k}(\tau_{i_k})$. The multivariate interpolation matrix is given by $\mathbf{M} = \mathsf{M}_3 \otimes \mathsf{M}_2 \otimes \mathsf{M}_1$. The interpolation problem in matrix form reads

$$\mathbf{M}\mathbf{c}_\alpha = \mathbf{m}_\alpha \tag{14}$$

where $[\mathbf{c}_\alpha]_{\mathrm{i}} = c_{\mathrm{i},\alpha}$ and $[\mathbf{m}_\alpha]_{\mathrm{i}} = m_\alpha(\boldsymbol{\xi}_{\mathrm{i}})$.

4.3. Implementation aspects

In practice, we fix the interpolation space for all α, such that the interpolation matrix $\mathbf{M}$ is computed only once and the interpolation problem is solved for $(q+1)^2$ right-hand sides. The right-hand sides in the offline stage are evaluated to high precision using a Julia wrapper around Algoim [54]. Note that the inverse of the Kronecker product interpolation matrix is $\mathbf{M}^{-1} = \mathsf{M}_3^{-1} \otimes \mathsf{M}_2^{-1} \otimes \mathsf{M}_1^{-1}$. This greatly reduces the computational cost of the interpolation as the univariate matrices are small enough to be inverted directly. Similarly, the evaluation of the spline interpolant is inexpensive, since the interpolation matrix must never be explicitly assembled. The inference of the moments $\tilde{m}_\alpha$ from the interpolant scales with $\mathcal{O}(N^{4/3})$, where N denotes the dimension of the interpolation space,

$$\tilde{m}_\alpha(\boldsymbol{\xi}) = \mathbf{M}^{-1}(\boldsymbol{\xi})\, \mathbf{c}_\alpha. \tag{15}$$

While the polynomial degree of the interpolation space is fixed for a particular target space for the quadrature, the knot vectors can vary between the 15 cut cases. We refine the spaces per case in a manual fashion as needed to reduce the interpolation error in the test data. The test data is generated as a Cartesian product of the midpoints of the Greville abscissae,

$$\mu_{i_k} = \frac{\tau_{i_k} + \tau_{i_k+1}}{2}, \qquad i_k = 1, \dots, n_k - 1. \tag{16}$$

To identify regions of the parameter space that require refinement, we analyze the pointwise error in the test data, $|m_\alpha(\boldsymbol{\mu}) - \tilde{m}_\alpha(\boldsymbol{\mu})|$, and refine accordingly. This is performed manually for a small radius range and all cut cases, and the resulting interpolation spaces were used for all other radii in the interpolation range.

In order to have additional flexibility in the interpolation, we set up separate interpolants for radii in the ranges $[R, R+1]$ with $R = 1, \dots, 30$. Thus, we can always extend the interpolation range if necessary.

4.4. Adaptive quadrature

The cut parametrization used in our data-driven approach has the strict constraint that the radius of the circle approximating the cut must be larger than the width of the cell, $R > h$. This constraint was introduced to limit the number of special cases arising in the partitioning of the parameter space of circular cuts. In the context of cut finite element methods, a violation of the constraint implies geometric features that are of the order of the element size. Thus, in practice we require refinement of the embedding mesh to resolve such geometric features. With sufficient refinement, the constraint is satisfied.

To improve robustness in general applications, we implement an adaptive subdivision technique based on quadtrees. We subdivide cells that violate the constraint $R > h$ until the constraint is satisfied and our data-driven approach is applicable. The stencils of quadrature rules generated using this approach are Cartesian only on the leaves of the quadtree that represents the adaptive subdivision. We note that in our case, the goal of the quadtree approach is not to increase the accuracy of cut-cell integration as for instance in the finite cell method [48, 57]. The objective is merely to satisfy the constraint $R > h$. A similar subdivision process is used in Algoim if a suitable height function direction cannot be found on a coarse cell [54].

5. Numerical experiments and results

In the following, we present numerical experiments in which we use our spline interpolants to obtain quadrature rules for various cut domains. We first use the data-driven quadrature rules to integrate a constant function over a cut domain. In particular, we measure the area of a squircle that is deformed by sinusoidal terms. We compare the area with results that we obtained with quadrature rules computed with the Algoim library [54]. Next, we show convergence rates for a potential flow problem around an elliptic cross-section. Finally, we perform an eigenvalue analysis of stiffness matrices integrated with different quadrature rules to assess the corresponding accuracy in more detail.

5.1. Accuracy and performance of spline-based moment interpolation

Figure 10 shows the response surface of the nine moments corresponding to a quadrature rule of order two and the sixth cut case. The response surface in the sixth case is particularly interesting because it covers a large partition of the parameter space, see Figure 8. We fix the radius to $R = 5$ and plot the response surface in terms of the normalized radial position $\hat{z}$ and the normalized angular position $\hat{\theta}$. The depicted grid represents a slice of the actual data set, i.e. the grid of Greville points used in the interpolation. The sensitivity of the qualitative shape of the response surface to the cut radius R is low. Therefore, Figure 10 is qualitatively representative for other radii. We can see that the interpolation space was refined near $\hat{\theta} = 1$ in order to reduce the error in that region of the parameter space. The response surface is also constant for $\hat{z} = 1$ as can be verified in Figure 8, where the range for θ degenerates to a point.

Figure 11 illustrates the pointwise error distribution in more detail. The violin plots show the statistical distribution of the magnitude of the pointwise error in the test dataset for a fixed radius $R = 5$ and all nine moments in all 15 cut cases. In many cut cases, the error spans a wide range of values. The refinement of the interpolation space will reduce the error in all moments, but the error range between the moments tends to remain unaffected by the refinement. Thus, while some moments might be already approximated up to machine precision, others might not yet reach the required accuracy. In some cut cases, a few moments exhibit larger errors. A particular cut case with a larger error margin is the case number seven. In that case the errors in the first, fourth and seventh moment reach values $\mathcal{O}(10^{-5})$ with a small number of grid points. As this is the accuracy level we aim for in our data-driven approach, we choose not to increase the fidelity of the current spline interpolant. Overall, the mean error in the test data is $\mathcal{O}(10^{-12})$ and the maximum error is $\mathcal{O}(10^{-5})$, albeit the standard deviation in all cases is $\mathcal{O}(10^{-8})$ on average.

In Figure 12 we compare our data-driven approach to generate quadrature rules on cells cut by random non-polynomial contours to a direct application of moment-fitting. Our data-driven approach shifts the computational effort of moment evaluation to an offline stage, where we compute tensor-product splines that approximate the response surface of the moments. In our current implementation we observe average speedup factors of 50.46 for rules of equal accuracy order.

Remark 5. In this work, the spline spaces used to approximate the response surface have on average around 7,000 degrees of freedom for $q = 2$ and close to 14,000 degrees of freedom for $q = 3$. The ad-hoc refinement approach we employ in this work is tedious and surely not optimal. An automated and more sophisticated approach to choose appropriate spline spaces for the interpolation could lead to smaller interpolants that achieve better accuracy, in

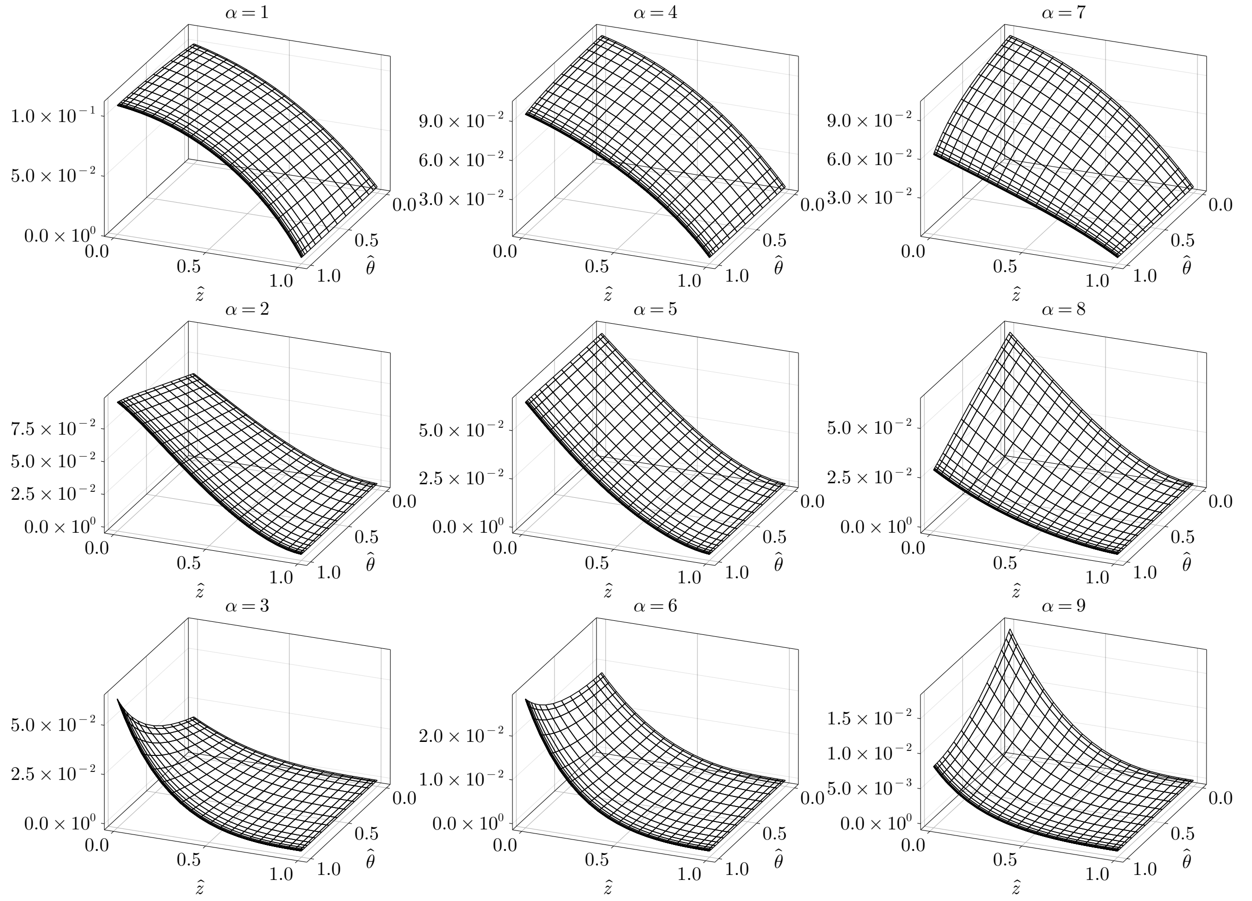


Figure 10: Response surface of moments $\int_\Omega \phi_\alpha \, \mathrm{d}\boldsymbol{x}$, $\alpha = 1, \ldots, (q+1)^2$, in terms of the normalized radial position $\hat{z} \in [0, 1]$ and the normalized angular position $\hat{\theta} \in [0, 1]$ in cut case six for a fixed radius $R = 5$ and the quadrature order $q = 2$. The qualitative shape of the response function has low sensitivity with respect to the radius R.

particular if the spline spaces are to be specialized for different ranges of the radius of the circular cut and/or the individual moments. We leave that aspect to future work.

5.2. Accuracy of interpolation-based area integration

Let us consider the following level-set function of a squircle with a sinusoidal deformation,

$$\psi(\boldsymbol{x}) = \left(\frac{x_1}{2}\right)^4 + \left(\frac{x_2}{2}\right)^4 + \cos\left(\frac{3}{2}x_1\right) - \sin\left(\frac{3}{2}x_2\right) - 1.5. \tag{17}$$

We are interested in computing the area $A = \int_\Omega \mathrm{d}\boldsymbol{x}$ of the region Ω enclosed by the contour $\psi(\boldsymbol{x}) = 0$ using our data-driven quadrature approach. Figure 13 illustrates the region Ω and the quadrature points computed using our data-driven approach. The embedding domain is meshed by 16×16 elements. The black line shows the exact contour line. The orange line plotted on top of the slightly thicker black line shows the contour approximation by circular arcs. The quadrature points are plotted as gray points with a diameter that is proportional to the magnitude of the corresponding weight. Note that the stencil forms a Cartesian product of quadrature points on the complete patch.

We compute a reference area A_{ref} enclosed by the exact contour by evaluating a high-order quadrature rule obtained from Algoim [54] on a mesh with $1{,}024^2$ cells. Table 1 shows the absolute error with respect to the reference area, $|A - A_{\mathrm{ref}}|$, for consequent uniform subdivisions of the embedding domain and different orders of the quadrature rule. On coarse discretizations, we observe that for $q = 2$ the proposed approach to cut-cell integration performs significantly better than the corresponding quadrature rule obtained via Algoim [54]. For $q = 3$ the proposed method

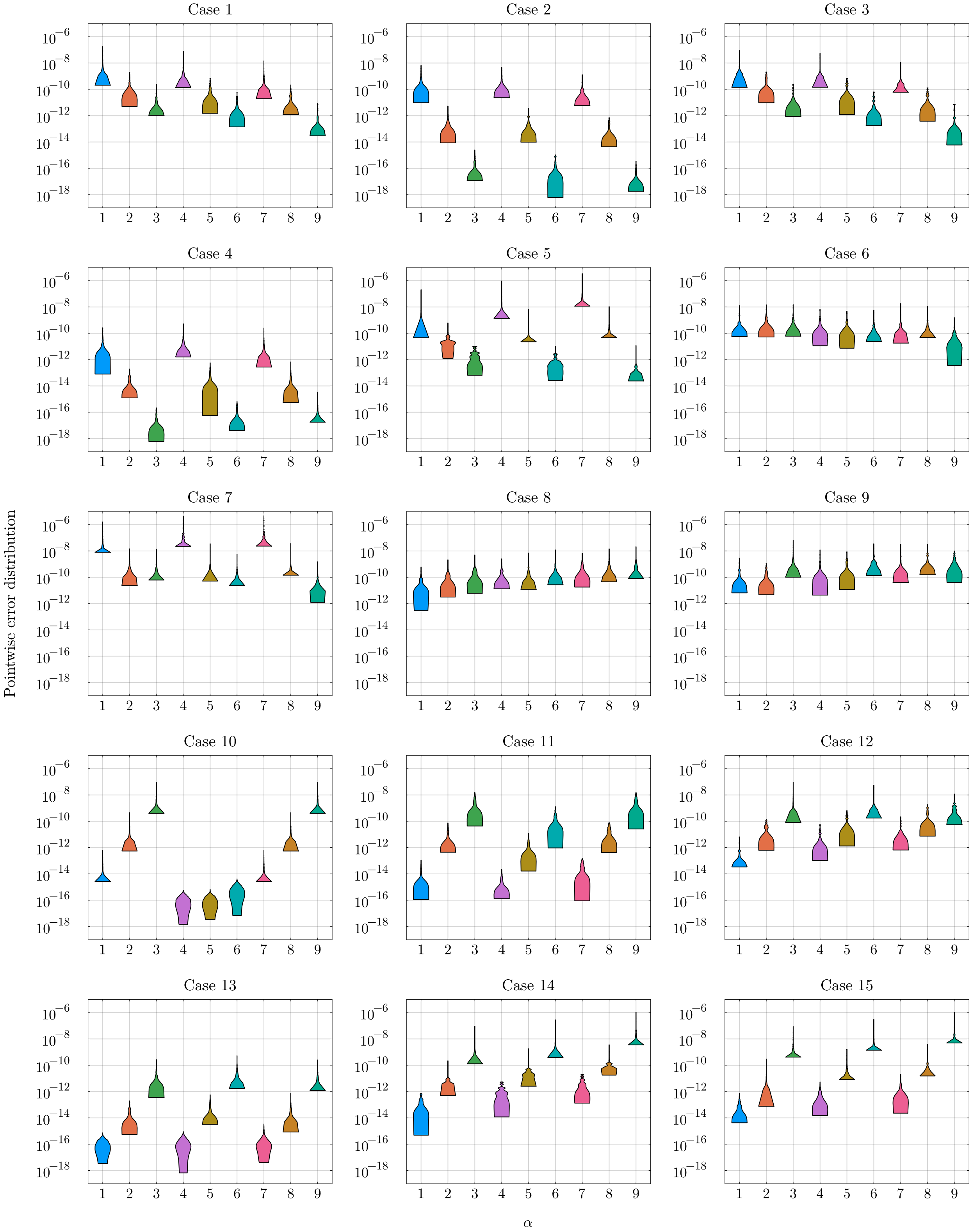


Figure 11: Distribution of the pointwise error $|m_\alpha - \tilde{m}_\alpha|$ of moments $m_\alpha = \int_\Omega \phi_\alpha \, \mathrm{d}\boldsymbol{x}$, $\alpha = 1, \dots, (q+1)^2$ and the spline interpolant estimates $\tilde{m}_\alpha$, on the test data grid for radii R in the interval $[5, 6]$ and the quadrature order $q = 2$.

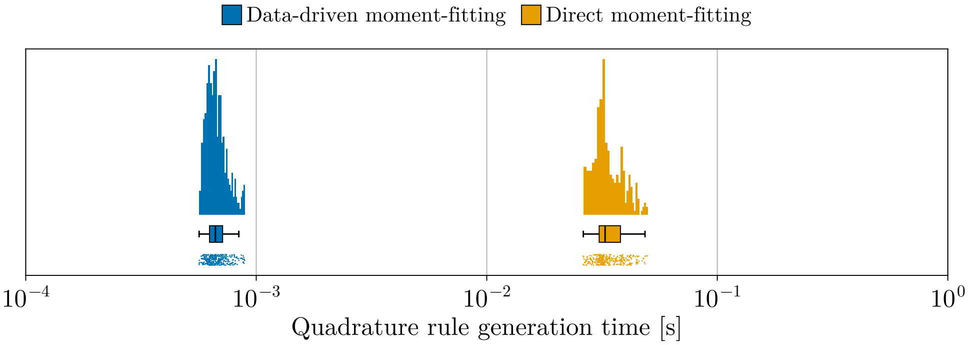


Figure 12: Tukey style plot of the distribution of computational time needed to generate quadrature rules on cells cut by random non-polynomial contours. The data-driven approach, where the moments are obtained from an evaluation of a tensor-product spline computed in an offline stage, outperforms direct moment-fitting with online evaluation of the moments on average by a factor of 50.46.

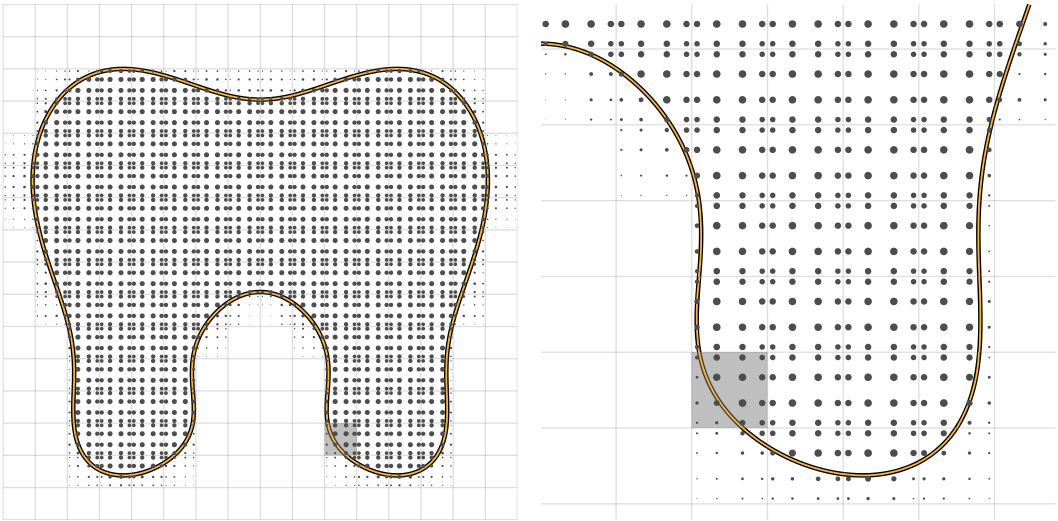


Figure 13: Squircle with sinusoidal deformation and the corresponding quadrature points on the domain $[-3,3]^2$ meshed by 16×16 elements. The diameter of a quadrature point is proportional to the magnitude of the corresponding weight.

still outperforms Algoim [54] in all cases but one, although the difference is much smaller for third-order rules. For $q = 4$ and finer meshes our method is outperformed by Algoim [54]. We run into the accuracy limit of the interpolants used in the approximation of the moments. Interpolants of higher fidelity are necessary to further reduce the error. This simple test case shows that given an appropriate interpolation, smooth contours can be accurately captured by the proposed method. The accuracy improves with mesh refinement, until the fidelity limit of the interpolant is reached.

5.3. Accuracy of cut finite element analysis using interpolation-based quadrature

In this example, we use our technique for cut-cell integration to form the system matrix in a cut finite element method. We use the method to model an incompressible, irrotational flow of an inviscid fluid around a cylinder with an elliptic cross-section. The ellipse describing the cross-section has principal axes $a = 1/3$ and $b = 1/6$ and is aligned with the grid. Far from the cylinder we assume a constant unit pressure and a uniform flow with unit velocity U. The complex potential $F_{\text{ref}}(z) = u_{\text{ref}}(x, y) + i\hat{u}_{\text{ref}}(x, y)$ that characterizes the flow in terms of $z = x + iy$ is

$$F_{\text{ref}}(z) = -U\left(\frac{a+b}{2}\right)\left[\frac{z+\sqrt{z^2-c^2}}{a+b} + \frac{z-\sqrt{z^2-c^2}}{a+b}\right], \tag{18}$$

Table 1
Absolute error in the area measure in the test case of a squircle with sinusoidal deformation. The embedding domain $[-3,3]^2$ is meshed uniformly resulting in N square elements. The reference area was computed using high-order Algoim quadrature rule [54] on a mesh with $1{,}024^2$ elements.

Order	Approach	$N = 4^2$	$N = 8^2$	$N = 16^2$	$N = 32^2$	$N = 64^2$	$N = 128^2$
$q = 2$	Ours	$3.5447 \cdot 10^{-2}$	$2.8964 \cdot 10^{-4}$	$1.8544 \cdot 10^{-4}$	$6.1210 \cdot 10^{-5}$	$1.0542 \cdot 10^{-5}$	$9.9767 \cdot 10^{-6}$
	Algoim	$5.1852 \cdot 10^{0}$	$1.2172 \cdot 10^{0}$	$3.1580 \cdot 10^{-1}$	$7.8422 \cdot 10^{-2}$	$1.9216 \cdot 10^{-2}$	$5.0909 \cdot 10^{-3}$
$q = 3$	Ours	$3.5446 \cdot 10^{-2}$	$2.8967 \cdot 10^{-4}$	$1.8546 \cdot 10^{-4}$	$6.1202 \cdot 10^{-5}$	$1.0541 \cdot 10^{-5}$	$9.9799 \cdot 10^{-6}$
	Algoim	$1.7056 \cdot 10^{-1}$	$2.3931 \cdot 10^{-2}$	$1.3779 \cdot 10^{-3}$	$3.6315 \cdot 10^{-4}$	$4.2792 \cdot 10^{-5}$	$2.1142 \cdot 10^{-6}$
$q = 4$	Ours	$3.5446 \cdot 10^{-2}$	$2.8965 \cdot 10^{-4}$	$1.9857 \cdot 10^{-4}$	$5.7083 \cdot 10^{-5}$	$1.3262 \cdot 10^{-5}$	$6.3017 \cdot 10^{-6}$
	Algoim	$3.5308 \cdot 10^{-2}$	$1.0392 \cdot 10^{-3}$	$1.8291 \cdot 10^{-5}$	$2.4212 \cdot 10^{-7}$	$3.2349 \cdot 10^{-7}$	$1.0572 \cdot 10^{-8}$

where $c = \sqrt{a^2 - b^2}$ is the linear eccentricity of the elliptic cross-section. The real part u_{ref} corresponds the velocity potential and the imaginary part $\hat{u}_{\text{ref}}$ is the conjugate stream function to u_{ref}.

We formulate the boundary value problem in terms of the velocity potential. Let us consider a bounded two dimensional domain $\Omega = \Omega_s \setminus \Omega_c$. The embedding domain Ω_s is a square $[-1,1]^2$. The region Ω_c subtracted from Ω_s is the elliptic disk representing the cross-section of the cylinder. The boundary of the domain Ω is $\partial\Omega = \Gamma_D \cup \Gamma_N$, where $\Gamma_D = \partial\Omega_s$ and $\Gamma_N = \left\{ \boldsymbol{x} \in \mathbb{R}^2 \,:\, (x_1/a)^2 + (x_2/b)^2 = 1.0 \right\}$. The unit normal on $\partial\Omega$ is denoted by $\boldsymbol{n}$. The velocity potential $u \,:\, \Omega \to \mathbb{R}$ that characterizes the flow satisfies the following boundary value problem,

$$\begin{aligned} \Delta u &= 0 && \text{in} \quad \Omega \\ u &= u_{\text{ref}} && \text{on} \quad \Gamma_D \\ \nabla u \cdot \boldsymbol{n} &= 0 && \text{on} \quad \Gamma_N. \end{aligned} \tag{19}$$

We want to find a velocity potential $u \in H^1(\Omega)$ that satisfies the weak form of the boundary value problem above. Let $V_0 = \left\{ v \in H^1(\Omega) \,:\, v = 0 \text{ on } \Gamma_D \right\}$. Given a lifting function $u_D \in H^1(\Omega)$ with $u_D = u_{\text{ref}}$ on Γ_D and $u = u_0 + u_D$, we identify a homogeneous potential $u_0 \in V_0$ that satisfies

$$\int_\Omega \nabla u_0 \cdot \nabla v \,\mathrm{d}x = -\int_\Omega \nabla u_D \cdot \nabla v \,\mathrm{d}x \quad \forall v \in V_0. \tag{20}$$

In the finite dimensional setting, we search for $u_0^h \in V_0^h \subset V_0$ and set V_0^h to tensor-product B-splines that satisfy homogeneous boundary conditions on the Dirichlet boundary,

$$V_0^h = \left\{ B_{\mathrm{i}} \,:\, \mathrm{i} = (i_1, i_2),\ 2 \leq i_k \leq (n_k - 1),\ 1 \leq k \leq 2 \right\}. \tag{21}$$

The discretized variational formulation of the potential flow problem in the immersed setting reads

$$\int_\Omega \nabla u_0^h \cdot \nabla v \,\mathrm{d}x = -\int_\Omega \nabla u_D \cdot \nabla v \,\mathrm{d}x \quad \forall v \in V_0^h. \tag{22}$$

The potential u_0^h extends smoothly into Ω_c.

We use our data-driven quadrature rules to form the system matrix for polynomial orders $p = 2, 3$. We solve for the velocity potential and use the real part of the complex potential in Equation (18) to set the Dirichlet boundary conditions on Ω_s. Figure 14 shows a zoom into the embedding domain together with the quadrature points of a third-order quadrature rule. We observe that the circular approximation of the elliptic cut fits the contour accurately, even though the curvature of the cut contour is not constant within an element. Under refinement, the variation of the curvature of the cut contour within an element is reduced and the circular approximation of the cut improves. Given that the error in the contour approximation is lower than the discretization error and the quadrature weight interpolants are sufficiently accurate, we expect the optimal rate of convergence of the method to be preserved.

Figure 15 shows the velocity potential together with the pressure field and the streamlines obtained from a cut finite element approximation on an embedding mesh of 32^2 elements, where cut elements are integrated via our

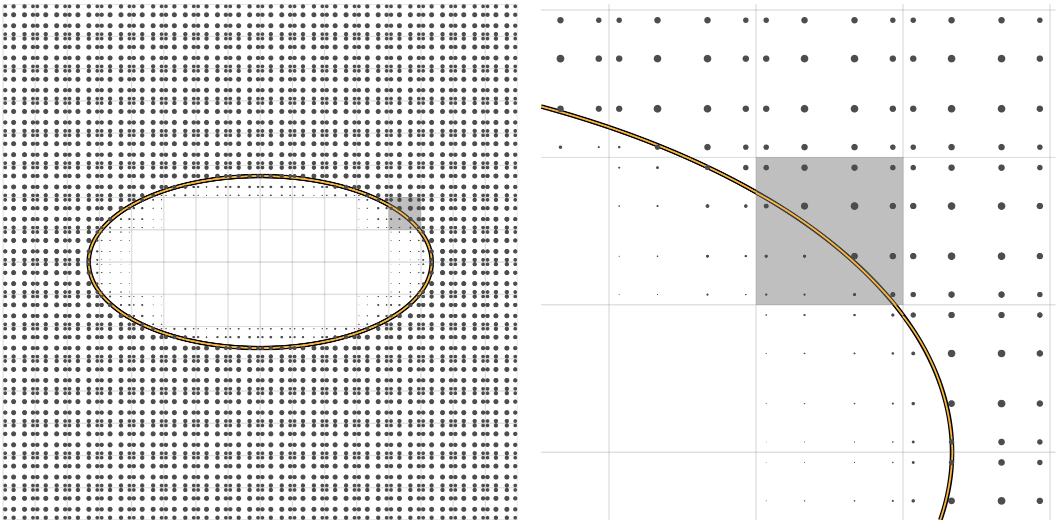


Figure 14: Elliptic cut of with principal axes $a = 1/3$ and $b = 1/6$ together with corresponding quadrature points on the domain $[-0.5, 0.5]^2$ meshed by 16×16 elements. The diameter of a quadrature point is proportional to magnitude of the corresponding weight.

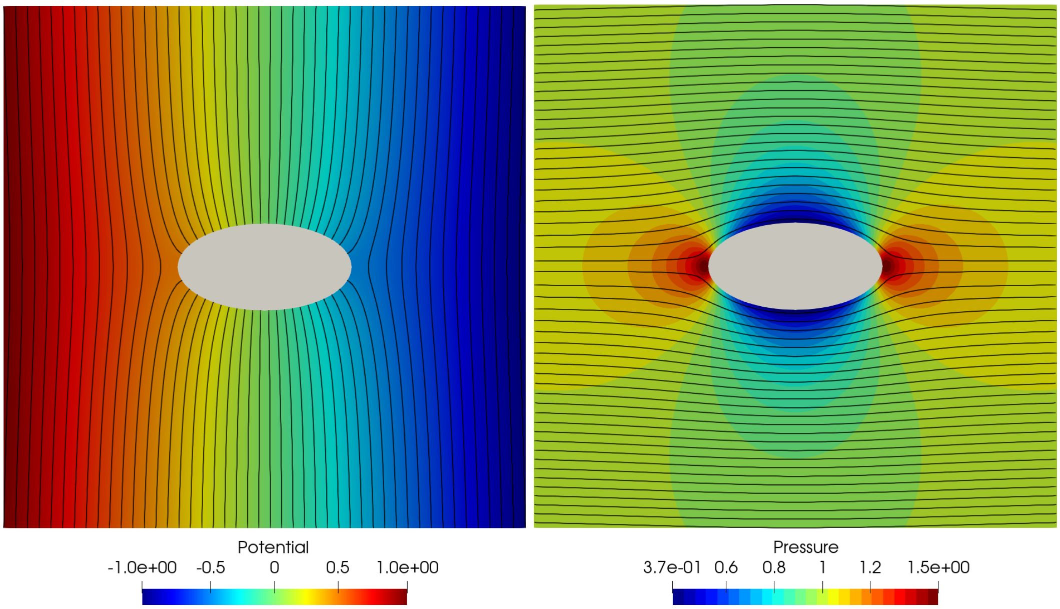


Figure 15: Velocity potential with contour lines and pressure along with streamlines for the test case of an elliptic cut.

interpolation-based approach. The pressure field is found directly from the gradient of the velocity potential and the Bernoulli's equation. The two stagnation points are located along the major axis of the elliptic cut. We do not observe any potential issues in the solution fields.

Figure 16 shows the convergence of the relative error in the L^2 norm for the velocity potential. The error is measured with respect to the analytical solution given in Equation (18) and is plotted against reference computations using quadrature rules generated by Algoim [54] for quadrature orders $q = 2, 3$. The error follows the reference error measure closely and the convergence rates are close to the expected rate $\mathcal{O}(h^{p+1})$.

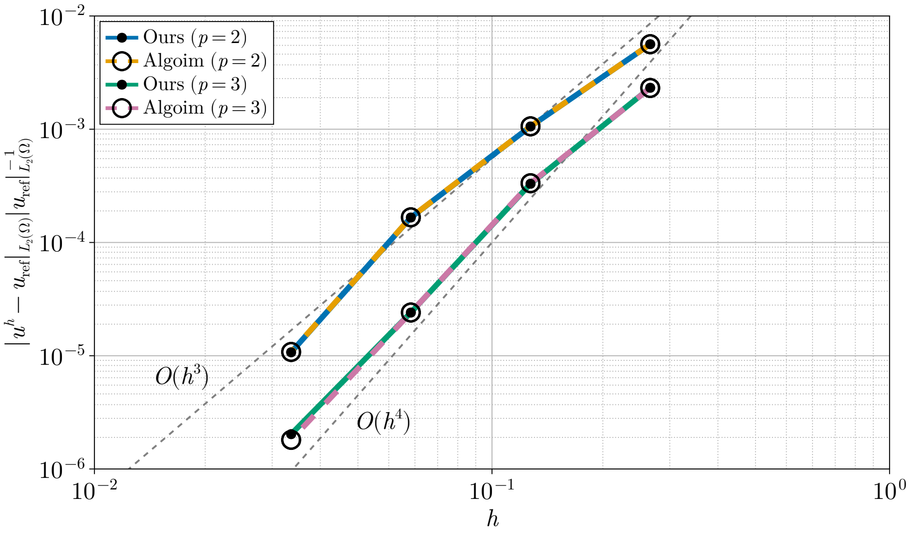


Figure 16: Convergence rate of the relative L^2 error in the velocity potential for the elliptic cut problem, integrated via quadrature rules of order $q = 2, 3$: comparison between our interpolation-based approach and Algoim [54].

5.4. Spectral accuracy of stiffness matrices

In the final example, we consider a gear-shaped geometry shown in Figure 17. We perform an eigenvalue analysis of the stiffness matrix which allows us to quantify the accuracy of our quadrature technique in more detail. We discretize the square embedding domain by 16 elements in each direction and set the polynomial degree to $p = 2$. Since the discretization does not resolve the geometric features, we resort to the adaptive variant of our quadrature approach. The resulting quadtree structure and the associated quadrature points are shown in Figure 17. Out of the 256 elements, 44 elements are cut.

We compare the spectrum of the stiffness matrix integrated with our interpolation-based approach to the spectra of stiffness matrices integrated with Algoim rules of two different accuracy levels. When we use the reference Algoim rule of order $q = p + 12$, the total number of quadrature points in all cut elements is $n_{qp,\mathrm{cut}} = 39{,}900$. When we use the Algoim rule of order $q = p+4$, we obtain 4,320 quadrature points in all cut elements. The latter number is roughly equal to the 4,176 points points generated by our interpolation-based method.

Figure 18 shows normalized spectra of the stiffness matrix when formed with our adaptive interpolation-based approach and the Algoim rules. We sort the eigenvalues by their magnitude. The largest errors occur towards the end of the spectrum where the eigenvalues are largest in magnitude. The largest error in the spectrum of the stiffness matrix integrated with our approach is 0.3%, while the largest error for Algoim is 3.8%. These errors can be interpreted in terms of the Hausdorff distance with respect to the normalized reference spectrum which is 0.0031 for our method and 0.0375 for Algoim. More importantly, we see that our quadrature approach leads to lower errors in the low modes of the spectrum than an Algoim quadrature rule with a comparable number of quadrature points. This result highlights the accuracy of our approach.

6. Summary, discussion and conclusions

In this work, we presented a data-driven quadrature technique for integration of cut cells. Our technique is based on accurate moment fitting quadrature on a parametrized configuration space of circular cuts, and a consequent spline interpolation of those moments on the parameter space that characterizes these cuts in terms of the radius R, radial position z and the angular position θ. The moments are not globally smooth in terms of the cut parameterization. To satisfy the regularity considerations, we introduced a partition of the parameter space into 15 distinct cut cases, on which integrals of smooth functions are smooth functions of R, z and θ. With that we were able to interpolate the moments for all 15 cut cases to an accuracy in the range of $\mathcal{O}(10^{-12})$ to $\mathcal{O}(10^{-8})$. In order to capture small geometric features, we adapted the well-known adaptive quadrature concept based on quadtree subdivision of a cell.

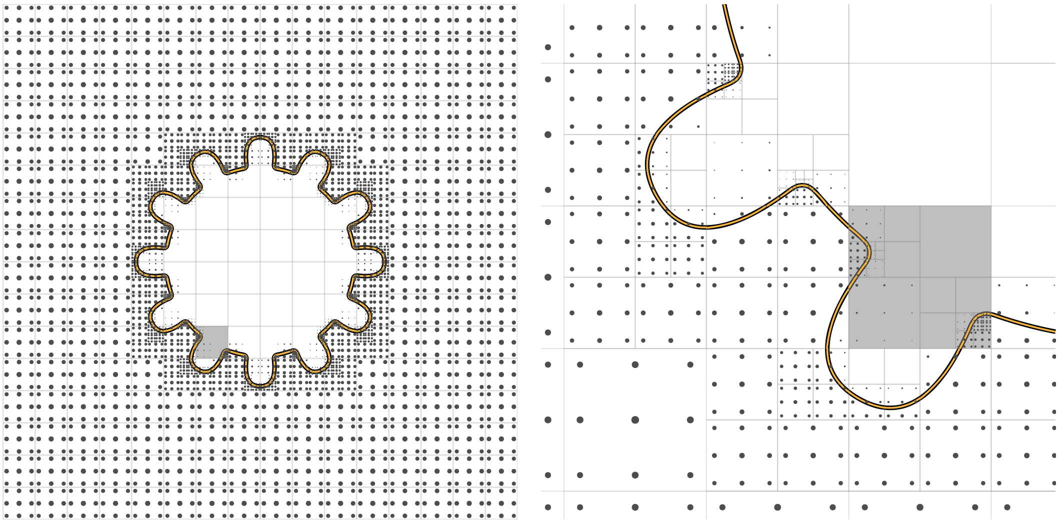


Figure 17: Gear-shaped cut contour on an embedding square $[-2,2]^2$ discretized with 16 elements in each direction. The quadrature rule is adaptively refined in each cell such that $R > h$ and thus our proposed approach is applicable.

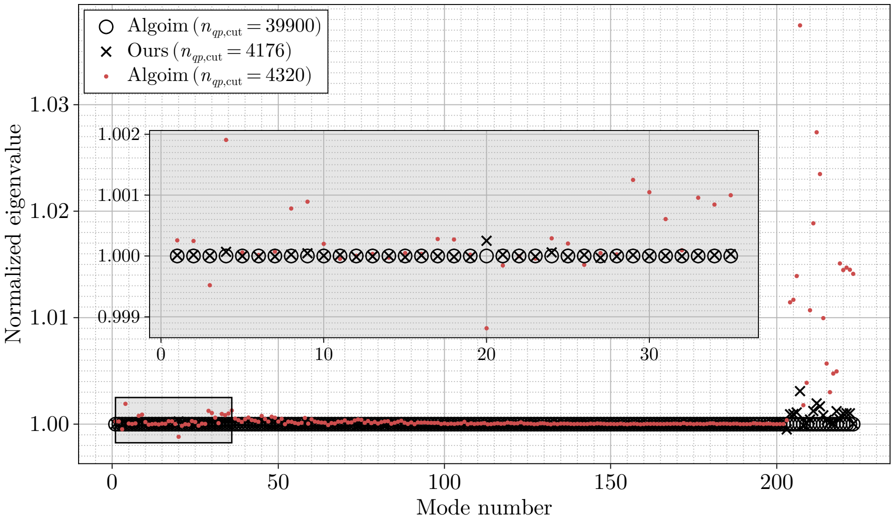


Figure 18: Stiffness matrix spectra in the gear example normalized with respect to a reference spectrum computed with an Algoim quadrature rule of order $(p + 12)$. The normalized spectrum of our method is plotted against the normalized spectrum obtained with Algoim and a comparable number of points (quadrature rule of order $(p + 4)$).

Potential advantages with respect to other cut-cell quadrature approaches

The accuracy of our interpolation-based quadrature approach was successfully validated in comparison to the state-of-the-art cut-cell quadrature library Algoim [54] for several test cases. In particular, we showed that our data-driven quadrature approach is more accurate than an Algoim [54] quadrature rule with a similar number of quadrature points. The high accuracy of our approach is a direct result of using spline interpolants devised from moments that are evaluated to high precision. Additionally, our approach enables fine-grained control over quadrature accuracy by adapting the spline spaces employed in the interpolation of the moments.

Unlike cut-cell quadrature techniques that generate rules for each cut cell individually – for example by recursive cell subdivision [48, 57], explicit reconstruction of the cut at the quadrature point level [27, 36, 59], individual moment fitting for cut cell [33, 45, 61], or recursive conversion to a graph of a height function and root finding [54] – our approach shifts expensive setup steps to an offline stage. Once the interpolants of moments are constructed, the generation of a quadrature rule reduces to inexpensive evaluations of tensor-product splines. The evaluations scale with $\mathcal{O}(N^{4/3})$, where N is the dimension of the interpolation space. Our approach therefore avoids expensive geometric operations or root finding.

In addition, our approach still results in quadrature rules on a Cartesian product stencil. It therefore enables straightforward and streamlined application of fast formation and assembly techniques, such as sum factorization, which outperform standard elementwise assembly procedures [2, 10, 31]. It does not rely on hybrid approaches that combine fast formation and assembly strategies for regions that do not intersect the cut contour and elementwise procedures for the cut elements [43, 44]. Moreover, the reuse of already available level-set evaluations from the cut-cell identification in parameter estimation also contributes to efficiency, since no additional high-cost geometry processing is required.

Practical applications

While the present work is primarily intended as a first methodological study, we envision that our technique could be promising in the context of large-scale cut finite element simulations. In particular, our approach could be beneficial where cut elements are abundant and evolving, such as problems in multiphase flow, fluid–structure interaction with moving boundaries, or topology optimization with evolving domains. In these contexts, the efficiency gain of our data-driven, spline-based quadrature generation, combined with sum-factorized finite element assembly, can lead to a significant reduction in overall computational cost. Moreover, the flexibility to adjust spline spaces provides a mechanism for balancing accuracy and efficiency depending on the application.

For practical adoption in combination with fast formation techniques, the present method requires an extension to non-smooth cuts. This extension would enable handling of sharp corners accurately and efficiently. Figure 19 shows that accurate representation of straight cuts and sharp corners currently requires adaptive refinement which precludes the application of sum factorization. The formulation of suitable parametrizations for straight cuts and sharp corners would reduce the number of quadrature points significantly and, at the same time, enable fast formation by sum factorization.

Furthermore, we have shown that moment-fitted quadrature rules using our data-driven approach exhibit high accuracy, since the interpolants approximate nearly exact moments. The efficiency of our method would greatly benefit from careful assessment and control of the quadrature accuracy versus the discretization error. Clarification of this question would also shed more light on the efficiency of our approach in comparison to other established methods, in particular in the context of finite element assembly using sum factorization and standard elementwise techniques.

Future work

In the future, we plan to extend the proposed approach to boundary integrals and cut cells in three dimensions. We also would like to explore a significant reduction of the size of the spline interpolants at a given accuracy level by optimizing the spaces for the interpolation of the moments. In addition, the versatility and robustness of our approach can be further improved by introducing a parametrization for straight cuts, which we currently approximate by circles of large radius, and for sharp corners.

CRediT authorship contribution statement

Michał Ł. Mika: Writing – original draft, Methodology, Investigation, Formal analysis, Software, Visualization. **René R. Hiemstra:** Writing – review & editing, Conceptualization, Methodology, Investigation, Software, Supervision. **Stein K.F. Stoter:** Writing – Review & Editing, Methodology. **Dominik Schillinger:** Writing – Review & Editing, Supervision, Resources, Project administration, Funding acquisition.

Acknowledgements

The results presented in this work were achieved as part of the ERC Starting grant project ‘ImageToSim’ that has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement no. 759001). The authors also gratefully acknowledge financial support from the German Research Foundation (Deutsche Forschungsgemeinschaft) through the DFG Emmy Noether Grant SCH 1249/2-1 (project number 326309100). René Hiemstra acknowledges funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101105786.

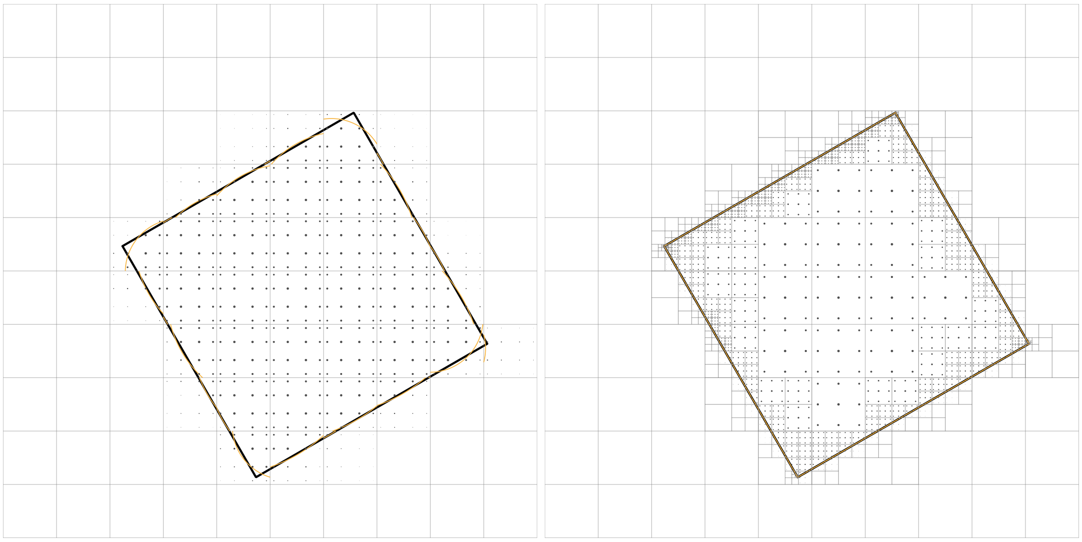


Figure 19: A unit square positioned at $(0.13, -0.09)$ and rotated by an angle of $\pi/6$. The embedding domain $[-1, 1]^2$ is discretized by 10×10 elements. Approximating the contour with circles of radius $R \leq 3$ results in significant geometric errors, as illustrated on the left side of the figure. The error in the area measure is 2.9×10^{-3}, which is to be viewed with caution, as the contour is significantly misrepresented. As shown on the right, adaptive refinement can be used to accurately represent sharp corners and straight edges. For a second order quadrature rule, the uniform mesh results in 342 quadrature points, whereas the adaptive refinement leads to 3132 quadrature points. Using the adaptively refined quadrature rule, the error in the area measure is 1.15×10^{-5}. Application of sum factorization is not possible with adaptively refined rules.

Declarations

The authors declare that they have no conflicts of interest.

The open-source code used in this work can be found at `https://github.com/SuiteSplines/QuadLearn.jl`.

References

[1] Annavarapu, C., Hautefeuille, M., Dolbow, J.E., 2012. A robust nitsche's formulation for interface problems. Computer Methods in Applied Mechanics and Engineering 225, 44–54.

[2] Antolin, P., Buffa, A., Calabrò, F., Martinelli, M., Sangalli, G., 2015. Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization. Computer Methods in Applied Mechanics and Engineering 285, 817–828.

[3] Badia, S., Caicedo, M.A., Martín, A.F., Principe, J., 2021. A robust and scalable unfitted adaptive finite element framework for nonlinear solid mechanics. Computer Methods in Applied Mechanics and Engineering 386, 114093.

[4] Badia, S., Neiva, E., Verdugo, F., 2022. Linking ghost penalty and aggregated unfitted methods. Computer Methods in Applied Mechanics and Engineering 388, 114232.

[5] Bergbauer, M., Munch, P., Wall, W.A., Kronbichler, M., 2025. High-performance matrix-free unfitted finite element operator evaluation. SIAM Journal on Scientific Computing 47, B665–B689.

[6] Bog, T., Zander, N., Kollmannsberger, S., Rank, E., 2015. Normal contact with high order finite elements and a fictitious contact material. Computers & Mathematics with Applications 70, 1370–1390.

[7] de Boor, C., 2001. A practical guide to splines. Springer, New York.

[8] Breitenberger, M., Apostolatos, A., Philipp, B., Wüchner, R., Bletzinger, K.U., 2015. Analysis in computer aided design: Nonlinear isogeometric b-rep analysis of shell structures. Computer Methods in Applied Mechanics and Engineering 284, 401–457.

[9] Bremer, J., Gimbutas, Z., Rokhlin, V., 2010. A nonlinear optimization procedure for generalized gaussian quadratures. SIAM Journal on Scientific Computing 32, 1761–1788.

[10] Bressan, A., Takacs, S., 2019. Sum factorization techniques in isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 352, 437–460.

[11] Bui, H.G., Schillinger, D., Meschke, G., 2020. Efficient cut-cell quadrature based on moment fitting for materially nonlinear analysis. Computer Methods in Applied Mechanics and Engineering 366, 113050.

[12] Burman, E., 2010. Ghost penalty. Comptes Rendus. Mathématique 348, 1217–1220.

[13] Burman, E., Claus, S., Hansbo, P., Larson, M.G., Massing, A., 2015. Cutfem: Discretizing geometry and partial differential equations. International Journal for Numerical Methods in Engineering 104, 472–501.

[14] Burman, E., Hansbo, P., 2012. Fictitious domain finite element methods using cut elements: Ii. a stabilized nitsche method. Applied Numerical Mathematics 62, 328–341.

[15] Burman, E., Hansbo, P., Larson, M.G., 2022. Cutfem based on extended finite element spaces. Numerische Mathematik 152, 331–369.

[16] Burman, E., Hansbo, P., Larson, M.G., Zahedi, S., 2025. Cut finite element methods. Acta Numerica 34, 1–121.

[17] Chu, B.D., Martin, F., Reif, U., 2022. Stabilization of spline bases by extension. Advances in Computational Mathematics 48, 23.

[18] Cottrell, J.A., Hughes, T.J., Bazilevs, Y., 2009. Isogeometric analysis: toward integration of CAD and FEA. John Wiley & Sons.

[19] Daviet, G., Shen, T., Sharp, N., Levin, D.I., 2025. Neurally integrated finite elements for differentiable elasticity on evolving domains. ACM Trans. Graph. 44.

[20] de Prenter, F., Lehrenfeld, C., Massing, A., 2018. A note on the stability parameter in nitsche's method for unfitted boundary value problems. Computers & Mathematics with Applications 75, 4322–4336.

[21] de Prenter, F., Verhoosel, C., van Zwieten, G., van Brummelen, E., 2017. Condition number analysis and preconditioning of the finite cell method. Computer Methods in Applied Mechanics and Engineering 316, 297–327.

[22] de Prenter, F., Verhoosel, C.V., van Brummelen, E.H., Larson, M.G., Badia, S., 2023. Stability and conditioning of immersed finite element methods: Analysis and remedies. Archives of Computational Methods in Engineering 30, 3617–3656.

[23] Duczek, S., Gabbert, U., 2015. Efficient integration method for fictitious domain approaches. Computational Mechanics 56, 725–738.

[24] Düster, A., Parvizian, J., Yang, Z., Rank, E., 2008. The finite cell method for three-dimensional problems of solid mechanics. Computer Methods in Applied Mechanics and Engineering 197, 3768–3782.

[25] Embar, A., Dolbow, J., Harari, I., 2010. Imposing dirichlet boundary conditions with nitsche's method and spline-based finite elements. International journal for numerical methods in engineering 83, 877–898.

[26] Engquist, B., Tornberg, A.K., Tsai, R., 2005. Discretization of dirac delta functions in level set methods. Journal of Computational Physics 207, 28–51.

[27] Fries, T.P., Omerović, S., 2016. Higher-order accurate integration of implicit geometries. International Journal for Numerical Methods in Engineering 106, 323–371.

[28] Garhuom, W., Düster, A., 2022. Non-negative moment fitting quadrature for cut finite elements and cells undergoing large deformations. Computational Mechanics 70, 1059–1081.

[29] Guo, Y., Heller, J., Hughes, T.J., Ruess, M., Schillinger, D., 2018. Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the step exchange format. Computer Methods in Applied Mechanics and Engineering 336, 39–79.

[30] Hiemstra, R., Schillinger, D., 2022. Learning cut-cell integration by means of deep neural networks. The 8th European Congress on Computational Methods in Applied Sciences and Engineering.

[31] Hiemstra, R.R., Sangalli, G., Tani, M., Calabrò, F., Hughes, T.J., 2019. Fast formation and assembly of finite element matrices with application to isogeometric linear elasticity. Computer Methods in Applied Mechanics and Engineering 355, 234–260.

[32] Höllig, K., Reif, U., Wipper, J., 2001. Weighted extended b-spline approximation of dirichlet problems. SIAM Journal on Numerical Analysis 39, 442–462.

[33] Hubrich, S., Di Stolfo, P., Kudela, L., Kollmannsberger, S., Rank, E., Schröder, A., Düster, A., 2017. Numerical integration of discontinuous functions: moment fitting and smart octree. Computational Mechanics 60, 863–881.

[34] Jomo, J., de Prenter, F., Elhaddad, M., D'Angella, D., Verhoosel, C., Kollmannsberger, S., Kirschke, J., Nübel, V., van Brummelen, E., Rank, E., 2019. Robust and parallel scalable iterative solutions for large-scale finite cell analyses. Finite Elements in Analysis and Design 163, 14–30.

[35] Konyukhov, A., Lorenz, C., Schweizerhof, K., 2015. Various contact approaches for the finite cell method. Computational Mechanics 56, 331–351.

[36] Kudela, L., Zander, N., Bog, T., Kollmannsberger, S., Rank, E., 2015. Efficient and accurate numerical quadrature for immersed boundary methods. Advanced Modeling and Simulation in Engineering Sciences 2, 10.

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

[38] Legrain, G., 2021. Non-negative moment fitting quadrature rules for fictitious domain methods. Computers & Mathematics with Applications 99, 270–291.

[39] Leidinger, L., Breitenberger, M., Bauer, A., Hartmann, S., Wüchner, R., Bletzinger, K.U., Duddeck, F., 2019. Explicit dynamic isogeometric b-rep analysis of penalty-coupled trimmed nurbs shells. Computer Methods in Applied Mechanics and Engineering 351, 891–927.

[40] Ma, J., Rokhlin, V., Wandzura, S., 1996. Generalized gaussian quadrature rules for systems of arbitrary functions. SIAM Journal on Numerical Analysis 33, 971–996.

[41] Martorell, P.A., Badia, S., 2024. High order unfitted finite element discretizations for explicit boundary representations. Journal of Computational Physics 511, 113127.

[42] Marussig, B., Hiemstra, R., Hughes, T.J., 2018. Improved conditioning of isogeometric analysis matrices for trimmed geometries. Computer Methods in Applied Mechanics and Engineering 334, 79–110.

[43] Marussig, B., Hiemstra, R., Schillinger, D., 2023. Fast immersed boundary method based on weighted quadrature. Computer Methods in Applied Mechanics and Engineering 417, 116397.

[44] Meßmer, M., Teschemacher, T., Leidinger, L.F., Wüchner, R., Bletzinger, K.U., 2022. Efficient cad-integrated isogeometric analysis of trimmed solids. Computer Methods in Applied Mechanics and Engineering 400, 115584.

[45] Müller, B., Kummer, F., Oberlack, M., 2013. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. International Journal for Numerical Methods in Engineering 96, 512–528.

[46] Nguyen, L., Stoter, S., Baum, T., Kirschke, J., Ruess, M., Yosibash, Z., Schillinger, D., 2017. Phase-field boundary conditions for the voxel finite cell method: Surface-free stress analysis of ct-based bone structures. International Journal for Numerical Methods in Biomedical Engineering 33, e2880. E2880 cnm.2880.

[47] Nguyen, L.H., Stoter, S.K., Ruess, M., Sanchez Uribe, M.A., Schillinger, D., 2018. The diffuse nitsche method: Dirichlet constraints on phase-field boundaries. International Journal for Numerical Methods in Engineering 113, 601–633.

[48] Parvizian, J., Düster, A., Rank, E., 2007. Finite cell method. Computational Mechanics 41, 121–133.

[49] Parvizian, J., Düster, A., Rank, E., 2012. Topology optimization using the finite cell method. Optimization and Engineering 13, 57–78.

[50] Poluektov, M., Łukasz Figiel, 2022. A cut finite-element method for fracture and contact problems in large-deformation solid mechanics. Computer Methods in Applied Mechanics and Engineering 388, 114234.

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

[52] Ruess, M., Schillinger, D., Bazilevs, Y., Varduhn, V., Rank, E., 2013. Weakly enforced essential boundary conditions for nurbs-embedded and trimmed nurbs geometries on the basis of the finite cell method. International Journal for Numerical Methods in Engineering 95, 811–846.

[53] Ruess, M., Schillinger, D., Özcan, A.I., Rank, E., 2014. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. Computer Methods in Applied Mechanics and Engineering 269, 46–71.

[54] Saye, R.I., 2022. High-order quadrature on multi-component domains implicitly defined by multivariate polynomials. Journal of Computational Physics 448, 110720.

[55] Schillinger, D., Dede, L., Scott, M.A., Evans, J.A., Borden, M.J., Rank, E., Hughes, T.J., 2012a. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of nurbs, immersed boundary methods, and t-spline cad surfaces. Computer Methods in Applied Mechanics and Engineering 249, 116–150.

[56] Schillinger, D., Harari, I., Hsu, M.C., Kamensky, D., Stoter, S.K., Yu, Y., Zhao, Y., 2016. The non-symmetric nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. Computer Methods in Applied Mechanics and Engineering 309, 625–652.

[57] Schillinger, D., Ruess, M., 2015. The finite cell method: A review in the context of higher-order structural analysis of cad and image-based geometric models. Archives of Computational Methods in Engineering 22, 391–455.

[58] Schillinger, D., Ruess, M., Zander, N., Bazilevs, Y., Düster, A., Rank, E., 2012b. Small and large deformation analysis with the p-and b-spline versions of the finite cell method. Computational Mechanics 50, 445–478.

[59] Stavrev, A., Nguyen, L.H., Shen, R., Varduhn, V., Behr, M., Elgeti, S., Schillinger, D., 2016. Geometrically accurate, efficient, and flexible quadrature techniques for the tetrahedral finite cell method. Computer Methods in Applied Mechanics and Engineering 310, 646–673.

[60] Stoter, S.K., Divi, S.C., van Brummelen, E.H., Larson, M.G., de Prenter, F., Verhoosel, C.V., 2023. Critical time-step size analysis and mass scaling by ghost-penalty for immersogeometric explicit dynamics. Computer Methods in Applied Mechanics and Engineering 412, 116074.

[61] Sudhakar, Y., Wall, W.A., 2013. Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods. Computer Methods in Applied Mechanics and Engineering 258, 39–54.

[62] Varduhn, V., Hsu, M.C., Ruess, M., Schillinger, D., 2016. The tetrahedral finite cell method: Higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes. International Journal for Numerical Methods in Engineering 107, 1054–1079.

[63] Wang, Y., Lai, C.Y., 2024. Multi-stage neural networks: Function approximator of machine precision. Journal of Computational Physics 504, 112865.

[64] Wegert, Z.J., Manyer, J., Mallon, C.N., Badia, S., Challis, V.J., 2025. Level-set topology optimisation with unfitted finite elements and automatic shape differentiation. Computer Methods in Applied Mechanics and Engineering 445, 118203.

[65] Yamaguchi, Y., Moriguchi, S., Terada, K., 2021. Extended b-spline-based implicit material point method. International Journal for Numerical Methods in Engineering 122, 1746–1769.

A. Partitioning of the circular cut parameter space

The analysis of the parameter space of the circular cuts of radius R revealed 15 distinct smooth parameter partitions valid under the assumption that the cell edge size h is smaller than R. The following characteristic angular positions θ and radial positions z have been identified:

Case	Radial positions	Angular positions	Case	Radial positions	Angular positions
1	$z_2 \le z \le z_1$	$\theta_1 \le \theta \le \pi/4$	9	$z_6 \le z < z_5$	$\theta_6 \le \theta \le \pi/4$
2	$z_3 \le z < z_2$	$\theta_3 \le \theta < \theta_1$	10	$z_7 \le z < z_6$	$0 \le \theta < \theta_7$
3	$z_3 \le z < z_2$	$\theta_1 \le \theta \le \pi/4$	11	$z_7 \le z < z_6$	$\theta_7 \le \theta < \theta_6$
4	$z_4 \le z < z_3$	$0 \le \theta < \theta_1$	12	$z_7 \le z < z_6$	$\theta_6 \le \theta \le \pi/4$
5	$z_4 \le z < z_3$	$\theta_1 \le \theta \le \pi/4$	13	$z_8 \le z < z_7$	$0 \le \theta < \theta_6$
6	$z_5 \le z < z_4$	$0 \le \theta < \theta_4$	14	$z_8 \le z < z_7$	$\theta_6 \le \theta \le \pi/4$
7	$z_5 \le z < z_4$	$\theta_4 \le \theta \le \pi/4$	15	$z_9 \le z < z_8$	$\theta_8 \le \theta \le \pi/4$
8	$z_6 \le z < z_5$	$0 \le \theta < \theta_6$			

Table 2
Ranges for the radial position z and the angular position θ for each parameter partition representing a smooth cut case.

$$
\begin{aligned}
z_1 &= R + \frac{d}{2}\\
z_2 &= \frac{h}{2\sin(\theta_2)}\\
z_3 &= R + \frac{h}{2}\\
z_4 &= \sqrt{R^2 - \frac{h^2}{4}} + \frac{h}{2}\\
z_5 &= \sqrt{R^2 - \frac{d^2}{4}}\\
z_6 &= \frac{h}{2\sin(\theta_5)}\\
z_7 &= R - \frac{h}{2}\\
z_8 &= \sqrt{R^2 - \frac{h^2}{4}} - \frac{h}{2}\\
z_9 &= R - \frac{d}{2}
\end{aligned}
$$

$$
\begin{aligned}
\theta_1 &= \arcsin\left(\frac{z^2 + h^2/2 - R^2}{zh\sqrt{2}}\right) - \frac{\pi}{4}\\
\theta_2 &= \arctan\left(\frac{h}{2R + h}\right)\\
\theta_3 &= \arccos\left(\frac{R + h/2}{z}\right)\\
\theta_4 &= \arcsin\left(\frac{R^2 - z^2 - h^2/2}{zh\sqrt{2}}\right) + \frac{\pi}{4}\\
\theta_5 &= \arctan\left(\frac{h}{2R - h}\right)\\
\theta_6 &= \arccos\left(\frac{R^2 - z^2 - h^2/2}{zh\sqrt{2}}\right) - \frac{\pi}{4}\\
\theta_7 &= \arccos\left(\frac{R - h/2}{z}\right)\\
\theta_8 &= \arcsin\left(\frac{R^2 - z^2 - h^2/2}{zh\sqrt{2}}\right) - \frac{\pi}{4}
\end{aligned}
$$

The definitions of the characteristic points are summarized in Table A.

B. Response surfaces of moments

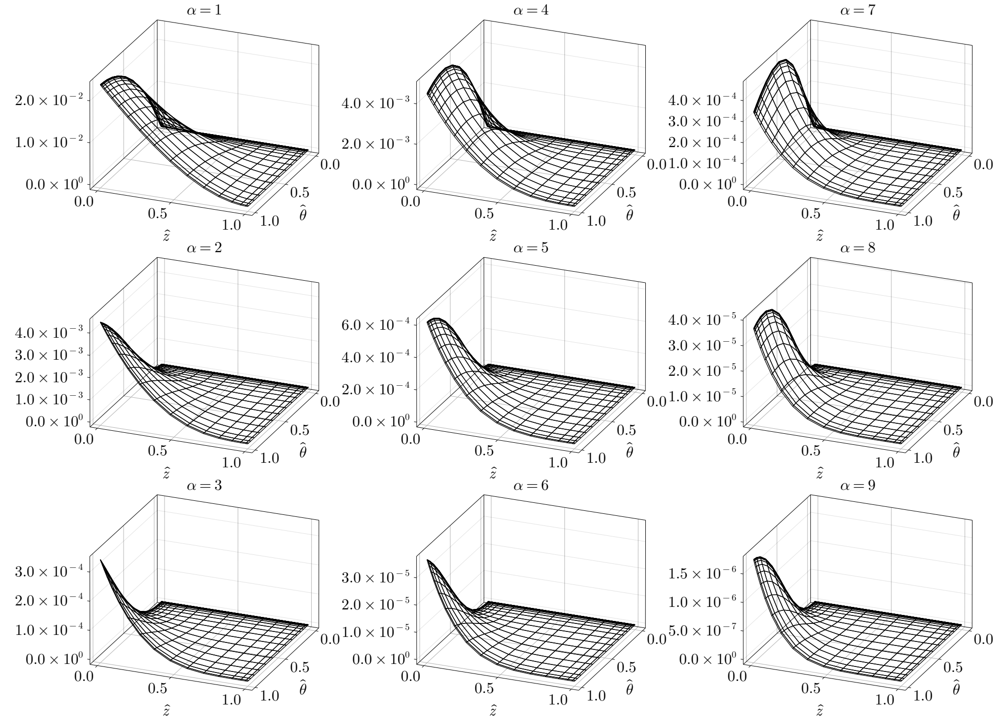


Figure 20: Response surface of moments $\int_{\Omega} \phi_{\alpha} \, \mathrm{d}\boldsymbol{x}$, $\alpha = 1, \ldots, (q+1)^2$, in terms of the normalized radial position $\hat{z} \in [0, 1]$ and the normalized angular position $\hat{\theta} \in [0, 1]$ in cut case one for a fixed radius $R = 5$ and the quadrature order $q = 2$.

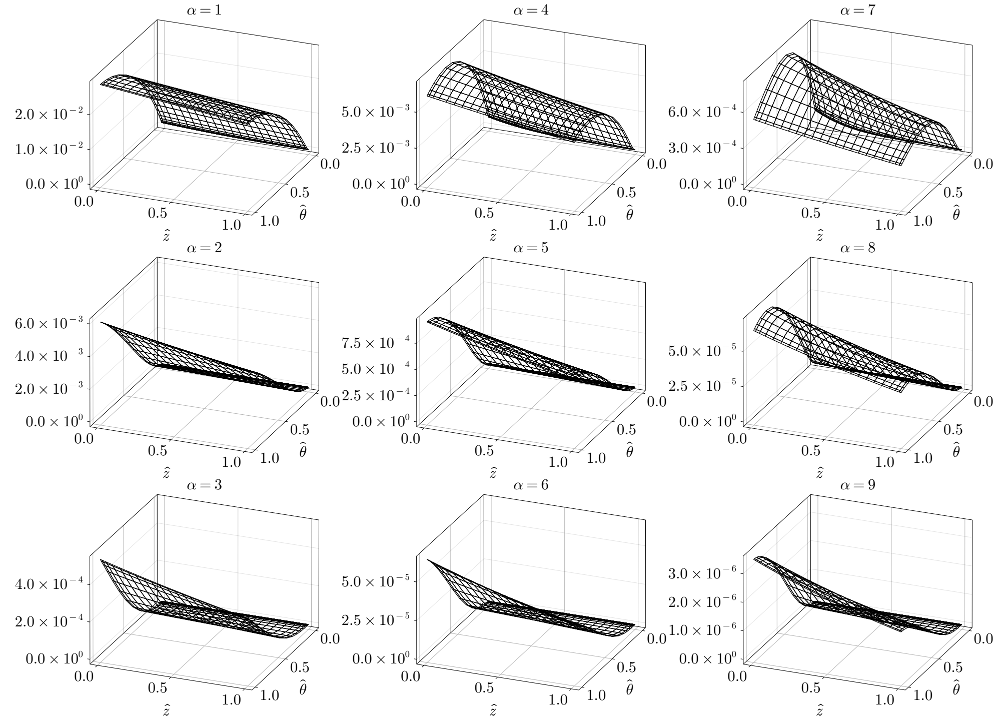


Figure 21: Response surface of moments $\int_{\Omega} \phi_{\alpha} \, \mathrm{d}\boldsymbol{x}$, $\alpha = 1, \ldots, (q+1)^2$, in terms of the normalized radial position $\hat{z} \in [0, 1]$ and the normalized angular position $\hat{\theta} \in [0, 1]$ in cut case three for a fixed radius $R = 5$ and the quadrature order $q = 2$.

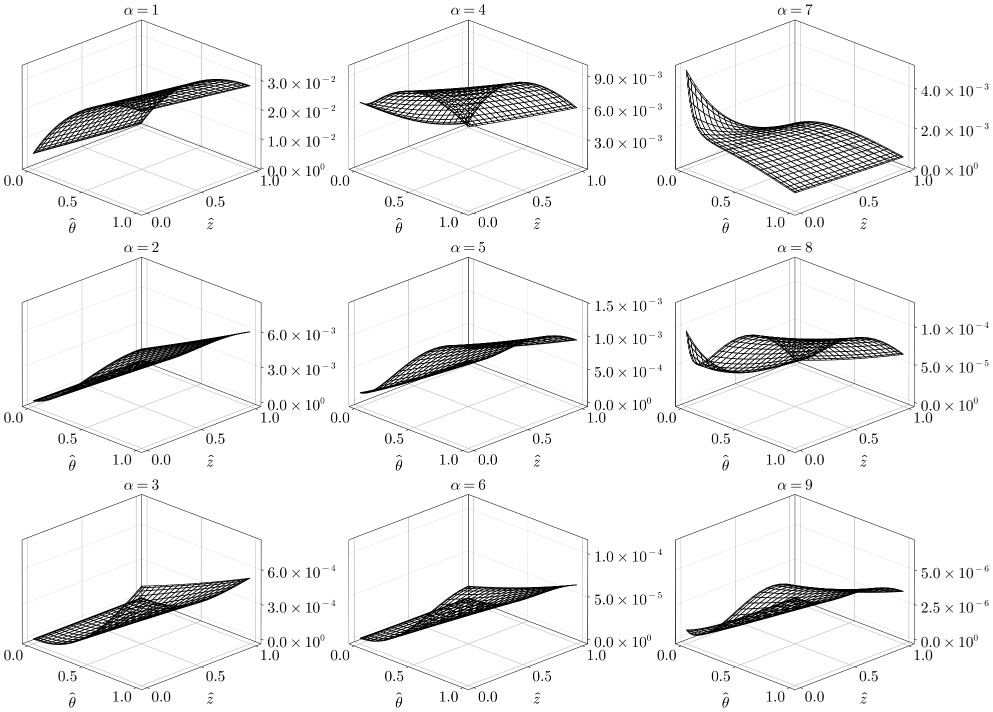

$\alpha = 1$ $\alpha = 4$ $\alpha = 7$ $\alpha = 2$ $\alpha = 5$ $\alpha = 8$ $\alpha = 3$ $\alpha = 6$ $\alpha = 9$

Figure 22: Response surface of moments $\int_{\Omega} \phi_{\alpha} \, \mathrm{d}\boldsymbol{x}$, $\alpha = 1, \ldots, (q+1)^2$, in terms of the normalized radial position $\hat{z} \in [0, 1]$ and the normalized angular position $\hat{\theta} \in [0, 1]$ in cut case five for a fixed radius $R = 5$ and the quadrature order $q = 2$.

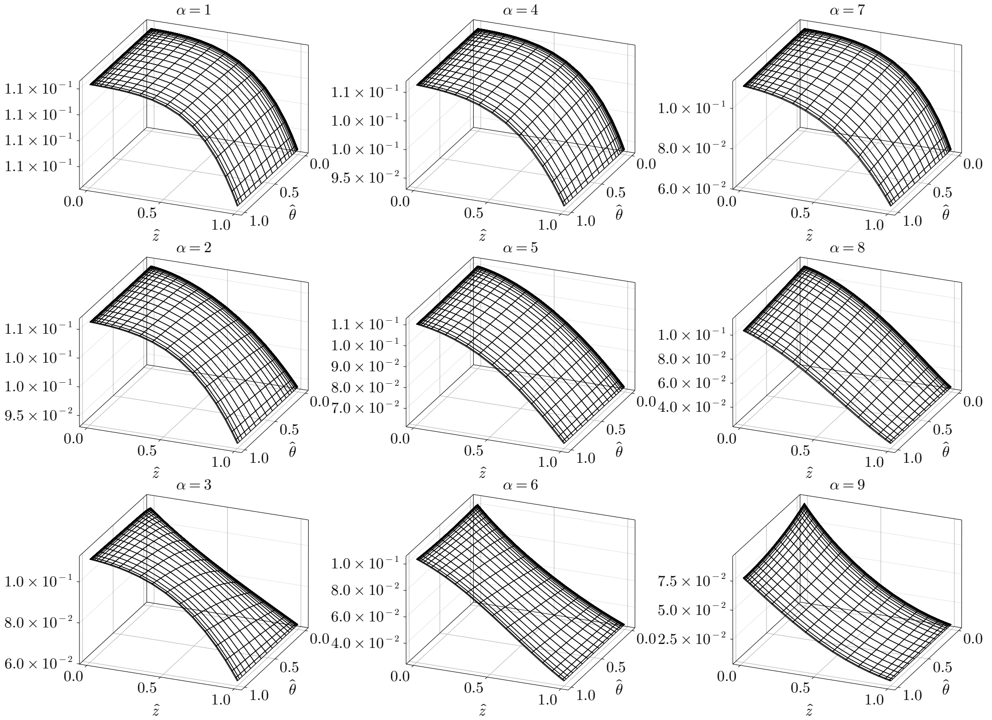

$\alpha = 1$ $\alpha = 4$ $\alpha = 7$ $\alpha = 2$ $\alpha = 5$ $\alpha = 8$ $\alpha = 3$ $\alpha = 6$ $\alpha = 9$

Figure 23: Response surface of moments $\int_{\Omega} \phi_{\alpha} \, \mathrm{d}\boldsymbol{x}$, $\alpha = 1, \ldots, (q+1)^2$, in terms of the normalized radial position $\hat{z} \in [0, 1]$ and the normalized angular position $\hat{\theta} \in [0, 1]$ in cut case nine for a fixed radius $R = 5$ and the quadrature order $q = 2$.