

# Efficient quadrature rules for subdivision surfaces in isogeometric analysis

Pieter J. Barendrecht<sup>a</sup>, Michael Bartoň<sup>b,\*</sup>, Jirí Kosinka<sup>a</sup>

<sup>a</sup>*Bernoulli Institute, University of Groningen, Nijenborgh 9, 9747 AG, Groningen, the Netherlands*

<sup>b</sup>*BCAM – Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Basque Country, Spain*

---

## Abstract

We introduce a new approach to numerical quadrature on geometries defined by subdivision surfaces based on quad meshes in the context of isogeometric analysis. Starting with a sparse control mesh, the subdivision process generates a sequence of finer and finer quad meshes that in the limit defines a smooth subdivision surface, which can be of any manifold topology. Traditional approaches to quadrature on such surfaces rely on per-quad integration, which is inefficient and typically also inaccurate near vertices where other than four quads meet. Instead, we explore the space of possible groupings of quads and identify the optimal macro-quads in terms of the number of quadrature points needed. We show that macro-quads consisting of quads from one or several consecutive levels of subdivision considerably reduce the cost of numerical integration. Our rules possess a tensor product structure and the underlying univariate rules are Gaussian, i.e., they require the minimum possible number of integration points in both univariate directions.

The optimal quad groupings differ depending on the particular application. For instance, computing surface areas, volumes, or solving the Laplace problem lead to different spline spaces with specific structures in terms of degree and continuity. We show that in most cases the optimal groupings are quad-strips consisting of  $(1 \times n)$  quads, while in some cases a special macro-quad spanning more than one subdivision level offers the most economical integration.

Additionally, we extend existing results on exact integration of subdivision splines. This allows us to validate our approach by computing surface areas and volumes with known exact values. We demonstrate on several examples that our quadratures use fewer quadrature points than traditional quadratures. We illustrate our approach to subdivision spline quadrature on the well-known Catmull-Clark scheme based on bicubic splines, but our ideas apply also to subdivision schemes of arbitrary bidegree, including non-uniform and hierarchical variants. Specifically, we address the problems of computing areas and volumes of Catmull-Clark subdivision surfaces, as well as solving the Laplace and Poisson PDEs defined over planar unstructured quadrilateral meshes in the context of isogeometric analysis.

*Keywords:* Numerical integration, subdivision surface, non-tensor-product splines, Gaussian quadrature rules, isogeometric analysis.

---

## 1. Introduction

Subdivision surfaces [39] are a popular modelling tool due to their ability to represent shapes of arbitrary manifold topology and are the representation of choice in 3D animated films [17]. The most popular subdivision scheme is that developed by Catmull and Clark [12]. Subdivision surfaces have also been used in the context of numerical analysis. The seminal papers [13, 21] in this area are based on the subdivision scheme of Loop [30] and pre-date the advent of isogeometric analysis (IgA) [15].

---

\*Corresponding author

*Email addresses:* [p.j.barendrecht@rug.nl](mailto:p.j.barendrecht@rug.nl) (Pieter J. Barendrecht), [mbarton@bcamath.org](mailto:mbarton@bcamath.org) (Michael Bartoň), [j.kosinka@rug.nl](mailto:j.kosinka@rug.nl) (Jirí Kosinka)

The isogeometric paradigm is focused primarily on the use of tensor product B-splines and their hierarchical counterparts [18, 19], which proved extremely successful in solving PDEs on surfaces. However, due to the inherent restriction to trivial topologies when modelling with (hierarchical) tensor product constructions, (hierarchical) subdivision surfaces have recently been employed within the isogeometric paradigm [3, 4, 36, 52, 53]. New models can be designed directly using subdivision, or existing CAD models can be converted to the subdivision representation with arbitrary accuracy [46, 47].

Nevertheless, special care needs to be taken when using subdivision blending functions in IgA. First, the linear independence of these functions, a precursor for fitting and numerical analysis, is not trivial, in contrast to the tensor product case of B-splines. This is now well understood [37], including the hierarchical setting [55]. Second, as some subdivision blending functions do not admit a closed form and consist of infinitely many polynomial pieces, efficient quadrature rules remain elusive. While recent efforts [27, 34, 51] address this issue to some extent, the employed quadratures are, as we show in the present paper, not optimal. Before moving on, we note that there exist several finite patch-wise constructions [35, 44, 50] that have been used in the context of numerical analysis. Depending on the application, these might be used as an alternative to a subdivision-based approach. However, in cases when one wants to use the exact subdivision surface as designed in a 3D modeller, there is no other option than to use the presented approach.

Efficient numerical integration is an essential ingredient of IgA. A quadrature rule, or *quadrature* in short, is a computational scheme to approximate an integral by a weighted sum of function evaluations. The points where the function is evaluated are known as *nodes*. It is desired to use quadrature rules that require the minimal number of nodes (Gaussian quadrature) while guaranteeing the exactness of the integration for every function from the space under consideration. In our case, with IgA as the main application, we focus on polynomial subdivision spline spaces.

Even though complete results on existence and uniqueness of univariate Gaussian quadratures for splines were derived in the late 70s by Micchelli and Pinkus [33], the actual rules (nodes and weights) have not been known until very recently [6–8, 25]. Quasi-optimal quadratures are used frequently in the IgA community [2, 26]. The half-point rule of Hughes et al. is asymptotically Gaussian, i.e., the rule is exact and optimal as the number of elements approaches infinity. However, whenever applied to a domain with finitely many elements, the rule does not integrate exactly. To overcome this drawback, additional nodes can be added [2], but at the expense of using more nodes with potentially negative weights.

When building mass and stiffness matrices in IgA, alternative efficient integration methods have been proposed recently [9, 31, 32]. While [9] design a weighted quadrature for each row of the mass/stiffness matrix separately, [31, 32] exploit the observation that, under certain conditions, the optimal convergence rate of the linear system can be achieved despite the fact that the integration rule is not exact. In contrast to that work, we focus on quadrature rules that are *exact* for some specific spaces, that is, our quadrature gives exact integrals up to machine precision.

The homotopy continuation approach [6] can be used to derive Gaussian quadrature rules for univariate splines over finite domains. In particular, this methodology is well suited for non-uniform spline spaces. In this work, we take advantage of homotopy continuation and derive Gaussian quadrature rules for specific tensor-product spline spaces that appear when solving PDEs on subdivision surfaces based on quad meshes. In particular, we investigate the integration in the neighbourhood of *extraordinary vertices* (EVs). At these points of valency other than four, the subdivision splines do not possess the tensor product structure. We show that the number of required nodes can be in some situations reduced by grouping quads from several consecutive subdivision levels at EVs. This allows us to significantly reduce the number of nodes compared to traditional approaches while still preserving exact integration.

To validate our new quadrature rules for subdivision surfaces, we extend existing results on exact integration of subdivision splines [24] and demonstrate that our quadrature rules produce the expected exact values of areas and volumes of piecewise polynomial and subdivision surfaces [20, 22, 38]. We use the well-known scheme of Catmull-Clark [12] in our examples, but our ideas extend to other subdivision schemes based on tensor-product B-splines, including arbitrary-degree [49], non-uniform [10, 11], and (truncated) hierarchical [3, 52, 53] variants. Although our ideas apply to various model problems in the context of isogeometric analysis, we demonstrate the functionality and efficiency of our quadrature method on area and volume

computations as well as solving Laplace and Poisson PDEs on Catmull-Clark subdivision surfaces defined by (planar) unstructured quadrilateral meshes.

We start our study by summarising relevant known facts regarding subdivision splines and extend existing results on their integration (Section 2). We then focus on the univariate spline spaces (Section 3) that form the building blocks for obtaining efficient quadrature rules required in solving PDEs on Catmull-Clark subdivision surfaces (Section 4). Our new quadrature rules are numerically validated on several examples with exact solutions (Section 5). Finally, we conclude the paper and discuss avenues for future research (Section 6).

## 2. Subdivision splines and exact integration

Our goal is to find efficient and exact quadrature rules for subdivision splines. To this end, we recall existing results on subdivision splines [39] and extend known approaches to integration of subdivision splines [24], products of subdivision splines, and most importantly we address integration of products of derivatives of subdivision splines. Such integration arises in the context of IgA on subdivision surfaces. While the ideas and techniques discussed in this section generalise to other subdivision schemes, for the simplicity of argument, we focus solely on the case of Catmull-Clark subdivision.

### 2.1. Catmull-Clark subdivision surfaces

Interpreting a manifold quad mesh as the control net of a Catmull-Clark subdivision surface  $S$ , each quad corresponds to a surface patch  $S_i$  such that  $S = \bigcup_i S_i$ . Each patch  $S_i$  is parameterised on some domain  $D_i$  as  $\vec{f}_i(u, v) : D_i \rightarrow \mathbb{R}^3$ ,  $(u, v) \mapsto \vec{f}_i(u, v) = (f_i^1(u, v), f_i^2(u, v), f_i^3(u, v))$ . Using Stam’s method [48], we parameterise each patch on the unit square  $\Omega = [0, 1]^2$ .

When all four vertices of the quad corresponding to a surface patch  $S_i$  have the regular valency  $n = 4$ , the surface patch is simply a bicubic patch (and thus can be integrated as such). Otherwise, one or more vertices have valency  $n \neq 4$ , the so-called *extraordinary vertices* (EVs), and the corresponding surface patch does not have a tensor-product structure. We focus on the case where a quad contains at most one EV (which can always be established by subdividing an initial quad mesh once, or an initial polygonal mesh at most twice). In this case, the surface patch is composed of an infinite sequence of L-shaped spline regions. Considering the  $n$  quads sharing one EV, the L-shaped spline regions compose an infinite sequence of *spline rings* around the EV; see Figure 1.

The control net of a patch  $S_i$  consists of  $2n + 8$  control points which are collectively referred to as  $P_i$ ; see Figure 2. The control net can be subdivided using a  $(2n + 8) \times (2n + 8)$  subdivision matrix  $A_n$ . Using an extended  $(2n + 17) \times (2n + 8)$  matrix  $\bar{A}_n$  instead of  $A_n$ , an additional 9 new control points can be obtained (shown in yellow in Figure 2, middle). Subdividing the control net using  $\bar{A}_n$  results in a subdivided patch composed of four subpatches, three of which can be evaluated and/or integrated straight away as bicubic patches, and a fourth one for which the corresponding quad in the subdivided control net contains the (updated) EV.

The process can be repeated by first subdividing the original control net for the patch  $(l - 1)$  times using  $A_n$  followed by one more subdivision step using  $\bar{A}_n$ . The relevant control points of the three subpatches ready to be evaluated are selected using the extraction matrices  $X_n^{(k)}$  with  $k \in \{1, 2, 3\}$ ; see Figure 2, right. It follows that each subpatch  $S_i^{k,l}$  is parameterised on a square domain  $\Omega^{k,l}$  (also referred to as a tile), where here, and in the rest of the paper,  $l \in \mathbb{Z}^+$  refers to the subdivision level,  $k$  marks the tiles as shown in Figure 3, and  $n$  the valency of the EV. Note that  $S_i = \bigcup_{k,l} S_i^{k,l}$ .

In short, points on a surface subpatch  $S_i^{k,l}$  can be evaluated as

$$S_i^{k,l}(s, t) = \vec{f}_i(u, v) \Big|_{\Omega^{k,l}} = \left( X_n^{(k)} \bar{A}_n A_n^{l-1} P_i \right)^T \vec{M}(s, t), \quad (1)$$

where  $\vec{M}(s, t)$  are the bicubic uniform B-splines (i.e., the segments spanning one knot span in both parametric directions) defined on the unit square  $\Omega$ . This means that  $(u, v) \in \Omega^{k,l}$  needs to be mapped to  $(s, t)$  using a bilinear map  $m^{k,l}(u, v)$  before  $S_i^{k,l}(s, t)$  is evaluated [48]; see Figure 3, right.

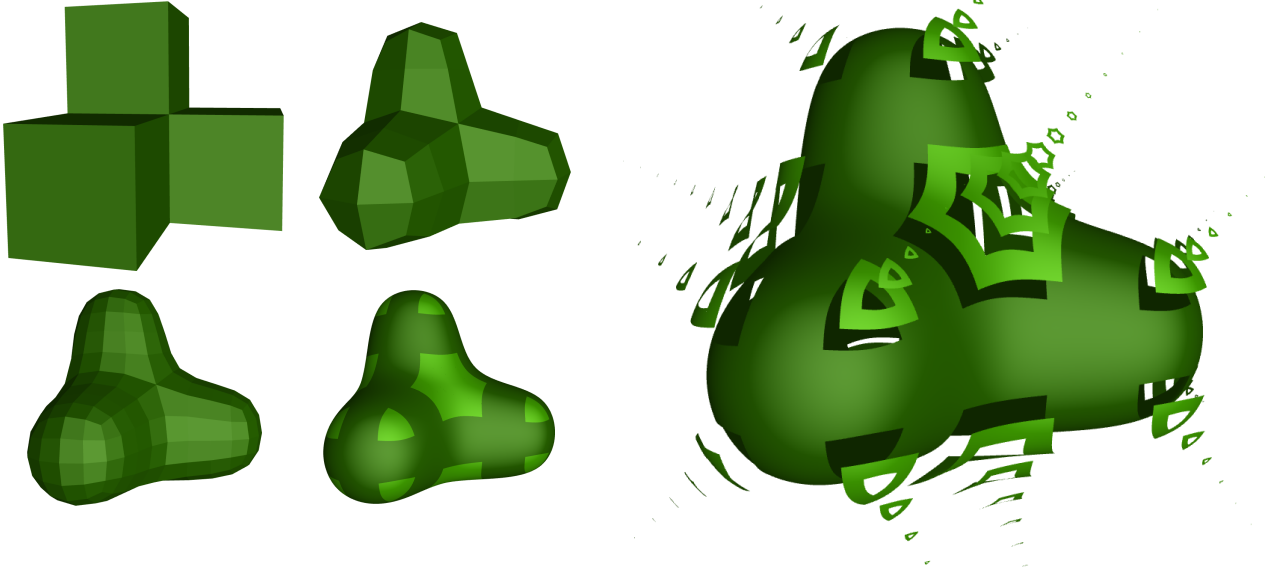


Figure 1: Initial mesh followed by two steps of Catmull-Clark subdivision (left). Limit surface (middle), composed of regular regions (dark green) and irregular regions (light green) with spline rings offset for visualisation purposes (right).

## 2.2. Subdivision splines

In the case of Catmull-Clark subdivision, the subdivision matrix  $A_n$  is non-defective for all  $n \geq 3$  [48] and can therefore be expressed as  $V_n \Lambda_n V_n^{-1}$ . Note that  $V_n$  consists of the  $2n+8$  right eigenvectors  $\vec{\alpha}_{n,p}$  (columns), also referred to as the *natural configuration(s)*, whereas  $V_n^{-1}$  consists of the  $2n+8$  left eigenvectors, also referred to as the *limit stencils*.  $\Lambda_n$  is a diagonal matrix with entries (eigenvalues)  $\lambda_{n,p}$ . It follows that  $A_n^{l-1} = V_n \Lambda_n^{l-1} V_n^{-1}$ .

We can now rewrite (1) as

$$\vec{f}_i(u, v)|_{\Omega^{k,l}} = P_i^T V_n^{-T} \Lambda_n^{l-1} \left( X_n^{(k)} \bar{A}_n V_n \right)^T \vec{M}(m^{k,l}(u, v)). \quad (2)$$

It follows that the  $2n+8$  subdivision splines  $\varphi_{n,p}(u, v)$  (i.e., the blending functions associated with the control points  $P_{i,p}$ ) are defined as

$$\varphi_{n,p}(u, v)|_{\Omega^{k,l}} = \vec{\beta}_{n,p}^T \Lambda_n^{l-1} \left( X_n^{(k)} \bar{A}_n V_n \right)^T \vec{M}(m^{k,l}(u, v)), \quad (3)$$

where  $\vec{\beta}_{n,p}$  are the columns of  $V_n^{-1}$ .

## 2.3. Eigenfunctions

We point out that  $\Lambda_n^{l-1} \left( X_n^{(k)} \bar{A}_n V_n \right)^T \vec{M}(m^{k,l}(u, v))$  is the column of functions often referred to as *eigenfunctions*  $\vec{\psi}_n(u, v)$  [24, 48]. These are the result of setting  $\vec{P} = \vec{\alpha}_{n,p}$  for  $p = 1 \dots 2n+8$ , where  $\vec{P}$  only has one component instead of the usual three components  $x$ ,  $y$  and  $z$  (and is therefore a vector and not a matrix). This way,  $\vec{P}^T V_n^{-T} = \left( V_n^{-1} \vec{P} \right)^T = \left( V_n^{-1} \vec{\alpha}_{n,p} \right)^T$ , which is a row of zeroes with a 1 at the  $p^{\text{th}}$  position. It follows that the  $p^{\text{th}}$  eigenfunction is

$$\psi_{n,p}(u, v)|_{\Omega^{k,l}} = \lambda_{n,p}^{l-1} \left( X_n^{(k)} \bar{A}_n \vec{\alpha}_{n,p} \right)^T \vec{M}(m^{k,l}(u, v)). \quad (4)$$

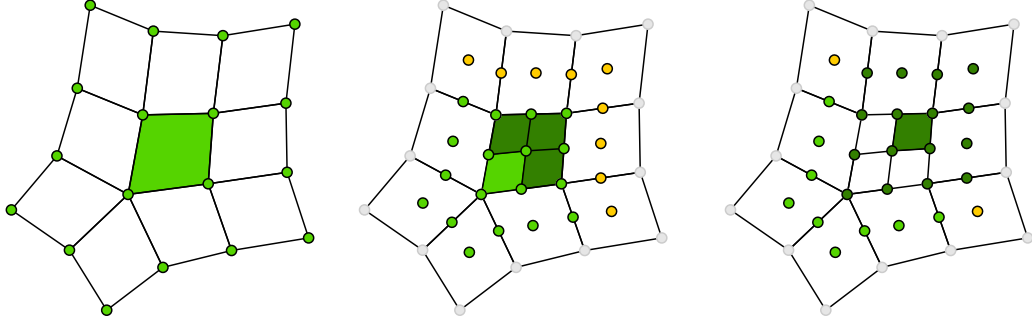


Figure 2: The  $2n + 8$  control points  $P_i$  (light green) defining a surface patch in the neighbourhood of an EV of valency  $n = 5$  (left). One step of (virtual) subdivision using  $\bar{A}_n$ , resulting in  $2n + 17$  new control points ( $2n + 8$  light green and 9 yellow) (middle). Application of  $X_n^{(k)}$ , in this case  $X_n^{(2)}$ , which extracts the 16 points (dark green) required to evaluate points on the subpatch  $S_i^{2,1}$  (right).

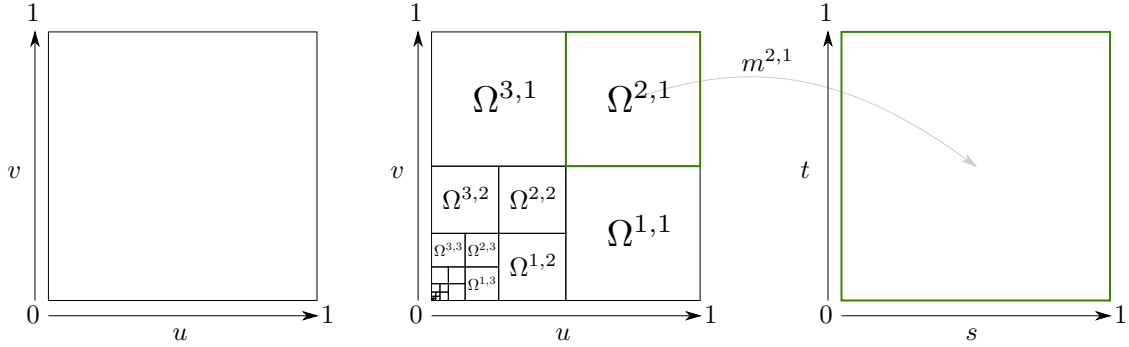


Figure 3: The unit square  $\Omega$  (left) composed of  $\Omega^{k,l}$ , where  $k \in \{1, 2, 3\}$  and  $l \in \mathbb{Z}^+$  (middle). Each subtile  $\Omega^{k,l}$  is mapped by  $m^{k,l}$  to the unit square before the subpatch  $S_i^{k,l}$  is evaluated (right).

Note that we can now define the subdivision splines as

$$\varphi_{n,p}(u, v) = \vec{\beta}_{n,p}^T \vec{\psi}_n(u, v). \quad (5)$$

In most of the expressions that follow in this section, the focus lies on eigenfunctions. Results involving subdivision splines are readily obtained by using (5).

#### 2.4. Exact integration of subdivision splines

As illustrated in Section 2.1, the surface patches around an EV are in fact infinitely piecewise bicubic. Integration on the associated domain is therefore not straightforward. In this subsection we take a closer look at such integration, which originally appeared in [24, Appendix B] and later in [42].

We start with the observation that on any tile  $\Omega^{k,l}$  the  $p^{\text{th}}$  subdivision spline (3) can be integrated exactly as it is a linear combination of the bicubic uniform B-splines:

$$\mathcal{I}_{n,p}^{k,l} \stackrel{\text{def}}{=} \iint_{\Omega^{k,l}} \varphi_{n,p}(u, v) du dv = \iint_{\Omega} \vec{\beta}_{n,p}^T \Lambda_n^{l-1} \left( X_n^{(k)} \bar{A}_n V_n \right)^T \vec{M}(s, t) |J| ds dt, \quad (6)$$

where  $|J| = \left| \frac{\partial(u,v)}{\partial(s,t)} \right|$ . From the bilinear map  $m^{k,l}(u, v)$ , which maps a square tile  $\Omega^{k,l}$  to the unit square, it follows that  $|J| = \begin{vmatrix} (\frac{1}{2})^l & 0 \\ 0 & (\frac{1}{2})^l \end{vmatrix} = (\frac{1}{4})^l = \frac{1}{4} \cdot (\frac{1}{4})^{l-1}$ .

We can now rewrite (6) as

$$\begin{aligned}\mathcal{I}_{n,p}^{k,l} &= \frac{1}{4} \left(\frac{1}{4}\right)^{l-1} \vec{\beta}_{n,p}^T \Lambda_n^{l-1} \left(X_n^{(k)} \bar{A}_n V_n\right)^T \iint_{\Omega} \vec{M}(s,t) ds dt \\ &= \frac{1}{4} \vec{\beta}_{n,p}^T \left(\frac{1}{4} \Lambda_n\right)^{l-1} \left(X_n^{(k)} \bar{A}_n V_n\right)^T \sum_{G=1}^4 \omega_{2,G} \vec{M}(s_{2,G}, t_{2,G}),\end{aligned}\quad (7)$$

where  $\omega_{2,G}$  are the integration weights and  $(s_{2,G}, t_{2,G})$  the nodes of a suitable quadrature. In the case of Catmull-Clark subdivision, one can use the  $2 \times 2$  Gauss-Legendre scheme which guarantees exactness of the integration for the space of bicubic polynomials.

The next observation is that we can integrate over all three tiles  $\Omega^{1,l}$ ,  $\Omega^{2,l}$ , and  $\Omega^{3,l}$  at once, simply by replacing  $X_n^{(k)}$  by  $\mathcal{X}_n \stackrel{\text{def}}{=} X_n^1 + X_n^2 + X_n^3$  in the expression above.

Finally, to integrate over all levels  $l \in (1, \infty]$  at once, i.e., to integrate over  $\Omega$ , we write

$$\begin{aligned}\mathcal{J}_{n,p} &\stackrel{\text{def}}{=} \iint_{\Omega} \varphi_{n,p}(u,v) du dv \\ &= \lim_{h \rightarrow \infty} \frac{1}{4} \vec{\beta}_{n,p}^T \left( \left(\frac{1}{4} \Lambda_n\right)^0 + \left(\frac{1}{4} \Lambda_n\right)^1 + \dots + \left(\frac{1}{4} \Lambda_n\right)^h \right) (\mathcal{X}_n \bar{A}_n V_n)^T \sum_{G=1}^4 \omega_{2,G} \vec{M}(s_{2,G}, t_{2,G}) \\ &= \frac{1}{4} \vec{\beta}_{n,p}^T \left( I - \frac{1}{4} \Lambda_n \right)^{-1} (\mathcal{X}_n \bar{A}_n V_n)^T \sum_{G=1}^4 \omega_{2,G} \vec{M}(s_{2,G}, t_{2,G}).\end{aligned}\quad (8)$$

This is possible, because the sum of  $\left(\frac{1}{4} \Lambda_n\right)^h$  is a *geometric series* which is guaranteed to converge as the dominant eigenvalue (i.e., the largest value in the diagonal matrix  $\Lambda_n$ ) of  $A_n$  is 1 (which is then multiplied by  $\frac{1}{4}$ ).

#### 2.4.1. Integrating partial derivatives

Integrating the first order partial derivatives  $\frac{\partial \varphi_{n,p}(u,v)}{\partial u}$  and  $\frac{\partial \varphi_{n,p}(u,v)}{\partial v}$  of a subdivision spline  $\varphi_{n,p}(u,v)$  follows the same approach. The only notable difference is the additional factor  $\frac{\partial s}{\partial u} = \frac{\partial t}{\partial v} = 2^l = 2 \cdot 2^{l-1}$ , which results in the replacement of the factors  $\frac{1}{4}$  by  $\frac{1}{2}$  in (7) and (8). And instead of  $\vec{M}(s,t)$ , partial derivatives of  $\vec{M}(s,t)$  are now used.

In case of the second order partial derivatives, there is a factor  $4^l = 4 \cdot 4^{l-1}$ . Note that these can still be integrated exactly, because the dominant eigenvalue  $\lambda_{n,1} = 1$  in  $\Lambda_n$  is associated with the constant eigenfunction  $\psi_{n,1}(u,v)$ , whose derivatives vanish. Thus,  $\lambda_{n,1}$  can safely be replaced by a value less than 1 for this computation, and therefore  $(I - \Lambda_n)$  is still invertible.

#### 2.5. Integration of products (of derivatives)

Although exact integration of subdivision splines or their partial derivatives is possible, the practical applications of these integrals presented in the previous section are rather limited. We now take a closer look at integrals of double and triple products of the subdivision splines or their partial derivatives, which are required for area, volume (Section 5), and other computations.

##### 2.5.1. Products

We first consider the integral of the product of two subdivision splines  $\varphi_{n,p}$  and  $\varphi_{n,q}$ , initially on  $\Omega^{k,l}$  and then on  $\Omega$ . From (3) it follows that

$$\begin{aligned}\mathcal{I}_{n,p,q}^{k,l} &\stackrel{\text{def}}{=} \iint_{\Omega^{k,l}} \varphi_{n,p}(u,v) \varphi_{n,q}^T(u,v) du dv \\ &= \iint_{\Omega^{k,l}} \vec{\beta}_{n,p}^T \Lambda_n^{l-1} \left(X_n^{(k)} \bar{A}_n V_n\right)^T \vec{M}(m^{k,l}(u,v)) \vec{M}^T(m^{k,l}(u,v)) \left(X_n^{(k)} \bar{A}_n V_n\right) \Lambda_n^{l-1} \vec{\beta}_{n,q} du dv,\end{aligned}\quad (9)$$

where  $\varphi_{n,p}^T$  is interpreted as the transpose of the right-hand side of (3). Note that  $\vec{M}\vec{M}^T$  is the tensor-product of  $\vec{M}$  with itself, which is a matrix of dimension  $16 \times 16$  with its entries being bidegree 6 polynomials. We use the  $4 \times 4$  Gauss-Legendre scheme to integrate these entries exactly. Integrating on the  $(s, t)$  unit square instead of  $\Omega^{k,l}$  again results in the familiar determinant of the Jacobian  $|J| = \left(\frac{1}{4}\right)^l = \frac{1}{4} \cdot \left(\frac{1}{2}\right)^{l-1} \left(\frac{1}{2}\right)^{l-1}$ , such that

$$\mathcal{I}_{n,p,q}^{k,l} = \frac{1}{4} \vec{\beta}_{n,p}^T \left(\frac{1}{2} \Lambda_n\right)^{l-1} \left(X_n^{(k)} \bar{A}_n V_n\right)^T \left[ \sum_{G=1}^{16} \omega_{4,G} \vec{M}(s_{4,G}, t_{4,G}) \vec{M}^T(s_{4,G}, t_{4,G}) \right] \left(X_n^{(k)} \bar{A}_n V_n\right) \left(\frac{1}{2} \Lambda_n\right)^{l-1} \vec{\beta}_{n,q}. \quad (10)$$

We now consider integration on  $\Omega$ . Defining

$$\mathcal{C}_n^{(k)} \stackrel{\text{def}}{=} \left(X_n^{(k)} \bar{A}_n V_n\right)^T \left[ \sum_{G=1}^{16} \omega_{4,G} \vec{M}(s_{4,G}, t_{4,G}) \vec{M}^T(s_{4,G}, t_{4,G}) \right] \left(X_n^{(k)} \bar{A}_n V_n\right) \quad (11)$$

and the shorthand notation  $\Gamma_n \stackrel{\text{def}}{=} \frac{1}{2} \Lambda_n$ , it follows that

$$\begin{aligned} \mathcal{S}_n^{(k)} &\stackrel{\text{def}}{=} \lim_{h \rightarrow \infty} \Gamma_n^0 \mathcal{C}_n^{(k)} \Gamma_n^0 + \Gamma_n^1 \mathcal{C}_n^{(k)} \Gamma_n^1 + \dots + \Gamma_n^h \mathcal{C}_n^{(k)} \Gamma_n^h \\ &= \mathcal{C}_n^{(k)} + \Gamma_n \mathcal{S}_n^{(k)} \Gamma_n \end{aligned}$$

and thus

$$\begin{aligned} \left[\mathcal{S}_n^{(k)}\right]_{ab} &= \left[\mathcal{C}_n^{(k)}\right]_{ab} + [\Gamma_n]_{aa} [\Gamma_n]_{bb} \left[\mathcal{S}_n^{(k)}\right]_{ab} \\ &= \frac{\left[\mathcal{C}_n^{(k)}\right]_{ab}}{1 - [\Gamma_n]_{aa} [\Gamma_n]_{bb}}, \end{aligned} \quad (12)$$

where  $\left[\mathcal{S}_n^{(k)}\right]_{ab}$  refers to the entry at the  $a^{\text{th}}$  row and  $b^{\text{th}}$  column of  $\mathcal{S}_n^{(k)}$ , mutatis mutandis for  $\left[\mathcal{C}_n^{(k)}\right]_{ab}$ . Finally,

$$\mathcal{J}_{n,p,q} \stackrel{\text{def}}{=} \iint_{\Omega} \varphi_{n,p}(u, v) \varphi_{n,q}^T(u, v) du dv = \frac{1}{4} \vec{\beta}_{n,p}^T \left(\sum_{k=1}^3 \mathcal{S}_n^{(k)}\right) \vec{\beta}_{n,q}. \quad (13)$$

### 2.5.2. Products of derivatives

In the case of products of first order partial derivatives, we now have the factor  $2^l = 2 \cdot 2^{l-1}$  appearing on *both* sides of the right hand side of (10).  $\mathcal{C}_n^{(k)}$  now includes the products of the partial derivatives of  $\vec{M}$  instead of products of  $\vec{M}$  (compare with the approach taken for integrating partial derivatives of subdivision splines in Section 2.4.1). This means that we now have  $\Gamma = \Lambda_n$ . Observing that  $\left[\mathcal{S}_n^{(k)}\right]_{11} = 0$  as partial derivatives of the constant eigenfunction vanish, the first order partial derivatives are square integrable. However, the second order partial derivatives are *not* square integrable using Stam's parameterisation as the factor  $4^l = 4 \cdot 4^{l-1}$  on both sides of (10) results in  $\Gamma = 2\Lambda_n$ ; see also [40].

### 2.5.3. Triple products

In order to deal with triple products (which are required for volume computations), we now switch to tensor notation [28]. Third-order tensors are typeset using calligraphic symbols.

The first step is to rewrite the expression for integrals of (derivatives of) subdivision splines (8) from Section 2.4 as

$$\begin{aligned}\vec{\mathcal{C}}_n^{(k)} &\stackrel{\text{def}}{=} \left( X_n^{(k)} \bar{A}_n V_n \right)^T \sum_{G=1}^4 \omega_{2,G} \vec{M}(s_{2,G}, t_{2,G}), \\ \Gamma_n &\stackrel{\text{def}}{=} \frac{1}{4} \Lambda_n, \\ \vec{\mathcal{S}}_n^{(k)} &\stackrel{\text{def}}{=} \lim_{h \rightarrow \infty} \Gamma_n^0 \vec{\mathcal{C}}_n^{(k)} + \Gamma_n^1 \vec{\mathcal{C}}_n^{(k)} + \dots + \Gamma_n^h \vec{\mathcal{C}}_n^{(k)} \\ &= \vec{\mathcal{C}}_n^{(k)} + \Gamma_n \vec{\mathcal{S}}_n^{(k)},\end{aligned}$$

and thus

$$\left[ \vec{\mathcal{S}}_n^{(k)} \right]_a = \frac{\left[ \vec{\mathcal{C}}_n^{(k)} \right]_a}{1 - [\Gamma_n]_{aa}}, \quad (14)$$

$$\mathcal{J}_{n,p} = \frac{1}{4} \vec{\beta}_{n,p}^T \left( \sum_{k=1}^3 \vec{\mathcal{S}}_n^{(k)} \right). \quad (15)$$

Comparing (14) to (12), a pattern emerges which turns out also to hold for triple (and in fact, still higher-order) products. As such, we obtain the following expression,

$$\left[ \mathbb{S}_n^{(k)} \right]_{abc} = \frac{\left[ \mathbb{C}_n^{(k)} \right]_{abc}}{1 - [\Gamma_n]_{aa} [\Gamma_n]_{bb} [\Gamma_n]_{cc}}. \quad (16)$$

In this setting,  $\mathbb{C}_n^{(k)}$  is defined as

$$\mathbb{C}_n^{(k)} \stackrel{\text{def}}{=} \left( X_n^{(k)} \bar{A}_n V_n \right)^T \times_1 \left( \left[ \sum_{G=1}^{25} \omega_{5,G} \mathbb{M}(s_{5,G}, t_{5,G}) \right] \times_2 \left( X_n^{(k)} \bar{A}_n V_n \right) \right) \times_3 \left( X_n^{(k)} \bar{A}_n V_n \right), \quad (17)$$

where  $\times_1$ ,  $\times_2$ , and  $\times_3$  indicate the 1, 2, and 3-mode products of a tensor with a matrix (resulting in tensors where each entry corresponds to the product of the matrix and the corresponding column, row or tube fiber of the tensor), respectively.  $\mathbb{M}$  is the  $16 \times 16 \times 16$  third-order tensor containing triple products of segments of uniform bicubic B-splines, i.e., bivariate functions of bidegree 9, for which we use a  $5 \times 5$  Gauss-Legendre scheme.

Ultimately, we obtain (cf. (15) and (13))

$$\mathcal{J}_{n,p,q,r} \stackrel{\text{def}}{=} \iiint_{\Omega} \varphi_{n,p}(u, v) \varphi_{n,q}(u, v) \varphi_{n,r}(u, v) du dv = \frac{1}{4} \vec{\beta}_{n,p}^T \left( \left( \sum_{k=1}^3 \mathbb{S}_n^{(k)} \right) \bar{\times}_2 \vec{\beta}_{n,q} \right) \vec{\beta}_{n,r}, \quad (18)$$

where  $\bar{\times}_2$  indicates the 2-mode product of a tensor with a vector (resulting in a matrix where each entry corresponds to the inner product of the corresponding row fibre of  $\mathbb{S}_n^{(k)}$  with  $\vec{\beta}_{n,q}$ ; see [28]).

Using this tensor notation, the above method can readily be extended to computations of integrals involving higher-order products to compute for instance centroids, moments of inertia, and radii of gyration of subdivision surfaces.

*Remark 1.* For exact per-quad integration, as for instance in (18), one can use the tensor product of standard polynomial Gauss rules for the corresponding degrees (bidegree 9 in (18) requires 5 quadrature points in each parametric direction). However, due to the continuity of (subdivision) splines across elements, this integration requires redundantly many quadrature points. We now focus on more efficient integration schemes that integrate exactly splines spanning several elements.



### 3. Gaussian quadrature for univariate B-splines

In this section, we give a brief overview of fundamental facts about piecewise polynomials (B-splines) and associated Gaussian quadrature rules. We then focus on particular univariate spline spaces that appear in the components of the tensor-product setting when using IgA on Catmull-Clark subdivision surfaces.

#### 3.1. B-splines

Piecewise polynomials were introduced by Schoenberg in the late 50s [41] and have been widely studied and applied since then [14, 16]. To define a univariate spline space over  $[a, b]$  consisting of  $N$  polynomial pieces, we first define a *knot vector* as

$$\mathbf{t} \stackrel{\text{def}}{=} (a = \underbrace{t_0, \dots, t_0}_{m_0}, \underbrace{t_1, \dots, t_1}_{m_1}, \dots, \underbrace{t_N, \dots, t_N}_{m_N} = b), \quad (19)$$

which can be split into the *domain partition*  $\mathbf{p} = (t_0, t_1, \dots, t_N) \in \mathbb{R}^{N+1}$  and the *vector of multiplicities*  $\mathbf{m} = (m_0, m_1, \dots, m_N) \in \mathbb{N}^{N+1}$  with  $1 \leq m_i \leq d+1$ ,  $i = 0, \dots, N$ , where  $d$  is the polynomial degree. We assume that  $\mathbf{t}$  is an *open knot* vector on  $[a, b]$ , i.e.,  $m_0 = m_N = d+1$ . We denote by  $\pi_d$  the space of polynomials of degree at most  $d$  and define the spline space associated with  $\mathbf{t}$  as

$$S_{\mathbf{t}}^{N,d} = \{f \in C^{d-m_i} \text{ at } t_i, i = 0, \dots, N \text{ and } f|_{(t_{i-1}, t_i)} \in \pi_d, i = 1, \dots, N\}. \quad (20)$$

For example, for  $C^2$ -continuous cubic splines appearing in Catmull-Clark subdivision, we have  $d = 3$  and  $m_i = 1$  for  $i = 1, \dots, N-1$ . We refer the reader to [14, 16] for a detailed introduction to splines.

#### 3.2. Gaussian quadrature rules for univariate B-splines

Consider the spline space  $S_{\mathbf{t}}^{N,d}$  of degree  $d$  defined over an open knot vector  $\mathbf{t}$  such that the space has non-zero support on  $N$  elements in  $[a, b]$ , and assume the space is of even dimension  $2m$ . Then, according to [33], there exists a Gaussian rule that exactly integrates every function from the space, i.e.,

$$\mathcal{Q}_a^b[f] \stackrel{\text{def}}{=} \sum_{i=1}^m \omega_i f(\tau_i) = \int_a^b f(t) dt \stackrel{\text{def}}{=} I_a^b[f] \quad (21)$$

for any  $f \in S_{\mathbf{t}}^{N,d}$ , where  $\tau_i$  are the quadrature *nodes* and  $\omega_i$  the quadrature *weights*.

Let  $\mathbf{D} = \{D_i\}_{i=1}^{2m}$  be a basis of  $S_{\mathbf{t}}^{N,d}$ . Since the rule  $\mathcal{Q}$  exactly integrates the basis, the  $m$  nodes and weights can be collected into a  $2m$ -dimensional vector

$$\mathbf{x} = (\tau_1, \dots, \tau_m, \omega_1, \dots, \omega_m) \in \mathbb{R}^{2m}$$

that solves the well-constrained polynomial system

$$\mathcal{Q}_a^b[D_i] = I_a^b[D_i], \quad i = 1, \dots, 2m. \quad (22)$$

To build this system, one must know a-priori the correct knot-interval of each node, as explained later in Remark 2.

As was shown in [6], knowing the quadrature rule for a certain spline space, one can derive quadratures for ‘similar’ spline spaces using the homotopy continuation concept. More precisely, consider a source space (e.g., a discontinuous polynomial space) with a known Gaussian rule (the union of classical polynomial Gauss rules in our example), and consider a transformation of the spline space realised by knot vector modification. The continuous transformation of a Gaussian rule can be viewed as a curve in  $2m$ -dimensional space, and the homotopy continuation method numerically traces this curve from the starting position (source rule) to the final position (target rule). We refer the reader to [6] for a detailed description of this concept.

In case of odd dimension of  $S_{\mathbf{t}}^{N,d}$ , one can either use a super-space of even dimension by incrementing the dimension by one (for instance by adding a new knot or increasing the spline degree by one), or by employing Gauss-Radau quadrature [8].

We now focus on particular univariate spline spaces which are encountered when applying IgA on Catmull-Clark subdivision surfaces.

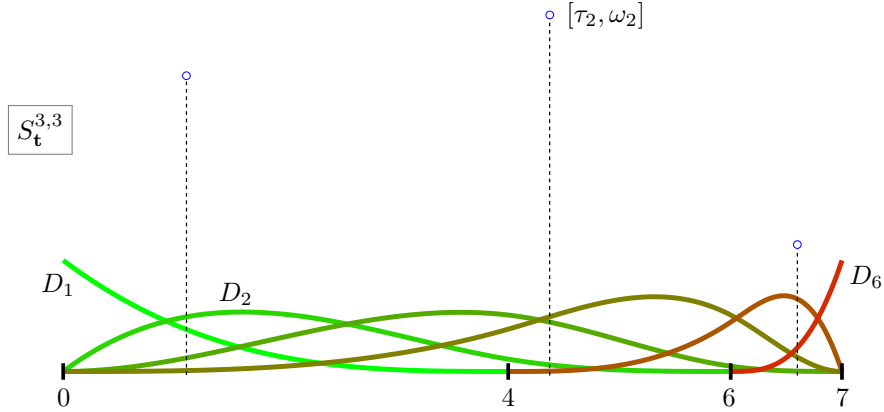


Figure 4: The basis functions  $D_1, D_2, \dots, D_6$  of the six-dimensional spline space  $S_{\mathbf{t}}^{3,3}$  over the open knot vector  $\mathbf{t} = (0, 0, 0, 0, 4, 6, 7, 7, 7, 7)$  are shown. Gaussian quadrature for this space requires 3 nodes and weights (blue circles), computed by solving (23). The values of nodes and weights are listed in Table 1 (top).

### 3.3. Quadrature rules for Catmull-Clark blending functions

We use the homotopy continuation method [6] outlined above to compute the nodes and weights of Gaussian rules for  $C^2$  cubic spline spaces over two particular knot vectors. The choice of these spaces will become apparent in Section 4. Due to the small sizes of the system (22) corresponding to these spaces, we directly derive the equations for the target rules. These can then be used to compute solutions (i.e., Gaussian quadrature rules) with arbitrary numerical precision.

The first spline space spans  $N = 3$  elements and is given by the knot vector  $\mathbf{t} = (0, 0, 0, 0, 4, 6, 7, 7, 7, 7)$ . The dimension of this space is six, therefore three Gaussian nodes are sufficient to exactly integrate this space. These nodes lie one in each element, i.e.,  $\tau_1 \in [0, 4]$ ,  $\tau_2 \in [4, 6]$ , and  $\tau_3 \in [6, 7]$ , see Figure 4. The Gaussian rule is given by the root of the algebraic system

$$\begin{aligned}
 \frac{1}{64}\omega_1(4-x_1)^3 &= 1, \\
 \omega_1\left(\frac{3}{4}x_1 - \frac{5}{16}x_1^2 + \frac{19}{576}x_1^3\right) + \frac{1}{72}\omega_2(2-x_2)^3 &= \frac{3}{2}, \\
 \omega_1\left(\frac{1}{8}x_1^2 - \frac{47}{2016}x_1^3\right) + \omega_2\left(\frac{28}{3} + \frac{7}{2}x_2 - \frac{3}{4}(4+x_2)^2 + \frac{25}{504}(4+x_2)^3\right) + \frac{1}{21}\omega_3(1-x_3)^3 &= \frac{7}{4}, \\
 \frac{1}{168}\omega_1x_1^3 + \omega_2\left(-\frac{112}{9} - \frac{14}{3}x_2 + \frac{7}{6}(4+x_2)^2 - \frac{23}{252}(4+x_2)^3\right) + & \\
 \omega_3\left(\frac{2072}{9} + \frac{175}{3}x_3 - \frac{28}{3}(6+x_3)^2 + \frac{31}{63}(6+x_3)^3\right) &= \frac{7}{4}, \\
 \frac{1}{18}\omega_2x_2^3 + \omega_3\left(-\frac{5720}{9} - \frac{478}{3}x_3 + \frac{79}{3}(6+x_3)^2 - \frac{13}{9}(6+x_3)^3\right) &= \frac{3}{4}, \\
 \omega_3x_3^3 &= \frac{1}{4},
 \end{aligned} \tag{23}$$

where the quantities  $x_1$ ,  $x_2$ , and  $x_3$  uniquely determine the positions of the nodes in their respective intervals, i.e.,  $\tau_1 = x_1$ ,  $\tau_2 = x_2 + 4$ , and  $\tau_3 = x_3 + 6$ . The obtained Gaussian rule is listed in Table 1 (top).

The second  $C^2$  cubic spline space we are particularly interested in is defined over the knot vector  $\mathbf{t} = (0, 0, 0, 0, 4, 6, 7, 8, 9, 9, 9, 9)$  and spans  $N = 5$  elements. This space is eight-dimensional and therefore we seek four nodes and weights to form the Gaussian rule. Homotopy continuation reveals that there is a single node in each knot-interval, except the middle one. This is encoded as the *nodal layout*  $(1, 1, 0, 1, 1)$ ; see [6,

Table 1: Univariate Gaussian quadrature rules for  $C^2$  cubic spline spaces over three (top) and five (bottom) elements. The rules were computed by solving systems (23) and (24), respectively, with the precision of 20 decimal digits.

$N = 3, \mathbf{t} = (0, 0, 0, 0, 4, 6, 7, 7, 7, 7)$		
$j$	$\tau_j$	$\omega_j$
1	1.11228459014357198166	2.65776637585316417534
2	4.37848409182500837502	3.20449953933037579726
3	6.60343858989701741989	1.13773408481646002741
$N = 5, \mathbf{t} = (0, 0, 0, 0, 4, 6, 7, 8, 9, 9, 9, 9)$		
$i$	$\tau_i$	$\omega_i$
1	1.13385119030944848407	2.71821477440833186253
2	4.53862051148258691251	3.45626788472875559044
3	7.26324566051338820450	1.96082618333924664344
4	8.66124083192921037142	0.86469115752366590359

Section 4.1]. The corresponding  $(8 \times 8)$  algebraic system reads

$$\begin{aligned}
\frac{1}{64}\omega_1(4-x_1)^3 &= 1, \\
\omega_1\left(\frac{3}{4}x_1 - \frac{5}{16}x_1^2 + \frac{19}{576}x_1^3\right) + \frac{1}{72}\omega_2(2-x_2)^3 &= \frac{3}{2}, \\
\omega_1\left(\frac{1}{8}x_1^2 - \frac{47}{2016}x_1^3\right) + \omega_2\left(\frac{28}{3} + \frac{7}{2}x_2 - \frac{3}{4}(4+x_2)^2 + \frac{25}{504}(4+x_2)^3\right) &= \frac{7}{4}, \\
\frac{1}{168}\omega_1x_1^3 + \omega_2\left(-\frac{32}{3} - 4x_2 + (4+x_2)^2 - \frac{13}{168}(4+x_2)^3\right) + \frac{1}{8}\omega_3(1-x_3)^3 &= 2, \\
\frac{1}{24}\omega_2x_2^3 + \omega_3\left(358 + \frac{159}{2}x_3 - \frac{21}{2}(7+x_3)^2 + \frac{11}{24}(7+x_3)^3\right) + \frac{1}{6}\omega_4(1-x_4)^3 &= \frac{5}{4}, \\
\omega_3\left(-\frac{849}{2} - \frac{369}{4}x_3 + \frac{51}{4}(7+x_3)^2 - \frac{7}{12}(7+x_3)^3\right) + \\
\omega_4\left(\frac{4077}{4} + \frac{783}{4}x_4 - \frac{93}{4}(8+x_4)^2 + \frac{11}{12}(8+x_4)^3\right) &= \frac{3}{4}, \\
\frac{1}{4}\omega_3x_3^3 + \omega_4\left(-\frac{7359}{4} - \frac{1389}{4}x_4 + \frac{171}{4}(8+x_4)^2 - \frac{7}{4}(8+x_4)^3\right) &= \frac{1}{2}, \\
\omega_4x_4^3 &= \frac{1}{4},
\end{aligned} \tag{24}$$

with  $\tau_1 = x_1$ ,  $\tau_2 = x_2 + 4$ ,  $\tau_3 = x_3 + 7$ , and  $\tau_4 = x_4 + 8$ . The numerical solution of this system is listed in Table 1, bottom.

*Remark 2.* We emphasise that it is in general very difficult to determine the nodal layout of quadrature rules over non-uniform knot vectors, even if they span only a few elements. For the two particular knot vectors considered in this section, we computed the Gaussian rules using homotopy continuation [6], and, as a by-product, we revealed the nodal layout to be  $(1, 1, 1)$  for  $N = 3$  and  $(1, 1, 0, 1, 1)$  for  $N = 5$ . As a consequence, the nodes and weights can be computed directly from the systems (23) and (24), respectively. For small  $N$ , one could guess/conjecture the correct layout and solve one (or several) algebraic system(s). For large  $N$ , however, such an approach is not feasible because of the exploding number of possible layouts.

#### 4. Bivariate spline spaces and efficient integration in subdivision

The quadrature rules derived above are optimal in terms of the number of quadrature points, i.e., are Gaussian. For example in the  $N = 3$  case, only three nodes are needed. This is in contrast to traditional per-element approaches where two nodes per element are required, and thus six nodes altogether. This then easily extends to the tensor-product setting, as illustrated in Figure 5, where the tensor-product space of

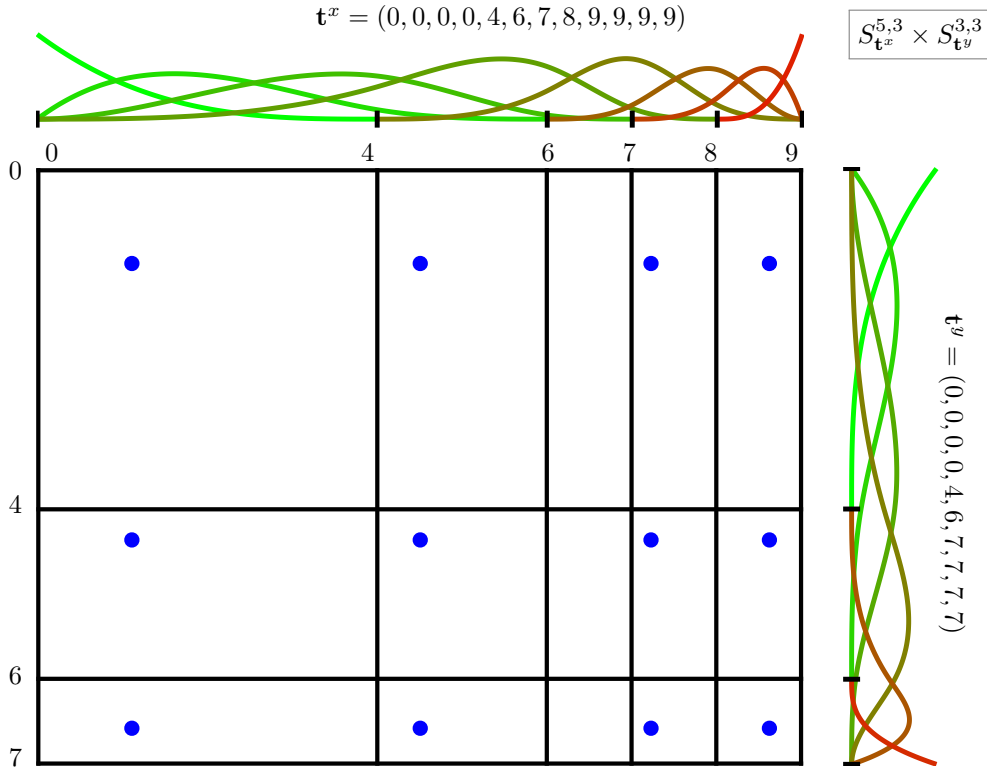


Figure 5: Tensor product Gaussian rule over  $3 \times 5$  elements. The univariate cubic  $C^2$ -continuous spline spaces contain eight and six basis functions, respectively. The tensor product quadrature rule that exactly integrates this spline space requires only 12 quadrature points (blue dots) while the polynomial Gauss rule requires 36. The corresponding values of nodes and weights are shown in Table 1.

the two spaces discussed above is used. The corresponding quadrature is then simply the tensor-product of the two univariate quadratures and the savings in the number of nodes needed are even higher.

However, the situation is more complicated on subdivision control meshes; see Figure 6. At EVs, the tensor-product structure is broken. Moreover, as explained in Section 2, one obtains infinitely many spline rings defined over just as many rings of quads. And so compared to the univariate case, there arises a question how to group quads into macro-elements (or strips) to take advantage of the smoothness of subdivision splines in order to reduce the number of nodes needed while still guaranteeing exactness of the rule. This is further complicated by the fact that different spline spaces (which are needed in various applications) lead to different macro-elements.

It is the aim of this section to identify subdivision splines spaces that arise in IgA on subdivision surfaces and to explore the space of possible macro-elements that lead to exact rules with a reduced number of nodes when compared to per-element quadrature.

We primarily address quadrature on quads incident with EVs because these require special attention. Regular quads with no incident EVs correspond to bicubic patches, as explained in Section 2.1, and can thus be integrated as such.

Let  $d$  be the polynomial degree and  $c$  be the continuity of the original spline space. We denote by  $(d, c)$  the degree-continuity pair that the spline space possesses. For example in the case of Catmull-Clark subdivision, the underlying univariate space is encoded by  $(3, 2)$ . In the tensor-product setting, this becomes  $(3, 2) \times (3, 2)$ . This is then the space to be used when deriving a quadrature rule for the subdivision splines themselves. However, in applications one typically needs to consider other spaces. For example, for area computations one needs to integrate the product of first derivatives of subdivision splines, which we demonstrate in Section 5. This then leads to the tensor-product space given by  $(5, 1) \times (5, 1)$ . This space and the spaces for volume computation and solving the Laplace problem are summarised in Table 2. Other spaces can be

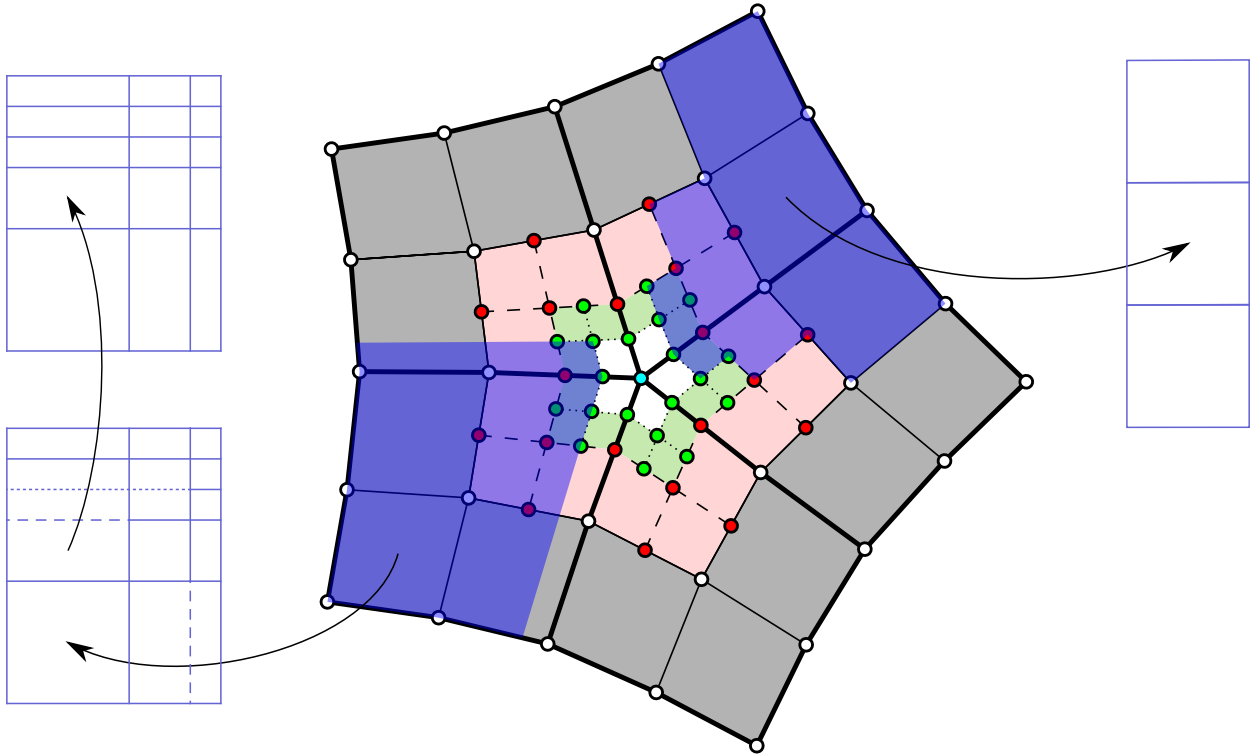


Figure 6: A 3-ring neighbourhood of an extraordinary vertex of valency 5 (marking of edges: original level is thick, after one refinement thin, after two dashed, and after three refinement steps the new edges are dotted). For a  $C^2$  bicubic spline space, the optimal quad-grouping in terms of quadrature points consists of five macro-elements (one shown in blue, left) positioned cyclically around the EV. In order to give the macro-element a tensor-product structure, virtual knot-insertion is performed (top left). Despite this redundancy, it is more economical in terms of quadrature nodes than using single-element integration or strips of three quads (shown on the right); cf. Table 2.

derived similarly.

Having identified spline spaces of particular interest in IgA on subdivision surfaces, we now look at the problem of grouping elements together to reduce the number of nodes needed. Our first observation is that the structure of the control mesh at an EV of valency  $n$  possesses (topological)  $n$ -fold symmetry. This should be reflected in the new rule. Additionally, we seek a rule that is independent of  $n$ . This suggests that quads forming a spline ring should be grouped into  $n$  strips of three quads each; see Figure 6, right.

However, it turns out that in some cases, a better grouping strategy exists. The key observation here is that we can group quads not only from one subdivision level, but from several levels at a time; see Figure 6, left. As we show below, the optimal grouping for the  $(3, 2) \times (3, 2)$  space is based on quads from three consecutive subdivision levels. This leads to rectangular *macro-elements* composed of 3 by 5 quads. Turning back to Section 3 and Figure 5, it follows that this macro-element requires only 12 nodes, while integrating each of the involved quads separately needs 36, i.e., three times as many, nodes.

Table 2 lists the number of nodes in the case of Catmull-Clark subdivision for three particular applications (area and volume computation, and the Laplace problem) using three integration strategies: per-element integration (classical polynomial Gauss), strips, and macro-elements.

We have described on an example that the grouping of quads heavily influences the number of nodes needed. We now proceed to a detailed exploration of the space of macro-elements and how they compare in terms of required integration nodes, including higher-degree subdivision schemes. Note that strips are a special case of macro-elements which group quads from one subdivision level only and thus always contain

Application	Spline space	Gauss	Strips	Macro-elements ( $r = 3$ )
Subdivision spline	$(3, 2) \times (3, 2)$	$2 * 2 * 9 = 36$	$2 * 3 * 3 = 18$	$3 * 4 = \mathbf{12}$
Area	$(5, 1) \times (5, 1)$	$3 * 3 * 9 = 81$	$3 * 7 * 3 = \mathbf{63}$	$7 * 11 = 77$
Volume	$(8, 1) \times (8, 1)$	$5 * 5 * 9 = 225$	$5 * 12 * 3 = \mathbf{180}$	$12 * 19 = 228$
Laplace	$(4, 1) \times (6, 2)$	$3 * 4 * 9 = 108$	$3 * 8 * 3 = \mathbf{72}$	$6 * 12 = \mathbf{72}$

Table 2: The numbers of quadrature points for Catmull-Clark subdivision, i.e., the original tensor product spline space  $(3, 2) \times (3, 2)$ . Three particular applications (rows) are compared based on using three quad-grouping strategies (columns). To allow for a fair comparison, the number of nodes corresponds to integrating three rings at a time because the macro-elements in this case span three rings.

only three quads.

*Remark 3.* The fact that strips (and also macro-elements) always have three quads in one of their directions stems from the fact that we implicitly assume the considered subdivision schemes are binary. In case of higher arity  $a$ , the number of quads in the same direction in a strip/macro-element is  $2a - 1$ . As subdivision schemes with higher arities are very rare, we consider only  $a = 2$ , i.e., the binary case, in this paper.

#### 4.1. Exploration of macro-elements

We now generalise the above ideas to arbitrary-degree subdivision surfaces [49] and explore that space of macro-elements spanning  $r$  subdivision levels. These macro-elements are rectangular  $r \times (r + 2)$  arrays of quads. Since consecutive rings of quads do not share all knot-lines (see Figure 6), knot insertion (see Figure 6, left) is needed to turn the collection of quads from  $r$  consecutive levels into a macro-element. On the other hand, this knot insertion is only virtual and is never actually performed. Note that strips do not require any (virtual) knot insertion since only one level is involved (see Figure 6, right).

The bivariate quadrature rule associated with a macro-element reads

$$\int_{\Omega} f(x, y) dx dy = \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} f(\tau_i, \tau_j) w_i w_j, \quad f \in S_{\mathbf{t}_x}^{r+2, d} \times S_{\mathbf{t}_y}^{r, d}, \quad (25)$$

where  $m_x$  and  $m_y$  are the numbers of nodes of the univariate Gaussian quadrature rules. An example of such a rule for  $r = 3$  with  $\Omega = [0, 9] \times [0, 7]$  is shown in Figure 5; the particular nodes and weights are listed in Table 1.

As we are able to derive univariate Gaussian rules for spline spaces with non-uniform knot vectors (Section 3), we can derive rules for macro-elements as well. At first sight it might appear that the situation can only improve as  $r$  grows. However, the necessary virtual knot insertion (see Figure 6) partially negates the advantage gained from using macro-elements over strips if  $r$  is sufficiently large, depending on the underlying spline space.

We explore the space of macro-elements for three particular applications: area and volume computation, and the Laplace problem. We assume the original spline space of degree  $d$  has the maximum possible continuity, i.e., it is encoded by  $(d, d - 1)$ . We compute the number of quadrature points needed to exactly integrate the ‘application’ spline space. Recall that for the integration of a piecewise polynomial function of degree  $d$ , traditional Gaussian quadrature needs  $\lceil \frac{d+1}{2} \rceil$  nodes per element, while spline Gaussian rules for  $c$ -continuous splines require asymptotically (for a large number of elements) only  $\lceil \frac{d-c}{2} \rceil$  nodes.

Figure 7 shows such an exploration for the area computation of a subdivision surface. The explored variables are the polynomial degree  $d$  and the number of rings  $r$  in the macro-element. Three quadrature schemes are compared: the classical (polynomial) per-element Gaussian quadrature, grouping into strips (of size  $1 \times 3$ ), and the grouping based on macro-elements of size  $r \times (r + 2)$ . For each grouping strategy, the bivariate functions show the number of quadrature points needed to compute the area integrals over rings of  $r$  consecutive levels of subdivision at an extraordinary vertex of valency  $n$ . In particular, we obtain

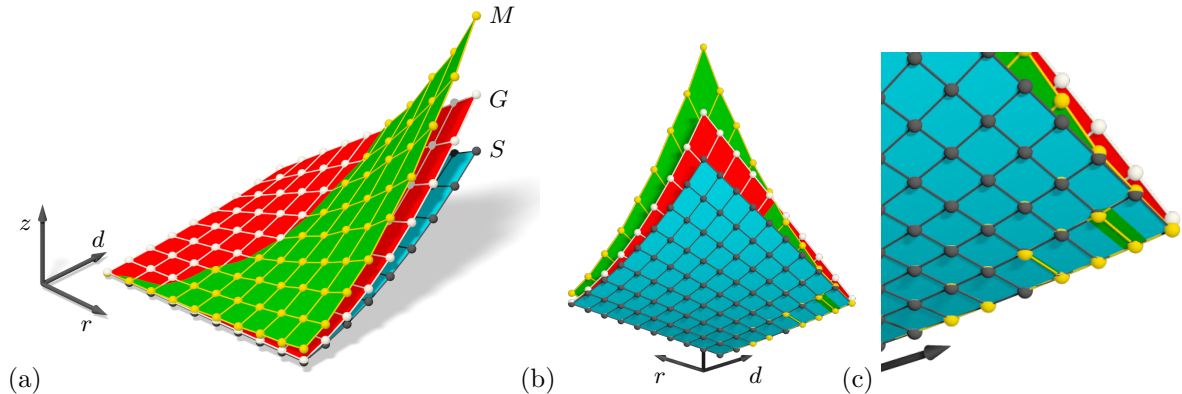


Figure 7: Exploration of the number of quadrature nodes needed in area computation on subdivision surfaces of degree  $d$  with various quad-grouping strategies. (a) In the  $d$ -direction, we explore the degree of the original spline space, while the  $r$ -direction corresponds to the size of the macro-elements. The bivariate graphs ( $z$ -values) correspond to the total number of quadrature points needed for exact integration over rings from  $r$  levels of subdivision at an EV, see (26). Three strategies are shown: macro-elements ( $M$ ), standard (polynomial) Gaussian quadrature ( $G$ ), and segmentation into strips ( $S$ ). (b) The bottom view shows that, in general, the most economical segmentation is formed by strips. (c) A zoom-in to the region along the  $d$ -axis, where, for higher polynomial degrees, the most economical integration is provided by macro-elements.

$$\begin{aligned}
 z_{Area}^S(d, r) &= n \cdot r \cdot U(2d - 1, 3) \cdot U(2d - 1, 1), \\
 z_{Area}^M(d, r) &= n \cdot U(2d - 1, r) \cdot U(2d - 1, r + 2), \\
 z_{Area}^G(d, r) &= 3 \cdot n \cdot r \cdot d^2,
 \end{aligned} \tag{26}$$

where

$$U(d, N) = \lceil (d + N)/2 \rceil \tag{27}$$

is the number of univariate spline Gaussian quadrature points and  $N$  is the number of elements.

A similar exploration in the case of volume computations reveals that grouping to  $(3 \times 1)$  strips is the most economical strategy for Catmull-Clark subdivision. Other spaces and macro-elements can be explored in a similar fashion.

For the Laplace problem, an analogous count to (26) gives the number of quadrature points as a function of  $d$  and  $r$  needed for exact integration. However, we emphasise that the space for the Laplace problem is not symmetric: the degree and continuity are different in different directions. This effectively means that the number of needed quadrature nodes would be the sum of the two node counts for the different directions in order to account for the asymmetry. To address this, we turn to the super-space containing the two spaces as sub-spaces. In the case of Catmull-Clark subdivision and the Laplace problem, this results into the (super-)space  $(6, 1) \times (6, 1)$ , which is symmetric. Continuing with our analysis based on this symmetric space, the most economical numerical integration is revealed to be based on macro-elements with  $r = 2$ ; see the Appendix for the concrete values of the nodes and weights.

#### 4.2. Tensor-product Gaussian rules for Catmull-Clark subdivision

The exploration of polynomial degrees and macro-elements has revealed the most economical quad-grouping for each application. For Catmull-Clark subdivision, the integration of the original space is cheapest by employing macro-elements with  $r = 3$ , while for area and volume computation the optimal grouping is formed by strips. For the Laplace problem, the most economical grouping, however, consists of  $(2 \times 4)$  macro-elements ( $r = 2$ ), which require  $6 \times 11 = 66$  nodes, while per-element polynomial Gauss rules (for bi-degree 6 polynomials) require  $6 \times 16 = 96$  nodes on the same macro-element; see Table 7 in Appendix.

Table 3: Univariate Gaussian quadrature for a  $C^1$  quintic spline space,  $(d, c) = (5, 1)$ , over  $N = 3$  uniform elements. This rule requires  $2N + 1$  Gaussian nodes and can be computed for arbitrary  $N$  using a recursive formula [5].

Area $N = 3, \mathbf{p} = (0, 1, 2, 3), \mathbf{m} = (6, 4, 4, 6)$		
$j$	$\tau_j$	$\omega_j$
1	0.12251482265544137787	0.30201742881457235729
2	0.54415184401122528880	0.48501960822246467975
3	1.00642424970771128383	0.44658741711143457868
4	1.5	0.53275109170305676856
5	1.99357575029228871617	0.44658741711143457868
6	2.45584815598877471120	0.48501960822246467975
7	2.87748517734455862213	0.30201742881457235729

Table 3 summarises the nodes and weights of the univariate Gaussian rule for the  $C^1$  quintic space over  $N = 3$  uniform elements. This rule, combined with classical polynomial Gaussian rule for  $d = 5$ , forms the bivariate tensor product rule that acts on strips in area computation of Catmull-Clark subdivision surfaces. This most economical grouping strategy brings approximately a 22% reduction in the number of nodes when compared to per-element integration (63 vs. 81, cf. Table 2) without affecting the exactness of the rule.

Tables with quadratures for other spaces arising in IgA on Catmull-Clark subdivision surfaces are included in Appendix.

## 5. Examples

We now illustrate the effectiveness of several proposed quadrature schemes in the context of subdivision surface area and volume computations, as well as isogeometric analysis on planar geometries. We implemented a framework for subdivision surfaces in MATLAB in order to compute the approximate integrals, and, where possible, to validate our results against exact solutions that are computed using the geometric series approach presented in Section 2.

We start by surface area and enclosed volume, and then move on to the more challenging Laplace and Poisson PDEs.

### 5.1. Surface area

Given a *planar* Catmull-Clark subdivision surface  $S$  parameterised as in Section 2.1 (note that  $f_i^3(u, v) = 0$  for all  $S_i$  in this case), we are interested in the surface area  $\mathcal{A}$  of  $S$ . We have that

$$\mathcal{A} = \iint_S 1 dS = \sum_i \iint_{S_i} 1 dS_i = \sum_i \iint_{\Omega} \left| \frac{\partial \vec{f}_i}{\partial u} \times \frac{\partial \vec{f}_i}{\partial v} \right| du dv, \quad (28)$$

where

$$\left| \frac{\partial \vec{f}_i}{\partial u} \times \frac{\partial \vec{f}_i}{\partial v} \right| = \left( \frac{\partial f_i^1}{\partial u} \frac{\partial f_i^2}{\partial v} - \frac{\partial f_i^2}{\partial u} \frac{\partial f_i^1}{\partial v} \right) = \left| \frac{\partial(x, y)}{\partial(u, v)} \right|. \quad (29)$$

It follows that  $\mathcal{A}$  can be computed exactly using the geometric series approach. As the integrand is of bidegree 5, we use a  $3 \times 3$  Gauss-Legendre scheme for each subpatch. Effectively,

$$\iint_{\Omega} \frac{\partial f_i^1}{\partial u} \frac{\partial f_i^2}{\partial v} du dv = \vec{P}_{i,x}^T \left( \iint_{\Omega} \frac{\partial \varphi_n}{\partial u} \frac{\partial \varphi_n^T}{\partial v} du dv \right) \vec{P}_{i,y}, \quad (30)$$



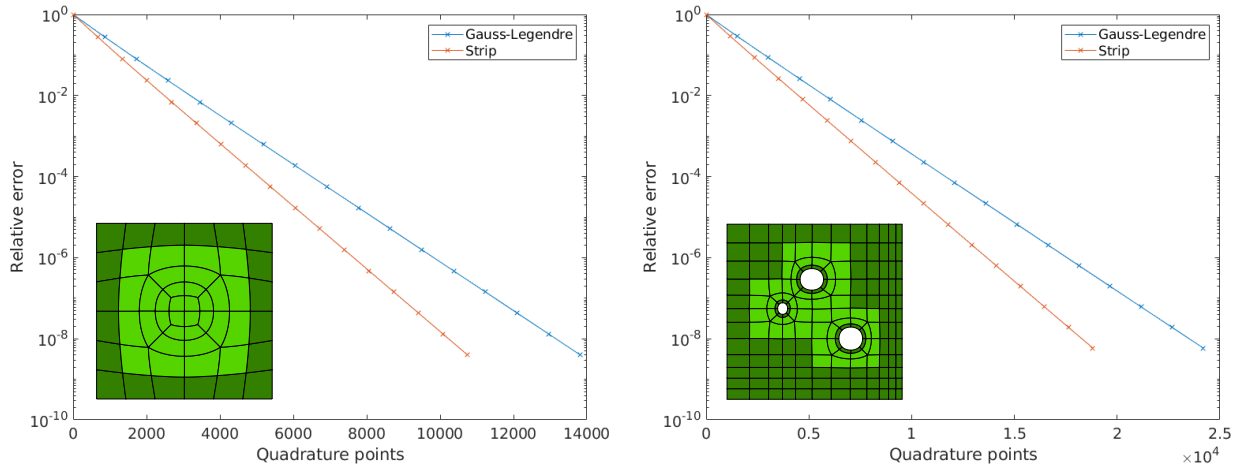


Figure 8: Convergence plots (relative error in surface area  $\mathcal{A}$ ) comparing  $3 \times 3$  Gauss-Legendre (blue) to the  $3 \times 7$  strip approach (orange) for two planar meshes.

where the double integral on the right-hand side can be (pre)computed as

$$\iint_{\Omega} \frac{\partial \varphi_n}{\partial u} \frac{\partial \varphi_n^T}{\partial v} du dv = V_n^{-T} \left( \sum_{k=1}^3 \mathcal{S}_n^{(k)} \right) V_n^{-1}. \quad (31)$$

Recall that  $\mathcal{S}_n^{(1)} + \mathcal{S}_n^{(2)} + \mathcal{S}_n^{(3)}$  is the  $(2n + 8) \times (2n + 8)$  matrix of integrals of products of first order partial derivatives of the eigenfunctions on  $\Omega$ ; see Section 2.5.

Once the exact surface area is computed, we can approximate the surface area using different quadrature approaches and study their convergence behaviour. Below we compare the use of the  $3 \times 3$  Gauss-Legendre scheme per subpatch to the  $3 \times 7$  strip per three subpatches as explained in Section 4. The example meshes and resulting convergence behaviour are illustrated in Figure 8.

The first observation is that the  $3 \times 3$  Gauss-Legendre scheme and the  $3 \times 7$  Strip yield the same values (up to machine precision, the highest difference we observed was in the order of  $10^{-15}$ ) for the surface area at a fixed level. Secondly, the strip approach clearly outperforms Gauss-Legendre as it uses fewer integration points.

## 5.2. Volumes

Given a solid  $V$  with the (non-planar) boundary  $S$  being a Catmull-Clark subdivision surface parameterised as above (i.e., composed of patches  $S_i$ ), and a vector field  $\vec{F}$ , the divergence theorem (Gauss's theorem) says that

$$\iiint_V (\nabla \cdot \vec{F}) dV = \iint_S (\vec{F} \cdot \vec{m}) dS, \quad (32)$$

where  $\vec{m}$  is the *unit* normal to the surface  $S$ , which is defined as

$$\vec{m} = \frac{\frac{\partial \vec{f}_i}{\partial u} \times \frac{\partial \vec{f}_i}{\partial v}}{\left| \frac{\partial \vec{f}_i}{\partial u} \times \frac{\partial \vec{f}_i}{\partial v} \right|}. \quad (33)$$

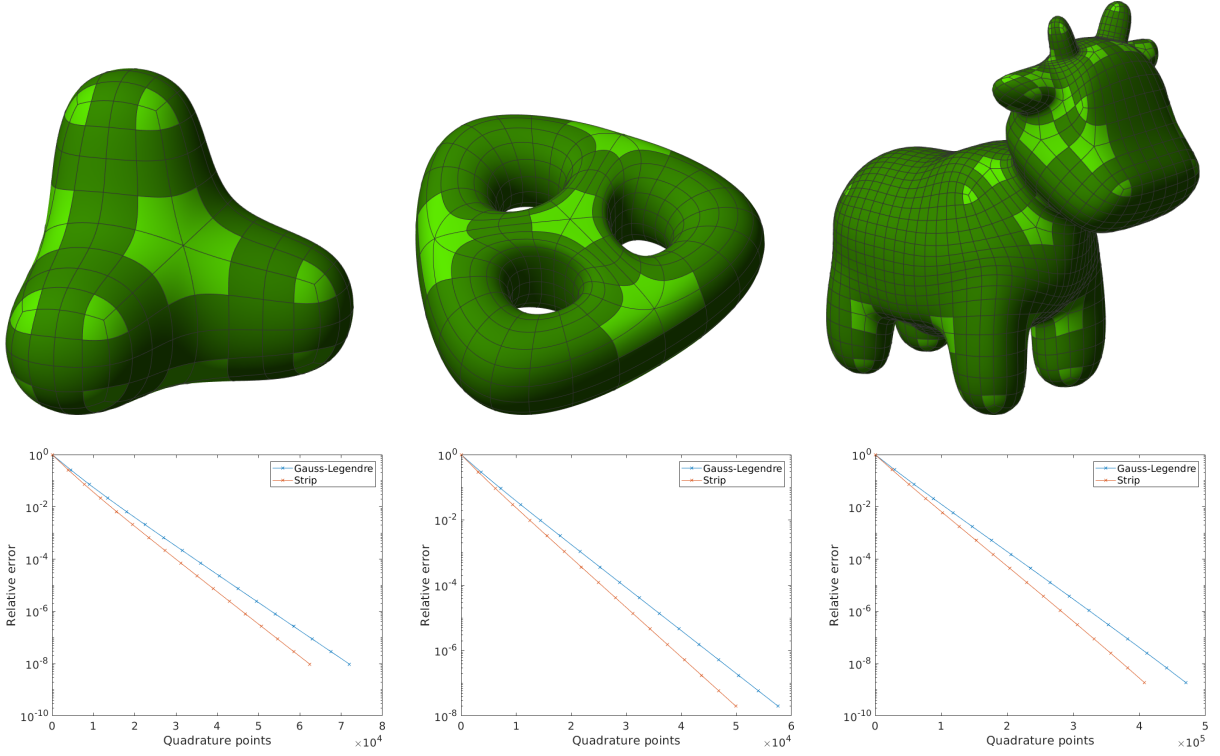


Figure 9: Convergence plots (relative error in volume  $\mathcal{V}$ ) comparing  $5 \times 5$  Gauss-Legendre (blue) to the  $5 \times 13$  strip approach (orange) for the Tripod, TripleTorus and Spot (courtesy of Keenan Crane) meshes. Compare to Figure 8.

Defining  $\vec{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^T$  and choosing  $\vec{F} = (0, 0, z)^T = (0, 0, f_i^3)^T$ , rewriting (32) results in

$$\begin{aligned}
\iiint_V (\vec{\nabla} \cdot \vec{F}) \, dV &= \iint_S (0, 0, f_i^3)^T \cdot \vec{m} \, dS \\
&= \sum_i \iint_{S_i} (0, 0, f_i^3)^T \cdot \vec{m} \, dS_i \\
&= \sum_i \iint_{\Omega} (0, 0, f_i^3)^T \cdot \vec{m} \left| \frac{\partial \vec{f}}{\partial u} \times \frac{\partial \vec{f}}{\partial v} \right| \, du \, dv \\
&= \sum_i \iint_{\Omega} (0, 0, f_i^3)^T \cdot \left( \frac{\partial \vec{f}}{\partial u} \times \frac{\partial \vec{f}}{\partial v} \right) \, du \, dv \\
&= \sum_i \iint_{\Omega} f_i^3 \left( \frac{\partial f_i^1}{\partial u} \frac{\partial f_i^2}{\partial v} - \frac{\partial f_i^2}{\partial u} \frac{\partial f_i^1}{\partial v} \right) \, du \, dv. \tag{34}
\end{aligned}$$

As  $\iiint_V (\vec{\nabla} \cdot \vec{F}) \, dV = \iiint_V 1 \, dV$ , the value of this integral is in fact the volume  $\mathcal{V}$  of our body  $V$ , which can be computed exactly using the geometric series approach. As the integrand is of bidegree 8, a  $5 \times 5$  Gauss-Legendre scheme is used on each subpatch.

As a benchmark, we use the *Tripod* mesh (see Figure 1). From [23] the volume of this mesh is known to be  $\mathcal{V} = 2.50400547615920543764371490988$ . Using our MATLAB implementation, we obtained a volume of  $\mathcal{V} = 2.504005476159202$ , which is identical up to machine precision.

In Figure 9 we compare the convergence behaviour of the  $5 \times 5$  Gauss-Legendre scheme per subpatch to

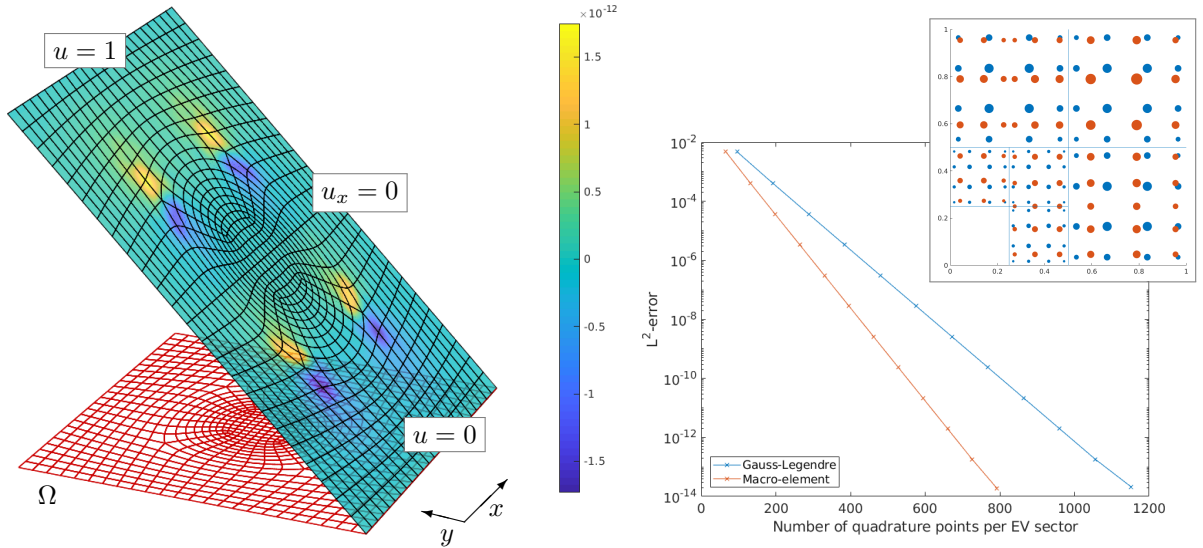


Figure 10: The patch test with known analytic solution. The approximate solution to Laplace’s problem (35) on a square plate with Dirichlet boundary conditions on the  $x$ -aligned edges and Neumann conditions in the  $y$ -direction. The colour-coding shows the difference between the exact solution (linear function) and the approximation (left). The  $L^2$  error as a function of the number of quadrature points when refining the rings surrounding the EVs is shown (right). Right top: the two quadrature schemes used: polynomial Gauss (blue) and spline Gaussian rule (orange); the radii of the dots reflect the weights.

the  $5 \times 13$  strip per three subpatches as explained in Section 4. Again, the strip approach outperforms the Gauss-Legendre approach, albeit by a lower ratio than in the case of surface area.

### 5.3. Isogeometric analysis

We consider the Laplace and Poisson PDEs in an isogeometric context [15] and solve them on the two planar meshes introduced in Section 5.1; see Fig. 8.

The exploration of macro-elements discussed in Section 4.1 reveals that the most economical grouping for the Laplace problem consists of  $2 \times 4$  quads. Moreover, in order to further reduce the number of function evaluations, we consider a spline space that contains both  $(4, 1)$  and  $(6, 2)$ , i.e., the (super-)space  $(6, 1)$ . The corresponding univariate spline Gaussian rules for the non-uniform spline spaces over 2 and 4 elements can be found in Appendix, Table 7.

We compare two integration schemes: the standard polynomial tensor-product Gauss rule used per element, as has so far been the case in the existing literature, and the spline Gaussian rule for the  $(6, 1) \times (6, 1)$  spline space over the  $2 \times 4$  macro-element. While the classical approach uses  $6 \times 4 \times 4 = 96$  quadrature points, the newly derived spline Gaussian rule requires only  $6 \times 11 = 66$  quadrature points. Due to this fact, our rules perform favourably in the error versus computational cost comparisons.

As a first example, we consider the Laplace problem with mixed boundary conditions on the unit square:

$$\begin{cases} \Delta u = 0 & \text{on } \Omega = [0, 1] \times [0, 1], \\ u = 0 & \text{on } y = 0, \\ u = 1 & \text{on } y = 1, \\ u_x = 0 & \text{on } x = 0 \text{ and } x = 1 \end{cases} \quad (35)$$

using a mesh containing EVs of valencies  $n = 3$  and  $n = 5$ ; see Fig. 8, left. The unique analytic solution is linear, see Figure 10 left, which shows the resulting approximation colour-coded by the error to the exact solution. As subdivision splines have linear precision [1] and are thus able to capture the solution exactly, this simulation boils down to comparing the effectiveness of different quadrature schemes. Figure 10, right, shows the  $L^2$  error between the approximate and exact solutions over  $\Omega$  when using the two integration

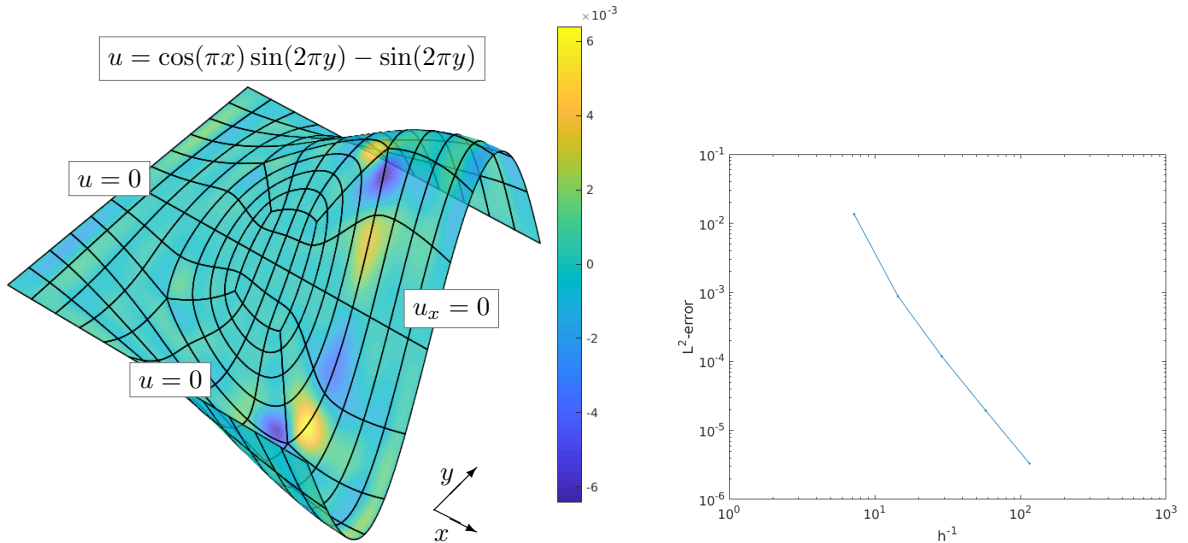


Figure 11: Approximation of the solution to Poisson's PDE on a square plate with three vanishing Dirichlet boundary conditions, and one vanishing Neumann condition ( $u_x = 0$ ) on  $x = 1$ . The colour shows the difference between the exact solution and the approximation (left). The convergence behaviour for  $h$ -refinement is shown on the right, where  $h = \sqrt{A/e}$  with  $A = 1$  the surface area of the geometry and  $e$  the number of elements.

schemes described above. The total error is plotted as a function of the number of quadrature points needed to integrate a neighbourhood of one EV. As expected, we observe a better error versus computational cost behaviour using the spline Gaussian integration scheme.

We note that the challenge of integrating subdivision splines was identified e.g. in [45, Appendix C4]. With our new approach this issue is partially mitigated. However, it should be emphasised that it is not only numerical quadrature that is to blame for numerical integration challenges when using subdivision. The general reason for convergence issues when using Catmull-Clark splines in IgA are the derivatives of the second and third eigenfunctions which are unbounded at EVs for valencies  $n > 4$ . As a result, the same holds for certain derivatives of the subdivision splines [40], and therefore also for the Jacobian, which comes into play in cases of non-trivial geometry maps.

As a second example, we consider Poisson's problem  $-\Delta u = f$  on the unit square  $\Omega$  with the same mesh as above, and with

$$f = \pi^2(5 \cos(\pi x) - 4) \sin(2\pi y),$$

vanishing Dirichlet boundary conditions along three boundary edges, and a vanishing Neumann boundary condition along the  $x = 1$  edge. The analytic solution of this problem is  $u = \cos(\pi x) \sin(2\pi y) - \sin(2\pi y)$ . In this case, the exact solution does not lie in the space spanned by the subdivision splines. Our results are displayed in Figure 11, showing the largest error around EVs with  $n > 4$ .

Finally, we solve the Laplace equation, using various mixed boundary conditions, on the square plate with three holes; see Fig. 8, right. Two results are shown in Figure 12. In these cases, the analytic solutions are unknown, and therefore we compute the Laplacian of the approximation (computed on the reference element). The main approximation errors in these cases appear again around EVs (which is expected), but also at the (interior) boundaries. Here, so-called *phantom points* [43] are used to define the elements on the boundary, which effectively means that the space is locally spanned by only 12 instead of the usual 16 basis functions. While this behaviour can be improved by using full Bézier edge-conditions (see [29] for a full treatment of end-conditions in the univariate case of subdivision based on polynomials) and also by considering modified subdivision weights [54], these investigations fall beyond the scope of the present paper, although our new approach to quadrature for subdivision splines still applies regardless of what edge-conditions and subdivision weights at EVs are used.

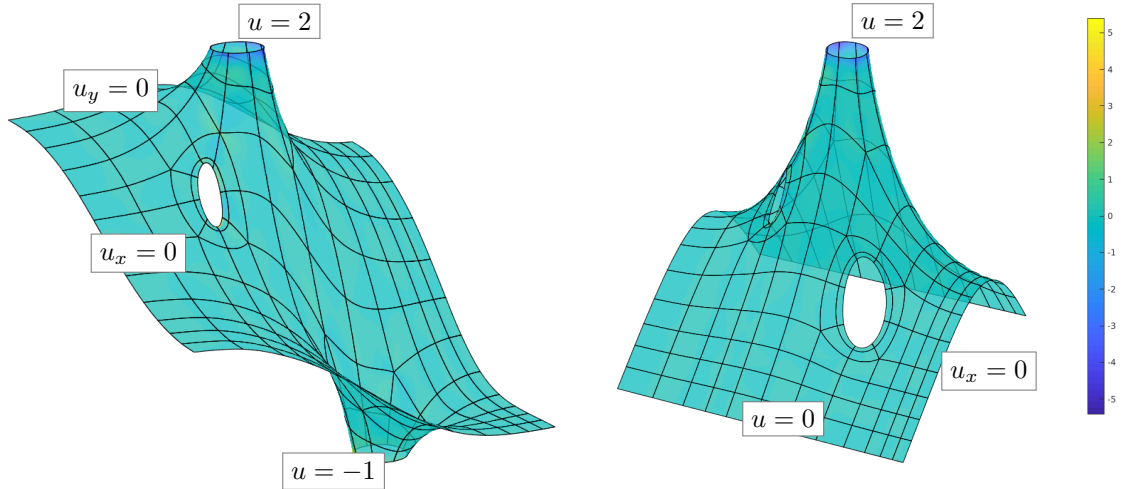


Figure 12: The Laplace equation solved on the square plate using the mesh with three holes from Fig. 8 right. Left: two Dirichlet conditions were applied to the outline of two of the holes and zero Neumann conditions elsewhere. Right: zero Dirichlet boundary conditions were applied on two sides of the plate and a non-zero value was set on one of the holes. The colour represents the Laplacian of the approximation.

## 6. Conclusion

We have addressed the problem of quadrature for subdivision surfaces based on tensor-product polynomials. We have shown that the naive per-element quadrature can be improved by considering various groupings of quadrilateral elements while still preserving exactness of the integration. Building on existing results on exact integration of subdivision splines, we have shown how to extend those results to multiple products of (derivatives of) subdivision splines. This allowed us to validate our efficient quadrature rules on several examples for which we could compute the area or volume exactly. The exact integration technique, using the tensor approach, can also be used to compute the centre of mass and higher order moments of volumes enclosed by subdivision surfaces.

On top of areas and volumes, we have demonstrated the effectiveness of our new quadrature rules in the context of isogeometric analysis. Namely, we have solved the Laplace and Poisson problems on Catmull-Clark subdivision surfaces defined over planar unstructured quadrilateral meshes.

We have focused on the case of Catmull-Clark subdivision, but our ideas extend also to higher degree subdivision [49], non-uniform [10, 11] subdivision, and (truncated) hierarchical [3, 52, 53] variants. This follows from the fact that all these schemes generate uniformly refined regions near extraordinary vertices.

The main focus of this paper is on extraordinary regions (light green in Figure 9). It remains an interesting avenue for future research to employ optimal tensor-product quadrature on the remaining elements grouped into rectangular regions such that the required number of nodes is reduced, or ideally minimised.

Additionally, our rules for macro-element integration may not be optimal. This stems from the fact that the underlying spline space is not of a tensor-product nature (Figure 6, left). Instead, one could attempt to design an optimal quadrature for the spline space defined on the underlying T-mesh. To the best of our knowledge, there are no theoretical results on the existence and uniqueness of quadrature rules for such non-tensor-product spline spaces.

Finally, it would be interesting to extend our ideas, both in the exact and numerical setting, to integration of volumetric subdivision splines.

*Acknowledgements.* The second author has been partially supported by the Basque Government through the BERC 2014-2017 program, by Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SEV-2013-0323, and the Project of the Spanish Ministry of Economy and Competitiveness with reference MTM2016-76329-R (AEI/FEDER, EU).

## References

- [1] Greg Arden. *Approximation Properties of Subdivision Surfaces*. PhD thesis, University of Washington, 2001.
- [2] F. Auricchio, F. Calabrò, T. J. R. Hughes, A. Reali, and G. Sangalli. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249-252(1):15–27, 2012.
- [3] Kosala Bandara, Thomas Rüberg, and Fehmi Cirak. Shape optimisation with multiresolution subdivision surfaces and immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 300:510–539, 2016.
- [4] Pieter Barendrecht. Isogeometric analysis for subdivision surfaces. M.S. thesis, Eindhoven University of Technology, 2013.
- [5] M. Bartoň, R. Ait-Haddou, and V. M. Calo. Gaussian quadrature rules for quintic splines. *Journal of Computational and Applied Mathematics*, 322:57–70, 2017.
- [6] M. Bartoň and V.M. Calo. Gaussian quadrature for splines via homotopy continuation: rules for  $C^2$  cubic splines. *Journal of Computational and Applied Mathematics*, 296:709–723, 2016.
- [7] M. Bartoň and V.M. Calo. Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 305:217–240, 2016.
- [8] M. Bartoň and V.M. Calo. Gauss-Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis. *Computer-Aided Design*, 82:57–67, 2017.
- [9] F. Calabrò, G. Sangalli, and M. Tani. Fast formation of isogeometric Galerkin matrices by weighted quadrature. *Computer Methods in Applied Mechanics and Engineering*, 316:606–622, 2017.
- [10] Thomas J. Cashman. NURBS-compatible subdivision surfaces. Technical Report UCAM-CL-TR-773, University of Cambridge, Computer Laboratory, 2010. (Doctoral thesis).
- [11] Thomas J. Cashman, Ursula H. Augsdörfer, Neil A. Dodgson, and Malcolm A. Sabin. NURBS with extraordinary points: high-degree, non-uniform, rational subdivision schemes. *ACM Trans. Graphics*, 28(3):1, 2009.
- [12] Edwin Catmull and Jim Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978.
- [13] Fehmi Cirak, Michael Ortiz, and Peter Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47(12):2039–2072, 2000.
- [14] E. Cohen, R. F. Riesenfeld, and G. Elber. *Geometric Modeling with Splines: An Introduction*. A. K. Peters, 2001.
- [15] J. A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, 2009.
- [16] C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.
- [17] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *Proceedings of SIGGRAPH '98*, pages 85–94, New York, USA, 1998. ACM.
- [18] E.J. Evans, M.A. Scott, X. Li, and D.C. Thomas. Hierarchical T-splines: Analysis-suitability, bézier extraction, and application as an adaptive basis for isogeometric analysis. *Comp. Methods in Applied Mech. and Eng.*, 284:1–20, 2015.
- [19] Carlotta Giannelli, Bert Jüttler, and Hendrik Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.
- [20] Carlos Gonzalez-Ochoa, Scott McCammon, and Jörg Peters. Computing moments of objects enclosed by piecewise polynomial surfaces. *ACM Trans. Graph.*, 17(3):143–157, July 1998.
- [21] Eitan Grinspun, Petr Krysl, and Peter Schröder. CHARMS: a simple framework for adaptive simulation. *ACM transactions on graphics (TOG)*, 21(3):281–290, 2002.
- [22] Jan Hakenberg and Ulrich Reif. On the volume of sets bounded by refinable functions. *Applied Mathematics and Computation*, 272:2–19, 2016.
- [23] Jan Hakenberg, Ulrich Reif, Scott Schaefer, and Joe Warren. Volume enclosed by subdivision surfaces, 2014.
- [24] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. In *Proceedings of SIGGRAPH '93*, pages 35–44, New York, NY, USA, 1993. ACM.
- [25] René R. Hiemstra, Francesco Calabrò, Dominik Schillinger, and Thomas J.R. Hughes. Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:966–1004, 2017.
- [26] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(58):301 – 313, 2010.
- [27] Bert Jüttler, Angelos Mantzaflaris, Ricardo Perl, and Martin Rumpf. On numerical integration in isogeometric subdivision methods for PDEs on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 302:131–146, 2016.
- [28] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- [29] Jiří Kosinka, Malcolm Sabin, and Neil Dodgson. Creases and boundary conditions for subdivision curves. *Graphical Models*, 76(5):240–251, 2014.
- [30] Charles Loop. Smooth subdivision surfaces based on triangles. M.S. thesis, University of Utah, 1987.
- [31] A. Mantzaflaris and B. Jüttler. Exploring matrix generation strategies in isogeometric analysis. In *International Conference on Mathematical Methods for Curves and Surfaces*, pages 364–382. Springer, 2012.
- [32] A. Mantzaflaris and B. Jüttler. Integration by interpolation and look-up for galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:373–400, 2015.
- [33] C.A. Micchelli and A. Pinkus. Moment theory for weak Chebyshev systems with applications to monosplines, quadrature formulae and best one-sided  $L^1$  approximation by spline functions with fixed knots. *SIAM J. Math. Anal.*, 8:206–230, 1977.

- [34] Thien Nguyen, Kecstutis Karčiauskas, and Jörg Peters. A comparative study of several classical, discrete differential and isogeometric methods for solving Poissons equation on the disk. *Axioms*, 3(2):280–299, 2014.
- [35] Thien Nguyen and Jörg Peters. Refinable  $c^1$  spline elements for irregular quad layout. *Computer aided geometric design*, 43:123–130, 2016.
- [36] Qing Pan, Guoliang Xu, Gang Xu, and Yongjie Zhang. Isogeometric analysis based on extended Catmull-Clark subdivision. *Computers & Mathematics with Applications*, 71(1):105–119, 2016.
- [37] J. Peters and X. Wu. On the local linear independence of generalized subdivision functions. *SIAM Journal on Numerical Analysis*, 44(6):2389–2407, 2006.
- [38] Jörg Peters and Ahmad Nasri. Computing volumes of solids enclosed by recursive subdivision surfaces. In *Computer Graphics Forum*, volume 16. Wiley Online Library, 1997.
- [39] Jörg Peters and Ulrich Reif. *Subdivision Surfaces*. Springer Publishing Company, Incorporated, 2008.
- [40] Ulrich Reif and Peter Schröder. Curvature integrability of subdivision surfaces. *Advances in Computational Mathematics*, 14(2):157–174, 2001.
- [41] I. J. Schoenberg. Spline functions, convex curves and mechanical quadrature. *Bulletin of the American Mathematical Society*, 64(6):352–357, 1958.
- [42] Bernd Schwald. Exakte Volumenberechnung von durch Doo-Sabin-Flächen begrenzten Körpern. Diplomarbeit, Universität Stuttgart, 1999.
- [43] Jean E. Schweitzer. *Analysis and Application of Subdivision Surfaces*. PhD thesis, University of Washington, 1996.
- [44] Michael A Scott, Robert N Simpson, John A Evans, Scott Lipton, Stephane PA Bordas, Thomas JR Hughes, and Thomas W Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013.
- [45] Michael Andrew Scott. *T-splines as a design-through-analysis technology*. PhD thesis, The University of Texas at Austin, 2011.
- [46] Jingjing Shen, Jiří Kosinka, Malcolm Sabin, and Neil Dodgson. Conversion of trimmed NURBS surfaces to Catmull-Clark subdivision surfaces. *Computer Aided Geometric Design*, 31(7–8):486–498, 2014.
- [47] Jingjing Shen, Jiří Kosinka, Malcolm Sabin, and Neil Dodgson. Converting a CAD model into a non-uniform subdivision surface. *Computer Aided Geometric Design*, 48:17–35, 2016.
- [48] Jos Stam. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 395–404. ACM, 1998.
- [49] Jos Stam. On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree. *Computer Aided Geometric Design*, 18(5):383–396, 2001.
- [50] Deepesh Toshniwal, Hendrik Speleers, and Thomas JR Hughes. Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations. *Computer Methods in Applied Mechanics and Engineering*, 327:411–458, 2017.
- [51] Anna Wawrzinek and Konrad Polthier. Integration of generalized B-spline functions on Catmull-Clark surfaces at singularities. *Computer-Aided Design*, 78:60–70, 2016.
- [52] Xiaodong Wei, Yongjie Zhang, Thomas J.R. Hughes, and Michael A. Scott. Truncated hierarchical Catmull-Clark subdivision with local refinement. *Computer Methods in Applied Mechanics and Engineering*, 291:1–20, 2015.
- [53] Xiaodong Wei, Yongjie Jessica Zhang, Thomas J.R. Hughes, and Michael A. Scott. Extended truncated hierarchical Catmull-Clark subdivision. *Computer Methods in Applied Mechanics and Engineering*, 299:316–336, 2016.
- [54] Qiaoling Zhang, Malcolm Sabin, and Fehmi Cirak. Subdivision surfaces with isogeometric analysis adapted refinement weights. *Computer-Aided Design*, to appear, pages –, 2018.
- [55] Urška Zore, Bert Jüttler, and Jiří Kosinka. On the linear independence of truncated hierarchical generating systems. *Journal of Computational and Applied Mathematics*, 306:200–216, 2016.

## Appendix

Building on Section 4.2, we provide here further quadrature rules for IgA on Catmull-Clark subdivision surfaces. Table 4 lists quadrature weights and nodes for volume computations of Catmull-Clark subdivision surfaces. Since the space for volumes, encoded by  $(8, 1)$ , is odd-dimensional, we derive the rules for the super-space  $(9, 1)$ ; see Section 3.2. Tables 5 and 6 list the quadratures for the Laplace problem, and Table 7 lists the quadratures for the symmetric super-space  $(6, 1) \times (6, 1)$  based on the Laplace problem in the case of Catmull-Clark subdivision surfaces.

Table 4: Univariate Gaussian quadrature rule for  $(9, 1)$  spline space over 3 uniform elements requires 13 nodes. As the rule is symmetric on  $[0, 3]$ , only a half of the nodes and weights are listed.

Volume $N = 3, \mathbf{p} = (0, 1, 2, 3), \mathbf{m} = (10, 8, 8, 10)$		
$j$	$\tau_j$	$\omega_j$
1	0.04850054944699732930	0.12248110464981389735
2	0.23860073755186230506	0.24745843345844748980
3	0.51704729510436750234	0.29425875345698032366
4	0.79585141789677286330	0.24839430102735088178
5	1.00090607111914459160	0.17790851486646824132
6	1.21134238368896236357	0.25712717145291590323
7	1.5	0.30474344217604652572

Table 5: A tensor-product Gaussian quadrature rule for the bivariate  $(4, 1) \times (6, 2)$  spline space over a  $(2 \times 4)$  macro-element. The spline domain is  $[0, 3] \times [0, 5]$  and the univariate Gauss and Gauss-Radau rules require 4 and 10 quadrature points, respectively. The univariate spline spaces are determined by the vectors of domain partition  $\mathbf{p}$  and knot multiplicities  $\mathbf{m}$ .

Laplace $N = 2, \mathbf{p} = (0, 2, 3), \mathbf{m} = (5, 3, 5)$		
$j$	$\tau_j$	$\omega_j$
1	0.32477486069392855534	0.78876244370399555618
2	1.35604155085298648755	1.09264344411573453245
3	2.25083388735975581774	0.69304300547816049813
4	2.82512529206289843012	0.42555110670210941323
$N = 4, \mathbf{p} = (0, 2, 3, 4, 5), \mathbf{m} = (7, 4, 4, 4, 7)$		
$i$	$\tau_i$	$\omega_i$
1	0.19052657519817435490	0.47425747562494119385
2	0.88181173152846924269	0.83826850627481542469
3	1.71047438973948556930	0.55746258852597421156
4	2.34753427886705920757	0.53275109170305676856
5	2.87574463212964794257	0.49929337010420860255
6	3.37179679810814274788	0.50883916356590194111
7	3.87176715160094656967	0.47210164430791984607
8	4.32830369490259602227	0.45667398166681429285
9	4.76048908340607043439	0.37160238216233486747
10	5.0	0.07040063234148658683



Table 6: A tensor-product Gaussian quadrature rule for the bivariate  $(4, 1) \times (6, 2)$  spline space over a  $(4 \times 2)$  macro-element. The spline domain is  $[0, 5] \times [0, 3]$  and the univariate Gauss and Gauss-Radau rules require 7 and 6 quadrature points, respectively. The univariate spline spaces are determined by the vectors of domain partition  $\mathbf{p}$  and knot multiplicities  $\mathbf{m}$ .

Laplace $N = 4, \mathbf{p} = (0, 2, 3, 4, 5), \mathbf{m} = (5, 3, 3, 3, 5)$		
$j$	$\tau_j$	$\omega_j$
1	0.32663942662113820131	0.79337483714920146417
2	1.36524863800600350281	1.10473881734167104548
3	2.29707458769046276440	0.76407234749797429483
4	2.99521911193444218418	0.66017704797348517237
5	3.65972984948668893031	0.66145266492856317594
6	4.28583891701452915953	0.60483974435429649832
7	4.83079091801234405543	0.41134454075480834890
$N = 2, \mathbf{p} = (0, 2, 3), \mathbf{m} = (7, 4, 7)$		
$i$	$\tau_i$	$\omega_i$
1	0.18929920157860591514	0.47118377034506463716
2	0.87592598534849803621	0.83226590934825704935
3	1.69615819517585826027	0.73708779089838448307
4	2.30073577200987403220	0.50381097513891305256
5	2.75342083858704091335	0.38326896902881618915
6	3.0	0.07238258524056458872

Table 7: A tensor-product Gaussian quadrature rule for the bivariate  $(6, 1) \times (6, 1)$  spline space over a  $(4 \times 2)$  macro-element. The spline domain is  $[0, 3] \times [0, 5]$  and the univariate Gauss rules require 6 and 11 quadrature points, respectively. The univariate non-uniform spline spaces are determined by the vectors of domain partition  $\mathbf{p}$  and knot multiplicities  $\mathbf{m}$ .

Laplace $N = 2, \mathbf{p} = (0, 2, 3), \mathbf{m} = (7, 5, 7)$		
$j$	$\tau_j$	$\omega_j$
1	0.18157383506514091169	0.45189514054419685494
2	0.83953858246370294859	0.79645311303315804981
3	1.61821551805769233602	0.68494255832124327010
4	2.14562797834238869443	0.41579106433514857036
5	2.56293217303852229280	0.41595389963394340537
6	2.90561383028460713586	0.23496422413230984942
$N = 4, \mathbf{p} = (0, 2, 3, 4, 5), \mathbf{m} = (7, 5, 5, 5, 7)$		
$i$	$\tau_i$	$\omega_i$
1	0.18185290017891797150	0.45259280749113676534
2	0.84086288940035991270	0.79777568296969278972
3	1.62121562909000760386	0.68762477345815644137
4	2.15912602677494595113	0.43769372591712756838
5	2.60475816431512313246	0.44626288773165612947
6	2.99880804262144621298	0.35211507920734371708
7	3.38609997545107673461	0.43521953213902864887
8	3.81355819154319342282	0.38605131464693100757
9	4.16981064456985704150	0.36711516474717107854
10	4.57152802239185791389	0.40704416177654188371
11	4.90739232126353097188	0.23050486991521396993