

Quadratic splines on quad-tri meshes

Construction and an application to simulations on watertight reconstructions of trimmed surfaces

Toshniwal, Deepesh

DOI

[10.1016/j.cma.2021.114174](https://doi.org/10.1016/j.cma.2021.114174)

Publication date

2022

Document Version

Final published version

Published in

Computer Methods in Applied Mechanics and Engineering

Citation (APA)

Toshniwal, D. (2022). Quadratic splines on quad-tri meshes: Construction and an application to simulations on watertight reconstructions of trimmed surfaces. *Computer Methods in Applied Mechanics and Engineering*, 388, 1-29. Article 114174. <https://doi.org/10.1016/j.cma.2021.114174>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Quadratic splines on quad-tri meshes: Construction and an application to simulations on watertight reconstructions of trimmed surfaces

Deepesh Toshniwal

Delft Institute of Applied Mathematics, Delft University of Technology, The Netherlands

Received 3 January 2021; received in revised form 2 September 2021; accepted 3 September 2021

Available online xxxx

Abstract

Given an unstructured mesh consisting of quadrilaterals and triangles (we allow both planar and non-planar meshes of arbitrary topology), we present the construction of quadratic splines of mixed smoothness — C^1 smooth away from the unstructured regions of \mathcal{T} and C^0 smooth otherwise. The splines have several useful B-spline-like properties – partition of unity, non-negativity, local support and linear independence – and allow for straightforward imposition of boundary conditions. We propose a non-nested refinement process for the splines with multiple advantages — a simple computer implementation, reduction in the footprint of C^0 smoothness, boundary preservation, and excellent approximation behaviour in simulations. Furthermore, the refinement process leaves the splines invariant on the mesh boundary. Numerical tests indicate that the spline spaces demonstrate optimal approximation behaviour in the L^2 and H^1 norms under mesh refinement, and provide a viable approach to simulations on watertight reconstructions of trimmed surfaces.

© 2021 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Keywords: Isogeometric analysis; Quadrilateral-triangle meshes; Analysis-suitable splines; Optimal approximation; Trimmed surfaces

1. Introduction

Isogeometric Analysis [1] generalizes classical finite element analysis [2] and, at the same time, intends to seamlessly unify it with the field of Computer-Aided Design [3].¹ Achieving this latter objective would encapsulate the entire engineering design-through-analysis workflow in a uniform framework, yielding a significant boost to the efficiency of the current approaches. Currently the designs output by CAD software are, in general, not suitable for analysis and, consequently, a significant portion of the time in a design-through-analysis workflow is spent neither on CAD nor on FEA but is dominated by generating analysis suitable meshes for FEA or IGA [4].

As such a focal problem in IGA is analysis-suitable spline representations of complex surface and volumetric geometries. This is a multi-faceted problem. Firstly, to be able to create analysis-suitable spline geometries of arbitrary topologies, it is necessary to be able to define analysis-suitable spline spaces on general unstructured meshes. This means moving beyond structured quadrilateral and hexahedral meshes for surface and volumetric geometries, respectively. Secondly, CAD extensively utilizes trimmed representations for modelling surfaces, i.e., surfaces are defined via implicit Boolean operations (union, difference, intersection). We call them implicit

E-mail address: d.toshniwal@tudelft.nl.

¹ Abbreviations: IGA for Isogeometric Analysis, FEA for Finite Element Analysis, and CAD for Computer-Aided Design.

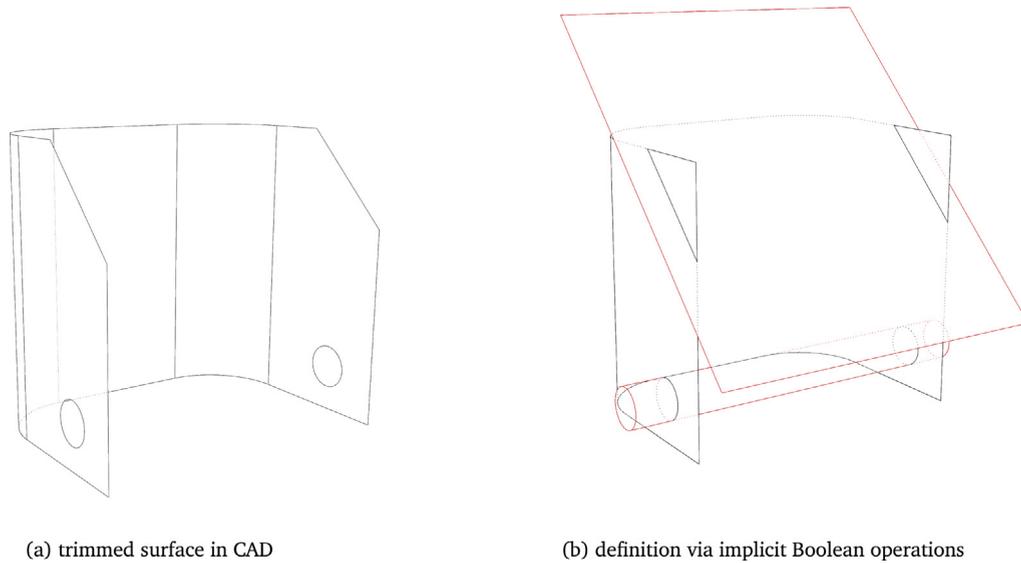


Fig. 1. The prevalent surface modelling paradigm in CAD is to define surfaces via implicit Boolean operations. For instance, the bracket geometry in (a) is represented through Boolean operations on the black extruded surface and the red plane and cylinders in (b). We call these Boolean operations implicit because they are not actually performed and change the underlying descriptions of neither of the three surfaces in (b). They only delineate which portions of the black surface in (b) should be made visible in (a); the rest of the black surface is present but is rendered invisible to a user.

Boolean operations because they are not actually performed to yield modified surfaces; rather, they are processed for visual purposes only, that is, to determine which portions of the surfaces to hide and which ones to show. Consequently, a surface object as the bracket in Fig. 1(a) may be being represented in CAD in an implicit manner through Boolean operations on the three surfaces in Fig. 1(b) — a plane, a cylinder and an extruded surface. This internal CAD representation of the geometry is not directly suitable for analysis. The interested reader is directed to [5], an excellent review article on the second and the third aspects mentioned here. Finally, most of the geometries built in CAD are boundary representations, i.e., they only work with surfaces that either define thin objects (e.g., the midsurface of a car body) or surfaces that bound volumetric objects (e.g., the two-dimensional boundary of a ship propeller). Consequently, if the objective is to perform analysis on volumetric objects, then a volumetric parameterization needs to be inferred from their boundary representations — a non-trivial task.

In this article, our primary focus is on the first aspect of the problem, analysis-suitable splines on unstructured meshes, and we present an application of our approach to the second problem, simulations on trimmed geometries. We begin by presenting our contributions in Section 1.1. Next, we highlight some prior work in these areas in Section 1.2 and then present a short outline of this paper in Section 1.3.

1.1. Our contributions

Given an unstructured mesh \mathcal{T} consisting of quadrilaterals and triangles (we allow both planar and non-planar meshes of arbitrary topology), we present the construction of quadratic mixed smoothness splines on \mathcal{T}^2 .

- The splines have several useful B-spline-like properties: partition of unity, non-negativity, local support and linear independence. The spline degree-of-freedom structure is simple and allows for straightforward imposition of boundary conditions. The splines are also C^1 smooth away from the unstructured regions of \mathcal{T} and C^0 smooth otherwise. All of these are made more precise in Proposition 3.1.
- We propose a convergent non-nested refinement process for the spline spaces that is, regardless, nested on the boundaries. This has two benefits both of which improve upon [6]. The non-nestedness of the refinement allows

² The terminology “blended” has been used in [6] to indicate that the splines have mixed smoothness. We opt for the explicit usage of “mixed smoothness” or “mixed” since “blended” is also frequently used in other contexts, e.g., to refer to “blending” functions.

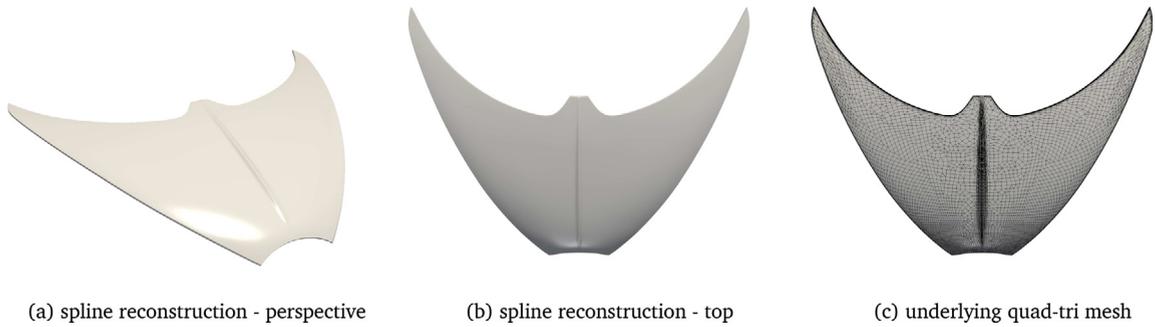


Fig. 2. An example of an application to watertight and editable reconstructions of CAD models. Shown here is the (unofficial) CAD model of a Bugatti Chiron hood that has been reconstructed using a quad-tri mesh and our splines. Figures (a) and (b) show different views of the model, and (c) shows the underlying quad-tri Bézier mesh. The quad-tri mesh generated by Rhinoceros® contains 13.8K faces, and the total number of quadratic spline degrees of freedom for this mesh are 14.2K (13.8K interior, 400 boundary). Also see [Remark 1.1](#).

us to “shrink” the neighbourhoods of C^0 smoothness in the mesh \mathcal{T} and leads to a very simplified computer implementation. At the same time, the refinement process leaves the spline invariant on the mesh boundary; this is especially useful if the boundaries of a spline geometry are composed of special curves such as conic sections. Numerical tests indicate that the spline spaces also demonstrate optimal approximation behaviour in the L^2 and H^1 norms for second-order problems under mesh refinement. Conceptually, this approach can be seen as an amalgamation of the “design” and “analysis” philosophies from [7] – it offers the ease of working with the design space while also being suitable for analysis; c.f. [Remark 4.4](#).

- The above properties, combined with the relative ease of generating quadrilateral-triangle meshes for trimmed surfaces, present the latter as one possible application of our splines; see [Fig. 2](#) as an example. We demonstrate such a workflow focused on trimmed surfaces in [Section 5](#).

Finally, the ideas we present in two dimensions can be easily extended to three dimensions to create splines on unstructured hexahedral-tetrahedral meshes. A forthcoming article [8] will utilize the purely quadrilateral and hexahedral case for building globally C^1 splines.

Remark 1.1. In this article, we utilize automatically generated quadrilateral-triangle meshes, such as in [Fig. 2\(c\)](#). Such meshes can be highly unstructured (e.g., too many extraordinary points, triangles), even unnecessarily so – for instance, it is clear that a “cleaner” mesh (e.g., fewer and feature-aligned elements, more structure) can be generated for the geometry in [Fig. 2\(a\)](#). We nevertheless work with such meshes to show the flexibility of our method. In practice, if the option to generate cleaner meshes is available, it should be preferred.

1.2. Related literature

The related literature can be roughly placed into four different categories. The first two concern the different approaches for building analysis-suitable splines on unstructured meshes – parametric smoothness and geometric smoothness – and the last two categories are related to the application we present — analysis on (watertight reconstructions of) trimmed surfaces. Note that this is an area of active research and we do not attempt to present an exhaustive overview of it here. Instead we point the readers to the comprehensive literature reviews presented in [9] for the first two categories and to [5] for the last two.

1.2.1. IGA-suitable splines on unstructured meshes

To generate surfaces from unstructured meshes using parametric smoothness conditions, a popular approach is that of subdivision [10]. Subdivision surfaces can work with triangular [11,12], quadrilateral [13,14] and quadrilateral-triangle meshes [15,16] of arbitrary topologies. Subdivision surfaces utilize spline functions that contain an infinity of polynomial pieces in unstructured regions of a mesh. The use of these descriptions in analysis presents several drawbacks such as expensive numerical quadrature and sub-optimal approximation behaviour. See [17–22] for examples of the use of subdivision surfaces in IGA.

Remark 1.2. On the face of it, our approach has the most in common with Doo–Sabin subdivision [15] in the sense that both generate with piecewise-quadratic splines on quadrilateral-triangle meshes. However, the approaches are not directly comparable as ours is a finite construction while Doo–Sabin subdivision surfaces employ infinite polynomial pieces in the unstructured regions of the mesh; the two approaches coincide on the structured regions of the mesh. As highlighted in the summary of our contributions, our finite approach is also able to overcome the drawbacks in analysis with subdivision schemes.

Parametric smoothness conditions can also be utilized to generate finite piecewise-polynomial splines and spline geometries if singularities in the unstructured regions of the mesh are permitted. This idea has been utilized in [23,24] to create the notion of smooth but degenerate Bézier patches or D-patches. This idea has been utilized in [7,25,26] to create bi-cubic spline spaces for IGA that possess a host of favourable properties for both design and analysis. In particular, [7,26] build spline spaces that demonstrate optimal approximation behaviour under mesh refinement in the analysis of fourth-order problems; see [27,28] for other examples of applications to high-order PDEs. For the special class of unstructured meshes that contain polar or collapsed-edge singularities, parametric smoothness was employed in [29] to create C^k smooth spline spaces, for all $k \geq 0$, that possess favourable properties for both freeform design and analysis. The case $k = 1$ has been explicitly laid out in the recent article [30] to build low-degree smooth ellipses and ellipsoids and in [31] for simulating incompressible flows on deforming surfaces.

Geometric smoothness between polynomial pieces can also be utilized to build smooth splines [32] and spline geometries [33]. While this is a well-known concept in CAD [3, Chapter 8], it has recently begun to be studied in the context of IGA also [34]. The case of C^1 smooth splines of degree (p, p) on planar multi-patch domains have been studied in [35,36] for bilinear parameterizations. This has been extended to the case of the so-called analysis-suitable G^1 parameterizations of planar domains in [37] and the spaces are shown to possess optimal approximation properties for L^2 projection problems. The latter spaces have also been recently combined with the Argyris triangular element for building C^1 spline spaces on planar quadrilateral-triangle meshes [38]. Some recent studies have also begun to lay out the practical foundations for geometrically smooth spline spaces on planar meshes with higher order smoothness; e.g., [36].

Finally, construction of smooth splines on planar triangulations has also been a long standing area of research [39]. When working with structured triangulations, Box splines are perhaps the most attractive approach [40]. Imposition of strong boundary conditions with Box splines is not straightforward but they can be utilized in the framework of immersed methods [41,42] to perform IGA. On more general planar triangulations, macro element techniques can be utilized for building smooth spline spaces; see [43,44] for construction and application of Powell–Sabin splines as well as the recent developments presented in [45].

1.2.2. Analysis on trimmed surfaces

The first class of methods here proposes to un-trim the trimmed geometries by reparameterizing them to create new watertight spline representations [46–50]. The reparameterizations can be done with splines on triangular meshes [49], on quadrilateral meshes [46–48] or using mixed triangular and quadrilateral meshes [50,51]. In particular, [46–48] all suggest a new design philosophy where, at each stage of design, a watertight spline geometry is retained by continuous reparameterization after every Boolean operation. This is a philosophy where our spline constructions can be readily utilized with the added advantage that we do not require robust quadrilateral-layout computations [48,52,53]. Similarly, our construction can also be directly used with the untrimmed (so-called hybrid boundary representations) geometries created by [50,51] that utilize both triangular and quadrilateral elements to reconstruct the geometries near trimming curves.

Remark 1.3. A problem that may arise with trimming operations is the presence of gaps and overlaps that make the generation of any kind of watertight mesh (triangular, quadrangular, etc.) difficult. This is a problem that affects every method adversely and is addressed in the field of mesh repair. For instance, quad-layout computations cannot be performed without a base watertight mesh [48,52,53]. We omit this problem from our discussions.

The second class of methods is where analysis is performed directly on trimmed representations without reparameterization; e.g., see [54–58]. In these methods, an immersed approach to the analysis is taken for appropriately treating trimming curves that lie within the boundaries of parametric patches. Furthermore, for accurate numerical quadrature, many such methods create a boundary conforming quadrilateral-triangle mesh on which integration is performed; e.g., [56]. Consequently, such meshes can also be directly combined with our splines for creating watertight and editable reconstructions of the trimmed geometries.

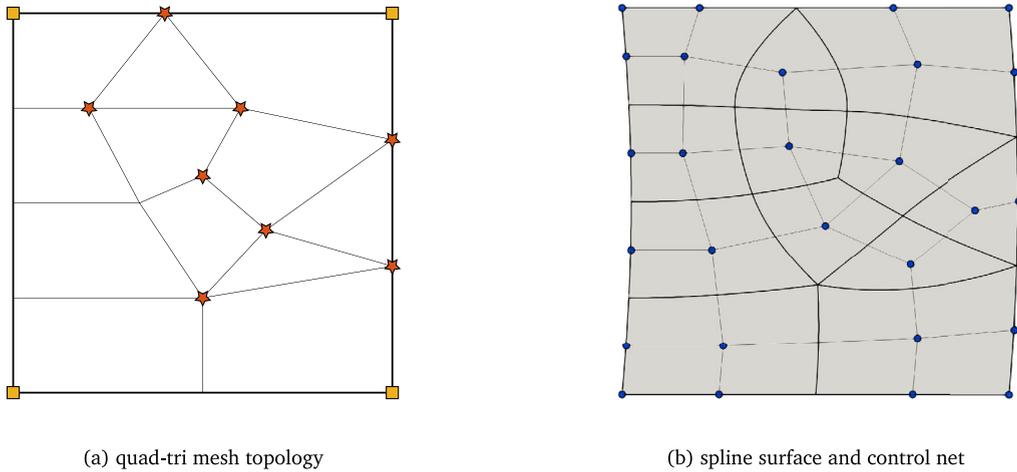


Fig. 3. An example quad-tri mesh \mathcal{T} is shown in (a). The boundary edges of \mathcal{T} have been displayed using heavier lines than the interior edges. The chosen corner vertices have been indicated by placing yellow squares on them, and the extraordinary vertices have been indicated by placing red stars on them. We will use the same convention throughout this document. Note that the image in (a) is meant to be interpreted only as a topological construct and not a visualization of the actual spline geometries. For instance, an example spline geometry built from the mesh in (a) is shown in (b); also see Section 3.1 and Fig. 4 for further elaboration. The control points have been displayed as filled blue disks, the control net has been displayed with thin lines that connect the control points, and the boundaries of the Bézier mesh have been displayed with thick lines. Note that the Bézier elements in (b) are in one-to-one correspondence with the mesh elements in (a).

Remark 1.4. A related problem in working with trimmed surfaces is volumetric completion [51,52,59] – if the trimmed surface defines the boundary of a volumetric object, then how can one generate the interior mesh and spline representation of that object. We do not discuss the volumetric case in this document.

1.3. Outline

We begin in Section 2 by introducing the related notation for the specific type of unstructured meshes that we focus on — quadrilateral-triangle meshes. Next, in Section 3 we present the construction of a set of quadratic spline basis functions on the mesh; Proposition 3.1 collects their useful properties. Then, in Section 4 we discuss the (non-nested) refinement of the quadratic splines; Proposition 4.2 characterizes our refinement rules. Finally, in Section 5 we present some numerical tests focusing on the analysis-suitability of the splines, including test cases that involve analysis on trimmed surfaces.

2. Unstructured polygonal meshes

We are interested in solving scalar and vector-valued PDEs on complex, 2-dimensional geometries of arbitrary topology. For instance, planar geometries in \mathbb{R}^2 or surfaces in \mathbb{R}^3 . While creating purely quadrilateral meshes for such geometries is a difficult task and a focus of active research [48,52,53], general polygonal meshes are significantly easier to create. Then, splines defined on polygonal meshes can help create such complex geometries, and can thereafter be used for numerically solving PDEs on them.

We focus here on *quadrilateral-triangle meshes*, i.e., meshes composed of quadrilaterals and triangles. This is not a restriction since more general polygons in a given mesh can be decomposed into triangles. Therefore, our attention on only quadrilateral and triangles is not restrictive. In this section, we define some relevant notation for such meshes; see Fig. 3 for reference.

Before we begin, we would like to point out that we consider \mathcal{T} as a topological construct only — in general, the quadrilaterals and triangles in \mathcal{T} will not be assumed to occupy a common parametric domain. Similarly, the meshes are not restricted to be planar or of trivial topology either. Representations as in Fig. 3(a) will only be for the purpose of specifying the connectivity of the different quadrilaterals and triangles with each other. As such,

spline geometries and spline functions on those geometries will be built by appropriately selecting the degrees of freedom for splines on \mathcal{T} . An example corresponding to the mesh in Fig. 3(a) is shown in Fig. 3(b).

A *quadrilateral-triangle mesh*, denoted \mathcal{T} and abbreviated as *quad-tri mesh*, is simply a polygonal mesh such that the polygons in it are quadrilaterals and triangles. We assume that \mathcal{T} is without any hanging nodes and that the interiors of all polygons are disjoint. Each quadrilateral or triangle in \mathcal{T} will be called a *face* of \mathcal{T} , or simply a *face*. For $k \leq 2$, the k -dimensional geometric components of the mesh will be collected in sets \mathcal{T}_k . That is, vertices in \mathcal{T}_0 , edges in \mathcal{T}_1 , and faces in \mathcal{T}_2 .

We assume that \mathcal{T} is such that each edge is shared by at most two faces of the mesh. If any edge is contained in only one face then it is called a *boundary edge*, otherwise it is called an *interior edge*; boundary edges have been displayed with slightly thicker lines in Fig. 3(a). Any vertices that lie on a boundary edge are called *boundary vertices*, otherwise they are called *interior vertices*. The set of interior vertices and edges will be denoted by $\overset{\circ}{\mathcal{T}}_k$ for $k = 0$ and 1, respectively. Moreover, a subset $\overline{\mathcal{T}}_0 \subset \mathcal{T}_0 \setminus \overset{\circ}{\mathcal{T}}_0$ is allowed to be classified as *corner vertices* — this will help create spline geometries with corners. For visual consistency, we will always denote the corner vertices of a mesh by placing yellow squares on them as in Fig. 3(a). Note that the subset $\overline{\mathcal{T}}_0$ can be arbitrarily specified for a given mesh; in particular, it can be chosen to be empty or it can contain all the boundary vertices. We also assume that there are no ‘kissing vertices’ in the mesh. That is, for any two faces σ, σ' that share a common vertex γ , there is a sequence of faces $\sigma_0, \dots, \sigma_\ell$ that all contain γ such that $\sigma_0 = \sigma$, $\sigma_i = \sigma'$, and $\sigma_i \cap \sigma_{i-1} \in \overset{\circ}{\mathcal{T}}_1$ for $i = 1, \dots, \ell$.

Remark 2.1. We interpret the corner vertices, $\overline{\mathcal{T}}_0$, as user-specified features of the geometry — as mentioned above, this will help create spline geometries with corners. This kind of a feature set can also be easily extended to the interior of the mesh by classifying, for instance, some interior edges as *creased edges*.

Finally, the *valence* of a vertex, edge or face of \mathcal{T} is defined to be the number of faces that contain it. In particular,

- the valence of any face is exactly 1 since each face contains itself;
- the valence of boundary and interior edges is 1 and 2, respectively, by the above definitions.

For $\phi \in \mathcal{T}_k$, $k \leq 2$, we will denote the valence of ϕ with μ_ϕ . Vertices of \mathcal{T} will be called *extraordinary vertices* if they belong to a triangular face, or if they are interior vertices with valences $\neq 4$, or if they are non-corner boundary vertices with valences > 2 . For visual consistency, we will always denote the extraordinary vertices of a mesh by placing red stars on them as in Fig. 3(a). For instance, the top two extraordinary vertices in Fig. 3(a) are the corners of a triangular face and have valence 4.

Remark 2.2. In this manuscript, the label “unstructured” applies if the mesh contains a non-grid-like arrangement of quadrilaterals or if the mesh contains any triangles.

3. Quadratic mixed smoothness splines

Given a quad-tri mesh \mathcal{T} , in this section we define *quadratic, mixed smoothness spline basis functions* on \mathcal{T} , or simply *mixed quadratic B-splines*. They will be C^1 smooth on almost all parts of the mesh and C^0 smooth elsewhere; this will be made precise shortly. However, in contrast to the approach in [6], the neighbourhoods with C^0 smoothness will shrink with each refinement; see Section 4 for details. As pointed out in Section 1.1, this significantly simplifies the data structures required in the implementation of these mixed quadratic B-splines.

We divide the definition of the mixed quadratic B-splines on \mathcal{T} into two parts. First we specify the degree of freedom structure and then we specify how the splines associated to each degree of freedom are defined on each face of \mathcal{T} .

3.1. Degrees of freedom

The degrees of freedom, or *dofs* in short, are divided into three categories; the spline construction will be specified for each category separately.

- **Face dofs:** We associate one degree of freedom to each face of \mathcal{T} , i.e., one degree of freedom for each member of \mathcal{T}_2 .

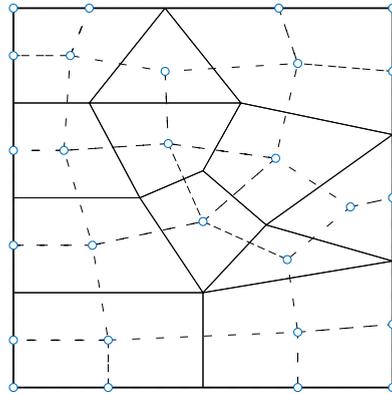


Fig. 4. For the example mesh shown in Fig. 3(a), the above image shows the dof structure. All dofs have been displayed as unfilled blue circles. There is one dof assigned to each mesh face, each boundary edge and each corner vertex, and the connectivity of these dofs is displayed with dotted lines; see Section 3.1 for details. The depiction here is to be treated only as containing topological information; the values of the dofs can be chosen arbitrarily of the locations shown here. For instance, choosing the dofs as the filled blue disks in Fig. 3(b), the surface shown there is obtained.

- Boundary edge dofs: We associate one degree of freedom to each boundary edge of \mathcal{T} , i.e., one degree of freedom for each member of $\mathcal{T}_1 \setminus \overset{\circ}{\mathcal{T}}_1$.
- Corner vertex dofs: We associate one degree of freedom to each corner vertex of \mathcal{T} , i.e., one degree of freedom for each member of $\overline{\mathcal{T}}_0$.

Thus, the total number of face, boundary edge and corner vertex dofs are $|\mathcal{T}_2|$, $|\mathcal{T}_1| - |\overset{\circ}{\mathcal{T}}_1|$ and $|\overline{\mathcal{T}}_0|$, respectively. The total number of dofs will be denoted as

$$n := |\mathcal{T}_2| + |\mathcal{T}_1| - |\overset{\circ}{\mathcal{T}}_1| + |\overline{\mathcal{T}}_0|. \tag{1}$$

For visual consistency, in the topological description of the mesh (as in Fig. 3(a)), we will denote each dof by placing an unfilled blue circle on the associated face/boundary edge/corner vertex of \mathcal{T} , and the connectivity of the dofs will be denoted with thin dotted lines; see Fig. 4. When discussing the concrete values these dofs take (e.g., when showing specific spline geometries as in Fig. 3(b) or when discussing the computation of new dofs during refinement in Section 4.2), the dofs will be denoted with filled blue disks, and their connectivity will be denoted with thin solid lines; see Fig. 3(b).

3.2. Definitions of mixed quadratic B-splines

Let us now define the mixed quadratic B-splines on \mathcal{T} , one for each dof. This will be done by specifying their local polynomial descriptions in terms of Bernstein–Bézier polynomials on quadrilaterals and triangles via so-called *extraction matrices*. Extraction matrices specify how the face-local Bernstein–Bézier polynomials can be linearly combined to yield the face-local description of a spline basis function; these were introduced in [60,61] and have been used, for instance, for defining splines on unstructured quadrilateral meshes in [6,7,29] and for multi-degree splines in [62,63].

Following the categorization of the dofs in Section 3.1, we will also call the associated splines *face*, *boundary edge* and *corner vertex splines*, respectively. Equivalently, we will say that a spline is associated to $\phi \in \mathcal{T}_k$, $k = 0, 1, 2$, if it is associated to the dof attached to ϕ . Denote these spline functions as B_i , $i = 1, \dots, n$. The associated spline space on \mathcal{T} is then going to be defined as

$$\mathcal{B} := \text{span}(B_i : i = 1, \dots, n). \tag{2}$$

The face-local descriptions of all splines will be specified by combining extraction matrices with quadratic and bi-quadratic Bernstein–Bézier polynomials defined on the triangular and quadrilateral faces of \mathcal{T} , respectively. The convention is explained in Section 3.2.1, and the definitions are subsequently presented in Sections 3.2.2–3.2.4.

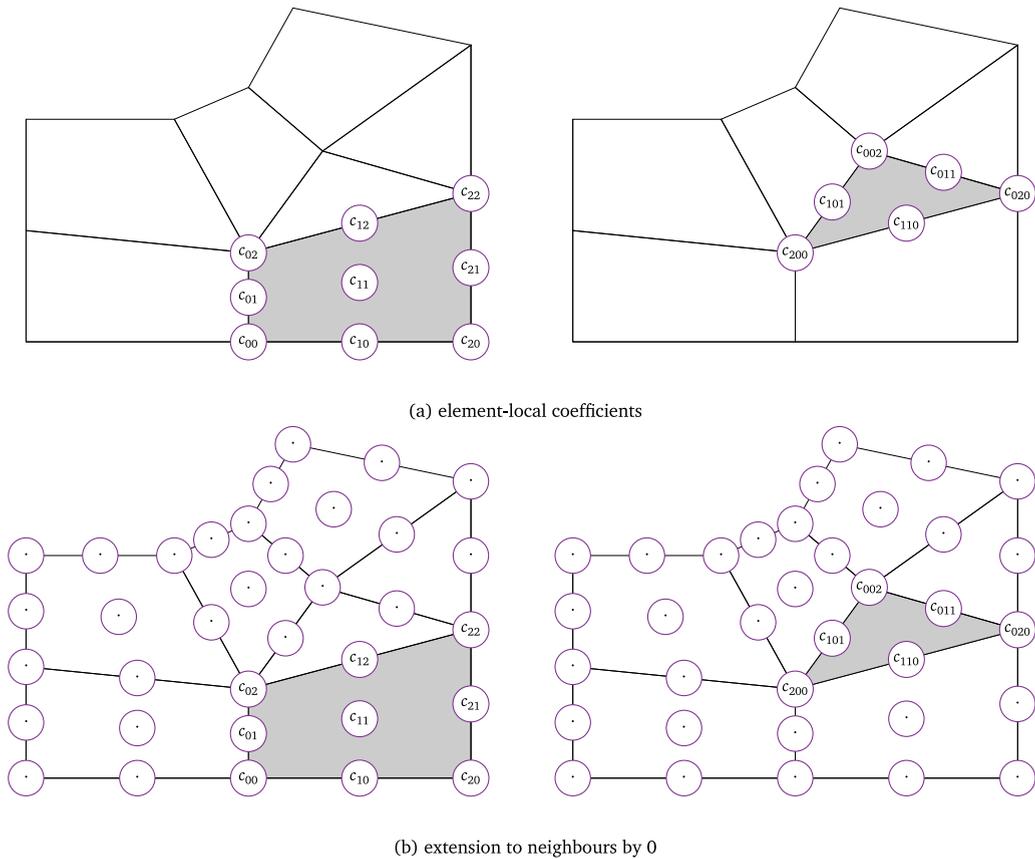


Fig. 5. Here we take a submesh of the one from Fig. 3(a) to illustrate how the extraction coefficients of mixed quadratic B-splines will be graphically denoted. Given a face $\sigma \in \mathcal{T}$, a mixed quadratic B-spline associated to σ or to a boundary edge/corner vertex belonging to σ will be defined graphically by specifying its Bernstein-Bézier coefficients on σ ; see figure (a). This description will be extended to neighbouring faces of \mathcal{T} with zero coefficients denoted here with a \cdot in figure (b). See Section 3.2.1 for further elaboration.

3.2.1. Extraction matrix convention

We will graphically denote the extraction matrix for each mixed quadratic B-spline. For all mixed quadratic B-splines, this will be done by specifying their Bernstein-Bézier coefficients on a single face of \mathcal{T} ; the face-local description will be extended to the neighbours. We elaborate upon this convention here and use Fig. 5 for reference. For visual consistency and to differentiate local Bernstein-Bézier coefficients from spline dofs, all local polynomial coefficients will be displayed inside magenta coloured disks.

Consider a quadrilateral face $\sigma_{\square} \in \mathcal{T}_2$ and let B_i be a spline associated either to σ_{\square} or to a boundary edge/corner vertex that belongs to σ_{\square} . Then, we will present the definition of B_i graphically as on the left in Fig. 5(a) by specifying 9 coefficients. The shown coefficients are to be interpreted as defining the following local spline description,

$$B_i|_{\sigma_{\square}} = \sum_{j,k=0}^2 c_{jk} b_{jk,\square}, \tag{3}$$

where $b_{jk,\square}$ is the jk -th bi-quadratic Bernstein-Bézier polynomial defined on σ_{\square} by interpreting it as the unit square $\square = [0, 1]^2$ (the origin is placed at the corner with the coefficient c_{00}) and local coordinates $\xi := (u, v)$,

$$b_{jk,\square}(\xi) := \binom{2}{j} \binom{2}{k} (1-u)^j (1-v)^k u^j v^k. \tag{4}$$

Similarly, consider now a triangular face $\sigma_\Delta \in \mathcal{T}_2$ and let B_i be a spline associated either to σ_Δ or to a boundary edge/corner vertex that belongs to σ_Δ . Then, we will present the definition of B_i graphically as on the right in Fig. 5(a) by specifying 6 coefficients. The shown coefficients are to be interpreted as defining the following local spline description,

$$B_i|_{\sigma_\Delta} = \sum_{\substack{j,k,\ell \geq 0 \\ j+k+\ell=2}} c_{jkl} b_{jkl,\Delta}, \tag{5}$$

where $b_{jkl,\Delta}$ is the jkl -th quadratic Bernstein–Bézier polynomial defined on σ_Δ by interpreting it as the standard triangle Δ (the origin is placed at the corner with the coefficient c_{200}) and with $\xi := (u, v, w)$ as the barycentric coordinates of a point inside it,

$$b_{jkl,\Delta}(\xi) := \binom{2}{j, k, \ell} u^j v^k w^\ell. \tag{6}$$

Finally, the face-local descriptions specified as above on any face σ are extended to any neighbouring face σ' with the help of zero coefficients; see Fig. 5(b). With this graphical convention in place, let us now define the three different types of splines in the following sections.

3.2.2. Face splines

Consider a face $\sigma \in \mathcal{T}_2$ and let B_i be the mixed quadratic B-spline associated to its dof. Let the edges and vertices of σ be numbered as shown at the top in Fig. 6(a). Then, the face-local coefficients for B_i are defined as in Fig. 6(b). The figure uses a *boundary flag* for vertices and edges which is a characteristic function defined as follows for $\phi \in \mathcal{T}_k, k = 0, 1$,

$$\chi_{\partial,\phi} = \begin{cases} 1, & \phi \in \overset{\circ}{\mathcal{T}}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

3.2.3. Boundary edge splines

Consider a face $\sigma \in \mathcal{T}_2$. Let the edges and vertices of σ be numbered as shown at the top in Fig. 6(a), and let τ_1 be a boundary edge and B_i be the mixed quadratic B-spline associated to its dof. Then, the face-local coefficients for B_i are defined as in Fig. 6(c). The figure uses a *corner flag* for vertices which is a characteristic function defined as follows for $\gamma \in \mathcal{T}_0$,

$$\chi_{\wedge,\gamma} = \begin{cases} 0, & \gamma \in \overline{\mathcal{T}}_k, \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

3.2.4. Corner vertex splines

Consider a face $\sigma \in \mathcal{T}_2$. Let the edges and vertices of σ be numbered as shown at the top in Fig. 6(a). Let γ_1 be a boundary vertex that has been chosen to be a corner vertex, and let B_i be the mixed quadratic B-spline associated to its dof. Then, the face-local coefficients for B_i are defined as in Fig. 6(d).

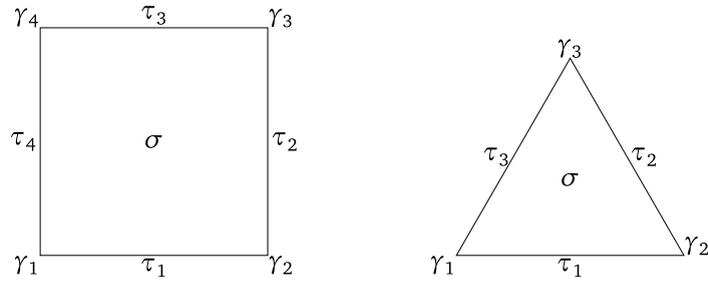
3.3. Spline functions

Given the above mixed quadratic B-spline definitions, we can create a spline function f by linearly combining the mixed quadratic B-splines with coefficients $f_i \in \mathbb{R}$,

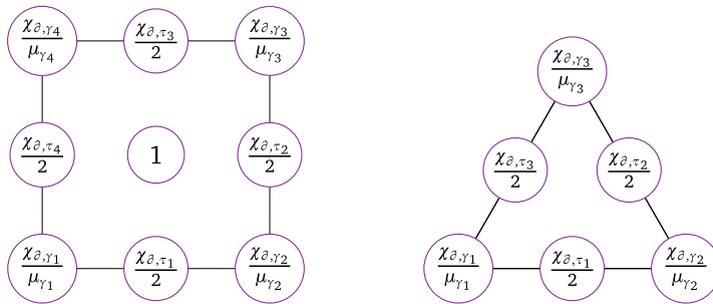
$$f := \sum_{i=1}^n f_i B_i. \tag{9}$$

Following the piecewise definition of mixed quadratic B-splines, the above definition is also interpreted in a piecewise manner. That is, for any $\sigma \in \mathcal{T}_2$ and local coordinates ξ as in Eqs. (4) and (6),

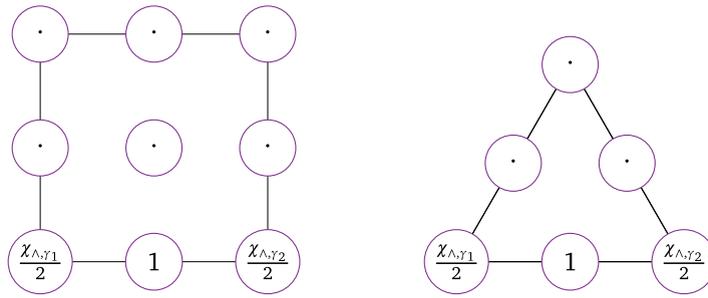
$$f|_{\sigma}(\xi) = \sum_{i=1}^n f_i B_i|_{\sigma}(\xi), \tag{10}$$



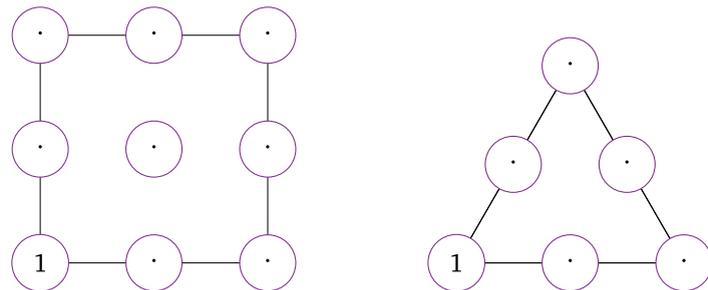
(a) face-local mesh labels



(b) face spline associated to σ



(c) boundary-edge spline associated to τ_1



(d) corner-vertex spline associated to γ_1

Fig. 6. This figure defines all mixed quadratic B-splines that may be associated to a face, boundary edge or a corner vertex. In (a), the local mesh neighbourhood of a quadrilateral and triangular face of \mathcal{T} are shown with vertices γ_i and edges τ_i . In (b), the face-local coefficients of a mixed quadratic B-spline associated to σ are presented. In (c), assuming that τ_1 is a boundary edge, the face-local coefficients of a mixed quadratic B-spline associated to τ_1 are presented. Finally, in (d), assuming that γ_1 is a corner vertex, the face-local coefficients of a mixed quadratic B-spline associated to γ_1 are presented.

where $B_i|_{\sigma}$ follow from Section 3.2.

These kinds of linear combinations can be used, for instance, to create a bivariate spline geometry. This can be done by choosing appropriate control points $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \dots, n$, and defining

$$\mathbf{x} := \sum_{i=1}^n \mathbf{x}_i B_i. \tag{11}$$

See again Fig. 3(b) for a spline geometry built in this manner. The locations of the filled blue disks in the plane correspond to the values of the dofs chosen.

3.4. Properties

The definitions of spline functions outlined in Section 3.2 immediately imply properties that are useful in numerical simulations. We collect the properties in the following result.

Proposition 3.1.

- (a) *Non-negativity:* On any $\sigma \in \mathcal{T}_2$ and any $i = 1, \dots, n$, $B_i|_{\sigma} \geq 0$.
- (b) *Partition of unity:* On any $\sigma \in \mathcal{T}_2$, $\sum_{i=1}^n B_i|_{\sigma} \equiv 1$.
- (c) *Local support:* If B_i is associated to a $\phi \in \mathcal{T}_k$, $k = 0, 1, 2$, then $B_i|_{\sigma} = 0$ for any $\sigma \in \mathcal{T}_2$ such that $\sigma \cap \phi = \emptyset$.
- (d) *Boundary Kronecker–Delta:* All B_i associated to face dofs are identically zero on the boundary of \mathcal{T} .
- (e) *Linear independence:* If \mathcal{T} contains at least one quadrilateral or at least one interior vertex of odd valence, then $\{B_i : i = 1, \dots, n\}$ form a basis for \mathcal{B} .
- (f) *Smoothness:* All B_i are at least C^0 smooth globally. Moreover,
 - let $\sigma \in \mathcal{T}_2$ not contain any extraordinary vertices and boundary edges/vertices, then all faces splines supported on σ are C^1 smooth across its edges;
 - let τ_1, τ_2, τ_3 and τ_4 meet at an interior vertex γ , $\mu_{\gamma} = 4$, with the edges labelled in a clockwise or counter-clockwise manner around γ , then all splines restricted to the pair τ_1, τ_3 or τ_2, τ_4 are C^1 smooth at γ ;
 - let $\tau \in \mathring{\mathcal{T}}_1 \setminus \mathring{\mathcal{T}}_1$ not contain any corner vertices, then all boundary edge splines supported on τ are C^1 smooth across the endpoints of τ .

Proof. The properties follow immediately by construction; the reasoning is as below.

- (a) Non-negativity is implied by the fact that all local polynomial coefficients in Figs. 6(b)–(d) are non-negative.
- (b) Partition of unity can be shown by taking an arbitrary $\sigma \in \mathcal{T}_2$ and by examining its face-local coefficients for all splines. We consider one of the cases here; the remaining cases are similarly proved. With the local numbering for a quadrilateral face as in Fig. 6(a), consider the coefficient c_{00} on σ and let $\gamma_1 \in \mathring{\mathcal{T}}_0$. Then, there are μ_{γ_1} faces that share γ_1 ; when the mixed quadratic B-splines associated to all these faces are restricted to σ , the value of c_{00} is equal to $\frac{1}{\mu_{\gamma_1}}$ for each. The value of c_{00} is 0 for all other mixed quadratic B-splines. Consequently, for the sum of all mixed quadratic B-splines restricted to σ , the value of the coefficient c_{00} is 1. The same can be shown for the remaining coefficients.
- (c) Local support follows by the extraction convention from Section 3.2.1; see Fig. 5(b).
- (d) The boundary Kronecker-Delta property follows by definition of the extractions in Fig. 6(b)–(d). Indeed, all face splines have zero Bernstein–Bézier coefficients on the boundary, leaving only the boundary-edge and corner-vertex splines as supported on the boundary. In fact, the mixed quadratic B-splines restricted to the boundary are simply univariate quadratic B-splines.
- (e) Linear independence can be shown by showing that $f = \sum_{i=1}^n f_i B_i = 0 \Rightarrow f_i = 0$ for all i . We discuss three cases: (I) boundary-edge and corner-vertex splines, (II) quadrilateral-face splines and (III) triangle-face splines. We will use the fact that, since $f = 0$, all Bernstein–Bézier coefficients on all faces must also be zero.

Case I: From Fig. 6(c), consider the Bernstein–Bézier coefficient in the middle of a boundary edge (i.e., c_{10} for a quadrilateral and c_{110} for a triangle). Then, the mixed quadratic B-spline associated to this edge is

the only one that contributes to this Bernstein–Bézier coefficient. Therefore, all coefficients f_i associated to boundary-edge splines should be zero. Similarly, from Fig. 6(d), all coefficients f_i associated to corner-vertex splines should be zero.

Case II: With reference to Fig. 6(b), consider the coefficients c_{11} on a quadrilateral face σ . Since the only mixed quadratic B-spline that contributes to c_{11} is the one associated to σ , we see that the appropriate coefficient f_i for that mixed quadratic B-spline must be zero. Thus, the coefficients f_i for all mixed quadratic B-splines associated to quadrilateral faces must be zero.

Case III: There are two sub-cases of interest here, both of which imply that there is at least one face such that the associated face-spline coefficient is zero.

- Let \mathcal{T} contain at least one quadrilateral face, then the coefficient of the associated face-spline must be zero from *Case II*.
- Let \mathcal{T} be a triangulation (i.e., none of its faces are quadrilateral) and let γ be an interior vertex of odd valence μ_γ . Let $\sigma_1, \dots, \sigma_{\mu_\gamma}$ be triangles containing γ and such that $\sigma_i \cap \sigma_{i-1} \in \overset{\circ}{\mathcal{T}}_1$, $i = 1, \dots, \mu_\gamma$, with σ_0 understood to be σ_{μ_γ} . If c_{110}^i is the Bernstein–Bézier coefficient in the middle of the edge $\sigma_i \cap \sigma_{i-1}$, it is the average of the face-spline coefficients associated to σ_i and σ_{i-1} . Then, since μ_γ is odd and all c_{110}^i are zero, the face-spline coefficients associated to σ_i , $i = 1, \dots, \mu_\gamma$, must also be zero.

Following both of the above sub-cases, we can iteratively apply the following argument to show that the coefficients for all face-splines must vanish.

Let σ_i be a triangular face that shares an edge with another (quadrilateral or triangular) face σ . Let the Bernstein–Bézier coefficients on the common edge τ be $c_{101}, c_{011}, c_{110}$. If the coefficient of the face spline associated to σ is 0, then the coefficient of the face spline associated to σ_i must also be zero. This is because, from Fig. 6(b), c_{011} is the average of these two coefficients.

(f) C^0 smoothness of all B_i follows from the extraction convention from Section 3.2.1; see Fig. 5(b).

- Let $\sigma \in \mathcal{T}_2$ not contain any extraordinary vertices or boundary vertices/edges. Then, the 1-ring of faces surrounding σ are all quadrilaterals and together they form a 3×3 grid of quadrilateral faces. Then, it is readily verified that the definitions in Fig. 5(b) imply that the Bernstein–Bézier forms of the splines satisfy C^1 smoothness conditions across all edges of σ . In particular, the mixed quadratic B-spline associated to σ is the standard C^1 tensor product bi-quadratic B-spline supported on this 3×3 grid of faces.
- Let τ_1, τ_2, τ_3 and τ_4 meet at an interior vertex γ , $\mu_\gamma = 4$, with the edges labelled in a clockwise or counter-clockwise manner around γ . Then, restricted to the pair τ_1, τ_3 , the local Bernstein–Bézier coefficients of any spline satisfy the univariate C^1 smoothness conditions at γ . Note that this is independent of whether γ is shared by only quadrilaterals or both quadrilaterals and triangles.
- Let $\tau \in \mathcal{T}_1 \setminus \overset{\circ}{\mathcal{T}}_1$ not contain any corner vertices. Then, the definitions in Fig. 5(c) imply that, restricted to the boundary, the Bernstein–Bézier forms of the splines satisfy C^1 smoothness conditions across both endpoints of τ . In particular, the mixed quadratic B-spline associated to τ is a standard univariate C^1 quadratic B-spline when restricted to τ and its two neighbouring boundary edges. ■

4. Mesh and spline refinement

Refinement of the mesh can help improve the resolving power of splines for the purpose of, for instance, obtaining a better approximation to the solution of a PDE. In this section, we outline how a given quad-tri mesh and spline functions defined on it are refined.

4.1. Refining \mathcal{T} and \mathcal{B}

Recall from Section 2 that \mathcal{T} is a topological construct only; thus we only need to specify how the connectivity and polygonal-composition of \mathcal{T} are to be updated. In particular, we implement a refinement process for \mathcal{T} that produces a purely quadrilateral mesh. This refinement process can be explained as follows: each n -sided polygon of τ is split into n -quadrilaterals. More precisely, all quadrilaterals are split into 2×2 quadrilaterals and each triangle

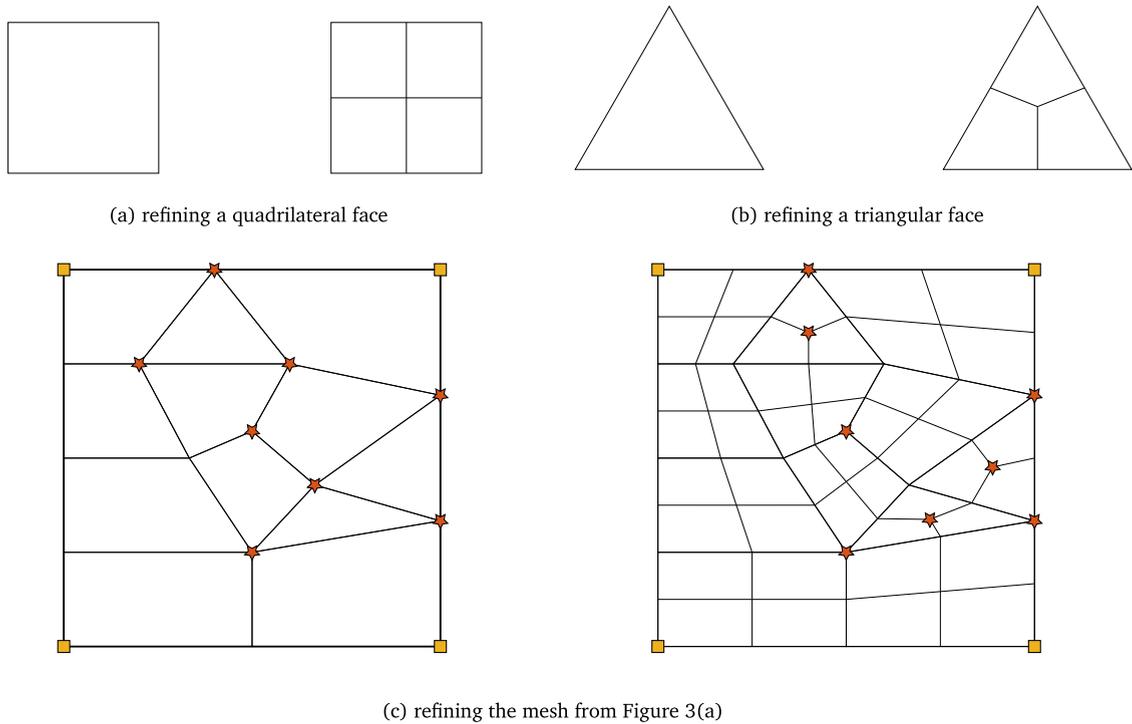


Fig. 7. The above figure shows how the mesh faces are split when performing refinement. Each quadrilateral and triangular face is split into 4 and 3 new faces as shown in figures (a) and (b), respectively. Figure (c) shows how the mesh topology changes after refinement for the mesh previously shown in Fig. 3(a). See Section 4.1 for details.

is split into 3 quadrilaterals; see Fig. 7. We will denote refined quantities with a “hat” — for instance, the refined mesh will be denoted as $\widehat{\mathcal{T}}$.

During this process, as is clear from Fig. 7, $|\mathcal{T}_1| + |\mathcal{T}_2|$ new mesh vertices are added; we assume that the old mesh vertices retain their labels. It is also assumed that no hanging nodes are introduced in this process. Moreover, since the old vertices retain their labels, the corner vertices of $\widehat{\mathcal{T}}$ are chosen to be identical to the corner vertices of \mathcal{T} , i.e., $\widehat{\mathcal{T}}_0 := \overline{\mathcal{T}}_0$.

Then, the refined degree of freedom structure for $\widehat{\mathcal{T}}$ is again created as per Section 3.1 for $\widehat{\mathcal{T}}$, and the associated mixed quadratic B-splines \widehat{B}_i , $i = 1, \dots, \widehat{n}$, and the refined spline space $\widehat{\mathcal{B}}$ are created following Sections 3.2.2–3.2.4. In particular,

$$\begin{aligned} \widehat{n} &= |\widehat{\mathcal{T}}_2| + |\widehat{\mathcal{T}}_1| - |\overset{\circ}{\widehat{\mathcal{T}}}_1| + |\overline{\widehat{\mathcal{T}}}_0|, \\ &= |\widehat{\mathcal{T}}_2| + 2|\mathcal{T}_1| - 2|\overset{\circ}{\mathcal{T}}_1| + |\overline{\mathcal{T}}_0|, \end{aligned} \tag{12}$$

since \mathcal{T} and $\widehat{\mathcal{T}}$ have the number of corner vertices, and the latter has twice as many boundary edges as the former.

Remark 4.1. If the \mathcal{T} contains no extraordinary vertices and no corner vertices, then $\mathcal{B} \subset \widehat{\mathcal{B}}$; this is not true in general and can be seen by considering Proposition 3.1(f) with regard to \mathcal{T} and $\widehat{\mathcal{T}}$. In brief, we opt for a non-nested refinement process here, with the non-nestedness limited to (a) neighbourhoods of extraordinary vertices and (b) faces that contain non-corner boundary vertices of valence > 2 ; these neighbourhoods shrink with each refinement step. An upshot is that the data structures in the implementation of splines are significantly simplified compared to other mixed smoothness constructions on unstructured meshes that use nested refinements [6,7].



Fig. 8. The above depicts refinement of boundary edge dofs. Figure (a) shows the edge labels for the coarse (top) and fine (bottom) meshes. Figure (b) shows how the coarse Bernstein–Bézier form of a spline f on τ (top) can be used to compute the boundary edge dofs for the new boundary edges (bottom).

4.2. Refining spline functions

Let us now outline how a spline function $f \in \mathcal{B}$ is refined to create a spline function $\widehat{f} \in \widehat{\mathcal{B}}$. We will do this graphically by outlining how the old dofs are combined to yield the new dofs. We start by addressing the updates of corner vertex dofs, then boundary edge dofs, and finally the face dofs.

In the following, we denote the old dofs for f with f_i (see Eq. (9)), and the new dofs with \widehat{f}_i . Graphically, the former will not be shown and, following our convention, the latter will be shown as filled blue circles.

4.2.1. Corner vertex dofs

As mentioned above, the number of corner vertices remains fixed during refinement and retain their labels; see Fig. 7(c) for an example. The associated dofs map from f to \widehat{f} via the identity map. That is, let f_i be the dof associated to a corner vertex of \mathcal{T} and let \widehat{f}_i be the dof associated to the same corner vertex but now as a member of $\widehat{\mathcal{T}}$. Then, we set

$$\widehat{f}_j := f_j . \tag{13}$$

4.2.2. Boundary edge dofs

Let $\tau \in \mathcal{T}$ be a boundary edge belonging to $\sigma \in \mathcal{T}_2$ that is being split into two new boundary edges $\widehat{\tau}_L, \widehat{\tau}_R \in \widehat{\mathcal{T}}_1$; see Fig. 8(a). Recall Eqs. (3) and (5) that define the local representation of mixed quadratic B-splines on a face σ that contains τ . Then, the transpose of those operations map the mixed quadratic B-spline coefficients $f_i, i = 1, \dots, n$, to the local Bernstein–Bézier coefficients on σ and, therefore, on τ . Let the Bernstein–Bézier coefficients corresponding to τ be as at the top of Fig. 8(b). Then, the dofs corresponding to $\widehat{\tau}_L$ and $\widehat{\tau}_R$ are computed as in the bottom of Fig. 8(b).

4.2.3. Face dofs

Let us now describe the refinement of the remaining dofs — the face dofs. This will be done in two steps. In *Step 1*, a prediction for the new face dofs is made. Next, in *Step 2*, a correction is applied to some of these face dofs in the unstructured regions of the mesh.

Step 1. Let $\sigma \in \mathcal{T}_2$ be refined into (4 or 3) new faces of $\widehat{\mathcal{T}}_2$ as shown in Fig. 9. Then, as in the case of boundary edge dof refinement, use the transpose of the operations in Eqs. (3) and (5) to map the mixed quadratic B-spline coefficients $f_i, i = 1, \dots, n$, to the local Bernstein–Bézier coefficients on σ . Let these coefficients be as displayed at the right in Figs. 9(a) and (b) for quadrilateral and triangular faces, respectively. Then, the new face dofs are computed by averaging a subset of these coefficients as shown.

Step 2. The above step yields the final values for almost all new face dofs, but we apply a correction to some of the new face dofs. More precisely, let all faces sharing an extraordinary vertex of \mathcal{T} be quadrilaterals, and let this extraordinary vertex be the only extraordinary vertex those faces contain (i.e., no other vertex of those faces is extraordinary). Then, the correction step is applied to the face dofs obtained by splitting these coarse quadrilaterals. We consider first the case of interior vertices and then boundary vertices.

Let the labelling of the coarse faces and the new face dofs be as in Fig. 10(a) for an interior extraordinary vertex. Moreover, let the Bernstein–Bézier coefficients for the spline f on the edge shared by σ_m and $\sigma_{\text{mod}(m,\mu)+1}$

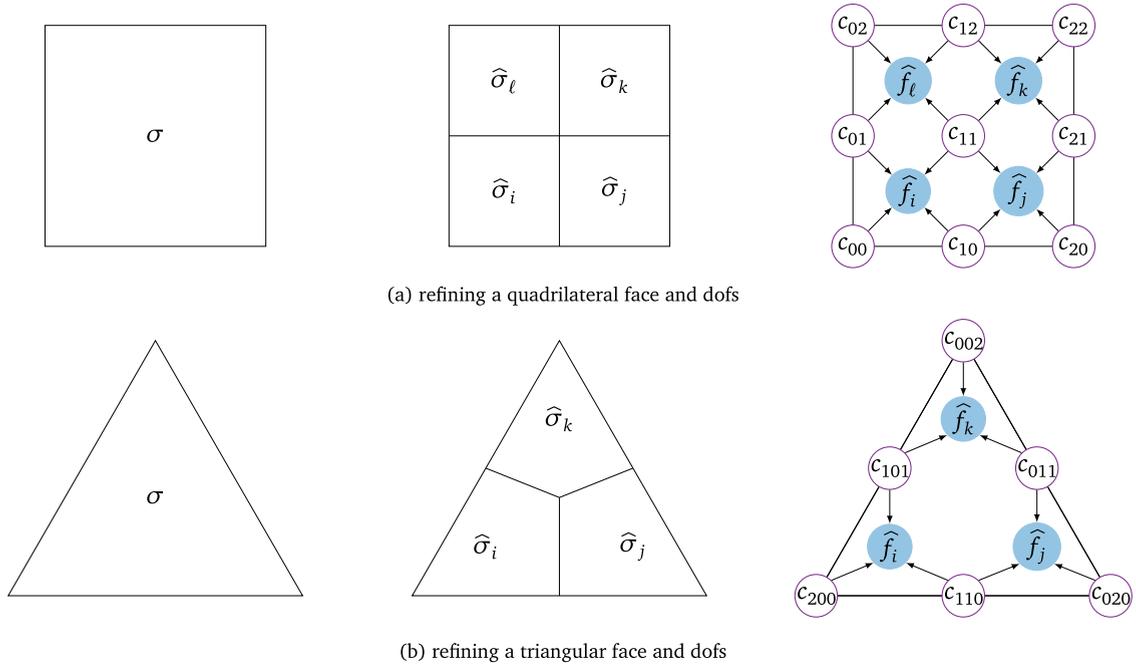


Fig. 9. The above shows *Step 1* in the refinement of face dofs. The coarse quadrilateral and triangular elements on the left are split into new quadrilaterals labelled as in the middle. In the rightmost images, it is shown how the Bernstein–Bézier form of the coarse faces are used to compute the new face dofs via averaging; the weights for the quadrilateral element are $\frac{1}{4}$ and for the triangular element are $\frac{1}{3}$.

be $c_{m,00}, c_{m,01}, c_{m,02}$ for $m = 1, \dots, \mu$. Then, the corrected face dofs $\hat{f}_{1,i}, \dots, \hat{f}_{\mu,i}$ are computed as

$$\begin{bmatrix} \hat{f}_{1,i} \\ \hat{f}_{2,i} \\ \vdots \\ \hat{f}_{\mu,i} \end{bmatrix} = \mathring{S}_\mu \left(\begin{bmatrix} c_{1,00} & c_{1,01} & c_{1,02} \\ c_{2,00} & c_{2,01} & c_{2,02} \\ \vdots \\ c_{\mu,00} & c_{\mu,01} & c_{\mu,02} \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix} - \begin{bmatrix} \hat{f}_{1,\ell} & \hat{f}_{2,j} \\ \hat{f}_{2,\ell} & \hat{f}_{3,j} \\ \vdots \\ \hat{f}_{\mu,\ell} & \hat{f}_{1,j} \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} \right) + \mathring{Q}_\mu \begin{bmatrix} \hat{f}_{1,i} \\ \hat{f}_{2,i} \\ \vdots \\ \hat{f}_{\mu,i} \end{bmatrix}, \tag{14}$$

where \mathring{S}_μ and \mathring{Q} are circulant matrices. If μ is odd, they are defined to be

$$\mathring{S}_\mu = \begin{bmatrix} 2 & -2 & 2 & \dots & -2 & 2 \\ 2 & 2 & -2 & \dots & 2 & -2 \\ & & \ddots & \ddots & \ddots & \ddots \\ -2 & 2 & -2 & \dots & 2 & 2 \end{bmatrix}, \quad \mathring{Q}_\mu = \mathbf{0}, \tag{15}$$

and if μ is even, they are defined to be

$$\mathring{S}_\mu = \begin{bmatrix} 4 - \frac{4}{\mu} & -\left(4 - \frac{8}{\mu}\right) & \left(4 - \frac{12}{\mu}\right) & \dots & 0 \\ 0 & 4 - \frac{4}{\mu} & -\left(4 - \frac{8}{\mu}\right) & \dots & \frac{4}{\mu} \\ & & \ddots & \ddots & \ddots \\ -\left(4 - \frac{8}{\mu}\right) & \left(4 - \frac{12}{\mu}\right) & -\left(4 - \frac{16}{\mu}\right) & \dots & 4 - \frac{4}{\mu} \end{bmatrix}, \quad \mathring{Q}_\mu = \frac{1}{\mu} \begin{bmatrix} 1 & -1 & \dots & -1 \\ -1 & 1 & \dots & 1 \\ & & \ddots & \ddots \\ -1 & 1 & \dots & 1 \end{bmatrix}. \tag{16}$$

Next, let the labelling of the coarse faces and the new face dofs be as in Fig. 10(a) for a boundary extraordinary vertex or a corner vertex of valence > 1 . Moreover, let the Bernstein–Bézier coefficients for the spline f on the

edge shared by σ_m and σ_{m+1} be $c_{m,00}, c_{m,01}, c_{m,02}$ for $m = 1, \dots, \mu - 1$. Then, the corrected face dofs $\widehat{f}_{1,i}, \dots, \widehat{f}_{\mu,i}$ are computed as

$$\begin{bmatrix} \widehat{f}_{1,i} \\ \widehat{f}_{2,i} \\ \vdots \\ \widehat{f}_{\mu,i} \end{bmatrix} = S_\mu^\partial \left(\begin{bmatrix} c_{1,00} & c_{1,01} & c_{1,02} \\ c_{2,00} & c_{2,01} & c_{2,02} \\ \vdots \\ c_{\mu-1,00} & c_{\mu-1,01} & c_{\mu-1,02} \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix} - \begin{bmatrix} \widehat{f}_{1,\ell} & \widehat{f}_{2,j} \\ \widehat{f}_{2,\ell} & \widehat{f}_{3,j} \\ \vdots \\ \widehat{f}_{\mu-1,\ell} & \widehat{f}_{\mu,j} \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} \right) + Q_\mu^\partial \begin{bmatrix} \widehat{f}_{1,i} \\ \widehat{f}_{2,i} \\ \vdots \\ \widehat{f}_{\mu,i} \end{bmatrix}, \tag{17}$$

where S_μ^∂ is defined to be

$$S_\mu^\partial = R_\mu + J_\mu R_\mu J_{\mu-1}, \tag{18}$$

and Q_μ^∂, R_μ and J_k are defined to be the following matrices of sizes $\mu \times \mu, \mu \times (\mu - 1)$ and $k \times k, k \geq 1$, respectively,

$$Q_\mu^\partial = \frac{1}{\mu} \begin{bmatrix} 1 & -1 & \dots & (-1)^{\mu-1} \\ -1 & 1 & \dots & (-1)^\mu \\ & \ddots & \ddots & \ddots \\ (-1)^{\mu-1} & (-1)^\mu & \dots & (-1)^{2\mu-2} \end{bmatrix},$$

$$R_\mu = \begin{bmatrix} 4 - \frac{4}{\mu} & -\left(4 - \frac{8}{\mu}\right) & \left(4 - \frac{12}{\mu}\right) & \dots & (-1)^\mu \frac{4}{\mu} \\ & \left(4 - \frac{8}{\mu}\right) & -\left(4 - \frac{12}{\mu}\right) & \dots & (-1)^{\mu+1} \frac{4}{\mu} \\ & & \left(4 - \frac{12}{\mu}\right) & \dots & (-1)^{\mu+2} \frac{4}{\mu} \\ & & & \ddots & \vdots \\ & & & & (-1)^{2\mu-2} \frac{4}{\mu} \\ & & & & 0 \end{bmatrix}, J_k = \begin{bmatrix} & & & & 1 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ 1 & & & & \end{bmatrix}$$

4.3. Properties

In general, the splines f and \widehat{f} will be different due to the non-nestedness of the spline spaces $\mathcal{B} \not\subseteq \widehat{\mathcal{B}}$. Nevertheless, the refinement steps proposed in the previous section minimize the difference between f and \widehat{f} in a specific sense. This is made more precise in the following result.

Proposition 4.2. *Let \widehat{f} be obtained from f via the refinement process outlined in Section 4.2. Then the following hold true.*

- (a) *Preserves constants: The refinement process leaves constant splines invariant.*
- (b) *Boundary preservation: Let τ be a boundary edge of \mathcal{T} and f_τ the local polynomial representation of f restricted to τ . With reference to Fig. 8, let the origin of local coordinates on τ be at the left end. Then,*

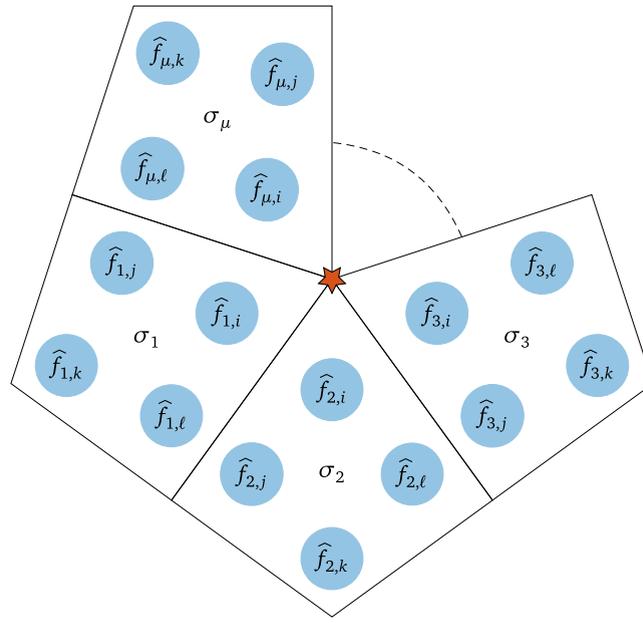
$$\widehat{f}|_{\widehat{\tau}_L} = f_\tau|_{[0,0.5]}, \quad \widehat{f}|_{\widehat{\tau}_R} = f_\tau|_{[0.5,1]}.$$

- (c) *Structured quadrilateral preservation: Let σ be a quadrilateral that contains no extraordinary vertices and no boundary vertices/edges, and f_σ the local polynomial representation of f restricted to σ . With reference to Fig. 9(a), let the local coordinates on σ be in $[0, 1]^2$ with the origin at the bottom left corner. Then,*

$$\widehat{f}|_{\widehat{\sigma}_i} = f_\sigma|_{[0,0.5]^2}, \quad \widehat{f}|_{\widehat{\sigma}_j} = f_\sigma|_{[0.5,1] \times [0,0.5]}, \quad \widehat{f}|_{\widehat{\sigma}_k} = f_\sigma|_{[0.5,1]^2}, \quad \widehat{f}|_{\widehat{\sigma}_l} = f_\sigma|_{[0,0.5] \times [0.5,1]}.$$

- (d) *Midpoint interpolation on unstructured edges: Consider the settings shown in Fig. 10(a) and (b). Denote with f_{σ_m} the local polynomial representation of f restricted to σ_m and let the local coordinates on all σ_m and $\widehat{\sigma}_{m,i}$ be in $[0, 1]^2$ with the origin at the extraordinary vertex. Then, for all $m = 1, \dots, \mu$,*

$$\widehat{f}|_{\widehat{\sigma}_{m,i}}(0, 1) = f_{\sigma_m}(0, 0.5).$$



(a) interior extraordinary vertex

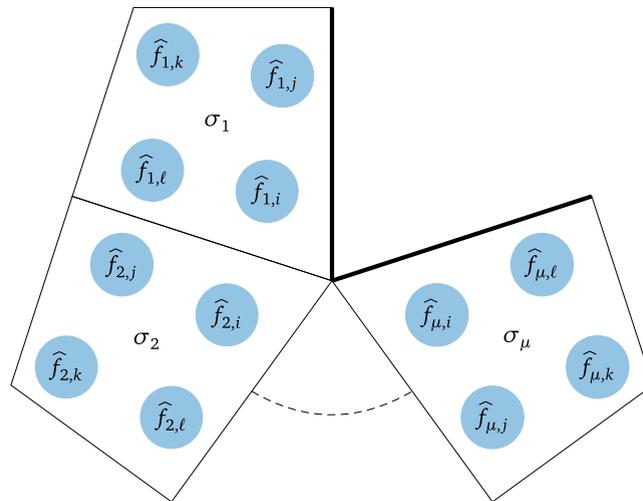


Fig. 10. The above shows *Step 2* in the refinement of face dofs. Let $\sigma_1, \dots, \sigma_\mu \in \mathcal{T}_2$ be quadrilaterals that share an extraordinary vertex and, moreover, this is the only extraordinary vertex that each face contains; figure (a) shows the case for an interior extraordinary vertex and figure (b) for a boundary extraordinary vertex. Then, let $\widehat{f}_{m,i}, \widehat{f}_{m,j}, \widehat{f}_{m,k}$ and $\widehat{f}_{m,l}$ be the face dofs obtained by splitting $\sigma_m, m = 1, \dots, \mu$; the subscripts of i, j, k, l are as in Fig. 9(a). Then, a correction is applied to the dofs $\widehat{f}_{m,i}, m = 1, \dots, \mu$, as per Eqs. (14), (15) and (16) for figure (a), and Eqs. (17) and (18) for figure (b).

Proof. The reasoning for all four properties is as below.

- (a) The refinement of corner vertex, boundary edge and face dofs outlined in Eq. (13), Figs. 8 and 9, respectively, clearly satisfy the claim. Then the only cases that remain correspond to refinement correction in Eqs. (14) and (17). We consider Eq. (17); the other case is similar. If all the coarse dofs are equal to 1, then all the fine face dofs that appear on the right hand side of Eq. (17) are also equal to 1, and so are the $c_{m,00}, c_{m,01}$ and $c_{m,02}, m = 1, \dots, \mu - 1$. Moreover, we see that the m th row of $S_\mu^\partial, m = 1, \dots, \mu$, sums to 2 for even μ and sums to $2 + (-1)^m \frac{2}{\mu}$ for odd valences. Similarly, the m th row of $Q_\mu^\partial, m = 1, \dots, \mu$, sums to 0 for

even μ and sums to $(-1)^{m+1} \frac{1}{\mu}$ for odd valences. Consequently, we see that for both odd and even valences the refinement correction leaves the face dofs invariant, i.e., the computation on the right of Eq. (17) yields new face dofs that are again equal to 1. The claim follows.

- (b) The refinement outlined in Fig. 8 corresponds to the refinement (by bisection) of univariate quadratic B-splines. Then, boundary preservation follows from this fact, and from the boundary Kronecker–Delta property and the proof of boundary-edge smoothness in Proposition 3.1.
- (c) On structured quadrilaterals, the refinement in Fig. 9 corresponds to the refinement (by bisection) of tensor product bi-quadratic B-splines. Then, this property also follows from the proof of face smoothness in Proposition 3.1.
- (d) Consider Fig. 10(a) and the notation used in Eq. (14). Then,

$$f_{\sigma_m}(0, 0.5) = \frac{c_{m,00} + 2c_{m,01} + c_{m,02}}{4}, \quad \widehat{f}|_{\widehat{\sigma}_{m,i}}(0, 1) = \frac{\widehat{f}_{m,\ell} + \widehat{f}_{m_+,j} + \widehat{f}_{m,i} + \widehat{f}_{m_+,i}}{4},$$

where $m_+ := \text{mod}(m, \mu) + 1$. If we equate these two expressions and fix all new face dofs except $\widehat{f}_{m,i}$, $m = 1, \dots, \mu$, we get a system of μ equations. For odd valences this system is full rank while for even valences it is rank deficient by 1. Nevertheless, it can be verified that in both cases Eq. (14) provides an explicit and symmetric form of the solution. The case in Fig. 10(b) follows similarly. Note that Q_μ^∂ does not affect midpoint interpolation since it lies in the nullspace of the face dof averaging above. ■

Remark 4.3. As will be shown in Section 5, the refinement correction outlined above has a positive effect on error convergence rates when performing isogeometric analysis. However, it needs to be emphasized that our refinement correction should be viewed differently from others that aim to improve error convergence by introducing refinements that intend to capture singularities in the exact solution [64] or refinements that intend to counter the lack of local approximation power [7,26]. Both these classes of refinements aim to introduce local mesh sizes that shrink faster than elsewhere. On the other hand, our mesh refinement is different as it only aims to reproduce an aspect of regular, nested, uniform mesh refinements, i.e., mid-point interpolation; see Proposition 4.2(d).

Remark 4.4. Here, we have introduced local non-nestedness around extraordinary points to prioritize simplified implementation as well as imposition of maximal smoothness wherever possible; as is shown later in Section 5, the resulting spaces also demonstrate optimal approximation in the L^2 and H^1 norms for second-order PDEs. This can be helpful in applications and offers an alternative viewpoint to earlier constructions [6,7,26] where a detailed discussion of nested unstructured spline spaces was presented. Those constructions were presented for C^2/C^1 bicubic splines and also demonstrated optimal approximation behaviour, albeit at the expense of more involved implementations and, after refinement, the need to impose non-maximal smoothness even in structured regions of the mesh. In fact, our approach can be seen as motivated by the same considerations as the “design space” from [7] (e.g., ease of implementation, maximal smoothness imposition, standard control nets). Nevertheless, the resulting refinement schemes are significantly different – the “design space” was formulated for cubic splines, had no notion of refinement correction and was unsuitable for analysis due to poor approximation behaviour (specifically, linear convergence).

4.4. Geometric consistency errors

The non-nestedness of the refinement implies that there will be localized changes to the geometry of interest in the neighbourhoods of extraordinary points. Nevertheless, as sketched in the Appendix, a subdivision interpretation of the refinement scheme shows convergence in the limit of refinement. However, in practice, this limit of refinement is never reached so a simple but interesting quantity to measure is the distance between its successive refinements. This was previously done in [7] for a non-nested refinement scheme. A general quantification of this distance depends on the particular choice of geometry. For instance, a planar surface built from a mesh with only interior extraordinary points will be geometrically identical at all refinement levels, only its parameterization will change. Taking as an example a non-planar surface from the Shape Obstacle Course [65], we investigate the maximum distance between successively refined representations. The coarsest representation (containing 36 quadrilateral elements) and its two successive refinements are shown in Fig. 11. The maximum distance between the geometries in figures (a) and (b) is 1.1×10^{-2} , and between the geometries in figures (b) and (c) is 5.4×10^{-3} . These results are of the same order of magnitude as the ones from [7].

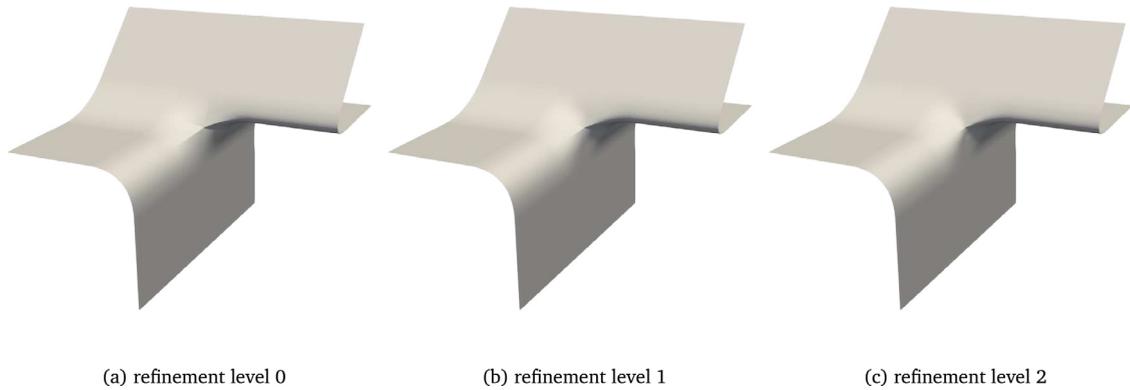


Fig. 11. Borrowing a surface from the Shape Obstacle Course [65], we show how the geometric changes due to the non-nested refinements are localized around the extraordinary point. The reduction in the footprint of C^0 smoothness with refinement is clearly visible in the above figures. The maximum distance between the geometry at refinement level 0 and 1 is 1.1×10^{-2} , and between the geometry at refinement level 1 and 2 is 5.4×10^{-3} , comparable to the ones from [7].

5. Numerical tests

In this section we present several numerical tests using the mixed quadratic B-splines presented in this document. We begin in Section 5.1 by presenting numerical experiments to investigate the approximation properties of the spline spaces. We do this by performing convergence tests on quadrilateral meshes that, at the coarsest level, contain one interior vertex that is an extraordinary vertex.

Remark 5.1. As per Section 4, since all triangular faces of a coarse mesh are converted to quadrilateral faces during mesh refinement, they have no impact on the convergence behaviour that the spline spaces demonstrate. Nevertheless, the last example in Section 5.1 presents results on a family of meshes that are pure triangulations at all refinement levels; see Fig. 14. In general we expect that the numerical results will be better for meshes with fewer triangles. If controlling the number of triangles is in the hands of the user, pure quadrangulations should be chosen.

Remark 5.2. In the examples depicting reconstructed trimmed surfaces (Figs. 2, 13 and 16), the coarsest watertight reconstruction is done by generating a quad-tri mesh \mathcal{T} using Rhinoceros[®], building the spline space on \mathcal{T} as per Section 3, and then solving for the spline control points by requiring interpolation of the face centres, boundary-edge midpoints and corner vertices. An exception is the hole boundary of Fig. 13 where the corresponding boundary control points are chosen to exactly represent a circle. In general, more sophisticated fitting algorithms can be formulated and employed, but this is out of this manuscript's scope. In particular, if the B-spline/NURBS representations of the trimming curves are available, then they can be directly used to set the boundary control points.

Subsequently, we present the first example that employs a trimmed geometry created using the CAD software Rhinoceros[®]— convergence tests for a Poisson problem for a plate with a hole. Next, in Section 5.2, we present a classical linear elasticity benchmark — an infinite plate under horizontal tension with a circular hole at its centre. Finally, we present a sample workflow for performing isogeometric analysis on trimmed surfaces by solving the Laplace–Beltrami eigenvalue problem for a bracket geometry in Section 5.3.

In the following tests, we first build a spline geometry $\Omega \subset \mathbb{R}^d$ using Eq. (11). Representing the piecewise-defined geometric map from \mathcal{T} to Ω as $F : \mathcal{T} \rightarrow \Omega$, the space of mixed B-splines on Ω is defined through a push-forward,

$$\mathcal{S} = \{f \in C^0(\Omega) : f \circ F \in \mathcal{B}\} . \quad (19)$$

We will also use the following space with homogeneous boundary conditions,

$$\mathcal{S}_0 = \{f \in \mathcal{S} : f|_{\partial\Omega} = 0\} . \quad (20)$$

5.1. Convergence tests

We start by solving an H^1 projection problem to numerically study the approximation offered by mixed quadratic splines under mesh refinement. The problem studied is: find $f_h \in \mathcal{S}$ such that

$$\begin{aligned}
 - \int_{\Omega} \nabla g \cdot \nabla f \, d\Omega &= \int_{\Omega} g \nabla^2 f_{\text{exact}} \, , \quad \forall g \in \mathcal{S}_0 \, , \\
 \int_{\partial\Omega} g f \, d\Omega &= \int_{\partial\Omega} g f_{\text{exact}} \, , \quad \forall g \in \{g \in \mathcal{S} : g|_{\partial\Omega} \neq 0\} \, .
 \end{aligned}
 \tag{21}$$

The exact solution for all convergence tests is chosen to be

$$f_{\text{exact}}(x, y) = \sin\left(\pi x + \frac{\pi}{3}\right) \sin\left(\pi y + \frac{\pi}{5}\right) \, .
 \tag{22}$$

Remark 5.3. Imposition of boundary conditions for the mixed quadratic splines is straightforward, given the boundary Kronecker–Delta property from Proposition 3.1.

The first set of tests is carried out on quadrilateral meshes containing a single extraordinary vertex. The spline geometries and the control points at the coarsest level are shown on the left in Fig. 12. The error convergence with mesh refinement is shown in the plots on the right, with the error norms plotted against the inverse of the square root of the number of dofs. We plot six different errors:

- $\|e\|_{\max}$, $\|e\|_0$ and $\|e\|_1$: the $L^\infty(\Omega)$, $L^2(\Omega)$ and $H^1(\Omega)$ norms of the error with refinements performed as in Section 4;
- $\|\bar{e}\|_{\max}$, $\|\bar{e}\|_0$ and $\|\bar{e}\|_1$: the $L^\infty(\Omega)$, $L^2(\Omega)$ and $H^1(\Omega)$ norms of the error with refinements performed as in Section 4 but *without* the refinement correction from Eq. (14).

As shown in the plots, the mixed quadratic splines demonstrate optimal convergence rates in the L^2 and H^1 norms when the refinement is performed as in Section 4. On the other hand, without the refinement correction from Eq. (14), the convergence rates deteriorate rapidly with increasing valence of the extraordinary vertex. This same deterioration is also observed for the error measured in the L^∞ norm and, except for the valence 3 geometry, the refinement correction leads to improved but quadratic convergence rates for the error, i.e., suboptimal by one order.

Remark 5.4. Following the numerical experiments, we conjecture that the spline spaces constructed here possess optimal approximation properties (i.e., optimal error estimates can be derived for them). Note that, since the refinements are non-nested, it is in general unclear on how errors on successively refined meshes should be compared during numerical experiments. More generally, further research is needed into the formulation of appropriate numerical experiments and metrics appropriate for investigating approximation with non-nested spaces. For instance, it is expected that integral error metrics (L^2 and H^1 norms) are less sensitive to changes in parameterization than local error metrics (L^∞ norm).

Remark 5.5. Valence 3 extraordinary points are an isolated case where spline constructions seem to perform well, even if they demonstrate suboptimal error convergence for higher valences. This is true for several constructions from the literature and also seems to be the case for the no-refinement-correction case shown in the first row of Fig. 12.

In the second set of tests, we repeat the convergence tests but now for the watertight reconstruction of a trimmed geometry. We start by creating a square plate with a hole in Rhinoceros[®] using trimming; the trimmed representation of the geometry is shown in Fig. 13(a). Using the built-in meshing tool in Rhinoceros[®], we immediately obtain the quad-tri mesh shown in Fig. 13(b) and are able to reconstruct a so-called watertight and editable spline geometry using mixed quadratic splines; the latter is shown in Fig. 13(c). The error convergence plot is presented in Fig. 13(d) and optimal convergence rates are observed in both the $L^2(\Omega)$ and $H^1(\Omega)$ norms.

A final set of convergence tests on $\Omega = [0, 1]^2$ is conducted. Here, we create two sequences of refined meshes $\overline{\mathcal{T}}^k$ and \mathcal{T}^k , $k = 0, 1, \dots$

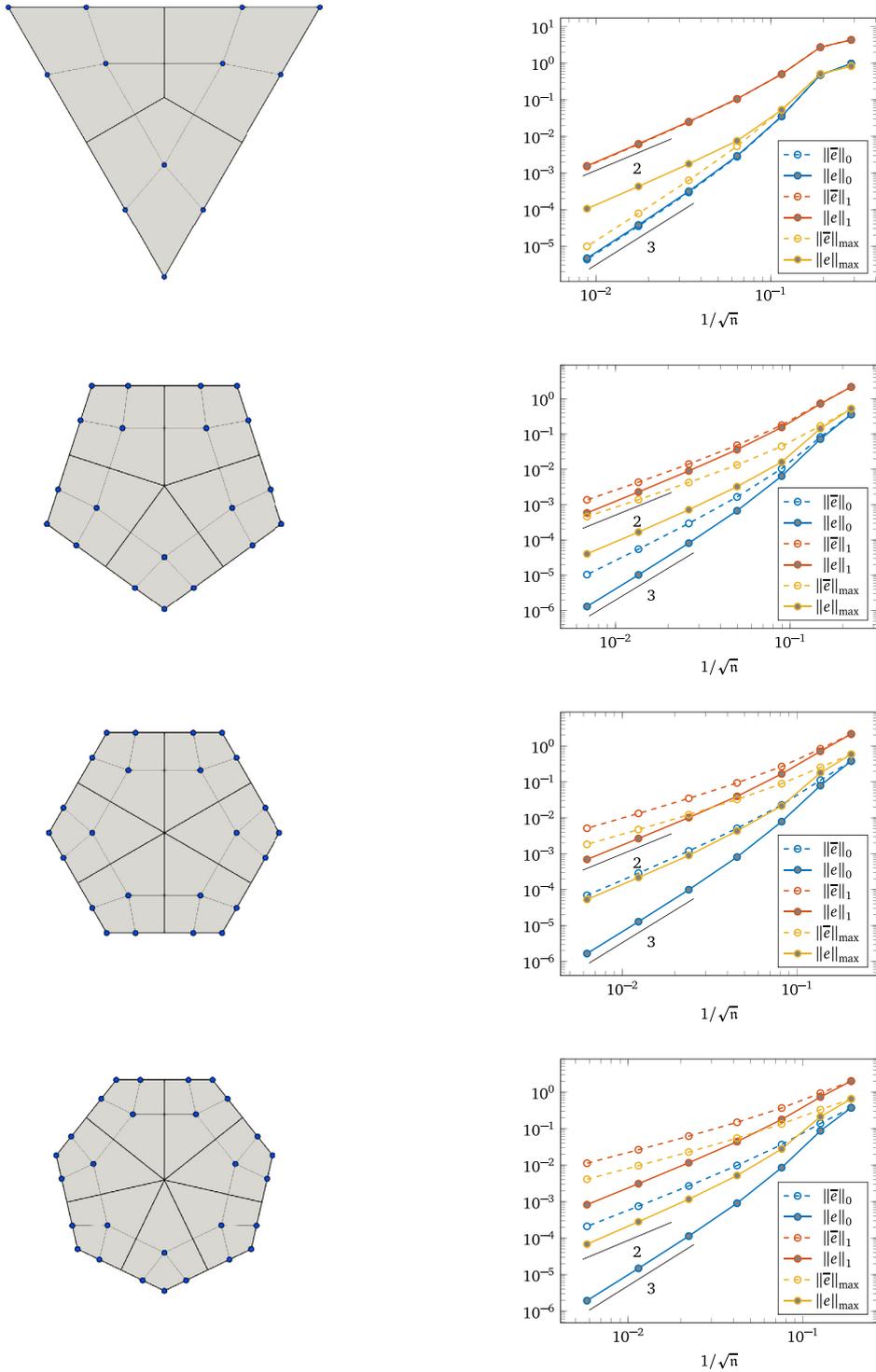


Fig. 12. The above plots show the error convergence with mesh refinement when solving the problem in Eq. (21) on the different spline geometries in the left column. Error e corresponds to when refinement is performed as in Section 4, and error \bar{e} corresponds to when the refinement correction in Eq. (14) is not implemented. The refinement correction is able to recover optimal convergence rates in the L^2 and H^1 norms on all geometries. Error in the L^∞ norm is more sensitive to parameterization changes (cf. Remark 5.4). Except for valence 3 (cf. Remark 5.5), refinement correction also leads to improved but suboptimal convergence in the L^∞ norm.

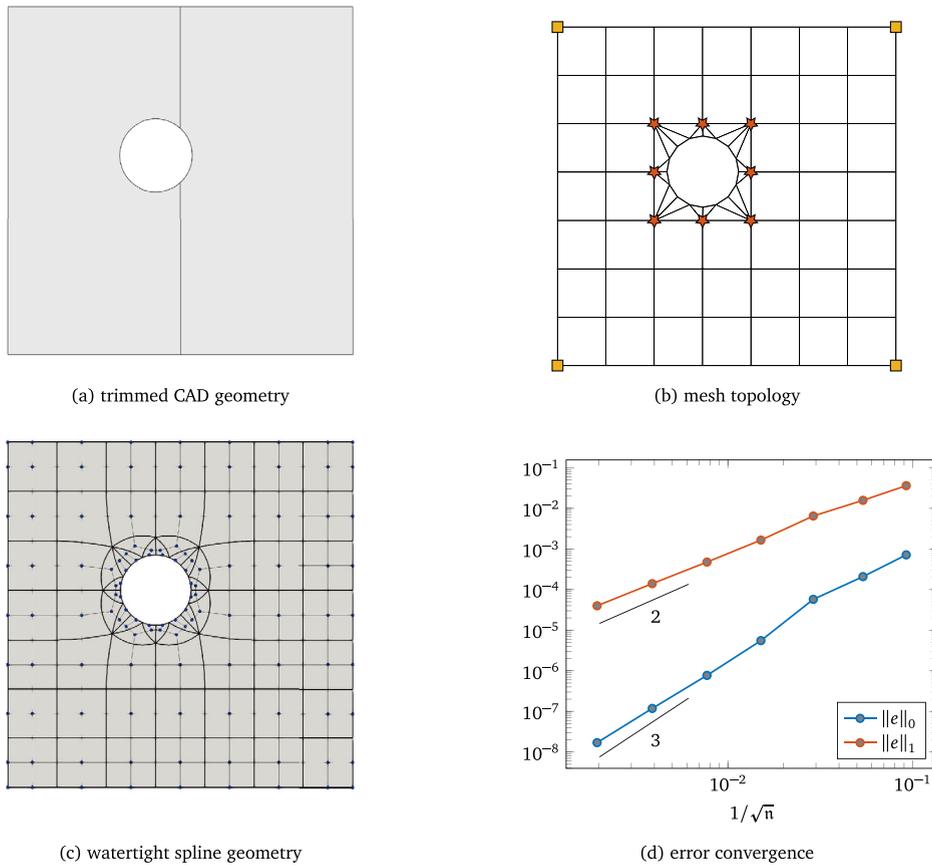


Fig. 13. The above shows the error convergence plot for a trimmed geometry and the problem in Eq. (21). Figure (a) shows the trimmed geometry created in Rhinoceros® and (b) shows the quad-tri mesh created for it automatically by Rhinoceros®. Mixed quadratic splines on the mesh in (b) are used to create a watertight and editable representation of the trimmed geometry, the former being shown in figure (c). Figure (d) shows that optimal approximation behaviour is observed during mesh refinement. At the coarsest level, the quad-tri mesh contains 69 faces, and the total number of quadratic spline degrees of freedom for this mesh are 117 (69 interior, 48 boundary).

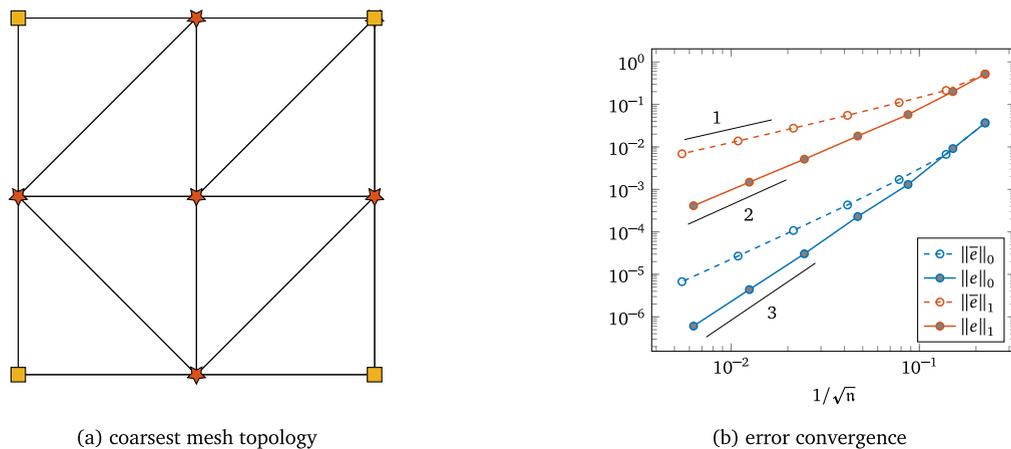


Fig. 14. The above compares the error convergence plots when an initially pure triangulation is refined as per the process in Section 4 (error denoted by e) versus when a sequence of refined pure triangulations is built (error denoted by \bar{e}). The error converges sub-optimally by 1 order on the sequence of triangulated meshes, but it converges optimally when the refinements are carried out as per Section 4.

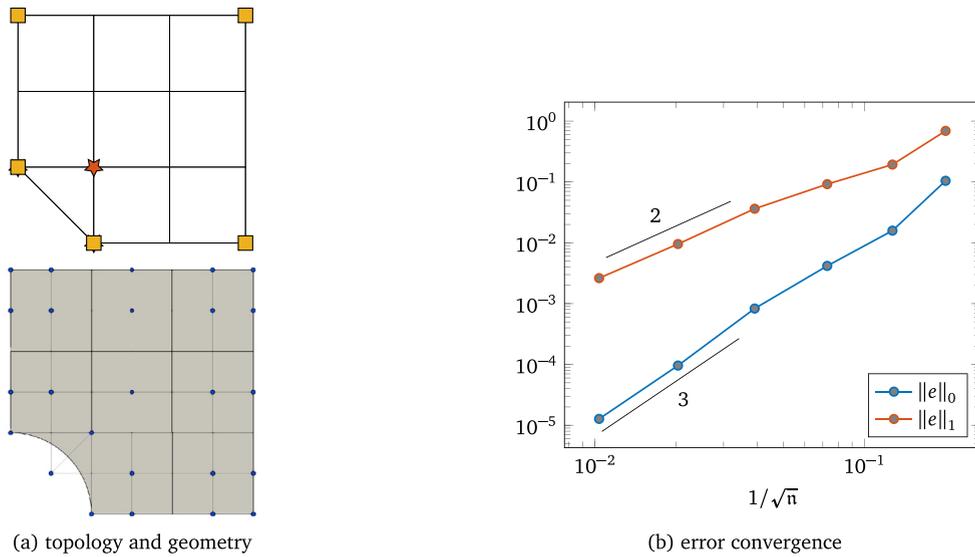


Fig. 15. The above shows the error convergence plot for the problem in Eq. (24). Figure (a) shows the mesh topology and spline geometry that model a quarter of a plate with a hole at the centre of it. Rational mixed quadratic splines on the mesh in the top figure in (a) can be used to create the geometry shown below with an exact representation of the boundary of the circular hole. Figure (b) shows that optimal approximation behaviour is observed during mesh refinement for both the L^2 and H^1 norms of the error. At the coarsest level, the quad-tri mesh contains 9 faces, and the total number of quadratic spline degrees of freedom for this mesh are 25 (9 interior, 16 boundary).

- The mesh family $\overline{\mathcal{T}}^k$ is built by creating a uniform Type I triangulation [66] of Ω with 2^{k+3} triangles, and then flipping the edge shared by the two triangles that contain the origin. This edge flipping ensures that there is one interior vertex of \mathcal{T} with an odd valence, which in turn implies the linear independence of the mixed quadratic B-splines from Proposition 3.1.
- The mesh family \mathcal{T}^k is built by choosing $\mathcal{T}^0 = \overline{\mathcal{T}}^0$. Then, for $k \geq 1$, \mathcal{T}^k is obtained by refining \mathcal{T}^{k-1} as per Section 4.

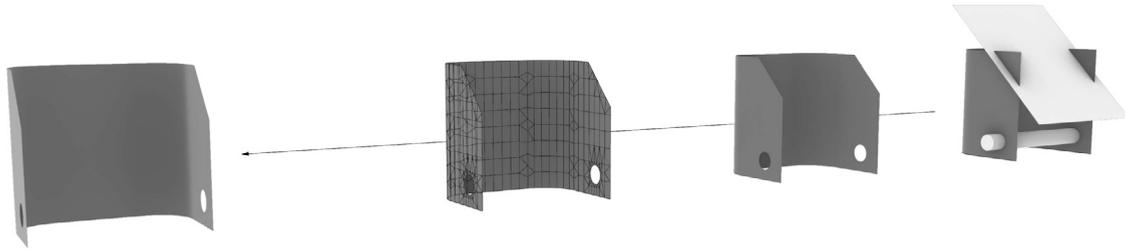
See Fig. 14(a) to see the topology and geometry of coarsest meshes $\mathcal{T}^0 = \overline{\mathcal{T}}^0$. The error convergence is shown in Fig. 14(b). As can be seen, the convergence on mesh family $\overline{\mathcal{T}}^k$ is suboptimal by 1 order in both L^2 and H^1 norms, while the convergence on mesh family \mathcal{T}^k is optimal.

5.2. Infinite plate with a circular hole

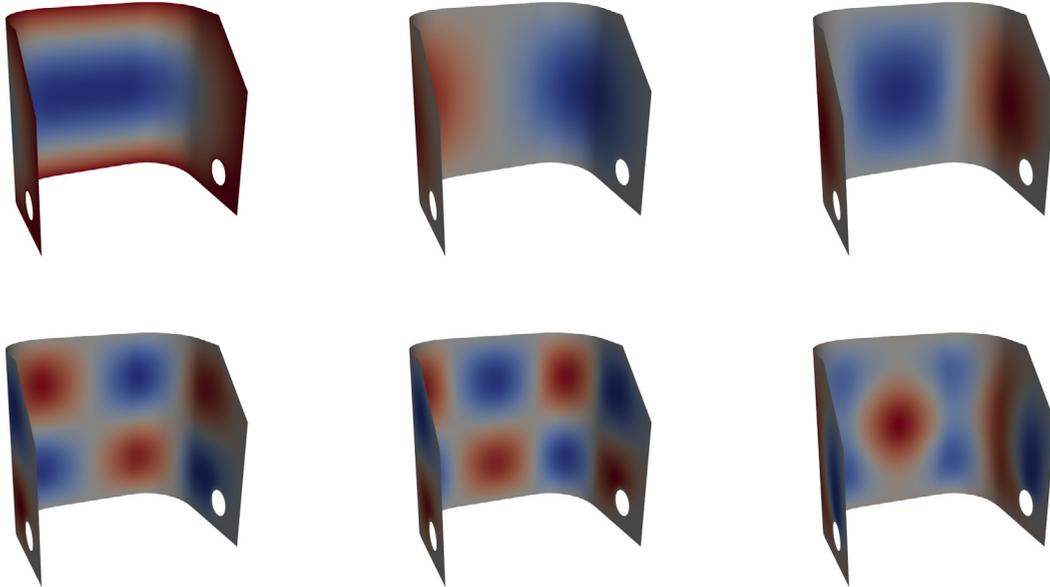
For the second set of numerical tests, we simulate the well-known linear elasticity benchmark where an infinite plate with a circular hole is subjected to tension at infinity assuming plane strain. Usually, symmetry conditions are utilized to work with only a finite quarter of the plate; we can model this quarter plate with the quadrilateral-triangle mesh in Fig. 15(a). In particular, we can exactly represent the circular hole geometry at the centre of the plate by choosing weights and values of the boundary dofs such that the boundary is exactly circular; see, e.g., [67]. See [68] for a detailed discussion of this problem.

With the displacements in a Cartesian basis denoted with $\mathbf{u} = (u_1, u_2)$ and $\boldsymbol{\varepsilon}$ and \mathbf{D} defined as

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \begin{bmatrix} u_{1,1} \\ u_{2,2} \\ u_{1,2} + u_{2,1} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix},$$



(a) Workflow: surface trimming (right) to watertight spline (left)



(b) Laplace–Beltrami: select eigenmodes

Fig. 16. The figure in (a) shows an automatic workflow that is enabled by mixed quadratic splines: boolean operations on the right yield a trimmed geometry for which a quad-tri mesh can be created in Rhinoceros[®], and the latter mesh can be utilized to create a watertight and editable representation of the trimmed geometry. Figure (b) shows some of the eigenmodes corresponding to the Laplace–Beltrami eigenvalue problem in Eq. (26). Shown are modes 1, 2, 3 (top) and 9, 12, 14 (bottom). The quad-tri mesh generated by Rhinoceros[®] contains 414 faces, and the total number of quadratic spline degrees of freedom for this mesh are 540 (414 interior, 126 boundary).

the weak form of the problem we solve is: find $\mathbf{u} \in \mathcal{S}^2$ such that

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w})^T \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{u}) \, d\Omega = \mathbf{0}, \quad \forall \mathbf{w} \in \mathcal{S}_0^2, \tag{23}$$

$$\int_{\Omega} \mathbf{w} \cdot \mathbf{u} \, d\Omega = \int_{\Omega} \mathbf{w} \cdot \mathbf{u}_{\text{exact}} \, d\Omega, \quad \forall \mathbf{w} \in \{g \in \mathcal{S} : g|_{\partial\Omega} \neq 0\}^2 \dots$$

The exact solution to the infinite plate problem in polar coordinates is

$$u_{1,\text{exact}}(r, \theta) = \frac{TR}{9\mu} \left[(\kappa + 1) \frac{r}{R} \cos(\theta) + \frac{2R}{r} ((\kappa + 1) \cos(\theta) + \cos(3\theta)) - 2 \frac{R^3}{r^3} \cos(3\theta) \right], \tag{24}$$

$$u_{2,\text{exact}}(r, \theta) = \frac{TR}{9\mu} \left[(\kappa - 3) \frac{r}{R} \sin(\theta) + \frac{2R}{r} ((-\kappa + 1) \sin(\theta) + \sin(3\theta)) - 2 \frac{R^3}{r^3} \sin(3\theta) \right].$$

Here, R is the radius of the circular hole, T is the horizontal tension applied at infinite and $\kappa = 3 - 4\nu$ for plane strain. The problem parameters are chosen to be

$$\nu = 0.3, \quad E = 10^6, \quad T = 10^6, \quad R = 1, \quad \lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \quad \mu = \frac{E}{2(1 + \nu)}. \quad (25)$$

5.3. Laplace–Beltrami eigenvalue problem on a trimmed bracket

As the last set of numerical approximation examples, we demonstrate a sample workflow for working with curved trimmed geometries; see Fig. 16. We start by creating a bracket geometry by trimming an extruded surface with a cylinder and an oblique plane in the rightmost image in Fig. 16(a). The image to its left is the trimmed, non-watertight and non-editable geometry. The image to its left shows the quad-tri mesh created by an in-built meshing tool of Rhinoceros[®], and finally the leftmost image shows the reconstructed, watertight and editable spline geometry on this quad-tri mesh. Note that this is an entirely automated process that only requires that a watertight quad-tri mesh be generated by Rhinoceros[®]; cf. Remark 1.3. Once this is done, we are able to solve a Laplace–Beltrami eigenvalue problem [69] on the mixed quadratic spline geometry. This problem is defined as: find $f \in \mathcal{S}_0$ and $\lambda \in \mathbb{R}$ such that

$$\int_{\Omega} \nabla_{\Omega} g \cdot \nabla_{\Omega} f \, d\Omega = \int_{\Omega} g f \, d\Omega, \quad \forall g \in \mathcal{S}_0. \quad (26)$$

Select eigenmodes for the problem are shown in Fig. 16(b).

6. Conclusions

We have presented an analysis-suitable construction of quadratic splines on quadrilateral-triangle meshes. These are at most biquadratic piecewise-polynomial splines and have mixed smoothness — C^1 in structured regions of the mesh and C^0 elsewhere. The splines possess several B-spline-like properties as shown in Proposition 3.1 and especially simplify boundary condition imposition. We have proposed a non-nested refinement scheme for these splines that considerably simplifies their implementation, shrinks the neighbourhoods of C^0 smoothness with mesh refinement, preserves splines on domain boundaries, and demonstrates excellent approximation behaviour for second-order problems. Quadrilateral-triangle meshes can be constructed for trimmed surfaces with relative ease; this makes watertight and editable reconstructions of and simulations on the latter a viable application of our approach. We have presented first numerical examples to demonstrate this workflow.

The construction we have presented is simple and can be extended in several directions. We briefly mention some of them here and, where applicable, outline the ongoing and completed work. Firstly, even though the ideas we present here are for bivariate splines, the extension to trivariate splines on hexahedral-tetrahedral meshes is straightforward. The incorporation of local-refinement on top of the construction would increase the flexibility of the method, and in the structured regions of the mesh this could be done using a hierarchical B-spline-type approach. Another interesting direction is the extension of the approach to higher polynomial degrees. From an applications point of view, our splines can be particularly useful in metal forming simulations [70] with trimmed CAD surfaces and a Reissner–Mindlin shell model. This is a manufacturing process for creating objects by deforming initially planar sheet metal, and it often involves C^0 creases in both the geometry and the solution. Thus simulations of this process are where C^1/C^0 spline constructions on arbitrary topology meshes can be helpful. Here, another interesting but minor extension would be to incorporate creased edges as another user/simulation-specified feature; see Remark 2.1.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The author was supported by project number 212.150 funded through the Veni research programme by the Dutch Research Council (NWO), The Netherlands. All comments from the anonymous reviewers are gratefully acknowledged.

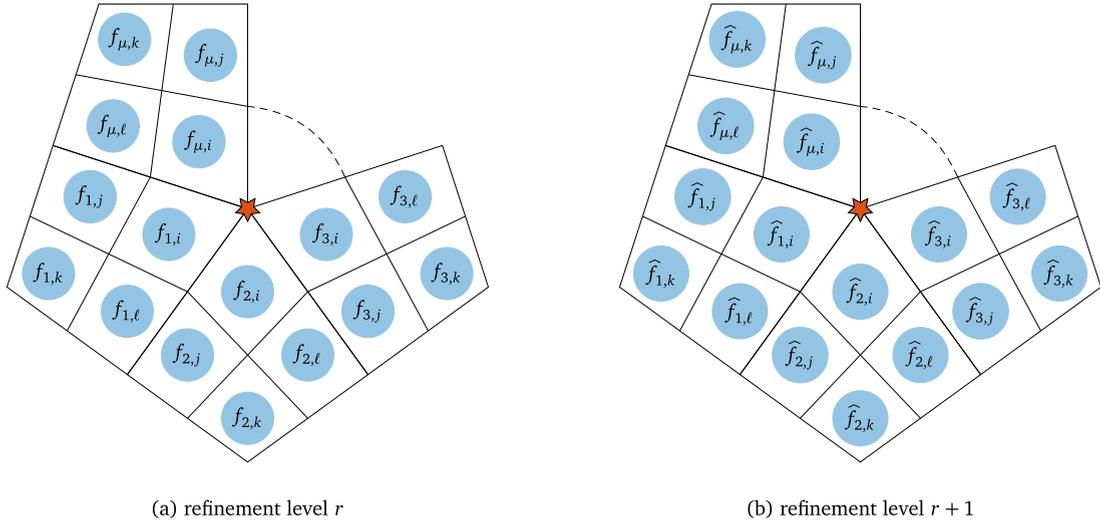


Fig. A.17. The above figures (a) and (b) show the 4μ dofs that are in the immediate neighbourhood of a given extraordinary point at refinement level r and $r + 1$, respectively, for some $r \geq 0$. (The figure has only topological information; the neighbourhoods have been drawn with the same sizes only to make the dof labels visible.) As shown in Eq. (A.1), the dofs in (b) are obtained from those in (a) by a linear mapping which can be deduced from Section 4.

Appendix. Convergence of the geometry in the limit of refinement

Our refinement scheme can also be interpreted as a subdivision scheme. This interpretation and the theory of subdivision surfaces [10] can be used to show that, in the limit, the refinements converge to a C_0^1 surface, i.e., a surface that is C^1 everywhere except at extraordinary points where it is only C^0 . We sketch the argument here for an interior extraordinary point.

For some refinement level $r \geq 0$, consider the 4μ dofs in the immediate neighbourhood of an interior extraordinary point of valence μ as shown in Fig. A.17(a); the 4μ dofs in the immediate neighbourhood at refinement level $r + 1$ are shown in Fig. A.17(b). With the dofs labelled as in Fig. A.17, we can find a matrix A such that

$$\begin{bmatrix} \widehat{f}_i \\ \widehat{f}_j \\ \widehat{f}_k \\ \widehat{f}_\ell \end{bmatrix} = A \begin{bmatrix} f_i \\ f_j \\ f_k \\ f_\ell \end{bmatrix}, \tag{A.1}$$

where

$$f_m = \begin{bmatrix} f_{1,m} \\ f_{2,m} \\ \vdots \\ f_{\mu,m} \end{bmatrix}, \quad \widehat{f}_m = \begin{bmatrix} \widehat{f}_{1,m} \\ \widehat{f}_{2,m} \\ \vdots \\ \widehat{f}_{\mu,m} \end{bmatrix}, \quad m \in \{i, j, k, \ell\}. \tag{A.2}$$

The matrix A can be deduced from Eq. (14) and its entries depend only on μ . In particular, the matrix has a structure such that Eq. (A.1) can be broken down into two pieces,

$$\widehat{f}_i = A_0 f_i, \quad \begin{bmatrix} \widehat{f}_j \\ \widehat{f}_k \\ \widehat{f}_\ell \end{bmatrix} = A_1 \begin{bmatrix} f_i \\ f_j \\ f_k \\ f_\ell \end{bmatrix}. \tag{A.3}$$

Here, A_1 contains (repeated) submatrices of the standard knot refinement matrix for bi-quadratic C^1 B-splines [67] and is stochastic (i.e.g, all rows sum to 1). The matrix A_0 can be deduced from Eq. (14) and the explicit expressions are provided below for odd μ ,

$$A_0 = \begin{bmatrix} \frac{\mu+1}{2\mu} & \frac{1}{2\mu} & \frac{1}{2\mu} & \cdots & \frac{1}{2\mu} & \frac{1}{2\mu} \\ \frac{1}{2\mu} & \frac{\mu+1}{2\mu} & \frac{1}{2\mu} & \cdots & \frac{1}{2\mu} & \frac{1}{2\mu} \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ \frac{1}{2\mu} & \frac{1}{2\mu} & \frac{1}{2\mu} & \cdots & \frac{1}{2\mu} & \frac{\mu+1}{2\mu} \end{bmatrix} \quad (\text{A.4})$$

and even μ ,

$$A_0 = \begin{bmatrix} \frac{2\mu+1}{4\mu} & \frac{5}{4\mu} & \frac{1}{4\mu} & \cdots & \frac{1}{4\mu} & \frac{5}{4\mu} \\ \frac{5}{4\mu} & \frac{2\mu+1}{4\mu} & \frac{5}{4\mu} & \cdots & \frac{5}{4\mu} & \frac{1}{4\mu} \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ \frac{5}{4\mu} & \frac{1}{4\mu} & \frac{5}{4\mu} & \cdots & \frac{5}{4\mu} & \frac{2\mu+1}{4\mu} \end{bmatrix} \quad (\text{A.5})$$

Note that A_0 is also stochastic, and therefore so is A . Consequently, the dominant eigenvalue of A is equal to 1 and, furthermore, the above structure of A can be used to verify that its subdominant eigenvalue is equal to 0.5 for all $\mu \geq 3$. Consequently, this refinement scheme is convergent [10].

References

- [1] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Comput. Methods Appl. Mech. Engrg.* 194 (2005) 4135–4195.
- [2] T. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Courier Corporation, 2012.
- [3] G. Farin, J. Hoschek, M.-S. Kim, *Handbook of Computer Aided Geometric Design*, Elsevier, 2002.
- [4] P.T. Boggs, A. Althsuler, A.R. Larzelere, E.J. Walsh, R.L. Clay, M.F. Hardwick, *Dart system analysis.*, Technical report, Sandia National Laboratories, 2005.
- [5] B. Marussig, T.J. Hughes, A review of trimming in isogeometric analysis: challenges, data exchange and simulation aspects, *Arch. Comput. Methods Eng.* 25 (4) (2018) 1059–1127.
- [6] X. Wei, Y. Zhang, D. Toshniwal, H. Speleers, X. Li, C. Manni, J.A. Evans, T. Hughes, Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis, *Comput. Methods Appl. Mech. Engrg.* 341 (2018) 609–639.
- [7] D. Toshniwal, H. Speleers, T. Hughes, Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations, *Comput. Methods Appl. Mech. Engrg.* 327 (2017) 411–458.
- [8] K.J. Koh, D. Toshniwal, F. Cirak, An optimally convergent C^1 spline construction for unstructured quadrilateral and hexahedral meshes, (Preprint Available on Request).
- [9] T.J. Hughes, G. Sangalli, T. Takacs, D. Toshniwal, Smooth multi-patch discretizations in isogeometric analysis, in: *Handbook of Numerical Analysis*, Elsevier, 2020, <http://dx.doi.org/10.1016/bs.hna.2020.09.002>, URL <http://www.sciencedirect.com/science/article/pii/S1570865920300144>.
- [10] J. Peters, U. Reif, *Subdivision Surfaces*, Springer-Verlag, 2008.
- [11] C. Loop, *Smooth subdivision surfaces based on triangles*, Master's Thesis, University of Utah, 1987.
- [12] J. Stam, Evaluation of loop subdivision surfaces, in: *SIGGRAPH'98 CDROM Proceedings*, Citeseer, 1998.
- [13] E. Catmull, J. Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, *Comput. Aided Des.* 10 (6) (1978) 350–355.
- [14] J. 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*, ACM Press, 1998, pp. 395–404.
- [15] D. Doo, M. Sabin, Behaviour of recursive division surfaces near extraordinary points, *Comput. Aided Des.* 10 (6) (1978) 356–360.
- [16] J. Stam, C. Loop, Quad/triangle subdivision, in: *Computer Graphics Forum*, 22, (1) Wiley Online Library, 2003, pp. 79–85.
- [17] F. Cirak, M. Ortiz, P. Schröder, Subdivision surfaces: A new paradigm for thin-shell finite-element analysis, *Internat. J. Numer. Methods Engrg.* 47 (2000) 2039–2072.
- [18] F. Cirak, M.J. Scott, E.K. Antonsson, M. Ortiz, P. Schröder, Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision, *Comput. Aided Des.* 34 (2) (2002) 137–148.
- [19] D. Burkhart, B. Hamann, G. Umlauf, Iso-geometric Finite Element Analysis Based on Catmull-Clark : ubdivision Solids, 29(5) 1575–1584, <http://dx.doi.org/10.1111/j.1467-8659.2010.01766.x>, URL <http://doi.wiley.com/10.1111/j.1467-8659.2010.01766.x>.
- [20] T. Nguyen, K. Karčiauskas, J. Peters, A comparative study of several classical, discrete differential and isogeometric methods for solving Poisson's equation on the disk, *Axioms* 3 (2014) 280–299.

- [21] X. Wei, Y. Zhang, T. Hughes, M. Scott, Truncated hierarchical Catmull–Clark subdivision with local refinement, *Comput. Methods Appl. Mech. Engrg.* 291 (2015) 1–20.
- [22] X. Li, X. Wei, Y.J. Zhang, Hybrid non-uniform recursive subdivision with improved convergence rates, *Comput. Methods Appl. Mech. Engrg.* 352 (2019) 606–624.
- [23] U. Reif, A refineable space of smooth spline surfaces of arbitrary topological genus, *J. Approx. Theory* 90 (1997) 174–199.
- [24] U. Reif, TURBS-topologically unrestricted rational B-splines, *Constr. Approx.* 14 (1998) 57–77.
- [25] T. Nguyen, J. Peters, Refinable C^1 spline elements for irregular quad layout, *Comput. Aided Geom. Design* 43 (2016) 123–130.
- [26] H. Casquero, X. Wei, D. Toshniwal, A. Li, T. Hughes, J. Kiendl, Y.J. Zhang, Seamless integration of design and kirchhoff–love shell analysis using analysis-suitable unstructured T-splines, *Comput. Methods Appl. Mech. Engrg.* 360 (2020) 112765.
- [27] C. Zimmermann, D. Toshniwal, C.M. Landis, T.J. Hughes, K.K. Mandadapu, R.A. Sauer, An isogeometric finite element formulation for phase transitions on deforming surfaces, 2019.
- [28] H. Casquero, C. Bona-Casas, D. Toshniwal, T.J. Hughes, H. Gomez, Y.J. Zhang, The divergence-conforming immersed boundary method: Application to vesicle and capsule dynamics, *J. Comput. Phys.* 425 (2021) 109872, <http://dx.doi.org/10.1016/j.jcp.2020.109872>, URL <http://www.sciencedirect.com/science/article/pii/S002199912030646X>.
- [29] D. Toshniwal, H. Speleers, R. Hiemstra, T.R. Hughes, Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis, *Comput. Methods Appl. Mech. Engrg.* 316 (2017) 1005–1061.
- [30] H. Speleers, D. Toshniwal, A general class of C^1 smooth rational splines: Application to construction of exact ellipses and ellipsoids, *Comput. Aided Des.* 132 (2021) 102982, <http://dx.doi.org/10.1016/j.cad.2020.102982>, URL <http://www.sciencedirect.com/science/article/pii/S0010448520301755>.
- [31] D. Toshniwal, T.J. Hughes, Isogeometric discrete differential forms: Non-uniform degrees, Bézier extraction, polar splines and flows on surfaces, *Comput. Methods Appl. Mech. Engrg.* 376 (2021) 113576.
- [32] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, T. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, *Comput. Methods Appl. Mech. Engrg.* 254 (2013) 197–221.
- [33] M. Bercovier, T. Matskewich, Smooth Bézier Surfaces over Unstructured Quadrilateral Meshes, in: *Lecture Notes of the Unione Matematica Italiana*, Springer International Publishing, URL <https://www.springer.com/us/book/9783319638409>.
- [34] M. Kapl, V. Vitrih, B. Jüttler, K. Birner, Isogeometric analysis with geometrically continuous functions on two-patch geometries, *Comput. Math. Appl.* 70 (2015) 1518–1538.
- [35] A. Collin, G. Sangalli, T. Takacs, Analysis-suitable G^1 multi-patch parametrizations for C^1 isogeometric spaces, *Comput. Aided Geom. Design* 47 (2016) 93–113.
- [36] M. Kapl, V. Vitrih, Space of C^2 -smooth geometrically continuous isogeometric functions on two-patch geometries, *Comput. Math. Appl.* 73 (2017) 37–59.
- [37] M. Kapl, G. Sangalli, T. Takacs, An isogeometric C^1 subspace on unstructured multi-patch planar domains, *Comput. Aided Geom. Design* 69 (2019) 55–75.
- [38] J. Grošelj, M. Kapl, M. Knez, T. Takacs, V. Vitrih, A super-smooth C^1 spline space over planar mixed triangle and quadrilateral meshes, *Comput. Math. Appl.* 80 (12) (2020) 2623–2643, <http://dx.doi.org/10.1016/j.camwa.2020.10.004>, URL <http://www.sciencedirect.com/science/article/pii/S0898122120303977>.
- [39] L. Schumaker, *Spline Functions: Basic Theory*, third ed., Cambridge University Press, 2007.
- [40] C. De Boor, C. Deboor, K. Höllig, S. Riemenschneider, *Box splines*, vol. 98, Springer Science & Business Media, 1993.
- [41] T. Kanduč, C. Giannelli, F. Pelosi, H. Speleers, Adaptive isogeometric analysis with hierarchical box splines, *Comput. Methods Appl. Mech. Engrg.* 316 (2017) 817–838.
- [42] C. Giannelli, T. Kanduč, F. Pelosi, H. Speleers, An immersed-isogeometric model: Application to linear elasticity and implementation with thbox-splines, *J. Comput. Appl. Math.* 349 (2019) 410–423.
- [43] H. Speleers, Construction of normalized B-splines for a family of smooth spline spaces over Powell–sabin triangulations, *Constr. Approx.* 37 (1) (2013) 41–72.
- [44] H. Speleers, C. Manni, F. Pelosi, M.L. Sampoli, Isogeometric analysis with Powell–sabin splines for advection–diffusion–reaction problems, *Comput. Methods Appl. Mech. Engrg.* 221–222 (2012) 132–148, <http://dx.doi.org/10.1016/j.cma.2012.02.009>, URL <http://www.sciencedirect.com/science/article/pii/S0045782512000515>.
- [45] J. Grošelj, H. Speleers, Super-smooth cubic Powell–sabin splines on three-directional triangulations: B-spline representation and subdivision, *J. Comput. Appl. Math.* 386 (2021) 113245, <http://dx.doi.org/10.1016/j.cam.2020.113245>, URL <http://www.sciencedirect.com/science/article/pii/S0377042720305367>.
- [46] B. Urlick, B. Marussig, E. Cohen, R.H. Crawford, T.J. Hughes, R.F. Riesenfeld, Watertight boolean operations: A framework for creating CAD-compatible gap-free editable solid models, *Comput. Aided Des.* 115 (2019) 147–160, <http://dx.doi.org/10.1016/j.cad.2019.05.034>.
- [47] B. Urlick, R.H. Crawford, T.J. Hughes, E. Cohen, R.F. Riesenfeld, Reconstruction of gap-free intersections for trimmed NURBS surfaces, in: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 59179, American Society of Mechanical Engineers, 2019, V001T02A026.
- [48] R.R. Hiemstra, K.M. Shepherd, M.J. Johnson, L. Quan, T.J. Hughes, Towards untrimmed NURBS: CAD embedded reparameterization of trimmed B-rep geometry using frame-field guided global parameterization, *Comput. Methods Appl. Mech. Engrg.* 369 (2020) 113227, <http://dx.doi.org/10.1016/j.cma.2020.113227>.
- [49] G.J. Hettinga, J. Kosinka, Conversion of B-rep CAD models into globally G^1 triangular splines, *Comput. Aided Geom. Design* 77 (2020) 101832, <http://dx.doi.org/10.1016/j.cagd.2020.101832>, URL <http://www.sciencedirect.com/science/article/pii/S0167839620300194>.
- [50] Y. Song, E. Cohen, Making trimmed B-spline B-reps watertight with a hybrid representation, in: *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection, 2019.

- [51] Y. Song, E. Cohen, Refinement for a hybrid boundary representation and its hybrid volume completion, *SMAI J. Comput. Math.* 5 (2019) 3–25.
- [52] N. Lei, X. Zheng, J. Jiang, Y.-Y. Lin, D.X. Gu, Quadrilateral and hexahedral mesh generation based on surface foliation theory, *Comput. Methods Appl. Mech. Engrg.* 316 (2017) 758–781.
- [53] M. Campen, D. Zorin, Similarity maps and field-guided T-splines: A perfect couple, *ACM Trans. Graph.* 36 (4) (2017) 1–16.
- [54] R. Schmidt, R. W. Isogeometric analysis of trimmed NURBS geometries, *Comput. Methods Appl. Mech. Engrg.* 241–244 (2012) 93–111, <http://dx.doi.org/10.1016/j.cma.2012.05.021>, URL <http://www.sciencedirect.com/science/article/pii/S0045782512001843>.
- [55] M. Breitenberger, A. Apostolatos, B. Philipp, R. W. Analysis in computer aided design: Nonlinear isogeometric B-rep analysis of shell structures, *Comput. Methods Appl. Mech. Engrg.* 284 (2015) 401–457, <http://dx.doi.org/10.1016/j.cma.2014.09.033>, URL <http://www.sciencedirect.com/science/article/pii/S0045782514003569>, Isogeometric Analysis Special Issue.
- [56] P. Antolin, A. Buffa, M. Martinelli, Isogeometric analysis on V-reps: First results, *Comput. Methods Appl. Mech. Engrg.* 355 (2019) 976–1002, <http://dx.doi.org/10.1016/j.cma.2019.07.015>, URL <http://www.sciencedirect.com/science/article/pii/S0045782519304116>.
- [57] L. Coradello, P. Antolin, R. Vázquez, A. Buffa, Adaptive isogeometric analysis on two-dimensional trimmed domains based on a hierarchical approach, *Comput. Methods Appl. Mech. Engrg.* 364 (2020) 112925, <http://dx.doi.org/10.1016/j.cma.2020.112925>, URL <http://www.sciencedirect.com/science/article/pii/S0045782520301080>.
- [58] L.F. Leidingner, M. Breitenberger, A.M. Bauer, T. Oberbichler, S. Hartmann, R. Wüchner, K.-U. Bletzinger, F. Duddeck, L. Song, Explicit Isogeometric B-Rep Analysis on Trimmed NURBS-Based Multi-Patch CAD Models in LS-DYNA, in: *Proceedings of 12th European LS-DYNA Conference*, 2019.
- [59] Y. Song, E. Cohen, Volume completion for trimmed B-reps, in: *2019 23rd International Conference in Information Visualization–Part II*, IEEE, 2019, pp. 147–155.
- [60] M. Borden, M. Scott, J. Evans, T. Hughes, Isogeometric finite element data structures based on Bézier extraction of NURBS, *Internat. J. Numer. Methods Engrg.* 87 (2011) 15–47.
- [61] M. Scott, M. Borden, C. Verhoosel, T. Sederberg, T. Hughes, Isogeometric finite element data structures based on Bézier extraction of T-splines, *Internat. J. Numer. Methods Engrg.* 88 (2011) 126–156.
- [62] D. Toshniwal, H. Speleers, R. Hiemstra, C. Manni, T. Hughes, Multi-degree B-splines: Algorithmic computation and properties, *Comput. Aided Geom. Design* 76 (2020) 101792.
- [63] H. Speleers, Algorithm 999: Computation of multi-degree B-splines, *ACM Trans. Math. Software* 45 (2019) Article No. 43.
- [64] S. Chan, I. Tuba, W. Wilson, On the finite element method in linear fracture mechanics, *Eng. Fract. Mech.* 2 (1) (1970) 1–17.
- [65] SurfLab: Surface gallery, http://www.cise.ufl.edu/research/SurfLab/shape_gallery.shtml.
- [66] M.-J. Lai, L.L. Schumaker, *Spline Functions on Triangulations*, vol. 110, Cambridge University Press, 2007.
- [67] L. Piegl, W. Tiller, *The NURBS Book*, second ed., Springer-Verlag, 2012.
- [68] S.P. Timoshenko, J. Goodier, *Theory of elasticity*, Int. J. Bulk Solids Storage Silos (2011).
- [69] L. Dedè, A. Quarteroni, Isogeometric analysis for second order partial differential equations on surfaces, *Comput. Methods Appl. Mech. Engrg.* 284 (2015) 807–834.
- [70] D. Benson, Y. Bazilevs, M.-C. Hsu, T. Hughes, Isogeometric shell analysis: The Reissner–mindlin shell, *Comput. Methods Appl. Mech. Engrg.* 199 (5–8) (2010) 276–289.