Isogeometric collocation using analysis-suitable T-splines of arbitrary degree

Hugo Casquero[a,*], Lei Liu[b], Yongjie Zhang[b], Alessandro Reali[c,d], Hector Gomez[a]

[a] *Departamento de Métodos Matemáticos, Universidade da Coruña, Campus de A Coruña, 15071, A Coruña, Spain.*
[b] *Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.*
[c] *Department of Civil Engineering and Architecture, University of Pavia, via Ferrata 3, 27100, Pavia, Italy.*
[d] *Technische Universität München – Institute for Advanced Study, Lichtenbergstraße 2a, 85748, Garching, Germany.*

Abstract

This paper deals with the use of analysis-suitable T-splines of arbitrary degree in combination with isogeometric collocation methods for the solution of second- and fourth-order boundary-value problems. In fact, analysis-suitable T-splines appear to be a particularly efficient locally refinable basis for isogeometric collocation, able to conserve the cost of only one point evaluation per degree of freedom typical of standard NURBS-based isogeometric collocation. Furthermore, T-splines allow to easily create highly non-uniform meshes without introducing elements with high aspect ratios; this makes it possible to avoid the numerical instabilities that may arise in the case of problems characterized by reduced regularity when Neumann boundary conditions are imposed in strong form and elements with high aspect ratio are used. The local refinement properties of T-splines can be also successfully exploited to approximate problems where point loads are applied. Finally, several numerical tests are herein presented in order to confirm all the above-mentioned features, as well as the good overall convergence properties of the combination of isogeometric collocation and analysis-suitable T-splines.

Keywords:
Collocation, Analysis-suitable T-splines, Arbitrary-degree T-splines, Isogeometric analysis, Bézier extraction, Kirchhoff plates

1. Introduction

Isogeometric analysis (IGA) was first introduced ten years ago by [1] with the main aim of bridging Computer Aided Design (CAD) and Finite Element Analysis (FEA). The basic IGA concept, based on the isoparametric paradigm, consisted of adopting the same basis

*Corresponding author.
Email address: hugo.casquero@udc.es (Hugo Casquero)

functions used for geometry representations in CAD systems — such as, e.g., Non-Uniform Rational B-Splines (NURBS) — for the approximation of field variables. The original goal was a cost-saving simplification of the typically expensive mesh generation and refinement processes required by standard FEA. In addition, thanks to the high-regularity properties of its basis functions, IGA showed a better accuracy per-degree-of-freedom and an enhanced robustness with respect to standard FEA [2]. Such a superior behavior was exploited in a number of applications ranging from solids and structures (see, e.g., [3, 4, 5, 6, 7, 8, 9, 10, 11] to fluids (see, e.g., [12, 13, 14, 15, 16] and fluid-structure interaction (see, e.g., [17, 18, 19, 20, 21, 22, 23, 24, 25]). Moreover, the newly available higher regularity opened also the door to geometrically flexible discretizations of higher-order partial differential equations in primal form such as, e.g., phase-field models [26, 27, 28, 29, 30, 31], Kirchhoff-Love shells [32, 33, 34], among others.

A well-known important issue of IGA is related to the development of efficient integration rules able to reduce the high array formation costs induced by standard Gaussian quadrature, in particular when higher-order approximations are employed. *Ad-hoc* quadrature rules were proposed by several authors (cf., among others, [35] and references therein), but the development of a general and effective solution for Galerkin-based IGA methods is still an open problem. In an attempt to address this issue taking full advantage of the special possibilities offered by IGA and in particular by the available higher regularity, isogeometric collocation (IGA-C) schemes were proposed in [36, 37]. The main idea consisted of the discretization of the governing partial differential equations in strong form, within the isoparametric paradigm, reducing the number of evaluations needed for array formation to only one per degree of freedom. The aim was to optimize the computational cost still relying on IGA geometrical flexibility and accuracy[1]. Detailed comparisons with both IGA and FEA Galerkin-based approaches were carried out in [38], showing IGA-C advantages in terms of accuracy versus computational cost, in particular for higher-order approximation degrees. Since its introduction, many promising significant works on IGA-C were published in different fields, including phase-field modeling [39], contact [40, 41], nonlinear elasticity [41], as well as several interesting studies in the context of structural elements [42, 43, 44, 45, 46]. A version of IGA-C based on generalized B-splines was recently proposed in [47].

It is important to remark that a fundamental ingredient towards the full success of any computational method is represented by its capabilities in terms of local refinement, and, clearly, IGA-C cannot be an exception. Accordingly, a locally refinable version of IGA-C, based on hierarchical splines, was introduced in [38]. The results proposed in that work are surely promising and convincing, but it has also to be noted that, to avoid linear dependence or instability problems, the concept of "weighted IGA-C" needed to be introduced, leading to a higher number of evaluations per degree of freedom in the transition regions between refinement levels.

[1] In general, IGA collocation features look particularly attractive when evaluation and formation costs are dominant, as in the case, e.g., of explicit structural dynamics [37].

The main goal of the present work is to introduce a new locally refinable version of IGA-C based on analysis-suitable T-splines (ASTS). In fact, ASTS-based IGA probably represents so far the most promising and studied locally refinable version of isogeometric analysis[2] and, as a consequence, exploring the potential guaranteed by the combination of IGA-C and T-splines looks like a natural way to go. In addition, the use of ASTS within IGA-C avoids the need of resorting to the above-mentioned weighted collocation concept needed in the case of hierarchical splines, thus conserving the cost of only one evaluation per degree of freedom typical of NURBS-based IGA-C.

We also highlight that one of the drawbacks of NURBS-based IGA-C, recently observed in [40], is the fact that spurious oscillations in the numerical solution may arise in the case of problems characterized by reduced regularity, when Neumann boundary conditions are imposed in strong form and elements with high aspect ratio are used. The adoption of T-splines may be of help also in these situations, allowing to easily create highly non-uniform meshes without introducing elements with high aspect ratios, and thus avoiding the appearance of spurious instabilities.

Finally, another problem that, to our knowledge, has still to be tackled in the context of IGA-C is the application of point loads, which have to be modeled as approximate Dirac deltas. To be properly solved, such a problem clearly requires a strong local refinement around the load point of application, and this could be easily guaranteed by the use of ASTS.

For all the above reasons, the adoption of analysis-suitable T-splines of arbitrary degree within isogeometric collocation methods looks particularly attractive, and this work aims at revealing its potential for the solution of second-order and fourth-order boundary-value problems.

The paper is organized as follows. In the second section, a detailed overview on T-splines and their analysis suitable version is given, while the third section is devoted to the description of isogeometric collocation methods for second-order and fourth-order problems. In the fourth section several numerical tests are presented to show the performance of the proposed combination of isogeometric collocation and analysis-suitable T-splines; these include four second-order problems characterized by different geometries and solutions that can be regular, singular, or presenting sharp layers or spikes, as well as a fourth-order problem consisting of a simply supported circular Kirchhoff plate loaded by a point force. Finally, in the last section, conclusions are drawn.

2. T-spline overview

In this Section, we present a short overview of the T-spline technology. T-splines were originated in the design field [48, 49, 50] as a generalization of NURBS in order to circumvent some of the existent limitations of NURBS surfaces, namely, the lack of local refinement and the difficulty to create watertight surfaces in general cases. The amount of

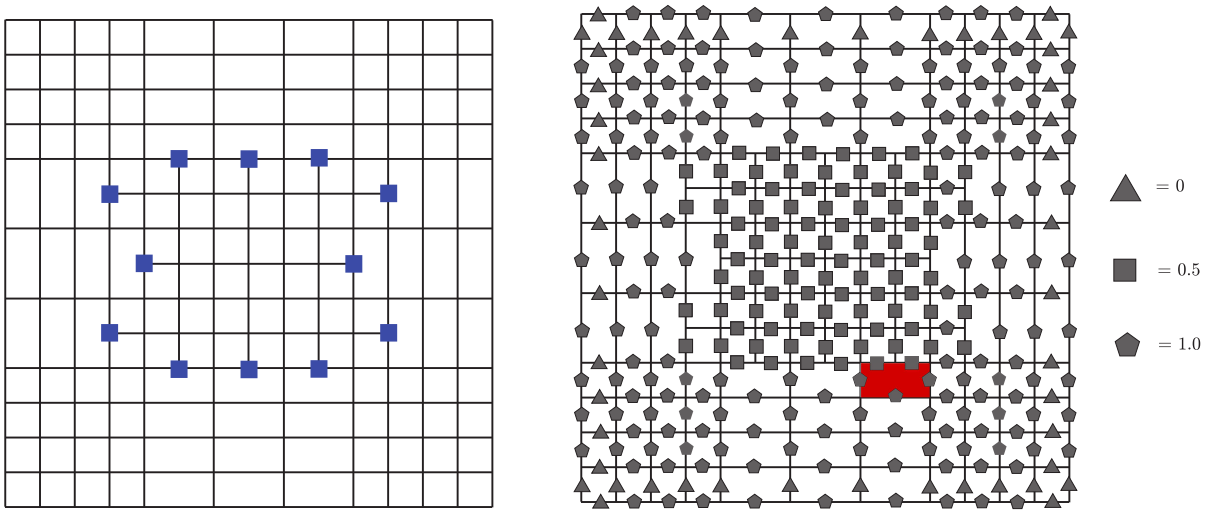
[2]See the following Sect. 2 for an overview on T-splines, including the basic references on the topic.

degrees of freedom needed in order to perform finite element analysis in an accurate way is typically much higher than the number of degrees of freedom that we need to represent exactly a certain geometric design. As a consequence, the geometrical model coming from a CAD program is generally refined previously to be used in isogeometric analysis, which makes even more important the ability to perform local refinement. Pursuing this aim, T-splines were recently brought to analysis [51]. Relevant advances have been carried out since then: mathematical background of T-splines has been studied [52, 53, 54, 55], several procedures to create volumetric T-splines have been designed [56, 57, 58, 59, 60], the conversion of a trimmed NURBS surface to an untrimmed T-spline surface was tackled in [61], the conversion of an unstructured quadrilateral mesh to a standard T-spline surface was explained in [62], and ASTS-based isogeometric analysis was applied to several problems in computational mechanics [3, 63, 64, 65, 66, 67, 68, 69, 70]. In the design realm, cubic T-splines are prominent, but arbitrary-degree T-splines have recently received attention [71]. Naturally, the need for flexibility in the order of the approximation becomes more pressing when design is to be integrated with analysis. Here, we describe analysis-suitable T-splines of arbitrary degree in two dimensions (relying on their existing mathematical background [53, 54]) and apply them to isogeometric collocation. The main difficulty stems from the fact that T-splines of even and odd degree are generated in a slightly different way. In our presentation, we assume that the reader is familiarized with NURBS-based isogeometric analysis [2] and the concept of Bézier extraction [72, 73]. Under that assumption, this Section is practically self-contained. For simplicity, we use the same degree in both directions, but our algorithms can be easily extended to the general situation following the ideas presented in [53]. Throughout the paper, the degree of the functions will be denoted by p.

2.1. T-mesh in index space and knot interval configuration

The most fundamental object of the T-spline technology is the so-called *T-mesh in index space*. Although there are slightly different definitions of the T-mesh in index space in the literature, here we follow that given in [51]. When no confusion arises, we will simply use the term T-mesh *in lieu* of T-mesh in index space. Essentially, the T-mesh in index space is a polygonal tiling of a two-dimensional region, which encodes the topological information necessary to generate T-splines. Fig. 1(a) shows a sample T-mesh in index space. The polygons of the T-mesh are typically referred to as *faces*[3]. The corners of the faces are called *nodes* or *vertices*, and a line joining (exactly) two vertices is an *edge*. An essential difference of the T-mesh in index space with respect to its NURBS analogue is the presence of *T-junctions*, which are similar to hanging nodes in finite element analysis [see the blue squares in Fig. 1(a)]. Note that we exclude *ab initio* the presence of extraordinary nodes [74, 64, 75] in the T-mesh. To our knowledge, defining splines with $\mathcal{C}^1$ or higher continuity in the propinquity of an extraordinary node is still an open problem, which constitutes a

[3]Occasionally, the word element is used in the literature as a synonym of face, but we will reserve it for something else.



(a) T-mesh in index space

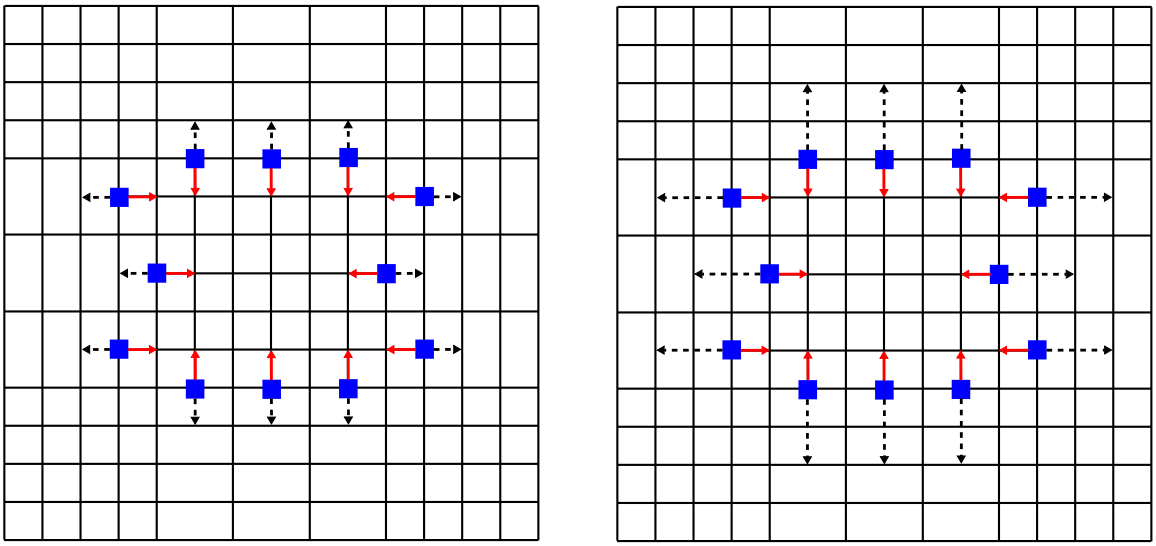
(b) Knot interval configuration

Figure 1: (Color online) (a) T-mesh in index space with twelve T-junctions marked with blue squares. (b) Valid knot interval configuration for the T-mesh represented in Fig. 1(a). The pentagons, squares and triangles correspond to knot intervals of 1, 1/2, and 0, respectively. This knot interval configuration is valid for $p = 2$ and $p = 3$, which are the cases used in this Section to present the T-spline technology.

significant obstacle for the application of our collocation methods[4].

Once the concept of T-mesh has been introduced, our goal is to define spline functions over the T-mesh. These functions are called *T-spline blending functions*, and will be eventually used for analysis. To construct T-spline blending functions, we need to define a valid *knot interval configuration*. A knot interval configuration is defined by assigning a non-negative real number to each T-mesh edge. These non-negative real numbers are called *knot intervals* and may be interpreted as the difference of two adjacent knots [49]. We may also think of knot intervals as the length of the edges in a parametric space. We employ knot interval configurations which satisfy two conditions: (1) the knot intervals assigned to opposite sides of a given T-mesh face must sum to the same value; (2) the T-mesh has $\text{int}(p/2)$ outer rings of zero-length knot intervals, where $\text{int}(\cdot)$ represents the integer part of a real number. Fig. 1(b) shows a valid knot configuration of the T-mesh depicted in Fig. 1(a). Condition (1) is illustrated by the red face in Fig. 1(b). In particular, in the horizontal edges, we have one knot interval of value 1 in the bottom side and two knot intervals of value 1/2 in the top side. The fulfillment of this condition guarantees that if the T-mesh is refined removing all T-junctions, there will be an underlying parametric space which is well defined. Condition (2) is convenient because it permits to infer easily the number of T-spline blending functions associated to a T-mesh. In particular, given

[4]In [64], we consider a hybrid variational-collocation method for fluid-structure interaction which may be used on T-meshes with extraordinary nodes. The reason for this is that the algorithm is designed in a way that no derivatives need to be computed at the extraordinary nodes.



(a) $p = 2$ (b) $p = 3$

Figure 2: (Color online) Extended T-meshes associated to the T-mesh represented in Fig. 1(a) for $p = 2$ and $p = 3$. Face extensions are represented by black dashed lines and edge extensions are marked with solid red lines. Note that the red arrows hide the T-mesh edges located underneath them in the plot.

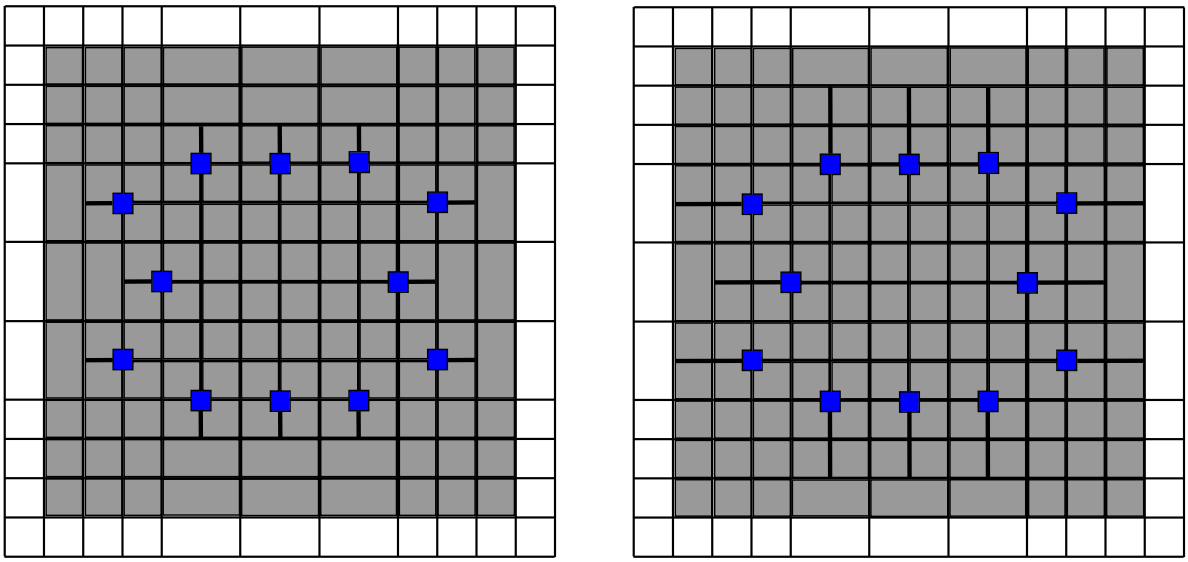
a T-mesh, the global number of blending functions associated to it equals the number of vertices for odd-degree splines and the number of faces for even-degree functions.

Remark:

We would like to mention that some authors define the T-mesh in index space and the knot interval configuration by setting p zero-length outer rings [53]. Although this definition has certainly some advantages, here we use the definition given in [51] to maintain the correspondence between the number of functions and the number of vertices (odd degree) or faces (even degree).

2.2. Extended T-mesh and elemental T-mesh

To define the extended T-mesh and the elemental T-mesh, we need to introduce first the concept of *T-junction extension*. A T-junction extension is composed of a face extension and an edge extension. A *face extension* is a directed line segment obtained as follows: we start at the T-junction and move in the direction of the missing edge until $\text{int}((p+1)/2)$ orthogonal edges are encountered. Face extensions are marked with black dashed arrows in Fig. 2 for $p = 2$ [panel (a)] and $p = 3$ [panel (b)]. We define an *edge extension* as a directed line segment obtained by moving in the opposite direction of the face extension until $\text{int}(p/2)$ orthogonal edges are encountered. Edge extensions are highlighted with red solid arrows in Fig. 2. When the T-junction extensions are plotted on top of the T-mesh, we get the *extended T-mesh*. The extended T-meshes for $p = 2$ and $p = 3$ corresponding to the T-mesh of Fig. 1(a) are shown in Fig. 2.



(a) $p = 2$ (b) $p = 3$

Figure 3: (Color online) Elemental T-meshes associated to the T-mesh represented in Fig. 1(a) for $p = 2$ and $p = 3$. The rectangles which are not shaded are not part of the elemental T-mesh. By comparing these plots with the extended T-meshes, it may be observed how faces of the T-mesh in index space have been split by face extensions giving rise to two elements in the elemental T-mesh.

The *elemental T-mesh* is a new polygonal tiling that may be obtained from the extended T-mesh and the knot interval configuration. The polygons that form the elemental T-mesh will be called *elements*. T-mesh faces of zero area (i.e., zero measure) do not contribute elements to the elemental T-mesh and all T-mesh faces of non-zero area give rise to at least one element in the elemental T-mesh. Those which are not crossed by a T-junction extension give rise to exactly one element, and those crossed by T-junction extensions may give rise to more than one. Fig. 3 shows as shaded areas delimited by black lines the polygons that compose the elemental T-mesh. Note, in particular, how some T-mesh faces have been split into two elements in the elemental T-mesh. Notice also how the zero-surface faces do not form elements (see the outer ring). The elements of the elemental T-mesh are important objects for analysis because they delimit areas in which all T-spline blending functions are $\mathcal{C}^\infty$. The elements of the elemental T-mesh will be pushed forward to physical space using the isoparametric concept. Each element of the elemental T-mesh gives rise to a so-called Bézier element [73]. Bézier elements are suitable regions to perform numerical integration in an isogeometric analysis code based on variational forms.

2.3. Anchors, local knot interval vectors and local blending function coordinates

An *anchor* is simply a point of the index space to which we associate a T-spline blending function. The anchor associated to the A-th T-spline blending function will be denoted by $\mathbf{s}_A$. Note that here, and in what follows, indices denoted by capital letters take values of the global function numeration. Given a T-mesh in index space, the anchors are located at the vertices if p is odd and at the centers of the faces if p is even. Using the T-mesh

in Fig. 1(a) as our starting point, Fig. 4 shows the anchors associated to three particular blending functions with global indices A, B, and C, for even- (left) and odd-degree (right) splines. The location of the anchors is marked with a star, a square and a pentagon in the plot. Once we know the location of the anchor for a given function, we need to define a *local knot interval vector* in each direction of the index space. Let us denote by $\Delta_Q\widehat{\Xi}^i$ the local knot interval vector of the Q-th function (or anchor) in the direction i, where $i = 1$ represents the horizontal direction and $i = 2$ the vertical one. The vector $\Delta_Q\widehat{\Xi}^i$ has length $p + 1$ for all Q and for all i, and may be written in component notation as $\Delta_Q\widehat{\Xi}^i = \{\Delta_Q\widehat{\xi}^i_1, \Delta_Q\widehat{\xi}^i_2, ..., \Delta_Q\widehat{\xi}^i_{p+1}\}$. A given component of the vector $\Delta_Q\widehat{\Xi}^i$ corresponds to the knot interval associated to a particular edge of the T-mesh that follows the direction i. Let us now explain how to compute the local knot interval vectors of a given anchor from the knot interval configuration. Put simply, the problem boils down to selecting a set of edges associated to the anchor and fill $\Delta_Q\widehat{\Xi}^i$ with their knot intervals. To identify those edges, we proceed as follows: Let us illustrate the process for $i = 1$, that is, let us define the vector $\Delta_Q\widehat{\Xi}^1$, which is associated to the horizontal direction. The procedure is identical for the vertical direction. The first step of the process is to place a horizontal segment centered in the anchor with sufficient length to cross exactly $p+2$ orthogonal edges. Note that "centered in the anchor" means that the segment crosses the same number of orthogonal edges on the left- and right-hand sides of the anchor. This horizontal segment is represented in Fig. 4 by a thick, semi-transparent red line (for the time being, let us restrict our attention to the anchors $\mathbf{s}_A$ and $\mathbf{s}_B$). The fact that the segment crosses $p + 2$ orthogonal edges implies that it spans the length of a particular set of $p + 1$ edges. The knot intervals associated to those edges taken from left to right are the components of $\Delta_Q\widehat{\Xi}^1$. Note that condition (1) imposed to the knot interval configuration in Sect. 2.1 makes the choice of edges well defined. When an anchor is sufficiently close to the boundary to prevent us from defining a large enough segment so as to cross $p+2$ orthogonal edges, we will add zero-length edges accordingly (see the anchors $\mathbf{s}_C$ in Fig. 4, which include thick black lines indicating the number of extra zero-length edges that need to be added). When we repeat this process in the vertical direction we obtain a cross centered in the anchor (defined by the horizontal and vertical red segments), which defines the support of the function in the obvious way. The supports are marked in Fig. 4 as rectangular colored areas. Using the bottom-left corner of each of these colored areas as origin, we can define the so-called *local blending function coordinates*. Note that there is one of these systems associated to each of the anchors and that they have been marked in Fig. 4 with arrows. Using as reference system the local blending function coordinates associated to a generic anchor $\mathbf{s}_Q$, the support of the Q-th function, denoted $\widehat{\square}_Q$, is given by $\widehat{\square}_Q = [0, \sum_{j=1}^{p+1} \Delta_Q\widehat{\xi}^1_j] \times [0, \sum_{j=1}^{p+1} \Delta_Q\widehat{\xi}^2_j]$

2.4. T-spline blending functions, Bézier extraction, and geometrical mapping

We proceed now to define the T-spline blending function associated to a generic anchor A. This function will be denoted by R_A. With the information given in Sect. 2.3, it is possible to anticipate which basis functions will have support on a given element e. To simplify notation, we introduce a local numbering for the blending functions as it is typically done in the finite element method. Here, we follow [76], and use the array IEN

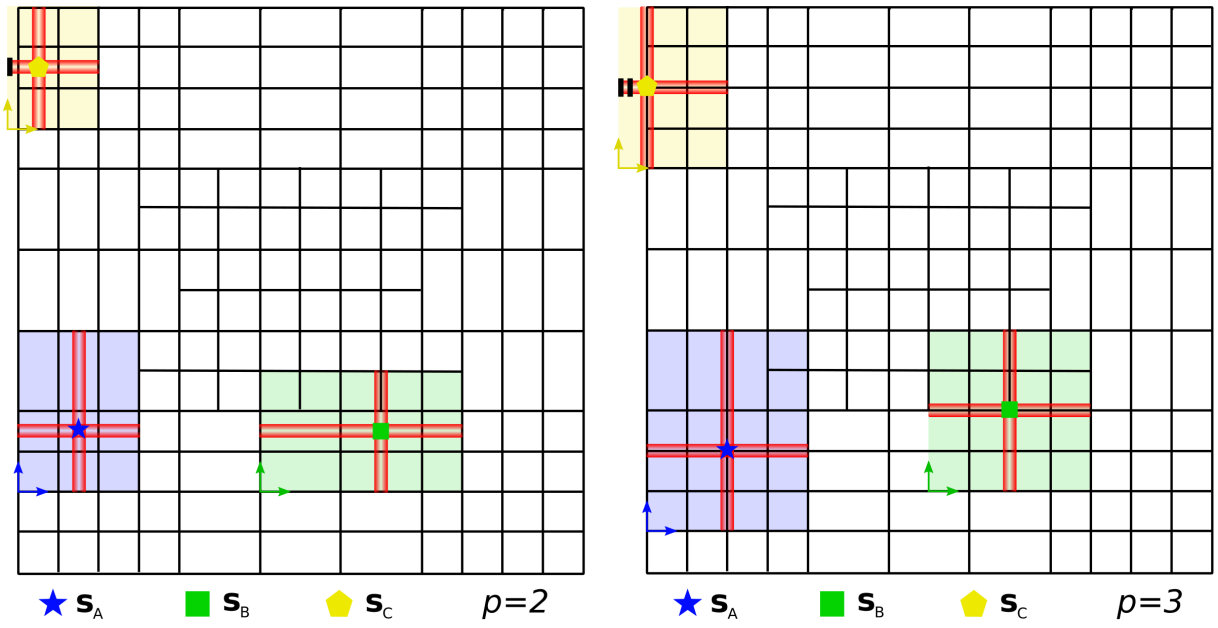


Figure 4: (Color online) Anchors, local knot interval vectors, and local blending function coordinates for $p = 2$ (left) and $p = 3$ (right). The local knot interval vectors associated with the A-th, B-th, and C-th global blending functions for $p = 2$ are $\Delta_A\widehat{\Xi}^1 = \{0, 1, 1\}$, $\Delta_A\widehat{\Xi}^2 = \{1, 1, 1\}$, $\Delta_B\widehat{\Xi}^1 = \{1, 1, 1\}$, $\Delta_B\widehat{\Xi}^2 = \{1, 1, 1/2\}$, $\Delta_C\widehat{\Xi}^1 = \{0, 0, 1\}$, $\Delta_C\widehat{\Xi}^2 = \{1, 1, 0\}$. The local knot interval vectors associated with the A-th, B-th, and C-th global blending functions for $p = 3$ are $\Delta_A\widehat{\Xi}^1 = \{0, 1, 1, 1\}$, $\Delta_A\widehat{\Xi}^2 = \{1, 1, 1, 1\}$, $\Delta_B\widehat{\Xi}^1 = \{1/2, 1/2, 1/2, 1\}$, $\Delta_B\widehat{\Xi}^2 = \{1, 1, 1/2, 1/2\}$, $\Delta_C\widehat{\Xi}^1 = \{0, 0, 0, 1\}$, $\Delta_C\widehat{\Xi}^2 = \{1, 1, 1, 0\}$.

to establish a correspondence between local and global numbering. In particular, we use the formula $A = \mathrm{IEN}(a, e)$, where A is a global blending function index, a is a local-to-element blending function counter, and e denotes the element number. Let us define the vector $\boldsymbol{N}^e = \{N_a^e\}_{a=1}^{n^e}$, where the functions N_a^e are progenitors of the blending functions with support on the element e. Note that, as emphasized with the superscript e in n^e, the length of the vector $\boldsymbol{N}^e$ changes from one element to another. This is a peculiar feature of T-splines which does not hold true for NURBS-based analysis. The concept of Bézier extraction permits computing $\boldsymbol{N}^e$ as a linear combination of the canonical tensor product Bernstein polynomials defined on a fixed parent element, namely, $\square = [-1, +1]^2$. This can be expressed mathematically as

$$\boldsymbol{N}^e(\boldsymbol{\xi}) = \boldsymbol{C}^e\boldsymbol{B}(\boldsymbol{\xi}) \quad \boldsymbol{\xi} \in \square\,, \tag{1}$$

where $\boldsymbol{B} = \{B_i^p\}_{i=1}^{(p+1)^2}$ is a vector containing the two-dimensional Bernstein polynomials of degree p in the domain $\square$ (see [74] for a precise description). The linear operator $\boldsymbol{C}^e$ can be represented by a rectangular matrix and it is called (element) Bézier extraction operator. Although we will not go into the details of the computation of $\boldsymbol{C}^e$, which may be found in [73], we point out that the Bézier extraction operator is computed from the knot interval configuration of the T-mesh using knot insertion.

In practical cases, it may be useful to utilize rational functions, typically to reproduce

exactly a particular geometry. Given a set of global weights $\{w_A\}_{A=1}^{n}$ and their local counterparts for element e, namely, $\{w_a^e\}$, we can rationalize the functions in $\boldsymbol{N}^e$ as

$$R_a^e(\boldsymbol{\xi}) = \frac{w_a^e N_a^e(\boldsymbol{\xi})}{\sum_{b=1}^{n^e} w_b^e N_b^e(\boldsymbol{\xi})} \ , \tag{2}$$

where R_a^e is the a-th rational T-spline blending function over the element e. This finalizes the process of constructing the blending functions in parameter space.

Finally, to perform computations on non-trivial geometries, we need to map the parent element to physical space using a geometrical mapping $\boldsymbol{x}^e : \square \mapsto \Omega^e$, which goes from the parent element to the Bézier element Ω^e. The basic information needed to produce this mapping is a set of geometry control points $\{\boldsymbol{P}_B\}_{B=1}^{n}$ and the set of weights we just introduced, namely, $\{w_B\}_{B=1}^{n}$. A given geometry control point $\boldsymbol{P}_A$ will have local indices associated, such that $\boldsymbol{P}_A = \boldsymbol{P}_a^e$. The same argument applies to weights. Thus, the geometrical mapping local to element e can be defined as

$$\boldsymbol{x}^e(\boldsymbol{\xi}) = \sum_{a=1}^{n^e} \boldsymbol{P}_a^e R_a^e(\boldsymbol{\xi}) \quad \boldsymbol{\xi} \in \square \ . \tag{3}$$

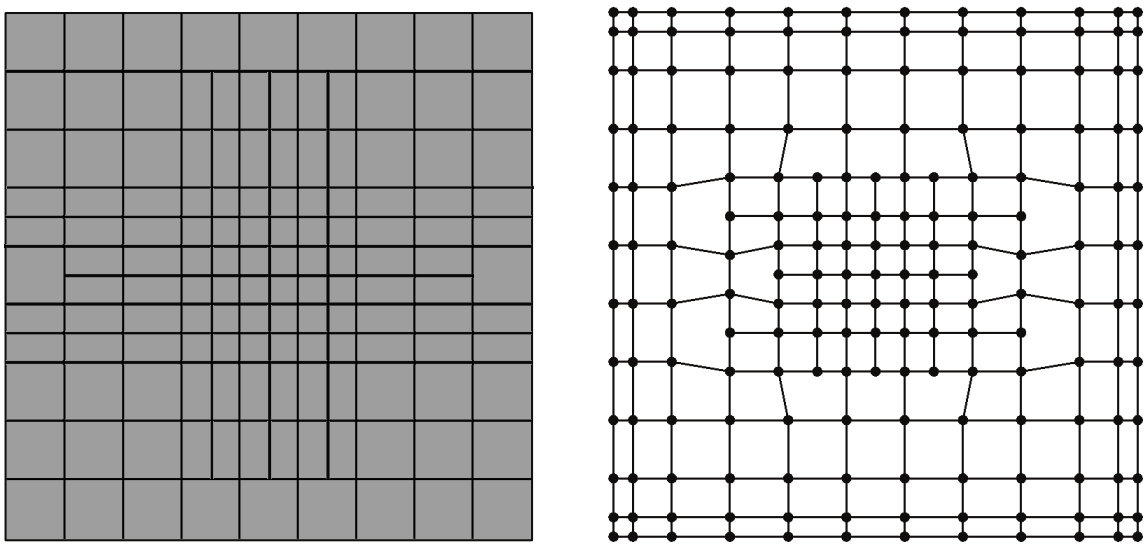
Using the T-spline geometrical mapping given by Eq. (3), we can get the blending functions in physical space. In particular, the restriction of R_A to the element e is just the push forward of R_a^e. Eq. (3) will also be used to map the mesh to physical space. In summary, as shown by Eqs. (1)–(3), the use of Bézier extraction allows to use T-splines in a finite element code modifying just the shape function subroutine.

Remark:

Fig. 5 shows the Bézier mesh and the control mesh associated to the T-mesh in Fig. 1(a) for a particular affine mapping with $p = 3$. The Bézier mesh could also be defined easily for $p = 2$ (or any other degree), but we do not report it here for conciseness. However, for $p = 2$ (even degree, in general) we do not generate the control meshes. For even-degree T-splines, due to the presence of T-junctions, there may be triangles or hexagons in the control mesh. The connectivity of the control points may be obtained from the dual T-mesh. A particular way to represent the control mesh of even degree T-splines is given in [71].

2.5. Analysis-suitable T-splines

If no topological constraints are imposed to the T-mesh, the resulting T-spline blending functions may not satisfy some relevant properties, such as, for example, linear independence or polynomial partition of unity. Analysis-suitable T-splines are a subset of T-splines which emanate from T-meshes satisfying a simple topological restriction, namely, no horizontal T-junction extension intersects a vertical T-junction extension [55, 53]. ASTS retain the local refinement capabilities of T-splines, while satisfying all the important mathematical properties of NURBS. In particular, the blending functions associated to ASTS satisfy *a priori* the following properties:



(a) Bézier mesh, $p = 3$ (b) Control mesh, $p = 3$

Figure 5: (Color online) Bézier mesh and control mesh associated to the T-mesh represented in Fig. 1(a) for a particular affine mapping with $p = 3$.

(1) Partition of unity
(2) Pointwise non-negativity
(3) Linear independence

Property (1) implies that affine transformations of a T-spline object in physical space may be produced by simply applying the transformation to the control points. Properties (1) and (2) guarantee that T-spline objects satisfy the convex-hull property. In [53, 54], the authors also prove that ASTS are dual compatible, which implies optimal approximation properties.

3. Collocation algorithms for second- and fourth-order problems

In this section, we describe our algorithm for isogeometric collocation using ASTS. We focus on boundary-value problems, which involve second- and fourth-order partial-differential operators on mapped geometries. Collocation methods require the evaluation of the strong form of the equation at the so-called collocation points. Therefore, we need to compute the derivatives of the shape functions in physical space up to the maximum order attained in the PDE. Although straightforward, these computations are not standard in computational mechanics, so we include them here for completeness.

3.1. High-order spatial derivatives of rational T-spline blending functions

Our starting point is the elemental geometric mapping introduced in Eq. (3). Assuming this mapping is invertible, we define its inverse function $\boldsymbol{\xi}^e : \Omega^e \mapsto \square$ such that $\boldsymbol{x}^e(\boldsymbol{\xi}^e(\boldsymbol{x})) =$

$\boldsymbol{x}$ for all $\boldsymbol{x} \in \Omega^e$. Taking the derivative of this expression, we obtain the relation

$$\text{(repeated indices sum)} \quad x^e_{i,\alpha}\, \xi^e_{\alpha,j} = \delta_{ij}, \tag{4}$$

where x^e_i and ξ^e_α are, respectively, the i-th component of $\boldsymbol{x}^e$, and the α-th component of $\boldsymbol{\xi}^e$. Latin and Greek indices denote spatial and parametric coordinates, respectively. An index after a comma denotes partial differentiation, for example, $x^e_{i,\alpha} = \partial x^e_i / \partial \xi_\alpha$. Finally, δ_{ij} is the Kronecker's delta. To obtain the derivatives of the T-spline blending functions in physical space, we start with the following identity

$$R_A\left(\boldsymbol{x}^e\left(\boldsymbol{\xi}\right)\right) = R^e_a\left(\boldsymbol{\xi}\right) \quad \boldsymbol{\xi} \in \square, \tag{5}$$

Eq. (5) simply expresses the correspondence between basis functions in physical and parametric space. If we compute the derivative of Eq. (5), it follows that

$$\text{(repeated indices sum)} \quad R_{A,i}\, x^e_{i,\alpha} = R^e_{a,\alpha} \tag{6}$$

Multiplying this equation with $\xi^e_{\alpha,j}$ and using Eq. (4), we can easily obtain $R_{A,i}$ as

$$\text{(repeated indices sum)} \quad R_{A,i} = R^e_{a,\alpha}\, \xi^e_{\alpha,i}. \tag{7}$$

The higher-order spatial derivatives are obtained analogously, that is, deriving Eq. (5) several times by way of the chain rule, and solving for the spatial derivatives using Eq. (4). Proceeding this way, we obtain the second-, the third-, and the fourth-order derivatives, which take on the expressions

$$\text{(repeated indices sum)} \quad R_{A,ij} = [R^e_{a,\alpha\beta} - R_{A,m}\, x^e_{m,\alpha\beta}]\xi^e_{\alpha,i}\, \xi^e_{\beta,j}, \tag{8}$$

$$\begin{aligned}\text{(repeated indices sum)} \quad R_{A,ijk} = [&R^e_{a,\alpha\beta\gamma} - R_{A,mn}(x^e_{n,\beta\gamma}\, x^e_{m,\alpha} + x^e_{m,\alpha\gamma}\, x^e_{n,\beta} + x^e_{m,\alpha\beta}\, x^e_{n,\gamma}) \\ &- R_{A,m}\, x^e_{m,\alpha\beta\gamma}]\, \xi^e_{\alpha,i}\, \xi^e_{\beta,j}\, \xi^e_{\gamma,k},\end{aligned} \tag{9}$$

$$\begin{aligned}\text{(repeated indices sum)} \quad R_{A,ijkl} = \; &[R^e_{a,\alpha\beta\gamma\delta} - R_{A,mnq}(x^e_{q,\gamma\delta}\, x^e_{n,\beta}\, x^e_{m,\alpha} + x^e_{n,\beta\delta}\, x^e_{q,\gamma}\, x^e_{m,\alpha} \\ &+ x^e_{m,\alpha\delta}\, x^e_{q,\gamma}\, x^e_{n,\beta} + x^e_{q,\delta}(x^e_{n,\beta\gamma}\, x^e_{m,\alpha} + x^e_{m,\alpha\gamma}\, x^e_{n,\beta} + x^e_{m,\alpha\beta}\, x^e_{n,\gamma})) \\ &- R_{A,mn}(x^e_{n,\beta\gamma\delta}\, x^e_{m,\alpha} + x^e_{n,\beta\gamma}\, x^e_{m,\alpha\delta} + x^e_{n,\beta\delta}\, x^e_{m,\alpha\gamma} + x^e_{m,\alpha\gamma\delta}\, x^e_{n,\beta} \\ &+ x^e_{n,\gamma\delta}\, x^e_{m,\alpha\beta} + x^e_{m,\alpha\beta\delta}\, x^e_{n,\gamma} + x^e_{m,\alpha\beta\gamma}\, x^e_{n,\delta}) \\ &- R_{A,m}\, x^e_{m,\alpha\beta\gamma\delta}]\xi^e_{\alpha,i}\, \xi^e_{\beta,j}\, \xi^e_{\gamma,k}\, \xi^e_{\delta,l}.\end{aligned} \tag{10}$$

The above formulas give the spatial derivatives of R_A in terms of the parametric derivatives of its local counterpart R^e_a. However, the functions R^e_a are rational splines defined as

$$R^e_a\left(\boldsymbol{\xi}\right) = \frac{w^e_a N^e_a\left(\boldsymbol{\xi}\right)}{w^e(\boldsymbol{\xi})}\ , \quad \text{where} \quad w^e(\boldsymbol{\xi}) = \sum_{b=1}^{n^e} w^e_b N^e_b(\boldsymbol{\xi}) \tag{11}$$

The function N_a^e is the lowest-level function in an isogeometric code, so let us compute the derivatives of R_a^e in terms of those of N_a^e. Using basic manipulations, we obtain

$$R^e_{a,\alpha} = \frac{w^e_a\, N^e_{a,\alpha}}{w^e} - \frac{w^e_a\, N^e_a\, w^e_{,\alpha}}{(w^e)^2}. \tag{12}$$

which may be rewritten as

$$R^e_{a,\alpha} = \frac{w^e_a\, N^e_{a,\alpha} - R^e_a\, w^e_{,\alpha}}{w^e}. \tag{13}$$

Proceeding analogously, we obtain

$$R^e_{a,\alpha\beta} = \frac{w^e_a\, N^e_{a,\alpha\beta} - R^e_a\, w^e_{,\alpha\beta} - R^e_{a,\beta}\, w^e_{,\alpha} - R^e_{a,\alpha}\, w^e_{,\beta}}{w^e}, \tag{14}$$

$$\begin{aligned} R^e_{a,\alpha\beta\gamma} = {} & \frac{w^e_a\, N^e_{a,\alpha\beta\gamma} - R^e_a\, w^e_{,\alpha\beta\gamma} - R^e_{a,\alpha}\, w^e_{,\beta\gamma} - R^e_{a,\beta}\, w^e_{,\alpha\gamma}}{w^e} \\ & - \frac{R^e_{a,\gamma}\, w^e_{,\alpha\beta} + R^e_{a,\beta\gamma}\, w^e_{,\alpha} + R^e_{a,\alpha\gamma}\, w^e_{,\beta} + R^e_{a,\alpha\beta}\, w^e_{,\gamma}}{w^e}, \end{aligned} \tag{15}$$

$$\begin{aligned} R^e_{a,\alpha\beta\gamma\delta} = {} & \frac{w^e_a\, N^e_{a,\alpha\beta\gamma\delta} - R^e_{a,\delta}\, w^e_{,\alpha\beta\gamma} - R^e_a\, w^e_{,\alpha\beta\gamma\delta} - R^e_{a,\alpha\delta}\, w^e_{,\beta\gamma} - R^e_{a,\alpha}\, w^e_{,\beta\gamma\delta}}{w^e} \\ & - \frac{R^e_{a,\beta\delta}\, w^e_{,\alpha\gamma} + R^e_{a,\beta}\, w^e_{,\alpha\gamma\delta} + R^e_{a,\gamma\delta}\, w^e_{,\alpha\beta} + R^e_{a,\gamma}\, w^e_{,\alpha\beta\delta} + R^e_{a,\beta\gamma\delta}\, w^e_{,\alpha}}{w^e} \\ & - \frac{R^e_{a,\beta\gamma}\, w^e_{,\alpha\delta} + R^e_{a,\alpha\gamma\delta}\, w^e_{,\beta} + R^e_{a,\alpha\gamma}\, w^e_{,\beta\delta} + R^e_{a,\alpha\beta\delta}\, w^e_{,\gamma} + R^e_{a,\alpha\beta}\, w^e_{,\gamma\delta}}{w^e} \\ & - \frac{R^e_{a,\alpha\beta\gamma}\, w^e_{,\delta}}{w^e}. \end{aligned} \tag{16}$$

The above formulas completely describe the computations needed to obtain the derivatives of the basis functions in physical space up to order four.

3.2. Collocation method for second-order problems

We illustrate our collocation method for second-order PDEs using a classical example, namely, Poisson equation. Let us assume that Ω is an open set, representing the problem domain. The boundary of Ω is denoted Γ. We call $\boldsymbol{n}$ the unit outward normal to Γ. We assume that Γ admits the decomposition $\Gamma = \overline{\Gamma_D \cup \Gamma_N}$ with $\overline{\Gamma_D \cap \Gamma_N} = \emptyset$, where $\emptyset$ denotes the empty set. To guarantee the well-posedness of the problem, we impose the constraint $\Gamma_D \neq \emptyset$, while Γ_N is allowed to be the empty set. The problem can be formulated as: given

$f : \Omega \mapsto \mathbb{R}$, $u_\Gamma : \Gamma_D \mapsto \mathbb{R}$ and $h_\Gamma : \Gamma_N \mapsto \mathbb{R}$, find $u : \Omega \mapsto \mathbb{R}$ such that,

$$\Delta u = f \qquad \text{in} \quad \Omega, \tag{17}$$

$$u = u_\Gamma \qquad \text{on} \quad \Gamma_D, \tag{18}$$

$$\nabla u \cdot \boldsymbol{n} = h_\Gamma \qquad \text{on} \quad \Gamma_N. \tag{19}$$

Eqns. (18) and (19) represent the Dirichlet and Neumann boundary conditions, respectively. We assume that Ω can be represented by a T-spline geometrical object, which is mapped from a rectangular parameter space. To accommodate in our formulation typical numerical benchmarks (e.g., solutions on squares) and to fix ideas, we assume that the corners of the parameter space are mapped to points with discontinuous normal in physical space. We also allow for the data on the boundary to be discontinuous across the corners. Since the corner points will be collocation points for the boundary conditions in our algorithm for second-order PDEs, we need a procedure to define the normal and the boundary data at those points. Following the standard convention and the theory developed in [37], at those points we just take the arithmetic average of the data computed from the two intersecting boundary edges. In what follows, we will use the notation $\boldsymbol{n}^\star$ to refer to a modified normal which takes the value $\boldsymbol{n}$ everywhere except at the corners, where it has been redefined using the averaging procedure. Similarly, we use the notation $u_\Gamma^\star$ and $h_\Gamma^\star$.

Using the same geometrical mapping that we utilized to construct Ω, we can define a functional space in physical coordinates to approximate the solution to our problem. Let us call that space $\mathcal{V}^h = \text{span}\{R_A\}_{A=1}^n$. The R_A's are guaranteed to be linearly independent due to the use of ASTS, so $\dim(\mathcal{V}^h) = n$. Then, we can postulate the following approximation to the solution

$$u^h(\boldsymbol{x}) = \sum_{A=1}^{n} u_A R_A(\boldsymbol{x}), \tag{20}$$

which reduces our problem to finding the value of the u_A's for $A = 1, \ldots, n$. The strategy to compute the u_A's boils down to imposing that u^h satisfies the PDE or the boundary conditions at a total of n collocation points in physical space, which gives rise to a square system of equations. Note that we need u^h to be at least $\mathcal{C}^2$-continuous in the neighborhood of collocation points. The simplest way to satisfy this requirement is to take $p > 2$ in the algorithm and define a knot interval configuration that produces functions of maximum continuity. Let us call $\mathcal{M}^h = \{\boldsymbol{\tau}_B\}_{B=1}^n$ the set of collocation points in physical space. Then, the problem can be stated as: find $u^h(\boldsymbol{x}) = \sum_{A=1}^n u_A R_A(\boldsymbol{x})$ such that

$$\Delta u^h(\boldsymbol{\tau}_B) = f(\boldsymbol{\tau}_B) \qquad \forall \boldsymbol{\tau}_B \in \Omega, \tag{21}$$

$$u^h(\boldsymbol{\tau}_B) = u_\Gamma^\star(\boldsymbol{\tau}_B) \qquad \forall \boldsymbol{\tau}_B \in \Gamma_D, \tag{22}$$

$$\nabla u^h(\boldsymbol{\tau}_B) \cdot \boldsymbol{n}^\star = h_\Gamma^\star(\boldsymbol{\tau}_B) \qquad \forall \boldsymbol{\tau}_B \in \Gamma_N. \tag{23}$$

To complete the description of our algorithm we just need to define the location of the collocation points, which controls the accuracy and stability of the method. In this article, we use Greville points as our collocation sites. Each T-spline basis function has a Greville point associated that can be obtained from its local knot interval vector. To define

the Greville point associated to the A-th function, we initially define the location of the point using the local basis function coordinates defined in Sect. 2.3, and we use the usual definition of Greville points, which for a spline of degree p is just the average of p knot values (see [36]). Due to our choice to work with knot intervals, rather than knot values, which is customary in the T-spline community, we report here the formula that expresses the location of the Greville point in terms of the knot intervals, namely,

$$\widehat{\boldsymbol{\tau}}_A = \frac{1}{p}\sum_{k=1}^{p}(p+1-k)\begin{Bmatrix}\Delta_A\widehat{\xi}_k^1\\ \Delta_A\widehat{\xi}_k^2\end{Bmatrix} \tag{24}$$

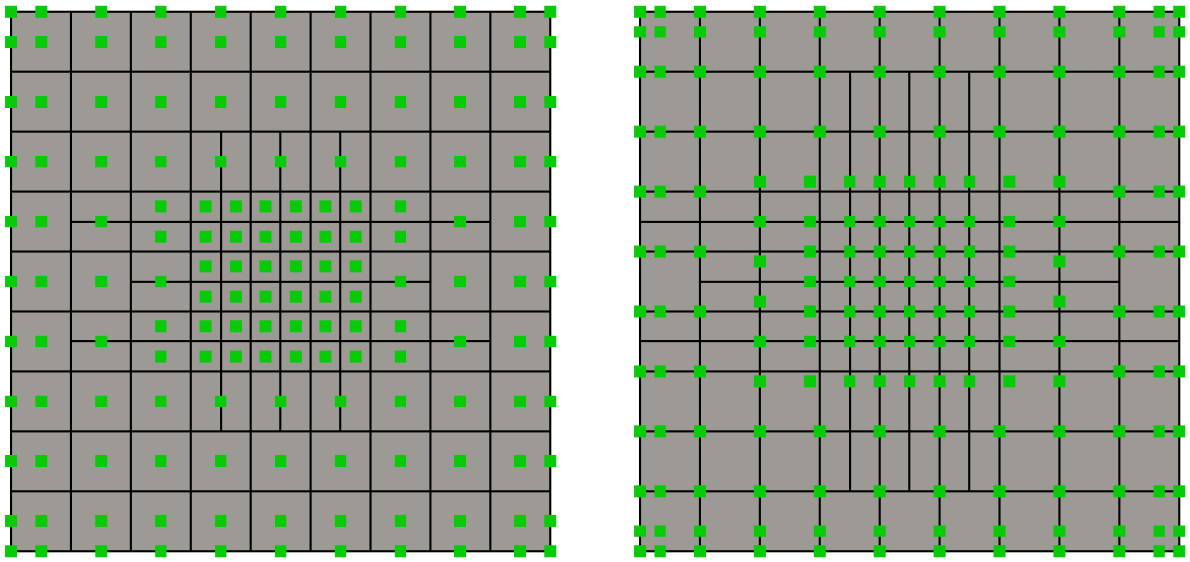
Once we know the location of the A-th Greville point in the local basis function domain, we need to find its coordinates in the parent element. The latter will be denoted by $\widetilde{\boldsymbol{\tau}}_A$. This is done in two steps using the classical affine mappings needed for Bézier extraction which are explained in detail in [73]. First, we use the affine mapping that goes from the local basis function coordinates to the domain of its corresponding Bézier element and then we use the mapping that goes from the Bézier element to the parent element. The Greville points in physical space are computed using the element geometrical mapping as $\boldsymbol{\tau}_A = \boldsymbol{x}^e\left(\widetilde{\boldsymbol{\tau}}_A\right)$. Fig. 6 shows the Greville points in physical space for the T-mesh shown Fig. 1(a) using $p = 2$ and $p = 3$ for a particular geometrical mapping. Note that due to our definition of the knot interval configuration and the local knot interval vectors, the T-spline blending functions with support in the vicinity of the boundary behave similarly to NURBS derived from open knot vectors. This naturally leads to some Greville points falling on the boundary. These collocation points are used to impose boundary conditions.

Remark:

In isogeometric collocation methods, Neumann boundary conditions are usually collocated as shown in Eq. (23). Dirichlet boundary conditions, however, are normally built into the space strongly. Here, in order to simplify the algorithm, we decided to collocate Dirichlet boundary conditions as shown in Eq. (22). Both approaches are equivalent when we work with simple Dirichlet boundary conditions, such as for example, constants.

3.3. Collocation method for fourth-order problems

We illustrate our collocation method for fourth-order PDEs using a classical example, namely, the Kirchhoff plate model. As before, Ω and Γ denote the problem domain and boundary, respectively, and $\boldsymbol{n}$ the unit outward normal. We assume that the boundary Γ admits the decompositions $\Gamma = \overline{\Gamma_w \cup \Gamma_Q}$ and $\Gamma = \overline{\Gamma_\varphi \cup \Gamma_M}$ with $\overline{\Gamma_w \cap \Gamma_Q} = \emptyset$ and $\overline{\Gamma_\varphi \cap \Gamma_M} = \emptyset$. To ensure the well-posedness of the problem, we impose the constraint $\Gamma_w \neq \emptyset$. We formulate the following boundary-value problem over the spatial domain Ω as follows: given the functions $g : \Omega \mapsto \mathbb{R}$, $w_\Gamma : \Gamma_w \mapsto \mathbb{R}$, $\varphi_\Gamma : \Gamma_\varphi \mapsto \mathbb{R}$, $M_\Gamma : \Gamma_M \mapsto \mathbb{R}$, and



(a) $p = 2$ (b) $p = 3$

Figure 6: (Color online) Bézier meshes associated to the T-mesh represented in Fig. 1(a) for a particular affine mapping with $p = 2$ and $p = 3$. The green squares denote the locations of Greville points in physical space.

$Q_\Gamma : \Gamma_Q \mapsto \mathbb{R}$, find $w : \Omega \mapsto \mathbb{R}$ such that,

$$
\begin{aligned}
& D\Delta^2 w = g && \text{in} \quad \Omega, && (25)\\
& w = w_\Gamma && \text{on} \quad \Gamma_w, && (26)\\
& -\nabla w \cdot \boldsymbol{n} = \varphi_\Gamma && \text{on} \quad \Gamma_\varphi, && (27)\\
& \nu D\Delta w + (1-\nu) D\boldsymbol{n} \cdot (\nabla\nabla w)\boldsymbol{n} = M_\Gamma && \text{on} \quad \Gamma_M, && (28)\\
& D(\nabla(\Delta w) + (1-\nu)\boldsymbol{\Psi}(w)) \cdot \boldsymbol{n} = Q_\Gamma && \text{on} \quad \Gamma_Q. && (29)
\end{aligned}
$$

Here, w is the deflection of the plate, $D = Et^3/(12(1-\nu^2))$ is the bending stiffness of the plate, ν is the Poisson ratio, E is the Young modulus, t is the thickness of the plate, g is the load per unit surface, w_Γ is the prescribed deflection, φ_Γ is the prescribed rotation, M_Γ is the prescribed bending moment, Q_Γ is the prescribed effective shear which is a combination of the shear forces and the twisting moments, and $\boldsymbol{\Psi}(\cdot)$ is the following third-order operator

$$
\boldsymbol{\Psi}(w) = \left\{ \begin{array}{c} \frac{\partial^3 w}{\partial x \partial^2 y} \\ \frac{\partial^3 w}{\partial y \partial^2 x} \end{array} \right\}. \tag{30}
$$

As we did in Sect. 3.2, we use the Greville points as collocation points. Greville points in physical space are computed exactly in the same way. The collocation algorithm, however, is different. The difference stems from the fact that a well-posed fourth-order boundary value problem requires two boundary conditions at each point of the boundary. Unfortunately, if we imposed two boundary conditions at each Greville point that falls on the boundary

and the PDE at all the interior Greville points, we would end up with more equations than unknowns. To circumvent this problem, a new algorithm was proposed in [42] for NURBS-based isogeometric collocation of Kirchhoff plates. Since such a recipe can be directly applied also in the case of ASTS-based collocation, it is the solution adopted herein. In the following we show the basic idea discussing a simple particular case; interested readers are referred to [42] for further details.

Despite the adopted collocation strategy is sufficiently general to accommodate any combination of boundary conditions represented by Eqs. (26)–(29), we illustrate it focusing on a simplified case where the same set of boundary conditions is applied on the entire boundary. Accordingly, we consider a simply supported plate, that is, $\Gamma_w = \Gamma_M = \Gamma$. Displacement boundary conditions ($w = 0$) can be equivalently strongly enforced in the space or collocated on all boundary collocation points. The problem is to properly collocate the remaining equations, i.e., the bending moment boundary conditions ($\nu D\Delta w + (1 - \nu)D\boldsymbol{n} \cdot (\nabla\nabla w)\boldsymbol{n} = 0$) and the PDE ($D\Delta^2 w = g$), guaranteeing to have a number of equations equal to the number of unknowns. Our strategy is to collocate the PDE at all Greville points skipping the two outermost layers, while the bending moment boundary conditions are collocated only on boundary collocation points, following the additional criteria that corner points are skipped and the equations corresponding to the two closest points to a corner are averaged. In this way, the linear system of equations to be solved is square and our algorithm is completely described. To give a visual example of which points have to be used to collocate the PDE and the bending moment boundary conditions, we report in Fig. 7 a T-mesh in parametric space along with its Greville points. The black square Greville points are used to collocate the PDE, while the bending moment boundary conditions are collocated either on blue star or green pentagon points; the equations relative to green pentagon points adjacent to the same corner are then averaged. Note that red circle Greville points are not used to collocate the PDE or the bending moment boundary conditions.

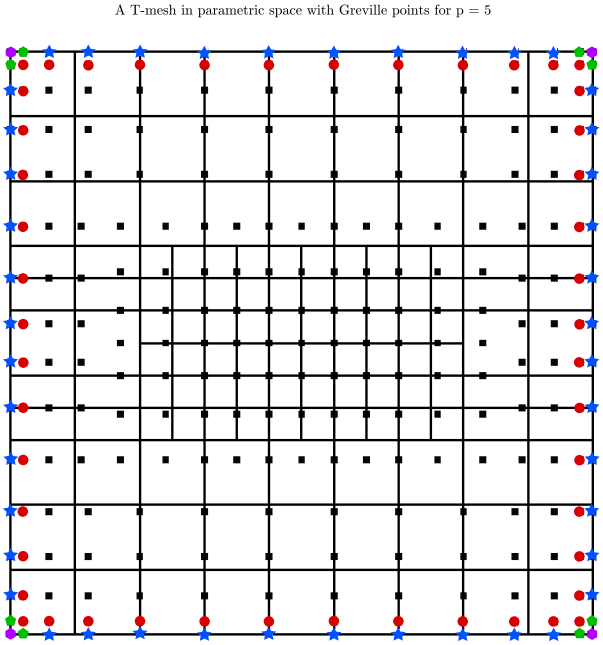


Figure 7: (Color online) T-mesh in parametric space along with its Greville points. The black square Greville points are used to collocate the PDE, while the bending moment boundary conditions are collocated either on blue star or green pentagon points; the equations relative to green pentagon points adjacent to the same corner are then averaged. Note that red circle Greville points are not used to collocate the PDE or the bending moment boundary conditions.

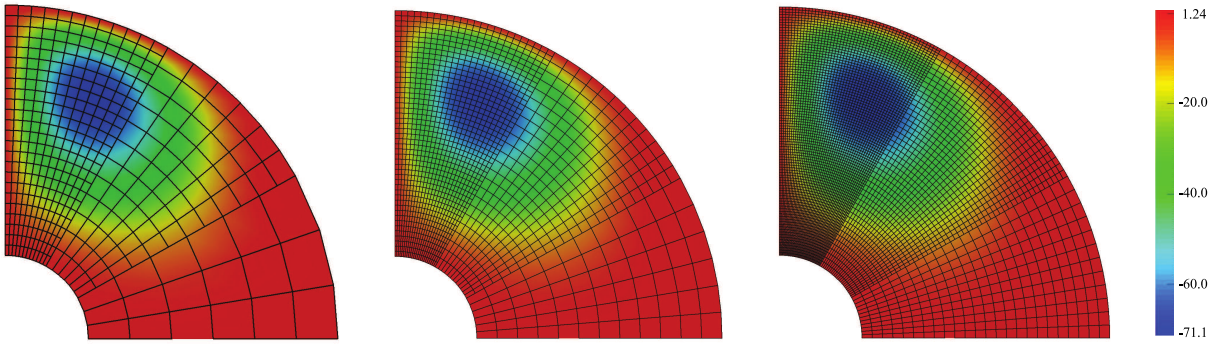


Figure 8: (Color online) Second-order problem with smooth solution. Numerical solution on three different meshes using $p = 4$. The solutions are indistinguishable at the scale of the plot.

4. Numerical results

In this section, we present five numerical examples that illustrate the performance of the proposed collocation methods based on analysis-suitable T-splines of arbitrary degree. First, we consider second-order elliptic problems with smooth and rough solution fields. We use these examples to investigate the numerical convergence rates of our algorithm. Then, we solve a common benchmark problem for the advection-diffusion equation. The solution exhibits boundary and internal layers where we use local refinement. We also consider a reaction-diffusion problem which develops spikes in the solution to illustrate the advantages of local refinement in this case. Finally, we apply our method to a Kirchhoff plate problem. Here, we take advantage of local refinement in order to capture the deformations caused by a point load in a mapped geometry. The code used to perform these simulations has been developed on top of the scientific library PETSc [77, 78].

4.1. Second-order problem with smooth solution

We focus on the reaction-diffusion boundary-value problem

$$-\Delta u + u = f \qquad \text{in} \quad \Omega, \tag{31}$$

$$u = 0 \qquad \text{on} \quad \Gamma. \tag{32}$$

The computational domain Ω is a quarter of an annulus situated in the positive quadrant of the Cartesian coordinate system. The inner radius R_i is equal to 1.0 and the outer radius R_o is equal to 4.0. We follow [38] and manufacture the source term f such that the exact solution to the problem is

$$u(x, y) = \theta(x, y)^2[\rho(x, y)^2 - 1][\rho(x, y)^2 - 16]\sin(x), \tag{33}$$

where $\rho(x, y) = \sqrt{x^2 + y^2}$ and $\theta(x, y) = \arctan(y/x)$ are the polar coordinates. Before we proceed further, let us denote the $L^2(\Omega)$ norm of a function u by $||u||_{L^2}$ and its H^1 seminorm by $|u|_{H^1}$. As usually, we take

$$||u||_{L^2} = \left(\int_\Omega u^2 \mathrm{d}\Omega\right)^{1/2} \quad \text{and} \quad |u|_{H^1} = \left(\int_\Omega |\nabla u|^2 \mathrm{d}\Omega\right)^{1/2} \tag{34}$$

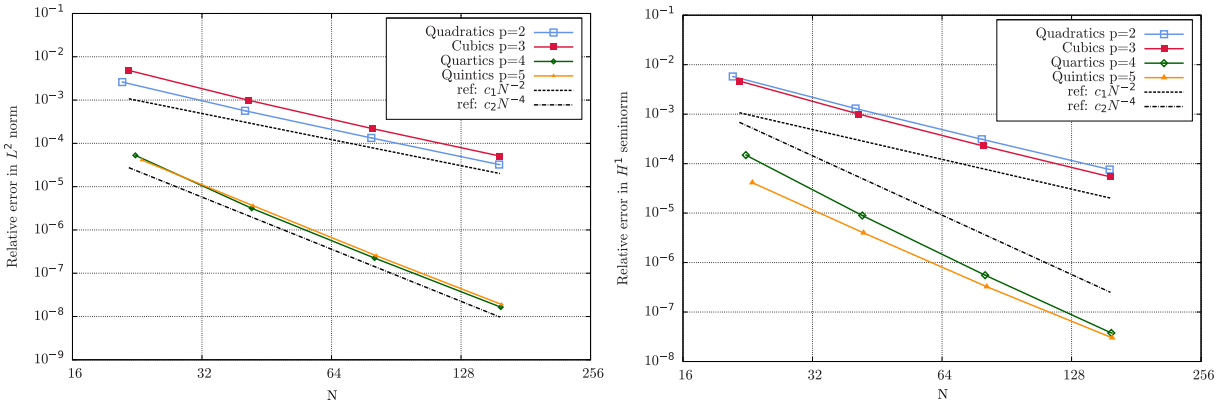


Figure 9: (Color online) Numerical convergence rates of r_{L^2} and r_{H^1}. The data suggest that the algorithm converges to the exact solution with rates $\mathcal{O}(p)$ and $\mathcal{O}(p-1)$ for even and odd degrees respectively.

We define the error of the numerical solution as $e^h = u^h - u$. The $L^2(\Omega)$ norm and the $H^1(\Omega)$ seminorm of the relative error are defined as

$$r_{L^2} = \frac{||e^h||_{L^2}}{||u||_{L^2}} \quad \text{and} \quad r_{H^1} = \frac{|e^h|_{H^1}}{|u|_{H^1}} \tag{35}$$

We solve this problem using quadratic, cubic, quartic, and quintic ASTS. We start considering a NURBS patch with 6×9 elements that represents the domain Ω exactly. Then, we perform quad-tree refinement using three levels of refinement in each mesh as it is illustrated in Fig. 8. We choose this type of refinement to be able to compare with the data in [38]. Fig. 8 also shows the numerical solutions for $p = 4$, which are indistinguishable at the scale of the plot. In Fig. 9 we report the numerical convergence rates of r_{L^2} and r_{H^1}. The plots show the evolution of the errors with respect to $N = \sqrt{n}$, which is the square root of the total number of degrees of freedom. The data suggest that the rates of convergence are of the type $\mathcal{O}(p)$ for even orders and $\mathcal{O}(p-1)$ for odd degrees. These results are consistent with those reported in [38].

Remarks:

1. One of the main advantages of NURBS-based isogeometric collocation is its efficiency, due to the fact that only one point evaluation per degree of freedom is needed. However, such an efficiency may be limited in many practical applications because NURBS lack local refinement capabilities. As anticipated in the introduction, in [38], isogeometric collocation was extended to hierarchical NURBS. This allowed local refinement, but the one point evaluation per degree of freedom was lost in the regions between levels of refinement due to the use of weighted collocation. The generalization of isogeometric collocation to ASTS combines the advantages of the two previous methodologies, allowing local refinement but still guaranteeing only one point evaluation per degree of freedom.

2. As highlighted earlier, IGA-C requires u^h to be at least $\mathcal{C}^2$-continuous in the neighborhood of collocation points, and the simplest way to satisfy this requirement is to take $p > 2$ and define a knot interval configuration that produces functions of maximum continuity. However, extensive numerical testing shows that also the $\mathcal{C}^1$ quadratic case appears to give good results in all situations[5], which is a significant practical result.

4.2. Second-order problem with rough solution

Here, we solve the Laplace equation on an L-shaped domain. More specifically, the problem is solved on $\Omega = [-1, 1]^2 \setminus ([0, 1] \times [-1, 0])$, as depicted in the left panel of Fig. 10. The problem can be stated as

$$\Delta u = 0 \qquad \text{in} \quad \Omega, \tag{36}$$

$$u = u_\Gamma \qquad \text{on} \quad \Gamma_D, \tag{37}$$

$$\nabla u \cdot \boldsymbol{n} = h_\Gamma \qquad \text{on} \quad \Gamma_N. \tag{38}$$

where Γ_D is composed by the re-entrant boundaries and $\Gamma_N = \Gamma \setminus \Gamma_D$ (cf. the left panel of Fig. 10). The functions u_Γ and h_Γ are defined in such a way that the exact solution is

$$u(x, y) = \rho(x, y)^{\frac{2}{3}} \sin\left(\frac{2\theta(x, y)}{3}\right), \tag{39}$$

where ρ and θ denote again the usual polar coordinates. It may be observed that the first partial derivatives of u, which can be expressed as,

$$\frac{\partial u}{\partial x} = -\frac{2\sin(\theta/3)}{3\sqrt[3]{\rho}}; \quad \frac{\partial u}{\partial y} = \frac{2\cos(\theta/3)}{3\sqrt[3]{\rho}} \tag{40}$$

tend to infinity as ρ tends to zero. Therefore, the exact solution of the problem is not smooth and this is expected to create a barrier in the convergence rate if uniform refinement is utilized. It is known, however, that local refinement can help alleviate this problem. The goal of this example is to show how our algorithm performs in this situation. In our computations, we take advantage of the existing symmetry in the problem and work with half of the domain as indicated on the top of the middle panel of Fig. 10. Our initial mesh is a NURBS patch with 10×10 elements which is plotted along with the corresponding numerical solution in Fig. 10. Then, we subsequently add several levels of refinement in order to have enhanced resolution close to the singularity as shown in the remaining panels of Fig. 10. The solutions are indistinguishable at the scale of the plot, but the numerical convergence rates shown in Fig. 11 are revealing. Here, we plot the evolution of r_{H^1} (see Eq. (35)) as a function of $N = \sqrt{n}$ for quadratic, cubic, quartic, and quintic ASTS. We show data for uniformly (dashed lines) and locally (solid lines) refined meshes. For uniform

[5]Cleary, in correspondence of a collocation point of $\mathcal{C}^1$-regularity, a single value has to be assigned to the discontinuous second derivative (e.g., the average of the different corresponding derivative values).

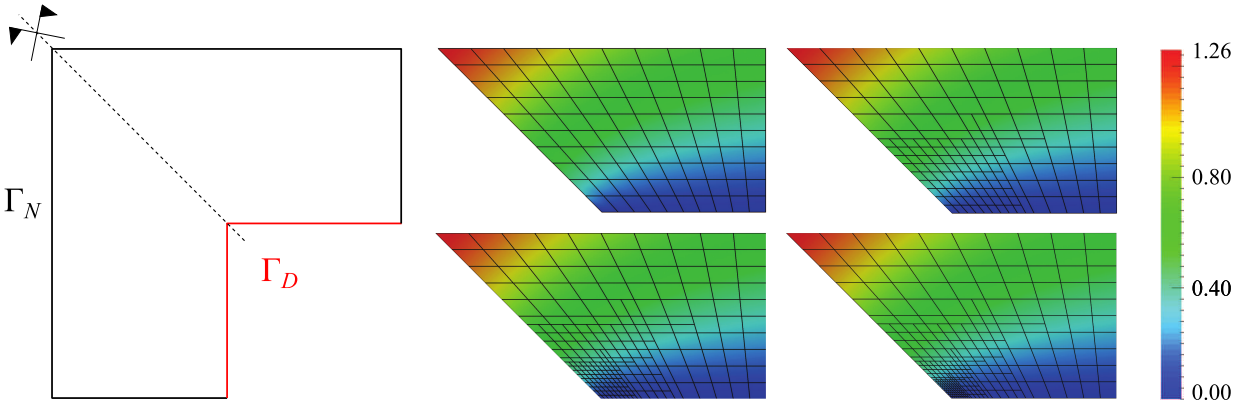


Figure 10: (Color online) Problem domain and boundary decomposition showing the symmetry of the problem (left). Making use of the symmetry, we compute the numerical solution on a uniform NURBS mesh and several locally-refined T-spline grids using $p = 2$ (middle and right panels).

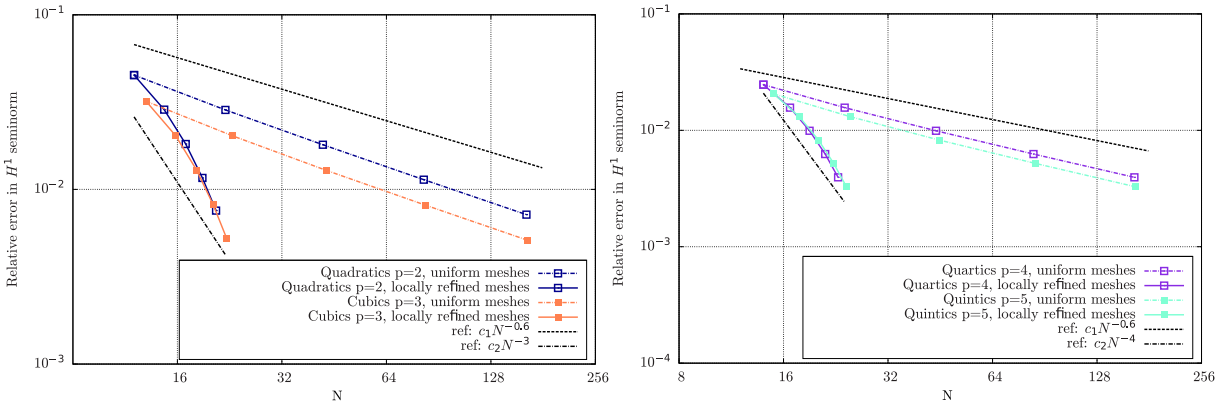


Figure 11: (Color online) Numerical rates of convergence of r_{L^2} and r_{H^1} using uniform (dashed lines) and local refinement (solid lines). With uniform refinement the convergence rate stalls at approximately 0.6 for all p. Local refinement produces significantly higher apparent convergence rates.

refinement the rate of convergence stalls at approximately $\mathcal{O}(0.6)$ irrespectively of p due to the lack of smoothness of the solution. Local refinement allows to achieve an apparent convergence rate which is significantly higher.

Remark:

In [40], it was reported the appearance of spurious oscillations in the numerical solution if Neumann boundary conditions are collocated in the standard way when the problem is characterized by a reduced regularity and elements have very high aspect ratios. Two alternatives were proposed in [40] in order to fix this issue, the so-called hybrid collocation and enhanced collocation methods. However, those techniques introduce small drawbacks in the overall algorithm, namely, hybrid collocation decreases the efficiency since the computation of boundary integrals is needed and enhanced collocation requires to determine a user-defined constant. Within NURBS-based IGA-C, the presence of elements with high aspect ratios is unavoidable in many practical situations, since refinement is always global for a NURBS patch. Instead, the local refinement properties of ASTS allow T-junctions making it possible to avoid elements with high aspect ratios. For example, if the local refinement is done as in Fig. 10, all elements have aspect ratio equal to one, except for a few elements with aspect ratio equal to two due to T-junction extensions. Therefore, as the current example clearly shows, the use of ASTS allows to collocate Neumann boundary conditions in the standard way without giving rise to any spurious oscillation in the numerical solution, also in the case of reduced regularity problems and highly locally refined meshes.

4.3. Advection-diffusion with boundary and internal layers

This section analyzes a classical advection-diffusion example typically referred to as advection skew to the mesh. The boundary-value problem can be written as

$$\boldsymbol{a} \cdot \nabla u - \nabla \cdot (\kappa \nabla u) = 0 \qquad \text{in} \quad \Omega, \tag{41}$$

$$u = u_\Gamma \qquad \text{on} \quad \Gamma. \tag{42}$$

where $\boldsymbol{a}$ denotes the (constant) velocity vector and κ the diffusion coefficient. The velocity field as well as the computational domain and u_Γ are depicted in Fig. 12. The angle of the velocity field is $\theta = \pi/4$. This is a classical benchmark in the literature of fluid mechanics and stabilized methods [79, 80, 81]. However, we do not focus on stabilization here, but on local refinement. The solution to the problem is controlled by a dimensionless quantity called Péclet number, which is defined as

$$Pe = \frac{|\boldsymbol{a}|L}{\kappa}, \tag{43}$$

where L is a length scale of the problem. We take $L = 1$, which is the side length of the problem domain. The Péclet number controls the relative strength of advection with

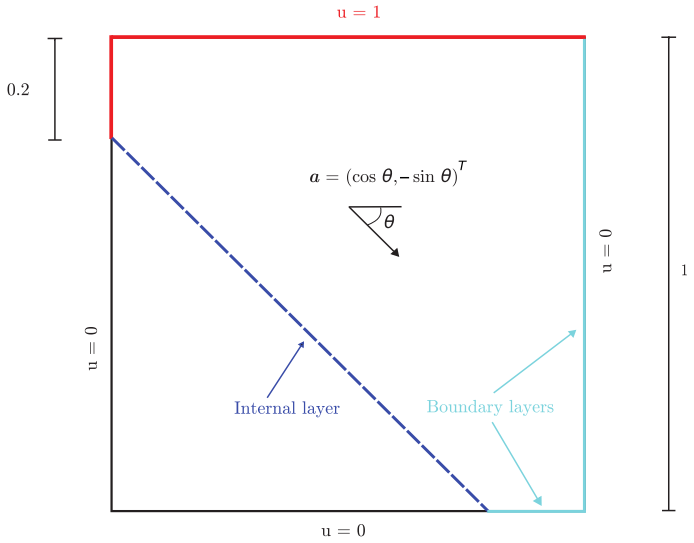


Figure 12: (Color online) Problem description and boundary conditions for the advection-diffusion benchmark example.

respect to diffusion. It is known from theoretical developments that, for large Pe, the diffusion operator is relevant only in thin layers characterized by the length scale $\sim \kappa/|\boldsymbol{a}|$. Therefore, for large Péclet number we expect sharp layers in the solution emanating from the discontinuity points in the boundary data as depicted in Fig. 12. Unless such layers are properly resolved by the mesh or a suitable stabilization is introduced, the numerical solution is expected to present spurious oscillations.

We solve this problem for $Pe = 10^3$ using cubic ASTS-based collocation, without introducing any stabilization term. We locally refine the mesh close to the boundary and internal layers, using five levels of refinement as shown on the left hand side of Fig. 13. The total number of degrees of freedom in the space that emanates from the mesh in $n = 5492$. The right hand side of Fig. 13 shows the numerical solution using our algorithm. The color scale goes from the minimum to the maximum value of the numerical solution. It can be observed that the usual undershoots and overshoots of the numerical solution are rather small with our algorithm, which shows the effectiveness of the approach.

4.4. Reaction-diffusion with spikes

The boundary-value problem is defined by

$$cu - \nabla \cdot (\kappa \nabla u) = 0 \qquad \text{in} \quad \Omega, \tag{44}$$

$$u = u_\Gamma \qquad \text{on} \quad \Gamma. \tag{45}$$

where the computational domain and u_Γ are depicted in Fig. 14 (top left panel). The problem parameters are the diffusion coefficient κ and the reaction rate c, but dimensional analysis can be used to show that the solution depends only on the Damköhler number

$$Da = \frac{cL^2}{\kappa}. \tag{46}$$

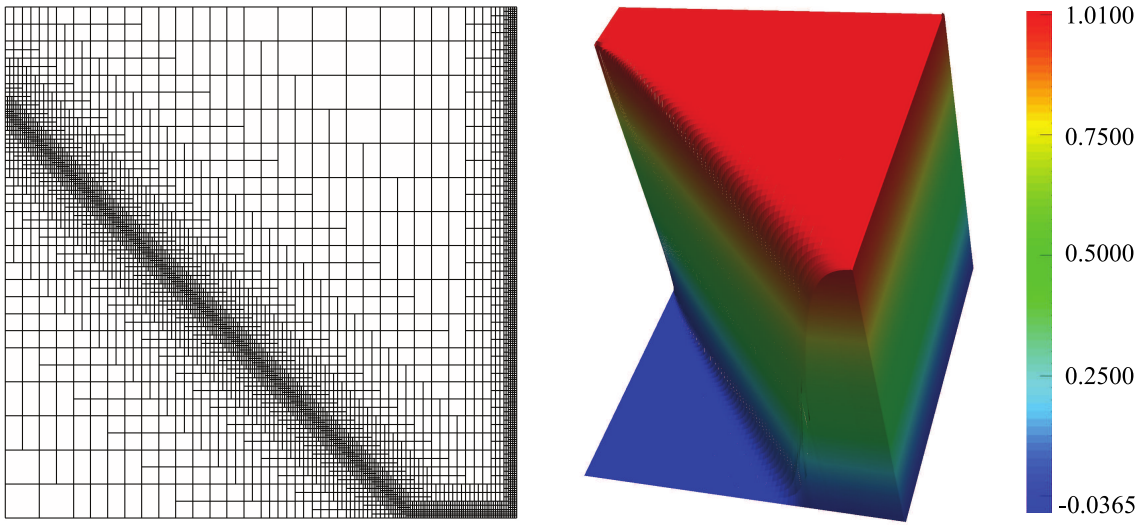


Figure 13: (Color online) Bézier mesh (left) and solution field (right) for the advection skew to the mesh problem. The Bézier mesh has five levels of refinement and gives rise to a total number of degrees of freedom $n = 5492$. The solution field is plotted with a color scale that ranges from the minimum to the maximum of the numerical solution showing that the usual undershoots and overshoots encountered in this problem are rather small.

Here, $L = 1$ is the side length of Ω. For large values of Da the solution is expected to be zero everywhere except close to the corners, where it progressively takes larger values until it reaches one as prescribed by the boundary conditions. We take the value $Da = 10^3$ in our simulations, following the studies on the same problem of [51]. We solve this problem using a quadratic ASTS mesh with seven levels of refinement in order to accurately capture the corner phenomena. The analysis results are plotted in the bottom panels of Fig. 14, where we only show a small region of the domain delimited in the top right panel. It can be seen that the solution appears to be stable even in the vicinity of very large gradients.

4.5. Circular Kirchhoff plate with a point load

Kirchhoff plates with distributed loads were solved using NURBS-based collocation methods in [42]. Here, we solve Kirchhoff plates with point loads using the local refinement capability of ASTS in order to refine close to the load[6]. A simply supported circular plate with radius $r = 1$ is considered. We apply a point load $P = 1$ to the plate's center. The equations that govern this problem can be obtained from Eqs. (25)–(29) taking $\Gamma_w = \Gamma$,

[6]We remark that, to our knowledge, this is the first time that isogeometric collocation is used to approximate problems with point loads.

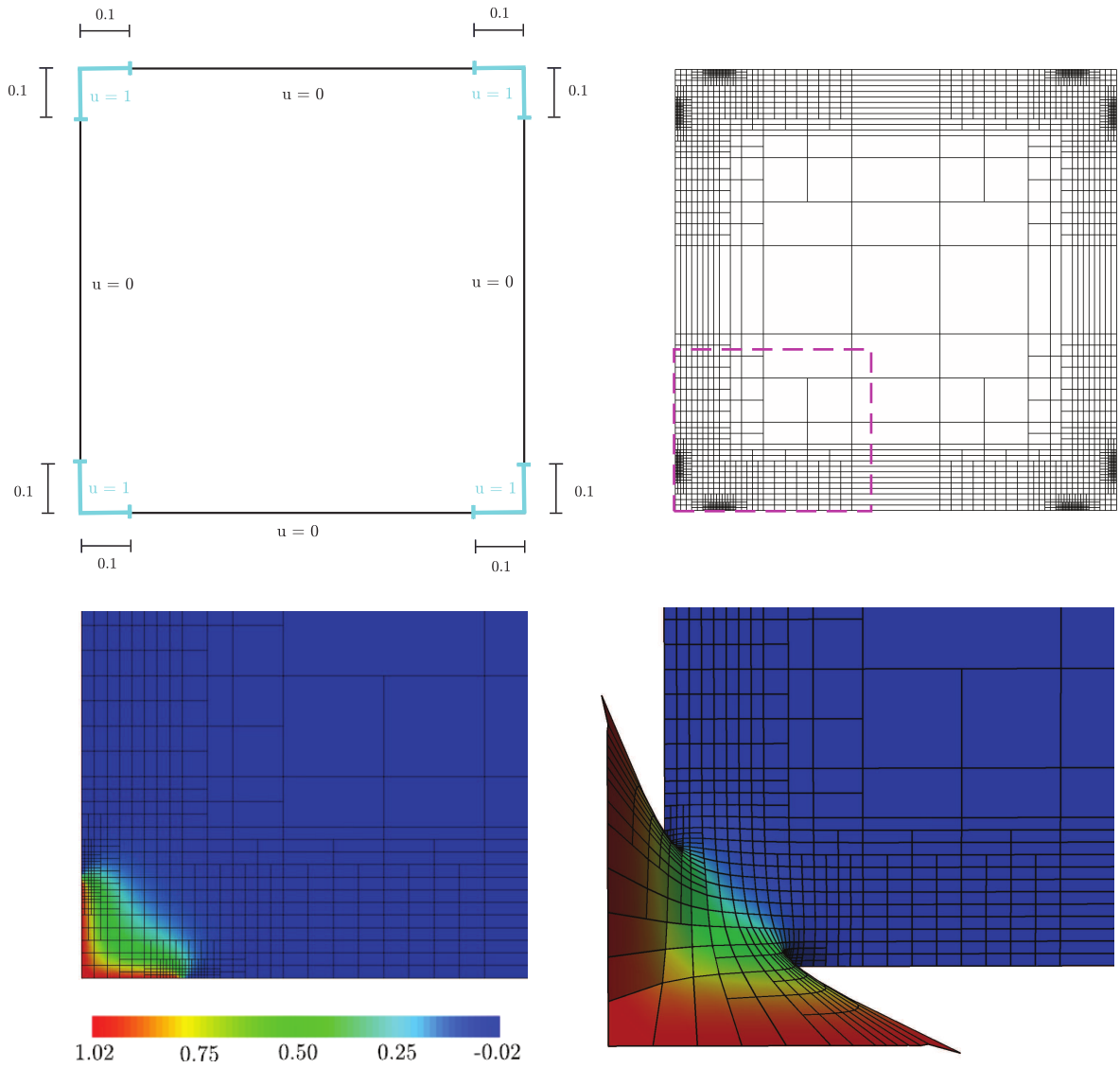


Figure 14: (Color online) From left to right and top to bottom, we present a panel showing the problem setup, the computational domain and the imposed boundary conditions. Then, we show the Bézier mesh ($n = 2563$) where we have used seven levels of refinement. In the Bézier mesh we mark a small area with a dashed magenta line. The bottom panels show this area zoomed in. On the left hand side, we plot the solution using a color scale that goes from the minimum to the maximum of the numerical solution. The solution looks stable even in the vicinity of very large gradients.

$\Gamma_M = \Gamma$, $w_\Gamma = 0$, and $M_\Gamma = 0$, yielding

$$D\Delta^2 w = g \qquad \text{in} \quad \Omega, \tag{47}$$

$$w = 0 \qquad \text{on} \quad \Gamma, \tag{48}$$

$$\nu D\Delta w + (1-\nu) D\boldsymbol{n} \cdot (\nabla\nabla w)\boldsymbol{n} = 0 \qquad \text{on} \quad \Gamma. \tag{49}$$

The Poisson ratio is assumed to be $\nu = 0.3$ and the rest of the material properties are selected such that $D = 1$. Finally, g is a load per unit surface which approximates the point load P. We model g as a function of the spline space called g^h. Let us assume for the time being that our discretization produces a Greville point in the plate's center, which can be achieved, for example, using a judicious geometrical mapping. As a precursor to g^h, let us define p^h as a function living in the discrete space which takes the value one in the Greville point that falls in the center of the plate and vanishes in the remaining Greville points. Therefore, we take $p^h = \sum_{A=1}^{n} p_A R_A$ such that

$$p^h(\boldsymbol{\tau}_A) = \begin{cases} 1 & \text{if } \boldsymbol{\tau}_A \text{ falls in the plate's center,} \\ 0 & \text{otherwise.} \end{cases} \tag{50}$$

The function g^h is simply defined as $g^h = \lambda p^h$, where the constant λ is determined imposing that g^h is statically equivalent to P, that is

$$\int_\Omega p^h \mathrm{d}\Omega = P \tag{51}$$

This completely defines g^h.

For the considered example, it is known that the exact solution for the deflection at the center of the plate is

$$w_0^{ex} = \frac{Pr^2}{16\pi D}\left(\frac{3+\nu}{1+\nu}\right). \tag{52}$$

We solve this problem using quintic ASTS. We start with a uniform NURBS mesh and end up with five levels of refinement in order to have higher resolution close to the center of the plate. Fig. 15 shows the coarsest and finest meshes along with the corresponding computed deflection fields. In Fig. 16, we plot the convergence of the deflection of the plate's center normalized with respect to the exact solution. A good convergence behavior is observed, and the relative error of the numerical solution with respect to the exact solution for the mesh with five levels of refinement is 0.06%.

Remark:

We acknowledge that our simple strategy for imposing point loads requires to have a Greville point where the load is applied. Although this can always be achieved with a judicious mapping, in the future, we will investigate in more detail the possibility of developing more general strategies to impose point loads within the isogeometric collocation framework, taking also inspiration from works like [82].

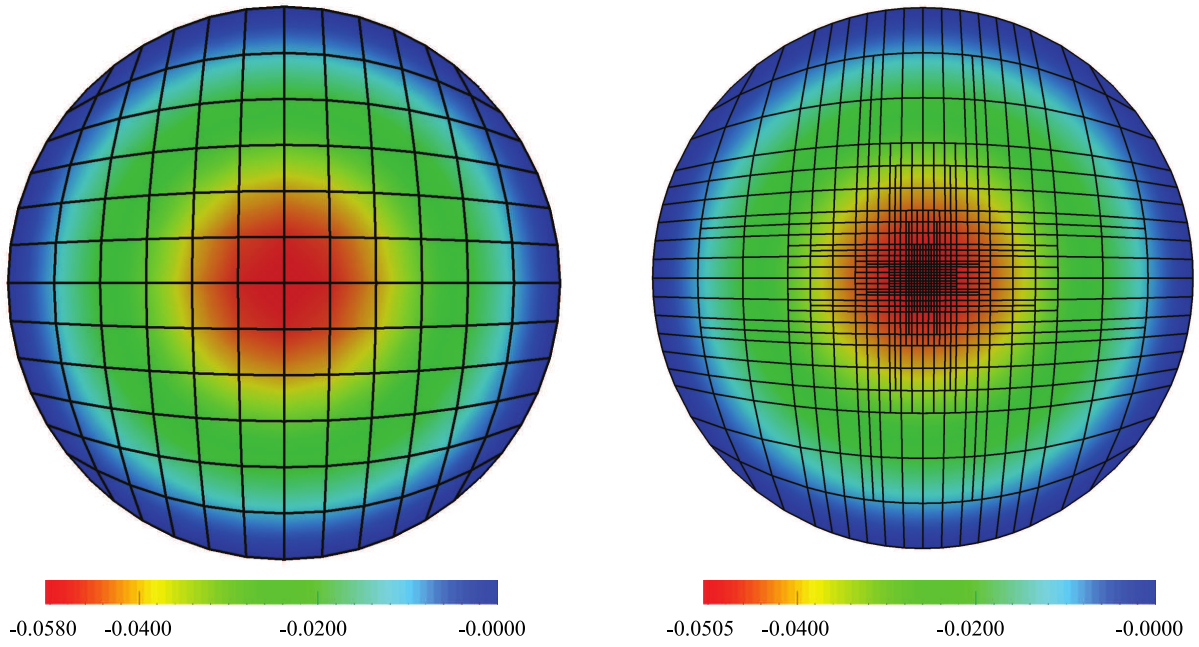


Figure 15: (Color online) Initial NURBS mesh (left) and final quintic ASTS mesh (right) along with the corresponding computed deflection fields. The mesh on the right hand side has five levels of refinement. We used $p = 5$ in both cases.

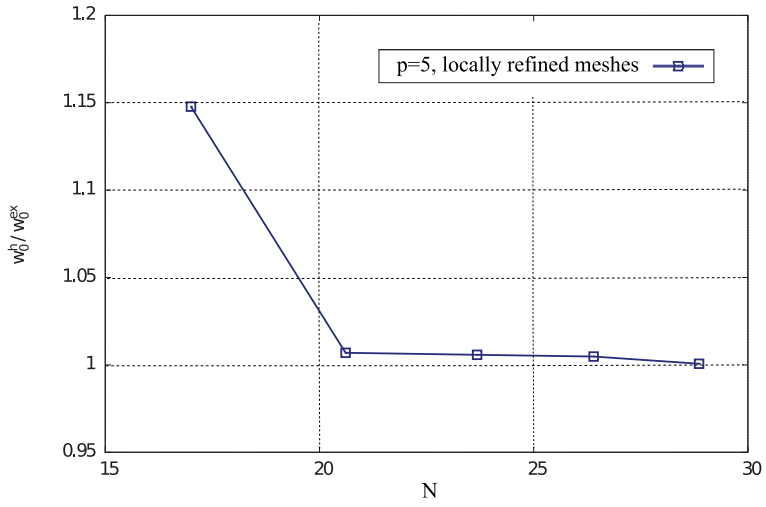


Figure 16: (Color online) Convergence of the numerical deflection at the center of the plate using quintic ASTS. We start with a uniform NURBS mesh and introduce a level of refinement in each new mesh. N denotes the square root of the total number of degrees of freedom.

5. Conclusions

This paper presents novel locally refinable isogeometric collocation methods based on analysis-suitable T-splines of arbitrary degree. The combination of isogeometric collocation and T-splines allows local refinement while keeping one point evaluation per degree of freedom, has good approximation properties, and is geometrically flexible. Moreover, the proposed methods are stable, accurate, and robust as shown by several significant numerical examples, where both second- and fourth-order boundary-value problems have been considered. We believe that the methodologies developed herein are highly suitable to tackle in an efficient way a large variety of demanding problems for which local refinement capabilities constitute a major advantage. Among others, these include shells and contact problems, topics that we plan to study in the near future.

Acknowledgements

Part of this work was accomplished during a four-month visit of HC at Carnegie Mellon University in 2014, for which financial support from the PhD student grant UDC-Inditex is gratefully acknowledged. HG and HC were partially supported by the European Research Council through the FP7 Ideas Starting Grant (project # 307201). LL and YZ were partially supported by the PECASE (project # N00014-14-10234). AR was partially supported by the European Research Council through the FP7 Ideas Starting Grant (project # 259229). This support is gratefully acknowledged. Finally, we acknowledge the open source scientific library PETSc and their developers.

References

[1] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis CAD, finite elements, NURBS, exact geometry and mesh refinement, Computational Methods in Applied Mechanics and Engineering 194 (2005) 4135–4195.

[2] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric Analysis Toward Integration of CAD and FEA, Wiley, 2009.

[3] M. Borden, C. Verhoosel, M. Scott, T. J. R. Hughes, C. Landis, A phase-field description of dynamic brittle fracture, Computer Methods in Applied Mechanics and Engineering 217-220 (2012) 77–95.

[4] J. Caseiro, R. Valente, A. Reali, J. Kiendl, F. Auricchio, R. Alves de Sousa, Assumed natural strain NURBS-based solid-shell element for the analysis of large deformation elasto-plastic thin-shell structures, Computer Methods in Applied Mechanics and Engineering 284 (2015) 861–880.

[5] R. Dhote, H. Gomez, R. Melnik, J. Zu, Isogeometric analysis of a dynamic thermo-mechanical phase-field model applied to shape memory alloys, Computational Mechanics 53 (2013) 1235–1250.

[6] T. Elguedj, Y. Bazilevs, V. M. Calo, T. J. R. Hughes, $\overline{B}$ and $\overline{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements, Computer methods in applied mechanics and engineering 197 (2008) 2732–2762.

[7] F. Auricchio, M. Conti, M. Ferraro, S. Morganti, A. Reali, R. Taylor, Innovative and efficient stent flexibility simulations based on isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 295 (2015) 347–361.

[8] T. J. R. Hughes, J. Evans, A. Reali, Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems, Computational Methods in Applied Mechanics and Engineering 272 (2014) 290–320.

[9] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, T. J. R. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Computer Methods in Applied Mechanics and Engineering 199 (2010) 357–373.

[10] S. Morganti, F. Auricchio, D. Benson, F. Gambarin, S. Hartmann, T. J. R. Hughes, A. Reali, Patient-specific isogeometric structural analysis of aortic valve closure, Computer Methods in Applied Mechanics and Engineering 294 (2015) 428–448.

[11] F. Auricchio, L. Beirão da Veiga, A. Buffa, C. Lovadina, A. Reali, G. Sangalli, A fully locking-free isogeometric approach for plane linear elasticity problems: A stream function formulation, Computer Methods in Applied Mechanics and Engineering 197 (2007) 160–172.

[12] I. Akkerman, Y. Bazilevs, V. M. Calo, T. J. R. Hughes, S. Hulshoff, The role of continuity in residual-based variational multiscale modeling of turbulence, Computational Mechanics 41 (2008) 371–378.

[13] Y. Bazilevs, V. M. Calo, J. A. Cottrell., T. J. R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Computer Methods in Applied Mechanics and Engineering 197 (2007) 173–201.

[14] J. Liu, H. Gomez, J. A. Evans, T. J. R. Hughes, C. M. Landis, Functional entropy variables: A new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier–Stokes–Korteweg equations, Journal of Computational Physics 248 (2013) 47–86.

[15] H. Gomez, T. J. R. Hughes, X. Nogueira, V. M. Calo, Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1828–1840.

[16] H. Gomez, X. Nogueira, A new space–time discretization for the Swift–Hohenberg equation that strictly respects the Lyapunov functional, Communications in Nonlinear Science and Numerical Simulation 17 (2012) 4930–4946.

[17] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, Y. Zhang, Isogeometric fluid-structure interaction: Theory, algorithms, and computations, Computational Mechanics 43 (2008) 3–37.

[18] Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, T. J. R. Hughes, Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, Computer Methods in Applied Mechanics and Engineering 196 (2007) 2943–2959.

[19] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, T. Tezduyar, 3D simulation of wind turbine rotors at full scale. part I: Geometry modeling and aerodynamics, International Journal for Numerical Methods in Fluids 65 (2011) 207–235.

[20] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wuchner, K.-U. Bletzinger, 3D simulation of wind turbine rotors at full scale. part II: Fluid-structure interaction modeling with composite blades, International Journal for Numerical Methods in Fluids 65 (2011) 236–253.

[21] H. Casquero, C. Bona-Casas, H. Gomez, A NURBS-based immersed methodology for fluid-structure interaction, Computer Methods in Applied Mechanics and Engineering 284 (2015) 943–970.

[22] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, T. J. R. Hughes, An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves, Computer Methods in Applied Mechanics and Engineering 284 (2015) 1005–1053.

[23] M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. Sacks, T. J. R. Hughes, Fluid-structure interaction analysis of bioprosthetic heart valves: Significance of arterial wall deformation, Computational Mechanics 54 (2014) 1055–1071.

[24] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. Wu, J. Mineroff, A. Reali, Y. Bazilevs, M. Sacks, Dynamic and fluid-structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models, Computational Mechanics 55 (2015) 1211–1225.

[25] J. Bueno, C. Bona-Casas, Y. Bazilevs, H. Gomez, Interaction of complex fluids and solids: theory, algorithms and application to phase-change-driven implosion, Computational Mechanics (2014) 1–14.

[26] H. Gomez, V. M. Calo, Y. Bazilevs, T. J. R. Hughes, Isogeometric analysis of the Cahn-Hilliard phase-field model, Computer Methods in Applied Mechanics and Engineering 197 (49-50) (2008) 4333–4352.

[27] H. Gomez, L. Cueto-Felgueroso, R. Juanes, Three-dimensional simulation of unstable gravity-driven infiltration of water into a porous medium, Journal of Computational Physics 238 (2013) 217–239.

[28] H. Gomez, X. Nogueira, An unconditionally energy-stable method for the phase field crystal equation, Computer Methods in Applied Mechanics and Engineering 249 (2012) 52–61.

[29] H. Gomez, T. J. R. Hughes, Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models, Journal of Computational Physics 230 (2011) 5310–5327.

[30] G. Vilanova, I. Colominas, H. Gomez, Capillary networks in tumor angiogenesis: From discrete endothelial cells to phase-field averaged descriptions via isogeometric analysis, International Journal for Numerical Methods in Biomedical Engineering 29 (2013) 1015–1037.

[31] G. Vilanova, I. Colominas, H. Gomez, Coupling of discrete random walks and continuous modeling for three-dimensional tumor-induced angiogenesis, Computational Mechanics 53 (2013) 449–464.

[32] J. Kiendl, K.-U. Bletzinger, J. Linhard, R. Wuchner, Isogeometric shell analysis with Kirchhoff-Love elements, Computer Methods in Applied Mechanics and Engineering 198 (2009) 3902–3914.

[33] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, K.-U. Bletzinger, The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches, Computer Methods in Applied Mechanics and Engineering 199 (2010) 2403–2416.

[34] J. Kiendl, M.-C. Hsu, M. C. Wu, A. Reali, Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials, Computer Methods in Applied Mechanics and Engineering 291 (2015) 280–303.

[35] C. Adam, T. J. R. Hughes, S. Bouabdallah, M. Zarroug, H. Maitournam, Selective and reduced numerical integrations for NURBS-based isogeometric analysis., Computer Methods in Applied Mechanics and Engineering 284 (2015) 732–761.

[36] F. Auricchio, L. Beirão Da Veiga, T. J. R. Hughes., A. Reali, G. Sangalli, Isogeometric collocation methods, Mathematical Models and Methods in Applied Sciences 20 (2010) 2075–2107.

[37] F. Auricchio, L. Bẽirao da Veiga, T. J. R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation for elastostatics and explicit dynamics, Computer Methods in Applied Mechanics and Engineering 249 (2012) 2–14.

[38] D. Schillinger, J. Evans, A. Reali, M. Scott, T. J. R. Hughes, Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Computer Methods in Applied Mechanics and Engineering 267 (2013) 170–232.

[39] H. Gomez, A. Reali, G. Sangalli, Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models, Journal of Computational Physics 262 (2014) 153–171.

[40] L. De Lorenzis, J. Evans, T. J. R. Hughes, A. Reali, Isogeometric collocation: Neumann boundary conditions and contact, Computer Methods in Applied Mechanics and Engineering 284 (2015) 21–54.

[41] R. Kruse, N. Nguyen-Thanh, L. De Lorenzis, T. J. R. Hughes, Isogeometric collocation for large deformation elasticity and frictional contact problems, Computer Methods in Applied Mechanics and Engineering 296 (2015) 73–112.

[42] A. Reali, H. Gomez, An isogeometric collocation approach for Bernoulli-Euler beams and Kirchhoff plates, Computer Methods in Applied Mechanics and Engineering 284 (2015) 623–636.

[43] L. Beirão da Veiga, C. Lovadina, A. Reali, Avoiding shear locking for the Timoshenko beam problem via isogeometric collocation methods, Computer Methods in Applied Mechanics and Engineering 241-244 (2012) 38–51.

[44] F. Auricchio, L. Beirão da Veiga, J. Kiendl, C. Lovadina, A. Reali, Locking-free isogeometric collocation methods for spatial Timoshenko rods, Computer Methods in Applied Mechanics and Engineering 263 (2013) 113–126.

[45] J. Kiendl, F. Auricchio, L. Beirão da Veiga, C. Lovadina, A. Reali, Isogeometric collocation methods for the Reissner–Mindlin plate problem, Computer Methods in Applied Mechanics and Engineering 284 (2015) 489–507.

[46] J. Kiendl, F. Auricchio, T. J. R. Hughes, A. Reali, Single-variable formulations and isogeometric discretizations for shear deformable beams, Computer Methods in Applied Mechanics and Engineering 284 (2015) 988–1004.

[47] C. Manni, A. Reali, H. Speleers, Isogeometric collocation methods with generalized b-splines, Computers and Mathematics with Applications-doi:10.1016/j.camwa.2015.03.027.

[48] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Trans. Graph. 22 (2003) 477–484.

[49] T. W. Sederberg, J. Zheng, X. Song, Knot intervals and multi-degree splines, Computer Aided Geometric Design 20 (7) (2003) 455–468.

[50] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, in: ACM Transactions on Graphics (TOG), Vol. 23, ACM, 2004, pp. 276–283.

[51] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. J. R. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using T-splines, Computer Methods in Applied Mechanics and Engineering 199 (2010) 229–263.

[52] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, M. A. Scott, On linear independence of T-spline blending functions, Computer Aided Geometric Design 29 (2012) 63–76.

[53] L. Beirão da Veiga, A. Buffa, G. Sangalli, R. Vázquez, Analysis suitable T-splines of arbitrary degree: Definition, linear independence, and approximation properties, Mathematical Models and Methods in Applied Sciences 23 (11) (2013) 1979–2003.

[54] L. Beirão da Veiga, A. Buffa, D. Cho, G. Sangalli, Analysis-suitable T-splines are dual-compatible, Computer methods in applied mechanics and engineering 249 (2012) 42–51.

[55] M. Scott, X. Li, T. Sederberg, T. J. R. Hughes, Local refinement of analysis-suitable T-splines, Computer Methods in Applied Mechanics and Engineering 213-216 (2012) 206–222.

[56] Y. Zhang, W. Wang, T. J. R. Hughes, Solid T-spline construction from boundary representations for genus-zero geometry, Computer Methods in Applied Mechanics and Engineering 249-252 (2012) 185–197.

[57] Y. Zhang, W. Wang, T. J. R. Hughes, Conformal solid T-spline construction from boundary T-spline representations, Computational Mechanics 51 (2013) 1051–1059.

[58] W. Wang, Y. Zhang, L. Liu, T. J. R. Hughes, Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology, Computer-Aided Design 45 (2) (2013) 351–360.

[59] L. Liu, Y. Zhang, T. J. R. Hughes, M. A. Scott, T. W. Sederberg, Volumetric T-spline construction using boolean operations, Engineering with Computers 30 (2014) 425–439.

[60] L. Liu, Y. Zhang, Y. Liu, W. Wang, Feature-preserving T-mesh construction using skeleton-based polycubes, Computer-Aided Design 58 (2015) 162–172.

[61] L. Liu, Y. Zhang, X. Wei, Weighted T-splines with application in reparameterizing trimmed NURBS surfaces, Computer Methods in Applied Mechanics and Engineering 295 (2015) 108 – 126.

[62] W. Wang, Y. Zhang, M. A. Scott, T. J. R. Hughes, Converting an unstructured quadrilateral mesh to a standard T-spline surface, Computational Mechanics 48 (2011) 477–498.

[63] Y. Bazilevs, M.-C. Hsu, M. Scott, Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines, Computer Methods in Applied Mechanics and Engineering 249-252 (2012) 28–41.

[64] H. Casquero, L. Lei, C. Bona-Casas, J. Zhang, H. Gomez, A hybrid Galerkin-collocation immersed method for fluid-structure interaction using unstructured T-splines, International Journal for Numerical Methods in Engineering. DOI: 10.1002/nme.5004.

[65] C. V. Verhoosel, M. A. Scott, R. de Borst, T. J. R. Hughes, An isogeometric approach to cohesive zone modeling, International Journal for Numerical Methods in Engineering 87 (2011) 336–360.

[66] C. V. Verhoosel, M. A. Scott, T. J. R. Hughes, R. de Borst, An isogeometric analysis approach to gradient damage models, International Journal for Numerical Methods in Engineering 86 (2011) 115–134.

[67] R. Dimitri, L. D. Lorenzis, M. Scott, P. Wriggers, R. Taylor, G. Zavarise, Isogeometric large deformation frictionless contact using T-splines, Computer Methods in Applied Mechanics and Engineering 269 (2014) 394–414.

[68] R. Simpson, M. Scott, M. Taus, D. Thomas, H. Lian, Acoustic isogeometric boundary element analysis, Computer Methods in Applied Mechanics and Engineering 269 (2014) 265–290.

[69] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric methods for computational electromagnetics: B-spline and T-spline discretizations, Journal of Computational Physics 257(B) (2014) 1291–1320.

[70] H. Casquero, L. Lei, Y. Zhang, A. Reali, J. Kiendl, H. Gomez, Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff-Love shells, in preparation.

[71] G. Finnigan, Arbitrary degree T-splines, Masters thesis, Brigham Young University, 2008.

[72] M. J. Borden, M. A. Scott, J. A. Evans, T. J. R. Hughes, Isogeometric finite element data structures based on Bézier extraction of NURBS, International Journal for Numerical Methods in Engineering 87 (1-5) (2011) 15–47.

[73] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, T. J. R. Hughes, Isogeometric finite element data structures based on Bézier extraction of T-splines, International Journal for Numerical Methods in Engineering 88 (2011) 126–156.

[74] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, T. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, Computer Methods in Applied Mechanics and Engineering 254 (2013) 197–221.

[75] L. Liu, Y. Zhang, X. Wei, Handling extraordinary nodes with weighted T-spline basis functions, 24th International Meshing Roundtable, accepted, 2015.

[76] T. J. R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.

[77] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, H. Zhang, PETSc Web page, `http://www.mcs.anl.gov/petsc` (2014).

[78] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory (2013).

[79] A. N. Brooks, T. J. Hughes, Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations, Computer methods in applied mechanics and engineering 32 (1) (1982) 199–259.

[80] L. P. Franca, S. L. Frey, T. J. Hughes, Stabilized finite element methods: I. application to the advective-diffusive model, Computer Methods in Applied Mechanics and Engineering 95 (2) (1992) 253–276.

[81] H. Gomez, I. Colominas, F. Navarrina, J. París, M. Casteleiro, A hyperbolic theory for advection-diffusion problems: Mathematical foundations and numerical modeling, Archives of Computational Methods in Engineering 17 (2) (2010) 191–211.

[82] Q. Dang, M. Ehrhardt, On Dirac delta sequences and their generating functions, Applied Mathematics Letters 25 (2012) 2385–2390.