International Journal for Uncertainty Quantification, 11(3):27–57 (2021)

ISOGEOMETRIC METHODS FOR KARHUNEN-LOÈVE REPRESENTATION OF RANDOM FIELDS ON ARBITRARY MULTIPATCH DOMAINS

R. Jahanbin & S. Rahman*

College of Engineering, The University of Iowa, Iowa City, Iowa 52242, USA

*Address all correspondence to: R. Jahanbin, College of Engineering, The University of Iowa, Iowa City, Iowa 52242, USA, E-mail: ramin-jahanbin@uiowa.edu

Original Manuscript Submitted: 6/18/2020; Final Draft Received: 9/20/2020

This paper introduces isogeometric Galerkin and collocation methods for solving the Fredholm integral eigenvalue problem on arbitrary multipatch domains, delivering the Karhunen-Loève expansion for random field discretization. In both methods, the unknown eigenfunctions are projected onto concomitant finite-dimensional approximation spaces, where nonuniform rational B-splines and analysis-suitable T-splines reside. In the context of isogeometric analysis, the geometry is modeled precisely, and identical basis functions with significant approximating power are employed for modeling the geometry and constructing the approximation spaces. Numerical analyses of two- and three-dimensional engineering problems indicate that the Galerkin- and collocation-derived eigensolutions are both convergent and accurate. However, the collocation method, by eliminating one d-dimensional domain integration in forming the system matrices, produces eigensolutions markedly more economically than the Galerkin method. Highly effective in large-scale applications, the isogeometric collocation method imparts a tremendous boost to computational expediency. As a result, subsequent uncertainty quantification analysis of complex engineering structures requiring multipatch geometry representation can now be performed using the proposed methods for random field discretization.

KEY WORDS: *uncertainty quantification, Karhunen-Loève expansion, isogeometric analysis, NURBS, T-splines, Bézier extraction*

1. INTRODUCTION

Random input parameters with spatial variability are prevalent in engineering and applied sciences. Geometrical characteristics, thermal conductivity, mechanical strength, and applied loads are examples of such randomness. In these cases, the uncertainty emerges not only from sample to sample, but also from point to point within a domain of interest. Since there are infinitely many points in the domain, the continuous-parameter random field is infinite-dimensional. Hence, for the computational methods to be empirically applicable, representing the random field in terms of a finite number of random variables is crucial. The question here is not only about the quality of the approximation, but also about the computational effort in the subsequent uncertainty quantification (UQ) analysis.

Numerous methods have been developed to provide surrogates for continuous-parameter random fields. One prominent choice is the Karhunen-Loève (K-L) expansion [1,2], which entails spectral decomposition of the covariance function into an infinite series, comprising deterministic functions of spatial arguments and uncorrelated random variables [3]. The K-L expansion is generally favorable because it is optimal in the global mean-square error with respect to the number of random variables in the expansion [4]. However, the expansion must be truncated judiciously at a decent trade-off between accuracy and a minimal number of random variables for computational expediency in the ensuing UQ analysis [5]. Moreover, to write the expansion, one must solve the Fredholm eigenvalue problem [6] for the eigensolutions, which can be challenging, especially in the case of complex domains and/or inseparable covariance functions [7]. As a result, numerical methods with fitting eigenvalue solvers are fashionable for approximating the eigensolutions [8,9].

2152–5080/21/$35.00

Isogeometric analysis (IGA) [10,11] has bypassed the need for converting computer-aided design (CAD) models to traditional finite element analysis (FEA) discretization. In addition, many canonical shapes are exactly modeled with smooth B-splines and nonuniform rational B-splines (NURBS). Regarding multipatch domains, however, there have been studies on watertight modeling of complex shapes [12,13]. The watertight trimmed NURBS [14] have been introduced accordingly, and T-splines [12] have been employed in the framework of IGA [15]. T-splines were originally developed as a generalization of NURBS and enabled the influential feature of local refinement [16,17], which saved valuable time in mesh generation and analysis. The linear independence of T-splines [18,19] is crucial to form analysis-suitable basis functions [20]. Furthermore, the Bézier extraction of T-splines as a finite element data structure [21] and a means to evaluate properties, such as the partition of unity [22], has been established. Moreover, hierarchical T-splines [23,24] and truncated T-splines [25] have been proposed to model complex geometries more conveniently. However, the recently developed technologies have not yet been commercialized in a software package to enable analysis of any arbitrary geometry with a user-friendly graphical interface.

Several numerical methods for solving the Fredholm eigenvalue problem have been reviewed in the work by Betz et al. [4], such as the Nyström formulation, the collocation approach, and the Galerkin finite element method (FEM) [26], to name just three. The classical FEM utilizes an approximation space of polynomials with C^0-continuity at the element boundaries. On the other hand, the mesh-free approach [27,28] generally results in smoother eigensolutions than those obtained by classical FEM. For fast computation of the K-L expansion coefficients, an embedding technique [29] has been developed to avoid mesh generation, and a method has been proposed based on the domain independence property of the K-L expansion with the Nyström formulation of the problem [30]. For solving the Fredholm integral problem with IGA methods, investigations employing both Galerkin [31] and collocation [32] projections have been reported. Both methods so far are limited to tensor-product single-patch geometries, where the former involved sixfold integral computations for three-dimensional problems, thereby typically demanding extensive resources and time, which was addressed by the latter. To efficiently compute high-dimensional integrals in the context of IGA, low-rank tensor approximation methods [33] and sparse grid methods [34] have been proposed. In UQ applications, however, since the former approximates the integrand, it may not be as accurate when the function is complex or inseparable, while the latter is generally effective when the dimension of integration is very large. The descretized random field has been used in UQ analysis, for example, stochastic isogeometric analysis [35,36]. T-splines are mostly used to solve conventional boundary value problems (BVPs). With the latest progress with T-splines, however, the previous research on K-L expansion [31,32] must be improved on to handle arbitrary multipatch domains. In this sense, the prefix "iso" refers to utilizing identical basis functions in geometrical modeling and solving an eigenvalue problem, rather than the heavily studied BVPs in engineering and applied mathematics. The motivation here is to accurately discretize general random fields on arbitrary multipatch domains by harnessing the approximating power of NURBS and T-splines, while the geometry is modeled exactly to the extent possible.

This paper presents novel isogeometric Galerkin and collocation methods for random field discretization on multipatch domains. The multipatch NURBS and T-spline methods are used to project the solution of the Fredholm integral problem to the approximation spaces, where NURBS and T-splines reside. The paper is organized as follows. Section 2 provides a formal definition for a general random field along with a concise description of K-L approximation. The paraphernalia of IGA, namely, NURBS, T-splines, and Bézier extraction of T-splines, are introduced in Section 3 for the paper to be self-contained. Section 4 entails the formulation of the Fredholm integral problem on multipatch domains via Galerkin and collocation projections and discusses how the system matrices are constructed. Furthermore, the eigenvalue solvers are briefly explained. Two numerical examples are provided in Section 5 to investigate the accuracy and computational effort of the proposed methods. A brief discussion is presented in Section 6, before the conclusions are drawn in Section 7. Furthermore, Appendix A concisely introduces B-splines.

2. RANDOM FIELD DISCRETIZATION

Let $\mathbb{N} := \{1, 2, \ldots\}$, $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$, and $\mathbb{R} := (-\infty, +\infty)$ represent the sets of natural, non-negative integers, and real numbers, respectively. Denote by d the dimension of the physical domain $\mathcal{D}$ of a geometrical object, which can be a line ($d = 1$), surface ($d = 2$), or solid ($d = 3$). In theory, $d = 1, 2, 3$, but multidimensional problems ($d = 2, 3$)

are emphasized in this work. Also, it is assumed that $\mathcal{D} \subset \mathbb{R}^d$ is a closed bounded domain. These standard notations will be used throughout the paper.

Let a complete probability space be defined by $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω, $\mathcal{F}$, and $\mathbb{P} : \mathcal{F} \to [0, 1]$ are the sample space, σ-field, and a probability measure, respectively. The random variables in the Hilbert space $L^2(\Omega, \mathcal{F}, \mathbb{P})$ have finite second-moment properties, as do the random fields defined on the closed bounded domain $\mathcal{D}$ in the Hilbert space $L^2(\mathcal{D} \times \Omega)$.

2.1 Random Field

A real-valued random field on the domain $\mathcal{D} \subset \mathbb{R}^d$, where $d = 1, 2,$ or 3, is defined as a mapping $\alpha : \mathcal{D} \times \Omega \to \mathbb{R}$, such that at a physical point $\mathbf{z} \in \mathcal{D}$, $\alpha(\mathbf{z}, \cdot)$ is a random variable corresponding to the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For this random field, denote by

$$\mu(\mathbf{z}) := \mathbb{E}[\alpha(\mathbf{z}, \cdot)],$$

and

$$\Gamma(\mathbf{z}, \mathbf{z}') := \mathbb{E}[(\alpha(\mathbf{z}, \cdot) - \mu(\mathbf{z}))(\alpha(\mathbf{z}', \cdot) - \mu(\mathbf{z}'))],$$

the mean and continuous covariance function of $\alpha(\mathbf{z}, \cdot)$, respectively, where $\mathbb{E}$ is the expectation operator associated with the probability measure $\mathbb{P}$ and $\Gamma : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ is a Hilbert-Schmidt kernel, satisfying

$$\int_{\mathcal{D}} \int_{\mathcal{D}} |\Gamma^2(\mathbf{z}, \mathbf{z}')| d\mathbf{z} d\mathbf{z}' < \infty. \tag{1}$$

Equation (1) implies that the covariance function must be square-integrable. There is no fundamental restriction on the way the distance between $\mathbf{z}$ and $\mathbf{z}'$ is defined in the covariance function $\Gamma(\mathbf{z}, \mathbf{z}')$. In other words, different types of distance between any given pair points in the domain can be defined, as long as the covariance function satisfies (1). For instance, for circular or spherical domains, the great-circle distance may be a relatively more convenient choice than the Euclidean distance. This study considers homogeneous random fields only.

For the equations to be simpler, without loss of generality, assume that the random fields are zero-mean; i.e., $\mu(\mathbf{z}) = 0$ in the remainder of the paper.

2.2 Fredholm Integral Problem of Second Kind

For a random field, as described in Section 2.1, define by

$$(\mathcal{G}_\Gamma \phi)(\mathbf{z}) := \int_{\mathcal{D}} \Gamma(\mathbf{z}, \mathbf{z}') \phi(\mathbf{z}') d\mathbf{z}' \quad \forall \phi(\mathbf{z}) \in L^2(\mathcal{D}), \tag{2}$$

a Hilbert-Schmidt integral operator $\mathcal{G}_\Gamma : L^2(\mathcal{D}) \to L^2(\mathcal{D})$ [37] with $L^2(\mathcal{D})$ representing a Hilbert space of real-valued square-integrable functions on $\mathcal{D} \subset \mathbb{R}^d$. The integral operator defined in Eq. (2) is linear, compact, positive-semidefinite, and self-adjoint. The Fredholm integral equation of the second kind is thus formulated by dint of the Hilbert-Schmidt operator as

$$(\mathcal{G}_\Gamma \phi)(\mathbf{z}) = \lambda \phi(\mathbf{z}) \text{ or } \int_{\mathcal{D}} \Gamma(\mathbf{z}, \mathbf{z}') \phi(\mathbf{z}') d\mathbf{z}' = \lambda \phi(\mathbf{z}), \tag{3}$$

for which there is an infinite sequence of eigensolutions $\{\lambda_i, \phi_i(\mathbf{z})\}_{i \in \mathbb{N}}$. Having the eigenfunctions normalized as $\|\phi_i(\mathbf{z})\|^2_{L^2(\mathcal{D})} := \int_{\mathcal{D}} \phi_i^2(\mathbf{z}) d\mathbf{z} = 1$, the eigensolutions have several useful properties as follows:

1. The eigenvalues $\lambda_i,\ i \in \mathbb{N}$ are real and non-negative.
2. The eigenvalues have zero as the only point of accumulation.
3. The eigenfunctions $\phi_i(\mathbf{z}),\ i \in \mathbb{N}$ are members of the Hilbert space $L^2(\mathcal{D})$ and are mutually orthonormal.
4. The eigenfunctions form an orthornormal basis of $L^2(\mathcal{D})$. In other words, $L^2(\mathcal{D}) = \text{span}\{\phi_i(\mathbf{z})\}_{i \in \mathbb{N}}$.

The first two of the aforementioned properties will be revisited throughout the paper.

2.3 K-L Expansion

Let $\{X_i\}_{i\in\mathbb{N}}$ be an infinite sequence of *zero*-mean, standardized, uncorellated random variables. In mathematical form,

$$\mathbb{E}[X_i] = \int_\Omega X_i(\omega) d\mathbb{P}(\omega) = 0, \quad i \in \mathbb{N},$$

$$\mathbb{E}[X_i X_j] = \int_\Omega X_i(\omega) X_j(\omega) d\mathbb{P}(\omega) = \delta_{ij}, \quad i,j \in \mathbb{N}.$$

A random variable X_i, for $\lambda_i \neq 0$, is defined as

$$X_i := \frac{1}{\sqrt{\lambda_i}} \int_{\mathcal{D}} \alpha(\mathbf{z},\cdot)\phi_i(\mathbf{z}) d\mathbf{z}.$$

By solving the Fredholm integral problem in Eq. (3), the K-L expansion for a random field can be written as the infinite series [1,2]

$$\alpha(\mathbf{z},\cdot) \sim \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(\mathbf{z}) X_i, \tag{4}$$

which includes deterministic functions $\phi_i(\mathbf{z})$ with spatial arguments $\mathbf{z} = (z_1,\ldots,z_d)$ and uncorrelated random variables X_i. This expansion is convergent in mean-square to the correct limit and is widely used in various fields of engineering and applied sciences [36,38]. When truncated, it represents a continuous-parameter random field in terms of a tractable number of random variables. The next section encompasses the approximation of the K-L expansion.

2.4 K-L Approximation

In practice, the K-L expansion must be truncated, which admits an approximation for the random field discretization. Although there are many ways to truncate the expansion, one simple approach would be to arrange the eigenvalues in a descending order, followed by retaining the terms associated with the top $N \in \mathbb{N}$ eigenvalues. This approach takes advantage of the fact that the eigenvalues have zero as their only point of accumulation, as stated in Section 2.2. Thus, the truncated expansion is

$$\alpha_N(\mathbf{z},\cdot) = \sum_{i=1}^{N} \sqrt{\lambda_i}\phi_i(\mathbf{z}) X_i, \tag{5}$$

where the statistical variation of the original random field $\alpha(\mathbf{z},\cdot)$ is swapped with that estimated by N uncorrelated random variables, $X_1, X_2, \ldots, X_N$, and their associated eigenpairs $\{\lambda_i, \phi_i(\mathbf{z})\}_{i=1,\ldots,N}$. The truncated expansion in Eq. (5) has an error-minimizing property [39].

The quality of the approximation in Eq. (5) is judged based on N. For a given N, the effectiveness of an approximation hinges on how fast the eigenvalues decay. The rate of decay in the eigenvalues depends on the covariance function $\Gamma(\mathbf{z},\mathbf{z}')$ of the random field in Eq. (3). The larger the N, the more accurate the random field discretization becomes. On the other hand, a large N typically mandates encountering more complex subsequent UQ analyses, as solving such problems generally demands significant computational resources. Despite the intensive research on developing techniques for efficiently solving high-dimensional stochastic problems [40,41], the topic is still a burning issue. Thus, avoiding such complexity, if possible, is deemed wiser. Consequently, N must be selected judiciously, if not optimally.

Note that many details of the K-L expansion have been suppressed from this paper to avoid redundancy. The readers are advised to consult the existing literature for proofs, error analysis, and other theoretical aspects of the expansion [1–5].

3. ISOGEOMETRIC ANALYSIS

IGA was originally introduced to bridge the gap between CAD and FEA by using identical basis functions, such as B-splines, NURBS, and T-splines, in geometrical modeling and computational analysis. In many engineering applications, the geometrical domains of interest have complex shapes, thus limiting the effectiveness of a single-patch IGA. Indeed, for representing such geometrically complex shapes, a multipatch description by partitioning the given geometry into several pieces, referred to as "patches," is necessary. The multipatch modeling is also motivated by the requirements of local mesh refinement and complex topology, e.g., multiply connected domains, among others. Unfortunately, the formulation of a multipatch IGA is relatively more complicated, depending on how the patches are arranged and subsequently coupled. The arrangement may range from unstructured, arbitrarily shaped patches on one end to highly structured conforming patches on the other end. Due to the regularity, the latter extreme of conforming patches, if they are simply connected and topologically equivalent to rectangles, is most beneficial in applications. This is chiefly because the multivariate basis functions for each patch can be derived from tensor products of univariate basis functions, for instance, when using B-splines. However, this type of partition is also complicated because of the strict restrictions involved. Hence, research is ongoing on this very topic [13,42].

While IGA is heavily studied for solving BVPs, such as those encountered in computational-mechanics problems, here, the prefix "iso" refers to employing identical basis functions in geometrical modeling and solving the Fredholm integral eigenvalue problem, where the latter is mandated by the K-L expansion of a random field. So far, the IGA formulation for solving the aforementioned eigenvalue problem is available for only tensor-product, single-patch domains [31,32]. To the best knowledge of the authors, no IGA formulations exist for random field discretization on arbitrary multipatch domains. An IGA that accounts for multiple patches is not just a mere extension of an analysis that handles a single patch. Indeed, allowing multiple patches brings new technical questions, prominently the ones related to the treatment of the interfaces between adjacent patches and local mesh refinement. For the remainder of this section, NURBS and T-splines are discussed as tools for solving the Fredholm integral eigenvalue problem on multipatch domains. For additional details on NURBS and T-splines, readers are encouraged to start with the works of Cottrell et al. [11] and Bazilevs et al. [15], respectively.

3.1 Multiple Patches

In most practical applications, an IGA for a physical object with a complex geometry requires decomposition into multiple patches. Given a closed bounded domain $\mathcal{D} \subset \mathbb{R}^d$ of the physical object, let $n_p \in \mathbb{N}$ be the total number of patches with open bounded domains $\mathcal{D}_q \subseteq \mathcal{D}$, $q = 1, \ldots, n_p$, such that

$$\bar{\mathcal{D}} = \mathcal{D} = \bigcup_{q=1}^{n_p} \bar{\mathcal{D}}_q,$$

where the overline symbolizes set closure. If two distinct patches q and q', $q \neq q'$ with respective domains $\mathcal{D}_q$ and $\mathcal{D}_{q'}$ are adjacent, then they may overlap only at their interface $\Gamma_{qq'}$ (say); that is,

$$\bar{\mathcal{D}}_q \cap \bar{\mathcal{D}}_{q'} = \Gamma_{qq'} \neq \emptyset.$$

Figure 1(a) shows a schematic decomposition of a two-dimensional four-hole bracket, which consists of eight distinct patches with domains $\mathcal{D}_1$ through $\mathcal{D}_8$ and eight interfaces Γ_{12}, Γ_{13}, Γ_{24}, Γ_{35}, Γ_{46}, Γ_{57}, Γ_{68}, and Γ_{78}. The interfaces are illustrated by cyan lines. The geometry and solution of interest for each of these patches can be described by NURBS or T-splines alone, as commonly done for a single-patch domain. Here, a patch is simply an image, obtained through a NURBS/T-splines mapping, of the two-dimensional rectangle presented in Fig. 1(b). In other words, a patch in the physical space is topologically equivalent to a rectangle, so that the geometry of a patch can be produced by enforcing a certain deformation to the rectangle in the parametric space. Consequently, a complex domain with holes or discontinuities is easily reconstructed by the union of such multiple patches. In general, multiple patches are often required for a domain endowed with a topology fundamentally different from a rectangle ($d = 2$) or a cube ($d = 3$).

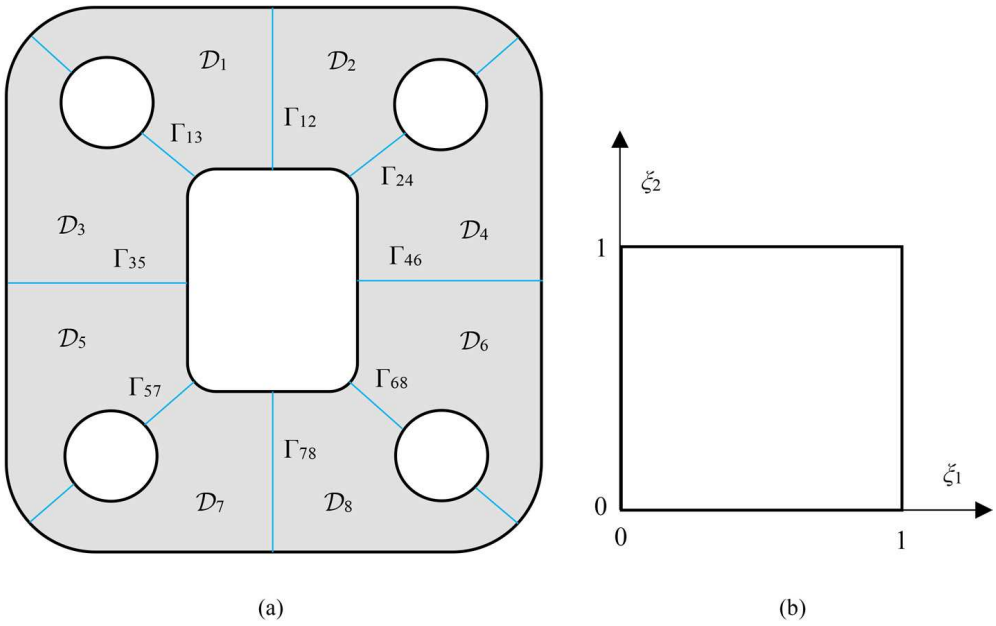

FIG. 1: A multipatch decomposition of a four-hole bracket: (a) an eight-patch domain; (b) parametric domain for an arbitrary patch

Note that in this work, a multipatch domain is specifically a region that is partitioned into a finite number of single patches. The parametric domain $[0,1]^d$ is mapped to each of these patches, which are linked together by respecting C^0-continuity between them. All patches are conforming in the sense that the individual solutions from any two adjacent patches match at their interface. As a result, the C^0-continuity on global solutions is guaranteed across all interfaces for the entire domain. This can be accomplished by at least two approaches: (1) a Lagrange multiplier approach where equations representing appropriate equality constraints are supplemented; or (2) an elimination approach where the global degrees of freedom are reduced by enforcing identical spline parametrization of an interface between any two adjacent patches. The second approach, in conjunction with NURBS and T-splines, was adopted in this work.

3.2 Analysis with NURBS

A brief introduction to B-splines for a parametric domain $\hat{D} = [0,1]^d$ and coordinate $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_d) \in \hat{D}$, is provided in Appendix A. This section focuses on NURBS basis functions for an arbitrary patch of a general multipatch domain and describes how the adjacent patches are linked in a C^0-continuous manner at the patch interfaces.

3.2.1 NURBS for an Arbitrary Single Patch

For IGA to be able to model a wider range of complex shapes, the NURBS functions have been proposed as generalization of B-splines. Associated with an arbitrary patch q with domain $\bar{\mathcal{D}}_q \subseteq \mathcal{D} \subset \mathbb{R}^d$ and coordinate direction k, where $k = 1, \ldots, d$, let $n_k^q \in \mathbb{N}$ be the total number of univariate basis functions, $p_k^q \in \mathbb{N}_0$ the polynomial degree, and $\boldsymbol{\xi}_k^q$ the knot vector. Define two multi-indices $\mathbf{i}_q := (i_1^q, \ldots, i_d^q) \in \mathbb{N}^d$ and $\mathbf{p}_q := (p_1^q, \ldots, p_d^q) \in {\mathbb{N}_0}^d$, as well as the collection $\boldsymbol{\Xi}_q := (\boldsymbol{\xi}_1^q, \ldots, \boldsymbol{\xi}_d^q)$ of all d knot vectors in $\hat{D}$. For the first multi-index, denote by

$$\mathcal{I}_q := \{\mathbf{i}_q = (i_1^q, \ldots, i_d^q) : 1 \leq i_k^q \leq n_k^q,\ k = 1, \ldots, d\} \subset \mathbb{N}^d,$$

a multi-index set, which has cardinality $|\mathcal{I}_q|$. Denote by n_c^q the total number of multivariate basis functions or the total number of control points associated with patch q. Then, from the tensor-product construction of multivariate NURBS functions, $n_c^q = |\mathcal{I}_q|$.

In reference to Appendix A, denote by $\{\mathrm{B}_{\mathbf{i}_q,\mathbf{p}_q,\boldsymbol{\Xi}_q}(\boldsymbol{\xi})\}_{\mathbf{i}_q\in\mathcal{I}_q}$ the set of n_c^q number of multivariate B-spline functions for patch q. With an arbitrary order of choice, arrange its elements by

$$\big\{\mathrm{B}_{\mathbf{i}_q,\mathbf{p}_q,\boldsymbol{\Xi}_q}(\boldsymbol{\xi})\big\}_{\mathbf{i}_q\in\mathcal{I}_q}=\Big\{\mathrm{B}_1^q(\boldsymbol{\xi}),\ldots,\mathrm{B}_{n_c^q}^q(\boldsymbol{\xi})\Big\},$$

such that $\mathrm{B}_i^q(\boldsymbol{\xi})$, $i=1,\ldots,n_c^q$ represents the ith multivariate B-spline basis function of the aforementioned set. In this way, the same B-spline basis functions in the set can be indexed with a single integer i. Obviously, $\mathrm{B}_i^q(\boldsymbol{\xi})$ also depends on $\mathbf{p}_q$ and $\boldsymbol{\Xi}_q$, but the two latter symbols are suppressed for brevity.

Associated with each $i=1,\ldots,n_c^q$, denote by $w_i^q\in\mathbb{R}^+$ a constant positive weight. As a result, the weight function $w_q:\hat{\mathcal{D}}\to\mathbb{R}$ is defined as

$$w_q(\boldsymbol{\xi}):=\sum_{i=1}^{n_c^q}w_i^q\mathrm{B}_i^q(\boldsymbol{\xi}).$$

Thereafter, the multivariate NURBS function $\mathrm{R}_i^q:\hat{\mathcal{D}}\to\mathbb{R}$ for the qth patch is defined as [43,44]

$$\mathrm{R}_i^q(\boldsymbol{\xi}):=\frac{w_i^q\mathrm{B}_i^q(\boldsymbol{\xi})}{w_q(\boldsymbol{\xi})}=\frac{w_i^q\mathrm{B}_i^q(\boldsymbol{\xi})}{\displaystyle\sum_{j=1}^{n_c^q}w_j^q\mathrm{B}_j^q(\boldsymbol{\xi})},$$

yielding the NURBS function space

$$\mathcal{R}_h^q:=\mathrm{span}\{\mathrm{R}_i^q(\boldsymbol{\xi})\}_{i=1,\ldots,n_c^q},$$

over the domain $\bar{\mathcal{D}}_q$ of patch $q,\ q=1,\ldots,n_p$. Similar to the B-spline functions, the NURBS function $\mathrm{R}_i^q(\boldsymbol{\xi})$ also depends on $\mathbf{p}_q$ and $\boldsymbol{\Xi}_q$, but, again, their symbols are excluded for conciseness. The NURBS functions inherit all of the important properties of B-splines [11]: (1) they constitute a partition of unity; (2) they have the same continuity and support; (3) they possess the property of affine transformations; (4) setting all weights to be equal, a NURBS function reduces to a scaled B-spline function; and (5) the NURBS surfaces and solids are projective transformations of tensor-product, piecewise polynomial entities.

For $i=1,\ldots,n_c^q$, let $\mathbf{c}_i^q\in\mathbb{R}^d$ be a control point corresponding to the physical domain $\bar{\mathcal{D}}_q$. Thus, using NURBS, the physical domain $\bar{\mathcal{D}}_q\subset\mathbb{R}^d$ is readily available through a geometrical mapping $\mathrm{H}_q:\hat{\mathcal{D}}_q\to\bar{\mathcal{D}}_q\subset\mathbb{R}^d$, defined as

$$\mathrm{H}_q(\boldsymbol{\xi}):=\sum_{i=1}^{n_c^q}\mathrm{R}_i^q(\boldsymbol{\xi})\mathbf{c}_i^q,\tag{6}$$

which may refer to a line ($d=1$), surface ($d=2$), or volume ($d=3$). A similar mapping can be applied to the solution of interest. It is assumed that (6), which represents a linear projection, is invertible almost everywhere in $\bar{\mathcal{D}}_q$, leading to the NURBS function

$$\bar{\mathrm{R}}_i^q(\mathbf{z}):=\mathrm{R}_i^q(\boldsymbol{\xi})\circ\mathrm{H}_q^{-1},$$

in the physical subdomain $\bar{\mathcal{D}}_q$. Define the space of such NURBS functions as the push-forward of the NURBS space $\mathcal{R}_h^q$ in Eq. (4) by

$$\mathcal{V}_h^q:=\mathrm{span}\big\{\mathrm{R}_i^q\circ\mathrm{H}_q^{-1}\big\}_{i=1,\ldots,n_c^q}=\mathrm{span}\big\{\bar{\mathrm{R}}_i^q\big\}_{i=1,\ldots,n_c^q}.$$

In addition, using Eq. (6), the physical mesh $\mathcal{K}_h^q$ (say) of the qth patch can be viewed as the image of the parametric mesh $\mathcal{Q}_h^q$ (say), that is,

$$\mathcal{K}_h^q:=\{K=\mathrm{H}_q(Q):Q\in\mathcal{Q}_h^q\},\quad q=1,\ldots,n_p,$$

where the element K of the physical mesh is the image of the element Q of the parametric mesh.

For more details on IGA and methods of mesh refinement, readers are directed to the work by Hughes et al. [10]. This work employs simple knot insertion as the mesh refinement method. Note that the proposed formulation is based on the tensor product structure of the domain corresponding to an arbitrary patch q. A patch-conforming NURBS method will be discussed next to connect multipatch domains.

3.2.2 NURBS for Multiple Patches

When dealing with multipatch decomposition, there is a need to model accurately and conveniently each subdomain $\bar{\mathcal{D}}_q,\ q = 1,\ldots,n_p$. One approach for doing so is to divide the entire domain into a number of tensor-product subdomains (patches), as described in the prior section. The remainder of this section is devoted to a description of coupling between two adjacent patches, enabling computational analysis for the entire domain.

In the C^0-patch-conforming NURBS analysis, which is one of the selected methods in this paper due its simplicity, the compatibility of NURBS patches is ensured by implementing identical NURBS representations at the patch interfaces. The interfaces can be lines or surfaces for two-dimensional or three-dimensional models, respectively. As an example, Fig. 2 depicts a schematic of two adjacent patches in the physical domain $\mathcal{D}$. The knot lines are depicted in black. The interface, which is illustrated by a cyan line, must have identical NURBS representation on both sides. Recall that NURBS objects are represented by polynomial orders, knot vectors, control points, and weights. In Fig. 2, the control points distinct to Patch 1 and Patch 2 are shown by magenta closed squares and yellow closed circles, respectively. In this work, the control points on the patch interfaces belonging to any two adjacent patches, shown in green triangles, are shared and identical to make sure that the models are completely watertight. Mathematically, this requirement implies

$$\mathrm{H}_{q|\Gamma_{q,q'}} = \mathrm{H}_{q'|\Gamma_{q,q'}}, \quad q, q' = 1,\ldots,n_p, \quad q \neq q', \tag{7}$$

for all adjacent patches, where H_q is defined in Eq. (6). The imposed conditions at the patch interfaces, whether they are lines or surfaces, ensure C^0-continuity of any response in those particular regions. For computational purposes, the arrays corresponding to each patch are assembled into a global array before the problem is solved.

A NURBS space defined for the whole multipatch domain $\mathcal{D}$ can be imagined by collecting the basis functions $\bar{\mathrm{R}}_i^q(\mathbf{z})$ from all n_p patches. However, as these basis functions are previously defined only on individual subdomains, extended basis functions that cover the whole domain $\mathcal{D}$ are required. Henceforth, define

$$\tilde{\mathrm{R}}_i^q(\mathbf{z}) = \begin{cases} \bar{\mathrm{R}}_i^q(\mathbf{z}) & \mathbf{z} \in \bar{\mathcal{D}}_q \\ 0 & \text{otherwise} \end{cases}, \quad i = 1,\ldots,n_c^q, \tag{8}$$

as the extended functions, obtained for patch $q,\ q = 1,\ldots,n_p$.

Collect all n_c^q functions from Eq. (8) for all n_p patches, yielding the set

$$\left\{\tilde{\mathrm{R}}_i^q(\mathbf{z})\right\}_{q=1,\ldots,n_p;\ i=1,\ldots,n_c^q},$$

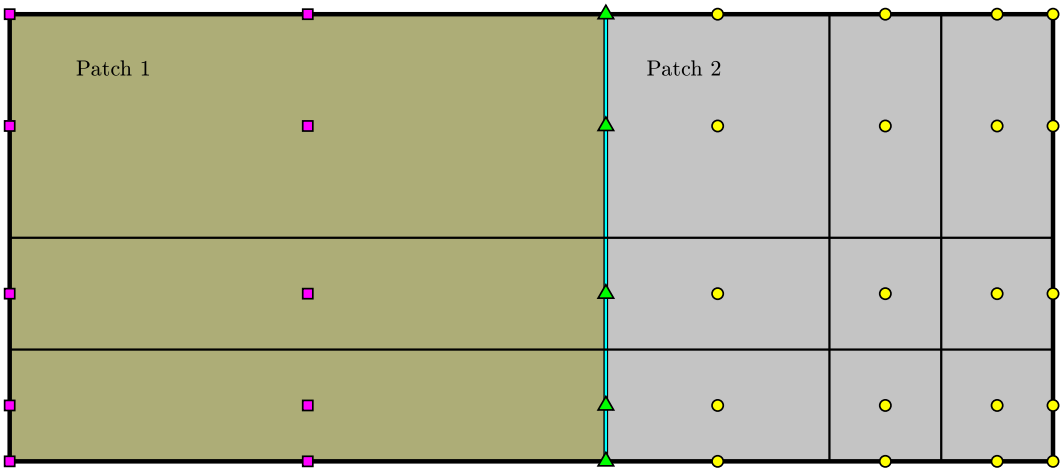

FIG. 2: Two adjacent two-dimensional NURBS patches in the physical domain

of $\sum_{q=1}^{n_p} n_c^q$ functions. But not all of these functions are linearly independent as the C^0-continuity condition in Eq. (7) has been enforced. Consequently, some of these functions have to be filtered out to obtain a reduced set,

$$\left\{\tilde{\mathrm{R}}_{k_1}(\mathbf{z}),\ldots,\tilde{\mathrm{R}}_{k_{n_t}}(\mathbf{z})\right\} \subseteq \left\{\tilde{\mathrm{R}}_i^q(\mathbf{z})\right\}_{q=1,\ldots,n_p;\ i=1,\ldots,n_c^q}, \quad 1=k_1<\ldots<k_{n_t}=n_t,$$

of extended basis functions, where

$$n_t \leq \sum_{q=1}^{n_p} n_c^q,$$

is the dimension of the NURBS space $\mathcal{V}_h$ (say) for $\mathcal{D}$. Here, n_t depends on the number and arrangement of the patch interfaces and, hence, cannot be expressed explicitly for an arbitrary domain $\mathcal{D}$. Then, the NURBS space

$$\mathcal{V}_h := \mathrm{span}\left\{\tilde{\mathrm{R}}_{k_1}(\mathbf{z}),\ldots,\tilde{\mathrm{R}}_{k_{n_t}}(\mathbf{z})\right\},\ 1=k_1<\ldots<k_{n_t}=n_t,$$

is spanned by the extended basis functions. Furthermore, the extended basis functions can be collected into an n_t-dimensional column vector as

$$\mathbf{R}(\mathbf{z}) = \left\{\begin{array}{c} \mathrm{R}_1(\mathbf{z}) \\ \vdots \\ \mathrm{R}_{n_t}(\mathbf{z}) \end{array}\right\} := \left\{\begin{array}{c} \tilde{\mathrm{R}}_{k_1}(\mathbf{z}) \\ \vdots \\ \tilde{\mathrm{R}}_{k_{n_t}}(\mathbf{z}) \end{array}\right\}, \tag{9}$$

where $\mathrm{R}_i(\mathbf{z}) := \tilde{\mathrm{R}}_{k_i}(\mathbf{z})$, $i=1,\ldots,n_t$. The vector $\mathbf{R}(\mathbf{z})$ will be used for obtaining approximate solutions of the Fredholm integral eigenvalue problem, to be discussed in Section 4.

Finally, using Eq. (6), the physical mesh $\mathcal{K}_h$ (say) for the entire domain can be obtained as

$$\mathcal{K}_h = \bigcup_{q=1}^{n_p} \mathcal{K}_h^q = \bigcup_{q=1}^{n_p} \{K = \mathrm{H}_q(Q) : Q \in \mathcal{Q}_h^q\}.$$

The described patch-conforming technique can be simply generalized for modeling multipatch three-dimensional geometries. However, two major restrictions of C^0-patch-conforming NURBS are that (1) each patch has a tensor-product structure, somewhat restricting the topology of the patch geometry, and (2) the refinement is propagated from patch to patch [10], thereby unnecessarily escalating the number of elements in the computational model. The latter issue reveals the need for local refinement in the mesh, and is a major motivation behind employing T-splines in IGA, to be discussed next.

3.3 Analysis with T-Splines

The NURBS functions, albeit being convenient in free-form surface modeling, have difficulties with modeling multipatch geometries, specifically, the ones concerning gaps and overlaps on the patch interfaces. Consequently, T-splines were introduced as a generalization of NURBS, thereby improving the meshing process through local refinement capabilities. The name "T-splines" refers to the fact that T-junctions are allowed in the so-called "T-mesh." Sederberg et al. [12] were the first to elaborate on T-splines.

The description of T-splines in this section pertains to a general physical domain. In reference to Fig. 1, the domain can be a domain $\mathcal{D}_q$ of an arbitrary single patch q or the entire domain $\mathcal{D}$ of the physical object, if the latter is simple enough to be modeled by T-splines. Therefore, there is no reference to the patch geometry in most of this section, although the coupling between two patches will be addressed at the very end.

3.3.1 T-Mesh and T-Splines

For simplicity without loss of generality, the paraphernalia of T-splines is discussed for two-dimensional domains. For given polynomial degrees p_1 and p_2 and number of control points n_c in the physical domain, let there be some anchors

$\hat{\mathbf{c}}_i$, $i = 1, \ldots, n_c$ in the parametric domain $\hat{\mathcal{D}}$. The anchors $\hat{\mathbf{c}}_i$ are associated with the control points $\mathbf{c}_i$, $i = 1, \ldots, n_c$, which control the representation of the physical domain, in a one-to-one correspondence. The appearance of T-meshes is different for odd and even $p_k,\ k = 1, 2$. Figure 3 demonstrates a T-mesh tiling in $\mathbb{R}^2$ in the physical and parametric spaces, respectively. The elements in Fig. 3(a) are illustrated by a shaded area in Fig. 3(b). Denote by n_e the total number of elements in the domain. For numerical calculations, each element e, $e = 1, \ldots, n_e$ of the T-mesh can be mapped to the parent space on $[-1, 1]^2$ for numerical integration. As observed, local refinement is conducted by introducing T-junctions. The anchors are also depicted with black closed circles. For even or odd p_k, the anchors are either centered in the tiles, or placed on the tiles' vertices, respectively.

Attached to each anchor in Fig. 3(b), there exists a single T-spline function. To define such T-spline for each and every anchor, two local knot vectors in the ξ_1 and ξ_2 coordinate directions must be defined. The initiative behind local knot vectors originates from point-based splines [12] and is the foundation for local refinement. The local knot vector in, say, the ξ_1 direction is constructed by taking an anchor i and then marching horizontally in both the left and right directions until either $(p_1/2 + 1)$ edges (for even p_1) or $(p_1 + 1)/2$ edges (for odd p_1) are crossed. Eventually, a local knot vector of length $(p_1 + 2)$ is formed. The same procedure is carried out in the ξ_2 direction to construct the corresponding local knot vector. Thus, for an anchor i, the local knot vector in the kth coordinate direction is defined as

$$\boldsymbol{\xi}_k^i = \{\xi_{k,1}^i, \xi_{k,2}^i, \ldots, \xi_{k,p_k+2}^i\}, \quad k = 1, 2, \quad \text{and} \quad i = 1, \ldots, n_c,$$

where $\xi_{k,1}^i \leq \xi_{k,2}^i \leq \ldots \leq \xi_{k,p_k+2}^i$. If there are no more edges to march towards, the most recently placed knot is repeated. As an example, take anchor A situated at $(0.75, 0.6)$ in Fig. 3(b) and let $p_1 = p_2 = 2$. The local knot vectors in the ξ_1 and ξ_2 directions are $\boldsymbol{\xi}_1^A = \{0.2, 0.5, 1, 1\}$ and $\boldsymbol{\xi}_2^A = \{0, 0.2, 1, 1\}$, respectively. This simply implies that the subdomain covered by anchor A, shown with a green rectangle in Fig. 3(b), is $[0.2, 1] \times [0, 1]$. As another example, the two local knot vectors for anchor B are $\boldsymbol{\xi}_1^B = \{0, 0, 0.2, 0.5\}$ and $\boldsymbol{\xi}_2^B = \{0, 0, 1, 1\}$, and the corresponding T-spline basis function for anchor B is nonzero on the domain $[0, 0.5] \times [0, 1]$.

The continuity reduction lines, shown in red in Fig. 3(b), are obtained by drawing all the knot lines in the local knot vectors. If a knot line is not already an existing edge, a continuity reduction line is formed. As a result, the elements are refined by these lines, as consistently indicated in Fig. 3. For more details of continuity reduction lines, refer to the work by May et al. [22].

For a general T-mesh comprising n_c anchors in d dimensions, n_c groups of local knot vectors in the ξ_1 through ξ_d directions are formed. Thus, in direction $k,\ k = 1, \ldots, d$, a T-spline corresponding to anchor i is defined, predictably, in a recursive manner. The *zero*-order T-spline is defined by

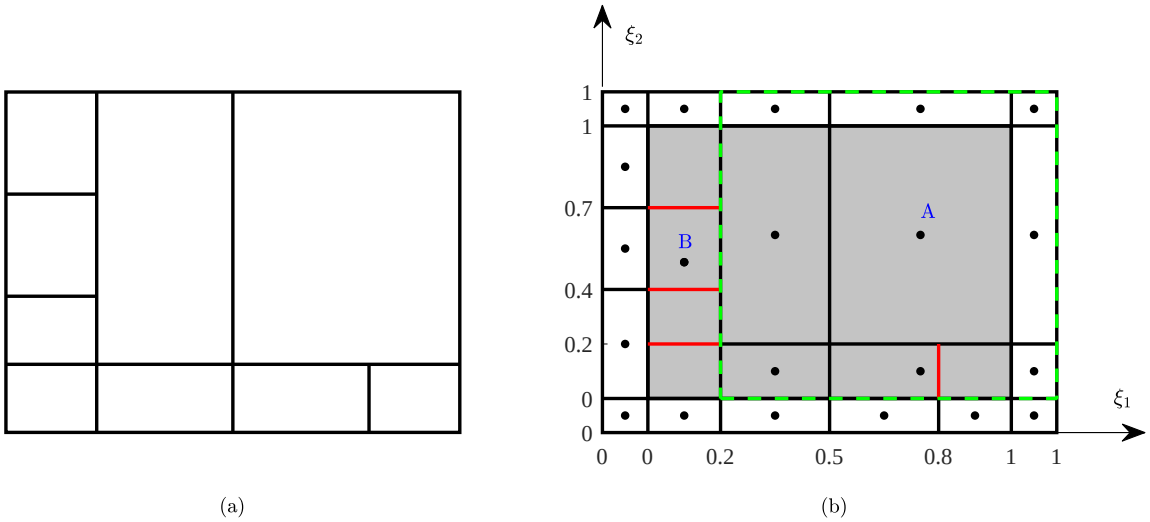

FIG. 3: A T-mesh tiling in two dimensions: (a) physical domain; (b) parametric domain

$$U^{i,k}_{i_k,0,\boldsymbol{\xi}_k}(\xi_k) = \begin{cases} 1, & \xi^i_{k,i_k} \le \xi_k < \xi^i_{k,i_k+1}, \\ 0, & \text{otherwise} \end{cases} \quad i_k = 1, \dots, p_k + 1. \tag{10}$$

Thus, for $p_k \geq 1$, the higher-order T-splines are constructed by the Cox-de Boor formula [45] as

$$U^{i,k}_{i_k,p_k,\boldsymbol{\xi}_k}(\xi_k) = \frac{\xi_k - \xi^i_{k,i_k}}{\xi^i_{k,i_k+p_k} - \xi^i_{k,i_k}} U^{i,k}_{i_k,p_k-1,\boldsymbol{\xi}_k}(\xi_k) + \frac{\xi^i_{k,i_k+p_k+1} - \xi_k}{\xi^i_{k,i_k+p_k+1} - \xi^i_{k,i_k+1}} U^{i,k}_{i_k+1,p_k-1,\boldsymbol{\xi}_k}(\xi_k). \tag{11}$$

The recursive computation of T-splines for anchor $i,\ i = 1, \dots, n_c$, and coordinate direction $k,\ k = 1, \dots, d$, by Eq. (11) yields a single T-spline of degree p_k, denoted by $U^{i,k}_{1,p_k,\boldsymbol{\xi}_k}(\xi_k)$.

The multivariate T-spline functions with the vector argument $\boldsymbol{\xi} = (\xi_1, \dots, \xi_d)$ are obtained for each anchor i, $i = 1, \dots, n_c$ by the tensor product of its corresponding univariate T-splines, introduced in Eqs. (10) and (11). Given $\mathbf{p} := (p_1, \dots, p_d) \in {\mathbb{N}_0}^d$ and $\boldsymbol{\Xi}^i := (\boldsymbol{\xi}^i_1, \dots, \boldsymbol{\xi}^i_d)$ for anchor i, the multivariate T-spline function $\mathrm{U}^i_{\mathbf{p},\boldsymbol{\Xi}^i} : \hat{\mathcal{D}} \to \mathbb{R}$ is defined as

$$\mathrm{U}^i_{\mathbf{p},\boldsymbol{\Xi}^i}(\boldsymbol{\xi}) := \prod_{k=1}^{d} U^{i,k}_{1,p_k,\boldsymbol{\xi}_k}(\xi_k). \tag{12}$$

It is important to recognize that the tensor-product construction of multivariate T-splines is performed individually for each anchor, whereas a multivariate NURBS is obtained from tensor-product development collectively over all elements. As a result, T-splines afford a greater flexibility in local mesh refinement with fewer elements than NURBS.

To provide the T-splines with the capability of modeling even more complex geometries, such as circles, ellipses, cylinders, spheres, ellipsoids, and tori, the multivariate T-splines in Eq. (12) are rationalized in a similar way as carried out for B-splines in Section 3.2.1. Therefore, with appropriately defining the weights $w_i \in \mathbb{R}^+,\ i = 1, \dots, n_c$, multivariate rational T-splines $\mathrm{T}^i_{\mathbf{p},\boldsymbol{\Xi}^i} : \hat{\mathcal{D}} \to \mathbb{R}$ are defined as

$$\mathrm{T}^i_{\mathbf{p},\boldsymbol{\Xi}^i}(\boldsymbol{\xi}) := \frac{w_i \mathrm{U}^i_{\mathbf{p},\boldsymbol{\Xi}^i}(\boldsymbol{\xi})}{\sum_{j=1}^{n_c} w_j \mathrm{U}^j_{\mathbf{p},\boldsymbol{\Xi}^j}(\boldsymbol{\xi})}, \tag{13}$$

which also satisfy the partition of unity property, among others. Note that rational T-splines are also simply referred to as T-splines [15]. Assign a control point $\mathbf{c}_i$ to each T-spline and represent a line ($d = 1$), surface ($d = 2$), or volume ($d = 3$) by the linear projection $\tilde{\mathrm{H}}(\boldsymbol{\xi})$, defined as

$$\tilde{\mathrm{H}}(\boldsymbol{\xi}) := \sum_{i=1}^{n_c} \mathrm{T}^i_{\mathbf{p},\boldsymbol{\Xi}^i}(\boldsymbol{\xi}) \mathbf{c}_i. \tag{14}$$

Hence,

$$\bar{\mathrm{T}}^i_{\mathbf{p},\boldsymbol{\Xi}^i}(\mathbf{z}) := \mathrm{T}^i_{\mathbf{p},\boldsymbol{\Xi}^i}(\boldsymbol{\xi}) \circ \tilde{\mathrm{H}}^{-1}, \tag{15}$$

becomes the rational T-spline for anchor $i,\ i = 1, \dots, n_c$ in the physical domain.

Note that not every set of T-splines is suitable for analysis, as they may be linearly dependent [18]. For the analysis to be stable, judicious tiling of the T-mesh is crucial so that the underlying T-splines are linearly independent. There are mathematical and topological ways to determine whether a T-mesh is analysis-suitable or not [19,22]. This will be addressed further in Section 3.3.3.

3.3.2 Bézier Extraction Operator

The Bézier extraction operator is a means for storing IGA data in a structured framework [21]. Moreover, local refinement may be conveniently carried out using the Bézier extraction operator. Consider a T-mesh similar to the one depicted in Fig. 3(b). Each anchor i, $i = 1, \dots, n_c$, has a T-spline function attached to it that is nonzero on a region in the domain. The idea behind Bézier extraction is to express the T-splines on those particular regions in terms

of Bernstein polynomials. Given $p_k,\ k=1,\dots,d$, define (p_k+1) univariate Bernstein basis polynomials with the argument $\eta_k \in [-1,1]$ by

$$\tilde{V}_k^j(\eta_k) = \frac{1}{2^{p_k}}\binom{p_k}{j-1}(1-\eta_k)^{p_k-(j-1)}(1+\eta_k)^{j-1}, \quad j=1,\dots,p_k+1. \tag{16}$$

With the linear mapping $\xi_k : [-1,1] \to [0,1]$, $k=1,\dots,d$, denote by $V_k^{j,e}(\xi_k)$ the Bernstein polynomial of degree $(j-1)$ in coordinate direction k on element e, $e=1,\dots,n_e$. There is a total of

$$n_f = \prod_{k=1}^{d}(p_k+1),$$

d-variate Bernstein polynomials that are nonzero on element e. Collect such multivariate functions in an n_f-dimensional set as

$$\{\mathrm{V}_j^e(\boldsymbol{\xi})\}_{j=1,\dots,n_f} := \left\{ \prod_{k=1}^{d} V_k^{j_k,e}(\xi_k) \right\}_{j_k=1,\dots,p_k+1,\ k=1,\dots,d}, \tag{17}$$

where $j_k = 1,\dots,p_k+1,\ k=1,\dots,d$. With an arbitrary order of choice, arrange the elements of the set in Eq. (17) in an n_f-dimensional column vector $\mathbf{V}^e(\xi)$ as

$$\mathbf{V}^e(\boldsymbol{\xi}) = \begin{Bmatrix} \mathrm{V}_1^e(\boldsymbol{\xi}) \\ \vdots \\ \mathrm{V}_{n_f}^e(\boldsymbol{\xi}) \end{Bmatrix}.$$

Recall that $\mathrm{U}^i_{\mathbf{p},\boldsymbol{\Xi}^i}(\boldsymbol{\xi})$ is a d-variate T-spline function corresponding to anchor i in $\hat{\mathcal{D}}$. A chunk of this function over element e, denoted by $\mathrm{U}^{i,e}_{\mathbf{p},\boldsymbol{\Xi}^i}(\boldsymbol{\xi})$, can be obtained from

$$\mathrm{U}^{i,e}_{\mathbf{p},\boldsymbol{\Xi}^i}(\boldsymbol{\xi}) = \mathbf{C}^{i,e\intercal}\mathbf{V}^e(\boldsymbol{\xi}), \quad i=1=1,\dots,n_c, \quad e=1,\dots,n_e, \tag{18}$$

where $\mathbf{C}^{i,e}$ is an n_f-dimensional column vector of appropriate coefficients. In other words, the d-variate T-splines on every element are expressed by linear combinations of the Bernstein polynomials. This argument is valid, since the Bernstein polynomials proposed in Eq. (16) form a basis over $[-1,1]$. The process in Eq. (18) is referred to as elementwise Bézier extraction.

Assemble the Bézier extraction contributions of every element into the $n_f \times n_e$ matrices,

$$\mathbf{C}^i = \big[\mathbf{C}^{i,1},\dots,\mathbf{C}^{i,n_e}\big] \ \text{ and } \ \mathbf{V}(\boldsymbol{\xi}) = \big[\mathbf{V}^1(\boldsymbol{\xi}),\dots,\mathbf{V}^{n_e}(\boldsymbol{\xi})\big],$$

and denote by

$$\mathbf{U}^i(\boldsymbol{\xi}) = \mathbf{C}^i\mathbf{V}^\intercal(\boldsymbol{\xi}),$$

the $n_f \times n_f$ matrix of T-splines corresponding to anchor i. The Bézier extraction operator $\mathbf{C}$ as an $n_f \times n_c n_e$ matrix can thus be formed by collecting the contributions of all anchors as

$$\mathbf{C} = \big[\mathbf{C}^1,\dots,\mathbf{C}^{n_c}\big],$$

which admits the linear transformation

$$\mathbf{U}(\boldsymbol{\xi}) = \mathbf{C}\mathbf{V}_t^\intercal(\boldsymbol{\xi}), \tag{19}$$

where

$$\mathbf{U}(\boldsymbol{\xi}) = \big[\mathbf{U}^i(\boldsymbol{\xi}),\dots,\mathbf{U}^{n_c}(\boldsymbol{\xi})\big] \text{ and } \mathbf{V}_t(\boldsymbol{\xi}) = \big[\underbrace{\mathbf{V}(\boldsymbol{\xi}),\dots,\mathbf{V}(\boldsymbol{\xi})}_{n_c \text{ times}}\big],$$

are $n_f \times n_c n_f$ and $n_f \times n_c n_e$ matrices, respectively. Matrix $\mathbf{C}$ in Eq. (19) is referred to as the Bèzier extraction operator and is a key matrix in conveniently representing the T-splines corresponding to a T-mesh. This matrix is helpful for assessing the linear independence of the T-splines as well as local mesh refinement [22]. This work uses the Bèzier extraction operator for the refinement of analysis-suitable T-meshes.

3.3.3 Analysis-Suitable T-Splines

The T-splines introduced in Eq. (13) form a basis for analysis, if they are linearly independent. Buffa et al. [18] were the first to present particular bicubic T-meshes for which the T-splines were not linearly independent. Consequently, analysis-suitable T-splines were officially defined [19].

One topological approach to assess the analysis-suitability of a T-mesh, and consequently the linear independence of its corresponding T-splines, is based on T-node extensions. A T-node is a T-junction inside the parametric domain. There generally are many T-nodes in a T-mesh. See Fig. 4 as an example. From every T-node marked with a closed green circles in Fig. 4, two extension lines are emitted, namely, face extension and edge extension [17], where the former is emitted in the direction where there is no edge, and the latter is drawn in the opposite direction. The two types of extension lines are illustrated by blue and magenta lines, respectively. A T-mesh is called analysis-suitable when no T-node extensions intersect. Examples of non-analysis-suitable and analysis-suitable T-meshes are presented in Figs. 4(a) and 4(b), respectively. The intersections are marked with red crosses. Analysis-suitable T-splines are those in correspondence with analysis-suitable T-meshes.

In order to make a T-mesh analysis-suitable, one must add or remove edges accordingly and check for the topological constraints to be satisfied. Research is ongoing on mesh refinement algorithms, where the T-mesh is efficiently maintained analysis-suitable. One approach is to use different discretization layers, which is referred to as hierarchical T-mesh [23]. While there is no knowledge on where the elements need to be refined *a priori*, adaptive refinement methods have also been proposed [24]. For detailed topological descriptions on analysis-suitable T-splines and complete mathematical proof, refer to the existing literature [17,19].

3.3.4 T-Mesh Refinement with Bèzier Extraction Operator

T-splines of arbitrary degree can be refined by using the Bézier extraction operator. Imagine a T-mesh $\mathcal{T}_R$ as a refinement of a coarser T-mesh $\mathcal{T}_0$. In mathematical form, $\mathcal{T}_0 \subseteq \mathcal{T}_R$. Denote by $\mathbf{U}_0(\boldsymbol{\xi})$ and $\mathbf{U}_R(\boldsymbol{\xi})$ the T-splines for $\mathcal{T}_0$ and $\mathcal{T}_R$, respectively, as defined in Eq. (19). Furthermore, denote by $n_{c,0}$ and $n_{c,R}$ the numbers of control points associated with $\mathcal{T}_0$ and $\mathcal{T}_R$, respectively. Note that the representation of the surface or the volume in the physical space should not change by refining the initial T-mesh. Therefore, write

$$\mathbf{U}_0(\boldsymbol{\xi}) = \mathbf{T}\mathbf{U}_R(\boldsymbol{\xi}), \tag{20}$$

to express the T-splines of the coarse mesh $\mathcal{T}_0$ in terms of those in the refined mesh $\mathcal{T}_R$ via the transformation matrix $\mathbf{T}$, which is of dimension $n_{c,0} \times n_{c,R}$ and is referred to as the refinement matrix. By substituting Eq. (19) in Eq. (20),

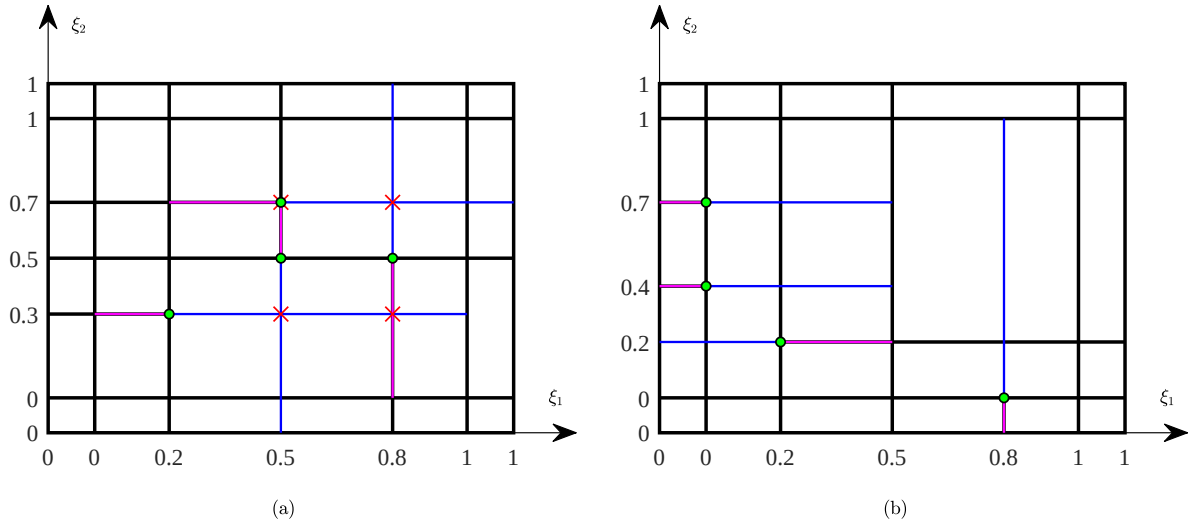

FIG. 4: Two T-meshes: (a) non-analysis-suitable T-mesh; (b) analysis-suitable T-mesh

$$\mathbf{C}_0 = \mathbf{T}\mathbf{C}_R,$$

where $\mathbf{C}_0$ and $\mathbf{C}_R$ are the Bèzier extraction operators associated with the meshes $\mathcal{T}_0$ and $\mathcal{T}_R$, respectively. For each anchor i, $i = 1, \ldots, n_{c,0}$, algebraic matrix manipulations yield

$$\mathbf{C}_0^i = \mathbf{C}_R^\intercal \mathbf{T}^i, \tag{21}$$

with $\mathbf{T}^i$ representing the ith contribution to $\mathbf{T}$. On assembly, the matrix

$$\mathbf{T} = [\mathbf{T}^1, \ldots, \mathbf{T}^{n_{c,0}}]^\intercal,$$

is formed to be implemented in Eq. (20).

As the refinement matrix $\mathbf{T}$ is used for local mesh refinement, the weighted control points $\mathbf{c}_R^w := \left[\mathbf{c}_{R,w}^1, \ldots, \mathbf{c}_{R,w}^{n_{c,R}}\right]^\intercal$ for $\mathcal{T}_R$ can be obtained from the weighted control points $\mathbf{c}_0^w := \left[\mathbf{c}_{0,w}^1, \ldots, \mathbf{c}_{0,w}^{n_{c,0}}\right]^\intercal$ by the projection [22]

$$\mathbf{c}_R^w = \mathbf{T}^\intercal \mathbf{c}_0^w, \tag{22}$$

where

$$\mathbf{c}_{0,w}^i := \{w_i c_1^i, \ldots, w_i c_d^i\}, \quad i = 1, \ldots, n_{c,0}.$$

Note that c_d^i is the dth coordinate of the ith control point, and w_i is its associated weight.

One requirement for refinement by the Bèzier extraction operator is that the T-splines corresponding to the coarse mesh must be writable in terms of those related to the fine mesh. In other words, the two sets of basis functions must be nested. Otherwise, there is no unique solution for $\mathbf{T}^i$ in Eq. (21).

3.3.5 Coupling between T-Mesh Patches

As stated previously, the parametric domain modeled with a T-mesh discussed so far can be mapped to individual patches separately or to the entire domain of the physical object. If the domain is relatively simple, there is no need for multiple patches, as one single T-mesh can model the entire geometry. However, for a relatively complex domain, multiple patches may be required even for a T-mesh. In this case, one must be able to couple two adjacent T-meshes. This is accomplished as follows.

In Figs. 5(a) and 5(b), two analysis-suitable T-meshes are proposed for the two patches introduced in Section 3.2.2. These two T-meshes are coupled in Fig. 5(c) in such a way as to yield C^0-continuity at the patch interface. In the coupling region illustrated by a cyan rectangle, there must be common edges in the two adjacent T-meshes to enforce C^0-continuity conditions. The implementation of T-splines and conditions to retain analysis-suitability ensure that the refinement does not necessarily propagate from one patch to another, in contrast to what occurred in Fig. 2 for NURBS patches, thereby controlling the number of elements in the computational analysis more freely. Figure 5(d) demonstrates the analysis-suitable IGA model with T-splines for the two-patch domain, where the interface is marked with a cyan line. Refer to the work by Bazilevs et al. [15] for more details.

Based on the descriptions so far, one concludes that T-splines exhibit more flexibility than NURBS in modeling complex geometries, generally by dividing them into a finite number of simple patches. Although the C^0-continuity at the patch interfaces may be adequate for many practical applications, a higher-order continuity between two adjacent patches is generally more straightforward to enforce by T-splines than by NURBS. Having said this, no higher-order continuity was implemented in this work.

4. PROJECTION METHODS

As physical domains become more complex, numerical computations are increasingly required. This section entails Galerkin and collocation projection methods to solve the Fredholm integral eigenvalue problem defined by Eq. (3) on arbitrary multipatch domains. By obtaining the eigensolutions, the K-L expansion in Eq. (5) is at hand. To solve the problem numerically, NURBS and rational T-splines are employed. To avoid redundancy, generic basis functions will be used in the formulations, which shall be substituted by NURBS or T-splines as needed. To the best knowledge of the authors, the K-L expansion using IGA for multipatch domains is being reported here for the first time.

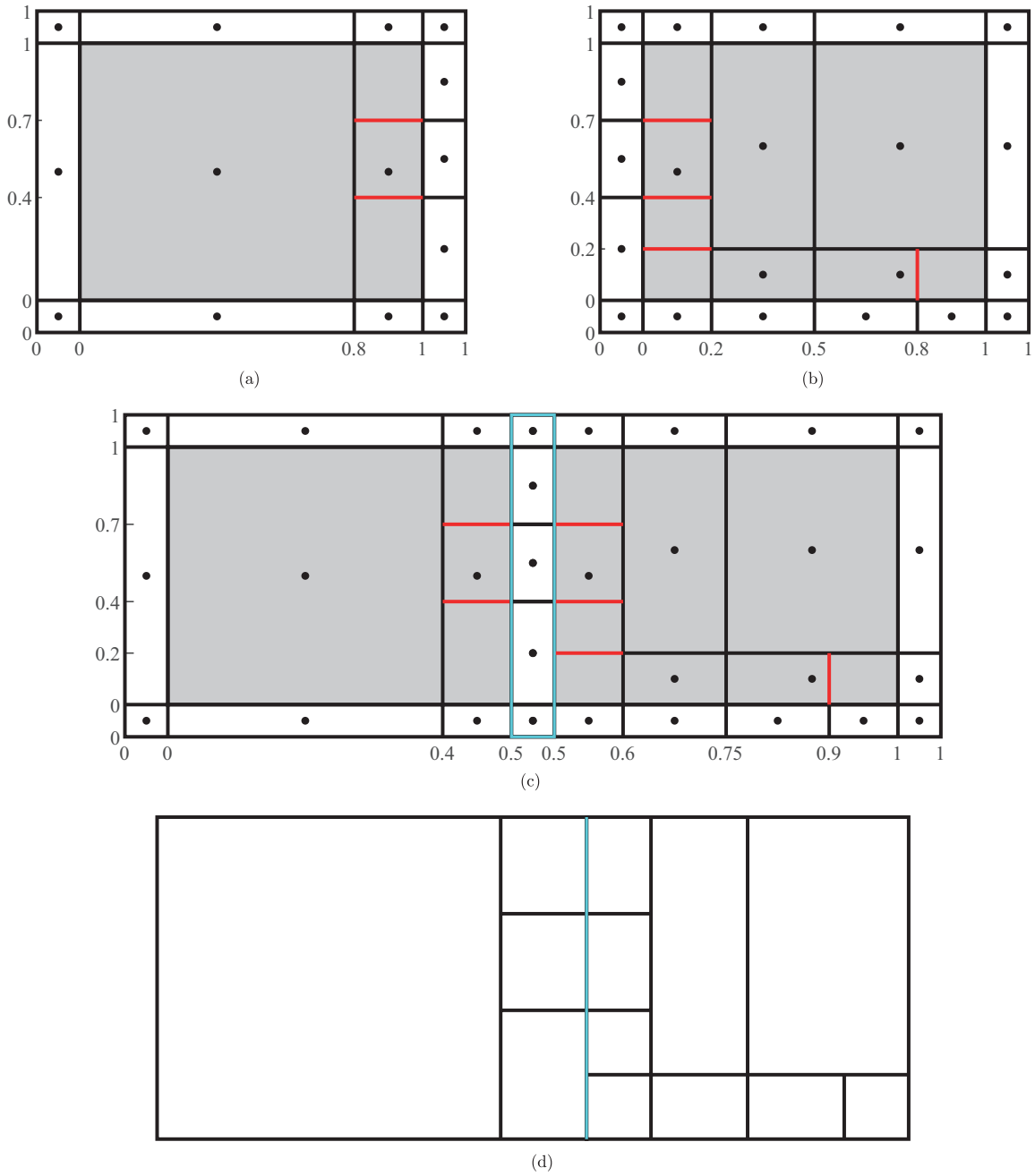

FIG. 5: An analysis-suitable T-mesh for a two-patch domain: (a) T-mesh for Patch 1 in parameter space; (b) T-mesh for Patch 2 in parameter space; (c) C^0-continuous patch coupling; (d) finite element model in physical domain

4.1 Galerkin Projection Method

The Fredholm integral eigenvalue problem, when transformed to its weak form by the Galerkin approach, can be discretized, resulting in the linear matrix eigenvalue problem as [31]

$$\mathbf{A}\mathbf{f}_h = \lambda_h \mathbf{B}\mathbf{f}_h, \tag{23}$$

where λ_h and $\mathbf{f}_h$ are the approximate eigenvalues and eigenvectors. Here, the system matrices $\mathbf{A}$ and $\mathbf{B}$ have their components

$$A_{ij} := \int_{\mathcal{D}} \int_{\mathcal{D}} \Gamma(\mathbf{z}, \mathbf{z}') \mathrm{G}_i(\mathbf{z}) \mathrm{G}_j(\mathbf{z}') d\mathbf{z} d\mathbf{z}', \quad i, j = 1, \ldots, n_b, \tag{24}$$

and

$$B_{ij} := \int_{\mathcal{D}} \mathrm{G}_i(\mathbf{z}) \mathrm{G}_j(\mathbf{z}) d\mathbf{z}, \quad i, j = 1, \ldots, n_b, \tag{25}$$

where $\mathrm{G}_i(\mathbf{z})$, $i = 1, \ldots, n_b$ are $n_b \in \mathbb{N}$ generic basis functions, spanning the finite-dimensional approximation space of eigenfunctions on $\mathcal{D}$. A specific choice of these basis functions depends on whether NURBS or rational T-splines are used for IGA. Therefore, there are two cases as follows.

1. If NURBS functions are used as the basis, then $n_b = n_t$, with n_t representing the total number of control points for the entire domain, as described in Section 3.2. In this case,

   $$\mathrm{G}_i(\mathbf{z}) = \mathrm{R}_i(\mathbf{z}), \ i = 1, \ldots, n_t,$$

   where $\mathrm{R}_i(\mathbf{z})$ is the ith NURBS function in the physical domain, as defined in Eq. (9). As a result, $\mathbf{A}$ and $\mathbf{B}$ are both $n_t \times n_t$ square matrices.

2. If T-spline functions are used as the basis, then $n_b = n_c$, with n_c representing the total number of anchors for the entire domain, whether obtained for a single patch or multiple patches, as described in Section 3.2. In this case,

   $$\mathrm{G}_i(\mathbf{z}) = \bar{\mathrm{T}}_{\mathbf{p}, \boldsymbol{\Xi}^i}(\mathbf{z}), \ i = 1, \ldots, n_c,$$

   where $\bar{\mathrm{T}}_{\mathbf{p}, \boldsymbol{\Xi}^i}(\mathbf{z})$ is the ith T-spline function in the physical domain, as defined in Eq. (15). Consequently, $\mathbf{A}$ and $\mathbf{B}$ are both $n_c \times n_c$ square matrices.

By forming the system matrices, (23) is ready to be solved by a standard eigensolver to obtain the eigenvalues λ_h and eigenvectors $\mathbf{f}_h$. For both cases, the approximate eigenfunctions are interpolated by using the basis functions as

$$\phi_h(\mathbf{z}) = \sum_{i=1}^{n_b} f_{h,i} \mathrm{G}_i(\mathbf{z}), \tag{26}$$

where $f_{h,i}$ is the ith component of $\mathbf{f}_h$.

Recall that the true eigensolutions to Eq. (3) are real-valued, as stated in Section 2.2. It can be proven that in Eq. (23), the matrix $\mathbf{A}$ is symmetric and positive-semidefinite and the matrix $\mathbf{B}$ is symmetric and positive-definite. This ensures that the approximate eigenvalues λ_h and eigenfunctions $\phi_h(\mathbf{z})$ are real-valued. The proof, which is suppressed from this paper along with more details on how to construct the system matrices, is similar to the one provided by Rahman [31] for single-patch domains.

Note that the computation of A_{ij} and B_{ij} in Eqs. (24) and (25) involves two domain integrals and one domain integral, respectively. The evaluation of the former integral is burdensome, especially for large-scale three-dimensional problems. For instance, the construction of matrix $\mathbf{A}$ in a three-dimensional problem involves sixfold integration. Therefore, the Galerkin approach, albeit favorable because of its accuracy, is generally computationally intensive. Thus, the goal is to improve the efficiency by implementing the collocation projection.

4.2 Collocation Projection Method

The Fredholm integral eigenvalue problem can also be solved in its strong form by setting the residual function

$$r_h := \int_{\mathcal{D}} \Gamma(\mathbf{z}, \mathbf{z}') \phi_h(\mathbf{z}') d\mathbf{z}' - \lambda_h \phi_h(\mathbf{z}),$$

equal to zero at some select points $\mathbf{z}_i \in \mathcal{D},\ i = 1, \ldots, n_b \in \mathbb{N}$. For multipatch domains, a similar linear matrix eigenvalue problem can be formed as [32]

$$\hat{\mathbf{A}}\hat{\mathbf{f}}_h = \hat{\lambda}_h \hat{\mathbf{B}}\hat{\mathbf{f}}_h. \tag{27}$$

Here, the elements of the system matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are computed from

$$\hat{A}_{ij} := \int_{\mathcal{D}} \Gamma(\mathbf{z}_i, \mathbf{z}')\mathrm{G}_j(\mathbf{z}')d\mathbf{z}',\ i, j = 1, \ldots, n_b \tag{28}$$

and

$$\hat{B}_{ij} := \mathrm{G}_j(\mathbf{z}_i),\ i, j = 1, \ldots, n_b, \tag{29}$$

also involving n_b generic basis functions $\mathrm{G}_i(\mathbf{z})$, $i = 1, \ldots, n_b$. These basis functions, obtained from either NURBS- or T-splines-based parametrization, are the same as those defined in Section 4.1. Here, only the collocation points need to be defined. Again, there are two cases as follows.

1. If NURBS functions are used as the basis, then $n_b = n_t$, with n_t representing the total number of control points for the entire domain, as described in Section 3.2. Moreover, given the knot vectors $\boldsymbol{\Xi}^q = \{\boldsymbol{\xi}_1^q, \ldots, \boldsymbol{\xi}_d^q\}$ for patch $q,\ q = 1, \ldots, n_p$, use the Greville abscissa to obtain the collocation points in the parametric domain $\hat{\mathcal{D}}$ as

$$\bar{\xi}^q_{k,i_k} := \frac{1}{p_k^q} \sum_{l=1}^{p_k^q} \xi^q_{k,i_k+l},\ i_k = 1, \ldots, n_k^q,\ k = 1, \ldots, d, \text{ and } q = 1, \ldots, n_p,$$

   where p_k^q and ξ^q_{k,i_k+q} are the degree in the coordinate direction k of patch q and the $(i_k + l)$th knot in $\boldsymbol{\xi}_k^q$, respectively. The number of collocation points is equal to the number of control points in every patch. Thus, the collocation points are generated in the physical space by

$$\mathbf{z}_i = \mathrm{H}_q(\bar{\boldsymbol{\xi}}_i),\ \bar{\boldsymbol{\xi}}_i = (\bar{\xi}^q_{1,i_1}, \ldots, \bar{\xi}^q_{d,i_d}) \in \hat{\mathcal{D}},\ i_k = 1, \ldots, n_k,\ k = 1, \ldots, d,\ q = 1, \ldots, n_p. \tag{30}$$

   Recall that according to Section 3.2.2, some of the control points at the patch interfaces and their corresponding NURBS functions have been dropped in order to attain linearly independent basis functions that are C^0-continuous at the patch interfaces while spanning the whole domain. By using Eq. (30), coincident collocation points are also generated on the patch interfaces. As the collocation points are actually in one-to-one correspondence with the control points, the redundant collocation points at the patch interfaces must also be removed accordingly. This guarantees having full-rank system matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$. Eventually, there are n_t collocation points on the domain. Furthermore, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are $n_t \times n_t$ square matrices.

2. If T-spline functions are used as basis functions, $n_b = n_c$, with n_c representing the total number of anchors for the entire domain, as described in Section 3.3.1. In this case, project the anchors $\hat{\mathbf{c}}_i,\ i = 1, \ldots, n_c$ from $\hat{\mathcal{D}}$ to the physical domain by Eq. (14). For instance, plug in the coordinates of anchor A in Fig. 3, which is located at $(0.75, 0.6)$, in Eq. (14). Thus, the collocation points $\mathbf{z}_i,\ i = 1, \ldots, n_c$ are nothing but the projection of such anchors in the physical space. This guarantees having the same number of collocation points as control points. Therefore, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are $n_c \times n_c$ square matrices.

Similar to Eq. (26), the eigenfunctions for both cases are computed from the interpolation

$$\phi_h(\mathbf{z}) = \sum_{i=1}^{n_b} \hat{f}_{h,i}\mathrm{G}_i(\mathbf{z}), \tag{31}$$

where $\hat{f}_{h,i}$ is the ith component of $\hat{\mathbf{f}}_h$.

By comparing Eq. (28) with Eq. (24), and Eq. (29) with Eq. (25), it is obvious that by adopting the collocation approach, one domain integral has been dropped from the calculations. More importantly, the dimension of one domain integral has been reduced by half from $2d$ to d. This is expected to tremendously boost the computational expediency, as previously observed for the case of single-patch domains [32]. However, as the matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ in Eq. (27) are not generally positive-definite, real-valued eigensolutions cannot be guaranteed. This will be revisited in the numerical examples section.

4.3 Eigenvalue Solvers

The quality of random field discretization by the K-L expansion depends not only on the truncation parameter N in Eq. (5), but also on the accuracy to which the eigensolutions λ_i and $\phi_i(\mathbf{z}),\ i = 1,\ldots,N$ are approximated by the Galerkin and collocation projection methods. The eigensolutions obtained from the Galerkin formulation are guaranteed to be real-valued, since the system matrices are symmetric and positive-(semi)definite. On the other hand, the collocation projection, albeit efficiently providing convergent results, may not necessarily yield real-valued eigensolutions, if the discretization is not adequately fine. This is deemed a price to be paid, if efficiency is the goal in the numerical analyses. The results obtained by both methods converge to the correct limit, as the mesh is refined [31,32].

For a meticulous subsequent UQ analysis not addressed in this work, the eigensolutions have to be computed accurately, which relies greatly on the eigenvalue solver opted. The choice of the eigenvalue solver for analysis strongly depends on the structure of the system matrices $\mathbf{A}$ and $\mathbf{B}$ in the isogeometric Galerkin method or $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ in the collocation method. For the Galerkin projection, the system matrices formulated in Eqs. (24) and (25) are symmetric. Hence, a number of iterative methods, such as the fast Fourier technique [46] and Krylov subspace methods [47], have been proposed, where the former is applicable only to rectangular domains and homogeneous random fields, and the latter is expensive for large matrices. Therefore, hierarchical matrix techniques [9] and fast multipole methods [48] have been studied as remedies. In particular, hierarchical matrices have proven to demand significantly less storage while dealing with large-scale problems. For nonsymmetric system matrices, such as those introduced by the collocation projection in Eqs. (28) and (29), other iterative algorithms, such as the QZ [49] and Lanczos [50] methods, are the most popular. Note that in the methods proposed in this study, $\mathbf{B}$ and $\hat{\mathbf{B}}$ are banded, sparse matrices, while $\mathbf{A}$ and $\hat{\mathbf{A}}$ are both generally dense. That being noted, MATLAB (Version 2019a) [51] has built-in functions with balancing and preconditioning options to solve various types of eigenvalue problems.

5. NUMERICAL EXAMPLES

Two numerical examples in two and three dimensions are presented to study the accuracy and efficiency of the proposed isogeometric Galerkin and collocation methods for solving the Fredholm integral eigenvalue problem on multipatch domains. Identical quadratic ($p_k = 2$) NURBS and T-spline basis functions were used for geometrical modeling and computational analysis in both numerical examples. Moreover, the T-meshes employed in this work were all analysis-suitable and were refined by using the Bézier extraction operator. All patch couplings were C^0-continuous. Furthermore, the integrals were calculated numerically by Gauss-Legendre quadrature, and the numerical eigensolutions reported in this section were obtained using appropriate eigensolvers available in MATLAB (Version 2019a) [51]. All numerical results reported in this paper were generated from a standard desktop personal computer.

5.1 A Two-Dimensional Connecting Rod

The first numerical example involves random field discretization on a two-dimensional connecting rod by the formulations proposed in Sections 4.1 and 4.2 for the isogeometric Galerkin and collocation methods, respectively. The goal is to study the numerical convergence of the eigensolutions while also investigating the efficiency delivered by the methods.

Define by $\alpha(\mathbf{z},\cdot)$ a two-dimensional homogeneous random field with the covariance function

$$\Gamma(\mathbf{z},\mathbf{z}') = \sigma^2 \exp\left(-\frac{\|\mathbf{z}-\mathbf{z}'\|}{bL}\right), \quad \mathbf{z},\mathbf{z}' \in \mathcal{D} \subset \mathbb{R}^2,$$

where $b = 0.5$, $L = 0.295$, and $\sigma = 0.1$. Recall that the rate of decay in the eigenvalues depends on the correlation length parameter b and L, which have been chosen arbitrarily. The random field is defined on the domain illustrated in Fig. 6(a) with the dimensions in consistent units. Moreover, Fig. 6(b) demonstrates the 14 tensor-product patches coupled in a C^0-continuous manner for representation of the domain. The patch interfaces are indicated by cyan lines.

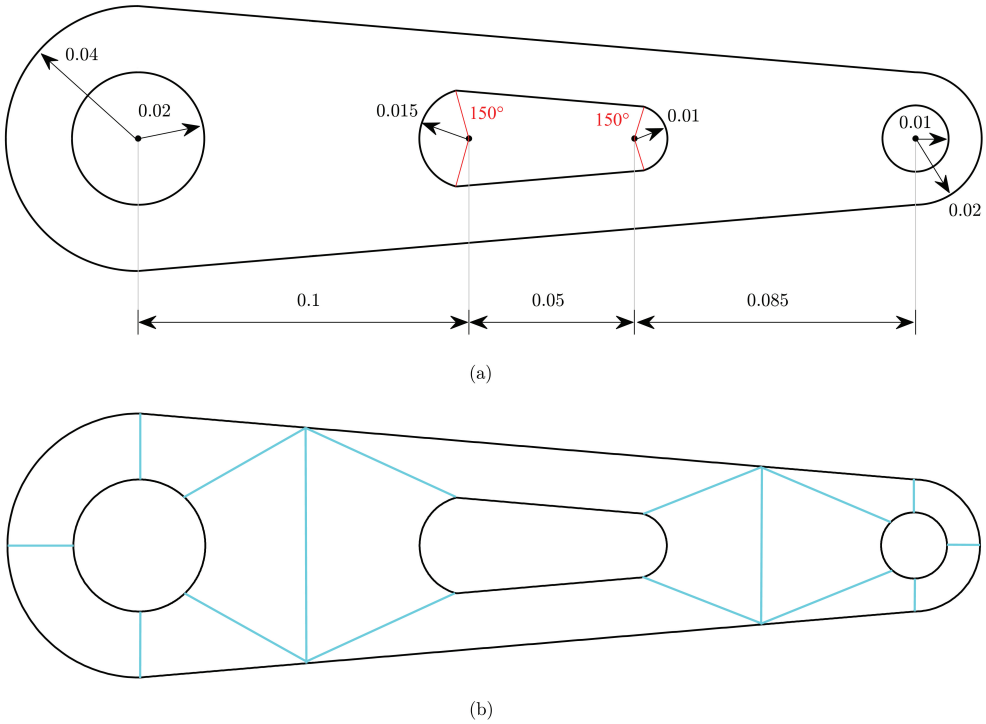

FIG. 6: A two-dimensional connecting rod in Example 1: (a) the geometry with dimensions in consistent units; (b) a fourteen-patch representation of the domain

5.1.1 IGA Discretization

The following settings were considered for both isogeometric Galerkin and collocation formulations. For the multipatch NURBS approach, the problem was solved for three refinements with NURBS, namely, NURBS-Mesh-1, NURBS-Mesh-2, and NURBS-Mesh-3, which are depicted in Fig. 7. For solving the problem with multipatch T-splines, on the other hand, two meshes were considered: Tmesh-1 and Tmesh-2, as shown in Fig. 8. For NURBS methods, the refinement was conducted by simple knot insertion. Moreover, for T-splines methods, the control points associated with the refined T-meshes were calculated by Eq. (22) using the refinement matrix obtained by using the Bézier extraction operator.

There are 14, 56, and 224 control points in NURBS-Mesh-1, NURBS-Mesh-2, and NURBS-Mesh-3, respectively. Moreover, the total numbers of control points for Tmesh-1 and Tmesh-2 are 92 and 256, respectively. Note the refinement propagation from one patch to another in Fig. 7, when the NURBS basis functions are kept C^0-continuous at the patch interfaces. On the other hand, the T-junctions in Fig. 8 have allowed the desired local refinement, especially around the interior holes. In all cases, the geometry of the connecting rod, which comprises straight and circular lines, is modeled exactly, regardless of the mesh refinement and basis functions chosen. Here, the control points and collocation points are demonstrated by closed green circles and closed magenta squares, respectively. Obviously, the collocation points are not used where the isogeometric Galerkin method is concerned.

5.1.2 Eigensolutions

Tables 1 and 2 list the six largest eigenvalues estimated by the various isogeometric Galerkin and collocation methods, respectively. The results are convergent, as the mesh is refined. Furthermore, the results obtained from the NURBS and T-spline basis functions are comparable in terms of accuracy. The same conclusion is valid for the eigenvalues estimated by the isogeometric Galerkin and collocation methods, although those delivered by the former are generally

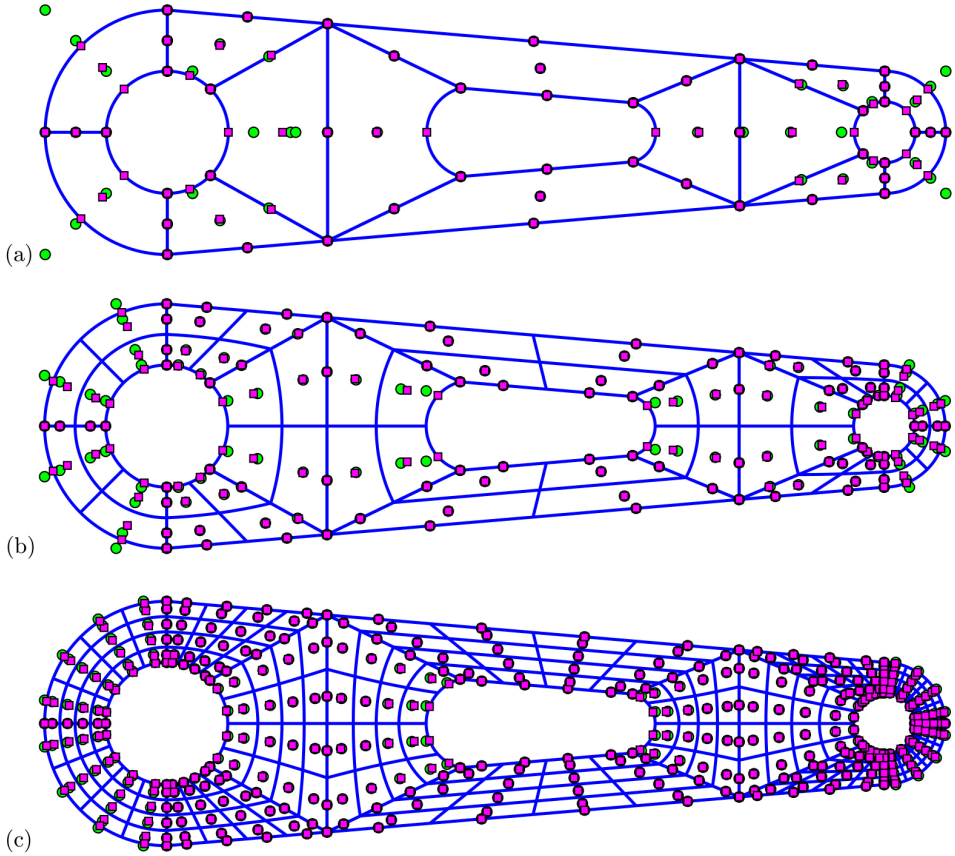

FIG. 7: Three IGA refinements for multipatch NURBS on the two-dimensional connecting rod in Example 1: (a) NURBS-Mesh-1; (b) NURBS-Mesh-2; (c) NURBS-Mesh-3

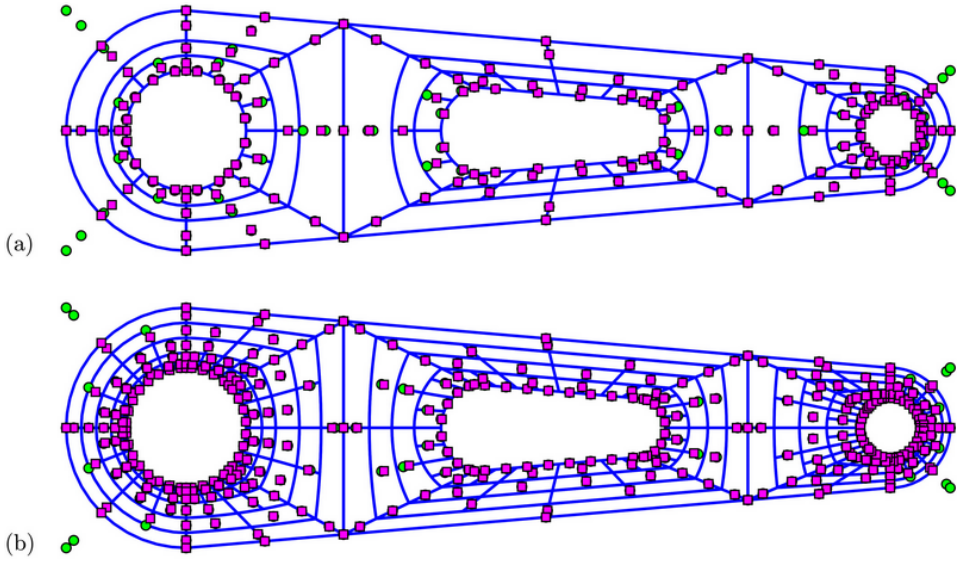

FIG. 8: Two IGA discretizations for multipatch T-splines on the two-dimensional connecting rod in Example 1: (a) Tmesh-1; (b) Tmesh-2

TABLE 1: Six largest eigenvalues estimated by the isogeometric Galerkin methods in Example 1

Mode	Multipatch NURBS			T-splines	
	NURBS-Mesh-1	NURBS-Mesh-2	NURBS-Mesh-3	Tmesh-1	Tmesh-2
1	7.857105×10^{-5}	7.854283×10^{-5}	7.853939×10^{-5}	7.854792×10^{-5}	7.854051×10^{-5}
2	2.386010×10^{-5}	2.384465×10^{-5}	2.384236×10^{-5}	2.384732×10^{-5}	2.384203×10^{-5}
3	8.029485×10^{-6}	8.006341×10^{-6}	8.003674×10^{-6}	8.007837×10^{-6}	8.002647×10^{-6}
4	4.292074×10^{-6}	4.274465×10^{-6}	4.271268×10^{-6}	4.271714×10^{-6}	4.268858×10^{-6}
5	3.940816×10^{-6}	3.922996×10^{-6}	3.920187×10^{-6}	3.915438×10^{-6}	3.911817×10^{-6}
6	2.151460×10^{-6}	2.152474×10^{-6}	2.151662×10^{-6}	2.134051×10^{-6}	2.135975×10^{-6}

TABLE 2: Six largest eigenvalues estimated by the isogeometric collocation methods in Example 1

Mode	Multipatch NURBS			T-splines	
	NURBS-Mesh-1	NURBS-Mesh-2	NURBS-Mesh-3	Tmesh-1	Tmesh-2
1	7.850899×10^{-5}	7.853767×10^{-5}	7.853849×10^{-5}	7.852758×10^{-5}	7.853235×10^{-5}
2	2.377607×10^{-5}	2.383488×10^{-5}	2.384157×10^{-5}	2.379473×10^{-5}	2.380977×10^{-5}
3	7.950480×10^{-6}	7.995213×10^{-6}	8.002188×10^{-6}	7.978778×10^{-6}	7.990312×10^{-6}
4	4.138326×10^{-6}	4.265461×10^{-6}	4.269689×10^{-6}	4.158671×10^{-6}	4.205105×10^{-6}
5	3.832094×10^{-6}	3.898096×10^{-6}	3.917790×10^{-6}	3.866458×10^{-6}	3.882279×10^{-6}
6	2.022631×10^{-6}	2.130468×10^{-6}	2.148870×10^{-6}	2.067065×10^{-6}	2.099071×10^{-6}

expected to be more accurate than those of the latter. Note that unlike the Galerkin method, the collocation formulation may not necessarily yield all real-valued eigenvalues, as mentioned in Section 4.2. However, the top six eigensolutions in this example are all real-valued in both cases of basis functions.

Figures 9 and 10 illustrate the sixth eigenfunction obtained by the isogeometric Galerkin and collocation methods via (26) and (31), respectively. By comparing the two figures, the collocation projection methods have accurately estimated the eigenfunctions, especially as the mesh is well-refined. Moreover, the results by either method are convergent, regardless of the basis functions chosen. The eigenfunctions of relatively higher modes are generally more challenging to estimate by numerical methods. This was the reason for choosing the sixth eigenfunction for visual investigations.

5.1.3 Computational Effort

As the isogeometric collocation methods have proved to be of satisfactory accuracy, their computational cost in comparison with their Galerkin counterparts is of interest. Table 3 indicates the computational efforts of the isogeometric collocation and Galerkin methods, where for various meshes with NURBS and T-splines, the size of the computational problem is stated in terms of the numbers of elements and control points. The CPU times for constructing the system matrices and for solving the problem by the collocation isogeometric methods are listed in seconds under the headings "Time-1" and "Time-2," respectively. The next two columns, namely, "Ratio-1" and "Ratio-2," state the ratio of the CPU time taken by the isogeometric Galerkin methods over the CPU time needed for the isogeometric collocation methods. For instance, for Tmesh-2, 170.06 and 311.79 sec are required to construct the system matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ and to solve the problem, respectively. In this case, it takes 10.26 and 6.02 times more time, respectively, for the Galerkin method to carry out the aforementioned tasks.

The relatively low computational efficiency of the isogeometric Galerkin methods is strongly prohibitive for large-scale problems. According to Table 3, while most of the CPU time is allocated to the construction of system matrices, the isogeometric collocation methods are tremendously more efficient than those based on the Galerkin approach. The reason is that the isogeometric Galerkin formulation entails $2d$-dimensional integration to compute

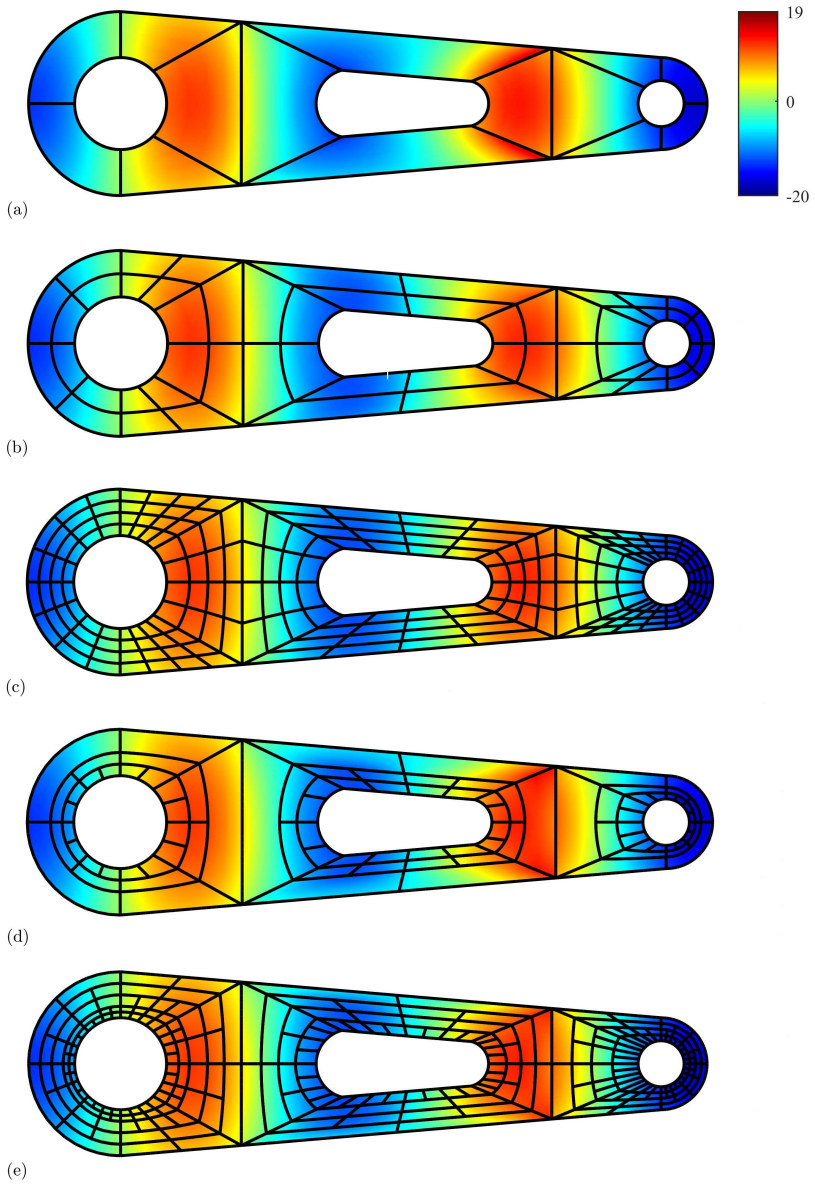

FIG. 9: Sixth eigenfunction obtained by the isogeometric Galerkin methods with multipatch NURBS and T-splines in Example 1: (a) NURBS-Mesh-1; (b) NURBS-Mesh-2; (c) NURBS-Mesh-3; (d) Tmesh-1; (e) Tmesh-2

each component in lieu of d-dimensional integration in collocation. This argument becomes even stronger as the mesh is refined. For instance, Ratio-1 increases from 1.63 for NURBS-Mesh-1 to 6.27 for NURBS-Mesh-3. This was evident for single-patch domains in a previous study by Jahanbin and Rahman [32]. Moreover, the computational efforts corresponding to the isogeometric methods based on NURBS and T-splines are comparable, considering the mesh sizes.

The first numerical example confirmed the accuracy of the proposed isogeometric collocation- and Galerkin-based methods with NURBS and T-spline basis functions for solving the Fredholm eigenvalue problem on a multi-patch two-dimensional domain. It was also shown that the collocation methods were far more computationally efficient than their Galerkin counterparts. This efficiency is expected to become more prominent for three-dimensional

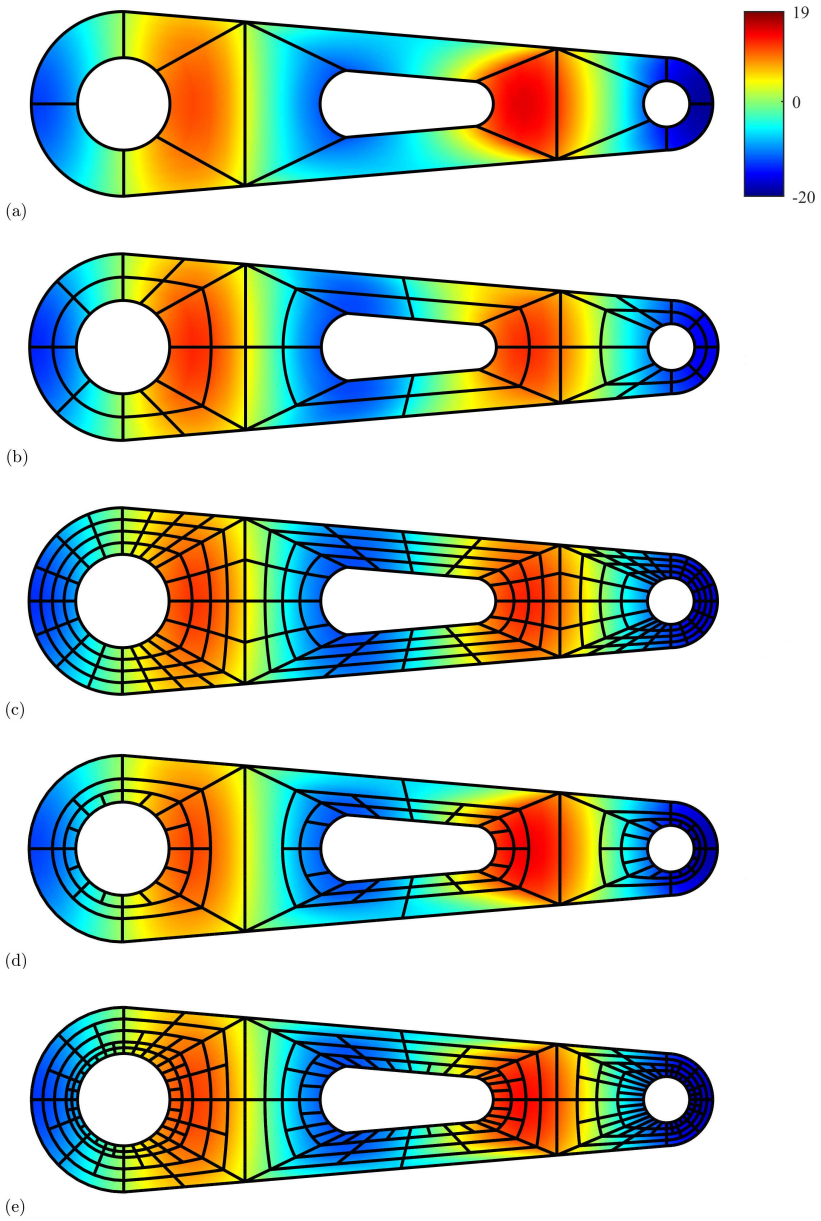

FIG. 10: Sixth eigenfunction obtained by the isogeometric collocation methods with multipatch NURBS and T-splines in Example 1: (a) NURBS-Mesh-1; (b) NURBS-Mesh-2; (c) NURBS-Mesh-3; (d) Tmesh-1; (e) Tmesh-2

domains, where for large systems of equations, the isogeometric Galerkin method may even be computationally prohibitive. The next numerical example will address this issue.

5.2 A Three-Dimensional Disk Brake

The second example entails random field discretization on a three-dimensional disk brake. An actual brake system in a passenger vehicle, illustrated in Fig. 11(a) [52], slows the motion of a wheel by pushing the brake pads against a rotor with a set of calipers.

TABLE 3: Computational efficiency of the isogeometric collocation methods in comparison with their Galerkin counterparts in Example 1

Mesh	Mesh Specifications		CPU Time			
	No. of Elements	No. of Control Pts.	Time-1[a]	Time-2[b]	Ratio-1[c]	Ratio-2[d]
NURBS-Mesh-1	14	120	2.76	4.62	2.01	1.63
NURBS-Mesh-2	56	216	24.08	26.81	5.54	5.08
NURBS-Mesh-3	224	492	266.70	270.05	6.33	6.27
Tmesh-1	92	205	30.47	75.64	7.07	3.42
Tmesh-2	256	392	170.06	311.79	10.26	6.02

[a] Time-1 = Elapsed CPU time for constructing the system matrices by the collocation method in seconds.
[b] Time-2 = Elapsed CPU time for obtaining the eigensolutions by the collocation method in seconds.
[c] Ratio-1 = Galerkin method CPU time/Collocation method CPU time for constructing the system matrices.
[d] Ratio-2 = Galerkin method total CPU time/Collocation method total CPU time for obtaining the eigensolutions.

Define a three-dimensional homogeneous random field with the covariance function

$$\Gamma(\mathbf{z},\mathbf{z}') = \sigma^2 \exp\left(-\frac{\|\mathbf{z}-\mathbf{z}'\|}{bL}\right), \quad \mathbf{z},\mathbf{z}' \in \mathcal{D} \subset \mathbb{R}^3, \tag{32}$$

where $b = 0.5$, $L = 0.288$, and $\sigma = 0.1$. Figures 11(b) and 11(c) demonstrate two views of the three-dimensional model for the disk brake. The disk of outer diameter 0.288 in consistent units is defined, as depicted in Fig. 11(d). Figure 11(e) depicts a quarter of the disk with four patches that are coupled in a C^0-continuous manner, where the patch interfaces are marked with cyan lines. As a result, there are 16 patches in the entire domain.

Since solving this three-dimensional problem with isogeometric Galerkin formulation is computationally intensive and multipatch NURBS lack flexibility in conveniently modeling such domain, only the isogeometric collocation method with T-splines is implemented. Figure 12 illustrates the top view and three-dimensional view of the three IGA T-meshes used, namely, Tmesh-1, Tmesh-2, and Tmesh-3. The T-meshes are all analysis-suitable. Similar to Example 1, the disk geometry is modeled precisely in all cases. The control points and collocation points are indicated by closed green circles and closed magenta squares, respectively. Some T-junctions are observable in the top views.

Table 4 lists the six largest eigenvalues estimated by the isogeometric collocation method with T-splines. Based on the results, the eigenvalues are convergent as the mesh is refined. However, recall that the collocation formulation does not guarantee real-valued eigensolutions. Here, there are a few cases of complex-valued numbers in the top six eigensolutions. However, the imaginary parts are relatively small and negligible and are hence dropped. The aforementioned numbers are marked with asterisks in Table 4.

Figure 13 illustrates six eigenfunctions for Tmesh-3, which correspond to the six largest eigenvalues listed in Table 4. As observed, the eigenfunctions are elegantly approximated by smooth T-splines. Moreover, the eigensolutions mostly appear in pairs, since ignoring the four interior small holes, the geometry is axisymmetric. Evidently, the second and third eigenfunctions are paired, as are the fourth and fifth eigenfunctions. This can be inferred from their associated eigenvalues listed in Table 4 as well.

Akin to Table 3, the numbers of elements and control points, along with the computational costs in seconds to construct the system matrices and to solve the problem by isogeometric collocation methods with T-splines, are listed in Table 5. Note that the system matrix $\hat{\mathbf{B}}$ in Eq. (27) is a typically a banded matrix, whereas the $\hat{\mathbf{A}}$ matrix is not sparse. Hence, the $\hat{\mathbf{B}}^{-1}\hat{\mathbf{A}}$ matrix, the eigensolutions of which being desirable, is typically not sparse. With this clarification, solving the problem for the finest mesh (Tmesh-3) took more than 14 hours on a standard desktop personal computer. Based on prior experience [32], it would have taken more than 6 days for its Galerkin counterpart. This, once again, underscores the computational expediency of the proposed isogeometric collocation method.

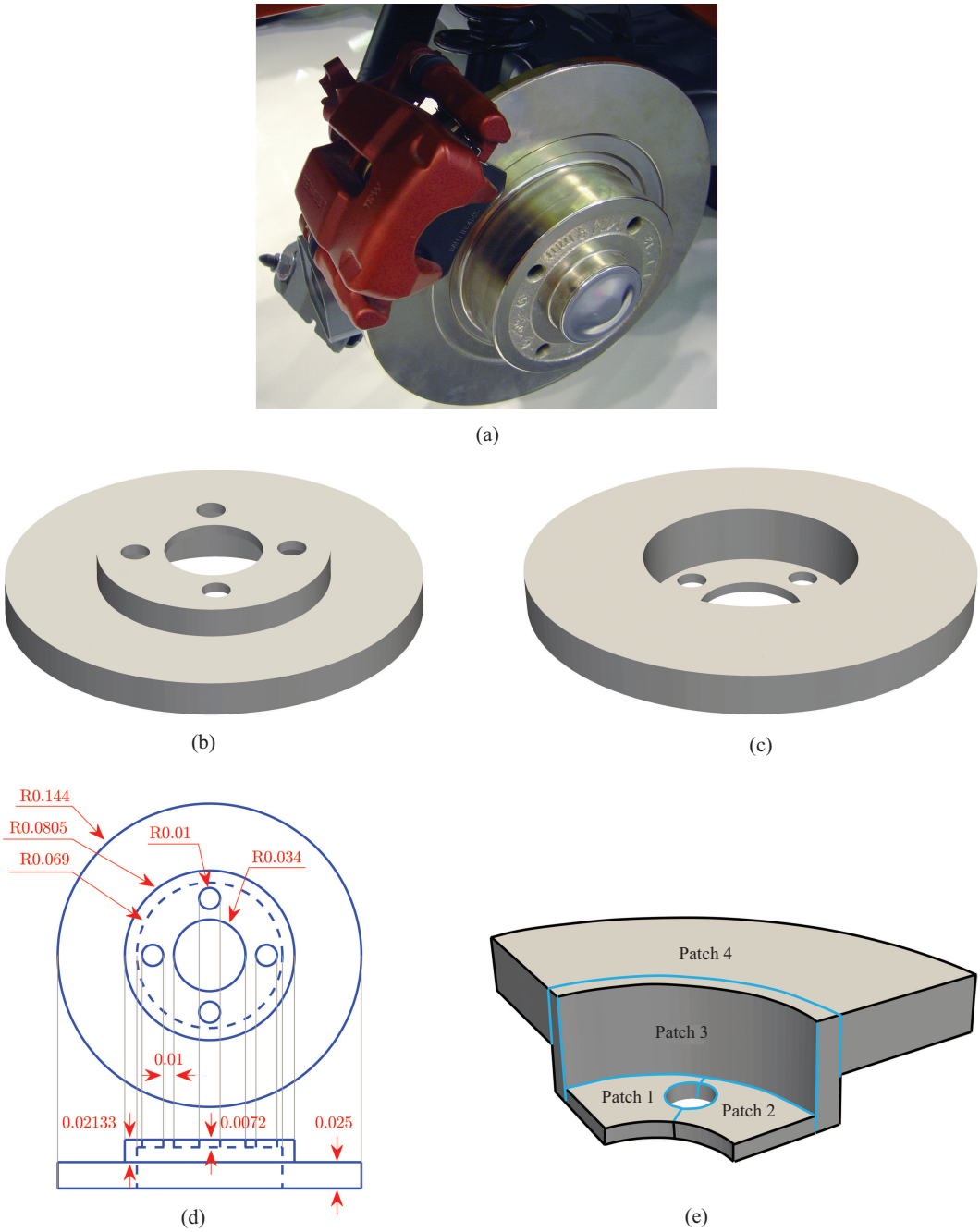

FIG. 11: A three-dimensional disk brake in Example 2; (a) Actual brake system in a passenger vehicle [52]; (b) three-dimensional view of the simplified model from the top; (c) three-dimensional view of the simplified model from the bottom; (d) geometry with consistent units; (e) C^0-continuous patch coupling over a quarter of the disk

6. DISCUSSION

In this work, the prefix "iso" refers to using identical basis functions for geometrical modeling and approximating the eigensolutions of a well-known integral problem in UQ, rather than the heavily researched BVPs. Isogeometric

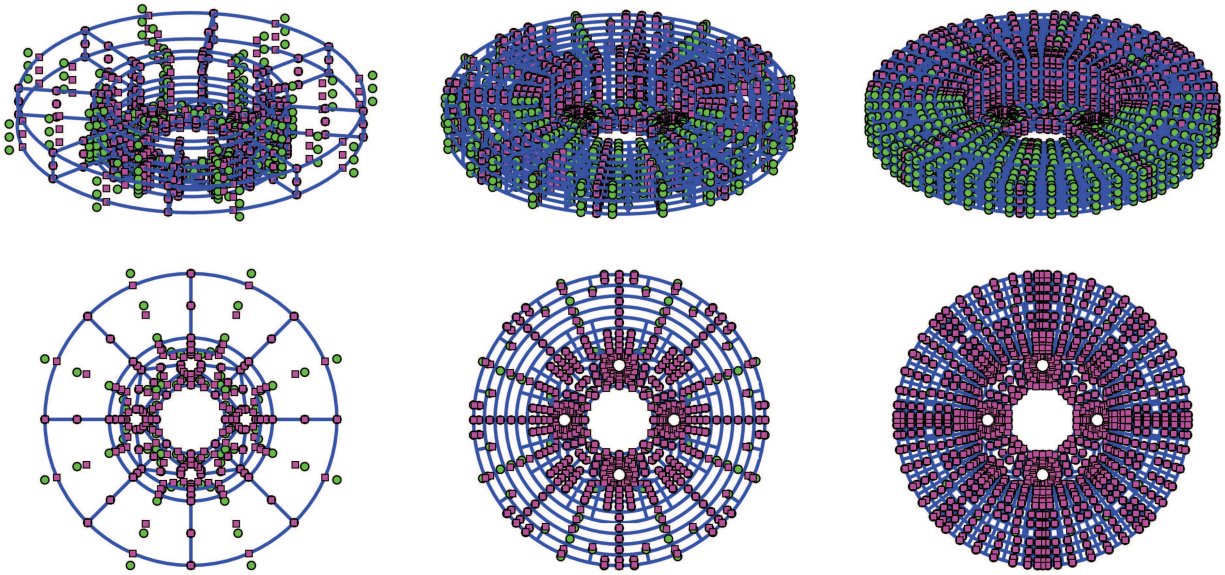

FIG. 12: Three-dimensional view and top view of the analysis-suitable T-meshes in Example 2: (a) Tmesh-1; (b) Tmesh-2; (c) Tmesh-3

TABLE 4: Six largest eigenvalues estimated by the isogeometric collocation method with T-splines in Example 2

Mode	Tmesh-1	Tmesh-2	Tmesh-3
1	5.279380×10^{-6}	5.481527×10^{-6}	5.536382×10^{-6}
2	$1.456313 \times 10^{-6*}$	$1.536563 \times 10^{-6*}$	$1.551816 \times 10^{-6*}$
3	$1.456313 \times 10^{-6*}$	$1.536563 \times 10^{-6*}$	$1.551816 \times 10^{-6*}$
4	4.801366×10^{-7}	5.136870×10^{-7}	5.198493×10^{-7}
5	4.785992×10^{-7}	5.119457×10^{-7}	5.178820×10^{-7}
6	2.147980×10^{-7}	2.369918×10^{-7}	2.382781×10^{-7}

*A negligible imaginary part is dropped.

Galerkin and collocation methods were proposed to solve the Fredholm integral problem on any arbitrary or complex-shaped geometry by dint of multipatch NURBS or analysis-suitable T-splines. The impact of different types of covariance function with a wide range of function regularities and correlation lengths had been studied [31,32], where in-depth error analyses were conducted. In contrast, this work focused on the practical aspects of the methods, especially on multipatch geometries. However, to analyze the problems with more complex geometries in a convenient manner, user-friendly commercial packages that possess T-spline features need to be developed.

That being noted, volumetric mesh generation for T-splines on a user-friendly graphical interface is not a trivial task. Currently, the "T-splines for Rhino" package [53] is able to model complex surfaces before generating analysis-suitable surface T-meshes. Scott et al. [54] have used Rhino for solving BVPs by boundary-element methods and collocation projection. One motivation behind the boundary-element formulation in the aforementioned work might have been the ability of Rhino to provide surface, and not volumetric, T-meshes. Nevertheless, research is ongoing for three-dimensional T-spline mesh generation on complex geometries with commercial packages that possess friendly graphical user interfaces. The authors envision the integration of the proposed isogeometric collocation and Galerkin methods with commercial packages as an influential step forward in the random field discretization process, when the K-L expansion is implemented to UQ analysis.

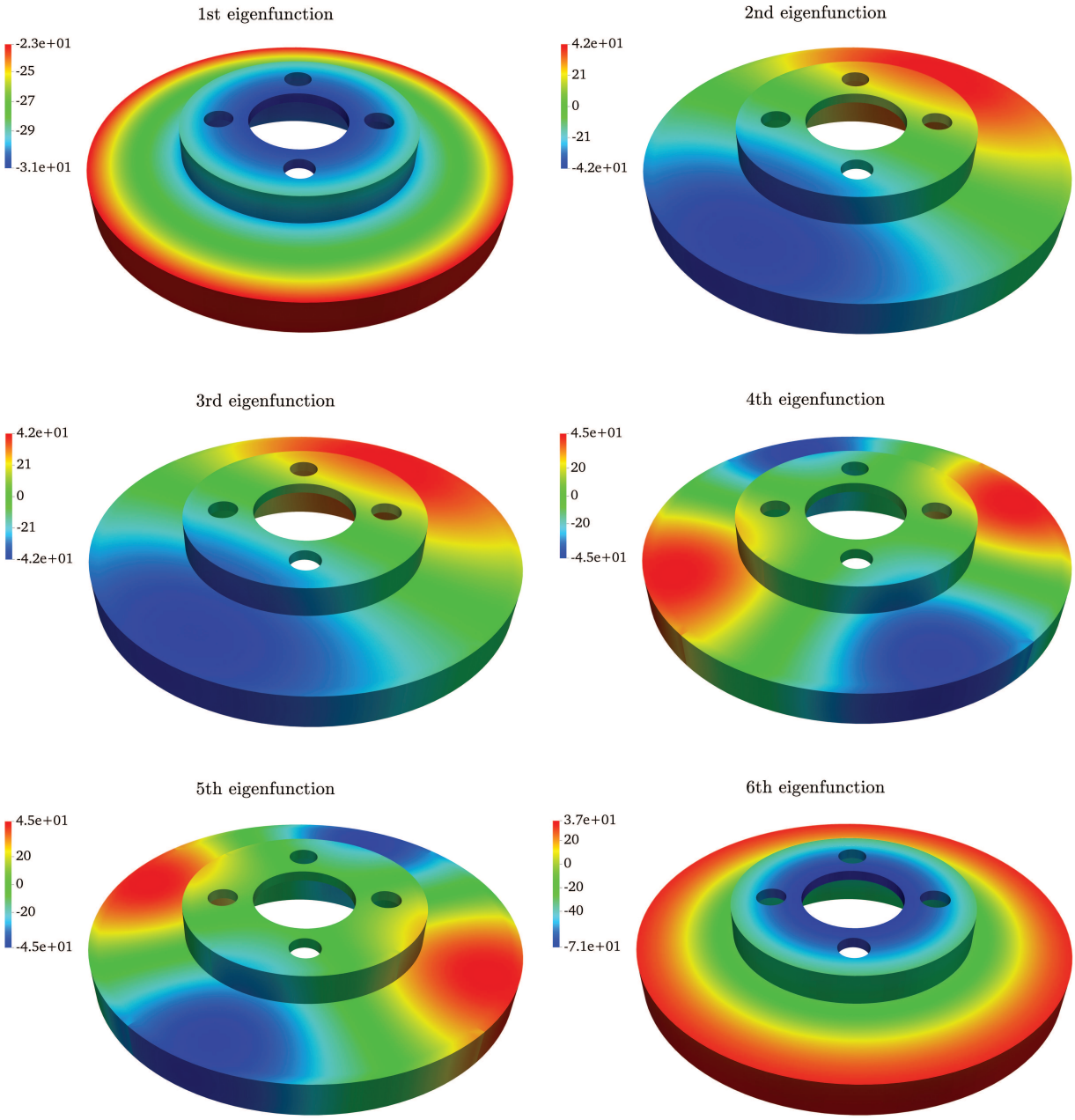

FIG. 13: Six eigenfunctions corresponding to the six largest eigenvalues for random field discretization by the isogeometric collocation method with T-splines for Tmesh-3 in Example 2

TABLE 5: Mesh details and computational effort in Example 2

Mesh	No. of Elements	No. of Control Points	Time for System Matrices (s)	Total Time (s)
Tmesh-1	56	1068	71.85	85.95
Tmesh-2	504	2764	4122.34	4220.28
Tmesh-3	2016	6096	50867.75	51351.77

7. CONCLUSION AND OUTLOOK

Two new isogeometric methods, one rooted in the Galerkin projection and the other stemming from the collocation projection, were developed for solving the Fredholm integral eigenvalue problem on arbitrary multipatch domains, enabling random field discretization by means of the well-known K-L expansion. Compared with similar existing works on IGA, the methods proposed can now handle markedly more complex geometries, including multiply connected bodies, that cannot be represented by tensor-product, single-patch domains alone. In both methods, the unknown eigensolutions are projected onto concomitant finite-dimensional approximation spaces, where NURBS and analysis-suitable rational T-splines are exploited as basis functions. Using these approximation spaces, finite-dimensional matrix eigenvalue problems are formulated, where the system matrices are constructed by NURBS or rational T-splines as basis and subsequent domain integration. Finally, the eigensolutions are obtained using standard methods of linear algebra. Similar to the existing methods but applicable to single-patch domains only, the isogeometric Galerkin and collocation methods developed here preserve an exact geometrical representation of complex engineering structures and exploit the regularity of isogeometric basis functions to the extent possible, furnishing patchwise smooth but globally continuous eigensolutions. Numerical results of eigensolutions obtained from two- and three-dimensional engineering problems indicate that the collocation method, when compared with the isogeometric Galerkin method, is not only accurate and convergent, but also substantially more economical. The isogeometric collocation method achieves this desirable feat by sidestepping one d-dimensional domain integration in forming the system matrices, whereas a $2d$-dimensional domain integration is mandated in the isogeometric Galerkin method. As a result, the collocation method in the context of IGA offers a tremendous boost to computational expediency when generating the K-L expansion.

Although the methods were developed to handle any arbitrary multipatch domain, there is a need to develop commercial packages capable of conveniently generating three-dimensional volumetric meshes equipped with NURBS or rational T-splines. Therefore, the integration of the proposed methods with commercial packages will be fruitful for random field discretization on more complex geometries in practical applications.

ACKNOWLEDGMENT

This research was supported by the U.S. National Science Foundation (Grant No. CMMI-1607398).

REFERENCES

1. Karhunen, K., Über Lineare Methoden in Der Wahrscheinlichkeitsrechnung, *Ann. Acad. Sci. Fenn. Ser. A.*, **37**:3–79, 1947.
2. Loève, M., *Probability Theory, Vol. II*, New York: Springer, 1977.
3. Stefanou, G. and Papadrakakis, M., Assessment of Spectral Representation and Karhunen-Loève Expansion Methods for the Simulation of Gaussian Stochastic Fields, *Comput. Methods Appl. Mech. Eng.*, **196**:2465–2477, 2007.
4. Betz, W., Papaioannou, I., and Straub, D., Numerical Methods for the Discretization of Random Fields by Means of the Karhunen-Loève Expansion, *Comput. Methods Appl. Mech. Eng.*, **271**:109–129, 2014.
5. Huang, S.P., Quek, S.T., and Phoon, K.K., Convergence Study of the Truncated Karhunen-Loève Expansion for Simulation of Stochastic Processes, *Int. J. Numer. Methods Eng.*, **52**:1029–1043, 2001.
6. Fredholm, E.I., Sur Une Classe d'Equations Fonctionnelles, *Acta Math.*, **27**:365–390, 1903.
7. Atkinson, K., *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge: Cambridge University Press, 1997.
8. Schwab, C. and Todor, R.A., Karhunen-Loève Approximation of Random Fields by Generalized Fast Multipole Methods, *J. Comput. Phys.*, **217**:100–122, 2006.
9. Khoromskij, B.N., Litvinenko, A., and Matthies, H.G., Application of Hierarchical Matrices for Computing the Karhunen-Loève Expansion, *Comput.*, **84**:49–67, 2009.
10. Hughes, T.J.R., Cottrell, J.A., and Bazilevs, Y., Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement, *Comput. Methods Appl. Mech. Eng.*, **194**:4135–4195, 2005.

11. Cottrell, J.A., Hughes, T.J.R., and Bazilevs, Y., *Isogeometric Analysis: Toward Integration of CAD and FEA*, New York: John Wiley & Sons, 2009.
12. Sederberg, T.W., Zheng, J., Bakenov, A., and Nasri, A., T-Splines and T-NURCCs, *ACM Trans. Graph.*, **22**(3):477–484, 2003.
13. Urick, B., Marussig, B., Cohen, E., Crawford, R.H., Hughes, T.J.R., and Riesenfeld, R.F., Watertight Boolean Operations: A Framework for Creating CAD-Compatible Gap-Free Editable Solid Models, *Comput.-Aided Des.*, **115**:147–160, 2019.
14. Sederberg, T.W., Finnigan, G.T., Li, X., and Lin, H., Watertight Trimmed NURBS, *ACM Trans. Graph.*, **27**(3):79-1–79-8, 2008.
15. Bazilevs, Y., Calo, V.M., Cottrell, J.A., Evans, J.A., Hughes, T.J.R., Lipton, S., Scott, M.A., and Sederberg, T.W., Isogeometric Analysis Using T-Splines, *Comput. Methods Appl. Mech. Eng.*, **199**:229–263, 2010.
16. Dörfel, M.R., Jüttler, B., and Simeon, B., Adaptive Isogeometric Analysis by Local H-Refinement with T-Splines, *Comput. Methods Appl. Mech. Eng.*, **199**:264–275, 2010.
17. Scott, M.A., Li, X., Sederberg, T.W., and Hughes, T.J.R., Local Refinement of Analysis-Suitable T-Splines, *Comput. Methods Appl. Mech. Eng.*, **213–216**:206–222, 2012.
18. Buffa, A., Cho, D., and Sangalli, G., Linear Independence of the T-Spline Blending Functions Associated with Some Particular T-Meshes, *Comput. Methods Appl. Mech. Eng.*, **199**:1437–1445, 2010.
19. Li, X., Zheng, J., Sederberg, T.W., Hughes, T.J.R., and Scott, M.A., On Linear Independence of T-Spline Blending Functions, *Comput. Aided Geom. Des.*, **29**:63–76, 2012.
20. da Veiga, L., Buffa, A., Sangalli, G., and Vázquez, R., Analysis-Suitable T-Splines of Arbitrary Degree: Definition, Linear Independence, and Approximation Properties, *Math. Models Methods Appl. Sci.*, **23**(11):1979–2003, 2013.
21. Scott, M.A., Borden, M.J., Verhoosel, C.V., Sederberg, T.W., and Hughes, T.J.R., Isogeometric Finite Element Data Structures Based on Bèzier Extraction of T-Splines, *Int. J. Numer. Methods Eng.*, **88**:126–156, 2011.
22. May, S., Vignollet, J., and de Borst, R., The Role of the Bèzier Extraction Operator for T-Splines of Arbitrary Degree: Linear Dependencies, Partition of Unity Property, Nesting Behaviour and Local Refinement, *Int. J. Numer. Methods Eng.*, **103**:547–581, 2015.
23. Evans, E.J., Scott, M.A., Li, X., and Thomas, D.C., Hierarchical T-Splines: Analysis-Suitability, Bèzier Extraction, and Application as an Adaptive Basis for Isogeometric Analysis, *Comput. Methods Appl. Mech. Eng.*, **284**:1–20, 2015.
24. Chen, L. and de Borst, R., Adaptive Refinement of Hierarchical T-Splines, *Comput. Methods Appl. Mech. Eng.*, **337**:220–245, 2018.
25. Wei, X., Zhang, Y., Liu, L., and Hughes, T.J.R., Truncated T-Splines: Fundamentals and Methods, *Comput. Methods Appl. Mech. Eng.*, **316**:349–372, 2017.
26. Ghanem, R. and Spanos, P.D., *Stochastic Finite Elements: A Spectral Approach*, New York: World Publishing Corp., 1991.
27. Belytschko, T., Lu, Y.Y., and Gu, L., Element-Free Galerkin Methods, *Int. J. Numer. Methods Eng.*, **37**:229–256, 1994.
28. Rahman, S. and Xu, H., A Meshless Method for Computational Stochastic Mechanics, *Int. J. Comput. Methods Eng. Sci. Mech.*, **64**:41–58, 2005.
29. Papaioannou, I., *Non-Intrusive Finite Element Reliability Analysis*, Saarbrücken, Germany: SVH-Verlag, 2013.
30. Pranesh, S. and Ghosh, D., Faster Computation of the Karhunen-Loève Expansion Using Its Domain Independence Property, *Comput. Methods Appl. Mech. Eng.*, **285**:125–145, 2015.
31. Rahman, S., A Galerkin Isogeometric Method for Karhunen-Loève Approximation of Random Fields, *Comput. Methods Appl. Mech. Eng.*, **338**:533–561, 2018.
32. Jahanbin, R. and Rahman, S., An Isogeometric Collocation Method for Efficient Random Field Discretization, *Int. J. Numer. Methods Eng.*, **117**(3):344–369, 2019.
33. Mantzaflaris, A., Jütler, B., Khoromskij, B.N., and Langer, U., Low Rank Tensor Methods in Galerkin-Based Isogeometric Analysis, *Comput. Methods Appl. Mech. Eng.*, **316**:1062–1085, 2017.
34. Beck, J., Sangalli, G., and Tamellini, L., A Sparse-Grid Isogeometric Solver, *Comput. Methods Appl. Mech. Eng.*, **335**:128–151, 2018.
35. Li, K., Gao, W., Wu, D., Song, C., and Chen, T., Spectral Stochastic Isogeometric Analysis of Linear Elasticity, *Comput. Methods Appl. Mech. Eng.*, **332**:157–190, 2018.
36. Jahanbin, R. and Rahman, S., Stochastic Isogeometric Analysis in Linear Elasticity, *Comput. Methods Appl. Mech. Eng.*, **364**:112928, 2020.

37. Dunford, N. and Schwartz, J.T., *Linear Operators, Spectral Theory, Self Adjoint Operators in Hilbert Space, Part 2*, Hoboken, NJ: Wiley-Interscience, 1988.
38. Matthies, H.G. and Keese, A., Galerkin Methods for Linear and Nonlinear Elliptic Stochastic Partial Differential Equations, *Comput. Methods Appl. Mech. Eng.*, **194**:1295–1331, 2005.
39. Le Maître, O.P. and Knio, O.M., *Spectral Methods for Uncertainty Quantification*, New York: Springer Science + Business Media, 2010.
40. Xu, H. and Rahman, S., A Generalized Dimension-Reduction Method for Multidimensional Integration in Stochastic Mechanics, *Int. J. Numer. Methods Eng.*, **61**:1992–2019, 2004.
41. Xiong, F., Greene, S., and Chen, W., A New Sparse Grid Based Method for Uncertainty Propagation, *Struct. Multidisc. Optim.*, **41**:335–349, 2010.
42. Cottrell, J.A., Hughes, T.J.R., and Reali, A., Studies of Refinement and Continuity in Isogeometric Structural Analysis, *Comput. Methods Appl. Mech. Eng.*, **196**:4160–4183, 2007.
43. Piegl, L.A. and Tiller, W., *The NURBS Book*, 2nd ed., Berlin: Springer-Verlag, 1997.
44. Farin, G.E., *NURBS Curves and Surfaces: From Projective Geometry to Practical Use*, Natick, MA: A.K. Peters, Ltd., 1999.
45. De Boor, C., On Calculation with B-Splines, *J. Approx. Theory*, **6**:50–62, 1972.
46. Frigo, M. and Johnson, S.G., FFTW: An Adaptive Software Architecture for the FFT, in *Proc. of 1998 ICASSP Conference*, 1998.
47. Saad, Y., *Numerical Methods for Large Eigenvalue Problems*, New York: Halsted Press, 1992.
48. Greengard, L. and Rokhlin, V., A New Version of the Fast Multipole Method for the Laplace Equation in Three Dimensions, *Acta Numer.*, **6**:229–269, 1997.
49. Golub, G.H. and van Loan, C.F., *Matrix Computations*, 3rd ed., Baltimore, MD: Johns Hopkins University Press, 1996.
50. Wu, K. and Simon, H., Thick-Restart Lanczos Method for Large Symmetric Eigenvalue Problems, *SIAM J. Matrix Anal. Appl.*, **22**:602–616, 2000.
51. MATLAB, Version 2019a, Natick, MA: The MathWorks Inc., 2019.
52. Wikipedia, Disk Brake, accessed May, from https://en.wikipedia.org/wiki/Disc_brake, 2020.
53. T-Splines Inc., from http://www.tsplines.com/rhino/, 2011.
54. Scott, M.A., Simpson, R.N., Evans, J.A., Lipton, S., Bordas, S.P.A., Hughes, T.J.R., and Sederberg, T.W., Isogeometric Boundary Element Analysis Using Unstructured T-Splines, *Comput. Methods Appl. Mech. Eng.*, **254**:197–221, 2013.

APPENDIX A. B-SPLINES

This Appendix briefly introduces B-splines for the paper to be self-contained. For $d = 1$, 2, or 3, consider a d-dimensional Cartesian coordinate system in the parametric domain $\hat{\mathcal{D}} = [0,1]^d$, where an arbitrary point has coordinate $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_d)$. For the coordinate direction k, where $k = 1, \ldots, d$, define a positive integer $n_k \in \mathbb{N}$ and a non-negative integer $p_k \in \mathbb{N}_0$, representing the total number of basis functions and polynomial degree, respectively. Given n_k and p_k, introduce on the parametric interval $[0,1] \subset \mathbb{R}$, an ordered knot vector

$$\boldsymbol{\xi}_k := (0 = \xi_{k,1}, \xi_{k,2}, \ldots, \xi_{k,n_k+p_k+1} = 1), \quad \xi_{k,1} \leq \xi_{k,2} \leq \ldots \leq \xi_{k,n_k+p_k+1}, \tag{A.1}$$

where ξ_{k,i_k} is the i_kth knot with $i_k = 1, 2, \ldots, n_k + p_k + 1$ representing the knot index for the coordinate direction k. A knot vector is called open if its first and last knots appear $(p_k + 1)$ times. Open knot vectors are standard in the CAD literature [43].

The B-spline functions for a given degree are defined in a recursive manner using the knot vector. Denote by $B^k_{i_k,p_k,\boldsymbol{\xi}_k}(\xi_k)$ the i_kth univariate B-spline function with degree p_k and knot vector $\boldsymbol{\xi}_k$ for the coordinate direction k. Given the *zero*-degree basis functions,

$$B^k_{i_k,0,\boldsymbol{\xi}_k}(\xi_k) = \begin{cases} 1, & \xi_{k,i_k} \leq \xi_k < \xi_{k,i_k+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{A.2}$$

for $k = 1, \ldots, d$, all higher-order B-spline functions are efficiently generated by the recursive Cox-de Boor formula [45],

$$B^k_{i_k,p_k,\boldsymbol{\xi}_k}(\xi_k) = \frac{\xi_k - \xi_{k,i_k}}{\xi_{k,i_k+p_k} - \xi_{k,i_k}} B^k_{i_k,p_k-1,\boldsymbol{\xi}_k}(\xi_k) + \frac{\xi_{k,i_k+p_k+1} - \xi_k}{\xi_{k,i_k+p_k+1} - \xi_{k,i_k+1}} B^k_{i_k+1,p_k-1,\boldsymbol{\xi}_k}(\xi_k), \tag{A.3}$$

where $1 \leq k \leq d$, $1 \leq i_k \leq n_k$, $1 \leq p_k < \infty$, and $0/0$ is considered as *zero*.

The B-spline functions for any $k = 1, \ldots, d$ and $p_k \in \mathbb{N}_0$ satisfy the following desirable properties [10,43,45]: (1) they are non-negative, (2) they are locally supported, (3) they are linearly independent, and (4) they form a partition of unity. As a result, the B-spline functions in Eqs. (A.2) and (A.3) form a basis on [0,1], which is crucial for developing computational methods.

The multivariate B-splines in d variables $\xi_1, \ldots, \xi_d$ are constructed from the tensor product of the univariate B-splines stemming from the chosen knot vectors $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d$. Define two multi-indices $\mathbf{i} := (i_1, \ldots, i_d)$ and $\mathbf{p} := (p_1, \ldots, p_d)$, and a collection $\boldsymbol{\Xi} := (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d)$ of knot vectors. For the first multi-index, denote by

$$\mathcal{I} := \{\mathbf{i} = (i_1, \ldots, i_d) : 1 \leq i_k \leq n_k,\ 1 \leq k \leq d\}, \tag{A.4}$$

a multi-index set. Then, for $\mathbf{i} \in \mathcal{I}$, $\mathbf{p} \in \mathbb{N}_0^d$, and $\boldsymbol{\Xi}$, the multivariate B-spline function $\mathrm{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}} : \hat{\mathcal{D}} \to \mathbb{R}$ is defined as

$$\mathrm{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}(\boldsymbol{\xi}) := \prod_{k=1}^{d} B^k_{i_k,p_k,\boldsymbol{\xi}_k}(\xi_k), \tag{A.5}$$

with the corresponding tensor-product B-spline space,

$$\mathcal{B}_h := \bigotimes_{k=1}^{d} \mathrm{span}\{B^k_{i_k,p_k,\boldsymbol{\xi}_k}(\xi_k)\}_{i_k=1,\ldots,n_k} = \mathrm{span}\{\mathrm{B}_{\mathbf{i},\mathbf{p},\boldsymbol{\Xi}}(\boldsymbol{\xi})\}_{\mathbf{i}\in\mathcal{I}}. \tag{A.6}$$

Note that the functions in $\mathcal{B}_h$ are piecewise polynomials of degree p_k along each coordinate direction $k = 1, \ldots, d$. Due to the tensor-product structure, multivariate B-spline functions inherit most of the aforementioned properties of their univariate counterparts, namely, non-negativity, local support, linear independence, partition of unity, and regularity.