Received: 8 June 2018 | Revised: 11 September 2018 | Accepted: 14 September 2018

DOI: 10.1002/nme.5959

RESEARCH ARTICLE

WILEY

An isogeometric collocation method for efficient random field discretization

Ramin Jahanbin | Sharif Rahman

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

Correspondence
Sharif Rahman, College of Engineering, The University of Iowa, Iowa City, Iowa 52242.
Email: sharif-rahman@uiowa.edu

Funding information
U.S. National Science Foundation, Grant/Award Number: CMMI-1607398

Summary

This paper presents an isogeometric collocation method for a computationally expedient random field discretization by means of the Karhunen-Loève expansion. The method involves a collocation projection onto a finite-dimensional subspace of continuous functions over a bounded domain, basis splines (B-splines) and nonuniform rational B-splines (NURBS) spanning the subspace, and standard methods of eigensolutions. Similar to the existing Galerkin isogeometric method, the isogeometric collocation method preserves an exact geometrical representation of many commonly used physical or computational domains and exploits the regularity of isogeometric basis functions delivering globally smooth eigensolutions. However, in the collocation method, the construction of the system matrices for a d-dimensional eigenvalue problem asks for at most d-dimensional domain integrations, as compared with $2d$-dimensional integrations required in the Galerkin method. Therefore, the introduction of the collocation method for random field discretization offers a huge computational advantage over the existing Galerkin method. Three numerical examples, including a three-dimensional random field discretization problem, illustrate the accuracy and convergence properties of the collocation method for obtaining eigensolutions.

KEYWORDS

B-splines, Fredholm integral eigenvalue problem, Karhunen-Loève expansion, NURBS, uncertainty quantification

1 | INTRODUCTION

Computational methods for stochastic analysis of complex mechanical systems often call for random field description of uncertain media, geometry, and loads. In most applications, it is essential that a continuous-parameter random field be parameterized by a finite number of random variables. This process, designated as random field discretization, is usually aimed at striking a satisfactory trade-off between precision and computational efficiency. In other words, the resultant number of random variables must be sufficiently large to ensure an accurate representation of the random field, but, at the same time, the number must be tractable to subsequent stochastic analysis.

A frequently used approach for random field discretization is built on the Karhunen-Loève (KL) expansion.[1-3] The expansion generates approximations of random fields consisting of finite sums of deterministic functions of spatial arguments with random coefficients. The expansion is attractive for a few reasons: (i) it holds for both homogeneous and

inhomogeneous fields; (ii) it is optimal in the sense that the mean-square error committed by a finite-term KL approximation is minimized; and (iii) the sequence of KL approximations converges in mean-square to the correct limit. However, the construction of the KL expansion requires solution of a Fredholm integral eigenvalue problem. Analytical solutions are known when the computational domains are trivial, such as rectangular domains, and when the covariance functions are separable and described by simple functions.[4-6] In applications facing complex geometry, though, the domains are hardly simple, and the covariance functions are likely inseparable and cannot be characterized by elementary functions. Therefore, the eigensolutions must be obtained numerically and hence approximately.

A powerful gateway to numerical solution of the integral eigenvalue problem is the Galerkin projection method. Similar to solving partial differential equations, a weak or variational form of the eigenvalue problem is derived first and then the Galerkin discretization, associated with a finite-dimensional subspace of the function space of interest, is applied next, leading to a matrix eigenvalue problem. Depending on the subspaces, several variants of the Galerkin method, namely, the finite-element method (FEM),[7] mesh-free method,[8,9] and recently developed isogeometric method,[10] exist for solving the integral eigenvalue problem. The basis functions of these Galerkin methods are interpolatory polynomial, non-interpolatory rational functions, and basis splines (B-splines) or nonuniform rational B-splines (NURBS), respectively. Among them, the isogeometric method is known to faithfully reproduce the computer-aided design (CAD) geometry, especially the free-form surfaces and conic sections, whereas the same cannot be said about FEM or the mesh-free method. Nonetheless, all existing Galerkin methods suffer from a fundamental computational burden: the construction of a system matrix for a d-dimensional eigenvalue problem mandates a $2d$-dimensional domain integration. For instance, on a three-dimensional ($d = 3$) physical or computational domain, the system matrix requires a six-fold integration. As a result, the implementation of an industrial-scale matrix eigenvalue problem in high dimension is computationally intensive and likely prohibitive. Therefore, alternative projection methods avoiding the double integration, such as the collocation method and possibly others, should be pursued to solve the integral eigenvalue problem efficiently, but with an accuracy as close as possible to that of the Galerkin methods. Given the high-order continuity of B-splines and NURBS, a collocation method in conjunction with isogeometric analysis is expected to be favorable over traditional implementations. This is the principal motivation for this work.

This paper presents an isogeometric collocation method for solving the integral eigenvalue problem stemming from the KL expansion of a random field with a general square-integrable covariance function and a general tensor-product computational domain in one-to-three dimensions. The method involves a collocation projection onto a finite-dimensional isogeometric subspace of a Hilbert space, formulation of the associated matrix eigenvalue problem by constructing the isogeometric function space spanned by B-splines and NURBS, and solution of the resultant matrix eigenvalue problem by standard methods. The paper is structured as follows. A brief exposition of NURBS paraphernalia and isogeometric concept is given in Section 2. Section 3 formally defines a random field and its KL expansion, followed by truncation of the KL expansion and a description of associated error measures. Section 4 presents the proposed isogeometric method for solving the integral eigenvalue problem. The construction of system matrices involved in the matrix eigenvalue problem is explained. The results from three numerical examples of increasing dimensions and hence complexity are reported in Section 5 and the appendix. Section 6 discusses the effectiveness of the isogeometric collocation method and future work. Finally, conclusions are drawn in Section 7.

2 | ISOGEOMETRIC ANALYSIS

Let $\mathbb{N} := \{1, 2, \cdots\}$, $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$, $\mathbb{R} := (-\infty, +\infty)$, $\mathbb{R}_0^+ := [0, +\infty)$, and $\mathbb{R}^+ := (0, +\infty)$ represent the sets of positive integer (natural), nonnegative integer, real, nonnegative real, and positive real numbers, respectively. Denote by d the dimension of the physical or computational domain $\mathcal{D}$ of a geometrical object, which can be a curve, surface, and solid in $\mathbb{R}^d$. In this work, $d = 1, 2, 3$, and $\mathcal{D} \subset \mathbb{R}^d$ is a closed bounded set. These standard notations will be used throughout the paper.

The isogeometric analysis (IGA) employs basis functions from CAD, such as B-splines and NURBS, directly in computational analysis. Hughes et al[11] were the first to propose the isogeometric paradigm and its computational framework. For the paper to be self-contained, a brief summary of NURBS-based IGA is presented here. The description is restricted to geometries modeled as a single patch. However, for NURBS-based IGA, it is sometimes necessary to represent the physical or computational domain by a multipatch geometric model, for example, when analyzing multiply-connected domains. The multipatch geometries were not considered in this work.

2.1 | Knot vectors

Consider a d-dimensional Cartesian coordinate system in the parametric domain $\hat{D} = [0,1]^d$, where an arbitrary point has coordinate $\boldsymbol{\xi} = (\xi_1, \dots, \xi_d)$. For the coordinate direction k, where $k = 1, \dots, d$, define a positive integer $n_k \in \mathbb{N}$ and a nonnegative 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

$$\boldsymbol{\Xi}_k := (0 = \xi_{k,1}, \xi_{k,2}, \dots, \xi_{k,n_k+p_k+1} = 1), \ \xi_{k,1} \le \xi_{k,2} \le \cdots \le \xi_{k,n_k+p_k+1},$$

where ξ_{k,i_k} is the i_kth knot with $i_k = 1, 2, \dots, n_k + p_k + 1$ representing the knot index for the coordinate direction k. Although not absolutely necessary, assume that $\xi_{k,1} = 0$ and $\xi_{k,n_k+p_k+1} = 1$ for any k, so that all parametric intervals are the same as $[0,1]$. The knots may be equally spaced or unequally spaced, resulting in a uniform or nonuniform distribution. More importantly, the knots may be repeated, that is, a knot ξ_{k,i_k} of the knot vector $\boldsymbol{\Xi}_k$ may appear $1 \le m_{k,i_k} \le p_k + 1$ times, where m_{k,i_k} is referred to as its multiplicity. The multiplicity has important implications on the regularity properties of B-spline functions. To monitor knots without repetitions, say, there are r_k distinct knots in $\boldsymbol{\Xi}_k$. Collect them into an auxiliary knot vector $\mathbf{Z}_k := (\zeta_{k,1}, \dots, \zeta_{k,r_k})$ and define the vector $\mathbf{M}_k := (m_{k,1}, \dots, m_{k,r_k})$ of their corresponding multiplicities such that

$$\boldsymbol{\Xi}_k = (0 = \underbrace{\zeta_{k,1}, \dots, \zeta_{k,1}}_{m_{k,1} \text{ times}}, \underbrace{\zeta_{k,2}, \dots, \zeta_{k,2}}_{m_{k,2} \text{ times}}, \dots, \underbrace{\zeta_{k,r_k}, \dots, \zeta_{k,r_k}}_{m_{k,r_k} \text{ times}} = 1), \quad \sum_{i_k=1}^{r_k} m_{k,i_k} = n_k + p_k + 1.$$

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.[12]

2.2 | Univariate B-splines

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

$$N^k_{i_k,0}(\xi_k) = \begin{cases} 1, & \xi_{k,i_k} \le \xi_k < \xi_{k,i_k+1}, \\ 0, & \text{otherwise}, \end{cases}$$

for $k = 1, \dots, d$, all higher-order B-spline functions are efficiently generated by the recursive Cox-de Boor formula,[13,14]

$$N^k_{i_k,p_k}(\xi_k) = \frac{\xi_k - \xi_{k,i_k}}{\xi_{k,i_k+p_k} - \xi_{k,i_k}} N^k_{i_k,p_k-1}(\xi_k) + \frac{\xi_{k,i_k+p_k+1} - \xi_k}{\xi_{k,i_k+p_k+1} - \xi_{k,i_k+1}} N^k_{i_k+1,p_k-1}(\xi_k), \tag{1}$$

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

The B-spline functions for any $k = 1, \dots, d$ and $p_k \in \mathbb{N}_0$ satisfy the following desirable properties[11-14]: (i) they are nonnegative, that is, $N^k_{i_k,p_k}(\xi_k) \ge 0$ for all i_k and ξ_k; (ii) they are locally supported on the interval $[\xi_{k,i_k}, \xi_{k,i_k+p_k+1}]$ for all i_k; (iii) they are linearly independent, that is, if $\sum_{i_k=1}^{n_k} c^k_{i_k} N^k_{i_k,p_k}(\xi_k) = 0$, then $c^k_{i_k} = 0$ for all i_k; (iv) they form a partition of unity, that is, $\sum_{i_k=1}^{n_k} N^k_{i_k,p_k}(\xi_k) = 1$, $\xi_k \in [\xi_{k,1}, \xi_{k,n_k+p_k+1}]$; and (v) they are everywhere pointwise C^∞-continuous except at the knots ξ_{k,i_k} of multiplicity m_{k,i_k}, where it is $C^{p_k - m_{k,i_k}}$-continuous, provided that $1 \le m_{k,i_k} < p_k + 1$.

Define by

$$\mathcal{N}_k := \mathcal{N}_k(\boldsymbol{\Xi}_k; p_k) := \text{span}\left\{ N^k_{i_k,p_k}(\xi_k) \right\}_{i_k=1,\dots,n_k}$$

the space of univariate B-splines with degree p_k for the coordinate direction k. Figure 1A shows eight univariate cubic B-spline basis functions $N^1_{i_1,p_1}(\xi_1)$, $i_1 = 1, \dots, n_1$, when $n_1 = 8$, $p_1 = 3$, and $\boldsymbol{\Xi}_1 = \{0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1\}$. The multiplicity of each interior knot is one. Therefore, the basis functions are C^2-continuous at all interior nodes. Clearly, the regularities of B-splines depend on the multiplicities of the knots selected.

*The nouns *degree* and *order* associated with IGA are used synonymously in the paper.

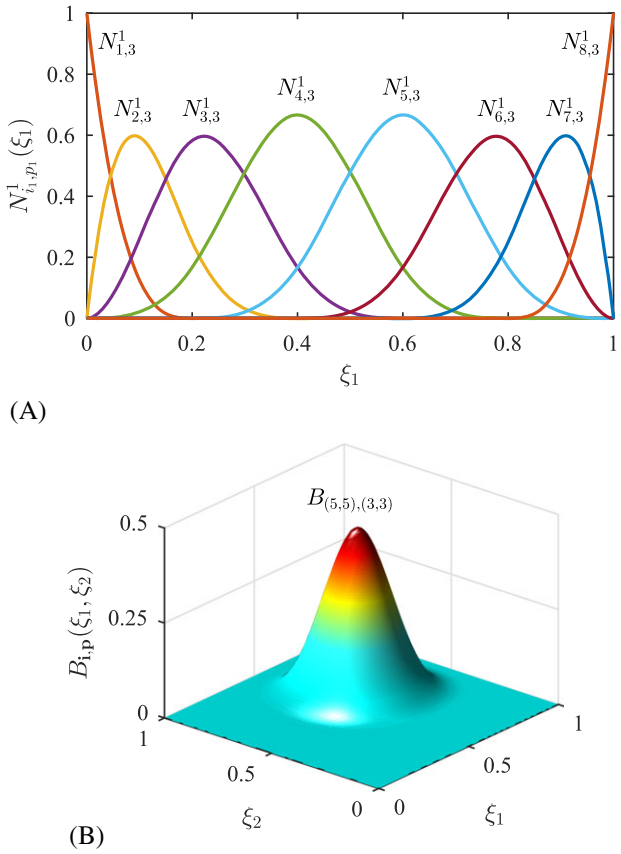

FIGURE 1 Cubic B-splines generated from the knot vectors $\Xi_1 = \Xi_2 = (0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1)$ with $n_1 = n_2 = 8$ and $p_1 = p_2 = 3$. A, Eight univariate B-splines for the coordinate direction ξ_1; B, A bivariate B-spline from the tensor product of $N^1_{5,3}(\xi_1)$ and $N^2_{5,3}(\xi_2)$ [Colour figure can be viewed at wileyonlinelibrary.com]

2.3 | Multivariate B-splines

The multivariate B-splines in d variables $\xi_1, \dots, \xi_d$ are constructed from the tensor product of the univariate B-splines stemming from the chosen knot vectors $\Xi_1, \dots, \Xi_d$. The corresponding auxiliary knot vectors and multiplicity vectors are $\mathbf{Z}_1, \dots, \mathbf{Z}_d$ and $\mathbf{M}_1, \dots, \mathbf{M}_d$, respectively. A mesh $\mathcal{Q}_h$ in the parametric domain $\hat{D} = [0,1]^d$ is defined by its partition into d-dimensional parametric elements Q, that is,

$$\mathcal{Q}_h := \left\{ Q = \otimes_{k=1}^{d} \left(\zeta_{k,i_k}, \zeta_{k,i_k+1} \right) \,:\, 1 \le i_k < r_k - 1 \right\}.$$

Relatedly, if the size of an element $Q \in \mathcal{Q}_h$ is defined as $\hat{h}_Q := \mathrm{diam}(Q)$, then $\hat{h} := \max_{Q \in \mathcal{Q}_h} \{\hat{h}_Q\}$ defines the global mesh size in the parametric domain.

Define two multi-indices $\mathbf{i} := (i_1, \dots, i_d)$ and $\mathbf{p} := (p_1, \dots, p_d)$. For the first multi-index, denote by

$$\mathcal{I} := \{\mathbf{i} = (i_1, \cdots, i_d) \,:\, 1 \le i_k \le n_k,\ 1 \le k \le d\}$$

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

$$B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) := \prod_{k=1}^{d} N^k_{i_k,p_k}(\xi_k) \tag{2}$$

with the corresponding tensor-product B-spline space

$$\mathcal{B}_h := \bigotimes_{k=1}^{d} \mathcal{N}_k(\Xi_k; p_k) = \bigotimes_{k=1}^{d} \mathrm{span}\left\{ N^k_{i_k,p_k}(\xi_k) \right\}_{i_k=1,\dots,n_k} = \mathrm{span}\{B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})\}_{\mathbf{i} \in \mathcal{I}}. \tag{3}$$

Note that the functions in $\mathcal{B}_h$ are piecewise polynomials of degree p_k along each coordinate direction $k = 1, \dots, d$. Figure 1B depicts a bivariate cubic B-spline, which is generated from the knot vectors $\Xi_1 = \Xi_2 = (0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1)$ and tensor product of $N^1_{5,3}(\xi_1)$ and $N^2_{5,3}(\xi_2)$.

Due to the tensor-product structure, multivariate B-spline functions inherit most of the aforementioned properties of their univariate counterparts, namely, nonnegativity, local support, linear independence, partition of unity, and regularity. The functions are C^∞-continuous in the interior of each element $Q \in \mathcal{Q}_h$, while, across element boundaries, the regularity is decided by the directional regularity in each coordinate.

2.4 | Nonuniform rational B-spline (NURBS)

With the multivariate B-spline functions and their space described by (2) and (3), the multivariate NURBS functions and the corresponding space can now be defined using a projective transformation.[12,15] Associated with each $\mathbf{i} \in \mathcal{I}$, denote by $w_\mathbf{i} \in \mathbb{R}^+$ a constant positive weight. As a result, the weight function $w : \hat{\mathcal{D}} \to \mathbb{R}$ can be defined as

$$w(\boldsymbol{\xi}) := \sum_{\mathbf{i}\in\mathcal{I}} w_\mathbf{i} B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}).$$

Using the properties of B-splines, it is elementary to show that the weight function is also positive. Given $\mathbf{i} \in \mathcal{I}$ and $\mathbf{p} \in \mathbb{N}_0^d$, the multivariate NURBS function $R_{\mathbf{i},\mathbf{p}} : \hat{\mathcal{D}} \to \mathbb{R}$ is defined as[12,15]

$$R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) := \frac{w_\mathbf{i} B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})}{w(\boldsymbol{\xi})} = \frac{w_\mathbf{i} B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})}{\sum_{\mathbf{i}\in\mathcal{I}} w_\mathbf{i} B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})},$$

producing the NURBS function space

$$\mathcal{R}_h := \operatorname{span}\{R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})\}_{\mathbf{i}\in\mathcal{I}} \tag{4}$$

on the parametric domain $\hat{\mathcal{D}}$.

The NURBS functions described in the preceding inherit all of the important properties from their piecewise polynomial counterparts as follows[11]: (i) they constitute a partition of unity; (ii) the NURBS and B-splines functions have the same continuity and support; (iii) they possess the property of affine transformations; (iv) setting all weights to be equal, a NURBS function reduces to a scaled B-spline function; and (v) the NURBS surfaces and solids are projective transformations of tensor-product, piecewise polynomial entities.

Using multivariate B-splines and NURBS functions, a geometric object in $\mathbb{R}^d$, such as a curve, surface, or solid, can be readily generated. For each $\mathbf{i} \in \mathcal{I}$, let $\mathbf{C}_\mathbf{i} \in \mathbb{R}^d$ be a control point. Denote by $n_c := |\mathcal{I}|$ the cardinality of $\mathcal{I}$, representing the number of such control points. Call the collection of such control points $\{\mathbf{C}_\mathbf{i}\}_{\mathbf{i}\in\mathcal{I}}$ to be a control mesh. Using NURBS functions, the physical domain $\mathcal{D} \subset \mathbb{R}^d$ is obtained by a geometrical mapping $\mathbf{x} : \hat{\mathcal{D}} \to \mathcal{D} \subset \mathbb{R}^d$, which is described more explicitly by

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{\mathbf{i}\in\mathcal{I}} R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) \mathbf{C}_\mathbf{i}. \tag{5}$$

A similar mapping can be defined using multivariate B-spline functions. However, not all objects or domains, some of which are commonly used in engineering, can be represented by B-splines. For instance, free-form surfaces and conic sections, such as circles, ellipses, cylinders, spheres, ellipsoids, and tori, cannot be described by piecewise polynomials. In contrast, NURBS functions equipped with judiciously selected weights can represent them exactly.[12,15] Therefore, the use of NURBS, that is, (5), becomes necessary in the CAD community.

Using the geometrical mapping (5), the physical mesh $\mathcal{K}_h$, say, of the physical domain can now be viewed as the image of the parametric mesh $\mathcal{Q}_h$, that is,

$$\mathcal{K}_h := \{K = \mathbf{x}(Q) : Q \in \mathcal{Q}_h\},$$

where the element K of the physical mesh is the image of the element Q of the parametric mesh. Moreover, define the space of NURBS functions in the physical domain $\mathcal{D}$ as the push-forward of the NURBS space $\mathcal{R}_h$ in (4) via

$$\mathcal{V}_h := \operatorname{span}\left\{R_{\mathbf{i},\mathbf{p}} \circ \mathbf{x}^{-1}\right\}_{\mathbf{i}\in\mathcal{I}} = \operatorname{span}\left\{\bar{R}_{\mathbf{i},\mathbf{p}}\right\}_{\mathbf{i}\in\mathcal{I}}, \tag{6}$$

where $\bar{R}_{\mathbf{i},\mathbf{p}} := R_{\mathbf{i},\mathbf{p}} \circ \mathbf{x}^{-1}$ is the NURBS function in the physical domain. It is assumed that the mapping (5) is invertible *almost everywhere* in $\mathcal{D}$ and has smooth inverse on each element K of the physical mesh $\mathcal{K}_h$.

2.5 | Refinement

The accuracy of IGA depends on the enrichment of the NURBS spaces $\mathcal{R}_h$ and $\mathcal{V}_h$ in (4) and (6) via refinement. There are several types of refinement. A simple and straightforward type, namely knot insertion, is equivalent to h-refinement commonly used in FEM. For knot insertion, a finer mesh is constructed by adding knots to the existing knot vectors

without changing the geometry. As an example, consider inserting a new knot $\xi'_k \in [\xi_{k,l}, \xi_{k,l+1})$, $1 \leq l \leq n_k + p_k$, to the existing knot vector $\Xi_k := (\xi_{k,1}, \xi_{k,2}, \ldots, \xi_{k,n_k+p_k+1})$, which produces n_k B-spline functions. Applying the Cox-de Boor formula (1) to the new knot vector, say,

$$\Xi'_k := \left(\xi'_{k,1}, \xi'_{k,2}, \ldots, \xi'_{k,n_k+p_k+2}\right) = \left(\xi_{k,1}, \xi_{k,2}, \ldots, \xi_{k,l}, \xi'_k, \xi_{k,l+1}, \ldots, \xi_{k,n_k+p_k+1}\right),$$

a new set of $n_k + 1$ basis functions is created with their span nesting the span of existing basis functions. The process can be repeated for additional knots. Moreover, the h-refinement can be performed globally in all d coordinate directions or individually in select coordinate directions. Henceforth, for a NURBS object in $\mathbb{R}^d$, a new set of control points should be defined for the new basis functions to obtain an object that is geometrically and parametrically the same as the original one. In other words, the geometry of the physical domain is preserved. Through knot insertion or h-refinement, there are increases in the number of elements as well as in the number of basis functions and, consequently, in the number of control points.

Readers interested in further details of the h-refinement, including a few other alternatives, are referred to the book by Cottrell et al.[16]

2.6 | Theoretical analysis of NURBS

Let $\mathcal{X}$ be a Banach space and a function $u(\mathbf{x}) \in \mathcal{X}$ be defined on a closed bounded set $\mathcal{D}$ in $\mathbb{R}^d$. The most popular choices of $\mathcal{X}$ are the space $C(\mathcal{D})$ of continuous functions with the uniform or supremum norm $||\cdot||_\infty$, the Hilbert space $L^2(\mathcal{D})$ of real-valued square-integrable functions with the norm $||\cdot||_{L^2(\mathcal{D})}$, and the Sobolev space $H^l(\mathcal{D})$ of order l, $l \in \mathbb{N}_0$ equipped with appropriate norm and seminorms. The space $C(\mathcal{D})$ is commonly used in the collocation method, whereas the spaces $L^2(\mathcal{D})$ and $H^l(\mathcal{D})$ are employed in the Galerkin method.[17]

For error estimates of NURBS-based IGA, consider interpolating a function $u \in \mathcal{X}$, where $\mathcal{X}$ is $C(\mathcal{D})$, $L^2(\mathcal{D})$, or $H^l(\mathcal{D})$. However, the decay of interpolation error depends on how the NURBS space is refined. Bazilevs et al[18] studied approximation and stability properties of NURBS-based IGA in the context of h-refinement. Using the celebrated Bramble-Hilbert lemma[19] in bent Sobolev spaces, they developed error estimates similar to those for finite-element analysis. In particular, Theorem 1 of the aforementioned work proves convergence of NURBS-based function interpolations when $u \in L^2(\mathcal{D})$. Since $C(\mathcal{D})$ is a dense subspace of $L^2(\mathcal{D})$, the NURBS-based function interpolations should also converge for any $u \in C(\mathcal{D})$.

3 | KARHUNEN-LOÈVE REPRESENTATION

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, where Ω is a sample space, $\mathcal{F}$ is a σ-field on Ω, and $P : \mathcal{F} \to [0,1]$ is a probability measure. Denote by $L^2(\Omega, \mathcal{F}, \mathbb{P})$ a Hilbert space of random variables defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and by $L^2(\mathcal{D} \times \Omega)$ a Hilbert space of random fields defined on $\mathcal{D}$. A random variable or random field, if it is a member of the associated Hilbert space, has finite second-moment properties.

3.1 | Random field

A real-valued random field α defined on a closed bounded domain $\mathcal{D} \subset \mathbb{R}^d$, where $d = 1$, 2, or 3, is a mapping $\alpha : \mathcal{D} \times \Omega \to \mathbb{R}$ such that for each $\mathbf{x} \in \mathcal{D}$, $\alpha(\mathbf{x}, \cdot)$ is a random variable with respect to $(\Omega, \mathcal{F}, \mathbb{P})$. Given the expectation operator $\mathbb{E}$ with respect to the probability measure $\mathbb{P}$, denote by $\mu(\mathbf{x}) := \mathbb{E}[\alpha(\mathbf{x}, \cdot)]$ the mean function and by

$$\Gamma(\mathbf{x}, \mathbf{x}') := \mathbb{E}[(\alpha(\mathbf{x}, \cdot) - \mu(\mathbf{x}))(\alpha(\mathbf{x}', \cdot) - \mu(\mathbf{x}'))], \quad \mathbf{x}, \mathbf{x}' \in \mathcal{D},$$

the covariance function of $\alpha(\mathbf{x}, \cdot)$. Without loss of generality, assume that $\mu(\mathbf{x}) = 0$. A non-*zero*-mean random field can be obtained by just adding the mean function to a *zero*-mean random field.

The random fields are often assumed to be homogeneous or stationary, meaning that their finite-dimensional probability distributions are invariant under arbitrary translations. This implies that the covariance function is a function of the argument difference $\mathbf{x} - \mathbf{x}'$. Moreover, random fields are sometimes assumed to be isotropic, that is, invariant under orthogonal transformations. In such a case, the covariance function becomes a function of the distance $\|\mathbf{x} - \mathbf{x}'\|$.

3.2 | Karhunen-Loève expansion

Let $\alpha(\mathbf{x},\cdot) \in L^2(\mathcal{D}\times\Omega)$ be a real-valued random field with *zero* mean, continuous covariance function $\Gamma : \mathcal{D}\times\mathcal{D} \to \mathbb{R}$, and associated Hilbert-Schmidt operator $\mathcal{G}_\Gamma : L^2(\mathcal{D}) \to L^2(\mathcal{D})$ defined by[20]

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

It is well-known that $\mathcal{G}_\Gamma : L^2(\mathcal{D}) \to L^2(\mathcal{D})$ is a linear, compact, positive-semidefinite, and self-adjoint operator. There is an infinite sequence of eigenpairs $\{\lambda_i,\phi_i(\mathbf{x})\}_{i\in\mathbb{N}}$ of $\mathcal{G}_\Gamma$, which is the solution of

$$(\mathcal{G}_\Gamma\phi)(\mathbf{x}) = \lambda\phi(\mathbf{x}) \text{ or } \int_{\mathcal{D}} \Gamma(\mathbf{x},\mathbf{x}')\phi(\mathbf{x}')\mathrm{d}\mathbf{x}' = \lambda\phi(\mathbf{x}), \tag{7}$$

known as the Fredholm integral equation of the second kind. Moreover, the eigensolutions, where the eigenfunctions have been normalized such that $\|\phi_i(\mathbf{x})\|^2_{L^2(\mathcal{D})} := \int_{\mathcal{D}}\phi_i^2(\mathbf{x})\mathrm{d}\mathbf{x} = 1$, have the following properties: (i) the eigenvalues $\lambda_i \in \mathbb{R}_0^+$, $i \in \mathbb{N}$, are real and nonnegative having zero as the only point of accumulation; (ii) the eigenfunctions $\phi_i(\mathbf{x}) \in L^2(\mathcal{D})$, $i \in \mathbb{N}$, corresponding to distinct eigenvalues, are mutually orthonormal; (iii) the number of eigenfunctions corresponding to nonzero eigenvalues is finite; and (iv) the sequence of eigenfunctions $\{\phi_i(\mathbf{x})\}_{i\in\mathbb{N}}$ forms an orthonormal basis of $L^2(\mathcal{D})$, that is, $L^2(\mathcal{D}) = \mathrm{span}\{\phi_i(\mathbf{x})\}_{i\in\mathbb{N}}$.

Given an infinite sequence of eigenpairs $\{\lambda_i,\phi_i(\mathbf{x})\}_{i\in\mathbb{N}}$ of $\mathcal{G}_\Gamma$, the random field admits a convergent infinite series expansion

$$\alpha(\mathbf{x},\cdot) \sim \sum_{i=1}^{\infty}\sqrt{\lambda_i}\phi_i(\mathbf{x})X_i, \tag{8}$$

where $\{X_i\}_{i\in\mathbb{N}}$ is an infinite sequence of *zero*-mean, standardized, uncorrelated random variables, that is,

$$\mathbb{E}[X_i] = \int_\Omega X_i(\omega)\mathrm{d}\mathbb{P}(\omega) = 0,$$

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

with each random variable X_i defined, for $\lambda_i \neq 0$, as

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

The infinite series on the right hand side of (8) converges in mean-square to the correct limit and is commonly referred to as the KL expansion.

The KL expansion, originally developed by Karhunen and Loève,[1-3] enjoys widespread use in many applied and pure sciences, including solid mechanics, fluid dynamics, physics, biology, signal processing, to name a few. There are a multitude of books and papers on the KL expansion. Readers interested in further details should consult some of these works.[21-24]

3.3 | Karhunen-Loève approximation

The KL expansion contains an infinite number of eigenpairs and random variables. In practice, the number must be finite, meaning that the expansion must be truncated. A straightforward approach, assuming that the eigenvalues have been arranged in a descending sequence, entails retaining the first $N \in \mathbb{N}$ terms of the expansion. The result is an N-term truncation or KL approximation

$$\alpha_N(\mathbf{x},\cdot) = \sum_{i=1}^{N}\sqrt{\lambda_i}\phi_i(\mathbf{x})X_i \tag{9}$$

of $\alpha(\mathbf{x},\cdot)$, comprising eigenpairs $\{\lambda_i,\phi_i(\mathbf{x})\}_{1\le i\le N}$ and random variables X_i, $i = 1,\ \ldots\ ,N$. This is commonly referred to as random field discretization. In consequence, the statistical variation of random field $\alpha(\mathbf{x},\cdot)$ is being swapped with those possessed by N uncorrelated random variables $X_1,\ \ldots\ ,X_N$. Therefore, the value of N should be selected judiciously not only for maintaining desired accuracy in the discretization, but also for computational expediency.

A second-moment error analysis facilitates a simple way of evaluating the quality of a KL approximation. Indeed, given the N-term truncation in (9) and the variance $\sigma^2(\mathbf{x}) = \Gamma(\mathbf{x},\mathbf{x})$ of $\alpha(\mathbf{x},\cdot)$, a global error measure reads[10]

$$\tilde{e}_N = 1 - \frac{1}{|\mathcal{D}|}\sum_{i=1}^{N}\lambda_i\int_{\mathcal{D}}\frac{\phi_i^2(\mathbf{x})}{\sigma^2(\mathbf{x})}\mathrm{d}\mathbf{x}, \tag{10}$$

where $|\mathcal{D}| := \int_{\mathcal{D}} d\mathbf{x}$ is the Lebesgue measure, such as length for $d = 1$, area for $d = 2$, or volume for $d = 3$, of $\mathcal{D}$. According to (10), the truncated KL expansion always underestimates the variance of the original random field. Given a value of N, the effectiveness of an N-term KL approximation is predicated on how fast the eigenvalues decay with respect to the eigenmodes. The rate of decay depends strongly on the properties of the covariance function, especially the correlation length parameter of the covariance function. A remarkable property of the KL expansion is its error-minimizing property; that is, given a fixed N, the KL approximation in (9) has been proven to be optimal among all series expansion methods with respect to a global mean-square error.[7]

3.4 | Remarks

If α is a Gaussian random field, then X_i, $i = 1, \ldots, N$, in (9) are independent standard Gaussian random variables. In consequence, each element of the sequence $\{\alpha_N\}_{N\in\mathbb{N}}$ is a Gaussian random field. However, if α is a non-Gaussian random field, then X_i, $i = 1, \ldots, N$, are uncorrelated yet dependent non-Gaussian random variables. In this case, finding their probability distribution is not trivial. One class of non-Gaussian random fields for which the use of KL expansion can possibly be exploited is the class of translation random fields, where a non-Gaussian random field is defined as a nonlinear, memoryless transformation of a Gaussian random field.[25] However, even then, there are conditions on the covariance properties that must be fulfilled before proceeding with the transformation.[9,25]

4 | ISOGEOMETRIC COLLOCATION METHOD

The KL approximation for random field discretization is grounded on the knowledge of eigensolutions of the integral eigenvalue problem defined by (7). However, analytical or exact solutions of the eigenvalue problem exist when the covariance function $\Gamma : \mathcal{D}\times\mathcal{D} \to \mathbb{R}$ is separable and has simpler functional forms, such as exponential functions, or the domain $\mathcal{D}$ is rectangular. For general covariance functions or domains, numerical methods are often needed to solve the eigenvalue problem. In this section, the basis functions from isogeometric analysis, such as B-splines and NURBS, in conjunction with the collocation projection, are employed to solve the eigenvalue problem.

Let $C(\mathcal{D})$ be the space of continuous functions on $\mathcal{D}$ with the supremum or uniform norm $||\cdot||_\infty$. Recognize that $\mathcal{G}_\Gamma$ is a compact operator from $C(\mathcal{D})$ to $C(\mathcal{D})$. Denote by $\{\mathcal{S}_h\}_{h>0}$ a sequence of finite-dimensional approximating subspaces of $C(\mathcal{D})$. Replace λ and $\phi(\mathbf{x})$ in (7) with $\lambda_h \in \mathbb{R}_0^+$ and $\phi_h(\mathbf{x}) \in \mathcal{S}_h$, respectively, to define a residual

$$r_h := (\mathcal{G}_\Gamma\phi_h)(\mathbf{x}) - \lambda_h\phi_h(\mathbf{x}) = \int_{\mathcal{D}} \Gamma(\mathbf{x},\mathbf{x}')\phi_h(\mathbf{x}')\mathrm{d}\mathbf{x}' - \lambda_h\phi_h(\mathbf{x}). \tag{11}$$

Denote by d_h the dimension of $\mathcal{S}_h$ and by $\{\psi_1(\mathbf{x}), \ldots, \psi_{d_h}(\mathbf{x})\}$ a basis of $\mathcal{S}_h$. Therefore, the function $\phi_h(\mathbf{x}) \in \mathcal{S}_h$ can be written as

$$\phi_h(\mathbf{x}) = \sum_{j=1}^{d_h} c_j\psi_j(\mathbf{x}), \quad \mathbf{x}\in\mathcal{D}, \tag{12}$$

with $c_j \in \mathbb{R}, j = 1, \ldots, d_h$, representing the associated coefficients. When (12) is substituted into (11), the result is

$$r_h = \sum_{j=1}^{d_h} c_j \left[\int_{\mathcal{D}} \Gamma(\mathbf{x},\mathbf{x}')\psi_j(\mathbf{x}')\mathrm{d}\mathbf{x}' - \lambda_h\psi_j(\mathbf{x})\right], \ \mathbf{x}\in\mathcal{D}. \tag{13}$$

4.1 | Collocation projection

The collocation projection method for solving the Fredholm integral Equation (7) entails finding an eigenpair $\{\lambda_h, \phi_h(\mathbf{x})\} \subset \mathbb{R}_0^+ \times \mathcal{S}_h$ such that the residual r_h vanishes at select distinct points, referred to as collocation points, of $\mathcal{D}$. Pick a set of distinct points $\mathbf{x}_i \in \mathcal{D}$ such that

$$\det\left[\psi_j(\mathbf{x}_i)\right] \neq 0, \quad i,j = 1, \ldots, d_h. \tag{14}$$

Note that both the basis functions ψ_j and the collocation points $\mathbf{x}_i$ depend on h, but for notational simplicity the dependence is suppressed. Evaluating the residual r_h from (13) at the collocation points and then setting them equal to *zero*

yields a linear $d_h \times d_h$ matrix eigenvalue problem

$$\sum_{j=1}^{d_h} c_j \left[\int_{\mathcal{D}} \Gamma(\mathbf{x}_i, \mathbf{x}')\psi_j(\mathbf{x}')\mathrm{d}\mathbf{x}' - \lambda_h \psi_j(\mathbf{x}_i) \right] = 0, \quad i, j = 1, \ldots, d_h. \tag{15}$$

The solution of (15) determines the expansion coefficients $c_j, j = 1, \ldots, d_h$, and hence finds the eigenpair $\{\lambda_h, \phi_h(\mathbf{x})\}$ relative to the subspace $\mathcal{S}_h$. The solution exists and is unique, provided that the condition (14) is upheld. In general, the solution of (15) is an approximate solution of (7).

For convergence analysis, place the collocation method in a more abstract form by introducing the interpolatory projection operator $P_h : C(\mathcal{D}) \to \mathcal{S}_h$. In other words, given $\phi(\mathbf{x}) \in C(\mathcal{D})$, $P_h\phi(\mathbf{x})$ is the element of $\mathcal{S}_h$ that interpolates $\phi(\mathbf{x})$ at the collocation points $\{\mathbf{x}_1, \ldots, \mathbf{x}_{d_h}\}$. As explained by Atkinson,[17] P_h is a bounded linear operator on $C(\mathcal{D})$, has norm

$$||P_h|| = \sup_{\mathbf{x}\in\mathcal{D}} \sum_{j=1}^{d_h} |\psi_j(\mathbf{x})|,$$

and is an idempotent operator, that is, $P_h^2 = P_h$. Then the collocation Equation (15) becomes equivalent to

$$(P_h \mathcal{G}_\Gamma \phi_h)(\mathbf{x}) = \lambda_h \phi_h(\mathbf{x}).$$

The next step is to ensure that

$$\lim_{h\to 0} ||P_h\mathcal{G}_\Gamma - \mathcal{G}_\Gamma|| = 0. \tag{16}$$

Since $\mathcal{G}_\Gamma$ is compact on $C(\mathcal{D})$, a sufficient condition for fulfilling (16) is

$$\lim_{h\to 0} ||P_h\phi(\mathbf{x}) - \phi(\mathbf{x})||_{L^\infty} = 0, \ \phi(\mathbf{x}) \in C(\mathcal{D}).$$

Once (16) is ensured, it then follows that $\|\phi(\mathbf{x}) - \phi_h(\mathbf{x})\|_\infty \to 0$ as $h \to 0$, demonstrating convergence of collocation solutions in the supremum norm. Moreover, the rate at which $\phi_h(\mathbf{x})$ converges to $\phi(\mathbf{x})$ is identical to the rate at which $P_h\phi(\mathbf{x})$ converges to $\phi(\mathbf{x})$.[17]

4.2 | Isogeometric approximation

The collocation projection discussed in the preceding subsection is general because the finite-dimensional subspaces $\{\mathcal{S}_h\}_{h>0}$ of $C(\mathcal{D})$ have yet to be specified. In this work, the subspaces derived from B-splines and NURBS functions, which are the building blocks of IGA, are proposed.

Proposition 1. *Let the functions $\bar{R}_{\mathbf{i},\mathbf{p}}(\mathbf{x})$, $\mathbf{i} \in \mathcal{I}$, of the NURBS space $\mathcal{V}_h$, defined in (6), be arranged according to $\bar{R}_{i,\mathbf{p}}(\mathbf{x})$, $i = 1, \ldots, n_c$, where $n_c = |\mathcal{I}|$, equal to the number of control points of IGA, is the dimension of $\mathcal{V}_h$. Select (i) the finite-dimensional subspaces of $C(\mathcal{D})$ as*

$$\mathcal{S}_h = \mathcal{V}_h = \mathrm{span}\left\{\bar{R}_{\mathbf{i},\mathbf{p}}\right\}_{\mathbf{i}\in\mathcal{I}} = \mathrm{span}\left\{\bar{R}_{i,\mathbf{p}}\right\}_{i=1}^{n_c} \subset C(\mathcal{D}), \ \ h > 0,$$

and (ii) distinct collocation points $\mathbf{x}_1, \ldots, \mathbf{x}_{n_c} \in \mathcal{D}$ such that the condition (14) is fulfilled. Then, relative to the subspace $\mathcal{V}_h$, the solution of the linear matrix eigenvalue problem,

$$\mathbf{A}\mathbf{f}_h = \lambda_h \mathbf{B}\mathbf{f}_h, \tag{17}$$

yields an eigenvalue λ_h and an eigenvector $\mathbf{f}_h$. Here, $\mathbf{A} \in \mathbb{R}^{n_c\times n_c}$ and $\mathbf{B} \in \mathbb{R}^{n_c\times n_c}$ are system matrices, which have components

$$A_{ij} := \int_{\mathcal{D}} \Gamma(\mathbf{x}_i, \mathbf{x}')\bar{R}_{j,\mathbf{p}}(\mathbf{x}')\mathrm{d}\mathbf{x}', \ i, j = 1, \ldots, n_c, \tag{18}$$

and

$$B_{ij} := \bar{R}_{j,\mathbf{p}}(\mathbf{x}_i), \ i, j = 1, \ldots, n_c. \tag{19}$$

Henceforth, the corresponding eigenfunction is obtained as

$$\phi_h(\mathbf{x}) = \sum_{j=1}^{n_c} f_{h,j}\bar{R}_{j,\mathbf{p}}(\mathbf{x}), \tag{20}$$

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

Proof. Since $\mathcal{S}_h = \mathcal{V}_h$, replace d_h, c_j, and $\psi_j(\mathbf{x})$ in (15) with n_c, $f_{h,j}$, and $\bar{R}_{j,\mathbf{p}}(\mathbf{x})$, respectively, yielding

$$\sum_{j=1}^{n_c} f_{h,j} \left[\int_{\mathcal{D}} \Gamma(\mathbf{x}_i, \mathbf{x}') \bar{R}_{j,\mathbf{p}}(\mathbf{x}') \mathrm{d}\mathbf{x}' - \lambda_h \bar{R}_{j,\mathbf{p}}(\mathbf{x}_i) \right] = 0, \quad i, j = 1, \ldots, n_c. \tag{21}$$

From the definitions of the system matrices $\mathbf{A}$ and $\mathbf{B}$ as in (18) and (19), (21) becomes

$$\sum_{j=1}^{n_c} A_{ij} f_{h,j} = \lambda_h \sum_{j=1}^{n_c} B_{ij} f_{h,j},$$

which is the same as (17). The expression of $\phi_h(\mathbf{x})$ in (20) follows readily. □

Does the isogeometric collocation method for solving the integral eigenvalue problem converge? The question can be readily answered using the work of Bazilevs et al,[18] which demonstrates that the sequence of isogeometric approximations using NURBS functions and h-refinement converges for any function on the Hilbert space and hence for $\phi(\mathbf{x}) \in C(\mathcal{D})$. Moreover, the collocation projection $P_h : C(\mathcal{D}) \to \mathcal{S}_h$ satisfies the condition (16). Therefore, the eigensolutions from the collocation method addressed in this work should converge as discussed in Section 4.1.

4.3 | Construction of system matrices

The solution of the matrix eigenvalue value problem described in Proposition 1 depends on the system matrices $\mathbf{A}$ and $\mathbf{B}$. Therefore, it is important to discuss the computational effort in constructing these matrices. While forming matrix $\mathbf{B}$ is trivial, assembling matrix $\mathbf{A}$ requires domain integration in the physical space. In general, the integral cannot be determined exactly. Therefore, the matrix must be estimated by numerical integration. However, the use of NURBS functions in isogeometric analysis introduces the parametric domain as explained in Section 2. This slightly complicates the matter because, for numerical integration, an additional domain $[-1, +1]^d$ is needed. The latter domain is commonly referred to as the parent element in the isogeometric literature.[16]

Consider an arbitrary element $Q \in \mathcal{Q}_h$ in the parametric domain $\hat{\mathcal{D}} = [0, 1]^d$. Each such element can be viewed as the image of the parent element $[-1, +1]^d$ defined by an affine mapping $\boldsymbol{\xi} : [-1, +1]^d \to Q$ or, equivalently, by $\boldsymbol{\xi}(\boldsymbol{\eta})$, where $\boldsymbol{\eta}$ is the coordinate of the parent element. Similarly, there is a corresponding element $K \in \mathcal{K}_h$ in the physical domain $\mathcal{D} \subset \mathbb{R}^d$, which is the image of that very element Q in the parametric domain. Recall that $\mathbf{x} : \hat{\mathcal{D}} \to \mathcal{D}$, that is, $\mathbf{x}(\boldsymbol{\xi})$, is the mapping between the parametric and physical domains. The same mapping is used for these two corresponding elements. To integrate a function on an element of the physical domain, a pullback of the physical element to the parent element is required. This is accomplished using the composition of the inverses, that is, $\mathbf{x}^{-1} : K \to Q$ and $\boldsymbol{\xi}^{-1} : Q \to [-1, +1]^d$, of the two aforementioned mappings.

Let the Jacobians of the mappings $\boldsymbol{\xi}(\boldsymbol{\eta})$ and $\mathbf{x}(\boldsymbol{\xi})$ be defined as $\mathbf{J}_{\boldsymbol{\eta}} := [\partial\boldsymbol{\xi}/\partial\boldsymbol{\eta}]$ and $\mathbf{J}_{\boldsymbol{\xi}} := [\partial\mathbf{x}/\partial\boldsymbol{\xi}]$, respectively. As the mapping is affine, the calculation of the partial derivatives $\partial\xi_i/\partial\eta_j$, $i, j = 1, \ldots, d$, is straightforward. However, to determine the partial derivatives $\partial x_i/\partial\xi_j$, $i, j = 1, \ldots, d$, the derivatives of NURBS and B-Spline functions are involved. Due to brevity, explicit details of the derivatives of NURBS and B-spline functions are not reported here as they are available elsewhere.[16]

Given the mappings and their respective Jacobians, the components of the system matrix $\mathbf{A}$ are then evaluated by summing contributions from all element-level integrations on the parent element, that is,

$$\begin{aligned} A_{ij} &:= \textstyle\int_{\mathcal{D}} \Gamma(\mathbf{x}_i, \mathbf{x}') \bar{R}_{j,\mathbf{p}}(\mathbf{x}') \mathrm{d}\mathbf{x}' \\ &= \sum_{K \in \mathcal{K}_h} \int_K \Gamma(\mathbf{x}_i, \mathbf{x}') \bar{R}_{j,\mathbf{p}}(\mathbf{x}') \mathrm{d}\mathbf{x}' \\ &= \sum_{Q \in \mathcal{Q}_h} \int_Q \Gamma\left(\mathbf{x}_i, \mathbf{x}'(\boldsymbol{\xi}')\right) \bar{R}_{j,\mathbf{p}}(\mathbf{x}'(\boldsymbol{\xi}'))| \det \mathbf{J}_{\boldsymbol{\xi}}| \mathrm{d}\boldsymbol{\xi}' \\ &= \sum \int_{[-1,+1]^d} \Gamma\left(\mathbf{x}_i, \mathbf{x}'(\boldsymbol{\xi}'(\boldsymbol{\eta}'))\right) \bar{R}_{j,\mathbf{p}}(\mathbf{x}'(\boldsymbol{\xi}'(\boldsymbol{\eta}')))| \det \mathbf{J}_{\boldsymbol{\xi}}|| \det \mathbf{J}_{\boldsymbol{\eta}'}| \mathrm{d}\boldsymbol{\eta}'. \end{aligned} \tag{22}$$

Here, the summation in the last line of (22) is over all $n_e := |\mathcal{Q}_h| = |\mathcal{K}_h|$ elements of IGA. The final integral in (22) is estimated by a suitable numerical integration scheme, such as the Gauss-Legendre quadrature. Even though the NURBS functions are not necessarily polynomials, the Gauss quadrature is still effective.[16] In this case, the same quadrature rule employed for a pth-order polynomial can be used for a NURBS function built from an underlying pth-order B-spline.

Having said so, the Gauss quadrature is not an optimal choice for IGA. That is why current research is focused on finding optional or near-optimal numerical integration techniques to tackle NURBS functions.[26]

It is important to contrast the system matrices from the isogeometric collocation method with those from the Galerkin isogeometric method.[10] In the Galerkin method, there exists a matrix eigenvalue problem similar to (17), where the associated system matrices, denoted by $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ in this paper, have their components[10]

$$\bar{A}_{ij} := \int_{\mathcal{D}}\int_{\mathcal{D}} \Gamma(\mathbf{x},\mathbf{x}')\bar{R}_{i,\mathbf{p}}(\mathbf{x})\bar{R}_{j,\mathbf{p}}(\mathbf{x}')\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{x}',\ i,j=1,\ \dots,n_c, \tag{23}$$

and

$$\bar{B}_{ij} := \int_{\mathcal{D}} \bar{R}_{i,\mathbf{p}}(\mathbf{x})\bar{R}_{j,\mathbf{p}}(\mathbf{x})\mathrm{d}\mathbf{x},\ i,j=1,\ \dots,n_c. \tag{24}$$

Compared with matrix $\mathbf{A}$ in (18), matrix $\bar{\mathbf{A}}$ in (23) mandates a $2d$-dimensional domain integration, a fundamental prerequisite of the Galerkin method. While forming $\mathbf{A}$ requires an effort similar to that of assembling the mass matrix in solid mechanics, building $\bar{\mathbf{A}}$ is computationally arduous as it is d-order more expensive than forming $\mathbf{A}$. For instance, on a three-dimensional ($d = 3$) domain, $\bar{\mathbf{A}}$ requires a six-fold integration as opposed to a three-fold integration needed by $\mathbf{A}$. Furthermore, an additional d-dimensional domain integration is needed to assemble matrix $\bar{\mathbf{B}}$ in (24), where no such integration is involved in (19) to form matrix $\mathbf{B}$. Therefore, the collocation method proposed is expected to deliver a hefty computational advantage over the Galerkin method in solving the resultant matrix eigenvalue problem.

It is important to note that the double integration in (23) needs to be performed over the intersection of the supports of $\bar{R}_{i,\mathbf{p}}(\mathbf{x})$ and $\bar{R}_{j,\mathbf{p}}(\mathbf{x}')$. The local support representing such intersection is much smaller than the entire computational domain. The mismatch between the local support and computational domain becomes more pronounced when the mesh size is very large, eroding the computational advantage of the collocation method over the Galerkin method. This topic will be further examined in Section 5, where the numerical results are presented. Nonetheless, the collocation method proposed is still expected to be substantially more efficient than the Galerkin method.

4.4 | Collocation points

According to (17), the matrix eigenvalue problem is completely defined once the collocation points have been selected. Such a selection is vitally important, because it determines the stability and convergence properties of the collocation method. Guided by the CAD literature[27,28] and the experience from the computational mechanics community,[29,30] two types of collocation points are employed as follows.

4.4.1 | Greville abscissae

A widely adopted strategy for selecting the collocation points entails the images of so-called Greville abscissae.[27] Along the kth coordinate direction of the parametric domain $\hat{\mathcal{D}} = [0,1]^d$, $k = 1,\ \dots,d$, the Greville abscissae $\bar{\xi}_{k,i_k}$, $i_k = 1,\ \dots,n_k$, are n_k points defined from the knot vector as

$$\bar{\xi}_{k,i_k} := \frac{1}{p_k}\sum_{q=1}^{p_k} \xi_{k,i_k+q}.$$

It is easy to see that $\bar{\xi}_{k,1} = 0$ and $\bar{\xi}_{k,n_k} = 1$ for any k, whereas all the remaining points are in $(0,1)$. Finally, the Greville collocation points are defined by the tensor product structure, that is,

$$\mathbf{x}_i = \mathbf{x}(\bar{\boldsymbol{\xi}}_i),\ \bar{\boldsymbol{\xi}}_i = \left(\bar{\xi}_{1,i_1},\ \dots,\bar{\xi}_{d,i_d}\right) \in \hat{\mathcal{D}} = [0,1]^d,\ \forall\, i_k = 1,\ \dots,n_k,\ \prod_{k=1}^{d} n_k = n_c.$$

Known for a generally stable interpolation property, the Greville abscissae have been a default choice for many years. However, there are cases where its stability degrades, especially when very high-order interpolations combined with nonuniform grids are employed.[31]

4.4.2 | Demko abscissae

Another prominent choice for the collocation points, proved to be stable for any mesh and degree, is the Demko abscissae.[28] Along the kth coordinate direction of the parametric domain $\hat{\mathcal{D}} = [0,1]^d$, $k = 1,\ \dots,d$, the Demko abscissae $\tilde{\xi}_{k,i_k}$, $i_k = 1,\ \dots,n_k$, are n_k points defined as the maxima and minima of the Chebyshev splines, that is, the values of Chebyshev splines for which the extrema take on ± 1. Unlike the Greville abscissae, there is no explicit formula for the Demko abscissae. However, the Demko abscissae can be easily determined by an iterative algorithm, for instance, the `chbpnt` program of MATLAB[32] and others.[27] Akin to the Greville abscissae, the Demko abscissae lie inside $[0,1]$.

Finally, the Demko collocation points are defined by the tensor product structure, that is,

$$\mathbf{x}_i = \mathbf{x}(\tilde{\boldsymbol{\xi}}_i),\ \tilde{\boldsymbol{\xi}}_i = (\tilde{\xi}_{1,i_1},\ \ldots,\tilde{\xi}_{d,i_d}) \in \hat{\mathcal{D}} = [0,1]^d,\ \forall\ i_k = 1,\ \ldots,n_k,\ \prod_{k=1}^{d} n_k = n_c.$$

Other approaches for selecting collocation points stem from the Gauss quadrature points, which are the zeros of the Gauss-Legendre polynomials. See the book by Atkinson[17] for further details. The collocation points defined by knot locations have also been used by Wang.[33] For the latter case, additional points have to be defined, so that the number of collocation points matches the number of basis functions.

It is important to clarify that the numbers of collocation points and basis functions in this work are the same. They are both equal to the number of control points n_c of IGA. However, this is not absolutely necessary. For instance, the number of collocation points can be larger than the number of basis functions. This becomes relevant when a relatively few basis functions of IGA are sufficient for a good approximation of an eigenfunction $\phi(\mathbf{x})$, whereas a large number of collocation points are required to minimize globally the residual r_h in a least-squares sense or by other means. The result is a finite-dimensional quadratic eigenvalue problem comprising rectangular system matrices.[34] The quadratic eigenvalue problem can be transformed into a linear eigenvalue problem, often referred to as linearization, using an augmented system of eigensolutions. This paper does not deal with quadratic eigenvalue problems.

4.5 | Eigenvalue solvers

The matrix eigenvalue problem from the collocation method, associated with the subspace $\mathcal{S}_h \in C(\mathcal{D})$, comprises nonsymmetric systems matrices **A** and **B**. In addition, there are multiple ways to select the collocation points. Therefore, it is difficult to know if the resultant eigensolutions are real-valued for any $h > 0$. This is one disadvantage of the collocation method. In contrast, the symmetry and positive-(semi)definiteness of the system matrices from the Galerkin method ensure real-valued eigensolutions for any $h > 0$. Nonetheless, the collocation solutions for a sequence of the subspaces $\{\mathcal{S}_h\}_{h>0}$ is convergent to the correct limit, that is, the eigensolutions $\lambda_h \to \lambda$ and $\phi_h(\mathbf{x}) \to \phi(\mathbf{x})$ as $h \to 0$, where λ and $\phi(\mathbf{x})$ are the real-valued eigenvalue and eigenfunction of (7). Therefore, if the number of control points n_c is sufficiently large, the eigensolutions of (17) are expected to be real-valued. This issue will be revisited when presenting numerical examples.

Depending on the structure and properties of the system matrices **A** and **B**, a number of iterative numerical methods can be used to solve the matrix eigenvalue problem. For symmetric matrices, Krylov subspace methods[35,36] are common choices, although they can be expensive when the systems matrices are dense. Because of this reason, a few researchers[37,38] have employed the hierarchical matrix technique,[39] which, for sufficiently smooth covariance functions, has been claimed to produce computationally efficient eigensolutions. Other efficient methods comprise a Fast Fourier technique,[40] but it works only for homogeneous random fields and rectangular domains. For nonsymmetric matrices, which is the case in this study, two commonly used methods or algorithms are the QZ algorithm[41] and Lanczos method,[41-43] both of which are available as in-built functions in MATLAB (Version 2016a).[32] The QZ algorithm is frequently used with a balancing option to lower the condition number of the system matrices. On the other hand, the Lanczos method, when employed in conjunction with some preconditioning, may lead to optimal or near-optimal convergence. Clearly, the size of system matrices contributes to the cost of the eigenvalue calculations. However, so does the property of the covariance function. If the system matrices tend to be dense or full, the iterative solvers, meant for sparse systems, become less effective. Therefore, more efficient solvers, such as the fast multipole method,[44] have been employed by some researchers.[23]

5 | NUMERICAL EXAMPLES

Three example problems in one, two, and three dimensions are presented to illustrate the isogeometric collocation method for solving the integral eigenvalue problem associated with the KL expansion. The random fields have *zero* means and are homogeneous and isotropic in all examples. The polynomial order p of B-splines, leading to NURBS, and the order of Gauss-Legendre quadrature to estimate the system matrices vary from example to example. The collocation points are defined using Greville and/or Demko abscissae. The isogeometric collocation analysis and subsequent matrix eigenvalue calculations were performed using MATLAB (Version 2016a). The eigenvalue calculations were checked using both the QZ algorithm and the Lanczos method. All eigenvalue calculations reported here are from the QZ algorithm, performed in tandem with standard balancing option. They were verified using the Lanczos method with preconditioning and several starting vectors.

5.1 | One-dimensional problem

Consider a one-dimensional random field $\alpha(x,\cdot)$ with the covariance function $\Gamma(x,x') = \mathbb{E}[\alpha(x,\cdot)\alpha(x',\cdot)]$ defined on $\mathcal{D} = [0,1]$. Three types of covariance functions, described by sinusoidal, Gaussian, and exponential functions, were selected. Mathematically,

$$\Gamma(x,x') = \begin{cases} \sigma^2 \frac{bL}{|x-x'|} \sin\left(\frac{|x-x'|}{bL}\right), & \text{(sinusoidal)}, \\ \sigma^2 \exp\left(-\frac{|x-x'|^2}{(bL)^2}\right), & \text{(Gaussian)}, \\ \sigma^2 \exp\left(-\frac{|x-x'|}{bL}\right), & \text{(exponential)}, \end{cases}$$

where $\sigma^2 = 1, L = 1$. For the correlation length parameter, $b = 0.1$ for the sinusoidal covariance function and $b = 1$ for the Gaussian and exponential covariance functions. These covariance functions are commonly used in engineering applications. The objective is to study numerical convergence of the isogeometric collocation method.

Five polynomial orders of p ranging from one through five, representing linear, quadratic, cubic, quartic, and quintic elements, respectively, were employed. The knot vectors for the coarsest one-element IGA meshes are as follows: (i) $\Xi = (0,0,1,1)$ for linear elements; (ii) $\Xi = (0,0,0,1,1,1)$ for quadratic elements; (iii) $\Xi = (0,0,0,0,1,1,1,1)$ for cubic elements; (iv) $\Xi = (0,0,0,0,0,1,1,1,1,1)$ for quartic elements; and (v) $\Xi = (0,0,0,0,0,0,1,1,1,1,1,1)$ for quintic elements. The corresponding control points and weights for the coarsest one-element IGA meshes are listed in Table A1. As the weights are all equal to one, the NURBS functions are the same as B-splines. By adding new knots and control points, the one-element mesh for each polynomial order was h-refined uniformly, resulting in a series of progressively finer meshes with the number of elements increasing in size. The element size h is reciprocal to the number of elements and hence constant for a fixed number of elements. Both Greville and Demko abscissae were chosen for the collocation points. Figure 2 depicts for an isogeometric mesh comprising eight elements ($h = 0.125$) and $p = 1,2,3,4,5$ the distributions of respective collocation points and knots from both choices.

The relative errors in calculating eigenvalues and eigenfunctions are defined as

$$e_{h,\lambda} := \frac{|\lambda - \lambda_h|}{\lambda} \text{ and } e_{h,\phi} := \frac{||\phi(x) - \phi_h(x)||_\infty}{||\phi(x)||_\infty},$$

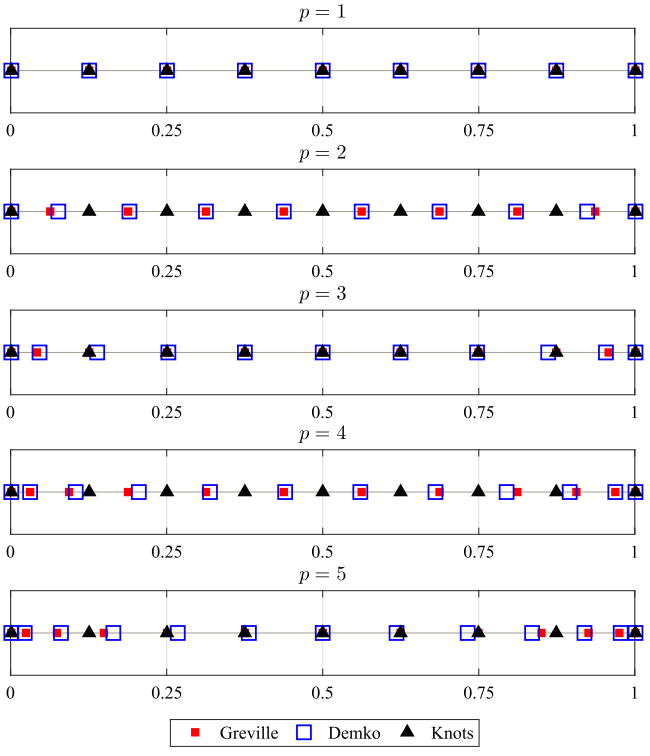

FIGURE 2 Distributions of collocation points from the Greville and Demko abscissae and knots for the eight-element discretization in Example 1 [Colour figure can be viewed at wileyonlinelibrary.com]

respectively. Here, the eigenpair $\{\lambda, \phi(x)\}$ was obtained exactly for the exponential covariance function.[7] Since no exact solutions exist for the sinusoidal and Gaussian covariance functions, respective IGA solutions, obtained from a high-order, highly refined mesh comprising 2^{12} 16th-order elements and a 48-point Gauss-Legendre quadrature rule, were used as reference solutions. However, when $p = 1 - 5$, 10-, 30-, and 800-point Gauss-Legendre quadrature rules were employed in obtaining all approximate IGA solutions for the sinusoidal, Gaussian, and exponential covariance functions, respectively. The orders of numerical integration were chosen adaptively and according to the regularity of the covariance functions. They are large enough to minimize possible errors due to numerical integration.

Obtained for each polynomial order p, Figure 3A and 3B display the relative error e_{h,λ_5} in the fifth eigenvalue for the sinusoidal (top), Gaussian (middle), and exponential (bottom) covariance functions when the collocations points are obtained from the Greville and Demko abscissae, respectively. The exact or reference solutions are $1.759789850392 \times 10^{-2}$, $1.173953119186 \times 10^{-5}$, and $1.227891385452 \times 10^{-2}$, respectively. There is nothing special about the fifth mode, which was selected arbitrarily. For all three covariance functions, the error in the eigenvalue decays with the reduction of the

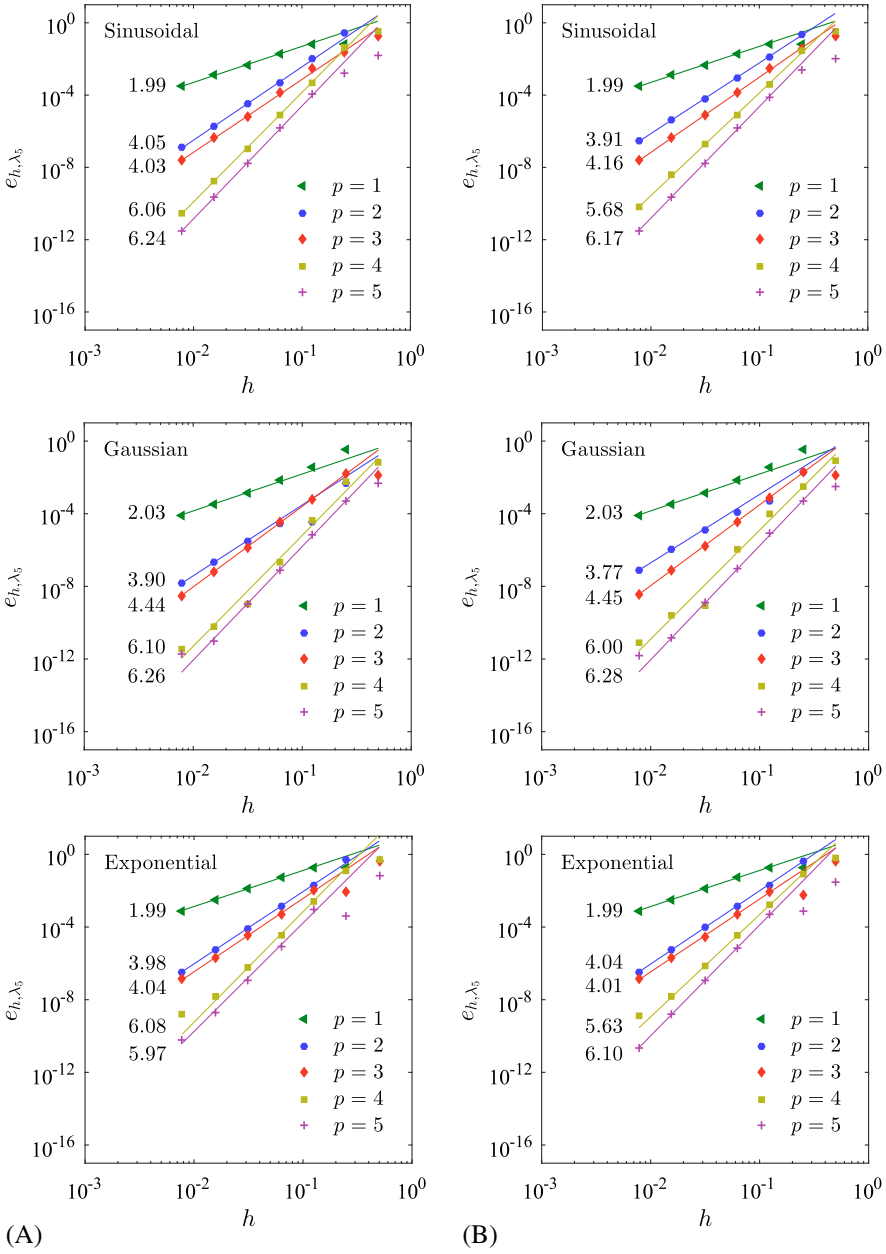

FIGURE 3 Relative error in the fifth eigenvalue e_{h,λ_5} as a function of the element size h for three covariance functions in Example 1. A, Collocation points from Greville abscissae; B, Collocation points from Demko abscissae. Here, the numbers indicate decay rates of error [Colour figure can be viewed at wileyonlinelibrary.com]

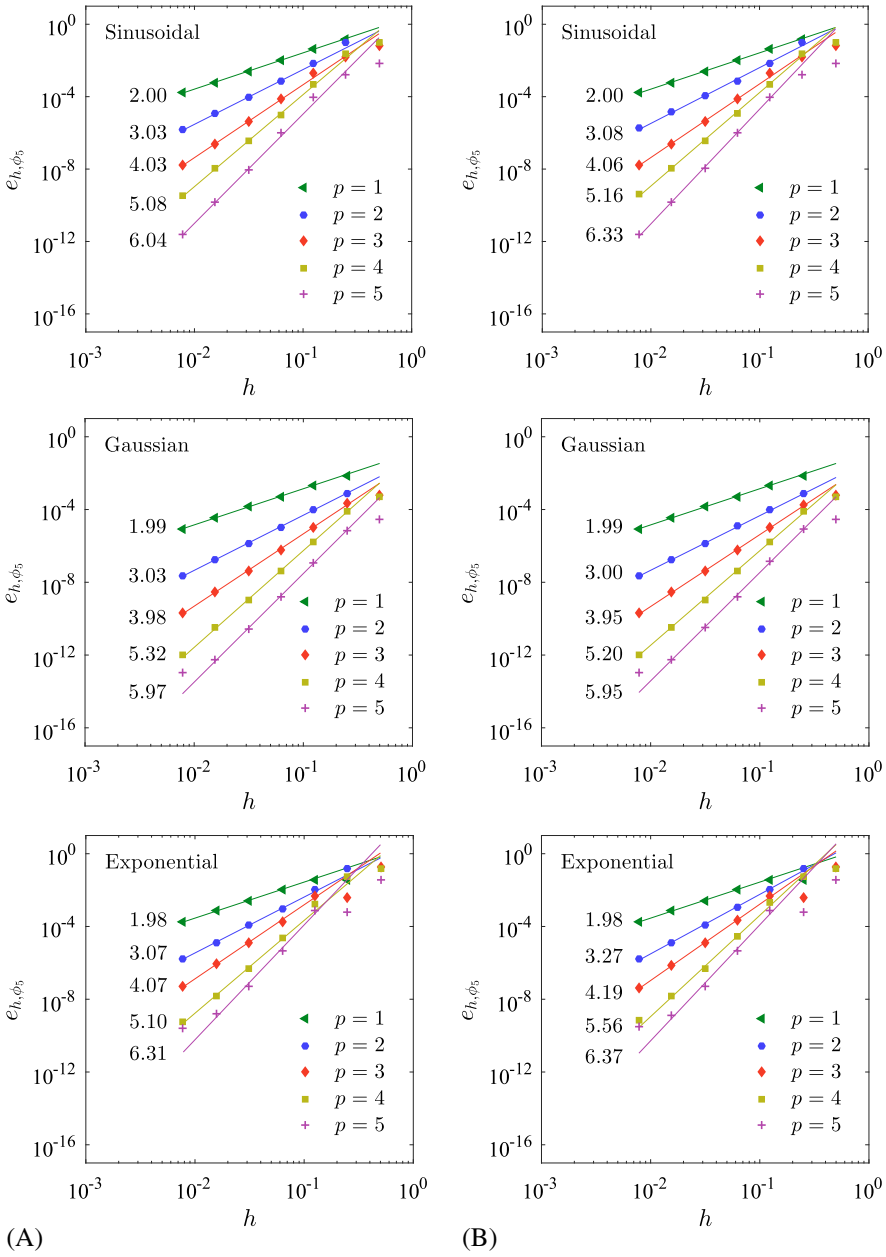

FIGURE 4 Relative error in the fifth eigenfunction e_{h,ϕ_5} as a function of the element size h for three covariance functions in Example 1. A, Collocation points from Greville abscissae; B, Collocation points from Demko abscissae. Here, the numbers indicate decay rates of error [Colour figure can be viewed at wileyonlinelibrary.com]

element size, as expected, regardless of p. However, the rates of decay in all three subplots of Figure 3A and 3B are faster for the higher-order IGA. For instance, the decay rate of order nearly $\mathcal{O}(p+1)$ or $\mathcal{O}(p+2)$ is observed when p is odd or even, respectively, for all three covariance functions. Therefore, the minimum convergence rate is $\mathcal{O}(p+1)$, which is consistent with the theoretical result reported by Wang [33†] No meaningful difference is noted in the convergence rates obtained using the collocation points from the Greville and Demko abscissae. Therefore, existing choices for the collocation points can be used in solving the eigenvalue problem considered in this work.

Finally, Figure 4A and 4B present the relative error e_{h,ϕ_5} in the fifth eigenfunction for the sinusoidal (top), Gaussian (middle), and exponential (bottom) covariance functions, obtained using the Greville and Demko abscissae, respectively. The convergence rates of orders nearly $\mathcal{O}(p+1)$ are found regardless of the parity of p. This is because the exact

†The asymptotic convergence rate is $\mathcal{O}((p+1)/\gamma)$, where γ is the referred to as the index of an eigenvalue. According to Atkinson,[45] $\gamma = 1$ for self-adjoint operators.

eigenfunctions for all three covariance functions are sufficiently smooth continuous functions. Moreover, these numerical results agree with the theoretical error analysis of Wang[33] and Atkinson,[17] who proved the following: the sequence of eigenfunctions obtained from the collocation method entailing p-order B-splines converge to the exact solution at the rate of $\mathcal{O}(p+1)$, which is identical to the rate at which the B-spline interpolations of degree p converge to the C^{p+1}-continuous exact eigenfunction.

5.2 | Two-dimensional problem

For the second example, let $\alpha(\mathbf{x}, \cdot)$ be a two-dimensional random field with the covariance function

$$\Gamma(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{bL}\right), \ \mathbf{x}, \mathbf{x}' \in \mathcal{D} \subset \mathbb{R}^2, \tag{25}$$

defined on a domain constructed by subtracting a quarter of a disk of radius R from a square of size L, as depicted in Figure 5A. The following covariance and geometry parameters were selected: $\sigma^2 = 0.01, L = 20, R = 1$, and $b = 0.5$. The collocation points were generated using the Greville abscissae.

The initial knot vectors and polynomial orders for the coarsest two-element IGA mesh are as follows: (i) $\boldsymbol{\Xi}_1 = (0, 0, 0, 0.5, 1, 1, 1)$, $\boldsymbol{\Xi}_2 = (0, 0, 0, 1, 1, 1)$; and (ii) $p_1 = p_2 = 2$. The initial two-element mesh and corresponding control net are depicted in Figure 5B and 5C, respectively, where the control points are shown in red closed circles. The control points and weights are given in Table A2 in the Appendix. The weights were selected in such a way that the resulting NURBS functions could reproduce exactly a circular arc. Employing a global h-refinement strategy, the knot index space was successively divided, and new control points and weights were added as required. Figure 6 displays six meshes, that is, Meshes 1 through 6, with the corresponding collocation points marked in red closed squares. The number of elements in these meshes varies from two to 2048. The two-element mesh (Mesh 1), obtained using the initial knot index space, control points, and weights in Table A1 in the Appendix, represents already the exact geometry of the physical domain $\mathcal{D}$.

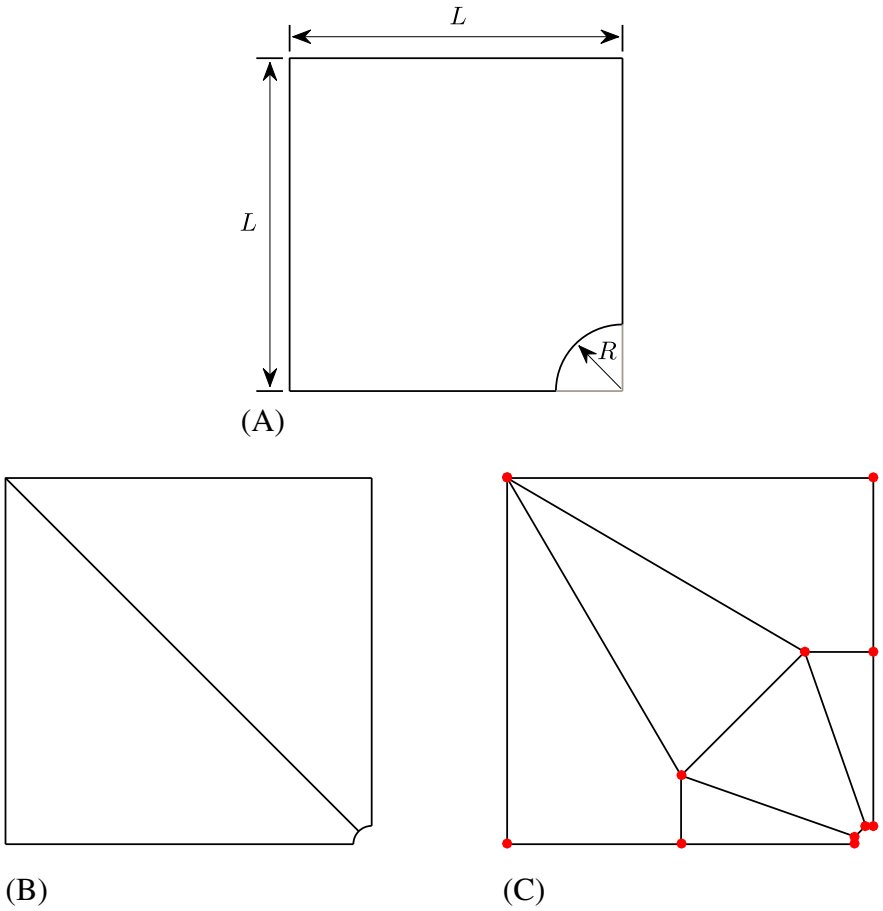

FIGURE 5 A quarter of a plate with a circular hole in Example 2. A, Physical domain; B, Two-element initial mesh; C, Control net. Here, the control points are denoted by red closed circles

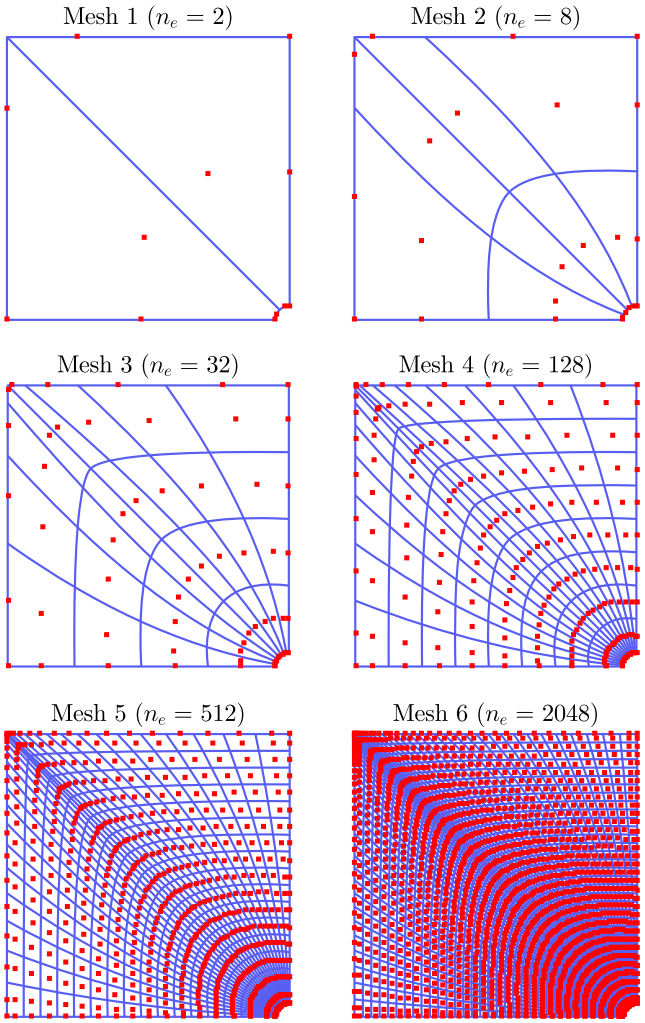

FIGURE 6 Six isogeometric analysis meshes obtained by h-refinement in Example 2. Here, the collocation points are denoted by red closed squares

In fact, all finer meshes, albeit they have more and more elements, represent the exact geometry of D. This is in sharp contrast with FEM, where a mesh, especially when it is coarse, will always induce some geometrical errors. A $3p_k$-point Gauss-Legendre quadrature was employed in each coordinate direction k, where $k = 1, 2$, for constructing the system matrices.

Table 1 presents the first ten largest eigenvalues for the covariance function in (25) calculated by the isogeometric collocation method for all six meshes in Figure 6. For Mesh 1, the fourth and fifth modes are associated with complex-valued eigensolutions, which is possible due to a very coarse discretization entailing only two elements. For finer meshes starting from even Mesh 2, all eigensolutions, at least for the first ten modes shown, are real-valued. Clearly, the eigenvalues converge with respect to the number of elements or mesh refinement, as expected. The sequence of eigenfunctions from various meshes tells the same tale, but detailed results from all these meshes are suppressed for brevity. Instead, only the first four eigenfunctions, obtained using Mesh 6 (2048 elements), are illustrated as contour plots in Figure 7. Again, as the results of collocation method using the Demko abscissae are similar, they are not reported here. Note that, unlike Example 1, Example 2 does not have analytical eigensolutions for the nonrectangular domain and inseparable covariance function examined.

Given that the same problem can be solved by the Galerkin isogeometric method,[10] a comparison between the existing Galerkin and proposed collocation methods should be intriguing. For this reason, Table 2 lists the eigenvalues obtained from the Galerkin method using the same six meshes. Indeed, all ten eigenvalues calculated by the Galerkin and collocation methods agree immensely well for finer meshes, corroborating the accuracy of the latter. The same can be said when contrasting the eigenfunctions in Figure 7 with those obtained by the Galerkin method using Mesh 6 in Figure 8. Any difference between two respective eigenfunctions from Figures 7 and 8 is indiscernible to the naked eye.

TABLE 1 Ten largest eigenvalues estimated by the collocation method using six meshes in Example 2

Mode	Eigenvalue					
	Mesh 1 (2 elems.)	Mesh 2 (8 elems.)	Mesh 3 (32 elems.)	Mesh 4 (128 elems.)	Mesh 5 (512 elems.)	Mesh 6 (2048 elems.)
1	1.594082651	1.610406781	1.613611305	1.614414152	1.614492766	1.614499714
2	0.365228404	0.438184379	0.439032509	0.439423192	0.439506972	0.439516320
3	0.331137260	0.414098919	0.435417100	0.437756818	0.437964684	0.437980201
4	–[a]	0.160729652	0.178760730	0.180254726	0.180490077	0.180510437
5	–[a]	0.098868655	0.133787546	0.136366707	0.136618095	0.136639055
6	0.073231756	0.084004232	0.123331055	0.126075032	0.126281733	0.126296757
7	0.051158166	0.054148662	0.074034584	0.075150660	0.075282606	0.075299154
8	0.030657379	0.041896851	0.066839955	0.074016392	0.074559674	0.074598985
9	0.022376609	0.034685509	0.046795851	0.050093786	0.050404752	0.050426773
10	0.002750985	0.033945558	0.046116508	0.049961984	0.050300654	0.050325517

[a] Complex-valued eigensolutions.

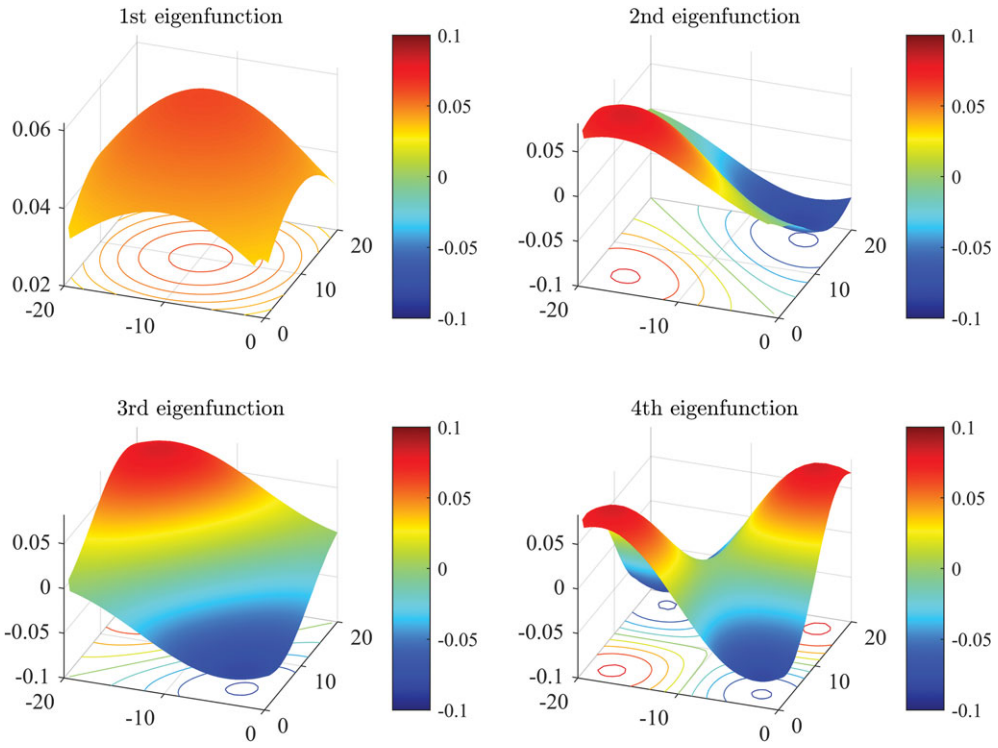

FIGURE 7 Contour plots of the first four eigenfunctions obtained using the isogeometric collocation method and Mesh 6 in Example 2

To judge the respective computational efforts in obtaining eigensolutions, Table 3 lists for all six meshes the ratios of the central-processing-unit (CPU) times by the Galerkin and collocation methods, calculated using an Intel Core i7-7700k, 4.20 GHz, 64 GB RAM personal computer. Two CPU ratios were defined for both methods: (i) the first CPU ratio involves times required to construct the system matrices only; and (ii) the second CPU ratio entails total times needed to obtain the eigensolutions. With the exception of the coarsest two-element mesh (Mesh 1), both CPU ratios are greater than *one*, meaning that the collocation method, in general, is significantly more economical than the Galerkin method as the mesh size increases. The ratios escalate for finer meshes; for instance, when using Meshes 4 through 6, the CPU ratios are nearly ten. For Mesh 6, there is a slight downtick in the CPU ratios. This is due to the aspect of local support, which becomes relevant for a large mesh size, as discussed in Section 4.3. Nevertheless, the collocation method offers a huge computational benefit over the Galerkin method, while retaining an accuracy very close to that of the latter. The primary reason for this computational efficiency stems from the requisite effort in constructing the systems matrices:

TABLE 2 Ten largest eigenvalues estimated by the Galerkin method using six meshes in Example 2

Mode	Eigenvalue					
	Mesh 1 (2 elems.)	Mesh 2 (8 elems.)	Mesh 3 (32 elems.)	Mesh 4 (128 elems.)	Mesh 5 (512 elems.)	Mesh 6 (2048 elems.)
1	1.621717640	1.615855701	1.614674440	1.614522533	1.614503164	1.614500736
2	0.443735175	0.440581205	0.439757808	0.439549093	0.439521337	0.439517825
3	0.426765339	0.436580845	0.438054866	0.437998789	0.437983707	0.437981733
4	0.174584715	0.175944397	0.180598416	0.180538574	0.180515877	0.180512700
5	0.129889894	0.129774563	0.136577350	0.136661372	0.136643947	0.136641269
6	0.120246606	0.123604098	0.126242791	0.126318766	0.126300806	0.126298313
7	0.076218052	0.073564511	0.075249814	0.075332317	0.075305254	0.075301444
8	0.054535018	0.063849764	0.072526706	0.074600945	0.074604171	0.074602205
9	0.038081769	0.046677161	0.050014641	0.050439386	0.050431033	0.050428828
10	0.027258973	0.045695342	0.049457664	0.050340735	0.050330457	0.050327867

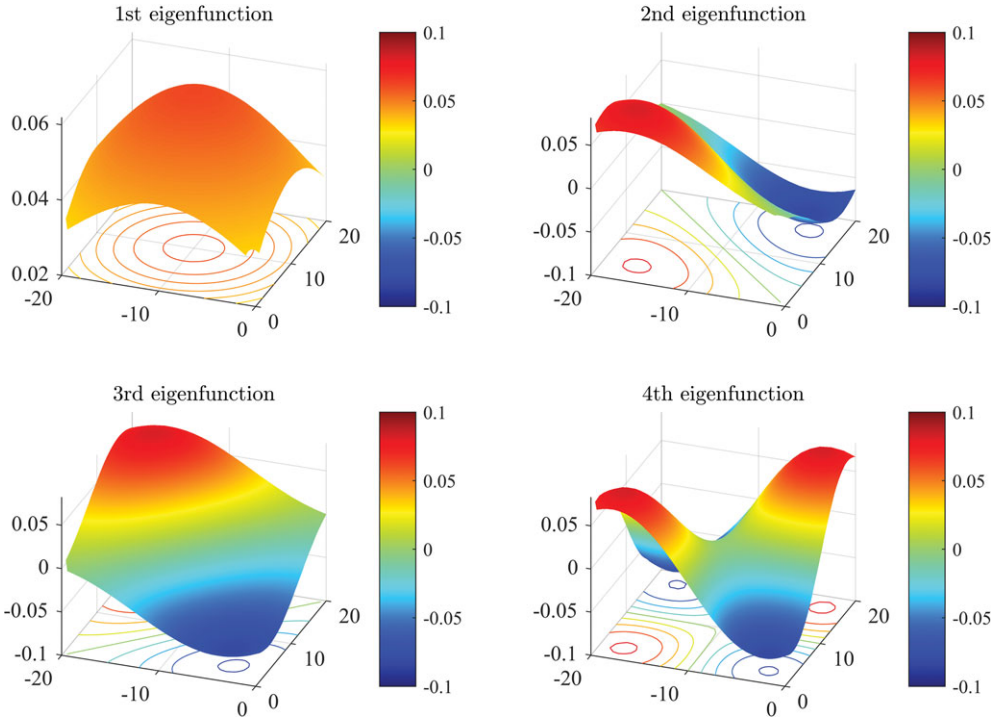

FIGURE 8 Contour plots of the first four eigenfunctions obtained using the Galerkin isogeometric method and Mesh 6 in Example 2

TABLE 3 Ratio of CPU times by the Galerkin and collocation methods in Example 2

Mesh	No. of Elems. (n_e)	No. of Control Pts. (n_c)	CPU Ratio-1[a]	CPU Ratio-2[b]
1	2	12	0.81	1.25
2	8	24	2.79	2.37
3	32	60	7.04	6.57
4	128	180	9.63	9.55
5	512	612	10.17	10.15
6	2048	2244	8.63	8.61

[a] CPU ratio-1 = Galerkin CPU time/Collocation CPU time for constructing system matrices.
[b] CPU ratio-2 = Galerkin total CPU time/Collocation total CPU time for obtaining eigensolutions.

only a two-dimensional numerical integration is required in the collocation method, whereas a four-dimensional numerical integration is mandated in the Galerkin method.

5.3 | Three-dimensional problem

The final example entails a three-dimensional random field $\alpha(\mathbf{x}, \cdot)$ with the covariance function

$$\Gamma(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{bL}\right), \ \mathbf{x}, \mathbf{x}' \in \mathcal{D} \subset \mathbb{R}^3, \tag{26}$$

defined on a pipe with 90-degree elbow bend of inner radius R_i, outer radius R_o, and mean radius R of the circular bend along the center-line of the pipe, as displayed in Figure 9A. The covariance and geometry parameters were chosen as follows: $\sigma^2 = 0.01, R_i = 1, R_o = 2, R = 3, L = 2$ and $b = 0.5$. The collocation points were selected from the Greville abscissae.

Four IGA meshes, the first of which being an 16-quadratic-element mesh, followed by three progressively finer meshes with 64, 512, and 4096 quadratic elements, were analyzed. They were all obtained by h-refinement of a four-element

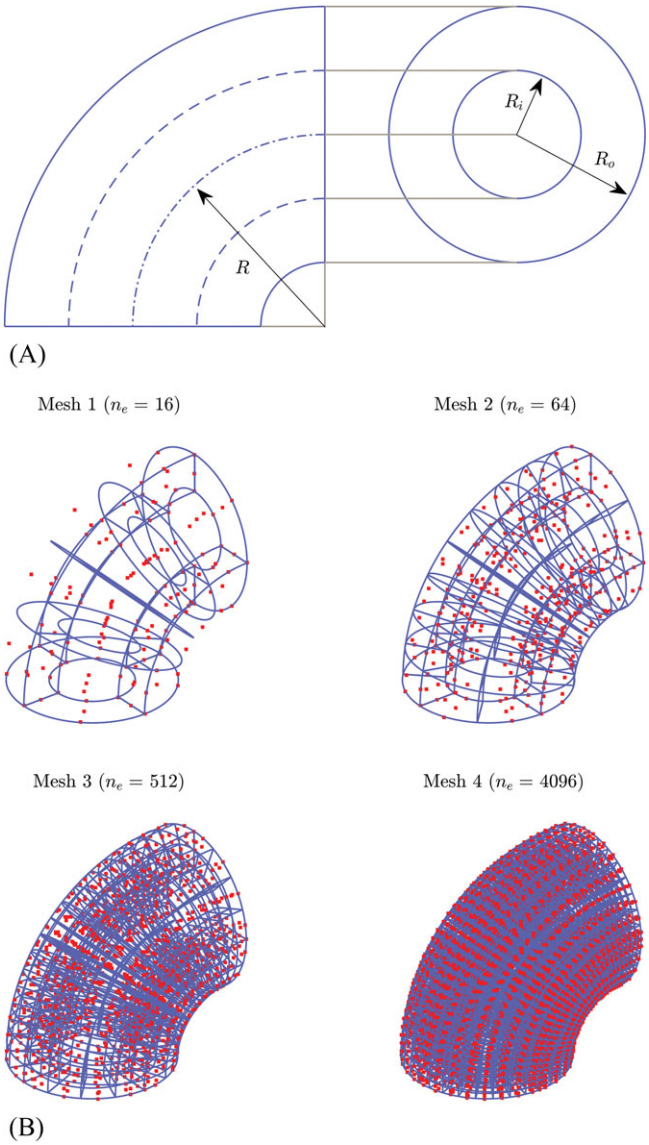

FIGURE 9 A pipe with a 90-degree elbow bend in Example 3. A, Physical domain's sketch; B, Four isogeometric analysis meshes obtained by h-refinement. Here, the collocation points are denoted by red closed squares

TABLE 4 Twenty largest eigenvalues estimated by the isogeometric collocation method using four meshes in Example 3

	Eigenvalue			
Mode	Mesh 1 (16 Elements)	Mesh 1 (64 Elements)	Mesh 3 (512 Elements)	Mesh 4 (4096 Elements)
1	0.048530643	0.048668751	0.048641336	0.048648279
2	0.026104574	0.026378282	0.026362172	0.026369282
3	0.023898134	0.024026313	0.023990458	0.023998581
4	0.021851144	0.021984555	0.021956708	0.021964713
5	0.013781928	0.014042400	0.014057488	0.014065980
6	0.013525158	0.014007580	0.014003940	0.014012649
7	0.009805095	0.010282621	0.010278206	0.010287650
8	0.009765434	0.009628604	0.009769190	0.009788065
9	0.008237604	0.009576794	0.009626953	0.009643886
10	0.007489059	0.007732165	0.007789040	0.007799208
11	0.007397192	0.007603880	0.007655505	0.007665671
12	0.005617336	0.005779476	0.005902062	0.005918628
13	0.004649038	0.005283183	0.005349418	0.005364384
14	0.004514105	0.005250339	0.005266784	0.005277672
15	0.004355879	0.004375997	0.004466269	0.004477329
16	0.003209290	0.004271584	0.004366757	0.004381390
17	0.003165791	0.004073149	0.004313248	0.004345170
18	0.002852608	0.004051383	0.004282523	0.004312001
19	0.002522366	0.003320937	0.003407271	0.003422594
20	0.002437858	0.002949144	0.003016793	0.003029771

starter mesh with the knot vectors defined in Table A3 in the Appendix. The numbers of control points and weights, 162 for Mesh 1, 390 for Mesh 2, 1512 for Mesh 3, and 7548 for Mesh 4, are too numerous to be listed. All four meshes are displayed in Figure 9B along with the collocation points marked in red closed squares. Again, they represent the exact geometry of the physical domain regardless of the mesh. A $2p_k$-point Gauss-Legendre quadrature was employed in each coordinate direction k, where $p_1 = p_2 = p_3 = 2$.

Table 4 lists the first twenty largest eigenvalues obtained by the isogeometric collocation method using the aforementioned four meshes for the covariance function defined in (26). The respective eigenvalues from all four meshes, especially from Meshes 3 and 4, are very close to each other. The same observation holds for the eigenfunctions. To be concise, however, only the first six eigenfunctions obtained for Mesh 4 are displayed in Figure 10. Again, no analytical solutions exist for this problem, but the relative stability of eigensolutions from the four meshes lends credence to the collocation method.

Regarding the computational effort, the same two CPU ratios defined in Example 2 were also employed in this example but only for Meshes 1-3. Mesh 4, too refined, was deemed impractical for obtaining the associated Galerkin solution. Table 5 presents for the first three meshes both CPU ratios by the Galerkin and collocation methods. The absolute CPU times needed to construct the system matrices and to obtain the eigensolutions for Meshes 1-3 by both methods are also tabulated except for Mesh 4. Again, the CPU ratios sharply rise with the mesh size as expected. For Mesh 3 (512 elements), the ratios have climbed to nearly ten, demonstrating high computational cost of the Galerkin method. No depreciation in the computational efficiency of the collocation method was observed, at least, for the first three meshes. For Mesh 4, the collocation method required nearly 15 hours to obtain the eigensolutions, where the construction of system matrices consumed about 99.5 percent of the total CPU time. If the CPU ratios of ten prevail for Mesh 4, then the Galerkin method would take more than six days to obtain practically the same results of the collocation method. Therefore, when solving integral eigenvalue problems similar to the one in Example 3 and even more realistic three-dimensional problems not studied here, the use of the Galerkin method may become impractical if not prohibitive. In contrast, the collocation method proposed provides an accurate and computationally efficient way of generating eigensolutions at a reasonable cost.

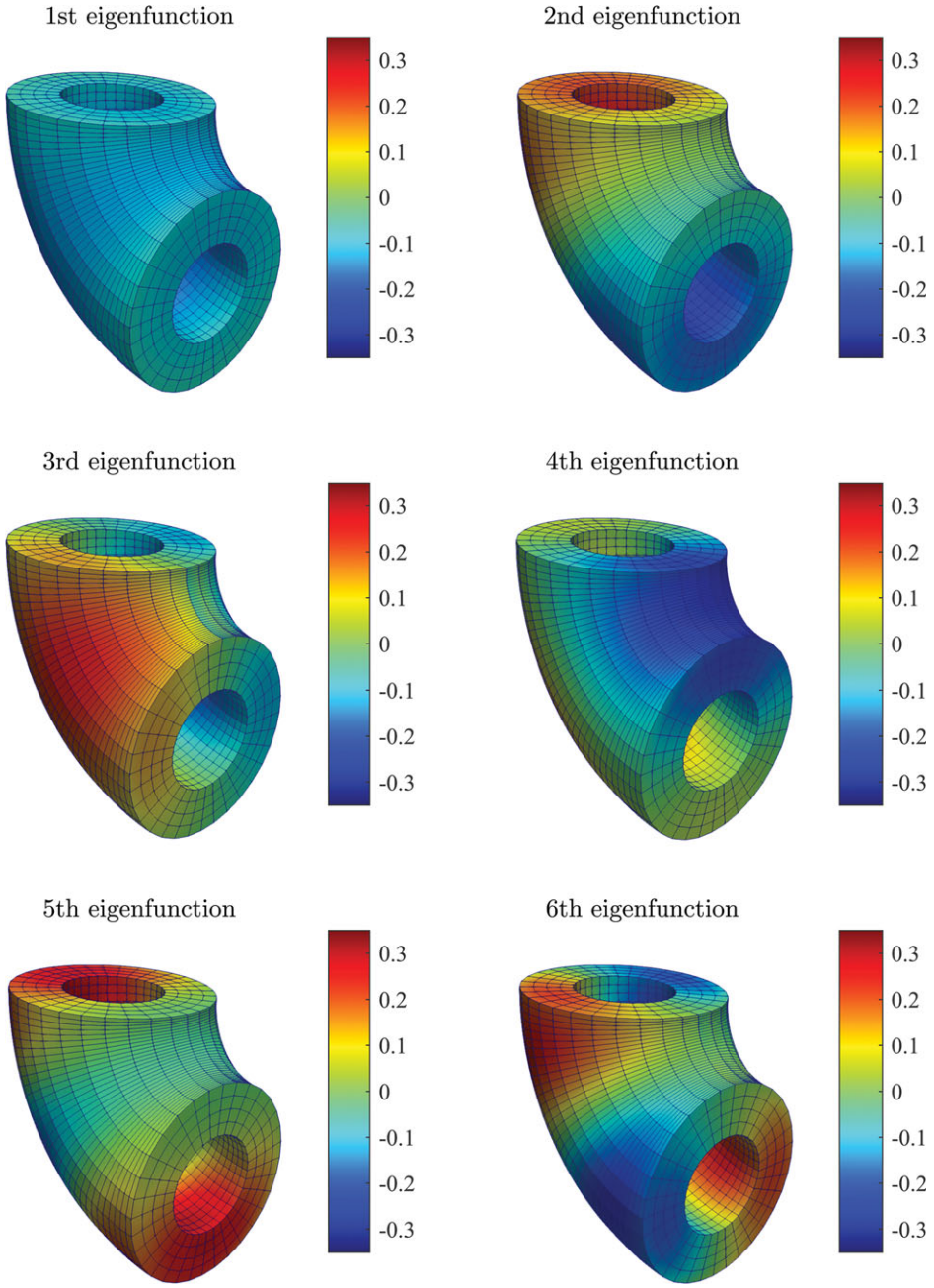

FIGURE 10 Contour plots of the first six eigenfunctions obtained using the isogeometric collocation method and the finest mesh in Example 3

TABLE 5 CPU times by the Galerkin and collocation methods in Example 3

Mesh	CPU time by Galerkin, s		CPU time by collocation, s		CPU ratio-1[a]	CPU ratio-2[b]
	syst. matrices	total	syst. matrices	total		
1	9.13	9.24	2.67	2.78	3.42	3.32
2	144.28	144.45	24.87	25.04	5.80	5.77
3	9124.11	9126.36	897.91	900.32	10.16	10.14
4	–[c]	–[c]	53,215.06	53,462.55	–[d]	–[d]

[a]CPU ratio-1 = Galerkin CPU time/Collocation CPU time for constructing system matrices.
[b]CPU ratio-2 = Galerkin total CPU time/Collocation total CPU time for obtaining eigensolutions.
[c]Not calculated.
[d]Not applicable.

6 | DISCUSSION

The Galerkin method incorporating finite-element subspaces reigns supreme when solving boundary-value problems in solid mechanics. Nowadays, the Galerkin method is practically synonymous with FEM. On the other hand, the collocation method in the context of finite-element analysis has little more to offer, rendering it inessential and rarely used in practice. The root cause for the limited application of the collocation method lies in the finite-element basis functions that are usually no more than C^0-continuous. Indeed, the lack of smoothness erases the viability of developing finite-dimensional approximations using the strong form of governing equations. In contrast, an IGA presents an opportunity to develop such approximations because it is possible to generate smoother basis functions for geometrically complex domains. Therefore, the collocation method deserves a second look where one can capitalize on the geometric flexibility of IGA, including inheriting a direct link to CAD.

Since the KL expansion entails solving an integral equation in lieu of a differential equation, the smoothness of the basis functions, essential for the collocation method, may appear to be less relevant. However, as briefly stated in Section 4 and explained more so by Atkinson,[17] the accuracy and convergence properties of an approximate solution, be it generated by Galerkin, collocation, or any other projection method, strongly depend on the smoothness properties of the unknown solution. The preceding statement is valid whether the governing equation is an integral equation or a differential equation. Hence, the collocation method is very much germane to this study involving the KL expansion.

The principal advantage of the collocation method is computational efficiency, where the cost of numerical integration in the Galerkin method is markedly reduced. This subject becomes particularly relevant and perhaps necessary when solving the integral eigenvalue problem stemming from the Karhunen-Loève expansion of random field. By sidestepping the need for a weak form of the governing integral equations, the collocation method eliminates one dimension-order of domain integration in forming the finite-dimensional matrix equations. As a result, for a d-dimensional integral eigenvalue problem, the collocation method requires only a d-dimensional domain integration, whereas a $2d$-dimensional domain integration is involved in the Galerkin method. The potential benefits are substantial.

While the paper introduces isogeometric collocation for random field discretization, there is much additional research needed. Future works on the collocation method should include theoretical error estimates, stability analysis of finite-dimensional matrix equations, and the ability to analyze multipatch geometry of complex domains. More importantly, the design of complex structures or systems requires estimation of the probabilistic characteristics of an output response variable of interest and the underlying risk of failure. The IGA analysis presented here provides only a means to parameterize a random field by a finite number of input random variables. Therefore, future endeavors on uncertainty quantification and reliability analysis are in order.

7 | CONCLUSION

An isogeometric collocation method was developed for solving an integral eigenvalue problem, resulting in a computationally efficient discretization of random fields by means of the well-known Karhunen-Loève expansion. The method employs a collocation projection onto a finite-dimensional subspace of continuous functions over a bounded domain. Using B-splines and NURBS functions as the basis of the subspace, a finite-dimensional matrix eigenvalue problem is formulated, where the system matrices are constructed by basis functions and domain integration. Finally, the eigensolutions are obtained using standard methods of linear algebra. Similar to the existing Galerkin isogeometric method, the isogeometric collocation method preserves an exact geometrical representation of many commonly used physical or computational domains and exploits the regularity of isogeometric basis functions delivering globally smooth eigensolutions. However, the chief advantage of the collocation method is computational efficiency in constructing the system matrices and the resulting matrix equations, where the cost of numerical integration in the Galerkin method is greatly diminished. By avoiding the need for a weak form of the governing integral equations, the collocation method shaves off one dimension-order of domain integration in establishing the finite-dimensional matrix equations. Consequently, given a d-dimensional integral eigenvalue problem, the collocation method needs only a d-dimensional domain integration, whereas a $2d$-dimensional domain integration is required in the Galerkin method. Therefore, the introduction of the collocation method for random field discretization offers a hefty computational benefit over the existing Galerkin method. The results from three numerical problems in all three dimensions indicate that the isogeometric collocation method developed provides accurate, convergent, and computationally efficient eigensolutions.

ACKNOWLEDGEMENT

This work was supported by the U.S. National Science Foundation under grant CMMI-1607398.

ORCID

Sharif Rahman http://orcid.org/0000-0003-0837-9871

REFERENCES

1. Karhunen K. Über lineare methoden in der wahrscheinlichkeitsrechnung. *Ann Acad Sci Fenn Ser A I Math*. 1947;37:3-79.
2. Loève M. Fonctions aléatoires de second ordre. In: *Processus Stochastiques et Mouvement Brownien*. Paris, France: Gauthier Villars; 1948:299-352.
3. Loève M. *Probability Theory II*. New York, NY: Springer-Verlag New York Inc; 1977.
4. Adler RJ. *The Geometry of Random Fields*. New York, NY: Wiley; 1981.
5. Davenport WB, Root WL. *An Introduction to the Theory of Random Signals and Noise*. New York, NY: McGraw-Hill; 1958.
6. Van Trees HL. *Detection, Estimation, and Modulation Theory*. Vol. 1. New York, NY: Wiley; 1968.
7. Ghanem RG, Spanos PD. *Stochastic Finite Elements: A Spectral Approach*. New York, NY: Dover Publications Inc; 1991.
8. Rahman S, Xu H. A meshless method for computational stochastic mechanics. *Int J Comput Methods Eng Sci Mech*. 2005;6(1):41-58.
9. Rahman S. Meshfree methods in computational stochastic mechanics. In: Haldar A, ed. *Recent Developments in Reliability-Based Civil Engineering*. Singapore, Singapore: World Scientific Publishing Company; 2005.
10. Rahman S. A Galerkin isogeometric method for Karhunen-Loève approximation of random fields. *Comput Methods Appl Mech Eng*. 2018;338:533-561.
11. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Methods Appl Mech Eng*. 2005;194(39-41):4135-4195.
12. Piegl LA, Tiller W. *The NURBS Book*. 2nd ed. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 1997.
13. Cox MG. *The Numerical Evaluation of B-splines*. Teddington, UK: National Physical Laboratory; 1971.
14. De Boor C. On calculation with B-splines. *J Approx Theory*. 1972;6:50-62.
15. Farin G. *NURBS: From Projective Geometry to Practical Use*. 2nd ed. Natick, MA: A K Peters, Ltd; 1999.
16. Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Chichester, UK: John Wiley & Sons; 2009.
17. Atkinson KE. *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge, UK: Cambridge University Press; 1997.
18. Bazilevs Y, Beirão de Veiga L, Cottrell JA, Hughes TJR, Sangalli G. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Math Models Methods Appl Sci*. 2006;16(7):1031-1090.
19. Bramble JH, Hilbert SR. Estimation of linear functionals on Sobolev spaces with application to Fourier transforms and spline interpolation. *SIAM J Numer Anal*. 1970;7(1):112-124.
20. Dunford N, Schwartz JT. *Linear Operators Part II: Spectral Theory, Self Adjoint Operators in Hilbert Space*. New York, NY: Wiley-Interscience; 1988.
21. Le Maître OP, Knio OM. *Spectral Methods for Uncertainty Quantification*. New York, NY: Springer Science + Business Media; 2010.
22. Smith RC. *Uncertainty Quantification: Theory, Implementation, and Applications*. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2013.
23. Todor RA, Schwab C. Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients. *IMA J Numer Anal*. 2007;27(2):232-261.
24. Matthies HG, Keese A. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput Methods Appl Mech Eng*. 2005;194(12-16):1295-1331.
25. Grigoriu M. *Applied Non-Gaussian Processes: Examples, Theory, Simulation, Linear Random Vibration, and MATLAB Solutions*. Englewood Cliffs, NJ: Prentice-Hall; 1995.
26. Auricchio F, Calabró F, Hughes TJR, Reali A, Sangalli G. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Comput Methods Appl Mech Eng*. 2012;249-252:15-27.
27. De Boor C. *A Practical Guide to Splines*. New York, NY: Springer; 2001.
28. Demko S. On the existence of interpolation projectors onto spline spaces. *J Approx Theory*. 1985;43:151-156.
29. Auricchio F, Da Veiga L, Hughes TJR, Reali A, Sangalli G. Isogeometric collocation methods. *Math Models Methods Appl Sci*. 2010;20(11):2075-2107.
30. Reali A, Hughes TJR. An introduction to isogeometric collocation methods. In: *Isogeometric Methods for Numerical Simulation*. Vienna, Austria: Springer; 2015:173-204. *CISM International Centre for Mechanical Sciences*; vol. 561.
31. Jia R-Q. Spline interpolation at knot averages. *Constr Approx*. 1988;4(1):1-7.
32. MATLAB version 2016a. Natick, MA: The MathWorks, Inc.; 2016.

33. Wang JY. On the numerical computation of eigenvalues and eigenfunctions of compact integral operators using spline functions. *IMA J Appl Math*. 1976;18(2):177-188.
34. Betz W, Papaioannou I, Straub D. Numerical methods for the discretization of random fields by means of the Karhunen-Loève expansion. *Comput Methods Appl Mech Eng*. 2014;271:109-129.
35. Parlett BN. *The Symmetric Eigenvalue Problem*. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1998.
36. Saad Y. *Numerical Methods for Large Eigenvalue Problems*. New York, NY: Halsted Press; 1992.
37. Eiermann M, Ernst OG, Ullmann E. Computational aspects of the stochastic finite element method. *Comput Vis Sci*. 2007;10(1):3-15.
38. Khoromskij BN, Litvinenko A, Matthies HG. Application of hierarchical matrices for computing the Karhunen-Loève expansion. *Computing*. 2009;84(1-2):49-67.
39. Börm S, Grasedyck L, Hackbusch W. Introduction to hierarchical matrices with applications. *Eng Anal Bound Elem*. 2003;27:405-422.
40. Frigo M, Johnson SG. FFTW: an adaptive software architecture for the FFT. In: Proceedings of the 1998 IEEE International Conference on 1998 Acoustics, Speech and Signal Processing (ICASSP); 1998; Seattle, WA.
41. Golub GH, van Loan CF. *Matrix Computations*. 3rd ed. Baltimore, MD: The John Hopkins University Press; 1996.
42. Lehoucq RB, Sorensen DC, Yang C. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1998.
43. Wu K, Simon H. Thick-restart Lanczos method for large symmetric eigenvalue problems. *SIAM J Matrix Anal Appl*. 2000;22(2):602-616.
44. Greengard L, Rokhlin V. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numer*. 1997;6:229-269.
45. Atkinson K. Convergence rates for approximate eigenvalues of compact integral operators. *SIAM J Numer Anal*. 1975;12(2):213-222.

How to cite this article: Jahanbin R, Rahman S. An isogeometric collocation method for efficient random field discretization. *Int J Numer Methods Eng*. 2019;117:344–369. https://doi.org/10.1002/nme.5959

APPENDIX

IGA DETAILS OF NUMERICAL EXAMPLES

Tables A1 and A2 list the control points and weights for the coarsest mesh in Examples 1 and 2, respectively. Table A3 describes the knot vectors for all four meshes in Example 3.

TABLE A1 Control points and weights for the coarsest one-element isogeometric analysis mesh in Example 1

$p=1$		$p=2$		$p=3$		$p=4$		$p=5$	
$\mathbf{C_i}$	w_i	$\mathbf{C_i}$	w_i	$\mathbf{C_i}$	w_i	$\mathbf{C_i}$	w_i	$\mathbf{C_i}$	w_i
$(0,0)$	1	$(0,0)$	1	$(0,0)$	1	$(0,0)$	1	$(0,0)$	1
$(1,0)$	1	$\left(\frac{1}{2},0\right)$	1	$\left(\frac{1}{3},0\right)$	1	$\left(\frac{1}{4},0\right)$	1	$\left(\frac{1}{5},0\right)$	1
		$(1,0)$	1	$\left(\frac{2}{3},0\right)$	1	$\left(\frac{1}{2},0\right)$	1	$\left(\frac{2}{5},0\right)$	1
				$(1,0)$	1	$\left(\frac{3}{4},0\right)$	1	$\left(\frac{3}{5},0\right)$	1
						$(1,0)$	1	$\left(\frac{4}{5},0\right)$	1
								$(1,0)$	1

TABLE A2 Control points, weights, and collocation points for the coarsest two-element isogeometric analysis mesh in Example 2

Control Point	Weight	Collocation Point
$(-1,0)$	1	$(-1,0)$
$\left(-1,\sqrt{2}-1\right)$	$\frac{1+\sqrt{2}}{2\sqrt{2}}$	$(-0.9298,0.3681)$
$\left(-\sqrt{2}+1,1\right)$	$\frac{1+\sqrt{2}}{2\sqrt{2}}$	$(-0.3681,0.9298)$
$(0,1)$	1	$(0,1)$
$\left(-\frac{21}{2},1\right)$	1	$(-10.5,0)$
$\left(-\frac{21}{2},\frac{15}{4}\right)$	1	$(-10.3184,5.8198)$
$\left(-\frac{15}{4},\frac{21}{2}\right)$	1	$(-5.8198,10.3184)$
$\left(0,\frac{21}{2}\right)$	1	$(0,10.5)$
$(-20,0)$	1	$(-20,0)$
$(-20,20)$	1	$(-20,15)$
$(-20,20)$	1	$(-15,20)$
$(0,20)$	1	$(0,20)$

TABLE A3 Knot vectors for all four isogeometric analysis meshes in Example 3[a]

Mesh 1	Mesh 2
$\Xi_1=\left(0,0,0,\frac{1}{4},\frac{1}{4},\frac{1}{2},\frac{1}{2},\frac{3}{4},\frac{3}{4},1,1,1\right)$	$\Xi_1=\left(0,0,0,\frac{1}{8},\frac{1}{4},\frac{1}{4},\frac{3}{8},\frac{1}{2},\frac{1}{2},\frac{5}{8},\frac{3}{4},\frac{3}{4},\frac{7}{8},1,1,1\right)$
$\Xi_2=(0,0,0,1,1,1)$	$\Xi_2=(0,0,0,1,1,1)$
$\Xi_3=\left(0,0,0,\frac{1}{4},\frac{1}{2},\frac{3}{4},1,1,1,\right)$	$\Xi_3=\left(0,0,0,\frac{1}{8},\frac{1}{4},\frac{3}{8},\frac{1}{2},\frac{5}{8},\frac{3}{4},\frac{7}{8},1,1,1,\right)$
Mesh 3	**Mesh 4**
$\Xi_1=\left(0,0,0,\frac{1}{16},\frac{1}{8},\frac{3}{16},\frac{1}{4},\frac{1}{4},\frac{5}{16},\frac{3}{8},\frac{7}{16},\frac{1}{2},\right.$	$\Xi_1=\left(0,0,0,\frac{1}{32},\frac{1}{16},\frac{3}{32},\frac{1}{8},\frac{5}{32},\frac{3}{16},\frac{7}{32},\frac{1}{4},\frac{1}{4},\frac{9}{32},\frac{5}{16},\right.$
$\left.\frac{1}{2},\frac{9}{16},\frac{5}{8},\frac{11}{16},\frac{3}{4},\frac{3}{4},\frac{13}{16},\frac{7}{8},\frac{15}{16},1,1,1\right)$	$\left.\frac{11}{32},\frac{3}{8},\frac{13}{32},\frac{7}{16},\frac{15}{32},\frac{1}{2},\frac{1}{2},\frac{17}{32},\frac{9}{16},\frac{19}{32},\frac{5}{8},\frac{21}{32},\frac{11}{16},\right.$
	$\left.\frac{23}{32},\frac{3}{4},\frac{3}{4},\frac{25}{32},\frac{13}{16},\frac{27}{32},\frac{7}{8},\frac{29}{32},\frac{15}{16},\frac{31}{32},1,1,1\right)$
$\Xi_2=\left(0,0,0,\frac{1}{2},1,1,1\right)$	$\Xi_2=\left(0,0,0,\frac{1}{4},\frac{1}{2},\frac{3}{4},1,1,1\right)$
$\Xi_3=\left(0,0,0,\frac{1}{16},\frac{1}{8},\frac{3}{16},\frac{1}{4},\frac{5}{16},\frac{3}{8},\frac{7}{16},\frac{1}{2},\right.$	$\Xi_3=\left(0,0,0,\frac{1}{32},\frac{1}{16},\frac{3}{32},\frac{1}{8},\frac{5}{32},\frac{3}{16},\frac{7}{32},\frac{1}{4},\frac{9}{32},\frac{5}{16},\right.$
$\left.\frac{9}{16},\frac{5}{8},\frac{11}{16},\frac{3}{4},\frac{13}{16},\frac{7}{8},\frac{15}{16},1,1,1\right)$	$\left.\frac{11}{32},\frac{3}{8},\frac{13}{32},\frac{7}{16},\frac{15}{32},\frac{1}{2},\frac{17}{32},\frac{9}{16},\frac{19}{32},\frac{5}{8},\frac{21}{32},\frac{11}{16},\right.$
	$\left.\frac{23}{32},\frac{3}{4},\frac{25}{32},\frac{13}{16},\frac{27}{32},\frac{7}{8},\frac{29}{32},\frac{15}{16},\frac{31}{32},1,1,1\right)$

[a] 1 = cross-section-circumferential direction; 2 = cross-section-radial direction; 3 = bend direction.