

# A Rapid and Efficient Isogeometric Design Space Exploration Framework with Application to Structural Mechanics

J. Benzaken, A. J. Herrema, M.-C. Hsu, J. A. Evans

## 1 Abstract

In this paper, we present an isogeometric analysis framework for design space exploration. While the methodology is presented in the setting of structural mechanics, it is applicable to any system of parametric partial differential equations. The design space exploration framework elucidates design parameter sensitivities used to inform initial and early-stage design. Moreover, this framework enables the visualization of a full system response, including the displacement and stress fields throughout the domain, by providing an approximation to the system solution vector. This is accomplished through a collocation-like approach where various geometries throughout the design space under consideration are sampled. The sampling scheme follows a quadrature rule while the physical solutions to these sampled geometries are obtained through an isogeometric method. A surrogate model to the design space solution manifold is constructed through either an interpolating polynomial or pseudospectral expansion. Examples of this framework are presented with applications to the Scordelis-Lo roof, a Flat L-Bracket, and an NREL 5MW wind turbine blade.

## 2 Introduction - Motivations

The field of computer aided engineering (CAE) has fostered a grandiose vision with regards to the progression of engineering design and analysis protocols, suggesting engineering systems can be iteratively optimized to obey specified performance constraints such as cost and load-bearing capability. However, the realization of this vision has been severely inhibited due to the growing demand for complex engineering systems along with the design-through-analysis bottleneck, that is, the apparent disconnect between design and analysis. Presently, iterative engineering optimization routines employ computational system models with highly simplified physics and geometry which ultimately result in an over-designed, sub-optimal final design.

Isogeometric analysis (IGA) is a computational approach that offers the possibility of integrating finite element analysis (FEA) into conventional computer aided design (CAD) tools [14, 34, 4]. In IGA, the geometric entities described by CAD software, namely NURBS and T-splines, serve as the basis for all subsequent engineering analyses. As the same geometric descriptions are used for both the FEA and CAD, models may be designed, tested, and adjusted in one integrated stage. Moreover, as the exact CAD model is employed in IGA at every stage in the design process, IGA yields accurate analysis results even during early-stage design.

As IGA provides a seamless integration between FEA and CAD, it constitutes an ideal analysis technology for design optimization. However, in early stage design, the phrase “design optimization” is platitudinous in that the problem of finding the “optimal design” is not well-defined. In fact, a primary objective of early stage design is to identify suitable optimization criteria in addition to design constraints. Determination of such criteria requires an understanding of how design parameters affect system response and ultimately performance. With this in mind, a framework enabling a designer to explore full system response in real-time would be invaluable.

In this paper, we present a design space exploration framework by unifying techniques emerging from the Uncertainty Quantification (UQ) community [59, 6, 54, 42, 51, 18] with isogeometric analysis [14, 34, 4] and parametric modeling [27, 7, 41, 32, 19] in a deterministic fashion. The framework allows a designer to “explore” various solutions fields derived from the displacement vector as a function of specified geometric design parameters for a given parametric family of geometries. This is accomplished by sampling the solution at a set of designs in a subset of the solution manifold during an offline stage. An intelligent sampling scheme combined with sparse collocation provides a means to dramatically reduce this computational cost [3, 44, 43, 24]. Afterwards, the model is constructed by either fitting an interpolating polynomial to sampled data or constructing a spectral

representation of the polynomial in this neighborhood.

It should be emphasized that with our design space exploration framework, a designer can visualize the full system response. That is, a designer can visualize the solution field across the physical domain for a particular geometry in the design space. This is in contrast with state-of-the-art design space exploration frameworks which only allow a designer to examine pre-selected quantities of interest.

The remainder of this paper is structured as follows. In Section 3, we review the basics of isogeometric analysis cast in the setting of linear elasticity and Kirchhoff-Love shells where the corresponding variational forms are derived and presented. This discussion continues with the introduction of geometric families, the means by which we relate design parameters to the parametric solution manifold. In Section 4, we present a set of schemes which can be used for sampling the design space in a manner that maximizes the fidelity of the surrogate model. Moreover, we discuss the various methods in which we subsequently employ either nodal or modal representations of the solution manifold. In Section 5, we present numerical examples of how the design space exploration methodology performs. In particular, we consider the Scordelis-Lo roof, a canonical problem from the “shell obstacle course,” a Flat L-Bracket, and an NREL 5MW wind turbine blade, which together demonstrates the effectiveness of the design space exploration paradigm for design problems with moderate dimensionality. Finally, in Section 6, we make concluding remarks as well as present future directions and applications of the design space exploration paradigm.

### 3 Parametrized Isogeometric Analysis for Geometric Families

In this section, we review the fundamentals of IGA including the representation of geometry and subsequent analysis which is performed on the intrinsic mesh present in the geometric description. Additionally, we introduce the concept of families of geometries which allows us to naturally parametrize designs through the IGA paradigm. Lastly, we ground these topics in example, where we present the variational form of several partial differential equations in parametric form for use in design space exploration.

#### 3.1 Fundamentals of Isogeometric Analysis

Let us begin by presenting the fundamental concepts underlying IGA including geometric representation and the numerical approximation to the solutions of partial differential equations via Galerkin’s method. Recall that in IGA, the basis employed for geometric definition in CAD is also used for analysis. Most engineering and free-form CAD software suites use the *Non-Uniform Rational B-Spline (NURBS)* basis for constructing geometry [48]. More recently, the advent of *T-splines* have served as a generalization to NURBS which permit IGA on “water-tight” geometries of arbitrary complexity [58, 57, 21, 39]. However, for simplicity of exposition, we restrict our discussion here to NURBS.

A NURBS function is a projective transformation of a B-spline function in one higher spatial dimension. Therefore, to motivate the discussion of IGA, we first present the construction of the B-spline basis. We generate B-spline basis functions in a parametric fashion by first constructing a *knot vector* which describes the support and continuity of the set of resulting basis functions. A knot vector is a non-decreasing set of real numbers  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$  where  $n$  is the number of basis functions and  $p$  is the polynomial degree. We assume without loss of generality that  $\xi_1 = 0$  and  $\xi_{n+p+1} = 1$ . In IGA, we typically employ an *open knot vector* such that the first and last knots are repeated  $p + 1$  times. **This ensures that the basis will interpolate the geometry at the boundaries in one-dimension, or the corners in higher dimensions.** The knot vector, along with the *Cox-deBoor formula* shown below, defines the univariate B-spline basis functions for intermediate  $\xi \in (0, 1)$ :

$$\hat{N}_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \hat{N}_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \hat{N}_{i+1,p-1}(\xi)$$
$$\hat{N}_{i,0}(\xi) = \begin{cases} 1, & \xi_i \leq \xi < \xi_{i+1} \\ 0, & \text{elsewhere} \end{cases}$$

The multivariate B-spline basis is obtained through a tensor product of one-dimensional basis functions. In particular, to define the  $d_p$ -dimensional B-spline basis functions for multi-indices  $\mathbf{i} = (i_1, i_2, \dots, i_{d_p})$  and  $\mathbf{p} = (p_1, p_2, \dots, p_{d_p})$  representing basis function number and polynomial degree respectively, we write:

$$\hat{N}_{\mathbf{i}, \mathbf{p}}(\boldsymbol{\xi}) = \bigotimes_{j=1}^{d_p} \hat{N}_{i_j, p_j}(\xi_j) \quad \forall \boldsymbol{\xi} \in \hat{\Omega}$$

where  $\hat{\Omega} = (0, 1)^{d_p}$  is the so-called *parametric domain*. Before proceeding to geometry definition in this setting, we must first discuss the construction of the NURBS basis from the B-spline basis presented above. Given a set of  $n_{\text{basis}} = \prod_{i=1}^{d_p} n_i$   $d_p$ -dimensional B-spline basis functions and a set of  $n_{\text{basis}}$  NURBS *weights*,  $w_i \in \mathbb{R}^+$ , we define the corresponding set of  $d_p$ -dimensional NURBS basis functions via:

$$\hat{R}_{\mathbf{i}, \mathbf{p}}(\boldsymbol{\xi}) = \frac{w_i \hat{N}_{\mathbf{i}, \mathbf{p}}(\boldsymbol{\xi})}{w(\boldsymbol{\xi})}, \quad \text{where} \quad w(\boldsymbol{\xi}) = \sum_{\mathbf{i}} w_i \hat{N}_{\mathbf{i}, \mathbf{p}}(\boldsymbol{\xi})$$

Now, in order to construct geometry using this basis and subsequently perform analysis on this geometry, we construct the *control mesh*, a network of  $n_{\text{basis}}$  *control points* in  $d_s$ -dimensions which, together with the complete set of NURBS basis functions, construct a  $d_s$ -dimensional geometry  $\Omega \subset \mathbb{R}^{d_s}$  which will serve as our *physical domain*. More specifically, given a set of NURBS control points  $\vec{P}_i$  and weights  $w_i$ , the parametric representation of the geometry  $\vec{\Phi}(\boldsymbol{\xi}) : \hat{\Omega} \rightarrow \Omega$  is given by:

$$\vec{\Phi}(\boldsymbol{\xi}) = \sum_{\mathbf{i}} \vec{P}_i \hat{R}_{\mathbf{i}, \mathbf{p}}(\boldsymbol{\xi}) \quad \forall \boldsymbol{\xi} \in \hat{\Omega} \quad (1)$$

Note we have dropped the subscript  $\mathbf{p}$  for notational ease, and we will proceed to do so henceforth. At this juncture, it is useful to delineate between the two different notions of dimension we have introduced heretofore. The dimension  $d_p$  refers to the *parametric dimension* (e.g.,  $d_p = 1$  for a curve,  $d_p = 2$  for a surface, and  $d_p = 3$  for a volume). Alternatively, the dimension  $d_s$  refers to the *spatial dimension*. It is required that  $d_p \leq d_s$ . For instance, a surface ( $d_p = 2$ ) may be embedded within  $\mathbb{R}^3$  but not within  $\mathbb{R}^1$ . Later, we will introduce the dimension  $d_\mu$  which will refer to the number of design parameters. Figure 1 shows an example of a NURBS curve in two-dimensional space and NURBS surface in three-dimensional space.

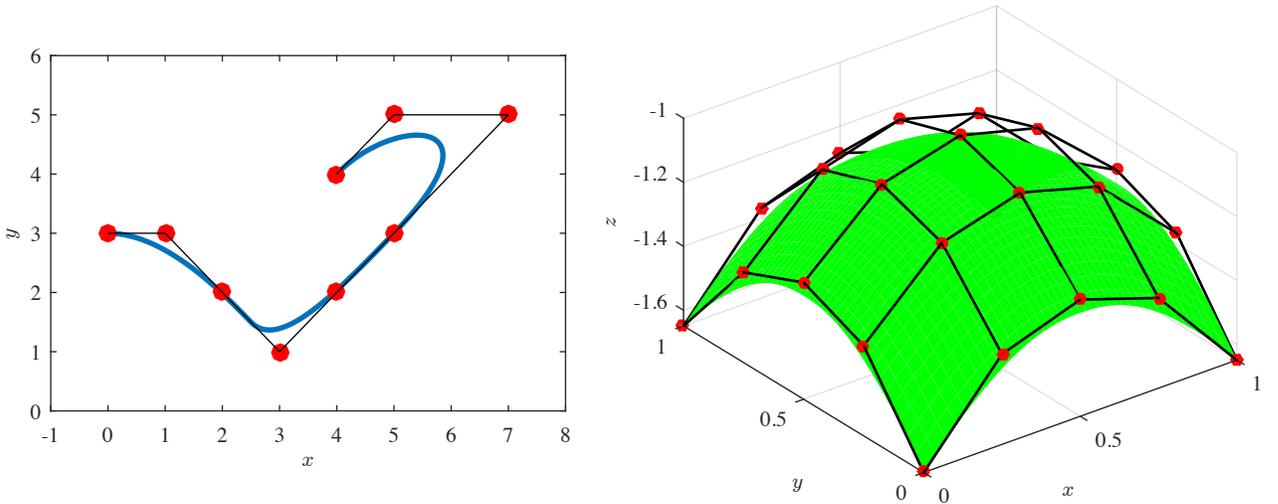


Figure 1: An example NURBS curve (left) and NURBS surface (right) with their corresponding control meshes.

In IGA, we seek a finite-dimensional approximation to a solution of a system of partial differential equations (PDEs). For simplicity, consider a general boundary-value problem (BVP) of the form:

$$\begin{aligned} \mathcal{L}(\vec{u}) &= \mathcal{F} \quad \forall \mathbf{x} \in \Omega \\ \mathcal{B}(\vec{u}) &= \mathcal{G} \quad \forall \mathbf{x} \in \partial\Omega \end{aligned} \quad (2)$$

where  $\mathcal{L}(\cdot)$  is a differential operator, possibly nonlinear, and  $\mathcal{B}(\cdot)$  is a boundary operator. Both operators act on the unknown solution field  $\vec{u} : \Omega \rightarrow \mathbb{R}^{n_{dof}}$ . In the context of structural mechanics,  $\vec{u}$  typically denotes the displacement field and  $n_{dof} = 2$  or  $3$ . As the PDE system is typically cast over a spatial variable, we require an appropriate space of basis functions defined in physical space in which to express the solution. To arrive at such a space, we leverage the isoparametric concept provided by the isogeometric analysis paradigm to uniquely relate topologically equivalent parametric and physical domains. The isoparametric concept is elucidated through the notion of the “pull back” and “push forward” operations which explicitly describe how the physical variable  $\mathbf{x}$  is related to the parametric variable  $\boldsymbol{\xi}$  through the geometric mapping:

$$\text{push forward: } \mathbf{x} = \vec{\Phi}(\boldsymbol{\xi}) \quad \text{pull back: } \boldsymbol{\xi} = \vec{\Phi}^{-1}(\mathbf{x}) \quad (3)$$

Hence, the NURBS basis functions defined in the parametric domain also define splines in the physical domain as directed by the geometric mapping (1):

$$R_i(\mathbf{x}) = \hat{R}_i\left(\vec{\Phi}^{-1}(\mathbf{x})\right) \quad \text{and} \quad \hat{R}_i(\boldsymbol{\xi}) = R_i\left(\vec{\Phi}(\boldsymbol{\xi})\right) \quad (4)$$

In IGA, we use the same basis for design and analysis, so we seek a discrete solution of the form:

$$\vec{u}^h(\mathbf{x}) = \sum_i \vec{d}_i R_i(\mathbf{x})$$

where the terms  $\vec{d}_i \in \mathbb{R}^{n_{dof}}$  are referred to as *control variables*. With the admissible trial functions defined, we invoke Galerkin’s method by multiplying the PDE system by a corresponding NURBS weighting function, integrating by parts, strongly enforcing Dirichlet boundary conditions, and weakly enforcing Neumann boundary conditions. This yields a residual system of the form:

$$\mathbf{R}(\mathbf{d}) = \mathbf{0}$$

where  $\mathbf{R}$  is a vector of residuals and  $\mathbf{d}$  is a solution vector of control variables. In the linear setting, this equation reduces further to the linear system:

$$\mathbf{K}\mathbf{d} = \mathbf{F}$$

where  $\mathbf{K}$  is the system stiffness matrix and  $\mathbf{F}$  is the system forcing vector. To assemble and solve this system, finite elements are constructed through a process known as “Bézier extraction” where a transformation, referred to as the extraction operator, is constructed that describes the B-spline basis locally in terms of the Bernstein polynomials [8, 56]. These elements are then assembled in a global stiffness matrix and a global system solve is performed to obtain the displacement vector. For an in-depth overview of IGA and its implementation, the reader is referred to [14, Chapter 3].

In the following section, we explore processes in which a changing geometric definition propagates and affects the resulting displacement field. The isoparametric concept will allow us to pose the variational form of a partial differential equation in terms of the physical basis functions, rendering the control variable a function solely of geometry.

## 3.2 Parametrized Isogeometric Analysis for Families of Geometries

The seamless connection between CAD geometry and analysis in IGA allows us to define a unique relationship between *design parameters* and the corresponding CAD geometry. These parameters define the control points and weights associated with a geometric configuration. However it should be noted that, in general, control points are not individual design parameters to be tuned. Instead, we seek a reduced set of these design parameters which describe *features* and the overall shape of the design, in contrast to specific control point locations. For instance, these design parameters may be the radius of hole or fillet, the thickness of a load-bearing plate, or the Lamé parameters which relate stress and strain through a constitutive relationship.

Let us now formalize the above concepts. Namely, let us construct a *design space*  $\mathcal{D} \subset \mathbb{R}^{d_\mu}$ , where  $d_\mu$  is the dimension of the parametric space associated with design parameters. We refer to each member of  $\mathcal{D}$ , denoted

as  $\boldsymbol{\mu} \in \mathcal{D}$ , as a *design variable*, and it contains a selection of design parameters governing the material and geometric properties for a given design as discussed above. We assume throughout the paper that  $\mathcal{D}$  is a hyperrectangle, that is, a Cartesian product of intervals:  $\mathcal{D} = (a_1, b_1) \times (a_2, b_2) \times \dots \times (a_{d_\mu}, b_{d_\mu})$ . This will allow us to easily adopt existing sampling-based surrogate modeling approaches as shown in Section 4. With the above terminology established, we connect a CAD geometry to the design space through the notion of a family of geometries:

**Definition (Family of Geometries):** A family of geometries  $\{\Omega_\mu\}_{\mu \in \mathcal{D}}$  is a set of open domains  $\Omega_\mu \subset \mathbb{R}^{d_\mu}$  defined from a family of NURBS mappings:

$$\vec{\Phi}_\mu(\boldsymbol{\xi}) = \sum_i \vec{P}_i(\boldsymbol{\mu}) \hat{R}_i(\boldsymbol{\xi}) \quad \forall \boldsymbol{\xi} \in \hat{\Omega}$$

That is,  $\Omega_\mu = \vec{\Phi}_\mu(\hat{\Omega})$  for every  $\mu \in \mathcal{D}$ .

Note that in the above definition, we have implicitly defined the control mesh to be a function of the design variable. To make the concept of a family of geometries clear, consider the example of a Scordelis-Lo roof, as illustrated in Figure 2. The design variable:

$$\boldsymbol{\mu} = (L, R, \varphi, t, E, \nu) \quad (5)$$

completely describes the Scordelis-Lo Roof family of geometries. In other words, we are able to generate the unique control points and weights as a function of the design variable (5). See Appendix A for more details.

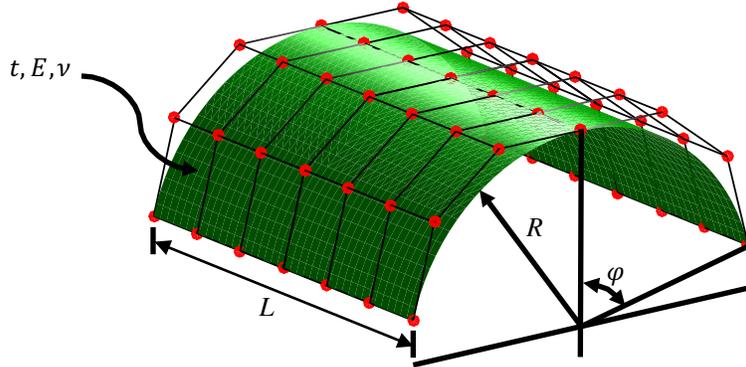


Figure 2: The Scordelis-Lo Roof Family

In parametrized IGA for a family of geometries, we consider the effect of changing geometry on the resulting solution. In particular, we parametrize the partial differential equation (2) in terms of the design variable  $\boldsymbol{\mu}$ :

$$\begin{aligned} \mathcal{L}(\vec{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) &= \mathcal{F}(\boldsymbol{\mu}) \quad \forall \mathbf{x} \in \Omega_\mu \\ \mathcal{B}(\vec{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) &= \mathcal{G}(\boldsymbol{\mu}) \quad \forall \mathbf{x} \in \partial\Omega_\mu \end{aligned} \quad (6)$$

Leveraging the concepts (3) and (4), we claim that the discrete solution to such a problem can be expressed as:

$$\vec{u}^h(\mathbf{x}, \boldsymbol{\mu}) = \sum_i \vec{d}_i(\boldsymbol{\mu}) R_i(\mathbf{x}) = \sum_i \underbrace{\vec{d}_i(\boldsymbol{\mu})}_{\text{unknown}} \underbrace{\hat{R}_i(\vec{\Phi}_\mu^{-1}(\mathbf{x}))}_{\text{known}} \quad (7)$$

The effect of  $\boldsymbol{\mu}$  on the physical basis functions is known through the isoparametric concept. With the concept of geometric families defined, the terms  $\vec{d}_i(\boldsymbol{\mu})$  in (7) are determined by a set of nonlinear algebraic equations:

$$\mathbf{R}(\boldsymbol{\mu}, \mathbf{d}(\boldsymbol{\mu})) = \mathbf{0} \quad \forall \boldsymbol{\mu} \in \mathcal{D} \quad (8)$$

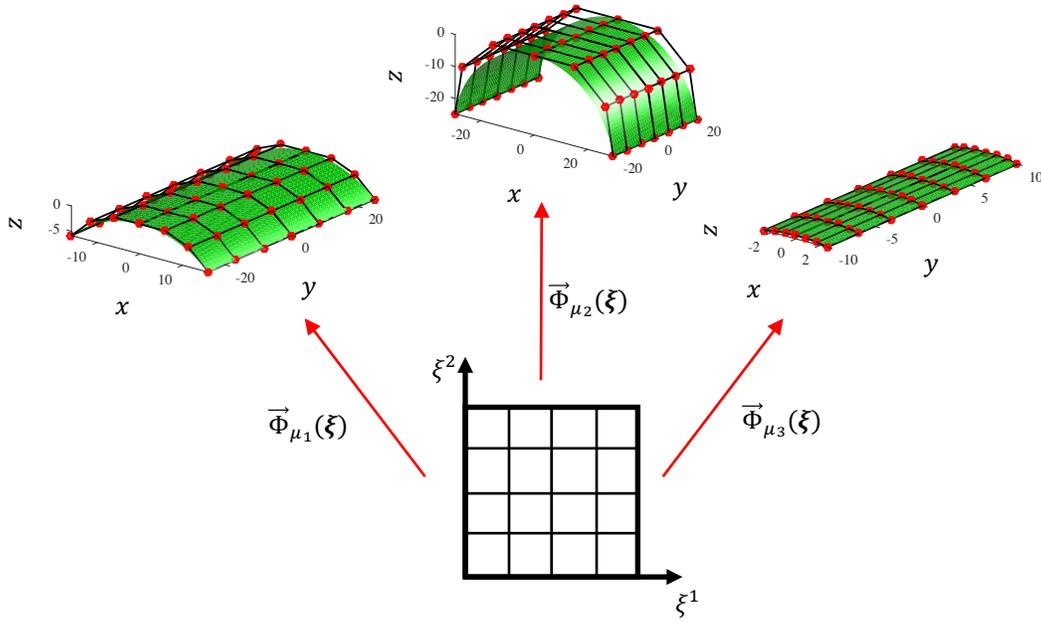


Figure 3: A representative sample of Scordelis-Lo Roof geometries belonging to the same family. Notice that after defining the appropriate mapping  $\vec{\Phi}_{\mu_i}(\xi)$ , the same parametric domain can be mapped to various geometries seamlessly.

where  $\mathbf{R}(\boldsymbol{\mu}, \mathbf{d}(\boldsymbol{\mu}))$  is a vector of residuals and  $\mathbf{d}(\boldsymbol{\mu})$  is a solution vector collecting the unknown control variables. In the linear setting, this system reduces to solving the system:

$$\mathbf{K}(\boldsymbol{\mu}) \mathbf{d}(\boldsymbol{\mu}) = \mathbf{F}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D} \quad (9)$$

The discretized PDE systems (8) and (9) provides a vessel to explore the design space. Namely, for every  $\boldsymbol{\mu}$ , the solution  $\mathbf{d}(\boldsymbol{\mu})$  is the parametric system response. However, every time  $\boldsymbol{\mu}$  is changed, one must not only solve the system, but also assemble it. For an effective rapid design space exploration paradigm, we want to construct a framework for determining the solution for a family of geometries in a neighborhood about some nominal design, rather than for one particular design, so it is greatly desirable to reduce the computational expense of solving (8) and (9).

There exists two primary approaches to reduce this computational expense through the construction of suitable surrogate models:

1. Dimensionality Reduction of the Spatial Approximation: In this instance, a linear system of the form (9) for every parametric instance of  $\boldsymbol{\mu}$  is assembled and solved. However, the associated computational expense is reduced by employing a reduced-order model. This approach utilizes techniques such as reduced basis methods [40] and proper orthogonal decompositions [50] for the surrogate model construction.
2. Dimensionality Reduction of the Parametric Representation: In this instance, rather than solving a linear system for every parametric instance of  $\boldsymbol{\mu}$ , we solve the full system (9) for a set  $\{\boldsymbol{\mu}_i\}_i$  and extrapolate the obtained solutions throughout the design space to effectively construct a surrogate model.

In this paper, we consider the second approach listed above due to ease of implementation. More specifically, the second approach allows us to build sampling-based surrogate models using a wrapper about existing IGA simulation technologies. It should be mentioned that the parametrized analysis paradigm that we have presented here is quite difficult to implement within a classical finite element analysis framework. This is because changes to the geometry require a complete re-meshing within standard design-through-analysis platforms built on finite elements. Consequently, most existing design space exploration environments allow users to visualize only

quantities of interest such as max displacement, von Mises stress, or power consumption [36, 60]. By contrast, our environment allows users to visualize full system response.

Before further discussing the construction of our surrogate model, we ground the notion of parametrized IGA in example. We begin by considering the application of these concepts to linear elasticity.

### 3.2.1 Application to Linear Elasticity

In linear elasticity, the components of the infinitesimal strain tensor are given by the symmetric part of the gradient of the displacement field:

$$\tilde{\varepsilon}_{\alpha\beta}(\vec{u}) = \frac{1}{2} \left( \frac{\partial u^\beta}{\partial x^\alpha} + \frac{\partial u^\alpha}{\partial x^\beta} \right) \quad (10)$$

where  $\vec{u}$  is the Cartesian displacement field. With an appropriate material model such as:

$$D^{\alpha\beta\lambda\nu}(\boldsymbol{\mu}) = \frac{E(\boldsymbol{\mu})}{2(1+\nu(\boldsymbol{\mu}))} \left( \delta^{\alpha\lambda}\delta^{\beta\nu} + \delta^{\alpha\nu}\delta^{\beta\lambda} + \frac{2\nu(\boldsymbol{\mu})}{1-\nu(\boldsymbol{\mu})}\delta^{\alpha\beta}\delta^{\lambda\nu} \right) \quad (11)$$

where  $\delta^{\alpha\beta}$  is the Kronecker delta, we can relate the stress to the strain, and furthermore displacement, via the generalized Hooke's law:

$$\sigma^{\alpha\beta}(\boldsymbol{\mu}) = D^{\alpha\beta\lambda\nu}(\boldsymbol{\mu})\tilde{\varepsilon}_{\lambda\nu}$$

The strong form of the parametric system of partial differential equations governing linear elasticity is given in terms of internal stresses by:

**Strong Form (Linear Elasticity):** For every  $\boldsymbol{\mu} \in \mathcal{D}$ , find  $\vec{u}(\mathbf{x}) \in C^1(\Omega_\mu)$  such that:

$$\begin{aligned} \sigma_{,\beta}^{\alpha\beta} &= F_\mu^\alpha, & \mathbf{x} &\in \Omega_\mu \\ u^\alpha &= g_\mu^\alpha, & \mathbf{x} &\in \Gamma_{D_\mu} \\ \sigma^{\alpha\beta}(n_\mu)_\beta &= h_\mu^\alpha, & \mathbf{x} &\in \Gamma_{N_\mu} \end{aligned}$$

where  $F_\mu^\alpha$  is the  $\alpha^{th}$  Cartesian component of the parametrized forcing,  $g_\mu^\alpha$  is the  $\alpha^{th}$  component of the parametrized Dirichlet boundary condition over the parametric Dirichlet boundary  $\Gamma_{D_\mu}$ , and  $h_\mu^\alpha$  is the  $\alpha^{th}$  component of the parametrized Neumann boundary condition over the parametric Neumann boundary  $\Gamma_{N_\mu}$ . Note that here,  $(n_\mu)_\beta$  are the Cartesian components of the normal director of  $\Gamma_{N_\mu}$ .

In most cases, the strong form is in fact too strong for numerical considerations. Therefore we seek the weak solution to this problem by invoking the principle of virtual work. In particular, we consider the strain field  $\varepsilon_{\alpha\beta}(\vec{u})$  which minimizes the potential energy configuration of the system against a space of test functions. The *trial* and *test* spaces for the weak linear elasticity problem are defined in their parametric form as:

$$\mathcal{S}_{LE}(\boldsymbol{\mu}) := \left\{ \vec{u} : \Omega_\mu \rightarrow \mathbb{R}^{d_s} \mid \vec{u} \in (\mathcal{H}^1(\Omega_\mu))^{d_s} \text{ and } \vec{u}|_{\Gamma_{D_\mu}} = \vec{g}_\mu \right\}$$

and

$$\mathcal{V}_{LE}(\boldsymbol{\mu}) := \left\{ \vec{w} : \Omega_\mu \rightarrow \mathbb{R}^{d_s} \mid \vec{w} \in (\mathcal{H}^1(\Omega_\mu))^{d_s} \text{ and } \vec{w}|_{\Gamma_{D_\mu}} = 0 \right\}$$

respectively. Note however, the solution  $\vec{u}$  is a function of both the spatial variable  $\mathbf{x}$  and the design variable  $\boldsymbol{\mu}$ . Hence, as  $\boldsymbol{\mu}$  changes, so do the test and trial spaces which govern the isogeometric solution. The lack of tensor-product structure between  $\mathbf{x}$  and  $\boldsymbol{\mu}$ , e.g.,  $\mathbf{x} = \mathbf{x}(\boldsymbol{\mu})$ , prohibits us from explicitly defining a tensor-product space which encompasses the solution's dependencies. Thus, we define the following space as the domain of our solution for  $\mathbf{x} \in \mathbb{R}^{d_s}$  and  $\boldsymbol{\mu} \in \mathbb{R}^{d_\mu}$  through a group variable:

$$Q := \left\{ (\mathbf{x}, \boldsymbol{\mu}) \in \mathbb{R}^{d_s+d_\mu} \mid \mathbf{x} \in \Omega_\mu, \boldsymbol{\mu} \in \mathcal{D} \right\}$$

With this definition, we can define the trial space in which our parametric solution resides:

$$\mathcal{S}_{LE} := \left\{ \vec{u} : Q \rightarrow \mathbb{R}^{d_s} \mid \vec{u}(\cdot, \boldsymbol{\mu}) \in \mathcal{S}_{LE}(\boldsymbol{\mu}) \right\}$$

Then, the variational form of this partial differential equation is given by the  $L^2$  inner product with an arbitrary test function  $\vec{w}(\cdot, \boldsymbol{\mu}) \in \mathcal{V}_{LE}(\boldsymbol{\mu})$  followed by an integration by parts. This allows us to express the partial differential equation in variational form as:

**Variational Form (Linear Elasticity):** Find  $\vec{u} \in \mathcal{S}_{LE}$  such that:

$$a_{LE}(\vec{w}, \vec{u}(\cdot, \boldsymbol{\mu}); \boldsymbol{\mu}) = L_{LE}(\vec{w}; \boldsymbol{\mu}) \quad \forall \vec{w} \in \mathcal{V}_{LE}(\boldsymbol{\mu})$$

for all  $\boldsymbol{\mu} \in \mathcal{D}$  where:

$$a_{LE}(\vec{w}, \vec{v}; \boldsymbol{\mu}) = \int_{\Omega_{\boldsymbol{\mu}}} D^{\alpha\beta\lambda\nu}(\boldsymbol{\mu}) \tilde{\varepsilon}_{\alpha\beta}(\vec{w}) \tilde{\varepsilon}_{\lambda\nu}(\vec{v}) d\Omega_{\boldsymbol{\mu}} \quad \forall \vec{w} \in \mathcal{V}_{LE}(\boldsymbol{\mu}), \vec{v} \in \mathcal{S}_{LE}(\boldsymbol{\mu})$$

and

$$L_{LE}(\vec{w}; \boldsymbol{\mu}) = \int_{\Omega_{\boldsymbol{\mu}}} w_{\alpha} F_{\boldsymbol{\mu}}^{\alpha} d\Omega_{\boldsymbol{\mu}} + \int_{\Gamma_{N_{\boldsymbol{\mu}}}} w_{\alpha} h_{\boldsymbol{\mu}}^{\alpha} d\Gamma_{N_{\boldsymbol{\mu}}} \quad \forall \vec{w} \in \mathcal{V}_{LE}(\boldsymbol{\mu})$$

For isogeometric implementation, we must convert the above weak formulations into a system of algebraic equations. This is accomplished through the application of Galerkin's method where we work in the finite-dimensional subspaces  $\mathcal{S}_{LE}^h(\boldsymbol{\mu}) \subset \mathcal{S}_{LE}(\boldsymbol{\mu})$  and  $\mathcal{V}_{LE}^h(\boldsymbol{\mu}) \subset \mathcal{V}_{LE}(\boldsymbol{\mu})$ . These spaces are defined using the NURBS basis scaled by vector-valued control variables. In particular, the trial and test spaces are defined as:

$$\mathcal{S}_{LE}^h(\boldsymbol{\mu}) := \left\{ \vec{u}^h : \Omega_{\boldsymbol{\mu}} \rightarrow \mathbb{R}^{d_s} \mid \vec{u}^h(\mathbf{x}) = \sum_{\mathbf{i}} \vec{d}_{\mathbf{i}} R_{\mathbf{i}}(\mathbf{x}) \text{ and } \vec{u}^h|_{D_{\boldsymbol{\mu}}} = \vec{g}_{\boldsymbol{\mu}}^h \right\}$$

$$\mathcal{V}_{LE}^h(\boldsymbol{\mu}) := \left\{ \vec{w}^h : \Omega_{\boldsymbol{\mu}} \rightarrow \mathbb{R}^{d_s} \mid \vec{w}^h(\mathbf{x}) = \sum_{\mathbf{i}} \vec{c}_{\mathbf{i}} R_{\mathbf{i}}(\mathbf{x}) \text{ and } \vec{w}^h|_{D_{\boldsymbol{\mu}}} = 0 \right\}$$

where we note that the splines in this space must be at least  $C^0$ -continuous. Moreover, the finite-dimensional solution to the parametric partial differential equation is a member of the space:

$$\mathcal{S}_{LE}^h := \left\{ \vec{u}^h : Q \rightarrow \mathbb{R}^{d_s} \mid \vec{u}^h(\cdot, \boldsymbol{\mu}) \in \mathcal{S}_{LE}^h(\boldsymbol{\mu}) \right\}$$

Additionally, note that splines in this space are of the form:

$$\vec{u}(\mathbf{x}, \boldsymbol{\mu}) = \sum_{\mathbf{i}} \vec{d}_{\mathbf{i}}(\boldsymbol{\mu}) R_{\mathbf{i}}(\mathbf{x})$$

To obtain the Galerkin form of the PDE, we perform the  $L^2$  inner product between members of the finite-dimensional test and trial spaces as defined above.

**Galerkin's Method (Linear Elasticity):** Find  $\vec{u}^h \in \mathcal{S}_{LE}^h$  such that:

$$a_{LE}(\vec{w}^h, \vec{u}^h(\cdot, \boldsymbol{\mu}); \boldsymbol{\mu}) = L_{LE}(\vec{w}^h; \boldsymbol{\mu}) \quad \forall \vec{w}^h \in \mathcal{V}_{LE}^h(\boldsymbol{\mu})$$

for all  $\boldsymbol{\mu} \in \mathcal{D}$ . This amounts to solving the matrix system:

$$\mathbf{K}_{LE}(\boldsymbol{\mu}) \mathbf{d}(\boldsymbol{\mu}) = \mathbf{F}_{LE}(\boldsymbol{\mu})$$

where

$$[\mathbf{K}_{LE}(\boldsymbol{\mu})]_{PQ} = a_{LE}(R_i \hat{e}_A, R_j \hat{e}_B; \boldsymbol{\mu})$$

and

$$[\mathbf{F}_{LE}(\boldsymbol{\mu})]_P = L_{LE}(R_i \hat{e}_A; \boldsymbol{\mu})$$

where  $P, Q$  are associated with an indexing scheme returning a global row number for each degree of freedom  $A$  and basis function  $i$ . See [33, Chapter 2] for more details.

### 3.2.2 Application to Kirchhoff-Love Shell Analysis

We now continue by considering the application of parametrized IGA to thin shell structures with specific reference to Kirchhoff-Love shells. As opposed to the previous application of linear elasticity, the constitutive laws appearing in thin shell analysis depend on geometric design parameters. Nonetheless, our proposed design space exploration framework naturally extends to this problem setting.

As a shell structure is significantly thinner in one direction compared to the other two, through-thickness effects may be idealized. Rather than modeling the shell structure as a 3-dimensional elastic entity, the geometry is represented as a 2-dimensional surface, known as the midsurface. For large strains and displacements, the nonlinear, Green-Lagrange finite strain tensor is used while in the case of small strains and displacements, the infinitesimal Green-Lagrange strain tensor is employed. Moreover in the case of the former, implementation requires an iterative minimization routine while the latter only requires one global system construction and linear solve.

To proceed, we consider the *shell midsurface mapping*  $\vec{\Phi}_{\boldsymbol{\mu}}$  in its parametric form. This map describes how the *shell midsurface* is embedded in  $\mathbb{R}^3$  and takes the form of the geometric mapping (1) in the NURBS setting. The physical shell structure has thickness  $t = t(\boldsymbol{\mu})$  for which the thickness variable  $\xi^3 \in [-t/2, t/2]$ . Note here that we have assumed the shell thickness to be a function of the design parameters. In the general setting, the shell mapping is nonlinear hence intrinsic curvatures are present in the shell manifold  $\Omega_{\boldsymbol{\mu}}$ . These curvatures are necessary to accurately describe general shell structures (e.g., aircraft skin panels and automobile bodies) however they present difficulty in analysis since the material coordinate system is generally not aligned with any *global* coordinate system. Therefore we resort to using the parametric basis described by the NURBS parameterization of the midsurface as the basis for analysis, granting the ability to discuss the Kirchhoff-Love shell model and its implementation. This parametric midsurface basis, used to define the shell basis for analysis, is given by the in-plane covariant vectors and the normal director obtained through their cross product:

$$\vec{A}_{\alpha}(\xi^1, \xi^2; \boldsymbol{\mu}) = \frac{\partial \vec{\Phi}_{\boldsymbol{\mu}}(\xi^1, \xi^2)}{\partial \xi^{\alpha}} \quad \text{and} \quad \vec{A}_3(\xi^1, \xi^2; \boldsymbol{\mu}) = \frac{\vec{A}_1(\xi^1, \xi^2; \boldsymbol{\mu}) \times \vec{A}_2(\xi^1, \xi^2; \boldsymbol{\mu})}{\|\vec{A}_1(\xi^1, \xi^2; \boldsymbol{\mu}) \times \vec{A}_2(\xi^1, \xi^2; \boldsymbol{\mu})\|}$$

The shell mapping is then given by:

$$\vec{X}_{\boldsymbol{\mu}}(\xi^1, \xi^2, \xi^3) = \vec{\Phi}_{\boldsymbol{\mu}}(\xi^1, \xi^2) + \xi^3 \vec{A}_3(\xi^1, \xi^2; \boldsymbol{\mu})$$

where the subscript  $\boldsymbol{\mu}$  indicates the dependence of the mapping on the design variable. With this mapping defined, the parametric shell basis is given by the various derivatives of this mapping:

$$\vec{G}_i(\xi^1, \xi^2, \xi^3; \boldsymbol{\mu}) = \frac{\partial \vec{X}_{\boldsymbol{\mu}}(\xi^1, \xi^2, \xi^3)}{\partial \xi^i} \quad (12)$$

The unknown displacement field is given by the difference between the deformed geometry  $\vec{x}_{\boldsymbol{\mu}}$  and the undeformed geometry  $\vec{X}_{\boldsymbol{\mu}}$ :

$$\vec{U}_{\boldsymbol{\mu}}(\xi^1, \xi^2, \xi^3) = \vec{x}_{\boldsymbol{\mu}}(\xi^1, \xi^2, \xi^3) - \vec{X}_{\boldsymbol{\mu}}(\xi^1, \xi^2, \xi^3) = \vec{u}_{\boldsymbol{\mu}}(\xi^1, \xi^2) + \xi^3 \vec{\theta}_{\boldsymbol{\mu}}(\xi^1, \xi^2)$$

where  $\vec{u}_{\boldsymbol{\mu}}$  and  $\vec{\theta}_{\boldsymbol{\mu}}$  represent the translational and rotational displacements of the midsurface, respectively.

To complete the problem definition for thin shell structures, we must relate the stress field to a measure of strain. The Green-Lagrange strain tensor is conjugate to the second Piola-Kirchhoff stress tensor which is used to determine the minimal potential energy configuration associated with the structural deformation of the shell under some external loading. With this representation  $\vec{U}_\mu(\xi^1, \xi^2, \xi^3)$  of a general shell displacement, we can construct the nonlinear global stiffness with the Green-Lagrange finite strain tensor given by:

$$\varepsilon_{ij}(\vec{U}_\mu; \boldsymbol{\mu}) = \frac{1}{2} \left( \vec{G}_i(\boldsymbol{\mu}) \cdot \frac{\partial \vec{U}_\mu}{\partial \xi^j} + \vec{G}_j(\boldsymbol{\mu}) \cdot \frac{\partial \vec{U}_\mu}{\partial \xi^i} + \left( \vec{G}_k(\boldsymbol{\mu}) \cdot \frac{\partial \vec{U}_\mu}{\partial \xi^i} \right) \left( \vec{G}_k(\boldsymbol{\mu}) \cdot \frac{\partial \vec{U}_\mu}{\partial \xi^j} \right) \right) \quad (13)$$

For a geometrically nonlinear shell, we then relate the second Piola-Kirchhoff stress tensor to the Green-Lagrange strain tensor by:

$$D^{\alpha\beta\lambda\nu}(\boldsymbol{\mu}) = \frac{E(\boldsymbol{\mu})}{2(1+\nu(\boldsymbol{\mu}))} \left( G^{\alpha\lambda}(\boldsymbol{\mu})G^{\beta\nu}(\boldsymbol{\mu}) + G^{\alpha\nu}(\boldsymbol{\mu})G^{\beta\lambda}(\boldsymbol{\mu}) + \frac{2\nu(\boldsymbol{\mu})}{1-\nu(\boldsymbol{\mu})} G^{\alpha\beta}(\boldsymbol{\mu})G^{\lambda\nu}(\boldsymbol{\mu}) \right) \quad (14)$$

where in this setting the covariant basis vectors defined via (12), are used to construct the metric tensor:

$$G_{\alpha\beta}(\boldsymbol{\mu}) = \vec{G}_\alpha(\boldsymbol{\mu}) \cdot \vec{G}_\beta(\boldsymbol{\mu}), \quad G^{\alpha\beta}(\boldsymbol{\mu}) = [G_{\alpha\beta}(\boldsymbol{\mu})]^{-1}$$

which is used in the material model definition (14), hence the large dependency on  $\boldsymbol{\mu}$ .

The Kirchhoff-Love shell assumption is that the normal director of the midsurface remains straight and normal to the deformed midsurface, i.e., there is no transverse shear strain present in the deformed structure. Figure 4 depicts the displacement profile dictated by the Kirchhoff-Love shell paradigm. This assumption enforces a coupling between the translational and rotational degrees of freedom in the system through a gradient. As a consequence, the displacement field, hence strain and stress fields, can be expressed entirely in terms of translational midsurface displacements. Moreover it permits a decomposition of the strain tensor into purely surface entities, particularly the membrane strain  $\gamma_{\alpha\beta}(\boldsymbol{\mu})$  and bending strain  $\chi_{\alpha\beta}(\boldsymbol{\mu})$  tensors, which are described by the change in first and second fundamental forms, respectively. With a shell material model (14), the stress in the shell model may also be separated into two symmetric tensors, the membrane stress tensor  $n^{\alpha\beta}(\boldsymbol{\mu})$  and the bending moment tensor  $m^{\alpha\beta}(\boldsymbol{\mu})$ . The principle of virtual work is then invoked and, after integration through-thickness, the global system is constructed and subsequently minimized using a root-finding method such as Newton-Raphson, yielding the control variables which provide the nonlinear structural response. For brevity, we discuss only high-level details of the derivation and the reader is referred to [37] for more details.

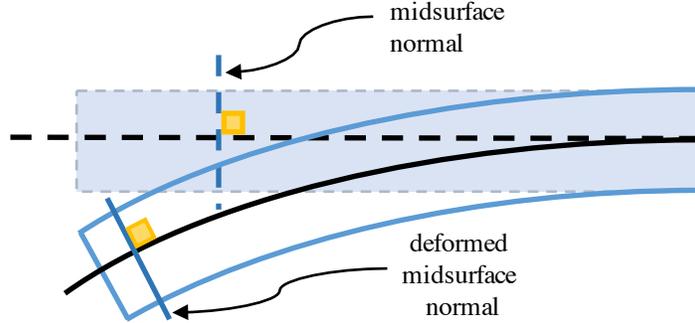


Figure 4: The undeformed and deformed Kirchhoff-Love shell. Observe that the midsurface normal remains normal to the deformed midsurface.

The Kirchhoff-Love shell may also be linearized. In this case, a similar construction for the global stiffness is performed where a linearization of the Green-Lagrange finite strain tensor occurs, dropping the last term of (13), resulting in a tensor which resembles the infinitesimal Cartesian Green-Lagrange strain tensor (10). This process yields a weak form where a membrane strain term, identical to the geometrically nonlinear Kirchhoff-Love shell, appears however the linearized curvature tensor  $\rho_{\alpha\beta}(\boldsymbol{\mu})$  appears in lieu of the bending strain tensor. The reader is referred to [11] for a discussion of linear shell models.

With the above terminology established, we are now ready to state the variational form of the Kirchhoff-Love shell problem. We present only the discrete setting as the continuous problem is analogous. The *trial* and *test* spaces for the Galerkin form of the Kirchhoff-Love shell are defined as:

$$\mathcal{S}_{KL}^h(\boldsymbol{\mu}) := \left\{ \vec{u}^h : \Omega_{\boldsymbol{\mu}} \rightarrow \mathbb{R}^{d_s} \mid \vec{u}^h(\mathbf{x}) = \sum_{\mathbf{i}} \vec{d}_{\mathbf{i}} R_{\mathbf{i}}(\mathbf{x}) \text{ and } \vec{u}|_{\Gamma_{D_{\boldsymbol{\mu}}}} = \vec{g}_{\boldsymbol{\mu}}^h \right\}$$

and

$$\mathcal{V}_{KL}^h(\boldsymbol{\mu}) := \left\{ \vec{w}^h : \Omega_{\boldsymbol{\mu}} \rightarrow \mathbb{R}^{d_s} \mid \vec{w}^h(\mathbf{x}) = \sum_{\mathbf{i}} \vec{c}_{\mathbf{i}} R_{\mathbf{i}}(\mathbf{x}) \text{ and } \vec{w}|_{\Gamma_{D_{\boldsymbol{\mu}}}} = 0 \right\}$$

respectively, where  $\Gamma_{D_{\boldsymbol{\mu}}}$  denotes to the Dirichlet boundary of the geometric domain  $\Omega_{\boldsymbol{\mu}}$ . We only consider the case of weakly-clamped shells so that  $\Gamma_{N_{\boldsymbol{\mu}}} = \emptyset$ . Note that splines in the above spaces are required to be at least  $C^1$ -continuous so that the resulting strain field is well-defined. Analogous to the case of linear elasticity, we define the finite-dimensional trial space in which our parametric solution lives via:

$$\mathcal{S}_{KL}^h := \{ \vec{u}^h : Q \rightarrow \mathbb{R}^{d_s} \mid \vec{u}^h(\cdot, \boldsymbol{\mu}) \in \mathcal{S}_{KL}^h(\boldsymbol{\mu}) \}$$

Then, the discrete Kirchhoff-Love problem is defined as:

**Galerkin's Method (Kirchhoff-Love shell):** Find  $\vec{u}^h \in \mathcal{S}_{KL}^h$  such that:

$$a_{KL}(\vec{w}^h; \vec{u}^h(\cdot, \boldsymbol{\mu}), \boldsymbol{\mu}) = L_{KL}(\vec{w}^h; \boldsymbol{\mu}) \quad \forall \vec{w}^h \in \mathcal{V}_{KL}^h(\boldsymbol{\mu})$$

for all  $\boldsymbol{\mu} \in \mathcal{D}$  where:

$$a_{KL}(\vec{w}^h; \vec{u}^h, \boldsymbol{\mu}) = \begin{cases} \int_{\Omega_{\boldsymbol{\mu}}} D^{\alpha\beta\lambda\nu}(\boldsymbol{\mu}) \left[ t(\boldsymbol{\mu}) \gamma_{\alpha\beta}(\vec{u}^h; \boldsymbol{\mu}) \gamma_{\lambda\nu}(\vec{w}^h; \boldsymbol{\mu}) + \frac{(t(\boldsymbol{\mu}))^3}{12} \chi_{\alpha\beta}(\vec{u}^h; \boldsymbol{\mu}) \chi_{\lambda\nu}(\vec{w}^h; \boldsymbol{\mu}) \right] d\Omega_{\boldsymbol{\mu}} & \text{Geometrically Nonlinear Kirchhoff-Love Shell} \\ \int_{\Omega_{\boldsymbol{\mu}}} D^{\alpha\beta\lambda\nu}(\boldsymbol{\mu}) \left[ t(\boldsymbol{\mu}) \gamma_{\alpha\beta}(\vec{u}^h; \boldsymbol{\mu}) \gamma_{\lambda\nu}(\vec{w}^h; \boldsymbol{\mu}) + \frac{(t(\boldsymbol{\mu}))^3}{12} \rho_{\alpha\beta}(\vec{u}^h; \boldsymbol{\mu}) \rho_{\lambda\nu}(\vec{w}^h; \boldsymbol{\mu}) \right] d\Omega_{\boldsymbol{\mu}} & \text{Linear Kirchhoff-Love Shell} \end{cases}$$

and

$$L_{KL}(\vec{w}; \boldsymbol{\mu}) = \int_{\Omega_{\boldsymbol{\mu}}} t(\boldsymbol{\mu}) w_{\alpha} F_{\boldsymbol{\mu}}^{\alpha} d\Omega_{\boldsymbol{\mu}}$$

Note that a linear system, similar to that of linear elasticity, arises from implementation of the linear Kirchhoff-Love shell. In the case of the nonlinear Kirchhoff-Love shell, a iterative minimization routine is employed to reduce the resulting residual which arises from the weak form presented above.

To demonstrate the broad applicability of the design space parameterization (6), we present these parametric PDEs in a consolidated manner.

**Galerkin's Method (Parametric PDE):** Find  $\vec{u}^h \in \mathcal{S}_A^h$  such that:

$$a_A(\vec{w}^h; \vec{u}^h(\cdot, \boldsymbol{\mu}), \boldsymbol{\mu}) = L_A(\vec{w}^h, \boldsymbol{\mu}) \quad \forall \vec{w}^h \in \mathcal{V}_A^h(\boldsymbol{\mu})$$

for all  $\boldsymbol{\mu} \in \mathcal{D}$  where  $A = LE, KL$  for linear elasticity and Kirchhoff-Love shells respectively.

## 4 Design Space Collocation Techniques for Parametrized Isogeometric Analysis

In this section, we present a number of techniques of constructing a surrogate model for the solution manifold in parametrized IGA. To set the stage, recall that the displacement field in parametrized structural mechanics takes the form:

$$\vec{u}(\mathbf{x}, \boldsymbol{\mu}) = \sum_{\mathbf{i}} \vec{d}_{\mathbf{i}}(\boldsymbol{\mu}) R_{\mathbf{i}}(\mathbf{x})$$

The basis functions  $R_{\mathbf{i}}(\mathbf{x})$  are known while the control variables  $\vec{d}_{\mathbf{i}}(\boldsymbol{\mu})$  are unknown functions of the design parameters. In our surrogate modeling approach, we approximate the solution vector of (9) using a basis expansion of the form:

$$\mathbf{d}(\boldsymbol{\mu}) = \sum_{\mathbf{i}} \mathbf{c}_{\mathbf{i}} \Psi_{\mathbf{i}}(\boldsymbol{\mu})$$

where  $\{\Psi_{\mathbf{i}}(\boldsymbol{\mu})\}_{\mathbf{i}}$  is a finite-dimensional set of orthogonal basis functions scaled by the coefficients  $\mathbf{c}_{\mathbf{i}}$ .

There are many suitable choices of basis functions but we only consider two herein. In particular, our choice of basis functions lie in the realms of *nodal* and *modal* manifold representations. A nodal paradigm considers the familiar *Lagrange basis*, which interpolates nodal data at sampling points while intermediate data is obtained through a linear combination of such basis functions scaled by their corresponding nodal values [2]. On the other hand, a modal representation considers a truncated spectral expansion of a function. An orthogonal polynomial series is first specified in which to represent the function. The modal coefficients are then associated with the terms in this series. Each coefficient quantifies the modal behavior intrinsic to the aggregate solution field. For both the nodal and modal approaches, we employ parametric sampling to obtain approximations of the coefficients  $\mathbf{c}_{\mathbf{i}}$ . In this manner, our surrogate modeling approach is non-intrusive in that our modeling methodology may be used as a “black box” in conjunction with existing IGA packages. This dramatically simplifies implementation as compared with intrusive approaches such as the parametric Galerkin method.

Before proceeding with the derivation of such polynomial surrogate models, we formally introduce the concept of a *multi-index*, since it greatly simplifies the notation in following sections and will establish a uniform language to be used consistently throughout this paper. A multi-index is an array of indices used to reference the dimensions present in a variable. Consider the multi-index  $\mathbf{i} = (i_1, i_2, \dots, i_d)$ . We can then enumerate a variable in several dimensions as  $\boldsymbol{\mu}_{\mathbf{i}} = (\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_d})$ . It is often times convenient to introduce the concept of a multi-index norm. In this paper, the two norms of interest will be the *sup-norm* and *1-norm* given by:

$$\|\mathbf{i}\|_{\infty} = \max_k |i_k| \quad \text{and} \quad \|\mathbf{i}\|_1 = \sum_{k=1}^d |i_k|$$

respectively. Moreover, after specification of a multi-index norm and a given non-negative integer  $k$ , we can define:

$$\mathcal{P}_k^d := \{ \mathbf{i} \in (\mathbb{Z}^*)^d \mid 0 \leq \|\mathbf{i}\|_1 \leq k \} \quad \text{and} \quad \mathcal{Q}_k^d := \{ \mathbf{i} \in (\mathbb{Z}^*)^d \mid 0 \leq \|\mathbf{i}\|_{\infty} \leq k \}$$

as the spaces of *isotropic* and *tensor-product* multi-indices, respectively. The meaning of this terminology will be made clear in a moment when we define polynomial spaces associated with these spaces of multi-indices. Note that these multi-index spaces are nested, specifically  $\mathcal{P}_k^d \subseteq \mathcal{Q}_k^d$ , and we can simply write summations and products over multivariate quantities using multi-index notation. For example, the tensor-product summation and product operations can be expressed compactly as:

$$\sum_{\mathbf{i} \in \mathcal{P}_k^d} f(\boldsymbol{\mu}_{\mathbf{i}}) = \sum_{i_1=1}^{\ell} \sum_{i_2=1}^{\ell} \dots \sum_{i_d=1}^{\ell} f(\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_d}) \quad \text{and} \quad \prod_{\mathbf{i} \in \mathcal{Q}_k^d} f(\boldsymbol{\mu}_{\mathbf{i}}) = \prod_{i_1=1}^{\ell} \prod_{i_2=1}^{\ell} \dots \prod_{i_d=1}^{\ell} f(\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_d})$$

With the above multi-index notation established, we can now define what we mean by isotropic polynomial and tensor-product polynomial. Specifically, we define the respective spaces of isotropic and tensor-product

polynomials of degree  $p$  for the domain  $\mathcal{D} \subset \mathbb{R}^d$  via:

$$\mathcal{P}_p(\mathcal{D}) := \left\{ f \in L^2(\mathcal{D}) \mid f(\boldsymbol{\mu}) = \sum_{\mathbf{i} \in \mathcal{D}_p^d} f_{\mathbf{i}} \mu_1^{i_1} \mu_2^{i_2} \dots \mu_d^{i_d} \right\} \text{ and } \mathcal{Q}_p(\mathcal{D}) := \left\{ f \in L^2(\mathcal{D}) \mid f(\boldsymbol{\mu}) = \sum_{\mathbf{i} \in \mathcal{Q}_p^d} f_{\mathbf{i}} \mu_1^{i_1} \mu_2^{i_2} \dots \mu_d^{i_d} \right\}$$

At this juncture, it is not quite clear what we mean by ‘‘isotropic.’’ This is made clear by recognizing that if  $f \in \mathcal{P}_p(\mathcal{D})$  and  $\nu : \mathcal{D} \rightarrow \mathbb{R}^d$  is an affine mapping, then the composition mapping  $g = f \circ \nu$  is also a member of  $\mathcal{P}_p(\mathcal{D})$ <sup>1</sup>. Consequently, the space of isotropic polynomials is invariant under an affine change of coordinates. By contrast, this is not the case for the space of tensor-product polynomials. The concepts and formalism associated with constructing nodal and modal representations of the solution manifold from the tools discussed in heretofore are presented in the following sections.

## 4.1 Nodal Solution Manifold Representation

With a nodal manifold representation, we approximate the control variables as:

$$\mathbf{d}(\boldsymbol{\mu}) \approx \sum_{\mathbf{i}} \mathbf{d}(\boldsymbol{\mu}_{\mathbf{i}}) \mathcal{N}_{\mathbf{i}}(\boldsymbol{\mu}) \quad (15)$$

where  $\{\boldsymbol{\mu}_{\mathbf{i}}\}_{\mathbf{i}}$  are a set of interpolation nodes and  $\{\mathcal{N}_{\mathbf{i}}\}_{\mathbf{i}}$  are a set of interpolating polynomials satisfying:

$$\mathcal{N}_{\mathbf{i}}(\boldsymbol{\mu}_{\mathbf{j}}) = \begin{cases} 1, & \text{if } \mathbf{i} \equiv \mathbf{j} \\ 0, & \text{otherwise} \end{cases}$$

where equivalence of multi-indices is understood in the component-wise sense, i.e.,  $i_k = j_k$   $k = 1, 2, \dots, d$ . In the univariate setting, we build the interpolating polynomials from a set of  $p + 1$  *unique* interpolation nodes  $\{\mu_i\}_{i=1}^{p+1}$  which construct the Lagrange interpolants:

$$\mathcal{N}_i(\mu) = \prod_{\substack{j=1 \\ j \neq i}}^{p+1} \frac{\mu - \mu_j}{\mu_i - \mu_j}$$

For the multivariate setting, we must take a bit more care. First of all, we do not allow the interpolation nodes to be defined in an arbitrary manner. Instead, we enforce them to be defined via a tensor-product of one-dimensional sets of unique interpolation nodes. Then each interpolation node takes the form:

$$\boldsymbol{\mu}_{\mathbf{i}} = \left( \mu_{i_1}^{(1)}, \mu_{i_2}^{(2)}, \dots, \mu_{i_d}^{(d)} \right) \quad \forall \mathbf{i} \in \mathcal{D}_p^d$$

for a desired tensor-product polynomial approximation of degree  $p$ . We build the corresponding interpolating polynomial as:

$$\mathcal{N}_{\mathbf{i}}(\boldsymbol{\mu}) = \bigotimes_{k=1}^d \mathcal{N}_{i_k}^{(k)}(\mu_k) = \bigotimes_{k=1}^d \prod_{\substack{j_k=1 \\ j_k \neq i_k}}^{p_k+1} \frac{\mu_k^{(k)} - \mu_{j_k}^{(k)}}{\mu_{i_k}^{(k)} - \mu_{j_k}^{(k)}}$$

Note that the solution coefficients  $\mathbf{d}(\boldsymbol{\mu}_{\mathbf{i}})$  are the collocated values of the control variables at the interpolation nodes. Thus, they are obtained by running an IGA simulation for a select number of design instances.

The choice of interpolation nodes to use with such an interpolating scheme remains to be discussed. It is worth noting that an interpolatory representation of the solution manifold, such as the one described in this section, exhibits behavior dependent on the choice of sampling scheme. For example, a uniformly-distributed collection of collocation points results in a solution manifold suffering from the Runge phenomenon, providing acceptable function approximations in the interior of the design space while exhibiting poor approximability near the boundaries. On the other hand, a Chebyshev distribution of sample points, where the node density is

<sup>1</sup>Here, we have used a slight abuse of notation in extending the domain of  $f$  to all of  $\mathbb{R}^d$ .

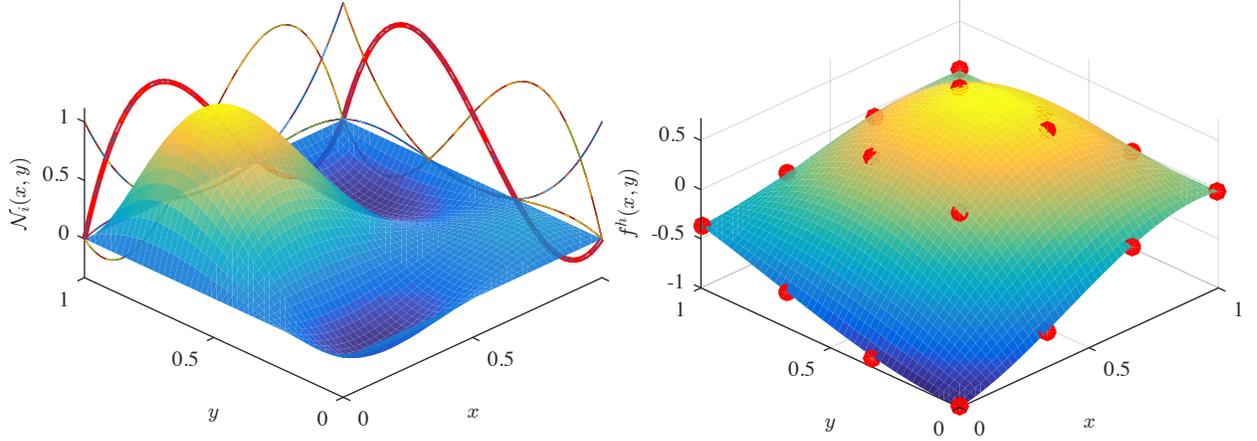


Figure 5: A representative tensor-product Lagrange interpolating basis function (left) and an interpolatory representation of the function  $f(x, y) = \sin(\pi x) \sin(\pi y) - e^{-x^2 - 4y^2}$  using equispaced nodes and tensor-product Lagrange polynomials (right).

larger near the boundaries of the domain, distributes the error more uniformly throughout the surrogate solution manifold.

For the purposes of design space exploration, we consider the tensor-product Gauss-Legendre quadrature scheme for nodal collocation points. This selection is made due to the high-fidelity of the resulting surrogate model while minimizing the required number of points. In particular, the nodal expansion which interpolates the Gauss abscissa is equivalent to a spectral expansion in terms of a Legendre orthogonal polynomial series, maintaining the same fidelity therein [13].

Note that in using the above procedure, we have built an interpolating basis for the tensor-product approximation space  $\mathcal{Q}_p(\mathcal{D})$ . We can build an interpolating basis for the alternative multivariate approximation space  $\mathcal{P}_p(\mathcal{D})$  in an analogous manner, but a stable choice of interpolation nodes for  $\mathcal{P}_p(\mathcal{D})$  is not known at this time. One may also resort to other forms of multi-dimensional interpolation such as radial basis functions, but such an approach is beyond the scope of this paper.

## 4.2 Modal Solution Manifold Representation

With a modal manifold representation, we first write the control variables in terms of the infinite sum:

$$\mathbf{d}(\boldsymbol{\mu}) = \sum_{\|\mathbf{i}\|_\infty < \infty} \mathbf{a}_i \mathcal{M}_i(\boldsymbol{\mu})$$

where  $\{\mathbf{a}_i\}_i$  are a set of spectral coefficients defined as:

$$\mathbf{a}_i = \int_{\mathcal{D}} \mathbf{W}(\boldsymbol{\mu}) \mathbf{d}_i(\boldsymbol{\mu}) \mathcal{M}_i(\boldsymbol{\mu}) d\mathcal{D}$$

and  $\{\mathcal{M}_i\}_i$  are a set of orthonormal polynomial basis functions satisfying:

$$\langle \mathcal{M}_i, \mathcal{M}_j \rangle_w = \int_{\mathcal{D}} \mathbf{W}(\boldsymbol{\mu}) \mathcal{M}_i(\boldsymbol{\mu}) \mathcal{M}_j(\boldsymbol{\mu}) d\mathcal{D} = \begin{cases} 1, & \text{if } \mathbf{i} \equiv \mathbf{j} \\ 0, & \text{otherwise} \end{cases}$$

where  $\langle \cdot, \cdot \rangle_w$  is the weighted  $L^2$  inner product and  $\mathbf{W}(\boldsymbol{\mu}) \geq 0$  is a pre-defined weighting function. A spectral representation of the solution manifold provides a characterization of the various sensitivities with respect to the design parameters. Namely, the average displacement throughout the design space is given by  $\mathbf{a}_0$  while the magnitude of the following spectral coefficients,  $\mathbf{a}_i$ , quantify the prominence of particular higher-order modes

in the aggregate solution manifold. In this paper, we consider the case when  $W(\boldsymbol{\mu}) \equiv 1$ , in which case the corresponding orthonormal basis functions coincide with tensor-product Legendre polynomials. For instance, when  $\mathcal{D} = (-1, 1)^d$ , we have that:

$$\mathcal{M}_{\mathbf{i}}(\boldsymbol{\mu}) = \bigotimes_{k=1}^d \mathcal{L}_{i_k}(\mu_k)$$

where  $\{\mathcal{L}_n\}_{n=0}^{\infty}$  are the univariate Legendre polynomials which are discussed further in Appendix B.

In practice, our infinite sum representation must obviously be truncated after some finite number of terms, resulting in a computationally tractable surrogate model. We consider two such approximations herein, an isotropic polynomial approximation and a tensor-product polynomial approximation. An isotropic polynomial approximation takes the form:

$$\mathbf{d}(\boldsymbol{\mu}) = \sum_{\mathbf{i} \in \mathcal{P}_p^{d,\boldsymbol{\mu}}} \mathbf{a}_{\mathbf{i}} \mathcal{M}_{\mathbf{i}}(\boldsymbol{\mu})$$

while a tensor-product polynomial approximation takes the form:

$$\mathbf{d}(\boldsymbol{\mu}) = \sum_{\mathbf{i} \in \mathcal{Q}_p^{d,\boldsymbol{\mu}}} \mathbf{a}_{\mathbf{i}} \mathcal{M}_{\mathbf{i}}(\boldsymbol{\mu})$$

Since  $\mathcal{P}_p(\mathcal{D}) \subseteq \mathcal{Q}_p(\mathcal{D})$ , the isotropic approximation includes less terms than the tensor-product approximation, but both isotropic and tensor-product approximations are *spectral* approximations exhibiting exponential convergence rates. Hence, the isotropic approximation is often preferred due to its comparatively lower computational expense.

From Figure 6, it is clear that such a modal representation cannot be interpreted analogously to an interpolation. In the latter case, exact function values are realized at collocation points whereas the former is a least-squares approximation to the function.

Note that in order for us to recover both the isotropic and tensor-product approximations, we must compute the spectral coefficients:

$$\mathbf{a}_{\mathbf{i}} = \langle \mathbf{d}, \mathcal{M}_{\mathbf{i}} \rangle_w = \int_{\mathcal{D}} W(\boldsymbol{\mu}) \mathbf{d}(\boldsymbol{\mu}) \mathcal{M}_{\mathbf{i}}(\boldsymbol{\mu}) d\mathcal{D}$$

This presents a challenge for two reasons: (i) We often do not know how to perform the integration analytically and (ii) we do not know the exact form of  $\mathbf{d}(\boldsymbol{\mu})$  as it is the unknown we are trying to solve for. To overcome these issues, we turn to numerical quadrature. Briefly speaking, we approximate:

$$\mathbf{a}_{\mathbf{i}} = \int_{\mathcal{D}} W(\boldsymbol{\mu}) \mathbf{d}(\boldsymbol{\mu}) \mathcal{M}_{\mathbf{i}}(\boldsymbol{\mu}) d\mathcal{D} \approx \sum_{\mathbf{q}} \mathbf{d}(\boldsymbol{\mu}_{\mathbf{q}}) \mathcal{M}_{\mathbf{i}}(\boldsymbol{\mu}_{\mathbf{q}}) w_{\mathbf{q}}$$

where  $\{\boldsymbol{\mu}_{\mathbf{q}}\}_{\mathbf{q}}$  is a set of quadrature points and  $\{w_{\mathbf{q}}\}_{\mathbf{q}}$  is a set of corresponding quadrature weights. We refer to the resulting design space approximation as a *pseudospectral* approximation since the exact spectral coefficients have been approximated. With quadrature, we avoid the need for analytical integration and we also only need to determine  $\mathbf{d}(\boldsymbol{\mu})$ , that is, sample the solution space, at a set of quadrature points. In the next section, we provide an overview of quadrature schemes one may utilize including tensor-product quadrature and sparse quadrature.

### 4.3 Design Space Sampling and Numerical Integration

It remains to select sampling schemes and quadrature rules to recover the interpolated values in the nodal manifold representation and the spectral coefficients in the modal manifold representation. Our objective is to obtain sampling and quadrature schemes which not only lead to accurate surrogate models but are also computationally efficient. Since our selection of interpolation nodes in the nodal manifold representation follows a quadrature scheme, all of our schemes begin with a univariate quadrature rule which is then extended to the multivariate setting.

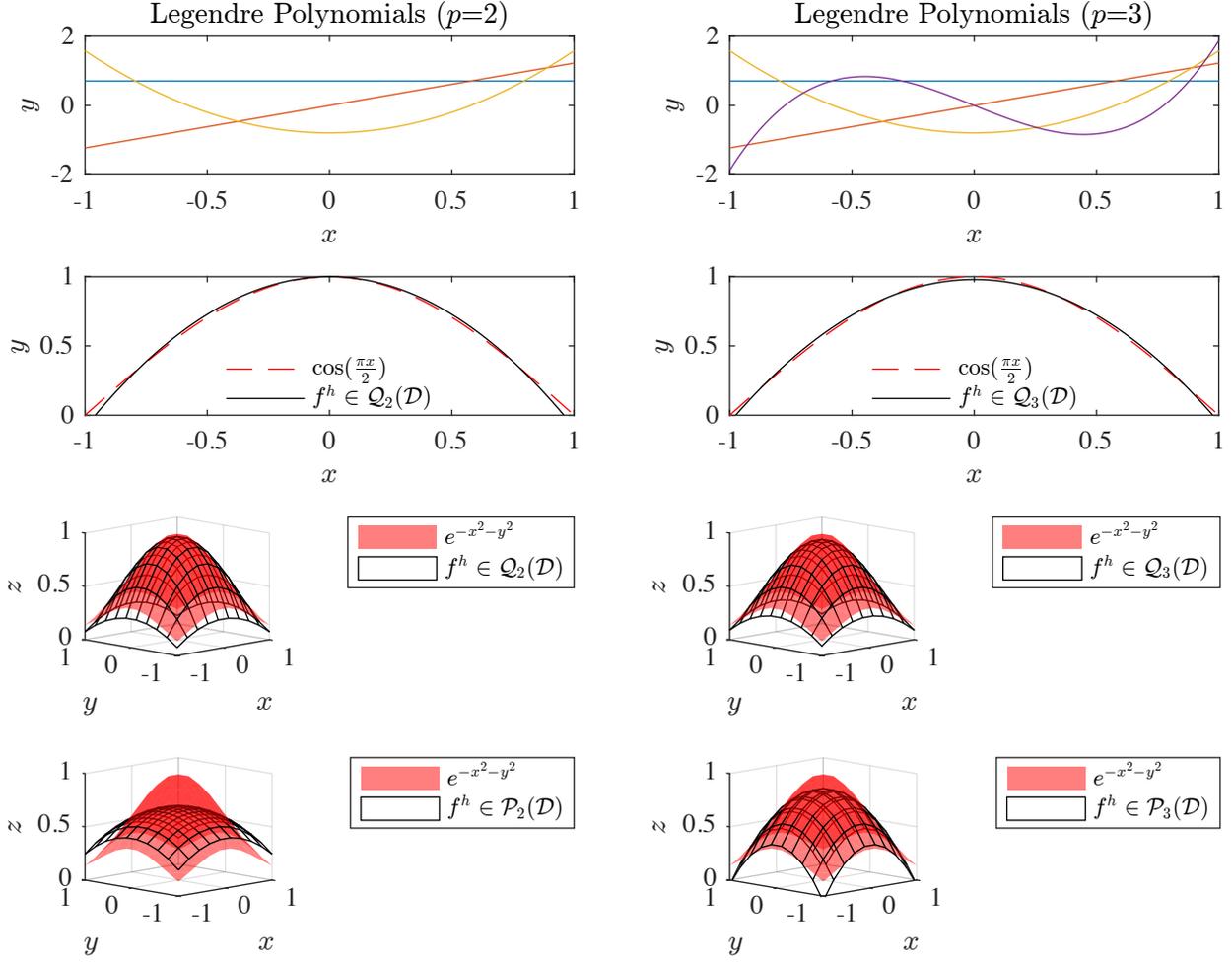


Figure 6: The 1<sup>st</sup> row of the figure shows the Legendre polynomial basis sets for  $p = 2$  and  $p = 3$ . The 2<sup>nd</sup> row shows the 1-dimensional pseudospectral approximation of  $f(x) = \cos(\pi x/2)$  using the above sets of Legendre polynomials, requiring 5 points and 9 points, respectively, for accurate integration using the univariate Clenshaw-Curtis quadrature rule over the domain  $\mathcal{D} = (-1, 1)$ . The 3<sup>rd</sup> row shows the pseudospectral representation of the function  $f(x, y) = e^{-x^2-y^2}$  using a tensor-product Clenshaw-Curtis quadrature rule in a tensor-product Legendre basis, requiring 25 points and 81 points respectively, resulting in 2<sup>nd</sup> and 3<sup>rd</sup> degree tensor-product polynomial approximations over the domain  $\mathcal{D} = (-1, 1)^2$ . The 4<sup>th</sup> row shows the pseudospectral representation of the function  $f(x, y)$  using a Delayed Smolyak Clenshaw-Curtis quadrature rule in the isotropic Legendre basis, requiring 13 points and 29 points respectively, resulting in 2<sup>nd</sup> and 3<sup>rd</sup> degree isotropic polynomial approximations over the domain  $\mathcal{D} = (-1, 1)^2$ .

### 4.3.1 Univariate Quadrature

A general univariate quadrature scheme takes the form:

$$\mathcal{I}_w^1 f = \int_a^b W(x) f(x) dx \approx U_{n_q}^{(1)} f = \sum_{q=1}^{n_q} w_q f(x_q)$$

where  $(a, b)$  is the integration domain,  $f : (a, b) \rightarrow \mathbb{R}$  is the function to be integrated,  $W(x) \geq 0$  is a specified weighting function,  $n_q$  is the number of quadrature points,  $\{x_q\}_{q=1}^{n_q}$  are the quadrature points, and  $\{w_q\}_{q=1}^{n_q}$  are the quadrature weights. We have used the notation  $\mathcal{U}_{n_q}$  to denote a *univariate* scheme with  $n_q$  points.

Table 1: A comparison between the Gauss-Legendre, Clenshaw-Curtis, and Kronrod-Patterson quadrature schemes considered in this paper.

Univariate Scheme	Nested?	$n_\ell$	$p_\ell$
Gauss-Legendre	✗	$\ell$	$2n_\ell - 1$
Clenshaw-Curtis	✓	$\begin{cases} 1, & \ell = 1 \\ 2^{\ell-1} + 1, & \ell \geq 2 \end{cases}$	$\begin{cases} n_\ell, & n_\ell \text{ even} \\ n_\ell + 1, & n_\ell \text{ odd} \end{cases}$
Kronrod-Patterson	✓	$2^\ell - 1$	$\frac{3n_\ell - 1}{2}$

There are a great many number of univariate quadrature schemes to choose from, each with their own sets of pros and cons. We consider three herein: Gauss-Legendre quadrature [26], Clenshaw-Curtis quadrature [12], and Kronrod-Patterson quadrature [46]. For brevity, we explicitly define these rules in Appendix B and discuss their properties here. The Gauss-Legendre scheme is ideal in that it employs a minimal number of quadrature points for a desired level of polynomial accuracy. On the other hand, the Clenshaw-Curtis and Kronrod-Patterson schemes exhibit a so-called nestedness property, to be described later, which alleviates the computational expense associated with an increase in the model polynomial fidelity in the multi-dimensional setting. Each of the three aforementioned schemes consist of not only one quadrature rule but rather a family of quadrature rules. We refer to the  $\ell^{\text{th}}$  member of a given family as the  $\ell^{\text{th}}$  level, and it contains  $n_\ell$  quadrature points and weights and exhibits a polynomial degree accuracy of  $p_\ell$ . That is, the quadrature rule  $\mathcal{U}_{n_\ell}$  exactly integrates polynomials of degree  $p_\ell$ . Table 1 characterizes the Gauss-Legendre, Clenshaw-Curtis, and Kronrod-Patterson schemes and their properties.

We are now ready to define what we mean by nested. Namely, a family of quadrature rules is *nested* if  $\cup_{k=0}^{\ell-1} \{\mathbf{x}_q\}_k \subseteq \{\mathbf{x}_q\}_\ell$  where  $\{\mathbf{x}_q\}_\ell$  is the set of quadrature nodes of level  $\ell$ . This property will allow us to build highly efficient quadrature schemes in the multi-dimensional setting using the so-called Smolyak method.

### 4.3.2 Tensor-Product Multivariate Quadrature

We define a tensor-product multivariate quadrature rule as a tensor-product of univariate quadrature rules. In particular, given a set of  $n_\ell$ -point univariate quadrature rules,  $\{\mathcal{U}_{n_\ell}^{(i)}\}_{i=1}^d$ , each associated with the  $\ell^{\text{th}}$  level of a family of quadrature rules, we construct the multivariate rule:

$$\mathcal{F}_\ell^d f = \bigotimes_{i=1}^d \mathcal{U}_{n_\ell}^{(i)} f$$

for  $f : \mathcal{D} \rightarrow \mathbb{R}$  where  $\mathcal{D} \subset \mathbb{R}^d$  is a hyperrectangle. Then, to approximate the integral of a function, we write:

$$\mathcal{I}_W^d f = \int_{\Omega} W_1(x_1)W_2(x_2)\cdots W_d(x_d)f(x_1, x_2, \dots, x_d)d\Omega \approx \mathcal{F}_\ell^d f = \sum_{i_1=1}^{n_\ell} \sum_{i_2=1}^{n_\ell} \cdots \sum_{i_d=1}^{n_\ell} w_{i_1} w_{i_2} \cdots w_{i_d} f(x_{i_1}, x_{i_2}, \dots, x_{i_d})$$

Figure 7 shows the tensor-product collocation schemes for Clenshaw-Curtis and Gauss-Legendre quadrature rules for two-dimensional integration.

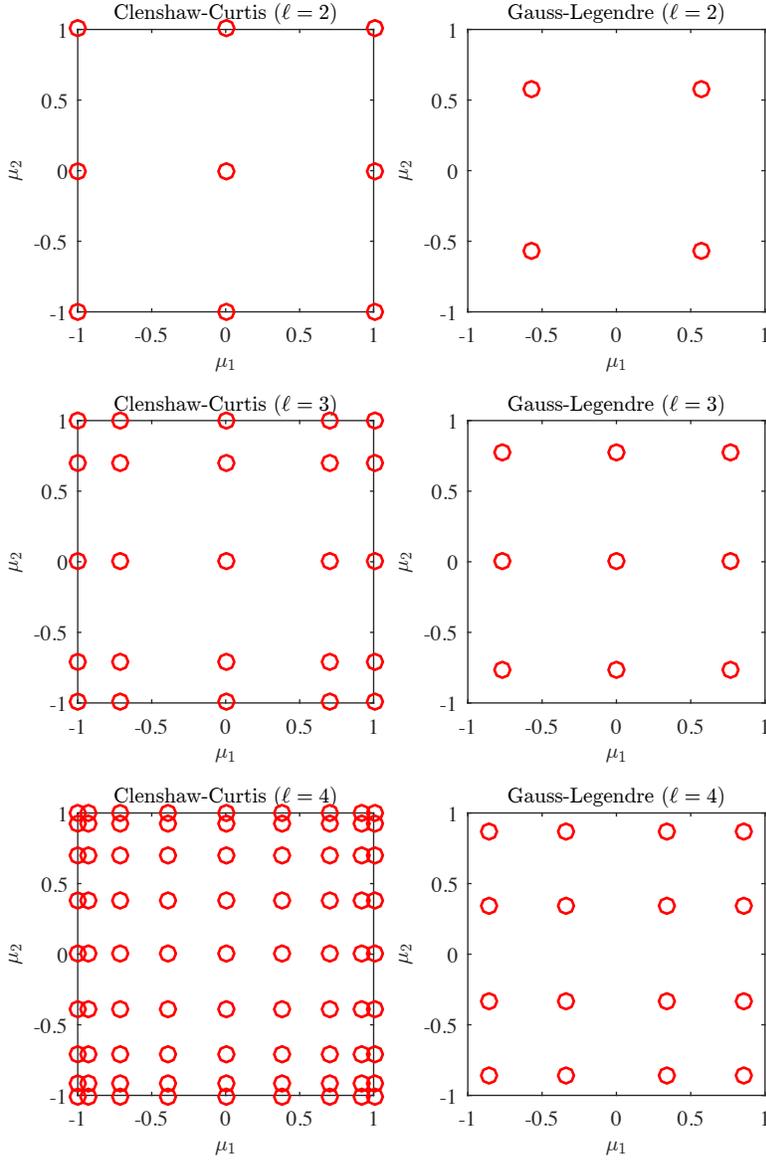


Figure 7: Two-dimensional tensor-product Clenshaw-Curtis (left) and Gauss-Legendre (right) quadrature rules for the integration domain  $\mathcal{D} = (-1, 1)^2$ . The  $\ell^{\text{th}}$ -level Clenshaw-Curtis quadrature rule is capable of integrating  $f \in \mathcal{Q}_{p_\ell}(\mathcal{D})$  with  $p_\ell = 2^{\ell-1} + 1$  for  $\ell > 1$  exactly while each Gauss-Legendre grid level  $\ell$  is capable of integrating  $f \in \mathcal{Q}_{p_\ell}(\mathcal{D})$  with  $p_\ell = 2\ell - 1$  exactly.

Using tensor-product quadrature, we are able to obtain suitable interpolation points for the nodal manifold representation as well as suitable approximations of the spectral coefficients for both isotropic and tensor-product modal manifold approximations. In the nodal setting, we determine the polynomial order of the set of Lagrange basis functions by selecting the appropriate number of collocation points, i.e.,  $p = n - 1 \geq 0$  where  $n$  is the number of univariate sample points. The selection of these points, and hence the polynomial order, are dictated by the error bound given by the Taylor series truncation error [2]. On the other hand, in the modal setting we seek to accurately represent the integrals defining the spectral coefficients using numerical integration:

$$\mathbf{a}_i = \int_{\mathcal{D}} \mathbf{W}(\boldsymbol{\mu}) \mathbf{d}(\boldsymbol{\mu}) \mathcal{M}_i(\boldsymbol{\mu}) d\mathcal{D} \approx \mathcal{F}_\ell^d \mathbf{d} = \sum_{i \in \mathcal{Q}_\ell^d} \mathbf{d}(\boldsymbol{\mu}_i) \mathcal{M}_i(\boldsymbol{\mu}_i) w_i$$

In this case, we select  $\ell$  such that  $p_\ell = 2p$  where  $p$  is the desired polynomial degree of the modal manifold representation, i.e.,  $p = \max_i \deg(\mathcal{M}_i)$ . This has been shown to yield a stable and accurate numerical integration scheme [23, Chapter 4], and when  $\mathbf{d}(\boldsymbol{\mu})$  is a tensor-product polynomial of degree  $p$ , the numerical integration scheme is exact.

Unfortunately it should be noted that a tensor-product quadrature rule suffers from the so-called *curse of dimensionality*. Particularly, each additional dimension, i.e. design parameter, causes an *exponential* increase in the number of required quadrature points. This sampling approach scales as  $N_\ell = n_\ell^d$ , where  $N_\ell$  is the total number of collocation points for level  $\ell$ ,  $n_\ell$  is the number of points associated with a univariate quadrature rule of level  $\ell$ , and  $d$  is the dimension of the design space. Therefore, as the complexity of the design space grows, this approach quickly becomes intractable.

### 4.3.3 Sparse Multivariate Quadrature

To alleviate the curse of dimensionality associated with tensor-product quadrature, we consider sparse quadrature schemes obtained through the use of Smolyak sparse grids. Our objective is the construction of a numerical integration scheme capable of integrating a multi-dimensional *isotropic* polynomial with a minimal number of quadrature points. Recall that here isotropic refers to a polynomial with total degree  $\|\mathbf{p}\|_1 \leq p$  for some  $p \geq 0$ . To begin, we define difference operators for each parametric dimension from a family of quadrature rules via:

$$\Delta_j^{(i)} = \begin{cases} \mathcal{U}_{n_1}^{(i)}, & \text{if } j = 1 \\ \mathcal{U}_{n_{j+1}}^{(i)} - \mathcal{U}_{n_j}^{(i)}, & \text{otherwise} \end{cases}$$

Then, the corresponding level  $\ell$  Smolyak quadrature scheme is defined for  $f : \mathcal{P} \rightarrow \mathbb{R}$  as:

$$\mathcal{S}_\ell^d f = \sum_{\boldsymbol{\alpha} \in \mathcal{P}_\ell^d} \bigotimes_{i=1}^d \Delta_{\alpha_i+1}^{(i)} f$$

The above quadrature rule is exact for at least all isotropic polynomials of degree less than or equal to  $p_\ell$ , though it is not quite exact for tensor-product polynomials of degree  $p_\ell$ . Consequently, Smolyak quadrature should be employed for isotropic polynomial approximations. However, as we shall see, Smolyak quadrature schemes employ far fewer quadrature points than tensor-product schemes.

It is often times convenient, especially for implementation, to represent the Smolyak scheme in terms of the univariate quadrature operators  $\mathcal{U}_{n_\ell}^{(i)}$ . After algebraic manipulation, we arrive at the representation:

$$\mathcal{S}_\ell^d f = \sum_{\ell \leq \|\mathbf{k}\|_1 \leq d+\ell-1} (-1)^{d+\ell-\|\mathbf{k}\|_1-1} \binom{d-1}{\|\mathbf{k}\|_1-\ell} \bigotimes_{i=1}^d \mathcal{U}_{n_{k_i}}^{(i)} f$$

From the above expression, we see that a Smolyak scheme is simply comprised of many tensor-product univariate quadrature schemes. If these schemes happen to exhibit nestedness, then there is an interlacing of quadrature points in between levels which provides additional accuracy with a minimal increase in required function evaluations. Consequently, we only consider nested quadrature families when building sparse quadrature rules for design space exploration.

As seen in Table 1, the number of quadrature points  $n_\ell$  and corresponding degree  $p_\ell$  with the univariate Clenshaw-Curtis and Kronrod-Patterson schemes increases exponentially fast with the level  $\ell$ . Unfortunately, this means this is also the case for Smolyak quadrature rules built from these families. This exponential growth per level is not ideal for application to design space exploration since we ultimately desire an economical, low-fidelity surrogate model. Fortunately, as presented in [47, 10], these exponentially-growing sequences can be delayed to slow the growth in required sampling data. In these delayed sequences, some of the quadrature rules are repeated to lower the rank of approximation and hence the required number of univariate quadrature points. With this in mind, the *delayed Clenshaw-Curtis rule* is defined as:

$$\mathcal{U}_\ell^{CC, \text{delayed}} = \mathcal{U}_{k_\ell}^{CC}, \quad k_\ell = \begin{cases} 1, & \ell = 1 \\ \lceil \log_2(\ell - 1) \rceil + 1, & \ell > 1 \end{cases}$$

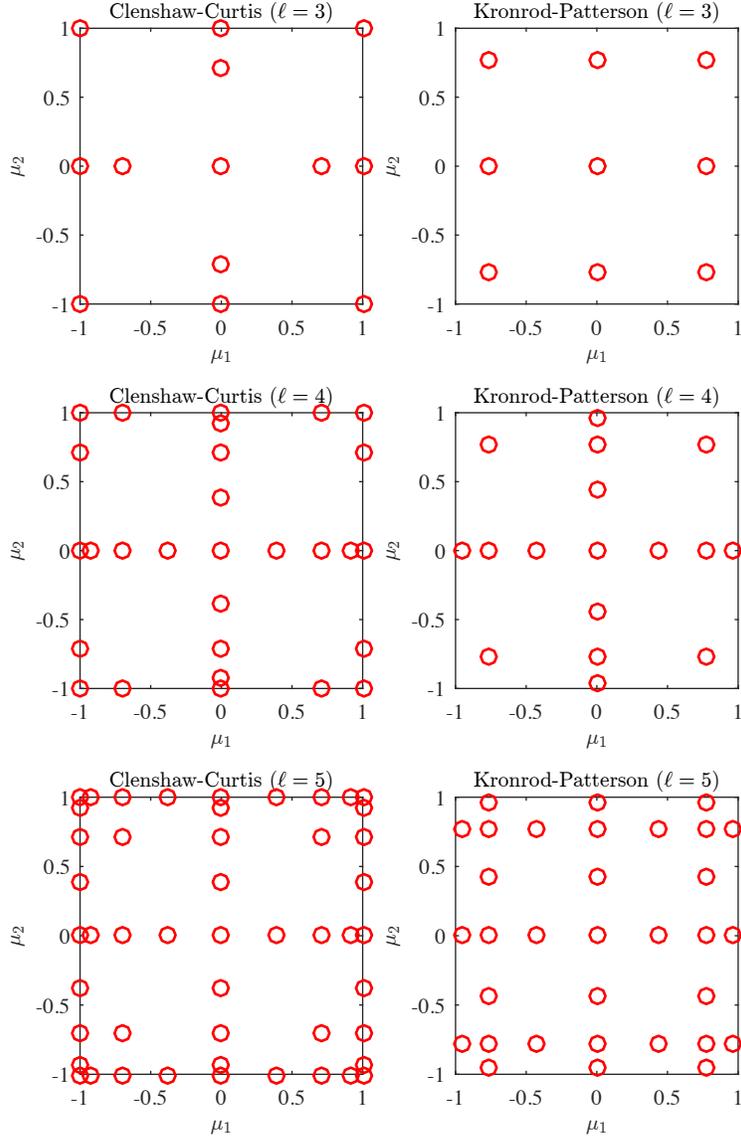


Figure 8: Smolyak collocation schemes for the delayed, two-dimensional Clenshaw-Curtis (left) and Kronrod-Patterson (right) quadrature rules for the integration domain  $\mathcal{D} = (-1, 1)^2$ . Both delayed Clenshaw-Curtis and Kronrod-Patterson grids are capable of integrating  $f \in \mathcal{P}_{p_\ell}(\mathcal{D})$  with  $p_\ell = 2\ell - 1$  exactly. Note that the nestedness of the Clenshaw-Curtis naturally permits Smolyak implementation, contrary to the Gauss-Legendre scheme which requires the Kronrod extension and Patterson's recursion.

while the *delayed Kronrod-Patterson* univariate rule is defined as:

$$\mathcal{U}_\ell^{KP, \text{delayed}} = \mathcal{U}_{k_\ell}^{KP}, \quad \log_2 \left( \frac{4\ell}{3} \right) \leq k_\ell \leq \log_2 \left( \frac{8\ell - 1}{3} \right)$$

where, for simplicity of presentation, the subscripts  $\ell$  and  $k_\ell$  refer to the level rather than the number of quadrature points. Because of the presence of the logarithmic factors in the definitions of the delayed univariate rules, the number of quadrature points  $n_\ell$  associated with the univariate sequences increases only linearly with the level  $\ell$ . Consequently, the number of quadrature points  $n_\ell$  associated with Smolyak quadrature schemes built from these sequences increases algebraically with the level  $\ell$ . This is illustrated in Figure 8 which displays

the Smolyak quadrature points for the delayed Clenshaw-Curtis and Kronrod-Patterson quadrature rules for levels  $\ell = 3, 4, 5$ .

It should be mentioned that the delayed Clenshaw-Curtis and Kronrod-Patterson sequences were constructed such that the associated Smolyak quadrature schemes built from these sequences exhibit a polynomial accuracy of  $p_\ell \geq 2\ell - 1$  for  $\ell \geq 1$ , and for relatively low levels,  $p_\ell = 2\ell - 1$ . This guides the selection of sparse grid quadrature rules for the computation of spectral coefficients for isotropic polynomial coefficients. Recall that to accurately compute the coefficients for an isotropic polynomial of degree  $p$ , we should employ a quadrature rule with accuracy  $2p$ . Therefore, we can use a level  $\ell = p + 1$  Smolyak quadrature scheme built from one of the two delayed sequences. With this in mind, and recalling the rules displayed in Figures 7 and 8, we see that these Smolyak quadrature schemes employ fewer quadrature points for a desired polynomial accuracy than their tensor-product counterparts, as contended earlier, and this reduction becomes far more dramatic with an increase in the dimensionality of the design space.

As a final remark, it is worth noting that the tensor-product quadrature scheme can be expressed in terms of the difference operators defining the Smolyak scheme. Namely, we have:

$$\mathcal{F}_\ell^d f = \sum_{\alpha \in \mathcal{Q}_\ell^d} \bigotimes_{i=1}^d \Delta_{\alpha_i+1}^{(i)} f = \bigotimes_{i=1}^d \mathcal{U}_{n_\ell}^{(i)} f$$

Consequently, the primary difference between the tensor-product and Smolyak schemes lies in the specification of the multi-index spaces  $\mathcal{P}_\ell^d$  and  $\mathcal{Q}_\ell^d$ .

## 5 Numerical Results

This section contains a selection of numerical examples illustrating the convergence behavior of the aforementioned methodology. In particular, we consider: (i) the Scordelis-Lo roof, (ii) a flat L-bracket, and (iii) an NREL 5MW wind turbine blade. Each subsection begins with a brief description of the methodologies discussed in previous sections with application to the problem under consideration. A convergence analysis is presented thereafter, where we will consider the effect of the polynomial order on the expansion to quantify the accuracy of our sampling-based surrogate modeling methodology as well as the effect of a change in the number of quadrature points, which assess the accuracy of the modal coefficients computed through the discrete integration. Lastly, we will numerically demonstrate the capability of design optimization facilitated through the design space exploration paradigm by maximizing or minimizing quantities of interest over a pre-specified design space for the Scordelis-Lo roof.

### 5.1 Scordelis-Lo Roof

Earlier in this paper, the Scordelis-Lo roof was introduced in Section 3 as a canonical shell code validation problem. Due to the relatively simple design-variable description of this shell, it is a premier problem choice for demonstrating the concepts presented in this paper. As such, we will study all aspects of our methodology using this problem, including: (i) the effectiveness of both nodal and modal solution manifold representations, (ii) the effectiveness of both isotropic and tensor-product polynomial approximations in the design space, and (iii) the accuracy of both tensor-product and sparse grid quadrature schemes as applied in design space exploration.

#### 5.1.1 Problem Formulation and Methodology

For the following discussion, the design space  $\mathcal{D}_{\text{roof}} \subset \mathbb{R}^4$  is specified as:

$$\mathcal{D}_{\text{roof}} := \{ \boldsymbol{\mu} = (L, R, t, \varphi) \mid 45 \leq L \leq 55, \quad 20 \leq R \leq 30, \quad .2 \leq t \leq .3, \quad 35^\circ \leq \varphi \leq 45^\circ \}$$

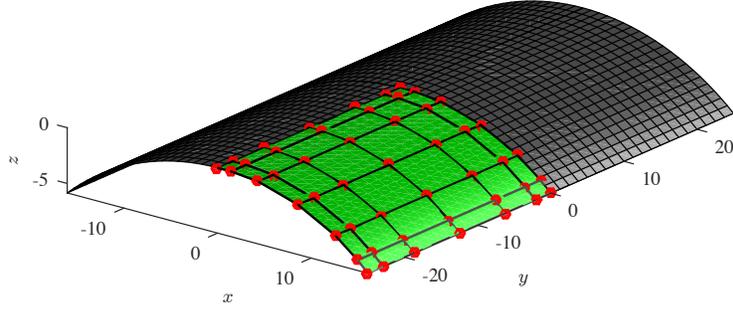


Figure 9: The computational domain of the Scordelis-Lo Roof. For sake of computational expense, only a quarter of the roof is modeled with symmetry boundary conditions. Therefore, refinement with respect to this geometry occurs only in this quadrant. This particular geometry has 16 elements.

Note that this design space is restricted in the sense that  $E = 5.5 \times 10^8$  and  $\nu = 0.05$  are held fixed and moreover are not treated as design parameters, in comparison with our earlier discussion in Section 3.

The computational domain used for analysis is shown in Figure 9. Due to the symmetry of the cylindrical shell geometry, only one quarter of the domain is analyzed. Homogeneous Dirichlet boundary conditions are applied to the cylindrical part of the free boundary while the remaining straight portion of the free boundary is unconstrained. Symmetry boundary conditions are applied by enforcing homogeneous normal derivatives, as well as zero in-plane displacement, across the symmetric boundary. The shell is loaded by a uniform gravitational field  $\vec{g} = (0, 0, -90)^T$ . Two tensor-product meshes of 16 and 64 elements are considered to assess convergence of the sampling-based surrogate modeling strategy under mesh refinement.

Both modal and nodal representations of the solution manifold to this design space are constructed with both tensor-product and sparse sampling. For ease of user interaction and demonstration, the results were compiled into a Graphical User Interface (GUI), which convey the properties of this design approach in comparison to conventional design methodologies. Figures 10 and 11 below show screenshots of such a GUI.

### 5.1.2 Convergence Analysis

We continue our discussion by analyzing convergence behavior. In particular, we consider the error,  $e^h = u_{\text{IGA}}^h - u_{\text{surrogate}}^h$ , in both the  $L^2$  and energy norms:

$$\|e^h\|_{L^2}^2 = \int_{\Omega} |e^h|^2 d\Omega \quad \text{and} \quad \|e^h\|_e^2 = a_{KL}(e^h, e^h) = t \int_{\Omega} \gamma_{\alpha\beta}(e^h) C^{\alpha\beta\lambda\delta} \gamma_{\lambda\delta}(e^h) d\Omega + \frac{t^3}{12} \int_{\Omega} \rho_{\alpha\beta}(e^h) C^{\alpha\beta\lambda\delta} \rho_{\lambda\delta}(e^h) d\Omega$$

effectively assessing the accuracy of the displacement field and the strain energy, respectively. To ensure the accuracy of the figures presented, the  $L^2$  and energy norms are computed for a total of 10 randomly-selected admissible geometries in the design space and then averaged. It should be noted that we do not compare our solution to the exact solution as we are only assessing the effectiveness of our surrogate modeling strategy. Instead, we are comparing the results obtained from our surrogate model to those obtained from IGA.

The first numerical test performed is shown in Table 2 below. In this instance, we use  $n_\ell$  Gauss-Legendre quadrature nodes in each parametric direction to construct a tensor-product Lagrange polynomial of degree  $p_\ell$  which interpolates the solution manifold at the Gauss-Legendre collocation points. From the table we see that the error decreases exponentially fast as the polynomial degree  $p_\ell$  is increased. However, as the nodal manifold representation employs a tensor-product polynomial approximation in the design space, the number of required interpolation points increases very quickly with polynomial degree. Note moreover that the error behavior is the same for both the 16 element mesh and the 64 element mesh.

The second numerical test, shown in Table 3, uses a modal solution manifold representation with a tensor-

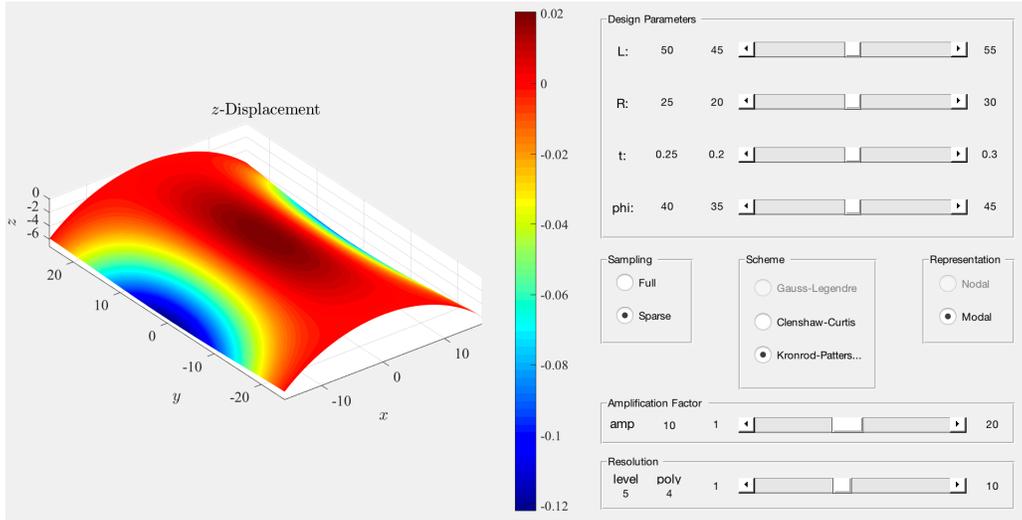


Figure 10: The “nominal” Scordelis-Lo roof geometry. A sparse, level 5 Kronrod-Patterson grid is used to compute the pseudospectral coefficients of the surrogate displacement field in the isotropic Legendre basis.

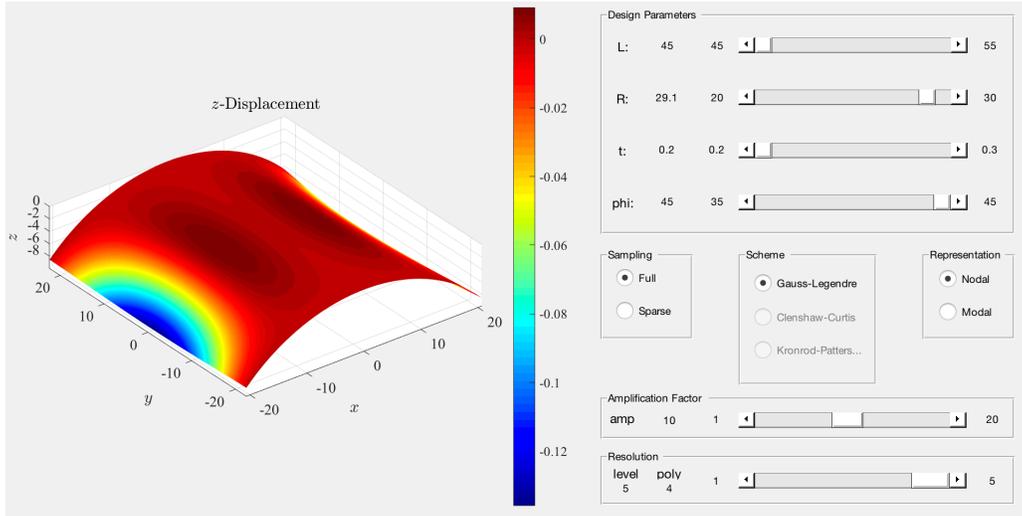


Figure 11: An admissible geometry to the design space specified above. In this particular example, a full (tensor-product), level 5 Gauss-Legendre grid is used to construct a nodal representation of the displacement field using a tensor-product Lagrange basis in the design space.

Table 2: Results for the Scordelis-Lo roof with a Nodal solution manifold representation using a full tensor-product interpolation scheme with the Gauss-Legendre quadrature nodes. In the table below,  $\ell$  is the level parameter,  $p_\ell$  is the polynomial order of the Lagrange basis used to represent the solution manifold,  $n_\ell$  is the number of interpolation points used to attain the specified representation, and  $n_{dof}$  is the number of basis functions in  $\mathcal{Q}_{p_\ell}(\mathcal{D}_{\text{roof}})$ .

Collocation Scheme	Number of Elements	$\ell$	$p_\ell$	$n_{dof}$	$n_\ell$	Average $L^2$ error	Average Energy Error
Gauss-Legendre	16	1	0	1	1	$1.6377 \times 10^{-1}$	$7.1909 \times 10^1$
		2	1	16	16	$3.2600 \times 10^{-2}$	2.3733
		3	2	81	81	$2.2544 \times 10^{-3}$	$7.8076 \times 10^{-1}$
		4	3	256	256	$4.3639 \times 10^{-4}$	$7.3727 \times 10^{-2}$
		5	4	625	625	$6.9755 \times 10^{-5}$	$9.9253 \times 10^{-3}$
	64	1	0	1	1	$1.3030 \times 10^{-1}$	$4.6528 \times 10^1$
		2	1	16	16	$2.2413 \times 10^{-2}$	2.1955
		3	2	81	81	$3.7941 \times 10^{-3}$	$3.4959 \times 10^{-1}$
		4	3	256	256	$5.2635 \times 10^{-4}$	$1.0806 \times 10^{-1}$
		5	4	625	625	$7.1170 \times 10^{-5}$	$1.2703 \times 10^{-2}$

product polynomial approximation in the design space. The pseudospectral coefficients are computed using the Gauss-Legendre quadrature scheme. Recall that in this case, the polynomial order of the basis function set is characterized as a tensor-product of univariate polynomial basis functions of degree  $p_\ell = \ell - 1$ . Again, we see that the error decreases exponentially fast as the polynomial degree  $p_\ell$  is increased, and in fact, the reported errors in Tables 2 and 3 are nearly identical. Consequently, these nodal and modal approaches exhibit very similar behavior for tensor-product polynomial approximations of the solution manifold.

The last numerical test employs the sparse collocation framework through Smolyak grids, presented in Section 4. Both the delayed Kronrod-Patterson and delayed Clenshaw-Curtis Smolyak schemes are used to assess the numerical properties of the sparse modal representation of the solution manifold. The results are presented in Table 4 in which the aforementioned collocation schemes are employed to compute the pseudospectral coefficients for the sparse Legendre basis contained in  $\mathcal{P}_p(\mathcal{D}_{\text{roof}})$ .

In Table 4, we see that both the delayed Kronrod-Patterson and Clenshaw-Curtis schemes exhibit similar convergence rates as a function of level  $\ell$ . However, the required number of quadrature points to achieve the same accuracy using Clenshaw-Curtis is approximately three times that of Kronrod-Patterson. For both rules, the error behavior is the same for both the 16 element mesh and the 64 element mesh.

It is also important to assess the accuracy of computation of the sparse pseudospectral coefficients. To determine this accuracy, the sparse coefficients of  $\mathcal{P}_p(\mathcal{D}_{\text{roof}})$  are computed for  $0 \leq p \leq 4$  using a tensor-product univariate quadrature rule, in contrast to the sparse quadrature rule used in Table 4. The results of this analysis are depicted in Table 5. From this table, we can see that the tensor-product Gauss-Legendre scheme overall performs better in the computation of the sparse pseudospectral coefficients in comparison to the sparse integration rule. However, the results are only marginally better but with a significant increase in computational cost. This suggests that the sparse rules can and should be employed in practice.

The results from Tables 2, 3, 4, and 5 are summarized in Figures 12 and 13. In Figure 12, the  $L^2$  error is plotted against the required number of samples  $n_\ell$  while Figure 13 depicts the Energy error against required number of samples. Once again, note that the tensor-product Gauss-Legendre quadrature scheme using a Nodal surrogate model produces similar results to its modal equivalent. Additionally, we see that the isotropic surrogate model with sparse quadrature performs similarly to the full sampling. With increasing design space dimensionality, we predict the attractiveness of sparse Kronrod-Patterson with an isotropic polynomial basis to be much more pronounced. This is due to the curse of dimensionality which will exponentially increase the number of required function evaluations for a tensor-product quadrature rule.

Table 3: Results for the Scordelis-Lo roof with a Modal solution manifold representation using a tensor-product Gauss-Legendre quadrature scheme for numerical computation of the pseudospectral coefficients. In the table below,  $\ell$  is the level parameter,  $p_\ell$  is the polynomial order of the Legendre basis used to represent the solution manifold,  $n_\ell$  is the number of quadrature points used to attain the specified representation, and  $n_{dof}$  is the number of basis functions in  $\mathcal{Q}_{p_\ell}(\mathcal{D}_{\text{roof}})$ .

Quadrature Scheme	Number of Elements	$\ell$	$p_\ell$	$n_{dof}$	$n_\ell$	Average $L^2$ error	Average Energy Error
Gauss-Legendre	16	1	0	1	1	$1.6377 \times 10^{-1}$	$7.1909 \times 10^1$
		2	1	16	16	$3.2600 \times 10^{-2}$	2.3733
		3	2	81	81	$2.2544 \times 10^{-3}$	$7.8076 \times 10^{-1}$
		4	3	256	256	$4.3639 \times 10^{-4}$	$7.3727 \times 10^{-2}$
		5	4	625	625	$6.9755 \times 10^{-5}$	$9.9253 \times 10^{-3}$
	64	1	0	1	1	$1.3030 \times 10^{-1}$	$4.6528 \times 10^1$
		2	1	16	16	$2.2413 \times 10^{-2}$	2.1955
		3	2	81	81	$3.7941 \times 10^{-3}$	$3.4959 \times 10^{-1}$
		4	3	256	256	$5.2635 \times 10^{-4}$	$1.0806 \times 10^{-1}$
		5	4	625	625	$7.1170 \times 10^{-5}$	$1.2703 \times 10^{-2}$

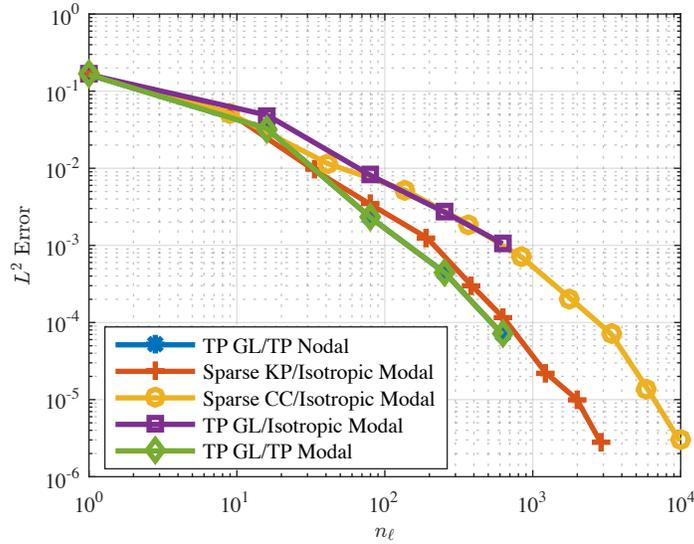


Figure 12: The  $L^2$  error in the displacement field for the various surrogate models considered applied to the Scordelis-Lo roof problem. The first half of each legend entry refers to the sampling scheme employed while the second half refers to the surrogate model used. Note here that TP denotes “Tensor-Product”, GL denotes “Gauss-Legendre”, KP denotes “Kronrod-Patterson”, and CC denotes “Clenshaw-Curtis”. Additionally, “Nodal” indicates the use of the Lagrange polynomials as an interpolating basis while “Modal” denotes the use of the Legendre orthogonal polynomials.

Table 4: Results for the Scordelis-Lo roof with a modal solution manifold representation using sparse, delayed Kronrod-Patterson and Clenshaw-Curtis schemes for the computation of the sparse pseudospectral coefficients in the isotropic Legendre basis. In the table below,  $\ell$  denotes the level parameter,  $n_\ell$  is the number of function evaluations required to accurately capture the pseudospectral coefficients, and  $n_{dof}$  is the number of basis functions in  $\mathcal{P}_{p_\ell}(\mathcal{D}_{\text{roof}})$ .

Quadrature Scheme	Number of Elements	$\ell$	$n_{dof}$	$n_\ell$	Average $L^2$ error	Average Energy Error
Delayed Kronrod-Patterson	16	1	1	1	$1.6377 \times 10^{-1}$	$7.1909 \times 10^1$
		2	5	9	$5.1812 \times 10^{-2}$	$1.7968 \times 10^1$
		3	15	33	$9.5744 \times 10^{-3}$	3.9784
		4	35	81	$3.3485 \times 10^{-3}$	$9.8703 \times 10^{-1}$
		5	70	193	$1.2528 \times 10^{-3}$	$2.1011 \times 10^{-1}$
		6	126	385	$3.0431 \times 10^{-4}$	$6.8352 \times 10^{-2}$
		7	210	641	$1.1234 \times 10^{-4}$	$1.8871 \times 10^{-2}$
		8	330	1217	$2.2538 \times 10^{-5}$	$5.8715 \times 10^{-3}$
		9	495	1985	$1.0071 \times 10^{-5}$	$1.3624 \times 10^{-3}$
	10	715	2881	$2.7734 \times 10^{-6}$	$4.9535 \times 10^{-4}$	
	64	1	1	1	$1.3030 \times 10^{-1}$	$4.6528 \times 10^1$
		2	5	9	$2.1879 \times 10^{-2}$	7.0638
		3	15	33	$9.3368 \times 10^{-3}$	2.5295
		4	35	81	$2.1281 \times 10^{-3}$	$7.0158 \times 10^{-1}$
		5	70	193	$6.4161 \times 10^{-4}$	$1.2545 \times 10^{-1}$
		6	126	385	$1.8277 \times 10^{-4}$	$4.9450 \times 10^{-2}$
		7	210	641	$7.4230 \times 10^{-5}$	$8.9910 \times 10^{-3}$
		8	330	1217	$1.9857 \times 10^{-5}$	$3.5557 \times 10^{-3}$
9		495	1985	$8.4778 \times 10^{-6}$	$6.4857 \times 10^{-4}$	
10	715	2881	$2.3575 \times 10^{-6}$	$2.9674 \times 10^{-4}$		
Delayed Clenshaw-Curtis	16	1	1	1	$1.6377 \times 10^{-1}$	$7.1909 \times 10^1$
		2	5	9	$5.1394 \times 10^{-2}$	$1.8089 \times 10^1$
		3	15	41	$1.1524 \times 10^{-2}$	4.0875
		4	35	137	$5.2109 \times 10^{-3}$	1.1882
		5	70	369	$1.8731 \times 10^{-3}$	$3.8849 \times 10^{-1}$
		6	126	849	$6.9447 \times 10^{-4}$	$1.6440 \times 10^{-1}$
		7	210	1777	$2.0808 \times 10^{-4}$	$3.4681 \times 10^{-2}$
		8	330	3377	$7.1409 \times 10^{-5}$	$1.0398 \times 10^{-2}$
		9	495	5953	$1.3878 \times 10^{-5}$	$2.4518 \times 10^{-3}$
	10	715	9857	$2.9136 \times 10^{-6}$	$5.5005 \times 10^{-4}$	
	64	1	1	1	$1.3030 \times 10^{-1}$	$4.6528 \times 10^1$
		2	5	9	$2.4812 \times 10^{-2}$	7.2903
		3	15	41	$6.8846 \times 10^{-3}$	3.1244
		4	35	137	$3.6722 \times 10^{-3}$	$7.7122 \times 10^{-1}$
		5	70	369	$1.5087 \times 10^{-3}$	$3.3838 \times 10^{-1}$
		6	126	849	$3.1583 \times 10^{-4}$	$1.0724 \times 10^{-1}$
		7	210	1777	$1.0943 \times 10^{-4}$	$4.3432 \times 10^{-2}$
		8	330	3377	$8.2457 \times 10^{-5}$	$7.9000 \times 10^{-3}$
9		495	5953	$1.0503 \times 10^{-5}$	$2.0733 \times 10^{-3}$	
10	715	9857	$2.0608 \times 10^{-6}$	$3.4954 \times 10^{-4}$		

Table 5: Results for the Scordelis-Lo roof with a modal solution manifold representation using an isotropic Legendre basis with a tensor-product univariate Gauss-Legendre quadrature scheme. In the table below,  $\ell$  denotes the level parameter,  $n_\ell$  is the number of function evaluations required to accurately capture the pseudospectral coefficients, and  $n_{dof}$  is the number of basis functions in  $\mathcal{P}_{p_\ell}(\mathcal{D}_{\text{roof}})$ .

Quadrature Scheme	Number of Elements	$\ell$	$n_{dof}$	$n_\ell$	Average $L^2$ error	Average Energy Error
Gauss-Legendre	16	1	1	1	$1.6377 \times 10^{-1}$	$7.1909 \times 10^1$
		2	5	16	$4.8885 \times 10^{-2}$	$1.8471 \times 10^1$
		3	15	81	$8.0005 \times 10^{-3}$	3.9283
		4	35	256	$2.6645 \times 10^{-3}$	$8.6922 \times 10^{-1}$
		5	70	625	$1.0542 \times 10^{-3}$	$1.6657 \times 10^{-1}$
	64	1	1	1	$1.3030 \times 10^{-1}$	$4.6528 \times 10^1$
		2	5	16	$2.4659 \times 10^{-2}$	7.4684
		3	15	81	$7.2669 \times 10^{-3}$	2.2228
		4	35	256	$2.0511 \times 10^{-3}$	$5.7762 \times 10^{-1}$
		5	70	625	$5.2271 \times 10^{-4}$	$1.1514 \times 10^{-1}$

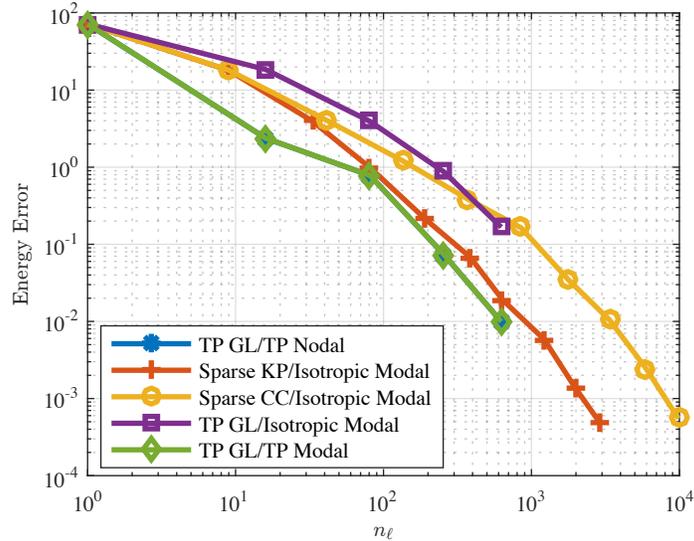


Figure 13: The energy error in the displacement field for the various surrogate models considered applied to the Scordelis-Lo roof problem. The first half of each legend entry refers to the sampling scheme employed while the second half refers to the surrogate model used. Note here that TP denotes “Tensor-Product”, GL denotes “Gauss-Legendre”, KP denotes “Kronrod-Patterson”, and CC denotes “Clenshaw-Curtis”. Additionally, “Nodal” indicates the use of the Lagrange polynomials as an interpolating basis while “Modal” denotes the use of the Legendre orthogonal polynomials.

### 5.1.3 Design Optimization and Comparison with Exact Optima

The design space exploration paradigm presented in this paper allows the user to pose a multi-constraint optimization problem, for which the solution can be approximated after the surrogate model has been constructed. In particular, we can specify an objective functional that is to be minimized through a constrained optimization routine. For example, consider the functional which combines the effects of midpoint displacement and root

Table 6: Optimization of the functional presented in this section using the full IGA model throughout the design space.

Model	$\alpha_1$	$\alpha_2$	Optimal Design Parameters (L, R, t, $\varphi$ )	Function Calls	$\sigma_1$	$d_z$	$\varepsilon_1$
Isogeometric Solution	1	0	(45, 20, 0.3, 45)	215	-841.1226	-0.0501	-0.3414
	$\frac{2}{3}$	$\frac{1}{3}$	(55, 30, 0.2, 38.35)	321	-1524.0956	-0.2687	-1.4085
	$\frac{1}{2}$	$\frac{1}{2}$	(55, 30, 0.2, 35)	243	-1644.8266	-0.3010	-1.4606
	$\frac{1}{3}$	$\frac{2}{3}$	(55, 30, 0.2, 35)	180	-1644.8448	-0.3010	-1.4606
	0	1	(55, 30, 0.2, 35)	150	-1644.9230	-0.3011	-1.4607

strain:

$$\mathcal{F}(\alpha_1, \alpha_2; \boldsymbol{\mu}) = \alpha_1 d_z(\boldsymbol{\mu}) + \alpha_2 \varepsilon_1(\boldsymbol{\mu})$$

Here,  $\varepsilon_1$  is the largest principle strain at the clamped end of the roof,  $d_z$  is the displacement of the midpoint of the free edge, and  $\alpha_1, \alpha_2$  are weights which describe the significance of each quantity in the optimization routine.

Without our design space exploration framework, a functional such as the one above can be minimized, or maximized, through a constrained optimization code. However, each iteration in such a routine requires a complete system construction, assembly, and solve. In the design space exploration paradigm, each iteration is simply a function call to the surrogate model of the solution manifold that has been constructed. Although the output of the optimization routine in this context is an approximation to the true value, it provides a refined search window for additional, high-fidelity design space exploration tools to converge on the true optimum.

For the Scordelis-Lo roof, we employ a constrained optimization routine and compare results of a full system construction and solution to the design space exploration surrogate model. The design space is identical to that used in the previous sections and the initial guess to the optimization routine is a random geometry in this design space. For these simulations, we use the 64 element mesh and consider two surrogate models: (i) a tensor-product polynomial approximation interpolating the solution at Gauss-Legendre quadrature points, and (ii) a pseudospectral isotropic polynomial approximation where spectral coefficients are computed with the delayed Kronrod-Patterson quadrature rule. The results of this optimization routine are shown in Tables 6, 7, and 8.

For convergence criteria, we assert that the quantities of interest are within 1% of their true value. We observe that the interpolatory surrogate model converges to the optimal design location quickly in that it generally converges by  $\ell = 2$ , except for  $(\alpha_1, \alpha_2) = (1, 0)$ , where it converges by  $\ell = 4$ . On the other hand, the isotropic polynomial surrogate model converges to the optimal design location slower, typically by  $\ell = 3$ , except for  $(\alpha_1, \alpha_2) = (2/3, 1/3)$ . Recall however, that the computational expense associated with constructing the isotropic polynomial surrogate model is lower than the tensor-product polynomial approximation. We also consider the convergence of the surrogate model predicted max principal stress to that of the IGA solution. In the case of the interpolating surrogate model, note that the stress generally converges by  $\ell = 3$ , slightly slower than the rate of convergence to the optimal design location. In the case of the isotropic polynomial surrogate model, the max principal stress generally converges to the true value by  $\ell = 5$ .

## 5.2 Flat L-Bracket

We next consider application of our design space exploration framework to the analysis of a flat L-Bracket. The L-bracket is a linear-elastic model with a 17-dimensional design space, demonstrating the versatility of the proposed design space exploration framework. Moreover, full tensor-product sampling with either nodal or modal solution manifold representations is not feasible due to the high-dimensionality of this problem. For example, a linear, tensor-product pseudospectral or interpolatory model of the solution manifold would require

Table 7: Optimization of the functional presented in this section for varying level and  $\alpha_i$  for a tensor-product, interpolating surrogate model.

Model	$\alpha_1$	$\alpha_2$	$p$	Optimal Design Parameters (L, R, t, $\varphi$ )	Function Calls	$\sigma_1$	$d_z$	$\varepsilon_1$
Interpolatory with Gauss-Legendre	1	0	0	(45.69, 24, 0.26, 41.98)	5	-1564.7212	-0.1214	-0.6749
			1	(45, 20, 0.3, 35)	282	-873.3413	-0.0495	-0.3130
			2	(45, 20, 0.3, 35)	236	-902.2156	-0.0509	-0.3212
			3	(45, 20, 0.3, 45)	212	-840.5739	-0.0501	-0.3411
			4	(45, 20, 0.3, 45)	181	-841.1787	-0.0501	-0.3414
	$\frac{2}{3}$	$\frac{1}{3}$	0	(55, 30, 0.3, 45)	186	-947.7444	-0.1214	-0.8791
			1	(55, 30, 0.2, 37.05)	195	-1536.1837	-0.2698	-1.4058
			2	(55, 30, 0.2, 37.79)	176	-1530.2811	-0.2687	-1.4061
			3	(55, 30, 0.2, 38.46)	167	-1518.8445	-0.2674	-1.4051
			4	(55, 30, 0.2, 38.37)	187	-1523.7463	-0.2686	-1.4084
	$\frac{1}{2}$	$\frac{1}{2}$	0	(55, 30, 0.3, 45)	141	-947.7431	-0.1214	-0.8791
			1	(55, 30, 0.2, 35)	157	-1595.2322	-0.2903	-1.4453
			2	(55, 30, 0.2, 35)	132	-1639.3973	-0.2984	-1.4600
			3	(55, 30, 0.2, 35)	160	-1643.2761	-0.3006	-1.4591
			4	(55, 30, 0.2, 35)	178	-1645.0724	-0.3011	-1.4607
	$\frac{1}{3}$	$\frac{2}{3}$	0	(55, 30, 0.3, 45)	133	-947.7425	-0.1214	-0.8791
			1	(55, 30, 0.2, 35)	128	-1595.2309	-0.2903	-1.4453
			2	(55, 30, 0.2, 35)	154	-1639.4217	-0.2984	-1.4601
			3	(55, 30, 0.2, 35)	133	-1643.3374	-0.3006	-1.4591
			4	(55, 30, 0.2, 35)	139	-1645.1882	-0.3011	-1.4608
0	1	0	(55, 30, 0.3, 45)	123	-947.7419	-0.1214	-0.8791	
		1	(55, 30, 0.2, 35)	111	-1595.2319	-0.2903	-1.4453	
		2	(55, 30, 0.2, 35)	142	-1639.4180	-0.2984	-1.4601	
		3	(55, 30, 0.2, 35)	115	-1643.3302	-0.3006	-1.4591	
		4	(55, 30, 0.2, 35)	131	-1645.1909	-0.3011	-1.4608	

Table 8: Functional optimization for varying level and  $\alpha_i$  for a sparse, isotropic surrogate model.

Model	$\alpha_1$	$\alpha_2$	$\ell$	Optimal Design Parameters (L, R, t, $\varphi$ )	Function Calls	$\sigma_1$	$d_z$	$\varepsilon_1$
Pseudospectral with Delayed Kronrod-Patterson	1	0	1	(47.42, 20, 0.24, 43.44)	5	-1620.7782	-0.1214	-0.6987
			2	(45, 20, 0.3, 45)	90	-525.6763	-0.0133	-0.2302
			3	(45, 20, 0.3, 35)	147	-815.8440	-0.0436	-0.2969
			4	(45, 20, 0.3, 45)	136	-813.3066	-0.0479	-0.3363
			5	(45, 20, 0.3, 45)	140	-847.3026	-0.0507	-0.3431
			6	(45, 20, 0.3, 45)	125	-840.7822	-0.0501	-0.3413
			7	(45, 20, 0.3, 45)	136	-841.3068	-0.0501	-0.3414
			8	(45, 20, 0.3, 45)	162	-841.0824	-0.0501	-0.3414
			9	(45, 20, 0.3, 45)	148	-841.1001	-0.0501	-0.3414
			10	(45, 20, 0.3, 45)	133	-841.1252	-0.0501	-0.3414
	$\frac{2}{3}$	$\frac{1}{3}$	1	(55, 30, 0.3, 45)	126	-947.7444	-0.1214	-0.8791
			2	(55, 30, 0.2, 45)	88	-1333.4704	-0.2011	-1.2796
			3	(55, 30, 0.2, 41.56)	113	-1408.0096	-0.2360	-1.3363
			4	(55, 30, 0.2, 39.22)	123	-1493.7329	-0.2607	-1.3921
			5	(55, 30, 0.2, 38.56)	150	-1518.9885	-0.2675	-1.4071
			6	(55, 30, 0.2, 38.37)	130	-1524.1443	-0.2687	-1.4087
			7	(55, 30, 0.2, 38.35)	106	-1524.0718	-0.2687	-1.4085
			8	(55, 30, 0.2, 38.35)	95	-1524.2580	-0.2687	-1.4086
			9	(55, 30, 0.2, 38.35)	131	-1524.1108	-0.2687	-1.4085
			10	(55, 30, 0.2, 38.36)	129	-1524.1015	-0.2687	-1.4085
	$\frac{1}{2}$	$\frac{1}{2}$	1	(55, 30, 0.3, 45)	165	-947.7431	-0.1214	-0.8791
			2	(55, 30, 0.2, 45)	90	-1333.4712	-0.2011	-1.2796
			3	(55, 30, 0.2, 36.39)	108	-1521.7261	-0.2737	-1.3933
			4	(55, 30, 0.2, 35)	150	-1623.7197	-0.2969	-1.4484
			5	(55, 30, 0.2, 35)	127	-1641.9099	-0.3005	-1.4600
			6	(55, 30, 0.2, 35)	133	-1644.2682	-0.3009	-1.4603
			7	(55, 30, 0.2, 35)	143	-1644.8948	-0.3011	-1.4607
			8	(55, 30, 0.2, 35)	101	-1644.9328	-0.3011	-1.4607
			9	(55, 30, 0.2, 35)	127	-1644.9215	-0.3011	-1.4607
			10	(55, 30, 0.2, 35)	110	-1644.9249	-0.3011	-1.4607
	$\frac{1}{3}$	$\frac{2}{3}$	1	(55, 30, 0.3, 45)	143	-947.7425	-0.1214	-0.8791
			2	(55, 30, 0.2, 45)	91	-1333.4715	-0.2011	-1.2796
			3	(55, 30, 0.2, 35)	117	-1541.9198	-0.2823	-1.4008
			4	(55, 30, 0.2, 35)	99	-1623.9744	-0.2970	-1.4485
			5	(55, 30, 0.2, 35)	107	-1641.9102	-0.3005	-1.4600
			6	(55, 30, 0.2, 35)	97	-1644.2650	-0.3009	-1.4603
			7	(55, 30, 0.2, 35)	118	-1644.9110	-0.3011	-1.4607
			8	(55, 30, 0.2, 35)	109	-1644.9338	-0.3011	-1.4607
			9	(55, 30, 0.2, 35)	92	-1644.9182	-0.3011	-1.4607
			10	(55, 30, 0.2, 35)	99	-1644.9260	-0.3011	-1.4607
0	1	1	(55, 30, 0.3, 45)	144	-947.7419	-0.1214	-0.8791	
		2	(55, 30, 0.2, 45)	127	-1333.4718	-0.2011	-1.2796	
		3	(55, 30, 0.2, 35)	95	-1541.9196	-0.2823	-1.4008	
		4	(55, 30, 0.2, 35)	110	-1623.9775	-0.2970	-1.4485	
		5	(55, 30, 0.2, 35)	102	-1641.9127	-0.3005	-1.4600	
		6	(55, 30, 0.2, 35)	98	-1644.2677	-0.3009	-1.4603	
		7	(55, 30, 0.2, 35)	126	-1644.9140	-0.3011	-1.4607	
		8	(55, 30, 0.2, 35)	109	-1644.9364	-0.3011	-1.4607	
		9	(55, 30, 0.2, 35)	120	-1644.9208	-0.3011	-1.4607	
		10	(55, 30, 0.2, 35)	132	-1644.9289	-0.3011	-1.4607	

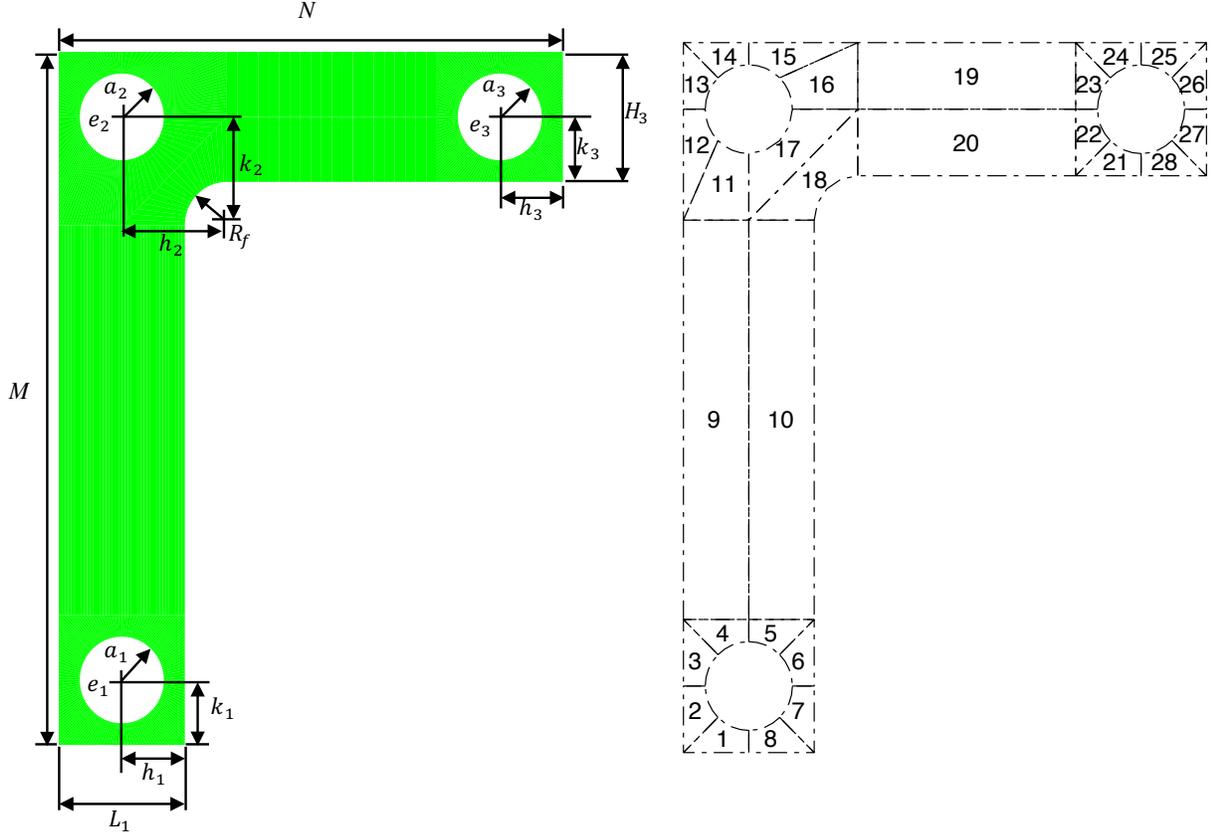


Figure 14: The design parameters associated with the Flat L-Bracket problem (left) and the patches associated with implementing a multi-patch isogeometric linear-elastic solver (right).

$2^{17} = 131,072$  samples. Therefore, we resort to the sparse collocation schemes presented in previous sections, which capture the modes described by the total order of the underlying polynomial basis set. Additionally, the analysis model is constructed from 28  $C^0$ -continuous NURBS patches, demonstrating the applicability of this exploration framework in the multi-patch setting.

### 5.2.1 Problem Formulation and Methodology

Figure 14 below shows the design parameters for the L-Bracket as well as the patch layout used for analysis. Figure 15 below shows the boundary conditions and forcing used in the design space exploration problem. The multi-patch model is discretized and refined to a 480-element mesh. Additionally, the holes described by  $a_1$  and  $a_2$  are constrained to zero displacement, effectively simulating a bolt. The remaining hole has an applied loading of  $P = 30 \times 10^6$ , directed upwards at a  $45^\circ$  angle and cosine-distributed across the boundary.

In reference to Figure 14, the design space of the L-bracket is characterized by the design variable:

$$\mathcal{D}_L := \{\boldsymbol{\mu} = (M, N, L_1, H_3, R_f, a_1, e_1, t_{1h}, t_{1k}, a_2, e_2, t_{2h}, t_{2k}, a_3, e_3, t_{3h}, t_{3k}) \mid \boldsymbol{\mu} \in \mathbb{R}^{17}\} \quad (16)$$

which is comprised of the 17 design parameters associated with the L-bracket, as illustrated in Figure 14. We construct two design spaces of the form (16) for the L-Bracket geometry. The parameters defining the spaces

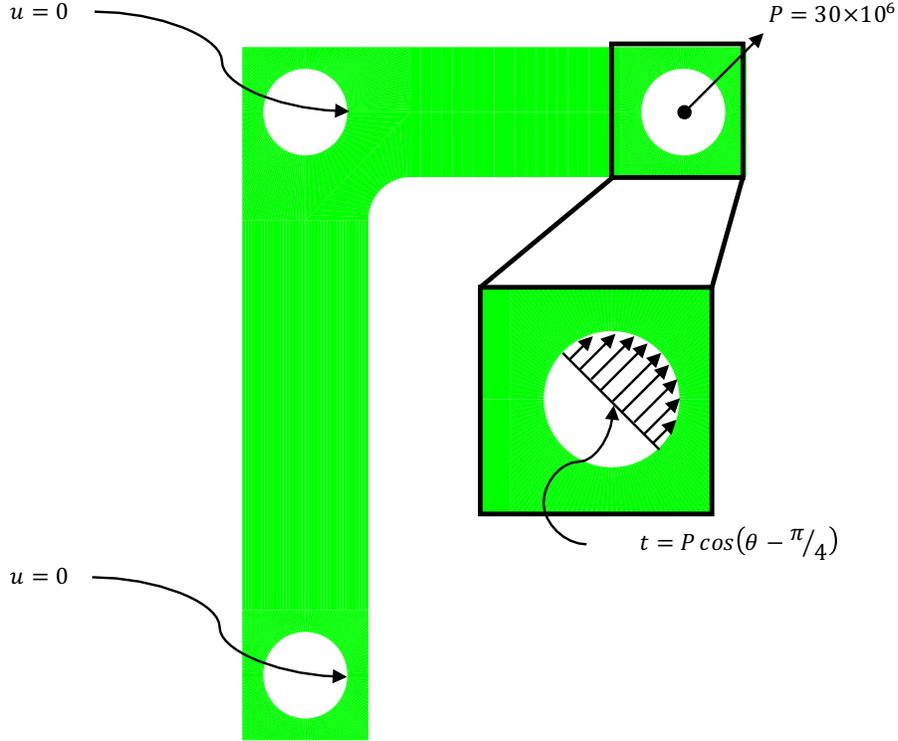


Figure 15: The boundary conditions and applied loading used in the L-Bracket problem. The lower and center L-Bracket holes are constrained with homogeneous Dirichlet boundary conditions. The loading on the remaining hole is cosine distributed for  $-\pi/4 \leq \theta \leq 3\pi/4$  and is applied as a Neumann traction.

are shown below.

$$\mathcal{D}_{L,\text{minor}} = \left( \begin{array}{c} 5.5 \leq M \leq 6.5 \\ 3 \leq N \leq 4 \\ 1 \leq L_1 \leq 1.5 \\ 1 \leq H_3 \leq 1.5 \\ 0.15 \leq R_f \leq 0.25 \\ 0.15 \leq a_1 \leq 0.3 \\ 0 \leq e_1 \leq 0.2 \\ -0.2 \leq t_{1h} \leq 0.2 \\ -0.2 \leq t_{1k} \leq 0.2 \\ 0.15 \leq a_2 \leq 0.3 \\ 0 \leq e_2 \leq 0.2 \\ -0.2 \leq t_{2h} \leq 0.2 \\ -0.2 \leq t_{2k} \leq 0.2 \\ 0.15 \leq a_3 \leq 0.3 \\ 0 \leq e_3 \leq 0.2 \\ -0.2 \leq t_{3h} \leq 0.2 \\ -0.2 \leq t_{3k} \leq 0.2 \end{array} \right) \quad \mathcal{D}_{L,\text{extreme}} = \left( \begin{array}{c} 5.5 \leq M \leq 6.5 \\ 3 \leq N \leq 4 \\ 1 \leq L_1 \leq 1.5 \\ 1 \leq H_3 \leq 1.5 \\ 0.15 \leq R_f \leq 0.25 \\ 0.15 \leq a_1 \leq 0.35 \\ 0 \leq e_1 \leq 0.3 \\ -0.6 \leq t_{1h} \leq 0.6 \\ -0.6 \leq t_{1k} \leq 0.6 \\ 0.15 \leq a_2 \leq 0.35 \\ 0 \leq e_2 \leq 0.3 \\ -0.6 \leq t_{2h} \leq 0.6 \\ -0.6 \leq t_{2k} \leq 0.6 \\ 0.15 \leq a_3 \leq 0.35 \\ 0 \leq e_3 \leq 0.3 \\ -0.6 \leq t_{3h} \leq 0.6 \\ -0.6 \leq t_{3k} \leq 0.6 \end{array} \right)$$

The first design space is “minor” in the sense that the admissible design parameter perturbations lie in a localized neighborhood about a nominal design. The second design space is “extreme” since much larger perturbations and imperfections are admissible in this space. The two spaces are considered to assess the moderate dimensionality of the problem on the effectivity of the design space exploration paradigm. Representative geometries from these two design spaces are depicted in Figures 16 and 17 respectively.

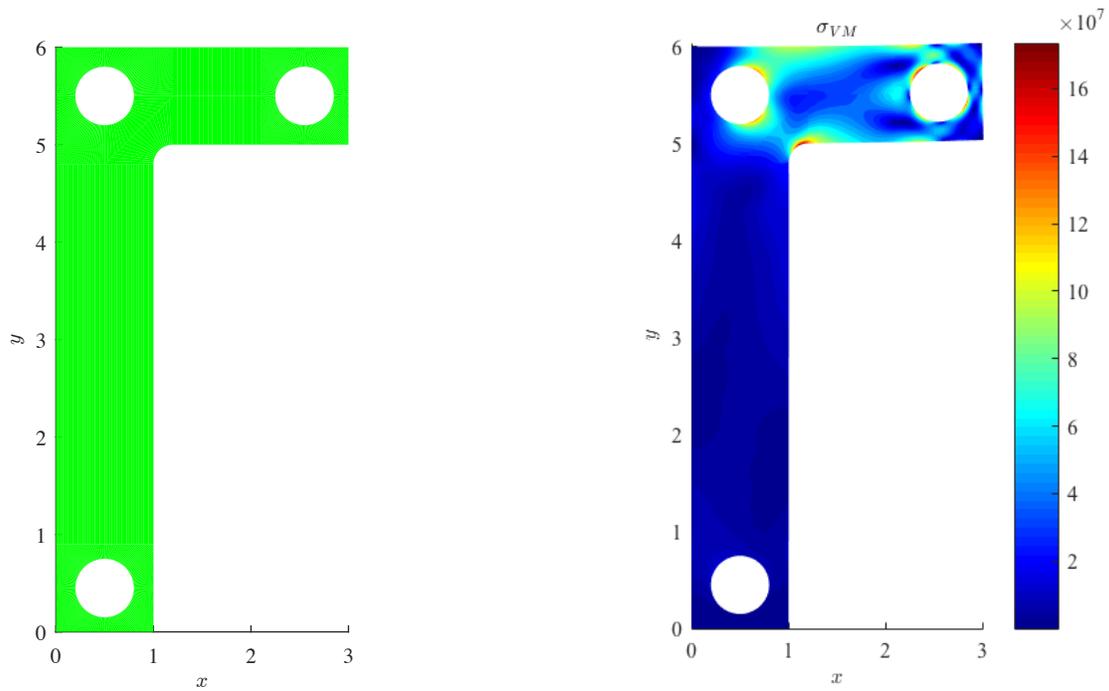


Figure 16: A sample geometry contained within  $\mathcal{P}_{p_\ell}(\mathcal{D}_{L,\text{minor}})$  (left) along with the resulting Von Mises stress field (right).

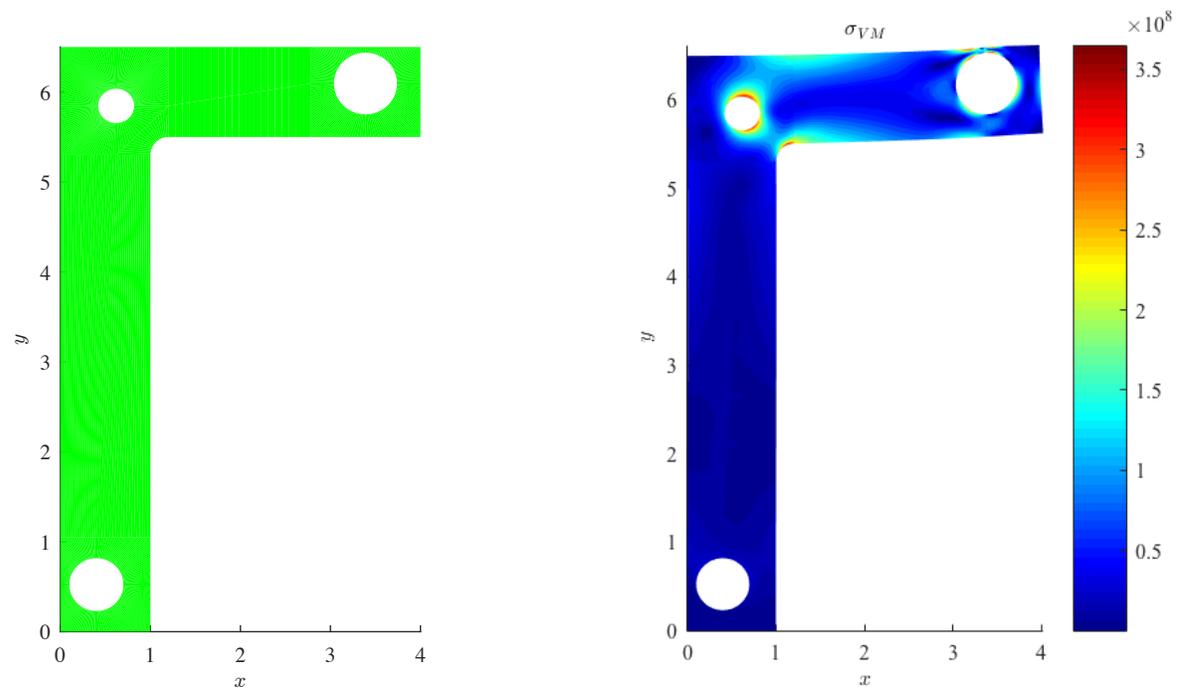


Figure 17: A sample geometry within  $\mathcal{P}_{p_\ell}(\mathcal{D}_{L,\text{extreme}})$  (left) along with the resulting Von Mises stress field (right).

Table 9: Results for the Flat L-Bracket with a modal solution manifold representation of the  $\mathcal{D}_{L,\text{minor}}$  using the sparse, delayed Kronrod-Patterson with an isotropic Legendre basis. In the table below,  $\ell$  denotes the level parameter,  $n_{dof}$  are the number of basis functions contained in  $\mathcal{P}_{p_\ell}(\mathcal{D}_{L,\text{minor}})$  and  $n_\ell$  are the number of function evaluations required to accurately capture the pseudospectral coefficients.

Collocation Scheme	Number of Elements	$\ell$	$n_{dof}$	$n_\ell$	Average $L^2$ error	Average Energy Error
Kronrod-Patterson	480	1	1	1	$2.3589 \times 10^{-5}$	$1.3782 \times 10^2$
		2	18	35	$5.3034 \times 10^{-6}$	$4.8530 \times 10^1$
		3	171	579	$1.4566 \times 10^{-6}$	$1.7515 \times 10^1$
		4	1140	6087	$4.9801 \times 10^{-7}$	3.5950

To ensure that only sensical geometries live in these design spaces, we define the position of the L-Bracket holes' center through a dimensionless design parameter which designates the percentage offset from the true center. The  $t_{ih}$  and  $t_{ik}$ ,  $i = 1, 2, 3$  parameters above define the hole offset via a convex combination of related design parameters, for example  $h_1(t_{1h}) = (a_1 - L_1/2)t_{1h} + L_1/2$ . The remaining quantities of  $k_1, h_2, k_2, h_3, k_3$ , are defined in a similar manner.

It should be noted that the L-Bracket problem is motivated by the design space exploration problem of identifying suitable geometric tolerances for manufacturing. That is, one is often interested in the design question: *How large may we allow a set of tolerances to be without compromising performance?* Our framework allows one to address this question in a direct and efficient manner.

### 5.2.2 Convergence Analysis

A convergence analysis is performed for the pseudospectral model of the solution manifold with the Flat L-Bracket. In particular, we measure the error in both the  $L^2$  and energy-norms to assess the convergence of the displacement and strain energy fields, respectively. These norms are given by:

$$\|e^h\|_{L^2}^2 = \int_{\Omega} |e^h|^2 d\Omega \quad \text{and} \quad \|e^h\|_e^2 = \int_{\Omega} \varepsilon(e^h) : D : \varepsilon(e^h) d\Omega$$

The results of the convergence analysis are shown below for the  $\mathcal{D}_{L,\text{minor}}$  and  $\mathcal{D}_{L,\text{extreme}}$  in Tables 9 and 10, respectively. To ensure the accuracy of the figures presented, the  $L^2$  and energy norms are computed for a total of 10 randomly-selected admissible geometries in the design space and then averaged.

From these tables, we see that our surrogate modeling strategy quickly converges as the polynomial degree  $p_\ell$  is increased. Similar to the Scordelis-Lo roof, the  $L^2$ -error converges quicker than the energy error. The effectiveness of our surrogate modeling strategy is portrayed through these tables since quick convergence is seen for geometries in both  $\mathcal{D}_{L,\text{minor}}$  and  $\mathcal{D}_{L,\text{extreme}}$ .

## 5.3 Wind Turbine Blade

As was discussed in Section 2, the inconsistent geometric descriptions used in design and analysis models constitute a significant rift in modern engineering work flows. While IGA addresses this issue fundamentally, many industrial design strategies remain disconnected and iterative, requiring repetitive user interaction. Typically, a designer first develops a CAD model using a particular software platform. The model can then be uploaded into a CAE platform to convert the geometry model information into mesh information suitable for analysis. Upon completing analysis, if the user wishes to change a design parameter, the entire process is repeated.

Previous work addressed this challenge by constructing a computational framework capable of performing

Table 10: Results for the Flat L-Bracket with a modal solution manifold representation of the  $\mathcal{D}_{L,\text{extreme}}$  using the sparse, delayed Kronrod-Patterson with an isotropic Legendre basis. In the table below,  $\ell$  denotes the level parameter,  $n_{dof}$  are the number of basis functions contained in  $\mathcal{P}_{p_\ell}(\mathcal{D}_{L,\text{extreme}})$  and  $n_\ell$  are the number of function evaluations required to accurately capture the pseudospectral coefficients.

Collocation Scheme	Number of Elements	$\ell$	$n_{dof}$	$n_\ell$	Average $L^2$ error	Average Energy Error
Kronrod-Patterson	480	1	1	1	$1.9409 \times 10^{-5}$	$2.3368 \times 10^2$
		2	18	35	$9.9425 \times 10^{-6}$	$1.0326 \times 10^2$
		3	171	579	$4.0818 \times 10^{-6}$	$5.2883 \times 10^1$
		4	1140	6087	$1.4370 \times 10^{-6}$	$2.2798 \times 10^1$

NURBS-based parametric design, isogeometric analysis, and result visualization seamlessly in a single software environment. The framework is unique in that it is capable of utilizing complex, CAD-based geometric operations to generate parametric models for IGA. Basic heuristic methods were employed to perform optimization of a subset of design variables of an NREL 5MW wind turbine blade, a complex geometry defined by many design variables in [31]. However, given the seamlessness of the framework, it can be readily modified to incorporate design space discretization techniques, demonstrating the viability of such techniques for industrial applications.

### 5.3.1 Problem Formulation and Methodology

Here we consider application of our surrogate modeling methodology to the parametric design and analysis of an NREL 5MW wind turbine blade within the aforementioned design space exploration framework. Our goals are twofold: (i) to demonstrate that our surrogate modeling strategy seamlessly extends to the nonlinear and time-dependent settings and (ii) to demonstrate the effectiveness of our methodology in the context of an industrial-strength application.

Wind turbine blade geometries are typically defined by a set of parameters at discrete locations, or “stations,” along the length of the blade. The usual parameters are radial location of the section along the blade, chord length, airfoil shape, and twist degree. These parameters define the shape and orientation of each two-dimensional station in three-dimensional space; lofting all airfoil profiles produces a wind turbine blade shell geometry. In [35] a complete definition of a wind turbine blade geometry intended for use in 5MW offshore applications.

For this paper, all geometries are considered to have the same loading configuration. The root edge of the blade is clamped by eliminating all motion of the two innermost rings of control points, and a uniform flapwise traction of 250 Pa is applied in the reference configuration. The wind turbine blade is modeled as a nonlinear Kirchhoff-Love shell with a composite material model. A simplified composite layup is also used, shown in Figure 18. The blade is constructed primarily of a thin shell of uni-directional carbon, indicated by green, covering the entire blade, with additional uni-directional carbon thickness in the spar cap zone, indicated by purple, and buildup of SNLTriax on the root, indicated by blue. See [53] for details about material properties. Moreover, the T-spline analysis model is comprised of 1440 elements. From the traction loading, the shell model is iteratively solved until reaching its maximum tip displacement. The resulting displacement field in this state is the output which is used in our design space exploration surrogate modeling paradigm.

### 5.3.2 Implementation

For implementation, we employed and extended an isogeometric Parametric Design Optimization (PDO) framework based on the CAD software Rhinoceros 3D (Rhino) [31]. Rhino has a NURBS-based geometry kernel,

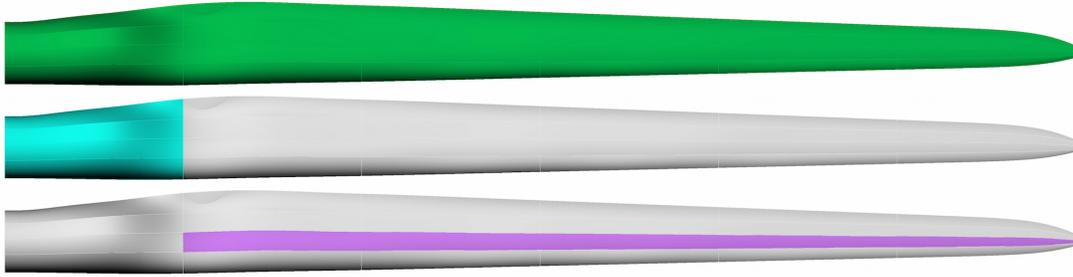


Figure 18: Simplified composite layup used for wind turbine optimization. Green color (top) indicates base uni-directional carbon across entire blade, blue color (middle) indicates root buildup of SNLTriax, and purple color (bottom) indicates spar cap region made up of additional uni-directional carbon.

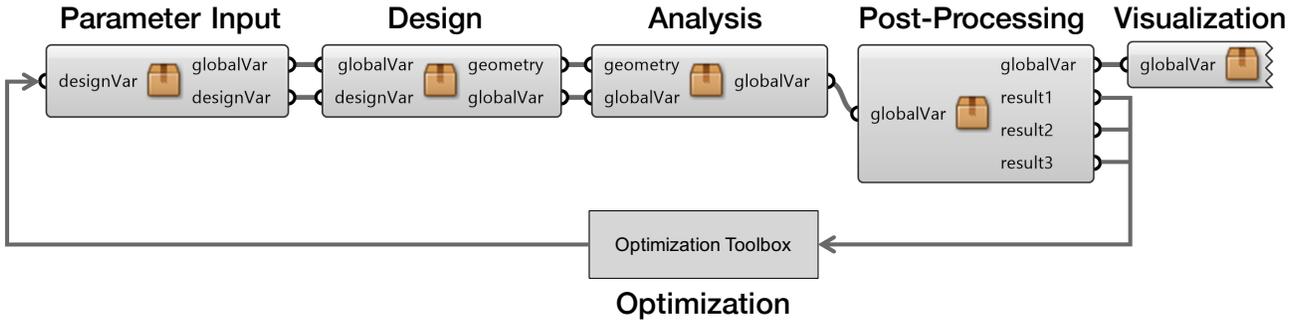


Figure 19: The basic layout of the isogeometric PDO framework, laid out in Grasshopper 3D, a visual programming plugin for Rhino, configured for iterative use.

making it an attractive choice for IGA, and also features Grasshopper, a visual programming interface that enables parametric design within Rhino. The basic structure of the isogeometric PDO framework configured for iterative use is shown in Figure 19.

As Figure 19 demonstrates, each of the stages in the entire design and analysis work flow are contained within a single software platform. The rectangular components in Figure 19 are referred to as Grasshopper “clusters” and each contains a subset of predefined Grasshopper functions and C# scripting components. The “Design” cluster, for example, contains the appropriate geometric functions, such as point interpolation, curve rotation and translation, and lofts, necessary to construct a NURBS-based engineering model. The exact contents of the “Design” cluster will of course depend on the engineering application. The “Analysis” cluster contains functions that output the geometry, without performing meshing, and call the analysis code. The analysis code in these examples is IGA-based and seeks the midsurface displacement of a composite Kirchhoff-Love shell, originally formulated in [5] and reproduced in [31]. The results are then directly read and visualized within the Rhino viewport using the components inside the “Visualization” cluster.

The approach described above unites the paradigms of parametric design and high-fidelity FEA, allowing seamless design and analysis of complex engineering models within a single workspace. In the configuration shown in Figure 19, however, the approach is limited in that it relies heavily on iteration. Hence, a designer cannot actively interrogate a particular point in the design space. Instead, once a point is selected, analysis results are available only after the entire structural analysis is completed.

In this work, the Grasshopper-based isogeometric PDO framework is reconfigured for use with the design space discretization methodology presented in the previous sections. As Figure 20 illustrates, the “Design” and “Analysis” clusters can first be used to automatically analyze the displacements  $\mathbf{d}(\boldsymbol{\mu}_i)$  at a predetermined set of collocation points  $\boldsymbol{\mu}_i$ . This analysis information is then utilized to construct the surrogate model which approximates the solution manifold  $\mathbf{d}(\boldsymbol{\mu})$ . In particular, given a design variable  $\boldsymbol{\mu}$ , the associated displacement vector

## Initial analysis stage



## Design exploration stage

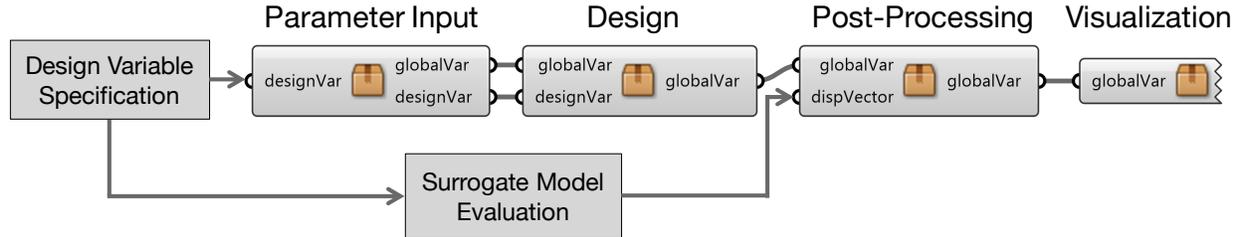


Figure 20: The isogeometric PDO framework configured for use with the design space exploration methodology, including analysis-heavy, iterative offline stage (top) and GUI-driven, rapidly evaluated design exploration stage (bottom).

can be efficiently produced via evaluation of the solution manifold. As shown in Figure 20, this displacement vector is provided to the Grasshopper interface such that visualization of the solution, based on the original geometry’s position and nodal displacements, is performed. The designer can therefore use the GUI to select a design point of interest and quickly see the resultant geometry and any other post-processed results within the Rhino viewport, as shown in Figure 20.

We emphasize that, in the workflow demonstrated by Figure 20, only the initial, offline stage requires relatively computationally expensive finite element calculations. Once the results of these analyses have been used to construct the surrogate model however, a GUI can be used to rapidly explore the entirety of the design space and associated high-fidelity analysis results.

The NREL 5MW wind turbine blade in the PDO framework uses 19 radial locations, the parameters described in [35], each with independent chord lengths and airfoil cross-sections, to generate a turbine blade geometry. To reduce the dimensionality of the design space, we construct a quadratic B-spline with 4 parametric degrees of freedom which describes the chord line,  $c$ , as a function of radial position,  $r$ , as our design variable. Moreover, we do not consider the effect of varying airfoil cross-section on the displacement field and instead hold these nominal specifications constant. We define the turbine blade design space  $\mathcal{D}_{\text{turb}} \subset \mathbb{R}^4$  via:

$$\mathcal{D}_{\text{turb}} := \{\boldsymbol{\mu} = (P_{5x}, P_{5y}, P_{7x}, P_{7y}) \mid 9 \leq P_{5x} \leq 13, \quad 4.1 \leq P_{5y} \leq 4.7, \quad 22 \leq P_{7x} \leq 30, \quad 4.2 \leq P_{7y} \leq 4.6\}$$

where these parameters define the control variables  $\vec{P}_3$  and  $\vec{P}_5$  which describe the B-spline curve given by

$$c_{\boldsymbol{\mu}}(r) = \sum_{i=1}^8 \vec{P}_i(\boldsymbol{\mu}) N_i(r)$$

Figure 21 depicts the chord line of design space,  $\mathcal{D}_{\text{turb}}$ , for the wind turbine blade.

In addition to the design space presented in Figure 21, a representative set of geometries in this design space are shown in Figures 22 and 23. These figures demonstrate how altering the chord design parameter affects the resulting geometry.

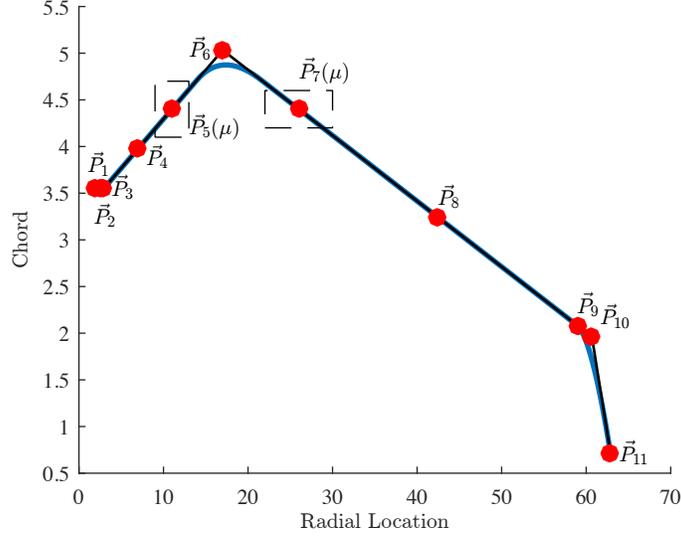


Figure 21: The chord line describing design space of an NREL 5MW wind turbine blade,  $\mathcal{D}_{\text{turb}}$ . The chord line as a function of radial location is constructed using a piecewise quadratic B-spline. The two red control points in the dashed box,  $\vec{P}_5$  and  $\vec{P}_7$  are the design parameters for the turbine blade, which are free to move within the box. The B-spline curve shown is the nominal configuration for the turbine blade.

### 5.3.3 Convergence Analysis

Similar to before, we perform a convergence analysis and assessment on the wind turbine blade using a nodal representation of the solution manifold. This choice of surrogate model was informed by the results of the Scordelis-Lo roof namely, a nodal surrogate modeling strategy is preferred for a low-dimensional design space. In particular, we measure the errors in the displacement field and maximum in-plane strain field, both of which are measured in the  $L^2$ -norm. For  $e^h = u_{\text{IGA}}^h - u_{\text{surrogate}}^h$  and  $\varepsilon(e^h) = \varepsilon(u_{\text{IGA}}^h) - \varepsilon(u_{\text{surrogate}}^h)$ , these fields are given by:

$$\|e^h\|_{L^2}^2 = \int_{\Omega} |e^h|^2 d\Omega \quad \text{and} \quad \|e^h\|_{\varepsilon}^2 = \|\varepsilon(e^h)\|_{L^2}^2 = \int_{\Omega} |\varepsilon(e^h)|^2 d\Omega$$

These norms effectively assess the accuracy of the surrogate model's capability of representing the displacement field as well as the corresponding strain field. Table 11 below depicts the convergence behavior of the nodal surrogate model constructed using the Lagrange polynomials with the tensor-product Gauss-Legendre sampling scheme.

We see that both the  $L^2$  and  $\varepsilon$  errors decrease very quickly from the first level to the third level. In fact, both errors drop by over an order of magnitude from the first level to the third level. However, it should be noted that the convergence rates slow by the fourth level. This is because the coarse-scale solution behavior across the design space has been resolved while a high polynomial degree is required to accurately capture the remaining fine-scale solution features. Nonetheless, the average  $L^2$  and  $\varepsilon$  errors are quite small and acceptable by the 4<sup>th</sup> level. In addition to the convergence table above, we present the strain fields for a representative set of turbine geometries present in  $\mathcal{D}_{\text{turb}}$  in Figure 24. This figure demonstrates that our surrogate modeling strategy is very effective in resolving the solution fields of interest in nonlinear structural mechanics, namely the strain and stress fields.

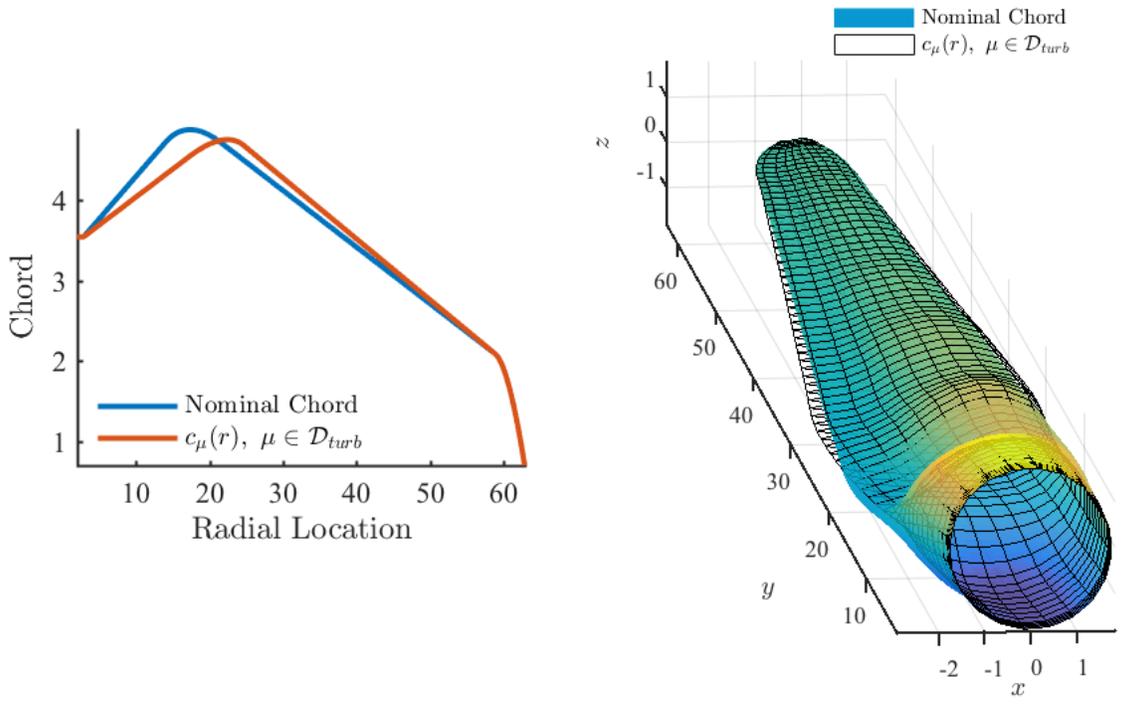


Figure 22: A sample geometry contained in  $\mathcal{D}_{turb}$ . This design has a tip wise shift in the maximal chord location.

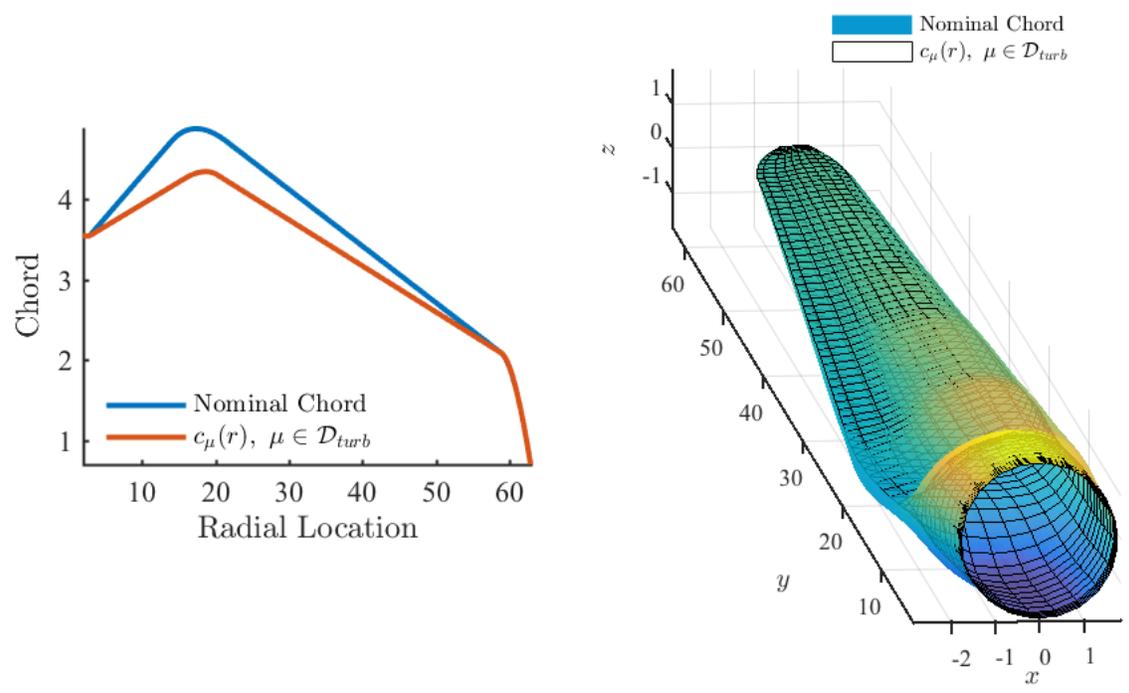


Figure 23: A sample geometry contained in  $\mathcal{D}_{turb}$ . This design has a smaller maximal chord location.

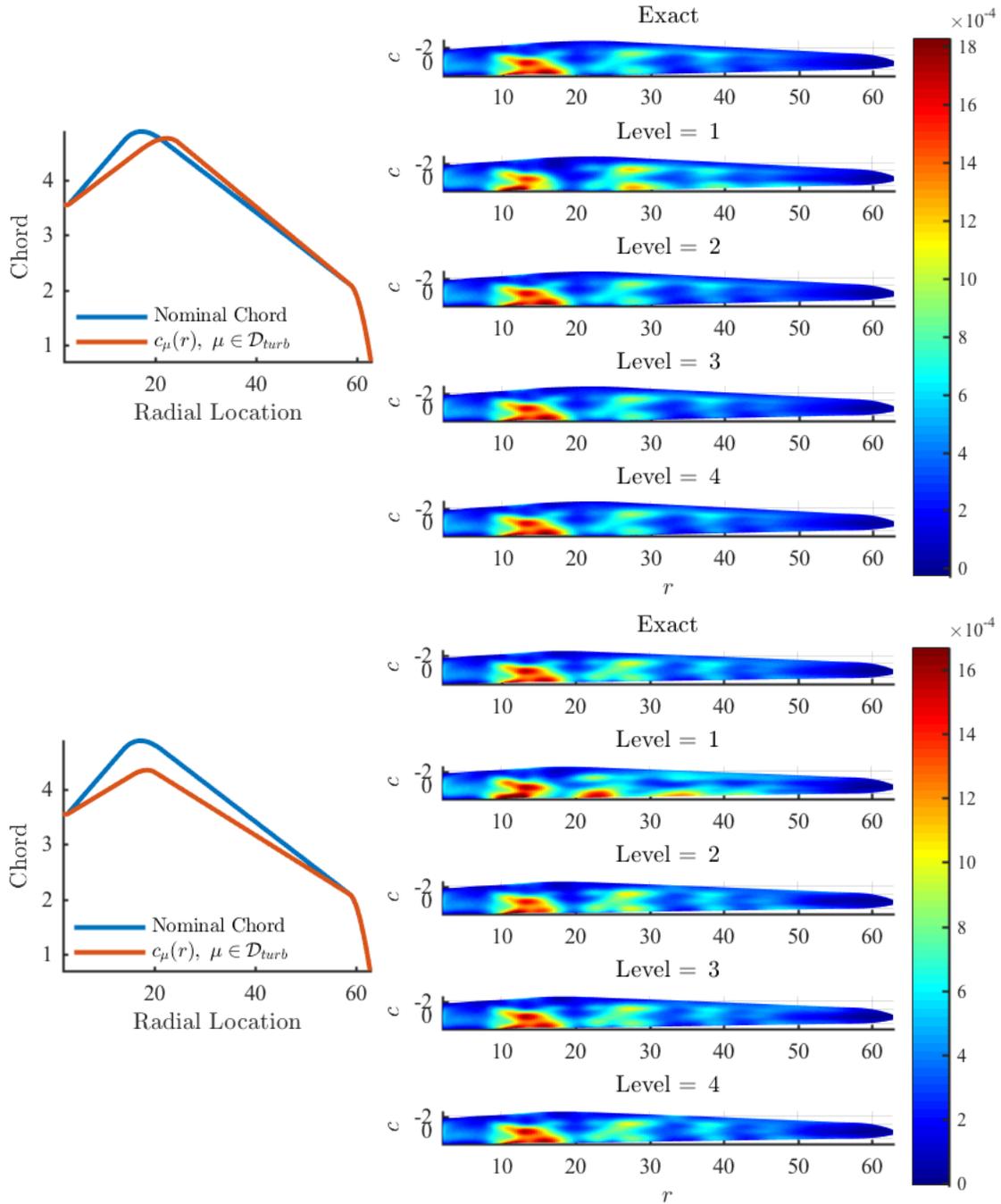


Figure 24: In the left plots, we present representative chord lines associated with geometries admissible to  $\mathcal{D}_{turb}$ . The blue lines correspond to the nominal turbine geometry while the red lines are the chord lines associated with a turbine geometry present in the design space under consideration. In the right column of plots, the corresponding strain fields for the exact IGA solution, as well as the nodal surrogate model for levels 1 through 4, are shown.

Table 11: Results for an NREL 5MW with a nodal discretization of the design space using tensor-product Gauss-Legendre nodes. These results use 10 randomly chosen design variables and compares the isogeometric solution to the solution predicted by the nodal surrogate design space representation. In the table below,  $\ell$  denotes the level parameter,  $n_{dof}$  are the number of basis functions contained in  $\mathcal{Q}_{p\ell}(\mathcal{D}_{\text{turb}})$  and  $n_\ell$  are the number of function evaluations.

Collocation Scheme	Number of Elements	$\ell$	$n_{dof}$	$n_\ell$	Average $L^2$ error	Average $\varepsilon$ Error
Gauss-Legendre	1440	1	1	1	$3.8484 \times 10^{-1}$	$3.0465 \times 10^{-3}$
		2	16	16	$5.2918 \times 10^{-2}$	$3.3269 \times 10^{-4}$
		3	81	81	$2.1019 \times 10^{-2}$	$1.4241 \times 10^{-4}$
		4	256	256	$9.0516 \times 10^{-3}$	$9.7949 \times 10^{-5}$

## 6 Conclusions and Future Work

In this paper, we have presented a framework for rapid and efficient design space exploration based on a combination of IGA and smart sampling. Although the methodology was presented in the context of structural mechanics, the modeling paradigm is agnostic in terms of its range of applicability. Moreover, this framework provides a means for obtaining an approximation to the full-system response whereas state-of-the-art approaches only provide exploration capability for quantities of interest present in the solution field. Various sampling techniques were considered where tensor-product grids provided a high-fidelity representation of the solution space for low-dimensional problems and Smolyak-style sparse grids were effective in alleviating the curse of dimensionality in solution fields comprised of a few dominant displacement mode shapes for high-dimensional problems. Both nodal and modal representations of the solution manifold were considered through the use of interpolating polynomial basis functions and pseudospectral expansions. The methodology was demonstrated to be effective in exploring the design spaces of the Scordelis-Lo roof, a Flat L-Bracket, and an NREL 5MW wind turbine blade.

In short, the following conclusions were drawn through the numerical experiments conducted in this paper:

1. In general, both nodal and modal solution manifold representations provide the same fidelity of the surrogate model per degree of freedom and at the same computational cost for the case of tensor-product polynomial design space discretizations. However, the nodal representation performed slightly better than modal in terms of solution accuracy. This is due to numerical integration errors incurred while computing the pseudospectral coefficients through numerical integration.
2. In the case where the solution is inherently sparse with little cross-derivative information, isotropic polynomial design space discretizations out-performed tensor-product polynomial design space discretizations in terms of accuracy versus cost. In the case where cross-derivative fidelity is necessary, as in the example of the Scordelis-Lo roof, the tensor-product discretization performed better.
3. Both tensor-product and sparse quadrature rules were considered for sampling the design space and numerical integration. Tensor-product quadrature rules were more effective at computing the pseudospectral coefficients as expected, however computing the resulting approximation error through both approaches generally maintained the same order of magnitude. As the sparse rules are considerably cheaper, it is recommended that they be used in practice.
4. For high-dimensional problems, as in the example of the Flat L-Bracket, isotropic design space discretizations perform well in conjunction with sparse quadrature rules while tensor-product design space discretizations are not feasible due to the curse of dimensionality.
5. In all cases, Kronrod-Patterson achieved a higher accuracy per computational cost in comparison to Clenshaw-Curtis.

Overall, we found that the design space exploration methodology was an excellent and economical solution for problems with solution manifolds that have inherently “smooth” characteristics. In other words, solution manifolds which have a response dominated by global feature manipulation were the most effective in this framework. This was exemplified in the cases of the Scordelis-Lo roof, the Flat L-Bracket, and an NREL 5MW wind turbine blade, where the solution manifold is comprised of a sparse set of displacement modes. The appropriate combination and scaling of these modes, afforded through the methodologies presented throughout this paper, resulted in an accurate approximation to the exact displacement profile.

It should be remarked, however, that our methodology is not quite as effective in representing “rough” solution manifolds which have a response dominated by localized features. In particular, it does not work well for problems with solution manifolds that are discontinuous or have discontinuous derivatives which are associated with, for instance, bifurcations. This is due to the fact that we have essentially considered a  $p$ -adaptive design space exploration paradigm in this paper. In future work, the issue of solution manifold “roughness” will be addressed through dimension-adaptive refinement procedures. In particular, extending our framework to an  $hp$ -adaptive framework may remedy the issue of roughness by effectively splitting the design space along discontinuous and non-differentiable loci. Note that  $hp$  methods based on isotropic design space mesh refinement suffer from the curse of dimensionality in a tensor-product setting [17]. However, employing this refinement protocol with anisotropic refinement will alleviate these concerns [25, 9]

The methodology presented in this paper also lays the foundation for a plethora of research topics from shape optimization to geometric uncertainty quantification, analogous to those presented in [30, 55, 49, 45, 16, 61, 62]. Future work will leverage the surrogate modeling methodology presented herein for stochastic problems, where classic uncertainty quantification tools and techniques are readily available. In this setting, the design parameters can be interpreted as random variables which capture inherent modeling and manufacturing uncertainties. This lends to the notion of a stochastic geometric family which can be used to quantify the impact of tolerances on engineering designs, namely how geometric uncertainties in design propagate through the solution manifold. Similar ideas have been considered in the field of shape uncertainty quantification, on the analysis of random domains, such as those presented in [29, 15, 22]. Additionally, tools such as compressed sensing shows promise in obtaining coefficients to high-order modes, and consequently an increase in surrogate model fidelity, without incurring additional sampling expense in solution fields which are inherently sparse [20, 28].

## 7 Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research under Grant No. FA9550-14-1-0113. The authors would also like to thank Alireza Doostan for helpful discussions that greatly improved the quality of this work. A. J. Herrema was/is supported by the U.S. National Science Foundation (NSF) Grant No. DGE-1069283 which funds the activities of the Integrative Graduate Education and Research Traineeship (IGERT) in Wind Energy Science, Engineering, and Policy (WESEP) at Iowa State University. We would also like to thank the reviewers for their constructive feedback which contributed to improve the quality of this work.

## 8 References

- [1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions*. Dover Publications, 1965.
- [2] K. E. Atkinson. *An Introduction to Numerical Analysis*. Wiley, 1978.
- [3] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.
- [4] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. Isogeometric analysis using t-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5):229–263, 2010.

- [5] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *IJNMF*, 65:236–253, 2011.
- [6] P. Benner, A. Onwunta, and M. Stoll. Low-rank solution of unsteady diffusion equations with stochastic coefficients. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):622–649, 2015.
- [7] G. Blatman and B. Sudret. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25(2):183–197, 2010.
- [8] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on bézier extraction of nurbs. *International Journal for Numerical Methods in Engineering*, 87(1-5):15–47, 2011.
- [9] J. Burkardt. The combining coefficient for anisotropic sparse grids. *Interdisciplinary Center for Applied Mathematics & Information Technology Department, Virginia Tech*, 2010.
- [10] J. Burkardt. Slow exponential growth for clenshaw curtis sparse grids, 2012.
- [11] D. Chapelle and K. J. Bathe. *The Finite Element Analysis of Shells - Fundamentals*. Computational Fluid and Solid Mechanics. Springer, 2003.
- [12] C. W. Clenshaw and A. R. Curtis. A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2(1):197–205, 1960.
- [13] P. G. Constantine, M. S. Eldred, and E. T. Phipps. Sparse pseudospectral approximation method. *Computer Methods in Applied Mechanics and Engineering*, 229:1–12, 2012.
- [14] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, 1st edition, 2009.
- [15] M. Dambrine, H. Harbrecht, and B. Puig. Computing quantities of interest for random domains with second order shape sensitivity analysis. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(5):1285–1302, 2015.
- [16] M. C. Delfour and J.-P. Zolésio. *Shapes and Geometries*. Society for Industrial and Applied Mathematics, second edition, 2011.
- [17] L. Demkowicz. *Computing with hp-Adaptive Finite Elements: Volume 1 One and Two Dimensional Elliptic and Maxwell problems*. CRC Press, 2006.
- [18] A. Doostan, R. G. Ghanem, and J. Red-Horse. Stochastic model reduction for chaos representations. *Computer Methods in Applied Mechanics and Engineering*, 196(37):3951–3966, 2007.
- [19] A. Doostan and H. Owhadi. A non-adapted sparse approximation of pdes with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.
- [20] A. Doostan and H. Owhadi. A non-adapted sparse approximation of pdes with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.
- [21] M. R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with t-splines. *Computer methods in applied mechanics and engineering*, 199(5):264–275, 2010.
- [22] K. Eppler, H. Harbrecht, and R. Schneider. On convergence in elliptic shape optimization. *SIAM Journal on Control and Optimization*, 46(1):61–83, 2007.
- [23] B. Fornberg. *A Practical Guide to Pseudospectral Methods*, volume 1. Cambridge university press, 1998.
- [24] B. Ganapathysubramanian and N. Zabarar. Sparse grid collocation schemes for stochastic natural convection problems. *Journal of Computational Physics*, 225:652–685, 2007.

- [25] T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, 71(1):65–87, 2003.
- [26] G. H. Golub and J. H. Welsch. Calculation of gauss quadrature rules. *Mathematics of computation*, 23(106):221–230, 1969.
- [27] M. Grigoriu. Parametric models for samples of random functions. *Journal of Computational Physics*, 297:47–71, 2015.
- [28] J. Hampton and A. Doostan. Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies. *Journal of Computational Physics*, 280:363–386, 2015.
- [29] H. Harbrecht. On output functionals of boundary value problems on stochastic domains. *Mathematical Methods in the Applied Sciences*, 33(1):91–102, 2010.
- [30] J. Haslinger and R. A. E. Mäkinen. *Introduction to Shape Optimization*. Society for Industrial and Applied Mathematics, 2003.
- [31] A. J. Herrema, N. M. Wiese, C. N. Darling, A. Krishnamurthy, B. Ganapathysubramanian, and M.-C. Hsu. A framework for parametric design optimization using isogeometric analysis. *CMAME*, 2016. In review.
- [32] M.-C. Hsu, C. Wang, A. J. Herrema, D. Schillinger, A. Ghoshal, and Y. Bazilevs. An interactive geometry modeling and parametric design platform for isogeometric analysis. *Computers & Mathematics with Applications*, 70(7):1481–1500, 2015.
- [33] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Courier Corporation, 2012.
- [34] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39):4135–4195, 2005.
- [35] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, Golden, CO, 2009.
- [36] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi. Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration. In *Proceedings of the conference on Design, Automation and Test in Europe*, pages 423–428. European Design and Automation Association, 2009.
- [37] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with kirchhoff-love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49):3902–3914, 2009.
- [38] D. Laurie. Calculation of gauss-kronrod quadrature rules. *Mathematics of Computation of the American Mathematical Society*, 66(219):1133–1145, 1997.
- [39] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, and M. A. Scott. On linear independence of t-spline blending functions. *Computer Aided Geometric Design*, 29(1):63–76, 2012.
- [40] A. Manzoni, F. Salmoiraghi, and L. Heltai. Reduced basis isogeometric methods (rb-iga) for the real-time simulation of potential flows about parametrized naca airfoils. *Computer Methods in Applied Mechanics and Engineering*, 284:1147–1180, 2015.
- [41] H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41:35–52, 2009.
- [42] L. W.-T. Ng and M. S. Eldred. Multifidelity uncertainty quantification using nonintrusive polynomial chaos and stochastic collocation. In *Proceedings of the 14th AIAA Non-Deterministic Approaches Conference, number AIAA-2012-1852, Honolulu, HI*, volume 43, 2012.

- [43] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2411–2442, 2008.
- [44] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.
- [45] A. Novruzzi and M. Pierre. Structure of shape derivatives. *Journal of Evolution Equations*, 2(3):365–382, 2002.
- [46] T. N. L. Patterson. The optimum addition of points to quadrature formulae. *Mathematics of Computation*, 22(104):847–856, 1968.
- [47] K. Petras. Smolyak cubature of given polynomial degree with few nodes for increasing dimension. *Numerische Mathematik*, 93(4):729–753, 2003.
- [48] L. Piegl and W. Tiller. *The NURBS Book*. Springer Science & Business Media, 2012.
- [49] O. Pironneau. *Optimal shape design for elliptic systems*. Springer Science & Business Media, 2012.
- [50] S. S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International journal for numerical methods in fluids*, 34(5):425–448, 2000.
- [51] M.T. Reagan, H.N. Najm, B.J. Debusschere, O.P. Le Maître, O.M. Knio, and R. G. Ghanem. Spectral stochastic uncertainty quantification in chemical systems. *Combustion Theory and Modelling*, 8(3):607–632, 2004.
- [52] H. Reid. Clenshaw curtis quadrature, 2014. Available at <http://homerreid.dyndns.org/teaching/18.330/Notes/ClenshawCurtis.pdf>.
- [53] B. R. Resor. Definition of a 5MW/61.5m wind turbine blade reference model. Technical Report SAND2013-2569, Sandia National Laboratories, Albuquerque, NM, 2013.
- [54] D. E. Schiavazzi, A. Doostan, G. Iaccarino, and A. L. Marsden. A generalized multi-resolution expansion for uncertainty propagation with application to cardiovascular modeling. *Computer Methods in Applied Mechanics and Engineering*, 2016.
- [55] V. H. Schulz. A riemannian view on shape optimization. *Foundations of Computational Mathematics*, 14(3):483–501, 2014.
- [56] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on bézier extraction of t-splines. *International Journal for Numerical Methods in Engineering*, 88(2):126–156, 2011.
- [57] M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable t-splines. *Computer Methods in Applied Mechanics and Engineering*, 213:206–222, 2012.
- [58] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and t-nurccs. In *ACM transactions on graphics (TOG)*, volume 22, pages 477–484. ACM, 2003.
- [59] F. Vidal-Codina, N.-C. Nguyen, M. B. Giles, and J. Peraire. A model and variance reduction method for computing statistical outputs of stochastic elliptic partial differential equations. *Journal of Computational Physics*, 297:700–720, 2015.
- [60] Y. Xie, G. H. Loh, B. Black, and K. Bernstein. Design space exploration for 3d architectures. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 2(2):65–103, 2006.
- [61] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Computer methods in applied mechanics and engineering*, 191(43):4927–4948, 2002.
- [62] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of computational physics*, 187(1):137–167, 2003.

## Appendix A - Scordelis-Lo Roof Geometric Definition

In this appendix, we provide the control points and weights associated with the parametric Scordelis-Lo roof, shown in Figure 25. As mentioned in Section 5, the computational domain is a quarter of the full roof. Additionally, the roof is constructed using a quadratic set of NURBS functions and then degree elevated to cubics for Kirchhoff-Love shell analysis.

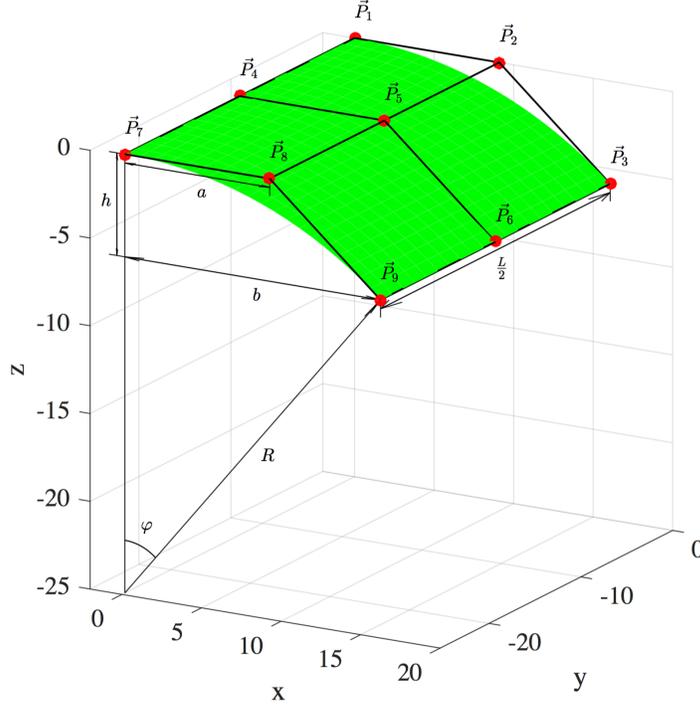


Figure 25: The Scordelis-Lo roof modeled using quadratic NURBS. The intermediate parameters  $h = R(1 - \cos \varphi)$ ,  $a = R \tan \varphi/2$ , and  $b = (R - h) \tan \varphi$  are introduced for ease of notation.

Control Point	Spatial Location	NURBS weight
$\vec{P}_1$	$(0, 0, 0)$	1
$\vec{P}_2$	$(a, 0, 0)$	$\cos \varphi/2$
$\vec{P}_3$	$(b, 0, -h)$	1
$\vec{P}_4$	$(0, -L/4, 0)$	1
$\vec{P}_5$	$(a, -L/4, 0)$	$\cos \varphi/2$
$\vec{P}_6$	$(b, -L/4, -h)$	1
$\vec{P}_7$	$(0, -L/2, 0)$	1
$\vec{P}_8$	$(a, -L/2, 0)$	$\cos \varphi/2$
$\vec{P}_9$	$(b, -L/2, -h)$	1

## Appendix B - Survey of Quadrature Rules

In this appendix, we provide an overview of the various univariate quadrature schemes considered in this paper. In particular, we discuss the Gauss-Legendre, Clenshaw-Curtis, and Kronrod-Patterson schemes. Numerical quadrature rules which approximate the integral of  $f(x)$  over the interval  $(-1, 1)$  are constructed such that:

$$\int_{-1}^1 W(x)f(x)dx = \sum_{i=1}^n w_i f(x_i) + R_n \quad (17)$$

for some weighting function  $W(x) \geq 0$ , where  $w_i$  are the *quadrature weights* and  $x_i$  are the *quadrature nodes*. We strictly consider the case  $W(x) \equiv 1$  herein. These quadrature rules have *polynomial exactness*  $p_n$  if  $R_n = 0$  for all polynomials  $p(x)$  with  $\deg p(x) \leq p_n$ . Numerical approximations over general intervals  $(a, b)$  are obtained via a change of coordinates.

**Gauss-Legendre:** The quadrature points  $\{x_i\}_{i=1}^n$  for the  $n$ -point Gauss-Legendre quadrature scheme are the zeros of  $\mathcal{L}_n(x)$ , the  $n^{\text{th}}$  Legendre polynomial which satisfies:

$$\mathcal{L}_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left[ (x^2 - 1)^n \right] \quad n = 0, 1, 2, \dots \quad (18)$$

while the weights  $\{w_i\}_{i=1}^n$  are defined as [1]:

$$w_i = \frac{2}{(1 - x_i^2)[\mathcal{L}'_n(x_i)]^2}$$

The  $n$ -point Gauss-Legendre quadrature scheme has a polynomial exactness of  $p_n = 2n - 1$ .

**Clenshaw-Curtis:** For  $\ell \in \mathbb{Z}^+$ , the quadrature points  $\{x_i\}_{i=0}^{n_\ell}$  for the  $(n_\ell + 1)$ -point (nested) Clenshaw-Curtis quadrature scheme are given by the Chebyshev points:

$$x_k = \cos\left(\frac{k\pi}{n_\ell}\right), \quad k = 0, 1, 2, \dots, n_\ell = 2^{\ell-1}$$

The weights  $\{w_i\}_{i=0}^{n_\ell}$  are given via [52]:

$$w_i = \sum_{j=0}^n D_{ij} W_j$$

where  $D_{ij}$  is the discrete cosine transform and  $W_j$  are weights in the frequency space:

$$D_{ij} = \begin{cases} 1/n_\ell, & j = 0 \\ 2/n \cos(ij\pi/n_\ell), & j = 1, 2, \dots, n_\ell - 1 \\ (-1)^i/n_\ell, & j = n_\ell \end{cases}, \quad W_j = \begin{cases} 1, & j = 0 \\ \frac{2}{1-j^2}, & j \neq 0, \text{ even} \\ 0, & j \text{ odd} \end{cases}$$

The  $(n_\ell + 1)$ -point Clenshaw-Curtis quadrature scheme has a polynomial exactness of  $p_\ell = n_\ell$ .

**Kronrod-Patterson:** The Kronrod-Patterson family starts with the familiar midpoint rule and the corresponding three-point Kronrod extension [38]. Higher-order rules are then built in a recursive manner, resulting in a nested scheme. Due to its complexity, the construction of higher-order rules is omitted here and the interested reader is pointed to [46] for more information. Table 12 displays the quadrature points and weights associated with the one-, three-, and seven-point Kronrod-Patterson rules.

Table 12: The first three levels of the univariate Kronrod-Patterson quadrature rule used to evaluate integrals of the form (17). Note the nestedness of the Kronrod-Patterson rule with respect to increasing level.

Level 1		Level 2		Level 3	
$x_i$	$w_i$	$x_i$	$w_i$	$x_i$	$w_i$
0	2	-0.774596669241483	0.555555555555556	-0.960491268708020	0.104656226026467
		0	0.888888888888889	-0.774596669241483	0.268488089868333
				-0.434243749346803	0.401397414775962
				0	0.450916538658474
				0.434243749346803	0.401397414775962
		0.774596669241483	0.555555555555556	0.774596669241483	0.268488089868333
		0.960491268708020	0.104656226026467		