Title

Automated shape and thickness optimization for non-matching isogeometric shells using free-form deformation.

Permalink

https://escholarship.org/uc/item/0qb1q0rc

Journal

Engineering with Computers, 40(6)

ISSN

0177-0667

Authors

Zhao, Han
Kamensky, David
Hwang, John
et al.

Publication Date

2024

DOI

10.1007/s00366-024-01947-7

Peer reviewed

Engineering with Computers (2024) 40:3495–3518
https://doi.org/10.1007/s00366-024-01947-7

ORIGINAL ARTICLE

Automated shape and thickness optimization for non-matching isogeometric shells using free-form deformation

Han Zhao[1] · David Kamensky[1] · John T. Hwang[1] · Jiun-Shyan Chen[1,2]

Received: 2 August 2023 / Accepted: 18 January 2024 / Published online: 1 March 2024

Abstract

Isogeometric analysis (IGA) has emerged as a promising approach in the field of structural optimization, benefiting from the seamless integration between the computer-aided design (CAD) geometry and the analysis model by employing non-uniform rational B-splines (NURBS) as basis functions. However, structural optimization for real-world CAD geometries consisting of multiple non-matching NURBS patches remains a challenging task. In this work, we propose a unified formulation for shape and thickness optimization of separately parametrized shell structures by adopting the free-form deformation (FFD) technique, so that continuity with respect to design variables is preserved at patch intersections during optimization. Shell patches are modeled with isogeometric Kirchhoff–Love theory and coupled using a penalty-based method in the analysis. We use Lagrange extraction to link the control points associated with the B-spline FFD block and shell patches, and we perform IGA using the same extraction matrices by taking advantage of existing finite element assembly procedures in the FEniCS partial differential equation (PDE) solution library. Moreover, we enable automated analytical derivative computation by leveraging advanced code generation in FEniCS, thereby facilitating efficient gradient-based optimization algorithms. The framework is validated using a collection of benchmark problems, demonstrating its applications to shape and thickness optimization of aircraft wings with complex shell layouts.

Keywords Isogeometric analysis · Kirchhoff–Love shells · Non-matching coupling · Free-form deformation · Lagrange extraction · Aircraft wing optimization · FEniCS

1 Introduction

Shell structures exhibit exceptional stiffness and strength-to-self-weight ratios, and are extensively employed in various engineering fields, such as aerospace, automotive, and marine engineering [1]. The performance of such structures is greatly influenced by geometric and material properties. Thus, structural optimization plays a vital role in obtaining superior designs for shell structures. In this paper, we present an optimization approach based on free-form deformation (FFD) [2] to achieve the optimal shape and thickness distribution for isogeometric shell structures.

Structural analysis is involved in the optimization process to evaluate the response of the current design and guide the subsequent iterations. The finite element (FE) method [3] is a well-established approach used to approximate PDEs with Lagrange polynomial basis functions. However, the discretization of the computational domain through interconnected simple elements, known as meshing, for complicated geometries is the primary challenge in FE analysis. The FE mesh generation and related process can account for up to 80% of total analysis time [4]. Alternatively, isogeometric analysis (IGA) [5, 6] offers the possibility to bypass FE mesh generation by approximating the solution using the smooth non-uniform rational B-spline (NURBS) [7] basis functions. NURBS is the industrial standard widely used to represent computer-aided design (CAD) models, making IGA an ideal method to streamline the design-through-analysis process.

IGA has gained growing interest since its introduction not only due to the unified description between design and

✉ Jiun-Shyan Chen
jsc137@ucsd.edu

[1] Department of Mechanical and Aerospace Engineering, University of California San Diego, 9500 Gilman Drive, Mail Code 0411, La Jolla, CA 92093, USA

[2] Department of Structural Engineering, University of California San Diego, 9500 Gilman Drive, Mail Code 0085, La Jolla, CA 92093, USA

analysis models but also the regularity provided by NURBS basis functions. The smoothness in splines allows direct discretization of the Kirchhoff–Love shell model [8], a fourth-order PDE that requires at least C^1 continuity with the Galerkin method. Various applications including wind turbines [9–11], bioprosthetic heart valves [12–14], and car hoods [15] have employed the isogeometric Kirchhoff–Love shell formulations [16–20] and demonstrated exceptional results. However, practical CAD geometries are often too complex to be represented by a single tensor product NURBS surface. To make the CAD models with multiple NURBS patches directly available for analysis, [21] introduces a fictitious strip to add bending stiffness at patch interfaces with conforming discretizations. Additionally, various methods, such as Nitsche's method [22, 23], penalty method [24, 25], and super-penalty method [26, 27], have been applied to CAD models with non-matching NURBS surfaces, further expanding the applicability of IGA in dealing with complex geometries.

The seamless integration between CAD and analysis models in IGA makes it a natural choice for design optimization. The updated design in the optimization process can be precisely captured in the analysis, which in turn ensures accurate responses due to the exact geometry representation and excellent approximation capabilities of spline basis functions [28]. Shape optimization using IGA has been investigated extensively [29–34]. Many applications, such as beam structures [35], vibrating membranes [36], shell structures [37, 38], and complex photonic crystals [39], show superior design by employing IGA in optimization. Moreover, topology optimization [40, 41] also benefits from the same spline basis in design models and analysis. Shape optimization for shell structures with stiffeners has been explored in [42, 43] using the FFD concept, a B-spline solid is extruded from a "master" part, which is stiffened with several "slave" stiffeners, to modify the shape of the whole shell structure.

In this work, we employ the open-source framework PENGoLINS [44] for automated IGA of non-matching shell structures using the penalty method in [24]. The shape of the shell structure is updated through a trivariate B-spline FFD block, which encompasses the entire shell structure, without differentiating the "master" and "slave" parts. The FFD block modifies the Lagrange control points of all shell patches concurrently to preserve the surface–surface intersections. Subsequently, we obtain the resulting NURBS surfaces of shells using the Lagrange extraction technique [45], which is also implemented in the IGA using FE subroutines. Moreover, this approach is also applicable to shell thickness optimization where the thickness distribution is continuous at patch intersections. By integrating these two design variables, simultaneous shape and thickness optimization for non-matching shells can be effectively achieved. This combined optimization approach enables the exploration of complex design spaces while preserving the geometric integrity of the non-matching shell structure. To demonstrate the capability of the proposed method, we apply it to the design optimization of aircraft wings, effectively navigating the unconventional design space.

The structure of this paper is outlined as follows: we introduce commonly used notations and terminologies in Sect. 2 for reference. Section 3 reviews the penalty-based formulation for coupling of non-matching isogeometric Kirchhoff–Love shells and the computational algorithm for automated IGA of non-matching shell structures. We present the FFD-based shape and thickness optimization approach and formulate the sensitivities using the Lagrange extraction technique in Sect. 4. The optimization approach is validated using a suite of benchmark problems in Sect. 5 and is applied to aircraft wing design optimization in Sect. 6, where superior design solutions are demonstrated. Lastly, we conclude the effectiveness of the proposed method and discuss potential future directions in Sect. 7.

2 Notation and terminology

In this section, we provide a summary of commonly used notions and terminologies for reference, as the formulations in the following sections can become complex due to the use of the extraction concept in IGA, interval quadrature meshes for coupling separate spline patches, and FFD B-spline blocks in optimization.

- p_{sh}: order of spline surfaces for shell structures.
- $\mathcal{V}^{\text{I,IGA}}$: IGA function space for shell patch S^{I}. Superscript I indicates shell patch index. For single-patch formulations, index I is neglected.
- $n^{\text{I,IGA}}$: number of degrees of freedom (DoFs) in $\mathcal{V}^{\text{I,IGA}}$.
- $\mathbf{N}^{\text{I,IGA}}$: spline basis functions in $\mathcal{V}^{\text{I,IGA}}$.
- $\mathbf{u}^{\text{I,IGA}}$: displacement in IGA DoFs for shell patch S^{I}.
- $\mathbf{P}^{\text{I,IGA}}$: NURBS control points for shell patch S^{I}.
- $\mathcal{V}^{\text{I,FE}}$: FE function space for shell patch S^{I}.
- $n^{\text{I,FE}}$: number of DoFs in $\mathcal{V}^{\text{I,FE}}$.
- $\mathbf{N}^{\text{I,FE}}$: Lagrange polynomial basis functions in $\mathcal{V}^{\text{I,FE}}$.
- $\mathbf{u}^{\text{I,FE}}$: displacement in FE DoFs for shell patch S^{I}.
- $\mathbf{P}^{\text{I,FE}}$: Lagrange control points for shell patch S^{I}.
- $\mathbf{M}^{\text{I}}$: extraction matrix for I-th shell patch that represents spline basis functions with Lagrange basis functions.
- ξ^{MI}: parametric coordinates of intersection $\mathcal{L}$ with respect to shell patch S^{I}.
- $\mathcal{V}^{\gamma\text{M}}$: function space of the interval quadrature mesh. Superscript $\gamma \in \{0, 1\}$ indicates the derivative order of the functions interpolated from $\mathcal{V}^{\text{I,FE}}$ to $\mathcal{V}^{\gamma\text{M}}$.

- $\mathbf{u}^{\gamma\text{MI}}$: interpolated displacement with γ-th derivative from $\mathcal{V}^{\text{I,FE}}$ to $\mathcal{V}^{\gamma\text{M}}$.
- $\mathbf{P}^{\gamma\text{MI}}$: interpolated control point functions with γ-th derivative from $\mathcal{V}^{\text{I,FE}}$ to $\mathcal{V}^{\gamma\text{M}}$.
- $\mathbf{T}^{\gamma\text{MI}}$: interpolation or transfer matrix that interpolates γ -th derivative of functions from $\mathcal{V}^{\text{I,FE}}$ to $\mathcal{V}^{\gamma\text{M}}$.
- p_{FFD}: order of B-spline basis functions for the FFD block.
- $\mathbf{N}_{\text{FFD}}$: basis functions of 3D B-spline FFD block.
- $\mathbf{P}_{\text{FFD}}$: B-spline control points of FFD block.

In Sect. 3.1, we review the basic Kirchhoff–Love shell and penalty formulations, and the IGA and FE function spaces are not differentiated in the subsection.

3 Coupling of non-matching isogeometric shells

In aerospace structural applications, many components, such as aircraft wings, empennage, and fuselage, can be modeled using shell theory. The Kirchhoff–Love shell model [16] is employed in this work with IGA discretization. Typically, aircraft CAD geometries are composed of a collection of NURBS patches, and we adopt a penalty-based coupling method [24] to perform analysis for shells composed of multiple patches with arbitrary intersections and assume St. Venant–Kirchhoff material model. We first review the basic kinematics of Kirchhoff–Love shell theory and the coupling formulation for a pair of intersecting shell patches. We then elucidate the computational procedures for the analysis of non-matching shell structures.

3.1 Penalty-based shell coupling

The Kirchhoff–Love shell model neglects transverse shear strains, with straight lines normal to the midsurface remaining straight and normal to the midsurface before and after the deformation, and shell thickness kept unchanged. Thus, the displacement field of the 3D shell can be uniquely represented by the displacement of its midsurface. In the reference configuration, the midsurface of a single-patch shell can be represented by $\mathbf{X}(\boldsymbol{\xi})$, where $\boldsymbol{\xi} = \{\xi_1, \xi_2\}$, are parametric coordinates of the midsurface. In the deformed configuration with displacement $\mathbf{u}(\boldsymbol{\xi})$, the shell midsurface is expressed as

$$\mathbf{x}(\boldsymbol{\xi}) = \mathbf{X}(\boldsymbol{\xi}) + \mathbf{u}(\boldsymbol{\xi}) \,. \tag{1}$$

Taking derivatives of midsurface with respect to parametric coordinates, we can obtain the covariant basis vectors in reference and deformed configurations on the tangent plane as

$$\mathbf{A}_\alpha = \frac{\partial \mathbf{X}}{\partial \xi_\alpha} \quad \text{and} \quad \mathbf{a}_\alpha = \frac{\partial \mathbf{x}}{\partial \xi_\alpha} \quad \text{for } \alpha \in \{1,2\} \,, \tag{2}$$

and unit vectors normal to the midsurface read as

$$\mathbf{A}_3 = \frac{\mathbf{A}_1 \times \mathbf{A}_2}{\|\mathbf{A}_1 \times \mathbf{A}_2\|} \quad \text{and} \quad \mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\|\mathbf{a}_1 \times \mathbf{a}_2\|} \,. \tag{3}$$

Membrane strain $\boldsymbol{\varepsilon}$ and change of curvature $\boldsymbol{\kappa}$ can be derived from (2) and (3). Applying the appropriate material model, we can obtain the corresponding normal forces $\mathbf{n}$ and bending moments $\mathbf{m}$. The internal virtual work of Kirchhoff–Love shell is given by

$$\delta W_{\text{int}} = \int_{\mathcal{S}} (\mathbf{n} : \delta\boldsymbol{\varepsilon} + \mathbf{m} : \delta\boldsymbol{\kappa}) \, \mathrm{d}\mathcal{S} \,, \tag{4}$$

where $\mathcal{S}$ is the midsurface of the shell. External virtual work is defined as

$$\delta W_{\text{ext}} = \int_{\Omega} \rho \mathbf{B} \cdot \delta\mathbf{u} \, \mathrm{d}\Omega + \int_{\Gamma} \mathbf{T} \cdot \delta\mathbf{u} \, \mathrm{d}\Gamma \,, \tag{5}$$

where ρ is the density, $\mathbf{B}$ is the body force, and $\mathbf{T}$ is the traction. Ω is the 3D domain of the shell volume with a thickness of t about the midsurface, and Γ is the 2D surface boundary where $\mathbf{T}$ is applied. The principle of virtual work states

$$\delta W = \delta W_{\text{int}} - \delta W_{\text{ext}} = 0 \,. \tag{6}$$

(6) is a nonlinear equation and can be solved using the Newton–Raphson method,

$$\frac{\partial^2 W}{(\partial \mathbf{u})^2} \Delta\mathbf{u} = -\frac{\partial W}{\partial \mathbf{u}} . \tag{7}$$

Here, we only go through the fundamental equations of the Kirchhoff–Love shell model, readers are referred to [17, Section 3] for detailed derivation.

Practical complex shell structures typically consist of multiple NURBS patches. For example, a pair of intersecting shells joined at a certain angle, which is prevalent when modeling the stiffeners of an aircraft wing, cannot be described by one NURBS surface. To make the CAD geometry directly available for IGA, separate NURBS patches need to be coupled together so that both displacement continuity and joint angle conservation on the surface–surface intersections are maintained during deformation.

In this project, we use the penalty-based shell coupling formulation proposed by Herrema et al. [24], which offers a good balance between accuracy and simplicity. Consider two shells modeled with spline patches $\mathcal{S}^{\text{A}}$ and $\mathcal{S}^{\text{B}}$ that share one intersection $\mathcal{L}$, as depicted in Fig. 1. The unit vector $\mathbf{a}_t^{\text{A}}$ is tangent to the intersection in the deformed configuration. In the reference and deformed configurations, conormal vectors are defined as

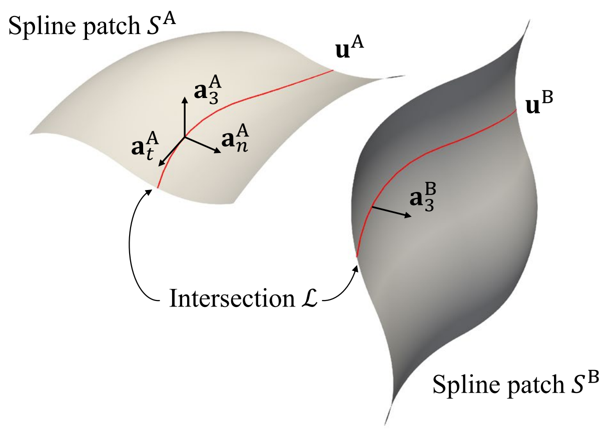

Spline patch S^{A}
$\mathbf{u}^{\mathrm{A}}$
$\mathbf{a}_3^{\mathrm{A}}$
$\mathbf{a}_t^{\mathrm{A}}$
$\mathbf{a}_n^{\mathrm{A}}$
$\mathbf{u}^{\mathrm{B}}$
$\mathbf{a}_3^{\mathrm{B}}$
Intersection $\mathcal{L}$
Spline patch S^{B}

Fig. 1 Spline patches $\mathcal{S}^{\mathrm{A}}$ and $\mathcal{S}^{\mathrm{B}}$ with one intersection $\mathcal{L}$ (indicated with red curves), where $\mathbf{u}$ and $\mathbf{a}_3$ are displacement and unit normal vector of midsurface, $\mathbf{a}_t$ and $\mathbf{a}_n$ are unit tangent vector and unit conormal vector on the intersection

$$\mathbf{A}_n^{\mathrm{A}} = \mathbf{A}_t^{\mathrm{A}} \times \mathbf{A}_3^{\mathrm{A}} \quad \text{and} \quad \mathbf{a}_n^{\mathrm{A}} = \mathbf{a}_t^{\mathrm{A}} \times \mathbf{a}_3^{\mathrm{A}} \, . \tag{8}$$

Subsequently, shell patches $\mathcal{S}^{\mathrm{A}}$ and $\mathcal{S}^{\mathrm{B}}$ are coupled through the following penalty energy

$$\begin{aligned} W_{\mathrm{pen}}^{\mathrm{AB}} = \frac{1}{2} \int_{\mathcal{L}} \alpha_{\mathrm{d}} \left\| \mathbf{u}^{\mathrm{A}} - \mathbf{u}^{\mathrm{B}} \right\|^2 + \alpha_{\mathrm{r}} \Big(\big(\mathbf{a}_3^{\mathrm{A}} \cdot \mathbf{a}_3^{\mathrm{B}} - \mathbf{A}_3^{\mathrm{A}} \cdot \mathbf{A}_3^{\mathrm{B}} \big)^2 \\ + \big(\mathbf{a}_n^{\mathrm{A}} \cdot \mathbf{a}_3^{\mathrm{B}} - \mathbf{A}_n^{\mathrm{A}} \cdot \mathbf{A}_3^{\mathrm{B}} \big)^2 \Big) \, \mathrm{d}\mathcal{L} \, , \end{aligned} \tag{9}$$

where the α_{d} term penalizes the differences of the displacement on the intersection to maintain C^0 continuity, and the α_{r} term penalizes the change of the angle between $\mathcal{S}^{\mathrm{A}}$ and $\mathcal{S}^{\mathrm{B}}$ to keep joint angle conservation. Displacement and angle conservation penalty parameters α_{d} and α_{r} are defined based on a problem-independent, dimensionless penalty coefficient α scaled with element size and material properties. For isotropic materials, the penalty parameters are given by

$$\alpha_{\mathrm{d}} = \frac{\alpha E t}{h(1-\nu^2)} \quad \text{and} \quad \alpha_{\mathrm{r}} = \frac{\alpha E t^3}{12 h (1-\nu^2)} \, , \tag{10}$$

where E is Young's modulus, ν is Poisson ratio, t is shell thickness, and h is element size. Details about the derivation of penalty parameters can be found in [24, Section 2]. The recommended value of the penalty coefficient, $\alpha = 1000$, is consistently applied across all benchmark problems and applications in this paper and yields accurate analysis results.

3.2 Computational procedures for non-matching shells

In this section, we go through the computational procedures for structural analysis of isogeometric Kirchhoff–Love shells. The stiffness matrix of non-matching shells is formulated and used as the sensitivity in the design optimization in Sect. 4.

3.2.1 IGA of Kirchhoff–Love shells using extraction

The concept of extraction [45–48] is utilized in the implementation of IGA, whose spline basis functions can be represented exactly by the linear combination of Lagrange basis functions. These Lagrange basis functions can be used in the classical FEM, allowing IGA to be implemented using finite element software with pre-defined extraction operators. An open-source IGA Python library named tIGAr is developed by Kamensky et al. [49] using the finite element software FEniCS [50]. Implementation and technical details are discussed in Sect. 4.4. In this section, we illustrate the basic mathematical operations and workflow of IGA using extraction.

To perform IGA, an extraction matrix $\mathbf{M}$ is generated to represent functions defined in spline function space $\mathcal{V}^{\mathrm{IGA}}$ with FE basis functions in $\mathcal{V}^{\mathrm{FE}}$. The relation between these two sets of basis functions is given by

$$\mathbf{N}^{\mathrm{IGA}} = \mathbf{M}^{\mathrm{T}} \mathbf{N}^{\mathrm{FE}} \, , \tag{11}$$

where $\mathbf{N}^{\mathrm{IGA}}$ are IGA basis functions and $\mathbf{N}^{\mathrm{FE}}$ are FE basis functions. Each column of $\mathbf{M}$ is the linear combination of $\mathbf{N}^{\mathrm{FE}}$ giving an IGA basis function. In the analysis, we first create an extraction matrix $\mathbf{M}$ and assemble the stiffness matrix $\mathbf{K}^{\mathrm{FE}}$ and force vector $\mathbf{F}^{\mathrm{FE}}$ in $\mathcal{V}^{\mathrm{FE}}$ using existing finite element subroutines. Then the displacement in $\mathcal{V}^{\mathrm{IGA}}$ is solved as

$$(\mathbf{M}^{\mathrm{T}} \mathbf{K}^{\mathrm{FE}} \mathbf{M}) \mathbf{u}^{\mathrm{IGA}} = \mathbf{M}^{\mathrm{T}} \mathbf{F}^{\mathrm{FE}} \, , \tag{12}$$

with problem-specific boundary conditions applied to $\mathbf{M}^{\mathrm{T}} \mathbf{K}^{\mathrm{FE}} \mathbf{M}$ and $\mathbf{M}^{\mathrm{T}} \mathbf{F}^{\mathrm{FE}}$.

Remark 1 For the purpose of clarity, we assume control points of spline surfaces have unit weights. Therefore, rational spline basis functions are the same as homogeneous spline basis functions, both denoted as $\mathbf{N}^{\mathrm{IGA}}$. In practice, weights need to be taken into consideration for correct geometric mapping and analysis.

For single-patch Kirchhoff–Love shell analysis, stiffness matrix $\mathbf{K}^{\mathrm{FE}}$ is the second derivative of total work $\frac{\partial^2 W}{(\partial \mathbf{u}^{\mathrm{FE}})^2}$. $\mathbf{M}^{\mathrm{T}} \frac{\partial^2 W}{(\partial \mathbf{u}^{\mathrm{FE}})^2} \mathbf{M}$ changes basis of $\frac{\partial^2 W}{(\partial \mathbf{u}^{\mathrm{FE}})^2}$ from $\mathcal{V}^{\mathrm{FE}}$ to $\mathcal{V}^{\mathrm{IGA}}$, and the formulation of IGA stiffness matrix can also be expressed as

$$\mathbf{M}^{\mathrm{T}}\frac{\partial^2 W}{(\partial \mathbf{u}^{\mathrm{FE}})^2}\mathbf{M} = \left(\frac{\partial \mathbf{u}^{\mathrm{FE}}}{\partial \mathbf{u}^{\mathrm{IGA}}}\right)^{\mathrm{T}}\frac{\partial^2 W}{(\partial \mathbf{u}^{\mathrm{FE}})^2}\frac{\partial \mathbf{u}^{\mathrm{FE}}}{\partial \mathbf{u}^{\mathrm{IGA}}} = \frac{\partial^2 W}{(\partial \mathbf{u}^{\mathrm{IGA}})^2}\ . \tag{13}$$

The right-hand side (RHS) of (12) for Kirchhoff–Love shell is equivalent to

$$\mathbf{M}^{\mathrm{T}}\left(-\frac{\partial W}{\partial \mathbf{u}^{\mathrm{FE}}}\right) = \left(\frac{\partial \mathbf{u}^{\mathrm{FE}}}{\partial \mathbf{u}^{\mathrm{IGA}}}\right)^{\mathrm{T}}\left(-\frac{\partial W}{\partial \mathbf{u}^{\mathrm{FE}}}\right) = -\frac{\partial W}{\partial \mathbf{u}^{\mathrm{IGA}}}\ . \tag{14}$$

Therefore, (12) is recovered as the linear system in $\mathcal{V}^{\mathrm{IGA}}$ to solve for displacements increments in IGA DoFs,

$$\frac{\partial^2 W}{(\partial \mathbf{u}^{\mathrm{IGA}})^2}\Delta \mathbf{u}^{\mathrm{IGA}} = -\frac{\partial W}{\partial \mathbf{u}^{\mathrm{IGA}}}\ . \tag{15}$$

Remark 2 While the Kirchhoff–Love shell theory neglects transverse shear strains, membrane locking can still occur in cases of curved geometry. In our current implementation, we use the standard Gauss quadrature rule to perform numerical integration without including special treatment to avoid membrane locking. We have conducted numerical studies in our earlier work [44, Section 4] to assess the accuracy and convergence of the framework for a series of benchmark problems. Solutions converge quickly to the reference solution with increased refinement, particularly evident for cubic NURBS surfaces, which are utilized in the numerical examples and applications in Sects. 5 and 6. Nevertheless, incorporating methods to alleviate potential membrane locking in isogeometric Kirchhoff–Love shells into our open-source framework will be an appealing research topic in future. We refer readers who are interested in techniques for removing membrane locking to [51–58].

3.2.2 Integration of penalty energy on patch intersections

For shell geometries that comprise multiple NURBS surfaces with non-matching intersections, the penalty energy is introduced to couple separate shell patches. In this section, we demonstrate the process to integrate the penalty energy (9) by using quadrature meshes. Again, taking two intersecting shells with one intersection as an illustrative example, the schematic configuration in parametric and physical spaces is shown in Fig. 2. We generate a geometrically 2D, topologically 1D interval mesh Ω^{M}, which is named quadrature mesh, in the parametric space to represent the intersection curve for the penalty energy integration. The parametric coordinates of the surface–surface intersection with respect to two spline patches are denoted with $\boldsymbol{\xi}^{\mathrm{MA}}$ and $\boldsymbol{\xi}^{\mathrm{MB}}$. The overall algorithm to compute $W^{\mathrm{AB}}_{\mathrm{pen}}$ on Ω^{M} are outlined as follows:

1. Define function spaces $\mathcal{V}^{0\mathrm{M}}$ and $\mathcal{V}^{1\mathrm{M}}$ on Ω^{M} with linear FE basis functions to create displacements, control point functions, and their first derivatives.
2. Move quadrature mesh Ω^{M} to parametric coordinate $\boldsymbol{\xi}^{\mathrm{MA}}$ to generate transfer matrices between function spaces of shell patch $\mathcal{S}^{\mathrm{A}}$ and quadrature mesh Ω^{M}, where the

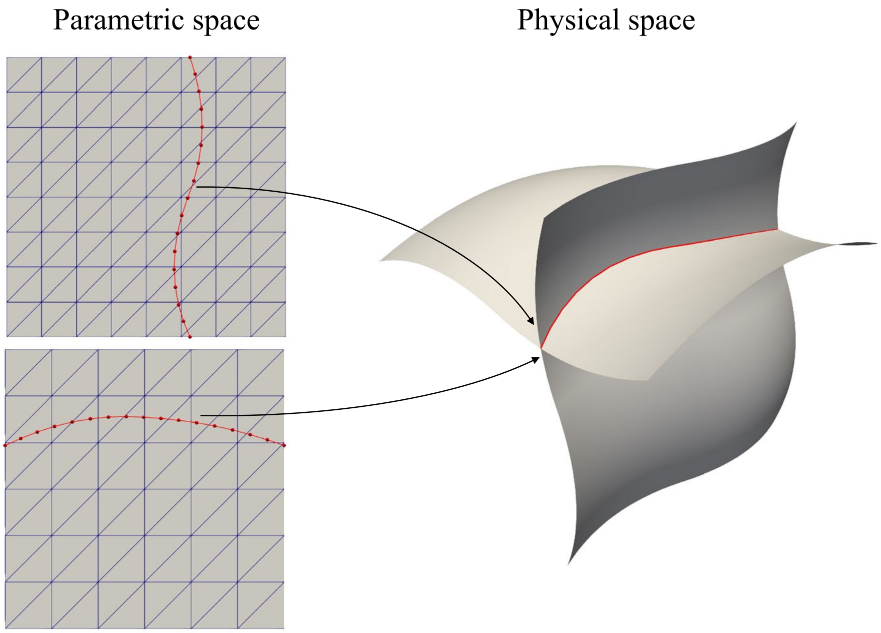


Fig. 2 A pair of shell patches share one surface–surface intersection in the physical space, associated parametric surfaces are illustrated. A topologically 1D, geometrically 2D quadrature mesh (red interval mesh) is created in the parametric space and moved to parametric locations of the intersection with respect to two shells to create transfer matrices. Geometric and displacement quantities of shell patches are interpolated to the quadrature mesh, where the penalty energy is integrated

configuration is shown in the upper left part of Fig. 2. The following transfer matrices are created:

(a) $\mathbf{T}^{0\mathrm{MA}} : \mathcal{V}^{\mathrm{A,FE}} \rightarrow \mathcal{V}^{0\mathrm{M}}$, which interpolates functions from $\mathcal{V}^{\mathrm{A,FE}}$ to $\mathcal{V}^{0\mathrm{M}}$. Each entry is defined as

$$\mathbf{T}^{0\mathrm{MA}}_{ij} = \mathbf{N}^{\mathrm{A,FE}}_{i}(\xi^{\mathrm{MA}}_{j}) \,, \tag{16}$$

where $\mathbf{N}^{\mathrm{A,FE}}_{i}$ is the ith basis function in $\mathcal{V}^{\mathrm{A,FE}}$, and ξ^{MA}_{j} is the jth coordinate of Ω^{M}.

(b) $\mathbf{T}^{1\mathrm{MA}} : \mathcal{V}^{\mathrm{A,FE}} \rightarrow \mathcal{V}^{1\mathrm{M}}$, which interpolates the first derivative of functions from $\mathcal{V}^{\mathrm{A,FE}}$ to $\mathcal{V}^{1\mathrm{M}}$. $\mathbf{T}^{1\mathrm{MA}}$ is defined as

$$\mathbf{T}^{1\mathrm{MA}}_{ij} = \mathbf{N}^{\mathrm{A,FE}}_{i,\xi}(\xi^{\mathrm{MA}}_{j}) \,, \tag{17}$$

where $\mathbf{N}^{\mathrm{A,FE}}_{i,\xi}$ is the parametric derivative of ith basis function in $\mathcal{V}^{\mathrm{A,FE}}$.

3. Compute displacement $\mathbf{u}^{0\mathrm{MA}}$, geometric mapping $\mathbf{P}^{0\mathrm{MA}}$, and their first derivatives $\mathbf{u}^{1\mathrm{MA}}, \mathbf{P}^{1\mathrm{MA}}$ from parametric domain of $\mathcal{S}^{\mathrm{A}}$ to Ω^{M} using transfer matrices generated from last step,

$$\begin{aligned} \mathbf{u}^{\gamma\mathrm{MA}} &= \mathbf{T}^{\gamma\mathrm{MA}}\mathbf{u}^{\mathrm{A,FE}} && \text{and} \\ \mathbf{P}^{\gamma\mathrm{MA}} &= \mathbf{T}^{\gamma\mathrm{MA}}\mathbf{P}^{\mathrm{A,FE}} && \text{for } \gamma \in \{0,1\} \,. \end{aligned} \tag{18}$$

4. Substitute interpolated quantities from previous step into (1)–(3) to obtain basis vectors on Ω^{M} before and after deformation: $\mathbf{A}^{\mathrm{MA}}_{i}$ and $\mathbf{a}^{\mathrm{MA}}_{i}$ for $i \in \{1,2,3\}$.
5. Use (8) to compute conormal vectors: $\mathbf{A}^{\mathrm{MA}}_{n}$ and $\mathbf{a}^{\mathrm{MA}}_{n}$.
6. Compute physical element size h^{A}_{X} of $\mathcal{S}^{\mathrm{A}}$, then interpolate it to Ω^{M} to obtain h^{MA}_{X}.
7. Move quadrature mesh Ω^{M} to parametric coordinate ξ^{MB}, which is depicted in the lower left configuration in Fig. 2. Then create transfer matrices $\mathbf{T}^{0\mathrm{MB}}$ and $\mathbf{T}^{1\mathrm{MB}}$.
8. Repeat steps 3, 4 and 6 to obtain corresponding displacement $\mathbf{u}^{0\mathrm{MB}}$, normal vectors $\mathbf{A}^{\mathrm{BM}}_{3}$ and $\mathbf{a}^{\mathrm{BM}}_{3}$, and element size h^{MB}_{X} from $\mathcal{S}^{\mathrm{B}}$.
9. Calculate penalty parameters based on (10) using averaged physical element size $h^{\mathrm{M}}_{X} = \frac{1}{2}(h^{\mathrm{MA}}_{X} + h^{\mathrm{MB}}_{X})$ for each node on Ω^{M},

$$\alpha^{\mathrm{M}}_{\mathrm{d}} = \frac{\alpha E t^{\mathrm{M}}}{h^{\mathrm{M}}_{X}(1-\nu^2)} \quad \text{and} \quad \alpha^{\mathrm{M}}_{\mathrm{r}} = \frac{\alpha E (t^{\mathrm{M}})^3}{12 h^{\mathrm{M}}_{X}(1-\nu^2)} \,. \tag{19}$$

Remark 3 In this section, we assume $\mathcal{S}^{\mathrm{A}}$ and $\mathcal{S}^{\mathrm{B}}$ have the same thickness $t^{\mathrm{M}} = t^{\mathrm{A}} = t^{\mathrm{B}}$ in penalty parameters, same for Young's modulus and Poisson ratio. For the thickness optimization problem discussed in Sect. 4.3, shell patches may have different thicknesses, or each shell patch has variable thickness distribution. In these scenarios, we use an identical method to the element size calculation to interpolate thickness from shell patches to quadrature mesh and take the average, i.e., $t^{M} = \frac{1}{2}(t^{\mathrm{MA}} + t^{\mathrm{MB}})$.

10. Based on (9), we can integrate the penalty energy on Ω^{M} with quantities computed from previous steps,

$$\begin{aligned} W^{\mathrm{AB}}_{\mathrm{pen}} = \frac{1}{2}\int_{\Omega^{\mathrm{M}}} \Bigg(& \alpha^{\mathrm{M}}_{\mathrm{d}} \|\mathbf{u}^{0\mathrm{MA}} - \mathbf{u}^{0\mathrm{MB}}\|^2 \\ & + \alpha^{\mathrm{M}}_{\mathrm{r}} \Big((\mathbf{a}^{\mathrm{MA}}_{3} \cdot \mathbf{a}^{\mathrm{MB}}_{3} - \mathbf{A}^{\mathrm{MA}}_{3} \cdot \mathbf{A}^{\mathrm{MB}}_{3})^2 \\ & + (\mathbf{a}^{\mathrm{MA}}_{n} \cdot \mathbf{a}^{\mathrm{MB}}_{3} - \mathbf{A}^{\mathrm{MA}}_{n} \cdot \mathbf{A}^{\mathrm{MB}}_{3})^2 \Big) \Bigg) J^{\mathrm{M}} \, \mathrm{d}\Omega \,, \end{aligned} \tag{20}$$

where J^{M} is the associated line Jacobian mapping Ω^{M} to $\mathcal{L}$.

3.2.3 Assembly of non-matching system

With the coupled formulations for intersecting shells, we can perform IGA on the non-matching shells directly. For a pair of shells with one intersection illustrated in Fig. 2, the total virtual energy in equilibrium is given by

$$\delta W_{\mathrm{t}} = \delta W^{\mathrm{A}} + \delta W^{\mathrm{B}} + \delta W^{\mathrm{AB}}_{\mathrm{pen}} = 0 \,, \tag{21}$$

which can be solved by the Newton–Raphson method. We can solve the linearized problem of (21) to obtain the full displacement $\mathbf{u}^{\mathrm{IGA}} = \left[\mathbf{u}^{\mathrm{A,IGA}} \;\; \mathbf{u}^{\mathrm{B,IGA}}\right]^{\mathrm{T}}$ with one iteration,

$$\begin{bmatrix} \frac{\partial^2 W_{\mathrm{t}}}{(\partial \mathbf{u}^{\mathrm{A,IGA}})^2} & \frac{\partial^2 W_{\mathrm{t}}}{\partial \mathbf{u}^{\mathrm{A,IGA}} \partial \mathbf{u}^{\mathrm{B,IGA}}} \\ \frac{\partial^2 W_{\mathrm{t}}}{\partial \mathbf{u}^{\mathrm{B,IGA}} \partial \mathbf{u}^{\mathrm{A,IGA}}} & \frac{\partial^2 W_{\mathrm{t}}}{(\partial \mathbf{u}^{\mathrm{B,IGA}})^2} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{\mathrm{A,IGA}} \\ \mathbf{u}^{\mathrm{B,IGA}} \end{bmatrix} = - \begin{bmatrix} \frac{\partial W_{\mathrm{t}}}{\partial \mathbf{u}^{\mathrm{A,IGA}}} \\ \frac{\partial W_{\mathrm{t}}}{\partial \mathbf{u}^{\mathrm{B,IGA}}} \end{bmatrix} \,. \tag{22}$$

Formulations for the derivative of W_{t} can be obtained by means of the chain rule. Using $\mathcal{S}^{\mathrm{A}}$ for demonstration, the derivative of W_{t} with respect to $\mathbf{u}^{\mathrm{A,IGA}}$ is given by

$$\frac{\partial W_{\mathrm{t}}}{\partial \mathbf{u}^{\mathrm{A,IGA}}} = (\mathbf{M}^{\mathrm{A}})^{\mathrm{T}} \left(\frac{\partial W^{\mathrm{A}}}{\partial \mathbf{u}^{\mathrm{A,FE}}} + \sum_{\gamma=0}^{1} (\mathbf{T}^{\gamma\mathrm{MA}})^{\mathrm{T}} \frac{\partial W^{\mathrm{AB}}_{\mathrm{pen}}}{\partial \mathbf{u}^{\gamma\mathrm{MA}}} \right) \,. \tag{23}$$

The second derivatives of W in the left-hand side (LHS) of (22) can be computed with the following formulations. Taking $\mathcal{S}^{\mathrm{A}}$ for illustration, the diagonal block reads as

$$\frac{\partial^2 W_{\mathrm{t}}}{(\partial \mathbf{u}^{\mathrm{A,IGA}})^2} = (\mathbf{M}^{\mathrm{A}})^{\mathrm{T}} \left(\frac{\partial^2 W^{\mathrm{A}}}{(\partial \mathbf{u}^{\mathrm{A,FE}})^2} + \sum_{\delta=0}^{1} \sum_{\gamma=0}^{1} (\mathbf{T}^{\gamma\mathrm{MA}})^{\mathrm{T}} \frac{\partial^2 W^{\mathrm{AB}}_{\mathrm{pen}}}{\partial \mathbf{u}^{\gamma\mathrm{MA}} \partial \mathbf{u}^{\delta\mathrm{MA}}} \mathbf{T}^{\delta\mathrm{MA}} \right) \mathbf{M}^{\mathrm{A}} \,, \tag{24}$$

and the off-diagonal block can be expressed as

$$\frac{\partial^2 W_t}{\partial \mathbf{u}^{A,IGA} \partial \mathbf{u}^{B,IGA}} = (\mathbf{M}^A)^T \left(\sum_{\delta=0}^{1} \sum_{\gamma=0}^{1} (\mathbf{T}^{\gamma MA})^T \frac{\partial^2 W_{pen}^{AB}}{\partial \mathbf{u}^{\gamma MA} \partial \mathbf{u}^{\delta MB}} \mathbf{T}^{\delta MB} \right) \mathbf{M}^B . \tag{25}$$

In (23)–(25), expressions of derivatives of W^A can be found in [16, Section 3], and [24, Section 2.2] spells out derivatives of W_{pen}^{AB}. These derivatives are computed automatically by the implementation discussed in Sect. 4.4 using computer algebra and code generation capabilities in FEniCS. Substituting (23)–(25) into (22), one can solve the displacements in IGA DoFs for $\mathcal{S}^A$ and $\mathcal{S}^B$.

Remark 4 For clarity, we only demonstrate the non-matching system with two shell patches. However, (22) can be readily extended to shell geometries consisting of more than two surfaces. We refer interested readers to [44, Section 3.2] for the assembly of shell structures with m patches and arbitrary intersections.

4 FFD-based design optimization

CAD geometries of Kirchhoff–Love shells can be used for analysis without finite element mesh generation by employing formulations from Sect. 3. This provides attractive features for shape optimization of shell structures, where the discretization of shell patches stays unaltered during shape evolution. The updated geometry in optimization iterations stays consistent with the analysis model. As a result, this approach minimizes the effort required for geometry processing while simultaneously enhancing accuracy.

In the context of shape optimization for non-matching shell structures, it is crucial to ensure that updated shell patches remain properly connected. Failure to maintain this connectivity can result in separation or self-penetration of shell patches during the optimization iteration. Such issues would lead to false analysis and yield unrealistic optimal shapes. To tackle this challenge, we adopt the FFD-based [2] technique combined with Lagrange extraction to update shell geometry and demonstrate the workflow in Sect. 4.1. A comparable concept can be applied to thickness optimization to ensure continuous thickness distribution at the surface–surface intersections if needed. Sensitivities for both shape and thickness optimization are given in the subsequent sections.

4.1 Non-matching shells update through FFD block

We use a cylindrical roof consisting of four non-matching shell patches that are shown in the upper-left part of Fig. 3 as an example to demonstrate the FFD-based shape optimization approach. Note that this approach can be applied to shell structures with an arbitrary number of patches.

4.1.1 Approximating shell patches with extraction

For the initial CAD geometry consisting of m Kirchhoff–Love shell patches, define a set of NURBS surface functions $\{\mathbf{S}^A(\xi), \mathbf{S}^B(\xi), \dots, \mathbf{S}^{I^m}(\xi)\}$, and the I-th shell patch $\mathbf{S}^I(\xi)$ is expressed as

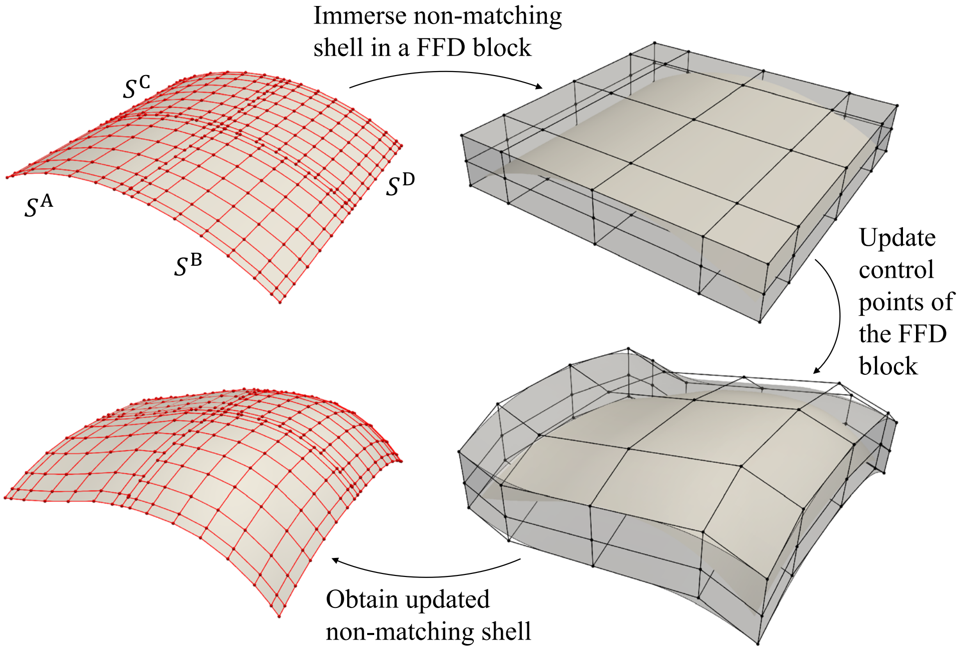


Fig. 3 Workflow of FFD-based shape optimization for non-matching shell structures. A cylindrical roof consisting of four non-matching NURBS patches is first immersed in a trivariate B-spline block. We update the control points of the FFD block to deform the shape of the non-matching cylindrical roof. Control nets of the NURBS surfaces are indicated with red color, and black is used for the control net of the FFD block

$$\mathbf{S}^{\mathrm{I}}(\xi) = \mathbf{N}^{\mathrm{I,IGA}}(\xi)\mathbf{P}^{\mathrm{I,IGA}} \,, \tag{26}$$

where $\mathbf{N}^{\mathrm{I,IGA}}(\xi)$ are the spline basis functions of degree p_{sh} in $\mathcal{V}^{\mathrm{I,IGA}}$. We omit degree p_{sh} in the notation for clarity. $\mathbf{P}^{\mathrm{I,IGA}}$ are the NURBS control points for surface function $\mathbf{S}^{\mathrm{I}}$.

Using the extraction concept discussed in Sect. 3.2.1, NURBS surface function $\mathbf{S}^{\mathrm{I}}(\xi)$ can be represented with Lagrange polynomials as well,

$$\mathbf{S}^{\mathrm{I}}(\xi) = \mathbf{N}^{\mathrm{I,FE}}(\xi)\mathbf{P}^{\mathrm{I,FE}} \,, \tag{27}$$

where $\mathbf{N}^{\mathrm{I,FE}}(\xi)$ are basis functions in the finite element function space $\mathcal{V}_{\mathrm{s}}^{\mathrm{I,FE}}$ with nodal interpolatory property, and $\mathbf{P}^{\mathrm{I,FE}}$ are Lagrange control points, or nodal values of $\mathbf{S}^{\mathrm{I}}$. Plugging nodal coordinate $\xi^{\mathrm{I,FE}}$ of $\mathcal{V}_{\mathrm{s}}^{\mathrm{I,FE}}$ into (27), coordinate of the NURBS surface $\mathbf{S}^{\mathrm{I}}$ is represented with nodal values in the discrete setting,

$$\mathbf{S}^{\mathrm{I}}(\xi^{\mathrm{I,FE}}) = \mathbf{N}^{\mathrm{I,FE}}(\xi^{\mathrm{I,FE}})\mathbf{P}^{\mathrm{I,FE}} = \mathbf{P}^{\mathrm{I,FE}} \,. \tag{28}$$

Based on (11), Lagrange control points can be obtained through the extraction matrix and NURBS control points. We have the following relation,

$$\mathbf{S}^{\mathrm{I}}(\xi^{\mathrm{I,FE}}) = \mathbf{P}^{\mathrm{I,FE}} = \mathbf{M}^{\mathrm{I}}\mathbf{P}^{\mathrm{I,IGA}} \,. \tag{29}$$

4.1.2 Relating FFD block control points and shell geometry

The first step of Fig. 3 illustrates the initial configuration of a collection of intersecting non-matching shell patches S, where red control nets are displayed. To enforce connectivity of the intersections during optimization, we immerse S in a trivariate B-spline block, which is referred to as an FFD block, and use control points of the FFD block as design variables. A schematic demonstration is shown in the second step of Fig. 3. The FFD B-spline block is defined as

$$\mathbf{V}(\boldsymbol{\theta}) = \mathbf{N}_{\mathrm{FFD}}(\boldsymbol{\theta})\mathbf{P}_{\mathrm{FFD}} \,, \tag{30}$$

where $\boldsymbol{\theta}$ is the parametric coordinate of the FFD block, $\mathbf{N}_{\mathrm{FFD}}(\boldsymbol{\theta})$ are B-spline solid basis functions of degree p_{FFD} with knots vector, and $\mathbf{P}_{\mathrm{FFD}}$ are B-spline block control points.

To simplify formulation and implementation, we use an identity mapping for the FFD block B-spline block, so that the parametric coordinate coincides with the physical coordinate,

$$\mathbf{V}(\boldsymbol{\theta}) = \mathbf{N}_{\mathrm{FFD}}(\boldsymbol{\theta})\mathbf{P}_{\mathrm{FFD}} = \boldsymbol{\theta} \,. \tag{31}$$

Substituting (29) into (31), NURBS surfaces of the non-matching shells can be expressed using the FFD block basis functions and control points,

$$\mathbf{V}\big(\mathbf{S}^{\mathrm{I}}(\xi)\big) = \mathbf{N}_{\mathrm{FFD}}\big(\mathbf{S}^{\mathrm{I}}(\xi)\big)\mathbf{P}_{\mathrm{FFD}} = \mathbf{S}^{\mathrm{I}}(\xi) \,. \tag{32}$$

In the continuous context of (32), shell patches will not separate in the final configuration as long as they are interconnected in the initial geometry. As the shape update of the FFD block is continuous, there is no relative movement between patches within the FFD block. In the discrete space, we can relate the NURBS control points of the shell patches to the control points of the FFD block,

$$\mathbf{N}_{\mathrm{FFD}}\big(\mathbf{S}^{\mathrm{I}}(\xi^{\mathrm{I,FE}})\big)\mathbf{P}_{\mathrm{FFD}} = \mathbf{N}_{\mathrm{FFD}}(\mathbf{P}^{\mathrm{I,FE}})\mathbf{P}_{\mathrm{FFD}} = \mathbf{M}^{\mathrm{I}}\mathbf{P}^{\mathrm{I,IGA}} \,. \tag{33}$$

The control points of the NURBS surface $\mathbf{P}^{\mathrm{I,IGA}}$ of shell patches can be updated through the control points of the FFD block $\mathbf{P}_{\mathrm{FFD}}$. Let $\mathbf{N}_{\mathrm{FFD}}(\mathring{\mathbf{P}}^{\mathrm{I,FE}}) := \mathbf{A}_{\mathrm{FFD}}^{\mathrm{I}}$, where $\mathring{\mathbf{P}}^{\mathrm{I,FE}}$ denotes Lagrange control points of spline patch I in the baseline configuration. Then $\mathbf{P}^{\mathrm{I,IGA}}$ can be computed as

$$\mathbf{P}^{\mathrm{I,IGA}} = \Big(\big((\mathbf{M}^{\mathrm{I}})^{\mathrm{T}}\mathbf{M}^{\mathrm{I}}\big)^{-1}(\mathbf{M}^{\mathrm{I}})^{\mathrm{T}}\mathbf{A}_{\mathrm{FFD}}^{\mathrm{I}}\Big)\mathbf{P}_{\mathrm{FFD}} \,. \tag{34}$$

It is noted that we need to solve the system using Moore–Penrose pseudo inverse due to the non-square nature of the extraction matrix $\mathbf{M}^{\mathrm{I}}$, which has dimensions of $n^{\mathrm{I,FE}} \times n^{\mathrm{I,IGA}}$. For the extraction matrix, we have $n^{\mathrm{I,FE}} > n^{\mathrm{I,IGA}}$, which means that we are solving an overdetermined system. Therefore, $\mathbf{P}^{\mathrm{I,IGA}}$ is considered as a least square fit in (34) rather than an exact solution. The shape

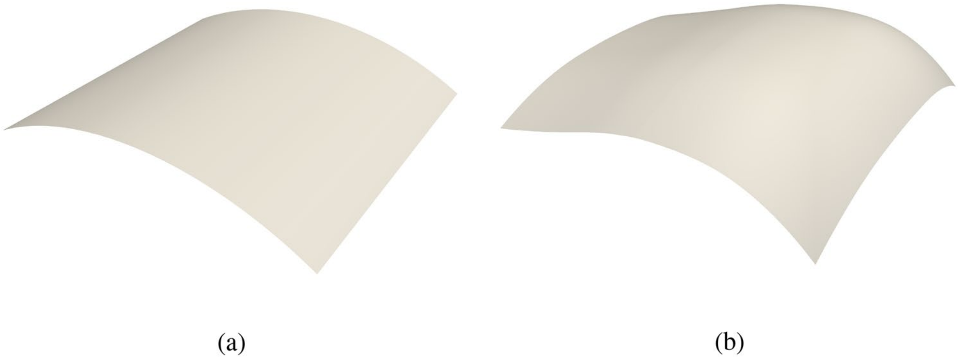


Fig. 4 **a** Initial configuration of the cylindrical roof geometry consisting of four non-matching NURBS patches. **b** Updated NURBS surfaces using FFD block

update strategy using FFD block is illustrated in the third step in Fig. 3, and the resulting shell patches with control net are depicted in the fourth step. A comparison between the initial non-matching cylindrical roof and updated NURBS surfaces is shown in Fig. 4, where the surface–surface intersections keep overlapping within tolerance in the updated configuration.

The procedures to update control points of non-matching shells with m patches are summarized as follows:

1. In the preprocessing step, generate sparse matrices of evaluation of FFD block B-spline basis functions at shells' Lagrange control points in the initial configuration $\{\mathbf{A}^{\mathrm{I}}_{\mathrm{FFD}}\}$ and Lagrange extraction matrices $\{\mathbf{M}^{\mathrm{I}}\}$, for $\mathrm{I} \in \{\mathrm{A}, \mathrm{B}, \dots, \mathrm{I}^m\}$.
2. At optimization iteration step i^{opt}, obtain updated control points of the FFD block $\left(\mathbf{P}_{\mathrm{FFD}}\right)^{i^{\mathrm{opt}}}$. Compute updated Lagrange control points $\left(\mathbf{P}^{\mathrm{I,FE}}\right)^{i^{\mathrm{opt}}}$ for all shell patches,
$$\mathbf{A}^{\mathrm{I}}_{\mathrm{FFD}}\left(\mathbf{P}_{\mathrm{FFD}}\right)^{i^{\mathrm{opt}}} = \left(\mathbf{P}^{\mathrm{I,FE}}\right)^{i^{\mathrm{opt}}} . \tag{35}$$
3. Solve NURBS control points $\left(\mathbf{P}^{\mathrm{I,IGA}}\right)^{i^{\mathrm{opt}}}$ at step i^{opt} through Moore–Penrose pseudo inverse,
$$\mathbf{M}^{\mathrm{I}}\left(\mathbf{P}^{\mathrm{I,IGA}}\right)^{i^{\mathrm{opt}}} = \left(\mathbf{P}^{\mathrm{I,FE}}\right)^{i^{\mathrm{opt}}} . \tag{36}$$
4. Perform IGA with updated shell geometry, evaluate objective function and derivatives if needed, then proceed with optimization iteration.

Though the control points of the shell patches are computed in the least square fit sense, the updated geometry can still retain the intersection with sufficient discretization. A sliced view of the intersection, in accompaniment with NURBS and Lagrange control points, between the right two spline patches $\mathcal{S}^{\mathrm{C}}$ and $\mathcal{S}^{\mathrm{D}}$ in the first step of Fig. 3 is shown in Fig. 5, where we use coarser discretizations to make the comparison clearer. In the updated configuration, the two cubic intersecting edges are still overlapping even with only 5 and 6 NURBS control points.

Remark 5 Since there is no relative movement between intersecting spline patches within the FFD block, which can be achieved with adequate control points in the discrete context, parametric coordinates of surface–surface intersections remain unchanged during shape updates. Therefore, transfer matrices introduced in Sect. 3.2.2 can be reused to interpolate data from spline patches to quadrature meshes when integrating penalty energies in the optimization iteration. These matrices only need to be generated once at the preprocessing stage.

4.2 Sensitivities for shape optimization

Utilizing the capabilities of direct analysis for non-matching isogeometric shells and incorporating FFD-based shape updates, we are able to conduct shape optimization for the shell structures in a seamless manner. The problem that optimizes the shape of non-matching shells can be formulated as follows,

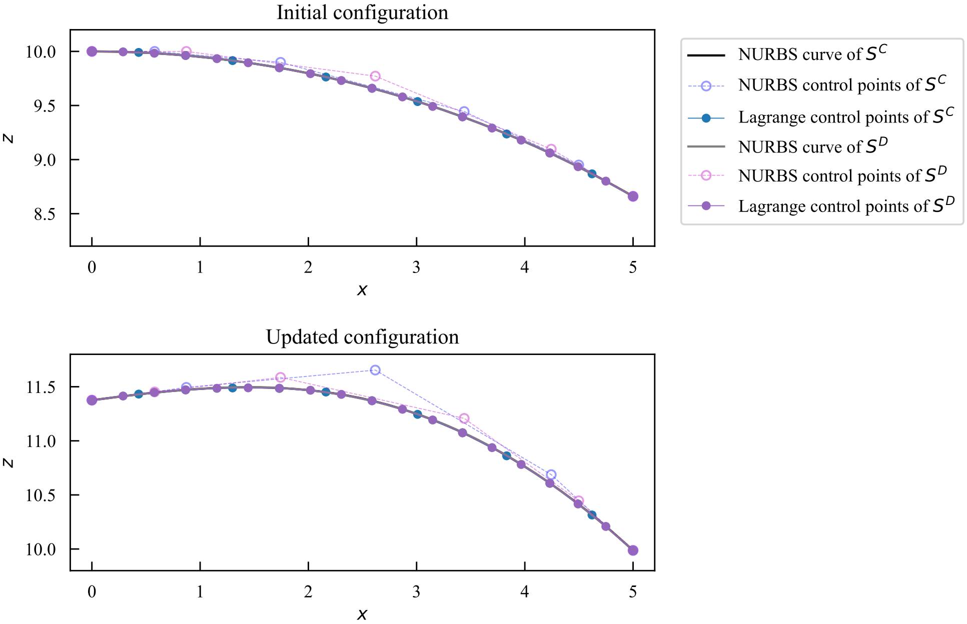


Fig. 5 Sliced view of the intersecting edges between shell patches $\mathcal{S}^{\mathrm{C}}$ and $\mathcal{S}^{\mathrm{D}}$ of the cylindrical roof. The two edges remain overlapping in the updated configuration

$$\begin{aligned} \text{minimize} \quad & f^{\text{obj}}\left(\mathbf{P}_{\text{FFD}}\right) \\ \text{subject to} \quad & g^{\text{con}}_{i_g}\left(\mathbf{P}_{\text{FFD}}\right) \leq 0 \text{ , for } i_g \in \{1, 2, \dots, n_g\} \\ & h^{\text{con}}_{i_h}\left(\mathbf{P}_{\text{FFD}}\right) = 0 \text{ , for } i_h \in \{1, 2, \dots, n_h\} \\ & \mathbf{P}_{\text{FFD}l} \leq \mathbf{P}_{\text{FFD}} \leq \mathbf{P}_{\text{FFD}u} \text{ ,} \end{aligned} \tag{37}$$

where control points of the FFD block $\mathbf{P}_{\text{FFD}}$ are design variables, f^{obj} is the objective function, $g^{\text{con}}_{i_g}$ are inequality constraints, and $h^{\text{con}}_{i_h}$ are equality constraints. $\mathbf{P}_{\text{FFD}l}$ and $\mathbf{P}_{\text{FFD}u}$ are lower and upper limits for the design variables.

To perform gradient-based design optimization, we formulate total derivatives of the objective function with respect to design variables as

$$\frac{\mathrm{d}f^{\text{obj}}}{\mathrm{d}\mathbf{P}_{\text{FFD}}} = \left(\frac{\partial f^{\text{obj}}}{\partial \mathbf{P}^{\text{IGA}}} + \frac{\partial f^{\text{obj}}}{\partial \mathbf{u}^{\text{IGA}}} \frac{\mathrm{d}\mathbf{u}^{\text{IGA}}}{\mathrm{d}\mathbf{P}^{\text{IGA}}} \right) \frac{\mathrm{d}\mathbf{P}^{\text{IGA}}}{\mathrm{d}\mathbf{P}_{\text{FFD}}} \text{ ,} \tag{38}$$

where $\mathbf{P}^{\text{IGA}} = \left[\mathbf{P}^{\text{A,IGA}} \ \mathbf{P}^{\text{B,IGA}} \ \dots \ \mathbf{P}^{\text{I}^m\text{,IGA}}\right]^{\text{T}}$ is the full vector of NURBS control points, and similarly, $\mathbf{u}^{\text{IGA}} = \left[\mathbf{u}^{\text{A,IGA}} \ \mathbf{u}^{\text{B,IGA}} \ \dots \ \mathbf{u}^{\text{I}^m\text{,IGA}}\right]^{\text{T}}$ is the full vector of shell displacements in IGA DoFs.

Partial derivatives $\frac{\partial f^{\text{obj}}}{\partial \mathbf{P}^{\text{IGA}}}$ and $\frac{\partial f^{\text{obj}}}{\partial \mathbf{u}^{\text{IGA}}}$ in (38) can be computed and depend on the objective function combined with extraction matrices,

$$\frac{\partial f^{\text{obj}}}{\partial \mathbf{P}^{\text{IGA}}} = \mathbf{M}^{\text{T}} \frac{\partial f^{\text{obj}}}{\partial \mathbf{P}^{\text{FE}}} \quad \text{and} \quad \frac{\partial f^{\text{obj}}}{\partial \mathbf{u}^{\text{IGA}}} = \mathbf{M}^{\text{T}} \frac{\partial f^{\text{obj}}}{\partial \mathbf{u}^{\text{FE}}} \text{ ,} \tag{39}$$

where $\mathbf{P}^{\text{FE}} = \left[\mathbf{P}^{\text{A,FE}} \ \mathbf{P}^{\text{B,FE}} \ \dots \ \mathbf{P}^{\text{I}^m\text{,FE}}\right]^{\text{T}}$, $\mathbf{u}^{\text{FE}} = \left[\mathbf{u}^{\text{A,FE}} \ \mathbf{u}^{\text{B,FE}} \ \dots \ \mathbf{u}^{\text{I}^m\text{,FE}}\right]^{\text{T}}$. $\mathbf{M} = \text{diag}(\mathbf{M}^{\text{A}}, \mathbf{M}^{\text{B}}, \dots, \mathbf{M}^{\text{I}^m})$ is a diagonal block matrix for global extraction. Calculation of partial derivatives in (39) is automated using FEniCS. Formulation for total derivative $\frac{\mathrm{d}\mathbf{P}^{\text{IGA}}}{\mathrm{d}\mathbf{P}_{\text{FFD}}}$ is introduced in (34). As for total derivative $\frac{\mathrm{d}\mathbf{u}^{\text{IGA}}}{\mathrm{d}\mathbf{P}^{\text{IGA}}}$, we have the implicit relation between $\mathbf{P}^{\text{IGA}}$ and $\mathbf{u}^{\text{IGA}}$,

$$\mathbf{r}_{\text{t}} = \mathbf{R}_{\text{t}}(\mathbf{P}^{\text{IGA}}, \mathbf{u}^{\text{IGA}}) = \frac{\partial W_{\text{t}}(\mathbf{u}^{\text{IGA}}, \mathbf{P}^{\text{IGA}})}{\partial \mathbf{u}^{\text{IGA}}} = \mathbf{0} \text{ ,} \tag{40}$$

where W_{t} is the total energy of the non-matching shells defined in (21). Once an updated $\mathbf{P}^{\text{IGA}}$ is obtained, the shell displacements need to be solved using (21) until the residual vector $\mathbf{r}_{\text{t}}$ reaches a tolerance. Thus, $\mathbf{r}_{\text{t}}$ is supposed to remain as $\mathbf{0}$ despite the change of $\mathbf{P}^{\text{IGA}}$, and we have the following derivative

$$\frac{\mathrm{d}\mathbf{r}_{\text{t}}}{\mathrm{d}\mathbf{P}^{\text{IGA}}} = \frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{P}^{\text{IGA}}} + \frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{u}^{\text{IGA}}} \frac{\mathrm{d}\mathbf{u}^{\text{IGA}}}{\mathrm{d}\mathbf{P}^{\text{IGA}}} = \mathbf{0} \text{ ,} \tag{41}$$

and the total derivative $\frac{\mathrm{d}\mathbf{u}^{\text{IGA}}}{\mathrm{d}\mathbf{P}^{\text{IGA}}}$ in (38) is given by

$$\frac{\mathrm{d}\mathbf{u}^{\text{IGA}}}{\mathrm{d}\mathbf{P}^{\text{IGA}}} = -\left(\frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{u}^{\text{IGA}}} \right)^{-1} \frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{P}^{\text{IGA}}} \text{ .} \tag{42}$$

Partial derivative $\frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{u}^{\text{IGA}}}$ is equivalent to $\frac{\partial^2 W_{\text{t}}}{\partial (\mathbf{u}^{\text{IGA}})^2}$ and is the stiffness matrix defined in (22). Analogously, we use a pair of shell patches to illustrate the formulation of partial derivative $\frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{P}^{\text{IGA}}}$,

$$\frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{P}^{\text{IGA}}} = \begin{bmatrix} \frac{\partial^2 W_{\text{t}}}{\partial \mathbf{u}^{\text{A,IGA}} \partial \mathbf{P}^{\text{A,IGA}}} & \frac{\partial^2 W_{\text{t}}}{\partial \mathbf{u}^{\text{A,IGA}} \partial \mathbf{P}^{\text{B,IGA}}} \\ \frac{\partial^2 W_{\text{t}}}{\partial \mathbf{u}^{\text{B,IGA}} \partial \mathbf{P}^{\text{A,IGA}}} & \frac{\partial^2 W_{\text{t}}}{\partial \mathbf{u}^{\text{B,IGA}} \partial \mathbf{P}^{\text{B,IGA}}} \end{bmatrix} \text{ .} \tag{43}$$

Partial derivatives in (43) have identical expressions to (24) and (25).

Extend partial derivatives in (43) to shell structures with an arbitrary number of patches, and substitute $\frac{\mathrm{d}\mathbf{u}^{\text{IGA}}}{\mathrm{d}\mathbf{P}^{\text{IGA}}}$ in (38) with (42), we can obtain the total derivative of the shape optimization

$$\frac{\mathrm{d}f^{\text{obj}}}{\mathrm{d}\mathbf{P}_{\text{FFD}}} = \left(\frac{\partial f^{\text{obj}}}{\partial \mathbf{P}^{\text{IGA}}} - \frac{\partial f^{\text{obj}}}{\partial \mathbf{u}^{\text{IGA}}} \left(\frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{u}^{\text{IGA}}} \right)^{-1} \frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{P}^{\text{IGA}}} \right) \frac{\mathrm{d}\mathbf{P}^{\text{IGA}}}{\mathrm{d}\mathbf{P}_{\text{FFD}}} \text{ .} \tag{44}$$

4.3 Sensitivities for thickness optimization

The idea of FFD-based shape update can be applied to shell thickness optimization, where the shell thickness is treated as an extra field of the NURBS control points. We can use (34) to build the relation of the thickness between shell patches and FFD block,

$$\mathbf{t}^{\text{I,IGA}} = \left(\left((\mathbf{M}^{\text{I}}_{\text{s}})^{\text{T}} \mathbf{M}^{\text{I}}_{\text{s}} \right)^{-1} (\mathbf{M}^{\text{I}}_{\text{s}})^{\text{T}} \mathbf{A}^{\text{I}}_{\text{FFDs}} \right) \mathbf{t}_{\text{FFD}} \text{ ,} \tag{45}$$

where $\mathbf{t}^{\text{I,IGA}}$ is the thickness for shell S^{I} in IGA DoFs, and $\mathbf{t}_{\text{FFD}}$ is the corresponding thickness field of the FFD block. Subscript s in $\mathbf{M}^{\text{I}}_{\text{s}}$ and $\mathbf{A}^{\text{I}}_{\text{FFDs}}$ denotes matrices for scalar fields. Note that $\mathbf{t}_{\text{FFD}}$ is not the actual thickness of the B-spline solid geometry, but an extra set of the control points on the FFD block to update the thickness of the non-matching shells. Accordingly, the identical shape update strategy Sect. 4.1.2 is applicable to thickness update. FFD-based thickness optimization also offers the benefit that shell thickness remains continuous on the surface–surface intersections.

Replacing control points of the FFD block in (37) with $\mathbf{t}_{\text{FFD}}$, one can have the problem description of thickness optimization. Since both Kirchhoff–Love shell total work W^{A} and W^{B}, and penalty energy $W^{\text{AB}}_{\text{pen}}$ involve shell thickness, the total derivative and associated partial derivatives of the thickness optimization problem can be acquired by replacing $\mathbf{P}_{\text{FFD}}$, $\mathbf{P}^{\text{FE}}$, $\mathbf{P}^{\text{IGA}}$ with $\mathbf{t}_{\text{FFD}}$, $\mathbf{t}^{\text{FE}}$ and $\mathbf{t}^{\text{IGA}}$ in Eqs. (42)–(44), respectively. The total derivative of FFD-based thickness optimization reads as

$$\frac{\mathrm{d}f^{\text{obj}}}{\mathrm{d}\mathbf{t}_{\text{FFD}}} = \left(\frac{\partial f^{\text{obj}}}{\partial \mathbf{t}^{\text{IGA}}} - \frac{\partial f^{\text{obj}}}{\partial \mathbf{u}^{\text{IGA}}} \left(\frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{u}^{\text{IGA}}} \right)^{-1} \frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{t}^{\text{IGA}}} \right) \frac{\mathrm{d}\mathbf{t}^{\text{IGA}}}{\mathrm{d}\mathbf{t}_{\text{FFD}}} \, . \tag{46}$$

In some applications, one may choose to have a constant thickness for each shell patch. This can be easily achieved within the current framework by relating the shell thickness in IGA DoFs to one scalar value $t^{\text{I,const}}$ as

$$\mathbf{t}^{\text{I,IGA}} = \mathbf{c}^{\text{I}} t^{\text{I,const}} \, , \tag{47}$$

where $\mathbf{c}^{\text{I}} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^{\text{T}}$ is a unit column vector contains $n^{\text{I,IGA}}$ entries. The total derivative for piecewise constant thickness optimization can be obtained by replacing $\frac{\mathrm{d}\mathbf{t}^{\text{IGA}}}{\mathrm{d}\mathbf{t}_{\text{FFD}}}$ in (46) with the following derivative,

$$\frac{\mathrm{d}\mathbf{t}^{\text{IGA}}}{\mathrm{d}\mathbf{t}^{\text{const}}} = \operatorname{diag}(\mathbf{c}^{\text{A}}, \mathbf{c}^{\text{B}}, \dots, \mathbf{c}^{I^m}) \, . \tag{48}$$

The FFD block is not needed in piecewise constant only thickness optimization.

These two approaches can be combined together to achieve a more realistic design, where specific sections of the structure necessitate a continuous thickness distribution while constant thickness is better suited for other patches. In the implementation, shell patches can be separated into various groups. One group comprises the shell patches immersed within an FFD block allowing for a continuous thickness distribution. On the other hand, the shell patches not contained in an FFD block are assumed to have a constant thickness. Moreover, shell patches originating from different FFD blocks would exhibit discontinuous thickness at their intersections. Therefore, the combined thickness optimization approach provides more flexibility.

4.4 Implementation

This section introduces the open-source implementation for the proposed optimization method in Sects. 4.1–4.3. We employ the open-source Python library PENGoLINS [44] for IGA of non-matching shell structures. PENGoLINS utilizes functionalities in pythonOCC [59] to approximate patch intersections in CAD geometries. Additionally, it leverages advanced code generation in FEniCS [50] and extraction operators in tIGAr [49] to perform IGA. CAD models in IGES or STEP formats can be imported into PENGoLINS, making them directly available for IGA. The code generation capabilities and computer algebra in FEniCS allow us to obtain partial derivatives in (23)–(25), as well as (39) automatically. Integrating computed analytical derivatives with matrices $\mathbf{M} = \operatorname{diag}(\mathbf{M}^{\text{A}}, \mathbf{M}^{\text{B}}, \dots, \mathbf{M}^{I^m})$ and $\mathbf{A}_{\text{FFD}} = \operatorname{diag}(\mathbf{A}^{\text{A}}_{\text{FFD}}, \mathbf{A}^{\text{B}}_{\text{FFD}}, \dots, \mathbf{A}^{I^m}_{\text{FFD}})$, we can calculate the total derivatives in (44) and (46) for shape and thickness optimization.

With the availability of automated derivatives, we use the open-source framework CSDL [60] for conducting gradient-based large-scale optimization. Each partial derivative computed from FEniCS, in combination with predefined matrices, is passed to a CSDL sub-model, which contains explicit or implicit operations. A system-level model is created to connect all sub-models to enable efficient and modularized design optimization. For the FFD-based shape optimization problem, which involves minimizing the internal energy of a non-matching shell structure, four essential sub-models are required and listed in Table 1. All the derivatives required for these models are defined in Sects. 3.2 and 4.2.

The models listed in Table 1 and other constraint models are modularized in a Python library named GOLDFISH (Gradient-based Optimization, Large-scale Design Framework for Isogeometric SHells). This library provides users with the option to use predefined models or create custom models to build the system-level model for their specific optimization needs. The library also includes an open-source optimizer, SLSQP [61], which can be utilized for shape or thickness optimization of non-matching shells. For optimization problems involving a large number of design variables, the SNOPT [62] optimizer is available for faster convergence. Furthermore, we have integrated the analysis framework with another optimization library OpenMDAO [63], which is developed by NASA. This integration further enhances the accessibility of the GOLDFISH library, enabling users to benefit from the collective expertise of the OpenMDAO community. The GOLDFISH library is hosted on a GitHub repository [64] and open to the public.

5 Benchmark problems

A series of benchmark problems are considered to verify the effectiveness of the optimization method. Sections 5.1–5.3 illustrate that baseline non-matching shell structures with

Table 1 Essential CSDL models for FFD-based shape optimization

Model	Input(s)	Output	Operation	Derivative(s)
CP FFD to FE	$\mathbf{P}_{\text{FFD}}$	$\mathbf{P}^{\text{FE}}$	Explicit	$\mathbf{A}_{\text{FFD}}$
CP FE to IGA	$\mathbf{P}^{\text{FE}}$	$\mathbf{P}^{\text{IGA}}$	Implicit	$\mathbf{M}$
Disp	$\mathbf{P}^{\text{IGA}}$	$\mathbf{u}^{\text{IGA}}$	Implicit	$\frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{P}^{\text{IGA}}}, \frac{\partial \mathbf{R}_{\text{t}}}{\partial \mathbf{u}^{\text{IGA}}}$
Int energy	$\mathbf{P}^{\text{IGA}}, \mathbf{u}^{\text{IGA}}$	W_{int}	Explicit	$\frac{\partial W_{\text{int}}}{\partial \mathbf{P}^{\text{IGA}}}, \frac{\partial W_{\text{int}}}{\partial \mathbf{u}^{\text{IGA}}}$

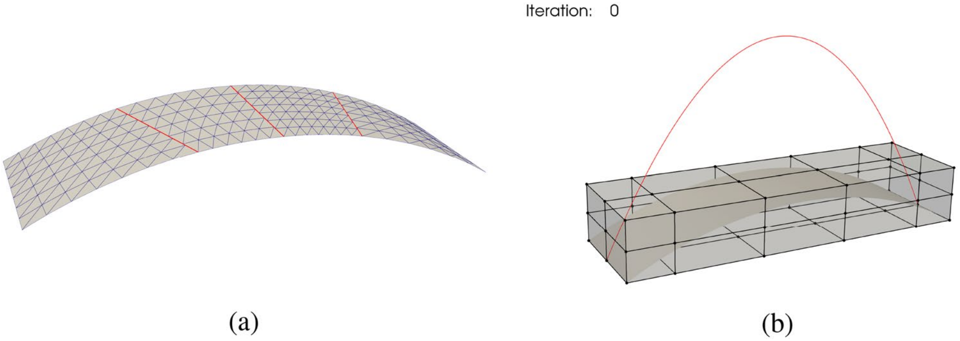


Fig. 6 **a** Baseline geometry of a non-matching arch consisting of four NRUBS patches, three surface–surface intersections are indicated with red lines. **b** Initial configuration of the arch immersed in an FFD block, where black lines and dots denote the control net. The analytical optimal shape is plotted with a red curve

arbitrary intersections are able to accurately converge to the analytical optimum. Section 5.4 studies the capability and flexibility of the framework for thickness optimization.

5.1 Arch shape optimization

An arch fixed at two ends and subjected to a constant downward load per unit horizontal length is modeled by a Kirchhoff–Love shell theory. Detailed problem definitions can be found in [38, Section 8]. To test the effectiveness of FFD-based shape optimization for the non-matching shells approach, we model the arch using four NURBS patches with three intersections, where the arch geometry in the baseline configuration is shown in Fig. 6a. We immerse the arch geometry in a trivariate B-spline block in the initial configuration, as is illustrated in Fig. 6b. The analytical optimal solution is given by a quadratic parabola, where the ratio between the height of the arch and the horizontal distance of two fixed edges is 0.54779.

We use quadratic B-spline for the FFD block in all three directions, $p_{\mathrm{FFD}} = 2$. The arch shell patches are described by cubic NURBS surfaces, $p_{\mathrm{sh}} = 3$, with 1086 DoFs in total. This benchmark problem minimizes the internal energy of the shell structure, with vertical coordinates of the control points of the FFD block being the design variables. From the control net in Fig. 6b, it can be observed that there are 54 design variables. Two constraints are applied to this problem. The first constraint ensures that the lines in the FFD control net are parallel to the axial direction of the arch, keeping the arch devoid of tilting or twisting during the optimization process. The second constraint fixes the bottom layer of FFD control points so that the two edges of the arch remain in the initial position. We use the SLSQP optimizer with a tolerance of 10^{-12} to perform the optimization, snapshots of the optimization iteration are demonstrated in Fig. 7.

The arch converges to the analytical optimum shape after 40 iterations. The shape of the FFD block in the final configuration is shown in Fig. 7. As anticipated, the optimized

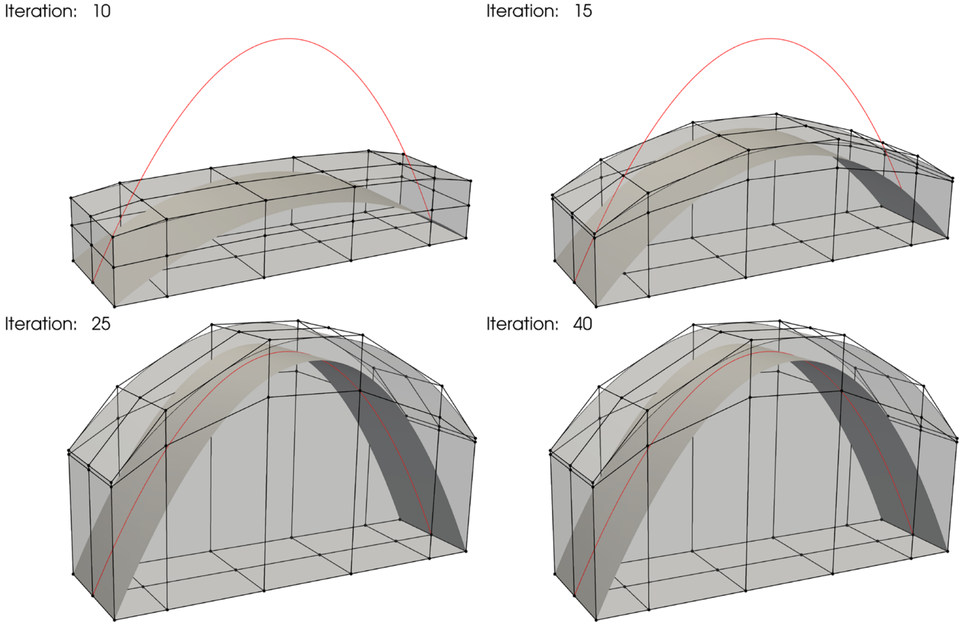


Fig. 7 Snapshots of non-matching arch shape optimization

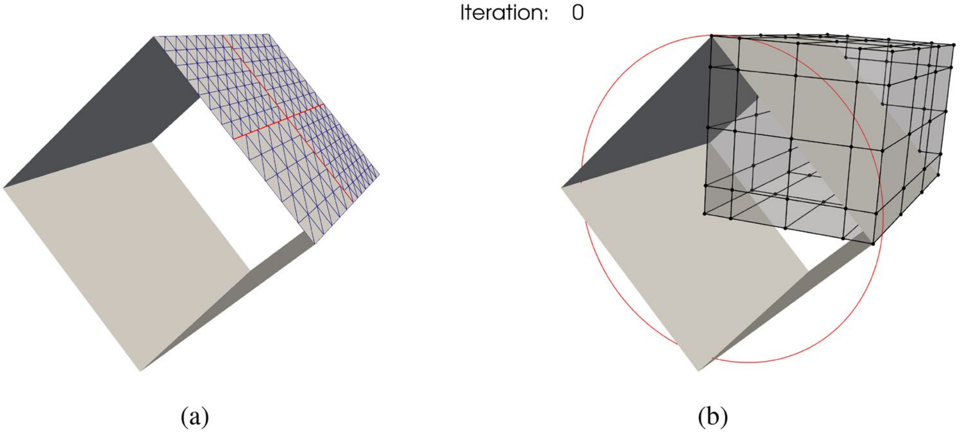


Fig. 8 **a** Baseline geometry of the square tube, a quarter of the tube is modeled using four non-matching B-spline patches with four intersections. **b** Initial configuration of the FFD block with control net. The optimal cross section is depicted by a red circle

arch geometry is still contained in the FFD block. The height to base span ratio in this problem is measured as 0.54748, exhibiting a relative error of 0.057% compared to the exact value. Considering the coarse discretization of the arch geometry, the results are encouraging.

5.2 Tube shape optimization

A square tube in the baseline configuration is subjected to an internal constant pressure. The optimal shape is given by a cylindrical tube [38, Section 7]. We use four cubic NURBS surfaces to model one-quarter of the square tube, where the initial geometry is illustrated in Fig. 8a. The square tube geometry contains four non-matching intersections with 1035 DoFs in total, and symmetric boundary conditions are applied in the analysis. The geometry is immersed in a cubic B-spline block to perform FFD-based shape optimization, as shown in Fig. 8b, where the cross-section of the optimal shape is indicated by the red curve. In this example, the horizontal and vertical coordinates of the FFD control points are design variables, totaling 200 design variables. Similar constraints are employed in this problem as those in the arch shape optimization. Control points on the left and bottom layers are fixed, where the lines in the FFD control net that are parallel to the tube axis maintain their orientations.

The optimization problem converges successfully after 42 iterations using SLSQP optimizer with a tolerance of 10^{-12}, and snapshots are depicted in Fig. 9. As expected, the initial square tube converges to the cylindrical tube.

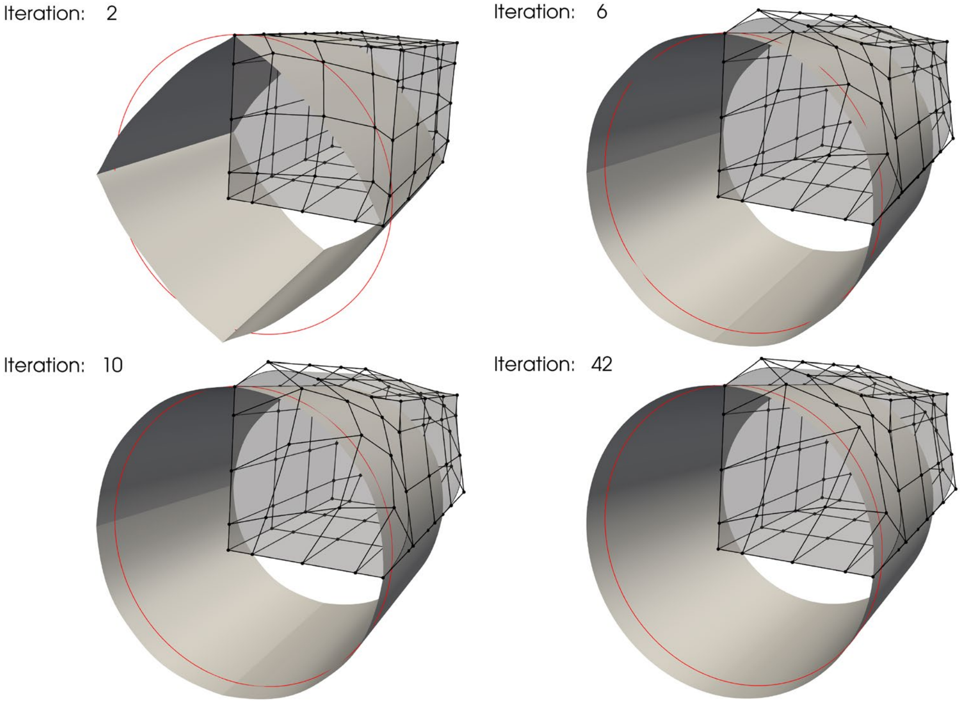


Fig. 9 Iteration history for tube shape optimization under follower pressure

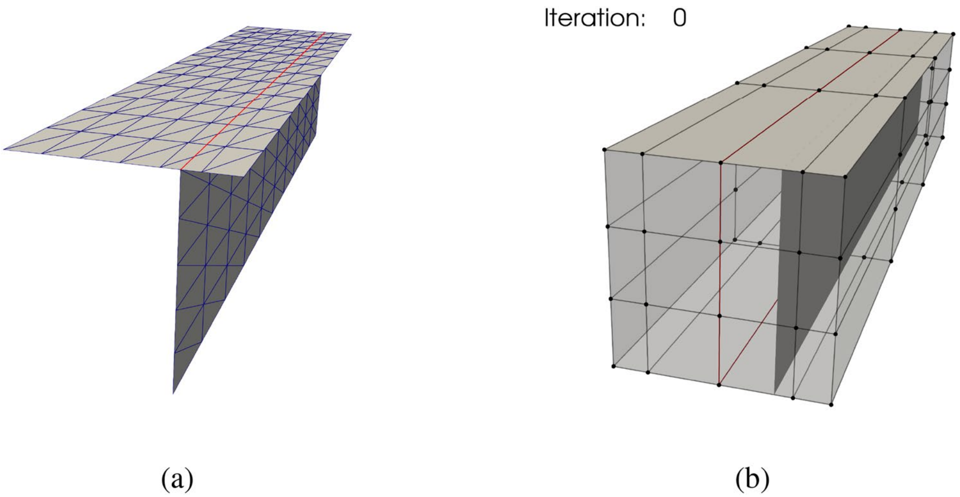


Fig. 10 **a** Baseline configuration of a T-beam whose vertical patch is at the three-quarter position of the horizontal patch. **b** The T-beam is placed in an FFD B-spline block. The optimal position of the vertical patch is depicted with red lines

5.3 T-beam shape optimization

In this section, a T-beam is considered to test the method for shell structures with intersections in the middle. We model a T-beam using two NURBS patches which are shown in Fig. 10a. In the baseline design, the vertical patch in the T-beam is located at the three-quarter position, where the mismatched intersection is indicated with a red line.

In this benchmark problem, we aim to minimize the internal energy of the T-beam by updating the horizontal coordinates of shell patches' control points. Subsequently, the T-beam is placed in a cubic FFD block, where the

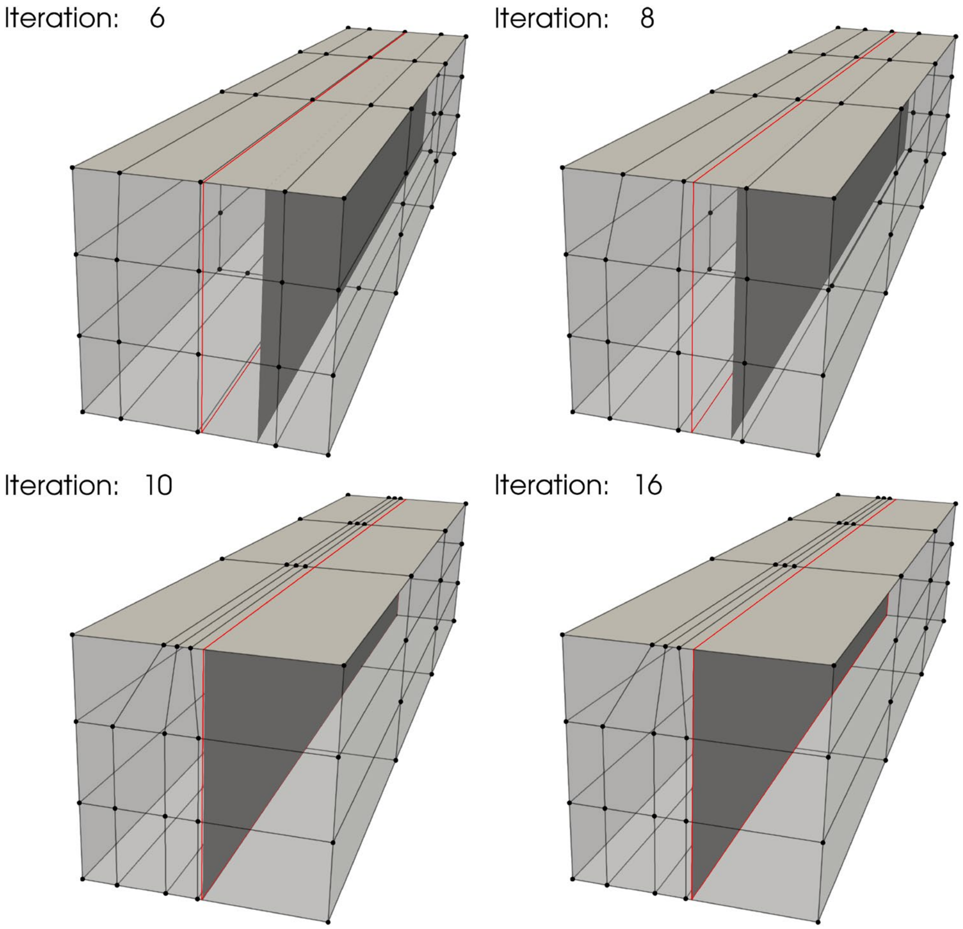


Fig. 11 Screenshots of T-beam shape optimization process

horizontal coordinates of the FFD block's control points serve as design variables. The rear end of the T-beam is fixed. Given a downward distributed load, the optimal shape of the T-beam is expected to have a junction at the center of the horizontal patch under a constant volume constraint. The optimal position of the vertical patch is depicted in Fig. 10b with red lines. The right and left layers of the FFD block's control points are fixed by employing equality constraints, and control points stay collinear in the axial direction. Using a tolerance of 10^{-15}, the SLSQP optimizer converges successfully after 16 iterations, and the optimization process is shown in Fig. 11.

In Fig. 11, the T-beam converges to the optimal solution, indicating that the proposed method is effective for non-matching shell structures with intersections. A more complicated demonstration is presented in Sect. 6.2, where the geometry contains curved T-junctions.

5.4 Thickness optimization of a clamped plate

As stated in Sect. 4.3, the FFD-based optimization methodology can also be applied to thickness optimization. In the following section, we first demonstrate a piecewise constant thickness optimization for a clamped non-matching plate, in which the FFD block is not needed. Subsequently, we proceed to perform the variable thickness optimization for the same geometry.

5.4.1 Piecewise constant thickness optimization

For the thickness optimization example, a unit square plate composed of six cubic non-matching NURBS surfaces is considered. The geometry, which is shown in Fig. 12a, exhibits 5 intersections with a total of 1449 DoFs. We apply a clamped boundary condition on the left side and with a line force applied to the right side in the normal direction of the plate. All patches of the plate have an initial thickness of 0.01 m. Using the strategy introduced in Sect. 4.3, we perform piecewise constant thickness optimization for the clamped plate to minimize the internal energy under the constant volume constraint. This problem only has 6 design variables. In this and the following sections, the SNOPT optimizer is used for faster convergence. The optimal thickness is plotted in Fig. 12b.

The observed optimal piecewise constant thickness in Fig. 12b shows material redistributes toward the clamped side, which provides enhanced support to the plate. The

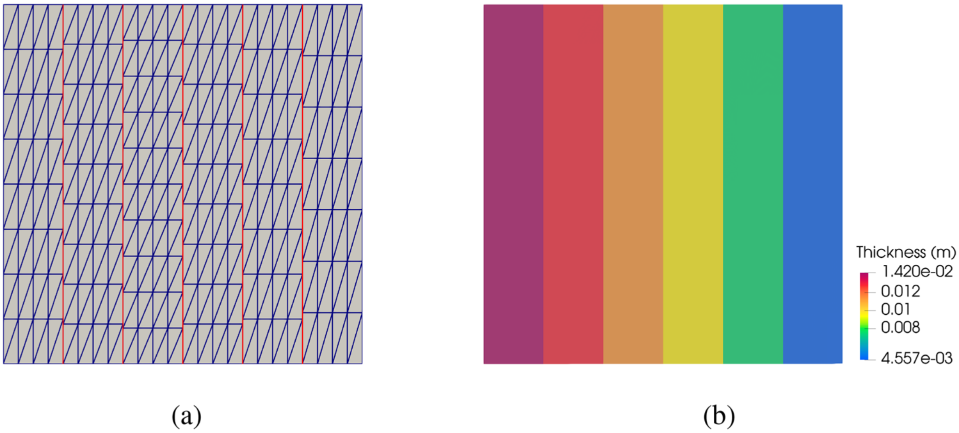


Fig. 12 **a** A unit square plate consisting of six non-matching patches, intersections are indicated with red lines. **b** Final plate thickness for piecewise constant thickness optimization

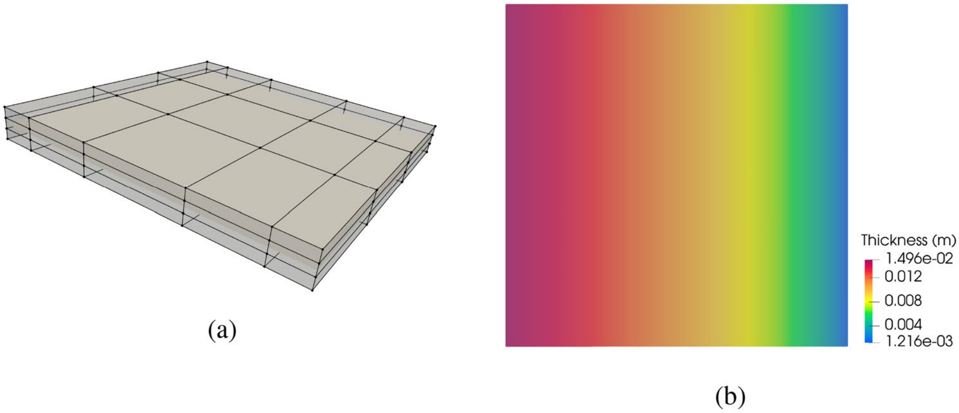


Fig. 13 **a** The non-matching plate is immersed in an FFD block. **b** Optimized thickness distribution using the FFD-based approach

internal energy in the final configuration is 37.17% less than the baseline configuration.

5.4.2 Variable thickness optimization

We now perform variable thickness optimization for the non-matching plate. The plate is placed in a cubic B-spline block, which is shown in Fig. 13a. Besides the constant volume constraint, we only optimize the plate thickness in one direction that is perpendicular to the intersections.

With the FFD-based approach, the continuity of shell thickness is maintained at all intersections. Figure 13b depicts the converged solution, where a smooth thickness distribution is observed. The smooth thickness profile offers an improved design compared to the piecewise constant thickness approach, resulting in a 40.20% reduction of the initial internal energy. A comparison of optimization iteration of normalized internal energy between the two methods is illustrated in Fig. 14a. The FFD-based thickness optimization approach converges to a smaller internal energy.

To validate the proposed method, we compare the continuous thickness profile to an optimal thickness configuration of a cantilever beam [65]. The cantilever beam is modeled using the Euler–Bernoulli beam theory, where a point load is applied to the free end. Since the Kirchhoff–Love shell is an extension of the Euler–Bernoulli beam, both models are expected to yield identical thickness distributions. The normalized thickness profiles of these two models, along with the piecewise constant thickness profile, are plotted in Fig. 14b. A good agreement is observed between the variable thickness of the Kirchhoff–Love shell at the center line and the Euler–Bernoulli beam. Meanwhile, the cross-sectional view of the piecewise constant thickness shows a similar trend to the Euler–Bernoulli beam, albeit with discontinuities at the intersections.

Table 2 Reduction of internal energy of the clamped plate for different degrees of the FFD block

p_{FFD}	1	2	3	4
Internal energy reduction (%)	39.76	40.11	40.20	40.22

We then investigate the effect of basis function order of continuity in the FFD block. Using the same knots vectors as illustrated in Fig. 13a, we increase the order of the B-spline basis functions from linear (C^0) to quartic (C^3) and compare the amounts of reduced internal energy relative to the baseline configuration. These data points are summarized in Table 2. The results presented in Table 2 indicate that an FFD block with quadratic B-spline basis functions can achieve a better optimal thickness distribution for the clamped plate. The internal energy with quadratic FFD block only exhibits 0.27% of relative difference compared to the quartic FFD block. Table 2 also suggests that elevating the order of continuity of the FFD block leads to better designs with lower internal energy, particularly when transitioning from linear to quadratic B-spline basis functions.

The plate example demonstrates that both piecewise constant and variable thickness optimization can be conducted in the proposed framework. One can select desired thickness distribution, or a mixed approach of these two,

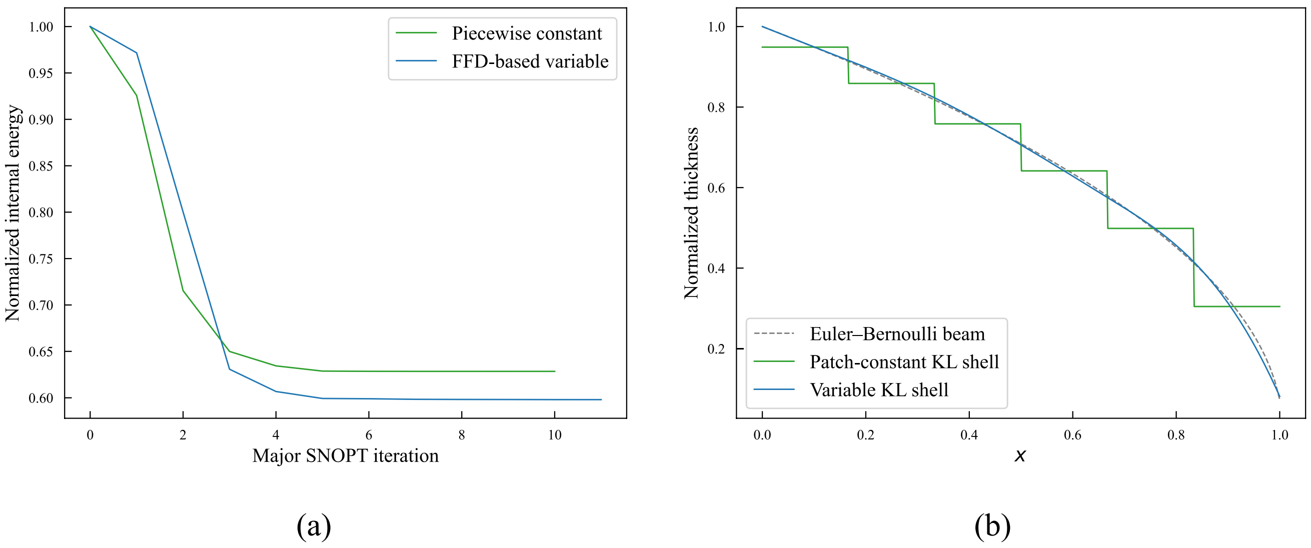


Fig. 14 **a** Optimization process of normalized internal energy for two approaches. **b** Cross-sectional view of piecewise constant thickness and variable thickness, and comparison with Euler–Bernoulli beam thickness optimization

demonstrated in Sect. 6, based on the physical conditions and problem requirements to obtain an optimal design.

6 Application to aircraft wings

Two aircraft wing design optimization problems are performed. Section 6.1 considers a Parallel Electric-Gas Architecture with Synergistic Utilization Scheme (PEGASUS) wing thickness optimization problem, and Sect. 6.2 demonstrates a simultaneous shape and thickness optimization for an Electric Vertical Takeoff and Landing (eVTOL) aircraft wing. Both applications illustrate that a design-analysis-optimization workflow can be achieved with the proposed framework.

6.1 PEGASUS wing thickness optimization

For the PEGASUS wing problem, we first verify the accuracy of the structural analysis using PENGoLINS for a shell structure with a large number of patches and intersections. Then two types of thickness design optimizations are performed in the following section.

6.1.1 Structural analysis of the PEGASUS wing

The PEGASUS wing CAD model is created using a customized geometry engine, an exploded view of the wing with IGA discretization is shown in Fig. 15. The CAD model consists of 90 NURBS patches, two outer skins (one lower skin and one upper skin) and two spars (one front spar and one rear spar) connecting two adjacent ribs. The NURBS surfaces in the PEGASUS wing are represented using cubic basis functions with maximal continuity, resulting in 19,572 DoFs in total. Additionally, 280 patch intersections are formed in the wing structure.

The PEGASUS wing is made of material with Young's modulus 69 GPa and Poisson's ratio 0.35, and the wing span is 12.22 m. At the wing root, the chord is 1.52 m and the airfoil thickness is 0.37 m. A uniform initial thickness is 5 mm for all patches. Considering an aircraft take-off weight of 9000 kg, a distributed upward pressure with a magnitude of 132.5 N/m^2 is determined by dividing half of the take-off weight by the surface area of the wing. Clamped boundary conditions are imposed at the wing root. Importing the PEGASUS wing geometry in IGES format to PENGoLINS, we perform structural analysis directly without finite element mesh generation. Given an analysis-suitable CAD file, the only required geometry preprocessing is to approximate surface–surface intersections, which is a much easier effort than finite element mesh generation and is automated in PENGoLINS using the functionality provided by pythonOCC. Figure 16a shows all the intersections presented in the PEGASUS wing, while the displacements computed by PENGoLINS are visualized in Fig. 16c. Figure 16e shows the distribution of von Mises stress on top surfaces ($\xi_3 = h/2$) of the PEGASUS wing.

To validate the proposed non-matching coupling method for complex shell structures, we conduct a much refined FE analysis (utilizing quadratic triangular elements with 118644 DoFs) for the PEGASUS wing using COMSOL [66]. Figure 16b displays an extensively refined finite element mesh for the COMSOL FE analysis. Displacements solved in COMSOL and corresponding von Mises stress are depicted in Fig. 16d and f, respectively. Figures 16c and d indicate that the displacements obtained from PENGoLINS closely match the results from COMSOL. The maximum displacement magnitude in PENGoLINS is

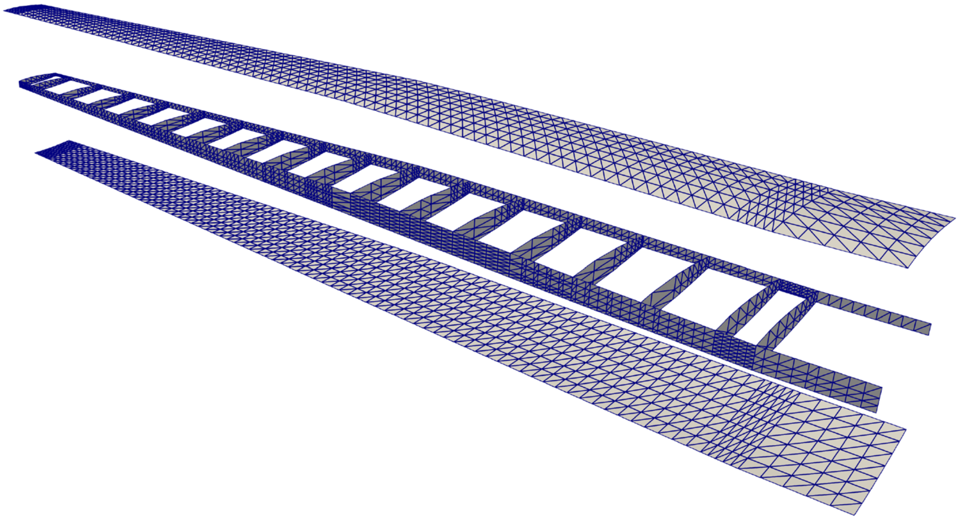
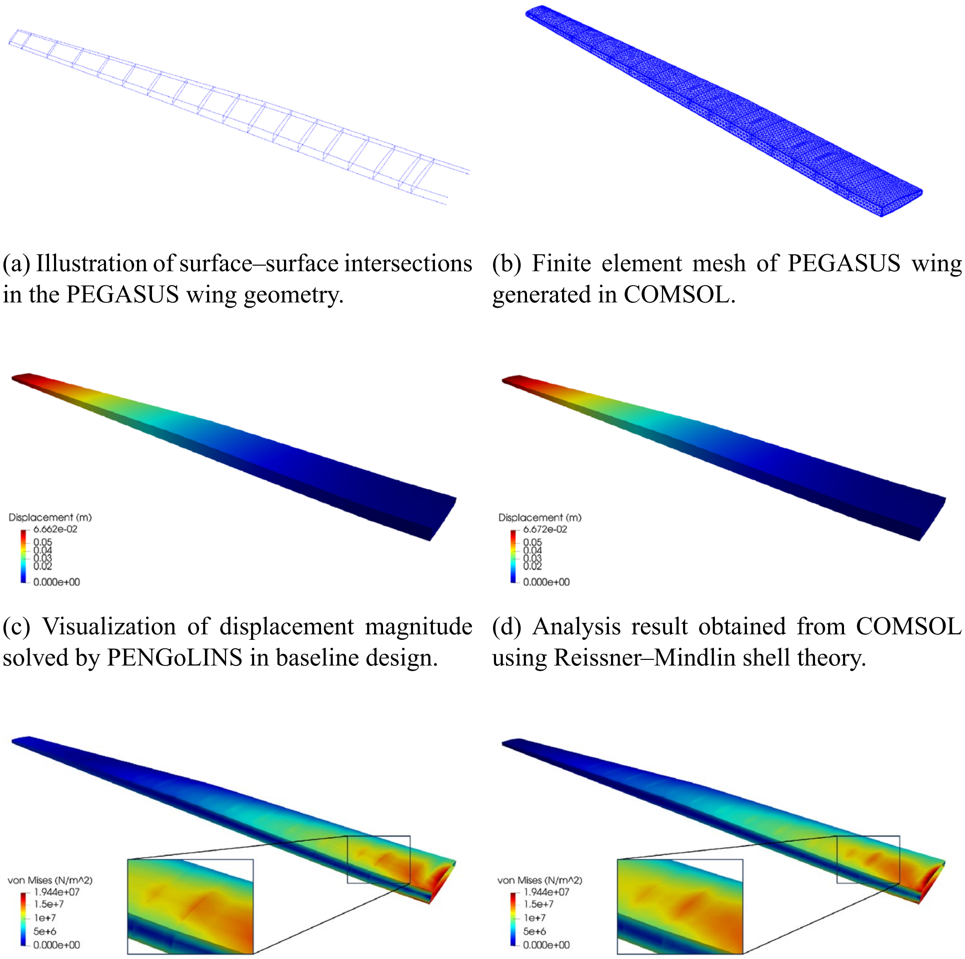


Fig. 15 CAD geometry of the PEGASUS wing which is composed of 90 NURBS patches with 280 intersections, totaling 19,572 DoFs



(a) Illustration of surface–surface intersections in the PEGASUS wing geometry.

(b) Finite element mesh of PEGASUS wing generated in COMSOL.

(c) Visualization of displacement magnitude solved by PENGoLINS in baseline design.

(d) Analysis result obtained from COMSOL using Reissner–Mindlin shell theory.

(e) von Mises stress on top surfaces of PEGASUS wing computed by PENGoLINS.

(f) Distribution of von Mises stress obtained from COMSOL.

Fig. 16 Structural analysis of the PEGASUS wing using PENGoLINS, and the resulting displacement magnitude and von Mises stress are compared with the corresponding outputs obtained from COMSOL

0.06662 m, which has a relative difference of 0.15% compared to the corresponding value of 0.06672 m in COMSOL. This aligns well with the findings of a numerical experiment presented in [44, Section 5.3]. The von Mises stress distributions between PENGoLINS and COMSOL also exhibit good agreement as shown in Fig. 16e and f, where zoom-in views are included for the observation of von Mises stress at patch intersections. Note that while Reissner–Mindlin shell theory was employed in COMSOL, the mixed interpolation of tensorial components (MITC) [67] technique implemented in COMSOL has been carefully validated and is reliable for comparison with our solution based on Kirchhoff–Love shell theory in the limit of thin shells. The simulation results for the PEGASUS wing indicate that PENGoLINS provides good accuracy for complex shell structures, which is crucial for the subsequent design optimization.

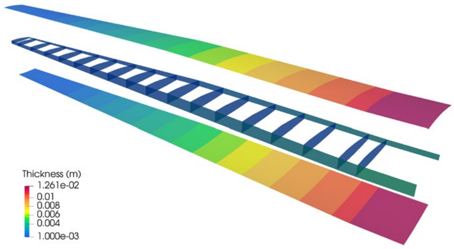


Fig. 17 Optimization result of the PEGASUS wing with piecewise constant thickness

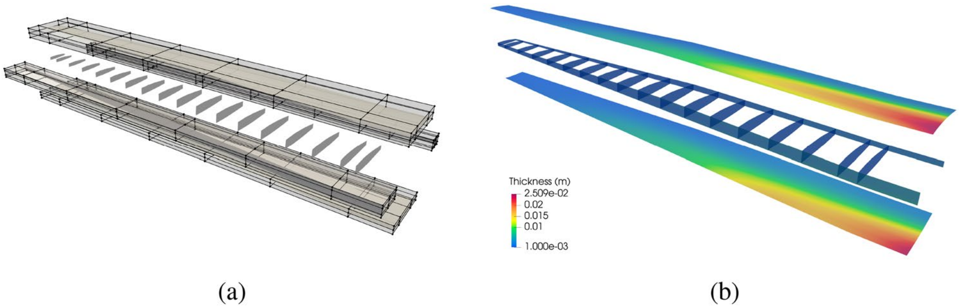
6.1.2 Thickness optimization of the PEGASUS wing

Similar to the thickness optimization of the clamped plate discussed in Sect. 5.4, we apply the same methodology to the PEGASUS wing for piecewise constant thickness optimization. The same boundary conditions are employed throughout the optimization. In the piecewise constant thickness optimization case, a total of 90 design variables are included with lower and upper bounds of 1 mm and 100 mm, respectively. The initial thickness for all patches is taken as 5 mm. A constant volume constraint is employed in the optimization. The resulting shell thickness with minimum internal energy is depicted in Fig. 17. The shell patch with the maximum thickness is observed at the wing root in the outer skins, while the thickness decreases consistently along the span direction for both the outer skins and spars, following a pattern similar to that of the clamped plate. The thicknesses of the internal ribs and the wingtip are close to the lower bound since the bending moments are mainly carried by the lower and upper skins given the distributed upward load. Therefore, the majority of material is redistributed toward the clamped root of the outer skins. The optimized design in Fig. 17 gives an internal energy 38.17% less than that of the baseline configuration.

To achieve an improved design, we consider variable thickness in the outer skins and spars of the PEGASUS wing, while keeping the internal ribs with piecewise constant thickness. The configuration of the FFD blocks for variable thickness optimization is illustrated in Fig. 18a. Four FFD blocks with quadratic bases are created to allow for variation in thickness within a single spline patch while ensuring continuity at patch intersections, with a total of 402 design variables used for this problem. By minimizing the internal energy again, the optimal thickness distribution is obtained as shown in Fig. 18b. Both applications in this section use the SNOPT optimizer with a tolerance of 10^{-4}. In the optimized design, the outer skins of the PEGASUS wing at the clamped edge still have the largest thickness, where the thickness distribution is smooth at the surface–surface intersections within each FFD block. The decreasing thickness trend in outer skins and spars remains consistent with the optimal piecewise constant thickness case. Comparing the combined optimization strategy with the piecewise constant method, the maximum thickness in the former is higher while maintaining the same volume. Additionally, the internal energy is reduced by 44.71% compared to the baseline design. These observations indicate that the combined thickness optimization method demonstrates a more efficient utilization of material than the piecewise constant method.

6.2 Simultaneous optimization for eVTOL wing

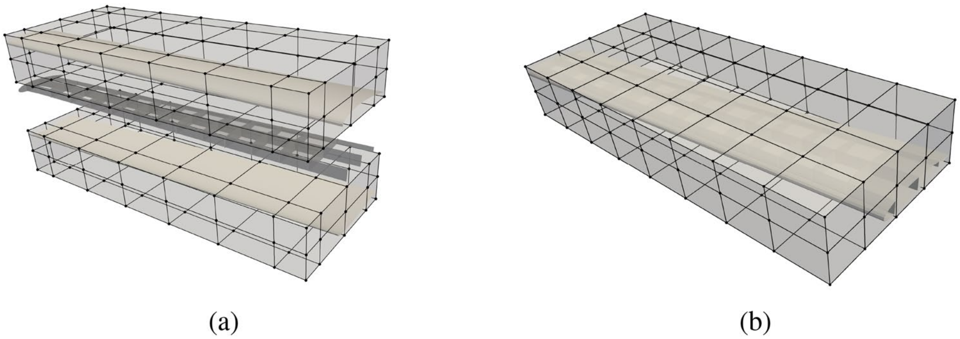
With the continuous advancements in aviation battery technology [68], eVTOL aircraft have emerged as a promising solution for cost-effective urban mobility [69]. In this section, we use a more advanced eVTOL wing to demonstrate the capabilities of the FFD-based optimization approach, where both the thickness and shape control points are considered as design variables simultaneously. By incorporating both thickness and shape coordinates in the design of shell structures, we can utilize the material more efficiently than considering thickness optimization only. The CAD geometry of the eVTOL wing, material parameters, and the corresponding structural analysis can be found in [44, Section 5]. For the optimization problem, we implement the same clamped boundary conditions and distributed upward pressure as those in the previous application. The magnitude



(a)

(b)

Fig. 18 **a** Configuration of the combined thickness optimization. Each group of outer skins and spars is placed in one FFD block, and the remaining internal ribs have a piecewise constant thickness. **b** Optimal thickness distribution of PEGASUS wing



(a)

(b)

Fig. 19 **a** FFD blocks for eVTOL wing thickness optimization, where the lower and upper skins have a variable thickness. Wingtips and internal ribs and spars have a piecewise constant thickness. **b** FFD block for shape optimization of pressure is set to 120 $\mathrm{N/m^2}$, approximated by dividing the take-off weight by the surface area of the eVTOL wing. The baseline configuration of the eVTOL wing, which is composed of 27 cubic NURBS patches with 87 intersections, is illustrated in Fig. 19. We note that we perform the shape optimization for the eVTOL wing without including an aerodynamic model (which would be needed for a well-posed wing design problem) purely to provide a demonstration of the FFD-based method for complex aerospace structures.

To achieve a meaningful design for the eVTOL wing, we use two quadratic B-spline FFD blocks for thickness optimization. This configuration allows for varying thicknesses in the lower and upper skins of the eVTOL, while using piecewise constant thicknesses for the internal stiffeners and wingtips. The arrangement of the thickness FFD blocks is illustrated in Fig. 19a. Furthermore, a cubic B-spline FFD block is created for shape optimization, as depicted in Fig. 19b. Only the vertical coordinates of control points for the shape FFD block, denoted as $\mathbf{P}_{\mathrm{FFD3}}$, are updated. In total, there are 642 design variables involved in this optimization process, and a constant volume constraint is applied as well. Regarding the thickness design variables, the lower and upper limits are selected as 1 mm and 50 mm, respectively. All shell components have initial thicknesses of 3 mm.

One challenge encountered during shape optimization of complex shell structures, such as eVTOL wings, is the potential occurrence of oscillatory or highly distorted geometries in the updated designs. These distorted shapes can lead to poor element quality, resulting in spurious energy and affecting the accuracy of the analysis. To mitigate oscillation or radical change of shell components, an additional term is introduced in the objective function to regularize the gradient of the shape. The objective is formulated as

$$f^{\mathrm{obj}} = \sum_{\mathrm{I}=1}^{m} \left(W^{\mathrm{I}}_{\mathrm{int}} + \lambda \frac{E(\mathring{t}^{\mathrm{I}})^3}{12\mathring{h}^{\mathrm{I}}_{A}(1-\nu^2)} \int_{\mathcal{S}^{\mathrm{I}}} \|\nabla \mathbf{P}^{\mathrm{I}}_3 - \nabla \mathring{\mathbf{P}}^{\mathrm{I}}_3\|^2 \, \mathrm{d}\mathcal{S} \right), \tag{49}$$

where I is the index of shell patches, λ is a dimensionless regularization coefficient, h^I_A is the element area of shell patch I in the physical space, $\mathbf{P}^{\mathrm{I}}_3$ is the vertical component of

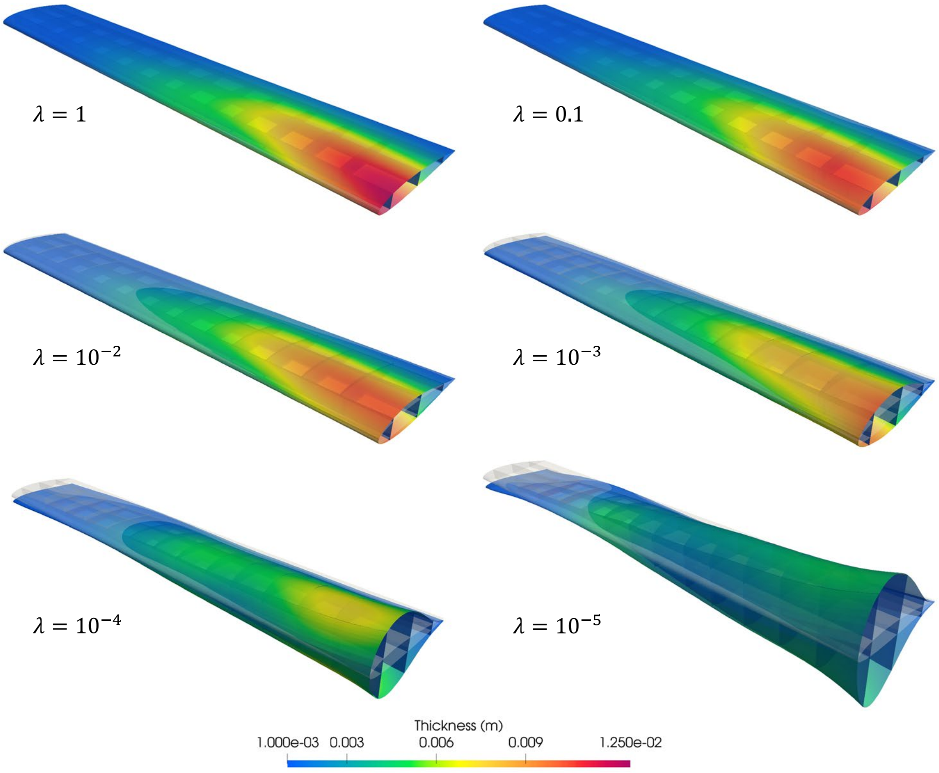


Fig. 20 Optimized solutions of the eVTOL wing with varying regularization coefficient

Table 3 Reduction of internal energy of the eVTOL wing after simultaneous optimization with varying regularization coefficients

λ	1	0.1	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Internal energy reduction (%)	49.07	51.58	66.45	83.23	92.82	95.09

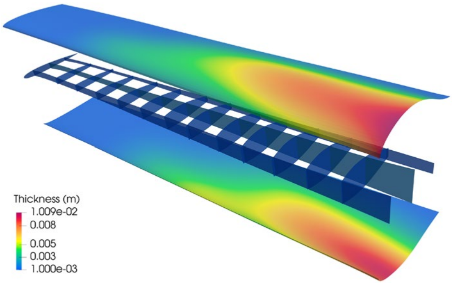


Fig. 21 Exploded view of optimal design of eVTOL wing with regularization coefficient $\lambda = 10^{-3}$, which results in 83.23% reduction of internal energy compared to the baseline design

the shape variable for shell patch I, and $\mathcal{S}^{\mathrm{I}}$ is the midsurface of shell patch I. $\mathring{(\cdot)}$ denotes quantities in the baseline configuration. The regularization term in (49) serves as an additional artificial bending energy associated with the curvature of the eVTOL wing. Therefore, the shape oscillation can be eliminated by adjusting the regularization coefficient λ. The SNOPT optimizer is employed with a tolerance of 10^{-3}. Figure 20 demonstrates the optimized eVTOL wing designs achieved with varying λ values, providing insights into the influence of regularization on the final shape and thickness distribution. It can be seen that the patch intersections of the optimized shapes in Fig. 20 are still connected using the FFD-based approach. The reductions of internal energy for different λ are listed in Table 3.

For this eVTOL wing optimization problem, different regularization coefficients yield distinct outcomes. A regularization coefficient of 1 and 0.1 provide optimal designs dominated by thickness update, resembling the patterns observed in the combined thickness optimization of the PEGASUS wing, refer to Fig. 18b. The optimized shapes in these two cases are similar to the baseline configuration since the artificial bending energies are the major contributor to the objective function due to the large regularization coefficients. Even slight variations in the shape variables result in substantial increases in the objective function. Conversely, small regularization coefficients, i.e., 10^{-4} and 10^{-5}, lead to considerable changes in the shape of the eVTOL wing and reduction of internal energy, amounting to 92.82% and 95.09%, respectively. However, these cases exhibit noticeable oscillations in the wingtip area, which can lead to ill-conditioned systems. On the other hand, employing regularization coefficients with values of 10^{-2} and 10^{-3} yields balanced solutions, where both the thickness redistribution and shape updates contribute to the optimal design. The material moves toward the clamped area, accompanied by a widening of the cross-section to provide increased wing support. An exploded view of the optimal design with $\lambda = 10^{-3}$ is shown in Fig. 21. This optimal design achieves an internal energy reduction of 83.23% compared to the baseline configuration.

7 Conclusion

In this paper, we have introduced an FFD-based shape and thickness optimization approach for shell structures composed of separately-parametrized NURBS surfaces. The integration of this method with the Lagrange extraction technique enables IGA with existing FE toolkits and provides a connection between the FFD block and non-matching shell patches. By employing the FFD block approach, the updated shell geometry and thickness remain properly connected at patch intersections throughout the optimization process. This feature prevents undesired shape discontinuities in shell structures.

We made use of the penalty-based coupling formulation for Kirchhoff–Love shells to perform IGA in the optimization. The automation of analytical derivative computations is achieved through code generation in FEniCS, enabling gradient-based multidisciplinary design optimization. This automation streamlines the optimization process and allows for efficient exploration of design spaces. The unified NURBS representation shared by both the design geometry and analysis model enhances accuracy per DoF in the analysis and precise design updates. Moreover, the proposed framework circumvents FE mesh generation and streamlines design–analysis–optimization workflow for complex shell structures. Consequently, the automated workflow is expected to accelerate the conceptual design of novel eVTOL aircraft with minimal manual effort.

A suite of benchmark problems is adopted to verify the effectiveness of the FFD-based optimization approach proposed in this paper. Both the shape optimization and thickness optimization results agree well with analytical solutions or other established references. Furthermore, we have applied the framework to two different aircraft wings. This demonstration highlights the potential of the proposed method in exploring complex design spaces and obtaining superior designs for innovative aircraft structures. Future works can extend the shape optimization for aircraft wings with more realistic loads by coupling the structural solver with an aerodynamic solver [70]. Additionally, practical constraints such as von Mises stress can be incorporated into the optimization process to ensure that the obtained designs meet the requirements of real-world applications. To promote code transparency and potential contributions to the shell optimization community, we make the Python library GOLDFISH openly accessible in a GitHub repository [64].

Acknowledgements This work is supported by National Aeronautics and Space Administration grant number 80NSSC21M0070. We thank Michael A. Warner (UCSD) and Darshan Sarojini (UCSD) for providing the CAD geometry of the PEGASUS wing in Fig. 15.

Data availability The codes used to replicate the results presented in this paper are openly accessible under open-source licenses on GitHub at the following link: https://github.com/hanzhao2020/GOLDFISH.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Farshad M (2013) Design and analysis of shell structures, vol 16. Springer, Geneva
2. Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. SIGGRAPH Comput. Graph. 20(4):151–160. https://doi.org/10.1145/15886.15903
3. Johnson C (2012) Numerical solution of partial differential equations by the finite element method. Dover books on mathematics series. Dover Publications, New York
4. Hardwick MF, Clay RL, Boggs PT, Walsh EJ, Larzelere AR, Altshuler A (2005) DART system analysis. Technical Report SAND2005-4647, Sandia National Laboratories
5. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Comput Methods Appl Mech Eng 194:4135–4195
6. Cottrell JA, Hughes TJR, Bazilevs Y (2009) Isogeometric analysis: toward integration of CAD and FEA. Wiley, Chichester
7. Piegl L, Tiller W (1996) The NURBS book. Springer, Berlin
8. Reddy JN (2006) Theory and analysis of elastic plates and shells. CRC Press, Boca Raton
9. Bazilevs Y, Hsu M-C, Kiendl J, Wüchner R, Bletzinger K-U (2011) 3D simulation of wind turbine rotors at full scale. Part II: fluid–structure interaction modeling with composite blades. Int J Numer Methods Fluids 65(1–3):236–253
10. Johnson EL, Hsu M-C (2020) Isogeometric analysis of ice accretion on wind turbine blades. Comput Mech 66(2):311–322
11. Herrema AJ, Kiendl J, Hsu M-C (2019) A framework for isogeometric-analysis-based optimization of wind turbine blade structures. Wind Energy 22(2):153–170
12. Kamensky D, Hsu M-C, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR (2015) An immersogeometric variational framework for fluid-structure interaction: application to bioprosthetic heart valves. Comput Methods Appl Mech Eng 284:1005–1053
13. Kamensky D, Xu F, Lee C-H, Yan J, Bazilevs Y, Hsu M-C (2018) A contact formulation based on a volumetric potential: application to isogeometric simulations of atrioventricular valves. Comput Methods Appl Mech Eng 330:522–546
14. Xu F, Morganti S, Zakerzadeh R, Kamensky D, Auricchio F, Reali A, Hughes TJR, Sacks MS, Hsu M-C (2018) A framework for designing patient-specific bioprosthetic heart valves using immersogeometric fluid–structure interaction analysis. Int J Numer Methods Biomed Eng 34(4):2938
15. Zareh M, Qian X (2019) Kirchhoff–Love shell formulation based on triangular isogeometric analysis. Comput Methods Appl Mech Eng 347:853–873
16. Kiendl J, Bletzinger K-U, Linhard J, Wüchner R (2009) Isogeometric shell analysis with Kirchhoff–Love elements. Comput Methods Appl Mech Eng 198(49–52):3902–3914
17. Kiendl J (2011) Isogeometric analysis and shape optimal design of shell structures. Ph.D. thesis, Lehrstuhl für Statik, Technische Universität München
18. Kiendl J, Hsu M-C, Wu MCH, Reali A (2015) Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. Comput Methods Appl Mech Eng 291:280–303. https://doi.org/10.1016/j.cma.2015.03.010
19. Casquero H, Liu L, Zhang Y, Reali A, Kiendl J, Gomez H (2017) Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff–Love shells. Comput Aided Des 82:140–153 (**Isogeometric design and analysis**)
20. Casquero H, Wei X, Toshniwal D, Li A, Hughes TJR, Kiendl J, Zhang YJ (2020) Seamless integration of design and Kirchhoff–Love shell analysis using analysis-suitable unstructured T-splines. Comput Methods Appl Mech Eng 360:112765
21. Kiendl J, Bazilevs Y, Hsu M-C, Wüchner R, Bletzinger K-U (2010) The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. Comput Methods Appl Mech Eng 199:2403–2416
22. Guo Y, Heller J, Hughes TJR, Ruess M, Schillinger D (2018) Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format. Comput Methods Appl Mech Eng 336:39–79
23. Benzaken J, Evans JA, McCormick SF, Tamstorf R (2021) Nitsche's method for linear Kirchhoff–Love shells: formulation, error analysis, and verification. Comput Methods Appl Mech Eng 374:113544
24. Herrema AJ, Johnson EL, Proserpio D, Wu MCH, Kiendl J, Hsu M-C (2019) Penalty coupling of non-matching isogeometric Kirchhoff–Love shell patches with application to composite wind turbine blades. Comput Methods Appl Mech Eng 346:810–840
25. Leonetti L, Liguori FS, Magisano D, Kiendl J, Reali A, Garcea G (2020) A robust penalty coupling of non-matching isogeometric Kirchhoff–Love shell patches in large deformations. Comput Methods Appl Mech Eng 371:113289
26. Coradello L, Kiendl J, Buffa A (2021) Coupling of non-conforming trimmed isogeometric Kirchhoff–Love shells via a projected super-penalty approach. Comput Methods Appl Mech Eng 387:114187
27. Coradello L, Loli G, Buffa A (2021) A projected super-penalty method for the C^1-coupling of multi-patch isogeometric Kirchhoff plates. Comput Mech 67(4):1133–1153
28. Evans JA, Bazilevs Y, Babuška I, Hughes TJR (2009) n-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. Comput Methods Appl Mech Eng 198(21):1726–1741 (**Advances in Simulation-Based Engineering Sciences-Honoring J. Tinsley Oden**)
29. Wall WA, Frenzel MA, Cyron C (2008) Isogeometric structural shape optimization. Comput Methods Appl Mech Eng 197(33–40):2976–2988

30. Cho S, Ha S-H (2009) Isogeometric shape design optimization: exact geometry and enhanced sensitivity. Struct Multidiscip Optim 38:53–70
31. Ha S-H, Choi K, Cho S (2010) Numerical method for shape optimization using T-spline based isogeometric method. Struct Multidiscip Optim 42:417–428
32. Qian X (2010) Full analytical sensitivities in NURBS based isogeometric shape optimization. Comput Methods Appl Mech Eng 199(29–32):2059–2071
33. Li K, Qian X (2011) Isogeometric analysis and shape optimization via boundary integral. Comput Aided Des 43(11):1427–1437
34. Azegami H, Fukumoto S, Aoyama T (2013) Shape optimization of continua using NURBS as basis functions. Struct Multidiscip Optim 47:247–258
35. Nagy AP, Abdalla MM, Gürdal Z (2010) Isogeometric sizing and shape optimisation of beam structures. Comput Methods Appl Mech Eng 199(17–20):1216–1230
36. Manh ND, Evgrafov A, Gersborg AR, Gravesen J (2011) Isogeometric shape optimization of vibrating membranes. Comput Methods Appl Mech Eng 200(13–16):1343–1353
37. Nagy AP, Ijsselmuiden ST, Abdalla MM (2013) Isogeometric design of anisotropic shells: optimal form and material distribution. Comput Methods Appl Mech Eng 264:145–162
38. Kiendl J, Schmidt R, Wüchner R, Bletzinger K-U (2014) Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. Comput Methods Appl Mech Eng 274:148–167. https://doi.org/10.1016/j.cma.2014.02.001
39. Qian X, Sigmund O (2011) Isogeometric shape optimization of photonic crystals via coons patches. Comput Methods Appl Mech Eng 200(25–28):2237–2255
40. Seo Y-D, Kim H-J, Youn S-K (2010) Shape optimization and its extension to topological design based on isogeometric analysis. Int J Solids Struct 47(11–12):1618–1640
41. Qian X (2013) Topology optimization in B-spline space. Comput Methods Appl Mech Eng 265:15–35
42. Hirschler T, Bouclier R, Duval A, Elguedj T, Morlier J (2019) The embedded isogeometric Kirchhoff–Love shell: from design to shape optimization of non-conforming stiffened multipatch structures. Comput Methods Appl Mech Eng 349:774–797
43. Hao P, Wang Y, Jin L, Ma S, Wang B (2023) An isogeometric design-analysis-optimization workflow of stiffened thin-walled structures via multilevel NURBS-based free-form deformations (MNFFD). Comput Methods Appl Mech Eng 408:115936
44. Zhao H, Liu X, Fletcher AH, Xiang R, Hwang JT, Kamensky D (2022) An open-source framework for coupling non-matching isogeometric shells with application to aerospace structures. Comput Math Appl 111:109–123. https://doi.org/10.1016/j.camwa.2022.02.007
45. Schillinger D, Ruthala PK, Nguyen LH (2016) Lagrange extraction and projection for NURBS basis functions: a direct link between isogeometric and standard nodal finite element formulations. Int J Numer Methods Eng 108(6):515–534. https://doi.org/10.1002/nme.5216
46. Borden MJ, Scott MA, Evans JA, Hughes TJR (2011) Isogeometric finite element data structures based on Bézier extraction of NURBS. Int J Numer Methods Eng 87:15–47
47. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR (2011) Isogeometric finite element data structures based on Bézier extraction of T-splines. Int J Numer Methods Eng 88:126–156
48. Fromm JE, Wunsch N, Xiang R, Zhao H, Maute K, Evans JA, Kamensky D (2023) Interpolation-based immersed finite element and isogeometric analysis. Comput Methods Appl Mech Eng 405:115890
49. Kamensky D, Bazilevs Y (2019) tIGAr: Automating isogeometric analysis with FEniCS. Comput Methods Appl Mech Eng 344:477–498. https://doi.org/10.1016/j.cma.2018.10.002
50. Logg A, Mardal K-A, Wells GN et al (2012) Automated solution of differential equations by the finite element method. Springer, Geneva
51. Stolarski H, Belytschko T (1982) Membrane locking and reduced integration for curved elements. J Appl Mech 49(1):172
52. Greco L, Cuomo M, Contrafatto L (2018) A reconstructed local B formulation for isogeometric Kirchhoff–Love shells. Comput Methods Appl Mech Eng 332:462–487
53. Bieber S, Oesterle B, Ramm E, Bischoff M (2018) A variational method to avoid locking-independent of the discretization scheme. Int J Numer Methods Eng 114(8):801–827
54. Casquero H, Golestanian M (2022) Removing membrane locking in quadratic NURBS-based discretizations of linear plane Kirchhoff rods: CAS elements. Comput Methods Appl Mech Eng 399:115354
55. Casquero H, Mathews KD (2023) Overcoming membrane locking in quadratic NURBS-based discretizations of linear Kirchhoff–Love shells: CAS elements. Comput Methods Appl Mech Eng 417:116523
56. Greco L, Cuomo M, Contrafatto L, Gazzo S (2017) An efficient blended mixed B-spline formulation for removing membrane locking in plane curved Kirchhoff rods. Comput Methods Appl Mech Eng 324:476–511
57. Nguyen T-H, Hiemstra RR, Schillinger D (2022) Leveraging spectral analysis to elucidate membrane locking and unlocking in isogeometric finite element formulations of the curved Euler-Bernoulli beam. Comput Methods Appl Mech Eng 388:114240
58. Sauer RA, Zou Z, Hughes TJ (2023) A simple and efficient hybrid discretization approach to alleviate membrane locking in isogeometric thin shells. arXiv preprint arXiv:2312.16944
59. Paviot T, Feringa J (2018) Pythonocc. Technical report, 3D CAD/CAE/PLM development framework for the Python programming language
60. Gandarillas V, Joshy AJ, Sperry MZ, Ivanov AK, Hwang JT (2022) A graph-based methodology for constructing computational models that automates adjoint-based sensitivity analysis. Struct Multidiscip Optim **(submitted for publication)**
61. Kraft D (1988) A software package for sequential quadratic programming. Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt
62. Gill PE, Murray W, Saunders MA (2005) SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev 47(1):99–131
63. Gray JS, Hwang JT, Martins JR, Moore KT, Naylor BA (2019) Openmdao: an open-source framework for multidisciplinary design, analysis, and optimization. Struct Multidiscip Optim 59:1075–1104
64. https://github.com/hanzhao2020/GOLDFISH: GOLDFISH source code
65. https://openmdao.org/newdocs/versions/latest/examples/beam_optimization_example.html: Optimizing the Thickness Distribution of a Cantilever Beam Using the Adjoint Method
66. COMSOL Multiphysics Reference Manual, Version 5.6. COMSOL, Inc. www.comsol.com
67. Bathe K-J, Dvorkin EN (1986) A formulation of general shell elements-the use of mixed interpolation of tensorial components. Int J Numer Methods Eng 22(3):697–722
68. Liu W, Deng Z, Li J, Hu X (2022) Investigating the electrothermal behavior of evtol batteries in urban air mobility applications. In: 2022 IEEE 25th international conference on intelligent transportation systems (ITSC), Macau, China, pp 40–45. https://doi.org/10.1109/ITSC55140.2022.9922362

69. Polaczyk N, Trombino E, Wei P, Mitici M (2019) A review of current technology and research in urban on-demand air mobility applications. In: Vertical flight society autonomous VTOL technical meeting and Electric VTOL symposium, Mesa, USA
70. van Schie SP, Zhao H, Yan J, Xiang R, Hwang JT, Kamensky D (2023) Solver-independent aeroelastic coupling for large-scale multidisciplinary design optimization. In: AIAA Scitech 2023 Forum, National Harbor, USA, p 0727

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.