Computer-Aided Design 178 (2025) 103810

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

# Minimal surface-guided higher-order mesh generation for CAD models

Kaixin Yu [a,b], Bohan Wang [a], Xuejuan Chen [a], Ying He [b], Jianjun Chen [a,c,*]

[a] *School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, 310027, China*
[b] *College of Computing and Data Science, Nanyang Technological University, Singapore*
[c] *State Key Lab of CAD&CG, Zhejiang University, Hangzhou, 310058, China*

## ARTICLE INFO

*Keywords:*
Higher-order mesh generation
Curvilinear mesh
Mesh optimization
Minimal surface

## ABSTRACT

This paper presents a novel method for generating higher-order meshes for CAD surfaces by leveraging minimal surface theory to improve element shapes. We explore the concept of higher-order mesh distortion through deformation gradients and introduce an energy function designed to minimize the surface area of these meshes, providing a theoretical justification for its effectiveness in untangling. The process of mesh generation starts with segmenting CAD surfaces into linear elements, followed by the insertion of higher-order nodes within these elements. These nodes are then projected onto the CAD surface to form the initial higher-order elements. By optimizing energy functions related to minimal surfaces and the projection distances, we achieve high-quality, geometrically accurate higher-order surface meshes. Our method has been validated on complex geometries, showcasing its potential in creating effective higher-order meshes for industrial CAD models.

## 1. Introduction

Over the past two decades, higher-order computational fluid dynamics (CFD) techniques have garnered significant attention. Higher-order numerical methods potentially offer greater accuracy, faster convergence, and reduced computational costs compared to traditional lower-order algorithms.

Currently, most of the CFD solver development is focused on lower-order techniques (first or second-order). However, many complex flow problems exceed the capabilities of lower-order methods. For example, airflow-generated tip vortices significantly impact the rotor's aerodynamic loads on helicopters [1]. Lower-order methods require a fine mesh to achieve the desired accuracy, while higher-order methods can avoid this issue. This occurs as the dispersion and dissipation of higher-order methods are much lower than those of lower-order methods, making them preferable for wave propagation problems [2]. Recently, methods like the discontinuous Galerkin technique have advanced significantly in fields such as computational acoustics [3–5], computational electromagnetics [6,7], solid mechanics [8,9], and computational fluid dynamics [10–12]. The classical lower-order methods provided by most existing industrial codes are increasingly unable to meet the requirements of high-precision and large-scale numerical simulations. Therefore, higher-order methods are likely to become a potential direction for the future development of CFD.

For many years, the development of automated higher-order mesh generation has been a challenging and unresolved problem. This is a major impediment that has prevented the wider adoption of these techniques. Two main approaches exist for generating higher-order meshes: direct [13–19] and indirect methods [2,20–29]. Direct methods generate higher-order meshes directly within the specified region. Conversely, indirect methods initiate with a foundational linear mesh, followed by the insertion of higher-order nodes, which are then projected onto the CAD surfaces. Subsequent optimizations or deformations are then applied to enhance the overall injectivity and quality of the higher-order mesh. Due to its high efficiency, the indirect method has become the predominant approach for higher-order mesh generation. In the past two decades, remarkable progress has been made in higher-order volume mesh generation with the indirect method. Examples include methods based on solid mechanics [2,22,23,25,29], Winslow equations [27], local remeshing operator [30,31], radius basic function [26], based on the optimization method [32,33], etc.

The traditional indirect method struggles with intricate geometries, particularly in high-curvature regions and near irregular boundaries. For example, as shown in Fig. 1, when the curvature of the CAD surface is large, the initial linear mesh is unable to capture the curvature characteristics of these areas. If the nodes of the higher-order mesh are simply projected onto the CAD surface, such a simple strategy lead to an uneven distribution of nodes, resulting in distorted and tangled elements. Such erroneous higher-order surface meshes cause great trouble in the subsequent numerical calculation based on higher-order surface meshes and higher-order volume mesh generation. This issue has not been given enough attention in previous studies, only a

---

[*] Corresponding author.
*E-mail address:* chenjj@zju.edu.cn (J. Chen).

https://doi.org/10.1016/j.cad.2024.103810
Received 17 April 2024; Received in revised form 5 August 2024; Accepted 24 September 2024

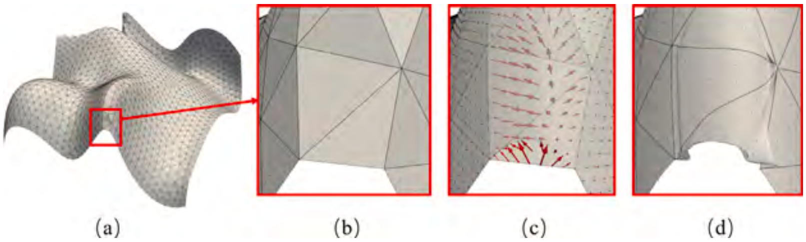


**Fig. 1.** Distortion in higher-order surface meshes due to projection-based algorithms: (a) Initial linear mesh. (b) Higher-order node insertion. (c) Projection onto the CAD surface (indicated by the red arrow). (d) Resulting distorted and tangled higher-order surface mesh. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

few original methods have been proposed to address it, indicating that more research is needed.

Existing methods for addressing such problems can be broadly divided into two categories. The first [21,28] is to adjust the position of higher-order element nodes in parameter space and map them back to physical space, Sherwin established a spring system in the parameter domain and minimizing the elastic potential energy of the system to optimize the quality of the higher-order mesh [28]. And Gargallo proposed a method to measure the distortion between the higher-order surface mesh and the original CAD surface, and applied this method to the generation of higher-order surface mesh [21]. The second is to adjust the position of nodes on higher-order elements directly in physical space, in [22], where higher-order surface mesh nodes are placed using geodesics on the CAD surface. In Ref. [24,34,35], the coordinates of the higher-order mesh node points are optimized by minimizing the elastic energy.

A major challenge in constructing higher-order meshes for CAD surfaces is ensuring mesh quality while maintaining the precise placement of nodes on the surface. Despite theoretical assertions that mesh generation techniques should be independent of the choice of parameterization [21], practical implementations often experience issues related to the quality of parameterization. These issues can lead to instability, particularly when handling singularities on CAD surfaces where parameterization may become significantly distorted. Moreover, there are instances where the parameterization of the CAD surfaces are not readily available [36], making the optimization of higher-order elements in the physical space a viable alternative approach.

In this paper, we introduce the concept of minimal surfaces [37–39] as a means to optimize tangled phenomena in higher-order surface meshes. To the best of our knowledge, this method has not been explored in this field before. And our research demonstrates a significant link between minimal surface theory and higher-order surface mesh. Additionally, we employ an iterative approach wherein the coordinates of higher-order nodes are adjusted in physical space. This adjustment process aims to simultaneously minimize the energy associated with minimal surfaces and ensure that the nodes remain on the original CAD surface through projection. Notably, our method operates independently of CAD surface parameterization, serving as a complement to purely parametric methods.

Our contributions include:

(1) Proposing an innovative theoretical framework for assessing the distortion of higher-order surface meshes. We establish a connection between deformation gradients and scaled Jacobians by analyzing surfaces' local differential properties.

(2) Introducing a novel energy function tailored for the creation of higher-order meshes. Minimizing this energy function allows for guiding the higher-order mesh towards a minimal surface, effectively reducing mesh distortion and achieving a streamlined outcome. Furthermore, we demonstrate theoretically that minimal surfaces can eliminate the tangle phenomenon in higher-order meshes.

(3) We propose a novel method for creating high-quality, higher-order surface meshes. Beyond the traditional initial mesh generation processes such as nodes insertion and projection, we incorporate arc-length parameterization and geodesic algorithms to redistribute nodes along the boundaries of the higher-order mesh. Subsequently, an iterative strategy is employed to minimize the energy associated with the mesh surfaces and projection distances, thus optimizing the layout of internal nodes. Lastly, element fusion techniques are applied to optimize shared boundaries, effectively addressing localized distortions induced by geodesics.

This paper is organized as follows. Section 2 gives a summary of the methods covered in this paper. Section 3 explains the pointwise distortion measure of higher-order meshes and the concept of minimal surface. Section 4 introduces the procedure for optimizing nodes in the generation of higher-order surface meshes. The assessment of our algorithm using intricate industrial examples is described in Section 5, followed by a recap of the contributions in Section 6.

## 2. Overview

*Input*: Our algorithm takes a parameterized CAD surface as its input.

*Output*: The result of our algorithm is a finished higher-order surface mesh, represented by $\{T_H^p\}_{i=1}^{N_e}$, where $p$ is the order of the elements and $N_e$ is the number of elements.

The overall steps of our algorithm are as follows:

1. *Generating linear elements*: Employing the current linear mesh generator, we acquire an initial linear triangular mesh assisted by the size field [40,41].

2. *Generating initial higher-order elements*: The order $p$ of higher-order elements needed for mesh generation dictates the insertion of higher-order mesh nodes into the linear mesh. These nodes are then projected onto the CAD surface to establish the initial higher-order mesh. However, this initial mesh frequently presents issues such as self-intersections between elements, necessitating optimization for resolution.

3. *Classifying the higher-order nodes*: In the context of the higher-order mesh obtained from the previous step, we categorize the nodes of the elements into four groups: corner nodes $\{\boldsymbol{P}_C\}$, boundary nodes $\{\boldsymbol{P}_L\}$, edge nodes $\{\boldsymbol{P}_B\}$, and interior nodes $\{\boldsymbol{P}_I\}$. Nodes in $\{\boldsymbol{P}_C\}$ are positioned at the corners of $T_H^p$, aligning with the vertices of the initial linear triangle. $\{\boldsymbol{P}_L\}$ represents nodes positioned along the boundary curve of the CAD surface. $\{\boldsymbol{P}_B\}$ denotes nodes located along the edges of $T_H^p$, while $\{\boldsymbol{P}_I\}$ refers to nodes within $T_H^p$. Furthermore, for each higher-order element $T_H^p$, a local subdivision $T_S^p$ is generated, which plays a pivotal role in subsequent optimization processes.

4. *Optimizing the coordinates of the higher-order nodes*: Subsequently, utilizing the aforementioned categorization, we optimize the nodes of the higher-order mesh in the following sequence: firstly, optimize $\{\boldsymbol{P}_L\}$ based on arc length; second, optimize $\{\boldsymbol{P}_B\}$ regarding geodesics; third, optimize $\{\boldsymbol{P}_I\}$ according to minimal surface theory; finally, perform an operation termed element fusion to optimize $\{\boldsymbol{P}_B\}$ and $\{\boldsymbol{P}_I\}$ within specific local regions.

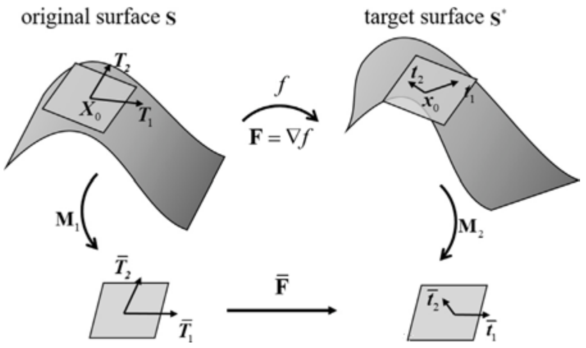


**Fig. 2.** Notation for the deformation mapping between the original and target surfaces.

## 3. Distortion measure

### 3.1. Define the deformation gradient between two 3D surfaces

We define the deformation gradient between two surfaces as a tensor that quantifies the local differential transformation of an original surface $\mathbf{S}$ into a target surface $\mathbf{S}^*$. This process is formalized through a differentiable mapping $f : \mathbf{S} \rightarrow \mathbf{S}^*$, where a point $\boldsymbol{X} \in \mathbf{S}$ is mapped to a point $\boldsymbol{x} = f(\boldsymbol{X}) \in \mathbf{S}^*$. The deformation gradient $\mathbf{F}$ at $\boldsymbol{X}$ is given by the Jacobian matrix of $f$ at $\boldsymbol{X}$ (see Fig. 2):

$$\mathbf{F} = \nabla f(\boldsymbol{X}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}.$$

In the local tangent plane of $\boldsymbol{X}$, this gradient transforms the tangent vectors of $\mathbf{S}$ to those of $\mathbf{S}^*$. If $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are tangent vectors at $\boldsymbol{X}$, and $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are their images under deformation, then the action of $\mathbf{F}$ is described by:

$$\mathbf{F}\boldsymbol{T}_i = \boldsymbol{t}_i, \quad i \in \{1, 2\}.$$

Considering the complexity of defining $\mathbf{F}$ in three dimensions, we project these vectors onto the tangent plane, leading to a two-dimensional representation. Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be rotation matrices that map the 3D tangent vectors to their 2D counterparts in their respective tangent planes:

$$\mathbf{M}_1\boldsymbol{T}_i = \overline{\boldsymbol{T}}_i, \quad \mathbf{M}_2\boldsymbol{t}_i = \overline{\boldsymbol{t}}_i, \quad i \in \{1, 2\}.$$

The two-dimensional deformation gradient $\overline{\mathbf{F}}$ relates these planar tangent vectors as:

$$\overline{\mathbf{F}\boldsymbol{T}}_i = \overline{\boldsymbol{t}}_i, \quad i \in \{1, 2\}.$$

Hence, $\overline{\mathbf{F}}$ can be expressed as:

$$\overline{\mathbf{F}} = \mathbf{D}_t\mathbf{D}_o^{-1},$$

where $\mathbf{D}_o$ and $\mathbf{D}_t$ are matrices formed by column vectors $\overline{\boldsymbol{T}}_i$ and $\overline{\boldsymbol{t}}_i$, respectively.

The determinant of $\overline{\mathbf{F}}$ encapsulates the local areal change and orientation inversion caused by deformation:

$$\det(\overline{\mathbf{F}}) = k\frac{\|\overline{\boldsymbol{t}}_1 \times \overline{\boldsymbol{t}}_2\|}{\|\overline{\boldsymbol{T}}_1 \times \overline{\boldsymbol{T}}_2\|} = k\frac{\|\mathbf{M}_2\boldsymbol{t}_1 \times \mathbf{M}_2\boldsymbol{t}_2\|}{\|\mathbf{M}_1\boldsymbol{T}_1 \times \mathbf{M}_1\boldsymbol{T}_2\|} = k\frac{\|\boldsymbol{t}_1 \times \boldsymbol{t}_2\|}{\|\boldsymbol{T}_1 \times \boldsymbol{T}_2\|}, \tag{1}$$

with $k = \pm 1$ signifying the preservation or reversal of orientation. A negative determinant implies that the local surface orientation is tangled post-deformation.

### 3.2. Distortion measure between the higher-order mesh and the CAD surface

As illustrated in Fig. 3, for a higher-order element, the mapping between the reference and physical elements is formalized as follows:

$$\phi_1 : \widetilde{T_H^p} \subset \mathbb{R}^2 \longrightarrow T_H^p \subset \mathbb{R}^3,$$

$$\boldsymbol{\zeta}(\xi, \eta) \xmapsto{\phi_1} \boldsymbol{X}(x, y, z),$$

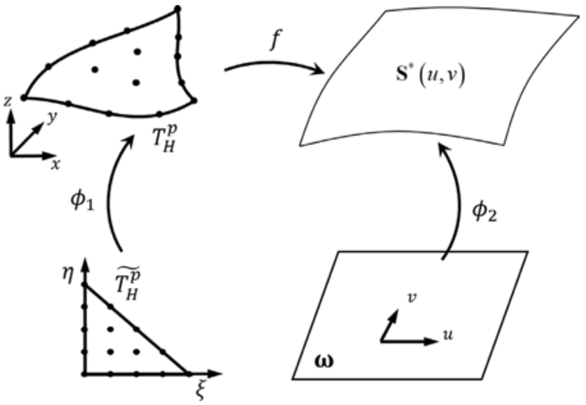


**Fig. 3.** Illustration of the mapping between a higher-order element and CAD surface.

where $\boldsymbol{\zeta}(\xi, \eta)$ denotes the local coordinates in the reference element domain, and $\boldsymbol{X} = \sum_{i=1}^{n_p} \boldsymbol{X}_i L_i^p(\boldsymbol{\zeta})$ represents the physical coordinates, with $n_p$ being the number of nodes in the element and $L_i^p(\boldsymbol{\zeta})$ the Lagrange shape functions for interpolation.

Transitioning to the CAD surface, the mapping of points from the parametric to the physical domain is given by:

$$\phi_2 : \boldsymbol{\omega} \subset \mathbb{R}^2 \longrightarrow \mathbf{S}^* \subset \mathbb{R}^3,$$

$$\boldsymbol{\sigma}(u, v) \xmapsto{\phi_2} \boldsymbol{x}(x, y, z),$$

indicating a transformation from parametric space $\boldsymbol{\omega}$ to the CAD surface $\mathbf{S}^*$.

Here, the original surface is represented by the higher-order element $T_H^p$, while the target surface is denoted by the CAD surface $\mathbf{S}^*$. According to Eq. (1), the determinant of the point-wise deformation gradient between these two surfaces is given as:

$$\det(\overline{\mathbf{F}}) = k\frac{\|\boldsymbol{X}_\xi \times \boldsymbol{X}_\eta\|}{\|\boldsymbol{x}_u \times \boldsymbol{x}_v\|}, \tag{2}$$

where $\boldsymbol{X}_\xi$ and $\boldsymbol{X}_\eta$ are the derivatives of the physical coordinates with respect to the local coordinate system, and are calculated as follows:

$$\boldsymbol{X}_\xi = \begin{bmatrix} \sum_{i=1}^{n_p} X_i^1 \frac{\partial L_i^p(\boldsymbol{\zeta})}{\partial \xi} \\ \sum_{i=1}^{n_p} X_i^2 \frac{\partial L_i^p(\boldsymbol{\zeta})}{\partial \xi} \\ \sum_{i=1}^{n_p} X_i^3 \frac{\partial L_i^p(\boldsymbol{\zeta})}{\partial \xi} \end{bmatrix}, \boldsymbol{X}_\eta = \begin{bmatrix} \sum_{i=1}^{n_p} X_i^1 \frac{\partial L_i^p(\boldsymbol{\zeta})}{\partial \eta} \\ \sum_{i=1}^{n_p} X_i^2 \frac{\partial L_i^p(\boldsymbol{\zeta})}{\partial \eta} \\ \sum_{i=1}^{n_p} X_i^3 \frac{\partial L_i^p(\boldsymbol{\zeta})}{\partial \eta} \end{bmatrix},$$

and $\boldsymbol{x}_u$, $\boldsymbol{x}_v$ are the corresponding derivatives in the CAD surface model.

For a higher-order element, the quality measure, known as the scaled Jacobian [13,21,22,42], is defined as:

$$J_S = \frac{\min \det(\overline{\mathbf{F}})}{\max \det(\overline{\mathbf{F}})}. \tag{3}$$

$J_S$ is evaluated based on a fixed number of discrete locations within the element domain and checking for validity is expensive [42]. In this paper, $J_S$ is calculated at specific points on $\widetilde{T_H^p}$, which are chosen to match the integration points of a quadrature rule of order $2p$ for higher-order elements of order $p$ [13,22].

Referring to Eqs. (2) and (3), we identify two methodologies to diminish distortion in higher-order meshes.

Firstly, minimizing the discrepancy between the differential areas of $T_H^p$ and $\mathbf{S}^*$, represented by $\|\boldsymbol{X}_\xi \times \boldsymbol{X}_\eta\|$ and $\|\boldsymbol{x}_u \times \boldsymbol{x}_v\|$, can be achieved through energy function optimization in $\mathbf{S}^*$'s parameter space. However, this method is highly dependent on the parameterization of the surface $\mathbf{S}^*$.

Alternatively, ensuring consistent local differential area $\|\boldsymbol{X}_\xi \times \boldsymbol{X}_\eta\|$ across $T_H^p$, while maintaining its nodes with $\mathbf{S}^*$, provides a more stable and geometry-preserving approach. This strategy reduces reliance on

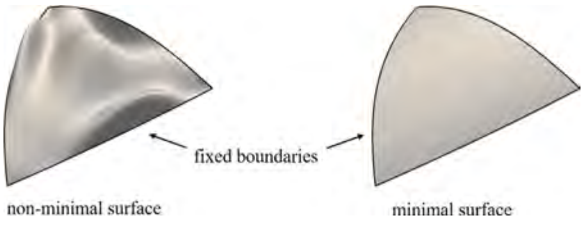


**Fig. 4.** Illustration of the difference between minimal and non-minimal surface.

the parameterization of $\mathbf{S}^*$, enhancing robustness, especially in the presence of singularities or complex parameter spaces. This paper adopts this latter approach for its reliability and adaptability.

### 3.3. Continuous energy functional for attaining minimal surface

Observations suggest that tangling on a surface often leads to an increase in the total surface area, deviating from the concept of a minimal surface. This phenomenon arises due to local inversions that introduce wrinkles or folds, thereby expanding the area relative to an untangled surface. As depicted in Fig. 4, the non-minimal surface exhibits noticeable tangling, resulting in a larger surface area compared to the minimal surface.

We define a functional space $\mathbb{S}$ to mathematically address the untangling process:

$$\mathbb{S} = \left\{ \mathbf{S} \in C^2(\Omega, \mathbb{R}^3) \mid \mathbf{S}_\xi \times \mathbf{S}_\eta \neq 0, \mathbf{S}_{|\partial\Omega} = L \right\},$$

where $\mathbf{S}(\xi, \eta) : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ represents surfaces within this space. To further our analysis, we introduce the following theorem concerning minimal surfaces.

**Theorem 1.** *Let $\rho : \Omega \to \mathbb{R}$ be a scalar function defined over the domain $\Omega$. The energy functional $E(\mathbf{S})$ is defined as:*

$$E(\mathbf{S}) = \int_\Omega \rho \|\mathbf{S}_\xi \times \mathbf{S}_\eta\|^2 d\xi d\eta. \tag{4}$$

*At the critical points of $E(\mathbf{S})$, the surface $\mathbf{S}$ tends towards a minimal surface, while maintaining $\rho\|\mathbf{S}_\xi \times \mathbf{S}_\eta\|$ constant across $\Omega$ [39].*

**Corollary 1.** *Minimizing $E(\mathbf{S})$ for a higher-order element $T_H^p$ with fixed boundaries results in a minimal surface, under the condition that $\rho\|\mathbf{S}_\xi \times \mathbf{S}_\eta\|$ remains constant over $\Omega$. This constraint ensures untangled higher-order mesh generation.*

**Proof of Corollary 1.** If tangling occurs on any part of the element $T_H^p$, it implies that the normals in that region would orient oppositely, resulting in a change in the quantity $\rho\|\mathbf{S}_\xi \times \mathbf{S}_\eta\|$. However, the stipulation is that $\rho\|\mathbf{S}_\xi \times \mathbf{S}_\eta\|$ remains unchanged. Thus, $T_H^p$ remains untangled. ☐

In practical applications, achieving an exact minimal surface for higher-order elements is hindered by several factors:

(1) The boundaries of the higher-order mesh must adhere to the CAD surface, which prevents them from being simple straight lines.

(2) A precise higher-order nodes distribution must be maintained over the CAD surface's geometry to ensure the mesh's geometric accuracy.

(3) Due to the lack of an analytical solution for minimal surfaces, reliance on numerical methods is necessary. However, achieving a perfect minimal surface is challenging because of the limited precision of numerical methods.

Therefore, the aim of the algorithm proposed in this paper is to make higher-order surface meshes approach minimal surfaces as closely as possible while considering these three constraints.

## 4. Methodology

### 4.1. Optimization of the boundary nodes $\{\boldsymbol{P}_L\}$

Optimal node placement in one-dimensional higher-order meshes is achieved through arc length parameterization. Let $\xi_i$ (for $i = 0, 1, \ldots, p$) represent the node distribution along the boundary of the reference element. This distribution is essential for defining the coordinate distribution in the physical space. A fundamental relationship guiding this distribution is expressed as follows:

$$\operatorname{arc}\left(\boldsymbol{X}_p\right) - \operatorname{arc}\left(\boldsymbol{X}_0\right) = \frac{\operatorname{arc}\left(\boldsymbol{X}_{i+1}\right) - \operatorname{arc}\left(\boldsymbol{X}_i\right)}{\xi_{i+1} - \xi_i}.$$

Here, the function $\operatorname{arc}(\cdot)$ takes a three-dimensional coordinate as input and yields the arc length along the boundaries of the CAD surface corresponding to that coordinate.

This approach uses CAD surface boundary arc lengths to determine the relative spacing of higher-order nodes. Adhering to this relation optimizes node distribution along the boundary, enhancing the model's accuracy and compatibility with the CAD geometry.

### 4.2. Optimization of the edge nodes $\{\boldsymbol{P}_B\}$

For distributing higher-order nodes on element boundaries, we use an optimization function treating the connections between adjacent higher-order nodes as springs:

$$E_e = \sum_{i=1}^{p-1} \frac{\left(\boldsymbol{X}_{i+1} - \boldsymbol{X}_i\right)^2}{\xi_{i+1} - \xi_i}.$$

Fig. 5(a) illustrates the optimization process, where minimizing the energy function makes the solution converge to the geodesic between two endpoints on the CAD surface. Optimizing each edge of the higher-order mesh yields a geodesic triangle with minimal boundary distortion. Employing geodesics as boundary curves for higher-order meshes is a reasonable approach, given their characteristic of being the straightest curves on a CAD surface or curves with minimal distortion.

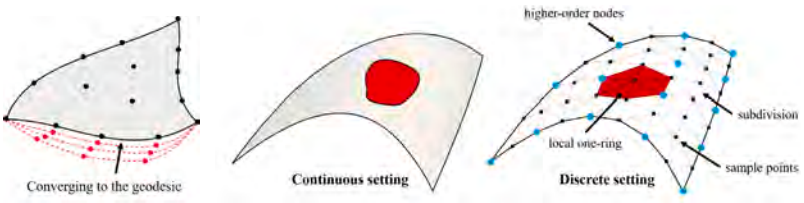
However, this method focuses only on minimizing curve curvature, ignoring other factors. It should be noted that geodesics may not always represent the optimal solution, especially when dealing with surfaces that possess highly irregular boundaries. In such cases, geodesics might be placed too close to the boundary curves of the CAD surface, resulting in undesirable distortion and compression of the elements within that particular region. We introduce 'element fusion' operation to address this issue, detailed in Section 4.4.

### 4.3. Optimization of the interior nodes $\{\boldsymbol{P}_I\}$

#### 4.3.1. Design of practical energy function

From this point forward, the boundary curves of the higher-order mesh have been essentially adjusted (excluding the element fusion operation to be introduced later). We aim to achieve a nearly minimal surface of the higher-order mesh by adjusting $\{\boldsymbol{P}_I\}$ to minimize the energy functional in Eq. (4).

We recognize that the energy functional represents a weighted area integral of the higher-order element, carrying significant geometric implications. However, due to the inherent nonlinearity of the surface, directly minimizing this integral presents challenges. To address this, we adopt a practical engineering approach by using piecewise linear approximation techniques for this continuous nonlinear integral. The initial step involves constructing a structured triangular mesh for a $p$-order element, which is then refined by setting a subdivision count $\alpha$. The choice of $\alpha$ affects the approximation accuracy of the continuous integral, with a higher $\alpha$ yielding a more precise approximation. Essentially, we obtain a subdivision for each $T_H^p$, denoted as ${T_S}_{i=1}^{n_s}$, where $n_s = 4^\alpha p^2$, as shown in Fig. 5(b). (In our implementation, we initially set $\alpha = 0$. If the subdivision count is insufficient and leads



(a) (b)

**Fig. 5.** (a) The adjustment process of the edge nodes $\boldsymbol{P}_B$ in higher-order surface meshes involves iterative steps that incrementally align with the geodesics on the CAD surfaces. The red dashed line illustrates this iterative process. (b) Transformation of a higher-order mesh from continuous to discrete setting to facilitate minimal surface computation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to a failure in untangling, we will gradually increase the subdivision count $\alpha$.) Directly minimizing the energy functional globally remains challenging. Instead, we adopt a local approach by considering the neighborhood of a sample point $\boldsymbol{X}_i$ on the subdivision. We estimate the local surface area by considering the average area in the local one-ring of $\boldsymbol{X}_i$. Denoting the neighbors around point $\boldsymbol{X}_i$ as $\boldsymbol{X}_{ij}$ $(j = 1, 2, \dots, 6)$, the energy function can be expressed as (inspired by [39]):

$$E_i(\boldsymbol{X}_i) = \frac{1}{2}\sum_{j=1}^{6} w_{ij} \left\| A_{ij}(\boldsymbol{X}_i) \right\|^2 . \tag{5}$$

In Eq. (5), $A_{ij}(\boldsymbol{X}_i)$ represents the area of the local triangle formed by point $\boldsymbol{X}_i$ and its adjacent vertices $\boldsymbol{X}_{ij}$ and $\boldsymbol{X}_{i,j+1}$. The weights $w_{ij}$ control how the area is measured. When $w_{ij} = 1$, minimizing the functional energy tends to equalize the area of the local average region. Alternatively, if $w_{ij} = \frac{1}{|\boldsymbol{e}_{ij}|}$, where $\boldsymbol{e}_{ij} = \boldsymbol{X}_{i,j+1} - \boldsymbol{X}_{ij}$, the functional energy tends to equalize the angle of the local average region.

Meanwhile, to further simplify the problem, the movement of $\boldsymbol{X}_i$ is restricted to the tangent plane of the higher-order element. Let us denote the initial position of $\boldsymbol{X}_i$ as $\boldsymbol{X}_i^{(0)}$. The two tangent vectors of the higher-order element are denoted as $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$. Consequently, the feasible range of motion for the point $\boldsymbol{X}_i$ can be expressed as the linear combination of $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$:

$$\boldsymbol{X}_i = \boldsymbol{X}_i^{(0)} + s_1\boldsymbol{T}_1 + s_2\boldsymbol{T}_2 .$$

This approach simplifies the minimization of Eq. (5) to the identification of an optimal point within the tangent plane. This point should be positioned in a manner that ensures the variation of the weighted area of local one-ring is minimal. The gradient of the energy functional is computed as follows:

$$\begin{aligned}\nabla E_i(\boldsymbol{X}_i) &= \sum_{j=1}^{6} w_{ij} A_{ij}(\boldsymbol{X}_i) \times \boldsymbol{e}_{ij} \\ &= \sum_{j=1}^{6} w_{ij} \left[\boldsymbol{e}_{ij} \times (\boldsymbol{X}_i - \boldsymbol{X}_{ij}) \times \boldsymbol{e}_{ij}\right] .\end{aligned}$$

To minimize the variation in area, it is necessary to completely eliminate the tangential component of $\nabla E_i(\boldsymbol{X}_i)$, ensuring that the gradient is parallel to the normal of the surface. Therefore, by taking the inner product of $\nabla E_i(\boldsymbol{X}_i)$ and the two tangent vectors $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$, we can formulate a system of equations:

$$\begin{aligned}&\sum_{j=1}^{6} w_{ij} \left\langle \left[\boldsymbol{e}_{ij} \times (\boldsymbol{X}_i^{(0)} + s_1\boldsymbol{T}_1 + s_2\boldsymbol{T}_2 - \boldsymbol{X}_{ij}) \times \boldsymbol{e}_{ij}\right], \boldsymbol{T}_1 \right\rangle = 0, \\ &\sum_{j=1}^{6} w_{ij} \left\langle \left[\boldsymbol{e}_{ij} \times (\boldsymbol{X}_i^{(0)} + s_1\boldsymbol{T}_1 + s_2\boldsymbol{T}_2 - \boldsymbol{X}_{ij}) \times \boldsymbol{e}_{ij}\right], \boldsymbol{T}_2 \right\rangle = 0.\end{aligned}$$

Solving this system yields coefficients $s_1$ and $s_2$, pinpointing the location on the tangent plane that minimizes local energy. Summing the energy function for all internal points converts the continuous integral into a discrete form:

$$E_{\Sigma} = \sum_{i=1}^{n_I} E_i(\boldsymbol{X}_i) = \frac{1}{2}\sum_{i=1}^{n_I}\sum_{j=1}^{6} w_{ij} \| A_{ij}(\boldsymbol{X}_i) \|^2 . \tag{6}$$

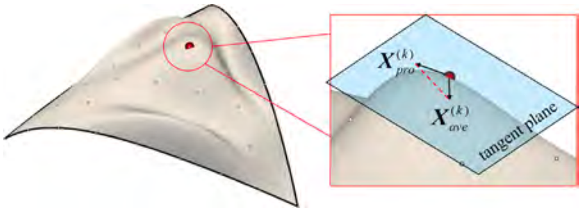


**Fig. 6.** Iterative algorithm stages involve identifying the coordinate $\boldsymbol{X}_{ave}^{(k)}$ on the tangent plane that conforms to the minimal surface, projecting it onto the CAD surface to obtain $\boldsymbol{X}_{pro}^{(k)}$, and computing each iteration's coordinate using a varying decay factor, given by $\boldsymbol{X}_i^{(k)} = \beta^{(k)}\boldsymbol{X}_{ave}^{(k)} + (1-\beta^{(k)})\boldsymbol{X}_{pro}^{(k)}$.

To minimize the energy of the entire higher-order element, we utilize a method to progressively reduce the energy of each internal vertex through iterations. This iterative process enables us to achieve the minimum value of the total discrete energy as described in Eq. (6). In this equation, $n_I$ denotes the quantity of internal sampling points. Notwithstanding potential overlaps in area computation, as subdivision number $\alpha$ increases, the minimization of $\lim_{n_I \to \infty} E_{\Sigma}$ becomes equally significant as the minimization of the continuous integral detailed in Eq. (4), as the sampling density increases.

In addition, to ensure sufficient geometric accuracy, we must place the nodes of the higher-order mesh on the CAD surface. This implies that we also need to minimize the following energy:

$$E_{pro} = \sum_{i=1}^{n_I} \left\| \boldsymbol{X}_i - \mathbf{S}^*(u, v) \right\|^2 . \tag{7}$$

### 4.3.2. *Iterative algorithm for energy minimizing*

In an ideal situation, higher-order elements would efficiently adjust and reorganize their nodes to decrease energy while ensuring that they are sufficiently close to the CAD surface for accurate geometric representation. However, in reality, it is difficult to minimize both energy functions simultaneously. One of the main challenges stems from the conflicting objectives of the two functions. The minimization of Eq. (6) focuses on reducing surface wrinkles to achieve a result of minimal area, while minimizing the projection energy (Eq. (7)) aims to decrease the distance between higher-order nodes and the CAD surface. Minimizing one function may potentially increase the other, and vice versa. Therefore, we suggest an iterative algorithm, outlined in Algorithm 1, to tackle this issue.

The process starts by projecting all internal points of higher-order elements onto the CAD surface, which initially distorts the elements. For the decay factor in our algorithm, we employ the following function [36,43]:

$$\lambda = \frac{k-1}{k_{\max}-1}, \quad \beta(\lambda) = \frac{(1-\lambda)^2}{\lambda^2 + (1-\lambda)^2}.$$

Here, $\lambda$ represents the iterative parameter. Meanwhile, for the optimization of interior points in a single higher-order mesh, we set all the $w_{ij}$ to 1.

**Algorithm 1** Iterative algorithm for higher-order elements with fixed boundary.

```
Input: Set of points $\{\boldsymbol{X}_i\}$ with $n^I$ points, number of iterations $k_{\max}$, convergence thresholds $\epsilon_\Sigma$ and $\epsilon_{\text{pro}}$
Output: Optimized set of interior node positions $\{\boldsymbol{X}_i\}$
function IterativeOptimization($\{\boldsymbol{X}_i\}$, $k_{\max}$, $\epsilon_\Sigma$, $\epsilon_{\text{pro}}$)
    for $k = 1 \dots k_{\max}$ do
        $\lambda^{(k)} \leftarrow \frac{k-1}{k_{\max}-1}$
        $\beta^{(k)} \leftarrow \frac{(1-\lambda^{(k)})^2}{(\lambda^{(k)})^2+(1-\lambda^{(k)})^2}$
        for $i = 1 \dots n^I$ do
            $\boldsymbol{X}_{ave}^{(k)} \leftarrow \operatorname{argmin} E_i^{(k)}$
            $\boldsymbol{X}_{pro}^{(k)} \leftarrow \operatorname{argmin} E_{\text{pro}}^{(k)}$
            $\boldsymbol{X}_i^{(k)} \leftarrow \beta^{(k)} \boldsymbol{X}_{ave}^{(k)} + (1-\beta^{(k)}) \boldsymbol{X}_{pro}^{(k)}$
        end for
        Calculate $E_\Sigma^{(k)}$ using Equation (6)
        Calculate $E_{\text{pro}}^{(k)}$ using Equation (7)
        if $|\Delta E_\Sigma^{(k)}| < \epsilon_\Sigma$ and $|\Delta E_{\text{pro}}^{(k)}| < \epsilon_{\text{pro}}$ then
            return Optimized set of $\{\boldsymbol{X}_i\}$
        end if
    end for
end function
```

In the first optimization iteration, $\beta^{(0)} = 1$, so the goal of optimization is to reduce $E_i$. In the following optimization iterations, the energy is divided into two parts: one focusing on minimizing the surface area of the higher-order mesh to obtain the coordinates $\boldsymbol{X}_{\text{ave}}^{(k)}$, and the other concentrating on moving the nodes of the higher-order mesh closer to the CAD surface to obtain the coordinates $\boldsymbol{X}_{\text{pro}}^{(k)}$. Fig. 6 illustrates this process.

As the decay factor decreases, the higher-order mesh approaches the CAD surface, inducing perturbations to minimize the surface. However, each iteration's beneficial impact on the higher-order mesh parameterization remains. This phenomenon can primarily be attributed to the decreased projection distance required to reach the CAD surface at each iteration $k$. The distortion introduced during the projection process depends on this projection distance. Consequently, as the projection distance diminishes, so does the interference introduced by projection on minimizing the energy of minimal surfaces, resulting in a mesh with significantly improved shape. This, in turn, leads to an improvement in the quality of subdivisions, allowing them to more closely approximate the original continuous surface and enhance the quality of their sampling for continuous integration. As a result, with the reduction of this perturbation, each optimization step gradually minimizes the disturbance on the projection distance and the interference on minimizing the energy of minimal surfaces. Through these iterative processes, the algorithm eventually converges.

At $k = k_{\max}$, when $\beta^{(k_{\max})} = 0$, or when the two energies cease to change, we obtain the final higher-order mesh. For our implementation, we used the parameters $k_{\max} = 100$, $\epsilon_\Sigma = 1 \times 10^{-6}$, and $\epsilon_{\text{pro}} = 1 \times 10^{-8}$. Additionally, we only performed iterative optimization on elements with $J_S < 0.7$.

### 4.4. Higher-order element fusion

In previous studies [22,34,35], it has been noted that using geodesic curves on CAD surfaces to distribute the boundary curves of higher-order meshes is a common practice. However, this advantage is limited to the curve itself when considered in isolation. The dynamics change when attempting to incorporate this curve into the shared boundary of two higher-order elements. In cases where geodesic curves are placed close to the boundary of a CAD surface with an inward concave shape, the geodesic curve may encroach too closely on the CAD surface's boundary curve. This proximity can pose challenges for our internal point optimization algorithm to find adequate space for optimization, leading to unnecessary wrinkles in the higher-order elements, as illustrated in Fig. 7(a).

**Algorithm 2** Element Fusion Operation Algorithm

```
Input: Higher-order elements, threshold $\rho$
Output: Optimized higher-order elements
for each higher-order edge $e_k^H$ do
    Extract the two higher-order elements $T_{H1}^p$, $T_{H2}^p$ corresponding to $e_k^H$
    Calculate the minimum $J_S$ for $T_{H1}^p$, $T_{H2}^p$
    if the minimum $J_S < \rho$ then
        Extract and fuse the subdivisions of $T_{H1}^p$, $T_{H2}^p$
        Extract the interior point set of the subdivisions $\{\boldsymbol{X}_i\}$
        call IterativeOptimization($\{\boldsymbol{X}_i\}$, $k_{\max}$, $\epsilon_\Sigma$, $\epsilon_{\text{pro}}$)
        Check if the minimum $J_S$ increases after optimization
        if the minimum $J_S$ increases then
            call IterativeOptimization for $T_{H1}^p$
            call IterativeOptimization for $T_{H2}^p$
        else
            Reverse optimization and restore the elements
        end if
    end if
end for
```

To address these challenges, we introduce a technique called element fusion in this study. The process of element fusion is illustrated in Fig. 7. The specific steps of the algorithm are described in Algorithm 2. First, we iterate through all edges of the higher-order mesh and extract the two higher-order elements corresponding to each edge. If the minimum $J_S$ of these two elements is less than the specified threshold $\rho$ (typically set to 0.4, an empirical parameter), we initiate the element fusion operation. We introduce a help element to guide the formation of this common boundary. Then, as illustrated in Fig. 7(b), we merge the subdivisions of these two higher-order elements.

In Eq. (5), we set $w_{ij}$ to $\frac{1}{|e_{ij}|}$, indicating our objective is to equalize the angles of the subdivisions, thereby making the common boundary smoother. By employing the iterative algorithm mentioned earlier, we minimize this energy function to achieve smooth common boundary curves, as illustrated in Fig. 7(c) and (d).

Once the boundary curves are adjusted, we re-fix the boundaries of the two higher-order meshes and set $w_{ij}$ to 1. Then, we proceed to re-optimize these two elements using Algorithm 1.

Simultaneously, given that the main objective of this procedure is to enhance the minimum quality threshold of the two elements, it is possible that poor elements might see an enhancement in quality while help elements may experience a decline during the optimization process. Consequently, it is essential to verify the enhancement in the minimum quality of elements both before and after fusion following the completion of each operation. If an enhancement is observed, the operation is considered successful; otherwise, it is necessary to return to the initial state.

### 4.5. Higher-order element adaptive refinement

Adjusting the nodes of higher-order meshes can effectively improve mesh quality. However, some CAD models may have very sharp geometric curvature changes, and it may also be difficult to estimate the appropriate density and quality for the initial linear mesh beforehand. In these cases, optimizing node coordinates alone may not be sufficient to generate a high-quality or even an effective higher-order mesh. Therefore, this paper suggests that, in addition to smoothing techniques, topological optimization should be included to further improve the quality of the higher-order mesh.

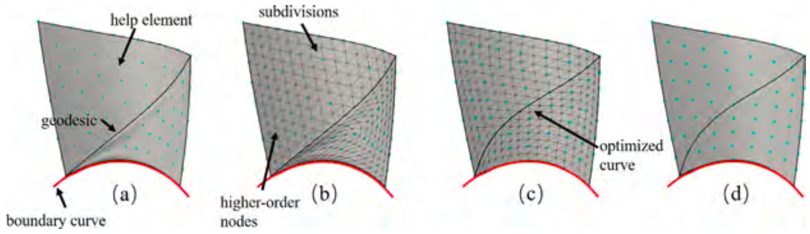


**Fig. 7.** Illustration of the element fusion operation. (a) Distortion in higher-order meshes arises due to the proximity of geodesics to the boundary curves of CAD surface. This is mitigated by introducing help elements to optimize the shared boundary curves between elements. (b) Fusing the subdivisions. (c) The improved shared boundaries of higher-order meshes obtained after optimization using element fusion operations. (d) The final optimized higher-order mesh.

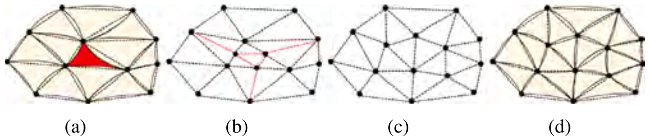


**Fig. 8.** Local adaptive refinement process of higher-order surface mesh. (a) Elements that do not meet the quality requirements are highlighted in red. (b) The local higher-order mesh is reverted to linear elements and the corresponding linear elements are decomposed. (c) The quality of the local linear mesh is improved. (d) The linear mesh is elevated back to the final higher-order mesh . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 1**
Basic geometric information of linear meshes for four models.

| Model | $N_E$ | $N_P$ | $N_S$ |
|---|---|---|---|
| Freeform surface | 2556 | 1361 | 1 |
| Propeller | 6510 | 3679 | 13 |
| Horten | 11 350 | 6602 | 31 |
| DLR-f6 | 15 484 | 8915 | 45 |

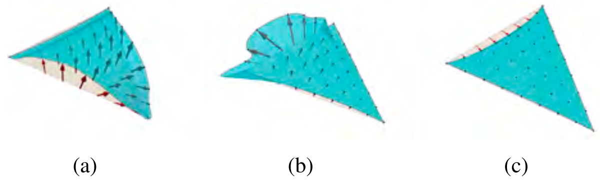


**Fig. 9.** Three types of higher-order elements distortions. The white elements represent initial linear elements, cyan elements represent distorted higher-order elements, and red arrows denote projection operations. (a) High curvatures. (b) Convex boundaries. (c) Concave boundaries. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 2**
Mesh quality of Freeform Surface in different optimization stages.

| Stage | $J_S$ | | |
|---|---|---|---|
| | avg. | min. | max. |
| Initial | 0.849 | −1.798 | 0.996 |
| Stage one | 0.850 | −0.793 | 0.996 |
| Stage two | 0.851 | −0.712 | 0.996 |
| Stage three | 0.868 | 0.153 | 0.996 |
| Stage four | 0.868 | 0.498 | 0.996 |

As demonstrated in Fig. 8, if certain elements do not meet quality requirements after a series of smoothing algorithms, the local adaptive refinement algorithm is initiated. In Fig. 8(a), the highlighted elements signify lower-quality higher-order elements. To refine the higher-order mesh in this region, we revert the higher-order elements in this local area to linear elements and decompose the corresponding low-quality elements, as shown in Fig. 8(b). After decomposition, linear mesh optimization algorithms such as edge swapping, edge collapsing, edge splitting, and smoothing are used to optimize the mesh quality, as shown in Fig. 8(c). Subsequently, the optimized linear mesh in this region is elevated back to a higher-quality higher-order mesh, as shown in Fig. 8(d).

## 5. Numerical experiments

We have selected four examples to test our algorithm: Freeform Surface, Propeller, DLR-f6 aircraft, and Horten. The basic geometric information is provided in Table 1. Here, $N_E$, $N_P$, and $N_S$ represent the number of linear elements, vertices, and surfaces of the model, respectively. The first example is a single-surface case, while the remaining three are complex industrial applications. The original CAD models have regions of high curvature and irregular boundaries, posing challenges in generating quality higher-order surface meshes. All the numerical experiments were performed on a personal computer (AMD Ryzen 5 3600 6-Core Processor, memory 16 GB) in a serial setting.

Distortions primarily occurred in three scenarios: (1) **Variation in Surface Curvature:** Distortions occur where the surface curvature changes significantly, challenging the uniform distribution of higher-order nodes, as shown in Fig. 9(a). (2) **Convex Boundary:** These features cause the boundary nodes of the higher-order mesh to stretch compared to interior nodes, leading to distortions illustrated in Fig. 9(b). (3) **Concave Boundary:** Concave features at the boundaries compress boundary nodes during projection, leading to distortions, as shown in Fig. 9(c). This effect is pronounced where geodesics closely approach the CAD boundaries, necessitating element fusion operations to alleviate the distortions.

In this study, we demonstrate the optimization of three types of distortions by our algorithm through four case studies. Our approach comprises the following steps to validate the effectiveness and reliability of the algorithm: Initially, we utilize the Freeform Surface model to validate the experimental outcomes at various phases of our algorithm, assess the convergence of the iterative process, and evaluate its adaptability across diverse element orders. Furthermore, we extend our validation to three complex industrial models to further demonstrate the reliability of our algorithm in practical scenarios.

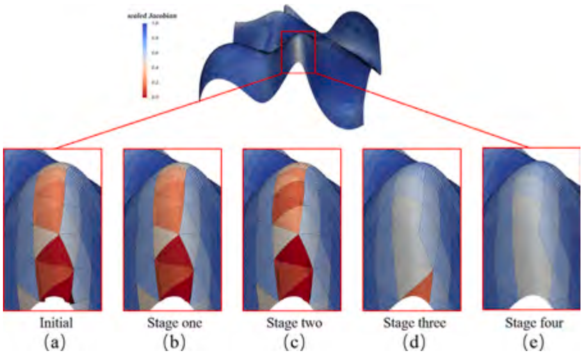
### 5.1. *Test of single-surface example*

We demonstrated the efficacy of our algorithm through a case study by constructing a higher-order mesh with $p = 8$ for the Freeform Surface model.

#### 5.1.1. *Compare with different optimization stages*

Fig. 10 demonstrates the local optimization process of our algorithm on the Freeform Surface model's higher-order mesh, sequentially showcasing the refinement of boundary nodes $\{\boldsymbol{P}_L\}$, edge nodes $\{\boldsymbol{P}_B\}$, interior nodes $\{\boldsymbol{P}_I\}$, and the element fusion step. A color bar represents the scaled Jacobian value for each element, indicating the improvement in mesh quality.

Table 2 details the mesh quality metrics through the optimization stages, illustrating the enhancements from the initial mesh towards an



Initial (a) Stage one (b) Stage two (c) Stage three (d) Stage four (e)

**Fig. 10.** Comparison of the higher-order mesh ($p = 8$) generated by the Freeform Surface Model before and after each optimization step. (a) Initial higher-order mesh. (b) Stage one: boundary nodes $\{\boldsymbol{P}_L\}$ is optimized. (c) Stage two: edge nodes $\{\boldsymbol{P}_B\}$ is optimized. (d) Stage three: interior nodes $\{\boldsymbol{P}_I\}$ is optimized. (e) Stage four: element fusion operation is executed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

optimized result through optimization at different stages. The average, minimum, and maximum $J_S$ values underscore the effectiveness of each optimization phase.

Providing both a global view and detailed local perspectives, Figs. 11(a) and 11(b) compare the initial and optimized higher-order meshes. Our algorithm effectively optimizes the mesh into ideal shapes across areas with convex boundaries, concave boundaries, and significant curvature changes while maintaining sufficient geometric accuracy.

Furthermore, Fig. 12(c) contrasts the mesh quality histograms before and after optimization, with the initial condition shown in red and the optimized state in blue. The two histograms are superimposed, facilitating a direct visual comparison that reveals a marked decrease in low-quality elements and an increase in high-quality elements.

### 5.1.2. Demonstration of algorithm convergence

In our iterative algorithm, we aim to reduce two energies: one denoted as $E_\Sigma$ and the other as $E_{\text{pro}}$. As illustrated in Algorithm 1, as the final step of the algorithm consistently sets the weight $\beta^{(k)}$ of $E_{\text{pro}}$ to 1 to minimize its energy, our primary focus lies on the convergence of the former throughout the iterations. To assess the reduction of $E_\Sigma$, we introduce a normalized energy term:

$$\overline{E}_\Sigma = \frac{E_\Sigma - \min(E_\Sigma)}{\max(E_\Sigma) - \min(E_\Sigma)}.$$

The reduction process of this energy and its impact on optimizing higher-order meshes correspond to steps (c) to (d) in Fig. 10 of the optimization process.

In Fig. 12(a), we demonstrate the process of minimizing the normalized energy while optimizing internal points using the iterative algorithm on the complete higher-order mesh. At first, due to the lower importance of the projection energy, the focus is on reducing the weighted area of the entire higher-order mesh surface, leading to a quick decrease in the normalized energy. At this stage, the primary objective of optimization is to adjust the higher-order mesh to approximate a minimal surface.

After approximately 20 to 30 steps, with the increase in the weight of the projection energy, there is a slight rebound in the normalized energy. This phenomenon mainly arises from the necessity for the higher-order mesh to closely match the CAD surface, which deviates from the optimization goal of minimal surface. Nevertheless, the advantageous effects of parameterizing the higher-order mesh persist throughout the optimization process, as depicted in Fig. 12(b).

In Fig. 12(b), we observe the six elements with the poorest quality in the original higher-order mesh and track their enhancement as the optimization progresses. It is clear that as the higher-order mesh better conforms to the CAD surface, the mesh quality remains almost consistent during the optimization procedure. Following an initial enhancement phase, the mesh quality gradually stabilizes. This illustrates that our method can effectively maintain both geometric precision and higher-order mesh quality simultaneously.

**Table 3**
Mesh quality of Freeform Surface with different orders.

| $p$/Nodes | Stage | $J_S$ | | | $N_T$ | t (s) |
|---|---|---|---|---|---|---|
| | | avg. | min. | max. | | |
| 2<br>5k | Init.<br>Opt. | 0.893<br>0.894 | 0.071<br>0.387 | 0.997<br>0.997 | 0<br>0 | 62.3 |
| 4<br>20k | Init.<br>Opt. | 0.865<br>0.875 | −0.568<br>0.467 | 0.996<br>0.996 | 6<br>0 | 157.9 |
| 6<br>46k | Init.<br>Opt. | 0.857<br>0.871 | −0.383<br>0.464 | 0.996<br>0.996 | 7<br>0 | 259.5 |
| 8<br>82k | Init.<br>Opt. | 0.849<br>0.868 | −1.798<br>0.498 | 0.996<br>0.996 | 9<br>0 | 286.8 |

In our mesh optimization approach, a crucial local technique known as element fusion is utilized to improve the quality of poor elements. This technique, as shown in Fig. 10(d) and (e), significantly enhances the quality of the deficient element. Fig. 12(c) illustrates the convergence of the combined energy of the poor element and its help element during the element fusion process, along with the curve depicting the gradual influence on the quality of both elements. Interestingly, as the optimization energy decreases gradually, the quality of the poor element consistently improves to a notable extent.

### 5.1.3. Compare with different element order

Meshes of different orders were created for the Freeform Surface to assess the capability of our method in generating higher-order meshes of arbitrary degree. The outcomes of the meshing process are illustrated in Fig. 13, while detailed quality metrics and the time required for mesh optimization can be found in Table 3. The number of tangled elements are denoted as $N_T$. Additionally, histograms displaying the $J_S$ values for higher-order meshes of varying degrees were generated both pre and post optimization, offering a visual comparison of the optimization impact, as depicted in Fig. 15.

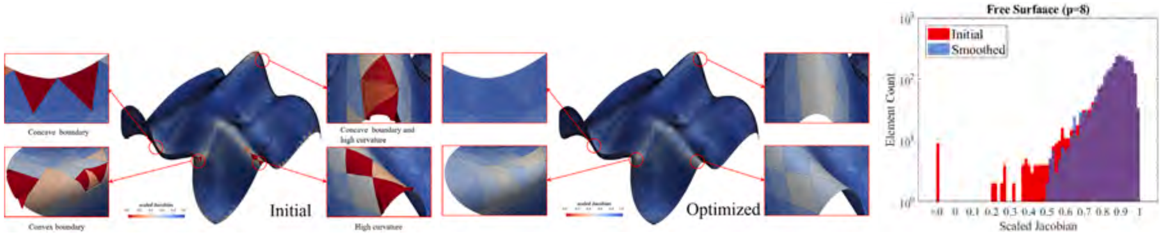
### 5.1.4. Compare with different linear mesh inputs

To verify the robustness of our method, we generated linear meshes with varying densities as input to test whether effective higher-order meshes could be produced. As shown in Fig. 14, we varied the density of the initial linear mesh by 20 times, reducing it from the order of 10,000 to the order of 500, yet we still produced effective higher-order meshes (with the worst $J_S = 0.250$). This demonstrates the robustness of our algorithm to different inputs.

## 5.2. Test of multiple-surface examples

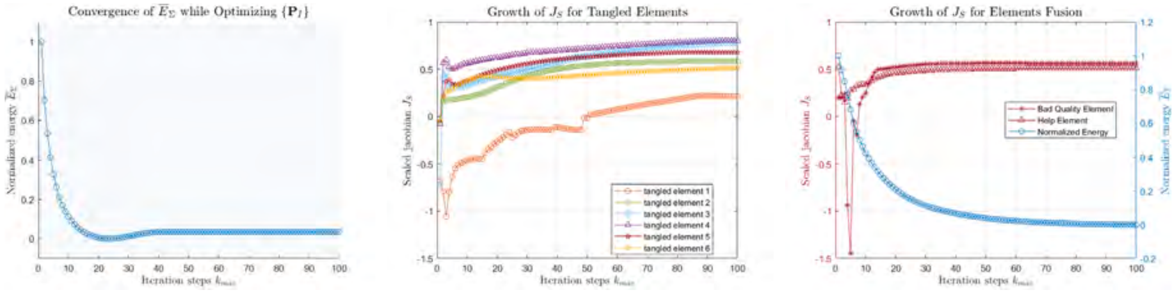
We assess the performance of our algorithm using three multiple-surface test cases. We present the initial and optimized higher-order meshes, offering global and local perspectives. This comparison illustrates our algorithm's effectiveness in correcting distortions in higher-order meshes. A comparison of the mesh quality before and after optimization for these three models is presented in Table 4. This includes the total number of the higher-order nodes, the average, minimum, and maximum $J_S$ values, the change in tangled elements $N_T$ and the time for optimizing.

*Propeller.* For the propeller model, a higher-order mesh with order $p = 6$ was developed, showcasing its initial, optimized states and histogram of quality in Fig. 16(a). The optimization targeted the propeller blades' leading edges, where high curvature was a challenge, and the shaft-blade junctions, where concave boundaries complicated the mesh



(a) (b) (c)

**Fig. 11.** Result of higher-order mesh ($p = 8$) for the Freeform Surface model: Initial vs. optimized, with magnified regions showing high curvature, concave boundaries, and convex boundaries. (a) Initial meshes. (b) Optimized meshes. (c) Histogram of mesh quality.



(a) (b) (c)

**Fig. 12.** The energy convergence curve and quality improvement curve of the Freeform Surface model when optimizing higher-order mesh ($p = 8$) using the iterative algorithm. (a) The convergence curve of the normalized energy during iteration. (b) The quality improvement curve of the $J_S$ for the worst six elements during iteration. (c) Demonstration of the gradual convergence of energy and the improvement in the quality of higher-order elements in the element fusion process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

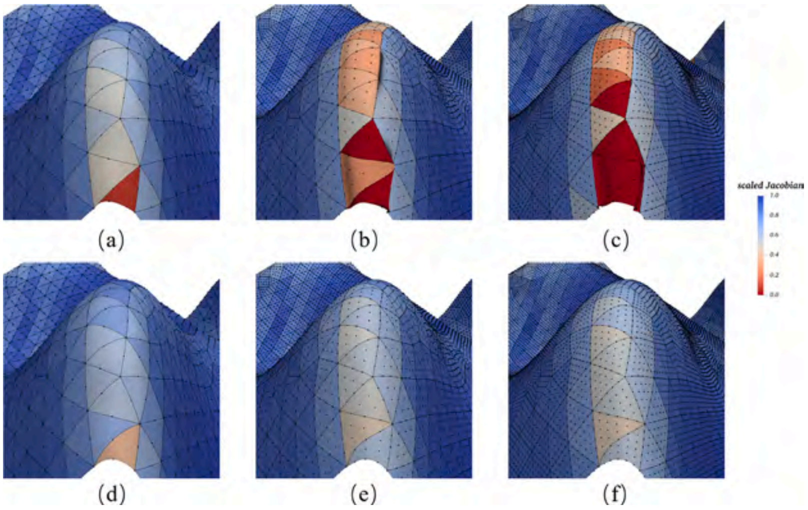


**Fig. 13.** Varying results for higher-order meshes with different orders generated for the Freeform Surface model. Initial elements: (a) $p = 2$. (b) $p = 4$. (c) $p = 6$. Optimized elements: (d) $p = 2$. (e) $p = 4$. (f) $p = 6$.

**Table 4**
Mesh quality comparison before and after optimization for three complex models.

| Model | p/Nodes | Stage | $J_S$ | | | $N_T$ | t (s) |
|---|---|---|---|---|---|---|---|
| | | | avg. | min. | max. | | |
| Propeller | 6<br>117k | Init.<br>Opt. | 0.868<br>0.954 | −3.161<br>0.235 | 1.000<br>1.000 | 124<br>0 | 603.3 |
| DLR-f6 | 4<br>122k | Init.<br>Opt. | 0.961<br>0.969 | −1.372<br>0.328 | 0.998<br>1.000 | 14<br>0 | 193.2 |
| Horten | 6<br>204k | Init.<br>Opt. | 0.931<br>0.939 | −2.811<br>0.189 | 1.000<br>0.999 | 24<br>0 | 217.2 |

quality. Initially, 124 out of 6602 elements were found to be invalid. After applying our algorithm, these elements were corrected, underscoring the algorithm's ability to rectify both boundary and internal distortions effectively. The element fusion operator proved invaluable in areas where the geodesic approach was insufficient.

*Horten.* The Horten model, characterized by its long wingspan and curved wingtips, posed unique challenges for higher-order mesh generation. A mesh of order $p = 6$ was produced, initially depicted in Fig. 16(b) left. The sparse linear mesh used in the initial stages resulted in low-quality mesh areas, particularly noticeable in regions requiring precise curvature representation. Our algorithm's optimization led to a substantial increase in mesh quality, as depicted in Fig. 16(b) center. The distribution of higher-order nodes was optimized, enhancing the

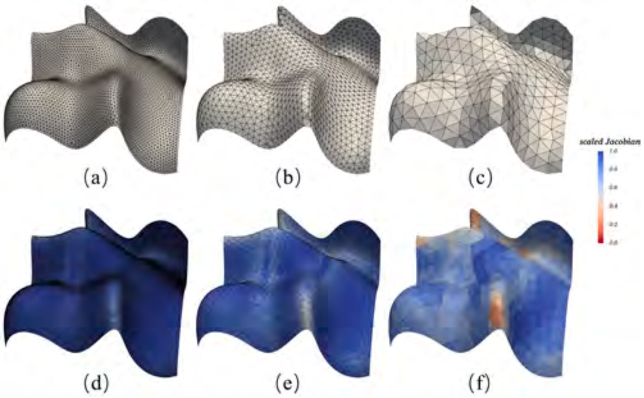


**Fig. 14.** Comparison of higher-order meshes generated from linear mesh inputs with varying densities for the Freeform Surface model (p = 6). Input linear meshes: (a) $N_E$ = 10,498. (b) $N_E$ = 2556. (c) $N_E$ = 516. Output higher-order meshes: (d) $N_E$ = 10,498, minimum $J_S$ = 0.606. (e) $N_E$ = 2556, minimum $J_S$ = 0.464. (f) $N_E$ = 516, minimum $J_S$ = 0.250.

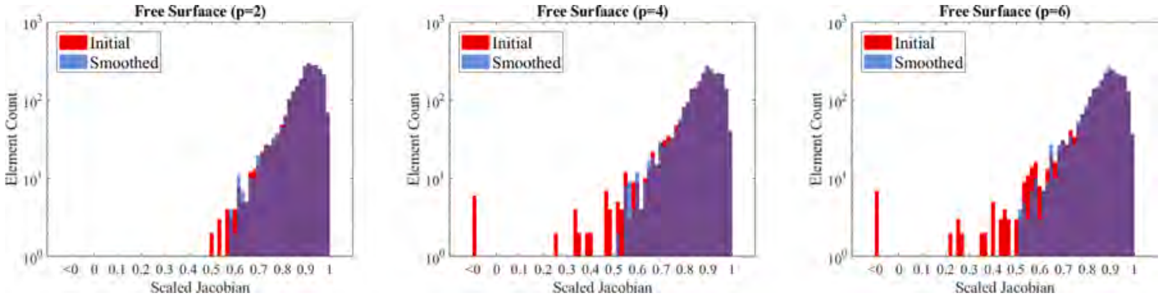


**Fig. 15.** Histogram comparison demonstrating the mesh quality for higher-order meshes with different orders generated for the Freeform Surface model.

mesh's geometric accuracy, which is evident in the quality histogram provided in Fig. 16(b) right.

*DLR-f6 aircraft.* The DLR-f6 aircraft model was meshed with a higher-order mesh of $p = 4$, with the initial, optimized meshes and the histogram displayed in Fig. 16(c). The front of the aircraft and the leading edge of the wing, areas with intricate shapes and sharp curvature changes, initially led to an uneven nodes distribution of higher-order elements. The concave boundary between the fuselage and the wing further exacerbated the element distortion. Our algorithm effectively reduces higher-order mesh distortion in areas of high curvature and eliminates interlocking between the higher-order mesh boundaries and CAD boundary curves at concave boundaries through the element fusion operation. Especially in the wing areas with high curvature, the optimization was notably effective.

*Time performance.* As shown in Tables 3 and 4, there is room for efficiency improvement in our iterative algorithm. This algorithm generates a minimal surface for each higher-order surface mesh and extensively uses projection to gradually fit the target CAD surface. For the complex models we tested, generating higher-order meshes with a significant number of tangled elements typically takes optimization times on the order of minutes.

*Shape discrepancies.* In addition to mesh quality, we also need to consider the conformity of the higher-order mesh to the geometry. To characterize the shape discrepancies between the final higher-order mesh and the initial CAD model, we use the normal deviation metric $\Delta n = \|n_H - n^*\|$, defined as the norm of the deviation between the unit normal $n_H$ at a point on the higher-order mesh and the unit normal $n^*$ at the corresponding point. We use the Horten model to demonstrate the impact of different linear mesh subdivision numbers $\alpha$ on the final results. As shown in Fig. 17, more subdivisions result in a finer approximation of the continuous energy function, leading to a better result with smaller $\Delta n$.

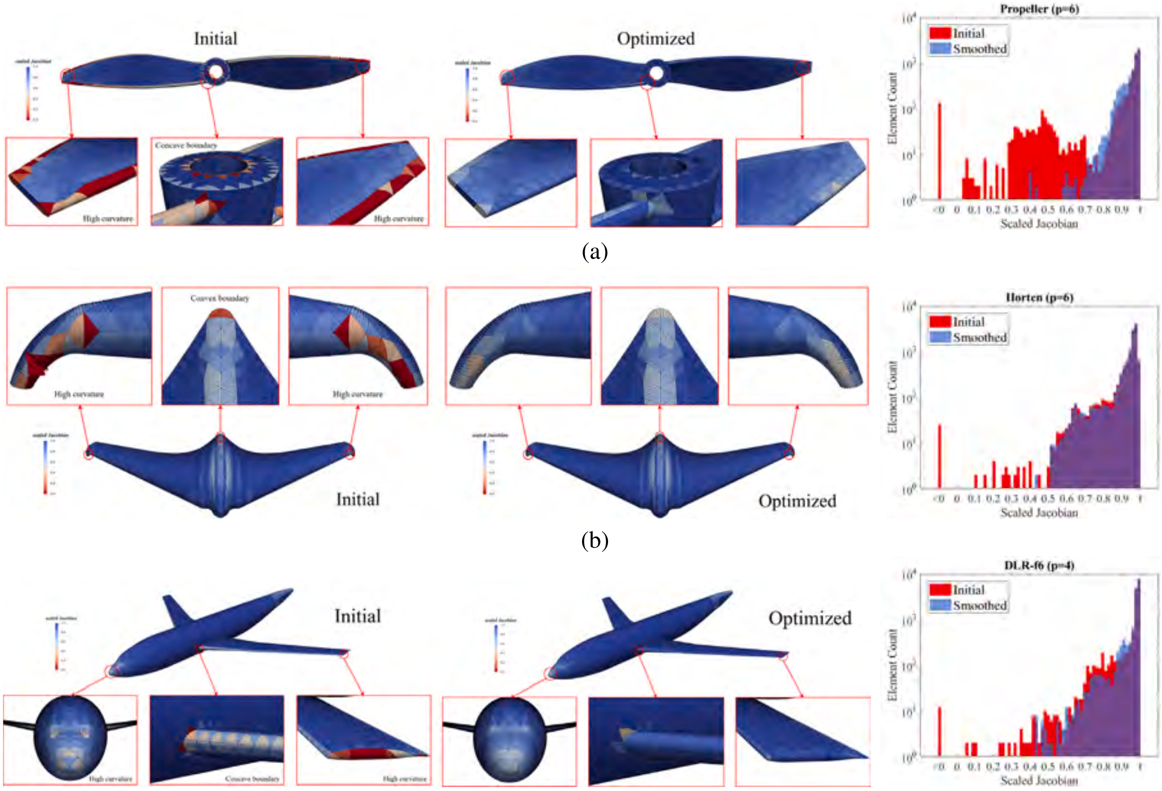
In Fig. 18, we present some other local views to demonstrate the changes of $\Delta n$ for three complex models before and after optimization. The initially generated higher-order meshes exhibit significant normal deviations, indicating a poor fit with the original CAD model. Our algorithm effectively repairs these distorted regions, resulting in a higher degree of alignment between the unit normals of the final higher-order meshes and the original CAD models. This means that the higher-order meshes generated by our method can impose more accurate geometric boundary conditions, providing significant advantages for specific numerical methods such as the boundary element method.

*The effects of local adaptive refinement.* After optimization, if the local quality of the higher-order mesh does not meet the requirements, we can apply adaptive refinement to optimize the poor elements and their neighboring areas. In our implementation, if the $J_S$ of the higher-order mesh is less than 0.8 after node optimization, we employ the methods described in Section 4.5 to locally optimize the quality of the mesh in the affected elements and their neighboring regions. As shown in Fig. 19, for regions with high curvature, applying such operations can increase the density and quality of the local linear mesh, thereby improving the quality of the corresponding higher-order mesh. This approach helps to avoid issues where the initial linear mesh has insufficient density or quality in certain areas, thereby significantly improving the overall quality of the higher-order mesh.

## 6. Conclusion and future work

This paper presents a new method for creating high-quality, higher-order CAD surface meshes, linking minimal surfaces with higher-order meshing and introducing an iterative algorithm based on minimal surface theory for mesh enhancement. We also propose element fusion to potentially enhance boundary mesh quality. Initial experiments indicate the method's effectiveness in generating accurate higher-order meshes for intricate CAD models.

While the method offers certain advantages, it also highlights potential for technical refinement. Using subdivisions in higher-order meshes may not fully capture the continuous nature of minimal surfaces



(a)

(b)

(c)

**Fig. 16.** The results of higher-order meshes for three complex models: initial meshes, optimized meshes, and mesh quality histogram. (a) **Propeller** ($p = 6$): The high curvature feature at the leading edge of the propeller blade leads to significant distortion when using projection-based methods. The concave boundary at the junction of the shaft and the blade also makes it highly susceptible for higher-order mesh boundaries to intersect with CAD surface boundaries. (b) **Horten** ($p = 6$): The significant curvature variation at the wing's trailing edge results in substantial distortion when using projection-based methods. The convex boundary at the aircraft's tail end further causes considerable distortion in the elements. (c) **DLR-f6 aircraft** ($p = 4$): The high-curvature features located at the nose of the aircraft, the leading edge of the wings, and the sides of the wings lead to significant distortion in the initial higher-order mesh. The concave boundary where the wing meets the fuselage further causes self-intersection of the higher-order mesh boundaries.

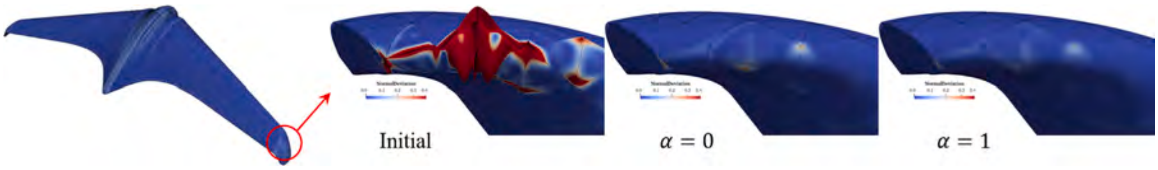


**Fig. 17.** Comparison of the normal deviation of the higher-order mesh after optimization under different subdivision number $\alpha$.

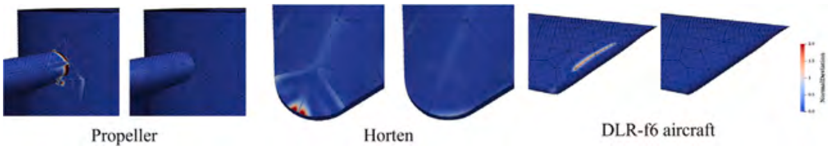


**Fig. 18.** We present the changes in normal deviation for each model before and after applying the mesh optimization algorithm. After optimization, the normal deviation between the higher-order mesh and the original CAD model is significantly reduced.

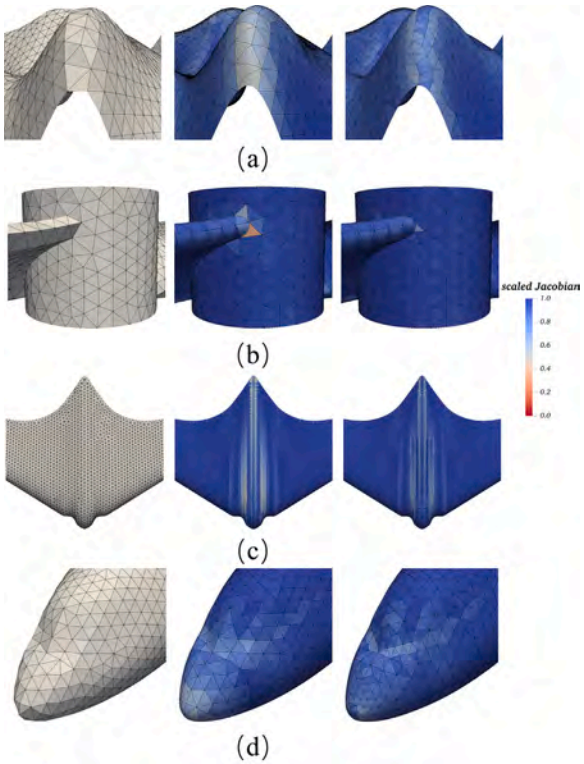


**Fig. 19.** For each model, we present a localized view of the initial linear mesh, the smoothed higher-order mesh, and the higher-order mesh after an additional step of adaptive refinement. This sequence illustrates the improvement in the quality of the worst elements (minimum $J_S$) achieved through adaptive refinement techniques. (a) Minimum $J_S$ of Freeform Surface ($p = 8$) is improved from 0.498 to 0.661. (b) Minimum $J_S$ of Propeller ($p = 6$) is improved from 0.235 to 0.507. (c) Minimum $J_S$ of Horten ($p = 6$) is improved from 0.189 to 0.478. (d) Minimum $J_S$ of DLR-f6 ($p = 4$) is improved from 0.328 to 0.414.

because of the nonlinearity of the equations, possibly limiting its untangling capabilities under specific conditions. Future efforts will aim to replace subdivisions with numerical integration and focus on boosting the efficiency and applicability of this technique to higher-order volume mesh generation.

## CRediT authorship contribution statement

**Kaixin Yu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Bohan Wang:** Writing – review & editing, Visualization. **Xuejuan Chen:** Writing – review & editing. **Ying He:** Writing – review & editing, Supervision. **Jianjun Chen:** Supervision, Resources.

## Declaration of competing interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

## Data availability

Data will be made available on request.

## Acknowledgments

This research was supported by the Specialized Research Projects of Huanjiang Laboratory (Grant No. KM20220616) and the Science and Technology on Scramjet Laboratory Fund (Grant No. 2022-JCJQ-LB020-05).

The authors also gratefully acknowledge the support from the China Scholarship Council (CSC) program (Project ID: 202306320385).

## References

[1] Wang ZJ, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, Deconinck H, Hartmann R, Hillewaert K, Huynh HT, et al. High-order CFD methods: current status and perspective. Internat J Numer Methods Fluids 2013;72(8):811–45.
[2] Poya R, Sevilla R, Gil AJ. A unified approach for a posteriori high-order curved mesh generation using solid mechanics. Comput Mech 2016;58(3):457–90.
[3] Chevaugeon N, Remacle J-F, Gallez X, Ploumans P, Caro S. Efficient discontinuous Galerkin methods for solving acoustic problems. In: 11th AIAA/CEAS aeroacoustics conference. 2005, p. 2823.
[4] Wilcox LC, Stadler G, Burstedde C, Ghattas O. A high-order discontinuous Galerkin method for wave propagation through coupled elastic–acoustic media. J Comput Phys 2010;229(24):9373–96.
[5] Stanglmeier M, Nguyen N, Peraire J, Cockburn B. An explicit hybridizable discontinuous Galerkin method for the acoustic wave equation. Comput Methods Appl Mech Engrg 2016;300:748–69.
[6] Sevilla R, Hassan O, Morgan K. The use of hybrid meshes to improve the efficiency of a discontinuous Galerkin method for the solution of Maxwell's equations. Comput Struct 2014;137:2–13.
[7] Hesthaven JS, Warburton T. Nodal high-order methods on unstructured grids: I. Time-domain solution of Maxwell's equations. J Comput Phys 2002;181(1):186–221.
[8] Liu R, Wheeler MF, Yotov I. On the spatial formulation of discontinuous Galerkin methods for finite elastoplasticity. Comput Methods Appl Mech Engrg 2013;253:219–36.
[9] Soon S. Hybridizable discontinuous Galerkin method for solid mechanics. University of Minnesota; 2008.
[10] Cockburn B. Discontinuous Galerkin methods for computational fluid dynamics. In: Encyclopedia of computational mechanics second edition. Wiley Online Library; 2018, p. 1–63.
[11] Nguyen NC, Peraire J, Cockburn B. A class of embedded discontinuous Galerkin methods for computational fluid dynamics. J Comput Phys 2015;302:674–92.
[12] Atak M, Beck A, Bolemann T, Flad D, Frank H, Hindenlang F, Munz C-D. Discontinuous Galerkin for high performance computational fluid dynamics. In: High performance computing in science and engineering '14. Springer; 2015, p. 499–518.
[13] Dey S, O'Bara RM, Shephard MS. Towards curvilinear meshing in 3D: the case of quadratic simplices. Comput Aided Des 2001;33(3):199–209.
[14] Mohammadi F, Dangi S, Shontz SM, Linte CA. A direct high-order curvilinear triangular mesh generation method using an advancing front technique. In: International conference on computational science. Springer; 2020, p. 72–85.
[15] Mohammadi F, Shontz SM. A direct method for generating quadratic curvilinear tetrahedral meshes using an advancing front approach. In: Proc. of the 29th International Meshing Roundtable. http://dx.doi.org/10.5281/zenodo.5559211, https://par.nsf.gov/biblio/10357889.
[16] Mandad M, Campen M. Bézier guarding: precise higher-order meshing of curved 2D domains. 2020;39(4). http://dx.doi.org/10.1145/3386569.3392372, https://doi.org/10.1145/3386569.3392372.
[17] Mandad M, Campen M. Guaranteed-quality higher-order triangular meshing of 2D domains. ACM Trans Graph 2021;40(4):1–14.
[18] Khanteimouri P, Mandad M, Campen M. Rational Bézier guarding. In: Computer graphics forum. Vol. 41, Wiley Online Library; 2022, p. 89–99.
[19] Khanteimouri P, Campen M. 3D Bézier guarding: Boundary-conforming curved tetrahedral meshing. ACM Trans Graph 2023;42(6):1–19.
[20] Ruiz-Gironés E, Sarrate J, Roca X. Defining an 2-disparity measure to check and improve the geometric accuracy of non-interpolating curved high-order meshes. Procedia Eng 2015;124:122–34.
[21] Gargallo-Peiró A, Roca X, Peraire J, Sarrate J. A distortion measure to validate and generate curved high-order meshes on CAD surfaces with independence of parameterization. Internat J Numer Methods Engrg 2016;106(13):1100–30.
[22] Xie ZQ, Sevilla R, Hassan O, Morgan K. The generation of arbitrary order curved meshes for 3D finite element analysis. Comput Mech 2013;51(3):361–74.
[23] Persson P-O, Peraire J. Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In: 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. 2009, p. 949.
[24] Toulorge T, Geuzaine C, Remacle J-F, Lambrechts J. Robust untangling of curvilinear meshes. J Comput Phys 2013;254:8–26.
[25] Turner M, Peiró J, Moxey D. Curvilinear mesh generation using a variational framework. Comput Aided Des 2018;103:73–91.

[26] Zhao Z, Li M, He L, Shao S, Zhang L. High-order curvilinear mesh generation technique based on an improved radius basic function approach. Internat J Numer Methods Fluids 2019;91(3):97–111.

[27] Fortunato M, Persson P-O. High-order unstructured curved mesh generation using the Winslow equations. J Comput Phys 2016;307:1–14.

[28] Sherwin S, Peiró J. Mesh generation in curvilinear domains using high-order elements. Internat J Numer Methods Engrg 2002;53(1):207–23.

[29] Moxey D, Ekelschot D, Keskin Ü, Sherwin S, Peiró J. A thermo-elastic analogy for high-order curvilinear meshing with control of mesh validity and quality. Procedia Eng 2014;82:127–35.

[30] Feuillet R, Loseille A, Alauzet F. Optimization of P2 meshes and applications. Comput Aided Des 2020;124:102846.

[31] Shi A, Persson P-O. Local element operations for curved simplex meshes. Internat J Numer Methods Engrg 2024;125(2):e7379.

[32] Gargallo-Peiró A, Roca X, Peraire J, Sarrate J. Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. Internat J Numer Methods Engrg 2015;103(5):342–63.

[33] Ruiz-Gironés E, Roca X, Sarrate J. High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation. Comput Aided Des 2016;72:52–64.

[34] Turner M, Moxey D, Sherwin SJ, Peiró J. Automatic generation of 3D unstructured high-order curvilinear meshes. In: ECCOMAS proceedings. 2016.

[35] Green MD, Kirilov KS, Turner M, Marcon J, Eichstädt J, Laughton E, Cantwell CD, Sherwin SJ, Peiró J, Moxey D. NekMesh: An open-source high-order mesh generation framework. Comput Phys Comm 2024;298:109089.

[36] Bock K, Stiller J. Optimizing triangular high-order surface meshes by energy-minimization. Eng Comput 2018;34(4):659–70.

[37] Renka RJ, Neuberger J. Minimal surfaces and Sobolev gradients. SIAM J Sci Comput 1995;16(6):1412–27.

[38] Renka RJ. A trust region method for constructing triangle-mesh approximations of parametric minimal surfaces. Appl Numer Math 2014;76:93–100.

[39] Renka RJ. Two simple methods for improving a triangle mesh surface. In: Computer graphics forum. Vol. 35, Wiley Online Library; 2016, p. 46–58.

[40] Yu K, Chen J, Fu K, He J, Zheng J, Zheng Y. On the efficiency of the advancing-front surface mesh generation algorithm. Comput Aided Des 2022;153:103403.

[41] Liu Z, Chen J, Xia Y, Zheng Y. Automatic sizing functions for unstructured mesh generation revisited. Eng Comput 2021;38(10):3995–4023.

[42] Geuzaine C, Johnen A, Lambrechts J, Remacle JF, Toulorge T. The generation of valid curvilinear meshes. In: IDIHOM: Industrialization of high-order methods-a top-down approach: results of a collaborative research project funded by the European union, 2010-2014. Springer; 2015, p. 15–39.

[43] Bock K, Stiller J, et al. Energy-minimizing curve fitting for high-order surface mesh generation. Appl Math 2014;5(21):3318.