



# The generation of tetrahedral meshes for NURBS-enhanced FEM

Xi Zou<sup>1</sup> · Sui Bun Lo<sup>1</sup> · Ruben Sevilla<sup>1</sup> · Oubay Hassan<sup>1</sup> · Kenneth Morgan<sup>1</sup>

Received: 28 March 2024 / Accepted: 10 May 2024  
© The Author(s) 2024

## Abstract

This work presents the first method for generating tetrahedral-based volume meshes dedicated to the NURBS-enhanced finite element method (NEFEM). Built upon the developed method of generating feature-independent surface meshes tailored for NEFEM, the proposed mesh generation scheme is able to grow volume elements that inherit the feature-independence by using the surface mesh as the initial boundary discretisation. Therefore, the generated tetrahedral elements may contain triangular faces that span across multiple NURBS surfaces whilst maintaining the exact boundary description. The proposed strategy completely eliminates the need for de-featuring complex watertight CAD models. At the same time, it eliminates the uncertainty originated from the simplification of CAD models adopted in industrial practice and the error introduced by traditional isoparametric mesh generators that produce polynomial approximations of the true boundary representation. Thanks to the capability of having element faces traversing multiple geometric surfaces, small geometric features in the CAD model no longer restrict the minimum element size, and the user-required mesh spacing in the generated mesh is better satisfied than in traditional meshes that require local refinement. To demonstrate the ability of the proposed approach, a variety of CAD geometries are meshed with the proposed strategy, including examples relevant to the fluid dynamics, wave propagation and solid mechanics communities.

**Keywords** mesh generation · NURBS-enhanced finite element method (NEFEM) · De-featuring · Persistent geometry · High-order approximation

## 1 Introduction

Computer simulation often requires generating a mesh from a computer-aided design (CAD) model which is mathematically described by non-uniform rational B-splines (NURBS) [1]. This process typically involves two steps: the preparation of the geometry and the discretisation. The preparation of the geometry requires substantial human intervention,

and cannot be performed automatically with computer software. This is because (1) CAD models throughout the digitised industry pipeline are prone to contain geometric issues [2, 3] due to the tolerance of floating point arithmetic; (2) realistic CAD models normally contain multi-scale geometric features [4, 5] that may or may not have an impact on a particular simulation. The reliance on human experience and expertise to make decisions to repair and/or de-feature a CAD model seriously hampers the current ability to perform simulations with high fidelity geometric models. Standard mesh generators normally create excessive refined elements near regions with small geometric features, ignoring the user-defined mesh size or spacing. The unnecessary local refinement, often results in highly distorted or badly shaped elements, introducing a negative impact to the efficiency of simulations and the quality of results. For instance, unnecessary small elements can pose a severe restriction in an explicit time stepping scheme for the simulation of transient problems [6]. This is particularly important for exploiting the high-order approximation [7–10], where coarse meshes are preferred. On the other hand, the existence of small

---

✉ Xi Zou  
xi.zou@swansea.ac.uk

Sui Bun Lo  
s.lo@swansea.ac.uk

Ruben Sevilla  
r.sevilla@swansea.ac.uk

Oubay Hassan  
o.hassan@swansea.ac.uk

Kenneth Morgan  
k.morgan@swansea.ac.uk

<sup>1</sup> Faculty of Science and Engineering, Swansea University, Swansea SA1 8EN, Wales, UK

elements, even only one, can drastically drive the simulation away from an affordable cost, and, in a sense, hurdles a widespread adoption of high-order methods in industry.

Despite of the existence of fully automatic standard mesh generators, including academic packages such as [11–13] and numerous commercial packages, the software dedicated to geometry preparation, such as [14, 15], is only semi-automatic. Large resources have been invested to the research and development of automatic de-featuring complex CAD models [16–18]. However, fully automatised de-featuring has not yet been achieved. Machine learning techniques have been used to classify some common geometric features [19] in a large assembly, though it requires a significant amount of pre-labelled data. In addition, it is difficult to predict how the feature will affect the simulation, especially when it involves multiphysics or depends on geometry-related parameters. A naive de-featuring could unintentionally reduce the fidelity of the physical problem, and introduce unintended uncertainty. For instance, the level of de-featuring required in problems on heat transfer, stress analysis, aerodynamics or wave propagation is completely different. When different levels of de-featuring are performed on the same geometry, it is not always possible to cater for all necessary simulations. This means either the same geometric model is considered for all the physics with a non-optimal de-featuring, or different geometries are produced to simulate different physics, resulting in misaligned and inconsistent simulation results and preventing multi-objective optimisation processes.

In addition to the difficulties in de-featuring CAD models, the standard mesh generation also introduces a geometric uncertainty that is well-known but often ignored. This is because the *isoparametric* element [20], which is the most common approach used in existing solvers, uses the same function space, spanned by polynomials, to approximate both the geometry and the unknown fields. In particular, when isoparametric elements are used to discretise the domain a piecewise polynomial approximation of the true boundary representation is introduced, and the discrepancy between the true and the approximation geometry increases when coarse meshes are considered, which is often the case in high-order solvers. The accuracy of the geometric approximation can be improved by refining the mesh based on the boundary curvature, known as the *h*-refinement, or by increasing the number of interpolating nodes and adapting them to the curved boundary, regarded as the *p*-refinement. However, insufficient attention is often paid to the fact that the resulting mesh provides only a piecewise polynomial approximation of the original CAD geometry, and the geometric continuity between elements is only  $C^0$ . This implies that the result could possibly converge to a

solution defined on a different geometric domain with its boundary described by the polynomial functions rather than the desired NURBS B-rep of the CAD geometry [21]. When high-order isoparametric elements are employed on coarse meshes, the piecewise polynomial approximation of the geometry with  $C^0$  continuity between elements can induce nonphysical effects, such as stress singularity in solid mechanics, diffraction in wave propagation, as well as entropy production and flow separation in fluid dynamics. Such effects, as thoroughly discussed in [22–24], can lead the *p*-adaptive analysis to an incorrect solution.

Originally proposed in [25] in a two-dimensional framework, the NURBS-enhanced finite element method (NEFEM) provides a simple and efficient approach to guarantee the true geometry from a CAD model is persistent throughout the whole simulation pipeline. This method was later extended to three dimensions in [26] and has been tested in various problems including fluid dynamics, electromagnetic wave propagation, heat transfer and stress analysis [27]. In the treatment of NURBS geometry, NEFEM is similar to the isogeometric analysis approach [28]. Both methods uses the NURBS parametrisation directly from the CAD data to describe the shape of discretised elements at the boundary. Therefore, the de-featuring of the geometry is avoided, and the geometric approximation error is completely eliminated. However, the isogeometric approach also requires the use of NURBS functions as the basis to approximate the unknown solution, which demands a fundamental modification on the geometry kernel of CAD software, as the three-dimensional NURBS parametrisation is not available in industrial CAD platforms [29] where a CAD model is constructed following the B-rep philosophy. Moreover, isogeometric methods do not prevent the existence of small elements caused by small geometric features. Unlike the isogeometric approach, NEFEM uses standard polynomial functions for the approximation of solution, addressing the problem by a complete separation of the geometry description and solution approximation. This means an enriched element, with NURBS functions describing its geometric shape, and polynomial functions approximating the solutions within the domain, is no longer isoparametric. In this way, the NEFEM element sizes are not restricted by the geometric features, and are controlled by user-specified spacing. Similar to the virtual topology techniques [30] that are able to modify the topological structure of the CAD model without changing the geometry, a NURBS-enhanced element is allowed to *traverse* different surfaces to achieve the desired element size. In addition, the traversing NEFEM elements can span across surfaces connected with only  $C^0$  continuity.

Apart from the benefits introduced by NEFEM, its extensive application to industrial problems involving complex geometries has been impeded by the lack of a capable mesh generator. This has also led to development of strategies using the NEFEM rationale but without the need of a mesh, namely unfitted [31–33] or meshless methods [34, 35]. A two-dimensional NEFEM mesh generator was developed in [36], where the use of NEFEM showed to accelerate the simulation of an electromagnetic wave scattering problem by a factor of 140. To extend this success to three-dimensional problems, a surface mesh generator tailored for NEFEM was developed in [37].

This paper presents, for the first time, the 3D volume mesh generation framework tailored for NEFEM. The proposed method is based on the NEFEM surface mesh generator [37], with extra modifications to allow further validity checks during the volume element creation. The generated surface mesh, which already uses the NURBS B-rep and enables surface elements to traverse multiple curves or surfaces, is used as the discretised boundary for volume meshing, thus ensuring the preservation of the true geometry from the CAD model. Thus, naturally the first layer of NEFEM volume elements grown upon these surface elements will inherit the exact geometry. To ensure the validity of the NEFEM volume elements, intersection checks are performed during the creation process. The detailed algorithms are listed in pseudo codes, and explained with illustrations. Examples are presented to demonstrate the ability of the developed framework. In addition numerical validity checks are performed on the generated meshes.

The outline of the paper is as follows. The fundamentals of NEFEM is introduced in Sect. 2, where the definition of NEFEM entities is extended to account for the possibility of triangular elements spanning across multiple surfaces. Section 3 briefly recalls the NEFEM surface mesh generation and introduces the necessarily additional checks appended to this process to enable NEFEM volume mesh generation. Section 4 presents the proposed technique to generate NEFEM volume meshes. In Sect. 4.4, the generation of high-order nodal distributions on NEFEM surface elements is detailed. Several mesh examples, of increasing complexity, are presented in Sect. 5, to illustrate the potential of the proposed technique. Finally, Sect. 6 summarises the main conclusions.

## 2 Tetrahedral mesh tailored for NEFEM

This section aims at introducing characteristics of the NURBS-enhanced element and of the mesh tailored for NURBS-enhanced FEM. Formal definitions of the geometric entities, which lay the foundation of the proposed mesh generation strategy, are detailed.

### 2.1 Characteristics of NEFEM elements and mesh

In three dimensions, a standard isoparametric FE mesh is typically generated in a hierarchical manner, discretising the CAD model from points, curves, surfaces and volumes. During this process, the geometric entities are associated with the mesh entities. To be specific, points in the CAD model define nodes in the FEM mesh; curves are discretised into edges; surfaces are approximated with faces or facets such as triangles or quadrilaterals; and volumes are divided into polyhedral elements such as tetrahedra, prisms, pyramids or hexahedra. After the mesh has been created, a set of nodes and element connectivities are stored and the CAD geometric information is usually discarded in the forthcoming simulations.

The foundation of NEFEM [38] is the separation of the geometric approximation and functional approximation, which are tightly coupled in isoparametric FEs and isogeometric methods. To decouple these two concepts, NEFEM generalises the definition of an FE: the geometry is exactly described by means of the NURBS parametrised B-rep that, directly obtained from CAD models, whereas the functional approximation is defined using polynomials, as in standard FEM. As a result, NEFEM elements require new quadrature rules to ensure that the exact B-rep is accounted for by the solver.

When a CAD model contains small geometric features, e.g. short curves or sliver surfaces, standard meshing algorithms disregard the user-specified mesh spacing, generating small elements that are often badly shaped. NEFEM has introduced a new class of FE, which is dedicated to lifting the restriction of small geometric features inducing small elements, so that compliance with the user-defined spacing is significantly improved.

As an example, let us consider a CAD model of a hollow fairing of a turbine engine, represented in Fig. 1a, where all NURBS curves are rendered in blue. A high-order triangular surface mesh generated by a standard mesh generator is shown in Fig. 1b. The virtual topology approach has been employed to enable the generation of elements across surfaces with a smooth transition of the outward unit normal. However, it can be clearly observed that small and badly shaped triangles are present in trailing edge regions where the desired element size is much larger than the sliver feature. In contrast, the NEFEM mesh shown in Fig. 1c has employed some NURBS-enhanced elements that traversed the surfaces, even without a smooth transition of the normal, resulting in compliant element sizes.

In two dimensions, a typical NEFEM element is defined as a curved triangle where at least one edge is geometrically defined as a combination of trimmed NURBS curves.

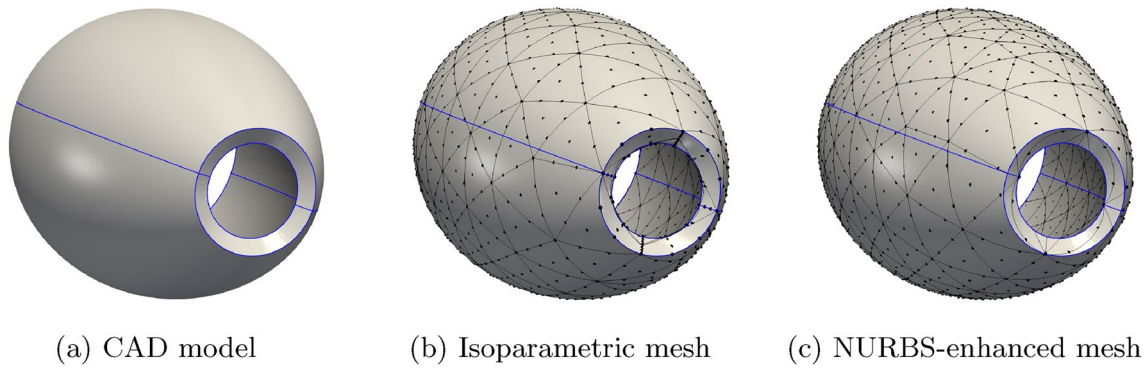


Fig. 1 Meshing a hollow fairing model with isoparametric and NURBS-enhanced surface elements

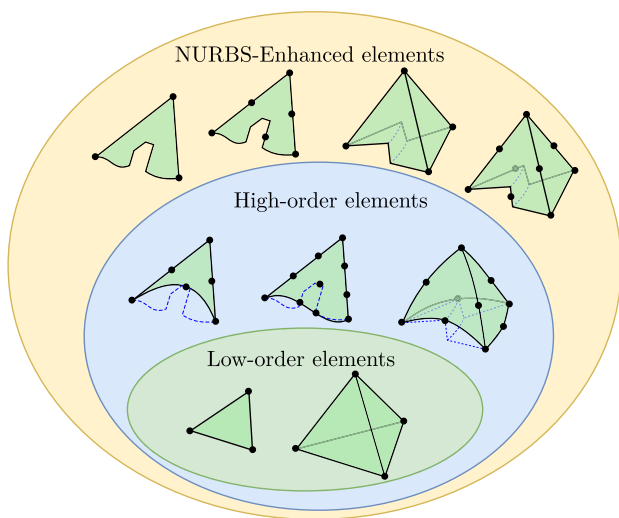


Fig. 2 Illustration of the generalisation introduced by NEFEM

Similarly, in three dimensions, a typical NEFEM element can be defined as a tetrahedron where at least one edge or face is geometrically defined as a collection of trimmed NURBS curves or surfaces, respectively. The new concept of element is illustrated in Fig. 2 and a detailed discussion can be found in [38].

Among the illustrative examples of NEFEM elements in Fig. 2, it can be observed that the exact B-rep is always preserved by the corresponding NEFEM element, regardless of the order of approximation used in the element. In particular, a NEFEM element with low-order interpolation nodes is capable of representing a curved boundary. In addition, Fig. 2 also shows that the face of a NEFEM element can be made of a collection of NURBS

surface patches, even with abrupt changes of the normal within the face.

It is worth noting that NEFEM elements can be restricted to just one layer of elements in contact with the boundary of the domain. The large majority of elements in a NEFEM mesh do not have any edge or face on the boundary, and the standard isoparametric FEM approach is used. This implies that NEFEM elements are only used near the boundary, and a negligible computational overhead will be introduced when compared to the cost of standard FEs.

### 2.2 Definition of NEFEM elements

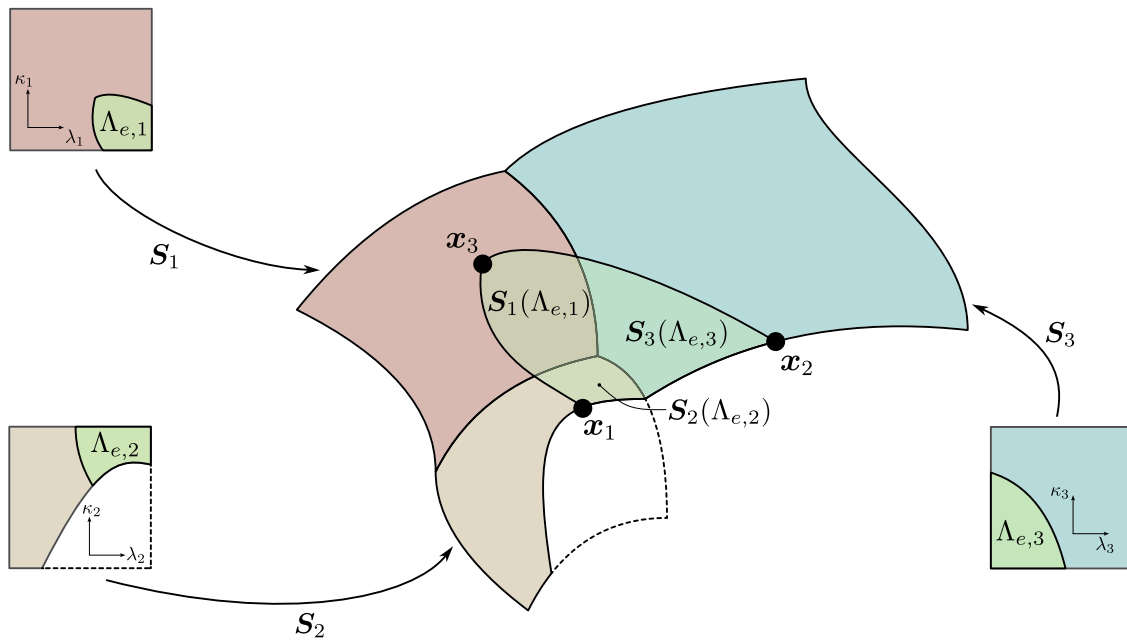
The boundary of an open bounded domain  $\Omega \in \mathbb{R}^3$ , denoted by  $\partial\Omega$ , is assumed to be described by a set of  $n_s$  NURBS surfaces  $\mathcal{S} := \{\mathcal{S}_j\}_{j=1, \dots, n_s}$ . The intersections of the surfaces define a set of  $n_c$  NURBS curves  $\mathcal{C} := \{\mathcal{C}_i\}_{i=1, \dots, n_c}$ . Each curve  $\mathcal{C}_i$  or surface  $\mathcal{S}_j$  is parametrised as

$$\begin{aligned} \mathcal{C}_i(\lambda) : \lambda \in [0, 1] &\quad \mapsto \mathbf{x} \in \partial\Omega \subset \mathbb{R}^3; \\ \mathcal{S}_j(\lambda, \kappa) : (\lambda, \kappa) \in [0, 1]^2 &\quad \mapsto \mathbf{x} \in \partial\Omega \subset \mathbb{R}^3. \end{aligned}$$

For further details on NURBS curves and surfaces, the reader is referred to [1].

In three dimensions, the surface mesh generation strategy proposed in [37] introduced the mapping to define the geometry of surface NEFEM elements using the exact B-rep from CAD data. In particular, a NEFEM surface element, which could traverse several NURBS surfaces  $\{\mathcal{S}_i(\lambda, \kappa)\}_{i=1, \dots, n_{sdo}}$ , is defined as the image of a piecewise mapping from the parameter domains  $\Lambda_{e,i} \subset \mathbb{R}^2$  to the physical domain  $\mathcal{S}_i(\Lambda_{e,i}) \subset \mathbb{R}^3$ , where  $n_{sdo}$  denotes the number of subdomains of the piecewise mapping. An example is presented in Fig. 3 to illustrate the mapping for a NURBS surface element traversing three surfaces rendered in different colours.





**Fig. 3** Illustration of the piecewise NEFEM geometric mappings for a triangular NEFEM surface element traversing three NURBS surfaces ( $S_1, S_2, S_3$ ), where  $S_2$  is a trimmed surface

The piecewise surface mapping is then extended to the volume element. Let us consider a finite element  $\Omega_e \subseteq \Omega \subset \mathbb{R}^3$  with  $n_{\text{dof}}$  degrees of freedom. The parametrised geometry of the element is defined by a mapping from the parameter domain  $\Pi_e \subset \mathbb{R}^3$  to the physical domain  $\Omega_e \subset \mathbb{R}^3$ . The parameter domain of a NEFEM element is defined as  $\Pi_e := \Lambda_e \times [0, 1]$ , where  $\Lambda_e$  is the parameter domain of the surface element in  $S(\lambda, \kappa)$ . The mapping to define the geometry of a volumetric NEFEM element is given by

$$\begin{aligned} \psi : \Pi_e &\longrightarrow \Omega_e; \\ (\lambda, \kappa, \vartheta) &\longmapsto \psi(\lambda, \kappa, \vartheta) := (1 - \vartheta)S(\lambda, \kappa) + \vartheta\mathbf{x}_4, \end{aligned} \tag{1}$$

where  $\mathbf{x}_4$  denotes the node interior to the domain. This mapping can be viewed as a particular case of the blending function method [39], where the internal edges of the tetrahedral element  $\Omega_e$  are assumed to be straight. More complex mappings can be designed to consider tetrahedral elements involving multiple faces or edges on the boundary. Further details have been introduced in [38].

In NEFEM, the shape functions,  $\{N_i(\mathbf{x})\}_{i=1}^{n_{\text{dof}}}$  and the derivatives are defined and evaluated at the integration points of each individual element, directly in the physical domain, and the elemental matrices and vectors are computed in an *ad hoc* manner for each element. Therefore, the incorporation of NEFEM elements into an existing solver can be easily achieved by creating a new element type that

encloses the CAD data and is accompanied by tailored quadrature rules for NEFEM elements.

In the case that the face of  $\Omega_e$  on the B-rep CAD involves multiple NURBS surfaces  $\{S_i(\lambda_i, \kappa_i)\}$ , the geometric mapping  $\psi$  is accordingly defined in a piecewise manner, i.e.  $\{\psi_i\}$ . The parameter domains are also defined in a piecewise manner, and thus the element  $\Omega_e$  consists of three sub-tetrahedra which are mapped from the three prisms in the corresponding parameter domains. This mapping is illustrated in Fig. 4 for a tetrahedral element with a bottom face traversing three trimmed NURBS surfaces that are rendered in distinguished colours. Note that due to the particular design of  $\psi$  in (1), all top faces of the prisms are mapped to the same node  $\mathbf{x}_4$ .

### 2.3 Geometric entities facilitating subdivision of NEFEM elements

Standard FEM mesh generation algorithms are designed to generate surface elements to approximate the B-rep of CAD model, which is then discarded in subsequent volume meshing procedures. The NEFEM surface mesh generation framework proposed in [37] has introduced the necessary geometric entities for constructing NEFEM surface elements, which embedded the NURBS parametrisation into the definition. In this section,

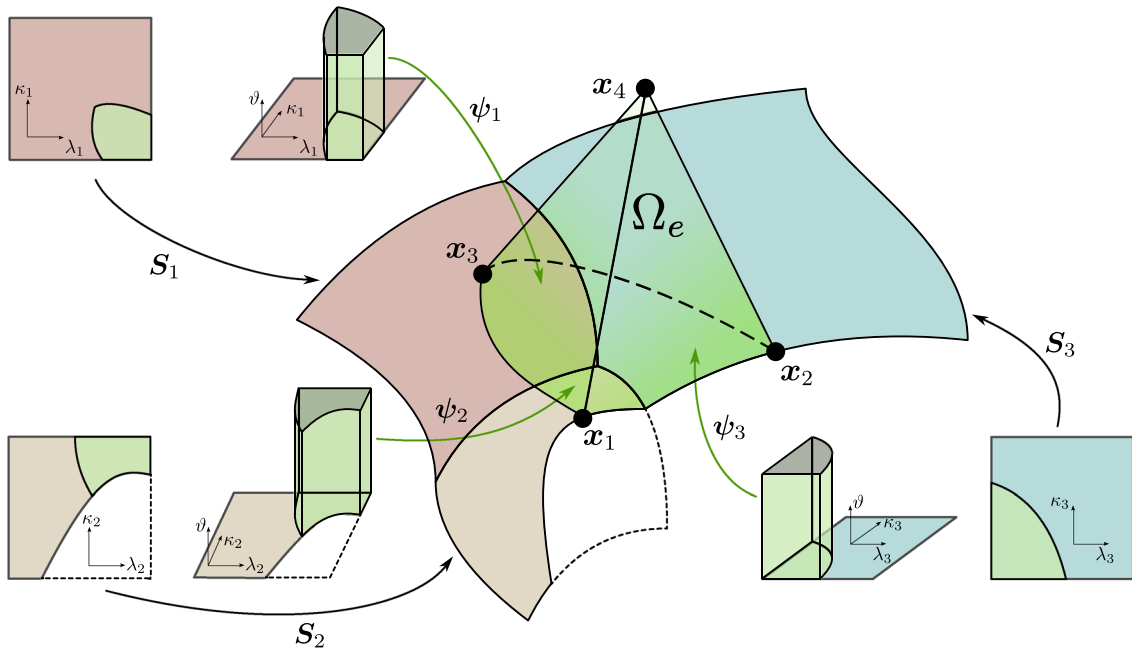


Fig. 4 Illustration of the piecewise NEFEM geometry mappings for a tetrahedral element with one face defined on three NURBS surfaces ( $S_1, S_2, S_3$ ), where  $S_2$  is a trimmed surface. Note that the tetrahedral element comprises three pieces

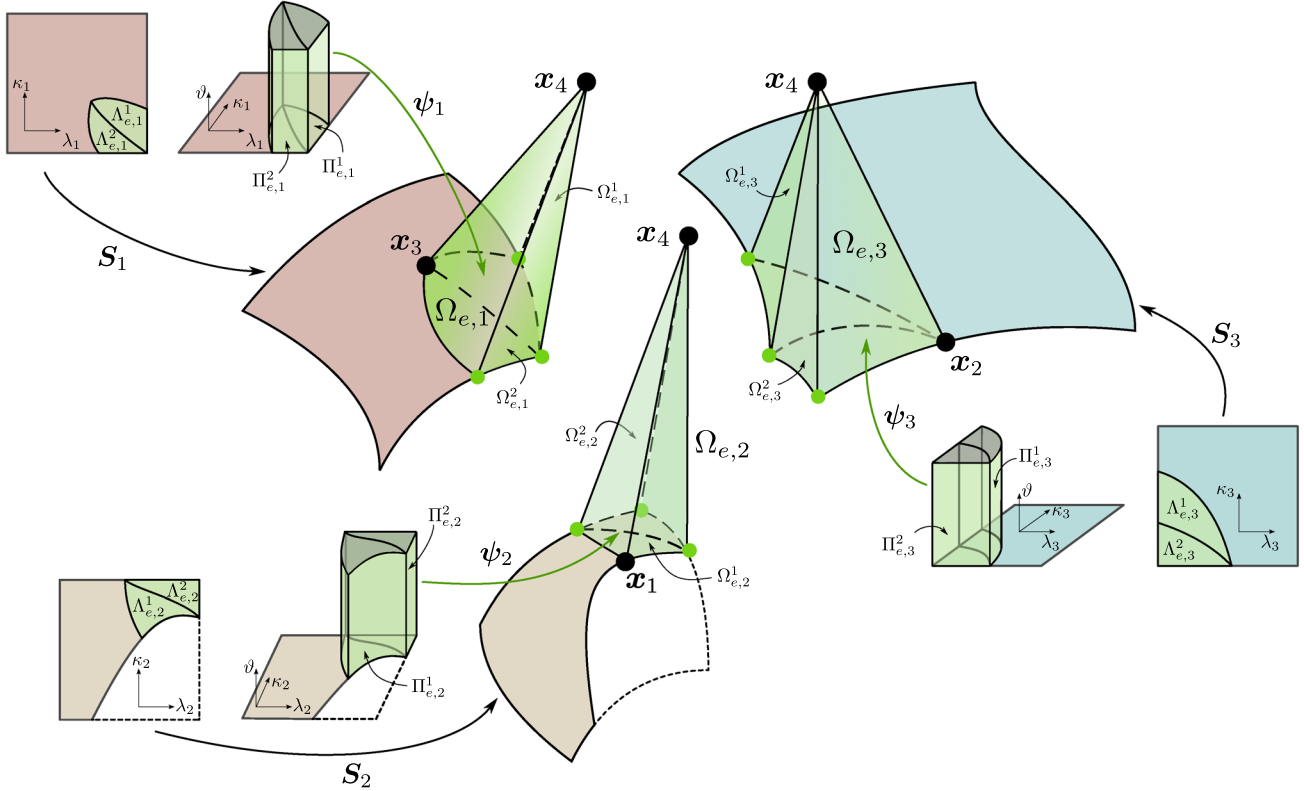


Fig. 5 Illustration of the subdivision for the tetrahedral element  $\Omega_e$  in Fig. 4

definitions of the geometric entities are introduced to facilitate the creation of a NEFEM volume element which requires subdivision.

In practice, the piecewise definition of a NEFEM element face, as the bottom face shown in Fig. 4, is described by a subdivision based on the surfaces. This leads to a *sub-mesh* with each constituent cell sticking onto one of the involved surfaces. Note that the sub-mesh terminology is limited to the surface mesh. Using the sub-mesh as the underlying faces, the tetrahedral element can be further subdivided. Figure 5 illustrates the subdivision of the example tetrahedral element  $\Omega_e$  in Fig. 4 into three subdomains  $\Omega_{e,i}$ , where  $i = 1, 2, 3$  denotes the  $i$ -th prism in the corresponding parameter domain  $\Pi_{e,i}$ . In general, a subdivided element is expressed as

$$\Omega_e = \bigcup_{i=1}^{n_{\text{sdo}}} \Omega_{e,i}, \tag{2}$$

where  $n_{\text{sdo}}$  is the number of physical subdomains according to the embedded NURBS geometry. To maintain a consistent meshing strategy with simplified data structure storing the element information, each subdomain is further subdivided into several *subcells* in tetrahedral shape. For instance, as shown in Fig. 5, the subdomain  $\Omega_{e,2}$  is further divided into subcells  $\Omega_{e,2}^1$  and  $\Omega_{e,2}^2$ . This is facilitated by subdividing the corresponding parameter domains, such as splitting  $\Pi_{e,2}$  into  $\Pi_{e,2}^1$  and  $\Pi_{e,2}^2$ , as well as the parameter domains of the underlying NURBS surface, such as dividing  $\Lambda_{e,2}$  into  $\Lambda_{e,2}^1$  and  $\Lambda_{e,2}^2$ . Eventually, the tetrahedral element  $\Omega_e$  is divided into six subcells in total and all in tetrahedron shape.

In this work, the geometric entities of NEFEM elements are defined at the subcell level. This is consistent with the NEFEM surface mesh, so that similar data structure can be used. Due to the prismatic structure of the geometric mapping (1), all edges of the subcell connecting the top node  $x_4$  are straight lines.

### 2.3.1 Geometric definition of subedges

For a non-self-intersecting NURBS surface  $S$ , any of its boundary or trimming curves  $C(\lambda)$  can be projected back to its parametric space to obtain the parametric curve or *p-curve*  $c(\lambda) \in \mathbb{R}^2$ , which is defined as

$$c(\lambda) = S^{-1}(C(\lambda)). \tag{3}$$

The p-curve may be further mapped to the physical space by the NURBS parametrisation  $S$ , therefore the NURBS curve  $C(\lambda) \in \mathbb{R}^3$  is also referred to as the *physical p-curve*.

The edges of a subcell, firstly defined in the parametric space, are also known as the *parametric subedges*. In particular, considering the prismatic structure of the parameter domain, viz.  $\Pi_e = \Lambda_e \times [0, 1]$ , only three scenarios are considered for a parametric subedge:

1. The subedge is interior to the underlying parametric space  $\Lambda_{e,i}$ , i.e. with at most one vertex on a p-curve;
2. The subedge is on the boundary or trimming p-curve of the parametric space, i.e. with both vertices on a p-curve;
3. The subedge is a ridge of the prism.

The first type, parametric subedges with at most one vertex on a p-curve are defined, using a cubic description, namely

$$\begin{aligned} \Phi : [0, 1] &\longrightarrow \Gamma_\lambda \in \Lambda_{e,i}; \\ \xi &\longmapsto \Phi(\xi) := \sum_{k=1}^4 \lambda_k N_k(\xi), \end{aligned} \tag{4}$$

where  $\lambda_k = (\lambda_k, \kappa_k)$  for  $k = 1, \dots, 4$  are the four points in  $\Lambda_{e,i}$  that define the edge  $\Gamma_\lambda$ , and  $N_k(\xi)$  are the four one-dimensional cubic Lagrange polynomials defined on the reference interval  $[0, 1]$ .

The second type, a parametric subedge with both vertices on a p-curve, is simply defined by trimming the parent p-curve as

$$\begin{aligned} \Psi : [0, 1] &\longrightarrow \Gamma_\lambda \subset \partial\Lambda_{e,i}; \\ \xi &\longmapsto \Psi(\xi) := C((1 - \xi)\lambda_1 + \xi\lambda_2), \end{aligned} \tag{5}$$

where  $C(\lambda) : [0, 1] \rightarrow \Lambda_{e,i}$  is the parametrisation of the p-curve to which the edge  $\Gamma_\lambda$  belongs, and  $[\lambda_1, \lambda_2]$  is the trimming interval for the subedge.

The last type, parametric subedges that are ridges of the prism, are defined the same way as the first type, but with the parametric space  $\Lambda_{e,i}$  replaced by the parameter domain of the prism  $\Pi_{e,i}$ , and thus the defining points  $\lambda_k = (\lambda_k, \kappa_k, \vartheta_k) \in \Pi_{e,i}$ . It is noted the internal points, namely  $\lambda_3$  and  $\lambda_4$ , are restricted to the straight line defined by  $\lambda_1$  and  $\lambda_2$ .

The parametric subedges are then mapped to the physical space by the geometric mapping  $\psi$ , to obtain the *physical subedges*.

### 2.3.2 Geometric definition of subfaces

Similarly, the faces of a subcell are firstly defined in the parametric space as the *parametric subfaces*, and then mapped by  $\psi$  to obtain the *physical subfaces*. There are two types of parametric subfaces of interest:

1. The subface is part of the underlying parametric space  $\Lambda_{e,i}$  for a boundary surface, namely  $\Lambda_{e,i}^j \subset \Lambda_{e,i}$ ;
2. The subface is an extrusion face of the prismatic parameter domain  $\Pi_{e,i}$ , namely  $\partial\Pi_{e,i}^j \subset \partial\Pi_{e,i}$ .

The first type, a parametric subface  $\Lambda_{e,i}^j \subset \Lambda_{e,i}$  is defined using the mapping

$$\begin{aligned} \Theta : [0, 1]^2 &\longrightarrow \Lambda_{e,i}^j(\xi, \eta) \longmapsto \Theta(\xi, \eta) \\ &:= (1 - \eta)\Psi_1(\xi) + \xi\Psi_2(\eta) \\ &\quad + (1 - \xi)\Psi_3(\eta) - (1 - \xi) \\ &\quad (1 - \eta)\mathbf{x}_1 - \xi(1 - \eta)\mathbf{x}_2, \end{aligned} \quad (6)$$

where  $\Psi_1$  is the parametrisation of the first subedge, connecting nodes 1 and 2 of the subface,  $\Psi_2$  is the parametrisation of the second subedge, connecting nodes 2 and 3 of the subface and  $\Psi_3$  is the parametrisation of the third subedge, connecting nodes 1 and 3 of the subface. This mapping can be seen as a particular case of the blending function method [39]. The corresponding physical subfaces are simply defined as  $\partial\Omega_{e,i}^j = \mathbf{S}(\Lambda_{e,i}^j)$ .

The second type, a parametric subface  $\partial\Pi_{e,i}^j \subset \partial\Pi_{e,i}$  is defined using the mapping

$$\begin{aligned} \Upsilon : [0, 1]^2 &\longrightarrow \partial\Pi_{e,i}^j \\ (\xi, \eta) &\longmapsto \Upsilon(\xi, \eta) := \Psi(\xi, \eta), \end{aligned} \quad (7)$$

where  $\Psi(\xi) \in \Lambda_{e,i}$  is the parametrisation of the underlying subedge. Note that this type of parametric subface is in quadrilateral shape, but the mapped physical subface is in triangular shape.

### 2.3.3 Geometric supporting points

For NEFEM elements containing an edge or a face traversing multiple curves or surfaces, the intersection of the edge and a NURBS curve is inevitable. To account for the intersection between the edge and a geometric curve, an enhanced edge description is proposed in [37] by introducing *geometric supporting points* or *GS-points*. The set of GS-points associated to one element  $\Omega_e$  is given by the intersections of  $\partial\Omega_e$  with the physical p-curves plus the intersections between physical p-curves that are inside  $\Omega_e$ . In this paper, a convention is introduced that all vertex nodes are rendered by black dots, while all GS-points are shown as green dots, as illustrated in Fig. 4 and 5.

It is worth emphasising that GS-points are introduced to formally define the geometry of a NEFEM element and its subcells. They do not induce any new degrees of freedom to a solver that adopts the NEFEM meshes. Similarly, subcells are only introduced due to the piecewise nature of a NEFEM

surface element, but the only element that being used by the solver is  $\Omega_e$ . Because of the geometry-defining feature of the GS-points, they will be used for devising piecewise quadratures for numerical integration over the faces traversing multiple surfaces and the associated elements. Further usage of the GS-points during the mesh generation process will be discussed in Sect. 3 and 4.

## 3 NEFEM surface mesh generation

In this section, the generation of NEFEM triangular surface mesh is briefly recalled. It is worth emphasising that it is assumed that the upstream CAD geometry is watertight. The NEFEM surface mesh is a prerequisite for generating the volume mesh, as it provides the boundary discretisation tailored for NEFEM solvers. The surface mesh is desired to satisfy the following requirements:

1. The characteristic element size is dominated by the user-specified spacing, and it is not restricted by the size of geometric features in the CAD model;
2. The surface mesh must not introduce geometric discretisation error as it encapsulates the NURBS definition of the geometry;
3. The surface elements should pass visibility checks to enable the efficient creation of volume elements that avoids self-intersections.

The first two requirements have been addressed in previous work [37]. Note that the geometry entities introduced in Sect. 2.3 are compatible with the second requirement. The last requirement is posed here to facilitate the volume mesh generation, and it will be further detailed in Sect. 3.2.

The NEFEM surface mesh generation starts from an initial surface mesh obtained by a standard mesh generator with a user-defined mesh size. Despite that this initial mesh is likely to contain numerous elements violating the user-specified spacing, it is required to be watertight and free of self-intersections. A remeshing is then performed on the initial surface mesh, with a dedicated process to allow creating elements traversing multiple surfaces around geometric features, so that the element sizes become compliant with the user specification.

During the surface mesh generation, GS-points are typically created using operations such as edge collapse, edge split or edge flip. Besides, the GS-points can slide along the parent intersection curve, so that the element could achieve a better quality.

Local mesh refinement can be performed by specifying a spacing function, as done with traditional mesh generators.



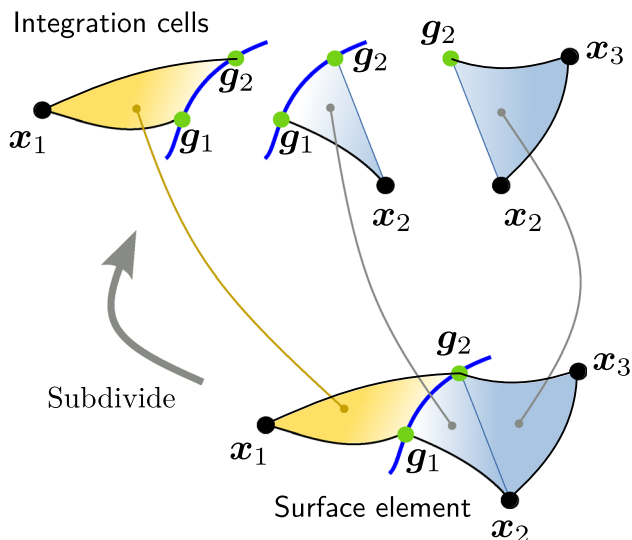
Examples in [37] show the ability to produce locally refined NEFEM surface meshes.

### 3.1 The sub-mesh

As detailed in Sect. 2.3, a sub-mesh is required for the definition of a surface element or a face of a volume element that traverses multiple surfaces. The sub-mesh, together with the GS-points, is used for quadrature as it is a subdivision of the traversing surface element which naturally forms the *integration cells*. The sub-mesh may also be used to represent the physical interface, such as the interface of different materials, inside a NEFEM element [40]. It is worth noting that an integration cell cannot traverse surfaces, and an element traversing multiple surfaces must contain at least two integration cells.

In addition, the sub-mesh also plays an important role during the NEFEM mesh generation. It is well-known that operations such as edge split and edge flip are commonly involved in a mesh generation process. Dissimilar to the operations in standard mesh generators, the sub-mesh is required during the NEFEM mesh generation because the operations may involve multiple surfaces and their intersection curves. For instance, when flipping an edge between two NURBS-enhanced triangular elements, the subdivision of both elements is necessary for searching the new diagonal traversing multiple surfaces. In this procedure, the GS-points also play a role as the nodes for the sub-mesh of each element. Detailed discussion on this procedure can be found in [37].

A typical example for the sub-mesh of a NEFEM surface element is presented in Fig. 6. The triangular surface



**Fig. 6** Sub-mesh of a typical NEFEM surface element. Integration cells belong to different surfaces are filled with distinguishing colours. The intersection curve is coloured in blue

element is defined with three nodes  $x_1$ ,  $x_2$  and  $x_3$ . The two edges  $E(x_1, x_2)$  and  $E(x_1, x_3)$  are traversing a NURBS curve rendered in blue, inducing two GS-points,  $g_1$  and  $g_2$ , to register the edge-curve intersections. The two GS-points and three nodes have defined the vertices for three integration cells that belong to two different surfaces rendered in distinguished colours.

### 3.2 Additional validity check

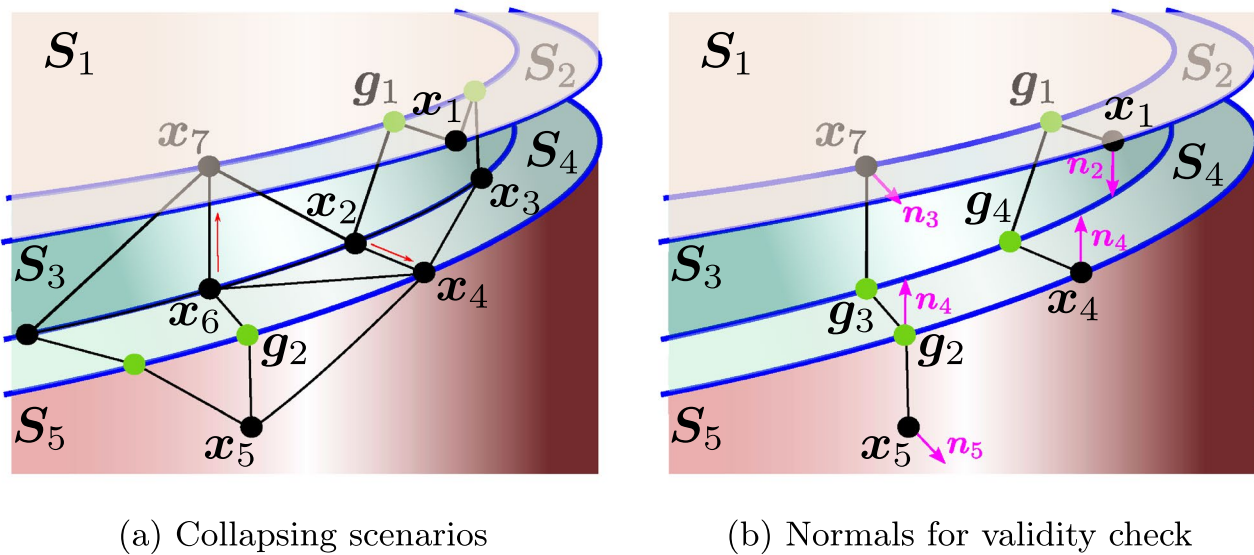
During the NEFEM surface mesh generation, validity checks have been introduced as it is detailed in [37]. To facilitate the creation of volume elements in the next stage, additional validity checks must be appended to the generation process of NEFEM surface elements.

Firstly, an additional check is performed prior to collapsing an edge of a NEFEM surface element that traverses multiple intersection curves or surfaces. This check is crucial to minimise the cost of future visibility checks in Sect. 4.3 to avoid possible self-intersection in the corresponding NEFEM volume element. Secondly, an intersection check between a surface element edge and a NURBS curve is performed after having created or updated the NEFEM elements. This check is dedicated to avoiding the surface element edge penetrating surfaces, causing a self-intersecting mesh. In addition to this check, it will also try to fix the self-intersection by curving the subedges.

The first check is performed when trying to create a new NURBS-enhanced edge that comprises several subedges. This new edge is traversing multiple surfaces that are connected with only  $C^0$  continuity. In other words, the normals to the surfaces along the edge do not transit smoothly, and sharp dihedral angles will occur at a node or a GS-point, causing difficulties for future generation of a valid volume element. To tackle this, the angles between normals to the surfaces at each node and each GS-point are computed and checked, as detailed in Algorithm 1. As illustrated in Fig. 7, a local feature appearing to be a U-shaped channel involves a set of five surfaces  $\mathcal{S} := \{S_i\}$  for  $i = 1, 2, \dots, 5$ , where all dihedral angles between adjacent surfaces are  $90^\circ$ . When trying to collapse a short edge inside the channel, all related surface normals at the involved nodes of the sub-mesh are compared with the normal at the target node. The criterion for the validity check is chosen as

$$\begin{cases} n_i \cdot n_j \geq -1/2, & \Rightarrow \text{pass;} \\ n_i \cdot n_j < -1/2, & \Rightarrow \text{fail,} \end{cases} \quad (8)$$

where  $n_i$  and  $n_j$  are the normals to surfaces  $S_i$  and  $S_j$ , respectively. All the normals are computed locally at the corresponding nodes of the sub-mesh.



**Fig. 7** Illustration of the validity check at a U-channel feature. **a** Two edge collapsing scenarios: from  $x_6$  to  $x_7$ , and from  $x_2$  to  $x_4$ . **b** Proposed new NURBS-enhanced edges, showing selected normals for validity checks

**Algorithm 1** Validity check for collapsing a NEFEM surface element edge.

---

**Input:** Edge to collapse  $E(x_b, x_t)$ , with base node  $x_b$ , target node  $x_t$   
**Output:** Boolean value `isValid`

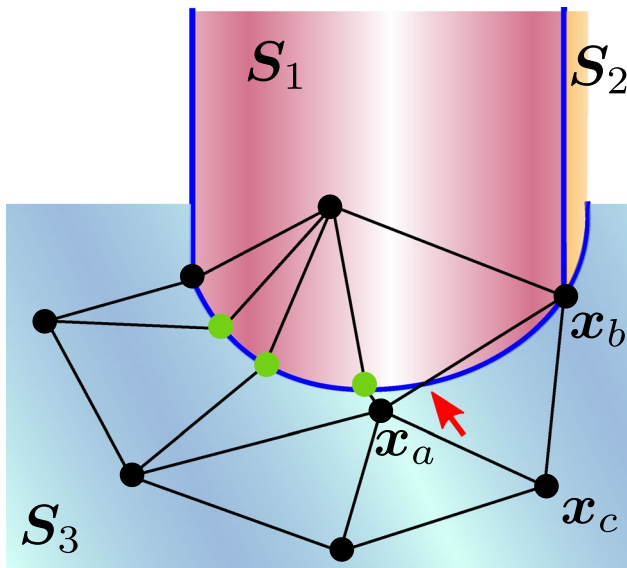
- 1 Collect the set,  $\mathcal{S}_t$ , of all parent surfaces of the target node;
- 2 Collect the set,  $\mathcal{S}_c$ , of all traversed surfaces for the collapse;
- 3 **for**  $S_i \in \mathcal{S}_t$  **do**
- 4     **if**  $S_i \notin \mathcal{S}_c$  **then**
- 5         Remove  $S_i$  from  $\mathcal{S}_t$ ;
- 6     **end if**
- 7 **end for**
- 8 Identify the involved normals at the target node  $\mathcal{N}_t$ ;
- 9 Collect the set,  $\mathcal{N}_c$ , of all normals at involved sub-nodes for the collapse;
- 10 Initialise `isValid`  $\leftarrow$  `true`;
- 11 **for**  $n_i \in \mathcal{N}_t$  **do**
- 12     **for**  $n_j \in \mathcal{N}_c$  **do**
- 13         **if**  $n_i \cdot n_j < -1/2$  **then**
- 14             `isValid`  $\leftarrow$  `false`;
- 15             **return**;
- 16         **end if**
- 17     **end for**
- 18 **end for**

---

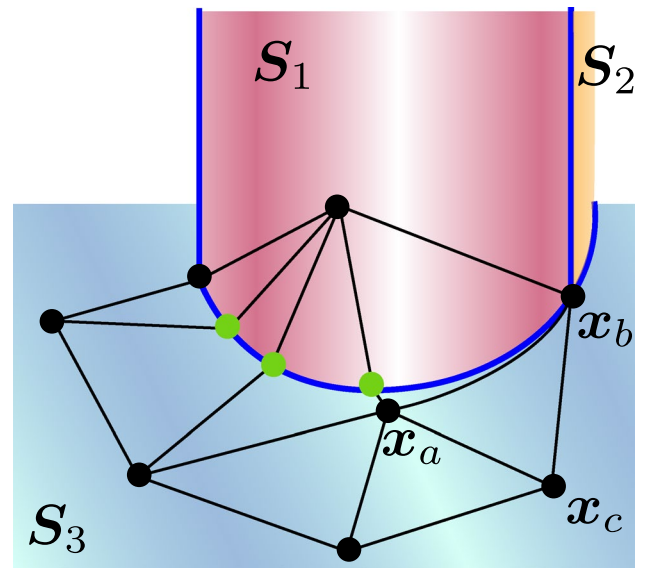
Figure 7b presents two examples when testing possible new NURBS-enhanced edges before the edge collapse. The first candidate edge,  $E(x_5, x_7)$ , is obtained from collapsing node  $x_6$  to  $x_7$ , and the normal  $n_3$  at the target node  $x_7$  is compared with all normals involved in this collapse, such as  $n_4$  at GS-point  $g_2$  and  $n_5$  at node  $g_5$ . This case will pass the validity check. A second option involves the candidate NURBS-enhanced edge  $E(x_1, x_4)$ , the normal  $n_4$  at target node  $x_4$  is opposite to  $n_2$  at  $x_1$ . Therefore, the

second configuration will fail the validity check and thus the collapse of edge  $E(x_2, x_4)$  will be prevented.

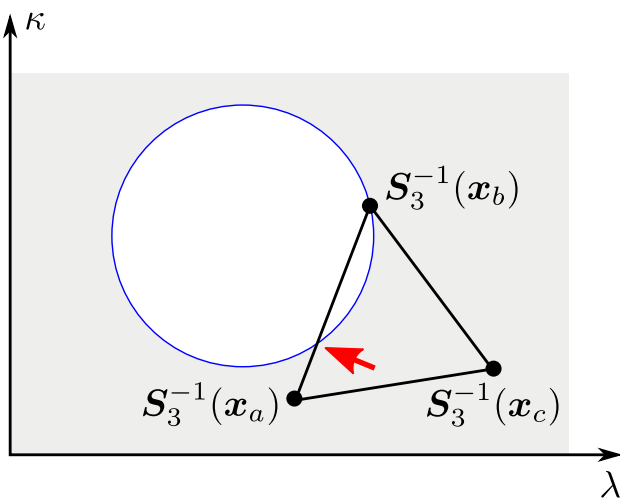
In the rare case that a subedge intersects with an intersection curve and penetrates a neighbour surface, as shown in Fig. 8a, the second validity check will detect and fix this by curving the subedge. The intersection between edge  $E(x_a, x_b)$  and the intersection curve of surface  $S_1$  and  $S_3$  can easily be detected by seeding a number of sampling points along the intersection curve. This



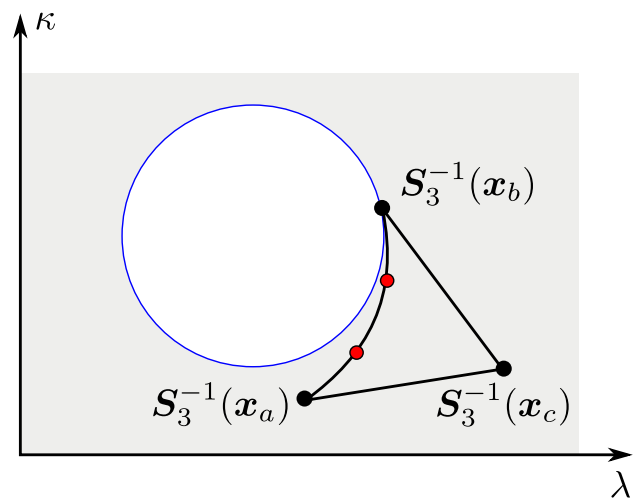
(a) Edge intersecting with curve



(b) Fixed by curving the edge



(c) Parameter space of  $S_3$



(d) Isoparametric cubic edge

**Fig. 8** Illustration of the self-intersection fix at the bottom of a cylindrical feature. **a** Edge  $E(x_a, x_b)$  intersecting with an intersection curve at the red arrow. **b** Fixed the intersection by curving  $E(x_a, x_b)$  within

surface  $S_3$ . **c** The parameter space of surface  $S_3$  featuring intersection with the trimming circle. **d** The cubic curve to fix the intersection

type of self-intersection occurs because surface  $S_3$  is trimmed by a circle. The trimming circle is the image of a circle in the parameter space of the NURBS surface  $S_3(\lambda, \kappa)$ , as shown in Fig. 8c. A simple fix to this problem is performed by replacing the straight edge  $E(x_a, x_b)$

in the parametric space by a cubic curve, as shown in Fig. 8d. The fixed scenario after the second validity check is shown in Fig. 8b. Note that this check and fix is performed at the subedge level.

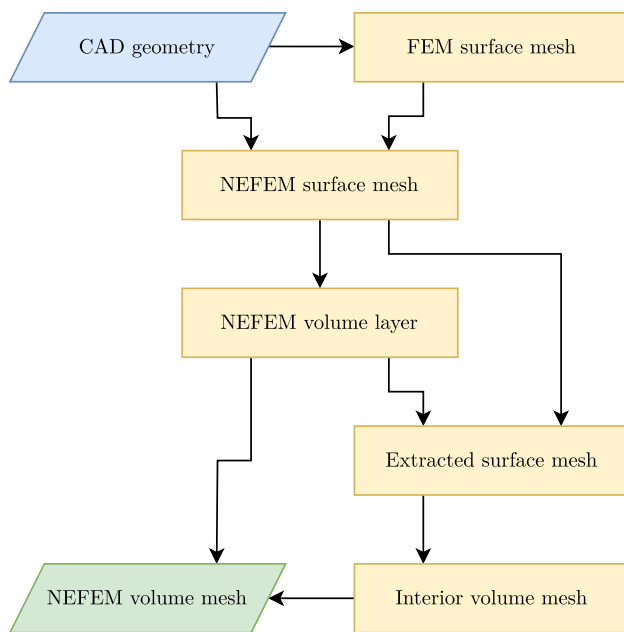


Fig. 9 NEFEM volume mesh generation procedure

## 4 NEFEM volume mesh generation

This section presents the main purpose of this paper on the generation of NEFEM volume meshes comprised by tetrahedral elements. According to the geometric entity that is part of the B-rep, the tetrahedral element that is of interest falls into one of the two types:

- An element with at least one face located on the boundary.
- An element with at least one edge but with no faces located on the boundary.

An element of each type is further classified in terms of the number of intersection curves and surfaces of the CAD model that it involves. Elements with faces or edges not traversing multiple surfaces are grown using the same techniques available in standard mesh generators. A flag is assigned to each tetrahedral element for the purpose of accounting for the NURBS B-rep by the solvers, as special care must be taken for elements with the traversing faces or edges.

### 4.1 Volume meshing strategy

The volume mesh generation starts from a valid NEFEM surface mesh which already encapsulates the GS-points as well as the sub-mesh for traversing surface elements.

It is appreciated that the NEFEM elements, whether or not traversing multiple curves or surfaces, are only required

at the boundary. Therefore, the strategy for the generation of a NEFEM volume mesh is to first grow a layer of NURBS-enhanced volume elements that covers the featured surface. Next, the exterior facets of the layer of grown volume elements are extracted to form a new surface mesh. This extracted surface mesh will only contain standard triangle elements, so that it can be sent to a standard volume mesh generator to obtain the volume elements of the remaining part of the domain. Finally, the NEFEM volume mesh is obtained by stitching the NEFEM element layer and the standard interior elements. In this work, the FLITE mesh generator [11] is used to create standard meshes. The overall procedure is illustrated in Fig. 9.

**Remark 1** The stitching of boundary layer mesh and interior mesh, as discussed in [41, 42], is an established procedure in standard mesh generation. The presented work is dedicated to generating the geometric-persistent mesh layer that is valid for NEFEM solvers. As the interior mesh is generated after the boundary layer mesh, the stitching of the two meshes is naturally done with merely renumbering the corresponding nodes. Besides, a flag is associated to each element to indicate whether it is a NEFEM element or a standard element. This flag will be used to distinguish the two classes of element by the NEFEM solver.

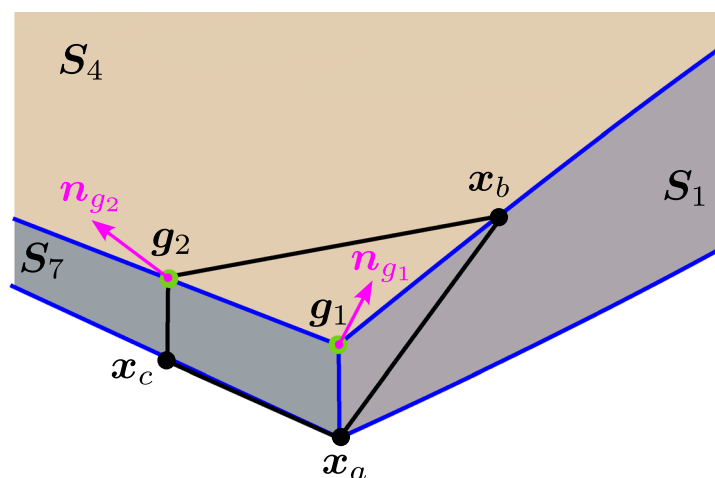
**Remark 2** In the case of a bounding box is required to generate the volume mesh, the bounding box is not considered for generating the NEFEM volume element layer due to its simplicity, but will be taken into account for the standard volume mesh generation for the interior elements.

### 4.2 Growing volume elements on the boundary

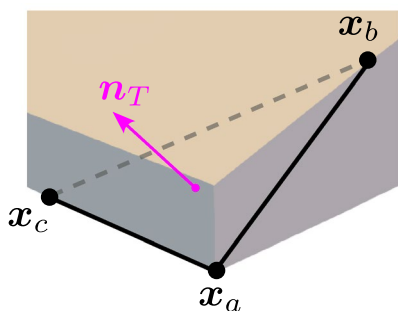
To guide the growth of volume elements into the three-dimensional domain, the normal vectors are firstly computed based on the NEFEM surface mesh. Unlike standard triangle elements, a NEFEM triangle element can have a non-unique definition of its normal as it may traverse multiple surfaces. Thus, it is not trivial to evaluate the normal to a face or its edges.

As required in Sect. 3.1, an integration cell is associated to a unique parent surface. Therefore, for each integration cell, the normal vector is unique to that parent surface, and a smoothed normal can be obtained at each node of the sub-mesh that may be a node of the mesh or a GS-point. This also implies that a sequence of normal vectors can be extracted along an element edge as it traverses multiple curves or surfaces. Several smoothing options have been tested, such as surface-based averaging, weighted averaging and Laplacian. It is found that the surface-based averaging provides satisfactory normal vectors in the tested geometries.

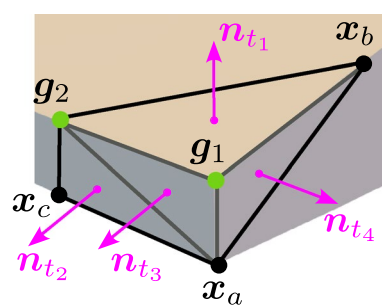




(a) Normal vectors at GS-points



(b) Normal vector at plane element centroid



(c) Normal vectors at integration cell centroids

Fig. 10 Choices of normal vectors to grow a tetrahedral element

Several possible choices of the normal to guide the growth of volume element based on a typical NEFEM surface element are illustrated in Fig. 10. The surface element with vertices  $\{x_a, x_b, x_c\}$  traverses surfaces  $\{S_1, S_4, S_7\}$ . The smoothed normal at GS-points  $g_1$  and  $g_2$  are shown in Fig. 10a, the naive normal evaluated using only the vertices is plotted in Fig. 10b at the centroid of a planar triangle determined by the three vertices. In contrast, Fig. 10c shows the four normal associated to each integration cell depicted at the corresponding centroid.

The two major process used to grow volume elements based on the surface mesh are edge-based and face-based. The first process is to grow a pair of tetrahedra based on an edge of the surface mesh. The second process is to grow a tetrahedron based on a surface element. For both processes, the key is to find a suitable guiding normal.

The first attempt involves a loop through all edges in the surface mesh, and the check of dihedral angle  $\theta$  between faces connected by the edge. For an edge between two surface elements, four vertices are involved in the process,

including the vertices defining the edge, and other two vertices opposing each other. The two triangular faces may be collapsed and the two opposing vertices be merged when  $\theta$  is small. Otherwise a tetrahedron will be created by linking the two opposing vertices, creating a new edge to form the tetrahedron. In most cases, it is necessary to find a normal on the edge to create a top node above the edge and try to link it with all vertices and GS-points of the two triangles sharing the edge. During this linking process, the validity checks are performed: a self-intersection check via evaluating the volume of the newly formed subcell tetrahedra, a general edge-face intersection check between the new edges and existing faces as well as between the new faces and existing edges. The validity checks are detailed in Sect. 4.3. If a self-intersection is identified, the position of the top vertex is tuned by sliding along the normal vector as well as sliding the base point of the normal vector along the edge to find another suitable location that is free from self-intersection. The procedure is detailed in Algorithm 2. After having grown a pair of

tetrahedra, the connectivity of the volume mesh and the surface mesh are updated by removing the closed faces and edges. The process is repeated until all edges are closed or leaving few edges cannot be closed.

Figure 11 illustrates a typical scenario for volume element growth. Three surface elements are traversing surfaces  $S_1$  and  $S_2$  that are rendered in red and yellow. Note that the connection between  $S_1$  and  $S_2$  is of  $C^0$  continuity, and exhibiting

**Algorithm 2** Growing volume elements based on an edge.

---

**Input:** Edge to grow  $E(\mathbf{x}_1, \mathbf{x}_2)$  a pair of tetrahedron, with the opposing nodes  $\mathbf{x}_3$  and  $\mathbf{x}_4$ , and the adjacent surface elements  $\Lambda_L := T(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$  and  $\Lambda_R := T(\mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_4)$

**Output:** Position of the new top node  $\mathbf{x}_5$ , thus forming the two tetrahedra  $\Omega_L(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5)$  and  $\Omega_R(\mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_5)$

- 1 Evaluate the dihedral angle  $\theta$  between  $\Lambda_L$  and  $\Lambda_R$ ;
- 2 Find the required spacing  $h$  for the new top node  $\mathbf{x}_5$ ;
- 3 Evaluate the distance  $d$  between  $\mathbf{x}_3$  and  $\mathbf{x}_4$ ;
- 4 **if**  $\theta < 5^\circ$  **then**
- 5 Merge  $\Lambda_L$  and  $\Lambda_R$  into one triangle;
- 6 **return**;
- 7 **end if**
- 8 **if**  $5^\circ \leq \theta < 120^\circ$  and  $d < 1.5h$  **then**
- 9 Propose a tetrahedron by creating the edge  $E(\mathbf{x}_3, \mathbf{x}_4)$ ;
- 10 Check the validity for  $\Omega(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$  considering the tetrahedral subcells to form;
- 11 **if** *Validity check passed* **then**
- 12 Create the volume element  $\Omega(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ ;
- 13 **end if**
- 14 **else**
- 15 Collect a set of candidate guiding normals  $\mathcal{N} := \{ \mathbf{n}_i \}$ ;
- 16 **for**  $\mathbf{n}_i \in \mathcal{N}$  **do**
- 17 Propose a new vertex  $\mathbf{x}_5$  above  $E(\mathbf{x}_1, \mathbf{x}_2)$ ;
- 18 Propose two tetrahedra  $\Omega_L(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5)$  and  $\Omega_R(\mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_5)$ ;
- 19 Check the validity for  $\Omega_L$  and  $\Omega_R$  considering the tetrahedral subcells to form;
- 20 **if** *Validity check passed* **then**
- 21 Create the volume elements  $\Omega_L$  and  $\Omega_R$ ;
- 22 **end if**
- 23 **end for**
- 24 **end if**
- 25 Update connectivity of the volume mesh;
- 26 Update connectivity of the sub-mesh by removing the closed subcell faces;

---

a sharp dihedral angle. During an edge-based loop, a suitable normal vector  $\mathbf{n}_{g_7}$  is found at GS-point  $\mathbf{g}_7$  of edge  $E(\mathbf{x}_2, \mathbf{x}_4)$ , as depicted by Fig. 11a. Two tetrahedral elements, coloured in blue and green, are grown with the guidance of  $\mathbf{n}_{g_7}$ , sharing the same new vertex  $\mathbf{x}_6$ , as shown in Fig. 11b. Other tetrahedra are grown during this edge-based loop, including the one with a new vertex  $\mathbf{x}_7$ . A second edge loop will be performed to close the edges between two grown tetrahedra where the dihedral angles between element faces are checked. In the scenario presented in Fig. 11c, a new tetrahedron will be created by simply linking the existing vertices  $\mathbf{x}_6$  and  $\mathbf{x}_7$ .

A second attempt is another loop through the unclosed surface elements to grow a tetrahedron based on a surface element. The normal vector is evaluated at the centroid of the surface element, and the new vertex is proposed above the surface element. The new vertex might be substituted by an existing node nearby when the distance is sufficiently small. The normal vector is then used to link the new vertex with all vertices and GS-points of the surface element. The validity checks are performed in the same way as the edge-based process. The procedure is detailed in Algorithm 3.

**Algorithm 3** Growing volume elements based on a face.

---

**Input:** Face to grow  $\Lambda := T(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$  a tetrahedron  
**Output:** Position of the new top node  $\mathbf{x}_4$ , thus forming the two tetrahedra  $\Omega(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$

- 1 Collect a set of candidate guiding normals  $\mathcal{N} := \{\mathbf{n}_i\}$ ;
- 2 **for**  $\mathbf{n}_i \in \mathcal{N}$  **do** Normal loop
  - 3 Propose a new vertex  $\mathbf{x}_4$  above  $E(\mathbf{x}_1, \mathbf{x}_2)$ ;
  - 4 Find the required spacing  $h$  for  $\mathbf{x}_4$ ;
  - 5 Collect a set of existing nearby nodes  $\mathcal{X} := \{\mathbf{x}_i\}$ ;
  - 6 Append  $\mathbf{x}_4$  to  $\mathcal{X}$ ;
  - 7 **for**  $\mathbf{x}_i \in \mathcal{X}$  **do**
    - 8 Evaluate the distance  $d$  between  $\mathbf{x}_i$  and  $\mathbf{x}_4$ ;
    - 9 **if**  $d < 0.5h$  **then**
      - 10 Propose a tetrahedron  $\Omega(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_i)$ ;
      - 11 Check the validity for  $\Omega$  considering the tetrahedral subcells to form;
      - 12 **if** *Validity check passed* **then**
        - 13 Create the volume elements  $\Omega$ ;
        - 14 Update connectivity of the volume mesh;
        - 15 Update connectivity of the sub-mesh by removing the closed subcell faces;
        - 16 **break** Normal loop;
      - 17 **end if**
    - 18 **end if**
  - 19 **end for**
- 20 **end for**

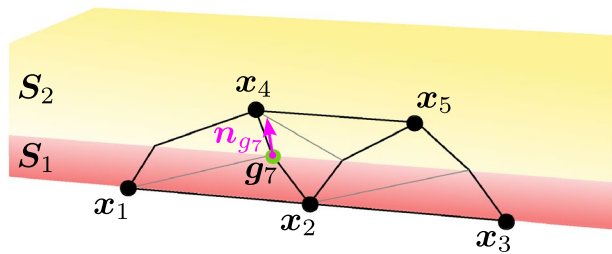
---

As a tetrahedral element is grown based on a surface element, such as shown in Fig. 5, it inherits the subdivision of the surface element in the way that three sub-tetrahedra are grown to form the integration cells for computing quadrature over the NURBS-enhanced tetrahedron. This can also be considered as a straightforward subdivision of the tetrahedron element with the guidance of the sub-mesh on a traversing face, as illustrated in Fig. 12.

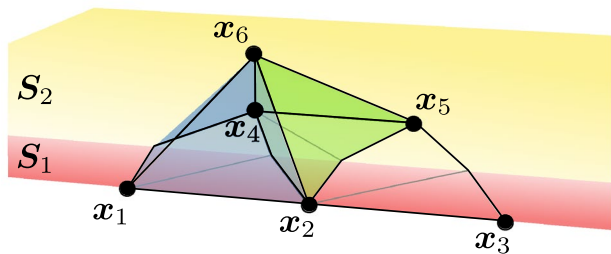
It is worth noting that the tetrahedron grown from a surface element is not necessarily a NEFEM element. This is because the surface element may not traverse multiple surfaces, and the boundary surface is planar, thus the tetrahedron is not required to be a NEFEM element. In this case, the tetrahedron is grown as a standard tetrahedron with the subcell being itself, and is still included in the NEFEM volume layer. The validity checks are performed in the same way as the introduced process.

### 4.3 Validity checks for volume elements

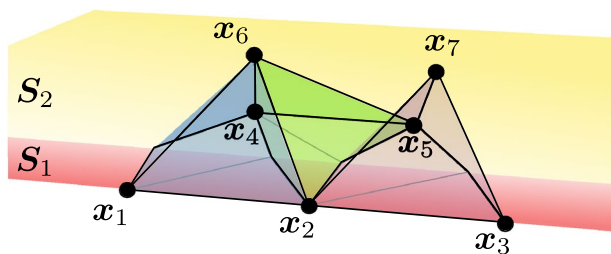
Two major validity checks are performed during the growth of volume elements. The first check, also known as the *visibility check* [36], is to ensure that the new volume



(a) Normal vectors at a GS-point to guide the growth



(b) Two tetrahedra grown

(c) Other grown tetrahedron, vertices  $x_6$  and  $x_7$  to be linked to form a new tetrahedron**Fig. 11** Typical scenario to grow tetrahedral elements

element is free from self-intersection. The second check is to ensure that the new volume element is free from intersection with the existing mesh entities. This section will focus on the first check only, as the second check is similar to the validity checks in established standard mesh generation process.

Taking the creation process of triangle to tetrahedron for example, the objective is to ensure the top vertex of the tetrahedron is visible to any point in the base sub-mesh. When the visibility requirement is met, all ridges of the tetrahedron, excluding the ones corresponding to edges of the base triangle, will be straight, and this enables an efficient subdivision of the volume element into volumetric integration cells.

It is worth noting that the visibility requirement is not mandatory for a valid NEFEM element, a self-intersecting element can be fixed by curving the interior edges to maintain the validity. In this case, the interior physical subedges can be curved cubically using the isoparametric mapping of (4) in Sect. 2.3.1. However, the strategy presented here tries to maintain the maximum number of interior edges as straight with the objective to accelerate the solver.

At some convex geometric features, special care has to be taken to ensure the visibility from the top node to the bottom sub-nodes. The feature of a sharp step in Fig. 13 presents a scenario in which a violation occurs, resulting in a self-intersecting NEFEM volume element.

The volume element shown in Fig. 13a is based on the surface element in Fig. 13c, which traverses surfaces  $\{S_1, S_2, S_3\}$  as well as intersection curves  $\{C_1, C_2\}$ , and four GS-points have been included. It can be seen that the dihedral angles on the intersection curves are considerably sharp and include both convex and concave instances. Besides, surface  $S_2$  appears to be a narrow strip folding between surfaces  $S_1$  and  $S_3$ . A surface integration cell with nodes  $\{g_1, x_2, g_2\}$ , as shown in Fig. 13d, forms the bottom face of the volume integration cell illustrated in Fig. 13b. As partially highlighted by red dashed lines, the edges  $E(x_4, g_1)$  and  $E(x_4, g_2)$  penetrate both  $S_1$  and  $S_2$ . In other words, the top node  $x_4$  lacks visibility to the bottom nodes of the sub-mesh  $g_1$  and  $g_2$ , and results in a self-intersecting subdivision for the volume element. This self-intersecting volume element can be fixed by curving the edges  $E(x_4, g_1)$  and  $E(x_4, g_2)$ , and potentially  $E(x_4, x_2)$  and  $E(x_4, x_3)$ .

#### 4.4 Extension to high-order

For high-order approximations, more nodes are required to be added to the edges, faces and interior of NEFEM volume elements. It is worth emphasising first that, due to the non-isoparametric nature of NEFEM elements, the location of the high-order nodes are not linked to any geometric mapping and will only be used to define high-order polynomial basis within the element. Therefore, the high-order nodes can be added to the NEFEM elements without any modification to the NEFEM mesh. The only requirement is that the high-order nodes must be located on the edges or faces or inside of the NEFEM element to achieve optimal convergence [25]. It is also appreciated the high-order nodes are not created during the volume



mesh generation, but are appended to the NEFEM mesh after the meshing process.

The high-order nodes on the boundary surfaces are initially created during the surface meshing, as it is detailed in [37]. For volume elements, a similar strategy is used to add high-order nodes to the edges and faces. However, the interior nodes for elements with interpolation polynomial order  $p > 3$  must satisfy the requirement that they cannot be located outside the volume, thus a different placement strategy is used. This strategy is based on an elasticity analogy, in which the interior nodes are initially mapped from the distribution in a standard reference tetrahedral element, and then the boundary face are deformed to adapt the NURBS B-rep. During the deformation, the vertex nodes are fixed and Dirichlet boundary conditions are applied to the face nodes on the B-rep surfaces, so the interior nodes are moved accordingly by solving a linear elasticity problem. The details of this strategy have been presented in [43], and similar works such as [44, 45]. To improve the efficiency, in this work, the elastic deformations are limited to a number of user-defined layers of volume elements surrounding the boundary. In Fig. 14, an example is presented to demonstrate the elastic deformation of the NEFEM volume elements. It can be seen in Fig. 14b a few traversing NEFEM elements have deformed when the elevation of polynomial order

to  $p = 6$ , and the interior volume element above has also deformed accordingly.

### 5 Examples

This section presents some examples to demonstrate the NEFEM volume mesh generation strategy detailed in the previous section. The examples are selected to cover a wide range of geometric features, including sharp dihedral angles, sliver surfaces, steps, etc. In particular, validation of the generated NEFEM meshes for the first example, as analytical surface area and volume are available, is also presented.

#### 5.1 Flat plate with two cylinders

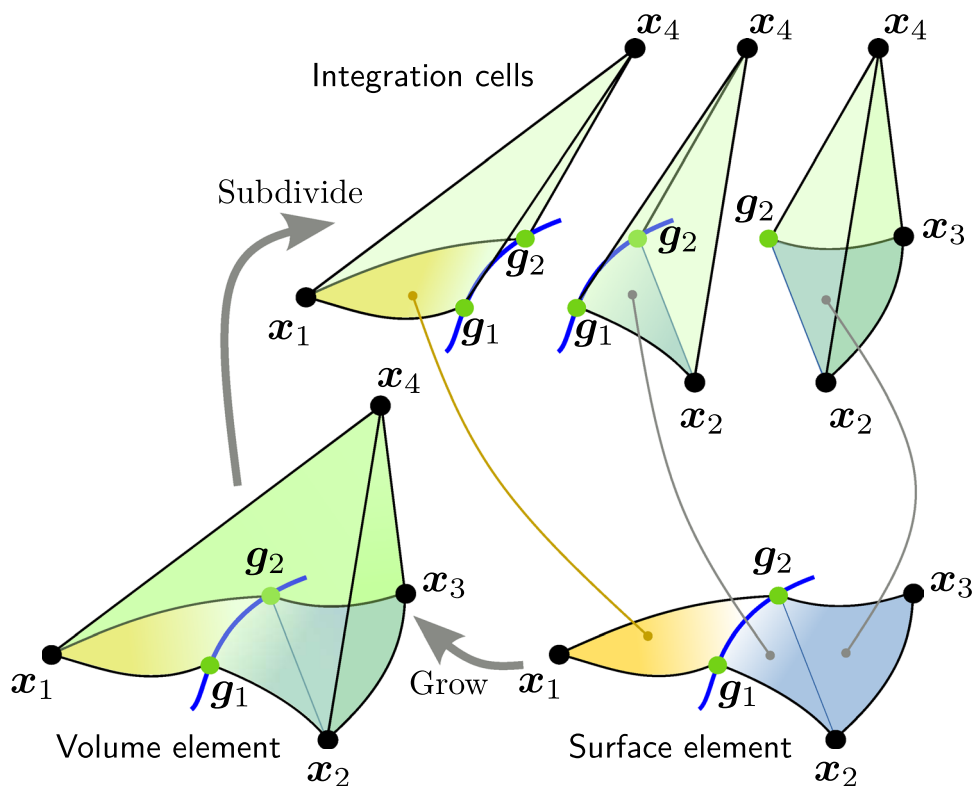
In the initial illustration, we examine a flat plate featuring two cylinders, one flat with a substantial radius and the other slender with a small radius, as depicted in Fig. 15.

Detailed geometric specifications are outlined in Table 1.

Figure 16 showcases the original FEM mesh, which deviates from the user-defined spacing.

Subsequently, employing the methodology proposed in [37], the NEFEM surface mesh is generated, maintaining a

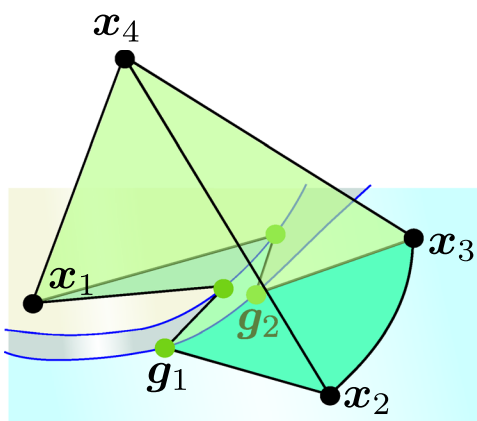
Fig. 12 Integration cells of a typical NEFEM volume element grown from the sub-mesh of the surface element in Fig. 6



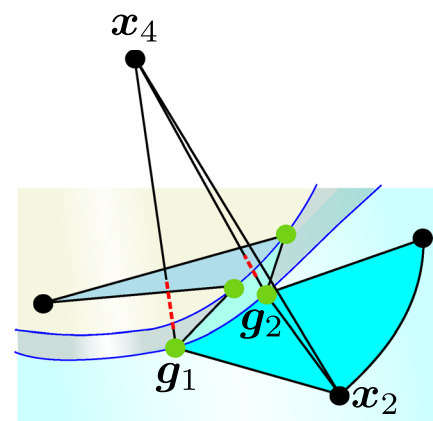
consistent mesh size irrespective of plate thickness, cylinder heights, or diameters, as demonstrated in Fig. 17a.

Utilising this boundary discretisation and the specified element size, the first layer of NEFEM volume elements is crafted following the approach delineated in Sect. 4, as shown in Fig. 17b. Notably, volume elements spanning surfaces are highlighted in green. The interior volume elements are crafted based on the extracted exterior faces of the NEFEM volume layer and the surface mesh, creating a standard triangulation through the Delaunay method for tetrahedral mesh. This model yields 121 tetrahedra in the NEFEM volume layer and 50 815 interior volume elements. Figure 17c, visually underscores the significantly reduced number of NEFEM volume elements compared to standard elements, minimising the computational impact on the solver. This enhancement is particularly crucial for transient simulations utilising explicit time marching. Table 2 illustrates, the minimum element edge length normalised with the user-specified spacing for the NEFEM mesh has increased by over tenfold.

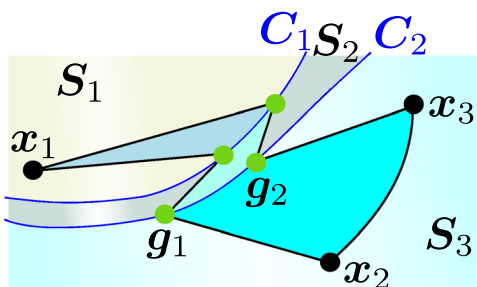
In the context of this specific example, where analytical determination of the exact surface area and volume is feasible, the NEFEM elements, embedded with exact NURBS geometry information, allow for validation. This validation involves computing the relative error of surface area and volume, utilising the evaluated exact area and volume as reference. Employing Gauss-Legendre quadrature rules with varying numbers of quadrature points, Fig. 18 portrays a diminishing trend in relative errors to machine precision with increasing quadrature points ( $n_{\text{GQ}}$ ) in each dimension. This dual-validation approach not only confirms the integrity of the mesh but also underscores the advantages introduced by NEFEM in mitigating geometric representation errors. Notably, the quadratures are executed on each subcell for NEFEM elements, ensuring results are independent of the interpolation polynomial order.



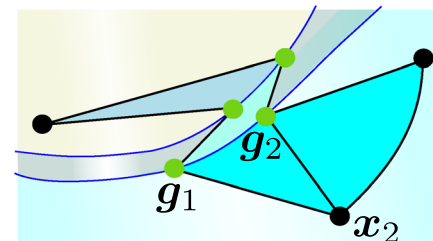
(a) Volume element with a step feature



(b) A volume subcell with self-intersections



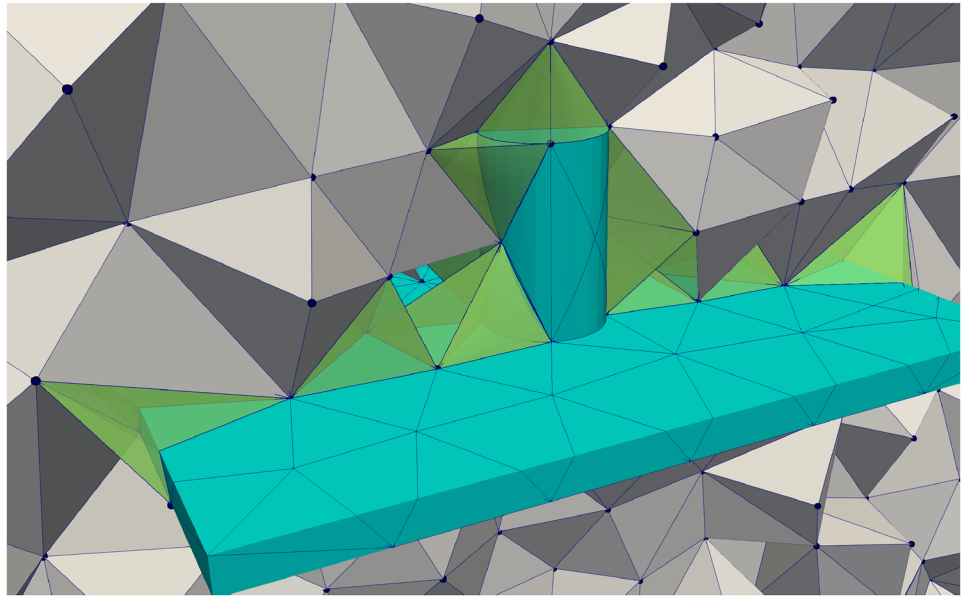
(c) Underlying surface element of (a)



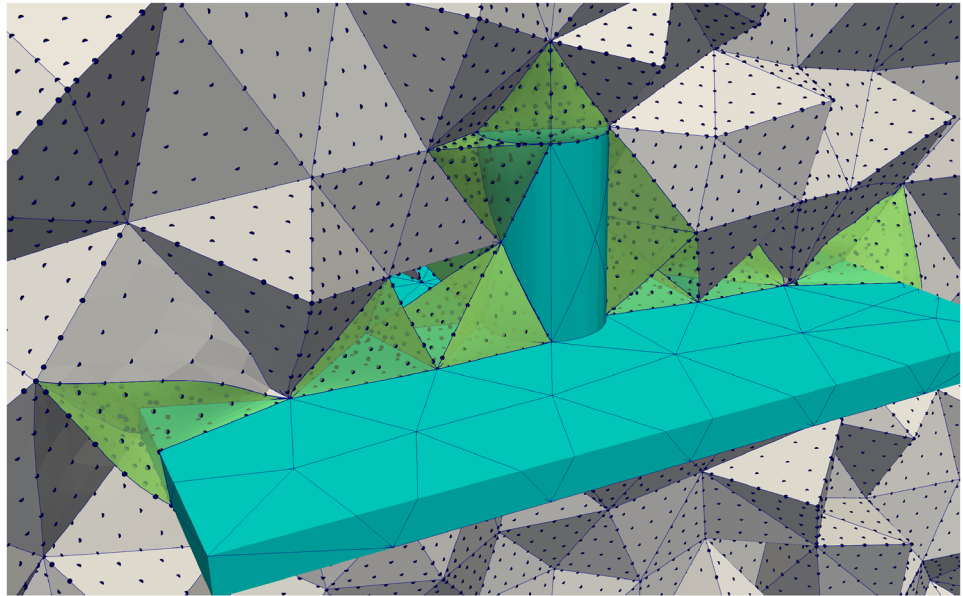
(d) Underlying surface subcell of (b)

**Fig. 13** Illustration of visibility issue for a NEFEM volume element and its base surface element at a step feature

**Fig. 14** Deforming the NEFEM elements (rendered in green) to ensure the validity after adding high-order nodes



(a) NEFEM volume mesh (clipped)



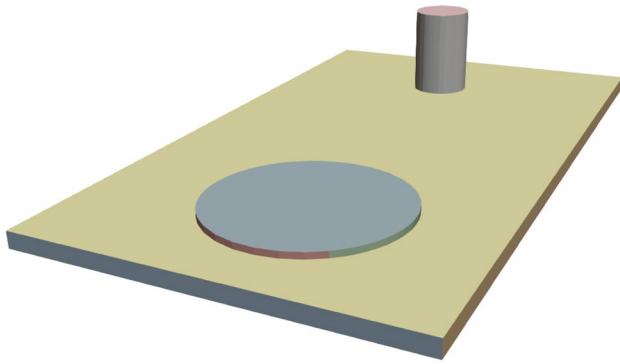
(b) Deformed volume mesh after adding high-order nodes ( $p = 6$ )

## 5.2 Hollow fairing for a turbine engine

The second illustrative case, as introduced in Section 2, focuses on generating the NEFEM volume mesh for a turbine engine fairing utilising a uniform spacing function. The CAD model, shown in Fig. 1a, incorporates four surfaces representing the outer and inner shells, along with six slender surfaces denoting the leading and trailing edges

that interconnect these two shells. The model's characteristic dimensions are detailed in Table 3, underscoring substantial variations in the lengths of curves within the CAD model. Notably, the longest curve is approximately 54 times larger than the shortest one, introducing a significant range in dimensions.

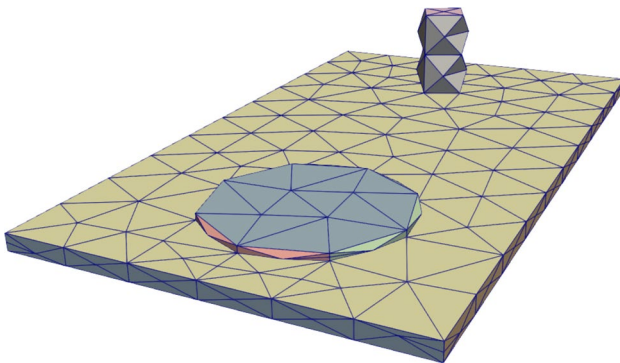
Crucially, the user-specified spacing in this scenario, denoted as  $h(\mathbf{x}) = 100$ , necessitates elements with a



**Fig. 15** NURBS surfaces in the CAD model of a flat plate with two cylinders

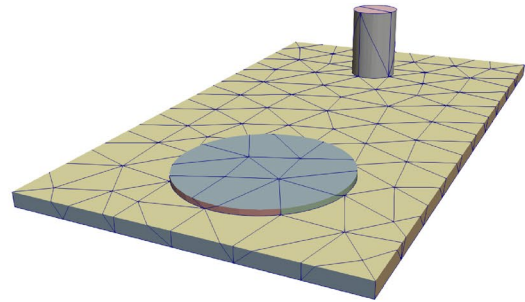
**Table 1** Geometric data of the flat plate with cylinders model

|                          |       |
|--------------------------|-------|
| Number of NURBS Surfaces | 12    |
| Number of NURBS Curves   | 24    |
| Minimum curve length     | 0.019 |
| Maximum curve length     | 2.000 |

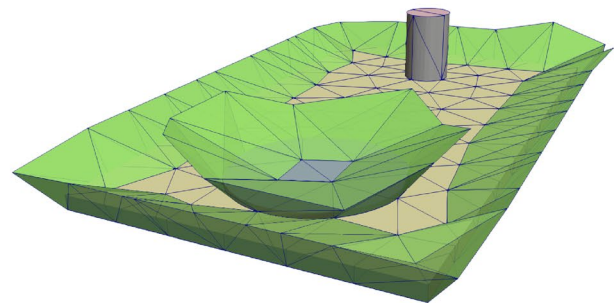


**Fig. 16** FEM surface mesh of the flat plate intersected by two cylinders, showing elements not complying with the user-defined spacing

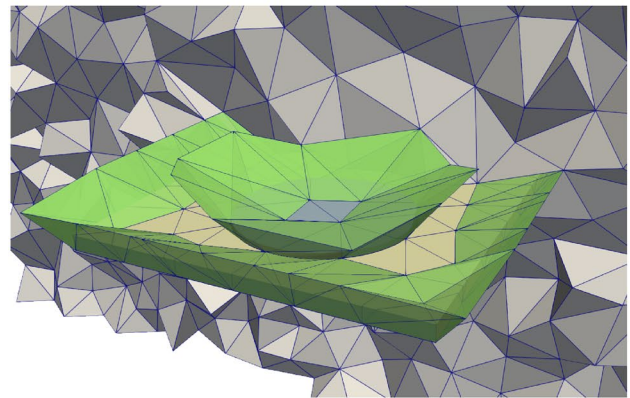
representative length over six times larger than the shortest curve in the CAD model. Consequently, the initial FEM mesh exhibits several undersized elements that deviate markedly from the desired spacing, as portrayed in Fig. 1b. The NEFEM mesh effectively addresses this issue by eliminating small elements on both leading and trailing edges. It achieves an impressive 9.64-fold increase in the minimum edge length, as outlined in Table 4. This enhancement translates to a comparable acceleration factor for explicit time integration, achievable by utilising a considerably larger time step.



(a) NEFEM surface mesh



(b) Traversing volume elements rendered in green



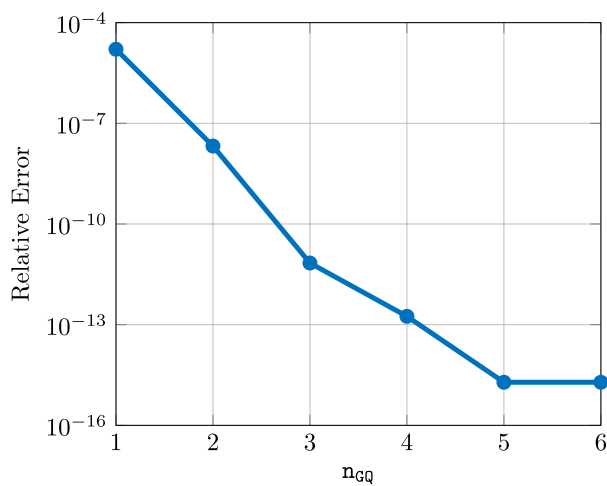
(c) Clipped NEFEM volume mesh with interior volume elements

**Fig. 17** NEFEM meshing process for the flat plate intersected by two cylinders

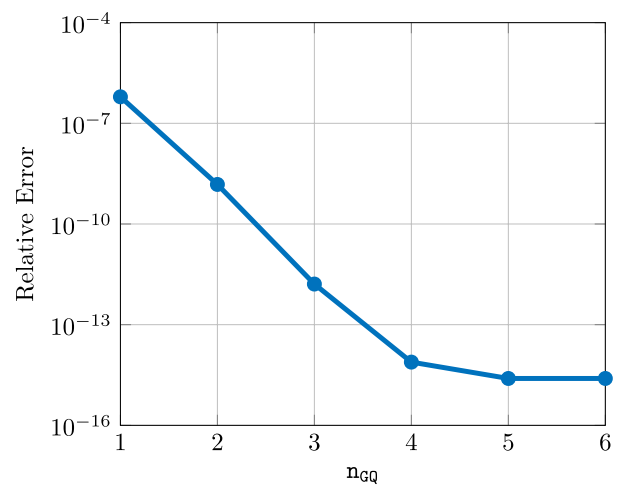
**Table 2** Normalised edge lengths for the flat plate intersected by two cylinders

|                                   |       |
|-----------------------------------|-------|
| Minimum edge length in FEM mesh   | 0.045 |
| Minimum edge length in NEFEM mesh | 0.458 |
| Increase factor                   | 10.18 |





(a) Convergence for area of NEFEM surface mesh.



(b) Convergence for volume of NEFEM volume mesh.

**Fig. 18** Validation of the meshes for the model of a flat plate with two cylinders. The relative errors decrease to machine precision with increasing quadrature points

The traversing volume elements are visually presented in Fig. 19, highlighted in green with subtle transparency to facilitate observation of the curved boundary. For clarity, a specific NEFEM tetrahedron is isolated and presented with zoom-in views in Fig. 19c and 19d. From Fig. 19c, a distinct sharp ridge within a face is evident, originating from the sharp dihedral between the two surfaces the element traverses. It is crucial to emphasise that this ridge is embedded within the element and does not constitute an edge of the element, and no nodes are associated with it.

### 5.3 Wing with a blunt trailing edge

This illustration showcases the creation of NEFEM meshes for a wing featuring a blunt trailing edge, emphasising the capability to accommodate user-specified non-uniform mesh spacing. The NURBS surfaces defining the wing are visually represented in Fig. 20 and comprehensive geometric data is concisely summarised in Table 5. To encompass the entire wing within a CFD-oriented meshing strategy, a substantial bounding box is appended to the CAD model, ensuring ample coverage.

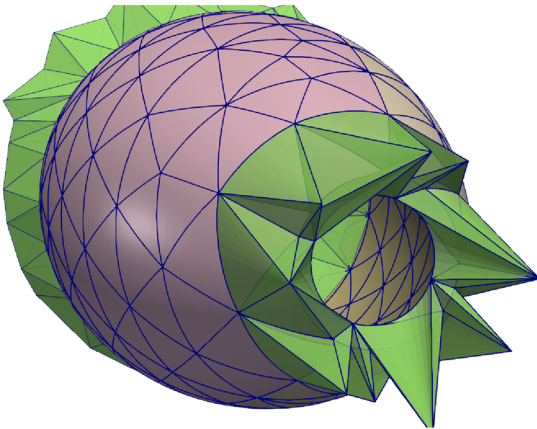
Notably, a non-uniform mesh spacing is implemented using two line sources at both the leading and trailing edges, employing a stretching ratio of 5. Despite the refinement introduced by these line sources, the prescribed mesh size remains larger than the length of the shortest curve. The ensuing FEM mesh, shown in Fig. 21, reveals an abundance

of small elements at the blunt trailing edge and wing tip due to the constraints of the thin thickness.

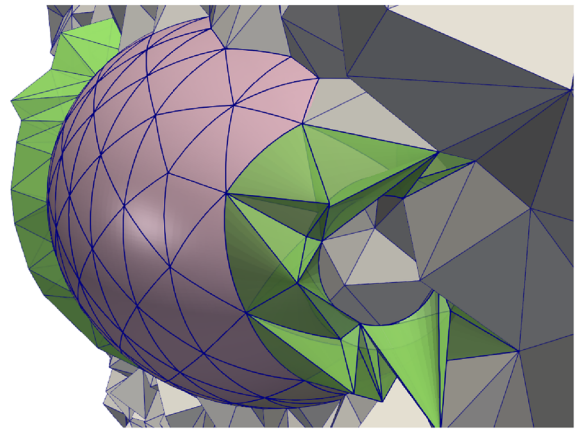
Subsequently, Fig. 22a illustrates how the NEFEM surface mesh effectively eliminates these small elements. Based on this surface mesh, Fig. 22b shows the creation of traversing NEFEM volume elements at the blunt trailing edge, complemented by the interior volume mesh presented in 22c. Evidently, the number of traversing elements, totalling 179, is notably fewer compared to the staggering 1 458 208 interior volume elements. This signifies that minimal additional computational costs will be incurred during integration over the NEFEM elements. Table 6 provides further insight, indicating that the normalised minimum edge length in the NEFEM mesh exceeds 7 times that of the FEM mesh. This reaffirms that the NEFEM element constitutes only a negligible portion of the overall mesh but significantly enhances the minimum element size, facilitating large time-stepping in explicit solvers.

### 5.4 Falcon aircraft

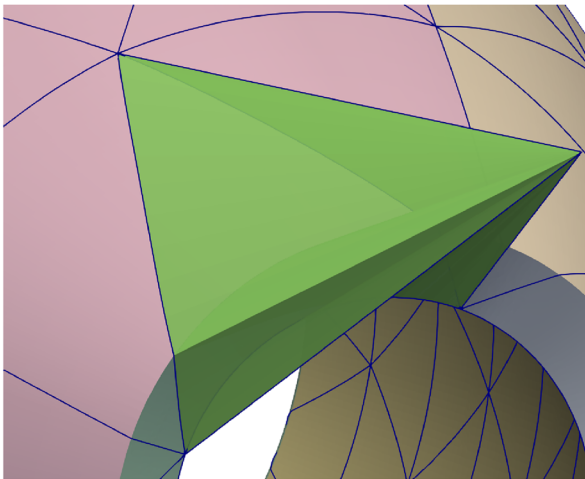
In this example, we consider a comprehensive aircraft model to underscore the adept handling of intricate geometries. The CAD geometry exhibits an array of complex features, including notably short curves and petite surfaces, particularly evident at the wing tips. The wing's characteristic thickness, approximately 0.2, poses a specific challenge during surface mesh generation, given its brevity compared to the minimum curve length.



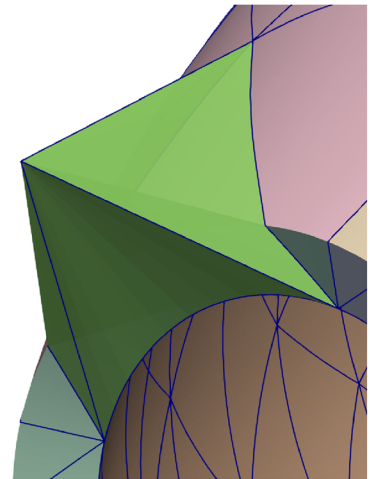
(a) Traversing volume elements rendered in green



(b) Clipped NEFEM volume mesh with interior volume elements



(c) A selected NEFEM tetrahedron



(d) The tetrahedron in (c), viewing from right

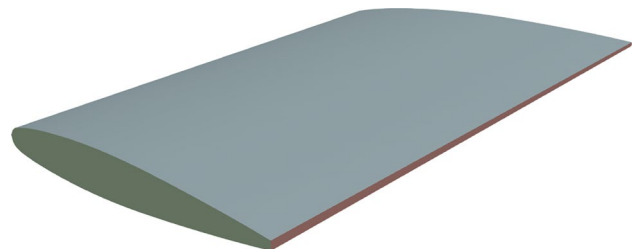
**Fig. 19** NEFEM meshing process for the hollow fairing of a turbine engine

**Table 3** Geometric data of the hollow fairing model

|                          |        |
|--------------------------|--------|
| Number of NURBS Surfaces | 10     |
| Number of NURBS Curves   | 22     |
| Minimum curve length     | 15.62  |
| Maximum curve length     | 850.74 |

**Table 4** Normalised edge lengths for the hollow fairing model

|                                   |       |
|-----------------------------------|-------|
| Minimum edge length in FEM mesh   | 0.076 |
| Minimum edge length in NEFEM mesh | 0.734 |
| Increase factor                   | 9.64  |

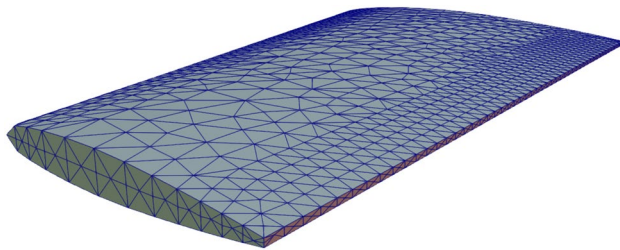


**Fig. 20** NURBS surfaces in the CAD model of a wing with blunt trailing edge

Figure 23 elucidates the NURBS surfaces of the CAD model, while Table 7 succinctly summarises the geometric data, revealing a substantial ratio between maximum

**Table 5** Geometric data of the wing model

|                          |          |
|--------------------------|----------|
| Number of NURBS Surfaces | 5        |
| Number of NURBS Curves   | 9        |
| Minimum curve length     | 7.27     |
| Maximum curve length     | 1 381.12 |



**Fig. 21** FEM surface mesh of the wing, showing elements not complying with the user-defined spacing

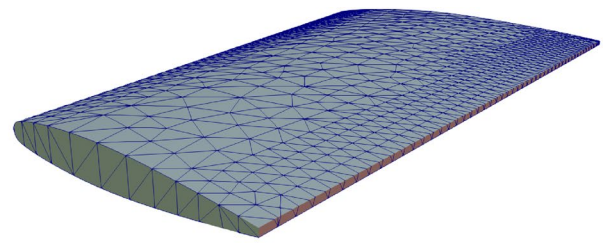
and minimum curve lengths, approximately 28.68. Consequently, the FEM mesh, depicted in Fig. 24, anticipates the emergence of small elements violating the user-defined spacing. Despite all surface elements being NURBS-enhanced, originating from the discretisation of the intricate B-rep, only a sparse number of traversing elements are necessary to enhance the minimal element size.

The first layer of NEFEM volume elements, strategically concentrating on the building of traversing volume elements, comprises a modest number of tetrahedral elements, as showcased in Fig. 25b, c. The total count of tetrahedral elements in the volume mesh stands at 229 693, with 28 NEFEM tetrahedra involving multiple definitions during integration. Table 8 presents a noteworthy enhancement, with the normalised minimum edge length in the NEFEM mesh surpassing that of the FEM mesh by 4.5 times.

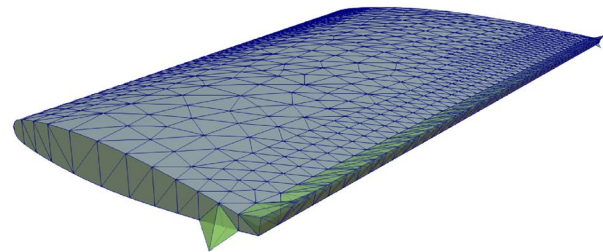
It is imperative to highlight the significance of this improvement, as even a small subset of elements exhibiting spacing well below the user-defined threshold can render the solution of a transient problem with explicit time marching unfeasible. Consequently, the capability to overcome this restriction bears substantial implications for the solver.

### 5.5 Mechanical component

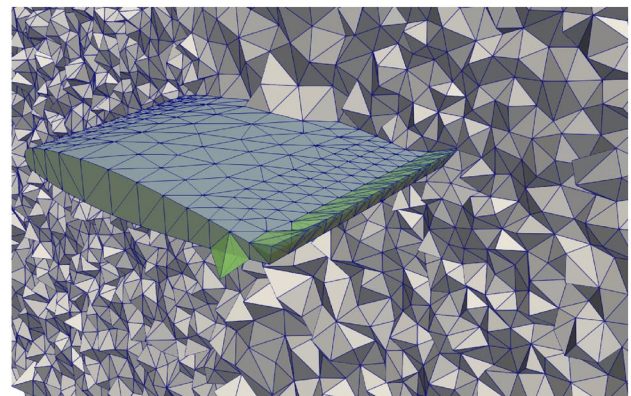
This specific case is designed to showcase the adeptness of the proposed methodology in generating a volume mesh within a singular enclosed domain. The focal point is a knurled knob, depicted in Fig. 26, where surfaces are delineated in distinct colours and delicately transparent to afford



(a) NEFEM surface mesh



(b) Traversing volume elements rendered in green



(c) Clipped NEFEM volume mesh with interior elements

**Fig. 22** NEFEM mesh process for the wing with a blunt trailing edge

**Table 6** Normalised edge lengths for the wing

|                                   |      |
|-----------------------------------|------|
| Minimum edge length in FEM mesh   | 0.06 |
| Minimum edge length in NEFEM mesh | 0.46 |
| Increase factor                   | 7.67 |

visibility into the internal structure. The model encompasses several intricate features, including small curved surfaces, diminutive steps or fillets, and curved holes. Notably, the length of the longest curve exceeds that of the shortest one



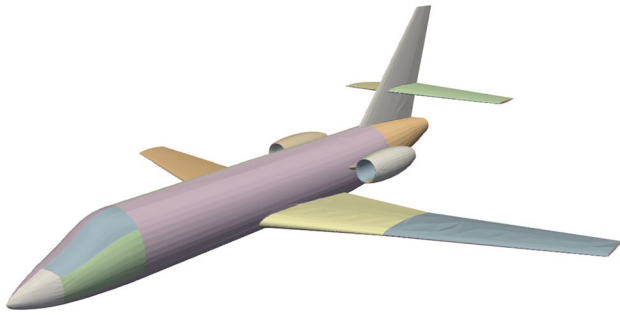
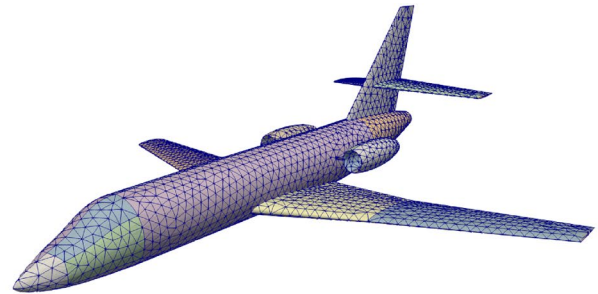


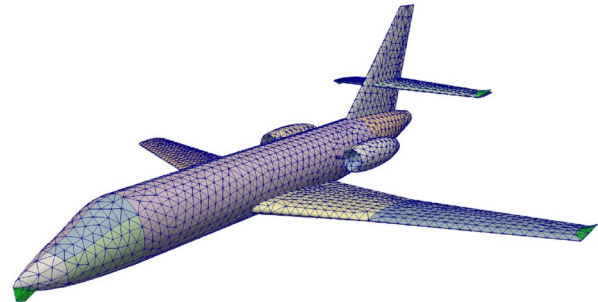
Fig. 23 NURBS surfaces in the CAD model of the Falcon aircraft

Table 7 Geometric data of the Falcon model

|                          |       |
|--------------------------|-------|
| Number of NURBS Surfaces | 48    |
| Number of NURBS Curves   | 100   |
| Minimum curve length     | 0.37  |
| Maximum curve length     | 10.61 |



(a) NEFEM surface mesh



(b) Traversing volume elements rendered in green

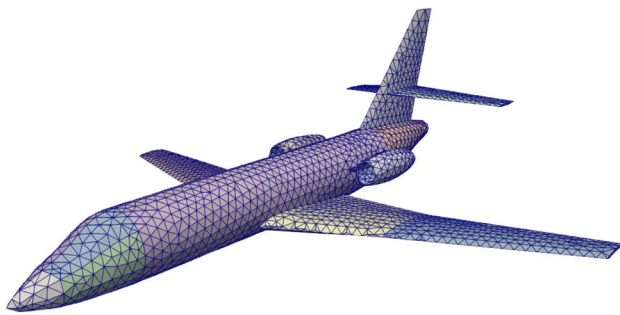
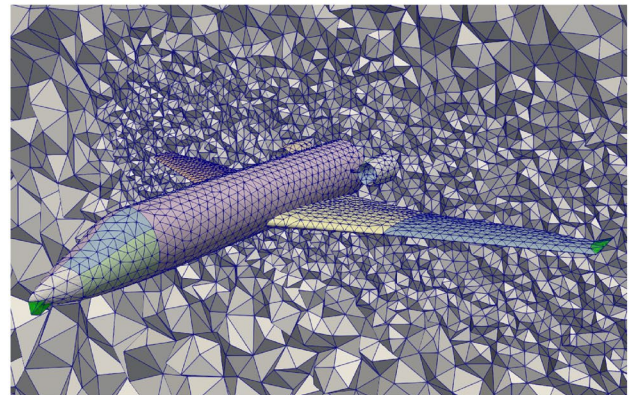


Fig. 24 FEM surface mesh of the Falcon, showing elements not complying with the user-defined spacing



(c) Clipped NEFEM volume mesh with interior volume elements

by a factor of approximately 40.84, as succinctly detailed in Table 9.

Figure 27, the initial FEM surface mesh is presented, revealing numerous small elements constrained by the intricate features.

In stark contrast, the NEFEM meshes, shown in Fig. 28, effectively eliminate these diminutive elements, resulting in a notable 5.82-fold increase in the minimum edge length, meticulously documented in Table 10. Traversing volume elements, depicted in green, are discernible in Fig. 28b offering a clear view through the slight transparency applied to the NEFEM surface mesh. The clipped NEFEM volume mesh, inclusive of interior volume elements, is illustrated in Fig. 28c. For a more detailed examination, an alternative

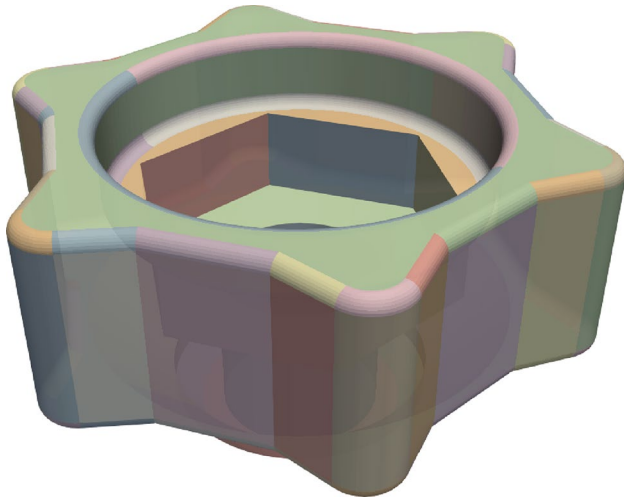
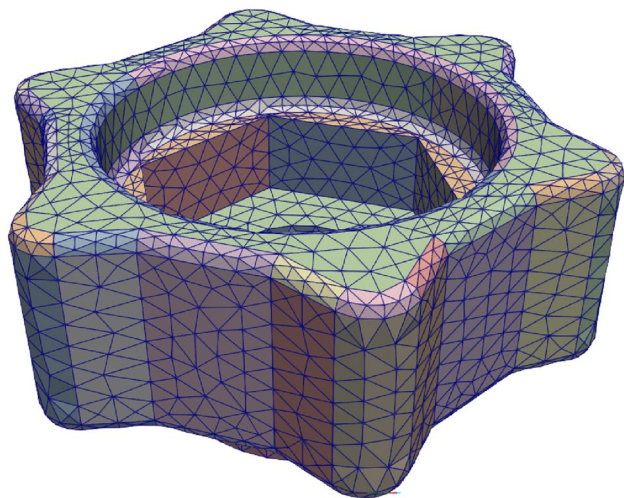
Fig. 25 NEFEM mesh process for the Falcon aircraft

Table 8 Normalised edge lengths for the Falcon model

|                                   |      |
|-----------------------------------|------|
| Minimum edge length in FEM mesh   | 0.06 |
| Minimum edge length in NEFEM mesh | 0.27 |
| Increase factor                   | 4.50 |

**Table 9** Geometric data of the mechanical component model

|                          |       |
|--------------------------|-------|
| Number of NURBS Surfaces | 103   |
| Number of NURBS Curves   | 240   |
| Minimum curve length     | 1.23  |
| Maximum curve length     | 50.23 |

**Fig. 26** NURBS surfaces in the CAD model of the knurled knob**Fig. 27** FEM surface mesh of the knurled knob, showing elements not complying with the user-defined spacing

perspective of Fig. 28b is selected and presented with zoom-in views Fig. 29.

Once again, this example underscores the consistent traits of the NEFEM volume mesh, including geometric preservation and the limited number of traversing elements.

## 5.6 Electromagnetic simulation using NEFEM meshes

In this example, a numerical example is presented to show the validity of NEFEM volume meshes for performing simulations. A NEFEM volume mesh around a plate with a short thick cylinder on the top surface is generated using the strategy proposed in this paper. The mesh is shown in Fig. 30.

A simulation of the scattering of a plane electromagnetic wave is considered using a high-order discontinuous Galerkin method [46] equipped with the NEFEM formulation. The simulation is performed in the time domain and the quantity of interest is the so-called radar cross section (RCS) [47].

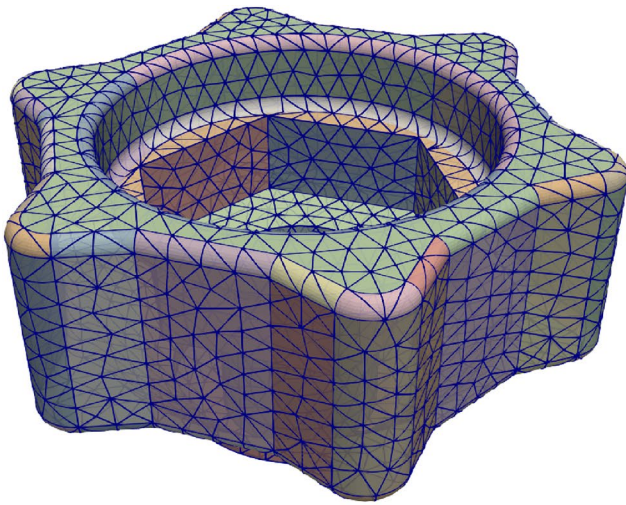
Figure 31 shows the computed RCS results for vertical and horizontal polarisations and for an increasing degree of the approximation used for the solution.

The results show the validity of the NEFEM meshes and it shows that the results with quadratic and cubic approximation of the solution are almost identical. In fact, even when using a linear approximation of the solution, it can be observed that the exact boundary representation enables to compute an accurate RCS.

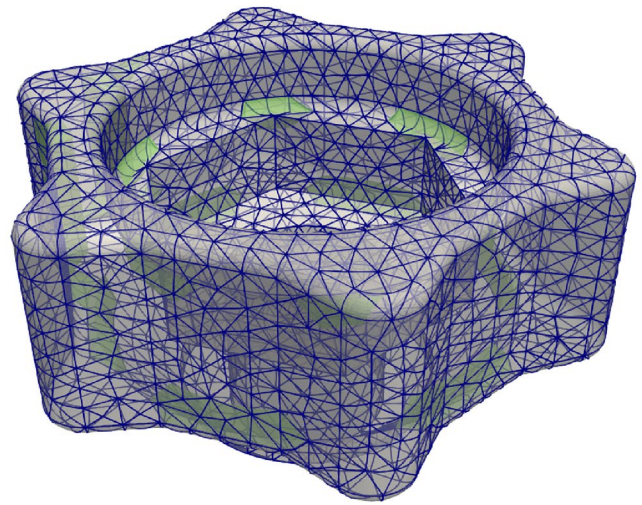
## 6 Concluding remarks

In this work, a mesh generation framework dedicated to creating volume meshes tailored for NEFEM has been presented for the first time. The technique is capable of generating volume meshes which preserves the exact boundary representation provided by the NURBS parametrisation from the CAD model. In the NEFEM mesh, the geometric definition at the featured boundary is encapsulated in NEFEM elements that may traverse multiple curves or surfaces. As a result, the small geometric features present in the CAD model no longer restrict the element size in a NEFEM mesh. This has completely removed the need for the time-consuming de-featuring process on complex CAD models and, at the same time, has eliminated the geometric representation error introduced by the de-featuring process or by traditional mesh generators.

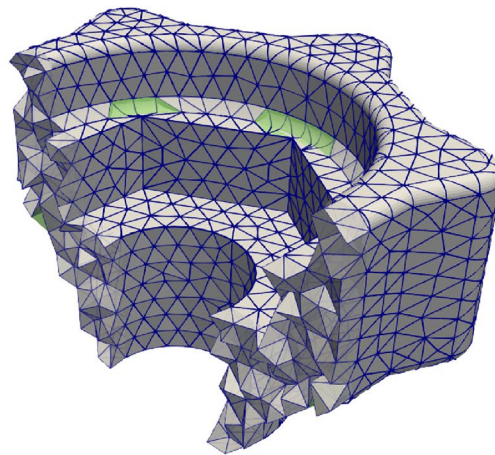




(a) NEFEM surface mesh

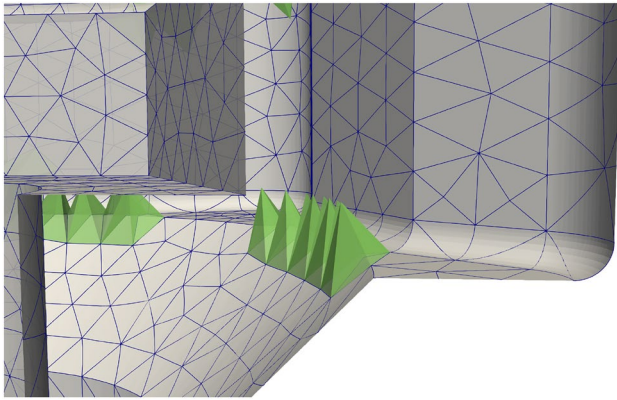


(b) Traversing volume elements rendered in green



(c) Clipped NEFEM volume mesh with interior volume elements

**Fig. 28** NEFEM mesh process for the knurled knob

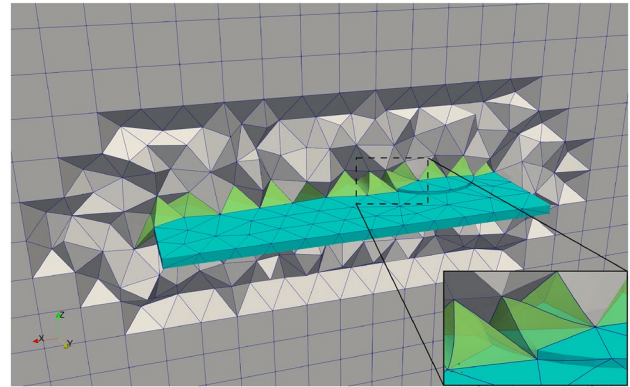


**Fig. 29** Zoomed view of the clipped NEFEM surface mesh with traversing volume elements

**Table 10** Normalised edge lengths for the mechanical component model

|                                   |      |
|-----------------------------------|------|
| Minimum edge length in FEM mesh   | 0.17 |
| Minimum edge length in NEFEM mesh | 0.99 |
| Increase factor                   | 5.82 |

Given a CAD geometry in the form of B-rep, the proposed strategy begins with generating an initial FEM surface mesh using a standard mesh generator. Then, during the NEFEM surface mesh generation guided by the user-defined spacing, elements near the undersized geometric features are remeshed, and the new elements are allowed to traverse multiple surfaces, provided they pass the dedicated validity check. This process results in a NEFEM surface mesh customised for the next volume mesh generation stage. Various normal vectors are defined and computed on the NEFEM surface elements to guide the growth of volume elements. During the growth of each volume element, self-intersection checks are performed to ensure the element validity. The volume elements grown from traversing surface elements form the first layer of NEFEM volume elements, including a few traversing elements, whose exterior faces are extracted and merged with the NEFEM surface mesh, so that a standard volume mesh generator can be used to obtain the remaining of the interior volume mesh. The final NEFEM volume mesh



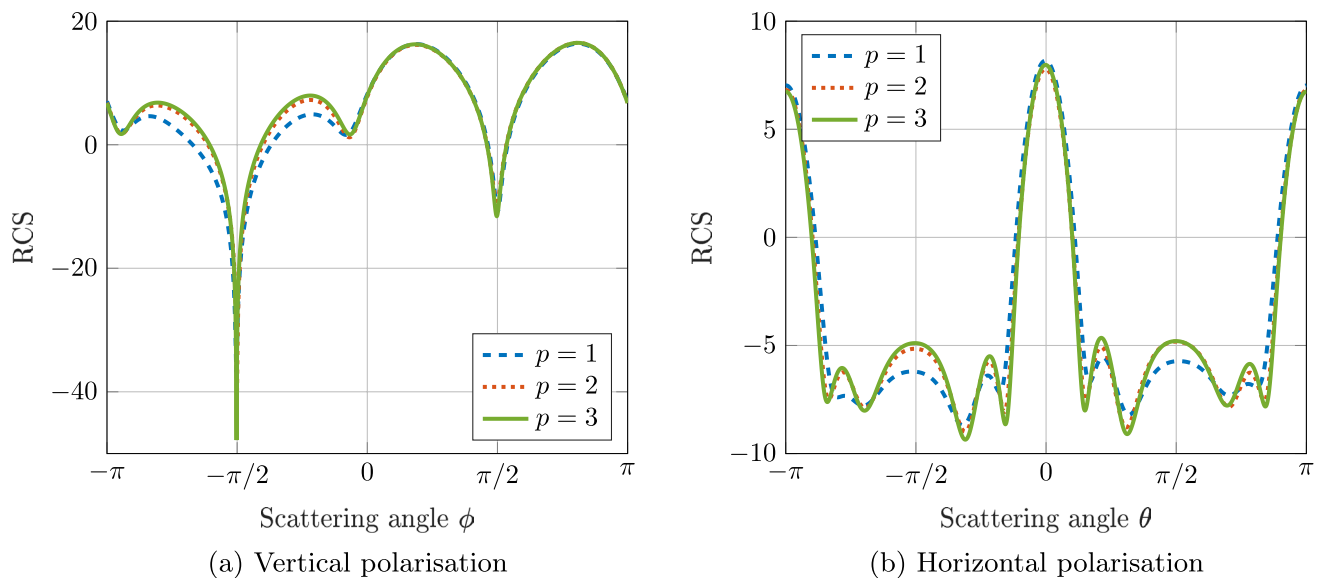
**Fig. 30** NEFEM volume mesh for the electromagnetic simulation

is obtained by combining the NEFEM volume layer and the interior volume mesh.

Examples have been presented to demonstrate the applicability and potential of the proposed meshing framework. For completeness, the CAD model, the initial FEM surface mesh, the NEFEM surface and volume meshes are shown. The examples involve geometries where the CAD model contains very small edges, such as the blunt trailing edges of the hollow fairing and the wing. The resulting NEFEM meshes demonstrate a spacing closely matching the user-specification, even when the CAD model contains small features. The extension to high-order interpolations can be easily achieved by adding high-order nodes to the NEFEM elements and deforming the elements via an elasticity analogy, which is independent on the definition of geometric mapping for the NURBS-enhanced elements.

In the examples shown, all elements with at least one face or edge on the boundary are treated as NEFEM elements. However, for problems with quantities of interest in specific parts of the domain, it is possible for the user to specify the surfaces of the CAD model that will be discretised with NEFEM elements and keep a traditional isoparametric description near boundaries away from the region of interest.

Future work will involve devising a new definition of the element quality and its improvement strategies. Ultimately, it



**Fig. 31** RCS for an increasing order of approximation of the solution

is expected to integrate the generated mesh to existing FEM solvers with an API for further applications.

**Acknowledgements** The authors gratefully acknowledge the financial support of the EPSRC project *A feature-independent mesh generation and integrated solution framework* with reference EP/T009071/1.

**Data availability** The data that support the findings of this study are available from the authors upon reasonable request.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Piegl L, Tiller W (1995) *The NURBS Book*. Springer, London. <https://doi.org/10.1007/978-3-642-97385-7>
- Gammon M, Bucklow H, Fairey R (2018) A review of common geometry issues affecting mesh generation. *AIAA Aerospace Sci Meeting*. <https://doi.org/10.2514/6.2018-1402>
- Park MA, Kleb WL, Jones WT, Krakos JA, Michal TR, Loseille A, Haimes R, Dannenhoffer J (2019) Geometry modeling for unstructured mesh adaptation. In: *AIAA Aviation 2019 Forum*, p. 2946. <https://doi.org/10.2514/6.2019-2946>
- Thakur A, Banerjee AG, Gupta SK (2009) A survey of cad model simplification techniques for physics-based simulation applications. *Comput-Aided Design* 41(2):65–80. <https://doi.org/10.1016/j.cad.2008.11.009>
- Shapiro V, Tsukanov I, Grishin A (2011) Geometric issues in computer aided design/computer aided engineering integration. *J Comput Inform Sci Eng* 11(2) <https://doi.org/10.1115/1.3593416>
- Katz A, Sankaran V (2011) Mesh quality effects on the accuracy of CFD solutions on unstructured meshes. *J Comput Phys* 230(20):7670–7686. <https://doi.org/10.1016/j.jcp.2011.06.023>
- Hesthaven JS, Warburton T (2002) Nodal high-order methods on unstructured grids I. time-domain solution of Maxwell's equations. *Journal of Computational Physics* 181(1), 186–221 <https://doi.org/10.1006/jcph.2002.7118>
- Solin P, Segeth K, Dolezel I (2003) *Higher-Order Finite Element Methods*. Chapman and Hall/CRC, New York. <https://doi.org/10.1201/9780203488041>
- Wang ZJ, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, Deconinck H, Hartmann R, Hillewaert K, Huynh HT (2013) High-order CFD methods: current status and perspective. *Int J Numer Methods Fluids* 72(8):811–845. <https://doi.org/10.1002/fld.3767>
- Huynh H, Wang ZJ, Vincent P (2014) High-order methods for computational fluid dynamics: a brief review of compact differential formulations on unstructured grids. *Computers & Fluids* 98, 209–220 <https://doi.org/10.1016/j.compfluid.2013.12.007>
- Sørensen KA, Hassan O, Morgan K, Weatherill NP (2003) A multigrid accelerated hybrid unstructured mesh method for 3D compressible turbulent flow. *Comput Mech* 31(1–2):101–114. <https://doi.org/10.1007/s00466-002-0397-9>
- Geuzaine C, Remacle J-F (2009) Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 79(11):1309–1331 <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2579>
- Si H (2015) Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.* 41(2) <https://doi.org/10.1145/2629697>
- ITI Incorporated: (2023) CADfix. International TechneGroup Incorporated, International TechneGroup Incorporated. <https://www.iti-global.com/cadfix>



15. Altair Engineering: (2023) Altair HyperMesh. Altair Engineering, Inc., Altair Engineering, Inc. <https://altair.com/hypermesh>
16. Mobley AV, Carroll MP, Canann SA (1998) An object oriented approach to geometry defeaturing for finite element meshing. In: 7th International Meshing Roundtable (IMR), pp. 547–563. <https://www.osti.gov/biblio/4089>
17. Danglade F, Pernot J-P, Véron P (2014) On the use of machine learning to defeature CAD models for simulation. *Comput-Aided Design Appl* 11(3):358–368. <https://doi.org/10.1080/16864360.2013.863510>
18. Owen S, Shead TM, Martin S (2020) CAD Defeaturing Using Machine Learning. In: 28th International Meshing Roundtable (IMR). <https://doi.org/10.5281/zenodo.3653426>
19. Owen SJ, Carbajal AJ, Peterson MG, Ernst CD (2024) Machine learning classification and reduction of cad parts. In: Ruiz-Gironés, E., Sevilla, R., Moxey, D. (eds.) *SIAM International Meshing Roundtable 2023*, pp. 93–114. Springer, Cham. [https://doi.org/10.1007/978-3-031-40594-5\\_5](https://doi.org/10.1007/978-3-031-40594-5_5)
20. Ergatoudis J, Irons BM, Zienkiewicz OC (1968) Curved isoparametric “quadrilateral” elements for finite element analysis. *Internat J Solids Struct* 4(1):31–42. [https://doi.org/10.1016/0020-7683\(68\)90031-0](https://doi.org/10.1016/0020-7683(68)90031-0)
21. Xue D, Demkowicz L (2005) Control of geometry induced error in *hp* finite element (FE) simulations. I. Evaluation of FE error for curvilinear geometries. *International Journal of Numerical Analysis and Modeling* 2(3), 283–300 [http://global-sci.org/intro/article\\_detail/ijnam/932.html](http://global-sci.org/intro/article_detail/ijnam/932.html)
22. Sevilla R, Huerta A (2018) HDG-NEFEM with degree adaptivity for Stokes flows. *J Sci Comput* 77(3):1953–1980. <https://doi.org/10.1007/s10915-018-0657-2>
23. Sevilla R (2019) HDG-NEFEM for two dimensional linear elasticity. *Computers & Structures* 220, 69–80 <https://doi.org/10.1016/j.compstruc.2019.05.005>
24. Giacomini M, Sevilla R (2019) Discontinuous Galerkin approximations in computational mechanics: hybridization, exact geometry and degree adaptivity. *SN Appl Sci* 1(9):1–15. <https://doi.org/10.1007/s42452-019-1065-4>
25. Sevilla R, Fernández-Méndez S, Huerta A (2008) NURBS-enhanced finite element method (NEFEM). *Int J Numer Methods Eng* 76(1):56–83. <https://doi.org/10.1002/nme.2311>
26. Sevilla R, Fernández-Méndez S, Huerta A (2011) 3D-NURBS-enhanced finite element method (NEFEM). *Int J Numer Methods Eng* 88(2):103–125. <https://doi.org/10.1002/nme.3164>
27. Dawson M, Sevilla R, Morgan K (2018) The application of a high-order discontinuous galerkin time-domain method for the computation of electromagnetic resonant modes. *Applied Mathematical Modelling* 55, 94–108 <https://doi.org/10.1016/j.apm.2017.10.030>
28. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Methods Appl Mech Eng* 194(39–41):4135–4195. <https://doi.org/10.1016/j.cma.2004.10.008>
29. Taylor NJ, Haimes R (2018) Geometry modelling: Underlying concepts and requirements for computational simulation. In: 2018 Fluid Dynamics Conference, p. 3402. <https://doi.org/10.2514/6.2018-3402>
30. Sheffer A, Bercovier M, Blacker T, Clements J (2000) Virtual topology operators for meshing. *International Journal of Computational Geometry & Applications* 10(03):309–331. <https://doi.org/10.1142/S0218195900000188>
31. Legrain G (2013) A NURBS enhanced extended finite element approach for unfitted CAD analysis. *Comput Mech* 52(4):913–929. <https://doi.org/10.1007/s00466-013-0854-7>
32. Marco O, Sevilla R, Zhang Y, Ródenas JJ, Tur M (2015) Exact 3D boundary representation in finite element analysis based on Cartesian grids independent of the geometry. *Int J Numer Methods Eng* 103(6):445–468. <https://doi.org/10.1002/nme.4914>
33. Navarro-García H, Sevilla R, Nadal E, Ródenas JJ (2021) High-order discontinuous Galerkin method for time-domain electromagnetics on geometry-independent Cartesian meshes. *Int J Numer Methods Eng* 122(24):7632–7663. <https://doi.org/10.1002/nme.6846>
34. Sevilla R, Barbieri E (2014) NURBS distance fields for extremely curved cracks. *Comput Mech* 54(6):1431–1446. <https://doi.org/10.1007/s00466-014-1067-4>
35. Greco F, Coox L, Maurin F, Desmet W (2017) NURBS-enhanced maximum-entropy schemes. *Computer Methods in Applied Mechanics and Engineering* 317, 580–597 <https://doi.org/10.1016/j.cma.2016.12.024>
36. Sevilla R, Rees L, Hassan O (2016) The generation of triangular meshes for NURBS-enhanced FEM. *Int J Numer Methods Eng* 108(8):941–968. <https://doi.org/10.1002/nme.5247>
37. Zou X, Lo SB, Sevilla R, Hassan O, Morgan K (2023) The Generation of 3D Surface Meshes for NURBS-Enhanced FEM. *Computer-Aided Design*, 103653 <https://doi.org/10.1016/j.cad.2023.103653>
38. Sevilla R, Fernández-Méndez S, Huerta A (2011) NURBS-enhanced finite element method (NEFEM): a seamless bridge between CAD and FEM. *Arch Comput Methods Eng* 18(4):441–484. <https://doi.org/10.1007/s11831-011-9066-5>
39. Gordon WJ, Hall CA (1973) Transfinite element methods: blending-function interpolation over arbitrary curved element domains. *Numer Math* 21(2):109–129. <https://doi.org/10.1007/BF01436298>
40. Soghrafi S, Merel RA (2016) Nurbs enhanced hifem: A fully mesh-independent method with zero geometric discretization error. *Finite Elements in Analysis and Design* 120, 68–79 <https://doi.org/10.1016/j.finel.2016.06.007>
41. Owen SJ, Saigal S (2001) Formation of pyramid elements for hexahedra to tetrahedra transitions. *Comput Methods Appl Mech Eng* 190(34):4505–4518. [https://doi.org/10.1016/S0045-7825\(00\)00330-3](https://doi.org/10.1016/S0045-7825(00)00330-3)
42. Field DA (2001) Automatic generation of transitional meshes. *Int J Numer Methods Eng* 50(8):1861–1876. <https://doi.org/10.1002/nme.98>
43. Xie ZQ, Sevilla R, Hassan O, Morgan K (2013) The generation of arbitrary order curved meshes for 3D finite element analysis. *Comput Mech* 51(3):361–374. <https://doi.org/10.1007/s00466-012-0736-4>
44. Sherwin SJ, Peiró J (2002) Mesh generation in curvilinear domains using high-order elements. *Int J Numer Methods Eng* 53(1):207–223. <https://doi.org/10.1002/nme.397>
45. Gargallo-Peiró A, Roca X, Peraire J, Sarrate J (2014) Distortion and quality measures for validating and generating high-order tetrahedral meshes. *Engineering with Computers*, 1–15 <https://doi.org/10.1007/s00366-014-0370-1>
46. Sevilla R, Hassan O, Morgan K (2014) The use of hybrid meshes to improve the efficiency of a discontinuous Galerkin method for the solution of Maxwell’s equations. *Computers & Structures* 137, 2–13 <https://doi.org/10.1016/j.compstruc.2013.01.014>
47. Knott EF, Shaeffer JF, Tuley MT (2004) *Radar Cross Section*, 2nd edn. SciTech Publishing, Raleigh, NC. <https://doi.org/10.1049/SBRA026E>