

# TOWARDS A VOLUME MESH GENERATOR TAILORED FOR NEFEM

Xi Zou      Sui Bun Lo      Ruben Sevilla      Oubay Hassan      Kenneth Morgan

*Zienkiewicz Institute for Modelling, Data and AI, Faculty of Science and Engineering  
Swansea University, Swansea SA1 8EN, Wales, United Kingdom  
xi.zou@swansea.ac.uk*

## ABSTRACT

The conversion of digital models from computer-aided design (CAD) to meshes for computer-aided engineering (CAE) is a well-known bottleneck, because of the amount of human intervention required in the cumbersome de-featuring process. This paper introduces the strategy to address this issue by generating the geometric-persistent volume mesh tailored for the NURBS-enhanced finite element method (NEFEM). The NEFEM has been developed to decouple the geometry representation and the numerical approximation to address this issue. NEFEM uses CAD data, the non-uniform rational B-spline (NURBS) that describe the boundary representation (B-rep), to describe the geometry of the computational domain, and it adopts polynomial shape functions for the numerical approximation of the unknown solution. The need for de-featuring is avoided by NEFEM, and the geometric error is completely removed. As a dedicated NEFEM mesh generator is crucial for problems involving complex geometries, this paper introduces the strategy devised to generate volume meshes tailored for NEFEM solvers. The proposed approach is based on previously developed NEFEM surface mesh generation method, and new requirements for validity checks are imposed for the volume mesh generation. A layer of NEFEM volume elements is firstly created, followed by the interior volume mesh which can be obtained from a standard mesh generator. The resulting mesh maintains the exact geometric B-rep, and the element sizes are fully dominated by the user-defined spacing and not by the presence of small geometric features. Examples are presented for an illustrative demonstration of the proposed technique.

**Keywords:** mesh generation, exact geometry, NURBS-enhanced FEM, de-featuring

## 1. INTRODUCTION

Contemporary industrial design requires building computer aided engineering (CAE) models suitable for simulation. This task is known to be a major bottleneck due to the excessive human intervention required when processing the upstream computer aided design (CAD) model [1, 2]. This is mainly due to the fact that CAD models often contain excessive details, which prevent generating a mesh that leads to an efficient numerical simulation [3, 4].

In general, the generation of meshes from complex CAD models largely depends upon the type of simulation, because of the numerous multiscale features which may or may not be negligible for the physical

problem of interest. Traditional mesh generators produce small, often distorted, elements, when the mesh size desired by the user significantly exceeds the dimension of the geometric features. Large research efforts have been made into methods of de-featuring complex CAD models [5, 6]. However, fully automated de-featuring has not yet been achieved. Firstly, it is not always possible to make an accurate prediction of the effect of the de-featuring before actually performing simulations. Secondly, the de-featuring requirements differ from problem to problem, due to their physical nature. For instance, a small feature can be negligible in a low frequency acoustic analysis, but could have a significant impact on the same problem at higher frequencies. Finally, de-featuring also relies

on the desired approximation level, and this is highly dependent on the perspective of the analyst.

This problem has been addressed by the virtual topology concept [7], that provides the capability to modify topological entities without changing the geometry. The strategy is particularly attractive for methods involving high-order interpolation, where coarse elements with a high-order polynomial approximation [8] are preferred to exploit the potential benefits. Nevertheless, the mesh has to be refined to guarantee accurate and reliable results at features involving abrupt geometric changes in terms of the normal to the boundary representation (B-rep) [9].

It is known that the commonly used isoparametric elements discretise the boundary of the computational domain as an approximation of the true B-rep. The accurate CAD data, typically the curves and surfaces parametrised by non-uniform rational B-splines (NURBS), are not used during the numerical simulation. This is also true for a simulation with high order methods. As a result, the polynomial approximated boundary representation will introduce unavoidable geometric error, which can be the dominating error in the simulation [10, 11, 12, 13].

Isogeometric methods replace the approximating polynomial functions with NURBS functions, trying to make use of the exact representation of the domain [14]. However, it requires a fundamental change in the way CAD models are prepared. Traditionally, geometry modelling kernels embedded in all industrial CAD platforms [15] focus on the B-rep of a CAD geometry, while isogeometric methods require a tri-variate NURBS description of the solid domain. Furthermore, in isogeometric methods, small elements are still required when small geometric features are present in the original CAD model.

The NURBS-enhanced finite element method (NEFEM) [16] addresses this problem by a complete separation of the concepts of geometry and solution approximation. Within NEFEM, the NURBS parametrised B-rep, available from the CAD model, is used only for the geometric description of the domain boundary, whereas standard polynomials are used for the approximation of the solution. With such a separation, the error due to geometric approximation is completely removed. In addition, the introduced NEFEM elements are able to traverse curves and surfaces in the B-rep. This implies that the element sizes are not restricted by small geometric features, but are entirely controlled by the user specification, and the need for de-features is consequently avoided. Published solutions to electromagnetics, fluid dynamics, solid mechanics and heat transfer problems demonstrate the potential of the method [17, 18, 19, 20], but the lack of a dedicated mesh generator for NEFEM

has hampered its application to complex problems.

An automatic NEFEM mesh generator for two-dimensional simulations was introduced in [21]. The NEFEM surface mesh generation in three-dimensional space was recently presented for the first time in [22]. This paper presents the latest efforts made towards the development of a three-dimensional volume mesh generator tailored for NEFEM. The boundary discretisation is first performed by remeshing a standard surface mesh, allowing faces traversing multiple surfaces whilst maintaining the exact B-rep. The generation process for the NEFEM volume mesh is discussed in detail, including the new entities that have been devised to store the information required by NEFEM elements. Several illustrative examples will be presented to show the potential of the proposed technique.

## 2. NEFEM FUNDAMENTALS

Let us consider an open bounded domain  $\Omega \subset \mathbb{R}^3$ . The boundary of the domain, denoted by  $\partial\Omega$ , is described by a collection of NURBS curves  $\mathcal{C} := \{\mathbf{C}_i\}_{i=1}^{n_c}$  and surfaces  $\mathcal{S} := \{\mathbf{S}_j\}_{j=1}^{n_s}$ . In particular, each boundary curve or surface can be parametrised as

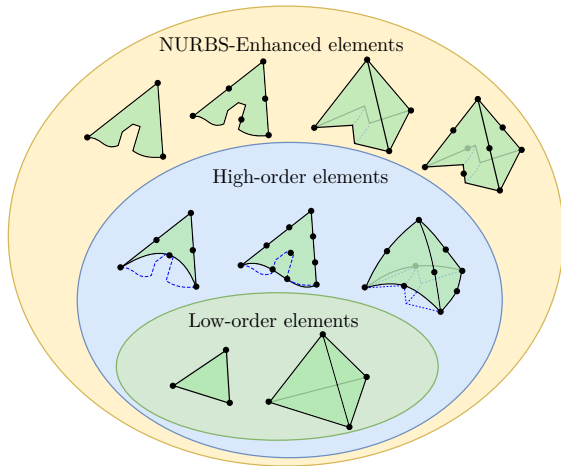
$$\begin{aligned} \mathbf{C}_i &: [0, 1] \longrightarrow \mathbf{C}_i([0, 1]) \subseteq \partial\Omega \subset \mathbb{R}^3; \\ \mathbf{S}_j &: [0, 1]^2 \longrightarrow \mathbf{S}_j([0, 1]^2) \subseteq \partial\Omega \subset \mathbb{R}^3. \end{aligned}$$

A standard FEM mesh is typically generated in a bottom-up manner, following the point, curve, surface and volume hierarchy of the CAD model. In this process, the geometric entities are associated with the meshing entities. Specifically, points of the CAD model define mesh nodes, curves are discretised into edges, surfaces are discretised into facets such as triangles or quadrilaterals, and volumes are divided into elements such as tetrahedra, prisms, pyramids or hexahedra. This procedure naturally induces small elements if the CAD model contains short curves or small surfaces. NEFEM is dedicated to lifting the restriction of the small geometric features inducing small elements, and a new class of elements is introduced.

### 2.1 NEFEM rationale

The key idea of NEFEM [16] is the separation of the geometric approximation and functional approximation that are tightly coupled in isoparametric finite elements and isogeometric methods. By decoupling these two concepts, NEFEM generalises the definition of a finite element: the geometry is exactly described by means of the NURBS parametrised B-rep that can be directly obtained from CAD models, whereas the functional approximation is defined using polynomials, as in standard FEM. As a result, the introduced NEFEM element requires new quadrature rules to ensure that the exact B-rep is accounted for by the solver.

In two dimensions, a typical NEFEM element can be defined as a curved triangle where at least one edge is geometrically defined as a combination of trimmed NURBS curves. 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 design is illustrated in Figure 1 and a detailed discussion can be found in [16].



**Figure 1:** Illustration of the generalisation introduced by the concept of NEFEM elements.

In the illustrative example of Figure 1, 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, Figure 1 also shows that the face of a NEFEM element can be comprised of a collection of NURBS surface patches, even with abrupt changes of the normal within the face.

It is worth noting that NEFEM elements are restricted to a 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 finite elements.

## 2.2 Geometric mapping of NEFEM elements

In standard FEM, the shape functions and their derivatives are defined and evaluated at the integra-

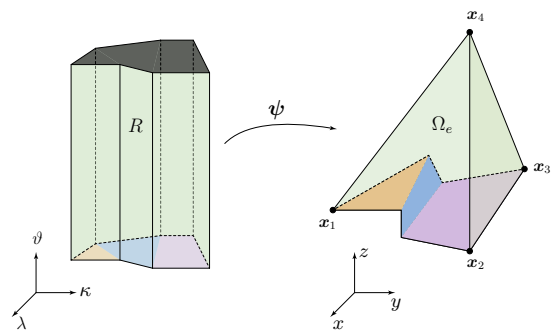
tion points in the reference element. This information is stored and used to compute the elemental matrices and vectors required by the solver. For each element, the isoparametric mapping between the reference and physical element is used.

In NEFEM, the shape functions and the derivatives are defined and evaluated at the integration points of each individual element, directly in the physical domain, and the elementwise matrices and vectors are computed in an *ad hoc* manner. 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 quadratures for NEFEM elements.

To facilitate the quadrature scheme for volumetric NEFEM elements, a particular geometric mapping is devised to encapsulate the NURBS parametrisation of the geometric entities. For instance, in three-dimensional space, a mapping between a polygonal prism and a NEFEM tetrahedron is defined as

$$\begin{aligned} \Psi : R &\longrightarrow \Omega_e \\ (\lambda, \kappa, \vartheta) &\longmapsto \Psi(\lambda, \kappa, \vartheta) := (1 - \vartheta)\mathcal{S}(\lambda, \kappa) + \vartheta\mathbf{x}_4, \end{aligned} \quad (1)$$

where  $\mathcal{S}(\lambda, \kappa)$  is the parametrisation of the curved boundary face, which might be piecewise when involving multiple NURBS surfaces, and  $\mathbf{x}_4$  denotes the node interior to the domain. This mapping is illustrated in Figure 2 for a tetrahedral element with one face traversing three trimmed NURBS surfaces that are rendered in distinguished colours.



**Figure 2:** Illustration of the NEFEM geometry mapping for a tetrahedral element with one face defined on three trimmed NURBS surfaces.

In practice, the piecewise definition of a NEFEM element face, as the bottom face shown in Figure 2, 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. This elementwise sub-mesh is further discussed later in Section 3.3.

### 3. NEFEM SURFACE MESH GENERATION

In this section, the generation of NEFEM surface mesh is briefly recalled. In addition, newly developed checks that are performed to ensure that a valid volume mesh can be generated from the surface mesh are presented. Triangle and tetrahedron elements are considered in this work.

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 must encapsulate the NURBS definition of the geometry;
3. The surface elements should pass visibility checks to enable the efficient creation of volume elements avoiding self-intersections.

The first two requirements have been addressed in previous work [22]. The last requirement is posed to facilitate the volume mesh generation, and it will be detailed later in Section 4.3.

#### 3.1 Surface meshing strategy

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.

#### 3.2 GS-points

To register the intersection between an element edge traversing multiple surfaces and a geometric curve, the so-called *geometric supporting points*, or *GS-points* are introduced. The GS-points are associated with their parent elements and are used for mesh generation purposes. It is worth emphasising that they do not introduce any additional degrees of freedom in the solver. However, they will be used when devising

piecewise quadratures for numerical integration over the faces traversing multiple surfaces and the associated elements.

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. In this paper, a convention is introduced to render all vertex nodes in black dots, while all GS-points are illustrated with green dots, as illustrated in Figure 9.

#### 3.3 The sub-mesh

As mentioned in Section 2.2, the 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, along with the GS-points, is used for quadrature as it naturally forms the integration cells. The sub-mesh can also be used to represent physical interfaces inside a NEFEM element [19]. It is worth noting that an integration cell cannot traverse surfaces, and an element traversing multiple surfaces must contain at least two integration cells.

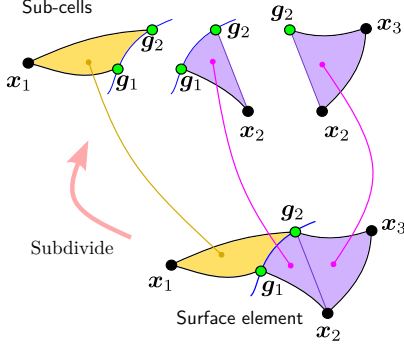
In addition, the sub-mesh also plays an important role during the NEFEM mesh generation. Operations like edge split and edge flip are common in a mesh generation process. Unlike the operations in standard mesh generators, the sub-mesh is required during the NEFEM mesh generation because the operations 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.

A typical example of the sub-mesh of a NEFEM surface element is shown in Figure 3. The definition of the triangular surface element with three nodes  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\mathbf{x}_3$ , where edges  $E(\mathbf{x}_1, \mathbf{x}_2)$  and  $E(\mathbf{x}_1, \mathbf{x}_3)$  are traversing a NURBS curve, requires the two GS-points,  $\mathbf{g}_1$  and  $\mathbf{g}_2$ , to register the edge-curve intersections. The two GS-points and three nodes have defined the vertices for three sub-cells that belong to two different surfaces.

#### 3.4 Validity check

Validity checks are performed during the creation of NEFEM surface elements to facilitate the creation of volume elements in the next stage.

The first check is performed before collapsing an element edge that traverses multiple intersection curves



**Figure 3:** Sub-mesh of a typical NEFEM surface element. Sub-cells belong to different surfaces are filled with distinguishing colours. The intersection curve is coloured in blue.

or surfaces. This check is closely related to the visibility check in Section 4.3 to avoid possible self-intersection in the volume elements. In addition, a second check is performed, after having created or updated the NEFEM elements, and this check will also try to fix self-intersections by curving the sub-edges.

The first check is carried out when trying to create a new NURBS-enhanced edge. 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 Figure 4, a local feature appearing to be a U-shaped channel involves five surfaces  $\{S_i\}$  for  $i = 1, \dots, 5$ . When trying to collapse the short edges 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

$$\begin{cases} \mathbf{n}_i \cdot \mathbf{n}_j \geq -1/2, & \Rightarrow \text{pass;} \\ \mathbf{n}_i \cdot \mathbf{n}_j < -1/2, & \Rightarrow \text{fail,} \end{cases} \quad (2)$$

where  $\mathbf{n}_i$  is the normal to surface  $S_i$ , and the normals are computed locally at the corresponding nodes of the sub-mesh.

Figure 4(b) presents two examples when testing possible new NURBS-enhanced edges before the edge collapse. The first candidate edge,  $E(\mathbf{x}_5, \mathbf{x}_7)$ , is obtained from collapsing node  $\mathbf{x}_6$  to  $\mathbf{x}_7$ , and the normal  $\mathbf{n}_3$  at the target node  $\mathbf{x}_7$  is compared with all normals involved in this collapse, such as  $\mathbf{n}_4$  at GS-point  $\mathbf{g}_2$  and  $\mathbf{n}_5$  at node  $\mathbf{g}_5$ . This case will pass the validity check. A second option involves the candidate NURBS-enhanced edge  $E(\mathbf{x}_1, \mathbf{x}_4)$ , the normal  $\mathbf{n}_4$  at target node  $\mathbf{x}_4$  is opposite to  $\mathbf{n}_2$  at  $\mathbf{x}_1$ . Therefore, the second configuration will fail the validity check and thus the collapse of edge  $E(\mathbf{x}_2, \mathbf{x}_4)$  will be prevented.

In the rare case that a sub-edge intersects with an

---

**Algorithm 1:** Validity check routine.

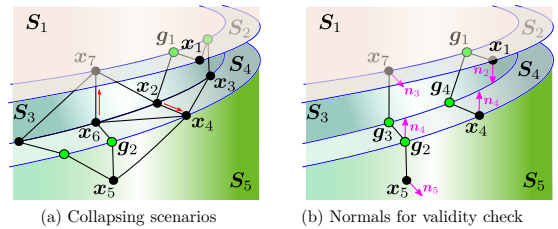
---

**Input:** Edge to collapse  $E(\mathbf{x}_b, \mathbf{x}_t)$ , with base node  $\mathbf{x}_b$ , target node  $\mathbf{x}_t$

**Output:** Boolean value `isValid`

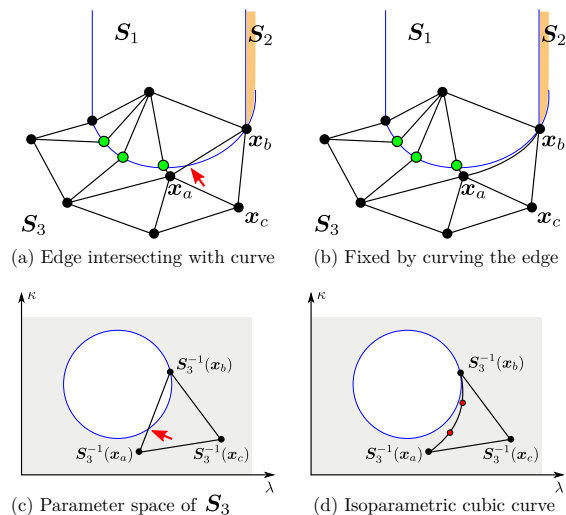
- 1 Collect the set,  $\mathcal{S}_t$ , of all parent surfaces of the target node;
  - 2 Collect the set,  $\mathcal{S}_c$ , of all traversed surfaces for the collapse;
  - 3 **for**  $S_i \in \mathcal{S}_t$  **do**
  - 4     **if**  $S_i \notin \mathcal{S}_c$  **then**
  - 5         Remove  $S_i$  from  $\mathcal{S}_t$ ;
  - 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` = `true`;
  - 11 **for**  $n_i \in \mathcal{N}_t$  **do**
  - 12     **for**  $n_j \in \mathcal{N}_c$  **do**
  - 13         **if**  $\mathbf{n}_i \cdot \mathbf{n}_j < -1/2$  **then**
  - 14             `isValid` = `false`;
  - 15             **return**;
  - 16         **end if**
  - 17     **end for**
  - 18 **end for**
- 

intersection curve, as shown in Figure 5(a), the second validity check will detect it and fix it by curving the sub-edge. The intersection between edge  $E(\mathbf{x}_a, \mathbf{x}_b)$  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 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 Figure 5(c). A simple fix to this problem is performed by replacing the straight edge  $E(\mathbf{x}_a, \mathbf{x}_b)$  in the parametric space by a cubic curve, as shown



**Figure 4:** Illustration of the validity check at a U-channel feature. (a) Two edge collapsing scenarios: from  $\mathbf{x}_6$  to  $\mathbf{x}_7$ , and from  $\mathbf{x}_2$  to  $\mathbf{x}_4$ . (b) Proposed new NURBS-enhanced edges, showing selected normals for validity checks.

in Figure 5(d). The fixed scenario after the second validity check is shown in Figure 5(b).



**Figure 5:** Illustration of the self-intersection fix at the bottom of a cylindric 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.

## 4. NEFEM VOLUME MESH GENERATION

This section presents the latest efforts made towards the generation of NEFEM volume meshes.

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:

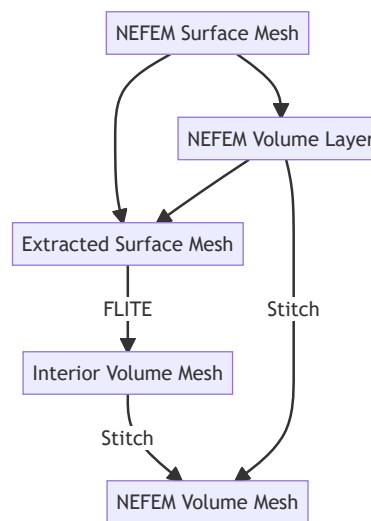
- An element with at least one face located on the boundary.
- An element with at least one edge but with no faces located on the boundary.

An element of each type is further classified in terms of the number of intersection surfaces of the CAD model that it traverses. Elements with faces or edges not traversing multiple surfaces are grown using the same technique available in standard mesh generators. They are still flagged for the purpose of accounting for the NURBS boundary representation by the solver. However, special care must be taken for elements with faces or edges traversing multiple surfaces.

### 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 integration cells.

The strategy for the volume meshing is to first generate a layer of NURBS-enhanced volume elements that covers the featured surface. Next, the exterior facets of the first layer of volume elements are extracted to form a new surface mesh. This extracted surface mesh will only contain standard 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. The NEFEM volume mesh is finally achieved by stitching the NEFEM layer and the standard interior elements. In this work, the FLITE mesh generator [23] is used to create standard meshes, and the procedure is listed in Figure 6.



**Figure 6:** NEFEM volume mesh generation procedure.

**Remark 1** The stitching of boundary layer mesh and interior mesh, as discussed in [24, 25], is an established procedure in standard mesh generation. The presented strategy 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.

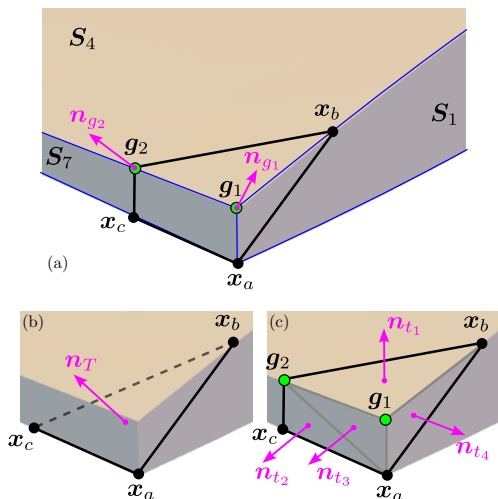
### 4.2 Growing volume elements

To guide the growth of volume elements into the three-dimensional domain, the normal vectors are firstly computed based on the surface mesh. Unlike standard triangle elements, a NEFEM triangle element can have a non-unique definition of its normal as it can traverse

multiple surfaces. Thus, it is not trivial to evaluate the normal for a face or its edges.

As mentioned in Section 3.3, each 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 surfaces. Several smoothing options have been tested, such as surface-based averaging, weighted averaging and Laplacian. It was found that the surface-based averaging provides satisfactory normal vectors in the tested geometries.

The possible choices of the normal to grow a volume element from a typical NEFEM surface element are illustrated in Figure 7. The surface element with vertices  $\{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c\}$  traverses surfaces  $\{S_1, S_4, S_7\}$ . The smoothed normal at GS-points  $g_1$  and  $g_2$  are shown in Figure 7(a), the naive normal evaluated using only the vertices is plotted in Figure 7(b) at the apparent centroid. In contrast, Figure 7(c) shows the four normal associated to each integration cell depicted at the centroid of each integration cell.

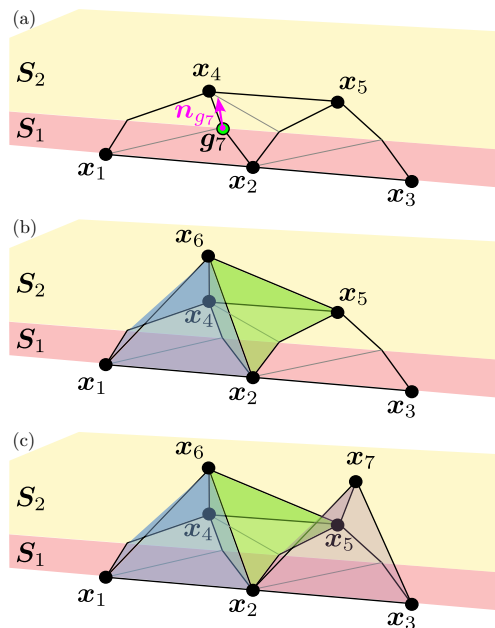


**Figure 7:** Choices of normal vectors to grow a tetrahedral element. (a) Normal vectors at GS-points. (b) Normal vector at plane element centroid. (c) Normal vectors at integration cell centroids.

The major process used to grow volume elements based on the surface mesh is finding a suitable normal. The first attempt is to loop through all edges in the surface mesh, and check the dihedral angle  $\theta$ . A tetrahedron will be created by linking the two vertices when  $\theta < 2\pi/3$ , closing the two triangles joint by the edge. 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 self-intersection checks are performed via evaluating the volume of the newly formed sub-cell tetrahedra. If a self-intersection is identified, the normal vector is tuned by scaling. The base point of the normal vector is changed by sliding along the edge to find another suitable location that is free from self-intersection.

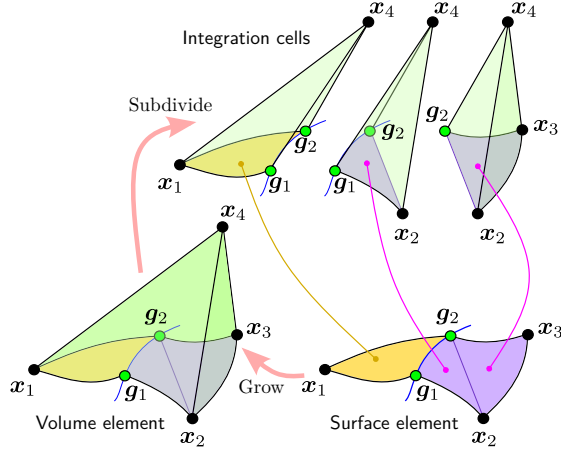
A typical scenario for volume element growth is illustrated in Figure 8. Three surface elements are traversing surfaces  $S_1$  and  $S_2$  that are rendered in red and yellow. During an edge-based loop, a suitable normal vector  $\mathbf{n}_{g_7}$  is found at GS-point  $g_7$  of edge  $E(\mathbf{x}_2, \mathbf{x}_4)$ , as depicted by Figure 8(a). 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 Figure 8(b). 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 Figure 8(c), a new tetrahedron will be created by simply linking the existing vertices  $\mathbf{x}_6$  and  $\mathbf{x}_7$ .



**Figure 8:** Typical scenario to grow tetrahedral elements. (a) Normal vectors at GS-point to guide the growth. (b) Two tetrahedra grown. (c) Other grown tetrahedra, vertices  $\mathbf{x}_6$  and  $\mathbf{x}_7$  to be linked to form a new tetrahedron.

As a tetrahedral element is grown based on a surface element, such as shown in Figure 3, it inherits the subdivision of the surface element and three sub-tetrahedra are grown to form the integration cells

for computing quadrature over the NURBS-enhanced tetrahedron. This can also be viewed as a straightforward subdivision of the tetrahedron element with the guidance of the sub-cells on a traversing face, as illustrated in Figure 9.



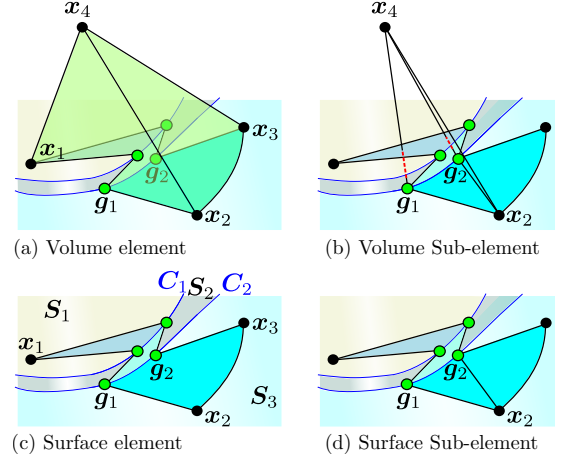
**Figure 9:** Sub-mesh of a typical NEFEM volume element grown from a NEFEM surface element.

### 4.3 Self-intersection check

Taking the creation process of triangle to tetrahedron as the example, the objective is to ensure the top vertex of the tetrahedron is visible to any point in the base triangle. 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. 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 Figure 10 presents a scenario in which a violation occurs, resulting in a self-intersecting NEFEM volume element. The volume element shown in Figure 10(a) is based on the surface element in Figure 10(c), 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



**Figure 10:** Illustration of visibility issue for a NEFEM volume element and its base surface element at a step feature. (a) Volume element based on surface element in (c). (b) A volume sub-cell based on (d) exhibits self-intersections within the volume element. (c) Surface element. (d) A surface sub-cell.

surfaces  $S_1$  and  $S_3$ . A surface integration cell with nodes  $\{g_1, x_2, g_2\}$ , as shown in Figure 10(d), forms the bottom face of the volume integration cell illustrated in Figure 10(b). As 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  $E(x_4, x_3)$ .

## 5. EXAMPLES

This section presents some examples to demonstrate the strategy described in the previous section to generate NEFEM volume meshes.

### 5.1 A flat plate with two cylinders

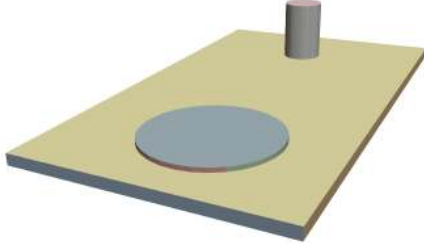
The first example considers a flat plate with two cylinders, as shown in Figure 11. The geometric data is listed in Table 1. The original FEM mesh, not com-

**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

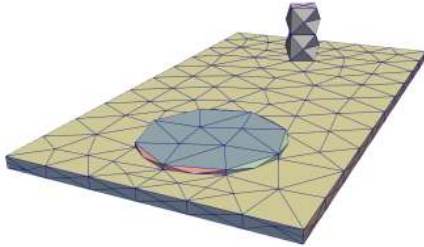
plying with the user-defined spacing, is shown in Fig-





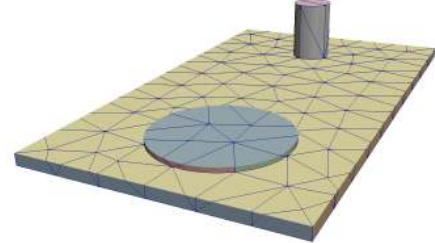
**Figure 11:** NURBS surfaces in the CAD model of a flat plate with two cylinders.

ure 12. The NEFEM surface mesh, obtained by using

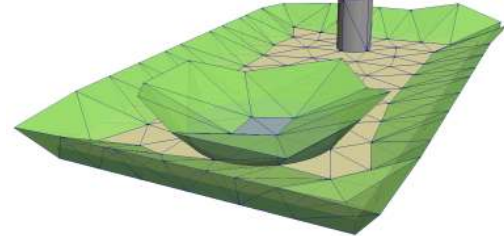


**Figure 12:** FEM surface mesh of the flat plate intersected by two cylinders, showing elements not complying with the user-defined spacing.

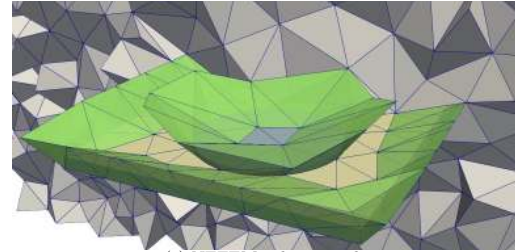
the strategy proposed in [22], is generated with a uniform mesh size that is independent of the thickness of the plate and the heights or diameters of the cylinders, as presented in Figure 13(c). Based on this boundary discretisation, with the desired element size, the first layer of NEFEM volume elements is generated with the strategy described in Section 4, as shown in Figure 13(d). The volumetric elements are rendered in green. The interior volumetric elements are generated based on the extracted exterior faces of the NEFEM volume layer and the surface mesh, which form a standard triangulation, with the Delaunay method for tetrahedral mesh. For this model, there are 121 tetrahedra in the NEFEM volume layer, and 50 815 interior volumetric elements. As it can be seen in Figure 13(e), there are much fewer NEFEM volume elements than the standard elements, thus the computational cost introduced by the NEFEM elements would have minor impact to the solver, while the minimum element size is significantly improved. As it is listed in Table 2, the minimum element edge length normalised with the user-specified spacing for NEFEM mesh has increased more than 10 times. This is considered crucial when transient simulations using an explicit time marching are of interest.



(a) NEFEM surface mesh



(b) NEFEM volume layer



(c) NEFEM volume mesh

**Figure 13:** NEFEM meshing process for the flat plate intersected by two cylinders. (a) NEFEM surface mesh. (b) NEFEM volume element layer rendered in green. (c) Clipped NEFEM volume mesh including the interior volumetric elements.

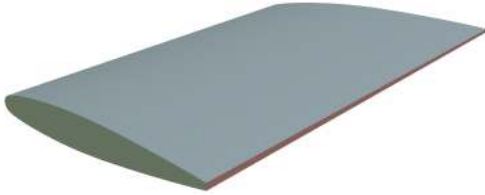
**Table 2:** Normalised edge lengths for the flat plate intersected by two cylinders.

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

## 5.2 A wing with a blunt trailing edge

This example considers the generation of the NEFEM mesh for a wing with a blunt trailing edge. In this example, the ability to handle a user-specified non-uniform mesh spacing is also considered.

The NURBS surfaces defining the wing are presented in Figure 14 and the geometric data for this model is summarised in Table 3. A non-uniform mesh spacing has been specified using two line sources at both the leading and trailing edges with a stretching ratio equal to five. Though refinement is introduced by the line sources, the prescribed mesh size is greater than the length of the shortest curve. The resulting FEM mesh, shown in Figure 15, contains numerous small elements



**Figure 14:** NURBS surfaces in the CAD model of a wing with blunt trailing edge.

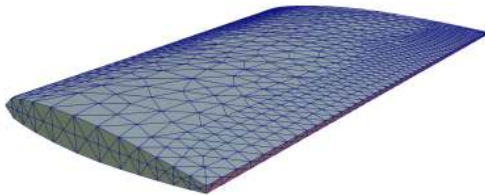
**Table 3:** Geometric data of the wing model.

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

**Table 4:** 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

at the blunt trailing edge as well as the wing tip. The

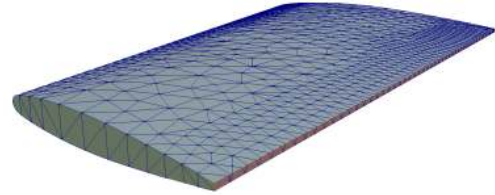


**Figure 15:** FEM surface mesh of the wing, showing elements not complying with the user-defined spacing.

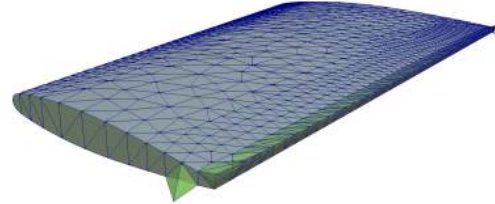
NEFEM surface mesh shown in Figure 16(c) has eliminated the small elements, where the NURBS volume layer are created at the blunt trailing edge feature as presented in Figure 16(d). The number of NURBS-enhanced tetrahedral elements is 179, while that of the interior volumetric elements is 1 458 208. From Table 4 it can be seen the normalised minimum edge length in NEFEM mesh exceeds 7 times compared to the FEM mesh. This again demonstrates that the NEFEM element takes only a negligible portion of the whole mesh, but can improve the minimum element size to enable large time-stepping in explicit solvers.

### 5.3 Falcon aircraft

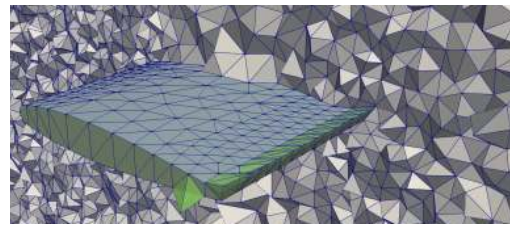
In this example, a full aircraft model is considered to demonstrate the capability of handling complex geometries. A variety of geometric features are present in the CAD geometry, such as very short curves and small surfaces, particularly at the wing tips. The char-



(a) NEFEM surface mesh



(b) NEFEM volume layer

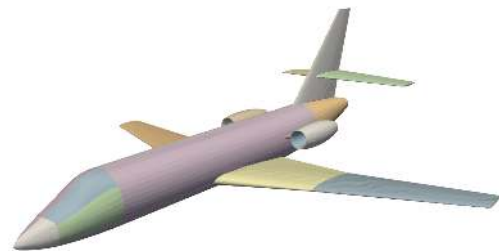


(c) NEFEM volume mesh

**Figure 16:** NEFEM mesh process for the wing with a blunt trailing edge. (a) NEFEM surface mesh. (b) NEFEM volume element layer rendered in green. (c) Clipped NEFEM volume mesh including the interior volumetric elements.

acteristic thickness of the wing is about 0.2, which is shorter than the minimum curve length, and this poses a challenge especially for the surface mesh generation.

The NURBS surfaces of the CAD model are presented in Figure 17 and the geometric data is summarised in Table 5. The original FEM mesh, not complying



**Figure 17:** NURBS surfaces in the CAD model of the Falcon aircraft.

with the user-defined spacing, is shown in Figure 18. Despite all surface elements are NURBS-enhanced as they discretised the complex B-rep, only a few elements are traversing multiple surfaces. Therefore, the

**Table 5: 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

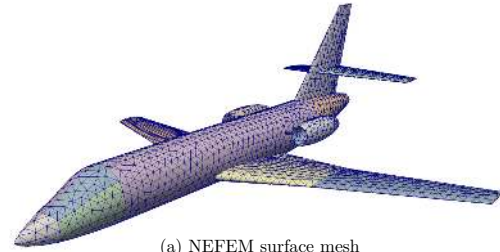
**Figure 18:** FEM surface mesh of the Falcon, showing elements not complying with the user-defined spacing.**Table 6: Normalised edge lengths for the Falcon model.**

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

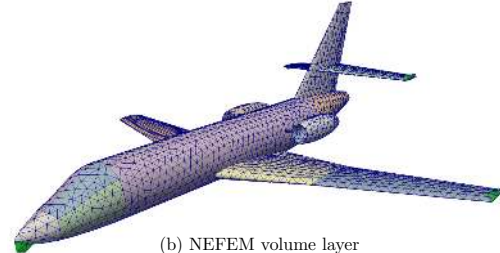
first layer of NEFEM volume elements, as it focuses on the growth of traversing volume element, contains a small number of tetrahedral elements, as presented in Figure 19. The total number of tetrahedral elements in the volume mesh is 229 693, in which the number of NEFEM tetrahedron involving multiple definition during integration is 28. In Table 6 it is presented that the normalised minimum edge length in NEFEM mesh has improved 4.5 times than the FEM mesh. It is worth noting that, a very small number of elements with a spacing well below the user-defined spacing is enough to make unfeasible the solution of a transient problem with explicit time marching. Therefore, the ability to lift this restriction has massive implications for the solver.

## 6. CONCLUDING REMARKS

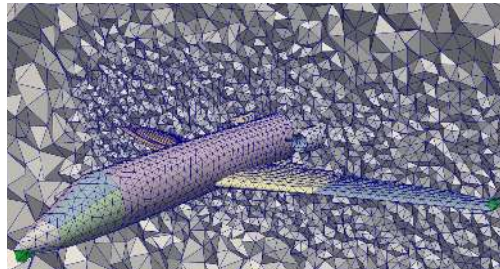
A method dedicated to generating volume meshes tailored for NEFEM has been presented for the first time. The technique is capable of generating volume meshes where the exact boundary representation, provided by the NURBS parametrisation from the CAD model, is encapsulated in the geometric definition of NEFEM elements. As a result, the small geometric features present in the CAD model no longer restrict the element size in NEFEM meshes. This completely removes the need for the time-consuming de-featuring process on complex CAD models and, at the same time, eliminates the geometric error introduced by the



(a) NEFEM surface mesh



(b) NEFEM volume layer



(c) NEFEM volume mesh

**Figure 19:** NEFEM mesh process for the Falcon. (a) NEFEM surface mesh. (b) NEFEM volume element layer rendered in green. (c) Clipped NEFEM volume mesh including the interior volumetric elements.

de-featuring process or by traditional mesh generators.

Given a CAD geometry in the form of B-rep, the proposed strategy starts by generating an initial surface mesh using a standard mesh generator. Guided by the user-defined spacing, elements near the under-sized 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 suitable for the volume mesh generation stage. Various normal vectors are defined and are 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, 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.

Examples have been presented to demonstrate the applicability and potential of the proposed method. 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 wing with a blunt trailing edge. The resulting NEFEM meshes demonstrate a spacing closely matching the user-specification, even when the CAD model contains small features.

Future work will involve the new definition and improvement of the element quality, the extension to high-order interpolations, and the integration with a NEFEM solver for practical applications.

## References

- [1] Thakur A., Banerjee A.G., Gupta S.K. “A survey of CAD model simplification techniques for physics-based simulation applications.” *Computer-Aided Design*, vol. 41, no. 2, 65–80, 2009
- [2] Shapiro V., Tsukanov I., Grishin A. “Geometric issues in computer aided design/computer aided engineering integration.” *Journal of Computing and Information Science in Engineering*, vol. 11, no. 2, 2011
- [3] Gammon M., Bucklow H., Fairey R. “A review of common geometry issues affecting mesh generation.” *AIAA Aerospace Sciences Meeting*, 2018
- [4] Park M.A., Kleb W.L., Jones W.T., Krakos J.A., Michal T.R., Loseille A., Haines R., Dannenhoffer J. “Geometry Modeling for Unstructured Mesh Adaptation.” *AIAA Aviation 2019 Forum*, p. 2946. 2019
- [5] Mobley A.V., Carroll M.P., Canann S.A. “An Object Oriented Approach to Geometry Defeaturing for Finite Element Meshing.” *7th International Meshing Roundtable (IMR)*, pp. 547–563. 1998
- [6] Danglade F., Pernot J.P., Véron P. “On the use of machine learning to defeature CAD models for simulation.” *Computer-Aided Design and Applications*, vol. 11, no. 3, 358–368, 2014
- [7] Sheffer A., Bercovier M., Blacker T., Clements J. “Virtual topology operators for meshing.” *International Journal of Computational Geometry & Applications*, vol. 10, no. 03, 309–331, 2000
- [8] Wang Z.J., Fidkowski K., Abgrall R., Bassi F., Caraeni D., Cary A., Deconinck H., Hartmann R., Hillewaert K., Huynh H.T., et al. “High-order CFD methods: current status and perspective.” *International Journal for Numerical Methods in Fluids*, vol. 72, no. 8, 811–845, 2013
- [9] Blacker T.D., Owen S.J., Staten M.L., et al. “CUBIT geometry and mesh generation toolkit 15.1 user documentation.” Tech. rep., Sandia National Lab.(SNL-NM), 2016
- [10] Dawson M., Sevilla R., Morgan K. “The application of a high-order discontinuous Galerkin time-domain method for the computation of electromagnetic resonant modes.” *Applied Mathematical Modelling*, vol. 55, 94–108, 2018
- [11] Sevilla R. “HDG-NEFEM for two dimensional linear elasticity.” *Computers & Structures*, vol. 220, 69–80, 2019
- [12] Bassi F., Rebay S. “High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations.” *Journal of Computational Physics*, vol. 138, no. 2, 251–285, 1997
- [13] Xue D., Demkowicz L. “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*, vol. 2, no. 3, 283–300, 2005
- [14] Hughes T.J.R., Cottrell J.A., Bazilevs Y. “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement.” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39–41, 4135–4195, 2005
- [15] Taylor N.J., Haines R. “Geometry Modelling: Underlying Concepts and Requirements for Computational Simulation.” *2018 Fluid Dynamics Conference*, p. 3402. 2018
- [16] Sevilla R., Fernández-Méndez S., Huerta A. “NURBS-Enhanced Finite Element Method (NEFEM): A seamless bridge between CAD and FEM.” *Archives of Computational Methods in Engineering*, vol. 18, no. 4, 441–484, 2011
- [17] Sevilla R., Fernández-Méndez S., Huerta A. “NURBS-enhanced finite element method (NEFEM).” *International Journal for Numerical Methods in Engineering*, vol. 76, no. 1, 56–83, 2008
- [18] Sevilla R., Fernández-Méndez S., Huerta A. “NURBS-enhanced finite element method for Euler equations.” *International Journal for Numerical Methods in Fluids*, vol. 57, no. 9, 1051–1069, 2008
- [19] Soghrati S., Merel R.A. “NURBS enhanced HIFEM: A fully mesh-independent method with zero geometric discretization error.” *Finite Elements in Analysis and Design*, vol. 120, 68–79, 2016

- [20] Tan M.H., Safdari M., Najafi A.R., Geubelle P.H. “A NURBS-based interface-enriched generalized finite element scheme for the thermal analysis and design of microvascular composites.” *Computer Methods in Applied Mechanics and Engineering*, vol. 283, 1382–1400, 2015
- [21] Sevilla R., Rees L., Hassan O. “The generation of triangular meshes for NURBS-enhanced FEM.” *International Journal for Numerical Methods in Engineering*, vol. 108, no. 8, 941–968, 2016
- [22] Zou X., Sevilla R., Hassan O., Morgan K. “Towards a Surface Mesh Generator Tailored for NE-FEM.” *29th International Meshing Roundtable (IMR), Virtual Conference*. Oct. 2021
- [23] Sørensen K., Hassan O., Morgan K., Weatherill N. “A multigrid accelerated hybrid unstructured mesh method for 3D compressible turbulent flow.” *Computational Mechanics*, vol. 31, no. 1-2, 101–114, 2003
- [24] Owen S.J., Saigal S. “Formation of pyramid elements for hexahedra to tetrahedra transitions.” *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 34, 4505–4518, May 2001
- [25] Field D.A. “Automatic generation of transitional meshes.” *International Journal for Numerical Methods in Engineering*, vol. 50, no. 8, 1861–1876, 2001