TOWARDS A SURFACE MESH GENERATOR TAILORED FOR NEFEM

Xi Zou Ruben Sevilla Oubay Hassan Kenneth Morgan

Zienkiewicz Centre for Computational Engineering, Faculty of Science and Engineering
Swansea University, Swansea SA1 8EN, Wales, United Kingdom
xi.zou@swansea.ac.uk

ABSTRACT

Preparing CAD models for CAE simulations is a major bottleneck when dealing with small geometric features. The process often involves de–featuring, which cannot be fully automated and requires many hours of human interaction because it depends upon the physics considered. The NURBS–enhanced finite element method (NEFEM) decouples the concepts of geometry and solution approximation to avoid this problem. NEFEM describes the geometry using the boundary representation (B–rep) from CAD models, and approximates the solution using polynomial functions. As a result, NEFEM completely removes the error due to geometric approximation and the need for de–featuring. In this paper the meshing strategy in two dimensions is recalled. The proposed approach involves a modified advancing front technique where boundary elements have edges that are described by a collection of boundary curves. The strategy required to utilise the meshes into an existing solver is described and an example is used to demonstrate the benefits of the proposed approach. Next, a novel three dimensional surface mesh generation strategy is presented. The technique starts with an initial FEM mesh, where elements smaller than the user defined spacing are present due to CAD model containing small features. A locally enhanced advancing front method is then used to modify the mesh and ensure that the final mesh contains element sizes that better match the user defined spacing. The resulting surface mesh contains elements that cross intersection curves of the CAD and exactly retain the original B–rep. Several examples are presented to show the potential of the proposed technique.

Keywords: exact geometry, NURBS–enhanced FEM, de–featuring, advancing front

1. INTRODUCTION

Preparing geometric models suitable for engineering simulations is known to be a major bottleneck when dealing with complex geometries. This is caused by the excessive number of human hours that are required to transfer the information from a computer aided design (CAD) model to a suitable computer aided engineering (CAE) model. CAD models frequently involve a level of detail much greater than that required to perform a numerical simulation [1].

Over the last two decades, a large effort has been placed on the research of algorithms for the de–featuring of complex CAD models [2, 3]. However, de–featuring cannot be fully automated. First, it is usually not possible to know, a priori, the effect of de–featuring on the results of a simulation. Second, the requirements of the de–featuring depends on the physics to be simulated. As an example, mesh requirements are different for fluid dynamics, electromagnetics, acoustics, heat transfer or structural mechanics applications. Finally, de–featuring is also highly dependent upon the level of approximation required. For instance, the de–featuring required by a viscous turbulent flow simulation is different to the de–featuring required by a inviscid flow simulation.

When small geometric features are present in a CAD model, traditional mesh generators will produce small, often distorted, elements, when the spacing specified

by the user is much larger than the size of the small features. This has a number of implications in the efficiency of the solver. Even if high quality elements can be generated, a large discrepancy in element sizes might have an impact on the iterative solver used to solve the large sparse systems that are encountered with an implicit time integrator. When an explicit time integrator is used, small elements pose a major restriction on the size of the time step that can be used in a numerically stable time marching process.

The virtual topology concept [4] addresses this issue, by providing the capability to merge topological entities without introducing any change in the geometry. This strategy is particularly attractive for high–order methods, where exploitation of the potential benefits require the use of coarse elements with a high–order polynomial approximation [5]. However, when features induce abrupt changes in the normal to the boundary representation (B–rep), the mesh must be refined to guarantee accurate and reliable results [6].

In addition, when isoparametric elements are used, the boundary of the computational domain becomes an approximation of the true B–rep. Even with high order methods, the geometric error induced by a polynomial boundary representation can be the dominant error in a numerical simulation [7, 8].

Isogeometric methods propose a change of basis functions, from polynomial to NURBS, to provide an exact representation of the domain. However, this requires a profound change in the way CAD models are understood. Rather than working with the B–rep representation, as is done by the geometry modelling kernels embedded in all industrial CAD platforms [9], isogeometric methods require a trivariate NURBS description of the solid domain. Furthermore, isogeometric methods require the use of small elements when small features are present in the original CAD model.

The NURBS–enhanced finite element method (NEFEM) [10] addresses this problem by completely decoupling the concepts of geometric and solution approximation. These two concepts are tightly coupled in the large majority of available solvers through the isoparametric mapping between a well–shaped reference element and the deformed geometric–fitting elements. With NEFEM, the geometric description of the boundary uses the B–rep, immediately available in the CAD model, whereas the approximation of the solution still uses polynomial functions. As a result, the error due to geometric approximation is completely removed. This concept avoids completely the need for de–featuring and, at the same time, can use meshes in which the size of the elements is entirely decided by the user and is no longer limited by the presence of geometric features that are smaller than the specified spacing. The potential of the method has been demonstrated using academic problems in electromagnetics, fluids, solids and heat transfer problems [11, 12], but the application to complex geometries has been hampered by the lack of automatic mesh generation tools.

The technique is particularly attractive in a high–order context, where the use of coarse curvilinear meshes is of interest. Here, it is known that a standard isoparametric formulation can introduce geometric errors that dominate over the solution approximation error [13, 14].

The development of an automatic mesh generator for NEFEM in two dimensions was recently reported in [15]. This paper will present the results of initial efforts made towards the development of a fully automatic surface mesh generator for NEFEM, where boundary faces span across multiple surfaces and retain the exact B–rep. The surface mesh generation process will be detailed, including the new data structures that have been devised to store the information required by NEFEM elements. New algorithms, in particular, the local enhanced advancing front method, have been implemented to fulfil the creation of boundary faces that span across multiple surfaces. Several examples will be presented to show the potential of the proposed technique.

2. NEFEM FUNDAMENTALS

Let us consider an open bounded domain $\Omega \subset \mathbb{R}^{\mathtt{n_{sd}}}$, where $\mathtt{n_{sd}}$ is the number of spatial dimensions. The boundary of the domain is assumed to be described by a collection of M non–uniform rational B–splines (NURBS). In two dimensions, each boundary curve is parametrised as

$$\boldsymbol{C} : [0,1] \longrightarrow \boldsymbol{C}([0,1]) \subseteq \partial\Omega \subset \mathbb{R}^2.$$

Similarly, in three dimensions every boundary surface is assumed to be parametrised as

$$\boldsymbol{S} : [0,1]^2 \longrightarrow \boldsymbol{S}([0,1]^2) \subseteq \partial\Omega \subset \mathbb{R}^3.$$

A finite element mesh is usually generated in a hierarchical manner. Given a CAD model, its entities are classified as zero dimensional points, one dimensional curves, two dimensional surfaces and three dimensional volumes. The points of the CAD model are defined as nodes by the mesh generator. Curves are discretised using edges, surfaces are discretised using facets, e.g. triangles or quadrilaterals, and, finally, volumes are discretised using elements, e.g. tetrahedra, hexahedra, prisms or pyramids. This approach naturally introduces small elements when the CAD model contains small curves or surfaces.

2.1 NEFEM rationale

The key idea of NEFEM [10] is to decouple the concepts of geometric approximation and functional approximation that are strongly coupled in isoparametric finite elements and isogeometric methods. The geometry is exactly represented by means of the B–rep that is readily available in CAD models, while the functional approximation is defined using polynomials, as in standard finite elements. This introduces a new type of finite element that requires new quadrature rules to ensure that the exact B–rep is accounted for by the solver.

In two dimensions, this new element type is defined as a triangular element where at least one edge is geometrically defined as a collection of trimmed NURBS curves. Similarly, in three dimensions, the new element type is defined as a tetrahedral element where at least one face is geometrically defined as a collection of trimmed NURBS surfaces. This new concept is illustrated in Figure 1.

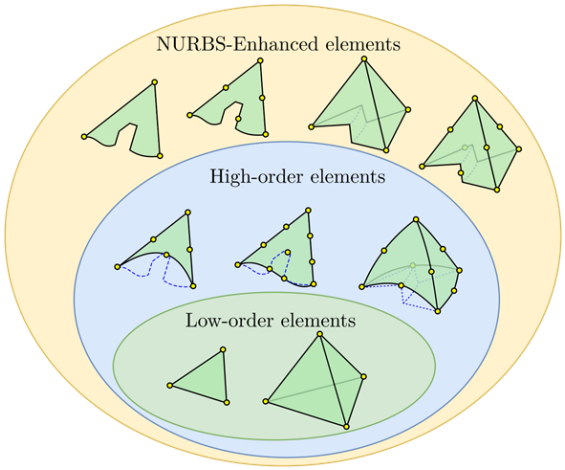

Figure 1: Illustration of the generalisation introduced by the concept of NEFEM elements.

It can be observed that in NEFEM the geometry is always given by the exact B–rep, independent of the order of approximation used in the element. In addition, this figure also shows that a face of an element can be described using a collection of NURBS, even with abrupt changes in the normal on a single face.

It is worth noting that NEFEM elements are restricted to a layer of elements around curved boundaries. The large majority of elements in a mesh do not have one edge or face on the boundary, so that a standard isoparametric FEM approach can be used. This means that activating this type of element near the boundary introduces a negligible computational overhead compared to standard finite elements.

2.2 NEFEM solver

To compute the element matrices and vectors required by any finite element solver, the information usually stored is the value of the shape functions at the integration points in a reference element. The incorporation of the NEFEM concept into an existing solver can be readily accomplished by devising tailored quadrature schemes for those elements with at least one face or one edge given by a collection of trimmed NURBS.

To define the numerical quadrature in a NEFEM triangle, a mapping between a rectangle and the triangle is defined as

$$\begin{aligned} \boldsymbol{\psi} &: R \longrightarrow \Omega_e \\ (\lambda, \vartheta) &\longmapsto \boldsymbol{\psi}(\lambda, \vartheta) := (1-\vartheta)\boldsymbol{C}(\lambda) + \vartheta \boldsymbol{x}_3, \end{aligned} \tag{1}$$

Here, $\boldsymbol{C}(\lambda)$ is the parametrisation of the curved boundary, which might be given by several NURBS curves and $\boldsymbol{x}_3$ denotes the interior node. The mapping is illustrated in Figure 2 for a triangular element with one edge defined by five trimmed NURBS curves The different colours on the boundary denote the different trimmed NURBS.

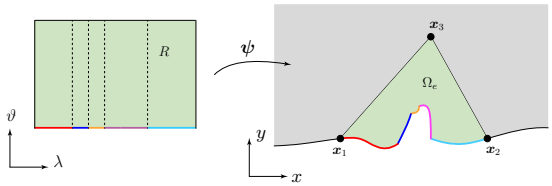

Figure 2: Illustration of the NEFEM mapping for a triangular element with one edge defined by five trimmed NURBS curves.

This mapping enables the definition of quadrature in the rectangle R, which decouples the complexity of the boundary curve (i.e., the rational definition of NURBS) with respect to the interior direction, ϑ. In practice a one dimensional Gaussian quadrature is used in each direction, as discussed in [16].

The mapping in three dimensions follows a similar principle. A mapping between a polygonal prism and the NEFEM tetrahedron is defined as

$$\begin{aligned} \boldsymbol{\Psi} &: R \longrightarrow \Omega_e \\ (\lambda, \kappa, \vartheta) &\longmapsto \boldsymbol{\Psi}(\lambda, \kappa, \vartheta) := (1-\vartheta)\boldsymbol{S}(\lambda, \kappa) + \vartheta \boldsymbol{x}_4, \end{aligned} \tag{2}$$

where $\boldsymbol{S}(\lambda, \kappa)$ is the parametrisation of the curved boundary, which might be given by several NURBS surfaces, and $\boldsymbol{x}_4$ denotes the interior node. The mapping is illustrated in Figure 3 for a tetrahedral element

with one face defined by three trimmed NURBS surfaces.

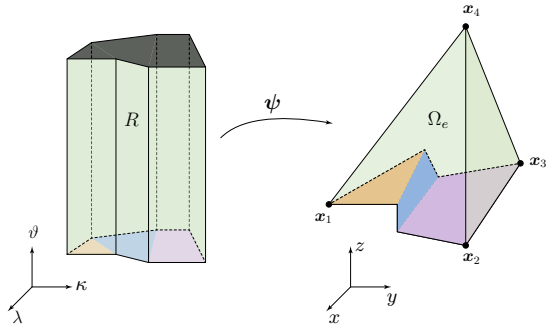

Figure 3: Illustration of the NEFEM mapping for a tetrahedral element with one face defined by three trimmed NURBS surfaces.

Once the numerical quadrature is devised, the polynomial shape functions are evaluated at the Gauss points. In NEFEM this is usually done directly in the physical space, with Cartesian coordinates. Since this is the information required by any finite element solver, this allows the incorporation of the NEFEM rationale into existing solvers in a transparent manner.

3. 2D NEFEM MESH GENERATION

This section summarises the strategy, originally presented in [15], for generating triangular NEFEM meshes. The goal is to motivate the need for a 3D NEFEM mesh generator, given the promising results obtained in 2D. The two key requirements for a NEFEM element are

1. the characteristic element size is strictly defined by the user and not related to the size of the geometric features in the model;
2. the edges that belong to the boundary are geometrically defined as a collection of trimmed NURBS curves.

The first requirement will ensure that the smallest element size is not dictated by the presence of small geometric features in the CAD model. However, this requirement implies that the traditional hierarchical approach used in many mesh generators cannot be followed.

3.1 Boundary discretisation

Given the B–rep of a two dimensional domain, a sampling set of points is first defined on the boundary curves [17]. This distribution of sampling points is locally refined near small geometric features to ensure that the discrepancy between the true geometric boundary and its piecewise linear approximation is kept within a specified tolerance.

Once the sampling points are created, the boundary curves are combined into *loops*. A loop is defined as an ordered set of boundary curves, where the shortest curve is assumed to be first.

Starting from the first curve in a loop, a vertex $\boldsymbol{x}_{i-1}$ is created and a candidate boundary vertex $\boldsymbol{x}_i$ is identified, such that the distance from $\boldsymbol{x}_{i-1}$ is given by the desired spacing. It is worth noting that distances are computed by traversing over the trimmed curves forming the loop.

The mesh front formed by vertices $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_i$ is checked to ensure that a valid triangle of the desired size will be formed when discretising the domain. To this end, the horizon of a vertex is defined to be the most distant point on the boundary that can be connected to the vertex using a straight line without intersecting the boundary. The intersection between the horizons of $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_i$, denoted by $\boldsymbol{P}$ is used to check the validity of the front. If the distance from the vertices to $\boldsymbol{P}$ is less than the desired spacing, the front is considered as valid. Otherwise the front is rejected and the candidate vertex $\boldsymbol{x}_i$ is moved towards the first vertex $\boldsymbol{x}_{i-1}$. Figure 4 illustrates the process followed to check the validity of a mesh front.

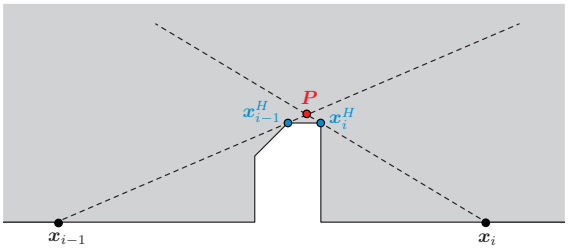

Figure 4: Illustration of the check performed to identify the validity of a mesh front. The super-index H is used to denote the horizon of a boundary vertex.

It is worth noting that the validity check described here is introduced to guarantee that the internal edges of the mesh will be straight. By modifying the parametrisation of the element, it is possible to relax, or completely eliminate, this validity check.

3.2 Domain discretisation

The generation of the triangular elements in the domain requires accounting for the exact B–rep, to ensure that internal edges do not intersect the boundary

of the domain. To this end, a modification to the advancing front method is proposed.

Given a mesh front, formed by two vertices $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, the intersection of the lines defining the horizon of the two vertices, $\boldsymbol{P}$, is defined, as shown in Figure 5.

The angle bisector of the two horizon lines and the point $\boldsymbol{P}$ define a line where the interior point of the triangle can be placed. The internal point is defined as $\boldsymbol{x}_3 := \boldsymbol{P} + \alpha \boldsymbol{v}$, where $\boldsymbol{v}$ is the direction given by the bisector of the horizons and α is selected such that the internal edges of the element do not exceed the desired element size, as given by the spacing function. The procedure is illustrated in Figure 5.

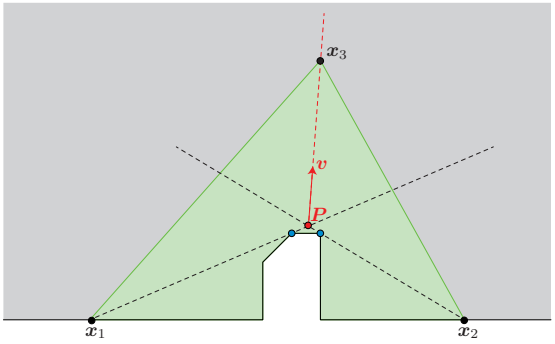

Figure 5: Illustration of modified advancing front method to build a valid NEFEM triangle.

It is worth noting that the proposed modified advancing front method can lead to elements that are not parametrised with the mapping of Equation (2). As the mapping is only used to define a numerical quadrature for the solver, a strategy to create a composite quadrature in the element by subdivision is devised for more complex cases such as the one in Figure 5. The subdivision is only used for numerical integration purposes, so no element or nodes are added to the NEFEM data structure. Full details on the automatic algorithm to build the quadratures on NEFEM elements can be found in [15].

3.3 High–order extension

One of the main attractive properties of NEFEM is the ability to decouple the concepts of geometric and functional approximation. The exact B–rep is always embedded in the geometric definition of NEFEM elements and the degree used for the approximation can be chosen according to the required accuracy. In this section, the strategy used to devise a high order nodal distribution in NEFEM elements is described. This is the only requirement to allow NEFEM to build arbitrary order functional approximations in the new type of element proposed.

Let us consider a triangular NEFEM element with vertices $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$, where at least one edge is described by a collection of trimmed NURBS curves. A high order nodal distribution of the desired degree can be easily generated in the auxiliary element given by the three vertices, by mapping a nodal distribution defined in a reference element. To illustrate the process, an equally spaced nodal distribution of degree five is shown in Figure 6.

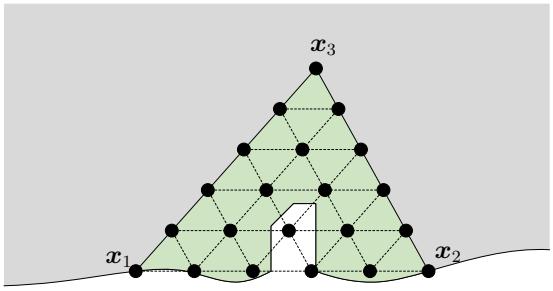

Figure 6: High order nodal distribution defined on the auxiliary triangle defined by the three vertices.

To obtain the valid high–order nodal distribution in the actual NEFEM element, an elasticity problem is formulated. The element is considered to be a linear elastic solid and Dirichlet boundary conditions are imposed in the whole boundary. For interior edges a homogeneous boundary condition is imposed, as internal edges are assumed straight. For edges on the boundary, a nodal distribution that satisfies the desired spacing of the high order nodes is generated over the NURBS curves. The difference between the position of the boundary nodes and the respective nodes in the distribution generated over the straight sided triangle is used as the Dirichlet boundary condition. Solving the corresponding elastic problem, the position of the internal nodes is computed. Figure 7 shows the resulting nodal distribution in the NEFEM element. It is well known that the linear elastic analogy may lead to non-valid elements if the deformation imposed on the boundary is large. To alleviate this issue, an incremental approach is considered when the boundary deformation is large.

3.4 Numerical example

To illustrate the benefits following from the use of NEFEM elements, a two dimensional electromagnetic scattering example is considered. The problem involves the wave scattering by a section of a satellite that contains many small geometric features. The geometry under consideration is shown in Figure 8.

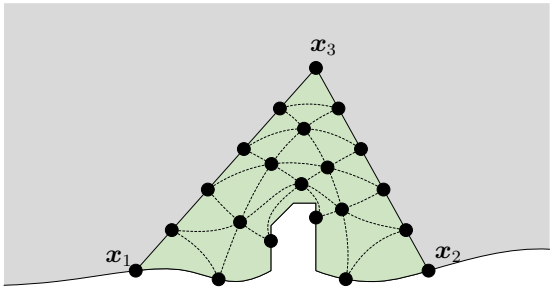

Figure 7: High order nodal distribution obtained after solving an elemental elasticity problem.

The boundary is made of 139 NURBS curves, where

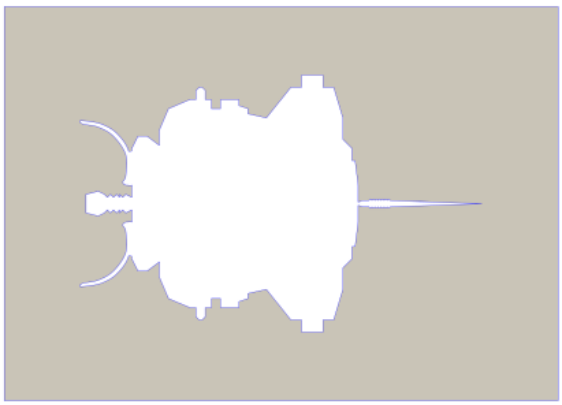

Figure 8: Geometry of a section of a satellite.

the shortest curve has length 0.01. For this problem a mesh of size 0.06 suffices to capture the scattered wave using cubic elements, meaning that the desired element size is six times larger than the shortest curve. It is worth noting that using a standard mesh generator the smallest element size would be less than or equal to 0.01 and this would introduce an important restriction in the time step when using an explicit time marching algorithm.

Using the proposed mesh generation approach, a NEFEM mesh is generated. The proposed elasticity analogy is then used to extend the mesh to high order. For this example a nodal distribution to build a functional approximation of degree four in each element is constructed. Two detailed views of the mesh are shown in Figure 9. The mesh clearly shows the ability of the method to produce elements with edges described as a collection of trimmed NURBS. The smallest edge on the boundary is near 0.04, which is four times larger than the smallest geometric feature.

The transverse magnetic scattered field, computed

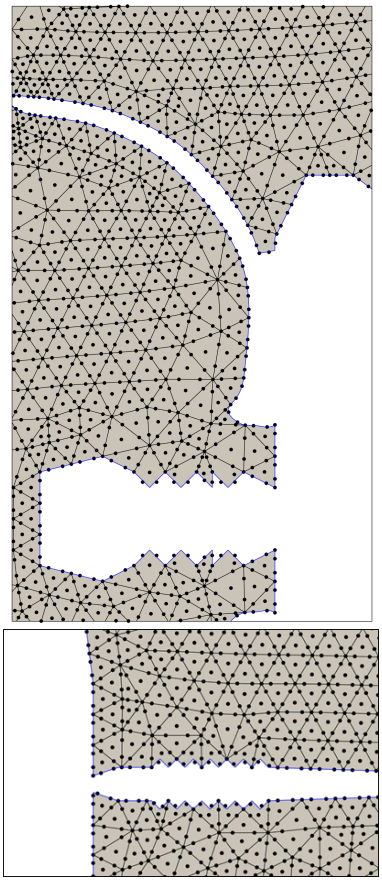

Figure 9: Two detailed views of the NEFEM mesh around the satellite of Figure 8 showing the high order nodal distribution.

using an in–house NEFEM high–order discontinuous Galerkin code, for solving the transient Maxwell's equations, is shown in Figure 10.

To assess the quality of the solution computed with NEFEM, the radar cross section is compared to a reference solution using low order elements with a mesh that resolves all the geometric features present in the CAD model. The NEFEM solution is in excellent agreement with the reference solution. In addition, the solution computed with NEFEM achieved convergence, to the time harmonic steady state, 140 times faster than the solution computed with low order elements.

This benefit is due to two reasons. First, the NEFEM mesh has less degrees of freedom, but, more importantly, the minimum element size is not dictated by any geometric features. Therefore, the computation with NEFEM can use a much larger time step, as the restriction induced by the explicit time marching is not as severe as that resulting from the use of meshes

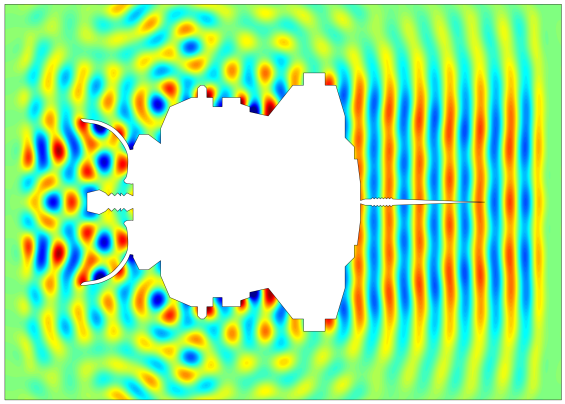

Figure 10: The transverse magnetic scattered field computed with the NEFEM mesh shown in Figure 9.

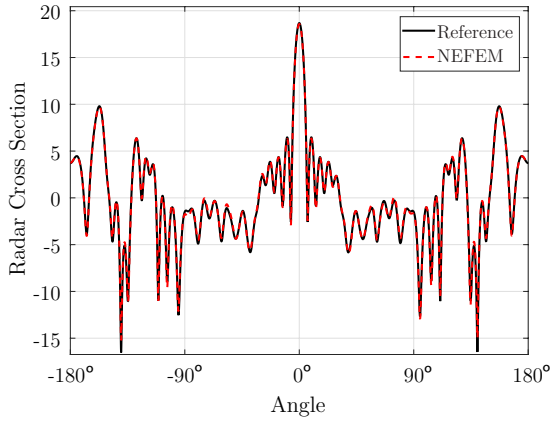

Figure 11: Comparison of the radar cross section computed using the NEFEM mesh shown in Figure 9 and a reference solution.

that resolve all the features.

4. 3D SURFACE NEFEM MESH GENERATION

This section presents the current effort made towards the generation of 3D NEFEM meshes, which is focused on the generation of valid NEFEM surface meshes.

In three dimensions, the sampling of the boundary surfaces is not considered to be an efficient approach because generating a dense set of sampling points to capture all the geometric features for complex geometric configurations could be expensive. Instead, this work considers an initial mesh generated using a conventional approach. The mesh generator targets a desired mesh spacing given by a background mesh and a set of point, line and surface sources. In addition, standard mesh refinement based on curvature can be activated. As many available mesh generators, a hierarchical approach is considered. Points of the CAD model are considered mesh nodes. Curves of the CAD model are discretised using edges and the surfaces are discretised using triangles. In this mesh, the characteristic element size, understood as the smallest height of the element, in some regions might be much smaller than the desired element size, due to the presence of small geometric curves and/or surfaces in the CAD B–Rep. All the B–Rep geometry models are considered to be watertight manifold surfaces.

As an example, Figure 12(a) shows the NURBS surfaces defining a geometry that consists of a flat plate intersected by two cylinders. Compared to the length and width, the thickness of the flat plate is much smaller. Besides, the diameter is much larger than the height for the flat cylinder, whereas the height is relatively larger than the diameter for the tall cylinder.

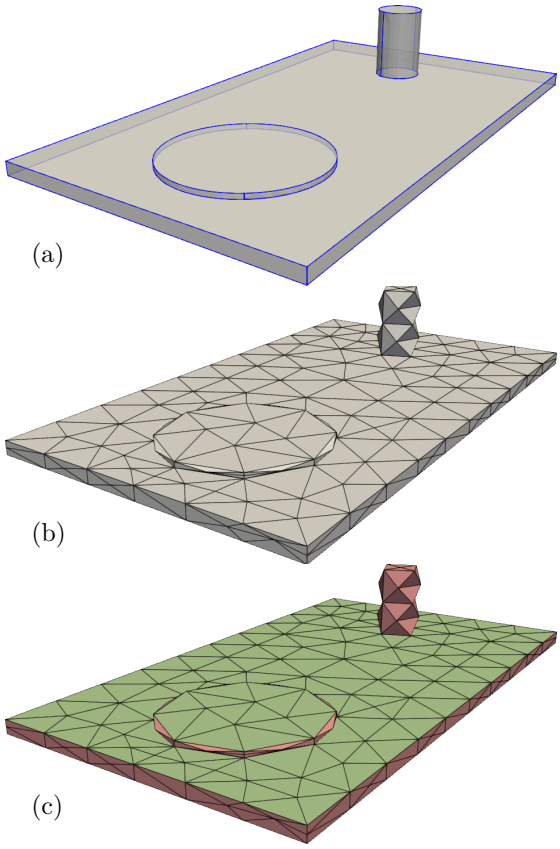

Figure 12: A flat plate intersected by two cylinders. (a) B–rep of the domain; (b) FEM surface mesh of the domain; (c) FEM surface mesh of the domain, where the faces in green and red denote the sets f and F respectively.

The surface mesh obtained using a standard mesh generator and with a uniform user defined spacing is shown in Figure 12(b). The presence of small faces, required to resolve the small features, can be clearly observed.

The set of boundary faces, $\mathcal{F}$, is subdivided into two disjoint sets, f and F, of faces with characteristic size smaller than and complying with, respectively, the user defined spacing function $h(\boldsymbol{x})$. Figure 12(c) illustrates the classification of the faces based on their characteristic element size. Faces shown in green form the set F of faces that satisfy the desired spacing, whereas faces shown in red form the set f of faces with characteristic size smaller than the user specified spacing function.

This section presents a strategy, named the local enhanced advancing front (LEAF) method, to successively transform the original FEM mesh into a triangular mesh in which elements span across different surfaces until the final set of small faces f is empty. The aim is to end up with a surface mesh in which the element size closely matches the original spacing defined by the user, irrespective of the presence of small geometric features. The LEAF method is applied to each vertex along each intersection curves in the model, typically following the natural order inherited from the initial mesh. It deals with the vertices involving only one intersection curve first. Those vertices belonging to more than one intersection curve are treated afterwards.

4.1 The LEAF method

Let us consider an intersection curve $\boldsymbol{C}$ such that a mesh node $\boldsymbol{P}_k = \boldsymbol{C}(\lambda_k)$ on $\boldsymbol{C}$ induces a closed triangle fan, $\mathcal{T}$, that intersects both sets f and F, as shown in Figure 13(a).

The nomenclature used in Figure 13 and in the remaining of this section is summarised in Figure 14.

The intersection curve $\boldsymbol{C}$ subdivides the closed triangle fan $\mathcal{T}$ into two triangle fans, namely $T := \{T_i\}_{i=1,\dots,N_T} \subset F$ with satisfactory elements and $t := \{t_j\}_{j=1,\dots,N_t} \subset f$ with small elements. Since the mesh is generated from a watertight manifold B–Rep geometry, the condition that $\mathcal{T}$ forming a valid closed triangle fan surrounding $\boldsymbol{P}_k$ requires that

$$N_T + N_t \geq 3. \tag{3}$$

This inequality is simply reflecting the fact that a closed triangle fan cannot be formed with less than three triangles because the angles of a triangles need to be strictly less than 180°.

The LEAF method aims at progressively transforming

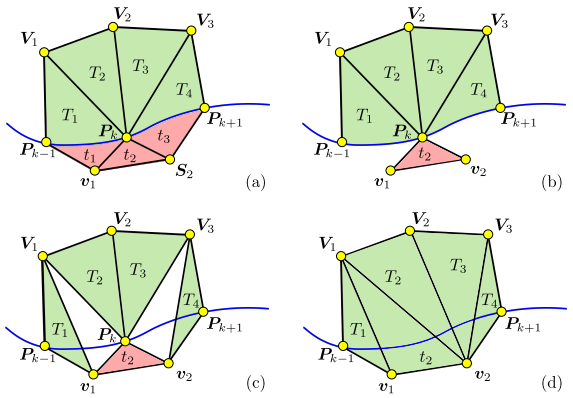

Figure 13: Illustration of the process of eliminating small triangles using the LEAF method. The figure shows (a) the initial patch, (b) and (c) two intermediate steps, and (d) the updated connectivity.

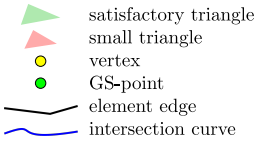

Figure 14: The legend of all illustrations used in this paper.

the original triangle fan $\mathcal{T}$ into a triangle strip, where the point $\boldsymbol{P}_k$ is removed and the resulting elements cross the intersection curve.

Initially, the parametric coordinate of the intersection curve, λ, is used to order the triangles in the triangle fans T and t, as well as the nodes that belong to the intersection curve. Next, the first and last triangles in t are removed, as shown in Figure 13(b). The connectivity of the first and last triangles in T is modified by substituting the node $\boldsymbol{P}_k$ by the first and last nodes of the fan t, respectively, as illustrated in Figure 13(c). Finally, the connectivity of the remaining triangles in T and t is updated. For a triangle $T_i \in T$, the node $\boldsymbol{P}_k$ is replaced by $\boldsymbol{v}_i$, if $\boldsymbol{v}_i$ exists, or by $\boldsymbol{v}_{N_t}$ otherwise. For a triangle $t_j \in t$, the node $\boldsymbol{P}_k$ is replaced by $\boldsymbol{V}_{j-1}$. This process leads to a triangle strip, as illustrated in Figure 13(d), with no elements in t and with all the elements crossing the intersection curve. The procedure is described in the listing shown in Algorithm 1. The procedure described here is applied to all the triangle fans, in every intersection curve that contain elements in t, until the set t is empty.

It is worth noting that the case considered here to illustrate the LEAF method considers a point $\boldsymbol{P}_k$ that belongs to one intersection curve. The algorithm also handles points that belong to more than one intersec-

Algorithm 1: Connectivity update routine for the LEAF method.

```
Sort T and {V_i} in parameter orientation;
Sort t and {v_j} in parameter orientation;
Remove t_1 and t_{N_t};
Replace vertex P_k with v_1 for T_1;
Replace vertex P_k with v_{N_t} for T_{N_T};
for i ← 2 to N_T − 1 do
    if i < N_t then
        Replace vertex P_k with v_i for T_i;
    else
        Replace vertex P_k with v_{N_t} for T_i;
    end if
end for
for j ← 2 to N_t − 1 do
    if j < N_T then
        Replace vertex P_k with V_{j−1} for t_j;
    else
        Replace vertex P_k with V_{N_T} for t_j;
    end if
end for
```

tion curve. Similarly, the example considered to illustrate the method considers triangular faces that cross one intersection curve, but the proposed strategy is capable of producing faces that cross multiple intersection curves or the same intersection curve more than once. It is important to emphasise that a NEFEM surface triangle is defined as a collection of trimmed NURBS surfaces, so it encapsulates all the geometric information of the B–rep, including the intersection curves that it crosses.

4.2 Enhanced edge description

The triangle strips created by the LEAF method remove all elements considered to be small. However, a triangle strip does not conform with the exact B–rep, because an edge connecting a node $\boldsymbol{V}_i$ and a node $\boldsymbol{v}_j$ crosses an intersection curve and cannot be defined by two points only. This section introduces the concept of *geometric support* points (GS–points) that will be used to define an enhanced edge, i.e. an edge that crosses an intersection curve.

It is important to emphasise that GS–points are purely used for an enhanced description of the edges and they are not extra degrees of freedom in the NEFEM solver.

For a given intersection curve, $\boldsymbol{C}$, the nodes on the intersection curve are considered, namely $\boldsymbol{P}_{k-1}$, $\boldsymbol{P}_k$ and $\boldsymbol{P}_{k+1}$, where it is worth noting that the point $\boldsymbol{P}_k$ is no longer present in the surface mesh connectivity. The parametric coordinate of the first and last points is denoted by λ_{k-1} and λ_{k+1} respectively.

It can be proved that the number of GS–points to be created, N_G, is equal to the number of edges in the triangle strip that cross the curve $\boldsymbol{C}$ and is given in terms of the number of triangles in the triangle fans T and t as

$$N_G = N_T + N_t - 3, \tag{4}$$

and the condition given by (3) ensures that $N_G \geq 0$.

A constructive proof of the equality (4) is as follows. Let us assume that the set of vertices $\{\boldsymbol{V}_i\}$ and $\{\boldsymbol{v}_j\}$ are ordered by following the direction of the intersection curve. To LEAF method requires the constructions of edges connecting these two sets of vertices that lead to a valid triangulation, i.e. without creating intersections. A simple strategy consists on selecting the last vertex in the second set, namely $\boldsymbol{v}_{N_v}$ and connecting it to all the vertices in the first set. This operation creates N_V edges. To complete the triangulation it is only necessary to join the vertices $\{\boldsymbol{v}_j\}_{j=1,\ldots,N_v-1}$ with $\boldsymbol{V}_1$, creating $N_v - 1$ edges. The total number of edges, equal to the number of GS–points N_G is then equal to $N_V + N_v - 1$. To conclude the proof it is only necessary to note that $N_T = N_V - 1$ and $N_t = N_v - 1$.

The GS–points are created by selecting a set of N_G parameters in the interval $[(\lambda_{k-1}+\lambda_k)/2, (\lambda_k+\lambda_{k+1})/2]$, as shown in Figure 15(a), and each intersection point is associated to one of the edges crossing the intersection curve. The NEFEM triangles are defined by enhancing the definition of the edges crossing the intersection curves, as illustrated in Figure 15(b).

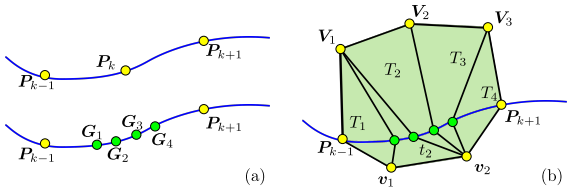

Figure 15: Illustration of (a) creation of GS–points, and (b) association to NEFEM triangle edges.

It is worth remarking that the progressive use of the LEAF method is only necessary to update the elements adjacent to the intersection curves that are considered small. The resulting mesh inherits the majority of the elements from the original mesh, generated using a conventional low order mesh generator.

A detailed view of the NEFEM mesh obtained after applying the LEAF method to the conventional mesh of Figure 12(b) is shown in Figure 16.

The resulting mesh clearly exhibits the ability of the proposed method to remove all small elements and produce a NEFEM mesh where elements with enhanced edges cross the intersection curves of the B–

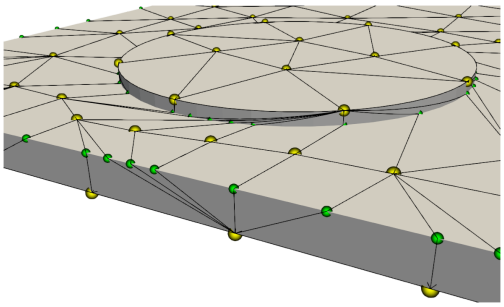

Figure 16: Detailed view of the NEFEM surface mesh for the flat plate intersected by two cylinders. Element vertices and GS–points are shown with dots in yellow and green, respectively.

rep. However, it can be clearly observed that the quality of the resulting NEFEM elements is directly linked to the precise definition of the parametric coordinates to create the GS–points. The next section presents a simple procedure to improve the quality of the NEFEM elements.

4.3 Mesh cosmetics

The GS–points used to define the enhanced edges can be distributed over the intersection curves to improve the quality of the NEFEM elements. If they are naively equi–distributed in the parametric interval $[(\lambda_{k-1}+\lambda_k)/2, (\lambda_k+\lambda_{k+1})/2]$ the resulting mesh might have largely distorted elements, as shown in Figure 16.

To adjust the GS–points $\boldsymbol{G}_l$ on an intersection curve $\boldsymbol{C}$, a spacing function $g(\lambda)$ is defined in the parametric space. A natural option is to obtain $g(\lambda)$ by interpolating the spacing field $h(\boldsymbol{x})$ of the initial FEM mesh and mapping it back to the parametric space. Then, the spacing between $\boldsymbol{G}_l$ and $\boldsymbol{G}_{l+1}$ is set to follow a distribution specified by $g(\lambda_l)$. In practice, given a GS–point $\boldsymbol{G}_l = \boldsymbol{C}(\lambda_l)$, the parameter for the next GS–point $\boldsymbol{G}_{l+1}$ is such that $G(\lambda_{l+1}) = 0$, where

$$G(\lambda) := g(\lambda_l) - \int_{\lambda_l}^{\lambda} \|\boldsymbol{C}'(s)\| \, \mathrm{d}s. \qquad (5)$$

This inspired by the strategy used to define high quality nodal distributions in [18]. The non–linear scalar equation (5) is solved, for $l = 1, \ldots, N_G - 1$, using a bisection method with an adaptive Gaussian quadrature for the numerical evaluation of the integral. For the example shown in Figure 16, the resulting NEFEM mesh after the GS–points are adjusted is shown in Figure 17(a).

In addition, standard cosmetic processes such as diagonal swapping and smoothing can be performed with routines similar to those used in conventional mesh

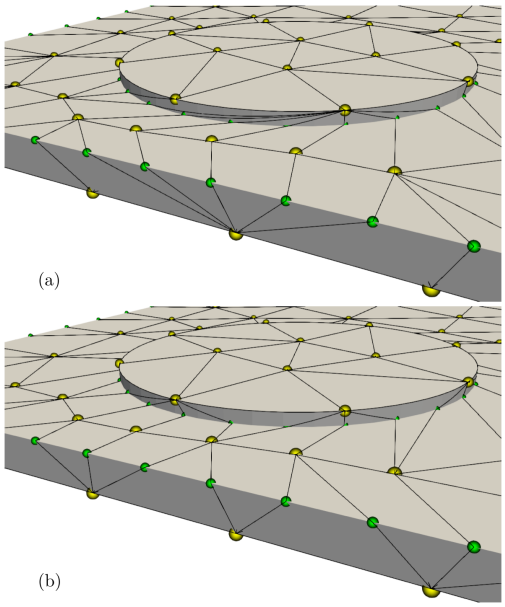

Figure 17: Local views of NEFEM surface mesh for the flat plate intersected by two cylinders. (a) intermediate result after GS–point adjustments, (b) final result after diagonal swapping. Element vertices and GS–points are shown with dots in yellow and green,respectively.

generators. The diagonal swapping strategy implemented locally modifies the connectivity of the NEFEM mesh so that the angles of the enhanced triangles are maximised. It is worth noting that the angles of an enhanced triangle must be computed using the tangent to the enhanced edges by employing the NURBS description of these edges. The length of the resulting enhanced edges is also checked to ensure that the resulting characteristic size of the elements do not substantially differ from the user defined spacing. It is worth emphasising that the length of an enhanced edge must be computed using the exact definition of the boundary and not just the distance between the vertices. As an example, the resulting surface, after applying diagonal swapping, is shown in Figure 17(b), and this clearly illustrates the improved quality obtained.

5. EXAMPLES

This section presents three numerical examples to illustrate the strategy described in the previous section to generate NEFEM surface meshes in three dimensions.

5.1 A flat plate with two cylinders

The performance of this strategy, for generating surface NEFEM meshes, is analysed here in detail for the CAD model shown in Figure 12(a). The geometric data for this model is listed in Table 1.

Table 1: Geometric data of the flat plate with cylinders model.

Number of NURBS Surfaces	12
Dimension of the plate	$2 \times 1.143 \times 0.05$
Diameter of Cylinder 1	0.58
Height of Cylinder 1	0.019
Diameter of Cylinder 2	0.183
Height of Cylinder 2	0.281

The desired element size, equal to 0.2, is substantially larger than many of the features present in the model. When a mesh of uniform spacing is generated with a standard mesh generator, the surface mesh shown in Figure 12(b) is obtained. The mesh contains a number of elements that are substantially smaller than the size requested by the user, due to the presence of small geometric features. For this example, the mesh data are summarised in Table 2.

Table 2: Surface mesh summary of the flat plate with cylinders model.

Surface mesh	FEM	NEFEM
Number of elements	476	302
Number of nodes	240	153
Minimum edge length	0.0095	0.0944
Average edge length	0.1617	0.2043
Desired mesh size	0.2	0.2

After applying the proposed method proposed, the NEFEM mesh obtained is shown in Figure 18.

A detailed view of the mesh is shown in Figure 18(b). The surfaces are rendered in different colours, the vertices are represented with yellow dots and the GS–points are shown in green. It can be observed that the location information for intersecting points, among three (or more) surfaces, is preserved with either a vertex or a GS–point.

The NEFEM faces appear as folded triangles, with the folding line being the intersection curve between the surfaces that they cross. For the curved surfaces, the elements retain the exact B–rep. Therefore, NEFEM faces are a collection of trimmed NURBS surfaces. It is worth noting that the proposed algorithms defines

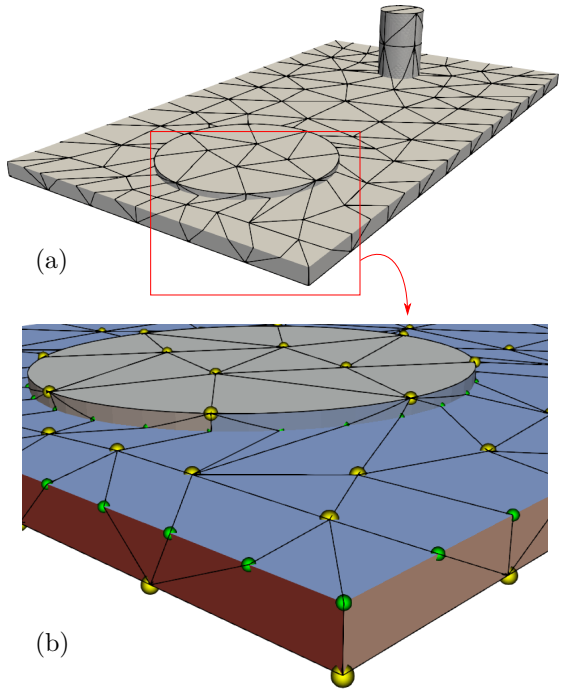

Figure 18: NEFEM surface mesh for the flat plate intersected by two cylinders (shown in Figure 12(a)). (a) Global view, (b) Detailed view with individual surfaces coloured differently. Element vertices and GS–points are shown with dots in yellow and green, respectively.

faces with only $\mathcal{C}^0$ continuity across the intersection curves. This is feasible due to the incorporation of the exact B–rep into the solver, as described in Section 2.2.

The total number of triangular faces and nodes in the NEFEM surface mesh is 302 and 153 respectively. More importantly, the minimum element edge length is 0.0944, which is approximately one order of magnitude higher than that in the original mesh. The average element edge length is increased from 0.1617 to 0.2043, which is very close to the desired value of 0.2. It is worth noting that some interior triangles on the top plane have a slightly shorter edge, due to the fact that they are inherited from the initial FEM mesh and are never updated.

It can be observed that the element size in the NEFEM mesh is totally independent of the thickness of the plate or of the flat cylinder and the size of the NEFEM elements is nearly uniform. This demonstrates the ability of the proposed method to create a boundary discretisation with desired element size, regardless of the small geometric features. To improve the quantification of the benefit of the proposed approach, Figure 19 shows histograms of the normalised edge lengths for both meshes. The histograms clearly show

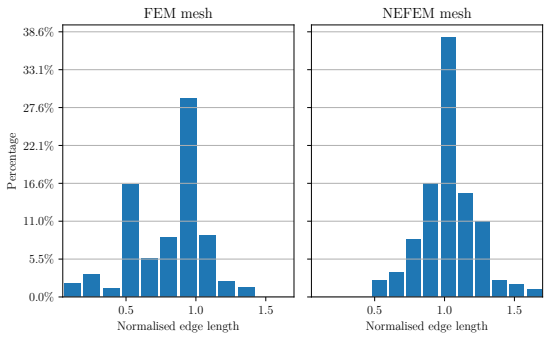

Figure 19: Histograms of normalised edge lengths of FEM and NEFEM meshes for the flat plate with cylinders.

an increase in the number of elements where the normalised size is close to one in the NEFEM mesh. It is worth noting that some elements with a normalised edge length below 0.5 remain in the NEFEM mesh, as their size is above what is considered a small element. The current implementation allows the selection of a threshold value, when classifying the original triangular mesh into the set of small faces and the set of large faces.

5.2 A turbine engine fairing

The next example involves the generation of a surface NEFEM mesh for a turbine engine fairing. The CAD model, shown in Figure 20(a) and (b), contains four large surfaces, representing the outer and inner shells, and six thin and small surfaces, connecting the shells at the leading and trailing edges. For this model, all surfaces are curved NURBS surfaces, so that all elements in the surface mesh must be NEFEM elements to ensure that the exact boundary representation of the domain is made persistent during the simulation stage. The major dimensions of the model are listed in Table 3. In addition, the curve lengths in the CAD model vary significantly, e.g. the length of the longest curve is of more than 50 times that of the shortest curve. Due to this variation, the initial FEM mesh contains a number of elements considered too small, as depicted in Figure 20(c).

Table 3: Geometric data of the fairing model.

Number of NURBS Surfaces	10
Axial length	800
Diameter of inlet	520
Diameter of outlet	103.5

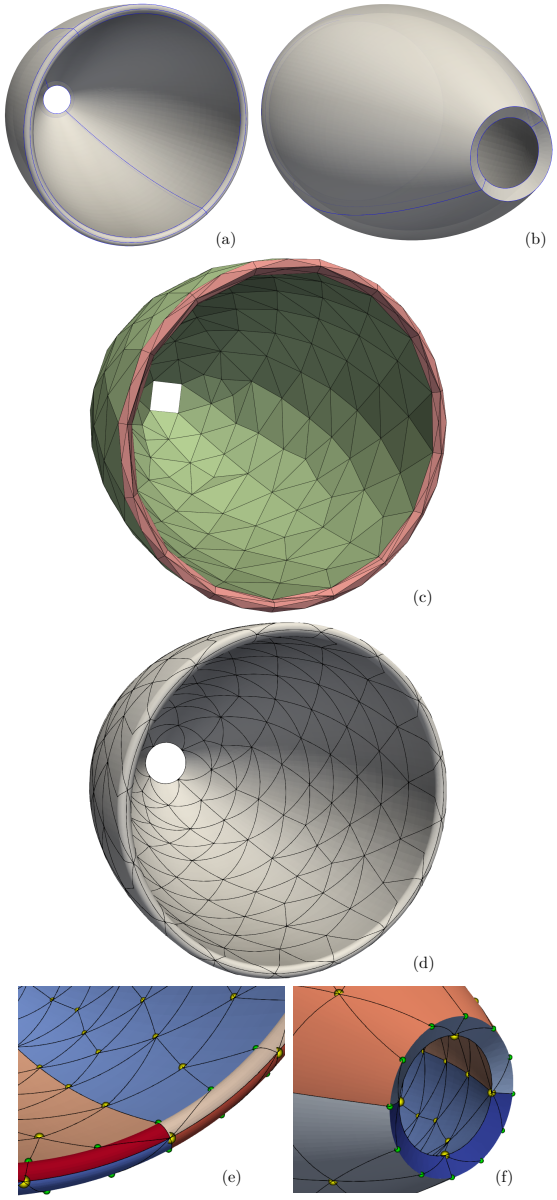

Figure 20: The turbine engine fairing model. (a) CAD geometry front view; (b) CAD geometry back view; (c) FEM mesh with small elements in red; (d) NEFEM mesh. Detailed views of the NEFEM mesh with individual surfaces are coloured differently for (e) leading edge and (f) trailing edge. Element vertices and geometric support points are shown with dots in yellow and green, respectively.

This figure shows the elements considered too small in red, whereas the elements in green have the desired size. It is worth noting that due to the nature of this geometry, the elements in red are not only too small, but they are also highly stretched. This will induce

challenges for explicit solvers due to the restriction on the time step and also for implicit solvers due to the increase in the condition number induced by the stretched elements and the large discrepancy of elements sizes in the mesh.

The generated NEFEM surface mesh, obtained using the proposed method, is shown in Figure 20(d). The desired mesh size for this model is set as 100.0, which is much larger than the length of the shortest intersection curve. As a result, the minimum element edge length of the FEM mesh is compromised. The mesh data are summarised in Table 4. Obviously the NEFEM mesh better respects the specified size.

Detailed views of the NEFEM mesh are shown in Figure 20(e) and (f) with surfaces coloured individually and vertices and GS–points rendered. The ability of the proposed method to produce triangular faces that cross the intersection curves is clearly observed.

Table 4: Surface mesh summary of the fairing model.

Surface mesh	FEM	NEFEM
Number of elements	343	269
Number of nodes	686	538
Minimum edge length	7.47	48.52
Average edge length	87.69	100.02
Desired mesh size	100	100

It is worth noting that using the virtual topology paradigm [4], the surfaces in Figure 20(e) would be merged as there is a smooth transition of the normal between the different surfaces. However, due to the very localised change of curvature, traditional high-order elements would only approximate the exact B–rep. It is also important to emphasise that the boundary representation provided by traditional isoparametric elements is only $\mathcal{C}^0$, so the smoothness of the B–rep is not preserved in the solver [10]. This is relevant in different applications. For instance, in stress analysis, a $\mathcal{C}^0$ continuity might lead to a stress concentration. In fluid mechanics, corners are known to introduce non–physical entropy. In electromagnetics, corners can lead to strong singularities of the electromagnetic field. Therefore, the persistence of the true CAD model in the solver, via the NEFEM approach, is expected to bring several advantages, not only in terms of efficiency but also in terms of reliability of the results.

The histograms of the normalised edge length for both the original FEM mesh and the resulting NEFEM mesh are depicted in Figure 21. The results clearly show the ability of the proposed method to produce a mesh in which the majority of elements have an edge length very close to the desired size.

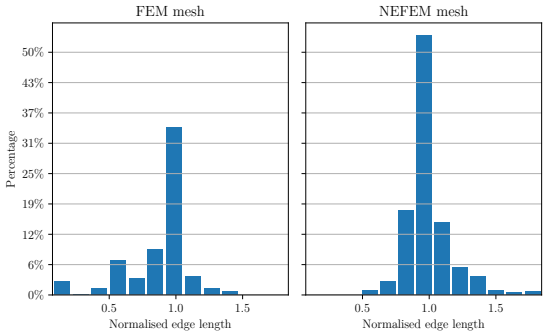

Figure 21: Histograms of normalised edge lengths of FEM and NEFEM meshes for the turbine engine fairing.

5.3 A wing model

The last examples considers the generation of a NEFEM surface mesh of a wing with a blunt trailing edge. The objective of this example is to show the ability of the method to handle a desired non-uniform mesh spacing. The NURBS surfaces defining the wing are represented in Figure 22(a), and the main dimensions of the model are listed in Table 5. For this model, a non-uniform spacing function is specified using two line sources at the leading and trailing edges with a stretching ratio equal to 5. Despite the refinement introduced by the line sources, the spacing defined by the user is greater than the length of the shortest intersection curve. The resulting FEM mesh, illustrated in Figure 22(b), yields small elements at the blunt trailing edge and the wing tip. Small elements are shown in red, whereas green elements have the desired size.

Table 5: Geometric data of the wing model.

Number of NURBS Surfaces	4
Span	1200
Length of chord line at root	777.8
Length of chord line at tip	423.3
Maximum aerofoil thickness at root	78.2
Maximum aerofoil thickness at tip	43.9
Height of blunt trailing edge	7.23

The generated NEFEM surface mesh is shown in Figure 22(c). It can observed that NEFEM faces of the desired size contain parts of the trimmed surface that defines the blunt trailing edge. It can also be seen that NEFEM elements include parts of the trimmed

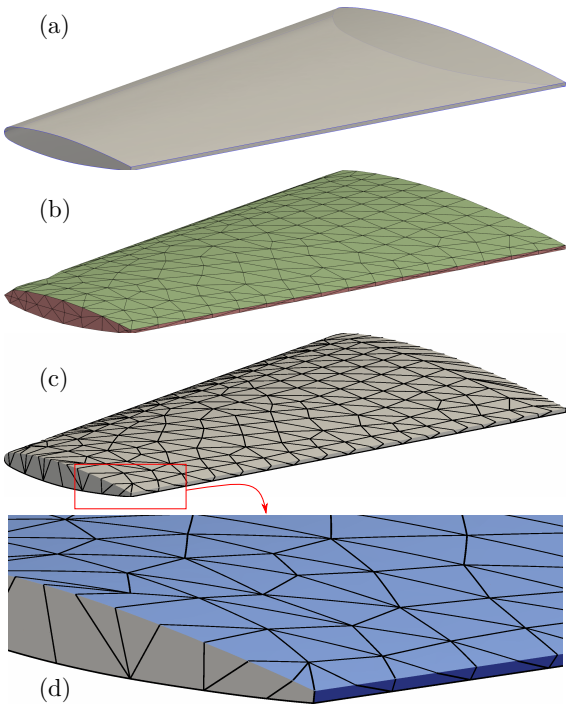

Figure 22: The wing model with a blunt trailing edge. (a) CAD geometry; (b) FEM mesh with small elements in red; (c) NEFEM mesh; (d) Zoomed view of the NEFEM mesh.

surface that defines the flat tip of the wing.

In this example, the position of the GS–points is computed by using the spacing function induced by the line sources initially used to generate the FEM mesh, as explained in Section 4.3. This results in a non-uniform distribution of the GS–points. A detailed local view near the leading edge is presented in Figure 22(d).

To conclude, the histograms of the normalised edge length for both the original FEM mesh and the resulting NEFEM mesh are depicted in Figure 23. It is worth emphasising that the non-uniform spacing function is used to normalise the element edge lengths. As in the previous examples, the proposed technique is shown to be effective at producing NEFEM elements with size closer to the desired spacing, contrary to the original FEM mesh.

6. CONCLUDING REMARKS

A method to produce surface meshes for the NURBS–enhanced finite element method (NEFEM) has been presented. The method is capable of generating surface meshes where the exact boundary representation, given by the non-uniform rational B–splines (NURBS) of the CAD model, is embedded in the geometric definition of NEFEM elements. The size of the elements in NEFEM meshes is not restricted by the small geometric features present in the CAD model. This strategy completely removes the time consuming process of de–featuring complex CAD models and, at the same time, removes the uncertainty generated by the de–featuring process.

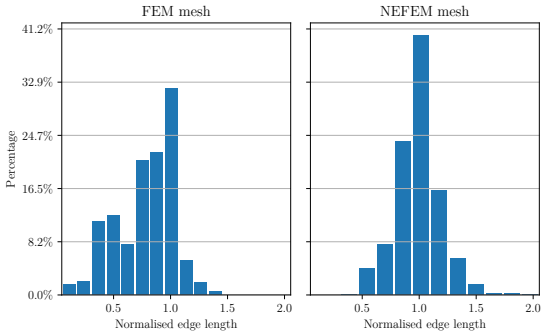

Figure 23: Histograms of normalised edge lengths of FEM and NEFEM meshes for the wing model.

The proposed strategy starts by generating an initial surface mesh using a standard low-order mesh generator. In regions with elements much smaller than the desired spacing, a technique called the locally enhanced advancing front (LEAF) method is proposed. By means of local modifications of the original mesh, the LEAF method removes all elements considered too small and introduces NEFEM elements, with faces defined as a collection of trimmed NURBS surfaces. The resulting NEFEM surface meshes contain elements that span across multiple surfaces in order to ensure that the spacing closely matches the user–defined spacing function.

Numerical examples are used to demonstrate the applicability and potential of the proposed technique. The examples involve geometries where the CAD model contain very small edges, such as a wing with a blunt trailing edge. NEFEM surface meshes are shown to provide a spacing closely matching the user–defined spacing function, even when the CAD model contains small features.

The application of the proposed approach to more complex geometric configurations requires a number of developments. First it is necessary to implement recursively the LEAF method to enable the creation of NEFEM triangles that cross multiple surfaces. In addition the surface mesh generation strategy needs to account for the future creation of the volume mesh. Despite it is possible to recursively apply the LEAF method to produce NEFEM surface meshes of more complex configurations, this might lead to surface

meshes that cannot be used to create a valid volume mesh. Therefore, when enhancing the surface triangles, a strategy to ensure that this triangles can be used to create valid tetrahedra must be devised.

Future work will focus on the improvement of the mesh cosmetics approach, the definition of new quality metrics for NEFEM surface meshes and the extension to high-order.

References

[1] Park M.A., Kleb W.L., Jones W.T., Krakos J.A., Michal T.R., Loseille A., Haimes R., Dannenhoffer J. "Geometry Modeling for Unstructured Mesh Adaptation." *AIAA Aviation 2019 Forum*, p. 2946. 2019

[2] Robinson T.T., Armstrong C.G., Fairey R. "Automated mixed dimensional modelling from 2D and 3D CAD models." *Finite Elements in Analysis and Design*, vol. 47, no. 2, 151–165, 2011

[3] Niu Z., Martin R.R., Sabin M., Langbein F.C., Bucklow H. "Applying database optimization technologies to feature recognition in CAD." *Computer-aided design and applications*, vol. 12, no. 3, 373–382, 2015

[4] 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

[5] 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

[6] 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

[7] 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

[8] Sevilla R. "HDG-NEFEM for two dimensional linear elasticity." *Computers & Structures*, vol. 220, 69–80, 2019

[9] Taylor N.J., Haimes R. "Geometry Modelling: Underlying Concepts and Requirements for Computational Simulation." *2018 Fluid Dynamics Conference*, p. 3402. 2018

[10] 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

[11] 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

[12] 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

[13] 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

[14] 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

[15] 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

[16] Sevilla R., Fernández-Méndez S. "Numerical integration over 2D NURBS shaped domains with applications to NURBS-enhanced FEM." *Finite Elements in Analysis and Design*, vol. 47, no. 10, 1209–1220, 2011

[17] Peiró J. *Surface grid generation*, chap. 19. CRC Press, 1999

[18] Xie Z.Q., Sevilla R., Hassan O., Morgan K. "The generation of arbitrary order curved meshes for 3D finite element analysis." *Computational Mechanics*, vol. 51, no. 3, 361–374, 2013