

Simultaneous untangling and smoothing of quadrilateral and hexahedral meshes using an object-oriented framework

**E. Ruiz-Gironés¹, X. Roca², J. Sarrate¹, R. Montenegro³,
J. M. Escobar³**

¹**Laboratori de Càlcul Numèric (LaCàN),
Departament de Matemàtica Aplicada III,
Universitat Politècnica de Catalunya, Barcelona, Spain.**

²**Aerospace Computational Design Laboratory,
Department of Aeronautics and Astronautics,
Massachusetts Institute of Technology, Cambridge, USA.**

³**Instituto Universitario de Sistemas Inteligentes y Aplicaciones
Numéricas en Ingeniería (SIANI),
Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran
Canaria, Spain.**

Abstract

In this work, we present a simultaneous untangling and smoothing technique for quadrilateral and hexahedral meshes. The algorithm iteratively improves a quadrilateral or hexahedral mesh by minimizing an objective function defined in terms of a regularized algebraic distortion measure of the elements. We propose several techniques to improve the robustness and the computational efficiency of the optimization algorithm. In addition, we have adopted an object-oriented paradigm to create a common framework to smooth meshes composed by any type of elements, and using different minimization techniques. Finally, we present several examples to show that the proposed technique obtains valid meshes composed by high-quality quadrilaterals and hexahedra, even when the initial meshes contain a large number of tangled elements.

Keywords: Mesh generation; hexahedral mesh; mesh smoothing; mesh untangling; object-oriented framework.

1 Introduction

The accuracy and performance of a finite element analysis depend on the quality of the mesh on which the spatial domain has been discretized. Most three-dimensional meshes are composed by tetrahedral elements as they can be automatically generated to discretize an arbitrary domain by rapid and mature algorithms. While more challenging to mesh, hexahedral elements have several advantages over tetrahedral meshes both in structural and in fluid mechanics. Unfortunately, ready access to these advantages is hampered by the significant challenges regarding the generation of hexahedral meshes. Several algorithms have been developed to generate hexahedral meshes. On the one hand, specific algorithms have been developed for specific types of geometries. For instance, the submapping method [1, 2] is devoted to blocky geometries or the sweep method [3, 4, 5, 6] is designed for extrusion geometries. On the other hand, several methods have been developed to mesh any geometry with an unstructured hexahedral mesh. For instance, 1. dual methods such as whisker weaving [7] or the local dual contributions [8]; 2. primal methods that directly generates the hexahedral mesh such as grid based methods [9, 10, 11], advancing front methods [12, 13], or the receding front method [14], and 3. primal methods that transform a tetrahedral mesh into a hexahedral mesh [15, 16]. However, fully automatic unstructured hexahedral mesh generation algorithms are still not available and, at this stage, no automated technique exists which compares to tetrahedral methods such as Delaunay or advancing front techniques. Moreover, the hexahedral meshes obtained by these algorithms may contain poorly shaped or even inverted elements that can affect and invalidate any subsequent analysis. For this reason, it is of the major importance to apply a mesh optimization algorithm after the mesh generation process is finished to untangle the inverted elements and to increase the quality of the whole mesh.

In this work, we present a technique that iteratively untangles and smooths a given quadrilateral or hexahedral mesh by minimizing a regularized objective function defined in terms of an algebraic distortion measure [17, 18, 19]. This objective function allows simultaneously untangling and smoothing the initial mesh. The distortion of quadrilateral and hexahedral elements is defined in terms of their decomposition in triangular and tetrahedral sub-elements, respectively [20, 21]. Thus, the optimization methods for quadrilateral and hexahedral meshes inherit the properties of the methods for triangular and tetrahedral meshes [22, 23, 24, 25, 26]. We have implemented two optimization methods to obtain the final configuration of nodes. Specifically, we have considered the steepest-descent and the Newton-Raphson line-search methods. It is important to point out that the proposed method optimizes the quality of quadrilateral and hexahedral meshes by a local node relocation process. That is, without modifying the mesh connectivity.

This paper is based in the work presented in [27]. However, in the current paper we include several new contributions that clearly improve the preliminary version of the simultaneous untangling and smoothing procedure for quadrilateral and hexahedral meshes. We regularize the objective function that measures the distortion of the quadrilateral and hexahedral elements in order to penalize the elements with lower

quality. Moreover, we incorporate three features to increase the robustness and the computational efficiency of the optimization procedure. First, for each free node, we propose to translate to the origin, and scale to a reference size the local sub-mesh. Second, when applying the minimization algorithm on each free node, we do not iterate until convergence. Instead, we perform only one iteration. Third, we reduce the cost of evaluating the objective function and its derivatives by only considering the sub-elements adjacent to the free node. In addition, we have implemented a new optimization algorithm based on a line-search approach, and we have included the steepest descent and the Newton-Raphson methods. Note that these methods require the use of the first and second derivatives of the objective function. Therefore, in this paper we include a new appendix where we detail their analytical expressions. To implement the proposed untangling and smoothing algorithm, in this work we have adopted an object-oriented paradigm to create a common framework to smooth meshes composed by any kind of elements. Furthermore, this new framework also allows using a generic line-search minimization method to optimize the objective function. Finally, new examples with real geometries are used to show the robustness and strengths of the proposed algorithm.

The paper is organized as follows. In Section 2, we survey the literature on mesh smoothing techniques. In Section 3, we review the concept of an algebraic shape quality metric for triangular and tetrahedral elements. Then, in Section 4, we extend to quadrilateral and hexahedral elements the distortion and quality measures previously reviewed. Section 5 is devoted to detail the simultaneous untangling and smoothing technique for quadrilateral and hexahedral meshes. In Section 6, we present an object oriented framework to implement the different objective functions and minimization methods. In Section 7, we present several examples illustrating that the proposed technique can be used to obtain valid meshes composed by high-quality quadrilaterals and hexahedra. Finally, the conclusions and the future work are summarized in Section 8.

2 Related work

Mesh smoothing is a key process for keeping mesh shape regularity and avoiding a costly re-meshing process. Several smoothing algorithms have been developed during the last decades to improve the quality of a given mesh. Most of them are based on geometrical or numerical reasoning [28, 29]. Although they are fast from the computational point of view, they have some drawbacks. For instance, they can move nodes outside of a domain near non-convex boundaries. In addition, they are not designed to optimize any quality measure. To overcome these drawbacks, Knupp [20, 21, 26] introduced a family of quality measures placed within an algebraic framework that have been intensively used during the last decade. These quality measures facilitate the development of smoothing methods based on its optimization [22, 23, 24, 25, 26].

The objective function of the minimization algorithm for a given node is based on a measure of the quality of adjacent elements. The minimization process can be

performed globally [26] or based on a local approach [25]. In the global approach, a non-linear equation has to be minimized for all the nodes at the same time. In the local approach, all the free nodes are smoothed sequentially until the objective function is optimized. Usually, the used objective functions are appropriate to improve the quality of a valid mesh, but they do not work properly when there are inverted elements. This is because the objective functions present singularities (barriers) when the mesh contains inverted elements. Thus, the classical smoothing approach involves two steps [26, 25]. The first step untangles the inverted elements, and second one improves the quality of the mesh. Since they are different processes, the objective function for each step is different.

To avoid the need of defining two objective functions for the untangling and smoothing process, references [17, 18, 19] propose a regularization of the objective function for tetrahedral meshes. Using the regularized objective function, the authors obtain a simultaneous untangling and smoothing process. This technique has been considered in different finite element applications. For example, it has been the key of the mecano method [30], its recent application for isogeometric solid modeling [31] and high-order mesh generation [32, 33, 34, 35].

In this work, we analyze the extension of the simultaneous untangling and smoothing technique to quadrilaterals and hexahedral elements. In addition, we improve the robustness and the performance of the algorithm using several techniques discussed in this paper. Finally, we have adopted an object-oriented paradigm to implement the simultaneous untangling and smoothing technique in order to create a common framework to smooth meshes composed by any type of elements. This framework also allows to use different line-search minimization algorithms to optimize the objective function.

3 Algebraic quality measures for triangles and tetrahedra

Knupp [20], introduced a family of distortion and quality measures placed within an algebraic framework. In this framework, the distortion measure of a simplex (triangle or tetrahedron) is measured in terms of the deviation from an *ideal* simplex. That is, the element that represents the desired shape to achieve. In this section, we summarize the formulation of the algebraic framework for triangles and tetrahedra. Although it is already presented in [27], we include it here for completeness.

Let t denote a triangle in the physical space, and t_I denote the *ideal triangle*. According to [20], to measure the deviation of the physical triangle from the ideal one, we want to determine the affine mapping (see Figure 1):

$$\begin{aligned} \mathbf{f}_S : t_I &\longrightarrow t \\ \tilde{\mathbf{x}} &\longmapsto \mathbf{x} = \mathbf{S}\tilde{\mathbf{x}} + \mathbf{v}, \end{aligned} \tag{1}$$

where $\tilde{\mathbf{x}}$ are the coordinates of the ideal element.

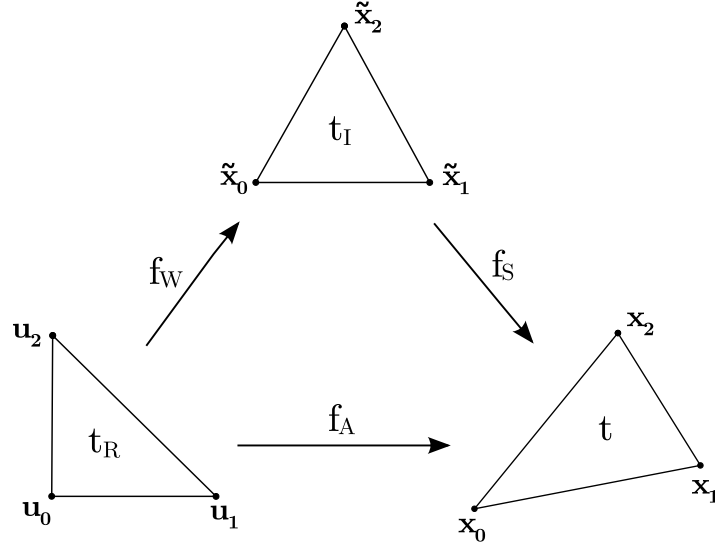


Figure 1: Affine mappings for triangular elements.

To this end, we consider a *reference triangle*, t_R , delimited by vertices $\mathbf{u}_0 = (0, 0)^T$, $\mathbf{u}_1 = (1, 0)^T$ and $\mathbf{u}_2 = (0, 1)^T$ in the parametric space. If the ideal element, t_I , is delimited by the vertices: $\tilde{\mathbf{x}}_0 = (\tilde{x}_0, \tilde{y}_0)^T$, $\tilde{\mathbf{x}}_1 = (\tilde{x}_1, \tilde{y}_1)^T$, and $\tilde{\mathbf{x}}_2 = (\tilde{x}_2, \tilde{y}_2)^T$, the affine mapping that transforms the reference element to the ideal one can be written as (see Figure 1):

$$\begin{aligned} \mathbf{f}_W : t_R &\longrightarrow t_I \\ \mathbf{u} &\longmapsto \tilde{\mathbf{x}} = \mathbf{W}\mathbf{u} + \tilde{\mathbf{x}}_0, \end{aligned} \quad (2)$$

where

$$\mathbf{W} = \begin{pmatrix} \tilde{x}_1 - \tilde{x}_0 & \tilde{x}_2 - \tilde{x}_0 \\ \tilde{y}_1 - \tilde{y}_0 & \tilde{y}_2 - \tilde{y}_0 \end{pmatrix}.$$

Similarly, if the triangle in the physical space is delimited by nodes $\mathbf{x}_0 = (x_0, y_0)^T$, $\mathbf{x}_1 = (x_1, y_1)^T$ and $\mathbf{x}_2 = (x_2, y_2)^T$, the affine mapping that maps the reference element, t_R , into the physical triangle, t , can be written as (see Figure 1):

$$\begin{aligned} \mathbf{f}_A : t_R &\longrightarrow t \\ \mathbf{u} &\longmapsto \mathbf{x} = \mathbf{A}\mathbf{u} + \mathbf{x}_0, \end{aligned} \quad (3)$$

where

$$\mathbf{A} = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix}.$$

The desired affine mapping, Equation (1), that maps the ideal triangle to the physical triangle can be determined by composing the inverse of \mathbf{f}_W , Equation (2), with function \mathbf{f}_A , Equation (3). That is,

$$\begin{aligned} \mathbf{f}_S = \mathbf{f}_A \circ \mathbf{f}_W^{-1} : t_I &\xrightarrow{\mathbf{f}_W^{-1}} t_R \xrightarrow{\mathbf{f}_A} t \\ \tilde{\mathbf{x}} &\longmapsto \mathbf{u} = \mathbf{f}_W^{-1}(\tilde{\mathbf{x}}) \longmapsto \mathbf{x} = \mathbf{f}_A(\mathbf{u}). \end{aligned} \quad (4)$$

Therefore, the analytical expression of the affine mapping \mathbf{f}_S is

$$\begin{aligned}\mathbf{x} &= \mathbf{f}_S(\tilde{\mathbf{x}}) = \mathbf{f}_A(\mathbf{f}_W^{-1}(\tilde{\mathbf{x}})) = \mathbf{A}\mathbf{f}_W^{-1}(\tilde{\mathbf{x}}) + \mathbf{x}_0 = \mathbf{A}(\mathbf{W}^{-1}\tilde{\mathbf{x}} + \tilde{\mathbf{v}}) + \mathbf{x}_0 \\ &= \mathbf{A}\mathbf{W}^{-1}\tilde{\mathbf{x}} + \mathbf{A}\tilde{\mathbf{v}} + \mathbf{x}_0,\end{aligned}\quad (5)$$

being $\tilde{\mathbf{v}} = -\mathbf{W}^{-1}\tilde{\mathbf{x}}_0$. Comparing this expression with Equation (1) we realize that:

$$\mathbf{S} = \mathbf{A}\mathbf{W}^{-1}, \quad (6)$$

$$\mathbf{v} = \mathbf{A}\tilde{\mathbf{v}} + \mathbf{x}_0. \quad (7)$$

Note that if we consider $\tilde{\mathbf{x}}_0 = (0, 0)^T$, then $\tilde{\mathbf{v}} = (0, 0)^T$ and consequently, from Equation (7), $\mathbf{v} = \mathbf{x}_0$. The Jacobian matrix of the affine mapping (1), \mathbf{S} , is called *weighted Jacobian matrix* in reference [20].

Note that using this formulation, we can select different ideal elements to generate elements with different geometric properties. For isotropic triangular meshes, the ideal element is selected as the equilateral triangle. Thus, we have

$$\mathbf{W} = \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix}.$$

Similar reasoning can be developed for tetrahedral elements. In this case, if the tetrahedron in the physical space is delimited by the vertices: $\mathbf{x}_0 = (x_0, y_0, z_0)^T$, $\mathbf{x}_1 = (x_1, y_1, z_1)^T$, $\mathbf{x}_2 = (x_2, y_2, z_2)^T$, and $\mathbf{x}_3 = (x_3, y_3, z_3)^T$, then we have:

$$\mathbf{A} = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{pmatrix}.$$

In addition, for isotropic tetrahedral meshes, the *ideal* element is defined as an equilateral tetrahedron. Thus, we have

$$\mathbf{W} = \begin{bmatrix} 1 & 1/2 & 1/2 \\ 0 & \sqrt{3}/2 & \sqrt{3}/6 \\ 0 & 0 & \sqrt{2/3} \end{bmatrix}.$$

The weighted Jacobian matrix \mathbf{S} defined in (6) contains information about how much we have “distorted” the ideal element to become the physical one. Based on this matrix, Knupp [20] introduced the *shape distortion measure*

$$\eta(\mathbf{S}) = \frac{|\mathbf{S}|^2}{n\sigma(\mathbf{S})^{2/n}}. \quad (8)$$

where n is the space dimension ($n = 2$ for triangles and $n = 3$ for tetrahedra), $\sigma(\mathbf{S}) = \det(\mathbf{S})$ is the determinant of \mathbf{S} , and $|\mathbf{S}| = \sqrt{(\mathbf{S}, \mathbf{S})}$ is the Frobenius norm, being (\cdot, \cdot) a dot product for matrices defined as

$$(\mathbf{A}, \mathbf{B}) = \text{tr}(\mathbf{A}^T \mathbf{B}). \quad (9)$$

If the physical element is not inverted the image of this function is the interval $[1, \infty]$, achieving ∞ only when the physical element is degenerated, and 1 when it becomes the ideal element.

The *shape quality measure* is defined as the inverse of the shape distortion measure (see [20])

$$q(\mathbf{S}) = \frac{1}{\eta(\mathbf{S})} = \frac{n\sigma(\mathbf{S})^{2/n}}{|\mathbf{S}|^2}. \quad (10)$$

Note that the shape quality measure reaches a maximum value of 1 for the ideal element, and a minimum value 0 for degenerated elements.

The shape distortion measure (8) has been widely used in optimization procedures to smooth triangular and tetrahedral meshes, see for instance [20, 22, 23, 24, 25]. However, it presents asymptotes (or barriers) when $\sigma(\mathbf{S}) = 0$. In addition, it also presents local minima for tangled (inverted) configurations. Thus, the optimization algorithm might lead to smoothed meshes that contain inverted elements.

To overcome this drawback, Escobar *et al.* [17, 18] introduced a regularization for the shape distortion measure (8) in order to obtain a new measure without asymptotes and with a global minimum *near* the optimal position of the original shape distortion function (8). Specifically, they replace σ in (8) by

$$h(\sigma) = \frac{1}{2} \left(\sigma + \sqrt{\sigma^2 + 4\delta^2} \right), \quad (11)$$

where δ is an arbitrary small parameter that is chosen depending on the problem (see [17, 18, 36, 37] for further details). Taking into account equation (11), the regularized shape distortion measure for an element is

$$\eta^*(\mathbf{S}) = \frac{|\mathbf{S}|^2}{nh(\sigma(\mathbf{S}))^{2/n}}. \quad (12)$$

Note that if a sufficiently small value of δ is chosen, the minimum of the original and the modified shape distortion measures are close, see references [17, 32], and the mesh quality is not compromised. However, in our implementation the δ parameter is only set to non-zero values when an invalid (tangled) configuration of the mesh is being optimized. Once the elements in that configuration are valid, δ is set to zero for all the elements.

4 Distortion and quality measures for quadrilateral and hexahedral elements

According to Knupp [26], the distortion measure of a quadrilateral element can be computed by decomposing it into four triangles, see Figure 2. Similarly, the distortion measure of an hexahedral element can be computed by decomposing it into eight

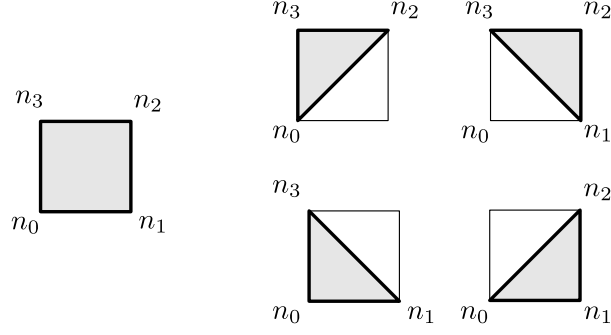


Figure 2: Decomposition of a quadrilateral into four triangles.

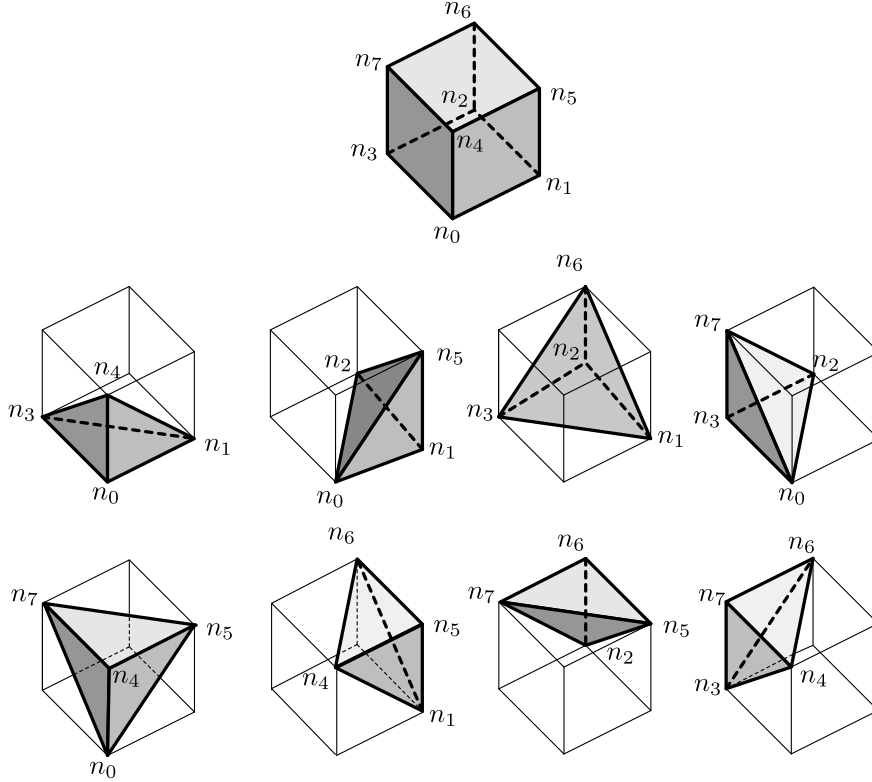


Figure 3: Decomposition of a hexahedron into eight tetrahedra.

tetrahedra, see Figure 3. Thus, given a quadrilateral (hexahedral) element, if we denote the shape distortion measure of the l th triangle (tetrahedron) by $\eta(\mathbf{S}^l)$, the shape distortion measure of a quadrilateral (hexahedral) elements is:

$$\eta = \left[\frac{1}{N} \sum_{l=1}^N (\eta(\mathbf{S}^l))^p \right]^{1/p} = \left[\frac{1}{N} \sum_{l=1}^N \left(\frac{|\mathbf{S}^l|^2}{n\sigma(\mathbf{S}^l)^{2/n}} \right)^p \right]^{1/p}, \quad (13)$$

where N is the number of sub-elements in which the original element is decomposed, and p is a parameter that can be used to penalize low-quality elements. Note that the values of the shape distortion measure (13) are also in the range $[1, \infty)$.

The shape distortion measure for quadrilaterals and hexahedra (13) presents barriers (where $\sigma(\mathbf{S}^l) = 0$) and local minima for tangled configurations that prevent its use in a minimization procedure. Therefore, the regularization introduced in Equation (11) is also used when the distortion measure of each sub-element is computed. That is, we substitute $\sigma(\mathbf{S}^l)$ by $h(\sigma(\mathbf{S}^l))$ in Equation (13). Thus, the regularized shape distortion measure for quadrilaterals and hexahedra is:

$$\eta^* = \left[\frac{1}{N} \sum_{l=1}^N (\eta^*(\mathbf{S}^l))^p \right]^{1/p} = \left[\frac{1}{N} \sum_{l=1}^N \left(\frac{|\mathbf{S}^l|^2}{nh(\sigma(\mathbf{S}^l))^{2/n}} \right)^p \right]^{1/p}. \quad (14)$$

For isotropic meshes, the ideal quadrilateral element is a square. Hence, the ideal triangles in which it is decomposed are the isosceles right triangles, see Figure 2. Similarly, the ideal hexahedron is a cube and the ideal tetrahedra are the isosceles right tetrahedra, see Figure 3. Therefore, the mappings between the reference and the ideal element for quadrilaterals and hexahedra are given by

$$\mathbf{W} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (15)$$

respectively. If we consider node n_i of the quadrilateral element of Figure 2, and we denote by \mathbf{u}_i its coordinates in the reference configuration, then mapping (15) assumes that \mathbf{u}_i is mapped to the node with the right angle of the ideal sub-element. The same applies to the hexahedral case. That is, to obtain transformations (15) we have to assign the node of the physical element to the node of the ideal sub-element with an inner angle of 90° . Using Equation (14), and according to Equation (10), the regularized shape quality measure for quadrilateral (hexahedral) elements is defined as:

$$q^* = \frac{1}{\eta^*} = \frac{1}{\left[\frac{1}{N} \sum_{l=1}^N (\eta^*(\mathbf{S}^l))^p \right]^{1/p}}. \quad (16)$$

Note that the values of the quality measure (16) for valid elements are in the range $[0, 1]$.

5 Simultaneous untangling and smoothing of quadrilateral and hexahedral meshes

5.1 Objective function

Given a quadrilateral or hexahedral mesh \mathcal{M} , let \mathcal{V} be the *set of free nodes* (the nodes that can be moved to improve the quality of the mesh), and let v be a given node $v \in \mathcal{V}$. We define the *local sub-mesh* associated to v , \mathcal{M}_v , as the set of elements that contain node v , see Figure 4(a).

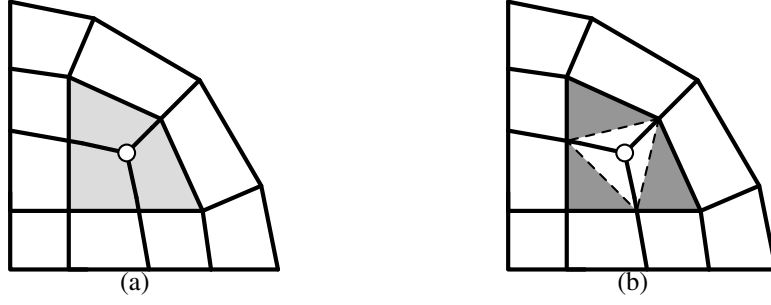


Figure 4: Quadrilateral mesh with a marked node: (a) in light gray elements that belong to the marked node sub-mesh ; and (b) in dark gray sub-triangles of the quadrilateral decomposition not adjacent to the marked node.

The simultaneous untangling and smoothing algorithm for quadrilateral and hexahedral meshes is based on the optimization of an objective function defined in terms of the regularized distortion measure (14) of the elements in the local sub-mesh. Let \mathbf{x}_v be the coordinates of a free node $v \in \mathcal{V}$, and assume that the local sub-mesh, \mathcal{M}_v , is composed by M elements (quadrilaterals or hexahedra in our case). We define the objective function as:

$$K_{\eta^*}(\mathbf{x}_v) = \frac{1}{M} \sum_{k=1}^M (\eta_k^*(\mathbf{x}_v))^p,$$

where $\eta_k^*(\mathbf{x}_v)$ is computed according to Equation (14) as:

$$\eta_k^*(\mathbf{x}_v) = \left[\frac{1}{N} \sum_{l=1}^N \left(\frac{|\mathbf{S}_k^l(\mathbf{x}_v)|^2}{nh(\sigma(\mathbf{S}_k^l(\mathbf{x}_v)))^{2/n}} \right)^p \right]^{1/p},$$

being \mathbf{S}_k^l the weighted Jacobian matrix of the l th sub-element of the k th element in the local sub-mesh. Parameter p allows increasing the penalization of low quality elements. Note that as we increase p , the objective function takes higher values for low-quality elements. In our application, we use $p = 2$. Finally, the objective function for node v can be written as:

$$K_{\eta^*}(\mathbf{x}_v) = \frac{1}{M} \frac{1}{N} \sum_{k=1}^M \sum_{l=1}^N \left(\frac{|\mathbf{S}_k^l(\mathbf{x}_v)|^2}{nh(\sigma(\mathbf{S}_k^l(\mathbf{x}_v)))^{2/n}} \right)^p. \quad (17)$$

5.2 Minimization procedure

In this work, we deduce a continuous minimization procedure to simultaneously untangle and smooth quadrilateral and hexahedral meshes. Specifically, for each free node, we propose to minimize the objective function (17) defined on its local sub-mesh. Algorithm 1 details the procedure to improve the quality of a mesh, \mathcal{M} . The main idea consists on looping on all the free nodes (Lines 6-9). For each free node,

Algorithm 1 Procedure to smooth a mesh, \mathcal{M} .

```

1: function smoothMesh(Mesh  $\mathcal{M}$ , Real  $maxDisplacement$ )
2:   DistortionMeasure  $f \leftarrow K_{\eta^*}$  ( $\triangleright$ ) See Equation (17)
3:   Boolean  $isConverged \leftarrow \text{false}$ 
4:   while not  $isConverged$  do
5:     Real  $displacement \leftarrow 0$ 
6:     for all  $v \in \mathcal{V}$  do
7:       Real  $currentDisplacement \leftarrow \text{smoothNode}(v, maxIterations, f)$ 
8:        $displacement \leftarrow \max\{displacement, currentDisplacement\}$ 
9:     end for
10:     $isConverged \leftarrow (displacement \leq maxDisplacement)$ 
11:  end while
12: end function

```

we modify its position while keeping fixed the position of all the other nodes (Line 7). The process ends when the maximum displacement of the nodes is below a threshold prescribed by the user (Line 10).

To compute the new position of a free node, v , we minimize the objective function (17) on the local sub-mesh using a line-search iterative process, see Algorithm 2. Let \mathbf{x}_v^k denote the location of a free node v at iteration k of the line-search process. Then, the location at the $k + 1$ iteration, \mathbf{x}_v^{k+1} , is computed as:

$$\mathbf{x}_v^{k+1} = \mathbf{x}_v^k + \alpha_v^k \mathbf{p}_v^k, \quad (18)$$

where \mathbf{p}_v^k is the advancing direction and α_v^k the step length. There are several methods to compute both the step length and the advancing vector. In this work, we have considered the steepest-descent and the Newton-Raphson methods, see reference [38] for more details. Note that to use these methods, it is required the first and second derivatives of the objective function detailed in A.

To increment the robustness of the optimization procedure, we translate the sub-mesh, \mathcal{M}_v , to the origin of coordinates, and we scale it using a reference size before starting the actual minimization process, Line 3 in Algorithm 2. On the one hand, the translation to the origin reduces the relative error of the process, since the coordinates of the nodes are of the same order of magnitude as the size of the elements. On the other hand, the scaling of the elements increases the robustness, since the optimization process is independent of the actual size of the elements. To increment the computational efficiency of the optimization procedure we do not iterate until convergence in Algorithm 2. Instead we propose to perform only a given number of iterations of the line-search procedure, Line 5.

For a given free node $v \in \mathcal{V}$, in a quadrilateral (hexahedral) mesh, all the sub-elements of the sub-mesh, \mathcal{M}_v , contribute to the value of the objective function (17). However, some of them are not adjacent to the free node. For instance, Figure 4(b) shows a local sub-mesh composed by three quadrilateral elements with three triangular sub-elements (marked in grey) that are not adjacent to node v . It is important to point

Algorithm 2 Procedure to smooth a node, v .

```

1: function smoothNode(Node  $v$ , Int  $maxNodeIterations$ , DistortionMeasure
    $f$ )
2:    $\mathcal{M}_v \leftarrow \text{getSubMesh}(v)$ 
3:    $\text{translateAndScale}(\mathcal{M}_v)$ 
4:   Vector  $\mathbf{x}_v^0 \leftarrow \text{getPoint}(v)$ 
5:   for  $k \leftarrow 1, maxNodeIterations$  do
6:     Vector  $\mathbf{p}_v^{k-1} \leftarrow \text{getAdvancingDirection}(\mathbf{x}_v^{k-1}, \mathcal{M}_v, f)$ 
7:     Real  $\alpha_v^{k-1} \leftarrow \text{getStepLength}(\mathbf{x}_v^{k-1}, \mathbf{p}_v^{k-1}, \mathcal{M}_v, f)$ 
8:     Vector  $\mathbf{x}_v^k \leftarrow \mathbf{x}_v^{k-1} + \alpha_v^{k-1} \mathbf{p}_v^{k-1}$  ( $\triangleright$ ) See Equation (18)
9:   end for
10:   $\text{undoScaleAndTranslate}(\mathcal{M}_v)$ 
11: end function

```

out that these sub-elements do not contribute to the first and second derivatives of the objective function. Thus, to not deteriorate the performance of the proposed algorithm, they are not considered in the computation of the objective function and its derivatives. That is, we only consider three sub-triangles in the quadrilateral decomposition and four tetrahedra in the hexahedron decomposition. Thus, the computational cost of the optimization procedure can be approximately reduced by a 25% for quadrilateral meshes and a 50% for hexahedra meshes.

6 Smoothing framework for two and three dimensional meshes

The smoothing procedure detailed in Algorithms 1 and 2 is independent of the selected objective function (i.e. the element type) and the minimization method. For this reason, in this work we have adopted an object-oriented paradigm to create a common framework to smooth meshes composed by any kind of elements. Object-oriented frameworks have been extensively used in mesh generation, see for instance [39, 40, 41, 42]. Using the optimization procedure detailed in Section 5, we are able to untangle and smooth meshes composed either by triangles, quadrilaterals, tetrahedra or hexahedra by assigning the corresponding objective function to each type of element, see Line 2 in Algorithm 1.

To compute the values and the derivatives of each type of objective function, we use the object structure defined in Figure 5. We define a base object, `DistortionMeasure`, that implements the basic interface for all the distortion measures. Specifically, this base object provides an interface to obtain the value and the derivatives of the distortion measure. The different distortion measures will overwrite these functions in order to obtain their corresponding values and derivatives. In addition, `DistortionMeasure` object provides an interface to assign the node in which the shape distortion measure is evaluated, and the δ parameter to be used, if required.

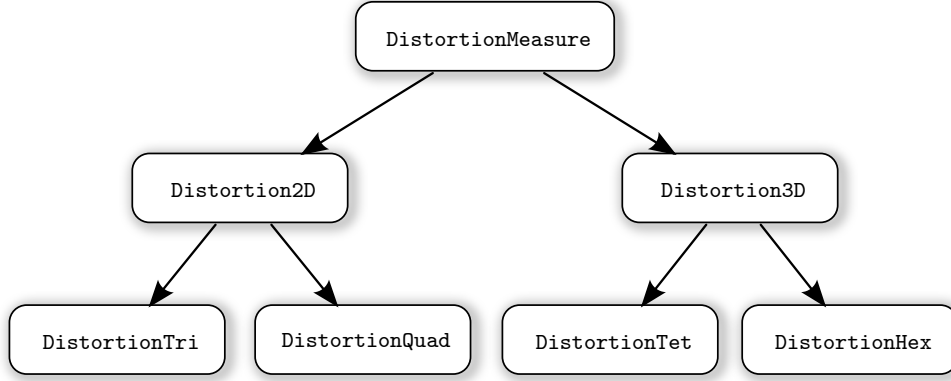


Figure 5: Object structure to model an objective function for the different types of elements.

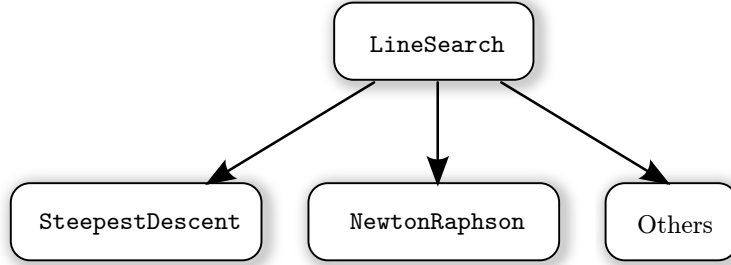


Figure 6: Object structure to model a minimization algorithm for different optimization methods.

`Distortion2D` and `Distortion3D` objects are derived from `DistortionMeasure`. These two objects define the interface for the two-dimensional and three-dimensional shape distortion measures. From the three points that define a triangle, `Distortion2D` computes the corresponding weighted Jacobian matrix, \mathbf{S} . Then, it calculates the value $\eta^*(\mathbf{S})$, Equation (14), and its derivatives for the given matrix \mathbf{S} . Similarly, `Distortion3D` computes the weighted Jacobian matrix, \mathbf{S} , from the four points that define a tetrahedron, and then calculates the value $\eta^*(\mathbf{S})$ and its derivatives.

`DistortionTri` and `DistortionQuad` are derived objects from `Distortion2D`, while `DistortionTet` and `DistortionHex` derive from `Distortion3D`. The derived classes are responsible of calculating the actual value of the function and its derivatives. To this end, the derived classes add the contributions the sub-elements adjacent to the given node, according to Equation (17). To compute each contribution, the derived classes perform a call to the parent object with the nodal coordinates that define each the sub-element. Hence, `DistortionTri` and `DistortionTet` perform only one call, while `DistortionQuad` and `DistortionHex` perform three and four calls, respectively.

We also propose an object structure to implement the line-search minimization algorithms, see Figure 6. We first define a base object, `LineSearch`, that implements

the common interface for all the line-search minimization methods. This object is responsible of controlling the maximum number of iterations and the tolerances for the convergence criteria, as well as implementing the main loop defined in Algorithm 2. The computation of the advancing direction and the step length is performed by the derived objects `SteepestDescent` and `NewtonRaphson`, see Lines 6 and 7 of Algorithm 2, respectively. Using this diagram class, we can decouple the smoother algorithm from the minimization method and, for this reason, it is easier to add additional minimization algorithms with different properties.

The main advantage of this smoothing framework is that it provides a unified implementation to smooth meshes composed by any type of elements, and to optimize the objective function using the desired minimization algorithm. It is only necessary to assign the correct objective function to smooth the corresponding mesh, and the desired minimization algorithm to optimize the objective function. Moreover, different element distortion measures can be used to select different geometric properties on the final mesh. For instance, the mesh can be adapted to a size field to better reproduce element size functions, or to create boundary layers.

7 Examples

In this section, we present several examples to illustrate the robustness and applicability of the proposed method. The first two examples focus on the robustness of the method. In particular they present a quadrilateral and hexahedral mesh such that the location of all inner nodes is randomized, generating a large number of inverted elements. The last three examples are devoted to the optimization of hexahedral meshes for real-world geometries. Specifically, the third and four examples present an application to semi-structured hexahedral meshes for multi-sweeping [3, 4, 5, 6] mechanical pieces. In the last example, we improve the quality of a unstructured hexahedral mesh generated using the receding front method [14].

For all the examples, we have used $p = 2$ in Equation (17) and one iteration for the line-search minimization methods, see Algorithm 2. In addition, for all of them, we present a table summarizing the quality statistics of the meshes. In particular, we provide: the minimum, the maximum, the mean and the standard deviation of the elements quality, and the number of tangled elements in the mesh. We highlight that in all cases, the smoothed mesh increases the minimum and mean values of the mesh quality and decreases its standard deviation. Moreover, all the smoothed meshes do not contain any inverted element.

All the algorithms presented in Section 5, and all the objects described in Section 6 have been implemented using C++ in the meshing environment `ez4u` [43, 44]. All the examples were executed in an Intel i7-3770 CPU at 3.40GHz with 16 GB of RAM under Ubuntu 13.10 operating system. Although no special attention is focused on optimizing the developed code, it should be mentioned that the smallest example (the gear geometry) took less than one second to smooth and untangle the mesh. For

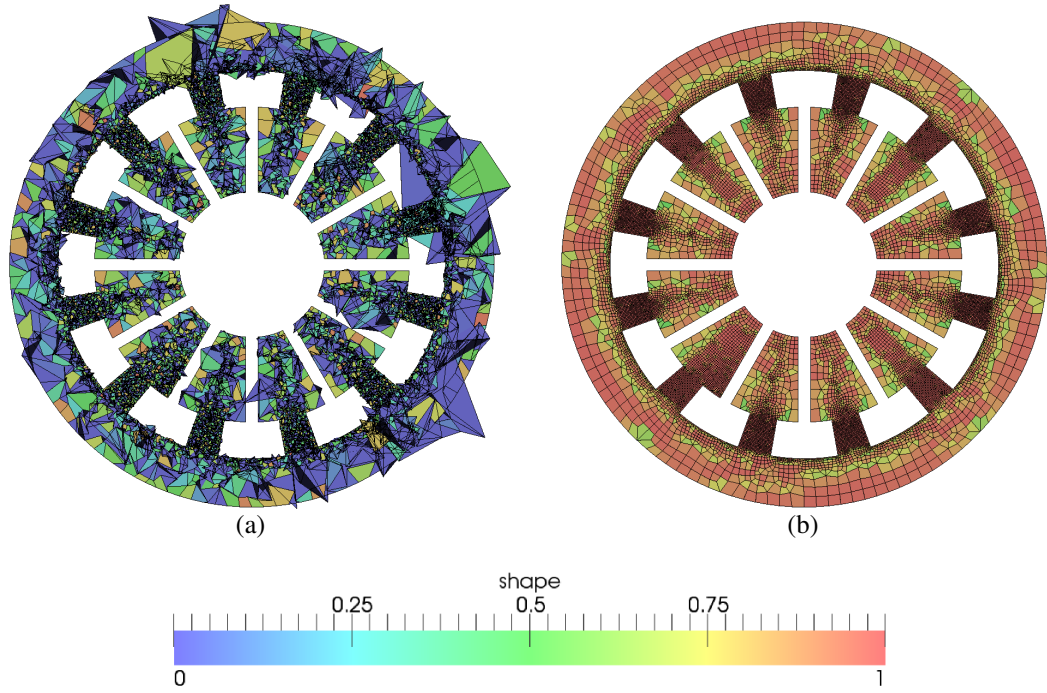


Figure 7: Quadrilateral mesh generated for a pressure plate: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

the medium-size examples (the pressure plate, the mechanical piece and the linking rod) around seven second were needed to obtain the final mesh. Finally, the largest example took around eighty seconds to optimize the mesh.

2D quadrilateral mesh for a pressure plate. This example presents a two-dimensional quadrilateral mesh generated for a pressure plate. It is composed by 19101 nodes and 18099 elements. The interior nodes of the mesh have been moved to a random position before applying the proposed simultaneous untangling and smoothing algorithm, see Figure 7(a). Then, we have applied the optimization algorithm using the Newton-Raphson method obtaining a valid mesh (without tangled elements) composed by high-quality elements, see Figure 7(b).

Figure 8(a) presents the element quality distribution for the generated mesh with a random position for the nodes. Note that it contains 11141 inverted elements. Figure 8(b) shows the element quality distribution after applying the proposed smoother algorithm. In this case, there are no degenerated elements and the mesh quality has been improved. In particular, the minimum and mean values of the shape quality have been incremented to 0.43 and 0.93, respectively, see Table 1.

3D hexahedral mesh of a mechanical piece. This example presents a hexahedral mesh generated for a mechanical piece using the multi-sweeping method (14781 nodes and 11370 elements). In this example, similar to the previous one, we have ran-

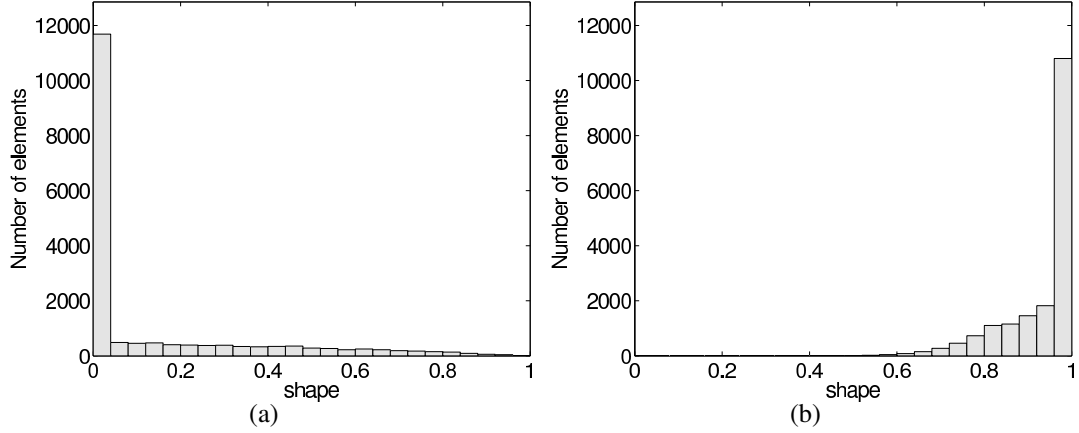


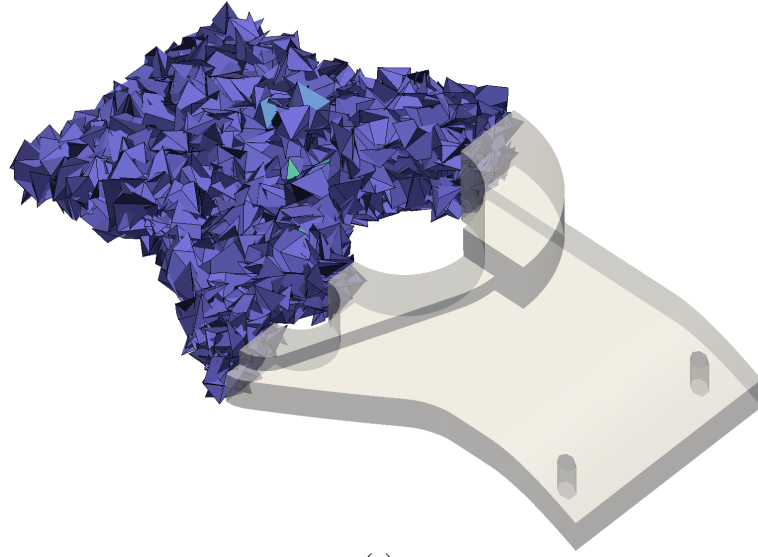
Figure 8: Element quality distribution of the quadrilateral mesh generated for a pressure plate: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

domized the location of the inner nodes, see Figure 9(a). Then, we have applied the proposed hexahedral smoothing algorithm using the Newton-Raphson method, and we have obtained an untangled high-quality mesh, See Figure 9(b). Note that the proposed optimization algorithm is able to recover a valid position for the inner nodes, generating a high-quality mesh even when the initial one contains 9856 inverted elements, see Figure 10.

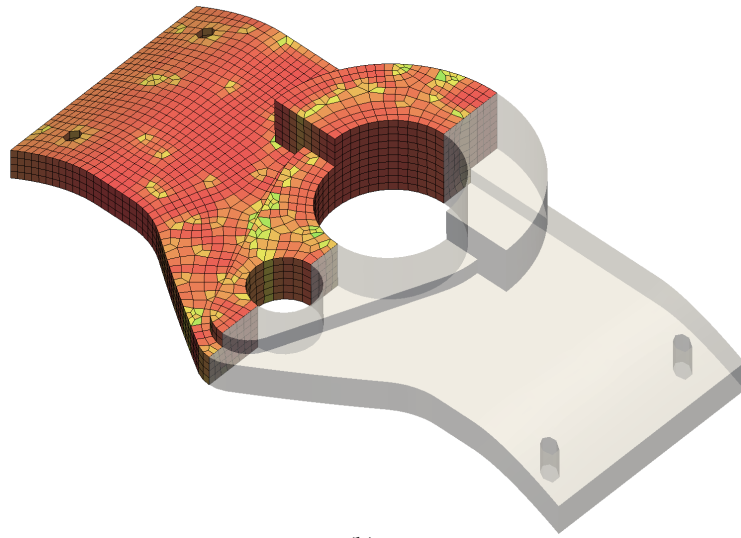
Table 1 details the statistical information for the randomized and optimized meshes for the mechanical piece. Similar to the previous 2D example, we highlight that using the proposed smoothing algorithm we are able to untangle the inverted elements and increase the minimum and the mean shape quality from 0.0 to 0.57 and from 0.03 to 0.92, respectively.

3D semi-structured mesh for a gear. In this example, we improve the quality of a hexahedral mesh for a gear generated using the multi-sweeping method. It is composed by 5246 nodes and 3606 elements. Figure 11(a) presents the output of the multi-sweeping method when the position of the inner nodes is computed taking into account only target cap surfaces. Note that the method generates stretched elements in the upper layer and, for this reason, low-quality elements appear. When we apply the proposed smoother algorithm using the Newton-Raphson method, a high-quality hexahedral mesh without inverted elements is obtained, see Figure 11(b). Figure 12(a) shows the element quality distribution for the initial mesh generated for a gear. Note that the multi-sweeping method has generated a great number of low-quality elements and 4 inverted elements. When the optimization algorithm is applied, the quality of the mesh is improved and the mesh does not contain any inverted element.

Table 2 presents a comparison of the statistical information of the original gear mesh and the smoothed meshes obtained using a Newton-Raphson and steepest descent line-search approaches. The statistical information for the optimized meshes are similar. That is, we have obtained comparable results when different optimization



(a)



(b)

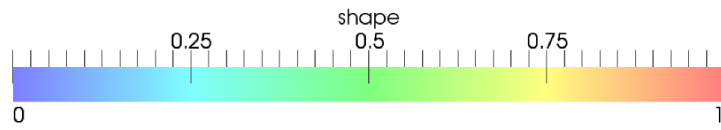


Figure 9: Hexahedral mesh generated for a mechanical piece: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

methods to minimize the objective function are used.

3D semi-structured mesh for a linking rod. In this example, we present the mesh generated for a linking rod using the multi-sweeping algorithm (14783 nodes

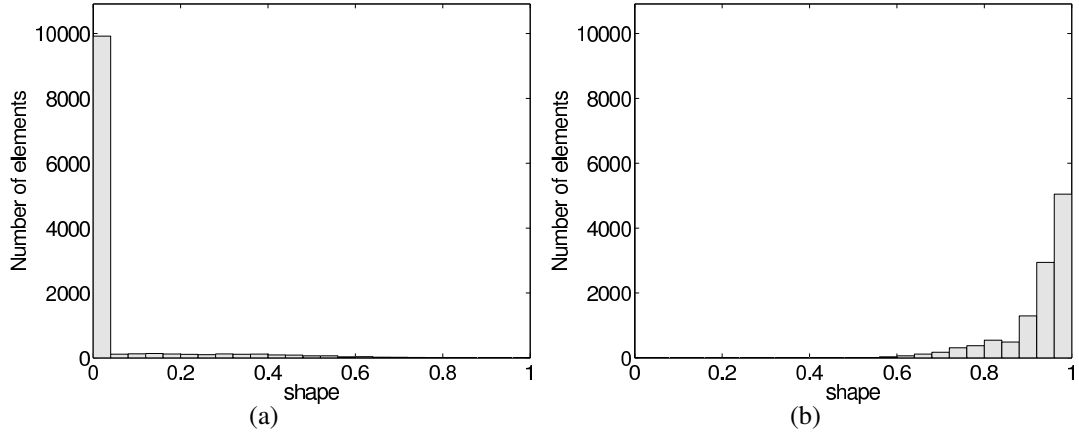


Figure 10: Element quality distribution of the hexahedral mesh generated for a mechanical piece: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

and 11316 elements). Similar to the previous example, the position of the inner nodes is computed taking into account only target cap surfaces. Thus, the mesh generated using this method contains several areas with low-quality elements, see Figure 13(a). To improve its quality we have applied the optimization algorithm obtaining a mesh composed by valid and high-quality elements, see Figure 13(b).

Figure 14(a) presents the element quality distribution for the initial mesh. Note that this mesh contains inverted and low-quality elements. Figure 14(b) shows the element quality distribution for the linking rod mesh after applying the proposed optimization algorithm using the Newton-Raphson method. Note that the improved mesh does not contain tangled elements and that the minimum quality is above 0.36.

Table 2 presents a comparison of the statistical information of the original linking rod mesh and the smoothed meshes obtained using a Newton-Raphson and steepest descent line-search approach. The original mesh contains 16 inverted elements and the smoothed meshes do not contain any inverted element. In this case, the results between the mesh obtained using a Newton-Raphson solver and the mesh obtained using a steepest descent solver are also comparable. In fact they provide meshes with the same minimum and mean values of the element quality.

3D unstructured mesh for the exterior domain of an aircraft. In this example we present the mesh for the exterior domain of an aircraft generated using the receding front method. The mesh, composed by 32454 nodes and 30962 elements, contains inverted and low-quality elements due to the behavior of the mesh generation algorithm, see Figure 15(a). The quality of the mesh can be improved by applying the proposed optimization algorithm, see Figure 15(b). Note that the optimized mesh does not contain inverted elements and the quality of the elements has been improved.

Figure 16(a) shows the element quality distribution of the mesh generated for the exterior domain of a plane. Note that the receding front method has generated low-quality and inverted elements. Figure 16(b) presents the element quality distribution

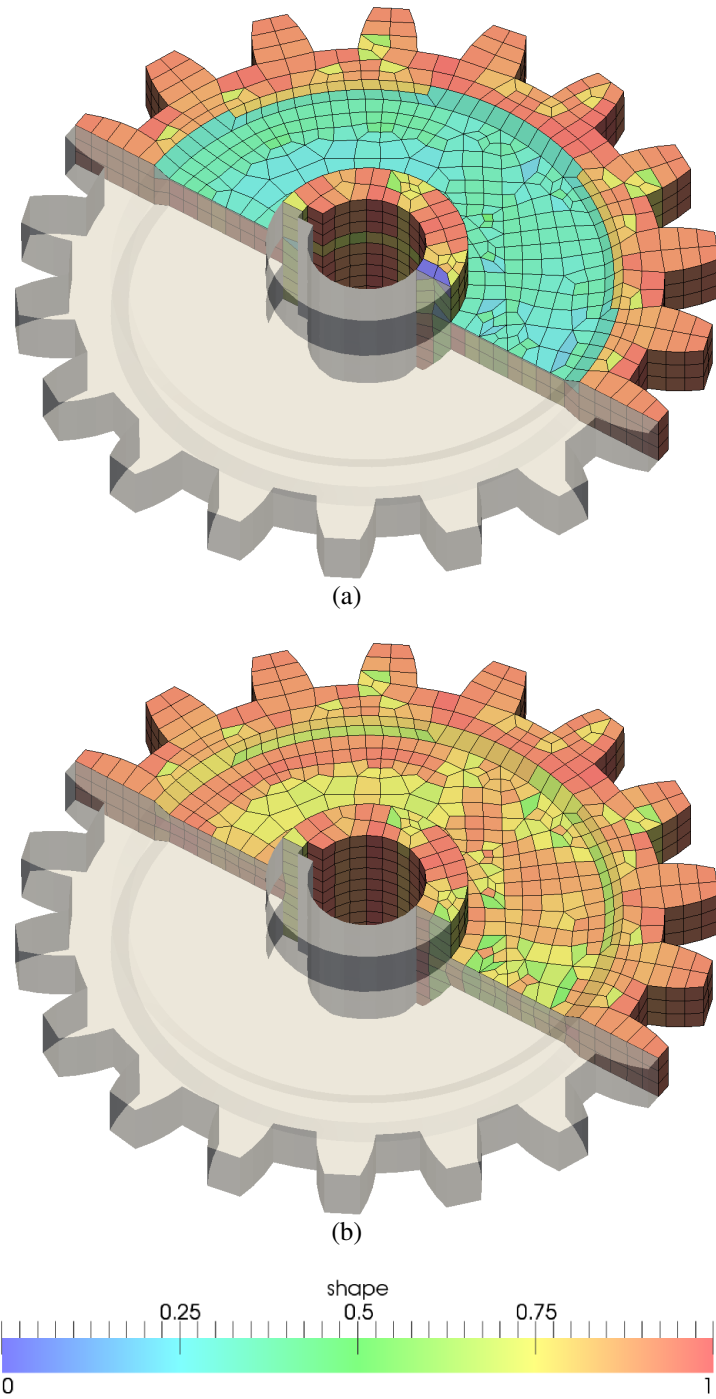


Figure 11: Hexahedral mesh generated for a gear using the multi-sweeping method: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

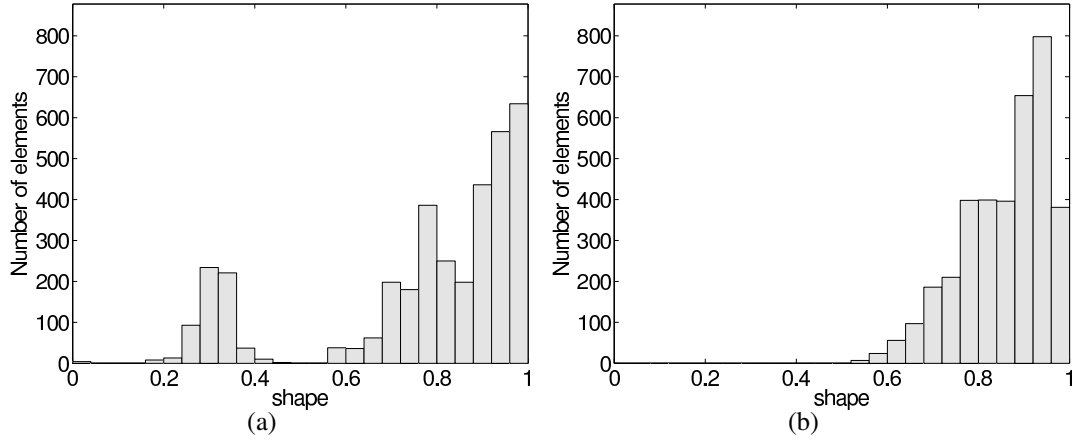


Figure 12: Element quality distribution of the hexahedral mesh generated for a gear: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

Geom.	Mesh	Min.Q.	Max.Q.	Mean Q.	Std.Dev.	Tang.
Pressure plate	Initial	0.00	0.98	0.13	0.22	11141
	Smoothed	0.43	1.00	0.93	0.08	0
Mechanical piece	Initial	0.00	0.84	0.03	0.11	9856
	Smoothed	0.57	0.99	0.92	0.07	0

Table 1: Shape quality statistics of the meshes for the pressure plate and the mechanical piece.

of the mesh after applying the proposed smoothing algorithm. Note that in this case, the minimum element quality has been increased.

Table 2 presents a comparison of the statistical information of the original mesh for the exterior domain of a plane and the smoothed meshes obtained using a Newton-Raphson and steepest descent line-search approaches. Similarly to the other examples, the smoothed meshes improve the minimum quality of the original mesh, which contains 149 inverted elements. The proposed optimization algorithm obtains a mesh without inverted elements, using either a Newton-Raphson or a steepest descent approach. In addition, similar statistical results of the elements quality are obtained for the mesh optimized using the Newton-Raphson and the steepest descent methods.

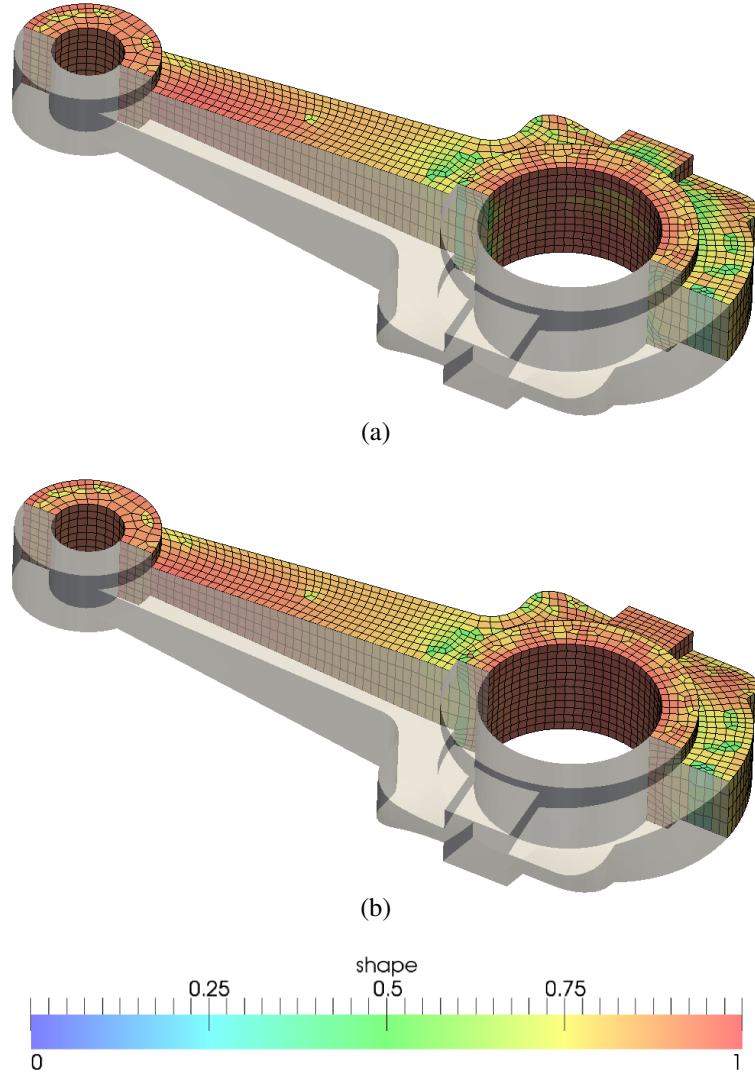


Figure 13: Hexahedral mesh generated for a linking rod using the multi-sweeping method: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

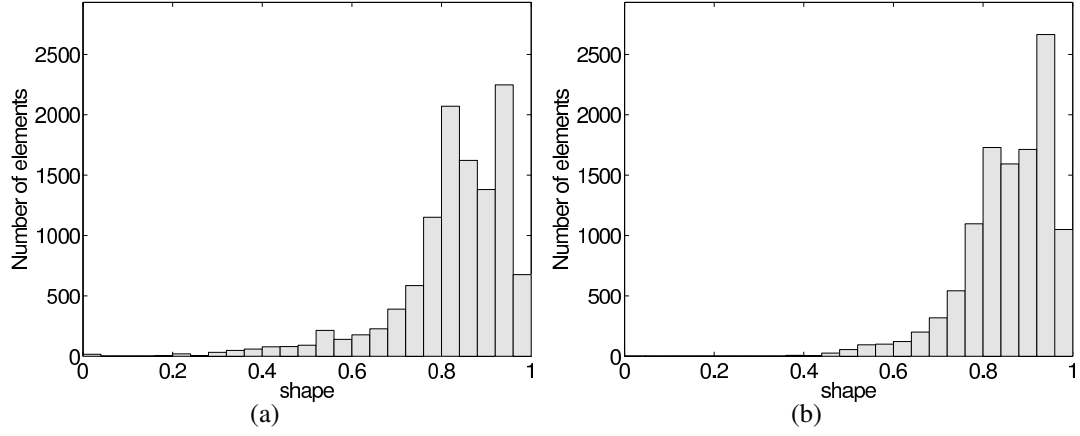
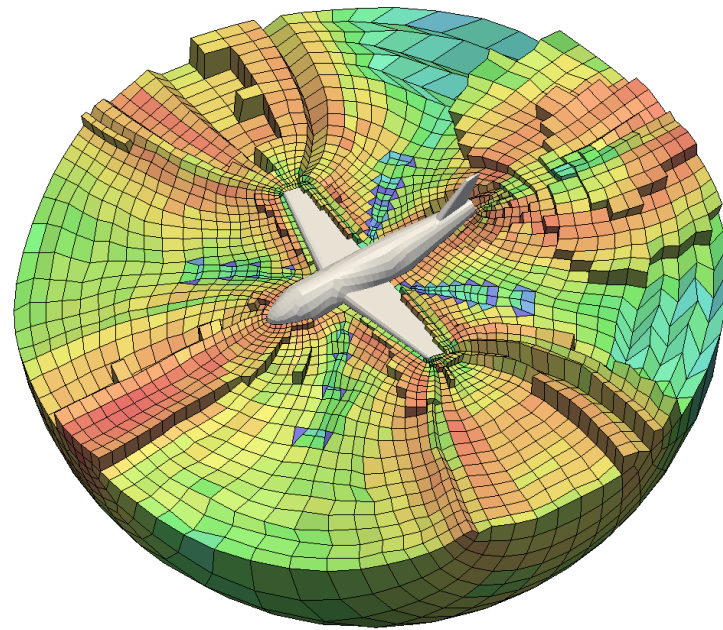


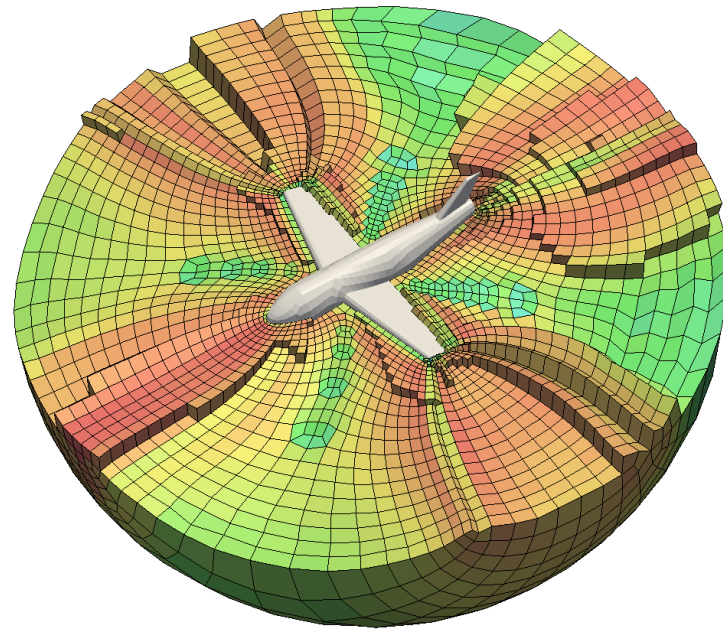
Figure 14: Element quality distribution of the hexahedral mesh generated for a linking rod: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

Geom.	Mesh	Min.Q.	Max.Q.	Mean Q.	Std.Dev.	Tang.
Gear	Initial	0.00	0.99	0.76	0.23	4
	N-R	0.41	0.99	0.83	0.11	0
	Step. Desc.	0.39	0.99	0.86	0.09	0
Linking rod	Initial	0.00	0.99	0.82	0.13	16
	N-R	0.36	0.99	0.85	0.10	0
	Step. Desc.	0.36	0.99	0.85	0.09	0
Aircraft	Initial	0.00	0.99	0.73	0.16	149
	N-R	0.31	0.99	0.76	0.15	0
	Step. Desc.	0.31	0.99	0.79	0.13	0

Table 2: Shape quality statistics of the meshes for the gear, linking rod and aircraft.



(a)



(b)

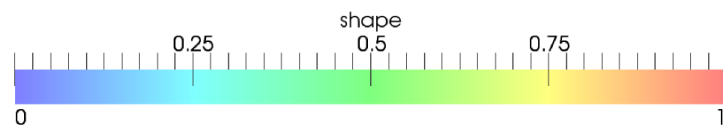


Figure 15: Element quality distribution of the hexahedral mesh generated for an aircraft: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

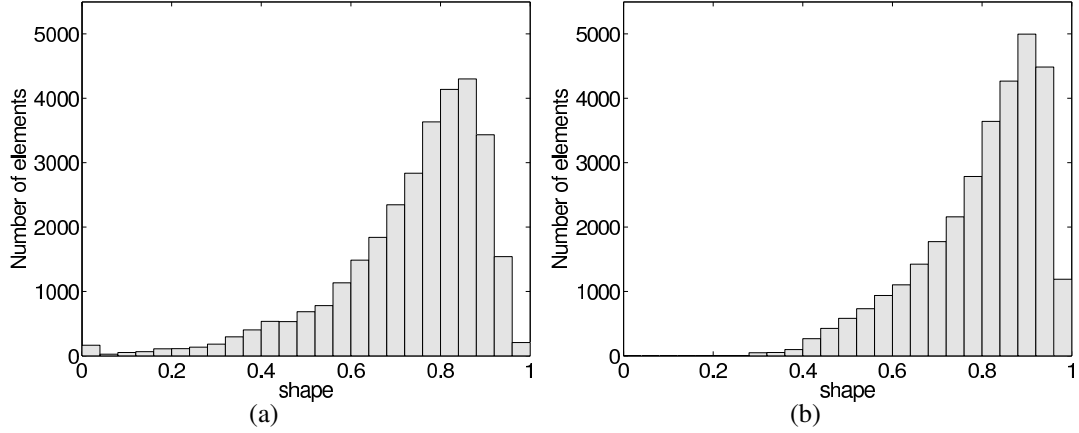


Figure 16: Element quality distribution of the hexahedral mesh generated for a plane: (a) before applying the untangling and smoothing algorithm; and (b) after applying the untangling and smoothing algorithm.

8 Concluding remarks

In this work, we have presented an optimization algorithm to simultaneously untangle and smooth quadrilateral and hexahedral meshes. The method minimizes an objective function defined in terms of a regularized algebraic distortion measure. In order to improve the computational efficiency, we only perform one iteration of the line-search optimization algorithm for each free node. Experience has shown that this regularization improves the time to obtain an optimized mesh. Moreover, to not deteriorate the performance of the optimization process, the evaluation of the objective function only takes into account the relevant sub-elements in the quadrilateral and hexahedral decomposition. In addition, we have also improved the robustness and the precision of the minimization process by translating the local sub-mesh of each node to the origin of coordinates and then scale it to a reference size. To implement the proposed untangling and smoothing algorithm, we have adopted an object-oriented paradigm to create a common framework to smooth meshes composed by any kind of elements and using a generic line-search minimization method to optimize the objective function.

Several aspects of the algorithm can be extended in the near future. Specifically, we have to improve the computational time to smooth a mesh, specially in the three-dimensional case. While the algorithm is competitive to smooth quadrilateral meshes, it is required to reduce the time to smooth a hexahedral mesh. The objective function for hexahedral elements is more expensive to compute than the objective function for quadrilateral elements. For instance, a hexahedron is decomposed in more sub-elements than a quadrilateral and, in addition, there are more hexahedra than quadrilaterals adjacent to a given node. Moreover, the objective function related to quadrilaterals is computed in two dimensions while the objective function related to hexahedra is computed in three-dimensions, which further increase the cost of computing the objective function.

We plan to apply the proposed optimization framework to improve the quality of mixed meshes composed by tetrahedral and hexahedral elements. Moreover, note that pyramid and prismatic elements can also be included in these mixed meshes since each one of these elements, similar to hexahedral elements, can be decomposed in several tetrahedra. While the proposed algorithm is able to smooth the presented meshes taking a few seconds, it would be interesting to further decrease the computational time to smooth a mesh for large industrial problems. To this end, we propose to use a parallelization technique to distribute the tasks to different processors. Moreover, in order to avoid race conditions and to minimize the communication between the processors, different coloring techniques would be considered. Finally, it may be of interest to compare the performance of the proposed optimization method when it is applied to meshes composed by different kind of elements.

A First and second derivatives of the objective function

In this appendix, we detail the first and second derivatives of the objective function, K_{η^*} , see Equation (17). To this end, we express the first and second derivatives of the objective function in terms of the derivatives of the shape distortion measure, η^* , see Equation (14) using the chain rule.

$$\frac{\partial K_{\eta^*}}{\partial x_i} = \frac{p}{NM} \sum_{k=1}^M \sum_{l=1}^N \frac{\partial \eta_{kl}^*}{\partial x_i} (\eta_{kl}^*)^{p-1}, \quad (19)$$

$$\frac{\partial^2 K_{\eta^*}}{\partial x_i \partial x_j} = \frac{p}{NM} \sum_{k=1}^M \sum_{l=1}^N (p-1) \frac{\partial \eta_{kl}^*}{\partial x_i} \frac{\partial \eta_{kl}^*}{\partial x_j} (\eta_{kl}^*)^{p-2} + \frac{\partial^2 \eta_{kl}^*}{\partial x_i \partial x_j} (\eta_{kl}^*)^{p-1}, \quad (20)$$

where x_i and x_j are two generic coordinates, and η_{kl}^* is the shape distortion measure of the l th sub-element contained in the k th element of the local sub-mesh. To obtain the final expression for the derivatives of the objective function, we need to deduce the first and second derivatives of the regularized distortion measure for a sub-element, see Equation (12). Applying the chain rule we obtain

$$\frac{\partial \eta^*}{\partial x_i} = \frac{1}{n} \left(\frac{\partial (\mathbf{S}, \mathbf{S})}{\partial x_i} \frac{1}{h^{2/n}} + (\mathbf{S}, \mathbf{S}) \frac{\partial}{\partial x_i} \left(\frac{1}{h^{2/n}} \right) \right), \quad (21)$$

where the dot product (9) is used. We decompose Equation (21) in two terms. They can be expressed as

$$\frac{\partial (\mathbf{S}, \mathbf{S})}{\partial x_i} \frac{1}{h^{2/n}} = 2 \left(\frac{\partial \mathbf{S}}{\partial x_i}, \mathbf{S} \right) \frac{1}{h^{2/n}} \quad (22)$$

$$(\mathbf{S}, \mathbf{S}) \frac{\partial}{\partial x_i} \left(\frac{1}{h^{2/n}} \right) = -(\mathbf{S}, \mathbf{S}) \frac{2}{n} \frac{1}{h^{2/n}} \frac{\partial \sigma}{\partial x_i} \left(\frac{1}{\sqrt{\sigma^2 + 4\delta^2}} \right), \quad (23)$$

where h is defined in Equation (11) and

$$\frac{\partial \sigma}{\partial x_i} = \det(\mathbf{S}) \text{tr} \left(\mathbf{S}^{-1} \frac{\partial \mathbf{S}}{\partial x_i} \right).$$

Combining Equations (22) and (23), we obtain the final expression for the first derivative of the distortion measure

$$\frac{\partial \eta^*}{\partial x_i} = 2\eta^* \left[\frac{1}{|\mathbf{S}|^2} \left(\frac{\partial \mathbf{S}}{\partial x_i}, \mathbf{S} \right) - \frac{1}{n} \frac{\partial \sigma}{\partial x_i} \frac{1}{\sqrt{\sigma^2 + 4\delta^2}} \right], \quad (24)$$

The second derivative of the shape distortion measure is

$$\begin{aligned} \frac{\partial^2 \eta^*}{\partial x_i \partial x_j} &= 2 \frac{\partial \eta^*}{\partial x_j} \left[\left(\frac{\partial \mathbf{S}}{\partial x_i}, \mathbf{S} \right) \frac{1}{|\mathbf{S}|^2} - \frac{1}{n} \frac{\partial \sigma}{\partial x_i} \frac{1}{\sqrt{\sigma^2 + 4\delta^2}} \right] + \\ &2\eta^* \frac{\partial}{\partial x_j} \left[\left(\frac{\partial \mathbf{S}}{\partial x_i}, \mathbf{S} \right) \frac{1}{|\mathbf{S}|^2} - \frac{1}{n} \frac{\partial \sigma}{\partial x_i} \frac{1}{\sqrt{\sigma^2 + 4\delta^2}} \right]. \end{aligned} \quad (25)$$

Similarly to the first derivative case, we split Equation (25) in two terms. The first term is

$$\frac{\partial \eta^*}{\partial x_j} \left[\left(\frac{\partial \mathbf{S}}{\partial x_i}, \mathbf{S} \right) \frac{1}{|\mathbf{S}|^2} - \frac{1}{n} \frac{\partial \sigma}{\partial x_i} \frac{1}{\sqrt{\sigma^2 + 4\delta^2}} \right] = \frac{1}{\eta^*} \frac{\partial \eta^*}{\partial x_i} \frac{\partial \eta^*}{\partial x_j}. \quad (26)$$

The second term of Equation 25 is decomposed again into two terms. The first one is

$$\begin{aligned} 2\eta^* \frac{\partial}{\partial x_j} \left[\left(\frac{\partial \mathbf{S}}{\partial x_i}, \mathbf{S} \right) \frac{1}{|\mathbf{S}|^2} \right] &= \\ 2\eta^* \left[\left(\frac{\partial^2 \mathbf{S}}{\partial x_i \partial x_j}, \mathbf{S} \right) \frac{1}{|\mathbf{S}|^2} + \left(\frac{\partial \mathbf{S}}{\partial x_i}, \frac{\partial \mathbf{S}}{\partial x_j} \right) \frac{1}{|\mathbf{S}|^2} - \frac{2}{|\mathbf{S}|^2} \left(\frac{\partial \mathbf{S}}{\partial x_j}, \mathbf{S} \right) \right], \end{aligned} \quad (27)$$

and the second one

$$2\eta^* \frac{\partial}{\partial x_j} \left(-\frac{1}{n} \frac{\partial \sigma}{\partial x_i} \frac{1}{\sqrt{\sigma^2 + 4\delta^2}} \right) = 2\eta^* \frac{\sigma}{n} \frac{\partial \sigma}{\partial x_i} \frac{\partial \sigma}{\partial x_j} \frac{1}{(\sqrt{\sigma^2 + 4\delta^2})^3}. \quad (28)$$

Finally, introducing Equations (26), (27) and (28) in Equation (25), we obtain the final expression for the second derivatives of the shape distortion measure

$$\begin{aligned} \frac{\partial^2 \eta^*}{\partial x_i \partial x_j} &= \frac{1}{n} \frac{\partial \eta^*}{\partial x_i} \frac{\partial \eta^*}{\partial x_j} + 2\eta \left[\frac{1}{|\mathbf{S}|^2} \left(\frac{\partial \mathbf{S}}{\partial x_i}, \frac{\partial \mathbf{S}}{\partial x_j} \right) - \right. \\ &\left. 2 \frac{1}{|\mathbf{S}|^4} \left(\frac{\partial \mathbf{S}}{\partial x_i}, \frac{\partial \mathbf{S}}{\partial x_j} \right) + \frac{\sigma}{n} \frac{\partial \sigma}{\partial x_i} \frac{\partial \sigma}{\partial x_j} \frac{1}{(\sqrt{\sigma^2 + 4\delta^2})^3} \right]. \end{aligned} \quad (29)$$

Acknowledgements

This work has been partially supported by the Spanish Government and FEDER under grant contracts: CGL2011-29396-C03-01 and DPI2011-23058, and by CONACYT-SENER ('Fondo Sectorial CONACYT SENER HIDROCARBUROS' grant contract: 163723).

References

- [1] M. Whiteley, D.R. White, S. Benzley, T.D. Blacker, “Two and three-quarter dimensional meshing facilitators”, *Eng Comput*, 12(3-4): 144–154, 1996.
- [2] E. Ruiz-Gironés, J. Sarrate, “Generation of structured hexahedral meshes in volumes with holes”, *Finite Elements in Analysis and Design*, 46(10): 792 – 804, 2010.
- [3] M.L. Staten, S.A. Canann, S.J. Owen, “BMSweep: locating interior nodes during sweeping”, *Eng Comput*, 15: 212–218, 1999.
- [4] D.R. White, S. Saigal, S.J. Owen, “CCSweep: an automatic decomposition of multi-sweep volumes”, *Eng Comput*, 20: 222–236, 2004.
- [5] X. Roca, J. Sarrate, “An automatic and general least-squares projection procedure for sweep meshing”, *Eng Comput*, 26(4): 391–406, 2010.
- [6] E. Ruiz-Gironés, X. Roca, J. Sarrate, “Using a computational domain and a three-stage node location procedure for multi-sweeping algorithms”, *Adv Eng Softw*, 42(9): 700 – 713, 2011.
- [7] T.J. Tautges, T.D. Blacker, S.A. Mitchell, “The whisker weaving algorithm: a connectivity-based method for constructing all-hexahedral finite element meshes”, *Int J Numer Meth Eng*, 39(19): 3327–3350, 1996.
- [8] X. Roca, J. Sarrate, “Local dual contributions: representing dual surfaces for block meshing”, *Int J Numer Meth Eng*, 83(6): 709–740, 2010.
- [9] R. Schneiders, “A grid-based algorithm for the generation of hexahedral element meshes”, *Eng Comput*, 12(3): 168–177, 1996.
- [10] Y. Zhang, C. Bajaj, “Adaptive and quality quadrilateral/hexahedral meshing from volumetric data”, *Comput Methods Appl Mech Engrg*, 195(9-12): 942–960, 2006.
- [11] J. Qian, Y. Zhang, “Sharp feature preservation in octree-based hexahedral mesh generation for CAD assembly models”, in *Proceedings of the 19th International Meshing Roundtable*, pages 243–262. Springer, 2010.
- [12] T.D. Blacker, R.J. Meyers, “Seams and wedges in Plastering: a 3-D hexahedral mesh generation algorithm”, *Eng Comput*, 9(2): 83–93, 1993.
- [13] M.L. Staten, R.A. Kerr, S.J. Owen, T.D. Blacker, M. Stupazzini, K. Shimada, “Unconstrained plastering-Hexahedral mesh generation via advancing-front geometry decomposition”, *Int J Numer Meth Eng*, 81(2): 135–171, 2010.

- [14] E. Ruiz-Gironés, X. Roca, J. Sarrate, “The receding front method applied to hexahedral mesh generation of exterior domains”, *Eng Comput*, 28: 391–408, 2012.
- [15] S. Meshkat, D. Talmor, “Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh”, *Int J Numer Meth Eng*, 49 (1-2): 17–30, 2000.
- [16] S.J. Owen, S. Saigal, “H-Morph: an indirect approach to advancing front hex meshing”, *Int J Numer Meth Eng*, 49(1-2): 289–312, 2000.
- [17] J.M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, J.M. González-Yuste, “Simultaneous untangling and smoothing of tetrahedral meshes”, *Comput Methods Appl Mech Engrg*, 192(25): 2775–2787, 2003.
- [18] J. Escobar, G. Montero, R. Montenegro, E. Rodríguez, “An algebraic method for smoothing surface triangulations on a local parametric space”, *Int J Numer Meth Eng*, 66(4): 740–760, 2006.
- [19] J.M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, J.M. González-Yuste, “SUS Code - Simultaneous mesh Untangling and Smoothing code”, 2010, URL www.dca.iusiani.ulpgc.es/proyecto2012-2014/html/Software.html.
- [20] P.M. Knupp, “Algebraic mesh quality metrics”, *SIAM J Sci Comput*, 23(1): 193–218, 2001.
- [21] P.M. Knupp, “Algebraic mesh quality metrics for unstructured initial meshes”, *Finite Elements in Analysis and Design*, 39(3): 217–241, 2003.
- [22] P.M. Knupp, “Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I - A framework for surface mesh optimization”, *Int J Numer Meth Eng*, 48: 401–420, 2000.
- [23] P.M. Knupp, “Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II - A framework for volume mesh optimization and the condition number of the Jacobian matrix”, *Int J Numer Meth Eng*, 48: 1165–1185, 2000.
- [24] L.A. Freitag, P.M. Knupp, “Tetrahedral mesh improvement via optimization of the element condition number”, *Int J Numer Meth Eng*, 53: 1377–1391, 2002.
- [25] L.A. Freitag, P. Plassmann, “Local optimization-based simplicial mesh untangling and improvement”, *Int J Numer Meth Eng*, 49: 109–125, 2000.
- [26] P.M. Knupp, “A method for hexahedral mesh shape optimization”, *Int J Numer Meth Eng*, 58(2): 319–332, 2003.

- [27] T. Wilson, J. Sarrate, X. Roca, R. Montenegro, J. Escobar, “Untangling and smoothing of quadrilateral and hexahedral meshes”, in B.H.V. Topping (Editor), *Proceedings of the Eight International Conference on Engineering Computational Technology*. Civil-Comp Press, Stirlingshire, UK, Paper 36, 2012. doi:10.4203/ccp.100.36.
- [28] L. Herrmann, “Laplacian-isoparametric grid generation scheme”, *Int J Numer Meth Eng*, 102(5): 749–756, 1976.
- [29] S. Giuliani, “An algorithm for continuous rezoning of the hydrodynamic grid in arbitrary lagrangian-eulerian computer codes”, *Nucl Eng Des*, 72(2): 205–212, 1982.
- [30] R. Montenegro, J.M. Cascón, J.M. Escobar, E. Rodríguez, G. Montero, “An automatic strategy for adaptive tetrahedral mesh generation”, *Appl Numer Math*, 59: 2203–2217, 2009.
- [31] J.M. Escobar, J.M. Cascón, E. Rodríguez, R. Montenegro, “A new approach to solid modeling with trivariate T-splines based on mesh optimization”, *Comput Methods Appl Mech Engrg*, 200: 3210 – 3222, 2011.
- [32] A. Gargallo-Peiró, X. Roca, J. Sarrate, “A surface mesh smoothing and untangling method independent of the CAD parameterization”, *Computational Mechanics*, pages 1–23, 2013.
- [33] A. Gargallo-Peiró, X. Roca, J. Sarrate, “Defining Quality Measures for High-Order Triangles and Curved Mesh Generation”, in *Proceedings of the 20th International Meshing Roundtable*, 2011.
- [34] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, “Defining Quality Measures For Meshes on Parameterized CAD Surfaces”, in *Proceedings of the 21th International Meshing Roundtable*, 2012.
- [35] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, “Defining quality measures for validation and generation of high-order tetrahedral meshes”, in *Proceedings of the 22th International Meshing Roundtable*, 2013.
- [36] T.J. Wilson, “Simultaneous untangling and smoothing of tetrahedral meshes”, Master’s thesis, ETSE Camins Canals i Ports de Barcelona, Universitat Politècnica de Catalunya, June 2011.
- [37] A. Gargallo-Peiró, “Smoothing and untangling of meshes on parameterized surfaces”, Master’s thesis, Facultat de Matemàtiques, Universitat Politècnica de Catalunya, February 2011.
- [38] J. Nocedal, S. Wright, *Numerical optimization*, Springer Verlag, 1999.

- [39] A.V. Mobley, J.R. Tristano, C.M. Hawkings, “An Object-Oriented Design for Mesh Generation and Operation Algorithms”, in *Proceedings of the 10th Annual International Meshing Roundtable*, pages 7–10. Citeseer, 2001.
- [40] J.F. Remacle, M.S. Shephard, “An algorithm oriented mesh database”, *International Journal for Numerical Methods in Engineering*, 58(2): 349–374, 2003.
- [41] F. Ledoux, J.C. Weill, Y. Bertrand, “GMDS: A Generic Mesh Data Structure”, in *17th International Meshing Roundtable*, 2008.
- [42] D. Rypl, B. Patzák, “Object oriented implementation of the T-spline based isogeometric analysis”, *Advances in Engineering Software*, 50: 137–149, 2012.
- [43] X. Roca, J. Sarrate, E. Ruiz-Gironés, “A graphical modeling and mesh generation environment for simulations based on boundary representation data”, in *Congreso de Métodos Numéricos y Computacionales en Ingeniería*, 2007.
- [44] X. Roca, E. Ruiz-Gironés, J. Sarrate, “ez4u: Mesh generation environment”, <http://www.lacan.upc.edu/ez4u.htm>, 2010.