Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2014) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

23rd International Meshing Roundtable (IMR23)

Optimizing mesh distortion by hierarchical iteration relocation of the nodes on the CAD entities

Eloi Ruiz-Gironés[a,*], Xevi Roca[b], Jose Sarrate[a]

[a] *Laboratori de Càlcul Numèric (LaCàN),*
Departament de Matemàtica Aplicada III (MA III),
Universitat Politècnica de Catalunya (UPC),
Campus Nord UPC, 08034 Barcelona, Spain.
{eloi.ruiz, jose.sarrate}@upc.edu
[b] *Aerospace Computational Design Laboratory,*
Department of Aeronautics and Astronautics,
Massachusetts Institute of Technology, Cambridge, MA 02139, USA.
xeviroca@mit.edu

Abstract

Mesh untangling and smoothing is an important part of the meshing process to obtain high-quality discretizations. The usual approach consists on moving the position of the interior nodes while considering fixed the position of the boundary ones. However, the boundary nodes may constrain the quality of the whole mesh, and high-quality elements may not be generated. Specifically, thin regions in the geometry or special configurations of the boundary edges may induce low-quality elements. To overcome this drawback, we present a smoothing and untangling procedure that moves the interior nodes as well as the boundary ones, via an optimization process. The objective function is defined as a regularized distortion of the elements, and takes the nodal Cartesian coordinates as input arguments. When dealing with surface and edge nodes, the objective function uses the nodal parametric coordinates in order to avoid projecting them to the boundary. The novelty of the approach is that we consider a single target objective function (mesh distortion) where all the nodes, except the vertex nodes, are free to move on the corresponding CAD entity. Although the objective function is defined globally, for implementation purposes we propose to perform a node-by-node process. To minimize the objective function, we consider a block iterated non-linear Gauss-Seidel method using a hierarchical approach. That is, we first smooth the edge nodes, then the face nodes, and finally the inner nodes. This process is iterated using a node-by-node Gauss-Seidel approach until convergence is achieved.

Keywords: Mesh smoothing; Mesh untangling; Algebraic quality; Function optimization; Hierarchical approach

* Corresponding author. Tel.: +34 93 401 18 42 ; fax: +34 93 401 1825.
E-mail address: eloi.ruiz@upc.edu

1. Introduction

It is well known that the accuracy of a finite element method calculation depends on the mesh quality of the domain discretization. For this reason, mesh untangling and smoothing is a key process that improves the element quality. Hence, several algorithms have been developed to improve the mesh quality. One of the most versatile methods consists on defining a quality measure function [1–3] that takes the nodal coordinates as arguments. Then, an optimization algorithm is applied to compute the *best* configuration of nodes that maximizes the element quality. Different optimization strategies can be applied, such as performing a global optimization problem, in which all nodes are moved at the same time, or iteratively move one node at a time, until convergence is achieved.

Typically, when a smoothing algorithm is applied, the boundary nodes are considered fixed, and the method obtains the *best* position of the interior nodes to improve the mesh quality. However, the boundary mesh may constrain the quality of the whole mesh and low-quality elements could be obtained. For instance, specific configurations of the boundary edges or thin regions in the volume may hinder the task of obtaining a high-quality mesh. Several works deal with improving the quality of the boundary mesh, composed of triangles or quadrilaterals, restricting the position of the nodes on the boundary surfaces, either by projecting the nodes to the geometrical entities, [4–8], or by using the parametric space, [9–13]. Then, when a high-quality boundary mesh is obtained, the algorithm to smooth the inner nodes is applied. However, this approach may not obtain the volume mesh of optimal quality since it takes into account two distortion measures: one distortion measure for the two-dimensional boundary nodes, and one distortion measure for the three-dimensional inner nodes. In addition, low-quality elements may be obtained due to the position of edge nodes, which are not considered in this approach.

To overcome this issue, we propose a hierarchical smoothing and untangling process based on minimizing a global objective function defined in terms of the distortion of the three dimensional elements, that takes the nodal coordinates as arguments [13,14]. The novelty of the approach is that we consider a single target objective function (mesh distortion) where all the nodes, except the vertex nodes, are free to move on the corresponding container CAD entity. While the objective function is globally defined for all the nodes, we propose to optimize it using a hierarchical approach in three stages. First, we improve the quality of the three-dimensional elements moving the edge nodes along the curves. Second, we improve the quality of the three-dimensional elements moving the face nodes on the surfaces. Finally, we improve the quality of the elements moving the interior nodes. During the first two steps of the proposed method, we have to ensure that the boundary nodes are located on the corresponding edges and faces. To this end, we consider the parameterization of edges and faces provided by the CAD model. Specifically, the unknowns that correspond to the edge and surface nodes are determined by their parametric coordinates on the CAD entities. In this manner, we can avoid expensive and non-robust projections of the nodes on the CAD entities. The main advantage of the proposed approach is that the location of the edge and surface nodes is not fixed. This results in an optimization process that is less constrained than the standard approach (fixed boundary nodes) and therefore, a better configuration of nodes can be achieved. It is important to point out that the node type is assigned while the mesh is generated hierarchically. That is, when the curve, surface and volume meshes are generated, the corresponding nodes are marked with the appropriate type.

The remaining of the paper is structured as follows. In Section 2, we review the existing literature related to the presented work. In Section 3, we present a simple two-dimensional example to illustrate this work. In Section 4, we review the concept of an algebraic shape distortion measure for two and three dimensional elements. In Section 5, we present the proposed hierarchical smoothing. Then, in Section 6, we present several examples in order to show that the proposed method is able to untangle inverted elements and improve the overall mesh quality. Finally, in Section 7, we present the concluding remarks and the future work.

2. Related work

In [1–3], the authors introduced a framework to define algebraic element quality and distortion in terms of the Jacobian of the mapping between an ideal and a physical element. Using these distortion measures, several authors proposed to perform an optimization process to compute the optimal node position that minimize the element distortion, see [3,15–18]. The minimization process can be performed globally or based on a local approach. In the global approach, an objective function 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.

The objective functions introduced by Knupp present asymptotes (barriers) when inverted elements appear in the mesh. For this reason, when tangled elements are present, the optimization process is not able to recover a valid configuration of the nodes to obtain untangled elements. To overcome this drawback, in [3,17], a two-stage smoothing process is proposed. In the first stage, an objective function to untangle the inverted elements is used. In the second stage, an objective function to improve the quality is used. That is, the optimization process is driven by two objective functions: the first one to untangle the mesh, and the second one to smooth the mesh. Other authors used the log-barrier method in order to avoid tangled elements, see [19–21]. In reference [22,23], the authors proposed a regularization in the objective function in order to avoid the singularities. Thus, a simultaneous untangling and smoothing process is performed by using only the regularized objective function. Note that in our smoothing method, we use this approach to avoid the barriers in the objective function.

Several algorithms deal with mesh smoothing on parameterized surfaces. These algorithms could be classified as indirect or direct. The indirect algorithms first compute the ideal position of a node and then they project back the node to the surface, see [4–8]. On the other hand, direct methods express the objective function in terms of the parametric coordinates of the nodes [9–13]. Thus, there is no need to project the nodes back to the surface. Note that these techniques are usually performed to improve the quality of a volume mesh enclosed by a boundary mesh. First, a smoothing of the boundary mesh is performed and then, a smoothing of the interior nodes is performed to improve the quality of the whole mesh. Note that two different objective functions are used: one for the boundary nodes, and another different one for the interior nodes. The former is expressed in terms of the distortion of the two-dimensional boundary elements, and the later is expressed in terms of the distortion of three-dimensional elements of the volume.

Different methods have been proposed to improve the quality of three-dimensional elements by moving the inner and boundary nodes. Note that the main issue is to impose that the boundary nodes should approximate the geometry boundary. To this end, it is possible to include a penalty term in the objective function that enforces that the corresponding nodes move close to the boundary [24]. It is also possible to impose that the nodes are strictly on the geometry boundary. In [25] the authors proposed a mesh quality improvement method that relocates the volume and surface nodes, but not the curve nodes. In some applications, such as curved high-order mesh generation or structured meshing, it could be also required to move the curve nodes. In [20] a method to smooth and untangle curved high-order meshes using the parametric coordinates of the surface and curve nodes was proposed. Specifically, they minimize an objective function that penalizes both large deformations and small values of the determinant of the iso-parametric mapping. In reference [26], the authors showed a smoothing method to increase the quality of iso-geometric meshes by moving the nodes on edges and surfaces. In our work, we also propose to move the nodes on the surface and edge nodes based on the parametric coordinates. The difference is that the goal in our proposed optimization problem is to minimize a regularized mesh distortion.

3. Motivation

To highlight the importance of moving boundary nodes during a smoothing and untangling process we use a simple two-dimensional quadrilateral mesh. Figure 1(a) presents the initial mesh generated using the submapping method. Note that the initial geometry is composed of four edges. Due to the curvature of the edges, when they are discretized using intervals of the same length, the inner mesh contains inverted elements. When we apply a smoothing technique to improve the mesh quality, the method is not able to obtain a high-quality mesh because the boundary nodes are fixed, see Fig. 1(b). The objective of this work is to develop a smoother that is able to move the interior nodes as well as the boundary nodes according to a distortion measure. Using this procedure, a high-quality mesh can be generated, even when the boundary discretization constrains the quality of the whole mesh, see Fig. 1(c).

4. Algebraic distortion and quality measures

In reference [1], the authors propose an algebraic framework to compute the quality and distortion of a given element. Given a physical element, e, in the physical domain, and an ideal element, e_I, that represents the desired shape, there exists a transformation, $\boldsymbol{\phi}$, such that the ideal element is transformed to the physical element. Since the

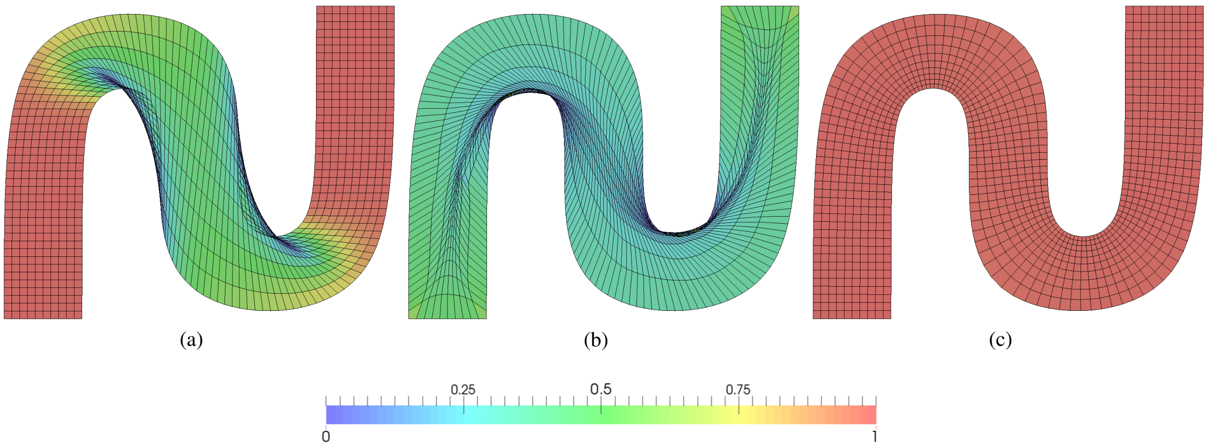

Fig. 1. Simple quadrilateral mesh generated using the submapping method: (a) original mesh; (b) smoothing interior nodes; and (c) smoothing interior and boundary nodes.

mapping $\boldsymbol{\phi}$ can be difficvult to compute, a master element, e_M, and a transformation from e_M to e_I, denoted as $\boldsymbol{\phi}_I$, are considered. Note that $\boldsymbol{\phi} = \boldsymbol{\phi}_P \circ \boldsymbol{\phi}_I^{-1}$, where $\boldsymbol{\phi}_P$ transforms the master element, e_M, to the physical element, e_P. Note that we use a master element, e_M, such that the mappings $\boldsymbol{\phi}_I$ and $\boldsymbol{\phi}_P$ are easy to define. For instance, in the case of iso-parametric elements,

$$\boldsymbol{\phi}_P(\boldsymbol{\xi}) = \sum_{i=1}^{n_p} \mathbf{x}_i N_i(\boldsymbol{\xi}), \quad \boldsymbol{\phi}_I(\boldsymbol{\xi}) = \sum_{i=1}^{n_p} \mathbf{y}_i N_i(\boldsymbol{\xi}),$$

where n_p is the number of element nodes, $\mathbf{x}_i$ and $\mathbf{y}_i$, for $i = 1, \ldots, n_p$ are the coordinates of the nodes of the physical and ideal element, respectively, and $N_i(\boldsymbol{\xi})$ are a base of shape functions in the master element.

The point-wise distortion of element e is measured in terms of the Jacobian, $\mathbf{D}\boldsymbol{\phi}$, as:

$$\eta(\mathbf{y}) = \frac{|\mathbf{D}\boldsymbol{\phi}(\mathbf{y})|^2}{n\sigma(\mathbf{D}\boldsymbol{\phi}(\mathbf{y}))^{2/n}}, \tag{1}$$

where $\sigma(\cdot)$ is the determinant, n is the space dimension and $|\cdot| = \sqrt{(\cdot,\cdot)}$ is the Frobenius norm, being $(\cdot,\cdot)$ a dot product for matrices, defined as

$$(\mathbf{A}, \mathbf{B}) = \mathrm{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{B}).$$

Note that while $\boldsymbol{\phi}$ is an affine transformation for linear triangles and tetrahedra, it is not the case for quadrilaterals and hexahedra. In the general case, $\mathbf{D}\boldsymbol{\phi}$ can be computed as:

$$\mathbf{D}\boldsymbol{\phi}(\mathbf{y}) = \mathbf{D}\boldsymbol{\phi}_P(\boldsymbol{\phi}_I^{-1}(\mathbf{y})) \cdot \mathbf{D}\boldsymbol{\phi}_I^{-1}(\mathbf{y}).$$

According to [1], the point-wise quality of an element is the inverse of the point-wise distortion. That is:

$$\mathrm{q}(\mathbf{y}) = \frac{1}{\eta(\mathbf{y})}.$$

The shape distortion measure can be used to improve the mesh quality by means of a minimization process, see [3,15–18]. However, the shape distortion measure presents asymptotes when $\sigma(\mathbf{D}\boldsymbol{\phi}) = 0$. This prevents the use of this distortion measure in a continuous optimization procedure. To overcome this drawback, in reference [22] a regularization of Eqn. (1) is introduced as

$$\eta^*(\mathbf{y}) = \frac{|\mathbf{D}\boldsymbol{\phi}(\mathbf{y})|^2}{nh(\sigma(\mathbf{D}\boldsymbol{\phi}(\mathbf{y})))^{2/n}}, \tag{2}$$

where

$$h(\sigma) = \frac{1}{2}\left(\sigma + \sqrt{\sigma^2 + 4\delta^2}\right). \tag{3}$$

The δ parameter is defined as a small parameter, and its value depends on the problem. For a further analysis on the selection of δ, see [4,13,22,27,28].

Let $\mathcal{M}$ be the physical mesh and $\mathcal{M}_I$ be the set of ideal elements. A scalar product of functions on $\mathcal{M}_I$ is defined as:

$$\langle f, g \rangle_{\mathcal{M}_I} = \int_{\mathcal{M}_I} f(\mathbf{y})g(\mathbf{y})\mathrm{d}\mathbf{y} = \sum_{i=1}^{n_e} \langle f, g \rangle_{e_I},$$

where n_e is the number of elements in the mesh, and

$$\langle f, g \rangle_{e_I} = \int_{e_I} f(\mathbf{y})g(\mathbf{y})\mathrm{d}\mathbf{y}.$$

Since $\boldsymbol{\phi}_I$ is a diffeomorphism, each elemental integral can be computed in the corresponding master element as:

$$\langle f, g \rangle_{e_I} = \int_{e_M} f(\boldsymbol{\phi}_I(\boldsymbol{\xi}))g(\boldsymbol{\phi}_I(\boldsymbol{\xi})) \det \mathbf{D}\boldsymbol{\phi}_I(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi}.$$

Given the point-wise element distortion measure, Eqn. (2), the element distortion, $\eta_e^*(\mathbf{x}_1, \ldots, \mathbf{x}_{n_p})$, is computed as

$$\eta_e^*(\mathbf{x}_1, \ldots, \mathbf{x}_{n_p}) = \frac{\|\eta_e^*(\mathbf{y})\|_{e_I}}{\|1\|_{e_I}}. \tag{4}$$

Note that the point-wise element distortion, $\eta^*(\mathbf{y})$, is a function of $\mathbf{y} \in e_I$, but it also depends on $\mathbf{x}_i$, for $i = 1, \ldots, n_p$. For this reason, the element distortion, η_e^*, only depends on the coordinates of the nodes, $\mathbf{x}_i$, for $i = 1, \ldots, n_p$. Note that the element quality, q_e, is defined as:

$$\mathrm{q}_e = \frac{1}{\eta_e^*}.$$

5. Simultaneous untangling and smoothing

In this section, we detail the optimization approach to simultaneously untangle and smooth a mesh. First, we define an objective function according to the point-wise shape distortion measure (2). Then, this objective function is adapted to obtain the quality of the elements when there are some nodes that belong to an edge or a face. Using the distortion measures, we will obtain a new node position that minimize the distortion of the elements.

5.1. Objective function

5.1.1. Base objective function

In this section, we will define an objective function that is evaluated on the position of the nodes in the physical space. The objective function takes into account the parameterization of the boundary curves and surfaces in order to untangle the mesh and improve the quality of the elements, even those that have nodes on the boundary of the geometric model. According to Eqn. (2), the ideal mesh satisfies

$$\eta^* = 1, \qquad \forall \mathbf{y} \in \mathcal{M}_I.$$

However, since this condition cannot be accomplished, we impose it in a least squares sense as:

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_{n_f}; \mathbf{p}_1, \ldots, \mathbf{p}_{n_p}) = \frac{1}{2}\|\eta^*(\mathbf{D}\boldsymbol{\phi}) - 1\|^2_{\mathcal{M}_I}, \tag{5}$$

where n_f is the number of free nodes (nodes that are not fixed), and n_p is the number of fixed nodes. In our case, the only fixed nodes are the ones that belong to the vertices of the geometry.

We need to impose that the boundary nodes in the smoothed mesh belong to the boundary. To avoid non-robust and expensive projections from the computational point of view, we describe the objective function (5) in terms of the parametric coordinates of the boundary nodes. That is:

$$\begin{aligned} \mathbf{x}_j &= \mathbf{x}(\mathbf{u}_j), \qquad j = 1, \dots, n_s \\ \mathbf{x}_k &= \mathbf{x}(t_k), \qquad k = 1, \dots, n_c, \end{aligned}$$

where n_s and n_c are the number of mesh points on the surfaces and curves of the geometry, respectively, and $\mathbf{u}_j$ and t_k are the parametric coordinates of surface and edge nodes, respectively. Using the parametric coordinates, the global objective function can be expressed as:

$$f(\mathbf{x}_1, \dots, \mathbf{x}_{n_i}, \mathbf{u}_i, \dots, \mathbf{u}_{n_s}, t_1, \dots, t_{n_c}; \mathbf{p}_1, \dots, \mathbf{p}_{n_p}) = \frac{1}{2} \|\eta^*(\mathbf{D}\boldsymbol{\phi}) - 1\|^2_{\mathcal{M}_I}, \tag{6}$$

where n_i is the number of inner nodes, and $n_i + n_s + n_c = n_f$.

5.1.2. Interior nodes objective function

Note that we have deduced an objective function that depends on the coordinates of all the free mesh nodes (global). Since we use nodal elements, we can deduce an objective function that only depends on the coordinates of one specific node (local). Thus, the local objective function allows the implementation of a non-linear Gauss-Seidel procedure. The local objective function for a single node, v, is defined as the addition of the elemental contributions of the adjacent elements:

$$f_v = \sum_{e \in \mathcal{M}_v} \frac{1}{2} \|\eta^*(\mathbf{D}\boldsymbol{\phi}_e) - 1\|^2_{e_I}, \tag{7}$$

where $\mathcal{M}_v$ is the local sub-mesh that contains node v, and $\boldsymbol{\phi}_e$ is the mapping between the ideal element, e_I, and the physical element, e.

5.1.3. Edge nodes objective function

We need to impose that the edge nodes in the smoothed mesh are located on the boundary edges. To this end, we rewrite the objective function (7) in terms of the parametric coordinates of the boundary nodes. Given a parametric curve, $\gamma(t)$, and a node on the curve, v, the corresponding objective function is expressed as the composition of functions:

$$f_\gamma(t) = f_v(\gamma(t)) \tag{8}$$

The derivatives of the objective function (8) with respect to the parametric coordinate t can be expressed in terms of Function (7) and the curve parameterization, $\gamma(t)$, by means of the chain rule. That is,

$$\nabla f_\gamma(t) = \nabla f_v(\gamma(t)) \cdot \gamma(t)', \tag{9}$$

and

$$H f_\gamma(t) = (\gamma')^{\mathrm{T}} \cdot \nabla f_v(\gamma(t)) \cdot \gamma' + \nabla f_v(\gamma(t)) \cdot \gamma''. \tag{10}$$

The derivatives of the curve parameterization, γ, are provided by the CAD engine. In our applications we use Open-Cascade as the geometric engine [29]. We only need to compute the derivatives of the objective function (7) to restrict the movement of a node on a curve. For this reason, given a generic objective function, it is straightforward to obtain a new one for a node that belongs to a curve.

5.1.4. Face nodes objective function

Given a node, v, on a surface, we define the corresponding objective function by composing (7) with the surface parameterization, φ. That is:

$$f_\varphi(\mathbf{u}) = f_v(\varphi(\mathbf{u})). \tag{11}$$

The derivatives of the objective function (11) with respect to coordinates $\mathbf{u}$ are deduced by means of the chain rule. That is,

$$\nabla f_\varphi(\mathbf{u}) = \mathbf{D}\varphi(\mathbf{u}) \cdot \nabla f_v(\varphi(\mathbf{u})), \tag{12}$$

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

1: **function** `smoothMesh`(Mesh $\mathcal{M}$, Real $maxDisp$)
2:     Boolean $isConverged \leftarrow$ false
3:     **while not** $isConverged$ **do**
4:         Real $disp \leftarrow 0$
5:         **for all** $v \in \mathcal{E}$ **do** ▷ Edge nodes
6:             Function $f_\gamma \leftarrow$ `getCurveFuncion` ▷ Eq. (8)
7:             Real $nodeDisp \leftarrow$ `smoothNode`(v,f_γ)
8:             $disp \leftarrow \max\{disp, nodeDisp\}$
9:         **end for**
10:         **for all** $v \in \mathcal{F}$ **do** ▷ Face nodes
11:             Function $f_\varphi \leftarrow$ `getSurfaceFuncion` ▷ Eq. (11)
12:             Real $nodeDisp \leftarrow$ `smoothNode`(v,f_φ)
13:             $disp \leftarrow \max\{disp, nodeDisp\}$
14:         **end for**
15:         **for all** $v \in \mathcal{V}$ **do** ▷ Volume nodes
16:             Function $f_v \leftarrow$ `getBaseFuncion` ▷ Eq. (7)
17:             Real $nodeDisp \leftarrow$ `smoothNode`(v,f_v)
18:             $disp \leftarrow \max\{disp, nodeDisp\}$
19:         **end for**
20:         $isConverged \leftarrow (disp \le maxDisp)$
21:     **end while**
22: **end function**

and

$$Hf_\varphi(\mathbf{u}) = H\varphi(\mathbf{u}) : \nabla f_v(\varphi(\mathbf{u})) + (\mathbf{D}\varphi(\mathbf{u}))^{\mathrm{T}} \cdot \nabla f_v(\varphi(\mathbf{u})) \cdot \mathbf{D}\varphi(\mathbf{u}), \tag{13}$$

where $H\varphi(\mathbf{u})$ is a third order tensor whose components are defined as

$$\left[H\varphi(\mathbf{u})\right]_{ijk} = \frac{\partial^2 \varphi_k(\mathbf{u})}{\partial u_i \partial u_j},$$

and

$$\left[H\varphi(\mathbf{u}) : \nabla f_v(\varphi(\mathbf{u}))\right]_{ij} = \sum_{k=1}^{n} \frac{\partial f(\varphi(\mathbf{u}))}{\partial x_k} \cdot \frac{\partial^2 \varphi_k(\mathbf{u})}{\partial u_i \partial u_j}.$$

Note that we only need to compute the derivatives of the original objective function, because the derivatives of the surface parameterization are provided by the CAD engine.

5.2. Optimization approach

In this section, we detail the optimization approach to obtain the new position of the nodes that untangles the mesh and improves the element quality. To this end, we propose an iterative process to minimize the objective function (7), one node at a time, see Algorithm 1. The nodes are divided in three sets, according to their location. First, we smooth the edge nodes, prescribing the position of all the other nodes, Lines 5–9. Second, we smooth the face nodes by assuming that the positions of the remaining nodes are fixed, Lines 10–14. Third, we smooth the interior nodes fixing the position of all the other nodes, 15–19. For each node, v, the appropriate objective function is used, depending on the location of the node. The process is iterated until the maximum displacement of the nodes is below a threshold prescribed by the user, Line 20.

The new position of a node is computed using a line-search iterative process. Let $\mathbf{x}_v^k$ be the position of node v at iteration k. The new position at iteration $k+1$ is defined as

$$\mathbf{x}_v^{k+1} = \mathbf{x}_v^k + \alpha^k \mathbf{p}^k, \tag{14}$$

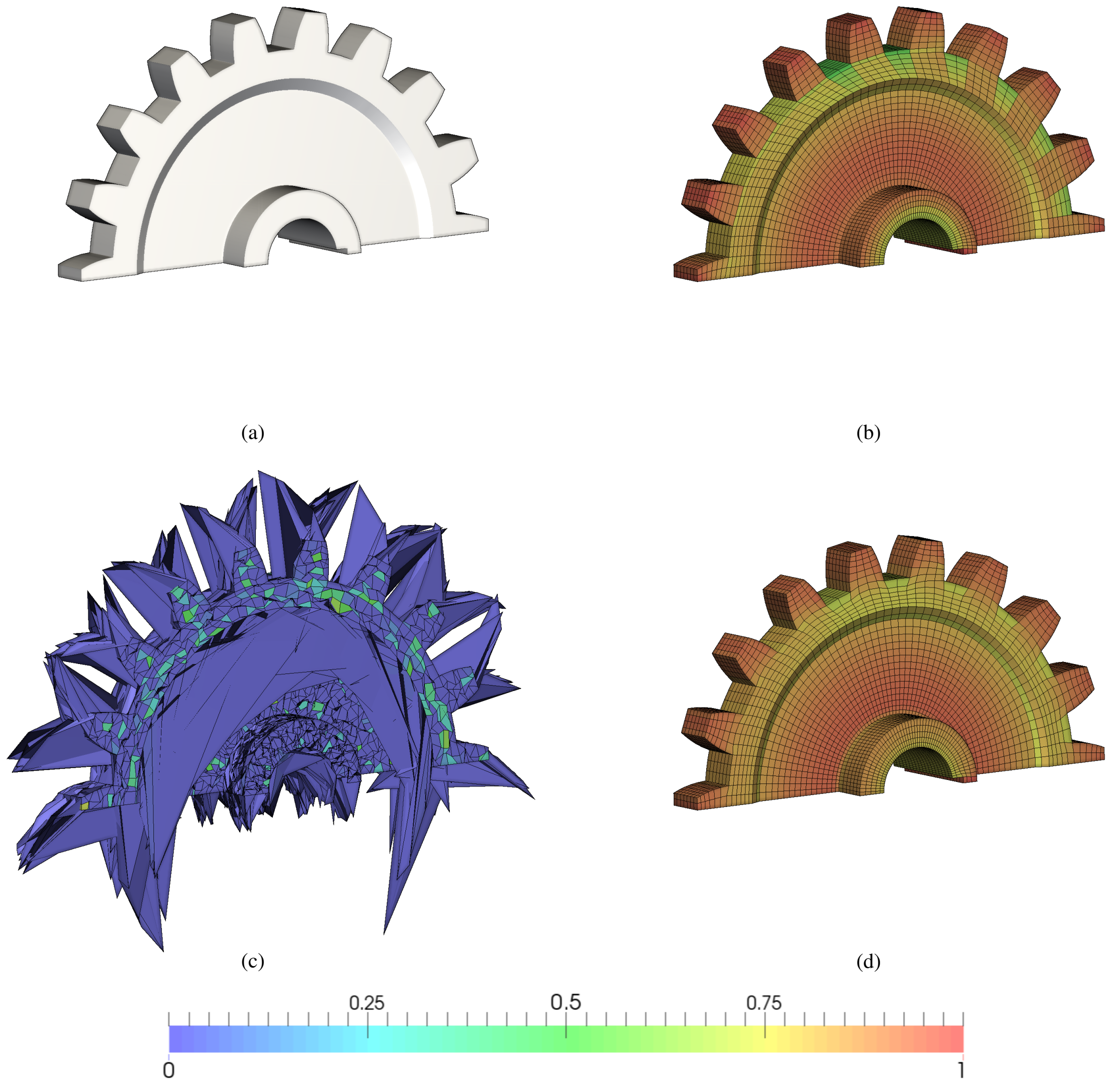

Fig. 2. Hexahedral mesh optimization of the mesh generated for the half of a gear: (a) geometry definition; (b) mesh before applying the hierarchical smoothing; (c) mesh with randomized node position; and (d) mesh after applying the hierarchical smoothing.

where $\mathbf{p}^k$ is an advancing direction and α^k is a step length. To compute both the advancing direction and the step length, there are several methods, like the steepest descent or the Newton-Raphson methods, see [30]. Note that to apply this algorithm, the first and the second derivatives of the objective function are needed.

6. Examples

In this section, we present several examples that illustrate the advantages of the proposed method to untangle and smooth hexahedral and tetrahedral meshes. The first example correspond to a hexahedral mesh, the second example shows a tetrahedral mesh, and the third example presents an high-order hexahedral mesh. For each example we show the original mesh and the smoothed mesh, and we present a table summarizing the shape quality statistics of the mesh elements. Specifically, we provide: the number of tangled elements, the minimum, the maximum, the

Table 1. Element quality statistics of the linear meshes for the half of a gear and the propeller.

	Gear mesh			Propeller mesh	
	original	randomized	smoothed	initial	smoothed
elements	15629	15629	15629	1087164	1087164
inverted elem.	0	13585	0	0	0
minimum	0.50	0	0.70	0.21	0.25
maximum	0.98	0.75	0.97	1.0	0.99
mean	0.85	0.03	0.85	0.80	0.83
deviation	0.07	0.09	0.06	0.12	0.10

mean and the standard deviation of the mesh quality. We highlight that in all cases the smoothed mesh increases the minimum and mean values of the mesh quality and decreases its standard deviation. Note that the proposed objective function corresponds to an $\mathcal{L}^2$-norm and therefore, it improves the average element distortion. The objective function penalizes, by construction, the highly distorted elements. This is in agreement with the obtained quality statistics, the average and minimum quality (maximum distortion) increase (decrease). However, the maximum quality (minimum distortion) decreases (increases) to accommodate the quality (distortion) improvement of the whole mesh.

6.1. Linear meshes

6.1.1. Hexahedral mesh for a gear

In this example, we apply the proposed hierarchical method to improve the quality of a hexahedral mesh generated for the half of a gear using the submapping method [31,32], see Fig. 2(b). Due to the solution of the integer problem that leads to the number of edge intervals during the mesh generation process, some skewness is introduced in the mesh. Before applying the proposed smoothing algorithm, a high-tangled mesh is obtained by moving the nodes randomly, Fig. 2(c). Note that the only fixed nodes are the ones located at the vertices of the original geometry. All the other nodes have been randomly moved and, for this reason, a large number of inverted elements is obtained. Figure 2(d) presents the mesh after applying the hierarchical smoother to the randomized mesh. The final mesh does not contain inverted elements and, in addition, the smoothing algorithm is able to reduce the skewness of the original mesh that is introduced by the submapping method.

Table 1 details the mesh quality statistics for the three meshes. The original mesh does not contain any inverted elements. The randomized mesh contains 13585 inverted elements, and the hierarchical smoother is able to obtain a mesh without inverted elements. The mean and maximum qualities are almost equal for the original and the smoothed mesh, but the minimum quality of the smoothed mesh is higher, and the deviation is lower.

6.1.2. Tetrahedral mesh for a propeller

In the second example, we use the proposed method to improve the quality of a tetrahedral mesh generated for a propeller. In this case, the geometry contains close surfaces that constrains the quality of the mesh in those thin regions, see Fig. 3(a). The initial mesh is shown in Fig. 3(b), which contains over a million elements. The interior nodes of this mesh are optimized without moving the boundary nodes. Next, we apply the proposed hierarchical smoothing to the initial mesh and a higher quality mesh is obtained, see Fig. 3(c).

Table 1 presents the element quality statistics for the initial mesh and the smoothed mesh using the proposed hierarchical method. The minimum quality is lower in the initial mesh, as well as the mean quality. The maximum quality is similar in both meshes, and, as expected, the deviation is lower in the mesh smoothed using the hierarchical method.

6.2. High-order hexahedral mesh

In this example, we illustrate the applicability of the hierarchical smoother for high-order hexahedral meshes. We first generate a hexahedral mesh for a simple geometry, and then applying the high-order version of the hierarchical smoothing we generate the high-order mesh. Figure 4(a) shows the geometry definition, and Fig. 4(b) shows the linear mesh generated using the submapping method. Since the edge nodes are equi-distributed along the geometry

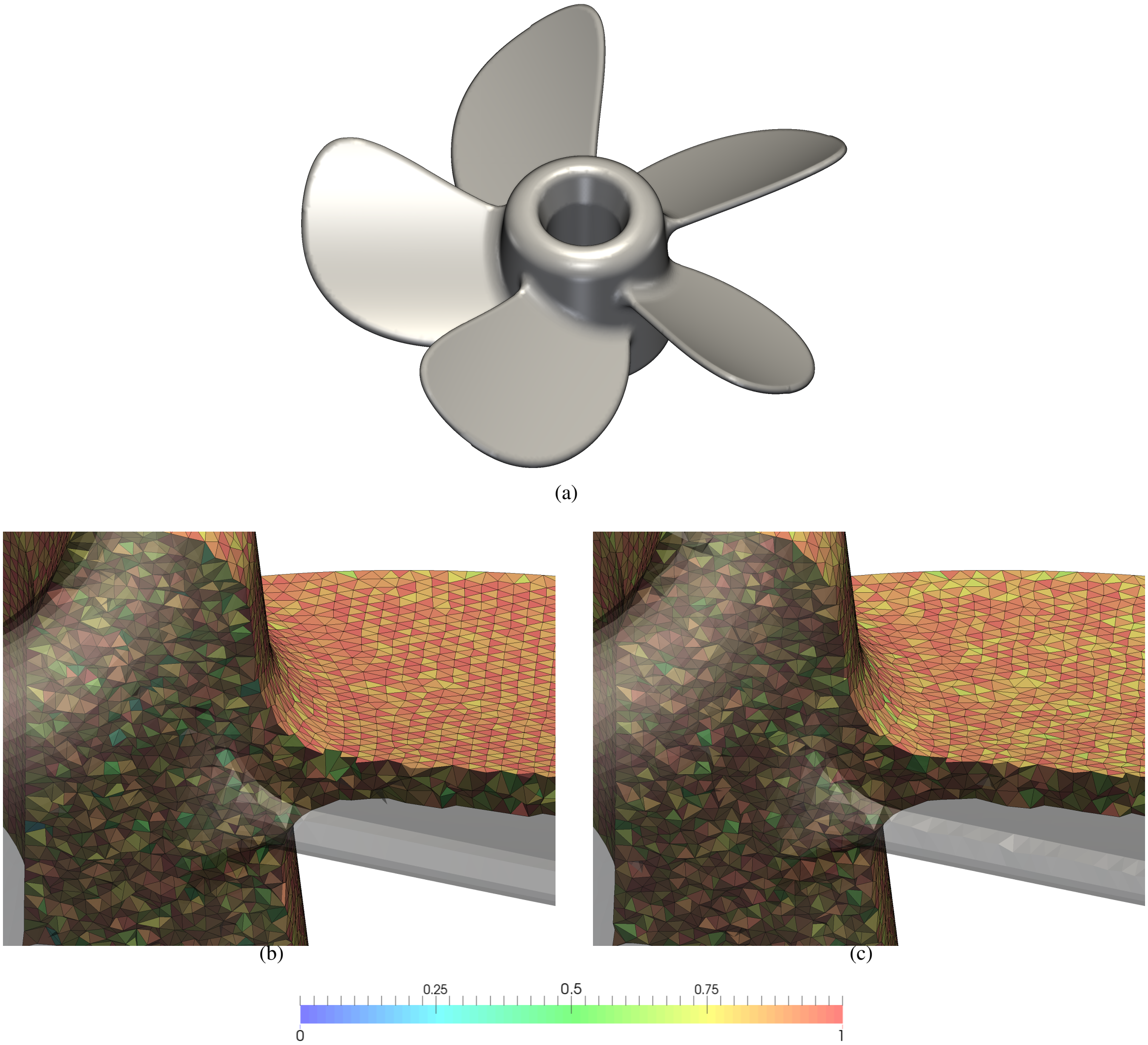

Fig. 3. Tetrahedral mesh optimization of the mesh generated for a propeller: (a) geometry definition; (b) mesh before applying the hierarchical smoothing; and (c) mesh after applying the hierarchical smoothing.

edges, a mesh with a large number of inverted elements is generated. The application of the hierarchical smoother leads to a high-quality mesh without any inverted elements. Then, the interpolation degree of the optimized mesh is increased to $p = 4$, see Fig. 4(c). Finally, Fig. 4(d) shows the mesh obtained after applying the high-order version of the hierarchical smoothing. Note that a high-quality mesh that follows the curvature of the geometry is obtained.

Table 2 presents the element quality statistics for the linear and high-order meshes generated in this example. The original linear mesh contains 104 elements of which twelve are inverted. Thus, the minimum quality is zero. When the linear version of hierarchical smoother is applied, we obtain a mesh without any inverted elements. The maximum quality is about the same as the original mesh, but the minimum and mean quality has increased, and the deviation is decreased. The high-order mesh obtained from the smoothed linear mesh does not contain any inverted elements. In addition, the high-order mesh presents higher minimum and mean quality than the linear one, while that maximum quality and the deviation remain the same. When we apply the high-order version of the hierarchical smoother, the element quality is better. That is, the minimum, the maximum and the mean quality have all increased.

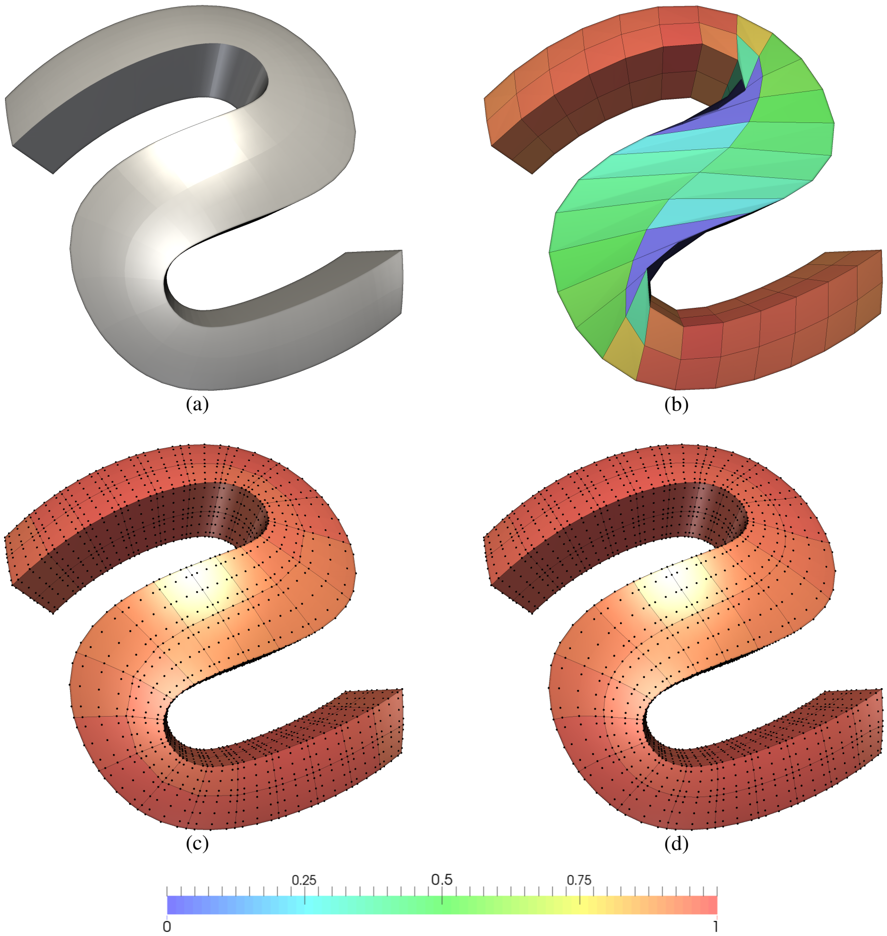

Fig. 4. High-order hexahedral mesh ($p = 4$) optimization for a toy geometry: (a) geometry definition; (b) initial linear mesh; (c) high-order mesh obtained from the smoothed linear mesh using the hierarchical smoothing; and (d) high-order mesh obtained using the high-order hierarchical smoothing.

Table 2. Element quality statistics of the linear and high-order hex-meshes.

	Linear mesh		High-order mesh	
	original	smoothed	from smoothed linear mesh	after high-order smoother
elements	104	104	104	104
inverted elem.	12	0	0	0
minimum	0	0.82	0.89	0.91
maximum	0.98	0.97	0.98	0.98
mean	0.66	0.90	0.95	0.96
deviation	0.33	0.04	0.02	0.02

7. Conclusions

In this work, we have presented a novel technique to hierarchically untangle and smooth meshes composed by triangles, quadrilaterals for 2D applications and tetrahedra or hexahedra for 3D applications. The untangling and smoothing process is accomplished by optimizing a single objective function defined in terms of a regularized distortion measure of the elements. The objective function is defined globally and takes the nodal coordinates as input arguments. The main contribution is that the boundary nodes also contribute to the value of the objective function using the parametric coordinates of the geometric entity where they belong. In this way, the boundary nodes can also be moved during the optimization process. This results in meshes with better quality, specially when the boundary of the mesh constrains the element quality of the whole mesh, like thin regions of the geometry. Contrary to other approaches that use two separate smoothing processes, one for the boundary mesh that takes into account a distortion measure for two-dimensional elements, and another one for the interior nodes that takes into account a distortion measure for three-dimensional elements, we use a single objective function that only takes into account the distortion of the three-dimensional elements. To ensure that boundary nodes are placed on its corresponding geometrical entity, we restrict this objective function using the parameterization of the boundary entities.

Although we have deduced a global objective function, for implementation purposes we have adopted a local optimization approach. That is, instead of moving all the nodes at the same time, we perform a node-by-node iteration until convergence is achieved. This allows to perform a hierarchical smoothing approach which is divided into three stages. First, we smooth the edge nodes; second, we improve the location of face nodes; and third, we smooth the inner nodes. During each stage, we consider the quality of the adjacent three-dimensional elements and, for this reason, we only use one element quality function during the whole process. Since we did not prove that the proposed objective function is convex, there is no convergence guarantee. In practical applications, we use backtracking line search to improve the global convergence features of the implementation. It is important to point out that with all the tested examples we have converged to a local minima that provides a valid mesh, and we did not experience non-convergent or oscillatory behavior during the optimization process. It is also worth noticing that the time to smooth the example meshes is of the order of the minutes.

The algorithm in the current form requires the utilization of CAD engine that provides the evaluation of parameterizations and their first and second derivatives. The method could be modified to deal with non-parameterized representations of the geometry, e.g. piece-wise linear representations. In this case, it could be required to obtain local parameterizations that provide the required derivatives for the optimization method of choice.

Several aspects of the algorithm can be extended in the near future. The proposed approach is an optimization based method with untangling capabilities constrained to move the nodes on the CAD entities. Hence, it requires more floating point operations per node than other methods, e.g. Laplacian smoothing. It would be interesting to explore additional techniques to reduce the computational cost of smoothing a given mesh. For instance, we could use a parallel version of the presented algorithm to reduce the time to smooth a mesh. With this objective in mind, we need to apply a partition algorithm to the mesh nodes in order to avoid that different processors access and modify the same node position at the same time. While we have applied the hierarchical smoother to linear meshes, we have shown that it can also be applied to high-order meshes. For this reason, in the near future we will further improve the proposed algorithm to better handle this kind of meshes. In addition, we have chosen to implement the global function in a node-by-node manner to simplify the solver implementation, reduce the memory footprint, and exploit the local behavior of the objective function. However, we need to explore additional minimization approaches in order to investigate the robustness and the performance of different solvers. With this objective in mind, one of the most interesting approaches is to consider a global solver, in which all the nodes are moved at the same time. It is important to point out that the global implementation could lead to an ill-conditioned Jacobian matrix since the scales of motion in the curves, surfaces, and volumes are different. This could require to scale the optimization variables to improve the conditioning as proposed in [20]. We have considered to apply the proposed approach in industrial applications where the current approach has a high impact. For instance, domains that present thin regions or surfaces shared by two volumes (assembly models). To deal with these configurations, we have considered to use the same objective function since it will enforce to relocate the shared surface nodes to improve the distortion of the global mesh composed by the sub-volume meshes.

Acknowledgements

This research was partially supported by CONACYT-SENER (“Fondo Sectorial CONACYT SENER HIDROCARBUROS”, grand contract 163723).

References

[1] P M Knupp. Algebraic mesh quality metrics. *SIAM J Sci Comput*, 23(1):193–218, 2001.

[2] P M Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elem Anal Des*, 39(3):217–241, 2003.

[3] P M Knupp. A method for hexahedral mesh shape optimization. *Int J Numer Meth Eng*, 58(2):319–332, 2003.

[4] J M Escobar, G Montero, R Montenegro, and E Rodríguez. An algebraic method for smoothing surface triangulations on a local parametric space. *Int J Numer Meth Eng*, 66(4):740–760, 2006.

[5] J M Escobar, R Montenegro, E Rodríguez, and G Montero. Simultaneous aligning and smoothing of surface triangulations. *Eng Comput*, 27(27):17–29, 2011.

[6] Pascal J Frey and Houman Borouchaki. Geometric surface mesh optimization. *Comput Visual Sci*, 1(3):113–121, 1998.

[7] Yongjie Zhang, Chandrajit Bajaj, and Guoliang Xu. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. *Commun numer meth eng*, 25(1):1–18, 2009.

[8] X Jiao, D Wang, and H Zha. Simple and effective variational optimization of surface and volume triangulations. *Eng Comput*, 27(27):81–94, 2011.

[9] R Garimella and M Shashkov. Polygonal surface mesh optimization. *Eng Comput*, 20(20):265–272, 2004.

[10] R Garimella, M Shashkov, and P Knupp. Triangular and quadrilateral surface mesh quality optimization using local parametrization. *Comput Methods Appl Mech Engrg*, 193(193):913–928, 2004.

[11] K Shivanna, N Grosland, and V Magnotta. An Analytical Framework for Quadrilateral Surface Mesh Improvement With an Underlying Triangulated Surface Definition. In *19th IMR*, pages 85–102, 2010.

[12] A Gargallo-Peiró, X Roca, J Peraire, and J Sarrate. Defining quality measures for meshes on parameterized cad surfaces. In *21st IMR*, 2012.

[13] Abel Gargallo-Peiró, Xevi Roca, and Sarrate. A surface mesh smoothing and untangling method independent of the CAD parameterization. *Computational Mechanics*, 53(4):587–609, 2014.

[14] A Gargallo-Peiró, X Roca, J Peraire, and J Sarrate. Defining quality measures for validation and generation of high-order tetrahedral meshes. In *22nd IMR*, 2013.

[15] 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.

[16] 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.

[17] L. A. Freitag and P. Plassmann. Local optimization-based simplicial mesh untangling and improvement. *Int J Numer Meth Eng*, 49:109–125, 2000.

[18] L. A. Freitag and P M Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *Int J Numer Meth Eng*, 53:1377–1391, 2002.

[19] Shankar P Sastry, Suzanne M Shontz, and Stephen A Vavasis. A log-barrier method for mesh quality improvement and untangling. *Engineering with Computers*, pages 1–15, 2012.

[20] Thomas Toulorge, Christophe Geuzaine, Jean-François Remacle, and Jonathan Lambrechts. Robust untangling of curvilinear meshes. *Journal of Computational Physics*, 254:8–26, 2013.

[21] Shankar P Sastry and Suzanne M Shontz. A parallel log-barrier method for mesh quality improvement and untangling. *Engineering with Computers*, pages 1–13, 2014.

[22] J M Escobar, E Rodríguez, R Montenegro, G Montero, and J M González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Comput Methods Appl Mech Engrg*, 192(25):2775–2787, 2003.

[23] J M Escobar, E Rodríguez, R Montenegro, G Montero, and J M González-Yuste. SUS Code - Simultaneous mesh Untangling and Smoothing code, 2010.

[24] J Yin and C Teodosiu. Constrained mesh optimization on boundary. *Eng Comput*, 24(3):231–240, 2008.

[25] Alan Kelly, Lukasz Kaczmarczyk, and Chris J Pearce. Mesh Improvement Methodology for 3D Volumes with Non-Planar Surfaces. In *21st IMR*, 2011.

[26] Lei Liu, Yongjie Zhang, Thomas J.R. Hughes, Michael A. Scott, and Thomas W. Sederberg. Volumetric t-spline construction using boolean operations. *Engineering with Computers*, pages 1–15, 2013.

[27] T J Wilson. Simultaneous untangling and smoothing of thetrahedral meshes. Master's thesis, ETSE Camins Canals i Ports de Barcelona, Universitat Politècnica de Catalunya, June 2011.

[28] A Gargallo-Peiró. Smoothing and untangling of meshes on parameterized surfaces. Master's thesis, Facultat de Matemàtiques, Universitat Politècnica de Catalunya, February 2011.

[29] OpenCASCADE. Open CASCADE Technology, 3D modeling & numerical simulation. `http://www.opencascade.org`, 2013.

[30] J Nocedal and S Wright. *Numerical optimization*. Springer Verlag, 1999.

[31] David Roger White. *Automatic, quadrilateral and hexahedral meshing of pseudo-cartesian geometries using virtual subdivision*. PhD thesis, Brigham Young University, 1996.

[32] E Ruiz-Gironés and J Sarrate. Generation of structured hexahedral meshes in volumes with holes. *Finite Elem Anal Des*, 46(10):792 – 804, 2010.