

An automatic and general least-squares projection procedure for sweep meshing

Xevi Roca · Josep Sarrate

Abstract In this paper, we present a new and automatic node projection algorithm to generate hexahedral meshes in extrusion geometries. It is designed to preserve the shape of the cap surfaces in the inner layers of a sweeping mesh. The algorithm is based on least-squares approximation of affine mappings. We report that the functionals that have been traditionally used to compute the affine mapping generate four undesired effects on the inner layers of nodes. Namely, the flattening, skewing, offset scaling, and flipping of the inner layer meshes. To overcome these drawbacks we first analyze several properties of a new functional that depends on two vector parameters. Second, we introduce the concept of the pseudo-area and pseudo-normal vectors defined by a loop of nodes. In addition, we prove several geometrical properties of these vectors. Third, based on the properties of the new functional and on the definition of the pseudo-normal vector, we detail a new projection algorithm that automatically selects the functional vector parameters. The aim of this paper is to provide the implementation details to developers, although we also present the background of the algorithm. Finally, several mesh examples are discussed to assess the properties of the proposed algorithm.

Keywords Finite element method · Mesh generation · Hexahedral elements · Sweep · Node projection · Affine mapping

1 Introduction

Several fast and robust algorithms have been developed to generate unstructured tetrahedral meshes [1, 2]. However, fully automatic unstructured hexahedral mesh generation algorithms are still not available. Therefore, special attention has been focused on existing algorithms that decompose the entire geometry into several simpler pieces that can be considered as union of one-to-one extrusion volumes. Sweeping is one of the most robust and efficient algorithms to mesh these simpler volumes with hexahedral elements. Several algorithms have been devised to generate hexahedral meshes by projecting the cap surfaces along the sweep path [3–6]. In all of them the crucial step is the placement of the inner nodes. From the computational point of view, sweep methods based on a least-squares approximation of an affine mapping are the fastest alternative to compute these projections [7]. Several functionals have been introduced to perform the least-squares approximation, see Sect. 2. In spite of their computational efficiency (both in terms of CPU time and memory), these methods present several drawbacks. For instance, the minimization of these functionals may lead to a set of normal equations with a singular system matrix for very usual geometrical configurations. In addition, the obtained mesh may present several undesired features such as flattening and skewness, see [8] for details.

In order to overcome these shortcomings, in reference [8] we introduced a new functional that depends on two vector parameters that can be selected by the user. However, only a feasible selection of these parameters, based on

This work was partially sponsored by the Spanish *Ministerio de Ciencia e Innovación* under grants DPI2007-62395, BIA2007-66965 and CGL2008-06003-C03-02/CLI.

X. Roca (✉) · J. Sarrate
Laboratori de Càlcul Numèric, ETSE de Camins Canals i Ports
de Barcelona, Universitat Politècnica de Catalunya, Edifici C2,
Jordi Girona 1-3, 08034 Barcelona, Spain
e-mail: xevi.roca@upc.edu

J. Sarrate
e-mail: jose.sarrate@upc.edu

our experience, was provided. In this paper we first prove the relationship between the optimal solution of the classical functional and the optimal solution of the new functional proposed in [8]. In addition, we propose a definition of a measure of the normal vector to a given loop of nodes that we denote by *pseudo-normal*. Based on the previous relationship and the definition of the pseudo-normal, we prove and detail a new algorithm that automatically selects the functional parameters. These parameters are selected in order to preserve the shape of the inner part of projected meshes, i.e. *offset data*. It is important to point out that the geometrical cases that lead to a set of normal equations with a singular system matrix are identified from the singular value decomposition (SVD) of the optimal solution of the classical functional. Moreover, to increase the computational efficiency of the proposed algorithm, the minimization of the new functional adequately reuses the optimal solution of the classical functional. Finally, we present two simple examples that show the robustness and the reliability of the proposed algorithm.

2 Problem statement and functional definitions

Let $X = \{\mathbf{x}^i\}_{i=1,\dots,m} \subset \mathbb{R}^n$ be a set of source points, and $Y = \{\mathbf{y}^i\}_{i=1,\dots,m} \subset \mathbb{R}^n$ a set of target points with $m \geq n$. In a sweep application $\{\mathbf{x}^i\}_{i=1,\dots,m}$ are the nodes that belong to the boundary of the projected layer (where the initial layer is the source surface mesh). Similarly, $\{\mathbf{y}^i\}_{i=1,\dots,m}$ are the nodes that belong to the boundary of the target layer. Our goal is to find a mapping $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that

$$\mathbf{y}^i = \phi(\mathbf{x}^i), \quad i = 1, \dots, m. \quad (1)$$

We approximate ϕ by an affine mapping φ from \mathbb{R}^n to \mathbb{R}^n ,

$$\varphi(\mathbf{x}) = \mathbf{A}(\mathbf{x} - \mathbf{c}^X) + \mathbf{c}^Y, \quad (2)$$

where

$$\mathbf{c}^X := \frac{1}{m} \sum_{i=1}^m \mathbf{x}^i \quad \text{and} \quad \mathbf{c}^Y := \frac{1}{m} \sum_{i=1}^m \mathbf{y}^i$$

are the geometrical centers of the sets X and Y , respectively. The affine mapping φ is computed by minimizing the functional

$$\begin{aligned} F(\mathbf{A}) &:= \sum_{i=1}^m \|\mathbf{y}^i - \mathbf{c}^Y - \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X)\|^2 \\ &= \sum_{i=1}^m \|\bar{\mathbf{y}}^i - \mathbf{A}\bar{\mathbf{x}}^i\|^2, \end{aligned} \quad (3)$$

where $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{c}^X$ and $\bar{\mathbf{y}} = \mathbf{y} - \mathbf{c}^Y$, see details in [5, 8]. The minimization of functional F is equivalent to imposing the following m constraints

$$\mathbf{A}(\mathbf{x}^i - \mathbf{c}^X) = \mathbf{y}^i - \mathbf{c}^Y, \quad i = 1, \dots, m, \quad (4)$$

being the unknowns the coefficients of the $n \times n$ matrix

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{pmatrix}.$$

These constraints can be expressed in matrix form as

$$\mathbf{A}\bar{\mathbf{X}} = \bar{\mathbf{Y}}, \quad (5)$$

where

$$\bar{\mathbf{X}} := \begin{pmatrix} x_1^1 - c_1^X & \cdots & x_n^1 - c_n^X \\ \vdots & & \vdots \\ x_1^m - c_1^X & \cdots & x_n^m - c_n^X \end{pmatrix}$$

and

$$\bar{\mathbf{Y}} := \begin{pmatrix} y_1^1 - c_1^Y & \cdots & y_n^1 - c_n^Y \\ \vdots & & \vdots \\ y_1^m - c_1^Y & \cdots & y_n^m - c_n^Y \end{pmatrix}.$$

Therefore, the minimization of F is equivalent to solving,

$$\bar{\mathbf{X}}^T \mathbf{A}^T = \bar{\mathbf{Y}}^T. \quad (6)$$

However, meshes generated by minimizing functional (3) may present three main problems that are illustrated in Fig. 1:

- *Flattening*. Given a non-planar mesh the projection algorithm generates a planar mesh in the following two conditions:
 - If a non-planar mesh with planar boundary, see Fig. 1a, is projected to another loop of nodes (planar or not), then the minimization of functional (3) leads to a set of normal equations with singular system matrix, see [8]. In practice, the singular value decomposition is used to solve the set of normal equations. In this case the inner part of the projected mesh will be planar. Hence, the shape of the source surface mesh will be lost.
 - If a non-planar mesh is projected to an inner layer with a planar boundary by minimizing (3), see Fig. 1b, then the projected mesh will always be planar, see [8].
- *Flipping*. If a loop of nodes is curved towards one direction of the sweep path and it is projected to another loop curved in the opposite direction, see Fig. 1c, then the solution of the minimization of (3) projects offset data inversely to the expected orientation. This may lead to tangled meshes or to distorted hexahedral elements.

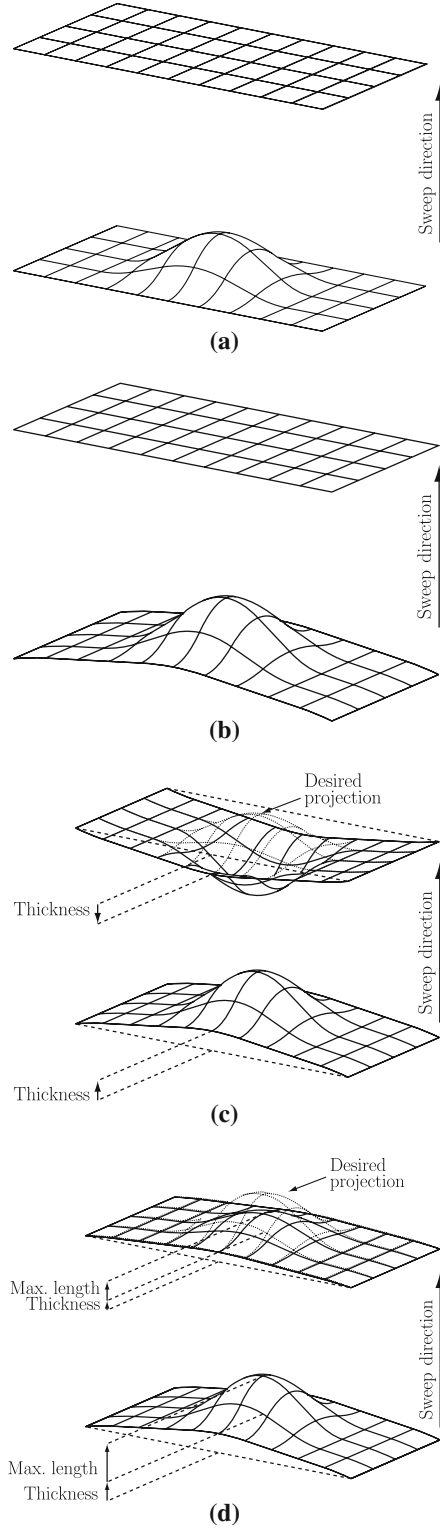


Fig. 1 Graphical representation of the undesired effects. **a** A non-planar mesh with planar boundary is projected to a loop. A planar mesh is obtained; **b** a non-planar mesh is projected to a planar loop. A planar mesh is also obtained; **c** a non-planar loop is projected to another non-planar loop curved on the opposite direction. Offset data direction is inverted; and **d** a non-planar loop is projected to another non-planar loop. Offset data length is scaled

- *Offset scaling.* Figure 1d shows a curved loop that is projected to another loop that is less curved in the same direction. In this case offset data is proportionally scaled. This increment in the scale of offset data may also lead to distorted hexahedral meshes.

It is important to point out that these geometrical configurations are extremely common in CAD models. We will use the term *hyperplanar* to denote a linear variety of dimension $n - 1$ (a plane for $n = 3$ and a straight line for $n = 2$). In particular, given a hyperplanar set of points, X , we define the *homogeneous hyperplane of X* as the subspace of vectors

$$\mathbb{H} = \{\mathbf{v} \in \mathbb{R}^n | \langle \mathbf{n}^X, \mathbf{v} \rangle = 0\}, \quad (7)$$

where $\mathbf{n}^X \in \mathbb{R}^n$ is a unitary normal vector to X .

To solve the previous drawbacks, Knupp introduced another change of coordinates: $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{c}^X + \mathbf{c}^Y - \mathbf{c}^X$ and $\bar{\mathbf{y}} = \mathbf{y} - \mathbf{c}^X$, see [5, 8] for details. Moreover, using these new coordinates the following functional was also defined in [5]

$$\begin{aligned} G(\mathbf{A}) &:= \sum_{i=1}^m \|\mathbf{y}^i - \mathbf{c}^X - \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X + \mathbf{c}^Y - \mathbf{c}^X)\|^2 \\ &= \sum_{i=1}^m \|\bar{\mathbf{y}}^i - \mathbf{A}\bar{\mathbf{x}}^i\|^2. \end{aligned} \quad (8)$$

Therefore, we are looking for a linear mapping \mathbf{A} such that it approximately transforms, in the least-squares sense, $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}^i\}_{i=1,\dots,m}$ to $\bar{\mathbf{Y}} = \{\bar{\mathbf{y}}^i\}_{i=1,\dots,m}$.

However, functional (8) also presents two important shortcomings:

- If the set of source points, X , is hyperplanar and $\mathbf{c}^Y - \mathbf{c}^X \in \mathbb{H}$, then the minimization of functional G leads to a set of normal equations with singular system matrix, see [8].
- *Skewing.* If a non-planar surface mesh with planar boundary is projected to an inner layer which is non-parallel to the boundary of the source surface, see Fig. 2, then the projected nodes do not preserve the shape of the original surface mesh and a skewing effect is introduced, see [8] for details.

There are two main strategies to project meshes using the previous functionals. The first one always projects from the cap surfaces mesh to inner layers. The second one projects, starting from the cap surfaces, from one layer to the next one in an *advancing front manner*. Note that the previous flattening, flipping, offset scaling and skewing effects do not depend on the strategy used to project meshes. However, for particular geometrical configurations some of them can be magnified if the first option is used.

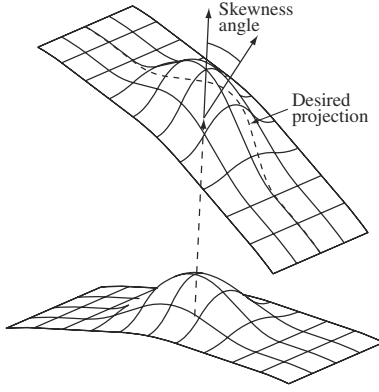


Fig. 2 A non-planar mesh is projected to a non-parallel loop. A skewed mesh is obtained. The desired profile is depicted with a dotted curve

In order to overcome the drawbacks arising from the minimization of functionals F and G , we introduced the following functional, see [8]

$$H(\mathbf{A}; \mathbf{u}^X, \mathbf{u}^Y) := \sum_{i=1}^m \|\mathbf{y}^i - \mathbf{c}^Y - \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X)\|^2 + \sum_{i=1}^m \|\mathbf{u}^Y - \mathbf{A}\mathbf{u}^X\|^2, \quad (9)$$

where \mathbf{u}^X and \mathbf{u}^Y belong to \mathbb{R}^n . It is important to point out that vectors \mathbf{u}^X and \mathbf{u}^Y in (9) can be properly selected in order to obtain several desired properties of functional H .

It has been proved that if the set of source points is hyperplanar it is always possible to select a vector \mathbf{u}^X such that the minimization of H leads to a set of normal equations with a full rank matrix, see [8]. However, given any arbitrary geometry, no algorithm has already been established to properly define vectors \mathbf{u}^X and \mathbf{u}^Y in order to preserve offset data of projected meshes. One of the goals of the present paper is to determine how to select vectors \mathbf{u}^X and \mathbf{u}^Y to define an automatic and robust algorithm to sweep meshes in a one-to-one volume.

3 Analysis of functional H

In this section, we present new properties of functional H that are of major importance to deduce the general node projection algorithm. First, we prove four lemmas that will allow us relating the solutions of the minimization of functionals F and H .

Lemma 1 *If X is a hyperplanar set of points and $\mathbf{u}^X \notin \mathbb{H}$, then $\mathbb{R}^n = \text{span}(\mathbf{u}^X) \oplus \mathbb{H}$.*

Proof In this case, the homogeneous hyperplane defined by X is a subspace of \mathbb{R}^n with dimension equal to $n - 1$.

Since $\mathbf{u}^X \notin \mathbb{H}$ we have that $\text{span}(\mathbf{u}^X) \cap \mathbb{H} = \{\mathbf{0}\}$. Thus, $\mathbb{R}^n = \text{span}(\mathbf{u}^X) \oplus \mathbb{H}$. \square

Lemma 2 *Let X be a hyperplanar set of points. Assume that $\mathbf{u}^X \notin \mathbb{H}$, $\mathbf{u}^Y \in \mathbb{R}^n$, and $\mathbf{A} \in \mathcal{L}(\mathbb{R}^n)$ are given. Then, there exists a mapping $\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y] : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that:*

- (i) $\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y] \in \mathcal{L}(\mathbb{R}^n)$
- (ii) $\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y](\mathbf{u}^X) = \mathbf{u}^Y$
- (iii) $\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y](\mathbf{v}_{\mathbb{H}}) = \mathbf{A}\mathbf{v}_{\mathbb{H}}, \quad \forall \mathbf{v}_{\mathbb{H}} \in \mathbb{H}$.

Proof Given $\mathbf{v} \in \mathbb{R}^n$, Lemma 1 states that \mathbb{R}^n can be represented as the direct sum of the subspaces $\text{span}(\mathbf{u}^X)$ and \mathbb{H} . Therefore, for every $\mathbf{v} \in \mathbb{R}^n$ there exist $\mathbf{v}_{\mathbb{H}} \in \mathbb{H}$ and $\lambda \in \mathbb{R}$ such that

$$\mathbf{v} = \mathbf{v}_{\mathbb{H}} + \lambda \mathbf{u}^X. \quad (10)$$

Hence, we define the image of $\mathbf{v} \in \mathbb{R}^n$ by $\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y]$ as

$$\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y](\mathbf{v}) := \mathbf{A}\mathbf{v}_{\mathbb{H}} + \lambda \mathbf{u}^Y. \quad (11)$$

It is straightforward to prove that $\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y]$ defined in such a way is linear, and that it verifies properties (ii) and (iii). \square

Lemma 2 is the first key issue of the proposed algorithm. To illustrate its application in practical situations, consider that the set of points X define a planar loop of nodes, see Fig. 3. Moreover, assume that the shape of the inner part is non-planar. Let \mathbf{A} be a numerical solution of the minimization of functional F . Since the loop of nodes X is planar, then the linear mapping \mathbf{A} will introduce the flattening effect. Thus, it maps all vectors that do not belong to plane \mathbb{H} to the image of this plane by \mathbf{A} . Therefore, the shape of the inner part of the layer delimited by the loop of nodes X is lost. Under these conditions, Lemma 2 ensures that using the linear mapping Θ we are able to preserve the shape of the inner layer (i.e. the offset data). Note that property (ii) of Lemma 2 states that the linear mapping Θ maps parallel vectors to \mathbf{u}^X (recall that vector $\mathbf{u}^X \notin \mathbb{H}$) into a prescribed direction (selected by vector \mathbf{u}^Y). In addition, property (iii) of Lemma 2 ensures that any vector that belongs to the plane \mathbb{H} is mapped according to the linear mapping \mathbf{A} . In practice, we will determine this matrix \mathbf{A} by minimizing functional F , as we will see in Proposition 1.

Lemma 3 *Let X be a hyperplanar set of points, and assume that $\mathbf{u}^X \notin \mathbb{H}$ and $\mathbf{u}^Y \in \mathbb{R}^n$. Then,*

$$F(\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y]) = F(\mathbf{A}).$$

Proof Since X is hyperplanar, $\mathbf{x}^i - \mathbf{c}^X \in \mathbb{H}$, for $i = 1, \dots, m$. Therefore, by the third property of Lemma 2, $\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y](\mathbf{x}^i - \mathbf{c}^X) = \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X)$, for $i = 1, \dots, m$. Finally, according to the definition of the functional F

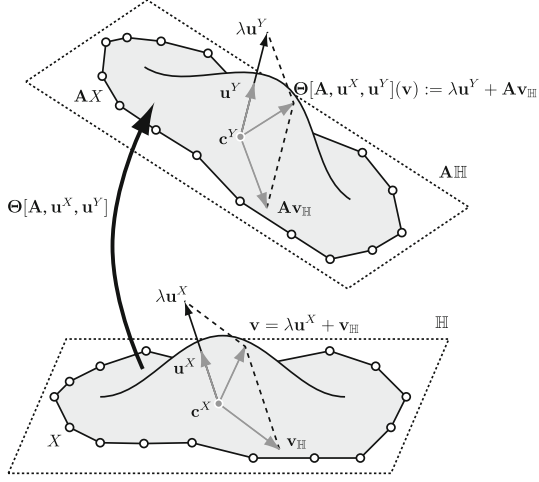


Fig. 3 Transformation of a given vector \mathbf{v} by mapping Θ when X is a planar set

$$\begin{aligned}
 F(\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y]) &= \sum_{i=1}^m \|\mathbf{y}^i - \mathbf{c}^Y - \Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y](\mathbf{x}^i - \mathbf{c}^X)\|^2 \\
 &= \sum_{i=1}^m \|\mathbf{y}^i - \mathbf{c}^Y - \mathbf{A}(\mathbf{x}^i - \mathbf{c}^X)\|^2 = F(\mathbf{A}).
 \end{aligned}$$

□

Lemma 4 Let X be a hyperplanar set of points, and assume that $\mathbf{u}^X \notin \mathbb{H}$ and $\mathbf{u}^Y \in \mathbb{R}^n$. Then,

$$H(\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y]; \mathbf{u}^X, \mathbf{u}^Y) = F(\Theta[\mathbf{A}, \mathbf{u}^X, \mathbf{u}^Y])$$

Proof This result follows from the definitions of functionals F and H , and Lemma 2. □

Proposition 1 Let X be a hyperplanar set of points, and assume that $\mathbf{u}^X \notin \mathbb{H}$ and $\mathbf{u}^Y \in \mathbb{R}^n$. If $\mathbf{A}^F \in \mathcal{L}(\mathbb{R}^n)$ and $\mathbf{A}^H \in \mathcal{L}(\mathbb{R}^n)$ are such that

$$\begin{aligned}
 F(\mathbf{A}^F) &= \min_{\mathbf{A} \in \mathcal{L}(\mathbb{R}^n)} F(\mathbf{A}), \\
 H(\mathbf{A}^H; \mathbf{u}^X, \mathbf{u}^Y) &= \min_{\mathbf{A} \in \mathcal{L}(\mathbb{R}^n)} H(\mathbf{A}; \mathbf{u}^X, \mathbf{u}^Y),
 \end{aligned}$$

then $\Theta[\mathbf{A}^F, \mathbf{u}^X, \mathbf{u}^Y] = \mathbf{A}^H$.

Proof We have already proved, see details in [8], that the minimization of functional H is equivalent to solving n uncoupled overdetermined linear systems. Thus, the minimization of functional H has one and only one solution. In addition, we define $R(\mathbf{A}; \mathbf{u}^X, \mathbf{u}^Y) := \|\mathbf{u}^Y - \mathbf{A}\mathbf{u}^X\|^2$. Hence,

$$H(\mathbf{A}; \mathbf{u}^X, \mathbf{u}^Y) = F(\mathbf{A}) + R(\mathbf{A}; \mathbf{u}^X, \mathbf{u}^Y). \quad (12)$$

We consider the following sequence of equalities and inequalities:

$$\begin{aligned}
 F(\mathbf{A}^F) &\leq F(\mathbf{A}^H) \quad \mathbf{A}^F \text{ minimizes } F \\
 &\leq F(\mathbf{A}^H) + R(\mathbf{A}^H; \mathbf{u}^X, \mathbf{u}^Y) \quad R(\mathbf{A}; \mathbf{u}^X, \mathbf{u}^Y) \geq 0 \\
 &= H(\mathbf{A}^H; \mathbf{u}^X, \mathbf{u}^Y) \quad \text{by Eq. (12)} \\
 &\leq H(\Theta[\mathbf{A}^F, \mathbf{u}^X, \mathbf{u}^Y]; \mathbf{u}^X, \mathbf{u}^Y) \quad \mathbf{A}^H \text{ minimizes } H \\
 &= F(\Theta[\mathbf{A}^F, \mathbf{u}^X, \mathbf{u}^Y]) \quad \text{by Lemma 4} \\
 &= F(\mathbf{A}^F) \quad \text{by Lemma 3.}
 \end{aligned}$$

Note that the first and the last terms are the same. Therefore, all the inequalities are in fact equalities. Hence, from the previous sequence of equalities we prove that $\Theta[\mathbf{A}^F, \mathbf{u}^X, \mathbf{u}^Y]$ and \mathbf{A}^H minimize the functional H . Since the minimization of H has a unique solution we have that $\Theta[\mathbf{A}^F, \mathbf{u}^X, \mathbf{u}^Y] = \mathbf{A}^H$, and the proposition holds. □

To summarize, based on the four previous lemmas, Proposition 1 states a strong relationship between the optimal solutions of functional F and H , namely \mathbf{A}^F and \mathbf{A}^H . This relationship is the keystone for the general algorithm to minimize functional H . It states that when X is a hyperplanar set of points we can obtain the unique solution of the minimization of H by means of one of the optimal solutions of the minimization of F . Specifically, \mathbf{A}^H can be obtained as the linear transformation that maps \mathbf{u}^X to \mathbf{u}^Y , and any vector $\mathbf{v}_H \in \mathbb{H}$ to $\mathbf{A}\mathbf{v}_H$, see Fig. 3 for a 3D interpretation. Hence, the two remaining tasks are: (1) to automatically select the two vector parameters \mathbf{u}^X and \mathbf{u}^Y according to the given geometry; and (2) to determine how to compute the linear transformation Θ , see Sect. 5.

4 Preserving offset data

In this section, we introduce several definitions and results in order to formalize some desirable properties of node projection algorithms. The key issue is the definition of a *measure* of the *normal vector* to a given loop of nodes. Recall that in projection algorithms the inner layers are described by a loop of nodes. That is, there is not an underlying surface carrying any additional information. Moreover, in a wide range of applications the loops of nodes are not planar. Therefore the normal vector to this kind of loops is not defined. However, given a loop of nodes we will define a pseudo-normal vector and we will relate it to the preservation of the shape of the inner part of the projected mesh, the offset data.

Definition 1 (Loop) Given a set of m points $X = \{\mathbf{x}^i\}_{i=1, \dots, m} \subset \mathbb{R}^3$, a loop is the closed poly-line constructed by joining \mathbf{x}^i with \mathbf{x}^{i+1} for $i = 1, \dots, m$. We consider that $\mathbf{x}^{m+1} \equiv \mathbf{x}^1$.

In several applications it is necessary to sweep a non-simple connected surface along the extrusion path. These

surfaces are defined by one outer boundary and as many inner boundaries as holes they have. Therefore, we need to consider sets of points composed by several loops. Specifically, one counter-clockwise oriented loop corresponding to the outer boundary, and several clockwise oriented loops corresponding to the inner holes.

Definition 2 (Multi-loop) A set of m points $X = \{\mathbf{x}^i\}_{i=1,\dots,m} \subset \mathbb{R}^3$ is a multi-loop if it is organized in p loops X_1, \dots, X_p .

Definition 3 (Pseudo-area) Given a vector $\mathbf{c} \in \mathbb{R}^3$, the pseudo-area of a loop $X = \{\mathbf{x}^i\}_{i=1,\dots,m} \subset \mathbb{R}^3$ is

$$\mathbf{a}_{\text{pseudo}}^X := \frac{1}{2} \sum_{i=1}^m (\mathbf{x}^i - \mathbf{c}) \times (\mathbf{x}^{i+1} - \mathbf{c}).$$

The pseudo-area of a multi-loop $X = X_1 \cup \dots \cup X_p$ organized in p loops is

$$\mathbf{a}_{\text{pseudo}}^X := \mathbf{a}_{\text{pseudo},1}^X + \dots + \mathbf{a}_{\text{pseudo},p}^X,$$

where $\mathbf{a}_{\text{pseudo},1}^X, \dots, \mathbf{a}_{\text{pseudo},p}^X$ are the pseudo-areas of loops X_1, \dots, X_p , respectively.

Note that $\|(\mathbf{x}^i - \mathbf{c}) \times (\mathbf{x}^{i+1} - \mathbf{c})\|$ is twice the area of the triangle $\widehat{\mathbf{x}^i \mathbf{x}^{i+1} \mathbf{c}}$, see Fig. 4. Moreover, if X is a planar multi-loop, then the pseudo-area, $\mathbf{a}_{\text{pseudo}}^X$, is equal to the area enclosed by X .

In order to prove that pseudo-area is well defined, the next proposition states that the pseudo-area vector does not depend on the selected $\mathbf{c} \in \mathbb{R}^3$. Moreover, it is invariant under translations, and its norm is also invariant under orthogonal transformations.

Proposition 2 (Invariance of pseudo-area) Let $X = \{\mathbf{x}^i\}_{i=1,\dots,m} \subset \mathbb{R}^3$ be a set of points. The pseudo-area vector verifies

(i) Given $\mathbf{c} \in \mathbb{R}^3$ then

$$\mathbf{a}_{\text{pseudo}}^X = \frac{1}{2} \sum_{i=1}^m (\mathbf{x}^i - \mathbf{c}) \times (\mathbf{x}^{i+1} - \mathbf{c}) = \frac{1}{2} \sum_{i=1}^m \mathbf{x}^i \times \mathbf{x}^{i+1}.$$

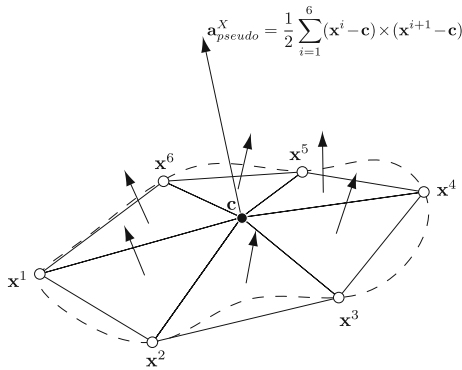


Fig. 4 Geometrical interpretation of the pseudo-area vector

- (ii) Given $\mathbf{t} \in \mathbb{R}^3$ the pseudo-area of X is equal to the pseudo-area of $X + \mathbf{t} = \{\mathbf{x}^i + \mathbf{t}\}_{i=1,\dots,m}$.
- (iii) Given an orthogonal transformation \mathbf{N} , then the pseudo-area of $\mathbf{N}X = \{\mathbf{N}\mathbf{x}^i\}_{i=1,\dots,m}$ is $\mathbf{N}\mathbf{a}_{\text{pseudo}}^X$.

Proof Given $\mathbf{c} \in \mathbb{R}^3$, and taking into account that X is a loop, i.e. $\mathbf{x}^{m+1} \equiv \mathbf{x}^1$, then

$$\begin{aligned} \mathbf{a}_{\text{pseudo}}^X &= \frac{1}{2} \sum_{i=1}^m (\mathbf{x}^i - \mathbf{c}) \times (\mathbf{x}^{i+1} - \mathbf{c}) \\ &= \frac{1}{2} \left[\sum_{i=1}^m \mathbf{x}^i \times \mathbf{x}^{i+1} + \sum_{i=1}^m \mathbf{c} \times (\mathbf{x}^i - \mathbf{x}^{i+1}) + \sum_{i=1}^m \mathbf{c} \times \mathbf{c} \right] \\ &= \frac{1}{2} \sum_{i=1}^m \mathbf{x}^i \times \mathbf{x}^{i+1}, \end{aligned}$$

and property (i) holds. Given $\mathbf{t} \in \mathbb{R}^3$, property (ii) is a direct consequence of property (i) applied to $\mathbf{c} = -\mathbf{t}$.

By property (i) and taking into account that \mathbf{N} is orthogonal we have that

$$\mathbf{a}_{\text{pseudo}}^{\mathbf{N}X} = \frac{1}{2} \sum_{i=1}^m \mathbf{N}\mathbf{x}^i \times \mathbf{N}\mathbf{x}^{i+1} = \mathbf{N}\mathbf{a}_{\text{pseudo}}^X.$$

□

Thus, given a loop X , we have proved that the norm of the pseudo-area vector is a geometrical invariant associated to the loop. Furthermore, it only depends on the ordering and the relative geometrical location of the points.

Proposition 3 (Projected area) If a multi-loop X is projected on an orthogonal plane to its pseudo-area vector, $\mathbf{a}_{\text{pseudo}}^X$, then the obtained polygon has area equal to $\|\mathbf{a}_{\text{pseudo}}^X\|$.

Proof By definition of pseudo-area of a multi-loop, it suffices to prove the result for a single loop. Given a loop X , the projection of the points $\{\mathbf{x}^i\}_{i=1,\dots,m}$ on the orthogonal plane to $\mathbf{a}_{\text{pseudo}}^X$ are the points $\mathbf{x}_{\mathbf{a}^\perp}^i := \mathbf{x}^i - \mathbf{x}_{\mathbf{a}}^i$, where

$$\mathbf{x}_{\mathbf{a}}^i := \frac{\langle \mathbf{x}^i, \mathbf{a}_{\text{pseudo}}^X \rangle}{\langle \mathbf{a}_{\text{pseudo}}^X, \mathbf{a}_{\text{pseudo}}^X \rangle} \mathbf{a}_{\text{pseudo}}^X.$$

Hence, we have that $\mathbf{x}^i = \mathbf{x}_{\mathbf{a}}^i + \mathbf{x}_{\mathbf{a}^\perp}^i$ for $i = 1, \dots, m$. The pseudo-area of X is

$$\begin{aligned} \mathbf{a}_{\text{pseudo}}^X &= \frac{1}{2} \sum_{i=1}^m \mathbf{x}^i \times \mathbf{x}^{i+1} \\ &= \frac{1}{2} \sum_{i=1}^m (\mathbf{x}_{\mathbf{a}}^i + \mathbf{x}_{\mathbf{a}^\perp}^i) \times (\mathbf{x}_{\mathbf{a}}^{i+1} + \mathbf{x}_{\mathbf{a}^\perp}^{i+1}) \\ &= \frac{1}{2} \sum_{i=1}^m \mathbf{x}_{\mathbf{a}}^i \times \mathbf{x}_{\mathbf{a}^\perp}^{i+1} + \mathbf{x}_{\mathbf{a}^\perp}^i \times \mathbf{x}_{\mathbf{a}}^{i+1} + \mathbf{x}_{\mathbf{a}^\perp}^i \times \mathbf{x}_{\mathbf{a}^\perp}^{i+1} \end{aligned}$$

Note that $\mathbf{x}_{\mathbf{a}}^i \times \mathbf{x}_{\mathbf{a}^\perp}^{i+1}$ and $\mathbf{x}_{\mathbf{a}^\perp}^i \times \mathbf{x}_{\mathbf{a}}^{i+1}$ are orthogonal to $\mathbf{a}_{\text{pseudo}}^X$. Thus, their sum is also orthogonal to $\mathbf{a}_{\text{pseudo}}^X$.

Furthermore, $\mathbf{x}_{a^\perp}^i \times \mathbf{x}_{a^\perp}^{i+1}$ is parallel to $\mathbf{a}_{\text{pseudo}}^X$. Therefore, the term $\mathbf{x}_a^i \times \mathbf{x}_{a^\perp}^{i+1} + \mathbf{x}_{a^\perp}^i \times \mathbf{x}_a^{i+1}$ cannot contribute to the parallel component to $\mathbf{a}_{\text{pseudo}}^X$ and it has to be null. Hence, we have that

$$\mathbf{a}_{\text{pseudo}}^X = \frac{1}{2} \sum_{i=1}^m \mathbf{x}_{a^\perp}^i \times \mathbf{x}_{a^\perp}^{i+1},$$

which is the area of the polygon obtained from the projection of the loop X on the plane orthogonal to $\mathbf{a}_{\text{pseudo}}^X$. \square

Hence, the view of the loop from the direction of the pseudo-area has an area equal to the norm of the pseudo-area of X . Therefore, we can interpret the direction of the pseudo-area as a vector *normal* to the loop. Moreover, according to Proposition 4 (see below) the pseudo-area vector is the direction that provides the view of the loop with maximum area.

Definition 4 (Pseudo-normal) The pseudo-normal of a multi-loop X is the unitary vector

$$\mathbf{n}_{\text{pseudo}}^X := \mathbf{a}_{\text{pseudo}}^X / \|\mathbf{a}_{\text{pseudo}}^X\|,$$

where $\mathbf{a}_{\text{pseudo}}^X$ is the pseudo-area of X .

Lemma 5 Let $\mathbf{v} \in \mathbb{R}^3$ be a given vector with $\|\mathbf{v}\| = 1$. Let $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{y} \in \mathbb{R}^3$ be two arbitrary vectors. We decompose these vectors as $\mathbf{x} = \mathbf{x}_v + \mathbf{x}_{v^\perp}$ and $\mathbf{y} = \mathbf{y}_v + \mathbf{y}_{v^\perp}$, where $\mathbf{x}_v := \langle \mathbf{x}, \mathbf{v} \rangle \mathbf{v}$, $\mathbf{x}_{v^\perp} := \mathbf{x} - \mathbf{x}_v$, $\mathbf{y}_v := \langle \mathbf{y}, \mathbf{v} \rangle \mathbf{v}$, and $\mathbf{y}_{v^\perp} := \mathbf{y} - \mathbf{y}_v$. Then,

$$\langle \mathbf{x} \times \mathbf{y}, \mathbf{v} \rangle = \langle \mathbf{x}_{v^\perp} \times \mathbf{y}_{v^\perp}, \mathbf{v} \rangle.$$

Proof Taking into account the proposed decomposition of vectors \mathbf{x} and \mathbf{y} , and that $\langle \cdot, \cdot \rangle$ is bilinear we have that

$$\begin{aligned} \langle \mathbf{x} \times \mathbf{y}, \mathbf{v} \rangle &= \langle \mathbf{x}_v \times \mathbf{y}_v, \mathbf{v} \rangle + \langle \mathbf{x}_v \times \mathbf{y}_{v^\perp}, \mathbf{v} \rangle \\ &\quad + \langle \mathbf{x}_{v^\perp} \times \mathbf{y}_v, \mathbf{v} \rangle + \langle \mathbf{x}_{v^\perp} \times \mathbf{y}_{v^\perp}, \mathbf{v} \rangle. \end{aligned}$$

Therefore, since $\mathbf{x}_v \times \mathbf{y}_v$, $\mathbf{x}_v \times \mathbf{y}_{v^\perp}$ and $\mathbf{x}_{v^\perp} \times \mathbf{y}_v$ are orthogonal to \mathbf{v} , we have that $\langle \mathbf{x} \times \mathbf{y}, \mathbf{v} \rangle = \langle \mathbf{x}_{v^\perp} \times \mathbf{y}_{v^\perp}, \mathbf{v} \rangle$.

Proposition 4 (Maximum projected area) Let $\mathbf{v} \in \mathbb{R}^3$ with $\|\mathbf{v}\| = 1$ be an arbitrary unitary vector. Consider the signed area enclosed by the projection of a multi-loop X to the orthogonal plane to \mathbf{v} . Then, this area is maximized when $\mathbf{v} = \mathbf{n}_{\text{pseudo}}^X$. Moreover, the value of the enclosed area is $\|\mathbf{a}_{\text{pseudo}}^X\|$.

Proof The value of the signed area of the projection of X onto a plane orthogonal to \mathbf{v} is

$$a_{v^\perp}^X := \frac{1}{2} \sum_{i=1}^m \langle \mathbf{x}_{v^\perp}^i \times \mathbf{x}_{v^\perp}^{i+1}, \mathbf{v} \rangle.$$

Taking into account Lemma 5 and that $\langle \cdot, \cdot \rangle$ is bilinear we have that

$$a_{v^\perp}^X = \left\langle \frac{1}{2} \sum_{i=1}^m \mathbf{x}^i \times \mathbf{x}^{i+1}, \mathbf{v} \right\rangle.$$

By means of Proposition 2 we have that

$$a_{v^\perp}^X = \langle \mathbf{a}_{\text{pseudo}}^X, \mathbf{v} \rangle.$$

Then,

$$\max_{\substack{\mathbf{v} \in \mathbb{R}^3 \\ \|\mathbf{v}\|=1}} \frac{1}{2} \sum_{i=1}^m \langle \mathbf{x}_{v^\perp}^i \times \mathbf{x}_{v^\perp}^{i+1}, \mathbf{v} \rangle = \max_{\substack{\mathbf{v} \in \mathbb{R}^3 \\ \|\mathbf{v}\|=1}} \langle \mathbf{a}_{\text{pseudo}}^X, \mathbf{v} \rangle. \quad (13)$$

To finalize, the maximization problem (13) is solved for

$$\mathbf{v} = \frac{\mathbf{a}_{\text{pseudo}}^X}{\|\mathbf{a}_{\text{pseudo}}^X\|} = \mathbf{n}_{\text{pseudo}}^X,$$

and the maximum value is

$$\langle \mathbf{a}_{\text{pseudo}}^X, \mathbf{n}_{\text{pseudo}}^X \rangle = \left\langle \mathbf{a}_{\text{pseudo}}^X, \frac{\mathbf{a}_{\text{pseudo}}^X}{\|\mathbf{a}_{\text{pseudo}}^X\|} \right\rangle = \|\mathbf{a}_{\text{pseudo}}^X\|.$$

Note that if X is a planar loop, the pseudo-normal $\mathbf{n}_{\text{pseudo}}^X$ is equal to the unitary normal \mathbf{n}^X to X . Note that if X is a planar loop, the pseudo-normal $\mathbf{n}_{\text{pseudo}}^X$ is equal to the unitary normal \mathbf{n}^X to X . The pseudo-normal provides a kind of *normal* when there is not an underlying surface, and only the loop of points is available. All the information along the direction of the pseudo-normal is understood as *offset data*. We claim that in order to avoid flattening, flipping, offset scaling and skewing effects we have to obtain affine mappings that preserve the length, direction and orientation of offset data.

5 Algorithm implementation

In this section, we detail the algorithm that we have developed in order to properly select the parameters \mathbf{u}^X and \mathbf{u}^Y and obtain the affine projection. The basic idea is that we can efficiently use the minimization of functional F to minimize H . The key issue is to realize that $\mathbf{A}^H = \mathbf{\Theta}[\mathbf{A}^F, \mathbf{u}^X, \mathbf{u}^Y]$ when X is hyperplanar, see Proposition 1. Therefore, the optimal solution of functional H can be computed from the optimal solution of functional F if a proper criterion to select the vectors \mathbf{u}^X and \mathbf{u}^Y is defined.

The general algorithm, for hyperplanar and non-hyperplanar set of points X , consists of four steps. First, the optimal solution \mathbf{A}^F is computed. Second, we find the singular value decomposition of \mathbf{A}^F . Third, we define a criterion to select the vectors \mathbf{u}^X and \mathbf{u}^Y taking into account the singular value decomposition of \mathbf{A}^F . In addition, if the

Table 1 Algorithm to obtain the affine projection

STEP 1. Compute the optimal solution \mathbf{A}^F according to Equation (15).
STEP 2. Compute the SVD of \mathbf{A}^F according to Equation (16).
STEP 3. Set the values of \mathbf{u}^X and \mathbf{u}^Y :
3.a If $w_i > 0$, for $i = 1, \dots, n$. Set $\mathbf{u}^X = \mathbf{n}^X_{\text{pseudo}}$ and $\mathbf{u}^Y = \mathbf{n}^Y_{\text{pseudo}}$.
3.b If $w_i > 0$, for $i = 1, \dots, n-1$, and $w_n = 0$. Set $\mathbf{u}^X = \mathbf{v}_n$ and $\mathbf{u}^Y = \mathbf{u}_n$.
3.c If $w_i \geq 0$, for $i = 1, \dots, n-2$ and $w_{n-1} = w_n = 0$. Degenerated case not applicable to real situations. Stop the algorithm.
STEP 4. For any $\bar{\mathbf{x}} \in \mathbb{R}^n$ compute the linear part of the affine projection as
$\mathbf{A}(\bar{\mathbf{x}}) = \mathbf{A}^F(\bar{\mathbf{x}} - \langle \bar{\mathbf{x}}, \mathbf{u}^X \rangle \mathbf{u}^X) + \langle \bar{\mathbf{x}}, \mathbf{u}^X \rangle \mathbf{u}^Y$.
Compute the desired affine mapping according to Equation (2)
$\varphi(\mathbf{x}) := \mathbf{A}(\mathbf{x} - \mathbf{c}^X) + \mathbf{c}^Y$.

set of points X and/or Y are hyperplanar (a planar source and/or target surfaces in 3D applications) a geometrical interpretation of the chosen vectors \mathbf{u}^X and \mathbf{u}^Y is also presented. Finally, in the fourth step we compute the affine mapping by first computing its linear part. Table 1 summarizes the proposed algorithm.

5.1 First step: computation of the optimal solution \mathbf{A}^F

In order to minimize functional F we compute the minimum norm solution of Eq. (5). To this end, we use the singular value decomposition of the system matrix

$$\bar{\mathbf{X}}^T = \bar{\mathbf{U}} \bar{\mathbf{W}} \bar{\mathbf{V}}^T, \quad (14)$$

where $\bar{\mathbf{U}}$ is an $m \times n$ matrix with orthogonal columns, $\bar{\mathbf{W}}$ is a $n \times n$ diagonal matrix with positive or zero elements (the singular values)

$$\bar{\mathbf{W}} := \begin{pmatrix} \bar{w}_1 & & \\ & \ddots & \\ & & \bar{w}_n \end{pmatrix},$$

such that $\bar{w}_1 \geq \bar{w}_2 \geq \dots \geq \bar{w}_{n-1} \geq \bar{w}_n \geq 0$, and $\bar{\mathbf{V}}$ is an $n \times n$ orthogonal matrix. We denote by $\bar{\mathbf{v}}_i \in \mathbb{R}^n$, for $i = 1, \dots, n$, the columns of matrix $\bar{\mathbf{V}}$.

We minimize functional F by computing the minimum norm solution of Eq. (5). To this end, we use the singular value decomposition of matrix $\bar{\mathbf{X}}^T$ presented in (14). Specifically, we compute \mathbf{A}^F as

$$\mathbf{A}^F = \bar{\mathbf{Y}} \bar{\mathbf{U}} \bar{\mathbf{W}}^+ \bar{\mathbf{V}}^T, \quad (15)$$

where

$$\bar{\mathbf{W}}^+ := \begin{pmatrix} \bar{w}_1^+ & & \\ & \ddots & \\ & & \bar{w}_n^+ \end{pmatrix},$$

and

$$\bar{w}_i^+ = \begin{cases} 0 & \text{if } \bar{w}_i = 0 \\ \frac{1}{\bar{w}_i} & \text{if } \bar{w}_i \neq 0 \end{cases} \quad \text{for } i = 1, \dots, n.$$

5.2 Second step: singular value decomposition of \mathbf{A}^F

Once we have computed the optimal solution \mathbf{A}^F according to (15), we compute its singular value decomposition

$$\mathbf{A}^F = \mathbf{U} \mathbf{W} \mathbf{V}^T, \quad (16)$$

where \mathbf{U} and \mathbf{V} are two $n \times n$ orthogonal matrices, and \mathbf{W} is a $n \times n$ diagonal matrix with positive or zero elements (the singular values)

$$\mathbf{W} := \begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{pmatrix},$$

such that $w_1 \geq w_2 \geq \dots \geq w_{n-1} \geq w_n \geq 0$. We denote by $\mathbf{u}_i \in \mathbb{R}^n$ and $\mathbf{v}_i \in \mathbb{R}^n$, for $i = 1, \dots, n$, the columns of matrices \mathbf{U} and \mathbf{V} , respectively.

Remark 1 Let \mathbf{M} be an $m \times n$ matrix, such that its singular value decomposition is $\mathbf{M} = \mathbf{U}_M \mathbf{W}_M \mathbf{V}_M^T$. On one hand, the columns of the orthogonal matrix \mathbf{V}_M with an associated singular value equal to zero span $\text{Ker } \mathbf{M}$. On the other hand, the columns of the orthogonal matrix \mathbf{U}_M with an associated positive singular value span $\text{Range } \mathbf{M}$, see Refs. [9, 10].

5.3 Third step: selection of vectors \mathbf{u}^X and \mathbf{u}^Y and geometrical interpretation

From Eq. (14) we realize that when the set of points X is hyperplanar the diagonal matrix $\bar{\mathbf{W}}$ has a null singular value: $\bar{w}_n = 0$. In this case, the singular value decomposition of the optimal solution \mathbf{A}^F will also have a null singular value: $w_n = 0$. Therefore, to properly choose \mathbf{u}^X and \mathbf{u}^Y we have to analyze the $\text{Ker } \mathbf{A}^F$ and the $\text{Range } \mathbf{A}^F$.

Lemma 6 If $\dim \text{Ker } \mathbf{A}^F = 1$, then $\text{Ker } \mathbf{A}^F = \text{span}(\mathbf{v}_n)$.

Proof Since $\dim \text{Ker } \mathbf{A}^F = 1$ and $w_1 \geq w_2 \geq \dots \geq w_{n-1} \geq w_n \geq 0$ we have that $w_i > 0$ for $i = 1, \dots, n-1$ and $w_n = 0$. To finalize, by Remark 1 we know that $\text{Ker } \mathbf{A}^F = \text{span}(\mathbf{v}_n)$. \square

Lemma 7 If $\dim \text{Ker } \mathbf{A}^F = 1$, then $(\text{Range } \mathbf{A}^F)^\perp = \text{span}(\mathbf{u}_n)$, where $^\perp$ denotes the orthogonal space.

Proof Since $\dim \text{Ker } (\mathbf{A}^F) = 1$ and $w_1 \geq w_2 \geq \dots \geq w_{n-1} \geq w_n \geq 0$ we have that $w_i > 0$ for $i = 1, \dots, n-1$, and $w_n = 0$. Taking into account Remark 1, we know that $\text{Range } \mathbf{A}^F = \text{span}(\mathbf{u}_1, \dots, \mathbf{u}_{n-1})$. To finalize, since \mathbf{U} is orthogonal we have that $\langle \mathbf{u}_n, \mathbf{u}_i \rangle = 0$, for $i = 1, \dots, n-1$. Therefore $(\text{Range } \mathbf{A}^F)^\perp = \text{span}(\mathbf{u}_n)$. \square

Lemmas 6 and 7 define the desired criterion to select vectors \mathbf{u}^X and \mathbf{u}^Y . That is, to obtain the optimal solution \mathbf{A}^H , we first find the optimal solution \mathbf{A}^F , and based on its singular value decomposition we select the vectors \mathbf{u}^X and \mathbf{u}^Y .

It is important to point out that if $\dim \text{Ker } \mathbf{A}^F = 1$ then the vectors \mathbf{u}_n and \mathbf{v}_n have a geometrical interpretation. Therefore, we will prove two additional results that provide a geometrical interpretation to the obtained selection of vectors \mathbf{u}^X and \mathbf{u}^Y in our algorithm. Specifically, Lemma 9 states that if X is a hyperplanar set of points, then \mathbf{v}_n and \mathbf{n}^X generate $\text{Ker } \mathbf{A}^F$ (i.e. they are two parallel vectors). Moreover, Lemma 10 states that if Y is a hyperplanar set of points, then \mathbf{u}_n and \mathbf{n}^Y generate $(\text{Range } \mathbf{A}^F)^\perp$ (i.e. they are also two parallel vectors).

Lemma 8 If X is a hyperplanar set of points and \mathbf{n}^X is an unitary normal vector to X , then

$$\text{Ker } \bar{\mathbf{X}}^T = \text{span}(\bar{\mathbf{v}}_n) = \text{span}(\mathbf{n}^X).$$

Proof Since X is hyperplanar then $\text{Rank } \bar{\mathbf{X}}^T = n-1$, see [8]. That is, $\dim \text{Ker } \bar{\mathbf{X}}^T = 1$. Therefore, $\text{Ker } \bar{\mathbf{X}}^T = \text{span}(\bar{\mathbf{v}}_n)$, see Remark 1. Since \mathbf{n}^X is a unitary normal vector we have that $\bar{\mathbf{X}}^T \mathbf{n}^X = \mathbf{0}$, see Eq. (7). Hence, $\mathbf{n}^X \in \text{Ker } \bar{\mathbf{X}}^T$. Thus, $\text{span}(\mathbf{n}^X) = \text{Ker } \bar{\mathbf{X}}^T = \text{span}(\bar{\mathbf{v}}_n)$. \square

Lemma 9 Let X be a hyperplanar set of points and \mathbf{A}^F the optimal solution of functional F computed according to Eq. (15). If \mathbf{n}^X is an unitary normal vector to X and $\dim \text{Ker } \mathbf{A}^F = 1$, then

$$\text{Ker } \mathbf{A}^F = \text{Ker } \bar{\mathbf{X}}^T = \text{span}(\bar{\mathbf{v}}_n) = \text{span}(\mathbf{v}_n) = \text{span}(\mathbf{n}^X).$$

Proof Since $\bar{\mathbf{V}}$ is an orthogonal matrix we have that $\bar{\mathbf{V}}^T \bar{\mathbf{v}}_n = (0 \dots 0 1)^T$. Moreover, since X is hyperplanar and $\bar{w}_1 \geq \bar{w}_2 \geq \dots \geq \bar{w}_{n-1} \geq \bar{w}_n \geq 0$, we have that $\bar{w}_n = 0$. Therefore, $\bar{\mathbf{W}}^+ \bar{\mathbf{V}}^T \bar{\mathbf{v}}_n = \bar{\mathbf{W}}^+ (0 \dots 0 1)^T = \mathbf{0}$. Hence

$$\mathbf{A}^F \bar{\mathbf{v}}_n = \bar{\mathbf{Y}} \bar{\mathbf{U}} \bar{\mathbf{W}}^+ \bar{\mathbf{V}}^T \bar{\mathbf{v}}_n = \mathbf{0}.$$

That is, $\bar{\mathbf{v}}_n \in \text{Ker } \mathbf{A}^F$. Since $\dim \text{Ker } \mathbf{A}^F = 1$ we have that $\text{Ker } \mathbf{A}^F = \text{span}(\bar{\mathbf{v}}_n)$. To finalize, we only have to apply Lemmas 6 and 8. \square

Lemma 10 If Y is a hyperplanar set of points, $\dim \text{Ker } \mathbf{A}^F = 1$, and \mathbf{n}^Y is an unitary normal vector to Y , then

$$(\text{Rank } \mathbf{A}^F)^\perp = \text{span}(\mathbf{u}_n) = \text{span}(\mathbf{n}^Y).$$

Proof First, since Y is hyperplanar then $\bar{\mathbf{Y}}^T \mathbf{n}^Y = \mathbf{0}$, or equivalently

$$(\mathbf{n}^Y)^T \bar{\mathbf{Y}} = \mathbf{0}^T. \quad (17)$$

Next, we will prove that $(\mathbf{n}^Y)^T \mathbf{U} \mathbf{W} \mathbf{V}^T = \mathbf{0}^T$

$$\begin{aligned} (\mathbf{n}^Y)^T \mathbf{U} \mathbf{W} \mathbf{V}^T &= (\mathbf{n}^Y)^T \mathbf{A}^F \quad \text{by Eq. (16)} \\ &= (\mathbf{n}^Y)^T \bar{\mathbf{Y}} \bar{\mathbf{U}} \bar{\mathbf{W}}^+ \bar{\mathbf{V}}^T \quad \text{by Eq. (15)} \\ &= \mathbf{0}^T \quad \text{by Eq. (17).} \end{aligned}$$

Since \mathbf{V} is orthogonal it is invertible. Thus

$$(\mathbf{n}^Y)^T \mathbf{U} \mathbf{W} = \mathbf{0}^T,$$

which is equivalent to the following set of conditions

$$\begin{aligned} (\mathbf{n}^Y)^T \mathbf{u}_1 w_1 &= 0 \\ &\vdots \\ (\mathbf{n}^Y)^T \mathbf{u}_{n-1} w_{n-1} &= 0 \\ (\mathbf{n}^Y)^T \mathbf{u}_n w_n &= 0. \end{aligned}$$

Since $\dim \text{Ker } \mathbf{A}^F = 1$, then $w_i > 0$, for $i = 1, \dots, n-1$, and $w_n = 0$. Therefore, \mathbf{n}^Y is orthogonal to $\mathbf{u}_1, \dots, \mathbf{u}_{n-1}$ (the first $n-1$ columns of matrix \mathbf{U}). To finalize, using Lemma 7, we have that $\text{span}(\mathbf{u}_n) = (\text{Range } \mathbf{A}^F)^\perp = \text{span}(\mathbf{n}^Y)$. \square

On one hand, in our algorithm we select $\mathbf{u}^X = \mathbf{v}_n$. That is, we choose \mathbf{u}^X as the vector that generates $\text{Ker } \mathbf{A}^F$ when $\dim \text{Ker } \mathbf{A}^F = 1$. On the other hand, we also propose to select $\mathbf{u}^Y = \mathbf{u}_n$. In other words, we select \mathbf{u}^Y as the vector that generates the orthogonal space to $\text{Range } \mathbf{A}^F$ when $\dim \text{Ker } \mathbf{A}^F = 1$. Therefore, Lemma 9 states that if X is a hyperplanar set of points, then our algorithm selects \mathbf{u}^X as the unitary normal vector to X : $\mathbf{u}^X = \mathbf{n}^X$, which is in fact the natural choice. Moreover, Lemma 10 states that if Y is a hyperplanar set of points, then our algorithm selects \mathbf{u}^Y as the unitary normal vector to Y : $\mathbf{u}^Y = \mathbf{n}^Y$, which is also the obvious choice. Table 2 presents the geometrical interpretation of the proposed selection of vectors \mathbf{u}^X and \mathbf{u}^Y .

Note that, according to Table 2, when both loops of nodes X and Y are non-hyperplanar we select \mathbf{u}^X and \mathbf{u}^Y as $\mathbf{n}_{\text{pseudo}}^X$ and $\mathbf{n}_{\text{pseudo}}^Y$, respectively. That is, we select \mathbf{u}^X and \mathbf{u}^Y as a measure of the normal directions to the loops X

Table 2 Selection of vectors \mathbf{u}^X and \mathbf{u}^Y according to the sets X and Y

	Y hyperplanar	Y non-hyperplanar
X hyperplanar	$\dim \text{Ker } \mathbf{A}^F = 1$ $\mathbf{u}^X = \mathbf{n}^X = \mathbf{n}_{\text{pseudo}}^X$ $\mathbf{u}^Y = \mathbf{n}^Y = \mathbf{n}_{\text{pseudo}}^Y$	$\dim \text{Ker } \mathbf{A}^F = 1$ $\mathbf{u}^X = \mathbf{n}^X = \mathbf{n}_{\text{pseudo}}^X$ $\mathbf{u}^Y = \mathbf{u}_n$
X non-hyperplanar	$\dim \text{Ker } \mathbf{A}^F = 1$ $\mathbf{u}^X = \mathbf{v}_n$ $\mathbf{u}^Y = \mathbf{n}^Y = \mathbf{n}_{\text{pseudo}}^Y$	$\dim \text{Ker } \mathbf{A}^F = 0$ $\mathbf{u}^X = \mathbf{n}_{\text{pseudo}}^X$ $\mathbf{u}^Y = \mathbf{n}_{\text{pseudo}}^Y$

and Y . In this case, and according to property (iii) of Proposition 2, we are mapping the component in the direction of the pseudo-normal of loop X to the pseudo-normal of loop Y .

5.4 Fourth step: computation of the affine mapping

Once we have selected vectors \mathbf{u}^X and \mathbf{u}^Y , for any centered vector $\bar{\mathbf{x}} \in \mathbb{R}^n$ we compute the linear part of the affine projection as

$$\mathbf{A}(\bar{\mathbf{x}}) := \mathbf{A}^F(\bar{\mathbf{x}} - \langle \bar{\mathbf{x}}, \mathbf{u}^X \rangle \mathbf{u}^X) + \langle \bar{\mathbf{x}}, \mathbf{u}^X \rangle \mathbf{u}^Y.$$

In the case that X is a hyperplanar set we can decompose, by Lemma 2, $\bar{\mathbf{x}}$ as $\bar{\mathbf{x}}_{\mathbb{H}} + \lambda \mathbf{u}^X$. Therefore, the obtained linear transformation maps $\bar{\mathbf{x}}_{\mathbb{H}}$ into $\mathbf{A}^F \bar{\mathbf{x}}_{\mathbb{H}}$ and $\lambda \mathbf{u}^X$ to $\lambda \mathbf{u}^Y$. Hence, by Proposition 1 we know that this linear mapping is the optimal solution of the minimization of functional H , obtained by means of minimizing F .

Finally, we compute the desired affine mapping according to Eq. (2)

$$\varphi(\mathbf{x}) = \mathbf{A}(\bar{\mathbf{x}}) + \mathbf{c}^Y = \mathbf{A}(\mathbf{x} - \mathbf{c}^X) + \mathbf{c}^Y.$$

That is, to obtain the optimal solution \mathbf{A}^H , we first find the optimal solution \mathbf{A}^F , and based on its singular value decomposition we select the vectors \mathbf{u}^X and \mathbf{u}^Y .

6 Examples and discussion

In this section six examples are presented to assess the capabilities and advantages of the proposed algorithm. To highlight the analyzed capabilities in the first four examples we have selected four simple geometries and we have discretized them with a coarse mesh, as suggested in [11]. The last two examples illustrate that the developed algorithm, coupled with volume decomposition, can be successfully used to mesh industrial CAD models. In all examples we first project the source surface onto the target surface [12]. Second, we obtain a structured mesh on the linking sides using a transfinite interpolation algorithm (TFI) [1]. Third, we compute the inner node location

starting from each cap surface and computing the position of the new layer from the previous one in an *advancing front manner*. Finally, we compute the final inner node position by weighting the position obtained from both cap surfaces [3, 5, 8]. Note that in these examples neither the boundary error correction proposed in [3, 7, 13] nor any smoothing procedure is applied to the generated meshes. In the first four examples we analyze the capability of the projection algorithm to reproduce the shape of the inner part of the projected mesh. That is, we test the flattening, flipping, offset scaling, and skewing effects. Other issues such as the application to skewed and twisted sweep paths, layers defined by non-affine or non-convex boundaries have been already addressed, see [8, 12] for details. It is important to point out that in the first four examples all the boundary loops are affine. That is, given two boundary loops an affine mapping exists that exactly maps one into the other. Therefore, functionals F and G become null in each projection (see [8] for details) and the boundary error correction will not improve the initial mesh since there is not error in the projection of the boundaries. In these four examples the inner layers are numbered starting at the inner level next to the bottom cap surface. In order to measure the quality of the hexahedral mesh we use the hexahedron shape metric, f_{shape} , defined in [14]. Note that f_{shape} is a normalized measure. Therefore, it always lays in the range $[0, 1]$.

Flattening test. The goal of the first example is to illustrate that the flattening effect introduced by the minimization of functional F can be avoided using the proposed algorithm. In this example the extrusion volume is defined by a straight sweep path and two non-planar surfaces that have the inner part curved in the same orientation, see Fig. 5a and b. A constant element size is prescribed and 12 inner layers are generated along the extrusion path, see Fig. 5c.

Figure 6a and b shows the central cross-section and the mesh quality of the obtained meshes minimizing functionals F and using the proposed algorithm, respectively. The boundary loops of the cap surfaces are such that while the sweeping process advances from one layer to the next one, the boundary loops become flatter and flatter until a planar boundary loop is reached at one quarter of the sweep path, see Fig. 5c. Therefore, the flattening and offset scaling effects produced by the minimization of functional F appear, being the flattening effect the most important. That is, in each projection the inner shape of the projected mesh becomes more planar. When the planar loop at one quarter of the extrusion path is reached, see Fig. 6a, a planar projected mesh is obtained and the offset data of the cap surfaces is completely lost. Nevertheless, the proposed algorithm also imposes that the optimal solution has to map \mathbf{u}^X into \mathbf{u}^Y . According to our selection of these vectors, we

Fig. 5 Flattening test.
a Perspective view of the wire-frame model; **b** front view of the wire-frame model; and **c** surface mesh

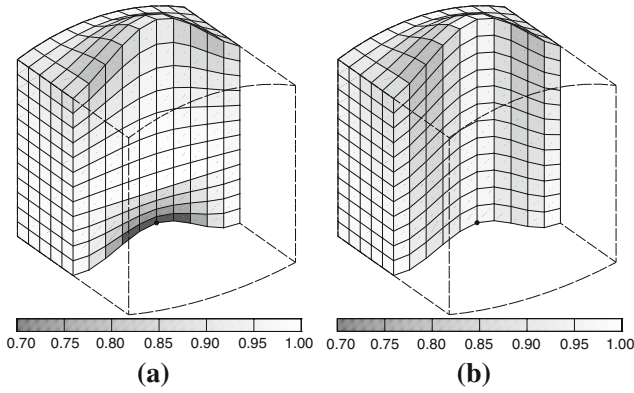
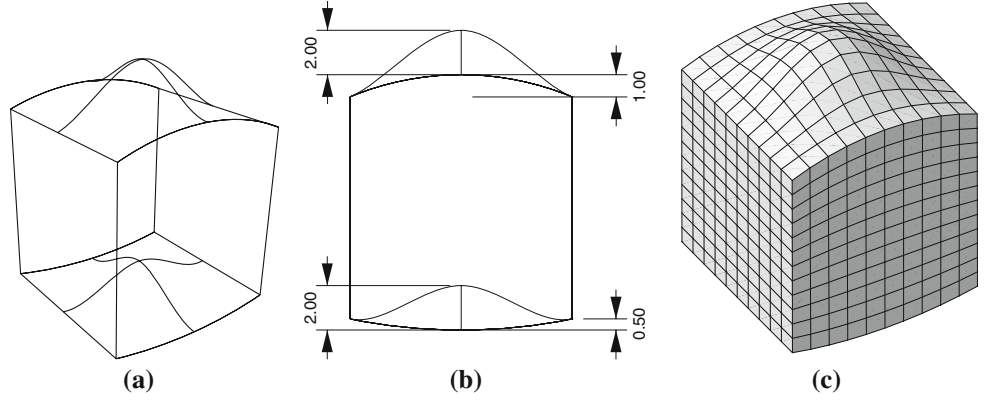


Fig. 6 Central cross-section and shape quality values of the obtained meshes **(a)** minimizing functional F ; and **(b)** using the proposed algorithm

take into account the offset data of the surface meshes, and in each projection, the location of the inner nodes of the new layer resemble the shape of the cap meshes, see Fig. 6b.

Figure 7a and b shows the histograms of the distribution of the elements according to the shape quality measure for the generated meshes. Note that using the proposed algorithm we are able to increase the minimum quality value, $\min(f_{\text{shape}})$. However, the minimization of F generates elements with a higher value of $\max(f_{\text{shape}})$. The general behavior, which we have also observed in other examples, is that the proposed algorithm tends to increase the minimum quality value and to concentrate the quality of the elements around the mean value. Figure 7c details the statistical values of the quality of the mesh obtained minimizing both functionals.

For each inner layer of the generated meshes we compute the minimum and the maximum distance between two consecutive nodes in the extrusion direction. Note that the node at which these distances are reached may be different for each pair of two consecutive inner layers. We denote by *middle node* the mesh node located at the center of the cap

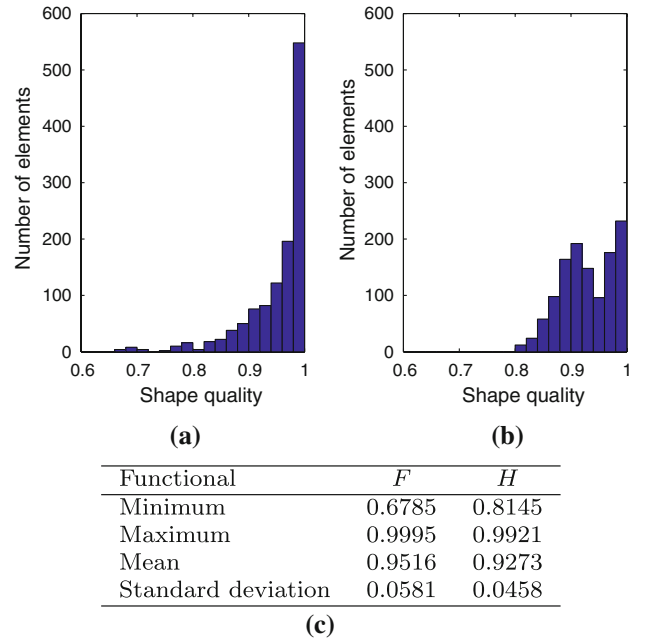


Fig. 7 Distribution of the elements according to the shape quality measure: **a** histogram for functional F ; **b** histogram for functional H ; and **c** statistical values for both functionals

surfaces, marked with a \bullet in Fig. 6. In addition, we compute the distance between the middle node locations for each two consecutive layers. Figure 8a and b plots the obtained values. Note that the minimization of functional F generates inner layers with a wide range of variation for these distances. On the contrary, the proposed algorithm maintains these three distances almost constant in all layers. Figure 8c presents for each layer the angle defined between the two mesh edges adjacent to the middle node and contained in the cutting plane presented in Fig. 6. The bottom cap surface corresponds to the first level and the top cap surface corresponds to the twelfth level. Note that for the mesh obtained by the minimization of functional F the value of this angle first increases from 160° to 180° where a planar inner layer is generated. Then, it decreases again to

Fig. 8 Measures of the shape of the inner layers: **a** distances between two consecutive layers for the minimization of functional F ; **b** distances between two consecutive layers for the minimization of functional H ; and **c** angle between two adjacent edges at the middle node

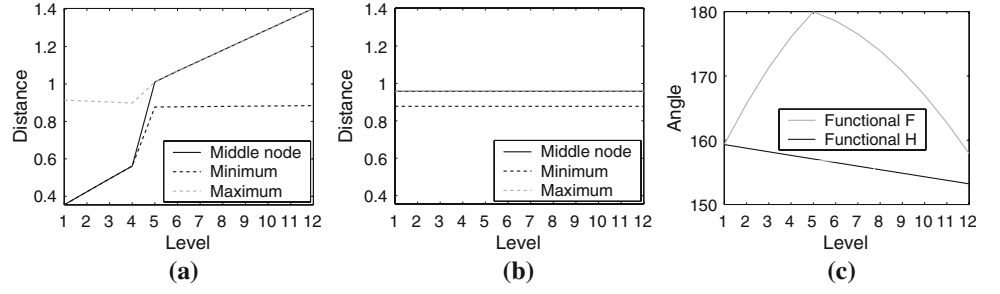


Fig. 9 Flipping test. **a** Perspective view of the wire-frame model; **b** front view of the wire-frame model; and **c** surface mesh

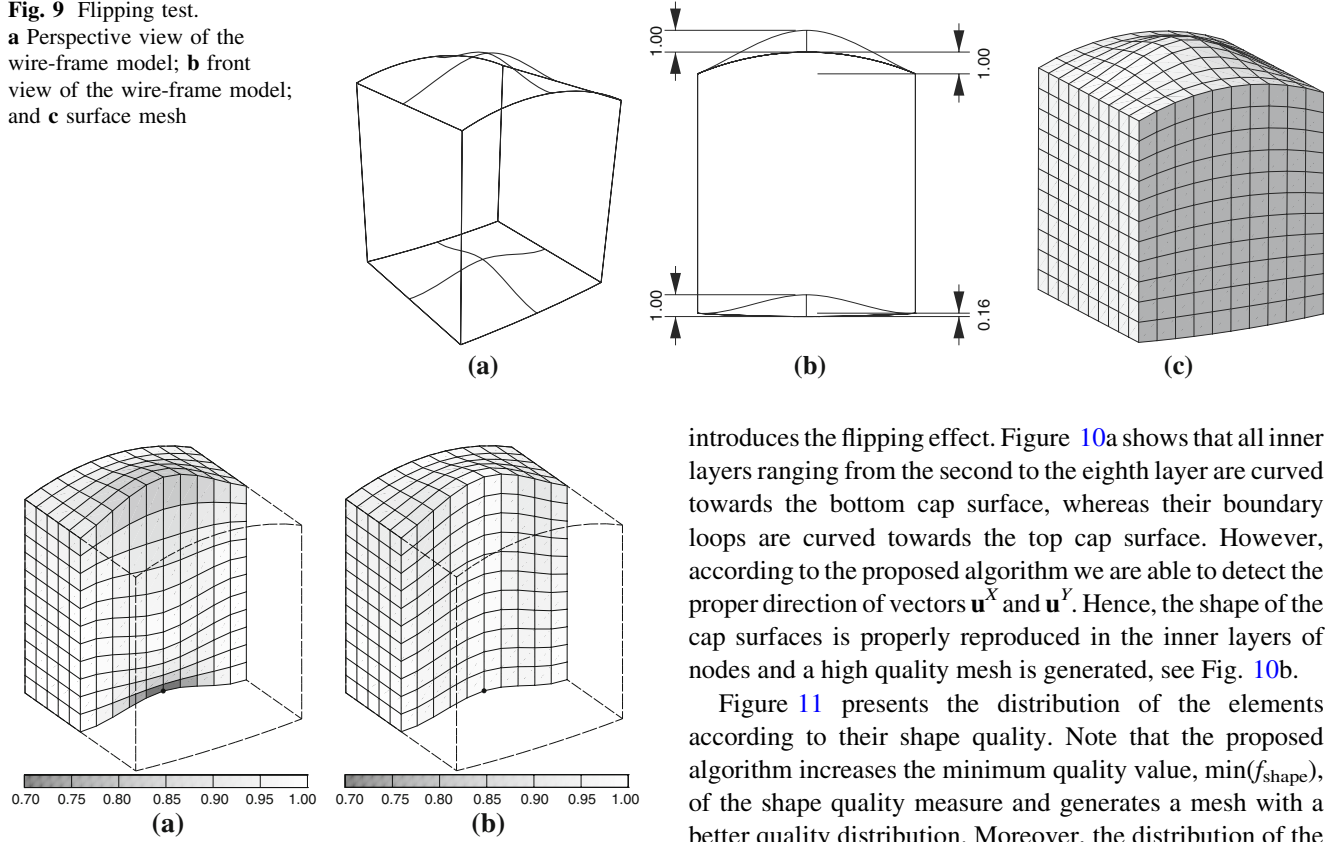


Fig. 10 Central cross-section and shape quality values of the obtained meshes **(a)** minimizing functional F ; and **(b)** using the proposed algorithm

160°. On the contrary, for the mesh obtained by the proposed algorithm the value of the angle monotonically decreases from 160° to 155°.

Flipping test. In the second example we illustrate the ability of the proposed algorithm to avoid the flipping effect introduced by the minimization of functional F . In this example we discretize an extrusion volume defined by two non-planar cap surfaces and a straight sweep path, see Fig. 9a and b. Due to the shape of the boundary loops of the cap surfaces, the loops of nodes that define the boundary of the inner layers are curved towards the top surface except the first one which is curved towards the bottom surface, see Fig. 9c. Therefore, the minimization of functional F

introduces the flipping effect. Figure 10a shows that all inner layers ranging from the second to the eighth layer are curved towards the bottom cap surface, whereas their boundary loops are curved towards the top cap surface. However, according to the proposed algorithm we are able to detect the proper direction of vectors \mathbf{u}^x and \mathbf{u}^y . Hence, the shape of the cap surfaces is properly reproduced in the inner layers of nodes and a high quality mesh is generated, see Fig. 10b.

Figure 11 presents the distribution of the elements according to their shape quality. Note that the proposed algorithm increases the minimum quality value, $\min(f_{\text{shape}})$, of the shape quality measure and generates a mesh with a better quality distribution. Moreover, the distribution of the elements is reduced to the interval $f_{\text{shape}} \in [0.91, 0.99]$. Similar to the first example we mark the middle node with a \bullet in Fig. 10. Figure 12 presents, for each layer, the angle defined between the two mesh edges adjacent at the middle node and contained in the cross-section presented in Fig. 10. It clearly shows that the flipping effect appears when functional F is minimized since the angle increases from 170° in the first inner layer to 190° in the sixth inner layer, and then decreases to 170° in the eleventh inner layer. On the contrary, for the mesh obtained by the proposed algorithm the value of the angle monotonically decreases from 169° to 164°.

Offset scaling test. The goal of the third example is to illustrate how the proposed algorithm reduces the offset scaling effect. Figure 13 presents two views of the wire-frame model of the geometry corresponding to this example. It is defined by two non-planar cap surfaces and a

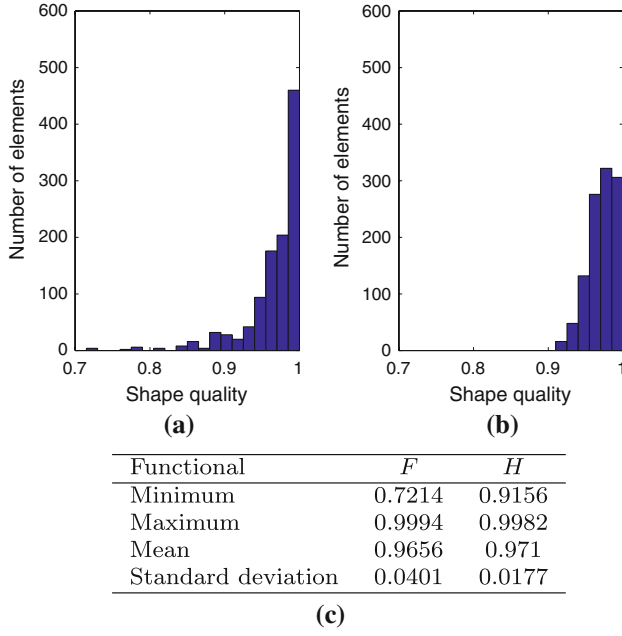


Fig. 11 Distribution of the elements according to the shape quality measure: **a** histogram for functional F ; **b** histogram for functional H ; and **c** statistical values for both functionals

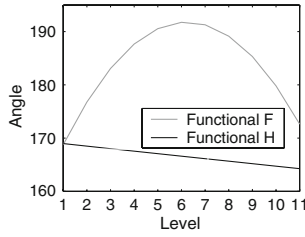


Fig. 12 Angle between two adjacent mesh edges at middle node

straight extrusion path. Since the thickness of the top boundary is higher than the thickness of the bottom boundary, see Fig. 13b, the offset scaling effect will appear when the mesh is obtained minimizing functional F . Figure 14a shows the cross-section and the quality of the obtained mesh when functional F is minimized. Note that the inner layers of elements are more curved at the bottom of the extrusion path leading to highly distorted hexahedral elements at the top of the sweep path. On the contrary, Fig. 14b presents the cross-section and the quality of the generated mesh when the new algorithm is used. Note that in this case a more graded distribution of the element size is obtained along the sweep path, and the inner layers of elements reproduce the shape of the cap surfaces.

Figure 15a and b shows the element distribution according to the shape quality of the element for the meshes obtained by means of functional F and H , respectively. Fig. 15c details the statistical values of the quality for both meshes. Note that using the proposed algorithm

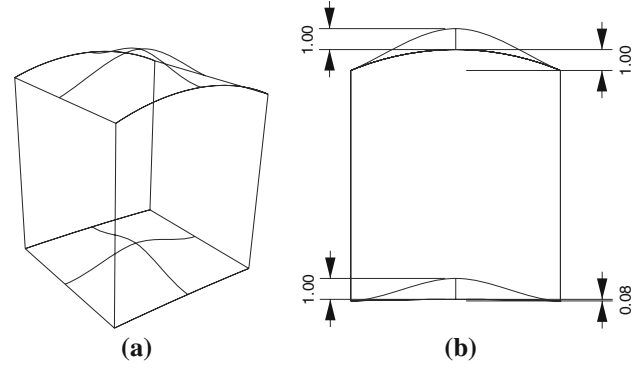


Fig. 13 Wire-frame model of the geometry used for the offset scaling test: **a** perspective view; and **b** front view

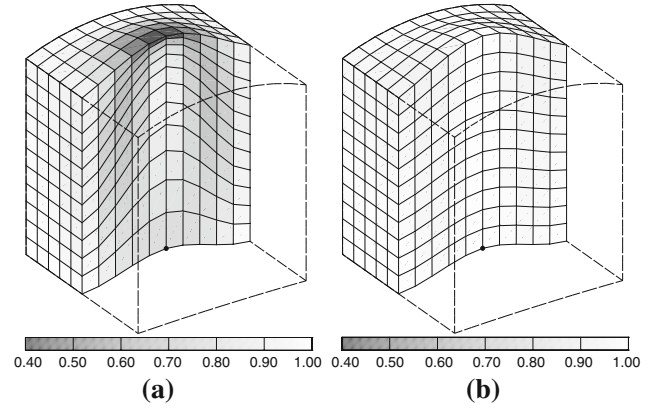
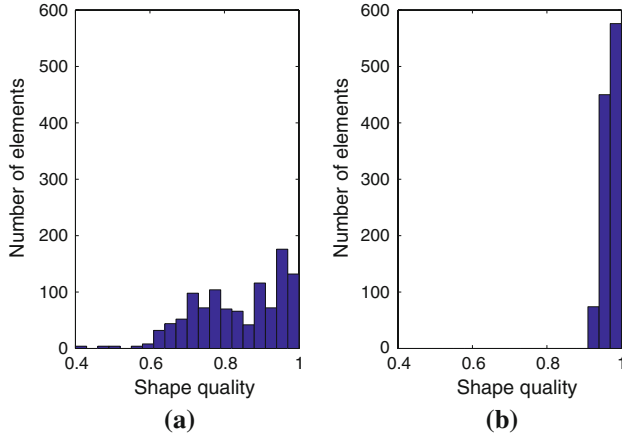


Fig. 14 Central cross-section and shape quality values of the obtained meshes (a) minimizing functional F ; and (b) using the proposed algorithm

the minimum value of the quality measure is more than two times higher than the minimum value obtained minimizing functional F .

For each one of the generated meshes, Fig. 16a and b plots the maximum and the minimum distance between two consecutive nodes in the extrusion direction. In addition, they also plot the distance between the middle node locations in two consecutive layers. Similar to the first example, the proposed algorithm maintains these three distances almost constant in all layers.

Skewing test. In the fourth example we illustrate the ability of the proposed algorithm to avoid the skewing effect introduced by the minimization of functional G . In this example a non-planar surface with a planar boundary is swept along a circular extrusion path. Figure 17 presents a wire-frame model of the geometry. Since the boundary loop of the cap surfaces is planar we know that the minimization of functional F will generate planar inner layers. Therefore, we do not present the meshes obtained by minimizing F . The minimization of functional G generates non-planar inner layers, see Fig. 18a. Since the planar loop



Functional	F	H
Minimum	0.3949	0.9163
Maximum	0.9972	0.9981
Mean	0.8345	0.9700
Standard deviation	0.1190	0.0178

(c)

Fig. 15 Distribution of the elements according to the shape quality measure: **a** histogram for functional F ; **b** histogram for functional H ; and **c** statistical values for both functionals

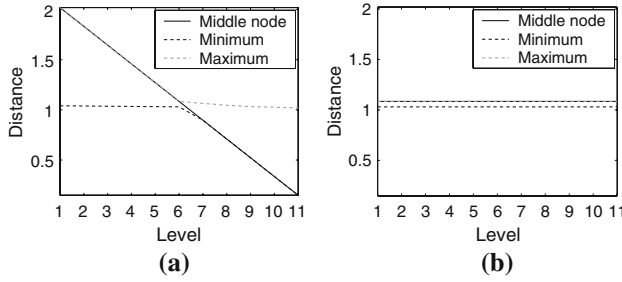


Fig. 16 Minimum, maximum and middle node distances from one level to next one: **a** for the minimization of functional F ; and **b** for the minimization of functional H

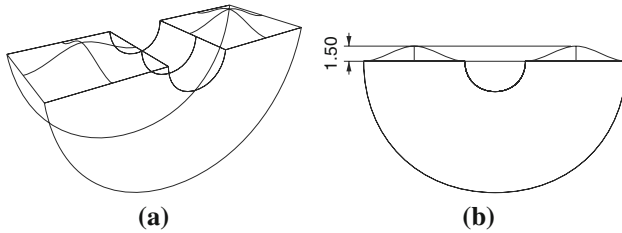


Fig. 17 Wire-frame model of the geometry used for the skewing test: **a** perspective view; and **b** front view

of nodes that define each layer are non-parallel and the vector $\mathbf{c}^Y - \mathbf{c}^X$ is a fixed vector of the optimal solution of the minimization of G (see [8] for details), the skewing effect appears on the generated mesh. Figure 18a presents a cross-section and the quality of the obtained mesh minimizing functional G . Note that the shape of the inner layers

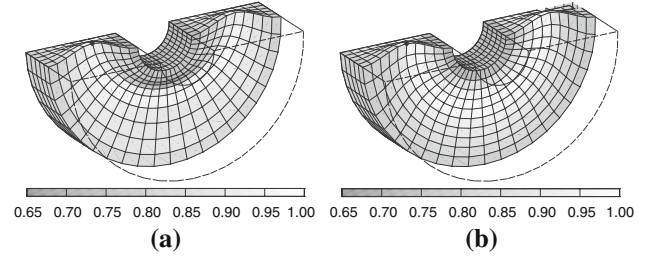
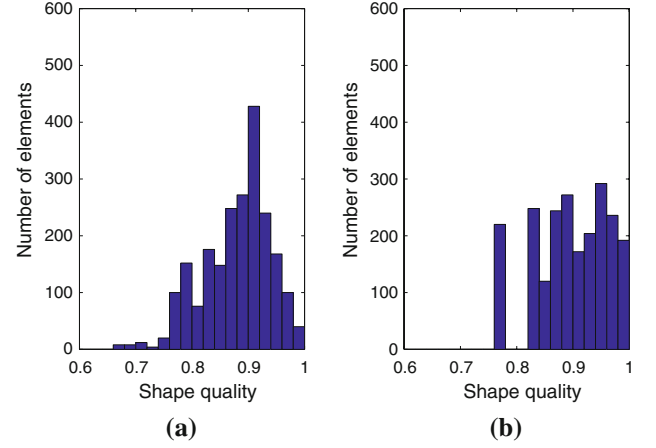


Fig. 18 Central cross-section and shape quality values of the obtained meshes (a) minimizing functional G ; and (b) using the proposed algorithm



Functional	G	H
Minimum	0.6621	0.7648
Maximum	0.9828	0.9920
Mean	0.8792	0.8973
Standard deviation	0.0595	0.0629

(c)

Fig. 19 Distribution of the elements according to the shape quality measure: **a** histogram for functional G ; **b** histogram for functional H ; and **c** statistical values for both functionals

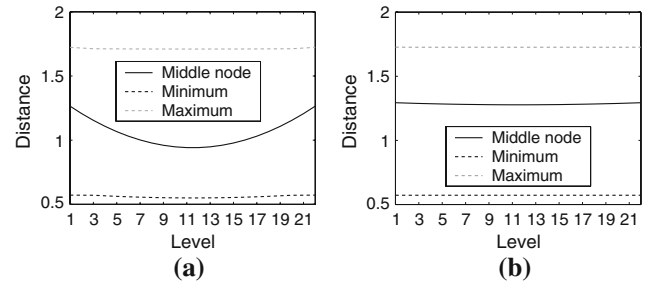


Fig. 20 Minimum, maximum and middle node distances from one level to next one: **a** for the minimization of functional G ; and **b** for the minimization of functional H

is skewed towards the center of the extrusion path. Moreover, distorted hexahedra are generated in the layers located at one half of the extrusion path. On the contrary, Fig. 18b shows a cross-section and the quality of the mesh

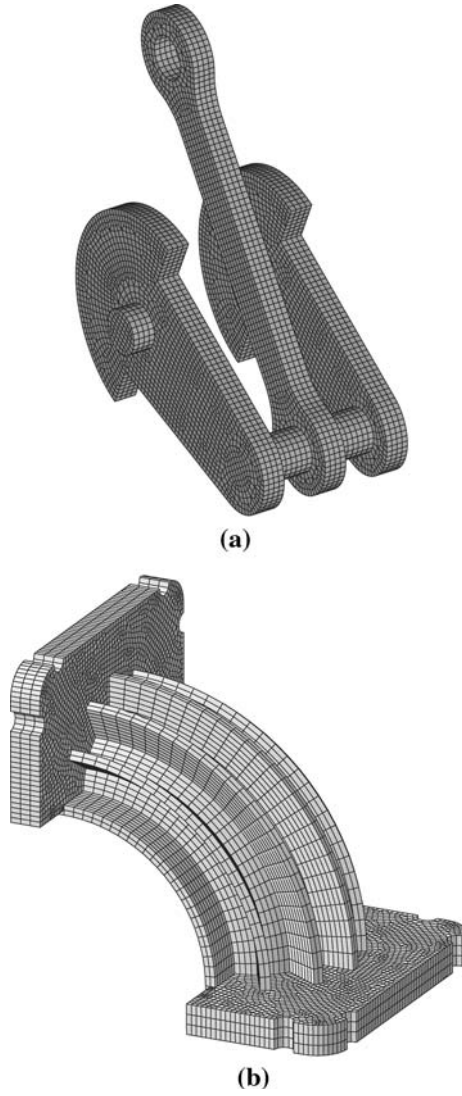


Fig. 21 Application of the proposed algorithm to the discretization of extrusion geometries defined by several one-to-one volumes: **a** crank shaft; and **b** heat sink

generated using the proposed algorithm. In this case, the inner layers are not skewed and reproduce the shape of the cap surfaces.

Figure 19a and b presents the distribution of the elements according to the shape quality measure for the meshes generated minimizing functional G and using the proposed algorithm. Note that the minimum value of the shape quality is increased if the proposed algorithm is used. Figure 19c details the statistical values of the quality of the mesh obtained minimizing both functionals.

Figure 20 shows the minimum, maximum and middle node distances for each pair of two consecutive inner layers corresponding to the meshes generated by means of functionals G and H . Note that both functionals generate meshes with the same values for the minimum and the

maximum distances. However, the distance between two consecutive locations of the middle point reach a minimum value at the middle of the extrusion path whereas the proposed algorithm maintains almost constant this distance.

Finally, the last two examples present two applications of the developed algorithm to extrusion geometries composed by several sweep volumes. Figure 21a shows the mesh generated for a crank shaft model. A conformal mesh is generated over the shared surfaces that define this model. Figure 21b presents the mesh generated for a heat sink. Note that in this example the extrusion path is decomposed in three parts. The first and the last parts are straight whereas the second one has a circular sweep path. In both cases hexahedral elements of high quality are obtained.

7 Conclusions

In this paper we have proposed and detailed a node projection algorithm to obtain hexahedral meshes in one-to-one sweep geometries. Note that one-to-one projection procedures belong to the core of many-to-many sweep algorithms. We show that this algorithm, in conjunction with the boundary error procedure, is of major importance to preserve non-planar shape of the cap surfaces in the inner layers of the hexahedral mesh. Moreover, we claim that the presented algorithm has two additional advantages. First, it provides better node location than the minimization of functionals F and G . Second, since it takes into account the offset data of the cap surfaces (via the vectors \mathbf{u}^X and \mathbf{u}^Y), it triggers the boundary correction procedure when the boundary loops of the layers are affine. To summarize, using this algorithm we are able to overcome flattening, skewness, offset scaling, and flipping effects introduced by the minimization of the traditional functionals.

References

1. Thompson JF, Soni B, Weatherill N (1999) Handbook of grid generation. CRC Press, Boca Raton
2. Owen SJ (1998) A survey of unstructured mesh generation technology. In: 7th international meshing roundtable, Sandia National Laboratories, pp 239–267
3. Blacker T (1996) The Cooper Tool. In: 5th international meshing roundtable, Sandia National Laboratories, pp 13–30
4. Mingwu L, Benzley SE (1996) A multiple source and target sweeping method for generating all-hexahedral finite element meshes. In: 5th international meshing roundtable, Sandia National Laboratories, pp 217–225
5. Knupp PM (1998) Next-generation sweep tool: a method for generating all-hex meshes on two-and-one-half dimensional geometries. In: 7th international meshing roundtable, Sandia National Laboratories, pp 505–513

-
6. Staten ML, Canann SA, Owen SJ (1999) BMSweep: locating interior nodes during sweeping. *Eng Comput* 15:212–218
 7. Scott MA, Earp MA, Benzley SE, Stephenson MB (2004) Adaptive sweeping techniques. In: 14th international meshing roundtable, Sandia National Laboratories, pp 417–432
 8. Roca X, Sarrate J, Huerta A (2005) A new least-squares approximation of affine mappings for sweep algorithms. In: 14th international meshing roundtable, Sandia National Laboratories, pp 433–448
 9. Gill PE, Murray W, Wright MH (1991) Numerical linear algebra and optimization. Addison-Wesley, Edwood City
 10. Lawson C, Hanson R (1974) Solving least squares problems. Prentice-Hall, Englewood Cliffs
 11. Tautges TJ, White DR, Leland RW (2004) Twelve ways to fool the masses when describing mesh generation performance. In: 13th international meshing roundtable, Sandia National Laboratories, pp 181–190
 12. Roca X, Sarrate J, Huerta A (2006) Mesh projection between parametric surfaces. *Commun Numer Methods Eng* 22:591–603
 13. White DR, Saigal S, Owen SJ (2004) CCSweep: automatic decomposition of multi-sweep volumes. *Eng Comput* 20:222–236
 14. Knupp PM (2004) Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elem Anal Des* 39:217–241