Finite Elements in Analysis and Design 72 (2013) 35–46

Contents lists available at SciVerse ScienceDirect

Finite Elements in Analysis and Design

journal homepage: www.elsevier.com/locate/finel

Nagata patch interpolation using surface normal vectors evaluated from the IGES file

D.M. Neto [a,*], M.C. Oliveira [a], L.F. Menezes [a], J.L. Alves [b]

[a] CEMUC, Department of Mechanical Engineering, University of Coimbra, Polo II, Rua Luís Reis Santos, Pinhal de Marrocos, 3030-788 Coimbra, Portugal
[b] CT2M, Department of Mechanical Engineering, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal

ARTICLE INFO

Article history:
Received 23 October 2012
Received in revised form
22 February 2013
Accepted 29 March 2013
Available online 9 May 2013

Keywords:
Nagata patch interpolation
Surface normal vector
IGES format
Trimmed NURBS surface
Piecewise linear mesh
Smooth contact surface description

ABSTRACT

This paper presents an algorithm to accurately evaluate the surface normal vector in any vertex of a finite element mesh, in order to be able to efficiently apply the Nagata patch interpolation as surface mesh smoothing method when solving contact problems. The proposed algorithm considers that the surface geometry is also described by trimmed NURBS surfaces, with input data available in IGES file format. For each mesh vertex, the proposed approach comprises the following three steps: surface global search, local search, and normal vector evaluation. In the global search procedure, all trimmed NURBS surfaces composing the geometric model are ordered by proximity to the vertex. After that, local search is performed to find both the correct NURBS surface and the local coordinates of the vertex, which are defined by its projection on the selected surface. The vertex normal vector is them determined based on the first derivatives of the NURBS surface at the projection point. To highlight the feasibility of the developed algorithm, a mesh smoothing example is presented, emphasising the influence of the vertex normal vector approximation on the interpolation accuracy.

1. Introduction

The application of the Finite Element Method (FEM) in the numerical simulation of sheet metal forming processes requires, among other parameters, the geometric description of the tools involved. Typically, the forming tools can be considered as rigid, which allows to describe them using only their exterior surfaces [1–6]. The numerical treatment of the contact with friction between the deformable body and the tools is a rather complicated task in the field of computational mechanics, due to the strong nonlinearities involved and the non-smooth behaviour of the contact and friction laws [7,8]. The selected tool surface description scheme dictates the way to evaluate the kinematic variables, which are responsible for measuring the relative motion between the sheet and the tools, in order to accurately enforce the frictional contact constraints in the numerical procedure. The approach proposed by Laursen and Simo [7] is one of the most commonly used to evaluate the kinematic variables, since it uses the local coordinate system of the tool surfaces. The identification of the reference position on the tool surface, for each node candidate to contact, is performed through the so-called global and local contact search algorithms [1,9]. In the global contact search, the candidate surfaces are selected in order to minimise the computational cost of the local contact search, where the closest point projection of the each node onto the correct tool surface is achieved, to subsequently assess the kinematic variables.

In sheet metal forming processes, the methods adopted to describe the tools are commonly divided into: (i) analytical function, (ii) parametric patch and (iii) piecewise linear mesh [1,10]. The first tool description scheme is geometrically very accurate and the associated contact search algorithm is simple and efficient. However, it can only be applied for tools with simple geometries. The tools described with parametric patches, such as Gregory [11], Bézier [3,4], Spline [5,12,13] and Non-Uniform Rational B-Spline (NURBS) [14], provide a smooth description of the surfaces, which can attain C^2 continuity. However, these patches are characterised by high order interpolation degree, leading to high computational cost in the contact treatment due to the high number of operations involved in both the local contact search procedure and the kinematic variables evaluation [11,15]. Hence, most of FEM codes resort to piecewise linear mesh scheme for surface description due to its wide application to many types of contact problems, ability to describe complex geometries and simplicity of the contact search algorithms [2,16]. Nevertheless, in order to attain accurate tool descriptions, this method requires a very high density of elements in curved contact zones. Furthermore, the artificial non-smoothness of the resulting contact surfaces leads to non-physical oscillations of the contact forces and can result in convergence problems during the iterative procedure. Therefore, a way to overcome these problems is

[*] Corresponding author. Tel.: +351 23979 0700; fax: +351 23979 0701.
E-mail addresses: diogo.neto@dem.uc.pt (D.M. Neto), marta.oliveira@dem.uc.pt (M.C. Oliveira), luis.menezes@dem.uc.pt (L.F. Menezes), jlalves@dem.uminho.pt (J.L. Alves).

0168-874X/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.finel.2013.03.004

combining an interpolation method with the piecewise linear mesh scheme in order to smooth the mesh with parametric patches, leading to significant improvements both in the tool model accuracy and in the convergence behaviour [4,11,14]. One possible interpolation method is the one proposed by Nagata [17]. The Nagata patch interpolation algorithm is a simple method used to smooth finite element meshes, which requires the knowledge of the surface normal vector in each vertex of the surface mesh [17,18]. However, the information about the normal vectors is unavailable in the typical piecewise mesh files, which contain only the vertices positions and the connectivity of each finite element. In computer graphics and vision science, the strategy commonly adopted to estimate the normal vector in each vertex of the surface mesh is based on the weighted sum of the normal vectors of the facets sharing that vertex [19–21]. The results obtained with this strategy present some errors that are difficult to evaluate, since the correct normal vector is usually inaccessible. It is known that these errors tend to be higher both for coarse and unstructured meshes [19,21]. However, from the user point of view, these kind of meshes are the most attractive to use in numerical simulation since they are fast and easily obtained using automatic generation. Typically the Computer-Aided Engineering (CAE) engineer starts the analysis based on the Computer-Aided Design (CAD) model with the tool design to be used in the numerical simulation. Then, this CAD model is used to generate the tool surface mesh, which is subsequently used in the numerical model. The idea behind this work is to evaluate the surface normal vectors, using the geometric information available in the CAD model, in order to accurately employ the Nagata patch interpolation as smoothing method in the contact with friction problems formulation.

Among the CAD data exchange neutral formats, presently the Initial Graphics Exchange Specification (IGES) file format is the most widely used, to transfer information between CAD and CAE software packages. It contains all the information required to the mathematical definition of the geometry and is organised in a structured manner, following a standard specification [22]. The geometry information is represented in the form of trimmed NURBS surfaces, which is the most general parametric surface description method adopted in CAD [15,23]. NURBS are the standard for surface modelling in most of computer graphics and CAD systems, since many of the typical surface forms used, such as flat planes and quadric surfaces (e.g. cylinders, spheres, ellipsoids of revolution) as well as more complex surfaces, are easily and accurately represented by them [23].

This paper presents an algorithm to evaluate the surface normal vector in each vertex of a piecewise mesh, which is based on the projection of the vertex in the corresponding NURBS surface, defined based on the information available in the IGES file. A brief review of the Nagata patch interpolation method is presented in Section 2, as well as the influence of the normal vector in the interpolation accuracy when applied to a circular arc. Section 3 contains a brief mathematical description of trimmed NURBS surfaces, as well of the IGES file format organisation and description of the entities required to define each surface of the model. Section 4 presents the algorithm to evaluate the normal vector in each mesh vertex, which is divided in a global search procedure for trimmed NURBS surfaces, followed by an orthogonal projection of the vertex on the surface. A numerical example of smoothing a piecewise linear mesh with Nagata patches is presented in Section 5, highlighting the feasibility of the proposed algorithm.

2. Nagata patch interpolation

The Nagata patch is a quadratic parametric interpolator for piecewise meshes [17]. The minimum interpolation degree used to represent a curve, accompanied with its local interpolation structure, leads to a reduced cost in the frictional contact treatment [11,24]. Furthermore, the *quasi-G*^{1} continuity attained between patches guarantees a smooth transition of contact forces when nodes slide across patch boundaries [25]. The recognised drawback associated with the applicability of smoothing methods to 3D unstructured meshes is completely overcome using the Nagata interpolation, since it can be applied both to structured and unstructured surface meshes. In fact, the criteria considered required to develop an appropriate interpolation method for contact surface smoothing, according to [11], are entirely fulfilled by Nagata patch, being this a promising interpolation method to be applied in computational contact mechanics [18,25].

The Nagata patch interpolation algorithm recovers the curvature of surfaces with good accuracy using the position and normal vectors of each vertex of the piecewise model. The mesh refinement increases the accuracy achieved by the Nagata surface description. In fact, for some geometries (e.g. spherical and cylindrical), the maximum interpolation shape error presents quartic convergence with the mesh size [17,26]. The interpolation of an edge is given by the following curve:

$$\mathbf{C}(\xi)=\mathbf{x}_0+(\mathbf{x}_1-\mathbf{x}_0-\mathbf{c})\xi+\mathbf{c}\xi^2, \tag{1}$$

where ξ is the local coordinate of the curve, which satisfy the condition $0\le\xi\le1$. The position vector of each edge end point is given by $\mathbf{x}_0$ and $\mathbf{x}_1$, while its unit normal vectors are given by $\mathbf{n}_0$ and $\mathbf{n}_1$, respectively. The coefficient $\mathbf{c}$, which adds the curvature to the edge is determined as:

$$\mathbf{c}(\mathbf{x}_0,\mathbf{x}_1,\mathbf{n}_0,\mathbf{n}_1)=\begin{cases}\frac{[\mathbf{n}_0,\mathbf{n}_1]}{1-a^2}\begin{bmatrix}1 & -a\\ -a & 1\end{bmatrix}\begin{Bmatrix}\mathbf{n}_0\cdot(\mathbf{x}_1-\mathbf{x}_0)\\ -\mathbf{n}_1\cdot(\mathbf{x}_1-\mathbf{x}_0)\end{Bmatrix} & (a\neq\pm1)\\ \frac{[\mathbf{n}_0,\pm\mathbf{n}_0]}{2}\begin{Bmatrix}\mathbf{n}_0\cdot(\mathbf{x}_1-\mathbf{x}_0)\\ \mp\mathbf{n}_0\cdot(\mathbf{x}_1-\mathbf{x}_0)\end{Bmatrix}=0 & (a=\pm1)\end{cases}, \tag{2}$$

where $a=\mathbf{n}_0\cdot\mathbf{n}_1$, is the cosine of the angle between the normal vectors and $[\mathbf{n}_0,\mathbf{n}_1]$ represents a matrix with the first column equal to vector $\mathbf{n}_0$ and the second one equal to vector $\mathbf{n}_1$. The above formulation, described for an edge, can be easily extended to general *n*-sided patches, such as triangular and quadrilateral patches. These patches can be employed together in order to easily build geometrically complex models, exploiting the surface mesh generator features to generate a mesh composed by both triangular and quadrilateral finite elements. Regardless of the mesh typology, the information necessary to define a Nagata patch is only the surface normal vector in each vertex of the mesh, providing the local support of the interpolation. For more details about the formulation of the patches, the reader should refer to [17,26]. Some improvements to the original Nagata interpolation have been recently proposed by [26] in order to increase its robustness when applied in the contact surface description. The global idea is avoid the generation of very sharp surfaces, situation that can arise due to the quadratic interpolation degree.

The procedure to smooth discretised surfaces using the Nagata patch interpolation can be divided in three sequential steps: surface mesh generation, evaluation of the normal vector in each vertex and Nagata patch interpolation. Concerning the evaluation of the normal vectors, two strategies can be followed: (i) the one proposed in this study or (ii) using only the nodes position and the connectivity of the mesh (recommended only if the CAD geometry is unavailable). Since the second strategy uses less information, it is less accurate [26].

2.1. Influence of the normal vectors in the interpolation accuracy

The application of Nagata patches to smooth the contact surface description requires only the knowledge of the surface normal vector in each vertex of the piecewise linear mesh, being this the only variable that influences the interpolation accuracy for a specific mesh. In this section the circular arc geometry is analysed, in order to study the influence of the normal vectors accuracy in the interpolation error. Several circular arcs with different arc lengths are used to quantity the radial error attained by the Nagata interpolation, when inducing a perturbation in the vertex normal vector. Considering a circular arc of radius R, the radial error of its interpolation is defined as follows:

$$\delta_R(\xi) = \frac{\tilde{R}(\xi) - R}{R} \times 100\%, \tag{3}$$

where $\tilde{R}(\xi)$ is the local radius of the arc resulting from the Nagata interpolation and ξ is the local coordinate of the curve, as defined in Eq. (1). Note that this radius is not constant in all arc length, presenting always the value R at the ends of curve (points $\mathbf{x}_0$ and $\mathbf{x}_1$), as shown in Fig. 1. The interpolated arc is totally defined by means of Eqs. (1) and (2), being the input data composed by the position vector and unit normal vector in each end point of the circular arc.

Since the length of the circular arc to be interpolated influences the accuracy of the Nagata interpolation, this is indirectly defined through the central angle θ indicated in Fig. 1. The six central angles used in this study (5°, 10°, 15°, 20°, 30° and 40°) were selected taking into account the typical arc length range obtained from the discretisation of curved contact surfaces with piecewise meshes. For this simple geometry, the normal vectors necessary to the interpolation can be deduced from the analytical function of the circular arc. The influence in the Nagata interpolation accuracy of a small perturbation δ_n in the analytical normal vector is studied considering that the perturbations imposed are always applied symmetrically, i.e. the perturbation applied to the normal vector $\mathbf{n}_0$ presents the same amplitude and opposite direction to the one applied in the normal vector $\mathbf{n}_1$. This symmetry is shown in Fig. 1, as well as the definition of the positive direction for the perturbation δ_n. The interpolation accuracy is measured through the maximum norm of the radial error attained by the Nagata interpolation. Fig. 2 presents the effect of the perturbation induced in the analytical normal vector on the maximum norm of the radial error $|\delta_R|$, for the six predefined central angles. The range of perturbation is limited in the present analysis between $-4.5°$ and $3.5°$, since its behaviour is nearly linear for $|\delta_n| > 1°$ (see Fig. 2). For positive values of perturbation, the maximum value of $|\delta_R|$ increases both with the central angle θ and with the perturbation amplitude. Since the radial error of the Nagata interpolation of a circular arc using analytical normal vectors is always positive (see Fig. 1), the increase of the perturbation amplitude in the negative direction leads to a transition of the radial error between positive and negative values. This transition is identified in the error evolution through the lower value attained by the maximum value of $|\delta_R|$, which occurs for values of perturbation slightly negatives. In fact, as it is shown in Fig. 2, the radial error observed in the Nagata interpolation of a circular arc when applying the analytical normal vectors is not the lowest possible. However, as it is shown also in the figure, the Nagata interpolation radial error obtained with the analytical normal vectors tends to the minimum when the circular arc length decreases. Although, the error analysis performed in this section is limited to a simple geometry, since it requires the analytical evaluation of the vertex normal vectors, it highlights the importance of an accurate evaluation of the vertex normal vectors, particularly for coarse piecewise meshes. However, in more complex shapes or 3D complex geometries, it is not possible to evaluate analytically the surface normal vector. In those cases, the recommended strategy is to use the CAD geometry, when available.

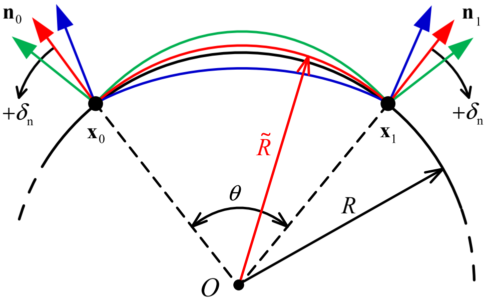

Fig. 1. Schematic representation of the influence of the normal vectors in the Nagata interpolation applied to a circular arc.

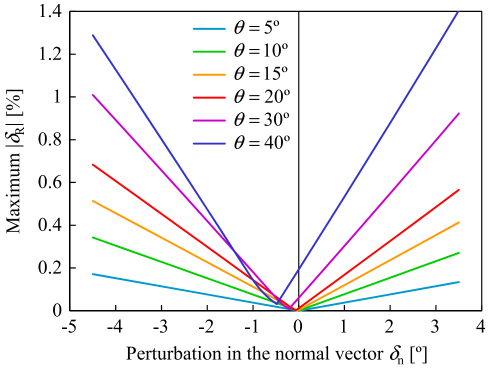

Fig. 2. Effect of the normal vector perturbation δ_n in the maximum $|\delta_R|$ value attained in the Nagata interpolation of several circular arcs defined by the central angle θ.

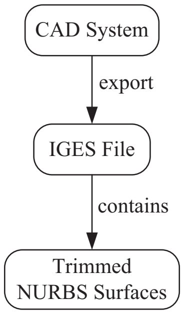

Fig. 3. Procedure for extracting trimmed NURBS surfaces from a CAD system through neutral IGES file format.

3. Trimmed NURBS surfaces and IGES file format

Today there are many CAD software packages available in the market, using all different types of data file formats. However, these software packages also support neutral CAD data file exchange formats, which are used to transfer data between different CAD systems and also with other applications. The two most important and powerful neutral formats are Initial Graphics Exchange Specification (IGES) and STandard for the Exchange of Product model data (STEP). It is commonly stated that the IGES file format is preferable for exporting 3D surface models, while the STEP file format should be adopted for transferring 3D solid models [27–29].

Meanwhile, the most general and advanced parametric surface description method in CAD systems is the NURBS, due to its generality and excellent properties [23,30]. Nevertheless, most of 3D surface models are composed by trimmed NURBS surfaces, which result from the intersection between adjacent basis NURBS surfaces (i.e. surfaces with a rectangular parametric mapping). The

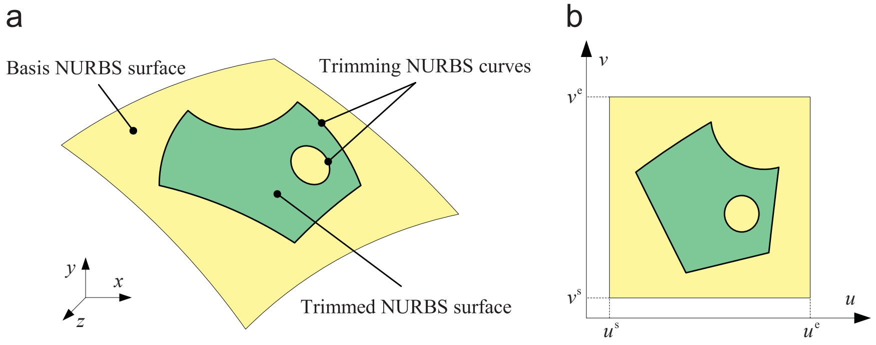

Fig. 4. Representation of a trimmed NURBS surface in the: (a) Euclidean space; (b) parametric domain.

trimmed NURBS surfaces are increasingly used in CAD and computer graphics due to the high flexibility required to represent complex models [15,31,32]. All information needed to define each trimmed NURBS surface, present in the geometric model, is available in the IGES file format. This information is stored in separated geometry entities following the IGES specification. Fig. 3 shows a flowchart with the procedure used to extract the trimmed NURBS surface definition from a general CAD system by means of the IGES file format. The following subsections describe the parameters involved in the trimmed NURBS definition and the way this information is organised in the file, i.e. IGES specification [22].

3.1. Trimmed NURBS surfaces

A trimmed NURBS surface is a basis NURBS surface bounded by a set of trimming curves, which resulted from the intersection with neighbouring surfaces, as schematically shown in Fig. 4. Although the trimming curves can be of any form, when dealing with NURBS entities it is recommended to also represent them in NURBS form [15]. The detailed mathematical description of NURBS surfaces can be found in the literature e.g. [23]. A basis NURBS surface of degree (p,q) has the following form:

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}}, \quad u^{\mathrm{s}} \le u \le u^{\mathrm{e}} \text{ and } v^{\mathrm{s}} \le v \le v^{\mathrm{e}}, \tag{4}$$

where $\mathbf{P}_{i,j}$ is the position vector of the $\{(n+1)(m+1)\}$ control points, which form a bidirectional control net, and $w_{i,j}$ is the weight of each control point. The $N_{i,p}(u)$ and $N_{j,q}(v)$ are the non-rational B-spline basis functions of degree p and q, respectively, defined over the following knot vectors:

$$\mathbf{U} = \left\{ \underbrace{u^{\mathrm{s}}, \ldots, u^{\mathrm{s}}}_{p+1}, u_{p+1}, \ldots, u_{r-p-1}, \underbrace{u^{\mathrm{e}}, \ldots, u^{\mathrm{e}}}_{p+1} \right\}, \tag{5}$$

$$\mathbf{V} = \left\{ \underbrace{v^{\mathrm{s}}, \ldots, v^{\mathrm{s}}}_{q+1}, v_{q+1}, \ldots, v_{s-q-1}, \underbrace{v^{\mathrm{e}}, \ldots, v^{\mathrm{e}}}_{q+1} \right\}, \tag{6}$$

where $r = n+p+1$ and $s = m+q+1$. The most usual way to define the B-spline basis functions is through the Cox-de Boor recursion formula [33,34]. For the case of the u parametric coordinate, the ith B-spline basis function of degree p is defined as:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases},$$
$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u)\,, \tag{7}$$

where u_i are elements of the knot vector (Eq. (5)) called knots, which satisfy the relation $u_i \le u_{i+1}$.

Considering that a NURBS surface is bounded by a set of M trimming curves and all of them are NURBS curves, the kth trimming NURBS curve, with degree l_k, presents the following form [23]:

$$\mathbf{C}^k(t) = \frac{\sum_{i=0}^{h_k} N_{i,l_k}(t) w^k{}_i \mathbf{P}^k{}_i}{\sum_{i=0}^{h_k} N_{i,l_k}(t) w^k{}_i}, \quad t^{\mathrm{sk}} \le t \le t^{\mathrm{ek}}, \quad k = 1, \ldots, M, \tag{8}$$

where $\mathbf{P}^k{}_i$ are the position vectors of the $\{h_k+1\}$ control points and $w^k{}_i$ is the weight of each control point. The knot vector, required to evaluate the non-rational B-spline basis function $N_{i,l_k}(t)$, is defined as:

$$\mathbf{T}^k = \left\{ \underbrace{t^{\mathrm{sk}}, \ldots, t^{\mathrm{sk}}}_{l_k+1}, t^k{}_{l_k+1}, \ldots, t^k{}_{g_k-l_k-1}, \underbrace{t^{\mathrm{ek}}, \ldots, t^{\mathrm{ek}}}_{l_k+1} \right\}, \tag{9}$$

where $g_k = h_k + l_k + 1$. Therefore, in order to define a trimmed NURBS surface it is necessary to know all the parameters defining the basis NURBS surface as well as the M trimming curves.

3.2. IGES file format

The purpose of this section is to provide a direct link between the mathematical parameters presented in Section 3.1 and the information included in the IGES file format. The neutral IGES file, an American National Standard (ANS) format, is actually the most widely used format for exchanging product data among all important CAD/CAM/CAE systems [15]. This neutral file supports three types of data exchange formats: (i) fixed line length ASCII, (ii) compressed ASCII and (iii) binary [22]. Although CAD systems usually preserve transcript files in a binary format, the most commonly used format for data transfer is the fixed line length, where the entire file is divided into lines of 80 characters [15,27]. Therefore, in this study all information required for each trimmed NURBS surface definition is extracted from this type of format. This IGES format is partitioned into five sequentially numbered main sections, organised in the following order:

- Start section (S);
- Global section (G);
- Directory entry section (D);
- Parameter data section (P);
- Terminate section (T).

The columns from 1 to 72 comprise the information which varies according to the file current section, which is identified in column 73 by the letter (S, G, D, P or T), previously specified in brackets. The remainder columns (74–80) contain the line numbers of every file section.

The start section (S) usually contains, in human readable free-form, comments of the sender. In the global section (G) the general file characteristics are presented, such as the parameter delimiter, the record terminator used in the subsequent sections and the units of the model. The default values for the delimiter and the terminator parameters are the comma and the semi-comma, respectively. In the present analysis, the information presented in this section is ignored, except the delimiter and the terminator parameters. The directory entry section (D) contains two consecutive lines to define each entity. This section provides the pointer to the parameter data section (P) containing the information for each entity. In fact, the most important information extracted from the directory entry section (D) is both the first line number and the total number of lines used, in the parameter data section (P), for describing each entity present in the IGES. The parameter data section (P) always starts with the identification of the entity number, followed by the complete information about all parameters associated with the entity. Finally, the terminate section (T) comprises a single line describing the number of lines employed in each of the previous four sections [27].

3.2.1. *Curve and surface geometry entities*

The IGES 5.3 [22] describes about 88 different entities, which can be categorised as geometry and non-geometry. The geometry entities define the physical shape of a model including points, curves, surfaces, solids and relations between entities. On the other hand, the non-geometric entities specify annotation, definition and structure, providing attributes of entities such as colour and status [29]. In the present study, only geometry entities are required to collect the information necessary to define each trimmed NURBS surface, used in the geometric model. Table 1 presents a brief description of the necessary five geometry entities, as well as their identification number in the IGES specification. The pathway followed to identify all mathematical parameters related with trimmed NURBS surfaces is schematically shown in Fig. 5. Thus, from the IGES file available in fixed line length format it is possible to extract all the surface parameters, through the information contained in the entities specified in Table 1. In order to know the data arrangement within the file for these geometry entities, a detailed description of each one is presented, following the IGES specification.

The information presented in the rational B-Spline surface entity (No. 128) corresponds to the basis NURBS surface definition, described previously in Eqs. (4)–(6). In order to easily identify all the numerical parameters involved in those equations, the information associated with this entity, always presents the following specification:

$$128, n, m, p, q, u^{c}, v^{c}, pr, u^{p}, v^{p}, \overbrace{u_0, u_1, \ldots, u_r}^{\mathbf{U}}, \overbrace{v_0, v_1, \ldots, v_s}^{\mathbf{V}}, \overbrace{w_{0,0}, w_{1,0}, \ldots, w_{n,m}}^{w_{i,j}},$$
$$\underbrace{P^{x}_{0,0}, P^{y}_{0,0}, P^{z}_{0,0}, P^{x}_{1,0}, P^{y}_{1,0}, P^{z}_{1,0}, \ldots, P^{x}_{n,m}, P^{y}_{n,m}, P^{z}_{n,m}}_{\mathbf{P}_{i,j}}, u^{s}, u^{e}, v^{s}, v^{e};$$

Note that some numerical parameters existent in the above specification are not used directly in the definition of a basis NURBS surface (see Eqs. (4)–(6)). Hence, Table 2 presents a brief description of all the parameters employed to describe the rational B-Spline surface entity in the parameter data section. For more detailed information please refer to IGES 5.3 [22]. The rational B-Spline curve entity (No. 126) stores the information related with the NURBS curves. The specification for this entity is the following:

$$126, h, l, c^{p}, t^{c}, pr, t^{p}, \overbrace{t_0, t_1, \ldots, t_g}^{\mathbf{T}}, \overbrace{w_0, w_1, \ldots, w_h}^{w_i},$$
$$\underbrace{P^{x}_{0}, P^{y}_{0}, P^{z}_{0}, P^{x}_{1}, P^{y}_{1}, P^{z}_{1}, \ldots, P^{x}_{h}, P^{y}_{h}, P^{z}_{h}}_{\mathbf{P}_{i}}, t^{s}, t^{e}, n^{x}, n^{y}, n^{z};$$

Table 1
Description of the entities used to represent trimmed NURBS surfaces [22].

Entity type	Description
No. 128	Rational B-Spline surface
No. 126	Rational B-Spline curve
No. 102	Composite curve
No. 142	Curve on a parametric surface
No. 144	Trimmed parametric surface

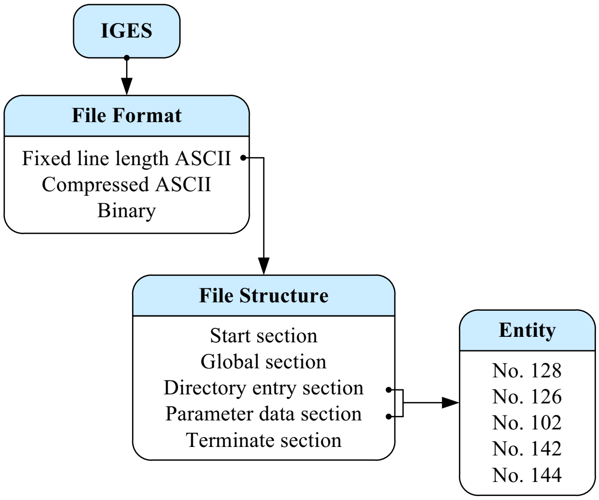

Fig. 5. IGES file format and structure, highlighting the sections containing entity types related with trimmed NURBS surfaces.

The information arrangement in the curve definition is analogous to the one used in the surface entity. In the same way, the numerical parameters involved in Eqs. (8) and (9), which define a general NURBS curve, can also be easily identified in the above specification. The parameters used in the parameter data section to define the rational B-Spline curve entity are presented in Table 2, together with its brief description. All NURBS curves composing the model are represented following the above specification. However, in case of a curve used to define the boundary of a trimmed NURBS surface, its definition is presented in the IGES file using two different domains. Hence, in addition to the specification presented above, the same geometric curve is also specified in the parametric domain of the basis NURBS surface, which is trimmed by the curve. The main difference to the previous specification is the domain where the control points are defined. While in the above specification they are defined in the Euclidean space $(P^{x}_{i}, P^{y}_{i}, P^{z}_{i})$, in the second specification they are defined in the parametric domain of the basis NURBS surface, as following:

$$126, h, l, c^{p}, t^{c}, pr, t^{p}, \overbrace{t_0, t_1, \ldots, t_g}^{\mathbf{T}}, \overbrace{w_0, w_1, \ldots, w_h}^{w_i},$$
$$\underbrace{P^{u}_{0}, P^{v}_{0}, P^{w}_{0}, P^{u}_{1}, P^{v}_{1}, P^{w}_{1}, \ldots, P^{u}_{h}, P^{v}_{h}, P^{w}_{h}}_{\mathbf{P}_{i}}, t^{s}, t^{e}, n^{x}, n^{y}, n^{z};$$

Since the parametric domain of the surface is a 2D space, the third coordinate of each control point is always zero $P^{w}_{i} = 0.0$. Note that the curve degree as well as the number of control points is not obligatorily the same in the two specifications. This last specification is the most useful to define the parametric domain of each trimmed NURBS surface.

The remaining three entities contain only additional information to define the relations between surfaces and curves, i.e., topological information [22]. The composite curve entity (No. 102)

Table 2
Description of the variables used in entity numbers 128 and 126.

Entity	Variable	Description
128	n/m	Number of control points less one in u/v parametric direction
	p/q	Surface degree in u/v parametric direction
	$u^{\text{c}}/v^{\text{c}}$	Closed $(=1)$ or open $(=0)$ surface in u/v parametric direction
	pr	Polynomial $(=1)$ or rational $(=0)$ surface representation
	$u^{\text{p}}/v^{\text{p}}$	Periodic $(=1)$ or non-periodic $(=0)$ surface in u/v parametric direction
	$\mathbf{U}=\{u_0,\ldots,u_r\}$	Surface's knot vector in u parametric direction (see Eq. (5))
	$\mathbf{V}=\{v_0,\ldots,v_s\}$	Surface's knot vector in v parametric direction (see Eq. (6))
	$w_{i,j}=\{w_{0,0},\ldots,w_{n,m}\}$	Weight associated to each control point
	$\mathbf{P}_{i,j}=\{\mathbf{P}_{0,0},\ldots,\mathbf{P}_{n,m}\}$	Position vector of each control point
	$u^{\text{s}}/v^{\text{s}}$	Starting value of u/v in the surface definition (see Eq. (4))
	$u^{\text{e}}/v^{\text{e}}$	Ending value of u/v in the surface definition (see Eq. (4))
126	h	Number of control points less one
	l	Curve degree
	c^{p}	Planar $(=1)$ or not planar $(=0)$ curve
	t^{c}	Closed $(=1)$ or open $(=0)$ curve
	pr	Polynomial $(=1)$ or rational $(=0)$ curve representation
	t^{p}	Periodic $(=1)$ or non-periodic $(=0)$ curve
	$\mathbf{T}=\{t_0,\ldots,t_g\}$	Curve's knot vector (see Eq. (9))
	$w_i=\{w_0,\ldots,w_h\}$	Weight associated to each control point
	$\mathbf{P}_i=\{\mathbf{P}_0,\ldots,\mathbf{P}_h\}$	Position vector of each control point
	t^{s}	Starting value of t in the curve definition (see Eq. (8))
	t^{e}	Ending value of t in the curve definition (see Eq. (8))
	$n^{\text{x}},n^{\text{y}},n^{\text{z}}$	Unit normal (if curve is planar)

Table 3
Description of the variables used in entity numbers 102, 142 and 144.

Entity	Variable	Description
102	n_{sc}	Number of simple curves composing the composite curve
	sc_i	Line number of the i simple curve $(i=1,\ldots,\ n_{sc})$
	wc	The way the curve on the surface was created
	se	Pointer to the surface on which the curve lies
142	cc_{uv}	Pointer to the composite curve defined in the parametric domain
	cc_{xyz}	Pointer to the composite curve defined in the Euclidean space
	rep	Preferred representation in the sending system
	se	Pointer to the surface that is to be trimmed
	tr	Trimmed $(=1)$ or untrimmed $(=0)$ surface definition
144	n_{cci}	Number of closed curves defining the inner boundary of the trimmed surface
	cps^{ob}	Pointer to the curve on parametric surface that define the outer boundary of the trimmed surface
	cps_i^{ib}	Pointer to the i closed curve that defines the inner boundary $(i=1,\ldots,\ n_{cci})$

is an assembly of the individual simple curves that result in a continuous curve. This entity is simply an ordered list of curves, where the terminate point of each simple curve is the start point of the succeeding curve. The specification adopted to indicate this entity in the parameter data section is the following:

$$102,\ n_{sc},\ sc_1,\ sc_2,\ldots,\ sc_{n_{sc}};$$

where n_{sc} denotes the number of simple curves that build the composite curve entity. The parameter sc_i corresponds to the pointer to each i simple curve entity (No. 126), defined in the directory entry section, by means of the corresponding line number in the parameter data section (values indicated in the columns 66–72). In fact, note that all pointers used in the parameter data section correspond to the line number in the directory entry section, associated to the entity. Table 3 presents a brief definition of the parameters used to define the composite curve entity. The curve on a parametric surface entity (No. 142) associates a composite curve with a surface and classifies the curve as lying on the surface. The specification used in the IGES format file for this entity is the following:

$$142,\ wc,\ se,\ cc_{uv},\ cc_{xyz},\ rep;$$

The surface entity (No. 128) on which the curve lies is identified through the pointer denoted by se. Since the composite curve can be defined in two different ways, the curve lying on the surface is identified by two pointers for the same composite curve entity (No. 102). Thus, the pointer cc_{uv} is used to designate the composite curve in the parametric domain of the surface (u,v), while the pointer cc_{xyz} is applied to indicate the same curve, but in the Euclidean space. Table 3 presents a brief description of all parameters employed to define this entity. Finally, the trimmed parametric surface entity (No. 144) contains information about the basis NURBS surface that is to be trimmed as well as the set of trimming curves which define the boundary of the trimmed surface. The specification adopted in the IGES format to represent this entity is the following:

$$144,\ se,\ tr,\ n_{cci},\ cps^{\text{ob}},\ cps_1^{\text{ib}},\ cps_2^{\text{ib}},\ldots,\ cps_{n_{cci}}^{\text{ib}};$$

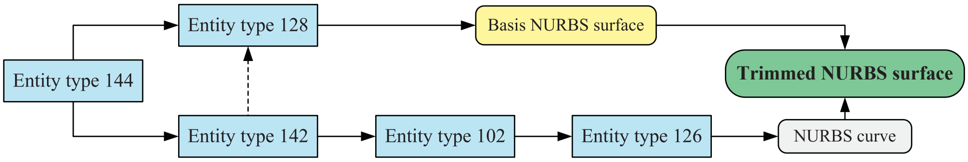

Fig. 6. Procedure to define a trimmed NURBS surface from entity types present in the IGES file.

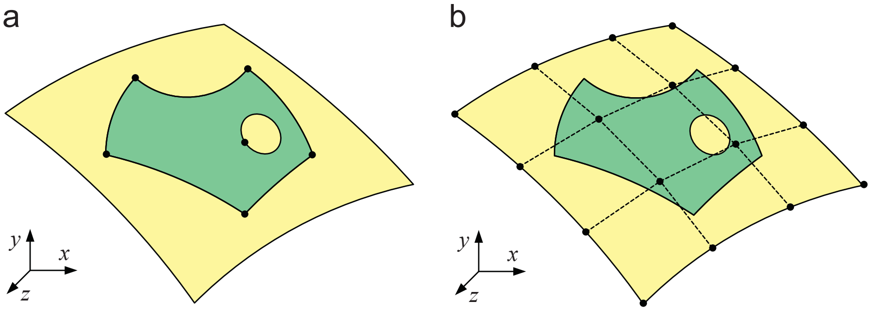

Fig. 7. Set of surface points used in the global search: (a) vertices of the trimmed surface; (b) grid of points equally spaced in the parametric domain.

The pointer *se* indicates the basis NURBS surface entity (No. 128) and the numeric parameter *tr* specifies the type of surface domain that defines the trimmed surface. Hence, the value $tr=0$ means that the domain of the trimmed surface is equal to the one of the basis NURBS surface, i.e. the surface is not trimmed. On the other hand, if the surface domain is defined by a set of trimming curves, the parameter takes the value $tr=1$. The number of closed curves which compose the inner boundary is designated by n_{cci}. The pointer cps^{ob} identifies the curve on the parametric surface entity (No. 142) that constitutes the outer boundary of the trimmed surface, while cps_i^{ib} designates the *i* closed inner boundary curve entity (No. 142) [15,22]. For more details about the parameters used to define the trimmed parametric surface entity (No. 144) see Table 3.

The five geometry entities previously presented are connected through hierarchical relationships, as shown in Fig. 6. This figure presents the sequential procedure performed to extract the necessary information to define each trimmed NURBS surface present in the IGES file format [35]. Then, the first entity to identify in the IGES file is the trimmed parametric surface (No. 144), which holds the highest level of the hierarchy.

4. Vertex normal vector evaluation

Typically the forming tools involved in the numerical simulation of sheet metal forming processes can be considered rigid, which allows to model only their exterior surfaces. The discretisation of the contact surface with the aid of a piecewise linear mesh can be subsequently smoothed using a surface interpolation method, allowing the use of a relatively coarse piecewise mesh for tool description [24,25]. Usually, the tool geometrical model is provided in IGES file format to the mesh generation software, which produces a piecewise mesh over the trimmed NURBS surfaces. Independently of the mesh generation software used, all output mesh files contain both the coordinates of each mesh vertex and the connectivity of each element. However, the application of the Nagata patch interpolation algorithm as surface smoothing method requires also the knowledge of the normal vector of the original surface, in each vertex of the piecewise mesh [17]. In this case, the information available in the IGES file, which was previously used in the mesh generation, should be applied to determine the vertex normal vector. The proposed algorithm to evaluate the normal vector in each vertex consists in the following three steps: surface global search, local search and normal vector evaluation. Each one of these steps is described in detail in the following subsections.

4.1. Global search: ordering of trimmed NURBS surfaces

Globally, the tool geometry is composed by several trimmed NURBS surfaces, which can easily reach several hundreds, particularly in complex models. However, each vertex of the piecewise linear mesh was created over only one NURBS surface that is *a priori* unknown. Since the vertex corresponds to a point of an unknown NURBS surface, the first step of the proposed algorithm is to identify its laying surface. Knowing the surface it is possible to evaluate the coordinates of the vertex in the parametric space of the NURBS surface and, consequently, the surface normal vector at that location. Since the model can be composed by many trimmed surfaces, the aim of the global search is ordering all surfaces composing the model according to their distance to the vertex under analysis, in order to be able to apply the computationally more expensive local search to a limited set of surfaces.

For each mesh vertex, the application of the global search approach requires a loop over all trimmed NURBS surfaces composing the model. Therefore, the strategy followed in the proposed algorithm is to evaluate only simple quantities, i.e., distances between points, avoiding computational inefficient calculations based on the NURBS surfaces definition [23]. In order to evaluate these distances, some points of the trimmed NURBS surfaces are required, which should preferably belong to the trimmed surface domain. Thus, two methods can be followed: (i) use the information available about the trimmed surface domain, as shown in Fig. 7(a); or (ii) use only the information available about the basis NURBS surface (see Fig. 7(b)). In the last approach, Eq. (4) can be applied to build a grid a points, with some predefined values on the parametric domain (u,v), as schematically represented in Fig. 7(b). However, in this study, the global search will be performed adopting the first strategy, i.e. the vertices of each trimmed NURBS surface, defined in the Euclidean space, will be used to represent a simple approximation of the surface boundary.

The IGES file format do not comprise an individual entity with the coordinates of the set of points that defines the vertices of the surface boundaries. However, these vertices can be defined through the end

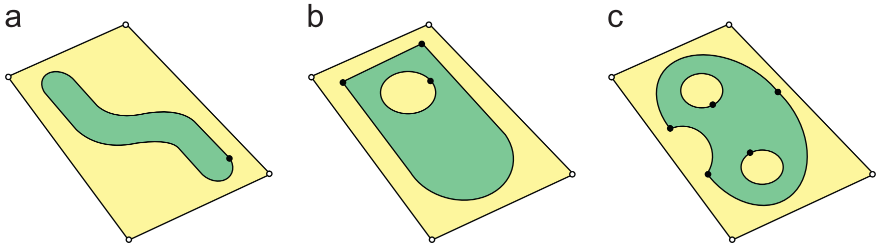

Fig. 8. Example of trimmed NURBS surfaces with outer boundary composed by few trimming curves: (a) one curve; (b) two curves; (c) three curves.

points of the trimming curves that describe the surface boundaries. For each trimmed NURBS surface, the first step is to identify the corresponding entity No. 144. This entity includes the cps^{ob} (cf. Table 3) pointer to the curve on the parametric surface entity (No. 142) defining the outer boundary surface, as well as a pointer for each closed curve on the parametric surface, composing the inner boundaries of the surface. Within each entity No. 142, the cc_{xyz} (cf. Table 3) pointer defines the composite curve entity (No. 102), where the simple NURBS curves entity (No. 126) have their control points defined in the Euclidean space. An important property of the NURBS curves is that any curve always starts in the first control point and ends in the last control point [23]. Therefore, the application of this property allows extracting directly the starting point of each curve from the information available in the entity No. 126 of the IGES specification, avoiding the calculation of additional points. Note that the number of extracted points is equal to the number of trimming NURBS curves (see Fig. 7(a)).

In some particular cases this approach can be inefficient due to the reduced number of points in the definition of the outer surface boundary. Fig. 8 shows three examples of trimmed NURBS surfaces that present a reduced amount of trimming NURBS curves in the outer boundary definition. The problem is more evident in surfaces trimmed by only one curve (see Fig. 8(a)), for which the surface domain cannot be correctly approximated using only one vertex. Therefore, in this work, a special methodology is proposed when the outer boundary of the trimmed surface is defined with less than four trimming curves. In that case, instead of using the trimmed surface vertices, the four vertices of the basis NURBS surface are employed in the global search (represented in Fig. 8 by the hollow points). The same approach is applied for the case of untrimmed NURBS surfaces.

The tools geometry can be composed by many surfaces with a large difference in size (area), particularly in complex models [36]. This can lead to problems in the global search for large surfaces, since the distance between element vertices is much larger than for smaller surfaces. In order to avoid this problem, the Euclidean distance between the mesh vertex and the surface points should be calculated not only for the surface vertices but also for the centroid of the trimmed NURBS surface. However, it is important to mention that it is not mandatory for this auxiliary point to be a surface point, since it will be used only for global search. Therefore, a simple strategy is adopted for its determination, considering it only as a function of the outer surface vertices coordinates in the Euclidean space. Following this approach, the position vector of the approximated centroid c_v of a surface, defined by a set of k vertices, is given by:

$$\mathbf{x}^{\mathrm{c}} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{x}_i^{\mathrm{v}}, \tag{10}$$

where $\mathbf{x}_i^{\mathrm{v}}$ is the position vector of the v_i surface outer vertex. Thus, the total number of points evaluated in the global search for each trimmed NURBS surface is $k+1$.

Following the strategy mentioned above, the first step of the global search algorithm is evaluating the coordinates of the vertices and the corresponding centroid, for each trimmed NURBS surface. This calculation is performed just once at the beginning of the process for all surfaces, being this information used for all mesh vertices. Afterwards, for each mesh vertex, a loop is performed in order to calculate its Euclidean distance to each of the trimmed NURBS vertices and centroids. This allows creating an ordered list of all trimmed NURBS surfaces, for each piecewise linear mesh vertex. The surfaces are ordered by increasing distance, where each trimmed NURBS surface is represented by the nearest point of the set of points evaluated.

4.2. Local search: vertex projection on the NURBS surface

The local search procedure is performed to find both the correct trimmed NURBS surface and the local coordinates of the mesh vertex in the surface domain. Since the local search is computationally more expensive, it is applied according to the ordered surfaces list previously determined, in order to improve its computational efficiency. Note that the local search loop over the list of surfaces terminates when a trimmed NURBS surface meets the requirements, reducing the number of surfaces to be tested. The coordinates of each mesh vertex in the parametric domain of the trimmed NURBS surface are obtained from the orthogonal projection of the mesh vertex on the surface [14,37–40]. The main feature of this projection is the small distance between the mesh vertex and its projected point on the surface, improving the convergence of the projection algorithm. Furthermore, C^2 continuity is assured inside the NURBS surfaces since the tool models provided by the CAD packages are typically very smooth and without sharp edges. All these elements enable the use of the Newton–Raphson method to find the problem solution. Fig. 9(a) presents schematically the projection point P′ obtained from the orthogonal projection of a generic point P on the surface $\mathbf{S}(u,v)$. The distance vector $\mathbf{r}(u,v)$, which connects the point P to an arbitrary point of the surface, is defined as:

$$\mathbf{r}(u,v) = \mathbf{S}(u,v) - \mathbf{x}^{\mathrm{P}}(x,y,z)\ , \tag{11}$$

where $\mathbf{x}^{\mathrm{P}}(x,y,z)$ represents the position vector of point P. Following Piegl and Tiller [23] the orthogonality conditions imposed to the projection are based on the dot product functions, which result in a non-linear system of equations, given by:

$$\begin{cases} f(u,v) = \mathbf{S}_u(u,v)\cdot\mathbf{r}(u,v) = 0 \\ g(u,v) = \mathbf{S}_v(u,v)\cdot\mathbf{r}(u,v) = 0 \end{cases}, \tag{12}$$

where $\mathbf{S}_u(u,v) = \partial\mathbf{S}(u,v)/\partial u$ and $\mathbf{S}_v(u,v) = \partial\mathbf{S}(u,v)/\partial v$ denote the first partial derivatives of the NURBS surface. The system of equations presented in Eq. (12) can be iteratively solved by applying the Newton–Raphson method. In this case, the iteration

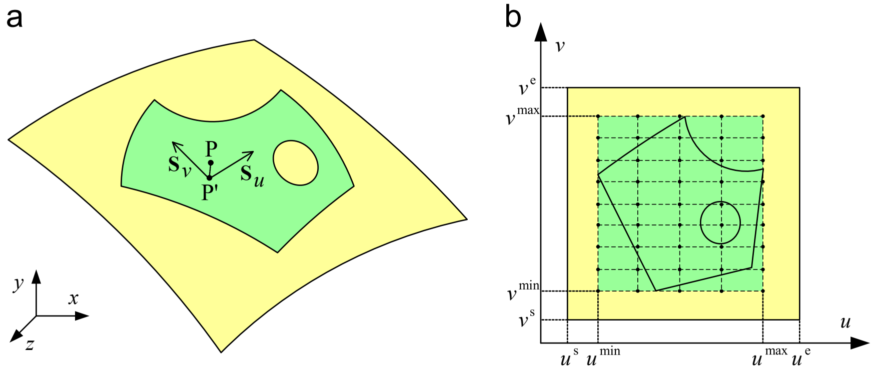

Fig. 9. Local search procedure: (a) orthogonal projection of a point on a surface; (b) grid of points defined in the smallest rectangular domain containing the trimmed surface.

$i+1$ is presented as:

$$\mathbf{X}^{(i+1)}=\mathbf{X}^{(i)}-\left[\mathbf{J}^{(i)}\right]^{-1}\mathbf{F}(\mathbf{X}^{(i)}), \tag{13}$$

with

$$\mathbf{F}(\mathbf{X}^{(i)})=\begin{Bmatrix} f(\mathbf{X}^{(i)}) \\ g(\mathbf{X}^{(i)}) \end{Bmatrix}, \quad \mathbf{X}^{(i)}=\begin{Bmatrix} u^{(i)} \\ v^{(i)} \end{Bmatrix}. \tag{14}$$

The Jacobian matrix of $\mathbf{F}(\mathbf{X})$ at $\mathbf{X}^{(i)}$ is given by:

$$\mathbf{J}^{(i)}=\begin{bmatrix} \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} \\ \frac{\partial g}{\partial u} & \frac{\partial g}{\partial v} \end{bmatrix}^{(i)}=\begin{bmatrix} \left|\mathbf{S}_u\right|2+\mathbf{r}\cdot\mathbf{S}_{uu} & \mathbf{S}_u\cdot\mathbf{S}_v+\mathbf{r}\cdot\mathbf{S}_{uv} \\ \mathbf{S}_u\cdot\mathbf{S}_v+\mathbf{r}\cdot\mathbf{S}_{uv} & \left|\mathbf{S}_v\right|^2+\mathbf{r}\cdot\mathbf{S}_{vv} \end{bmatrix}^{(i)}, \tag{15}$$

where $\mathbf{S}_{uu}(u,v)=\partial^2\mathbf{S}(u,v)/\partial u^2$ and $\mathbf{S}_{vv}(u,v)=\partial^2\mathbf{S}(u,v)/\partial v^2$ denote the second partial derivatives of the NURBS surface, while the mixed partial derivative is $\mathbf{S}_{uv}(u,v)=\partial^2\mathbf{S}(u,v)/\partial v\partial u$ [41]. The convergence rate of the Newton–Raphson method is greatly influenced by the surface geometric characteristics, mainly the surface degree in both parametric directions (p,q), as well as by the initial solution selected for the iterative procedure, $\mathbf{X}^{(0)}$ [38].

In this study, the strategy followed to improve the convergence rate is to use a surface point as close as possible to the P′ as initial solution for the iterative method [38]. Thus, a uniform grid of points is created over each trimmed NURBS surface in order to use the nearest grid point as initial solution. Note that this point must be defined in the local coordinates of the surface, as shown in Eq. (14). The number of points created is related to the number of surface control points in each parametric direction. Hence, for each surface parametric direction (u,v), the number of grid points is given by:

$$N_{GP}=2N_{CP}+1, \tag{16}$$

where N_{CP} is the number of control points in the same parametric direction. Note that in this study, the maximum number of grid points allowed in each direction was limited to 17, in order to control the computational cost of the local search. Since all piecewise mesh vertices are within the trimmed surface domain, the grid of points is generated considering only the smallest rectangular domain covering the trimmed surface domain. Following the same approach used in the global search, this domain is determined using the end points of the trimming curves, defined in the parametric domain of the basis NURBS surface, as shown in Fig. 9(b). These points (control points) can be directly extracted from the IGES file using the information available in each entity No. 126, mentioned in the entity No. 102, which in turn is referred in entity No. 142 through the pointer cc_{uv} (cf. Table 3). Note that for the particular cases of trimmed surface with less than four trimming curves, the same methodology of the global search is employed, which corresponds to using the domain of the basis NURBS surface. Thus, the values that delimit the domain of each surface are defined by the conditions $u\in[u^{\min},u^{\max}]$ and $v\in[v^{\min},\ v^{\max}]$, where the maximum and minimum local coordinate are obtained from the trimming curves vertices evaluated in the parametric domain of the surface. To conclude, the initial solution is the grid point closer to the mesh vertex, obtained based on the Euclidean distance between each point of the grid and the piecewise mesh vertex.

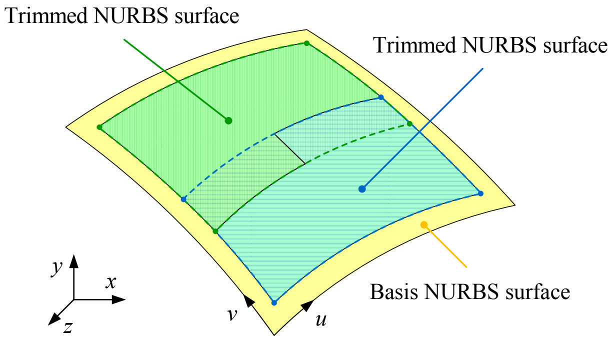

Fig. 10. Overlap of the rectangular domains defined for each trimmed NURBS surface.

The convergence criterion defined for the non-linear system of equations presented in Eq. (12), is based on the simultaneously fulfilment of the four following conditions:

$$\begin{cases} |\mathbf{r}^{(i)}|\le\varepsilon_r^{\text{conv}} \\ |\mathbf{X}^{(i+1)}-\mathbf{X}^{(i)}|\le\varepsilon_x^{\text{conv}} \\ u^{\min}\le u^{(i)}\le u^{\max} \\ v^{\min}\le v^{(i)}\le v^{\max} \end{cases}, \tag{17}$$

where $\varepsilon_r^{\text{conv}}$ and $\varepsilon_x^{\text{conv}}$ are predefined threshold values. Since all mesh vertices are points generated on a trimmed NURBS surface, the distance between the vertex and the surface point is very small and restricted by the parameter $\varepsilon_r^{\text{conv}}$, which limits the amplitude of the distance vector $\mathbf{r}(u,v)$. On the other hand, in order to reduce the possibility of projecting the mesh vertex outside of the trimmed NURBS surface domain, the last two conditions limit the domain of local coordinates (u,v) to a simple rectangular domain, defined as shown in Fig. 9(b). An approximate domain can be adopted since the first condition ensures that the mesh vertex lies on the basis NURBS surface. In fact, this procedure can result in the selection of surfaces wherein the mesh vertex is projected outside the real domain of a trimmed NURBS surface. Fig. 10 presents a simple example where the defined rectangular domains of two trimmed surfaces are partially overlapped. In this situation, a mesh vertex on the overlap region will be projected in the first surface of the list provided by the global search algorithm, since the conditions of Eq. (17) are satisfied for both surfaces. However, since the final goal is evaluating the surface normal vector, which is determined using the information regarding the basis NURBS surface, both trimmed surfaces provide the same normal vector. In fact, the conjunction of the first condition with the last two conditions, in Eq. (17), avoids a wrong selection of the basis NURBS surface used to evaluate the surface normal vector. Hence, if convergence is attained within the maximum allowed number of iterations, *IN*, for a specific trimmed surface of the ordered list, then that surface is selected and the local search process terminates. This means that for each mesh vertex only one solution is obtained, i.e. the first surface that fulfil the conditions presented in Eq. (17).

4.3. Normal vector evaluation

Consider that the projection point P′ determined in the local search algorithm presents coordinates $(\overline{u},\overline{v})$, in the parametric domain of the trimmed NURBS surface. Thus, the unit normal vector of the mesh vertex can be calculated based on the first derivatives of the NURBS surface in the projection point as:

$$\mathbf{n}(\overline{u},\overline{v}) = \frac{\mathbf{S}_u(\overline{u},\overline{v})\cdot\mathbf{S}_v(\overline{u},\overline{v})}{|\mathbf{S}_u(\overline{u},\overline{v})\cdot\mathbf{S}_v(\overline{u},\overline{v})|}\ . \tag{18}$$

5. Algorithm validation: example of mesh smoothing

In this section the proposed algorithm is tested to evaluate the surface normal vector in each vertex of a piecewise linear mesh, considering an example of mesh smoothing with Nagata patches. The selected geometry corresponds to the die of a cross tool deep drawing operation [42], which is discretised with both triangular and quadrilateral piecewise linear elements, using an unstructured mesh. The Nagata patch interpolation is applied to smooth the piecewise mesh using the evaluated surface normal vector for each vertex and afterwards the shape error is determined. Due to geometrical and material symmetry conditions in the numerical simulation, only one quarter of the global model is considered. The dimensions are not presented since they are not required for the analysis presented in this study. Fig. 11 presents the three sequential phases required to obtain the tool model compose by Nagata patches, based on a model provided through the IGES file format (see Fig. 11(a)). The generation of the piecewise linear mesh from the tool model is the first step to be performed (see Fig. 11(b)). Afterward, the proposed algorithm is applied in order to evaluate the surface normal vector in each vertex of the mesh. Finally, the Nagata patch interpolation algorithm is applied to recover the surface curvature of the piecewise mesh, as shown in Fig. 11(c).

5.1. Shape error evaluation

The results of the mesh smoothing operation are dependent on the strategy adopted to compute the normal vectors of mesh vertices [24,25]. Therefore, in order to evaluate the accuracy and efficiency of the developed algorithm, a small angular perturbation $\Delta\alpha$ will be introduced in the normal vectors estimated with the proposed algorithm. The selected values for the perturbation are within the range of the angular deviations typically obtained when the normal vectors are approximated via weighting adjacent facet normal vectors [21,26]. This allows verifying the influence of the surface normal vectors in the Nagata patch interpolation accuracy and evaluating the algorithm efficiency. The direction of the perturbation is generated randomly for each mesh vertex, but with a fixed value for all vertices. The evaluation of the Nagata patch model accuracy is performed through the shape error distribution in each Nagata patch generated. The shape error is determined as:

$$\delta_{Shape} = (\mathbf{P}_{Nagata} - \mathbf{P}_{CAD})\cdot\mathbf{n}_{CAD}, \tag{19}$$

where $\mathbf{P}_{Nagata}$ is the position vector of a generic point over the Nagata patch, $\mathbf{P}_{CAD}$ is the corresponding orthogonal projection on the CAD model (trimmed NURBS surface) and $\mathbf{n}_{CAD}$ is the unit surface normal vector at the projected point.

The selected geometry corresponds to the one presented in Fig. 11, which presents a total of 23 trimmed NURBS surfaces. The maximum number of iterations allowed to solve the non-linear system of equations presented in Eq. (12) was assumed as being equal to $IN = 8$. The predefined threshold values for the convergence criterion were $\varepsilon_r^{\text{conv}} = 0.05$ and $\varepsilon_x^{\text{conv}} = 0.01$ (see Eq. (17)). Fig. 12(a) and (b) presents the shape error distribution of the Nagata patch interpolation applied to the proposed model for two values of angular perturbation, $\Delta\alpha = 0^\circ$ and $\Delta\alpha = 2^\circ$, respectively. Thus, Fig. 12(a) shows the error distribution resulting from the application of the Nagata mesh smoothing method using the normal vectors calculated with the proposed algorithm. The resulting error distribution presents symmetry since the model is also symmetric, although the mesh is not completely symmetric. The maximum positive value of the shape error attained is 0.076 mm in the hyperbolic section of the torus, which presents a fillet radius of 7 mm. The maximum negative value is −0.037 mm and occurs in the elliptic section of the torus, which corresponds to a fillet radius of 21.3 mm. For the flat model surfaces the shape error is null, as expected. On the other hand, the introduction of a small angular perturbation $\Delta\alpha = 2^\circ$ results in the highly asymmetric error distribution, shown in Fig. 12(b). In this case, the initial flat areas of the model present non-null shape error values. In fact, the higher values of shape error occur in the edges of the patches located in originally flat areas of the model. Moreover, the error increases with the increase of the patch dimension, which limits the application of large patches for describing flat surfaces.

In order to study the evolution of the shape error range with the introduced angular perturbation, five non-zero perturbation values were tested. Table 4 summarises both the maximum shape error values attained as well as its range for all tested models. The last row of the table presents the relative increase of the shape error range, taking as reference the model with no angular perturbation ($\Delta\alpha = 0^\circ$). A severe increase of the shape error range is observed with the increase of the angular perturbation value. In fact, a small angular perturbation of 1° leads to an increase in the shape error range higher than 65%, while for an angular perturbation value of 5° the increase in the shape error range is higher than 1000%. The

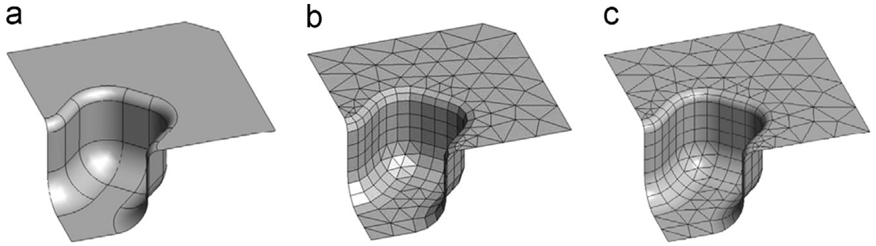

Fig. 11. Die geometry for a cross tool forming process, described by: (a) IGES model; (b) piecewise linear mesh; (c) Nagata patches.

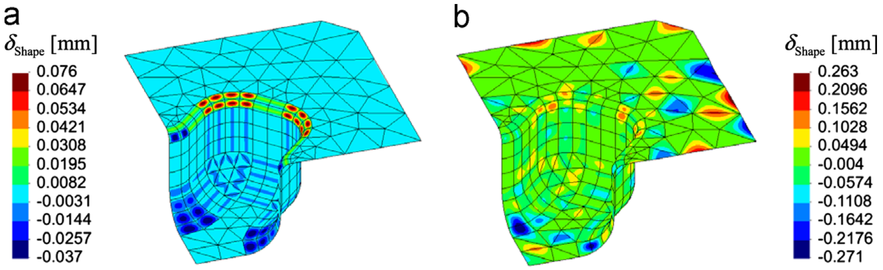

Fig. 12. Shape error distribution on the Nagata patch interpolation with angular perturbation of: (a) $\Delta\alpha=0^\circ$; (b) $\Delta\alpha=2^\circ$.

Table 4
Shape error obtained by the Nagata interpolation for increasing angular perturbation values.

Shape error	$\Delta\alpha=0^\circ$	$\Delta\alpha=1^\circ$	$\Delta\alpha=2^\circ$	$\Delta\alpha=3^\circ$	$\Delta\alpha=4^\circ$	$\Delta\alpha=5^\circ$
Maximum positive [mm]	0.0755	0.1087	0.2629	0.3945	0.5263	0.7333
Maximum negative [mm]	−0.0364	−0.0783	−0.2710	−0.3689	−0.4922	−0.6229
Range [mm]	0.1119	0.1870	0.5340	0.7634	1.0185	1.2742
Range increase [%]	–	67.1	377.1	582.2	810.2	1038.7

normal vectors are accurately evaluated using the proposed algorithm, as shown in the selected example through the high levels of geometric accuracy attained in the Nagata patch interpolation.

6. Conclusions

This paper presents an algorithm to evaluate the surface normal vector in piecewise mesh vertices, based on the information available in the IGES file format. The geometric information is extracted from the IGES files based on trimmed NURBS surfaces definition, which is commonly used in CAD/CAE systems. The application of the Nagata patch interpolation as smoothing method requires an accurate normal vector evaluation in order to avail the full potential of the interpolation. The introduction of small angular perturbations, in the normal vector orientation, leads to large shape errors in the interpolated geometry, relatively to the one obtained with the normal vectors evaluated using the proposed algorithm. In fact, for an angular perturbation of 1° the shape error range can increase more than 65%. The presented algorithm is an efficient and accurate method to combine with the Nagata patch interpolation method, leading to much lower interpolation errors when compared with interpolations that comprise perturbations in the normal vector evaluation. The proposed algorithm can be easily extended to other scientific fields that require local surface properties estimation, which can be evaluated using the information extracted from the IGES file format.

Acknowledgements

The authors gratefully acknowledge the financial support by FEDER funds through the program COMPETE (FCOMP-01-0124-FEDER-010301) and by national funds through the Portuguese Foundation for Science and Technology (FCT) under the projects with reference PTDC/EME-TME/103350/2008 and PEst-C/EME/UI0285/2011. The first author is also grateful to the FCT for the PhD Grant SFRH/BD/69140/2010.

References

[1] A. Santos, A. Makinouchi, Contact Strategies to deal with different tool descriptions in static explicit FEM of 3-D sheet metal forming simulation, J. Mater. Process. Technol. 50 (1995) 277–291.
[2] N. Narasimhan, M.R. Lovell, Predicting springback in sheet metal forming: an explicit to implicit sequential solution procedure, Finite Elem. Anal. Des. 33 (1999) 29–42.
[3] L.F. Menezes, C. Teodosiu, Three-dimensional numerical simulation of the deep-drawing process using solid finite elements, J. Mater. Process. Technol. 97 (2000) 100–106.
[4] P. Wriggers, L. Krstulovic-Opara, J. Korelc, Smooth C^1-interpolations for two dimensional frictional contact problems, Int. J. Numer. Methods Eng. 51 (2001) 1469–1495.
[5] M. Al-Dojayli, S.A. Meguid, Accurate modeling of contact using cubic splines, Finite Elem. Anal. Des. 38 (2002) 337–352.
[6] M.C. Oliveira, J.L. Alves, L.F. Menezes, Algorithms and strategies for treatment of large deformation frictional contact in the numerical simulation of deep drawing process, Arch. Comput. Method Eng. 15 (2008) 113–162.
[7] T.A. Laursen, J.C. Simo, A continuum-based finite element formulation for the implicit solution of multibody large deformation frictional contact problems, Int. J. Numer. Methods Eng. 35 (1993) 3451–3485.
[8] G. Pietrzak, A. Curnier, Large deformation frictional contact mechanics: continuum formulation and augmented Lagrangian treatment, Comput. Methods Appl. Mech. Eng. 177 (1999) 351–381.
[9] D.J. Benson, J.O. Hallquist, A single surface contact algorithm for the post-buckling analysis of shell structures, Comput. Methods Appl. Mech. Eng. 78 (1990) 141–163.
[10] S. Zhuang, M.G. Lee, Y.T. Keum, J.H. Kim, R.H. Wagoner, Improved contact procedure for implicit finite element sheet forming simulation, Int. J. Numer. Methods Eng. 83 (2010) 1759–1779.
[11] M.A. Puso, T.A. Laursen, A 3D contact smoothing method using Gregory patches, Int. J. Numer. Methods Eng. 54 (2002) 1161–1194.
[12] N. El-Abbasi, S.A. Meguid, A. Czekanski, On the modelling of smooth contact surfaces using cubic splines, Int. J. Numer. Methods Eng. 50 (2001) 953–967.
[13] V. Padmanabhan, T.A. Laursen, A framework for development of surface smoothing procedures in large deformation frictional contact analysis, Finite Elem. Anal. Des. 37 (2001) 173–198.
[14] M. Stadler, G.A. Holzapfel, J. Korelc, C^n continuous modelling of smooth contact surfaces using NURBS and application to 2D problems, Int. J. Numer. Methods Eng. 57 (2003) 2177–2203.
[15] H.B. Shim, E.K. Suh, Contact treatment algorithm for the trimmed NURBS surface, J. Mater. Process. Technol. 104 (2000) 200–206.
[16] S.P. Wang, E. Nakamachi, The inside-outside contact search algorithm for finite element analysis, Int. J. Numer. Methods Eng. 40 (1997) 3665–3685.
[17] T. Nagata, Simple local interpolation of surfaces using normal vectors, Comput. Aided Geom. Des. 22 (2005) 327–347.
[18] T. Hama, T. Nagata, C. Teodosiu, A. Makinouchi, H. Takuda, Finite-element simulation of springback in sheet metal forming using local interpolation for tool surfaces, Int. J. Mech. Sci. 50 (2008) 175–192.
[19] D.S. Meek, D.J. Walton, On surface normal and Gaussian curvature approximations given data sampled from a smooth surface, Comput. Aided Geom. Des. 17 (2000) 521–543.
[20] S.-G. Chen, J.-Y. Wu, Estimating normal vectors and curvatures by centroid weights, Comput. Aided Geom. Des. 21 (2004) 447–458.
[21] S. Jin, R.R. Lewis, D. West, A comparison of algorithms for vertex normal computation, Visual Comput. 21 (2005) 71–82.
[22] Initial Graphics Exchange Specification: IGES 5.3, IGES/PDES Organization, US Product Data Association, 1996.
[23] L. Piegl, W. Tiller, The NURBS Book, second ed., Springer, New York, 1997.
[24] D.M. Neto, M.C. Oliveira, J.L. Alves, L.F. Menezes, Local interpolation for tools surface description, AIP Conf. Proc. 1252 (2010) 479–486.

[25] M. Hachani, L. Fourment, 3D contact smoothing method based on quasi-C^1 interpolation, in: G. Zavarise, P. Wriggers (Eds.), Trends in Computational Contact Mechanics, Springer, Berlin Heidelberg, 2011, pp. 23–40.

[26] D.M. Neto, M.C. Oliveira, L.F. Menezes, J.L. Alves, Improving Nagata patch interpolation applied for tool surface description in sheet metal forming simulation, Comput.-Aided Des. 45 (2013) 639–656.

[27] D. Basu, S. Kumar, Importing mesh entities through IGES/PDES, Adv. Eng. Softw. 23 (1995) 151–161.

[28] X. Zhang, J. Wang, K. Yamazaki, M. Mori, A surface based approach to recognition of geometric features for quality freeform surface machining, Comput.-Aided Des. 36 (2004) 735–744.

[29] M.P. Bhandarkar, B. Downie, M. Hardwick, R. Nagi, Migrating from IGES to STEP: one to one translation of IGES drawing to STEP drafting data, Comput. Ind. 41 (2000) 261–277.

[30] L. Piegl, On NURBS: a survey, IEEE Comput. Graph. Appl. 11 (1991) 55–71.

[31] N. Litke, A. Levin, P. Schröder, Trimming for subdivision surfaces, Comput. Aided Geom. Des. 18 (2001) 463–482.

[32] C.C.L. Wang, Y. Wang, M.M.F. Yuen, On increasing the developability of a trimmed NURBS surface, Eng. Comput. 20 (2004) 54–64.

[33] M. Cox, The numerical evaluation of B-splines, J. Inst. Math. Appl. 10 (1972) 134–149.

[34] C. de Boor, On calculating with B-splines, J. Approx. Theory 6 (1972) 50–62.

[35] S.C. Chung, CAD/CAM integration of on-the-machine measuring and inspection system for free-form surfaces, in: ASPE (Eds.), Proceedings of American Society for Precision Engineering, Raleigh, 1999, pp. 267–270.

[36] M.C. Oliveira, J.L. Alves, L.F. Menezes, Improvement of a frictional contact algorithm for strongly curved contact problems, Int. J. Numer. Methods Eng. 58 (2003) 2083–2101.

[37] L. Piegl, W. Tiller, Parametrization for surface fitting in reverse engineering, Comput.-Aided Des. 33 (2001) 593–603.

[38] Y.L. Ma, W.T. Hewitt, Point inversion and projection for NURBS curve and surface: Control polygon approach, Comput. Aided Geom. Des. 20 (2003) 79–99.

[39] S.-M. Hu, J. Wallner, A second order algorithm for orthogonal projection onto curves and surfaces, Comput. Aided Geom. Des. 22 (2005) 251–260.

[40] A.J. Baptista, J.L. Alves, D.M Rodrigues, L.F. Menezes, Trimming of 3D solid finite element meshes using parametric surfaces: application to sheet metal forming, Finite Elem. Anal. Des. 42 (2006) 1053–1060.

[41] M.-C. Tsai, C.-W. Cheng, M.-Y. Cheng, A real-time NURBS surface interpolator for precision three-axis CNC machining, Int. J. Mach. Tools Manuf. 43 (2003) 1217–1227.

[42] G. Maeder, Material forming and dimensioning problems: expectations from the car industry, in: D. Banabic (Ed.), Advanced Methods in Material Forming, Springer, Berlin Heidelberg, 2007, pp. 19–33.