PERGAMON

Computers and Structures 80 (2002) 1399–1408

Computers & Structures

www.elsevier.com/locate/compstruc

Some algorithms to correct a geometry in order to create a finite element mesh

R. Ribó, G. Bugeda *, E. Oñate

International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Campus Norte UPC, 08034 Barcelona, Spain

Received 16 February 2001; accepted 4 May 2002

Abstract

One of the major difficulties in meshing 3D complex geometries is to deal with non-proper geometrical definitions coming from CAD systems. Typically, CAD systems do not take care of the proper definition of the geometries for the analysis purposes. In addition, the use of standard CAD files (IGES, VDA, ...) for the transfer of geometries between different systems introduce some additional difficulties.

In this work, a collection of algorithms to repair and/or to improve the geometry definitions are provided. The aim of these algorithms is to make as easy as possible the generation of a mesh over complex geometries given some minimum requirements of quality and correctness. The geometrical model will be considered as composed of a set of NURBS lines and trimmed surfaces.

Some examples of application of the algorithms and of the meshes generated from the corrected geometry are also presented in this work.

Keywords: Mesh generation; Geometry correction; NURBS

1. Introduction

Mesh generation using structured and unstructured grids is still nowadays one of the bottlenecks in the practical application of the finite element method (see [4,9,11,14]).

Geometrical models created for finite element mesh generation purposes are typically created by the mesh generation software or imported from an external CAD system. In both cases, the geometrical model must satisfy a series of quality constraints.

Normally, geometrical models are constructed as a set of NURBS lines and trimmed surfaces (see [1,3,6–8]). These entities can be mathematically correct and suitable for other uses like visualization or CAM but, in many cases, they are not good enough for meshing operations. In these cases, it is necessary to adapt/repair the geometrical entities by changing their mathematical description while maintaining the same geometrical shape. The final entities should be better suited for the mesh generation operations (see [5,10]).

The algorithms proposed in this paper are considered as a set of filters that will be applied to a geometry definition just after being imported or created. The algorithms are:

- collapse of entities,
- correction of the list of *knots*,
- reparametrization,
- conversion to similar cubic entity,
- union of curves,
- reorientation of the boundary of the surfaces,
- collapse of small angles.

* Corresponding author. Fax: +34-93-401-6517.
E-mail address: bugeda@cimne.upc.es (G. Bugeda).
URL: http://cimne.upc.es/.

0045-7949/02/$ - see front matter
PII: S0045-7949(02)00105-0

With the help of these algorithms, it is possible to automate the process of importing geometry from CAD and mesh on it while maintaining an acceptable level of quality criteria. This process should be performed without the need of manual intervention.

To understand better what follows, a short description of the NURBS lines and surfaces entities is given. A deeper description can be found in [2,12].

1.1. NURBS lines

A NURBS line is a geometrical entity, described as a parametric line in the 3D space, that is defined with a set of *knots*, a set of *control points*, and a set of *weights* if it is rational (Fig. 1).

The *knots* are a list of non-decreasing numbers $u_0,\ldots,u_{L+n}$ that begin in the lower range of the parameter space (usually 0.0) and finish in the high range of it (usually 1.0). A *knot* has multiplicity r if its value is repeated r times inside the list of *knots*. The initial and the end knots must have multiplicity $r = n+1$, where n is the degree of the curve.

The *control points* $\vec{P}_1,\ldots,\vec{P}_L$ are a list of points in the 3D space that are part of the NURBS definition. The line will interpolate the first and the last point and will smoothly approximate the other points.

The *weights* $\omega_1,\ldots,\omega_L$ are a set of non-negative real numbers, one for every control point. They allow the shape of the curve to change and also to have an exact representation of conic curves like circles or ellipses.

The number of *knots*, N_k must be equal to $N_k = N_p + n + 1$, where N_p is the number of control points.

The evaluation of a NURBS curve for a given value of the parameter u can be done in a recursive manner. First, it is necessary to identify the interval $[u_I, u_{I+1}]$ in the list of knots that contains the value of u ($u \in [u_I, u_{I+1}]$). Next, the following recursive expression can be applied:

$$\vec{P}_i^k(u) = \frac{(1-\alpha_i^k)\omega_{i-1}^{k-1}\vec{P}_{i-1}^{k-1}(u) + \alpha_i^k\omega_i^{k-1}\vec{P}_i^{k-1}(u)}{\omega_i^k}$$

$$\text{with} \begin{vmatrix} k = 1,\ldots,n-r \\ i = I-n+k+1,\ldots,I+1 \\ \vec{P}_i^0(u) = \vec{P}_i \\ \omega_i^0 = \omega_i \end{vmatrix} \tag{1}$$

$$\alpha_i^k = \frac{u-u_{i-1}}{u_{i+n-k}-u_{i-1}}$$

$$\omega_i^k = (1-\alpha_i^k)\omega_{i-1}^{k-1} + \alpha_i^k\omega_i^{k-1}$$

then:

$$\vec{s}(u) = \vec{P}_{I+1}^{n-r}(u)$$

1.2. NURBS surfaces

A NURBS surface is the extension of the NURBS line to one additional dimension in the parametric space. Most of the properties of the NURBS curves applies here. There is a *list of knots* for every parametric direction u and v. The *control points* are a set of P_{ij} points with $i \in [1,\ldots,L_u]$ and $j \in [1,\ldots,L_v]$ (Fig. 2).

The evaluation of the surface can be made in different ways:

- To evaluate first the NURBS curve corresponding to one of the parametric directions u (maintaining v constant) and obtain a NURBS line. Then, to evaluate the resulting NURBS curve in the second direction v.

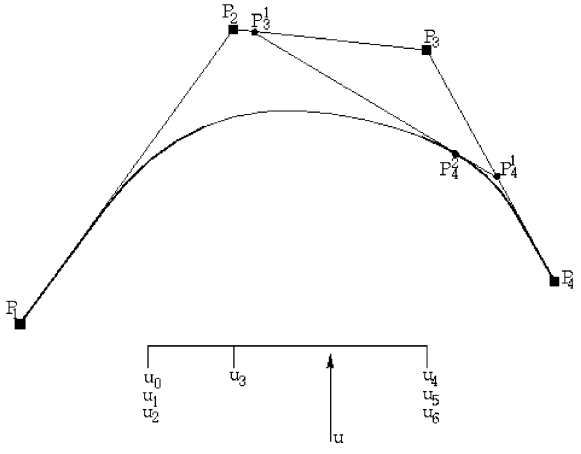

Fig. 1. Example of a quadratic NURBS line with four control points evaluated for a given value of u.

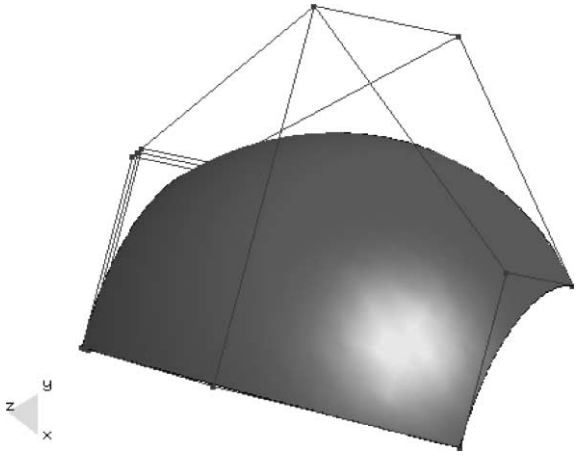

Fig. 2. NURBS surface with the corresponding control polygon.

- Recursive evaluation: similar to the expression (1) given for the NURBS curve.
- Evaluation by means of the *b-spline* base. This technique is also usable for curves.

The evaluation with the *b-spline* base is done by defining a base of *b-splines*, N_i, as a recursive function in the following recursive way:

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \leqslant u < u_i \\ 0 & \text{if not.} \end{cases}$$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i-n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

Then, the following expression is applied:

$$\vec{s}(u,v) = \frac{\sum_i \sum_j \omega_{ij} \vec{P}_{ij} N_i^m(u) N_j^n(v)}{\sum_i \sum_j \omega_{ij} N_i^m(u) N_j^n(v)} \tag{2}$$

2. Collapse of entities

The standard CAD exchange formats (IGES, VDA, ...) do not contain the topological connection between the different surface patches defining a closed geometry. These files just contain the mathematical description of the patches. Nevertheless, in order to proceed with a correct mesh generation it is necessary to check the correctness of the 3D geometry by ensuring, for example, that it is defined by a completely closed surface. In addition, many mesh generation algorithms need to know the neighboring relation between patches in order to advance in their work. For this reason, it is necessary to obtain the topological connection between the different surface patches.

An important additional difficulty is that, very often, the boundary curves defining neighboring patches are very similar, but not identical. Normally, the corresponding pairs of curves are close enough for the different visualization or CAM operations, but not for the necessary operations for the analysis. This makes the task of identifying neighbor patches very difficult.

In order to solve this difficulty it is necessary to define a geometrical tolerance. When two different geometrical entities are separated by a distance smaller than the prescribed tolerance they are considered to be identical and they are collapsed. This tolerance can be specified by the user or can be obtained in an automatic way as a certain percentage of the total size of the geometrical model.

Given a geometrical tolerance ϵ, the collapse criteria for the different geometrical entities are the following:

- Two points are collapsed into a single one if:

  $\|\vec{P}_1 - \vec{P}_2\| < \epsilon$

  where $\vec{P}_i$ is the vector of coordinates of point *i*.
- Two curves are collapsed if they share their end points and the maximum distance between them is smaller than ϵ. In addition, they are also collapsed if a curve belongs to the interior of another one. This last property means that, in addition to the first criterion, its end points $\vec{P}_i$ accomplish

  $\text{dist}(\vec{P}_i, L) < \epsilon$

  where $\vec{L}(t)$ is the second curve. In the latter case, the final result of the collapse operation is not a single curve but a group of curves as shown in Fig. 3.
- Two patch surfaces are collapsed if they share their boundary curves and a similar maximum distance criterion is accomplished.
- Two volumes are considered as equivalent if they share their boundary surfaces.

The computation of the maximum distance between different curves or surfaces is made in an approximated way as follows: a fixed number of points $\vec{P}_i$ are chosen in the interior of the curve or surface depending on their geometrical characteristics. The distance point-curve or point-surface d_i is computed for each of the selected points. Then, $d_{\max}$ is approximated as $d_{\max} = \max(d_i)$.

In general, a big number of control points and a high degree of the line or surface formulations will imply a bigger geometrical complexity. For this reason, the number of selected points will be related to the two mentioned values.

In the computation of the d_i values the $\vec{P}_i'$ point is obtained as the *mapping* of the original point over the

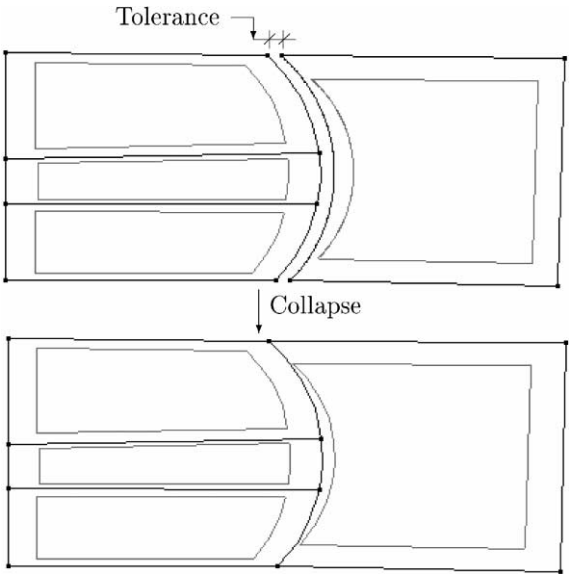

Fig. 3. The collapse procedure reduces the initial four curves to only three.

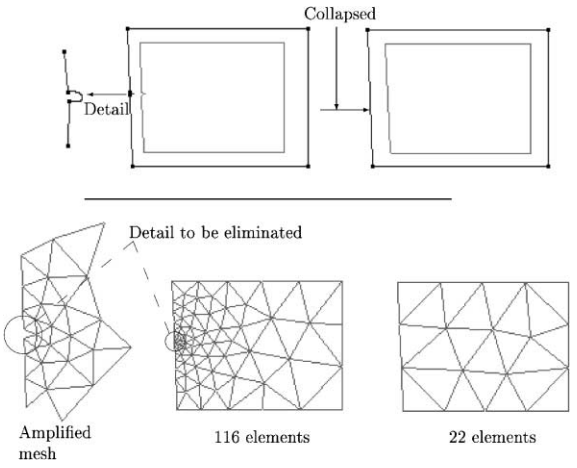

Fig. 4. Saving of elements when unnecessary details are eliminated.

geometrical entity and $d_i = \|\vec{P}_i - \vec{P}'_i\|$. The *mapping* operation is the transformation of an original point to the closest point contained into a geometrical entity like a curve or surface. This procedure is described in [12].

The collapse operation is also useful to eliminate small details of the geometry that can be of no relevance for the analysis. If some lines or surface patches are smaller than the tolerance, they can disappear and their neighbor entities are reconnected. This operation is normally called *feature reduction*. From the practical point of view, many details that are crucial for the design task, can be totally irrelevant for the analysis. This operation can save a lot of elements and degrees of freedom in the final finite element mesh. In Fig. 4 a graphical example of this possibility is shown.

3. Correction of the list of knots

The list of *knots* is a set of non-decreasing real values that belong to the parametric space and are used for the definition of a NURBS. A more detailed definition of *knot* can be obtained in [2].

In the formulation of a NURBS curve or surface the list of *knots* cannot contain any *knot* with a multiplicity higher than the degree of the mathematical entity. Nevertheless, in many cases the standard exchange files contain this type of error. In these cases, it is necessary to correct the list of *knots* by eliminating the overdefined *knot* and the associated weights and points. In this way, if we have a n degree NURBS curve defined by L points with the following list of *knots*:

$$k_0, \ldots, k_i, k_{i+1}, \ldots, k_{i+n+p}, \ldots, k_{L+n} \quad \text{with} \quad k_i = k_{i+1} = \cdots = k_{i+n+p} \tag{3}$$

$$\vec{P}_1, \ldots, \vec{P}_L \tag{4}$$

$$\omega_1, \ldots, \omega_L \tag{5}$$

the list (3) will be reduced to:

$$k_0, \ldots, k_i, k_{i+1}, \ldots, k_{i+n}, k_{i+n+p+1}, \ldots, k_{L+n}$$

and the list of control points (4) and weights (5) to:

$$\vec{P}_1, \ldots, \vec{P}_{i+1}, \vec{P}_{i+p+2}, \ldots, \vec{P}_L \tag{6}$$

$$\omega_1, \ldots, \omega_{i+1}, \omega_{i+p+2}, \ldots, \omega_L \tag{7}$$

Note that the new list will have $L - p$ control points and that the resulting curve will be an approximation to the original one, being the original one a mathematically incorrect representation of a curve.

4. Reparametrization

A very desiderable property for the mesh generation algorithm is to use the arc length as the defining parameter over the curve or surface. This property is not normally available in the NURBS curves or surfaces. This can be partially corrected by using a constant distance between each pair of consecutive control points. Nevertheless, in order to obtain the best possible quality during the mesh generation process it is convenient to improve the parametrization from the very beginning. The corresponding changes that are produced in the curves or surfaces during the improvement of their parametric definition are named reparametrization.

A major problem arises when there are big discrepancies between the spacing of distances between the control points and the spacing between the corresponding associated points of the curves or surfaces. This problem can be detected by a comparison between the modulus of the derivatives in the following way:

$$\frac{1}{F} < \frac{\left\| \left. \frac{\mathrm{d}\vec{L}}{\mathrm{d}t} \right|_{t=k_{i-}} \right\|}{\left\| \left. \frac{\mathrm{d}\vec{L}}{\mathrm{d}t} \right|_{t=k_{i+}} \right\|} < F \tag{8}$$

where the k_i in (8) represents an interior *knot* and F is a maximum acceptable parameter that, for acceptable parametrizations, can have values between 4 and 5. In this formula, $\vec{L}(t)$ represent the curve.

The first step of the correcting process consists of decomposing the curve into the addition of the set of the equivalent Bézier curves. This can be done by inserting multiple *knots* until a multiplicity equal to the order of the mathematical entity is reached. The process of inserting *knots* is described in [2].

From the geometrical and the parametrization points of view, the defined curve is completely equivalent to the

original one. If we have a n degree curve and L defining points with the following values:

$$\begin{matrix} k_0,\ldots,k_i,\ldots,k_{L+n} \\ \vec{P}_1,\ldots,\vec{P}_L \end{matrix} \tag{9}$$

new *knots* are inserted with the same value of the already existing *knots* into (9) until they all have a multiplicity of the same order as the curve

$$\underbrace{k_0,\ldots,k_0}_{n+1},\ldots,\underbrace{k_i,\ldots,k_i}_{n+1},\ldots,\underbrace{k_{L+n},\ldots,k_{L+n}}_{n+1} \tag{10}$$

After inserting these knots, the number of control points of the curve will increase. The new number of points will Bézier equal to the sum of the difference between the original multiplicities and the order of each of the original *knots*.

Taking the list of points and grouping them in sets of $n+1$ points, it is possible to build successive Bézier curves with the same degree than the original curve and the same continuity between successive Bézier curves. These new curves will not depend on the list of *knots* and, therefore, their shape will not change if any of the *knots* is moved. Consequently, the increments between each groups of *knots* can be recomputed in order to obtain a better parametrization. A possible modification based in the *chord length* parametrization consists in creating the following new list:

$$k_0,\ldots,k_n,\ldots,k_i,\ldots,k_{i+n},\ldots,k_{L'},\ldots,k_{L'+n} \tag{11}$$

$$k_0=\cdots=k_n \quad k_i=\cdots=k_{i+n} \quad k_{L'}=\cdots=k_{L'+n} \tag{12}$$

$$k_{i+n+1}-k_{i+n}=\frac{l_c}{l_{tot}} \tag{13}$$

where l_c in (13) is the length of the corresponding Bézier curve and l_{tot} is the total length.

The new curve parametrized in this way will have the same geometrical shape than the original one but with a different parametric definition. Fig. 5 shows the typical improvement than can be obtained with the described algorithm.

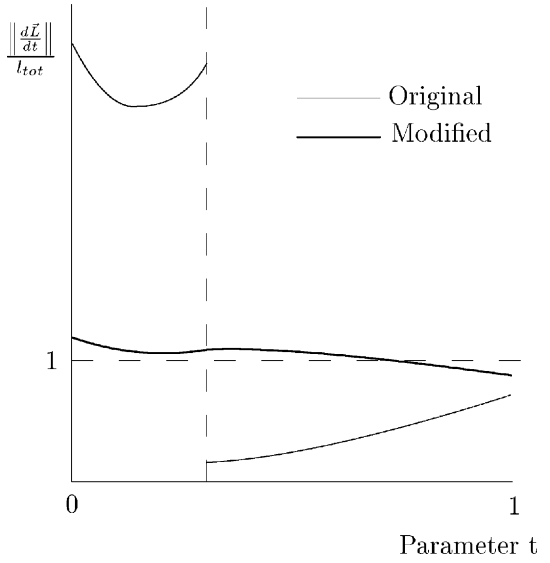

Fig. 5. Comparison between the derivatives of the curve before and after the reparametrization.

5. Conversion to similar cubic entity

In some cases, the correction described in the last section is not enough for ensuring a good parametrization. Typically, these cases arise when one or more control points are repeated, or else when the orders of magnitude of the distances between points have big discrepancies between them.

In these cases it is acceptable to use a new curve or surface not identical to the original one but with a good enough approximation. We should keep in mind that the exchange of CAD files is always carried out with different geometrical tolerances. Hence, it seem logical to allow geometrical changes below the limits of this tolerance.

The process consists of computing a set of points $\vec{P}_i$ belonging to the interior of the curve. These points will be the base of an interpolation algorithm that will be used for the definition of the new curve with the required approximation to the original one. The criteria for the selection of the number of points L_I is obtained by correlating the number of control points L with the accepted tolerance. A new cubic curve can be interpolated from the new set of points using different procedures like the one described in [2]. Fig. 6 shows an

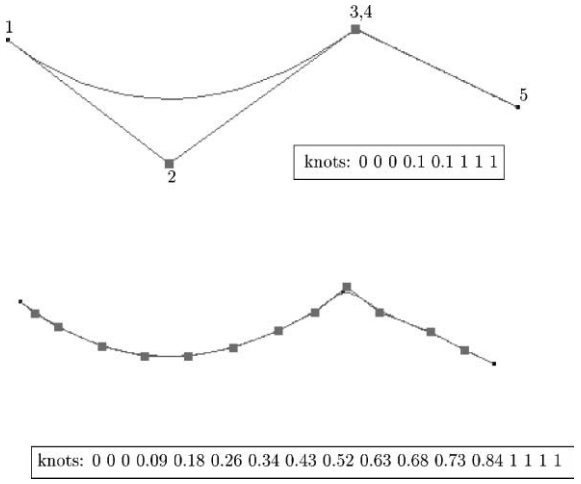

Fig. 6. Conversion of a NURBS to an approximated interpolant cubic NURBS.

example of this type of conversion. In this case, the discrepancy between both curves is high due to a point with C^0 continuity.

6. Union of curves

It is possible to convert a set of connected NURBS curves into a single curve. Normally, it is advantageous to keep the number of curves as small as possible because it can make the mesh generation task easier. Typically, the presence of extremely small segment lines can increase the complexity of the mesh generation.

The criteria for joining different curves are the following:

1. Enough degree of continuity between the different segments must exist. Typically, a C^1 continuity is considered as enough.
2. None of the segments can support any individual surface patch. The set of segments to be joined must belong to the boundaries of the same surface patches.

Fig. 7 shows different examples of these situations.

Before the union can proceed, the different curves must have the same degree. Hence, the degree of all the segments will be increased until they reach the maximum value n. Next, the different control points will be joined and a new list of *knots* will be computed starting from the original ones in the following way:

$$k_{\mathrm{A},0},\ldots,k_{\mathrm{A},L_{\mathrm{A}+n}} \quad \text{curve A} \tag{14}$$

$$k_{\mathrm{B},0},\ldots,k_{\mathrm{B},L_{\mathrm{B}+n}} \quad \text{curve B} \tag{15}$$

The new list will be:

$$\alpha k_{\mathrm{A},0},\ldots,\alpha k_{\mathrm{A},L_{\mathrm{A}+n}},\alpha k_{\mathrm{A},L_{\mathrm{A}+n}}+\beta k_{\mathrm{B},0},\ldots,\alpha k_{\mathrm{A},L_{\mathrm{A}+n}} + \beta k_{\mathrm{B},L_{\mathrm{B}+n}} \tag{16}$$

$$\alpha=\frac{l_{\mathrm{A}}}{l_{\mathrm{A}}+l_{\mathrm{B}}} \quad \beta=\frac{l_{\mathrm{B}}}{l_{\mathrm{A}}+l_{\mathrm{B}}} \tag{17}$$

where l_{A} and l_{B} in (17) are the respective lengths of the curves.

7. Subdivision of curves

Some of the algorithms related with the mesh generation task solve small non-linear systems of equations that involve the derivatives of the analytical expression defining the curves. The presence of strong changes of these derivatives inside the curve can produce convergence difficulties in the solution of the mentioned systems. Due to this reason, it is very convenient to maintain a C^1 continuity over the whole curve (in practice, small angle discontinuities can also be allowed). Hence, it is convenient to subdivide (cut) the original curves at points not satisfying the required continuity.

This subdivision is made through the insertion of additional *knots* at the required cutting points until a multiplicity equal to the order of the curve is reached. The new curves will be:

$$k_0,\ldots,\underbrace{k_i,\ldots,k_i}_{n+1},\ldots,k_{L+n}$$

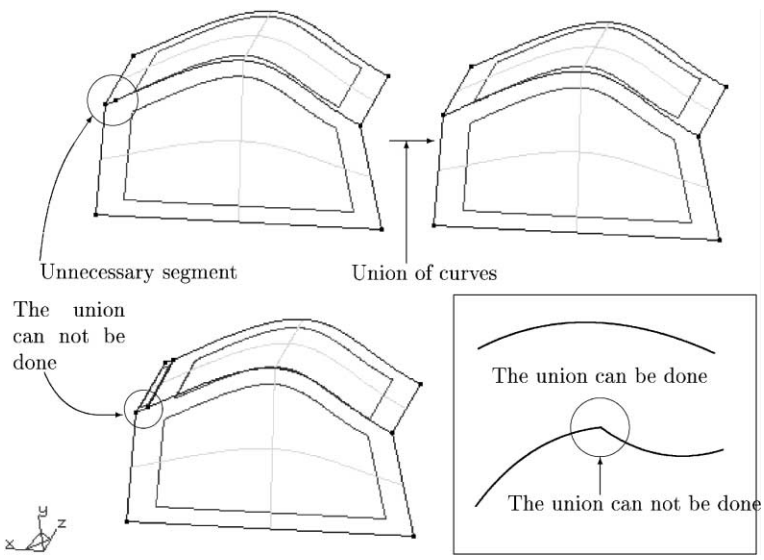

Fig. 7. Different criteria for accepting (or not) the union of curves.

$$\frac{k_0}{k_i},\ldots,\frac{k_i}{k_i} \qquad \text{curve A}$$

$$\frac{k_{i+n+1}-k_{i+n+1}}{k_{L+n}-k_i},\ldots,\frac{k_{L+n}-k_{i+n+1}}{k_{L+n}-k_i} \qquad \text{curve B}$$

The control points will be simply spread depending on the number of *knots* corresponding to each curve.

8. Reorientation of the boundary of the surfaces

Some of the mesh generation algorithms require a proper orientation of all the boundary curves. A curve $\vec{L}(t)$ that belongs to the boundary of a surface with a normal vector $\vec{N}$ is considered as well oriented if the vector $\vec{V}$ defined by

$$\vec{V}=\vec{N}\times\frac{\mathrm{d}\vec{L}}{\mathrm{d}t}$$

always points towards the interior of the surface.

In some cases, the information imported from a CAD system does not satisfy the mentioned criterion. It is then convenient to check this possibility and to change the corresponding orientation when necessary (Fig. 8).

Nevertheless, some times it is not easy to identify the interior of a surface. In this work, we use the fact that for non-trimmed surfaces the curves must belong to the boundary of a NURBS surface. For trimmed surfaces, it is necessary to look for any of the trimming curves placed on the surface.

A boundary line of a NURBS surface patch is well oriented if:

$$\frac{\mathrm{d}\vec{L}(t)}{\mathrm{d}t}\cdot\frac{\delta\vec{S}(u,v)}{\delta u}>0 \quad \vec{L}\in v=0 \tag{18}$$

$$\frac{\mathrm{d}\vec{L}(t)}{\mathrm{d}t}\cdot\frac{\delta\vec{S}(u,v)}{\delta u}<0 \quad \vec{L}\in v=1 \tag{19}$$

where $\vec{S}(u,v)$ is the parametric surface. If the curve lays on $u=0$ or 1 the corresponding similar expressions will be used.

An additional convenient check consists of computing the normal vector to the surface in its center and to compute the approximated normal vector to the boundary curves. The approximate normal vector can be computed as:

$$\vec{N}_L=\sum_i \vec{L}(t_i)-\vec{C}\times\vec{L}(t_i+\Delta t_i)-\vec{C} \tag{20}$$

where $\vec{C}$ is the center of the surface. The boundary curves will be well oriented if:

$$\vec{N}\cdot\vec{N}_L>0$$

9. Collapse of small angles

A very common problem in files imported from CAD systems is that some of the surfaces contain almost null angles that make impossible the generation of a proper mesh. In those cases, it is convenient to collapse part of the lines that define these angles until converting the angles into bigger ones allowing to place acceptable mesh elements.

The collapsing angle will be computed depending on the given tolerance ϵ. In general, the resulting criterion should guarantee that the size of the resulting curve is in agreement with the rest of the contiguous curves (see Fig. 9).

10. Examples

The algorithms presented in previous sections have been implemented into the GiD pre/postprocessing system [13] developed at CIMNE. The following examples are a set of representative applications of these

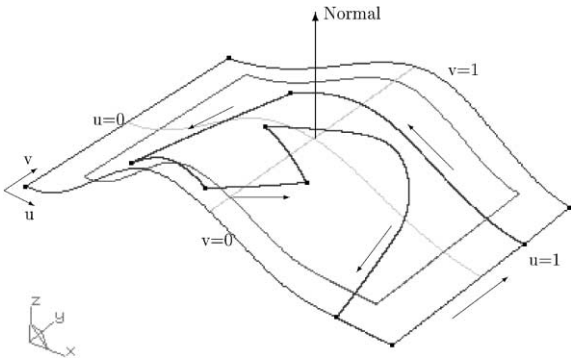

Fig. 8. Criterion of signs and orientations of the trimming lines of a trimmed surface.

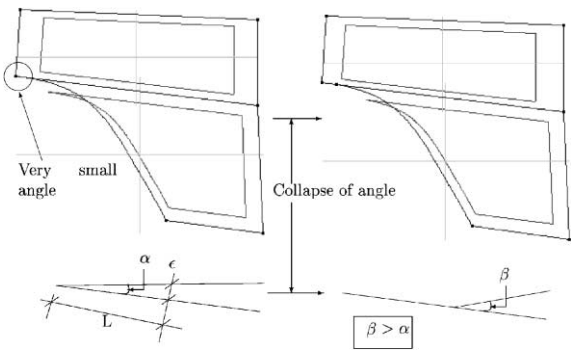

Fig. 9. Collapse of a too small angle. The collapse will be accepted if L is bigger than a minimum value.

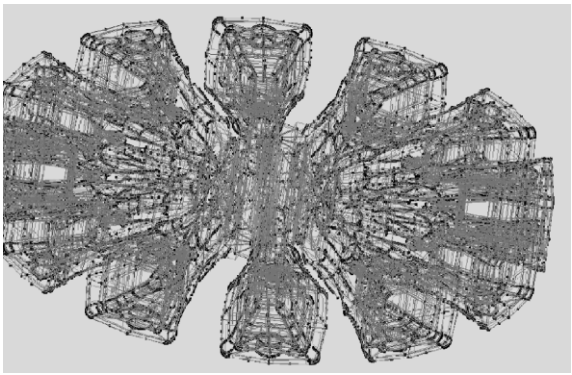

Fig. 10. Geometry of a set of casting teeth for construction machines. This set is the one used in the casting process.

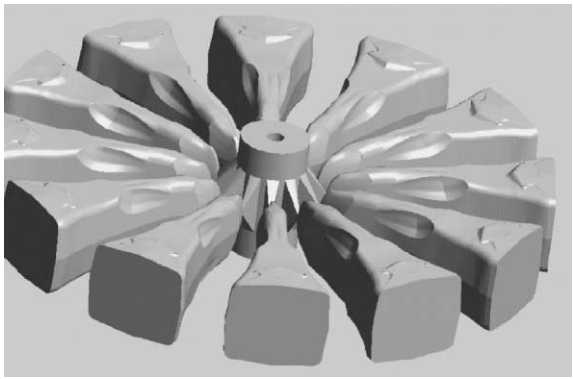

Fig. 11. Fotorrealistic render of the previous geometry.

algorithms for the preparation of different geometries in order to be meshed. Typically, the geometry has been defined using a CAD system and the corresponding information has been introduced into GiD using standard CAD files (IGES, VDA, ...). In all cases, the use of the presented algorithms has carried out the improvements/reparations needed to allow meshing of the geometry without the necessity of any additional operation.

10.1. Geometry of a set of casting teeth

This example (see Figs. 10 and 11), corresponds with the generation of a finite element mesh over the solid model of a casting set of teeth. The number of surface patches used for the geometry definition is around 400. In this particular case, before the mesh generation it has been necessary to correct the geometry in order to create a closed volume (without gaps). The final obtained mesh has around 40,000 elements.

10.2. Tarazona cathedral

Figs. 12 and 13 represent a structural analysis model with a non-linear damage material model of the Tarazona cathedral. In this case, it has been necessary to correct many of the surfaces used to describe the fine

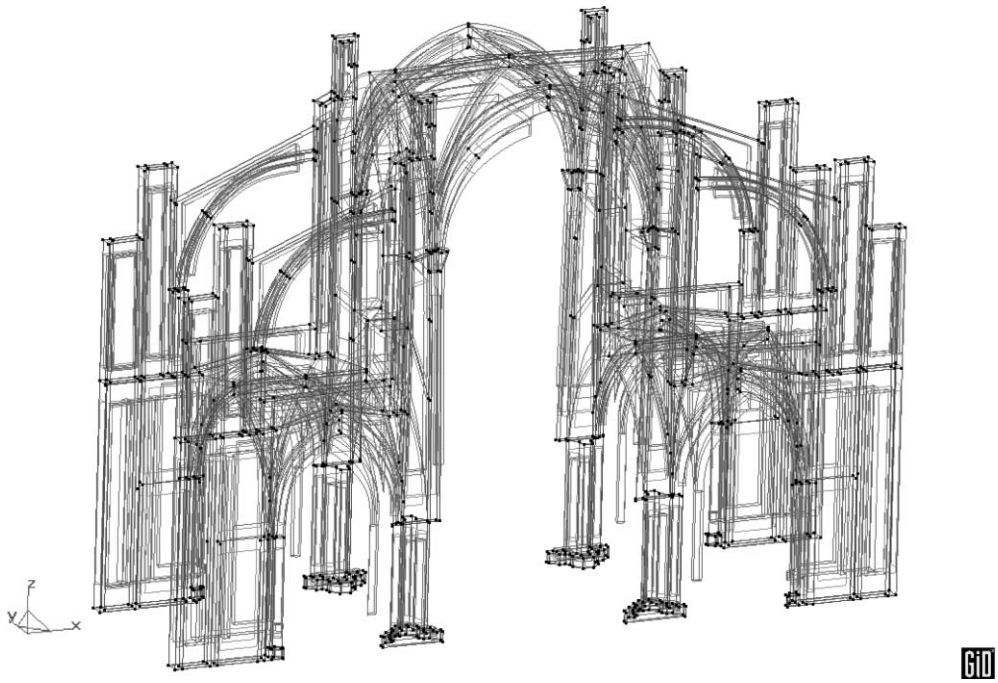

Fig. 12. Geometry of the Tarazona cathedral (Spain).

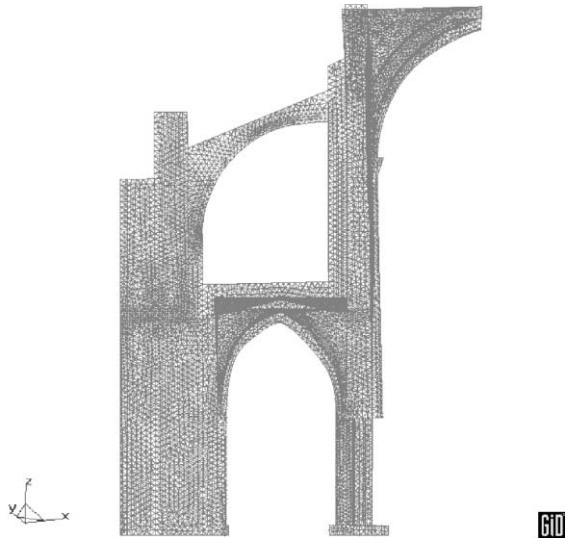

Fig. 13. Mesh of the cathedral.

details of the shape in some parts of the geometry. The final mesh has around 40,000 elements.

10.3. Sheet stamping processes

This is a typical case corresponding with a sheet stamping analysis (see Fig. 14). The geometry has been modeled using traditional CAD systems and before proceeding to the classical mesh generation operations it has been necessary to use all the correction algorithms presented in these pages. Some of the problems that usually appear in this type of geometries are:

- some surfaces are too small and must disappear,
- other surfaces have a bad mathematical description,
- some of them are correctly defined but with some properties that are not suitable for meshing on them,
- some of the geometrical details are very small and produce extremely complex geometries.

Fig. 14 shows a typical mesh for this type of cases with around 400,000 finite elements.

10.4. Aluminium casting

These type of geometrical models have the same problems as the previous one. However, they are more complicated to improve due to the inherent complexity of their surface patches (see Figs. 15 and 16).

11. Conclusions

CAD models are not always best suited as starting points for mesh generation. CAD models and the

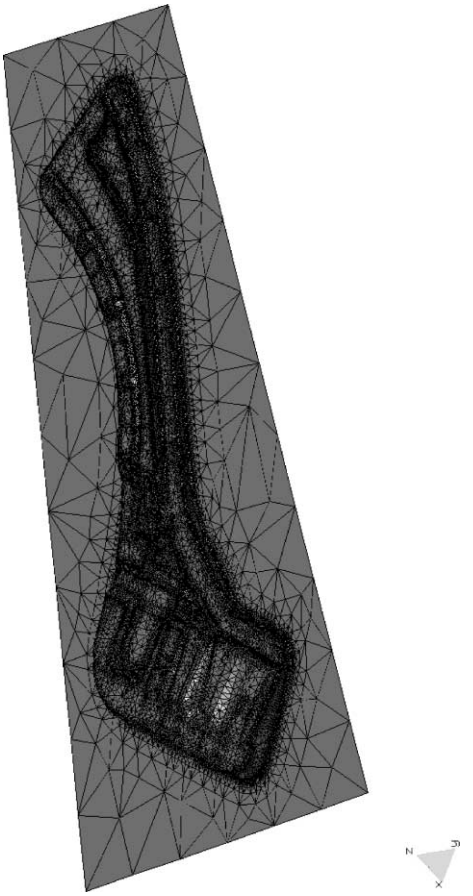

Fig. 14. Analysis of sheet stamping processes.

Fig. 15. Mesh of an aluminium casting model.

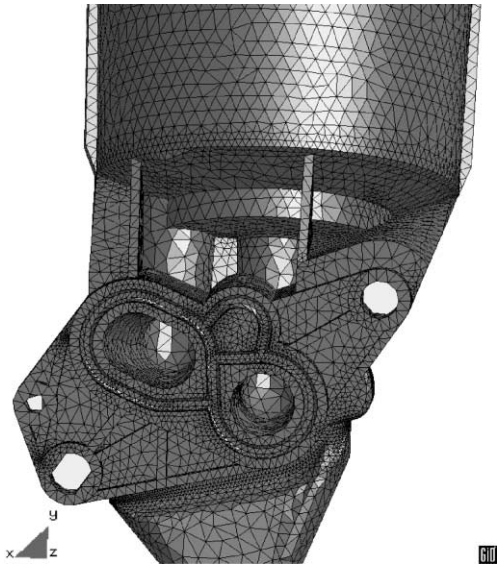

Fig. 16. Detail of the previous mesh.

corresponding exchange files are more suited for design and CAM operations than for the analysis task. For this reason, it is normally necessary to repair the CAD models in order to make them suitable for mesh generation.

The proposed algorithms have shown to be efficient in dealing with three different problems:

- The elimination of some small features that are necessary for the design but are not important for the numerical analysis.
- The correction of some mistakes and accuracies that are found in many geometrical interchange files.
- The change in the mathematical definition of some entities in order to make them better suited for mesh generation.

All these algorithms can be executed automatically as a filter to the imported or created geometry reducing the human interaction to a minimum and facilitating a lot the mesh generation task.

An academic version of the GiD pre/postprocessing system, where the algorithms presented in this paper have been implemented, can be freely downloaded at http://gid.cimne.upc.es.

References

[1] Knuth DN. In: The art of computer programming, vol. 3. Addison-Wesley; 1973.
[2] Farin G. Curves and surfaces for CADG. 3rd ed. New York: Academic Press Inc; 1993.
[3] Foley JD, van Dam A, Feiner SK, Hughes JF. Computer graphics. Principles and Practice. 2nd ed. Addison-Wesley; 1993.
[4] George PL. Automatic mesh generation. Application to finite element methods. New York: Wiley; 1991.
[5] Le Gouez JM, Boheas MA, Miles CJ. Data requirements for analysis modelling. Technical report, NAFEMS, 1995.
[6] Hoschek J, Lasser D. Fundamentals of computer aided geometric design. Addison-Wesley; 1993.
[7] Lancaster P, Salkauskas K. Curve and surface fitting. 1st ed. Academic Press; 1988.
[8] Levin A. Interpolating nets of curves by smooth subdivision surfaces. Computer Graphics Proceedings (SIGGRAPH) 1999:57–64.
[9] Löhner R, Parikh P. Generation of three-dimensional unstructured grids by the advancing front method. Int J Numer Meth Fluids 1988;(8):1135–49.
[10] NAFEMS. Integrate cad and analysis, 1996.
[11] Oñate E. Cálculo de estructuras por el método de los elementos finitos. CIMNE 1995.
[12] Ribó R. Development of an integrated system for the geometry dealing, mesh generation and data for the finite element analysis. Ph.D. Thesis, Universitat Politécnica de Catalunya, 2000.
[13] Ribó R et al. GiD reference manual. CIMNE, 1998.
[14] Zienkiewicz OC, Taylor R. In: The finite element method 5th ed, vol. I. John Wiley and Sons; 2000.