If the torsion is sufficiently small (the radius of torsion T is large), we can neglect the gap according to $\vec{b}$ to the plane $[\vec{\tau}, \vec{\nu}]$ and, locally, use the meshing method in the plane. In the opposite case, the parabola of the plane $[\vec{\tau}, \vec{\nu}]$ is not a good approximation of the curve and the analysis must be extended to the next term so as to account for the torsion. Thus we approach the curve by a cubic of $\mathbb{R}^3$ and we control the gap between this cubic and the curve via the following term in the development. It is obvious that the calculations required are very technical and, for that reason, we will go no further into this topic.

14.5.2 Curves of a parametric surface

We distinguish here several types of curves :

- the curves interface between patches,
- the curves traced on a patch,
- the curves images of a segment in parametric space which, approached by a segment, form the edges of the surface mesh elements.

All these points will be dealt with in the next chapter.

Chapter 15

Surface meshing and re-meshing

Introduction

In this chapter, we discuss some methods that make it possible to construct the mesh of a segment or a curve traced on a surface, as well as the mesh of a surface itself. As in the previous chapter, there are two possible approaches to dealing with this meshing problem, depending on the definition of the geometric model. In practical terms, the problem is of a different nature, depending on whether a CAD type definition is known or the sole data available corresponds to a discretization (*i.e.*, a polyhedral approximation) of the given surface. The first case leads rather to envisaging the development of appropriate mesh generation methods, while in the second case, mesh modification methods will be preferred.

Mesh generation for surfaces representing the boundaries of a three-dimensional domain is, undoubtedly, a crucial point, especially for the computational schemes in finite element methods. As well as this field of application, the construction of surface meshes is also an important feature in applications such as visualization, virtual reality, animation, etc.

This chapter contains several sections. First, we briefly discuss curve meshing (already mentioned in Chapter 14), in the particular case where they are traced onto a surface (or along the sides of a patch, for example).

Then, using some examples, we illustrate the different types of surface meshes that can be constructed, *a priori*. These examples allow us to specify the nature of the given problem and to indicate the various possible approaches. We will see that there exists a direct approach (where the surface itself is meshed) and an indirect approach (where a mesh is constructed in a parametric space and then mapped onto the surface).

We have decided to exclude here the case of implicit surfaces (discussed in Chapter 16) and explicit surfaces[1], and focus on the case of a surface defined by one or more patches (*i.e.*, a parametric surface). To this end, we examine first the case of a single patch, before considering composite surfaces. In the latter situation, we discuss a possible alternative, depending on whether the resulting mesh is *patch-dependent* or not.

Then, we change data types to focus on surfaces defined in a discrete way, *i.e.*, from a mesh. We study an approach that tends to provide an adequate mesh of the given surface, based on a series of modifications of the initial mesh, which forms the sole data of the problem.

All that has been mentioned, except the last case, assumes that the patches defining the surface are correct (in particular, that global conformity between patches is ensured, and that there are no overlaps, no holes, no folds nor any other undesirable artifact). To conclude, we deal with what can be termed a *true problem*. In practice, the surfaces to be meshed are usually not suitably defined with regard to the desired objective. Geometric modeling systems (C.A.D.) allow us to define composite surfaces from a (bottom-up or top-down) analysis for which the constraints (imposed by the manufacturing or visualization process) are rather different from those required by the meshing techniques. Among the divergences, let us mention the problem related to the numerical accuracy of the models.

15.1 Curve meshing (curve member of a surface)

The curves that are of interest here are essentially the curves interface between two patches and/or the curves traced on a surface.

The different methods for curve meshing have been reviewed in Chapter 14 and the reader is referred to this chapter. In this section, we will simply make some practical remarks about the specificities of this problem, *i.e.*, when we consider that the curves are part of a surface.

We should immediately point out that a classical meshing technique for curves in $\mathbb{R}^3$ (such as described in Chapter 14) is still valid here, provided that care is taken with :

- the local curvatures of the surface (in the vicinity of the curve) and/or,
- potential specifications and, obviously,
- the envisaged application.

Indeed, from the strictly geometric point of view, the mesh of a straight segment requires a unique element. However, this straight segment is a (potential) edge of a surface mesh triangle that forms a polyhedral approximation of the surface. Hence, the edge lengths are proportional to the principal radii of curvature of the surface. In the isotropic case, the construction of an equilateral element then needs to

[1]Such a surface can always be expressed in a parametric form (as indicated in Chapter 11), thus making it possible, in principle, to apply the techniques described in this chapter.

bound the mesh element size. This operation can be performed by considering, not only the intrinsic properties of the curve, but also those of its close neighborhood. In the anisotropic case, the reasoning is quite similar, except that this time the classical equilateral (Euclidean) element is not longer the aim.

When the curve considered is an interface curve between two patches (a boundary curve), the meshing problem is rather similar. The desired edge size of the discretization also depends on the local curvatures of the curve and of its two (or more) adjacent surfaces.

15.2 First steps in surface meshing

In this section, we try to give a global view of surface mesh generation and we make some remarks about this interesting problem. To this end, we start by introducing several examples of surface meshes obtained using different mesh generation techniques[2]. This will then allow us to state the expected properties of such meshes. Finally, we introduce various possible solutions (the main techniques employed being described in detail later).

15.2.1 Various approaches to surface meshing

Before going further in the discussion of the principle, we will illustrate, on a given geometry[3], several examples that are representative of different mesh related aspects (Figures 15.1 to 15.2).

Surface mesh by example. Let us focus first on structured mesh generation. The mesh in Figure 15.1 (left-hand side) was obtained using a multi-block type method, the different patches having been meshed using a transfinite interpolation on a quadrilateral (Chapter 4). In such an approach, the geometry of the surface is simply governed by the geometry of the four sides of each patch. Therefore, the surface mesh is defined only by the discretization of the boundary edges (composed here by 10 segments).

The mesh in Figure 15.1 (right-hand side) was generated by mapping a regular grid (a 10×10 structured grid) onto the surface. Hence, in contrast to the previous case, all resulting mesh vertices belong to this surface.

Let us now turn to unstructured meshing techniques based on the use of parametric spaces associated with the patches. The mesh in Figure 15.2 (i) was constructed by mapping onto the surface uniform isotropic meshes in the parametric spaces.

The mesh in Figure 15.2 (ii) was obtained by mapping onto the surface anisotropic meshes in the parametric spaces. These meshes were constructed so as to ensure that the resulting mesh, after mapping, will be uniform.

[2]The aim here is not to examine the different approaches used in detail, but rather to point out the key problems in surface mesh generation.

[3]The perspicacious reader may have recognized the spout of the well-known *Utah teapot*, composed of four parametric patches.

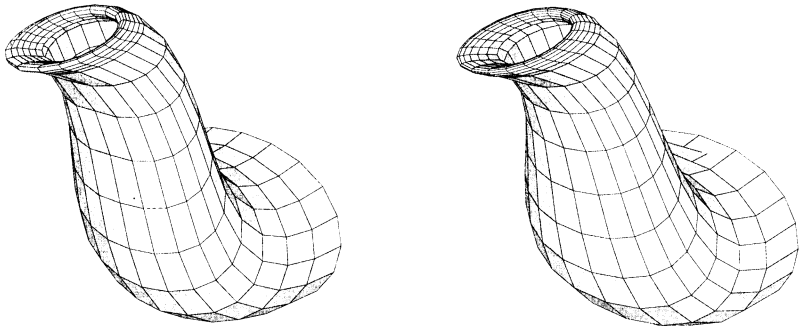


Figure 15.1: *Multiblocs method based on a local transfinite interpolation (left-hand side). Naive parametric mesh, uniform grid, and projection onto the surface using the surface definition itself.*

The mesh in Figure 15.2 (iii) was obtained by mapping onto the surface the anisotropic meshes of the parametric spaces, controlled by the minimal radius of curvature (so as to bound the angles between the mesh edges and the tangent planes to the surface).

The mesh in Figure 15.2 (iv) was obtained by mapping the anisotropic meshes of the parametric spaces, controlled by the two principal radii of curvature.

The mesh in Figure 15.2 (v) was obtained by mapping the anisotropic meshes in the parametric spaces, controlled this time by bounding the ratio between the two principal radii of curvature.

Finally, the mesh in Figure 15.2 (vi) was obtained in a similar fashion, by controlling more subtly the use of the two radii of curvature.

It is thus easy to realize that numerous possibilities exist to obtain a surface mesh, some of which may be more relevant than others. Therefore, the question arises :

"What is really needed and what has to be done to get it ?"

We will now try to answer this tedious and haunting question as precisely as possible.

15.2.2 Desired properties

As can be easily imagined, the desired surface mesh must verify certain properties. Obviously, or rather *a priori* (as will be seen later), the vertices of the mesh elements must belong to the surface (that is to the geometry, defined in one way or another).

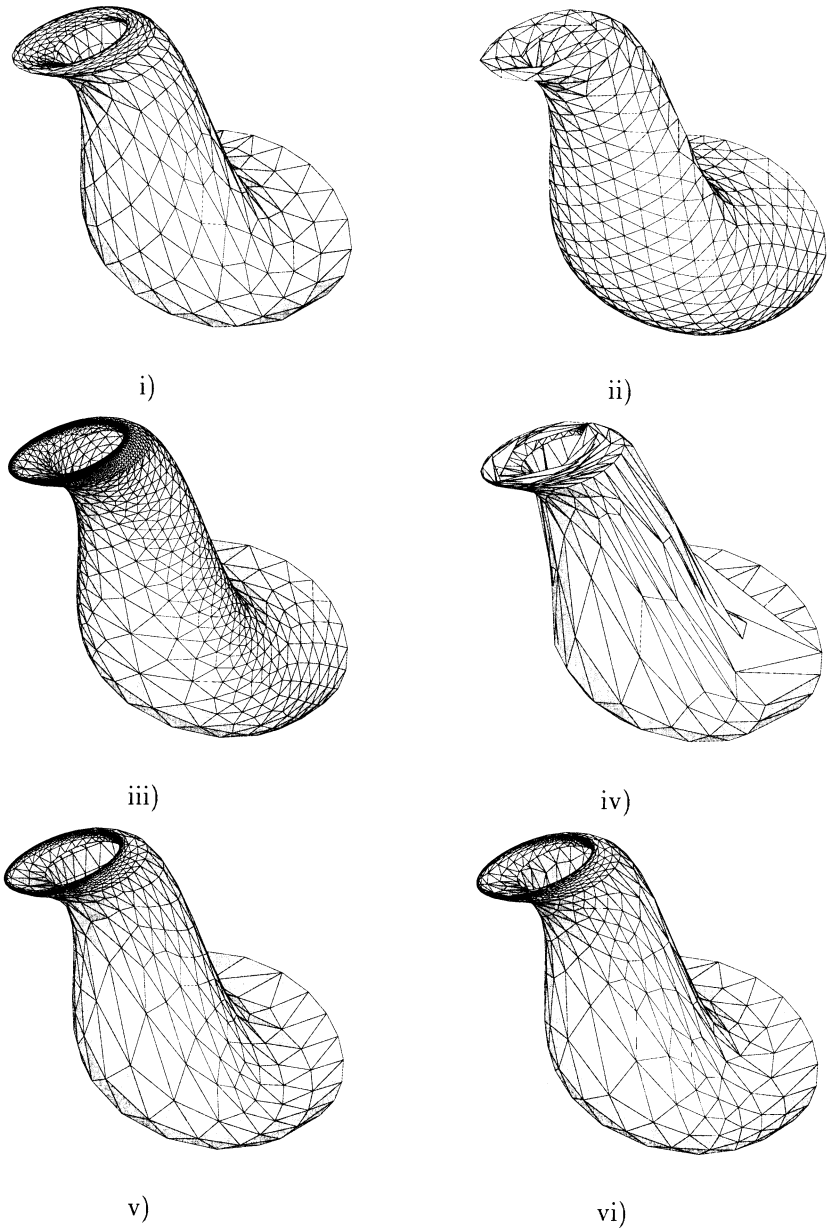


Figure 15.2: *Unstructured meshes : i) Uniform unstructured mesh in the parametric space and mapping, ii) unstructured mesh projected onto the surface to obtain a uniform mesh at this level, iii) Isotropic mesh based on the minimum of the radii of curvature, iv) Anisotropic mesh without a stretching limit, v) Anisotropic mesh with a (user-specified) control on the ratio between the two principal radii of curvature, vi) Anisotropic mesh with an automatic adjustment of the maximal ratio between the two principal radii of curvature.*

Let Σ be the given surface and let $P, Q, R, \ldots$, be the vertices of mesh $\mathcal{T}$. Thus, we want to have :

$$\forall P \in \mathcal{T} \implies P \in \Sigma, \tag{15.1}$$

As certain methods do not guarantee this feature (for all vertices in $\mathcal{T}$), we can simply require that the minimal distance between any point P and Σ is as small as possible. In this case, we introduce an accuracy threshold ε that controls this gap.

Nevertheless, this feature is only a usually desired necessary condition (cf. Chapter 14) that does not guarantee however that the resulting mesh will be satisfactory. Hence, a stronger constraint can be expressed. For instance, we can require all mesh edges to be as close to the true surface as possible. In this case, for each edge PQ of $\mathcal{T}$, we want to have :

$$d(PQ, \Sigma) \leq \varepsilon, \tag{15.2}$$

where ε represents the desired tolerance value and $d(PQ, \Sigma)$ is the maximal distance (the gap) between the edge PQ and the surface Σ.

Pursuing the analysis, it appears that Property (15.2) is a necessary although not a sufficient condition. The following property comes to reinforce this condition :

$$d(PQR, \Sigma) \leq \varepsilon, \tag{15.3}$$

for each triple of points (P, Q, R), vertices of a triangle of $\mathcal{T}$ (simplicial mesh). This relation indicates that the distance (the gap) between any triangle PQR and the surface Σ must be bounded. For quadrilateral (or hybrid) meshes, we have a similar property.

Finally, the surface mesh must have the same aspect as the surface (*i.e.*, it must reflect the geometry). Hence, if the surface is locally of order C^1 or G^1, the mesh must represent this feature as closely as possible (possible oscillations must be avoided in this case). Similarly, discontinuities (singular points, ridges or crest lines) must be present in the resulting mesh. These requirements can be formulated more concisely as follows :

`Smooth surface mesh.` (15.4)

Exercise 15.1 *Find some examples where Property (15.1) holds while Property (15.2) is not satisfied. Similarly, discuss the relation between the two other properties.*

Remark 15.1 *Note that the above properties concern only the geometrical point of view of the surface. In the case where element sizes (densities, directions, etc.) are specified, this constraint must be coupled, in some sense, with the previous ones.*

To conclude, the problem here is to express the desired properties in terms of requirements that are usable by the mesh generation method considered. Thus, it is easy to imagine that a suitable method is one that can be controlled up to a certain level (via a metric field).

15.2.3 Direct surface meshing

A *direct* mesh generation method is a method acting directly at the level of the surface considered (without using, for example, a parametric space). Among the most commonly used techniques, we find some of the methods already discussed in the general chapters.

Method related to a specific definition. When the surface is defined in a particular way, for example as a *sweeping surface*, it is possible to take advantage of this feature to construct the surface mesh. Thus, if the surface is constructed by sweeping a curve Γ_1 along a curve Γ_2, a mesh can be easily obtained by sweeping a discretization of Γ_1 along a discretization of Γ_2 (or conversely).

This type of method is obviously limited in its range of application and, moreover, does not allow the quality of the geometric approximation (the mesh) of the true surface to be effectively controlled. Moreover, special care must be taken regarding potential degeneracies that can appear locally on the resulting surface [Rogers,Adams-1989].

Extension of a two-dimensional method. An algebraic method (when the surface defines a quadrilateral-shaped domain) or an advancing-front type method (Chapter 6) can be extended to process surfaces. This extension feature is less obvious for a Delaunay type method (Chapter 7), which is more adapted to meshing a surface by meshing the associated parametric spaces.

For an algebraic method (Chapters 4 and 13), the resulting surface is strongly related to the generation method. In fact, the geometry of the surface is defined by the method, usually from the mesh of the domain boundaries only. Notice that no explicit control of the gaps (at the level of the edges or faces) between the discretization and the surface can be obtained. Therefore, this type of method is of interest for particular geometries (although the low cost may be considered an advantage). From this point of view, parts of surfaces are frequently well processed by such approach (a more sophisticated method then being unnecessary, or even too costly, to implement).

The advancing-front approach, or more precisely its extension to surface processing, is more general[4]). This type of approach is also widely used in industrial meshing technologies.

The main principle of this method follows, broadly speaking, the classical scheme of an advancing-front method in two dimensions (see [Löhner,Parikh-1988], [Nakahashi,Sharov-1995], etc.) :

- identify the key features of the boundary of the patch considered. These points are those making it possible to define a set of almost regular lines along the side of the patch. We then naturally retrieve the singular points of the surface among those points,

[4]Notice by the way that it is common to find, when dealing with the extension of a known method, the assertion that this operation is both natural and easy. In practice, this judgment must be strongly qualified, the difficulties encountered not being of the same order.

- define a mesh vertex for each of these points and identify the set of lines joining two such vertices,
- mesh these lines (refer to the different curve meshing techniques discussed previously),
- connect the meshed lines together so as to define closed curves[5],
- use in each of the parts so defined a classical advancing-front approach. From a front edge (the initial front being formed by the discretized lines of the contour), construct (or choose) a candidate point on the surface to form an equilateral triangle. Several ways exist to construct (choose) a candidate point. The simplest idea is to consider, for a given front edge, an average tangent plane (Π) in the middle of this edge (for example, by considering the tangent planes at its endpoints). An ideal point is defined in this plane at a distance that makes it possible to create an equilateral triangle, and then projected onto the surface, the corresponding triangle then being formed. The front is updated,
- repeat the previous stage, as long as the front is not empty.

Figure 15.3 illustrates the main steps of this method : i) characteristic points are identified, ii) the contour of the patch is constructed and then meshed, iii) a unique contour is defined and iv) an optimal triangle is formed.

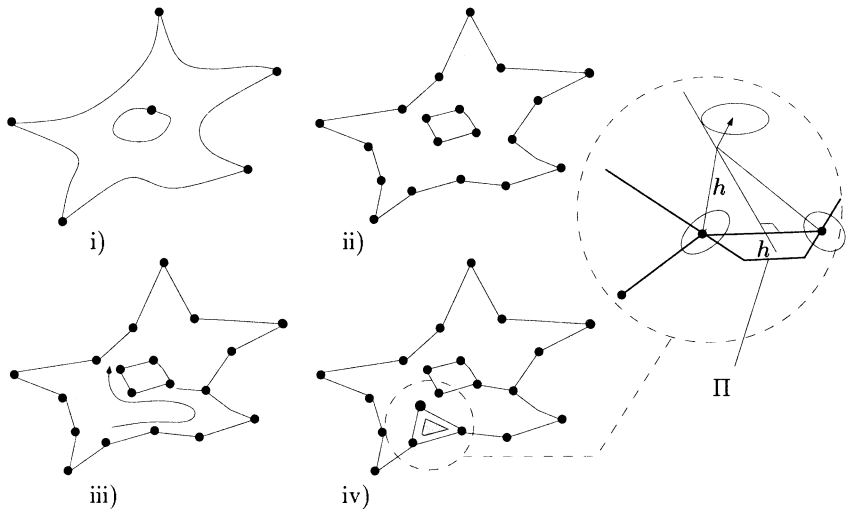


Figure 15.3: *Main steps of an advancing-front type strategy.*

The proposed scheme, while being conceptually simple, hides however certain difficulties. Such difficulties exist in two dimensions, up to a certain point, but they are relatively easy to overcome. An immediate example where such a problem arises is the case where the candidate point is located outside the domain. In two dimensions, this means that there must be an intersection between the edges incident to that point (forming a new triangle) and one or more edges of the current mesh. Hence, a simple intersection test allows us to detect and to avoid this configuration.

[5]This point, which is unnecessary in two dimensions, is a source of substantial simplification here.

It is more or less obvious that, for a surface, the same situation does not necessarily lead to an intersection. Therefore, the detection of this situation requires the use of more complex data structures (a neighborhood space, in three dimensions).

The case where the projection of the point considered (supposed optimal) on the true surface does not exist, meaning that this point is probably located straight down a hole, outside the surface or even leads to a topological ambiguity (for example, two neighboring surfaces exist and it is not possible to decide which one is the surface on which the point must lie).

In practice, provided the validity of the point creation process is ensured, as well as its valid connection with an edge, it is usual to conduct this creation stage iteratively. An initial optimal point is defined *a priori* and the corresponding solution is analyzed to possibly optimize the location. In other words, the process consists in using the additional information now available, about the point considered.

The topological and geometrical difficulties being supposedly resolved, notice that the proposed method must however incorporate a control on the accuracy of the approximation of the surface obtained. This control[6] can be performed by looking at the gap between a mesh edge and the surface (control of the interpolation error between a segment and a curve) and, more precisely, by controlling the gap between a mesh triangle and the surface (control of the interpolation error between a triangle and a surface, this control cannot be restricted to the sole control of the triangles edges). We propose here to reexamine the ideas developed in Chapter 10, which make it possible, using the second derivatives (the Hessian), to obtain such a control.

Remark 15.2 *The coupling between a direct approach and a parametric space offer additional information that makes it possible to overcome some of the difficulties mentioned previously [Frykestig-1994].*

Octree-type method. We have seen (Chapter 5) that an approach based on a spatial decomposition of the domain of interest makes it possible to generate a mesh of this domain. This type of approach also allows us to construct a mesh of the domain boundary if the discretization of the surface is not a data of the problem. We will now examine this last feature.

Given a bounding box of the domain, it is possible to generate directly a mesh of the surface, without using a parametric space (cf. [Grice *et al.* 1988] and [Shephard,Georges-1991]). The construction of the decomposition tree follows the general classical scheme already described, with only a few slight modifications (in

[6]And this will be true for the whole range of methods discussed in this chapter.

fact numerous variants exist). The basic idea consists in inserting in the tree the entities of the geometric model[7] in the increasing order of their dimensions. More precisely, the construction scheme includes the following stages :

- identifying corners and points of discontinuity of the model,
- inserting these points in the current tree, the stopping criterion for subdivision being unchanged (*i.e.,* at most one point per cell),
- then, for each terminal cell, using the local properties of the surface in order to decide on the possible refinement of the cell. Recall that the element size is related to the cell size, the latter also being related to the local curvature of the surface (which can be estimated at a finite number of points in the cell by sampling),
- and, when all cells have a size compatible with the given size map (*intrinsic* or specified), the critical point consists in determining the interactions between the tree cells and the geometric model, which can lead to refining the cells.

The analysis of the interactions between a cell and the geometric model involves calculating intersections. In fact, one has to determine exactly (*i.e.,* to the required accuracy) the intersections of corners, edges and faces of the octants with the surface [Kela-1989]. The intersection points are then used to create edges (connecting to such points) and loops (joining several edges and forming a closed contour) in each terminal cell. These entities will provide the vertices, edges and faces (possibly subdivided into triangles) of the final mesh. When the result of these tests is not known (if the intersection points are not returned by the modeler) or if this result is ambiguous, more costly operations may be used.

Before discussing the meshing stage itself, a balancing procedure is applied on the tree structure, using the [2:1] rule (limiting the difference between neighboring cells to 2 levels).

The creation of the mesh elements is performed as in the classical case. Here however, only the boundary cells are of interest. There is no *a priori* limit on the geometric complexity of the entities contained in a tree cell. In practice, several criteria allow a cell to be refined when the complexity becomes too great to handle. Notice that meshing the portion of surface enclosed within a cell is an operation which is approximatively of the same order of complexity as a classical automatic meshing technique. More details on creating mesh elements in boundary cells can be found in [Shephard *et al.* 1988b] and [Shephard,Georges-1991], among others.

The main difficulties associated with the octree approach are mainly related to :

- the creation of badly shaped elements (the intersections between the octants and the surface are not well handled),
- the topological ambiguity problems mentioned above that can compromise the meshing process of boundary octants.

[7] Instead of the entities of the boundary discretization.

Figure 15.4 shows an example of a surface mesh obtained using a direct octree-based method. Notice that the resulting mesh has not been optimized, the alignment of vertices clearly denotes the octants-surface interactions.

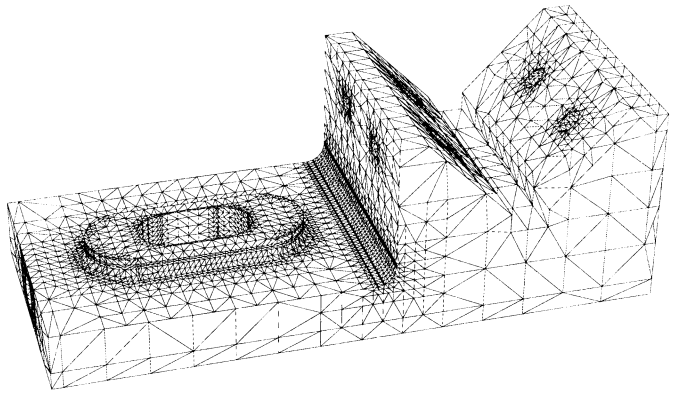


Figure 15.4: *Example of a surface mesh generated using a direct octree-based approach, without optimization (data courtesy of Mac Neal-Schwendler Corp.).*

Other methods. Among the other possible methods, we find those methods based on grid mapping and all methods based on iso-surfaces (implicit surfaces, *Marching-cubes* type algorithms, voxels, etc., Chapter 16) and the methods for which the input data is a cloud of points located on the surface (for example, [Hoppe *et al.* 1991], [Boissonnat *et al.* 1999]).

In this section, we only give some idea of the approaches based on the mapping of a predefined grid (which are indeed close to certain methods discussed in Chapter 8).

- Grid-mapping methods.

The principle underlying these methods is very simple. We define a grid (a regular mesh) in a plane located above the surface, in such a way that the projection of this grid, in a suitable direction $\vec{d}$, encloses entirely the surface. Supplied with this assumption, we analyze the projection of the grid cells on the surface in the direction $\vec{d}$. To this end, we examine (Figure 15.5) the projection of the grid nodes :

- if the direction line $\vec{d}$ coming from a node intersects the surface, the intersection point is kept as a mesh vertex,
- otherwise, we find mainly two configurations : either the projection goes through a hole (a loop of vertices exist in some neighborhood of the projected point that belongs to the previous class) or the projection falls outside the surface (an external side of the surface exists in some neighborhood).

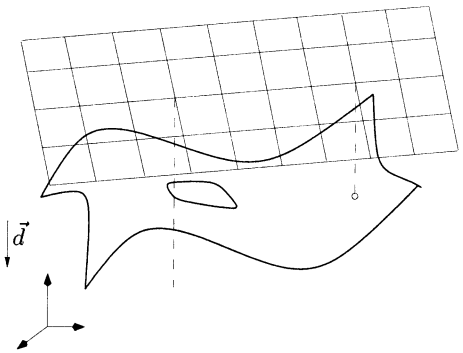


Figure 15.5: *Projection of a grid onto the surface. Two cases are visible, the projection is found (right-hand side), the projection does not exist (left-hand side) because of a hole. The third case, where the projection is outside the surface, is not described here.*

Notice that this approach makes the mesh independent of the patches describing the surface. However, it is obvious that the main drawback of the approach is related to projection problems (independently of the fact that the approximation of the surface by the mesh is not controlled and that, in addition, a uniform grid is always used).

In such an approach, several problems must be taken into account. Among the pertinent questions that arise, let us mention the following ones :

- How should the plane support of the grid (with respect to the surface) be chosen ?
- Is such a plane always defined (bijectivity of the projection) ?
- What are the potential perverse effects following a bad choice of the plane ?
- What is the (possible) dependency between this choice and the presumed result ?

Assuming these questions to be settled, several points need to be discussed.

The first obvious observation is that if the four nodes of a cell have a projection on the surface, then the resulting quadrilateral so formed is a candidate mesh element. On the other hand, any different situation requires a more subtle discussion as will be seen later.

If the four projections of a cell vertices exist, it is possible to form a quadrilateral. This element needs to be validated as a mesh element. This consists essentially in finding if this quadrilateral does not mask a hole (whose size is necessarily smaller than the grid stepsize) or does not overlap a "gap" between two portions of the surface (connected component problem or topology violation). In other words, we must verify the topological compatibility of the mesh, *i.e.*, preserve the surface topology.

The case where one or more vertices do not project onto the surface corresponds to the existence of a portion of boundary (internal, a hole, or external, an external side) in a neighborhood whose size can be determined by looking at the projections of the adjacent nodes. An analysis of these cases is needed in order to find the corresponding configuration (or, at least, a plausible configuration from the point of view of topological compatibility) :

- the existence of a projection defect for a point, although there exists a loop of its neighbors that projects without difficulty, presumes the existence of a hole whose size is related to this loop. Provided this configuration is identified, it is possible to find a mesh that also includes a hole (in fact, a potentially rough approximation of the true hole) that therefore maintains the topology of the surface. To this end, we first detect an approximate side of this hole and we modify the neighboring elements so as to account for;
- the same phenomenon without a suitable loop of neighbors representative of the existence of a surface border. As above, it is possible (although tedious) to find an approximation of this border preserving the initial topology.

These remarks indicate the complexity of the problems to be taken into account and suggest that, in order to be applicable, this type of method must be coupled with rather complex techniques (adaptation, *i.e.*, local refinement, quick search for intersection, verification of the topological consistency of the surface, etc.)

Hence, to conclude, this type of method is only really applicable for relatively simple surfaces (low curvature, no hole and no topological ambiguity).

15.2.4 Construction in two dimensions and mapping

Unlike the methods mentioned above, here the mesh generation method is applied in an adequate $\mathbb{R}^2$ domain, then, via a projection function, the mesh constructed in this domain is projected onto the $\mathbb{R}^3$ surface. Three types of method may be considered, the so-called "classical" methods (*quadtree*, advancing-front and Delaunay). In this context, the problem to be solved consists in governing the mesh generation in the one or the several parametric space(s) in order to guarantee the resulting mesh to be adequate after being projected onto the surface. In the following, we discuss more specifically the Delaunay type approach while observing that the other classical methods can also be used (at least, if the mesh in the parametric space is isotropic, a classical limit for the *quadtree* type methods).

15.3 A single patch

We focus first on a surface composed of a single patch. We will deal later with surfaces composed of several patches, for which two situations need to be considered, the *patch-dependent* case where a discretization of the patch boundaries is necessarily present in the final mesh and the *patch-independent* case where the inter-patch boundaries are not necessarily present in the resulting mesh.

Remark 15.3 *We will now detail some of the notions and methods that could be used to complete the direct approach described previously.*

15.3.1 Typical patches

Before dealing with the single patch case, let us recall the various types of patches used in C.A.D. systems.

The simplest patches and ones which are widely used are polygons with a small number of sides (thus, we usually find triangles and quadrilaterals). However, this type of patch does not offer the flexibility necessary and desirable to define certain surfaces. This is why more complex patches have been also developed. For example, those where only part of the patch is used (hole(s)) and those where the useful area does not correspond to the whole patch (the border limiting the useful area is not the "natural" side of the patch).

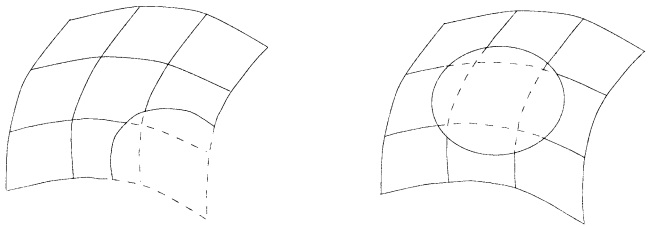


Figure 15.6: *Two examples of complex patches, left-hand side, only a part of the patch is used, right-hand side, a hole exists in the surface.*

For the sake of simplicity, we can, at first, leave to one side all these special patches and deal with complete patches. Then, it is relatively easy to see what may happen with the other types of patches.

15.3.2 Desired features in the parametric case

Let Σ denotes the surface, σ its parameterization and Ω its parametric space ($\Omega \subset \mathbb{R}^2$). Let us denote by $A, B, \ldots$ the points of Ω and by $P, Q, \ldots$ their images via σ. As stated previously, the surface mesh must satisfy certain properties (Relations (15.1) and following). Relation (15.1) simply means that :

$$A = (u, v) \in \Omega \quad \Longrightarrow \quad P = \sigma(A) = \sigma(u, v) \in \Sigma\,, \tag{15.5}$$

which is ensured if σ is the exact definition of Σ. Otherwise, the distance from P to Σ must be smaller than a suitable threshold ε (relative or absolute).

Relation (15.2) indicates that the edges are close to the surface :

$$AB \in \Omega \quad \Longrightarrow \quad PQ = \sigma(A)\sigma(B) \text{ is such that } d(PQ, \Sigma) \leq \varepsilon\,, \tag{15.6}$$

where ε is a given threshold of accuracy.

Remark 15.4 *Saying that $PQ = \sigma(A)\sigma(B)$ is only an approximation, indeed, $\sigma(AB)$ is* a priori *a curve traced on Σ and, in particular, $\sigma(AB) \neq \sigma(A)\sigma(B)$.*

Similarly, Relation (15.3) becomes :

$$ABC \in \Omega \Longrightarrow PQR = \sigma(A)\sigma(B)\sigma(C) \text{ is such that } d(PQR, \Sigma) \leq \varepsilon\,, \tag{15.7}$$

hence, the distance from triangle PQR, the image[8] of the triangle ABC, to the surface Σ is bounded.

Finally, Relation (15.4) must be verified, in particular to preserve, via its mesh, the possible regularity of the underlying surface.

The problem is then, if possible, to express these requirements in mesh requirements in the parametric space. To this end, we return to the basic definitions of parametric patches (Chapters 11 and 13), to the notion of metric (Chapter 10) and to that of control space (Chapter 1).

Briefly and before going into further detail, let us recall that the edge length in $\mathbb{R}^3$ can be expressed as a function of the corresponding edge AB in $\mathbb{R}^2$ and of the first fundamental form of Σ. Similarly, the control of the distance between an edge and Σ can be done by using the second fundamental form of the surface. On the other hand, the control of the distance from a triangle to Σ requires a more subtle analysis.

15.3.3 Meshing a patch

As already mentioned, any meshing technique in the plane is *a priori* a conceivable method. However, we will restrict ourselves to a Delaunay-type method (Chapter 7). We have seen in fact that this type of method could rather easily follow a given field of constraints (of specifications). The idea is to govern the mesh in the parametric space in such a way that the desired properties are (automatically) satisfied, after mapping on the surface.

Geometric aspects. Thus, we want to control the surface mesh by controlling the mesh of the patch in the parametric space. At first, notice that the control in the parametric space is essentially performed by controlling the lengths of the mesh edges.

We must then evaluate the length of a mesh edge. To find this value, we go back to the definition of the length of a curve, then we will improve this by considering that the curve is an edge.

The context is illustrated in Figure 15.7. The surface is denoted Σ, its parametric space (u, v) is Ω. Moreover, we consider an interval $[a, b]$ in $\mathbb{R}$. The curve PQ of the surface Σ is the image by a function γ of the interval $[a, b]$ in $\mathbb{R}$. This curve is also the image by the function σ of a curve AB of Ω and this curve AB is the image by a function ω of the interval $[a, b]$. We then have :

$$\sigma : \Omega \longrightarrow \mathbb{R}^3,\ (u, v) \longmapsto \sigma(u, v)\,,$$
$$\gamma : [a, b] \longrightarrow \mathbb{R}^3,\ t \longmapsto \gamma(t)\,,$$
$$\omega : [a, b] \longrightarrow \Omega,\ t \longmapsto \omega(t)\,,$$

[8] In fact, the triangle PQR is the triangle whose vertices are the images of the corresponding vertices in the parametric space (see the remark about the edges).

with the relation linking γ, ω and σ :

$$\gamma = \sigma \circ \omega .$$

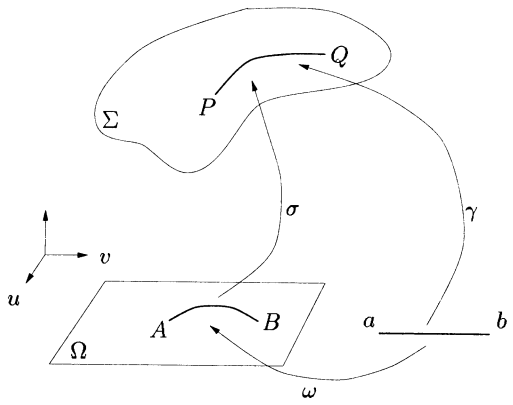


Figure 15.7: *The segment* $[a, b]$ *in* $\mathbb{R}$. *The curve* AB *of* Ω, *image of* $[a, b]$ *by* ω. *The curve* PQ *of* Σ, *image of* ab *by* γ *is also the image of* AB *by* σ.

The length of the curve PQ, Chapter 11, can be written as :

$$l(PQ) = \int_a^b \|\gamma'(t)\|\, dt = \int_a^b \sqrt{\langle \gamma'(t), \gamma'(t) \rangle}\, dt .$$

Using a vector notation, the dot product $\langle \gamma'(t), \gamma'(t) \rangle$, is expressed by :

$$\langle \gamma'(t), \gamma'(t) \rangle = {}^t\gamma'(t)\gamma'(t) = {}^t\omega'(t){}^t\sigma'(\omega(t))\sigma'(\omega(t))\omega'(t) ,$$

as $\gamma = \sigma \circ \omega$. That is, by denoting $\vec{\tau}_1 = \sigma'_u$ and $\vec{\tau}_2 = \sigma'_v$,

$$\langle \gamma'(t), \gamma'(t) \rangle = {}^t\omega'(t) \begin{pmatrix} \vec{\tau}_1 \\ \vec{\tau}_2 \end{pmatrix} (\vec{\tau}_1\, \vec{\tau}_2)\, \omega'(t) .$$

If we now assume that $\omega(t)$, for t between $a = 0$ and $b = 1$, is an edge, the edge AB, we have $\omega(t) = A + t\overrightarrow{AB}$ and $\omega'(t) = \overrightarrow{AB}$ and, hence, we find :

$$\langle \gamma'(t), \gamma'(t) \rangle = {}^t\overrightarrow{AB} \begin{pmatrix} \vec{\tau}_1 \\ \vec{\tau}_2 \end{pmatrix} (\vec{\tau}_1\, \vec{\tau}_2)\, \overrightarrow{AB}$$

and, for PQ, the length is :

$$l(PQ) = \int_0^1 \sqrt{{}^t\overrightarrow{AB} \begin{pmatrix} \vec{\tau}_1 \\ \vec{\tau}_2 \end{pmatrix} (\vec{\tau}_1\, \vec{\tau}_2)\, \overrightarrow{AB}}\, dt ,$$

which is also the classical formula :

$$l(PQ) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\mathcal{M}_1(A + t\overrightarrow{AB})\overrightarrow{AB}}\, dt .$$

where $\mathcal{M}_1$ is the matrix expression of the first fundamental form of the surface Σ. We have thus found the link between the length of an edge AB in the parametric space and the length of the curve of Σ, image by σ of this edge.

In our case, during the meshing stage, the image of an edge of Ω will be an edge of Σ. In fact, this is equivalent to approaching the *curve* PQ by the *edge* PQ, that is to have :

$$\sigma(AB) \approx \sigma(A)\sigma(B) ,$$

which can be also written differently by saying $\sigma(A + t\overrightarrow{AB}) \approx P + t\overrightarrow{PQ}$, hence, to say that :

$$\|\overrightarrow{PQ}\| \approx l(PQ) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\mathcal{M}_1(A + t\overrightarrow{AB})\overrightarrow{AB}}\, dt ,$$

where PQ is an edge and, thus, we have found a link between the length of an edge AB in the parametric space and the expected approximate value of the length of an edge PQ of Σ. This link allows us to control the mesh on Σ by controlling the mesh of Ω.

We now deal with the case where a metric is specified on the surface. Let $\mathcal{M}_3$ be the associated current matrix. This matrix is a 3×3 matrix (hence the index $_3$). The problem is then to find the relation between a length on Σ and the corresponding length in Ω, which is equivalent to exhibiting the matrix $\mathcal{M}_2$, a 2×2 matrix, relative to $\mathcal{M}_3$.

In principle, it is sufficient to follow the same reasoning as previously, simply changing the definition of the dot product (Chapter 10). We now have :

$$\langle \gamma'(t), \gamma'(t) \rangle = {}^t\gamma'(t)\mathcal{M}_3\gamma'(t) ,$$

or :

$$\langle \gamma'(t), \gamma'(t) \rangle = {}^t\omega'(t) \left\lfloor \begin{pmatrix} \vec{\tau}_1 \\ \vec{\tau}_2 \\ \vec{\nu} \end{pmatrix} \mathcal{M}_3\, (\vec{\tau}_1\, \vec{\tau}_2\, \vec{\nu}) \right\rfloor_2 \omega'(t) ,$$

with $\vec{\nu}$ the unit normal and the notation $\lfloor\ \rfloor_2$ that indicates that only the first two lines and two columns of the matrix are considered. By denoting Π the transition matrix from the canonical basis of $\mathbb{R}^3$ to the local basis at the current point $M = P + t\overrightarrow{PQ}$, this relation can be written :

$$< \gamma'(t), \gamma'(t) >= {}^t\omega'(t) \lfloor {}^t\Pi\, \mathcal{M}_3\, \Pi \rfloor_2\, \omega'(t) .$$

Posing :

$$\mathcal{M}_2 = \lfloor {}^t\Pi \mathcal{M}_3 \Pi \rfloor_2 ,$$

which is, as mentioned, the matrix formed by the first two lines and two columns of the matrix ${}^t\Pi\mathcal{M}_3\Pi$. Thus, we have found the matrix $\mathcal{M}_2$ corresponding to the given matrix $\mathcal{M}_3$. The length of the curve PQ for the metric $\mathcal{M}_3$ is then :

$$\|\overrightarrow{PQ}\| \approx l_{\mathcal{M}_3}(PQ) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\mathcal{M}_2(A+t\overrightarrow{AB})\overrightarrow{AB}}dt\,.$$

To construct a mesh, we approach the length of the edge PQ by the length of the curve PQ and we say that $PQ = P + t\overrightarrow{PQ}$, then, we want to have as above :

$$l_{\mathcal{M}_3}(PQ) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\mathcal{M}_2(A+t\overrightarrow{AB})\overrightarrow{AB}}dt\,.$$

Remark 15.5 *By taking $\mathcal{M}_3 = I_d$, we find $\mathcal{M}_2 = \lfloor {}^t\Pi I_d \Pi \rfloor_2 = \lfloor {}^t\Pi\Pi \rfloor_2 = \mathcal{M}_1$, the first fundamental form of Σ. We then find the classical case as a particular case of the general situation.*

This leads to the following remark :

Remark 15.6 *A Euclidean length of 1 in Σ corresponds to the choice $\mathcal{M}_2 = \mathcal{M}_1$ for Ω. In other words, the unit circle of Σ corresponds to the ellipse $\mathcal{M}_1$ of Ω.*

The control of the gap between an edge and the surface is performed by using a metric $\mathcal{M}_3$ that still needs to be defined. To govern the surface mesh, the mesh of Σ when a metric $\mathcal{M}_3$ is specified, we must then govern the mesh of Ω for a metric $\mathcal{M}_2$ (2×2 matrix) constructed as indicated above. This control will allow us, depending on the choice of $\mathcal{M}_3$, to the edge to satisfy such or such property. For example, as will be seen later, an adequate choice of $\mathcal{M}_3$ will mean that this edge will not be further from the surface than a given threshold value.

From the metric point of view, we consider $\mathcal{M}_3$ and we want to have ;

$$l(PQ) = 1 \quad \text{for} \quad \mathcal{M}_3\,,$$

that is :

$$1 = \int_0^1 \sqrt{{}^t\overrightarrow{PQ}\mathcal{M}_3(P+t\overrightarrow{PQ})\overrightarrow{PQ}}dt\,.$$

To this unit value for $\mathcal{M}_3$ on Σ corresponds a unit value for $\mathcal{M}_2$ in Ω :

$$1 = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\mathcal{M}_2(A+t\overrightarrow{AB})\overrightarrow{AB}}dt\,.$$

The aim is then to construct a mesh in Ω for which the edges are of unit length so as to ensure the same property for the corresponding edges on Σ.

Exercise 15.2 *Verify that :*

$$\mathcal{M}_3 = \begin{pmatrix} \frac{1}{h^2} & 0 & 0 \\ 0 & \frac{1}{h^2} & 0 \\ 0 & 0 & \frac{1}{h^2} \end{pmatrix}$$

implies that the edges on Σ are of uniform length of size h. On a simple example, to be defined, show the control matrix $\mathcal{M}_2$.

Choice of the control metrics. A matrix of the form :

$$\mathcal{M}_3 = \begin{pmatrix} \frac{1}{h^2} & 0 & 0 \\ 0 & \frac{1}{h^2} & 0 \\ 0 & 0 & \frac{1}{h^2} \end{pmatrix}. \tag{15.8}$$

specifies, as already seen, a field of uniform sizes h on the surface.

A matrix of the form :

$$\mathcal{M}_3(P) = \begin{pmatrix} \frac{1}{h^2(P)} & 0 & 0 \\ 0 & \frac{1}{h^2(P)} & 0 \\ 0 & 0 & \frac{1}{h^2(P)} \end{pmatrix}, \tag{15.9}$$

where now, h depends on the position, specifies a variable field of sizes on the surface. If we pose $h(P) = \alpha\rho(P)$ where $\rho(P)$ is the smallest of the radii of curvature ρ_1 and ρ_2 at point P of Σ and α is a adequate coefficient, we obtain an *isotropic* control of the sizes related to the radii of curvature (thus to the geometry of Σ).

A matrix of the form :

$$\mathcal{M}_3(P) = {}^t\mathcal{D}(P) \begin{pmatrix} \frac{1}{\alpha^2\rho_1^2(P)} & 0 & 0 \\ 0 & \frac{1}{\beta^2\rho_2^2(P)} & 0 \\ 0 & 0 & \lambda \end{pmatrix} \mathcal{D}(P) \tag{15.10}$$

with $\mathcal{D}(P)$ the principal directions at P, α and β adequate coefficients and λ an arbitrary scalar value gives an *anisotropic* control over the geometry that takes the two principal radii of curvature into account.

Also, a matrix of the general form :

$$\mathcal{M}_3(P) = {}^t\mathcal{R}(P) \begin{pmatrix} \frac{1}{h_1^2(P)} & 0 & 0 \\ 0 & \frac{1}{h_2^2(P)} & 0 \\ 0 & 0 & \frac{1}{h_3^2(P)} \end{pmatrix} \mathcal{R}(P) \tag{15.11}$$

with $\mathcal{R}(P)$ directions and h_i three sizes, gives a control on the three directions of $\mathcal{R}(P)$ and specifies the length expected in these specific directions.

Control obtained depending on the choice fixed : the edges. We have *a priori* four types of control matrices (Relations (15.8) to (15.11)).

Let consider a matrix of the form (15.9) and the choice $h(P) = \alpha\,\rho(P)$. The question is to fix α so as to obtain a certain control within a given ε of the gap between the mesh and the surface. Recall that ρ is the smallest of the two principal radii of curvature ρ_1 and ρ_2 at P.

By definition (Chapter 11) the circle of radius ρ centered at point $O = P + \rho\,\vec{\nu}$ is an approximation at the order two of the curve intersection of the surface with any plane supported by $\vec{\nu}$ with ρ the radius of curvature in this particular plane. Hence, approximating the surface by an edge while controlling the accuracy is equivalent to controlling, for this particular case, the gap between a discretization of the osculating circle (the circle above) of the plane and the latter.

We can see immediately that fixing $\alpha = 1$ is equivalent to discretizing this circle with 6 edges. The length of the circle for the metric (15.9) is indeed $2\,\pi \approx 6$. Then, Figure 15.8 (left-hand side), taking an edge of length ρ, we obtain $\delta = \rho\,(1 - \sqrt{\frac{3}{4}})$ and thus $\frac{\delta}{\rho} = 1 - \sqrt{\frac{3}{4}} \approx 0.15$. The relative gap to the circle is 15% .

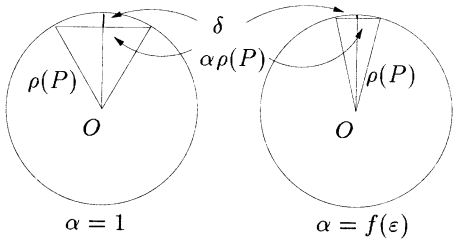


Figure 15.8: *Discretization of a circle with a stepsize $\rho(P)$ (left-hand side), or with a stepsize $\alpha\,\rho(P)$ (right-hand side).*

On the other hand, let us assume that the discretized edge is of size $\alpha\,\rho$, then we found :

$$\delta = \rho\,(1 - \sqrt{1 - \frac{\alpha^2}{4}})$$

and thus, setting $\frac{\delta}{\rho} < \varepsilon$ comes down to fixing :

$$\alpha \leq 2\,\sqrt{\varepsilon\,(2 - \varepsilon)} \tag{15.12}$$

and, hence, an accuracy within a given ε imposes that α is bounded as previously.

Remark 15.7 *We define thus a* relative *control, within ε, of the distance between an edge and the surface.*

Let us consider now the anisotropic case, the control matrix is that of Relation (15.10) in which the coefficients α and β are involved. For a fixed ε, the simplest choice consists in following Relation (15.12) for α and to setting $\beta = \alpha$.

For ρ_1 and the corresponding principal direction, we find a gap δ_1, for ρ_2 and its direction, we have δ_2. According to the choice of α and β, we have $\delta_1 = \varepsilon\,\rho_1$ while $\delta_2 = \varepsilon\,\rho_2$. Hence, if ρ_2 and ρ_1 are different, we have $\delta_2 > \delta_1$, the *absolute* gap is not the same in the two principal directions and thus this value varies according to the direction. Setting a constant gap comes down to setting $\delta_1 = \delta_2$ and leads to fixing :

$$\alpha \leq 2\,\sqrt{\varepsilon\,(2 - \varepsilon)}$$

as above and to defining :

$$\beta \leq 2\,\sqrt{\varepsilon\frac{\rho_1}{\rho_2}\left(2 - \varepsilon\frac{\rho_1}{\rho_2}\right)}. \tag{15.13}$$

Remark 15.8 *Thus we have a* local absolute *control, within ε, of the distance between an edge and the surface, depending on the directions.*

The case of other matrices (*i.e.*, non geometric *a priori*) prescribes sizes based on physical criteria (related to the behavior of the solution to the problem considered). There is no specific reason while this specification must be consistent with the surface geometry. Hence, we must deduce from the given metrics a metric that reflects the desired physical aspect as well as taking the geometrical constraints into account.

Control obtained : faces. The previous discussion gives hints about how to control the gap between the mesh edges and the surface. However, such a control gives no guarantee that the triangles created with these edges will be close to the surface or that the resulting mesh has the desired regularity (the presence of folds, for example). Hence, a specific process must be performed to ensure these properties.

The basic idea is to modify the field $\mathcal{M}_3$ or to involve mesh adaptation principles.

- Regarding the discussion in Chapter 10, we know that the gap between a triangle and the surface needs to account for a upper bound of the Hessian of the surface on this triangle. This bound allows us to find, within a given tolerance, a bound on the edge size. This bound then serves to define the matrix $\mathcal{M}_3$ and, more precisely, its coefficients.

- A rather different idea consists in using the mesh adaptation principle (Chapter 21). An initial mesh is constructed, analyzed and, depending on the result, the process is iterated with the field of the metrics deduced from the analysis.

A meshing method. To clarify the ideas, let us consider a Delaunay-type method (Chapter 7). Meshing a patch comes down to :

- choosing the type of control (the metric) allowing us to obtain the desired mesh (*i.e.*, to fix the field of matrices $\mathcal{M}_3$),
- deducing the field of corresponding matrices $\mathcal{M}_2$,
- meshing the parametric domain boundaries based on the field of $\mathcal{M}_2$ (Chapter 14),
- meshing the parametric domain, from the boundary discretization, using an anisotropic Delaunay-type method (Chapter 7),
- mapping this mesh on the surface Σ via the function σ.

The mesh in Ω following the approach described in Chapter 7 consists in constructing an initial mesh based on the sole boundary vertices and in adding points along the edges. Then, these points are inserted and the process is iterated on the resulting mesh. The key is thus to properly calculate the edge lengths. This calculation naturally requires the use of the surface, hence the meshing technique differs from a purely two dimensional meshing technique, in that it must have access to the surface (in a discrete way, for instance).

We now provide an example. Figures 15.9 to 15.12, [Borouchaki *et al.* 1999], show different meshes of the patch defined by :

- Ω, the parametric space is the circle of radius 10,
- the function σ defining the surface is

$$\begin{cases} u^3 + 10\,u \\ v^3 + 10\,v \\ 100 \sin u \cos v \end{cases}$$

The surface meshes in Figures 15.9 to 15.12 (right-hand side) have been created according to this principle. Figures 15.9 to 15.12 (left-hand side) show the four two-dimensional meshes of the corresponding parametric spaces.

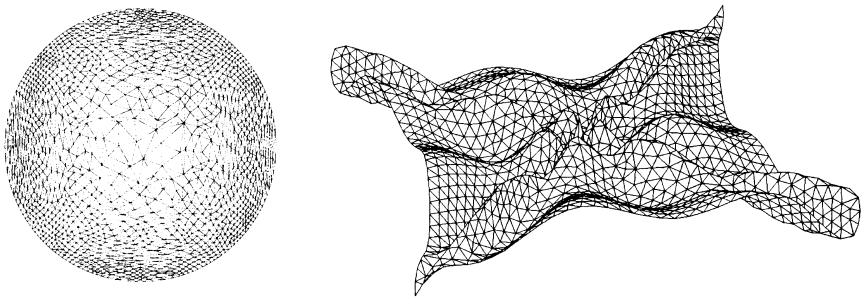


Figure 15.9: *Uniform mesh on Σ. Left-hand side : mesh of the parametric space, right-hand side : uniform mesh with a stepsize $h = 50$.*

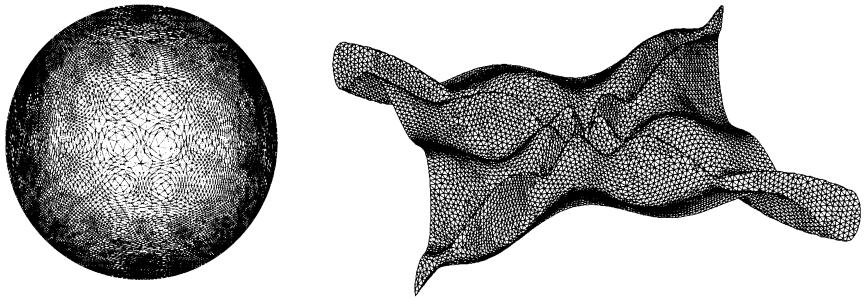


Figure 15.10: *Uniform mesh on Σ. Left-hand side : mesh of the parametric space, right-hand side : uniform mesh with a stepsize $h = 20$.*

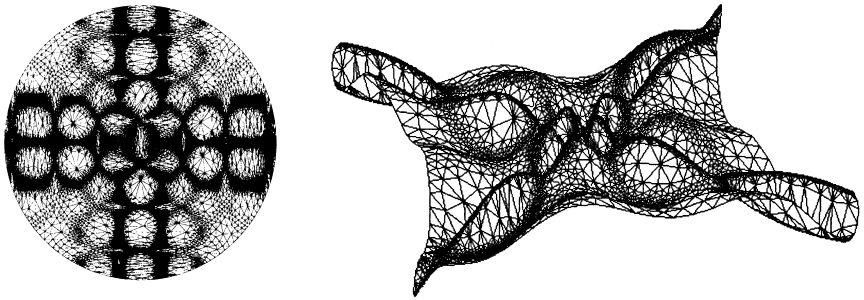


Figure 15.11: *Left-hand side : mesh of the parametric space, right-hand side : isotropic mesh controlled by ρ.*

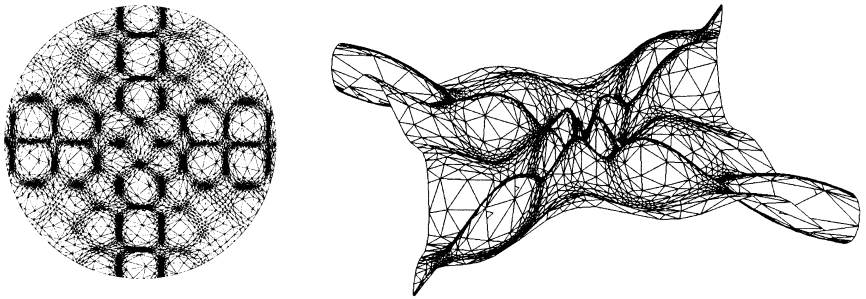


Figure 15.12: *Left-hand side : mesh of the parametric space, right-hand side : anisotropic mesh controlled by ρ_1 and ρ_2.*

15.4 Multi-patches surface (patch dependent)

When the surface is defined by several patches (each of them being parameterized), the previous algorithm (or a similar one) can be used, provided that the interfaces between the different patches are meshed first, in a unique way, so as to obtain a conforming join from one patch to another.

The patch-dependent approach consists in processing the patches one by one and thus in preserving the interfaces between patches.

Synthetic scheme. The meshing scheme uses the idea of the multibloc methods (Chapter 4). It involves two successive stages :

- meshing the interface curves between patches,
- meshing each patch using the discretization of its boundaries as defined in the previous stage.

15.4.1 Meshing the interfaces between patches

The first stage of meshing a composite surface consists in meshing the interface curves. This operation makes it possible to guarantee that the curves shared by several surfaces are meshed in a consistent (conforming) way.

To this end, we apply the technique for meshing the curves of $\mathbb{R}^3$ already described in Chapter 14 and reviewed at the beginning of this chapter. We will not spend more time on this issue here.

At completion of this stage, the interface curves between patches are meshed in a geometric way or to take a specified field of sizes into account.

15.4.2 Meshing the patches

Using the discretized patch boundaries, we will construct the meshes of patches. The union of all meshes of the patches will lead to a correct mesh of the surface, the interfaces having been correctly defined during the previous stage.

The principle of the meshing technique for a multi-patches surface consists in meshing each patch separately via its associated parametric space. The final mesh is the union of the meshes of the different patches. The conformity of the resulting mesh is ensured by the conformity of the interfaces between patches (see above).

Notice that such an approach may lead to some rather large disparities between the element sizes from one patch to another. This is notably the case when the mesh is a geometric mesh, based on the model curvatures (Figure 15.13, left-hand side).

To obtain a mesh in which the size gradation is controlled, the idea is to use the resulting mesh to construct a control space. More precisely, with each mesh vertex is associated information on the local size and, possibly, the stretching direction of the elements. This discrete field of sizes (metrics) being created, we can apply a smoothing procedure on it to bound the size variations from one point to another (Chapter 10). Once this operation has been performed, it is then possible to use

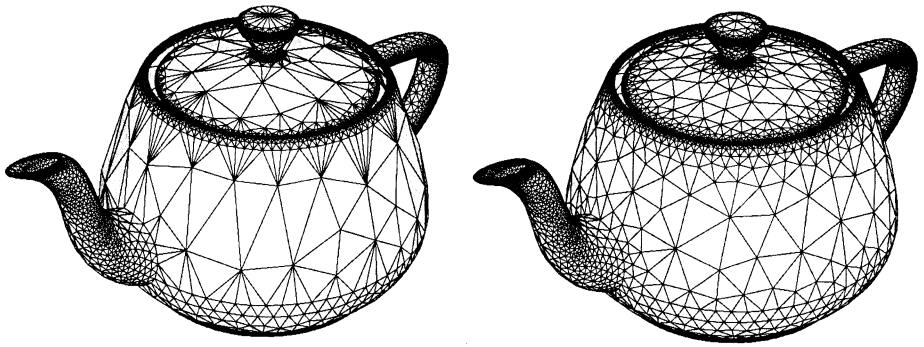


Figure 15.13: *Isotropic geometric mesh of the minimal radius of curvature of a multi-patches surface, the Utah teapot (left-hand side) and isotropic geometric mesh incorporating a size map correction (right-hand side).*

this information to govern the creation of a new surface mesh in the parametric spaces (Figure 15.13, right-hand side).

We will now deal with the processing of surfaces to construct patch-independent meshes.

15.5 Multi-patches surface (patch-independent)

For various reasons, we now want the surface mesh not to follow the patches in some regions. Consider first very small patches, which are probably not useful from the numerical point of view and which, if meshed, would lead to very large meshes. Consider also very thin patches having a small edge as compared with the neighboring ones. This case necessarily induces the creation of very stretched elements.

Figure 15.14 shows an example of such situations. In this figure, we show the entire geometry (i) and the enlargement of an area composed of stretched patches (ii). Any mesh respecting these patches (for example, the mesh in iii) contains flat elements that are *a priori* inappropriate to the calculations.

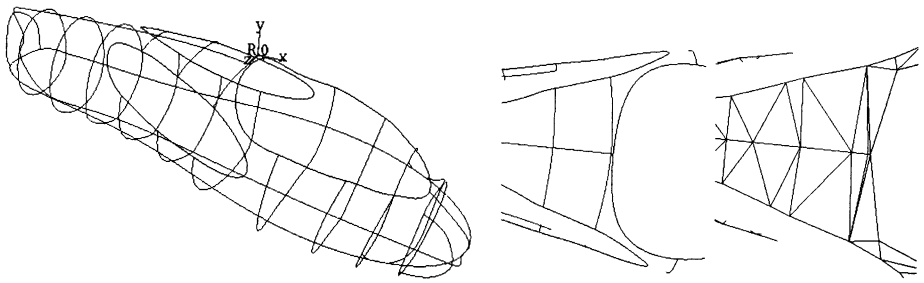
Thus, it seems necessary to get rid of the boundaries of some patches. To this end, two approaches can be envisaged. We can follow an *indirect* approach that consists in constructing a mesh respecting the patch boundaries and then in modifying this mesh to get rid of this constraint in the regions where it leads to undesirable effects. We can also consider a *direct* approach that, prior to any meshing, get rid of this type of situation.

15.5.1 Indirect approach

Each patch is processed via the previously described method (patch-dependent) then, we get rid of (some) patch boundaries. The operators involved in this process are the classical operators for surface mesh modification and surface mesh optimization (as described more precisely later in Chapter 19). We distinguish essentially a node relocation operator (that preserves the connections between the vertices) and operators that, with fixed point positions, modify their connections.

Among these operators, we find the edge swaps (that change the common edge between two faces) and refinement or derefinement operators that make it possible to :

- subdivide a given edge and to modify the faces sharing the edge accordingly,
- suppress one or several edges either by merging nodes or by temporarily creating a hole (Figure 15.15), and then remeshing it.



i) ii) iii)

Figure 15.14: *An example of mesh construction preserving the patches defining the geometry and the perverse effects of this constraint on the resulting mesh. In i), we show the limits of several patches modeling the geometry, in ii) and in iii), we show respectively the detail of a region of this description and the resulting mesh in which two flat triangles are constructed.*

Whichever the operator is used, one has to preserve the geometric and topological coherence of the result. Firstly, the result must stay close to the surface. Then, the resulting topology must be identical to the initial one. Notice that it is important to classify the entities of the mesh (points, edges and faces) and to propagate this information from the current mesh and the modified mesh (so as, in particular, to be capable later of identifying the entities subjected to boundary conditions for the problem considered).

15.5.2 Direct approach

A direct approach is based on the fact that meshing techniques are available to process a patch. The idea is then to represent a set of patches by a single patch. Hence, the difficulty is to construct this single patch. This is relatively easy to

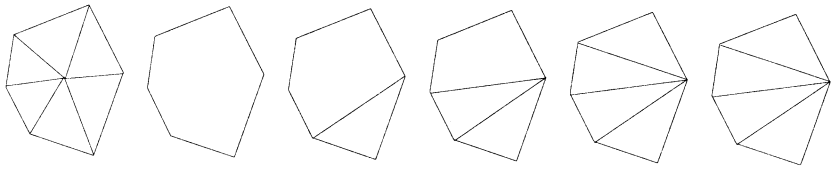


Figure 15.15: *Left-hand side, the initial mesh. From left to right, the sequence of remeshing of a hole formed by removing the ball of the point deleted.*

perform when the geometry is defined by a polyhedron (*i.e.*, when the set of patches of this definition forms such a polyhedron). Moreover, this method can be extended to an arbitrary surface. Figure 15.16 illustrates an application of this method[9]. We see in figure, i), the initial surface, ii), a planar mapping of the complete surface where each triangle corresponds in a one to one way to a piece of the surface. Therefore, an injective function σ can be constructed depending on the parameters u and v to project any point of the parametric space onto the surface. This being done, an anisotropic mesh of the parametric space must be created, its mapping onto the surface leading to the desired mesh. The example of the mesh in the figure is simpler, a regular (uniform) grid over the parametric space is developed, in iii) and, in iv), we show the mechanical mapping of this regular mesh on the surface.

The main ideas of the method are the following :

- identification of the boundaries (principal and secondary) and of the sides of the holes,
- subdivision into triangles of the parametric space associated with each patch,
- topological merging of the triangles,
- identification of the edges of the main boundary and of the secondary boundaries and sides of the holes,
- topological meshing of the secondary contours and holes (edges and faces are associated with these contours),
- node distribution for the main contour along the boundary of a convex domain,
- determination of the positions of the other nodes using a barycentrage technique from their neighbors. The resulting mesh allows us to define the parametric space associated to the whole surface. A bijection exists between each triangle of this two-dimensional mesh and a triangular-shaped portion of the surface.

The convex planar domain (corresponding to the parametric space) is meshed so as to take the constraints into account, for example regarding the boundaries

[9] Work in progress at the 3S laboratory, Grenoble, by F. Noël.

of the secondary contours or the holes. A governed meshing technique can thus be used (for example an anisotropic Delaunay-type method) to mesh the parametric space (see above).

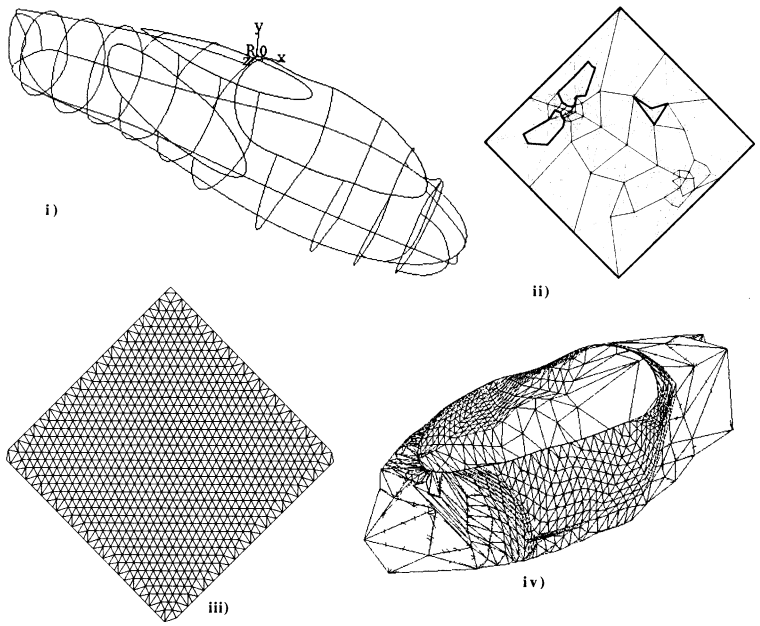


Figure 15.16: *An example of mesh construction based on a single patch definition. We see, i), the definition with patches of the entire surface. In ii), we show the planar mapping of these patches. In iii), we define a regular grid in the parametric space and, in iv), we provide the mapping of this regular mesh onto the surface.*

The resulting mesh in the parametric space is then mapped onto the surface using the one to one function linking the mesh triangles to the corresponding triangular cells on the surface.

Notice that this technique of planar mapping needs still to be improved and that the example provided reflects only preliminary experiments on this very promising topic.

15.6 Discrete surface (re-meshing process)

We consider here a situation in which the data of the problem is of a discrete nature. Actually, we have a meshed surface (a triangulation) with which additional information may be associated. These specifications, when given, allow us to know the particularities and specificities of the underlying geometric model (corners, ridges and/or constraints, tangents, etc.). Without such information, the task of the remeshing process will be more tedious, [Löhner-1995], [Löhner-1996a].

We will first specify how the underlying surface can be defined from the given triangulation. Then, we will focus on the remeshing problem, treated in practice via geometric and topological modifications applied to the initial mesh. We will indicate in particular the main features of a remeshing algorithm.

15.6.1 Definition of a discrete surface

We will see in Chapter 19 that the intrinsic properties of a surface (normals, principal radii of curvature, etc.) can be extracted[10] from an initial mesh of the surface, so as to construct the metric of the tangent plane. Recall that this metric (defined in the tangent planes associated with the given mesh vertices) makes it possible to control (bound) the gap between the mesh edges and the underlying surface.

Identification of the singularities. If the singularities of the model are not explicitly specified, a pre-processing stage makes it possible to identify these features. This stage, which is almost fully automatic, must be user-supervised. We have already mentioned that a ridge can be identified using a threshold on the adjacent face angle and that a corner is, notably, a vertex sharing three ridges.

Other entities (constraints) can be provided (by the modeling system) which must be preserved in the final mesh.

Construction of a geometric support. This operation is central to the remeshing approach. It is indeed aimed at defining, internally, a geometry (that is an analytical representation of the underlying surface, the initial mesh being supposed to be an approximation of this surface). In practice, the geometric support is a composed surface of class G^1 (*i.e.*, ensuring the identity of the tangent planes between adjacent patches).

Among the various approaches possible, one of particular interest is that suggested by Walton and Meek [Walton,Meek-1996], based on the Gregory patches [Gregory-1974] (Chapter 13).

This support, suitably defined, can be used to know the exact position of a vertex on a surface (usually the closest location), given a current mesh vertex and a direction.

15.6.2 Re-meshing a discrete surface

The remeshing of a surface defined by a polyhedral approximation (a mesh) involves topological modifications (edge swapping, vertex merging, edge splitting, etc.) and geometric modifications (node relocation) which will be considered in detail in Chapter 19.

[10] That is evaluated approximatively.

Optimal mesh. The objective of the remeshing process if to get a mesh in which the elements have a size (and possibly an orientation) conforming to the geometric size field. In other words, the aim is to obtain an *optimal mesh* with respect to the given geometric specifications.

To this end, the element edges of the current mesh are analyzed and possibly optimized in size, so that any edge AB of the final mesh has a length l_{AB} (in the metric specified) such that :

$$\frac{1}{\sqrt{2}} \leq l_{AB} \leq \sqrt{2}, \quad \forall AB \in \mathcal{T}. \tag{15.14}$$

Remeshing algorithm. The general scheme of the surface remeshing algorithm can be written as follows :

- initializations : identification of the singularities (corners, ridges, etc.);
- evaluation of the intrinsic properties of the surface (curvatures, normals, etc.);
- construction of a geometric support of class G^1;
- remeshing :
  - for each edge AB of the current mesh,
  - if $l_{AB} > \sqrt{2}$, subdivide the edge into unit length segments,
  - else if $l_{AB} < \frac{\sqrt{2}}{2}$, merge the two endpoints of the edges,
  - if the mesh has been modified, apply edge swappings on the newly created faces;
- optimization of the resulting mesh using node relocation as well as edge swapping.

The Figure 15.17 illustrates an example of a discrete surface remeshing. The surface is defined from the mesh represented in Figure 15.4.

15.7 Ill-defined multi-patches surface

As pointed out, complex surfaces can be defined with a rather important number of patches. In what precedes, we have implicitly assumed that the set of patches defining the surface considered formed, by itself, a conforming mesh (actually, rather a covering-up) of this surface (Chapter 1). In practice, this assumption is most likely not verified.

Actually we find several situations preventing the desired conformity. This is related to the fact that the geometric definitions of the patches are not, *a priori*, directly motivated by this criterion but are rather of a visual nature or related to a possible manufacturing process. Moreover, the numerical accuracy used in a C.A.D. system may be variable from place to place in a given model and, is definitely not conceived in a meshing view.

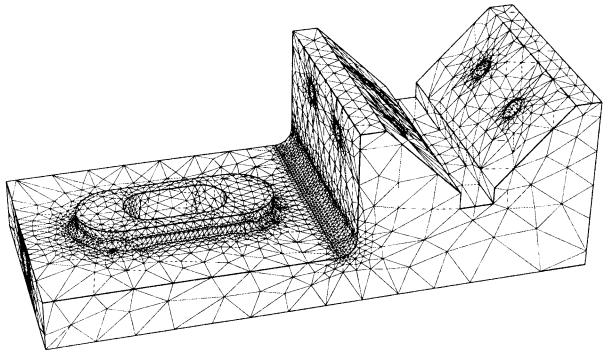


Figure 15.17: *Geometric remeshing of a discrete surface.*

In practice, the awkward situations encountered correspond to :

- overlappings between two or more patches,
- gaps (holes) between patches assumed to be joined,
- non conforming joins between patches regarding corners assumed to be but not identical (within a tolerance) and supposedly common borders. For example, the boundary of a patch matches only part of the boundary of another patch instead of the entire patch, etc.

All these defects, detected from a mesh generator but not visible in general by the user and with no effect in a manufacturing process, essentially concern defects related to points (absence or duplication of points), lines (bad join between successive or common lines, absence or duplication) or surfaces (holes, partial duplication). Moreover, as already mentioned, disparate edge sizes (patch boundaries) lead to ill-suited patches with respect to a possible mesh. Attempting to mesh, for example patch by patch, a surface composed of patches having these pathologies would certainly lead to a failure[11].

Thus, before attempting to mesh, it is necessary to fix, with respect to the objective, the surface components. This task is especially important, as pointed out, but is rather tedious to carry out. To date, there is no fully automatic, quick and of universal range method allowing such a surface to be *repaired*. Some mostly interactive (*i.e.*, that require the user intervention) software products exist, that may ease this type of process. The idea is to offer a maximum of features regarding the visualization and the detection of the defects as well as to provide all types of local modification tools (node relocation, edge swapping, creation or deletion of entities, etc.) in order to correct the surface while ensuring that the

[11] And, notice that it is the best desirable result. In fact, if the mesh generation algorithm fails to detect these defects, there may be no other easy way of detecting that the resulting mesh is wrong. Hence, any calculation performed on such a mesh would give surprising results.