Chapter 18

Mesh optimization

Introduction

Optimizing a mesh with respect to a given criterion is an operation that is frequently used with various goals in mind for a wide range of applications. First, optimization in itself is useful because the quality (the convergence of the computational schemes, the accuracy of the results) of the numerical solutions computed at the mesh nodes clearly depends on the quality of the mesh. In this respect, mesh generation methods usually include an optimization stage that takes place at the end of the entire mesh generation process. An optimization process may serve some more specific purposes such as the mesh adaptation, for instance, included in an adaptive computational procedure. Moreover, the tools involved in optimization methods can be also used in some particular applications (mesh simplification being a significant example).

★
★ ★

The aim of this chapter is to introduce some methods designed for mesh optimization purposes. First, some information is given on how to compute element surface areas and volumes. Applications based on surface area and volume values are discussed, including localization and intersection problems. Then we turn to the definition of mesh quality. Afterwards, we introduce some local tools for mesh optimization.

Having introduced these tools, and with regard to the given objectives, mesh optimization methods are discussed both in terms of strategies and computational aspects. Actually, mesh optimization can be considered as a step of a mesh generation method (in general, the last step of the method). It also can be seen as a stand-alone process.

This chapter only considers planar and volumic meshes. Moreover only the geometrical aspect is considered meaning that P^1 meshes (and more generally meshes whose element nodes are identical to element vertices) are discussed. Surface meshes, which are slightly different, will be discussed in the next chapter.

18.1 About element measurement

In this section, we discuss how to compute the surface areas or volumes of the different types of mesh elements. Then we give some indications about how such values can be used for various purposes. Note that the elements we are interested in are defined with an orientation (see Chapter 1).

18.1.1 Element surface area

The only element for which the surface area can be obtained directly is the triangle (due to its simplicial aspect). Thus, let K be a triangle whose vertex coordinates are denoted by x_i, y_i, $(i = 1, 3)$, then the surface area of K is :

$$S_K = \frac{1}{2} \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix}, \tag{18.1}$$

or, in an equivalent form :

$$S_K = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}. \tag{18.2}$$

In the case of a quad, the surface area can be computed as the sum of the surface areas of the two triangles formed by considering one of its diagonals (either diagonal being suitable, see Figure 18.1).

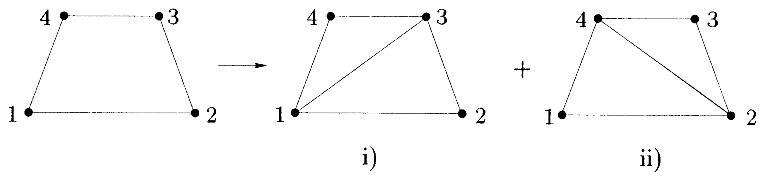

Figure 18.1: *Analysis of a quad by means of the four corresponding triangles. Two triangles allow for the surface area calculation while four triangles are necessary to check the convexity of the quad (see below).*

18.1.2 Element volume

In this case, only the volume of a tetrahedron is easy to compute. Let K be a tet whose vertex coordinates are denoted by x_i, y_i, z_i, $(i = 1, 4)$, then the volume of K is :

$$V_K = \frac{1}{6} \begin{vmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{vmatrix}, \tag{18.3}$$

or, similarly :

$$V_K = \frac{1}{6} \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}. \tag{18.4}$$

To compute the volume of a pentahedron, it must be subdivided into three tets (Figure 18.2).

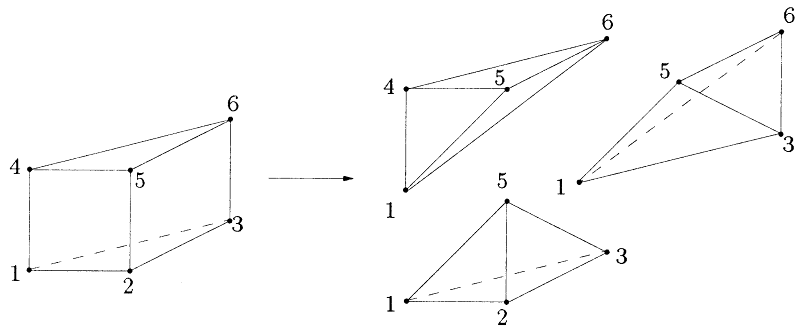

Figure 18.2: *Analysis of a pentahedron by means of three tets. Note that only one of the various possible partitions is given.*

For a hexahedral element, six or five tets are necessary. The partition based on six tets involves first splitting the hex into two pentahedral elements to which the above partition into three tets is applied (Figure 18.3).

A pattern with five tets is also possible (Figure 18.4) where the tet inside the volume is considerably different to the four others (unlike the pattern with 6 elements) for a regular initial hex.

18.1.3 Other measurements

Surface areas or volumes can provide other information which is useful in this context.

Convexity of a quad. A quad element in two dimensions can be analyzed to decide whether it is convex by using four triangle surface areas. Actually we define the four triangles that can be constructed using one or the other diagonal of the quad (Figure 18.1). Then we compute the four surface areas, S_i, of these triangles. Hence, if the four S_i's are positive, the quad is convex. Otherwise, if one of the S_i's is negative, the quad is non-convex. If one of the S_i's is null, the quad is said to be degenerated (*i.e.*, two consecutive sides are aligned). If two of the S_i's are negative, each being one part of the two possible decompositions, then the quad is self-intersecting while if two negative S_i's correspond to the same decomposition, the quad is negative (Figure 18.5).

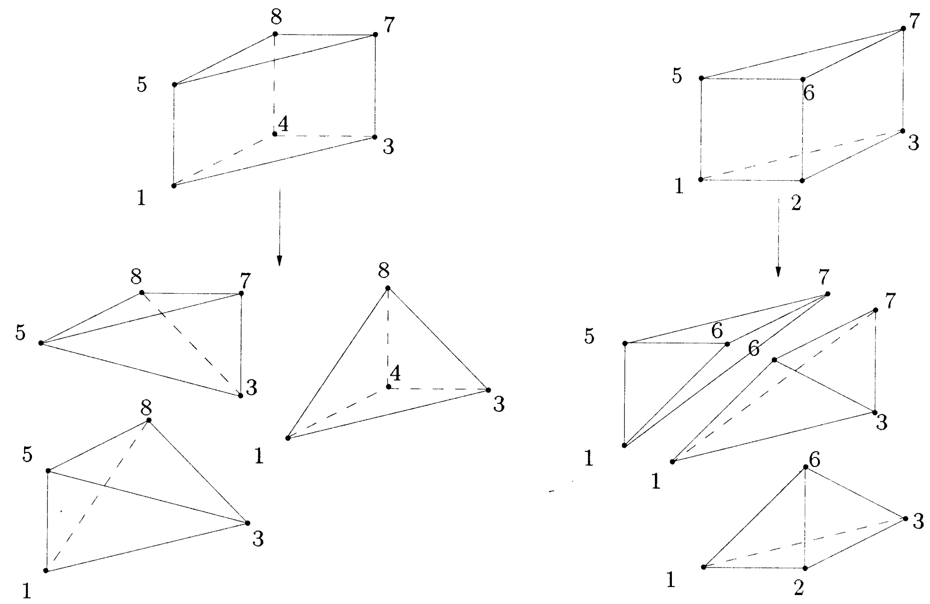

Figure 18.3: *Analysis of a hex by means of six tets. The initial hex* 12345678 *is primarily split into two pentahedra (*123567 *and* 134578*). In this example, the resulting partition is not a conforming mesh of the initial hex. Thus, such a partition is suitable for volume computation while a different one must be considered for a different purpose (for instance, if one wants to convert a hex mesh into a tet mesh).*

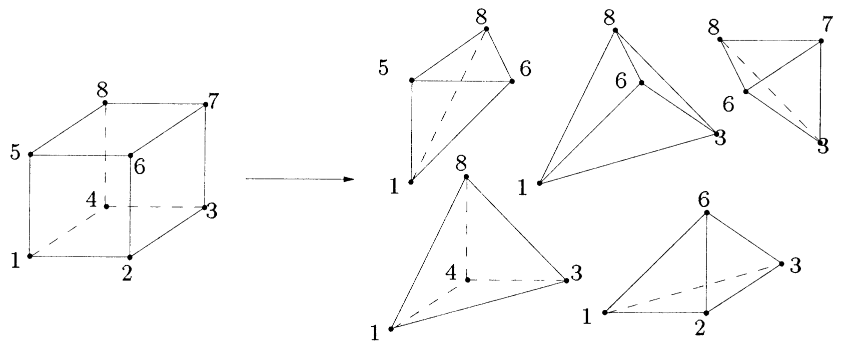

Figure 18.4: *Analysis of a hex by means of five tets.*

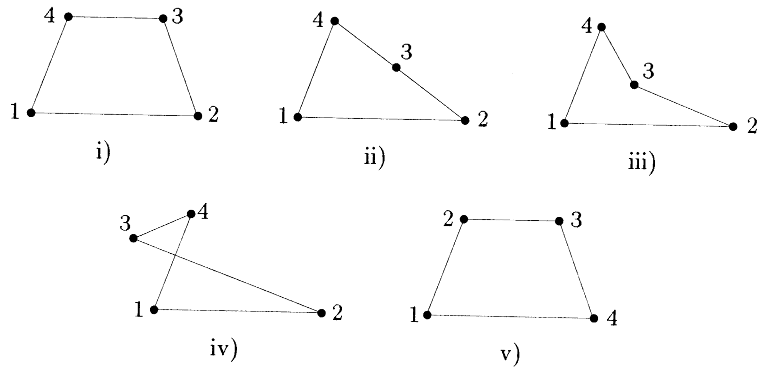

Figure 18.5: *The various configurations for a quad. A positive quad i), a degenerated quad ii), a non convex quad iii), a self-intersecting quad iv) and a negative quad v).*

Localization processes. The localization of a given point in a mesh is a frequent issue that arises in various situations. In Chapter 7, we saw that it enables us to find, in a current mesh, the element within which a given point falls, for instance, the point we want to insert in this mesh. Also in Chapter 2, this localization problem was mentioned in the examples on data structures and basis algorithms.

Here, we consider this problem again while observing that the localization of a given point in a given mesh can be completed with the help of surface area or volume evaluations. For the sake of simplicity we first restrict ourselves to simplicial meshes and we consider a convex domain.

Let P be a point and $\mathcal{T}$ be a simplicial mesh whose elements are denoted by K. The method relies on the observation that, given an element K in $\mathcal{T}$, we can compute the $d+1$ (d being the spatial dimension) surface areas (volumes) of the virtual-simplices defined by joining P to the $d+1$ edges (faces) of K, and if the $d+1$ quantities are positive then P falls in K, otherwise, if there exist one or several negative (null) quantities, then P can be classified as a member of such or such precise half-plane (half-space).

As a consequence, let K_0 be an arbitrary element in $\mathcal{T}$, we compute the $d+1$ surface areas (volumes) associated with K and we decide to pass to the neighboring element of K_0 through the edge (face) related to the negative quantity (we assume that the neighboring relationships from element to element are known). Then the so-defined element is used as the *basis* of the localization process and we repeat the same procedure.

One should note that using a control space (see Chapter 1) enables us to pick as the element K_0 an element not too far from P, resulting in a low cost algorithm.

Remark 18.1 *For meshes other than simplicial ones, the same method applies while, in some cases, replacing the elements by simplices. Thus, the problem could be significantly more expensive in terms of CPU time.*

Remark 18.2 *Non convex domains lead to a tedious solution of the localization problem. Indeed, it could be necessary to pass through a boundary in order to reach the solution. Thus, the latter problem must be addressed which is not so easy.*

Intersection processes. Similarly, surface area or volume computations may serve to solve some intersection problems. Indeed, we define some adequate virtual-elements whose surface area or volume signs allow the decision.

18.2 Mesh quality (classical case)

Relatively easy to define for simplicial meshes, the notion of quality must be carefully addressed for other types of meshes. In every case, several criteria may be used to evaluate mesh quality.

18.2.1 Shape or aspect ratio

This notion is basically associated with simplicial elements (triangles[1] or tetrahedra). The aspect ratio of a given simplex K is defined by :

$$\mathcal{Q}_K = \alpha_2 \frac{h_{max}}{\rho_K}, \qquad (18.5)$$

where h_{max} is the element *diameter*, *i.e.*, its longest edge while ρ_K is the inradius of element K. Notice that this value varies between 1 to ∞ and that moreover the closer $\mathcal{Q}_K$ is to 1, the better triangle K is. Indeed, $\mathcal{Q}_K$ measures the so-called *degradation* of element K. In two dimensions, we have :

$$\mathcal{Q}_K = \alpha_2 \frac{h_{max} p_K}{S_K} \qquad (18.6)$$

where p_K is the half-perimeter of K and S_K is the surface area of K. Similarly, in three dimensions, we have :

$$\mathcal{Q}_K = \alpha_3 \frac{h_{max} S_K}{V_K} \qquad (18.7)$$

where, now, S_K is the sum of the face surface areas and V_K is the volume of K. In these expressions, α is a normalization factor which results in a value one for the equilateral (regular, in three dimensions) simplex.

Exercise 18.1 *Find the value of the normalization coefficients α of Relationships (18.6) and (18.7).*

Remark 18.3 *In terms of computation, it is advisable to compute the inverse of the above relationships. In this way a null surface (volume) element does not lead to a numerical problem.*

In Figure 18.6, we give a graphic impression of the way in which the quality of a triangle, Relation (18.6), varies as a function of the location of the three vertices.

[1] See the next chapter for surface triangles.

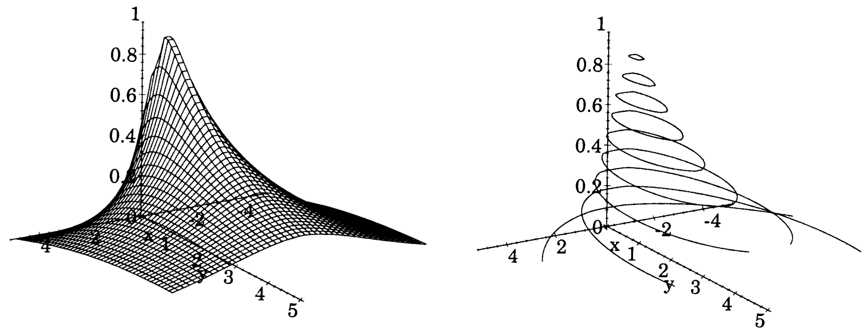

Figure 18.6: *Variation of the aspect ratio as a function of vertex C of triangle ABC, given AB. Edge AB spans the interval $[-0.5, 0.5]$ along the x-axis (rear) and C is in the half-plane bounded by this axis. Left: quality function. Right: iso-contour values of the quality function. For the sake of clarity, this figure displays the inverse of the quality defined in Relationships (18.5) and (18.6) (Note that the maximum value 1 is obtained when C is at location $(0, \frac{\sqrt{3}}{2})$).*

18.2.2 Other criteria

Numerous quality functions can be used as an alternative way to determine the quality of a given simplex. The simplest one is :

$$\mathcal{Q}_K = \beta_2 \frac{h_s^2}{S_K} \qquad (18.8)$$

in two dimensions, with β a normalization factor and $h_s = \sqrt{\sum_{i=1}^{3} L_i^2}$ where L_i is the length of edge i of triangle K. Similarly, in three dimensions, we have :

$$\mathcal{Q}_K = \beta_3 \frac{h_s^3}{V_K} \qquad (18.9)$$

where $h_s = \sqrt{\sum_{i=1}^{6} L_i^2}$, L_i also being the length of edge i of tetrahedron K.

Exercise 18.2 *Find the value of the normalization coefficients β for the above cases.*

Remark 18.4 *See the previous remark regarding the case where null elements may exist.*

Apart from the two above functions to appreciate the quality of a simplex, we encounter numerous other ways. For more details about this, one may consult, among others, [Cavendish *et al.* 1985], [Baker-1989b] or [deCougny *et al.* 1990] or again [Dannelongue,Tanguy-1991] and a synthesis by [Parthasarathy *et al.* 1993].

Before enumerating some of the possible quality measures (in three dimensions), we introduce a few notations. For a given element K, with volume V_K, the inradius is denoted by ρ_K, the circumradius is r_K. The length of edge i of K is L_i, the surface area of face i is S_i. We now introduce $S_K = \sum S_i$, the sum of the surface areas of the faces of K, h_{max} the diameter of K, *i.e.*, $h_{max} = \max_i L_i$, and $h_{min} = \min_i L_i$ and, at lastly, L_{mean} the average of the L_i's. Then, the quality measures are as follows :

- $\frac{r_K}{\rho_K}$, the ratio between the radii of the two relevant spheres,
- $\frac{r_K}{h_{max}}$ (or $\frac{r_K}{h_{min}}$), the ratio between the circumradius and the element diameter (the shortest edge),
- $\frac{h_{max}}{h_{min}}$ or $\frac{h_{min}}{h_{max}}$, the ratio between the edges with extremal lengths,
- $\frac{V_K^4}{(\sum_{i=1}^{4} S_i^2)^3}$, the ratio between the volume and the surface areas of the faces,
- $\frac{h_{mean}^3}{V_K}$, the ratio between the average edge length and the volume,
- δ_{max}, the maximal dihedral angle between two faces,
- θ_{min}, the minimal solid angle associated with the vertices of K.

with, for the two last measures, the following definitions :

Definition 18.1 *The dihedral angle between two faces is the value*

$$\pi \pm \arccos\langle \vec{n}_1, \vec{n}_2 \rangle$$

depending on the configuration of the two faces with respect to $\vec{n}_i$, the normals of the faces under interest.

Definition 18.2 *The solid angle at a vertex is the surface area of the portion of a unit sphere centered at this vertex bounded by the three faces sharing this point.*

Exercise 18.3 *Find the equivalent definitions in two dimensions (for those cases where it makes sense).*

Exercise 18.4 *Give the normalization factors resulting in a unit value for the above quality measures (when it makes sense).*

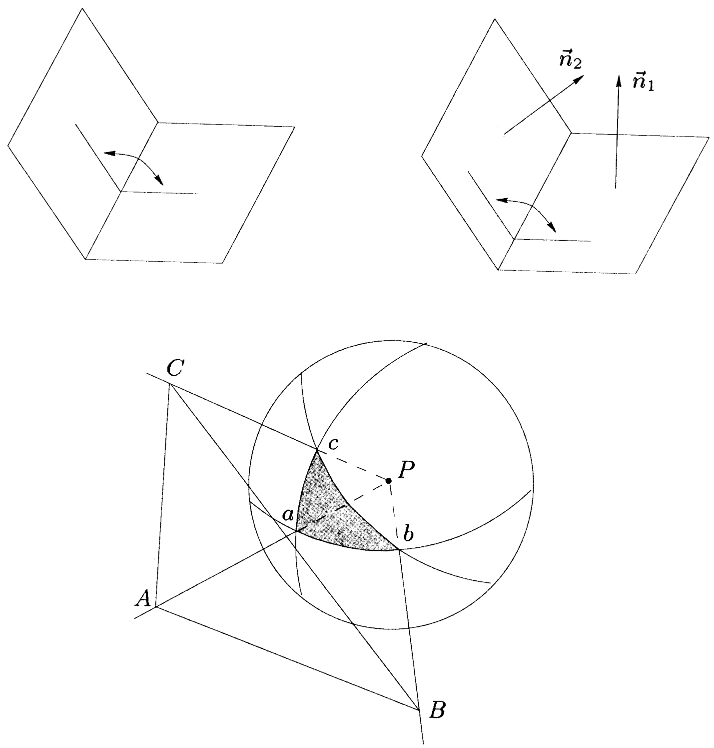

Figure 18.7: *Dihedral angle (top) and solid angle (bottom).*

18.2.3 Simplicial element classification

The above quality criteria enable us to decide whether a simplex is good or not in terms of quality. It could be interesting to classify the bad-quality elements so as to discard one or the other causes from which this bad quality results. This task is covered in the exercises (see also [George,Borouchaki-1997] where a detailed discussion is given).

Exercise 18.5 *Show that, in terms of the geometrical aspect, there exist three types of triangles (Hint : there are only two types of ill-shaped triangles). Show the criterion leading to this classification without ambiguity.*

Exercise 18.6 *Similarly, exhibit the eight types of tetrahedral elements (Hint : consider both the volume and the type of the element faces following the triangle classification).*

Exercise 18.7 *Discuss how the above measures allow (or are not suitable in all cases) for the element classification.*

18.2.4 Non simplicial elements

In this section, we discuss the quality of quad, hex and pentahedral elements.

Quad. In two dimensions[2], quads are usually appreciated through several quantities including the so-called *aspect ratio*, *skew parameter* along with two *taper coefficients* (one taper being related to one axis), see [Robinson-1987] and [Robinson-1988]. Although quite natural for some peculiar geometries, the above notions are not so well defined in a general context. Thus one can look for other types of measurements. One idea could be to find in the above series of measures those which apply in this case. For instance, the ratio $\frac{h_{max}}{h_{min}}$ where h_{min} and h_{max} denote the smallest and the largest edges can be considered. Nevertheless, this measure must be coupled with information about angles between two edges.

We would like to propose a new[3] formula which offers several advantages. First, it looks like Relationship (18.7). Second, it appears to be an efficient way to discriminate between the elements (in terms of quality). Then, negative or non-convex quads are detected on the fly. Finally, only one measure is involved. Thus, we propose the following :

$$\mathcal{Q}_K = \alpha \frac{h_{max}\, h_s}{S_{min}}\,, \tag{18.10}$$

where α is a normalization factor ($\alpha = \frac{\sqrt{2}}{8}$), S_{min} is the minimum of the four surface areas that can be associated with K. Indeed, these surface areas are those of the four triangles that can be defined (review the way to decide whether a quad is convex or not, Figure 18.1), $h_s = \sqrt{\sum_{i=1}^{4} L_i^2}$ with L_i the length of edge i of K and h_{max} is the longest length among the four edges and the two diagonals. In practice, we compute S_{min} and if this value is correct, we pursue the test, otherwise the quad being invalid, it is useless to continue.

Remark 18.5 *Another way to judge a quad is to consider that its quality is that of the worst triangle that can be constructed based on three of the quad vertices.*

Remark 18.6 *Following Remark 18.3, it is advisable to compute the inverse of the above quality function (to avoid a possible overflow when one of the surface areas involved is null).*

Figure 18.8 gives an impression of how the measure of Relation (18.10) varies as a function of the geometry of the analyzed quad. In this respect, the classical aspect ratio, skew parameter and taper coefficients as well as the global measure are displayed for an arbitrary variation.

Preliminary remarks about hex and pentahedral elements. Examining the quality of the element faces only gives a rough idea of the element quality. Indeed, the case of the *sliver* tet where the four faces are well-shaped while the volume is (almost) zero clearly indicates that the face qualities alone are not sufficient to qualify an element. In addition, quad faces are not necessarily planar

[2] See the next chapter for a surface quad.
[3] As far as we know.

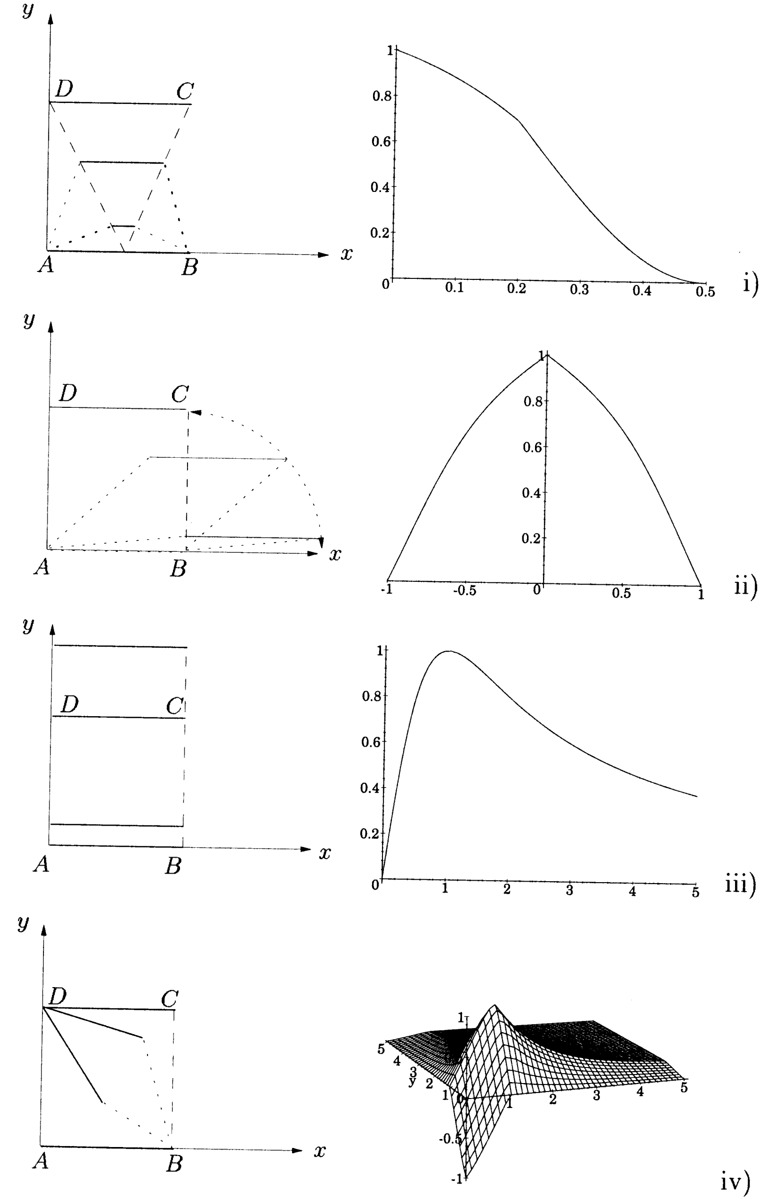

Figure 18.8: *Variation of the quality function in a quadrilateral (edge AB is fixed). Optimal quadrilateral is denoted by $ABCD$. i) taper measure : points C and D vary along the dashed line. ii) skewness measure : the point C varies along the circle of radius $1 = ||AB||$ (edge CD is parallel to AB). iii) aspect ratio measure : the edge CD (along the y-axis) varies in the interval* $[0,5]$. *iv) global measure : points A, B and D are fixed, point C varies in the interval* $[0,5] \times [0,5]$.

thus introducing some extent of torsion that must be taken into account to qualify an element with such a face. Thus, other measurements are needed which are not purely two-dimensional.

Quad face. In advance (see Chapter 19), we introduce the *roughness* (or smoothness) of a quad face. We consider the two diagonals of the quad $ABCD$, *i.e*, AC and BD and we consider the two dihedral angles that are defined in this way. Then, the smoothness of $ABCD$ is :

$$\mathcal{S}_{ABCD} = \min(\mathcal{P}_{AC}, \mathcal{P}_{BD}), \tag{18.11}$$

where

$$\mathcal{P}_{AC} = \frac{1 + \langle \vec{n}_{ABC}, \vec{n}_{ACD} \rangle}{2} \quad \text{and} \quad \mathcal{P}_{BD} = \frac{1 + \langle \vec{n}_{ABD}, \vec{n}_{BCD} \rangle}{2},$$

are the two *edge planarities* involved in the construction. In these relations, $\vec{n}_{ABC}$, for instance, denotes the unit normal related to triangle ABC.

In practice, the face smoothness measure will be used to quantify the *torsion* of a three-dimensional element.

Hex. As there is not a unique formula as for a quad or different criteria for analysis, it is possible to appreciate the quality of an element by observing the quality of the various partitions of it into tets. The worst tet then gives the desired answer which, combined with the element face qualities, enables us to conclude.

Pentahedral element. The previous idea can also be retained, the element analysis is based on a partition by means of tets.

18.3 Mesh quality (isotropic and anisotropic case)

The previous discussion allows mesh appreciation when the only concern is the shape aspect of the elements in the mesh, *i.e.*, without other considerations (such as metric specifications). We now turn to a different view point. In what follows, a metric map is supplied and the mesh is evaluated to decide if it conforms to this specification or not. In other words, the problem concerns the appreciation of a given mesh with regard to some specified size (isotropic case) or directional and size prescriptions (anisotropic case).

18.3.1 Efficiency index

Let l_i be the length of the edge i with respect to the given metric. The *efficiency index* of a mesh is defined as the average value of the squares of the differences to 1 of all mesh edge lengths (let na be the number of mesh edges), hence :

$$\tau = 1 - \frac{1}{na} \cdot \sum_{i=1}^{na} (1 - e_i)^2, \tag{18.12}$$

with $e_i = l_i$ if $l_i \leq 1$, $e_i = 1/l_i$ if $l_i > 1$.

This coefficient seems adequate for a quick estimation of the mesh quality with respect to a given metric map. Table 18.1 presents the sensitivity of this measure in the case of an isotropic map, l being constant for all mesh edges (which is highly unlikely although it shows the effect of the size variation on τ) and indicates that the edge lengths are l times too long or too short (a value $l = 5$ or $l = 0.2$ means that all edges are 5 times too long or 5 times too short). The optimal value is $l = 1$ and in fact, any value of τ greater than 0.91 ensures a reasonable mesh quality with respect to the metric map.

l	100	20	10	5	3	2	$\sqrt{2}$	1.3	1.2	1.1	1
τ	0.019	0.097	0.19	0.36	0.51	0.75	0.91	.9467	.9722	.9917	1.

Table 18.1: Sensitivity of the efficiency index.

The reader could consult this table in order to interpret the numerical results obtained in a given governed problem.

18.3.2 Element quality

In this section, we turn to a general point of view.

Simplicial elements. In this case element quality reduces to the above aspect ratio. When the metric map the mesh must conform to is isotropic, we return to Relation (18.6) or equivalent relations. In the case of an anisotropic metric map, the notion of aspect ratio is more delicate. In fact, an approximate expression can be used. For instance, in two dimensions, $\mathcal{Q}_K$, the quality of triangle K, can be defined as :

$$\mathcal{Q}_K = \max_{1 \leq i \leq 3} Q_K^i, \tag{18.13}$$

where Q_K^i is the triangle quality in the Euclidean space related to the metric specified at any vertex P_i of K. To evaluate the quality Q_K^i of K, one has just to transform the Euclidean space associated with the metric specified at any vertex P_i in the classical Euclidean space and to consider the quality of the triangle K_i which is the triangle image of K. This means that :

$$Q_K^i = Q_{K_i}. \tag{18.14}$$

Let $(\mathcal{M}_i)_{1 \leq i \leq 3}$ be the metrics specified at the vertices $(P_i)_{1 \leq i \leq 3}$ of K. We have :

$$\mathcal{M}_i = \mathcal{P}_i \Lambda_i \mathcal{P}_i^{-1} \qquad 1 \leq i \leq 3, \tag{18.15}$$

where $\mathcal{P}_i$ is the matrix enabling us to transform the canonical basis into the basis associated with the eigenvectors of $\mathcal{M}_i$ and $\Lambda_i = \begin{pmatrix} \lambda_{i,1} & 0 \\ 0 & \lambda_{i,2} \end{pmatrix}$ is the diagonal matrix formed by the eigenvalues of $\mathcal{M}_i$. Let $(h_{i,j})_{1 \leq i \leq j \leq 2}$ be the quantities defined by $h_{i,j} = 1/\sqrt{\lambda_{i,j}}$; these values represent the unit length in the direction of the eigenvector related to the eigenvalue $\lambda_{i,j}$ of $\mathcal{M}_i$. The matrix $\mathcal{T}_i$ transforming the Euclidean space associated with $\mathcal{M}_i$ in the usual Euclidean space is defined by :

$$\mathcal{T}_i = \mathcal{H}_i \mathcal{P}_i \,, \tag{18.16}$$

where $\mathcal{H}_i$ is the diagonal matrix $\begin{pmatrix} 1/h_{i,1} & 0 \\ 0 & 1/h_{i,2} \end{pmatrix}$. As a result, the vertices of K_i are $\mathcal{H}_i\mathcal{P}_iP_1, \mathcal{H}_i\mathcal{P}_iP_2$ and $\mathcal{H}_i\mathcal{P}_iP_3$, respectively, and we have :

$$Q_K^i = \alpha \frac{h'_{max} \displaystyle\sum_{1\leq j<k\leq 3} ||\mathcal{H}_i\mathcal{P}_i\overrightarrow{P_jP_k}||}{Det(\mathcal{H}_i\mathcal{P}_i\overrightarrow{P_1P_2}, \mathcal{H}_i\mathcal{P}_i\overrightarrow{P_1P_3})} \tag{18.17}$$

with α the same normalization factor as in the classical case, and

$$h'_{max} = \max_{1\leq j<k\leq 3} ||\mathcal{H}_i\mathcal{P}_i\overrightarrow{P_jP_k}||.$$

However, as

$$Det(\mathcal{P}_i) = 1 \,,$$

$$Det(\mathcal{H}_i) = \sqrt{\lambda_{i,1}\lambda_{i,2}} = \sqrt{Det(\mathcal{M}_i)}$$

and

$$||\mathcal{H}_i\mathcal{P}_i\overrightarrow{P_jP_k}|| = \sqrt{{}^t\overrightarrow{P_jP_k}\mathcal{M}_i\overrightarrow{P_jP_k}} \,;$$

we have

$$Q_K^i = \alpha \frac{\displaystyle\max_{1\leq j<k\leq 3} \sqrt{{}^t\overrightarrow{P_jP_k}\mathcal{M}_i\overrightarrow{P_jP_k}} \sum_{1\leq j<k\leq 3} \sqrt{{}^t\overrightarrow{P_jP_k}\mathcal{M}_i\overrightarrow{P_jP_k}}}{\sqrt{Det(\mathcal{M}_i)}Det(\overrightarrow{P_1P_2}, \overrightarrow{P_1P_3})} \,. \tag{18.18}$$

Exercise 18.8 *Show that the above relation reduces to the classical aspect ratio formula when we consider the case of isotropic metric maps (i.e., the corresponding matrices are identity matrices with a given ratio).*

Elements other than simplices. A quad with unit length edges and $\sqrt{2}$ length diagonals are the targeted values. In practice, these values are sufficient to appreciate the element (and no angle consideration is needed). Similar notions extend to hex and pentahedral elements.

18.3.3 Optimal mesh

A formal notion of an optimal mesh is relatively tedious to define. This question was already raised in Chapter 1, while noticing that optimality must be considered with regard to the reason for which the mesh has been constructed and therefore with regard to its further use.

Given $\mathcal{Q}_K$ a quality measure for the element K in mesh $\mathcal{T}$, we have previously defined the mesh quality as :

$$\mathcal{Q}_\mathcal{T} = \max_{K\in\mathcal{T}} \mathcal{Q}_K \,.$$

In a later section we will see that other characteristic values may be suitably used to analyze a mesh, such as the mean of the element qualities, the distribution of the elements based on their quality, etc. Whatever the choice, it may be noticed that the notion of optimality lies in theory in a family of meshes. For instance, given several meshes, one can say that the optimal mesh is that for which the chosen measure is optimal (minimal if we return to the usual definition about the shape of the elements in an isotropic simplicial mesh). In practice, the size (the number of vertices or elements) must also be used as one of the parameters in the analysis (so as to minimize the computational cost in a numerical simulation using this mesh, for example). Therefore, it could be stated that an optimal mesh is that for which the chosen quality function is optimal while, at the same time, its number of vertices (elements) is minimal.

In practical terms, it may be concluded that the optimal mesh is the one which gives a suitable compromise between various criteria. The problem then reduces to only one quality measure. In fact, in the isotropic case in two dimensions and for a mesh composed only of triangles, a quality value of 1 (*i.e.*, close to 1) implies that :

- the elements in the mesh have a quality value close to 1,
- the number of elements is minimal.

A quality value close to 1, for a triangle, implies that its edges have a length close to 1 (or a constant value h). For a given size map, we again see that this means that the edge lengths are close to 1 (with respect to this size map). Therefore, the definition we propose now is rather natural (and intuitive) :

Definition 18.3 *A unit mesh is a mesh with unit length edges.*

In two dimensions, a unit triangle (with unit edges) is optimal (it is equilateral) while this is not the case for a tet. In fact,

Remark 18.7 *A triangle with unit length edges is necessarily a good element whose surface area is* $\frac{\sqrt{3}}{4}$. *In contrast, a tet with unit length edges*[4] *may have a volume as small as we want thus corresponding to a ill-shaped element (the infamous sliver).*

Following this remark, the notion of optimality is made more precise. It includes a length aspect combined with a surface (volume) aspect.

Definition 18.4 *An optimal simplicial mesh is a unit mesh in which the element surface areas are* $\frac{\sqrt{3}}{4}$ *in two dimensions or in which the element volumes are* $\frac{\sqrt{2}}{12}$ *in three dimensions.*

Remark 18.8 *Notice (again) that optimality is here related to a quality measure with regard to a size map and not directly related to the number of elements (see the above remark about this aspect).*

[4] Consider a tet where four edge lengths are one and where the two other edge lengths are strictly $\sqrt{2}$. The edges have therefore a length "close" to 1 and nevertheless, the tet volume is null !

After the two above definitions together with the previous remarks and in practical terms, the efficiency index is a consistent way to judge a mesh in two dimensions. In three dimensions, the same analysis must be based on edge length appreciation and on the aspect ratio of the elements (in order to see whether the element volumes are consistent or not).

Remark 18.9 *For elements other than simplices, unit edge length could be a reasonable requirement as coupled with other considerations in some cases (for instance $\sqrt{2}$ length diagonals for a quad as previously mentioned).*

18.3.4 Remarks about optimality

Discussing optimality may raise to some interesting issues. Various questions may be discussed. For a given problem (based on what data are known) :

- is there a mesh with unit quality ?
- is there a mesh with minimal size ?

For an *a priori* given number of elements :

- is there a mesh having this number of elements and, if so, what is its quality ?

Note, in practice, and mostly in three dimensions, that the objective is to have good quality meshes (*i.e.*, with a quality value close to the theoretical value) and that a mesh not too far from this abstract target is generally considered to be satisfactory.

18.4 Tools for mesh optimization

Various local tools can be used for optimization purposes. The most popular include the following :

- node relocation,
- edge collapsing (to remove a vertex),
- edge swapping,
- face swapping,
- vertex degree relaxation (which, as will be seen, can be achieved by a judicious combination of the three above tools),
- edge splitting (to add a vertex), etc.

Actually, mesh optimization tools can be classified into two categories. Those maintaining mesh connectivity (*i.e.*, acting on the vertex positions) and those acting on mesh connectivity (while the vertex locations are preserved).

18.4.1 Optimization maintaining the connectivities

In this category of local tools, we basically encounter those methods which result in moving the element vertices. All methods that can be developed in this sense can be seen as a variant of the well known Laplacian smoothing, [Field-1988], [Frey,Field-1991].

Basically, a process acting on node relocation concerns the so-called balls. Let us recall that :

Definition 18.5 *Let P be a vertex in mesh $\mathcal{T}$, the ball associated with P is the set of elements in $\mathcal{T}$ having P as a vertex.*

A ball could be a closed ball (the vertex is an internal vertex) or an open ball (the vertex is a boundary vertex). In the following, we only consider closed balls.

The simplest node relocation method can be written as :

$$P' = \frac{1}{n}\sum_{j=1}^{n} P_j\,, \tag{18.19}$$

where the P_j's are the vertices of the ball other than P. A first variation consists in adding some relative weights, thus, we obtain :

$$P' = \frac{\sum_{j=1}^{n} \alpha_j P_j}{\sum_{j=1}^{n} \alpha_j} \tag{18.20}$$

where α_j is an appropriate weight associated with point P_j.

Nevertheless, before going further, we propose replacing the above scheme by a relaxation method. In fact, efficiency reasons can be involved along with the two following observations.

Remark 18.10 *Above point P' (one or the other) could fall outside the ball in the case of a non-convex ball.*

and, as a consequence

Remark 18.11 *Moving a given point to its "optimal" location may result in an invalid mesh, thus cancelling the operation. A non-optimal relocation, however, may improve the mesh quality to some degree.*

Hence, an auxiliary point, P^*, is introduced. For instance such as in Relationship (18.19) :

$$P^* = \frac{1}{n}\sum_{j=1}^{n} P_j\,, \tag{18.21}$$

and a relaxation scheme is defined as :

$$P = (1-\omega)P + \omega P^*\,, \tag{18.22}$$

where ω is the relaxation parameter[5]. Following this method, we now introduce various vertex relocation methods.

[5] It seems advisable to set ω close to one in two dimensions and smaller in three dimensions.

Laplacian smoothing. The relaxed variant of this well-known method uses auxiliary point defined in Relationship (18.21). Following the previous remark, an explicit check of the positiveness of the surface areas (volumes) is required.

Remark 18.12 *Classical optimization strategies can be used for smoothing purposes. A cost function is defined which is assumed to be sufficiently smooth. Descent directions are then exhibited and the process is governed as in a classical optimization process, [Freitag,Gooch-1997]. It should be noted that most of the quality functions described above are non differentiable thus leading to a tedious optimization process.*

Weighted smoothing. In this case, a weight is associated with each point making it possible to define the auxiliary point by

$$P^* = \frac{\sum_{j=1}^{n} \alpha_j P_j}{\sum_{j=1}^{n} \alpha_j}, \tag{18.23}$$

where an appropriate choice of the α_j's must be made (see below).

Smoothing based on element quality. Provided with a simplicial mesh, we consider the ball of a given point P. Let f_j be the external edges (faces in three dimensions) of this ball. Then the elements in the ball are nothing more than the combinations (P, f_j) (where $j = 1, n$, n being the number of elements in the ball). Thus an ideal point I_j is associated with each f_j in such a way as to ensure an optimal quality (aspect ratio) for the virtual element (I_j, f_j). Using these points, we define the smoothing process as :

$$P^* = \frac{\sum_{j=1}^{n} \alpha_j I_j}{\sum_{j=1}^{n} \alpha_j}, \tag{18.24}$$

where the α_j can be defined as follows :

- $\alpha_j = 1$, the weights are constant and we return to the classical method,
- $\alpha_j = 0$, for every element of the ball except for the worst one (in terms of quality) for which we take $\alpha_j = 1$,
- $\alpha_j = Q_{K_j}$, the weights are related to the quality of the elements,
- $\alpha_j = Q^2_{K_j}$, the weights are related to the square of the element qualities,
- or, more generally, $\alpha_j = g(Q_{K_j})$, meaning that the weights are related to a certain function g of the quality of the elements in the ball.

This method can also be applied in an anisotropic case by using the relevant definition of the quality, which leads to a different positioning of the points I_j.

Smoothing based on edge lengths. In this case, the key-idea is to define the I_j's so as to obtain as far as possible a unit length for the edges emanating from these points. The unit length notion is based on the metric map which is assumed. Thus, a relation like :

$$I_j = P_j + \frac{\overrightarrow{P_jP}}{||\overrightarrow{P_jP}||}\,\overline{h}_j\,, \tag{18.25}$$

where the P_j's are the vertices of the external faces in the ball and $\overline{h}_j$ is the average size related to the edge P_jP approaches the desired result. Indeed, the above relation is nothing more than :

$$I_j = P_j + \frac{\overrightarrow{P_jP}}{l_{\mathcal{M}}(P_jP)}, \tag{18.26}$$

where $l_{\mathcal{M}}(P_jP)$ is the length of the edge P_jP evaluated in the metric $\mathcal{M}$ associated with the edge.

Non simplicial elements. To some degree, the above methods extend to this case. In particular, for a quadrilateral element, an edge length based smoothing operator must lead to unit edge lengths and a $\sqrt{2}$ length for the two diagonals of the elements (as previously indicated).

Global smoothing. The above discussion concerns a local smoothing procedure where the points are considered one at a time. A global smoothing procedure can be developed leading to the solution of a global problem. It is, however, uncertain whether such a method is really efficient.

Topologically driven node relocation. The objective is now to relocate a vertex and, in addition, to

- move it away from a given edge,
- move it away from a given plane,
- move it along a given edge,
- etc.

Any method of the above type can be used by adding the constraint during the analysis of the criterion that must be optimized. In some cases, the goal is not to optimize the mesh but just to maintain some degree of quality, meaning that the main concern is to remove an undesired topological pattern rather than to effectively optimize the mesh.

Exercise 18.9 *Revisit the smoothing techniques in the case where a ball may be an open ball (for a boundary vertex for instance).*

18.4.2 Optimization maintaining the vertex positions

We turn to various local tools that leave the vertex location unchanged.

Edge swapping in two dimensions. Edge swapping[6], or simply swap, is a rather simple topological operation leading to swap the edge shared by two elements. In the case of triangular elements, the swap is possible since the quad formed by the two adjacent triangles is a convex polygon. Swap can also be performed in quad meshes or mixed meshes (whose elements include both triangles and quadrilaterals). In general, a swap procedure must first be validated to ensure that the resulting mesh is still valid and, second, be evaluated with regard to the optimization criterion that must be enhanced.

Edge swap can be seen as an optimization procedure in itself or it can be used as one ingredient in some more sophisticated processes (node removal, degree relaxation, etc.).

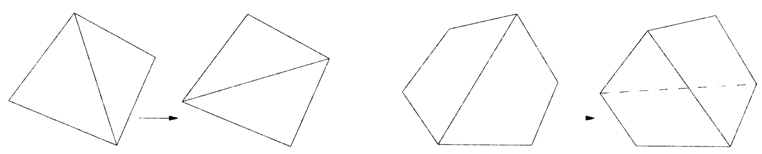

Figure 18.9: *Edge swapping for a pair of adjacent triangles (left-hand side) and in the case of adjacent quads (right-hand side) where two solutions are possible* a priori.

Generalized edge swapping (the three-dimensional case). In what follows, only simplicial meshes are discussed. Within this context, we first consider the extension to three dimensions of two-dimensional edge swapping. This leads to a *face swapping* where the face common to two tetrahedra is removed, an edge is created and the two-element initial polyhedron is replaced by a three tetrahedron polyhedron (note that only convex polyhedra can be successfully dealt with).

The inverse local transformation can be defined leading to replacing three tetrahedra sharing an edge by two tetrahedra by suppressing the edge considered. Actually, a more general transformation corresponds to this operation. It acts on a so-called *shell*. Recall :

Definition 18.6 *Let $\alpha\beta$ be an edge in mesh $\mathcal{T}$, the shell associated with $\alpha\beta$ is the set of elements sharing this edge.*

As for balls, a shell could be opened or closed. In what follows we only discuss the case of closed shells where the edge of the shell is an internal edge.

Then the key-idea is to consider such shells. Formally speaking, the generalized edge swapping operator leads to considering all the possible triangulations of a

[6]Also referred to as diagonal swapping or diagonal flipping.

pseudo-polygon associated with the edge. The vertices of this polygon are defined by the shell vertices other than α and β, the two endpoints of the edge defining the shell. Figure 18.10 shows these possible remeshings in the case of a five-element shell, every triangulation being formed by joining all the triangles of the polygon with both α and β, so as to define the pair of tetrahedra which are part of the desired mesh.

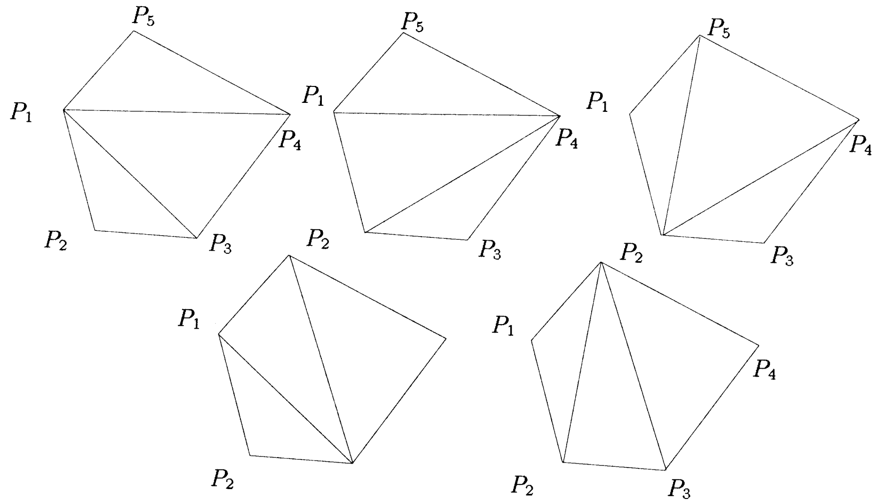

Figure 18.10: *The five triangulations related to a five-point polygon.*

The Catalan number of order n

$$Cat(n) = \frac{(2n-2)!}{n!(n-1)!}$$

gives the maximal number of *topologically* possible triangulations, N_n, of a shell[7] of n elements. Indeed, we have

$$N_n = Cat(n-1)\,. \qquad (18.27)$$

Exercise 18.10 *Establish the previous relation.*

n	3	4	5	6	7	8	9	10	11	12	13
N_n	1	2	5	14	42	132	429	1,430	4,862	16,796	58,786
Tr_n	1	4	10	20	35	56	84	120	165	220	..

Table 18.2: Number of topologically different triangulations that are valid as a function of the number of vertices in the polygon related to one edge.

[7]The topologically possible solutions in three dimensions are constructed by enumerating all the two-dimensional *geometric* valid remeshings of a convex (planar) polygon with n vertices

Table 18.2 gives N_n, the number of possible triangulations as a function of n. It also indicates Tr_n the number of different triangles in each possible triangulation. In this enumeration, the validity of the triangulations is not considered (only the topological aspect is taken into account).

Remark 18.13 *As previously mentioned, the swap procedure requires that the polyhedron under treatment is convex for a two or three element pattern while this requirement is not strictly needed for patterns involving more than three elements. Indeed, the swap must be validated by checking explicitly the positiveness of the element volumes that are concerned.*

Remark 18.14 *The generalized swap is a tedious problem for elements other than simplices.*

Edge collapsing. Provided with an edge, $\alpha\beta$, we replace this edge by only one point A. Formally speaking, this leads to positioning α on β, or conversely or again finding a point location between α and β. Figure 18.11 shows the three possible solutions on an example. From a practical point of view, it is sufficient to check if the ball of point A, resulting from the reduction, is valid. To this end, we examine the shell $\alpha\beta$ and we check the validity of the balls of α and β when these two vertices are replaced by the point A.

This operator may be classified among the geometric operators as it maintains the connectivities, if the new connections to A are seen as the "union" of the former connections to α and β.

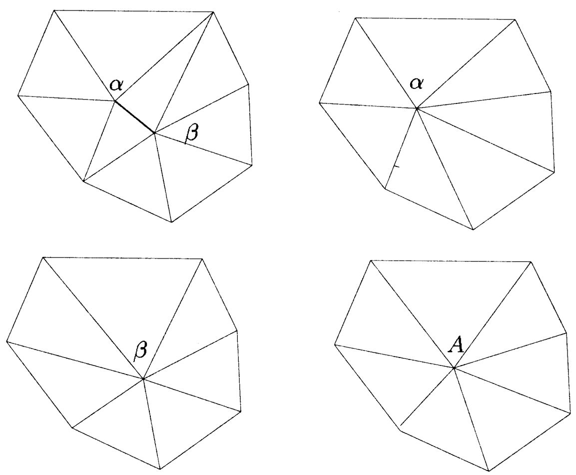

Figure 18.11: *Edge collapsing. The initial pattern can be replaced by three different configurations, vertex β is collapsed with vertex α, vertex α is collapsed with vertex β or these two vertices are collapsed using vertex A, for example the midpoint of the initial edge.*

Degree relaxation. First, we give the definition of the degree of a mesh vertex.

Definition 18.7 *The degree or the valence of a mesh vertex is the number of edges*[8] *emanating from this point.*

We consider the edges and not the elements because in the matrices used in finite element calculus and for P^1 simplicial meshes, the edges determine the matrix bandwidth.

Now the question is deciding what an optimal degree is. In two dimensions, a value of six is desired for simplicial meshes while a value of four is optimal for quad meshes. In three dimensions, a value of twelve allows for tetrahedral meshes[9] while six is the optimal degree of a vertex in a hexahedral mesh.

A point with a degree less than the targeted value is said to be *under-connected*, while a point with a degree larger than this value is said to be *over-connected*. Note that the same notion applies to an edge.

Relaxing the degree of a mesh consists in modifying the vertex (or edge) degrees, by means of topological operators, so as to tend on average to the targeted value, see [Frey,Field-1991].

Remark 18.15 *Optimizing a mesh with bad vertex degrees may result in poor results. This means that the optimization tools are penalized when dealing with such situations. On the other hand, optimization tools may lead to nice results when a degree relaxation has been carried out beforehand.*

Ill-constrained entities. This notion applies to vertices (as previously seen) as well as to edges, faces or elements, which will now be discussed.

Definition 18.8 *A triangle is said to be over-constrained if two of its edges are members of the domain boundary. Similarly, a tet with three boundary faces is over-constrained.*

Thus, in terms of edges or faces (extending the previous definition to elements other than simplices), we have the following :

Definition 18.9 *An internal edge is said to be over-constrained if its two end-points are members of the domain boundary. Similarly, an internal face whose extremities are in the boundary is over-constrained.*

For most problems, such ill-constrained entities must be avoided. Thus, optimization tools can serve to suppress this kind of pathologies.

Exercise 18.11 *Consider again the modification operators when the shells are opened (i.e., those associated with a boundary edge).*

[8] It is also, in two dimensions, the number of elements sharing the point.
[9] This value corresponds to a ball with 20 optimal elements, *i.e.*, the triangulation by a regular icosahedron of the space centered at the point defining the ball.

18.4.3 Non-obtuse mesh

This point concerns simplicial meshes in two dimensions. Non-obtuse meshes are required for some particular applications. For instance, a problem solved by means of a finite volume method takes advantage of non-obtuse meshes. To specify the notion of a non-obtuse mesh, we firstly give the formal definition of such a mesh.

Definition 18.10 *A two-dimensional simplicial mesh is said to be non-obtuse if it does not include any obtuse angles, such angles being defined by the pairs of edges sharing a vertex.*

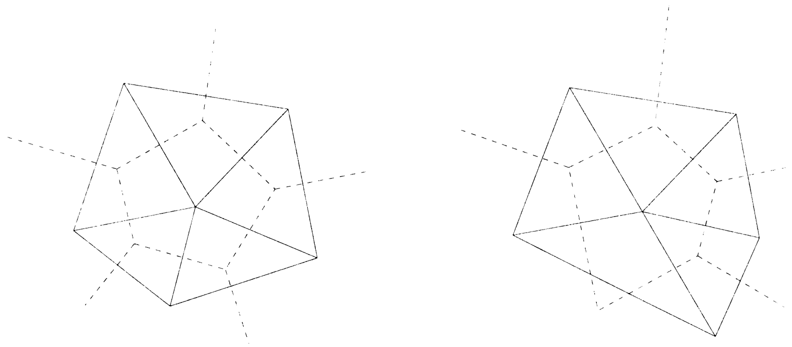

Figure 18.12: *Obtuse and non-obtuse meshes. The cells associated with the triangles sharing a given vertex are displayed. As mesh is non-obtuse, the cell around the point is fully included in the ball of this point (left-hand side) while, on the other hand, it has a part outside this ball (right-hand side) when the mesh is obtuse.*

As mentioned earlier, non-obtuse meshes are of special interest in some applications. This is due to the fact that the cells around the element vertices are fully included in the corresponding balls. Note that the cells are constructed from the perpendicular bisector related to the element edges (Figure 18.12). The fundamental property is the orthogonality of these cells and the current mesh (more precisely the mesh edges and the cell edges are orthogonal). Hence, this nice feature coupled with the internal aspect of the cells can be a benefit for some problems.

Remark 18.16 *We return here to the Voronoï cells , the duals of the triangulation (see Chapter 7)* **only** *in the case of a Delaunay mesh. In this respect, one could note that even a Delaunay mesh is not necessarily a non-obtuse mesh. See, for instance, Figure 18.13 where a point outside the circle ensures the Delaunay property but could be outside the two lines depicted in the figure thus resulting in an obtuse angle.*

Clearly, a non-obtuse mesh allows the construction of cells enjoying the above properties (*i.e.*, an orthogonality property while entirely inside the domain[10]).

[10] Note that defining the cells around the vertices by means of the median lines results in cells fully inside the domain but the orthogonality feature is lost.

Thus, given a mesh resulting from such or such a method, it could be interesting to develop an algorithm that allows us to suppress the obtuse angles (if any).

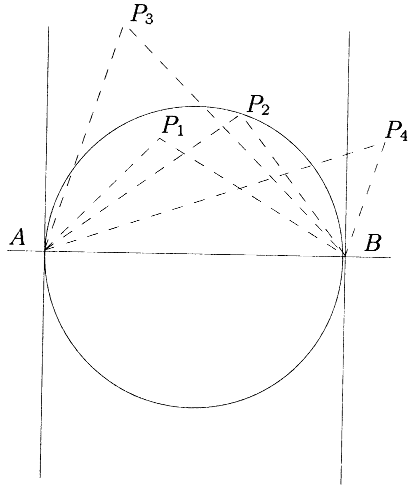

Figure 18.13: *Given an edge AB, we define the circle whose diameter is AB. Then we construct the two lines orthogonal to AB passing through A and B. This results in three regions. The interior of the circle, the exterior of this circle exterior to the two above lines and the exterior of this circle inside the two lines. Clearly a point like P_1 (or P_4) leads to an obtuse triangle and a point like P_3 leads to a non-obtuse triangle.*

A precise analysis of Figure 18.13 can be fruitful to define an algorithm based on local modifications that makes it possible to suppress the obtuse angles.

A rough idea of the method could be to process all balls in the mesh (see Definition 18.5). Let P be the vertex defining a ball, we consider the edges external to this ball (thus, these edges act as the edge AB in Figure 18.13). We define the circles and the two orthogonal lines associated with these edges to exhibit a region intersection of the suitable area where P can be located. If this region is non empty, then P is relocated inside and all the elements in the ball are non-obtuse[11]. Otherwise, a more subtle process must be defined. For instance one can try to move A and/or B along AB so as to obtain a smaller edge thus resulting in a smaller circle. Note also that vertices opposite a boundary edge have some degree of rigidity. To overcome this fact, points can be required to further subdivide the boundary edges.

Remark 18.17 *In the above method, essentially based on heuristics, no proof of convergence is given. Nevertheless, careful use of the classical optimization tools governed by the previous scheme, results in the desired solution in most cases.*

[11] For the sake of simplicity, we use the term non-obtuse triangle to describe an element whose angles are non-obtuse.

To conclude, one should note the analogy with the condition of Delaunay admissibility for an edge as described in Chapter 9.

Delaunay triangulations and non-obtuse meshes. As pointed out, a Delaunay triangulation in two dimensions is not necessarily a non-obtuse triangulation. In fact we have a property of maximization about the minimum angle included in a pair of adjacent triangles and not the opposite property. Nevertheless, a method for point placement can be found in [Chew-1989b] which results in a bound for the angle of the mesh element in the case of a Delaunay strategy for vertex connection. Given a domain, *i.e.*, a polygonal discretization of its boundary, a Delaunay algorithm based on the boundary vertices and using as internal points the circumcenters of the elements allows a mesh where the angles are bounded. Note that when a circumcenter falls outside the domain, the corresponding boundary edge is subdivided by introducing its midpoint.

Following this remark where an upper bound on the angles exists, one could observe that a Delaunay triangulation is not necessarily non-obtuse. Conversely, a non-obtuse triangulation is necessarily a Delaunay triangulation.

Now, we look at what could be the extension to three dimensions of the notion of a non-obtuse triangulation. Before introducing a reasonable characterization, we return to the two-dimensional case. Given a triangulation, if we consider the circles of minimum radius (the smallest circles) that enclose the triangles, then for a non-obtuse triangulation, we have the following properties :

- the smallest circle enclosing a non-obtuse triangle is its circumcircle,
- a non-obtuse triangle is *self-centered*, meaning that its circumcenter falls inside the triangle.

Thus, based on the above property, a triangulation in three dimensions is termed non-obtuse if all of its tets are self-centered.

18.5 Strategies for mesh optimization

First we compute the initial quality of the point, the edge, the element or the set of such entities included in the initial configuration. We then compute the same quality for the entity or all the entities related to the solution or the different possible solutions based on a simulation. Finally, we decide to effectively apply the optimization process, as a function of the quality evolution. Several strategies can be chosen to govern the decision. The process is applied :

- if the resulting configuration is strictly improved,
- if the resulting configuration is improved to some extent,
- in the case of multiple possible solutions, by selecting the first valid solution occurring in the simulation or by choosing the best solution among all,

- and so on.

Another issue consists of defining the way in which the operator is used. One can decide to process

- all the mesh entities starting from the first and going to the last,
- only some entities selected *ad-hoc* (using a heap based on a relevant criterion, edge length if edges are to be processed, or using a quality threshold, etc.),
- all the entities, or only some of them, randomly picked,
- and so on.

The question is now to design an automatic and global optimization method by deciding on a strategy for both the choice of the local operators and the order in which to use them (cf. hereafter), assuming that the strategy related to a given local operator is fixed.

Several observations can be helpful in defining such a strategy. Assuming that a local operator sequence is given, a stopping criterion must first be defined. In fact, several classes of criteria are possible, as indicated below. The process is repeated as long as :

- the mesh is affected by one operation,
- the mesh is affected by one or several operations,
- a given threshold (in terms of quality) has not been achieved,
- and so on.

Once this has been decided, a strategy must be defined. There is some flexibility regarding the possible choices. Indeed, one can

- apply every local operator to all the entities concerned by its application, and then turn to a different operator,
- consider a given mesh entity and apply all the possible local operators before turning to a different entity,
- combine the two above approaches.

It is also possible to classify the pathologies following the degree of optimization that could be expected and to deal with the mesh entities accordingly. In other words, the worst entities are dealt with first.

Remark 18.18 *An immediate question about an optimization process is to know if the optimum has been reached. In practice, the purpose is to improve the mesh and achieving the optimum or not remains a purely theoretical, non trivial issue. For example, the presence of wells, the fact that the function in optimization is differentiable or not, etc., are parameters that act on the conclusion. From a practical point of view, using some degree of* randomization *in the possible choices may, in most cases, avoid the cases where a well is found. On the other hand, looking for a strict optimum may turn out to be costly and, ultimately, for a relatively little gain in efficiency.*

18.6 Computational issues

In this section, we discuss some computational aspects related to the above tools. First, we consider how to construct the balls or shells which, as previously seen, are the local supports of the procedure. Then we give some indications about how to develop optimization tools.

Ball construction. For a simplicial mesh, we refer the reader to Chapter 2 where some solutions resulting in a ball construction are described. For other types of meshes, similar methods can be defined.

Shell construction. Again Chapter 2 presents a method for shell construction in simplicial meshes that can be extended to the other mesh types.

Choice of a criterion. As previously indicated, there may exist several criteria for quality evaluation. In such a case, it is necessary to decide which criterion to choose. In practice, if we take two different criteria which vary in the same way (*i.e.*, both measure the quality in any case), optimizing one of these automatically leads to optimizing the other.

Computing a criterion. When the simulation of a given optimization tool includes a large number of possible solutions (as is the case when considering the generalized swap for a shell), CPU cost considerations impose the optimization of the "simulation-effective application" pair. The operations used in the simulation are of a purely geometric nature (surface or volume computations, element quality evaluations, before and after the process in question has been applied), while the effective application of the operator leads to defining the new elements and the new neighborhood relationships (if these must be maintained) that can be affected in the process. A careful computer implementation of these two phases enables us to minimize the global CPU cost of the whole process.

In order to reduce the cost it could be noted that some quantities involved in a given optimization process, while part of the global evaluation of the configuration, remain constant during the process (for instance the external faces of the ball of a given point P are not affected by any relocation of point P. In this example the neighboring relationships are also preserved).

General scheme for a local optimization procedure. Using an optimization operator is quite simple. First, its result is simulated regarding both the validity and the quality evolution of the elements concerned. If an improvement is observed in the simulation phase, the simulated output is retained. When several solutions are possible, the best one is selected (or the first possible solution which has been observed). Thus, instead of computing the full criterion, one could look first at the surface (volume) and if it is negative, there is no point in pursuing the computation. As a consequence, when numerous criteria of this type must be evaluated, one could first compute all the surfaces and, if one of these quantities is wrong, stop the process.

Exercise 18.12 *Return to Table 18.2 and examine how the CPU cost could be minimized.*

18.7 Application examples

In this section, we first indicate how to judge a mesh and then we give some particular examples of mesh optimization.

18.7.1 Mesh appreciation

Mesh analysis is a crucial and difficult point when investigating meshes with a large number of elements, especially in the three-dimensional case. Two complementary approaches can be followed : a graphic visualization and a purely numerical analysis.

Both methods have advantages and drawbacks. First of all, in some cases, using graphic software could be helpful to give some idea of the appearance of the mesh aspect or its quality. Nevertheless, despite the powerful graphic software available, this method of mesh appreciation could be unsuitable when meshes with a large number of elements are considered (the screen is too black) or, simply, for three-dimensional meshes. In addition, the CPU time necessary to display a large mesh could be rather long.

On the other hand, numerical analysis of a mesh must be defined carefully. The aim here is to find one or several pertinent criteria that are easily readable and reflect unambiguously the aspect of the mesh one wishes to examine.

Visualization. For efficiency, graphic visualization must offer numerous tools which allow the easy examination of the mesh under consideration. In addition, these tools must be incorporated in a system which must be as user-friendly as possible.

In terms of mesh correctness, a *shrink* applied to the mesh elements is a fast way to detect any overlapping or defaults of connectivity. Nevertheless, it is not so easy to detect a negative surface (volume) element. One possible way to make this check possible is to associate a color with the element, this color being related to an oriented normal.

In terms of quality functions, using colors and cuts (in three dimensions) allows us to display some isocontours of these functions.

Numerical appreciation. The numerical verification of meshes is based on the computation of quantities associated with them. For example, in terms of mesh correctness, it is of interest to be sure that both the surfaces or volumes of elements are all positive and that mesh connectivity is right. In terms of mesh quality, element quality extrema, average element quality and histograms showing the distribution of elements according to their quality give a quick understanding of the mesh under consideration.

Positiveness of element surface areas or volumes is obvious to check by simply computing these quantities. To check whether or not a mesh is correct in terms of connectivity, one has to construct the adjacency graph associated with the mesh (*i.e.*, to establish for every element the list of its neighboring elements) and to verify that this graph is closed[12].

Mesh quality is easy to obtain by computing the quality of all the entities concerned based on the quality function we are interested in. Then various numerical values (extrema, mean value, adequate norms, etc.) as well as histograms of distribution of the analyzed entities according to their quality can be used.

18.7.2 A few examples

We now give an example in two dimensions (for the sake of clarity) that concerns the optimization of a triangular mesh. Figure 18.14 shows the mesh in its initial configuration (left-hand side) and after being optimized (right-hand side). In this case (in two dimensions), a simple view makes it possible to see the efficiency of the optimization process since the mesh includes a reasonable number of elements.

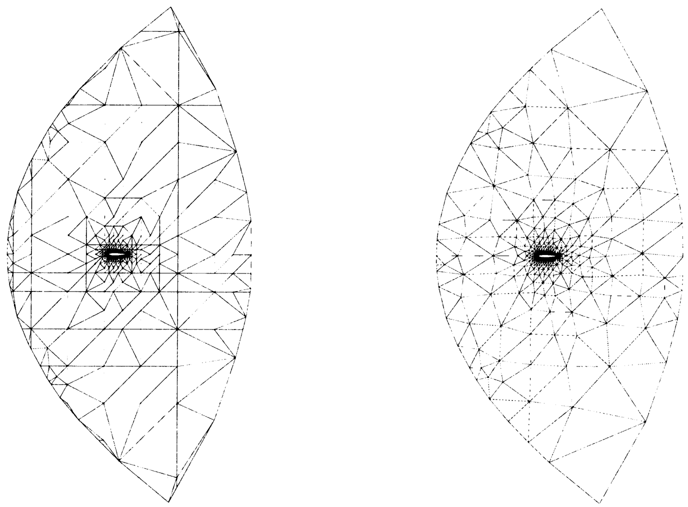

Figure 18.14: *Optimization of a mesh in two dimensions. The brute mesh (quadtree) of the domain (left-hand side) and the optimized mesh (right-hand side).*

To demonstrate the effect of shape optimization in three dimensions, we observe the distribution of the mesh elements with regard to their quality value. As a visualization has no real interest, Table 18.3 presents various parameters characterizing the mesh before and after optimization. The example given (after [George,Borouchaki-1997]) corresponds to a tet mesh.

[12]For a non manifold mesh, a more subtle method must be considered. For instance, an edge on a surface mesh could be shared by more than two elements.

Q	$1-2$	$2-3$	$3-10$	>10	*target*	$\mathcal{Q}_\mathcal{T}$	*ne*	*np*
before	63	21	12	3	8.30	755	3,917	1,004
·	2,475	824	486	132				
after	72	21	6	0	8.30	11.44	3,608	1,015
·	2,620	759	224	5				

Table 18.3: Shape quality of the tet mesh before and after optimization.

Table 18.3 gives, in the first line, the distribution, as a percentage of the total number of elements, of the elements according to their quality and the ranks from 1 to 2, from 2 to 3, from 3 to 10 and larger than 10. In the second line, we give the number of corresponding elements. The value denoted as *target* is the quality value of the best possible tet that can be created based on the worst face in the domain boundary. $\mathcal{Q}_\mathcal{T}$ is the global quality value of the mesh, *i.e.*, the value of the worst element, *ne* and *np* respectively note the number of elements and the number of vertices in the mesh. The last two lines show the same quantities for the mesh after optimization. A rapid examination of these figures gives an immediate impression of the efficiency of the optimization process.

Specifically, one can see that $\mathcal{Q}_\mathcal{T}$ is close to *target* (*i.e.*, in the same range) and that the percentages of elements in the various quality ranges have been changed as desired. Nevertheless, there are still some elements with a relatively poor quality. This fact is generally due to two reasons. The worst quality (the value *target*) is not necessarily attainable if the domain, for instance, is such that moving the point related to the worst face in the boundary is not possible (or the required swaps at some vicinity of this face are not permitted). Moreover, the optimization process, due to its computer implementation in terms of strategy, does not always lead to the optimum (as previously indicated). Note also that the optimization procedures have been applied only on the internal vertices and edges in the mesh, the boundary mesh entities remaining unchanged.

For other examples, when the optimization criterion is no longer the element shape but includes some other aspects (via a metric, for example), we suggest that the reader refers to some other chapters in the book which deal more specifically with this problem.