Chapter 19

Surface mesh optimization

Introduction

In Chapter 18, we described several methods to optimize planar or volumic meshes based on criteria notably related to the shape and the size of the elements. We mentioned that these methods cannot generally be applied directly to surfaces. This is why we now deal with the optimization of surface meshes which, while using the same general principles as the methods described in Chapter 18, nevertheless presents numerous specificities.

Surface meshes play an important role in various numerical applications. Hence, for finite element methods, it is well established that the quality of the geometric approximation may affect the accuracy of the numerical results as well as the convergence of the computational scheme [Ciarlet-1991]. In this type of application, a surface mesh is conceived, in principle, as the description of the boundary of a computational domain in three dimensions (cf. Chapters 5 to 7). Actually, to be useful, these meshes must conform to certain criteria, related to the geometry of the surfaces they represent (we expect an element size variation based on the local curvature) or to the physical behavior of the problems studied (element density greater in regions where the gradient of the solution varies). But, in the last case, following a physical criterion does not mean excluding conformity to the geometric properties of the surface.

Moreover, it is frequent that a given surface mesh is not satisfactory, either because it corresponds to too coarse an approximation of the surface, or because it contains too many elements to be exploitable. We focus here on the optimization of such a mesh with respect to the geometry it represents, so as to obtain a mesh of a geometric nature (for which the gap between the discretization and the geometry of the surface is bounded by a given tolerance value) and such that the quality (in shape and/or in size) of the triangles and/or the quadrilaterals is acceptable for finite element calculations.

Surface mesh modification and optimization operators need to access certain information about the surface and its properties. This information serves, in particular, to position a point on the surface or to locate a point on the surface, given a point and a direction. If such a surface is known exactly (given an analytical definition, for example) or indirectly (using queries to a geometric modeling system, for example), this information is easily accessible. However, if the input is already a mesh, the surface approached by the polyhedral representation is not known. The given discretization will then be used to construct a geometric support (*i.e.*, a mathematical representation) having adequate regularity and continuity properties. This support can then be queried to obtain the information about the surface required by the optimization process.

Once the intrinsic properties of the surface are known (or at least estimated), it is possible to construct the metric of the tangent plane (cf. Chapter 15). This metric will be used to govern the mesh modification and mesh modification procedures. In particular, the edge lengths are calculated in this metric[1].

★
★ ★

In this chapter, we specify the criteria and the methods used to optimize surface meshes. The first section introduces the shape and size quality measures adapted to surface meshes. In the second section, we indicate how to retrieve, from the given discretization, the intrinsic properties of the surface (radii of curvature, normals, etc.) and thus to construct the metric associated with the tangent planes that will be used to govern optimization algorithms. The third section deals with the problem of constructing a geometric support when the input is a mesh. Mesh optimization operators and algorithms are introduced respectively in the fourth and fifth sections. Finally, several examples of optimized surface meshes are given in the sixth section. Examples of simplified meshes are also proposed to illustrate a particular application of the optimization, the surface mesh simplification.

19.1 Quality measures

In this section, we deal with how to evaluate the quality of surface meshes. This information will be useful during a global surface mesh optimization procedure. We mainly examine the case of meshes composed exclusively of triangles, giving some indications nonetheless about quadrilateral meshes.

19.1.1 Surface mesh quality (classical case)

Recall that (cf. Chapter 18) the shape ratio of a triangle K is defined by the relation :

$$\mathcal{Q}_K = \alpha \frac{h_{max}}{\rho_K} = \alpha \frac{h_{max} p_K}{S_K} \,, \tag{19.1}$$

where h_{max} is the diameter of K, ρ_K is the in-radius, p_K is the half-perimeter and S_K is the area of K. The coefficient α is chosen in such a way that the quality of an equilateral triangle is equal to one.

[1] Which may also be combined with a "physical" metric which is, for instance, representative of the behavior of a solution during a numerical computation.

This formula indicates the degradation of an element K as compared with the equilateral triangle. In practice, we use the inverse value of $\mathcal{Q}_K$, to avoid numerical problems.

Remark 19.1 *Notice that this quality measure makes it possible to appreciate a triangle with no other metric consideration but the geometry.*

This element quality measure allows us to define a more global quality measure, for a whole set of triangles. Hence, the quality of the ball $\mathcal{B}(P)$ of a point P is given by :

$$\mathcal{Q}_{\mathcal{B}(P)} = \max_{K \in \mathcal{B}(P)} \mathcal{Q}_K \,. \tag{19.2}$$

By extension, the quality of a surface mesh $\mathcal{T}$ is defined by the relation :

$$\mathcal{Q}_{\mathcal{T}} = \max_{K \in \mathcal{T}} \mathcal{Q}_K \,. \tag{19.3}$$

We can, similarly, define an average quality value for a mesh, using the relation :

$$\overline{\mathcal{Q}_{\mathcal{T}}} = \frac{1}{ne} \sum_{i=1}^{ne} \mathcal{Q}_{K_i} \,, \tag{19.4}$$

where ne denotes the number of mesh elements.

19.1.2 Surface mesh quality (general case)

If size specifications are given or if a metric map is provided, the previous approach is slightly modified. One now has to decide whether or not the current mesh conforms to these specifications.

Efficiency index. Let l_{AB} be the length of an edge AB in the metric specified. The *efficiency index* allows us to estimate the average deviation of the lengths as compared to 1 (the reference value). More precisely, this index is defined as :

$$\tau = 1 - \frac{1}{na} . \sum_{i=1}^{na} (1 - e_i)^2 \,, \tag{19.5}$$

where na denotes the number of mesh edges and $e_i = l_i$ if $l_i \leq 1$, $e_i = 1/l_i$ if $l_i > 1$.

This measure enables a rapid estimation of the conformity of a mesh with respect to a given (isotropic or anisotropic) size map. In practice, a value $\tau \geq 0.91$ indicates that the mesh respects the specification well. The reader can refer to Table 18.1 (Chapter 18) to appreciate the sensitivity of the index in the isotropic case.

Quality of a triangle (in two dimensions). If the metric map is isotropic, we retrieve *naturally* Relation (19.1). When the metric map is anisotropic, the quality of a triangle, in two dimensions, is then defined by the relation :

$$\mathcal{Q}_K = \max_{1 \leq i \leq 3} \mathcal{Q}_K^i \,, \tag{19.6}$$

where $\mathcal{Q}_K^i$ represents the quality of triangle K measured in the Euclidean space related to vertex P_i of K. If $(\mathcal{M}_i)_{1 \leq i \leq 3}$ is the metric specified at the vertices P_i of K, we can also write (see the proof in Chapter 18) :

$$\mathcal{Q}_K^i = \alpha \frac{\max\limits_{1 \leq j < k \leq 3} \sqrt{{}^t\overrightarrow{P_jP_k}\mathcal{M}_i\overrightarrow{P_jP_k}} \displaystyle\sum_{1 \leq j < k \leq 3} \sqrt{{}^t\overrightarrow{P_jP_k}\mathcal{M}_i\overrightarrow{P_jP_k}}}{|\sqrt{Det(\mathcal{M}_i)}|Det(\overrightarrow{P_1P_2}, \overrightarrow{P_1P_3})} \,, \tag{19.7}$$

where α is a coefficient of normalization.

This quality measure, defined in the plane (*i.e.*, in two dimensions), can be extended to surface triangles. The metric $\mathcal{M}_i$ at a vertex P_i is then defined in the tangent plane $\Pi(P_i)$ associated with this vertex and the expression of $\mathcal{Q}_K^i$ is modified accordingly [Frey,Borouchaki-1999].

Quality of a surface triangle. Let us consider the vertex P of triangle K. Let $\vec{n}_K$ be the unit normal to the plane of triangle K and let $\vec{n}(P)$ be the unit normal to the surface (at the tangent plane $\Pi(P)$) at P. We note by θ the angle between the vectors $\vec{n}_K$ and $\vec{n}(P)$ and we denote $\tilde{K}$ the image of triangle K by a rotation of angle θ around the axis defined by the vector $\vec{n}_K \wedge \vec{n}(P)$ (cf. Figure 19.1).

By construction, the triangle $\tilde{K}$ belongs to the tangent plane $\Pi(P)$. Its quality $\mathcal{Q}_K$ can then be measured with respect to the metric $\mathcal{M}(P)$ defined in the tangent plane $\Pi(P)$ at vertex P, using Relation (19.7). This comes down to defining the quality $\mathcal{Q}_K$ of a surface triangle as :

$$\mathcal{Q}_K = \max_{P_i \in K} \mathcal{Q}_{\tilde{K}_i}^i \,. \tag{19.8}$$

By extension, the quality $\mathcal{Q}_\mathcal{T}$ of a surface mesh $\mathcal{T}$ and its average quality $\overline{\mathcal{Q}_\mathcal{T}}$ are defined, as in the classical case, by the following relations :

$$\mathcal{Q}_\mathcal{T} = \max_{K \in \mathcal{T}} \mathcal{Q}_K \,, \qquad \overline{\mathcal{Q}_\mathcal{T}} = \frac{1}{ne} \sum_{i=1}^{ne} \mathcal{Q}_{K_i} \,. \tag{19.9}$$

Quality of a surface quadrilateral. We have seen (Chapter 18) that in two dimensions, the shape quality $\mathcal{Q}_K$ of a quadrilateral K can be evaluated, in the isotropic case, using Formula (18.10) :

$$\mathcal{Q}_K = \alpha \frac{h_{max}\, h_s}{S_{min}} \,, \tag{19.10}$$

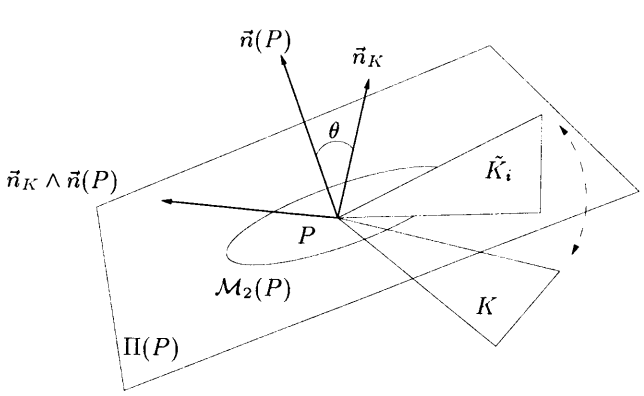


Figure 19.1: *Evaluation of the quality of a surface triangle in the tangent plane $\Pi(P)$ associated with the vertex P. The metric $\mathcal{M}_2(P)$ represents the metric of the tangent plane at P.*

where α is a normalization factor($\alpha = \frac{\sqrt{2}}{8}$), S_{min} is the minimum among the four surfaces that can be associated with K, $h_s = \sqrt{\sum_{i=1}^{4} h_i^2}$ with h_i the length of edge i of K and h_{max} the largest length among the four edges and the two diagonals.

In the general case (when a metric map has been supplied), an optimal quadrilateral is a quadrilateral having unit length edges and diagonals of lengths $\sqrt{2}$.

We can also use the notion of *roughness* of a quadrilateral. To this end, we consider the two diagonals (AC and BD) of the quadrilateral $ABCD$ and we measure the two dihedral angles so defined. Then, the regularity of $ABCD$ is defined as :

$$\mathcal{P}_{ABCD} = \min_{A,B,C,D}(\mathcal{P}_{AC}, \mathcal{P}_{BD}) \,, \tag{19.11}$$

where

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

represent the values of *planarity* of the edges present in the construction (see below the geometric criteria). In these relations, $\vec{n}_K$ denotes the unit normal to the considered triangle K.

In practice, the measure of the regularity of a quadrilateral face will be used to quantify the *torsion* of a tridimensional element.

19.1.3 Quality of the geometric approximation

The quality measures introduced in the previous paragraphs translate numerically the deformation of a triangle and/or its conformity with respect to the size (metric) map specified. However, these measures do not allow[2] us to evaluate the quality

[2] Or only in a indirect way.

of the geometric approximation (of the discretization), which is the way the mesh reflects the geometry of the surface. In particular, it is important to make sure that the gap between the elements and the surface is locally controlled. Recall indeed that (Chapter 15) :

Definition 19.1 *A* geometric mesh *(of type P^1) within a given ε of a surface surface Σ is a piecewise linear discretization of this surface for which the relative gap to Γ is at any point of the order of ε.*

In other words, the problem is then to make sure that a given surface mesh is a geometric mesh for a fixed relative gap ε. To this end, we introduce several measures to evaluate the quality of the geometric approximation of a surface.

Geometric criteria. Several rather simple geometric criteria (almost intuitive) can be used to characterize a surface mesh. These measures are normalized (*i.e.*, range between 0 and 1) according to the usual principle that a value close to 1 indicates that the element (and by extension, the mesh) under consideration is satisfactory with respect to this criterion. These geometric criteria thus behave like quality measures rather than like degradation (deformation) measures of the mesh elements.

We will now specify several criteria such as *planarity, deviation* and *roughness* of a surface. These criteria are local measures of the behavior of the surface in the vicinity of a mesh vertex [Frey,Borouchaki-1998].

- Planarity

The geometric discontinuities of a surface are generally expressed by a rapid variation of the directions of the normals to the surface in the neighborhood of a point (or between two adjacent triangles). The ridges and singularities are characteristic examples of C^0 continuity. However, when the surface is supposed to be G^1 continuous in the neighborhood of a point, a rapid variation of the normal to the surface in this neighborhood is most likely an indication that the mesh density in this area is not able to capture the local variations of the surface. To evaluate this lack of density, we introduce the following definition :

Definition 19.2 *The* planarity $\mathcal{P}_P$ *at point P is defined as the maximal angle between the normal $\vec{n}_P$ to the surface at P and the normals $\vec{n}_{P_i}$ at the vertices P_i of the ball of P, other than P :*

$$\mathcal{P}_P = \frac{1}{2}(1 + \min_{P_i}\langle \vec{n}_P, \vec{n}_{P_i}\rangle) . \tag{19.12}$$

According to this principle, we can define the planarity $\mathcal{P}_{AB}$ of an edge AB as follows :

$$\mathcal{P}_{AB} = \frac{1}{2}(1 + \langle \vec{n}_{K_1}, \vec{n}_{K_2}\rangle) ,$$

where K_1 and K_2 are two triangles sharing the edge AB. Hence, the planarity of an edge is the measure of the dihedral angle between two triangles characterizing the geometric continuity of the surface along AB.

Remark 19.2 *When the planarity value $\mathcal{P}_P$ at P and the minimum of the planarity values $\mathcal{P}_{PP_i}$ at the edges PP_i incident to P are slightly different, the point P is a* singular point.

- Roughness

Considering the set of edges PP_i incident to a point P of the surface, we can define, from the previous measure, the local *roughness* $\mathcal{S}_P$ of the surface at point P as :

$$\mathcal{S}_P = \min_{P_i} \mathcal{P}_{PP_i} . \tag{19.13}$$

The degree of roughness[3] of the surface in the neighborhood of point P thus represents the minimal value of the planarity values over the set of edges incident to P.

- Deviation

A variant of the planarity measure at point P consists in evaluating the deviation of the mesh edges with respect to the geometry (*i.e.*, the surface). In other words, we attempt here to evaluate the maximal gap between the edges and the tangent plane $\Pi(P)$ at P. Thus, we suggest the following definition :

Definition 19.3 *The* deviation $\mathcal{D}_P$ *at point P corresponds to the maximal angle between the edges PP_i incident to P and the tangent plane $\Pi(P)$, calculated as follows :*

$$\mathcal{D}_P = 1 - \min_i |\langle \vec{n}_P, \vec{u}_i\rangle| , \tag{19.14}$$

where $\vec{u}_i$ represents the unit vector supported by the line PP_i and $\vec{n}_P$ is the unit normal vector to the surface at P.

Remark 19.3 *From a practical point of view, the deviation criterion is less accurate than the planarity. Indeed, consider a point P for which the three following relations are satisfied :*

$$\langle \vec{n}_P, \vec{u}_i\rangle = 0 , \qquad \langle \vec{n}_{P_i}, \vec{u}_i\rangle = 0 , \qquad \langle \vec{n}_P, \vec{n}_{P_i}\rangle = -1 .$$

At point P, the deviation of the edges with respect to the tangent plane is judged to be good, while the planarity criterion indicates that the surface is locally badly discretized in the neighborhood of P (Figure 19.2). Such a point P is then considered as a singular point (for which the normal to the surface is not defined).

[3] Notice that the notion of roughness of a piecewise linear interpolation surface (or a Cartesian surface) has been defined as the L^2 norm squared of the gradient of the function f defining the surface, integrated over the triangulation $\mathcal{T}$ [Rippa-1990] :

$$|f|^2_{\mathcal{T}} = \sum_{i=1}^{n} |f|^2_{K_i} , \qquad \text{with} \quad |f|^2_{K_i} = \int_{K_i} \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 dxdy ,$$

where K_i denotes a triangle of $\mathcal{T}$ and n is the number of elements of $\mathcal{T}$.

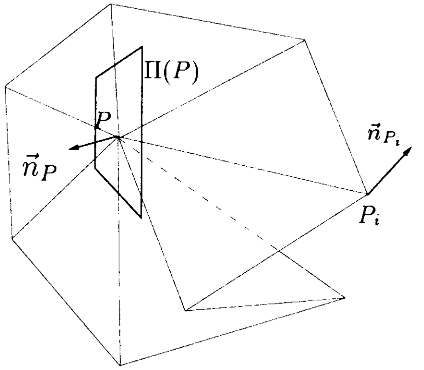


Figure 19.2: *Example for which the variation of the unit normals between two neighboring vertices is better captured by the planarity measure than by the measure of the local deviation at P.*

Notice that these criteria do not involve the map of the geometric metrics which may possibly be supplied. In other words, we consider here isotropic criteria. In the anisotropic case, the surface mesh is geometric if it complies with a geometric metric map, for which the edge lengths are locally proportional to the principal radii of curvature. To verify that a mesh complies with such a map, we introduce the following criterion.

Size quality. Suppose given a (geometric) metric map $\mathcal{M}$, the size quality $\mathcal{L}_{AB}$ of a mesh edge AB is defined by the relation :

$$\mathcal{L}_{AB} = \begin{cases} l_{AB} & \text{if} \quad l_{AB} \leq 1 \\ \dfrac{1}{l_{AB}} & \text{else} \end{cases} , \tag{19.15}$$

where l_{AB} represents the length of edge AB in the metric $\mathcal{M}$ specified (Chapter 10). That is, if edge AB is parameterized by $t \in [0,1]$:

$$l_{AB} = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\,\mathcal{M}(A + t\overrightarrow{AB})\,\overrightarrow{AB}}dt\,, \tag{19.16}$$

and, when the metric is position independent :

$$l_{AB} = \sqrt{{}^t\overrightarrow{AB}\,\mathcal{M}\,\overrightarrow{AB}}\,.$$

By extension, the size quality can be defined at a vertex P, depending on the edge PP_i incident to P, in the following manner :

$$\mathcal{L}_P = \min_{P_i} \mathcal{L}_{PP_i}\,. \tag{19.17}$$

Combined criterion. The previous criteria can be combined into a single weighted criterion $\mathcal{C}_P$ defined at a vertex P, for example in the following way :

$$\mathcal{C}_P = \mathcal{P}_P^{\alpha_1}\,\mathcal{D}_P^{\alpha_2}\,\mathcal{S}_P^{\alpha_3}\,\mathcal{L}_P^{\alpha_4}\,, \tag{19.18}$$

where the coefficients α_i are such that :

$$\sum_{\alpha_i=1}^{4} \alpha_i = 1\,.$$

Remark 19.4 *The context of the application makes it possible to specify the values of these coefficients, depending on their desired relative weights.*

A global measure and an average measure at the mesh level can also be defined as :

$$\mathcal{C}_\mathcal{T} = \min_{P\in\mathcal{T}} \mathcal{C}_P \qquad \text{et} \qquad \overline{\mathcal{C}}_\mathcal{T} = \frac{1}{n}p \sum_{P\in\mathcal{T}} \mathcal{C}_P\,, \tag{19.19}$$

where np is the number of vertices in the mesh $\mathcal{T}$.

We now have a set of criteria allowing us to analyze whether a given mesh is a geometric mesh or not. We will see later that these criteria can also be used to control the mesh modification and mesh optimization operators.

19.1.4 Optimal surface mesh

We have already mentioned that, given $\mathcal{Q}_K$ a quality measure for an element K of a mesh $\mathcal{T}$, the global mesh quality is defined as :

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

The notion of an optimal mesh theoretically refers to a set of meshes. As for planar or volumic meshes, we could say that the *optimal surface mesh* is that for which the measure considered is optimal. Formally speaking, the optimal surface mesh is that which, simultaneously :

- optimizes the geometric criterion (or the quality function) considered,
- minimizes the number of elements (vertices).

In practice, we are trying to comply as well as possible with the various geometric criteria, the quality measures and the need to minimize the number of elements. As in two dimensions, a surface mesh composed exclusively of triangles must have edge lengths close to 1. Here we again encounter the notion of unit mesh (Chapter 18) :

Definition 19.4 *A* unit mesh *is a mesh in which the elements edges are of unit length.*

Given this definition, we can say that :

Definition 19.5 *A surface mesh composed exclusively of triangles is optimal if it is a unit mesh with respect to the geometric metric map (*i.e., *proportional to the principal radii of curvature).*

A good way of evaluating the optimality of a surface mesh consists in using the efficiency index previously introduced.

Remark 19.5 *For quadrilaterals, we try to have unit edge lengths and diagonals of length* $\sqrt{2}$.

19.2 Discrete evaluation of surface properties

In the previous section, we have seen that the geometric criteria and the other quality measures involve quantities such as face normals or normals to the surface at mesh vertices. Moreover, we know that a geometric mesh is unit mesh according to the geometric metric map (*i.e.*, the map proportional to the principal radii of curvature). If the mesh is the unique data of the problem, the intrinsic properties of the surface (normals, tangent planes, radii of curvature, etc.) are not known explicitly.

In this section, we first show how to find these properties in an approximate (discrete) way. Then, we indicate how to establish the metric of the tangent plane. Finally, several practical aspects (related to the data structure) are mentioned.

19.2.1 Intrinsic properties

Normal and tangent plane. The tangent plane $\Pi(P)$ at a regular point P of the surface is defined from the unit normal at P. If the unit normal vector $\vec{n}(P)$ is not known, it can be approached as the average (possibly weighted) values of the normals $\vec{n}_{K_i}$ at the triangles incident to P (*i.e.*, the triangles of the ball of P) :

$$\vec{n}(P) = \frac{\sum\limits_{K_i} \omega_i \, \vec{n}_{K_i}}{\|\sum\limits_{K_i} \omega_i \, \vec{n}_{K_i}\|} \,. \qquad (19.20)$$

In practice, the weights ω_i associated with the normals at the triangles can be related to various (representative) quantities of the mesh, for example :

- the surfaces $S(K_i)$ of the triangles,
- the inverse of the surfaces $S(K_i)$ of the triangles (to emphasize the smallest elements),
- the angle $\alpha_i(P)$ at P of each triangle K_i,
- ...

The unit normal vector $\vec{n}(P)$ at P serves to define the tangent plane. In fact, the point P belongs to $\Pi(P)$ and the vector $\vec{n}(P)$ is a vector orthogonal to $\Pi(P)$ (Chapter 11 and Figure 19.3).

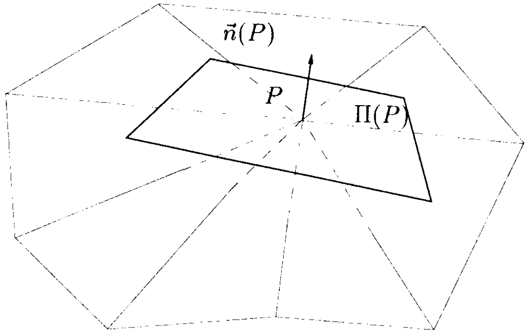


Figure 19.3: *Unit normal vector at P and associated tangent plane.*

Remark 19.6 *When the point P is a singular point, the tangent plane is not defined. If P is along a ridge, we can define a normal on each side of the edge. To this end, we consider the two open balls at P (limited by the ridge) and we calculate a unit normal vector using Formula (19.20).*

Remark 19.7 *Recall that for parametric surfaces (Chapter 11), the tangent plane is directed by the two tangent vectors* $\vec{\tau}_1(P)$ *and* $\vec{\tau}_2(P)$ *:*

$$\vec{n}(P) = \frac{\vec{\tau}_1(P) \wedge \vec{\tau}_2(P)}{\|\vec{\tau}_1(P) \wedge \vec{\tau}_2(P)\|} \,.$$

We will now examine how to determine the curvatures and the principal directions of curvature at the vertices of a given mesh.

Principal curvatures (summary). At a point P, which is supposed regular, of the mesh, the normal $\vec{n}(P)$ is calculated using Formula (19.20). Let us consider the ball $\mathcal{B}(P)$ of point P. The edges PP_i, for each $P_i \in \mathcal{B}(P)$, are supposed traced on the surface.

Given a tangent vector $\vec{\tau}(P) \in \Pi(P)$ to the surface at P, there exists a curve Γ traced on the surface admitting $\vec{\tau}(P)$ as tangent. Let $C(P)$ be the curvature of Γ at P, we have seen that the normal curvature[4] $\kappa_n(\vec{\tau}(P))$ of Γ at P is defined as (Chapter 11) :

$$\kappa_n(\vec{\tau}(P)) = C(P) \cos \alpha \,, \qquad (19.21)$$

where α is the angle between $\vec{n}(P)$ and $\vec{\nu}(P)$: $\cos \alpha = \langle \vec{\nu}(P), \vec{n}(P) \rangle$, $\vec{\nu}(P)$ being the unit normal vector to Γ at P (Figure 19.4).

To find the principal curvatures, we use Meusnier's theorem. We know that all curves traced on the surface and having the same tangent vector at P have the same curvature $\kappa(P) = |\kappa(\vec{\tau}(P))|$ at P. We consider then a particular curve, the normal section (corresponding to the intersection of a plane defined by the vectors $\vec{\tau}(P)$ and $\vec{n}(P)$). For such a curve, the vectors $\vec{n}(P)$ and $\vec{\nu}(P)$ are collinear. Recall indeed that if Γ is the intersection of the surface Σ by a normal section, the

[4] Notice that the sign of $C(P)$ changes with the orientation of $\vec{n}(P)$.

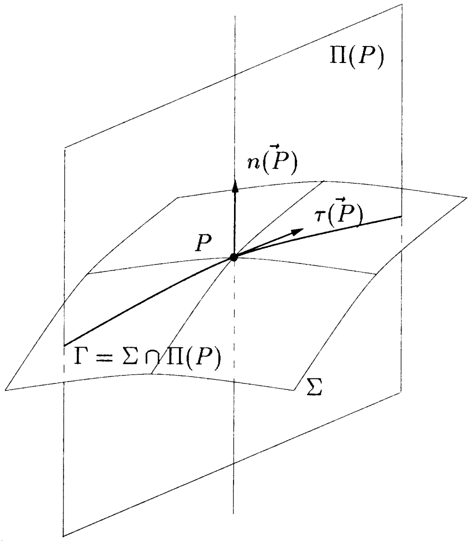


Figure 19.4: *A normal section at P and the curve Γ traced on the surface of vector director $\vec{\tau}(P)$.*

Meusnier's circle of diameter κ_n is the geometric locus of the points P_i endpoints of the segments PP_i such that $||\overrightarrow{PP_i}|| = C(P)$, the curvature of a curve whose normal $\vec{\nu}$ forms an angle α with the unit normal $\vec{n}$ to Σ at P.

By definition, an infinity of normal sections exist around P. Among these, let us consider the two sections (orthogonal together) whose normal curvatures are respectively minimal and maximal. These two curvatures are the *principal curvatures* at P, denoted $\kappa_1(P)$ and $\kappa_2(P)$, and the directions associated are the *principal directions*, of unit vectors $\vec{\tau}_1(P)$ and $\vec{\tau}_2(P)$.

Thus, we propose a way of evaluating the mean curvature and the principal curvatures at any vertex of a given mesh. More precisely, we will calculate the minimal radius of curvature and the principal radii of curvature that will serve later to define the geometric metric at any mesh vertex P.

Calculation of the minimal radius of curvature (isotropic case). Let $\mathcal{T}$ be a surface mesh representing locally the geometry of the surface. Let us consider then $\mathcal{B}(P)$ the ball of a vertex P (supposed regular). It seems natural to consider an edge PP_i of $\mathcal{B}(P)$ as the discretization of a curve Γ_i traced on the surface and belonging to a normal section.

Let Γ_i be such a curve, of normal parameterization $\gamma_i(s)$. Using a Taylor expansion at order 2 of $\gamma_i(s)$ at $P = \gamma_i(s_0)$, we can write :

$$\gamma_i(s_0 + \Delta s) = \gamma_i(s_0) + \Delta s \vec{\tau}(s_0) + \Delta s^2 \frac{1}{2\,\rho_i(s_0)} \vec{\nu}(s_0) + \mathcal{O}(\Delta s^3)\,, \qquad (19.22)$$

where Δs represents a small increment of s. The *osculating circle* $\mathcal{C}_i(P)$ to γ_i at P is the circle of radius $\rho_i(s_0)$ and its center O_i is given by :

$$O_i = P + \rho_i(s_0)\,\vec{\nu}(s_0)\,.$$

Consider then the point P_i defined by :

$$P_i = P + \Delta s \vec{\tau}(s_0) + \Delta s^2 \frac{1}{2\,\rho_i(s_0)} \vec{\nu}(s_0)\,,$$

that is the point of the parabola whose tangent is $\vec{\tau}(s_0)$ and which is at a distance Δs from $\gamma(s0)$. This point is thus very close to the osculating circle of the curve (Chapter 14). This allows us to write (Figure 19.5, left-hand side) :

$$\left\langle \overrightarrow{PP_i}\,,\, \frac{1}{2}\overrightarrow{PP_i} - \rho_i^* \vec{\nu}(s_0) \right\rangle = 0\,,$$

considering a circle of radius ρ_i^* passing through P and P_i.

According to the previous remark (relative to the proximity of P_i to the osculating circle), we can then approach $\rho_i(s_0)$ by ρ_i^* (for a sufficiently small value of Δs, refer to the discussion in Chapter 11). The circle of radius ρ_i^* is called the *approximate osculating circle* and we deduce the value of the radius of curvature :

$$\rho_i^*(s_0) = \frac{\langle \overrightarrow{PP_i}, \overrightarrow{PP_i} \rangle}{2\,\langle \vec{\nu}(s_0), \overrightarrow{PP_i} \rangle}\,. \qquad (19.23)$$

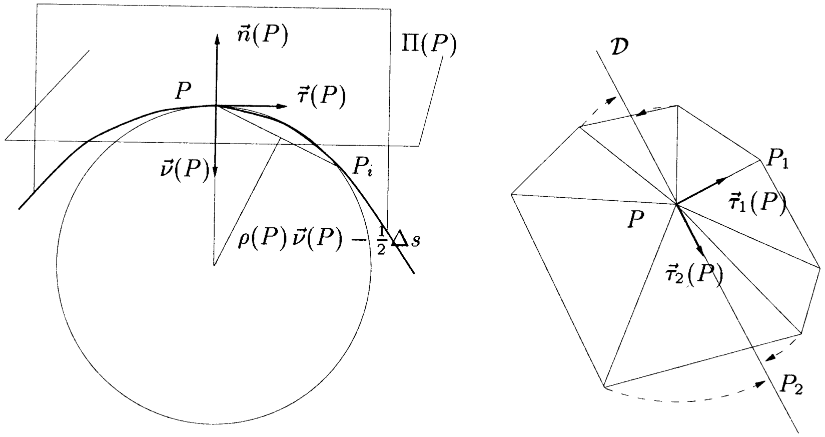


Figure 19.5: *Approximation of the osculating circle at point P (left-hand side). Trivial determination of the principal radii of curvature and the principal directions of curvature (right-hand side).*

In other words, when the point of abscissis $(s_0 + \Delta s)$ is approached by the point P_i, the approximate radius of curvature ρ_i^* is a good (accurate) approximation of the radius of curvature at P.

This result now makes it possible to define the minimal radius of curvature $\rho(P)$ at P by the relation :

$$\rho(P) = \min_{PP_i} \rho_i^* \,. \tag{19.24}$$

This value will allow us to define the isotropic metric (*i.e.*, the size here) at any vertex P of the mesh (see below).

Calculation of the principal radii of curvature ("naive" approach). The calculation of the minimal radius of curvature at a regular mesh vertex P using the previous approach can be slightly modified, so as to find the principal radii of curvature at P.

In fact, reconsider the previous reasoning about the edges PP_i incident to P. It is easy to find the edge PP_1 corresponding to the minimal radius of curvature, $\rho_1(P)$. Hence, the unit vector $\vec{\tau}_1(P)$ supported by the edge PP_j gives the direction of maximal curvature. To determine the maximal radius $\rho_2(P)$, we proceed as follows (Figure 19.5, right-hand side) :

- calculate the line $\mathcal{D}$ orthogonal to $\vec{\tau}_1(P)$,
- position then (in distance) the points P_i of $\mathcal{B}(P)$ (the ball of P) on $\mathcal{D}$,
- identify the point P_2 (on $\mathcal{D}$) such that $\overrightarrow{PP_j}$ is maximal,
- we have $\vec{\tau}_2(P) = \dfrac{\overrightarrow{PP_2}}{\|\overrightarrow{PP_2}\|}$ the unit vector indicating the direction of minimal curvature.

Remark 19.8 *An alternative consists in considering the maximal radius of curvature first and then in applying the above procedure to find the minimal radius of curvature.*

Remark 19.9 *This calculation, although very simple in principle, leads in practice to some problems. The fact of fixing one of the directions to find the other makes this procedure strongly dependent on the given mesh. If the given mesh is, for example, a triangulation obtained by refinement of a regular grid, the directions of curvature can be "shifted" as compared to the true directions (Figure 19.6).*

For this reason, we now indicate a more accurate approach to calculating the principal directions of curvature.

Calculation of the principal radii of curvature (approach 2). Various approaches have been proposed. Let us mention, in particular, one based on a least square approximation formula of *Dupin's indicatrix* suggested, among others, by [Todd,McLeod-1986], [Chen,Schmitt-1992]. However, to be valid, this calculation supposes that the vertices around the vertex P considered can be matched together two by two.

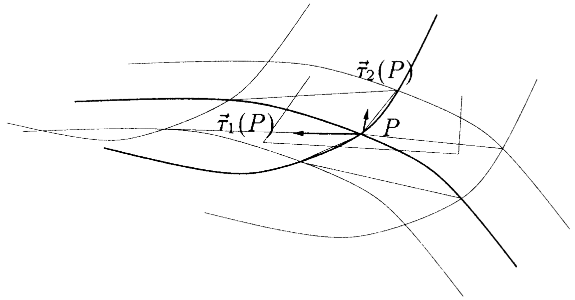


Figure 19.6: *Particular case where the calculation of the principal directions of curvature is distorted by the discretization.*

Dupin's indicatrix of the surface at P, which allows us to approximate locally the surface by a paraboloid, corresponding to a Taylor expansion at the order 2 is defined by the equation :

$$\kappa_1 x^2 + \kappa_2 y^2 = 1\,, \tag{19.25}$$

where $x = \sqrt{\rho}\cos\theta$ and $y = \sqrt{\rho}\sin\theta$ for a point $M(x,y)$ of the tangent plane at P (in polar coordinates, function of ρ and θ). This equation is that of a conic section, each point M in the direction θ is located at a distance $\sqrt{\rho}$ from P. If the point P is an elliptical point, the Dupin's indicatrix is an ellipse, if P is hyperbolic, the indicatrix is formed by two branches of hyperbolas (Chapter 11).

Another possible approach consists in using a least square approximation of the directions of the tangents and the normal curvatures to calculate the principal directions and curvatures. To this end, we calculate for each edge PP_i incident to P a normal curvature, based on the technique previously described. We thus obtain on $\mathcal{B}(P)$ a set of directions (tangents) and of normal curvatures. The normal curvature $\kappa_n(P)$ can be expressed in a quadratic form (Chapter 11). Let $[P, \vec{\tau}_1, \vec{\tau}_2]$ be an orthonormal basis of the tangent plane $\Pi(P)$, any vector $\vec{\tau}$ can be written as a linear combination $\vec{\tau} = \lambda\vec{\tau}_1 + \mu\vec{\tau}_2$. Then, the normal curvature in the direction $\vec{\tau}$ can be written as :

$$\kappa_n(\vec{\tau}) = (\lambda \;\; \mu) \begin{pmatrix} \kappa_n(\vec{\tau}_1) & \kappa_n^{12} \\ \kappa_n^{21} & \kappa_n(\vec{\tau}_2) \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix}, \tag{19.26}$$

where $\kappa_n^{12} = \kappa_n^{21}$. As the matrix is symmetric, the vectors $\vec{\tau}_1$ and $\vec{\tau}_2$ can be chosen so as to diagonalize the matrix : $\kappa_n^{12} = \kappa_n^{21} = 0$.

Let us write the normal curvature as follows :

$$\kappa_n(\vec{\tau}) = (\tau_x \;\; \tau_y)\; {}^t\mathcal{D} \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} \mathcal{D} \begin{pmatrix} \tau_x \\ \tau_y \end{pmatrix},$$

where τ_x and τ_y denote the components of $\vec{\tau}(P)$ in the basis and $\mathcal{D}$ corresponds to the principal directions at P. We can express $\mathcal{D}$ according to only one of the principal directions (the two vectors being orthogonal) :

$$\mathcal{D} = \begin{pmatrix} \tau_{1,x} & \tau_{1,y} \\ -\tau_{1,y} & \tau_{1,x} \end{pmatrix}.$$

Repeating the same process for the m edges PP_i incident to P, we finally obtain a system of m linear equations [Moreton-1992] :

$$\begin{pmatrix} u_{1,x}^2 & u_{1,x}u_{1,y} & u_{1,y}^2 \\ .. & .. & .. \\ u_{m,x}^2 & u_{m.x}u_{m,y} & u_{m,y}^2 \end{pmatrix} \begin{pmatrix} \tau_{1,x}^2\kappa_1 + \tau_{1,y}^2\kappa_2 \\ 2\tau_{1,x}\tau_{1,y}(\kappa_1 - \kappa_2) \\ \tau_{1,x}^2\kappa_2 + \tau_{1,y}^2\kappa_1 \end{pmatrix} = \begin{pmatrix} \kappa_{n,0} \\ .. \\ \kappa_{n,m} \end{pmatrix}, \quad (19.27)$$

where the unknowns are obviously the expressions depending on the principal curvatures and directions. This system is of the form $AX = B$.

Remark 19.10 *Notice however, that this system depends on the mesh (*i.e.*, of the number of edges incident to P). Hence, some of the equations may be redundant (think in particular of the pairs of opposite edges). It is thus important to identify and remove such equations.*

Remark 19.11 *Another particular case corresponds to the configuration where only three edges are incident to P. The system is then fully determined (3 equations and 3 unknowns).*

To resolve such a system (usually over-determined), we proceed as follows. Satisfying all the equations simultaneously is (in principle) not possible, hence we look for the best possible compromise, for example in the sense of the least squares (*i.e.*, the sum of the squares of the distances between the left and the right members is minimal). The general formula to approximate the least square solution consists in obtaining a well determined system (n equations and n unknowns) by multiplying by ${}^tA$ and by introducing $\bar{X}$ the approximate solution of X :

$$({}^tA\,A)\,\bar{X} = {}^tA\,B\,.$$

Once this system has been solved[5] (*i.e.*, the values of $\bar{X}$ are known) and according to the relation $\tau_{1,x}^2 + \tau_{1,y}^2 = 1$, we then have the system of four equations with four unknowns to solve :

$$\left\{ \begin{array}{lcl} \tau_{1,x}^2\kappa_1 + \tau_{1,y}^2\kappa_2 & = & \bar{x}_0 \\ 2\tau_{1,x}\tau_{1,y}(\kappa_1 - \kappa_2) & = & \bar{x}_1 \\ \tau_{1,x}^2\kappa_2 + \tau_{1,y}^2\kappa_1 & = & \bar{x}_2 \\ \tau_{1,x}^2 + \tau_{1,y}^2 & = & 1 \end{array} \right. . \quad (19.28)$$

The expressions of the principal curvatures are :

$$\kappa_1 = \frac{\bar{x}_1 + \bar{x}_3 - \sqrt{(\bar{x}_3 - \bar{x}_1)^2 + \bar{x}_2^2)}}{2} \text{ and } \kappa_2 = \frac{\bar{x}_1 + \bar{x}_3 + \sqrt{(\bar{x}_3 - \bar{x}_1)^2 + \bar{x}_2^2}}{2} \quad (19.29)$$

and the components of the vector $\vec{\tau}_1$ corresponding to the first principal direction are :

$$\tau_{1,x} = \sqrt{\frac{1}{2}\left(\frac{\bar{x}_1 - \bar{x}_3}{\kappa_1 - \kappa_2} + 1\right)} \text{ and } \tau_{1,y} = \sqrt{\frac{1}{2}\left(1 - \frac{\bar{x}_1 - \bar{x}_3}{\kappa_1 - \kappa_2}\right)}, \quad (19.30)$$

the unit vector $\vec{\tau}_2$ being orthogonal to $\vec{\tau}_1$.

[5] using a classical method of linear algebra (see for instance [Golub,VanLoan-1983]).

Exercise 19.1 *Retrieve the two expressions of the principal curvatures and directions from the Relations (19.28).*

An extension of this approach consists in constraining each pair of opposite edges incident to a vertex P to join with a continuity of order G^2. The pairs of edges are G^1 continuous when they share the same tangent vectors. G^2 continuity means that the curves defined by the pairs of opposite edges share the same binormal vector [Shirman,Séquin-1991], [Moreton-1992].

Calculation of the principal curvatures (approach 3). Finally, to conclude this discussion about the calculation of the principal curvatures and directions, we present here a third approach (suggested by [Hamann-1993], among others).

The basic idea is to go back to a problem of parametric surface, for which the calculation of the principal curvatures can be performed analytically (using the fundamental forms, Chapter 11). To this end, we start by searching for an interpolation surface passing through P and approaching at best the endpoints of the edges PP_i incident to P. More precisely, this scheme consists in :

- finding the ball $\mathcal{B}(P)$ of P,
- calculating the projections of the $P_i \in \mathcal{B}(P)\,, p_i \neq P$ on the tangent plane $\Pi(P)$ at P,
- considering the projections as abscissas and the distances $||\overrightarrow{PP_i}||$ as ordinates in a local frame in $\Pi(P)$,
- constructing a polynomial of approximation (a quadric, for example) for these points in the local frame,
- calculating the principal curvatures according to this polynomial.

Remark 19.12 *Notice that with this approach, we also find a (over-determined) system of linear equations (depending on the number of incident edges).*

Let $\sigma(u,v)$ be the polynomial of approximation, the surface of approximation is then defined as the set of points $(u, v, \sigma(u,v)) \in \mathbb{R}^3$. The principal curvatures κ_1 and κ_2 of the polynomial :

$$\sigma(u,v) = \frac{1}{2}(c_{2,0}u^2 + 2c_{1,1}uv + c_{0,2}v^2)\,,$$

at point $(0, 0, \sigma(0,0))$ are given by the roots of the equation :

$$\kappa^2 - (c_{2,0} + c_{0,2})\kappa + c_{2,0}c_{0,2} - c_{1,1}^2 = 0\,. \quad (19.31)$$

Remark 19.13 *Variants of this approach consist in using cubical splines as approximation functions [Wang, Liang-1989] or Beta-splines [Tanaka* et al. *1990], for example, or even more simply in considering the circles circumscribed to the triangles (PP_iP_j) [Chen,Schmitt-1992].*

19.2.2 Metric of the tangent plane

Once the principal curvatures and directions have been calculated using one of the methods described previously, the metric $\mathcal{M}_3$ of the tangent plane can be constructed at each vertex P of the given mesh. We have already seen that the control of the gap between an edge and the surface for a given value ε fixed can be obtained via the metric $\mathcal{M}_3$, the so-called geometric metric (Chapter 15).

As we consider edges as curves traced on the surface, this leads to the matrix $\mathcal{M}_2(P)$, which is the trace of the matrix $\mathcal{M}_3(P)$ on the tangent plane $\Pi(P)$ at P :

$$\mathcal{M}_2(P) = \mathcal{M}_3(P) \cap \Pi(P) .$$

In other words, at any point P, the metric $\mathcal{M}_2(P)$ is the metric *induced* by $\mathcal{M}_3(P)$ on the tangent plane $\Pi(P)$ to the surface at P (Chapter 10).

Remark 19.14 *The metric $\mathcal{M}_2$ is only defined in the tangent planes associated with the vertices.*

It is obvious that the choice of the metric induces the nature of the mesh to obtain. In principle, we are looking for a metric having the general form :

$$\mathcal{M}_3(P) = \begin{pmatrix} a_P & b_P & c_P \\ b_P & d_P & e_P \\ c_P & e_P & f_P \end{pmatrix} . \qquad (19.32)$$

Hence, we consider the following *geometric* metrics (*i.e.*, in the tangent plane) :

- (anisotropic) metric of the principal curvatures (or metric of the principal radii of curvature) :

$$\mathcal{G}_2(P)_{\kappa_1,\kappa_2} = {}^t\mathcal{D}(P) \begin{pmatrix} \dfrac{\kappa_1^2(P)}{\alpha_\varepsilon^2} & 0 \\ 0 & \dfrac{\kappa_2^2(P)}{\beta_\varepsilon^2} \end{pmatrix} \mathcal{D}(P) , \qquad (19.33)$$

  where $\mathcal{D}(P)$ is a matrix corresponding to the principal directions $\vec{\tau}_1$ and $\vec{\tau}_2$ at P, where α_ε (resp. β_ε) is a coefficient allowing an anisotropic (relative) control of the gap ε to the geometry (Chapter 13), that takes into account the principal directions and curvatures;

- (isotropic) metric of the minimal radius of curvature :

$$\mathcal{G}_2(P)_\rho = \begin{pmatrix} \dfrac{1}{h^2(P)} & 0 \\ 0 & \dfrac{1}{h^2(P)} \end{pmatrix} , \qquad (19.34)$$

  where ρ is the minimal radius of curvature and the variable $h(P) = \alpha\rho(P)$ depends on the position and α is an adequate coefficient, related to the geometric approximation (*i.e.*, to the gap between the edges of the discretization and the surface);

- anisotropic (resp. isotropic) *physico-geometric* metric. Here we consider the metric $\mathcal{G}_2(P)_{\kappa_1,\kappa_2}$ (resp. $\mathcal{G}_2(P)_\rho$) intersected by an arbitrary field of metrics (for example, supplied after a calculation).

Exercise 19.2 *Retrieve and justify the value of the coefficients α and β used in Relation (19.33).*

In practice, we could also consider the isotropic map $\mathcal{M}_3(P)$ of the intrinsic sizes.

Definition 19.6 *The* intrinsic size *at any vertex $P \in \mathcal{T}$ corresponds to an average value (possibly weighted) of the Euclidean lengths of the edges PP_i incident to P.*

The average value corresponds to the solution of a minimization problem (between the desired size and the edge length). Actually, the problem is to find $h(P)$ for any vertex P so as to minimize the function :

$$H(P) = \sum_{P_i} (h(P) - \|\overrightarrow{PP_i}\|)^2 ,$$

where the P_i's denote the points of the ball of P but P.

Remark 19.15 *When the size $h(P)$ is, at any point P, smaller than or equal to $\rho(P)$, the metric $\mathcal{M}_3(P)$ so defined is a geometric metric.*

Once these metrics have been defined, the problem arises of evaluating the edge lengths with respect to the geometric metric.

Calculation of the edge lengths in $\mathcal{M}_2$. At a given point P, the length of an edge PP_i can be approximated using Formula (19.16) by considering the point $P^* = PP_i^* \cap \mathcal{M}_2(P)$ where P_i^* is the projection of point P_i in the tangent plane $\Pi(P)$ (Figure 19.7).

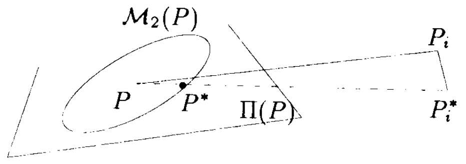


Figure 19.7: *Calculation of the length of edge PP_i in the metric $\mathcal{M}_2(P)$ associated with the tangent plane $\Pi(P)$ at P.*

From these results, we now have a theoretical framework to evaluate the conformity of the mesh with respect to a given metric $\mathcal{M}_3$.

Mesh conformity. We aim to decide whether a given mesh conforms to a geometric metric specification $\mathcal{G}_3$ defined at any mesh vertex.

Definition 19.7 *A mesh $\mathcal{T}$ conforms to a given metric map $\mathcal{M}_3$ (or more precisely to its restriction $\mathcal{M}_2$ in the tangent planes) if and only if :*

$$\frac{1}{\sqrt{2}} \leq l_{AB} \leq \sqrt{2}, \quad \forall AB \in \mathcal{T}. \tag{19.35}$$

When the metric map $\mathcal{M}_2$ considered is the map $\mathcal{G}_{\kappa_1,\kappa_2}$ (resp. $\mathcal{G}_\rho$), the mesh is an anisotropic (resp. isotropic) geometric mesh. Such a mesh is called a $\mathcal{G}$-mesh.

Remark 19.16 *When the metric map $\mathcal{G}_2$ is isotropic, the following assertions are equivalent :*

$$\mathcal{T} \text{ conforms to } \mathcal{G}_2 \iff \mathcal{T} \text{ conforms to } \mathcal{G}_3 .$$

Combination of metrics. When a metric map other than that corresponding to the geometric metrics (intrinsic map) is supplied, it is interesting to combine these two maps. To this end, we consider the metric map corresponding to the intersection of metrics $\mathcal{H}_2(P) = \mathcal{G}_2(P) \cap \mathcal{M}_2(P)$ in the tangent planes associated with the mesh vertices (Chapter 10).

Remark 19.17 *By definition, the map $\mathcal{H}_2$ is a geometric map.*

19.2.3 Computational aspects

We will close this section by noticing briefly several practical aspects related to the evaluation of the intrinsic properties of surface meshes.

Data structure. The approaches described above involve the ball of a mesh vertex P. It is thus important to use an efficient internal data structure that allows us to retrieve the elements of the ball of a vertex easily.

For a triangular surface mesh, it is convenient to use a topological data structure such as those introduced in Chapter 2. The triangles are considered as oriented triples[6] to which are associated (at most) three edge neighbors. We could also use a richer representation, using a connection matrix for each triangle.

The case of mixed meshes (composed of triangles and/or quadrilaterals) requires more care in the definition and the management of the data structure.

Identification of singular points. For meshes representing real surfaces (*i.e.*, the boundaries of a real domain), we need to carefully identify the singularities (corners, ridges, etc.) of the model. If the points of C^1 or G^1 discontinuity are not explicitly provided (for instance supplied by a geometric modeling system), we must carry out a pre-processing stage on the data to extract this information.

[6] This is not a restriction, the surfaces concerned being supposed orientable.

Hence, for example, a *ridge*[7] could be identified as a mesh edge such that the dihedral angle between the adjacent faces along the edge is larger than a given threshold. A *corner* is a mesh vertex where three (or more) ridges are incident or a vertex such that two incident ridges form a very acute angle.

In addition to the singular points, a certain number of entities (points, edges, faces) can be imposed, the *required* or *constrained* entities. These entities must be present in the resulting mesh (refer to [George,Borouchaki-1997] for a description of a data structure allowing this type of entity to be specified).

19.3 Constructing a geometric support

When the sole data available is a surface mesh, the construction of a geometric support makes it possible to define, internally to the procedure, a geometry. The latter is actually a mathematical representation of a surface, the given discretization being then supposed to be an approximation (at the second order) of it. This support must, at least, interpolate the vertices and the normals at the mesh vertices[8], so as to satisfy the conditions of continuity of order G^1, except at the points of discontinuity (which are supposed explicitly known).

The problem is then to construct (to invent, so to speak) a surface composed of order G^1, from the given surface mesh, each triangle serving to define a patch. To this end, two adjacent patches must necessarily have the same tangent plane along their common edge, if the latter is not a singularity of the surface (*i.e.*, is not a ridge).

19.3.1 Classical approaches

Several approaches have been proposed to construct a geometric support globally of class G^1 from a piecewise triangular representation of a surface. In principle, in all these approaches, each triangle serves as support of a patch (cf. Chapter 13).

The methods suggested by [Farin-1986] and [Piper-1987] seem well-suited to dealing with the problem of defining a geometric support. They consist in subdividing each triangle into three triangles and in defining on each newly created triangle a polynomial patch of degree 4 (quartic), such that the continuity of the transverse tangent planes along each boundary edge is ensured, on the one hand between the new triangles resulting from the subdivision of the original triangle and, on the other hand, between the couples of new triangles issued from the subdivision of two adjacent triangles.

This technique, however, requires rather large memory resources. Actually, after subdivision, 28 control points (9 of them common to the adjacent triangles) are associated with each initial mesh triangle. Moreover, each triangle leads to the definition of three patches. The definition of these patches greatly depends on the shape quality of the triangle support (thus, if the latter is badly shaped, for

[7] Also called *crest line.*

[8] *A priori*, as the initial mesh is a sufficiently accurate geometric approximation of the surface, this geometric support "emulates" reasonably the role of the geometric modeling system

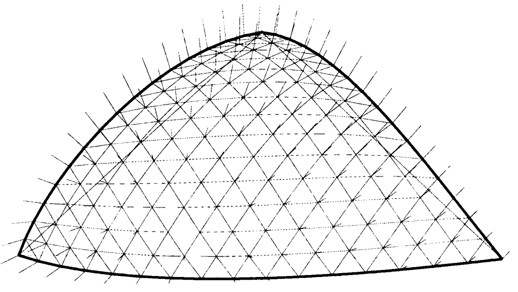


Figure 19.8: *Walton's patch of order G^1 associated with a surface triangle.*

example too stretched, the current method presents some instability in the patch definition).

19.3.2 Modified approach, Walton's patch

More recently, the method proposed by Walton and Meek [Walton,Meek-1996], presents the advantage, on the one hand, of explicitly taking into account the geometric specifications (notably the normals to the surface at the vertices) and, on the other hand, of being relatively simple in its formulation and less memory consuming than the previous approaches.

Basically, this method consists in defining a network of boundary curves for the patches, as well as the related transverse tangent planes, independently of one another, using an interpolation of the normals to the surface at the vertices. Each patch is then defined independently, from its boundary, by taking the specifications related to its boundary (the tangent planes) into account using Gregory's approach [Gregory-1974] (Chapter 13).

The specificity of this method lies in the definition of the network of curves from the sole data of the normals at the vertices (each curve and the related transverse tangent plane is completely defined from the normals at its endpoints). Each boundary curve represents a cubic polynomial whose principal normals (to the curve) are coincident with the normals to the surface at the endpoints. The transverse tangent planes (to the boundary curves) are generated from the tangent vectors at any point along these curves and from vectors resulting from a quadric interpolation of the binormals at the endpoints. Hence, the sole specification of the normals at the two endpoints of an edge is sufficient to define a boundary curve of a patch (based on the endpoints) as well as the transverse tangent plane to the surface along the curve. Finally, from the transverse tangent planes of any triangle, a Gregory patch is generated using a classical approach. Each triangle is thus associated with a patch defined analytically using a rational function in the interior and a quartic along its boundary. This method requires, for each triangle, only 9 control points common to the adjacent triangles.

Remark 19.18 *The scheme suggested by Walton-Meek can be extended to the case of C^0 discontinuities present in "realistic" geometries (ridges, corners, etc.).*

19.3.3 Constructing the geometric support

Here we briefly recall the construction principle for a surface composed of patches globally of class G^1 from an initial (geometric) surface mesh.

The construction scheme comprises the following stages :

- associate with each mesh edge AB a curved segment of degree 3 passing through A and B, while making sure that the principal normals in A and B are collinear to the unit normals to the surface in A and B,
- define the tangent plane generated by the vectors tangent to the curve and to the binormal and associated with the boundary curve,
- raise the degree of the boundary curves so as to construct a polynomial surface of degree 4 (Gregory) over each triangle.

The thus defined surface is of order G^1. This is related to the unique (unambiguous) definition of the tangents (from the normals to the edges), so as to ensure the desired transition between the patches.

Remark 19.19 *For domains presenting C^0 discontinuities, the geometric support is constructed as previously, except at the entities of discontinuity where the tangents are defined in order to obtain a C^0 continuous surface.*

19.3.4 Using the geometric support

As mentioned previously and to conclude on this topic, let us recall that the geometric support is used to answer the following queries :

- given a point and a direction, find the closest surface point,
- return the normal and the minimal radius of curvature at a surface point.

This support thus represents an analytical definition of the surface comparable to that generated by a modeling system (C.A.D.). This support enables us to know the position of the closest surface point from a given point and a specified direction. A second essential requisite concerns the geometric specifications of the surface in the vicinity of these points, a fundamental requirement for any local topological mesh modification (see below). This query provides two types of information, depending on the surface characteristics in the neighborhood of the point considered.

If the point presents (at least) a continuity of order C^2 (in practice, a continuity of order G^1, a tangent plane continuity, is sufficient), the normal to the surface and the minimal of the principal radii of curvature at this point is the sole information required. If the point presents a discontinuity of the tangent planes (the case of points located on a ridge), the two normals to the surface as well as the two minimals of the principal radii of curvature on both sides of the ridge and the tangent to this edge at this point are supplied. This information is returned by the modeling system together with the location of the point considered.

19.4 Optimization operators

As we have seen in Chapter 18, numerous specific tools have been developed for mesh optimization. Similarly, for surface meshes, these operators can be classified into two categories, either of topological or geometric nature. The first ones preserve the mesh connectivity but modify the positions of the mesh vertices and are thus, in some way, related to the geometry of the mesh (a point should be located on the surface). The main operation is the node relation. The second type affect the mesh topology, that is the mesh connectivity (*i.e.*, the connections between vertices, cf. Chapter 1) and are thus related to the geometry in a different way (for instance, an edge flip must preserve the quality of the geometric approximation). Among these operators, we can mention the edge swap, the entity (vertex or edge) deletion, for example.

In the following sections, we will describe these operators in detail, in a particular case of surface meshes.

19.4.1 Geometric optimization

The basic principle of a geometric optimization operator consists in moving the vertices (by modifying their coordinates) while making sure that these vertices remain (to a certain extent that will be specified later) on the surface.

Node relocation algorithms are usually based on the local notion of vertex ball (the union of all elements sharing a vertex P). The ball of a vertex can be *closed*, for an internal vertex, or *open*, for a boundary vertex. For the sake of convenience, we will only study here the case of closed balls.

Node relocation. A trivial node relocation process consists in moving the vertex P under consideration to the barycenter of the positions of the n vertices P_i of the ball of P. This can be formally written (Chapter 18) :

$$P = \frac{1}{n}\sum_{i=1}^{n} P_i\,. \tag{19.36}$$

Notice however that after this modification, the new position of the vertex P is no longer on the surface. Hence, the geometric support must be used to move the point back to the surface, that is to modify its coordinates again . A variation consists in weighting the previous barycentrage, Relation (18.20).

As for two and three dimensional meshes, a more efficient technique consists in introducing a relaxation. The role of the relaxation is to avoid moving the point too far from its original position, which is considered correct (geometrically speaking) during the iterations. Hence, we consider the point P^* defined as the P in Relation (19.36) :

$$P^* = \frac{1}{n}\sum_{i=1}^{n} P_i\,,$$

and the relaxation, of parameter ω, then consists in calculating (as in Chapter 18) the new position of the point P as :

$$P = (1-\omega)P + \omega P^*\,. \tag{19.37}$$

In practice, we can choose a sub-relaxation, for example by taking a value $\omega = 0.5$.

Smoothing techniques. As for two and three dimensional meshes, smoothing techniques can be applied to surface meshes. We indicate here two particular techniques, based on element quality or edge lengths.

- Smoothing based on the quality.

In this approach, we consider the ball of a given vertex P. With each external edge f_i of the ball (*i.e.*, each edge not connected to P) is associated an ideal point P_i^* such that the element formed by f_i and P_i^* has an optimal shape quality. The smoothing process is then defined as :

$$P^* = \frac{\sum\limits_{i=1}^{n} \alpha_i P_i^*}{\sum\limits_{i=1}^{n} \alpha_i}\,, \tag{19.38}$$

where the coefficients α_i can be chosen in several ways (cf. Chapter 18).

Remark 19.20 *This method is also applicable in the case of anisotropic meshes, provided the corresponding notion of quality is used.*

- Smoothing based on edge lengths.

The idea consists this time in defining the position of the optimal points P_i^* of the ball of P, so as to define internal edges of unit length. The unit length is related to the size (or metric) map specified. Hence, we obtain a relation like :

$$P_i^* = P_i + \frac{\overrightarrow{P_iP}}{l_{\mathcal{M}}(P_i,P)}\,, \tag{19.39}$$

where $l_{\mathcal{M}}(P_i,P)$ represents the length of edge P_iP in the metric $\mathcal{M}$ associated with P_iP.

Notice that in practice, regardless of the approach taken, we want to move the point P in the associated tangent plane $\Pi(P)$. To this end, we consider the projections f_i' of the external edges f_i of the ball in the tangent plane $\Pi(P)$ and we look for the position of the associated optimal point using any of the above approaches.

Constrained node relocation. Smoothing and node relocation techniques described previously make it possible to find the position of an optimal point (in the tangent plane associated with the point P to be moved). However, it is necessary to validate this position before accepting the move. In particular, the point P^* is accepted if :

- the geometric approximation associated with the ball of the new point is better than that related to the initial configuration. This can be expressed as :
$$\min\langle \vec{n}_{K_j}, \vec{\nu}(P_i)\rangle \leq \min\langle \vec{n}_{K'_j}, \vec{\nu}(P_i^*)\rangle \,,$$
- the shape quality of the elements of the ball of P^*, $\mathcal{B}(P^*)$, is better than that of $\mathcal{B}(P)$,
- the lengths of the internal edges of $\mathcal{B}(P^*)$ conform to the specified metric :
$$\frac{\sqrt{2}}{2} \leq l_{P^*P_i} \leq \sqrt{2}\,,$$

where $\vec{n}_{K_j}$ and $\vec{\nu}(P_i)$ denote respectively the normals to the triangles K_j and to the vertices P_j of the ball of P.

For a given vertex P, the node relocation consists in redefining the elements incident to this vertex. To this end, from each edge on the boundary of this configuration, we define an optimal element, thus a point, that belongs to the plane supporting the element of the configuration containing the edge. Then, we determine the barycenter of all these points to get an ideal point. The node relocation then consists in moving step by step the vertex toward the ideal point (considering the projections on the surface) if the previous criteria are satisfied.

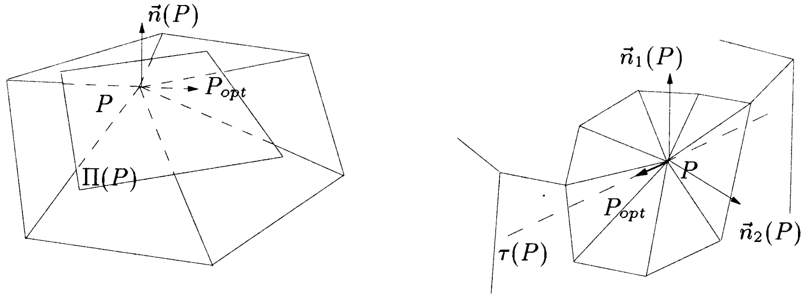


Figure 19.9: *Node relocation for a surface mesh. Left-hand side, node relocation for a G^1 vertex. Right-hand side, node relocation when the vertex is located in a discontinuity of the surface (ridge).*

Remark 19.21 *If the point P considered is located on a ridge (C^0 discontinuity), the relocation procedure is more tedious. Actually, we must consider open balls around P. In fact, the geometric constraints are slightly changed and the checks must be performed in each open ball (Figure 19.9, right-hand side).*

Edge splitting. The subdivision of an edge PQ is an operation that consists in inserting points along the edge, so as to split the edge into unit length segments in the metric specified. We retrieve here the problem of splitting a segment, already discussed in Chapter 14.

In practice, we replace this general procedure by a simpler one consisting in inserting the midpoint M of the edge PQ in the current mesh. More precisely, the midpoint of the edge is created (on the straight segment) and then mapped on the surface using the geometric support. This process involves replacing the two triangles sharing the initial edge by four new triangles sharing two by two the two sub-segments created after the insertion of the midpoint (Figure 19.10).

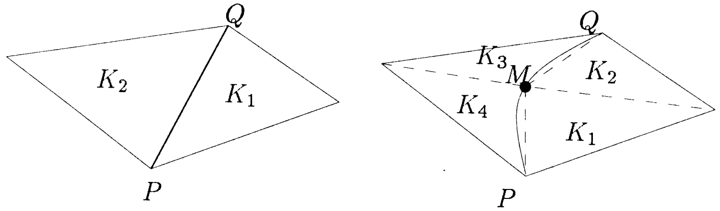


Figure 19.10: *Edge splitting by adding a point on the surface. The two triangles sharing the initial edge (left-hand side) are replaced by four triangles sharing the two newly created sub-segments (right-hand side).*

Specific geometric conditions must be satisfied for the subdivision to be applied :

- the geometric approximation of the new configuration must be better than (or equal to) that of the initial configuration,
- the lengths of the edges AB of the newly created triangles must conform (or at least be close to unit length) to the metric specified :
$$\frac{\sqrt{2}}{2} \leq l_{AB} \leq \sqrt{2}\,,$$
- the shape quality of the final configuration must be improved (as compared with that of the initial configuration).

Remark 19.22 *Notice that, the edge splitting operation is usually combined with edge swapping operations (see below), to form a more complex operator. In this case, the last geometric constraint can be slightly relaxed.*

19.4.2 Topological optimization

We now examine the different operators that make it possible to modify a mesh while preserving the vertex positions.

Edge swapping. Edge swapping is a simple operation consisting in replacing the edge common to two adjacent triangles by the *alternate edge* (*i.e.*, the diagonal linking the vertices not connected by the initial edge). In principle, this operation is similar to edge swapping in two dimensions. However, we will see that additional constraints are applied in the case of surface meshes.

In two dimensions, edge swapping is possible as soon as the polygon formed by the union of the two triangles is convex. For surfaces, we will examine if the configuration of the pair of alternative triangles (after swapping) is *optimal* (in a sense that will be specified) with respect to the geometric approximation and to the element shape quality compared with the original configuration.

Remark 19.23 *Notice than an edge can be swapped if and only if it is not constrained (is not a ridge, for example).*

For surfaces, verifying that the initial configuration is convex is only meaningful if the two triangles are exactly in the same plane (which is a rather particular case). However, according to the shape quality measures introduced previously, we can express several conditions that must be checked and satisfied in order to apply the edge swapping.

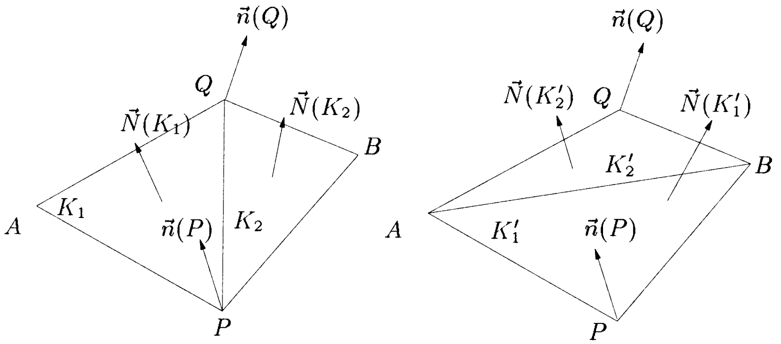


Figure 19.11: *Edge swapping on the surface, initial configuration (left-hand side) and alternative configuration (right-hand side).*

Let $K_1 = APQ$ and $K_2 = BQP$ be two adjacent triangles sharing the edge PQ (Figure 19.11) and let $K_1' = APB$ and $K_2' = BQA$ be the triangles of the alternate configuration. We denote by $\vec{N}(K_i)$ the normal to the triangle K_i and by $\vec{n}(P_i)$ the normal to the vertex P_i. The edge PQ can be changed into the edge AB if and only if :

- the (dihedral) angle between faces K_j' of the alternative configuration does not exceed the angle limit of a ridge (C^0 discontinuity). In other words, the surface must not be *"folded"* locally,
- the geometric approximation (of the underlying surface) of the alternative configuration is better than (or identical to) that of the initial configuration.

This can be expressed by the relation :

$$\min\langle \vec{N}(K_j), \vec{n}(P_i)\rangle \leq \min\langle \vec{N}(K_j'), \vec{n}(P_i)\rangle\,, \quad 1 \leq i \leq 3, 1 \leq j \leq 2\,,$$

- the shape quality of the triangles of the alternative configuration is better than that of the original configuration :

$$\min \mathcal{Q}_{K_j'} < \min \mathcal{Q}_{K_j}\,, \quad 1 \leq j \leq 2\,,$$

- the length l_{AB} of the edge AB conforms to the specified metric :

$$\frac{\sqrt{2}}{2} \leq l_{AB} \leq \sqrt{2}\,,$$

- the edge AB does not already exist in the mesh[9]. We can use a hash table (Chapter 2) to avoid this problem (Figure 19.12, left-hand side). A similar case (Figure 19.12, right-hand side) can be detected using the geometric criteria, the new face after swapping is almost coplanar with two (or more) existing faces.

Here, the order of the checks is arbitrary. In practice, the user must choose an order that favors, as much as possible, a quick reject (a condition not satisfied).

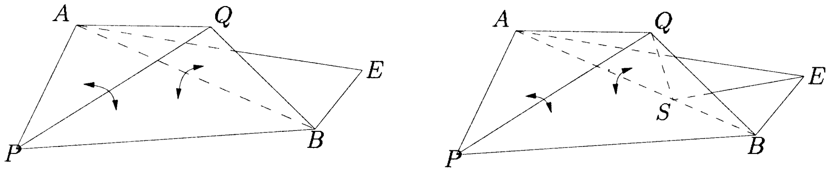


Figure 19.12: *Particular cases where the edge swapping PQ is not possible : the alternative edge AB already exists (left-hand side), the face ABQ and the faces ASQ, BSQ are almost coplanar (right-hand side).*

Edge deletion (vertex removal). This operator consists in retriangulating the ball of the vertex P to be deleted. In two dimensions, this operation comes down to triangulating a star-shaped polygon (with respect to the vertex P), without an internal point. To this end, several methods have been proposed. We suggest here a method that presents the advantage of being applicable to surface meshes. We recall first the principle of the method in two dimensions.

The basic principle of the operation aims at reducing the degree (*i.e.*, the number of incident edges, Chapter 18) of the vertex to be deleted to the value 3 or 4 (in the degenerate case). The cavity can then be retriangulated trivially (Figure 19.13).

The process involves applying to each edge incident to P an edge swapping, if possible (if the orientation of the resulting triangles is preserved). Schematically, the reduction algorithm can be written as follows :

[9] A situation which is impossible with planar or volumic meshes.

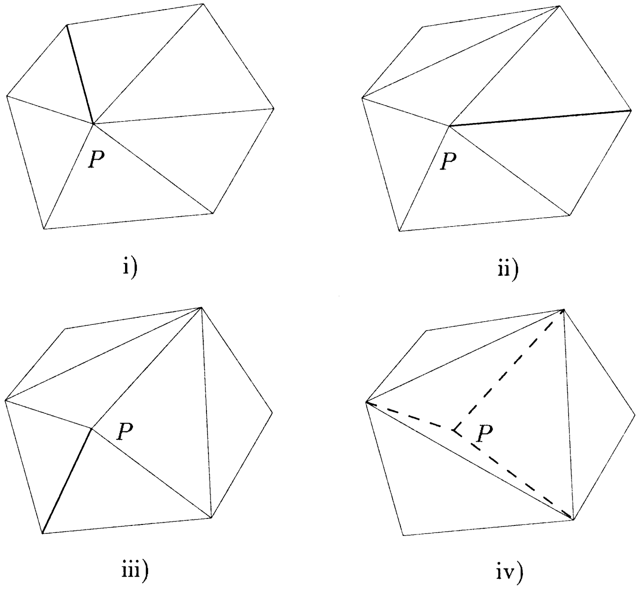


Figure 19.13: *Edge deletion by vertex removal. Edge swappings are applied iteratively (steps i) to iv)) to reduce the degree of the vertex P to 3. The three triangles sharing this vertex can then be removed.*

- apply edge swappings to each edge incident to P,
- iterate the process, if a modification has been carried out.

This process converges in principle toward a configuration of 3 or 4 edges incident to the vertex to be deleted (the second configuration representing a quadrilateral having two orthogonal diagonals).

The application of this algorithm to geometric surface meshes requires a minimum of attention. In two dimensions, the edge swapping is applied if the triangles of the alternative configuration are valid. For surfaces, the alternative configuration must be validated by the conditions described in the previous paragraph. We can however relax a little this constraint by requiring that each newly created triangle of the cavity not enclosing the vertex to be removed (hence a triangle that will be part of the final configuration) must be close to the surface. This is equivalent to imposing that the gap (the angle) between the normal to the triangle and the normals to the surface at the three vertices is bounded by a given tolerance value. We can also impose a condition regarding the shape quality of the triangles formed.

The deletion operator being defined, the convergence of the procedure is no longer ensured. In practice, this means that if an intermediate configuration is blocked, the deletion cannot be carried out and the initial situation must be restored. This leads to extra (useless) work.

For these reasons, we favor another edge deletion operator consisting in merging the two edge endpoints into a single point.

Identification (merging) of the edge endpoints. Given an edge PQ, we try to replace this edge by a single point. To this end, we consider two operators that reduce the edge either into one of its endpoints (cf. Figure 19.14), or into a new point (cf. Figure 19.15).

- Identification on one of the endpoints.

This operation can be seen as a specific re-writing (remeshing) of the ball of the vertex merged. The final configuration being known, it can thus be validated *a priori*, based on the geometric constraints. This is indeed a specific remeshing as all possible configurations are not analyzed systematically to retriangulate the vertex ball (see Chapter 18).

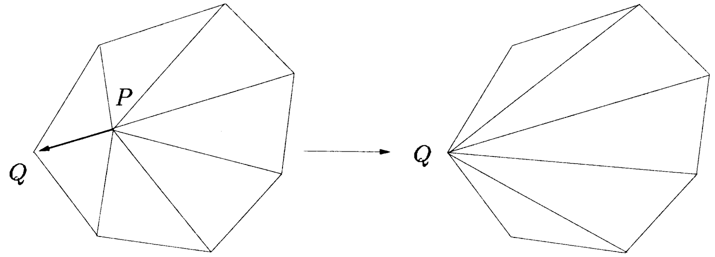


Figure 19.14: *Edge deletion by identifying its two endpoints (the vertex P is merged with the vertex Q).*

In practice, it is sufficient to substitute the vertex Q to the vertex P in all triangles of the ball (without the two triangles sharing the edge PQ) then to verify if the following conditions are satisfied :

- the shape quality of the new triangles is acceptable,
- the length l_{QP_i} of any edge of the triangles in the final configuration conforms to the specified metric,
- the geometric approximation of the final configuration is controlled :

$$\min\langle \vec{N}(K_j), \vec{n}(P_i)\rangle < \tau \,,$$

for each triangle K_j of vertices $P_i, i = 1, 3$. For an isotropic mesh, for example, the value τ represents the maximum gap allowed, $\tau = \cos\theta$, where θ is such that $\alpha = 2\sin\theta$, $\alpha\rho(P)$ being the size imposed locally by the metric map at each point P (depending on the minimal radius of curvature $\rho(P)$), cf. Chapter 15).

Remark 19.24 *We are not trying here to strictly improve the shape quality of the initial ball. In practice, if $\mathcal{Q}_P$ denotes the quality of the initial ball and $\mathcal{Q}$ the quality of the new mesh, we want to have $\mathcal{Q} \geq 0.6\mathcal{Q}_P$ (the coefficient 0.6 is chosen so that a configuration of six equilateral triangles can be retriangulated into four triangles).*

- Edge reduction.

The reduction of an edge into a new vertex is an operation similar to the previous one, although it is more costly. Actually, in this case, two balls must be retriangulated (the ball of P and that of Q). Moreover, the position of the new vertex M must be supplied by the geometric support associated with the mesh. The geometric and shape quality requirements are identical to those used in the identification of the edge endpoints.

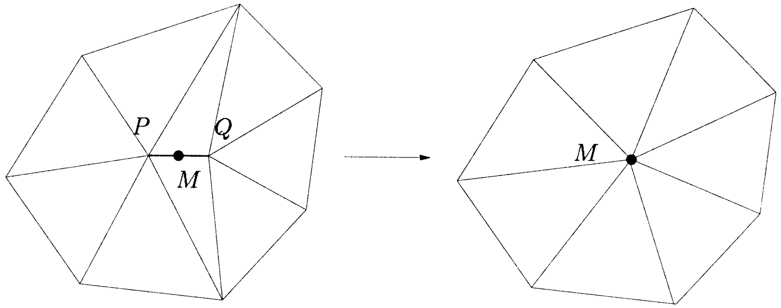


Figure 19.15: *Edge deletion by reduction. The vertices P and Q are merged into a new vertex M.*

Relaxation of the degree of a point. Recall that the degree of a vertex is the number of edges incident to this vertex. We have seen in Chapter 18 that in two dimensions, 6 is the optimal value for the degree of a mesh vertex. This value is also optimal for surface meshes.

The aim of the procedures for relaxing the degree of a point is to get rid of the *under-connected* and *over-connected* vertices (vertices for which the degree is respectively lesser than or greater than the target value). To this end, the process consists in iteratively applying edge swappings in order to reach the target value.

In the following section, we describe several techniques used for surface mesh optimization.

19.5 Optimization methods

The strategies carried out to optimize surface meshes are very close to those used for classical meshes in two and three dimensions.

The initial values of the quality measures relative to the mesh or to a particular set E of triangles (for example, a vertex ball or a pair of adjacent triangles) of the initial mesh are calculated and serve to initialize the optimization procedure. The same quantities are evaluated for each new configuration E' corresponding to a topological or geometric operation. The operator is applied if the optimization criteria are satisfied. In particular, the modification is carried out if the resulting configuration strictly improves the measure chosen ($\mathcal{Q}_{E'} > \mathcal{Q}_E$), or if the resulting configuration improves the measures by a certain coefficient ($\mathcal{Q}_{E'} > \alpha\mathcal{Q}_E$).

Among the various possible configurations, the one leading to the best (resp. first) optimization is picked. All these choices can also depend on the order in which the mesh entities are considered. Hence, it is possible to process the elements :

- in a natural order (*i.e.*, based on the increasing or decreasing indices),
- according to a specific relation of order (for example, sorted according to an increasing order of the edge lengths),
- of a certain type exclusively (for example, all triangles of a connected component, all boundary points, etc.),
- according to any other criterion, including a random order.

19.5.1 Shape optimization (classical case)

In the classical case, the surface mesh is optimized exclusively with respect to the element shape quality. From the initial mesh, a geometric support G^1 continuous is constructed to represent the underlying surface. This support is used notably to find the position of a point on the surface.

From the algorithmic point of view, the current mesh elements are iteratively analyzed and those whose quality exceeds a certain tolerance are removed. The deletion of badly-shaped elements is performed using the topological and geometric mesh modification tools.

In principle, a heuristic procedure is used which consists in simulating the application of an operator on the badly-shaped configuration and, based on the simulated results, in carrying out an optimization procedure on the mesh. Notice that it may sometimes be useful to degrade locally the mesh quality. Hence, for example, it may be interesting to merge two closely spaced points (event if the quality of the resulting configuration is worse than that of the initial one) before applying edge swapping to improve the shape quality of the new configuration.

19.5.2 Size optimization (isotropic and anisotropic case)

When one (or several) metric map(s) is (are) supplied, the optimization problem is a little more tedious. We first try to get back to a one size map problem, the geometric metric $\mathcal{G}_3$. Hence, the required metric intersections are carried out (Chapter 10).

Remark 19.25 *If the metric map supplied is not geometric, we calculate the metric map $\mathcal{G}_3$ using one of the approaches described in this chapter.*

The problem boils down to evaluating and then modifying a given mesh $\mathcal{T}$ according to a metric map $\mathcal{M}_3$. In practice, we use the induced metric $\mathcal{M}_2$, in other words, we consider the trace $\mathcal{M}_2(P)$ of $\mathcal{M}_3(P)$ in the tangent plane associated with each mesh vertex P.

The aim of size optimization for a surface mesh is to obtain a unit mesh with respect to the metric map $\mathcal{M}_2$. This approach is (obviously) based on the edge lengths analysis. Hence, mesh optimization involves applying the following process iteratively on the current mesh edges :

```
REPEAT
   FOR ALL edges $AB$ in $\mathcal{T}$
      calculation of $l_{AB}$
      IF $l_{AB} < \frac{1}{\sqrt{2}}$
         deletion of $AB$
      ELSE IF $l_{AB} > \sqrt{2}$
         subdivide $AB$ into $n$ sub-segments ($n$ depending of $l_{AB}$)
      END IF
   END FOR
   FOR ALL mesh vertices $P$
      node smoothing, coupled with
      local edge swappings
   END FOR
WHILE the current mesh is modified.
```

Remark 19.26 *The edges are, generally, processed in a random order. Thus, the edge splitting or vertex deletion operators are carried out alternatively.*

Remark 19.27 *A patch of the geometric support is associated with each newly created point so as to improve the further searching and localization procedures.*

Remark 19.28 *Finally, notice that it is more interesting (for practical considerations) to subdivide an edge using its midpoint, rather than inserting n points along this edge.*

19.5.3 Optimal mesh

Let us simply recall here that we aim at getting an optimal surface mesh, that is a mesh $\mathcal{T}$ such that :

$$\frac{1}{\sqrt{2}} \leq l_{AB} \leq \sqrt{2}, \quad \forall AB \in \mathcal{T}. \tag{19.40}$$

The above procedure (size optimization) aims at creating edges that satisfy this criterion.

19.6 Application examples

To conclude this chapter about surface mesh optimization, we now provide several examples of optimized isotropic surface meshes.

19.6.1 Surface mesh optimization examples

Table 19.1 presents some information related to the surface meshes. For the sake of clarity, we denote respectively by $\mathcal{T}_i^0$ the initial mesh, by $\mathcal{T}_i^g$ a geometric mesh (with respect to the map of the minimal radius of curvature) and by $\mathcal{T}_i^s$ a simplified geometric mesh. In this table, np and nt denote respectively the number of vertices and triangles in the mesh, ε is the relative distance to the surface allowed (in degrees, Chapter 15) and the quantities $\overline{\mathcal{P}_T}$, $\overline{\mathcal{D}_T}$, $\overline{\mathcal{R}_T}$ and $\overline{\mathcal{L}_T}$ correspond respectively to the average values of the geometric criteria of planarity, deviation, roughness and edge sizes in the meshes. Finally, the value $\overline{\mathcal{Q}_T}$ represents the shape quality of the mesh (*i.e.*, the quality of the worst element).

meshes	np	nt	ε	$\overline{\mathcal{P}_T}$	$\overline{\mathcal{D}_T}$	$\overline{\mathcal{R}_T}$	$\overline{\mathcal{L}_T}$	$\overline{\mathcal{Q}_T}$
$\mathcal{T}_1^0$	1,685	3,370	-	0.96	0.88	0.97	0.71	0.70
$\mathcal{T}_1^g$	36,329	72,658	0.01	0.99	0.96	0.99	0.95	0.86
$\mathcal{T}_1^s$	3,823	7,646	0.03	0.98	0.87	0.98	0.92	0.64
$\mathcal{T}_2^0$	67,106	134,212	-	0.80	0.57	0.75	0.72	0.46
$\mathcal{T}_2^g$	64,048	128,096	0.01	1.00	0.94	1.00	0.94	0.83
$\mathcal{T}_2^s$	11,814	23,628	0.02	0.98	0.88	0.99	0.96	0.84
$\mathcal{T}_3^0$	43,575	87,150	-	0.98	0.92	0.98	0.56	0.25
$\mathcal{T}_3^g$	77,291	154,582	0.02	0.98	0.87	0.99	0.95	0.84
$\mathcal{T}_3^s$	17,240	34,480	0.05	0.97	0.85	0.98	0.95	0.83

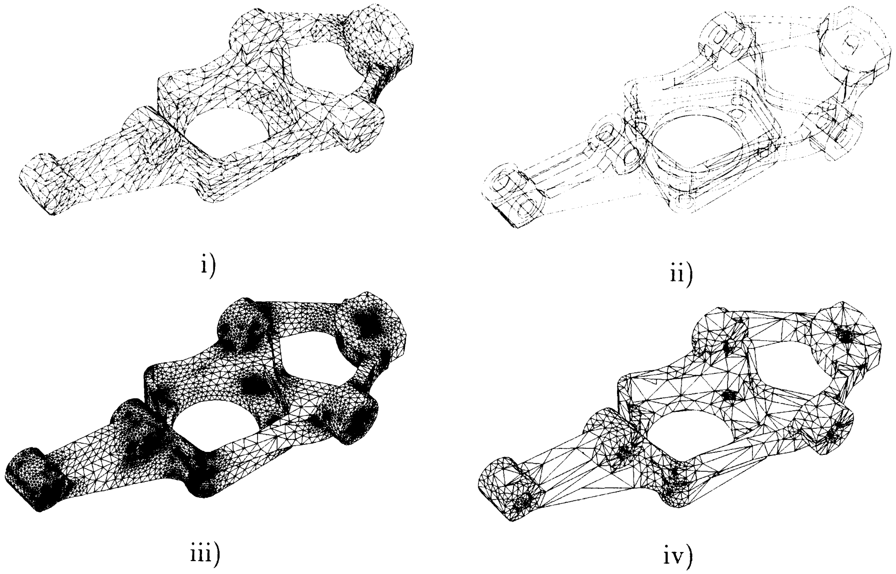
Table 19.1: Statistics related to various surface meshes.

Figure 19.16 shows an example of initial surface mesh, i). From this mesh, the edges separating the faces of C^0 continuity and the singular points have been identified, ii). Two meshes have then be generated from these specifications, an enriched and optimized mesh, iii) and a simplified mesh, iv). From the initial mesh, a surface composed of patches globally of class G^1 has been constructed thus defining the geometric support. This has been used to find the position of the vertices in the enriched mesh, obtained by subdividing the initial mesh edges. A simplified mesh has been extracted from the enriched mesh (see below).

19.6.2 Mesh simplification

A particular and interesting application of optimization concerns the geometric simplification of surface meshes. When the surface meshes have a number of elements that is too large to be used (for example, in numerical simulations or for visualization purposes, see below), it is desirable to reduce the number of elements, while preserving (to some extent) the geometric approximation of the surface. This is the aim of mesh *simplification*[10] algorithms.

[10] Also called *decimation, coarsening* or *reduction.*



i)

ii)

iii)

iv)

Figure 19.16: *Optimization of a surface mesh. i) initial surface mesh (data courtesy of Mac Neal-Schwendler Corp.) ii) identification of the ridges, G^1 discontinuities and constrained entities, iii) enriched geometric mesh (metric correction $\varepsilon = 1.5$, surface deviation 5 degrees) iv) simplified geometric mesh (surface deviation 14 degrees, without metric correction).*

Basic principle. In principle, mesh simplification is an operation of the same nature as optimization, in that it involves the same operators. In practice, given a (geometric) surface mesh supposed to comply with a given size map $\mathcal{G}_2$, the simplification aims at creating an optimal surface mesh with respect to a modified size map, corresponding to a greater surface approximation.

The first stage consists in calculating a geometric metric map $\widetilde{\mathcal{G}_2}$ from the map $\mathcal{G}_2$ for a greater tolerance ε than the original one. Then, a classical optimization procedure described previously is applied to obtain a simplified surface mesh.

The key to the method lies in respecting the following rules (see, for example, [Frey,Borouchaki-1998]) :

- the control of the distance to the geometry, *the simplified map is a geometric map*,
- the resulting mesh is optimal with respect to this modified map,
- the shape quality of the final mesh elements is controlled.

Remark 19.29 *Notice however that a vertex where the minimal radius of curvature ρ is smaller than the threshold ε (within a coefficient) can be simplified.*

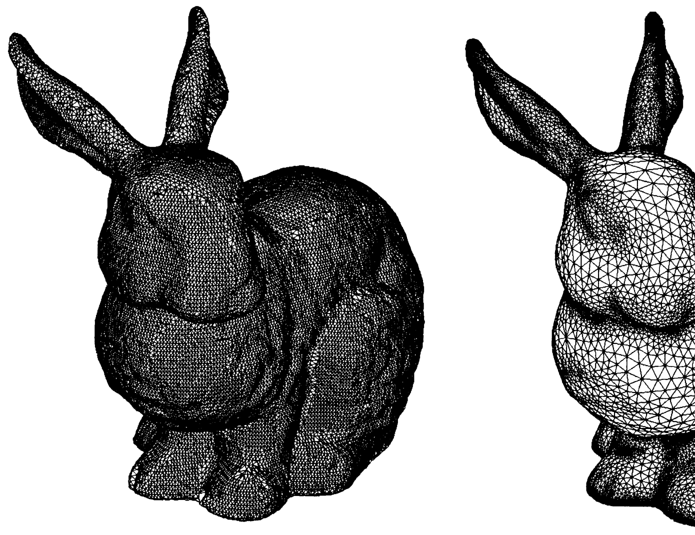


Figure 19.17: *Left-hand side : original mesh reconstructed from scanned data (Computer Science Department, Stanford University). Right-hand side : corresponding isotropic geometric mesh.*

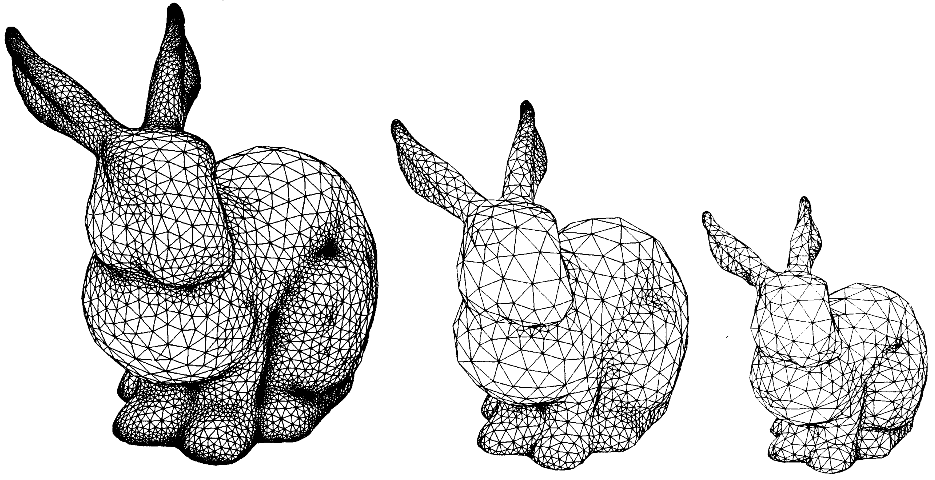


Figure 19.18: *Geometric simplifications of a surface mesh, corresponding to different deviations to the surface $\varepsilon = 0.03$, $\varepsilon = 0.1$ and $\varepsilon = 0.29$. Notice that as we move further from the object (as simulated here), the visual aspect remains very close to the original one.*

This idea makes it possible to remove details that are judged useless (at the given resolution) and, therefore, to construct simplified geometric meshes, within ε.

Actually, given the previous remark, it is easy to note that mesh simplification consists in working in a "strip" of a given width (obviously related to the specified tolerance value). The modifications are applied when the edges concerned by the operation remain in this strip.

Application to visualization. A "natural" application of mesh simplification is related to the representation of scenes (graphic visualization). Based on the distance and the point of view selected, it may be interesting to have various surface meshes at variable resolutions. Figure 19.18 illustrates this idea of mesh simplification adapted to graphic visualization.

Remark 19.30 *Moreover, mesh simplification enables the* compression *of surface meshes, which may be extremely useful, particularly for transmitting data [Taubin,Rossignac-1998].*

Chapter 20

A touch of finite elements

Introduction

Up to now we have discussed many aspects regarding mesh generation and mesh modification and while our final objective for such meshing activities is to provide meshes required for finite element simulations, we have not yet dealt with such methods[1] directly.

However, in some parts of the book, we have faced some mesh generation problems where more advanced knowledge about finite elements was necessary to understand what needed to be done (see, for instance, the numbering issues related to finite element nodes in Chapter 17). Also, in later chapters, we will discuss h-methods, p-methods and hp-methods (in terms of meshing technologies) and clearly, at that time, it will be necessary to know more about the corresponding finite element requirements. Thus, this chapter on the finite element method appears to be necessary before further meshing investigations.

Nonetheless, library bookshelves are straining under the weight of literature on finite element theory as well as practical manuals for finite element methods, and it is clearly impossible for this book to compete with such a wealth of specialized literature. Thus the point of view adopted here will be clearly motivated by the following question : " what must be known about finite elements in order to successfully deal with a meshing problem ?".

In what follows, the theoretical point of view is largely based on [Ciarlet-1991] to which the reader is referred for a more advanced view of the problem together with a comprehensive list of relevant references about finite element theory and practice. The above reference also includes a long list of other papers and books related to the theory and finite element applications in the engineering fields.

★
★ ★

[1] Only some brief allusions to finite element theory were given in Chapter 1 for instance.

Given this objective, the chapter is organized as follows. We consider a simple problem to which the finite element method is applied so as to find an approximate solution. The main aspects of the method are introduced. We start from the continuous P.D.E. problem which models the problem under investigation. We replace this problem by a discrete problem whose solution is approached by means of a finite element method. The notion of a finite element is discussed, defined as a geometrical element considered along with a set of degrees of freedom and a set of basic functions. Several types of finite elements are given including curved elements. Error estimate issues are briefly covered and related to the notions introduced in Chapter 10.

In terms of practice, curved finite elements are analyzed. Then, explicit use of a finite element approximation is discussed where we show how to compute a stiffness matrix, a right hand side and the resulting discrete system. This will present an opportunity to see how a finite element approximation can be implemented on a computer. Finally, some examples of popular finite elements are given to illustrate what types of interpolation, nodes and degrees of freedom can be encountered.

20.1 Introduction to a finite element style computation

To introduce the terminology, notations, etc., which will be used, we consider a very simple P.D.E. problem and we briefly mention the main steps of its finite element approximation. Furthermore, P^1 and P^2 examples of finite elements will be illustrated. In specific, we will show how to compute the stiffness matrix and the right-hand side of such elements for the P.D.E.'s used as an example.

Let us consider a simple academic example. Let Ω be a domain and Γ be its boundary. We assume that Ω is composed of two sub-domains, denoted as Ω_i, and that Γ includes three parts, denoted as Γ_i. We want to solve the following problem, find u, the solution to :

$$\left\{\begin{array}{rcll} -div(k_i\nabla u) & = & F_i & \text{in } \Omega_i \quad (i=1,2) \\ k_i\dfrac{\partial u}{\partial n}+g_i u & = & f_i & \text{on } \Gamma_i \quad (i=1,2) \\ u & = & 0 & \text{on } \Gamma_3 \end{array}\right. \qquad (20.1)$$

where ∇ represents the gradient and $\dfrac{\partial}{\partial n}$ is the derivative with respect to the normal along Γ_i.

This system of partial differential equations models a heat transfer problem[2]. The physical conditions and the material coefficients to which the problem is subjected include the conductivity coefficient k_i of domain Ω_i, the source term F_i for the sub-domain Ω_i, the flux term f_i for the boundary Γ_i, the transfer coefficient g_i for the boundary Γ_i and, finally, the Dirichlet condition $u = 0$ which is prescribed on Γ_3.

[2] *I.e.*, the initial problem is a heat transfer problem and physicists have proved that such a P.D.E. system is an adequate model for it.

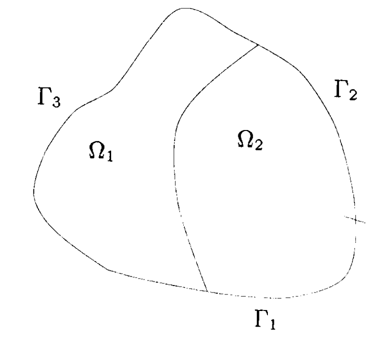


Figure 20.1: *The domain Ω with its two sub-domains and the domain boundary Γ with its three components.*

In what follows, we will describe the method, without going into too great a detail. The initial P.D.E. problem is then analyzed with a view to a finite element approximation. This analysis includes several steps. A weak formulation is established, then approximate solutions are constructed.

First, a better suited formulation, called the *variational formulation*, or *weak formulation*, is derived from the above formulation by means *of partial differential equations* (P.D.E.) (after a Green formula). This weak formulation can be written as follows :

$$\left\{\begin{array}{l} \text{Find} \quad u \in V \quad \text{such that} \\ a(u,v) = f(v)\,, \; \forall v \in V\,, \end{array}\right. \qquad (20.2)$$

where V is the space of admissible values. In this way, we meet the operator a and the form f which are associated with the operators of Relationship (20.1).

In fact, using a Green formula, one sees that if u conforms to (20.1) and if v is an appropriate differentiable function, we have successively :

$$\sum_{i=1}^{2}\int_{\Omega_i}(-\,div(k_i\,\nabla u) - F_i)\,v \;=\; 0\,,$$

$$\sum_{i=1}^{2}\left(\int_{\Omega_i} k_i\,\nabla u\,\nabla v \;-\; \int_{\Omega_i} F_i\,v \;-\; \int_{\Gamma_i} k_i\,\frac{\partial u}{\partial n}\,v\right) \;=\; 0\,,$$

$$\sum_{i=1}^{2}\left(\int_{\Omega_i} k_i\,\nabla u\,\nabla v \;+\; \int_{\Gamma_i} g_i\,u\,v \;-\; \int_{\Omega_i} F_i\,v \;-\; \int_{\Gamma_i} f_i\,v\right) \;=\; 0\,.$$

This leads to a formulation equivalent to (20.1), this variational formulation then serves as a starting point for the finite element approximation. We return to the formulation given in (20.2) by introducing the following definitions :

$$a(u,v) = \sum_{i=1}^{2}\left(\int_{\Omega_i} k_i\,\nabla u\,\nabla v \;+\; \int_{\Gamma_i} g_i\,u\,v\right)$$

$$f(v) = \sum_{i=1}^{2} \left(\int_{\Omega_i} F_i\, v + \int_{\Gamma_i} f_i\, v \right) .$$

This *continuous problem* is then replaced by an *approximate problem*. In practice, the explicit solution of the continuous problem is not generally possible. This has led to the investigation of approximate solutions using, in our context, the finite element method. Basically, this method consists of constructing a *finite dimensional* sub-space V_h of space V and defining u_h, an approximate solution of u, the solution to the following problem :

$$\left\{ \begin{array}{l} \text{Find} \quad u_h \in V_h \quad \text{such that} \\ a(u_h, v_h) = f(v_h)\,,\ \forall v_h \in V_h\,. \end{array} \right. \tag{20.3}$$

Under appropriate assumptions (see below), it can be shown that this problem has a unique solution, u_h, and that the *convergence* of u_h to the solution, u, is directly related to the manner in which the functions v_h of space V_h approach the functions v of space V and therefore the manner in which the space V_h is defined.

Hence, the finite element method consists of constructing a finite dimensional space V_h such that, on the one hand, a suitable approximation is obtained and, on the other hand, the actual computer implementation is not too difficult. This construction is based on the three following basic ideas :

1. the creation of a mesh, denoted by $\mathcal{T}_h$[3] or simply $\mathcal{T}$, of domain Ω so that the domain can be written as a finite union of elements K, the members of $\mathcal{T}$;
2. the definition of V_h as the set of functions v_h, whose restriction to each element K in $\mathcal{T}$ is, in general, a polynomial;
3. the existence of a basis for the space V_h whose functions have a *small support*.

In other words, constructing a mesh $\mathcal{T}_h$ is an essential prerequisite when defining space V_h. The functions in V_h are usually polynomials over every element in the mesh which are continuous over the neighboring elements (*i.e.*, across the element interfaces). The simplest case for triangular meshes (in two dimensions) is to choose piecewise linear functions which, after interpolation on the mesh node values, make it possible to define the desired solution everywhere.

Remark 20.1 *Index h in V_h (and in $\mathcal{T}_h$) is related to the fact that V_h depends on the mesh. This value h must be seen as a size parameter (the element diameter) which, in fact, defines a family of meshes and, therefore, a family of approximations and which, to demonstrate the theoretical convergence and accuracy (the approximation order), tends towards* 0.

[3] By convention, $\mathcal{T}_h$ denotes a triangulation of Ω such that :

$$h = \max_{K \in \mathcal{T}_h} h_K$$

where h_K is the diameter of polyhedron K.

The elements K in mesh $\mathcal{T}_h$ have a certain number of properties which are characteristics in the finite element method (Chapter 1).

The choice of the basis function in V_h consists in taking functions whose restriction at every element K is written as :

$$u_h(x) = \sum_{i=1}^{N} \phi_i(u)\, p_i(x)\,, \tag{20.4}$$

where p_i is the basis function i of the polynomial space previously defined (while $p_i(x)$ is the value of this basis function at a location x in K) and $\phi_i(u)$ is the value of the degree of freedom i associated with u_h while N denotes the *number of degrees of freedom* and the number of p_i's as well, *i.e.*, the dimension of the space engendered by these p_i's. As will be seen in greater detail below, a finite element is then characterized by a suitable choice of the following triple :

- K, a geometrical element;
- P_K, a finite dimensional space of functions defined over K;
- Σ_K, a set of degrees of freedom associated with the functions defined over K.

Hence, for all unknowns in the problem (here, the unique unknown, u, is a "temperature") we have $dimP_K = card(\Sigma_K) = N$ and a section below discusses this triple in more detail, the basic support for defining the finite elements.

Provided with these choices, Problem (20.3) is then replaced by a matrix problem. In fact, if locally, in each K, we have :

$$u_h(x) = \sum_{i=1}^{N} \phi_i(u)\, p_i(x)\,,$$

globally we have a similar expression, *i.e.*, :

$$u_h(x) = \sum_{j=1}^{M} \phi_j^*(u)\, p_j^*(x)\,,$$

where M is now the total number of degrees of freedom when the mesh is considered globally and when we have removed[4]the degrees of freedom of the nodes in Γ_3, while $\phi_j^*(u)$ and $p_j^*(x)$, at index j, are indeed the values $\phi_i(u)$ and $p_i(x)$ for the indices i of the corresponding elements K. Therefore, for (20.3), we have :

$$a(u_h, p_j^*(x)) = f(p_j^*(x)) \text{ for } j = 1, M$$

and the problem is now :

$$\left\{ \begin{array}{l} \text{Find} \quad U_k\ k = 1, M \quad \text{such that} \\ a(\sum_k U_k p_k^*(x), p_j^*(x)) = f(p_j^*(x))\,, \end{array} \right. \tag{20.5}$$

[4] This is a formal point of view, in practice, some techniques are used that allow for the Dirichlet boundary conditions with no specific numbering of the nodes as implicitly assumed here.

which can be written in a matrix form as follows :

$$\mathcal{A}\,U = \mathcal{B}$$

where $\mathcal{A}$ is the matrix with coefficient $a_{i,j} = a(p_i^*(x), p_j^*(x))$ and $\mathcal{B}$ is the *right-hand side*, whose component b_j is $b_j = f(p_j^*(x))$.

Before discussing the element level, note that the basis function j, in the linear case, is 1 at node j and 0 otherwise (the *hat* function). This property leads to sparse matrices, in fact, we have $a_{i,j} = 0$ for most pairs (i, j). For example, again in the simplest case, the triangle of degree 1, we have $a_{i,j} = 0$, unless the nodes i and j belong to the same triangle K.

20.2 Definition and first examples of finite elements

Here, we return to the formal definition of a finite element, using the above triple. Then, we give some examples of such triples.

20.2.1 Definition of a finite element

Now, we specify what the above triple means. As previously mentioned and again after [Ciarlet-1991], a finite element is defined by the triple[5] :

$$[K, P_K, \Sigma_K]\,.$$

In this triple K is the "geometric" element, *i.e.*, a mesh element. P_K denotes the space functions of the finite element. The basis functions in this space are referred to as the *shape functions* of the finite element. Finally, Σ_K is the set of degrees of freedom associated with K.

From a geometric point of view, and according to the space dimension, K may be a triangle, a quadrilateral, a tet, a pentahedron , a hex, etc.

Space P_K is generally made of polynomials. This space with the finite dimension N is built from N basis functions. Thus, if p_i is a basis function, any function p in P_K can be written (cf. Relation (20.4)) as :

$$p = \sum_{i=1}^{N} \alpha_i\, p_i\,,$$

where the α_is are some coefficients (in fact, the degrees of freedom).

The set of degrees of freedom Σ_K, the ϕ_i's in Relation (20.4) or, similarly, the above α_i's may be some values of the function p, where p is a function in P_K, at a node[6] in the element K. Such a node is denoted by a in the following. A finite element with this type of node is said to be a *Lagrange* type element.

[5] In fact, K itself is often referred to as a finite element. This clearly corresponds to a simplification in the notations since, indeed, this is K combined with P_K and Σ_K which is one finite element.

When the $\phi_i(p)$'s include at least one value of a derivative of p, for example, with evident notations, Dp, $\frac{\partial p}{\partial x}$, $\frac{\partial p}{\partial y}$, $\frac{\partial p}{\partial n}$, D^2p, $\frac{\partial^2 p}{\partial x^2}$, etc., we face a *Hermite* type finite element.

Following these definitions, one notes that P_K can be seen as a space of functions (the p's) or directly as a list of basis functions (the p_i's), which is clearly two ways of writing the same thing. Similarly, Σ_K can be defined as a set of degrees of freedom, the $\phi_i(p)$'s, and we reach a global node definition, or a set of nodes, the a's, with which one or several degree(s) of freedom is(are) associated. Based on the context, one or the other of these definitions will be used.

The important fact is that the triple $[K, P_K, \Sigma_K]$ properly defines a finite element if the set Σ_K is P_K-*unisolvent*. This means that there exist N functions p_i's in space P_K which are linearly independent. In particular, we have :

$$\phi_j(p_i) = \delta_{ij}$$

with δ_{ij} the Kronecker delta ($\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise).

A more general definition of this concept is to make use of one *reference element*, $\hat{K}$, together with a mapping function F_K so as to define the above triple from a reference triple denoted by :

$$[\hat{K}, \hat{P}, \hat{\Sigma}]\,.$$

In other words, a unique reference triple $[\hat{K}, \hat{P}, \hat{\Sigma}]$ may serve to characterize all finite elements in the mesh which have the same geometric type provided the corresponding functions F_K are given. Thus, starting from one triple $[\hat{K}, \hat{P}, \hat{\Sigma}]$ and one function F_K, it is possible to obtain the triple $[K, P_K, \Sigma_K]$ of every element K. If $\hat{p}$ (resp. $\hat{a}$) denotes the functions (resp. the nodes) in $\hat{P}$ (resp. in $\hat{K}$), we have :

$$\begin{aligned} K &= F_K(\hat{K}) \\ P_K &= \{p = \hat{p} \circ F_K^{-1}\,, \hat{p} \in \hat{P}\} \\ \Sigma_K &= \{p(a)\,, a = F_K(\hat{a})\,, \hat{a} \in \hat{K}\}\,. \end{aligned} \tag{20.6}$$

where, for the sake of simplicity, we assume only one degree of freedom of the form $p(.)$ and where $\hat{a}$ stands for the nodes in $\hat{K}$. As previously mentioned, P_K can be also written as :

$$P_K = \{p = \sum \alpha_i\, p_i \quad \text{with} \quad p_i = \hat{p}_i \circ F_K^{-1}\,, \hat{p}_i \in \hat{P}\,, i = 1, N\} \tag{20.7}$$

for convenience, while in turn and for the same reason, Σ_K can be expressed in two ways :

$$\Sigma_K = \{\phi_i(p)\,, i = 1, N\}\,. \tag{20.8}$$

$$\Sigma_K = \{a_j\,, j = 1, M\,; \phi_i(p(a_j^i))\,, i = 1, N\}\,. \tag{20.9}$$

[6] A node supports one (or several) degrees of freedom. According to their type, the nodes in a finite element may be its vertices, its mid-edge points, some points in its facets or some particular points inside K.

In other words, either using the entire list of degrees of freedom, or using the list of nodes and, for each of them, the list of the corresponding degrees of freedom.

To make sense, the above definition implies that F_K is *invertible*. This being satisfied, the two triples are equivalent (and, if in addition, F_K is affine as it is in the P^1 case, the two triples are affine equivalent). In this way, we have defined a family of finite elements which may be represented using a unique element, the reference element. Notice that making use of a reference element is not strictly required in the case of simplicial element, but this is a source of simplification due to its great ease of application.

Remark 20.2 *This notion of equivalence allows us to reduce the definition of a finite element to that of the corresponding reference element. As will be seen, this property is met again at the computational step, a part of the computation for a given element concerns the reference element and therefore does not change when we consider one element or another of the mesh.*

Remark 20.3 *In the case where the function is affine, we have affine elements. In contrast, other types of elements can be defined, for example, curved (or isoparametric) elements.*

20.2.2 First examples of finite elements

In this section we give some examples of finite elements which are rather simple (and widely used). At the end of this chapter, a certain number of other finite elements will be given.

The P^1 triangle in two dimensions. The first example is the well known P^1 Lagrange triangle in two dimensions (also referred to as Courant's triangle or as the $T3$ element). It is defined, Figure 20.2 (top), by means of the following reference definitions :

- $\hat K$, the straight triangle of "unit" side whose vertices are $\hat v_1 = (0,0)$, $\hat v_2 = (1,0)$ and $\hat v_3 = (0,1)$.
- $\hat P = P^1 = \{1-\hat x-\hat y\ ,\hat x\ ,\hat y\}$, *i.e.*, $dimP_K = 3$.
- $\Sigma = \{\hat a_i = \hat v_i\ ,i=1,3\ ;\phi_i(p) = p(a_i), a_i = F_K(\hat a_i),\ i=1,3\}$
- $F_K \in (P^1)^2$.

Remark 20.4 *Note that F_K can be written as $F_K = \{F_K^i\}_{i=1,d}$ where d is the spatial dimension and we have $F_K^i \in P^1$.*

The Q^1 quadrilateral in two dimensions. The Q^1 Lagrange quad in two dimensions (also referred to as the $Q4$ element) is defined as shown in Figure 20.2 (bottom) :

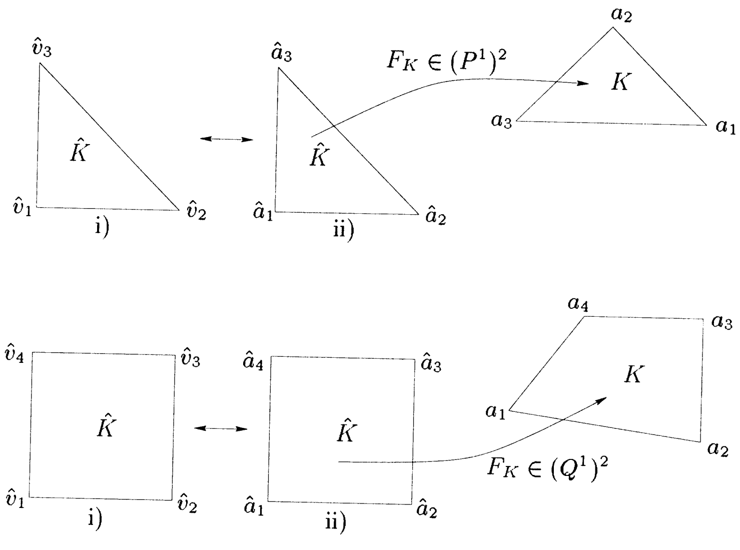


Figure 20.2: *Lagrange type finite elements of degree 1, the T3 and the Q4 finite elements. Left-hand side, the vertex labeling in the reference element (indeed $\hat v_1$ could simply be the vertex of label 1, etc.). Middle, the reference node labeling and, right-hand side, the current node labeling.*

- $\hat K$, the unit square whose first vertex $\hat v_1$ is at the origin[7];
- $\hat P = Q^1 = \{(1-\hat x)(1-\hat y)\ ,\hat x(1-\hat y)\ ,\hat x\hat y\ ,(1-\hat x)\hat y\}$. $dimP_K = 4$.
- $\Sigma = \{\hat a_i = \hat v_i\ ,i=1,4\ ;\phi_i(p) = p(a_i), a_i = F_K(\hat a_i),\ i=1,4\}$
- $F_K \in (Q^1)^2$.

The two P^2 triangles in two dimensions. Figure 20.3 (left-hand side), the *affine* P^2 Lagrange triangle in two dimensions (also referred to as the $T6$ element) is defined as :

- $\hat K$, the "unit" triangle as for the $T3$ element.
- $\hat P = P^2 = \{(1-\hat x-\hat y)(1-2\hat x-2\hat y)\ ,\hat x(2\hat x-1)\ ,\hat y(2\hat y-1)\ ,4\hat x(1-\hat x-\hat y)\ ,4\hat x\hat y\ ,4\hat y(1-\hat x-\hat y)\ \}$. $dimP_K = 6$.
- $\Sigma = \{\hat a_i = \hat v_i\ ,\hat a_{i+3} = \frac{\hat v_i+\hat v_{i+1}}{2}\ ,i=1,3\ ;\phi_i(p) = p(a_i), a_i = F_K(\hat a_i),\ i=1,6\}$
- $F_K \in (P^1)^2$.

The *isoparametric* P^2 Lagrange triangle in two dimensions is similarly defined but, in this case, we have $F_K \in (P^2)^2$.

The two Q^2 quadrilaterals in two dimensions. The Q^2 Lagrange quad in two dimensions (also referred to as the $Q8$ element), Figure 20.3 (right-hand side), is defined as :

[7] Obviously, it is also possible to center the square, to give it a side 2 and to define if from -1 to $+1$, for example.

- $\hat{K}$, the unit square.
- $\hat{P} = Q^2 = \{(1-\hat{x})(1-\hat{y})(1-2\hat{x}-2\hat{y})\ ,\hat{x}(1-\hat{y})(2\hat{x}-2\hat{y}-1)\ ,-\hat{x}\hat{y}(3-2\hat{x}-2\hat{y})\ ,(1-\hat{x})\hat{y}(-2\hat{x}+2\hat{y}-1)\ ,4\hat{x}(1-\hat{x})(1-\hat{y})\ ,4\hat{x}\hat{y}(1-\hat{y})\ ,4\hat{x}(1-\hat{x})\hat{y}\ ,4(1-\hat{x})\hat{y}(1-\hat{y})\}$. $dim P_K = 8$.
- $\Sigma = \{\hat{a}_i = \hat{v}_i\ ,\hat{a}_{i+4} = \frac{\hat{v}_i+\hat{v}_{i+1}}{2}\ ,i=1,4\ ;\phi_i(p) = p(a_i), a_i = F_K(\hat{a}_i),\ i=1,8\}$
- $F_K \in (Q^1)^2$ (affine case) or $F_K \in (Q^2)^2$ (isoparametric case).

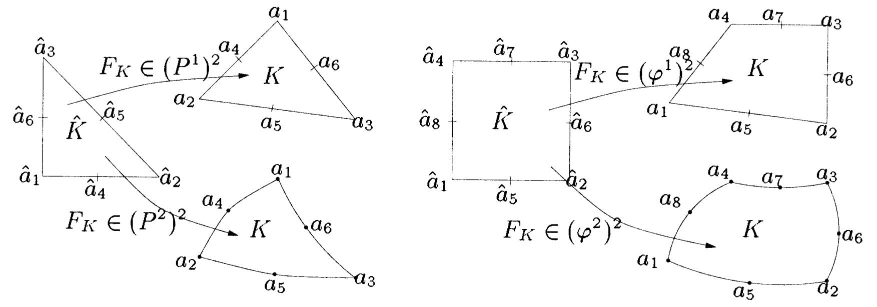


Figure 20.3: *Lagrange type finite elements of degree 2, affine (top) and isoparametric (bottom) T6 and Q8 finite elements together with the corresponding reference element.*

Remark 20.5 *In the case of an affine element, the mid-edge node is the midpoint of the edge and, for instance for the T6 element,* $a_4 = F_K(\hat{a}_4) = F_K(\frac{\hat{a}_1+\hat{a}_2}{2})$. *In this case, we have* $F_K(\frac{\hat{a}_1+\hat{a}_2}{2}) = \frac{F_K(\hat{a}_1)+F_K(\hat{a}_2)}{2}$ *and* a_4 *can be obtained as* $\frac{a_1+a_2}{2}$. *However, for the curved T6 element* $F_K(\frac{\hat{a}_1+\hat{a}_2}{2}) \neq \frac{F_K(\hat{a}_1)+F_K(\hat{a}_2)}{2}$ *and then* $a_4 = F_K(\hat{a}_4) = F_K(\frac{\hat{a}_1+\hat{a}_2}{2})$ *is not, in general,* $\frac{a_1+a_2}{2}$.

In a subsequent section, we will give various examples of common finite elements other than those presented here.

20.3 About error estimation and convergence

The initial problem, Problem (20.2), concerns a solution in space V of $a(u,v) = f(v)$. The approached problem, Problem (20.3), is to find a solution to $a(u_h, v_h) = f(v_h)$ in space V_h. If this problem is solved using one or several numerical schemes or quadratures, this leads to the solution $a_h(u_h, v_h) = f_h(v_h)$ in which a_h (resp. f_h) are approximations of a (resp. of f). The question is then to ensure that the approached solution u_h tends towards u with h in a pertinent way thanks to a proper choice of the various parameters and ingredients used in the various steps included in the approximation.

20.3.1 Local (element) quantity and global quantities

Passing from the initial continuous problem (formulated in domain Ω) to its numerical approximation and to the calculus corresponding to the mesh elements approaching Ω is thus made by replacing the global quantities by some local discrete quantities. The various relationships between these two types of quantities are as follows :

- Ω is replaced by $\mathcal{T} = \bigcup_{K \in \mathcal{T}} K$,
- Γ is approached by the "sides"∂K of the appropriate element K,
- a function v_h in V_h is replaced by its restriction in K, v_h/K, in space $P = P_K$,
- a global node b_j corresponds to a set of local nodes a_i, several elements K sharing this node b_j. One finds here a numbering problem leading to identifying the global index of a local node,
- similarly, the basis functions and the degrees of freedom are seen at a global and a local level,
- the restriction in K of a $v_h = \Pi_h v$ is $\Pi_K v$.

These notions (and notations) being done, we recall, using a simple example (an elliptic problem) the abstract basis of the convergence and error estimate issues.

20.3.2 Error estimation and convergence issues

Error estimate as well as convergence of the approximation is analyzed by observing if :

$$||u - u_h|| \longrightarrow 0$$

when $h \longrightarrow 0$ in some sense, *i.e.*, for an appropriate norm $||\,.\,||$.

From a mathematical point of view, the notion of a convergence is linked not to a (discrete) approximation but to a family of (discrete) approximations defined by a parameter h which tends towards zero. With each value of h is associated a space V_h and with each V_h is associated one approximation u_h of Problem (20.3) :

$$a(u_h, v_h) = f(v_h)\,.$$

Such a family is said to be convergent if :

$$\lim_{h \longrightarrow 0} ||u - u_h|| = 0\,.$$

After the previous discussion, the convergence of u towards u_h, *i.e.*, $||u - u_h|| \longrightarrow 0$, is based on Cea's lemma which proves that a constant C exists, independent of h, such that :

$$||u - u_h|| \leq C \inf_{v_h \in V_h} ||u - v_h||$$

and, therefore, a sufficient condition for convergence is that there exists a family of V_h such that :

$$\lim_{h \longrightarrow 0} \inf_{v_h \in V_h} ||u - v_h|| = 0\,.$$

Then, we need to show that :

$$||u - u_h|| \leq C(u)\, h^{\beta}\,,$$

which implies that :

$$\|u-u_h\| = \mathcal{O}(h^\beta)\,,$$

meaning that the convergence is of order β for the norm $\|\,.\,\|$, while it is obvious that different norms can be found according to the regularity of the problem and the nature of the approximation.

The distance between u and u_h, *i.e,* $\inf_{v_h\in V_h}\|u-v_h\|$, is estimated by looking at the distance between u and $\Pi_h u$ the interpolate of u. This leads to a result in :

$$\|u-u_h\| \le C\,\|u-\Pi_h u\|\,.$$

If we assume Ω to be a polygonal (resp. polyhedral) domain, then we have strictly :

$$\Omega = \sum_{K\in\mathcal{T}} K$$

where K is a polygonal (resp. polyhedron) and, in this case, straight finite elements (the above triple) can be chosen so as to construct the sub-spaces V_h (which are therefore included in V). Then, in such a situation, we have a *conforming* finite element method.

Hence,

$$(\Pi_h\, u)/K = \Pi_K u$$

for all K in $\mathcal{T}$ and, one can write :

$$\|u-\Pi_h\, u\| = \{\sum_{K\in\mathcal{T}}\|u-\Pi_K\, u\|^2\}^{\frac{1}{2}}\,.$$

As a consequence, estimating the error between u and u_h comes down to seeing what the local errors of interpolation are :

$$\|u-\Pi_K\, u\|\,.$$

We then show (after some assumptions about the interpolation) that there exists a constant C, independent of K, such that :

$$|v-\Pi_K v|_{m,K} \le C\,\frac{h_K^{k+1}}{\rho_K^m}\,|v|_{k+1,K}$$

for $0\le m\le k+1$ and for all v in $H^{k+1}(K)$. This relation is related to the degree k of the polynomials in P_K and two quantities already mentioned many times :

- the diameter of element K, say h_K and
- the radius of the circle (the sphere) inscribed in K, *i.e.*, ρ_K.

From a practical point of view and for the envisaged error estimate, we use one reference element, $\hat{K}$, on which the error is estimated. Then, we evaluate this error on element K (assumed to be affine equivalent). The full proof for this estimation is slightly delicate and makes use of various issues in functional analysis. This is beyond our scope here. It is enough to retain that, based on the problem considered and according to the assumed regularity, we obtain some majorations for some appropriate norms which enable us to prove the convergence of these approximations together with the order of this convergence.

20.3.3 Quadrature influence

As already seen, the actual computation often makes use of integration schemes. In this way, the problem to be solved is no longer $a(u_h,v_h)=f(v_h)$ but rather $a_h(u_h,v_h)=f_h(v_h)$. If ϕ stands for a function that must be integrated in K, we then compute :

$$\int_K \phi(x,y)dK = \sum_l \omega_l\phi(x_l,y_l)$$

where the quadrature is defined by means of l points (x_l,y_l) and l weights ω_l.

Bear in mind that the choice of the quadrature formula must be made in such a way as to retrieve the corresponding error estimate (the order of convergence) when no quadrature is used. In other words, the quality of the approximation is preserved.

For instance, this is the case when, for an approximation based on polynomials of degree k, we use a formula which exactly integrates the polynomials of degree $2k-2$.

20.3.4 Curved elements

Using curved or isoparametric elements is one way to deal with non polygonal (polyhedral) domains for which an approximation by means of straight finite elements leads to a poor geometric approximation.

For such domains whose boundary or a portion of the boundary is curved, we generally use two types of finite elements. Inside the domain[8], straight finite elements are employed. Near the domain boundary, we use finite elements with at least one curved edge (face), in order to have a more precise approximation of the underlying geometry. These elements are said to be "curved" or isoparametric. In this short section, we give some indications about this type of element and, specifically, we show how they can be constructed.

For such elements, the error analysis is made as it was for straight (affine) elements. In particular, provided the deformation between the curved elements and the straight ones is not too great, the same discussion holds. Similarly, the role of a quadrature formula is as above.

The concern is more specifically how to define an isoparametric element. This leads to a problem of element node definition which comes down to finding a (local) index and, more importantly, a suitable location for these nodes.

Let us again take the case of a curved $T6$ triangle. As previously seen, this element has as its nodes its three vertices and its three mid-edge points. In fact, these last three nodes are effectively the midpoints of the edges when these edges are straight segments. The case of a curved edge is slightly different. Here the theory and the practice are somewhat different. In theory, specifically for a convergence study, it is convenient to take as a mid-edge node for a given edge the point projection on the boundary of the mid-edge point; we assume in addition that the distance between these two points is small. This is clearly the case when

[8] In principle, as it is not necessarily true everywhere.

h tends towards zero, and therefore the convergence issue may be based on such an assumption. Nevertheless in practice, the curved edges are represented by a number of *arcs of parabola* and, *unfortunately*, the best approximation possible for a curve by such arcs does not mean that the projection of the midpoint of the edge is the best possible candidate to construct an arc of parabola close enough to the geometry. Moreover, since h does not tend towards zero, such a fact is not necessarily without effect. Thus, as will be seen in detail in Chapter 22, the mid-edge nodes are not necessarily the projections of the mid-edge points concerned, they are also related to the local curvature of the boundary near the region under examination.

To summarize, one needs to :

- mesh the curve related to the boundary edge by means of an arc of parabola as close as possible to the geometry (Chapter 22),
- check that the Jacobian of the transformation defined from the reference element to the current element is strictly positive in all points, so as to ensure the property of invertibility and thus results in elements with a positive surface area.

Nevertheless, this scheme merits some remarks. First, it is not always true, in practice, that the edge is close enough to the boundary curve. This condition would hold in regions with a high curvature (*i.e.*, where the minimal radius of curvature is small) only for a sufficiently small edge. Then, the P^2 element (if we consider this case) thus constructed would probably be adequate but certainly too small, therefore leading to a mesh with too many elements. This refutes the fact that we know in advance that, for an equivalent or a better geometric approximation, the size h of a curved element may be larger than the size of a straight element constructed at the same place. An intuitive idea is to consider that a P^1 mesh of size at least $2\,h$ allows the construction of a P^2 mesh with the same size which is equivalent or better than a P^1 mesh with the half size (*i.e.*, h). Note that in this way, for the same number of degrees of freedom, the number of elements is less. We refer the reader to Chapter 22, where there are some indications about the construction of P^2 meshes (the discussion is based on a P^1 mesh completed by one of the automatic mesh generation methods previously described).

Remark 20.6 *In this discussion, we have essentially seen as a curved element example the case of simplicial elements (in two dimensions) of degree 2. However, it is clear that there exist other isoparametric elements, with a degree other than 2, a dimension other than 2 and a geometric nature other than simplicial.*

Remark 20.7 *To end, notice that the case of elements in three dimensions is far from being trivial. Indeed, this case leads to surface meshing problems.*

20.4 Stiffness matrix and right-hand side

In this section, we firstly give the general expressions of the stiffness matrix and the right-hand side associated with the problem presented here. We give the corresponding element quantities (at the level of one element) and we show the case of an affine $T3$ triangle and for an affine $T6$ triangle together with an isoparametric $T6$ triangle. Then we discuss some specific questions regarding the information that must be stored or found in the mesh so as to make it possible to obtain these quantities. We also look at the quadrature formula and the way in which the global quantities are obtained by *assembling* the element quantities thus obtained.

20.4.1 General expression of a stiffness matrix

The stiffness matrix results from the contribution of $a(u_h, v_h)$ of Equation (20.1) through the two terms $-div(k_i \nabla u)$ and $g_i u$. After a Green formula, we have

$$a_h(u_h, v_h) = \int_{\Omega_i} k_i \nabla u_h \nabla v_h + \int_{\Gamma_i} g_i u_h v_h$$

and thus (omitting index i)

$$a_h(u_h, v_h) = \sum_{K \in \mathcal{T}} \int_K k \nabla u_h \nabla v_h + \sum_{K \in \mathcal{T}} \int_{K \cap \Gamma} g u_h v_h \,.$$

Apart from the coefficients, the quantities we are interested in are the restrictions over the elements of the various functions of the above relationship. For convenience we denote $[\mathcal{K}]$ the $d \times d$ matrix with coefficients k. We now consider $(u_h)/K$, the restriction over K of u_h. Following the approximation we have $(u_h)/K = \sum_i \phi_i(u)\, p_i$ where $\phi_i(u)$ is the degree of freedom i associated with function u_h and p_i is the basis polynomial i of space P_K (note that v_h is similarly considered).

With $[\mathcal{P}]$ the row $[p_1, p_2, ..., p_N]$ and $\{u_{i,K}\}$ the vector of the degrees of freedom in K, the above expression is conveniently written as

$$\{(u_h)/K\} = [\mathcal{P}]\,\{u_{i,K}\}\,.$$

With this concise notation, we have

$$\nabla(u_h)/K = {}^t\{\frac{\partial u_h/K}{\partial x}, \frac{\partial u_h/K}{\partial y}, \frac{\partial u_h/K}{\partial z}\}$$

which is simply expressed by

$$\{\nabla(u_h)/K\} = [\mathcal{DP}]\,\{u_{i,K}\}\,,$$

where

$$[\mathcal{DP}] = \begin{bmatrix} \frac{\partial p_1}{\partial x} & \frac{\partial p_2}{\partial x} & \cdots & \frac{\partial p_N}{\partial x} \\ \frac{\partial p_1}{\partial y} & \frac{\partial p_2}{\partial y} & \cdots & \frac{\partial p_N}{\partial y} \\ \frac{\partial p_1}{\partial z} & \frac{\partial p_2}{\partial z} & \cdots & \frac{\partial p_N}{\partial z} \end{bmatrix}$$

Thus, using these quantities, we have

$$\int_K k\nabla u_h \nabla v_h = {}^t\{v_{i,K}\} \int_K {}^t[\mathcal{DP}][\mathcal{K}][\mathcal{DP}]dK \ \{u_{i,K}\}$$

$$\text{and} \int_{\partial K} g u_h v_h = {}^t\{v_{i,K}\} \int_{\partial K} {}^t[\mathcal{P}]g[\mathcal{P}]d\partial K \ \{u_{i,K}\},$$

where $\int_K {}^t[\mathcal{DP}][\mathcal{K}][\mathcal{DP}]dK + \int_{\partial K} {}^t[\mathcal{P}]g[\mathcal{P}]d\partial K$ forms $[\mathcal{A}_K]$ the so-called *elementary stiffness matrix* of element K.

As previously indicated, this matrix is actually computed by means of the corresponding reference element $\hat{K}$. We have, following (20.7), $p_i = \hat{p}_i \circ F_K^{-1}$, thus

$$[\mathcal{P}] = [\hat{\mathcal{P}}] \text{ and } [\mathcal{DP}] = [\mathcal{DF}^{-1}][\hat{\mathcal{DP}}]$$

hold where $[\mathcal{DF}]$ denotes the derivatives of F_K. Using this yields :

$$[\mathcal{A}_K] = \int_{\hat{K}} {}^t[\hat{\mathcal{DP}}]\,{}^t[\mathcal{DF}^{-1}][\mathcal{K}][\mathcal{DF}^{-1}][\hat{\mathcal{DP}}]\mathcal{J}d\hat{K} + \int_{\partial\hat{K}} {}^t[\hat{\mathcal{P}}]g[\hat{\mathcal{P}}]\mathcal{J}_\Gamma d\partial\hat{K},$$

with $\mathcal{J}$ the Jacobian of F_K, *i.e.*, $\mathcal{J} = det(\mathcal{DF})$ and $\mathcal{J}_\Gamma$ the corresponding term on Γ, *i.e.*, we have $d\partial K = \mathcal{J}_\Gamma d\partial\hat{K}$. In terms of computation, the above integrals can be obtained directly (exact quadrature) or after a numerical quadrature based on some quadrature nodes (and coefficients). In this case, an integral is replaced by a summation and we need to evaluate the quantities of interest at the nodes of the quadrature formula.

Having the elementary matrices ready, we can obtained the global stiffness matrix $[\mathcal{A}]$. As can be easily established we have

$$[\mathcal{A}] = \sum_{K\in\mathcal{T}} {}^t[\mathcal{B}_K][\mathcal{A}_K][\mathcal{B}_K],$$

where $[\mathcal{B}_K]$ ensures the correspondence between the local node numbering (*i.e.*, when element K is considered) and the global node numbering (when all the mesh is considered). Thus $[\mathcal{B}_K]$ gives the correspondence between the local and the global numbering and enables us to assemble[9] the global matrix by merging in it the contributions due to the local matrices (see below).

20.4.2 General expression of a right-hand side

The right-hand side results from the contribution of $f(v_h)$ of Equation (20.1) through the term over the domain and that due to the boundary. We have,

$$f(v_h) = \int_{\Omega_i} F_i v_h + \int_{\Gamma_i} f_i v_h$$

and thus (omitting index i),

$$f(v_h) = \sum_{K\in\mathcal{T}} \int_K F v_h + \sum_{K\in\mathcal{T}} \int_{K\cap\Gamma} f v_h,$$

[9] Notice that some solution methods do not require the explicit construction of the global matrix.

or again

$$f(v_h) = \sum_{K\in\mathcal{T}} {}^t\{v_{i,K}\}\{\int_K {}^t[\mathcal{P}]F dK + \int_{K\cap\Gamma} {}^t[\mathcal{P}]f d\partial K\}.$$

The elementary contribution, the so-called *elementary right-hand side* is composed of the two terms $\int_K {}^t[\mathcal{P}]F dK + \int_{K\cap\Gamma} {}^t[\mathcal{P}]f d\partial K$. It is denoted by $[\mathcal{RHS}_K]$. The calculation is performed over $\hat{K}$:

$$[\mathcal{RHS}_K] = \int_{\hat{K}} {}^t[\mathcal{P}]F\mathcal{J}d\hat{K} + + \int_{\partial\hat{K}} {}^t[\mathcal{P}]f\mathcal{J}_\Gamma d\partial\hat{K}.$$

As before, using the matrix $[\mathcal{B}_K]$ allows us to obtain the global right-hand side :

$$[\mathcal{RHS}] = \sum_{K\in\mathcal{T}} {}^t[\mathcal{B}_K][\mathcal{RHS}_K].$$

Remark 20.8 *Elementary (global) quantities other than the stiffness matrix or the right-hand side, namely, the mass matrix (in an evolution problem), etc., can be obtained in the same way.*

20.4.3 About the $T3$ triangle

To demonstrate the above discussion, we look at the $T3$ triangle in two dimensions. In this example, we consider $d = 2$ and $N = 3$ while the basis polynomials are $[\hat{\mathcal{P}}] = \{1 - \hat{x} - \hat{y}\ , \hat{x}\ , \hat{y}\}$. Then, we have :

$$[\hat{\mathcal{DP}}] = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

Also $F_K = \{F_K^i\}_{i=1,2} \in (P^1)^2$. Then, if a_i stands for node i of K and (x_i, y_i) denotes its two coordinates, we define $a_{ij} = a_i - a_j$ and similar expressions for x_{ij} and y_{ij} and since

$$F_K(\hat{x}, \hat{y}) = (1 - \hat{x} - \hat{y})\{a_1\} + \hat{x}\{a_2\} + \hat{y}\{a_3\},$$

we have

$$F_K(\hat{x}, \hat{y}) = \{a_1\} + \hat{x}\,\{a_{21}\} + \hat{y}\,\{a_{31}\}.$$

Thus,

$$[\mathcal{DF}] = \begin{bmatrix} x_{21} & y_{21} \\ x_{31} & y_{31} \end{bmatrix}$$

and

$$\mathcal{J} = det(\mathcal{DF}) = x_{21}y_{31} - x_{31}y_{21}$$

$$[\mathcal{DF}]^{-1} = \frac{1}{\mathcal{J}} \begin{bmatrix} y_{31} & -y_{21} \\ -x_{31} & x_{21} \end{bmatrix}.$$

Now, we have all the ingredients ready for the stiffness matrix calculation. Since the latter is symmetric in our case, we just write its lower part. The integral over $\hat{K}$ leads to the term[10] :

$$[\mathcal{A}_K] = \frac{k}{\mathcal{J}} \int_{\hat{K}} \begin{bmatrix} y_{23}y_{23} + x_{23}x_{23} & \dots & \dots \\ y_{31}y_{23} + x_{31}x_{23} & y_{31}y_{31} + x_{31}x_{31} & \dots \\ -y_{21}y_{23} - x_{21}x_{23} & -y_{21}y_{31} - x_{21}x_{31} & y_{21}y_{21} + x_{21}x_{21} \end{bmatrix} d\hat{K},$$

which, since $\int_{\hat{K}} d\hat{K} = \frac{1}{2}$, reduces to

$$[\mathcal{A}_K] = \frac{k}{2\,\mathcal{J}} \begin{bmatrix} y_{23}y_{23} + x_{23}x_{23} & \dots & \dots \\ y_{31}y_{23} + x_{31}x_{23} & y_{31}y_{31} + x_{31}x_{31} & \dots \\ -y_{21}y_{23} - x_{21}x_{23} & -y_{21}y_{31} - x_{21}x_{31} & y_{21}y_{21} + x_{21}x_{21} \end{bmatrix}.$$

The integral over $\hat{\partial} K$ leads to boundary contributions if $\hat{\partial} K$ is a boundary edge where g acts. Let us assume that edge a_1a_2 is such a boundary edge, then the corresponding contribution is (g being assumed to be constant per element edge)

$$[\mathcal{A}_K] = g\,\mathcal{J}_\Gamma \int_0^1 \begin{bmatrix} (1-\hat{x})^2 & \dots & \dots \\ \hat{x}(1-\hat{x}) & \hat{x}^2 & \dots \\ 0 & 0 & 0 \end{bmatrix} d\hat{x}.$$

The element of integration is $\mathcal{J}_\Gamma d\hat{x}$. Indeed, for this edge, we have $a = a_1 + \hat{x}a_{21}$ where a is a point on edge a_1a_2. Then we have $x = x_1 + \hat{x}x_{21}$ and $y = y_1 + \hat{x}y_{21}$. Then

$$d\partial K = \sqrt{dx^2 + dy^2} = \sqrt{x_{21}^2 d\hat{x}^2 + y_{21}^2 d\hat{x}^2} = \sqrt{x_{21}^2 + y_{21}^2}\, d\hat{x},$$

and finally, if $\mathcal{J}_\Gamma = \sqrt{x_{21}^2 + y_{21}^2}$ we have $d\partial K = \mathcal{J}_\Gamma d\hat{x}$. Using an exact quadrature leads to

$$[\mathcal{A}_K] = g\,\mathcal{J}_\Gamma \begin{bmatrix} \frac{1}{3} & \dots & \dots \\ \frac{1}{6} & \frac{1}{3} & \dots \\ 0 & 0 & 0 \end{bmatrix}.$$

Similarly, provided they are in the adequate part of the boundary, the two other edges contribute in the same way. We have $\mathcal{J}_\Gamma = \sqrt{x_{32}^2 + y_{32}^2}$ (edge a_2a_3) or $\mathcal{J}_\Gamma = \sqrt{x_{31}^2 + y_{31}^2}$ (edge a_3a_1) and the corresponding contributions are successively

$$[\mathcal{A}_K] = g\,\mathcal{J}_\Gamma \begin{bmatrix} 0 & \dots & \dots \\ 0 & \frac{1}{3} & \dots \\ 0 & \frac{1}{6} & \frac{1}{3} \end{bmatrix}$$

and

$$[\mathcal{A}_K] = g\,\mathcal{J}_\Gamma \begin{bmatrix} \frac{1}{3} & \dots & \dots \\ 0 & 0 & \dots \\ \frac{1}{6} & 0 & \frac{1}{3} \end{bmatrix}.$$

[10] The conductivity is assumed to be isotropic (and constant per element), then $[\mathcal{K}]$ acts through the single coefficient k in all matrix coefficients. The general case where $[\mathcal{K}]$ is not isotropic results in a similar calculation provided several different values of k are employed in the appropriate terms.

Now the complete elementary stiffness matrix is the sum of the above contributions.

We turn now to the right-hand side calculation. For the term over K, we consider a quadrature formula based on the three element vertices, then, using $\hat{K}$, we have

$$\int_{\hat{K}} {}^t[\mathcal{P}] F \mathcal{J} d\hat{K} = \frac{\mathcal{J} F}{6} \sum_{l=1}^{3} {}^t[\mathcal{P}(\hat{x}_l, \hat{y}_l)],$$

which leads to the term :

$$\frac{\mathcal{J} F}{6} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ or } \frac{\mathcal{J}}{6} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix},$$

depending on whether F is assumed to be constant per element or used via its values at the element vertices. The boundary term is simply :

$$f\mathcal{J}_\Gamma \int_0^1 \begin{bmatrix} (1-\hat{x}) \\ \hat{x} \\ 0 \end{bmatrix} d\hat{x} = f\mathcal{J}_\Gamma \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix} \text{ or } \mathcal{J}_\Gamma \begin{bmatrix} \frac{f_1}{2} \\ \frac{f_2}{2} \\ 0 \end{bmatrix},$$

for edge a_1a_2 with $\mathcal{J}_\Gamma = \sqrt{x_{21}^2 + y_{21}^2}$. In the first solution we have assumed f to be constant per edge and an exact integration while in the second solution we have used a quadrature (based on the two edge endpoints) and f_1 (resp. f_2) are the corresponding values of f. For the two other edges, in the case where they contribute, we obtain a similar contribution, for instance,

$$\mathcal{J}_\Gamma \begin{bmatrix} 0 \\ \frac{f_2}{2} \\ \frac{f_3}{2} \end{bmatrix},$$

for edge a_2a_3 with now : $\mathcal{J}_\Gamma = \sqrt{x_{32}^2 + y_{32}^2}$.

20.4.4 About the linear $T6$ triangle

This section deals only with the affine triangle, and the isoparametric case will be discussed in the next section. In this example, we have $d = 2$ and we consider the so-called $T6$ triangle as a finite element. For both triangles (affine or curved), we have $N = 6$ and :

$$[\hat{P}] = \{(1-\hat{x}-\hat{y})(1-2\hat{x}-2\hat{y}), \hat{x}(2\hat{x}-1), \hat{y}(2\hat{y}-1), 4\hat{x}(1-\hat{x}-\hat{y}), 4\hat{x}\hat{y}, 4\hat{y}(1-\hat{x}-\hat{y})\}.$$

Then $[\widehat{\mathcal{DP}}]$ is now :

$$[\widehat{\mathcal{DP}}] = \begin{bmatrix} -3+4\hat{x}+4\hat{y} & 4\hat{x}-1 & 0 & 4-8\hat{x}-4\hat{y} & 4\hat{y} & -4\hat{y} \\ -3+4\hat{x}+4\hat{y} & 0 & 4\hat{y}-1 & -4\hat{x} & 4\hat{x} & 4-4\hat{x}-8\hat{y} \end{bmatrix},$$

hence, $[\widehat{\mathcal{DP}}]$ is now a function of $\hat{x}$ and $\hat{y}$. For convenience, we write this fact as :

$$[\widehat{\mathcal{DP}}] = [\widehat{\mathcal{DP}}(\hat{x}, \hat{y})].$$

For this affine case, the mapping function is the same as for the $T3$ element, *i.e.*,

$$F_K(\hat{x},\hat{y}) = \{a_1\} + \hat{x}\,\{a_{21}\} + \hat{y}\,\{a_{31}\}\,.$$

Thus, $[\mathcal{DF}]$, $\mathcal{J}$ and $[\mathcal{DF}]^{-1}$ are as in the $T3$ element. The general expression of the stiffness matrix, for element K, is :

$$[\mathcal{A}_K] = \int_{\hat{K}} {}^t[\widehat{\mathcal{DP}}(\hat{x},\hat{y})]\,{}^t[\mathcal{DF}^{-1}][\mathcal{K}][\mathcal{DF}^{-1}][\widehat{\mathcal{DP}}(\hat{x},\hat{y})]\mathcal{J}d\hat{K} + \int_{\partial\hat{K}} {}^t[\hat{\mathcal{P}}]g[\hat{\mathcal{P}}]\mathcal{J}_\Gamma d\partial\hat{K}\,.$$

In the first term, ${}^t[\mathcal{DF}^{-1}][\mathcal{K}][\mathcal{DF}^{-1}]$ is constant. Indeed, if $[\mathcal{FKF}]$ stands for this 2×2 matrix, we have :

$$[\mathcal{FKF}] = \frac{k}{\mathcal{J}^2}\begin{bmatrix} y_{31}y_{31} + x_{31}x_{31} & -y_{21}y_{31} - x_{21}x_{31} \\ -y_{21}y_{31} - x_{21}x_{31} & y_{21}y_{21} + x_{21}x_{21} \end{bmatrix}.$$

Thus, the first term of the stiffness matrix is :

$$\int_{\hat{K}} {}^t[\widehat{\mathcal{DP}}(\hat{x},\hat{y})][\mathcal{FKF}][\widehat{\mathcal{DP}}(\hat{x},\hat{y})]\mathcal{J}d\hat{K}\,,$$

and, although it is tedious, an exact integration can be used. The terms we have to compute are of the form :

$$\int_{\hat{K}} \frac{\partial p_i}{\partial x}\frac{\partial p_j}{\partial x}d\hat{K}\,,\ \int_{\hat{K}} \frac{\partial p_i}{\partial x}\frac{\partial p_j}{\partial y}d\hat{K} \text{ and } \int_{\hat{K}} \frac{\partial p_i}{\partial y}\frac{\partial p_j}{\partial y}d\hat{K}\,,$$

for $i = 1,6$ and $j = 1,6$ while we have :

$$\int_{\hat{K}} \dots\, d\hat{K} = \int_{\hat{x}=0}^{1}\int_{\hat{y}=0}^{1-\hat{x}} \dots\, d\hat{x}d\hat{y}\,.$$

Thus we need to compute 108 integrals of the above type. For example, we have :

$$\int_{\hat{K}} \frac{\partial p_1}{\partial x}\frac{\partial p_1}{\partial x}d\hat{K} = \frac{1}{2}\,,\ \int_{\hat{K}} \frac{\partial p_1}{\partial x}\frac{\partial p_2}{\partial x}d\hat{K} = \frac{1}{6}\,,\dots$$

On completion, we easily find the contribution of K to the stiffness matrix.

To complete this matrix, we need to compute the second part of the general expression. Let us examine the case where edge a_1a_2 contributes in this boundary contribution. First, the element of integration is as it was for the $T3$ triangle. Indeed, we have $d\partial K = \mathcal{J}_\Gamma d\partial\hat{K}$ where $\mathcal{J}_\Gamma$ depends on the edge under consideration. Then, we obtain (omitting symbol ^)

$$g\,\mathcal{J}_\Gamma \int_0^1 \begin{bmatrix} (1-x)^2(1-2x)^2 & \dots & \dots & \dots & \dots & \dots \\ -x(1-x)(1-2x)^2 & x^2(2x-1)^2 & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots \\ 4x(1-x)^2(1-2x) & 4x^2(1-x)(2x-1) & 0 & 16x^2(1-x)^2 & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} dx\,,$$

which, with $\mathcal{J}_\Gamma = \sqrt{x_{21}x_{21} + y_{21}y_{21}}$ and since an exact quadrature can be used, reduces to :

$$g\,\mathcal{J}_\Gamma \begin{bmatrix} \frac{2}{15} & \dots & \dots & \dots & \dots & \dots \\ -\frac{1}{30} & \frac{2}{15} & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots \\ \frac{1}{15} & \frac{1}{15} & 0 & \frac{8}{15} & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Now the complete elementary stiffness matrix is the sum of the two above contributions. Note that boundary contribution of edges other than that used here are dealt with in the same way as for the $T3$ element. For edge a_2a_3, we have : $\mathcal{J}_\Gamma = \sqrt{x_{32}x_{32} + y_{32}y_{32}}$ and the possible contribution is :

$$g\,\mathcal{J}_\Gamma \begin{bmatrix} 0 & \dots & \dots & \dots & \dots & \dots \\ 0 & \frac{2}{15} & \dots & \dots & \dots & \dots \\ 0 & -\frac{1}{30} & \frac{2}{15} & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \dots \\ 0 & \frac{1}{15} & \frac{1}{15} & 0 & \frac{8}{15} & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

while edge a_1a_3 possibly contributes through the term :

$$g\,\mathcal{J}_\Gamma \begin{bmatrix} \frac{2}{15} & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & \dots \\ -\frac{1}{30} & 0 & \frac{2}{15} & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots. \\ \frac{1}{15} & 0 & \frac{1}{15} & 0 & 0 & \frac{8}{15} \end{bmatrix},$$

with $\mathcal{J}_\Gamma = \sqrt{x_{31}x_{31} + y_{31}y_{31}}$ as can be easily seen.

The right-hand side comprises two possible terms. The term over K is :

$$\frac{\mathcal{J}F}{6}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \text{ or } \frac{\mathcal{J}}{6}\begin{bmatrix} 0 \\ 0 \\ 0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix},$$

if the quadrature used is based on the three mid-edge vertices and F is assumed to be constant per element or used via its three relevant values. The boundary term (for instance, for edge a_1a_2) is computed using a quadrature based on the edge endpoints and its midpoint, thus we have (with evident notation for f.) :

$$\frac{\mathcal{J}_\Gamma}{6}\begin{bmatrix} f_1 \\ f_2 \\ 0 \\ 4\,f_{12} \\ 0 \\ 0 \end{bmatrix},$$

with $\mathcal{J}_\Gamma = \sqrt{x_{21}x_{21} + y_{21}y_{21}}$ as above.

20.4.5 About the isoparametric $T6$ triangle

We now turn to an isoparametric $T6$ element. Basically the same approach can be retained. In specific $[\mathcal{P}]$, the basis polynomials, are those of the affine case. However, since $F_K(\hat{x},\hat{y}) = \sum_{i=1}^{6} p_i(\hat{x},\hat{y})\,\{a_i\}$, the matrix $[\mathcal{DF}]$ is no longer constant over K. Thus both $\mathcal{J}$ and $[\mathcal{DF}^{-1}]$ are functions of $\hat{x}$ and $\hat{y}$. Consequently a numerical quadrature must be employed and we obtain, as a K contribution :

$$\sum_{l=1}^{q} \omega_l\, {}^t[\hat{\mathcal{DP}}(\hat{x}_l,\hat{y}_l)][\mathcal{FKF}(\hat{x}_l,\hat{y}_l)][\hat{\mathcal{DP}}(\hat{x}_l,\hat{y}_l)]\mathcal{J}(\hat{x}_l,\hat{y}_l)\,.$$

Using the three mid-edges as quadrature nodes we have to compute $[\mathcal{FKF}]$, $[\hat{\mathcal{DP}}]$ and $\mathcal{J}$ at these nodes. Following :

$$F_K(\hat{x},\hat{y}) = \sum_{i=1}^{6} p_i(\hat{x},\hat{y})\,\{a_i\}\,,$$

we find :

$$[\mathcal{DF}] = \begin{bmatrix} -3+4x+4y & 4x-1 & 0 & 4-8x-4y & 4y & -4y \\ -3+4x+4y & 0 & 4y-1 & -4x & 4x & 4-4x-8y \end{bmatrix}\{a_i\}\,,$$

then, for the node $(\frac{1}{2},0)$ we find :

$$[\mathcal{DF}(\frac{1}{2},0)] = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -2 & 2 & 2 \end{bmatrix}\{a_i\}\,,$$

which reduces to :

$$[\mathcal{DF}(\frac{1}{2},0)] = \begin{bmatrix} -x_1+x_2 & -y_1+y_2 \\ -x_1-x_3-2x_4+2x_5+2x_6 & -y_1-y_3-2y_4+2y_5+2y_6 \end{bmatrix}$$

$$[\mathcal{DF}(\frac{1}{2},0)] = \begin{bmatrix} x_{21} & y_{21} \\ 2x_{54}+x_{61}+x_{63} & 2y_{54}+y_{61}+y_{63} \end{bmatrix}$$

and

$$[\mathcal{DF}(\frac{1}{2},0)^{-1}] = \begin{bmatrix} 2y_{54}+y_{61}+y_{63} & -y_{21} \\ -(2x_{54}+x_{61}+x_{63}) & x_{21} \end{bmatrix}\,,$$

whose determinant is $\mathcal{J}(\frac{1}{2},0)$. Similarly, and left as an exercise, one can obtain $[\mathcal{DF}(\frac{1}{2},\frac{1}{2})^{-1}]$ and $\mathcal{J}(\frac{1}{2},\frac{1}{2})$ as well as $[\mathcal{DF}(0,\frac{1}{2})^{-1}]$ and $\mathcal{J}(0,\frac{1}{2})$. From these quantities, we have $[\mathcal{FKF}]$ ready at the three quadrature nodes. Similarly, we compute the $[\hat{\mathcal{DP}}]$'s involved in the formula. We find successively :

$$[\hat{\mathcal{DP}}(\frac{1}{2},0)] = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -2 & 2 & 2 \end{bmatrix}\,,$$

$$[\hat{\mathcal{DP}}(\frac{1}{2},\frac{1}{2})] = \begin{bmatrix} 1 & 1 & 0 & -2 & 2 & -2 \\ 1 & 0 & 1 & -2 & 2 & -2 \end{bmatrix}\,,$$

$$[\hat{\mathcal{DP}}(0,\frac{1}{2})] = \begin{bmatrix} -1 & -1 & 0 & 2 & 2 & -2 \\ -1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}\,.$$

Now using the above $[\hat{\mathcal{DP}}(\hat{x}_l,\hat{y}_l)]$'s, we have all the ingredients required to compute the contribution of element K to the stiffness matrix.

Let us now consider the contribution of an edge part of the boundary which contributes to the stiffness matrix. We want to use a quadrature whose nodes are the edge endpoints along with the edge midpoints. Let a_1a_2 be this edge. If a stands for a point on this edge, we have

$$a = F_K(\hat{x},\hat{y}) = (1-\hat{x})(1-2\hat{x})a_1 + \hat{x}(2\hat{x}-1)a_2 + 4\hat{x}(1-\hat{x})a_4\,,$$

and, for instance,

$$x = (1-\hat{x})(1-2\hat{x})x_1 + \hat{x}(2\hat{x}-1)x_2 + 4\hat{x}(1-\hat{x})x_4\,,$$

thus,

$$dx = \{(4\hat{x}-3)x_1 + (4\hat{x}-1)x_2 + \hat{x}(4-8\hat{x})x_4\,\}d\hat{x}\,,$$

and similar expressions hold for y and dy. For $\hat{x}=0$, this reduces to :

$$dx = (-3x_1 - x_2 + 4x_4)d\hat{x} = (3x_{41} + x_{42})d\hat{x}$$

and we have, in this first quadrature node,

$$\mathcal{J}_\Gamma = \mathcal{J}_\Gamma(0) = \sqrt{(3x_{41}+x_{42})^2 + (3y_{41}+y_{42})^2}\,.$$

The same calculation for the midpoint $\hat{x} = \frac{1}{2}$ and for the other endpoint, $\hat{x} = 1$, leads to a similar expression for the corresponding $\mathcal{J}_\Gamma$. We obtain successively :

$$\mathcal{J}_\Gamma(\frac{1}{2}) = \sqrt{x_{21}^2 + y_{21}^2} \text{ and}$$

$$\mathcal{J}_\Gamma(1) = \sqrt{(x_{14}+3x_{24})^2 + (y_{14}+3y_{24})^2}\,.$$

Now, the boundary contribution is obtained based on these three quadrature nodes. We have :

$$\int_{\partial\hat{K}} {}^t[\mathcal{P}]g[\mathcal{P}]\mathcal{J}_\Gamma d\hat{x} = \sum_{l=1}^{3} \omega_l\, {}^t[\mathcal{P}(\hat{x}_l,0)]g(\hat{x}_l,0)[\mathcal{P}(\hat{x}_l,0)]\mathcal{J}_\Gamma(\hat{x}_l,0)\,,$$

with $\omega_l = (\frac{1}{6},\frac{4}{6},\frac{1}{6})$, this leads to :

$$[\mathcal{A}_K] = \frac{1}{6}\begin{bmatrix} g_1\mathcal{J}_\Gamma(0) & \dots & \dots & \dots & \dots & \dots \\ 0 & g_2\mathcal{J}_\Gamma(1) & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & 4g_{12}\mathcal{J}_\Gamma(\frac{1}{2}) & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\,.$$

The contribution of the two other edges, if relevant, is obtained in a similar way. We find successively :

$$[\mathcal{A}_K] = \frac{1}{6}\begin{bmatrix} 0 & \dots & \dots & \dots & \dots & \dots \\ 0 & g_2\mathcal{J}_\Gamma(0) & \dots & \dots & \dots & \dots \\ 0 & 0 & g_3\mathcal{J}_\Gamma(1) & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 4g_{23}\mathcal{J}_\Gamma(\frac{1}{2}) & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where

$$\mathcal{J}_\Gamma(0) = \sqrt{(3x_{52}+x_{53})^2+(3y_{52}+y_{53})^2}\,.$$

$$\mathcal{J}_\Gamma(\frac{1}{2}) = \sqrt{x_{32}^2+y_{32}^2} \text{ and}$$

$$\mathcal{J}_\Gamma(1) = \sqrt{(x_{25}+3x_{35})^2+(y_{25}+3y_{35})^2}\,,$$

and, for the last edge :

$$[\mathcal{A}_K] = \frac{1}{6}\begin{bmatrix} g_1\mathcal{J}_\Gamma(0) & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & \dots \\ 0 & 0 & g_3\mathcal{J}_\Gamma(1) & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 4g_{13}\mathcal{J}_\Gamma(\frac{1}{2}) \end{bmatrix},$$

where now,

$$\mathcal{J}_\Gamma(0) = \sqrt{(3x_{61}+x_{63})^2+(3y_{61}+y_{63})^2}\,.$$

$$\mathcal{J}_\Gamma(\frac{1}{2}) = \sqrt{x_{31}^2+y_{31}^2} \text{ and}$$

$$\mathcal{J}_\Gamma(1) = \sqrt{(x_{16}+3x_{36})^2+(y_{16}+3y_{36})^2}\,.$$

To complete this example, we have to compute the right-hand side. The term over K is computed based on a quadrature formula using as nodes the three vertices and the three edge midpoints. Thus we have :

$$[\mathcal{RHS}_K] = \int_{\hat{K}} {}^t[\mathcal{P}]F\mathcal{J}d\hat{K} = \frac{1}{6}\{\sum_{l=1}^{q} {}^t[\mathcal{P}(\hat{x}_l,\hat{y}_l)]F(\hat{x}_l,\hat{y}_l)\mathcal{J}(\hat{x}_l,\hat{y}_l)\}\,,$$

which, with evident notations, reduces to :

$$[\mathcal{RHS}_K] = \frac{1}{6}\begin{bmatrix} F_1\mathcal{J}(0,0) \\ F_2\mathcal{J}(1,0) \\ F_3\mathcal{J}(0,1) \\ F_{12}\mathcal{J}(\frac{1}{2},0) \\ F_{23}\mathcal{J}(\frac{1}{2},\frac{1}{2}) \\ F_{13}\mathcal{J}(0,\frac{1}{2}) \end{bmatrix}.$$

Nevertheless, the above formula is too poor and a more precise quadrature must be used. It is also based on the six nodes in K and, after [Glowinski-1973], it results in the following contribution :

$$[\mathcal{RHS}_K] = \frac{1}{360}\begin{bmatrix} 6F_1\mathcal{J}(0,0) - (F_2\mathcal{J}(1,0)+F_3\mathcal{J}(0,1)) - 4F_{23}\mathcal{J}(\frac{1}{2},\frac{1}{2}) \\ 6F_2\mathcal{J}(1,0) - (F_1\mathcal{J}(0,0)+F_3\mathcal{J}(0,1)) - 4F_{13}\mathcal{J}(0,\frac{1}{2}) \\ 6F_3\mathcal{J}(0,1) - (F_1\mathcal{J}(0,0)+F_2\mathcal{J}(1,0)) - 4F_{12}\mathcal{J}(\frac{1}{2},0) \\ 32F_{12}\mathcal{J}(\frac{1}{2},0) + 16\,(F_{23}\mathcal{J}(\frac{1}{2},\frac{1}{2}) + F_{13}\mathcal{J}(0,\frac{1}{2})) - 4F_3\mathcal{J}(0,1) \\ 32F_{23}\mathcal{J}(\frac{1}{2},\frac{1}{2}) + 16\,(F_{13}\mathcal{J}(0,\frac{1}{2}) + F_{12}\mathcal{J}(\frac{1}{2},0)) - 4F_1\mathcal{J}(0,0) \\ 32F_{13}\mathcal{J}(0,\frac{1}{2}) + 16\,(F_{12}\mathcal{J}(\frac{1}{2},0) + F_{23}\mathcal{J}(\frac{1}{2},\frac{1}{2})) - 4F_2\mathcal{J}(1,0) \end{bmatrix}.$$

The R-H-S boundary contributions, if any, are computed using a quadrature formula based on the two edge endpoints and the edge midpoint. For edge a_1a_2, we have :

$$[\mathcal{RHS}_K] = \int_{\partial\hat{K}} {}^t[\mathcal{P}]f\mathcal{J}_\Gamma d\partial\hat{K}$$

$$= \frac{1}{6}\{{}^t[\mathcal{P}(0,0)]f_1\mathcal{J}_\Gamma(0) + 4\,{}^t[\mathcal{P}(\frac{1}{2},0)]f_{12}\mathcal{J}_\Gamma(\frac{1}{2}) + {}^t[\mathcal{P}(1,0)]f_2\mathcal{J}_\Gamma(1)\}\,,$$

which reduces to :

$$[\mathcal{RHS}_K] = \frac{1}{6}\begin{bmatrix} f_1\mathcal{J}_\Gamma(0) \\ f_2\mathcal{J}_\Gamma(1) \\ 0 \\ 4\,f_{12}\mathcal{J}_\Gamma(\frac{1}{2}) \\ 0 \\ 0 \end{bmatrix},$$

where the $\mathcal{J}_\Gamma$'s are identical to those used for the boundary contribution of the same edge to the stiffness matrix. Other edge contributions are obtained in the same way.

Exercise 20.1 *Based on the previous examples, compute the stiffness matrix and the right-hand side for a P.D.E. modeling a two-dimensional elasticity problem (provided a small deformation problem with planar stress). Hint : we face a problem with two degrees of freedom per node where can be easily expressed using the previous quantities (polynomials, mapping, Jacobian, etc., see [Ciarlet-1986], [Ciarlet-1988]).*

20.4.6 Element quantities and data structure

In the previous finite element examples we have seen how to compute a stiffness matrix and a right-hand side. Based on these calculations, some practical details about what a mesh data structure could be have been implicitly introduced. Indeed, a mesh data structure must store any information which is required to make the desired calculation possible. In this respect, we have used various information, including :

- the vertex (node) numbering,
- the vertex (node) coordinates,
- the classification of the mesh elements in terms of sub-domains (or materials),
- the classification of the mesh boundaries (mesh edges in two dimensions) in terms of the part of the boundary in which they are located,
- the classification of the mesh vertices (nodes) with the same objective (in our example, to easily find the nodes subjected to a Dirichlet condition).

Numbering and coordinates are natural entries. The problem of membership is a mesh entity classification problem .

Provided with this information, both relevant calculations are possible while using adequate physical coefficients. In what follows, we give some details about these issues (restricting ourselves to a two-dimensional problem).

Vertex (node) coordinates. The node coordinates are obviously used when computing such or such elementary quantities. Note that provided with the coordinates of the three vertices of the affine $T6$ triangle, it is easy to retrieve the coordinates of the mid-edge nodes. Thus, it is not strictly necessary to store these values in the mesh data structure. On the other hand, the 6 pairs of coordinates must be stored for an isoparametric $T6$ triangle if we want to avoid calculating on the fly which would also require a geometric description of the domain boundaries.

Vertex (node) numbering. The (global) node numbering allows us to complete the global matrix and the global right-hand side of the discrete system (see below).

Classification of the mesh entities. The point is to know to which material belongs a given element. In this way, it is possible to assign the right material coefficient to this element. Similarly, we have to know which part of the boundary an element edge belongs to (provided it is not internal). Based on this knowledge, the contribution to the stiffness matrix or to the right-hand side is possible while assigning appropriate coefficients.

It is then desirable to associate an integer value[11] with the elements in the mesh (serving as material number or material pointer) and, similarly, an integer value (pointer) with all faces, edges and nodes for each element [Shephard-1988], [Beall,Shephard-1997].

Note that this data about elements, faces, edges and vertices makes it possible to easily find the information at the node level. Indeed, if a node is also a vertex, there is no problem and if a node is not a vertex it is only necessary to know where it is located. If it belongs to an edge, then its attribute for classification (the above integer) is that of this edge, etc.

[11] or pointer or any other equivalent manner.

Curved boundaries. Following the previous paragraph and based on the same classification, it is easy to see whether or not an edge belongs to a curved boundary. In such a case, the relevant boundary is identified and the node(s), if any, that must be constructed are created on this entity.

Mesh data structure. After the above discussion, the mesh data structure must be designed with these objectives in mind. In this respect, return to Chapter 1 and see the section about the external structure.

20.4.7 About integrals and quadratures

As seen above, it is sometimes possible, given some assumptions about the nature of the data used (about what are the polynomials, the complexity of the operators involved in the problem, the case where the data values are constant per element, for example), to compute exactly some of the integrals involved in the quantity sought. This is, nevertheless, not the case in general. Thus, making use of numerical scheme is generally required.

As seen and based on the degree of the functions in space P_K and on what needs to be computed, such or such formula must be retained. The objective is to make sure that the error due to the quadrature is "consistent" with regard to the interpolation error of the finite element itself.

Usually, the numerical scheme for quadrature is chosen in such a way as to exactly integrate a polynomial of degree n, where n is defined in accordance with the finite element, of the order of the expression resulting from the development of the operator and the nature of the data (for example, if the latter are assumed to be constant per entity (element, face, edge, etc.)).

20.4.8 The global system

As previously seen, assembling the contributions of the mesh elements takes the following form :

$$[\mathcal{A}] = \sum_{K \in \mathcal{T}} {}^t[\mathcal{B}_K][\mathcal{A}_K][\mathcal{B}_K]\,,$$

to complete the global matrix and :

$$[\mathcal{RHS}] = \sum_{K \in \mathcal{T}} {}^t[\mathcal{B}_K][\mathcal{RHS}_K]\,,$$

to obtain the global right-hand side of the system. In these expressions, matrix $[\mathcal{B}_K]$ related to element K is a boolean matrix (whose coefficients are 0 and 1), which makes it possible to merge the local contribution at the right place in the global quantity.

To specify what matrix $[\mathcal{B}_K]$ is, let us return to the above example with the simplest element, the $T3$ triangle. Thus, $[\mathcal{B}_K]$ is a $N \times M$ matrix where $N = 3$ is the number of degrees of freedom in the $T3$ triangle and M is the total number of degrees of freedom in the mesh[12]. All the coefficients of the matrix are null except three. Indeed we have $[\mathcal{B}_K]_{1,i_1} = 1$, $[\mathcal{B}_K]_{2,i_2} = 1$ and $[\mathcal{B}_K]_{3,i_3} = 1,$, if i_1, i_2 and i_3 are the global indices of the three degrees in K (*i.e.*, the degree with indices 1, 2 and 3 in this element).

From a practical point of view, the different matrices $[\mathcal{B}_K]$ do not really exist, a careful writing can be used which luckily avoids their construction (and their storage) while reflecting the mechanism they represent.

20.5 A few examples of popular finite elements

In the following list, we don't seek to be exhaustive, we just want to give several examples of finite elements representative of the various types of element that may be encountered. In this respect, examples of planar, surface and volume elements are given illustrated by different interpolation methods (Lagrange, Hermite, etc.) and different geometrical natures (straight element, curved element).

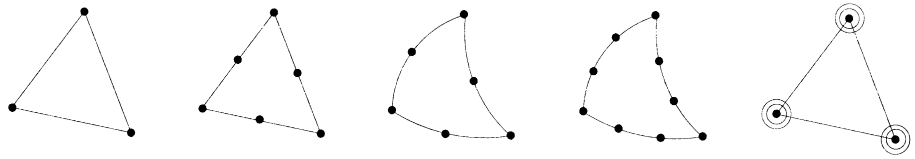


Figure 20.4: *Typical examples of triangular finite elements (in the plane). Element T3 Lagrange, the three node triangle, elements T6 Lagrange, the straight six node triangle and the isoparametric six node triangle, the Lagrange triangle with ten nodes and the Hermite triangle with 10 nodes.*

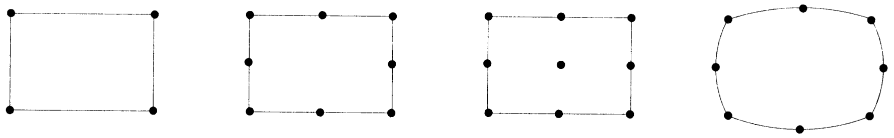


Figure 20.5: *Typical examples of quadrilateral finite element (in the plane). Affine Q4 and Q8 and isoparametric Q8 and Q9 Lagrange element, with respectively four, eight and nine nodes.*

[12] Including now the degrees corresponding to a Dirichlet condition.

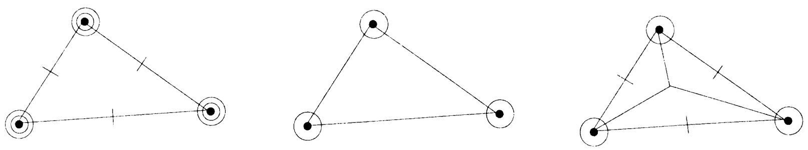


Figure 20.6: *Typical examples of triangular or quadrilateral finite elements (for surfaces, plates or shells).*

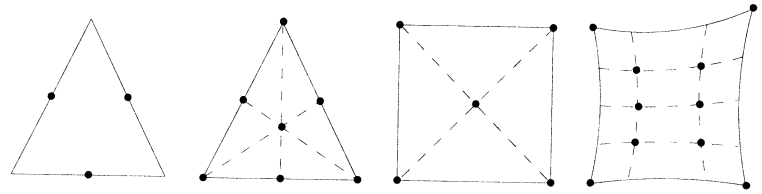


Figure 20.7: *Typical examples of triangular or quadrilateral hybrid finite elements where the nodes are less usual. From left to right, the nodes are the three mid-edge points (the vertices not being nodes), the nodes are the vertices and the intersection points of the two diagonals, the nodes are the vertices and the Gauss points.*

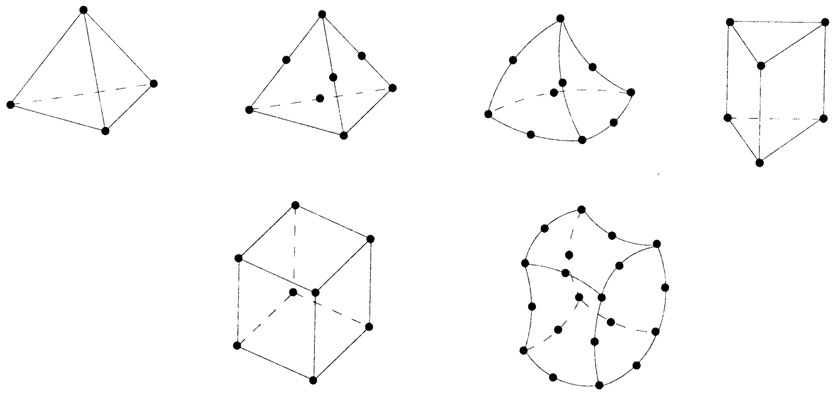


Figure 20.8: *Typical examples of volume finite elements, The T4 tet and the affine and isoparametric T10 tets. A pentahedral element with 10 nodes and an isoparametric hex with 20 nodes.*