Chapter 12

Curve modeling

Introduction

A curve can be defined using various categories of methods. Indeed, there are parametric, implicit or explicit curves. Using classical notations, a parametric curve is given by a function γ based on a parameter t. Then, given t in some interval, the curve is defined by $\gamma(t)$ where $\gamma(t) \in \mathbb{R}^d$, $d = 2$ or $d = 3$. Implicit curves in $\mathbb{R}^2$ are given via a relation like $f(x, y) = 0$ where x and y denote the coordinates. Explicit curves in $\mathbb{R}^2$ are defined by the pair $(x, y = f(x))$. In $\mathbb{R}^3$, non parametric curves are defined as the intersection of two surfaces.

In principle, functions f or γ depend on modeling parameters $(p_i)_{i \in [1,np]}$. These parameters can be defined in one of the following ways :

- through a direct definition of the design variables : with this approach, the user of a CAD system directly defines the values of the parameters p_i using a graphic interface. This way of processing is actually the simplest but it requires reasonably precise knowledge about the underlying models and their intrinsic variables,

- through the specification of the curve characteristics : the curve is defined by means of high level characteristics but the model, as completed by the CAD system, does not necessarily store this information. For instance, a circle can be defined while a NURBS type curve (see below) is generated inside the system. This type of curve specification includes the curve generation method based on a set of constraints,

- through an interpolation method : a set of points P_i and, for some of these methods, a set of derivatives at these points are supplied which form the parameters (the p_i's). Then, an interpolation procedure constructs a curve that passes through all of these points,

- through a smoothing technique : a set of points P_i (and, as above some derivatives for some methods) is supplied. Then a smoothing algorithm completes a curve that passes through the given points (or only some of them) while ensuring that a given criterion is minimized,

- through a combination of several of the above approaches.

★
★ ★

In this chapter we discuss the methods that are the most widely used in practice (for instance in CAD systems), namely a parametric definition. Implicit curves will be discussed in Chapter 16 while explicit curves are mostly of academic interest or are used only in some particular cases.

In the first section, we introduce the main ideas of the interpolation or smoothing based methods for parametric curve modeling. In the following sections, we describe the models that make use of a control polygon which form the basis of most curve definition systems.

Curves are defined from a set of points and various types of functions. First, a global definition can be constructed, which means that a single function is used to define the curve from P_0 to P_n, irrespective of the range of n. Second the curve from P_0 to P_n is defined using several segments whose junctions verify some regularity (in terms of continuity). In the first case, the curve passes through P_0 and P_n and passes through P_i, for $i = 1, n-1$, and we meet an interpolation method or doesn't pass through these points meaning that an extrapolation method or a smoothing based method is used.

Among interpolation methods we first find methods such as Lagrange interpolates using the control points as input data and Hermite interpolates where derivatives (tangents at the control points) are also involved. We can find other types of definitions such as the well-known Bézier method based on Bernstein polynomials of degree n and the rational Bézier method using rational polynomials also of degree n.

While nice in terms of regularity, such global definitions may lead to some problems of a different nature (complexity, unnecessary computational effort, oscillations, etc.). As a consequence, when n is large, methods with a low degree (such as 3 or 4) have been introduced. Then, whatever the value of n, a series of segments is defined where a low degree method is developed. We then obtain a nice local regularity and the issue is to make sure that the junctions between two consecutive segments is smooth enough to obtain sufficient global regularity. In this class of methods we can first use low degree Bézier (or similar) definitions or introduce the *Spline*-based methods leading to B-spline or more generally to NURBS-based methods.

The first sections aim at briefly introducing the methods that have been mentioned above. We don't claim to be exhaustive and for a complete view of curve definitions we refer the reader to the *ad-hoc* literature *i.e.*, [Mortenson-1985], [Bartels *et al.* 1987], [Bartels *et al.* 1988], [Léon-1991] or [Farin-1997] among many others. At the end of the chapter, some numerical problems regarding curve manipulation are discussed based on some comprehensive examples.

Notations. In what follows, t is the parameter, $\gamma(t)$ is the parametric expression of the curve we are interested in (as a curve is usually noted by Γ in the context of finite elements). If need be, we do not distinguish between $\gamma(t)$ (which for a given value of t is a point) and Γ (the whole curve).

12.1 Interpolation and smoothing techniques

As we are concerned with parametric curves, the function γ defining such a curve has the form :

$$\gamma(t) = \gamma(p_1, p_2, ..., p_{np}, t)\,,$$

where the p_i's are the parameters defining the shape of model γ (*i.e.*, that of curve Γ) and t is a real value parameter.

In the following we consider $n+1$ control points $P_0, P_1, ..., P_n$ which are supplied in $\mathbb{R}^d$. The problem is to define a curve using a function and these points. Then, various questions must be addressed including how to define these points and what type of functions must be constructed so as to obtain a suitable and easy to manipulate curve.

The purpose of any interpolation or smoothing method is to find the values of the parameters p_i involved in the definition, such that the resulting curve Γ is representative, in some sense, of the given set of points $P_0, P_1, ..., P_n$. At first, we are not interested in what the functions γ used to define the curve are. Then, for a specific interpolation or smoothing method (using a parameterization of the whole set of points), there exists a unique parameter that can be associated with any point in this set. The given points define a *polyline* and an interpolation or smoothing technique must complete a curve close to this line.

12.1.1 Parameterization of a set of points

The issue is to associate the parameters t_i with the points in the set P_i. At least, we must achieve a certain correspondence[1] between $\gamma(t_i)$ and P_i. Once the points have been sorted, the first assumption is that the t_i's are ordered in an increasing order :

$$t_i \leq t_{i+1}\,.$$

This being satisfied, any set of t_i's is valid *a priori*. Nevertheless, for simplicity and simple automatization, the most frequent choices correspond to :

- a uniform parameterization : the parameters are uniformly spaced in the given interval,

- a parameterization related to distances between the points : the parameters are then spaced in accordance with the length of the different segments $[P_i, P_{i+1}]$'s.

[1] Such a parameterization is not strictly required. The purpose is to find the np parameters. The t_i's are only additional parameters. If, in addition, there are more equations than unknowns, then the parameterization allows us to reduce the number of parameters by fixing some t_i.

Uniform parameterization. If the interval of parameters is $[t_{min}, t_{max}]$, then the t_i are defined following the rule :

$$t_i = t_{min} + \frac{i}{n} \times (t_{max} - t_{min})$$

Remark 12.1 *Notice that the curvilinear abscissa along the approached polyline does not follow, in general, the distribution of the t_i's. For instance, the parameter value $\frac{t_{min}+t_{max}}{2}$, the middle of the interval $[t_{min}, t_{max}]$ is not, in general, the parameter of the midpoint of this polyline.*

Parameterization conforming to the length ratios. The parameters t_i's can be defined in such a way as the ratio between to successive parameter values is exactly the ratio between the lengths of the two corresponding consecutive points :

$$\frac{t_{i+1} - t_i}{t_{max} - t_{min}} = \frac{||P_{i+1} - P_i||}{\sum_{j=0}^{n-1} ||P_{j+1} - P_j||} \quad ; \quad t_0 = t_{min}$$

where t_{min} and t_{max} are the bounds of the interval. In this case, the series of the t_i's can be obtained using the following algorithm :

Algorithm 12.1 *Parameterization following the length ratios.*

$t_0 = t_{min}$
$L = \sum_{j=0}^{n-1} ||P_{j+1} - P_j||$
FOR $i = 1$ to n
$\quad t_i = t_{i-1} + (t_{max} - t_{min}) \times \frac{||P_i - P_{i-1}||}{L}$
END FOR i.

Remark 12.2 *Taking these length ratios allows for a better match between the variation in curvilinear abscissa and that of the parameters but, however, does not lead to the proportional ratio between these two values : $\frac{ds}{dt} \neq cte$.*

12.1.2 Interpolation based methods

The interpolant properties of any curve definition method are related to the fact that the distance between the curve and any point P_i is zero. The best way to ensure this property is to impose that :

$$\gamma(p_1, p_2, ..., p_{np}, t_i) = P_i \ , \forall\, i \in [0, n] . \qquad (12.1)$$

Since the values of the t_i's must be chosen so as to be suitable for all the previously mentioned methods, we have $(n+1) \times d$ equations with np unknowns. Then we encounter three cases :

- $np \leq (n+1) \times d$ and the interpolation problem is *over-determined* : System (12.1) cannot be solved,
- $np = (n+1) \times d$ and the interpolation problem is well-posed : System (12.1) can be solved. However, since γ is non-linear with respect to the parameters p_i's, the system to be solved is a *non-linear* system,
- $np \geq (n+1) \times d$ and the interpolation problem is *under-determined* : System (12.1) must be completed with additional conditions, for instance, about tangencies, curvature values, etc.

Exercise 12.1 *Interpolate a circle from the following series of data points (when this makes sense) :*

- $\{(0,0); (5,8)\}$,
- $\{(0,0); (5,8); (3,1)\}$ *and*
- $\{(0,0); (5,8); (3,1); (-4,2)\}$.

Hint : in the two points case, fix one parameter t_1 or t_2 so as to define a specific circle (there is an infinity of circles passing through two points). With 3 points, the t_i's result from a non-linear system (which must be solved using an adequate technique, cf. Chapter 11). With 4 points, there is no longer an interpolation solution in most cases.

12.1.3 Smoothing based methods

Interpolation techniques are not suitable when the corresponding system is over-determined. In this case, smoothing techniques allow for the definition of a curve which is close, in some sense, to the set of points. This notion of a proximity is evaluated with respect to a criterion C :

$$C(p_1, p_2, ..., p_{np}, P_0, P_1, ..., P_n) .$$

The aim is then to minimize this criterion C for the set of values p_i's. To this end, we construct the system :

$$\frac{\partial C}{\partial p_i} = 0 .$$

In this system, there is still the same number of parameters and equations. Thus this system can be solved. However, it could be non-linear, meaning that appropriate methods must be used. The reader may refer to Section 11.4 where some methods suitable for this purpose are discussed.

Remark 12.3 (Linear regression by means of a least square method) *In numerous cases, the square of the distances is used in the definition of C :*

$$C(p_1, p_2, ..., p_{np}, P_0, P_1, ..., P_n) = \sum_{i=0}^{n} ||P_i - \gamma(p_1, p_2, ..., p_{np}, t_i)||^2 .$$

If the curve corresponding to this series of points is a line, γ is linear with its parameters. Thus, C is a polynomial of degree 2 and the resulting system is a linear system. Indeed, we turn to the system of the classical linear regression.

Remark 12.4 *When the parameters p_i's have been fully defined, the value of $C(p_1, p_2, ..., p_{np}, P_0, P_1, ..., P_n)$ is used to judge the quality of the curve in its approximation of the set of points.*

Exercise 12.2 *Generate a circle approximating the series of points in Exercise 12.1 when a least square criterion is prescribed.*

12.1.4 Combined methods

When the number of constraints, interpolation points, tangency conditions, curvatures, etc., is less than the number of parameters, one can combine a smoothing method with an interpolation technique. Then, the minimization of criterion C must be made under this interpolation constraint.

12.2 Lagrange and Hermite interpolation

12.2.1 Lagrange's interpolation scheme

After this general survey, we turn to the first class of interpolation methods, called the *Lagrange* type method. In this case, the control points are associated with a series of parameter values t_i, $(i = 0, n)$, where t_i corresponds to P_i. The modeling parameters, the p_i's, are the P_i's and the parameter t consists of the set of t_i's. Then the interpolation function :

$$\gamma(t) = \sum_{i=0}^{n} \phi_i(t) P_i \quad \text{with} \quad \phi_i(t) = \prod_{l=0}^{n} \frac{t - t_l}{t_i - t_l} \quad \text{for } l \neq i \qquad (12.2)$$

defines a curve, called the *Lagrange interpolate* of degree n. This curve is governed by the $n+1$ given points and *passes through* these points: in fact, since $\phi_i(t_j) = \delta_{ij}$ where δ_{ij} is the Kronecker delta, we have

$$\gamma(t_i) = P_i \, .$$

Remark 12.5 *The ϕ_i's form a basis of all polynomials of degree n. Moreover, they sum to 1.*

Remark 12.6 *$\gamma(t)$ can be written by means of the monomial basis. Following this form, we have*

$$\phi_i(t) = \sum_{i=0}^{n} a_i \, t^i \quad or \quad \gamma(t) = \sum_{i=0}^{n} A_i \, t^i \, ,$$

where the A_i's are indeed combinations of the given P_i's.

This interpolation function is C^∞ which is, at the same time, nice but probably unnecessary. On the other hand, when n is large this curve definition may produce oscillations and lead to an expensive computational effort. Thus, this type of polynomial function is mostly of theoretical interest or used in practice as a component of another curve definition.

12.2.2 Recursive form for a Lagrange interpolation

The above Lagrange interpolate can be written in a recursive manner. We introduce a polynomial of degree 1, which corresponds to the case $n = 1$ of the above general definition. This polynomial is defined in the interval $[t_i, t_{i+1}]$. It is given by

$$A_i^1(t) = \frac{t_{i+1} - t}{t_{i+1} - t_i} P_i + \frac{t - t_i}{t_{i+1} - t_i} P_{i+1}; \quad i = 0, ..., n-1 \, .$$

Using this relation[2], we construct a recursion. To this end, we assume that we have found a polynomial, say $A_0^{n-1}(t)$, that interpolates to the n first P_i's (*i.e.*, $i = 0, ..., n-1$) along with a polynomial $A_1^{n-1}(t)$ that interpolates to the n last P_i's (*i.e.*, $i = 1, ..., n$). With this material, we define the following sequence :

$$A_0^n(t) = \frac{t_n - t}{t_n - t_0} A_0^{n-1}(t) + \frac{t - t_0}{t_n - t_0} A_1^{n-1}(t) \, .$$

Then, we have

$$A_0^n(t_i) = P_i \, .$$

Proof. First, for $t = t_0$, we have $A_0^n(t_0) = A_0^{n-1}(t_0)$ which leads to having $A_0^n(t_0) = A_0^1(t_0)$. Then, following the definition of the A_1^i's with $i = 0$ and $t = t_0$, we have

$$A_0^n(t_0) = \frac{t_1 - t_0}{t_1 - t_0} P_0 = P_0 \, .$$

Similarly, for $t = t_n$, we have $A_0^n(t_n) = A_1^{n-1}(t_n)$, thus $A_0^n(t_n) = A_{n-1}^1(t_n)$, *i.e.*,

$$\frac{t_n - t_{n-1}}{t_n - t_{n-1}} P_n = P_n \, .$$

For the other t_i's, we just have to follow the way in which the recursion was defined. Indeed, we have $A_0^{n-1}(t_i) = A_1^{n-1}(t_i) = P_i$, then $A_0^n(t_i) = P_i$. □

The above recursion can be generalized in :

$$A_i^r(t) = \frac{t_{i+r} - t}{t_{i+r} - t_i} A_i^{r-1}(t) + \frac{t - t_i}{t_{i+r} - t_i} A_{i+1}^{r-1}(t) \, ; \, i = 0, ..., n-r \quad r = 1, ..., n$$

the so-called Aitken algorithm.

As a conclusion, the Lagrange interpolate can be written as initially stated or by means of the above Aitken algorithm, namely

$$\gamma(t) = A_0^n(t) \, .$$

[2] Where the notation A is short for *Aitkens*, as will be justified subsequently.

12.2.3 Matrix form for a Lagrange interpolation

In practice, given an adequate set of t_i's, the Lagrange interpolate can be expressed, for each component, in a simple matrix form :

$$\gamma(t) = [\mathcal{T}(t)][\mathcal{M}][\mathcal{P}]$$

with :

$$[\mathcal{T}(t)] = [t^n \quad t^{n-1} \quad \dots \quad t \quad 1]$$
$$[\mathcal{P}] = {}^t[P_0 \quad P_1 \quad \dots \quad P_n]$$

where $[\mathcal{M}]$ is a $(n+1) \times (n+1)$ matrix.

12.2.4 Lagrange forms of degree 1 and 2

In the case where parameter t is chosen as: $t_i - t_{i-1} = \frac{1}{n}$, *i.e.*, the sequence t_i is uniform, matrix $[\mathcal{M}]$ is written as follows[3] :

$$[\mathcal{M}] = \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}$$

for $n = 1$ (assuming $[t_0, t_1] = [0, 1]$). A similar expression, whatever n is, can be obtained (obviously unlikely to be suitable when n is large). For instance, for $n = 2$, $t_0 = 0$, $t_2 = 1$ and $t_1 = 0.5$, we find :

$$[\mathcal{M}] = \begin{pmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{pmatrix}.$$

Proof. Using the general formula for $n = 2$, we have :

$$\gamma(t) = \frac{(t-t_1)(t-t_2)}{(t_0-t_1)(t_0-t_2)} P_0 + \frac{(t-t_0)(t-t_2)}{(t_1-t_0)(t_1-t_2)} P_1 + \frac{(t-t_0)(t-t_1)}{(t_2-t_0)(t_2-t_1)} P_2 ,$$

since we set $t_0 = 0$, $t_1 = 0.5$ and $t_2 = 1$, we have :

$$\gamma(t) = (2t^2 - 3t + 1)\, P_0 + (-4t^2 + 4t)\, P_1 + (2t^2 - t)\, P_2 ,$$

thus, $[\mathcal{M}]$ is the above expression. □

Proof. Using the Aitken formula (under the same assumptions), we have successively :

$$A_0^2(t) = \frac{t_2 - t}{t_2 - t_0} A_0^1(t) + \frac{t - t_0}{t_2 - t_0} A_1^1(t) ,$$

$$A_0^1(t) = \frac{t_1 - t}{t_1 - t_0} P_0 + \frac{t - t_0}{t_1 - t_0} P_1 ,$$

[3] Our interest in this simple case will be made precise hereafter.

$$A_1^1(t) = \frac{t_2 - t}{t_2 - t_1} P_1 + \frac{t - t_1}{t_2 - t_1} P_2 ,$$

$$A_0^2(t) = \frac{(t-t_1)(t-t_2)}{(t_0-t_1)(t_0-t_2)} P_0 + \frac{(t-t_0)(t-t_2)}{(t_1-t_0)(t_1-t_2)} P_1 + \frac{(t-t_0)(t-t_1)}{(t_2-t_0)(t_2-t_1)} P_2 ,$$

which is the same expression as above, thus leading to the same matrix. □

Without restriction on the t_i's, we have :

$$[\mathcal{M}] = \begin{pmatrix} \frac{1}{(t_0-t_1)(t_0-t_2)} & \frac{1}{(t_1-t_0)(t_1-t_2)} & \frac{1}{(t_2-t_0)(t_2-t_1)} \\ \frac{-t_1-t_2}{(t_0-t_1)(t_0-t_2)} & \frac{-t_0-t_2}{(t_1-t_0)(t_1-t_2)} & \frac{-t_0-t_1}{(t_2-t_0)(t_2-t_1)} \\ \frac{t_1 t_2}{(t_0-t_1)(t_0-t_2)} & \frac{t_0 t_2}{(t_1-t_0)(t_1-t_2)} & \frac{t_0 t_1}{(t_2-t_0)(t_2-t_1)} \end{pmatrix}$$

12.2.5 Hermite interpolation scheme

Another class of interpolation methods uses *Hermite* type interpolants. In addition to the above $n + 1$ pair of control points and parameter values, we assume that the $n + 1$ derivatives $\dot{P}_0, \dot{P}_1, \dots, \dot{P}_n$ are provided. Then, the function :

$$\gamma(t) = \sum_{i=0}^{n} \tilde{\phi}_i(t) P_i + \sum_{i=0}^{n} \varphi_i(t) \dot{P}_i \qquad (12.3)$$

where

$$\tilde{\phi}_i(t) = \{1 - 2\,\phi_i'(t_i)(t - t_i)\}\, \phi_i(t)^2$$

and

$$\varphi_i(t) = (t - t_i)\phi_i(t)^2$$

defines a curve which is governed by the $n + 1$ given points and their tangents. This curve, called the *Hermite interpolate*, is such that :

$$\gamma(t_i) = P_i \quad \textit{and} \quad \gamma'(t_i) = \dot{P}_i$$

as can be shown, moreover it is C^∞. Nevertheless, we return to the previous remarks both in terms of how to define the parameter values and in terms of complexity and oscillations (if n is large).

12.2.6 The Hermite cubic form

We reduce to the case where $n = 1$ and we assume that P_0 and P_1 along with $\dot{P}_0$ and $\dot{P}_1$ are supplied. Then, since, after a variable change, $t_0 = 0$ and $t_1 = 1$ are assumed, we have :

$$\begin{aligned} \phi_0(t) &= (1 - t) & \text{and} \quad & \phi_0'(t) = -1 \\ \phi_1(t) &= t & \text{and} \quad & \phi_1'(t) = 1 \\ \tilde{\phi}_0(t) &= (1 + 2t)(1 - t)^2 & \text{and} \quad & \tilde{\phi}_1(t) = (1 - 2(t - 1))\, t^2 \\ \varphi_0(t) &= t(1 - t)^2 & \text{and} \quad & \varphi_1(t) = (t - 1)t^2 \end{aligned}$$

thus :

$$\gamma(t) = (1+2t)(1-2t+t^2)\,P_0 + (t^2 - 2t^3 + 2t^2)\,P_1 + (t - 2t^2 + t^3)\,\dot{P}_0 + (t^3 - t^2)\,\dot{P}_1 ,$$

which can be expressed in a matrix form as :

$$\gamma(t) = [\mathcal{T}(t)][\mathcal{M}][\mathcal{P}] \qquad (12.4)$$

where now :

$$[\mathcal{T}(t)] = [t^3 \quad t^2 \quad t \quad 1]$$

$$[\mathcal{P}] = {}^t\,[P_0 \quad P_1 \quad \dot{P}_0 \quad \dot{P}_1]$$

$$[\mathcal{M}] = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

which is known as the cubic Hermite interpolation or the cubic Coons basis.

Remark 12.7 *Actually, the cubic Hermite form concerns polynomials of degree 3 considered with a set of 4 data values (2 points and 2 derivatives). Thus, this simple example will be of interest when composite curves are seen (see below).*

Remark 12.8 *Higher order Hermite type interpolations can be constructed if higher order derivatives are supplied as input data.*

12.3 Explicit construction of a composite curve

In this section, we give a method suitable for constructing a composite curve using one of the previous approaches. Indeed, after the remark about the cost and the possible existence of oscillations when the number of points n is large, we want to define the curve Γ by a series of sub-curves with a low degree such that their junctions are under control. Moreover, this construction uses as input data a discrete approximation of the curve under investigation.

This approximation is indeed a *polyline* composed of a series of segments $[P_i, P_{i+1}]$. The construction is completed segment by segment using the available information (tangents, corners, etc.). Depending on the amount of information provided as input data, a polynomial based definition of degree 1, 2 or 3 can be easily obtained in any member of the polyline. In what follows, we consider one segment, say $[P_i, P_{i+1}]$, and we denote this segment by $[A, B]$.

Giving 4 input data, A, B and the tangents at A and at B, $\vec{\tau_A}$ and $\vec{\tau_B}$, we can obtain a function like :

$$\gamma(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 , \qquad (12.5)$$

where $a_i \in \mathbb{R}^d$ $(i = 0, 3)$ and t is a parameter ranging from 0 to 1. For the moment, we know nothing more about the meaning of t.

First, we consider the case where no tangent is provided. Using the only two data we have, we look for a function γ of the above type with $a_2 = a_3 = 0$, *i.e.*, a function of degree one. The constraints we want to conform to are $\gamma(0) = A$ along with $\gamma(1) = B$. Then, we must solve the system :

$$\begin{cases} a_0 = A \\ a_1 = B - A , \end{cases} \qquad (12.6)$$

meaning that $\gamma(t) = A + t(B - A)$. Indeed, the Lagrange form of degree 1, as defined in Section (12.2.1), is retrieved.

Remark 12.9 *Since the derivative is $\gamma'(t) = \vec{AB}$, we encounter two cases. First, if $||\vec{AB}|| = 1$, t is nothing other than s, the curvilinear abscissa (see Chapter 11), while, if $||\vec{AB}|| \neq 1$, t is an arbitrary parameter. In the previous case, a curvilinear abscissa can be defined by $s = t\,||\vec{AB}||$.*

When either of the tangents is not provided, we look for a function of degree two, *i.e.*, we fix $a_3 = 0$. Then, if $\vec{\tau_A}$ is known, we have :

$$\begin{cases} a_0 = A \\ a_1 = \vec{\tau_A} \\ a_2 = (B - A) - \vec{\tau_A} , \end{cases} \qquad (12.7)$$

while if only $\vec{\tau_B}$ is known, we obtain :

$$\begin{cases} a_0 = A \\ a_1 = 2(B - A) - \vec{\tau_B} \\ a_2 = -(B - A) + \vec{\tau_B} . \end{cases} \qquad (12.8)$$

Then, we assume that we know the tangents at A and at B, $\vec{\tau_A}$ and $\vec{\tau_B}$, then this information allows us to define a curve of degree 3. The four given information elements lead to the system :

$$\begin{cases} a_0 = A \\ a_0 + a_1 + a_2 + a_3 = B \\ a_1 = \vec{\tau_A} \\ a_1 + 2a_2 + 3a_3 = \vec{\tau_B} , \end{cases} \qquad (12.9)$$

whose solution is :

$$\begin{cases} a_0 = A \\ a_1 = \vec{\tau_A} \\ a_2 = 3(B - A) - 2\vec{\tau_A} - \vec{\tau_B} \\ a_3 = -2(B - A) + \vec{\tau_A} + \vec{\tau_B} . \end{cases} \qquad (12.10)$$

Then, $\gamma(t)$ is nothing other than the cubic Hermite form already mentioned in Section (12.2.5).

$\vec{\tau_A}$	$\vec{\tau_A}$	.	.	A_{-1}	$\vec{\tau_A}$	A_{-1}	A_{-1}	.
A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B
$\vec{\tau_B}$	.	$\vec{\tau_B}$	.	$\vec{\tau_B}$	B_{+1}	B_{+1}	.	B_{+1}
3	2	2	1	3	3	3	2	2

Table 12.1: Degree of the approximation (bottom line) according to the data type (in column).

Remark 12.10 *It is advisable to make sure that* $||\vec{\tau}||$ *is something like* $||AB||$. *Indeed* $||\vec{\tau}||$ *acts through two aspects. First we naturally encounter a directional aspect and then the module of the tangent controls the locality of this directional control.*

Other categories of situations are encountered when A and/or B are not corner(s) but when we know a point "before" A, denoted as A_{-1}, and/or a point "after" B, denoted as B_{+1}, then we can define a function of degree three or two by returning to the previous cases. The tangents at AB in A and/or B are evaluated by using the points A_{-1} and/or B_{+1}. For instance, $\vec{\tau_A} = A - A_{-1}$ or a similar expression. The final type that can be observed is when the data combine the different possibilities.

Remark 12.11 *Apart from the first case where t could be s or s could be easily defined resulting in a normal parameterization of AB, such a parameterization is not obtained in the other situations.*

Table 12.1 shows, as a function of the data categories, the degree that can be expected for the approximation of AB.

A few remarks. In general, the function $\gamma(t)$ constructed from two points and the two related tangents may offer different aspects depending on the type of this information. Some aspects may obviously be undesirable for the type of applications envisaged.

Thus, the presence of a loop in one segment, *i.e.*, when there are two different values t_1 and t_2 for which $\gamma(t_1) = \gamma(t_2)$ necessarily corresponds to ill-suited data. In fact, a nice property to guarantee what we need, is to have a function g of parameter t such that

- $g(t)$ is a strictly increasing function.

where $g(t) = d(A, proj_{AB}(\gamma(t))$ in segment AB. In this expression, d denotes the usual distance while $proj_{AB}(P)$ is the projection of point P onto AB (cf. Figure 12.1)

The assumed property implies in particular that the curve has no loop (between A and B). From a practical point of view, we can also assume that the distance between AB and the curve is bounded by a reasonably small threshold value. In

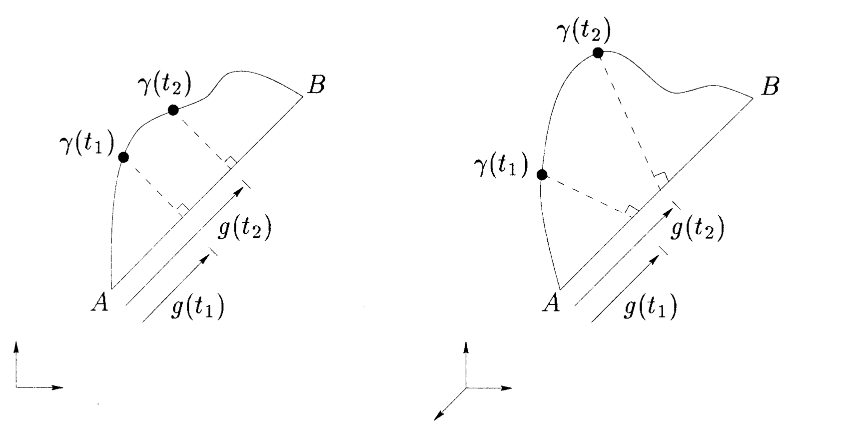

Figure 12.1: $\gamma(t)$ *and* $g(t)$ *for a curve in two dimensions (left-hand side) and for a curve in three dimensions (right-hand side).*

other words, the segment AB is close to the curve, meaning that the segments provided as input already correspond to a reasonable approximation of the geometry of the real curve.

12.4 Control polygon based methods

We now turn to a class of methods not based on an interpolation technique. Here, we are interested in methods that make use of a control polygon whose vertices act through some contributions defined via various functions.

12.4.1 Control polygon and curve definition

The set of control points enables us to define a *control polygon*. As a general statement, it is possible to define a curve in terms of the following equation :

$$\gamma(t) = \sum_{i=0}^{n} \varphi_i(t) P_i \qquad (12.11)$$

Once the basis functions $\varphi_i(t)$'s have been chosen, the curve is uniquely defined by the points P_i's in the control polygon. At a glance, any basis of functions makes this curve definition possible. However, some properties are usually required so as to ensure that the curve "looks like" its control polygon. These assumptions are indeed assumed to facilitate curve manipulation when used in a CAD system, particularly, if the user has only limited knowledge of the underlying geometric models.

The purpose of the following section is to clarify the above properties.

12.4.2 General properties

The following presents some of the properties commonly assumed in most of the models based on a control polygon.

Cauchy identity. This relationship is related to a normalization purpose. Then, we have :

$$\sum_{i=0}^{n} \varphi_i(t) \equiv 1 \quad \forall t\,.$$

This implies that Equation (12.11) is equivalent to :

$$\gamma(t) = \frac{\sum\limits_{i=0}^{n} \varphi_i(t) P_i}{\sum\limits_{i=0}^{n} \varphi_i(t)}\,. \qquad (12.12)$$

This expression of the curve clearly shows the barycentric form of the latter. The current point $\gamma(t)$ is the centroid of the set of points P_i's associated with the weights $\varphi_i(t)$.

Remark 12.12 (Curve translation) *We look at the construction of a new polygon resulting from the translation of the control polygon P_i by a vector v. The coordinates of the points of this new control polygon are denoted by $P_i + v$. The new curve γ' associated with this new polygon can be written as :*

$$\gamma'(t) = \sum_{i=0}^{n} \varphi_i(t)(P_i + v) = \sum_{i=0}^{n} \varphi_i(t) P_i + \underbrace{\left(\sum_{i=0}^{n} \varphi_i(t)\right)}_{\equiv 1} v = \gamma(t) + v$$

If the Cauchy identity were not assumed, then the form of the curve would be a function of the position of the control polygon. Conversely, this identity implies that the curve is only related to the shape of its control polygon.

Remark 12.13 (Linear transformation of a curve) *We now turn to a method that makes it possible to construct a new curve resulting from a linear transformation of the curve $\gamma(t)$. Then, this new curve, $\gamma'(t)$, conforms to the general expression :*

$$\gamma'(t) = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \gamma(t)\,.$$

Thus, we have :

$$\gamma'(t) = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \left[\sum_{i=0}^{n} \varphi_i(t) \begin{pmatrix} P_{i_x} \\ P_{i_y} \\ P_{i_z} \end{pmatrix}\right]$$

$$= \sum_{i=0}^{n} \varphi_i(t) \left[\begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} P_{i_x} \\ P_{i_y} \\ P_{i_z} \end{pmatrix}\right],$$

where the second equality is true only if the Cauchy condition is assumed. Thus, any linear transformation of a curve (for instance a rotation, a scaling, etc.) is easy to perform directly in the control polygon.

Positive functions. Inside the definition interval of γ, the functions $\varphi_i(t)$ are assumed to be larger than or equal to zero :

$$\varphi_i(t) \geq 0\,.$$

Provided with the Cauchy identity, the current point $\gamma(t)$ is the centroid of the set of points P_i's associated with *positive weights*, the $\varphi_i(t)$'s. Hence, all any convex region bounding the set of points P_i's encloses the curve.

Extremity conditions. If $[t_{min}, t_{max}]$ stands for the definition interval of γ, the functions φ_i satisfy the following :

$$\varphi_0(t_{min}) = 1 \quad \text{and} \quad \varphi_n(t_{max}) = 1\,.$$

Corollary 12.1 *The Cauchy condition and the previous relationships with positive functions imply that :*

$$\underbrace{\varphi_i(t_{min})}_{i \neq 0} \equiv 0 \quad \textit{and} \quad \underbrace{\varphi_i(t_{max})}_{i \neq n} \equiv 0\,.$$

Hence, at $t = t_{min}$, the sole point with an action on the curve is the point P_0. Moreover, we have the relationships $\gamma(t_{min}) = P_0$ and $\gamma(t_{max}) = P_n$. As a consequence, the curve endpoints are the endpoints of the corresponding control polygon.

Derivatives at curve endpoints. In addition, the following conditions can be assumed. At t_{min}, we impose :

$$\frac{d\varphi_0}{dt}(t_{min}) = -\frac{d\varphi_1}{dt}(t_{min}) \quad \text{and} \quad \underbrace{\frac{d\varphi_i}{dt}(t_{min})}_{i>1} \equiv 0$$

and at parameter t_{max} :

$$\frac{d\varphi_m}{dt}(t_{max}) = -\frac{d\varphi_{m-1}}{dt}(t_{max}) \quad \text{and} \quad \underbrace{\frac{d\varphi_i}{dt}(t_{max})}_{i<m-1} \equiv 0\,.$$

Hence, for $t = t_{min}$, we obtain the following equation :

$$\frac{d\gamma}{dt}(t_{min}) = \sum_{i=0}^{m} \frac{d\varphi_i}{dt}(t_{min}) P_i = \frac{d\varphi_0}{dt}(t_{min})(P_1 - P_0)\,.$$

Thus, the tangent at t_{min} is directed by the first segment in the control polygon. Similarly, the tangent at t_{max} has the same direction as the last segment in this polygon.

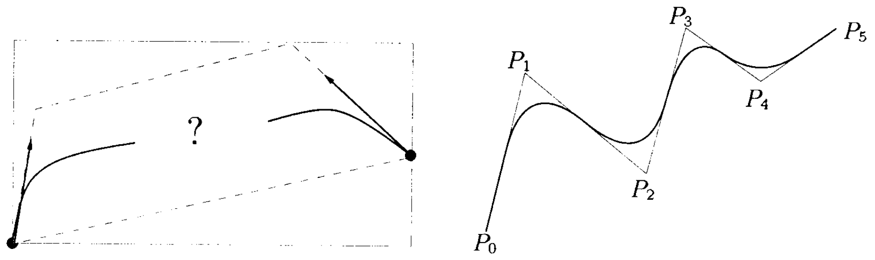

Figure 12.2: *Basis properties related to the definition of a control polygon. The shape of the curve is close to this governing polyline (broken line).*

Figure 12.2 depicts the various properties of the curves governed by a control polygon. At first, $\gamma(t)$ passes through P_0 and P_n while the other points are only control points. Moreover, the tangents at the curve endpoints follow the first and the last segments in the polygon. In addition, the curve lies inside any convex region including its control polygon.

In this figure, the polygon in dashed line is the smallest of the convex polygons containing the control polygon. CAD systems usually use the enclosing box shown in the figure by the rectangle to bound the region enclosing the curve. Apart from the characteristics at the curve endpoints, the only thing we can say *a priori* about the shape of the curve is that it "looks like" its control polygon.

12.5 Bézier curves

A popular method based on a control polygon makes use of *Bézier curves.*

12.5.1 Form of a Bézier curve

Provided with the $n+1$ control points, we define a Bézier curve using the following algebraic relation

$$\gamma(t) = \sum_{i=0}^{n} C_n^i t^i (1-t)^{n-i} P_i \,. \qquad (12.13)$$

with $C_n^i = \frac{n!}{(n-i)!i!}$ for $0 \leq i \leq n$ and $C_n^i = 0$ otherwise[4].

In other words, a Bézier curve relies on Bernstein polynomials . Indeed, these polynomials are defined by

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}$$

[4]Note that coefficient C_n^i has various notations. For instance, one can find the notation ${}^nC_i$ as well as $\binom{n}{i}$.

where t ranges from 0 to 1 and $i \in [0, n]$ while $B_{i,n}(t) = 0$ elsewhere. Thus, we have :

$$\gamma(t) = \sum_{i=0}^{n} B_{i,n}(t)\, P_i \,.$$

Exercise 12.3 *Show that the above properties hold.*

Remark 12.14 *The parameter value t ranges from 0 to 1. It is possible to use a different set of parameter values where t varies from a given t_0 to a given $t_n > t_0$. To return to an interval between 0 to 1, we simply have to introduce a variable change such as $\tilde{t} = \frac{t-t_0}{t_n-t_0}$. The way in which the t_i's vary allow for some flexibility in the curve definition.*

This curve definition is C^∞. Nevertheless, for a large n, the above remarks remain true. In addition, while more flexible than the previous approaches, some classical curves, such as conics are poorly represented by this Bézier form. To deal with such a problem, two solutions can be envisioned. On the one hand, a different global curve definition can be used (see for instance the rational Bézier) or, on the other hand, a low degree Bézier can be employed leading to using a composite definition for the curve (see below).

12.5.2 About Bernstein polynomials

First, it can be proved that Bernstein polynomials form a basis of all polynomials of degree n. In this respect, Bernstein polynomials can be written as :

$$B_{i,n}(t) = \sum_{j=i}^{n} (-1)^{j-i}\, C_n^j C_j^i\, t^j \,.$$

On the other hand, Bézier curves offer some facility for various computations. Indeed, since these curves are defined by means of Bernstein polynomials, various recursions about these polynomials allow us to elegantly manipulate these curves. First, the C_n^i's conform to the following recursion :

$$C_n^i = C_{n-1}^{i-1} + C_{n-1}^i \,.$$

In other words, the C_n^i's can be obtained by the Pascal triangle rule as illustrated in Table 12.2.

Then, the Bernstein polynomials can be obtained by recursion. Actually, we have :

$$B_{i,n}(t) = t\, B_{i-1,n-1}(t) \;+\; (1-t)\, B_{i,n-1}(t) \,. \qquad (12.14)$$

Proof. Merging the recursion about the C_n^i's in $B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}$, we obtain successively :

$$B_{i,n}(t) = C_{n-1}^{i-1} t^i (1-t)^{n-i} + C_{n-1}^i t^i (1-t)^{n-i} \,,$$

$$B_{i,n}(t) = C_{n-1}^{i-1} t\, t^{i-1}(1-t)^{n-1-(i-1)} + C_{n-1}^{i} t^i (1-t)(1-t)^{n-1-i},$$

$$B_{i,n}(t) = t\, C_{n-1}^{i-1} t^{i-1}(1-t)^{n-1-(i-1)} + (1-t)\, C_{n-1}^{i} t^i (1-t)^{n-1-i},$$

and the above recursion holds. □

-	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	....
$n=0$	1						
$n=1$	1	1					
$n=2$	1	2	1				
$n=3$	1	3	3	1			
$n=4$	1	4	6	4	1		
$n=5$	1	5	10	10	5	1	
...							

Table 12.2: The Pascal triangle rule.

Obviously we have :

$$B_{i,n}(t) = B_{n-i,n}(1-t).$$

Moreover,

$$B_{0,n}(t) = (1-t)\, B_{0,n-1}(t) \text{ thus } B_{0,n}(t) = (1-t)^n,$$

$$B_{n,n}(t) = t\, B_{n-1,n-1}(t) \text{ and } B_{n,n}(t) = t^n.$$

A recursion about the derivatives can be easily found. It is as follows :

$$B'_{i,n}(t) = n\,(B_{i-1,n-1}(t) - B_{i,n-1}(t)).$$

Various recursions related to high order derivatives, integrals and many other relationships can be exhibited. To conclude this discussion about Bernstein polynomials, one should note that all of the above recursions are also useful in the case of rational Bézier, composite Bézier, etc., curve representations.

12.5.3 De Casteljau form for a Bézier curve

Defining $D_i^0(t) = P_i$ for $i = 0, ..., n$, the recursion[5] :

$$D_i^r(t) = (1-t)\, D_i^{r-1}(t) + t\, D_{i+1}^{r-1}(t)$$

for $r = 1, n$ and $i = 0, n-r$ (and $D_i^r(t) = 0$ otherwise) is the so-called De Casteljau algorithm. Using this recursion, the curve defined by :

$$\gamma(t) = D_0^n(t),$$

is a practical way to construct a Bézier curve which does not directly involve the use of Bernstein polynomials.

[5] The notation D stands for *De Casteljau*.

12.5.4 Bézier curve of degree 3

Any Bézier curve can be written in the matrix form already introduced. For example, for $n = 3$, the general expression or the De Casteljau algorithm results in the matrix $[\mathcal{M}]$ given by :

$$[\mathcal{M}] = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

with

$$[\mathcal{T}(t)] = [t^3 \quad t^2 \quad t \quad 1]$$

$$[\mathcal{P}] = {}^t\,[P_0 \quad P_1 \quad P_2 \quad P_3]$$

Proof. Using Bernstein polynomials, we have successively :

$$B_{0,3}(t) = (1-t)^3,$$

$$B_{1,3}(t) = t\, B_{0,2}(t) + (1-t)\, B_{1,2}(t),$$

$$B_{1,2}(t) = t\, B_{0,1}(t) + (1-t)\, B_{1,1}(t),$$

$$B_{1,2}(t) = t\,(1-t) + (1-t)\,t = 2t\,(1-t),$$

$$B_{1,3}(t) = t\,(1-t)^2 + 2t\,(1-t)^2 = 3t\,(1-t)^2,$$

$$B_{2,3}(t) = t\, B_{1,2}(t) + (1-t)\, B_{2,2}(t),$$

$$B_{2,3}(t) = 2t^2\,(1-t) + (1-t)\,t^2 = 3t^2\,(1-t),$$

$$B_{3,3}(t) = t^3,$$

then :

$$\gamma(t) = B_{0,3}(t)\, P_0 + B_{1,3}(t)\, P_1 + B_{2,3}(t)\, P_2 + B_{3,3}(t)\, P_3,$$

$$\gamma(t) = (1-t)^3\, P_0 + 3t\,(1-t)^2\, P_1 + 3t^2\,(1-t)\, P_2 + t^3\, P_3,$$

and we retrieve the above matrix. □

Proof. Using De Casteljau algorithm, we have successively :

$$D_0^1(t) = (1-t)\, D_0^0(t) + t\, D_1^0(t),$$

$$D_0^1(t) = (1-t)\, P_0 + t\, P_1,$$

$$D_1^1(t) = (1-t)\, D_1^0(t) + t\, D_2^0(t),$$

$$D_1^1(t) = (1-t)\, P_1\, t\, P_2,$$

$$D_0^2(t) = (1-t)\, D_0^1(t) + t\, D_1^1(t),$$

$$D_0^2(t) = (1-t)^2\, P_0 + 2t\,(1-t)\, P_1 + t^2\, P_2,$$

$$D_1^2(t) = (1-t)\, D_1^1(t) + t\, D_2^1(t),$$

$$D_1^2(t) = (1-t)^2\, P_1 + 2t\,(1-t)\, P_2 + t^2\, P_3,$$

then :

$$D_0^3(t) = (1-t)^3\, P_0 + 3t\,(1-t)^2\, P_1 + 3t^2\,(1-t)\, P_2 + t^3\, P_3,$$

and we still retrieve the above matrix. □

12.5.5 Degree elevation of a Bézier curve

Degree elevation of a Bézier curve is a very useful process for various purposes (including the problem of finding some degree of continuity when composite curves (surfaces) are considered).

Given a Bézier curve of degree n corresponding to $n+1$ control points P_i, degree elevation leads to defining the same curve as a curve of degree $n+1$ based on $n+2$ control points, the Q_i's. The issue is to find these control points.

Let $\gamma(t) = \sum_{i=0}^{n} B_{i,n}(t)\, P_i$ be the given curve, which we want to be written as $\gamma(t) = \sum_{i=0}^{n+1} B_{i,n+1}(t)\, Q_i$. The solution is as follows :

$$Q_0 = P_0 \quad \text{and} \quad Q_{n+1} = P_n \quad \text{with}$$

$$Q_i = \frac{i\, P_{i-1} + (n+1-i)\, P_i}{n+1}, \quad i = 1, ..., n\,.$$

Proof. Since $B_{i,n}(t) = (1-t)\, B_{i,n}(t) + t B_{i,n}(t)$ holds where the $B_{i,n}(t)$'s are defined by $B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}$, we have :

$$B_{i,n}(t) = \frac{n+1-i}{n+1} B_{i,n+1}(t) + \frac{i+1}{n+1} B_{i+1,n+1}(t)\,,$$

then,

$$\gamma(t) = \sum_{i=0}^{n} B_{i,n}(t)\, P_i = \sum_{i=0}^{n} \left(\frac{n+1-i}{n+1} B_{i,n+1}(t)\, P_i + \frac{i+1}{n+1} B_{i+1,n+1}(t)\, P_i \right)\,,$$

and, after rearranging the terms, we obtain :

$$\gamma(t) = B_{0,n+1}\, P_0 + \sum_{1=0}^{n} B_{i,n+1} \left(\frac{i}{n+1} P_{i-1} + \frac{n+1-i}{n+1} P_i \right) + B_{n+1,n+1}\, P_n\,,$$

thus leading to the definition of the Q_i's. □

12.6 From composite curves to B-Splines

The main idea is to use locally one of the above curve definitions where a low degree is assumed. In this way, a curve is defined by a series of sub-curves. The issue is then to insure the desired smoothness between the different parts of the entire curve. This technique actually leads to constructing a *composite* curve.

12.6.1 Composite Bézier curves

Composite Bézier curves correspond to Bézier curves. We define a *knot sequence* of t_j's and a series of segments where Bézier polynomials of degree m are employed. Here, segment j, for $j \geq 1$, is denoted by seg_j. It is simply the interval :

$$seg_j = [t_j, t_{j+1}]\,.$$

With this background a composite Bézier curve is :

$$\gamma(t) = \sum_{j=1}^{n_m} \sum_{i=0}^{m} B^j_{i,m}(t)\, P_{i,j}\,, \tag{12.15}$$

where (to return to the case where t ranges from 0 to 1 for each segment), the $B^j_{i,m}(t)$'s are written by means of Bernstein polynomials as follows :

$$B^j_{i,m}(t) = B_{i,m}\left(\frac{t-t_j}{t_{j+1}-t_j}\right) \quad \text{if } t \in seg_j$$

$$B^j_{i,m}(t) = 0 \quad \text{otherwise}$$

and

$$P_{i,j} = P_{m\,(j-1)+i}$$

meaning that a segment runs from

$$P_{0,j} = P_{m\,(j-1)} \quad \text{to} \quad P_{m,j} = P_{mj}$$

and, finally n_m depends both on n and m. In fact, ones needs to have $n = (m+1) \times n_m$.

The curve only passes through the $P_{m\,(j-1)}$'s, the other points really being control points.

The definition of the t_j's can be made in several manners thus leading to some extent of flexibility. First, t_0 is naturally associated with P_0, then t_1 must be associated with P_m and so on. In fact t_j is related to $P_{m\,(j-1)}$. On the other hand, the values of the t_j's can be arbitrarily defined.

First, as already indicated, the entire curve is defined by a series of segments, in other words by means of local Bézier definitions and locally, the regularity in C^∞. Second, the global regularity is insured by an adequate definition of the junction from segment to segment. In this respect a G^1 or even a C^1 continuity can be obtained, say for $m = 3$, by acting on the control points previous and next to a segment endpoint. Actually, the three above points must be colinear (thus leading to a G^1 smoothness) and, in addition, must be in a certain ratio (to reach a C^1 property). Similarly a G^2 or a C^2 continuity, for $m = 4$, can be completed by acting on the two previous and the two next control points of a given segment endpoint. These requests allow for the desired regularity when the entire curve is considered but, on the other hand, imply some constraints that can impede the curve definition. A greater flexibility is then obtained by introducing the curve representation discussed in the next section.

Remark 12.15 *The above composite curve definition involves locally a single curve function. Thus, global continuity is obtained at some price. To overcome this problem local functions acting as basis functions may themselves be composite (see below).*

12.6.2 B-splines curves.

B-spline curves correspond to De Boor spline curves. Before going further in B-splines, we introduce the notion of a spline.

Given a sequence of $m+2$ knots $t_0 \leq t_1 \leq ... \leq t_{m+1}$, a general basis polynomial spline f of degree m is a function that satisfies the following properties :

- f is a polynomial of degree m on all intervals $[t_i, t_{i+1}]$, this restriction being denoted by f_i,
- f is such that $f_i(t_{i+1}) = f_{i+1}(t_{i+1})$,
- f is C^{m-1} at the junction points (*i.e.*, at the t_i's) when these knots are not multiple. At a node of multiplicity r, the continuity is C^{m-r}.

Remark 12.16 *Note that the above composite Bézier curve definition is included in this category of definitions while, in this case, the corresponding f_i's reduce to a single (non composite) function and the third condition is not automatically insured.*

Given a knot sequence of $m+2$ t_j's like $t_i, t_{i+1}, ..., t_{i+m+1}$, the basis spline $N_{i,m}$ related to these knots is constructed[6] using a recursion, in terms of index k. For $k = 0$, it leads to :

$$\begin{cases} N_{i,0}(t) = 1 & \text{if } t_{i-1} \leq t < t_i \text{ and} \\ N_{i,0}(t) = 0 & \text{else ,} \end{cases}$$

and, for $k = 1, 2, ..., m$, we have :

$$N_{i,k}(t) = \frac{t - t_{i-1}}{t_{i-1+k} - t_{i-1}} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_i} N_{i+1,k-1}(t) ,$$

meaning that the composite function $N_{i,k}(t)$ is non null only if $t \in [t_{i-1}, t_{i+k}]$.

The basis polynomials have a local and minimal support. Moreover, they are linearly independent. Indeed, they form a basis.

In the above recursion, if, for instance, $t_{i-1+k} - t_{i-1} = 0$, meaning that multiple knots are used, then the term in $N_{i,k-1}(t)$ does not contribute since $N_{i,k-1}(t) = 0$.

Remark 12.17 *Provided with an adequate choice of the t_i's, the above relation reduces to Relation (12.14) meaning that, in this particular case, the $N_{i,k}(t)$'s are nothing other than Bernstein polynomials. Indeed, we have to set $t_i = 0$ for $i = 0, 1, ..., k$ and $t_{k+1} = ... = t_{2k+1} = 1$.*

With this background (leaving aside the previous remark), given $n+1$ control points ($n \geq m$), a composite curve of $n - m + 1$ polynomial curves of degree m is defined as

$$\gamma(t) = \sum_{j=i+1-m}^{i+1} N_{j,m}(t) P_j . \quad (12.16)$$

[6] In the following, we discuss only one of the various possible ways to define a B-spline.

Remark 12.18 *Since the initial condition of the definition by recursion is satisfied for $N_{j,m}(t) \equiv 0$ when $j \notin [i+1-m, i+1]$, we also have :*

$$\gamma(t) = \sum_{j=0}^{m} N_{j,m}(t) P_j . \quad (12.17)$$

where one can retrieve the general equation of a model based on a control polygon.

Exercise 12.4 *Study the properties of Section 12.4.1 in the case of the basis functions $N_{j,m}(t)$.*

This curve, the so-called B-spline, uses $n+1$ basis splines and thus needs a sequence of $n + m + 2$ knots, namely $t_0, t_1, ..., t_{n+m+1}$. The first curved segment that fully uses the first $m+1$ control points, is the combination $N_{0,m} P_0 + N_{1,m} P_1 + ... + N_{m,m} P_m$, which is defined in $[t_{m-1}, t_m]$. Similarly, segment number k is the combination $N_{k,m} P_k + N_{k+1,m} P_{k+1} + ... + N_{k+m,m} P_{k+m}$, which is defined in $[t_{k+m-1}, t_{k+m}]$. Then the last segment is defined in $[t_{n-1}, t_n]$ corresponding to the combination $N_{n-m,m} P_{n-m} + N_{n-m+1,m} P_{n-m+1} + ... + N_{n,m} P_n$.

Following on from what has just been said, the curve doesn't pass through the control points if the knots are all distinct. Using multiple knots enables us to pass through some control points. Boundary conditions can be also achieved by using multiple or phantom (fictitious) control points.

In the following three examples, we assume that the t_i's are all distinct.

12.6.3 Degree 1 B-spline

In this case $m = 1$ is assumed, then using the general recursion with $k = m = 1$, we have :

$$N_{i,1}(t) = \frac{t - t_{i-1}}{t_i - t_{i-1}} N_{i,0}(t) + \frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i+1,0}(t) . \quad (12.18)$$

This B-spline consists of two components. In other words, it lives in two intervals. This composite function enables us to define the curve $\gamma(t)$ as :

$$\gamma(t) = \sum_{j=i}^{i+1} N_{j,1}(t) P_j . \quad (12.19)$$

The first interval that is well defined corresponds to $j = 0$ and then to $i = 0$. Hence, in $[t_0, t_1]$, both $N_{0,1}(t)$ and $N_{1,1}(t)$ have a contribution. Since only $N_{1,0}(t) \neq 0$, we simply have (for $i = 0$) :

$$N_{0,1}(t) = \frac{t_1 - t}{t_1 - t_0} ,$$

while (for $i = 1$) :

$$N_{1,1}(t) = \frac{t - t_0}{t_1 - t_0} .$$

Then, following Relation (12.19), in $[t_0, t_1]$, we have :

$$\gamma(t) = \frac{t_1 - t}{t_1 - t_0} P_0 + \frac{t - t_0}{t_1 - t_0} P_1 .$$

Using a uniform distribution of t_i's (for instance, $t_i = i$), a simple variable change allows us to find a definition in $[0,1]$. Indeed, we return to the classical linear interpolation function $\gamma(t) = (1-t)\, P_0 + t\, P_1$.

Similarly, to define the curve in $[t_1, t_2]$, we need to know both $N_{1,1}(t)$ and $N_{2,1}(t)$ which act through their components related to $N_{2,0}(t)$. Thus we consider the general recursion and we fix $i = 1$ to obtain the contribution of $N_{1,1}(t)$:

$$N_{1,1}(t) = \frac{t_2 - t}{t_2 - t_1} ,$$

and we fix $i = 2$ to find the contribution of $N_{2,1}(t)$:

$$N_{2,1}(t) = \frac{t - t_1}{t_2 - t_1} ,$$

hence, in $[t_1, t_2]$, we have :

$$\gamma(t) = \frac{t_2 - t}{t_2 - t_1} P_1 + \frac{t - t_1}{t_2 - t_1} P_2 .$$

This definition (with the above assumptions) leads to having $\gamma(t) = (1-t)\, P_1 + t\, P_2$.

Remark 12.19 *In practice, for a uniform node distribution, the B-spline can be defined everywhere using only the factors* $(1-t)$ *and* t. *To this end, a variable change is done to reduce the interval of interest to* $[0,1]$.

Remark 12.20 *The B-spline is symmetric (observe the case where* t *is replaced by* $1-t$*).*

As expected, this curve is $C^{m-1} = C^0$. Indeed, in the first interval we have $\gamma(1) = P_1$ as well as in the second interval where $\gamma(0) = P_1$. Note also that the definition is reversible. The curve defined using P_0, P_1, P_2 and the curve defined by P_2, P_1, P_0 are identical. More generally, the sequences $P_0, P_1, P_2, ..., P_n$ and $P_n, P_{n-1}, P_{n-2}, ..., P_0$ lead to the same curve.

12.6.4 Degree 2 B-spline

Using the recursion about the $N_{i,k}$ in the case where $k = m = 2$, we have :

$$N_{i,2}(t) = \frac{t - t_{i-1}}{t_{i+1} - t_{i-1}} N_{i,1}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_i} N_{i+1,1}(t) . \qquad (12.20)$$

This B-spline, comprising three components, lives in three sub-intervals. The curve defined by :

$$\gamma(t) = \sum_{j=i-1}^{i+1} N_{j,2}(t)\, P_j , \qquad (12.21)$$

is well defined since the interval that corresponds to $i = 1$. Hence, in $[t_1, t_2]$. In this interval, three B-splines have a contribution, namely $N_{0,2}(t)$, $N_{1,2}(t)$ and $N_{2,2}(t)$. Merging Relationship (12.18) into Relationship (12.20), we find :

$$\begin{aligned} N_{i,2}(t) &= \frac{t - t_{i-1}}{t_{i+1} - t_{i-1}} \left(\frac{t - t_{i-1}}{t_i - t_{i-1}} N_{i,0}(t) + \frac{t_{i+1} - t}{t_{i+1} - t_i} N_{i+1,0}(t) \right) \\ &+ \frac{t_{i+2} - t}{t_{i+2} - t_i} \left(\frac{t - t_i}{t_{i+1} - t_i} N_{i+1,0}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} N_{i+2,0}(t) \right) \end{aligned}$$

and, in terms of $N_{j,0}(t)$, for $j = i, i+1, i+2$,

$$\begin{aligned} N_{i,2}(t) &= \frac{t - t_{i-1}}{t_{i+1} - t_{i-1}} \frac{t - t_{i-1}}{t_i - t_{i-1}} N_{i,0}(t) \\ &+ \left(\frac{t - t_{i-1}}{t_{i+1} - t_{i-1}} \frac{t_{i+1} - t}{t_{i+1} - t_i} + \frac{t_{i+2} - t}{t_{i+2} - t_i} \frac{t - t_i}{t_{i+1} - t_i} \right) N_{i+1,0}(t) \\ &+ \frac{t_{i+2} - t}{t_{i+2} - t_i} \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} N_{i+2,0}(t) . \end{aligned}$$

The desired contributions are obtained, in interval $[t_1, t_2]$, by looking at the coefficients of $N_{2,0}(t)$ which are successively related to $i = 2$, $i = 1$ and $i = 0$. Then, we have :

$$N_{2,2}(t) = \frac{t - t_1}{t_3 - t_1} \frac{t - t_1}{t_2 - t_1} ,$$

$$N_{1,2}(t) = \frac{t - t_0}{t_2 - t_0} \frac{t_2 - t}{t_2 - t_1} + \frac{t_3 - t}{t_3 - t_1} \frac{t - t_1}{t_2 - t_1} ,$$

$$N_{0,2}(t) = \frac{t_2 - t}{t_2 - t_0} \frac{t_2 - t}{t_2 - t_1} .$$

Relation (12.21), in $[t_1, t_2]$, is

$$\gamma(t) = \frac{t_2 - t}{t_2 - t_0} \frac{t_2 - t}{t_2 - t_1} P_0 + \left(\frac{t - t_0}{t_2 - t_0} \frac{t_2 - t}{t_2 - t_1} + \frac{t_3 - t}{t_3 - t_1} \frac{t - t_1}{t_2 - t_1} \right) P_1 + \frac{t - t_1}{t_3 - t_1} \frac{t - t_1}{t_2 - t_1} P_2 ,$$

This relation, when uniformly spaced t_i's are used (i.e., $t_i = i$ or $t_i = \frac{i}{3}$), leads to :

$$\gamma(t) = \frac{(2-t)^2}{2} P_0 + ((2-t)\frac{t}{2} + (3-t)\frac{t-1}{2})\, P_1 + \frac{(t-1)^2}{2} P_2 ,$$

and, in $[0,1]$, it is simply :

$$\gamma(t) = \frac{(1-t)^2}{2} P_0 + \frac{1 + 2t - 2t^2}{2} P_1 + \frac{t^2}{2} P_2 ,$$

and we return to the two previous remarks. The curve is $C^{m-1} = C^1$. Indeed, $\gamma(1) = \frac{P_1 + P_2}{2}$ and $\gamma'(1) = -P_1 + P_2$ in the first interval where this curve is defined ($[t_1, t_2]$). These values are identical to $\gamma(0)$ and to $\gamma'(0)$ respectively in the next interval ($[t_2, t_3]$).

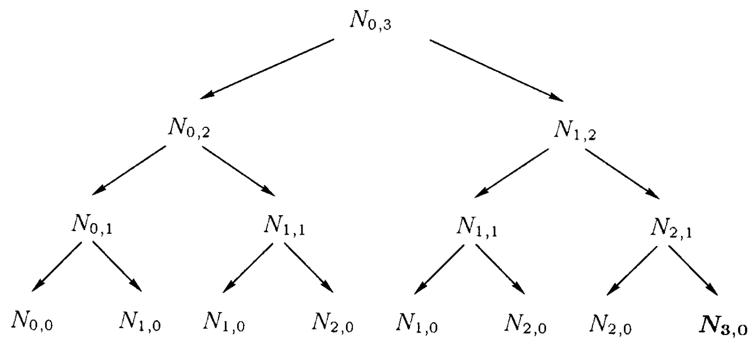

Figure 12.3: *Relationships between the $N_{i,k}$ from the root $N_{0,3}$ used to compute* $N_{0,3}$.

12.6.5 Degree 3 B-spline

In this case, the curve Γ is defined by :

$$\gamma(t) = \sum_{j=i-2}^{i+1} N_{j,3}(t)\, P_j\,. \tag{12.22}$$

Thus, the first interval where $\gamma(t)$ is well defined is $[t_2, t_3]$. To simplify the expression of $\gamma(t)$, we first consider the relationships between the $N_{i,k}$'s which are used in the present definition, *i.e.*, the $N_{i,k}$'s of interest when expanding the relation

$$N_{i,3}(t) = \frac{t - t_{i-1}}{t_{i+2} - t_{i-1}} N_{i,2}(t) + \frac{t_{i+3} - t}{t_{i+3} - t_i} N_{i+1,2}(t)\,.$$

The diagram relating these coefficients when evaluating the contribution of $N_{0,3}(t)$ is depicted in Figure 12.3. Since only $N_{3,0}(t)$ is non null, we just have to visit the branches (Figure 12.3) from the root to terminal node $N_{3,0}(t)$. We obtain successively :

$$N_{0,3}(t) = \frac{t_3 - t}{t_3 - t_0} N_{1,2}(t)\,,$$

$$N_{0,3}(t) = \frac{t_3 - t}{t_3 - t_0}\frac{t_3 - t}{t_3 - t_1} N_{2,1}(t)\,,$$

$$N_{0,3}(t) = \frac{t_3 - t}{t_3 - t_0}\frac{t_3 - t}{t_3 - t_2} N_{3,0}(t) \quad i.e., \quad \frac{t_3 - t}{t_3 - t_0}\frac{t_3 - t}{t_3 - t_2}\,.$$

For uniformly spaced t_i's, this relation reduces to

$$N_{0,3}(t) = \frac{(3-t)^3}{6}\,,$$

and, in the interval $[0,1]$, we obtain

$$N_{0,3}(t) = \frac{1}{6}(1-t)^3\,.$$

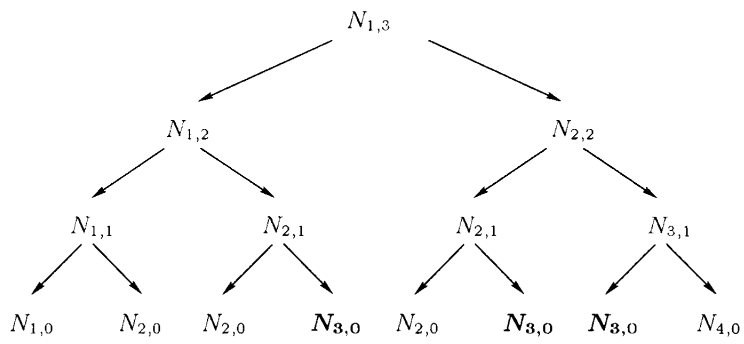

Figure 12.4: *Relationships between the $N_{i,k}$ from the root $N_{1,3}$ used to compute this value.*

To obtain the contribution of $N_{1,3}(t)$, we define the diagram with root $N_{1,3}(t)$ (Figure 12.4) and we examine the different branches leading to terminal node $N_{3,0}(t)$.

Thus, considering this tree, we obtain (after certain effort) :

$$N_{1,3}(t) = \frac{t - t_0}{t_3 - t_0}\frac{t_3 - t}{t_3 - t_1}\frac{t_3 - t}{t_3 - t_2} + \frac{t_4 - t}{t_4 - t_1}\left(\frac{t - t_1}{t_3 - t_1}\frac{t_3 - t}{t_3 - t_2} + \frac{t_4 - t}{t_4 - t_2}\frac{t - t_2}{t_3 - t_2}\right).$$

For a uniform spacing in t_i, we obtain :

$$N_{1,3}(t) = \frac{t\,(3-t)^2}{6} + \frac{4-t}{3}\left(\frac{(t-1)(3-t)}{2} + \frac{(4-t)(t-2)}{2}\right),$$

resulting, if $t \in [0,1]$, in :

$$N_{1,3}(t) = \frac{1}{6}(4 - 6t^2 + 3t^3)\,.$$

Similarly, we can obtain :

$$N_{2,3}(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3) \quad \text{and} \quad N_{3,3}(t) = \frac{1}{6}t^3\,.$$

These expressions assume that t lives in $[0,1]$ for any sub-interval. To complete this range of variation, a change variable must be used. For instance, let us consider the interval $[t_0, t_1, ..., t_4]$ where the curve is defined by the above relationships. Then, for interval $[t_0, t_1]$, we shift by t_0 and we scale by the factor $(t_1 - t_0)^{-1}$ to obtain the parameter t used in $N_{0,3}(t)$. For interval $[t_1, t_2]$, we shift by t_1 and we scale by the ratio $(t_2 - t_1)^{-1}$ to return to $N_{1,3}(t)$, and so on. Then $\gamma(t)$ is defined accordingly. Note that $\gamma(t)$ is C^2.

Exercise 12.5 *Check, for the three above B-splines, the C^{m-1} continuity.*

12.6.6 Specific controls

As previously suggested, some specific controls can be used. Basically, this relies on a proper definition of the nodes.

Multiple control points. Multiple vertices can be used to achieve the end termination of the curve definition (*i.e.*, γ passes through P_0 and P_n). They can also serve to control the shape of the curve at some neighborhood of a control point.

As an example, the uniform B-spline of degree 3 defined by the control points P_i for $i = 0, n$ does not pass through P_0 nor P_n. To complete this feature, one can define as vertices the sequence

$$P_0, P_0, P_0, P_1, P_2, ..., P_{n-1}, P_n, P_n, P_n ,$$

where P_0 and P_n both have a triple multiplicity.

Exercise 12.6 *Check that this definition of the control points insures that the curve passes through P_0 and P_n. Discuss the tangent at these endpoints.*

Phantom vertices. Phantom control points can be also defined in such a way as to obtain the previous property. In this way, fictitious vertices are defined before P_0 and after P_n. It is also possible to control a tangent in this way.

Multiple knots. Multiple knots (obviously, in a non uniform curve definition, see below for an example of such a definition) can also serve to control the corresponding curve. These multiple knots act in such a way as to modify the expression of the functions involved in the curve definition.

Each time a node t_i is repeated, the continuity level decreases by one at parameter t_i.

12.6.7 A Bézier curve by means of a B-Spline

In numerous applications, it is of interest to take a Bézier curve and to consider it as a B-Spline curve. As the B-Spline model is more general than the Bézier model, this operation is rather easy. Indeed, the control polygon associated with a Bézier curve is exactly the same if this curve is seen as a B-Spline. The only point to be ensured is that the nodes in the B-Spline are properly defined. To this end, one could use the following node sequence :

$$\{\underbrace{0, 0, ..., 0}_{n \text{ times}}, \underbrace{1, 1, ..., 1}_{n \text{ times}}\} .$$

Remark 12.21 *The Bézier curve is a polynomial of degree n in* $[0, 1]$. *Then, this continuity is infinite after n − 1 derivations, one obtains a zero value at any parameter t. If one selects the above sequence of nodes, consisting of two multiple nodes repeated n times, the continuity at $t = 0$ and at $t = 1$ is -1. In fact, the B-spline curve defined in this way stops at these parameter values and there is no continuity around $t = 0$ and $t = 1$.*

Exercise 12.7 *Show that in this case the B-Spline is a composite curve composed of one portion that is precisely the corresponding Bézier curve. Check that the recursion about a B-Spline curve is, in this case, equivalent to the recursion about a Bézier curve (based on the De Casteljau algorithm).*

12.6.8 Relationships between the parameters of a B-Splines

Throughout this section, some useful basis relationships are recalled which can be used when considering B-Splines curves. The following expressions provide a form for these relationships that make a B-Spline definition coherent. In these expressions,

- nk stands for the number of nodes in the node sequence,
- $\mathcal{C}(t)$ is the continuity range of the curve at parameter t. A negative value indicates that the continuity is not known at t,
- $\mathcal{M}(t)$ denotes the multiplicity of a parameter t in the node sequence. If the value of t is not in the node sequence, then $\mathcal{M}(t) = 0$. If the B-spline is uniform $\mathcal{M}(t_i) = 1$, for all t_i in the node sequence,
- np stands for the number of points defining the control polygon and
- m is the degree of the B-Spline.

Thus, we have :

$$order = m + 1$$

$$nk = m + 1 + np$$

$$\mathcal{C}(t) = order - \mathcal{M}(t) - 1$$

$$np \geq m + 1$$

12.7 Rational curves

To begin, we state what an homogeneous projection is.

Definition 12.1 *The* homogeneous projection *is an application T_h from $\mathbb{R}^{d+1}$ to $\mathbb{R}^d$ defined by :*

$$\{x_1\,\omega\ , x_2\,\omega\ , ...\ x_d\,\omega,\ \omega\} \mapsto \begin{cases} \textit{if } h = 0 \textit{ the point is reported at infinity} \\ \quad \textit{in the direction } \{x_1\ ,\ x_2\ ,\ ...\ x_d\} \\ \textit{otherwise } \{x_1\ ,\ x_2\ ,\ ...\ x_d\} \end{cases}$$

The main interest of such a projection is to allow the definition of a division by means of a projection. Hence, a rational curve in $\mathbb{R}^d$ is the image through such a projection of a polynomial curve in $\mathbb{R}^{d+1}$. The coordinate ω is called the *homogeneous coordinate* of the point under consideration.

12.7.1 Definition based on a control polygon

A *rational curve* Γ is defined by an equation like :

$$\gamma(t) = \frac{N(t)}{Q(t)}$$

where N and Q are some polynomials that can be described in terms of curves based on a control polygon. Thus, we have :

$$N(t) = \sum_{i=0}^{m} \varphi_i(t) P_i \quad \text{and} \quad Q(t) = \sum_{i=0}^{m} \varphi_i(t)\omega_i ,$$

which can also be written as :

$$\left\{ \begin{array}{c} N(t) \\ Q(t) \end{array} \right\} = \sum_{i=0}^{m} \varphi_i(t) \left\{ \begin{array}{c} P_i \\ \omega_i \end{array} \right\} .$$

This equation enables us to write Γ as :

$$\gamma(t) = T_h \left(\sum_{i=0}^{m} \varphi_i(t) \left\{ \begin{array}{c} P_i \\ \omega_i \end{array} \right\} \right) .$$

As a consequence, a rational curve is the homogeneous projection of a curve in $\mathbb{R}^{d+1}$ that is based on a control polygon in $\mathbb{R}^d$.

12.7.2 Rational Bézier curve

Following the above remark regarding conics, a rational Bézier definition can be defined. This representation involves the previous input ($n+1$ control points that form a control polygon and the corresponding parameter values) along with a sequence of weights ω_i. It is written as

$$\gamma(t) = \frac{\sum_{i=0}^{n} \omega_i B_{i,n}(t) P_i}{\sum_{i=0}^{n} \omega_i B_{i,n}(t)} . \tag{12.23}$$

Remark 12.22 *The question is how to define the t_i's as well as the ω_i's. For the t_i's, we return to the previous remarks. Regarding the weights, they are mostly used to give some preferences to some control points. A weight can be seen as a shape parameter. Actually, increasing the value of ω_i leads to pulling the curve towards the corresponding P_i.*

Remark 12.23 *If the weights are equal (for example, equal to one), we find again the classical definition of a non-rational Bézier curve (as $\sum_{i=0}^{n} B_{i,n}(t) = 1$).*

Exercise 12.8 *Prove that the $B_{i,n}(t)$'s sum to 1 (Hint : return to the recursion about the $B_{i,n}(t)$'s).*

As for Bézier curves, recursion formulas can be found for rational Bézier curves. Nevertheless, these recursions are a little complex (at least, in terms of notations). For instance, a rational Bézier curve may be evaluated by applying the De Casteljau form to both numerator and denominator of the above general expression.

While suitable for handling conics, rational Bézier curves have the same drawbacks as standard Bézier curves. In particular, a large value of n leads to the same remarks as above. For this reason, other functions have been developed, based on local definitions that enjoy the nice properties of the above *global* definitions while avoiding their disadvantages. A first approach involves using a standard (or rational) Bézier definition locally leading to a so-called composite method.

12.7.3 Rational B-spline curve (NURBS)

Similarly, a rational B-spline can be constructed using locally rational spline curves. This definition involves a sequence of weights ω_i and is written as

$$\gamma(t) = \frac{\sum_{i=0}^{n} \omega_i N_{i,m}(t) P_i}{\sum_{j=0}^{n} \omega_j N_{j,m}(t)} . \tag{12.24}$$

Note that NURBS stands for non uniform rational B-splines where the non uniformity concerns the t_i's. Such a distribution, unlike a uniform distribution, allows for a greater flexibility and may also be used for other kinds of curve representation.

In the following, we give an example of a quadratic NURBS. To this end, we return to a quadratic B-spline before defining the NURBS.

Non uniform quadratic B-spline with multiple nodes. We return to the case of a quadratic B-spline whose explicit form has been established previously in the case of distinct knots (see Section (12.6.2)). We now review the general formula for the $N_{i,j}$'s involved in this B-spline.

$$N_{i,2}(t) = \frac{t-t_{i-1}}{t_{i+1}-t_{i-1}} \frac{t-t_{i-1}}{t_i-t_{i-1}} N_{i,0}(t)$$
$$+ \left(\frac{t-t_{i-1}}{t_{i+1}-t_{i-1}} \frac{t_{i+1}-t}{t_{i+1}-t_i} + \frac{t_{i+2}-t}{t_{i+2}-t_i} \frac{t-t_i}{t_{i+1}-t_i} \right) N_{i+1,0}(t)$$
$$+ \frac{t_{i+2}-t}{t_{i+2}-t_i} \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} N_{i+2,0}(t) .$$

In the case of multiple nodes, this general relation reduces, for example, if $t_i = t_{i-1}$, we have $N_{i,0}(t) = 0$ and

$$N_{i,2}(t) = \left(\frac{t-t_{i-1}}{t_{i+1}-t_{i-1}} \frac{t_{i+1}-t}{t_{i+1}-t_i} + \frac{t_{i+2}-t}{t_{i+2}-t_i} \frac{t-t_i}{t_{i+1}-t_i} \right) N_{i+1,0}(t)$$

$$+ \frac{t_{i+2}-t}{t_{i+2}-t_i}\frac{t_{i+2}-t}{t_{i+2}-t_{i+1}}N_{i+2,0}(t).$$

and so on. For instance if $t_{i+2} = t_{i+1} = t_i$, we have simply :

$$N_{i,2}(t) = \frac{t-t_{i-1}}{t_{i+1}-t_{i-1}}\frac{t-t_{i-1}}{t_i-t_{i-1}}N_{i,0}(t).$$

We define a sequence of knots t_i for $i = 0, ..., 9$ as follows :

$$t_i = [0,0,0,1,1,2,2,3,3,3].$$

As a consequence we have successively $N_{1,0}(t) = N_{2,0}(t) = N_{4,0}(t) = N_{6,0}(t) = N_{8,0}(t) = N_{9,0}(t) = 0$. It is easy to obtain the $N_{i,2}$'s. In our example, we have, for $i = 0$, $N_{0,2}(t) = 0$. For $i = 1$, we find :

$$N_{1,2}(t) = \frac{t_{i+2}-t}{t_{i+2}-t_i}\frac{t_{i+2}-t}{t_{i+2}-t_{i+1}}N_{3,0}(t) = (1-t)^2 N_{3,0}(t).$$

For $i = 2$ and the following value of i, we have :

$$N_{2,2}(t) = \left(\frac{t-t_{i-1}}{t_{i+1}-t_{i-1}}\frac{t_{i+1}-t}{t_{i+1}-t_i} + \frac{t_{i+2}-t}{t_{i+2}-t_i}\frac{t-t_i}{t_{i+1}-t_i}\right)N_{3,0}(t) = 2t(1-t)N_{3,0}(t)$$

$$N_{3,2}(t) = \frac{t-t_{i-1}}{t_{i+1}-t_{i-1}}\frac{t-t_{i-1}}{t_i-t_{i-1}}N_{3,0}(t) + \frac{t_{i+2}-t}{t_{i+2}-t_i}\frac{t_{i+2}-t}{t_{i+2}-t_{i+1}}N_{5,0}(t).$$

$$N_{3,2}(t) = t^2 N_{3,0}(t) + (2-t)^2 N_{5,0}(t).$$

$$N_{4,2}(t) = \left(\frac{t-t_{i-1}}{t_{i+1}-t_{i-1}}\frac{t_{i+1}-t}{t_{i+1}-t_i} + \frac{t_{i+2}-t}{t_{i+2}-t_i}\frac{t-t_i}{t_{i+1}-t_i}\right)N_{5,0}(t)$$

$$N_{4,2}(t) = 2(t-1)(2-t)N_{5,0}(t)$$

$$N_{5,2}(t) = \frac{t-t_{i-1}}{t_{i+1}-t_{i-1}}\frac{t-t_{i-1}}{t_i-t_{i-1}}N_{5,0}(t) + \frac{t_{i+2}-t}{t_{i+2}-t_i}\frac{t_{i+2}-t}{t_{i+2}-t_{i+1}}N_{7,0}(t).$$

$$N_{5,2}(t) = (t-1)^2 N_{5,0}(t) + (3-t)^2 N_{7,0}(t).$$

$$N_{6,2}(t) = \left(\frac{t-t_{i-1}}{t_{i+1}-t_{i-1}}\frac{t_{i+1}-t}{t_{i+1}-t_i} + \frac{t_{i+2}-t}{t_{i+2}-t_i}\frac{t-t_i}{t_{i+1}-t_i}\right)N_{7,0}(t)$$

$$N_{6,2}(t) = 2(t-2)(3-t)N_{7,0}(t)$$

$$N_{7,2}(t) = \frac{t-t_{i-1}}{t_{i+1}-t_{i-1}}\frac{t-t_{i-1}}{t_i-t_{i-1}}N_{7,0}(t)$$

$$N_{7,2}(t) = (t-2)^2 N_{7,0}(t).$$

Written in terms of the interval $[0,1]$, we obtain :

$$N_{1,2}(t) = (1-t)^2 N_{3,0}(t).$$

$$N_{2,2}(t) = 2t(1-t)N_{3,0}(t)$$

$$N_{3,2}(t) = t^2 N_{3,0}(t) + (1-t)^2 N_{5,0}(t).$$

$$N_{4,2}(t) = 2t(1-t)N_{5,0}(t)$$

$$N_{5,2}(t) = t^2 N_{5,0}(t) + (1-t)^2 N_{7,0}(t).$$

$$N_{6,2}(t) = 2t(1-t)N_{7,0}(t)$$

$$N_{7,2}(t) = t^2 N_{7,0}(t).$$

The first part of the curve definition involves the sequence $[0,0,0,1]$, the second uses $[0,0,1,1]$, then we have successively $[0,1,1,2]$, $[1,1,2,2]$, $[1,2,2,3]$, $[2,2,3,3]$ and finally $[2,3,3,3]$.

An example of quadratic NURBS. Using the above B-spline, we give an example of a quadratic NURBS. In addition to the above sequence of knots, we consider a sequence of weights ω_i for $i = 1, ..., 7$ defined as :

$$\omega_i = [1, \frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}, 1].$$

Then, given the P_i,s, we consider the first segment where the NURBS is defined, say the portion corresponding to $[0,0,0,1]$. Following Relation (12.24), we obtain

$$\gamma(t) = \sum_{i=1}^{3}\left(\frac{\omega_i N_{i,2}(t) P_i}{\sum_{j=1}^{3}\omega_j N_{j,2}(t)}\right), \qquad (12.25)$$

which is :

$$\gamma(t) = \frac{(1-t)^2 P_1 + t(1-t)P_2 + t^2 P_3}{1-t-t^2}.$$

As P_i's, we take the following control polygon, $P_1 = {}^t(0,0)$, $P_2 = {}^t(1,0)$ and $P_3 = {}^t(\frac{1}{2}, \frac{\sqrt{3}}{2})$. Then, $\gamma(t)$ passes through P_1 and P_3, and $\gamma(t)$ is nothing other than a segment of the circle centered at point ${}^t(0, \frac{\sqrt{3}}{3})$ whose radius is $r = \frac{\sqrt{3}}{3}$.

Exercise 12.9 *Check that $\gamma(t)$ is the above portion of the previous circle.*

Indeed, we have defined a 120° sector of the full circle. To obtain the entire circle, we use the following control points P_3, P_4, P_5 and $P_5, P_6, P_7 = P_1$ as shown in Figure 12.5.

12.8 Curve definitions and numerical issues

Depending on the way the curve Γ is defined, the quantities we are interested in (length, tangent, normal, curvature, etc.) are more or less easy to compute. Moreover, only approximate values are obtained which are more or less accurate. This aspect is familiar to anyone who has made use of a curve (a surface) in a computer program and is clearly shown in the following with the help of some simple examples.

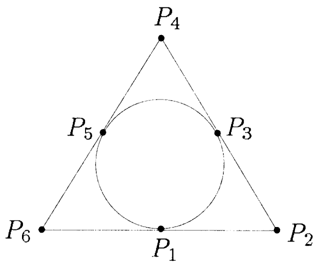

Figure 12.5: *The control polygon used to define a circle by a quadratic NURBS.*

In practice, Γ can be defined in terms of a parameter t, thus using a function $\gamma(t)$ or Γ can be known as a function of the curvilinear abscissa s by a relation like $\gamma(s)$. One could observe that, for a given Γ, these two functions, while denoted similarly, have different expressions.

12.8.1 An exact parameterization of a curve

We consider a very simple example. Let Γ be the quarter of a circle defined in terms of a parameter t ranging from 0 to $\frac{\pi}{2}$. Obviously the function[7] :

$$\gamma(t) = {}^t(r\,cos(t) \quad r\,sin(t))\,,$$

is an exact parameterization of the circle of radius r whose center is the origin. Then using the previous material (Chapter 11), we will compute the length of Γ and, for instance, its radius of curvature. We have :

$$\gamma'(t) = {}^t(-r\,sin(t) \quad r\,cos(t))\,,$$

then $||\gamma'(t)|| = r$ and, following Formula (11.1) : $s = \int_0^{\frac{\pi}{2}} r d\theta = r\frac{\pi}{2}$. To compute the curvature, we need to evaluate $\gamma''(t)$.

$$\gamma''(t) = {}^t(-r\,cos(t) \quad -r\,sint(t))\,,$$

then $\gamma'(t) \wedge \gamma''(t) = {}^t(0 \;\; 0 \;\; r^2)$ whose norm is r and the curvature is $C(t) = \frac{1}{r}$, meaning that the radius of curvature is nothing other than r, as expected. In conclusion, the a priori expected values, for both s and C, are obtained without any difficulty.

12.8.2 A given parameterization of a curve

We consider the same curve Γ as above. We define now an approximate representation of this curve using one of the methods discussed in this chapter (a composite curve). To this end, we assume that $A = {}^t(r \;\; 0)$, $B = {}^t(0 \;\; r)$ along

[7] The notation ${}^t(a \;\; b)$ stands for the vector whose components are a and b.

with[8] $\tau_A = {}^t(0 \;\; r\sqrt{2})$ and $\tau_B = {}^t(-r\sqrt{2} \;\; 0)$ are supplied and correspond to a parameter $t \in [0,1]$, *i.e.*, A (resp. B) and τ_A (resp. τ_B) correspond to $t = 0$ (resp. $t = 1$). Then, following Relationship (12.5), we can define a curve representation $\gamma(t)$ by :

$$\gamma(t) = A + \tau_A\, t + (3(B-A) - 2\tau_A - \tau_B)\; t^2 + (-2(B-A) + \tau_A + \tau_B)\; t^3\,,$$

then :

$$\gamma'(t) = \tau_A + 2\;(3(B-A) - 2\tau_A - \tau_B)\; t + 3\;(-2(B-A) + \tau_A + \tau_B)\; t^2\,,$$

which are such that $\gamma(0) = A$, $\gamma(1) = B$, $\gamma'(0) = \tau_A$ and $\gamma'(1) = \tau_B$. To obtain s between 0 and 1, the length of Γ, we have to compute $s(t) = \int_0^1 ||\gamma'(\theta)||\, d\theta$ which is :

$$\int_0^1 \sqrt{\langle\gamma'(\theta), \gamma'(\theta)\rangle}\, d\theta\,.$$

This expression involves the square root of a polynomial of degree four, which is therefore tedious to compute exactly. Thus, a quadrature may be used. Using a low degree formula, like the trapezoidal rule, leads to the following approximation

$$\int_0^1 \sqrt{\langle\gamma'(\theta), \gamma'(\theta)\rangle}\, d\theta = \frac{1}{2}\left(\sqrt{\langle\gamma'(0), \gamma'(0)\rangle} + \sqrt{\langle\gamma'(1), \gamma'(1)\rangle}\right) = r\sqrt{2}\,,$$

which is very imprecise (in particular, this length is equal to the length of segment AB) as might have been expected. By introducing the mid-value (say $t = \frac{1}{2}$) in the quadrature, we obtain :

$$s = \frac{1}{4}\left(\sqrt{\langle\gamma'(0), \gamma'(0)\rangle} + 2\sqrt{\langle\gamma'(\frac{1}{2}), \gamma'(\frac{1}{2})\rangle} + \sqrt{\langle\gamma'(1), \gamma'(1)\rangle}\right)\,,$$

i.e., about $1.517\, r$ which is again quite a bad evaluation. Introducing the quarters, we have now, for the length, a value close to $1.533\, r$ which again is not suitable. Using more nodes in the quadrature or using a more precise formula probably results in a better evaluation of the curve length but, on the other hand, leads to a large computational effort.

Indeed, the definition of the above control points and tangents leads to a curve which is not close enough to the real curve[9] and thus, the above rough approximation results in quite bad evaluations for the quantities we are interested in (the length in this case).

12.8.3 One curve parameterization with more precise definitions

We consider the same curve Γ and the same method of parameterization. But we now define this curve using its midpoint. We fix $r = 1$. In fact, we define half of the

[8] τ is short for $\vec{\tau}$.

[9] Anyway, we are trying to approach a circle by a polynomial of degree three !

-	-	2 nodes	3 nodes	5 nodes
2 control points	$[0, \frac{\pi}{2}]$	1.414	1.517	1.533
3 control points	$[0, \frac{\pi}{4}]$	1.5307	1.5598	1.5663
4 control points	$[0, \frac{\pi}{6}]$	1.5529	1.5661	1.5692
5 control points	$[0, \frac{\pi}{8}]$	1.5607	1.5682	1.57004
9 control points	$[0, \frac{\pi}{32}]$	1.5683	1.5701	1.57063

Table 12.3: Evaluation of the length of Γ as a function of its parameterization. The targeted value is $\frac{\pi}{2} \approx 1.5707$.

-	1 curve	2 curves	3 curves	4 curves	8 curves
$t = 0$	.63	.86	.93	.96	.9904
$t = .25$	1.04	1.01	1.008	1.004	1.0012
$t = .5$	1.31	1.07	1.03	1.019	1.0048

Table 12.4: Evaluation of the radius of curvature of Γ as a function of its parameterization. This radius must be constant and the targeted value is 1.

curve (*i.e.*, from 0 to $\frac{\pi}{4}$ in terms of angle). Then, we obtain for the length of the initial curve the values 1.5307, 1.5598 and 1.56633 based on the number of nodes in the quadrature. Splitting the initial curve into three composite curves (the first from 0 to $\frac{\pi}{6}$) we now have 1.5529, 1.5661 and 1.5692. Using four composite portions (the first from 0 to $\frac{\pi}{8}$), we have 1.5607, 1.5682 and 1.57004. The final example concerns eight composite curves (the first from 0 to $\frac{\pi}{32}$).

All these values are given in Table 12.3 in terms of the number of composite curves used (*i.e.*, how many control points and tangents have been used) and, for a given composite curve, in terms of the number of nodes used in the quadrature.

Similarly, using Relationship (11.4), we can evaluate the curvature of Γ as obtained when such or such number of composite curves are used to discretize the curve. To this end, we need to compute $\gamma''(t)$. For our example, this is simply

$$\gamma''(t) = 2\,(3(B-A) - 2\tau_A - \tau_B) + 6\,(-2(B-A) + \tau_A + \tau_B)\,t\,.$$

Table 12.4 presents the radius of curvature as a function of t for the different above parameterizations.

To sum up, both the lengths and the curvatures are obtained with a greater or lesser degree of accuracy. Approximate tangents and normals can also be obtained in the same way.

12.8.4 Using a polyline

The same exercise in the case where a polyline is given as the discrete definition of a curve Γ is now discussed. Let P_i, P_{i+1} be a segment of the polyline, $i = 0, n-1$, $\gamma(t)$ is defined as a curve passing through the P_i's. The length of $\gamma(t)$ is nothing other than the sum of the length of the $n+1$ segments constituting the polyline. As a function of n, we return to Table 12.3 (first row) to have an approximate

value of the length of Γ. The other characteristics of Γ can be obtained using the P_i, P_{i+1}'s.

If Γ is a curve in $\mathbb{R}^2$, a simple construction enables us to find approximate curvature, tangent and normal at some t.

Note that t can be chosen so that $t = 0$ for P_i and $t = 1$ for P_{i+1}, meaning that t is local to the segment under consideration. Another possible t definition globally considers the different segments and is based on their lengths, then we can obtain (after scaling) $t = 0$ for P_0 and $t = 1$ for P_n.

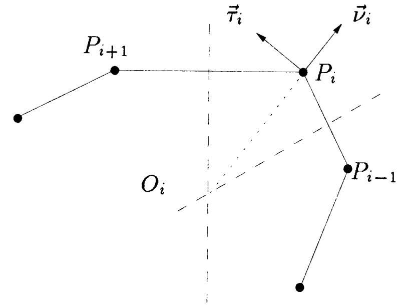

Figure 12.6: *Curvature, tangent and normal that can be obtained from a simple polyline (in two dimensions).*

We consider three consecutive vertices, P_{i-1}, P_i and P_{i+1}. We construct the perpendicular bisector of P_{i-1}, P_i and that of P_i, P_{i+1}. Provided the two segments are not aligned, they intersect in a point denoted as O_i, we define $r_i = ||O_iP_i||$ and $\vec{\nu}_i = \frac{\vec{P_iO_i}}{||P_iO_i||}$, then we introduce $\vec{\tau}_i$ as the unit vector perpendicular to $\vec{\nu}_i$. Indeed, using this simple construction results in approximate values for the center of curvature of Γ at point P_i, say O_i, the radius of curvature at this point (r_i), the unit normal and tangent at P_i ($\vec{\nu}_i$ and $\vec{\tau}_i$). Note that these quantities are not well defined at P_0 and P_n for an open polyline.

Thus, $\gamma(t)$ can be described in terms of these discrete quantities. In other words, local knowledge of the presumed curve can be accessed. For instance, one can use the osculating circle at point P_i and say that :

$$\gamma(t) \approx P_i + (t - t_i)\,\vec{\tau}_i + \frac{r_i}{2}(t - t_i)^2\,\vec{\nu}_i$$

in some vicinity of P_i, *i.e.*, for t close to t_i the parameter value associated with P_i.

To obtain a continuous definition of the curve, it is necessary to choose a mathematical representation. For instance, we can return to the above parameterization. In this case, a given characteristic can be evaluated not only at the segment endpoints but anywhere else as well (for any arbitrary value of parameter t).

For a curve in $\mathbb{R}^3$, we can also evaluate these characteristics. Indeed, the above geometric construction in two dimensions can be written in a more general form that extends to three dimensions.

First, we consider again two consecutive segments[10], P_{i-1}, P_i and P_i, P_{i+1}. We define the approximate normal at P_i as

$$\vec{\nu}_i \approx \frac{\dfrac{\overrightarrow{P_iP_{i-1}}}{||\overrightarrow{P_iP_{i-1}}||} + \dfrac{\overrightarrow{P_iP_{i+1}}}{||\overrightarrow{P_iP_{i+1}}||}}{\left\| \dfrac{\overrightarrow{P_iP_{i-1}}}{||\overrightarrow{P_iP_{i-1}}||} + \dfrac{\overrightarrow{P_iP_{i+1}}}{||\overrightarrow{P_iP_{i+1}}||} \right\|}. \tag{12.26}$$

Then we can obtain an approximation of r_i the radius of curvature at P_i. Actually, we can define :

$$\rho_i \approx \min_j \frac{1}{2} \frac{\langle \overrightarrow{P_iP_j}, \overrightarrow{P_iP_j} \rangle}{\langle \vec{\nu}_i, \overrightarrow{P_iP_j} \rangle}, \tag{12.27}$$

where $j = i-1$ or $i+1$. Note that a formula like :

$$\rho_i \approx \frac{1}{4} \left(\frac{\langle \overrightarrow{P_iP_{i-1}}, \overrightarrow{P_iP_{i-1}} \rangle}{\langle \vec{\nu}_i, \overrightarrow{P_iP_{i-1}} \rangle} + \frac{\langle \overrightarrow{P_iP_{i+1}}, \overrightarrow{P_iP_{i+1}} \rangle}{\langle \vec{\nu}_i, \overrightarrow{P_iP_{i+1}} \rangle} \right), \tag{12.28}$$

also provides an approximation of ρ_i.

Proof. Let M be the midpoint of P_iP_{i-1}, we construct O the point where $\vec{\nu}_i$ and the perpendicular bisector of P_iP_{i-1} intersect (thus a line passing through M). Let α be the angle formed between P_iM and $\vec{\nu}_i$, we have $\langle \overrightarrow{P_iM}, \vec{\nu}_i \rangle = ||\overrightarrow{P_iM}|| \cos\alpha$ and $\cos\alpha = \frac{||P_iM||}{r_i}$ from which Relationship (12.27) yields. □

Remark 12.24 *Relationships (12.26) and (12.27) extend to curves in three dimensions. They can also be extended in the case of a curve traced on a surface. In this case, a mean normal can be computed along with the minimum of the various radii of curvature that can be defined by considering the various P_j neighboring P_i.*

12.9 Towards a "pragmatic" curve definition ?

Due to the number and the variety of possible curve definitions and bearing in mind some numerical troubles that can happen, we now attempt to define a context which is well-suited to the meshing application at end. Basically, we are interested in curves from this viewpoint. Indeed, we want to mesh a curve (as will be seen in Chapter 14), that is in general the boundary of a domain (in the plane or in space). The aim is then to take this curve discretization as input data for the construction of a mesh of the domain of which this curve is a boundary.

It is then clear that a meshing process developed for this purpose must be as general and as stand-alone as possible while being reasonably simple. Thus, it seems unrealistic to guess that a meshing algorithm will know in detail *all* the curve representations that may be encountered (if so, we will face a "huge monster", tedious to design, hard to update, and so on.). For this reason, we want to simplify the context to that which is strictly necessary for the implementation of a curve meshing algorithm (the same is true for a surface meshing algorithm).

[10] One can consider the plan where P_{i-1} and P_{i+1} pass through.

12.9.1 Curve definitions and meshing topics

As previously seen, we encounter two types of representations for a curve. One approach makes use of a parametric definition while the other makes use of a discrete definition (*i.e.*, a mesh). In the first case, the curves result from a CAD system and actually exist only if it is possible to query this system in order to collect the useful geometric characteristics (by default to have the precise representation of the underlying model). In the second case, the discrete data is directly usable (given some assumptions) so as to collect the desired characteristics (or, at least, to find their approximate values).

Whether one or the other of these approaches is adopted, one can see that a meshing problem requires (for simplicity, let us just look at a curve meshing problem) :

- access to the CAD modeler used when defining the curve, in the first case, and
- an internal choice of a "CAD" enabling us to find the useful information from a discrete set of values, say a polyline, in the second case.

12.9.2 Key-ideas

The approach we would like to follow is to mesh a curve without explicitly using its CAD definition in order to be independent (obtaining thus some extent of generality and computational efficiency).

Thus, we adopt a rather different point of view from a classical approach. Our goal is to be independent of such or such curve definitions (and thus of the CAD system the user is familiar with) when a meshing task is demanded. To this end, the key-idea is as follows :

- the curve is defined by a sequence of control points and *possibly* by some other information,
- the curve passes through these points (in this sense, an interpolation type is retained),
- afterwards, a mathematical support is defined, using the points (and the extra information, if any) by means of a unique composite curve definition of low degree,
- then this mathematical definition becomes the curve definition and is used to define what is necessary for our purpose.

Following these principles, the user's responsibility (and thus that of the CAD system) is limited to providing a sequence of points which makes the construction of a sufficiently precise definition of the geometry possible.

From this point of view, a reasonable synthetic scheme for curve definition could be composed of two different steps :

Step 1 : (user's responsibility) use the CAD system available to construct a polyline while giving the corners[11] if any.

Step 2 : (pre-meshing process responsibility) use the previous polyline to construct a fixed geometric definition or use this polyline directly.

In this way, the process of meshing a curve (see Chapter 14) is based solely on the above geometric definition (and the original CAD definition is no longer considered).

12.9.3 Construction of a well-suited discrete definition

In practice, this issue is the key to obtaining a curve definition that is sufficiently accurate. The problem reduces to finding an appropriate polyline to represent the geometry.

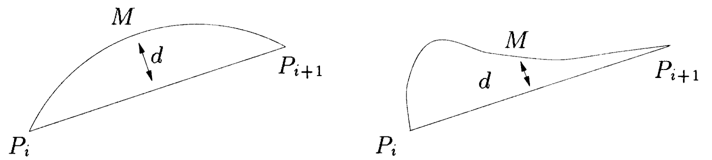

Figure 12.7: *Controlling the gap between a polyline and a curve.*

A rather obvious method can be used to control the accuracy of a "polylinization" of a curve. Using his favorite CAD system, the user defines one member of the desired polyline. Let P_i, P_{i+1} be this segment. Let t_i and t_{i+1} be the two parameter values corresponding to P_i and P_{i+1}. It is easy to obtain point M corresponding to $t = \frac{t_i+t_{i+1}}{2}$. Then one computes d the distance between M and P_i, P_{i+1}. Let δ be a threshold value, if $d < \delta \, ||P_i, P_{i+1}||$ then the current segment is judged correct. Otherwise, point M is used to define the two segments P_i, M and M, P_{i+1} which are in turn analyzed in the same way to decide whether or not additional subdivisions are required.

Note that, in general, the maximum gap is not necessarily obtained at the midpoint as defined above (see Figure 12.7). Thus, to prevent a bad-capture of the curve, sample points can be tested to decide whether the current segment is sufficiently fine.

[11] If the corners are not explicitly supplied, it is nevertheless possible to guess them. To this end, an analysis of the angles between two consecutive segments is made. Note that a threshold value must be used which may lead to certain ambiguities in some cases.

A brief conclusion

As we have just seen, there are various curve definitions each of which offers both advantages and weaknesses. It is therefore no easy matter to decide which definition is the most appropriate in general and, anyway, such a discussion is beyond the scope of this book. However, if one idea is to be retained after this discussion, we think it is clear that the adopted meshing techniques for curves (surfaces, in a similar way) must be, whenever possible, developed without regard to the model used during the design of the curve[12] (the surface).

On the other hand, the few examples of actual computation regarding curves defined by one or other of the models seen above have brought to light some perverse numerical effects which will required further consideration.

[12] In specific, we would like to start from a minimal interface under the responsibility of the CAD system.