Chapter 18

Polynomial Curves

18.1 Polar Forms and Control Points

The purpose of this short chapter is to show how polynomial curves are handled in terms of control points. This is a very nice application of affine concepts discussed in previous chapters and provides a stepping stone for the study of rational curves. This chapter is just a brief introduction. A comprehensive treatment of polynomial curves can be found in Gallier [70]).

The key to the treatment of polynomial curves in terms of control points is that polynomials can be multilinearized.[1] To be more precise, say that a map $f\colon \underbrace{\mathbb{R}^d \times \cdots \times \mathbb{R}^d}_{m} \to \mathbb{R}^n$ is *multiaffine* if it is affine in each of its arguments, and that a map $f\colon \underbrace{\mathbb{R}^d \times \cdots \times \mathbb{R}^d}_{m} \to \mathbb{R}^n$ is *symmetric* if it does not depend on the order of its arguments, i.e., $f(a_{\pi(1)}, \ldots, a_{\pi(m)}) = f(a_1, \ldots, a_m)$ for all $a_1, \ldots, a_m$, and all permutations π. Then, for every polynomial $F(t)$ of degree m, there is a unique symmetric and multiaffine map $f\colon \underbrace{\mathbb{R} \times \cdots \times \mathbb{R}}_{m} \to \mathbb{R}$ such that

$$F(t) = f(\underbrace{t, \ldots, t}_{m}), \quad \text{for all } t \in \mathbb{R}.$$

This is an old "folk theorem", probably already known to Newton. The proof is easy. By linearity, it is enough to consider a monomial of the form x^k, where $k \leq m$. The unique symmetric multiaffine map corresponding to x^k is

$$\frac{\sigma_k(t_1, \ldots, t_m)}{\binom{m}{k}},$$

where $\sigma_k(t_1, \ldots, t_m)$ is the kth elementary symmetric function in m variables, i.e.

$$\sigma_k = \sum_{\substack{I \subseteq \{1, \ldots, m\} \\ |I| = k}} \left(\prod_{i \in I} t_i \right).$$

Given a polynomial curve $F\colon \mathbb{R} \to \mathbb{R}^n$ of degree m

$$\begin{aligned} x_1(t) &= F_1(t), \\ \ldots &= \ldots \\ x_n(t) &= F_n(t), \end{aligned}$$

[1]The term "multilinearized" is technicaly incorrect, we should say "multiaffinized"!

where $F_1(t), \ldots, F_n(t)$ are polynomials of degree at most m, $F\colon \mathbb{R} \to \mathbb{R}^n$ arises from a unique symmetric multiaffine map $f\colon \mathbb{R}^m \to \mathbb{R}^n$, the *polar form of* F, such that

$$F(t) = f(\underbrace{t, \ldots, t}_{m}),$$

for all $t \in \mathbb{R}$ (see Ramshaw [141], Farin [58, 57], Hoschek and Lasser [90], or Gallier [70]). For example, consider the plane cubic defined as follows:

$$F_1(t) = \frac{3}{4}t^2 - \frac{3}{2}t - \frac{9}{4}, \qquad F_2(t) = \frac{3}{4}t^3 - \frac{3}{2}t^2 - \frac{9}{4}t.$$

We get the polar forms

$$\begin{aligned} f_1(t_1, t_2, t_3) &= \frac{1}{4}(t_1t_2 + t_1t_3 + t_2t_3) - \frac{1}{2}(t_1 + t_2 + t_3) - \frac{9}{4} \\ f_2(t_1, t_2, t_3) &= \frac{3}{4}t_1t_2t_3 - \frac{1}{2}(t_1t_2 + t_1t_3 + t_2t_3) - \frac{3}{4}(t_1 + t_2 + t_3). \end{aligned}$$

Also, for $r \neq s$, the map $f\colon \mathbb{R}^m \to \mathbb{R}^n$ is determined by the $m+1$ *control points* $(b_0, \ldots, b_m)$, where

$$b_i = f(\underbrace{r, \ldots, r}_{m-i}, \underbrace{s, \ldots, s}_{i}),$$

since

$$f(t_1, \ldots, t_m) = \sum_{k=0}^{m} \sum_{\substack{I \cup J = \{1,\ldots,m\} \\ I \cap J = \emptyset,\, \mathrm{card}(J) = k}} \prod_{i \in I} \left(\frac{s - t_i}{s - r}\right) \prod_{j \in J} \left(\frac{t_j - r}{s - r}\right) f(\underbrace{r, \ldots, r}_{m-k}, \underbrace{s, \ldots, s}_{k}).$$

For example, with respect to the affine frame $r = -1$, $s = 3$, the coordinates of the control points of the cubic defined earlier are:

$$\begin{aligned} b_0 &= (0, 0) \\ b_1 &= (-4, 4) \\ b_2 &= (-4, -12) \\ b_3 &= (0, 0). \end{aligned}$$

Conversely, for every sequence of $m+1$ points $(b_0, \ldots, b_m)$, there is a unique symmetric multiaffine map f such that

$$b_i = f(\underbrace{r, \ldots, r}_{m-i}, \underbrace{s, \ldots, s}_{i}),$$

namely

$$f(t_1, \ldots, t_m) = \sum_{k=0}^{m} \sum_{\substack{I \cup J = \{1,\ldots,m\} \\ I \cap J = \emptyset,\, \mathrm{card}(J) = k}} \prod_{i \in I} \left(\frac{s - t_i}{s - r}\right) \prod_{j \in J} \left(\frac{t_j - r}{s - r}\right) b_k.$$

Thus, there is a bijection between the set of polynomial curves of degree m and the set of sequences $(b_0, \ldots, b_m)$ of $m+1$ control points.

The figure below shows four contol points b_0, b_1, b_2, b_3 specifying a polynomial curve of degree 3, where $b_0 = f(r, r, r)$, $b_1 = f(r, r, s)$, $b_2 = f(r, s, s)$, $b_3 = f(s, s, s)$.

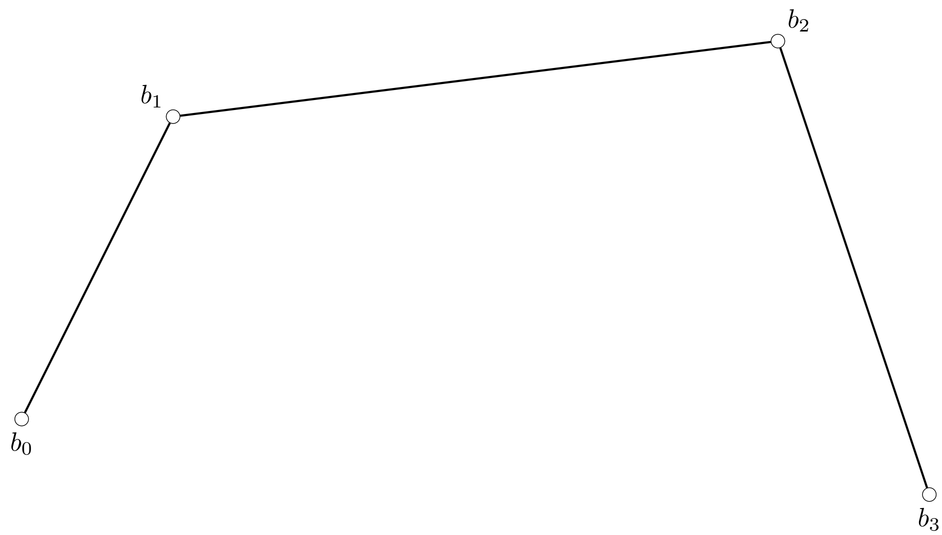

Figure 18.1: Control points and control polygon

The upshot of all this is that for algorithmic purposes, it is convenient to define polynomial curves in terms of polar forms. Recall that the canonical affine space associated with the field $\mathbb{R}$ is denoted as $\mathbb{A}$, unless confusions arise.

Definition 18.1.1 A (parameterized) *polynomial curve in polar form of degree m* is an affine polynomial map $F\colon \mathbb{A} \to \mathcal{E}$ of polar degree m, defined by its *m-polar form*, which is some symmetric m-affine map $f\colon \mathbb{A}^m \to \mathcal{E}$, where $\mathbb{A}$ is the real affine line, and $\mathcal{E}$ is any affine space (of dimension at least 2). Given any $r, s \in \mathbb{A}$, with $r < s$, a (parameterized) *polynomial curve segment $F([r, s])$ in polar form of degree m* is the restriction $F\colon [r, s] \to \mathcal{E}$ of an affine polynomial curve $F\colon \mathbb{A} \to \mathcal{E}$ in polar form of degree m. We define the *trace of F* as $F(\mathbb{A})$, and the the *trace of $F[r, s]$* as $F([r, s])$.

Typically, the affine space $\mathcal{E}$ is the real affine space $\mathbb{A}^3$ of dimension 3.

Remark: When defining polynomial curves, it is convenient to denote the polynomial map defining the curve by an upper-case letter, such as $F\colon \mathbb{A} \to \mathcal{E}$, and the polar form of F by the same, but lower-case letter, f. It would then be confusing to denote the affine space which is the range of the maps F and f also as F, and thus, we denote it as $\mathcal{E}$ (or at least, we use a letter different from the letter used to denote the polynomial map defining the curve). Also note that we defined a polynomial curve in polar form of degree at most m, rather than a polynomial curve in polar form of degree exactly m, because an affine polynomial map f of polar degree m may end up being degenerate, in the sense that it could be equivalent to a polynomial map of lower polar degree. For convenience, we will allows ourselves the abuse of language where we abbreviate "polynomial curve in polar form" to "polynomial curve".

We summarize the relationship between control points and polynomial curves in the following lemma.

Lemma 18.1.2 *Given any sequence of $m + 1$ points $a_0, \ldots, a_m$ in some affine space $\mathcal{E}$, there is a unique polynomial curve $F\colon \mathbb{A} \to \mathcal{E}$ of degree m, whose polar form $f\colon \mathbb{A}^m \to \mathcal{E}$ satisfies the conditions*

$$f(\underbrace{r, \ldots, r}_{m-k}, \underbrace{s, \ldots, s}_{k}) = a_k,$$

(where $r, s \in \mathbb{A}$, $r \neq s$). Furthermore, the polar form f of F is given by the formula

$$f(t_1, \ldots, t_m) = \sum_{k=0}^{m} \sum_{\substack{I \cup J = \{1, \ldots, m\} \\ I \cap J = \emptyset,\, |J| = k}} \prod_{i \in I} \left(\frac{s - t_i}{s - r} \right) \prod_{j \in J} \left(\frac{t_j - r}{s - r} \right) a_k,$$

and $F(t)$ is given by the formula

$$F(t) = \sum_{k=0}^{m} B_k^m[r, s](t)\, a_k,$$

where the polynomials

$$B_k^m[r, s](t) = \binom{m}{k} \left(\frac{s - t}{s - r} \right)^{m-k} \left(\frac{t - r}{s - r} \right)^{k}$$

are the Bernstein polynomials of degree m over $[r, s]$.

Note that since the polar form f of a polynomial curve F of degree m is symmetric, the order of the arguments is irrelevant. Often, when argument are repeated, we also omit commas between argument. For example, we abbreviate $f(\underbrace{r, \ldots, r}_{i}, \underbrace{s, \ldots, s}_{j})$ as $f(r^i s^j)$.

In the next section, we will abbreviate $f(\underbrace{t, \ldots, t}_{j}, \underbrace{r, \ldots, r}_{m-i-j}, \underbrace{s, \ldots, s}_{i})$ as $f(t^j r^{m-i-j} s^i)$.

18.2 The de Casteljau Algorithm

The definition of polynomial curves in terms of polar forms leads to a very nice algorithm known as the *de Casteljau algorithm*, to draw polynomial curves. Using the de Casteljau algorithm, it is possible to determine any point $F(t)$ on the curve, by repeated affine interpolations (see Farin [58, 57], Hoschek and Lasser [90], Risler [142], or Gallier [70]). The example below shows $F(1/2)$.

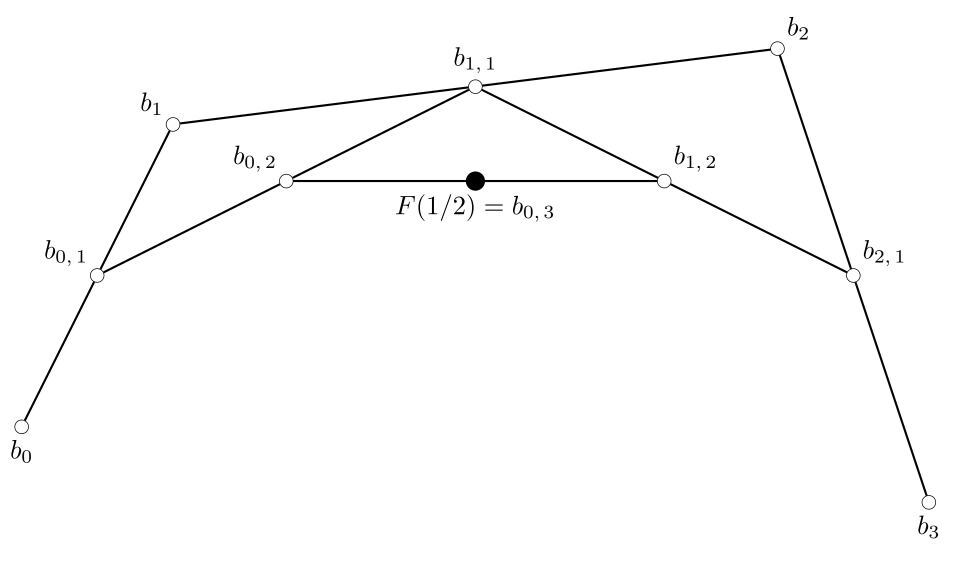

Figure 18.2: A de Casteljau diagram for $t = 1/2$

In the general case where a curve F is specified by $m+1$ control points $(b_0, \ldots, b_m)$ w.r.t. to an interval $[r, s]$, let us define the following points $b_{i,j}$ used during the computation of $F(t)$ (where f is the polar form of F):

$$b_{i,j} = \begin{cases} b_i & \text{if } j = 0,\ 0 \leq i \leq m, \\ f(t^j r^{m-i-j} s^i) & \text{if } 1 \leq j \leq m,\ 0 \leq i \leq m-j. \end{cases}$$

Then, we have the following equations:

$$\boxed{b_{i,j} = \left(\frac{s-t}{s-r}\right) b_{i,j-1} + \left(\frac{t-r}{s-r}\right) b_{i+1,j-1}.}$$

The result is

$$F(t) = b_{0,m}.$$

The computation can be conveniently represented in the following triangular form:

$$\begin{array}{ccccccccc}
0 & 1 & \ldots & j-1 & j & \ldots & m-k & \ldots & m \\
b_{0,0} & & & & & & & & \\
 & b_{0,1} & & & & & & & \\
b_{1,0} & & \ddots & & & & & & \\
 & & & b_{0,j-1} & & & & & \\
 & & & \vdots & b_{0,j} & & & & \\
 & & & b_{i,j-1} & \vdots & \ddots & & & \\
 & & & & b_{i,j} & & b_{0,m-k} & & \\
 & & & b_{i+1,j-1} & \vdots & & & & \\
 & & & \vdots & b_{m-k-j,j} & & \vdots & & b_{0,m} \\
 & & & b_{m-k-j+1,j-1} & & & & & \\
 & & & & \vdots & & b_{k,m-k} & & \\
 & b_{m-k-1,1} & & \vdots & & & & & \\
b_{m-k,0} & & & & b_{m-j,j} & & & & \\
 & \vdots & & b_{m-j+1,j-1} & & & & & \\
\vdots & & & & & & & & \\
b_{m-1,0} & & & & & & & & \\
 & b_{m-1,1} & & & & & & & \\
b_{m,0} & & & & & & & &
\end{array}$$

When $r \leq t \leq s$, each interpolation step computes a convex combination, and $b_{i,j}$ lies between $b_{i,j-1}$ and $b_{i+1,j-1}$. In this case, geometrically, the algorithm consists of a diagram consisting of the m polylines

$$(b_{0,0},\, b_{1,0}),\ (b_{1,0},\, b_{2,0}),\ (b_{2,0},\, b_{3,0}),\ (b_{3,0},\, b_{4,0}),\ \ldots,(b_{m-1,0},\, b_{m,0})$$

$$\begin{gathered}(b_{0,1},\, b_{1,1}),\, (b_{1,1},\, b_{2,1}),\, (b_{2,1},\, b_{3,1}),\, \ldots,(b_{m-2,1},\, b_{m-1,1})\\ (b_{0,2},\, b_{1,2}),\, (b_{1,2},\, b_{2,2}),\, \ldots,(b_{m-3,2},\, b_{m-2,2})\\ \ldots\\ (b_{0,m-2},\, b_{1,m-2}),\, (b_{1,m-2},\, b_{2,m-2})\\ (b_{0,m-1},\, b_{1,m-1})\end{gathered}$$

called *shells*, and with the point $b_{0,m}$, they form the *de Casteljau diagram*. Note that the shells are nested nicely. The polyline

$$(b_0,\, b_1),\, (b_1,\, b_2),\, (b_2,\, b_3),\, (b_3,\, b_4),\, \ldots,(b_{m-1},\, b_m)$$

is also called a *control polygon* of the curve. When t is outside $[r, s]$, we still obtain m shells and a de Casteljau diagram, but the shells are not nicely nested.

One of the best features of the de Casteljau algorithm is that it lends itself very well to recursion. Indeed, going back to the case of a cubic curve, it is easy to show that the sequences of points $(b_0, b_{0,1}, b_{0,2}, b_{0,3})$ and $(b_{0,3}, b_{1,2}, b_{2,1}, b_3)$ are also control polygons for the exact same curve (see Farin [58, 57], Hoschek and Lasser [90], Gallier [70]). Thus, we can compute the points corresponding to $t = 1/2$ with respect to the control polygons

$$(b_0, b_{0,1}, b_{0,2}, b_{0,3}) \quad \text{and} \quad (b_{0,3}, b_{1,2}, b_{2,1}, b_3),$$

and this yields a recursive method for approximating the curve. This method called the *subdivision method* applies to polynomial curves of any degree and can be used to render efficiently a curve segment F over $[r, s]$.

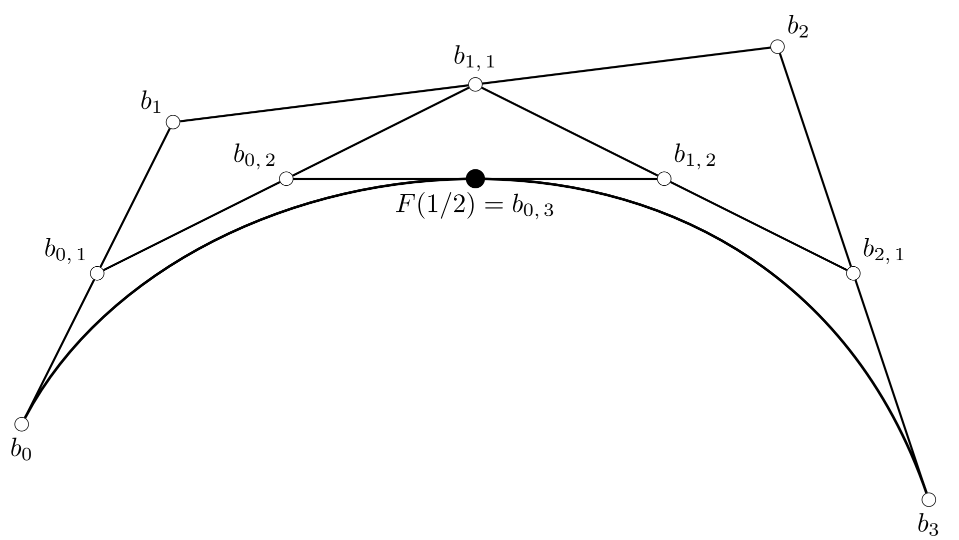

Figure 18.3: Approximating a curve using subdivision

For much more on polynomial curves, see Gallier [70].

Chapter 19

Polynomial Surfaces

19.1 Polar Forms

The purpose of this short chapter is to show how polynomial surfaces are handled in terms of control points. As Chapter 18, this Chapter is just a brief introduction and a stepping stone for the study of rational surfaces. A comprehensive treatment of polynomial surfaces can be found in Gallier [70]).

The deep reason why polynomial surfaces can be effectively handled in terms of control points is that multivariate polynomials arise from multiaffine symmetric maps (see Ramshaw [141], Farin [58, 57], Hoschek and Lasser [90], or Gallier [70]). Denoting the affine plane $\mathbb{R}^2$ as $\mathcal{P}$, traditionally, a *polynomial surface* in $\mathbb{R}^n$ is a function $F\colon \mathcal{P} \to \mathbb{R}^n$, defined such that

$$\begin{aligned} x_1 &= F_1(u, v),\\ \ldots &= \ldots\\ x_n &= F_n(u, v), \end{aligned}$$

for all $(u, v) \in \mathbb{R}^2$, where $F_1(U, V), \ldots, F_n(U, V)$ are polynomials in $\mathbb{R}[U, V]$.

There are two natural ways to polarize the polynomials defining F. The first way is to polarize *separately* in u and v. If p is the highest degree in u and q is the highest degree in v, we get a unique multiaffine map

$$f\colon (\mathbb{R})^p \times (\mathbb{R})^q \to \mathbb{R}^n$$

of degree $(p+q)$ which is symmetric in its first p arguments and symmetric in its last q arguments, such that

$$F(u, v) = f(\underbrace{u, \ldots, u}_{p}; \underbrace{v, \ldots, v}_{q}).$$

We get what is traditionally called a *tensor product surface*, or as we prefer to call it, a *bipolynomial surface of bidegree* $\langle p, q\rangle$ (or a *rectangular surface patch*). We also say that the multiaffine maps arising in polarizing separately in u and v are $\langle p, q\rangle$*-symmetric*.

The second way to polarize is to treat the variables u and v *as a whole*. This way, if F is a polynomial surface such that the maximum total degree of the monomials is m, we get a unique symmetric degree m multiaffine map

$$f\colon (\mathbb{R}^2)^m \to \mathbb{R}^n,$$

such that

$$F(u, v) = f(\underbrace{(u, v), \ldots, (u, v)}_{m}).$$

We get what is called a *total degree surface* (or a *triangular surface patch*).

Using linearity, it is clear that all we have to do is to polarize a monomial $u^h v^k$. It is easily verified that the unique $\langle p, q\rangle$-symmetric multiaffine polar form of degree $p+q$

$$f_{h,k}^{p,q}(u_1,\ldots,u_p;v_1,\ldots,v_q)$$

of the monomial $u^h v^k$ is given by

$$f_{h,k}^{p,q}(u_1,\ldots,u_p;v_1,\ldots,v_q)=\frac{1}{\binom{p}{h}\binom{q}{k}}\sum_{\substack{I\subseteq\{1,\ldots,p\},|I|=h\\ J\subseteq\{1,\ldots,q\},|J|=k}}\left(\prod_{i\in I}u_i\right)\left(\prod_{j\in J}v_j\right).$$

The denominator $\binom{p}{h}\binom{q}{k}$ is the number of terms in the above sum.

It is also easily verified that the unique symmetric multiaffine polar form of degree m

$$f_{h,k}^{m}((u_1,v_1),\ldots,(u_m,v_m))$$

of the monomial $u^h v^k$ is given by

$$f_{h,k}^{m}((u_1,v_1),\ldots,(u_m,v_m))=\frac{1}{\binom{m}{h}\binom{m-h}{k}}\sum_{\substack{I\cup J\subseteq\{1,\ldots,m\}\\ |I|=h,|J|=k,I\cap J=\emptyset}}\left(\prod_{i\in I}u_i\right)\left(\prod_{j\in J}v_j\right).$$

The denominator $\binom{m}{h}\binom{m-h}{k}=\binom{m}{h\;k\;(m-h-k)}$ is the number of terms in the above sum.

As an example, consider the following surface known as Enneper's surface:

$$\begin{aligned}F_1(U,V)&=U-\frac{U^3}{3}+UV^2\\ F_2(U,V)&=V-\frac{V^3}{3}+U^2V\\ F_3(U,V)&=U^2-V^2.\end{aligned}$$

We get the polar forms

$$\begin{aligned}f_1((U_1,V_1),(U_2,V_2),(U_3,V_3))&=\frac{U_1+U_2+U_3}{3}-\frac{U_1U_2U_3}{3}\\ &\quad+\frac{U_1V_2V_3+U_2V_1V_3+U_3V_1V_2}{3}\\ f_2((U_1,V_1),(U_2,V_2),(U_3,V_3))&=\frac{V_1+V_2+V_3}{3}-\frac{V_1V_2V_3}{3}\\ &\quad+\frac{U_1U_2V_3+U_1U_3V_2+U_2U_3V_1}{3}\\ f_3((U_1,V_1),(U_2,V_2),(U_3,V_3))&=\frac{U_1U_2+U_1U_3+U_2U_3}{3}-\frac{V_1V_2+V_1V_3+V_2V_3}{3}.\end{aligned}$$

19.2 Control Points For Triangular Surfaces

Given an affine frame Δrst in the plane (where $r, s, t \in \mathcal{P}$ are affinely independent points), it turns out that any symmetric multiaffine map $f\colon \mathcal{P}^m \to \mathcal{E}$ is uniquely determined by a family of $\frac{(m+1)(m+2)}{2}$ points (where $\mathcal{E}$ is any affine space, say $\mathbb{R}^n$). Let

$$\Delta_m = \{(i,\, j,\, k) \in \mathbb{N}^3 \mid i + j + k = m\}.$$

The following lemma is easily shown (see Ramshaw [141] or Gallier [70]).

Lemma 19.2.1 *Given a reference triangle Δrst in the affine plane $\mathcal{P}$, given any family $(b_{i,\,j,\,k})_{(i,j,k)\in\Delta_m}$ of $\frac{(m+1)(m+2)}{2}$ points in $\mathcal{E}$, there is a unique surface $F\colon \mathcal{P} \to \mathcal{E}$ of total degree m, defined by a symmetric m-affine polar form $f\colon \mathcal{P}^m \to \mathcal{E}$, such that*

$$f(\underbrace{r,\ldots,r}_{i},\underbrace{s,\ldots,s}_{j},\underbrace{t,\ldots,t}_{k}) = b_{i,\,j,\,k},$$

for all $(i, j, k) \in \Delta_m$. Furthermore, f is given by the expression

$$f(a_1,\ldots,a_m) = \sum_{\substack{I\cup J\cup K=\{1,\ldots,m\}\\ I,J,K \text{ pairwise disjoint}}} \left(\prod_{i\in I}\lambda_i\right)\left(\prod_{j\in J}\mu_j\right)\left(\prod_{k\in K}\nu_k\right) f(\underbrace{r,\ldots,r}_{|I|},\underbrace{s,\ldots,s}_{|J|},\underbrace{t,\ldots,t}_{|K|}),$$

where $a_i = \lambda_i r + \mu_i s + \nu_i t$, with $\lambda_i + \mu_i + \nu_i = 1$, and $1 \le i \le m$. A point $F(a)$ on the surface F can be expressed in terms of the Bernstein polynomials $B^m_{i,j,k}(U,V,T) = \frac{m!}{i!j!k!}\, U^i V^j T^k$, as

$$F(a) = f(\underbrace{a,\ldots,a}_{m}) = \sum_{(i,\,j,\,k)\in\Delta_m} B^m_{i,j,k}(\lambda,\mu,\nu)\; f(\underbrace{r,\ldots,r}_{i},\underbrace{s,\ldots,s}_{j},\underbrace{t,\ldots,t}_{k}),$$

where $a = \lambda r + \mu s + \nu t$, with $\lambda + \mu + \nu = 1$.

For example, with respect to the standard frame $\Delta rst = ((1,0,0),(0,1,0),(0,0,1))$, we obtain the following 10 control points for the Enneper surface:

$$\begin{array}{ccccccc}
 & & & \begin{array}{c} f(r,r,r) \\ \left(\frac{2}{3},0,1\right) \end{array} & & & \\
 & & \begin{array}{c} f(r,r,t) \\ \left(\frac{2}{3},0,\frac{1}{3}\right) \end{array} & & \begin{array}{c} f(r,r,s) \\ \left(\frac{2}{3},\frac{2}{3},\frac{1}{3}\right) \end{array} & & \\
 & \begin{array}{c} f(r,t,t) \\ \left(\frac{1}{3},0,0\right) \end{array} & & \begin{array}{c} f(r,s,t) \\ \left(\frac{1}{3},\frac{1}{3},0\right) \end{array} & & \begin{array}{c} f(r,s,s) \\ \left(\frac{2}{3},\frac{2}{3},-\frac{1}{3}\right) \end{array} & \\
\begin{array}{c} f(t,t,t) \\ (0,0,0) \end{array} & & \begin{array}{c} f(s,t,t) \\ \left(0,\frac{1}{3},0\right) \end{array} & & \begin{array}{c} f(s,s,t) \\ \left(0,\frac{2}{3},-\frac{1}{3}\right) \end{array} & & \begin{array}{c} f(s,s,s) \\ \left(0,\frac{2}{3},-1\right) \end{array}
\end{array}$$

A family $\mathcal{N} = (b_{i,\,j,\,k})_{(i,j,k)\in\Delta_m}$ of $\frac{(m+1)(m+2)}{2}$ points in $\mathcal{E}$ is called a *(triangular) control net, or Bézier net*. Note that the points in

$$\Delta_m = \{(i,\, j,\, k) \in \mathbb{N}^3 \mid i + j + k = m\},$$

can be thought of as a triangular grid of points in $\mathcal{P}$. For example, when $m = 5$, we have the following grid of 21 points:

$$\begin{array}{ccccccccccc}
 & & & & & 500 & & & & & \\
 & & & & 401 & & 410 & & & & \\
 & & & 302 & & 311 & & 320 & & & \\
 & & 203 & & 212 & & 221 & & 230 & & \\
 & 104 & & 113 & & 122 & & 131 & & 140 & \\
005 & & 014 & & 023 & & 032 & & 041 & & 050
\end{array}$$

We intentionally let i be the row index, starting from the left lower corner, and j be the column index, also starting from the left lower corner. The control net $\mathcal{N} = (b_{i,\,j,\,k})_{(i,j,k)\in\Delta_m}$ can be viewed as an image of the triangular grid Δ_m in the affine space $\mathcal{E}$. It follows from lemma 19.2.1 that there is a bijection between polynomial surfaces of degree m and control nets $\mathcal{N} = (b_{i,\,j,\,k})_{(i,j,k)\in\Delta_m}$.

19.3 Control Points For Rectangular Surfaces

Given any two affine frames $(\overline{r}_1, \overline{s}_1)$ and $(\overline{r}_2, \overline{s}_2)$ for the affine line $\mathbb{A}$, it turns out that a $\langle p, q\rangle$-symmetric multiaffine map

$$f\colon (\mathbb{A})^p \times (\mathbb{A})^q \to \mathcal{E}$$

is completely determined by the family of $(p+1)(q+1)$ points in $\mathcal{E}$

$$b_{i,\,j} = f(\underbrace{\overline{r}_1, \ldots, \overline{r}_1}_{p-i}, \underbrace{\overline{s}_1, \ldots, \overline{s}_1}_{i}; \underbrace{\overline{r}_2, \ldots, \overline{r}_2}_{q-j}, \underbrace{\overline{s}_2, \ldots, \overline{s}_2}_{j}),$$

where $0 \le i \le p$ and $0 \le j \le q$. The following lemma is easily shown (see Ramshaw [141] or Gallier [70]).

Lemma 19.3.1 *Let $(\overline{r}_1, \overline{s}_1)$ and $(\overline{r}_2, \overline{s}_2)$ be any two affine frames for the affine line $\mathbb{A}$, and let $\mathcal{E}$ be an affine space (of finite dimension $n \ge 3$). For any natural numbers p, q, for any family $(b_{i,\,j})_{0\le i\le p,\,0\le j\le q}$ of $(p+1)(q+1)$ points in $\mathcal{E}$, there is a unique bipolynomial surface $F\colon \mathbb{A}\times\mathbb{A} \to \mathcal{E}$ of degree $\langle p, q\rangle$, with polar form the $(p+q)$-multiaffine $\langle p, q\rangle$-symmetric map*

$$f\colon (\mathbb{A})^p \times (\mathbb{A})^q \to \mathcal{E},$$

such that

$$f(\underbrace{\overline{r}_1, \ldots, \overline{r}_1}_{p-i}, \underbrace{\overline{s}_1, \ldots, \overline{s}_1}_{i}; \underbrace{\overline{r}_2, \ldots, \overline{r}_2}_{q-j}, \underbrace{\overline{s}_2, \ldots, \overline{s}_2}_{j}) = b_{i,\,j},$$

for all $i, 1 \le i \le p$ and all $j, 1 \le j \le q$. Furthermore, f is given by the expression

$$f(\overline{u}_1, \ldots, \overline{u}_p; \overline{v}_1, \ldots, \overline{v}_q)$$

$$= \sum_{\substack{I\cap J=\emptyset\\ I\cup J=\{1,\ldots,p\}\\ K\cap L=\emptyset\\ K\cup L=\{1,\ldots,q\}}} \prod_{i\in I}\left(\frac{s_1-u_i}{s_1-r_1}\right)\prod_{j\in J}\left(\frac{u_j-r_1}{s_1-r_1}\right)\prod_{k\in K}\left(\frac{s_2-v_k}{s_2-r_2}\right)\prod_{l\in L}\left(\frac{v_l-r_2}{s_2-r_2}\right) b_{|J|,\,|L|}.$$

A point $F(\overline{u}, \overline{v})$ on the surface F can be expressed in terms of the Bernstein polynomials $B_i^p[r_1, s_1](u)$ and $B_j^q[r_2, s_2](v)$, as

$$F(\overline{u}, \overline{v}) = \sum_{\substack{0\le i\le p\\ 0\le j\le q}} B_i^p[r_1, s_1](u)\, B_j^q[r_2, s_2](v)\; f(\underbrace{\overline{r}_1, \ldots, \overline{r}_1}_{p-i}, \underbrace{\overline{s}_1, \ldots, \overline{s}_1}_{i}; \underbrace{\overline{r}_2, \ldots, \overline{r}_2}_{q-j}, \underbrace{\overline{s}_2, \ldots, \overline{s}_2}_{j}).$$

A family $\mathcal{N} = (b_{i,\,j})_{0\leq i\leq p,\, 0\leq j\leq q}$ of $(p+1)(q+1)$ points in $\mathcal{E}$, is often called a *(rectangular) control net, or Bézier net*. Note that we can view the set of pairs

$$\square_{p,q} = \{(i,j) \in \mathbb{N}^2 \mid 0 \leq i \leq p,\, 0 \leq j \leq q\},$$

as a rectangular grid of $(p+1)(q+1)$ points in $\mathbb{A} \times \mathbb{A}$. The control net $\mathcal{N} = (b_{i,\,j})_{(i,j)\in\square_{p,q}}$, can be viewed as an image of the rectangular grid $\square_{p,q}$ in the affine space $\mathcal{E}$. The portion of the surface F corresponding to the points $F(\overline{u}, \overline{v})$ for which the parameters $\overline{u}, \overline{v}$ satisfy the inequalities $r_1 \leq u \leq s_1$ and $r_2 \leq v \leq s_2$, is called a *rectangular (surface) patch, or rectangular Bézier patch*, and $F([\overline{r}_1, \overline{s}_1], [\overline{r}_2, \overline{s}_2])$ is the *trace of the rectangular patch*.

As an example, the *monkey saddle* is the surface defined by the equation

$$z = x^3 - 3xy^2.$$

It is easily shown that the monkey saddle is specified by the following rectangular control net of degree $(3, 2)$ over $[0, 1] \times [0, 1]$:

```
sqmonknet1 =  {{0, 0, 0}, {0, 1/2, 0}, {0, 1, 0}, {1/3, 0, 0},
   {1/3, 1/2, 0}, {1/3, 1, -1}, {2/3, 0, 0}, {2/3, 1/2, 0},
   {2/3, 1, -2}, {1, 0, 1}, {1, 1/2, 1}, {1, 1, -2}}
```

In the next section, we review a beautiful algorithm to compute a point $F(a)$ on a surface patch using affine interpolation steps, the de Casteljau algorithm.

19.4 The de Casteljau Algorithm and Subdivision

In this section, we quickly review how the de Casteljau algorithm can be used to subdivide a triangular patch into three subpatches. For more details, see Farin [58, 57], Hoschek and Lasser [90], Risler [142], or Gallier [70]. There are also versions of the de Casteljau algorithm for rectangular patches, but we will not go into this topic in order to keep the size of this book reasonable. Again, readers are invited to consult Farin [58, 57], Hoschek and Lasser [90], Risler [142], or Gallier [70].

Given an affine frame Δrst, given a triangular control net $\mathcal{N} = (b_{i,\,j,\,k})_{(i,j,k)\in\Delta_m}$, recall that in terms of the polar form $f\colon \mathcal{P}^m \to \mathcal{E}$ of the polynomial surface $F\colon \mathcal{P} \to \mathcal{E}$ defined by $\mathcal{N}$, for every $(i, j, k) \in \Delta_m$, we have

$$b_{i,\,j,\,k} = f(\underbrace{r, \ldots, r}_{i}, \underbrace{s, \ldots, s}_{j}, \underbrace{t, \ldots, t}_{k}).$$

Given $a = \lambda r + \mu s + \nu t$ in $\mathcal{P}$, where $\lambda + \mu + \nu = 1$, in order to compute $F(a) = f(a, \ldots, a)$, the computation builds a sort of tetrahedron consisting of $m + 1$ layers. The base layer consists of the original control points in $\mathcal{N}$, which are also denoted as $(b^0_{i,\,j,\,k})_{(i,j,k)\in\Delta_m}$. The other layers are computed in m stages, where at stage l, $1 \leq l \leq m$, the points $(b^l_{i,\,j,\,k})_{(i,j,k)\in\Delta_{m-l}}$ are computed such that

$$b^l_{i,\,j,\,k} = \lambda b^{l-1}_{i+1,\,j,\,k} + \mu b^{l-1}_{i,\,j+1,\,k} + \nu b^{l-1}_{i,\,j,\,k+1}.$$

During the last stage, the single point $b^m_{0,\,0,\,0}$ is computed. An easy induction shows that

$$b^l_{i,\,j,\,k} = f(\underbrace{a, \ldots, a}_{l}, \underbrace{r, \ldots, r}_{i}, \underbrace{s, \ldots, s}_{j}, \underbrace{t, \ldots, t}_{k}),$$

where $(i, j, k) \in \Delta_{m-l}$, and thus, $F(a) = b^m_{0,\,0,\,0}$.

Assuming that a is not on one of the edges of Δrst, the crux of the subdivision method is that the three other faces of the tetrahedron of polar values $b^l_{i,\,j,\,k}$ besides the face corresponding to the original control net, yield three control nets

$$\mathcal{N}ast = (b^l_{0,\,j,\,k})_{(l,j,k)\in\in\Delta_m},$$

corresponding to the base triangle Δast,

$$\mathcal{N}rat = (b^l_{i,\,0,\,k})_{(i,l,k)\in\in\Delta_m},$$

corresponding to the base triangle Δrat, and

$$\mathcal{N}rsa = (b^l_{i,\,j,\,0})_{(i,j,l)\in\in\Delta_m},$$

corresponding to the base triangle Δrsa. If a belongs to one of the edges, say rs, then the triangle Δrsa is flat, i.e. Δrsa is not an afine frame, and the net $\mathcal{N}rsa$ does not define the surface, but instead a curve. However, in such cases, the degenerate net $\mathcal{N}rsa$ is not needed anyway.

From an implementation point of view, we found it convenient to assume that a triangular net $\mathcal{N} = (b_{i,\,j,\,k})_{(i,j,k)\in\Delta_m}$ is represented as the list consisting of the concatenation of the $m+1$ rows

$$b_{i,\,0,\,m-i},\; b_{i,\,1,\,m-i-1},\; \ldots,\; b_{i,\,m-i,\,0},$$

i.e.,

$$f(\underbrace{r,\ldots,r}_{i},\,\underbrace{t,\ldots,t}_{m-i}),\; f(\underbrace{r,\ldots,r}_{i},\,s,\,\underbrace{t,\ldots,t}_{m-i-1}),\;\ldots,\; f(\underbrace{r,\ldots,r}_{i},\,\underbrace{s,\ldots,s}_{m-i-1},\,t),\; f(\underbrace{r,\ldots,r}_{i},\,\underbrace{s,\ldots,s}_{m-i}),$$

where $0 \le i \le m$. As a triangle, the net $\mathcal{N}$ is listed (from top-down) as

$$\begin{array}{ccccc}
f(\underbrace{t,\ldots,t}_{m}) & f(\underbrace{t,\ldots,t}_{m-1},\,s) & \ldots & f(t,\,\underbrace{s,\ldots,s}_{m-1}) & f(\underbrace{s,\ldots,s}_{m}) \\
 & \ldots & & \ldots & \\
 & & \ldots & & \\
 & f(\underbrace{r,\ldots,r}_{m-1},\,t) & & f(\underbrace{r,\ldots,r}_{m-1},\,s) & \\
 & & f(\underbrace{r,\ldots,r}_{m}) & &
\end{array}$$

The main advantage of this representation is that we can view the net $\mathcal{N}$ as a two-dimensional array net, such that $net[i,j] = b_{i,\,j,\,k}$ (with $i+j+k=m$). In fact, only a triangular portion of this array is filled. This way of representing control nets fits well with the convention that the affine frame Δrst is represented as follows:

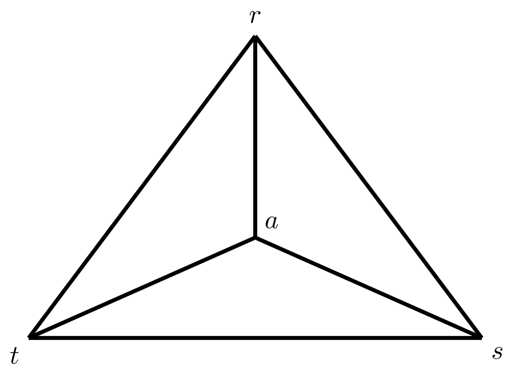

Figure 19.1: An affine frame

Instead of simply computing $F(a) = b^m_{0,0,0}$, the de Casteljau algorithm can be easily adapted to output the three nets $\mathcal{N}ast$, $\mathcal{N}rat$, and $\mathcal{N}rsa$.

Using the above version of the de Casteljau algorithm, it is possible to recursiveley subdivide a triangular patch. It would seem natural to subdivide Δrst into the three subtriangles Δars, Δast, and Δart, where $a = (1/3, 1/3, 1/3)$ is the center of gravity of the triangle Δrst, getting new control nets $\mathcal{N}ars$, $\mathcal{N}ast$ and $\mathcal{N}art$ using the functions described earlier, and repeat this process recursively. However, this process does not yield a good triangulation of the surface patch, because no progress is made on the edges rs, st, and tr, and thus, such a triangulation does not converge to the surface patch. Thus, in order to compute triangulations that converge to the surface patch, we need to subdivide the triangle Δrst in such a way that the edges of the affine frame are subdivided. There are many ways of performing such subdivisions, and we propose a method which has the advantage of yielding a very regular triangulation and of being very efficient.

The subdivision strategy that we propose is to divide the affine frame Δrst into four subtriangles Δabt, Δbac, Δcrb, and Δsca, where $a = (0, 1/2, 1/2)$, $b = (1/2, 0, 1/2)$, and $c = (1/2, 1/2, 0)$, are the middle points of the sides st, rt and rs respectively, as shown in the diagram below:

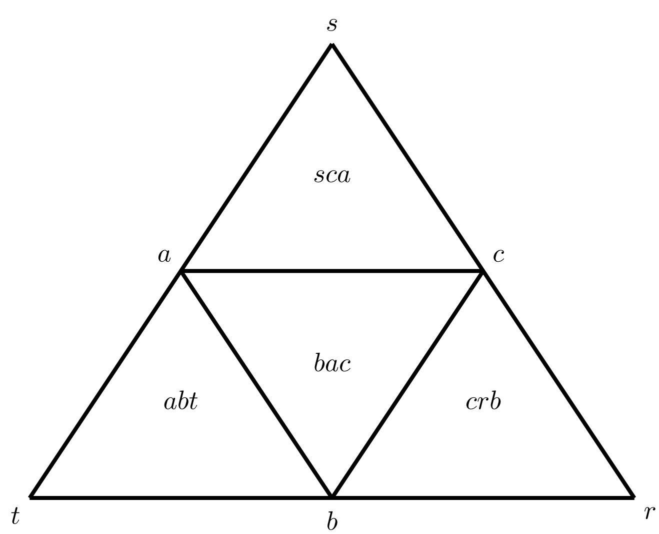

Figure 19.2: Subdividing an affine frame Δrst

It turns out that the four subpatches can be computed in four calls to the subdivison version of the de Casteljau algorithm. Details of such an algorithm can be found in Gallier [70], as well as subdivison algorithms for rectangular surfaces. The monkey saddle shown in figure 19.3 was obtained using a version of the de Casteljau algorithm.

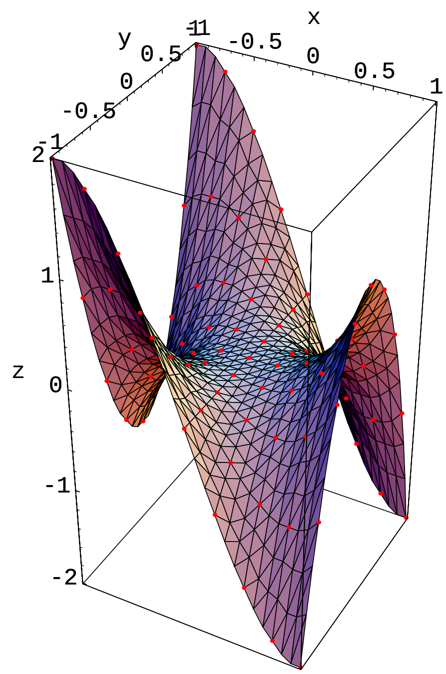

Figure 19.3: A monkey saddle