CHAPTER

SIX

Advanced Geometric Algorithms

6.1 Point Inversion and Projection for Curves and Surfaces

In this chapter we cover various topics which are rather fundamental in implementing NURBS geometry.

Given a point $\mathbf{P} = (x, y, z)$, assumed to lie on the NURBS curve $\mathbf{C}(u)$ of degree p, *point inversion* is the problem of finding the corresponding parameter, $\bar{u}$, such that $\mathbf{C}(\bar{u}) = \mathbf{P}$. Theoretically, point inversion can be solved in closed form if $p \leq 4$. The steps are:

1. using the strong convex hull property (P4.10), determine which spans of $\mathbf{C}(u)$ can possibly contain $\mathbf{P}$;
2. using knot insertion or refinement, extract the candidate spans and convert them to power basis form;
3. each span results in three polynomial equations in one unknown; if the three equations have a common solution, then $\mathbf{P}$ lies on that segment of the curve.

As an example of Step 3, assume $p = 2$. Then a span $\mathbf{r}(u)$ is given by the vector function (in four-dimensional space):

$$\mathbf{r}(u) = \mathbf{a}_0^w + \mathbf{a}_1^w u + \mathbf{a}_2^w u^2 \tag{6.1}$$

Let $\mathbf{a}_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i)$. Projecting Eq. (6.1) to three-dimensional space and setting it equal to $\mathbf{P} = (x, y, z)$, we obtain

$$\frac{w_2 x_2 u^2 + w_1 x_1 u + w_0 x_0}{w_2 u^2 + w_1 u + w_0} = x$$

$$\frac{w_2 y_2 u^2 + w_1 y_1 u + w_0 y_0}{w_2 u^2 + w_1 u + w_0} = y$$

$$\frac{w_2 z_2 u^2 + w_1 z_1 u + w_0 z_0}{w_2 u^2 + w_1 u + w_0} = z$$

which yields

$$\begin{aligned} w_2(x_2 - x)u^2 + w_1(x_1 - x)u + w_0(x_0 - x) &= 0 \\ w_2(y_2 - y)u^2 + w_1(y_1 - y)u + w_0(y_0 - y) &= 0 \\ w_2(z_2 - z)u^2 + w_1(z_1 - z)u + w_0(z_0 - z) &= 0 \end{aligned} \tag{6.2}$$

This method has its disadvantages, for example:

- Equation (6.2) cannot be solved in closed form for $p > 4$, and the solution for $p = 3$ and $p = 4$ is not without problems on a computer;
- there are numerical tolerance problems, e.g., one cannot expect the three solutions of Eq. (6.2) to be exactly equal; but when should they be considered equal? The problem is exacerbated when the point $\mathbf{P}$ is not precisely on the curve (a common occurrence);
- the software implementation of the previous solution is rather involved.

A simpler and completely adequate method is to use Newton iteration to minimize the distance between $\mathbf{P}$ and $\mathbf{C}(u)$ (see Figures 6.1a and 6.1b). $\mathbf{P}$ is considered to be on the curve if the minimum distance is less than a specified tolerance, ϵ_1. This effectively solves the more general problem of *point projection* to a curve. To obtain a start value, u_0, for the Newton iteration:

- if the point is known to lie on the curve (within tolerance), use the strong convex hull property to determine candidate spans; if it is a more general point projection problem, choose all spans as candidates;
- evaluate curve points at n equally spaced parameter values on each candidate span, and compute the distance of each point from $\mathbf{P}$. Choose u_0 to be the value yielding the point closest to $\mathbf{P}$; the number n is generally chosen by some heuristic method. We emphasize that a good start value is important in achieving reliable convergence.

Now assume we have the start value, u_0, and form the dot product function

$$f(u) = \mathbf{C}'(u) \cdot \big(\mathbf{C}(u) - \mathbf{P}\big)$$

The distance from $\mathbf{P}$ to $\mathbf{C}(u)$ is minimum when $f(u) = 0$, whether $\mathbf{P}$ is on the curve or not (Figures 6.1a and 6.1b). Denote by u_i the parameter obtained at the ith Newton iteration. Then

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)} = u_i - \frac{\mathbf{C}'(u_i) \cdot \big(\mathbf{C}(u_i) - \mathbf{P}\big)}{\mathbf{C}''(u_i) \cdot \big(\mathbf{C}(u_i) - \mathbf{P}\big) + |\,\mathbf{C}'(u_i)\,|^2} \tag{6.3}$$

Two zero tolerances can be used to indicate convergence:

ϵ_1 : a measure of Euclidean distance;

ϵ_2 : a zero cosine measure.

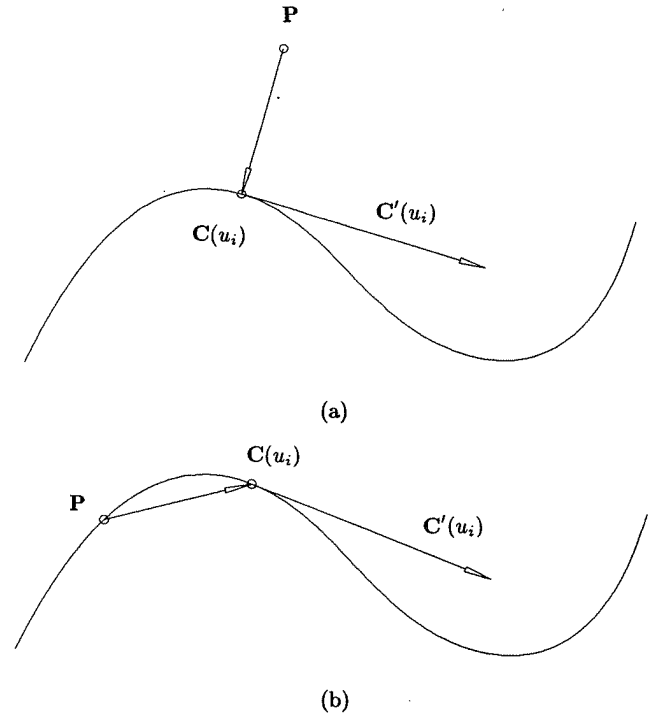


Figure 6.1. (a) Point projection; (b) point inversion.

Convergence criteria are then given by

$$\begin{aligned} &|\,(u_{i+1} - u_i)\,\mathbf{C}'(u_i)\,| \le \epsilon_1 \\ &|\,\mathbf{C}(u_i) - \mathbf{P}\,| \le \epsilon_1 \\ &\frac{|\,\mathbf{C}'(u_i) \cdot \big(\mathbf{C}(u_i) - \mathbf{P}\big)\,|}{|\,\mathbf{C}'(u_i)\,|\,|\,\mathbf{C}(u_i) - \mathbf{P}\,|} \le \epsilon_2 \end{aligned} \tag{6.4}$$

The criteria are checked in the following order:

1. point coincidence:
$$|\,\mathbf{C}(u_i) - \mathbf{P}\,| \le \epsilon_1$$
2. zero cosine:
$$\frac{|\,\mathbf{C}'(u_i) \cdot \big(\mathbf{C}(u_i) - \mathbf{P}\big)\,|}{|\,\mathbf{C}'(u_i)\,|\,|\,\mathbf{C}(u_i) - \mathbf{P}\,|} \le \epsilon_2$$

If at least one of these conditions is not satisfied, a new value, u_{i+1}, is computed using Eq. (6.3). Then two more conditions are checked:

3. ensure that the parameter stays within the range ($u_{i+1} \in [a, b]$)

   if the curve is not closed:

   $$\text{if } (u_{i+1} < a) \qquad u_{i+1} = a$$
   $$\text{if } (u_{i+1} > b) \qquad u_{i+1} = b$$

   if the curve is closed:

   $$\text{if } (u_{i+1} < a) \qquad u_{i+1} = b - (a - u_{i+1})$$
   $$\text{if } (u_{i+1} > b) \qquad u_{i+1} = a + (u_{i+1} - b)$$

4. the parameter does not change significantly, e.g., the point is off the end of the curve

   $$| (u_{i+1} - u_i)\, \mathbf{C}'(u_i) | \leq \epsilon_1$$

If any of conditions (1), (2), or (4) is satisfied, the iteration is halted. Figure 6.2 shows the projection of a set of points onto a NURBS curve.

Point inversion and projection for surfaces are analogous. Form the vector function

$$\mathbf{r}(u, v) = \mathbf{S}(u, v) - \mathbf{P}$$

and the two scalar equations

$$\begin{aligned} f(u, v) &= \mathbf{r}(u, v) \cdot \mathbf{S}_u(u, v) = 0 \\ g(u, v) &= \mathbf{r}(u, v) \cdot \mathbf{S}_v(u, v) = 0 \end{aligned} \tag{6.5}$$

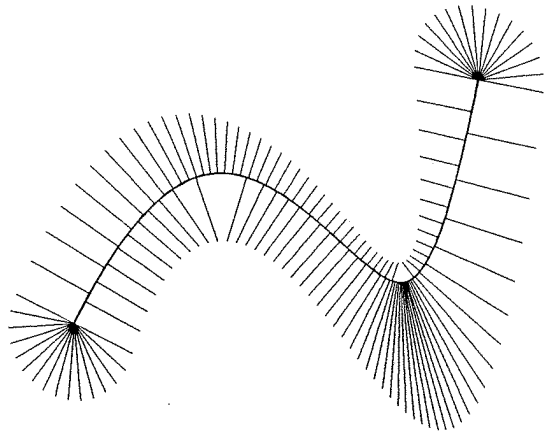


Figure 6.2. Projecting a set of points onto a NURBS curve.

We must solve Eq. (6.5). Let

$$\delta_i = \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} u_{i+1} - u_i \\ v_{i+1} - v_i \end{bmatrix}$$

$$J_i = \begin{bmatrix} f_u & f_v \\ g_u & g_v \end{bmatrix} = \begin{bmatrix} | \mathbf{S}_u |^2 + \mathbf{r} \cdot \mathbf{S}_{uu} & \mathbf{S}_u \cdot \mathbf{S}_v + \mathbf{r} \cdot \mathbf{S}_{uv} \\ \mathbf{S}_u \cdot \mathbf{S}_v + \mathbf{r} \cdot \mathbf{S}_{vu} & | \mathbf{S}_v |^2 + \mathbf{r} \cdot \mathbf{S}_{vv} \end{bmatrix}$$

$$\kappa_i = - \begin{bmatrix} f(u_i, v_i) \\ g(u_i, v_i) \end{bmatrix}$$

where all the functions in the matrix J_i are evaluated at (u_i, v_i). At the ith iteration we must solve the 2×2 system of linear equations in the unknown δ_i, given by

$$J_i \delta_i = \kappa_i \tag{6.6}$$

From δ_i we obtain

$$\begin{aligned} u_{i+1} &= \Delta u + u_i \\ v_{i+1} &= \Delta v + v_i \end{aligned} \tag{6.7}$$

Convergence criteria are given by

$$| (u_{i+1} - u_i)\, \mathbf{S}_u(u_i, v_i) + (v_{i+1} - v_i)\, \mathbf{S}_v(u_i, v_i) | \leq \epsilon_1$$

$$| \mathbf{S}(u_i, v_i) - \mathbf{P} | \leq \epsilon_1$$

$$\frac{| \mathbf{S}_u(u_i, v_i) \cdot (\mathbf{S}(u_i, v_i) - \mathbf{P}) |}{| \mathbf{S}_u(u_i, v_i) |\, | \mathbf{S}(u_i, v_i) - \mathbf{P} |} \leq \epsilon_2 \qquad \frac{| \mathbf{S}_v(u_i, v_i) \cdot (\mathbf{S}(u_i, v_i) - \mathbf{P}) |}{| \mathbf{S}_v(u_i, v_i) |\, | \mathbf{S}(u_i, v_i) - \mathbf{P} |} \leq \epsilon_2 \tag{6.8}$$

Again, the conditions are checked by

1. point coincidence:

   $$| \mathbf{S}(u_i, v_i) - \mathbf{P} | \leq \epsilon_1$$

2. zero cosine:

   $$\frac{| \mathbf{S}_u(u_i, v_i) \cdot (\mathbf{S}(u_i, v_i) - \mathbf{P}) |}{| \mathbf{S}_u(u_i, v_i) |\, | \mathbf{S}(u_i, v_i) - \mathbf{P} |} \leq \epsilon_2 \qquad \frac{| \mathbf{S}_v(u_i, v_i) \cdot (\mathbf{S}(u_i, v_i) - \mathbf{P}) |}{| \mathbf{S}_v(u_i, v_i) |\, | \mathbf{S}(u_i, v_i) - \mathbf{P} |} \leq \epsilon_2$$

If these conditions are not satisfied, a new value (u_{i+1}, v_{i+1}) is computed using Eq. (6.7). Then two more conditions are checked:

3. ensure that the parameters stay in range ($u_{i+1} \in [a, b]$ and $v_{i+1} \in [c, d]$):

   if the surface is not closed in the u direction:

   $$\text{if } (u_{i+1} < a) \qquad u_{i+1} = a$$
   $$\text{if } (u_{i+1} > b) \qquad u_{i+1} = b$$

if the surface is not closed in the v direction:

$$\text{if } (v_{i+1} < c) \quad v_{i+1} = c$$
$$\text{if } (v_{i+1} > d) \quad v_{i+1} = d$$

if the surface is closed in the u direction:

$$\text{if } (u_{i+1} < a) \quad u_{i+1} = b - (a - u_{i+1})$$
$$\text{if } (u_{i+1} > b) \quad u_{i+1} = a + (u_{i+1} - b)$$

if the surface is closed in the v direction:

$$\text{if } (v_{i+1} < c) \quad v_{i+1} = d - (c - v_{i+1})$$
$$\text{if } (v_{i+1} > d) \quad v_{i+1} = c + (v_{i+1} - d)$$

4. parameters do not change significantly, that is

$$| (u_{i+1} - u_i)\,\mathbf{S}_u(u_i, v_i) + (v_{i+1} - v_i)\,\mathbf{S}_v(u_i, v_i) | \leq \epsilon_1$$

Iteration is halted if any of conditions (1), (2), or (4) is satisfied. Figure 6.3 shows the projection of a set of points from the control net onto a NURBS surface.

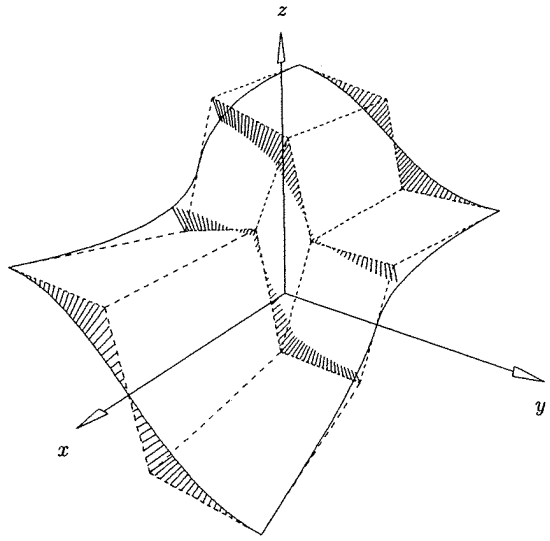


Figure 6.3. Projecting points of the control net onto a NURBS surface.

6.2 Surface Tangent Vector Inversion

Let $(\bar{u}, \bar{v})$ be a fixed parameter point and $\mathbf{P} = \mathbf{S}(\bar{u}, \bar{v})$ its image on the surface $\mathbf{S}(u, v)$ (see Figure 6.4). Let $\mathbf{T} = (dx, dy, dz)$ be a vector starting at $\mathbf{P}$ and lying in the tangent plane of $\mathbf{S}(u, v)$ at $\mathbf{P}$. Denote by $\mathbf{S}_u$ and $\mathbf{S}_v$ the first partial derivatives of $\mathbf{S}(u, v)$ at $(\bar{u}, \bar{v})$. If $\mathbf{S}_u \times \mathbf{S}_v \neq 0$, it follows from elementary differential geometry [DoCa76] that there exists a vector $\mathbf{W} = (du, dv)$ in the uv plane such that

$$\mathbf{T} = \mathbf{S}_u du + \mathbf{S}_v dv \tag{6.9}$$

Tangent vector inversion is the process of determining the vector $\mathbf{W}$. Equation (6.9) expands into three equations in two unknowns, that is

$$\begin{bmatrix} x_u & x_v \\ y_u & y_v \\ z_u & z_v \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \tag{6.10}$$

which we write as

$$M\mathbf{W} = \mathbf{T}$$

Equation (6.10) has a unique and exact solution (generally such a system does not) which we obtain by multiplying through by M^T (the transpose of M)

$$(M^T M)\,\mathbf{W} = M^T \mathbf{T}$$

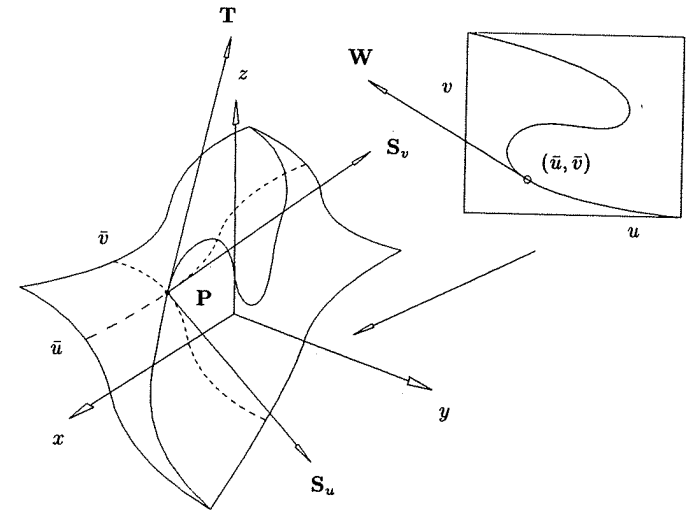


Figure 6.4. Surface tangent vector inversion.

or
$$\begin{bmatrix} |\mathbf{S}_u|^2 & \mathbf{S}_u \cdot \mathbf{S}_v \\ \mathbf{S}_u \cdot \mathbf{S}_v & |\mathbf{S}_v|^2 \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = \begin{bmatrix} \mathbf{S}_u \cdot \mathbf{T} \\ \mathbf{S}_v \cdot \mathbf{T} \end{bmatrix} \tag{6.11}$$

Solving this 2×2 system of equations yields the unknowns du and dv.

6.3 Transformations and Projections of Curves and Surfaces

Transformations and projections of curves and surfaces are common in geometric modeling and computer graphics. Transformations include translations, rotations, scalings, shears, and reflections. Parallel and perspective projections are essential in viewing three-dimensional geometry on a video screen or other two-dimensional output devices. There are many excellent discussions of transformations and projections in the literature, e.g., [Faux81; Fole90; Roge90]. In this section we assume the reader is familiar with the general concepts, and we elaborate only on the application of these concepts to NURBS curves and surfaces.

All transformations and projections are performed on NURBS curves and surfaces by applying operations to the control points and weights; hence, it suffices to restrict the discussion to curves. Equations (4.1) and (4.5) represent two different ways to think of a NURBS curve, namely, as a piecewise rational curve in Euclidean space, or as a nonrational B-spline curve in homogeneous space. The first method uses the B-spline basis functions to blend weighted three-dimensional control points; the second method uses the functions to blend four-dimensional control points. Correspondingly, there exist two methods for performing transformations and projections:

- apply the operations to the three-dimensional control points of $\mathbf{C}(u)$; for a perspective projection, new weights must also be computed;
- apply a 4×4 matrix to the four-dimensional control points of $\mathbf{C}^w(u)$.

Let
$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} = \sum_{i=0}^{n} R_{i,p}(u) \mathbf{P}_i \tag{6.12}$$

With the exception of perspective, all transformations and projections of $\mathbf{C}(u)$ are performed by applying the operation to the three-dimensional control points, $\mathbf{P}_i$; the weights, w_i, do not change. This follows from the Affine Invariance Property, P4.9 (which follows from the form of Eq. [6.12]). Let us consider projections. Figure 6.5 shows a general parallel projection of the control point, $\mathbf{P}_i$, to a projection plane given by the reference point, $\mathbf{Q}$, and the unit length normal vector, $\mathbf{N}$. $\bar{\mathbf{P}}_i$ denotes the projection of $\mathbf{P}_i$. If the direction of the projection is given by the vector $\mathbf{W}$, then

$$\bar{\mathbf{P}}_i = \mathbf{P}_i + \alpha \mathbf{W} \tag{6.13}$$

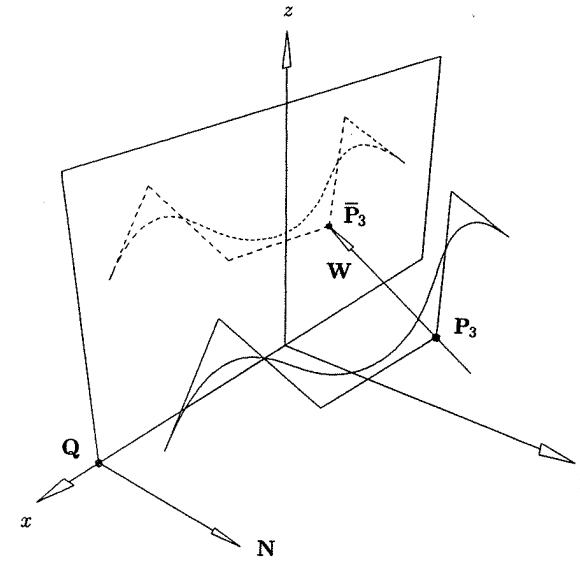


Figure 6.5. Parallel projection of NURBS curves.

From Eq. (6.13) we have
$$\alpha \mathbf{W} = \bar{\mathbf{P}}_i - \mathbf{P}_i - \mathbf{Q} + \mathbf{Q}$$
and
$$\alpha(\mathbf{N} \cdot \mathbf{W}) = \mathbf{N} \cdot (\bar{\mathbf{P}}_i - \mathbf{Q}) + \mathbf{N} \cdot (\mathbf{Q} - \mathbf{P}_i)$$

Since $\bar{\mathbf{P}}_i$ lies on the projection plane, $\mathbf{N} \cdot (\bar{\mathbf{P}}_i - \mathbf{Q}) = 0$, from which follows

$$\alpha = \frac{\mathbf{N} \cdot (\mathbf{Q} - \mathbf{P}_i)}{\mathbf{N} \cdot \mathbf{W}}$$

and finally
$$\bar{\mathbf{P}}_i = \mathbf{P}_i + \left(\frac{\mathbf{N} \cdot (\mathbf{Q} - \mathbf{P}_i)}{\mathbf{N} \cdot \mathbf{W}} \right) \mathbf{W} \tag{6.14}$$

The corresponding formula for surfaces is
$$\bar{\mathbf{P}}_{i,j} = \mathbf{P}_{i,j} + \left(\frac{\mathbf{N} \cdot (\mathbf{Q} - \mathbf{P}_{i,j})}{\mathbf{N} \cdot \mathbf{W}} \right) \mathbf{W} \tag{6.15}$$

Figure 6.6 shows a perspective projection. As before, let the projection plane be defined by $\mathbf{Q}$ and $\mathbf{N}$. Denote the eye position by $\mathbf{E}$. Then

$$\bar{\mathbf{P}}_i = (1 - \alpha) \mathbf{P}_i + \alpha \mathbf{E} \tag{6.16}$$

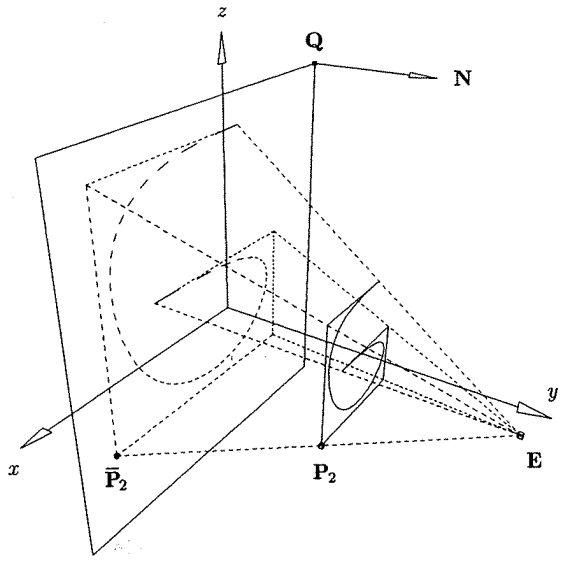


Figure 6.6. Perspective projection of NURBS curves.

It follows that

$$\alpha(\mathbf{E}-\mathbf{P}_i)=\bar{\mathbf{P}}_i-\mathbf{P}_i-\mathbf{Q}+\mathbf{Q}$$

and $\mathbf{N}\cdot(\bar{\mathbf{P}}_i-\mathbf{Q})=0$ implies that

$$\alpha=\frac{\mathbf{N}\cdot(\mathbf{Q}-\mathbf{P}_i)}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}$$

From

$$1-\alpha=1-\frac{\mathbf{N}\cdot(\mathbf{Q}-\mathbf{P}_i)}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}=\frac{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i-\mathbf{Q}+\mathbf{P}_i)}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}=\frac{\mathbf{N}\cdot(\mathbf{E}-\mathbf{Q})}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}$$

we obtain

$$\bar{\mathbf{P}}_i=\frac{\mathbf{N}\cdot(\mathbf{E}-\mathbf{Q})}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}\mathbf{P}_i+\frac{\mathbf{N}\cdot(\mathbf{Q}-\mathbf{P}_i)}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}\mathbf{E} \tag{6.17}$$

Equation (6.17) is the formula for computing the new control points of the projected curve.

We now derive the formula for the new weights. Let

$$\mathbf{P}=\mathbf{C}(u)=\sum_{i=0}^{n}R_{i,p}(u)\mathbf{P}_i$$

be an arbitrary point on $\mathbf{C}(u)$, and denote by $\bar{\mathbf{P}}$ the perspective projection of $\mathbf{P}$

onto the plane given by $\mathbf{Q}$ and $\mathbf{N}$. Recalling that

$$\sum_{i=0}^{n}R_{i,p}(u)=1$$

for all u, we obtain

$$\begin{aligned}\bar{\mathbf{P}}&=\frac{\mathbf{N}\cdot(\mathbf{E}-\mathbf{Q})}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P})}\mathbf{P}+\frac{\mathbf{N}\cdot(\mathbf{Q}-\mathbf{P})}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P})}\mathbf{E}\\&=\frac{\sum R_{i,p}\big(\mathbf{N}\cdot(\mathbf{E}-\mathbf{Q})\mathbf{P}_i\big)+\sum R_{i,p}\big(\mathbf{N}\cdot(\mathbf{Q}-\mathbf{P}_i)\big)\mathbf{E}}{\sum R_{i,p}\big(\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)\big)}\\&=\frac{\sum R_{i,p}\big(\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)\big)\left(\dfrac{\mathbf{N}\cdot(\mathbf{E}-\mathbf{Q})}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}\right)\mathbf{P}_i}{\sum R_{i,p}\big(\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)\big)}\\&\quad+\frac{\sum R_{i,p}\big(\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)\big)\left(\dfrac{\mathbf{N}\cdot(\mathbf{Q}-\mathbf{P}_i)}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}\right)\mathbf{E}}{\sum R_{i,p}\big(\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)\big)}\end{aligned}$$

Setting

$$\bar{w}_i=w_i\big(\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)\big) \tag{6.18}$$

and recalling that

$$R_{i,p}=\frac{N_{i,p}w_i}{\sum N_{i,p}w_i}$$

we obtain

$$\begin{aligned}\bar{\mathbf{P}}&=\frac{\sum N_{i,p}\bar{w}_i\left(\dfrac{\mathbf{N}\cdot(\mathbf{E}-\mathbf{Q})}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}\mathbf{P}_i+\dfrac{\mathbf{N}\cdot(\mathbf{Q}-\mathbf{P}_i)}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_i)}\mathbf{E}\right)}{\sum N_{i,p}\bar{w}_i}\\&=\frac{\sum N_{i,p}(u)\bar{w}_i\bar{\mathbf{P}}_i}{\sum N_{i,p}(u)\bar{w}_i}\end{aligned} \tag{6.19}$$

Equation (6.19) shows that the control points and weights given by Eqs. (6.17) and (6.18) are those of the projected curve. The formulas for surfaces are

$$\bar{\mathbf{P}}_{i,j}=\frac{\mathbf{N}\cdot(\mathbf{E}-\mathbf{Q})}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_{i,j})}\mathbf{P}_{i,j}+\frac{\mathbf{N}\cdot(\mathbf{Q}-\mathbf{P}_{i,j})}{\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_{i,j})}\mathbf{E} \tag{6.20}$$

and

$$\bar{w}_{i,j}=w_{i,j}[\mathbf{N}\cdot(\mathbf{E}-\mathbf{P}_{i,j})] \tag{6.21}$$

Figure 6.7 shows parallel projection of NURBS surfaces.

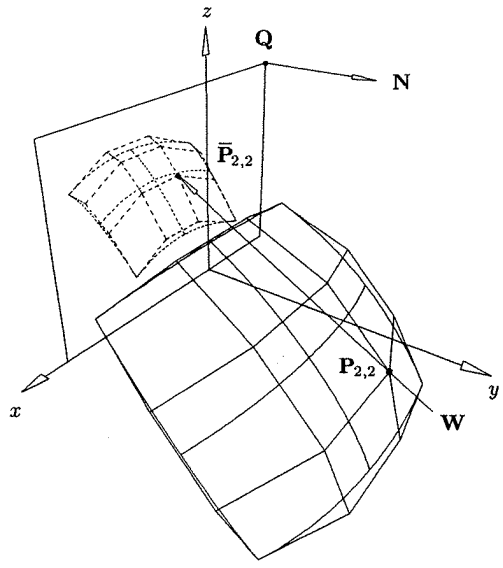


Figure 6.7. Parallel projection of NURBS surfaces.

Now let $\mathbf{C}^w(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i^w$. All transformations and projections can be packed into one 4×4 matrix of the form

$$A = \begin{bmatrix} B & T \\ P & a_{3,3} \end{bmatrix} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & t_0 \\ b_{1,0} & b_{1,1} & b_{1,2} & t_1 \\ b_{2,0} & b_{2,1} & b_{2,2} & t_2 \\ p_0 & p_1 & p_2 & a_{3,3} \end{bmatrix} \tag{6.22}$$

where the vectors T and P represent translation and perspective, respectively, and the 3×3 matrix, B, contains rotation, scaling, shear, reflection through a coordinate plane, and projection to a coordinate plane. The matrix A is to be applied to all control points $\mathbf{P}_i^w$ of $\mathbf{C}^w(u)$.

Examples

Let $\mathbf{P}_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i)$, and denote a transformed homogeneous control point by $\bar{\mathbf{P}}_i^w = A\mathbf{P}_i^w$.

Ex6.1 Translate $\mathbf{C}^w(u)$ by the vector $\mathbf{V} = (x, y, z)$. Then

$$\bar{\mathbf{P}}_i^w = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} = \begin{bmatrix} w_i(x_i + x) \\ w_i(y_i + y) \\ w_i(z_i + z) \\ w_i \end{bmatrix}$$

Ex6.2 Apply a rotation matrix R to $\mathbf{C}^w(u)$

$$\bar{\mathbf{P}}_i^w = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} & 0 \\ r_{1,0} & r_{1,1} & r_{1,2} & 0 \\ r_{2,0} & r_{2,1} & r_{2,2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} = \begin{bmatrix} w_i(r_{0,0}x_i + r_{0,1}y_i + r_{0,2}z_i) \\ w_i(r_{1,0}x_i + r_{1,1}y_i + r_{1,2}z_i) \\ w_i(r_{2,0}x_i + r_{2,1}y_i + r_{2,2}z_i) \\ w_i \end{bmatrix}$$

Ex6.3 Perspective projection: Let $\mathbf{E} = (0, 0, d)$ be the eye position lying on the z-axis, a distance d from the origin. Let the xy plane be the projection plane. Then

$$\bar{\mathbf{P}}_i^w = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & d \end{bmatrix} \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} = \begin{bmatrix} w_i d x_i \\ w_i d y_i \\ 0 \\ w_i(d - z_i) \end{bmatrix} \tag{6.23}$$

6.4 Reparameterization of NURBS Curves and Surfaces

Let $\mathbf{C}(u) = \big(x(u), y(u), z(u)\big)$ be an arbitrary parametric curve on $u \in [a, b]$, and assume that $u = f(s)$ is a scalar-valued function on $s \in [c, d]$ satisfying:

- $f'(s) > 0$ for all $s \in [c, d]$ ($f(s)$ is strictly increasing);
- $a = f(c)$ and $b = f(d)$ ($f(s)$ maps $[c, d]$ onto $[a, b]$).

The composition of $\mathbf{C}(u)$ and $f(s)$, given by

$$\mathbf{C}(s) = \mathbf{C}\big(f(s)\big) = \Big(x\big(f(s)\big), y\big(f(s)\big), z\big(f(s)\big)\Big)$$

is called a *reparameterization* of $\mathbf{C}(u)$ (see Figure 6.8).

$\mathbf{C}(s)$ is geometrically the same curve as $\mathbf{C}(u)$, but parametrically they are different. Applications of reparameterization include:

- Internal point mapping: Given parameter values $u_0, \ldots, u_n$ and $s_0, \ldots, s_n$, where $a < u_0 < \cdots < u_n < b$ and $c < s_0 < \cdots < s_n < d$, determine a function $u = f(s)$ such that $u_i = f(s_i)$, $i = 0, \ldots, n$. Then form $\mathbf{C}\big(f(s)\big)$. This forces the curve points $\mathbf{C}(u_i)$ to be assumed at the new parameter values, s_i;
- Modification of end derivatives: Reparameterization changes the curve derivatives; this follows from the Chain Rule, e.g.

$$\mathbf{C}'(s) = \mathbf{C}'(u)f'(s) \qquad \mathbf{C}''(s) = \mathbf{C}'(u)f''(s) + \mathbf{C}''(u)\big(f'(s)\big)^2$$

  Note that only the magnitudes of the end first derivatives change, but magnitudes and directions of the end second and higher derivatives change;
- Modification of end weights: If $\mathbf{C}^w(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i^w$ is a NURBS

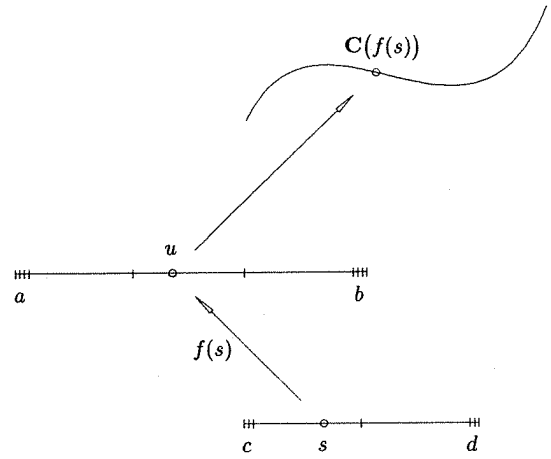


Figure 6.8. Curve reparameterization.

curve, it is sometimes useful to modify the end weights, w_0 and w_n, to have specific values without changing the curve geometry. We shall see that this is possible and in fact is simply equivalent to a reparameterization of $\mathbf{C}(u)$.

Examples

Ex6.4 Let

$$\mathbf{C}(u) = (x(u), y(u)) = (u, -2u^2 + 2u) \tag{6.24}$$

on $u \in [0,1]$, and assume we want a parameter $s \in [0,1]$ such that $u = 1/2$ corresponds to $s = 6/10$, that is

$$\mathbf{C}\left(u = \frac{1}{2}\right) = \mathbf{C}\left(s = \frac{6}{10}\right) = \left(\frac{1}{2}, \frac{1}{2}\right)$$

Then $u = f(s)$ must satisfy three constraints

$$0 = f(0) \qquad \frac{1}{2} = f\left(\frac{6}{10}\right) \qquad 1 = f(1)$$

We choose $f(s)$ to be a quadratic polynomial

$$u = f(s) = as^2 + bs + c$$

The first constraint implies $c = 0$, and the last two yield the linear equations

$$\begin{aligned} \frac{1}{2} &= \frac{9}{25}a + \frac{6}{10}b \\ 1 &= a + b \end{aligned} \tag{6.25}$$

Solving Eq. (6.25) yields

$$a = \frac{5}{12} \qquad b = \frac{7}{12}$$

thus

$$u = f(s) = \frac{5}{12}s^2 + \frac{7}{12}s \tag{6.26}$$

Substituting Eq. (6.26) into Eq. (6.24), we obtain the reparameterized curve

$$\mathbf{C}(s) = \left(\frac{5}{12}s^2 + \frac{7}{12}s, -\frac{25}{72}s^4 - \frac{35}{36}s^3 + \frac{11}{72}s^2 + \frac{7}{6}s\right) \tag{6.27}$$

The reader should verify that

$$\begin{gathered} \mathbf{C}(s=0) = \mathbf{C}(u=0) = (0,0) \\ \mathbf{C}\left(s = \frac{6}{10}\right) = \mathbf{C}\left(u = \frac{1}{2}\right) = \left(\frac{1}{2}, \frac{1}{2}\right) \\ \mathbf{C}(s=1) = \mathbf{C}(u=1) = (0,0) \end{gathered}$$

Note also that the derivatives have changed. For example, differentiating Eqs. (6.24) and (6.27) directly yields

$$\mathbf{C}'(u=0) = (1,2) \qquad \mathbf{C}'(s=0) = \left(\frac{7}{12}, \frac{7}{6}\right)$$

Figure 6.9 shows the curve and its first derivatives with respect to u and s at $u = s = 0$.

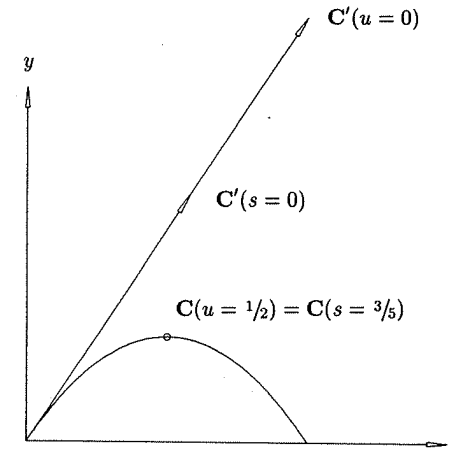


Figure 6.9. The reparameterized curve of Example Ex6.4.

Ex6.5 Consider the circular arc given by

$$\mathbf{C}(u) = \left(\frac{1-u^2}{1+u^2}, \frac{2u}{1+u^2}\right) \tag{6.28}$$

on $u \in [0,1]$. This arc is centered at the origin, has radius 1, and sweeps an angle of 90° (see Chapter 1, Eq. [1.2], Figure 1.2, Example 1.8, and Figure 1.19a). It is easy to check that

$$\mathbf{C}\left(u = \frac{1}{2}\right) = \left(\frac{6}{10}, \frac{8}{10}\right)$$

and

$$\begin{aligned} \mathbf{C}'(u=0) &= (0,2) \\ \mathbf{C}'(u=1) &= (-1,0) \end{aligned} \tag{6.29}$$

We want a reparameterization, $\mathbf{C}(s)$, of $\mathbf{C}(u)$ which is more symmetric, namely, $\mathbf{C}(s)$ should satisfy

$$|\,\mathbf{C}'(s=0)\,| = |\,\mathbf{C}'(s=1)\,| \tag{6.30}$$

We choose a linear rational function of the form

$$u = f(s) = \frac{as+b}{cs+d} \qquad s \in [0,1] \tag{6.31}$$

Clearly, we can assume that $d = 1$, and the conditions $f(0) = 0$ and $f(1) = 1$ imply $b = 0$ and $c = a - 1$. Hence, $f(s)$ has the form

$$u = \frac{as}{(a-1)s+1} \tag{6.32}$$

The condition given by Eq. (6.30) determines a. From Eq. (6.29) and $\mathbf{C}'(s) = \mathbf{C}'(u)\,f'(s)$, we require

$$2f'(s=0) = f'(s=1) \tag{6.33}$$

Differentiating Eq. (6.32) yields

$$f'(s) = \frac{a}{\big((a-1)s+1\big)^2}$$

and from Eq. (6.33) we obtain $2a = 1/a$. It follows that $2a^2 - 1 = 0$, thus $a = \pm\sqrt{2}/2$. Choosing $a = -\sqrt{2}/2$ causes a zero in the denominator of Eq. (6.32) in the interval $s \in [0,1]$, hence we select $a = \sqrt{2}/2$. Thus, we obtain

$$u = f(s) = \frac{\sqrt{2}s}{(\sqrt{2}-2)s+2} \tag{6.34}$$

Substituting Eq. (6.34) into Eq. (6.28) yields

$$\mathbf{C}(s) = \left(\frac{(1-\sqrt{2})s^2 + (\sqrt{2}-2)s + 1}{(2-\sqrt{2})s^2 + (\sqrt{2}-2)s + 1}, \frac{(1-\sqrt{2})s^2 + \sqrt{2}s}{(2-\sqrt{2})s^2 + (\sqrt{2}-2)s + 1}\right) \tag{6.35}$$

The reader can verify that $\mathbf{C}(s)$ satisfies Eq. (6.30) and also the condition

$$\mathbf{C}\left(s = \frac{1}{2}\right) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right) \tag{6.36}$$

that is, $s = 1/2$ maps to the midpoint of the circular arc.

Note that:

- if $\mathbf{C}(u)$ is a pth-degree polynomial (rational) curve and $f(s)$ is a qth-degree polynomial (rational) function, then $\mathbf{C}(s)$ is a pqth-degree polynomial (rational) curve. In particular, a linear reparameterization function of the form $u = as + b$ or $u = (as+b)/(cs+d)$ does not raise the degree of the curve;
- a linear rational function of the form $u = (as+b)/(cs+d)$ converts a polynomial curve to a rational curve of the same degree.

We now discuss the problem of reparameterizing a NURBS curve with a polynomial function. Let $\mathbf{C}^w(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i^w$ be a pth-degree NURBS curve on $u \in [a,b]$, and $u = f(s)$ a qth-degree polynomial on $s \in [c,d]$. For now let us assume $\mathbf{C}^w(u)$ has no internal knots, therefore U has the form

$$U = \{\underbrace{a,\ldots,a}_{p+1}, \underbrace{b,\ldots,b}_{p+1}\}$$

The reparameterized curve $\mathbf{C}^w(s)$ has degree pq and is defined on the knot vector

$$S = \{\underbrace{c,\ldots,c}_{pq+1}, \underbrace{d,\ldots,d}_{pq+1}\} \tag{6.37}$$

The $pq+1$ new control points can be computed by repeated application of the Chain Rule and the formulas for the end derivatives of a B-spline curve. We illustrate this by example.

Example

Ex6.6 Let $\mathbf{C}(u) = \sum_{i=0}^{2} N_{i,2}(u)\mathbf{P}_i$, $U = \{0,0,0,1,1,1\}$, be the B-spline representation of the parabolic arc of Example Ex6.4 (see Figures 6.9 and

6.10). The control points are $\{(0,0),(1/2,1),(1,0)\}$, and the reparameterization function is

$$u = f(s) = \frac{5}{12}s^2 + \frac{7}{12}s \quad \text{on} \quad s \in [0,1]$$

The reparameterized curve, of degree four, is given by

$$\mathbf{C}(s) = \sum_{i=0}^{4} N_{i,4}(s)\mathbf{Q}_i \qquad S = \{0,0,0,0,0,1,1,1,1,1\} \tag{6.38}$$

Clearly (see Figure 6.10)

$$\mathbf{Q}_0 = \mathbf{P}_0 \qquad \mathbf{Q}_4 = \mathbf{P}_2$$

From $\mathbf{C}'(s) = \mathbf{C}'(u)f'(s)$ and Eq. (3.7) we obtain

$$\begin{aligned}\mathbf{C}'(s=0) &= \frac{4}{1-0}(\mathbf{Q}_1 - \mathbf{Q}_0) = \mathbf{C}'(u=0)f'(s=0) \\ &= \frac{2}{1-0}(\mathbf{P}_1 - \mathbf{P}_0)\left(\frac{7}{12}\right)\end{aligned}$$

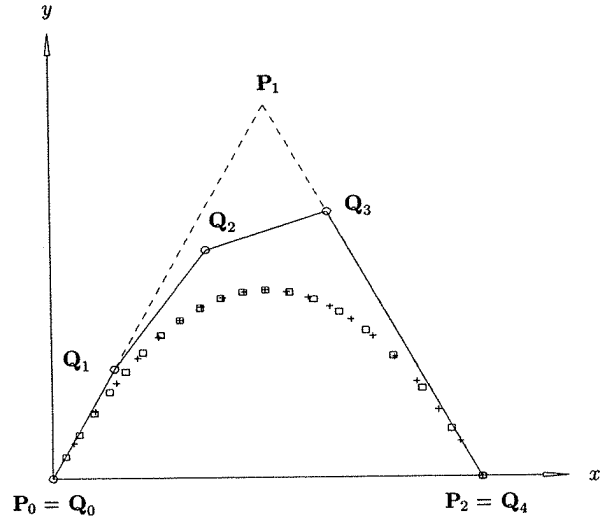


Figure 6.10. The reparameterized B-spline curve of Example Ex6.6; + shows the original parameterization and □ illustrates the new parameterization.

from which follows

$$\mathbf{Q}_1 = \frac{7}{24}(\mathbf{P}_1 - \mathbf{P}_0) + \mathbf{Q}_0 = \left(\frac{7}{48}, \frac{7}{24}\right)$$

Similarly, $4(\mathbf{Q}_4 - \mathbf{Q}_3) = 2(\mathbf{P}_2 - \mathbf{P}_1)(17/12)$, and

$$\mathbf{Q}_3 = \mathbf{Q}_4 - \frac{17}{24}(\mathbf{P}_2 - \mathbf{P}_1) = \left(\frac{31}{48}, \frac{17}{24}\right)$$

From $\mathbf{C}''(s) = \mathbf{C}'(u)f''(s) + \mathbf{C}''(u)\big(f'(s)\big)^2$ and Eq. (3.9), and after noting that

$$\mathbf{C}'(u=0) = (1,2) \qquad \mathbf{C}''(u=0) = (0,-4)$$
$$f'(s=0) = \frac{7}{12} \qquad f''(s=0) = \frac{5}{6}$$

we obtain

$$\mathbf{C}''(s=0) = 12(\mathbf{Q}_0 - 2\mathbf{Q}_1 + \mathbf{Q}_2) = \frac{5}{6}(1,2) + \left(\frac{7}{12}\right)^2(0,-4)$$

from which follows

$$\mathbf{Q}_2 = \frac{5}{72}(1,2) + \frac{49}{1728}(0,-4) + 2\left(\frac{7}{48}, \frac{7}{24}\right) = \left(\frac{156}{432}, \frac{263}{432}\right)$$

Hence, the general procedure for reparameterizing a pth-degree curve $\mathbf{C}^w(u)$ having no internal knots, with a qth-degree polynomial, $f(s)$, is to:

1. set $\mathbf{Q}_0 = \mathbf{P}_0$ and $\mathbf{Q}_{pq} = \mathbf{P}_p$;
2. let $\mathbf{C}^{w(i)}$ denote the ith derivative of $\mathbf{C}^w(u)$, and let $m_\ell = {}^{(pq+1)}/_2$ (integer arithmetic), and $m_r = pq - {}^{pq}/_2 - 1$. Compute the derivatives $\mathbf{C}^{w(i)}(u=a)$ for $i = 1,\ldots,m_\ell$, and $\mathbf{C}^{w(i)}(u=b)$ for $i = 1,\ldots,m_r$; note that the differentiation is carried out in homogeneous space;
3. let $f^{(i)}$ denote the ith derivative of $f(s)$; compute the derivatives $f^{(i)}(s=c)$ for $i = 1,\ldots,m_\ell$, and $f^{(i)}(s=d)$ for $i = 1,\ldots,m_r$;
4. using repeated application of the Chain Rule (which expresses $\mathbf{C}^{w(i)}(s)$ in terms of $\mathbf{C}^{w(i)}(u)$ and $f^{(i)}(s)$) and Eqs. (6.41) and (6.42), compute the new control points $\mathbf{Q}_1,\ldots,\mathbf{Q}_{m_\ell}$ and $\mathbf{Q}_{pq-1},\ldots,\mathbf{Q}_{pq-m_r}$.

Applying the Chain Rule four times yields

$$\mathbf{C}^{w(1)}(s) = \mathbf{C}^{w(1)}(u)f^{(1)}(s)$$

$$\mathbf{C}^{w(2)}(s) = \mathbf{C}^{w(1)}(u)f^{(2)}(s) + \mathbf{C}^{w(2)}(u)\big(f^{(1)}(s)\big)^2$$

$$\mathbf{C}^{w(3)}(s) = \mathbf{C}^{w(1)}(u)f^{(3)}(s) + 3\mathbf{C}^{w(2)}(u)f^{(1)}(s)f^{(2)}(s) + \mathbf{C}^{w(3)}(u)\big(f^{(1)}(s)\big)^3$$

$$\mathbf{C}^{w(4)}(s) = \mathbf{C}^{w(1)}(u)f^{(4)}(s) + \mathbf{C}^{w(2)}(u)\left[4f^{(1)}(s)f^{(3)}(s) + 3\left(f^{(2)}(s)\right)^2\right]$$
$$+ 6\mathbf{C}^{w(3)}(u)\left(f^{(1)}(s)\right)^2 f^{(2)}(s) + \mathbf{C}^{w(4)}(u)\left(f^{(1)}(s)\right)^4 \qquad (6.39)$$

The general formula

$$\mathbf{C}^{w(n)}(s) = \sum_{j=0}^{n} \sum_{\substack{k_1+k_2+\cdots+k_n=j \\ k_1+2k_2+\cdots+nk_n=n \\ k_1,k_2,\ldots,k_n\geq 0}} \mathbf{C}^{w(j)}(u) \frac{n!}{k_1!(1!)^{k_1}\cdots k_n!(n!)^{k_n}} \left(f^{(1)}(s)\right)^{k_1}\cdots\left(f^{(n)}(s)\right)^{k_n} \qquad (6.40)$$

is due to Faa di Bruno (see [Knut73], page 50). Although this formula is elegant, its implementation is not straightforward. The difficulty is in obtaining the n-tuples that satisfy the conditions

$$k_1 + k_2 + \cdots + k_n = j$$
$$k_1 + 2k_2 + \cdots + nk_n = n \qquad k_1, k_2, \ldots, k_n \geq 0$$

To illustrate this we use Eq. (6.40) to derive the first four derivatives given in Eq. (6.39):

n=1: $k_1 = 1$ must hold:

$$j = 0 : k_1 = 0 \rightarrow \text{ no solution}$$
$$j = 1 : k_1 = 1 \rightarrow (1)$$

which yields

$$\mathbf{C}^{w(1)}(s) = \mathbf{C}^{w(1)}(u)f^{(1)}(s)$$

n=2: $k_1 + 2k_2 = 2$ must hold:

$$j = 0 : k_1 + k_2 = 0 \rightarrow \text{ no solution}$$
$$j = 1 : k_1 + k_2 = 1 \rightarrow (0,1)$$
$$j = 2 : k_1 + k_2 = 2 \rightarrow (2,0)$$

yielding $\mathbf{C}^{w(2)}(s) = \mathbf{C}^{w(1)}(u)f^{(2)}(s) + \mathbf{C}^{w(2)}(u)\left(f^{(1)}(s)\right)^2$

n=3: $k_1 + 2k_2 + 3k_3 = 3$ must hold:

$$j = 0 : k_1 + k_2 + k_3 = 0 \rightarrow \text{ no solution}$$
$$j = 1 : k_1 + k_2 + k_3 = 1 \rightarrow (0,0,1)$$
$$j = 2 : k_1 + k_2 + k_3 = 2 \rightarrow (1,1,0)$$
$$j = 3 : k_1 + k_2 + k_3 = 3 \rightarrow (3,0,0)$$

from which we obtain

$$\mathbf{C}^{w(3)}(s) = \mathbf{C}^{w(1)}(u)f^{(3)}(s) + 3\mathbf{C}^{w(2)}(u)f^{(1)}(s)f^{(2)}(s) + \mathbf{C}^{w(3)}(u)\left(f^{(1)}(s)\right)^3$$

n=4: $k_1 + 2k_2 + 3k_3 + 4k_4 = 4$ must hold:

$$j = 0 : k_1 + k_2 + k_3 + k_4 = 0 \rightarrow \text{ no solution}$$
$$j = 1 : k_1 + k_2 + k_3 + k_4 = 1 \rightarrow (0,0,0,1)$$
$$j = 2 : k_1 + k_2 + k_3 + k_4 = 2 \rightarrow (1,0,1,0) \quad \text{and} \quad (0,2,0,0)$$
$$j = 3 : k_1 + k_2 + k_3 + k_4 = 3 \rightarrow (2,1,0,0)$$
$$j = 4 : k_1 + k_2 + k_3 + k_4 = 4 \rightarrow (4,0,0,0)$$

from which we obtain

$$\mathbf{C}^{w(4)}(s) = \mathbf{C}^{w(1)}(u)f^{(4)}(s) + \mathbf{C}^{w(2)}(u)\left[4f^{(1)}(s)f^{(3)}(s) + 3\left(f^{(2)}(s)\right)^2\right]$$
$$+ 6\mathbf{C}^{w(3)}(u)\left(f^{(1)}(s)\right)^2 f^{(2)}(s) + \mathbf{C}^{w(4)}(u)\left(f^{(1)}(s)\right)^4$$

Given the derivatives of $\mathbf{C}^w(s)$ in terms of those of $\mathbf{C}^w(u)$ and $f(s)$, the general formulas for $\mathbf{Q}_i^w$ are

$$\mathbf{Q}_i^w = \frac{(pq-i)!\Delta s^i}{(pq)!}\mathbf{C}^{w(i)}(s=c) + \sum_{j=0}^{i-1}(-1)^{i+j-1}\binom{i}{j}\mathbf{Q}_j^w \qquad (6.41)$$

where $\Delta s^i = (d-c)^i$, $i = 1, \ldots, m_\ell$, and

$$\mathbf{Q}_{pq-i}^w = (-1)^i\frac{(pq-i)!\Delta s^i}{(pq)!}\mathbf{C}^{w(i)}(s=d) + \sum_{j=0}^{i-1}(-1)^{i+j-1}\binom{i}{j}\mathbf{Q}_{pq-j}^w \qquad (6.42)$$

for $i = 1, \ldots, m_r$. For example, the first five control points from the left are

$$\mathbf{Q}_0^w = \mathbf{P}_0^w$$
$$\mathbf{Q}_1^w = \frac{\Delta s}{pq}\mathbf{C}^{w(1)}(s=c) + \mathbf{Q}_0^w$$
$$\mathbf{Q}_2^w = \frac{\Delta s^2}{pq(pq-1)}\mathbf{C}^{w(2)}(s=c) - \mathbf{Q}_0^w + 2\mathbf{Q}_1^w$$
$$\mathbf{Q}_3^w = \frac{\Delta s^3}{pq(pq-1)(pq-2)}\mathbf{C}^{w(3)}(s=c) + \mathbf{Q}_0^w - 3\mathbf{Q}_1^w + 3\mathbf{Q}_2^w$$
$$\mathbf{Q}_4^w = \frac{\Delta s^4}{pq(pq-1)(pq-2)(pq-3)}\mathbf{C}^{w(4)}(s=c) - \mathbf{Q}_0^w + 4\mathbf{Q}_1^w - 6\mathbf{Q}_2^w + 4\mathbf{Q}_3^w \qquad (6.43)$$

and from the right are

$$\begin{aligned}
\mathbf{Q}^w_{pq} &= \mathbf{P}^w_n \\
\mathbf{Q}^w_{pq-1} &= -\frac{\Delta s}{pq}\mathbf{C}^{w(1)}(s=d) + \mathbf{Q}^w_{pq} \\
\mathbf{Q}^w_{pq-2} &= \frac{\Delta s^2}{pq(pq-1)}\mathbf{C}^{w(2)}(s=d) - \mathbf{Q}^w_{pq} + 2\mathbf{Q}^w_{pq-1} \\
\mathbf{Q}^w_{pq-3} &= -\frac{\Delta s^3}{pq(pq-1)(pq-2)}\mathbf{C}^{w(3)}(s=d) + \mathbf{Q}^w_{pq} - 3\mathbf{Q}^w_{pq-1} + 3\mathbf{Q}^w_{pq-2} \\
\mathbf{Q}^w_{pq-4} &= \frac{\Delta s^4}{pq(pq-1)(pq-2)(pq-3)}\mathbf{C}^{w(4)}(s=d) - \mathbf{Q}^w_{pq} \\
&\quad + 4\mathbf{Q}^w_{pq-1} - 6\mathbf{Q}^w_{pq-2} + 4\mathbf{Q}^w_{pq-3}
\end{aligned} \tag{6.44}$$

A linear reparameterization function

$$u = \alpha s + \beta \qquad s \in [c,d] \tag{6.45}$$

is an interesting special case. Examining Eq. (6.39), note that

$$\mathbf{C}^{w(i)}(s) = \mathbf{C}^{w(i)}(u)\big(f^{(1)}(s)\big)^i = \alpha^i \mathbf{C}^{w(i)}(u) \tag{6.46}$$

Then $$\mathbf{Q}^w_1 = \frac{\Delta s}{pq}\mathbf{C}^{w(1)}(s) + \mathbf{Q}^w_0 = \frac{\alpha\Delta s}{\Delta u}(\mathbf{P}^w_1 - \mathbf{P}^w_0) + \mathbf{P}^w_0 \tag{6.47}$$

From $a = \alpha c + \beta$ and $b = \alpha d + \beta$ we obtain

$$\alpha = \frac{b-a}{d-c} = \frac{\Delta u}{\Delta s} \tag{6.48}$$

Substituting Eq. (6.48) into Eq. (6.47) yields

$$\mathbf{Q}^w_1 = \mathbf{P}^w_1$$

Using Eqs. (6.41), (6.46), and (6.48), one can show by induction that

$$\mathbf{Q}^w_i = \mathbf{P}^w_i \qquad \text{for all } i \tag{6.49}$$

Hence, neither the degree nor the (homogeneous) control points change; the linear function of Eq. (6.45) simply changes the parameter bounds from $[a,b]$ to $[c,d]$.

Now let $\mathbf{C}^w(u)$ be an arbitrary pth-degree NURBS curve on $u \in [a,b]$ with knot vector U, and let $u = f(s)$ be a qth-degree piecewise polynomial reparameterization function on $s \in [c,d]$. Now $f(s)$ can be in any form, but for convenience of terminology let us assume that it is in B-spline form, i.e.

$$u = f(s) = \sum_{i=0}^{n} N_{i,q}(s) f_i \tag{6.50}$$

where the $\{f_i\}$ are scalars. Denote the knot vector by S; then $\mathbf{C}^w\big(f(s)\big)$ is a pqth-degree NURBS curve on $s \in [c,d]$, and its knots and control points can be computed as follows:

1. let $\{s_i\}$ denote the set of distinct internal knots of $f(s)$, and let $\{u_i\} = \{f(s_i)\}$ denote their images; use knot refinement to insert all u_i and all original internal knots of U until they all have multiplicity p. $\mathbf{C}^w(u)$ is then in piecewise Bézier form; denote the refined knot vector by U';
2. let
$$s = g(u) = f^{-1}(s) \tag{6.51}$$
be the inverse function of $f(s)$; then form the new knot vector, S', whose distinct knots are the images $s_i = g(u_i)$ of the distinct knots of U'. All internal knots in S' appear with multiplicity pq;
3. use Eqs. (6.39)–(6.44) and the Δs_i obtained from the new s knots in Step 2 to compute the new control points of $\mathbf{C}^w(s)$, which is also in piecewise Bézier form;
4. apply knot removal to $\mathbf{C}^w(s)$ and S' to obtain the minimal representation of $\mathbf{C}^w(s)$. The continuity of $\mathbf{C}^w(s)$ and the multiplicities of its knots are known, hence general knot removal (as in Algorithm `A5.5`) is not required. Let s_i be a knot in S', and let $u_i = f(s_i)$; denote by m^u_i and m^s_i the multiplicities of u_i and s_i in the original knot vectors U and S, respectively $(m^u_i, m^s_i \geq 0)$. Then the multiplicity m_i of s_i in S' is
$$\begin{aligned}
m_i &= pq - p + m^u_i \qquad \text{if } m^s_i = 0 \\
m_i &= pq - q + m^s_i \qquad \text{if } m^u_i = 0 \\
m_i &= \max(pq - p + m^u_i, pq - q + m^s_i) \qquad \text{if } m^u_i \neq 0,\ m^s_i \neq 0
\end{aligned} \tag{6.52}$$

If $u = f(s) = \alpha s + \beta$, then $s = g(u) = (u-\beta)/\alpha$. The (homogeneous) control points do not change, and the new knots are obtained from the original ones by

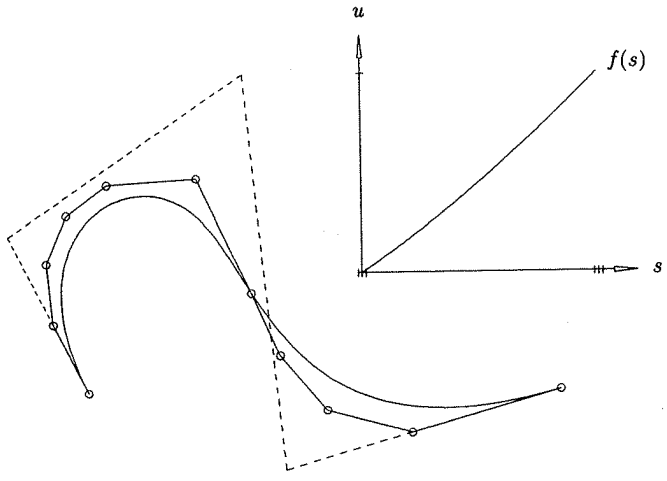
$$s_i = g(u_i) \tag{6.53}$$

For example, the knot vectors $U = \{0,0,0,\frac{1}{3},1,1,1\}$ and $S' = \{1,1,1,3,7,7,7\}$ are equivalent (define the same curve), since $s = 6u + 1$ maps U to S'.

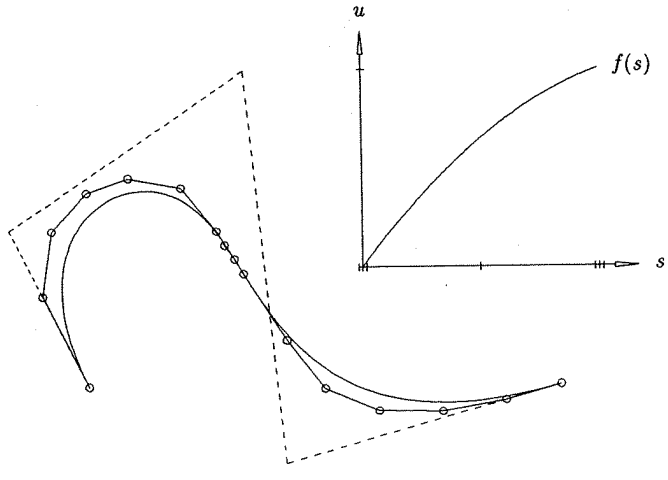
Figures 6.11a–6.11d show reparameterization examples. The curve is a cubic B-spline curve defined on $U = \{0,0,0,0,\frac{1}{2},1,1,1,1\}$. In Figure 6.11a the reparameterization function (shown in the upper right corner) is defined by the scalars $F = \{0, \frac{2}{5}, 1\}$, on the knot vector $S = \{0,0,0,1,1,1\}$. Figure 6.11b shows reparameterization defined by $F = \{0, \frac{2}{5}, \frac{9}{10}, 1\}$ and $S = \{0,0,0,\frac{1}{2},1,1,1\}$. In Figure 6.11c the function is given by $F = \{0, \frac{1}{2}, \frac{7}{10}, \frac{9}{10}, 1\}$ and $S = \{0,0,0,\frac{1}{2},\frac{1}{2},1,1,1\}$. Reparameterization with a linear B-spline function, given by $F = \{0,1\}$ and $S = \{0,0,1,1\}$, is depicted in Figure 6.11d.

We turn now to reparameterization with rational (or piecewise rational) functions of the form

$$u = f(s) = \frac{g(s)}{h(s)} \tag{6.54}$$



(a)

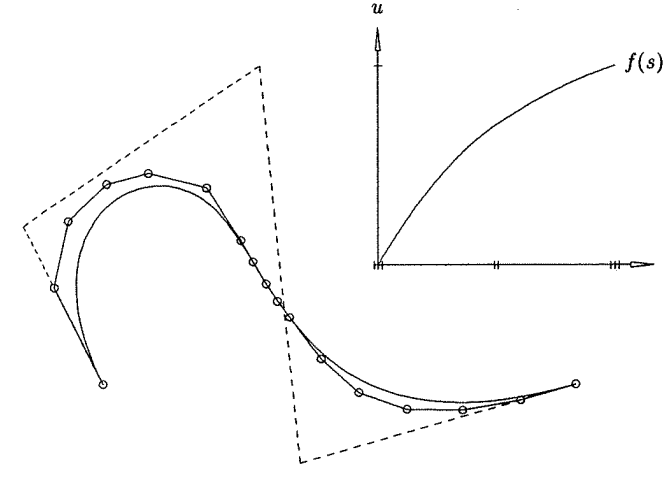


(b)

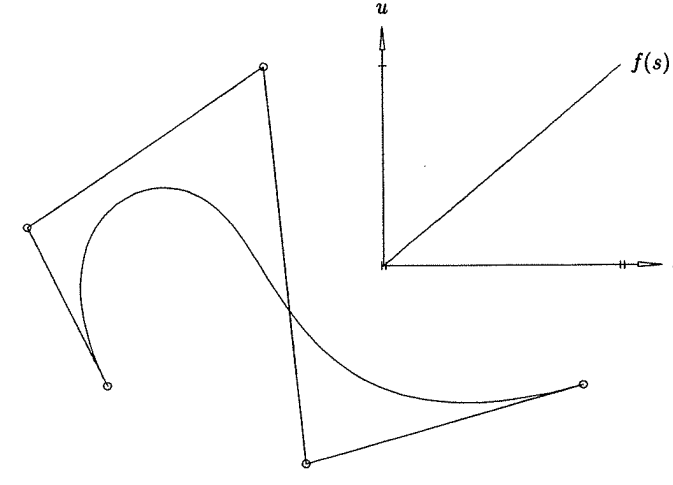
Figure 6.11. B-spline curve reparameterized with quadratic B-spline functions. (a) $F = \{0, 2/5, 1\}$, $S = \{0,0,0,1,1,1\}$; (b) $F = \{0, 2/5, 9/10, 1\}$, $S = \{0,0,0, 1/2, 1, 1, 1\}$.

As previously, if either $\mathbf{C}^w(u)$ or $f(s)$ has internal knots we can reparameterize in three steps:

1. insert knots to obtain $\mathbf{C}^w(u)$ as a piecewise Bézier curve;
2. reparameterize the Bézier segments;
3. remove unnecessary knots.



(c)



(d)

Figure 6.11. (*Continued.*) (c) $F = \{0, 1/2, 7/10, 9/10, 1\}$, $S = \{0,0,0, 1/2, 1/2, 1, 1, 1\}$; (d) $F = \{0, 1\}$, $S = \{0,0,1,1\}$.

Hence, we can assume that $\mathbf{C}^w(u)$ and $f(s)$ have no internal knots. Theoretically, one can reparameterize with a rational function using a technique similar to that used for polynomial functions. Let $r(u) = \sum_{i=0}^{p} a_i u^i$ be a pth-degree polynomial function. Then

$$r(s) = r\left(\frac{g(s)}{h(s)}\right) = \sum_{i=0}^{p} a_i \left(\frac{g(s)}{h(s)}\right)^i$$

and $$\bar{r}(s) = \big(h(s)\big)^p r(s) = \sum_{i=0}^{p} a_i \big(g(s)\big)^i \big(h(s)\big)^{p-i}$$

If $g(s)$ and $h(s)$ are qth-degree polynomials, then $\bar{r}(s)$ is a pqth-degree polynomial. Now let

$$\mathbf{C}^w(u) = \big(wx(u), wy(u), wz(u), w(u)\big)$$

be a pth-degree polynomial curve in homogeneous space. Then

$$\mathbf{C}^w(s) = \mathbf{C}^w\left(\frac{g(s)}{h(s)}\right) = \left[wx\left(\frac{g(s)}{h(s)}\right), \ldots, w\left(\frac{g(s)}{h(s)}\right)\right]$$

and

$$\bar{\mathbf{C}}^w(s) = \big(h(s)\big)^p \mathbf{C}^w(s) = \left[\big(h(s)\big)^p wx\left(\frac{g(s)}{h(s)}\right), \ldots, \big(h(s)\big)^p w\left(\frac{g(s)}{h(s)}\right)\right]$$

Concerning $\bar{\mathbf{C}}^w(s)$, note that

- $\bar{\mathbf{C}}^w(s)$ is a pqth-degree polynomial curve in homogeneous space;
- $\bar{\mathbf{C}}^w(s)$ and $\mathbf{C}^w(s)$ are different curves in homogeneous space, but they project to the same curve in Euclidean space, i.e., $\bar{\mathbf{C}}(s) = \mathbf{C}(s)$;
- using $\bar{\mathbf{C}}^w(s) = \big(h(s)\big)^p \mathbf{C}^w(g(s)/h(s))$, the Chain Rule, and end derivative formulas, one can compute the new homogeneous control points of $\bar{\mathbf{C}}^w(s)$.

The fundamental difference between reparameterization with a rational versus a polynomial function is:

- if $f(s)$ is polynomial then $\mathbf{C}^w(u)$ (and hence $\mathbf{C}(u)$) is reparameterized, but $\mathbf{C}^w(u)$ is not changed geometrically; furthermore, the end weights of $\mathbf{C}(u)$ do not change;
- if $f(s)$ is rational then a different curve, $\bar{\mathbf{C}}^w(s)$, is generated, which projects to the same (but reparameterized) curve, $\mathbf{C}(u)$. In general, the end weights of $\mathbf{C}(u)$ will change.

Example

Ex6.7 The circular arc given by Eq. (6.35) has a Bézier representation. Clearly, the Euclidean control points, $\{\mathbf{P}_i\} = \{(1,0),(1,1),(0,1)\}$, remain the same. The weights are determined by equating the power basis and Bézier forms of the denominator function of Eq. (6.35)

$$\big(2-\sqrt{2}\big)s^2 + \big(\sqrt{2}-2\big)s + 1 = (1-s)^2 w_0 + 2s(1-s)w_1 + s^2 w_2$$

Substituting $s = 0, 1, 1/2$, and solving for the w_i yields

$$\{w_i\} = \left\{1, \frac{\sqrt{2}}{2}, 1\right\} \tag{6.55}$$

Figure 6.12 shows the new curve, $\bar{\mathbf{C}}^w(s)$, and its projection, $\mathbf{C}(s)$. Compare this figure with Figure 1.22.

In practice, it is generally not necessary to use rational reparameterization functions of degree greater than one. Lee and Lucian [Lee91] give an excellent description of reparameterization using a linear rational function. We summarize their results here. Let

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \qquad u \in [a,b]$$

$$s = g(u) = \frac{\alpha u + \beta}{\gamma u + \delta}$$

$$u = f(s) = \frac{-\delta s + \beta}{\gamma s - \alpha} \qquad s \in [c,d]$$

($f(s)$ is the inverse of $g(u)$). Let

$$\mu(u) = \gamma u + \delta \qquad \lambda(s) = \gamma s - \alpha$$

To ensure that $g(u)$ and $f(s)$ are well-behaved, we assume

$$\alpha\delta - \gamma\beta > 0$$

$$\mu(u) \neq 0 \quad \text{for all } u \in [a,b]$$

$$\lambda(s) \neq 0 \quad \text{for all } s \in [c,d]$$

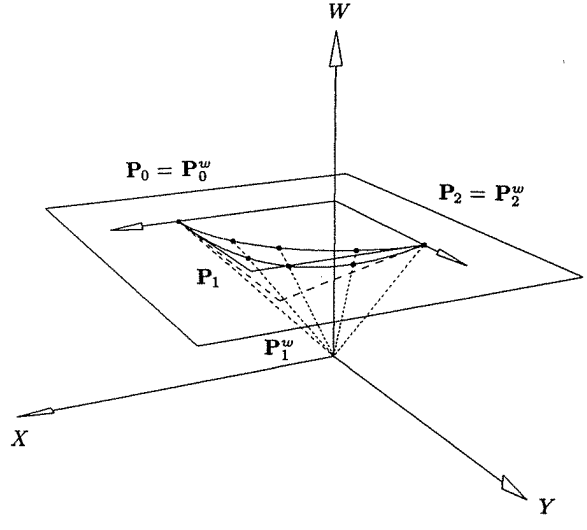


Figure 6.12. Homogeneous representation of the circular arc using weights $\{1, \sqrt{2}/2, 1\}$.

The reparameterized curve $\mathbf{C}(s)$ is obtained as follows:

1. the control points $\{\mathbf{P}_i\}$ do not change;
2. the new knots are the images under $g(u)$ of the original knots, $s_i = g(u_i)$;
3. the new weights $\{\bar{w}_i\}$ (modulo a common nonzero factor) are obtained by either

$$\bar{w}_i = w_i \prod_{j=1}^{p} \lambda(s_{i+j}) \tag{6.56}$$

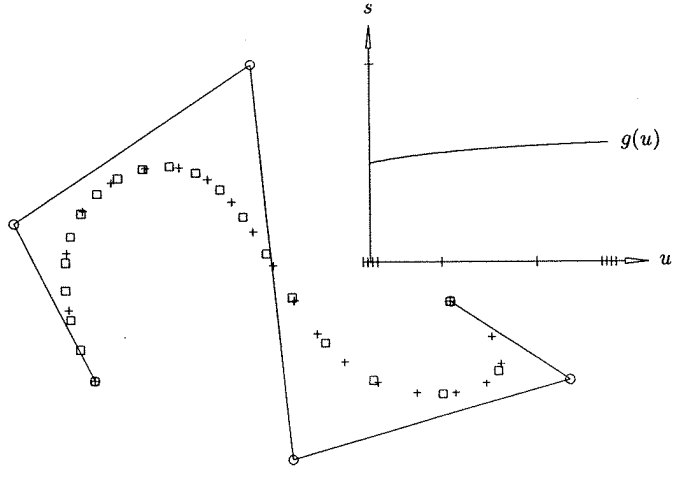
or

$$\bar{w}_i = \frac{w_i}{\prod_{j=1}^{p} \mu(u_{i+j})} \tag{6.57}$$

s_{i+j} and u_{i+j} are the new and old knots, respectively.

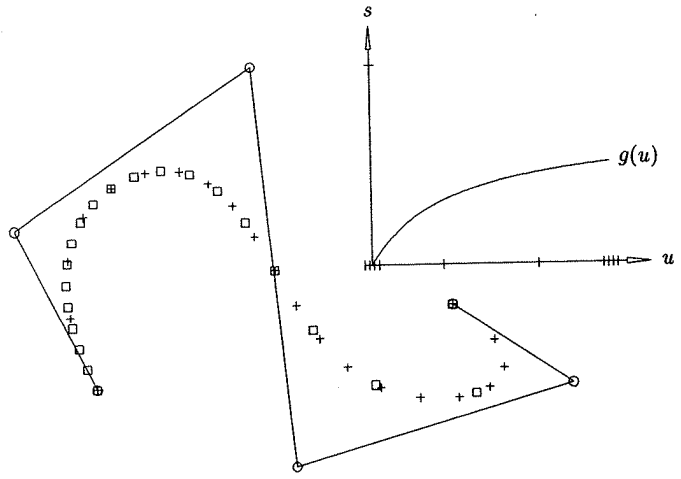
Figures 6.13a through 6.13c show reparameterization examples. The third-degree NURBS curve is defined originally on the knot vector $U = \{0, 0, 0, 0, {}^3/_{10}, {}^7/_{10}, 1, 1, 1, 1\}$. In Figures 6.13a and 6.13b all the weights are initially equal to one. In Figure 6.13c the original weights are $W = \{1, {}^1/_2, 2, 3, {}^1/_2, 1\}$. In Figure 6.13a the rational function is

$$g(u) = \frac{2u+1}{3u+2}$$

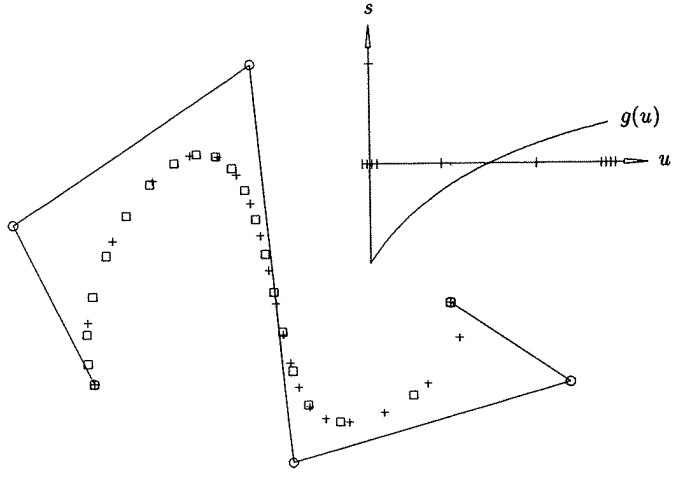


(a)

Figure 6.13. Rational reparameterization of NURBS curves; + shows the original parameterization, and □ illustrates the new parameterization. (a) $g(u) = {}^{(2u+1)}/_{(3u+2)}$; (b) $g(u) = {}^{2u}/_{(3u+1)}$; (c) $g(u) = {}^{(2u-1)}/_{(3u+2)}$.



(b)



(c)

Figure 6.13. (*Continued.*)

resulting in a new knot vector

$$U = \{0.5, 0.5, 0.5, 0.5, 0.55, 0.58, 0.6, 0.6, 0.6, 0.6\}$$

and weights

$$W = \{0.125, 0.08, 0.04, 0.01, 0.009, 0.008\}$$

+ shows the original parameterization, and □ marks the new one. Figure 6.13b applies

$$g(u) = \frac{2u}{3u+1}$$

yielding $U = \{0, 0, 0, 0, 0.31, 0.45, 0.5, 0.5, 0.5, 0.5\}$

and $W = \{1.0, 0.52, 0.16, 0.04, 0.02, 0.01\}$

Again, + shows the original and □ shows the new parameterization. Finally, in Figure 6.13c the function

$$g(u) = \frac{2u-1}{3u+2}$$

yields $U = \{-0.5, -0.5, -0.5, -0.5, -0.13, 0.09, 0.2, 0.2, 0.2, 0.2\}$

and $W = \{0.125, 0.04, 0.08, 0.05, 0.004, 0.008\}$

The formulas for surfaces are similar. For example, Eq. (6.56) generalizes to

$$\bar{w}_{i,j} = w_{i,j} \prod_{k=1}^{p} \lambda(s_{i+k}) \qquad j = 0, \ldots, m$$

in the u direction, and

$$\bar{w}_{i,j} = w_{i,j} \prod_{\ell=1}^{q} \lambda(t_{j+\ell}) \qquad i = 0, \ldots, n$$

in the v direction, where

$$s = g(u) = \frac{\alpha_u u + \beta_u}{\gamma_u u + \delta_u} \qquad t = h(v) = \frac{\alpha_v v + \beta_v}{\gamma_v v + \delta_v}$$

and $\lambda(s) = \gamma_u s - \alpha_u \qquad \lambda(t) = \gamma_v t - \alpha_v$

Equation (6.57) is generalized in a similar fashion.

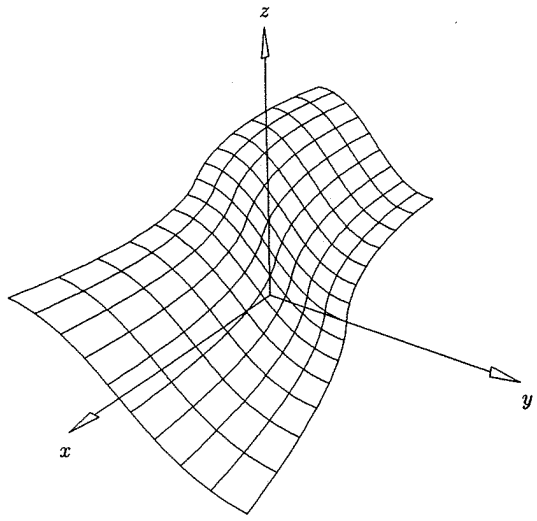
Figures 6.14a–6.14d show surface reparameterization examples. The test surface is a (3×2)-degree surface defined over

$$U = \{0, 0, 0, 0, 1, 1, 1, 1\} \qquad V = \{0, 0, 0, \tfrac{1}{2}, 1, 1, 1\}$$

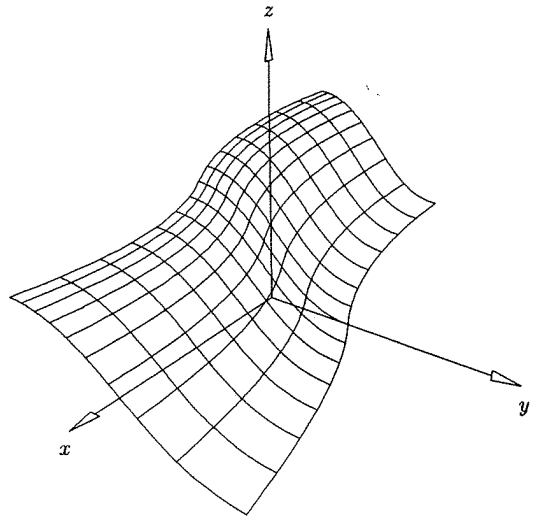
with weights all equal to one. The reparameterization functions are

$$g(u) = \frac{2u+1}{3u+2} \qquad h(v) = \frac{2v+1}{3v+2}$$

Figure 6.14a shows the original surface; note the even distribution of the parameter lines. In Figure 6.14b the surface was reparameterized in the u direction, in

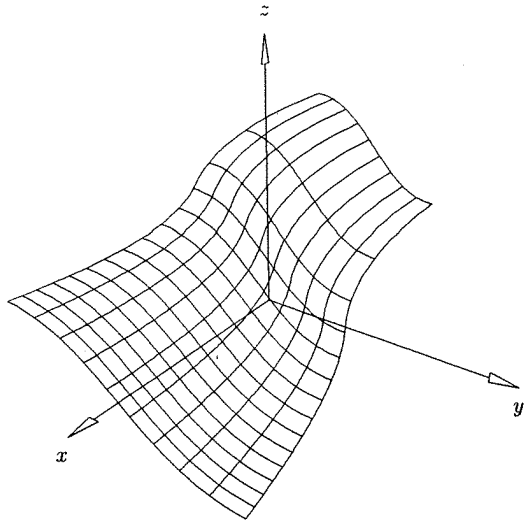


(a)



(b)

Figure 6.14. Rational reparameterization of NURBS surfaces using $g(u) = {}^{(2u+1)}/_{(3u+2)}$. (a) Original surface; (b) reparameterization in the u direction; (c) reparameterization in the v direction; (d) reparameterization in both directions.



(c)

Figure 6.14. (*Continued.*)

Figure 6.14c in the v direction, and in Figure 6.14d in both directions. Note the uneven distribution of parameter lines in Figures 6.14b–6.14d.

Now assume that the end weights of $\mathbf{C}(u)$ are not equal, i.e., $w_0 \neq w_n$. Since the linear rational function $s = g(u)$ has three degrees of freedom, it should be possible to determine $g(u)$ satisfying the three conditions

$$c = g(a) \qquad d = g(b) \qquad \bar{w}_0 = \bar{w}_n$$

Lee and Lucian [Lee91] derive a function satisfying these conditions

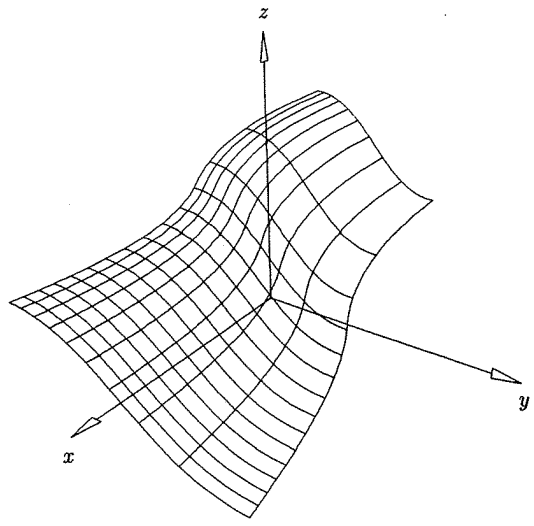
$$s = g(u) = \frac{\sqrt[p]{w_0}(b-u)c + \sqrt[p]{w_n}(u-a)d}{\sqrt[p]{w_0}(b-u) + \sqrt[p]{w_n}(u-a)} \tag{6.58}$$

Figure 6.15 shows a reparameterization example to make end weights equal. The original third-degree NURBS curve is defined by

$$U = \left\{0, 0, 0, 0, \frac{3}{10}, \frac{7}{10}, 1, 1, 1, 1\right\} \qquad W = \{2, 8, 3, 4, 2, 5\}$$

The new interval chosen is $[c, d] = [1/10, 4/5]$. The reparameterization function computed by Eq. (6.58) is

$$g(u) = \frac{1.242u + 0.126}{0.45u + 1.26}$$



(d)

Figure 6.14. (*Continued.*)

resulting in new knots and weights

$$U' = \{0.1, 0.1, 0.1, 0.1, 0.36, 0.63, 0.8, 0.8, 0.8, 0.8\}$$
$$W' = \{1, 3.61, 1.08, 1.06, 0.43, 1\}$$

where the end weights were chosen to be one. Again, + marks the original parameterization and □ the new parameterization.

We gave examples in Chapter 4 which showed that arbitrary changes in weights generally changed the shape of a rational NURBS curve (e.g., see Figures 4.2 and 4.4). In this section we changed weights without modifying the curve shape, the only change being in the parameterization of the curve. This suggests that for a given curve there must exist a relationship among the weights which determines the curve shape, that is, weight changes which maintain this relationship result in a reparameterization of the curve but not a change in shape. These relationships are called *shape invariants* or *shape factors*. For rational Bézier curves they depend only on the degree; for NURBS curves they depend on the degree and the knots. In general, the derivation of these shape invariants is quite complicated (see [Forr68; Vers75; Patt85; Lee87; Pieg91a]). We give a few here for reference; the c_i denote constants.

- Quadratic Bézier curve (conic):

$$\frac{w_0 w_2}{w_1^2} = c_1 \tag{6.59}$$

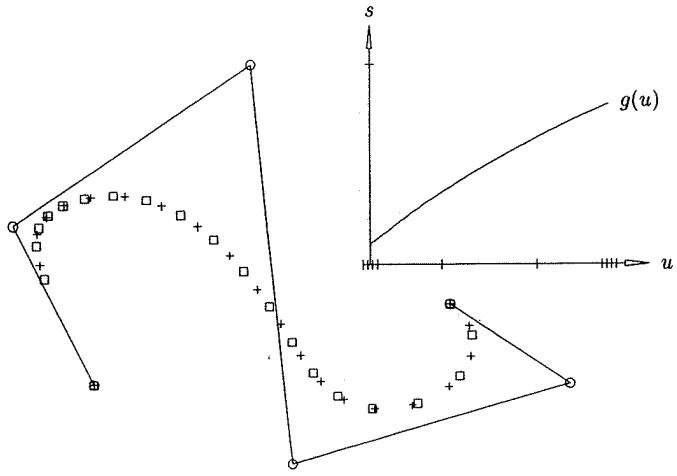


Figure 6.15. Curve reparameterization to make end weights equal.

Consider the circular arc, defined by either $\{w_i\} = \{1, 1, 2\}$ (Figure 1.19b) or $\{w_i\} = \{1, \sqrt{2}/2, 1\}$ (Eq. [6.55])

$$2 = \frac{(1)(2)}{1^2} = \frac{(1)(1)}{\left(\frac{\sqrt{2}}{2}\right)^2} = 2$$

Assuming fixed control points, any three weights which when substituted into Eq. (6.59) yield the value 2 produce the circular arc;

- Cubic Bézier curve:

$$\frac{w_0 w_2}{w_1^2} = c_1 \qquad \frac{w_1 w_3}{w_2^2} = c_2 \tag{6.60}$$

- A pth-degree Bézier curve:

$$\frac{w_0 w_2}{w_1^2} = c_1, \ldots, \frac{w_{p-2} w_p}{w_{p-1}^2} = c_{p-1} \tag{6.61}$$

- Quadratic NURBS curve $\mathbf{C}^w(u) = \sum_{i=0}^{n} N_{i,2}(u)\mathbf{P}_i^w$ on $U = \{u_0, \ldots, u_m\}$:

$$\frac{(u_{i+2} - u_{i+1})w_{i-1} + (u_{i+1} - u_i)w_i}{(u_{i+3} - u_{i+1})(u_{i+2} - u_i)w_i^2} \times$$

$$\frac{(u_{i+3} - u_{i+2})w_i + (u_{i+2} - u_{i+1})w_{i+1}}{(u_{i+3} - u_{i+1})(u_{i+2} - u_i)w_i^2} = c_i \tag{6.62}$$

for $i = 1, \ldots, n-1$.

We conclude the reparameterization of curves with mention of another interesting technique. Fuhr and Kallay [Fuhr92] develop a technique to interpolate a set of function values using a C^1 continuous piecewise linear rational function. Their method can be used to obtain C^1 continuous reparameterizations which do not raise the degree, but which map arbitrarily many internal points, $u_i = f(s_i)$.

A NURBS surface $\mathbf{S}^w(u, v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\mathbf{P}_{i,j}^w$ can be reparameterized in either the u, the v, or both directions. For example, it would be reparameterized with a polynomial function in the u direction by mapping the u knots to s knots (Eqs. [6.51] and [6.52]), and applying Eqs. (6.39) through (6.44) to the $m+1$ columns of control points.

6.5 Curve and Surface Reversal

In this section we use reparameterization to reverse the direction of a NURBS curve and the direction of the normal vectors of a surface.

Let $\mathbf{C}^w(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i^w$ be a NURBS curve on $u \in [a, b]$, and $u = f(s)$ a reparameterization function on $s \in [c, d]$. In Section 6.4 we assumed that $f'(s) > 0$ on $s \in [c, d]$. This was implicitly used in Example `Ex6.6` and in the derivation of Eqs. (6.41) through (6.44) and Eq. (6.49).

The concepts of Section 6.4 still apply if we assume $f'(s) < 0$ on $s \in [c, d]$. In particular, let

$$u = f(s) = -s + a + b \qquad s \in [a, b] \tag{6.63}$$

Observe that

$$b = f(a) \qquad a = f(b) \tag{6.64}$$

and that $f'(s) = -1$ on $s \in [a, b]$. The inverse of $f(s)$ is

$$s = g(u) = -u + a + b \tag{6.65}$$

From Eq. (6.53), the knots of $\mathbf{C}^w(s)$ are computed by

$$s_{m-p-i} = -u_{p+i} + a + b \qquad i = 1, \ldots, m - 2p - 1 \tag{6.66}$$

where u_i are the knots of $\mathbf{C}^w(u)$, $0 \le i \le m$. For example, if $U = \{0, 0, 0, 1, 3, 6, 6, 8, 8, 8\}$, then $S = \{0, 0, 0, 2, 2, 5, 7, 8, 8, 8\}$.

The control points $\mathbf{Q}_i^w$ of $\mathbf{C}^w(s)$ are computed as

$$\mathbf{Q}_0^w = \mathbf{C}^w\big(f(s = a)\big) = \mathbf{C}^w(u = b) = \mathbf{P}_n^w$$

$$\begin{aligned}
\mathbf{Q}_1^w &= \frac{s_{p+1} - a}{p}\mathbf{C}^{w(1)}(s = a) + \mathbf{Q}_0^w \\
&= \frac{b - u_{m-p-1}}{p}\mathbf{C}^{w(1)}(u = b)f'(s = a) + \mathbf{P}_n^w \\
&= \frac{b - u_{m-p-1}}{p}\frac{p}{b - u_{m-p-1}}(\mathbf{P}_n^w - \mathbf{P}_{n-1}^w)(-1) + \mathbf{P}_n^w \\
&= \mathbf{P}_{n-1}^w
\end{aligned}$$

In general

$$\mathbf{Q}_i^w = \mathbf{P}_{n-i}^w \tag{6.67}$$

The effect of Eqs. (6.63)–(6.67) is to reverse the direction of the curve. This is shown in Figure 6.16; note that the parameterization does not change.

Let $\mathbf{S}^w(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\mathbf{P}_{i,j}^w$ be a NURBS surface with knot vectors $U = \{u_0,\ldots,u_r\}$ and $V = \{v_0,\ldots,v_s\}$. A surface's parameterization can also be reversed. More specifically, u reversal produces

$$\mathbf{S}^w(s,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(s)N_{j,q}(v)\mathbf{Q}_{i,j}^w \tag{6.68}$$

on S and V, where

$$s_{r-p-i} = -u_{p+i} + u_0 + u_r \qquad i = 1,\ldots,r-2p-1 \tag{6.69}$$

and

$$\mathbf{Q}_{i,j}^w = \mathbf{P}_{n-i,j}^w \qquad i = 0,\ldots,n \quad j = 0,\ldots,m \tag{6.70}$$

Analogously, v reversal produces

$$\mathbf{S}^w(u,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(t)\mathbf{Q}_{i,j}^w \tag{6.71}$$

on U and T, where

$$t_{s-q-j} = -v_{q+j} + v_0 + v_s \qquad j = 1,\ldots,s-2q-1 \tag{6.72}$$

and

$$\mathbf{Q}_{i,j}^w = \mathbf{P}_{i,m-j}^w \qquad i = 0,\ldots,n \quad j = 0,\ldots,m \tag{6.73}$$

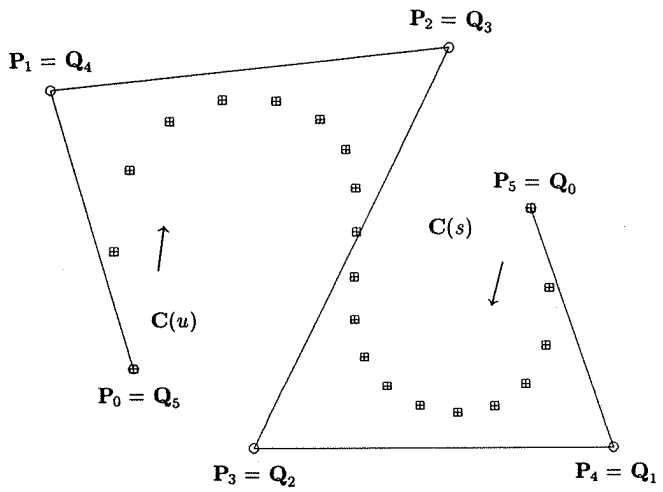


Figure 6.16. Curve reversal; no parameterization change occurs.

At an arbitrary parameter point (u,v), the surface normal vector is given by

$$\mathbf{N}(u,v) = \mathbf{S}_u(u,v) \times \mathbf{S}_v(u,v) \tag{6.74}$$

Clearly, the direction of $\mathbf{N}(u,v)$ can be reversed by reversing either $\mathbf{S}_u(u,v)$ or $\mathbf{S}_v(u,v)$ (but not both). This is accomplished by applying either Eqs. (6.68)–(6.70) or Eqs. (6.71)–(6.73), respectively.

Figures 6.17a–6.17c show surface reversal examples. In Figure 6.17a the original degree (3×2) surface is shown, as well as $\mathbf{S}_u$, $\mathbf{S}_v$, and $\mathbf{N}$. In Figures 6.17b and 6.17c the parameterizations are reversed in the u and v directions, respectively. Note that the normal vector is reversed in each case.

6.6 Conversion Between B-spline and Piecewise Power Basis Forms

The steps to convert a B-spline curve or surface to piecewise power basis form are:

1. use knot refinement, e.g., Algorithms `A5.4`–`A5.7`, to decompose the curve or surface into piecewise Bézier form;
2. applying matrix multiplications (change of basis) convert each Bézier curve segment (or surface patch) to power basis form;
3. apply matrix multiplications to reparameterize the power basis forms obtained in Step 2.

As usual, all equations are applied to four-dimensional control points and power basis coefficients; for convenience, we drop all w superscripts. Step 1 requires no further elaboration.

Suppose the jth curve segment, $\mathbf{C}_j(u)$, is defined on $u \in [u_j, u_{j+1}]$. Now the Bézier control points, $\mathbf{P}_i$, $i = 0,\ldots,p$, obtained in Step 1 for $\mathbf{C}_j(u)$ are valid, but our definition of a Bézier curve requires a $[0,1]$ parameterization (Eqs. [1.7] and [1.8]). Let

$$u = (u_{j+1} - u_j)s + u_j \tag{6.75}$$

Then $0 \le s \le 1$, and the jth segment has the Bézier form

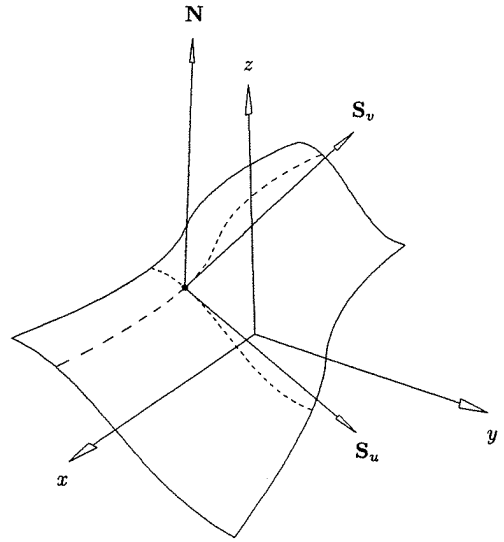
$$\mathbf{C}_j(s) = \sum_{i=0}^{p} B_{i,p}(s)\mathbf{P}_i = [B_{i,p}(s)]^T[\mathbf{P}_i] = \left[s^i\right]^T M_p[\mathbf{P}_i] \tag{6.76}$$

where $[B_{i,p}(s)]^T$ and $\left[s^i\right]^T$ are $1 \times (p+1)$ row vectors of scalar values, M_p is a $(p+1)\times(p+1)$ matrix of scalars, and $[\mathbf{P}_i]$ is a $(p+1)\times 1$ column vector of four-dimensional points. Setting

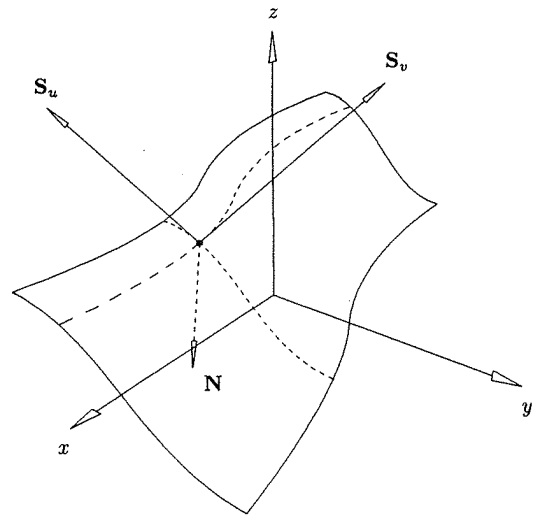
$$[\mathbf{a}_i] = M_p[\mathbf{P}_i] \tag{6.77}$$

we obtain

$$\mathbf{C}_j(s) = \left[s^i\right]^T[\mathbf{a}_i] \tag{6.78}$$

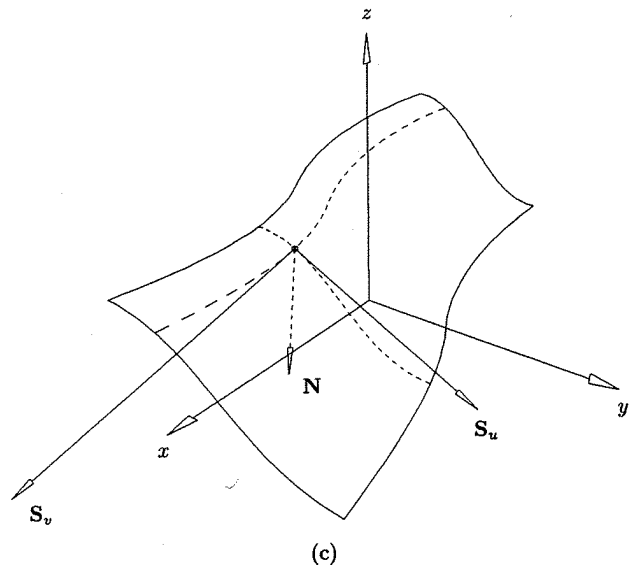


(a)



(b)

Figure 6.17. Surface reversal. (a) Original surface showing partial derivatives and the normal vector; (b) parameterization is reversed in the u direction; (c) parameterization is reversed in the v direction.



(c)

Figure 6.17. (*Continued.*)

which is the power basis form of the jth segment in the parameter s.

We now derive the matrices M_p for arbitrary p. As a warm-up and for reference, we compute the first three matrices. From Eq. (6.76) it follows that

$$[B_{i,p}(s)]^T = [s^i]^T M_p \tag{6.79}$$

For $p = 1, 2, 3$, and from Eq. (1.8), we obtain

$$[1-s \quad s] = [1 \quad s]\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

$$[(1-s)^2 \quad 2s(1-s) \quad s^2] = [1 \quad s \quad s^2]\begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$

$$[(1-s)^3 \quad 3s(1-s)^2 \quad 3s^2(1-s) \quad s^3] = [1 \quad s \quad s^2 \quad s^3]\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \tag{6.80}$$

Now we derive the general formula. Let $M_p(i, j)$ denote the element of M_p in

the ith row and the jth column. From Eq. (1.8) we have

$$(1-s)^p = \binom{p}{0} s^0 (1-s)^p = B_{0,p}(s) = \left[s^i\right]^T [M_p(i,0)]$$

$$= [1 \quad s \quad \cdots \quad s^p] \begin{bmatrix} M_p(0,0) \\ \vdots \\ M_p(p,0) \end{bmatrix}$$

From elementary algebra, expanding $(1-s)^p$ yields the binomial coefficients with alternating signs; hence

$$M_p(i,0) = (-1)^i \binom{p}{i} \qquad i = 0, \ldots, p \tag{6.81}$$

In general, the kth column of M_p is defined by

$$\binom{p}{k} s^k (1-s)^{p-k} = B_{k,p}(s) = \left[s^i\right]^T [M_p(i,k)]$$

Examining $\binom{p}{k} s^k (1-s)^{p-k}$ reveals that

- expanding the term $(1-s)^{p-k}$ yields the binomial coefficients

$$(-1)^j \binom{p-k}{j} \qquad j = 0, \ldots, p-k$$

- multiplying by s^k causes $M_p(j,k) = 0$ for $j = 0, \ldots, k-1$, and the values above are shifted down the column by k rows;
- all elements of the kth column are multiplied by $\binom{p}{k}$.

In summary, for $k = 0, \ldots, p$, the kth column of M_p is given by

$$M_p(j,k) = \begin{cases} 0 & j = 0, \ldots, k-1 \\ (-1)^{j-k} \binom{p}{k} \binom{p-k}{j-k} & j = k, \ldots, p \end{cases} \tag{6.82}$$

We leave it as an exercise to prove that

$$M_p(j,k) = M_p(p-k, p-j) \tag{6.83}$$

Note additional properties of the matrix M_p:

- the diagonal elements are the binomial coefficients for degree p;
- the elements in the last row and in the first column are also the degree p binomial coefficients, but with alternating signs;
- the elements in each row and each column sum to zero except in the first row and last column.

Equation (6.83) says that M_p is symmetric about the diagonal which runs from the bottom left element to the top right element.

Algorithm A6.1 computes M_p efficiently. An array `bin[i][j]`, which contains the precomputed binomial coefficients, is used.

```
ALGORITHM A6.1
  BezierToPowerMatrix(p,M)
    { /* Compute pth degree Bézier matrix  */
      /* Input:  p  */
      /* Output: M  */
    for (i=0; i<p; i++)  /* Set upper triangle to zero */
      for (j=i+1; j<=p; j++)   M[i][j] = 0.0;
    M[0][0] = M[p][p] = 1.0;  /* Set corner elements */
    if (p mod 2)  M[p][0] = -1.0;
      else        M[p][0] = 1.0;
    sign = -1.0;
    for (i=1; i<p; i++)  /* Compute first column, last row, */
      {           /* and the diagonal */
      M[i][i] = bin[p][i];
      M[i][0] = M[p][p-i] = sign*M[i][i];
      sign = -sign;
      }
    /* Compute remaining elements */
    k1 = (p+1)/2;    pk = p-1;
    for (k=1; k<k1; k++)
      {
      sign = -1.0;
      for (j=k+1; j<=pk; j++)
        {
        M[j][k] = M[pk][p-j] = sign*bin[p][k]*bin[p-k][j-k];
        sign = -sign;
        }
      pk = pk-1;
      }
    }
```

We now come to Step 3 of converting a B-spline curve to piecewise power basis, namely, reparameterizing Eq. (6.78) to obtain

$$\mathbf{C}_j(u) = \left[u^i\right]^T [\mathbf{b}_i] \tag{6.84}$$

the power basis segment in the original parameter, u. We work through the details for $p = 2$, then give the general formula without proof. Solving Eq. (6.75) for s yields

$$s = \frac{1}{u_{j+1} - u_j} u - \frac{u_j}{u_{j+1} - u_j} = cu + d \tag{6.85}$$

For $p = 2$, substituting $s = cu + d$ into Eq. (6.78) and equating the result with Eq. (6.84) yields

$$\left[s^i\right]^T \left[\mathbf{a}_i\right] = \left[(cu+d)^i\right]^T \left[\mathbf{a}_i\right] = \left[1 \quad cu+d \quad c^2u^2+2cdu+d^2\right]^T \left[\mathbf{a}_i\right]$$
$$= \left[1 \quad u \quad u^2\right] R \left[\mathbf{a}_i\right] = \left[1 \quad u \quad u^2\right] \left[\mathbf{b}_i\right] \quad (6.86)$$

where
$$R = \begin{bmatrix} 1 & d & d^2 \\ 0 & c & 2cd \\ 0 & 0 & c^2 \end{bmatrix}$$

From Eq. (6.86) it follows that

$$R\left[\mathbf{a}_i\right] = \left[\mathbf{b}_i\right] \quad (6.87)$$

The $\left[\mathbf{b}_i\right]$ are the desired power basis coefficients.

In general, for $s = cu + d$ the pth-degree reparameterization matrix is given by

$$R_p = \left[r_{i,j}\right]$$

where
$$r_{i,j} = \begin{cases} 0 & j < i \\ \binom{j}{i} c^i d^{j-i} & i \le j \quad i,j = 0,\ldots,p \end{cases} \quad (6.88)$$

Combining Eqs. (6.77) and (6.87), we obtain

$$\left[\mathbf{b}_i\right] = R_p M_p \left[\mathbf{P}_i\right] \quad (6.89)$$

for the power basis coefficients in terms of the Bézier control points. Figure 6.18 shows an example of B-spline to power basis conversion. A cubic NURBS curve with three segments was converted to piecewise power basis form. Figure 6.18 illustrates the curve and marks its second segment. Note that the segment is defined by the middle four control points. The middle curve segment is also shown, as are the vector coefficients, $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$, which are scaled by $1/10$ to make them fit into the figure. Note that the vector coefficients have no geometric relationship with the curve. The polynomial piece is defined on $[3/10, 7/10]$.

We complete the discussion of B-spline to power basis conversion with two final remarks:

- the IGES parametric splines (Entities 112 and 114) [IGE93] assume a parameterization for each segment starting at zero; this implies $d = 0$ in Eq. (6.85), and hence R_p reduces to a diagonal matrix with elements

$$r_{i,j} = \begin{cases} 0 & i \ne j \\ c^i & i = j \end{cases} \qquad i = 0,\ldots,p \quad (6.90)$$

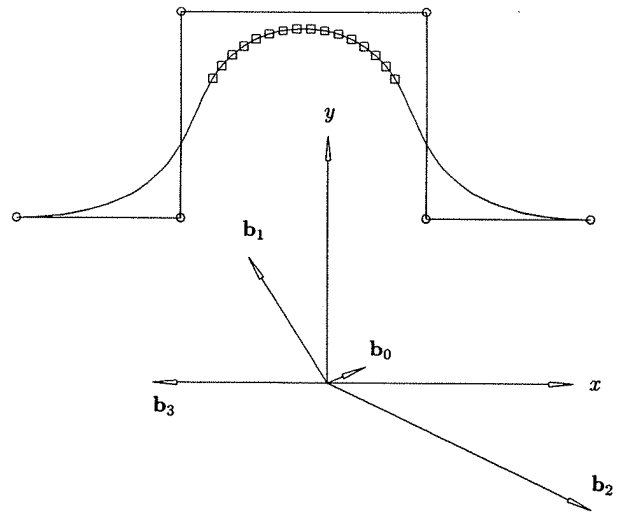


Figure 6.18. Converting a NURBS curve into a piecewise polynomial curve represented in power basis form.

- for surfaces, Eq. (6.89) becomes

$$\left[\mathbf{b}_{i,j}\right] = R_p M_p \left[\mathbf{P}_{i,j}\right] M_q^T R_q^T \quad (6.91)$$

where R_p and R_q^T are the reparameterization matrices taking u to s, and v to t, respectively ($0 \le s,t \le 1$).

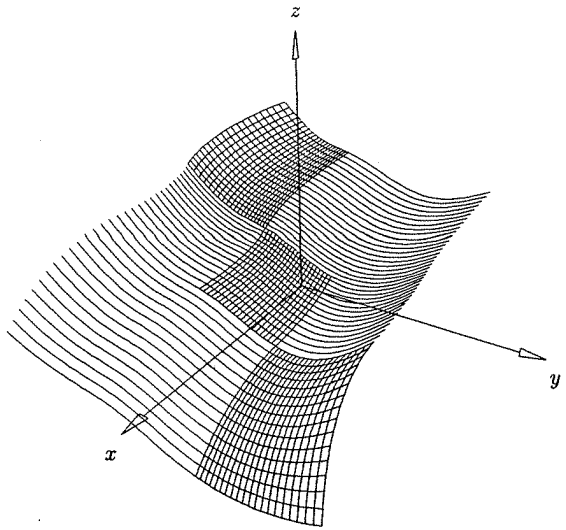
Figures 6.19a and 6.19b show B-spline surface to power basis conversion. The test surface is a degree (3×2) surface defined on

$$U = \left\{0,0,0,0,\frac{3}{10},\frac{7}{10},1,1,1,1\right\} \qquad V = \left\{0,0,0,\frac{3}{10},\frac{3}{5},1,1,1\right\}$$

Three polynomial patches are shown in Figure 6.19a: patch $(0,2)$ defined on $[0, 3/10] \times [3/5, 1]$; patch $(1,1)$ defined on $[3/10, 7/10] \times [3/10, 3/5]$; and patch $(2,0)$ defined on $[7/10, 1] \times [0, 3/10]$. Figure 6.19b illustrates some of the vector coefficients defining the patch $(1,1)$ in power basis form. The vectors are scaled by $1/2$ for better visualization. Note the lack of any geometric relationship between the patch geometry and the vector coefficients. Compare this figure with Figure 5.25 in Chapter 5.

We now discuss the conversion of a piecewise power basis curve or surface to B-spline form. The steps are:

1. apply matrix multiplications to reparameterize the power basis segments (patches) to the interval $[0,1]$;



(a)

Figure 6.19. Converting a NURBS surface into a collection of polynomial patches represented in power basis form. (a) Patches with indexes (0,2), (1,1), and (2,0) are shown; (b) some of the vector coefficients defining patch (1,1).

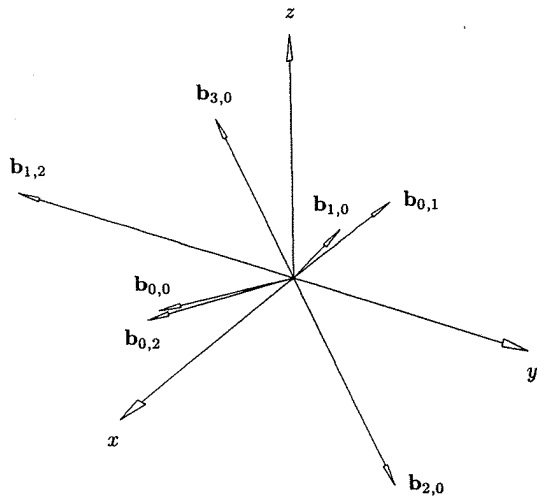
2. apply matrix multiplications to convert each power basis segment (patch) to Bézier form;
3. piece the segments (patches) together to form a B-spline curve or surface, with all internal knots having multiplicity equal to the degree (or degree plus one if C^0 continuity cannot be assumed); the values of the knots are those given by the parameter bounds of the original power basis segments (patches);
4. use knot removal to obtain the minimal representation of the B-spline curve or surface (Algorithm `A5.8` for curves, or see [Till92] for a surface algorithm).

Steps 3 and 4 require no further elaboration. Consider curves first. Let

$$\mathbf{C}_j(u) = \left[u^i\right]^T \left[\mathbf{a}_i\right] \tag{6.92}$$

be the power basis form of the jth curve segment, with $u \in [u_j, u_{j+1}]$. We need the corresponding Bézier form of the segment

$$\begin{aligned}\mathbf{C}_j(s) &= \left[B_{i,p}(s)\right]^T \left[\mathbf{P}_i\right] = \left[s^i\right]^T M_p \left[\mathbf{P}_i\right] \\ &= \left[u^i\right]^T R_p M_p \left[\mathbf{P}_i\right] = \left[u^i\right]^T \left[\mathbf{a}_i\right] = \mathbf{C}_j(u)\end{aligned}$$



(b)

Figure 6.19. (*Continued.*)

where $s \in [0, 1]$. Hence

$$\left[\mathbf{a}_i\right] = R_p M_p \left[\mathbf{P}_i\right]$$

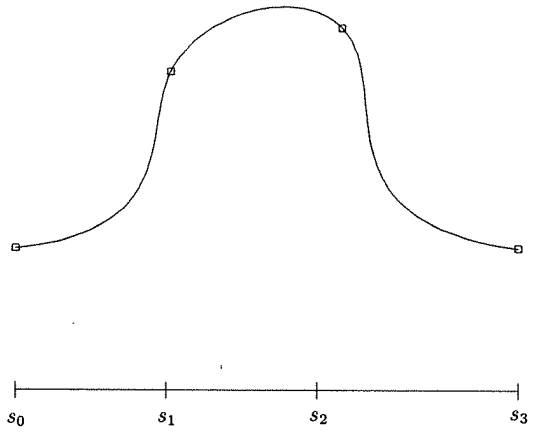
and

$$\left[\mathbf{P}_i\right] = M_p^{-1} R_p^{-1} \left[\mathbf{a}_i\right] \tag{6.93}$$

Figures 6.20a through 6.20c show examples of power basis to NURBS conversion. Figure 6.20a illustrates a piecewise cubic power basis curve defined on $s_0 < s_1 < s_2 < s_3$. In Figure 6.20b each piece is converted to Bézier form, that is, it is a B-spline curve with triple internal knots. Figure 6.20c shows the B-spline curve after knot removal. Note that the original curve was C^2 continuous at s_1, but only C^1 continuous at s_2; thus, s_2 is a double knot in the final knot vector. It is easy to invert the matrices M_1, M_2, and M_3, shown in Eq. (6.80), to obtain

$$M_1^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{2} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$



(a)

Figure 6.20. Converting a piecewise polynomial curve into NURBS form. (a) Piecewise polynomial defined over $s_0 < s_1 < s_2 < s_3$; (b) converting each piece into Bézier form; (c) the NURBS representation after knot removal.

$$M_3^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & 0 & 0 \\ 1 & \frac{2}{3} & \frac{1}{3} & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{6.94}$$

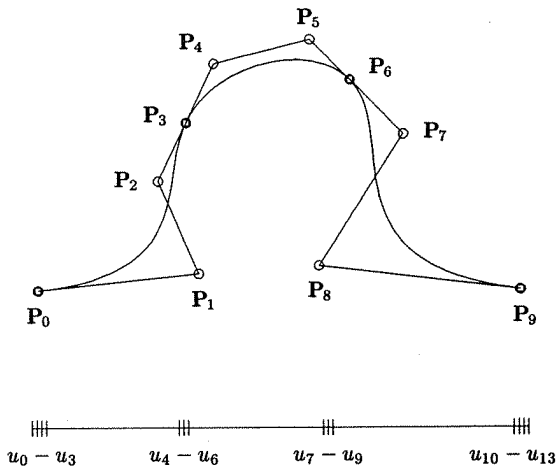
Given M_p, the formula for the kth column of M_p^{-1} is

$$M_p^{-1}(j,k) = \begin{cases} 0 & j = 0, \ldots, k-1 \\ \dfrac{1}{M_p(k,k)} & j = k \\ \dfrac{-\sum_{i=k}^{j-1} M_p(j,i) M_p^{-1}(i,k)}{M_p(j,j)} & j = k+1, \ldots, p \end{cases} \tag{6.95}$$

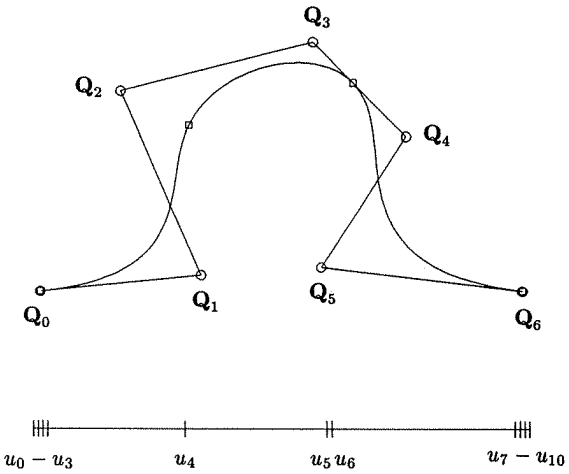
As was M_p, M_p^{-1} is symmetric about the diagonal running from the bottom left element to the top right element, hence

$$M_p^{-1}(j,k) = M_p^{-1}(p-k, p-j) \tag{6.96}$$

Algorithm `A6.2` computes M_p^{-1} (denoted by `MI`).



(b)



(c)

Figure 6.20. (*Continued.*)

```
ALGORITHM A6.2
  PowerToBezierMatrix(p,M,MI)
    { /*  Compute inverse of pth-degree Bézier matrix  */
      /*  Input:  p,M  */
      /*  Output: MI  */
```

```
for (i=0; i<p; i++)  /* Set upper triangle to zero */
  for (j=i+1; j<=p; j++)   MI[i][j] = 0.0;
/* Set first col, last row, and diagonal */
for (i=0; i<=p; i++)
  {
  MI[i][0] = MI[p][i] = 1.0;
  MI[i][i] = 1.0/M[i][i];
  }
/* Compute remaining elements */
k1 = (p+1)/2;    pk = p-1;
for (k=1; k<k1; k++)
  {
  for (j=k+1; j<=pk; j++)
    {
    d = 0.0;
    for (i=k; i<j; i++)  d = d-M[j][i]*MI[i][k];
    MI[j][k] = d/M[j][j];
    MI[pk][p-j] = MI[j][k];
    }
  pk = pk-1;
  }
}
```

Recalling Eqs. (6.85) and (6.86), we have $s = cu + d$ and

$$[s^i]^T = [(cu+d)^i]^T = [u^i]^T R_p \tag{6.97}$$

Inverting Eq. (6.85) yields

$$u = \frac{1}{c}s - \frac{d}{c} = (u_{j+1} - u_j)s + u_j = \gamma s + \delta$$

From Eq. (6.97) we have

$$[u^i]^T = [(\gamma s + \delta)^i]^T = [s^i]^T R_p^{-1}$$

which implies that R_p^{-1} is simply R_p with c replaced by γ and d by δ. To summarize

$$R_p^{-1} = [r_{i,j}]$$

with

$$r_{i,j} = \begin{cases} 0 & j < i \\ \binom{j}{i}\gamma^i \delta^{j-i} & i \le j \quad i,j = 0,\ldots,p \end{cases} \tag{6.98}$$

where $\gamma = u_{j+1} - u_j \qquad \delta = u_j$

We conclude with two remarks:

- for the IGES parametric splines, $\delta = 0$ and R_p^{-1} reduces to the diagonal matrix [IGE93] with elements

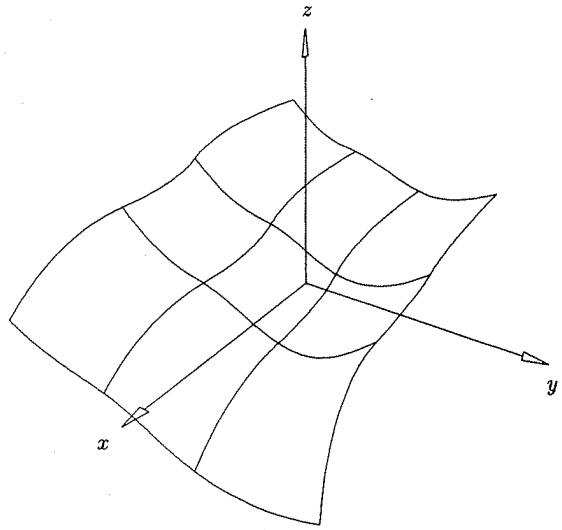
$$r_{i,j} = \begin{cases} 0 & i \neq j \\ \gamma^i & i = j \end{cases} \qquad i = 0,\ldots,p \tag{6.99}$$

- for surfaces Eq. (6.93) becomes

$$[\mathbf{P}_{i,j}] = M_p^{-1} R_p^{-1} [\mathbf{a}_{i,j}] (R_q^{-1})^T (M_q^{-1})^T$$

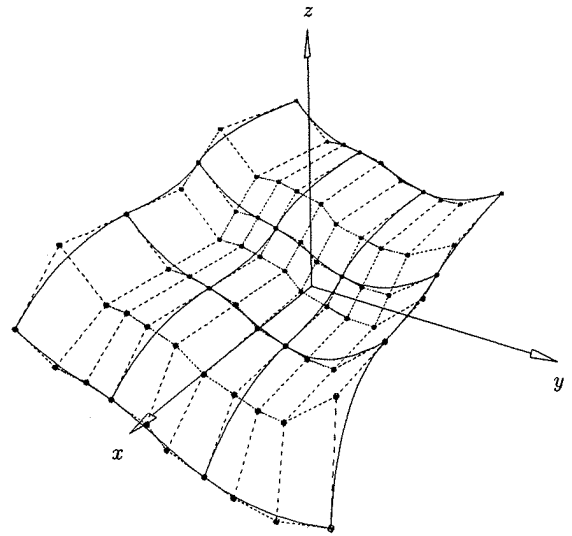
where R_p^{-1} and $(R_q^{-1})^T$ are the reparameterization matrices taking s to u and t to v, respectively $(0 \le s,t \le 1)$.

Figures 6.21a–6.21c illustrate power basis to NURBS surface conversion. The test surface is a collection of nine degree (3×2) polynomial patches shown in Figure 6.21a. Figure 6.21b illustrates the piecewise Bézier representation. Knot removal produces the surface shown in Figure 6.21c.

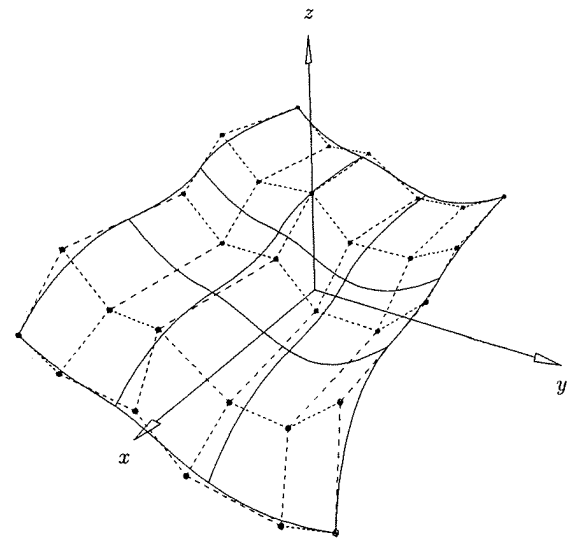


(a)

Figure 6.21. Converting a piecewise polynomial surface into NURBS form. (a) Piecewise polynomial surface; (b) converting each patch into Bézier form; (c) the NURBS representation after knot removal.



(b)



(c)

Figure 6.21. (*Continued.*)

EXERCISES

6.1. Consider the perspective projection of Example Ex6.3. The Euclidean coordinates of $\bar{\mathbf{P}}_i^w$ are

$$\bar{\mathbf{P}}_i = \left(\frac{dx_i}{d - z_i}, \frac{dy_i}{d - z_i}, 0 \right)$$

Carry out the projection using Eqs. (6.17) and (6.18); compare the results.

6.2. Use a quadratic function $u = f(s) = as^2 + bs + c$ to reparameterize the Bézier circular arc $\mathbf{C}(u)$ given by $\{\mathbf{P}_i\} = \{(1,0),(1,1),(0,1)\}$ and $\{w_i\} = \{1,1,2\}$, so that

$$\mathbf{P} = \mathbf{C}\left(s = \frac{1}{2}\right) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right)$$

Three steps are required:

- determine the u value at which $\mathbf{P}$ is attained. Hint: use Eq. (6.28);
- compute the coefficients a, b, c of $f(s)$;
- use Eq. (6.38) to determine the new homogeneous control points.

Have the end weights changed?

6.3. A cubic curve $\mathbf{C}^w(u)$ on $U = \{0,0,0,0,1,1,3,5,7,7,7,7\}$ is reparameterized with a nonrational quadratic NURBS function, $f(s)$, on $S = \{0,0,0,2,3,6,6,7,7,7\}$. What will the knot vector of the reparameterized curve $\mathbf{C}^w(s)$ look like? What is the continuity of $\mathbf{C}^w(s)$ at the various knots?

6.4. Check Eq. (6.34) and Eqs. (6.55)–(6.58) against one another, that is:

- using Eq. (6.34), do Eq. (6.56) and Eq. (6.57) yield Eq. (6.55);
- does Eq. (6.58) produce Eq. (6.34)?

Remember, all weights can be multiplied by a common factor.

6.5. Derive Eq. (6.83).

6.6. Let the cubic curve $\mathbf{C}(u)$ be defined by $U = \{0,0,0,0,1,2,2,2,2\}$ and the control points $\{\mathbf{P}_i\} = \{(0,0),(1,1),(3,2),(5,1),(4,-1)\}$. Use knot insertion (by hand) to decompose the curve into two Bézier segments, then convert the two segments to power basis form. Check your result by computing some endpoint and end derivative values from the power basis segments, and from the original B-spline curve.

6.7. Let $s = cu + d$. Form R_3 and R_3^{-1}, and verify that $R_3 R_3^{-1} = I$ (I is the identity matrix).