General Matrix Representations for Bezier and B-Spline Curves *

Elaine Cohen and Richard F. Riesenfeld
Department of Computer Science, University of Utah, 3160 Merrill Engineering Building, Salt Lake City, Utah 84112, U.S.A.

The matrix representation for a Bezier curve or arbitrary degree is given as well as the analogous representation for the uniform B-spline of arbitrary degree. Special matrices for calculating the uniform open B-spline curve are derived. This form is convenient to implement in hardware or software, particularly if a matrix facility is present.

Keywords: Bezier curve, open B-spline curve, matrix representation, analogous representation.

Dr. Elaine Cohen holds MS (1970) and PhD (1974) degrees in mathematics from Syracuse University, and a BA (1968) in mathematics from Vassar College. Since 1974 she has held a faculty position in Computer Science at the University of Utah, where she is currently a Research Associate Professor and co-head of the Computer Aided Geometric Design Group. Present research is centered about the theoretical and algorithmic problems associated with geometric modelling.

Richard F. Riesenfeld received a PhD degree from the School of Computer and Information Science at Syracuse University, Syracuse, NY, in 1973. He is Professor and Chairman of Computer Science, and Head of the Computer Aided Geometric Design Group (CAGD) in Computer Science at the University of Utah, where he joined the faculty in 1972. Professor Riesenfeld has published and consulted in the area of computer graphics and CAD/CAM, and has therefore gained an acquaintance with both the academic issues and the industrial applications and motivations in the field. The B-spline method of curve and surface design, which he proposed in his doctoral thesis, is used by many industries in many countries. His major research focus for the last year has been the design and implementation of a CAGD system called Alpha 1 which advances the states-of-the-art in both high quality computer graphics and free-form surface models. This system is designed to exploit the capabilities accorded by the Oslo Algorithm (1980) on which he worked collaboratively.

1. Introduction

The notion of using a polygon to define the gross shape of a curve was introduced by Bezier for use in interactive computer aided design [2]. This scheme was the first successful system which provided acceptable control over the general shape character of the intended curve. The Bezier method involves Bernstein polynomial weighting functions and leads to the definition of polynomial curve forms. It usually is presented as below.

* This work was supported in part by the National Science Foundation (MCS 8004116) and the U.S. Army Research Office (DAAG29-81K-0111).

North-Holland Publishing Company
Computers in Industry 3 (1982) 9–15

0166-3615/82/0000–0000/$02.75

Definition 1: The Bezier curve $\gamma(t)$ associated with the polygon $P = P_0P_1\ldots P_m$ is

$$\gamma(t) = \sum_{i=0}^{m} \theta_i(t) P_i, \tag{1}$$

where $\theta_i(t) = \binom{m}{i} t^i (1-t)^{m-i}$ is the Bernstein polynomial of degree m.

In [4] Gordon and Riesenfeld generalized this method to include spline curves by incorporating the B-spline basis functions as the weighting functions instead of the Bernstein polynomials that Bezier used.

Definition 2: Let $X = \{x_i\}$ be a set of real numbers such that $x_i \leqslant x_{i+1}$. The i-th normalized B-spline basis function of order M is defined as:

$$N_{i,M}(x) = \begin{cases} \begin{cases} 1 & x_i < x \leqslant x_{i+1} \\ 0 & \text{otherwise} \end{cases} & M = 1 \\ \dfrac{x - x_i}{x_{i+M-1} - x_i} N_{i,M-1}(x) + \dfrac{(x_{i+M} - x)}{x_{i+M} - x_{i+1}} N_{i+1,M-1}(x), & M > 1 \end{cases} \tag{2}$$

This is known as the deBoor–Cox Algorithm. If the spacing between the knots is equal, say 1, and X is bi-infinite, then it is called a *uniform periodic basis*, and

$N_{j,M}(x) = N_{j-1,M}(x-1)$ for all j,

$$g_k(s;t) = (s-t)_+^{k-1} = \begin{cases} (s-t)^{k-1} & s \geqslant t \\ 0 & s < t \end{cases}$$

each basis function is a translate of the other.

$$M_{i,k}(t) = g_k(t_i, \ldots, t_{i+k}; t)$$

Another convenient formulation is

$$N_{0M}(t) = \frac{1}{(M-1)!} \sum_{i=0}^{M} (-1)^i \binom{M}{i} (t-i)_+^{M-1}, \tag{3}$$

$$N_{i,k}(t) = (t_{i+k} - t_i) M_{i,k}(t)$$

$$f[x_k, \ldots, x_{k+m}] = \frac{1}{m!\, h^m} \Delta^m f_k \qquad \Delta^n f_k = \sum_{j=0}^{n} (-1)^j \binom{n}{j} f_{n+k-j}$$

which is nonzero only the interval $(0, M)$ and $(t-i)_+ = \max(0, t-i)$.

If the X vector is $X = \{0, 0, 0, 0, 1, 2, \ldots, n, n, n, n\}$ the basis functions are called the uniform *open basis* functions. $(M-j)_+^{k-1}$

In both cases the basis functions have $C^{(2)}$ continuity, and in all cases the nonzero basis functions sum to one.

The functions θ_i of eqs. (1) are just the translates of $N_{0,M}$ in the uniform periodic case or the calculated $N_{i,M}$ in the uniform open case. $\gamma(t)$ is defined for t in the range $[0, M]^{i,M}$.

In the next sections we develop matrix formulations for all three above cases.

2. Bezier Matrix

Here we seek an $(m+1) \times (m+1)$ matrix B_m which allows us to write (1) as the matrix equation:

$$\gamma(t) = \left[t^m t^{m-1} \ldots t\ 1 \right] B_m P_m.$$

B_m must transform the power basis $T_m = [t^m t^{m-1} \ldots t\ 1]$ to the Bernstein basis $\theta_i(t)$ as in (1). Now

$$\theta_i(t) = \binom{m}{i} t^i (1-t)^{m-i} = \binom{m}{i} t^i \sum_{k=0}^{m-i} \binom{m-i}{k} 1^{m-i-k} (-1)^k (t)^k$$

$$= \sum_{k=0}^{m-i} \binom{m}{i} \binom{m-i}{k} (-1)^k t^{i+k} = \sum_{q=0}^{m-i} \binom{m}{i} \binom{m-i}{m-i-q} (-1)^{m-i-q} t^{m-q}$$

$$= T_m \begin{bmatrix} \binom{m}{i}\binom{m-i}{m-i}(-1)^{m-i} \\ \binom{m}{i}\binom{m-i}{m-i-1}(-1)^{m-i-1} \\ \vdots \\ \binom{m}{i}\binom{m-i}{1}(-1)^{1} \\ \binom{m}{i}\binom{m-i}{0}(-1)^{0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} i \text{ rows} \end{matrix}$$

We combine the previous vector formulations to get:

$$\gamma(t) = \sum_{j=0}^{m} \theta_j(t) P_j = [\theta_0(t)\ \theta_1(t) \dots\ \theta_m(t)] \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_m \end{bmatrix}$$

$$= T_m \begin{bmatrix} \binom{m}{0}\binom{m}{m}(-1)^m & \binom{m}{1}\binom{m-1}{m-1}(-1)^{m-1} & \dots & \binom{m}{m}\binom{m-m}{m-m}(-1)^0 \\ \binom{m}{0}\binom{m}{m-1}(-1)^{m-1} & \binom{m}{1}\binom{m-1}{m-2}(-1)^{m-2} & \dots & 0 \\ \vdots & \vdots & & \\ \binom{m}{0}\binom{m}{1}(-1)^1 & \binom{m}{1}\binom{m-1}{0}(-1)^0 & \dots & 0 \\ \binom{m}{0}\binom{m}{0}(-1)^0 & 0 & \dots & 0 \end{bmatrix} P_m,$$

so

$$B_m = (b_{ij})_{i,j=0}^{m}$$

where

$$b_{ij} = \begin{cases} \binom{m}{j}\binom{m-j}{m-i-j}(-1)^{m-i-j} & 0 \leqslant i+j < m \\ 0 & \text{otherwise} \end{cases}.$$

It is sometimes desirable to decompose B_m into two simpler matrices.

$$C = \begin{bmatrix} \binom{m}{m}(-1)^m & \binom{m-1}{m-1}(-1)^{m-1} & \dots & \binom{m-m}{m-m}(-1)^0 \\ \binom{m}{m-1}(-1)^{m-1} & \binom{m-1}{m-2}(-1)^{m-2} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \binom{m}{1}(-1)^1 & \binom{m-1}{0}(-1)^0 & & 0 \\ \binom{m}{0}(-1)^0 & 0 & \dots & 0 \end{bmatrix}$$

and

$$D = \begin{bmatrix} \binom{m}{0} & & & 0 \\ & \binom{m}{1} & & \\ & & \ddots & \\ 0 & & & \binom{m}{m} \end{bmatrix}$$

so

$$B_m = CD$$

and

$$\gamma(t) = T_m B_m P_m.$$

3. Uniform Periodic B-spline Matrix

In this section we find the analogous matrix formulation for (3) so that we can write:

$$\gamma(t) = T_{M-1} S_M P_{i,M-1}, \qquad 0 \leqslant t \leqslant 1$$

where

$$T_{M-1} = \left[t^{M-1} \dots t\ 1\right]$$

and

$$P_{i,M-1} = \left[P_i P_{i+1} \dots P_{i+M-1}\right]'$$

is the moving polygon segment associated with the polynomial span being generated under a moving window. S_M is the matrix we seek to define. Let

$q_{j,M}(t) = N_{0M}(t+j)$ for $j = 0, 1, \dots, M-1$ and $0 \leqslant t \leqslant 1$. j表示是哪一段，对应求和的上限

So $Q_M = [q_{M-1,M}(t) \dots q_{0,M}(t)]$

$$\gamma(t) = Q_M P_{i,M-1} = T_{m-1} S_M P_{i,M-1};$$

hence, $Q_M = T_{M-1} S_M$. By substituting in (3) we get

$$q_{j,M}(t) = \frac{1}{(M-1)!} \sum_{i=0}^{j} (-1)^i \binom{M}{i} (t+j-i)^{M-1}. \tag{4}$$

It is convenient to decompose $Q_M = T^* A$ where

$$T^* = \left[(t+(M-1))^{M-1}\ (t+(M-2))^{M-1} \dots (t+1)^{M-1}\ t^{M-1}\right]$$

and we must determine A. Then we will have

$$Q_m = T^* A = T_{M-1} S_M = T_{M-1} C A$$

for some matrix C which we must also find. But $S_M = CA$. We may find A by reading the columns from (4). The first column corresponds to $j = m-1$ in (4).

$$A = \frac{1}{(M-1)!} \begin{bmatrix} (-1)^0\binom{M}{0} & 0 & \dots & 0 & 0 \\ (-1)^1\binom{M}{1} & (-1)^0\binom{M}{0} & \dots & 0 & 0 \\ (-1)^2\binom{M}{2} & (-1)^1\binom{M}{1} & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ (-1)^{M-2}\binom{M}{M-2} & (-1)^{M-3}\binom{M}{M-3} & \dots & (-1)^0\binom{M}{0} & 0 \\ (-1)^{M-1}\binom{M}{M-1} & (-1)^{M-2}\binom{M}{M-2} & \dots & (-1)^1\binom{M}{1} & (-1)^0\binom{M}{0} \end{bmatrix}$$

Now

$$T^* = \left[(t+k)^{M-1}\right]_{k=M-1,M-2,\dots,1,0} = \left[\sum_{i=0}^{M-1} \binom{M-1}{i} t^{M-1-i} k^i\right]$$

将T*拆开，T_{M-1}定义如下

$$= T_{M-1}\begin{bmatrix} \binom{M-1}{0}(M-1)^0 & \binom{M-1}{0}(M-2)^0 & \cdots & \binom{M-1}{0}1^0 & \binom{M-1}{0} \\ \binom{M-1}{1}(M-1)^1 & \binom{M-1}{1}(M-2)^1 & \cdots & \binom{M-1}{1}1^1 & 0 \\ \vdots & \vdots & & \vdots & \\ \binom{M-1}{M-2}(M-1)^{M-2} & \binom{M-1}{M-2}(M-2)^{M-2} & \cdots & \binom{M-1}{M-2}1^{M-2} & 0 \\ \binom{M-1}{M-1}(M-1)^{M-1} & \binom{M-1}{M-1}(M-2)^{M-1} & \cdots & \binom{M-1}{M-1}1^{M-1} & 0 \end{bmatrix}$$

$$= T_{M-1}C$$

So $C=(c_{ij})$ and $c_{ij}=\binom{M-1}{i}(M-(j+1))^i$; $i, j=0, 1,\ldots,M-1$

$$A=(a_{ij}) \text{ and } a_{ij}=\begin{cases} 0 \text{ for} & i<j \quad \text{条件} \\ (-1)^{i-j}\binom{M}{i-j}\dfrac{1}{(M-1)!} & i\geqslant j \end{cases}$$

$i,j=0,1,\ldots,M-1$

$$s_{ij} = \sum_{k=0}^{M-1} c_{ik} a_{kj} = \sum_{k=0}^{M-1} \binom{M-1}{i}(M-(k+1))^{i}(-1)^{k-j} \cdot \binom{M}{k-j}\frac{1}{(M-1)!}$$

and $S_M = CA$ so

$$s_{ij}=\frac{1}{(M-1)!}\binom{M-1}{i}\sum_{k=j}^{M-1}(M-(k+1))^i(-1)^{k-j}\binom{M}{k-j}$$

4. Uniform Open Cubic B-Spline Curves

We wish to determine matrices R_i such that

$$\gamma_j(t)=T_3R_jP_{j,3}, \qquad 0\leqslant t\leqslant 1$$

where

$$\gamma(t+j)=\gamma_j(t),$$

$$P_{j,3}=\begin{bmatrix} P_j \\ P_{j+1} \\ P_{j+2} \\ P_{j+3} \end{bmatrix}$$

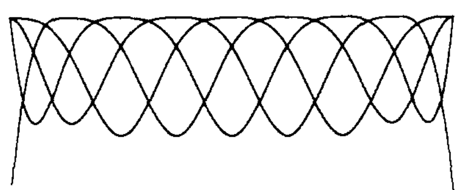

Fig. 1.

and $T_3=[t^3t^2t\ 1]$. We here assume that $M\geqslant 7$ and avoid smaller polygon special cases. Let the knots be $\{x_i\}$, where $x_0=x_1=x_2=x_3=0$; $x_j=j-3$; $j=4,\ldots,m$, $x_{m+1}=x_{m+2}=x_{m+3}=x_{m+4}=m-2$. The polygon has $m+1$ points and m sides. The basis functions have mirror symmetries at the ends which are reflected in the matrices. The curves r_{ij} are subscripted so that

i refers to the "span" or corresponding matrix, and

j indicates multiplication of the $(i+j)$-th vertex.

Clearly the fourth span, R_3 has all standard uniform periodic basis functions as do all successive spans until the $(m-6)$th span. Thus, from section 3,

$$R_i=S_4, \qquad i=3,4,\ldots,(M-6)$$

Next observe that in the third span, r_{21}, r_{22}, and r_{23} are spans of the uniform periodic B-splines $N_{3,4}$, $N_{4,4}$ and $N_{5,4}$, respectively. Since

$$\sum_{j=0}^{3} r_{2j}(x)=1,$$

$r_{20}(x)$ must have the function values of the fourth nonzero standard uniform periodic basis function. Thus,

$R_2 = S_4$

Analogously

$R_{M-5} = S_4$.

We now consider R_1 and R_0. r_{10} and r_{11} are nonstandard and must be computed via the deBoor–Cox algorithm. In R_0, r_{03} is standard and r_{00} must take a simple form. Since r_{00} has a triple zero at $t = 1$ and $r_{00}(0) = 1$, we derive $r_{00}(x) = (1-x)^3$. Thus r_{01} and r_{02} must be derived. Now since r_{01} and r_{10} are two spans of $N_{1,4}$ and r_{02} and r_{11} are two spans of $N_{2,4}$, this will simplify calculations. The deBoor–Cox algorithm yields

$$N_{1,4}(x) = \begin{cases} 3x - \frac{9}{2}x^2 + \frac{7}{4}x^3 & 0 \leq x \leq 1 \\ \frac{1}{4}(2-x)^3 & 1 \leq x \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{2,4}(x) = \begin{cases} \frac{3}{2}x^2 - \frac{11}{12}x^3 & 0 \leq x \leq 1 \\ \frac{7}{12}x^3 - 3x^2 + \frac{9}{2}x - \frac{3}{2} & 1 \leq x \leq 2 \\ (3-x)^3/6 & 2 \leq x \leq 3 \end{cases}$$

We have directly that $r_{0,1}(t) = (N)_{1,4}(t)$ and $r_{02}(t) = N_{2,4}(t)$.
Filling in the matrix yields

$$R_0 = \begin{bmatrix} -1 & 7/4 & -11/12 & 1/6 \\ 3 & -9/2 & 3/2 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

This completes our calculation of R_0.

For R_1 we must translate to the unit interval so

$r_{10}(t) = N_{1,4}(t+1), \qquad r_{11}(t) = N_{2,4}(t+1)$

After carrying out the arithmetic we find:

$$R_1 = \begin{bmatrix} -1/4 & 7/12 & -1/2 & 1/6 \\ 3/4 & -5/4 & 1/2 & 0 \\ -3/4 & 1/4 & 1/2 & 0 \\ 1/4 & 7/12 & 1/6 & 0 \end{bmatrix}$$

To complete this section we must find R_{m-4} and R_{m-3}. Consider R_{m-3}. $r_{(m-3)0}$ is a standard span and $r_{(m-3)3}$ must take a simple form, having a triple zero at 0, and value 1 at 1. So

$r_{(m-3)3}(t) = t^3$.

To find the other functions we note symmetries that yield:

$r_{(m-3)1}(t) = r_{02}(1-t) = \frac{11}{12}t^3 - \frac{5}{4}t^2 - \frac{1}{4}t + \frac{7}{12}$,

$r_{(m-3)2}(t) = r_{01}(1-t) = -\frac{7}{4}t^3 + \frac{3}{4}t^2 - \frac{3}{4}t + \frac{1}{4}$.

So

$$R_{m-3} = \begin{bmatrix} -1/6 & 11/12 & -7/4 & 1 \\ 1/2 & -5/4 & 3/4 & 0 \\ -1/2 & -1/4 & 3/4 & 0 \\ 1/6 & 7/12 & 1/4 & 0 \end{bmatrix}$$

For R_{m-4} we have $r_{(m-4)0}$ and $r_{(m-4)1}$ are standard spans and

$r_{(m-4)2}(t) = r_{11}(1-t) = N_{2,4}((1-t)+1) = -\frac{7}{12}t^3 + \frac{1}{2}t^2 + \frac{1}{2}t + \frac{1}{6}$

$r_{(m-4)3}(t) = r_{10}(1-t) = N_{1,4}((1-t)+1) = \frac{1}{4}t^3$

so

$$R_{m-4} = \begin{bmatrix} -1/6 & 1/2 & -7/12 & 1/4 \\ 1/2 & -1 & 1/2 & 0 \\ -1/2 & 0 & 1/2 & 0 \\ 1/6 & 2/3 & 1/6 & 0 \end{bmatrix}$$

Thus all nonstandard as well as standard basis functions are determined.

5. Conclusion

The matrix forms for curve generation were largely promoted by Coons in [3] and in his earlier work. These formulations are very compact to write, simple to program, and clear to understand. They manifest the desired basis as a matrix transformation of the common power basis. Furthermore, this implementation can be made extremely fast if appropriate matrix facilities are available in either hardware or software. In this paper we have shown how to derive the matrix formulations for arbitrary degree by approaching the task symbolically rather than numerically which would only give the answer for one specific degree. It is hoped that these expressions will result in considerable savings of time and effort for someone wishing to experiment with these schemes. It is recognized that actual evaluation times for these functions in this form are probably not optimal, although those measurements depend on the nature of the computing engine involved. The matrix forms are well-suited for an array processor, for example.

References

[1] Brian A. Barsky, "End Conditions and Boundary Conditions for Uniform B-spline Curve and Surface Representations," Computers in Industry 3 (1982), 17–29.

[2] P. Bezier, Numerical Control-Mathematics and Applications. John Wiley and Sons, London (1972).

[3] S.A. Coons, "Surface Patches and B-spline Curves," Computer Aided Geometric Design, edited by R. Barnhill and R.F. Riesenfeld, Academic Press, Inc. (1974).

[4] W.J. Gordon and R.F. Riesenfeld, "B-spline Curves and Surfaces," Computer Aided Geometric Design, edited by R. Barnhill and R.F. Riesenfeld, Academic Press, Inc. (1974).

[5] F. Yamaguchi, "A New Curve Fitting Method Using a CRT Computer Display," Computer Graphics and Image Processing, Vol 7, No 3 (1978).