General Matrix Representations for B-Splines*

Kaihuai Qin
Dept. of Computer Science & Technology
Tsinghua University
Beijing 100084, CHINA
E-mail: qkh-dcs@mail.tsinghua.edu.cn

Abstract

In this paper, the concept of basis matrix of B-splines is presented. A general matrix representation, which results in an explicitly recursive matrix formula, for nonuniform B-spline curves of an arbitrary degree is also presented by means of Toeplitz matrix. New recursive matrix representations for uniform B-spline curves and Bezier ones of an arbitrary degree are obtained as special cases of that for nonuniform B-spline curves. The recursive formula for basis matrix can be substituted for de Boor-Cox's one for B-splines, and it has better time complexity than de Boor-Cox's formula when used for conversion and computation of B-spline curves and surfaces between different CAD systems. Finally, some applications of the matrix representations are given in the paper.

1. Introduction

Matrix theory and its algorithms are very useful in computer-aided geometric design, since matrix is an important and basic tool in mathmatics. Matrix formulae of B-spline curves and surfaces have advantages of both simple computation of points on curves or surfaces and their derivatives, and easy analysis of the geometric properties of B-spline curves and surfaces. In 1982, Chang [1] gave the matrix formula of Bezier curves. Cohen and Riesenfeld [5] gave those for not only Bezier curves but also uniform B-splines of an arbitrary degree in 1982. In 1990, Choi, Yoo and Lee[2] proposed a procedure to symbolically evaluate a matrix for a B-spline curve using Boehm's knot-insertion algorithm [14]. Grabowski and Li [3] in 1992, Wang, Sun and Qin [4] in 1993 got the matrix by the analogously approaches instead of an explicit formula, respectively. So far a general matrix formula for nonuniform B-spline curves has not been found, although the matrix representation can be gotten by algorithms.

In this paper, recursive matrix formulae for B-splines and Bezier curves are presented, and some applications are given in the paper, too.

The organization of this paper is as follows: Section 2 describes how to represent de Boor-Cox formula for B-splines using Toeplitz matrix[7]. In the 3rd section, general matrices for B-spline curves are proposed. New recursive matrix formulae for representing uniform B-splines and Bezier curves are obtained as special cases of the basis matrix formula of nonuniform B-spline curves in Section 4. Some applications are shown in Section 5. Finally conclusions are given in Section 6.

2. Representing de Boor-Cox formula by Toeplitz matrix

First of all, let us look at how to represent a polynomial and a product of two polynomials using Toeplitz matrix[7]. Then Toeplitz matrix representation of de Boor-Cox formula for B-splines will be introduced.

2.1 Toeplitz matrix

The Toeplitz matrix is one whose elements on any line parallel to the main diagonal are all equal. A band Toeplitz matrix is defined as follows:

$$\mathbf{T}=\begin{bmatrix} a_0 & a_1 & \cdots & a_s & & 0 \\ a_{-1} & a_0 & \ddots & & a_s & \\ \vdots & & & & & \ddots \\ \vdots\ \cdots & \ddots & \ddots & \ddots & \cdots & a_s \\ & & & & & \vdots \\ a_{-r} & \cdots & \ddots & \ddots & \ddots & \\ & \ddots & & & & a_1 \\ 0 & a_{-r} & \cdots & & a_{-1} & a_0 \end{bmatrix}.$$

* Project supported by the NSF of China(Grant No. 69473010).

For example, a special Toeplitz matrix, or a lower triangular matrix

$$\mathbf{T} = \begin{bmatrix} a_0 & & & & & 0 \\ a_1 & a_0 & & & & \\ \vdots & \ddots & \ddots & & & \\ a_{n-1} & \cdots & \ddots & \ddots & & \\ & \ddots & \cdots & \ddots & \ddots & \\ 0 & & a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix}$$

can be constructed by the coefficients of a polynomial $f(x) = a_0+a_1x+a_2x^2+...+a_{n-1}x^{n-1}$ ($a_{n-1} \neq 0$).

2.2 Representing the product of two polynomials using Toeplitz matrix

Let $g(x) = c_0+c_1x+c_2x^2+...+c_{m-1}x^{m-1}$ ($c_{m-1} \neq 0$),
$q(x) = d_0+d_1x+d_2x^2+...+d_{n-1}x^{n-1}$ ($d_{n-1} \neq 0$).

One can obtain

$f(x) = g(x)q(x)$

$$= \mathbf{X}\begin{bmatrix} c_0 & & & & & & 0 \\ c_1 & c_0 & & & & & \\ \vdots & \ddots & \ddots & & & & \\ c_{m-1} & \cdots & \ddots & \ddots & & & \\ & \ddots & \cdots & \ddots & \ddots & & \\ & & c_{m-1} & \cdots & \ddots & c_0 & \\ 0 & & & c_{m-1} & \cdots & c_1 & c_0 \end{bmatrix}\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \mathbf{X}\begin{bmatrix} c_0 & & & 0 \\ c_1 & c_0 & & \\ \vdots & \ddots & \ddots & \\ \vdots & \cdots & \ddots & c_0 \\ \vdots & \cdots & \cdots & \vdots \\ c_{m-1} & c_{m-2} & \cdots & c_{m-n} \\ & c_{m-1} & \ddots & \vdots \\ & & \ddots & c_{m-2} \\ 0 & & & c_{m-1} \end{bmatrix}\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \end{bmatrix}$$

where $\mathbf{X} = \begin{bmatrix} 1 & x & x^2 & \cdots & x^{m+n-2} \end{bmatrix}$. The above is the Toeplitz matrix representation of the product of two polynomials.

2.3 Representing de Boor-Cox formula using Toeplitz matrix

The B-spline first introduced by Schoenberg[9] has been used in various fields such as computer-aided design, computer graphics, numerical analysis and so on. The normalized local support B-spline basis function of degree k-1 is defined by the following de Boor- Cox recursive formula[6,10]:

$$\begin{cases} B_{j,k}(t) = \dfrac{t-t_j}{t_{j+k-1}-t_j} B_{j,k-1}(t) + \dfrac{t_{j+k}-t}{t_{j+k}-t_{j+1}} B_{j+1,k-1}(t); \\ B_{i,1}(t) = \begin{cases} 1, & t \in [t_i, t_{i+1}); \\ 0, & t \notin [t_i, t_{i+1}) \end{cases} \end{cases} \quad (1)$$

with the convention $0/0=0$. By means of basis translation from B-spline to power basis[11,12] $B_{j,k-1}(u)$ can be represented as follows:

$$B_{j,k-1}(u) = \begin{bmatrix} 1 & t & t^2 & \cdots & t^{k-2} \end{bmatrix}\begin{bmatrix} N_{0,j}^{k-1} \\ N_{1,j}^{k-1} \\ \vdots \\ N_{k-2,j}^{k-1} \end{bmatrix},$$

Thus, Eq. (1) can be rewritten by Toeplitz matrix:

$$B_{j,k}(u) = \begin{bmatrix} 1 & u & u^2 & \cdots & u^{k-1} \end{bmatrix}\left(\begin{bmatrix} N_{0,j}^{k-1} & 0 \\ N_{1,j}^{k-1} & N_{0,j}^{k-1} \\ \vdots & N_{1,j}^{k-1} \\ N_{k-2,j}^{k-1} & \vdots \\ 0 & N_{k-2,j}^{k-1} \end{bmatrix}\begin{bmatrix} d_{0,j} \\ d_{1,j} \end{bmatrix} + \begin{bmatrix} N_{0,j+1}^{k-1} & 0 \\ N_{1,j+1}^{k-1} & N_{0,j+1}^{k-1} \\ \vdots & N_{1,j+1}^{k-1} \\ N_{k-2,j+1}^{k-1} & \vdots \\ 0 & N_{k-2,j+1}^{k-1} \end{bmatrix}\begin{bmatrix} h_{0,j} \\ h_{1,j} \end{bmatrix}\right),$$

$$u = \frac{t-t_i}{t_{i+1}-t_i}, u \in [0,1) \quad (2)$$

where

$$d_{0,j} = \frac{t_i - t_j}{t_{j+k-1}-t_j}, d_{1,j} = \frac{t_{i+1}-t_i}{t_{j+k-1}-t_j},$$

$$h_{0,j} = \frac{t_{j+k}-t_i}{t_{j+k}-t_{j+1}}, h_{1,j} = -\frac{t_{i+1}-t_i}{t_{j+k}-t_{j+1}}$$

with the convention $0/0=0$.

3. General matrices for nonuniform B-spline curves and surfaces

B-spline basis functions $B_{j,k}(t)$ are piecewise polynomials of degree k-1. If $t \in [t_i, t_{i+1})$, $t_i < t_{i+1}$, there are k B-spline basis functions of degree k-1 that are nonzero: $B_{j,k}(t)$, $j=(i-k+1),(i-k+2),...,i$. They can be represented in a matrix equation as follows:

$$\begin{aligned} &\begin{bmatrix} B_{i-k+1,k}(u) & B_{i-k+2,k}(u) & \cdots & B_{i,k}(u) \end{bmatrix} \\ &= \begin{bmatrix} 1 & u & u^2 & \cdots & u^{k-1} \end{bmatrix}\mathbf{M}^k(i), \\ &\quad u = (t-t_i)/(t_{i+1}-t_i), \quad u \in [0,1), \end{aligned} \quad (3)$$

where

$$\mathbf{M}^k(i)=\begin{bmatrix} N^k_{0,i-k+1} & N^k_{0,i-k+2} & \cdots & N^k_{0,i}\\ N^k_{1,i-k+1} & N^k_{1,i-k+2} & \cdots & N^k_{1,i}\\ \vdots & \vdots & \cdots & \vdots \\ N^k_{k-1,i-k+1} & N^k_{k-1,i-k+2} & \cdots & N^k_{k-1,i}\end{bmatrix}.$$

t_j are the knots.

Let $\boldsymbol{V}_j$ be the control vertices of a B-spline curve. The B-spline curve segment

$$\mathbf{c}_{i-k+1}(u)=\begin{bmatrix} B_{i-k+1,k}(u) & B_{i-k+2,k}(u) & \cdots & B_{i,k}(u)\end{bmatrix}\begin{bmatrix}\mathbf{V}_{i-k+1}\\ \mathbf{V}_{i-k+2}\\ \vdots \\ \mathbf{V}_i\end{bmatrix}$$

$$=\begin{bmatrix}1 & u & u^2 & \cdots & u^{k-1}\end{bmatrix}\mathbf{M}^k(i)\begin{bmatrix}\mathbf{V}_{i-k+1}\\ \mathbf{V}_{i-k+2}\\ \vdots \\ \mathbf{V}_i\end{bmatrix}, \qquad (4)$$

where $u=(t-t_i)/(t_{i+1}-t_i)$, $u\in[0,1)$. $\mathbf{M}^k(i)$ is referred to as the ith ***basis matrix*** of B-spline basis functions of degree k-1.

3.1 Recursive formula for basis matrices of B-splines of degree *k-1*

Theorem 1 The ith basis matrix $\boldsymbol{M}^k(i)$ of B-spline basis functions of degree k-1 can be obtained by a recursive equation as follows:

$$\begin{cases}\mathbf{M}^k(i)=\begin{bmatrix}\mathbf{M}^{k-1}(i)\\ \mathbf{0}\end{bmatrix}\begin{bmatrix}1-d_{0,i-k+2} & d_{0,i-k+2} & & & 0\\ & 1-d_{0,i-k+3} & d_{0,i-k+3} & & \\ & & \ddots & \ddots & \\ 0 & & & 1-d_{0,i} & d_{0,i}\end{bmatrix}+\\ \qquad\begin{bmatrix}\mathbf{0}\\ \mathbf{M}^{k-1}(i)\end{bmatrix}\begin{bmatrix}-d_{1,i-k+2} & d_{1,i-k+2} & & & 0\\ & -d_{1,i-k+3} & d_{1,i-k+3} & & \\ & & \ddots & \ddots & \\ 0 & & & -d_{1,i} & d_{1,i}\end{bmatrix},\\ \mathbf{M}^1(i)=[1]\end{cases} \qquad (5)$$

if $u=(t-t_i)/(t_{i+1}-t_i)$, $u\in[0,1)$.

Proof. Substituting Eq. (2) into Eq.(3) yields:

$$M^k(i)=\begin{bmatrix}N^{k-1}_{0,i-k+1} & 0\\ N^{k-1}_{1,i-k+1} & N^{k-1}_{0,i-k+1}\\ \vdots & N^{k-1}_{1,i-k+1}\\ N^{k-1}_{k-2,i-k+1} & \vdots\\ 0 & N^{k-1}_{k-2,i-k+1}\end{bmatrix}\begin{bmatrix}d_{0,i-k+1} & \overbrace{0 \cdots 0}^{k-1}\\ d_{1,i-k+1} & 0 \cdots 0\end{bmatrix}+$$

$$\begin{bmatrix}N^{k-1}_{0,i-k+2} & 0\\ N^{k-1}_{1,i-k+2} & N^{k-1}_{0,i-k+2}\\ \vdots & N^{k-1}_{1,i-k+2}\\ N^{k-1}_{k-2,i-k+2} & \vdots\\ 0 & N^{k-1}_{k-2,i-k+2}\end{bmatrix}\begin{bmatrix}0 & d_{0,i-k+2} & \overbrace{0 \cdots 0}^{k-2}\\ 0 & d_{1,i-k+2} & 0 \cdots 0\end{bmatrix}+$$

$$+\cdots+\begin{bmatrix}N^{k-1}_{0,i} & 0\\ N^{k-1}_{1,i} & N^{k-1}_{0,i}\\ \vdots & N^{k-1}_{1,i}\\ N^{k-1}_{k-2,i} & \vdots\\ 0 & N^{k-1}_{k-2,i}\end{bmatrix}\begin{bmatrix}\overbrace{0 \cdots 0}^{k-1} & d_{0,i}\\ 0 \cdots 0 & d_{1,i}\end{bmatrix}+$$

$$\begin{bmatrix}N^{k-1}_{0,i-k+2} & 0\\ N^{k-1}_{1,i-k+2} & N^{k-1}_{0,i-k+2}\\ \vdots & N^{k-1}_{1,i-k+2}\\ N^{k-1}_{k-2,i-k+2} & \vdots\\ 0 & N^{k-1}_{k-2,i-k+2}\end{bmatrix}\begin{bmatrix}h_{0,i-k+1} & \overbrace{0 \cdots 0}^{k-1}\\ h_{1,i-k+1} & 0 \cdots 0\end{bmatrix}+$$

$$\begin{bmatrix}N^{k-1}_{0,i-k+3} & 0\\ N^{k-1}_{1,i-k+3} & N^{k-1}_{0,i-k+3}\\ \vdots & N^{k-1}_{1,i-k+3}\\ N^{k-1}_{k-2,i-k+3} & \vdots\\ 0 & N^{k-1}_{k-2,i-k+3}\end{bmatrix}\begin{bmatrix}0 & h_{0,i-k+2} & \overbrace{0 \cdots 0}^{k-2}\\ 0 & h_{1,i-k+2} & 0 \cdots 0\end{bmatrix}+$$

$$+\cdots+\begin{bmatrix}N^{k-1}_{0,i+1} & 0\\ N^{k-1}_{1,i+1} & N^{k-1}_{0,i+1}\\ \vdots & N^{k-1}_{1,i+1}\\ N^{k-1}_{k-2,i+1} & \vdots\\ 0 & N^{k-1}_{k-2,i+1}\end{bmatrix}\begin{bmatrix}\overbrace{0 \cdots 0}^{k-1} & h_{0,i}\\ 0 \cdots 0 & h_{1,i}\end{bmatrix}$$

$$=\begin{bmatrix}N^{k-1}_{0,i-k+2} & 0\\ N^{k-1}_{1,i-k+2} & N^{k-1}_{0,i-k+2}\\ \vdots & N^{k-1}_{1,i-k+2}\\ N^{k-1}_{k-2,i-k+2} & \vdots\\ 0 & N^{k-1}_{k-2,i-k+2}\end{bmatrix}\begin{bmatrix}h_{0,i-k+1} & d_{0,i-k+2} & \overbrace{0 \cdots 0}^{k-2}\\ h_{1,i-k+1} & d_{1,i-k+2} & 0 \cdots 0\end{bmatrix}+$$

$$\begin{bmatrix}N^{k-1}_{0,i-k+3} & 0\\ N^{k-1}_{1,i-k+3} & N^{k-1}_{0,i-k+3}\\ \vdots & N^{k-1}_{1,i-k+3}\\ N^{k-1}_{k-2,i-k+3} & \vdots\\ 0 & N^{k-1}_{k-2,i-k+3}\end{bmatrix}\begin{bmatrix}0 & h_{0,i-k+2} & d_{0,i-k+3} & \overbrace{0 \cdots 0}^{k-3}\\ 0 & h_{1,i-k+2} & d_{1,i-k+3} & 0 \cdots 0\end{bmatrix}+$$

$$+\cdots+\begin{bmatrix} N_{0,i}^{k-1} & 0 \\ N_{1,i}^{k-1} & N_{0,i}^{k-1} \\ \vdots & N_{1,i}^{k-1} \\ N_{k-2,i}^{k-1} & \vdots \\ 0 & N_{k-2,i}^{k-1} \end{bmatrix}\begin{bmatrix} \overbrace{0 \;\cdots\; 0}^{k-2} & h_{0,i-1} & d_{0,i} \\ 0 \;\cdots\; 0 & h_{1,i-1} & d_{1,i} \end{bmatrix}$$

$$=\begin{bmatrix} \mathbf{M}^{k-1}(i) \\ \mathbf{0} \end{bmatrix}\begin{bmatrix} h_{0,i-k+1} & d_{0,i-k+2} & & & 0 \\ & h_{0,i-k+2} & d_{0,i-k+3} & & \\ & & \ddots & \ddots & \\ 0 & & & h_{0,i-1} & d_{0,i} \end{bmatrix}+$$

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{k-1}(i) \end{bmatrix}\begin{bmatrix} h_{1,i-k+1} & d_{1,i-k+2} & & & 0 \\ & h_{1,i-k+2} & d_{1,i-k+3} & & \\ & & \ddots & \ddots & \\ 0 & & & h_{1,i-1} & d_{1,i} \end{bmatrix}.$$

Notice that $h_{1,j-1} = -d_{1,j}$, $h_{0,j-1} = 1-d_{0,j}$, and $\mathbf{M}^1(i)=[1]$. Thus, Eq. (5) holds.

Eq. (5) can be regarded as a recursive definition of basis matrices. It can be used for both analysis of properties of NURBS curves and surfaces, and numerical or symbolic computation of NURBS curves and surfaces.

3.2 Examples of symbolic computation

Using Eq. (5) for the symbolic computation, when $u=(t-t_i)/(t_{i+1}-t_i) \in [0,\ 1)$ one can easily obtain the following basis matrices:

$\mathbf{M}^1(i) = [1]$,

$$\mathbf{M}^2(i)=\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix},$$

$$\mathbf{M}^3(i)=\begin{bmatrix} \dfrac{t_{i+1}-t_i}{t_{i+1}-t_{i-1}} & \dfrac{t_i-t_{i-1}}{t_{i+1}-t_{i-1}} & 0 \\ \dfrac{-2(t_{i+1}-t_i)}{t_{i+1}-t_{i-1}} & \dfrac{2(t_{i+1}-t_i)}{t_{i+1}-t_{i-1}} & 0 \\ \dfrac{t_{i+1}-t_i}{t_{i+1}-t_{i-1}} & -(t_{i+1}-t_i)\left(\dfrac{1}{t_{i+1}-t_{i-1}}+\dfrac{1}{t_{i+2}-t_i}\right) & \dfrac{t_{i+1}-t_i}{t_{i+2}-t_{i-1}} \end{bmatrix},$$

$$\mathbf{M}^4(i)=\begin{bmatrix} \dfrac{(t_{i+1}-t_i)^2}{(t_{i+1}-t_{i-1})(t_{i+1}-t_{i-2})} & 1-m_{0,0}-m_{0,2} & \dfrac{(t_i-t_{i-1})^2}{(t_{i+2}-t_{i-1})(t_{i+1}-t_{i-1})} & 0 \\ -3m_{0,0} & 3m_{0,0}-m_{1,2} & \dfrac{3(t_{i+1}-t_i)(t_i-t_{i-1})}{(t_{i+2}-t_{i-1})(t_{i+1}-t_{i-1})} & 0 \\ 3m_{0,0} & -3m_{0,0}-m_{2,2} & \dfrac{3(t_{i+1}-t_i)^2}{(t_{i+2}-t_{i-1})(t_{i+1}-t_{i-1})} & 0 \\ -m_{0,0} & m_{0,0}-m_{3,2}-m_{3,3} & m_{3,2} & \dfrac{(t_{i+1}-t_i)^2}{(t_{i+3}-t_i)(t_{i+2}-t_i)} \end{bmatrix}$$

where

$m_{3,2} = -m_{2,2}/3 - m_{3,3} - (t_{i+1} - t_i)^2/[(t_{i+2} - t_i)(t_{i+2} - t_{i-1})]$,

$m_{i,j}$ = element in row i, column j .

4. Special cases of basis matrices

It is well known that a B-spline curve with the knots, between which the spacing is equal, is referred to as a uniform B-spline curve; and a B-spline curve with the knot vector, $\left\{\overbrace{a_0,\cdots,a_0}^{k},\ \overbrace{a_1,\cdots,a_1}^{k}\right\}$ $(a_0 < a_1)$, is regarded as a Bezier curve. Analogously, the basis matrices for uniform B-spline curves and Bezier curves can be obtained by Eq. (5) using the corresponding knot vectors.

4.1 Basis matrices of uniform B-splines

For uniform B-spline curves and surfaces, the spacing between the knots is equal, say 1. Thus, one has

$$\begin{cases} d_{0,j} = \dfrac{i-j}{k-1}, \\ d_{1,j} = \dfrac{1}{k-1}. \end{cases} \qquad (6)$$

Instituting Eq. (6) into Eq. (5), one can get:

Theorem 2 The basis matrices of uniform B-splines can be represented by the following recursive formula:

$$\begin{cases} \mathbf{M}^k = \dfrac{1}{k-1}\left(\begin{bmatrix} \mathbf{M}^{k-1} \\ \mathbf{0} \end{bmatrix}\begin{bmatrix} 1 & k-2 & & & 0 \\ & 2 & k-3 & & \\ & & \ddots & \ddots & \\ 0 & & & k-1 & 0 \end{bmatrix}+\begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{k-1} \end{bmatrix}\begin{bmatrix} -1 & 1 & & & 0 \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ 0 & & & -1 & 1 \end{bmatrix}\right), \\ \mathbf{M}^1 = [1] \end{cases} \qquad (7)$$

if $u=(t-t_i)/(t_{i+1}-t_i)$, $u \in [0,1)$.

Unlike the basis matrices of nonuniform B-splines, the basis matrices of uniform B-splines of degree $k-1$ are independent of t_i.

Using equation (7) recursively step by step, one can also obtain the following matrix:

$$\mathbf{M}^k = \begin{bmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,k-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,k-1} \\ \vdots & \vdots & \cdots & \vdots \\ m_{k-1,0} & m_{k-1,1} & \cdots & m_{k-1,k-1} \end{bmatrix}, \quad (8)$$

where

$$m_{i,j} = \frac{1}{(k-1)!} C_{k-1}^{k-1-i} \sum_{s=j}^{k-1} (-1)^{s-j} C_k^{s-j} (k-s-1)^{k-1-i},$$

$$C_n^i = \frac{n!}{i!(n-i)!}.$$

Both Eqs. (7) and (8) can be used to calculate the basis matrices of uniform B-splines. Several examples of basis matrices for uniform B-splines are given as follows:

$$\mathbf{M}^1 = [1],$$

$$\mathbf{M}^2 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix},$$

$$\mathbf{M}^3 = \frac{1}{2!}\begin{bmatrix} 1 & 1 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix},$$

$$\mathbf{M}^4 = \frac{1}{3!}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix},$$

$$\mathbf{M}^5 = \frac{1}{4!}\begin{bmatrix} 1 & 11 & 11 & 1 & 0 \\ -4 & -12 & 12 & 4 & 0 \\ 6 & -6 & -6 & 6 & 0 \\ -4 & 12 & -12 & 4 & 0 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}.$$

4.2 Basis matrices for Bezier curves

Suppose that the knot vector of a B-spline curve of degree *k-1* is as follows:

$$\left\{ \overbrace{0, \cdots, 0}^{k}, \quad \overbrace{1, \cdots, 1}^{k} \right\}.$$

Thus,

$$\begin{cases} d_{0,j} = 0, \\ d_{1,j} = 1. \end{cases} \quad (9)$$

Substituting Eq.(9) into Eq. (5) , one can obtain:

Theorem 3 The basis matrices of Bezier curves can be represented by the following recursive formula:

$$\begin{cases} \mathbf{M}^k = \begin{bmatrix} \mathbf{M}^{k-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{k-1} \end{bmatrix} \begin{bmatrix} -1 & 1 & & 0 \\ & -1 & 1 & \\ & & \ddots & \ddots \\ 0 & & -1 & 1 \end{bmatrix}, \\ \mathbf{M}^1 = [1] \end{cases} \quad (10)$$

if $u = (t - t_i)/(t_{i+1} - t_i)$, $u \in [0,1)$.

Like the basis matrices of uniform B-splines, the basis matrices of Bezier curves of degree *k-1* are independent of t_i.

Using Eq. (10) recursively step by step, one can also obtain the following matrix for Bezier curves easily:

$$\mathbf{M}^k = \begin{bmatrix} m_{0,0} & & & 0 \\ m_{1,0} & m_{1,1} & & \\ \vdots & \vdots & \ddots & \\ m_{k-1,0} & m_{k-1,1} & \cdots & m_{k-1,k-1} \end{bmatrix}, \quad (11)$$

where

$$m_{i,j} = \begin{cases} (-1)^{i-j} C_{k-1}^{j} C_{k-1-j}^{i-j}, & i \ge j; \\ 0, & i < j. \end{cases}$$

$\boldsymbol{M}^k$ is a lower triangular $n \times n$ matrix.

Both Eqs. (10) and (11) can be used to calculate the basis matrices of Bezier curves. Several examples of the basis matrices for Bezier curves are given as follows:

$$\mathbf{M}^1 = [1],$$

$$\mathbf{M}^2 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix},$$

$$\mathbf{M}^3 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix},$$

$$\mathbf{M}^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix},$$

$$\mathbf{M}^5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 12 & -12 & 4 & 0 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix},$$

$$\mathbf{M}^6 = \begin{bmatrix} 1 & & & & & 0 \\ -5 & 5 & & & & \\ 10 & -20 & 10 & & & \\ -10 & 30 & -30 & 10 & & \\ 5 & -20 & 30 & -20 & 5 & \\ -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix}.$$

5. Applications

There are many applications of the basis matrices in practice. Some of them are given in this section.

5.1 Computation of derivatives of nonuniform B-spline curves

Assume there is a nonuniform B-spline curve of degree *k-1*

$$\mathbf{c}_{i-k+1}(u) = \mathbf{U}^k \mathbf{M}^k(i)\mathbf{V}^k(i),\ i=k\text{-}1,k,\ldots,n;$$
$$u = (t-t_i)/(t_{i+1}-t_i),\quad u\in[0,1) \qquad (12)$$

where

$$\mathbf{U}^k = \begin{bmatrix} 1 & u & u^2 & \cdots & u^{k-1}\end{bmatrix},\ \mathbf{V}^k(i) = \begin{bmatrix} \mathbf{V}_{i-k+1} \\ \mathbf{V}_{i-k+2} \\ \vdots \\ \mathbf{V}_i \end{bmatrix},$$

defined over the knot vector $\mathbf{T}=\{t_0, t_1, \ldots, t_{n+k}\}$, then

$$\frac{d^n}{du^n}\mathbf{c}_{i-k+1}(u) = \left(\frac{d^n}{du^n}\mathbf{U}^k\right)\mathbf{M}^k(i)\mathbf{V}^k(i)$$

where derivatives, or $d^n(\mathbf{U}^k)/du^n$, can be easily computed, for instance, $d\mathbf{U}^k/du = [0 \quad 1 \quad 2u \quad \ldots \quad (k\text{-}1)u^{k-2}]$.

5.2 Computation of derivatives of NURBS curves

For an NURBS curve of degree $k\text{-}1$

$$\mathbf{c}_{i-k+1}(u) = \mathbf{U}^k\mathbf{M}^k(i)\mathbf{P}^k(i)/\mathbf{U}^k\mathbf{M}^k(i)\mathbf{W}^k(i)$$
$$\stackrel{\Delta}{=} \mathbf{R}(u)/S(u),\quad 0\le u=(t-t_i)/(t_{i+1}-t_i)<1.$$

where

$$\mathbf{P}^k(i) = \begin{bmatrix} w_{i-k+1}\mathbf{V}_{i-k+1} \\ w_{i-k+2}\mathbf{V}_{i-k+2} \\ \vdots \\ w_i\mathbf{V}_i \end{bmatrix},\quad \mathbf{W}^k(i) = \begin{bmatrix} w_{i-k+1} \\ w_{i-k+2} \\ \vdots \\ w_i \end{bmatrix}.$$

The derivatives of an NURBS curve with respect to u can be easily obtained since the basis matrixes are independent of u.

$$\frac{d^n}{du^n}\mathbf{c}_{i-k+1}(u) = \sum_{j=0}^{n}\binom{n}{j}\frac{d^j}{du^j}\{\mathbf{R}(u)\}\frac{d^{n-j}}{du^{n-j}}\{1/s(u)\}.$$

Frequently used are the first and the second derivatives of NURBS curves:

$$\frac{d}{du}\mathbf{c}_{i-k+1}(u) = \mathbf{R}(u)\frac{-\frac{d}{du}S(u)}{S(u)^2} + \frac{\frac{d}{du}\mathbf{R}(u)}{S(u)},$$

$$\frac{d^2}{du^2}\mathbf{c}_{i-k+1}(u) = \mathbf{R}(u)\left\{\frac{\frac{d}{du}S(u)}{S(u)^2} - \frac{2\left(\frac{d}{du}S(u)\right)^2}{S(u)^3}\right\} +$$

$$2\mathbf{R}(u)\frac{-\frac{d}{du}S(u)}{S(u)^2} + \frac{\frac{d^2}{du^2}\mathbf{R}(u)}{S(u)}.$$

5.3 Degree raising for nonuniform B-spline curves

Degree raising for nonuniform B-spline curves is a common technique in CAGD. The basis matrix of B-splines can be used for degree raising of B-spline curves.

After its degree is elevated by 1, a segment of B-spline curve of degree $k\text{-}1$ defined by Eq.(12) can be re-written as follows:

$$\mathbf{c}_{i-k+1}(u) = \mathbf{U}^{k+1}\begin{bmatrix}\mathbf{M}^k(i) \\ \mathbf{0}\end{bmatrix}\mathbf{V}^k(i)$$
$$= \mathbf{U}^{k+1}\mathbf{M}^{k+1}(i)\mathbf{V}^{k+1}(i). \qquad (13)$$

Thus, one can obtain the control vertices for the degree-raised curve:

$$\mathbf{V}^{k+1}(i) = \left[\mathbf{M}^{k+1}(i)\right]^{-1}\begin{bmatrix}\mathbf{M}^k(i) \\ \mathbf{0}\end{bmatrix}\mathbf{V}^k(i). \qquad (14)$$

Suppose that there is a B-spline curve of degree $k\text{-}1$ with control vertices $\mathbf{V}_i$, $(i=0, 1, \ldots, n)$ defined over a knot vector

$$\left\{t_0 \quad t_1 \quad \cdots \quad t_{k-1} \quad \overbrace{t_k \cdots t_k}^{s_1} \quad \overbrace{t_{k+1} \cdots t_{k+!}}^{s_2} \quad \cdots \quad \overbrace{t_{k+m-1} \cdots t_{k+m-1}}^{s_m} \quad t_{n+1} \quad t_{n+2} \quad \cdots \quad t_{n+k}\right\}$$

where $s_1 + s_2 + \cdots + s_m = n\text{-}k\text{+}1$. In order for degree raising of the whole curve by 1, the multiplicity of each interior knot has to be added by 1 using a knot-insertion algorithm[8,13,17], so that the knot vector becomes as follows:

$$\left\{t_{-1} \quad t_0 \quad t_1 \quad \cdots \quad t_{k-1} \quad \overbrace{t_k \cdots t_k}^{s_1+1} \quad \overbrace{t_{k+1} \cdots t_{k+!}}^{s_2+1} \quad \cdots \quad \overbrace{t_{k+m-1} \cdots t_{k+m-1}}^{s_m+1} \quad t_{n+1} \quad t_{n+2} \quad \cdots \quad t_{n+k} \quad t_{n+k+1}\right\}.$$

Then, a B-spline curve of degree k can be obtained with degree raising of all the segments of the curve using Eq.(14). This idea for degree raising of B-splines is feasible, but it is less efficient than the elegant method for degree raising of B-splines in [13] and [17].

5.4 Degree reduction of B-spline curves

Degree reduction of B-spline curves is a difficult problem, since generally a B-spline curve of degree k can not be precisely represented by a curve of degree $k\text{-}1$. Of course, it can be approached by a B-spline curve of a less degree. One can obtain its least square solution of this problem for a segment of B-spline curve using Eq.(13)

$$\mathbf{V}^k(i) = \left(\begin{bmatrix}\left[\mathbf{M}^k(i)\right]^T & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{M}^k(i) \\ \mathbf{0}\end{bmatrix}\right)^{-1}\mathbf{M}^{k+1}(i)\mathbf{V}^{k+1}(i).$$

Similarly, one can obtain the least square solution of this problem for a whole B-spline curve if combining knot removal [15,16] (omitted here because of the limited space).

6. Conclusions

By means of the concept of the basis matrix proposed in the paper, the matrix representations of nonuniform and uniform B-splines and Bezier curves can be unified by a recursive formula. It is shown that the matrix representations for uniform B-splines and Bezier curves can be regarded as special cases of the basis matrix of nonuniform B-splines. Like de Boor-Cox recursive definition of B-splines, the basis matrices of B-splines can be defined by Eq.(5), too. With regard to B-spline surfaces, the basis matrices can be used for the surfaces in the same way as B-spline curves. The recursive basis matrix formula, or Eq.(5), can be substituted for the de Boor-Cox recursive function when used for computation of B-spline curves and surfaces.

In fact, the general matrix formula of nonuniform B-splines of an arbitrary degree, or Eq.(5), can be used both for symbolic or numerical computation of NURBS curves and surfaces, and for theoretical analysis of properties of NURBS curves and surfaces. In fact, the recursive basis-matrix formula, or Eq. (5), for nonuniform B-splines of an arbitrary degree is more efficient than the symbolic approach in [2] and the numerical algorithm in [3] in numerical evaluation. Assume that the execution times T satisfy[2] $T_{addition} \cong T_{subtraction}$, $T_{multiplication} \cong T_{division}$ and $T_{multiplication} \cong 1.12\ T_{addition}$. The comparison of the three methods for the basis matrices of nonuniform B-splines of degree *k-1* can be made as shown in Table 1.

Table 1 Comparison of three methods

	An arbitrary order (k)		$k = 4$			
	$T_{multiplication}$	$T_{addition}$	$T_{multiplication}$	$T_{addition}$	Sum total ($\cong T_{addition}$)	%
Choi *et al.* [2]	$8k^2(4^k-4)/3$	$4k^2(4^k-4)/3$	*10752*	*5376*	*17418*	*12013*
Grabowski & Li[3]	$2k^3-3k^2+k$	$(6k^2-9k+3)k/2$	*84*	*126*	*220*	*152*
The new method	$(4k^2-3k-1)k/3$	$(4k^2-3k-1)k/3+1$	*68*	*69*	*145*	*100*

References

[1] G. Chang. Matrix foundation of Bezier technique. *Computer-Aided Design*, 14(6):354-350, 1982.

[2] B. K. Choi, W. S. Yoo & C. S. Lee. Matrix representation for NURB curves and surfaces. *Computer-Aided Design*, 22(4):235-240, 1990.

[3] H. Grabowski & X. Li, Coefficient formula and matrix of nonuniform B-spline functions. *Computer-Aided Design*, 24(12):637-642, 1992.

[4] X. Wang, J. Sun & Kaihuai Qin. Symbolic matrix representation of NURBS and its applications. *Chinese Journal of Computers*, 16(1):29-34, 1993 (in Chinese).

[5] E. Cohen & R. F. Riesenfeld. General matrix representations for Bezier and B-spline curves. *Computers in Industry*, 3:9-15, 1982.

[6] C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6:50-62, 1972.

[7] H. Vidom. Toeplitz matrices, in: I. Hirschmann (ed.), *Studies in Real and Complex Analysis*. MAA Studies in Mathematics 3, 1965.

[8] Kaihuai Qin, Y Guan. Two algorithms for inserting knots into B-spline curves, *Chinese Journal of Computers*, 20(6):557-561, 1997 (in Chinese).

[9] I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Quart. Appl. Math*., 4(45-99):112-141, 1946.

[10] M. G. Cox. The numerical evaluation of B-splines. *J. Inst. Math. & Applic.*, 10:134-149, 1972.

[11] C. de Boor. *Practical Guide to Splines*. Springer-Verlag, 1978.

[12] L. Schumaker. *Spline Functions: Basic Theory*. John Wiley & Sons, New York, 1981.

[13] Kaihuai Qin. A new algorithm for degree-raising of nonuniform B-spline curves. *Chinese Journal of Computers*, 19(7):537 ~542, 1996 (in Chinese).

[14] W. Boehm. Inserting new knots into B-spline curves. *Computer-Aided Design*, 12(4):199-201, 1980.

[15] M. Eck & J. Hadenfeld. Knot removal for B-spline curves. *CAGD*, 12(3):259-262, 1995.

[16] L. L. Schumaker & S. S. Stanley. Shape-preserving knot removal. *CAGD*, 13(9): 851-872, 1996.

[17] Kaihuai Qin. A matrix method for degree raising of B-spline curves. *Science in China* (Series E). 40(1):71-81, 1997.