The Visual Computer

# General matrix representations for B-splines

Kaihuai Qin

Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, P.R. China
e-mail: qkh-dcs@tsinghua.edu.cn

In this paper, the concept of the 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 the Toeplitz matrix. New recursive matrix representations for uniform B-spline curves and Bézier curves of an arbitrary degree are obtained as special cases of that for nonuniform B-spline curves. The recursive formula for the basis matrix can be substituted for de Boor-Cox's formula for B-splines, and it has a better time complexity than de Boor-Cox's formula when used for computation and conversion of B-spline curves and surfaces between different CAD systems. Finally, some applications of the matrix representations are given in the paper.

**Key words:** B-splines – Matrix representations – Toeplitz matrix

The Visual Computer (2000) 16:177–186


## 1 Introduction

Matrix theory and its algorithms are very useful in computer-aided geometric design, since the matrix is an important and basic tool in mathematics. Matrix formulas of B-spline curves and surfaces have the advantages of both simple computation of points on curves or surfaces and their derivatives and of easy analysis of the geometric properties of B-spline curves and surfaces. In 1982, Chang [1] gave the matrix formula of Bézier curves. Cohen and Riesenfeld [5] gave matrix formulas, not only for Bézier curves but also for 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, and Wang, Sun and Qin [4] in 1993 got the matrix by analogous approaches instead of an explicit formula. In fact, Grabowski and Li rewrote the de Boor-Cox recursive formula in a polynomial form so that the polynomial coefficients can easily be collected into the matrix. At present, an explicit matrix formula for nonuniform rational B-spline (NURBS) curves has not been found, although the matrix representation can be obtained by algorithms.

In this paper, a generalized recursive matrix formula for NURBSs is presented, and new matrix formulas for uniform B-splines and Bézier curves are obtained. Some applications are given in the paper, too.

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

## 2 Representing the de Boor-Cox formula by the Toeplitz matrix

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

## 2.1 The Toeplitz matrix

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

$$\boldsymbol{T}=\begin{bmatrix} \alpha_0 & \alpha_1 \cdots & \alpha_s & & 0 \\ \alpha_{-1} & \alpha_0 \ddots & & \alpha_s & \\ \vdots & & & \ddots & \\ \vdots \cdots & \ddots & \ddots & \ddots \cdots & \alpha_s \\ & & & & \vdots \\ \alpha_{-r} & \cdots & \ddots & \ddots \quad \ddots & \alpha_1 \\ \ddots & & & & \\ 0 & \alpha_{-r} \cdots & \alpha_{-1} & & \alpha_0 \end{bmatrix}.$$

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

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

can be constructed by the coefficients of a polynomial

$$f(\boldsymbol{x})=a_0+a_1x+a_2x^2+\ldots+a_{n-1}x^{n-1}\ (a_{n-1}\neq 0).$$

## 2.2 Representing the product of two polynomials using the Toeplitz matrix

Let

$$g(\boldsymbol{x})=c_0+c_1x+c_2x^2+\ldots+c_{m-1}x^{m-1} \quad (c_{m-1}\neq 0),$$

$$q(\boldsymbol{x})=d_0+d_1x+d_2x^2+\cdots+d_{n-1}x^{n-1} \quad (d_{n-1}\neq 0).$$

One can obtain

$$f(\boldsymbol{x})=g(\boldsymbol{x})q(\boldsymbol{x})$$

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

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

where $\boldsymbol{X}=\left[1\ x\ x^2 \cdots x^{m+n-2}\right]$. This is the Toeplitz matrix representation of the product of two polynomials.

## 2.3 Representing the 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], that is

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

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:

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

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

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

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

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

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

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

with the convention $0/0 = 0$.

## 3 General matrices for NURBS 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: $B_{j,k}(t)$, $j = (i-k+1), (i-k+2), \ldots, i$. They can be represented in a matrix equation as follows:

$$\begin{aligned} &\left\lfloor B_{i-k+1,k}(u)\ B_{i-k+2,k}(u) \cdots B_{i,k}(u) \right\rfloor \\ &\quad = \left[1\ u\ u^2 \cdots u^{k-1}\right] \boldsymbol{M}^k(i), \end{aligned} \tag{3}$$

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

$$\boldsymbol{M}^k(i) = \begin{bmatrix} N_{0,i-k+1}^k & N_{0,i-k+2}^k & \cdots & N_{0,i}^k \\ N_{1,i-k+1}^k & N_{1,i-k+2}^k & \cdots & N_{1,i}^k \\ \vdots & \vdots & \cdots & \vdots \\ N_{k-1,i-k+1}^k & N_{k-1,i-k+2}^k & \cdots & N_{k-1,i}^k \end{bmatrix}$$

and $t_j$ are the knots.

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

$$\begin{aligned} \boldsymbol{c}_{i-k+1} &= \left[B_{i-k+1,k}(u)\ B_{i-k+2,k}(u) \cdots B_{i,k}(u)\right] \times \begin{bmatrix} \boldsymbol{V}_{i-k+1} \\ \boldsymbol{V}_{i-k+2} \\ \vdots \\ \boldsymbol{V}_i \end{bmatrix} \\ &= \left[1\ u\ u^2 \cdots u^{k-1}\right] \boldsymbol{M}^k(i) \begin{bmatrix} \boldsymbol{V}_{i-k+1} \\ \boldsymbol{V}_{i-k+2} \\ \vdots \\ \boldsymbol{V}_i \end{bmatrix}, \end{aligned} \tag{4}$$

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

$\boldsymbol{M}^k(i)$ is referred to as the $i$th basis matrix of the 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 $i$th basis matrix $\boldsymbol{M}^k(i)$ of B-spline basis functions of degree $k-1$ can be obtained by a recursive equation as follows:*

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

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

$$\boldsymbol{M}^k(i) = \begin{bmatrix} N^{k-1}_{0,i-k+1} & 0 \\ N^{k-1}_{1,i-k+1} & N^{k-1}_{0,i-k+1} \\ \vdots & N^{k-1}_{1,i-k+1} \\ N^{k-1}_{k-2,i-k+1} & \vdots \\ 0 & N^{k-1}_{k-2,i-k+1} \end{bmatrix} \times \begin{bmatrix} d_{0,i-k+1} & \overbrace{0\cdots0}^{k-1} \\ d_{1,i-k+1} & 0\cdots0 \end{bmatrix}$$

$$+ \begin{bmatrix} N^{k-1}_{0,i-k+2} & 0 \\ N^{k-1}_{1,i-k+2} & N^{k-1}_{0,i-k+2} \\ \vdots & N^{k-1}_{1,i-k+2} \\ N^{k-1}_{k-2,i-k+2} & \vdots \\ 0 & N^{k-1}_{k-2,i-k+2} \end{bmatrix} \times \begin{bmatrix} 0 & d_{0,i-k+2} & \overbrace{0\cdots0}^{k-2} \\ 0 & d_{1,i-k+2} & 0\cdots0 \end{bmatrix}$$

$$+\cdots+ \begin{bmatrix} N^{k-1}_{0,i} & 0 \\ N^{k-1}_{1,i} & N^{k-1}_{0,i} \\ \vdots & N^{k-1}_{1,i} \\ N^{k-1}_{k-2,i} & \vdots \\ 0 & N^{k-1}_{k-2,i} \end{bmatrix} \begin{bmatrix} \overbrace{0\cdots0}^{k-1} & d_{0,i} \\ 0\cdots0 & d_{1,i} \end{bmatrix}$$

$$+ \begin{bmatrix} N^{k-1}_{0,i-k+2} & 0 \\ N^{k-1}_{1,i-k+2} & N^{k-1}_{0,i-k+2} \\ \vdots & N^{k-1}_{1,i-k+2} \\ N^{k-1}_{k-2,i-k+2} & \vdots \\ 0 & N^{k-1}_{k-2,i-k+2} \end{bmatrix} \begin{bmatrix} h_{0,i-k+1} & \overbrace{0\cdots0}^{k-1} \\ h_{1,i-k+1} & 0\cdots0 \end{bmatrix}$$

$$+ \begin{bmatrix} N^{k-1}_{0,i-k+3} & 0 \\ N^{k-1}_{1,i-k+3} & N^{k-1}_{0,i-k+3} \\ \vdots & N^{k-1}_{1,i-k+3} \\ N^{k-1}_{k-2,i-k+3} & \vdots \\ 0 & N^{k-1}_{k-2,i-k+3} \end{bmatrix} \begin{bmatrix} 0 & h_{0,i-k+2} & \overbrace{0\cdots0}^{k-2} \\ 0 & h_{1,i-k+2} & 0\cdots0 \end{bmatrix}$$

$$+\cdots+ \begin{bmatrix} N^{k-1}_{0,i+1} & 0 \\ N^{k-1}_{1,i+1} & N^{k-1}_{0,i+1} \\ \vdots & N^{k-1}_{1,i+1} \\ N^{k-1}_{k-2,i+1} & \vdots \\ 0 & N^{k-1}_{k-2,i+1} \end{bmatrix} \begin{bmatrix} \overbrace{0\cdots0}^{k-1} & h_{0,i} \\ 0\cdots0 & h_{1,i} \end{bmatrix}.$$

Since it is well known by means of the local property of B-splines [11, 12] that the first term and the last term of the right-hand side of this equation are equal to zero, omitting them yields

$$
\begin{aligned}
\boldsymbol{M}^k(i) = {} & \begin{bmatrix} N^{k-1}_{0,i-k+2} & & 0 \\ N^{k-1}_{1,i-k+2} & N^{k-1}_{0,i-k+2} & \\ \vdots & N^{k-1}_{1,i-k+2} & \\ N^{k-1}_{k-2,i-k+2} & \vdots & \\ 0 & N^{k-1}_{k-2,i-k+2} & \end{bmatrix} \\
& \times \begin{bmatrix} h_{0,i-k+1} & d_{0,i-k+2} & \overbrace{0 \cdots 0}^{k-2} \\ h_{1,i-k+1} & d_{1,i-k+2} & 0 \cdots 0 \end{bmatrix} \\
& + \begin{bmatrix} N^{k-1}_{0,i-k+3} & & 0 \\ N^{k-1}_{1,i-k+3} & N^{k-1}_{0,i-k+3} & \\ \vdots & N^{k-1}_{1,i-k+3} & \\ N^{k-1}_{k-2,i-k+3} & \vdots & \\ 0 & N^{k-1}_{k-2,i-k+3} & \end{bmatrix} \\
& \times \begin{bmatrix} 0 \; h_{0,i-k+2} & d_{0,i-k+3} & \overbrace{0 \cdots 0}^{k-3} \\ 0 \; h_{1,i-k+2} & d_{1,i-k+3} & 0 \cdots 0 \end{bmatrix} + \cdots + \begin{bmatrix} N^{k-1}_{0,i} & 0 \\ N^{k-1}_{1,i} & N^{k-1}_{0,i} \\ \vdots & N^{k-1}_{1,i} \\ N^{k-1}_{k-2,i} & \vdots \\ 0 & N^{k-1}_{k-2,i} \end{bmatrix} \\
& \times \begin{bmatrix} \overbrace{0 \cdots 0}^{k-2} & h_{0,i-1} & d_{0,i} \\ 0 \cdots 0 & h_{1,i-1} & d_{1,i} \end{bmatrix}
\end{aligned}
$$

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

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

□

Equation 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 (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:

$$
\boldsymbol{M}^1(i) = [1], \quad \boldsymbol{M}^2(i) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \quad \boldsymbol{M}^3(i) = \begin{bmatrix} \frac{t_{i+1}-t_i}{t_{i+1}-t_{i-1}} & \frac{t_i-t_{i-1}}{t_{i+1}-t_{i-1}} & 0 \\ \frac{-2(t_{i+1}-t_i)}{t_{i+1}-t_{i-1}} & \frac{2(t_{i+1}-t_i)}{t_{i+1}-t_{i-1}} & 0 \\ \frac{t_{i+1}-t_i}{t_{i+1}-t_{i-1}} & -(t_{i+1}-t_i)\left(\frac{1}{t_{i+1}-t_{i-1}} + \frac{1}{t_{i+2}-t_i}\right) & \frac{t_{i+1}-t_i}{t_{i+2}-t_{i-1}} \end{bmatrix},
$$

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

where

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

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

# 4 Special cases of basis matrices

It is well known that a B-spline curve with equally spaced knots is referred to as a uniform B-spline curve; and a B-spline curve with the knot vector,

$$\Big\{\overbrace{a_0, \cdots, a_0}^{k}, \overbrace{a_1, \cdots, a_1}^{k}\Big\} \ (a_0 < a_1),$$

is regarded as a Bézier curve. Analogously, the basis matrices for uniform B-spline curves and Bézier curves can be obtained by (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 $t_j - t_{j-1} \equiv 1$. Thus, one has

$$\begin{cases} d_{0,j} = \frac{i-j}{k-1}, \\ d_{1,j} = \frac{1}{k-1}. \end{cases} \tag{6}$$

Substituting (6) into (5), one can get:

**Theorem 2.** *The basis matrices of uniform B-splines satisfy the following recursive formula:*

$$\begin{cases} \boldsymbol{M}^k = \frac{1}{k-1} \left\{ \begin{bmatrix} \boldsymbol{M}^{k-1} \\ \boldsymbol{0} \end{bmatrix} \begin{bmatrix} 1 & k-2 & & & 0 \\ & 2 & k-3 & & \\ & & \ddots & \ddots & \\ 0 & & & k-1 & 0 \end{bmatrix} \right. \\ \left. \qquad + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{k-1} \end{bmatrix} \begin{bmatrix} -1 & 1 & & 0 \\ & -1 & 1 & \\ & & \ddots & \ddots \\ 0 & & -1 & 1 \end{bmatrix} \right\}, \\ \boldsymbol{M}^1 = [1]. \end{cases} \tag{7}$$

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

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

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

$$\boldsymbol{M}^k = \begin{bmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,k-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,k-1} \\ \vdots & \vdots & \cdots & \vdots \\ m_{k-1,0} & m_{k-1,1} & \cdots & m_{k-1,k-1} \end{bmatrix}, \tag{8}$$

where

$$m_{i,j} = \frac{1}{(k-1)!} \boldsymbol{C}_{k-1}^{k-1-i} \sum_{s=j}^{k-1} (-1)^{s-j} \times \boldsymbol{C}_k^{s-j} (k-s-1)^{k-1-i},$$

$$\boldsymbol{C}_n^i = \frac{n!}{i!(n-i)!}.$$

Both (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:

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

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

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

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

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

## 4.2 Basis matrices for Bézier curves

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

$$\Big\{\overbrace{0, \cdots, 0}^{k}, \overbrace{1, \cdots, 1}^{k}\Big\}.$$

Thus,

$$\begin{cases} d_{0,j} = 0, \\ d_{1,j} = 1. \end{cases} \tag{9}$$

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

**Theorem 3.** *The basis matrices of Bézier curves satisfy the following recursive formula:*

$$\begin{cases} \boldsymbol{M}^k = \begin{bmatrix} \boldsymbol{M}^{k-1} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{k-1} \end{bmatrix} \begin{bmatrix} -1 & 1 & & 0 \\ & -1 & 1 & 0 \\ & & \ddots & \ddots \\ 0 & & -1 & 1 \end{bmatrix}, \\ \boldsymbol{M}^1 = [1] \end{cases} \tag{10}$$

*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 Bézier curves of degree $k-1$ are independent of $t_i$.
Using (10) recursively step by step, one can also obtain the following matrix for Bézier curves easily:

$$\boldsymbol{M}^k = \begin{bmatrix} m_{0,0} & & & 0 \\ m_{1,0} & m_{1,1} & & \\ \vdots & \vdots & \ddots & \\ m_{k-1,0} & m_{k-1,1} & \cdots & m_{k-1,k-1} \end{bmatrix}, \tag{11}$$

where

$$m_{i,j} = \begin{cases} (-1)^{i-j} \boldsymbol{C}_{k-1}^{j} \boldsymbol{C}_{k-1-j}^{i-j}, & i \geq j; \\ 0, & i < j. \end{cases}$$

$\boldsymbol{M}^k$ is a lower triangular $n \times n$ matrix.
Both (10) and (11) can be used to calculate the basis matrices of Bézier curves. Several examples of the basis matrices for Bézier curves are given as follows:

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

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

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

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

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

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

## 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 NURBS curves

Assume there is a NURBS curve of degree $k-1$.

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

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

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

Then

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

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

## 5.2 Computation of derivatives of NURBS curves

For a NURBS curve of degree $k-1$

$$\begin{aligned}\boldsymbol{c}_{i-k+1}(u) &= \boldsymbol{U}^k\boldsymbol{M}^k(i)\boldsymbol{P}^k(i)/\boldsymbol{U}^k\boldsymbol{M}^k(i)\boldsymbol{W}^k(i)\\ &\triangleq \boldsymbol{R}(u)/S(u),\\ 0 \le u &= (t-t_i)/(t_{i+1}-t_i) < 1\end{aligned}$$

where

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

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

$$\frac{\mathrm{d}^n}{\mathrm{d}u^n}\boldsymbol{c}_{i-k+1}(u) = \sum_{j=0}^{n}\binom{n}{j}\frac{\mathrm{d}^j}{\mathrm{d}u^j}\boldsymbol{R}(u)\frac{\mathrm{d}^{n-j}}{\mathrm{d}u^{n-j}}\frac{1}{S(u)}.$$

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

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}u}\boldsymbol{c}_{i-k+1}(u) &= \frac{-\boldsymbol{R}(u)}{S(u)^2}\frac{\mathrm{d}}{\mathrm{d}u}S(u) + \frac{1}{S(u)}\frac{\mathrm{d}}{\mathrm{d}u}\boldsymbol{R}(u),\\ \frac{\mathrm{d}^2}{\mathrm{d}u^2}\boldsymbol{c}_{i-k+1}(u) &= \boldsymbol{R}(u)\left\{\frac{2}{S(u)^3}\left(\frac{\mathrm{d}}{\mathrm{d}u}S(u)\right)^2\right.\\ &\qquad\left.-\frac{1}{S(u)^2}\frac{\mathrm{d}^2}{\mathrm{d}u^2}S(u)\right\}\\ &\qquad-\frac{2}{S(u)^2}\frac{\mathrm{d}}{\mathrm{d}u}\boldsymbol{R}(u)\frac{\mathrm{d}}{\mathrm{d}u}S(u)\\ &\qquad+\frac{1}{S(u)}\frac{\mathrm{d}^2}{\mathrm{d}u^2}\boldsymbol{R}(u).\end{aligned}$$

## 5.3 Degree raising for NURBS curves

Degree raising for NURBS 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 $k-1$ defined by (12) can be rewritten as follows:

$$\begin{aligned}\boldsymbol{c}_{i-k+1} &= \boldsymbol{U}^{k+1}\begin{bmatrix}\boldsymbol{M}^k(i)\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{V}^k(i)\\ &= \boldsymbol{U}^{k+1}\boldsymbol{M}^{k+1}(i)\boldsymbol{V}^{k+1}(i)\end{aligned}\tag{13}$$

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

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

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

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

where $s_1+s_2+\ldots+s_m = n-k+1$. 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, 15], so that the knot vector becomes as follows:

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

Then, a B-spline curve of degree k can be obtained with degree raising of all the segments of the curve using (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 [15].

## 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 cannot be precisely represented by a curve of degree $k-1$. Of course, it can be approximated by a B-spline curve of a lesser degree. One can obtain

**Table 1.** Comparison of the three methods

| | An arbitrary order ($k$) | | | $k=4$ | | | | |
|---|---|---|---|---|---|---|---|---|
| | $T_{\text{multiplication}}$ | $T_{\text{division}}$ | $T_{\text{addition}}$ | $T_{\text{multiplication}}$ | $T_{\text{division}}$ | $T_{\text{addition}}$ | Sum total ($\cong T_{\text{addition}}$) | Percentage |
| Choi et al. [2] | $\frac{4k^2(4^k-4)}{3}$ | $\frac{4k^2(4^k-4)}{3}$ | $\frac{4k^2(4^k-4)}{3}$ | 5376 | 5376 | 5376 | 17 418 | 12 266 |
| Grabowski and Li [3] | $\frac{(4k^2-6k+2)k}{3}$ | $\frac{k(2k^2-3k+1)}{3}$ | $\frac{(6k^2-9k+3)k}{2}$ | 56 | 28 | 126 | 220 | 155 |
| The new method | $\frac{(4k^2-6k+2)k}{3}$ | $k(k-1)$ | $\frac{(2k^2+k-3)k}{2}$ | 56 | 12 | 66 | 142 | 100 |

the least-square solution of this problem for a segment of B-spline curve using (13)

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

Similarly, one can obtain the least-square solution of this problem for a whole B-spline curve.

## 6 Conclusions

By means of the basis matrix proposed in this paper, the matrix representations of nonuniform and uniform B-splines and Bézier curves can be unified by a recursive formula. It is shown that the matrix representations for uniform B-splines and Bézier curves can be regarded as special cases of the basis matrix of NURBS. Like de Boor-Cox recursive definition of B-splines, the basis matrices of B-splines can be defined by (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 (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 NURBS of an arbitrary degree, or (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 (5), for NURBS of an arbitrary degree is more efficient than the symbolic approach in [2] and the numerical algorithm in [3] in numerical evaluation as well as the de Boor-Cox formula. Assume that the execution times $T$ satisfy [2] $T_{\text{addition}} \cong T_{\text{subtraction}}$, $T_{\text{multiplication}} \cong T_{\text{division}}$ and $T_{\text{multiplication}} \cong 1.12 T_{\text{addition}}$. The comparison of the three methods for the basis matrices of NURBS of degree $k-1$ is shown in Table 1.

*Acknowledgements.* This project was supported by the National Natural Science Foundation (NSF) of China.

## References


1. Chang G (1982) Matrix foundation of Bézier technique. Comput Aided Des 14:354–350
2. Choi BK, Yoo WS, Lee CS (1990) Matrix representation for NURB curves and surfaces. Comput Aided Des 22:235–240
3. Grabowski H, Li X (1992) Coefficient formula and matrix of nonuniform B-spline functions. Comput Aided Des 24:637–642
4. Wang X, Sun J, Qin K (1993) Symbolic matrix representation of NURBS and its applications (in Chinese). Chinese J Comput 16:29–34
5. Cohen E, Riesenfeld RF (1982) General matrix representations for Bézier and B-spline curves. Comput Ind 3:9–15
6. Boor C de (1972) On calculating with B-splines. J Approx Theory 6:50–62
7. Vidom H (1965) Toeplitz matrices. In: Hirschmann I (ed) Studies in Real and Complex Analysis. MAA Studies in Mathematics 3. Prentice-Hall, Inc., Englewood Cliffs, NJ, pp 179–209
8. Qin K, Guan Y (1997) Two algorithms for inserting knots into B-spline curves (in Chinese). Chinese J Comput 20:557–561
9. Schoenberg IJ (1946) Contributions to the problem of approximation of equidistant data by analytic functions. Quart Appl Math 4:112–141
10. Cox MG (1972) The numerical evaluation of B-splines. J Inst Math Appl 10:134–149
11. Boor C de (1978) Practical Guide to Splines. Springer, Berlin, Heidelberg, New York

12. Schumaker L (1981) Spline functions: basic theory. John Wiley, New York
13. Qin K (1996) A new algorithm for degree-raising of nonuniform B-spline curves (in Chinese). Chinese J Comput 19:537–542
14. Boehm W (1980) Inserting new knots into B-spline curves. Comput Aided Des 12:199–201
15. Qin K (1997) A matrix method for degree raising of B-spline curves. Sci China (Series E) 40:71–81

KAIHUAI QIN is a Professor of Computer Science and Technology, at Tsinghua University, Beijing, P.R. China. Dr. Qin was a Postdoctoral Fellow from 1990 to 1992, then joined the Department of Computer Science and Technology of Tsinghua University of China as an Associate Professor. He received his PhD and MEng from Huazhong University of Science and Technology (HUST) in 1990 and 1984, and his BEng from South China University of Technology in 1982. Before joining Tsinghua University, he was teaching as a TA first, then a lecturer at HUST from late 1984–1990. His research interests include computer graphics, CAGD, curves and surfaces, especially subdivision surfaces and NURBS modeling, physics-based geometric modeling, wavelets, medical visualization, surgical planning and simulation, virtual reality and intelligent and smart CAD/CAM.