A Tutorial on Uniform B-Spline

Yi Zhou
Neuromorphic Automation and Intelligence Lab (NAIL),
School of Robotics, Hunan University
Email: eeyzhou@hnu.edu.cn

September 28, 2023

This document facilitates understanding of core concepts about **uniform B-spline** and its matrix representation. All the contents are borrowed from [1, 3–5] and rephrased such that the symbolic system and definitions are unified.

1 Cox-de Boor Formula

Here we focus on the uniform case, namely all knots are evenly distributed.

A uniform B-spline of degree k is defined by the control points $\mathbf{P}_i$ $(i \in [0, N-1])$ and their corresponding weights, a.k.a the basis functions $B_{i,k}(\tau)$:

$$\mathbf{P}(\tau) \doteq \sum_{i=0}^{N-1} B_{i,k}(\tau)\mathbf{P}_i. \tag{1}$$

The number of knots are determined by $M = k + N + 1$, where $N = k + 1$. Here we do not specify the domain on which $\mathbf{P}$ is defined. It could be either $\mathcal{R}^d$ or $\mathrm{SE}(3)$.

The b-spline can also be regarded as a polynomial of the temporal parameter weighted by the control points. The polynomial of degree k (i.e., the basis function at the top level) is calculated recursively from degree 0 (i.e., the bottom level). The recursive method is called *Cox-de Boor* formula,

$$B_{i,0}(\tau) = \begin{cases} 1 & \text{if } \tau \in [\tau_i, \tau_{i+1}], \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

$$\begin{aligned} B_{i,k}(\tau) &= \frac{\tau - \tau_i}{\tau_{i+k} - \tau_i} B_{i,k-1}(\tau) + \frac{\tau_{i+k+1} - \tau}{\tau_{i+k+1} - \tau_{i+1}} B_{i+1,k-1}(\tau) \\ &= \frac{\tau - \tau_i}{k\Delta\tau} B_{i,k-1}(\tau) + \frac{\tau_{i+k+1} - \tau}{k\Delta\tau} B_{i+1,k-1}(\tau), \end{aligned} \tag{3}$$

where $\Delta\tau$ denotes the interval between successive knots.

How to read the Cox-de Boor formula (Eq. 3)?

To read the formula, we need to understand the meaning of the subscripts of the basis function. The first subscript i is associated to the corresponding control point $\mathbf{P}_i$ (which shares the same index i). It is also associated to the index of the **very left knot** (i.e., τ_i) in the corresponding non-zero domain (see the triangular computation scheme in [1]). The second subscript k denotes the degree of the basis function. The higher the degree, the wider the non-zero domain. In other words, the non-zero domain can be determined by the two subscripts, namely $[\tau_i, \tau_{i+k+1}]$ for $B_{i,k}$ as an example.

The *Cox-de Boor* formula can be read as: The basis function of degree k at position i is derived from two subordinate basis functions of degree $k-1$ at position i and $i+1$, respectively. The polynomial weights can be regarded as "linear interpolation coefficients"[1] normalized by the width of the corresponding non-zero domain, which can be calculated as the $\mathbf{2_{nd}}$ **subscript** $\mathbf{+1+}$ the $\mathbf{1_{st}}$ **subscript** $-$ the $\mathbf{1_{st}}$ **subscript**, namely the $2_{\mathbf{nd}}$ **subscript** $\mathbf{+1}$. In other words, the width of the non-zero domain for $B_{i,k}$ is $(k+1)\Delta\tau$.

1.1 Cumulative Formula

Eq. 1 can also be represented by the cumulative form,

$$\mathbf{P}(\tau) = \tilde{B}_{0,k}(\tau)\mathbf{P}_0 + \sum_{i=1}^{N-1} \tilde{B}_{i,k}(\tau)(\mathbf{P}_i - \mathbf{P}_{i-1}), \tag{4}$$
$$\tilde{B}_{i,k}(\tau) = \sum_{s=i}^{N-1} B_{s,k}(\tau).$$

2 Matrix Representation of the Cox-de Boor formula

B-splines have local support, which means that for a spline of degree k, only $k+1$ control points contribute to the value of the spline at a given τ. As shown in [4], it is possible to represent the spline coefficients using a matrix representation, which is constant for uniform B-splines.

An explicitly recursive matrix formula was presented in [4] for non-uniform B-spline curves of an arbitrary degree by means of the Toeplitz matrix. In this section, we first revisit the idea of the Toeplitz matrix, based on which the matrix representation of the Cox-de Boor formula is derived.

2.1 Toeplitz Matrix

The Toeplitz matrix is a banded-shape matrix, whose elements on any line parallel to the main diagonal are all equal. A special Toeplitz matrix is a lower triangular matrix

[1] Strictly speaking, this is not a linear interpolation, because the denominators of the two weights are $\tau_{i+k} - \tau_i$ and $\tau_{i+k+1} - \tau_{i+1}$, respectively, though being equal numerically.

$$\mathbf{T} = \begin{bmatrix} a_0 & 0 & 0 & 0 & 0 & 0 \\ a_1 & a_0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & 0 & 0 \\ a_n & \dots & \ddots & \ddots & 0 & 0 \\ 0 & a_n & \dots & \ddots & \ddots & 0 \\ 0 & 0 & a_n & \dots & a_1 & a_0 \end{bmatrix}, \tag{5}$$

whose elements are the coefficients of the following polynomial,

$$f(x) = a_0 + a_1 x + \ldots + a_n x^n (n \neq 0). \tag{6}$$

($\star$) **Toeplitz matrix can also be used to represent the product of two polynomials.** Here is a specific example. Let

$$\begin{aligned} g(x) &= c_0 + c_1 x + \ldots + c_2 x^2, \\ q(x) &= d_0 + d_1 x + \ldots + d_3 x^3. \end{aligned}$$

One can obtain the product $f(x) = g(x)q(x)$ in the matrix representation as,

$$\begin{aligned} f(x) &= \mathbf{X} \begin{bmatrix} c_0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & c_0 & 0 & 0 & 0 & 0 \\ c_2 & c_1 & c_0 & 0 & 0 & 0 \\ 0 & c_2 & c_1 & c_0 & 0 & 0 \\ 0 & 0 & c_2 & c_1 & c_0 & 0 \\ 0 & 0 & 0 & c_2 & c_1 & c_0 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ 0 \\ 0 \end{bmatrix} \\ &= \mathbf{X} \begin{bmatrix} c_0 & 0 & 0 & 0 \\ c_1 & c_0 & 0 & 0 \\ c_2 & c_1 & c_0 & 0 \\ 0 & c_2 & c_1 & c_0 \\ 0 & 0 & c_2 & c_1 \\ 0 & 0 & 0 & c_2 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}, \end{aligned} \tag{7}$$

where $\mathbf{X} = [1, x, x^2, \cdots, x^5]$. Note that the dimension (row) of the coefficient matrix is defined by the degree of the variable (i.e., 5 + 1 = 6).

2.2 Representing the Cox-de Boor Formula Using Toeplitz Matrix

To preserve numerical stability, it is typical to use a normalized variable u, which can be transferred from τ by means of basis translation [2]. Thus, the basis function $B_{i,k}(u)$ can be represented as

$$B_{i,k}(u) = [1 \; u \; u^2 \; \cdots \; u^k] \begin{bmatrix} N_{i,k}^0 \\ N_{i,k}^1 \\ N_{i,k}^2 \\ \vdots \\ N_{i,k}^k \end{bmatrix}, \tag{8}$$

where $N^{\{\cdot\}}_{\{\cdot,\cdot\}}$ denotes the coefficients of the polynomial. The colors of the super-/subscripts specify the association. Note that the superscript of $N^{\{\cdot\}}_{\{\cdot,\cdot\}}$ is only a symbol that specifies the association to the power of variable u rather than a power.

The following is the derivation of the basis translation originating from Eq. 3.

$$\begin{aligned} B_{i,k}(\tau) &= \frac{\tau-\tau_i}{\tau_{i+k}-\tau_i}B_{i,k-1}(\tau) + \frac{\tau_{i+k+1}-\tau}{\tau_{i+k+1}-\tau_{i+1}}B_{i+1,k-1}(\tau) \\ &= \frac{(\tau_{j+1}-\tau_j)(\tau-\tau_j+\tau_j-\tau_i)}{(\tau_{j+1}-\tau_j)(\tau_{i+k}-\tau_i)}B_{i,k-1}(\tau) \\ &+ \frac{(\tau_{j+1}-\tau_j)(\tau_{i+k+1}-\tau_j+\tau_j-\tau)}{(\tau_{j+1}-\tau_j)(\tau_{i+k+1}-\tau_{i+1})}B_{i+1,k-1}(\tau) \\ &= \left[\frac{\tau_j-\tau_i}{\tau_{i+k}-\tau_i} + \frac{\tau-\tau_j}{\tau_{j+1}-\tau_j}\frac{\tau_{j+1}-\tau_j}{\tau_{i+k}-\tau_i}\right]B_{i,k-1}(\tau) \\ &+ \left[\frac{\tau_{i+k+1}-\tau_j}{\tau_{i+k+1}-\tau_{i+1}} - \frac{\tau-\tau_j}{\tau_{j+1}-\tau_j}\frac{\tau_{j+1}-\tau_j}{\tau_{i+k+1}-\tau_{i+1}}\right]B_{i+1,k-1}(\tau), \end{aligned} \tag{9}$$

where $\tau \in [\tau_j, \tau_{j+1}]$. **In a specific case ($k = 3$), the non-zero domain is** $[\tau_3, \tau_4]$ **(namely j = 3), and** $i = 0, 1, \cdots, 3$**.** Let

$$u = \frac{\tau-\tau_j}{\tau_{j+1}-\tau_j}, \tag{10}$$

$$d_i^0 = \frac{\tau_j-\tau_i}{\tau_{i+k}-\tau_i},\ d_i^1 = \frac{\tau_{j+1}-\tau_j}{\tau_{i+k}-\tau_i}, \tag{11}$$

$$h_i^0 = \frac{\tau_{i+k+1}-\tau_j}{\tau_{i+k+1}-\tau_{i+1}},\ h_i^1 = -\frac{\tau_{j+1}-\tau_j}{\tau_{i+k+1}-\tau_{i+1}}, \tag{12}$$

with the convention $\frac{0}{0} = 0$. Then Eq. 9 turns to

$$B_{i,k}(u) = (d_i^0 + u\,d_i^1)B_{i,k-1}(u) + (h_i^0 + u\,h_i^1)B_{i+1,k-1}(u). \tag{13}$$

Using property ($\star$), Eq. 13 can be represented by a matrix. Here, for simplicity, we use a specific case ($k = 3$) as an example,

$$\begin{aligned} B_{i,3} = [1\ u\ u^2\ u^3]\Bigg\{ &\begin{bmatrix} N^0_{i,2} & 0 & | & 0 & 0 \\ N^1_{i,2} & N^0_{i,2} & | & 0 & 0 \\ N^2_{i,2} & N^1_{i,2} & | & N^0_{i,2} & 0 \\ 0 & N^2_{i,2} & | & N^1_{i,2} & N^0_{i,2} \end{bmatrix} \begin{bmatrix} d_i^0 \\ d_i^1 \\ - \\ 0 \\ 0 \end{bmatrix} \\ + &\begin{bmatrix} N^0_{i+1,2} & 0 & | & 0 & 0 \\ N^1_{i+1,2} & N^0_{i+1,2} & | & 0 & 0 \\ N^2_{i+1,2} & N^1_{i+1,2} & | & N^0_{i+1,2} & 0 \\ 0 & N^2_{i+1,2} & | & N^1_{i+1,2} & N^0_{i+1,2} \end{bmatrix} \begin{bmatrix} h_i^0 \\ h_i^1 \\ - \\ 0 \\ 0 \end{bmatrix} \Bigg\}, \end{aligned} \tag{14}$$

where $N^{\{\cdot\}}_{\{\cdot,\cdot\}}$ refers to the coefficients of polynomial $B_{i,k}$, and their superscripts still do not represent a power.

3 Representing B-Spline Curves with Basis Matrices

3.1 General Matrices for NURBS

Based on the basis translation introduced in Eq. 8, the B-spline formula (Eq. 1) can be represented as

$$\mathbf{P}(u) = \sum_{i=0} B_{i,k}(u)\mathbf{P}_i. \tag{15}$$

Still, we use $k = 3$ as a specific example, and therefore, we can obtain

$$\begin{aligned} \mathbf{P}(u)^T &= [B_{0,3}(u)\ B_{1,3}(u)\ B_{2,3}(u)\ B_{3,3}(u)] \begin{bmatrix} \mathbf{P}_0^T \\ \mathbf{P}_1^T \\ \mathbf{P}_2^T \\ \mathbf{P}_3^T \end{bmatrix} \\ &\overset{(Eq.\ 8)}{=} [1\ u\ u^2\ u^3] \underbrace{\begin{bmatrix} N_{0,3}^0 & N_{1,3}^0 & N_{2,3}^0 & N_{3,3}^0 \\ N_{0,3}^1 & N_{1,3}^1 & N_{2,3}^1 & N_{3,3}^1 \\ N_{0,3}^2 & N_{1,3}^2 & N_{2,3}^2 & N_{3,3}^2 \\ N_{0,3}^3 & N_{1,3}^3 & N_{2,3}^3 & N_{3,3}^3 \end{bmatrix}}_{\mathbf{M}^3(3)} \begin{bmatrix} \mathbf{P}_0^T \\ \mathbf{P}_1^T \\ \mathbf{P}_2^T \\ \mathbf{P}_3^T \end{bmatrix}, \end{aligned} \tag{16}$$

where $u = \frac{\tau - \tau_3}{\tau_4 - \tau_3} \in [0, 1]$. The matrix $\mathbf{M}^k(j)$ is referred to as **basis matrix**. The core of this section is to derive the recursive formula for the basis matrices of B-splines of degree k.

According to Eq. 14, the basis matrix $\mathbf{M}^3(3)$ can be represented as

$$
\begin{aligned}
\mathbf{M}^3(3) &= \begin{bmatrix} N_{0,3}^0 & 0 & 0 & 0 \\ N_{0,3}^1 & 0 & 0 & 0 \\ N_{0,3}^2 & 0 & 0 & 0 \\ N_{0,3}^3 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & N_{1,3}^0 & 0 & 0 \\ 0 & N_{1,3}^1 & 0 & 0 \\ 0 & N_{1,3}^2 & 0 & 0 \\ 0 & N_{1,3}^3 & 0 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} 0 & 0 & N_{2,3}^0 & 0 \\ 0 & 0 & N_{2,3}^1 & 0 \\ 0 & 0 & N_{2,3}^2 & 0 \\ 0 & 0 & N_{2,3}^3 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & N_{3,3}^0 \\ 0 & 0 & 0 & N_{3,3}^1 \\ 0 & 0 & 0 & N_{3,3}^2 \\ 0 & 0 & 0 & N_{3,3}^3 \end{bmatrix} \\
&= \begin{bmatrix} N_{0,2}^0 & 0 \\ N_{0,2}^1 & N_{0,2}^0 \\ N_{0,2}^2 & N_{0,2}^1 \\ 0 & N_{0,2}^2 \end{bmatrix} \begin{bmatrix} d_0^0 & 0 & 0 & 0 \\ d_0^1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} N_{1,2}^0 & 0 \\ N_{1,2}^1 & N_{1,2}^0 \\ N_{1,2}^2 & N_{1,2}^1 \\ 0 & N_{1,2}^2 \end{bmatrix} \begin{bmatrix} h_0^0 & 0 & 0 & 0 \\ h_0^1 & 0 & 0 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} N_{1,2}^0 & 0 \\ N_{1,2}^1 & N_{1,2}^0 \\ N_{1,2}^2 & N_{1,2}^1 \\ 0 & N_{1,2}^2 \end{bmatrix} \begin{bmatrix} 0 & d_1^0 & 0 & 0 \\ 0 & d_1^1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} N_{2,2}^0 & 0 \\ N_{2,2}^1 & N_{2,2}^0 \\ N_{2,2}^2 & N_{2,2}^1 \\ 0 & N_{2,2}^2 \end{bmatrix} \begin{bmatrix} 0 & h_1^0 & 0 & 0 \\ 0 & h_1^1 & 0 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} N_{2,2}^0 & 0 \\ N_{2,2}^1 & N_{2,2}^0 \\ N_{2,2}^2 & N_{2,2}^1 \\ 0 & N_{2,2}^2 \end{bmatrix} \begin{bmatrix} 0 & 0 & d_2^0 & 0 \\ 0 & 0 & d_2^1 & 0 \end{bmatrix} + \begin{bmatrix} N_{3,2}^0 & 0 \\ N_{3,2}^1 & N_{3,2}^0 \\ N_{3,2}^2 & N_{3,2}^1 \\ 0 & N_{3,2}^2 \end{bmatrix} \begin{bmatrix} 0 & 0 & h_2^0 & 0 \\ 0 & 0 & h_2^1 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} N_{3,2}^0 & 0 \\ N_{3,2}^1 & N_{3,2}^0 \\ N_{3,2}^2 & N_{3,2}^1 \\ 0 & N_{3,2}^2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & d_3^0 \\ 0 & 0 & 0 & d_3^1 \end{bmatrix} + \begin{bmatrix} N_{4,2}^0 & 0 \\ N_{4,2}^1 & N_{4,2}^0 \\ N_{4,2}^2 & N_{4,2}^1 \\ 0 & N_{4,2}^2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & h_3^0 \\ 0 & 0 & 0 & h_3^1 \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} N_{0,2}^0 & 0 \\ N_{0,2}^1 & N_{0,2}^0 \\ N_{0,2}^2 & N_{0,2}^1 \\ 0 & N_{0,2}^2 \end{bmatrix}}_{=\mathbf{0}} \begin{bmatrix} d_0^0 & 0 & 0 & 0 \\ d_0^1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} N_{1,2}^0 & 0 \\ N_{1,2}^1 & N_{1,2}^0 \\ N_{1,2}^2 & N_{1,2}^1 \\ 0 & N_{1,2}^2 \end{bmatrix} \begin{bmatrix} h_0^0 & d_1^0 & 0 & 0 \\ h_0^1 & d_1^1 & 0 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} N_{2,2}^0 & 0 \\ N_{2,2}^1 & N_{2,2}^0 \\ N_{2,2}^2 & N_{2,2}^1 \\ 0 & N_{2,2}^2 \end{bmatrix} \begin{bmatrix} 0 & h_1^0 & d_2^0 & 0 \\ 0 & h_1^1 & d_2^1 & 0 \end{bmatrix} + \begin{bmatrix} N_{3,2}^0 & 0 \\ N_{3,2}^1 & N_{3,2}^0 \\ N_{3,2}^2 & N_{3,2}^1 \\ 0 & N_{3,2}^2 \end{bmatrix} \begin{bmatrix} 0 & 0 & h_2^0 & d_3^0 \\ 0 & 0 & h_2^1 & d_3^1 \end{bmatrix} \\
&+ \underbrace{\begin{bmatrix} N_{4,2}^0 & 0 \\ N_{4,2}^1 & N_{4,2}^0 \\ N_{4,2}^2 & N_{4,2}^1 \\ 0 & N_{4,2}^2 \end{bmatrix}}_{=\mathbf{0}} \begin{bmatrix} 0 & 0 & 0 & h_3^0 \\ 0 & 0 & 0 & h_3^1 \end{bmatrix}.
\end{aligned} \tag{17}
$$

The first and last terms in Eq. 17 equal to $\mathbf{0}$, because the corresponding basis functions (i.e., $B_{0,2}$ and $B_{4,2}$) are not defined in $[\tau_3, \tau_4]$ (see the triangular computation scheme in [1]).

$$
\begin{aligned}
Eq.\ 17 =& \begin{bmatrix} N_{1,2}^{0} \\ N_{1,2}^{1} \\ N_{1,2}^{2} \\ 0 \end{bmatrix} \begin{bmatrix} h_0^0 & d_1^0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ N_{1,2}^{0} \\ N_{1,2}^{1} \\ N_{1,2}^{2} \end{bmatrix} \begin{bmatrix} h_0^1 & d_1^1 & 0 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} N_{2,2}^{0} \\ N_{2,2}^{1} \\ N_{2,2}^{2} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & h_1^0 & d_2^0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ N_{2,2}^{0} \\ N_{2,2}^{1} \\ N_{2,2}^{2} \end{bmatrix} \begin{bmatrix} 0 & h_1^1 & d_2^1 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} N_{3,2}^{0} \\ N_{3,2}^{1} \\ N_{3,2}^{2} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & h_2^0 & d_3^0 \end{bmatrix} + \begin{bmatrix} 0 \\ N_{3,2}^{0} \\ N_{3,2}^{1} \\ N_{3,2}^{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & h_2^1 & d_3^1 \end{bmatrix} \\
=& \begin{bmatrix} N_{1,2}^{0} & N_{2,2}^{0} & N_{3,2}^{0} \\ N_{1,2}^{1} & N_{2,2}^{1} & N_{3,2}^{1} \\ N_{1,2}^{2} & N_{2,2}^{2} & N_{3,2}^{2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_0^0 & d_1^0 & 0 & 0 \\ 0 & h_1^0 & d_2^0 & 0 \\ 0 & 0 & h_2^0 & d_3^0 \end{bmatrix} \\
&+ \begin{bmatrix} 0 & 0 & 0 \\ N_{1,2}^{0} & N_{2,2}^{0} & N_{3,2}^{0} \\ N_{1,2}^{1} & N_{2,2}^{1} & N_{3,2}^{1} \\ N_{1,2}^{2} & N_{2,2}^{2} & N_{3,2}^{2} \end{bmatrix} \begin{bmatrix} h_0^1 & d_1^1 & 0 & 0 \\ 0 & h_1^1 & d_2^1 & 0 \\ 0 & 0 & h_2^1 & d_3^1 \end{bmatrix} \\
\overset{(Eq.\ 16)}{=}& \begin{bmatrix} \mathbf{M}^2(3) \\ \mathbf{0}^T \end{bmatrix} \begin{bmatrix} h_0^0 & d_1^0 & 0 & 0 \\ 0 & h_1^0 & d_2^0 & 0 \\ 0 & 0 & h_2^0 & d_3^0 \end{bmatrix} + \begin{bmatrix} \mathbf{0}^T \\ \mathbf{M}^2(3) \end{bmatrix} \begin{bmatrix} h_0^1 & d_1^1 & 0 & 0 \\ 0 & h_1^1 & d_2^1 & 0 \\ 0 & 0 & h_2^1 & d_3^1 \end{bmatrix} \\
=& \begin{bmatrix} \mathbf{M}^2(3) \\ \mathbf{0}^T \end{bmatrix} \begin{bmatrix} 1-d_1^0 & d_1^0 & 0 & 0 \\ 0 & 1-d_2^0 & d_2^0 & 0 \\ 0 & 0 & 1-d_3^0 & d_3^0 \end{bmatrix} \\
&+ \begin{bmatrix} \mathbf{0}^T \\ \mathbf{M}^2(3) \end{bmatrix} \begin{bmatrix} -d_1^1 & d_1^1 & 0 & 0 \\ 0 & -d_2^1 & d_2^1 & 0 \\ 0 & 0 & -d_3^1 & d_3^1 \end{bmatrix},
\end{aligned} \tag{18}
$$

and $\mathbf{M}^0(3) = B_{3,0}(u) = 1$, where $u = \frac{\tau - \tau_3}{\tau_4 - \tau_3} \in [0, 1]$. To understand the second last equation, please recall Eq. 16 that the basis matrix $\mathbf{M}$ is made up by coefficients of the polynomials (basis functions). To further help memorizing the elements of the basis matrix, please refer to Fig. 1. To construct basis matrix $\mathbf{M}^k(j)$, just look up the column with the corresponding degree (the second subscript) in the blue triangle, and then apply Eq. 16.

Eq. 18 can be regarded as the recursive definition of basis matrix. It can be used in the symbolic computation of NURBS.

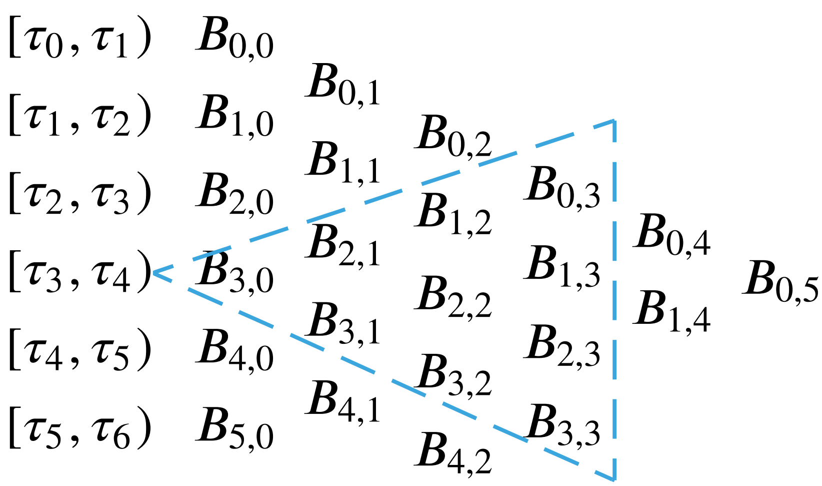

Figure 1: The triangular computation scheme of B-spline.

3.2 Basis Matrix $\mathrm{M}^k(j)$ of Uniform B-Spline

In this section, we provide the general term formula of $\mathrm{M}^k(j)$ for uniform B-spline.

$$
\begin{aligned}
\mathbf{M}^{k}(j) = {} & \frac{1}{k}\left\{ \begin{bmatrix} \mathbf{M}^{k-1}(j) \\ \mathbf{0}^T \end{bmatrix} \begin{bmatrix} k+1-j & j-1 & & & 0 \\ 0 & k+2-j & j-2 & & \\ & & \ddots & \ddots & \\ 0 & & & k+3-j & 0 \end{bmatrix} \right. \\
& \left. + \begin{bmatrix} \mathbf{0}^T \\ \mathbf{M}^{k-1}(j) \end{bmatrix} \begin{bmatrix} -1 & 1 & & & 0 \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ 0 & & & -1 & 1 \end{bmatrix} \right\} \\
\overset{(\star\star)}{=} {} & \frac{1}{k}\left\{ \begin{bmatrix} \mathbf{M}^{k-1}(j) \\ \mathbf{0}^T \end{bmatrix} \begin{bmatrix} 1 & k-1 & & & 0 \\ 0 & 2 & k-2 & & \\ & & \ddots & \ddots & \\ 0 & & & 3 & 0 \end{bmatrix} \right. \\
& \left. + \begin{bmatrix} \mathbf{0}^T \\ \mathbf{M}^{k-1}(j) \end{bmatrix} \begin{bmatrix} -1 & 1 & & & 0 \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ 0 & & & -1 & 1 \end{bmatrix} \right\},
\end{aligned} \tag{19}
$$

and $\mathbf{M}^0(j) = 1$.

Note that $(\star\star)$ holds based on the fact that $j = \frac{\overbrace{k + (k+1) + 1}^{\text{\# knots}}}{2} - 1 = k$. Unlike the basis matrices of NURBSs, the basis matrices of uniform B-splines of degree k are independent of t_j. The basis matrices for uniform B-splines are given as follows:

$$
\begin{aligned}
\mathbf{M}^0(j) &= 1, \\
\mathbf{M}^1(j) &= \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \\
\mathbf{M}^2(j) &= \frac{1}{2!}\begin{bmatrix} 1 & 1 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix}, \\
\mathbf{M}^3(j) &= \frac{1}{3!}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \qquad (20) \\
&\vdots
\end{aligned}
$$

There is no need to memorize Eq.(8) in [4].

3.3 Basis Matrices in the Cumulative Formula

For the cumulative formula (Eq. 4), we can obtain a similar representation. Here we still use the specific case ($k = 3$) as an example.

$$\mathbf{P}(u) = \tilde{B}_{0,3}(u)\mathbf{P}_0 + \sum_{i=1}^{3} \tilde{B}_{i,3}(u) \underbrace{(\mathbf{P}_i - \mathbf{P}_{i-1})}_{\mathbf{d}_i}, \qquad (21)$$

where $\tilde{B}_{i,k} = \sum_{s=i}^{3} B_{s,3}(u)$. Specifically,

$$
\begin{aligned}
\tilde{B}_{0,3} &= B_{0,3} + B_{1,3} + B_{2,3} + B_{3,3} \\
\tilde{B}_{1,3} &= B_{1,3} + B_{2,3} + B_{3,3} \\
\tilde{B}_{2,3} &= B_{2,3} + B_{3,3} \\
\tilde{B}_{3,3} &= B_{3,3}. \qquad (22)
\end{aligned}
$$

Following the format in Eq. 16, the differential cumulative formula (Eq. 21) can be represented, by dropping off u for simplicity, as

$$
\begin{aligned}
\mathbf{P}(u)^T &= [\tilde{B}_{0,3}\ \tilde{B}_{1,3}\ \tilde{B}_{2,3}\ \tilde{B}_{3,3}] \begin{bmatrix} \mathbf{P}_0^T \\ \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \mathbf{d}_3^T \end{bmatrix} \\
&= \Big\{ [B_{0,3}\ B_{1,3}\ B_{2,3}\ B_{3,3}] + [B_{1,3}\ B_{2,3}\ B_{3,3}\ 0] + \cdots \\
&\quad + [B_{2,3}\ B_{3,3}\ 0\ 0] + [B_{3,3}\ 0\ 0\ 0] \Big\} \begin{bmatrix} \mathbf{P}_0^T \\ \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \mathbf{d}_3^T \end{bmatrix} \\
&= [1\ u\ u^2\ u^3] \Big\{ [\mathbf{m}_0\ \mathbf{m}_1\ \mathbf{m}_2\ \mathbf{m}_3] + [\mathbf{m}_1\ \mathbf{m}_2\ \mathbf{m}_3\ 0] \\
&\quad + [\mathbf{m}_2\ \mathbf{m}_3\ 0\ 0] + [\mathbf{m}_3\ 0\ 0\ 0] \Big\} \begin{bmatrix} \mathbf{P}_0^T \\ \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \mathbf{d}_3^T \end{bmatrix} \\
&= [1\ u\ u^2\ u^3] \cdot \left[\textstyle\sum_{s=0}^3 \mathbf{m}_s \ \ \middle|\ \ \sum_{s=1}^3 \mathbf{m}_s \ \ \middle|\ \ \sum_{s=2}^3 \mathbf{m}_s \ \ \middle|\ \ \mathbf{m}_3 \right] \begin{bmatrix} \mathbf{P}_0^T \\ \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \mathbf{d}_3^T \end{bmatrix} \\
&\overset{\text{(substitute Eq. 20)}}{=} \frac{1}{3!}[1\ u\ u^2\ u^3] \cdot \begin{bmatrix} 6 & 5 & 1 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & -3 & 3 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0^T \\ \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \mathbf{d}_3^T \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} 1 & \frac{5+3u-3u^2+u^3}{6} & \frac{1+3u+3u^2-2u^3}{6} & \frac{u^3}{6} \end{bmatrix}}_{\boldsymbol{\lambda}} \begin{bmatrix} \mathbf{P}_0^T \\ \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \mathbf{d}_3^T \end{bmatrix} \\
&= \mathbf{P}_0^T + \sum_{i=1}^{3} \boldsymbol{\lambda}_i(u) \mathbf{d}_i^T, \qquad (23)
\end{aligned}
$$

where $\boldsymbol{\lambda}_0 = 1$, $u = \frac{\tau - \tau_3}{\tau_4 - \tau_3} \in [0, 1]$.

4 FAQs

- Q1: **Knots** vs **Control Points**
  A: Knots are a list of positions in the parametric domain (i.e., $\tau_i \in [0, 1]$.) For uniform B-splines, knots are evenly distributed in the parametric domain. The number of knots is determined if the degree of B-spline is known (see Sec.1).

Control points are design parameters from human's input. Once the degree k is set and the control points are provided, one can evaluate the value at any given position τ in the non-zero domain, which is spanned by the two knots in the middle (e.g., $[\tau_3, \tau_4]$ for a B-spline of degree 3).

Some papers, such as [5], somehow treat knots and control points identically. This is not consistent with the majority of the literature. Thus, we regard that in [5] as an improper (wrong) definition; better not use it.

- Q2: How to understand **basis translation**?
  This is actually a trivial operation (see Eq. 10). However, I felt confused when I read the descriptions in academic papers (e.g., the 2_{nd} paragraph in Section 4.2 of [5], and in Sec. IV of [3].) The confusion is caused mainly by the inconsistent symbolic definition and descriptions. In general, the **basis translation** just simply translates and re-scales the non-zero domain to $[0, 1]$ such that the numerical stability is preserved.

References

[1] B-spline basis functions: Definition. `https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/spline/B-spline/bspline-basis.html`.

[2] Carl De Boor. On calculating with b-splines. *Journal of Approximation theory*, 6(1):50–62, 1972.

[3] Wenchao Ding, Wenliang Gao, Kaixuan Wang, and Shaojie Shen. An efficient b-spline-based kinodynamic replanning framework for quadrotors. *IEEE Transactions on Robotics*, 35(6):1287–1306, 2019.

[4] Kaihuai Qin. General matrix representations for b-splines. *The Visual Computer*, 16(3-4):177–186, 2000.

[5] Christiane Sommer, Vladyslav Usenko, David Schubert, Nikolaus Demmel, and Daniel Cremers. Efficient derivative computation for cumulative b-splines on lie groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11148–11156, 2020.