# B 样条曲线及其导数的高效计算

许中兴
PLCT 实验室
智能软件研究中心
中国科学院软件研究所
xuzhongxing@iscas.ac.cn

## 1 背景

本文描述 Open Cascade[1] 对 NURBS 曲线求值和求导数的实现。代码的参考文献是 [2]. 这篇文章里第 19 节的递推公式上下标有问题。这个递推公式的来源是 [3]. 但是 [3] 里的关键公式 (1)(2) 直接从 [4] 得到，没有给出推导过程。然而 [4] 中给出的公式与 [3] 所使用的不太一样。此外，[3] 中的公式 2 有一个上标错误。所以，我在这里把完整的推导过程做一遍。

## 2 思路

虽然 B 样条曲线及其导数都有标准的公式定义 [5]，但是在实际 CAD 软件实现中，为了效率和计算稳定性的考虑，不会直接使用标准公式进行计算，而是通过对控制点进行递归计算，一次性得到一个节点区间上的曲线的 Taylor 展开形式。

整个过程的核心操作就是通过升高 order 和降低求导阶数来将最终要求的低 order 和高导数的控制点用高 order 和低导数的控制点表达出来，从而完成计算。因为我们最初给定的是高 order 低导数的控制点。

最终的计算由低 order 的 B 样条基函数表达是因为只有 order 为 1 的基函数才是常数。

# 3 术语定义

order: 曲线的阶数. degree: 曲线的次数。我们在这里只使用 order. $order = degree - 1$.

knot: 曲线的节点，在这篇文章里不显式表示出来。

# 4 符号定义

$u_i$: knots, 曲线的节点。

引入几个助记符变量:

$$\alpha_i = x - u_i$$
$$\beta_{i-r} = u_{i+k-r} - x$$
$$\gamma_i^r = u_{i+k-r} - u_i$$

$k$: 原始曲线的 order, 是固定的。

曲线的控制点有 3 个不同的维度。

- 和哪个 knot 相关
- 曲线可以用不同 order 的 B-spline 基函数来表示，不同 order 的基函数有不同的控制点。
- 曲线可以求不同次数的导数，也对应着不同的控制点。

注意，这 3 个维度是相互独立的。

$A_i^{r,t}$: 控制点，上下标的含义分别为，$i$ 表示这个控制点是和哪个 knot 相关联，$r$ 表示消去的 order 数。r 为 0 表示这个控制点是和原始的 k order B 样条基函数相关联的控制点。$t$ 表示求导的次数，t 为 0 表示原始曲线的控制点，t 为 1 表示求了 1 次导的曲线的控制点。

注意: 因为 $r$ 是消去的 order 数，所以为了降低 order 我们需要升高 $r$。

这里升高和降低是由最终的计算顺序决定的。推导公式的时候左右都是对称的，所以不用特别在意升高还是降低的说法。

# 5 基本公式

B-spline 基函数:

$$N_i^1 = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & otherwise \end{cases}$$

$$\begin{aligned} N_i^k &= \frac{u-u_i}{u_{i+k-1}-u_i}N_i^{k-1} + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}}N_{i+1}^{k-1} \\ &= \frac{\alpha_i}{\gamma_i^1}N_i^{k-1} + \frac{\beta_i}{\gamma_{i+1}^1}N_{i+1}^{k-1} \end{aligned}$$

B-spline 基函数的求导公式：

$$\begin{aligned} (N_i^k)' &= \frac{k-1}{u_{i+k-1}-u_i}N_i^{k-1} - \frac{k-1}{u_{i+k}-u_{i+1}}N_{i+1}^{k-1} \\ &= (k-1)(\frac{N_i^{k-1}}{\gamma_i^1} - \frac{N_{i+1}^{k-1}}{\gamma_{i+1}^1}) \end{aligned}$$

# 6 曲线的导数对应的控制点

设原始 k 阶 B 样条曲线公式为

$$C(x) = \sum_{i=0}^{n} A_i N_i^k$$

求和上下限为 0 到 $n$，为了简洁起见，我们后面都不写求和上下限。
$A_i$ 是原始的未求导未降阶的控制点，即可以写成 $A_i^{0,0}$
对曲线求一次导，得：

$$\begin{aligned} C(x)' &= \sum_i A_i^{0,0}(N_i^k)' \\ &= \sum_i A_i^{0,0}(k-1)(\frac{N_i^{k-1}}{\gamma_i^1} - \frac{N_{i+1}^{k-1}}{\gamma_{i+1}^1}) \\ &= \sum_i (k-1)(\frac{A_i^{0,0}}{\gamma_i^1} - \frac{A_{i-1}^{0,0}}{\gamma_i^1})N_i^{k-1} \\ &=: \sum_i (k-1)A_i^{1,1}N_i^{k-1} \end{aligned}$$

最后一行的 =: 表示 “定义为”。为什么可以定义为 $A_i^{1,1}$？因为控制点 $A_i^{1,1}$ 是对应于降了 1 阶的基函数，并对应于求了一次导的曲线。

从上面这个推导，我们可以得到第一个递推关系 (1)：

$$\gamma_i^{r+1} A_i^{r+1,t+1} = A_i^{r,t} - A_{i-1}^{r,t} \tag{1}$$

计算方向为从右向左计算，即用高阶低导的控制点计算出低阶高导的控制点。

# 7 不同阶的基函数的控制点之间的关系

这里我们应用一次降阶公式进行推导。

$$\begin{aligned}\sum_i A_i^{1,1} N_i^{k-1} &= \sum_i A_i^{1,1} (\frac{\alpha_i}{\gamma_i^2} N_i^{k-2} + \frac{\beta_{i-1}}{\gamma_{i+1}^2} N_{i+1}^{k-2}) \\ &= \sum_i (A_i^{1,1} \frac{\alpha_i}{\gamma_i^2} + A_{i-1}^{1,1} \frac{\beta_{i-2}}{\gamma_i^2}) N_i^{k-2} \\ &=: \sum_i A_i^{2,1} N_i^{k-2}\end{aligned}$$

我们得到第 2 个递推关系 (2):

$$\gamma_i^{r+1} A_i^{r+1,t} = \alpha_i A_i^{r,t} + \beta_{i-(r+1)} A_{i-1}^{r,t} \tag{2}$$

# 8 实际计算使用的递推关系

上面得到的 2 个递推关系是快速计算 B 样条曲线的关键。不过在计算的时候我们实际使用的递推关系还需要再推导一步。

把式 (1) 乘上 $\beta_{i-(r+1)}$ 和式 (2) 一起消去 $A_{i-1}^{r,t}$. 把式 (1) 乘上 $\alpha_i$ 和式 (2) 一起消去 $A_i^{r,t}$. 利用关系 $\alpha_i + \beta_{i-r} = \gamma_i^r$，可以得到 2 个新的递推关系：

$$\beta_{i-(r+1)} A_i^{r+1,t+1} = A_i^{r,t} - A_i^{r+1,t} \tag{3}$$

$$A_i^{r+1,t} = \alpha_i A_i^{r+1,t+1} + A_{i-1}^{r,t} \tag{4}$$

用于实际计算的是公式 (4), 计算方向从右向左，即利用高导的控制点，计算出低导的控制点，阶数不变。

# 9 实际计算过程

实际的计算分成 2 个阶段进行，第一个阶段，利用递推公式 (1)，得到阶数最低，求导次数最高的控制点。第二个阶段，利用递推公式 (4)，得到阶数最低，求导次数依次降低到 0 的一系列值。这 2 个计算阶段可以用 2 个三角形矩阵表示，详见 [3] 的图 3.

最终，我们得到曲线在 $x$ 处的全部 $k+1$ 个从 0 到 k 阶的导数值: $C(x), C^{(1)}(x), \ldots, C^{(k)}(x)$.

# 10 对多次计算的优化

在上面的计算过程中，一次完整的迭代可以得到关于曲线上一个点的全部阶数的导数值。我们利用曲线在节点处的 Taylor 展开公式，可以避免每次都进行迭代。具体过程如下。

我们知道，对于一个 n 次多项式，我们将它在某一个点进行 Taylor 展开，可以得到它的一个精确的表达式:

$$P(x) = T_0 + T_1(x-u) + \ldots + T_n(x-u)^n \tag{5}$$

其中，$T_i$ 是 $P(x)$ 在 $u$ 处的 $i$ 阶导数。

假设我们要计算区间 $[u_i, u_{i+1}]$ 里面的曲线值,我们只需要先计算一次曲线在 $u_i$ 处的各阶导数值,之后便可以利用5来计算所有的位于区间 $(u_i, u_{i+1})$ 里面的曲线值了。注: 实际计算中对于位于区间端点的点有可能需要做一些特殊处理，具体参见 Open Cascade 的实现。

注: 在计算公式5时，我们使用 Horner 方案，即实际上的计算顺序是: $T_0 + (x-u)[T_1 + (x-u)[T_2 + \ldots + (x-u)T_n]]$.

# 参考文献

[1] Open cascade. `https://www.opencascade.com/`.

[2] Wolfgang Boehm. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1:1–60, 1984.

[3] Wolfgang Boehm. Efficient evaluation of splines. *Computing*, 33:171–177, 1984.

[4] Carl de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6:50–62, 1972.

[5] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer, 1997.