② 101-103

有理 Bernstein 多项式及其各阶导数的一种有效算法*

TP391.72

张晓鹏

(西北大学数学系,710069,西安太白北路1号;31岁,男,讲师)

A

摘 要 给出了有理 Bernstein 多项式及其各阶导数的一种有效算法,具有乘除量小,内存少的优点,并与广义秦九韶算法等效。

关键词 有理伯恩斯坦多项式; 广义秦九韶算法; 递推算法

有理多项式, 导数, CAGD

分类号 O29;TP391.72

有理 Bernstein 多项式在计算机辅助几何设计中起着非常重要的作用,因为它具有良好的几何性质,也能表示各种常用曲线;其一般形式为[1,2]

$$X_n(u)=\sum_{j=0}^{n}b_j\cdot w_j\cdot B_j^n(t)\ /\ \sum_{j=0}^{n}w_j\cdot B_j^n(t)。\tag{1}$$

$u\in[u_0,u_1]$,$t=\dfrac{u-u_0}{u_1-u_0}$,其中 $B_j^n(t)=\dbinom{n}{j}\cdot t^j\cdot(1-t)^{n-j}$,$j=0,1,\cdots,n$ 是 Bernstein 基函数,b_j 为 Bernstein 系数,$w_j>0$ 为权,n 是 $X_n(u)$的次数。在许多应用中都需要这种多项式在任意一点 u 处的函数值及各阶导数值。

著名的广义秦九韶算法[3](广义 Hornor 格式[4])为多项式

$$S_n(u)=\sum_{j=0}^{n}a_j\cdot t^j。\tag{2}$$

$u\in[u_0,u_1]$,$t=\dfrac{u-u_0}{u_1-u_0}$的求值及求导数提供了最为有效的算法[2][3]:

```
(1) delta ← u1-u0;  t ← (u-u0)/delta;  f ← 1/delta;  c ← f
(2) FOR r←0 STEP 1 TO (n-1) DO
        FOR k ← (n-1) STEP -1 TO r DO
            a(k) ← a(k) + t * a(k+1)
(3) FOR r←1 STEP 1 TO n DO
        a(r) ← a(r) * f;
        f ← f * (r+1) * c。
```

最终存放在 $a(r)$单元的数值即为(2)的第 r 阶导数。该算法乘除总数为$(n^2+7n-2)/2$,内存单元的个数为 $n+7$。

文献5给出了 Bernstein 多项式的值及各阶导数值的有效算法,与对应的广义秦九韶算法等效。本文把结论推广到了有理 Bernstein 多项式。记 $g_j=w_j\cdot b_j(j=0,\cdots,n)$,$G_n(u)=\sum_{j=0}^{n}g_j\cdot B_j^n(t)$,$W_n(u)=$

* 收稿日期:1994-06-21

$\sum_{j=0}^{n} w_j \cdot B_j^n(t)$。在求得 $G_n(u)$ 和 $W_n(u)$ 各阶导数的条件下，应用乘积函数的高阶导数公式可递推求出 $X_n(u)$ 的各种导数值，具体关系如下：

$$\begin{cases} X_n^{(r)}(u) = [G_n^{(r)}(u) - \sum_{i=1}^{r} \binom{r}{i} \cdot W_n^{(i)}(u) \cdot X_n^{(r-i)}(u)] \,/\, W_n(u), \qquad (r > 0) \\ X_n^{(0)}(u) = X_n(u) = G_n(u) \,/\, W_n(u) \end{cases} \tag{3}$$

其中涉及的 $G_n^{(r)}(u)$，$W_n^{(i)}(u)$，$i,r=0,\cdots,n$ 均可由文献 5 中(5.1)～(5.4)式递推求得，固定 $r, W_n, W_n^{(1)}, \cdots, W_n^{(r)}, G_n, G_n^{(1)}, \cdots, G_n^{(r)}, X, X^{(1)}, \cdots, X^{(r-1)}$ 时，记 $a_0 = G^{(r)}, a_i = -W_n^{(i)} \cdot X^{(r-i)}, i=1,\cdots,r$，并定义递推公式

$$\begin{cases} a_{ij} = a_{i,j-1} + a_{i-1\,j-1}, \qquad i = 1,\cdots,r, \qquad j = 0,\cdots,i \\ a_{i,0} = a_i, \qquad i = 0,\cdots,r \end{cases} \tag{4}$$

可归纳出 $a_{i,j} = \sum_{k=0}^{j} \binom{j}{k} \cdot a_{k+i-j}, 0 \leqslant j \leqslant i \leqslant k$，那么，$a_{rr} = \sum_{i=0}^{r} \binom{r}{i} a_i$。公式(4)的应用免除了求 $\binom{r}{i}$ 的值及 $n(n+1)/2$ 次乘法。总结这些过程可写出求 $X_n(u)$ 各阶导数的算法如下：

已知：$u, u_0, u_1, w_0, w_1, \cdots, w_n, b_0, \cdots, b_n, m$

求：$X_n(u)$ 在 u 处的值及导数值

(1)基本赋值： delta ← $u1-u0$；t ← $(u-u0)$ /delta；f ← 1/delta；c ← f

(2)求向前差分值：

```
FOR r ← 1 STEP 1 TO n DO
    b(k) ← w(k) * b(k);
FOR r ← 1 STEP 1 TO n DO
    FOR k ← n STEP -1 TO r DO
        BEGIN
            b(k) ← b(k)-b(k-1);
            w(k) ← w(k)-w(k-1)
        END
```

(3)求 $G_n(u)$ 和 $W_n(u)$ 的各阶导数，记 $b(r)=G_n^{(r)}(u)$，$W(r)=w_n^{(r)}(u)$，并存放在单元 $\{b(r)\}_{r=0}^{n}$ 和 $\{w(r)\}_{r=0}^{n}$ 处。

```
FOR p ← 1 STEP 1 TO n DO
    FOR r ← 0 STEP 1 TO (n-p) DO
        BEGIN
            b(r) ← b(r)+t * b(r+1);
            w(r) ← w(r)+t * w(r+1);
        END
FOR r ← 1 STEP 1 TO n DO
    BEGIN
        f ← f * (n-r+1) * c;
        b(r) ← b(r) * f;
        w(r) ← w(r) * f
    END
```

(4)求 $X_n(u)$ 的次数不超过 m 的各阶导数，记 $X(r)=X_n^{(r)}(u), r=0,\cdots,m$

```
FOR r ← (n+1) STEP 1 TO m DO
    BEGIN
        b(r) ← 0;
```

```
            w(r) ← 0;
        END;
    x(0) ← b(0)/w(0);
    FOR r ← 1 STEP 1 TO m DO
        BEGIN
            a(0) ← b(r);
            FOR i ← 1 STEP 1 TO r DO
                a(i) ← -w(i) * X(r-i);
            FOR j ← 1 STEP 1 TO r DO
                FOR i ← 1 STEP 1 TO r DO
                    a(i) ← a(i)+a(i-1);
                    x(r) ← a(r)/w(0)
        END
```

注意，$X_n(u)$的高于n阶的导数一般不为 0，因而求导阶数为 0 到m。当$m=n$时，该算法乘除次数为$(3n^2+13n+10)/2$，内存单元数为$4n+12$。这已是这种求值的最低数目了，因为分子求导，分母求导，求转化式(3)都需必要的乘积，$w(r)$和$b(r)$的输入也需必要的内存，而$a(r)$的作用在于免除大量乘积运算。$X(r)$必须设立，因w,b,a数组都不能存$X_n(u)$的导数值。因此，此算法具有很快的运算速度。

最后来分析此算法的稳定性。令$c = \max\limits_{0\leqslant r,\ j,\ r+j\leqslant n}\{|\triangle^r g_j|,\ |\triangle^r w_j|\}$，$d = \max\{1,\ (\triangle_i)^{-n}\}$，$p = 2^n \cdot c \cdot d$，则有$\| g_n^{(r)}(u) \| \leqslant p$，$\| W_n^{(r)}(u) \| \leqslant p$。再令$w=1/\min\limits_{0\leqslant j\leqslant n}\{w_j\}$，$q=\max\{1,\ wp\}$，$S_r = \| X_n^{(r)}(u) \|$，则推得递推关系：$S_r \leqslant q \cdot [1 + \sum\limits_{i=0}^{r-1}\binom{r}{i} \cdot S_i]$，$r = 0,\cdots,m$；再归纳出不等式$S_r \leqslant 2^{r(r+1)/2} \cdot q^{r+1}$，于是，$\| X_n^{(r)}(u) \| \leqslant 2^{m(m+1)/2} \cdot q^{m+1}$，$r = 0,\cdots,m$。设$|u-\tilde{u}|<\varepsilon$，根据微分中值定理而得$\| X_n^{(r)}(u) - x^{(r)}(\tilde{u}) \| = \| X_n^{(r+1)}(\xi) \| \cdot |u - \tilde{u}| \leqslant 2^{(m+1)(m+2)/2} \cdot q^{m+2} \cdot \varepsilon$，$r = 0,\cdots,m$。因此，当$|u-\tilde{u}|<\varepsilon$足够小时，$X_n^{(r)}(u)$与$X_n^{(r)}(\tilde{u})$的差可充分小，所以此算法运行稳定。

参 考 文 献

1 Boehm W, Farin G, Kahmann J. A survey of curve and surface method in CAGD. Computer Aided Geometric Design, 1984, 1(1):1～60

2 Farin G. Curves and Surfaces for Computer Aided Geometric Design. Seccnd Edition. Boston: Academic Press Inc, 1990

3 李庆扬，王能超，易大义. 数值分析. 武汉：华中工学院出版社，1986

4 Schumaker L. Spline Functions: Basic Theory. New York: Wiley, 1981

5 张晓鹏，穆玉杰. *An efficient evaluation algorithm for Bézier curves*. 纯粹数学与应用数学，1994，10(增刊)：68～80

An Efficient Algorithm for Rational Bernstein Polynomials and Their All Derivatives

Zhang Xiaopeng

(Department of Mathematics, Northwest University, 710069, Xi'an)

Abstract This paper gives an efficient algorithm for rational Bernstein polynomials, which has few multiplications and divisions and costs little storage, as efficient as generalized Hornor's scheme.

Key words rational Bézier curves; generalized Hornor's scheme; recursion