Bernstein 多项式的移位-加算法

谷 峰

(浙江经济职业技术学院,浙江 杭州 310018)

摘 要：提出了一个基于 CODIC 的计算 Bernstein 多项式的移位-加算法。该算法可以在存在于许多领域的基本计算系统中实现。证明了算法的收敛性,给出了误差分析,做了数值实验,验证了算法的有效性和效率。

关键词：Bernstein 多项式;CORDIC;移位-加算法;基本计算系统

中图分类号：TP319 **文献标识码**：A **文章编号**：1674-7720(2010)17-0007-03

A shift-add algorithm for computing Bernstein polynomials

GU Feng

(Zhejiang Technology Institute of Economy, Hangzhou 310018, China)

Abstract: A shift-add algorithm based on coordinate rotation digital computer algorithm for computing Bernstein polynomials was presented in this paper. This algorithm can be implemented in basic computing system which exists in many areas. Convergence of the algorithm was proved. Error estimation was analyzed. A numerical experiment was carried out to validate algorithm's effectiveness and efficiency.

Key words: Bernstein polynomial; CORDIC; shift-add algorithm; basic computing system

Bernstein 多项式在数值计算领域起着重要的作用,在诸如全局优化[1]、构造 Bezier 曲线[2]等方面有着广泛使用。N 阶 Bernstein 多项式用迭代式表示为:$B_0^0(t)=1$,$B_i^k(t)=(1-t)B_i^{k-1}(t)+tB_{i-1}^{k-1}(t)$,其中 $k=1,2,\cdots,n$;$i=0,1,\cdots,k$。

在高级计算系统中,可以很容易地找到 Bernstein 多项式的算法[3]。例如,在 Mathematica 中,$B_i^n(t)$ 可以用 BernsteinBasis[n,i,t]计算。用高级语言编程计算 Bernstein 多项式也非常容易。本文讨论如何在基本计算系统(仅具备移位、加和逻辑运算功能的计算系统)中计算 Bernstein 多项式。基本计算系统存在于许多系统中,例如工业控制系统、军事应用系统、医疗应用系统等。典型的有单片机系统和 FPGA(Field Programmable Gate Arrays)等。

CORDIC 算法是可计算多种基本初等函数的移位-加算法[4-6]。参考文献[7-8]扩展了 CORDIC 算法,其收敛性和误差估计在参考文献[7]中做了分析。随着硬件技术的发展,这些快速统一移位-加算法可以用硬件实现,而且不需使用乘法器[9],成本较低,也可以用汇编语言编程实现。本文提出一个基于CORDIC 算法的 Bernstein 多项式移位-加算法。

1 算法的描述

参考文献[7]中定义了符号函数和正规序列:

$$sg(x)=\begin{cases}1 & x\geqslant 0\\ -1 & x<0\end{cases},\ \{\delta_i\}_0^\infty=\{2^{-i}\}_0^\infty$$

设 ε 为误差上界。Bernstein 多项式的移位-加算法包含一个主程序和一个子程序:

Subprogram UV(u,v,ε):

if $u=0$ or $v=0$ then result:=0. Stop.

$s:=1$. if $u<0$ then $\{s:=-s;u:=-u.\}$. if $v<0$ then $\{s:=-s;v:=-v.\}$.

$m:=0$; while $u>1$ do $\{u:=\frac{u}{2};m:=m+1.\}$

$N:=2;\varepsilon:=2^{-m}\times\varepsilon$; while $\delta_{N-2}>\varepsilon$ do $N:=N+1$;

$i:=1;x_1:=u,y_1:=v,z_1:=0$.

while $i<N$

do $\{s_i:=sg(x_i);x_{i+1}:=x_i+s_i\times\delta_i;z_{i+1}:=z_i-s_i\times\delta_i\times y_1;i:=i+1;\}$

$z_N:=s\times 2^m\times z_N$. result:=$z_N$. Stop.

MainProgram Bernstein(u,v,ε):

$B_0^0:=1$; for $i:=1$ to n do $\{B_{-1}^i:=0;B_{i+1}^i:=0;\}$;

$\varepsilon_1:=\varepsilon\times 2^{-1}$; while $UV(\varepsilon_1,n,\varepsilon\times 2^{-1})>\varepsilon$ do $\varepsilon_1:=\varepsilon_1\times 2^{-1}$;

for $i:=1$ to n

for $j:=0$ to i $B_j^i(t):=UV(1-t,B_j^{i-1}(t),\varepsilon_1)+UV(t,B_{j-1}^{i-1}(t),\varepsilon_1)$;

Output $B_j^n(t),j=0,\cdots,n$. Stop.

注:(1)算法中仅使用了移位运算(即 $2^{-i}\times t$)和加法运算。

(2)迭代次数 N 根据后面的定理 1 确定。

(3)主程序中的 $\varepsilon_1\leqslant\dfrac{\varepsilon}{2n}$。

(4)计算实践表明子程序运行的迭代次数一般不超过 28 步,因此是个快速算法。

2 算法的收敛性和误差分析

设 $x\approx F_N(x,\delta)=\sum_{i=0}^{N}sg(x_i)\delta_i$ 是 x 用 $\{\delta_i\}_0^{\infty}$ 的度量展开[7],由参考文献[7]知迭代过程

$$\begin{cases}x_0=0,z_0=0\\ s_i=sg(x-x_i),x_{i+1}=x_i+s_i\times 2^{-i},z_{i+1}=z_i-s_i\times 2^{-i}\times y\end{cases}\quad i=1,2\cdots \tag{1}$$

当 n 充分大时,有 $x_{n+1}\approx x,z_{n+1}\approx xy$。

subprogram $UV(u,v,\varepsilon)$即基于迭代过程(1)。

定理 1 设 $\{\delta_i\}_0^{+\infty}=\{2^{-i}\}_0^{+\infty}$。若 $x\in\left[-\sum_{i=0}^{\infty}\delta_i,\sum_{i=0}^{\infty}\delta_i\right]=[-2,2]$,则:

(1)迭代过程(1)中的 $\{x_i\}$ 收敛于 x,且有 $|x-x_N|\leqslant\delta_{N-1}$。

(2)迭代过程(1)中的 $\{z_i\}$ 收敛于 xy,且有误差估计 $|z_N-xy|\leqslant|y|(1+|x|)(1+\delta_N)\delta_N$。

(3)subprogram $UV(u,v,\varepsilon)$ 中的 $\{z_i\}$ 收敛于 xy,其计算误差不大于 ε。

证明:(1)由迭代过程(1)可知 $x_N-x_{N+k}=\sum_{j=N}^{N+k-1}S_j\delta_j$。由参考文献[7]中的引理 1,有 $|x_N-x_{N+k}|=\sum_{j=N}^{N+k-1}\delta_j\leqslant\sum_{j=N}^{\infty}\delta_j\leqslant\delta_{N-1}$,所以 $\{x_i\}$ 是 Cauchy 序列。由算法的设计知,必有 $x_i\to x$ $(i\to\infty)$。令 $k\to\infty$,得 $|x-x_N|\leqslant\delta_{N-1}$。

(2)设 x 用 $\{\delta_i\}_0^{\infty}$ 做的度量展开为 $x=\sum_{i=0}^{\infty}s_i\delta_i$[4],且 $\|(a_{ij})_{m\times n}\|_{\infty}=\max\limits_{1\leqslant i\leqslant n}\sum_{j=1}^{N}|a_{ij}|$。

对 $A=\begin{pmatrix}0&0\\1&0\end{pmatrix}$ 有:$\|A\|_{\infty}=1$,$\|e^{Ax}\|_{\infty}=\left\|\begin{pmatrix}1&0\\x&1\end{pmatrix}\right\|_{\infty}=1+|x|$。由 $N>0$ 及参考文献[3]引理 1,有 $|z_N-xy|\leqslant\left\|e^{Ax}\begin{pmatrix}y\\0\end{pmatrix}-e^{A\sum_{i=0}^{N}s_i\delta_i}\begin{pmatrix}y\\0\end{pmatrix}\right\|_{\infty}\leqslant\left\|\begin{pmatrix}y\\0\end{pmatrix}\right\|_{\infty}\|e^{Ax}\|_{\infty}\|A\|_{\infty}\left|\sum_{i=N+1}^{+\infty}s_i\delta_i\right|\cdot\|e^{\left|A\sum_{i=N+1}^{+\infty}s_i\delta_i\right|}\|_{\infty}=|y|(1+|x|)\cdot\sum_{i=N+1}^{+\infty}\delta_i\cdot\left(1+\left|\sum_{i=N+1}^{+\infty}s_i\delta_i\right|\right)\leqslant|y|(1+|x|)\cdot(1+\delta_N)\cdot\delta_N<|y|(1+|x|)\cdot\delta_{N-1}$。

(3)当 $u=0$ 或 $v=0$ 时,程序输出结果 $uv=0$。当 $uv\neq 0$ 时,u 和 v 在 subprogram $UV(u,v,\varepsilon)$ 中预处理为 $uv=s\times 2^m\times u_1v_1$。这里 s 为 1 或 -1。u_1 和 v_1 在 $(0,1]$ 中。由(2),计算结果 z'_N 有 $|z'_N-u_1v_1|<|u_1|(1+|v_1|)\cdot\delta_{N-1}\leqslant 2\delta_{N-1}<\delta_{N-2}$;迭代次数 N 满足 $\delta_{N-2}\leqslant 2^{-m}\times\varepsilon$。所以最终结果 $z_N=s\times 2^m\times z'_N$ 满足 $|z_N-uv|<2^m\times\delta_{N-2}\leqslant\varepsilon$。证毕。

定理 2 主程序 $Bernstein(n,t,\varepsilon)$ 的输出值 $B_j^n(t)(j=0,\cdots,n)$ 是 n 次 Bernstein 多项式的计算值。计算误差上界是 ε。

证明:归纳证明。当 $n=1$ 时,$B_{-1}^0(t)=0$,$B_0^0(t)=1$,$B_1^0(t)=0$,$B_{-1}^1(t)=0$,$B_0^1(t)=UV(1-t,B_0^0(t),\varepsilon_1)+UV(t,B_{-1}^0(t),\varepsilon_1)$,$B_1^1(t)=UV(1-t,B_1^0(t),\varepsilon_1)+UV(t,B_0^0(t),\varepsilon_1)$,$B_2^1(t)=0$。由程序知 $\varepsilon_1\leqslant\dfrac{\varepsilon}{2n}$,因此 $B_0^1(t)$ 和 $B_1^1(t)$ 的计算误差不大于 $2\times\varepsilon_1\leqslant\dfrac{\varepsilon}{n}$。

归纳假设 $B_j^{i-1}(t)(j=0,\cdots,i-1)$ 的计算误差不大于 $\dfrac{(i-1)\varepsilon}{n}$。

设 Bernstein 多项式的实际函数值是 $b_j^i(t)$,$UV(1-t,B_j^{i-1}(t),\varepsilon_1)$ 的计算误差是 ε_j,则 $|B_j^i(t)-b_j^i(t)|=|UV(1-t,B_j^{i-1}(t),\varepsilon_1)+UV(t,B_{j-1}^{i-1}(t),\varepsilon_1)-(1-t)b_j^{i-1}(t)-tb_{j-1}^{i-1}(t)|\leqslant(1-t)|B_j^{i-1}(t)-b_j^{i-1}(t)|+t|B_{j-1}^{i-1}(t)-b_{j-1}^{i-1}(t)|+2\times\dfrac{\varepsilon}{2n}\leqslant(1-t)\dfrac{(i-1)\varepsilon}{n}+t\dfrac{(i-1)\varepsilon}{n}+\dfrac{\varepsilon}{n}=\dfrac{i\varepsilon}{n}$ $j=0,1,\cdots,i$。即 $B_j^n(t)(j=0,1,\cdots,n)$ 的计算误差上界是 ε。定理得证。

3 数值实验

给定误差上界 $\varepsilon=0.000\ 000\ 5$,用本文给出的算法计算三次 Bernstein 多项式 $B_j^3(t)(j=0,1,2,3)$ 的数值。计算值见表 1。subprogram $UV(u,v,\varepsilon)$ 中的迭代次数不大于 28。三次 Bernstein 多项式的函数值列表如表 2 所示。由此可见计算误差符合要求。

参考文献

[1] NATARAJ P S V, AROUNASSALAME M. A new subdivision algorithm for the bernstein polynomial approach to global optimization [J]. International Journal of Automation and Computing, 2007,4(4):342-352.

表 1 三次 Bernstein 多项式的计算值

t	Bernstein 多项式			
	$B_0^3(t)$	$B_1^3(t)$	$B_2^3(t)$	$B_3^3(t)$
0.0	1.000 000 007 5	0	0	0
0.1	0.729 000 005 4	0.242 999 995 8	0.026 999 998 9	0.000 999 999 9
0.2	0.512 000 001 4	0.383 999 999 3	0.095 999 999 4	0.007 999 999 9
0.3	0.342 999 998 9	0.441 000 000 2	0.189 000 000 7	0.027 000 000 2
0.4	0.215 999 997 6	0.431 999 999 2	0.288 000 002 1	0.064 000 001 1
0.5	0.125 000 027 9	0.375 000 008 4	0.375 000 008 4	0.125 000 002 8
0.6	0.064 000 001 1	0.288 000 002 1	0.431 999 999 2	0.215 999 997 6
0.7	0.027 000 000 2	0.189 000 000 7	0.441 000 000 2	0.342 999 998 9
0.8	0.007 999 999 9	0.095 999 999 4	0.383 999 999 3	0.512 000 001 4
0.9	0.000 999 999 9	0.026 999 998 9	0.242 999 995 8	0.729 000 005 4
1.0	0	0	0	1.000 000 007 5

表 2 三次 Bernstein 多项式的函数值

t	Bernstein 多项式			
	$B_0^3(t)$	$B_1^3(t)$	$B_2^3(t)$	$B_3^3(t)$
0.0	1	0	0	0
0.1	0.729	0.243	0.027	0.001
0.2	0.512	0.384	0.096	0.008
0.3	0.343	0.441	0.189	0.027
0.4	0.216	0.432	0.288	0.064
0.5	0.125	0.375	0.375	0.125
0.6	0.064	0.288	0.432	0.216
0.7	0.027	0.189	0.441	0.343
0.8	0.008	0.096	0.384	0.512
0.9	0.001	0.027	0.243	0.729
1.0	0	0	0	1

[2] FARIN G. Curves and surfaces for computer-aided geometric design: a practical guide, 4th Ed. Academic Press, San Diego, 1997.

[3] FENG Jieqing, PENG Qunsheng. Fast algorithm for composition of the bernstein polynomials [J]. Journal of Computer-Aided Design & Computer Graophics, 2001,13(2).

[4] VOLDER J E. The CORDIC computing technique [J]. IRE Transactions on Electronic Computers, 1959,8(9):330-334.

[5] MULLER J M. Elementary functions, algorithms and implementation. Birkhauser Boston, 1st edition,1997. 2nd edition, 2006:133-156.

[6] EKLUND N. CORDIC: elementary function computation using recursive sequences [C]. International Conference on Technology, 1998.

[7] GU Feng. Convergence and error estimation of coordinate rotating algorithm and its expansion [J]. Chinese Journal of Numerical Mathematics and Applications, 2006,28(2):1-9.

[8] HU Xiaobo, HARBER R, BASS S. Expanding the range of convergence of the CORDIC algorithm[J]. IEEE Transactions on Computers, 1991,40(1):13-21.

[9] ANDRAKA R. A survey of CORDIC algorithms for FPGA based computers[C]. In Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays(FPGA) 1998:191-200.

(收稿日期:2010-04-10)

作者简介:

谷峰,男,1959 年生,理学硕士,主要研究方向:数值逼近、算法设计。