国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第15卷第4期 1993年12月 Vol. 15 No. 4

点数任意时的快速插值算法*

成 礼 智
（系统工程与数学系）

摘 要 本文建立了运算量级为 $O(n\log_2 m)$ 的多项式快速除法（其中，m，n 分别为除式与被除式的多项式次数），把点数 $n+1$ 为 2 的幂次的多项式快速插值推广到 $n+1$ 为任意数情形，提出了运算量级为 $O(n\log_2^2 n)$ 的快速插值算法。

关键词 快速算法，FFT，多项式插值，快速多项式除法
分类号 O241.3

多项式插值是数值分析中一个很基本的问题，其应用范围非常广泛。用拉格朗日公式进行插值是最古老也是最典型的算法。直接计算 $n+1$ 点的拉格朗日公式，运算量级为 $O(n^2)$. 1973 年，H. T. Kung 对 $n=2^t-1$ 的情形提出了运算量级为 $O(n\log_2^2 n)$ 的快速插值算法. 文献[2]利用模多项式运算对 $n=2^t-1$ 的情形得到同一量级的快速算法.

由于 2^t 是一个高度复合数，规模为 2^t 的科学计算问题常常可采用递推减半技术化为两个规模为 2^{t-1} 的同一类问题来计算，一直递推下去便可得到快速算法，当 n 为任意自然数时，有时可采用补零增广技术转化为规模 2^t 来计算，如多项式乘积、离散卷积及矩阵乘法等，具体方法可参见文献[3]、[4]，但对任意点多项式插值问题，补零增广方法不再有效. 本文首先研究了一般多项式的快速除法，利用多项式快速除法以及对已有的快速多项式插值进行改进得到了任意 $n+1$ 点多项式插值的快速算法，运算量级为 $O(n\log_2^2 n)$.

1 多项式的快速除法

考虑多项式

$$P(x)=a_n x^n+a_{n-1}x^{n-1}+\cdots+a_1 x+a_0$$

$$Q(x)=b_m x^m+b_{m-1}x^{m-1}+\cdots+b_1 x+b_0$$

$P(x)$ 被 $Q(x)$ 所除的商 $q(x)$ 记为 $q(x)=\left[\frac{P(x)}{Q(x)}\right]$，余式 $R(x)=P(x)-Q(x)q(x)$. 其中，$0\leqslant \deg R(x)<m$.

当 $m=2^t-1$，$n=2m$ 时，用文献[5]中三角 Toeplitz 矩阵求逆的快速算法或用文献

* 1992 年 4 月 6 日收稿

2]中的快速多项式乘法在$O(n\log_2 n)$时间步内可求出$\left[\frac{x^{2m}}{Q(x)}\right]$，当$n$、$m$为$n>m$的任意数时，文献[4]提供了运算量级为$O(n\log_2^2 m)$的求$q(x)$、$R(x)$的快速算法。

下面我们建立n，m为任意数时求$q(x)$运算量级为$O(n\log_2 m)$的快速算法。首先考虑$m=2^t-1$，$n=2m$时$\left[\frac{x^{2m}}{Q(x)}\right]$的计算。

算法 1.1 计算$\left[\frac{x^{2m}}{Q(x)}\right]$，记为 Recip($Q(x)$)．其中$m=2^t-1$.

(1)设$b_0\neq 0$，Recip(b_0)$=\frac{1}{b_0}$.

(2)对于$j=1$到t递推地作：

$$q_{1j}(x)=\text{Recip}\left(\sum_{i=2^{j-1}}^{2^j-1}b_i x^{i-2^{j-1}}\right)$$

$$r_j(x)=2q_{1j}(x)x^{(3\cdot 2)2^j-2}-(q_{1j}(x))^2\left(\sum_{i=0}^{2^j-1}a_i x_i\right)$$

$$q_j(x)=\left[\frac{r(x)}{x^{2^j-2}}\right]=\text{Recip}\left(\sum_{i=0}^{2^j-1}b_i x^i\right)$$

(3) Recip (Q (x)) $=q_t$ (x)

算法 1.1 的正确性可参见文献[2]，上述算法中，一个2^t-1点的 Recip($Q(x)$)计算化为一个$2^{t-1}-1$点的 Recip($Q_1(x)$)$=q_{1t}(x)$的计算以及多项式乘法来进行，因此若用 FFT 快速计算多项式乘法，算法 1.1 的运算量为$O(n\log_2 m)$.

当n、m为一般值时，为简单计，这里只考虑$m>n>2m$的情形，n为其它情况时，做法类似。无妨设m不是 2 的幂，则有自然数t使$2^{t-1}<m<2^t$，取$m_1=2^t-m-1$．我们通过下面的定理来计算$\left[\frac{P(x)}{Q(x)}\right]$.

定理 1.1 设$\deg Q(x)=m$，$\deg P(x)=n$，m_1，t的意义如前所述，则

$$\left[\frac{P(x)}{Q(x)}\right]=\left[\frac{P(x)\left[\frac{x^{2^{t-1}-2-m_1}}{Q(x)}\right]}{x^{2^{t-1}-2-m_1}}\right]$$

证

设 $$P(x)=Q(x)\left[\frac{P(x)}{Q(x)}\right]+R_1(x),\qquad 0\leqslant\deg R_1(x)<m$$

$$x^{2^{t-1}-2-m_1}=Q(x)\left[\frac{x^{2^{t-1}-2-m_1}}{Q(x)}\right]+R_2(x),\qquad 0\leqslant\deg R_2(x)<m$$

$$P(x)\left[\frac{x^{2^{t-1}-2-m_1}}{Q(x)}\right]=x^{2^{t-1}-2-m_1}\left[\frac{P(x)\left[\frac{x^{2^{t-1}-2-m_1}}{Q(x)}\right]}{x^{2^{t-1}-2-m_1}}\right]+R_3(x),$$

$$0\leqslant\deg R_3(x)<m$$

于是

$$\frac{P(x)}{Q(x)}=\frac{P(x)\left[x^{2^{t-1}-2-m_1}/Q(x)\right]}{x^{2^{t-1}-2-m_1}}=\frac{P(x)\left(\left[\frac{x^{2^{t-1}-2-m_1}}{Q(x)}\right]+\frac{R_2(x)}{Q(x)}\right)}{x^{2^{t-1}-2-m_1}}$$

即

$$\frac{R_1(x)}{Q(x)}+\left[\frac{P(x)}{Q(x)}\right]=\left[\frac{P(x)\left[\frac{x^{2^{t+1}-2-m_1}}{Q(x)}\right]}{x^{2^{t+1}-2-m_1}}\right]+\frac{R_2(x)+P(x)R_2(x)/Q(x)}{x^{2^{t+1}-2-m_1}}$$

整理得

$$\left[\frac{P(x)}{Q(x)}\right]-\left[\frac{P(x)\left[\frac{x^{2^{t+1}-2-m_1}}{Q(x)}\right]}{x^{2^{t+1}-2-m_1}}\right]=\frac{R_3(x)Q(x)+R_2(x)P(x)-R_1(x)x^{2^{t+1}-2-m_1}}{Q(x)x^{2^{t+1}-2-m_1}}$$

上式左边是一个整多项式。由于

$$\begin{aligned}\deg R_3(x)Q(x)&<2m<m+2^{t+1}-2-m_1\\ \deg R_2(x)P(x)&<m+n<m+2m=m+(2m+m_1)-m_1\\&<m+2(n+m_1)-m_1\\&=m+2^{t+1}-2-m_1\\ \deg R_1(x)x^{2^{t+1}-2-m_1}&<m+2^{t+1}-2-m_1\end{aligned}$$

上面三个不等式表明等式右端分子多项式的次数小于分母多项式的次数，这只能是

$$R_3(x)Q(x)+R_2(x)P(x)-R_1(x)x^{2^{t+1}-2-m_1}=0$$

即有

$$\left[\frac{P(x)}{Q(x)}\right]=\left[\frac{P(x)\left[\frac{x^{2^{t+1}-2-m_1}}{Q(x)}\right]}{x^{2^{t+1}-2-m_1}}\right]$$

（证毕）

由算法 1.1 与定理 1.2，我们得到$\left[\frac{P(x)}{Q(x)}\right]$的算法如下。

算法 1.2 计算$\left[\frac{P(x)}{Q(x)}\right]$，其中 $m<n<2m$.

(1) 用算法 1.1 求$\left[\frac{x^{2^{t+1}-2}}{x^{m_1}Q(x)}\right]=q_1(x)$

(2) 快速计算多项式乘积 $S(x)=q_1(x)P(x)$

(3) 截断得$\left[\frac{P(x)}{Q(x)}\right]=\left[\frac{S(x)}{x^{2^{t+1}-2-m_1}}\right]$

上述算法中，第一步用算法 1.1 求 $q_1(x)$，运算量为 $O(n\log_2 n)$，第二步用 FFT 计算多项式乘积 $S(x)$，运算量为 $O(n\log_2 n)$，第三步的截断只需作简单的移位操作，不需运算量，故算法 1.2 总的运算量为 $O(n\log_2 n)$.

算法 1.2 是文献[1]、[2]中算法的推广。n，m 为一般自然数时，比文献[4]中算法要低一个数量级。

2 任意点多项式插值的快速算法

已知点对(x_i,y_i)，$i=0,1,2,\cdots,n$，其中 $x_i(i=0,1,\cdots,n)$互不相等。所谓多项式插值就是求满足 $p(x_i)=y_i, i=0,1,\cdots,n$ 的 n 次多项式 $p(x)$. 当 $n+1$ 为 2 的幂时，文献[2]、[4]中已给出了 $O(n\log_2^2 n)$的快速算法，因此本文只讨论 $n+1$ 不为 2 的幂的情况。并假定

$2^{t-1}<n+1<2^t(t=[\log_2(n+1)]+1)$.

任取 x_{n+1}, x_{n+2}, …, x_{2^t-1}互不相同且均异于 x_i, $i=0,1,2,\cdots,n$. 作多项式 $g(x)=\prod_{i=n+1}^{2^t-1}(x-x_i)$. $g(x)$可按如下的“T”格式计算：

$$\overbrace{\quad\quad\quad\quad\quad\quad\quad\quad}^{n\text{ 个}}$$

(1) $x-x_{n+1}$ $x-x_{n+2}$ $x-x_{n+3}$ $x-x_{n+4}\cdots x-x_{2^t-1}$ 1 1…1 1

(2) $(x-x_{n+1})(x-x_{n+2})$ $(x-x_{n+3})(x-x_{n+4})$ … 1

(3) $\prod_{i=1}^{4}(x-x_{n+i})$ $\prod_{i=5}^{8}(x-x_{n+i})$ … 1

⋮ …

j, $\prod_{i=1}^{2^{j-1}}(x-x_{n+i})$ $\prod_{i=2^{j-1}+1}^{2^j}(x-x_{n+i})$ …

t, $\prod_{i=1}^{2^t-1-n}(x-x_{n+i})=g(x)$

用上述的“T”格式计算 $g(x)$需 t 步才能完成，每步作一些多项式乘积，每个多项式乘积可用两个正 FFT 与一个逆 FFT 来计算，类似于文献[3]中方法可推得计算 $g(x)$总的运算量为 $O(n\log_2^2 n)$. 设

$$g(x)=\sum_{i=0}^{2^t-1-n}d_i x^i=\sum_{i=0}^{n}d_i x^i$$

其中，$d_{2^t-n+1}=d_{2^t-n+2}=\cdots=d_n=0$. 用文献[4]中方法作多项式快速求值得到 $g(x_i)=c_i$, $i=0,1,2,\cdots,n$. 所需运算量为 $O(n\log_2^2 n)$. 由 c_i 以及文献[4]中得到的 2 的幂次时多项式快速插值算法，求通过点对(x_0, c_0y_0), (x_1, c_1y_1), …, (x_n, c_ny_n), $(x_{n+1}, 0)$, …, $(x_{2^t-1}, 0)$的插值多项式 $f(x)$. 求 $f(x)$的运算量为 $O(t^2\cdot 2^t)=O(n\log_2^2 n)$. 由于 $n<i\leqslant 2^t-1$ 时 $f(x_i)=0$. 故

$$f(x)=\Big(\prod_{i=n+1}^{2^t-1}(x-x_i)\Big)p(x)=g(x)p(x)$$

由上式知

$$p(x)=\left[\frac{f(x)}{g(x)}\right]$$

由算法 1.2 求 $p(x)$所需的运算量为 $O(n\log_2 n)$. 注意到 $0\leqslant i\leqslant n$ 时 $p(x_i)=f(x_i)/g(x_i)=c_iy_i/c_i=y_i$. 故 $p(x)$是通过点对$(x_i,y_i)(0\leqslant i\leqslant n)$的 n 次多项式，由插值多项式的唯一性知 $p(x)$为所求的插值多项式。

综上所述，任意点多项式的快速插值可描述为如下算法。

算法 2.1 （任意点的快速多项式插值）

设 $2^{t-1}\leqslant n+1<2^t$，求通过点$(x_i, y_i)(i=0,1,2,\cdots,n)$的插值多项式 $p(x)$.

(1) 若 $n+1$ 为 2 的幂，采用文献[4]中方法求 $p(x)$. 否则转到 2，

(2) 任取互不相同的 $x_i(n+1\leqslant i\leqslant 2^t-1)$异于 $x_j(0\leqslant j\leqslant n)$，用“$T$”格式快速计算

$$g(x)=\prod_{i=n+1}^{2^t-1}(x-x_i)$$

(3) 按文献[4]中方法快速求多项式 $g(x)$的值

$$g(x_i)=c_i, i=0,1,2,\cdots,n.$$

(4) 对于 $0\leqslant i\leqslant n$，计算 $u_i=c_iy_i$.

(5) 采用文献[4]中方法计算通过点(x_i, u_i)，$(0\leqslant i\leqslant n)$，$(x_i, 0)(n+1\leqslant i\leqslant 2^t-1)$的插值多项式 $f(x)$.

(6) 按算法 1.2 作快速多项式除法

$$p(x)=\left[\frac{f(x)}{g(x)}\right]$$

则 $p(x)$为所求插值多项式。

根据上面的分析知，对于 $n+1$ 为任意数情形，算法 2.1 提供了运算量级为 $O(n\log_2^2 n)$的快速多项式插值算法。

参 考 文 献

1 Kung H T. Fast evlation and interpolation. AD 755451, 1973

2 AHO, Hopcroft, Ullman. The Design and Analysis of Computer Algorithms. Addision—Wesley Publishing Company, 1976

3 蒋增荣，曾泳泓. 多项式变换及其应用，长沙：国防科技大学出版社，1989

4 游兆永. 线性代数与多项式的快速算法，上海科技出版社，1980

5 李磊. 三角 Toeplitz 系统的快速算法。计算数学，1987，(3)

Fast Interpolation for a Polynomial Through some Points

Cheng Lizhi

(Department of System Engineering and Mathematics)

Abstract

At first, a fast polynomial division algorithm is developed in this paper at O(n $\log_2$m) times, where , n, m are degrees of dividend and divisior polynomial, respectively, We then discuss a fast interpolation algorithm through n+1 points that extends n+1 with power of two to any number, the running times is O(n $\log_2^2$n).

Key words fast algorithm, fast Fourier transform, polynomial interpolation, fast polynomial division