国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第13卷第3期 1991年9月 Vol. 13 No. 3

多元多项式乘积的 FPT 算法

田泽荣 蒋增荣
（计算机系） （系统工程与应用数学系）

摘 要 本文详细讨论了多元多项式乘积的多项式变换（FPT）算法。首先给出了二元的情况，然后推广到了一般多元多项式，最后给出了这种算法在计算二维循环卷积中的应用，由此可见，这种算法在计算多维卷积和多维 DFT 时是很有效的。

关键词 卷积，多项式变换，多元多项式
分类号 TP319

近几年来，由于快速算法和大规模集成电路的发展，使得数字信号处理技术迅速发展，其应用日益广泛。卷积和离散富里叶变换在数字信号处理中起着重要作用，自从 1965 年由 Tukey 提出 FFT 后，各种有关 DFT 和卷积的快速算法竞相出现；1978 年，H. J. Nussbaumer 提出了有理数域上的多项式变换，这一理论很快得到发展并逐步完善，特别是它在计算 DFT 和卷积中的应用更为引人注目。本文就是利用这一有力工具来计算多元多项式乘积，为计算多维 DFT 和多维卷积开辟了一条新途径。

1 二元多项式乘积的 FPT 算法

1.1 算法

设 $A(z_1,z_2)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}a(m,n)z_1^m\cdot z_2^n$， $B(z_1,z_2)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}b(m,n)z_1^m\cdot z_2^n$

其中，$\{a(m,n)\}$，$\{b(m,n)\}$为二维复序列，$M=2^{t_1}$，$N=2^{t_2}$，$t_2\geqslant t_1$.

令 $C(z_1,z_2)\equiv A(z_1,z_2)B(z_1,z_2)\bmod(z_1^M+1),(z_2^N+1)$

下面给出 $c(z_1,z_2)$的算法

设 $c(z_1,z_2)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}c(m,n)z_1^mz_2^n$， $A_m(z_2)=\sum_{n=0}^{N-1}a(m,n)z_2^n$

$B_m(z^2)=\sum_{n=0}^{N-1}b(m,n)z_2^n$， $C_m(z_2)=\sum_{n=0}^{N-1}C(m,n)z_2^n$

则 $A(z_1,z_2)B(z_1,z_2)=\sum_{P=0}^{2M-2}\sum_{n=0}^{M-1}A_{p-n}(z_2)B_n(z_2)z_1^p$

其中，$0\leqslant p-n\leqslant M-1$.

* 1990 年 3 月 9 日收稿

故
$$A(z_1,z_2)B(z_1,z_2)\bmod(z_1^M+1)$$
$$\equiv\sum_{P=0}^{M-1}\sum_{n=0}^{M-1}A_{p-n}(z_2)B_n(z_2)z_1^p-\sum_{p=0}^{M-2}\sum_{n=0}^{M-1}A_{p+M-n}(Z_2)B_n(Z_2)Z_1^p$$

式中，$0\leqslant p-n\leqslant M-1$，$0\leqslant P+M-n\leqslant M+1$.

所以有
$$C_{M-1}(z_2)\equiv\sum_{n=0}^{M-1}A_{M-1-n}(z_2)B_n(z_2)\bmod(z_2^N+1)$$
$$C_P(z_2)\equiv\sum_{n=0}^{M-1}[A_{p-n}(z_2)-A_{p+M-n}(z_2)]B_n(z_2)\bmod(z_2^N+1)$$

式中，$0\leqslant P\leqslant M-2$，$0\leqslant P-n\leqslant M-1$，$0\leqslant p+M-n\leqslant M-1$

这可由以下非循环卷积求得：
$$C'_p(z_2)\equiv\sum_{n=0}^{M-1}A_{p-n}(z_2)B_n(z_2)\bmod(z_2^N+1)$$

其中，$0\leqslant p-n\leqslant M-1$，　$0\leqslant p\leqslant 2M-2$. 且 $0\leqslant p\leqslant M-2$ 时
$$C_p(z_2)=C'_p(z_2)-C'_{p+M}(z_2),C_{M-1}(z_2)=C'_{M-1}(z_2)$$

而 $C'_p(z_2)$可化成长度为 $2M$ 的循环卷积[3].

即令
$$A'_n(z_2)=\begin{cases}A_n(z_2) & n=0,1\cdots,M-1\\ 0 & n=M,M+1,\cdots,2M-1\end{cases}$$
$$B'_n(z_2)=\begin{cases}B_n(z_2) & n=0,1,\cdots,M-1\\ 0 & n=M,M+1,\cdots,2M-1\end{cases}$$

则有
$$C'_p(z_2)\equiv\sum_{n=0}^{2M-1}A'_n(z_2)B'_{(p-n)_{2M}}(z_2)\bmod(z_2^N+1)\qquad(0\leqslant p\leqslant 2M-1)$$

上式可用 FPT$(z_2^N+1,z_2^{N/M},2M)$计算。

1.2 运算量

记 $M_u(M,N)$，$A_d(M,N)$分别为所需复乘和复加数，则

$M_u(M,N)=2M\cdot M_u(N)$；$A_d(M,N)=2M\cdot A_d(N)+4MN\log_2(2M)+(M-1)N$

其中，$M_u(M,N)$，$A_d(N)$是二个一元多项式相乘所需的乘法量和加法量[2].

2　多元多项式乘积的 FPT 算法

设
$$,A(z_1,z_2,\cdots,z_k)=\sum_{n_1=0}^{N_1-1}\cdots\sum_{n_k=0}^{N_k-1}a(n_1,\cdots,n_k)z_1^{n_1}\cdots z_k^{n_k}$$
$$B(z_1,z_2,\cdots,z_k)=\sum_{n_1=0}^{N_1-1}\cdots\sum_{n_k=0}^{N_k-1}b(n_1,\cdots,n_k)z_1^{n_1}\cdots z_k^{n_k}$$
$$C(z_1,z_2,\cdots,z_k)=\sum_{n_1=0}^{N_1-1}\cdots\sum_{n_k=0}^{N_k-1}C(n_1,\cdots,n_k)z_1^{n_1}\cdots z_k^{n_k}$$
$$C(z_1,\cdots,z_k)\equiv A(z_1,\cdots,z_k)B(z_1,\cdots,z_k)\bmod(z_1^{N_1}+1),\cdots,(z_k^{N_k}+1)$$

下面给出完全类似二元情况的计算 $c(z_1,\cdots,z_k)$ 的方法。

令
$$A_{n_1}(z_2,\cdots,z_k)=\sum_{n_2=0}^{N_2-1}\cdots\sum_{n_k=0}^{N_k-1}a(n_1,\cdots,n_k)Z_{2^2}^{n_2}\cdots Z_{k^k}^{n_k}$$

$$B_{n_1}(z_2,\cdots,z_k)=\sum_{n_2=0}^{N_2-1}\cdots\sum_{n_k=0}^{N_k-1}b(n_1,\cdots,n_k)z_{2^2}^{n_2}\cdots z_{k^k}^{n_k}$$

$$C_{n_1}(z_1,\cdots,z_k)=\sum_{n_2=0}^{N_2-1}\cdots\sum_{n_k=0}^{N_k-1}c(n_1,\cdots,n_k)z_{2^2}^{n_2}\cdots z_{k^k}^{n_k}$$

式中，
$$n_1=0,1,\cdots,N_1-1.$$

则
$$A(z_1,\cdots,z_k)B(z_1,\cdots,z_k)\bmod(z_1^N+1)$$
$$\equiv\sum_{P=0}^{N_1-2}\Big[\sum_{n=0}^{N_1-1}(A_{P-n}(z_2,\cdots,z_k)-A_{P+N_1-n}(z_2,\cdots,z_k))$$
$$\cdot B_n(z_2,\cdots,Z_k)\Big]\cdot Z_1^P+\sum_{n=0}^{N_1-1}A_{N_1-1-n}(z_2,\cdots,z_k)B_n(z_2,\cdots,z_k)z_1^{N_1-1}$$

式中，$0\leqslant P-n\leqslant N_1-1$，$0\leqslant P+N_1-n\leqslant N_1-1$.

所以有

$C_{N_1-1}(z_2,\cdots,Z_k)$
$$\equiv\sum_{n=0}^{N_1-1}A_{N_1-1-n}(z_2,\cdots,z_k)\cdot B_n(z_2,\cdots,z_k)\bmod(z_2^{N_2}+1),\cdots,(z_k^{N_k}+1)$$

$C_P(z_2,\cdots,z_k)$
$$\equiv\sum_{n=0}^{N_1-1}[A_{P-n}(z_2,\cdots,z_k)-A_{P+N_1-n}(z_2,\cdots,z_k)]\times B_n(z_2,\cdots,z_k)\bmod(z_2^{N_2}+1),\cdots,(z_k^{N_k}+1)$$

其中，$0\leqslant p\leqslant N_1-2$，$0\leqslant p-n\leqslant N_1-1$，$0\leqslant p+N_1-n\leqslant N_1-1$.

这可转化成以下循环卷积的计算。

$$\overline{C}_P(z_2,\cdots,z_k)\equiv\sum_{n=0}^{2N_1-1}A'_{P-n}(z_2,\cdots,z_k)B'_n(z_2,\cdots,z_k)\bmod(z_2^{N_2}+1),\cdots,(z_k^{N_k}+1)$$
$$(0\leqslant p\leqslant 2N_1-1)$$

式中，
$$A'_n(z_2,\cdots,z_k)=\begin{cases}A_n(z_2,\cdots,z_k) & n=0,\cdots,N-1\\ 0 & n=N_1\cdots,2N_1-1\end{cases}$$
$$B'_n(z_2,\cdots,z_k)=\begin{cases}B_n(z_2,\cdots,z_k) & n=0,1,\cdots,N_1-1\\ 0 & n=N_1,\cdots,2N_1-1\end{cases}$$

而此循环卷积在 $N_i=2^{t_i}$ 时可用 FPT 计算($t_1\leqslant t_2\leqslant\cdots\leqslant t_k$).

3 二维循环卷积多项式乘积算法

3.1 算法简述

设$\{a(n_1,n_2)\},\{b(n_1,n_2)\}$为二维实序列，$n_i=0,1,\cdots,N_i-1$，$N_i=2^{t_i}$，$i=1,2$，$t_2\geqslant t_1$，

其循环卷积是

$$C(l_1,l_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}a(n_1,n_2)b(<l_1-n_1>_{N_1},<l_2-n_2>_{N_2})$$
$$(l_1=0,1,\cdots,N_1-1;l_2=0,1,\cdots,N_2-1)$$

定义
$$A(z_1,z_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}a(n_1,n_2)z_1^{n_1}\cdot z_2^{n_2}$$
$$B(z_1,z_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}b(n_1,n_2)z_1^{n_1}\cdot z_2^{n_2}$$
$$C(z_1,z_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}c(n_1,n_2)z_1^{n_1}\cdot z_2^{n_2}$$

则 $c(z_1,z_2)\equiv A(z_1,z_2)B(z_1,z_2)\bmod(z_1^{N_1}-1),(z_2^{N_2}-1)$

令 $z_1^{N_1}-1=\prod_{i=1}^{t_1+1}P_i(z_1)=(z_1^{2^{t_1-1}}+1)\cdots(z_1^2+1)(z_1+1)(z_1-1)$

$$z_2^{N_2}-1=\prod_{i=1}^{t_2+1}Q_i(z_2)=(z_2^{2^{t_2-1}}+1)\cdots(z_2^2+1)(z_2+1)(z_2-1)$$

$$C_{ij}(z_1,z_2)=C(z_1,z_2)\bmod P_i(z_1),Q_j(z_2)$$
$$(i=1,2,\cdots,t_1+1;j=1,2,\cdots,t_2+1)$$

先固定 i，由孙子定理

$$C(z_1,z_2)\equiv\sum_{j=1}^{t_2}C_{ij}(z_1,z_2)\left(-\frac{1}{2^j}\right)\frac{z_2^{N_2}-1}{z_2^{N_2/2^j}+1}+\frac{1}{2^{t_2}}C_{i,t_2+1}(z_1,z_2)\frac{z_2^{N_2}-1}{z_2-1}\bmod(z_2^{N_2}-1),P_i(z_1)$$
$$\triangleq H_i(z_1,z_2)\bmod(z_2^{N_2}-1),P_i(z_1)$$

再由孙子定理得

$$C(z_1,z_2)\equiv\sum_{i=1}^{t_1}H_i(z_1,z_2)\left(-\frac{1}{2^i}\right)\frac{z_1^{N_1}-1}{z_1^{N_1/2^i}+1}+\frac{1}{2^{t_1}}\cdot\frac{z^{N_1}-1}{z_1-1}\cdot H_{t_1+1}(z_1,z_2)\bmod(z_1^N-1),(z_2^{N_2}-1)$$

可见只要计算出 $C_{ij}(z_1,z_2)$即可。

令 $A_{ij}(z_1,z_2)\equiv A(z_1,z_2)\bmod P_i(z_1),Q_j(z_2)\quad B_{ij}(z_1,z_2)\equiv B(z_1,z_2)\bmod P_i(z_1),Q_j(z_2)$

则 $C_{ij}(z_1,z_2)\equiv A_{ij}(z_1,z_2)B_{ij}(z_1,z_2)\bmod P_i(z_1),Q_j(z_2)$
$$(1\leqslant i\leqslant t_1+1,1\leqslant j\leqslant t_2+1)$$

显然上式可由二元多项式乘积算法计算。

3.2 计算量估计

设 M_{ij}，A_{ij}分别是计算 $C_{ij}(z_1,z_2)$所需的乘法量和加法量，则

$$M=\sum_{i=1}^{t_1-1}\sum_{j=1}^{t_2-1}2^{i+1}M_u(2^j)+\sum_{i=2}^{t_1-1}2^{i+1}M_u(2^i)+2\sum_{j=2}^{t_2-1}M_u(2^j)-2\sum_{i=2}^{t_1-1}M_u(2^i)+10$$

$$A=\sum_{i=1}^{t_1-1}\sum_{j=1}^{t_2-1}A_{ij}+2\sum_{i=1}^{t_1-1}A_d(2^j)+2\sum_{i=1}^{t_2-1}A_d(2^i)+8N_1N_2-4(N_1+N_2)$$

式中，$A_d(N)$和$M_u(N)$如前面所说。

下面表格数据表明这种算法与FPT法相比乘法量显著减少。

表 1

维数	FPT算法实乘数	多项式乘积算法实乘数
$2^4\times2^4$	3168	1390
$2^5\times2^5$	15552	7090
$2^6\times2^6$	74112	42718
$2^7\times2^7$	344712	296618
$2^8\times2^8$	1574400	918630
$2^9\times2^9$	7080960	3910594
$2^{10}\times2^{10}$	31463424	21886006
$2^5\times2^{10}$	793728	586948
$2^8\times2^4$	320640	212818

由实例看出，多元多项式乘积算法在计算多维卷积时很有效，这为计算多维卷积和多维DFT提供了新的有力手段，但若想找到更一般的算法，还有大量的工作要做。

参 考 文 献

[1] H. J. 努斯鲍默著；胡光锐译. 快速富里叶变换和卷积算法. 上海科技文献出版社，1984

[2] 蒋增荣，曾永泓著. 多项式变换及其应用. 国防科技大学出版社，1989

[3] 蒋增荣. 数论变换. 上海科技出版社，1980

[4] Nuss baumer H J. Digital Filtering Using Polynomial Transform Electron. Lett, 1977, 13: 386－387

The FPT Algorithm of Multi-variable Polynomial Multiplication

Tian Zerong

(Department of Computer Science)

Jiang Zengrong

(Department of Applied Mathmatics and System Engineering)

Abstract

In this paper, we discussed the FPT algorithm of multi－variable polynomial multiplication in detail. Firstly, we give the case of two－variable. Secondary, we generalize it to multi－variable polynomial. At last, we apply it to the computation of the two dimensional cyclic convolution. As a result, this kind of algorithm is efficient in computation of M－D DFTs and M－D convolutions.

Key words convolution, polynomial transform, multi－varialbe polyomial