国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第12卷第3期 1990年9月 Vol.12 No.3

某些环中卷积的快速计算

曾 泳 泓
（系统工程与应用数学系）

摘 要 本文研究了环中卷积的快速计算问题，讨论了计算域中卷积通常使用的Winograd短卷积算法、快速富里叶变换算法以及多项式变换算法对一般环中卷积计算的可适用性。特别地，对应用广泛的矩阵多项式乘积、矩阵卷积及多项式卷积计算提出了比直接计算快得多的算法。

关键词 近世代数，环，卷积，快速算法
分类号 O153.3

在信号处理及一些其它学科中环中数字卷积的计算有着广泛的应用，例如：整数卷积可以转化成整数剩余环中卷积的计算，任何二维数字卷积都可以转化为多项式剩余环中的一维卷积，矩阵多项式相乘可认为是矩阵环中卷积的计算等等。这些数字卷积的计算在实际中有十分重要的意义，应用很广泛，所以我们有必要研究环中卷积的计算问题。

域上卷积和多项式乘积的快速计算及计算复杂性问题的研究已经取得了很多成果，众所周知的Winograd短卷积算法是其中有名的算法，这个算法指出了在无限域上，用$2n-1$个一般乘法可以计算出两个$n-1$次多项式相乘。在有限域及环上，算法的条件未必能满足，因此，这个算法是否能用于一般环上多项式相乘的计算便为一个值得研究的问题。计算卷积的另一个有名的算法便是利用快速富里叶变换去计算，但这个算法显然有个前提，即必须假定有离散富里叶变换存在，这个假定在复数域上总可以得到满足，但在一般环甚至域内，这种变换不一定存在。因此，我们必须研究在一般环内如何使用类似的算法，利用数论变换计算整数卷积便是一个成功的例子，但还有许多其它情形需要研究。此外，从计算复杂性角度考虑，域中卷积计算的复杂度下界和环中卷积计算的复杂度下界是否是一样的，退一步说，对环中卷积计算即使我们不能使用类似于通常的Winograd短卷积算法或FFT算法，是否存在另外的途径得到有相同或更低复杂度的算法。我们感兴趣的正是诸如此类的一些问题。这些问题不仅在理论上是重要的，也有广阔的应用前景。

1987 8月12日收稿

1 卷积计算的Winograd算法

下设R为一个有单位元的可换环，$P(x)$和$Q(x)$为环R上的二个多项式．我们考虑$P(x)$和$Q(x)$相乘的问题。

$$M(x)=P(x)Q(x) \tag{1}$$

并设$P(x)$和$Q(x)$的次数不超过$n-1$．若R是有理数域，(1)式可通过Winograd短多项式积算法实现，所用乘法数最少。若R为一般环，Winograd算法的条件通常不能满足。下面我们将这个算法推广到用于计算具有某些性质的环及代数中的多项式乘积与卷积。

若R上存在$2n-1$个元$x_0,x_1,\cdots,x_{2n-2}$，满足$x_i-x_j(i\neq j)$在R中可逆，令$T_k(x)=\prod\limits_{j\neq k}\dfrac{x-x_j}{x_k-x_j}$，$k=0,1,\cdots,2n-2$，则(1)式可以如下计算：

(1) 计算$M(x_k)=P(x_k)Q(x_k)$，$k=0,1,\cdots,2n-2$；

(2) $M(x)=\sum\limits_{k=0}^{2n-2}M(x_k)T_k(x)$。

算法的正确性可仿照域上插值公式的正确性进行证明。

算法的第一步需要$2n-1$个一般乘法，而第二步则不需一般乘法。这里一般乘法是指两个乘法因子都是输入变元的非平凡函数的乘法。所以，在上述条件下，环R上的两个$n-1$次多项式相乘只需$2n-1$个一般乘法。

现在考虑环R中长N的循环卷积计算，环R中循环卷积可化为对模x^N-1的多项式乘积

$$M(x)\equiv P(x)Q(x)\bmod(x^N-1) \tag{2}$$

其中$P(x)$和$Q(x)$的次数不超过$N-1$．

设$x^N-1=\prod\limits_{i=1}^{d}\phi_i(x)$，其中$\phi_1(x),\cdots,\phi_d(x)$两两互素（其定义见[5]），$\phi_i(x)$的次数为$n_i$，且$\phi_i(x)$是不可约的，根据一般环上多项式的孙子定理（见[5]），令

$$M_i(x)\equiv P_i(x)Q_i(x)\bmod\phi_i(x),i=1,2,\cdots,d \tag{3}$$

其中$P_i(x)\equiv P(x)\bmod\phi_i(x)$，$Q_i(x)\equiv Q(x)\bmod\phi_i(x)$，则$M(x)$可从$M_i(x)$构造出来，而且这个构造过程不需一般乘法。(3)式计算时，先计算多项式乘积$P_i(x)Q_i(x)$，这可以利用前面提出的算法，然后把$P_i(x)Q_i(x)$的计算结果对模$\phi_i(x)$求剩余，这里不需要一般乘法。因此(3)式的计算需要的一般乘法数为

$$\sum_{i=1}^{d}(2n_i-1)=2n-d$$

即(3)式计算只需$2n-d$个一般乘法。

上述算法的优越性在于减少了一般乘法，但加法数目可能会增加（同直接算法相比），所以上述算法通常用于计算规模较小的问题，对规模很大的问题，可利用小规模问题嵌套计算，这时一般乘法对算法复杂性的影响远远大于非一般乘法和加法，这时减少一般乘法意义很大。

下面考虑一种很有实际意义的情况。

设F为一个域，K是域F上的一个线性代数（其定义见[6]）。现在考虑K中两个多

项式相乘

$$M(x)=P(x)Q(x) \tag{4}$$

其中$P(x)$和$q(x)$的系数都是k的元素，其次数不超过$n-1$.

设$x_0,x_1,\cdots,x_{2n-2}$为域F中的$2n-1$个不同元素，(4)式可以如下计算：

step1　计算$M_i=P(x_i)Q(x_i)$，$i=0,1,\cdots,2n-2$.

step2　$M(x)=\sum_{i=0}^{2n-2}M_i\prod_{j\neq i}\frac{x-x_i}{x_i-x_j}$.

step1中用了$2n-1$个K中元素之间的乘法，step2中只需用加法和F中数乘K中元素的运算，所以和直接算法相比，该算法把K中元素之间的乘法从n^2减少到$2n-1$，代价是增加了加法和数乘K中元素的运算。当K中元素之间相乘的代价很高时，这个算法是优越的。

特别地，设K是域F上的N阶矩阵代数，这是一个不交换代数，K上的多项式即为矩阵多项式，K上两个多项式相乘就是矩阵多项式相乘。因此，二个$n-1$阶矩阵多项式相乘只需用$2n-1$个矩阵乘法，而直接计算需要n^2个矩阵乘法。在step1和step2中所用的数乘矩阵和矩阵加法的数目至多为$c\cdot n^2$个（c为常数），所以算法的总运算量为$(2n-1)\cdot N^3+c\cdot n^2N^2$，而直接算法的运算量为$n^2N^3+n(n-1)N^2$，$[(2n-1)\cdot N^3+cn^2N^2]/[n^2N^3+n(n-1)N^2]=\left(\frac{2}{n}-\frac{1}{n^2}+\frac{c}{N}\right)\Big/\left[1+\left(1-\frac{1}{n}\right)\frac{1}{N}\right]$，当$N$较大时，上式接近于$2/n$，所以新算法的总运算量只相当于直接算法的$2/n$倍。

2　DFT用于计算卷积

在复数域上，计算卷积的一个最有效的办法是利用FFT计算，前提是必须存在DFT.在一般环上，DFT型变换一般不存在，因此不能利用FFT算法。但是在某些特殊情况下，类似的算法仍可采用。

设F为一个域，K为域F上的线性代数，考虑K上多项式乘积或循环卷积的DFT算法。

考虑K中的一维循环卷积

$$M_k=\sum_{i=0}^{N-1}p_iq_{\langle k-i\rangle_N} \tag{5}$$

直接计算(5)式需要N^2个一般乘法以及$N(N-1)$个一般加法。

引理1　设α为域F中的N阶单位根，令

$$\bar{p}_k=\sum_{i=0}^{N-1}p_i\alpha^{ki} \tag{6}$$

$$\bar{q}_k=\sum_{i=0}^{N-1}q_i\alpha^{ki},\quad k=0,1,\cdots,N-1 \tag{7}$$

则

$$M_i=\frac{1}{N}\sum_{k=0}^{N-1}(\bar{p}_k\cdot\bar{q}_k)\alpha^{-ki},\quad i=0,1,\cdots,N-1 \tag{8}$$

证明

$$\frac{1}{N}\sum_{k=0}^{N-1}(\bar{p}_k\cdot\bar{q}_k)\alpha^{-ki}$$

$$=\frac{1}{N}\sum_{k=0}^{N-1}\sum_{j=0}^{N-1}\sum_{u=0}^{N-1}p_jq_u\alpha^{k(j+u-i)}$$

$$=\frac{1}{N}\sum_{j=0}^{N-1}\sum_{u=0}^{N-1}p_jq_u\sum_{k=0}^{N-1}\alpha^{k(j+u-i)}$$

$$=\sum_{j=0}^{N-1}p_jq_{\langle i-j\rangle_N}=M_i \qquad Q\cdot E\cdot D$$

所以K中长N的循环卷积的计算可如下进行：

step1 计算序列$\{p_i\}$和$\{q_i\}$的DFT(6)和(7)式。

step2 $r_k=\bar{p}_k\cdot\bar{q}_k$, $k=0,1,\cdots,N-1$.

step3 计算序列$\{r_k\}$的逆DFT(8)式。

当N为高度复合数时，(6)、(7)和(8)式可采用FFT型快速算法，所需运算量为$O(N\log N)$个数和K中元素相乘（数乘向量），以及$O(N\log N)$个K中元素之加法。所以计算(5)式所需的总运算量为：N个K中元素的乘法，$O(N\log N)$个数乘向量以及$O(N\log N)$个K中元素的加法。这比直接计算所需的运算量少得多。

特例1 若K为F上的n阶矩阵代数，则长N的矩阵序列循环卷积的计算只需用N个矩阵乘法，$O(N\log N)$个矩阵加法和$O(N\log N)$个数乘矩阵，总运算量为

$$n^3N+O(n^2N\log N)$$

这比直接计算的$O(n^3N^2)$运算量少得多。

特例2 计算域F上的多项式循环卷积

$$H_k(x)=\sum_{i=0}^{N-1}P_i(x)Q_{\langle k-i\rangle_N}(x)$$

其中$P_i(x)$和$Q_i(x)$的次数均不超过$n-1$.

按上述方法计算，只需N个多项式乘法，$O(N\log N)$个多项式加法以及$O(N\log N)$个数乘多项式，故总运算量为$O(Nn\log n)+O(nN\log N)$。而直接计算需要$O(N^2n\log n)$的运算量（设两个多项式相乘用FFT算法，运算量为$O(n\log n)$）。两者之比为$O(\log n+\log N)/O(N\log n)$，新算法的运算量少得多。

3 一般环中二维卷积的计算

首先设R为可换环，考虑R中二维循环卷积

$$c_{u,v}=\sum_{m=0}^{N-1}\sum_{n=0}^{M-1}a_{m,n}b_{\langle u-m\rangle,\langle v-n\rangle} \tag{9}$$

式中 $u=0,1,\cdots,M-1$, $v=0,1,\cdots,N-1$

在复数域或有理数域上，这种循环卷积可利用FFT或Winograd算法来作，然而在一般环R中，正如前面已经指出过的，通常FFT算法和Winograd算法的前提条件得不到满足，因而一般无法采用，我们期望用[5]中提出的可换环上多项式变换来计算。

令 $C_v(z)=\sum_{u=0}^{M-1}c_{u,v}z^u$,

类似地定义$A_v(z)$和$B_v(z)$，$v=0,1,\cdots,N-1$．于是

$$C_v(z)\equiv\sum_{n=0}^{N-1}A_n(z)B_{\langle v-n\rangle_N}(z)\qquad \bmod(z^M-1)\tag{10}$$

为简单起见，我们设$M=p^{t+r}$, $N=p^t$，t为正整数，$r\geqslant 0$, p为素数。

$$z^M-1=(z^{p^{t+r-1}}-1)P(z)$$

其中
$$P(z)=(z^{p^{t+r}}-1)/(z^{p^{t+r-1}}-1)$$

令
$$C_{v,1}(z)\equiv\sum_{n=0}^{N-1}A_{n,1}(z)B_{v-n,1}(z)\qquad \bmod P(z)\tag{11}$$

$$C_{v,2}(z)\equiv\sum_{n=0}^{N-1}A_{n,2}(z)B_{v-n,2}(z)\qquad \bmod(z^{p^{t+r-1}}-1)\tag{12}$$

其中
$$A_{n,1}(z)\equiv A_n(z)\qquad \bmod P(z)$$
$$A_{n,2}(z)\equiv A_n(z)\qquad \bmod(z^{p^{t+r-1}}-1)$$

类似地有$B_{n,1}(z)$和$B_{n,2}(z)$。求得$C_{v,1}(z)$和$C_{v,2}(z)$之后，根据孙子定理[5]可构造出$C_v(z)$，构造过程无需一般乘法。

若P在R中可逆，则$(P(z),z^{p^r},N)$构成多项式变换[5]，所以(11)式的计算可采用多项式变换去做，做法同域上的情形一样，需做三个多项式变换以及N个模$p(z)$内的多项式乘积。由于多项式变换不需要乘法，因此(11)式的计算所需的乘法次数（多项式相乘直接计算）为：

$$p^t[p^{t+r-1}(p-1)]^2$$

需(12)式仍为一个$N\times(M/p)$的循环卷积，其计算可重复上述做法。所以，若设$p^t\times p^l$卷积所需的乘法数为$M(t,l)$，则$p^t\times p^{t+r}$卷积所需的乘法次数为

$$M(t+r,t)=M(t,t+r)=p^t[p^{t+r-1}(p-1)]^2+M(t,t+r-1)\tag{13}$$

当$r=0$时，

$$M(t)=M(t,t)=p^t[p^{t-1}(p-1)]^2+M(t,t-1)$$

由(13)式，$M(t,t-1)=M(t-1,t)=p^{t-1}[p^{t-1}(p-1)]^2+M(t-1)$

故$M(t)=p^t[p^{t-1}(p-1)]^2+p^{t-1}[p^{t-1}(p-1)]^2+M(t-1)$

$$M(1)=p^t\ \text{（直接计算）}$$

所以

$$M(t)=p^t+\sum_{i=2}^{t}p^{3i-3}(p-1)^2(p+1)$$

$$=(p^{3t}-1)\frac{(p-1)^2(p+1)}{p^3-1}+p^t-(p-1)^2(p+1)$$

当$r\geqslant 1$时，

$$M(t,t+r)=\sum_{i=0}^{r-1}p^i[p^{t+i}(p-1)]^2+M(t)$$

$$=p^{3t}(p-1)^2\sum_{i=0}^{r-1}p^{2i}+M(t)$$

$$=p^{3t}\frac{p^{2r}-1}{p^2-1}(p-1)^2+M(t)$$

$$=p^{3t}(p^{2r}-1)\frac{(p-1)^2}{p^2-1}+\frac{p^{3t}-1}{p^3-1}(p-1)^2(p+1)+p^t-(p-1)^2(p+1)$$

而直接计算一个$p^t\times p^{t+r}$的卷积所需的运算量为$\bar{M}(t,t+r)=p^{2t}p^{2t+2r}=p^{4t+2r}$个乘法，所以

$$M(t,t+r)/\bar{M}(t,t+r)\doteq 1/p^t$$

即乘法运算量仅仅为直接算法的$1/p^t$倍。

值得注意的是，上述算法中用到的条件只有一个：p在R中可逆，这比要求有DFT存在要弱得多。

需要指出的是，[5]中只建立了一般可换环上的多项式变换理论，没有提及不可换环的情形，一般地说，[5]中的结果对不可换环未必成立，然而在某些情况下，即使对不可换环，多项式变换算法仍可使用。

设F为一个域，K为域F上的线性代数(可以不交换)，下面我们试图利用域F上定义的多项式变换来计算K中二维卷积(9)式。

引理2 若$(M(z),F(z),N)$构成域F上的多项式变换，$\{A_i(z)\}$，$\{B_i(z)\}$为K上的多项式序列$(i=0,1,\cdots,N-1)$，则循环卷积

$$C_k(z)\equiv\sum_{i=0}^{N-1}A_i(z)B_{\langle k-i\rangle_N}(z)\qquad \mathrm{mod}M(z)\tag{14}$$

可以如下计算：

令
$$\bar{A}_k(z)\equiv\sum_{i=0}^{N-1}A_i(z)F^{ik}(z)\qquad \mathrm{mod}\,M(z)$$

$$\bar{B}_k(z)\equiv\sum_{i=0}^{N-1}B_i(z)F^{ik}(z)\qquad \mathrm{mod}\,M(z)$$

则有
$$C_k(z)\equiv\frac{1}{N}\sum_{i=0}^{N-1}(\bar{A}_k(z)\cdot\bar{B}_k(z))F^{-ik}(z)\qquad \mathrm{mod}M(z)$$

可仿照域上的情形证明此引理。

现在考虑K中二维循环卷积(9)式，这时序列$a_{i,j}$和$b_{i,j}$均为K中的序列。若$C_v(z)$，$A_n(z)$，$B_n(z)$按前面的定义，则(9)式等价于

$$C_v(z)\equiv\sum_{n=0}^{N-1}A_n(z)B_{\langle v-n\rangle N}(z)\qquad \mathrm{mod}(z^M-1)\tag{15}$$

设$N=p^t$，$M=p^{t+r}$，t为正整数，$r\geqslant 0$，并令

$$P(z)=(z^{p^{t+r}}-1)/(z^{p^{t+r-1}}-1)$$

$$C_{v,1}(z)\equiv\sum_{n=0}^{N-1}A_{n,1}(z)B_{v-n,1}(z)\qquad \mathrm{mod}P(z)\tag{16}$$

$$C_{v,2}(z)\equiv\sum_{n=0}^{N-1}A_{n,2}(z)B_{v-n,2}(z)\qquad \mathrm{mod}(z^{p^{t+r-1}}-1)\tag{17}$$

设p在F中可逆，由于$(P(z),z^{p^r},N)$构成域F上的多项式变换，所以(16)式可用引理2的办法进行计算，只需3个多项式变换，不需要乘法；(17)式等价于一个$N\times(M/p)$的二维循环卷积，可同样计算。当求得$C_{v,1}(z)$和$C_{v,2}(z)$之后，根据孙子定理可构造出$C_v(z)$，这个过程不需乘法。因此，(15)式的计算和(10)式的计算过程完全一样，所以计算(15)式所需的乘法数（K中元素相乘数）只为直接算法的$1/p^t$倍。

特别地，当K为域F上的n维矩阵代数时，$p^t \times p^{t+r}$二维矩阵卷积的计算只需用

$$M(t,t+r)=p^{3t}(p^{2r}-1)\frac{(p-1)^2}{p^2-1}+\frac{p^{3t}-1}{p^3-1}(p+1)(p-1)^2+p^t-(p-1)^2(p+1)$$

个矩阵乘法，而直接计算需要p^{4t+2r}个矩阵乘法，新算法的乘法数为直接算法的$1/p^t$倍。若对多项式变换采用类似于FFT的技巧，则新算法的加法数同直接算法大致相同。

参 考 文 献

[1] Winograd S. Some Bilinear Forms Whose Multiplicative Complexity Depends on the Field of Constants. Math. Syst. Th. 10, 1977: 169~180

[2] Agarwal R C, Cooley J W. New Alogorthms for Digital Convolution. Intern. Conf. on ASSP Proc. P360

[3] Nussbaumer H J. Fast Fourier Transform and Convolution Algorithms. Springer-Verlag Berlin Heidelberg, New York, 1981

[4] 曾济宇. 有限域上的多项式变换. 国防科技大学学报, 1986, (2)

[5] 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

[6] Keneth Hoffman and Ray kunze. Linear Algebra. Second Edition. Prentice-Hall Inc., England Cliffs, New Jersey. P117

Fast Computation for Convolutions in Some Rings

Zeng Yonghong

(Department of Applied Mathematics and System Engineering)

Abstract

This paper discusses the problem of fast computation of convolutions in rings. The adaptability of the well-known Winograd algorithm, FPT algorithm and FFT algorithm which are proposed for computing convolutions in complex fields is studied in the case of rings. Especially, fast algorithms for computing multiplication of matrix polynomial, matrix convolution and polynomial convolution are proposed. They are of wide practical usage.

Key words: modern algebra, ring, convolution, fast algorithm