技术交流

车贝雪夫多项式约简算法的研究

朱应珍　　　　黄永玉

（福建省气象台，350001）　　（福建省建阳市气象局，354200）

提　　要

解释了车贝雪夫多项式化整计算通用算法中关于化整后的车贝雪夫多项式约简处理的技术原理，给出了一种可快速实现多项式约简处理的实用求算方法。

关键词： 约简　最大公因数　带余除法　递归算法

引　言

文献[1]提出了一种车贝雪夫多项式化整计算的通用算法，为多项式的实际应用带来了一定的方便。特别是在中尺度灾害性天气预报分析中采用嵌套场展开[2]时文献[1]的算法尤为见长。

通用算法须分两步实现：

第一步，按文献[1]所导出的递推算式

$\Psi_{0,x} = 1$

$\Psi_{1,x} = 2x - n - 1$

$\cdots\quad\cdots\quad\cdots$

$\Psi_{k+1,x} = [(2k+1) \times \Psi_{1,x} \times \Psi_{k,x} - (n^2 - k^2) \times \Psi_{k-1,x}]/(k+1)^2$

来求算出整数化的多项式值，这里，

n 为格点数，$x = 1,\cdots,n$

k 为阶数，$k = 0,1,\cdots,n-1$。

第二步，对整数化后的多项式进行约简处理，使经约简后的多项式具有最大公因数为 1 的性质，即

$(\Psi'_{k,1},\Psi'_{k,2},\cdots,\Psi'_{k,n}) = 1$

这里

$\Psi'_{k,x} = \Psi_{k,x}/d_{k,n}, x = 1,2,\cdots,n$；

$d_{k,n}$ 为第 k 阶多项式的最大公因数，即

$(\Psi_{k,1},\Psi_{k,2},\cdots,\Psi_{k,n}) = d_{k,n}$

$k = 0,1,\cdots,n-1$。

关于 $d_{k,n}$ 的求取，文献[1]所依据的数学原理是这样的一道定理[3]：

设 $a_1,a_2,\cdots,a_n$ 为 $n(>2)$ 个不同时为零的正整数，

$(a_1,a_2) = d_2,\ (d_2,d_3) = d_3,\cdots,$

$(d_{n-1},a_n) = d_n$，

则

$(a_1,a_2,\cdots,a_n) = d_n$

可见，多项式约简处理的基础是求两个整数 a 和 b 的最大公因数 (a,b)。

如何求取 (a,b)？文献[1]没有具体的说明，因而，引用者一般易于选择基于整数的“标准分解式”[3]来做为解决问题的出路：

设 a、b 均为正整数，则其相应的“标准分解式”可表达为

$a = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$，

$a_i \geqslant 0 (i = 1,2,\cdots,k)$；

$b = p_1^{\beta_1} p_2^{\beta_2} \cdots p_k^{\beta_k}, \beta_i \geqslant 0 (i = 1,2,\cdots,k)$；

a 和 b 的最大公因数可按下式求取：

$(a,b) = p_1^{\gamma_1} p_2^{\gamma_2} \cdots p_k^{\gamma_k}$，

这里　$\gamma_i = \min(\alpha_i,\beta_i)$；

$i = 1,2,\cdots,k$；

$p_i < p_j (i < j)$ 均为素数。

这种求法在理论上是正确的，但在计算实践上却不总是可行的，因为当 a（或 b）很大时，求其“标准分解式”的计算量会很大，甚至是不可实现的，如 $a = 78497$，在 586 微机上要花 34 秒钟才能完成分解任务。而实际问题是，当 n 较大时

$$\max_{\substack{0 \leqslant k \leqslant n-1 \\ 1 \leqslant x \leqslant n}} |\Psi_{k,x}|$$

往往可以达到很大的值。所以，基于“标准分解式”的多项式约简方案是不实用的。这表

明，要引用文献[1]的通用算法，必须另找适用的约简求算方案。笔者基于引用的需要，对此问题做了一些实用性的探讨，建立了一种实用性较好的约简求算方案（算法流程），可做为文献[1]的通用算法的一个有益补充，为文献[1]的引用者提供引用上的方便。

1 引入两个定量

1.1 定理1

设a和b是两个整数，$b>0$，则存在唯一整数q和r，使得关系式

$$a=bq+r,\ 0\leqslant r<b, \tag{1}$$

成立。

证 先证明q和r的存在。

作整数序列

$$\cdots,-3b,-2b,-b,0,b,2b,3b,\cdots,$$

∵ a为整数，

∴ a必在序列中的某两项之间，故必有一整数q，使得

$$bq\leqslant a<b(q+1) \tag{2}$$

成立。

由(2)得

$$0\leqslant a-bq<b \tag{3}$$

令$r=a-bq$，即有

$$a=bq+r,\ 0\leqslant r<b$$

∴ 式(1)成立，即q和r是存在的。

再证明q和r是唯一的。

设另有整数q'和r'也使式(1)成立，即有

$$bq+r=bq'+r' \tag{4}$$

由此可得

$$b(q-q')=r'-r$$

$$b|q-q'|=|r'-r| \tag{5}$$

∵ r,r'均为小于b的正数，

$$\therefore |r-r'|<b \tag{6}$$

因此，若$q\neq q'$，则应有

$$b|q-q'|\geqslant b \tag{7}$$

而与式(6)相矛盾。

∴ $q=q',r=r'$，即q和r是唯一的，证毕。

1.2 定理2

设a,b,c是任意三个不全为零的正整数，$a=bq+c$，q为整数，则有

$$(a,b)=(b,c)$$

证 显然有

$(a,b)|a$，

$(a,b)|b$，

∵ $c=a-bq$，q为整数，

∴ $(a,b)|c$

∵ b和c的最大公因数为(b,c)，

∴

$$(a,b)\leqslant(b,c) \tag{8}$$

同理可证

$(b,c)|c$，

$(b,c)|b$，

$(b,c)|a$.

$$\because\ (b,c)\leqslant(a,b) \tag{9}$$

∴ $(a,b)=(b,c)$ 证毕。

这里的算符$x|y$，表示整数y被整数x整除。

2 多项式约简的求算方法

据定理1可知，对于任给的正整数a和b，总可进行式(10)所示的递次带余除法运算：

$$\begin{aligned}&a=bq_1+r_1,\ 0<r_1<b;\\&b=r_1q_2+r_2,\ 0<r_2<r_1;\\&r_1=r_2q_3+r_3,\ 0<r_3<r_2;\\&\quad\cdots\cdots,\\&r_{n-2}=r_{n-1}q_n+r_n,\ 0<r_n<r_{n-1};\\&r_{n-1}=r_nq_{n+1}+r_{n+1},\ r_{n+1}=0\end{aligned} \tag{10}$$

这里$r_1,r_2,\cdots,r_{n+1}$为余数。

因为$b>r_1>r_2\cdots$，所以，经过有限次带余除法运算之后，总可得到余数为零的结果，设此时为$r_{n+1}=0$。

由定理2可知，在式(10)的带余除法运算过程中，必存在着如下连环关系：

$$\begin{aligned}&(a,b)=(b,r_1),\\&(b,r_1)=(r_1,r_2),\\&(r_1,r_2)=(r_2,r_3),\\&\cdots\ \cdots\ \cdots\\&(r_{n-1},r_n)=(r_n,r_{n+1})=r_n\end{aligned}$$

由此可得

$$(a,b) = r_n \tag{11}$$

式(11) 表明，欲求(a,b)，可通过进行如式(10) 所示的递次带余除法运算来实现。

自然，欲求$(\Psi_{k,1},\Psi_{k,2},\cdots,\Psi_{k,n})$只须令

$$a_i = |\Psi_{k,i}|,\ i = 1,2,\cdots,n$$

再按式(10)，依序求出

$$(a_1,a_2) = d_2,$$
$$(d_2,a_3) = d_3,$$
$$\cdots\quad\cdots\quad\cdots$$
$$(d_{n-1},a_n) = d_n。$$

令

$$d_{k,n} = d_n,$$
$$\Psi'_{k,x} = \Psi_{k,x}/d_{k,n}, x = 1,2,\cdots,n;$$

即可达到

$$(\Psi'_{k,1},\Psi'_{k,2},\cdots,\Psi'_{k,n}) = 1$$

的目的。

这就是说，多项式的约简算法，其实就是依基于定理 1 和定理 2 的式(10)，进行递次的带余除法运算。

3　带余除法的优化

对式(1)中的余数 r 的分析可以看出：

① 若$b > r > b/2$，则式(1) 可以改写为：

$$\begin{aligned} a &= bq + r \\ &= b(q+1) + (r-b) \\ &= bt + s \end{aligned}$$

此时，$t = q + 1, |s| = |r - b| < b/2$；

② 若 $0 \leqslant r < b/2$，则式(1) 可视为：

$$\begin{aligned} a &= bq + r \\ &= bt + s \end{aligned}$$

此时，$t = q, s = r < b/2$；

③ 若 $r = b/2$(b 为偶数)，则式(1) 可改写为两种形式：

A　$$\begin{aligned} a &= bq + r \\ &= bt + s \end{aligned}$$

此时，$t = q, s = r = b/2$；

B　$$\begin{aligned} a &= b(q+1) + (r-b) \\ &= bt + s \end{aligned}$$

此时，$t = q + 1, s = r - b = -b/2$。

由上分析可知，若不限定式(1) 中的余数 r 为正整数，则对于给定的正整数 a 和 b，总存在一对整数 t 和 s，使得

$$a = bt + s,\quad |s| \leqslant b/2 \tag{12}$$

成立。这里 $|s|$ 称为 a 被 b 除所得的“绝对最小剩余”。

这表明，亦可按式(12) 的要求，进行如式(10) 的带余除法运算。设最后一个不为零的余数 S_n，则根据定理 2 可知，此时必有

$$(a,b) = |S_n| \tag{13}$$

成立。

可见，求(a,b)的运算，也可基于式(12)，做如式(10) 的带余除法运算来实现。注意到，基于式(1) 的带余除法运算，只能保证每次的余数小于 b，而基于式(12) 的带余除法运算，则能保证余数的绝对值小于等于 $b/2$。这表明，基于式(12) 的带余除法运算使余数为零所需的计算步数必不多于基于式(1) 的带余除法运算使余数为零所需的计算步数。换言之，基于式(12) 来求(a,b)的速度要比基于式(1) 的快，计算实践可很好地证实这一推断。

例：$a = 576, b = 316$，求(a,b)。

解 1：$576 = 316 \times 1 + 260$，
$316 = 260 \times 1 + 56$，
$260 = 56 \times 4 + 36$，
$56 = 36 \times 1 + 20$，
$36 = 20 \times 1 + 16$，
$20 = 16 \times 1 + 4$，
$16 = 4 \times 4 + 0$
$\therefore (a,b) = 4$。

解 2：$576 = 316 \times 2 - 56$
$316 = 56 \times 6 - 20$
$56 = 20 \times 3 - 4$
$20 = 4 \times 5 + 0$
$\therefore (a,b) = 4$。

可见，基于式(12)的解 2 要比基于式(1)的解 1 少 3 步的运算。“速度”几乎快了一倍。这表明，取基于式(12)的多项式约简处理速度将可明显快于取基于式(1)的多项式约简处理速度。所以，在实用中，应取基于式(12)的带余除法运算为多项式约简处理的求算方法。

4 多项式约简计算流程设计

如图1所示，其中

$\langle y\rangle x$ 表示求 y/x 的余数；

$\max(x,y)$ 表示取 x,y 中之大者；

$\min(x,y)$ 表示取 x,y 中之小者；原始

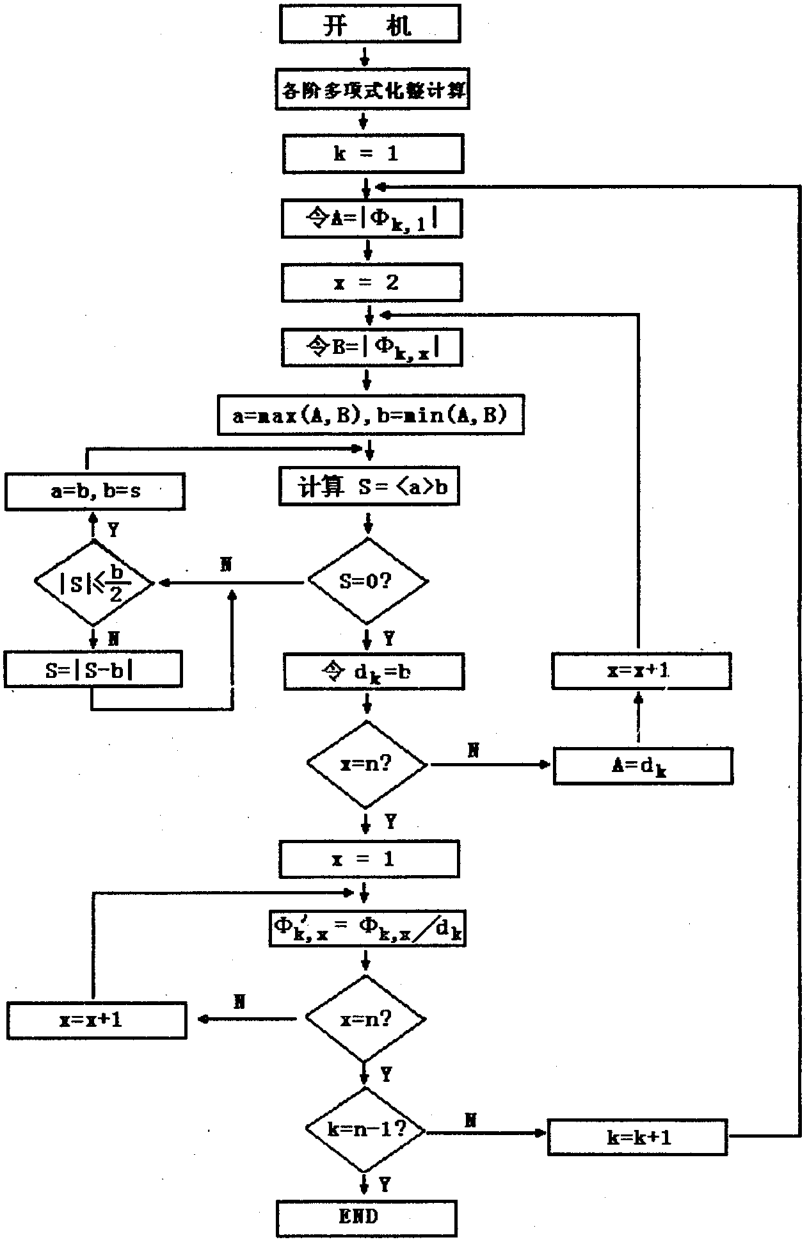

图1 多项式 $\Psi_{k,x}$ 约简处理流程框图

5 计算实例及比较

这里将基于式(12)的带余除法运算的多项式约简方案称为“递归算法”，而把基于“标准分解式”的多项式约简处理方案称为“原始

算法”。在同等的 n 值下，两种算法的速度比较列于表 1。

从表 1 可见，“递归算法”的速度比之“原始算法”快得多，后者基本上是不可用的，而前者则是完全可以满足实时应用的需求。“递归算法”的完整计算实例见表 2。

表 1 递归算法与原始算法的比较

N	10	11	12	13	14	15	16	17	18	19	20	21	22	23
递归算法 t/s	1	1	1	1	1	1	1	1	1	1	1	1	1	1
原始算法 t/s	273	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

注：程序以 c 语言编写，在 IBM-PC586 上运行，t 为完成满阶计算的时间，不包括显示和打印输出的耗时，当 $t \geqslant 300$ 秒而未能完成满阶计算时，视 t 为∞。

表 2 “递归算法”计算实例（$n = 21$）

$x=$	1	2	3	4	5	6	7	8	9	10	11
$k=1$	−10	−9	−8	−7	−6	−5	−4	−3	−2	−1	0
2	190	133	82	37	−2	−35	−62	−83	−98	−107	−110
3	−285	−114	12	98	149	170	166	142	103	54	0
4	969	0	−510	−680	−615	−406	−130	150	385	540	594
5	−3876	1938	3468	2618	788	−1063	−2354	−2819	−2444	−1404	0
6	6460	−7106	−6392	−918	3996	6075	5088	2001	−1716	−4628	−5720
7	−3230	5814	2006	−2754	−4266	−2565	543	3087	3822	2548	0
8	3230	−8398	1394	6246	3618	−2025	−5421	−4557	−588	3626	5390
9	−1292	4522	−3128	−3298	932	3479	2264	−931	−3136	−2646	0
10	1292	−5814	7276	1666	−5484	−4021	2194	5439	2744	−2646	−5292
11	−1615	9044	−16762	6052	12421	−2660	−12164	−4508	8428	10584	0
12	4845	−32946	82416	−74494	−28201	66140	35356	−49196	−55076	18816	64680
13	−570	4617	−14766	21361	−7122	−15130	8348	14406	−5096	−15288	0
14	114	−1083	4272	−8573	7702	1318	−7432	458	6968	−312	−6864
15	−19	209	−991	2569	−3671	2038	1678	−2762	−572	2808	0
16	95	−1197	6691	−21461	42027	−46900	15796	29532	−34762	−9126	38610
17	−10	143	−928	3577	−8942	14620	−14144	3604	9724	−11934	0
18	10	−161	1198	−5433	16626	−35700	53448	−51204	18564	26962	−48620
19	−1	18	−152	798	−2907	7752	−15504	23256	−25194	16796	0
20	1	−20	190	−1140	4845	−15504	38760	−77520	125970	−167960	184756

注：$n - x + 1 = 12 \cdots 21$ 的多项式值可由对称关系得到：

当 k 为偶数：$\Psi_{k,n-x+1} = \Psi_{k,x}$，当 k 为奇数：$\Psi_{k,n-x+1} = -\Psi_{k,x}$；

$$x = 1,2,\cdots \begin{cases} n/2 & n \text{ 为偶数} \\ (n+1)/2 & n \text{ 为奇数} \end{cases}$$

参考文献

1 张明席.车贝雪夫多项式化整计算的通用算法.《大气科学》文集，北京：科学出版社，1990：243～251.

2 张明席.车贝雪夫展开系数时间变量的意义及其应用.《大气科学》文集，北京：科学出版社，1990：235～243.

3 柯 召，孙 琦编著.数论讲义，北京：高等教育出版社，1987：6～8.

The Reduction Method of Chebyshev Polynomial

Huang Yongyu

(Jianyang City Meteorological Office, Fujian Province 354200)

Zhu Yingzhen

(Fujian Meteorological Observatory, Fuzhou 350001)

Abstract

The principle of the reduction method of Chebyshev polynomial in the common algorithm of integerizing chebyshev polynomial was interpreted. A calculating method was given to reduct the polynomial quickly.

Key Words: reduction Chebyshev polynomial largest common factor division with remainder recurrence arithmetic