ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

И.Н. Кухтина, Д.П. Шишков

Р-1497

АЛГОРИТМ ВЫЧИСЛЕНИЯ ЭЛЕМЕНТАРНЫХ СИММЕТРИЧЕСКИХ МНОГОЧЛЕНОВ

Дубна 1964

И.Н. Кухтина, Д.П. Шишков

Р-1497

АЛГОРИТМ ВЫЧИСЛЕНИЯ ЭЛЕМЕНТАРНЫХ СИММЕТРИЧЕСКИХ МНОГОЧЛЕНОВ

Дубна 1964

В в е д е н и е

Элементарными симметрическими многочленами [1] n -го порядка от переменных $M_1, M_2, \ldots, M_N$ называются суммы

$$\sum_{i_1 < i_2 < \ldots < i_n}^{N} M_{i_1} M_{i_2} \ldots M_{i_n} \quad , \quad n \leq N . \tag{1}$$

Они находят применение в ряде задач высшей алгебры, например, при вычислении коэффициентов полиномов по заданным корням или при вычислении любых рациональных симметрических функций [1].

Они играют существенную роль и в некоторых задачах квантовой механики и ядерной физики, например, при расчете энергий возбужденных состояний сильно деформированных ядер на основе сверхтекучей модели ядра с проецированными волновыми функциями [2], при вычислении фазовых объемов [3], исследовании $K^+ - p$ – взаимодействия при больших энергиях [4].

Кроме того, в ряде задач, например, в [2], встречаются суммы

$$\sum_{i_1 < i_2 < \ldots < i_n}^{N} M_{i_1} M_{i_2} \ldots M_{i_n} \left(C_{i_1} + C_{i_2} + \ldots + C_{i_n} \right), \quad n \leq N , \tag{2}$$

которые легко выражаются через элементарные симметрические многочлены.

В данной работе рассматривается эффективный метод для счета сумм (1) и (2) на электронных цифровых машинах (ЭЦМ).

ГЛАВА ПЕРВАЯ

§ 1. О вычислении элементарных симметрических многочленов

Суммы (1) образуются следующим образом: берем N элементов $M_1, M_2, \ldots, M_N$, из них образуем всевозможные сочетания n -го порядка $(n \leq N)$, перемножаем элементы каждого такого сочетания и складываем полученные произведения. Ясно, что количество слагаемых в (1) C_N^n.

Например,

$$\begin{aligned}\sum_{i_1 < i_2 < i_3}^{N} M_{i_1} M_{i_2} M_{i_3} &= M_1 M_2 M_3 + M_1 M_2 M_4 + M_1 M_2 M_5 + M_1 M_3 M_4 + \\ &+ M_1 M_3 M_5 + M_1 M_4 M_5 + M_2 M_3 M_4 + M_2 M_4 M_5 + M_3 M_4 M_5 \\ &= M_1 (M_2 (M_3 + M_4 + M_5) + M_3 (M_4 + M_5) + M_4 M_5) +\end{aligned}$$

$$+ M_2 (M_3 (M_4 + M_5) + M_4 M_5)$$
$$+ M_3 (M_4 M_5)$$
$$= \sum_{i_1=1}^{3} M_{i_1} \left(\sum_{i_2=i_1+1}^{4} M_{i_2} \left(\sum_{i_3=i_2+1}^{5} M_{i_3} \right) \right).$$

При помощи полной индукции для любых целых положительных n и N, $n \leq N$ можно доказать, что

$$\sum_{i_1<i_2<\ldots<i_n}^{N} M_{i_1} M_{i_2} \ldots M_{i_n} = \sum_{i_1=1}^{N-n+1} M_{i_1} \sum_{i_2=i_1+1}^{N-n+2} M_{i_2} \sum_{i_3=i_2+1}^{N-n+3} M_{i_3} \ldots \sum_{i_n=i_{n-1}+1}^{N} M_{i_n} . \qquad (3)$$

Формула (3) дает алгоритм счета сумм (1), который реализуется программным путем при помощи n циклов, вложенных один в другой.

Это создает, во-первых, большое неудобство при составлении стандартной программы вычисления сумм (1) для любых n и N, так как длина ее счетной части переменная (зависит от порядка n) и, во-вторых, для больших n и N время счета на существующих ЭЦМ становится настолько большим, что задача практически не выполнима.

Например, для ЭЦМ со скоростью 20 тыс. операций в секунду с плавающей запятой (которую будем иметь ввиду в дальнейшем), для $N = 36$ и $n = 20$ необходимо затратить 571 час.

Но (1) можно представить и в виде

$$\sum_{i_1<i_2<\ldots<i_n}^{N} M_{i_1} M_{i_2} \ldots M_{i_n} = \sum_{i_1=n}^{N} M_{i_1} \sum_{i_2=n-1}^{i_1-1} M_{i_2} \ldots \sum_{i_n=1}^{i_{n-1}-1} M_{i_n} , \qquad (4)$$

что можно доказать методом полной индукции.

Правую часть (4) можно разбить на 2 слагаемых:

$$\sum_{i_1=n}^{N} M_{i_1} \sum_{i_2=n-1}^{i_1-1} M_{i_2} \ldots \sum_{i_n=1}^{i_{n-1}-1} M_{i_n} = M_N \sum_{i_2=n-1}^{N-1} M_{i_2} \sum_{i_3=n-2}^{i_2-1} M_{i_3} \ldots \sum_{i_n=1}^{i_{n-1}-1} M_{i_n} +$$
$$+ \sum_{i_1=n}^{N-1} M_{i_1} \sum_{i_2=n-1}^{i_1-1} M_{i_2} \ldots \sum_{i_n=1}^{i_{n-1}-1} M_{i_n} = M_N \sum_{i_1<\ldots<i_{n-1}}^{N-1} M_{i_1} M_{i_2} \ldots M_{i_{n-1}} + \sum_{i_1<\ldots<i_n}^{N-1} M_{i_1} M_{i_2} \ldots M_{i_n} . \qquad (5)$$

Разбиение (5) легло в основу предлагаемого алгоритма вычисления сумм (1). Программно он реализуется двумя циклами для любого n и N, путем итерационного перехода от сумм (1) k-го порядка к суммам (1) $k+1$-го порядка, $k = 1, 2, \ldots, n-1$.

Это позволило написать компактную стандартную программу, которая дает возможность считать суммы (1) и для больших n и N.

Для сравнения эффективности указанных методов приводим время счета сумм (1) для некоторых случаев:

$N=36$, $n=20$ – 0,045 сек и 571 час,
$N=20$, $n=10$ – 0,015 сек и 42,4 сек,
$N=12$, $n=5$ – 0,006 сек и 0,138 сек.

§ 2. О вычислении сумм $\sum\limits_{i_1<i_2<\ldots<i_n}^{N} M_{i_1} M_{i_2} \ldots M_{i_n} (C_{i_1}+C_{i_2}+\ldots+C_{i_n})$

Эти суммы получаются аналогично суммам (1), например,

$$\sum_{i_1<i_2}^{4} M_{i_1} M_{i_2} (C_{i_1}+C_{i_2}) = M_1M_2(C_1+C_2) + M_1M_3(C_1+C_3) +$$
$$+ M_1M_4(C_1+C_4) + M_2M_3(C_2+C_3) + M_2M_4(C_2+C_4) +$$
$$+ M_3M_4(C_3+C_4) = C_1M_1(M_2+M_3+M_4) +$$
$$+ C_2M_2(M_1+M_3+M_4) + C_3M_3(M_1+M_2+M_4) + C_4M_4(M_1+M_2+M_3) .$$

При помощи полной индукции можно доказать, что

$$\sum_{i_1<i_2<\ldots<i_n}^{N} M_{i_1} M_{i_2} \ldots M_{i_n} (C_{i_1}+C_{i_2}+\ldots+C_{i_n}) = \sum_{k=1}^{N} C_k M_k \sum_{\substack{i_1<i_2<\ldots<i_{n-1} \\ i_p \neq k}}^{N-1} M_{i_1} M_{i_2} \ldots M_{i_{n-1}} \quad (6),$$

где внутренняя сумма есть сумма вида (1), составленная из тех же элементов $M_1, M_2, \ldots, M_N$ без M_k и имеющая порядок $n-1$.

Из формулы (6) следует, что, имея стандартную программу вычисления сумм (1), можно легко вычислять суммы (2). Время счета сумм (2) с параметрами n и N будет в N раз больше, чем время счета сумм (1) с теми же параметрами.

ГЛАВА ВТОРАЯ

§ 1. Описание алгоритма вычисления элементарных симметрических многочленов

M_i, $i=1,2,\ldots,N$, должны быть расположены в памяти в виде массива.

Необходимо иметь массив рабочих ячеек r_i, $i=0,1,\ldots,N-n+1$, с начальным содержанием $[r_0]=0$, $[r_i]=1$, $i=1,\ldots,N-n+1$.

Сумма (1) получается в ячейке r_{N-n+1}.

Вычислительная схема алгоритма выглядит следующим образом:

СОДЕРЖАНИЕ РАБОЧИХ ЯЧЕЕК ВО ВРЕМЯ СЧЕТА

Ячейки	Первоначальное содержимое	После первой итерации	
r_0	0	0	$= 0$
r_1	1	$[r_1]\times M_1 + [r_0] = M_1$	$= \sum_{i_1}^{1} M_{i_1}$
r_2	1	$[r_2]\times M_2 + [r_1] = M_1 + M_2$	$= \sum_{i_1}^{2} M_{i_1}$
r_k	1	$[r_k]\times M_k + [r_{k-1}] = M_1 + M_2 + \dots + M_k$	$= \sum_{i_1}^{k} M_{i_1}$
r_{N-n+1}	1	$[r_{N-n+1}]\times M_{N-n+1} + [r_{N-n}] = M_1 + M_2 + \dots + M_{N-n+1}$	$= \sum_{i_1}^{N-n+1} M_{i_1}$

Ячейки	После второй итерации	
r_0	0	$=0$
r_1	$[r_1]\times M_2 + [r_0] = M_1 M_2$	$= \sum_{i_1<i_2}^{2} M_{i_1} M_{i_2}$
r_2	$[r_2]\times M_3 + [r_1] = (M_1 + M_2) M_3 + M_1 M_2 = M_1 M_2 + M_1 M_3 + M_2 M_3$	$= \sum_{i_1<i_2}^{3} M_{i_1} M_{i_2}$
r_k	$[r_k]\times M_{k+1} + [r_{k-1}] = (M_1 + \dots + M_k) M_{k+1} + (M_1 M_2 + \dots + M_{k-1} M_k) = M_1 M_2 + M_1 M_3 + \dots + M_k M_{k+1}$	$= \sum_{i_1<i_2}^{k+1} M_{i_1} M_{i_2}$
r_{N-n+1}	$[r_{N-n+1}]\times M_{N-n+2} + [r_{N-n}]$	$= \sum_{i_1<i_2}^{N-n+2} M_{i_1} M_{i_2}$

Ячейки	После n -ой итерации	
r_0	0	$= 0$
r_1	$[r_1]\times M_n + [r_0] = M_1 M_2 \dots M_n$	$= \sum_{i_1<\dots<i_n}^{n} M_{i_1} \dots M_{i_n}$
r_2	$[r_2]\times M_{n+1} + [r_1]$	$= \sum_{i_1<\dots<i_n}^{n+1} M_{i_1} \dots M_{i_n}$
r_k	$[r_k]\times M_{n+k-1} + [r_{k-1}]$	$= \sum_{i_1<\dots<i_n}^{n+k-1} M_{i_1} \dots M_{i_n}$
r_{N-n+1}	$[r_{N-n+1}]\times M_N + [r_{N-n}]$	$= \sum_{i_1<\dots<i_n}^{N} M_{i_1} \dots M_{i_n}$

Описание этого алгоритма даем на языке Алгол-60 /5/, называя соответствующую процедуру ЭСМ (Элементарный Симметрический Многочлен).

```
Procedure ЭСМ (N) Порядок многочлена: (n)  Элементы: (M)
Результат: (S); value N, n ;
integer N, n; real S ; array M ;
begin integer p, q, s, t ;   array r [0:N-n+1] ;
    p: = N - n + 1 ;
    r [0]: = 0 ;
    for q: = 1 step 1 until p do r[q] : = 1;
              comment Произвелась засылка начальных значений в массив r
              рабочих ячеек;
    for s: = 1 step 1 until n do
        for t: = 1 step 1 until p do
            r[t] : = r[t] × M[s+ t - 1]+r[t-1];
    S : = r[p]
end ЭСМ ;
```

§ 2. Описание алгоритма вычисления сумм (2)

Суммы (2) с параметрами n и N подсчитываем на основании (6) при помощи сумм (1) с параметрами $n-1$ и $N-1$, используя при этом основное свойство симметрических функций /1/ по отношению к сомножителям M_i.

Для счета (2), помимо $(N-1)-(n-1)+2 = N-n+2$ рабочих ячеек вычисления сумм (1), необходимы одна рабочая ячейка s и одна ячейка для суммы (2).

Вычислительная схема алгоритма вычисления сумм (2) выглядит следующим образом:

Расположение массива M_i, $i = 1,2,\ldots, N$ в памяти машины во время счета

$[s]_{нач} = 0$

Ячейки	Перед первым шагом	Первый шаг	Перед вторым шагом	Второй шаг
K_1	M_1	$[K_1] \to s$, $[K_N] \to K_1$,	M_N	$[K_2] \to s$, $[K_N] \to K_2$,
K_2	M_2	$[s] \to K_N$,	M_2	$[s] \to K_N$,
$\vdots$	$\vdots$	$[S] + C_1 \times [s] \times \sum\limits_{i_1<\ldots<i_{n-1}}^{N-1} [K_{i_1}] \times [K_{i_2}] \times \ldots \times [K_{i_{n-1}}] =$	$\vdots$	$[S] + C_2 \times [s] \times \sum\limits_{i_1<\ldots<i_{n-1}}^{N-1} [K_{i_1}] \times \ldots \times [K_{i_{n-1}}]$
K_{N-1}	M_{N-1}	$= 0 + C_1 M_1 \sum\limits_{\substack{i_1<\ldots<i_{n-1}\\ i_k \neq 1}}^{N-1} M_{i_1} M_{i_2} \ldots M_{i_{n-1}}$	M_{N-1}	$= C_1 M_1 \sum\limits_{\substack{i_1<\ldots<i_{n-1}\\ i_k \neq 1}}^{N-1} M_{i_1} \ldots M_{i_{n-1}}$
....			....	
K_N	M_N	$\to S$	M_1	$+ C_2 M_2 \sum\limits_{\substack{i_1<\ldots<i_{n-1}\\ i_k \neq 2}}^{N-1} M_{i_1} \ldots M_{i_{n-1}}$ $\to S$

Перед N-ым шагом	Подготовка для N-го шага	Для N-го шага и после него	N-ый шаг
M_N	$[K_1]\to s$;	M_1	$[S]+C_N\times[s]\times \sum\limits_{\substack{i_1<\dots<i_{n-1} \\ i_k\neq N}}^{N-1} M_{i_1}\dots M_{i_{n-1}} =$
M_1	Общий сдвиг:	M_2	
M_2 ⋮ M_{N-2} .... M_{N-1}	$[K_{i+1}]\to[K_i]$, $i=1,\dots,N-1$; $[s]\to K_N$	M_3 ⋮ M_{N-1} .... M_N	$=\sum\limits_{i_1<i_2<\dots<i_n}^{N} M_{i_1}\dots M_{i_n}(C_{i_1}+\dots+C_{i_n})\to S$

Описание алгоритма вычисления суммы (2) даем на языке Алгол-60, называя соответствующую процедуру ЭСМВ (Элементарный Симметрический Многочлен с Весом), делая для большей ясности одно отступление от формализма этого языка: мы привели описание процедуры ЭСМ, которая используется в процедуре ЭСМВ не в описаниях тела процедуры ЭСМВ, как диктует Алгол-60, а отдельно в гл. 2, § 1.

```
Procedure     ЭСМВ (N) Порядок: (n) Элементы: (M) Веса: (C)
Результат: (S) ;     value N, n ;
integer   N, n ;     real S ;  array M, C ;

begin integer p, q ;   real s, S1 ;  procedure ЭСМ ;
    S := 0  ;
    for p := 1 step  1  until N do
        begin
         if  p < N
         then begin s := M[p] ; M[p] := M[N]; M[N] := s  end
         else begin s := M[1];
                  for q := 1 step 1 until N-1  do
                      M[q] := M[q+1] ;
                  M[N] := s
             end                        После этого все M_i стали точно на свои места
                                        как перед началом процедуры ЭСМВ;
      ЭСМ(N-1,  n-1,  M, S1 ) ;
      S := S + C[p] × s × S1

        end
  end ЭСМВ;
```

Примечание. Если среди M_i находятся некоторые очень большие числа, удобно предварительно поделить (если это возможно) все M_i на 2^k, чтобы избежать переполнения. Тогда $\Sigma = 2^{kn}\Sigma'$, где Σ' полученная сумма с введенным масштабом.

В заключение мы предлагаем формулы подсчета количества операций, реализующих предлагаемый алгоритм счета сумм (1) для трехадресной машины с индексным регистром:

$n(N-n+1)$	сложений,
$n(N-n+1)$	умножений и
$(n+2)(N-n+5)$	служебных операций.

Литература

1. Л.Я. Окунев. Высшая алгебра. Учпедгиз, 1958.
2. М.К. Волков, А.Павликовский, Р. Рыбарска, В.Г. Соловьев. О точности расчетов свойств сильно деформированных ядер на основе сверхтекучей модели. Известия АН СССР, сер. физическая, т. XXVII, 7, 1963.
3. Я.Г. Заставенко. Метод вычисления фазовых объемов. Препринт ОИЯИ Р-311, Дубна, 1959.
4. И.Н. Кухтина, Р. Рончка. Исследование K^+-p взаимодействий при больших энергиях с помощью теории множественного решения. Препринт ОИЯИ Р-1462, Дубна, 1963.
5. Сообщение об алгоритмическом языке Алгол-60. ЖВММФ, т.1, № 2, 1961.

Рукопись поступила в издательский отдел
23 декабря 1963 г.