第 11 卷 第 1 期　　电 子 科 学 学 刊　　Vol. 11, No. 1
1989 年 1 月　　JOURNAL OF ELECTRONICS　　Jan., 1989

产生全符号网络函数的 Coates 流图法* **

虞希清　　陆生勋
（杭州大学物理系，杭州）

摘要　本文引入符号码数组、常数数组和记数数组，前二数组用来描述增广矩阵的元素表达式，建立节点导纳方程，后一数组用来写出入度矩阵，然后根据入度矩阵产生 Coates 流图的全部 1-因子增益. 在展开行列式时利用符号代码合并同类项，消去相消项，从而得到无相消项的全符号网络函数.

关键词　线性网络分析；符号网络函数；网络拓扑分析

1. 引言

Mason 流图法是实现全符号网络函数常用的一种方法，典型的程序有 SNAP[1,2]. 这种方法建立的封闭系统拓扑图比较复杂，找回路的方法亦较繁锁，所以效率不高. 本文引入符号码数组、常数数组和记数数组. 前二数组用来描述方程中的元素表达式，建立节点导纳方程，后一数组用来写出入度矩阵，产生 Coates 流图的全部 1-因子. 为了简化计算结果，在行列式展开时，利用上述数组进行分段比较，合并同类项，得到无相消项的全符号网络函数. 实验表明，本方法与 Mason 流图法比较，所编程序在计算机上占用的存储空间较少，运算速度较快.

2. 概念与定理

定义　入度矩阵 B 定义为一个 $n \times (n+1)$ 阶矩阵：若有向图 G 的节点 i ($i = 1, 2, \cdots, n$) 关联 k($k \leqslant n$) 条入边 $(j_1, i), (j_2, i), \cdots, (j_k, i)$，则在 B 的第 i 行中的第 1 至 k 列依增序存放 $j_1, j_2, \cdots, j_k$，第 $k+1$ 列至 n 列存放 0，第 $n+1$ 列存放节点 i 的入度.

定理 1　在有向图 G 的入度矩阵 B 中，从 B 的每行前 n 列中各取一个非零元素并依行序排列在数组 P 中，如果此排列的各元素互不相同，排列所对应的子边集为 E'，则 E' 构成的边子图 (edge subgraph) 是图 G 的一个 1-因子.

证明　若图某一节点 i 的入度 k 为 0，则图中任何自回路和有向回路均不包含此节点，因此，图 G 无 1-因子. 按照入度矩阵定义，当 $0 < k \leqslant n$ 时，数组 P 中非零元素 $P(i)$ 的序数 i 对应边子图中入度为 1 的节点 i. 若 P 中有 n 个不同元素，则这些元素值对应边子图中出度均为 1 的 n 个节点 j_l ($l = 1, 2, \cdots, n$). n 个入度和出度均为 1 的节点所构成的边子图只有三种情况：(1)一条有向回路；(2) m($1 \leqslant m < n$) 条自回路和 l[$1 \leqslant l < (n-m)$] 条有向回路；(3) n 条自回路. 因此，数组 P 中不同元素的排列所对

* 1987 年 11 月 12 日收到，1988 年 4 月 11 日修改定稿.
** 国家自然科学基金资助项目.

应的边子图构成图 G 的一个 1-因子。

设 $G_c(A)$ 是 n 阶方阵 A 的伴随 Coates 流图，$G_c(A)$ 的 1-因子和 1-因子增益分别是 h 和 $f(h)$，则[3]

$$\det A = (-1)^n \sum_h (-1)^{L_h} f(h) \tag{1}$$

其中 L_h 表示 h 中的有向回路数

3. 节点导纳方程的符号编码数组

本文采用文献［4，5］的标准支路来描述节点导纳方程，它的增广矩阵为 $[Y_n I_s]$。$[Y_n I_s]$ 的每一元素都是一个表达式，它由若干项之和构成，而每一项又由常数和符号之积构成：

$$\text{常数} \cdot \text{符号} = b \cdot x$$

其中常数 b 包括正负符号，而符号 x 对应某一元件参数，为了进行符号编码，令

$$\text{符号 } x_1 \longleftrightarrow \text{符号码 } B^0,\quad \text{符号 } x_2 \longleftrightarrow \text{符号码 } B^1,$$
$$\text{符号 } x_3 \longleftrightarrow \text{符号码 } B^2,\quad \cdots\cdots$$

其中基数 B 选自自然数集合 $\{1, 2, \cdots, n\}$，并满足这样条件：增广矩阵中任一符号或符号之积在不同的行和不同的列最多出现 $B-1$ 次。

当 $b_i \cdot x_i$ 和 $b_j \cdot x_j$ 相乘时，按以下形式规定 $(b_i \cdot x_i) \cdot (b_j \cdot x_j)$ 的常数和符号码：

$$\text{常数} = b_i \cdot b_j,\quad \text{符号码} = B^{i-1} + B^{j-1}$$

例 1　图 1 所示的网络，增广矩阵为：

$$\begin{bmatrix} Y_4+Y_1+Y_5 & -Y_5 & -Y_1 & I_s \\ -Y_5-G_m & Y_2+Y_3+Y_5+G_m & -Y_2 & 0 \\ -Y_1+G_m & -Y_2-G_m & Y_1+Y_2+Y_6 & 0 \end{bmatrix} \tag{2}$$

矩阵中，任一符号在不同的行和列最多出现 2 次，故 $B-1=2$，即 $B=3$。网络元件符号编码的各常数 b_i 均为 1，各符号码如表 1 所示。$(-Y_5)\cdot(-Y_2)\cdot(-Y_1)$ 的符号编码为：

$$\text{常数} = (-1)\cdot(-1)\cdot(-1) = -1,\quad \text{符号码} = 3^4 + 3^1 + 3^0 = 85.$$

容易看出以上符号编码经数值运算后，可以再翻译成全符号的表达式，而且只要 B 的选择符合前面的条件，译码后的结果是唯一的。

表 1

符　号	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	I_s	G_m
符号码	3^0	3^1	3^2	3^3	3^4	3^5	3^6	3^7

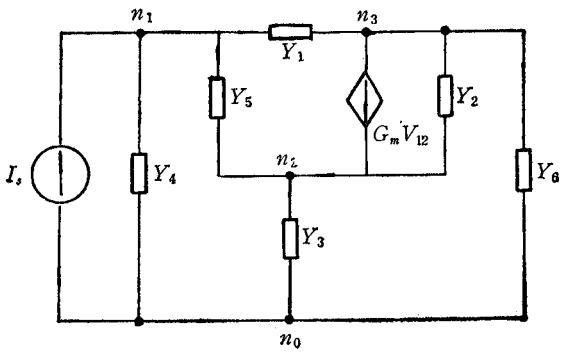

图 1

下面我们给出三个定义，使符号编码能完全表示节点导纳方程。

定义 1 符号码数组 A 为 $n\times(n+1)\times m$ 阶数组，数组的第一维和第二维与增广矩阵的行和列一一对应，第三维存放增广矩阵元素表达式中各项的符号码 x_i.

定义 2 常数数组 $A1$ 为 $n\times(n+1)\times m$ 阶数组，数组的第一维和第二维与增广矩阵的行和列一一对应. 第三维存放增广矩阵元素表达式中各项的常数 b_i（包括正负号）.

定义 3 记数数组 $A2$ 为 $n\times(n+1)$ 阶数组，它的行和列与增广矩阵的行和列一一对应，元素值表示增广矩阵相应的元素表达式的项数，增广矩阵中零元素的相应记数数组元素为零.

为叙述方便起见，由符号码数组或常数数组的第一维下标 i 和第二维下标 j 确定的那些符号码 $A(i,j,k)$ 或常数 $A1(i,j,k)(k=1,2,\cdots,A2(i,j))$ 称为数组在位置 (i,j) 纵向存放的符号码或常数.

例 2 增广矩阵(2)中的元素 $Y_n(1,1)$, $Y_n(2,3)$ 和 $Y_n(3,2)$ 对应的符号码数组和常数数组在位置 (1,1),(2,3) 和 (3,2) 纵向存放的符号码和常数如表 2 所示. 上述三个元素表达式的项数分别为 3，1 和 2，故记数数组的元素 $A2(1,1)=3$, $A2(2,3)=1$, $A2(3,2)=2$.

表 2

表达式	$Y_n(1,1)=+Y_4+Y_1+Y_5$				$Y_n(2,3)=-Y_2$		$Y_n(3,2)=-Y_2-G_m$		
第三维下标	$k=$	1	2	3	$k=$	1	$k=$	1	2
符号码数组	$A(1,1,k)$	3^3	3^0	3^4	$A(2,3,k)$	3^1	$A(3,2,k)$	3^1	3^7
常数数组	$A1(1,1,k)$	+1	+1	+1	$A1(2,3,k)$	−1	$A1(3,2,k)$	−1	−1

4. 行列式展开的算法和同类项的合并

根据记数数组 $A2$ 的定义，若 $A2(i,j)\neq 0$，则自节点 j 到 i 有一条权为 $Y_n(i,j)$ 的有向边，因此，从 $A2$ 可作出增广矩阵的伴随 Coates 流图和其入度矩阵. 根据定理 1，可写出求 Coates 流图全部 1-因子的算法流程图.

图 2 中，LP 为行指针，其值指向入度矩阵 B 的行，IP 为 $n\times 1$ 阶列指针数组，每行的值指向 B 的相应行的列号. 例如 $IP(3)=2$ 指向 B 的第 3 行第 2 列. P 为 $u\times n$ 阶数组（u 为 1-因子个数），存放 1-因子信息. 初始条件为 $IP(i)=1(i=1,2,\cdots,n)$.

有了 Coates 流图的全部 1-因子，根据公式 (1)，可得任一行列式的全符号展开式，(1)式中，$f(h)$ 是 1-因子全部边的权之乘积，即行列式的元素表达式之积. 为了简化行列式的展开式，须先展开 $f(h)$，再合并同类项. 下面，举例说明展开 $f(h)$ 的方法，其实，结果是有普遍性的，容易推广到一般情况.

增广矩阵 (2) 的 Y_n 的伴随 Coates 流图及其一个 1-因子如图 3，4 所示，1-因子增益展开式为:

$$\begin{aligned} f(h) &= Y_n(1,1)\cdot Y_n(2,3)\cdot Y_n(3,2) \\ &= Y_4Y_2^2+Y_4Y_2G_m+Y_1Y_2^2+Y_1Y_2G_m+Y_5Y_2^2+Y_5Y_2G_m \end{aligned} \tag{3}$$

表 3 示出了符号码数组在位置(1,1),(2,3)和(3,2)纵向存放的符号码和对应的元素表达

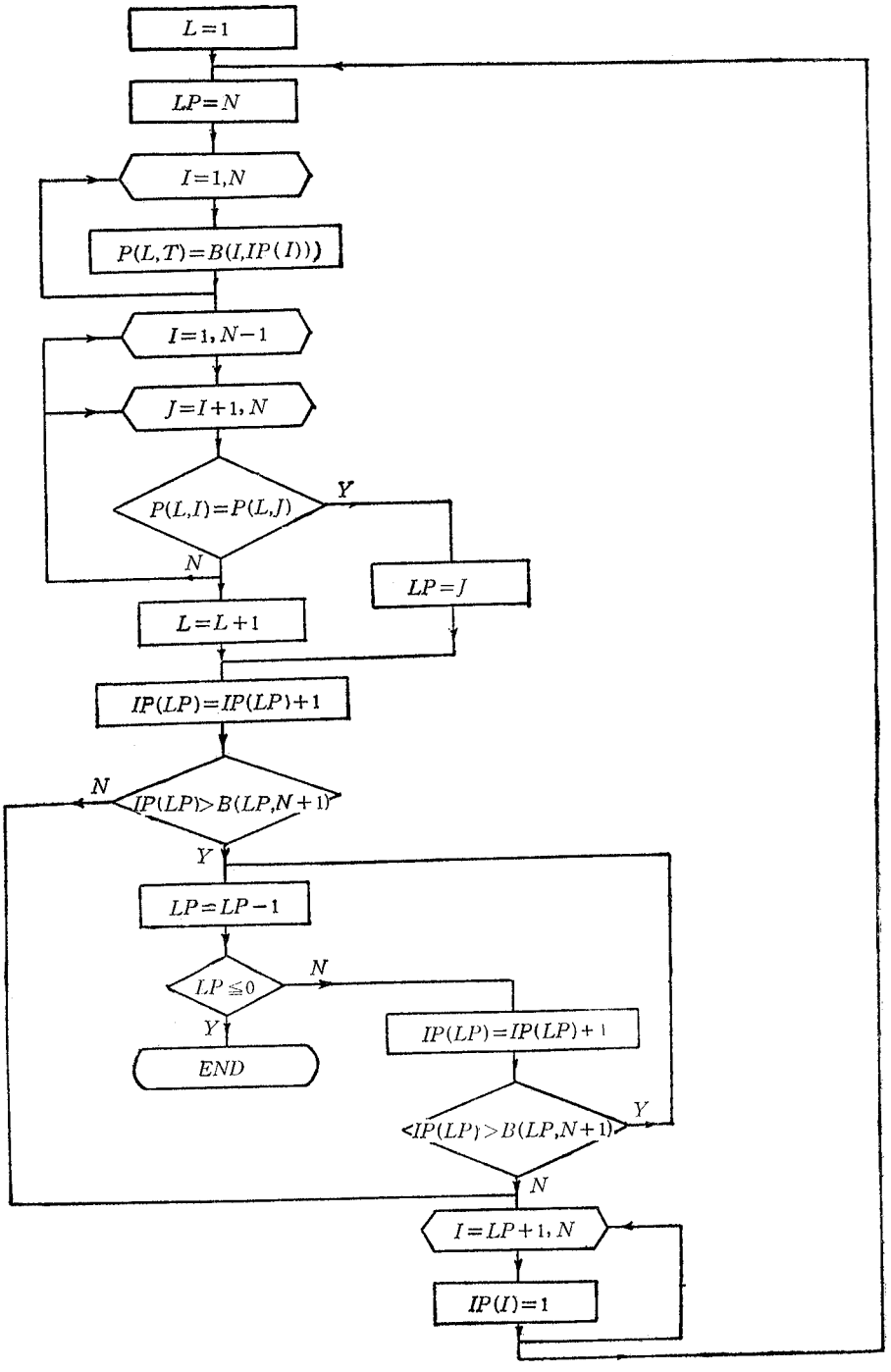

图 2　求 Coates 流图 1-因子的算法流程图

式中各项。显然，(3)式中各符号项等于表 3 中每行各取一个符号相乘，对应的符号码为每行各取一个符号码相加：

$$Y_4Y_2^2 \longleftrightarrow 3^3+3^1+3^1=33, \qquad Y_4Y_2G_m \longleftrightarrow 3^3+3^1+3^7=2217,$$

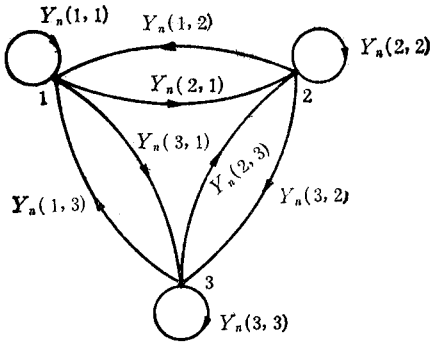

图 3

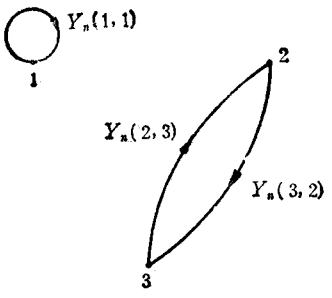

图 4

表 3

	$k=1$	$k=2$	$k=3$	
$Y_n(1,1)$ $A(1,1,k)$	Y_4 3^3	Y_1 3^0	Y_5 3^4	$A_2(1,1)$ $=3$
$Y_n(2,3)$ $A(2,3,k)$	Y_2 3^1			$A_2(2,3)$ $=1$
$Y_n(3,2)$ $A(3,2,k)$	Y_2 3^1	G_m 3^7		$A_2(3,2)$ $=2$

$$Y_1Y_2^2 \longleftrightarrow 3^0+3^1+3^1=7, \qquad Y_1Y_2G_m \longleftrightarrow 3^0+3^1+3^7=2191,$$
$$Y_5Y_2^2 \longleftrightarrow 3^4+3^1+3^1=87, \qquad Y_5Y_2G_m \longleftrightarrow 3^4+3^1+3^7=2271.$$

常数的处理方法和以上类似，只要将相加改为相乘。

按上述方法展开行列式的全部 1-因子增益后，将展开函数式中各项的符号码和常数存放在 $1\times L$ 阶数组 IC 和 $IC1$ 中，此处 L 为函数式的项数，用比较法进行合并和消项。

(1) 直接比较法　直接比较法是直接将 $IC(i)$ $(i=1, 2, \cdots, L-1)$ 与 $IC(j)$ $(j=i+1, i+2, \cdots, L)$ 进行比较，若 $IC(i)=IC(j)$，合并同类项，令 $IC1(i)=IC1(i)+IC1(j)$，消去相消项，令 $IC(j)=0$，$IC1(j)=0$. 这样，合并了展开式中的同类项，也消去了全部相消项。

直接比较法在计算机上的运算次数约为 $L^2/2$，随着网络规模的扩大，展开式的项数 L 按指数律增加，运算时间越来越长。

(2) 分段比较法　为提高运算速度，我们求出 L 项符号码的平均值，将符号码分成小于等于平均值和大于平均值二个部份，常数也随对应的符号码分段。然后对这二部份采用直接比较法进行合并和消项。这时的运算次数约为 $2\times(L/2)^2/2=L^2/4$. 如果 L 的值很大，还可按上面的方法继续分段比较。

最后，将非零的符号编码进行译码，就得到全符号网络函数。

5. 计算机程序和比较

上述方法已用 FORTRAN 语言编写成程序，在 IBM PC 机上调试通过，运行结果令人满意。

例 3　对于图 1 所示的网络，本程序可以解出任意一个节点的电压，例如节点 3 的电压为：

$$V_3 = \text{numerator } 3/\text{denominator}$$

$$\begin{aligned}\text{denominator} = {} & +Y_5Y_4Y_1 + Y_5Y_4Y_2 + Y_6Y_5Y_4 + Y_4Y_2Y_1 + Y_6Y_4Y_2 + Y_4Y_3Y_1 + Y_4Y_3Y_2 \\ & + Y_6Y_4Y_3 + G_mY_4Y_1 + G_mY_6Y_4 + Y_6Y_5Y_1 + Y_6Y_2Y_1 + Y_3Y_2Y_1 \\ & + Y_6Y_3Y_1 + G_mY_6Y_1 + Y_6Y_5Y_2 + Y_5Y_3Y_1 + Y_5Y_3Y_2 + Y_6Y_5Y_3 + G_mY_3Y_1\end{aligned}$$

$$\text{numerator}3 = I_sY_5Y_2 + I_sY_5Y_1 + I_sY_2Y_1 + I_sY_3Y_1 - G_mI_sY_3 + G_mI_sY_1$$

下面就本文的方法和 Mason 流图法[2]进行比较：

(1) 存储空间　对于图 5 所示的梯形网络，二者所需的存储空间见表 4。

表 4 （以单元为单位）

网络节点数		3	4	5	6
所需存储空间	Mason 流图法	97	288	816	4324
	Coates 流图法	34	117	200	319

表 5 （以秒为单位）

网络节点数		3	4	5	6
机器执行时间	Mason 流图法	1.13	1.84	3.98	>130.00
	Coates 流图法	1.04	1.42	2.14	3.24

(2) 运行时间　由于本程序采用节点导纳方程和 Coates 流图，运行时间比 Mason 流图法快得多。仍以图 5 为例，二者所需的时间见表 5。

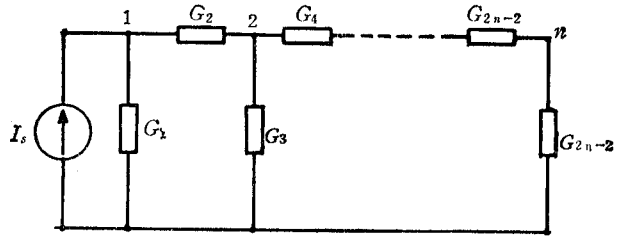

图 5

对于较大的网络，可先分解，然后以本文方法为基础进行运算，拟另写一文。

参　考　文　献

[1] P. M. Lin, G. E. Alderson, SNAP-A Computer Program for Generating Symbolic Network Function, School Elec. Eng., Purdue Univ., Lafayette, Ind., Rep. TR-EE70-16, Aug. 1970.

[2] 张惠廉，庄镇泉，电子线路的计算机辅助设计，人民教育出版社，1979，下册，244—259.

[3] W. K. Chen, Applied Graph Theory, North-Holland, 1976, p. 144.

[4] P. R. Adby, Applied Circuit Theory, Matrix and Computer Methods, Ellis Horwood Limited, 1980, Chap. 4.

[5] 陈树柏，左垲，张良震，网络图论及其应用，科学出版社，1982，第 5 章.

COATES GRAPH APPROACH FOR GENERATING SYMBOLIC NETWORK FUNCTION

Yu Xiqing Lu Shengxun

(Department of Physics, Hangzhou University, Hangzhou)

Abstract Symbolic code matrix, constant matrix and count matrix are defined. The first two matrixes are used to describe the elemental expressions of augmentation matrix and the node admittance equation is thus obtained. The third matrix is used to obtain the incoming degree matrix, and all the 1-factor gains of the coates graph are given according to the matrix. Using the code data, the determinant is expanded and all the same items in the expansion are merged. Thus the symbolic network function in which no term concellation occurs is generated.

Key words Linear network analysis; Symbolic network function; Topological network analysis