结构动静力分析原理
与 CAL 语言

张楚汉　王光纶　编著

机 械 工 业 出 版 社

结构动静力分析原理
与CAL语言

张楚汉　王光纶　编著

机 械 工 业 出 版 社

目　录

第一章　概　　述

§1－1　CAL语言的目的与来由

CAL语言是“Computer Analysis Language for the Static and Dynamic Analysis of Structural System”的缩写，它是美国加州大学伯克莱分校的E. Wilson教授1978年提出来的一个土木结构工程动静力分析的计算机语言，主要供教学使用。目前伯克莱分校土木系的研究生已广泛地使用这一语言进行课程练习的演算。由于它的灵活性和运用简便，许多师生也用CAL来进行中小型科研问题的计算。甚至对大型结构问题简化模型的初步研究，有时也使用CAL语言。因此，它已成为该校土木结构工程大学生与研究生普遍掌握和使用的一个计算语言。

CAL语言是在1963年研制出来的矩阵符号解释系统SMIS (Symbolic Matrix Interpretive System)基础上改进、发展起来的。其目的是为了在传统的结构分析教学方法与完全自动化的结构分析程序之间建立一座桥梁。美国多年的教学经验说明，对于一个结构工程师的培养来说，现行的自动化分析程序，可以使庞大的结构计算变成只填写若干控制卡片及数据卡片的简单过程，但是如果学生不懂得结构分析的基本假定与力学模型而单纯地使用这些自动化“暗盒”，他们就无法解释计算机计算成果的合理性，无法发现计算中的问题。这将使学生变成使用计算机的工具，而不是操纵计算机的主人。有鉴于此，伯克莱大学的《结构静动力分析》课程都很强调直接刚度法的教学，让学生掌握好结构单元力与变形的刚度分析以及刚度集成的全过程。这样一来，又出现了另一个矛盾。即如果让学生从形成单元刚度开始，一直到算出结构内力都自己动手编制程序，则将消耗大量的时间在繁复的程序编制过程中，反而影响对力学概念的理解。为了解决上述矛盾，E. Wilson教授根据多年的教学经验以及计算机程序方面的特长，提出并编制了CAL这一计算机教学语言，学生在使用CAL过程中必须具备结构分析中单元刚度分析与刚度集成的概念，而又避免让学生去编写繁杂的程序，这就较好地解决了上述矛盾。这就是CAL语言的目的及来由。

§1－2　CAL语言的功能与结构

CAL语言是用FORTRAN－Ⅳ编写的。它把结构动静力分析的过程拆成一系列可由用户控制的运算。每个运算由一个或多个子程序组成，运算的名字由字符串组成并作为数据输入到内存中与源程序中的运算名字进行比较，符合后，转去执行一系列的子程序。CAL-78程序可以在几种不同类型的大中型计算机上工作(如CDC－6400,CDC－7600,IBM－370等)，也可以在16位的小型机上使用；可用于批量的方式，也可以用时间分享的方式。

CAL运算的功能包括：①一般矩阵运算；②结构静力分析计算；③结构动力分析计算。在结构静力分析中包括了平面刚架，三维刚架，三维桁架，平面应力与应变的有限单元法。在结构动力分析中，包括了地震荷载的等距内插，结构振型和频率分析，直接逐步积分法，振型迭加

法及反应谱方法等。除了频域方法(Frequency Domain)外,几乎包括了所有主要的结构动力分析方法。

CAL 语言的主要局限性是:①程序没有有效地使用低速存储,矩阵的存储(包括零元素在内)都用满阵的方式存储,没有利用对称性及带状的特点。考虑到 CAL 主要用于教学,对自由度小于 100 的小型问题来说,这一限制是不严重的。②在动力分析中,特征值及特征向量的求法是采用雅可比方法,对 n 阶的广义特征值问题同时给出 n 个特征值与特征向量,对 n 很大的大型结构问题,这当然是费机时而又不必要的。因为对结构反应来说,具有显著贡献的总是前几个振型,充其量说前十多个振型一般已经足够。因此求大型结构问题的特征值最好采用子空间迭代法更为有效。但是同样地,对小型的教学课题来说,这一局限性也是不严重的。

下面将 CAL 的主要运算及其功能列表如下:

1. 一般矩阵运算

START——运算开始,删除前面装入或生成的所有数组

STOP——终止 CAL 程序

LOAD——装入数组

PRINT——打印数组

ZERO——生成零阵或单位阵

DUP——复制数组

ADD——矩阵相加

SUB——矩阵相减

MULT——矩阵相乘

TRAN——矩阵转置

SOLVE——矩阵求逆或解方程

DUPSM——从母阵中形成新的子矩阵

STOSM——将子阵置于母阵中

DUPDG——用矩阵的对角线项形成行阵

STODG——以行(或列)阵置于矩阵的对角线上

MAX——以各行最大绝对值形成新的列阵

NORM——以各列绝对值之和或平方和的平方根形成新的行阵

INVEL——以各项倒数形成新的矩阵

SQREL——以各项的平方根形成新的矩阵

LOG——以各项的自然对数形成新的矩阵

DELETE——删除数组

2. 直接刚度法运算

SLOPE——对水平梁或铅直柱单元用挠曲方程形成 4×4 的刚度矩阵,不考虑轴向变形

FRAME——对二维桁架单元形成 6×6 的刚度矩阵,单元轴线可具有任意方向,并考虑轴向变形

LOADI——形成一个与节点编码有关的整型数组

ADDK——刚度集成运算

MEMFRC——单元节点力运算

3. 结构静力分析运算

NODES——节点总数，座标与编码信息

BOUND——边界条件信息

BEAM——形成三维框架单元的刚度、质量及力-位移转换矩阵

TRUSS——形成三维桁架单元的刚度、质量及力-位移转换矩阵

PLANE——对平面应力与应变问题形成单元刚度、质量及应力-位移转换矩阵。单元节点有 3～8 个，由用户自选

LOADS——形成荷载矩阵

ADDSF——节点总刚度与总质量矩阵的集成

DISPL——打印节点位移

FORCE——计算及打印单元力

4. 结构系统的动力分析运算

FUNG——在一个时间函数 $f(t)$ 内进行插值运算

STEP——在指定参数条件下进行逐步积分法运算

EIGEN——特征值及特征向量的运算，用于振型迭加法

DYNAM——在指定时间荷载 $P(t)$ 的条件下，求解二阶线性微分方程，用于振型迭加法

PLOT——为指定的数组绘图

5. 循环运算

LOOP 和 NEXT——循环运算一共具有五层循环能力

SKIP——用于在同一层循环中跳过某些运算

6. 用户自定义运算

USERA
USERB ——为用户保留的自定义运算

在以上六类运算中，CAL 语言的 FORTRAN 源程序把它们分为三组，以便缩短主程序的长度。这三组是①Group 1，一般矩阵运算；②Group 2，结构静力计算；③Group 3，结构动力计算。它们之间的关系，如图 1－1，图 1－2，图 1－3，图 1－4 所示。

CAL-78 程序的运行是通过输入一系列的运算卡和数据卡来实现的。数据卡通常是跟在

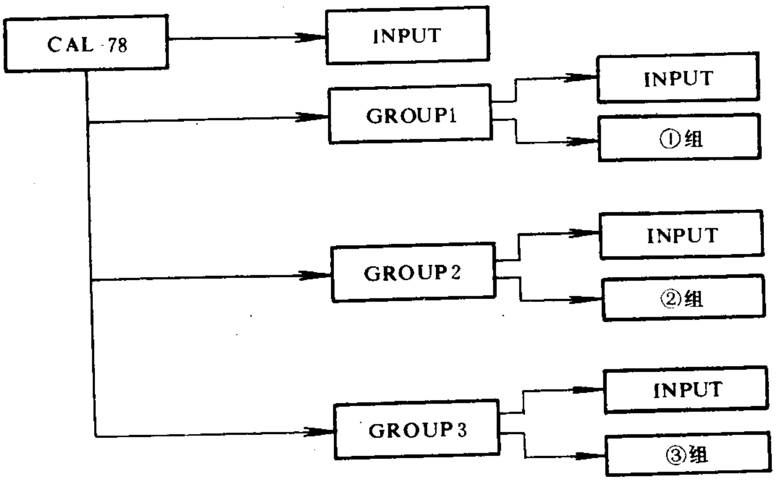

图 1－1 CAL－78 框图

需要输入数据的运算卡之后，而运算卡的一般形式是：

OP,M1,M2,M3,……,MI,N1,……,NI

其中 OP 是要执行的运算名。MI 是数组名，NI 是整数。这里特别要说明的是：CAL 版本由运用于 CDC—6400 计算机，改为适用于 IBM 系列的计算机时，字符与数字分别采用了左、右移的规则。因此，目前在 M—150F 机或 IBM—PC 微机内装上的 CAL 程序只能使用字符串组成的数组名，例如 MA，MB，……MI 等（I＝A、B、C……），而不能使用由字符与数字组成的数组名，如 M1，M2……MI 等（I＝1、2、3、……）。但为了叙述上的方便，本书仍然使用 M1,M2,……MI 作为数组名来进行叙述，请读者在使用 CAL 时特别注意。

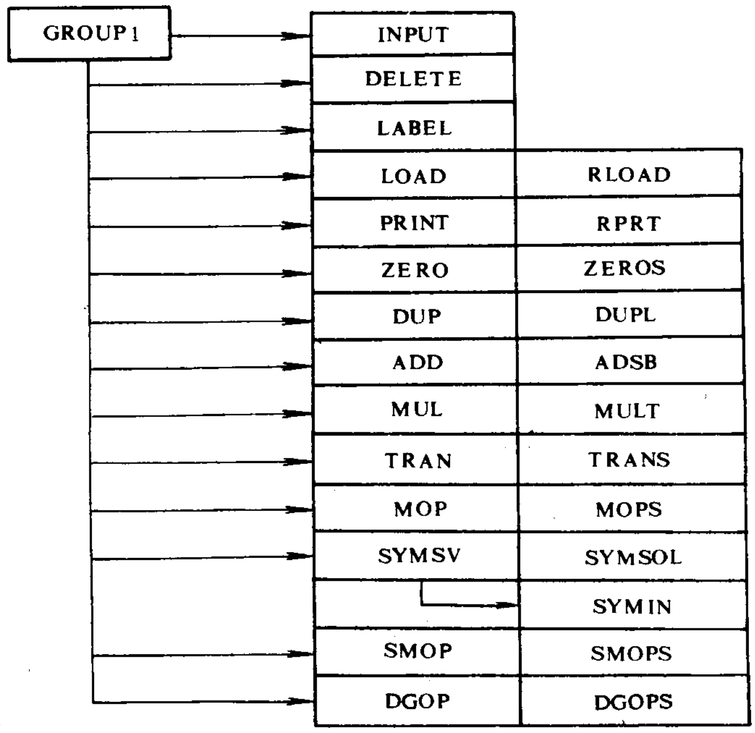

图 1—2　GROUP1 框图

CAL-78 内部数组的存储方式是采用一维动态存储。一维数组指定为 COMMON L（××××），此处××××是为所有数组保留的总存储量。在每个数组前面有一个六字的索引，内容包括了数组元素的总数，数组的列数，数组的名字，每一项要求的存储位置数及存储方式等。因此每个数组所要求的位置总数应是数组元素所占位置总数加 6。CAL 的数据管理系统包括使用下列三个子程序：

```
CALL  LIST  (NM,NA,NR,NC,NP)
CALL  LOCATE (NM,NA,NR,NC)
CALL  DELETE (NM)
```

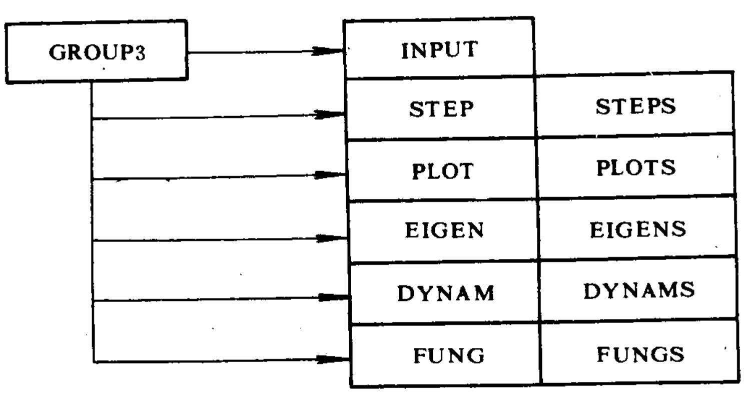

图 1—3　GROUP 3 框图

其中，NM 是数组名；NA 是数组 NM 中第一项元素在 L 中的位置；NR 是 NM 的行数；NC 是 NM

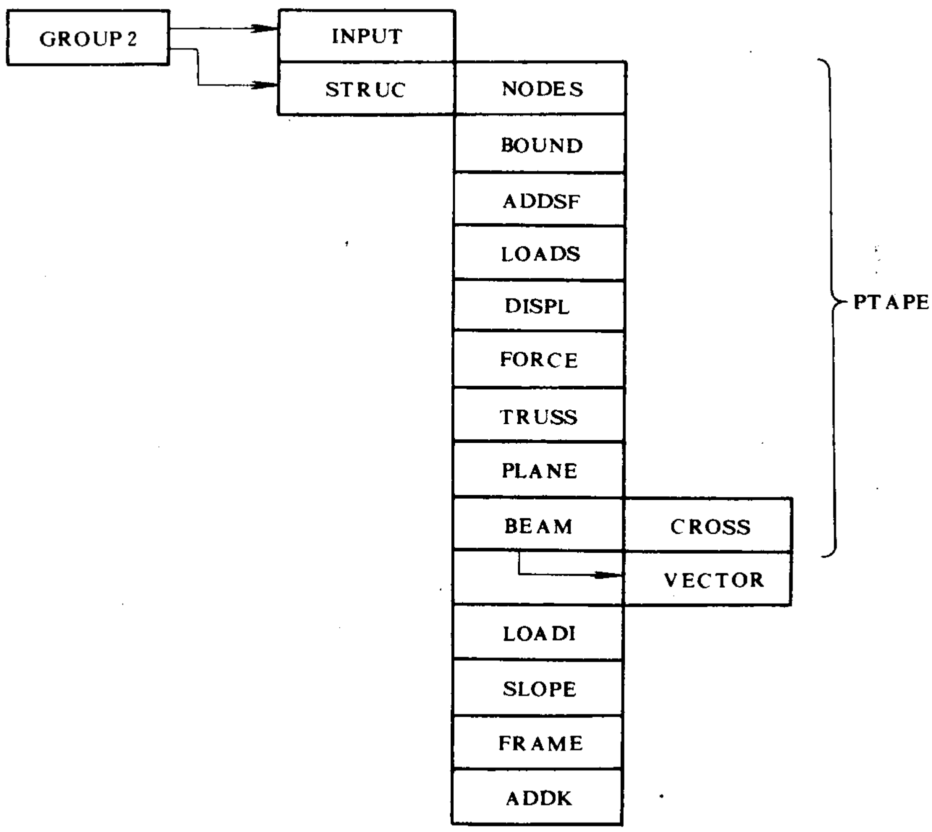

图 1－4　GROUP 2 框图

的列数;NP 是 NM 中每项元素要求的存储位置数。

如图 1－5 所示，子程序 LIST 为新定义的数组 NM 保留存储。子程序 LOCATE 用于查找 NM 数组在 L 中的起始位置(NA)和数组的行数(NR)与列数(NC)。子程序 DELETE 用于删除 NM 数组及其索引。因此，在 L 数组的末尾允许重新使用由子程序 DELETE 所释放的全部存储。这一数据管理方法对数组总数不大的教学问题来说，是非常简单而有效的。

在伯克莱分校结构工程的研究生课程中，比较广泛地使用 CAL 语言来进行作业演习的课程有：高等结构理论，结构动力学，有限单元法理论及结构工程中的数值方法等。学生在学习这些课程以前已具备了 FORTRAN 语言的一般知识，能编制中、小规模的计算机程序。CAL 语言本身的讲授并不单独设课，而是与高等结构理论的教学紧密地结合起来进行。例如，在讲授典型杆件的刚度矩阵分析时，同时就讲解有关的 CAL 子程序(如 SLOPE,FRAME,TRUSS 等)。在讲授刚度集成法概念时，同时就解剖集成法的子程序(如 ADDK,ADDSF 等)。

经验表明，CAL 语言的使用大大地推动了土木结构工程的教学工作。为学生掌握结构动静力分析理论及大型通用计算机程序奠定了良好的基础。另一方面，师生们对比较复杂的结构问题进行研究时，往往首先选用 CAL 进行简化模型的计算。这对科学研究的课题也起了一定的桥梁作用。

我国高等院校土木结构，水工结构，船舶结构及固体力学专业的师生们正在越来越普遍地把计算机用于教学、科学研究工作，现把 CAL 语言引进并推广到我们的教学过程中，无疑将加速这一日益发展的过程。

CAL 仍处于发展阶段，有许多地方仍需补充和发展，如最近在伯克莱分校又增加了一个随机振动计算的功能。这一功能今后也将补充到我们已经运行的 CAL 语言中。

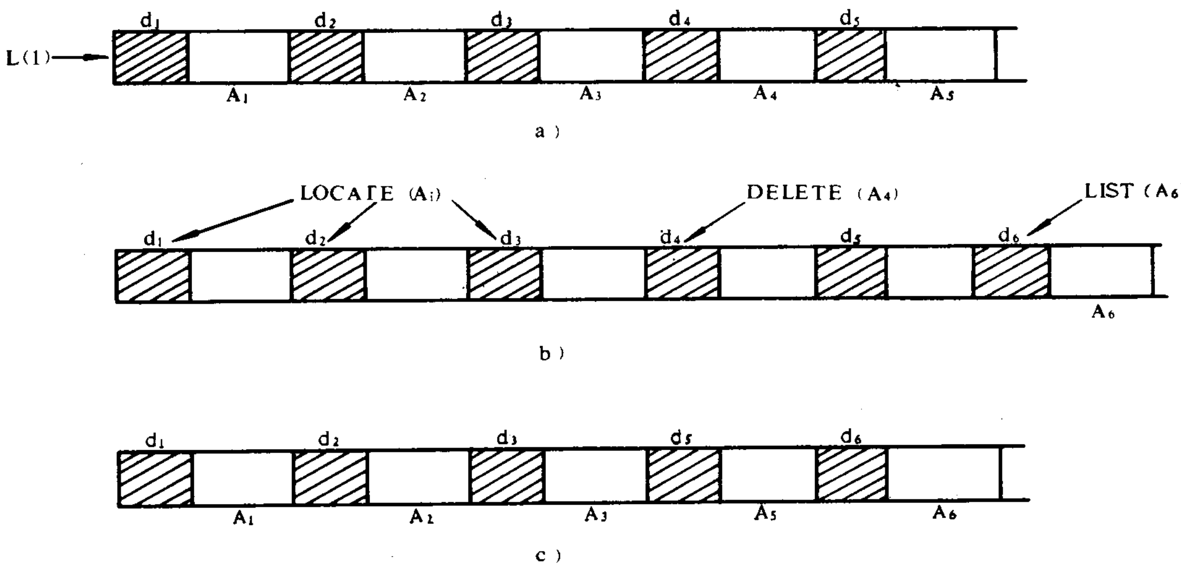

图 1—5　CAL 的数据管理系统示意

a)数组 A_i 及索引 d_i 排列情形　b)查找 A_i，定义 A_6 及删除 A_4　c)删除 A_4 以后的数组排列情形

第二章　一般矩阵运算

§2－1　CAL运算符使用说明

1. START

这一运算将消除前面装入或生成的所有数组。通常在一个CAL运算的开始，要设置这个运算符。

2. STOP

这一运算使CAL－78程序停止执行。

3. LOAD, M1⁺①, N1, N2, N3

本运算将生成一个实型数组M1，M1具有N1行和N2列。数组元素应按行式输入，数据卡紧随运算卡之后。如果N3＝0或空白，则数据输入格式为(8F10.0)；如果N3≠0，则在运算卡之后，数据卡之前，应加入附加的数据格式卡片。例如，输入格式是每张卡片4个数据，每个数据占15位，则格式卡片应为(4F15.0)。

举例　装入矩阵

$$A=\begin{bmatrix}2 & 1\\ 3 & 0\end{bmatrix}$$

其CAL运算卡为(注意：所有运算名均从卡片第一列输入)：

```
LOAD,A,2,2,1
(4F2.0)
2.1.3.0.
```

4. ZERO, M1⁺, N1, N2, N3, N4

本运算将生成一个具有N1行和N2列的零矩阵或单位阵。其中，M1(I,I)＝N3(I＝1,2,……N1)，M1(I,J)＝N4(I＝1,2,…N1；J＝1,2,…N2；I≠J)。

举例　生成一个3×3的单位阵，其CAL运算卡为：

```
ZERO,A,3,3,1,0
```

5. NO

YES

---

①〔注〕"＋"号上角表示该数组是新生成的数组，前面已定义的同名数组将被删除。
"－"号上角表示该数组经运算后，内容将被改变，数组名没有变化。

这两个运算是控制打印输出的，NO 表示免去其后的一切打印，一直到碰见 YES，再重新进行打印。这两个运算主要用于循环迭代运算，迭代结果未满足误差要求时，用 NO 运算，以避免打印冗长的结果，一直到迭代结果符合要求时，用 YES 运算把结果打印出来。

6. PRINT,M1,N1

本运算将打印数组 M1，打印格式为：当矩阵阶数<8 时，按原有的行与列打印。如矩阵阶数>8 时，每行打出 8 列数据。当 N1>0 时，运算将读入并打印出 N1 张说明卡。说明卡置于运算卡之后。

7. LABEL,N1

本运算将读入并打印 N1 张说明卡，每张说明卡的第一列代表打印格式的符号。例如，0 代表隔行打印，1 代表换页打印。

8. DUP,M1, M2⁺

本运算复制一个与原矩阵 M1 完全相同的矩阵 M2。

9. ADD, M1⁻ ,M2

本运算将矩阵 M1 与 M2 之和送入 M1 中，故原来的矩阵 M1 将被冲掉。

举例 已知

$$\boldsymbol{A}=\begin{bmatrix}-4 & 2\\ 6 & 6\end{bmatrix} \qquad \boldsymbol{B}=\begin{bmatrix}2 & -1\\ -1 & 4\end{bmatrix}$$

求 $\boldsymbol{A}+\boldsymbol{B}=?$

CAL 运算卡及数据卡如下：

```
START
LOAD,A,2,2,1
(4F5.0)
 -4.   2.   6.   6.
PRINT,A
LOAD,B,2,2,1
(2F5.0)
   2.0      -1.0
  -1.0       4.0
PRINT,B
ADD,A,B
PRINT,A
STOP
```

《CAL 运算输出结果》

矩阵 $\boldsymbol{A}+\boldsymbol{B}$ 计算的结果为：

★★PRINT,A

	1	2
1	—.2000000E+01	.1000000E+01
2	.5000000E+01	.1000000E+02

10. SUB, M1⁻ ,M2

本运算将矩阵 M1 与 M2 之差送入 M1 中,原来的矩阵 M1 将被冲掉。

11. MULT,M1,M2, M3⁺

本运算将生成矩阵 M3,它是矩阵 M1 与 M2 的乘积,即:

$$M3 = M1 \cdot M2$$

12. TRAN,M1, M2⁺

本运算将生成矩阵 M2,它是矩阵 M1 的转置,即:

$$M2 = M1^T$$

举例 计算 $X = A^TA$

$$A = \begin{bmatrix} -5 & 2 & 6 & 8 \\ 4 & -9 & 2 & 10 \end{bmatrix}$$

CAL 运算卡及数据卡如下:

```
START
LOAD,A,2,4,1
(4F5.0)
        -5.     2.     6.     8.
        4.     -9.     2.     10.
TRAN,A,B
MULT,B,A,C
PRINT,C
STOP
```

《CAL 运算输出结果》

矩阵 X 的计算结果为:

★★PRINT,C

	1	2	3	4
1	.4100000E+02	—.4600000E+02	—.2200000E+02	.0000000E+00
2	—.4600000E+02	.8500000E+02	—.6000000E+01	—.7400000E+02
3	—.2200000E+02	—.6000000E+01	.4000000E+02	.6800000E+02
4	.0000000E+00	—.7400000E+02	.6800000E+02	.1640000E+03

13. SCALE, M1⁻ ,M2

本运算用 M2 中的第一个元素 M2(1,1)乘以矩阵 M1,并将结果送入 M1 中,即:

$$M1 = M2(1, 1) \cdot M1$$

14. **SOLVE**, M1^{-},M2^{-} **或**

SOLVE, M1^{-},N1 **或**

SOLVE, M1^{-},M2^{-},N1

当 N1 = 0 时,本运算解方程组 $AX = B$,其中,M1 相当于对称矩阵 A,M2 相当于矩阵 B。矩阵 A 将被三角分解,并将结果 X 存于原来储存 B 的 M2 中,故原来的 M1 和 M2 将被冲掉。

当 N1=1 时,只将矩阵 A 进行三角分解,不求解方程。

当 N1=2 时,依据前面已进行三角分解的矩阵 A,和新送入的矩阵 B,解方程 $AX = B$,求 X,并将结果 X 送入 B 中。

当 N1 = 3 时,则将矩阵 A 求逆,并将逆矩阵送入 A 中。

当 N1 = 0或 = 1 时,矩阵 A 被分解成 LDL^T 形式,对角阵 D 储存于 A 的对角线项上。

举例　已知

$$\text{矩阵} \quad A = \begin{bmatrix} 5 & 4 & 3 \\ 4 & 7 & 4 \\ 3 & 4 & 4 \end{bmatrix}$$

$$\text{向量} \quad B = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$$

求解:(1)将矩阵 A 三角分解

(2)将矩阵 A 求逆

(3)解方程 $AX = B$

CAL 运算卡及数据卡如下:

```
START
LOAD,A,3,3,1
(3F2.0)
5. 4. 3.
4. 7. 4.
3. 4. 4.
PRINT,A
LOAD,B,3,1,1
(1F3.0)
2.0
-1.
3.0
PRINT,B
DUP,A,AA
SOLVE,AA,1
PRINT,AA
DUP,A,AAA
```

```
SOLVE,AAA,3
PRINT,AAA
SOLVE,A,B
PRINT,B
STOP
```

《CAL 运算输出结果》

矩阵 A 三角分解的结果为：

★★PRINT,AA

	1	2	3
1	.5000000E+01	.8000000E+00	.6000000E+00
2	.4000000E+01	.3800000E+01	.4210527E+00
3	.3000000E+01	.1600000E+01	.1526316E+01

矩阵 A 求逆的结果为：

★★PRINT,AAA

	1	2	3
1	.4137931E+00	−.1379310E+00	−.1724138E+00
2	−.1379310E+00	.3793104E+00	−.2758622E+00
3	−.1724138E+00	−.2758622E+00	.6551725E+00

解方程 X 的结果为：

★★PRINT,B

	1
1	.4482756E+00
2	−.1482759E+01
3	.1896552E+01

15. DUPSM,M1, M2⁺,N1,N2,N3,N4

本运算从母阵 M1 中形成子阵 M2,其中 M2 是 N3 行和 N4 列的矩阵,M2 的第一行第一列将是 M1 的第 N1 行第 N2 列元素,其余按顺序取之,即：

$$M2(1,1)=M1(N1,N2)$$

16. STOSM, M1⁻,M2,N1,N2

本运算存子阵 M2 于矩阵 M1 中,M2 的第一行第一列元素将存于 M1 的第 N1 行第 N2 列上,其余按顺序存入,即：

$$M1(N1,N2)=M2(1,1)$$

矩阵 M1 中存入子阵 M2 的区域,原信息将被冲掉。

17. DUPDG,M1, M2⁺

本运算把矩阵 M1 的对角线元素按顺序组成一个行阵 M2。

18. STODG, M1⁻,M2

本运算把 M2 行阵(或列阵)的各元素依次存储于矩阵 M1 的对角线上。

19. MAX, M1, M2⁺

本运算形成一个列阵 M2，其中每行元素是矩阵 M1 相应行中绝对值最大的元素。

20. NORM, M1, M2⁺, N1

若 N1＝0，形成行阵 M2，其中每列元素是矩阵 M1 中相应列各元素的绝对值之和。

若 N1≠0，形成行阵 M2，其每列元素是矩阵 M1 中相应列各元素的平方和之平方根。

21. INVEL, M1⁻

本运算将矩阵 M1 中各元素的倒数送回 M1 的相应位置中。

22. SQREL, M1⁻

本运算将 M1 中各元素的平方根送回 M1 的相应位置中。

23. LOG, M1⁻

本运算将 M1 中各元素的自然对数送回 M1 的相应位置中。

24. PROD, M1, M2⁺

本运算形成 1×2 的矩阵 M2，它是矩阵 M1 中所有项的连乘积 X，并表示成下列形式：$X = P\cdot 10^{E}$ 存储于 M2 中，其中，M2(1)＝P，M2(2)＝E。

25. DELETE, M1

本运算将删除内存中的数组 M1。

§2－2　一般矩阵运算的 FORTRAN 源程序

以上叙述了一般矩阵运算的表达形式。下面我们讨论如何用 FORTRAN 语言来实现这些运算。首先应说明，CAL 的运算卡片是作为数据输入内存的，在执行这些运算之前，有一系列的子程序来读入这些运算卡及数据卡，一部分子程序留待以后再进行讨论。这里我们先讨论如何用 FORTRAN 来执行这些运算。实际上，这一部分相当于 CAL－78 源程序最末一级的子程序。

例 1　矩阵的相加与相减。

运算卡为：ADD, M1⁻, M2 或 SUB, M1⁻, M2

执行这一运算的末级子程序为 ADSB

```
      SUBROUTINE    ADSB(A,B,NR,NC,S)
      DIMENSION   A(NR,NC),B(NR,NC)
      DO  100    I=1,NR
      DO  100    J=1,NC
100   A(I,J)=A(I,J)+S*B(I,J)
```

```
      RETURN
      END
```

其中,A,B 分别为矩阵名字;NR 为矩阵的行数;NC 为矩阵的列数;S 为判断加减的参数,若 S=1 表示矩阵 A,B 相加,若 S=-1 表示矩阵 A,B 相减。子程序 ADSB 实现了 $A=A\pm B$ 的运算。

例 2 矩阵相乘。

运算卡为:MULT,M1,M2, M3+

执行这一运算的末级子程序为 MULT

```
      SUBROUTINE    MULT(A,B,C,NR,NC,NT)
      DIMENSION    A(NR,NT),B(NT,NC),C(NR,NC)
      DO  200  I=1,NR
      DO  200  J=1,NC
      X=0.0
      DO  100   K=1,NT
100   X = X + A(I,K) * B(K,J )
200   C(I,J) = X
      RETURN
      END
```

本子程序执行了 $C=A\cdot B$ 的运算。

其中,A 为 NR×NF 的矩阵;B 为 NT×NC 的矩阵。注意,A 的列必须与 B 的行相等,否则将引起停机,乘积矩阵 C 是 NR×NC 阶的。

例 3 矩阵转置。

运算卡为:TRAN,M1, M2+

执行这一运算的末级子程序为 TRANS

```
      SUBROUTINE    TRANS(A,B,NR,NC)
      DIMENSION    A(NR,NC),B(NC,NR)
      DO  100   I=1,NR
      DO  100   J=1,NC
100   B(J,I)=A(I,J)
      RETURN
      END
```

本子程序执行了 $B=A^T$ 的运算。

其中,A 是 NR×NC 阶的矩阵;B 则必是 NC×NR 阶的矩阵。

例 4 复制数组。

运算卡为:DUP,M1, M2+

执行这一运算的末级子程序为 DUPL

```
      SUBROUTINE    DUPL(A,B,NR,NC)
      DIMENSION    A(NR,NC),B(NR,NC)
      DO  100   I=1,NR
```