多功能数据处理子程序——GTDT*[1)]

郑云龙　金在律
（大连工学院造船系）

A MULTI-FUNCTION DATA-PROCESSING SUBROUTINE——GTDT

Zheng Yun-long　Jin Zai-lu
(Department of Naval Architecture, Dalian University of Technology)

Abstract

In this paper, a multi-function data-processing subroutine is worked out using the FORTRAN-IV language. It can read free format data and duplicate data, multirecursion and complex statements, etc., and can deal with many sub-models. Therefore the input data are greatly reduced. Riporous checks are made on the input data, and error information is instantly fed back so as to make it easier to detect data errors. This subroutine is concise and versatile, and can be effectively employed to process any data that have regularities. It is especially useful for the input of finite element data.

一、前　言

众所周知，用有限元法进行结构分析时，必须输入大量的原始数据．数据量大、出错率就高，在实际计算中，常因为数据出错而导致运算失败．为解决这个问题，人们已经作过三方面的工作，首先是编制数据检查程序，专门用来检查原始数据．经检查"合格"的数据，便可以进行计算．数据检查程序依赖于有限元分析程序．其次是编制有限元前处理程序[1]，包括自动剖分网格和自动形成载荷等，使输入信息大幅度减少，并减少出错率．但前处理程序大都局限于某一类具体的问题，通用性很差．再其次是编制数据处理程序[2]，这种方法不依赖于分析程序，也不局限于某一类问题，仅从数据出发，利用数据本身的规律性进行加工处理，使输入信息减少，因而出错几率降低，同时也大大减轻数据准备工作．这种程序可以装入数据检查程序或有限元前处理程序．

本文提供--个多功能的数据处理子程序——SUBROUTINE GTDT (IOR, INT, REL) (GTDT 为 GeT DaTa 的缩写)，把它装在计算程序中，可以随意调用，每调用--

* 1987 年 4 月 27 日收到．

1) 本文曾在第二届全国计算力学会议上宣读．

次取出一个数据: IOR (Integer Or Real) 返回数据类型、INT (INTeger) 返回整数值、REL (REaL) 返回实数值. 数据按自由格式书写,与调用方式毫无关系. 若发现数据有错,及时报告出错信息. 数据中可以随意加注注解,增强数据的可读性. 原始数据可由数据文件或键盘输入.

二、程序的设计思想

如图 1 所示的平面有限元网格, 具有 45 个节点, 32 个矩形单元 (或 64 个三角形单元),要求输入节点坐标值和单元的节点号.

由图可见, x 坐标 "0.0, 2.0,…, 18.0" 九个数值各自重复五次, y 坐标 "0.0, 1.5, 3.0, 4.5, 6.0" 五个数值反复九次. 相邻单元的节点号在 x 轴方向相差5,在 y 轴方向相差 1. 如果这些有规律的数据全部由计算机生成,输入数据显然会减少很多. 为此, 在子程序 GTDT 中设计了如下八种功能.

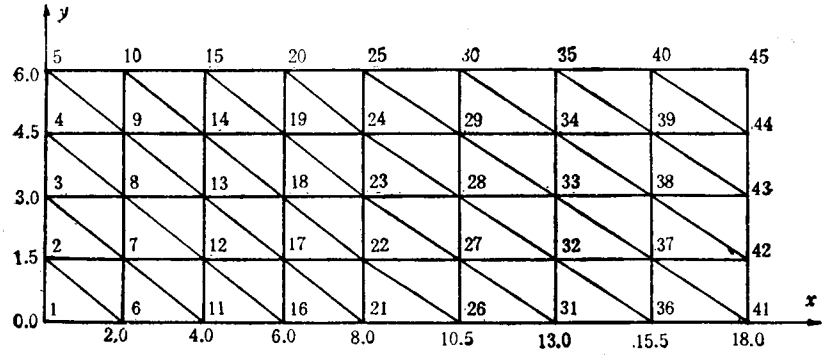

图 1 平面有限元网格

1. 输入自由格式数据或字母

SUBROUTINE GTDT (IOR, INT, REL) 可以读取按自由格式书写的数据或程序中规定的字母,其中哑元 IOR, INT 和 REL 均为返回量, IOR 为数据类型编号、INT 为所取得的整数值、REL 为所取得的实数值. 数据有如下七种类型,其中 ix, iy, iz 为

输入数据	IOR	INT	REL
整数 i	1	i	FLOAT(i)
实数 r	2	r 的整数部分	r
字母 X	3	ix	rx
字母 Y	4	iy	ry
字母 Z	5	iz	rz
字母 A	6	0	0.0
字母 OK	7	0	0.0

对应字母 X,Y,Z 的整数值, rx, ry, rz 为对应字母 X, Y, Z 的实数值,它们的初值均为零. 可再定义(见功能 8),例如,数据段

$$-5.5,X,1070,A,+1.23E04,$$

处理后成为

IOR:	2	3	1	6	2
INT:	−5	0	1070	0	1
REL:	−5.5	0.0	1070.0	0.0	1.23E4

2. 复制一个数据“a, D,”

在一个整数(或实数) a 之后打上一个字母 D，就将 a 的值复制一次，D 可以重复使用，如

$$5,D,D, \Rightarrow 5,5,5,$$
$$5.5,D, \Rightarrow 5.5,5.5.$$

3. 重复数据“$a*i$,”其中 a 是一个整数或实数，i 是正整数，表示数值 a 重复的次数。$a*i$ 处理后成为 i 个 a，即

$$a*i, \Rightarrow a,a,\cdots,a,(i\text{ 个}).$$

4. 多个重复数据“$R,i,a_1,a_2,\cdots,a_n,S$,”其中字母 R(Repeat) 和 S(Stop repeating) 分别表示重复的开始和结束、正整数 i 表示每个数据的重复次数. $a_k(k=1,2,\cdots,n$, n 为任意整数)是被重复的数据，可以是整型、实型或字母 X、Y 或 Z. 这段数据处理后成为

$$R,i,a_1,a_2,\cdots,a_n,S, \Rightarrow a_1,a_1,\cdots,a_1(i\text{个 }a_1),$$
$$a_2,a_2,\cdots,a_2(i\text{ 个 }a_2),$$
$$\cdots,$$
$$a_n,a_n,\cdots,a_n(i\text{个 }a_n).$$

5. 步长型循环“F,i,s,a,”其中字母 F(For) 为循环标志，正整数 i 为循环次数. s 和 a(均为整数或均为实数)分别为步长和初值. 处理后成为

$$F,i,s,a, \Rightarrow a,a+s,a+2s,\cdots,a+(i-1)s,.$$

6. 多重循环“$B,i,s,a_1,a_2,\cdots,a_n,E$,”其中字母 B(Begin) 和 E(End) 分别为循环开始和结束的标志，正整数 i 为循环次数，s(整数或实数)为步长，$a_1,a_2,\cdots,a_n$ (n 任意)为循环体，每一个 $a_k(k=1,2,\cdots,n)$ 可以是 1—8 任一种语句，但必须保证类型匹配，因而循环体是一个复合语句，并且这种循环是可以多层嵌套的(至多套 10 层). 这段数据处理后就将循环体的内容以步长 s 循环 i 次. 假定 $s,a_1,a_2,\cdots,a_n$ 均为数值(整数或实数)，则处理后便成为

$$B,i,s,a_1,a_2,\cdots,a_n,E, \Rightarrow$$
$$a_1,a_2,\cdots,a_n,a_1+s,a_2+s,\cdots,a_n+s,$$
$$\cdots$$
$$a_1+(i-1)s,a_2+(i-1)s,\cdots,a_n+(i-1)s,.$$

7. 子模式的定义与调用

(1) 定义：“$M,i,a_1,a_2,\cdots,a_n,N,i,$”，其中 M(Model) 和 N 分别为子模式开始与结束的标志，正整数 i ($1\leqslant i\leqslant 25$) 为子模式号，$a_1,a_2,\cdots,a_n$ (n 任意)为模体，每一个 $a_k(k=1,2,\cdots,n)$ 可以是 1—8 任意一种语句，即子模式也可以多层嵌套，模体也是一个复合语句. 注意，定义子模式的同时执行模体一次.

(2) 调用: "$C,i,$" 或 "$G,i,$" 其中字母 C(Call) 表示调用(G(Goto) 表示转向)正整数 i 为所调用的(已定义过的)子模式号,执行 "$C,i,$" 的结果是将第 i 子模式的模体再现一次. 例如,数据段

$$M,10,F,5,1,1,B,5,-1.0,5.0,E,OK,N,10,\cdots,C,10,$$

处理之后得如下结果:

$1,2,3,4,5,5.0,4.0,3.0,2.0,1.0,OK,\cdots,$

$1,2,3,4,5,5.0,4.0,3.0,2.0,1.0,OK.$

8. 定义, X,Y,Z

有两种定义方法

(1) DEF, a_x, a_y, a_z,

(2) $X=$, a_x, $Y=$, a_y, $Z=$, a_z,

其中 DEF(DEFine), $X=$, $Y=$, $Z=$ 是关键字, a_x, a_y, 和 a_z 是整数或者实数,分别定义 X, Y 和 Z. 第(1)种定义用于同时定义 X, Y 和 Z, 第(2)种定义用于只定义其中之一个或两个. 必须注意,(1)和(2)都是非执行语句,例如

$$1,X=,5,7,X,12,Y, \Rightarrow 1,7,5,12,0.$$

三、几点说明

1. 数据书写规则

(1) 空格一律删除,如 12␣. 3␣4 与 12.34 等价,但关键字 $X=$, OFF 等必须连写

(2) 分隔符",","与":"等价,大小写字母等价.

(3) 注解由"{","}"括起,其内容可以是不含 OFF 的任意不多于 80 个字符的字母数字串. 注解可以插在任意两个数据之间, OFF 所在行的剩余部分全作为注解处理.

(4) 设 N,D 和 J 均为十进制无符号整数(如 01,345,9980 等),则数据可以写成如下形式:

整数: N, $\pm N$;

实数: $\pm N.$, $\pm .D$, $\pm N.D$; $\pm N_E\pm J$, $\pm N._E\pm J$, $\pm .D_E\pm J$, $\pm N.D_E\pm J$.

(5) 正号可以省略,除 E 之外,其它的合法字母两侧的分隔符皆可省略, E 之后, $\pm$ 号之前和注解之前的分隔符亦可省略,且有

$$.,\Rightarrow,0, \qquad .,.\Rightarrow,0.0,$$

$$,*i, \Rightarrow ,0*i, \qquad ,.*i, \Rightarrow ,0.0*i,.$$

(6) 每段数据必须以 OFF 结束,至多 1840 个字符(一个屏幕 80 列 × 23 行),数据太多时可分成若干段输入,但同一个语句不可分隔在两个数据段之中. X, Y, Z 的定义值只在同一数据段中有效,进下段数据时恢复初值.

2. 多重循环的嵌套规则与 ALGOL 语言的多重循环完全相同. 图 2(a) 是合法的,而图 2(b) 是非法的,执行时会误取为图 2(c) 之形式. 但是,子模式的相互嵌套没有限制,只要不引起直接或间接的递归调用,例如图 2(d) 是合法的.

3. 在一定条件下,各功能之间具有等价关系(用 $\Longleftrightarrow$ 表示等价),例如

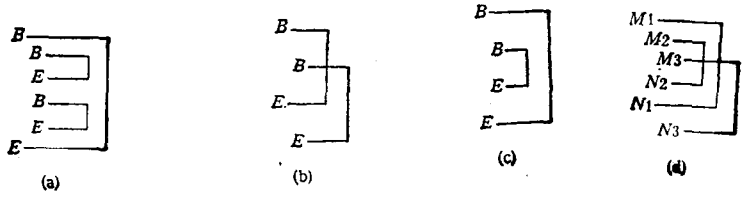

图 2

$a,D,\Longleftrightarrow a*Z,$

$a*i,\Longleftrightarrow F,i,0,a,$ (a 为整型),

或者

$a*i,\Longleftrightarrow F,i,0.0,a,$($a$ 为实型),

$F,i,s,a,\Longleftrightarrow B,i,s,a,E,$

$a_1*i,a_2*i,\cdots,a_n*i,\Longleftrightarrow R,i,a_1,a_2,\cdots,a_n,S,$

$DEF,a_1,a_2,a_3;\Longleftrightarrow X=,a_1;Y=,a_2;Z=,a_3.$

4. 步长型循环与多重循环，必须保证类型的一致性．当 $R,\cdots,S$ 及子模式出现在循环体中时，要保证类型相匹配．字母 X,Y,Z,A 和 OK 可以出现在任意一层的循环体中，因为它们既可代表整型，又可代表实型，完全由使用者规定．

5. 一个数据段输入完毕，GTDT 的全部工作单元可以收回另作别用，用完之后可以继续用于数据输入．

四、例　题

例 1. 输入图 1 的节点坐标 $X(45),Y(45)$，矩形元的节点号 $LR4(4,32)$，和三角元的节点号 $LR3(3,64)$．数据如下：

$\{X\}R,5,0.,2.,4.,6.,8.,10.5,13.0,15.5,18.0,S,$

$\{Y\}B,9,0.,F,5,1.5,.0,E,$

$\{LR4\}B8,5,B4,1,1,6,7,2,E,E,$

$\{LR3\}B8,5B4,1,1,6,2D6,7,EE.$

可见数据量仅仅是实际数据的 11.2%(46:. 410)．

例 2. 图 3 是船体左舷外板的简化模型，有 5 个舱、舱长 $L=17.4M$，半宽 $B=16M$，型深 $H=16.8M$．共划分 396 个矩型单元，410 个节点，输入节点坐标和单元节点号．

总共需要 $3\times410+4\times396=2814$ 个数据，用 GTDT 输入时只用大约150 个数据，减少了 18 倍．其数据文件如下：

$\{NPOINT=410\}$ *def* $\{l=\}348$ $\{B\overline{=}\}1600$ $\{H=\}$ 1680; $\{x=\}b5$, 1740 {*for each tank*}:

{#1-first trans. bulkhead}*25;

{#2-four rings} $b4,348,348*13,e;$

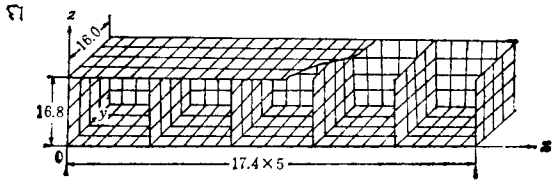

图3 空间有限元网格

```
    e {#3-last trans. bulkhead} 8700*25;
{y=}b5,00,
      {#1} m1,b5,0,b5,400,,ee,n1;
      {#2} b4,,m2f5,400,0n2 ydd g2,e;
    e {#3} c1:
{z=}b5,,
      {#1} m11 r5,,420,840,1260 zsn11;
      {#2} b4,,*4 f5,420,,zddd e;
    e {#3} c11:
off all the coordinates entered!
{link 7---type7 elements NELEM = 396}
B,5,77 {each tank}
{trans. bulkhead} b4,5,b4,1{1}1,2,7,6,ee:
{shell after the bulk head} b4,1{17}26,1,2,27,e
    {21} 30,5,10,31d10,15,32d15,20,33d20,25,38,
    b4,1{25}34,21,22,35,e:
{shell on rings} b3,13b7,1{29}39,26,27,40,e46,33,38,51,b4,1,{37}47,34,35,
48,ee:
{shell before the next bulkhead}b4,1{65}78,65,66,79,e,
    {69}82,69,70,87d70,71,92d71,72,97d72,77,102,
    b4,1,{73}98,73,74,99,e:
E {last bulkhead}b4,5b4,1{321}386,387,392,391,ee:
off element linkage entered!
```

利用 GTDT 输入数据，不仅数据量大幅度减少，而且数据清晰易读、便于修改。

参 考 文 献

[1] 王玉昌，结构应力分析程序的前处理程序——平面网格的自动剖分，数值计算与计算机应用，3(1983)。
[2] 刘元芳，邓可顺，裘春航，钟万勰，自由格式数据输入语句 SCAIN，计算结构力学及其应用，3(1984)。