THE UNCLE FINITE ELEMENT SCHEME

J.A. ENDERBY

United Kingdom Atomic Energy Authority,
Reactor Group, Risley, Warrington WA3 GAT, United Kingdom

SUMMARY

This paper describes, in outline, a completely general finite element scheme, implemented in the UKAEA Reactor Group. UNCLE is not a complete, self-contained program. It is a framework of routines that provide the common services required by all general purpose finite element programs, whether for heat transfer, stress analysis or any other linear (or piece-wise linear) problem. These services are:

(a) input of mesh, geometry, loads (etc) and material data;
(b) matrix and load vector calculation and assembly (including handling of standard boundary conditions);
(c) solution of global matrix (elimination and conjugate gradient methods);
(d) output (printed and graphical) of initial geometry, displacements, stresses, final geometry etc.;
(e) facilities for iteration for non-linear problems and time integration;
(f) mass matrix reduction, dynamic analysis of reduced problem and expansion of displacements to full problem.

The framework is written to handle 1, 2, 3 or more dimensions equally efficiently.

To produce a general purpose program for a particular range of applications it is only necessary to provide a set of element subroutines specialised to the application (heat transfer, framework analysis, continuum stress analysis etc.).

The subroutines which may be required are:

(a) to calculate an element matrix and right-hand sides;
(b) to calculate an element inertia matrix for dynamic calculations;
(c) to calculate element stresses and strains or any other derived quantities for output or to continue the calculation;
(d) to plot an element or element property.

The calculation of element right-hand sides is used to provide such facilities as thermal loads, self-weight, distributed loads, creep effects. UNCLE itself provides boundary conditions such as fixed or relative displacements at any node, but the variety of boundary conditions can be extended indefinitely by the invention of suitable "elements" (eg for radiation, special types of restraint).

The material input can be specialised to the particular application by the construction of a BLOCK DATA subroutine containing tables of material and property names and default values.

The data of the finite element mesh is given as an hierarchical structure which greatly simplifies both the presentation of data and the presentation of results for large or complex problems. ELEMENTS may be joined together to form CELLS. CELLS may be connected together in topologically Cartesian ARRAYS of 1, 2, 3 or more dimensions. In turn, ARRAYS may be connected together to form a structure, every node of which is indexed in a simple manner via this hierarchy. When combined with the UNCLE input routines that give general repetition and arithmetic facilities within lists of data items this gives rise to a completely general mesh generation facility.

Nodes are automatically ordered relative to the hierarchical structure specification and this permits optimal orderings to be achieved with very little effort on the part of the user. The global matrix is assembled prior to solution by a specially developed sort routine. The solution imposes no limits on the band width of the matrix and it is efficient, not only for narrow banded matrices, but also for matrices which are mainly narrow banded with local broadening of the band up to full width.

1. Introduction

1.1 Purpose

The objective of the UNCLE finite element scheme has been to provide a framework of common routines that supply the facilities that can be specified as common to all finite element problems. Since these facilities cover all aspects of input, matrix assembly, equation solving, restarts, iterative and dynamic calculations, graph plotting and other output, it means that a complete general purpose code covering a particular range of applications can be constructed by providing a small number of subroutines specific to each type of element required by the application.

By this means, full advantage may be taken of the virtue of the finite element method in unifying the treatment of diverse problems, such as stress-analysis, heat transfer, neutronics etc. The common standards imposed by the use of common facilities mean that there are no problems associated with the transfer of data from one calculation to another, so that, for instance it is straightforward to run a heat transfer calculation and a stress-analysis calculation in sequence.

The status of UNCLE is that all the features described in this paper up to the end of section 5 are operational and have been working in several UNCLE - based programs for some considerable time. The facilities described in section 6 are being developed as required. The dynamic facilities of section 7 are currently being written.

1.2 Element Generality

The definition of "finite element" adopted is consistent with this very general aim. A finite element is considered to be any part of a "structure" whose contribution to the structure can be completely described by a square matrix relating the "effect" to the "cause" for each "freedom" of the element and/or a vector giving the contribution ("loading") that each freedom of the element makes to the right-hand side of the "cause-effect" equations.

Cause and effect may be force and displacement, displacement and reaction, heat-source and temperature etc. Special boundary conditions may readily be included by the definition of suitable elements; special loading conditions may be introduced by elements which contribute only to the right-hand side of the element equations.

No restrictions are imposed on the number of nodes of an element or the number of freedoms at each node. Problems of 1, 2, 3 or even more dimensions are included in the same framework.

1.3 Table Handling

Because of the generality of the system different problems may give rise to tables of data, which not only differ greatly in size, but also differ greatly in the distribution of data between tables. To cope with this, without imposing arbitrary limitations on the use of various facilities, there is a completely dynamic system for handling tables within core and between core and backing store. Tables may automatically be moved about within core to allow another table to be extended. Tables, or portions of tables, not immediately required may be automatically removed from core to backing store if space is needed for other tables.

Thus there are no explicit size limits in any part of the problem specification. There are, of course, practical limitations on overall problem size, depending on the amount of core storage specified for a run; these limitations are determined by the size of tables, such as the table of nodal coordinates which may have to be accessed randomly at some stage of the calculation.

2. Data Structure

2.1 The Data Hierarchy

An important feature of UNCLE is an hierarchical data structure, which not only greatly simplifies the initial provision of data for problems, however large and complex, but also gives the user flexible control over the course of the calculation at all stages. The basic unit of structure is the ELEMENT. ELEMENTS may be joined together fo form CELLS. CELLS may be connected together in topologically regular ARRAYS of 1, 2, 3 or more dimensions. ARRAYS may then be connected together, in an arbitrary fashion to form a STRUCTURE.

When there is no regularity in the structure, an array may comprise a single large and complex cell with nodes numbered arbitrarily by the user and elements specified topologically by groups of node numbers. However, more usually, it is possible to subdivide parts of the structure into cells which lie on a mesh, which is often geometrically regular, but if not that, is at least topologically regular. Such a regular array gives a much more convenient way of indexing nodes than direct numbering. This is extremely important in more than 2 dimensions, when one loses the ability to number nodes on a scale drawing of the structure.

For instance, a cell may be referenced as:

10 in 1-dimension

2 & 7 in 2-dimensions

3 & 2 & 15 in 3-dimensions

A node of cell 2 & 7 can be referenced as 2 & 7 ∅ 3.

Cell interconnections can be defined as an implicit property of a cell, by giving node positions on a notional cell grid such that any node on a grid point which is common to two adjacent cells represents a connection between the cells. Thus, to define a 3-dimensional array of 20 by 15 by 30 cells as

#4, C1, 20 & 15 & 30

automatically establishes the topological connections between all 9000 type 1 cells in array 4.

Such arrays may be simply wrapped around to form a cylinder or toroid as:

#17, C1, 20 & CL10

#12, C2, CL4 & 5 & CL20

where CL indicates closure.

To refine the mesh in the neighbourhood of a point of rapid change, an array of finer mesh may be inserted within another array; a still finer may be inserted in that and so on.

Elements may be defined as SHARED so that, if they are common to two or more cells they are included only once, but if cells are deleted, the shared elements are only deleted if all the cells to which they belong are deleted.

It is not necessary for all nodes of an element to be defined explicitly. If the position of a node can be unambiguously determined by its position relative to the given nodes, it can be treated as a SUBNODE. No reference need be made to a subnode in the problem specification, though its displacements may be calculated and plotted in the calculation.

2.2 Geometry and Other Nodal Properties

The geometry of the system is specified separately from the topology and all other nodal properties of the structure, such as temperature, are specified in a similar manner.

Like the topological cell coordinates, geometrical coordinates are written as 25.4 in 1-dimension, 12.2 & 13 in 2-dimensions, 3.5 & 48 & 46.6 in 3-dimensions etc. The geometrical coordinates of a cell reference point are given by

12 & 6 = 14.2 & 36.1 & 48.9

for a 2-dimensional array in 3-dimensional space. If the cell reference point (ie the first corner point) of each cell is given, the coordinates of all the nodes in each cell are obtained automatically by interpolation.

However, the coordinates of individual nodes of a cell may be given directly; for instance

12 & 6 $ 2 = 14.2 & 38 & 48.9

for the second node of the above cell. If the coordinates of a cell are given or implied more than once in the problem specification it is the last value given or implied that is used in the calculation.

2.3 Concise Data Input

All UNCLE data is read through common subroutines, which handle the associations of coordinates with nodes or cell types and cell properties with cells and so on. These routines, in turn, read their data through a common input routine called LISTIN [1] which reads lists of numbers, words or special characters. Such lists may be written extremely concisely because of the very general notations provided by LISTIN for repetitions, with systematic or partly systematic variations, combined with the ability of LISTIN to accept arithmetical expressions.

4 (3) ≡ 3, 3, 3, 3

3 (7, 10) ≡ 7, 10, 7, 10, 7, 10

3 (1:1, 10:-1) ≡ 1, 10, 2, 9, 3, 8

(4:2:1), 21 ≡ 4, 6, 9, 13, 18, 21

(7, "9, 3, 7") ≡ 7, 9, 7, 3, 7, 7

Multiple repetitions are allowed

2 (4 (3), 1) ≡ 3, 3, 3, 3, 1, 3, 3, 3, 3, 1

3 (4:-1 (1:1), 2) ≡ 1, 2, 3, 4, 2, 1, 2, 3, 2, 1, 2, 2

3 (4 (<1:1> & 5:1)) ≡ 4 (1 & 5:1), 4 (2 & 5:1), 4 (3 & 5:1)

The same notations can be used with the topology, geometry, loadings or any part of the problem specification. For instance the coordinates of all nodes on a hollow cylinder parallel to the z-direction (Fig 1) can be written

CELL = 6 (13 (CS (0:30) & SN (0:30) & <0:0.5237>))

where CS and SN are sin and cos with arguments in degrees. Fig 2 shows a "wire-model" drawing of a thick pipe entering a thick spherical cylinder, produced by the FAUN [2] program. The input required to locate all the nodes is given in Table 1. This shows the use of variables (?1, ?2 etc) in the LISTIN notation; by altering certain of these variables, all the parameters of this model can be changed, including pipe-angle and length and the mesh sizes.

2.4 Mesh Generation

In fact the UNCLE data structure, taken together with the LISTIN data input condensation provides a completely general mesh generation facility. Some thought may be required in devising a suitable procedure for setting up the mesh in an entirely new situation, but a technique, once devised, can be used again in analogous situations with very little effort.

The meshes devised in this way have a regularity and uniformity superior to hand drawn meshes. The advantage over other automatic mesh generation systems is in its great generality and in the control that the user has over the mesh generated. It is easy to vary the fineness of the mesh for areas of potentially high stress. Further, however complicated the problem, the user still has complete control over the indexing of elements, so that there is no difficult in specifying by which nodes parts of the structure are connected; to which nodes or elements loads are to be applied or which nodes or freedoms are subject to boundary conditions or restraints. In fact the same facilities may be used for the generation of connections, loads, temperature, boundary values etc.

2.5 Material Data

UNCLE provides general facilities for handling material data for the elements. The amount of data for each element type may be different and may include both necessary and optional items. Data may also be made a function of position in space. The data specifications for any UNCLE program are written in a BLOCK DATA subroutine which contains all the keywords, default values and dimensions required for the input, tabulated listing and calculational use of the data.

The data is referenced by a pair of material numbers associated with each cell in an array and each element in a cell. This provides simple defaults for all elements in a cell to have the same properties or for all cells to have the same set of element properties, but is sufficiently general to allow individual elements to be given unique properties.

3. Imposed Conditions

3.1 Restraints

Some boundary conditions are provided as standard by the UNCLE framework. These are the ability to prescribe any displacement (temperature etc according to the nature of the problem) at any node to be zero or a given value. The displacement may be given relative to any set of orthogonal axes located at the node. These conditions are required to make the problem non-singular, but also provide for symmetry conditions along any boundary. A displacement may also be fixed relative to a displacement of any other node. This provides for rotational symmetry, periodic boundary conditions, the treatment of slides and hinges and can even be used for the combination of grid and plane stress problems.

3.2 General Boundary Conditions

Any other types of boundary condition are readily catered for by the generality of the UNCLE element. Thus restraints can be introduced by elements which relate reaction to displacement, rather than displacement to force. In heat transfer it is simple to define elements for radiation across gaps or radiation between distant surfaces.

3.3 External Loadings

External loadings may take the form of point loads or element loads. Point loads can be given for any freedom of a node, relative to any set of orthogonal axes at the node.

Element loads are prescribed for an element defined by a set of nodes. The data specifications for each type of element load are set up in a BLOCK DATA subroutine like the element material data so very great generality is possible. Further the element loads are calculated from the data at the same time as the element matrix is calculated, so the subroutine for calculating element loads has access to all information about the element, including the material data. Such element loads may be point loads away from nodes, distributed loads, lack-of-fit loads etc.

3.4 Internal Loadings

Internal loadings are those determined by internal properties and overall conditions which do not have to be provided as data for specific elements. These are calculated by the main element subroutines using material data and nodal properties, such as temperature, etc. Such internal loadings are thermal loads, self-weights,creep-loads etc.

4. Outline of UNCLE Methods

4.1 Ordering of Nodes

The user always refers to nodes by the cell and node indexing. However, internally, the nodes are referred to by node numbers generated by UNCLE when the nodes are ordered. These only correspond with the user numbering for a single single-cell array. The node ordering is nevertheless, determined by the user, in that nodes are numbered array by array in the order the arrays are input (except for inserted arrays which are numbered within the array in which they are inserted). Within arrays, nodes are numbered from the beginning of the array, strictly parallel to the first side or end of the array. This is possible because of the notional cell grid; the grid points can be examined along parallel lines and nodes are numbered as they occur on grid points. This method achieves systematic orderings for arrays quite irrespective of deleted cells or areas where a finer mesh is used.

Providing the array is input with its sides in increasing order of number of nodes per side, the ordering obtained for a single array is normally the very best that can be obtained. As pointed out later, for the main UNCLE solution algorithm the best node orderings of the whole structure will be those where each array is ordered optimally, irrespective of the way in which the arrays are joined together.

4.2 Element Ordering

Elements are ordered exactly as they occur, in going systematically through each cell of each array in the order input. In the ordering pass each element is expressed in terms of the internal node numbers; repeated shared elements are eliminated and each subnode found is associated with a primary node of the first element in which it occurs. At the same time a relationship is determined between node numbers and freedom numbers in the global matrix.

4.3 Construction of the Global Matrix and Right-Hand Sides

A pass is made through the elements in the order just described. For each element the appropriate element subroutine is entered with the appropriate nodal coordinates and material data provided and the subroutine returns the element matrix and/or right-hand sides. These are saved in their original form for future use. The standard UNCLE restraint conditions are applied at this time and then any remaining non-zero coefficients are output to disc in blocks, tagged with global row and column numbers to form the global matrix and global right-hand sides.

The element subroutine may make use of element data, such as forces or displacements which have been fed-back from previous steps of a calculation. It may also hand over to UNCLE, for storage and subsequent feed-back, any other information which may be needed by a subsequent step.

4.4 Ordering the Global Matrix

The global matrix itself is formed by a disc sort into column order. In this process, any coefficients with the same row and column number are added together rather than ordered. The sort algorithm was devised to ensure that any local disorder could be removed in a single pass; that already correctly ordered blocks could be skipped over on subsequent passes; that coefficients badly out of order are simply collected together to be treated as a block on the next pass, rather than moved gradually up through the bulk of coefficients.

Alternate passes are upwards through blocks ordered by ascending lowest row and column number and downwards through blocks ordered by descending highest row and column number. Table 2 shows the way in which, in each pass, the number of coefficients reduces as they are added together; the number of blocks involved in the sort reduces as blocks at the top and bottom become correctly sorted; the total number of passes is itself appreciably fewer than the theoretical number required for a 2-way sort/merge. No disc space is used other than that occupied by the original unsorted coefficients.

4.5 Solving the Global Equations

Since the global matrix is ordered, it is possible, very simply, to provide a choice of solution algorithms. The main algorithm is a sparse elimination method, but a conjugate gradient algorithm is also being provided, because it is known to be particularly effective for compact 3-dimensional solid structures. For large structures of that kind elimination becomes completely uneconomic, because of the large and irreducible bandwidth.

The elimination method used, makes no prior assumptions about the bandwidth. Each column of the matrix is brought into core in turn and all operations involving that column are carried out while it is still there. Where the bandwidth is sufficiently narrow (depending on the working space allowed) the numbers required for the calculation stay in core until they are no longer needed and the algorithm is as efficient as any narrow bandwidth algorithm. If the bandwidth broadens locally, numbers have to be transferred to disc and then recovered for a subsequent column and the process is slowed. However, as soon as the bandwidth becomes narrow again the numbers required are once again kept in core as long as they are needed. Thus, if the arrays of the structure each have narrow bandwidths, the broadening of the band where they are joined together is not a serious impediment to a rapid solution - the ordering which it is simplest to achieve is likely to be the best.

5. Results

5.1 Printed Output

Standard facilities in UNCLE list (see table 3) the cells, showing cell type, material types and node numbers; the elements, showing element type, material index and node numbers; the material data; nodal coordinates and other properties; loads and restraints.

After the solution, displacements, forces and reactions at restraints can be listed by standard facilities. However, it is also possible to provide element subroutines to output derived quantities such as element stresses and strains, suitable to the elements included in a particular UNCLE-based program.

5.2 Graphical Output

The graphical output is very largely an UNCLE provided facility. It is possible to plot the whole structure, or individual arrays or cells or to plot the skeleton of the structure, without cell detail. The plots may be before or after deformation. Any part of the structure may be viewed from any angle. Node numbers may be plotted or omitted.

For each type of element it is necessary to provide an appropriate plotting subroutine. However, the only requirement of such a routine is for it to determine which nodes of the element have to be joined together and whether by a straight line or a curve; everything else is determined by UNCLE.

The plotting facilities are under continual development.

5.3 Feedback

The standard UNCLE output routines may also produce displacements and forces, ordered by element (rather than by node) to be passed on to subsequent steps of a calculation.

6. Non-Linear and Time-Dependent Calculations

Non-linear problems, such as radiation, geometrical stiffness and plastic deformation and time-dependent problems, such as temperature transients and creep have the common feature of requiring iterative and/or recursive solution of the static system. The global matrix may sometimes remain constant, with only the right-hand side changing for a number of successive solutions. UNCLE will provide uniform facilities for control of these repetitive calculations and to handle the feedback of the appropriate data from one step to another. When the matrix is unchanged the solution for successive right-hand sides may be obtained rapidly using the factored form of the global matrix produced by the elimination algorithm. The specialisation of these recursive facilities to particular problems will lie in writing special element subroutines which calculate the new element matrix or right-hand sides from basic data and the old element matrix, right-hand sides and any other data fed back from previous steps.

7. Dynamic Analysis

As well as an element subroutine for the stiffness matrix, an element subroutine may be provided to calculate an inertia matrix. Then, when DYNAMIC is specified, UNCLE will also assemble a global inertia matrix. Following a standard static calculation to obtain a factorised global stiffness matrix, UNCLE may be requested to form a condensed stiffness and inertia matrix for a reduced number of nodes. Some or all of the master nodes may be specified. Those not specified will be chosen by UNCLE.

The eigen-value analysis will be carried out by a standard eigen-value package incorporated in UNCLE. Standard UNCLE facilities will then be used for calculating, outputting and plotting actual displacements, stresses and strains.

8. Conclusions

The principle adopted in developing UNCLE and UNCLE-based programs was that each problem that arose should be solved in as general manner as possible, so that any new facilities introduced should be useful for any finite element calculation. This has been very successful, so that coding specific to particular elements is confined to a small number of well-defined areas.

The principle of generalisation has resulted in input facilities which can be used for generation of meshes in the most complex circumstances.

References

[1] COLLIER, W. D., ENDERBY, J. A., "LISTIN: a Subroutine for the Concise Input of Data," TRG Report 2310 (R), 1973.

[2] ENDERBY, J. A., KNOWLES, J. A., "FAUN: a General Program for the Elastic Analysis of Frames, Pipeworks and Shells," Paper F6/4 to be presented at 3rd SMIRT conference, London, 1975.

TABLE 1 GEOMETRY INPUT FOR FIG. 2

```
GEOMETRY
?1=55; ?2=CS(30); ?3=SN(30);
A1
CELL = 10(?5=-18:4; 10(?4=-18:4-?1*?3; ?6=SQRT(?1*?1-?4*?4-?5*?5);
  2(?7=1-1:-2/?1; ?8=?4*?7+?1*?3; ?9=?5*?7; ?10=?6*?7-?1*?2;
?8&?9&?10)))
A2
CELL = 21(?4=9:18; ?5="10,10,5(10:-4),5(-10),5(-10:4),4(10)";
      ?6="2,6,5(10),5(10:-4),5(-10),4(-10:4)";
  6(2(?7=5:1*CS(?4); ?8=5:1*SN(?4); ?9=1-1:-2/?1; ?10=?9*?5+1:-2*?3;
  ?11=?9*?6; ?12=?7*(1-<0:1>/5)+<0:1>*?10/5;
  ?13=?8*(1-<0:1>/5)+<0:1>*?11/5; ?14=?12-?1*?3; ?15=?1-1:-2;
  ?16=SQRT(?15*?15-?14*?14-?13*?13); ?17=?16-?1*?2;
?12&?13&?17)))
A3
CELL = 21(?4=9:18; 5(2(?5=6:-1*CS(?4); ?6=6:-1*SN(?4);
  ?7=?5-?1*?3; ?8=?1+1:-2; ?9=SQRT(?8*?8-?7*?7-?6*?6); ?10=?9-?1*?2;
  ?11=?10*(1-<0:1>/4)+<0:1>*20/4;
?5&?6&?11)))
```

TABLE 2
SORT OF GLOBAL MATRIX

UNSORTED BLOCKS
32128 WORDS

BLOCK	LENGTH	MINC	MINR	MAXC	MAXR
1	816	1	1	89	89
8	816	1	1	144	144
2	816	7	7	96	96
3	816	13	13	108	107
4	816	25	25	114	114
5	816	31	31	124	125
6	816	43	43	132	132
7	816	49	49	142	143
9	816	73	73	300	300
10	816	86	88	312	312
11	816	97	97	318	318
12	816	107	106	330	330
13	816	115	115	330	336
14	816	127	127	348	348
15	816	133	133	355	359
17	816	145	145	366	366
16	816	145	145	361	369
24	816	145	145	222	222
18	816	151	151	312	312
25	816	154	226	234	234
19	816	163	163	324	324
26	816	163	163	24[illegible]	246
20	816	169	169	331	335
27	816	173	244	252	252
21	816	181	181	342	342
28	816	181	131	259	191
22	816	187	187	35[illegible]	353
29	816	193	193	270	270
30	816	199	199	275	217
23	816	199	199	30[illegible]	30[illegible]
31	816	211	211	286	288
32	816	217	217	293	224
33	816	229	229	306	336
34	816	235	235	317	242
35	816	247	247	324	324
36	816	253	253	330	336
37	816	265	265	342	342
38	816	271	271	354	354
39	816	283	283	360	360
40	304	289	289	360	360

AFTER FIRST PASS
22072 WORDS

BLOCK	LENGTH	MINC	MINR	MAXC	MAXR
24	816	345	129	360	360
40	816	349	338	360	360
39	816	332	336	360	337
16	816	329	321	344	354
21	816	31[illegible]	166	342	342
38	816	316	316	332	335
17	816	313	307	329	113
37	816	31[illegible]	295	310	315
27	816	270	300	31[illegible]	164
36	816	287	219	3[illegible]	294
35	816	274	272	257	215
34	816	26[illegible]	334	274	269
33	508	251	322	20[illegible]	332
23	816	239	233	251	320
30	816	226	151	239	232
29	816	212	150	225	297
22	372	206	210	212	146
20	816	192	264	2[illegible]	219
26	816	179	171	192	198
19	164	176	131	179	323
18	816	162	306	170	180
15	816	150	150	162	234
14	248	145	211	15[illegible]	146
12	816	131	129	145	156
11	816	116	122	131	127
10	116	114	330	110	121
7	816	10[illegible]	93	114	120
6	816	85	84	1[illegible]	29
5	816	65	65	65	86
4	816	17	17	65	64
3	264	1	1	17	16

SORTED BLOCKS
19136 WORDS

BLOCK	LENGTH	MINC	MINR	MAXC	MAXR
3	264	1	1	17	16
4	816	17	17	65	64
5	816	65	65	65	86
6	816	85	84	1[illegible]	29
7	816	10[illegible]	93	114	120
10	116	114	330	11[illegible]	121
11	816	116	122	131	127
12	816	131	129	145	156
14	248	145	211	15[illegible]	146
15	816	150	150	162	234
18	816	162	306	170	18[illegible]
19	164	176	171	179	323
26	816	179	171	192	198
20	816	192	264	[illegible]	219
22	372	206	210	212	146
29	816	217	150	225	297
30	816	226	151	239	232
23	816	239	233	251	320
33	508	251	322	[illegible]	332
34	816	26[illegible]	[illegible]	274	269
35	816	27[illegible]	272	[illegible]	[illegible]
36	816	289	219	[illegible]	225
27	816	29[illegible]	224	[illegible]	[illegible]5
37	[illegible]	31[illegible]	163	327	[illegible]7
17	[illegible]	322	175	[illegible]	[illegible]
38	816	334	[illegible]7	345	[illegible]
21	816	345	[illegible]	[illegible]	[illegible]
16	328	35[illegible]	[illegible]7	[illegible]	[illegible]

BLOCKS ALTERED IN SECOND PASS (blocks 36 to 16)

TABLE 3 EXAMPLE OF UNCLE OUTPUT

```
¬¬ONE SEGMENT OF 60 DEG PITCH CYLINDER
DIM 3 FREE 6
ELEMENTS
#1=SBEAM
CELLS
#1,4,2&2; EL 1,3_2,4=E1=M1;  1,2_3,4=M2
ARRAYS
#1,1&5,C1
DATA
MATERIAL
BEAM;  #1 A .05237 = AY 523.7 = IY 1.7455E-4 = IZ 1.197E-3 =
            J 3.491E-4 = E 3E7 = G 1.5E7 = THIRD 0&0&0
       #2 AY 523.7 = IZ 1.197E-3 = ROTATE -90 = R 1
TM
#1; 1,2&1,2_4,5&4,5=2(CS(60),-SN(60),SN(60),CS(60))
GEOMETRY
All CELL=6(2(CS(0:60)&SN(0:60)&<0:0.5237>))
RESTRAINTS
FIXED
All 1&1$1,2%1,2,3=ZERO
RELATIVE
All 1&5(1:1)$3%6(1:1)=1&5(1:1)$4%6(1:1)=ZERO
          1&1$1%4,5,6=1&1$2%4,5,6=ZERO
TRANSF
All T1=1&5(1:1)$2,4
LOADS
All  1&5$3%2=6059
END
```

```
                    DATA LISTING               TABLE 3  CONTINUED                 20:02:42 25MAR75      PAGE 1

BEAM
 M        A            IY           J            E            G            EXP          GRAV         AY        IZ           R

 1    5.2370E-02   1.7455E-04   3.4910E-04   3.0000E 07   1.5000E 07   0.0000E 00   0.0000E 00   523.7    1.1970E-03
                                                                                    0.0000E 00
 2    5.2370E-02   1.7455E-04   3.4910E-04   3.0000E 07   1.5000E 07   0.0000E 00   0.0000E 00   523.7    1.1970E-03   1.000
                                                                                    0.0000E 00

BEAM
 M          THIRD        ROTATE

 1      0.0000E 00
        0.0000E 00
        0.0000E 00
 2                      -90.00
```

```
NODE 1                  RESTRAINTS                 TABLE 3  CONTINUED                 20:02:42 25MAR75      PAGE 2

  NODE          NATURE OF RESTRAINT

   1         FREEDOM  1  IS FIXED AT ZERO
             FREEDOM  2  IS FIXED AT ZERO
             FREEDOM  3  IS FIXED AT ZERO
             FREEDOM  4  IS FIXED AT ZERO       RELATIVE TO FREEDOM  4  OF NODE    2
             FREEDOM  5  IS FIXED AT ZERO       RELATIVE TO FREEDOM  5  OF NODE    2
             FREEDOM  6  IS FIXED AT ZERO       RELATIVE TO FREEDOM  6  OF NODE    2
   2         TRANSFORMATION NUMBER     1
             FREEDOM  1  IS FIXED AT ZERO
             FREEDOM  2  IS FIXED AT ZERO
             FREEDOM  3  IS FIXED AT ZERO
             FREEDOM  4  IS FIXED AT ZERO       RELATIVE TO FREEDOM  4  OF NODE    1
             FREEDOM  5  IS FIXED AT ZERO       RELATIVE TO FREEDOM  5  OF NODE    1
             FREEDOM  6  IS FIXED AT ZERO       RELATIVE TO FREEDOM  6  OF NODE    1
   4         TRANSFORMATION NUMBER     1
             FREEDOM  1  IS FIXED AT ZERO       RELATIVE TO FREEDOM  1  OF NODE    3
             FREEDOM  2  IS FIXED AT ZERO       RELATIVE TO FREEDOM  2  OF NODE    3
             FREEDOM  3  IS FIXED AT ZERO       RELATIVE TO FREEDOM  3  OF NODE    3
             FREEDOM  4  IS FIXED AT ZERO       RELATIVE TO FREEDOM  4  OF NODE    3
             FREEDOM  5  IS FIXED AT ZERO       RELATIVE TO FREEDOM  5  OF NODE    3
             FREEDOM  6  IS FIXED AT ZERO       RELATIVE TO FREEDOM  6  OF NODE    3
   3         FREEDOM  1  IS FIXED AT ZERO       RELATIVE TO FREEDOM  1  OF NODE    4
             FREEDOM  2  IS FIXED AT ZERO       RELATIVE TO FREEDOM  2  OF NODE    4
             FREEDOM  3  IS FIXED AT ZERO       RELATIVE TO FREEDOM  3  OF NODE    4
             FREEDOM  4  IS FIXED AT ZERO       RELATIVE TO FREEDOM  4  OF NODE    4
             FREEDOM  5  IS FIXED AT ZERO       RELATIVE TO FREEDOM  5  OF NODE    4
             FREEDOM  6  IS FIXED AT ZERO       RELATIVE TO FREEDOM  6  OF NODE    4
   6         TRANSFORMATION NUMBER     1
             FREEDOM  1  IS FIXED AT ZERO       RELATIVE TO FREEDOM  1  OF NODE    5
             FREEDOM  2  IS FIXED AT ZERO       RELATIVE TO FREEDOM  2  OF NODE    5
             FREEDOM  3  IS FIXED AT ZERO       RELATIVE TO FREEDOM  3  OF NODE    5
             FREEDOM  4  IS FIXED AT ZERO       RELATIVE TO FREEDOM  4  OF NODE    5
             FREEDOM  5  IS FIXED AT ZERO       RELATIVE TO FREEDOM  5  OF NODE    5
             FREEDOM  6  IS FIXED AT ZERO       RELATIVE TO FREEDOM  6  OF NODE    5
   5         FREEDOM  1  IS FIXED AT ZERO       RELATIVE TO FREEDOM  1  OF NODE    6
             FREEDOM  2  IS FIXED AT ZERO       RELATIVE TO FREEDOM  2  OF NODE    6
             FREEDOM  3  IS FIXED AT ZERO       RELATIVE TO FREEDOM  3  OF NODE    6
             FREEDOM  4  IS FIXED AT ZERO       RELATIVE TO FREEDOM  4  OF NODE    6
             FREEDOM  5  IS FIXED AT ZERO       RELATIVE TO FREEDOM  5  OF NODE    6
             FREEDOM  6  IS FIXED AT ZERO       RELATIVE TO FREEDOM  6  OF NODE    6
   8         TRANSFORMATION NUMBER     1
             FREEDOM  1  IS FIXED AT ZERO       RELATIVE TO FREEDOM  1  OF NODE    7
             FREEDOM  2  IS FIXED AT ZERO       RELATIVE TO FREEDOM  2  OF NODE    7
             FREEDOM  3  IS FIXED AT ZERO       RELATIVE TO FREEDOM  3  OF NODE    7
             FREEDOM  4  IS FIXED AT ZERO       RELATIVE TO FREEDOM  4  OF NODE    7
             FREEDOM  5  IS FIXED AT ZERO       RELATIVE TO FREEDOM  5  OF NODE    7
             FREEDOM  6  IS FIXED AT ZERO       RELATIVE TO FREEDOM  6  OF NODE    7
   7         FREEDOM  1  IS FIXED AT ZERO       RELATIVE TO FREEDOM  1  OF NODE    8
             FREEDOM  2  IS FIXED AT ZERO       RELATIVE TO FREEDOM  2  OF NODE    8
             FREEDOM  3  IS FIXED AT ZERO       RELATIVE TO FREEDOM  3  OF NODE    8
             FREEDOM  4  IS FIXED AT ZERO       RELATIVE TO FREEDOM  4  OF NODE    8
             FREEDOM  5  IS FIXED AT ZERO       RELATIVE TO FREEDOM  5  OF NODE    8
             FREEDOM  6  IS FIXED AT ZERO       RELATIVE TO FREEDOM  6  OF NODE    8
  10         TRANSFORMATION NUMBER     1
             FREEDOM  1  IS FIXED AT ZERO       RELATIVE TO FREEDOM  1  OF NODE    9
```

NODE 10 RESTRAINTS TABLE 3 CONTINUED 20:02:42 25MAR75 PAGE 3

NODE	NATURE OF RESTRAINT		
	FREEDOM 2	IS FIXED AT ZERO	RELATIVE TO FREEDOM 2 OF NODE 9
	FREEDOM 3	IS FIXED AT ZERO	RELATIVE TO FREEDOM 3 OF NODE 9
	FREEDOM 4	IS FIXED AT ZERO	RELATIVE TO FREEDOM 4 OF NODE 9
	FREEDOM 5	IS FIXED AT ZERO	RELATIVE TO FREEDOM 5 OF NODE 9
	FREEDOM 6	IS FIXED AT ZERO	RELATIVE TO FREEDOM 6 OF NODE 9
9	FREEDOM 1	IS FIXED AT ZERO	RELATIVE TO FREEDOM 1 OF NODE 10
	FREEDOM 2	IS FIXED AT ZERO	RELATIVE TO FREEDOM 2 OF NODE 10
	FREEDOM 3	IS FIXED AT ZERO	RELATIVE TO FREEDOM 3 OF NODE 10
	FREEDOM 4	IS FIXED AT ZERO	RELATIVE TO FREEDOM 4 OF NODE 10
	FREEDOM 5	IS FIXED AT ZERO	RELATIVE TO FREEDOM 5 OF NODE 10
	FREEDOM 6	IS FIXED AT ZERO	RELATIVE TO FREEDOM 6 OF NODE 10
12	TRANSFORMATION NUMBER 1		
	FREEDOM 1	IS FIXED AT ZERO	RELATIVE TO FREEDOM 1 OF NODE 11
	FREEDOM 2	IS FIXED AT ZERO	RELATIVE TO FREEDOM 2 OF NODE 11
	FREEDOM 3	IS FIXED AT ZERO	RELATIVE TO FREEDOM 3 OF NODE 11
	FREEDOM 4	IS FIXED AT ZERO	RELATIVE TO FREEDOM 4 OF NODE 11
	FREEDOM 5	IS FIXED AT ZERO	RELATIVE TO FREEDOM 5 OF NODE 11
	FREEDOM 6	IS FIXED AT ZERO	RELATIVE TO FREEDOM 6 OF NODE 11
11	FREEDOM 1	IS FIXED AT ZERO	RELATIVE TO FREEDOM 1 OF NODE 12
	FREEDOM 2	IS FIXED AT ZERO	RELATIVE TO FREEDOM 2 OF NODE 12
	FREEDOM 3	IS FIXED AT ZERO	RELATIVE TO FREEDOM 3 OF NODE 12
	FREEDOM 4	IS FIXED AT ZERO	RELATIVE TO FREEDOM 4 OF NODE 12
	FREEDOM 5	IS FIXED AT ZERO	RELATIVE TO FREEDOM 5 OF NODE 12
	FREEDOM 6	IS FIXED AT ZERO	RELATIVE TO FREEDOM 6 OF NODE 12

EXTERNAL LOADS TABLE 3 CONTINUED 20:02:42 25MAR75 PAGE 4

NODE	FREEDOM	VALUE
11	2	6.0590E 03

A 1 C 1 & 1 CELL LISTING TABLE 3 CONTINUED 20:02:42 25MAR75 PAGE 5

COORDINATES	CTYPE	MTYPE	LIST OF NODES			
1 & 1	1	1	1	2	3	4
1 & 2	1	1	3	4	5	6
1 & 3	1	1	5	6	7	8
1 & 4	1	1	7	8	9	10
1 & 5	1	1	9	10	11	12

NODE 1 NODAL PROPERTIES TABLE 3 CONTINUED 20:02:42 25MAR75 PAGE 6

NODE	NF	GF	COORD 1	COORD 2	COORD 3
1	6	1	1.0000	0.0000	[illegible]
2	6	7	0.5000	0.8660	[illegible]
3	6	13	1.0000	0.0000	.5237
4	6	19	0.5000	0.8660	.5237
5	6	25	1.0000	0.0000	1.[illegible]474
6	6	31	0.5000	0.8660	1.[illegible]474
7	6	37	1.0000	0.0000	1.5711
8	6	43	0.5000	.8660	1.5711
9	6	49	1.0000	[illegible]	2.[illegible]48
10	6	55	0.5000	.8660	2.[illegible]48
11	6	61	1.0000	[illegible]	2.6185
12	6	67	0.5000	0.8660	2.6185

A 1 C 1 & 1 ELEMENT LISTING TABLE 3 CONTINUED 20:02:42 25MAR75 PAGE 7

COORDINATES	NO.	CTYPE	MTYPE		LIST OF NODES	
1 & 1	1	SBEAM	BEAM	1	1	3
	2	SBEAM	BEAM	1	2	4
	3	SBEAM	BEAM	2	1	2
	4	SBEAM	BEAM	2	3	4
1 & 2	1	SBEAM	BEAM	1	3	5
	2	SBEAM	BEAM	1	4	6
	3	SBEAM	BEAM	2	3	4
	4	SBEAM	BEAM	2	5	6
1 & 3	1	SBEAM	BEAM	1	5	7
	2	SBEAM	BEAM	1	6	8
	3	SBEAM	BEAM	2	5	6
	4	SBEAM	BEAM	2	7	8
1 & 4	1	SBEAM	BEAM	1	7	9
	2	SBEAM	BEAM	1	8	10
	3	SBEAM	BEAM	2	7	8
	4	SBEAM	BEAM	2	9	10
1 & 5	1	SBEAM	BEAM	1	9	11
	2	SBEAM	BEAM	1	1[illegible]	12
	3	SBEAM	BEAM	2	9	[illegible]
	4	SBEAM	BEAM	2	11	12

M_1/5.

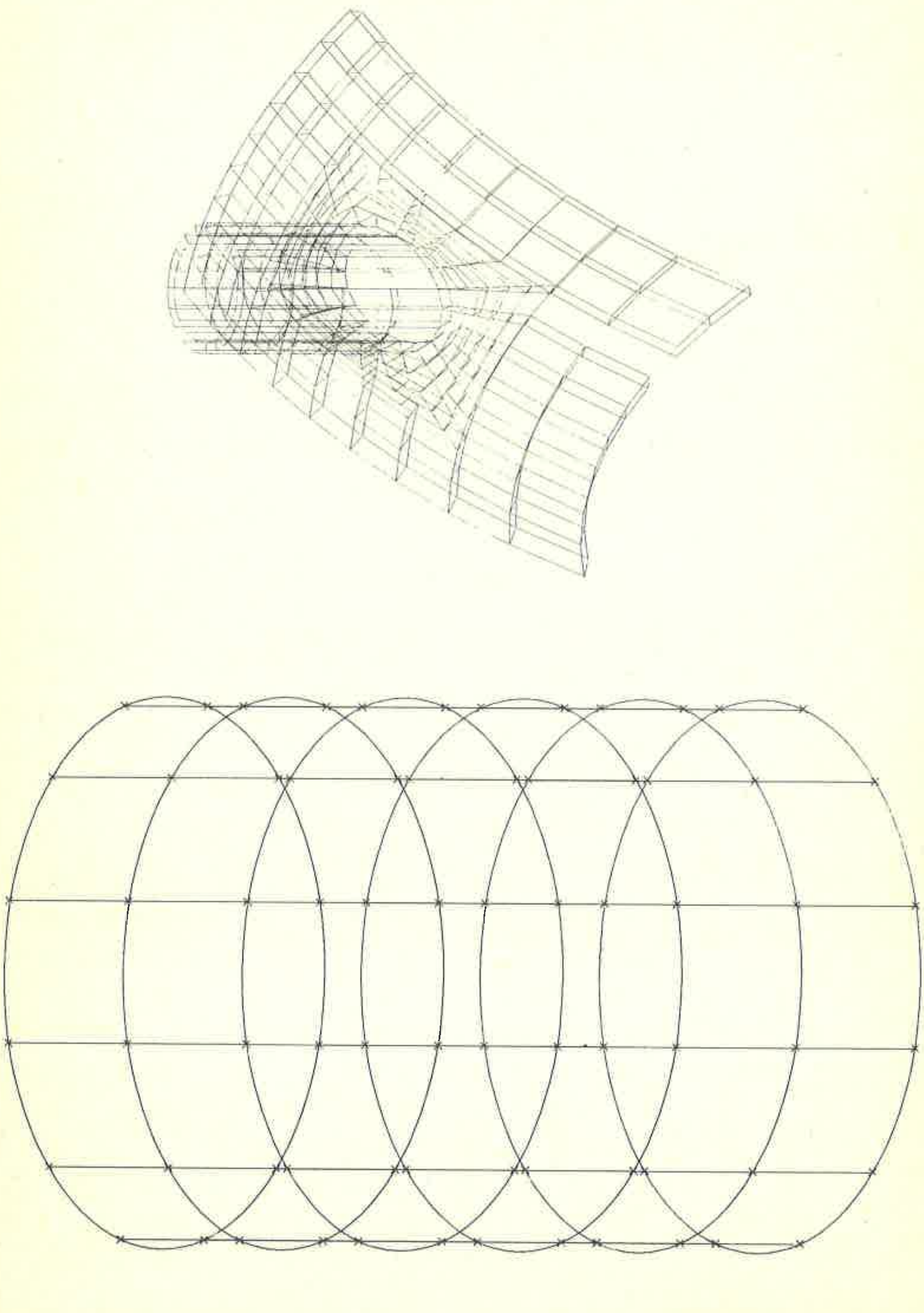

FIG 1 HOLLOW CYLINDER