WFPS-TME-001

JUNE 1975

MASTER

**Westinghouse Electric Corporation
Fusion Power Systems Department**

**FINITE ELEMENT AND NODE POINT GENERATION
COMPUTER PROGRAMS USED FOR
THE DESIGN OF TOROIDAL FIELD COILS
IN TOKAMAK FUSION DEVICES**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

WFPS TME-001

JUNE, 1975

FINITE ELEMENT AND NODE POINT GENERATION
COMPUTER PROGRAMS USED FOR
THE DESIGN OF TOROIDAL FIELD COILS
IN TOKAMAK FUSION DEVICES

Prepared by:

Robert A. Smith

Approved by:

T. C. Varljen
Engineering Manager
TFTR Project

Approved by:

C. K. Kim, Manager
Systems Engr. & Analysis
TFTR Project

Westinghouse Electric Corporation
Fusion Power Systems Department
Post Office Box 10864
Pittsburgh, Pennsylvania 15236

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ABSTRACT

The structural analysis of toroidal field coils in Tokamak fusion machines can be performed with the finite element method. This technique has been employed for design evaluations of toroidal field coils on the Princeton Large Torus (PLT), the Poloidal Diverter Experiment (PDX), and the Tokamak Fusion Test Reactor (TFTR). The application of the finite element method can be simplified with computer programs that are used to generate the input data for the finite element code. There are three areas of data input where significant automation can be provided by supplementary computer codes. These concern the definition of geometry by a node point mesh, the definition of the finite elements from the geometric node points, and the definition of the node point force/displacement boundary conditions. The node point forces in a model of a toroidal field coil are computed from the vector cross product of the coil current and the magnetic field. A computer program named PDXFORC which was originally developed to evaluate the node point forces for stress analysis models of the Poloidal Diverter Experiment, has been documented in another report. The computer programs named PDXNODE and ELEMENT are described in this paper. The program PDXNODE generates the geometric node points of a finite element model for a toroidal field coil. The program ELEMENT defines the finite elements of the model from the node points and from material property considerations. The program descriptions in this work include input requirements, the output, the program logic, the methods of generating complex geometries with multiple runs, computational time and computer compatibility. The output format of PDXNODE and ELEMENT make them compatible with PDXFORC and two general purpose finite element computer codes: (ANSYS) the Engineering Analysis System written by the Swanson Analysis Systems, Inc., and (WECAN) the Westinghouse Electric Computer Analysis general purpose finite element program. The Fortran listings of PDXNODE and ELEMENT are provided as a part of this report.

The geometric finite element modeling options for toroidal field coils provided by PDXNODE include one-fourth or one-half symmetric sections of circular coils, oval shaped coils or dee-shaped coils. These can be modeled with arbitrary thicknesses and with or without a beveled

wedging surface. A multi-run procedure with PDXNODE is described to define a method for generating geometries too complex for a single pass of a mesh generator. The program ELEMENT simplifies the finite element definition input to the finite element program for toroidal field coils whose geometric cross sections are non-uniform, and it allows for a better approximation of material properties in coils that are anisotropic due to the laminated layers of alternate epoxy and copper windings. The added analysis features provided by ELEMENT have been employed in modeling the PLT and TFTR toroidal field coils with integral support structures.

ACKNOWLEDGMENT

This work was supported by U.S. Energy Research and Development Administration Contract AT(11-1)-3073 and Subcontract No. 188 and 192 thereto. Reproduction, translation, publication, use, and disposal, in whole or in part, by or for the United States Government is permitted.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Administration, nor any person acting on behalf of the Administration:

A. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Administration" includes any employee or contractor of the Administration to the extent that such employee or contractor prepares, handles or distributes, or provides access to, any information pursuant to his employment or contract with the Administration.

APPLIED TECHNOLOGY

Any further distribution by any holder of this document, or of the data therein, to third parties representing foreign interests, foreign governments, foreign companies and foreign subsidiaries, or foreign divisions of United States companies, should be coordinated with the Director, Division of Thermonuclear Research, United States Energy Research and Development Administration.

TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
	ABSTRACT	ii
	ACKNOWLEDGMENT	iii
1.0	INTRODUCTION	1-1
2.0	PROGRAM PDXNODE	2-1
	2.1 Output Description of PDXNODE	2-1
	2.2 Input Description for PDXNODE	2-2
	2.3 PDXNODE Program Logic	2-6
	2.4 Control Cards for Multiple Geometry Generations	2-7
	2.5 Computation Time and Computer Compatibility for PDXNODE	2-9
3.0	PROGRAM ELEMENT	3-1
	3.1 Output Description of ELEMENT	3-1
	3.2 Input Description of ELEMENT	3-2
	3.3 ELEMENT Program Logic	3-3
	3.4 Computation Time and Computer Compatibility for ELEMENT	3-13
4.0	REFERENCES	4-1
<u>Appendix</u>	<u>Title</u>	<u>Page</u>
A	PDXNODE - PROGRAM LISTING	A-1
B	ELEMENT - PROGRAM LISTING WITH INPUT AND PARTIAL OUTPUT	B-1

LIST OF FIGURES

<u>Section</u>	<u>Title</u>	<u>Page</u>
1-1	Geometry and Finite Element Numbering for a Symmetric-half of a Toroidal Field Coil for the Computer Program PDXNODE	1-3
1-2	Notation Internal to Program PDXNODE	1-4
2-1	Example of PDXNODE Program Input Changes for the Generation of a DEE-Shaped Coil (from the TFTR Computation No. 2)	2-5
3-1	Example Finite Element Model for a Symmetric Section of a Toroidal Field Coil (taken from TFTR Computation 15)	3-4
3-2	Node Numbers at the First Layer of Node Points (Add NONC = 105 to these for Node Numbers at the Second Layer of Node Points)	3-5
3-3	The Input Finite Element Definition Card Locations Between the First Two Layers of Node Points	3-6
3-4	Coil Model Cross-Section at Element Layers 1, 2, 6, 7, and 8	3-7
3-5	Coil Model Cross Section at Element Layers 3 to 5, 9 to 17, 23 to 32, and 39 to 45	3-8
3-6	Coil Model Cross-Section at Element Layers 18, 19, 21, and 22	3-9
3-7	Coil Model Cross-Section at Element Layer 20	3-10
3-8	Coil Model Cross-Section at Element Layers 33, 34, 35, 36, 37, and 38	3-11

1.0 INTRODUCTION

The finite element stress analyses of toroidal field (TF) coils in experimental fusion machines have been performed on the Princeton Large Torus⁽¹⁾ (PLT), the Poloidal Diverter Experiment⁽²⁾ (PDX) and the Tokamak Fusion Test Reactor⁽³⁾ (TFTR). The source of loads imposed on a TF coil is the vector cross product of the coil current and the magnetic field. A method of evaluating these loads (Computer Program PDXFORC) is written and documented⁽⁴⁾ to be compatible with two general purpose finite element computer programs. These are ANSYS⁽⁵⁾ and WECAN.⁽⁶⁾ This work describes two additional computer programs that are compatible with ANSYS, WECAN, and PDXFORC. These programs are referred to as PDXNODE and ELEMENT. The program PDXNODE is a mesh generator for the node points that can be used to describe a finite element model of a toroidal field coil. The program ELEMENT in turn defines the finite elements with the node numbers, the material types, and the material numbers. The outputs of PDXNODE and ELEMENT are in a format so that they can be accepted from tape as input to ANSYS or WECAN. The choice is a users option. This documentation is sufficient to provide the method that one would employ to make node point and element definitions compatible with any general purpose finite element computer program. The Fortran coding of these computer programs is listed in the appendices of this paper.

The computer program PDXNODE generates a geometry with node points that are used to describe a finite element model of a toroidal field coil. The node points are assigned a number and a position within the finite element model in a global Cartesian coordinate system. The output format of the node point description is consistent with the input to two general purpose finite element computer programs. The output options are compatible with the WECAN input card 4.1 or the ANSYS input card F. The generated node points are used to describe the eight corners of the three-dimensional finite element. The computer program ELEMENT generates the finite element definition cards (or tape) in a format compatible with the WECAN input card 3.1 or the ANSYS input card E1.

The finite element geometry model that is used to represent toroidal field coils is generalized on Figure 1-1. This model shows three (3) distinct layers separated by gaps of arbitrary thickness. Three layers is not a limit. The finite element model can be one layer, or it can be greater than three layers. The number of finite elements shown through the thickness of each layer is respectively N_2 , N_3 , and N_4 . The model is also described by an outer leg with N_1 finite element spanning the G_7 length, an arc section with N_6 finite elements spanning the π radians and an inner leg with N_1 finite elements. The number of finite elements through the coil thickness G_{11} is N_5 . The modeling technique which enables the generation of an inner leg of different height from the outer leg is available. It is discussed in connection with the "DEE" shaped coil model. An additional modeling option is a coil, of one-fourth symmetry. All of the dimensions shown for the model on Figure 1-1 are required input to the computer program PDXNODE, and the nomenclature of Figure 1-1 is consistent with the program listing provided in Appendix A. Figure 1-2 which shows some of the notation internal to the program PDXNODE will be used to illustrate the discussion on the program logic.

The required input data to the program ELEMENT is the element definition cards (WECAN 3.1 or ANSYS E1) for the first layer of finite elements defined by the first two layers of node points at $K1 = 1$ and $K1 = 2$ on Figure 1-2. Normally this also would be the only required input to ANSYS or WECAN. The computer program ELEMENT is of benefit beyond normal input requirements when it is desirable to define different materials at each layer in the $K1$ internal count direction to approximate material anisotropy as was the case for the PLT toroidal field coil analysis.⁽¹⁾ Similarly, the program ELEMENT is an efficient tool for defining the elements in a coil where the cross-section is changing in such a way that the number of finite elements is not uniform in the $K1$ internal count direction. The example provided in Section 3.0 of this paper best illustrates the program ELEMENT usages. A listing is provided in Appendix B.

Figure 1-1. Geometry and Finite Element Numbering for a Symmetric-half of a Toroidal Field Coil for the Computer Program PDXNODE.

Figure 1-2. Notation Internal to Program PDXNODE

2.0 PROGRAM PDXNODE

The computer program PDXNODE has been used to generate the node points of finite element models for the structural analysis of toroidal field coils. The geometric options include one-fourth or one-half symmetric sections of circular coils like the TFTR and the PLT, oval shapes like the PDX or a DEE shape to include two different radii with arbitrary inner and outer leg heights. Any of these shapes may be modeled with or without a beveled wedging surface. The sections to follow describe the program output, the input requirements, the logic, the methods for generating complex geometries with multiple runs, computational time, and computer compatibility.

2.1 OUTPUT DESCRIPTION OF PDXNODE

There are five write statements to be found in the program listing in Appendix A. The first on line 178 is an indicator (ISUP = +99999) for ANSYS or WECAN to suppress printing of the input node point data. The last two write statements on lines 289 and 291 are the respective indicators for ANSYS and WECAN that the input of node point data is ended. The node point input data for the ANSYS option is written at line 279. The node point number is JI; its location in the global cartesian coordinate system is XX, YY, ZZ. This same information is written at line 281 for the WECAN option. The extra feature provided by the ANSYS option as shown by the line 279 write statement, is the provision for making a local coordinate system rotation for the node points defined by THXY, THYZ, and THXZ. This option is available in the WECAN program as part of the boundary condition input, and therefore any local coordinate system rotations for the node points in WECAN are performed with the computer program PDXFORC. The provision for this rotation when using either program is for the purpose of being able to specify force or displacement constraints in the same direction as the rotated local coordinate system. This rotation occurs at the boundary node points of a coil with a beveled wedging surface when it is modeled to have zero displacement normal to the wedge surface.

2.2 INPUT DESCRIPTION FOR PDXNODE

The geometry and finite element model information shown on Figure 1-1 are the basic coil input data. The coil dimensions G1 through G12 are input at lines 66 through 77 in the program listing. The only permissible zero is G7. The input data for the desired node point or finite element arrangement N1 through N6 is found in the listing at lines 58 through 63. The only permissible zero input here is N1, and this would correspond to a zero input for G7. All other input data to the program would define a desired modeling or printing option. The model input options are listed from line 42 through 53, with the following descriptions:

WECAN If this is the finite element program to be used, set the value at 1.0; otherwise, let WECAN = 0.0.

ANSYS If this is the finite element program to be used, set the value at 1.0; otherwise, let ANSYS = 0.0.

GEOM This is a key that will cause some write statements to be bypassed. When the value is set at 1.0, all of the write statements in the listing will execute. If the value is greater than 1.0, only the write statement on line 279 or 281 will be executed depending on the value of ANSYS and WECAN. This option is used to combine the output of several PDXNODE runs onto a single tape to form a geometry too complex for a single run. The first run (GEOM = 1.0) will contain all of the required input information for WECAN or ANSYS, and then subsequent runs (GEOM > 1.0) will contain only the node point information which is inserted into the node information on the attached tape from the previous run. Thus the final output format is always acceptable as tape input to the finite element program.

O4SYM This is a symmetry key. If the desired model is a geometry of one-fourth symmetry, set the value at 1.0, and the node point information will only occupy the region defined by $Z \geq 0.0$ and $X \geq 0.0$ on Figure 1-1. With this option N5 should be an even number. When O4SYM is set at 0.0 a one-half geometric model is generated to occupy the region defined by $Z \geq 0.0$ on Figure 1-1.

WEDGE This is a coil shape key. When the value is set at 1.0, a bevel will be generated at the coil wedge region consistent with the input geometry. If this is not desired as may be the case for constructing a geometry from multiple runs, set the value at 0.0, and the bevel will not be generated whereas the coil thickness will remain constant at the input value of G11.

DELTA This is a translator of node point position in the X-direction. The value would normally be input as 0.0. When an X-direction shift for all the node points is desired, the value is input as DELTA.

DELTH A positive input for this parameter will decrease the coil thickness G11 by that amount. This does not affect the coil wedge location C3, defined on Figure 1-2. Normally DELTA and DELTH would both be zero or both be non-zero.

S This is converter of units from input to output. Let the value be 1.0 if output is desired to be of the same units as input. If inputs are in meters and output is desired to be inches, let $S = 39.37$.

DEE This is a key for modifying the coil shape. If a value of 0.0 is input, the dee-shape modification data input at the listing lines 100 through 119 is ignored. When a value of 1.0 is input for DEE, the input values of G1 through G12

and N1 through N6 at the listing lines 58 through 77 are used to generate the geometry for $T1 \leq \pi/2$ defined by Figure 1-2. When $T1 > \pi/2$ with the input of $DEE = 1.0$, new values of geometry and element input are read and used to generate the geometry. These are on the listing lines 108 through 119. This is further illustrated by the example provided by Figure 2-1. The geometry input values designated by GD are the same as the G relative to the Figure 1-1 definitions.

COILS This is the number of toroidal field coils which are symmetrically located about the machine centerline.

LOCAL This is a coordinate system key which is an ANSYS option. It is available for WE CAN in the program PDXFORC. With the value set at zero, there will be a local coordinate rotation only at the beveled surface of the coil. If $LOCAL = 1$, all of the node points in the coil will be given a rotation in the YZ plane such that the nodal displacements and forces will be in the coil R, θ cylindrical system. The global Y becomes the local R and the global Z would become the local θ (or T1). The program PDXFORC computes the node forces in the global X, Y, Z, and therefore the option $LOCAL = 1$ can only be used for poloidal field analyses or at nodes where there is no Y or Z component of node point forces. This local rotation would input the Y or Z component of force in the wrong (rotated) direction. The $LOCAL = 1$ option will provide the displacement solution in the coil R, θ directions.

INK This is a constant increment that is added to all the node point numbers. When more than one run of PDXNODE is used to generate a geometry, the value of INK must be set to avoid having more than one node point with the same node number.

Figure 2-1. Example of PDXNODE Program Input Changes for the Generation of a DEE-Shaped Coil (From the TFTR Computation No. 2)

The remaining data input options pertain to suppressing the output of geometric regions that are not required by the model. These options are found in the program listing from lines 79 through 97. In the program there are three indices for node generation, and these are shown on Figure 1-2 as K, M, and K1. The input values of MINK and MAXK at lines 85 and 86 will limit the output to that range of K which generates the node points in the X-direction. The limits of output for the Z or T1 directional counter K1 are defined by lines 90 and 91. Similarly, the range of output in the Y or R direction can be limited by the specification of values for MINM and MAXM at the listing lines 95 and 96.

2.3 PDXNODE PROGRAM LOGIC

The definition of program parameters and the input data previously described is found within the first 121 lines of the listing. The program lines 99 through 144 are only entered if an input of DEE = 1.0 is entered at line 50. Otherwise, that input information may be ignored. From line 146 through line 176, the notation internal to the program which is shown on Figure 1-2 is defined from the input information. The main DO loop spans lines 180 through 286. The index $K1 = 1, J$ shows the direction of node point generation on Figure 1-2 to be up the outer leg in the positive Z-direction, through the arc and then down the inner leg in the negative Z-direction. At every value of K1 there are L times NN5 node points generated on the K1 plane. At every value of K1 and M there are NN5 node points generated along a line in the X-direction. The node points have equal spacing within a coil region in a given direction. The main DO loop counter K1 is initialized in the arc region as K2, and in going down the inner leg this count is re-initialized by K3. The Z-coordinate of node points on the coil outer leg is defined at line 185, and on the coil inner leg at line 193. The Y-coordinate of node points on the inner and outer leg portions are computed within the listing lines 198 through 217. On lines 198 through 203, P1 through P6 define the Y-coordinates of the inner and outer surfaces of each of the three layers for both the inner and outer legs. ITEST = 1 means the computation is on a leg ($T1 = 0$ or π), and ITEST = 2 means the computation is in

the arc region ($0 < T1 < \pi$). This is for return purposes in the main DO loop. See line 285, GOTO(3, 8), ITEST. The Y and Z-coordinates of the node points in the arc region are computed between lines 225 and 253. After each Y computation, there is a GOTO 28. This is where the NN5 node values of X coordinate locations are generated. This loop includes lines 264 through 282. Within lines 255 through 261, the coil thickness is TH and AA is used to define the local coordinate rotation THXY defined on line 274. The value of TH and AA are redefined in the wedge region only when $Y \leq C3$. At line 273 the X coordinate location of the nodes are computed. The JI on line 269 is the node point number.

2.4 CONTROL CARDS FOR MULTIPLE GEOMETRY GENERATIONS

The node point model for a toroidal field coil with uniform cross section and for which material isotropy and homogeneity will be assumed can generally be assembled with a single run of PDXNODE. When the model is to include with the copper windings, features such as inter-laminate insulation, coil steel support structures like copper wound in the channels of an I-beam or copper encased in a box beam, then multiple runs of PDXNODE can be made where the information from the latter run is inserted on the tape containing the information from the previous run.

When more than one run is made, a convenience in the stress code element numbering is obtained if the total node points per layer ($L * NN5$) can be maintained constant for the different runs. Unused node points are acceptable input to ANSYS or WECAN, but to minimize printed information and to minimize the maximum node point number, the printout suppressor options defined in the listing lines 79 through 97 are available. A recommended scheme to employ for multiple generations is to use the suppressed node numbers from previous runs as the printed node numbers in the latter runs. This will minimize the value of INK in the latter runs. A basic understanding of the limits imposed by the core storage of the computer facility and by the finite element program is essential to the understanding of modeling limits. A study of this nature is found for the ANSYS program on the CDC-7600 computer at the Westinghouse computer facility in Appendix F of Reference (1).

The following shows the control card setup employed on the Westinghouse CDC-7600 system for multiple generations:

DECK SETUP: RUN 1

Priority Card
Account Card
Compiler Control Cards
LGO.
UPDATE (N, D)
PURGE (, WMPX100, PW = CONTROL)
CATALOG (NEWPL, WMPX100, DS = \$ NODES, RUN 1 \$, ID = SCRAS)
REWIND (TAPE 6)
COPYSP (TAPE 6, OUTPUT)
☐ ← 7, 8, 9 Card
PROGRAM PDXNODE (TAPE 6, OUTPUT)
 • with
 • GEOM = 1.0
END
☐ ← 7, 8, 9 Card
* DECK, NODE 1
* REWIND, TAPE 6
* READ, TAPE 6
☐ ← 7, 8, 9 Card

DECK SETUP: RUN 2 or More

Priority Card
Account Card
Compiler Control Cards
LGO.
ATTACH (OLDPL, WMPX100)
UPDATE (N, D, L = 0)
PURGE (, WMPX200, PW = CONTROL)
CATALOG (NEWPL, WMPX200, DS = \$ COMBINED NODES FROM RUN 1 AND
 RUN 2 \$, ID = SCRAS)
REWIND (COMPILE)
COPYSP (COMPILE, OUTPUT)
☐ ← 7, 8, 9 Card

```

        PROGRAM PDXNODE (TAPE 6, OUTPUT)
        •       with
        •       GEOM = 2.0 or greater)
        □ ← 7, 8, 9 Card
        * IDENT, NODE 2
        * INSERT, NODE 1.2
        * REWIND, TAPE 6
        * READ, TAPE 6
        □ ← 7, 8, 9 Card

```

The node points from the first geometry run are catalogued on the updateable tape (WMPX100), where the deck name is NODE 1. The first printed (or catalogued) line of information, +99999, is named NODE 1.2. The second line of information is named NODE 1.3, and this is the information for the first node point. During the second or later runs, the nodes from the previous run are attached, ATTACH (OLDPL, WMPX100). The new information of the second run is generated and identified as NODE 2. It is then inserted after the +99999 line of information from the attachment, and then this is all catalogued as TAPE (NEWPL, WMPX200). When GEOM is 2.0 or greater, the ISUP and the IEND write statements are not executed, and therefore the insertion of new node point information does not alter an acceptable format of input to WECAN or ANSYS. For a third run the attachment tape would be (WMPX200) and, the catalog might be (WMPX300), the new identification could be *IDENT, NODE 3; and the insertion location can always remain as *INSERT, NODE 1.2, which places the newly generated node data between the +99999 card and the previously generated data.

2.5 COMPUTATION TIME AND COMPUTER COMPATABILITY FOR PDXNODE

The program PDXNODE as it is shown in Appendix A, has been used on the CDC-7600 computer. The programming is compatible with the IBM-360 machine. A typical JOB time on the CDC-7600 is about 6.0 decimal seconds for about 5000 node points. The program compilation time is about 0.60 CP seconds, and the execution time is about 0.00035 CP seconds per printed node point.

3.0 PROGRAM ELEMENT

The computer program ELEMENT generates the finite element definition cards in a format acceptable to two finite element computer programs. These correspond to the input card 3.1 for the WECAN program or the input card E1 for ANSYS. A listing of the program is provided in Appendix B where the control cards, input data, and a partial output is also shown.

3.1 OUTPUT DESCRIPTION OF ELEMENT

There are five write statements to be found in the listing at lines 69, 94, 97, 103, and 105. The `ISUP = +99999` written on line 69 is a key to both ANSYS and WECAN to suppress the printout of this input data. At line 103 `IEND = -1` is the indicator to ANSYS that the element input data has ended, and the line 105 write statement is the indicator for end of element input data to WECAN. The line 94 write statement is the element input data information for the ANSYS finite element computer program, and line 97 is for the WECAN program. The eight values of `NODE` are the node points that define the locations of the eight corners of the three-dimensional elastic or isoparametric solid. `MAT` is the index which assigns the material properties to this element. `ITYPE` is the index relating this element to a number from the element library table. For output to the WECAN program, the data on line 97 shows only a reverse of format locations for `MAT` and `ITYPE`. `ITABLE` refers to the index that would identify the real constant table if real constants are required for the elements. There are no real constants required for the 3-D elastic elements in WECAN or ANSYS. `INUM` is the number of first-level elements generated by this element card including this element card in the count, and `NINC` is the increment to be added to the node numbers on this card for the generation of each of the `INUM` elements. The output information defined by `NUMELF`, `NUMELL`, and `K` are not input data to WECAN or ANSYS. Their format locations are such (beyond column 90) that it is not recognized as input information from the updateable catalogued tape. `NUMELF` is the number of the first finite element generated by that card, and `NUMELL` is the number of the

last finite element generated on that card. The index K identifies the elements to be in a certain layer. There are a total of $2N1 + N6$ layers of elements defined by the notation of Figure 1-1 in this paper. The partial output from the program ELEMENT shown in Appendix B is for the first three layers. The first line of output is the ANSYS input data for the first ten finite elements. The first layer of output represents the ANSYS input for 54 finite elements. It is noted here that the first 80 columns of output from ELEMENT for the first layer ($K = 1$) of finite elements is identical to some of the required input data shown on the program listing lines 119 through 137.

3.2 INPUT DESCRIPTION FOR ELEMENT

The input data for program ELEMENT are listed at lines 50 through 54 and by the READ statement at line 63. At lines 50 and 51 a 1.0 for WECAN or ANSYS will format the output of ELEMENT for appropriate input to the users finite element program. A 0.0 is required for the option not chosen. At line 52, NELC is the number of input data cards to be read by ELEMENT, and these cards are shown at the listing lines 110 through 137. The input data cards must define a generalized layer of finite elements that can be manipulated to generate the total $2N1 + N6$ layers of elements. At line 53, LAYER is the $(2N1 + N6)$ total number of layers of elements to be generated. At line 54, NONC is the constant node number increment to be added to the node numbers of the last layer of elements to define the node numbers for the next layer of elements. Referring to the notation on Figure 1-2, NONC is equal to $(N2 + N3 + N4 + 3)$ times $(N5 + 1)$. This is the desired constant between runs when more than one geometry generation is required by the program PDXNODE. The description of the input data for the READ statements on lines 63 or 66 is identical to the description previously provided for the WRITE statements on lines 94 and 97. All of the logical IF statements containing J or K in the program should be thought of as input data to the program, because they are used to alter the information provided by the generalized input for the first layer of finite elements. The program logic as related to these input IF statements is described in the following section by using the illustration of a sample model.

3.3 ELEMENT PROGRAM LOGIC

The computer program PDXNODE generates with reference to Figure 1-1, $2N1 + N6 + 1$ layers of node points to define a symmetric section of a toroidal field coil. The region between any two layers of node points can be identified as a layer of three-dimensional finite elements. The structural model of the toroidal field coil is then defined by $2N1 + N6$ layers of finite elements. If the number of elements in each layer were constant and if the material number, types and tables were the same at each layer, then the program ELEMENT would not be used. For this situation the input to WECAN or ANSYS could be the first layer of finite elements plus one additional card called a second level element generator. When a coil with its supporting structural cross-section is non-uniform or when material properties within the model global coordinate system vary with coil angular position due to winding direction laminations, then the program ELEMENT can be used to avoid hand writing and punching of the finite element cards for every layer of finite elements in the model. The program logic is to take a generalized first layer of finite element data as input and to manipulate this data to generate the element definition cards for every layer in the model. In general the input data to ELEMENT will contain more information than what is required at each layer, and only the required information is printed to define the elements at a given layer.

The method or program logic is illustrated with an example model. The model is shown on Figure 3-1, and it was generated with four runs of PDXNODE where $N1 = 0$, $N2 = N3 = N4 = 4$, $N5 = 6$, and $N6 = 45$. The value of NONC = 105 was held constant for each run. Figure 3-2 illustrates the node point numbering scheme for the first layer of node points. The node numbers at the second layer of node points would be 105 greater than those shown at the first layer. Figure 3-3 shows the locations of the element definition cards ($J = 1, 28$) for the first layer of elements as defined by the first two layers of node points. These are the 28 input cards shown in Appendix B for the program ELEMENT. The arrow shows the elements defined by each card with its INUM value. Figures 3-4 through 3-8 show the model cross-section for every layer

Figure 3-2. Node Numbers at the First Layer of Node Points
(Add NONC = 105 to these for Node Numbers at
the Second Layer of Node Points)

Figure 3-3. The Input Finite Element Definition Card Locations Between the First Two Layers of Node Points

($J \geq 10$, $INUM = INUM + 2$ When $J = 23$ or 28)

Figure 3-4. Coil Model Cross-Section at Element Layers 1, 2, 6, 7, and 8

Figure 3-5. Coil Model Cross-Section at Element Layers 3 to 5, 9 to 17, 23 to 32, and 39 to 45

Figure 3-6. Coil Model Cross-Section at Element Layers 18, 19, 21, and 22

Figure 3-7. Coil Model Cross-Section at Element Layer 20

($J \geq 10$, $INUM = INUM + 2$ When $J = 10, 13, 18, 23, 28$)

Figure 3-8. Coil Model Cross-Section at Element Layers 33, 34, 35, 36, 37, and 38

of finite elements. These sections are not shown to geometric scale relative to Figure 3-1, but the element/node numbering patterns shown are schematically correct. The variances from the input data are shown on the Figures 3-4 through 3-8.

The outer DO loop in the program ELEMENT between lines 70 and 100 indexes the 45 layers of the model. The next DO loop between lines 71 and 99 operates on the 28 input element cards at each of the 45 layers. The first two cards in this loop on lines 72 and 73 are of the input type. The card on line 72 prevents element printout for any layer less than 18 or greater than 22 when J is less than 10, because the information on the first nine element input cards is used only at element layers 18 through 22 as shown by Figures 3-6 and 3-7. The card on line 73 allows the element information on input cards J = 2, 5 or 8 to be printed only at the element layer 20 as shown on Figure 3-7. The combined logic of these two cards at lines 72 and 73 permits the J = 1, 3, 4, 6, 7, and 9 input data cards to generate elements only at layers 18, 19, 21, and 22 as shown on Figure 3-6.

At the listing lines 74 and 75 the innermost DO loop generates the eight node points for the element on card J at layer K from the input nodes. Lines 76 through 82 show the other element definition parameters to be set equal to the input values. INUM was re-initialized here because it was changed on the very first layer at the input cards J = 23 and 28 as shown by Figure 3-4. The input type card shown on lines 89 and 90 performed the INUM change indicated by Figure 3-4, and the INUM input data variance indicated by Figure 3-8 was executed by the input-type card on lines 87 and 88. The only other data manipulation was performed on the input values of MAT for the copper/epoxy laminate material corresponding to J = 16, 21, and 26. This is shown at the listing lines 83 through 86. Material property input data to the ANSYS or WECAN finite element computer programs is input for the global X, Y, Z-coordinate directions. A toroidal field coil winding like that on the Princeton Large Torus (PLT) would exhibit cylindrical material property directions in a local R- θ coil coordinate system. To

provide a mechanism for better approximating the material properties, the program ELEMENT assigns a different material number to every element layer at different angular locations on the coil. Layers 1 and 45, 2 and 44, etc. have the same angular location. Thus 23 different values of MAT are assigned to the 45 layers of elements occupying the $J = 16, 21$ and 26 input data locations.

3.4 COMPUTATION TIME AND COMPUTER COMPATIBILITY FOR ELEMENT

The program ELEMENT as it is shown in Appendix B, has been used on the CDC-7600 computer. The programming is compatible with the IBM-360 machine. A typical JOB time on the CDC-7600 is about 3.0 decimal seconds for about 1000 lines of output. The program compilation time is about 0.40 CP seconds and the execution time is about 0.00058 CP second per printed line.

4.0 REFERENCES

- 1) Smith, R. A. and Fernandez, R., A Summary of Toroidal Field Coil Finite Element Stress Analysis for the Princeton Large Torus, WANL-TME-2869, May 30, 1975, Westinghouse Astronuclear Laboratory, P.O. Box 10864, Pittsburgh, Pa. 15236.
- 2) Smith, R. A., The Toroidal Field Coil Stress Analysis Method and Results for the Princeton PDX and PLT Controlled Fusion Experiments, WANL-TME-2860, June 30, 1974, Westinghouse Astronuclear Laboratory, P.O. Box 10864, Pittsburgh, Pa. 15236.
- 3) Two Component Torus Joint Conceptual Design Study, Volume III, Appendix A, Plasma Physics Laboratory, Princeton, New Jersey and Westinghouse Electric Corporation, Pittsburgh, Pennsylvania, June 1974.
- 4) Smith, R. A., Field Load and Displacement Boundary Condition Computer Program used for the Finite Element Analysis and Design of Toroidal Field Coils in a Tokamak, WFPS-TME-002, June 30, 1975, Westinghouse Electric Corp., Fusion Power Systems Department, P.O. Box 10864, Pittsburgh, Pa. 15236.
- 5) Swanson, J. A. and DeSalvo, G. J., ANSYS, Engineering Analysis Systems User's Manual, October 1, 1972, Elizabeth, Pennsylvania 15037.
- 6) WECAN, Westinghouse Electric Computer Analysis User's Manual, February 15, 1973, Proprietary Class 2, Westinghouse Research Laboratories, Pittsburgh, Pa. 15235.

APPENDIX A

PDXNODE - PROGRAM LISTING

```

1 — PROGRAM PDXNODE(TAPE6,OUTPUT)
C *** A PROGRAM TO GENERATE FINITE ELEMENT NODE POINTS FOR TOROIDAL **
C *** FIELD COILS IN FUSION DEVICES LIKE A TOKAMAK. THE NODES ARE **
C *** IN A CARTESIAN SYSTEM FOR COMPUTER PROGRAMS **
C *** ANSYS INPUT CARD F **
C *** OR **
C *** WECAN INPUT CARD 4.1 **
C *** PROGRAM ORIGINALLY WRITTEN FOR PDX TF COIL ANALYSIS,NOV.,1973,**
C *** BY R.A.SMITH,WESTINGHOUSE ASTRONUCLEAR FOR PRINCETON PLASMA **
10-C — *** PHYSICS LABORATORY,AEC CONTRACT AT(11-1)-3073,SUBCONTRACT 188.**
      DIMENSION XX(9000),YY(9000),ZZ(9000),THXY(9000)
C *** THE DIMENSIONING OF VARIABLES IS NOT NECESSARY IN THIS PROGRAM.***
C
C **** TFTR FIRST GEOMETRY RUN FOR COMPUTATION 16 ***
C
C **** DEFINITIONS AND INPUT ***
C **WECAN=THE WECAN KEY. WECAN=1.0 IS FOR WECAN FORMAT. =0.0 IS DENIAL
C **ANSYS=THE ANSYS KEY. ANSYS=1.0 IS FOR ANSYS FORMAT. =0.0 IS DENIAL
C **GEOM=THE RUN NUMBER FOR MULTIPLE GENERATION CASES
20-C — **O4SYM=A GEOMETRY KEY.WHEN O4SYM=1.0,THE GEOMETRY IS ONE-FOURTH
C * SYMMETRY AND N5 SHOULD BE EVEN. WHEN O4SYM=0.0,GEOM IS ONE-HALF.
C **WEDGE=A COIL SHAPE KEY. LET WEDGE=1.0 FOR A WEDGED SHAPE. FOR A
C * COIL OR PART OF A COIL WHERE NO BEVEL IS DESIRED,LET WEDGE=0.0.
C **DELTA=AN X-DIRECTION SHIFT OR TRANSLATION FOR ALL THE NODE POINTS
C **DELTH=A REDUCTION IN COIL THICKNESS(G11)THAT DOES NOT EFFECT THE
C * COIL WEDGE LOCATION(C3).INPUT IS POSITIVE .NORMALLY USED WHEN
C * DELTA IS NON ZERO.
C **S=CONVERTER FOR UNITS ON INPUT/OUTPUT. LET S=1.0 FOR SAME UNITS
C * ON OUTPUT AS INPUT. LET S=39.37 IF ALL INPUTS ARE IN METERS,AND
30-C — * THE DESIRED OUTPUT IS INCHES.
C **DEE=A COIL SHAPE KEY.WHEN DEE=1.0,A DEE SHAPED COIL WILL BE
C * GENERATED WITH INPUT DATA MODIFICATIONS. WHEN DEE=0.0 A DEE SHAPE
C * IS DENIED AND SUBSEQUENT GEOMETRY CHANGES WILL BE IGNORED.
C **COILS=THE NUMBER OF TOROIDAL FIELD COILS IN THE FUSION DEVICE
C **LOCAL=A COORDINATE SYSTEM KEY. AN ANSYS OPTION ONLY. AVAILABLE IN
C * PROGRAM PDXFORC FOR WECAN. IF LOCAL=1,THE COIL NODE POINTS WILL
C * RECEIVE A LOCAL ROTATION INTO THE COIL R-THETA COORDINATE SYSTEM.
C * IF LOCAL=0,NODES WILL REMAIN IN GLOBAL XYZ EXCEPT IN WEDGE REGION.
C **INK=A CONSTANT INCREMENT ON THE NODE NUMBERS
40-C — **ISUP=A PRINTOUT SUPPRESSOR IN BOTH WECAN AND ANSYS PROGRAMS

```

```

41 -C  **IEND=INDICATOR FOR ANSYS END OF NODE DATA
      WECAN=0.0
      ANSYS=1.0
      GEOM=2.0
      O4SYM=0.0
      WEDGE=1.0
      DELTA=0.0
      DELTH=0.
      S=1.0
50 — DEE=0.0
      COILS=20.
      LOCAL=0
      INK=5000
      ISUP=+99999
      IEND=-1
      C  **N1 THRU N6=FINITE ELEMENT NUMBER DEFINITIONS. THE ONLY PERMISSIBLE
      C  * ZERO IS N1.
      N1=0
      N2=4
60 — N3=4
      N4=4
      N5=6
      N6=45
      C  **G1 THRU G12 ARE THE GEOMETRY INPUT DATA FOR A CIRCULAR COIL.THE
      C  * ONLY PERMISSIBLE ZERO IS G7.
      G1=4.
      G2=.2
      G3=16.8598
      G4=.2
70 — G5=4.
      G6=110.236
      G7=.0
      G8=4.
      G9=16.8598
      G10=4.
      G11=16.2086
      G12=53.118
      C  **END OF GEOMETRY INPUT DATA FOR A CIRCULAR COIL
      C  *** START OF PRINTOUT SUPPRESSOR OPTIONS ***
80 -C—

```

81 -C **MINK *THE RANGE FOR WHICH NODE PRINTOUT IS DESIRED FOR THE K INDEX
C * MAXK *, WHICH IS THE X-DIRECTIONAL COUNTER. FOR FULL PRINTOUT LET
C * * MINK=1 AND LET MAXK=N5+1. THE 04SYM=1.0 OPTION CAN FURTHER
C * * DECREASE THE MAXK LIMIT.
MINK=1
MAXK=1+N5
C **MINK1 *THE RANGE OVER WHICH PRINTOUT IS DESIRED FOR THE K1 INDEX
C * MAXK1 *WHICH IS THE THETA/Z-DIRECTIONAL COUNTER. FOR FULL PRINTOUT
C * * LET MINK1=1 AND LET MAXK1=2*N1+N6+1
90 — MINK1=1
MAXK1=2*N1+N6+1
C **MINM *THE RANGE OVER WHICH PRINTOUT IS DESIRED FOR THE M INDEX
C * MAXM *WHICH IS THE Y/R-DIRECTIONAL COUNTER. FOR FULL PRINTOUT
C * * LET MINM=1 AND LET MAXM=N2+N3+N4+3
MINM=1
MAXM=N2+N3+N4+3
C *** END OF PRINTOUT SUPPRESSOR OPTIONS ***
GOTO99
98 CONTINUE
100 -C — **THE FOLLOWING ARE GEOMETRY CHANGES FOR A DEE SHAPED COIL *****
C **G2,G4, AND G11 DO NOT CHANGE TO OBTAIN THE DEE GEOMETRY.
DEE=0.0
GD2=G2
GD4=G4
GD11=G11
C ***** START OF DEE DATA CHANGES *****
C **A ZERO INPUT FOR ANY GD IS NOT PERMISSIBLE.
GD1=21.2
GD3=1.0
110 — GD5=1.0
GD6=90.
GD7=48.2
GD8=21.2
GD9=1.0
GD10=1.0
GD12=38.8
C ** N1 AND N6 MUST BE OF A COMBINATION THAT J IS UNCHANGED
N1=10
N6=25
120 -C — ***** END OF DEE DATA CHANGES *****

121 -C\$\$\$\$\$\$\$\$ NO INPUT DATA REQUIRED BEYOND THIS POINT IN LISTING \$\$\$\$\$\$\$\$
 C **C1,C2,C3,C10 AND C11 DO NOT CHANGE TO OBTAIN THE DEE GEOMETRY.

C4=(GD2+GD8+GD9)*S
 C5=(GD1+GD2+GD3)*S
 C6=GD6*S
 C7=GD7*S
 C8=GD8*S
 C9=GD1*S

130 — C12=(GD4+GD10)*S
 C13=(GD4+GD5)*S
 R1=GD12*S

AN1=N1
 AN6=N6
 K2=K1-(N1+1)
 AK2=K2
 T1=PI*AK2/AN6
 R2=R1+C8+.5*(C9-C8)
 R4=R1+.5*(C4+C5)
 R6=R1+.5*(C4+C12+C5+C13)
 R02=.5*(C9-C8)
 R04=.5*(C5-C4)
 R06=R04-.5*(C12-C13)
 GOT09

140 —

C **END OF DATA MODIFICATIONS FOR DEE SHAPED COIL *****

99 CONTINUE

C1=COILS
 PI=3.1415927
 T=2.*PI/C1

150 — C2=G11*S
 C3=WEDGE*.5*C2/TAN(T/2.)

C4=(G2+G8+G9)*S
 C5=(G1+G2+G3)*S
 C6=G6*S
 C7=G7*S
 C8=G8*S
 C9=G1*S
 C10=G2*S
 C11=G4*S

160 — C12=(G4+G10)*S
 C13=(G4+G5)*S

```

161—— R1=G12*S
        R2=R1+C8+.5*(C9-C8)
        R4=R1+.5*(C4+C5)
        R6=R1+.5*(C4+C12+C5+C13)
        R02=.5*(C9-C8)
        R04=.5*(C5-C4)
        R06=R04-.5*(C12-C13)
        AN1=N1
        AN2=N2
170—— AN3=N3
        AN4=N4
        AN5=N5
        AN6=N6
        J=2*N1+N6+1
        L=N2+1+N3+1+N4+1
        NN5=N5+1
        IF(GEOM.GT.1.0)GOTO100
        WRITE(6,55)ISUP
100 CONTINUE
180—— DO 50 K1=1,J
        IF((K1.LT.MINK1).OR.(K1.GT.MAXK1))GOTO50
        AK1=K1
        IF(K1.GT.(N1+1)) GOTO 7
        IF(C7.EQ.0.)GOTO101
        Z=C7*(AK1-1.)/AN1
        GOTO 102
101 Z=0.
102 T1=0.
        GOTO 2
190—— 1 K3=K1-(N1+1)-(N6-1)
        AK3=K3
        IF(C7.EQ.0.)GOTO103
        Z=C7*(1.-(AK3-1.)/AN1)
        GOTO 104
103 Z=0.
104 T1=PI
        2 CONTINUE
        P1=C6+R1*COS(T1)
        P2=C6+R02+R2*COS(T1)
200—— P3=C6+R02+(R2+C10)*COS(T1)

```

```

201 — P4=C6+R04+R4*COS(T1)
      P5=C6+R04+(R4+C11)*COS(T1)
      P6=C6+R06+R6*COS(T1)
      ITEST=1
      N=0
      3 N=N+1
      IF(N.GT.L)GOTO50
      M=N
      IF((M.LT.MINM).OR.(M.GT.MAXM))GOTO50
210 — AM=M
      IF(M.GT.(N2+1)) GOTO 4
      Y=P1+(P2-P1)*(AM-1.)/AN2
      GO TO 28
      4 IF(M.GT.(N2+1+N3+1)) GOTO 5
      Y=P3+(P4-P3)*(AM-(AN2+1.+1.))/AN3
      GO TO 28
      5 Y=P5+(P6-P5)*(AM-(1.+AN2+1.+AN3+1.))/AN4
      GOTO28
      7 IF(K1.GT.(N1+N6)) GOTO 1
220 — K2=K1-(N1+1)
      AK2=K2
      T1=PI*AK2/AN6
      IF(T1.LE.PI/2..OR.DEE.EQ.0.0)GOTO9
      GOTO98
      9 Q1=C7+R1*SIN(T1)
      Q2=C7+R2*SIN(T1)
      Q3=C7+(R2+C10)*SIN(T1)
      Q4=C7+R4*SIN(T1)
      Q5=C7+(R4+C11)*SIN(T1)
230 — Q6=C7+R6*SIN(T1)
      P1=C6+R1*COS(T1)
      P2=C6+R02+R2*COS(T1)
      P3=C6+R02+(R2+C10)*COS(T1)
      P4=C6+R04+R4*COS(T1)
      P5=C6+R04+(R4+C11)*COS(T1)
      P6=C6+R06+R6*COS(T1)
      ITEST=2
      N=0
      8 N=N+1
240 — IF(N.GT.L)GOTO50

```

```

241—— M=N
      IF((M.LT.MINM).OR.(M.GT.MAXM))GOTO50
      AM=M
      IF(M.GT.(N2+1)) GOTO 11
      Z=Q1+(Q2-Q1)*(AM-1.)/AN2
      Y=P1+(P2-P1)*(AM-1.)/AN2
      GOTO 28
11 IF(M.GT.(N2+1+N3+1)) GOTO 12
      Z=Q3+(Q4-Q3)*(AM-(AN2+2.))/AN3
250—— Y=P3+(P4-P3)*(AM-(AN2+2.))/AN3
      GOTO 28
12 Y=P5+(P6-P5)*(AM-(AN2+1.+AN3+1.+1.))/AN4
      Z=Q5+(Q6-Q5)*(AM-(AN2+AN3+3.))/AN4
      GO TO 28
28 TH=C2
      TH=TH-DELTH*S
      AA=0.
      IF(Y.GT.C3) GOTO 29
      TH=2.*Y*TAN(T/2.)
260—— TH=TH-DELTH*S
      AA=1.
29 CONTINUE
      KK=NN5*(M-1)+NN5*L*(K1-1)
      DO 42 K=1,NN5
      IF((K.LT.MINK).OR.(K.GT.MAXK))GOTO42
      IF(K.GT.(NN5+1)/2.AND.04SYM.EQ.1.0)GOTO42
      AK=K
      I=K+KK
      JI=I+INK
270—— YY(I)=Y
      ZZ(I)=Z
      XX(I)=TH*(.5*AN5+1.-AK)/AN5
      XX(I)=XX(I)+DELTA*S
      THXY(I)=-AA*T*(180./PI)*(1.5*AN5+1.-AK)/AN5
      THYZ=0.
      THXZ=0.
      IF(ANSYS.EQ.0.0.AND.WECAN.EQ.1.0)GOTO41
      IF(LOCAL.EQ.1)THYZ=T1*(180./PI)
40 WRITE(6,43)JI, XX(I), YY(I), ZZ(I), THXY(I),THYZ,THXZ
280—— GOTO42

```

```

281 — 41 WRITE(6,44)JI,XX(I),YY(I),ZZ(I)
      42 CONTINUE
      43 FORMAT(I6,6X,5F12.5,F8.5)
      44 FORMAT(I6,18X,3F12.5)
      GOTO(3,8)ITEST
      50 CONTINUE
      IF(GEOM.GT.1.0)GOTO52
      IF(ANSYS.EQ.0.0.AND.WECAN.EQ.1.0)GOTO51
      WRITE(6,55)IEND
290 —  GOTO52
      51 WRITE(6,54)
      52 CONTINUE
      54 FORMAT(18HEND OF WECAN NODES)
      55 FORMAT(I6)
295 —  STOP

```

APPENDIX B

ELEMENT - PROGRAM LISTING

WITH

INPUT AND PARTIAL OUTPUT

```

1-WMTC2,T10,P6000,USCRAS:
  ACCOUNT(AW38029)
  ATTACHR(A,HLFTNX69)
  ATTACHR(B,HLFCLX69)
  LIBRARY(A,B,*)
  FTN(OPT=1,R=3)
  LGO.
  UPDATE(N,D)
  PURGE(,WMTC15E,PW=CONTROL)
10-CATALOG(NEWPL,WMTC15E,DS=$TFTR FINITE ELEMENTS $,ID=SCRAS)
  REWIND(TAPE6)
  COPYSR(TAPE6,OUTPUT)
  ✓
    PROGRAM ELEMENT (INPUT,TAPE5=INPUT,TAPE6,OUTPUT)
  C   *** A PROGRAM TO GENERATE FINITE ELEMENT DEFINITION CARDS FOR   **
  C   *** TOROIDAL FIELD COILS IN FUSION DEVICES LIKE A TOKAMAK.THE   **
  C   *** OUTPUT FORMAT IS FOR THE                                     **
  C   *** E1 ANSYS ELEMENT DEFINITION CARD                             **
  C   *** OR THE                                                         **
20-C- *** WEKAN 3.1 ELEMENT DEFINITION CARD                             **
  C   *** THE INPUT DATA IS A GENERALIZED FIRST LAYER OF ELEMENTS.   **
  C   *** WRITTEN BY R.A.SMITH,FEB.1975                                 **
  C
  C   TFTR-COMPUTATION 15
  C
  C   *** DEFINITIONS ***
  C
  C   WEKAN=KEY FOR THIS FORMAT. A 1.0 AFFIRMS, AND A 0.0 DENIES
  C   ANSYS=KEY FOR THIS FORMAT. A 1.0 AFFIRMS, AND A 0.0 DENIES
30-C- LAYER=TOTAL LAYERS OF FINITE ELEMENTS
  C   K= THE COUNTER OF LAYERS
  C   NELC=THE NUMBER OF INPUT ELEMENT CARDS
  C   J= THE COUNTER FOR INPUT/OUTPUT CARDS PER LAYER
  C   I= THE NODE POINT COUNTER ON THE ELEMENTS
  C   NODE(K,J,I)= THE NODE POINT NUMBERS
  C   MAT(K,J)= THE MATERIAL NUMBER (FOR PROPERTIES) OF EACH LAYER
  C   ITYPE(K,J)= THE MATERIAL TYPE
  C   ITABLE(K,J)= THE MATERIAL TABLE CONSTANT NUMBER
  C   INUM(K,J)=THE NUMBER OF ELEMENTS GENERATED BY THAT CARD
40-C- NINC(K,J)=THE NODE INCREMENTATION FOR ELEMENT GENERATION

```

```

41 -C — NONC=THE NODE INCREMENT FROM LAYER TO LAYER
C      NUMELF=THE FIRST ELEMENT NUMBER GENERATED BY THAT CARD
C      NUMELL=THE LAST ELEMENT NUMBER GENERATED BY THAT CARD
C      ***THESE ELEMENT NUMBERS ARE PRINTED BEYOND COLUMN 90,BECAUSE      **
C      ***THEY ARE NOT RECOGNIZED AS INPUT TO ANSYS OR WECAN,AND ALSO      **
C      ***THE CREATED CARD ID NUMBERS OCCUPY COLUMNS 81 TO 90 FOR UPDATE **
C      ISUP=AN ANSYS OR WECAN PRINTOUT SUPPRESSOR
C      DIMENSION NODE(45,48,8),MAT(45,48),ITYPE(45,48),ITABLE(45,48),
1      INUM(45,48),NINC(45,48)
50 ——— ANSYS=1.0
        WECAN=0.0
        NELC=28
        LAYER=45
        NONC=105
        NUMELL=0
        ISUP=+99999
        L=(LAYER+1)/2
1      FORMAT(I6)
2      FORMAT(12I6,I3)
60 ——— 3      FORMAT(12I6,I3,15X,2I5,I3)
        4      FORMAT(21HEND OF WECAN ELEMENTS)
        IF(WECAN.EQ.1.0.AND.ANSYS.EQ.0.0)GOTO5
        READ(5,2) (((NODE(K,J,I),I=1,8),MAT(K,J),ITYPE(K,J),ITABLE(K,J),
1      INUM(K,J),NINC(K,J)),J=1,NELC),K=1,1)
        GOTO6
2      READ(5,2) (((NODE(K,J,I),I=1,8),ITYPE(K,J),MAT(K,J),ITABLE(K,J),
1      INUM(K,J),NINC(K,J)),J=1,NELC),K=1,1)
6      CONTINUE
        WRITE(6,1)ISUP
70 ——— DO10 K=1,LAYER
        DO9 J=1,NELC
        IF((J.LT.10).AND.(K.LT.18.OR.K.GT.22))GOTO9
        IF((J.EQ.2.OR.J.EQ.5.OR.J.EQ.8).AND.(K.NE.20))GOTO9
        DO7 I=1,8
7      NODE(K,J,I)=NODE(1,J,I)+(K-1)*NONC
        ITYPE(K,J)=ITYPE(1,J)
        ITABLE(K,J)=ITABLE(1,J)
        NINC(K,J)=NINC(1,J)
        MAT(K,J)=MAT(1,J)
80 ——— INUM(1,23)=2

```



```

81 — INUM(1,28)=2
      INUM(K,J)=INUM(1,J)
      IF((J.EQ.16.OR.J.EQ.21.OR.J.EQ.26).AND.(K.LE.23))
1MAT(K,J)=MAT(K,J)+K-1
      IF((J.EQ.16.OR.J.EQ.21.OR.J.EQ.26).AND.(K.GT.23))
1MAT(K,J)=MAT(L,J)-(K-L)
      IF((K.GE.33.AND.K.LE.38).AND.(J.EQ.10.OR.J.EQ.13.OR.J.EQ.18.OR.
1J.EQ.23.OR.J.EQ.28)) INUM(K,J)=INUM(K,J)+2
      IF(((K.LE.2).OR.(K.GE.6.AND.K.LE.8)).AND.(J.EQ.23.OR.J.EQ.28))
90 — 1INUM(K,J)=INUM(K,J)+2
      NUMELL=NUMELL+INUM(K,J)
      NUMELF=NUMELL+1-INUM(K,J)
      IF(WECAN.EQ.1.0.AND.ANSYS.EQ.0.0)GOTO8
      WRITE(6,3) (NODE(K,J,I),I=1,8),MAT(K,J),ITYPE(K,J),ITABLE(K,J),
1      INUM(K,J),NINC(K,J),NUMELF,NUMELL,K
      GOTO9
8 WRITE(6,3) (NODE(K,J,I),I=1,8),ITYPE(K,J),MAT(K,J),ITABLE(K,J),
1      INUM(K,J),NINC(K,J),NUMELF,NUMELL,K
9 CONTINUE
100 — 10 CONTINUE
      IF(WECAN.EQ.1.0.AND.ANSYS.EQ.0.0)GOTO15
      IEND=-1
      WRITE(6,1)IEND
      GOTO20
15 WRITE(6,4)
20 CONTINUE
      STOP
      END

```

110	5015	5022	5023	5016	5120	5127	5128	5121	1	1	2	7
	5029	5036	5037	5030	5134	5141	5142	5135	1	1	6	7
	5071	5078	5079	5072	5176	5183	5184	5177	1	1	2	7
	5016	5023	5024	5017	5121	5128	5129	5122	1	1	2	7
	5030	5037	5038	5031	5135	5142	5143	5136	1	1	6	7
	5072	5079	5080	5073	5177	5184	5185	5178	1	1	2	7
	5017	5024	22	15	5122	5129	127	120	1	1	2	7
	5031	5038	36	29	5136	5143	141	134	1	1	6	7
	5073	5080	78	71	5178	5185	183	176	1	1	2	7
	15	22	23	16	120	127	128	121	1	1	10	7
120	16	23	24	17	121	128	129	122	1	1	2	7

121	30	37	38	31	135	142	143	136	2	2	6	7
	72	79	80	73	177	184	185	178	1	1	2	7
	17	24	25	18	122	129	130	123	1	1	2	7
	31	38	39	32	136	143	144	137	2	2	1	1
	38	45	46	39	143	150	151	144	3	3	4	7
	66	73	74	67	171	178	179	172	2	2	1	1
	73	80	81	74	178	185	186	179	1	1	2	7
	18	25	26	19	123	130	131	124	1	1	2	7
	32	39	40	33	137	144	145	138	2	2	1	1
130	39	46	47	40	144	151	152	145	3	3	4	7
	67	74	75	68	172	179	180	173	2	2	1	1
	74	81	82	75	179	186	187	180	1	1	2	7
	19	26	27	20	124	131	132	125	1	1	2	7
	33	40	41	34	138	145	146	139	2	2	1	1
	40	47	48	41	145	152	153	146	3	3	4	7
	68	75	76	69	173	180	181	174	2	2	1	1
	75	82	83	76	180	187	188	181	1	1	2	7

✓
 *DECK,ELEM
 140- *REWIND,TAPE6
 *READ,TAPE6
 ✓

PARTIAL OUTPUT

99999

15	22	23	16	120	127	128	121	1	1	-0	10	7	1	10	1
16	23	24	17	121	128	129	122	1	1	-0	2	7	11	12	1
30	37	38	31	135	142	143	136	2	2	-0	6	7	13	18	1
72	79	80	73	177	184	185	178	1	1	-0	2	7	19	20	1
17	24	25	18	122	129	130	123	1	1	-0	2	7	21	22	1
31	38	39	32	136	143	144	137	2	2	-0	1	1	23	23	1
38	45	46	39	143	150	151	144	3	3	-0	4	7	24	27	1
66	73	74	67	171	178	179	172	2	2	-0	1	1	28	28	1
73	80	81	74	178	185	186	179	1	1	-0	2	7	29	30	1
18	25	26	19	123	130	131	124	1	1	-0	2	7	31	32	1
32	39	40	33	137	144	145	138	2	2	-0	1	1	33	33	1
39	46	47	40	144	151	152	145	3	3	-0	4	7	34	37	1
67	74	75	68	172	179	180	173	2	2	-0	1	1	38	38	1
74	81	82	75	179	186	187	180	1	1	-0	4	7	39	42	1
19	26	27	20	124	131	132	125	1	1	-0	2	7	43	44	1
33	40	41	34	138	145	146	139	2	2	-0	1	1	45	45	1
40	47	48	41	145	152	153	146	3	3	-0	4	7	46	49	1
68	75	76	69	173	180	181	174	2	2	-0	1	1	50	50	1
75	82	83	76	180	187	188	181	1	1	-0	4	7	51	54	1
120	127	128	121	225	232	233	226	1	1	-0	10	7	55	64	2
121	128	129	122	226	233	234	227	1	1	-0	2	7	65	66	2
135	142	143	136	240	247	248	241	2	2	-0	6	7	67	72	2
177	184	185	178	282	289	290	283	1	1	-0	2	7	73	74	2
122	129	130	123	227	234	235	228	1	1	-0	2	7	75	76	2
136	143	144	137	241	248	249	242	2	2	-0	1	1	77	77	2
143	150	151	144	248	255	256	249	4	3	-0	4	7	78	81	2
171	178	179	172	276	283	284	277	2	2	-0	1	1	82	82	2
178	185	186	179	283	290	291	284	1	1	-0	2	7	83	84	2
123	130	131	124	228	235	236	229	1	1	-0	2	7	85	86	2
137	144	145	138	242	249	250	243	2	2	-0	1	1	87	87	2
144	151	152	145	249	256	257	250	4	3	-0	4	7	88	91	2
172	179	180	173	277	284	285	278	2	2	-0	1	1	92	92	2
179	186	187	180	284	291	292	285	1	1	-0	4	7	93	96	2
124	131	132	125	229	236	237	230	1	1	-0	2	7	97	98	2
138	145	146	139	243	250	251	244	2	2	-0	1	1	99	99	2
145	152	153	146	250	257	258	251	4	3	-0	4	7	100	103	2
173	180	181	174	278	285	286	279	2	2	-0	1	1	104	104	2
180	187	188	181	285	292	293	286	1	1	-0	4	7	105	108	2
225	232	233	226	330	337	338	331	1	1	-0	10	7	109	118	3
226	233	234	227	331	338	339	332	1	1	-0	2	7	119	120	3
240	247	248	241	345	352	353	346	2	2	-0	6	7	121	126	3
282	289	290	283	387	394	395	388	1	1	-0	2	7	127	128	3
227	234	235	228	332	339	340	333	1	1	-0	2	7	129	130	3
241	248	249	242	346	353	354	347	2	2	-0	1	1	131	131	3
248	255	256	249	353	360	361	354	5	3	-0	4	7	132	135	3
276	283	284	277	381	388	389	382	2	2	-0	1	1	136	136	3
283	290	291	284	388	395	396	389	1	1	-0	2	7	137	138	3
228	235	236	229	333	340	341	334	1	1	-0	2	7	139	140	3
242	249	250	243	347	354	355	348	2	2	-0	1	1	141	141	3
249	256	257	250	354	361	362	355	5	3	-0	4	7	142	145	3
277	284	285	278	382	389	390	383	2	2	-0	1	1	146	146	3
284	291	292	285	389	396	397	390	1	1	-0	2	7	147	148	3
229	236	237	230	334	341	342	335	1	1	-0	2	7	149	150	3
243	250	251	244	348	355	356	349	2	2	-0	1	1	151	151	3
250	257	258	251	355	362	363	356	5	3	-0	4	7	152	155	3
278	285	286	279	383	390	391	384	2	2	-0	1	1	156	156	3
245	292	293	285	390	397	398	391	1	1	-0	2	7	157	158	3