30 7/19/86 . WB (2) (7) I-26958 DR-1812-0

6-18-86

SANDIA REPORT

SAND86—0910 • Unlimited Release • UC—32
Printed May 1986

The GENESIS Finite Element Mesh File Format

L. M. Taylor, D. P. Flanagan, W. C. Mills-Curran

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

DISCLAIMER

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A02
Microfiche copy: A01

SAND--86-0910

DE86 012740

Unlimited Release
UC-32
Printed May 1986

THE GENESIS FINITE ELEMENT MESH FILE FORMAT*

L. M. Taylor
D. P. Flanagan
W. C. Mills-Curran
Engineering Analysis Department
Sandia National Laboratories
Albuquerque, New Mexico

ABSTRACT

A new finite element mesh file format which provides a neutral interface from a mesh generation program to a finite element analysis program is described. The file format has been constructed to allow for multiple element types and flexible definition of boundary conditions. It provides for arbitrary element connectivity and attributes, so that continuum and structural elements can easily be accommodated in the same model. Element side boundary conditions as well as nodal point boundary conditions are supported. The file format is applicable to finite element models in any n-dimensional space. Both bandwidth and wavefront optimization schemes can be accommodated with the GENESIS file format.

*This work performed at Sandia National Laboratories is supported by the U.S. Department of Energy under Contract Number DE-AC04-76DP00789.

CONTENTS

1.0 INTRODUCTION

The Engineering Analysis Department has used the so-called TAPE9 file output from QMESH [1] as a standard finite element mesh file format for a number of years. Most of the department's finite element programs, such as JAC, SANCHO and HONDO, read this file format. Several translators have been written to convert a PATRAN neutral file to the TAPE9 format. While the versatility of the format is very limited, TAPE9 has served very successfully in integrating numerous mesh generation and analysis programs.

The central design goal of GENESIS is to improve the flexibility of the TAPE9 file format while retaining its simplicity. GENESIS integrates general purpose mesh generation programs with a wide variety of analysis programs throughout the Engineering Science Directorate. It embraces linear and non-linear analysis, implicit and explicit solution schemes, and both fluid and solid mechanics.

The GENESIS file format was developed to extend the capabilities of the TAPE9 format to include multiple element types and versatile boundary conditions. GENESIS allows elements to be defined with any number of connected nodes and any number of supplementary geometric attributes (for example, cross section properties of structural members).

Boundary conditions are specified in the TAPE9 file format via an awkward list of nodal flags and associated nodes GENESIS introduces several improvements in this area. First, the flag and node lists have been separated. Arrays of pointers and counters which delimit the nodes associated with each flag have been introduced. Second, a list of scale factors now accompanies the node list. This feature allows for non-uniformly distributed loads. Third, element side conditions are supported. This enhancement supports applications where the relationship of nodes to element sides is essential to the interpretation of the boundary condition (such as external pressure, contact surfaces, and heat flux).

Most of the characteristics of the TAPE9 file format are retained in GENESIS. The sequential "neutral file" format as opposed to a random "data base" is retained in order to keep the structure of GENESIS simple and easy to use for applications programmers. GENESIS contains unformatted (binary) data for efficiency.

Because the GENESIS file is unformatted, it is not easily transported between dissimilar computer systems. It is intended that mesh generation (or translator) programs which produce GENESIS should be supported on all appropriate systems so that the formatted (ASCII) input to these programs would be transported, rather than the GENESIS file itself.

GENESIS defines only the geometry of a finite element model. The association of finite element and material models with each element block, the definition of material properties, and the description of boundary and loading conditions is left to the analysis program. It is felt that these aspects of a finite element model are more efficiently defined within an analysis code than within a general purpose mesh generator. To incorporate these capabilities into GENESIS would be impractical due the wide spectrum of analysis codes which must be supported.

2.0 FILE FORMAT DESCRIPTION

The following sections prescribe the GENESIS file format. The GENESIS file format is structured so that the length of all records is defined before the record is read. This further implies that the reader can dynamically allocate memory for all arrays in GENESIS before reading them.

2.1 HEADING

The first record of the GENESIS file is an 80-character string in which to place a title or heading. This string is simply a comment.

2.2 PROBLEM SIZING PARAMETERS

The second record of GENESIS contains problem sizing parameters. These nine integers define the size of most arrays contained within this GENESIS file. The sizing parameters are read as a single record in the following order:

1) NUMNOD = Number of Nodes
2) NDIM = Number of Coordinates per node (number of spatial dimensions)
3) NUMEL = Total Number of Elements (includes all element types)
4) NELBLK = Number of Element Blocks
5) NUMNPS = Number of Nodal Point Sets
6) LNPSNL = Length of the Nodal Point Sets Node List
7) NUMESS = Number of Element Side Sets
8) LESSEL = Length of the Element Side Sets Element List
9) LESSNL = Length of the Element Side Sets Node List

2.3 NODAL POINT COORDINATES

The third record of GENESIS contains the nodal point coordinates. All coordinates are contained in a single record with the node number (1:NUMNOD) cycling faster than the component (X,Y,Z) index (1:NDIM). NUMNOD and NDIM are defined in Section 2.2.

2.4 OPTIMIZED ELEMENT ORDER MAP

The fourth record of GENESIS contains a gather/scatter map of length NUMEL (Section 2.2) which defines an element order with minimal wavefront width. This record can be skipped with a null read if wavefront optimization is not relevant to the reader. The map (MAP) relates the order in which a wavefront solver should process the elements. For example, MAP(3) gives the number of the third element to be processed by a wavefront solver. All element numbers must appear in this map exactly once.

2.5 ELEMENT BLOCKS

GENESIS stores element data in element blocks. All elements within an element block must share a common material type as well as a common finite element type. This eliminates the need for the material ID array which appeared on the TAPE9 file format.

The number of an element is implicitly defined by the order in which it appears in the GENESIS file. Elements are numbered consecutively across all element blocks.

The following subsections describe each of the three records which GENESIS contains for each element block. GENESIS contains a total of 3*NELBLK (Section 2.2) records of element block data.

2.5.1 ELEMENT BLOCK SIZING PARAMETERS

The first record in GENESIS for each element block contains a block ID and three parameters which define the size of all arrays in this block. These integers appear in the following order:

1) IDEBLK = ID for this Element Block
2) NELNOD = Number of Element Connectivity Nodes for this element type
3) NUMELB = Number of Elements in this Block
4) NATRIB = Number of Element Attributes required for this element type

Each element block must have a unique ID which is an arbitrary integer label assigned by the user via the mesh generation program. This ID allows the user to specify a group of elements to the analysis code without having to know the order in which element blocks are stored in the GENESIS file. The input data to the analysis code must associate each element block with a finite element model and a material model.

2.5.2 ELEMENT BLOCK CONNECTIVITY

The second record in a GENESIS element block contains element connectivity. The node index (1:NELNOD) for each element cycles faster then the element index (1:NUMELB) for this record. NELNOD and NUMELB are defined in the previous section.

GENESIS does not impose any rules on connectivity beyond the number of nodes per element. For example, GENESIS does not distinguish between a four-node quad and a four-node triangle (with a center node). This implies that the mesh generator and analysis code must agree on conventions involving connectivity (for example, a counterclockwise order).

2.5.3 ELEMENT BLOCK ATTRIBUTES

The third and final record in a GENESIS element block contains geometric attributes which are required to complete the specification of this element type. The attribute index (1:NATRIB) for each element cycles faster than the element index for this record. NATRIB is the fourth parameter defined in Section 2.5.1.

This feature is included mainly for structural elements which have pertinent cross section properties. This record may be null for many element types. Similar to connectivity, element attributes require conventions between the mesh generator and analysis programs which are not specified by the GENESIS standard.

2.6 NODAL POINT SETS

Nodal point sets are the first option which GENESIS provides to implement constraint or load conditions. A nodal point set refers to a set of nodes which have been associated with an unique ID. Each node contained in a given node set is registered in that node set's list and is assigned an individual distribution factor. A particular node may appear in any number of node sets, but GENESIS requires that all nodes within a given node set must be unique.

The following subsections describe the five records in GENESIS which contain all nodal point set data.

2.6.1 NODAL POINT SET IDS

The first nodal point set data record in GENESIS contains the ID of each node set. The node set ID is the mechanism whereby the user directs the analysis code to apply a certain constraint or load condition to a set of nodes. Akin to the element block ID (Section 2.5.1), the node set ID is an unique integer label assigned by the user.

The length of this record is NUMNPS, which is the fifth parameter defined in Section 2.2.

2.6.2 NODAL POINT SET COUNTS

The second nodal point set data record in GENESIS contains the number of nodes in each node set. The length of this record is also NUMNPS

2.6.3 NODAL POINT SET POINTERS

The third nodal point set data record in GENESIS contains a pointer for each node set which locates the nodes associated with this node set relative to a concatenated list. The length of this record is also NUMNPS.

2.6.4 NODAL POINT SET NODE LIST

The fourth nodal point set data record in GENESIS contains the lists of nodes for all sets concatenated into a single record. The data on the previous two records is used to identify that portion of this record which defines the nodes in a particular nodal point set.

The length of this record is LNPSNL, which is the sixth parameter defined in Section 2.2.

2.6.5 NODAL POINT SET DISTRIBUTION FACTORS

The fifth and final nodal point set data record in GENESIS contains a concatenated list of nodal distribution factors. This list has a one-to-one correspondence with the node set list described in the previous section. These multiplication factors can be used to prescribe the spatial distribution of a boundary condition. The length of this record is also LNPSNL.

2.7 ELEMENT SIDE SETS

Element side sets are the second option which GENESIS provides to implement constraint or load conditions. This feature is important to certain surface conditions, such as applied pressure and heat flux, where the connection of nodes to element sides is important. The element number of each side contained in a given side set is registered in an element list. All nodes connected to that side are recorded in a separate list. A particular element side may appear in any number of side sets, but GENESIS requires that all sides within a given side set must be unique. However, the nodes associated with a side set may (and generally will) appear multiple times within that set's node list. Duplicate element numbers may also appear within a side set if more than one side of a given element is part of that set.

The following subsections describe the eight records in GENESIS which contain all element side set data.

2.7.1 ELEMENT SIDE SET IDS

The first element side set data record in GENESIS contains the ID of each side set. The side set ID is the mechanism whereby the user directs the analysis code to apply a certain constraint or load condition to a set of element sides. As will all GENESIS IDs, the side set ID is an unique integer label assigned by the user.

The length of this record is NUMESS, which is the seventh parameter defined in Section 2.2.

2.7.2 ELEMENT SIDE SET ELEMENT COUNTS

The second element side set data record in GENESIS contains the number of element sides in each side set. Note that each element side is associated with exactly one element. The length of this record is also NUMESS.

2.7.3 ELEMENT SIDE SET NODE COUNTS

The third element side set data record in GENESIS contains the total number of element side nodes in each side set. The number of nodes per element side is implied by each element's type. The length of this record is also NUMESS.

2.7.4 ELEMENT SIDE SET ELEMENT POINTERS

The fourth element side set data record in GENESIS contains a pointer for each side set which locates the elements associated with this side set relative to a concatenated list. The length of this record is also NUMESS.

2.7.5 ELEMENT SIDE SET NODE POINTERS

The fifth element side set data record in GENESIS contains a pointer for each side set which locates the nodes associated with this side set relative to a concatenated list. The length of this record is also NUMESS.

2.7.6 ELEMENT SIDE SET ELEMENT LIST

The sixth element side set data record in GENESIS contains a concatenated list of elements which encompasses all side sets. The element number for each side is provided in order that the analysis code can determine material properties which may be relevant to a particular surface condition. The count and pointer for each side set relative to this list are defined in Sections 2.7.2 and 2.7.4, respectively.

The length of this record is LESSEL, which is the eighth parameter defined in Section 2.2.

2.7.7 ELEMENT SIDE SET NODE LIST

The seventh element side set data record in GENESIS contains a concatenated list of side nodes which encompasses all side sets. This list is stored in GENESIS such that the local node index cycles faster than the element side index (all connected nodes for side 1, then all connected nodes for side 2, etc.). It is strongly recommended that a given element side set include only one element type so the set's node list can be treated as a rectangular table. This list is likely to contain repeated node numbers since associated element sides tend to be connected. The count and pointer for each side set relative to this list are defined in Sections 2.7.3 and 2.7.5, respectively.

The length of this record is LESSNL, which is the ninth parameter defined in Section 2.2.

2.7.8 ELEMENT SIDE SET DISTRIBUTION FACTORS

The eighth and final element side set data record in GENESIS contains a concatenated list of nodal distribution factors. This list has a one-to-one correspondence with the node set list described in the previous section. These multiplication factors can be used to prescribe the spatial distribution of a boundary condition. The distribution factors for repeated nodes within a given element side set would normally be identical, but GENESIS does not require this restriction. The length of this record is also LESSNL.

REFERENCES

1. R. E. Jones, "Users Manual for QMESH - a Self-Organizing MESH Generation Program," SLA-74-0239, July 1974.

APPENDIX A GENESIS FILE FORMAT IN FORTRAN

The GENESIS file format is specified in the FORTRAN code fragment shown below. This is meant to serve as a quick reference guide for application programmers. The appropriate section of text is referenced below for each GENESIS record.

```
C Heading                                           (Section 2.1)    -
      CHARACTER*80 HEAD
      READ(9) HEAD
C
C Problem Sizing Parameters                         (Section 2.2)    -
      READ(9)NUMNOD,NDIM,NUMEL,NELBLK,NUMNPS,LNPSNL,NUMESS,LESSEL,LESSNL
C        NUMNOD   - Number of Nodes
C        NDIM     - Number of Coordinates per Node
C        NUMEL    - Total Number of Elements
C        NELBLK   - Number of Element Blocks
C        NUMNPS   - Number of Nodal Point Sets
C        LNPSNL   - Length of the Nodal Point Sets Node List
C        NUMESS   - Number of Side Sets
C        LESSEL   - Length of the Element Side Sets Element List
C        LESSNL   - Length of the Element Side Sets Node List
C
C Nodal Point Coordinates                           (Section 2.3)    -
      READ(9) ((COORD(I,J),I=1,NUMNOD),J=1,NDIM)
C
C Optimized Element Order Map                       (Section 2.4)    -
      READ(9) (MAP(I),I=1,NUMEL)
```

```
C
C Element Blocks
C
      DO 10 K=1,NELBLK
C
C Sizing Parameters for this Element Block              (Section 2.5.1) -
         READ(9) IDEBLK,NUMELB,NELNOD,NATRIB
C          IDEBLK - Element block identification (must be unique)
C          NUMELB - Number of elements in this block (the sum of NUMELB
C                   over all blocks must equal NUMEL above)
C          NELNOD - Number of nodes defining the connectivity for an
C                   element of this type
C          NATRIB - Number of element attributes for this element type
C
C Connectivity for this Element Block                   (Section 2.5.2) -
         READ(9) ((ICONK(J,I),J=1,NELNOD),I=1,NUMELB)
C
C Attributes for this Element Block                     (Section 2.5.3) -
         READ(9) ((ATRIBK(J,I),J=1,NATRIB),I=1,NUMELB)
   10 CONTINUE
C
C Nodal Point Sets Data
C
C Nodal Point Set IDs                                   (Section 2.6.1) -
      READ(9) (IDNPS(I),I=1,NUMNPS)
C Nodal Point Set Counts                                (Section 2.6.2) -
      READ(9) (NNNPS(I),I=1,NUMNPS)
C Nodal Point Set Pointers                              (Section 2.6.3) -
      READ(9) (IPTNPS(I),I=1,NUMNPS)
C Nodal Point Set Node List                             (Section 2.6.4) -
      READ(9) (LSTNPS(I),I=1,LNPSNL)
C Nodal Point Distribution Factors                      (Section 2.6.5) -
      READ(9) (FACNPS(I),I=1,LNPSNL)
```

```
C
C Element Side Sets Data
C
C Element Side Set IDs                              (Section 2.7.1) -
      READ(9) (IDESS(I),I=1,NUMESS)
C Element Side Set Element Counts                   (Section 2.7.2) -
      READ(9) (NEESS(I),I=1,NUMESS)
C Element Side Set Node Counts                      (Section 2.7.3) -
      READ(9) (NNESS(I),I=1,NUMESS)
C Element Side Set Element Pointers                 (Section 2.7.4) -
      READ(9) (IPEESS(I),I=1,NUMESS)
C Element Side Set Node Pointers                    (Section 2.7.5) -
      READ(9) (IPNESS(I),I=1,NUMESS)
C Element Side Set Element List                     (Section 2.7.6) -
      READ(9) (LTEESS(I),I=1,LESSEL)
C Element Side Set Node List                        (Section 2.7.7) -
      READ(9) (LTNESS(I),I=1,LESSNL)
C Element Side Set Distribution Factors             (Section 2.7.8) -
      READ(9) (FACESS(I),I=1,LESSNL)
C
      END
```

APPENDIX B HELPFUL HINTS

This appendix contains notes and remarks on particular sections of the GENESIS file format These comments are intended to aid programmers in adapting GENESIS to their application codes

B.1 NOTES ON SECTION 2.4 (Optimized Element Order Map)

The optimized element order map is required by the GENESIS file format to support an element numbering which is optimized for a wavefront solver. (These are popular in heat transfer and fluid dynamics codes.) Since GENESIS groups elements by element and material type, the order in which elements appear in the GENESIS file cannot always reflect a fully optimized order.

Some application programmers may find they require the inverse (IMAP) of the map (MAP) which appears in GENESIS. The inverse map can be generated via the following scatter operation:

```
      DO 10 I = 1 , NUMEL
         IMAP(MAP(I)) = I
   10 CONTINUE
```

The difference between MAP and IMAP is subtle. As an example assume that MAP(3) = 5, which says that the fifth element in GENESIS should be processed third by the wavefront solver. In this case, IMAP(5) = 3. Application programmers should carefully consider how to work this concept into the internal architecture of their code.

B.2 NOTES ON SECTION 2.5 (Element Blocks)

The element block structure of GENESIS represents the greatest architectural difference between GENESIS and the TAPE9 file format. The reason that element blocking was adopted is twofold. First, element blocks provide an efficient storage scheme for element types with vastly different connectivity and geometric attribute requirements (for example, a four-node quadrilateral continuum element versus a two-node structural bar). Second, element blocks encourage vector processing of elements.

Vector processing is inhibited by branching on an element-by-element basis. Elements must be grouped for processing by the same finite element and constitutive model routines. One can envision this concept as feeding the finite element data contiguously through the same finite element machine.

The GENESIS standard purposefully avoids defining protocol for individual element types, such as prescribing the order in which local nodes appear in the element's connectivity. This was done so that new element types can be accommodated without changing the GENESIS standard. An analysis code may require the end user to explicitly distinguish element types. For example, a "SHELL,10" input card may be required to tell the analysis code that the elements with block ID 10 are shells.

B.3 **NOTES ON SECTION 2.6** (Nodal Point Sets)

The Nodal Point Sets feature of GENESIS gives the user the ability to identify nodes of interest (for loads, constraints, etc.) with a flag. This flag can point to a single node or an entire group of nodes. Futhermore, any given node may belong to several different sets. With this capability the user does need not know the number of specific nodes, and the preprocessor is free to reorder the nodes to achieve an optimized bandwidth.

GENESIS does not impose any protocol on flags. For example, GENESIS does not differentiate between a displacement condition and a nodal point load set. This allows the range of capability for boundary conditions supported via GENESIS to be readily expanded. It also allows the same flag to be used in several different applications. For example, the same nodal point set may be used to define a temperature condition in one analysis, a zero displacement condition in a second analysis, and a nodal point force set in a third analysis.

B.4 NOTES ON SECTION 2.7 (Element Side Sets)

Element Side Sets are an extension of nodal point sets for boundary conditions which require the relationship of nodes to elements. For example, in order to apply a pressure it is necessary to know the area of the element side surface and on which side of the surface the element lies. Like nodal point sets, element side set flags provide a means for the end user to refer to a specific group of elements without knowing the numbers of the elements or the connected nodes. This feature allows the preprocessor to logically group and reorder elements without burdening the user.

Processing element side sets in application codes which support multiple element types requires special attention. Because the applications programmer may sort the elements read from GENESIS into different sets (e.g., shells, solids, beams), it may be tedious to locate an element by its "global" number (Section 2.5). In this case, an additional mapping array is easily defined which points from the "global" element number to the element type/set and to the "local" element number within that set. The following program segment illustrates the construction of this mapping array NELMAP.

```
C
C     Process shell elements.
C
      DO 100 ISHELL = 1, NSHELL
C
C        IPSHLL contains the global element number of each shell.  This
C        array was constructed as the elements were read from GENESIS.
C
         IEL = IPSHLL(ISHELL)
C
C        Identify this element as a shell in NELMAP with the flag ITSHEL.
C
         NELMAP(1,IEL) = ITSHEL
C
C        Store the "local" shell element number.
C
         NELMAP(2,IEL) = IEL
C
100   CONTINUE
```

Other element types would be processed in the same manner using different flags for NELMAP(1,IEL) to uniquely identify the type/group.

Distribution:

1510	J. W. Nunziato	
1511	D. K. Gartling	
1511	G. G. Weigand (5)	
1520	D. J. McCloskey	
1521	R. D. Krieg	
1522	R. C. Reuter	
1523	J. H. Biffle	
1523	D. P. Flanagan (20)	
1523	L. M. Taylor (20)	
1524	A. K. Miller	
1524	W. C. Mills-Curran (10)	
1530	L. W. Davison	
1540	W. C. Luth	
3141-1	S. A. Landenberger (5)	
3151	W. L. Garner (3)	
3154-1	C. H. Dalin (28)	
	(Unlimited Release for DOE/OSTI)	
8024	P. W. Dean	