

AD-A085 080

ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND)
DATA GENERATION COMPUTER PROGRAMS FOR SHELL FINITE ELEMENTS. (U)
JUL 79 S M FORD

F/6 13/13

UNCLASSIFIED

RAE-TM-STRUCTURES-954

DRIC-BR-70226

NL

[or]

AD-A085 080



END

DATE

FILED

6-80

DTIC

UNLIMITED

BR70226

TECH. MEMO
STRUCTURES 954

TECH. MEMO
STRUCTURES 954

②

①

SR

ADA 085080

ROYAL AIRCRAFT ESTABLISHMENT

LEVEL

DATA GENERATION COMPUTER PROGRAMS FOR SHELL FINITE ELEMENTS

by

Susan M. Ford

July 1979

DDC
RECEIVED
JAN 3 1980
RECEIVED
A

FILE COPY

UNLIMITED 79 12 27 194

18 DRIZ

19 BR-70226

ROYAL AIRCRAFT ESTABLISHMENT

9 Technical Memorandum Structures 954

12 55

Received for printing, 27 July 1979

6

DATA GENERATION COMPUTER PROGRAMS FOR SHELL FINITE ELEMENTS

by

10 Susan M. Ford

11 Jul 79

SUMMARY

A series of computer programs written in ICL 1900 series FORTRAN is presented to generate data for the finite element analysis of shells, with particular reference to spherical surfaces. The programs are appropriate to a version of the SEMILOOF element contained in an RAE Structures Department program and to TRIA3 and QUAD4 elements which are available in a NASTRAN package. Both descriptions and listings of the programs are given.

14 RAE-TM-STRUCTURES-954

Copyright ©
Controller HMSO London
1979

A

370450

LIST OF CONTENTS

	<u>Page</u>
1 INTRODUCTION	3
2 PROGRAM GENDATA	3
3 PATCH DATA GENERATION PROGRAMS	4
3.1 Programs GENPATCH and GENTRIPAT	4
3.2 Program GENIPAT	6
4 PROGRAM GENNASTBC	6
4.1 Subroutine PRESSURE	7
4.2 Subroutine KTHEPHI	8
4.3 Subroutine KTHETHE	9
4.4 Subroutine ETHEPHI	10
4.5 Subroutine ETHETHE	11
5 NASTRAN DATA GENERATION PROGRAMS	12
5.1 Program GENCQUAD4	12
5.2 Program GENCTRIA3	13
5.3 Program GENGRID	13
6 PROGRAM CHANGEDATA	14
7 CONCLUSIONS	14
Appendix A SEMILOOF computer program	15
List of principal symbols	31
References	32
Computer printout	Figure 1
Illustrations	Figures 2 - 4
Report documentation page	inside back cover

1 INTRODUCTION

The data input to finite element packages is often of a repetitive nature and in fixed format which makes manual data preparation both time-consuming and prone to error. The computer programs described and listed here form a less tedious and faster alternative to this manual stage of data preparation.

The programs are appropriate either to a finite element program containing the SEMILOOF thin shell element, held by Structures Department, RAE (see the Appendix) or for a NASTRAN package¹, containing the TRIA3 and QUAD4 facet shell elements, which are accessed via a commercial bureau. The majority of the programs generate data specifically for meshes on the surface of a sphere, and they have been used to generate all the data for the hemispherical cap and patch test problems considered by Morley and Morris².

Two types of program are described. One type alters a free format data file to one that can be incorporated directly into a data deck. The other type is for a particular mesh, *ie* a 'patch', in which only the essential variable parameters are input.

2 PROGRAM GENDATA

This program generates the x , y and z coordinates on the surface of a sphere of radius R when the corresponding θ and ϕ surface coordinates are given. The program can be used to create either SEMILOOF or NASTRAN QUAD4 data. In the latter case the GENDATA output is used as input to program GENGRID. For certain SEMILOOF meshes it is convenient to interpolate the midside node positions from vertex nodes. For this option, dummy values of θ and ϕ at the midside nodes are given initially and, when the calculation of the rest of the nodes is completed, the positions of these midside nodes are calculated. This is done by reading in the number of the midside node together with the nodes between which it lies and bisecting the geodesic line between the vertices. This is repeated until the end of the file is reached. Note that constant θ gives a "line of longitude" which is a geodesic line, but constant ϕ gives a "line of latitude" which is not a geodesic line.

The radius of the sphere, R , is set to 10.0 but this can be altered by changing a single line in the program.

The free format input data is as follows:-

NNODE		
NODE ₁	θ_1	ϕ_1
.	.	.
.	.	.
.	.	.
NODE _i	θ_i	ϕ_i
.	.	.
.	.	.
.	.	.
NODE nnode	θ_{nnode}	ϕ_{nnode}
NODE	NOA	NOB

where NNODE is the total number of vertex and midside nodes,
 θ_i and ϕ_i are the θ and ϕ coordinates in degrees of node $i, 1, 2, \dots, i, \dots, \text{NNODE}$,
 NODE is the midside node lying between NOA and NOB.

The output is to card punch CPO and is of the form NODE_i X_i Y_i Z_i
 where X, Y, Z are the x y z coordinates of NODE_i $1, 2, \dots, i, \dots, \text{NNODE}$.
 The nodes are input in sequential order and checked. If a node is found to be out of sequence a message is output to line printer LPO.

3 PATCH DATA GENERATION PROGRAMS

These programs generate the x, y and z coordinates, and boundary condition data for a patch of nine quadrilateral elements or 18 triangular elements on the surface of a sphere of radius $R = 10.0$, see Fig 4.

3.1 Programs GENPATCH and GENTRIPAT

Program GENPATCH calculates the data for a 3×3 regular mesh of SEMILOOF or NASTRAN QUAD4 elements as shown in Fig 2a&b respectively. Program GENTRIPAT calculates the data for a similar mesh of triangular SEMILOOF elements as in Fig 3a. The input for both of these programs is the same, *ie*

ITYPE ICASE THEINC PHIINC THEZERO PHIZERO

where ITYPE = 1 for SEMILOOF output,
 = 2 for NASTRAN QUAD4 output,

ICASE is a variable used in subsequent programs to specify the type of boundary displacement to be applied,

- = 0 for pressure loading,
- = 1 for $\kappa_{\phi\theta}$ constant,
- = 2 for $\kappa_{\phi\phi}$, $-\kappa_{\theta\theta}$ constant,
- = 3 for $\epsilon_{\phi\theta}$ constant,
- = 4 for $\epsilon_{\phi\phi}$, $-\epsilon_{\theta\theta}$ constant,

THEINC is the increment per element in the θ direction, cf Fig 4b,

PHIINC is the increment per element in the ϕ direction, cf Fig 4b,

THEZERO is the θ coordinate of the centre of the patch (usually 0.0 or 45.0),

PHIZERO is the ϕ coordinate of the centre of the patch (usually 90.0).

THEINC, PHIINC, THEZERO and PHIZERO are given in degrees.

During the execution of these programs arrays X, Y, Z are filled with the x, y and z coordinates of each node. Arrays ATHE and APHI are filled with the θ and ϕ coordinates in degrees. Note that the midside values of θ and ϕ are the mean of the values of θ and ϕ at the vertex nodes, thus the midside node is not necessarily on the geodesic line between the vertex nodes.

For SEMILOOF data, ie ITYPE = 1, the output to CPO is

NODE _i	X _i	Y _i	Z _i
-------------------	----------------	----------------	----------------

for $i = 1, 2, \dots, i, \dots, \text{NNODE}$. The output to CP1 is

ICASE			
N	θ_N	ϕ_N	
0	0.0	0.0	
M	θ_M	ϕ_M	ISIDE

where N is repeated for each vertex node around the boundary and M is repeated for each midside node around the boundary. ISIDE = 1, 2, 3 or 4 according to the side (see Fig 3a).

For NASTRAN QUAD4 data, ie ITYPE = 2, the output to CPO is

GRID	NODE _i	X _i	Y _i	Z _i
------	-------------------	----------------	----------------	----------------

for $i = 1, 2, \dots, i, \dots, \text{NNODE}$. This can be incorporated directly into a NASTRAN bulk data deck. The output to CP1 is

ICASE			
N	θ_N	ϕ_N	ISIDE

where N is repeated for each node on the boundary and ISIDE = 1, 2, 3, 4 according to the side and ISIDE is set equal to 5 for a node on the corner of the patch. This can then be used directly as input to program GENNASTBC to obtain the actual displacements around the boundary.

3.2 Program GENIPAT

This program generates data for a 3×3 patch of SEMILOOF or NASTRAN QUAD4 elements on the surface of a sphere. The θ and ϕ coordinates of the four central vertex nodes are given, enabling the user to generate the coordinates of a mesh with a regular boundary but an irregular interior, *eg* as in Fig 3c.

The first line of input is identical to the input for program GENPATCH, as described in section 3.1, *ie*

ITYPE	ICASE	THEINC	PHIINC	THEZERO	PHIZERO
-------	-------	--------	--------	---------	---------

the subsequent lines are, for SEMILOOF:

θ_{14}	ϕ_{14}
θ_{16}	ϕ_{16}
θ_{25}	ϕ_{25}
θ_{27}	ϕ_{27}

and for NASTRAN:

θ_6	ϕ_6
θ_7	ϕ_7
θ_{10}	ϕ_{10}
θ_{11}	ϕ_{11}

All θ and ϕ coordinates are given in degrees.

The output is similar to that of program GENPATCH.

4 PROGRAM GENNASTBC

This program generates the prescribed boundary condition data for a 3×3 patch of NASTRAN QUAD4 elements on the surface of a sphere as in Fig 2b. This can be done for each of five patch cases as specified by an input parameter.

The input to the program is the output on CP1 from program GENPATCH, *ie*

ICASE
N θ_N ϕ_N ISIDE

where ICASE = 0 for constant pressure loading,

= 1 for $\kappa_{\theta\phi}$ constant,

= 2 for $\kappa_{\phi\phi}$, $-\kappa_{\theta\theta}$ constant,

= 3 for $\epsilon_{\theta\phi}$ constant,

= 4 for $\epsilon_{\phi\phi}$, $-\epsilon_{\theta\theta}$ constant,

N is the node number on the boundary,

θ_N and ϕ_N are the θ and ϕ coordinates, in degrees at node N,

ISIDE is the side number 1, 2, 3 or 4 on which the node is situated or 5 if the node is on the corner of a patch, (ISIDE is only read for ICASE = 0).

The output is to three card punch files CPO, CP1, CP2 ready for inclusion in a NASTRAN bulk data deck.

4.1 Subroutine PRESSURE

This subroutine calculates the boundary conditions for a patch in a state of constant pressure acting in a complete sphere. The subroutine is called when ICASE = 0. For unit pressure loading U_ϕ , U_θ and W , the displacements in the ϕ and θ directions and outward normal to the shell surface respectively, are

$$\left. \begin{aligned} U_\phi &= 0 \\ U_\theta &= 0, \\ W &= \frac{kR^2}{2D(1+\nu)} \end{aligned} \right\} \quad (4-1)$$

where $k = \frac{h^2}{12R^2}$,

h is the shell thickness

D is the flexural rigidity

$$= \frac{Eh^3}{12(1-\nu^2)}$$

ν is Poisson's ratio.

The patch may be centred anywhere on the surface of the sphere. Conditions of symmetry are applied to each side of the patch and are output on single precision MPC (multi-point constraint) cards on card punch file CPO. (Files CP1 and CP2 remain empty.) On sides 1 and 3 (see Fig 3a) and 5, (where 5 is a corner of the patch) the conditions are

$$\left. \begin{aligned} -\cos \phi \cos \theta U_x - \cos \phi \sin \theta U_y + \sin \phi U_z &= 0 \\ -\sin \theta \omega_x + \cos \theta \omega_y &= 0 \end{aligned} \right\} \quad (4-2)$$

and on sides 2, 4 and 5 they are

$$\left. \begin{aligned} -\sin \theta U_x + \cos \theta U_y &= 0 \\ -\cos \phi \cos \theta \omega_x - \cos \phi \sin \theta \omega_y + \sin \phi \omega_z &= 0 \end{aligned} \right\} \quad (4-3)$$

where U_x, U_y, U_z and $\omega_x, \omega_y, \omega_z$ are the displacements and rotations in the x, y and z directions.

4.2 Subroutine KTHEPHI

This subroutine is called when ICASE = 1. It outputs the prescribed inextensional bending case displacements for the nearly constant curvature change state, where

$$\left. \begin{aligned} \bar{\kappa}_{\theta\phi} &\simeq \text{constant} \\ \kappa_{\phi\phi} &= -\kappa_{\theta\theta} \simeq 0. \end{aligned} \right\} \quad (4-4)$$

The prescribed displacements are

$$\left. \begin{aligned} U_\phi &= \frac{B}{2} \sin \phi \left(\tan^2 \frac{\phi}{2} + \cot^2 \frac{\phi}{2} \right) \cos 2\theta, \\ &= B(2 - \sin^2 \phi) \operatorname{cosec} \phi \cos 2\theta, \\ U_\theta &= -2B \cot \phi \sin 2\theta, \\ W &= B(2 + \sin^2 \phi) \cot \phi \operatorname{cosec} \phi \cos 2\theta. \end{aligned} \right\} \quad (4-5)$$

When the patch is centred at $\theta = 45^\circ$, $\phi = 90^\circ$ these displacements may be written as

$$\left. \begin{aligned} U_x &= 4B \cos \phi \cos^3 \theta / \sin \phi, \\ U_y &= -4B \cos \phi \sin^3 \theta / \sin \phi, \\ U_z &= -B \cos 2\theta (\sin^2 \phi - 2 \cos^2 \phi) / \sin^2 \phi, \\ \omega_\phi &= 4B \cos 2\theta / R \sin^3 \phi, \\ \omega_\theta &= 4B \sin 2\theta \cos \phi / R \sin^3 \phi, \end{aligned} \right\} \quad (4-6)$$

where $B = R^2(1 - \nu)/Eh^3$

and ω_θ and ω_ϕ are the rotations about the θ and ϕ directions.

The quantities U_x , U_y and U_z , *ie* freedoms 1, 2 and 3 are prescribed on double precision SPC cards on CPO. ω_θ and ω_ϕ are resolved into ω_x , ω_y and ω_z and are prescribed on double precision MPC cards on CP1, *ie*

$$\left. \begin{aligned} -\sin \theta \omega_x + \cos \theta \omega_y - \omega_\phi &= 0, \\ -\cos \theta \cos \phi \omega_x - \cos \phi \sin \theta \omega_y + \sin \phi \omega_z - \omega_\theta &= 0. \end{aligned} \right\} (4-7)$$

To represent the constant terms ω_θ and ω_ϕ in the multipoint constraints it is necessary to introduce scalar points which are prescribed to have unit displacement on single precision SPC cards on CP2.

4.3 Subroutine KTHETHE

This subroutine, called when ICASE = 2, is for the nearly constant curvature change state

where

$$\left. \begin{aligned} \kappa_{\phi\phi} &= -\kappa_{\theta\theta} \simeq \text{constant}, \\ \bar{\kappa}_{\theta\phi} &\simeq 0. \end{aligned} \right\} (4-8)$$

The prescribed displacements are

$$\left. \begin{aligned} U_\phi &= \frac{1}{12} \sin \phi \left(\tan^2 \frac{\phi}{2} - \cot^2 \frac{\phi}{2} \right) \cos 2\theta, \\ &= -\frac{1}{3} \cot \phi \cos 2\theta, \\ U_\theta &= \frac{1}{6} (2 - \sin^2 \phi) \operatorname{cosec} \phi \sin 2\theta, \\ W &= -\frac{1}{3} \operatorname{cosec}^2 \phi \cos 2\theta. \end{aligned} \right\} (4-9)$$

When the patch is centred at $\theta = 0^\circ$, $\phi = 90^\circ$ these displacements may be written as

$$\left. \begin{aligned} U_x &= -\cos^3 \theta (\cos^2 \phi + 1) / 3 \sin \phi, \\ U_y &= \sin^3 \theta (\cos^2 \phi + 1) / 3 \sin \phi, \\ U_z &= \cos 2\theta \cos \phi \left(1 - \frac{1}{\sin^2 \phi} \right) / 3, \end{aligned} \right\} (4-10)$$

$$\left. \begin{aligned} \omega_{\phi} &= -\cos \phi \cos 2\theta (\sin^2 \phi + 2) / 3R \sin^3 \phi, \\ \omega_{\theta} &= \sin 2\theta (2 \sin^2 \phi - \sin^4 \phi - 4) / 6R \sin^3 \phi. \end{aligned} \right\}$$

The quantities U_x, U_y, U_z are prescribed on double precision SPC cards on CPO. ω_{θ} and ω_{ϕ} are resolved, as in equation (4-7), into ω_x, ω_y and ω_z and are prescribed on double precision MPC cards on CP1. As in subroutine KTHEPHI scalar points are again introduced and their displacements fixed on single precision SPC cards output to CP2.

4.4 Subroutine ETHEPHI

This subroutine, called for ICASE = 3, prescribes boundary displacements for the nearly constant strain patch test

$$\left. \begin{aligned} \epsilon_{\theta\phi} &\simeq \text{constant} \\ \epsilon_{\phi\phi} &= -\epsilon_{\theta\theta} \simeq 0. \end{aligned} \right\} \quad (4-11)$$

The appropriate displacements are

$$\left. \begin{aligned} U_{\phi} &= -\frac{1}{12} \left[\left\{ 3 + (2 + \cos \phi) \cos \phi \right\} \tan^2 \frac{\phi}{2} \right. \\ &\quad \left. + \left\{ 3 - (2 - \cos \phi) \cos \phi \right\} \cot^2 \frac{\phi}{2} \right] \operatorname{cosec} \phi \cos 2\theta, \\ &= -\frac{1}{6} \operatorname{cosec}^3 \phi (4 - 2 \sin^2 \phi + \sin^4 \phi) \cos 2\theta, \\ U_{\theta} &= -\frac{1}{3} \cot \phi \operatorname{cosec}^2 \phi (2 + \sin^2 \phi) \sin 2\theta, \\ W &= \frac{1}{6} \cot \phi \operatorname{cosec}^2 \phi (2 + \sin^2 \phi) \cos 2\theta. \end{aligned} \right\} \quad (4-12)$$

For the case when the patch is centred at $\theta = 45^\circ$, $\phi = 90^\circ$ these displacements may be written as

$$\left. \begin{aligned} U_x &= \cos \phi \left\{ - (4 - 2 \sin^2 \phi + \sin^4 \phi) \cos \theta \cos 2\theta \right. \\ &\quad \left. + (2 + \sin^2 \phi) (2 \sin \theta \sin 2\theta + \sin \phi \cos \theta \cos 2\theta) \right\} / 6 \sin^3 \phi, \end{aligned} \right\}$$

$$\begin{aligned}
 U_y &= \cos \phi \left\{ -(4 - 2 \sin^2 \phi + \sin^4 \phi) \sin \theta \cos 2\theta \right. \\
 &\quad \left. - (2 + \sin^2 \phi)(2 \cos \theta \sin \theta - \sin \phi \sin \theta \cos 2\theta) \right\} / 6 \sin^3 \phi, \\
 U_z &= \frac{1}{2 \sin^2 \phi} (1 + \cos^2 \phi) \cos 2\theta, \\
 \omega_\phi &= \left\{ -(4 - 2 \sin^2 \phi + \sin^4 \phi) \sin \phi \cos 2\theta \right. \\
 &\quad \left. + 3(\cos^2 \phi + 1) \cos 2\theta \right\} / 6R \sin^4 \phi, \\
 \omega_\theta &= \cos \phi (2 + \sin^2 \phi) (1 - \sin \phi) \sin 2\theta / 3R \sin^4 \phi.
 \end{aligned} \tag{4-13}$$

The quantities U_x , U_y , U_z are prescribed on double precision SPC cards on CPO. ω_θ and ω_ϕ are resolved into ω_x , ω_y and ω_z , see equation (4-7), and prescribed on double precision MPC cards on CP1. Once again it is necessary to introduce scalar points, as described in subroutine KTHEPHI, and these displacements are fixed on single precision SPC cards output to CP2.

4.5 Subroutine ETHETHE

This subroutine is called when ICASE = 4. It prescribes the boundary displacements for the nearly constant strain state

$$\begin{aligned}
 \text{where} \quad \epsilon_{\phi\phi} &= -\epsilon_{\theta\theta} \hat{=} \text{constant}, \\
 \epsilon_{\theta\phi} &\hat{=} 0.
 \end{aligned} \tag{4-14}$$

The displacements are

$$\begin{aligned}
 U_\phi &= \frac{1}{12} \left[\left\{ 3 + (2 + \cos \phi) \cos \phi \right\} \tan^2 \frac{\phi}{2} \right. \\
 &\quad \left. - \left\{ 3 - (2 - \cos \phi) \cos \phi \right\} \cot^2 \frac{\phi}{2} \right] \operatorname{cosec} \phi \cos 2\theta, \\
 &= -\frac{2}{3} \cot \phi \operatorname{cosec}^2 \phi \cos 2\theta, \\
 U_\theta &= -\frac{2}{3} \operatorname{cosec}^3 \phi \sin^2 \theta, \\
 W &= \frac{1}{3} \operatorname{cosec}^2 \phi \cos 2\theta.
 \end{aligned} \tag{4-15}$$

When the patch is centred at $\theta = 0^\circ$, $\phi = 90^\circ$ these may be written as

$$\begin{aligned}
 U_x &= (3 \sin^2 \phi \cos \theta \cos 2\theta - 2 \cos 3\theta) / 3 \sin^3 \phi, \\
 U_y &= (3 \sin^2 \phi \sin \theta \cos 2\theta - 2 \sin 3\theta) / 3 \sin^3 \phi, \\
 U_z &= \cos \phi \cos 2\theta / \sin^2 \phi, \\
 \omega_\phi &= -4 \cos \phi \cos 2\theta / 3R \sin^3 \phi, \\
 \omega_\theta &= 0.
 \end{aligned}
 \tag{4-16}$$

The quantities U_x , U_y , U_z are prescribed on double precision SPC cards on CPO. ω_θ and ω_ϕ are resolved into ω_x , ω_y and ω_z , as in equation (4-7) and are prescribed on double precision MPC cards on CPI. Scalar points are introduced, as described in subroutine KTHEPHI, and their displacements fixed on single precision SPC cards output to CP2.

5 NASTRAN DATA GENERATION PROGRAMS

The programs in this section are to convert free format files containing either the coordinates of a point or the node numbers around an element into output that can be directly incorporated into a NASTRAN bulk data deck.

5.1 Program GENQUAD4

This program gives the element numbering around a QUAD4 quadrilateral plate element (which is a warped isoparametric plate membrane-bending element). The output is in the form of NASTRAN CQUAD4 cards.

The input is

```
N  N1  N2  N3  N4
```

where N is the element number.

N1, N2, N3, N4 are the node numbers around element N. This is repeated for each element until the end of the input file is reached.

The output in fixed format for a constant thickness, homogeneous isotropic material is in the form.

```
CQUAD4 N 1 N1 N2 N3 N4 0.
```

The third field refers to the property card and the eighth field, to the material orientation.

5.2 Program GENCTRIA3

This program outputs CTRIA3 cards that can be included directly into a NASTRAN bulk data deck. CTRIA3 cards are used to define a triangular plate element (compatible with the QUAD4 quadrilateral element).

The input, all integers in free format, is

```
N  N1  N2  N3
```

where N is the element number,

N1, N2, N3 are the node numbers around node N .

This last line is repeated until the end of the input file is reached.

The output is

```
CTRIA3  N  1  N1  N2  N3  0.
```

The value of 1 in the third field refers to the NASTRAN property card and 0. in the last field gives the orientation of the material, *ie* for a homogeneous isotropic material.

5.3 Program GENGRID

This program takes as input the node number and corresponding x, y and z coordinates of a mesh and outputs them in the form of GRID cards for including in a NASTRAN bulk data deck. In addition, there is the option of specifying freedoms that are constrained to zero, *ie* permanent single point constraints.

The input data is read in in free format and is

```
NNODE
```

```
Ni  Xi  Yi  Zi
```

```
Nj  IFREEDOM
```

where NNODE is the total number of nodes,

X_i, Y_i, Z_i are the x, y and z coordinates at node i ,

(i = 1, 2, ..., NNODE) ,

N_j is the node at which the freedom is being constrained,

IFREEDOM is a number consisting of any of the integers 1 + 6 with no embedded blanks, the integer contains

1	if	$Ux_j = 0$,
2	if	$Uy_j = 0$,
3	if	$Uz_j = 0$,
4	if	$\omega x_j = 0$,
5	if	$\omega y_j = 0$,
6	if	$\omega z_j = 0$,

j is repeated for each node to be constrained in this way.

The output is either of the form

GRID	I	blank	X_i	Y_i	Z_i	
------	---	-------	-------	-------	-------	--

or

GRID	I	blank	X_i	Y_i	Z_i	blank IFREEDOM.
------	---	-------	-------	-------	-------	-----------------

NB If

$$x_i, y_i \text{ or } z_i \leq -1$$

the output field width is exceeded and the number is flagged by an asterisk. The output file must then be edited to reduce the number of eight characters and any equal positive numbers reduced similarly to maintain symmetry.

6 PROGRAM CHANGEDATA

This program converts SEMILOOF data to 'facet' data by altering the position of the midside nodes to be halfway along the straight line joining the two vertex nodes. It takes as its input a SEMILOOF data file and outputs the node number and new coordinates to a card punch file that can then be edited back into the original file. If there are more than 200 nodes in the data file, the program has to be altered to increase the size of arrays X, Y and Z.

7 CONCLUSIONS

The computer programs presented here are a faster and more accurate method of data generation than manual preparation.

The programs have been used extensively to create all of the data required for the problems considered by Morley and Morris².

AppendixSEMILOOF PROGRAM

The program listed below generates the element stiffness matrix and associated matrices for the SEMILOOF finite element. The master segment reads in the data and then, for each element in turn, calls subroutine ELSTIF, which calls other subroutines to calculate the element stiffness matrix. Immediately before control is returned from subroutine ELSTIF, the matrices needed to calculate the stresses are written to channel 8. On returning to the master segment, the element stiffness matrix is written to channel 7 and the matrices containing pressure and gravity loads are written to channel 9. The next line of element connectors are then read and the matrix calculation process repeated.

```

SHORTLIST
PROGRAM(LOOF)
INPUT5=CRO
OUTPUT6=LPO
OUTPUT7=ED7/UNFORMATTED(DAF7)
OUTPUT8=ED8/UNFORMATTED(DAF8)
OUTPUT9=ED9/UNFORMATTED(DAF9)
TRACE 0
END
MASTER
C**** PROGRAM TO GENERATE THE ELEMENT STIFFNESS MATRICES AND ASSOCIATED
C**** MATRICES FOR THE SEMILOOF SHELL FINITE ELEMENT
COMMON/FRON/ELST(528),ELR(32,2),LNODS(8),LNODZ
COMMON/SHELL/AREA, ELXYZT(9,4),FRAM(3,3),POINT(3),
1 SIDE, THIK, WSHEL(13,45),XITA(2), XYZ(241,3)
COMMON/ERNFST/YMODT(10), POIST(10), THIKT(10), DENSTT(10),
1 PRESST(10), LMAT(241), STRESH(24,32)
DO 800 I=1,13
DO 800 J=1,45
WSHEL(I,J)=0.0
800 CONTINUE
C
C**** READ PROBLEM PARAMETERS.
C
4 READ(5,500) NELZ
500 FORMAT(I0)
WRITE(6,601)NELZ
601 FORMAT(/7H NELZ =,I4,9H ELEMENTS)
C
C**** READ MATERIAL PROPERTIES AND NODAL COORDINATES.
C
MATZ=1
DO 6 MAT = 1,MATZ
C**** YMODT CONTAINS YOUNGS MODULUS
C**** POIST CONTAINS POISSONS RATIO
C**** THIKT CONTAINS SHELL THICKNESS
C**** DENSTT CONTAINS MATERIAL DENSITY
C**** PRESST CONTAINS PRESSURE LOADING
READ(5,502) YMODT(MAT), POIST(MAT), THIKT(MAT), DENSTT(MAT),
1 PRESST(MAT)
502 FORMAT(E0.0,4F0.0)
6 CONTINUE
C**** NZ CONTAINS THE TOTAL NUMBER OF VETEX AND MIDSIDE NODES
READ(5,591) NZ
591 FORMAT(I0)
C**** FOR EACH NODE READ IN NODE NUMBER AND X,Y,Z COORDINATES
READ(5,504) (N,(XYZ(N,I), I = 1,3), N = 1,NZ)
504 FORMAT(I0,3F0.0)
WRITE(6,604) (N, (XYZ(N,I), I = 1,3), N = 1,NZ)
604 FORMAT(/3X,1HN,6X,1HX,9X,1HY,9X,1HZ/(1X,I3,3F10.6))
WRITE(6,606)
606 FORMAT(/15H ELEMENT NUMBER,5X,20HNODE NUMBERS = LNODS)
C
C**** FOR EACH NODE, READ IN ELEMENT CONNECTIONS AND CALL SEMILOOF ROUTINES
C
DO 7 NEL = 1,NELZ
LMAT(NEL)=1
N = NEL
READ(5,592) (LNODS(I), I = 1,8)
592 FORMAT(8I0)
WRITE(6,608) N, (LNODS(I), I=1,8)

```

```

408 FORMAT(2I11,9I5,(I22,9I5))
C**** LNODZ = 6 FOR A TRIANGULAR ELEMENT OR = 8 FOR A QUADRILATERAL ELEMENT
      LNODZ=8
      IF(LNODS(8).EQ.0)LNODZ=6
C
C**** CALL SUBROUTINE TO SET UP ELEMENT STIFFNESS MATRIX AND ASSOCIATED
C**** MATRICES
      CALL ELSTIF(NEL)
C**** WRITE LNODZ CONTAINING NUMBER OF NODES AROUND THE ELEMENT AND ELST
C**** CONTAINING THE ELEMENT STIFFNESS MATRIX TO CHANNEL 7
      WRITE(7)LNODZ
      WRITE(7)ELST
C**** SEE SUBROUTINE ELSTIF FOR PRINT OUT OF LNODZ,STRESM,PT,FR
C**** CONTAINING STRESS INFORMATION TO CHANNEL 8
C**** WRITE LNODZ AND ELR CONTAINING PRESSURE AND GRAVITY LOADS
C**** TO CHANNEL 9
      WRITE(9)LNODZ
      WRITE(9)(ELR(I,1),I=1,32)
7 CONTINUE
  STOP
  END
  BLOCK DATA
C
C**** TO INITIALIZE COEFFICIENTS FOR CORNER-MIDSIDE AND LOOF VERSIONS
C
C**** OF QUADRATIC TRIANGLE AND QUADRILATERAL FOR SUBROUTINE SFR.
C
  DIMENSION COEFA(166), COEFB(81)
  COMMON/COEF/COEF(247)
  EQUIVALENCE (COEF(1), COEFA(1)), (COEF(167),COEFB(1))
  DATA COEFA/ 1.,-3.,-3., 2., 4., 2., 0., 4., 0.,-4.,-4., 0., 0.,
1 -1., 0., 2., 0., 0., 0., 0., 0., 0., 0., 4., 0., 0., 0.,-1., 0., 0.,
2 2., 0., 0., 4., 0.,-4.,-4., 0.910683603, 1.577350269,
3 -6.041451884,-6.196152423, 2.464101615, 8.928203230, 1.732050808,
4 -0.244016936, 0.422649731, 2.041451884, 4.196152423,-4.464101615,
5 -4.928203230,-1.732050808, 0.333333333,-1.422649731,-2.577350269,
6 -1.464101615, 5.000000000, 5.464101615, 1.732050808, 0.333333333,
7 -2.577350269,-1.422649731, 5.464101615, 5.000000000,-1.464101615,
8 -1.732050808,-0.244016936, 2.041451884, 0.422649731,-4.928203230,
9 -4.464101615, 4.196152423, 1.732050807, 0.910683602,-6.041451884,
1 1.577350269, 8.928203230, 2.464101615,-6.196152422,-1.732050807,
2 -1.,6.,6.,-6.,-6.,-6.,0.,-25.,0.,0.,25.,25.,25.,-25.,-25.,0.,
3 .5,0.,-.5,-.5,0.,0.,0.,.5,0.,-.25,0.,0.,25.,-25.,25.,25.,-25.,0.,
4 .5,.5,0.,0.,0.,-.5,-.5,0.,0.,-.25,0.,0.,25.,25.,25.,25.,25.,0.,
5 .5,0.,.5,-.5,0.,0.,0.,-.5,0.,-.25,0.,0.,25.,-25.,25.,-25.,25.,0.,
6 .5,-.5,0.,0.,0.,-.5,.5,0.,0.,1.,0.,0.,-1.,0.,-1.,0.,0.,1./
  DATA COEFB/ 0.000000000, 0.216506351,-0.375000000,-0.093750000,
1 0.216506351, 0.281250000,-0.649519053, 0.375000000,-0.324759526,
2 -0.000000000,-0.216506351,-0.375000000,-0.093750000,-0.216506351,
3 0.281250000, 0.649519053, 0.375000000, 0.324759526, 0.000000000,
4 0.375000000, 0.216506351, 0.281250000,-0.216506351,-0.093750000,
5 -0.375000000,-0.649519053,-0.324759526, 0.000000000, 0.375000000,
6 -0.216506351, 0.281250000, 0.216506351,-0.093750000,-0.375000000,
7 0.649519053, 0.324759526,-0.000000000,-0.216506351, 0.375000000,
8 -0.093750000, 0.216506351, 0.281250000, 0.649519053,-0.375000000,
9 -0.324759526, 0.000000000, 0.216506351, 0.375000000,-0.093750000,
1 -0.216506351, 0.281250000,-0.649519053,-0.375000000, 0.324759526,
2 -0.000000000,-0.375000000,-0.216506351, 0.281250000,-0.216506351,
3 -0.093750000, 0.375000000, 0.649519053,-0.324759526,-0.000000000,
4 -0.375000000, 0.216506351, 0.281250000, 0.216506351,-0.093750000,
5 .375,-.649519053,.324759526,1.,0.,0.,-.75,0.,-.75,0.,0.,0./
  END

```

SUBROUTINE ELSTIF(NEL)

```

C
C**** DEMONSTRATION PROGRAM FOR ELEMENT STIFFNESS ETC.
REAL PT(3,4),FR(3,12)
C
  DIMENSION B(6,32), BV(192), DMOD(6,6), DMODV(36), XGAUS(4,4),
1  GVECT(6), GRAVY(3)
  COMMON/FRON/ELST(528),ELR(32,2),LNODS(8),LNODZ
  COMMON/SHELL/AREA, ELXYZT(9,4), FRAM(3,3), POINT(3),
1  SIDE, THIK, WSHEL(13,45), XITA(2), XYZ(241,3)
  COMMON/ERNEST/YMODT(10), POIST(10), THIKT(10), DENSTT(10),
1  PRESST(10), LMAT(241), STRESM(24,32)
  EQUIVALENCE (GVECT(1),R), (GVECT(2),S), (GVECT(3),T),
1  (GVECT(4),U), (GVECT(5),V), (GVECT(6),W),
2  (B(1,1),BV(1)), (DMOD(1,1),DMODV(1))
C
C**** INITIALISE ARRAYS.
C
  DATA XGAUS/0., 4*.5, 0., 2*.5, -.577350269, 2*.577350269,
1  3*-.577350269, 2*.577350269/
  LVABZ = 4*LNODZ
  IZ=(LVABZ*(LVABZ+1))/2
  DO 2 I = 1,IZ
2  ELST(I) = 0.0
  DO 3 I = 1,LVABZ
3  ELR(I,1) = 0.0
  DO 4 N = 1,LNODZ
  DO 4 I = 1,3
4  ELXYZT(N,I) = XYZ(LNODS(N),I)
C
C**** DEFINE PROBLEM CONSTANTS.
C
  MAT = LMAT(NEL)
  YMOD = YMODT(MAT)
  POIS = POIST(MAT)
  DENSTY = DENSTT(MAT)
  PRESS = PRESST(MAT)
  THIK = THIKT(MAT)
  DO 6 N = 1,LNODZ
6  ELXYZT(N,4) = THIK
  GRAVY(1) = 0.0
  GRAVY(2) = 0.0
  GRAVY(3) = 1.0

```

```

C
C**** CALL HALOOF AND CREATE MATRIX B AT THE INTEGRATING POINTS.
C
      IGAUSZ = LNODZ/2
      DO 26 IGAUS = 1, IGAUSZ
      XITA(1) = XGAUS(IGAUS, LNODZ-5)
      XITA(2) = XGAUS(IGAUS, LNODZ-4)
      CALL HALOOF(NEL)
      DO 8 N = 1, LVABZ
      B(1,N) = WSHEL(4,N)
      B(2,N) = WSHEL(7,N)
      B(3,N) = WSHEL(5,N) + WSHEL(6,N)
      B(4,N) = WSHEL(10,N)
      B(5,N) = WSHEL(12,N)
      8 B(6,N) = 2.0*WSHEL(11,N)
C
C**** CREATE MODULUS MATRIX FOR CALCULATING THE STIFFNESSES.
C
      DO 10 I = 1, 36
      10 DMODV(I) = 0.0
      GASH = YMOD*AREA*THIK/(1.0-POIS*POIS)
      DMODV(1) = GASH
      DMODV(2) = GASH*POIS
      DMODV(7) = GASH*POIS
      DMODV(8) = GASH
      DMODV(15) = 0.5*GASH*(1.0-POIS)
      GASH = GASH*THIK*THIK/12.0
      DMODV(22) = GASH
      DMODV(23) = GASH*POIS
      DMODV(28) = GASH*POIS
      DMODV(29) = GASH
      DMODV(36) = 0.5*GASH*(1.0-POIS)
C
C**** GET DB, A COLUMN AT A TIME, AND COMPUTE STIFFNESSES.
C
      NSTIF = 0
      DO 24 NROW = 1, LVABZ
      DO 14 K = 1, 6
      GASH = 0.0
      LDEL = 6*(NROW-K)
      LA = 6*K-5
      LZ = 6*K
      DO 12 L = LA, LZ
      L1=L+LDEL
      12 GASH=GASH+BV(L1)*DMODV(L)
      14 GVECT(K) = GASH
      IDEL = -5
      DO 16 KOL = 1, NROW
      IDEL = IDEL + 6
      NSTIF1=NSTIF+KOL
      16 ELST(NSTIF1)=ELST(NSTIF1)+R*BV(IDEL)+S*BV(IDEL+1)
      1 + T*BV(IDEL+2) + U*BV(IDEL+3) + V*BV(IDEL+4) + W*BV(IDEL+5)

```

```

C
C**** ACCUMULATE PRESSURE AND GRAVITY LOADS.
C
      GASH = ELR(NROW,1)
      DO 18 I = 1,3
      GRAVY(I) = POINT(I)
      GRAVY(3) = 0.0
      FACT = AREA*THIK*DENSTY*GRAVY(I)
      DO 18 J = 1,3
18     GASH = GASH + FACT*FRAM(I,J)*WSHEL(J,NROW)
      ELR(NROW,1) = GASH + PRESS*AREA*WSHEL(3,NROW)
C
C**** COMPUTE AND STORE STRESS MATRIX AT GAUSS POINTS.
C
      DO 22 I = 1,3
      J=6*(IGAUS-1)+I
      STRESM(J,NROW) = GVECT(I)/(AREA*THIK)
22     STRESM(J+3,NROW) = GVECT(I+3)*6.0/(AREA*THIK*THIK)
24     NSTIF = NSTIF + NROW
      DO 7 I=1,3
      PT(I,IGAUS)=POINT(I)
      DO 9 J=1,3
      NP=3*(IGAUS-1)+J
9      FR(I,NP)=FRAM(I,J)
7      CONTINUE
26     CONTINUE
C**** WRITE LNODZ,STRESM,PT,FR CONTAINING STRESS INFORMATION
C**** TO CHANNEL 8
      WRITE(8)LNODZ
      WRITE(8)STRESM
      WRITE(8)PT
      WRITE(8)FR
      RETURN
      END
      SUBROUTINE HALOOF(NEL)
C
C**** TO CREATE SHAPE FUNCTION ARRAY WSHEL, FOR SEMILOOF SHELL ELEMENT.
C
C**** WRITTEN BY BRUCE IRONS, JULY 1972,WASHINGTON D.C.
C
      DIMENSION AREAV(3), FRAME(3,3), GENSID(6,4),
1      SHEAR(11,43), SIGT(3), SWOP(6), THIKDD(3,3), TRANS(2,2),
2      VLOOF(3,36), WCORN(10,3), WLOOF(10,3), XGAUS(4,4), XILOOF(9,4),
3      XLOCAL(2), XYZDD(3,3), XYZPRE(8,4)
      COMMON/FRON/ELST(528),ELR(32,2),LNODS(8),LNODZ
      COMMON/SHELL/AREA, ELXYZT(9,4), FRAM(3,3), POINT(3),
1      SIDE, THIK, WSHEL(13,45), XITA(2), XYZ(241,3)
      EQUIVALENCE (T11,TRANS(1,1)), (T12,TRANS(1,2)),
1      (T21,TRANS(2,1)), (T22,TRANS(2,2))
      DATA GENSID/1., -1., 0., 3*-.5, 0., 1., -1., 4*1., 0., -1.,
1      4*0., 1., 0., -1., 2*1./,XILOOF/.211324866, 2*.788675134,
2      .211324866, 2*0., .3333333333, 4*0., .211324866, 2*.788675134,
3      .211324866, .3333333333, 2*0., -.577350269, .577350269, 2*1.,
4      .577350269, -.577350269, 2*-1., 0., 2*-1., -.577350269,
5      .577350269, 2*1., .577350269, -.577350269, 0. /,
6      XGAUS/0., 4*.5, 0., 2*.5, -.577350269, 2*.577350269,
7      3*-.577350269, 2*.577350269/, XYZPRE/32*0.0/, NOZPRE/0/

```

```

C
C**** GENERATE NSTAGE TO DEFINE PATH THROUGH HALOOF.
C
  N  ERROR = 1
  IF(LNODZ.NE.6. AND .LNODZ.NE.8) GO TO 99
  NSTAGE = 4
  IF(LNODZ.NF.NOZPRE) NSTAGE = 2
  NOZPRE = LNODZ
  DO 2 LNOD = 1, LNODZ
  DO 2 NX = 1, 4
  IF(ELXYZT(LNOD, NX).NE. XYZPRE(LNOD, NX)) NSTAGE = 2
2  CONTINUE
  IF(NSTAGE.EQ.4) GO TO 18

C
C**** INITIALIZATION FOR NEW ELEMENT. NSTAGE = 1. FIND CENTRE COORDINATES
C
  LIMZ = (3*LNODZ)/2 - 1
  LVABZ = 4*LNODZ
  LNODZA = LNODZ + 1
  LVABZA = LVABZ + 1
  LVABZZ = LVABZ + LIMZ
  DO 3 L = LNODZA, LIMZ
  DO 3 J = 1, LVABZZ
3  SHEAR(L, J) = 0.0
  DO 5 NX = 1, 4
  GASH = 0.0
  LNODZ1 = LNODZ/2
  DO 4 KORN = 1, LNODZ1
  DO 4 K = 1, 2
  K1 = 2*KORN + K - 2
  K2 = 216*K - 408 - LNODZ*(21*K - 41)
4  GASH = GASH
  1 + 8.0*ELXYZT(K1, NX)/FLOAT(K2)
5  ELXYZT(9, NX) = GASH

C
C**** DIAGNOSTICS FOR A NEW ELEMENT. RELATE COORDINATES TO CENTRE.
C
  DO 10 I = 1, LNODZ
  N  ERROR = 2
  IF(ELXYZT(I, 4).LE.0.0) GO TO 99
  IF(I.EQ.LNODZ) GO TO 9
  JA = I + 1
  DO 8 J = JA, LNODZ
  N  ERROR = 3
  IF(LNODS(I).EQ.LNODS(J)) GO TO 99
  DO 7 K = 1, 3
  IF(ELXYZT(I, K).NE.ELXYZT(J, K)) GO TO 8
7  CONTINUE
  N  ERROR = 4
  GO TO 99
8  CONTINUE
9  DO 10 NX = 1, 4
  IF(NX.NE.4) ELXYZT(I, NX) = ELXYZT(I, NX) - ELXYZT(9, NX)
10 XYZPRE(I, NX) = ELXYZT(I, NX)

```

```

C
C**** CREATE SWOP = 1.0 OR -1.0, TO IN
C**** CREATE SWOP = 1.0 OR -1.0, TO IMPLEMENT SIGN CHANGES AT LOOF NODES.
C
C**** ALSO INTERPOLATE TO ESTIMATE NORMAL THICKNESSES AT LOOF NODES.
C
      VLOOF(1,LVABZA) = ELXYZT(9,4)
      DO 12 NSIDE = 1,6
12     SWOP(NSIDE) = 1.0
          LAST = LNODZ - 1
          DO 14 NEXT = 1, LNODZ, 2
          MID = LAST + 1
          IF(LNODS(NEXT).LT.LNODS(LAST)) SWOP(MID/2) = -1.0
          LAST1=4*LAST-3
          VLOOF(1, LAST1) = .455341801*ELXYZT(LAST,4)
1      + .666666667*ELXYZT(MID,4) - .122008468*ELXYZT(NEXT,4)
          MID1=4*MID-3
          VLOOF(1, MID1) = -.122008468*ELXYZT(LAST,4)
1      + .666666667*ELXYZT(MID,4) + .455341801*ELXYZT(NEXT,4)
C
C**** ALSO CHECK THAT MIDSIDE NODES ARE REASONABLY CENTRAL.
C
      GASH = 0.0
      GISH = 0.0
      GUSH = 0.0
      DO 13 I = 1,3
      ELMID = ELXYZT(MID,I)
      GASH = GASH + (ELXYZT(NEXT,I)-ELMID)**2
      GISH = GISH + (ELXYZT(LAST,I)-ELMID)**2
13     GUSH = GUSH + (ELXYZT(LAST,I)+ELXYZT(NEXT,I)-ELMID-ELMID)**2
          N ERROR = 5
          IF(ABS(GASH-GISH).GT.0.040*(GASH+GISH)) GO TO 99
          N ERROR = 6
C     IF(GUSH.GT.0.25*(GASH+GISH)) GO TO 99
14     LAST = NEXT
C
C**** ORGANISE LOOP AROUND NODES, FOR NSTAGE = 2
C
C**** DO 76 NSTAGE = 2,4 (IN NSTAGE)
      15 NLOOF = 0
      16 NLOOF = NLOOF + 1
C**** DO 67 NLOOF = 1, LNODZ+1 IF NSTAGE = 2,
C**** OR DO 67 NLOOF = 1, (3*LNODZ)/2 IF NSTAGE = 3.
      DO 17 I = 1,2
      LNODZ2=LNODZ+I-6
      IF(NSTAGE.EQ.2. OR .NLOOF.LE.LNODZ)
1     XLOCAL(I)=XILOOF(NLOOF, LNODZ2)
C
C**** AND ALSO AROUND INTEGRATING POINTS IF NSTAGE = 3.
C
      NLOOF1=NLOOF-LNODZ
      LNODZ2=LNODZ+I-6
      IF(NSTAGE.EQ.3. AND .NLOOF.GT.LNODZ)
1     XLOCAL(I)=XGAUS(NLOOF1, LNODZ2)
17 CONTINUE
      GO TO 23

```

```

C
C**** OTHERWISE, ORGANISE SINGLE-SHOT OPTION, FOR NSTAGE = 4.
C
C**** TEST WHETHER INPUT POINT IS A LOOP NODE, PLUS OR MINUS 0.0001.
C
  18 DO 19 I = 1,2
  19 XLOCAL(I) = XITA(I)
     NLOOP= LNODZA
     DO 22 MAYBE = 1, LNODZ
     DO 20 I = 1,2
     IF(ABS(XLOCAL(I)-XLOOP(MAYBE, LNODZ+I-6)).GT.0.0001) GO TO 22
  20 CONTINUE
     NLOOP = MAYBE
  22 CONTINUE
C
C**** CREATE VALUES AND XI, ETA DERIVATIVES OF X, Y, Z IN XYZDD, T IN THIKDD
C
  23 CONTINUE
     CALL SFR(XLOCAL, WCORN, WLOOP, NSTAGE)
     K = 0
     DO 27 I = 1,3
     DO 26 J = 1,3
     GASH = 0.0
     DO 24 L = 1, LNODZ
     L1=L+K
  24 GASH=GASH+WCORN(L1, I)*ELXYZT(L, J)
     XYZDD(J, I) = GASH
     IF(NSTAGE.EQ.2) GO TO 26
     GASH = 0.0
     DO 25 L = 1, LNODZA
     L1=L+K
     L2=4*L-1
  25 GASH=GASH+WLOOP(L1, I)*VLOOP(J, L2)
     THIKDD(J, I) = GASH
  26 CONTINUE
  27 K = 1
C
C**** CREATE VECTOR AREA = VAREA, AT GIVEN POINT XI, ETA.
C
     CALL VECTOR(XYZDD(1,2), XYZDD(1,3), AREAV(1))
     CALL SCALAR(AREAV(1), AREAV(1), AREASQ)
     N ERROR = 7
     IF(AREASQ.EQ.0.0) GO TO 99
     AREA = SQRT(AREASQ)
C
C**** NORMALISE VECTOR AREA INTO FRAME, COL.3, AS LOCAL UNIT NORMAL Z.
C
C**** COLUMN 2 OF FRAME BECOMES UNIT Y AROUND EDGE.
C
     DO 30 I = 1,3
     FRAME(I,3) = AREAV(I)/AREA
     GASH = 0.0
     DO 29 J = 1,2
     NLOOP1=(NLOOP+1)/2
     LNODZ3=LNODZ+J-6
  29 GASH=GASH+GENSID(NLOOP1, LNODZ3)*XYZDD(I, J+1)
  30 FRAME(I,2) = GASH

```

```

C
C**** NORMALISE Y, AND IMPLEMENT SWOP BY REVERSING SIGN OF Y.
C
C**** PUT APPROXIMATE VECTOR THICKNESS ETC. INTO VLOOP, FOR NSTAGE = 2
C
      N   ERROR = 8
      CALL SCALAR(FRAME(1,2), FRAME(1,2), SIDESQ)
      IF(SIDESQ.EQ.0.0) GO TO 99
      SIDE = SQRT(SIDESQ)
      DO 31 I = 1,3
      NLOOP4=(NLOOP+1)/2
      FRAME(I,2)=FRAME(I,2)*SWOP(NLOOP4)/SIDE
      IF(NSTAGE.NE.2) GO TO 31
      NLOOP5=4*NLOOP-2
      VLOOP(I,NLOOP5)=FRAME(I,2)
      NLOOP6=4*NLOOP-1
      NLOOP7=4*NLOOP-3
      VLOOP(I,NLOOP6)=FRAME(I,3)+VLOOP(1,NLOOP7)
      NLOOP8=4*NLOOP
      VLOOP(I,NLOOP8)=FRAME(I,3)
31 CONTINUE
C
C**** AND COLUMN 1 IS UNIT X, THE OUTWARD POINTING IN-PLANE NORMAL.
C
      CALL VECTOR(FRAME(1,2), FRAME(1,3), FRAME(1,1))
C**** CHECK THAT NORMALS ARE REASONABLY PARAPPEL, WHILE NSTAGE = 2.
C
      IF(NSTAGE.GT.2) GO TO 35
      IF(NLOOP.EQ.1) GO TO 67
      KZ = 4*NLOOP-4
      DO 32 K = 4, KZ, 4
      NLOOP9=4*NLOOP
      CALL VECTOR(VLOOP(1,NLOOP9),VLOOP(1,K),POINT(1))
      CALL SCALAR(POINT(1), POINT(1), COSSQ)
      N   ERROR = 9
      IF(COSSQ.GT.0.75) GO TO 99
32 CONTINUE
C
C**** PLACE CONTRIBUTION OF CENTRAL NODE IN VLOOP (NSTAGE = 2 ONLY)
C
C**** COMPLETE LOOP NLOOP = 1 TO LNODZ+1 FOR NSTAGE = 2.
C
      IF(NLOOP.LE.LNODZ) GO TO 67
      THIKC = VLOOP(1,LVABZA)
      DO 33 I = 1,3
      DO 33 J = 1,2
      LVABZ1=LVABZ+J
33 VLOOP(I,LVABZ1)=FRAME(I,J)*THIKC
      GO TO 67
C
C**** CREATE THE 2X2 JACOBIAN MATRIX, AND INVERT IT. (NSTAGE = 3 OR 4)
C
35 DO 36 J = 1,2
      DO 36 I = 1,2
      CALL SCALAR(FRAME(1,I), XYZDD(1,J+1), TRANS(J,I))
36 CONTINUE
      GASH = T11
      T11 = T22/AREA
      T22 = GASH/AREA
      T12 = -T12/AREA
      T21 = -T21/AREA

```

```

C
C**** TRANSFORM WCORN AND WLOOF INTO LOCAL X,Y DERIVATIVES.
C
      DO 41 N = 1, LNODZA
      DO 41 I = 1, 2
      GASH = 0.0
      GISH = 0.0
      DO 40 J = 1, 2
      GASH = GASH + TRANS(I,J)*WCORN(N+11,J)
40    GISH = GISH + TRANS(I,J)*WLOOF(N+11,J)
      WCORN(N,I+1) = GASH
41    WLOOF(N,I+1) = GISH
C
C**** PUT THICKNESS AND DERIVATIVES INTO LOCAL COORDINATE SYSTEM.
C
      DO 45 I = 1, 3
      DO 44 J = 1, 2
      POINT(J) = 0.0
      DO 44 K = 1, 2
44    POINT(J) = POINT(J) + TRANS(J,K)*THIKDD(I,K+1)
      DO 45 J = 1, 2
45    THIKDD(I,J+1) = POINT(J)
      DO 48 J = 1, 3
      DO 47 I = 1, 3
      CALL SCALAR(THIKDD(1,J), FRAME(1,I), POINT(I))
47    CONTINUE
      DO 48 I = 1, 3
48    THIKDD(I,J) = POINT(I)
      THIK = THIKDD(3,1)
      N ERROR = 10
      IF(THIK.LE.0.0) GO TO 99
C
C**** FIND THE CHANGE IN LOCAL X,Y DERIVATIVES ACROSS THICKNESS OF SHELL.
C
      DO 57 LNOD = 1, LNODZA
      IF(NSTAGE.NE.4) GO TO 51
      DO 50 I = 2, 3
      GASH = 0.0
      DO 49 J = 1, 2
49    GASH = GASH + THIKDD(J,I)*WCORN(LNOD,J+1)
50    POINT(I) = GASH
C
C**** CREATE WSHEL = SHAPE FUNCTION ARRAY, DISPLACEMENT TERMS FIRST.
C
51    KORN = (LNOD+1)/2
      DO 54 K = 1, 3
      KOL = 2*KORN + 3*LNOD + K - 5
      IF(LNOD.GT.LNODZ) KOL = 5*LNODZ + 2 + K
      DO 53 N = 1, 3
      FACT = FRAME(K,N)
      WSHEL(N,KOL) = WCORN(LNOD,1)*FACT
      IF(NSTAGE.EQ.4. AND .N.EQ.3) FACT = 0.0
      DO 53 ND = 2, 3
      N3=N+N+ND
53    WSHEL(N3,KOL)=WCORN(LNOD,ND)*FACT

```

```

DO 54 N = 1,2
DO 54 ND = 2,3
WSHEL(N+7,KOL) = WSHEL(N+7,KOL)
N3=N+N+ND
1 -THIKDD(ND-1,1)*WSHEL(N3,KOL)/THIK
N4=N+N+ND+6
IF(NSTAGE.EQ.4) WSHEL(N4,KOL)=(POINT(ND)*FRAME(K,N)
1 + THIKDD(3,ND)*WCORN(LNOD,N+1)*FRAME(K,3))/THIK
54 CONTINUE
C
C**** INTRODUCE ROTATION TERMS WITH BENDING ACTION INTO WSHEL.
C
DO 57 L = 1,2
KOL = (L-1)*4*LNODZ + (2-L)*6*KORN + LNOD
IF(LNOD.GT.LNODZ) KOL = 5*LNODZ + 3 - L
DO 56 N = 1,2
LNOD1=4*LNOD+L-4
CALL SCALAR(VLOOP(1, LNOD1), FRAME(1,N), FACT)
WSHEL(N+7,KOL) = FACT*WLOOP(LNOD,1)/THIK
IF(NSTAGE.NE.4) GO TO 56
DO 55 ND = 2,3
N4=N+N+ND+6
55 WSHEL(N4,KOL)=FACT*WLOOP(LNOD,ND)/THIK
56 CONTINUE
DO 57 NROW = 1,7
57 WSHEL(NROW,KOL) = 0.0
C
C**** COMBINE LAST THREE COLUMNS OF WSHEL TO CREATE NORMAL DEFLECTION.
C
IF(LNODZ.EQ.6) GO TO 61
IZ = 3*NSTAGE + 1
DO 60 I = 1, IZ
GASH = 0.0
DO 59 K = 1,3
LNODZ2=4*LNODZ+4
59 GASH=GASH+WSHEL(I,K+42)*VLOOP(K, LNODZ2)
60 WSHEL(I,43) = GASH
61 IF(NSTAGE.EQ.4) GO TO 86
C
C**** CREATE ARRAY SHEAR, FOR INTRODUCING THE CONSTRAINTS (NSTAGE = 3)
C
IF(NLOOP.GT.LNODZ) GO TO 63
DO 62 I = 1, LVABZZ
SHEAR(NLOOP,I) = WSHEL(9,I)
SHEAR(11,I) = SHEAR(11,I) + WSHEL(8,I)*SIDE*THIK*SWOP((NLOOP+1)/2)
62 CONTINUE
GO TO 67
63 DO 66 KOL = 1, LVABZZ
DO 66 NXY = 1,2
LNODZ3=LNODZ+NXY
GASH=SHEAR(LNODZ3,KOL)
DO 65 MXY = 1,2
LNODZ4=4*LNODZ+NXY
CALL SCALAR(FRAME(1,MXY),VLOOP(1, LNODZ4), FACT)
65 GASH = GASH + WSHEL(MXY+7,KOL)*AREA*THIK*FACT
LNODZ3=LNODZ+NXY
66 SHEAR(LNODZ3,KOL)=GASH

```

```

C
C**** COMPLETE LOOP AROUND LOOF NODES ETC. TO CREATA VLOOF OR SHEAR.
C
  67 IF(NLOOF.LE.LNODZ. OR.
  1   (NSTAGE.EQ.3. AND .NLOOF.LT.(3*LNODZ)/2)) GO TO 16
  IF(NSTAGE.NE.2) GO TO 76
C
C**** CREATE PLUS-MINUS SUM OF THIKNESS VECTORS AT LOOF NODES (NSTAGE=2)
C
  DO 70 I = 1,3
  GASH = 0.0
  DO 68 N = 3,LVABZ,4
  68 GASH = -GASH + VLOOF(I,N)
  SIGT(I) = GASH
C
C**** AND THE 3X3 MATRIX ASSOCIATED WITH IT, STORED IN XYZDD.
C
  DO 70 J = 1,3
  GASH = 0.0
  IF(I.EQ.J) GASH = FLOAT(LNODZ)
  DO 69 N = 2,LVABZ,4
  69 GASH = GASH -VLOOF(I,N)*VLOOF(J,N)
  70 XYZDD(I,J) = GASH
C
C**** GET THE ADJUGATE OF THIS 3X3 SYMMETRIC POSITIVE DEFINITE MATRIX.
C
  K = 3
  DO 71 I = 1,3
  K1=6-I-K
  CALL VECTOR(XYZDD(1,I),XYZDD(1,K1),FRAME(1,K))
  71 K = I
  CALL SCALAR(XYZDD(1,1), FRAME(1,1), DETERM)
  DO 73 I = 1,3
  CALL SCALAR(FRAME(1,I), SIGT(1), PROD)
  73 POINT(I) = PROD/DETERM
C
C**** CORRECT VECTOR THICKNESS IN VLOOF.
C
  FACT = 1.0
  DO 75 N = 2,LVABZ,4
  FACT = -FACT
  CALL SCALAR(POINT(1), VLOOF(1,N), PROD)
  DO 74 I = 1,3
  74 VLOOF(I,N+1) = VLOOF(I,N+1) - FACT*(POINT(I)-PROD*VLOOF(I,N))
C
C**** CREATE DIFFERENTIAL DISPLACEMENT VECTORS TO DEFINE ROTATIONS.
C
C**** THIS COMPLFTES WORK FOR NSTAGE = 2
C
  TFIRST = VLOOF(1,N-1)
  CALL VECTOR(VLOOF(1,N), VLOOF(1,N+1), VLOOF(1,N-1))
  DO 75 I = 1,3
  75 VLOOF(I,N) = VLOOF(I,N)*TFIRST
  NSTAGE = 3
  GO TO 15

```

```

C
C**** SHEAR HAS BEEN CREATED IN NLOOF LOOP FOR NSTAGE = 3.
C
C**** CHOOSE PIVOT FOR REDUCING ARRAY SHEAR, AND DO ROW INTERCHANGE.
C
  76 CONTINUE
    DO 83 LIM = 1,LIMZ
      KP = LVABZ + LIM
      PIVOT = 0.0
      DO 79 L = LIM,LIMZ
        IF(ABS(PIVOT).GT.ABS(SHEAR(L,KP))) GO TO 79
        LBIG = L
        PIVOT = SHEAR(LBIG,KP)
      79 CONTINUE
      DO 80 K = 1,LVABZZ
        CHANGE = SHEAR(LBIG,K)
        SHEAR(LBIG,K) = SHEAR(LIM,K)
      80 SHEAR(LIM,K) = CHANGE/PIVOT
C
C**** REDUCE ARRAY SHEAR TO CREATE CONSTRAINT MATRIX.
C
C**** THIS COMPLETES WORK FOR NSTAGE = 3.
C
    DO 82 NROW = 1,LIMZ
      FACT = SHEAR(NROW,KP)
      IF(NROW.EQ.LIM. OR .FACT.EQ.0.0) GO TO 82
      DO 81 KOL = 1,LVABZZ
      81 SHEAR(NROW,KOL) = SHEAR(NROW,KOL) - FACT*SHEAR(LIM,KOL)
      82 CONTINUE
      83 CONTINUE
      NSTAGE = 4
      GO TO 18
C
C**** USE ARRAY SHEAR TO CONSTRAIN WSHEL AT THE GIVEN POINT XI,ETA.
C
    86 DO 88 I = 1,LVABZ
      DO 88 J = 1,13
        GASH = WSHEL(J,I)
        DO 87 K = 1,LIMZ
          LVABZK=K+LVABZ
          87 GASH=GASH-WSHEL(J,LVABZK)*SHEAR(K,I)
        88 WSHEL(J,I) = GASH
C
C**** IMPLEMENT SWOP TO EXCHANGE TWO NORMAL SLOPES.
C
    DO 92 N = 8,LVARZ, 8
      IF(SWOP(N/8).EQ.1.0) GO TO 92
      DO 91 J = 1,13
        CHANGE = WSHEL(J,N)
        WSHEL(J,N) = WSHEL(J,N-1)
      91 WSHEL(J,N-1) = CHANGE
      92 CONTINUE
C
C**** ASSEMBLE UXZ, UYZ, VXZ, VYZ TO CREATE WXX, WXY, WYY.
C
    DO 96 N = 1,LVARZ
      WSHEL(10,N) = -WSHEL(10,N)
      WSHEL(11,N) = -0.5*(WSHEL(11,N)+WSHEL(12,N))
      WSHEL(12,N) = -WSHEL(13,N)
      WRITE(6,640) N, (WSHEL(J,N). J = 1,12)
    96 CONTINUE

```

```

C
C**** PUT POINT, FRAM IN COMMON, ALSO AREA, SIDE WITH INTEGRATING FACTORS
C
  AREA = AREA*(FLOAT(LNODZ)-5.6)/2.4
  SIDE = SIDE*FLOAT(LNODZ-4)/4.0
  DO 98 I = 1,3
  POINT(I) = XYZDD(I,1) + ELXYZT(9,I)
  DO 98 J = 1,3
 98 FRAM(I,J) = FRAME(I,J)
  RETURN

C
C**** WRITE DIAGNOSTIC ERROR MESSAGE.
C
 99 WRITE(6,699) NERROR
699 FORMAT(/6H ERROR,15,18H IN SEGMENT HALCOF)
  STOP
  END
  SUBROUTINE VECTOR(U, V, W)

C
C**** TO COMPUTE VECTOR PRODUCT U*V INTO AREA W.
C
  DIMENSION U(3), V(3), W(3)
  K = 3
  DO 2 I = 1,3
  K3=6-I-K
  W(K3)=U(K)*V(I)-U(I)*V(K)
 2 K = I
  RETURN
  END
  SUBROUTINE SCALAR(U, V, PROD)

C
C**** TO COMPUTE SCALAR PRODUCT OF VECTORS U AND V.
C
  DIMENSION U(3), V(3)
  PROD = 0.0
  DO 2 I = 1,3
 2 PROD = PROD + U(I)*V(I)
  RETURN
  END
  SUBROUTINE SFR(XLOCAL, WCORN,WLOOF, NSTAGE)

C
C***** SHAPE FUNTION SUBROUTINE TO SERVE HAOOF.
C
  DIMENSION MD(4), TERMV(46), WCORN(10,3), WLOOF(10,3), XLOCAL(2)
  COMMON/FRON/ELST(528),ELR(32,2),LNODS(8),LNODZ
  COMMON/SHELL/AREA, ELXYZT(9,4), FRAM(3,3), POINT(3),
 1  SIDE, THIK, WSHFL(13,45), XITA(2), XYZ(241,3)
  COMMON/COEF/COEF(247)
  EQUIVALENCE (WSHFL(1,1), TERMV(1))
  DATA MD/8, 43, 90, 171/

C
C**** INITIALIZE AND PREPARE TO CALCULATE TERNV = POLYNOMIAL TERMS.
C
  XI = XLOCAL(1)
  ETA = XLOCAL(2)
  IF(ABS(XI).GT.1.0. OR .ABS(ETA).GT.1.0. OR
 1  .(LNODZ.EQ.6. AND .(XI.LT.0.0.OR.ETA*(1.0-XI-ETA).LT.0.0)))
 2  GO TO 99
  IA = 2

```

C
C**** CREATE POLYNOMIAL TERMS AND XI, ETA DERIVATIVES.

C
 TERMV(1) = 0.0
 TERMV(2) = 1.0
 NZ = (LNODZ+NSTAGE-3)/2
 DO 6 N = 1, NZ
 IAN = IA + N
 N2 = N + 15
 N3 = N + 30
 DO 4 J = IA, IAN
 TERMV(J+N) = TERMV(J)*XI
 TERMV(J+N2) = TERMV(J)*FLOAT(IAN-J)
 4 TERMV(J+N3) = TERMV(J-1)*FLOAT(J-IA)
 IA = IAN
 6 TERMV(IA+N) = TERMV(IA-1)*ETA

C
C**** CREATE SPECIAL COMBINATIONS FOR LOOF NODES, ETC.

C
 DO 8 I = 8, 38, 15
 IF(LNODZ.EQ.6) TERMV(I) =
 1 2.0*(TERMV(I)-TERMV(I+3)) + 3.0*(TERMV(I+1)-TERMV(I+2))
 IF(LNODZ.EQ.8) TERMV(I) = TERMV(I+2)
 IF(LNODZ.EQ.8) TERMV(I+2) = TERMV(I+6)
 8 CONTINUE

C
C**** USE TERMV TO FIND WCORN AND WLOOF AND XI, ETA DERIVATIVES.

C
 NFOISZ = (NSTAGE+1)/2
 DO 18 NFOIS = 1, NFOISZ
 NZ = (3*LNODZ)/2 + NFOIS - 4
 IF(NZ.NE.10) GO TO 12
 NZ = 9
 DO 10 I = 10, 40, 15
 10 TERMV(I) = TERMV(I+3) - TERMV(I+5)
 12 K = 0
 DO 18 I = 1, 3
 DO 16 N = 1, NZ
 GASH = 0.0
 LNODZ6 = LNODZ + NFOIS - 6
 MDEL = MD(LNODZ6) + N * NZ - 15 * I
 MA = 16 * I - 14
 MZ = 15 * I + NZ - 14
 DO 14 M = MA, MZ
 MMDEL = M + MDEL
 14 GASH = GASH + TERMV(M) * COEF(MMDEL)
 NK = N + K
 IF(NFOIS.EQ.1) WCORN(NK, I) = GASH
 NK = N + K
 IF(NFOIS.EQ.2) WLOOF(NK, I) = GASH
 16 CONTINUE
 18 K = 1
 RETURN

C
C**** ERROR DIAGNOSTICS, IF POINT LIES OUTSIDE ELEMENT.

C
 99 WRITE(6,610) XI, ETA
 610 FORMAT(/30H ERROR 11 IN SEGMENT SFR, XI =,F15.9,3X,SHETA =,F15.9)
 STOP
 END
 FINISH

LIST OF PRINCIPAL SYMBOLS

D	flexural rigidity = $Eh^3/12(1 - \nu^2)$
E	Young's modulus
k	$h^2/12R^2$
h	shell thickness
R	radius
U_x, U_y, U_z	components of displacement acting in the x, y, z direction
U_θ, U_ϕ	components of displacement acting in the θ, ϕ direction
W	component of displacement normal to the shell surface
$\epsilon_{\phi\phi}, \epsilon_{\theta\theta}, \epsilon_{\theta\phi}$	components of strain
$\kappa_{\phi\phi}, \kappa_{\theta\theta}, \bar{\kappa}_{\phi\theta}$	components of curvature change
ν	Poisson's ratio
$\omega_x, \omega_y, \omega_z$	components of rotation in the x, y, z directions
$\omega_\theta, \omega_\phi$	components of strain acting in the θ, ϕ directions
θ	spherical polar coordinate measured in the xy plane,
ϕ	spherical polar coordinate measured from the z axis to the xy plane

REFERENCES

<u>No.</u>	<u>Author</u>	<u>Title, etc</u>
1		MSC/NASTRAN users manual Vol 1. The MacNeal Scwendler corporation (1976)
2	L.S.D. Morley A.J. Morris	Conflict between finite elements and shell theory. In proc of 2nd World conference on finite elements. Bournemouth (1978) edited by J. Robinson

REPORTS QUOTED ARE NOT NECESSARILY
AVAILABLE TO MEMBERS OF THE INSTITUTE
OR TO COMMERCIAL ORGANISATIONS

```

LIST
PROGRAM(DATA)
INPUT1=CRO
OUTPUT2=LPO
OUTPUT3=CPQ
END
MASTER GENDATA
C**** PROGRAM TO GENERATE X,Y,Z COORDINATES ON THE SURFACE OF A SPHERE
C**** GIVEN THE CORRESPONDING THETA AND PHI COORDINATES.
C**** R=RADIUS
C**** NNODE=TOTAL NUMBER OF NODES
REAL X(241),Y(241),Z(241)
R=10.0
PI=3.1415926536
PIF=PI/180.0
READ(1,100)NNODE
DO 50 INODE=1,NNODE
READ(1,101)NODE,THE,PHI
IF(NODE.NE.INODE)WRITE(2,200)NODE,(NODE
C**** CONVERT THETA AND PHI TO RADIANS
THE=THE*PIF
PHI=PHI*PIF
X(NODE)=R*COS(THE)*SIN(PHI)
Y(NODE)=R*SIN(THE)*SIN(PHI)
Z(NODE)=R*COS(PHI)
50 CONTINUE
C**** READ IN AND CALCULATE MIDSIDE NODES NOT ALREADY CALCULATED
C**** NODE LIES HALFWAY ALONG THE GEODESIC LINE BETWEEN NOA AND NOB
60 READ(1,102,END=70)NODE,NOA,NOB
XFAC=(X(NO A)+X(NO B))**2
YFAC=(Y(NO A)+Y(NO B))**2
ZFAC=(Z(NO A)+Z(NO B))**2
X(NODE)=R*SQRT(XFAC/(XFAC+YFAC+ZFAC))
Y(NODE)=R*SQRT(YFAC/(XFAC+YFAC+ZFAC))
Z(NODE)=R*SQRT(ZFAC/(XFAC+YFAC+ZFAC))
GOTO 60
70 CONTINUE
DO 80 INODE=1,NNODE
WRITE(3,300)INODE,X(INODE),Y(INODE),Z(INODE)
80 CONTINUE
STOP
100 FORMAT(IU)
101 FORMAT(I0,3F0.0)
102 FORMAT(3I0)
200 FORMAT(1H0,'NODE',I4,' = NOD.',I4)
300 FORMAT(I4,3F15.10)
END
FINISH

```

Fig 1 cont'd

```

LIST
PROGRAM(DATA)
INPUT1=CR0
OUTPUT3=CP0
OUTPUT4=CP1
END
MASTER GENPATCH
C**** PROGRAM TO GENERATE X,Y,Z COORDINATES FOR A 3*3 PATCH OF
C**** RECTANGULAR ELEMENTS ON A SPHERICAL SURFACE.
C**** THE INPUT DATA IS
C**** ITYPE ICASE THEINC PHIINC THEZERO PHIZERO
C**** ITYPE=1 FOR SEMILOOP.
C**** ITYPE=2 FOR NASTPAN.
C**** ICASE=0 FOR PRESSURE LOADING
C**** ICASE=1 FOR KTHETHE
C**** ICASE=2 FOR KTHETHE,KPHIPHI
C**** ICASE=3 FOR ETHETHE
C**** ICASE=4 FOR ETHETHE,EPHIPHI
C**** THEINC=DEGREES SUBTENDED BY ELEMENT IN THETA DIRECTION.
C**** PHIINC=DEGREES SUBTENDED BY ELEMENT IN PHI DIRECTION.
C**** THEZERO=THETA COORDINATE AT CENTRE OF PATCH.
C**** PHIZERO=PHI COORDINATE AT CENTRE OF PATCH.
C**** R=RADIUS OF PATCH.
C**** ARRAY N CONTAINS SEMILOOP NODE NUMBERS AROUND BOUNDARY
C**** ARRAY M CONTAINS SEMILOOP NASTSIDE NODE NUMBERS AROUND BOUNDARY
C**** ARRAY NAST CONTAINS NASTRAN NODE NUMBERS AROUND BOUNDARY
C**** ARRAY NASTSIDE CONTAINS THE SIDE NUMBERS FOR THE NODES IN NAST
C**** A CORNER OF THE NASTRAN PATCH HAS SIDE=5
REAL X(40),Y(40),Z(40),ATHE(40),APHI(40)
INTEGER N(12),M(12),NAST(12),NASTSIDE(12)
DATA N/1,3,5,7,12,13,23,29,34,36,38,40/
DATA M/2,4,6,11,22,33,39,37,35,30,19,8/
DATA NAST/1,2,3,4,5,8,9,12,13,14,15,16/
DATA NASTSIDE/5,1,1,5,4,2,4,2,5,3,3,5/
R=10.0
PI=3.1415926536
PIF=PI/180.0
READ(1,101)ITYPE,ICASE,THEINC,PHIINC,THEZERO,PHIZERO
C**** CONVERT TO RADIANS
THEINC=THEINC*PIF
PHIINC=PHIINC*PIF
THEZERO=THEZERO*PIF
PHIZERO=PHIZERO*PIF
C****
C**** GOTO(0,40),ITYPE
C****

```

```

C**** CALCULATE COORDINATES FOR SEMICIRC OF PATCH
PHI=PHIZERO+1.5*PHIINC
NODE=1
DO 30 I=1,4
THE=THEZERO-1.5*THEINC
DO 10 J=1,7
X(NODE)=R*COS(THE)*SIN(PHI)
Y(NODE)=R*SIN(THE)*SIN(PHI)
Z(NODE)=R*COS(PHI)
ATHE(NODE)=THE/PIF
APHI(NODE)=PHI/PIF
NODE=NODE+1
THE=THE+THEINC/2.0
10 CONTINUE
IF(I.EQ.4)GOTO 30
THE=THEZERO-1.5*THEINC
PHI=PHI-0.5*PHIINC
DO 20 J=1,4
X(NODE)=R*COS(THE)*SIN(PHI)
Y(NODE)=R*SIN(THE)*SIN(PHI)
Z(NODE)=R*COS(PHI)
ATHE(NODE)=THE/PIF
APHI(NODE)=PHI/PIF
NODE=NODE+1
THE=THE+THEINC
20 CONTINUE
PHI=PHI-0.5*PHIINC
30 CONTINUE
WRITE(3,300)(NODE,X(NODE),Y(NODE),Z(NODE),NODE=1,40)
WRITE(4,400)ICASE
C**** PRINT NODE NUMBER,THETA,PHI FOR EACH BOUNDARY NODE.
DO 32 I=1,12
NODE=N(I)
32 WRITE(4,400)NODE,ATHE(NODE),APHI(NODE)
WRITE(4,401)
C**** PRINT NODE,THETA,PHI,SIDE NUMBER FOR EACH BOUNDARY MIDSIDE NODE
K=1
DO 34 I=1,4
DO 34 J=1,3
MID=M(K)
WRITE(4,400)MID,ATHE(MID),APHI(MID),I
34 K=K+1
STOP

```

Fig 1 cont'd

```
C**** CALCULATE COORDINATES FOR N, STRAN PATCH
4 PHI=PHIZERO+1.5*PHIINC
  NODE=1
  DO 60 I=1,4
    THE=THEZERO-1.5*THEINC
    DO 50 J=1,4
      X(NODE)=R*COS(THE)*SIN(PHI)
      Y(NODE)=R*SIN(THE)*SIN(PHI)
      Z(NODE)=R*COS(PHI)
      ATHE(NODE)=THE/PIF
      APHI(NODE)=PHI/PIF
      NODE=NODE+1
    THE=THE+THEINC
  50 CONTINUE
  PHI=PHI-PHIINC
  60 CONTINUE
C**** PRINT NODE X Y Z AT EACH NODE
  WRITE(3,302)(INODE,X(INODE),Y(INODE),Z(INODE),INODE=1,16)
C**** PRINT ICASE
  WRITE(4,402)ICASE
C**** PRINT NODE, THETA, PHI AND SIDE NUMBER FOR EACH BOUNDARY NODE.
  DO 70 I=1,12
    NODE=NAST(I)
    ISIDE=NASTSIDE(I)
    WRITE(4,400)NODE,ATHE(NODE),APHI(NODE),ISIDE
  70 CONTINUE
  STOP
100 FORMAT(2I0,4F0.1)
300 FORMAT(14,3F15.10)
302 FORMAT('GRID',4X,13,8X,3F8.5)
400 FORMAT(14,2F10.2,14)
401 FORMAT('U 0.0 0.0')
402 FORMAT(I1)
  END
  FINISH
```

```

LIST
PROGRAM(DATA)
INPUT1=CR0
OUTPUT3=CP0
OUTPUT4=CP1
END
MASTER GENTRIPAT
C**** THIS PROGRAM GENERATES DATA FOR A PATCH OF 18 TRIANGULAR
C**** SEMILOOF ELEMENTS ON THE SURFACE OF A SPHERE.
C**** THE INPUT DATA IS
C**** ITYPE ICASE THEINC PHIINC THEZERO PHIZERO
C**** WHERE ITYPE IS DUMMY
C**** ICASE IS THE BOUNDARY DISPLACEMENT TYPE 0,1,2,3 OR 4
C**** THEINC IS THE INCREMENT PER ELEMENT IN THE THETA DIRECTION
C**** PHIINC IS THE INCREMENT PER ELEMENT IN THE PHI DIRECTION
C**** THEZERO IS THE THETA COORDINATE AT THE CENTRE OF THE PATCH
C**** PHIZERO IS THE PHI COORDINATE AT THE CENTRE OF THE PATCH
C**** NODE X Y Z ARE WRITTEN TO CP FILE3 FOR EACH NODE AND MIDSIDE NODE
C**** NODE THE PHI FOR EACH NODE ON THE BOUNDARY
C**** MID THE PHI ISIDE FOR EACH MIDSIDE NODE ON THE BOUNDARY
C**** ARE WRITTEN TO CP FILE4 FOR INPUT TO THE SEMILOOF PROGRAM
      REAL X(49),Y(49),Z(49),ATHE(49),APHI(49)
      INTEGER N(12),M(12),NAST(12)
C**** N CONTAINS THE VERTEX NODES AROUND THE BOUNDARY OF THE SEMILOOF PATCH
C**** M CONTAINS THE MIDSIDE NODES AROUND THE BOUNDARY OF THE SEMILOOF PATCH
      DATA N/1,3,5,7,15,21,29,35,43,45,47,49/
      DATA M/2,4,6,14,28,42,43,46,44,36,22,8/
      R=10.0
      PI=3.1415926536
      PIF=PI/180.0
      READ(1,101)ITYPE,ICASE,THEINC,PHIINC,THEZERO,PHIZERO
C**** CONVERT TO RADIANS
      THEINC=THEINC*PIF
      PHIINC=PHIINC*PIF
      THEZERO=THEZERO*PIF
      PHIZERO=PHIZERO*PIF
C**** CALCULATE COORDINATES FOR SEMILOOF PATCH

```

Fig 1 cont'd

```
PHI=PHIZERO+1.5*PHIINC
NODE=1
DO 30 I=1,7
THE=THEZERO-1.5*THEINC
DO 10 J=1,7
X(NODE)=R*COS(THE)*SIN(PHI)
Y(NODE)=R*SIN(THE)*SIN(PHI)
Z(NODE)=R*COS(PHI)
ATHE(NODE)=THE/PIF
APHI(NODE)=PHI/PIF
NODE=NODE+1
THE=THE+THEINC/2.0
10 CONTINUE
THE=THEZERO-1.5*THEINC
PHI=PHI+0.5*PHIINC
30 CONTINUE
C**** PRINT RESULTS
WRITE(3,300)(NODE,X(NODE),Y(NODE),Z(NODE),NODE=1,49)

WRITE(4,400)ICASE
DO 32 I=1,12
NODE=N(I)
32 WRITE(4,400)NODE,ATHE(NODE),APHI(NODE)
WRITE(4,401)
K=1
DO 34 I=1,4
DO 34 J=1,3
MID=M(K)
WRITE(4,400)MID,ATHE(MID),APHI(MID),I
34 K=K+1
STOP
100 FORMAT(2I0,4F0.0)
300 FORMAT(14,3F15.10)
400 FORMAT(14,2F10.2,14)
401 FORMAT(10,0.0,0.0,0.0)
END
FINISH
```

```

LIST
PROGRAM(DATA)
INPUT1=CR0
OUTPUT3=CP0
OUTPUT4=CP1
END
MASTER GENPAT
C**** THIS PROGRAM CALCULATES THE X Y Z COORDINATES FOR A 3 BY 3 PATCH
C**** OF SEMILOOF OR NASTRAN QUADRILATERAL ELEMENTS
C**** WITH A REGULAR BOUNDARY BUT WITH IRREGULAR ELEMENTS IN THE INTERIOR.
C**** THE INPUT DATA IS
C**** ITYPE ICASE THEINC PHIINC THEZERO PHIZERO
C**** THEI PHII
C****          (I=14,16,25,27) FOR SEMILOOF OR (I=6,7,10,11) FOR NASTRAN
C**** WHERE ITYPE IS 1 FOR SEMILOOF OR 2 FOR NASTRAN DATA
C****          ICASE IS THE BOUNDARY DISPLACEMENT TYPE 0,1,2,3 OR 4
C****          THEINC IS THE ANGLE SUBTENDED PER ELEMENT IN THE THETA DIRECTION
C****          PHIINC IS THE ANGLE SUBTENDED PER ELEMENT IN THE PHI DIRECTION
C****          THEZERO IS THE THETA COORDINATE AT THE CENTRE OF THE PATCH
C****          PHIZERO IS THE PHI COORDINATE AT THE CENTRE OF THE PATCH
C****          THEI AND PHII ARE THE THETA AND PHI COORDINATES OF THE VERTEX
C**** NODES AROUND THE CENTRAL ELEMENT.
REAL X(40),Y(40),Z(40),ATHE(40),APHI(40)
INTEGER N(12),M(12),MID(36),NAST(12),NASTSIDE(12)
DATA N/1,3,5,7,12,13,23,29,34,36,38,40/
DATA M/2,4,6,11,22,33,39,37,35,30,19,8/
DATA MID/9,3,14,10,5,16,13,12,14,15,14,16,17,16,18,20,14,25,
1 21,16,27,24,23,25,26,25,27,28,27,29,31,25,36,37,27,38/
DATA NAST/1,2,3,4,5,8,9,12,13,14,15,16/
DATA NASTSIDE/5,1,1,5,4,2,4,2,5,3,3,5/
R=10.0
PI=3.1415926536
PIF=PI/180.0
READ(1,101)ITYPE,ICASE,THEINC,PHIINC,THEZERO,PHIZERO
C**** CALCULATE COORDINATES FOR SEMILOOF PATCH
C****
GOTO(0,75),ITYPE

```

Fig 1 cont'd

```

C****
      PHI=PHIZERO+1.5*PHIINC
      NODE=1
      DO 30 I=1,4
      THE=THEZERO-1.5*THEINC
      DO 10 J=1,7
      ATHE(NODE)=THE
      APHI(NODE)=PHI
      NODE=NODE+1
      THE=THE+THEINC/2.0
10    CONTINUE
      IF(I.EQ.4)GOTO 30
      THE=THEZERO-1.5*THEINC
      PHI=PHI-0.5*PHIINC
      DO 20 J=1,4
      ATHE(NODE)=THE
      APHI(NODE)=PHI
      NODE=NODE+1
      THE=THE+THEINC
20    CONTINUE
      PHI=PHI-0.5*PHIINC
30    CONTINUE
C**** READ IN THE THETA AND PHI COORDINATES OF THE NODES OF THE CENTRAL ELEMENT
      READ(1,102)ATHE(14),APHI(14)
      READ(1,102)ATHE(16),APHI(16)
      READ(1,102)ATHE(25),APHI(25)
      READ(1,102)ATHE(27),APHI(27)
      DO 40 I=1,34,3
      ATHE(MID(I))=(ATHE(MID(I+1))+ATHE(MID(I+2)))/2.0
40    APHI(MID(I))=(APHI(MID(I+1))+APHI(MID(I+2)))/2.0
      DO 50 I=1,40
      X(I)=R*COS(ATHE(I)*PIF)*SIN(APHI(I)*PIF)
      Y(I)=R*SIN(ATHE(I)*PIF)*SIN(APHI(I)*PIF)
50    Z(I)=R*COS(APHI(I)*PIF)
C**** PRINT RESULTS
      WRITE(3,300)(NODE,X(NODE),Y(NODE),Z(NODE),NODE=1,40)
      WRITE(4,400)ICASE
      DO 60 I=1,12
      NODE=N(I)
60    WRITE(4,400)NODE,ATHE(NODE),APHI(NODE)
      WRITE(4,401)
      K=1
      DO 70 I=1,4
      DO 70 J=1,5
      MI=M(K)
      WRITE(4,400)MI,ATHE(MI),APHI(MI),I
70    K=K+1
      STOP

```

```

C**** CALCULATE COORDINATES FOR NASTRAN PATCH
75 PHI=PHIZERO+1.5*PHIINC
   NODE=1
   DO 85 I=1,4
     THE=THEZERO-1.5*THEINC
     DO 80 J=1,4
       ATHE(NODE)=THE
       APhi(NODE)=PHI
       NODE=NODE+1
       THE=THE+THEINC
     80 CONTINUE
     PHI=PHI+PHIINC
   85 CONTINUE
   READ(1,102)ATHE(6),APHI(6)
   READ(1,102)ATHE(7),APHI(7)
   READ(1,102)ATHE(10),APHI(10)
   READ(1,102)ATHE(11),APHI(11)
   DO 90 I=1,12
     J=NAST(I)
   90 WRITE(4,400)J,ATHE(J),APHI(J),NASTSIDE(I)
C**** CALCULATE AND PRINT THE X,Y,Z COORDINATES
   DO 95 I=1,14
     X(I)=R*COS(ATHE(I)*PIF)*SIN(APHI(I)*PIF)
     Y(I)=R*SIN(ATHE(I)*PIF)*SIN(APHI(I)*PIF)
     Z(I)=R*COS(APHI(I)*PIF)
   95 WRITE(3,302)I,X(I),Y(I),Z(I)
   STUP
101 FORMAT(2I0,4F0.0)
102 FORMAT(2F0.0)
300 FORMAT(I4,3F15.10)
302 FORMAT('GRID',4X,I8,8X,3F8.6)
400 FORMAT(I4,2F10.2,I4)
401 FORMAT('0 0.0 0.0')
   END
   FINISH

```

Fig 1 cont'd

```
LIST
PROGRAM(NAST)
INPUT1=CR0
OUTPUT3=CP0
OUTPUT4=CP1
OUTPUT5=CP2
END
MASTER GENVASTB:
C**** THIS PROGRAM GENERATES THE BOUNDARY DISPLACEMENTS FOR A
C**** NASTRAN PATCH. U1, U2, U3, U4, U5, U6 ARE PRESCRIBED AT EACH NODE
C**** ON THE BOUNDARY IN THE FORM OF SPC OR MPC CARDS.
C**** ICASE IS READ IN AND THE APPROPRIATE SUBROUTINE IS CALLED.
C**** ICASE=0 PRESSURE LOADING
C**** ICASE=1 INEXTENSIONAL BENDING, KAPPA PHI THETA CONSTANT
C**** ICASE=2 INEXTENSIONAL BENDING, KAPPA PHI PHI, -KAPPA THE THE CONSTANT
C**** ICASE=3 MEMBRANE BEHAVIOUR, EPSILON PHI THETA CONSTANT
C**** ICASE=4 MEMBRANE BEHAVIOUR, EPSILON PHI PHI, -EPSILON THE THE CONSTANT
C**** R=RADIUS
C**** H=THICKNESS
C**** E=YOUNGS MODULUS
C**** PNU=POISSONS RATIO
C**** NOD=NUMBER OF NODES ON BOUNDARY
C**** IP=TOTAL NUMBER OF NODES IN PATCH
C**** PIF=PI/180 = MULTIPLICATION FACTOR TO CONVERT FROM DEGRESS TO RADIANS
C****
R=10.0
H=0.04
E=10.0E6
PNU=0.3
NOD=12
IP=16
PIF=3.1415926535/180.0
READ(1,100)ICASE
IF(ICASE.EQ.0)CALL PRESSURE(R,NOD,PIF)
IF(ICASE.EQ.1)CALL KTHETHEPHI(0,4,E,PNU,NOD,IP,PIF)
IF(ICASE.EQ.2)CALL KTHETHE(0,NOD,IP,PIF)
IF(ICASE.EQ.3)CALL ETHETHEPHI(0,NOD,IP,PIF)
IF(ICASE.EQ.4)CALL ETHETHE(0,NOD,IP,PIF)
STOP
100 FORMAT(I1)
END
SUBROUTINE PRESSURE(R,NOD,PIF)
```

```

C**** ICASE = 0
C**** THIS SUBROUTINE GENERATES THE BOUNDARY CONDITIONS FOR A 3 BY 3 PATCH
C**** OF NASTRAN QUAD4 ELEMENTS UNDER PRESSURE LOADING
C**** THE INPUT DATA CAN BE OBTAINED FROM PROGRAM GENPATCH
C**** THE OUTPUT IS UX,UY,UZ,FX,WY,WZ FOR EACH NODE AND IS WRITTEN
C**** TO CP FILE3 AS 1PC CARDS
      DO 50 II=1,NOD
      READ(1,101)I,THE,PHI,ISIDE
C**** ISIDE=5 REFERS TO A NODE ON THE CORNER OF THE PATCH
      THE=THE+PIF
      PHI=PHI+PIF
      GO TO(13,24,13,24,13),ISIDE
    13 CONTINUE
C**** SIDES 1 AND 3
      AX=-COS(PHI)*COS(THE)
      AY=-COS(PHI)*SIN(THE)
      AZ=SIN(PHI)
      RX=-SIN(THE)
      RY=COS(THE)
      RZ=0.
      WRITE(3,302)I,AX,I,AY,I
      WRITE(3,303)I,I,AZ
      WRITE(3,300)I,RX,I,RY,I
      WRITE(3,301)I,I,RZ
      IF(ISIDE.EQ.5)GOTO 24
      GO TO 50
    24 CONTINUE
C**** SIDES 2 AND 4
      AX=-SIN(THE)
      AY=COS(THE)
      AZ=0.
      RX=-COS(PHI)*COS(THE)
      RY=-COS(PHI)*SIN(THE)
      RZ=SIN(PHI)
      WRITE(3,304)I,AY,I,AX,I
      WRITE(3,305)I,I,AZ
      WRITE(3,306)I,RY,I,RX,I
      WRITE(3,307)I,I,RZ
    50 CONTINUE
101 FORMAT(10,2F0.0,10)
300 FORMAT('MPC',1,' ',13,'14',1,' ',F8.6,18,'15',1,' ',F8.6,
18X,'&MP',12)
301 FORMAT('&MP',12,11X,18,'16',1,' ',F8.6)
302 FORMAT('MPC',1,' ',13,'11',1,' ',F8.6,18,'12',1,' ',F8.6,
18X,'&MA',12)
303 FORMAT('&MA',12,11X,18,'13',1,' ',F8.6)
304 FORMAT('MPC',1,' ',13,'12',1,' ',F8.6,18,'11',1,' ',F8.6,
18X,'&MB',12)
305 FORMAT('&MB',12,11X,18,'13',1,' ',F8.6)
306 FORMAT('MPC',1,' ',13,'15',1,' ',F8.6,18,'14',1,' ',F8.6,
18X,'&MC',12)
307 FORMAT('&MC',12,11X,18,'16',1,' ',F8.6)
      STOP
      END

```

Fig 1 cont'd

```

SUBROUTINE KTHEPHI(R,H,E,PNU,NO,IP,PIF)
C**** ICASE = 1
C**** THIS SUBROUTINE GENERATES NASTRAN BOUNDARY DISPLACEMENTS FOR THE
C**** INEXTENSIONAL BENDING CASE KAPPA CONSTANT
C**** THETA PHI
REAL U(3)
B=R*R*(1.+PNU)/(E*H**3)
DO 10 I=1,NO
READ(1,110)N,THE,PHI
THE=THE*PIF
PHI=PHI*PIF
CPHI=COS(PHI)
SPHI=SIN(PHI)
CTHE=COS(THE)
STHE=SIN(THE)
U(1)=4.0*B*CPHI*CTHE*CTHE*CTHE/SPHI
U(2)=-4.0*B*CPHI*STHE*STHE*STHE/SPHI
U(3)=-B*COS(2.0*THE)*(SPHI*SPHI-2.0*CPHI*CPHI)/(SPHI*SPHI)
DO 5 J=1,3
K=100*J+N
WRITE(3,300)N,J,U(J),K
5 WRITE(3,301)K
IP=IP+1
WPHI=-4.0*B*COS(2.0*THE)/(R*SPHI*SPHI*SPHI)
WX=-STHE
WY=CTHE
WRITE(4,405)N,WX,IP
WRITE(4,415)IP,N,WY,IP
WRITE(4,425)IP,IP,WPHI,IP
WRITE(4,435)IP
WRITE(5,500)IP
IP=IP+1
WTHE=-4.0*B*CPHI*SIN(2.0*THE)/(R*SPHI*SPHI*SPHI)
WX=-CTHE*CPHI
WY=-CPHI*STHE
WZ=SPHI
WRITE(4,400)N,WY,IP
WRITE(4,410)IP,N,WX,IP
WRITE(4,420)IP,N,WZ,IP
WRITE(4,430)IP,IP,WTHE
WRITE(5,500)IP
10 CONTINUE
STOP
110 FORMAT(10,2F0.0)
300 FORMAT('SPC',15X,'1',2,16,E16.9,'&SPC',13)
301 FORMAT('*SPC',13)
400 FORMAT('MPC',10X,'1',116,15X,'5',E16.9,'&MPC1',12)
410 FORMAT('*MPC1',12,1X,116,15X,'4',E16.9,16X,'&MPC2',12)
405 FORMAT('MPC',10X,'1',116,15X,'4',E16.9,'&MPC1',12)
415 FORMAT('*MPC1',12,1X,116,15X,'5',E16.9,16X,'&MPC2',12)
420 FORMAT('*MPC2',12,17X,116,15X,'6',E16.9,'&MPC3',12)
425 FORMAT('*MPC2',12,17X,116,15X,E16.9,'&MPC3',12)
430 FORMAT('*MPC3',12,1X,116,16X,E16.9)
435 FORMAT('*MPC3',12)
500 FORMAT('SPC',12X,'1',18,13X,'1.0')
END

```

```

SUBROUTINE KTHETHE(R,NOD,IP,PIF)
C**** ICASE = 2
C**** THIS SUBROUTINE GENERATES NASTRAN BOUNDARY DISPLACEMENTS FOR THE
C**** INEXTENSIONAL BENDING CASE KAPPA      , -KAPPA      CONSTANT
C****      THETA THETA      PHI PHI
REAL U(3)
DO 10 I=1,NOD
READ(1,110)N,THE,PHI
THE=THE*PIF
PHI=PHI*PIF
CPHI=COS(PHI)
SPHI=SIN(PHI)
CTHE=COS(THE)
STHE=SIN(THE)
U(1)=-CTHE**3*(CPHI*CPHI+1.0)/(3.0*SPHI)
U(2)=STHE**3*(CPHI*CPHI+1.0)/(3.0*SPHI)
U(3)=COS(2.0*THE)*CPHI*(1.0-1.0/(SPHI*SPHI))/3.0
DO 5 J=1,3
K=100*J+N
WRITE(3,300)N,J,U(J).K
5 WRITE(3,301)K
IP=IP+1
WPHI=+CPHI*COS(2.0*THE)*(1.0+2.0/(SPHI*SPHI))/(3.0*R*SPHI)
WX=-STHE
WY=CTHE
WRITE(4,405)N,WX,IP
WRITE(4,415)IP,N,WY,IP
WRITE(4,425)IP,IP,WPHI,IP
WRITE(4,435)IP
WRITE(5,500)IP
IP=IP+1
WTHE=-SIN(2.0*THE)*(2.0-SPHI*SPHI-4.0/(SPHI*SPHI))/(6.0*R*SPHI)
WX=-CTHE*CPHI
WY=-CPHI*STHE
WZ=SPHI
WRITE(4,400)N,WY,IP
WRITE(4,410)IP,N,WX,IP
WRITE(4,420)IP,N,WZ,IP
WRITE(4,430)IP,IP,WTHE
WRITE(5,500)IP
10 CONTINUE
STOP
110 FORMAT(10,2F0.0)
300 FORMAT('SPC',15X,'1',2,16,E16.9,'&SPC',13)
301 FORMAT('SDC',13)
400 FORMAT('MPC',10X,'1',116,15X,'5',E16.9,'&MPC1',12)
410 FORMAT('MPC1',12,1X,116,15X,'4',E16.9,16X,'&MPC2',12)
405 FORMAT('MPC',10X,'1',116,15X,'4',E16.9,'&MPC1',12)
415 FORMAT('MPC1',12,1X,116,15X,'5',E16.9,16X,'&MPC2',12)
420 FORMAT('MPC2',12,17X,116,15X,'6',E16.9,'&MPC3',12)
425 FORMAT('MPC3',12,17X,116,15X,E16.9,'&MPC3',12)
430 FORMAT('MPC3',12,1X,116,16X,E16.9)
435 FORMAT('MPC3',12)
500 FORMAT('SPC',12X,'1',18,13X,'1.0')
END

```

Fig 1 cont'd

```
      SUBROUTINE ETHEPHI(R,NOD,IP,PIF)
C**** ICASE = 3
C**** THIS SUBROUTINE GENERATES NASTRAN BOUNDARY DISPLACEMENTS FOR THE
C**** MEMBRANE SOLUTION EPSILON          CONSTANT
C****                                THETA PHI
      REAL U(3)
      DO 10 I=1,NOD
      READ(1,110)N,THE,PHI
      THE=THE+PIF
      PHI=PHI+PIF
      CPHI=COS(PHI)
      SPHI=SIN(PHI)
      CTHE=COS(THE)
      STHE=SIN(THE)
      SPHI2=SPHI*SPHI
      U(1)=-((4.0-2.0*SPHI2+SPHI2+SPHI2)*CTHE*COS(2.0*THE)+(2.0+SPHI2)
1          *(2.0*STHE*SIN(2.0*THE)-SPHI*CTHE*COS(2.0*THE)))*CPHI
2          /((6.0*SPHI2+SPHI)
      U(2)=-((4.0-2.0*SPHI2+SPHI2+SPHI2)*STHE*COS(2.0*THE)-(2.0+SPHI2)
1          *(2.0*CTHE*SIN(2.0*THE)-SPHI*STHE*COS(2.0*THE)))*CPHI
2          /((6.0*SPHI2+SPHI)
      U(3)=(1.0+CPHI*SPHI)*COS(2.0*THE)/(2.0+SPHI2)
      DO 5 J=1,3
      K=100*J+N
      WRITE(3,300)N,J,U(J),K
      WRITE(3,301)K
      IP=IP+1
      WPHI=-((4.0-2.0*SPHI2+SPHI2+SPHI2)*SPHI*COS(2.0*THE)
1          +3.0*(CPHI*CPHI+1.0)*COS(2.0*THE))/(6.0*2*SPHI2*SPHI2)
      WX=-STHE
      WY=CTHE
      WRITE(4,405)N,WX,IP
      WRITE(4,415)IP,N,WX,IP
      WRITE(4,425)IP,IP,WPHI,IP
      WRITE(4,435)IP
      WRITE(5,500)IP
      IP=IP+1
      WTHE=-CPHI*((2.0+SPHI2)*(1.-SPHI)*SIN(2.0*THE)/(3.0*R*SPHI2*SPHI2)
      WX=-CTHE*CPHI
      WY=-CPHI*STHE
      WZ=SPHI
      WRITE(4,400)N,WY,IP
      WRITE(4,410)IP,N,WX,IP
      WRITE(4,420)IP,N,WZ,IP
      WRITE(4,430)IP,IP,WTHE
      WRITE(5,500)IP
10  CONTINUE
      STOP
110  FORMAT(I0,2F0.0)
300  FORMAT('SPC',15X,'1',2:10,'E16.9','2SPC',I3)
301  FORMAT('SPC',I3)
400  FORMAT('MPC',10X,'1',I16,10X,'5',E16.9,'&MPC1',I2)
410  FORMAT('MPC1',2:10,I16,15X,'4',E16.9,16X,'&MPC2',I2)
405  FORMAT('MPC',10X,'1',I16,15X,'6',E16.9,'&MPC1',I2)
415  FORMAT('MPC1',2:10,I16,15X,'5',E16.9,16X,'&MPC2',I2)
420  FORMAT('MPC2',2:10,I16,15X,'6',E16.9,'&MPC3',I2)
425  FORMAT('MPC2',2:10,I16,15X,E16.9,'&MPC3',I2)
430  FORMAT('MPC3',2:10,I16,16X,E16.9)
435  FORMAT('MPC3',I2)
500  FORMAT('SPC',12X,'1',I3,13X,'1.0')
      END
```

```

SUBROUTINE ETHETHE(R,NOD,IP,PIF)
C**** ICASE = 4
C**** THIS SUBROUTINE GENERATE NAITRAN BOUNDARY DISPLACEMENTS FOR THE
C**** MEMBRANE SOLUTION EPSILON      , -EPSILON      CONSTANT
C****                                THETA THETA      PHI PHI
      REAL U(3)
      DO 10 I=1,NOD
      READ(1,110)N,THE,PHI
      THE=THE*PIF
      PHI=PHI*PIF
      CPHI=COS(PHI)
      SPHI=SIN(PHI)
      CTHE=COS(THE)
      STHE=SIN(THE)
      U(1)=(3.0*SPHI*SPHI+SPHI+CTHE+COS(2.0*THE)-2.0*COS(3.0*THE))/
1(3.0*SPHI**3)
      U(2)=(3.0*SPHI*SPHI+STHE+COS(2.0*THE)-2.0*SIN(3.0*THE))
1 / (3.0*SPHI**3)
      U(3)=CPHI+COS(2.0*THE)/(SPHI*SPHI)
      DO 5 J=1,3
      K=100*J+N
      WRITE(3,300)N,J,U(J),K
5 WRITE(3,301)K
      IP=IP+1
      WPHI=+4.0*CPHI+COS(2.0*THE)/(3.0*R*SPHI**3)
      WX=-STHE
      WY=CTHE
      WRITE(4,405)N,WX,IP
      WRITE(4,415)IP,N,WY,IP
      WRITE(4,425)IP,IP,WPHI,IP
      WRITE(4,435)IP
      WRITE(5,500)IP
      IP=IP+1
      WTHE=0.0
      WX=-CTHE+CPHI
      WY=-CPHI+STHE
      WZ=SPHI
      WRITE(4,400)N,WY,IP
      WRITE(4,410)IP,N,WX,IP
      WRITE(4,420)IP,N,WZ,IP
      WRITE(4,430)IP,IP,WTHE
      WRITE(5,500)IP
10 CONTINUE
      STOP
100 FORMAT(I0)
110 FORMAT(I0,2F0.0)
300 FORMAT('SPC' ,15X,'1',2.16,E16.9,'8SPC',I3)
301 FORMAT('*SPC',I3)
400 FORMAT('MPC+',10X,'1',I16,15X,'5',E16.9,'8MPC1',I2)
410 FORMAT('*MPC1',I2,1X,I16,15X,'4',E16.9,16X,'8MPC2',I2)
405 FORMAT('MPC+',10X,'1',I16,15X,'4',E16.9,'8MPC1',I2)
415 FORMAT('*MPC1',I2,1X,I16,15X,'5',E16.9,16X,'8MPC2',I2)
420 FORMAT('MPC2',I2,17X,I16,15X,'6',E16.9,'8MPC3',I2)
425 FORMAT('*MPC2',I2,17X,I16,16X,E16.9,'8MPC3',I2)
430 FORMAT('MPC3',I2,1X,I16,16X,E16.9)
435 FORMAT('*MPC3',I2)
500 FORMAT('SPC',12X,'1',I8,13X,'1.0')
      END
      FINISH

```

Fig 1 cont'd

```
LIST
PROGRAM(DATA)
INPUT1=CRO
OUTPUT3=CP0
END
MASTER GENCQUAD4
10 READ(1,100,END=99)N,N1,N2,N3,N4
WRITE(3,300)N,N1,N2,N3,N4
GOTO 10
99 STOP
100 FORMAT(5I0)
300 FORMAT('CQUAD4 ',13.7X,'1'.4(8,' 0.')
```

```
LIST
PROGRAM(DATA)
INPUT1=CRO
OUTPUT3=CP0
END
MASTER GENCTRIA3
10 READ(1,100,END=99)N,N1,N2,N3
WRITE(3,300)N,N1,N2,N3
GOTO 10
99 STOP
100 FORMAT(4I0)
300 FORMAT('CTRIA3 ',13.7X,'1'.3(8,' 0.')
```

```
LIST
PROGRAM(DATA)
INPUT1=CR0
OUTPUT2=CP0
OUTPUT3=LP0
END
MASTER GENGRID
INTEGER IBC(250)
REAL X(250),Y(250),Z(250)
READ(1,102)NNODE
DO 10 I=1,NNODE
  READ(1,100)N,X(N),Y(N),Z(N)
  IBC(I)=0
10 CONTINUE
20 READ(1,101,END=25)N,IBC(N)
  GOTO 20
25 DO 30 I=1,223
  IF(IBC(I).EQ.0)WRITE(2,200)I,X(I),Y(I),Z(I)
30 IF(IBC(I).NE.0)WRITE(2,201)I,X(I),Y(I),Z(I),IBC(I)
  STOP
100 FORMAT(I0,3F0.0)
101 FORMAT(2I0)
102 FORMAT(I0)
201 FORMAT('GRID      ',13,8X,3F8.6,8X,I8)
200 FORMAT('GRID',4X,I8,8X,3F8.6)
END
FINISH
```

Fig 1 concl'd

```

LIST
PROGRAM(DGFN)
INPUT1=CR0
OUTPUT3=CP0
END
MASTER CHANGE DATA
C**** PROGRAM TO ALTER THE COORDINATES OF A SEMILOOP DATA FILE SO THAT
C**** POSITION OF THE MIDSIDE NODES IS HALFWAY ALONG THE STRAIGHT LINE
C**** JOINING THE TWO VERTEX NODES
REAL NU,X(200),Y(200),Z(200)
INTEGER EL(8)
C**** READ IN START OF DATA FILE
READ(1,100)NFL
READ(1,101)E,NU,THICK,PRESSURE,DENSITY
READ(1,100)NAM
C**** READ IN X,Y,Z COORDINATES
DO 10 I=1,NAM
  10 READ(1,102)J,X(I),Y(I),Z(I)
C**** FOR EACH ELEMENT READ IN ELEMENT CONNECTIONS
DO 20 INEL=1,NEL
  READ(1,103)EI
  K=8
  IF(EL(8).EQ.0)K=6
C**** RECALCULATE POSITION OF MIDSIDE NODES AROUND THE ELEMENT
DO 20 J=2,K,2
  NV1=EL(J-1)
  IF(J.NE.K)NV2=EL(J+1)
  IF(J.EQ.K)NV2=EL(1)
  NM=EL(J)
  X(NM)=(X(NV1)+X(NV2))/2.0
  Y(NM)=(Y(NV1)+Y(NV2))/2.0
  20 Z(NM)=(Z(NV1)+Z(NV2))/2.0
C**** WRITE NODE NUMBER,X,Y,Z FOR EACH NODE
DO 40 I=1,NAM
  40 WRITE(3,300)I,X(I),Y(I),Z(I)
STOP
100 FORMAT(10)
101 FORMAT(E0.0,4F0.0)
102 FORMAT(10,3F0.0)
103 FORMAT(810)
300 FORMAT(110,3F20.10)
END
FINISH
****

```

Fig 2

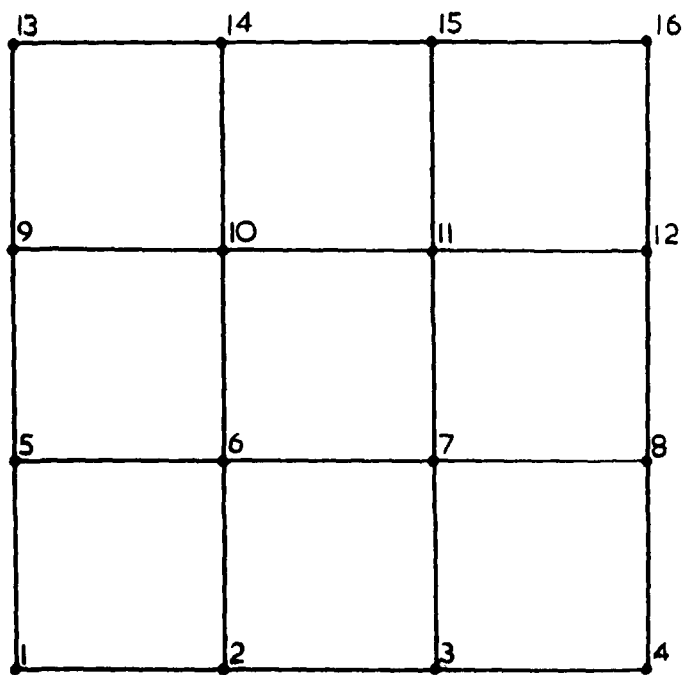
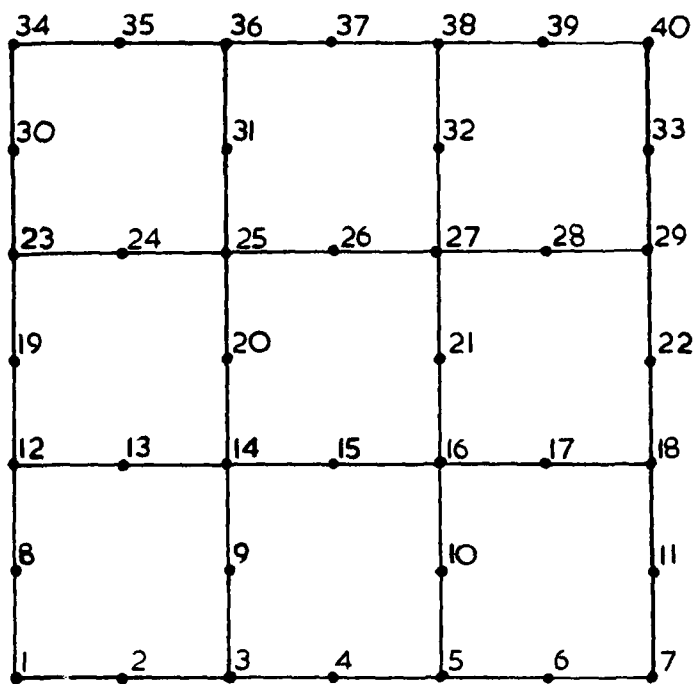


Fig 2 Mesh node numbering for patch tests

Fig 3a-c

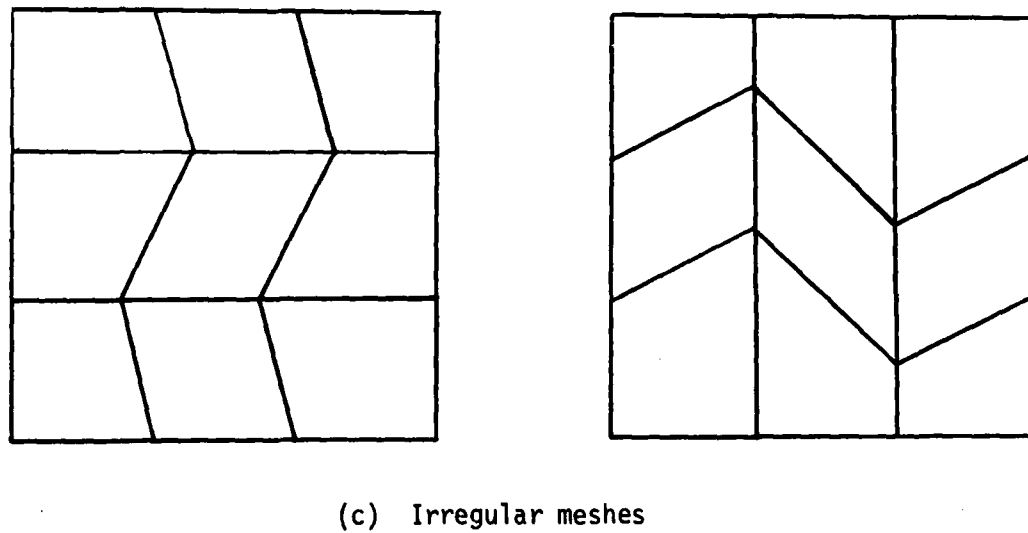
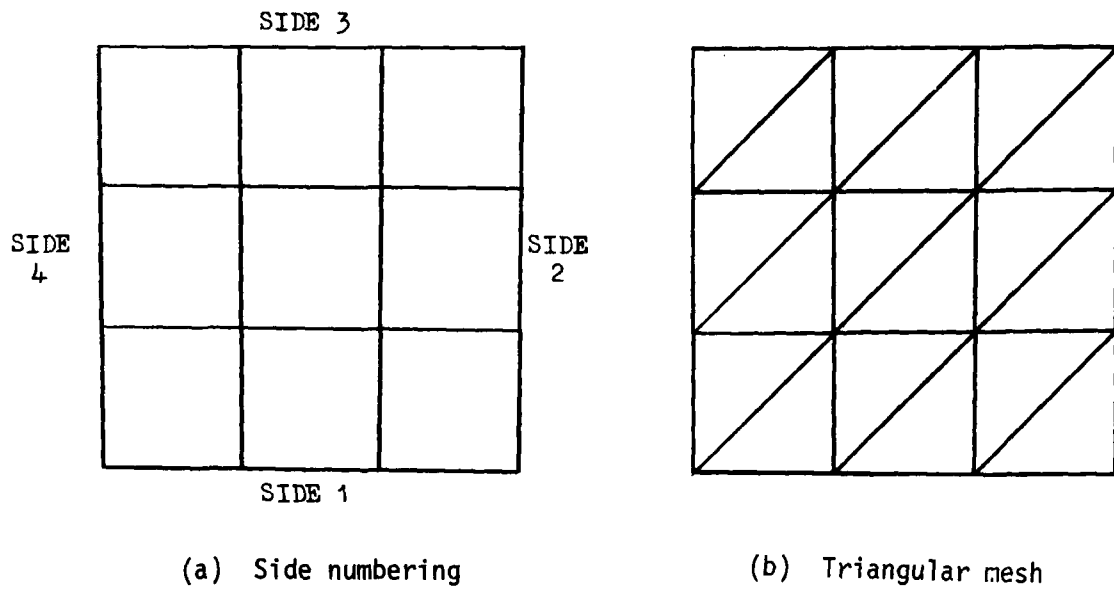
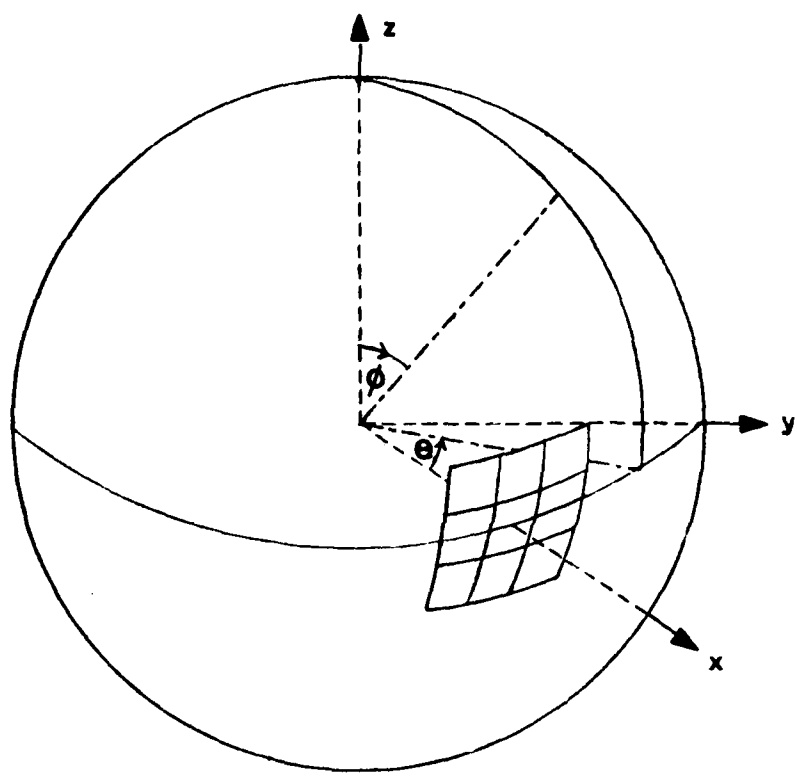
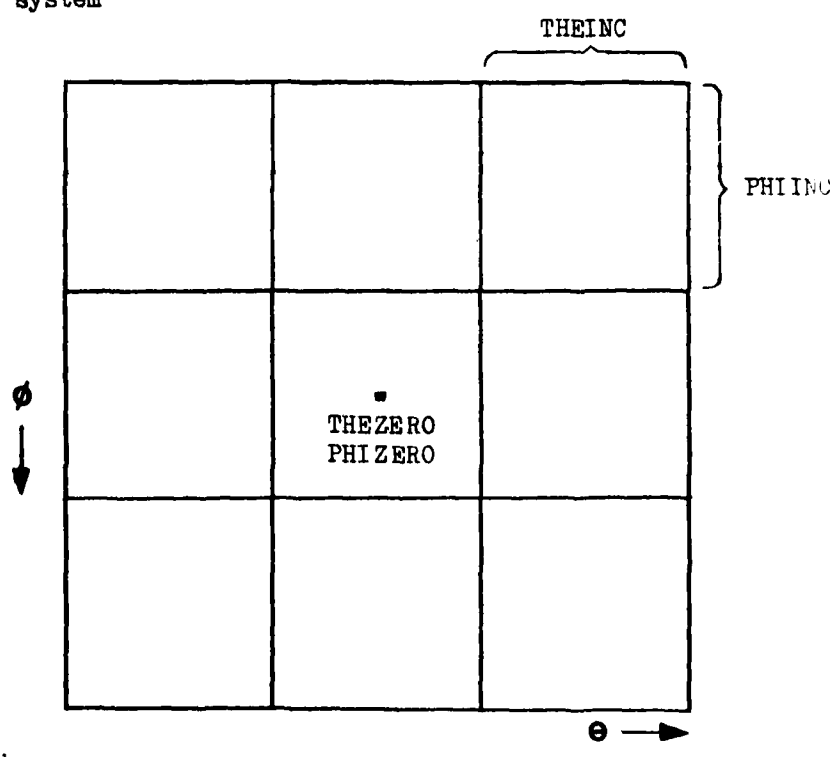


Fig 3a-c Typical meshes used for patch tests

Fig 4a&b



a) Co-ordinate system



b) Patch notation

Fig 4a&b Patch on the surface of a sphere

TM 582 4214

