

AD-A165 881

INSTRUCTION FOR USE OF THE DESADE CABLE ARRAYS STATIC
DEFLECTION PROGRAM(U) NAVAL FACILITIES ENGINEERING
COMHND WASHINGTON DC CHESAPEAKE DIV H SMH SEP 77
CHES/NAVFAC-FPO-7737

1/1

UNCLASSIFIED

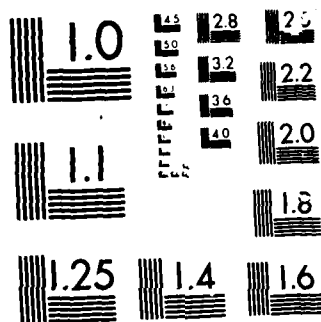
F/G 9/2

NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART

FPO
7737

TR-FPO-1E-23

AD-A165 801

INSTRUCTION FOR USE OF THE
DESADE
CABLE ARRAYS STATIC DEFLECTION PROGRAM

H. SHIH

SEPTEMBER 1977

DTIC
ELECTE
MAR 19 1986
S D

Approved by: S.C. LING, Manager
Engineering Analysis
Branch

S. C. Ling

Approved by: C.E. BODEY, DIRECTOR
Engineering and Design
Division

C. E. Bodey

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

OCEAN ENGINEERING AND CONSTRUCTION PROJECT OFFICE
CHESAPEAKE DIVISION
NAVAL FACILITIES ENGINEERING COMMAND
WASHINGTON, D.C. 20374

DTIC FILE COPY

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

AD-A165801

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION AVAILABILITY OF REP.
Approved for public release;
distribution is unlimited

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

4. PERFORMING ORGANIZATION REPORT NUMBER
FPO 7737, TR-FPO-1E-23

5. MONITORING ORGANIZATION REPORT #

6a. NAME OF PERFORM. ORG. 6b. OFFICE SYM
Ocean Engineering
& Construction
Project Office
CHESNAVFACENGCOM

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and Zip Code)
BLDG. 212, Washington Navy Yard
Washington, D.C. 20374-2121

7b. ADDRESS (City, State, and Zip)

8a. NAME OF FUNDING ORG. 8b. OFFICE SYM

9. PROCUREMENT INSTRUMENT INDENT #

8c. ADDRESS (City, State & Zip)

10. SOURCE OF FUNDING NUMBERS

PROGRAM	PROJECT	TASK	WORK UNIT
ELEMENT #	#	#	ACCESS #

11. TITLE (Including Security Classification)
Instruction for Use of the Desade Cable Arrays Static Deflection Program

12. PERSONAL AUTHOR(S)
H. Shih

13a. TYPE OF REPORT

13b. TIME COVERED
FROM TO

14. DATE OF REP. (YYMMDD) 15. PAGES
9/77 80

16. SUPPLEMENTARY NOTATION

17. COSATI CODES
FIELD GROUP SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if nec.)
Arrays, Cable

19. ABSTRACT (Continue on reverse if necessary & identify by block number)
This note describes the use of the DESADE program version presently running on the CDC 6600 through the FPO-1 computer terminal. A tri-moor cable array structure is given here as an example to illustrate the procedures in using this program. The original manual on which this note is based was (Con't)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT SAME AS RPT. 21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL
Jacqueline B. Riley

22b. TELEPHONE 22c. OFFICE SYMBOL
202-433-3881

DD FORM 1473, 84MAR

SECURITY CLASSIFICATION OF THIS PAGE

BLOCK 19 (Con't)

prepared by R.A. Skop. and J. mark of Naval Research Laboratory, Washington, D.C.

The program calculates the current - induced static deflections of structural cable arrays. Attached to this note are results of test runs made for the Linear Chari candidate configurations. This was designed to serve as a check for the on going Linear Chair analysis contract with DTNSRDC using the DSSM program.

TABLE OF CONTENTS

	<u>PAGE</u>
ABSTRACT	iii
1. Note of the Program	1
2. Array Description	2
3. Current Description	6
4. Program Structure	7
5. Program Flow Chart	8
6. Input Coding	9
7. Output Parameters	15
8. File Status	16
9. Computer Requirements and Cost	17
10. Run Steps	18
References	20
Appendix A - Results of test runs (with comparison to DSSM program runs for Linear Chair configurations)	
B - Sample of Input and Output	
C - Source Program Listing	

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

ABSTRACT

This note describes the use of the DESADE program version presently running on the CDC 6600 through the FPO-1 computer terminal. A tri-moor cable array structure is given here as an example to illustrate the procedures in using this program. The original manual on which this note is based was prepared by R. A. Skop and J. Mark of Naval Research Laboratory, Washington, D. C.

The program calculates the current - induced static deflections of structural cable arrays. Attached to this note are results of test runs made for the Linear Chair candidate configurations. This was designed to serve as an check for the on going Linear Chair analysis contract with DTNSRDC using the DSSM program.

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	23



1. Note of the program

- | | |
|--------------|--|
| Usefulness | 0 computing current - induced static deflections of structural cable arrays. |
| | 0 Most general (Ref. 1) |
| | 0 Experimentally validated (Ref. 3) |
| Method | 0 Use Method of imaginary reactions (Ref. 4) and method of successive approximations for treating position and configuration dependent forces (Ref. 5) |
| Assumptions | 0 The hydrodynamic force on the devices and cables is the normal drag. |
| | 0 The current is unidirectional and horizontal (with certain modification it could accept arbitrary current field with variable directions). |
| Capabilities | 0 arbitrarily configured arrays of up to 22 cables |
| | 0 variable cable materials |
| | 0 any number of discrete devices |
| Limitations | 0 Cable segments cannot be on the ocean floor. |
| | 0 Dimensions of discrete device must be small compared to overall array dimensions (not valid for moored submerged submarine.) |
| | 0 All parts of the array must be submerged (unless the surfaced device coordinates are specified), |

2. Array description

EXAMPLE : A sea spider (tri-moor) array structure is used here to illustrate the steps required for in preparing array structure input data. The array design is sketched in Figure 1. It is intended to calculate the static deflections and the forces of each structural member under hydrodynamic loading of an ocean current (Figure 2.)

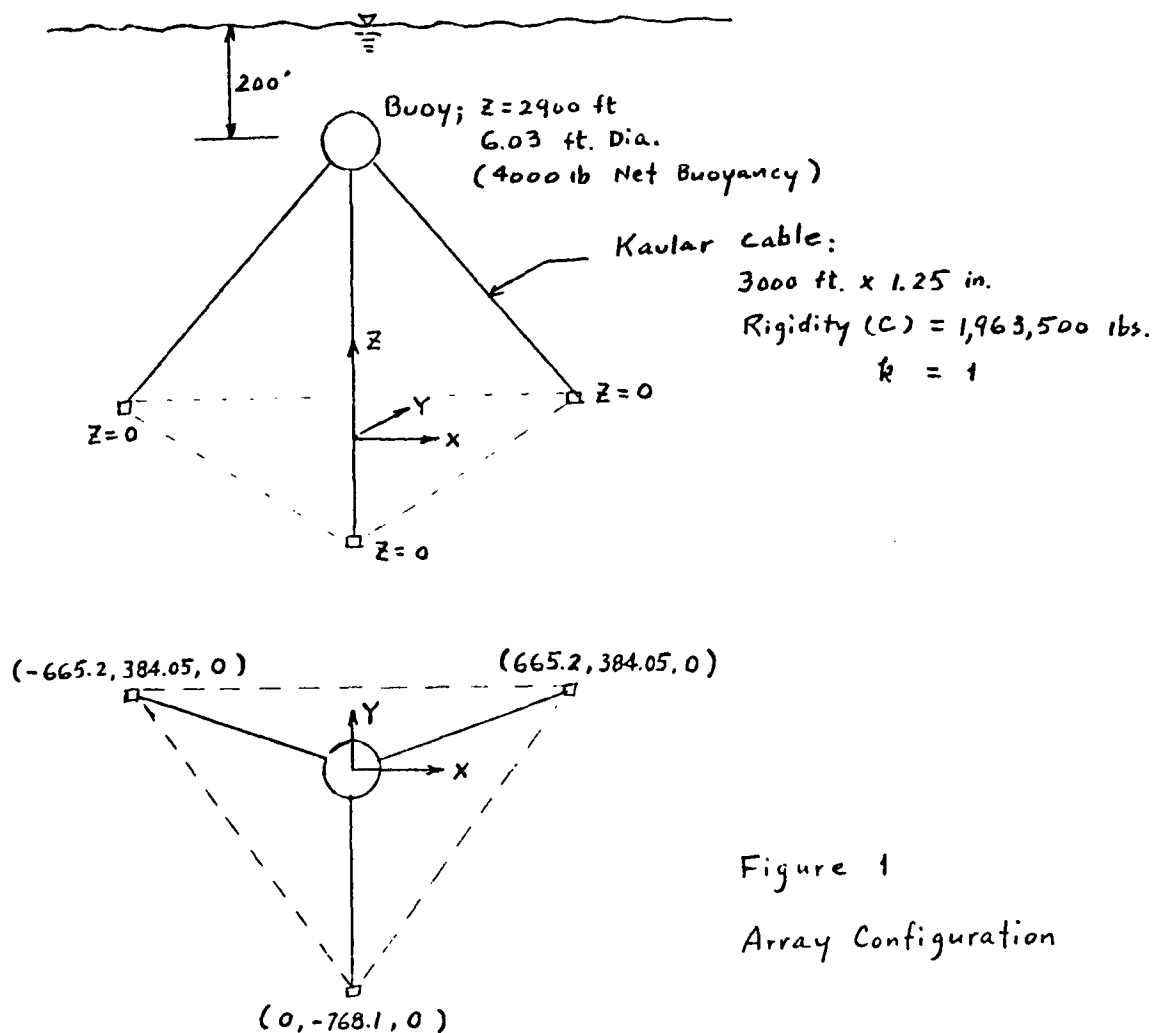


Figure 1
Array Configuration

STEPS

- Number the cables consecutively from one to the total number of cables in the array. Each cable so designed must have uniform properties (weight, diameter, drag coefficient, and constitutive relation). A change in property also requires a change in cable number.
- Number the junctions consecutively from one to the total number of junctions in the array. A junction may designate an anchor, the intersection point of two or more cables, the free end of a cable.
- Use a fixed, right-hand cartesian coordinate system to describe the configuration of the array in the space. The origin of the system can be arbitrarily located. Z axis is defined parallel to the direction of gravity and increasing upward. In this example the origin is placed at the geometric center of the equilateral triangle formed by the three anchors in the sea floor plan.
- Tabulate the anchor coordinates (Table 1.)

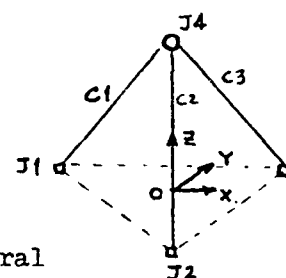
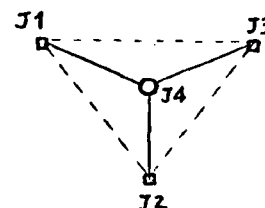
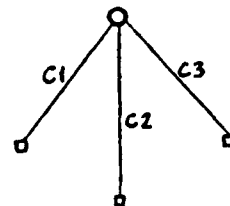


Table 1

Junction No. of anchor	Anchor coordinates		
	X (ft)	Y (ft)	Z (ft)
1	-665.2	384.05	0
2	0	-768.1	0
3	665.2	384.05	0

e. Reduce the array to statically determinate array structure. The cuts are made in the following manner:

1. Number of cuts = number of cables + number of anchors - number of junctions
2. Cuts must be made at end points of cables (adjacent to junctions).
3. The first group of cuts must be made so as to release all but one cable from an anchor.
4. The remaining required cuts are made within the array and must be located so as not to break the array into separated parts.
5. Assign each new cut a consecutive junction number continuing from the last-used junction number.
6. Tabulate the junction number

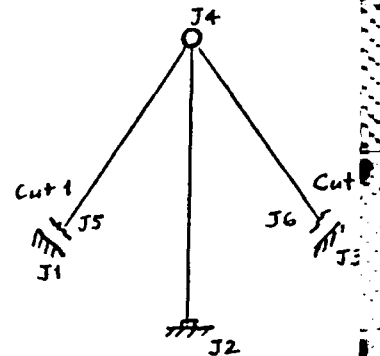
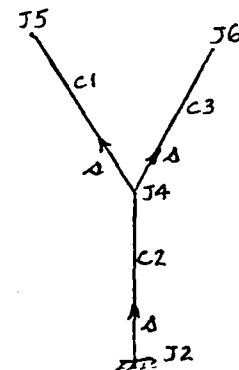


Table 2

Cut number	Original Junction Number at cut	New Junction Number at cut
1	1	5
2	3	6

7. Define the directions of increasing arc length, (s), along each cable by using a topological tree diagram for the reduced array structure (see sketch). The directions are indicated by the arrowheads in the climbing 'up' direction from root to top.



8. Following the directions given in step 7 tabulate the junction number of the starting ($s=0$) and ending ($s=L$) points for each cable (Table 3).

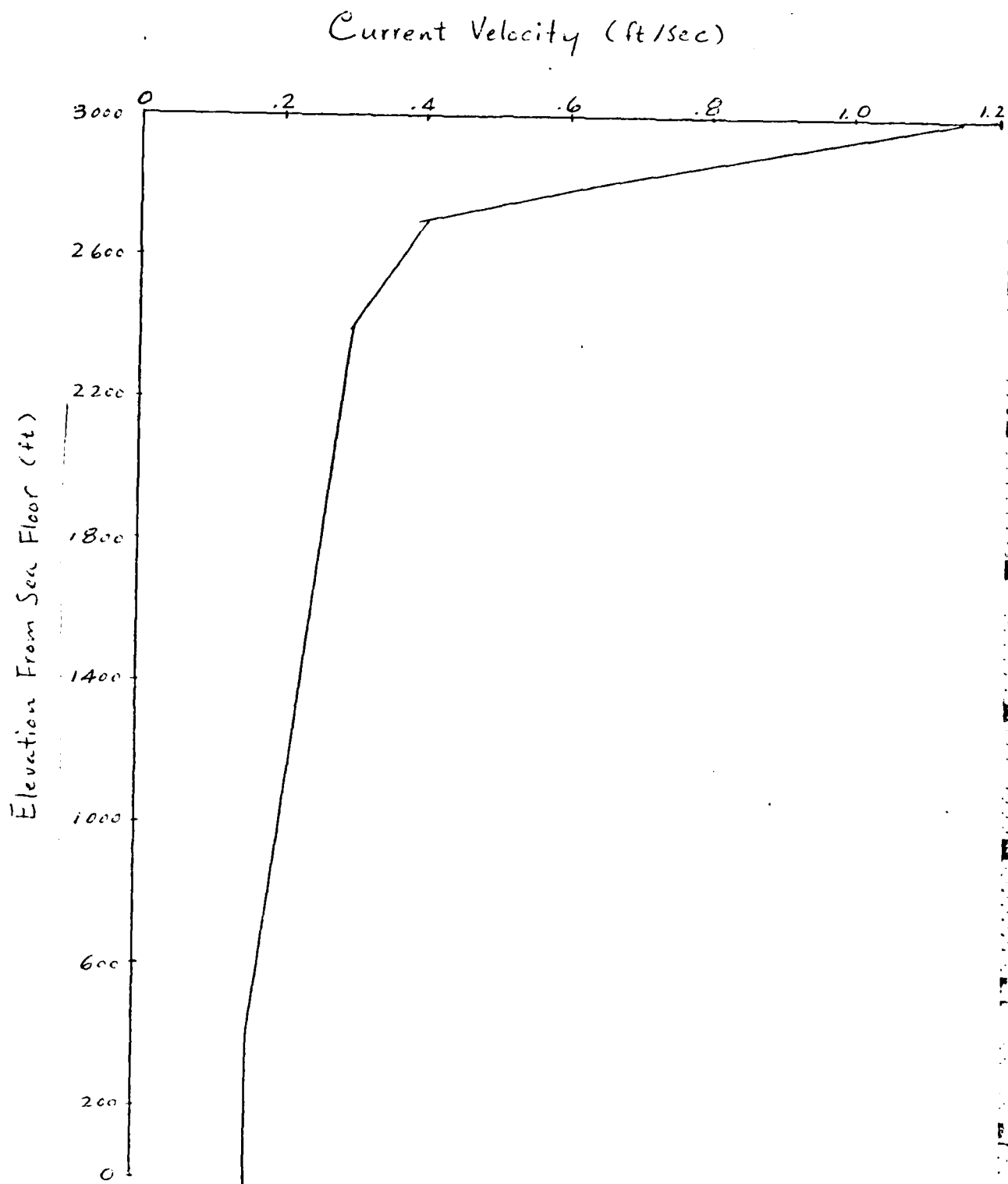


Figure 2 Current Profile in St Croix

Table 3

Cable number	Junction number at s=0	Junction number at s=L
1	4	5
2	2	4
3	4	6

The starting and ending points of each cable can be identified by following the direction of 'climbing up' the topological tree diagram. s is the arc length measured from starting junction. L is the length of each cable.

3. Current description

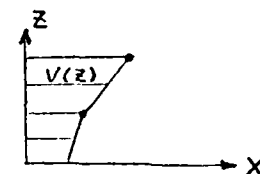
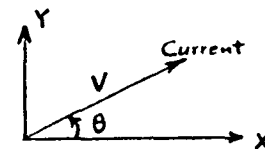
The standard current field is defined as

$$V = V(z) (\cos \theta \hat{i} + \sin \theta \hat{j})$$

where $v(z)$ = current value at z .

θ = current angle with respect to x axis

$\hat{i}, \hat{j}$ = unit vectors in the x and y axes



The program generates the velocity profile by connecting each current data point by a series of straight lines. The current below the minimum z value is considered the same as $V(z_{\text{minimum}})$. Using the current profile given in Figure 2 the input current data can be tabulated (Table 4.)

Table 4

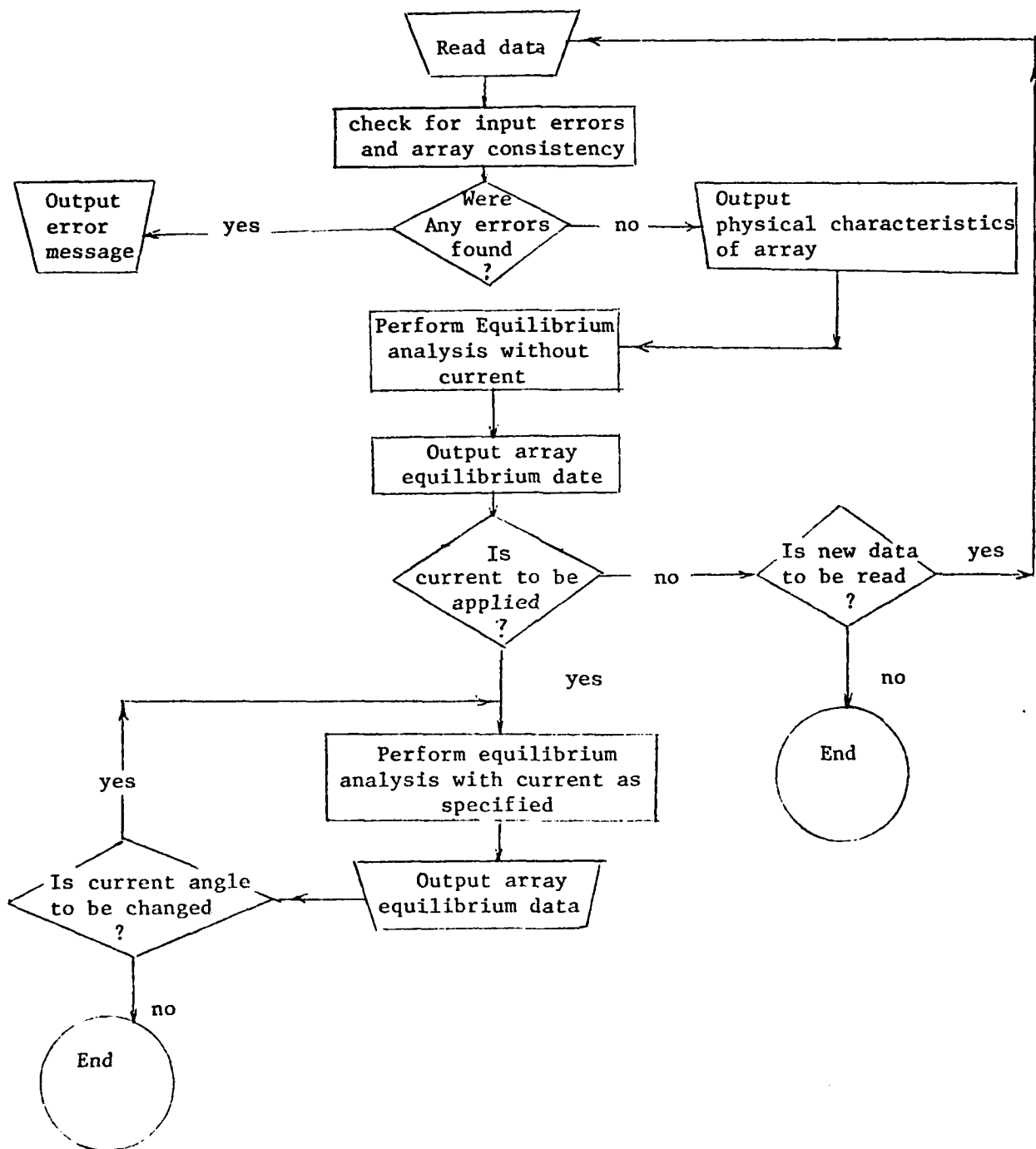
Z Coordinate (ft)	V (Z) (Knots)
0	.1
400	.1
2400	.18
2700	.24
2900	.53

4. Program Structure

In addition to the main program 'DESADE' there are 8 subroutines and 7 functions. These are described below:

Subroutines	Purpose
INPUT	Reads all data and identifies errors in data, deck structure, and array representation.
PHSOUT	Generates information concerning the physical characteristics of the structural cable array.
STROUT	Generates the array structural output.
TAPOUT	Generates the tape or cards giving the locations of the indexed devices.
ERROR	Generates error message when the accuracy required for the array equilibrium calculation could not be obtained.
RPOLY	Finds real roots of polynomial equations for use in evaluating maximum cable displacements and tension extreme.
SWITCH	Switches input data.
START	Calculates the initial guesses.
Functions	Purpose
TCAB(M,K)	Calculates the tension at node M of Cable K.
EXCAB(M,K)	Calculates (1 + strain) at node M of Cable K.
EFORCE (I)	Calculates the hydrodynamic drag force on each device in direction I.
CFORCE (I,M,N)	Calculates the force/length in direction I at node M on Cable N.
SPACE (I)	Calculates the location in space of any point on the array.
TANG (I)	Calculates the unit tangent to a cable at any point.
VELOC (I,P SPACE)	Specifies the I component of the current field at an arbitrary point in space, PSPACE (I).

5. Program Flow Chart



6. Input Coding

(1) I/O logical unit number card

Column	1	4	8	16	24	32	40	48	56	64
	Ncard	Card	Nread	Nwrite	Ntape	Iopt.	Blank	Mopt.	Kopt.	

Ncard = card number, (I4), From - 999 to 9999, used to sequence the data cards

Card = LUN, (A4), used to catalog each data card

Nread = 5, (I8), reader unit number

Nwrite = 6, (I8), printer unit number

Ntape = 2, (I8), service tape unit number

Iopt. = 0, (I8), Input option using source deck

Mopt. = 2 or 0, (I8), Output option to printer

Kopt. = 3 or blank, (I8), Output tape unit number

(2) number card of junctions

1	4	8	16
Ncard	Card	Njun.	

Card = Njun., (A4), card name

Njun. = number of junction in original Array, (I8)

(See Figure 1), $2 \leq N_{Jun} \leq 44$

(3) Anchor location cards (one for each anchor)

1	4	8	16	24	32	40
Ncard	Card	Njun	X	Y	Z	

Card = ANC, (A4)

Njun. = Junction number of anchor (original array)

$1 \leq N_{Jun} \leq 44$, (I8)

X = anchor X - coordinate (ft.), (F8.0)

Y = anchor Y - coordinate (ft.), (F8.0)

Z = anchor Z - coordinate (ft.), (F8.0)

Refer to Table 1.

(4) Cut data cards (one for each cut)

1	4	8	16	24
Ncard	Card	NJun,New	NJun,Old	

Card = $\sim \sim$ IR, (A4)

NJun,New = New junction number at cut, (I8)

NJun.Old = Original junction number at cut, (I8)

$$1 \leq \text{NJUN,Old} \leq 44$$

Refer to Table 2.

(5) Cable cards (one for each cable)

1	4	8	16	24	32	40	48	56	64	72	77	80
Ncard	Card	Nc	NJun.	NJun.	Wc	CDc	d	L	C	h	Nele.	

 $\begin{matrix} A=C \\ A=L \end{matrix}$

Card = $\sim$ CAB, (A4)

Nc = cable number; (I8), $1 \leq N_c \leq 22$

NJun,s=0 = Junction number at s=0, (I8), $1 \leq N \leq 44$

NJun,s=L = Junction number at s=L, (I8), $1 \leq N \leq 44$

Wc = cable unit weight in sea water (lb/ft), (F8.0)

(+) if positively buoyant

(-) if negatively buoyant

CDc = cable normal drag coefficient, (F8.0), (based on cable diameter)

d = cable diameter, (in.), (F8.0)

L = cable unstressed length (ft), (F8.0)

C = cable rigidity (lb), (F8.0)

$(T/C)^k = \epsilon$, T = tension, ϵ = Strain, k = constant

k = constant in cable constitutive relation, (F5.0)

Nele. = number of finite elements in cable calculation, (I3),

$0 < N \leq 50$

(6) Device on cable cards (one for each device)

1	4	8	16	24	32	40	48	56	64	72
N _{card}	Card	N _c	I _{type}	Index	W _d	C _{Dd}	A.	1	S _d	

Card = DCAB, (A4)

N_c = number of cable to which device is attached, (I8)

$1 \leq N \leq 22$

I_{type} = device type, (I8), 1 or 3 for in-line elongated device
(longitudinal axis aligns with cable axis), 2 or 4 for
other free devices or for devices inside the cable.

I_{Index} = Device index if type 1 or 2, (I8) $1 \leq I \leq 1000$, Type 1 and
2 must be indexed consecutively from one to the total number
of type 1 and 2 devices in the array.

W_d = device weight in sea water (lb.), (F8.0)

(+) if positively buoyant

(-) if negatively buoyant

C_{Dd} = device drag coefficient, (F8.0)

(based on frontal area or diameter)(blank if device is inside
the cable)

A. = Frontal area of device for type 2 or 4 (ft²), (F8.0)

or diameter of device for type 1 or 3, (in), (F8.0).

l = device length (if type 1 or 3), (ft), (F8.0).

S_d = Unstressed distance of device from $s=0$ junction of the cable, (ft), (F8.0).

(7) Device on junction cards (one for each device)

1	4	8	16	24	32	40	48	56
N_{card}	Card	N_{Jun}	I_{type}	I_{dex}	W_d	C_{od}	A.	

Card = DJNC, (A4)

$N_{Jun.}$ = number of junction to which device is attached, (I8)

$$1 \leq N \leq 44$$

I_{type} = Device type (2 or 4), (I8)

I_{dex} = Device index if type 2, (I8)

$1 \leq I \leq 1000$ (indexed consecutively from one to the total number of type 1 and 2 devices in the array.

W_d = Device weight in sea water (lb.) (F8.0)

(+) if positively buoyant

(-) if negatively buoyant

C_D = Device drag coefficient based on frontal area (F8.0)

A. = device frontal area (ft^2), (F8.0)

(8) Water density card

1	4	8	16
N_{card}	Card	ρ_f	

Card = ^DEN, (A4)

ρ_f = sea water density, (slug / ft^3), (F8.0)

- (9) End of data card

1	4	8
N _{card}	Card	

Card = ^EOD, (A4)

- (10) Flag card to specify current data

1	4	8	16
N _{card}	Card	I _{opt.current}	

Card = NDAT, (A4)

I_{opt. current} = Current option = 1 standard

= 2 nonstandard (use modified program)
(I8)

- (11) Accuracy card (following the 1st NDAT card)

1	4	8	16
N _{card}	Card	E	

Card = COMP, (A4)

E = accuracy specified (ft), (F8.0)

To insure the calculated coordinates of every point in the array are within $\pm E$ of their exact values. The obtainable $\bar{E}$ is limited by the computer capacity and by the largest linear dimension in the array.

$\bar{E} \leq 10^{m-n+3}$ where m = common logarithm of the largest linear dimension, n = number of significant figures carried in single precision (e.g. $n = 8$, $L = 25,000$ ft, $\bar{E} \geq 0.1$ ft.)

$\bar{E} = 1$ ft. was used in the test runs.

- (12) Current cards (one for each point)

1	4	8	16	24
N _{card}	Card	Z	V(Z)	

Card = ^VEL, (A4)

$Z = Z$ - coordinate of input point, (ft), (F8.0)

$V(Z)$ = Velocity at input point, (knots), (F8.0)

Up to 25 VEL cards are permitted. At least on input point must be below or equal the minimum Z coordinate of the anchors.

(13) Current direction card

1	4	8	16	24	32
N_{card}	Card	θ_o	$\Delta\theta$	θ_f	

Card = ^ ANG, (A4)

θ_o = initial current angle (deg.), (F8.0)

$\Delta\theta$ = increment in current angle (deg. > 0), (F8.)

θ_f = final current angle (deg. $\geq \theta_o$), (F8.0)

(14) End of data card

1	4	8
N_{card}	Card	

Card = ^ EOD, (A4)

(15) Termination card

1	4	8
N_{card}	Card	

Card = EOP, (A4)

7. Output Parameters

The outprints are self-explanatory and consist of:

a. Array descriptive output:

- o anchor locations
- o cuts information
- o cable properties
- o device properties
- o current field

b. Structural output:

- o current condition
- o cable forces and angles at each anchor
- o location of the array junctions
- o displacement of these junctions from the no-current coordinates
- o cable forces and angles at each junction
- o maximum and minimum tension and their location for each cable
- o location and displacement of devices from the no-current coordinates
- o tension at devices.

A sample output for the tri-moor is given in Appendix B-2.

8 . File Status

NAME	DESCRIPTION	STATUS	
		* ON TAPE	ON DISK
DESADEK	Source program	X	
DESADW	Source program modified to accept variable current directions.	X	X
DSDLGØK	Compiled program on NCS system	X	
GØDESAD	Procedure file to run by Scope system (Compile, catalog LGØ file, and run.)	X	X
GØDSAD	Procedure file to run by Scope system (run LGØ file.)	X	X
GØDSW	Procedure file to run by Scope system (Compile and run.)	X	X
DATA	Bench data from Ref. 1	X	X
DATASP	Data file for single post near	X	X
DATAGP1	Data file for goal post moor	X	X
DATAGP8	Data file for goal post w/eight additonal buoys on each post.	X	X
DATABGP	Data file for braced goal post post moor.	X	X
DATASS	Data file for sea spider moor.	X	X
DATA4	Data file for single post single buoy moor.	X	X
DATAW	Data file for DOCMS moor:	X	X

*TAPE NAME: KW2233 Created on 8-3-77

9. Computer Requirements and Cost

a. Requirements

DESADE is a Fortran IV program and is ready for CDC Fortran IV compilers. Memory requirements are approximately 30,000 words in a single precision. Access to one, two or three magnetic tape units, depending on the I/O options chosen, is required by the program.

b. Cost

Computer costs for the test runs are shown in the following table :

ARRAY CONFIGURATION	I/O time (SEC)	Execution time* (SEC)	total* SBU UNITS (SEC)	ESTIMATE** TOTAL COST (\$)
Single post	14	13	5/30's	9
Sea spider	14	54	110/30's	34
Goal Post	14	24	36/30's	16
Braced Goal Post	14	84	208/10's	58

* CDC Computer charge unit

** Based on over night rate (priority 2) plus terminal connecting charge and I/O charge. Current rates are (July, 1977) :

Priority	(Overnight)	(Daytime)		
	2	3	4	5
Rate (\$)	.24	.28	.34	.45

Terminal connection: \$9/hr.
Terminal I/O: \$20/1000 character

Note the compilation time (~ 4.5 sec) was saved for each run through the use of a compiled binary file (DESLGO)

10. Run Steps

(1) Log in XXXXXXXXXXXX KB, IHJ11*

[illegible]

(2) Prepare data file

- a. Either create your own file (see next section on how to establish data file).
- b. Or edit existing data files of similar array configuration.
- c. A sample of data file is shown on page 20 .

(3) Prepare run procedure file

- a. Edit existing file to desired condition (job priority, data file name, etc.)
- b. A sample of run procedure file is shown on page 19 .

(4) Submit job

- a. See example on page 19.

(5) Check for job status

- a. See example on page 19.

(6) Get day file when job is ready for output.

- a. See example on page 19.

(7) Print output

- a. See example on page 19.

* The underlined characters are user's entry. Consult with FPO-1E to update these entries.

— priority

— Data file

READY.
SUBMIT, GODSAD, ST=ECZ, T *
14.42.37 (AEDCBQK)

Assigned job name

READY: QSTATUS* — Check for job status

<u>STATUS</u>					
CARTEL NEW ACTIVE	RC				
CONFIRM ROUTE TO	TRAM				
CONFIRM ROUTE TO	TRAM				
77/09/01, 11:50:01	ACTIVE	CONFIRM	TRAM	TRAM	
77/09/01, 10:50:01	ACTIVE	CONFIRM	TRAM	TRAM	TRAM
YOUR LEAVE TODAY	NO				

[Handwritten notes:]

- Job AEOCBQK is ready for print
- RECEIVED - 08-17-2006 10:00 AM

[illegible]

Print Output

LNH, F-FILE

* The underlined characters are user's entries.

References

1. R. A. Skop, and J. Mark, 'A Fortran IV program for computing the static deflections of structural cable arrays', NRL Report 7640, August 1973.
2. N. D. Albertsen, 'A survey of techniques for the analysis and design of submerged mooring systems', CEL Report 815, Aug. 1974.
3. T. R. Kretschmer, G.A. Edgerton, and N. D. Albertsen, 'Seafloor construction experiment, SEACON II - an instrumented Tri-moor for evaluating undersea cable structure technology', CEL Report R848, Dec. 1976.
4. R. A. Skop, and G. J. O'Hara, 'The static equilibrium configuration of cable arrays by use of the method of imaginary reactions', NRL report 6819, Feb. 1969.
5. R. A. Skop, and R. E. Kaplan, 'The static configuration of a tri-moored, subsurface, buoy - cable array acted on by current - induced forces', NRL Report 6894, May 1969.
6. R. L. Webster, 'DSSM computer program', GE Report 1976.

APPENDIX A

Results of Test Runs

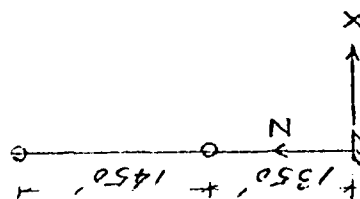
The test runs were made for the Linear Chair candidate configurations. The results have been used to compare the calculations made by DTNSRDC using the DSSM computer program (Ref. 6). In the following tables the DTNSRDC's results are enclosed by parentheses. Good agreement exists in most of the cases. The computer costs are comparable. However the finite element representation for each cable member is much finer in the test run calculations than those made by DTNSRDC.

Single Point Moor — Deviation in feet from zero position

Cable — 1 1/4" Dia.

Current — Uniform = .2 Knots

Buoyancy (lbs)	Top Buoy		Bottom Buoy	
	X - Axis Y	Z - Axis	X - Axis Y	Z - Axis
3000	(13.74) 14.6	(5.74) 5.9	(7.94) 8.4	(3.73) 3.8
4000	(10.18) 10.9	(7.87) 8.1	(5.87) 6.3	(5.09) 5.2

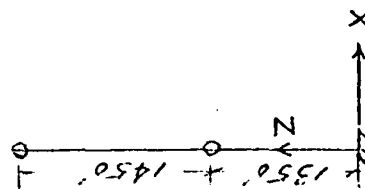


Single Point Moor — Deviation in feet from zero position

Cable — 1 1/4" Dia.

Current — St. Croix

Buoyancy (lbs)	Top Buoy		Bottom Buoy	
	X - Axis Y	Z - Axis	X - Axis Y	Z - Axis
3000	(13.26) 15.5	(5.74) 5.9	(6.03) 6.9	(3.74) 3.8
4000	(10.01) 12.1	(7.88) 8.1	(4.53) 5.3	(5.10) 5.2

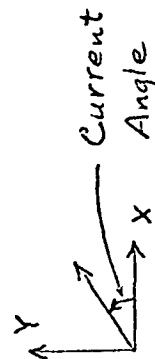
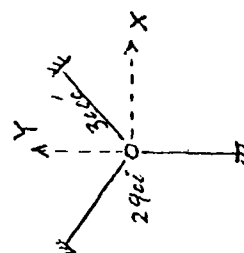


Sea Spider Moor — Deviation in feet from zero position

Cable — $1\frac{1}{4}$ " Dia. Buoyancy — 3000 lbs

Current — Uniform = .2 Knots

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	60°	90°	0°	30°	60°	90°	0°	30°	60°	90°
Top	(1.91)	(.46)	(.28)	(0)	(-.78)	(.55)	(2.04)	(2.71)	(-5.87)	(-5.84)	(-5.87)	(-5.9)
	2.4	1.1	.2	0	-.7	.7	2.3	3.1	-6.0	-5.9	-6.0	-6.0



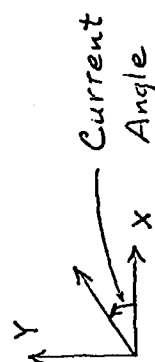
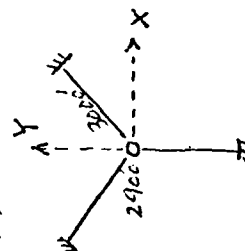
Sea Spider Moor — Deviation in feet from zero position

Cable — $1\frac{1}{4}$ " Dia. Buoyancy — 3000 lbs.

Current — St. Croix

Buoy Location	X - Axis			Y - Axis			Z - Axis		
	Current Angle			Current Angle			Current Angle		
	0°	30°	60°	90°	0°	30°	60°	90°	
Top ($G_0=12$)	8.7	8.2	5.0		1.0	4.9	7.0	-6.1	-6.2 -6.1

Top ($G_0=47$) 10.0 9.7 5.9 0 1.3 5.7 8.0 8.6 -6.3 -6.2 -6.1
(10.47) (0) (1.48) (9.38) (-6.2) (-6.1)



Sea Spider Moor — Deviation in feet from zero position

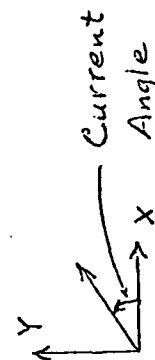
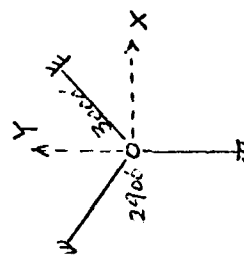
Cable — $1\frac{1}{4}$ " Dia. Buoyancy — 4000 lbs.

Current — Uniform = .2 knots

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	60°	90°	0°	30°	60°	90°	0°	30°	60°	90°
Top ($C_0 = .47$)	(-.84) -.5	(-1.19) -1.4	(-.88) -1.1	(0) 0	(-.53) -.5	(-.69) -.6	(-.46) -.4	(-.31) -.1	(-1.24) -1.3	(-1.23) -1.2	(-1.24) -1.2	(-1.35) -1.3

Top
($C_0 = .12$)

- .6 -1.5 -1.2 - .5 - .7 - .4 -1.3 -1.2 -1.2

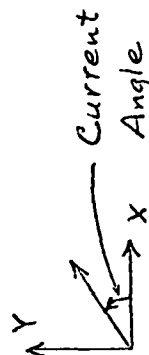
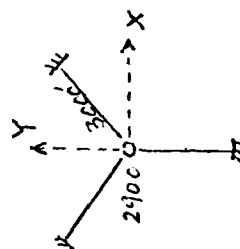


Sea Spider Moor — Deviation in feet from zero position

Cable — $1\frac{1}{4}$ " Dia. Buoyancy — 4000 lbs

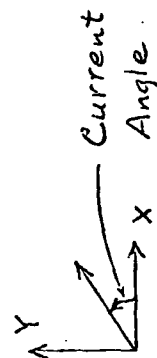
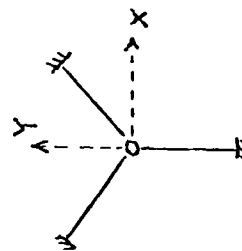
Current — St. Croix

Buoy Location	X - Axis					Y - Axis					Z - Axis				
	Current Angle					Current Angle					Current Angle				
	0°	30°	60°	90°		0°	30°	60°	90°		0°	30°	60°	90°	
Top	(3.0)			(0)		(.19)			(2.87)		(-1.26)			(-1.25)	
				0					2.7					-1.3	



Sea Spider Moor — Deviation in feet from zero position
 Cable — $1\frac{1}{4}$ " Buoyancy — 3000 lbs
 Current — Uniform = .3 kt.

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	60°	90°	0°	30°	60°	90°	0°	30°	60°	90°
Top	(12.0) 13.9	(7.0) 10.2	6.2	(0)	(-1.4) -7	(4.0) 5.9	12.4		-8.8	-8.4	-8.8	



Goal Post Moor — Deviation in feet from zero position

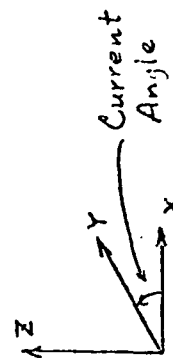
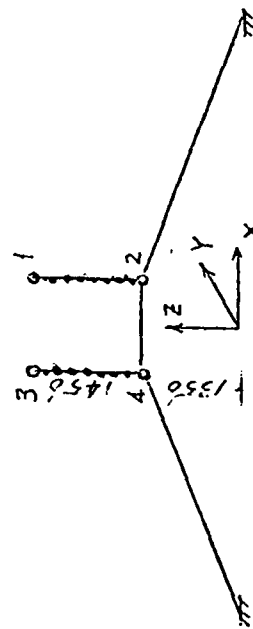
Cable — $1\frac{3}{4}$ " Dia.

Buoyancy — 2000 lbs plus

Current — Uniform = .2 knots

8 @ 250 lbs

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	30°	45°	90°	0°	30°	45°	90°
Top Right 1	(8.72) 8.9		6.4	(.46) .5	0		18.1	(28.27) 28.4	(16.37) 16.5		16.5	(15.75) 16.0
Bottom Right 2	(.66) .7		.6	(.46) .5	0		12.4	(20.21) 20.2	(15.88) 16.0		16.0	(15.26) 15.5
Top Left 3	(7.8) 7.9		5.5	(.46) .5	0		18.1	(28.27) 28.4	(15.43) 15.6		15.7	(15.75) 16.0
Bottom Left 4	(-.26) -.2		-.3	(-.46) -.5	0		12.4	(20.21) 20.2	(14.94) 15.1		15.2	(15.26) 15.5

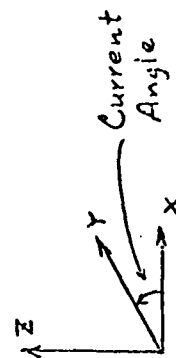
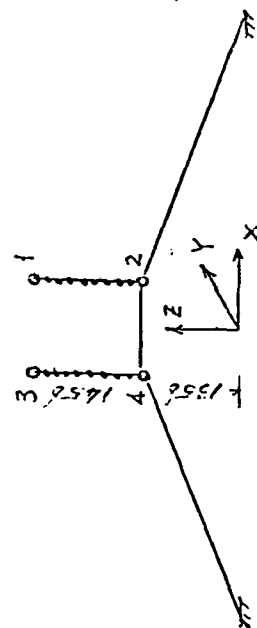


Goal Post Moor — Deviation in feet from zero position

Cable — $1 \frac{3}{4}$ " Dia. Buoyancy — 2000 lbs plus

Current — St. Croix 8 @ 250 lbs

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	30°	45°	90°	0°	30°	45°	90°
Top Right 1	(10.7) 11.9		8.5	(.46) .4	0		16.4	(22.15) 24.2	(16.24) 16.4		16.4	(15.88) 16.1
Bottom Right 2	(.62) .7		.6	(.46) .4	0		8.5	(12.09) 13.0	(15.57) 15.9		15.9	(15.37) 15.6
Top Left 3	(9.75) .9		7.6	(-.46) -.5	0		16.4	(22.15) 24.5	(15.57) 15.7		15.8	(15.85) 16.1
Bottom Left 4	(-.3) -.3		-.4	(-.46) -.5	0		8.5	(12.09) 13.0	(15.09) 15.2		15.4	(15.37) 15.6



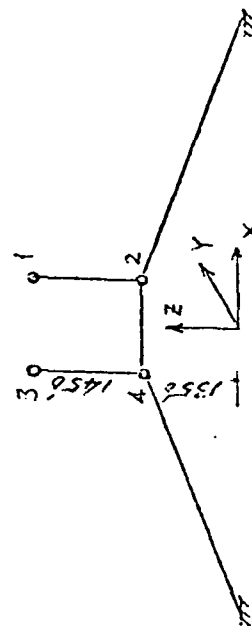
Goal Post Moor — Deviation in feet from zero position

Cable — $1\frac{3}{4}$ " Dia.

Buoyancy — 3000 lbs.

Current — Uniform = .2 Knots

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	30°	45°	90°	0°	30°	45°	90°
Top Right 1	(8.69) 9.2		6.6	(.46) .5	(0) 0		18.1	28.4	(16.34) 16.5		16.5	(15.76) 16.0
Bottom Right 2	(.65) .7		.6	(.46) .5	(0) 0		12.1	19.8	(15.85) 16.0		16.0	(15.27) 15.5
Top Left 3	(7.77) 8.3		5.7	(-.46) -.5	(0) 0		18.1	28.4	(15.48) 15.6		15.7	(15.76) 16.0
Bottom Left 4	(-.27) -.2		-.3	(-.46) -.5	(0) 0		12.1	19.9	(14.99) 15.1		15.2	(15.37) 15.5

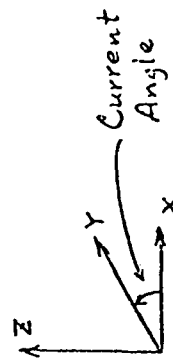
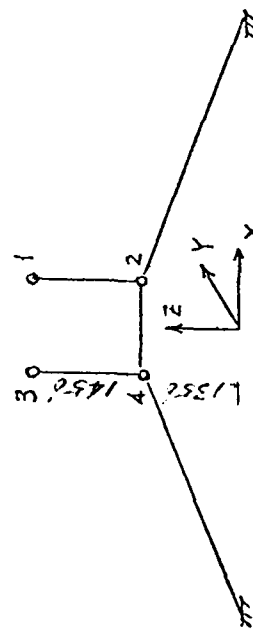


Buoyancy — 3000 lbs.

Cable — 1 $\frac{3}{4}$ " Dia.

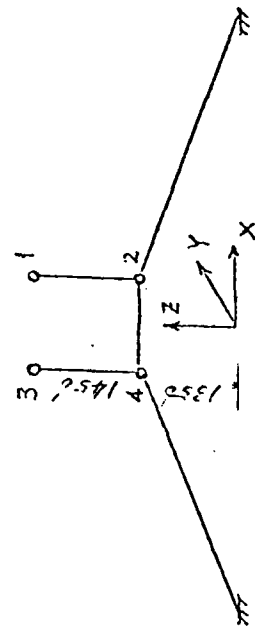
Current — St. Croix

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	30°	45°	90°	0°	30°	45°	90°
Top Right 1	(10.85) 12.5		8.9	(1.46) .4	(10) 0		16.8	(22.66) 24.9	(16.22) 16.4		16.4	(15.85) 16.1
Bottom Right 2	(.61) .7		.6	(1.46) .4	(10) 0		8.5	(11.83) 13.1	(15.75) 15.9		15.9	(15.37) 15.6
Top Left 3	(4.41) 11.5		8.0	(-4.46) -5	(0) 0		16.8	(22.66) 24.9	(15.57) 15.7		15.8	(15.85) 16.1
Bottom Left 4	(-.31) .3		-4	(-4.46) -5	(0) 0		8.5	(11.83) 13.1	(15.16) 15.2		15.4	(15.37) 15.6



Goal post Moor — Deviation in feet from zero position
 Cable — $1\frac{3}{4}$ " Dia. Buoyancy — 4000 lbs
 Current — Uniform = .2 Knots

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	30°	45°	90°	0°	30°	45°	90°
Top Right 1	(6.74) 7.2		5.3	(.64) .6	0		13.3	(20.09) 20.9	(22.90) 23.1		23.3	(22.51) 22.9
Bottom Right 2	(.78) .8		.8	(.64) .6	0		8.8	(14.13) 14.5	(22.24) 22.4		22.6	(21.82) 22.2
Top Left 3	(5.46) 5.9		4.0	(-.64) -.7	0		13.3	(20.09) 20.9	(22.27) 22.5		22.7	(22.51) 22.9
Bottom Left 4	(-.5) -.4		-5.5	(-.64) -.7	0		8.8	(14.33) 14.5	(21.59) 21.8		22.0	(21.82) 22.2



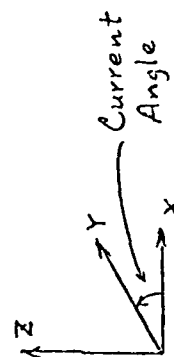
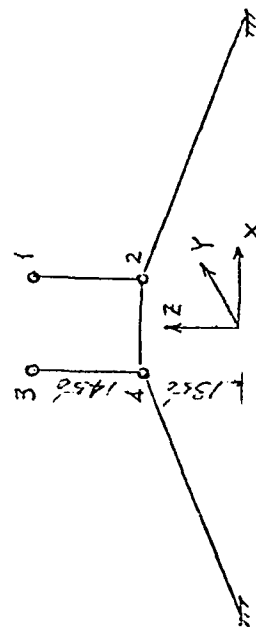
Goal Post Moor — Deviation in feet from zero position

Cable — $1\frac{3}{4}$ " Dia.

Buoyancy — 4000 lbs.

Current — St. Croix

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	30°	45°	90°	0°	30°	45°	90°
Top Right 1	(2.61) 10.3		1.4	(.64) .6	(0) 0		13.2	(16.64) 19.3	23.1		23.1	(22.55) 22.9
Bottom Right 2	(.76) .8		.7	(.64) .6	(0) 0		6.5	(8.86) 9.9	22.4		22.5	(21.87) 22.2
Top Left 3	(7.32) 8.9		6.1	(-.64) -.7	(0) 0		13.1	(16.64) 19.3	22.5		22.7	(22.55) 22.9
Bottom Left 4	(-.52) -.5		-6.6	(-.64) -.7	(0) 0		6.5	(8.86) 9.9	21.9		22.2	(21.87) 22.2



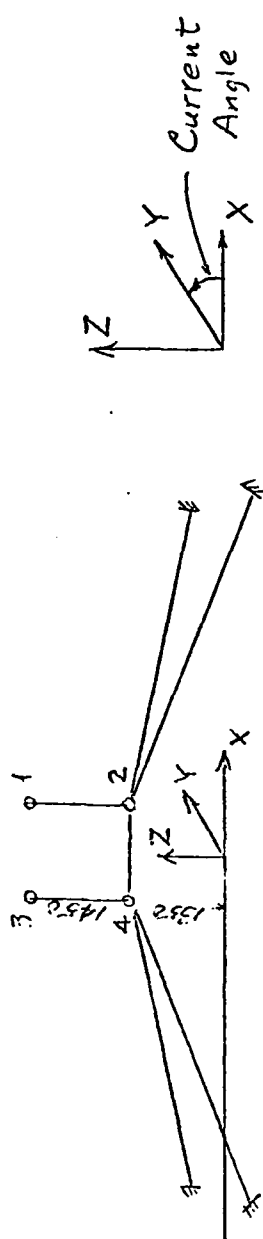
Braced Goal Post Moor — Deviation in feet from zero position

Cable — $1\frac{3}{4}$ " Dia.

Buoyancy — 3000 lbs.

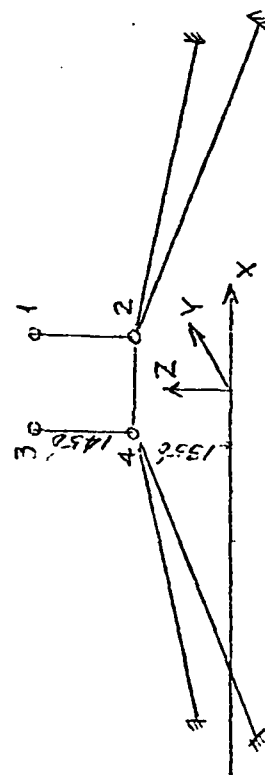
Current — Uniform = .2 Knots

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	60°	90°	0°	60°	90°	0°	60°	90°	0°	90°
Top Right 1	(8.41) 9.4		5.0	(.34) .4	.2		(8.24) 8.9	4.3		(3.15) 3.2		
Bottom Right 2	(.86) .9		.7	(.34) .4	.2		(.19) .4	3.8		(2.67) 2.7		
Top Left 3	(8.23) 8.7		4.3	(-.34) -.3	-.1		(8.24) 8.7	2.1		(3.15) 3.2		
Bottom Left 4	(.19) .2		0	(-.34) -.3	-.1		(.19) .1	1.6		(2.67) 2.7		



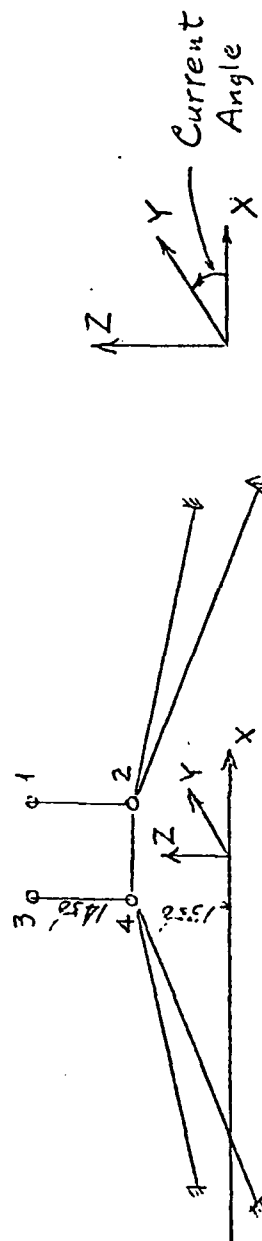
Braced Goal Post Moor — Deviation in feet from zero position
 Cable — $1\frac{3}{4}$ " Dia. Buoyancy — 4000 lbs.
 Current — Uniform = .2 Knots

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	45°	90°	0°	45°	90°	0°	45°
Top Right 1	(6.82) 7.3		5.3	(.46) .5	(0) .2		(6.1) 6.7	(7.45) 7.6		7.5	(6.68) 6.8	
Bottom Right 2	(.87) .9		.8	(.46) .5	(0) .2		(.15) .3	(6.77) 6.9		6.8	(6.0) 6.1	
Top Left 3	(5.88) 6.3		4.4	(-.46) -.5	(0) -.1		(6.1) 6.5	(5.94) 6.0		6.2	(6.68) 6.8	
Bottom Left 4	(-.69) -.1		-.1	(-.46) -.5	(0) -.1		(.15) .1	(5.23) 5.3		5.5	(6.0) 6.1	



Braced Goal Post Moor — Deviation in feet from zero position
 Cable — $1\frac{3}{4}$ " Dia. Buoyancy — 3000 lbs
 Current — St. Croix

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	45°	90°	0°	45°	90°		
Top Right 1	(11.69) 11.9		6.8	(.34) .4	(0) .2		(10.9) 11.4	(3.84) 3.8		4.1	(3.16) 3.2	
Bottom Right 2	(.7) .7		.8	(.34) .4	(0) .2		(.15) .3	(3.37) 3.4		3.6	(2.69) 2.7	
Top Left 3	(10.75) 11.1		6.1	(-.34) -.3	(0) -.1		(10.9) 11.2	(2.49) 2.4		2.3	(3.16) 3.2	
Bottom Left 4	(.03) .1		.1	(-.34) -.3	(0) -.1		(.15) .1	(2.02) 1.9		1.8	(2.69) 2.7	

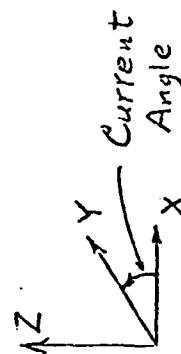
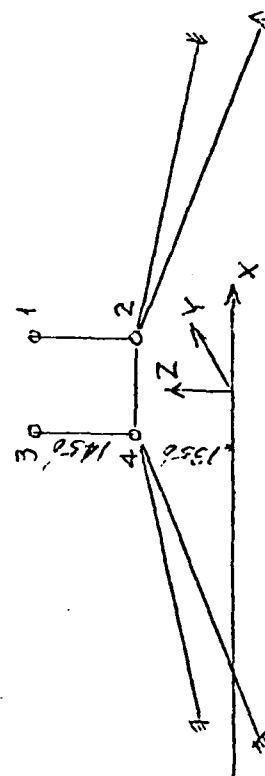


Braced Goal Post Moor — Deviation in feet from zero position

Cable — $1\frac{3}{4}$ " Dia. Buoyancy — 4000 lbs.

Current — St. Croix

Buoy Location	X - Axis				Y - Axis				Z - Axis			
	Current Angle				Current Angle				Current Angle			
	0°	30°	45°	90°	0°	45°	90°	0°	45°	90°	0°	90°
Top Right 1	(8.72) 9.6		6.9	(.48) .5	(0) .2		(8.26) 9.0	(6.49) 7.3		(6.49) 7.2	(6.49) 6.8	
Bottom Right 2	(.74) .8		.7	(.48) .5	(0) .2		(.1) .3	(6.5) 6.6		(6.5) 6.5	(6.5) 6.1	
Top Left 3	(7.94) 8.5		5.9	(-.48) -.5	(0) -.1		(8.26) 8.8	(6.2) 6.3		(6.2) 6.4	(6.2) 6.8	
Bottom Left 4	(-.21) -.2		-.2	(.48) .5	(0) -.1		(.1) .1	(5.52) 5.6		(5.52) 5.7	(5.52) 6.1	



APPENDIX B

Sample of Input and Output

Sample Input
(Tri-moor)

B-2

PROBABLE POSITION OF SINKING SHIP

SHIP NAME: USS ARIZONA POSITION AT: 100-00 00-00
 INDEX NO.: 100-00 00-00 DEVICE: 100-00 00-00
 SURVIVAL NO.: 100-00 00-00 AT: 100-00 00-00

SHIP ANCHORS

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ANCHOR COORDINATES

ANCHOR NO.	COORDINATE	POSITION AT	FORCE COMPONENTS of ANCHOR	DISPLACEMENT FROM
1	100-00 00-00	100-00 00-00	100-00 00-00	100-00 00-00
2	100-00 00-00	100-00 00-00	100-00 00-00	100-00 00-00
3	100-00 00-00	100-00 00-00	100-00 00-00	100-00 00-00

ANCHOR COORDINATES

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ANCHOR COORDINATES

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ANCHOR NO. COORDINATE POSITION AT FORCE COMPONENTS of ANCHOR DISPLACEMENT FROM

ARRAY JUNCTIONS

JUNC. NO.	CABLE AT JUNCTION	TENSION AT JUNCTION	CABLE ANGLES WRT X-AXIS XY-PLANE	X-COORD Y-COORD Z-COORD	DISPLACEMENT FROM NO CURRENT LOC. X-DISP Y-DISP Z-DISP
4	1	147.59	-72.64	2.4 1070.0	0.0 0.0 0.0
4	2	147.59	-72.64	2.4 1070.0	0.0 0.0 0.0
4	3	147.59	-72.64	2.4 1070.0	0.0 0.0 0.0

INDEXED DEVICES ALONG ARRAY CABLES

INDEX NO.	COORDINATE	TENSION AT DEVICE	X-COORD Y-COORD Z-COORD	DISPLACEMENT FROM NO CURRENT LOC. X-DISP Y-DISP Z-DISP
1	147.59	147.59	2.4 1070.0	0.0 0.0 0.0
2	147.59	147.59	2.4 1070.0	0.0 0.0 0.0
3	147.59	147.59	2.4 1070.0	0.0 0.0 0.0

ARRAY EQUILIBRIUM WITH CURRENT FROM 30.00 DEGREES

ARRAY ANCHORS

JUNC. NO. OF ANCHOR	CABLE AT ANCHOR	TENSION AT ANCHOR	X-COORD Y-COORD Z-COORD	DISPLACEMENT FROM NO CURRENT LOC. X-DISP Y-DISP Z-DISP
1	1	147.59	2.4 1070.0	0.0 0.0 0.0
2	2	147.59	2.4 1070.0	0.0 0.0 0.0
3	3	147.59	2.4 1070.0	0.0 0.0 0.0

ARRAY CABLES

CABLE NO.	MAXIMUM TENSION	S-COORD OF	MINIMUM TENSION	S-COORD OF	MAXIMUM DISP.	X-COORD Y-COORD Z-COORD	DISPLACEMENT FROM NO CURRENT LOC. X-DISP Y-DISP Z-DISP
1	1456.1	0.0	676.7	3000.0	7.9	241.7 150.4 1497.8	0.0 0.0 0.0
2	1456.7	3000.0	676.2	0.0	7.9	241.7 150.4 1497.8	0.0 0.0 0.0
3	1167.7	0.0	598.2	3000.0	3.9	241.7 150.4 1497.8	0.0 0.0 0.0

ARRAY JUNCTIONS

JUNC. NO.	CABLE AT JUNCTION	TENSION AT JUNCTION	CABLE ANGLES WRT X-AXIS XY-PLANE	X-COORD Y-COORD Z-COORD	DISPLACEMENT FROM NO CURRENT LOC. X-DISP Y-DISP Z-DISP
4	1	1456.1	-78.89	1.3 2320.0	0.0 0.0 0.0
4	2	1456.7	-78.89	1.3 2320.0	0.0 0.0 0.0
4	3	1167.7	-78.89	1.3 2320.0	0.0 0.0 0.0

ARRAY DATA

CABLE NO.	MAXIMUM TENSION	S-COORD OF JUNCTION	MINIMUM TENSION	S-COORD OF JUNCTION	MAXIMUM DISP.	S-COORD OF JUNCTION	LOCATION OF THIS POINT	NO CURRENT LOG. OF THIS POINT
1	119.5	0.0	700.0	3000.0	3000.0	0.0	X COORD Y COORD Z COORD	X COORD Y COORD Z COORD
2	130.1	3000.0	700.0	0.0	0.0	0.0		
3	130.1	0.0	700.0	3000.0	3000.0	0.0		

JUNC. NO.	CABLE AT JUNCTION	TENSION AT JUNCTION	CABLE ANGLE WRT X-AXIS	XY-PLANE	X COORD	Y COORD	Z COORD	DISPLACEMENT FROM NO CURRENT LOG.
1	1	130.1	147.79	-70.70	0.0	0.0	3000.0	X DISP Y DISP Z DISP
2	2	130.1	-70.00	-70.70	0.0	0.0	3000.0	
3	3	130.1	30.01	-70.70	0.0	0.0	3000.0	

INDEXED DEVICES ALONG ARRAY CABLES

DEVICE CABLE INDEX	S	TENSION AT JUNCTION	DEVICE COORDINATE	X COORD	Y COORD	Z COORD	DISPLACEMENT FROM NO CURRENT LOG.
1	1	130.1	147.79	0.00	0.00	3000.0	X DISP Y DISP Z DISP
2	2	130.1	-70.00	0.00	0.00	3000.0	
3	3	130.1	30.01	0.00	0.00	3000.0	

ARRAY ANCHORS

JUNC. NO. OF ANCHOR	CABLE AT ANCHOR	TENSION AT ANCHOR	X-COMP	Y-COMP	Z-COMP	HORZ. COMP	CABLE ANGLE WRT X-AXIS	XY-PLANE	X COORD	Y COORD	Z COORD	DISPLACEMENT FROM NO CURRENT LOG.
1	1	130.1	277.1	151.7	151.7	151.7	151.7	151.7	151.7	151.7	151.7	X DISP Y DISP Z DISP
2	2	130.1	9.1	252.4	252.4	252.4	252.4	252.4	252.4	252.4	252.4	
3	3	130.1	-104.9	111.7	111.7	111.7	111.7	111.7	111.7	111.7	111.7	

ARRAY CABLES

CABLE NO.	MAXIMUM TENSION	S-COORD OF JUNCTION	MINIMUM TENSION	S-COORD OF JUNCTION	MAXIMUM DISP.	S-COORD OF JUNCTION	LOCATION OF THIS POINT	NO CURRENT LOG. OF THIS POINT
1	119.5	0.0	700.0	3000.0	3000.0	0.0	X COORD Y COORD Z COORD	X COORD Y COORD Z COORD
2	130.1	3000.0	700.0	0.0	0.0	0.0		
3	130.1	0.0	700.0	3000.0	3000.0	0.0		

B-5

1 3

NO. OF CABLES IS 3

CABLE NO.	S-L JUNC	LENGTH	DIAMETER	WEIGHT/LENGTH	BRAC COEFFICIENT	RIGIDITY	CONSTITUTIVE EXPONENT	NO. OF ELEMENTS
1	4	3000.0	1.250	-.200	1.270	1963500.	1.000	50
2	2	3000.0	1.250	-.200	1.270	1963500.	1.000	50
3	4	3000.0	1.250	-.200	1.270	1963500.	1.000	50

PROPERTIES OF THE DEVICES LOCATED AT JUNCTIONS ARE AS FOLLOWS

DEVICE JUNC. NO.	DEVICE WEIGHT	DEVICE BRAC COEFFICIENT	DEVICE FRONTAL AREA
4	4000.00	.120	20.50

TOTAL NO. OF INDEXED DEVICES IS 1

CURRENT FIELD OPTION IS 1

Z COORDINATE OF CURRENT	VELOCITY OF CURRENT AT Z
0.00	.10
400.00	.10
2400.00	.10
2700.00	.24
2700.00	.53

ACCURACY REQUIRED IN CALCULATIONS IS 1.00

DEVICE LOCATION OUTPUT RECORD 1 REFERS TO THIS ARRAY
1 ARRAY EQUILIBRIUM WITH NO CURRENT

ARRAY ANCHORS

JUNC. NO. OF ANCHOR	CABLE AT ANCHOR	TENSION AT ANCHOR	X-COMP	Y-COMP	Z-COMP	HOR. COMP	CONC
1	1	700.0	230.7	173.3	733.1	265.4	265.4
2	2	700.7	0.0	266.5	733.0	264.0	264.0

B-4

CABLE ANCHORS WRT
X AXIS XY PLANE
100.61 100.61
700.00 700.00

FMA OF THE LOAD 101
 LEAST OF THE LOAD 110713

TRANSFER ADDRESS --- DECADE 65502

.403 OF SECONDS 1241002 CM STORAGE USED 65 TALLE NEEDED

1 ERRORS IN PARAMETRIC STUDY SOURCE DECK NUMBER 0

CARD NO.	CARD TYPE	OTHER INFORMATION

NO ERRORS DETECTED
 1 ERRORS IN PARAMETRIC STUDY SOURCE DECK NUMBER 1

CARD NO.	CARD TYPE	OTHER INFORMATION

NO ERRORS DETECTED
 PHYSICAL CHARACTERISTICS OF THE STRUCTURAL CABLE ARRAY

NO. OF ANCHORS IS 3

JUNCTION NO.	X-COORDINATE	Y-COORDINATE	Z-COORDINATE
1	-665.20	394.10	0.00
2	0.00	-760.10	0.00
3	665.20	394.10	0.00

NO. OF JUNCTIONS IN ORIGINAL ARRAY IS 4 ✓

NO. OF CUTS MADE IN ORIGINAL ARRAY IS 2 ✓

JUNCTION NO.	JUNCTION NO.
OF CUT	AT WHICH CUT MADE
5	1
6	3

*Sample Output
 (Tri-moor)*

Appendix C
Source Program Listing

```

      PROGRAM DESADE (INPUT, OUTPUT, TAPE2, TAPE3, TAPE5=INPUT, TAPE6=OUTPUT)
C
C A FORTRAN IV PROGRAM FOR COMPUTING THE STATIC DEFLECTIONS
C OF STRUCTURAL CABLE ARRAYS
C
C BY RICHARD A. SKOP AND JAMES MARK, OCEAN TECHNOLOGY DIVISION,
C NAVAL RESEARCH LABORATORY, WASHINGTON, D. C.
C
      COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,L,ES,FCAB,RCAB,JUMP,
      1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
      COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
      1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
      2OFLG,NIR,THETAS,THETAe,COMPD,THETAJ,NJUNC,RHO,TEST,
      3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
      4WCAB,IDEV,ICHECK,NDEV,NDATC
      DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
      1)
      DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
      DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
      DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
      DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
      DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
      DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
      DIMENSION ICHECK(44)
      DIMENSION DATAT(2150,10)
      EQUIVALENCE (DATAT(1),FEJUNC(1))
      INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
      INTEGER PATH
      REAL IR,IRS
C
C CALL INPUT TO READ DATA AND IDENTIFY ERRORS
C
      ITIME=1
      1000 CALL INPUT
C
C CHECK TO SEE IF ANY ERRORS IN DATA
C
      IF(FATE.NE.0.) GO TO 10000
C
C GET HERE IF NO ERRORS -- PRINT OUT PHYSICAL CHARACTERISTICS OF ARRAY
C
      WRITE(IPRNT,1001)
      1001 FORMAT(///,5X,18HNO ERRORS DETECTED)
      CALL PHSOUT
C
C KMULT IS A MULTIPLIER FOR CHANGING CURRENT ANGLE THETA
C
      KMULT=0
C
C JUMP=0---NO CURRENT    JUMP=1---CURRENT
C
      JUMP=0
C
C GET HERE TO CALCULATE FORCES AND IF SUCCESSIVE APPROXIMATION ROUTINE
C NOT SATISFIED -- ZERO FORCES

```

```

C
2011 DO 2012 J=1,NJUNC
      DO 2012 I=1,3
2012 FEJUNC(I,J)=0.
      DO 2013 N=1,NCAB
        INNN=NNODE(N)
        DO 2013 M=1,INNN
          DO 2013 I=1,3
2013 FCAB(I,M,N)=0.
C
C PICK UP DISCRETE DEVICE DATA FROM TEMPORARY STORAGE TAPE AND
C CALCULATE DEVICE FORCES -- DJNC FORCES ARE STORED IN FEJUNC --
C DCAB FORCES IN FCAB -- EFORCE(I) IS ROUTINE FOR CALCULATING
C DEVICE FORCES IN DIRECTION I
C
2015 READ(INTAPE,1) (DATA(K),K=1,10)
      1 FORMAT(F4.0,A4,8E15.8)
      IF(DATA(2).EQ.TEST(3)) GO TO 2017
      IF(DATA(2).EQ.TEST(4)) GO TO 2019
      IF(DATA(2).EQ.TEST(9)) GO TO 2021
      GO TO 2015
2017 K=DATA(3)
      DO 2018 J=1,3
        I=J
2018 FEJUNC(I,K)=FEJUNC(I,K)+EFORCE(I)
      GO TO 2015
2019 K= DATA(3)
      M= DATA(10)/H(K) + 1
      DO 2020 J=1,3
        I=J
2020 FCAB(I,M,K) = FCAB(I,M,K) + EFORCE(I)
      GO TO 2015
C
C GET HERE AT END OF TAPE
C
2021 REWIND INTAPE
C
C NOW CALCULATE THE FORCE/LENGTH IN DIRECTION I AT NODE M ON CABLE N
C CFORCE(I,M,N) IS ROUTINE FOR DOING THIS -- INTEGRATE BY TRAPEZOIDAL
C RULE OVER SEGMENT TO GET TOTAL FORCE AND ADD TO DCAB FORCES
C
      DO 2022 J=1,NCAB
        K=J
        INNN=NNODE(K)-1
        DO 2022 MM=1,INNN
          M=MM
          DO 2022 II=1,3
            I=II
2022 FCAB(I,M,K)=(CFORCE(I,M,K)+CFORCE(I,M+1,K))*H(K)/2.+FCAB(I,M,K)
C
C ALL FORCES ARE NOW CALCULATED AND EQUILIBRIUM CAN BE DETERMINED
C LEAP = 1 FIRST TIME THROUGH IMAGINARY REACTION ROUTINE
C LEAP = 2 ANY OTHER TIME
C SKIP THIS SECTION IF NO IR'S
C
      IF(INIR.EQ.0) GO TO 2031
      IF(JUMP.EQ.0) CALL START
      LEAP=1
C

```

```

DES056
DES057
DES058
DES059
DES060
DES061
DES062
DES063
DES064
DES065
DES066
DES067
DES068
DES069
DES070
DES071
DES072
DES073
DES074
DES075
DES076
DES077
DES078
DES079
DES080
DES081
DES082
DES083
DES084
DES085
DES086
DES087
DES088
DES089
DES090
DES091
DES092
DES093
DES094
DES095
DES096
DES097
DES098
DES099
DES100
DES101
DES102
DES103
DES104
DES105
DES106
DES107
DES108
DES109
DES110
DES111
DES112
DES113
DES114

```

```

C INITIALIZE DELTA
C
  DELTA=DELTAI
C
C GET HERE TO INITIALIZE TOTAL FORCES AT THE JUNCTIONS, TFJUNC, AND
C IF THE IMAGINARY REACTION ITERATION NOT SATISFIED -- ALSO IF NO IR'S
C
2031 DO 2032 J=1,NJUNC
      DO 2032 I=1,3
2032 TFJUNC(I,J) = FJUNC(I,J)
C
C ADD APPROPRIATE REACTIONS TO TFJUNC -- SKIP THIS SECTION IF NO IR'S
C
  IF(NIR.EQ.0) GO TO 2036
  DO 2035 J=1,NJUNC
  DO 2035 K=1,NIR
    IF( (J.EQ.IRJUNC(K)).OR.(J.EQ.ERJUNC(K)) ) GO TO 2033
  GO TO 2035
2033 DO 2034 I=1,3
2034 TFJUNC(I,J) = TFJUNC(I,J) + IR(I,J)
2035 CONTINUE
C
C TFJUNC IS NOW DETERMINED AND THE REACTIVE FORCES IN THE ARRAY, RCAB,
C CAN BE CALCULATED BY SUMMING FROM THE FREE ENDS TO THE FIXED ANCHOR
C
2036 DO 2047 N=1,NCAB
  INDEX= NCAB+1-N
  K= PATH(INDEX)
  INNN=NNODE(K)
  INLJ=LJUNC(K)
  DO 2041 I=1,3
2041 RCAB(I,INNN,K)=TFJUNC(I,INLJ)
  DO 2044 L=1,NCAB
    IF(LJUNC(K).EQ.ZJUNC(L)) GO TO 2042
  GO TO 2044
2042 DO 2043 I=1,3
2043 RCAB(I,INNN,K)=RCAB(I,INNN,K)+RCAB(I,1,L)
2044 CONTINUE
  DO 2046 MM=1,INNN
  M=INNN+1-MM
  IF(M.EQ.1) GO TO 2047
  DO 2045 I=1,3
2045 RCAB(I,M-1,K)=RCAB(I,M,K)+FCAB(I,M-1,K)
2046 CONTINUE
2047 CONTINUE
C
C ALL REACTIVE FORCES ARE NOW DETERMINED AND THE CONFIGURATION OF THE
C ARRAY, PCAB AND PJUNC, CAN BE FOUND BY INTEGRATING FROM THE FIXED
C ANCHOR TO THE FREE ENDS -- INTEGRATION BY THE TRAPEZOIDAL RULE IS
C AGAIN USED
C
  DO 2054 N=1,NCAB
  X=PATH(N)
  INLN=LJUNC(K)
  INNN=NNODE(K)
  INZJ=ZJUNC(K)
  DO 2051 I=1,3
2051 PCAB(I,1,K)=PJUNC(I,INZJ)
  DO 2052 MM=2,INNN

```

```

DES115
DES116
DES117
DES118
DES119
DES120
DES121
DES122
DES123
DES124
DES125
DES126
DES127
DES128
DES129
DES130
DES131
DES132
DES133
DES134
DES135
DES136
DES137
DES138
DES139
DES140
DES141
DES142
DES143
DES144
DES145
DES146
DES147
DES148
DES149
DES150
DES151
DES152
DES153
DES154
DES155
DES156
DES157
DES158
DES159
DES160
DES161
DES162
DES163
DES164
DES165
DES166
DES167
DES168
DES169
DES170
DES171
DES172
DES173

```

```

      M=MM
      DO 2052 I=1,3
      I=I
2052  PCAB(I,M,K)=PCAB(I,M-1,K)+(EXCAB(M-1,K)*RCAB(I,M-1,K)/TCAB(M-1,K)
      1+EXCAB(M,K)*RCAB(I,M,K)/TCAB(M,K))*H(K)/2.
      DO 2053 I=1,3
2053  PJUNC(I,INLN)=PCAB(I,INNN,K)
2054  CONTINUE
C
C  ARRAY CONFIGURATION NOW DETERMINED -- CHECK TO SEE IF IT SATISFIES
C  GEOMETRIC CONSTRAINTS -- SKIP THIS SECTION IF NO IR'S
C
      IF(NIR.EQ.0) GO TO 2071
C
C  CALCULATE ERROR E
C
      E2=0.
      DO 2061 N=1,NIR
      KEN=ERJUNC(N)
      KIN=IRJUNC(N)
      DO 2061 I=1,3
2061  E2=E2+( PJUNC(I,KEN) - PJUNC(I,KIN) )**2
      E=SQRT(E2)
C
C  COMPARE ERROR TO ACCURACY REQUIREMENTS
C
      IF(E.LE.COMPD/10.) GO TO 2071
C
C  GET HERE IF GEOMETRIC CONSTRAINTS NOT SATISFIED
C
      GO TO (2062,2065),LEAP
C
C  GET HERE FIRST TIME THROUGH IMAGINARY REACTION ROUTINE
C
2062 LEAP=2
C
C  STORE SUCCESSFUL POSITIONS AND REACTIONS
C
2063 ES=E
      DO 2064 N=1,NIR
      KEN=ERJUNC(N)
      KIN=IRJUNC(N)
      DO 2064 I=1,3
      PJUNCS(I,KEN)=PJUNC(I,KEN)
      PJUNCS(I,KIN)=PJUNC(I,KIN)
2064  IRS(I,KIN)=IR(I,KIN)
      GO TO 2066
C
C  GET HERE ANY OTHER TIME THROUGH IMAGINARY REACTION ROUTINE
C  SEE IF ITERATION SUCCESSFUL
C
2065 IF(E.LT.ES) GO TO 2063
C
C  REDUCE DELTA IF NOT SUCCESSFUL INTERATION
C
      DELTA=DELTA/2.
C
C  CALCULATE NEW IMAGINARY AND EQUILIBRATING REACTIONS AND GO BACK TO
C  RECALCULATE ARRAY EQUILIBRIUM

```

```

DES174
DES175
DES176
DES177
DES178
DES179
DES180
DES181
DES182
DES183
DES184
DES185
DES186
DES187
DES188
DES189
DES190
DES191
DES192
DES193
DES194
DES195
DES196
DES197
DES198
DES199
DES200
DES201
DES202
DES203
DES204
DES205
DES206
DES207
DES208
DES209
DES210
DES211
DES212
DES213
DES214
DES215
DES216
DES217
DES218
DES219
DES220
DES221
DES222
DES223
DES224
DES225
DES226
DES227
DES228
DES229
DES230
DES231
DES232

```

```

C
2066 DO 2067 N=1,NIR
      KEN=ERJUNC(N)
      DO 2067 I=1,3
2067 IR(I,KEN)=0.
      DO 2068 N=1,NIR
      KEN=ERJUNC(N)
      KIN=IRJUNC(N)
      DO 2068 I=1,3
      IR(I,KIN)=IRS(I,KIN)+DELTA*(PJUNCS(I,KEN)-PJUNCS(I,KIN))/ES
2068 IR(I,KEN)=IR(I,KEN)-IR(I,KIN)
C
C CHECK CHANGES IN IMAGINARY REACTIONS
C
      DO 2070 N=1,NIR
      KIN=IRJUNC(N)
      DO 2070 I=1,3
      IF(IR(I,KIN).NE.IRS(I,KIN)) GO TO 2031
2070 CONTINUE
C NO CHANGES -- TIME TO QUIT
C
      CALL ERROR
      GO TO 10000
C
C GET HERE IF ACCURACY REQUIREMENTS SATISFIED OR NO IRTS
C OUTPUT EQUILIBRIUM IF NO CURRENT -- IF CURRENT, FIRST CHECK TO SEE
C IF ACCURACY REQUIREMENT SATISFIED BY SUCCESSIVE APPROXIMATIONS
C
2071 JUM=JUMP+1
      GO TO (2072,2075),JUM
2072 IF((OFLG.EQ.0).OR.(OFLG.EQ.2)) CALL STROUT
      IF((OFLG.EQ.1).OR.(OFLG.EQ.2)) CALL TAPOUT
      IF(OFLG.EQ.1) GO TO 2100
      IF(JUM.NE.1) GO TO 2100
      DO 2200 N=1,NCAB
      INNN=NNODE(N)
      DO 2200 M=1,INNN
      DO 2200 I=1,3
      PCABO(I,M,N)=PCAB(I,M,N)
2200 RCABO(I,M,N)=RCAB(I,M,N)
      DO 2201 N=1,NJUNC
      DO 2201 I=1,3
2201 PJUNCO(I,N)=PJUNC(I,N)
C
C APPLY CURRENT IF REQUIRED
C
2100 IF(IVOPT.EQ.0) GO TO 9999
      JUMP=1
      THETA=THETAB + KMULT*THETAS
      IF(THETA.GT.THETAE) GO TO 9999
      KMULT=KMULT+1
C
C STORE EXISTING CONFIGURATION FOR COMPARISON PURPOSES
C
2073 DO 2074 N=1,NCAB
      INNN=NNODE(N)
      DO 2074 M=1,INNN
      DO 2074 I=1,3
2074 PCABE(I,M,N)=PCAB(I,M,N)

```

```

DES233
DES234
DES235
DES236
DES237
DES238
DES239
DES240
DES241
DES242
DES243
DES244
DES245
DES246
DES247
DES248
DES249
DES250
DES251
DES252
DES253
DES254
DES255
DES256
DES257
DES258
DES259
DES260
DES261
DES262
DES263
DES264
DES265
DES266
DES267
DES268
DES269
DES270
DES271
DES272
DES273
DES274
DES275
DES276
DES277
DES278
DES279
DES280
DES281
DES282
DES283
DES284
DES285
DES286
DES287
DES288
DES289
DES290
DES291

```

```

C
C RECALCULATE FORCES
C
      GO TO 2011
C
C GET HERE IF CURRENT -- CHECK SUCCESSIVE APPROXIMATION ACCURACY
C
2075 DO 2077 N=1,NCAS
      DO 2077 M=1,INNN
      U=0.
      DO 2076 I=1,3
2076  U=U+(PCABE(I,M,N)-PCAL(I,M,N))*2
C
C IF NOT ACCURATE STORE CONFIGURATION AND RECALCULATE FORCES
C
      IF (SQRT(U).GT.COMPD) GO TO 2073
2077 CONTINUE
C
C GET HERE IF POSITION ACCURATE AND OUTPUT POSITION
C
      GO TO 2072
C
C GO BACK FOR MORE DATA
C
9999 GO TO 1000
10000 CONTINUE
      WRITE(IPRNT,10001)
10001 FORMAT(1H1)
      END

```

```

DES292
DES293
DES294
DES295
DES296
DES297
DES298
DES299
DES300
DES301
DES302
DES303
DES304
DES305
DES306
DES307
DES308
DES309
DES310
DES311
DES312
DES313
DES314
DES315
DES316
DES317
DES318
DES319
DES320

```

```

SUBROUTINE INPUT
C
C THIS ROUTINE READS ALL DATA AND IDENTIFIES ERRORS IN
C DATA, DECK STRUCTURE, AND ARRAY REPRESENTATION
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETA,COMP,THETA,NJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
DIMENSION DATAT(2150,10)
DIMENSION ITEST(14)
EQUIVALENCE (DATAT(1),FEJUNC(1))
EQUIVALENCE (ITEST(1),TEST(1))
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
C
C THIS IS THE BEGINNING OF THE INPUT SECTION
C INITIALIZE ALL CONSTANTS, FLAGS, ARRAYS, AND COUNTERS
C
IF(ITIME.NE.1) GO TO 999
ITEST(1)=4H IR
ITEST(2)=4H ANC
ITEST(3)=4H DJNC
ITEST(4)=4H DCAB
ITEST(5)=4H CAB
ITEST(6)=4H DEN
ITEST(7)=4H COMP
ITEST(8)=4H ANG
ITEST(9)=4H EOD
ITEST(10)=4H NJNC
ITEST(11)=4H VEL
ITEST(12)=4H NDAT
ITEST(13)=4H LUN
ITEST(14)=4H EOP
PIP=3.14159265/180.
IFEOD=0
NDATC=0
KFLG=0
DO 1000 I=1,1000
1000 IDEV(I)=0
DO 1001 I=1,44
1001 ICHECK(I)=0
DO 1002 I=1,22
1002 ICAB(I)=0
NANC=0

```

```

INP001
INP002
INP003
INP004
INP005
INP006
INP007
INP008
INP009
INP010
INP011
INP012
INP013
INP014
INP015
INP016
INP017
INP018
INP019
INP020
INP021
INP022
INP023
INP024
INP025
INP026
INP027
INP028
INP029
INP030
INP031
INP032
INP033
INP034
INP035
INP036
INP037
INP038
INP039
INP040
INP041
INP042
INP043
INP044
INP045
INP046
INP047
INP048
INP049
INP050
INP051
INP052
INP053
INP054
INP055
INP056
INP057
INP058

```

```

IFRHO=0
IFJNC=0
NIR=0
IROW=1
999 FATE=0.
C
C CHECK LOGICAL UNITS
C
  IF(itime.gt.1) GO TO 1003
  itime=2
  READ 1004,(DATA(I),I=1,10)
1004 FORMAT(F4.0,A4,8F8.0,F5.0,I3)
  IF(DATA(2).NE.TEST(13)) GO TO 9018
  IPRNT=DATA(4)
C
C GENERATE ERROR MESSAGE HEADER
C
998 WRITE(IPRNT,9100) NDATE
9100 FORMAT(1H1,46HEKROKS IN PARAMETRIC STUDY SOURCE DECK NUMBER ,I2,/,
  1 12X,4HCARD,3X,4HCARD,6X,5HOTHER/1H ,4X,4HTYPE,4X,3HNO.,3X,4HTYPE,
  23X,11HINFORMATION )
  IF(IFEOD.EQ.0) GO TO 13
  IF(IFEOD.NE.0) GO TO 12701
C
C READ ONE INPUT RECORD INTO DATA ARRAY
C
1003 CONTINUE
  IF(IFLG.EQ.0) READ(IREAD,1004) (DATA(I),I=1,10),EX,NSEG
  IF(IFLG.EQ.1) READ(IREAD,1104) (DATA(I),I=1,10),EX,NSEG
1104 FORMAT(F4.0,A4,8E15.8,/E12.5,I3)
C
C TYPE AND BRANCH
C
  IF(IFEOD.EQ.0) GO TO 1006
  IF((DATA(2).EQ.TEST(12)).OR.(DATA(2).EQ.TEST(14))) GO TO 1006
  GO TO 9018
1006 DO 1005 I=1,14
  IF(DATA(2).EQ.TEST(I)) GO TO (1,2,3,4,5,6,9018,9018,9,10,9018,
  1 12,9018,14),I
1005 CONTINUE
C
C GET HERE IF CARD UNIDENTIFIABLE
C
  GO TO 9000
C
C GET HERE IF IR CARD READ
C
  1 IF(DATA(3)-DATA(4)) 101,9001,101
  101 DO 102 I=3,4
  IF( (DATA(I).LT. 1.).OR.(DATA(I).GT.44.)) GO TO 9001
  102 CONTINUE
C
C GET HERE IF DATA OK
C COUNT IR AND STORE DATA
C
  NIR=NIR+1
  IRJUNC(NIR)=DATA(3)
  ERJUNC(NIR)=DATA(4)
  DO 103 N=1,NIR

```

```

INP059
INP060
INP061
INP062
INP063
INP064
INP065
INP066
INP067
INP068
INP069
INP070
INP071
INP072
INP073
INP074
INP075
INP076
INP077
INP078
INP079
INP080
INP081
INP082
INP083
INP084
INP085
INP086
INP087
INP088
INP089
INP090
INP091
INP092
INP093
INP094
INP095
INP096
INP097
INP098
INP099
INP100
INP101
INP102
INP103
INP104
INP105
INP106
INP107
INP108
INP109
INP110
INP111
INP112
INP113
INP114
INP115
INP116
INP117

```

```

IF( (IRJUNC(NIR).EQ.ERJUNC(N)) GO TO 9004
IF( (IRJUNC(NIR).EQ.IRJUNC(N) ) .AND. (NIR.NE.N) ) GO TO 9004
IF( (ERJUNC(NIR).EQ.IRJUNC(N)) GO TO 9004
103 CONTINUE
GO TO 8000
C
C GET HERE IF ANC CARD READ
C
2 INDEX=DATA(3)
IF( (DATA(3).LT.1.) .OR. (DATA(3).GT.44.) ) GO TO 9001
IF( (ICHECK(INDEX).NE.0) GO TO 9002
C
C GET HERE IF DATA OK -- COUNT ANCHOR AND STORE DATA
C
NANC=NANC+1
ANJUNC(NANC)=DATA(3)
ICHECK(INDEX)=1
22 DO 21 I=1,3
21 PJUNC(I,INDEX)=DATA(I+3)
GO TO 8000
C
C GET HERE IF DJNC CARD READ
C
3 IF( (DATA(4).EQ.1.) .OR. (DATA(4).EQ.3.) ) GO TO 9001
IF( (DATA(4).EQ.2.) .AND. (DATA(5).LT.1.) ) GO TO 9001
IF( (DATA(5).GT.1000.) GO TO 9001
IF( (DATA(4).GT.4.) GO TO 9001
IF( (DATA(4).LT.1.) GO TO 9001
IF( (DATA(3).LT.1.) .OR. (DATA(3).GT.44.) ) GO TO 9001
IF( (DATA(4).EQ.4.) .AND. (DATA(5).NE.0.) ) GO TO 9001
IF( (DATA(7).LT.0.) .OR. (DATA(8).LT.0.) ) GO TO 9001
IF( (DATA(4).EQ.4.) GO TO 8000
INDEX=DATA(5)
IF( (IDEV(INDEX).NE.0) GO TO 9005
IDEV(INDEX)=1
GO TO 8000
C
C GET HERE IF DCAB CARD READ
C
4 DO 41 I=1,4
ID=DATA(4)
IF( (ID.EQ.1) GO TO 42
41 CONTINUE
GO TO 9001
42 IF( ((DATA(4).EQ.3.) .OR. (DATA(4).EQ.4.) .AND. (DATA(5).NE.0.) ) GO TO 9001
IF( (DATA(3).LT.1.) .OR. (DATA(3).GT.22.) ) GO TO 9001
IF( ((DATA(4).EQ.1.) .OR. (DATA(4).EQ.2.) .AND. ((DATA(5).LT.1.) .OR. (DATA(5).GT.1000.) ) ) GO TO 9001
IF( (DATA(7).LT.0.) .OR. (DATA(8).LT.0.) ) GO TO 9001
IF( ((DATA(4).EQ.2.) .OR. (DATA(4).EQ.4.) .AND. (DATA(9).NE.0.) ) GO TO 9001
IF( ((DATA(4).EQ.1.) .OR. (DATA(4).EQ.3.) .AND. (DATA(9).LE.0.) ) GO TO 9001
IF( (DATA(10).LT.0.) GO TO 9001
IF( (DATA(4).EQ.3.) .OR. (DATA(4).EQ.4.) ) GO TO 8000
INDEX=DATA(5)
IF( (IDEV(INDEX).NE.0) GO TO 9005
IDEV(INDEX)=1

```

INP118
 INP119
 INP120
 INP121
 INP122
 INP123
 INP124
 INP125
 INP126
 INP127
 INP128
 INP129
 INP130
 INP131
 INP132
 INP133
 INP134
 INP135
 INP136
 INP137
 INP138
 INP139
 INP140
 INP141
 INP142
 INP143
 INP144
 INP145
 INP146
 INP147
 INP148
 INP149
 INP150
 INP151
 INP152
 INP153
 INP154
 INP155
 INP156
 INP157
 INP158
 INP159
 INP160
 INP161
 INP162
 INP163
 INP164
 INP165
 INP166
 INP167
 INP168
 INP169
 INP170
 INP171
 INP172
 INP173
 INP174
 INP175
 INP176

```

      GO TO 8000
C
C GET HERE IF CAB CARD READ
C
      5 INDEX=DATA(3)
      IF( (DATA(3).LT.1.).OR.(DATA(3).GT.22.)) GO TO 9001
      IF( DATA(4).EQ.DATA(5) ) GO TO 9001
      IF( (DATA(4).GT.44.).OR.(DATA(5).GT.44.)) GO TO 9001
      IF( (DATA(4).LT.1.).OR.(DATA(5).LT.1.)) GO TO 9001
      IF( (DATA(7).LE.0.).OR.(DATA(8).LE.0.).OR.(DATA(9).LE.0.)
      1 .OR.(EX.LT.0.).OR.(DATA(10).LT.0.)) GO TO 9001
      IF((DATA(10).EQ.0.).AND.(EX.NE.0.)) GO TO 9001
      IF((DATA(10).NE.0.).AND.(EX.EQ.0.)) GO TO 9001
      IF( INSEG.LT.1).OR.(INSEG.GT.50) ) GO TO 9001
      IF( ICAB(INDEX).NE.0) GO TO 9003
      ICAB(INDEX)=1
      INDEX=DATA(5)
      IF( ICHECK(INDEX).NE.0) GO TO 9002
      ICHECK(INDEX)=1
C
C GET HERE IF DATA OK
C
      INDEX=DATA(3)
      ZJUNC(INDEX)= DATA(4)
      LJUNC(INDEX)= DATA(5)
      51 NNODE(INDEX)= NSEG+1
      WCAB(INDEX)= DATA(6)
      CDCAB(INDEX)=DATA(7)
      DCAB(INDEX)=DATA(8)
      H(INDEX)= DATA(9)/NSEG
      ECICAB(INDEX)= DATA(10)
      EXPCAB(INDEX)= EX
      GO TO 8000
C
C GET HERE IF DEN CARD READ
C
      6 IF(DATA(3).LE.0.) GO TO 9001
      IFRHO=IFRHO+1
      IF( IFRHO.GT.1 ) GO TO 9006
      RHO= DATA(3)
      GO TO 1003
C
C GET HERE IF EOD CARD READ
C
      9 IF EOD=IFEOD+1
      DO 90 J=1,10
      90 DATAT(IROW,J)=DATA(J)
      IRMAX=IROW
      IF(FATE.NE.0.) RETURN
      IF( IFLG.EQ.1) GO TO 91
      GO TO 500
      91 IREAD=ISAV1
      IFLG=0
      GO TO 500
C
C GET HERE IF NJNC CARD READ
C
      10 IF( (DATA(3).LT.2.).OR.(DATA(3).GT.44.)) GO TO 9001
      IFJNC=IFJNC+1

```

```

INP177
INP178
INP179
INP180
INP181
INP182
INP183
INP184
INP185
INP186
INP187
INP188
INP189
INP190
INP191
INP192
INP193
INP194
INP195
INP196
INP197
INP198
INP199
INP200
INP201
INP202
INP203
INP204
INP205
INP206
INP207
INP208
INP209
INP210
INP211
INP212
INP213
INP214
INP215
INP216
INP217
INP218
INP219
INP220
INP221
INP222
INP223
INP224
INP225
INP226
INP227
INP228
INP229
INP230
INP231
INP232
INP233
INP234
INP235

```

IF (IFJNC.GT.1) GO TO 9006	INP236
NOJUNC=DATA(3)	INP237
GO TO 1003	INP238
C	INP239
C GET HERE IF NDAT CARD READ	INP240
C	INP241
12 NDATC=NDATC+1	INP242
IF (NDATC.NE.1) GO TO 9018	INP243
GO TO 998	INP244
12701 IFCOMP=0	INP245
INDAT=0	INP246
IFVEL=0	INP247
IFANG=0	INP248
IF ((DATA(3).LT.0.).OR.(DATA(3).GT.2.)) GO TO 9001	INP249
ILOPT=DATA(3)	INP250
IF (NDATC.EQ.1) GO TO 1270	INP251
IF (ILOPT.EQ.0) GO TO 1201	INP252
IF (KFLG.EQ.0) GO TO 1271	INP253
IF (ILOPT.EQ.KCUR) GO TO 1201	INP254
GO TO 9117	INP255
1270 IF (ILOPT.EQ.0) GO TO 1275	INP256
1271 KFLG=1	INP257
KCUR=ILOPT	INP258
GO TO 1201	INP259
1275 KFLG=0	INP260
C	INP261
C READ ONE INPUT RECORD FROM PARAMETRIC STUDY SOURCE DECK	INP262
C	INP263
1201 READ (IREAD,1004) (DATN(I),I=1,10),EXX,NNSEG	INP264
DO 1209 I=1,14	INP265
IF (DATN(2).EQ.TEST(I)) GO TO (9018,1208,1208,1208,1208,9018,1207,	INP266
1 1203,1260,9018,1202,9018,9018,9018),I	INP267
1209 CONTINUE	INP268
CALL SWITCH	INP269
GO TO 9000	INP270
C	INP271
C GET HERE IF VEL CARD READ	INP272
C	INP273
1202 IFVEL=IFVEL+1	INP274
CALL SWITCH	INP275
IF (ILOPT.EQ.0) GO TO 9016	INP276
IF (IFVEL.NE.1) GO TO 1225	INP277
NVSEG=0	INP278
DO 1224 I=1,25	INP279
ZVEL(I)=0.	INP280
1224 VELZ(I)=0.	INP281
1225 NVSEG=NVSEG+1	INP282
IF (NVSEG.GT.25) GO TO 9006	INP283
ZVEL(NVSEG)=DATA(3)	INP284
DO 111 K=1,NVSEG	INP285
IF ((ZVEL(NVSEG).EQ.ZVEL(K)).AND.(K.NE.NVSEG)) GO TO 9006	INP286
111 CONTINUE	INP287
VELZ(NVSEG)=DATA(4)	INP288
GO TO 1201	INP289
C	INP290
C GET HERE IF ANG CARD READ	INP291
C	INP292
1203 IFANG=IFANG+1	INP293
CALL SWITCH	INP294

```

IF(IVOPT.EQ.0) GO TO 9018
IF(IFANG.EQ.1) NANG=0
IF(DATA(4).LE.0.) GO TO 9001
IF( DATA(5).LT.DATA(3) ) GO TO 9001
THETAB=DATA(3)
THETAS= DATA(4)
THETAE= DATA(5)
NANG=NANG+1
IF( NANG.GT.1) GO TO 9006
GO TO 1201

C
C GET HERE IF COMP CARD READ
C
1207 IFCOMP=IFCOMP+1
CALL SWTCH
IF(IFCOMP.EQ.1) NCOMP=0
IF(DATA(3).LE.0.) GO TO 9001
NCOMP= NCOMP+1
IF( NCOMP.GT.1) GO TO 9006
COMPD=DATA(3)
GO TO 1201

C
C GET HERE IF PARAMETERS ARE BEING CHANGED
C LOCATE RECORD TO BE CHANGED AND BRANCH
C
1208 IF(INDAT.NE.0) GO TO 12081
READ(INTAPE,31) ((DATAT(I,J),J=1,10),I=1,IRMAX)
REWIND INTAPE
INDAT=1
12081 DO 1290 I=1,IRMAX
IROW=I
IF((DATN(1).EQ.DATAT(IROW,1)).AND.(DATN(2).EQ.DATAT(IROW,2)))
1 GO TO 1205
IF(DATAT(IROW,2).EQ.TEST(9)) GO TO 9017
1290 CONTINUE
1205 DO 12051 J=1,10
12051 DATA(J)=DATAT(IROW,J)
DO 1206 I=1,5
IF(DATN(2).EQ.TEST(1)) GO TO (1206,1210,1220,1230,1240),I
1206 CONTINUE

C
C GET HERE IF ANC CARD READ
C
1210 IF(DATA(3).NE.DATN(3)) GO TO 9017
CALL SWTCH
INDEX=DATA(3)
GO TO 22

C
C GET HERE IF DJNC CARD READ
C
1220 DO 1222 I=3,5
IF(DATA(I).NE.DATN(I)) GO TO 9017
1222 CONTINUE
CALL SWTCH
IF((DATA(7).LT.0.).OR.(DATA(8).LT.0.)) GO TO 9001
GO TO 8000

C
C GET HERE IF DCAB CARD READ
C

```

```

INP295
INP296
INP297
INP298
INP299
INP300
INP301
INP302
INP303
INP304
INP305
INP306
INP307
INP308
INP309
INP310
INP311
INP312
INP313
INP314
INP315
INP316
INP317
INP318
INP319
INP320
INP321
INP322
INP323
INP324
INP325
INP326
INP327
INP328
INP329
INP330
INP331
INP332
INP333
INP334
INP335
INP336
INP337
INP338
INP339
INP340
INP341
INP342
INP343
INP344
INP345
INP346
INP347
INP348
INP349
INP350
INP351
INP352
INP353

```

```

1230 IF((DATA(3).NE.DATN(3)).OR.(DATA(5).NE.DATN(5))) GO TO 9017      INP354
DO 1231 I=1,3,2                                                         INP355
F=I                                                                       INP356
G=I+1                                                                     INP357
IF(((DATA(4).EQ.F).OR.(DATA(4).EQ.G)).AND.((DATN(4).EQ.F).OR.(DATN(4).EQ.G))) GO TO 1232 INP358
1231 CONTINUE                                                            INP359
GO TO 9017                                                                INP360
1232 CALL SWITCH                                                         INP361
IF( ((DATA(4).EQ.2.).OR.(DATA(4).EQ.4.)).AND.(DATA(9).NE.0.) )      INP362
1 GO TO 9001                                                            INP363
IF( (DATA(7).LT.0.).OR.(DATA(8).LT.0.) ) GO TO 9001                  INP364
IF( ((DATA(4).EQ.1.).OR.(DATA(4).EQ.3.)).AND.(DATA(9).LE.0.) )      INP365
1GO TO 9001                                                            INP366
IF( DATA(10).LT.0. ) GO TO 9001                                       INP367
GO TO 8000                                                              INP368
C                                                                        INP369
C GET HERE IF CAB CARD READ                                           INP370
C                                                                        INP371
1240 DO 1241 I=3,5                                                      INP372
IF(DATA(I).NE.DATN(I)) GO TO 9017                                       INP373
1241 CONTINUE                                                            INP374
CALL SWITCH                                                             INP375
EX=EXX                                                                    INP376
NSEG=NNSEG                                                              INP377
INDEX=DATA(3)                                                           INP378
IF( (DATA(7).LE.0.).OR.(DATA(8).LE.0.).OR.(DATA(9).LE.0.) )          INP379
1 .OR.(EX.LT.0.).OR.(DATA(10).LT.0.) ) GO TO 9001                     INP380
IF((DATA(10).EQ.0.).AND.(EX.NE.0.)) GO TO 9001                        INP381
IF((DATA(10).NE.0.).AND.(EX.EQ.0.)) GO TO 9001                        INP382
IF( NSEG.LT.1).OR.(NSEG.GT.50) ) GO TO 9001                           INP383
GO TO 51                                                                INP384
C                                                                        INP385
C GET HERE IF EOD CARD READ                                           INP386
C                                                                        INP387
1260 IFEOD=IFEOD+1                                                      INP388
IF(FATE.NE.0.) RETURN                                                  INP389
GO TO 501                                                              INP390
C                                                                        INP391
C GET HERE IF LUN CARD READ                                           INP392
C                                                                        INP393
13 INTAPE=DATA(5)                                                       INP394
IFLG=DATA(6)                                                            INP395
IF((IFLG.LT.0).OR.(IFLG.GT.1)) GO TO 9001                              INP396
ISAV1=DATA(3)                                                           INP397
IF(IFLG.EQ.0) IREAD=DATA(3)                                             INP398
IF(IFLG.EQ.1) IREAD=DATA(7)                                             INP399
OFLG=DATA(8)                                                            INP400
IF((OFLG.LT.0).OR.(OFLG.GT.2)) GO TO 9001                             INP401
IF(OFLG.EQ.0) OUTAPE=DATA(4)                                            INP402
IF(OFLG.NE.0) OUTAPE=DATA(9)                                           INP403
IF((IPRNT.EQ.IREAD).OR.(IPRNT.EQ.INTAPE).OR.((OFLG.NE.0).AND.      INP404
1(IPRNT.EQ.OUTAPE)).OR.(INTAPE.EQ.IREAD).OR.(INTAPE.EQ.OUTAPE).OR. INP405
2(IREAD.EQ.OUTAPE).OR.((IFLG.EQ.1).AND.(IREAD.EQ.ISAV1)))GO TO 9001 INP406
GO TO 1003                                                             INP407
14 IF((IFEOD-NDATC).NE.1) GO TO 9018                                    INP408
IF(OFLG.EQ.0) GO TO 141                                                 INP409
WRITE(OUTAPE,142) TEST(14)                                             INP410
142 FORMAT(A4,I4,3E15.8)                                              INP411

```

```

141 WRITE(IPRNT,140)
140 FORMAT(1H1,18HANALYSIS COMPLETED)
    FATE=1.
    RETURN
C
C CHECK TO SEE IF SUSPENDED ARRAY SOURCE DECK COMPLETE
C
500 IF((IFJNC.EQ.0).OR.(IFRHO.EQ.0).OR.(NANC.EQ.0)) GO TO 9007
510 IF(FATE.EQ.0.) GO TO 2000
    RETURN
C
C CHECK TO SEE IF PARAMETRIC STUDY SOURCE DECK COMPLETE
C
501 IF((IVOPT.EQ.0).OR.(IVOPT.EQ.2)) GO TO 5101
    NZL=0
    IF(NVSEG.EQ.0) GO TO 9008
    IF(IFVEL.EQ.0) GO TO 549
C
C SORT VELOCITY PROFILE BY Z-COORDINATE
C
550 DO 555 I=1,NVSEG
    K=I
    DO 555 J=K,NVSEG
        IF(ZVEL(I).LE.ZVEL(J)) GO TO 555
        TEMP=ZVEL(I)
        ZVEL(I)=ZVEL(J)
        ZVEL(J)=TEMP
        TEMP=VELZ(I)
        VELZ(I)=VELZ(J)
        VELZ(J)=TEMP
555 CONTINUE
549 INDEX=ANJUNC(1)
    ZANCM=PJUNC(3,INDEX)
    IF(NANC.EQ.1) GO TO 511
    DO 508 N=2,NANC
        INDEX=ANJUNC(N)
        IF(PJUNC(3,INDEX).LT.ZANCM) ZANCM=PJUNC(3,INDEX)
508 CONTINUE
511 DO 509 N=1,NVSEG
    IF(ZVEL(N).LE.ZANCM) NZL=NZL+1
509 CONTINUE
    IF((NZL.EQ.0).OR.(NANC.EQ.0)) GO TO 9008
5101 IF (INCOMP.EQ.0) GO TO 9008
    IF (INDAT.NE.0) GO TO 4000
505 CONTINUE
    RETURN
C
C CHECK ON CONTINUITY OF CABLE NUMBERING AND COUNT CABLES
C
2000 NCAB=ICAB(22)
    DO 2001 N=1,21
        NCAB=NCAB+ICAB(N)
        J=ICAB(N)-ICAB(N+1)
        IF (J.LT.0) GO TO 9009
2001 CONTINUE
    IF (NCAB.EQ.0) GO TO 9009
C
C CHECK ON CONTINUITY OF JUNCTION NUMBERING AND COUNT JUNCTIONS
C

```

```

INP413
INP414
INP415
INP416
INP417
INP418
INP419
INP420
INP421
INP422
INP423
INP424
INP425
INP426
INP427
INP428
INP429
INP430
INP431
INP432
INP433
INP434
INP435
INP436
INP437
INP438
INP439
INP440
INP441
INP442
INP443
INP444
INP445
INP446
INP447
INP448
INP449
INP450
INP451
INP452
INP453
INP454
INP455
INP456
INP457
INP458
INP459
INP460
INP461
INP462
INP463
INP464
INP465
INP466
INP467
INP468
INP469
INP470
INP471

```

2002 NJUNC=ICHECK(44)	INP472
DO 2003 N=1,43	INP473
NJUNC=NJUNC+ICHECK(N)	INP474
J=ICHECK(N)-ICHECK(N+1)	INP475
IF (J.LT.0) GO TO 9010	INP476
2003 CONTINUE	INP477
C	INP478
C CHECK ON CONTINUITY OF DEVICE NUMBERING AND COUNT DEVICES	INP479
C	INP480
2004 NDEV=IDEV(1000)	INP481
DO 2005 N=1,999	INP482
NDEV=NDEV+IDEV(N)	INP483
J=IDEV(N)-IDEV(N+1)	INP484
IF (J.LT.0) GO TO 9011	INP485
2005 CONTINUE	INP486
2007 IF (FATE.EQ.0.) GO TO 3000	INP487
RETURN	INP488
C	INP489
C GET HERE IF ARRAY NUMBERED CORRECTLY	INP490
C CHECK TO SEE IF NIR CONSISTENT	INP491
C	INP492
3000 NIRC=NCAB+NANC-NOJUNC	INP493
IF (NIR.NE.NIRC) GO TO 9013	INP494
3001 IF (FATE.EQ.0.) GO TO 4000	INP495
RETURN	INP496
C	INP497
C GET HERE TO MAKE FINAL CHECK ON INTAPE	INP498
C	INP499
4000 I=1	INP500
4009 DO 4001 J=1,10	INP501
4001 DATA(J)=DATAT(I,J)	INP502
I=I+1	INP503
IF (DATA(2).EQ.TEST(1)) GO TO 4002	INP504
IF (DATA(2).EQ.TEST(3)) GO TO 4003	INP505
IF (DATA(2).EQ.TEST(4)) GO TO 4004	INP506
IF (DATA(2).EQ.TEST(5)) GO TO 4005	INP507
IF (DATA(2).EQ.TEST(9)) GO TO 4006	INP508
GO TO 4009	INP509
C	INP510
C GET HERE FOR IR	INP511
C	INP512
4002 ID1=DATA(3)	INP513
ID2=DATA(4)	INP514
IF ((ID2.GT.NOJUNC).OR.(ID1.LE.NOJUNC).OR.(ID1.GT.NJUNC)) GO TO 9014	INP515
GO TO 4009	INP516
C	INP517
C GET HERE FOR DJNC	INP518
C	INP519
4003 ID=DATA(3)	INP520
IF (ID.GT.NJUNC) GO TO 9014	INP521
GO TO 4009	INP522
C	INP523
C GET HERE FOR DCAB	INP524
C	INP525
4004 ID=DATA(3)	INP526
IF (ID.GT.NCAB) GO TO 9014	INP527
RL=H(ID)*(NNODE(ID)-1)	INP528
IF (DATA(10).GE.RL) GO TO 9014	INP529
GO TO 4009	INP530

C	GET HERE FOR CAB	INP531
C		INP532
C	4005 ID=DATA(4)	INP533
	IF (ID.GT.NJUNC) GO TO 9014	INP534
	GO TO 4009	INP535
C		INP536
C	GET HERE FOR EOD	INP537
C		INP538
C	4006 WRITE(INTAPE,31) ((DATAT(I,J),J=1,10),I=1,IRMAX)	INP539
	31 FORMAT(F4.0,A4,8E15.8)	INP540
	REWIND INTAPE	INP541
	IF (INDATC.NE.0) RETURN	INP542
	IF (FATE.EQ.0.) GO TO 5000	INP543
	RETURN	INP544
C		INP545
C	GET HERE IF ALL OK AND CALCULATE PATH	INP546
C	K IS CABLE COUNTER	INP547
C		INP548
C	5000 K=0	INP549
C		INP550
C	LOOP=1 LOOKING FOR CABLES LEAVING ANCHORS	INP551
C		INP552
	LOOP=1	INP553
C		INP554
C	JMINP REMEMBERS FIRST VALUE OF K ON A LEVEL OF TREE	INP555
C		INP556
C	JMINP=1	INP557
C		INP558
C	JMAX REMEMBERS NUMBER OF CABLES ON A LEVEL OF TOPOGRAPHIC TREE	INP559
C		INP560
	JMAX=NANC	INP561
	IF (LOOP.EQ.1) GO TO 5002	INP562
5001	JMAX=K	INP563
	IF ((LOOP.EQ.1).AND.(K.NE.1)) GO TO 9015	INP564
	IF ((LOOP.EQ.2).AND.(JMIN.EQ.JMINP)) GO TO 9015	INP565
	IF (K.EQ.NCAB) GO TO 5008	INP566
	LOOP=2	INP567
5002	JMIN=JMINP	INP568
	DO 5009 J=JMIN,JMAX	INP569
C		INP570
C	LOOKING FOR CABLES LEAVING A JUNCTION	INP571
C		INP572
	DO 5007 N=1,NCAB	INP573
	GO TO (5003,5004),LOOP	INP574
5003	IF (ZJUNC(N).EQ.ANJUNC(J)) GO TO 5005	INP575
	GO TO 5007	INP576
5004	IPATHJ=PATH(J)	INP577
	IF (ZJUNC(N).EQ.LJUNC(IPATHJ)) GO TO 5005	INP578
	GO TO 5007	INP579
C		INP580
C	GET HERE IF CABLE N STARTS AT JUNCTION M	INP581
C		INP582
	5005 K=K+1	INP583
	PATH(K)=N	INP584
	GO TO (5007,5006),LOOP	INP585
C		INP586
C	REMEMBER HERE FIRST VALUE OF K ON TREE LEVEL	INP587
C		INP588
		INP589

5006 IF(JMAX.EQ.(K-1)) JMINP=K	INP590
5007 CONTINUE	INP591
5009 CONTINUE	INP592
GO TO 5001	INP593
5008 CONTINUE	INP594
IF(FATE.NE.0.) RETURN	INP595
WRITE(IPRNT,5010)	INP596
5010 FORMAT(///,5X,18HNO ERRORS DETECTED)	INP597
GO TO 1003	INP598
C	INP599
C PUT DATA INTO DATA ARRAY	INP600
C	INP601
8000 IF(IFEOD.NE.0) GO TO 8003	INP602
IF(((DATA(2).EQ.TEST(3)).OR.(DATA(2).EQ.TEST(4)))	INP603
1.AND.(DATA(5).NE.0.)) GO TO 8001	INP604
GO TO 8003	INP605
8001 ID=DATA(5)	INP606
DO 8002 J=1,10	INP607
DATAT(IROW,J)=DATAT(ID,J)	INP608
8002 DATAT(ID,J)=DATA(J)	INP609
GO TO 8005	INP610
8003 DO 8004 J=1,10	INP611
8004 DATAT(IROW,J)=DATA(J)	INP612
IF(IFEOD.NE.0) GO TO 1201	INP613
8005 IROW=IROW+1	INP614
IF(IROW.LE.2150) GO TO 1003	INP615
WRITE(IPRNT,8006)	INP616
8006 FORMAT(6X,2H17,18X,45HCOMMON/B1/ BOUND EXCEEDED. SEE USERS MANUAL	INP617
1.)	INP618
GO TO 1003	INP619
C	INP620
C THIS SECTION GENERATES ALL ERROR MESSAGES	INP621
C	INP622
9116 FATE=1.	INP623
WRITE(IPRNT,9500) IER,DATA(1),DATA(2)	INP624
9500 FORMAT(6X,12,4X,F4.0,3X,A4)	INP625
IF(IER.LQ.12) GO TO 4009	INP626
IF(IER.EQ.16) RETURN	INP627
IF(NDATC.EQ.0) GO TO 1003	INP628
GO TO 1201	INP629
9000 IER=0	INP630
GO TO 9116	INP631
9001 IER=1	INP632
GO TO 9116	INP633
9002 IER=2	INP634
GO TO 9116	INP635
9003 IER=3	INP636
GO TO 9116	INP637
9004 IER=4	INP638
GO TO 9116	INP639
9005 IER=5	INP640
GO TO 9116	INP641
9006 IER=6	INP642
GO TO 9116	INP643
9007 IER=7	INP644
WRITE(IPRNT,9501) IER,IFUNC,IFRHO,NANC	INP645
9501 FORMAT(6X,12,18X,513)	INP646
FATE=1	INP647
GO TO 510	INP648

9008 IER=14	INP649
FATE=1	INP650
WRITE(IPRNT,9501) IER, NCOMP, IVOPT, NVSEG, NZL, NANG	INP651
GO TO 505	INP652
9009 IER=8	INP653
FATE=1	INP654
WRITE(IPRNT,9502) IER, (ICAB(I), I=1, 22)	INP655
GO TO 2002	INP656
9010 IER=9	INP657
FATE=1	INP658
WRITE(IPRNT,9502) IER, (ICHECK(I), I=1, 44)	INP659
9502 FORMAT(6X, I2, 18X, 44I2)	INP660
GO TO 2004	INP661
9011 IER=10	INP662
FATE=1	INP663
WRITE(IPRNT,9504) IER, (IDEV(I), I=1, 1000)	INP664
9504 FORMAT(6X, I2, 12X, 100I1, 9(/, 20X, 100I1))	INP665
GO TO 2007	INP666
9013 IER=11	INP667
FATE=1	INP668
WRITE(IPRNT,9501) IER, NCAB, NANC, NOJUNC, NIRC, NIR	INP669
GO TO 3001	INP670
9014 IER=12	INP671
GO TO 9116	INP672
9015 IER=13	INP673
FATE=1	INP674
WRITE(IPRNT,9507) IER	INP675
GO TO 5008	INP676
9507 FORMAT(6X, I2, 18X, 94HIMPROPER ARRAY REDUCTION OR JUNCTION NUMBERING	INP677
1. CHECK TREE REPRESENTATION OF ARRAY (SEE ARRAY, /, 20X, 83HREDUCTION	INP678
2N SECTION OF USERS MANUAL) AGAINST JUNCTION NUMBERING ON ANC AND C	INP679
3AB CARDS.)	INP680
9017 DATA(1)=DATN(1)	INP681
DATA(2)=DATN(2)	INP682
9117 IER=15	INP683
GO TO 9116	INP684
9018 IER=16	INP685
IF(IFEOD.EQ.0) GO TO 9116	INP686
DATA(1)=DATN(1)	INP687
DATA(2)=DATN(2)	INP688
GO TO 9116	INP689
END	INP690

SUBROUTINE PHSOUT

```

C
C THIS ROUTINE GENERATES INFORMATION CONCERNING THE PHYSICAL
C CHARACTERISTICS OF THE STRUCTURAL CABLE ARRAY
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,PHS001
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO,PHS002
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,PHS003
1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,PHS004
2OFLG,NIR,THETAS,THETAE,COMPU,THETAB,NJUNC,RHO,TEST,PHS005
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,PHS006
4WCAB,IDLV,ICHECK,NDEV,NDATC,PHS007
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44),PHS008
1),PHS009
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22),PHS010
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22),PHS011
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22),PHS012
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14),PHS013
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22),PHS014
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000),PHS015
DIMENSION ICHECK(44),PHS016
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG,PHS017
INTEGER PATH,PHS018
REAL IR,IRS,PHS019
WRITE(IPRNT,25),PHS020
25 FORMAT(1H1, 54H PHYSICAL CHARACTERISTICS OF THE STRUCTURAL CABLE ARPHS021
1RAY),PHS022
WRITE(IPRNT,1) NANC,PHS023
1 FORMAT(///,19H NO. OF ANCHORS IS ,I2,//5X,64H JUNCTION NO. X-COOPHS024
1RDINATE Y-COORDINATE Z-COORDINATE),PHS025
DO 2 N=1,NANC,PHS026
INDEX=ANJUNC(N),PHS027
2 WRITE(IPRNT,3) INDEX,(PJUNC(I,INDEX),I=1,3),PHS028
3 FORMAT(10X,I2,4X,3(5X,F10.2,3X)),PHS029
NOJUNC=NCAB+NANC-NIR,PHS030
WRITE(IPRNT,4) NOJUNC,PHS031
4 FORMAT(///,39H NO. OF JUNCTIONS IN ORIGINAL ARRAY IS ,I2),PHS032
WRITE(IPRNT,5) NIR,PHS033
5 FORMAT(///,39H NO. OF CUTS MADE IN ORIGINAL ARRAY IS ,I2//1H ,2(5X,PHS034
112H JUNCTION NO.),//8X,6H OF CUT,6X,17H AT WHICH CUT MADE),PHS035
IF(NIR.EQ.0) GO TO 3C,PHS036
DO 6 N=1,NIR,PHS037
6 WRITE(IPRNT,7) IRJUNC(N),ERJUNC(N),PHS038
7 FORMAT(10X,I2,15X,I2),PHS039
30 WRITE(IPRNT,8) NCAB,PHS040
8 FORMAT(///,18H NO. OF CABLES IS ,I2,//5X,16H CABLE S=0 S=L,43X,4H,PHS041
1DRAG,19X,22H CONSTITUTIVE NO. OF,//6X,3H NO.,1X,2(2X,4H JUNC),3X,PHS042
26H LENGTH,03X,8H DIAMETER,3X,13H WEIGHT/LENGTH,3X,11H COEFFICIENT,3X,PHS043
38H RIGIDITY,6X,8H EXPONENT,5X,8H ELEMENTS),PHS044
DO 9 N=1,NCAB,PHS045
NSEG=NNODE(N)-1,PHS046
RL=H(N)*NSEG,PHS047
9 WRITE(IPRNT,10) N,ZJUNC(N),LJUNC(N),RL,DCAB(N),WCAB(N),CDCAB(N),PHS048
1ECICAB(N),EXPCAB(N),NSEG,PHS049
10 FORMAT(6X,I2,5X,I2,4X,I2,1X,F9.1,3(3X,F7.3,4X), F10.0,7X,F6.3,PHS050
19X,I2),PHS051
WRITE(IPRNT,11),PHS052
11 FORMAT(///,63H PROPERTIES OF THE DEVICES LOCATED AT JUNCTIONS ARE APHS053
15 FOLLOWS ,//6X,6H DEVICE,10X,6H DEVICE,9X,11H DEVICE DRAG,4X,14H DEVIPHS054

```

```
2CE FRONTAL,/,5X,9HJUNC. NO.,8X,6HWEIGHT,9X,11HCOEFFICIENT,9X,  
34HAREA)  
12 READ(INTAPE,13) (DATA(I),I=1,10)  
13 FORMAT(F4.0,A4,8E15.8)  
   IF(DATA(2).EQ.TEST(3)) GO TO 14  
   IF(DATA(2).EQ.TEST(9)) GO TO 16  
   GO TO 12  
14 JUNC=DATA(3)  
   WRITE(IPRNT,15) JUNC,(DATA(K),K=6,8)  
15 FORMAT(8X,12,9X,F10.2,6X,F10.3,6X,F10.2)  
   GO TO 12  
16 REWIND INTAPE  
   WRITE(IPRNT,24) NDEV  
24 FORMAT(///,32H TOTAL NO. OF INDEXED DEVICES IS,14)  
   WRITE(IPRNT,17) IVOPT  
17 FORMAT(///,25H CURRENT FIELD OPTION IS ,11)  
   IV=IVOPT+1  
   GO TO (22,18,22),IV  
18 WRITE(IPRNT,19)  
19 FORMAT(//7X12HZ-COORDINATE,6X,11HVELOCITY OF,/,8X,10HOF CURRENT,7X,  
112HCURRENT AT Z)  
   DO 20 N=1,NVSEG  
20 WRITE(IPRNT,21) ZVEL(N),VELZ(N)  
21 FORMAT(2X,5X,F10.2,10X,F7.2)  
22 WRITE(IPRNT,23) COMPD  
23 FORMAT(///,38H ACCURACY REQUIRED IN CALCULATIONS IS ,F6.2)  
   IF(OFLG.NE.0) WRITE(IPRNT,111) NDATE  
111 FORMAT(///,30H DEVICE LOCATION OUTPUT RECORD,13,21H REFERS TO THIS  
   IARRAY )  
   RETURN  
   END
```

PHS060
PHS061
PHS062
PHS063
PHS064
PHS065
PHS066
PHS067
PHS068
PHS069
PHS070
PHS071
PHS072
PHS073
PHS074
PHS075
PHS076
PHS077
PHS078
PHS079
PHS080
PHS081
PHS082
PHS083
PHS084
PHS085
PHS086
PHS087
PHS088
PHS089
PHS090

```

SUBROUTINE STROUT
C
C THIS ROUTINE GENERATES THE ARRAY STRUCTURAL OUTPUT
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETA,COMP,THETAB,NJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
DIMENSION TEMP1(3),TEMP2(3),TEMP3(3),DISP(3),PSPACE(3)
DIMENSION A(3),B(3),D(3),U(3),V(3),W(3),C(4),RQ(3),RI(3)
NOJUNC=NCAB+NANC-NIR
JUM=JUMP+1
GO TO(100,200),JUM
100 WRITE(IPRNT,1002)
1002 FORMAT(1H,35HARRAY EQUILIBRIUM WITH NO CURRENT ///)
GO TO 300
200 WRITE(IPRNT,1001) THETA
1001 FORMAT(1H,35HARRAY EQUILIBRIUM WITH CURRENT FROM,F8.2, 9H DEGREES
1 ///)
C
C GENERATE ANCHOR HEADERS
C
300 WRITE(IPRNT,1003)
1003 FORMAT(1H,13HARRAY ANCHORS )
WRITE(IPRNT,10031)
10031 FORMAT(1H,13H-----//)
WRITE(IPRNT,1004)
1004 FORMAT(1H, 32HJUNC. NO. CABLE AT TENSION AT ,14X,26HFORCE C
10MPONENTS AT ANCHOR,13X,16HCABLE ANGLES WRT)
WRITE(IPRNT,1005)
1005 FORMAT(1H,30HOF ANCHOR ANCHOR ANCHOR8X,6HX-COMP,6X,6HY-COMSTR
1P,6X,6HZ-COMP,3X,9HHOR.-COMP,5X,16HX-AXIS XY-PLANE)
C
C THIS SECTION CALCULATES FORCES AND ANGLES AT ANCHORS
C
DO 401 J1=1,NANC
J3=0
402 J2=1
404 IF (ANJUNC(J1).EQ.ZJUNC(J2)) GO TO 403
414 J2=J2+1
IF(J2.LE.NCAB) GO TO 404
IF(NIR.EQ.0) GO TO 401
412 J3=1
411 IF(ANJUNC(J1).EQ.ERJUNC(J3)) GO TO 405

```

```

STR001
STR002
STR003
STR004
STR005
STR006
STR007
STR008
STR009
STR010
STR011
STR012
STR013
STR014
STR015
STR016
STR017
STR018
STR019
STR020
STR021
STR022
STR023
STR024
STR025
STR026
STR027
STR028
STR029
STR030
STR031
STR032
STR033
STR034
STR035
STR036
STR037
STR038
STR039
STR040
STR041
STR042
STR043
STR044
STR045
STR046
STR047
STR048
STR049
STR050
STR051
STR052
STR053
STR054
STR055
STR056
STR057
STR058
STR059

```

```

413 J3=J3+1
    IF(J3.LE.NIR) GO TO 411
    GO TO 401
405 INDEX=1RJUNC(J3)
    DO 406 J4=1,NCAB
    IF(INDLX.EQ.LJUNC(J4)) GO TO 407
406 CONTINUE
    GO TO 401
407 MM=J4
    MX=NNODE(J4)
    T=TCAB(MX,MM)
    RX=-RCAB(1,MX,MM)
    RY=-RCAB(2,MX,MM)
    RZ=-RCAB(3,MX,MM)
    GO TO 408
403 MM=J2
    T=TCAB(1,MM)
    RX=RCAB(1,1,MM)
    RY=RCAB(2,1,MM)
    RZ=RCAB(3,1,MM)
408 RH=SQRT(RX**2 + RY**2)
    A1=ASIN(RY/RH)/PI
    A2=ASIN(RZ/T)/PI
    IF((RX.LT.0.).AND.(RY.GE.0.)) A1=180.-A1
    IF((RX.LT.0.).AND.(RY.LT.0.)) A1=-180.-A1
    WRITE(IPRNT,1005) ANJUNC(J1),MM,T,RX,RY,RZ,RH,A1,A2
1006 FORMAT(1H,3X,12,9X,12,5X,F10.1,3X,4(F10.1,2X),2(2X,F7.2,1X))
    IF(J2.LT.NCAB) GO TO 414
    IF( (J2.EQ.NCAB).AND.(J3.EQ.0) ) GO TO 412
    IF( (J2.EQ.NCAB).AND.(J3.NE.NJUNC) ) GO TO 413
401 CONTINUE
C
C GENERATE CABLE HEADERS
C
    WRITE(IPRNT,2000)
2000 FORMAT(1H,///1H,12HARRAY CABLES )
    WRITE(IPRNT,2001)
2001 FORMAT(1H,12H-----/)
    WRITE(IPRNT,2002)
2002 FORMAT(1H,120HCABLE MAXIMUM S-COORD MINIMUM S-COORD MAXIMUM
1 S-COORD LOCATION OF THIS POINT NO CURRENT LOC. OF THIS POS
2INT )
    WRITE(IPRNT,2003)
2003 FORMAT(1H,120H NO. TENSION OF TENSION OF DISP.
1 OF X-COORD Y-COORD Z-COORD X-COORD Y-COORD Z-COORD
2RD )
C
C THIS SECTION CALCULATES MAXIMUM AND MINIMUM CABLE TENSIONS
C AND MAXIMUM CABLE DISPLACEMENTS FROM NO CURRENT LOCATION
C BY EXTRAPOLATION BETWEEN CABLE NODES
C
C INITIALIZE EXTREMA
C
    DO 799 NN=1,NCAB
    N=NN
    TMAX=TCAB(1,N)
    STMAX=0.
    TMIN=IMAX
    STMIN=0.

```

STR060
 STR061
 STR062
 STR063
 STR064
 STR065
 STR066
 STR067
 STR068
 STR069
 STR070
 STR071
 STR072
 STR073
 STR074
 STR075
 STR076
 STR077
 STR078
 STR079
 STR080
 STR081
 STR082
 STR083
 STR084
 STR085
 STR086
 STR087
 STR088
 STR089
 STR090
 STR091
 STR092
 STR093
 STR094
 STR095
 STR096
 STR097
 STR098
 STR099
 STR100
 STR101
 STR102
 STR103
 STR104
 STR105
 STR106
 STR107
 STR108
 STR109
 STR110
 STR111
 STR112
 STR113
 STR114
 STR115
 STR116
 STR117
 STR118

```

      GO TO (802,801),JUM
801  DMAX=0.
      DO 700 I=1,3
800  DMAX=DMAX+(PCAB(I,1,N)-PCABO(I,1,N))*2
      DMAX=SQRT(DMAX)
      SDMAX=0.
802  MX=NNODE(N)-1
      DO 750 MM=1,MX
      M=MM
      M1=MM+1
C
C  CALCULATE EXTRAPOLATION QUANTITIES
C
      RR=0.
      RD=0.
      DD=0.
      DO 701 I=1,3
      D(I)=(RCAB(I,M1,N)-RCAB(I,M,N))/H(N)
      GO TO (804,803),JUM
803  A(I)=EXCAB(M,N)*RCAB(I,M,N)/TCAB(M,N)
      B(I)=(EXCAB(M1,N)*RCAB(I,M1,N)/TCAB(M1,N)-A(I))/H(N)
      U(I)=PCAB(I,M,N)-PCABO(I,M,N)
804  RR=RR+RCAB(I,M,N)*RCAB(I,M,N)
      RD=RD+RCAB(I,M,N)*D(I)
801  DD=DD+D(I)*D(I)
      GO TO (806,805),JUM
805  DO 702 I=1,3
      TEMP1(I)=RCAB(I,M,N)
      TEMP2(I)=RCAB(I,M1,N)
      RCAB(I,M,N)=RCABO(I,M,N)
802  RCAB(I,M1,N)=RCABO(I,M1,N)
      UU=0.
      UV=0.
      UVW=0.
      VW=0.
      WW=0.
      DO 703 I=1,3
      V(I)=EXCAB(M,N)*RCAB(I,M,N)/TCAB(M,N)
      W(I)=(EXCAB(M1,N)*RCAB(I,M1,N)/TCAB(M1,N)-V(I))/H(N)
      VI(I)=A(I)-V(I)
      WI(I)=(B(I)-W(I))/2.
      UU=UU+U(I)*U(I)
      UV=UV+U(I)*V(I)
      UVW=UVW+2.*U(I)*W(I)+V(I)*V(I)
      VW=VW+V(I)*W(I)
803  WW=WW+W(I)*W(I)
      DO 704 I=1,3
      RCAB(I,M,N)=TEMP1(I)
804  RCAB(I,M1,N)=TEMP2(I)
806  C(4)=0.
      C(3)=0.
      C(2)=DD
      C(1)=RD
      CALL KPCOLY(C,RD,RI)
C
C  CALCULATE TENSION EXTREMA IN SEGMENT
C
      JTIME=0
      DO 719 I=1,3

```

```

STR119
STR120
STR121
STR122
STR123
STR124
STR125
STR126
STR127
STR128
STR129
STR130
STR131
STR132
STR133
STR134
STR135
STR136
STR137
STR138
STR139
STR140
STR141
STR142
STR143
STR144
STR145
STR146
STR147
STR148
STR149
STR150
STR151
STR152
STR153
STR154
STR155
STR156
STR157
STR158
STR159
STR160
STR161
STR162
STR163
STR164
STR165
STR166
STR167
STR168
STR169
STR170
STR171
STR172
STR173
STR174
STR175
STR176
STR177

```

```

      IF( RI(I).NE.0.) GO TO 719
      IF((RQ(I).LE.0.)OR.(RQ(I).GE.H(N))) GO TO 719
      SIG=RQ(I)
710  TE=SQRT(RR+2.*RD*SIG+DD*SIG**2)
      IF(TE.GT.TMAX) GO TO 712
      IF(TE.LT.TMIN) GO TO 713
711  IF(JTIME.EQ.0) GO TO 719
      GO TO 715
712  TMAX=TE
      STMAX=H(N)*(M-1)+SIG
      GO TO 711
713  TMIN=TE
      STMIN=H(N)*(M-1)+SIG
      GO TO 711
719  CONTINUE
      JTIME=1
      SIG=H(N)
      GO TO 710
715  GO TO(750,716),JUM
716  C(4)=2.*WW
      C(3)=3.*VW
      C(2)=UVW
      C(1)=1
      CALL RPOLY(C,RQ,RI)
C
C CALCULATE MAXIMUM DISPLACEMENT IN SEGMENT
C
      JTIME=0
      DO 729 I=1,3
      IF (RI(I).NE.0.) GO TO 729
      IF((RQ(I).LE.0.)OR.(RQ(I).GE.H(N))) GO TO 729
      SIG=RQ(I)
720  DE=SQRT(UU+2.*UV*SIG+UVW*SIG**2+2.*VW*SIG**3+WW*SIG**4)
      IF(DE.GT.DMAX) GO TO 722
721  IF(JTIME.EQ.0) GO TO 729
      GO TO 750
722  DMAX=DE
      SDMAX= H(N)*(M-1)+SIG
      GO TO 721
729  CONTINUE
      JTIME=1
      SIG=H(N)
      GO TO 720
750  CONTINUE
C
C EXTREMA ALONG A CABLE NOW DETERMINED
C CALCULATE FINAL AND INITIAL COORDINATES OF MAXIMALLY DISPLACED POINT
C
      GO TO (810,809),JUM
809  RL=H(N)*MX
      IF(SDMAX.LT.RL) GO TO 760
      K=LJUNC(N)
      DO 751 I=1,3
      A(I)=PJUNC(I,K)
751  B(I)=PJUNC(I,K)
      GO TO 780
760  DATA(2)=TEST(4)
      DATA(3)=N
      DATA(10)=SDMAX

```

```

STR178
STR179
STR180
STR181
STR182
STR183
STR184
STR185
STR186
STR187
STR188
STR189
STR190
STR191
STR192
STR193
STR194
STR195
STR196
STR197
STR198
STR199
STR200
STR201
STR202
STR203
STR204
STR205
STR206
STR207
STR208
STR209
STR210
STR211
STR212
STR213
STR214
STR215
STR216
STR217
STR218
STR219
STR220
STR221
STR222
STR223
STR224
STR225
STR226
STR227
STR228
STR229
STR230
STR231
STR232
STR233
STR234
STR235
STR236

```

```

J=(SDMAX/H(N))+1
J1=J+1
DO 761 I=1,3
  I=I1
761 A(I)=SPACE(I)
  DO 762 I=1,3
    TEMP1(I)=RCAB(I,J,N)
    TEMP2(I)=RCAB(I,J1,N)
    TEMP3(I)=PCAB(I,J,N)
    RCAB(I,J,N)=RCABO(I,J,N)
    RCAB(I,J1,N)=RCABO(I,J1,N)
762 PCAB(I,J,N)=PCABO(I,J,N)
  DO 763 I=1,3
    I=I1
763 B(I)=SPACE(I)
  DO 764 I=1,3
    RCAB(I,J,N)=TEMP1(I)
    RCAB(I,J1,N)=TEMP2(I)
764 PCAB(I,J,N)=TEMP3(I)
780 CONTINUE
  WRITE(IPRNT,781)N,TMAX,STMAX,TMIN,STMIN,DMAX,SDMAX,(A(I),I=1,3),
  1 B(I),I=1,3)
  GO TO 799
810 WRITE(IPRNT,781)N,TMAX,STMAX,TMIN,STMIN
781 FORMAT(1H ,I4,1X,6F9.1,6F10.1)
799 CONTINUE

C
C GENERATE JUNCTION HEADERS
C
  WRITE(IPRNT,1007)
1007 FORMAT(1H ///1H ,15HARRAY JUNCTIONS)
  WRITE(IPRNT,10071)
10071 FORMAT(1H ,15H----- / )
  WRITE(IPRNT,1008)
1008 FORMAT(1H ,27HJUNC. CABLE AT TENSION AT,3X,16HCABLE ANGLES WRT,
110X,17HJUNCTION LOCATION,10X,33HDISPLACEMENT FROM NO CURRENT LOC.)
  WRITE(IPRNT,1009)
1009 FORMAT(1H ,15H NO. JUNCTION3X,8HJUNCTION4X,16HX-AXIS XY-PLANE,
168H X-COORD Y-COORD Z-COORD X-DISP Y-DISP Z-DISP )
  2DISP )

C
C THIS SECTION CALCULATES JUNCTION FORCES, LOCATIONS AND DISPLACEMENTS
C
  IF=0
  DO 601 J1=1,NOJUNC
  DO 605 K=1,NANC
  IF(ANJUNC(K).EQ.J1) GO TO 601
605 CONTINUE
  GO TO (602,603),JUM
603 DO 604 J2=1,3
604 DISP(J2)=PJUNC(J2,J1)-PJUNC0(J2,J1)
602 J6=1
608 IF(J1.EQ.2JUNC(J6)) GO TO 606
615 IF(J1.EQ.LJUNC(J6)) GO TO 607
617 J6=J6+1
  IF(J6.LE.NCAB) GO TO 608
  IF(NIR.LO.0) GO TO 601
  J7=1
609 IF(J1.EQ.ERJUNC(J7)) GO TO 610

```

```

STR237
STR238
STR239
STR240
STR241
STR242
STR243
STR244
STR245
STR246
STR247
STR248
STR249
STR250
STR251
STR252
STR253
STR254
STR255
STR256
STR257
STR258
STR259
STR260
STR261
STR262
STR263
STR264
STR265
STR266
STR267
STR268
STR269
STR270
STR271
STR272
STR273
STR274
STR275
STR276
STR277
STR278
STR279
STR280
STR281
STR282
STR283
STR284
STR285
STR286
STR287
STR288
STR289
STR290
STR291
STR292
STR293
STR294
STR295

```

```

616 J7=J7+1
    IF (J7.LE.NIR) GO TO 609
    GO TO 601
606 MM=J6
    T=TCAB(1,MM)
    RX=RCAB(1,1,MM)
    RY=RCAB(2,1,MM)
    RZ=RCAB(3,1,MM)
    GO TO 611
610 INDEX=IRJUNC(J7)
    DO 612 J8=1,NCAB
    IF (INDEX.EQ.LJUNC(J8)) GO TO 613
612 CONTINUE
613 MM=J8
    GO TO 614
607 MM=J6
614 MX=NNODE(MM)
    T=TCAB(MX,MM)
    RX=-RCAB(1,MX,MM)
    RY=-RCAB(2,MX,MM)
    RZ=-RCAB(3,MX,MM)
611 RH=SQRT(RX**2+RY**2) IF (RH.EQ.0) RH=1.
    A1=ASIN(RY/RH)/PI
    A2=ASIN(RZ/T)/PI
    IF ((RX.LT.0.).AND.(RY.GE.0.)) A1=180.-A1
    IF ((RX.LT.0.).AND.(RY.LT.0.)) A1=-180.-A1
    IF ((IF.EQ.0).AND.(JUM.EQ.1)) WRITE(IPRNT,620) J1,MM,T,A1,A2,(PJUNC
1(K,J1),K=1,3)
    IF ((IF.LQ.0).AND.(JUM.LQ.2)) WRITE(IPRNT,620) J1,MM,T,A1,A2,(PJUNC
1(K,J1),K=1,3), (DISP(I),I=1,3)
    IF (IF.EQ.1) WRITE(IPRNT,620) J1,MM,T,A1,A2
    IF=1
620 FORMAT(1H ,I4,6X,I2,4X,F10.1,3X,F7.2,3X,F7.2,3(1X,F10.1),1X,3(1X,
1F10.1))
    IF ((J6.LT.NCAB).AND.(J1.EQ.ZJUNC(J6))) GO TO 615
    IF ((J6.LT.NCAB).AND.(J1.EQ.LJUNC(J6))) GO TO 617
601 IF=0
C
C GENERATE INDEXED DEVICE HEADERS
C
    WRITE(IPRNT,519)
519 FORMAT(1H ,///1H ,34HINDEXED DEVICES ALONG ARRAY CABLES )
    WRITE(IPRNT,5191)
5191 FORMAT(1H ,34H----- - - - - - * /)
    WRITE(IPRNT,520)
520 FORMAT(1H ,7HDEVICE ,5HCABLE,6X,1HS,7X,10HTENSION AT,13X,15HDEVICE
1 LOCATION,12X,36H DISPLACMENT FROM NO CURRENT LOC. )
    WRITE(IPRNT,521)
521 FORMAT(1H ,109HINDEX NO. COORDINATE DEVICE X-COORD
1 Y-COORD Z-COORD X-DISP Y-DISP Z-DISP)
C
C THIS SECTION CALCULATES TENSIONS AT INDEXED DEVICES,
C DEVICE LOCATIONS, AND DEVICE DISPLACEMENTS
C
    IF (NDEV.EQ.0) GO TO 511
    DO 510 N=1,NDEV
    READ (INTAPE,501) (DATA(I),I=1,10)
501 FORMAT (14,A4,8E15.8)
    IF (DATA(2).LQ.TEST(3)) GO TO 510

```

STR296
 STR297
 STR298
 STR299
 STR300
 STR301
 STR302
 STR303
 STR304
 STR305
 STR306
 STR307
 STR308
 STR309
 STR310
 STR311
 STR312
 STR313
 STR314
 STR315
 STR316
 STR317
 STR318
 STR319
 STR320
 STR321
 STR322
 STR323
 STR324
 STR325
 STR326
 STR327
 STR328
 STR329
 STR330
 STR331
 STR332
 STR333
 STR334
 STR335
 STR336
 STR337
 STR338
 STR339
 STR340
 STR341
 STR342
 STR343
 STR344
 STR345
 STR346
 STR347
 STR348
 STR349
 STR350
 STR351
 STR352
 STR353
 STR354

```
K=DATA(3)
M=DATA(10)/H(K) + 1
INDEX=DATA(5)
SIGMA=DATA(10)-(M-1)*H(K)
TEN2=0.
DO 502 I=1,3
  I=11
  PSPACE(I)=SPACE(I)
502 TEN2=TEN2+(RCAB(I,M,K)+(RCAB(I,M+1,K)-RCAB(I,M,K))*SIGMA/H(K))**2
  TEN=SQRT(TEN2)
  GO TO (503,505),JUM
503 WRITE (IPRNT,509) INDEX,K,DATA(10),TEN,(PSPACE(I),I=1,3)
  GO TO 510
505 DO 506 I=1,3
  TEMP1(I)=RCAB(I,M,K)
  TEMP2(I)=RCAB(I,M+1,K)
  TEMP3(I)=PCAB(I,M,K)
  RCAB(I,M,K)=RCABO(I,M,K)
  RCAB(I,M+1,K)=RCABO(I,M+1,K)
506 PCAB(I,M,K)=PCABO(I,M,K)
  DO 507 I=1,3
  I=11
507 DISP(I)=PSPACE(I)-SPACE(I)
  DO 508 I=1,3
  RCAB(I,M,K)=TEMP1(I)
  RCAB(I,M+1,K)=TEMP2(I)
508 PCAB(I,M,K)=TEMP3(I)
  WRITE (IPRNT,509) INDEX,K,DATA(10),TEN,(PSPACE(I),I=1,3),
  1(DISP(I),I=1,3)
509 FORMAT(1H ,I4,4X,I2,4X,F9.1,3X,F9.1,1X,3(2X,F10.1),4X,3(1X,F10.1))
510 CONTINUE
511 REWIND INTAPE
RETURN
END
```

```
STR355
STR356
STR357
STR358
STR359
STR360
STR361
STR362
STR363
STR364
STR365
STR366
STR367
STR368
STR369
STR370
STR371
STR372
STR373
STR374
STR375
STR376
STR377
STR378
STR379
STR380
STR381
STR382
STR383
STR384
STR385
STR386
STR387
STR388
```

SUBROUTINE TAPOUT

```

C
C THIS ROUTINE GENERATES THE TAPE OR CARDS GIVING
C THE LOCATIONS OF THE INDEXED DEVICES
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,N1-,THETAS,THETA,COMP,THETAB,ANJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
DIMENSION PSPACE(3)
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
ID1=4H CUR
ID2=4H DEV
ID3=4H REC
1 FORMAT(A4,I4,3F10.2)
JUM=JUMP+1
GO TO (2,3) JUM
2 WRITE(OUTAPE,1) ID3,NDATC
WRITE(OUTAPE,1) ID1,JUMP
GO TO 4
3 WRITE(OUTAPE,1) ID1,JUMP,THETA
4 IF (NDEV.EQ.0) GO TO 8
DO 7 NN=1,NDEV
READ(INTAPE,5) (DATA(K),K=1,10)
5 FORMAT(F4.0,A4,8E15.8)
INDEX=DATA(5)
C
C CALCULATE LOCATION OF DEVICE IN SPACE
C
DO 6 J=1,3
I=J
6 PSPACE(I)=SPACE(I)
WRITE(OUTAPE,1) ID2,INDEX,(PSPACE(I),I=1,3)
7 CONTINUE
8 CONTINUE
REWIND +NTAPE
RETURN
END

```

TAP001
TAP002
TAP003
TAP004
TAP005
TAP006
TAP007
TAP008
TAP009
TAP010
TAP011
TAP012
TAP013
TAP014
TAP015
TAP016
TAP017
TAP018
TAP019
TAP020
TAP021
TAP022
TAP023
TAP024
TAP025
TAP026
TAP027
TAP028
TAP029
TAP030
TAP031
TAP032
TAP033
TAP034
TAP035
TAP036
TAP037
TAP038
TAP039
TAP040
TAP041
TAP042
TAP043
TAP044
TAP045
TAP046
TAP047
TAP048
TAP049
TAP050
TAP051
TAP052

SUBROUTINE ERROR

```

C
C THIS ROUTINE GENERATES A TYPE 18 ERROR MESSAGE
C GET HERE IF ALL IMAGINARY REACTIONS DO NOT CHANGE
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,ERR001
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNC,ERR002
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,ERR003
1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,ERR004
2OFLG,NIR,THETAS,THETA,COMP,THETA,ANJUNC,RHO,TEST,ERR005
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,ERR006
4WCAB,IDEV,ICHECK,NDEV,NDATC,ERR007
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNC(3,44),ERR008
1)ERR009
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22),ERR010
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22),ERR011
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22),ERR012
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14),ERR013
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22),ERR014
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000),ERR015
DIMENSION ICHECK(44),ERR016
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG,ERR017
INTEGER PATH,ERR018
REAL IR,IRS,ERR019
E=E*10.,ERR020
WRITE(IPRNT,1),ERR021
1 FORMAT(1H1,52H TYPE 18 ERROR. STRUCTURAL ANALYSIS NOT COMPLETED,ERR022
1/60H PRINTOUT GIVEN FOR DIAGNOSTIC PURPOSES. SEE USERS MANUAL./)ERR023
JUM=JUMP+1,ERR024
GO TO (4,6),JUM,ERR025
4 WRITE(IPRNT,5),ERR026
5 FORMAT(5X,40HEXISTING CURRENT CONDITION IS NO CURRENT /),ERR027
GO TO 6,ERR028
6 WRITE(IPRNT,7) THETA,ERR029
7 FORMAT(5X,38HEXISTING CURRENT CONDITION IS THETA = ,F3.0/),ERR030
8 WRITE(IPRNT,9) E,ERR031
9 FORMAT(5X,35HBEST VALUE OF ACCURACY OBTAINED IS ,F6.2),ERR032
DO 2 N=1,NCAB,ERR033
NN=N,ERR034
WRITE(IPRNT,10) NN,ERR035
10 FORMAT(//5X,29HTHE TENSIONS IN CABLE NUMBER ,I2,4H ARE,/),ERR036
L=NNODE(N),ERR037
DO 2 M=1,L,ERR038
MM=M,ERR039
T=TCAB(MM,NN),ERR040
S=H(NN)*(MM-1.),ERR041
2 WRITE(IPRNT,3) T,S,ERR042
3 FORMAT(7X,2HT=,F10.2,6H AT S=,F10.2),ERR043
RETURN,ERR044
END,ERR045
ERR046
ERR047
ERR048
ERR049
ERR050

```

SUBROUTINE RPOLY(C,RR,RI)

```

C
C THIS ROUTINE FINDS REAL ROOTS OF POLYNOMIAL EQUATIONS UP TO
C  $C(4) \cdot (X^3) + C(3) \cdot (X^2) + C(2) \cdot X + C(1) = 0$  FOR USE IN
C EVALUATING MAXIMUM CABLE DISPLACEMENTS AND TENSION EXTREMA
C
C REAL PARTS OF ROOTS ARE PLACED IN RR, IMAGINARY PARTS IN RI
C SINCE ONLY REAL ROOTS ARE OF INTEREST, ALL NON-REAL OR
C NON-EXISTING ROOTS RETURN RR=0, RI=1
C
      DIMENSION C(4),RR(3),RI(3)
      IF(C(4).NE.0.) GO TO 30
      IF(C(3).NE.0.) GO TO 20
      IF(C(2).NE.0.) GO TO 10
C
C GET HERE IF EQUATION IDENTICALLY SATISFIED
C
      100 DO 1 I=1,3
          RR(I)=0.
          1 RI(I)=1.
          RETURN
C
C GET HERE IF EQUATION LINEAR
C
      10 RR(1)=-C(1)/C(2)
      RI(1)=0.
      200 DO 11 I=2,3
          RR(I)=0.
          11 RI(I)=1.
          RETURN
C
C GET HERE IF EQUATION QUADRATIC
C
      20 DISC=C(2)**2-4.*C(1)*C(3)
      IF(DISC.GE.0.) GO TO 21
      GO TO 100
      21 RR(1)=(-C(2)+SQRT(DISC))/(2.*C(3))
      RI(1)=0.
      RR(2)=(-C(2)-SQRT(DISC))/(2.*C(3))
      RI(2)=0.
      RR(3)=0.
      RI(3)=1.
      RETURN
C
C GET HERE IF EQUATION CUBIC
C
      30 P=C(3)/C(4)
      Q=C(2)/C(4)
      R=C(1)/C(4)
      A=(3.*Q-P**2)/3.
      B=(2.*P**3-9.*P*Q+27.*R)/27.
      DISC=(B**2)/4.+(A**3)/27.
      IF(DISC)32,31,31
      31 DISC= SQRT(DISC)
      CAPA=(((-B/2.+DISC)**4)**(1./3.))/(-B/2.+DISC)
      CAPB=(((-B/2.-DISC)**4)**(1./3.))/(-B/2.-DISC)
      RR(1)=CAPA+CAPB-P/3.
      RI(1)=0.
      IF(DISC.GT.0.) GO TO 200

```

```

RP0001
RP0002
RP0003
RP0004
RP0005
RP0006
RP0007
RP0008
RP0009
RP0010
RP0011
RP0012
RP0013
RP0014
RP0015
RP0016
RP0017
RP0018
RP0019
RP0020
RP0021
RP0022
RP0023
RP0024
RP0025
RP0026
RP0027
RP0028
RP0029
RP0030
RP0031
RP0032
RP0033
RP0034
RP0035
RP0036
RP0037
RP0038
RP0039
RP0040
RP0041
RP0042
RP0043
RP0044
RP0045
RP0046
RP0047
RP0048
RP0049
RP0050
RP0051
RP0052
RP0053
RP0054
RP0055
RP0056
RP0057
RP0058
RP0059

```

```

RR(2)=-CAPA-P/3.
RI(2)=0.
RR(3)=RK(2)
RI(3)=0.
RETURN
32 DISC=2.*SQRT(-A/3.)
PHI=ACOS((+3.*B)/(A*DISC))/3.
RR(1)=-P/3.+DISC*COS(PHI)
RR(2)=-P/3.+DISC*COS(PHI+3.14159265*(2./3.))
RR(3)=-P/3.+DISC*COS(PHI+3.14159265*(4./3.))
RI(1)=0.
RI(2)=0.
RI(3)=0.
RETURN
END

```

RPO060
RPO061
RPO062
RPO063
RPO064
RPO065
RPO066
RPO067
RPO068
RPO069
RPO070
RPO071
RPO072
RPO073
RPO074

```

C SUBROUTINE SWITCH
C THIS ROUTINE SWITCHES INPUT DATA
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAE,FATE,NANG,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETA,COMP,THETAB,NJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,LCICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAE(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
DO 1 I=1,10
1 DATA(I)=DATN(I)
RETURN
END

```

SWT001
SWT002
SWT003
SWT004
SWT005
SWT006
SWT007
SWT008
SWT009
SWT010
SWT011
SWT012
SWT013
SWT014
SWT015
SWT016
SWT017
SWT018
SWT019
SWT020
SWT021
SWT022
SWT023
SWT024
SWT025
SWT026
SWT027

SUBROUTINE START

```

C THIS ROUTINE CALCULATES THE INITIAL GUESSES AT THE IMAGINARY
C AND EQUILIBRATING REACTIONS AND THE INITIAL DELTA BASED ON
C THE TOTAL WEIGHT OF THE ARRAY
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETAE,COMPD,THETAB,NJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,LXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
WEIGHT=0.
DO 1 J=1,NJUNC
1 WEIGHT=WEIGHT+FEJUNC(3,J)
DO 2 N=1,NCAB
INNN=NNODE(N)-1
DO 2 M=1,INNN
2 WEIGHT=WEIGHT+ FCAB(3,M,N)
DO 3 N=1,NIR
KER=ERJUNC(N)
DO 3 I=1,3
3 IR(1,KER)=0.
DO 4 N=1,NIR
KIR=IRJUNC(N)
KER=ERJUNC(N)
IR(1,KIR)=0.
IR(2,KIR)=0.
IR(3,KIR)=-WEIGHT/(NIR+1.0)
4 IR(3,KER)=IR(3,KER)-IR(3,KIR)
DELTA1= ABS(WEIGHT)/(NIR+1)
RETURN
END

```

STA001
 STA002
 STA003
 STA004
 STA005
 STA006
 STA007
 STA008
 STA009
 STA010
 STA011
 STA012
 STA013
 STA014
 STA015
 STA016
 STA017
 STA018
 STA019
 STA020
 STA021
 STA022
 STA023
 STA024
 STA025
 STA026
 STA027
 STA028
 STA029
 STA030
 STA031
 STA032
 STA033
 STA034
 STA035
 STA036
 STA037
 STA038
 STA039
 STA040
 STA041
 STA042
 STA043
 STA044
 STA045
 STA046

```

FUNCTION TCAB(M,K)
C
C THIS ROUTINE CALCULATES THE TENSION AT NODE M OF CABLE K
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANG,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETA,COMPU,THETAB,NJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
TCAB =SQRT(RCAB(1,H,K)**2 + RCAB(2,M,K)**2 + RCAB(3,M,K)**2 )
RETURN
END

```

TCA001
TCA002
TCA003
TCA004
TCA005
TCA006
TCA007
TCA008
TCA009
TCA010
TCA011
TCA012
TCA013
TCA014
TCA015
TCA016
TCA017
TCA018
TCA019
TCA020
TCA021
TCA022
TCA023
TCA024
TCA025
TCA026

```

FUNCTION EXCAB(M,K)
C
C THIS ROUTINE CALCULATES (1 + STRAIN) AT NODE M OF CABLE K
C
COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANG,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETA,COMPU,THETAB,NJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
IF(EXPCAB(K).EQ.0.) GO TO 1
EXCAB=1.+(TCAB(M,K)/LCICAB(K))*EXPCAB(K)
RETURN
1 EXCAB=1.
RETURN
END

```

EXC001
EXC002
EXC003
EXC004
EXC005
EXC006
EXC007
EXC008
EXC009
EXC010
EXC011
EXC012
EXC013
EXC014
EXC015
EXC016
EXC017
EXC018
EXC019
EXC020
EXC021
EXC022
EXC023
EXC024
EXC025
EXC026
EXC027
EXC028
EXC029

```

FUNCTION EFORCE(I)
C THIS ROUTINE CALCULATES THE DEVICE FORCES IN DIRECTION I
C USING THE NORMAL DRAG APPROXIMATION FOR IN-LINE DEVICES
COMMON /B1/ FEJUNC,IR,DELTAI,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCA60,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETA,COMP,THETAB,NJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVCPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCA60(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),LCICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
DIMENSION WTEL(3),VNORM(3),PSPACE(3)
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
C CALCULATE THE WEIGHT VECTOR OF A DEVICE
C
WTEL(1)=0.
WTEL(2)=0.
WTEL(3)=DATA(6)
C CHECK TO SEE IF CURRENT OR NO CURRENT
C
JUM=JUMP+1
GO TO(1,2),JUM
C GET HERE IF NO CURRENT
C
1 EFORCE = WTEL(1)
RETURN
C GET HERE IF CURRENT
C CALCULATE LOCATION OF DEVICE IN SPACE
C
2 DO 3 KK=1,3
K=KK
3 PSPACE(K)=SPACE(K)
C CHECK IF DEVICE IS IN-LINE OR FREE
C
IJMP=DATA(4)
GO TO (5,4,5,4),IJMP
C GET HERE IF FREE TYPE DEVICE -- CALCULATE MAGNITUDE OF THE CURRENT
C
4 VMAG=SQRT(VELOC(1,PSPACE)**2 + VELOC(2,PSPACE)**2 + VELOC(3,PSPACE)
1)**2 )

```

```

EF0001
EF0002
EF0003
EF0004
EF0005
EF0006
EF0007
EF0008
EF0009
EF0010
EF0011
EF0012
EF0013
EF0014
EF0015
EF0016
EF0017
EF0018
EF0019
EF0020
EF0021
EF0022
EF0023
EF0024
EF0025
EF0026
EF0027
EF0028
EF0029
EF0030
EF0031
EF0032
EF0033
EF0034
EF0035
EF0036
EF0037
EF0038
EF0039
EF0040
EF0041
EF0042
EF0043
EF0044
EF0045
EF0046
EF0047
EF0048
EF0049
EF0050
EF0051
EF0052
EF0053
EF0054
EF0055
EF0056
EF0057
EF0058
EF0059

```

```
C CALCULATE THE FORCE ON THE FREE DEVICE
C
  EFORCE = WTEL(I)+(RHO/2.)*DATA(7)*DATA(8)*VMAG*VELOC(I,PSPACE)
  RETURN
C
C GET HERE IF IN-LINE DEVICE
C CALCULATE THE TANGENTIAL PROJECTION OF THE CURRENT ON THE DEVICE
C TANG(I) EVALUATES THE UNIT TANGENT TO A CABLE AT ANY POINT
C
  5 VPROJ=0.
  DO 6 KK=1,3
    K=KK
  6 VPROJ=VPROJ + VELOC(K,PSPACE)*TANG(K)
C
C CALCULATE THE NORMAL COMPONENT OF THE CURRENT AND ITS MAGNITUDE
C
  DO 7 KK=1,3
    K=KK
  7 VNORM(K)=VELOC(K,PSPACE)-VPROJ*TANG(K)
    VNORM=SQRT(VNORM(1)**2 + VNORM(2)**2 + VNORM(3)**2)
C
C CALCULATE THE FORCE ON THE IN-LINE DEVICE
C
  EFORCE=WTEL(I)+(RHO/2.)*DATA(7)*((DATA(8)/12.)*DATA(9)*VNORM
1 *VNORM(I)
  RETURN
  END
```

```
EF0060
EF0061
EF0062
EF0063
EF0064
EF0065
EF0066
EF0067
EF0068
EF0069
EF0070
EF0071
EF0072
EF0073
EF0074
EF0075
EF0076
EF0077
EF0078
EF0079
EF0080
EF0081
EF0082
EF0083
EF0084
EF0085
EF0086
```

```

      FUNCTION CFORCE (I,M,N)
C
C THIS ROUTINE CALCULATES THE FORCE/LENGTH IN DIRECTION I AT NODE M
C ON CABLE N USING THE NORMAL DRAG FORCE APPROXIMATION
C
      COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
      1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
      COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
      1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
      2OFLG,NIK,THETAS,THETAE,COMPJ,THETAB,NJUNC,RHO,TEST,
      3NVSEG,ZVEL,ZELZ(25),ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
      4WCAB,IDEV,ICHECK,NDEV,NDATC
      DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
      1)
      DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
      DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
      DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
      DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
      DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
      DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
      DIMENSION ICHECK(44)
      DIMENSION WTCAB(3),VNORM(3),PSPACE(3)
      INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,CFLG
      INTEGER PATH
      REAL IR,IRS
C
C CALCULATE THE WEIGHT/LENGTH VECTOR
C
      WTCAB(1)=0.
      WTCAB(2)=0.
      WTCAB(3)=WCAB(N)
C
C CHECK TO SEE IF CURRENT OR NO CURRENT
C
      JUM=JUMP+1
      GO TO (1,2),JUM
C
C GET HERE IF NO CURRENT
C
      1 CFORCE=WTCAB(1)
      RETURN
C
C GET HERE IF CURRENT
C CALCULATE LOCATION OF NODE IN SPACE
C
      2 DO 3 K=1,3
      3 PSPACE(K)=PCAB(K,M,N)
C
C CALCULATE THE TANGENTIAL PROJECTION OF THE CURRENT ON THE CABLE
C
      VPROJ=0.
      DO 4 KK=1,3
      K=KK
      4 VPROJ=VPROJ + VELOC(K,PSPACE) * RCAB(K,M,N) / TCAB(M,N)
C
C CALCULATE THE NORMAL COMPONENT OF THE CURRENT AND ITS MAGNITUDE
C
      DO 5 KK=1,3
      K=KK

```

CF0001
CF0002
CF0003
CF0004
CF0005
CF0006
CF0007
CF0008
CF0009
CF0010
CF0011
CF0012
CF0013
CF0014
CF0015
CF0016
CF0017
CF0018
CF0019
CF0020
CF0021
CF0022
CF0023
CF0024
CF0025
CF0026
CF0027
CF0028
CF0029
CF0030
CF0031
CF0032
CF0033
CF0034
CF0035
CF0036
CF0037
CF0038
CF0039
CF0040
CF0041
CF0042
CF0043
CF0044
CF0045
CF0046
CF0047
CF0048
CF0049
CF0050
CF0051
CF0052
CF0053
CF0054
CF0055
CF0056
CF0057
CF0058
CF0059

```

5 VNORM(K)= VELOC(K,PSPACE) - VPROJ * RCAB(K,M,N) / TCAB(M,N)
  VNORM = SQRT(VNORM(1)**2 + VNORM(2)**2 + VNORM(3)**2 )
C
C CALCULATE THE FORCE/LENGTH
C
  CFORCE=WTCAB(I)+(RHO/2.)*CDCAB(I)*(DCAB(N)/12.)*EXCAB(M,N)*VNORM
1 *VNORM(I)
  RETURN
END

```

CF0060
CF0061
CF0062
CF0063
CF0064
CF0065
CF0066
CF0067
CF0068

```

FUNCTION SPACE(I)
C
C THIS ROUTINE CALCULATES THE LOCATION IN SPACE
C OF ANY POINT ON THE ARRAY
C
  COMMON /B1/ FEJUNC,IR,DELTA1,DELTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
  COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANC,ANJUNC,IRLAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETA,COMP,THETAB,ANJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
  DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
  DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
  DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
  DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
  DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
  DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
  DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
  DIMENSION ICHECK(44)
  INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
  INTEGER PATH
  REAL IR,IRS
  IF(DATA(2).EQ.TLST(3)) GO TO 1
  IF(DATA(2).EQ.TLST(4)) GO TO 2
C
C GET HERE IF JUNCTION POINT
C
  1 K=DATA(3)
  SPACE = PJUNC(I,K)
  RETURN
C
C GET HERE IF POINT ON A CABLE
C
  2 N=DATA(3)
  M= (DATA(10)/H(N)) +1
C
C CALCULATE DISTANCE, SIGMA, OF POINT FROM NODE M
C
  SIGMA = DATA(10) - (M-1)*H(N)
C
C CALCULATE EXTRAPOLATION QUANTITIES
C
  EM=EXCAB(M,N)*RCAB(I,M,N)/TCAB(M,N)
  EM1=EXCAB(M+1,N)*RCAB(I,M+1,N)/TCAB(M+1,N)
C
C CALCULATE LOCATION
C
  SPACE = PCAB(I,M,N)+EM*SIGMA+((EM1-EM)/H(N))*(SIGMA**2)/2.
  RETURN
END

```

SPA001
SPA002
SPA003
SPA004
SPA005
SPA006
SPA007
SPA008
SPA009
SPA010
SPA011
SPA012
SPA013
SPA014
SPA015
SPA016
SPA017
SPA018
SPA019
SPA020
SPA021
SPA022
SPA023
SPA024
SPA025
SPA026
SPA027
SPA028
SPA029
SPA030
SPA031
SPA032
SPA033
SPA034
SPA035
SPA036
SPA037
SPA038
SPA039
SPA040
SPA041
SPA042
SPA043
SPA044
SPA045
SPA046
SPA047
SPA048
SPA049
SPA050
SPA051
SPA052

FUNCTION VELOC(I, PSPACE)

VEL001

C THIS ROUTINE SPECIFIES THE I COMPONENT OF THE CURRENT FIELD
C AT AN ARBITRARY POINT IN SPACE, PSPACE(I)

VEL002

VEL003

VEL004

VEL005

VEL006

VEL007

VEL008

VEL009

VEL010

VEL011

VEL012

VEL013

VEL014

VEL015

VEL016

VEL017

VEL018

VEL019

VEL020

VEL021

VEL022

VEL023

VEL024

VEL025

VEL026

VEL027

VEL028

VEL029

VEL030

VEL031

VEL032

VEL033

VEL034

VEL035

VEL036

VEL037

VEL038

VEL039

VEL040

VEL041

VEL042

VEL043

VEL044

VEL045

VEL046

VEL047

VEL048

```

COMMON /B1/ FEJUNC,IR,DELTA,DLTA,IRS,TFJUNC,E,ES,FCAB,RCAB,JUMP,
1PJUNCS,PCAB,PCABE,PCABO,RCABO,THETA,PJUNCO
COMMON /B2/ NCAB,NNODE,ERJUNC,IRJUNC,DATA,DATN,H,PJUNC,
1CDCAB,DCAB,FATE,NANC,ANJUNC,IREAD,IPRNT,INTAPE,OUTAPE,ITIME,IFLG,
2OFLG,NIR,THETAS,THETA,COMPO,THETAU,NJUNC,RHO,TEST,
3NVSEG,ZVEL,VELZ,PIP,ECICAB,EXPCAB,ZJUNC,LJUNC,PATH,ICAB,IVOPT,
4WCAB,IDEV,ICHECK,NDEV,NDATC
DIMENSION FEJUNC(3,44),IR(3,44),IRS(3,44),TFJUNC(3,44),PJUNCO(3,44)
1)
DIMENSION FCAB(3,51,22),RCAB(3,51,22),PJUNCS(3,44),PCAB(3,51,22)
DIMENSION PCABE(3,51,22),PCABO(3,51,22),RCABO(3,51,22)
DIMENSION NNODE(22),ERJUNC(44),IRJUNC(44),DATA(10),DATN(10),H(22)
DIMENSION PJUNC(3,44),CDCAB(22),DCAB(22),ANJUNC(44),TEST(14)
DIMENSION ZVEL(25),VELZ(25),ECICAB(22),EXPCAB(22),ZJUNC(22)
DIMENSION LJUNC(22),PATH(22),ICAB(22),WCAB(22),IDEV(1000)
DIMENSION ICHECK(44)
DIMENSION PSPACE(3)
INTEGER OUTAPE,ZJUNC,ERJUNC,ANJUNC,OFLG
INTEGER PATH
REAL IR,IRS
GO TO (10,10,30),I
10 Z=PSPACE(3)
DO 11 KK=1,NVSEG
K=KK
IF(Z.GT.ZVEL(K)) GO TO 11
GO TO 12
11 CONTINUE
VFPS=(1.6878)*VELZ(K)
GO TO 14
12 IF(K.NE.1) GO TO 13
VFPS=(1.6878)*VELZ(K)
GO TO 14
13 SIGMA=Z-ZVEL(K-1)
SLOPE=(VELZ(K)-VELZ(K-1))/(ZVEL(K)-ZVEL(K-1))
VFPS=(1.6878)*(VELZ(K-1)+SLOPE*SIGMA)
14 GO TO (21,22),I
21 VELOC=VFPS*COS(THETA*PIP)
RETURN
22 VELOC=VFPS*SIN(THETA*PIP)
RETURN
30 VELOC=0.
RETURN
END

```

END

FILMED

4-86

DTIC