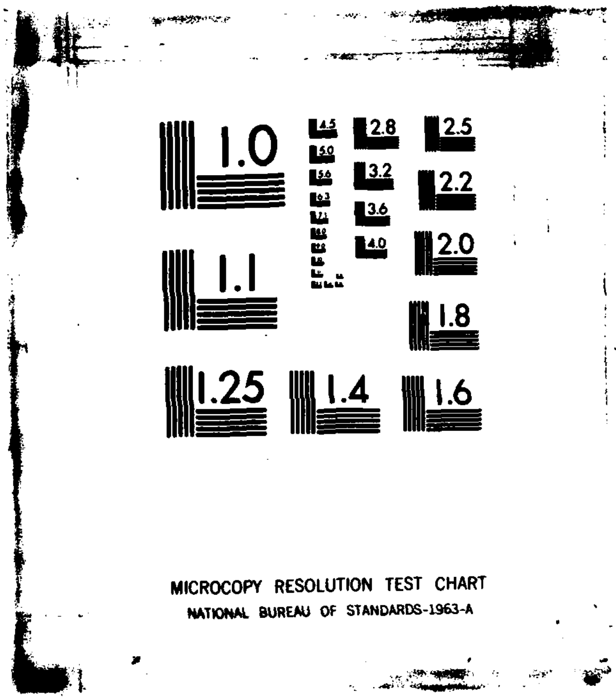
AD-A137 464 GENERAL ELECTROMAGNETIC OODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS..(U) BDM CORP ALBUQUERQU NM D L KADLEC ET AL. SEP 83 BDM/A-83-020-TR-VOL-3-PT-2 1/4

UNCLASSIFIED RADC-TR-83-217-VOL-3-PT-2 F30602-81-C-0084 F/G 20/14 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

RADC-TR-83-217, Vol III (of three), Pt 2
Final Technical Report
September 1983

GENERAL ELECTROMAGNETIC MODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS) Computer Code Documentation (Version 3)

The BDM Corporation

Dr. D. L. Kadlec and Dr. E. L. Coffey

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

RADC-TR-83-217, Vol III, Part 2 (of three) has been reviewed and is approved for publication.

APPROVED:

KENNETH R. SIARKIEWICZ
Project Engineer

APPROVED:

W.S. TUTHILL, Colonel, USAF
Chief, Reliability & Compatibility Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
RADC-TR-83-217, Vol III, Part 2

2. GOVT ACCESSION NO.
AD-A137464

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
GENERAL ELECTROMAGNETIC MODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS)
COMPUTER CODE DOCUMENTATION (Version 3)

5. TYPE OF REPORT & PERIOD COVERED
Final Technical Report
February 81 - July 83

6. PERFORMING ORG. REPORT NUMBER
BDM/A-83-020-TR

7. AUTHOR(s)
Dr. Diana L. Kadlec
Dr. Edgar L. Coffey

8. CONTRACT OR GRANT NUMBER(s)
F30602-81-C-0084

9. PERFORMING ORGANIZATION NAME AND ADDRESS
The BDM Corporation
1801 Randolph Road, S.E.
Albuquerque NM 87106

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62702F
23380333

11. CONTROLLING OFFICE NAME AND ADDRESS
Rome Air Development Center (RBCT)
Griffiss AFB NY 13441

12. REPORT DATE
September 1983

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Same

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
Same

18. SUPPLEMENTARY NOTES
RADC Project Engineer: Kenneth R. Siarkiewicz (RBCT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Electromagnetic Compatibility
Method of Moments (MOM)
Geometrical Theory of Diffraction (GTD)
Antenna Analysis
Matrix Equation Solution
MOM/GTD Hybridization
EM Radiation and Scattering

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

GEMACS solves electromagentic radiation and scattering problems. The Method of Moments (MOM) and Geometrical Theory of Diffraction (GTD) are used. MOM is formalized with the Electric Field Integral Equation (EFIE) for wires and the Magnetic Field Integral Equation (MFIE) for patches. The code employs both full matrix decomposition and Banded Matrix Iteration (BMI) solution techniques. The MOM, GTD and hybrid MOM/GTD techniques in the code are used to solve electrically small object problems, electrically

large object problems and combination sized object problems.

Volume I of this report is the User Manual. The code execution requirements, input language and output are discussed.

Volume II is the Engineering Manual. The theory and engineering approximations implemented in the code are discussed. Modeling criterion are given.

Volume III is the Computer Code Documentation Manual. This manual contains extensive software information of the code.

1. NAME: FLDDRV (OUTPUT)

2. PURPOSE: To convert a real/imaginary field data set to magnitude/ format and to set up the data set for subsequent printing and/or plotting.

3. METHOD: A previously calculated field data set is retrieved, the contents converted to magnitude/phase format, and resultant values stored. FLDOUT is called to print and plot the field data.

   FLDDRV will accept coordinates for three different systems: Cartesian, cylindrical, and spherical. The initial coordinates will be received in a specific order. This order determines the order in which each coordinate is incremented. The following is an example command:

   NEAR = EFIELD (CURDEN) LINLIN

DX = 1.	DY = 10.	DZ = 10.
X2 = 10.	Y2 = 10.	Z2 = 10.
Z1 = 0.	Y1 = 0.	X1 = 0.

   Z will be on the outer loop. Y will be on the middle loop, and X will be on the inner loop. This is determined by looking at the initial coordinates Z1, Y1, and X1.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CI	Array containing imaginary parts of electric field vectors for an observation point
CR	Array containing real parts of electric field vectors for an observation point
ESQR	Magnitude of the electric field squared
FRFLD	Logical far-field flag
I	Loop index
IBLANK	Hollerith field with all blank characters
IBLK	Loop index
IC	Loop index
ICOLA	Number of columns added to the field matrix

ICOLMN	Column counter for the output field matrix
ICOL1	ICOLMN + 1
ICORDT	Coordinate keyword table used to find all the coordinate positions and increments
ICOST	Coordinate order and system table. This table will tell which coordinates are required for a system (Cartesian, cylindrical, and spherical). Should an improper coordinate type be specified, an error is generated. It also will get the order for near-field patterns.
ICTYPE	Type of coordinate system, formed by adding the location of system constitutive parameters in ICOST: 6 = rectangular (1 + 2 + 3) 12 = cylindrical (4 + 5 + 3) 15 = spherical (4 + 5 + 6)
II	Pointer to PHSMAG array to store magnitude and phase of field components
INCORE	Logical flag which indicates that interpolation coefficients are stored in core memory
INDEX	Saves the order of the coordinates
INDEX1	Index to the U array for the first coordinate
INDEX2	Index to the U array for the second coordinate
INDEX3	Index to the U array for the third coordinate
INDXA	Index to the field data set in symbol table
INDXB	Index to the solution or source data set in segment table
IPLOT	Plot type to be used by the plotting routine
IROWA	Number of rows in a column

IROW2	Number of rows in the solution matrix
IU	The coordinate information array (equivalenced to U)
IWORDS	Message array
J	Loop index
JSAV	Index to ICORDT for coordinate system type
K	Loop index
KWA	Keyword argument
LINKA	Index of the field matrix to link the field matrix to the solution data
LOOP	The loop array containing the number of times to perform each loop
LOOP1	Outer loop limit
LOOP2	Middle loop limit
LOOP3	Inner loop limit
LOPINR	Index of inner pattern loop
LOPMID	Index of middle pattern loop
LOPOUT	Index of outer pattern loop
LORDER	Array containing the order for all coordinate systems
MASK	Used to determine the required coordinate for a system
N	Loop index
NAM	Hollerith format of NAMEA
NAME	Name of the solution data set
NAMEA	Name of the field data set
NAMEB	Name of the solution or source data set

NBITA	Far- or near-field attribute for the field matrix
NDX	Pointer to coordinate number in INDEX array
NDXINR	Index to the coordinate in the U array and in the inner loop
NDXMID	Index to the coordinate in the U array and in the middle loop
NDXOUT	Index to the coordinate in the U array and in the outer loop
NINC	The increment size for filling a column in the field matrix (number of real words per field point)
NP	Loop index
NPI	Index to TEMP for vector components of field point
NPIC	Pointer to TEMP to retrieve imaginary part of field value
NPRC	Pointer to TEMP to retrieve real part of field value
NPRFPT	NINC/2 (number of vector components at field point)
NSAV	Pointer to first TEMP entry for a pattern cut
NXTARG	Next argument to be evaluated
PHSMAG	Phase and magnitude component parts of the complex electric field for each position
PWRMAX	Maximum power of the structure
RDTODG	Conversion factor from radians to degrees
U	Array containing initial and final positions plus the increment for each coordinate

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DGTORD	/GEODAT/
FLTARG	/ARGCOM/
INTARG	/ARGCOM/
IPASS	/ARGCOM/
ISON	/ADEBUG/
KBFFLD	/PARTAB/
KBNFLD	/PARTAB/
KBREAL	/PARTAB/
KOLBIT	/PARTAB/
KOLLNK	/PARTAB/
KOLNAM	/PARTAB/
KOLROW	/PARTAB/
KWNAME	/PARTAB/
LUPRNT	/ADEBUG/
NCODES	/PARTAB/
NDATBL	/PARTAB/
NOPCOD	/ADEBUG/
NPDATA	/PARTAB/
NTFLPT	/ADEBUG/
NUMARG	/ARGCOM/
ZERO	/ADEBUG/

B. OUTPUT

OUTPUT	LOCATION
IERRF	/ADEBUG/
INTARG	/ARGCOM/

IWORDS	/ADEBUG/
NOGOFG	/SCNPAR/
TEMP	/TEMP01/

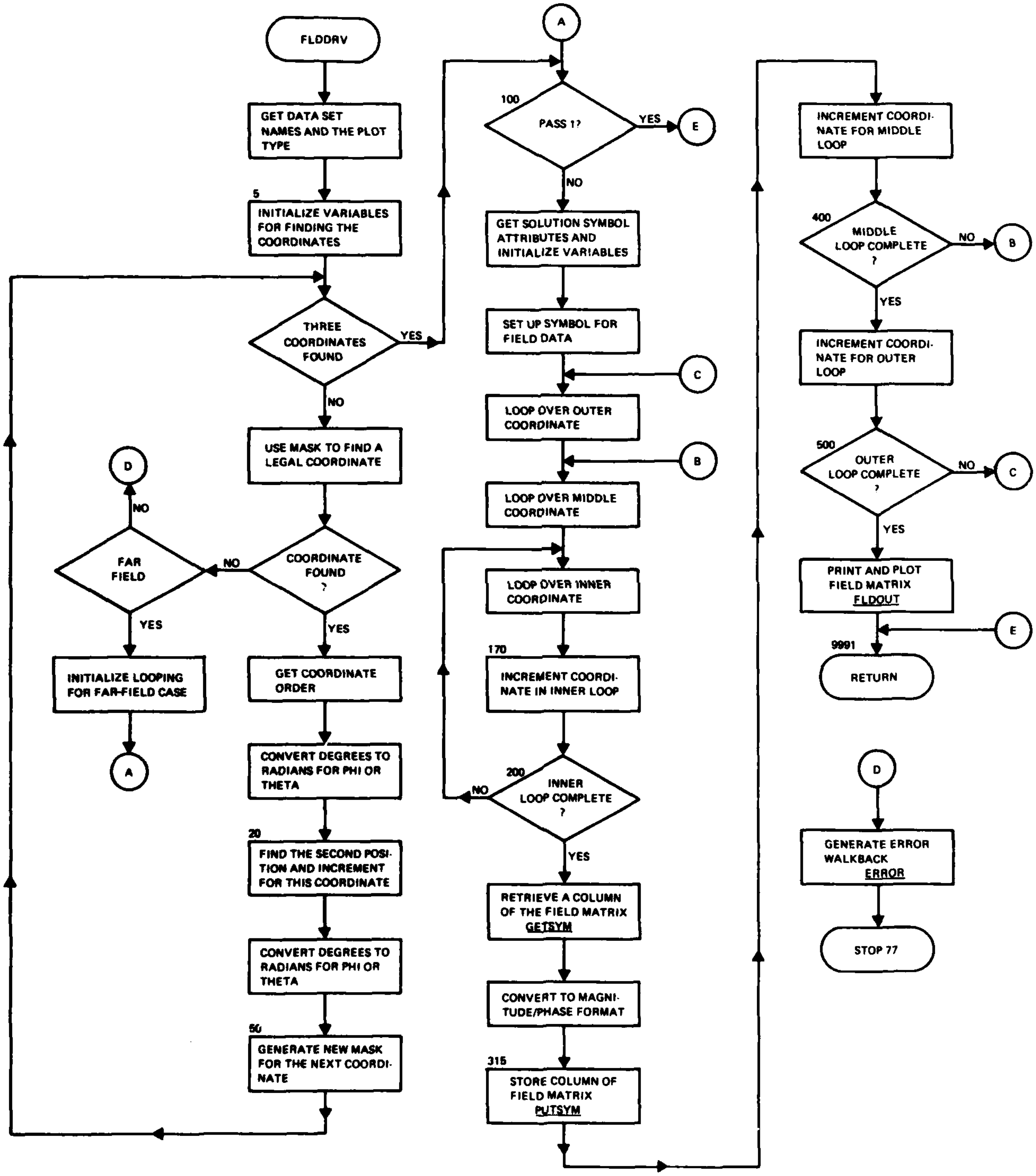
6. CALLING ROUTINE:

TSKXQT

7. CALLED ROUTINES:

ASSIGN	GETSYM	SYMDEF
CONVRT	IBITCK	SYMUPD
ERROR	PUTSYM	WLKBCK
FLDOUT	STATIN	
GETARG	STATOT	

FLDDRV (OUTPUT)



1. NAME: FLDOUT (OUTPUT)

2. PURPOSE: Print and plot electric field data.

3. METHOD: FLDOUT reads the plot matrix generated by FLDDRV. Then each column is listed and plotted on the line printer.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ANG	Polar coordinate
ARG	Electric field magnitude for each coordinate
CINC	Step size for inner loop
CV	Coordinate values
DYNAMR	Error size
EMAXSQ	Sum of the electric fields squared
FAR	Logical flag for a far-field case
HEAD	Array containing headers for printing
IDPLOT	Input plot type
INR	Pointer to the coordinate on the inner loop
IR	Index to record to be retrieved from the plot file
IREC1	First record of data file
IREC2	Last record of data file
JAXIS	The type of plot for subroutine PAGPLT
KIND	Type of coordinate system
LABEL	Array containing labels for printing
LBL	Temporary array for printing
LINK	Index to solution or geometry data set

LNKBIT	Attribute of linked solution or geometry data set
LOG	Logical flag for a log plot
NAMFLD	Name of the plot file
NBITS	Attribute word for the plot file
NBLK	Blank word
NDXFLD	Index to the plot file
NE	Plot file name to be printed
NEAR	Logical flag for near field
NEED	Number of words needed for temporary storage
NG	Name
NHDWRD	Number of words in the header of a file
NPRFPT	Number of words per point
NPRHDR	Size of the header
NPRREC	Number of rows in a column of a matrix
NT1	First record
NT1SAV	First record saved
NT2	Last record
NT2SAV	Last record saved
NUMREC	Number of entries in a column
PHS	Phase of the electric field components
PWR	Power of the electric field
PWRDWN	Power down at any field point referenced to the maximum for that calculation loop (log)
PWRMAX	Power maximum for that calculation loop

PWRRAT	Power ratio of power at a field point to the maximum for that calculation loop (lin)
PWRSQ	Power squared
RADMAX	Maximum radius
X	Rectangular position of a polar plot point
XMAX	Maximum value for X
XMIN	Minimum value for X
Y	Rectangular position of a polar plot point

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
EMAXSQ	F.P.
IDPLOT	F.P.
IREC1	F.P.
IREC2	F.P.
NAMFLD	F.P.
NDATBL	/PARTAB/
NPDATA	/PARTAB/

B. OUTPUT:

None

6. CALLING ROUTINE:

FLDDRV

7. CALLED ROUTINES:

ASSIGN

CONVRT

ERROR

FLDOUT (OUTPUT)

GETSYM

IBITCK

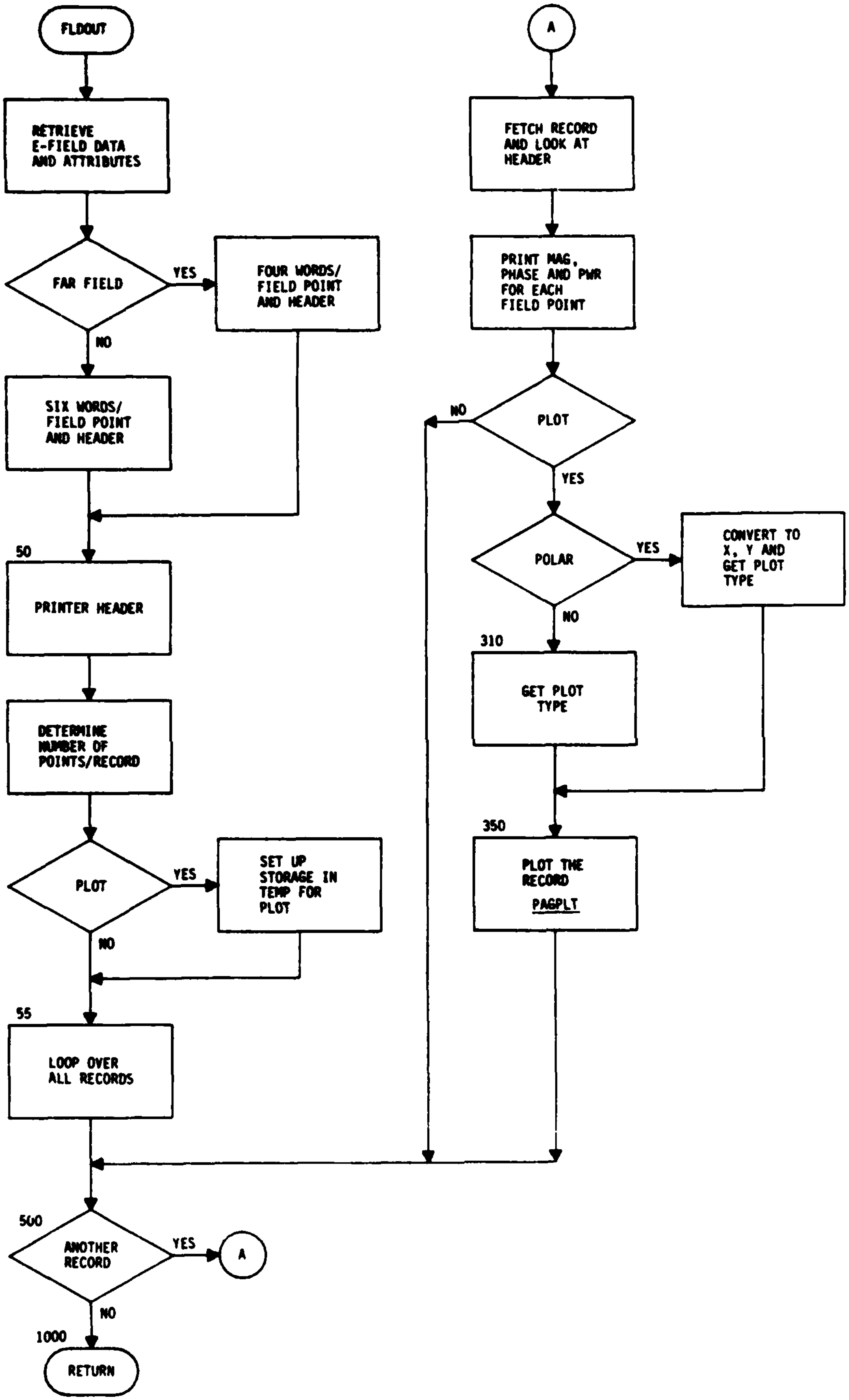
PAGPLT

STATIN

STATOT

WLKBCK

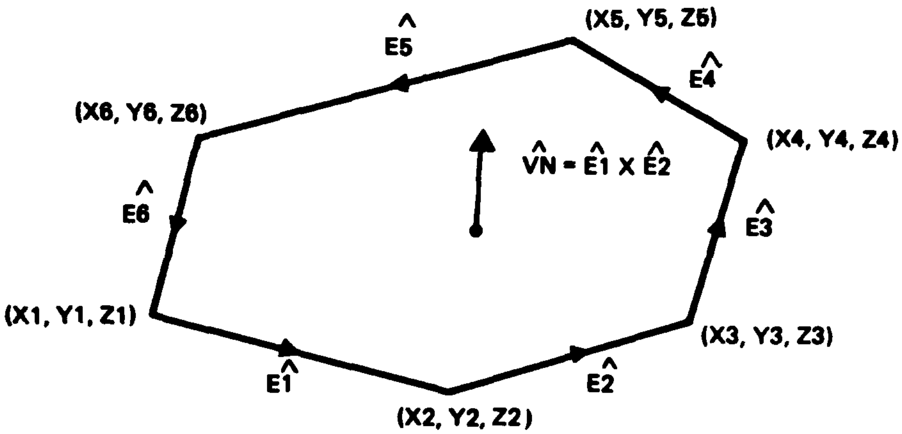
FLDOUT (OUTPUT)



1. NAME: FLTPLT (INPUT)

2. PURPOSE: To determine if all corners of a plate lie in a plane.

3. METHOD: Flatness is checked by comparing the plate normal to each plate edge. If the normal vector is perpendicular to all edge unit vectors, the plate is flat. The plate normal is generated from plate edges 1 and 2. Edges are generated from adjacent corners as shown in the figure.



4. INTERNAL VARIABLES:

VARIABLE	DESCRIPTION
DOT	Dot product of plate normal and edge unit vectors
E1X, E1Y, E1Z	Vector describing edge 1
E2X, E2Y, E2Z	Vector describing edge 2
IERF	Flatness flag (0 = flat plate)
M	Index over plate edges
M1	Index of first corner of edge M
M2	Index of second corner of edge M
ME	Number of edges

V	Edge unit vector
VM	Magnitude of edge vector
VN	Plate normal unit vector
VNM	Magnitude of plate normal vector
X	Array of corner x-coordinates
Y	Array of corner y-coordinates
Z	Array of corner z-coordinates

5. I/O VARIABLES:

A. INPUT — LOCATION

INPUT	LOCATION
ISOFF	/ADEBUG/
ISON	/ADEBUG/
ME	F.P.
X	F.P.
Y	F.P.
Z	F.P.

B. OUTPUT — LOCATION

OUTPUT	LOCATION
IERF	F.P.

6. CALLING ROUTINE:

PLATE

7. CALLED ROUTINES:

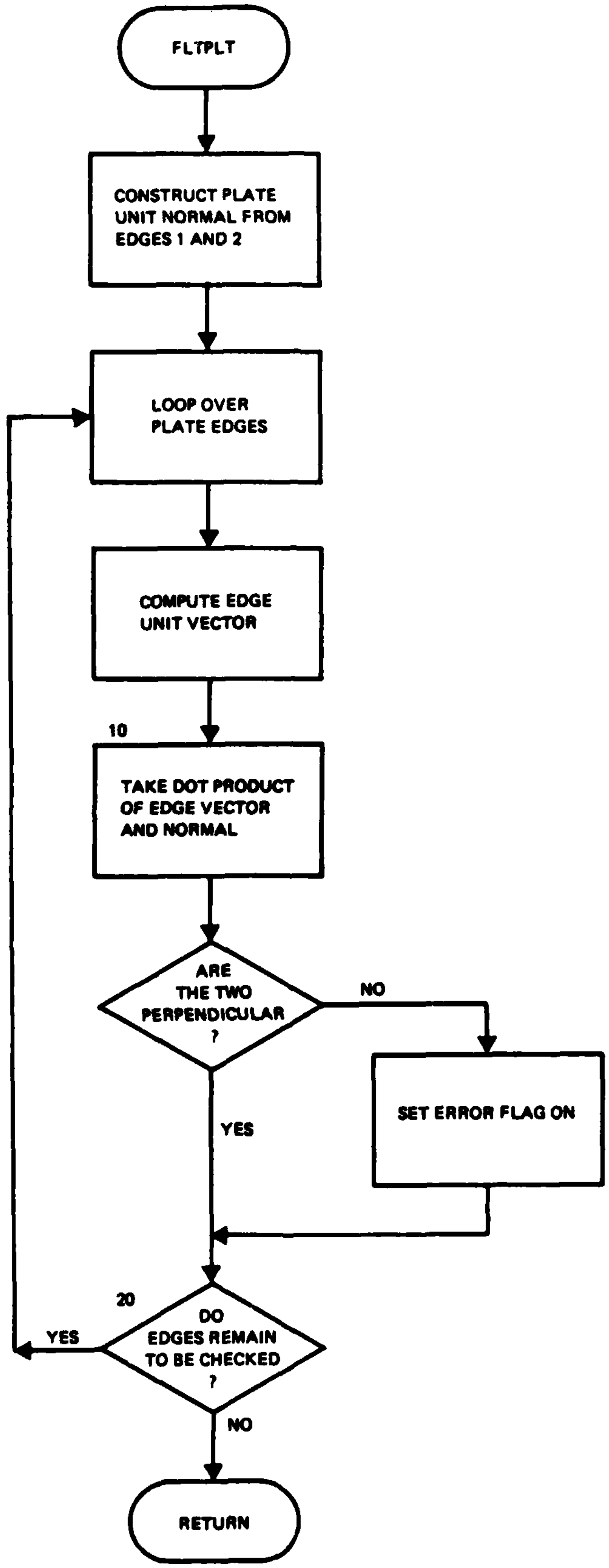
ASSIGN

STATIN

STATOT

WLKBCK

FLTPLT (INPUT)



1. NAME: FNDARG (INPUT)

2. PURPOSE: Find argument(s) for argument list.

3. METHOD: There are two forms of argument types that can be input to FNDARG:

(a) Argument type >0. This number is put directly into the argument list.

(b) Argument type <0. In this case, the SCAN table pointer is moved from left to right until an item that is not an operator or has not been used is found. The absolute value of the argument type is unpacked like keywords are in PARSE. The following is the formula used for argument types where MXARGT is the maximum number of argument types + 1 (32), MLTARG is the packed argument type(s), and ARGTYP is the array used to receive the argument type:

$$MLTARG_{i+1} = IFIX (MLTARG_i/MXARGT)*MXARGT$$

$$NARGTP_i = MLTARG_i - MLTARG_{i+1}$$

$$i = 1 \text{ to } 6$$

FNDARG expects the argument(s) in the command text to be in the same order as those in the argument type array. Starting at the current position of the SCAN table pointer, it loops through the array of argument types to pick up arguments from SCAN tables. These arguments will be put in the argument list by a portion of the code that handles each argument type. If the argument is not there or is the wrong type, a no op code is left in the argument list.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
INC	Internal function incrementing NPARGL
INDEX	Value to be put in the argument list
MINUS1	-1
MLTA	Second temporary location for multiple argument types
MLTARG	First temporary location for multiple argument types
NA	Index for unpacking arguments

NCDNTB	Is equated to NCODE (NTAB)
NDXARG	Temporary location for argument type
NKEYW	Keyword number
NLOOP	Loop level pointer
NPARGT	Input argument type
NTFMTP	Ta,k format table pointer
NTFP	Task format table pointer + 1
N1	Points to first argument in argument list to start scanning for ILP keyword in the DEBUG command

5. I/O VARIABLES:

A.

INPUT	LOCATION
IMINUS	/SCNPAR/
IPLUS	/SCNPAR/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
KOLCNT	/PARTAB/
KOLLBL	/PARTAB/
KOLTIM	/PARTAB/
KOLTSK	/PARTAB/
KWFMTP	/PARTAB/
KWOFF	/PARTAB/
KWTRAC	/PARTAB/
LOOPMX	/PARTAB/
LSTASK	/SCNPAR/
LSTDAT	/SCNPAR/

FNDARG (INPUT)

LSTINT	/SCNPAR/
LUPRNT	/ADEBUG/
MXARGT	/PARTAB/
MXMAT	/PARTAB/
NARGMX	/PARTAB/
NCODE	/SCNPAR/
NDATMX	/PARTAB/
NPARGL	/PARTAB/
NPARGT	F.P.
NPEARG	/INPERR/
NPELAB	/INPERR/
NPELNL	/INPERR/
NPELOO	/INPERR/
NPELOP	/INPERR/
NPESCN	/INPERR/
NPETSK	/INPERR/
NPLOOP	/PARTAB/
NPTASK	/PARTAB/
NTAB	/SCNPAR/
NTALPH	/ADEBUG/
NTEND	/ADEBUG/
NTFLPT	/ADEBUG/
NTINT	/ADEBUG/
NTKEYW	/ADEBUG/
NTSFMT	/PARTAB/

NTSYMB	/ADEBUG/
NVAL	/SCNPAR/

B. OUTPUT LOCATION

LTRACE	/ADEBUG/
LUDBUG	/ADEBUG/
MATCH	/SCNPAR/
NARGTB	/PARTAB/
NCODE	/SCNPAR/
NDEBUG	/SCNPAR/
NLOOPS	/PARTAB/
NOMTCH	/SCNPAR/
NPTASK	/PARTAB/
NTAB	/SCNPAR/

7. CALLING ROUTINE:

PARSE

8. CALLED ROUTINES:

ASSIGN

FABLO2

LITSCH

PLIST

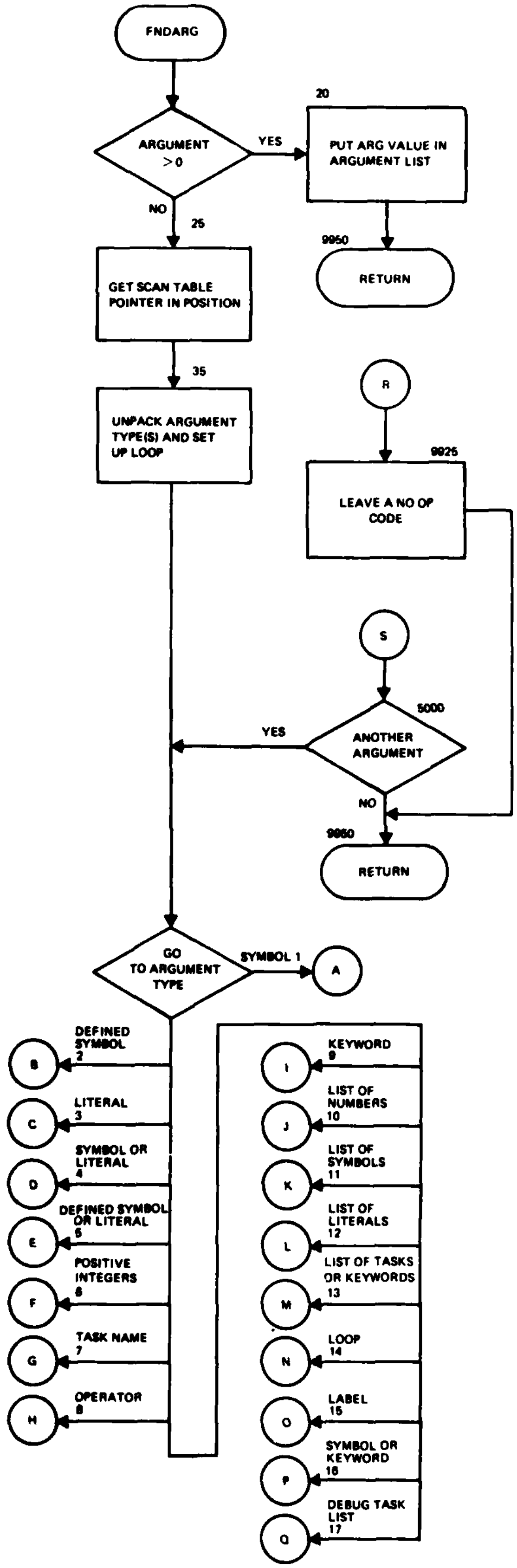
STATIN

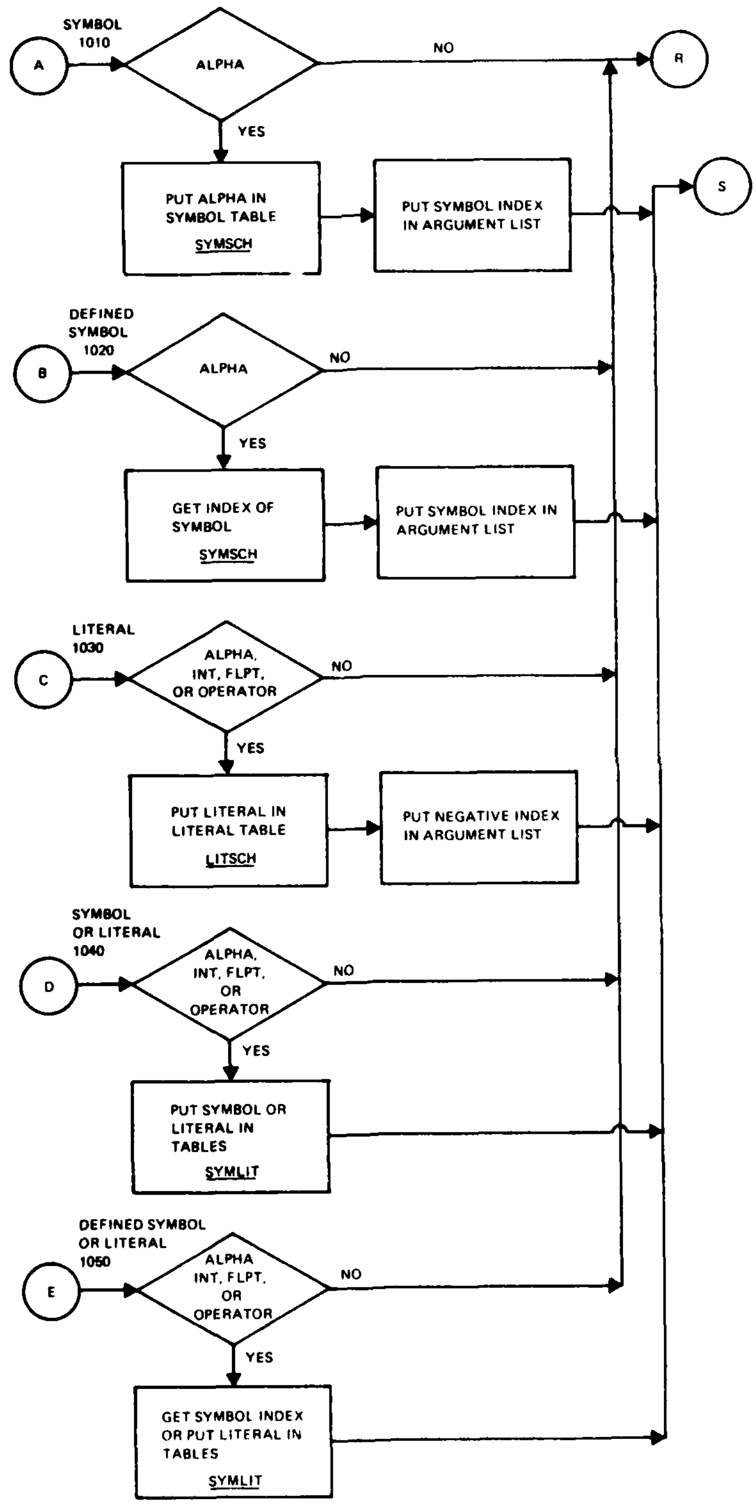
STATOT

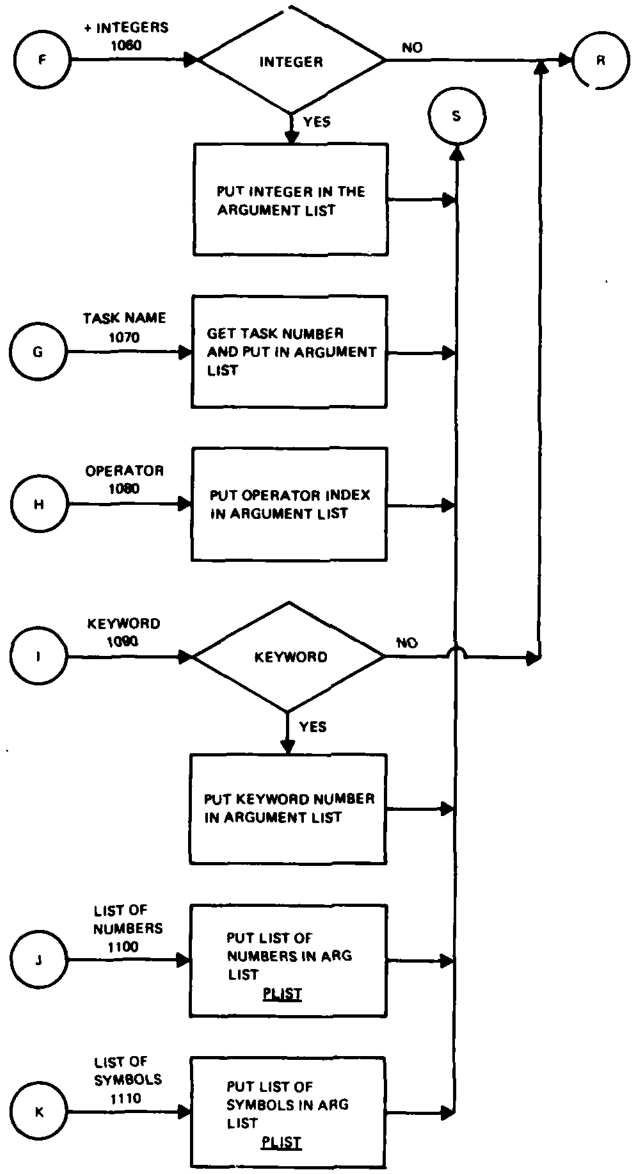
SYMLIT

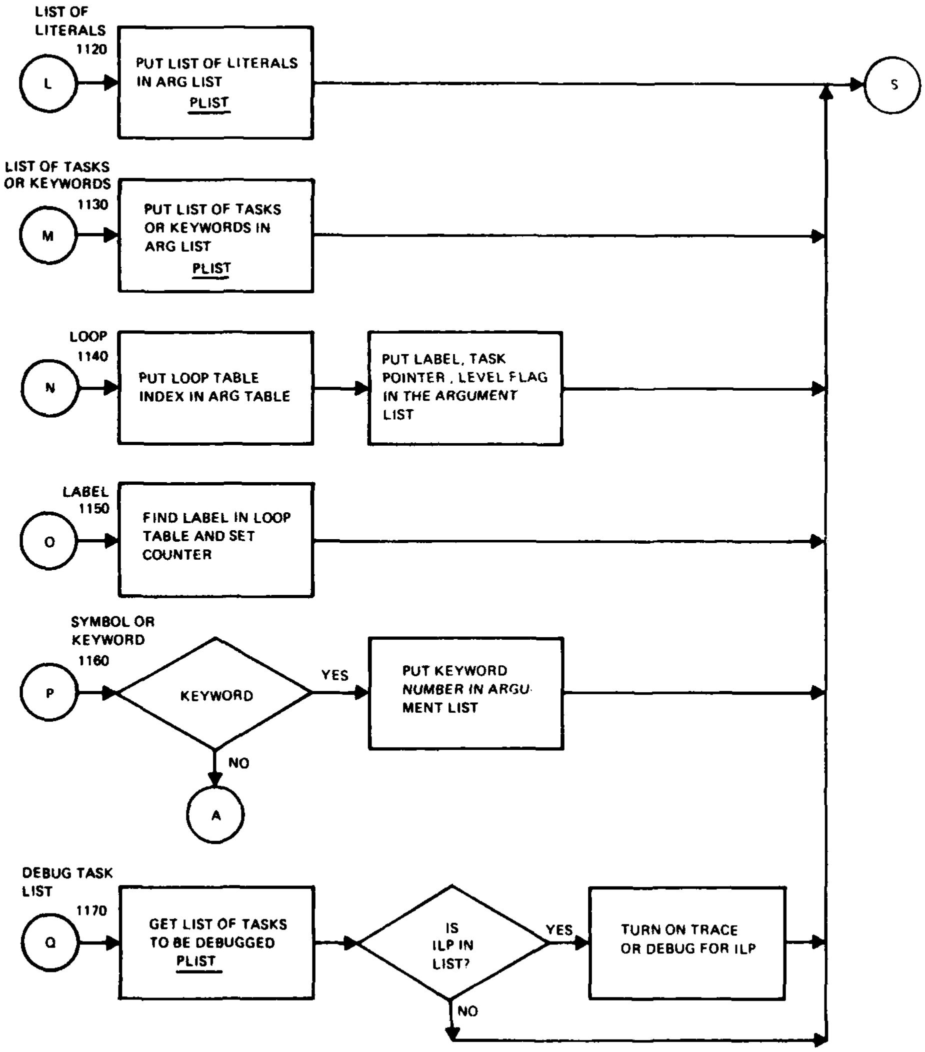
SYMSCH

WLKBCK


FNDARG
ARGUMENT > 0
YES
NO
20
PUT ARG VALUE IN ARGUMENT LIST
9950
RETURN
25
GET SCAN TABLE POINTER IN POSITION
35
UNPACK ARGUMENT TYPE(S) AND SET UP LOOP
R
9925
LEAVE A NO OP CODE
S
5000
ANOTHER ARGUMENT
YES
NO
9950
RETURN
GO TO ARGUMENT TYPE
SYMBOL 1
A
DEFINED SYMBOL 2
B
LITERAL 3
C
SYMBOL OR LITERAL 4
D
DEFINED SYMBOL OR LITERAL 5
E
POSITIVE INTEGERS 6
F
TASK NAME 7
G
OPERATOR 8
H
KEYWORD 9
I
LIST OF NUMBERS 10
J
LIST OF SYMBOLS 11
K
LIST OF LITERALS 12
L
LIST OF TASKS OR KEYWORDS 13
M
LOOP 14
N
LABEL 15
O
SYMBOL OR KEYWORD 16
P
DEBUG TASK LIST 17
Q


A
SYMBOL
1010
ALPHA
NO
R
YES
PUT ALPHA IN SYMBOL TABLE
SYMSCH
PUT SYMBOL INDEX IN ARGUMENT LIST
S
B
DEFINED SYMBOL
1020
ALPHA
NO
YES
GET INDEX OF SYMBOL
SYMSCH
PUT SYMBOL INDEX IN ARGUMENT LIST
C
LITERAL
1030
ALPHA, INT, FLPT, OR OPERATOR
NO
YES
PUT LITERAL IN LITERAL TABLE
LITSCH
PUT NEGATIVE INDEX IN ARGUMENT LIST
D
SYMBOL OR LITERAL
1040
ALPHA, INT, FLPT, OR OPERATOR
NO
YES
PUT SYMBOL OR LITERAL IN TABLES
SYMLIT
E
DEFINED SYMBOL OR LITERAL
1050
ALPHA INT, FLPT, OR OPERATOR
NO
YES
GET SYMBOL INDEX OR PUT LITERAL IN TABLES
SYMLIT


+ INTEGERS
1060
F
INTEGER
NO
R
YES
S
PUT INTEGER IN THE ARGUMENT LIST
TASK NAME
1070
G
GET TASK NUMBER AND PUT IN ARGUMENT LIST
OPERATOR
1080
H
PUT OPERATOR INDEX IN ARGUMENT LIST
KEYWORD
1090
I
KEYWORD
NO
YES
PUT KEYWORD NUMBER IN ARGUMENT LIST
LIST OF NUMBERS
1100
J
PUT LIST OF NUMBERS IN ARG LIST
PLIST
LIST OF SYMBOLS
1110
K
PUT LIST OF SYMBOLS IN ARG LIST
PLIST


LIST OF LITERALS
1120
L
PUT LIST OF LITERALS IN ARG LIST
PLIST
S
LIST OF TASKS OR KEYWORDS
1130
M
PUT LIST OF TASKS OR KEYWORDS IN ARG LIST
PLIST
LOOP
1140
N
PUT LOOP TABLE INDEX IN ARG TABLE
PUT LABEL, TASK POINTER, LEVEL FLAG IN THE ARGUMENT LIST
LABEL
1150
O
FIND LABEL IN LOOP TABLE AND SET COUNTER
SYMBOL OR KEYWORD
1160
P
KEYWORD
YES
PUT KEYWORD NUMBER IN ARGU-MENT LIST
NO
A
DEBUG TASK LIST
1170
Q
GET LIST OF TASKS TO BE DEBUGGED
PLIST
IS ILP IN LIST?
YES
TURN ON TRACE OR DEBUG FOR ILP
NO

1. NAME: FNDREC (GTD, INPUT, MOM, OUTPUT)

2. PURPOSE: Find the global record number of a global word location on a multiple edition data file.

3. METHOD: The attributes of the data set are extracted from the symbol table and the number of words per data set element calculated. The record number of a word in a full matrix is determined by simple division. For a triangular matrix or banded triangular matrix, the record number is found by stepping through the nonequal length matrix records until the total length of records stepped through is greater than or equal to the word location desired.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IBAND	Flag indicating a banded matrix
IFILE	Logical unit number on which data reside
ILOWER	Flag indicating a lower triangular matrix
IORDER	Flag indicating a transposed matrix
IREC	Global record number of LOC
ISV	Pointer to NDATBL entry for matrix
IUPPER	Flag indicating an upper triangular matrix
IWRD	Number of words analyzed for a triangular matrix
KBAND	Flag indicating that data set linked to matrix is banded
KLINK	Pointer to data set linked to data set linked to matrix
LINK	Pointer to data set linked to matrix
LOC	Matrix word for which record number is desired
LOCFST	Location of beginning of present edition of data set
LOCLST	Location of end of present edition of data set

LWRUPR	Flag indicating a triangular matrix
MATNAM	User-assigned name to this matrix
MORE	Flag indicating existence of attribute
NA	Hollerith version of MATNAM
NAMSAV	Matrix name in last call to FNDREC
NBITWD	Bit attribute word of matrix
NPRELM	Number of data words per matrix element
NPRPRT	Number of rows in parent matrix
NPRREC	Number of words in record of matrix
NUMROW	Number of rows in matrix

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
KBBAND	/PARTAB/
KBCPLX	/PARTAB/
KBDPRE	/PARTAB/
KBLWRT	/PARTAB/
KBORDR	/PARTAB/
KBUPRT	/PARTAB/
KOLAST	/PARTAB/
KOLBIT	/PARTAB/

	KOLFST	/PARTAB/
	KOLLNK	/PARTAB/
	KOLLOC	/PARTAB/
	KOLNAM	/PARTAB/
	KOLROW	/PARTAB
	LOC	F.P.
	LUPRNT	/ADEBUG/
	MATNAM	F.P.
	NDATBL	/PARTAB/
	NDFILE	/IOFLES/
	NPDATA	/PARTAB/
B.	OUTPUT	LOCATION
	IERRF	/ADEBUG/
	IREC	F.P.

6. CALLING ROUTINES:

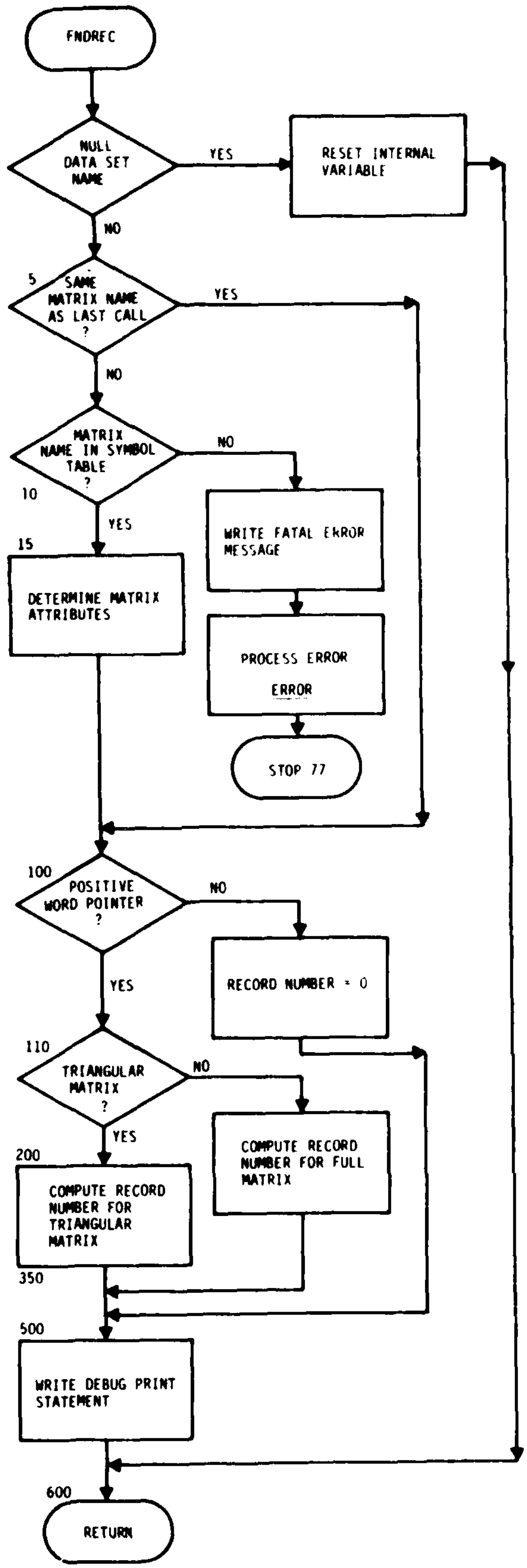
GETSYM

PUTSYM

7. CALLED ROUTINES:

ASSIGN	STATIN
CONVRT	STATOT
ERROR	WLKBCK
IBITCK	
IJMOD (intrinsic)	

FNDREC (GTD, INPUT, MOM, OUTPUT)



1. **NAME:** FRNELS (GTD)

2. **PURPOSE:** To compute the Fresnel integral,

$$f(x_s) = \int_0^{x_s} e^{-j\pi/2\ u^2}\, du = C(x_s) - j\ S(x_s)\ .$$

3. **METHOD:** The integral is evaluated using an approximation by J. Boersma (see reference A). The integral

$$f(x) = \int_0^{x} \frac{e^{-jt}}{\sqrt{2\pi t}}\, dt$$

is approximated as follows:

$$\text{for } 0 \le x \le 4 \quad f(x) \approx e^{-jx}\sqrt{\frac{x}{4}} \sum_{n=0}^{11} (a_n + jb_n) \left(\frac{x}{4}\right)^n$$

$$\text{for } x \ge 4 \quad f(x) \approx \frac{1-j}{2} + e^{-jx}\sqrt{\frac{4}{x}} \sum_{n=0}^{11} (c_n + d_n) \left(\frac{4}{x}\right)^n$$

(the constants a_n, b_n, c_n, and d_n are provided by Boersma and are defined in data statements in the subroutine).

Note that by performing a change of variable, the integral to be solved becomes of the form of the integral which Boersma solved;

$$t = \frac{\pi}{2}\ u^2$$

By applying this change of variable, we get

$$f(x_s) = \int_0^{x_s} e^{-j\frac{\pi}{2}u^2}\, du = \int_0^{x} \frac{e^{-jt}}{\sqrt{2\pi t}}\, dt$$

where

$$x = \frac{\pi}{2} x_s^2 \quad .$$

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A,B,CC,D	Constants used in evaluating integral
C	Real part of Fresnel integral
FI	Imaginary component of summation function
FR	Real component of summation function
PI	π
S	Imaginary part of Fresnel integral
XS	Limit of integration

5. I/O VARIABLES:

A. INPUT	LOCATION
PI	/PIS/
XS	F.P.

B. OUTPUT	LOCATION
C	F.P.
S	F.P.

6. CALLING ROUTINES:

DICOEF

DPI

FFCT

FKY

RPLSCL

SCLRPL

SCTCYL

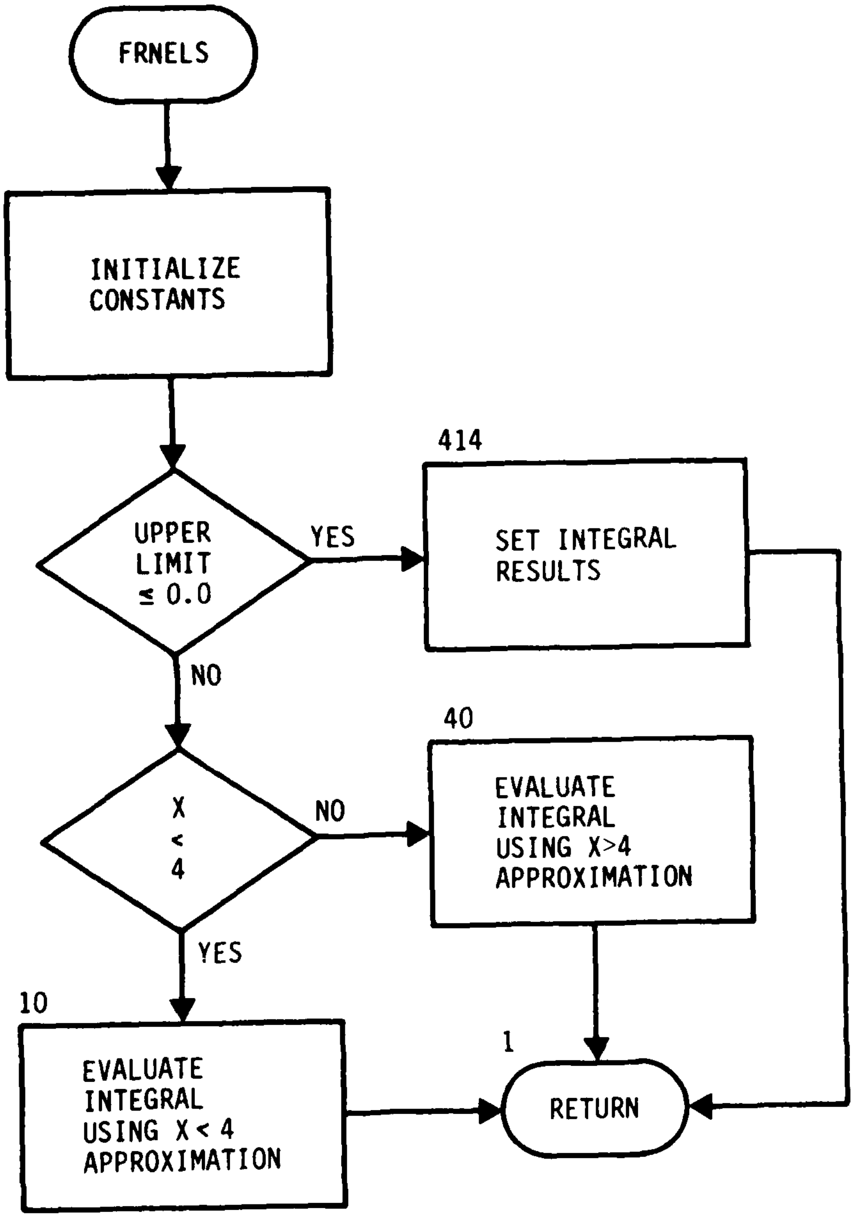
7. CALLED ROUTINE:

NONE

8. REFERENCE:

A. J. Boersma, "Computation of Fresnel Integrals," Math Comp., Vol. 14, 1960, p. 380.

FRNELS (GTD)


FRNELS
INITIALIZE CONSTANTS
UPPER LIMIT ≤ 0.0
YES
414
SET INTEGRAL RESULTS
NO
X < 4
NO
40
EVALUATE INTEGRAL USING X>4 APPROXIMATION
YES
10
EVALUATE INTEGRAL USING X < 4 APPROXIMATION
1
RETURN

1. NAME: FUNI (GTD)

2. PURPOSE: This function calculates the integrand of the integral in subroutine FKARG.

3. METHOD: The integrand of the integral evaluated in subroutine FKARG is given by

$$\text{FUNI(VR)} = \frac{1}{\sqrt{A^2\sin^2(\text{VR}) + B^2\cos^2(\text{VR})}}$$

This function simply performs this division and returns the result in FUNI.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	Radius of cylinder on x-axis
B	Radius of cylinder on y-axis
FUNI	Integrand used in FKARG
VR	Elliptic angle on cylinder in radians

5. I/O VARIABLES:

A. INPUT — LOCATION

INPUT	LOCATION
A	/GEOMEL/
B	/GEOMEL/
VR	F.P.

B. OUTPUT — LOCATION

OUTPUT	LOCATION
FUNI	FUNCTION

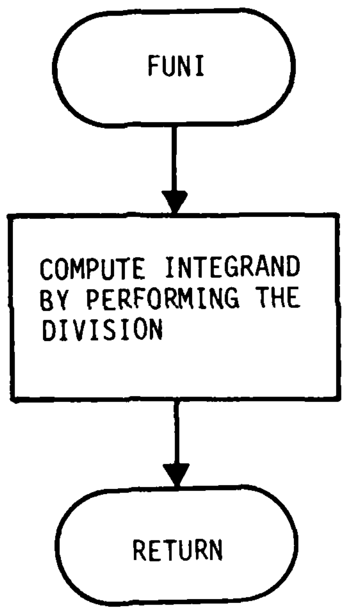
6. CALLING ROUTINE:

FKARG

7. CALLED ROUTINE:

NONE

FUNI (GTD)



1. NAME: GEODRV (INPUT)

2. FUNCTION: This subroutine interfaces the geometry processing subroutines to the task execution processor.

3. METHOD: This subroutine first retrieves the geometry processor arguments and initializes all of the tables. Then the subroutine calls WYRDRV to read the geometry input. Next the wires and then the patches are sorted into ascending order, the multiple junctions determined, and the junction linkage set up. Cylinders and end caps are linked by LNKGTD. Finally the subroutine prints out the ordered list of wires, patches, and GTD objects with their coordinates and other data.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AREAP	Area of a patch
COSALP	RHO/SEGL
COSBET	DX/RHO
DX	Difference in end points for x
DY	Difference in end points for y
DZ	Difference in end points for z
GAREA	Total surface area of wires
IBITS	Attributes to be added to the attribute word
IBLK	Data block index
IEND	Index pointing to the wire connected to the patch
IEND1	Bias connection data for end 1
IEND2	Bias connection data for end 2
ILIM	The number of segments in current data block
ILINE	Number of lines that have been printed out

INPBLK	First block containing patch data
IPASS	Integer value of pass through task list
IPAT	Patch segment identifier
IPLIM	The number of segments in patch data blocks
IPLOW	The location of first patch segment in patch data blocks
ISEG	Wire segment identifier
ITAG	Tag identifier
JBIAS1	Connection bias for end 1 of a wire segment
JBIAS2	Connection bias for end 2 of a wire segment
JBIAS3	Connection bias for wire segment connected to a patch
JCBIAS	Integer to indicate junction
LOCNAM	Location in the symbol table for the geometry data
LUSAVE	Saves LUTASK
LUTASK	Logical unit for reading task
MAXBLK	Total number of data blocks
MAXSEG	Maximum number of segments per data block
MLTJCT	Index to a multiple connection junction
MXBLKW	Number of wire data blocks
NAMGEO	Pointer to default geometry name
NCONT	The number of subpatches connected to a segment
NDXBLK	Index to the data block currently in use
NPATCH	The number of patch segments

NUMSEG	The total number of wire and patch segments
NWIRE	The number of wire segments
PAREA	The total surface patch area
PHI	The azimuthal angle of the patch normal vector
RADII	Wire radius
RH	Intermediate value for normal vector
RHO	$[(DX)^2 + (DY)^2]^{\frac{1}{2}}$ for wire segment
RXY	Same as RH
SEGL	Segment length
SINALP	DZ/SEGL
SINBET	DY/RHO
THETA	Polar angle of patch normal vector
T1X,T1Y,T1Z	X,Y, and Z components of $\hat{t}_1$ unit vector for patch
T2X, T2Y, T2Z	X,Y, and Z components of $\hat{t}_2$ unit vector for patch
UL	Segment half-length
XC,YC,ZC	X,Y, and Z coordinates of the center point of the wire segment
XCPA,YCPA,ZCPA	X,Y, and Z coordinates of the center point of the patch segment
XN,YN,ZN	X,Y, Z coordinates of end 1 of wire segment
XNPA,YNPA,ZNPA	X,Y, and Z components of patch normal vector
XP,YP,ZP	X,Y, and Z coordinates of end 2 of wire segment
XU,YU,ZU	X,Y, and Z components of segment half-length

5. I/O VARIABLES:

A.

INPUT	LOCATION
INTARG	/ARGCOM/
IPASS	/ARGCOM/
JBIAS1	/SEGMNT/
JBIAS2	/SEGMNT/
JBIAS3	/SEGMNT/
KBGEOM	/PARTAB/
KBREAL	/PARTAB/
KOLLNK	/PARTAB/
KOLNAM	/PARTAB/
LUTASK	/ADEBUG/
MAXBLK	/SEGMNT/
MAXCSY	/CSYSTM/
MAXDEF	/DEFDAT/
MAXPTS	/PNTTBL/
MAXRAD	/SEGMNT/
MAXSEG	/SEGMNT/
MLTJCT	/SEGMNT/
NAMSEG	/SEGMNT/
NCODES	/PARTAB/
NDATBL	/PARTAB/
NOPCOD	/ADEBUG/
NPRDEF	/DEFDAT/
NPRPT	/PNTTBL/
NPRSEG	/SEGMNT/

	NTINT	/ADEBUG/
	NTSYMB	/ADEBUG/
B.	OUTPUT	LOCATION
	CVAL	/CSYSTM/
	GAREA	/SEGMNT/
	IDCSYS	/CSYSTM/
	IDEFIN	/DEFDAT/
	IDFINS	/DEFDAT/
	IPTBUF	/PNTTBL/
	IPTS	/PNTTBL/
	IPTTBL	/PNTTBL/
	ISGTBL	/SEGMNT/
	NDEBUF	/DEFDAT/
	NOGOFG	/ADEBUG/
	NPATCH	/SEGMNT/
	NPTBUF	/PNTTBL/
	NRAD	/SEGMNT/
	NUMGTD	/GTDDAT/
	NUMPTS	/PNTTBL/
	NUMSEG	/SEGMNT/
	NWIRE	/SEGMNT/
	RAD	/SEGMNT/
	SCALE	/SEGMNT/
	SEGTBL	/SEGMNT/

6. CALLING ROUTINE:

TSKXQT

7. CALLED ROUTINES:

ASSIGN

BUBBLE

CLSFIL

GETARG

GETSEG

GETSYM

JCTION

LNKGTD

LNKJCT

MOVFIL

OPNFIL

PRTGTD

PUTSYM

RDEFIL

STATIN

STATOT

SUBPAT

SYMDEF

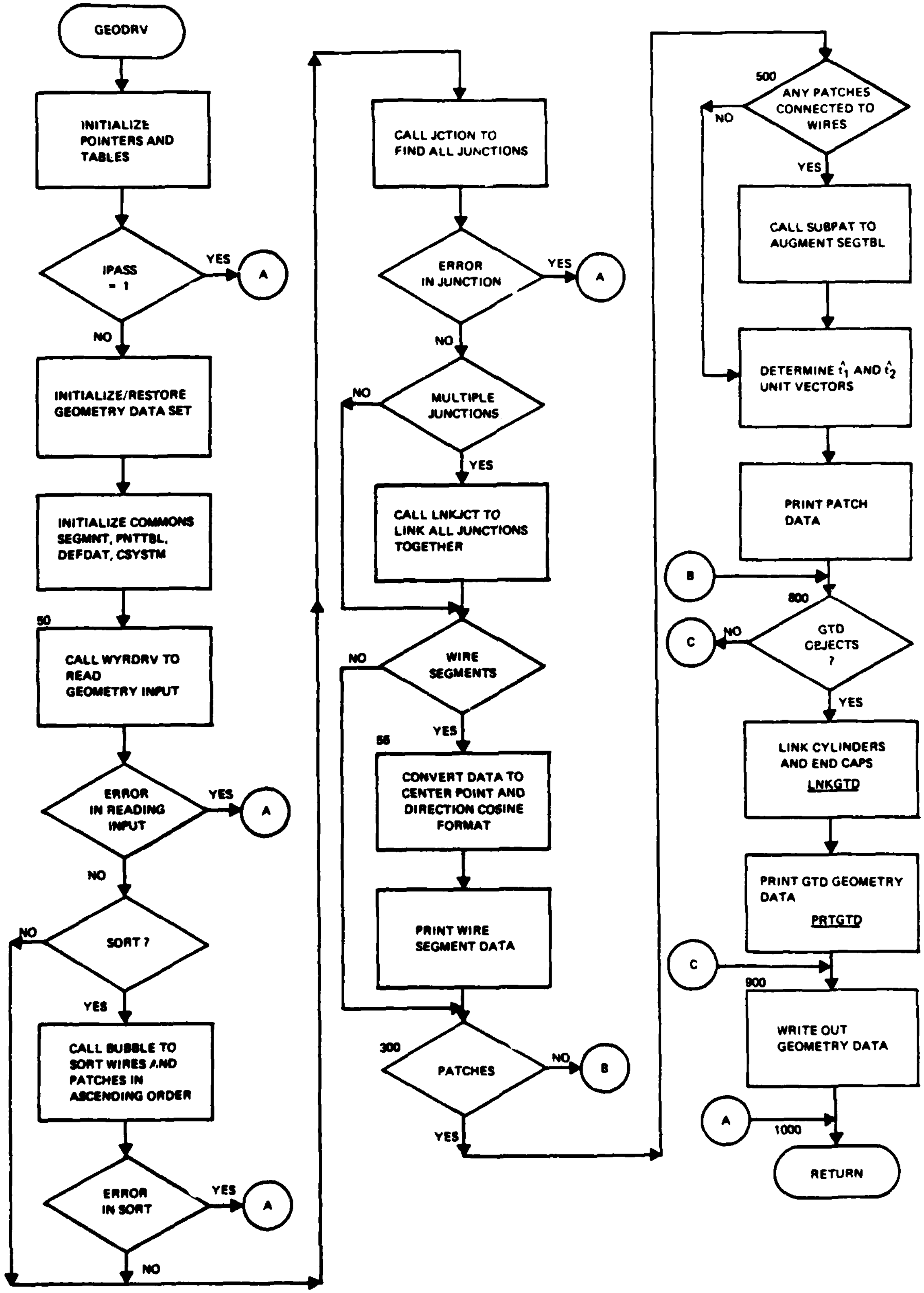
SYMUPD

WLKBCK

WRTFIL

WYRDRV

GEODRV (INPUT)



1. NAME: GEOM (GTD)

2. PURPOSE: This subroutine calculates a large number of constants that are fixed for a given geometry of plates. They are stored in common blocks for use in other routines of this module. GEOM is called whenever the source, frequency or geometry data set changes. Because of the diversity of operations done in GEOM, its description is broken into seven sections:

   (a) Identify edges which are common to more than one plate.

   (b) Compute unit vectors of edge-fixed coordinate systems for each edge on each plate.

   (c) Determine source image information for reflections from plates.

   (d) Calculate possible range for diffraction angle β_0 for each edge.

   (e) Determine wedge angles for plates with common edges.

   (f) Determine plates which are totally shadowed from the source.

   (g) Perform calculations for plates which intersect.

3. METHOD: Sections (a) and (b) are performed only if the frequency or geometry data set name has changed. An error can occur in section (b) if one of the plates is not flat. If this happens an error flag is set and control returns to the calling routine. If an error did not occur, then no matter which (the source, frequency, or data name) changed sections (c) through (g) are performed. Following is a description of the method for each section.

<u>Section (a)</u>: Identifies edges which are common to two plates.

This code compares the edges of one plate to the edges of all the other plates to see if any common corners are found. If two corners are common, it sets their corner locations identically equal and then checks to see if the other corners on that edge were found to be equal. If the two corners on one plate are at the same location as two corners on another plate, thus forming a wedge, the information is stored so it can later be used to calculate wedge angle numbers. Then another plate is taken as the starting plate and the procedure is repeated until all plates have been checked against all the other plates for wedges. Figure 1 shows two plates with a common edge.

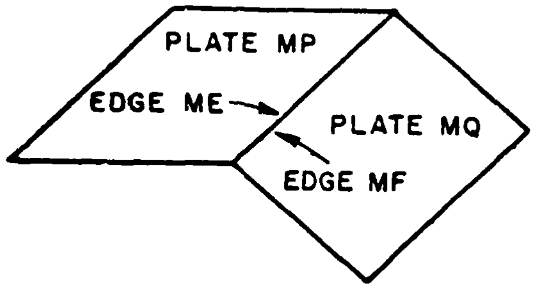
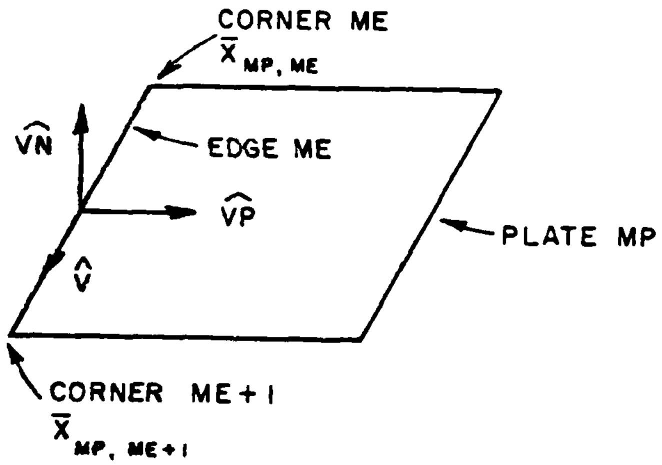


Figure 1. Illustration of Two Plates with a Common Edge

<u>Section (b)</u>: Computes edge-fixed coordinate system unit vectors for each edge.

The edge-fixed coordinate system follows the right-hand rule convention. As your fingers trace the path from corner 1 to corner 2, corner 3, etc. the thumb points in the direction of the normal. Figure 2 shows the essential geometry and associated equation definitions.



$\hat{V}_{MP,ME}$ = edge unit vector = $\hat{x}$ V(MP,ME,1) + $\hat{y}$ V(MP,ME,2) + $\hat{z}$ V(MP,ME,3)

$\widehat{VN}_{MP}$ = plate unit normal = $\hat{x}$ VN(MP,1) + $\hat{y}$ VN(MP,2) + $\hat{z}$ VN(MP,3)

$\widehat{VP}_{MP,ME}$ = edge unit binormal = $\hat{x}$ VP(MP,ME,1) + $\hat{y}$ VP(MP,ME,2) + $\hat{z}$ VP(MP,ME,3)

$\bar{X}_{MP,ME}$ = corner location = $\hat{x}$ X(MP,ME,1) + $\hat{y}$ X(MP,ME,2) + $\hat{z}$ X(MP,ME,3)

Figure 2. Edge-Fixed Coordinate System Unit Vectors

The edge unit vectors are found by,

$$\hat{V}_{MP,ME} = \frac{\bar{X}_{MP,ME+1} - \bar{X}_{MP,ME}}{|\bar{X}_{MP,ME+1} - \bar{X}_{MP,ME}|} .$$

The normals are found using

$$\hat{VN}_{MP} = \frac{\sum_{N=1}^{MEX} \hat{V}_{MP,N} \times \hat{V}_{MP,N+1}}{\left| \sum_{N=1}^{MEX} \hat{V}_{MP,N} \times \hat{V}_{MP,N+1} \right|} ,$$

which is an average over the normals computed by all the edges of the plate. This avoids a possible incorrect normal due to a convex edge geometry. The binormals are found by,

$$\hat{VP}_{MP,ME} = \hat{VN}_{MP} \times \hat{V}_{MP,ME} .$$

<u>Section (c)</u>: Calculates source image information for reflection from plates.

This section calculates the single and double reflection source image locations (see figure 3) by calling subroutine IMAGE. The constant FACTOR is also calculated for sources mounted on a plate. The code has the capability of allowing both electric and magnetic sources. However, version 3 of GEMACS only allows electric sources (IM set to zero at the beginning of this subroutine). Also in this version, the value of FACTOR will always be 1.0. Sources do not lie in the plate plane. The image source coordinate system axes unit vectors for single reflection are calculated by calling IMDIR.

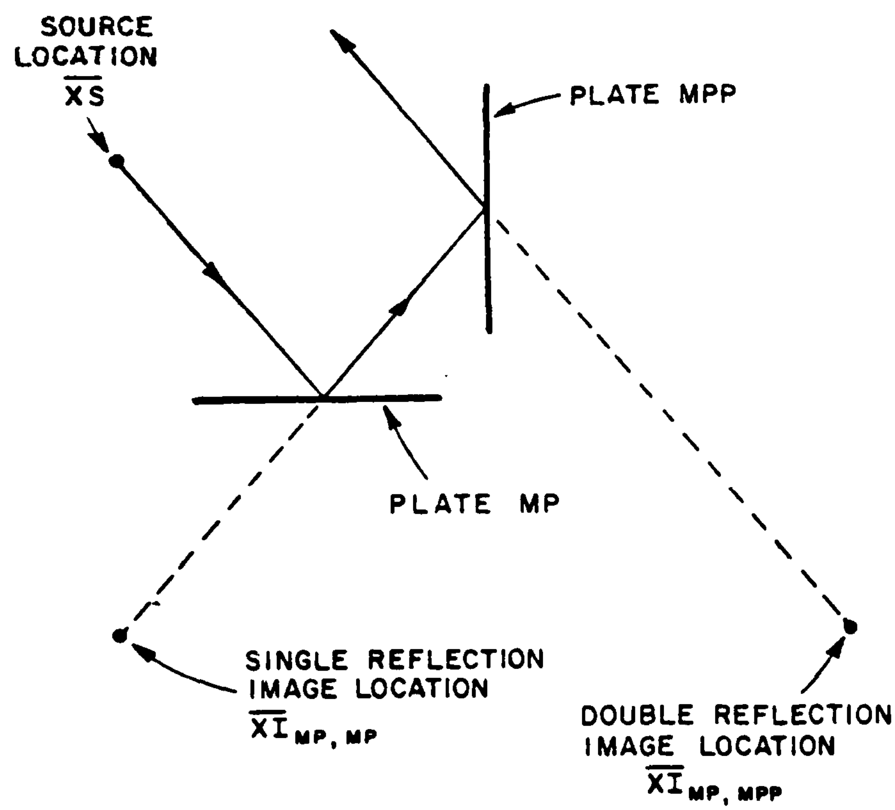


Figure 3. Geometry of Image Locations for a Doubly-Reflected Ray

Section (d): Determines permissible range for angle β_0 for diffraction of the source ray off a plate edge.

The law of diffraction dictates that diffraction from a plate edge is possible when

$$\cos\beta_1 \leq \cos\beta_o \leq \cos\beta_2,$$

where β_0 is the angle that the incident and diffracted rays make with the edge (see figure 4). β_1 and β_2 are diffraction angle limits and are defined in terms of their cosines as:

$$BD(MP,ME,1) = \cos\beta_1 = VI_1 \cdot \hat{V}$$

$$BD(MP,ME,2) = \cos\beta_2 = VI_2 \cdot \hat{V},$$

where

$$\hat{VI}_1 = \frac{\bar{X}_{ME} - \overline{XS}}{|\bar{X}_{ME} - \overline{XS}|}$$

$$\hat{VI}_2 = \frac{\bar{X}_{ME+1} - \overline{XS}}{|\bar{X}_{ME+1} - \overline{XS}|}$$

The code loops through plates and corners to calculate the VI incident unit vectors and then loops through both edges connected to a corner and takes the dot product VI·V for each edge to determine BD.

The vectors mentioned are represented in the code by the same name. These vectors are shown in figure 4.

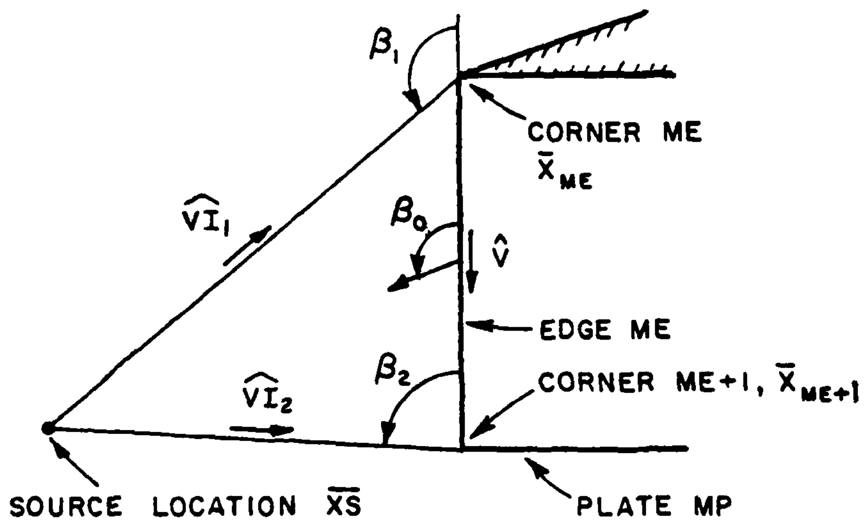


Figure 4. Geometry for Determining Diffraction Angle Range

<u>Section (e)</u>: Calculates wedge angles for plates with common edges.

This section calculates wedge angles for the plates identified in section (a) (by variable FNP) as having common edges. The wedge angle is specified using the wedge angle number FN, such that the wedge angle is given by

$$(2 - FN)\pi$$

as shown in figure 5. The wedge angle number is determined as follows:

$$FN = \frac{1}{\pi} \tan^{-1} \left(\frac{TOP}{BOT}\right)$$

where

$$TOP = \hat{VN}_{MQ} \cdot \hat{VP}_{MP}$$

$$BOT = -\hat{VN}_{MP} \cdot \hat{VN}_{MQ}$$

$$\hat{VN}_{MP} = \hat{x}\ VN(MP,1) + \hat{y}\ VN(MP,2) + \hat{z}\ VN(MP,3)$$

$$\hat{VP}_{MP} = \hat{x}\ VP(MP,ME,1) + \hat{y}\ VP(MP,ME,2) + \hat{z}\ VP(MP,ME,3)$$

$$\hat{VN}_{MQ} = \hat{x}\ VN(MQ,1) + \hat{y}\ VN(MQ,2) + \hat{z}\ VN(MQ,3)$$

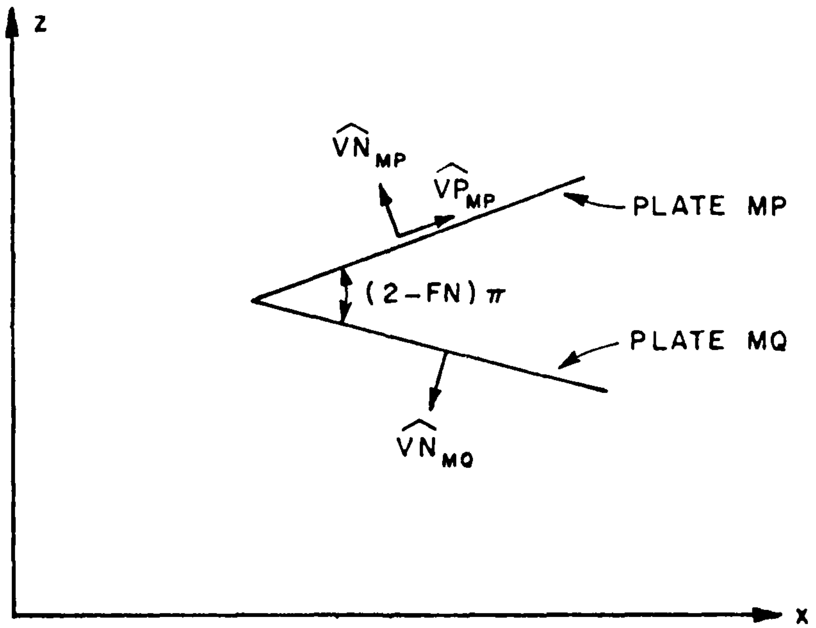


Figure 5. Geometry Used to Determine Wedge Angles of Plates with Common Edges

Section (f): Determines plates which are totally shadowed from the source.

If plate ML (see figures 6a and 6b) totally shadows plate MP from the source, then every ray drawn from the source to a corner of plate MP will intersect plate ML. The routine computes vectors from the source to each corner of plate MP and uses a shadow testing algorithm (in subroutine PLAINT) to check if any plate shadows all of the rays. If so (as in figure 6a), it is assumed that plate MP is totally shadowed from the source. If not (as in figure 6b), it is assumed that plate MP is not totally shadowed from the source (at least by one plate).

Section (g): Handles various calculations for plates which intersect each other.

This section performs four calculations. First it determines if one plate slightly passes through another plate, as shown in figure 7. If this happens, the protruding portion is deleted. Whether an edge intersects a plate or not is determined by calling PLAINT.

The next calculation is to determine the wedge angle number for these intersecting plates. The same method as in section 5 is used.

The third calculation is to find plates with common edges which cannot illuminate each other because their wedge angle is greater than 180°. This is performed by stepping through all the plate combinations and their edges and checking the wedge angle number. When the wedge angle is greater than 180°, those plate edges are identified by setting LIHD equal to .TRUE..

The last part is to determine which plates cannot illuminate each other. When a ray which is reflected or diffracted by one plate propagates in a direction away from the second plate, it cannot illuminate this second plate. These plates are also identified by setting their LIHD variable to .TRUE..

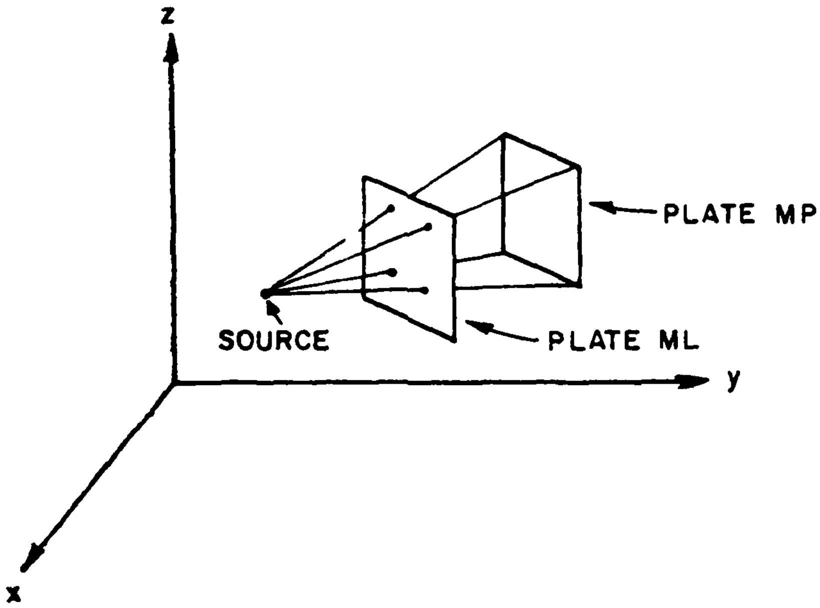


Figure 6a. Configuration Where Plate ML Totally Shadows Plate MP from Source

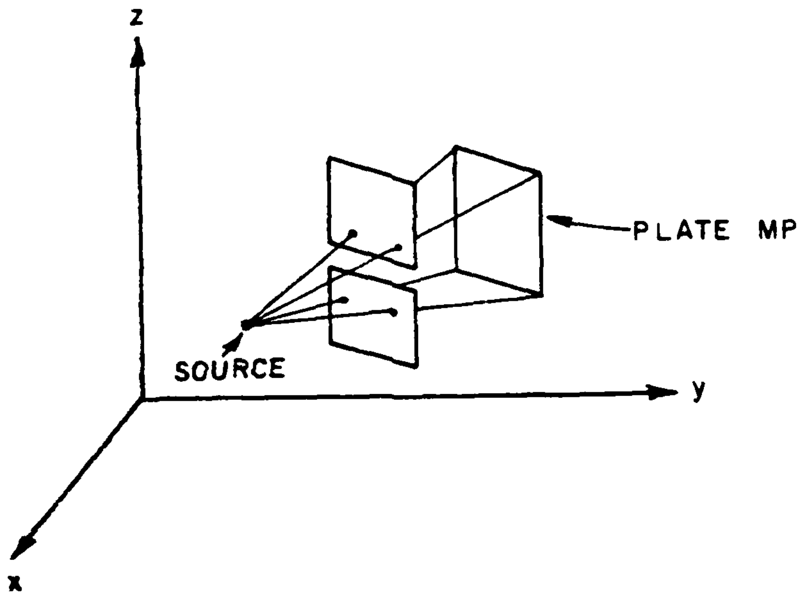


Figure 6b. Configuration Where Plate MP is not Totally Shadowed from the Source

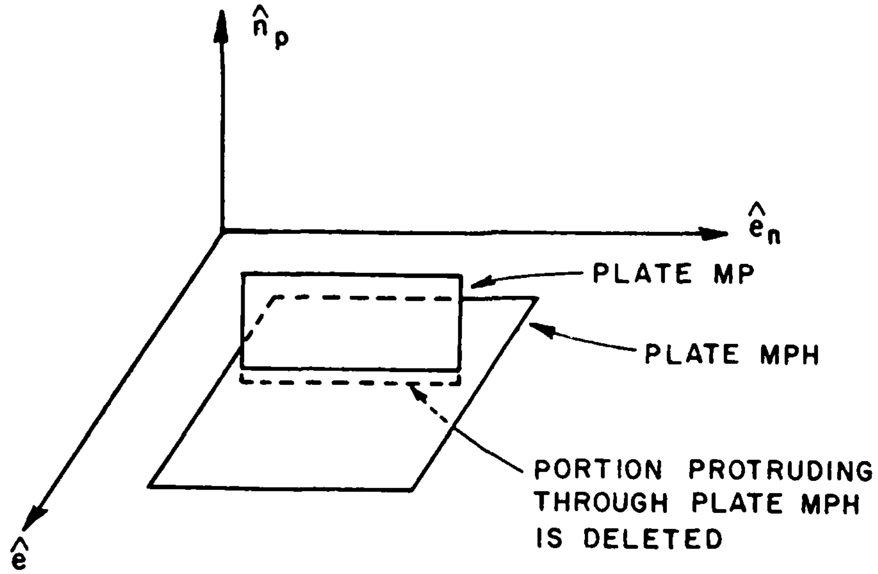


Figure 7. Illustration of a Plate Which Intersects Another

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN	Dot product of plate unit normal and vector from source to the plate (calculated in IMAGE)
ANI	Dot product of vector from corner 1 of plate MPP to the double reflection image location XI(MP,MPP) and the unit normal of plate MPP
ANN	Dot product of XSX and unit normal of plate MP
ANP	Dot product of vector from corner 1 of plate MPP to corner MEE of plate MP and unit normal of plate MPP
	Also dot product of XSX and binormal of edge ME of plate MP
BD	Cosines of angles defining bounds on diffraction angle

BOT	Negative dot product of unit normals of plates MP and MQ
DHIT	Distance from source to nearest hit point (from PLAINT)
DOT	Dot product of plate unit normal and edge unit normal
DS	Unit vector of ray from source to corner ME of plate MP (section (f)). Also unit vector of ray from image to source (section (c)). Also unit vector of edge ME (section (g)).
DSM	Normalization constant for DS
ENORM	Dot product of VN (the unit normal of plate MP) and the z axis of the source coordinate system
FACTOR	Magnitude adjustment for sources mounted on the surface of plates
FANG	Wedge angle
FN	Wedge angle number
FNP	Wedge angle number (also used in defining common edges)
IBSCER	An error flag which if not zero represents that an error has occurred and the program will follow through properly and abort
IFN	Index variable
IHIT	Stores plate numbers for plates intersected by an edge
IM	Set to zero to indicate electric sources only
J	DO loop variable
LCTD	Set true if cylinder shadows plate from source

LFRQFL	Logical variable set true if the frequency is the same as the last frequency used in this routine
LGDNFL	Logical variable set true if the geometry data set name (and therefore the geometry) is the same as the last one used in this routine
LHCT	Set true if ray hits cylinder (returned from CYLINT)
LHIT	Set true if ray intersects a plate (from PLAINT)
LIHD	Set true if plates MP and MPP cannot illuminate each other
LSHD	Set true if plate MP is totally shadowed from the source
LSRCFL	Logical variable set true if the source is the same as the last one used in this routine
LSTD	Set true if plate ML totally shadows plate MP from the source
LSTS	Set true if total shadowing algorithm is being used
LSURF	Logical variable indicating if the source under consideration is mounted on plate MP LSURF(MP) = T for source mounted on plate MP LSURF(MP) = F if source is not mounted on plate MP
LUPRNT	Output file number
MC	Index variable
ME	DO loop variable, also index variable
MEC	Working variable
MEE	Index variable

MEH	Index variable
MEN	Working variable
MEP	Integer array which contains the number of edges on each plate
MES	Computational variable
MEX	Number of edges on a given plate
MF	DO loop variable, also index variable
MFC	Working variable
MFN	Working variable
MFS	Computational variable
MFX	Number of edges on plate MQ
ML	DO loop variable
MME	Index variable
MP	DO loop variable (step through plates), also plate index variable
MPH	Number of the plate which a doubly diffracted ray would hit first
MPINX	Plate index variable
MPP	DO loop variable, also plate index variable
MPPINX	Plate index variable
MPX	Integer which indicates the number of plates in the geometry
MPXR	Maximum number of plates present
MQ	DO loop variable (step through plates), also index variable
MR	Index variable
MV	Index variable

N	DO loop variable
NFN	Working variable
NI	DO loop variable
NJ	DO loop variable
PHCR	Phi component of vector from source to plate corner in RCS
PHWAR	Angle which determines which side of the intersecting plates is illuminated
PI	π
SUM	Length of vector from source to edge ME of plate MP
SUMT	Length of vector from source to closest edge of plate MP
TOP	Dot product of binormal of common edge of plate MP and normal of plate MQ
TPI	2π
V	Matrix of x,y,z components defining edge unit vector in RCS
VAX	X,Y,Z components defining single reflection image source coordinate system axes in RCS components
VI	X,Y,Z components of unit vector of ray from source to corner ME of plate MP
VIM	Normalization constant of VI
VM	Distance between two neighboring corners on a plate, squared
VMAG	Distances between neighboring corners on plates
VN	X,Y,Z components defining plate unit normal directions in RCS components
VNM	Plate unit normal normalization constant

VP	Matrix containing edge unit binormal directions in reference coordinate system (RCS)
VXI	X,Y,Z components defining unit vectors of source image coordinate system axes in RCS
VXS	X,Y,Z components defining unit vectors of source coordinate system axes in RCS
X	Matrix containing all the corner locations for all the plates
XI	Array of components defining single and double reflection source image locations in RCS
XII	X,Y,Z components of single reflection image location calculated in subroutine IMAGE
XIN	X,Y,Z components of location of corner ME of plate MP
	Also, single reflection image location
	Also, XSII
XM	Distance between corner MF on plate MQ and corner ME on plate MP
XS	The x,y,z components of the source location
XSI	X,Y,Z components of source location moved a small amount in the direction of the plate normal (for source on plate plane)
XSII	X,Y,Z components of source image location calculated in subroutine IMAGE for source located at XSI
XSX	X,Y,Z components of vector from corner ME of plate MP to the source
XX	Dot product of binormal of edge ME of plate MP and normal of plate MR
YY	Dot product of normals of plates MP and MR

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
LFRQFL	/SAME/
LGDNFL	/SAME/
LSRCFL	/SAME/
LUPRNT	/ADEBUG/
MEP	/GEOPLA/
MPH	/HITPLT/
MPX	/GEOPLA/
MPXR	/GROUND/
PI	/PIS/
TPI	/PIS/
VXS	/SORINF/
XS	/SORINF/
X	/GEOPLA/

B. OUTPUT

OUTPUT	LOCATION
BD	/BNDFCL/
FACTOR	/SOURSF/
FNP	/FNANG/
LIHD	/LSHDT/
LSHD	/LSHDT/
LSTD	/LSHDP/
LSTS	/LSHDP/
LSURF	/SURFAC/
V	/GEOPLA/

VN	/GEOPLA/
VP	/GEOPLA/
VMAG	/EDMAG/
VXI	/IMAINF/
XI	/IMAINF/

6. CALLING ROUTINE:

GTDDRV

7. CALLED ROUTINES:

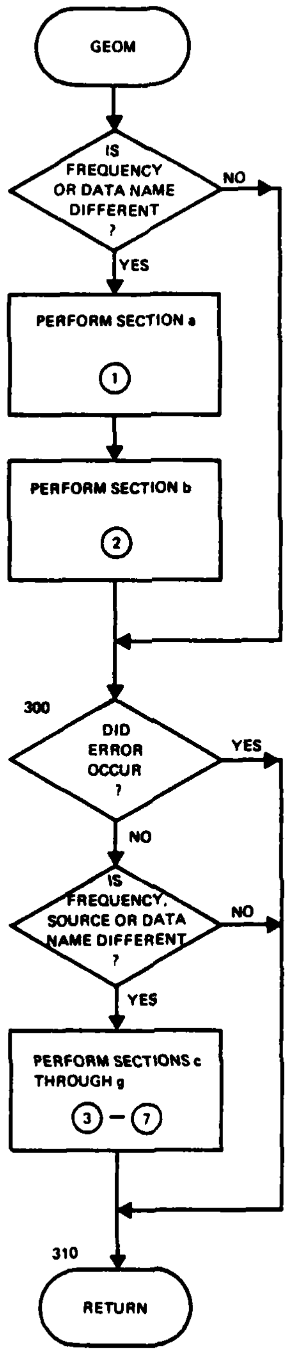
BTAN2

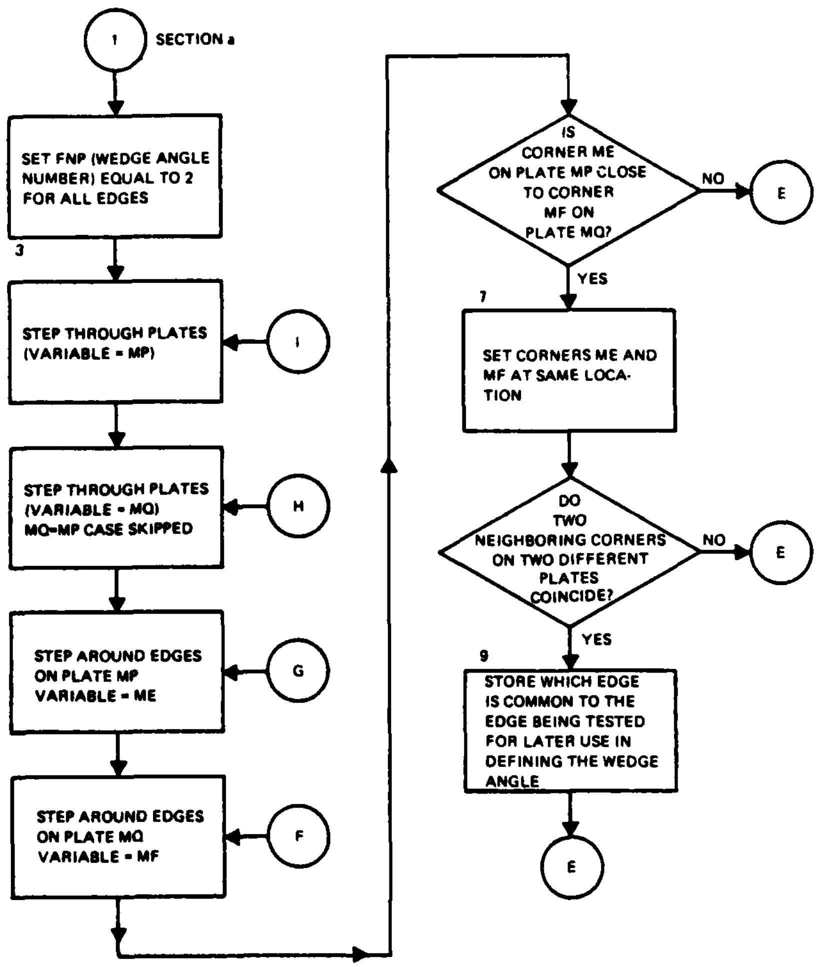
CYLINT

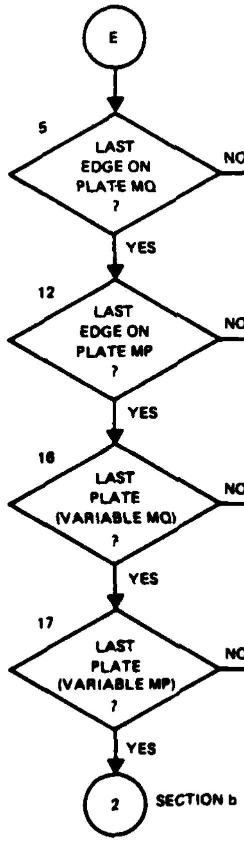
IMAGE

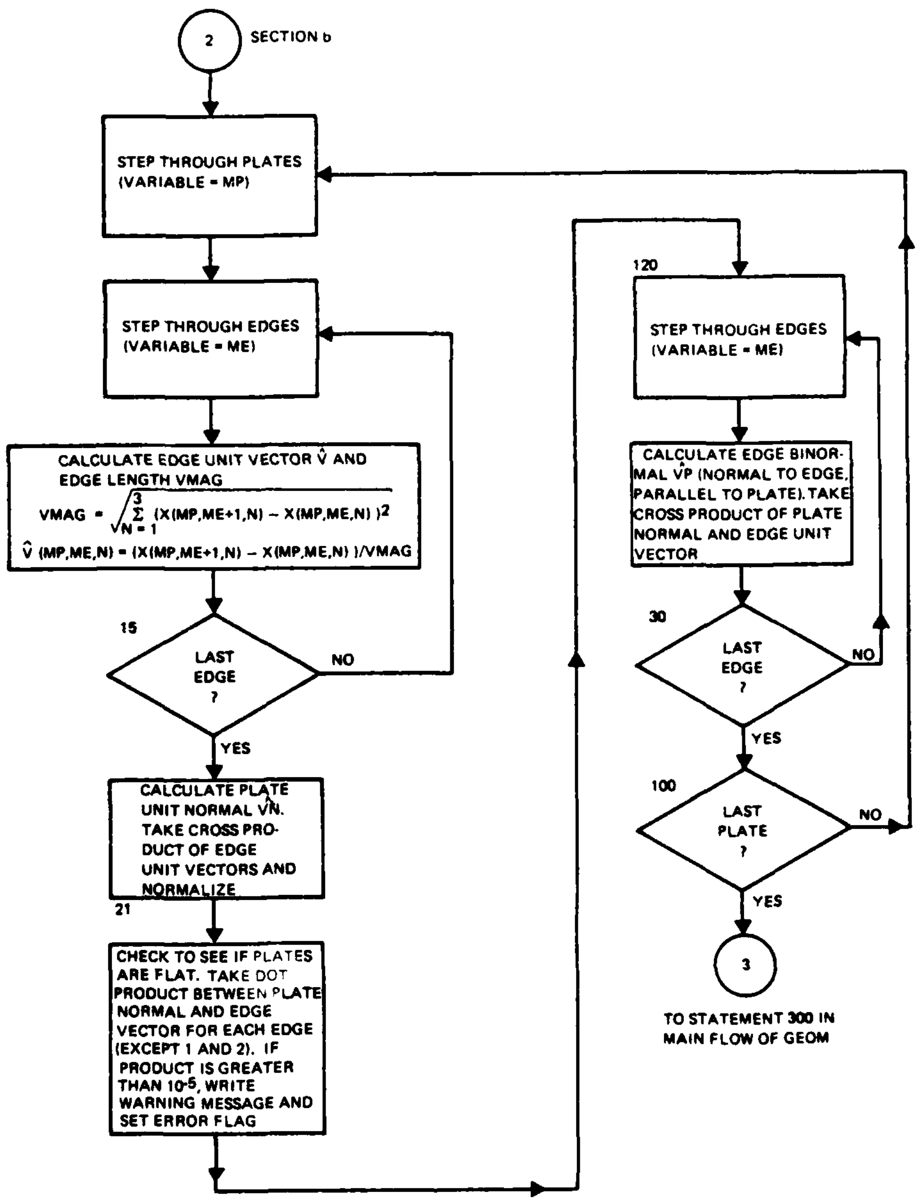
IMDIR

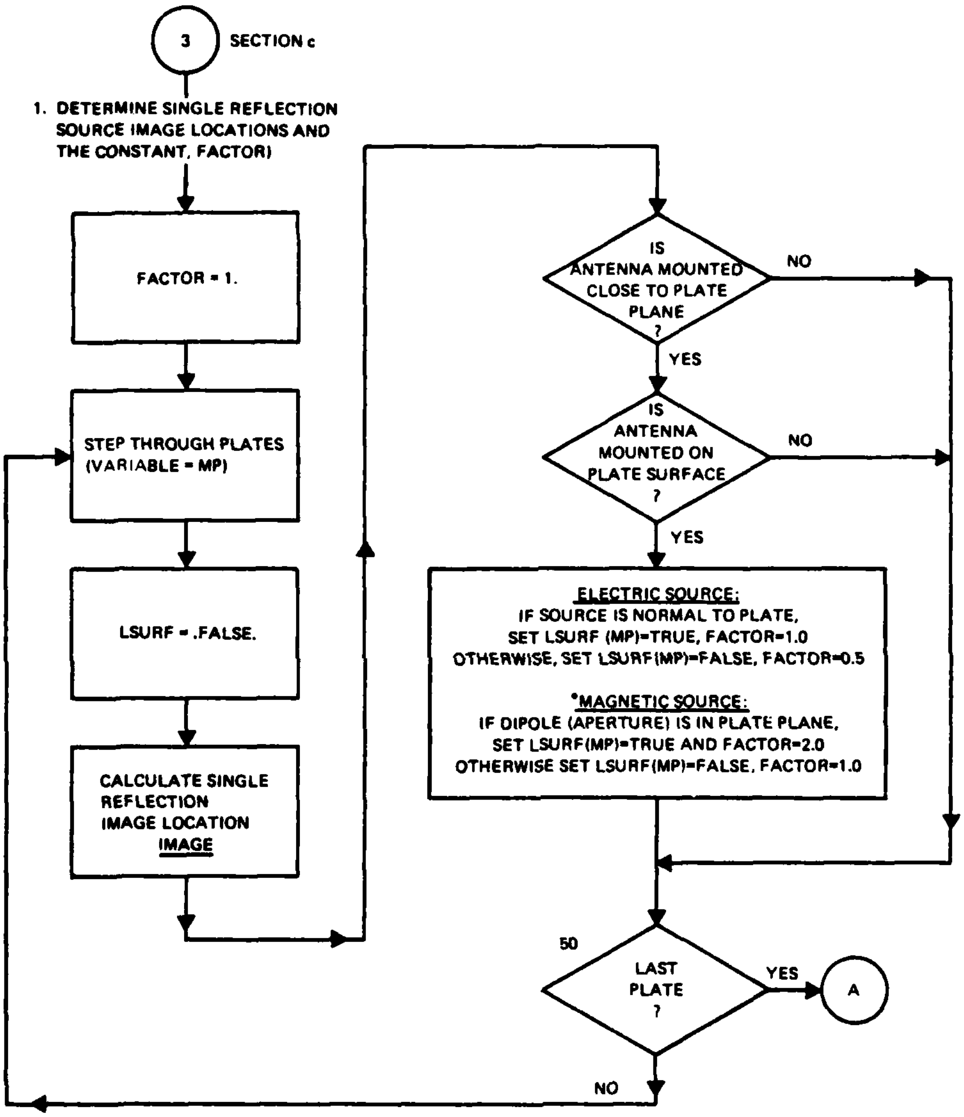
PLAINT


GEOM
IS FREQUENCY OR DATA NAME DIFFERENT ?
NO
YES
PERFORM SECTION a
1
PERFORM SECTION b
2
300
DID ERROR OCCUR ?
YES
NO
IS FREQUENCY, SOURCE OR DATA NAME DIFFERENT ?
NO
YES
PERFORM SECTIONS c THROUGH g
3 – 7
310
RETURN

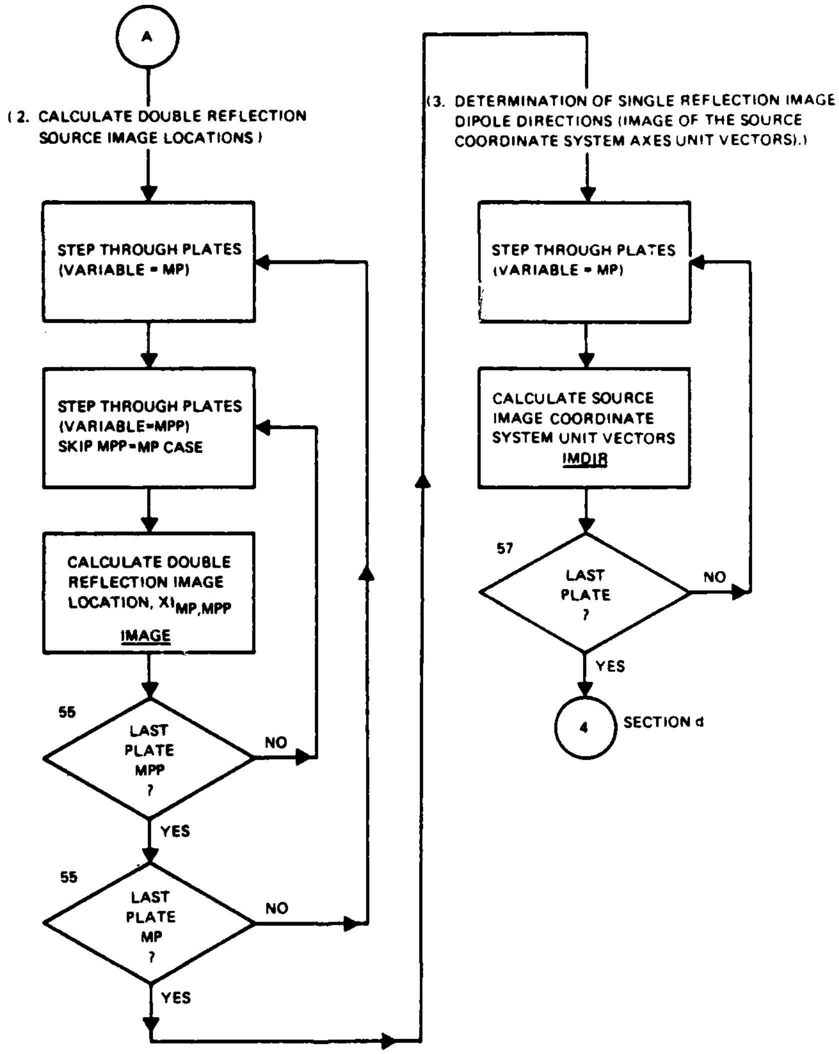

1
SECTION a
SET FNP (WEDGE ANGLE NUMBER) EQUAL TO 2 FOR ALL EDGES
3
STEP THROUGH PLATES (VARIABLE = MP)
I
STEP THROUGH PLATES (VARIABLE = MQ) MQ=MP CASE SKIPPED
H
STEP AROUND EDGES ON PLATE MP VARIABLE = ME
G
STEP AROUND EDGES ON PLATE MQ VARIABLE = MF
F
IS CORNER ME ON PLATE MP CLOSE TO CORNER MF ON PLATE MQ?
NO
E
YES
7
SET CORNERS ME AND MF AT SAME LOCATION
DO TWO NEIGHBORING CORNERS ON TWO DIFFERENT PLATES COINCIDE?
NO
E
YES
9
STORE WHICH EDGE IS COMMON TO THE EDGE BEING TESTED FOR LATER USE IN DEFINING THE WEDGE ANGLE
E

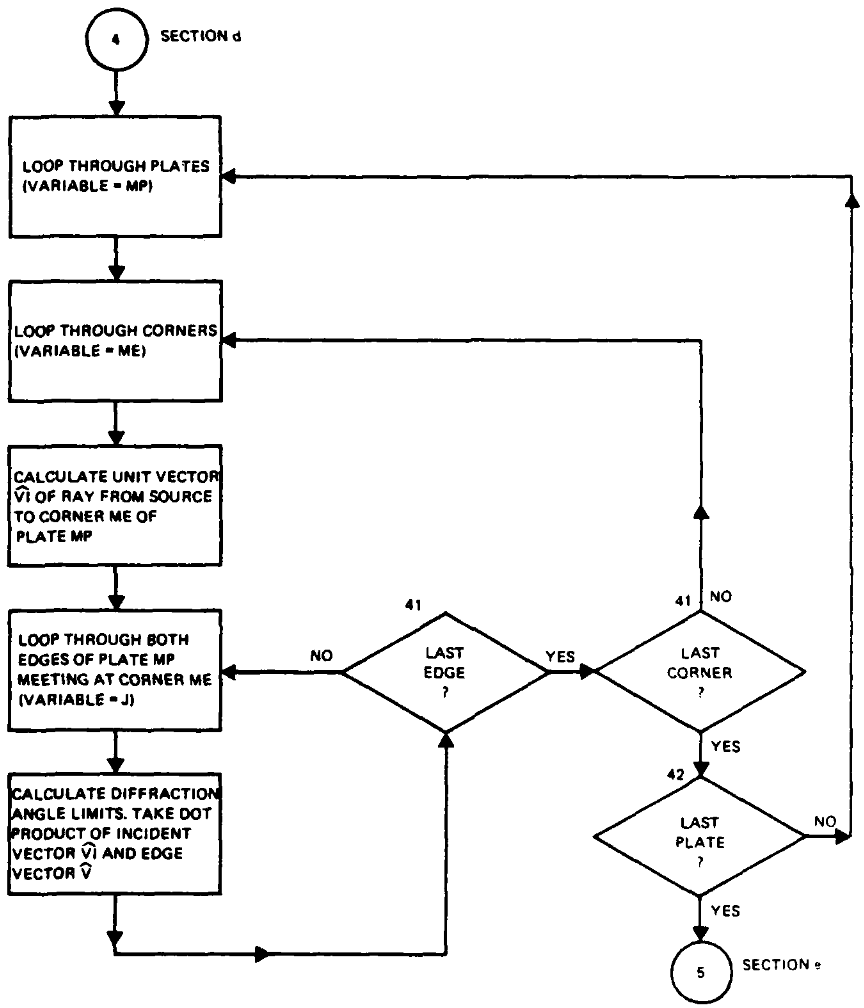

E
5
LAST EDGE ON PLATE MQ ?
NO
F
YES
12
LAST EDGE ON PLATE MP ?
NO
G
YES
16
LAST PLATE (VARIABLE MQ) ?
NO
H
YES
17
LAST PLATE (VARIABLE MP) ?
NO
I
YES
2
SECTION b

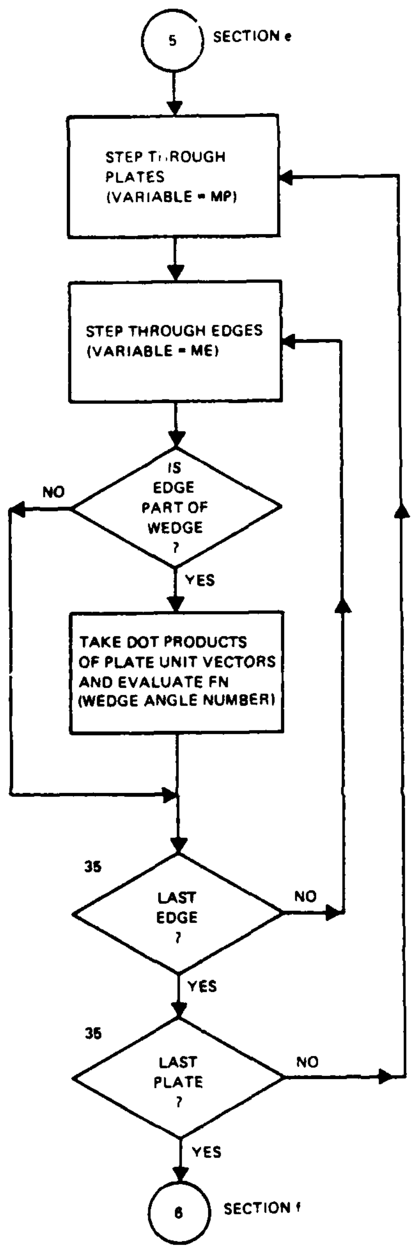

2
SECTION b
STEP THROUGH PLATES (VARIABLE = MP)
STEP THROUGH EDGES (VARIABLE = ME)
CALCULATE EDGE UNIT VECTOR $\hat{V}$ AND EDGE LENGTH VMAG
$VMAG = \sqrt{\sum_{N=1}^{3} (X(MP,ME+1,N) - X(MP,ME,N))^2}$
$\hat{V}(MP,ME,N) = (X(MP,ME+1,N) - X(MP,ME,N))/VMAG$
15
LAST EDGE ?
NO
YES
CALCULATE PLATE UNIT NORMAL $\hat{VN}$. TAKE CROSS PRODUCT OF EDGE UNIT VECTORS AND NORMALIZE
21
CHECK TO SEE IF PLATES ARE FLAT. TAKE DOT PRODUCT BETWEEN PLATE NORMAL AND EDGE VECTOR FOR EACH EDGE (EXCEPT 1 AND 2). IF PRODUCT IS GREATER THAN 10^{-5}, WRITE WARNING MESSAGE AND SET ERROR FLAG
120
STEP THROUGH EDGES (VARIABLE = ME)
CALCULATE EDGE BINORMAL $\hat{VP}$ (NORMAL TO EDGE, PARALLEL TO PLATE). TAKE CROSS PRODUCT OF PLATE NORMAL AND EDGE UNIT VECTOR
30
LAST EDGE ?
NO
YES
100
LAST PLATE ?
NO
YES
3
TO STATEMENT 300 IN MAIN FLOW OF GEOM

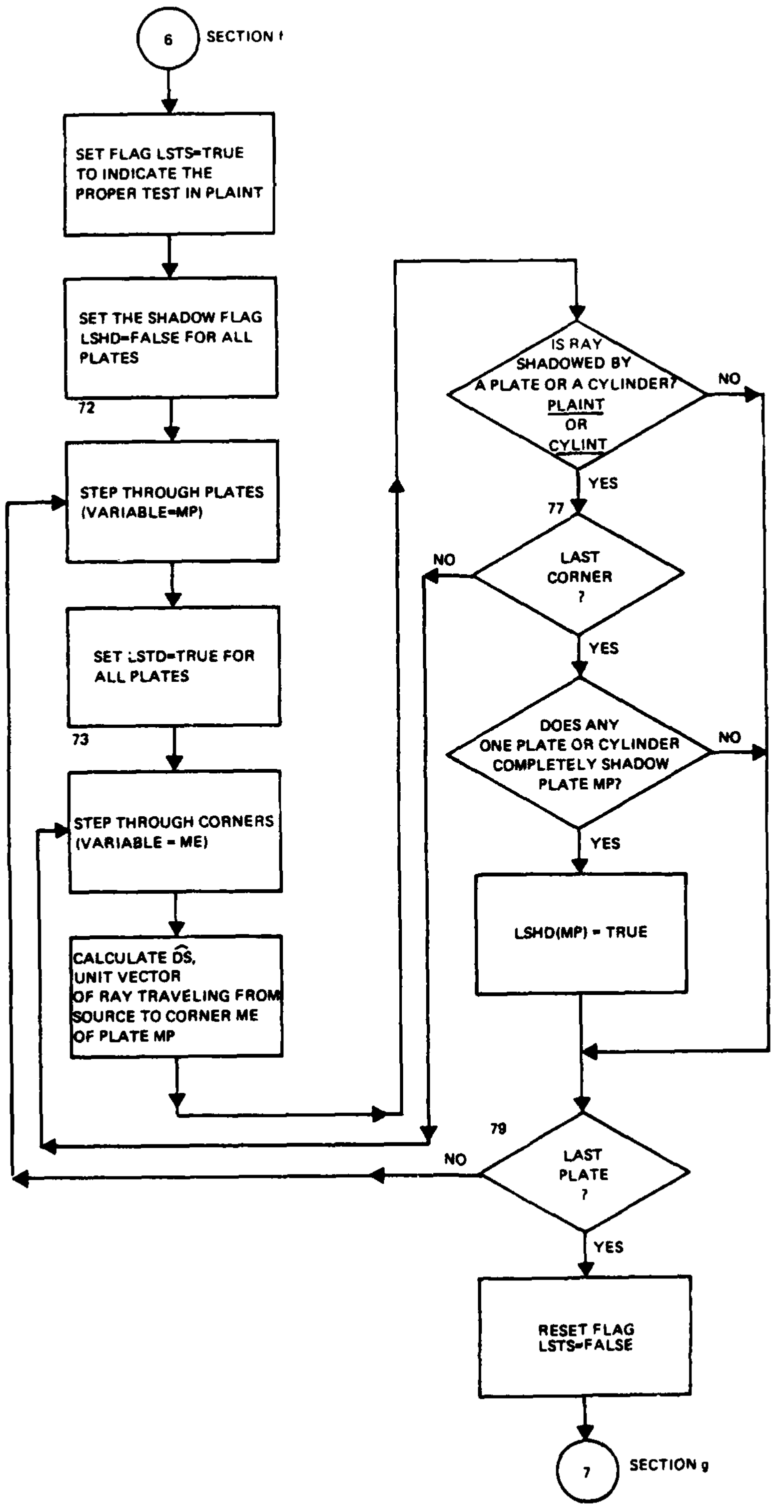


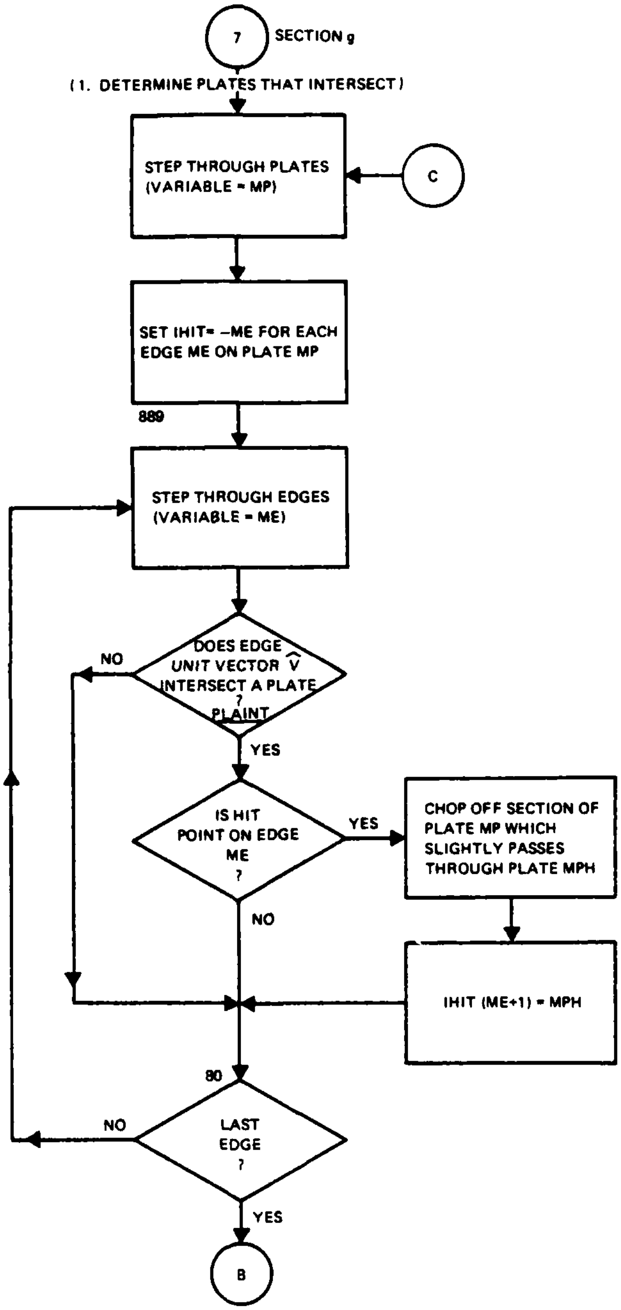
*NOT USED IN VERSION 3 OF GEMACS

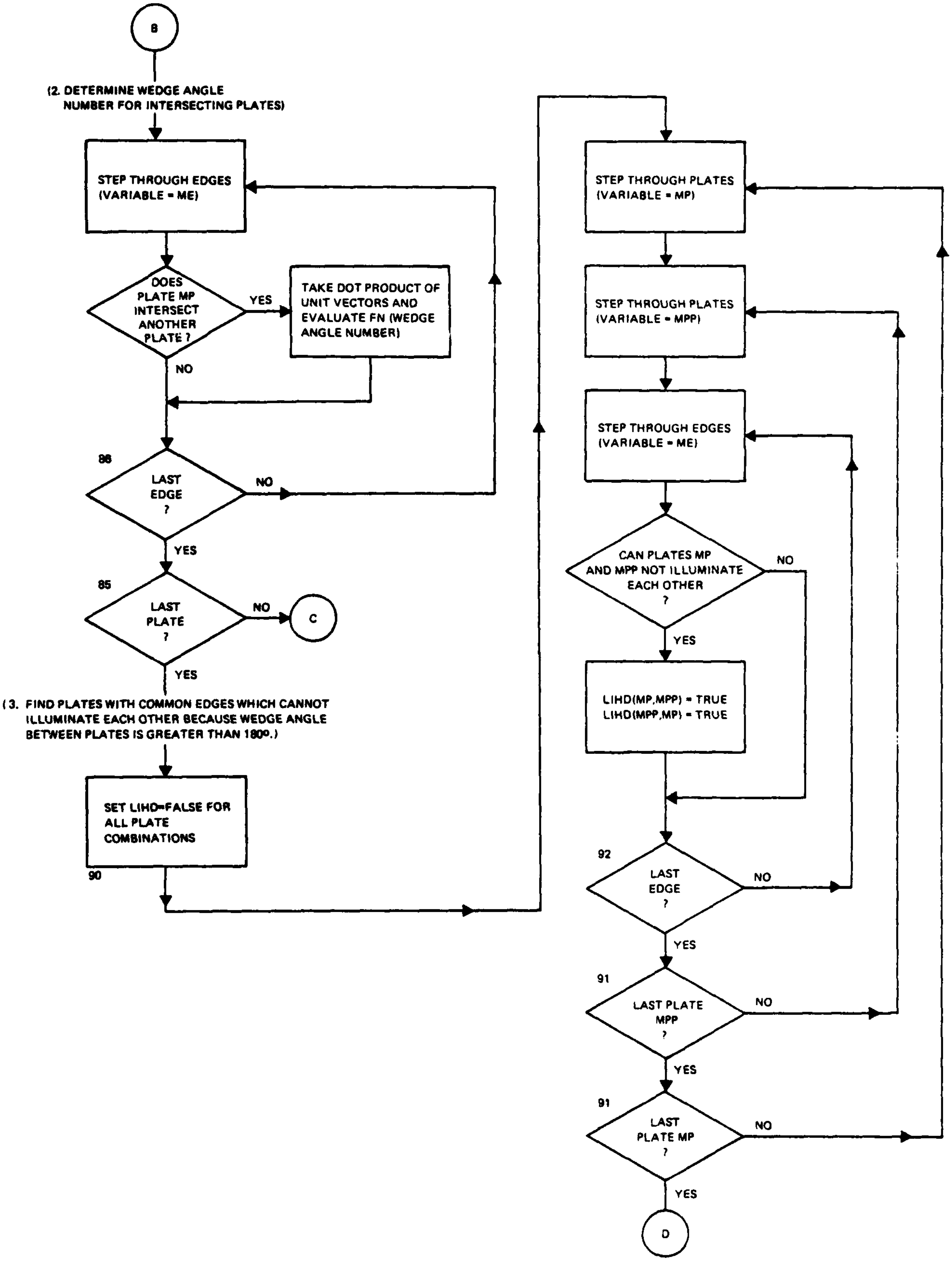

A
(2. CALCULATE DOUBLE REFLECTION SOURCE IMAGE LOCATIONS)
STEP THROUGH PLATES (VARIABLE = MP)
STEP THROUGH PLATES (VARIABLE=MPP) SKIP MPP=MP CASE
CALCULATE DOUBLE REFLECTION IMAGE LOCATION, XI MP,MPP
IMAGE
55
LAST PLATE MPP ?
NO
YES
55
LAST PLATE MP ?
NO
YES
(3. DETERMINATION OF SINGLE REFLECTION IMAGE DIPOLE DIRECTIONS (IMAGE OF THE SOURCE COORDINATE SYSTEM AXES UNIT VECTORS).)
STEP THROUGH PLATES (VARIABLE = MP)
CALCULATE SOURCE IMAGE COORDINATE SYSTEM UNIT VECTORS
IMDIR
57
LAST PLATE ?
NO
YES
4
SECTION d

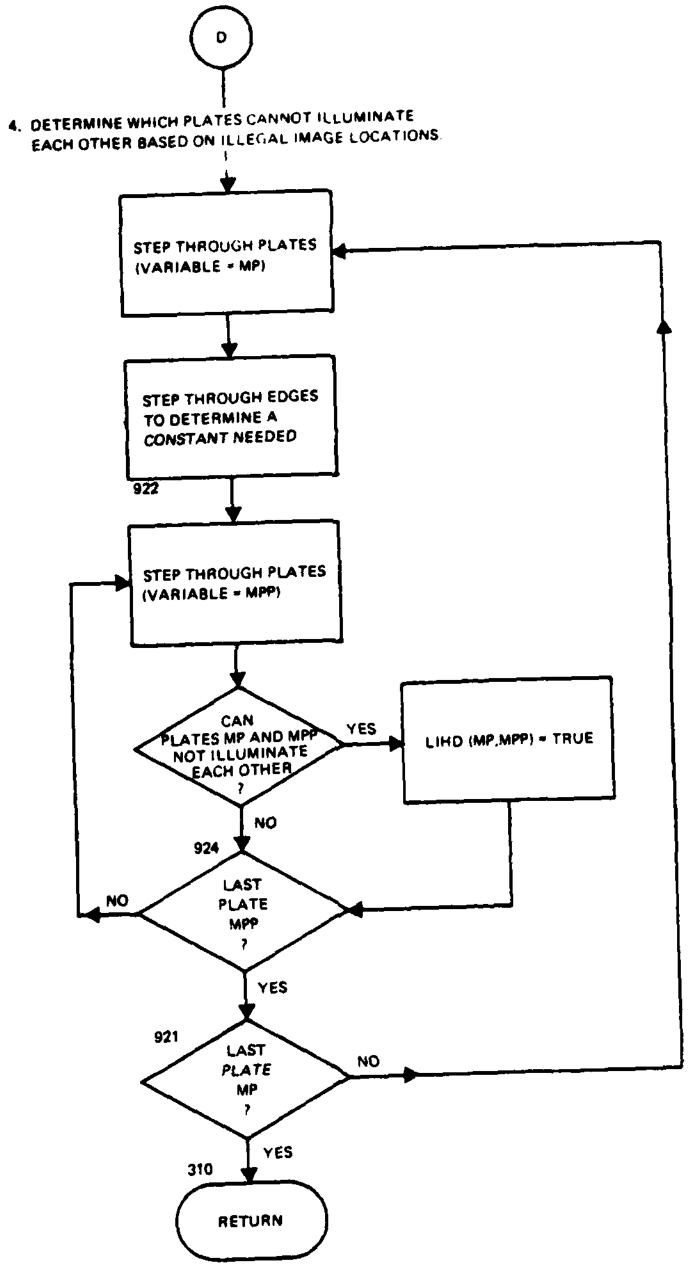

4
SECTION d
LOOP THROUGH PLATES (VARIABLE = MP)
LOOP THROUGH CORNERS (VARIABLE = ME)
CALCULATE UNIT VECTOR V̂I OF RAY FROM SOURCE TO CORNER ME OF PLATE MP
LOOP THROUGH BOTH EDGES OF PLATE MP MEETING AT CORNER ME (VARIABLE = J)
CALCULATE DIFFRACTION ANGLE LIMITS. TAKE DOT PRODUCT OF INCIDENT VECTOR V̂I AND EDGE VECTOR V̂
41
LAST EDGE ?
NO
YES
41
LAST CORNER ?
NO
YES
42
LAST PLATE ?
NO
YES
5
SECTION e


5
SECTION e
STEP THROUGH PLATES (VARIABLE = MP)
STEP THROUGH EDGES (VARIABLE = ME)
IS EDGE PART OF WEDGE ?
NO
YES
TAKE DOT PRODUCTS OF PLATE UNIT VECTORS AND EVALUATE FN (WEDGE ANGLE NUMBER)
35
LAST EDGE ?
NO
YES
35
LAST PLATE ?
NO
YES
6
SECTION f


6
SECTION f
SET FLAG LSTS=TRUE TO INDICATE THE PROPER TEST IN PLAINT
SET THE SHADOW FLAG LSHD=FALSE FOR ALL PLATES
72
STEP THROUGH PLATES (VARIABLE=MP)
SET LSTD=TRUE FOR ALL PLATES
73
STEP THROUGH CORNERS (VARIABLE = ME)
CALCULATE DS, UNIT VECTOR OF RAY TRAVELING FROM SOURCE TO CORNER ME OF PLATE MP
IS RAY SHADOWED BY A PLATE OR A CYLINDER? PLAINT OR CYLINT
NO
YES
77
LAST CORNER ?
NO
YES
DOES ANY ONE PLATE OR CYLINDER COMPLETELY SHADOW PLATE MP?
NO
YES
LSHD(MP) = TRUE
79
LAST PLATE ?
NO
YES
RESET FLAG LSTS=FALSE
7
SECTION g


7
SECTION g
(1. DETERMINE PLATES THAT INTERSECT)
STEP THROUGH PLATES (VARIABLE = MP)
C
SET IHIT= −ME FOR EACH EDGE ME ON PLATE MP
889
STEP THROUGH EDGES (VARIABLE = ME)
DOES EDGE UNIT VECTOR V̂ INTERSECT A PLATE PLAINT ?
NO
YES
IS HIT POINT ON EDGE ME ?
YES
CHOP OFF SECTION OF PLATE MP WHICH SLIGHTLY PASSES THROUGH PLATE MPH
NO
IHIT (ME+1) = MPH
80
LAST EDGE ?
NO
YES
B


B
(2. DETERMINE WEDGE ANGLE NUMBER FOR INTERSECTING PLATES)
STEP THROUGH EDGES (VARIABLE = ME)
DOES PLATE MP INTERSECT ANOTHER PLATE ?
YES
TAKE DOT PRODUCT OF UNIT VECTORS AND EVALUATE FN (WEDGE ANGLE NUMBER)
NO
86
LAST EDGE ?
NO
YES
85
LAST PLATE ?
NO
C
YES
(3. FIND PLATES WITH COMMON EDGES WHICH CANNOT ILLUMINATE EACH OTHER BECAUSE WEDGE ANGLE BETWEEN PLATES IS GREATER THAN 180°.)
SET LIHD=FALSE FOR ALL PLATE COMBINATIONS
90
STEP THROUGH PLATES (VARIABLE = MP)
STEP THROUGH PLATES (VARIABLE = MPP)
STEP THROUGH EDGES (VARIABLE = ME)
CAN PLATES MP AND MPP NOT ILLUMINATE EACH OTHER ?
NO
YES
LIHD(MP,MPP) = TRUE
LIHD(MPP,MP) = TRUE
92
LAST EDGE ?
NO
YES
91
LAST PLATE MPP ?
NO
YES
91
LAST PLATE MP ?
NO
YES
D


D
4. DETERMINE WHICH PLATES CANNOT ILLUMINATE EACH OTHER BASED ON ILLEGAL IMAGE LOCATIONS.
STEP THROUGH PLATES (VARIABLE = MP)
STEP THROUGH EDGES TO DETERMINE A CONSTANT NEEDED
922
STEP THROUGH PLATES (VARIABLE = MPP)
CAN PLATES MP AND MPP NOT ILLUMINATE EACH OTHER ?
YES
LIHD (MP,MPP) = TRUE
NO
924
LAST PLATE MPP ?
NO
YES
921
LAST PLATE MP ?
NO
YES
310
RETURN

NAME: GEOMC (GTD)

PURPOSE: To calculate geometry of the fixed cylinder structures.

METHOD: Four major calculations are performed in GEOMC. The first is to calculate the source image location for reflection from an end cap. The image source location is given by:

$$\overline{XIC} = \overline{XS} - 2 \cdot AN \widehat{VNC},$$

where

$$AN = (\overline{XS} - \overline{XC}) \cdot \widehat{VNC}$$

and $\overline{XS}$ is the vector to the source location, $\overline{XC}$ is the vector to the intersection point of the z axis and the end cap, and $\widehat{VNC}$ is the unit normal vector of the end cap. These quantities are shown in figure 1.

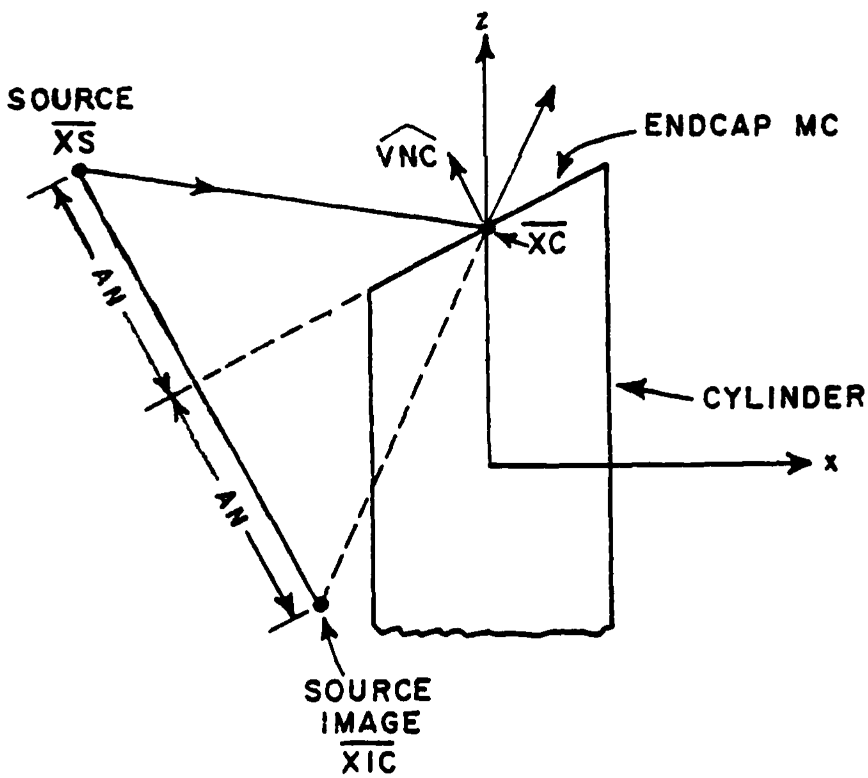


Figure 1. Geometry for Determining Source Image Location for Reflection from Cylinder End Cap

The second step is to calculate the source field factor, FACTOR, which is the coefficient of the source field used to obtain the correct field magnitude for sources mounted on plates or the end caps in order to compensate for image effects. This quantity can be calculated for either electric or magnetic sources. However version 3 of GEMACS only analyzes electric sources, so the value IM, which indicates in this subroutine an electric or magnetic source is set to zero (for an electric source). Subroutine CAPINT is called to determine if the source is mounted on the end cap.

The third major calculation is to compute the end cap image source axes directions. This is done by calling IMCDIR.

The above three calculations are performed for the more positive end cap and then for the more negative end cap. This is done in DO loop 515.

The last calculation is to compute the two tangent vectors from the source to the cylinder sides as shown in figure 2. These vectors are the boundaries within which the source ray must propagate to hit the cylinder. This information is stored in common block BNDSCL and used in other routines later. Subroutine TANG is called to compute the tangents.

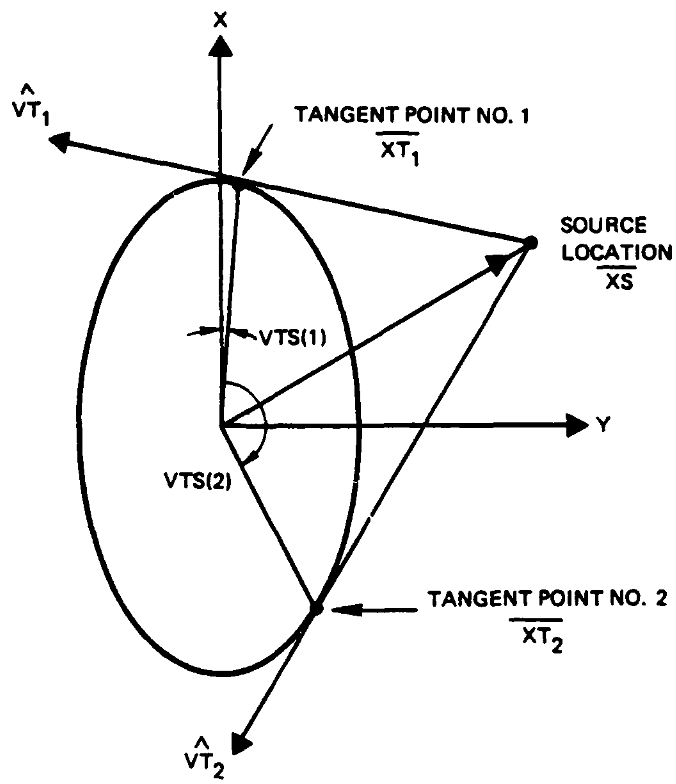


Figure 2. Illustration of Vectors from the Source Tangent to the Elliptic Cylinder

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN	Dot product of end cap normal and ray from end cap to source
BTS	Array which defines the unit vectors of the two source rays tangent to the cylinder such that the unit vector for the source ray tangent to tangent point 1 is given by: $\hat{T1} = \hat{x}\ BTS(1) + \hat{y}\ BTS(2)$ and the unit vector for the source ray tangent to tangent point 2 is given by: $\hat{T2} = \hat{x}\ BTS(3) + \hat{y}\ BTS(4)$
CNC	Cosine of the angle between the z axis and end cap plane where CNC(1) refers to the more positive end cap and CNC(2) refers to the more negative end cap (angle measured in x-z plane)
DHIT	Distance to hit point from subroutine CAPINT (never used)
DS	Unit vector of ray from source image to source
DSM	Distance from source image to source
DTS	Dot product of the two vectors from the source tangent to the cylinder
ENORM	Dot product of end cap normal and z axis of source coordinate system
FACTOR	Coefficient of the source field used to obtain the correct field magnitude for sources mounted on plates or end caps in order to compensate for image effects
IM	Set to zero at beginning of subroutine to indicate electric sources only are used
LHIT	Set true if ray hits end cap (from subroutine CAPINT)

LPLA	Logical variable set true to indicate at least one plate exists in the geometry
LSRFC	Set true if source is mounted on end cap MC
MC	End cap index variable
N	DO loop variable
NC	Sign change variable
NI	DO loop variable
NJ	DO loop variable
SNC	Sine of the angle between the z axis and end cap plane where SNC(1) refers to the more positive end cap and SNC(2) refers to the more negative end cap (angle measured in x-z plane)
VAX	End cap source image axes directions
VNC	X,Y,Z components of the end cap unit normal in reference coordinate system
VTS	Array which contains the two elliptical angles defining the tangent points on the cylinder from the source such that the tangent point N is given by: x = A*cos (VTS(N)) y = B*cos (VTS(N))
VXIC	X,Y,Z components of unit vectors defining axes of end cap source image coordinate system
VXS	A 3x3 matrix defining the source coordinate system axes unit vectors in the RCS
XIC	The x,y,z components of the source image location for single reflections from each end cap in RCS
XIN	Source image location in end cap MC
XS	The x,y,z components of the source location in the reference coordinate system

ZC	Point where end cap intersects z axis of reference coordinate system, where ZC(1) refers to the more positive end cap and ZC(2) refers to the more negative end cap

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
CNC	/GEOMEL/
LPLA	/LPLCY/
SNC	/GEOMEL/
VXS	/SORINF/
XS	/SORINF/
ZC	/GEOMEL/

B. OUTPUT

OUTPUT	LOCATION
BTS	/BNDSCL/
DTS	/BNDSCL/
FACTOR	/SOURSF/
LSRFC	/SRFACC/
VTS	/BNDSCL/
XIC	/IMCINF/

6. CALLING ROUTINE:

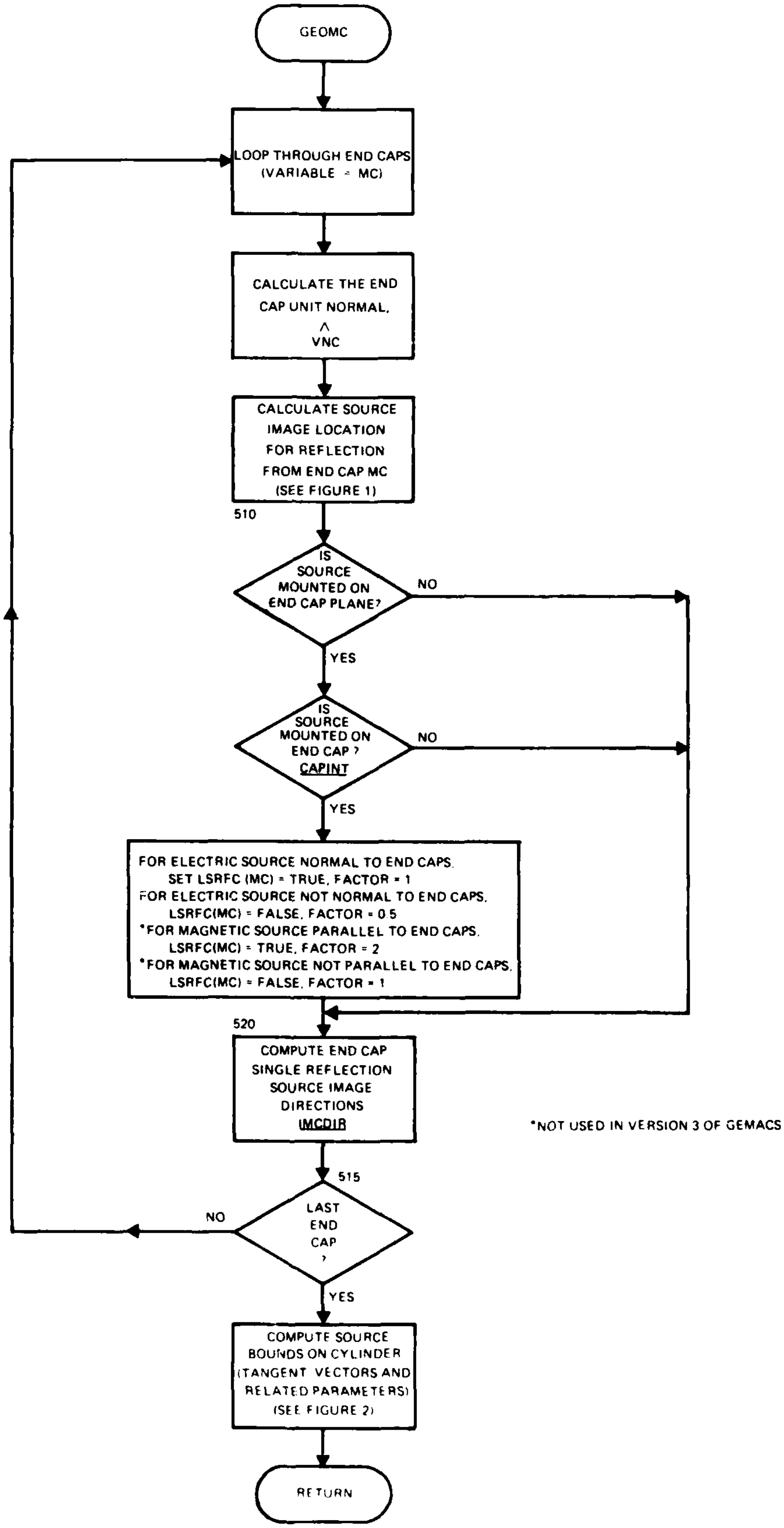
GTDDRV

7. CALLED ROUTINES:

CAPINT

IMCDIR

TANG


GEOMC
LOOP THROUGH END CAPS (VARIABLE = MC)
CALCULATE THE END CAP UNIT NORMAL, VNC
CALCULATE SOURCE IMAGE LOCATION FOR REFLECTION FROM END CAP MC (SEE FIGURE 1)
510
IS SOURCE MOUNTED ON END CAP PLANE?
NO
YES
IS SOURCE MOUNTED ON END CAP ? CAPINT
NO
YES
FOR ELECTRIC SOURCE NORMAL TO END CAPS, SET LSRFC (MC) = TRUE, FACTOR = 1
FOR ELECTRIC SOURCE NOT NORMAL TO END CAPS, LSRFC(MC) = FALSE, FACTOR = 0.5
*FOR MAGNETIC SOURCE PARALLEL TO END CAPS, LSRFC(MC) = TRUE, FACTOR = 2
*FOR MAGNETIC SOURCE NOT PARALLEL TO END CAPS, LSRFC(MC) = FALSE, FACTOR = 1
520
COMPUTE END CAP SINGLE REFLECTION SOURCE IMAGE DIRECTIONS IMCDIR
*NOT USED IN VERSION 3 OF GEMACS
515
LAST END CAP ?
NO
YES
COMPUTE SOURCE BOUNDS ON CYLINDER (TANGENT VECTORS AND RELATED PARAMETERS) (SEE FIGURE 2)
RETURN

1. NAME: GEOMPC (GTD)

2. PURPOSE: To compute variables pertaining to plate-cylinder interactions which are constant for a given set of plates and cylinder, a given source and a given frequency.

3. METHOD: This routine has four main sections:

   (a) Determine corners and edges which are attached to the cylinder.

   (b) Determine image bounds on cylinder for reflection from plates.

   (c) Determine permissible ranges for cylinder reflected, plate diffracted rays.

   (d) Determine permissible ranges for plate diffracted, cylinder reflected rays.

Section (a) only needs to be performed if the frequency or geometry data set name (thus the geometry) is different from the last time the routine was called, since the source is not associated with the calculations. The other three sections must be performed if the source, frequency, or data set name changed.

<u>Section (a)</u>: Determine corners and edges which are attached to the cylinder.

The code steps through all the corners on each plate. If a corner is on the cylinder curved surface, the indicator LDC for that particular plate and corner is set true. If an edge is attached to the cylinder, the wedge angle number FNP is set equal to -1 to flag this. See figure 1 for an illustration.

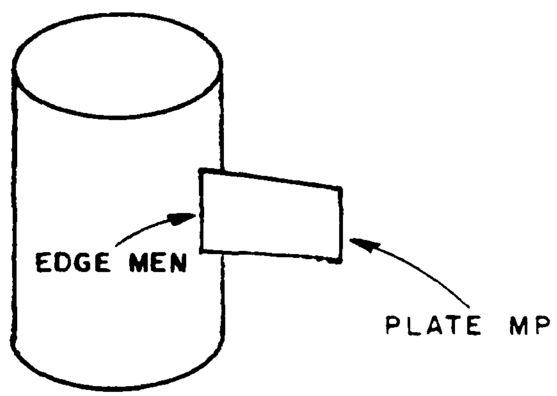


Figure 1. Illustration of Plate Attached to Cylinder

Section (b): Determine image bounds on cylinder for reflection from plates.

This is performed by determining the rays tangent to the cylinder sides from the source image through each plate (see figure 2). Subroutine TANG is called to determine the tangents and related geometry data which will be used later in other routines. These quantities are stored in common block BNDICL.

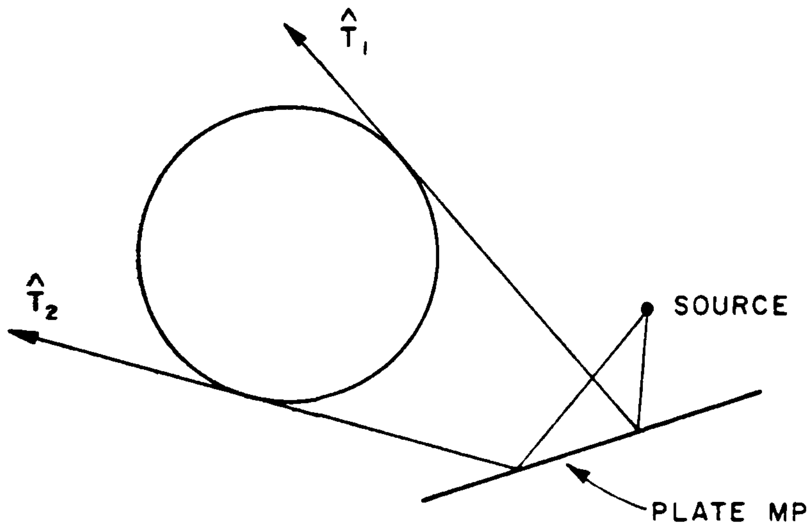


Figure 2. Illustration of Source Rays Reflected by Plate MP and Tangent to the Cylinder

Section (c): Determine permissible ranges for cylinder reflected, plate edge diffracted rays.

This task is accomplished by stepping through each edge on each plate and calling RFDFIN to determine the ray path for a reflected ray from the cylinder incident on the edge corners. Details for this method are given on pages 149-154 of reference A. Also see the description for subroutine RFDFIN. After the ray vectors and the angles they make with the edge (see figure 3) are known, the diffraction limits, which are the cosines of the angles, can be computed. This information is stored in common block BNDRCL.

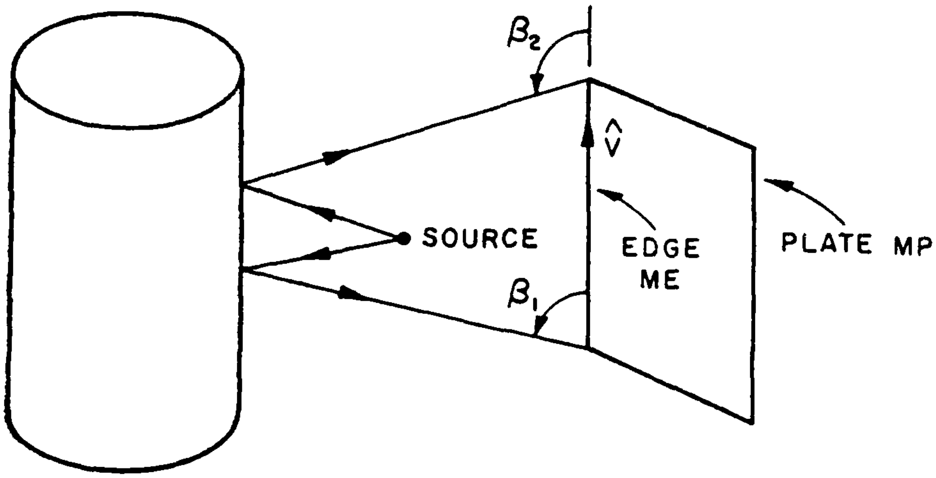


Figure 3. Illustration of Bounds for Cylinder Reflected, Plate Diffracted Region

The last part of this section computes a branch cut displacement angle for each edge on each plate. This angle is stored in common block BRNPHW. It is used later in the plate-cylinder reflected-diffracted, diffracted-reflected field calculations. It is used to avoid the 2π-to-0 transition in the phi angle of the x-y plane. The angle places the branch out behind the cylinder with respect to the plate.

Section (d): Determine permissible ranges for plate diffracted, cylinder reflected rays.

The geometry required for this section is shown in figure 4. Each edge on each plate is checked to see if it is a valid diffraction edge for the given source location. This is determined by the wedge angle number, which indicates if the source can illuminate the edge (or wedge) on the side closest to the cylinder. If diffraction is possible, LDC for this particular plate and edge is set true for checks later on in this routine. The path of the ray diffracted by a plate and reflected off the cylinder's more positive end cap is determined by choosing a starting reflection point on the cylinder's more positive end cap and calling subroutine DPTNFW to determine the plate diffraction point. Now that the vector direction between the diffraction and reflection points is known, the reflection angle bound can be calculated. This procedure is repeated using the more negative end cap. The favored starting location for the reflection and diffraction point is determined by the greater dot product. The dot products come from the two rays generated from the diffraction point to the cylinder sides on the same plane (perpendicular to the cylinder). The starting point parameters, the dot product and vector from the favored diffraction point tangent to the cylinder (2-D) are stored for use in subroutine DPLRCL. Details of the method used for the ray tracing in section (d) are given on pages 161-163 of reference A. Also see the description for subroutine DFRFPT.

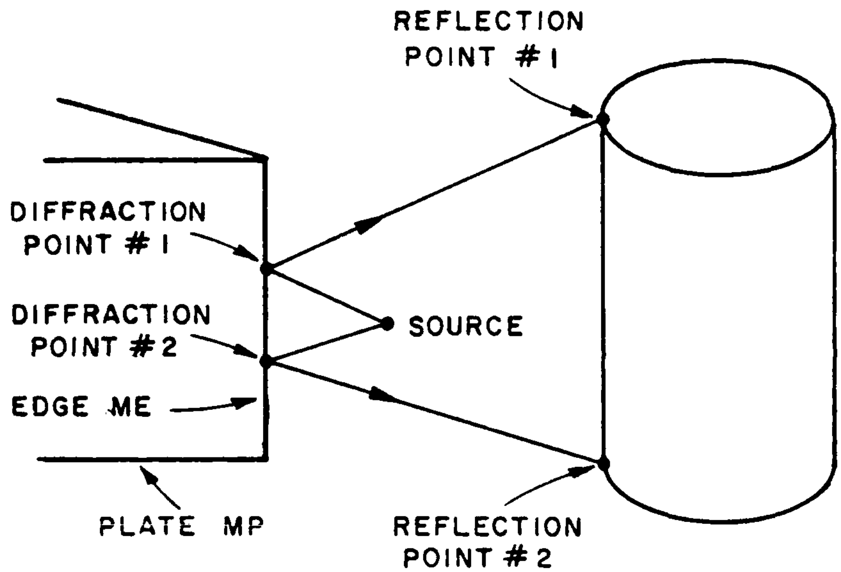


Figure 4. Illustration of Starting Point Path for Plate Diffracted, Cylinder Reflected Ray Tracing Algorithm

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	Radius of cylinder along x axis in wavelengths
B	Radius of cylinder along y axis in wavelengths

BCD	Diffraction limits for ray reflected by the cylinder and diffracted from plate
BTCN	X,Y components of unit vectors for rays tangent to the cylinder from diffraction point on plate edge (for more negative starting point on cylinder)
BTCP	X,Y components of unit vectors for rays tangent to the cylinder from diffraction point on plate edge (for more positive starting point on cylinder) (also see BTI)
BTDC	X,Y components of unit vectors for rays tangent to the cylinder from diffraction point on plate edge (for favored starting point on cylinder)
BTI	X,Y components of source image vectors tangent to the cylinder
CTC	Cotangent of the angle between the z axis and the end cap plane, where CTC(1) refers to the more positive end cap and CTC(2) refers to the more negative end cap (angle measured in x-z plane)
DDC	Matrix which contains the cosine of the starting reflected ray theta angle for plate diffraction, cylinder reflection
DTCN	Dot product of unit vectors of rays tangent to cylinder from diffraction point (for more negative starting reflection point on cylinder)
DTCP	Dot product of unit vectors of rays tangent to cylinder from diffraction point (for more positive starting reflection point on cylinder) (also see DTI)
DTDC	Dot product of unit vectors of rays tangent to cylinder from diffraction point (for favored starting point on cylinder)
DTI	Dot product of source image vectors tangent to the cylinder (single reflection from plate MP)

FNP	Wedge angle number
LCD	Set true if corner ME of plate MP is on cylinder
LDC	Set true if edge M of plate MP is strong diffracting part of wedge (FNP<Ø)
LFRQFL	Logical variable set true if the frequency is the same as it was the last time this routine was called
LGDNFL	Logical variable set true if the geometry data set name is the same as it was the last time this routine was called
LSHD	Logical variable used to indicate if a plate is totally shadowed from a source, where LSHD(MP) = T if plate MP is totally shadowed and F if not totally shadowed
LSRCFL	Logical variable set true if the source is the same as the last time routine was called
ME	Plate edge index
MEC	Index variable used to determine common edges
MEN	Index variable used to determine common edges
MEP	Array which contains the number of edges on each plate
MEX	Maximum number of edges on plate MP
MJ	End cap index
MP	Plate index
MPX	Integer of the total number of plates in the geometry
PDCR	Phi component of ray propagation direction after reflection from cylinder (ray diffracted by plate edge and then reflected by cylinder)

PHWR	Branch cut displacement angle for diffraction point along edge ME of plate MP
RC	Distance from z axis to plate corner
RE	Radius of cylinder at point defined by elliptical angle VC
TDCR	Theta component of ray propagation direction after reflection from cylinder (ray diffracted by plate edge and then reflected by cylinder)
UCD	Z component of reflection point location on cylinder for ray which is reflected by cylinder and diffracted by edge ME of plate MP
UDC	Z component of starting point location on cylinder (for ray tracing algorithm) for ray diffracted by plate edge and then reflected by cylinder
UR	Z location of the cylinder reflection point for ray reflected by cylinder and incident on edge corners
V	Matrix which contains the unit edge vector for all edges on all plates
VC	Elliptical angle defining location of a corner (2-D)
VCD	Elliptical angle defining reflection point on cylinder (2-D) for ray which is reflected by cylinder and diffracted by edge ME of plate MP
VDC	Elliptical angle defining starting point on cylinder (for ray tracing algorithm) for ray diffracted by plate edge and then reflected by cylinder
VI	X,Y,Z components of propagation direction of ray incident on cylinder reflection point
VMAG	Matrix which contains the length of all edges on all plates

VR	Phi angle defining location of reflection point on cylinder for ray reflected by the cylinder and then incident on the edge corners
VTCN	Phi angle defining vectors from the diffraction point tangent to the cylinder for the more negative end cap
VTCP	Phi angles defining vectors from source image point through plate MP tangent to the cylinder; also from edge diffraction point for the more positive end cap
VTI	Elliptical angle defining direction of the two rays from image source tangent to the cylinder (single reflection of source ray from plate MP)
X	Matrix which contains the x,y,z components of all plate corners in RCS
XC	Modified plate corner location used in determining cylinder reflection plate diffraction limits
XDC	X,Y,Z components of starting diffraction point location on edge ME (for ray tracing algorithm) for ray diffracted by plate edge and reflected by cylinder
XI	Matrix containing the x,y,z components of the source image locations for single and double reflections off all plate combinations
XOB	Cylinder reflection point location for ray diffracted by plate edge and then reflected by the cylinder Also, x,y,z components of starting reflection point on cylinder
XS	The x,y,z components of the source location in RCS
ZC	Point where end cap intersects z axis of reference coordinate system, where ZC(1) refers to the more positive end cap and ZC(2) refers to the more negative end cap

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
A	/GEOMEL/
B	/GEOMEL/
CTC	/GEOMEL/
FNP	/FNANG/
LFRQFL	/SAME/
LGDNFL	/SAME/
LSHD	/LSHDT/
LSRCFL	/SAME/
MEP	/GEOPLA/
MPX	/GEOPLA/
V	/GEOPLA/
VMAG	/EDMAG/
X	/GEOPLA/
XI	/IMAINF/
XS	/SORINF/
ZC	/GEOMEL/

B. OUTPUT

OUTPUT	LOCATION
BCD	/BNDRCL/
BTDC	/BNDDCL/
BTI	/BNDICL/
DDC	/BNDDCL/
DTDC	/BNDDCL/
DTI	/BNDICL/

LDC	/LDCBY/
PDCR	/BNDDCL/
PHWR	/BRNPHW/
TDCR	/BNDDCL/
UCD	/BNDRCL/
UDC	/BNDDCL/
VCD	/BNDRCL/
VDC	/BNDDCL/
VTI	/BNDICL/

6. CALLING ROUTINE:

GTDDRV

7. CALLED ROUTINES:

BTAN2

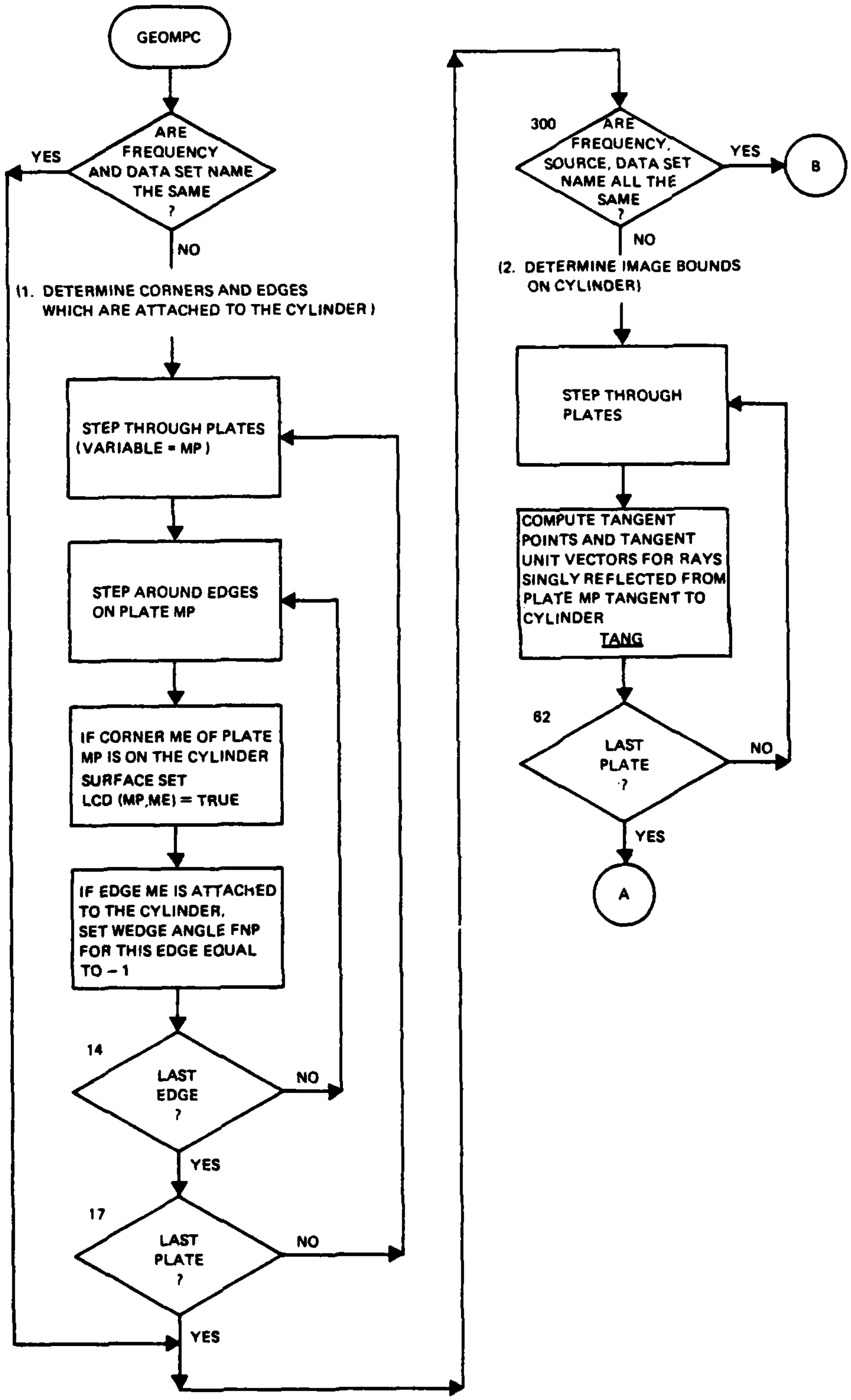
DPTNFW

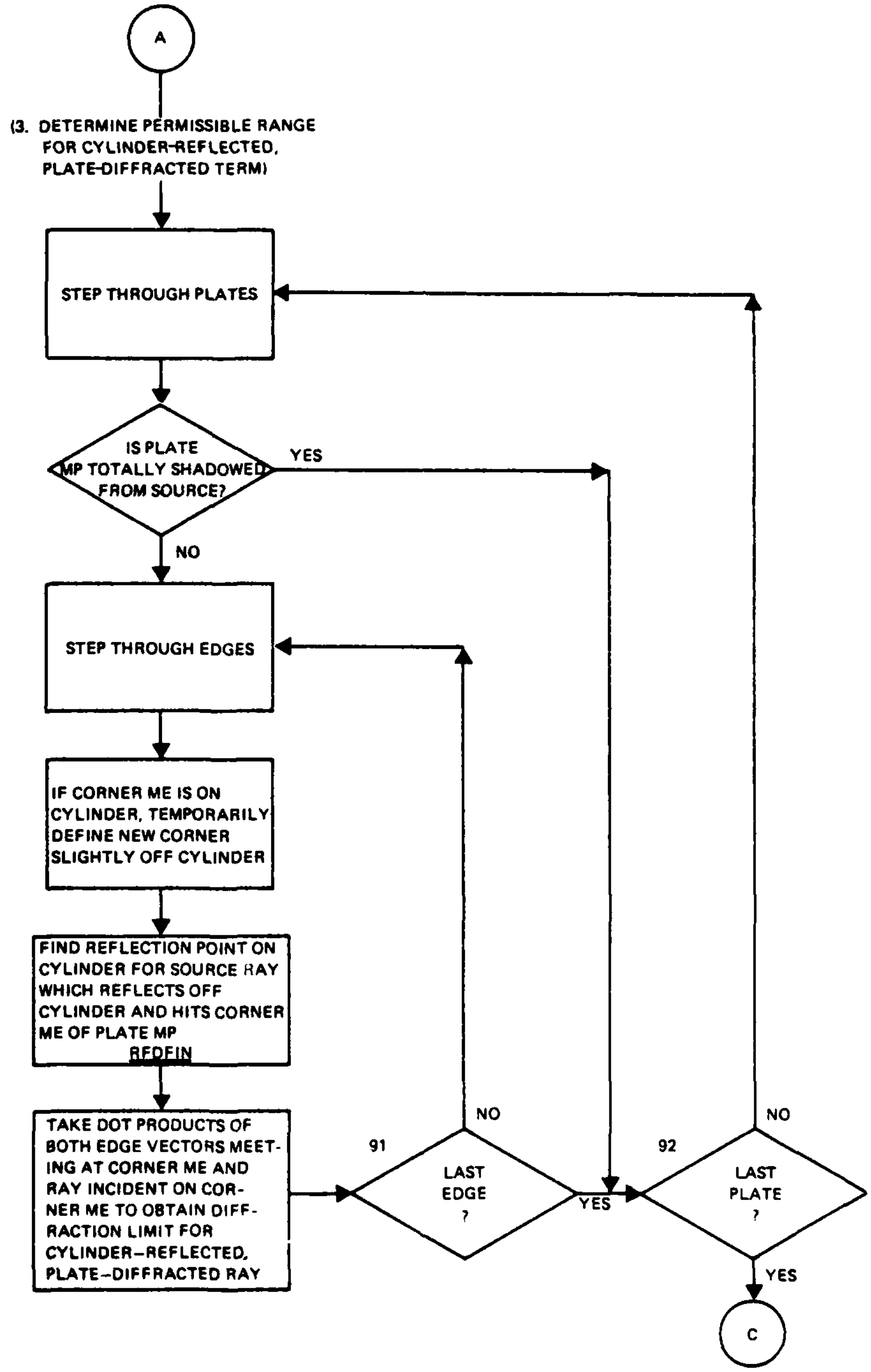
RFDFIN

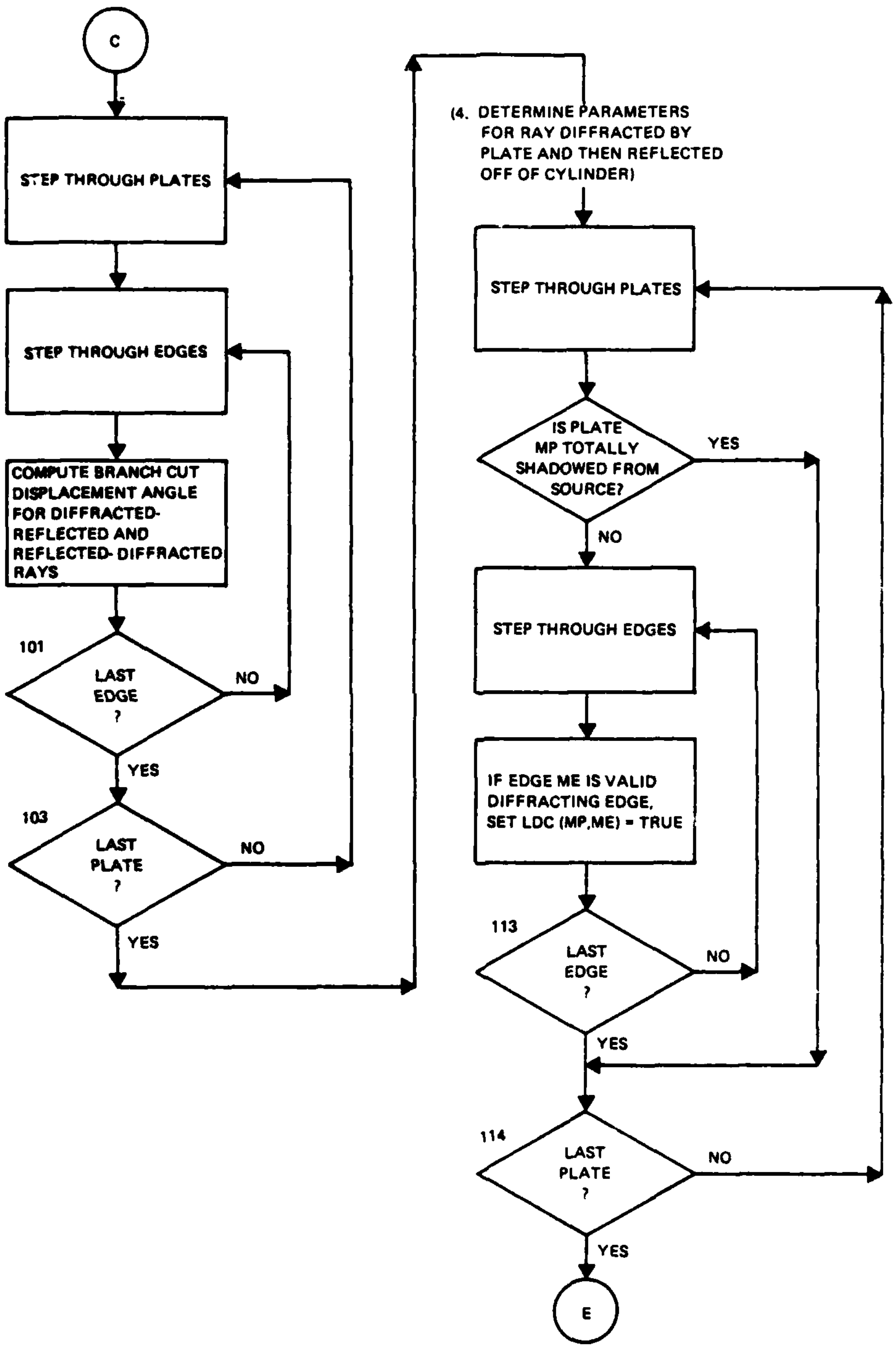
TANG

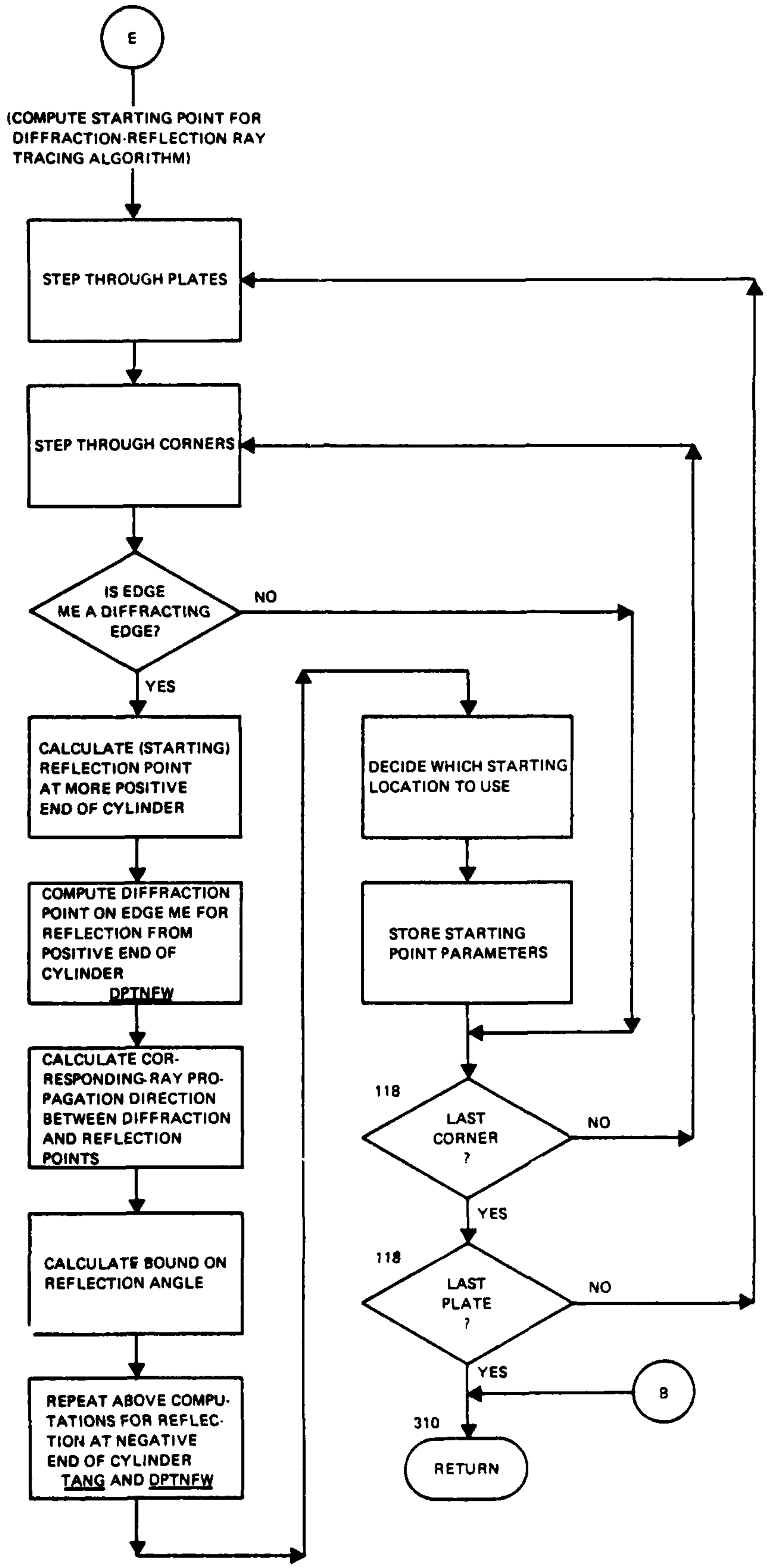
8. REFERENCE:

A. R. J. Marhefka, "Analysis of Aircraft Wing-Mounted Antenna Patterns," Report 2902-25, June 1976, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Grant No. NGL 36-008-138 for National Aeronautics and Space Administration.


GEOMPC
ARE FREQUENCY AND DATA SET NAME THE SAME ?
YES
NO
(1. DETERMINE CORNERS AND EDGES WHICH ARE ATTACHED TO THE CYLINDER)
STEP THROUGH PLATES (VARIABLE = MP)
STEP AROUND EDGES ON PLATE MP
IF CORNER ME OF PLATE MP IS ON THE CYLINDER SURFACE SET LCD (MP,ME) = TRUE
IF EDGE ME IS ATTACHED TO THE CYLINDER, SET WEDGE ANGLE FNP FOR THIS EDGE EQUAL TO − 1
14
LAST EDGE ?
NO
YES
17
LAST PLATE ?
NO
YES
300
ARE FREQUENCY, SOURCE, DATA SET NAME ALL THE SAME ?
YES
B
NO
(2. DETERMINE IMAGE BOUNDS ON CYLINDER)
STEP THROUGH PLATES
COMPUTE TANGENT POINTS AND TANGENT UNIT VECTORS FOR RAYS SINGLY REFLECTED FROM PLATE MP TANGENT TO CYLINDER
TANG
82
LAST PLATE ?
NO
YES
A


A
(3. DETERMINE PERMISSIBLE RANGE FOR CYLINDER-REFLECTED, PLATE-DIFFRACTED TERM)
STEP THROUGH PLATES
IS PLATE MP TOTALLY SHADOWED FROM SOURCE?
YES
NO
STEP THROUGH EDGES
IF CORNER ME IS ON CYLINDER, TEMPORARILY DEFINE NEW CORNER SLIGHTLY OFF CYLINDER
FIND REFLECTION POINT ON CYLINDER FOR SOURCE RAY WHICH REFLECTS OFF CYLINDER AND HITS CORNER ME OF PLATE MP
REFDIN
TAKE DOT PRODUCTS OF BOTH EDGE VECTORS MEETING AT CORNER ME AND RAY INCIDENT ON CORNER ME TO OBTAIN DIFFRACTION LIMIT FOR CYLINDER–REFLECTED, PLATE–DIFFRACTED RAY
91
LAST EDGE ?
NO
YES
92
LAST PLATE ?
NO
YES
C


C
STEP THROUGH PLATES
STEP THROUGH EDGES
COMPUTE BRANCH CUT DISPLACEMENT ANGLE FOR DIFFRACTED-REFLECTED AND REFLECTED-DIFFRACTED RAYS
101
LAST EDGE ?
NO
YES
103
LAST PLATE ?
NO
YES
(4. DETERMINE PARAMETERS FOR RAY DIFFRACTED BY PLATE AND THEN REFLECTED OFF OF CYLINDER)
STEP THROUGH PLATES
IS PLATE MP TOTALLY SHADOWED FROM SOURCE?
YES
NO
STEP THROUGH EDGES
IF EDGE ME IS VALID DIFFRACTING EDGE, SET LDC (MP,ME) = TRUE
113
LAST EDGE ?
NO
YES
114
LAST PLATE ?
NO
YES
E


E
(COMPUTE STARTING POINT FOR DIFFRACTION-REFLECTION RAY TRACING ALGORITHM)
STEP THROUGH PLATES
STEP THROUGH CORNERS
IS EDGE ME A DIFFRACTING EDGE?
NO
YES
CALCULATE (STARTING) REFLECTION POINT AT MORE POSITIVE END OF CYLINDER
COMPUTE DIFFRACTION POINT ON EDGE ME FOR REFLECTION FROM POSITIVE END OF CYLINDER
DPTNEW
CALCULATE CORRESPONDING RAY PROPAGATION DIRECTION BETWEEN DIFFRACTION AND REFLECTION POINTS
CALCULATE BOUND ON REFLECTION ANGLE
REPEAT ABOVE COMPUTATIONS FOR REFLECTION AT NEGATIVE END OF CYLINDER
TANG AND DPTNEW
DECIDE WHICH STARTING LOCATION TO USE
STORE STARTING POINT PARAMETERS
118
LAST CORNER ?
NO
YES
118
LAST PLATE ?
NO
YES
B
310
RETURN

1. NAME: GETARG (GTD, INPUT, MOM, OUTPUT)

2. PURPOSE: GETARG will get the specified argument from the argument list.

3. METHOD: By specifying the location in the INTARG array and the type of argument, GETARG will return with the needed argument. If a NOPCOD is found, the default value is returned in the INTARG location.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
LITTYP	Literal type code
LOCARG	Location of data, symbol table, or literal table
NAMSYM	Symbol name
NDFALT	Default for requested argument
NDXARG	Index to the INTARG array
NTPARG	Argument type
NUMSYM	Symbol table pointer

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
FLTLIT	/PARTAB/
INTARG	/ARGCOM/
IPASS	/ARGCOM/
ISOFF	/ADEBUG/
KOLCOD	/PARTAB/
KOLNAM	/PARTAB/
KOLVAL	/PARTAB/
LITNUM	/PARTAB/

LUPRNT	/ADEBUG/
NDATBL	/PARTAB/
NDFALT	F.P.
NDXARG	F.P.
NOPCOD	/ADEBUG/
NPDATA	/PARTAB/
NTFLPT	/ADEBUG/
NTINT	/ADEBUG/
NTPARG	F.P.
NTSYMB	/ADEBUG/

B. OUTPUT — LOCATION

FLTARG	/ARGCOM/
INTARG	/ARGCOM/

6. CALLING ROUTINES:*

ESPARM (2)	PRTSYM (3)
EXCDRV (2,3)	RESTRT (1)
FLDDRV (2,3,4)	SETDRV (3)
GEODRV (1)	SOLDRV (3)
GETGEO (1,2,3,4)	TSKXQT (2,3,4)
LODDRV (3)	ZIJDRV (2,3)

*1-INPUT
2-GTD
3-MOM
4-OUTPUT

7. CALLED ROUTINES:

ASSIGN

CONVRT

ERROR

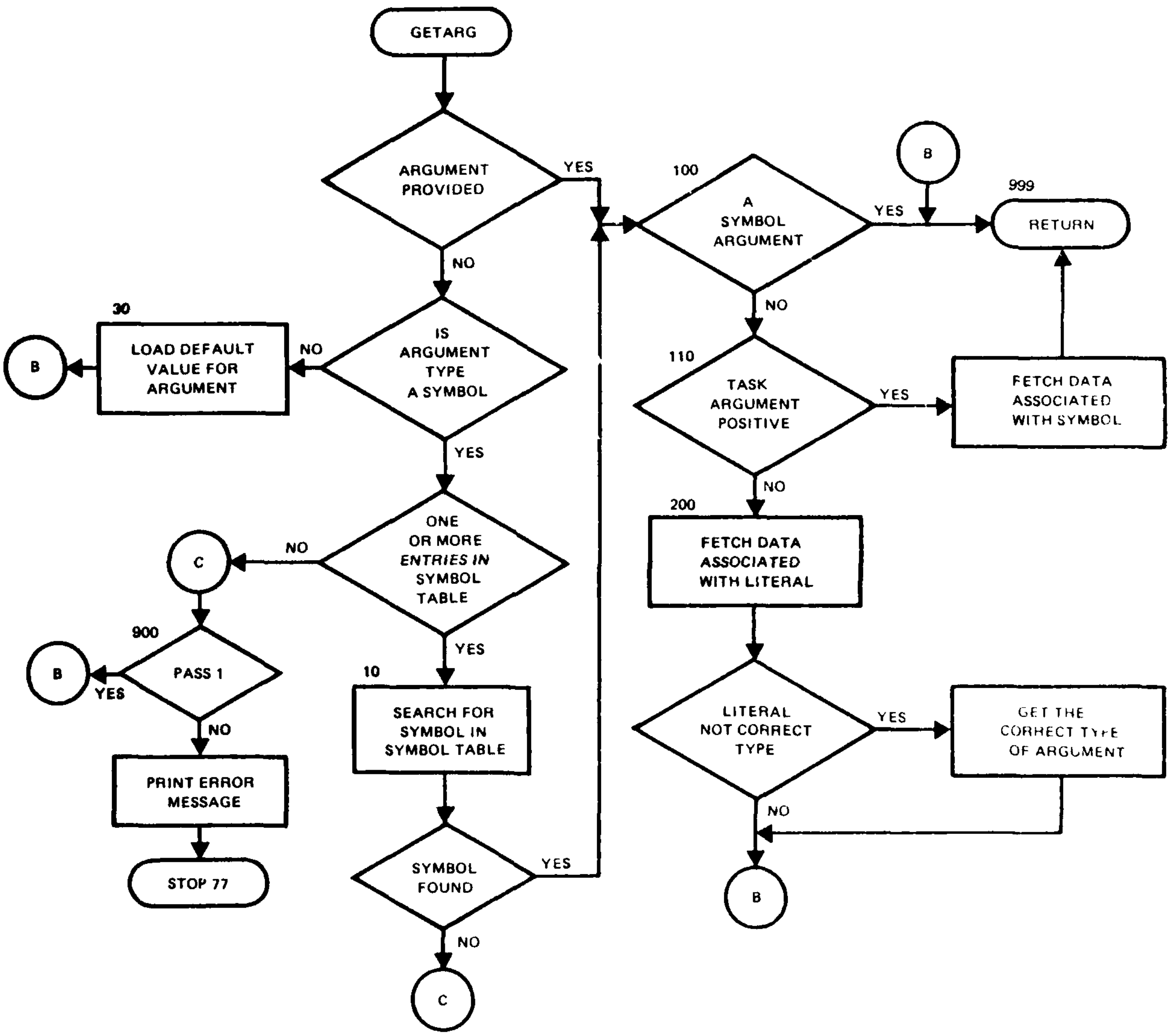
GETSYM

STATIN

STATOT

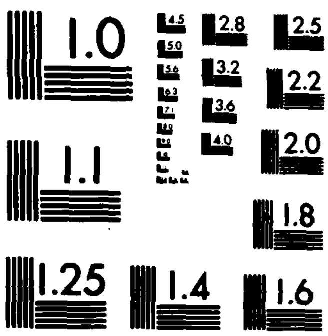
WLKBCK

GETARG (GTD, INPUT, MOM, OUTPUT)



AD-A137 464 GENERAL ELECTROMAGNETIC OODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS..(U) BDM CORP ALBUQUERQU NM D L KADLEC ET AL. SEP 83 BDM/A-83-020-TR-VOL-3-PT-2 2/4

UNCLASSIFIED RADC-TR-83-217-VOL-3-PT-2 F30602-81-C-0084 F/G 20/14 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1. NAME: GETFLD (GTD)

2. PURPOSE: This routine interprets the argument list data generated by ESRC and EFIELD commands in order to calculate source point and field point geometrical and electrical data.

3. METHOD: For ESRC data the basic parameters returned by ESPARM are used to calculate source (x,y,z) location, source field polarization, source amplitude, and source type, as follows:

SOURCE NAME	ISRCE SOURCE TYPE	GETFLD CALL#	SOURCE LOCATION OR PROPAGATION VECTOR	SOURCE AMPLITUDE	POLARIZATION
Dipole	-1	1		$(V_{THETA} + jV_{PHI})$	$\hat{x}$
		2	(x,y,z)		$\hat{y}$
		3			$\hat{z}$
Cylinder wave	-2	(Not implemented)			
Spherical	-3	1		Ex	$\hat{x}$
		2	(x,y,z)	Ey	$\hat{y}$
		3		Ez	$\hat{z}$
Plane wave	-4	1		0	--
		2	(-kx,-ky,-kz)	Eθ	$\hat{\theta}$
		3		Eϕ	$\hat{\phi}$

The source amplitude is $V_{THETA} + jV_{PHI}$ for the polarization selected by the user, and zero for the other two polarizations.

The fields from a spherical or plane wave include both co- and cross-polarized components, as specified by the ECC item.

For EFIELD data the observation point location is determined from the FLDDRV loop indices passed to GETFLD through common block FLDVAL. The quantities calculated for each coordinate system are:

COORDINATE SYSTEM	FIELD TYPE	OBSERVATION POINT LOCATION	NUMBER OF FIELD COMPONENTS	FIELD COMPONENT TANGENTS
Spherical	0 (FF)	$\hat{r}$	2	$\hat{\theta},\hat{\phi}$
Spherical	1 (NF)	(x,y,z)	3	$\hat{r},\hat{\theta},\hat{\phi}$
Rectangular	1 (NF)	(x,y,z)	3	$\hat{x},\hat{y},\hat{z}$
Cylindrical	1 (NF)	(x,y,z)	3	$\hat{\rho},\hat{\phi},\hat{z}$

If GETFLD is called with an invalid call number, an error message is printed, and execution terminates.

4. INTERNAL VARIABLES:

VARIABLE	DESCRIPTION
CURENT	Complex excitation parameter for the source
IC	Pointer to polarization component of source for call JCALL to GETFLD
ICT	Internal variable indicating coordinate system type
ISCTYP	Source-of-field type for this GETFLD call
ITYPE	Problem type: 2 = ESRC, 4 = EFIELD
JCALL	Source point number or observation point number for which this routine is called
LOOP1	Pointer to outer loop observation point position value
LOOP2	Pointer to center loop observation point position value
LOOP3	Pointer to inner loop observation point position value
M1, M2	Internal variables used to compute LOOP2 and LOOP3
NAMGEO	Name of geometry data set
NTAN	Number of source or field tangential components
PHI	Azimuthal angle in cylindrical or spherical coordinate system
R	Radial distance in cylindrical or spherical coordinate system
THETA	Polar angle in spherical coordinate system
TX,TY,TZ	Arrays of NTAN tangential unit vector components

UZ	Value of outer loop observation point parameter
VZ	Value of center loop observation point parameter
WZ	Value of inner loop observation point parameter
XC,YC,ZC	Source or field point location

5. I/O VARIABLES:

A. INPUT	LOCATION
CURENT	F.P.
E	/FLDVAL/
FARFLD	/FLDVAL/
ICTYPE	/FLDVAL/
ISRCE	/FLDVAL/
ITYPE	F.P.
JCALL	F.P.
L1	/FLDVAL/
L2	/FLDVAL/
L3	/FLDVAL/
LUPRNT	/ADEBUG/
NAMGEO	F.P.
NU	/FLDVAL/
U1	/FLDVAL/
V1	/FLDVAL/
W1	/FLDVAL/
X	/FLDVAL/

	Y	/FLDVAL/
	Z	/FLDVAL/
	ZERO	/ADEBUG/
B.	OUTPUT	LOCATION
	ISCTYP	F.P.
	NTAN	F.F.
	TX	F.P.
	TY	F.P.
	TZ	F.P.
	XC	F.P.
	YC	F.P.
	XC	F.P.

6. CALLING ROUTINE:

ZGTDRV

7. CALLED ROUTINES:

ASSIGN

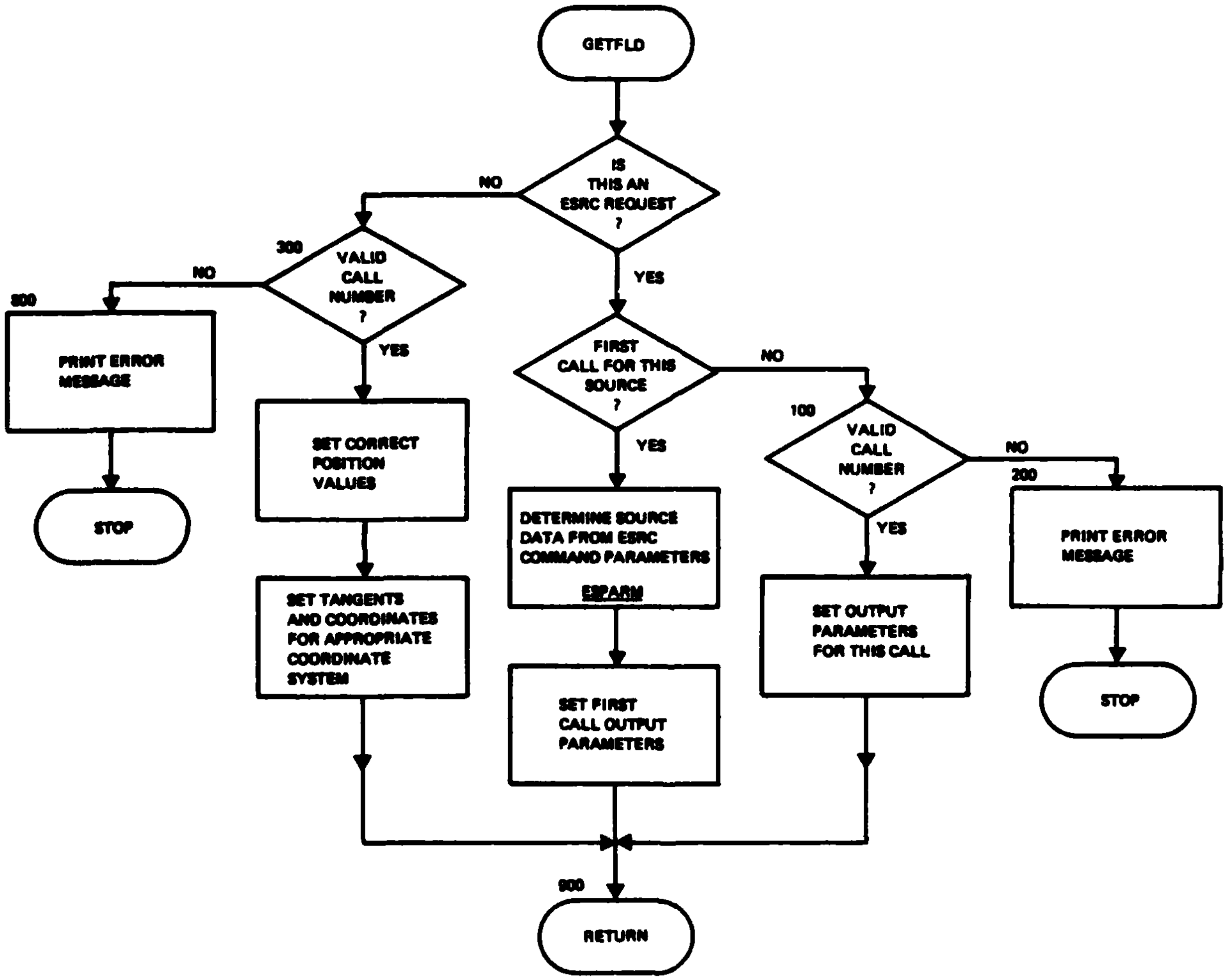
ESPARM

STATIN

STATOT

WLKBCK

GETFLD (GTD)


GETFLD
IS THIS AN ESRC REQUEST ?
NO
YES
300
VALID CALL NUMBER ?
NO
YES
800
PRINT ERROR MESSAGE
STOP
SET CORRECT POSITION VALUES
SET TANGENTS AND COORDINATES FOR APPROPRIATE COORDINATE SYSTEM
FIRST CALL FOR THIS SOURCE ?
NO
YES
DETERMINE SOURCE DATA FROM ESRC COMMAND PARAMETERS
ESPARM
SET FIRST CALL OUTPUT PARAMETERS
100
VALID CALL NUMBER ?
NO
YES
200
PRINT ERROR MESSAGE
STOP
SET OUTPUT PARAMETERS FOR THIS CALL
900
RETURN

1. NAME: GETGEO (GTD, MOM, OUTPUT)

2. PURPOSE: Determine the contents of the specified geometry data set.

3. METHOD: Each entry of the geometry data set pointed to by INTARG (LOCGEO) is checked for geometry type, and the appropriate counters incremented, when that geometry type's tag id is encountered in SEGTBL.

4. INTERNAL VARIABLES:

VARIABLE	DESCRIPTION
IBLK	Geometry SEGTBL block loop index
IL1M	Pointer to last entry in this SEGTBL block
INDEX	Pointer to geometry type: 1 = MOM, 2 = GTD
ITAG	Geometry entry tag id
ITAG1	GTD geometry type: 1 = Plate, 2 = Cylinder, 3 = End cap
IYRLOC	Geometry data set location in NDATBL
LOCGEO	Pointer (=2) to geometry specification in INTARG array
NAMEYR	Geometry data set name
NAMGEO	Pointer to default geometry name in NCODES
NCORN	Number of corners on GTD plate

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
INTARG	/ARGCOM/
IP217	/GEODAT/
ISEG	/SEGMNT/
ISGTBL	/SEGMNT/
KOLCOL	/PARTAB/

KOLNAM	/PARTAB/
LOCGEO	F.P.
INPUT	LOCATION
LUPRNT	/ADEBUG/
NAMSEG	/SEGMNT/
NCODES	/PARTAB/
NDATBL	/PARTAB/
NTSYMB	/ADEBUG/
SEGTBL	/SEGMNT/
ZERO	/ADEBUG/

B. OUTPUT — LOCATION

MAXBLK	/SEGMNT/
NAMSEG	/SEGMNT/
NPATCH	/SEGMNT/
NUMCYL	/GTDDAT/
NUMECP	/GTDDAT/
NUMGTD	/GTDDAT/
NUMPLT	/GTDDAT/
NUMSEG	/SEGMNT/
NWIRE	/SEGMNT/
UPDBLK	/SEGMNT/

6. CALLING ROUTINES:*

   EXCDRV (2,3)

   FLDDRV (2,3)

   SOLDRV (3)

   TSKXQT (2,3,4)

   ZIJDRV (2,3)

7. CALLED ROUTINES:

   ASSIGN

   CONVRT

   GETARG

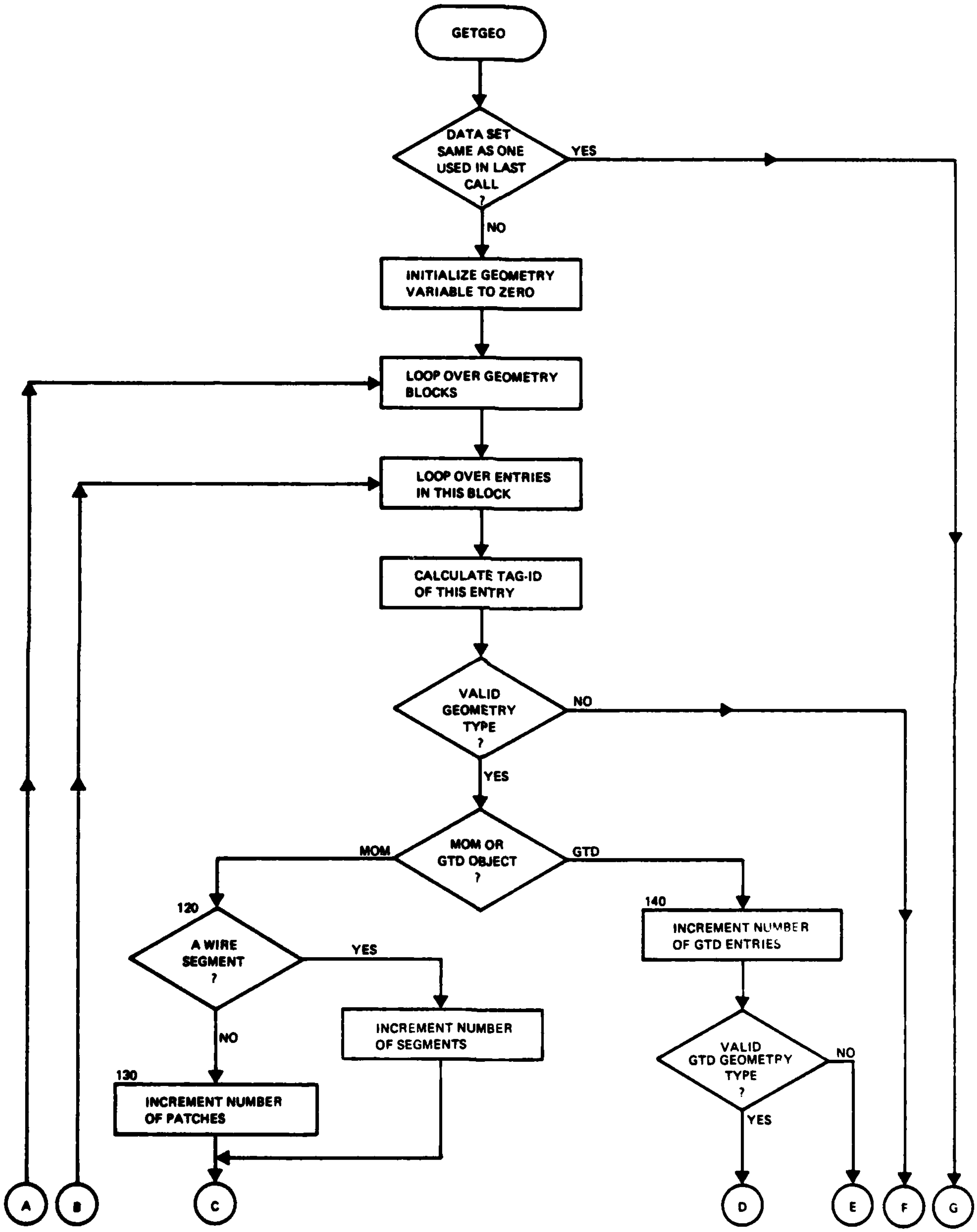
   GETSEG

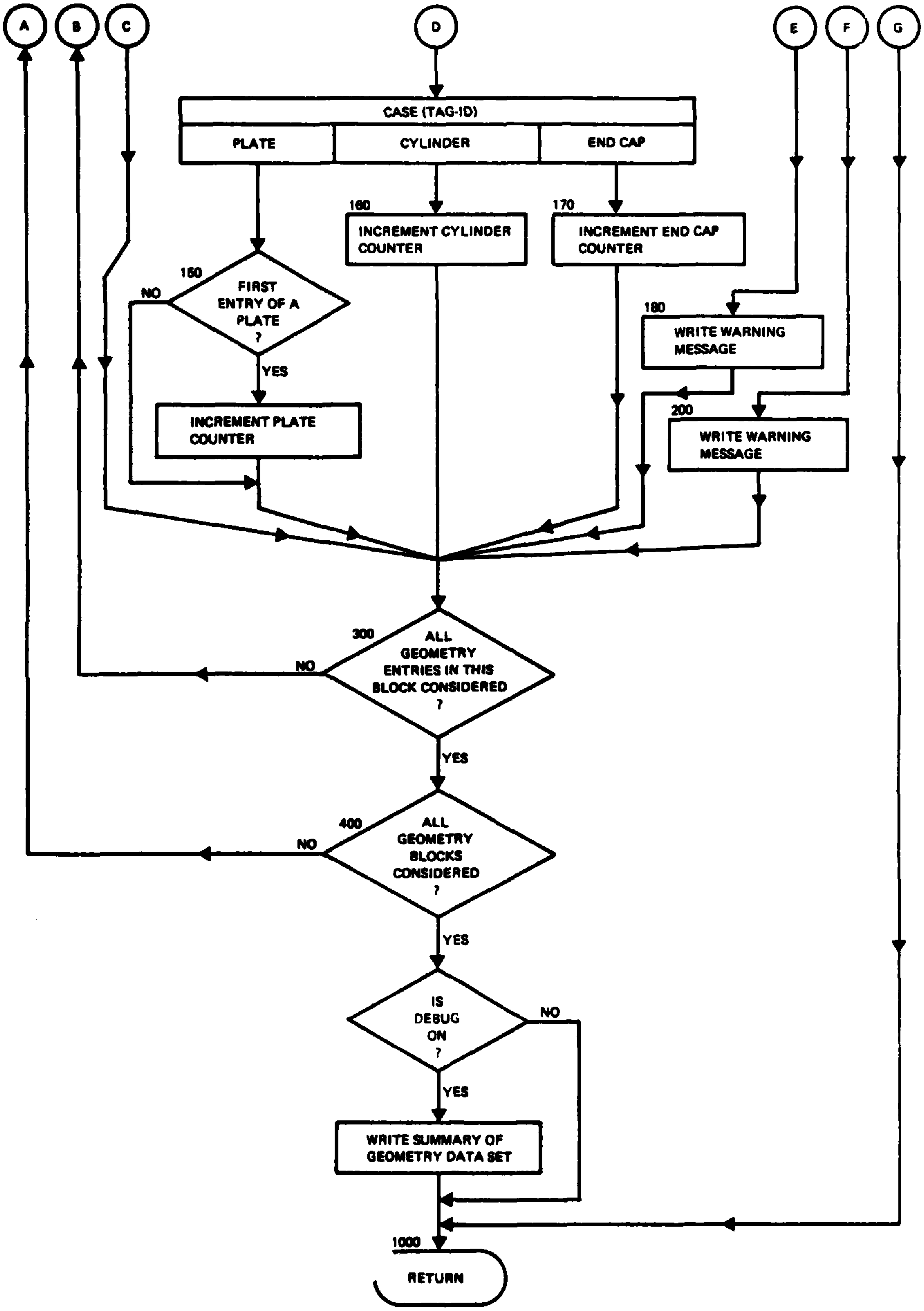
   STATIN

   STATOT

   WLKBCK

*2-GTD
3-MOM
4-OUTPUT


GETGEO
DATA SET SAME AS ONE USED IN LAST CALL ?
YES
NO
INITIALIZE GEOMETRY VARIABLE TO ZERO
LOOP OVER GEOMETRY BLOCKS
LOOP OVER ENTRIES IN THIS BLOCK
CALCULATE TAG-ID OF THIS ENTRY
VALID GEOMETRY TYPE ?
NO
YES
MOM OR GTD OBJECT ?
MOM
GTD
120
A WIRE SEGMENT ?
YES
NO
INCREMENT NUMBER OF SEGMENTS
130
INCREMENT NUMBER OF PATCHES
140
INCREMENT NUMBER OF GTD ENTRIES
VALID GTD GEOMETRY TYPE ?
NO
YES
A
B
C
D
E
F
G


A
B
C
D
E
F
G
CASE (TAG-ID)
PLATE
CYLINDER
END CAP
160
INCREMENT CYLINDER COUNTER
170
INCREMENT END CAP COUNTER
150
FIRST ENTRY OF A PLATE ?
NO
YES
INCREMENT PLATE COUNTER
180
WRITE WARNING MESSAGE
200
WRITE WARNING MESSAGE
300
ALL GEOMETRY ENTRIES IN THIS BLOCK CONSIDERED ?
NO
YES
400
ALL GEOMETRY BLOCKS CONSIDERED ?
NO
YES
IS DEBUG ON ?
NO
YES
WRITE SUMMARY OF GEOMETRY DATA SET
1000
RETURN

1. NAME: GETKWD (INPUT)

2. PURPOSE: To determine if the name in the calling argument is a keyword and, if so, to return the pointer to its location in the keyword table.

3. METHOD: The keyword table is searched until an entry for the argument NAME is found.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
INDEX	If not 0, pointer to the keyword name in the KWNAME array
NAME	Encoded name to be searched for in the keyword name array

5. I/O VARIABLES:

A. INPUT	LOCATION
KWMAX	/PARTAB/
KWNAME	/PARTAB/
NAME	F.P.
NCODES	/PARTAB/
NERCOD	/INPERR/
NTAB	/SCNPAR/
NTKEYW	/ADEBUG/
NVALMX	/SCNPAR/

B. OUTPUT	LOCATION
INDEX	F.P.
NCODE	/SCNPAR/
NSCNER	/SCNPAR/
NVAL	/SCNPAR/

6. CALLING ROUTINE:

SCAN

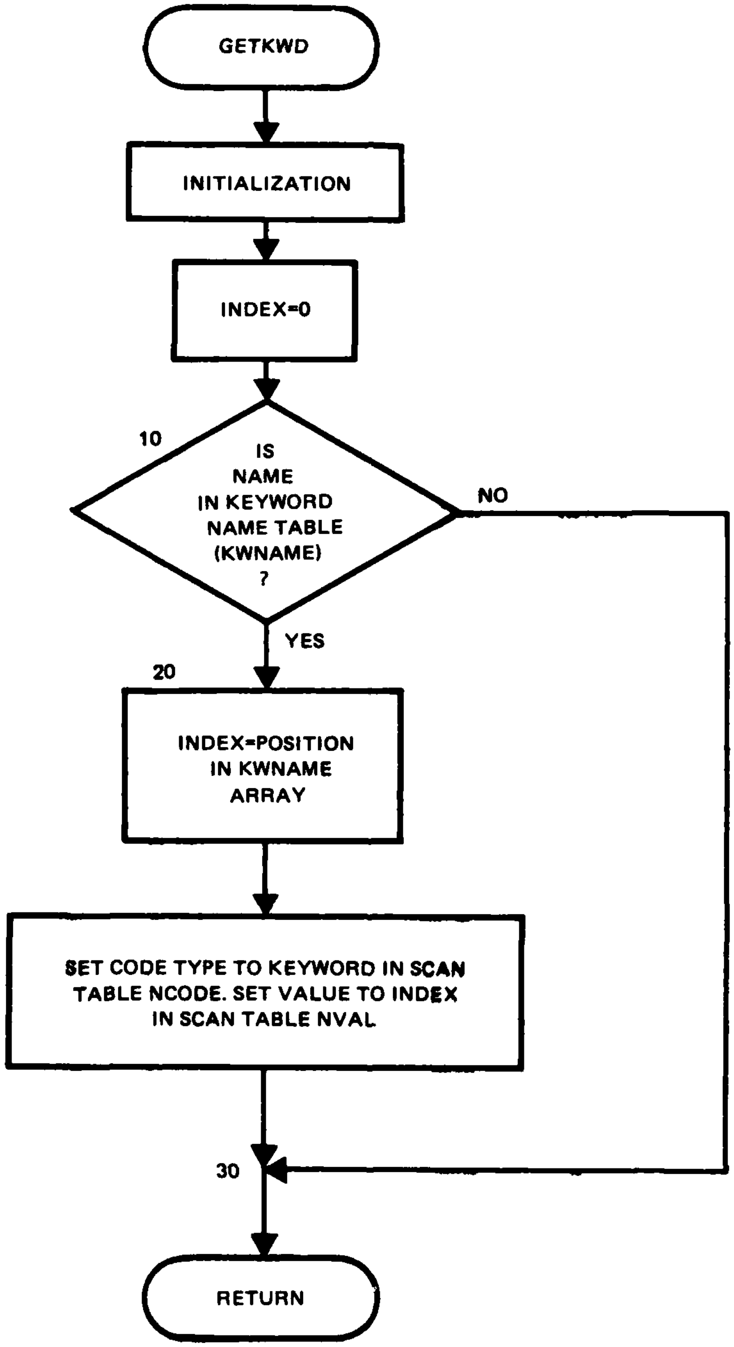
7. CALLED ROUTINES:

ASSIGN

STATIN

STATOT

WLKBCK


GETKWD
INITIALIZATION
INDEX=0
10
IS NAME IN KEYWORD NAME TABLE (KWNAME) ?
NO
YES
20
INDEX=POSITION IN KWNAME ARRAY
SET CODE TYPE TO KEYWORD IN SCAN TABLE NCODE. SET VALUE TO INDEX IN SCAN TABLE NVAL
30
RETURN

1. NAME: GETKWV (GTD, INPUT, MOM, OUTPUT)

2. PURPOSE: Get the value of a keyword.

3. METHOD: GETKWV uses a variable in common for each keyword. This value is returned to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
KW	NDXKWD
NDX	Index to the NCODES array for a keyword
NDXKWD	Keyword number
VALUKW	Keyword value to be returned

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
EPSR	/AMPZIJ/
FRQMHZ	/AMPZIJ/
ISON	/ADEBUG/
KWCOND	/PARTAB/
KWEPSR	/PARTAB/
KWFRQ	/PARTAB/
KWNAME	/PARTAB/
KWNMFL	/PARTAB/
KWTIME	/PARTAB/
LUPRNT	/ADEBUG/
NCODES	/PARTAB/
NDXKWD	F.P.
NFILES	/IOFLES/

	SIGMA	/AMPZIJ/
	TIMTGO	/SYSFIL/
B.	OUTPUT	LOCATION
	IERRF	/ADEBUG/
	VALUKW	F.P.

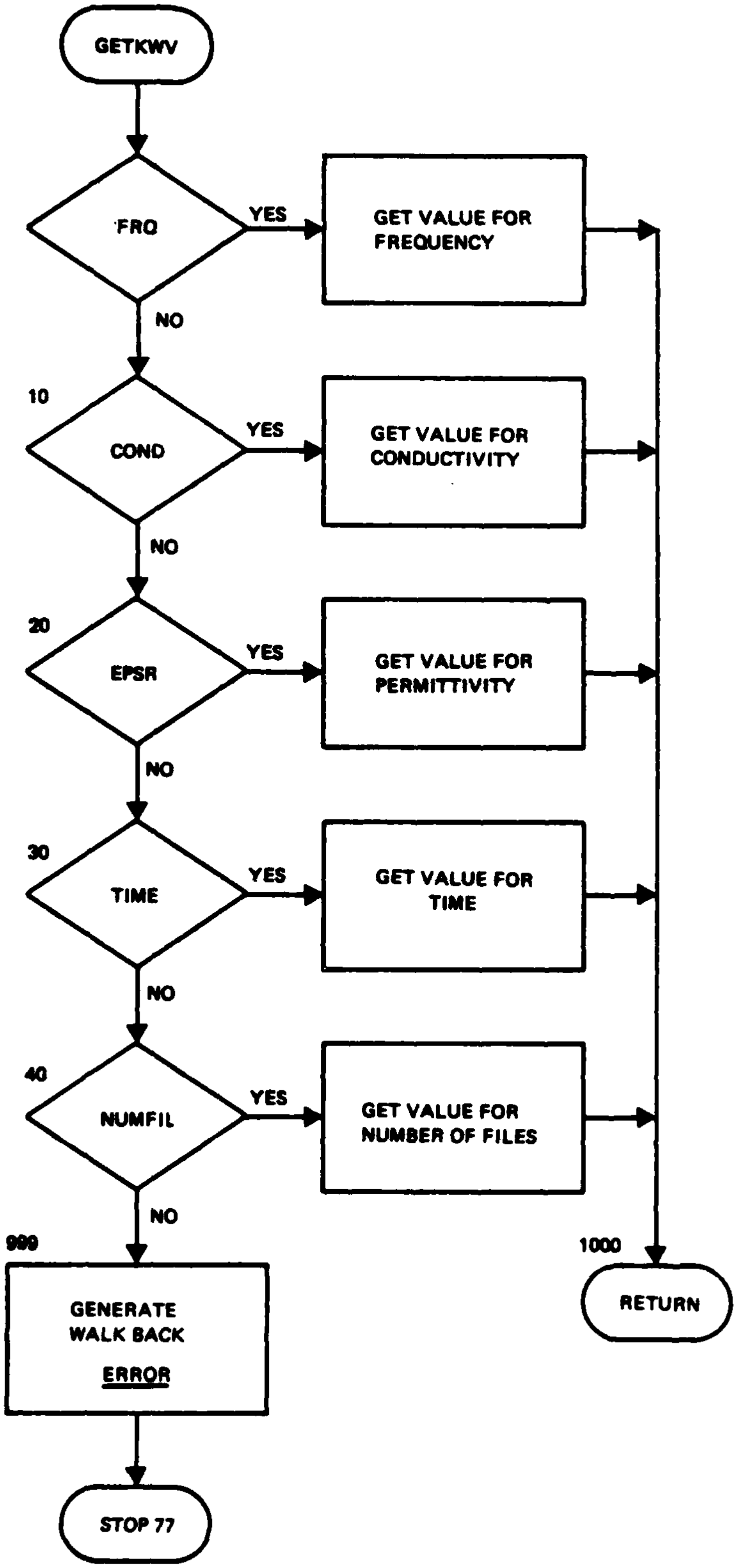
6. CALLING ROUTINE:

DMPDRV

7. CALLED ROUTINES:

ASSIGN	ERROR	STATOT
CONVRT	STATIN	WLKBCK

GETKWV (GTD, INPUT, MOM, OUTPUT)



1. NAME: GETPNT (INPUT)

2. PURPOSE: To retrieve the coordinates of the points specified in the input argument list.

3. METHOD: The array PTTBLE is searched for the correct point numbers. Failure to locate the point results in setting the point number negative in the calling argument and returning to the calling subroutine. If the point is found, its global coordinates are returned to the calling subroutine with a positive point number.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IP1	Point to be searched for
IP2	Number of second point to be searched for
I1	Point number for first point
I2	Point number for second point
NPTS	Interval variable for IPTS
X1	X coordinate of point IP1
X2	X coordinate of point IP2
Y1	Y coordinate of point IP1
Y2	Y coordinate of point IP2
Z1	Z coordinate of point IP1
Z2	Z coordinate of point IP2

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
IPTS	/PNTTBL/
IPTTBL	/PNTTBL/
I1	F.P.

I2	F.P.
PTTBLE	/PNTTBL/

B. OUTPUT — LOCATION

X1	F.P.
X2	F.P.
Y1	F.P.
Y2	F.P.
Z1	F.P.
Z2	F.P.

6. CALLING ROUTINES:

LNKGTD

PATCH

PLATE

PRTGTD

WYRDRV

7. CALLED ROUTINES:

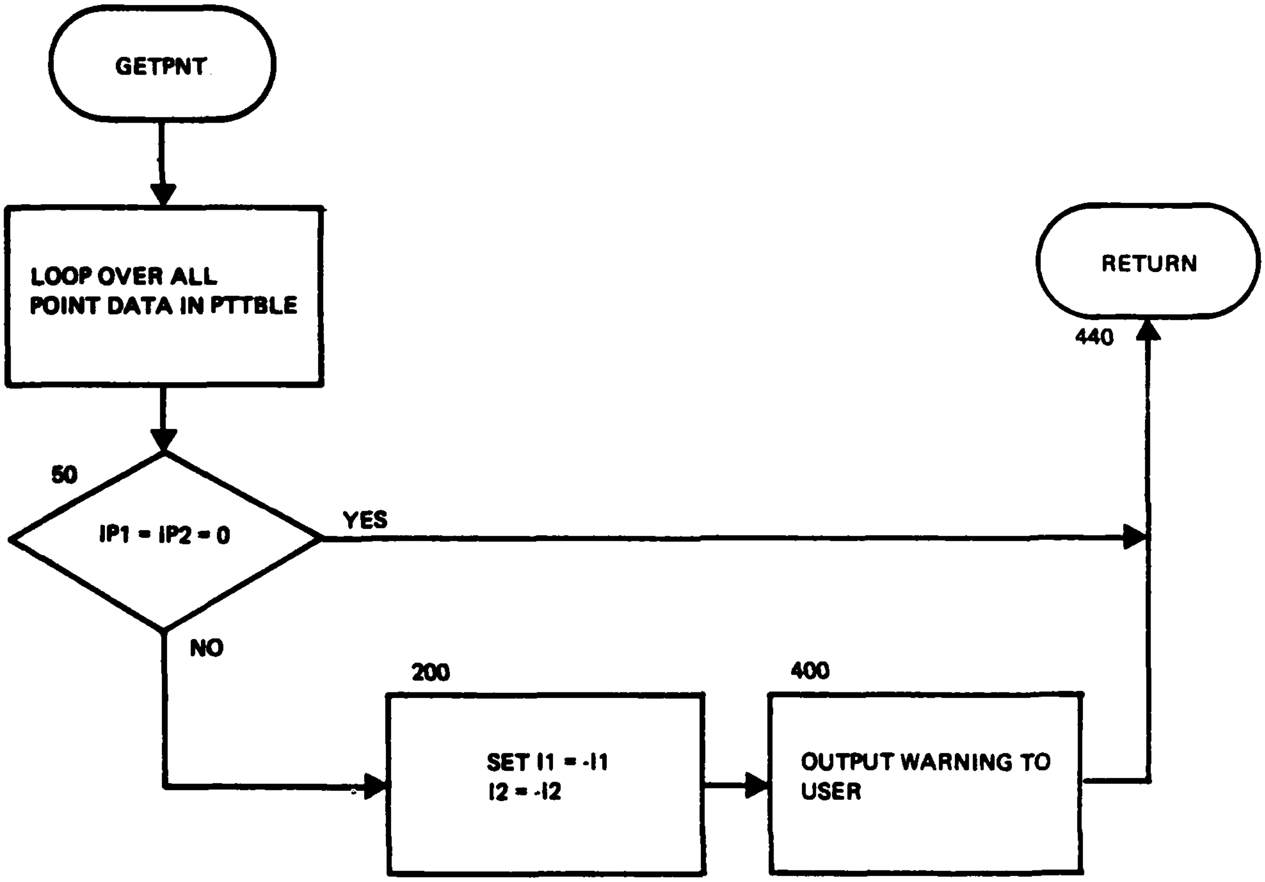
ASSIGN

STATIN

STATOT

WLKBCK

GETPNT (INPUT)



1. NAME: GETSEG (GTD, INPUT, MOM)

2. PURPOSE: This subroutine recovers the SEGTBL data associated with a given data block.

3. METHOD: If the current data block has been changed, then PUTSYM is called to store the current data block. Otherwise GETSYM is called to recover the requested data block.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ISEG	The number of filled entries in the last data block
MAXBLK	The total number of data blocks
MAXSEG	The maximum number of segments in each data block
NAMSEG	Input argument designating symbol name to be retrieved
NDXBLK	The current data block
NUMBLK	The requested data block

5. I/O VARIABLES

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
IPLTAG	/GTDDAT/
IP217	/GEODAT/
ISGTBL	/SEGMNT/
ISOFF	/ADEBUG/
LUPRNT	/ADEBUG/
MAXBLK	/SEGMNT/
MAXSEG	/SEGMNT/
NDXBLK	/SEGMNT/

PREVIOUS PAGE IS BLANK

NUMBLK	F.P.
SEGTBL	/SEGMNT/
UPDBLK	/SEGMNT/
B. OUTPUT	LOCATION
ISEG	/SEGMNT/
NDXBLK	/SEGMNT/
SEGTBL	/SEGMNT/
UPDBLK	/SEGMNT/

6. CALLING ROUTINES:*

BUBPLE (1)

CABC (3)

CNTGND (3)

CNVGTD (1)

EXCDRV (2,3)

FARFLD (3)

GEODRV (1)

GETGEO (1,2,3)

GTDDRV (2)

JCTION (1)

LNKGTD (1)

LNKJCT (1)

LODDRV (3)

LUDDRV (3)

NERFLD (3)

*1-INPUT
2-GTD
3-MOM

PLTDRV (1)

PLTSEG (1)

PRTGTD (1)

PUTSEG (1,2,3)

SEJCON (2,3)

SOLDRV (3)

SPWDRV (3)

SUBPAT (1)

WYRDRV (1)

7. CALLED ROUTINES:

ASSIGN

GETSYM

PUTSYM

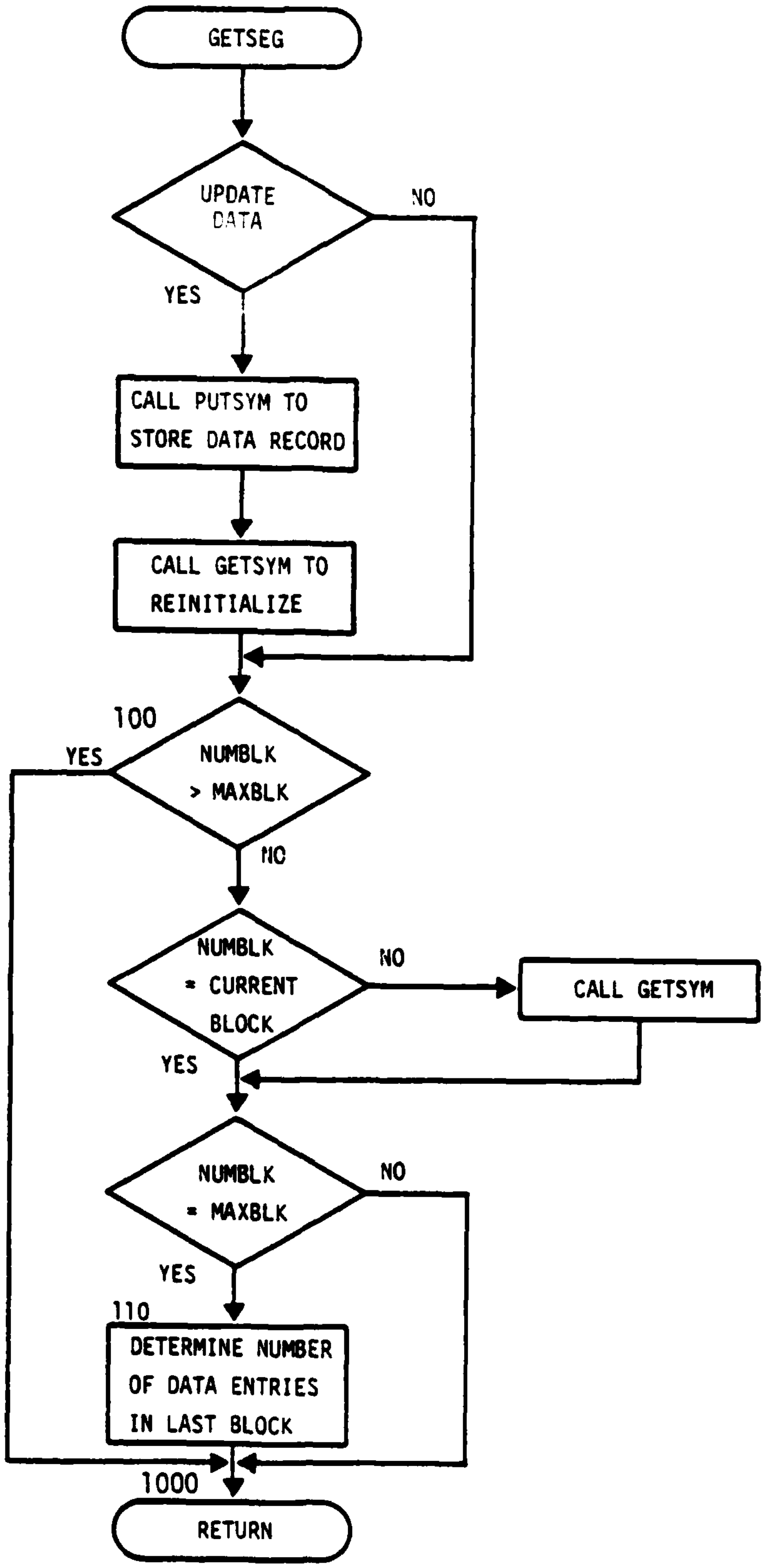
STATIN

STATOT

WLKBCK

*1-INPUT
2-GTD
3-MOM

GETSEG (GTD, INPUT, MOM)



1. NAME: GETSYM (GTD, INPUT, MOM, OUTPUT)

2. PURPOSE: To retrieve the file records specified for the symbol MATNAM and place them in the specified in-core array.

3. METHOD: Calculate the record length and symbol attributes. Determine the file number and edition calls to RDEFIL to retrieve the in the records specified.

Any GEMACS file may contain more than one edition of a data set. For example, when a data set is generated in a loop, there will be as many editions of the data set as there are passes through the loop. The format of a multiple edition data set is shown in figure 1. The global first word location of the edition being accessed (the "present" edition) is stored in NDATBL column KOLFST.

To access the correct record number of the present edition, GETSYM calls FNDREC which returns the first global record of the present edition. This value is added to the specified record numbers. For example, suppose for the file in figure 1, the present edition was edition #3 (LOCFST = 79) and GETSYM was called for record numbers 5 through 8. (These correspond to global records 23-26.) FNDREC returns the record number for word 79 as 19. Hence the global record numbers for file retrieval are:

$$IREC1 = 5 + 19 - 1 = 23$$

$$IREC2 = 8 + 19 - 1 = 26$$

File retrieval proceeds as shown in figure 2. Assume that the current location of the file (LOCNOW) is at the end of record 7 of the third edition (LOCNOW = 114). Since IRECNW (= 25) is greater than the first record to be read, the file is rewound. Each record of the file up to the last record desired is read, but only the desired records are retained, as shown in figure 3. The first 22 records are read, but each overwrites the previous one. When record 23 is read, it overwrites 22, but since record 23 is one we wish to retain, the storage pointer LOCSTR is incremented five words (length of record 23) so that record 24 when read will not overwrite record 23. In the same way, once record 24 has been read, LOCSTR is incremented another four words (length of record 24) so that record 25 can be read without overwriting record 24. Thus, at the end of GETSYM, the data on the requested records of the symbol are contained sequentially in the specified TMPBUF array.

Two important special cases arise with GETSYM. When GETSYM is called with a null symbol, all internal variables are reset to zero. Second, when GETSYM is called with a valid symbol, but with record numbers equal to zero, the file associated with that symbol is positioned at the beginning of the present edition as determined by LOCFST.

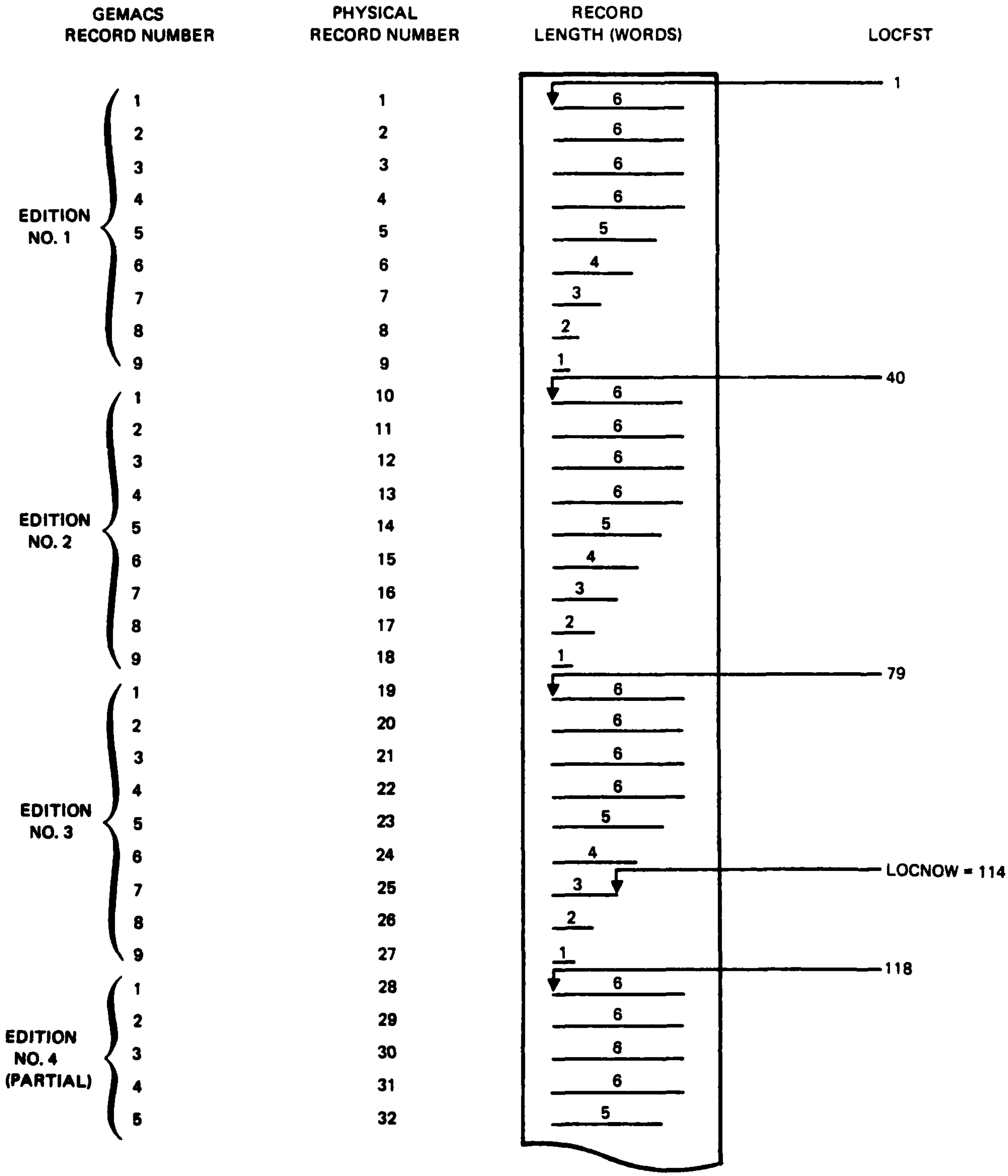


Figure 1. A Multiple Edition Data Set with Three Complete and One Partial Editions (The matrix stored is a banded, upper triangular, real matrix derived from a 9 x 9 real matrix.)

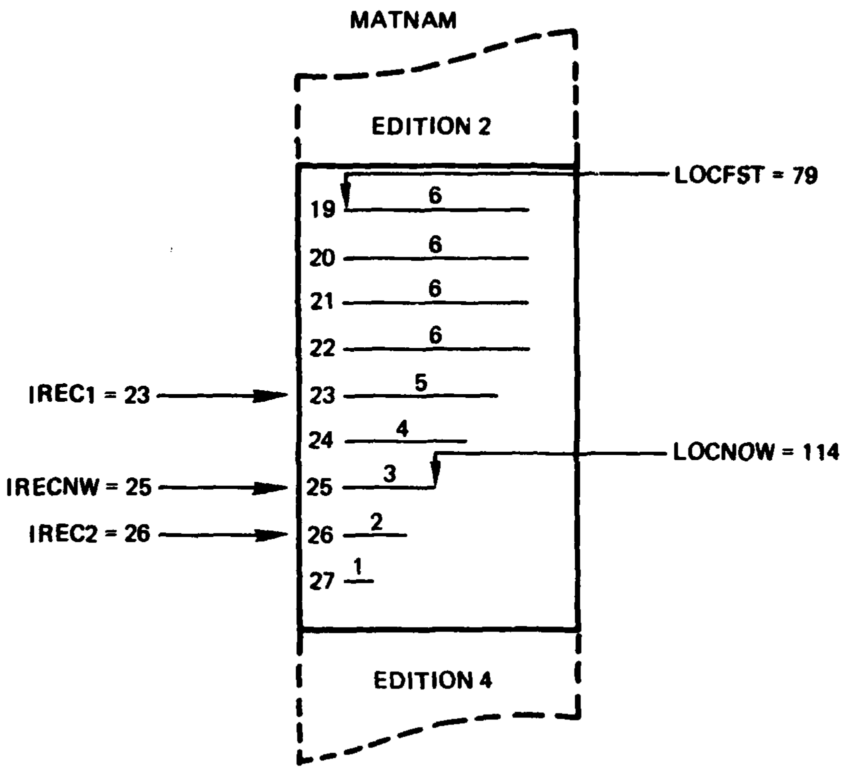


Figure 2. Important GETSYM Variables

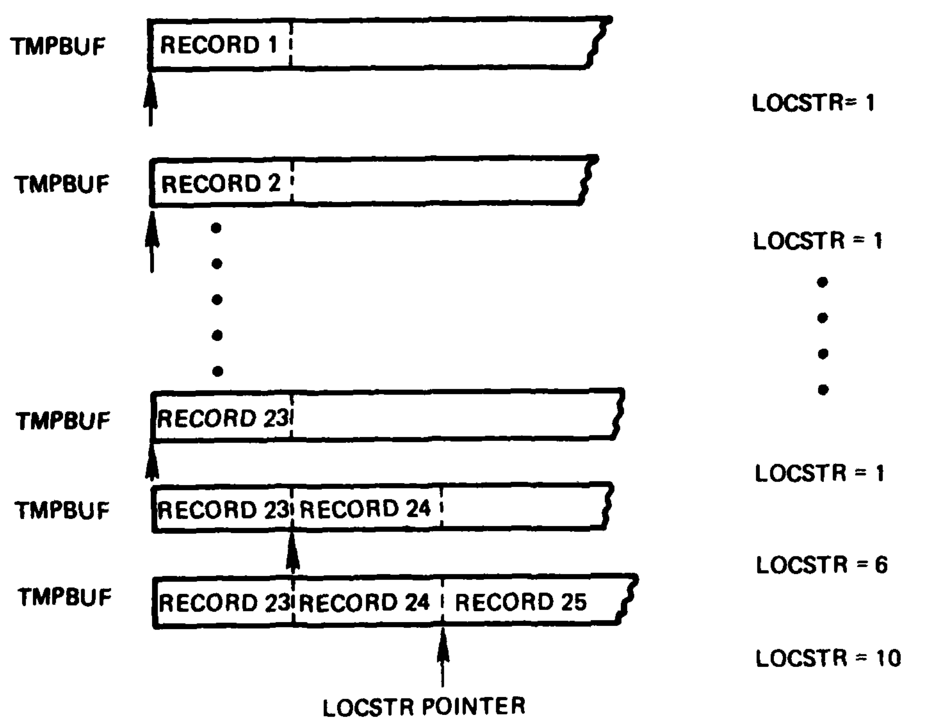


Figure 3. Illustrating How File Records Are Placed in TMPBUF

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IBAND	Flag indicating a banded data set
IFILE	The logical unit designation for the symbol
ILOWER	Flag indicating a lower triangular matrix
IORDER	Flag indicating a transposed matrix
IR	Index over file records to be read
IRC1	First local record to be retrieved
IRC2	Last local record to be retrieved
IREAD	First file record to be read
IRECNW	Record number of present file position
IRECST	Record number of beginning of the edition
IREC1	Global record number corresponding to IRC1
IREC2	Global record number corresponding to IRC2
IROWM1	First record to be retrived on file minus one
IR1	Internal variable equal to IRC1
IR2	Internal variable equal to IRC2
ISV	Pointer to symbol name in symbol table
IT	Pointer to TMPBUF array for in-core storage transfer
IUPPER	Flag indicating an upper triangular matrix
IWRD1	First word address for present edition of data set
KBAND	Flag indicating a banded parent data set
KLINK	Pointer to data set linked to parent

LINK	Pointer to parent data set in symbol table
LOCFST	Location of first word for this edition of symbol
LOCNOW	Current location of logical unit containing data set
LOCSTR	Location in the TMPBUF array
LSTWRD	Last word of data set to be retrieved
LWRUPR	Flag indicating a triangular matrix
MATNAM	Input argument designating symbol name to be retrieved
MAXWRD	Total number of words to be retrieved
MORE	Flag set if specified attribute is present in symbol
NO	Dummy variable (= 1)
NA	Alpha format of symbol name
NAMPRT	Symbolic name of parent data set
NAMSAV	Saved value of current MATNAM for next call
NBITWD	The bit set attribute word for the data set
NPRELM	Number of computer words per element of the data set
NPRPRT	Number of rows in parent
NPRREC	Number of rows in this record
NUMROW	Number of rows in symbol
TMPBUF	Input argument designating array into which data are to be retrieved

5. I/O VARIABLES:

A. INPUT	LOCATION
DBGPRT	/ADEBUG/

GETSYM (GTD, INPUT, MOM, OUTPUT)

	FLTSYM	/SYMSTR/
	IOFILE	/IOFLES/
	IRC1	F.P.
	IRC2	F.P.
	ISOFF	/ADEBUG/
	ISON	/ADEBUG/
	KBBAND	/PARTAB/
	KBCPLX	/PARTAB/
	KBDPRE	/PARTAB/
	KBLWRT	/PARTAB/
	KBORDR	/PARTAB/
	KBUPRT	/PARTAB/
	KOLBIT	/PARTAB/
	KOLFST	/PARTAB/
	KOLLNK	/PARTAB/
	KOLLOC	/PARTAB/
	KOLNAM	/PARTAB/
	KOLROW	/PARTAB/
	LUPRNT	/ADEBUG/
	MATNAM	F.P.
	NDATBL	/PARTAB/
	NPDATA	/PARTAB/
B.	OUTPUT	LOCATION
	IERRF	/ADEBUG/
	TMPBUF	F.P.

6. CALLING ROUTINES:*

BACSUB (3)

BANDIT (3)

DECOMP (3)

DMPDRV (1,2,3,4)

EGFMAT (3)

EXCDRV (2,3)

FLDDRV (3,4)

FLDOUT (4)

GEODRV (1)

GETARG (1,2,3,4)

GETSEG (1,2,3)

LODDRV (3)

LUDDRV (3)

PRTSYM (3)

PUTSEG (1,2,3)

PUTSYM (1,2,3,4)

REBLCK (3)

RESTRT (1)

SETDRV (3)

SOLDRV (3)

STRTUP (2,3,4)

*1-INPUT
2-GTD
3-MOM
4-OUTPUT

SUBPAT (1)

SYMDEF (1,2,3,4)

WRTCHK (1,2,3,4)

ZIJDRV (2,3)

7. CALLED ROUTINES:

ASSIGN

CONVRT

ERROR

FNDREC

IBITCK

IJMOD (intrinsic)

MOVFIL

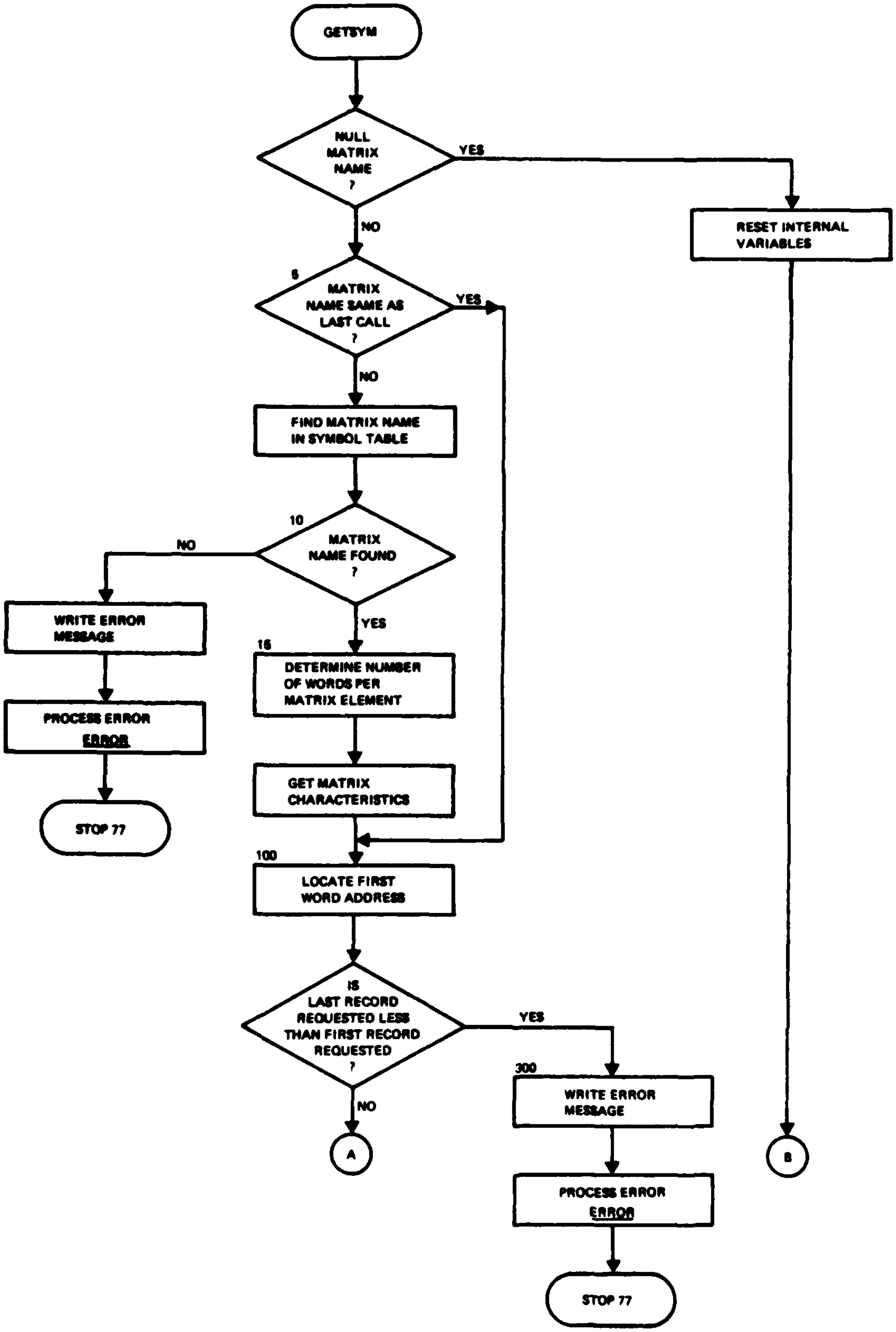
RDEFIL

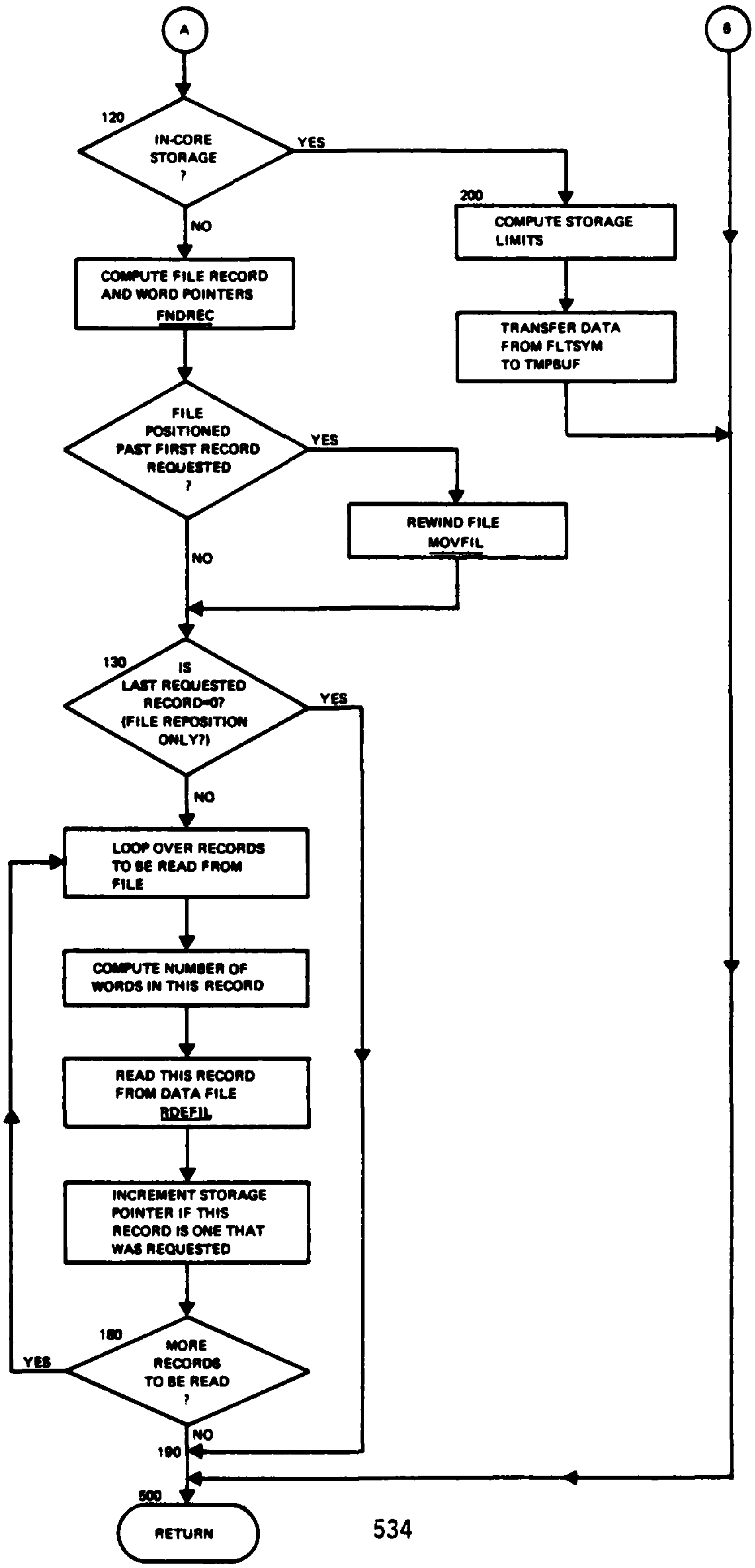
STATIN

STATOT

WLKBCK

*1-INPUT
2-GTD
3-MOM
4-OUTPUT


GETSYM
NULL MATRIX NAME ?
YES
NO
RESET INTERNAL VARIABLES
5
MATRIX NAME SAME AS LAST CALL ?
YES
NO
FIND MATRIX NAME IN SYMBOL TABLE
10
MATRIX NAME FOUND ?
NO
YES
WRITE ERROR MESSAGE
PROCESS ERROR
ERROR
STOP 77
15
DETERMINE NUMBER OF WORDS PER MATRIX ELEMENT
GET MATRIX CHARACTERISTICS
100
LOCATE FIRST WORD ADDRESS
IS LAST RECORD REQUESTED LESS THAN FIRST RECORD REQUESTED ?
YES
NO
A
300
WRITE ERROR MESSAGE
PROCESS ERROR
ERROR
STOP 77
B


A
B
120
IN-CORE STORAGE ?
YES
NO
200
COMPUTE STORAGE LIMITS
TRANSFER DATA FROM FLTSYM TO TMPBUF
COMPUTE FILE RECORD AND WORD POINTERS FNDREC
FILE POSITIONED PAST FIRST RECORD REQUESTED ?
YES
REWIND FILE MOVFIL
NO
130
IS LAST REQUESTED RECORD=0? (FILE REPOSITION ONLY?)
YES
NO
LOOP OVER RECORDS TO BE READ FROM FILE
COMPUTE NUMBER OF WORDS IN THIS RECORD
READ THIS RECORD FROM DATA FILE RDEFIL
INCREMENT STORAGE POINTER IF THIS RECORD IS ONE THAT WAS REQUESTED
180
MORE RECORDS TO BE READ ?
YES
NO
190
500
RETURN

1. NAME: GNDREF (MOM)

2. PURPOSE: This subroutine multiplies the field components in and normal to the plane of incidence by the proper ground plane reflection coefficient and returns the total field after reflection.

3. METHOD: For a perfect plane, the reflected field is set to the negative of the incident field. For an imperfect ground plane, the incident field is broken into its x, y, and z components. These are modified by the reflection coefficients and returned in the subroutine's calling arguments.

The reflected electric and magnetic fields are given by

$$\bar{E}_R = R_v\bar{E}_I + (R_H - R_v)(\bar{E}_I \cdot \hat{p})\,\hat{p}$$

and

$$\bar{H}_R = R_H\bar{H}_I + (R_v - R_H)(\bar{H}_I \cdot \hat{p})\,\hat{p}$$

where $\bar{E}_I$ and $\bar{H}_I$ are the incident fields reflected in a perfectly conducting ground, or the field due to the image of the source, and $\hat{p}$ is the unit vector normal to the plane of incidence, which is given by $\hat{z} \times \hat{k}$, where $\hat{k}$ is the propagation vector unit normal.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CABJ	The x component of the unit vector along the axis of the wire
FX	The x component of the initial complex electric or magnetic field (For a wire source this is the z component)
FXI	The imaginary part of FX
FXR	The real part of FX
FXY	For electric field $(R_H-R_v)\ (\bar{E}_I \cdot \hat{p})$ and for magnetic field $(R_v-R_H)(\bar{H}_I \cdot \hat{p})$
FY	The y component of the initial complex electric or magnetic field (For a wire source this is the ρ component)

FYI	The imaginary part of FY
FYR	The real part of FY
FZ	The z component of the initial complex electric or magnetic field (for a wire source this term is set to zero)
FZI	The imaginary part of FZ
FZR	The real part of FZ
IPATCH	A flag indicating a patch source
IPERF	A flag indicating a perfect ground
IWIRE	A flag indicating a wire source segment and that a local cylindrical coordinate system is used
PX	The x component of the unit vector normal to the wire axis
PY	The y component of the unit vector normal to the wire axis
REFH	The reflection coefficient for polarization normal to the plane of incidence
REFV	The reflection coefficient for polarization in the plane of incidence
RHOX,RHOY,RHOZ	X,Y, and Z components of the unit vector normal to the wire axis
SABJ	The y component of the unit vector along the axis of the wire
SALPR	The reflected z component of the unit vector along the axis of the wire
TFX,TFY,TFZ	The x,y, and z components of the complex electric or magnetic field upon exit

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
CABJ	/AMPZIJ/
FXI,FXR	F.P.

FYI,FYR	F.P.
FZI,FZR	F.P.
IPATCH	F.P.
IPERF	/AMPZIJ/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
IWIRE	F.P.
PX	/AMPZIJ/
PY	/AMPZIJ/
REFH	/AMPZIJ/
REFV	/AMPZIJ/
RHOX	/AMPZIJ/
RHOY	/AMPZIJ/
RHOZ	/AMPZIJ/
SABJ	/AMPZIJ/
SALPR	/AMPZIJ/

B.

OUTPUT	LOCATION
FXI,FXR	F.P.
FYI,FYR	F.P.
FZI,FZR	F.P.

6. CALLING ROUTINES:

NERFLD

NTRPLT

NTRPLU

ZIJSET

7. CALLED ROUTINES:

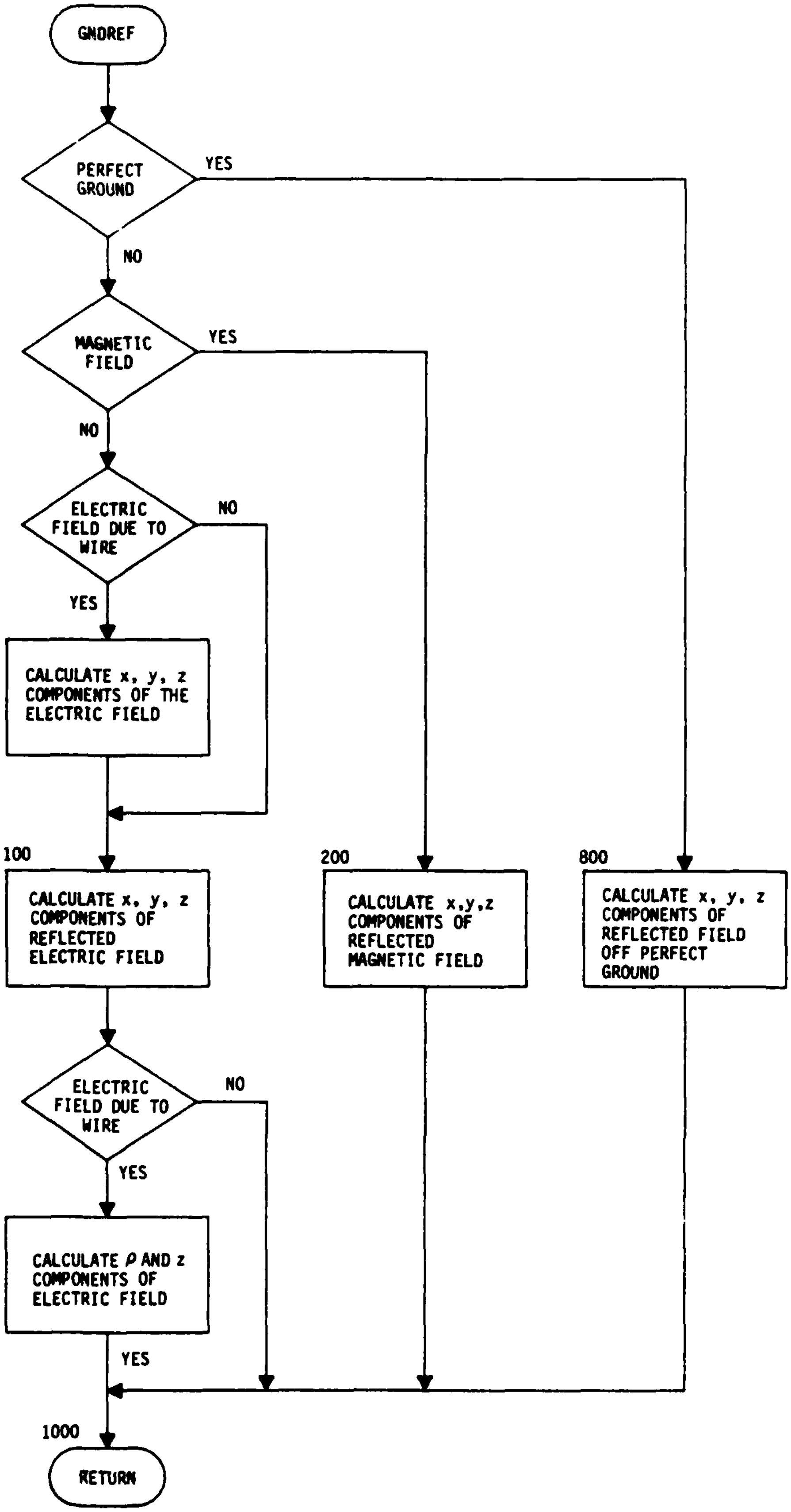
ASSIGN

STATIN

STATOT

WLKBCK

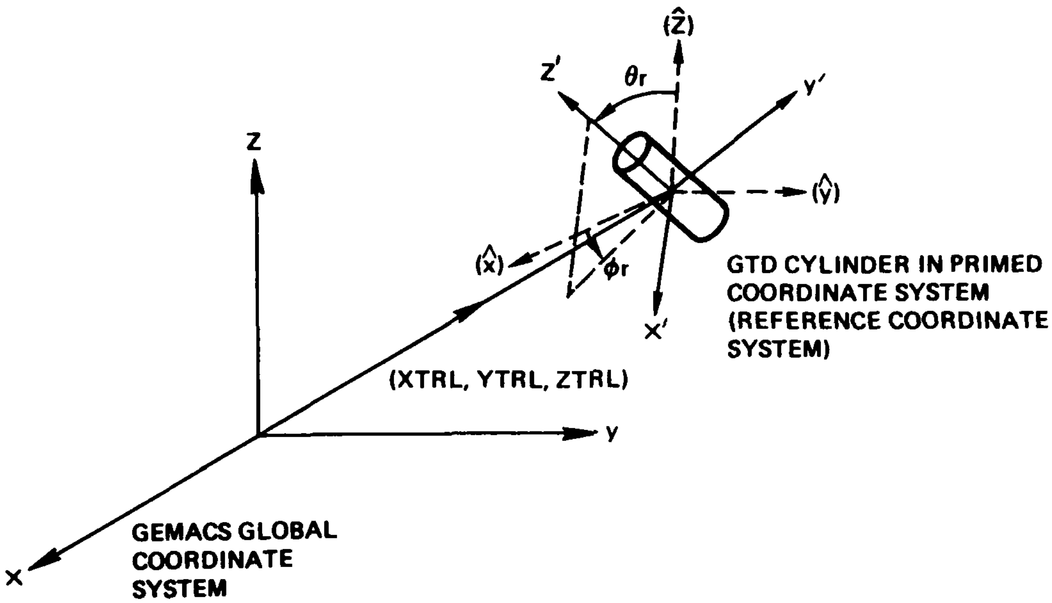
GNDREF (MOM)


GNDREF
PERFECT GROUND
YES
NO
MAGNETIC FIELD
YES
NO
ELECTRIC FIELD DUE TO WIRE
NO
YES
CALCULATE x, y, z COMPONENTS OF THE ELECTRIC FIELD
100
CALCULATE x, y, z COMPONENTS OF REFLECTED ELECTRIC FIELD
200
CALCULATE x,y,z COMPONENTS OF REFLECTED MAGNETIC FIELD
800
CALCULATE x, y, z COMPONENTS OF REFLECTED FIELD OFF PERFECT GROUND
ELECTRIC FIELD DUE TO WIRE
NO
YES
CALCULATE ρ AND z COMPONENTS OF ELECTRIC FIELD
YES
1000
RETURN

1. NAME: GTDCS (INPUT)

2. PURPOSE: To rotate and translate a GTD object as specified by a GEMACS coordinate system number.

3. METHOD: The GEMACS coordinate system table is searched for a match to the coordinate system number specified. The GTD object center point is set to the center of that coordinate system. Then the GTD object z-axis is rotated by the rotation angles of the coordinate system and the equivalent (θ,ϕ) angles are obtained from the rotated z-axis.



4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICSYS	Coordinate system of GTD geometry.
II	Pointer to coordinate system parameters in /CSYSTM/
RHO	X-Y plane projection of z axis unit vector
RHOD2	RHO * RHO

RX,RY,RZ	Coordinate system rotation angles about x-, y-, and z-axis, respectively.
ST	Sin (THETAR)
THETAR,PHIR	Polar and azimuth angles (radians) of GTD object z-axis
XD,YD,ZD	GTD object z-axis unit vector
XTRL,YTRL,ZTRL	GTD object center point after translation.

5. I/O VARIABLES:

A. INPUT — LOCATION

INPUT	LOCATION
CX	/CSYSTM/
CY	/CSYSTM/
CZ	/CSYSTM/
ICSYS	F.P.
IDCSYS	/CSYSTM/
LSTCSY	/CSYSTM/
LUPRNT	/ADEBUG/
ROX	/CSYSTM/
ROY	/CSYSTM/
ROZ	/CSYSTM/
ZERO	/ADEBUG/

B. OUTPUT — LOCATION

OUTPUT	LOCATION
PHIR	F.P.
THETAR	F.P.
XTRL	F.P.
YTRL	F.P.
ZTRL	F.P.

6. CALLING ROUTINES:

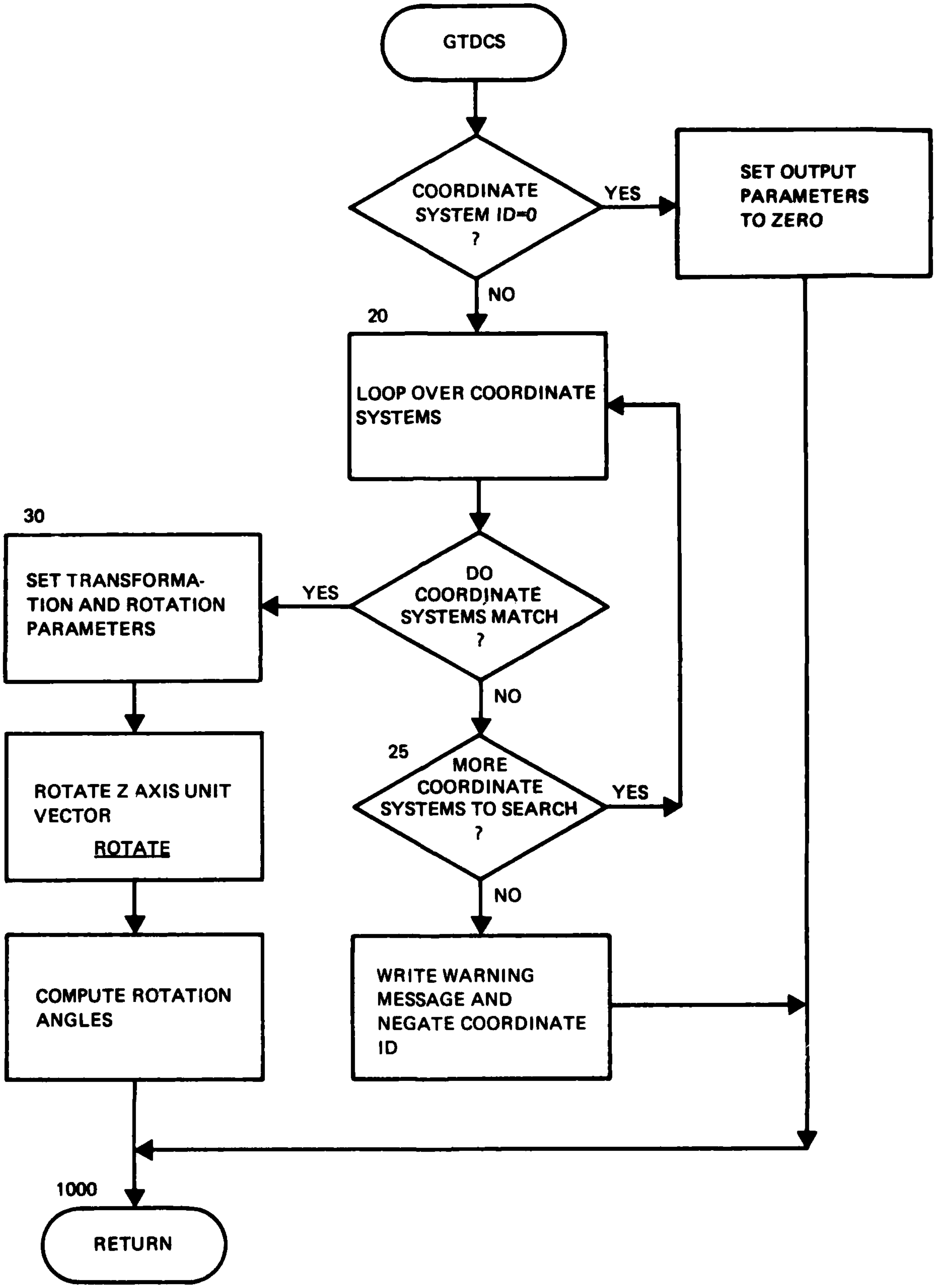
CYLNDR

LNKGTD

7. CALLED ROUTINES:

ASSIGN	STATOT
ROTATE	WLKBCK
STATIN	

GTDCS (INPUT)


GTDCS
COORDINATE SYSTEM ID=0 ?
YES
SET OUTPUT PARAMETERS TO ZERO
NO
20
LOOP OVER COORDINATE SYSTEMS
30
SET TRANSFORMATION AND ROTATION PARAMETERS
YES
DO COORDINATE SYSTEMS MATCH ?
NO
25
MORE COORDINATE SYSTEMS TO SEARCH ?
YES
ROTATE Z AXIS UNIT VECTOR
ROTATE
NO
COMPUTE ROTATION ANGLES
WRITE WARNING MESSAGE AND NEGATE COORDINATE ID
1000
RETURN

1. NAME: GTDDRV (GTD)

2. PURPOSE: This subroutine directs all the geometrical theory of diffraction calculations which determine the electric or magnetic scattered field.

3. METHOD: GTDDRV is the driver routine for all the geometrical theory of diffraction (GTD) calculations. There are three major computations which GTDDRV performs. The first is to initialize fields and special flags. The second is to compute or obtain information pertaining to the geometry, fixed geometry parameters, coordinate systems, and normalizing lengths. These are calculated depending upon whether the source, frequency, or geometry data set name has changed from the last time GTDDRV was called. (The procedure is pointed out in the accompanying flow chart.) Of course, if it is the first time GTDDRV is called, the source, frequency, and geometry name are considered to all be different. The third major calculation is to compute the specified fields. The total field accumulated from each specific field interaction requested and the error flag are the outputs of GTDDRV to its calling routine, ZGTDRV.

   There are also more subtle calculations which occur in this subroutine. For instance, if the geometry data set has been extended, subroutine TSKXQT will call GTDDRV to reset the geometry data set name flag, so that the next time GTDDRV is called to compute fields it will reenter the geometry under the original name, thus including any changes. Another function of GTDDRV is to make sure that the present number of geometrical theory of diffraction objects has not exceeded the allowable number within any one execution of the computer code. If they are exceeded, the error flag is set and control is returned to the calling routine. If the GTD objects are within their allowed limits, no error flag is set and computation continues. Frequency is calculated in megahertz in the calling routine. In GTDDRV the frequency is changed to gigahertz.

   To obtain the GTD geometry from the segment table (if it is different from the last time GTDDRV was called) the code pages through the segment table. The coordinates in the segment table are stored in units of meters. When the GTD objects are obtained, the original segment table block is put back into memory. This subroutine performs extensive checking to determine whether certain computations are required in order to save computation time. For example, if the geometry and frequency are the same as the last time GTDDRV was called, the geometry coordinates will not be renormalized from meters (from the segment table) to wavelengths because they are the same as they were the last time. All geometry lengths in the GTD calculations are in wavelengths. The conversion from meters to wavelengths is made in GTDDRV. The geometry in wavelengths is stored in common blocks.

To determine fixed constants associated with geometry, subroutines GEOM, GEOMC, and GEOMPC are called. GEOM determines fixed constants for plates and is called only if plates are present. GEOMC is called if a cylinder is present to determine the fixed constants associated with the cylinder. GEOMPC determines fixed constants for plate and cylinder interactions if both are present. An error may also occur in subroutine GEOM if a plate is found not to be flat. An error flag is set in that subroutine and when control returns to GTDDRV, the latter subroutine will return control to its calling routine with the fields set at zero.

The field computations are made depending upon the user's requests and are denoted by variables K and J. Case statements go to the appropriate section for either plate, cylinder, or plate-cylinder interactions. Within each of these is another case statement to compute the appropriate fields. Within the plate calculations, all plates and edges are considered; a DO-loop is performed around the called driver routines for each of the specific fields to be calculated. For each specific interaction, the fields are converted into x, y, z components in the reference coordinate system in subroutine XYZFLD. The fields are also accumulated in that subroutine in common block FLDXYZ. After all the specific fields are computed, the total field is rotated back to the original coordinate system of the calling routine and multiplied by the given source excitation. If a near-field analysis has been requested, the fields are converted to units of volts or amps per meter. The far field is already in these units.

At the end of GTDDRV, it is determined if the source is shadowed from the near-field observation point. If it is, a flag is set to indicate this. It is used in other calculations within the GTD module. Control is then returned to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AA	The elliptical cylinder major axis length in meters
AXCL	Test variable to determine if the x axis of the cylinder is in the reference coordinate system
AYCL	Test variable to determine if the y axis of the cylinder is in the reference coordinate system

AZCL	Test variable to determine if the z axis of the cylinder is in the reference coordinate system
BB	The elliptical cylinder minor axis length in meters
CURENT	The source excitation in real and imaginary components
D	Unit vector of the far-field observation direction or the near-field observation point location
DHIT	Distance to hit point (from subroutine PLAINT)
DHT	Distance from source to field point in wavelengths
EDCPH	Phi component of field diffracted from end cap rim in RCS
EDCRPP	Phi component of field diffracted from corners of edge ME of plate MP and then reflected by plate MR (corner diffraction)
EDCRPT	Theta component of field diffracted from the corners of edge ME of plate MP and then reflected by plate MR (corner diffraction)
EDCTH	Theta component of field diffracted from end cap rim in RCS
EDPCPH	Phi component of field diffracted from corners of edge ME of plate MP
EDPCTH	Theta component of field diffracted from corners of edge ME of plate MP
EDPPH	Phi component of field diffracted from edge ME of plate MP in RCS
EDPTH	Theta component of field diffracted from edge ME of plate MP in RCS
EDRCP	Phi component of field diffracted from edge ME of plate MP and reflected from the cylinder

EDRCT	Theta component of field diffracted from edge MP of plate MP and reflected from the cylinder
EDRPP	Phi component of field diffracted from edge ME of plate MP and then reflected by plate MR (edge diffraction)
EDRPT	Theta component of field diffracted from edge ME of plate MP and then reflected by plate MR (edge diffraction)
EIPH	Phi component of direct field from source in RCS
EITH	Theta component of direct field from source in RCS
EPH	The total electric field phi component
ERCAP	Phi component of field reflected from cylinder end cap in RCS
ERCAT	Theta component of field reflected from cylinder end cap in RCS
ERCPP	Phi component of geometrical optics field reflected from cylinder and then reflected from plate MR
ERCPT	Theta component of geometrical optics field reflected from cylinder and then reflected from plate MR
ERDPH	Phi component of field reflected from cylinder and diffracted by edge ME of plate MP
ERDTH	Theta component of field reflected from cylinder and diffracted by edge ME of plate MP
ERPCP	Phi component of geometrical optics field reflected by plate MR and then scattered by the cylinder
ERPCT	Theta component of geometrical optics field reflected by plate MR and then scattered by the cylinder

ERPDCP	Phi component of field reflected by plate MR and diffracted by the corners of edge ME of plate MP (corner diffraction)
ERPDCT	Theta component of field reflected by plate MR and diffracted by the corners of edge ME of plate MP (corner diffraction)
ERPDP	Phi component of field reflected by plate MR and diffracted by edge ME of plate MP (edge diffraction)
ERPDT	Theta component of field reflected by plate MR and diffracted by edge ME of plate MP (edge diffraction)
ERPH	Phi component of geometrical optics field reflected from cylinder
ERPPH	Phi component of field reflected from plate MP in RCS
ERPSP	Phi component of field reflected by plate MR and then scattered by the cylinder
ERPST	Theta component of field reflected by plate MR and then scattered by the cylinder
ERPTH	Theta component of field reflected from plate MP in RCS
ERRPP	Phi component of field reflected from plate MP and then plate MPP in RCS
ERRPT	Theta component of field reflected from plate MP and then plate MPP in RCS
ERSPP	Phi component of field scattered by the cylinder and then reflected by plate MR
ERSPT	Theta component of field scattered by the cylinder and then reflected by plate MR
ERTH	Theta component of geometrical optics field reflected from cylinder
ESPH	Phi component of field scattered by cylinder in RCS

ESTH	Theta component of field scattered by cylinder in RCS
ETH	The total electric field theta component
FLDPT	The location of the near-field observation point normalized to wavelengths or the far-field observation direction
FN	The wedge angle number for the particular plate and edge under consideration
FP1	The x component in meters of the near-field observation point or zero for far-field calculations in the global coordinate system
FP2	The y component in meters of the near-field observation point or the theta angle in radians of the far-field observation direction in the global coordinate system
FP3	The z component in meters of the near-field observation point or the phi angle in radians of the far-field observation direction in the global coordinate system
FRQG	The frequency in gigahertz
FRQGLA	This is the frequency in gigahertz which will be used the next time GTDDRV is called to check if the frequency has remained the same
FRQMHZ	Frequency in megahertz
FX	The x component of the electric or magnetic field in the reference coordinate system
FY	The y component of the electric or magnetic field in the reference coordinate system
FZ	The z component of the electric or magnetic field in the reference coordinate system
IANG	The number of angles to be considered
IBSCEO	A flag which is used to indicate that an error occurred in the GTD calculations (0

	indicates no error, 1 indicates error occurred)
ICSYS	The coordinate system number in which the cylinder was defined
IEHO	Indicates the field type wanted (1 for E-field, 0 for H-field)
IFDTYP	Indicates the type of parameters being sent to GTDDRV for the field (1 for near-field, 0 for far-field)
IGDNLA	This is the geometry data set name that will be used the next time GTDDRV is called to check if the geometry has been changed
IGLIM	The last line in this SEGTBL block that contains GTD object information
IGTDGM	The geometry data set name
ILINE	The line of the SEGTBL being considered
IMSRC	The source type
INPBLK	The SEGTBL block in which GTD objects begin
ISDWFL	A flag which indicates that the source is shadowed from the observation point or direction (0 indicates not shadowed, 1 indicates shadowed)
ITAG	GTD object type identifier
ITEM	Location of a particular GTD object parameter on the present ILINE of the SEGTBL array
J	Used to indicate which particular reflection--diffraction interactions were requested for the particular K interactions
K	The variable used to indicate which plate, cylinder, or plate-cylinder GTD interactions were requested
KJINT	This array denotes the scattering interactions requested

LFRQFL	This is a logical variable set true to indicate if the frequency changed from the last time GTDDRV was called
LGDNFL	This is a logical variable set true if the geometry data set name changed since the last time GTDDRV was called
LHIT	Logical variable (from subroutine PLAINT) set true if the near-field observation point is shadowed from the source
LNROT	A logical variable set true if the reference coordinate system and the global system are the same
LOC	The location of a specific K, J interaction in the KJINT array
LSRCFO	A logical variable set true to indicate if the source is the same as the last source when GTDDRV was called
MC	An index variable used to indicate which end cap is being considered
ME	An index variable used to specify a particular edge on a plate
MP	An index variable used to indicate a specific plate number
NCORN	The number of corners on a particular plate
NUM	The total number of GTD objects
ORIGIN	This variable represents the (0., 0., 0.) location. It is used as a dummy variable when referring the source axes to the reference coordinate system because these axes only need to be rotated and not translated
PWSRC	Plane wave to spherical wave conversion constant for incident electric field

SP1	The wire source radius in wavelengths or the patch area in square wavelengths
SP2	The wire source length in wavelengths or zero for patches
SSP1	The wire source radius in meters, or the patch area in square meters
SSP2	The wire source length in meters, or for patches it is zero
THTN	The theta angle from the z axis to the end cap normal of the more negative end cap (measured in the x-z plane)
THTNR	The angle between the z axis and the end cap plane of the more negative end cap (measured in the x-z plane)
THTP	The theta angle from the z axis to the end cap normal of the more positive end cap (measured in the x-z plane)
THTPR	The angle between the z axis and the end cap plane of the more positive end cap (measured in the x-z plane)
TR	The location of the cylinder center in wavelengths
TRO	The location of the center of the cylinder in the global coordinate system in meters
TXS	Direction cosine of the source tangent from the x axis
TYS	Direction cosine of the source tangent from the y axis
TZS	Direction cosine of the source tangent from the z axis
VXSS	The source axes unit vectors in the global coordinate system
WI	The complex source excitation

WL	The wavelength
XCSRC	X location of the source in meters in the global coordinate system
XFLD	The x component of the output electric or magnetic field in the global coordinate system in volts/meter for E-field or amps/meter for H-field
XNS	Direction cosine of the source normal from the x axis (used for patches)
XSS	The source location in meters in the global coordinate system
XX	The plate corner locations in meters as stored in SEGTBL in the global coordinate system
XXX	A temporary storage variable used for translating and rotating the geometry from the global coordinate system to the reference coordinate system (RCS)
YCSRC	Y location of the source in meters in the global coordinate system
YFLD	The y component of the output electric or magnetic field in the global coordinate system in volts/meter for E-field or amps/meter for H-field
YNS	Direction cosine of the source normal from the y axis (used for patches)
ZCN	The distance from the center of the cylinder to the point at which the z axis pierces the more negative end cap
ZCP	The distance between the center of the cylinder and the point at which the z axis pierces the more positive end cap
ZCSRC	Z location of the source in meters in the global coordinate system
ZFLD	The z component of the output electric or magnetic field in the global coordinate system in volts/meter for E-field or amps/meter for H-field

ZLOC	The distance from the center of the cylinder to the point at which the z axis pierces the end cap
ZNS	Direction cosine of the source normal from the z axis (used for patches)

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
CURENT	F.P.
DBGPRT	/ADEBUG/
FACTOR	/SOURSF/
FNP	/FNANG/
FP1	F.P.
FP2	F.P.
FP3	F.P.
FRQMHZ	F.P.
FX	/FLDXYZ/
FY	/FLDXYZ/
FZ	/FLDXYZ/
IEHO	F.P.
IFDTYP	F.P.
IGTDGM	F.P.
IMSRC	F.P.
IP217	/GEODAT/
ISDWFL	F.P.
ISGTBL	/SEGMNT/
KJINT	F.P.

LDC	/LDCBY/
LIHD	/LSHDT/
LSHD	/LSHDT/
LSRCFO	F.P.
LSRFC	/SRFACC/
LSURF	/SURFAC/
MAXSEG	/SEGMNT/
NDXBLK	/SEGMNT/
NPATCH	/SEGMNT/
NUMCYL	/GTDDAT/
NUMECP	/GTDDAT/
NUMPLT	/GTDDAT/
NUMSEG	/SEGMNT/
NWIRE	/SEGMNT/
PI	/PIS/
RPD	/PIS/
SEGTBL	/SEGMNT/
SSP1	F.P.
SSP2	F.P.
TXS	F.P.
TYS	F.P.
TZS	F.P.
XCL	/ROTRDT/
XCSRC	F.P.
XNS	F.P.

	YCL	/ROTRDT/
	YCSRC	F.P.
	YNS	F.P.
	ZCL	/ROTRDT/
	ZCSRC	F.P.
	ZNS	F.P.
B.	OUTPUT	LOCATION
	A	/GEOMEL/
	AS	/GTD/
	B	/GEOMEL/
	CAS	/GTD/
	CNC	/GEOMEL/
	CPS	/DIR/
	CTC	/GEOMEL/
	CTHS	/DIR/
	D	/DIR/
	DP	/THPHUV/
	DT	/THPHUV/
	FACTOR	/SOURSF/
	FLDPT	/NEAR/
	FX	/FLDXYZ/
	FY	/FLDXYZ/
	FZ	/FLDXYZ/
	IANG	/DOUBLE/

IBSCEO	F.P.
IEH	/EHFLD/
IM	/SRC/
LCORNR	/LOGDIF/
LCYL	/LPLCY/
LDEBUG	/TEST/
LFRQFL	/SAME/
LGDNFL	/SAME/
LGRND	/GROUND/
LNRFLD	/NEAR/
LPLA	/LPLCY/
LSLOPE	/LOGDIF/
LSRCFL	/SAME/
MEP	/GEOPLA/
MPX	/GEOPLA/
MPXR	/GROUND/
PHSR	/DIR/
SAS	/GTD/
SASP	/GTD/
SNC	/GEOMEL/
SP1	/SRC/
SP2	/SRC/
SPS	/DIR/
STHS	/DIR/
THSR	/DIR/

VXS	/SORINF/
WL	/OUTPTD/
X	/GEOPLA/
XCL	/ROTRDT/
XFLD	F.P.
XS	/SORINF/
YCL	/ROTRDT/
YFLD	F.P.
ZC	/GEOMEL/
ZCL	/ROTRDT/
ZFLD	F.P.

6. CALLING ROUTINES:

TSKXQT

ZGTDRV

7. CALLED ROUTINES:

ASSIGN

BTAN2

CYAXIS

CYLINT

DIFPLT

DPLRCL

DPLRPL

ENDIF

GEOM

GEOMC

GTDDRV (GTD)

GEOMPC

GETSEG

INCFLD

NFD

PLAINT

RCLDPL

REFCAP

REFPLA

ROTRAN

RPLDPL

RPLRPL

RPLSCL

SCLRPL

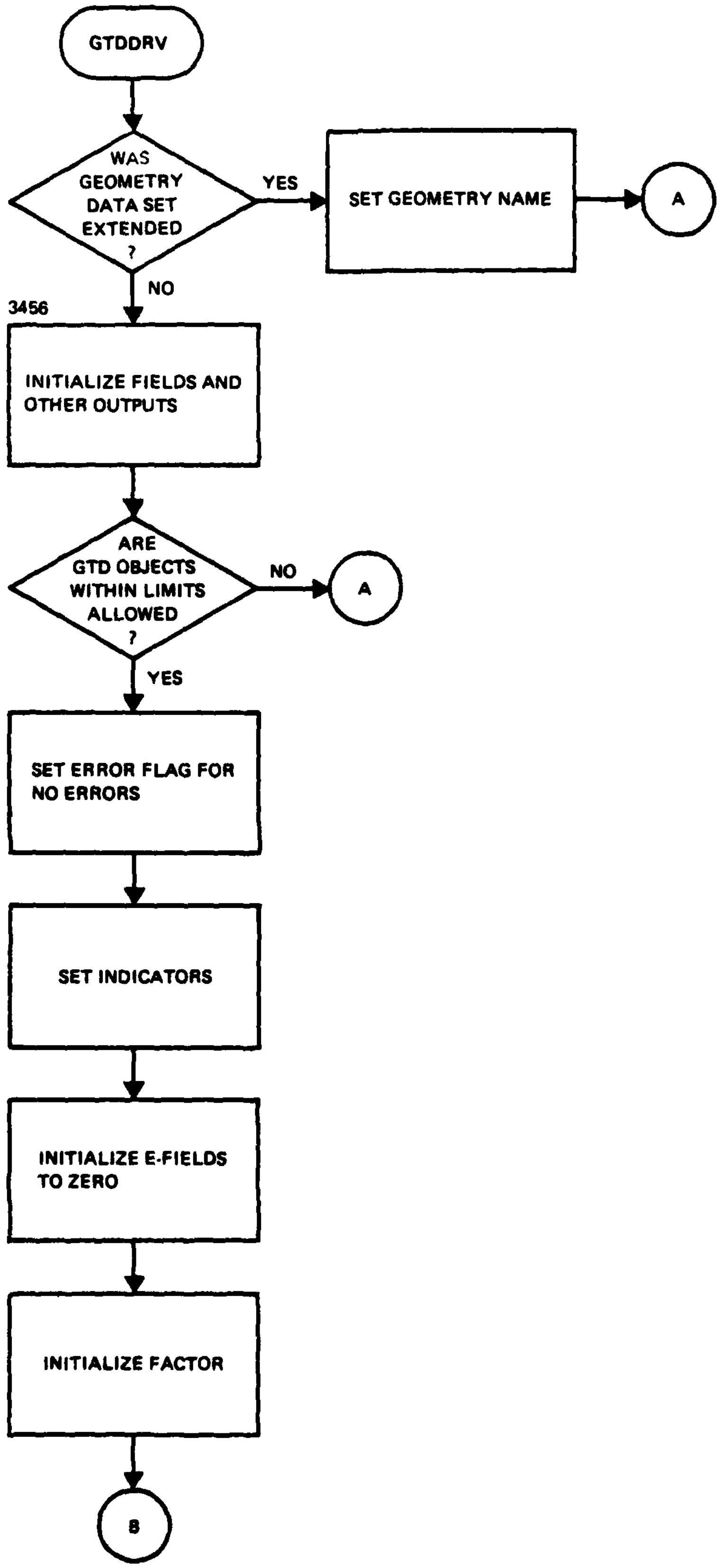
SCTCYL

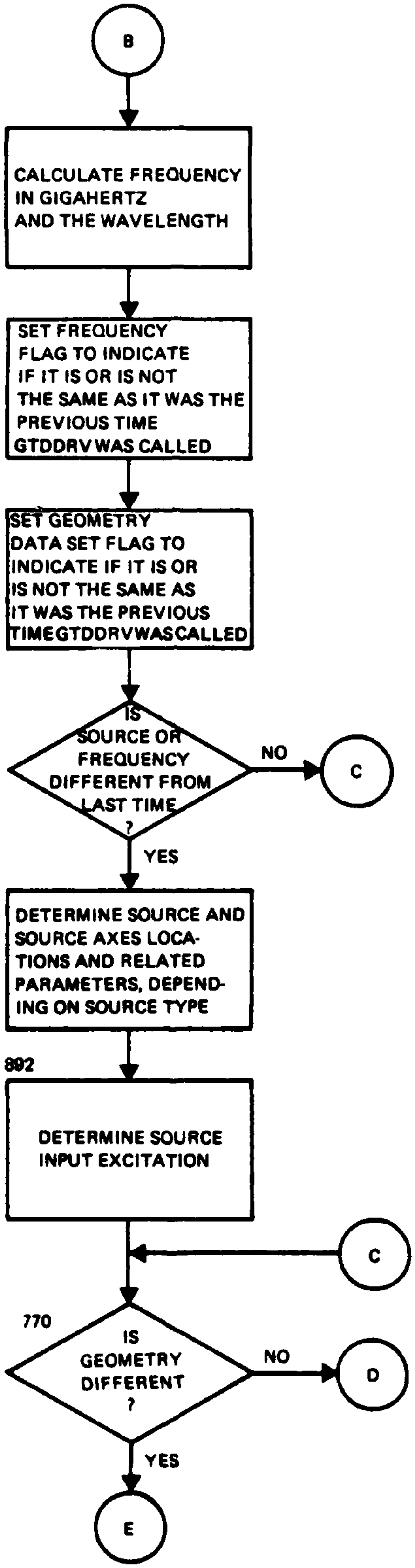
SMAGNF

STATIN

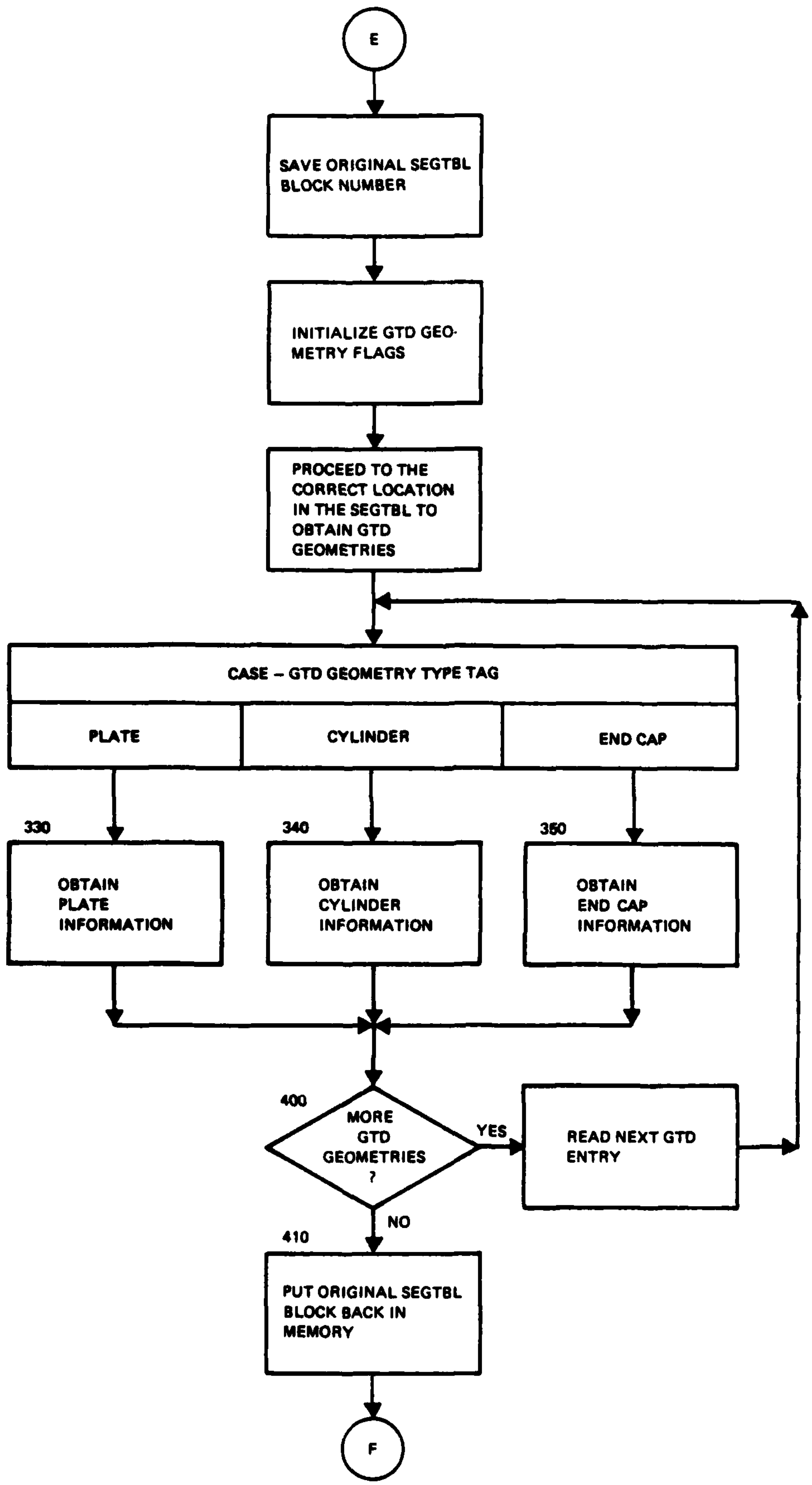
STATOT

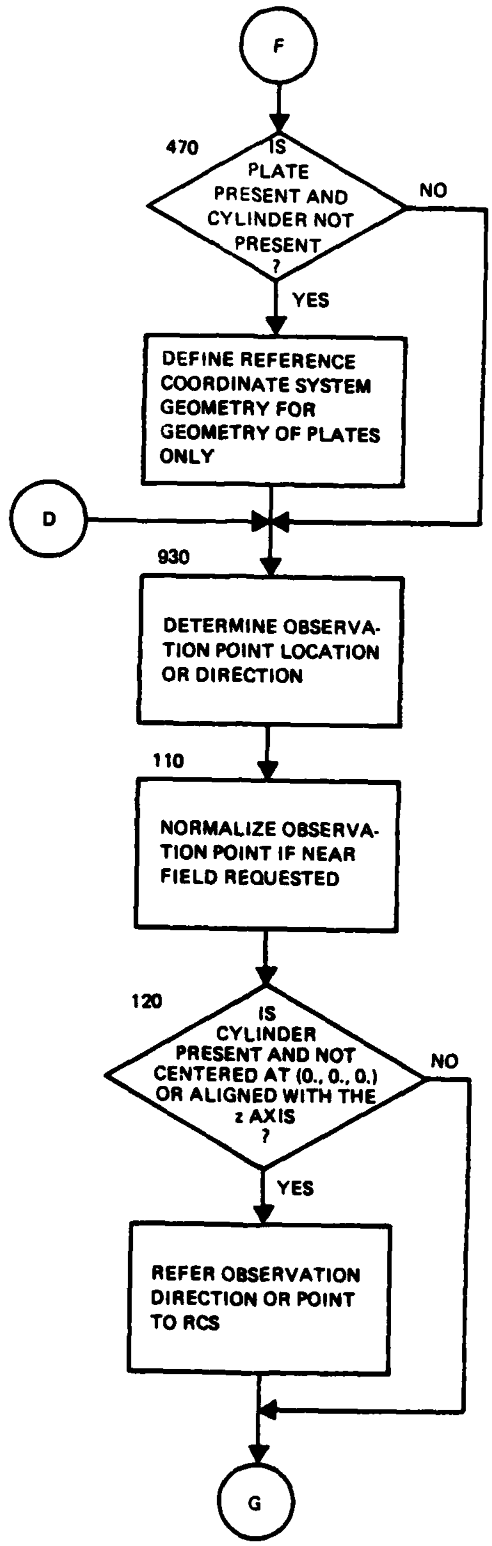
WLKBCK

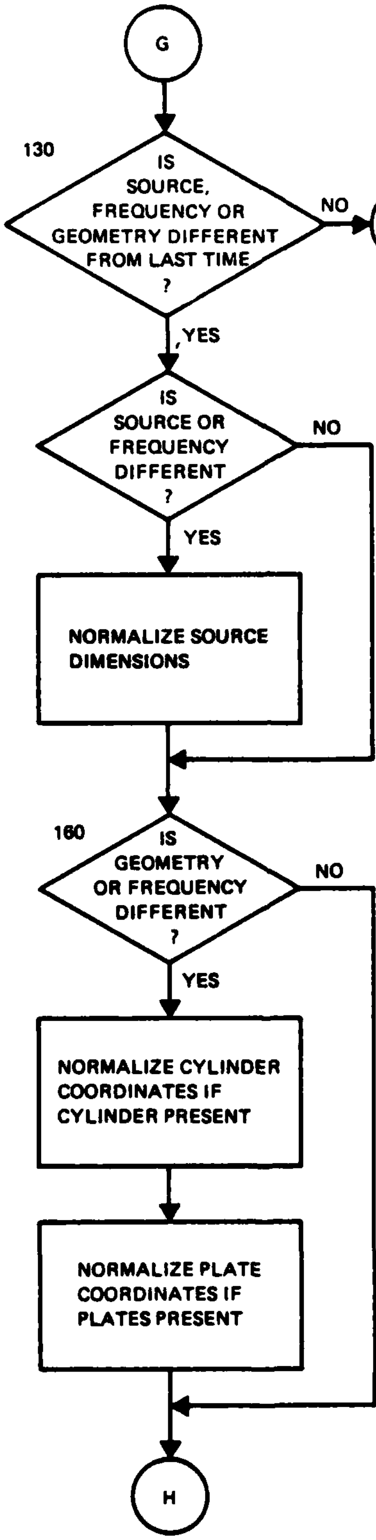

GTDDRV
WAS GEOMETRY DATA SET EXTENDED ?
YES
SET GEOMETRY NAME
A
NO
3456
INITIALIZE FIELDS AND OTHER OUTPUTS
ARE GTD OBJECTS WITHIN LIMITS ALLOWED ?
NO
A
YES
SET ERROR FLAG FOR NO ERRORS
SET INDICATORS
INITIALIZE E-FIELDS TO ZERO
INITIALIZE FACTOR
B

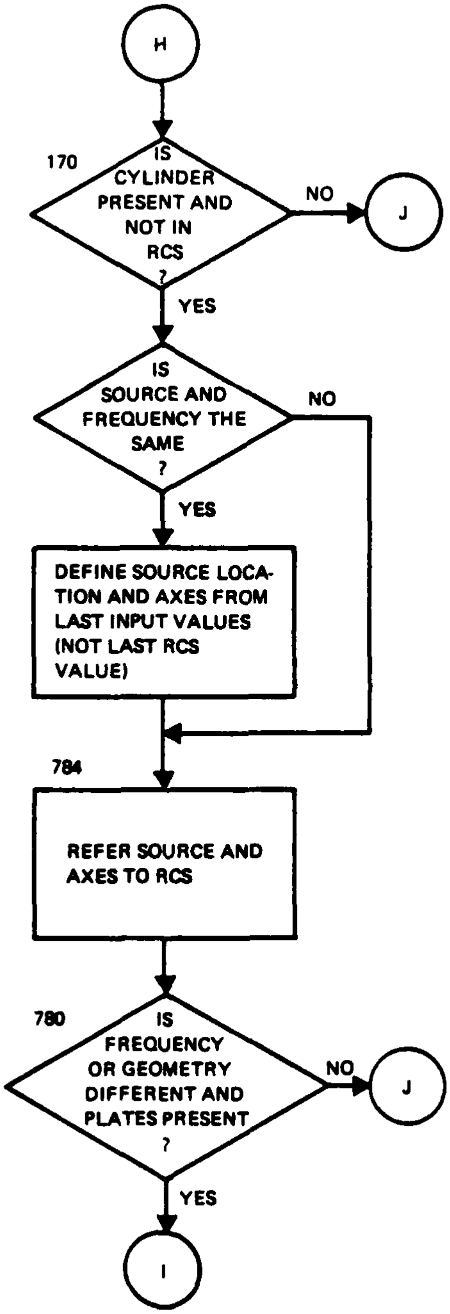

B
CALCULATE FREQUENCY IN GIGAHERTZ AND THE WAVELENGTH
SET FREQUENCY FLAG TO INDICATE IF IT IS OR IS NOT THE SAME AS IT WAS THE PREVIOUS TIME GTDDRV WAS CALLED
SET GEOMETRY DATA SET FLAG TO INDICATE IF IT IS OR IS NOT THE SAME AS IT WAS THE PREVIOUS TIME GTDDRV WAS CALLED
IS SOURCE OR FREQUENCY DIFFERENT FROM LAST TIME ?
NO
C
YES
DETERMINE SOURCE AND SOURCE AXES LOCATIONS AND RELATED PARAMETERS, DEPENDING ON SOURCE TYPE
892
DETERMINE SOURCE INPUT EXCITATION
C
770
IS GEOMETRY DIFFERENT ?
NO
D
YES
E

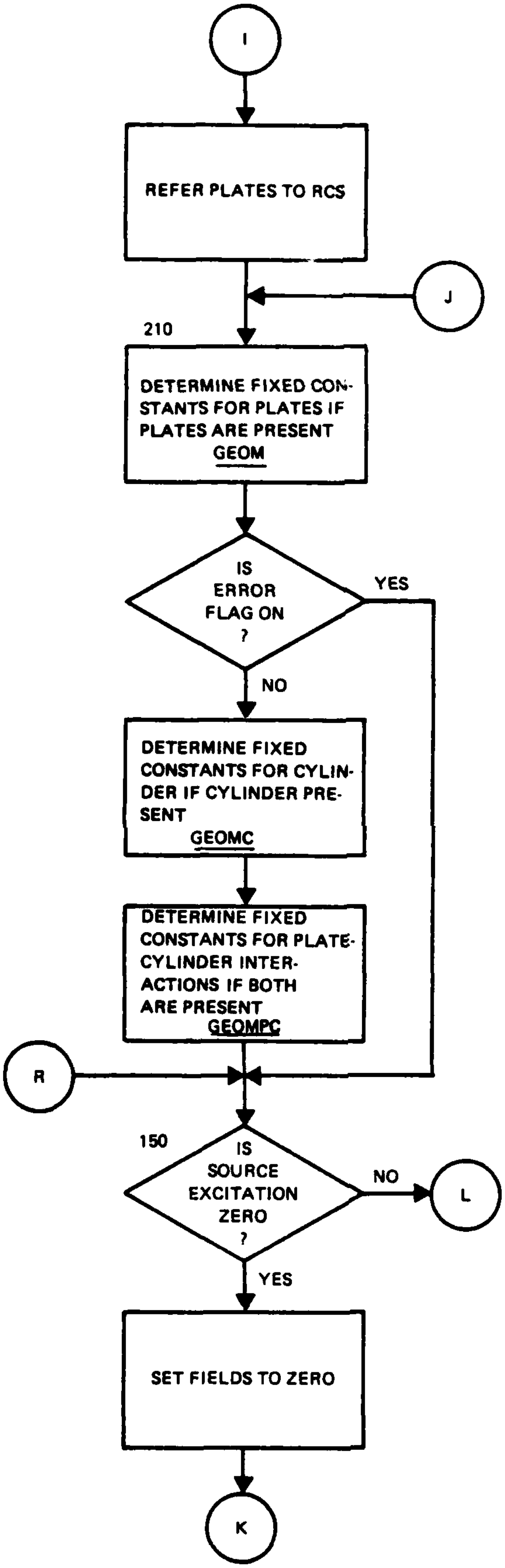
GTDDRV (GTD)

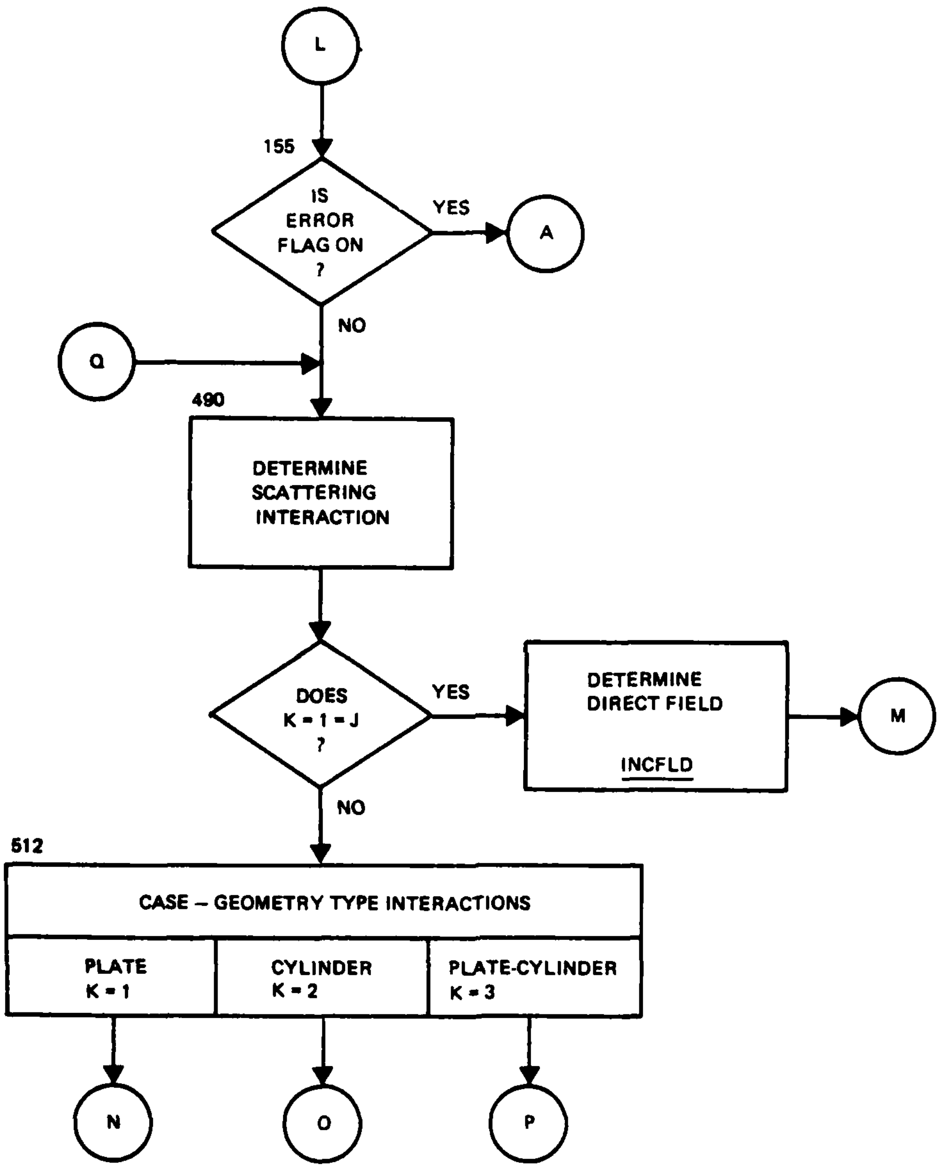

E
SAVE ORIGINAL SEGTBL BLOCK NUMBER
INITIALIZE GTD GEOMETRY FLAGS
PROCEED TO THE CORRECT LOCATION IN THE SEGTBL TO OBTAIN GTD GEOMETRIES
CASE – GTD GEOMETRY TYPE TAG
PLATE
CYLINDER
END CAP
330
OBTAIN PLATE INFORMATION
340
OBTAIN CYLINDER INFORMATION
350
OBTAIN END CAP INFORMATION
400
MORE GTD GEOMETRIES ?
YES
READ NEXT GTD ENTRY
NO
410
PUT ORIGINAL SEGTBL BLOCK BACK IN MEMORY
F

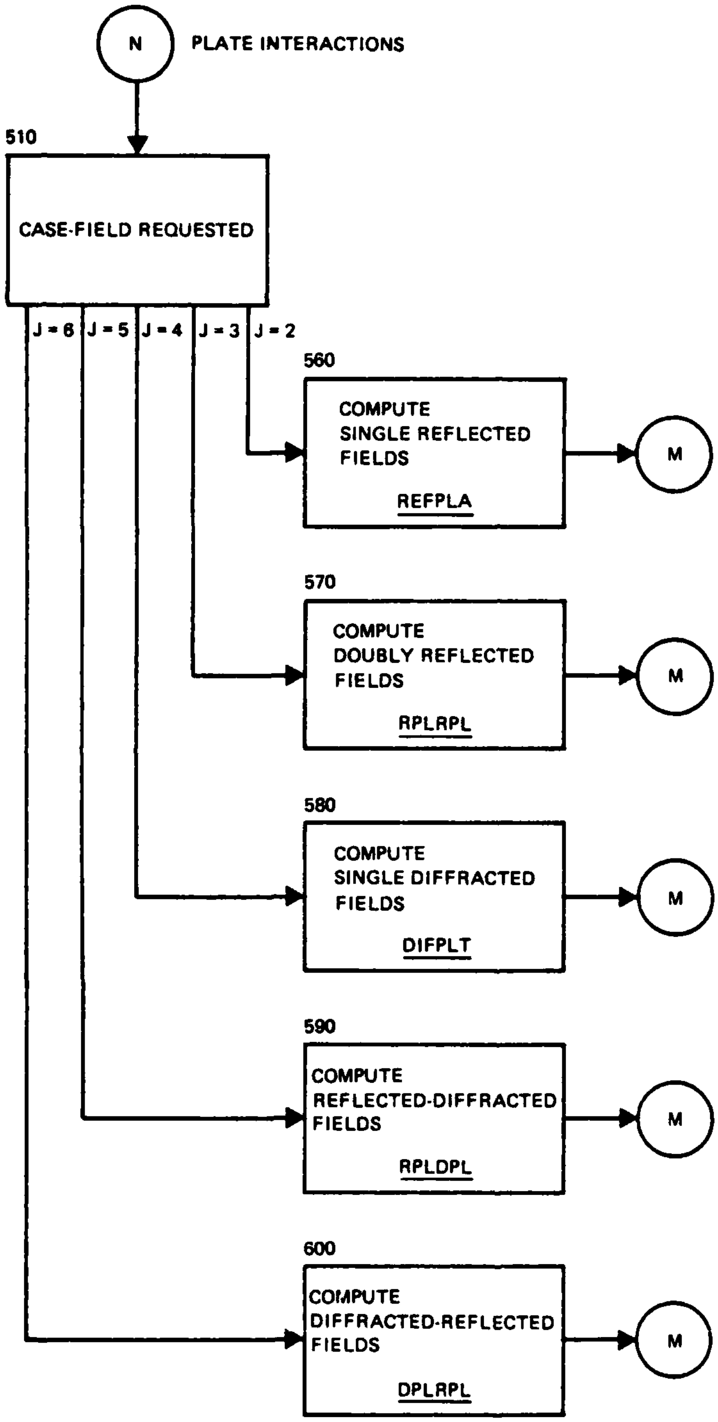

F
470
IS PLATE PRESENT AND CYLINDER NOT PRESENT ?
NO
YES
DEFINE REFERENCE COORDINATE SYSTEM GEOMETRY FOR GEOMETRY OF PLATES ONLY
D
930
DETERMINE OBSERVATION POINT LOCATION OR DIRECTION
110
NORMALIZE OBSERVATION POINT IF NEAR FIELD REQUESTED
120
IS CYLINDER PRESENT AND NOT CENTERED AT (0., 0., 0.) OR ALIGNED WITH THE z AXIS ?
NO
YES
REFER OBSERVATION DIRECTION OR POINT TO RCS
G

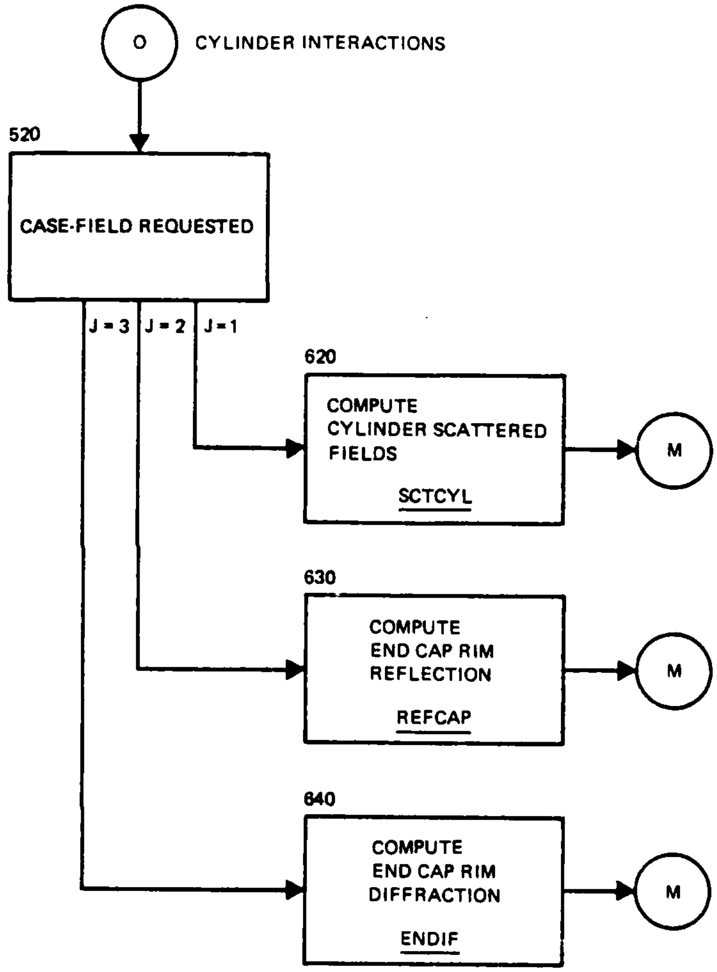

G
130
IS SOURCE, FREQUENCY OR GEOMETRY DIFFERENT FROM LAST TIME ?
NO
R
YES
IS SOURCE OR FREQUENCY DIFFERENT ?
NO
YES
NORMALIZE SOURCE DIMENSIONS
160
IS GEOMETRY OR FREQUENCY DIFFERENT ?
NO
YES
NORMALIZE CYLINDER COORDINATES IF CYLINDER PRESENT
NORMALIZE PLATE COORDINATES IF PLATES PRESENT
H

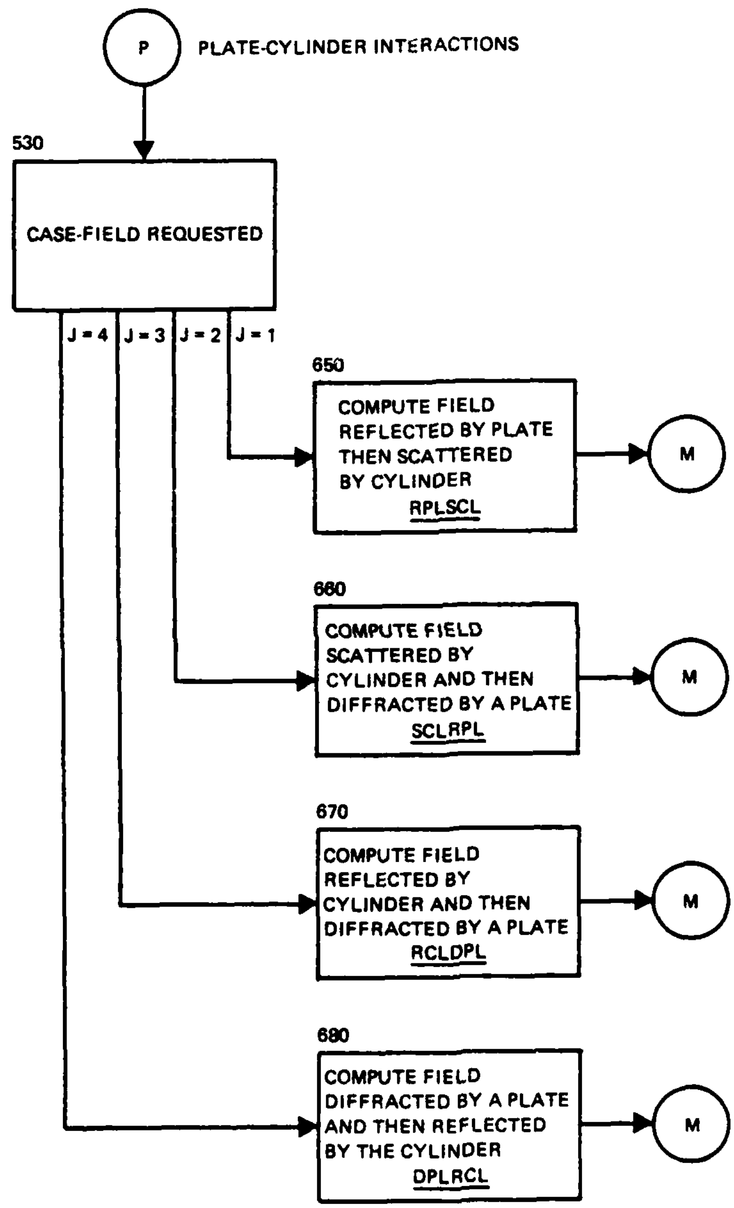

H
170
IS CYLINDER PRESENT AND NOT IN RCS ?
NO
J
YES
IS SOURCE AND FREQUENCY THE SAME ?
NO
YES
DEFINE SOURCE LOCATION AND AXES FROM LAST INPUT VALUES (NOT LAST RCS VALUE)
784
REFER SOURCE AND AXES TO RCS
780
IS FREQUENCY OR GEOMETRY DIFFERENT AND PLATES PRESENT ?
NO
J
YES
I

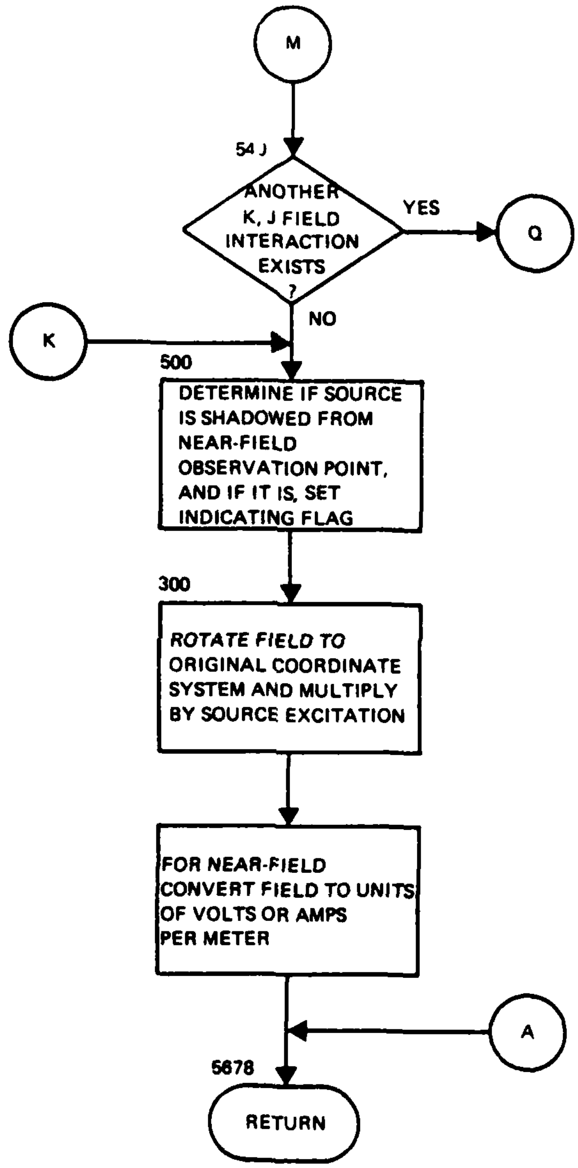

I
REFER PLATES TO RCS
J
210
DETERMINE FIXED CONSTANTS FOR PLATES IF PLATES ARE PRESENT
GEOM
IS ERROR FLAG ON ?
YES
NO
DETERMINE FIXED CONSTANTS FOR CYLINDER IF CYLINDER PRESENT
GEOMC
DETERMINE FIXED CONSTANTS FOR PLATE-CYLINDER INTERACTIONS IF BOTH ARE PRESENT
GEOMPC
R
150
IS SOURCE EXCITATION ZERO ?
NO
L
YES
SET FIELDS TO ZERO
K


L
155
IS ERROR FLAG ON ?
YES
A
NO
Q
490
DETERMINE SCATTERING INTERACTION
DOES K = 1 = J ?
YES
DETERMINE DIRECT FIELD
INCFLD
M
NO
512
CASE – GEOMETRY TYPE INTERACTIONS
PLATE K = 1
CYLINDER K = 2
PLATE-CYLINDER K = 3
N
O
P


N
PLATE INTERACTIONS
510
CASE-FIELD REQUESTED
J = 6
J = 5
J = 4
J = 3
J = 2
560
COMPUTE SINGLE REFLECTED FIELDS
REFPLA
M
570
COMPUTE DOUBLY REFLECTED FIELDS
RPLRPL
M
580
COMPUTE SINGLE DIFFRACTED FIELDS
DIFPLT
M
590
COMPUTE REFLECTED-DIFFRACTED FIELDS
RPLDPL
M
600
COMPUTE DIFFRACTED-REFLECTED FIELDS
DPLRPL
M


O
CYLINDER INTERACTIONS
520
CASE-FIELD REQUESTED
J = 3
J = 2
J = 1
620
COMPUTE CYLINDER SCATTERED FIELDS
SCTCYL
M
630
COMPUTE END CAP RIM REFLECTION
REFCAP
M
640
COMPUTE END CAP RIM DIFFRACTION
ENDIF
M


P
PLATE-CYLINDER INTERACTIONS
530
CASE-FIELD REQUESTED
J = 4
J = 3
J = 2
J = 1
650
COMPUTE FIELD REFLECTED BY PLATE THEN SCATTERED BY CYLINDER
RPLSCL
M
660
COMPUTE FIELD SCATTERED BY CYLINDER AND THEN DIFFRACTED BY A PLATE
SCLRPL
M
670
COMPUTE FIELD REFLECTED BY CYLINDER AND THEN DIFFRACTED BY A PLATE
RCLDPL
M
680
COMPUTE FIELD DIFFRACTED BY A PLATE AND THEN REFLECTED BY THE CYLINDER
DPLRCL
M


M
54 J
ANOTHER K, J FIELD INTERACTION EXISTS ?
YES
Q
NO
K
500
DETERMINE IF SOURCE IS SHADOWED FROM NEAR-FIELD OBSERVATION POINT, AND IF IT IS, SET INDICATING FLAG
300
ROTATE FIELD TO ORIGINAL COORDINATE SYSTEM AND MULTIPLY BY SOURCE EXCITATION
FOR NEAR-FIELD CONVERT FIELD TO UNITS OF VOLTS OR AMPS PER METER
A
5678
RETURN

1. NAME: IBITCK (GTD, INPUT, MOM, OUTPUT)

2. PURPOSE: To determine if a specified bit is set in a calling argument.

3. METHOD: A bit set is determined by taking the modulus base 2 of the input argument. If the result is even, the bit is not set; if the result is odd, the bit is set.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
FIRST	Logical variable TRUE on first entry into subroutine, FALSE on successive entries
IBIT	Input argument designating bit to be tested
IPWR2	Integer 2 to the power minus 1 of the bit position to be tested
IWRD	Input argument of word to be tested
JBIT	Pointer to the IPWR2 array
JWRD	The input word IWRD arithmetically right-shifted to have the bit IBIT as the rightmost bit
MANTSA	The number of bits in the mantissa of a floating point word of the host machine

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
IBIT	F.P.
IWRD	F.P.
MANTSA	/ADEBUG/

B. OUTPUT

OUTPUT	LOCATION
IBITCK	FUNCTION

6. CALLING ROUTINES*:

BACSUB (3)	LUDDRV (3)
BANDIT (3)	PLTDRV (1)
DMPDRV (1,2,3,4)	PRTSYM (3)
EFDGEO (1)	PUTSYM (1,2,3,4)
EGFMAT (3)	REBLCK (3)
EXCDRV (2,3)	RWFILS (1,2,3,4)
FLDDRV (2,3)	SETDRV (3)
FLDOUT (4)	SOLDRV (3)
FNDREC (1,2,3,4)	SYMDEF (1,2,3,4)
GETSYM (1,2,3,4)	

7. CALLED ROUTINES:

ASSIGN

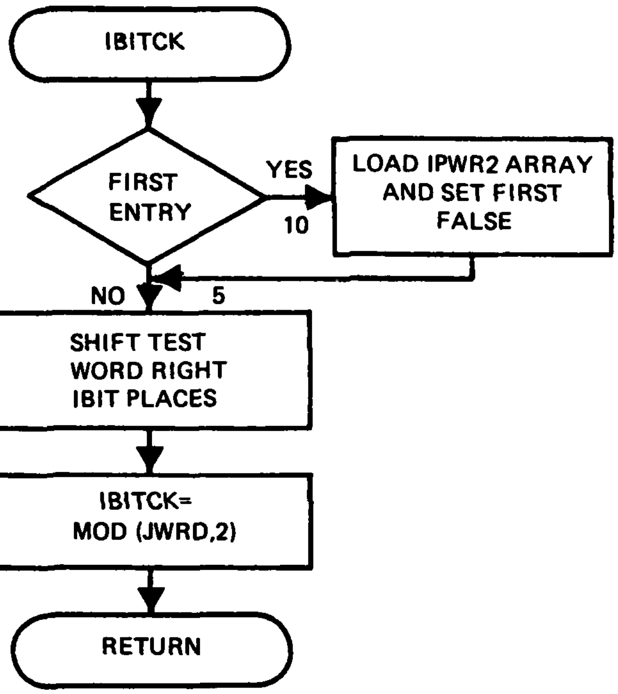
STATIN

STATOT

WLKBCK

*1-INPUT
2-GTD
3-MOM
4-OUTPUT

IBITCK (GTD, INPUT, MOM, OUTPUT)



1. NAME: IMAGE (GTD)

2. PURPOSE: To determine location of source image for reflection of source ray from plate MP. Double reflection image locations may be obtained by calling IMAGE twice: once for the source ray reflection from plate MP, and once for the reflection of the ray from the image location from the second plate.

3. METHOD: The source image location is given by

$$\overline{XII} = \overline{XIS} + 2\ AN\ \hat{VN} = \hat{x}\ XII(1) + \hat{y}\ XII(2) + \hat{z}\ XII(3)$$

where

$$AN = (\overline{X} - \overline{XIS}) \cdot \hat{VN}$$

and $\overline{X}$, $\overline{XIS}$, and $\hat{VN}$ are as follows and as shown in figure 1.

$$\hat{VN} = \text{plate unit normal} = \hat{x}\ VN(MP,1) + \hat{y}\ VN(MP,2) + \hat{z}\ VN(MP,3)$$

$$\overline{XIS} = \hat{x}\ XIS(1) + \hat{y}\ XIS(2) + \hat{z}\ XIS(3)$$

$$\overline{X} = \hat{x}\ X(MP,1,1) + \hat{y}\ X(MP,1,2) + \hat{z}\ X(MP,1,3)$$

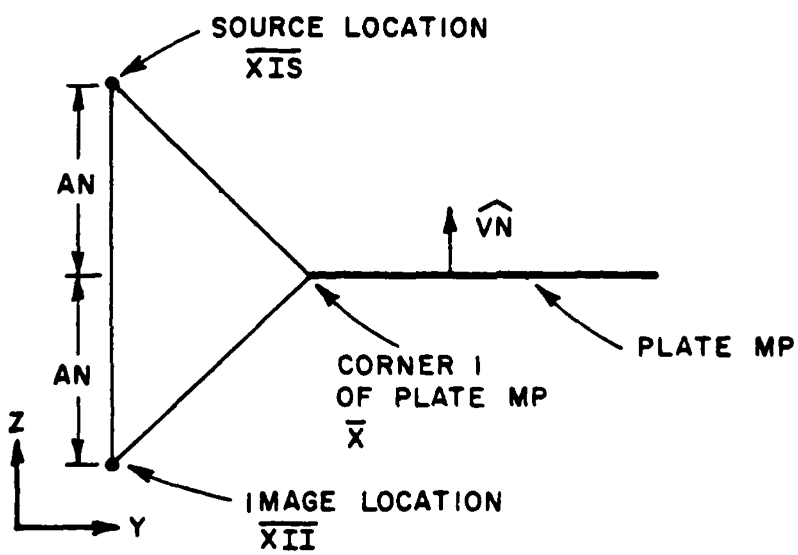


Figure 1. Geometry for Determining Source Image Location

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN	Distance from plate plane to source, calculated from the dot product of vector from source to edge one of plate MP and the plate unit normal (negative AN indicates source lies on the side of the plane into which VN points)
MP	Plate into which source is imaged
VN	Array which includes the normal vector of plate MP
X	Array which includes the corner locations of plate MP
XII	X,Y,Z components of image location in RCS
XIS	X,Y,Z components of source location in RCS

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
MP	F.P.
VN	/GEOPLA/
X	/GEOPLA/
XIS	F.P.

B. OUTPUT

OUTPUT	LOCATION
AN	F.P.
XII	F.P.

6. CALLING ROUTINES:

DPLRPL

GEOM

RCLRPL

SCLRPL

7. CALLED ROUTINES:

NONE

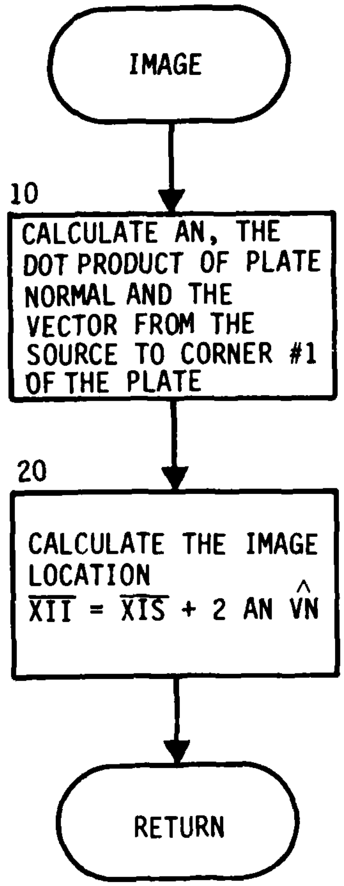


IMAGE
10
CALCULATE AN, THE DOT PRODUCT OF PLATE NORMAL AND THE VECTOR FROM THE SOURCE TO CORNER #1 OF THE PLATE
20
CALCULATE THE IMAGE LOCATION
$\overline{XII} = \overline{XIS} + 2\ AN\ \hat{VN}$
RETURN

1. NAME: IMCDIR (GTD)

2. PURPOSE: To determine the source image axes directions for a source after reflection from a given end cap.

3. METHOD: The source image axes unit vectors for an electric source imaged in an end cap are given by:

$$\hat{x}_{pi} = (-\hat{x}_p \cdot \hat{n})\hat{n} + (\hat{x}_p \cdot \hat{t})\hat{t} + (\hat{x}_p \cdot \hat{b})\hat{b}$$

$$\hat{z}_{pi} = (\hat{z}_p \cdot \hat{n})\hat{n} + (-\hat{z}_p \cdot \hat{t})\hat{t} + (-\hat{z}_p \cdot \hat{b})\hat{b}$$

$$\hat{y}_{pi} = \left(\hat{z}_{pi}\right) \times \left(\hat{x}_{pi}\right)$$

where $\hat{x}_p$, $\hat{y}_p$, and $\hat{z}_p$ are unit vectors of the source coordinate system axes, $\hat{x}_{pi}$, $\hat{y}_{pi}$, $\hat{z}_{pi}$ are the unit vectors of the source image coordinate system for the end cap, and $\hat{n}$ is the end cap unit normal and $\hat{t}$ is the unit vector tangent to the end cap. Illustrations of imaging the source axes and the end cap coordinate system are shown in figures 1 and 2.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
K	Index variable
L	Index variable
LL	Index variable
PB	Dot product of end cap unit binormal and unit vector of source axis being imaged
PN	Dot product of end cap unit normal and unit vector of source axis being imaged
PT	Dot product of end cap unit tangent and unit vector of source axis being imaged
VC	X,Y,Z components of unit vector tangent to end cap (in x-z plane)

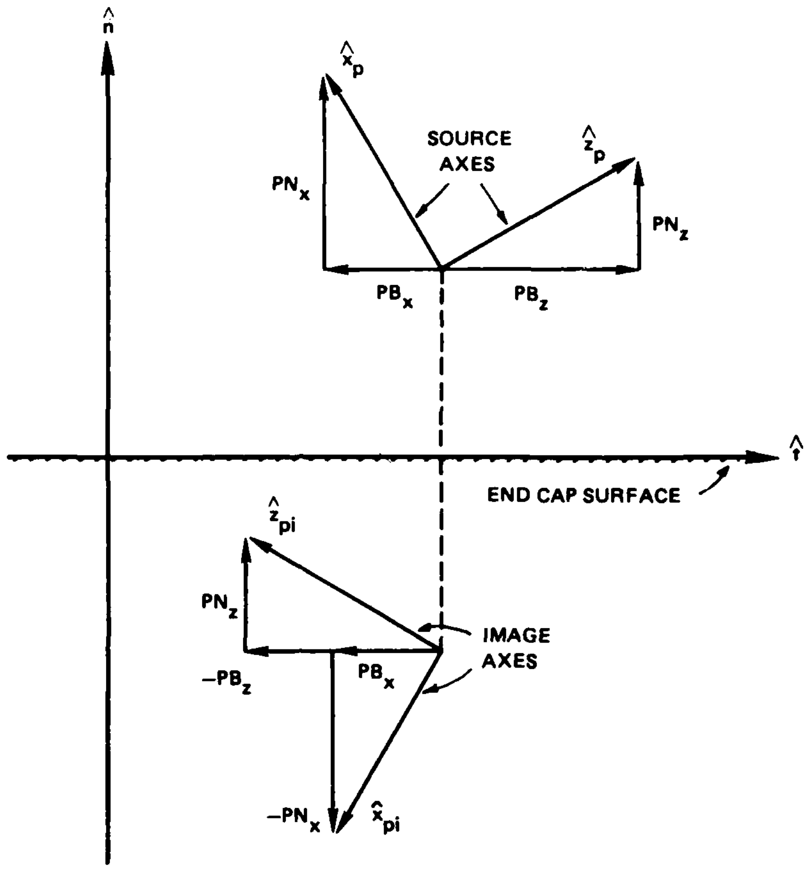


Figure 1. Imaging of Source Axes for Electric Source Through the End Cap Surface

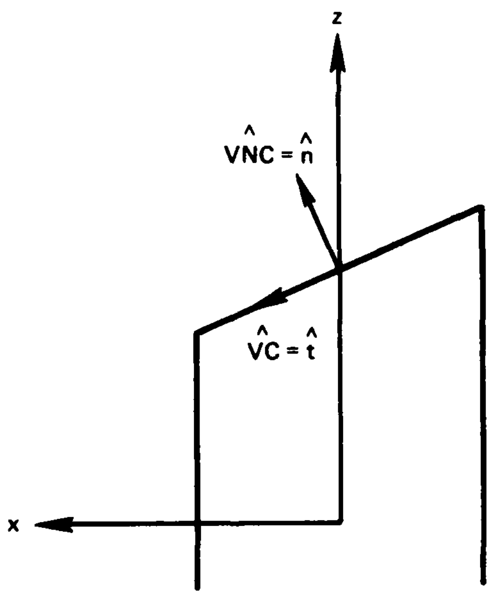


Figure 2. End Cap Coordinate System Shown on a Cylinder Projection into the X-Z Plane

VIMAG	Array of components defining the source image coordinate system axes in (x,y,z) reference coordinate system components
VNC	X,Y and Z components of end cap unit normal
VSOURC	Array of components defining the source coordinate system axes in (x, y, z) reference coordinate system components
VX,VY,VZ	X,Y, and Z components of source axis being imaged

5. I/O VARIABLES:

A. INPUT — LOCATION

VNC	F.P.
VSOURC	F.P.

B. OUTPUT — LOCATION

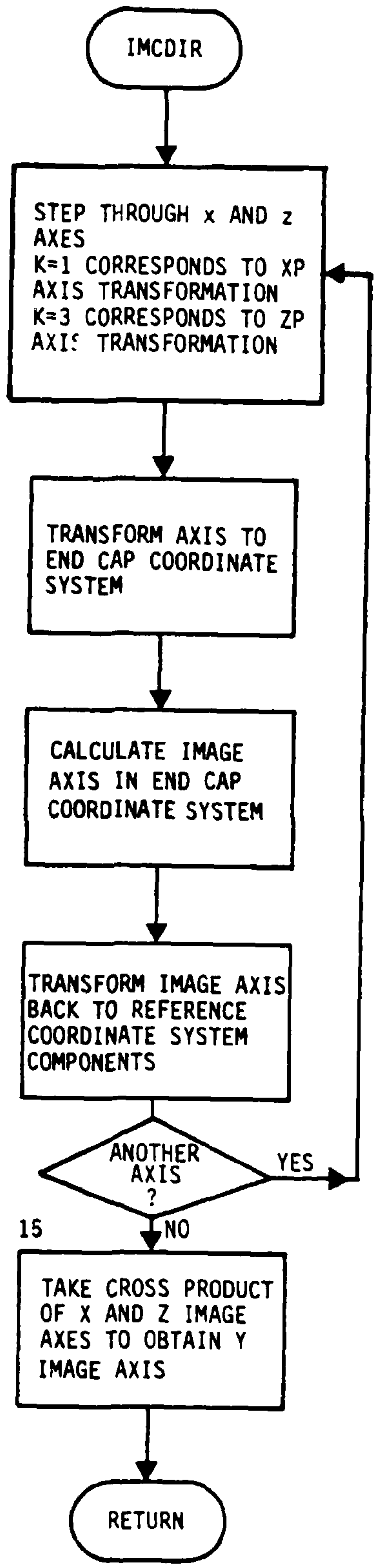
VIMAG	F.P.

6. CALLING ROUTINE:

GEOMC

7. CALLED ROUTINE:

NONE


IMCDIR
STEP THROUGH x AND z AXES
K=1 CORRESPONDS TO XP AXIS TRANSFORMATION
K=3 CORRESPONDS TO ZP AXIS TRANSFORMATION
TRANSFORM AXIS TO END CAP COORDINATE SYSTEM
CALCULATE IMAGE AXIS IN END CAP COORDINATE SYSTEM
TRANSFORM IMAGE AXIS BACK TO REFERENCE COORDINATE SYSTEM COMPONENTS
ANOTHER AXIS ?
YES
15
NO
TAKE CROSS PRODUCT OF X AND Z IMAGE AXES TO OBTAIN Y IMAGE AXIS
RETURN

1. NAME: IMDIR (GTD)

2. PURPOSE: To determine the source image axes directions for a source (or source image) after reflection from a given plate.

3. METHOD: The source image axes unit vectors for an electric source are given by:

$$\hat{x}_{pi} = (-\hat{x}_p \cdot \hat{n})\hat{n} + (\hat{x}_p \cdot \hat{t})\hat{t} + (\hat{x}_p \cdot \hat{b})\hat{b}$$

$$\hat{z}_{pi} = (\hat{z}_p \cdot \hat{n})\hat{n} + (-\hat{z}_p \cdot \hat{t})\hat{t} + (-\hat{z}_p \cdot \hat{b})\hat{b}$$

$$\hat{y}_{pi} = \hat{z}_{pi} \times \hat{x}_{pi}$$

where x_p, y_p, z_p are the unit vectors of the source coordinate system axes, and x_{pi}, y_{pi}, z_{pi} are the unit vectors of the source image coordinate system. Also n is the plate unit normal, t is the unit vector tangent to the plate and b=nxt. Unit vectors t and b are arbitrarily chosen to be the edge vector V and the binormal VP of edge one on the plate. The current flows in the z_p direction. An illustration of imaging the source coordinate system is shown in figure 1.

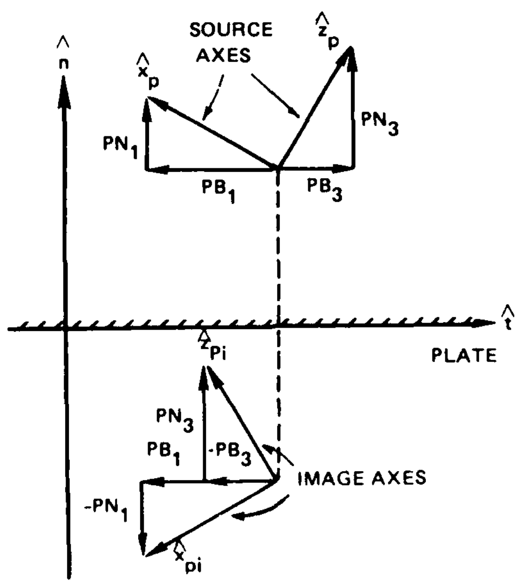


Figure 1. Imaging of Source Coordinate System for Electric Source (Shown in Two Dimension for Simplicity) Through the Plate Plane

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
K	K=1 corresponds to XP axis transformation, K=3 corresponds to ZP axis transformation
L	Incremental variable
MP	Plate of reflection
PB	Component of axis in plate plane normal to edge
PN	Component of axis normal to plate
PT	Component of axis parallel to plate edge
V	Array which includes the unit vector tangent to edge #1 on the plate where reflection occurs
VIMAG	X,Y,Z components defining unit vectors of the image source coordinate system axes in RCS
VN	Array which includes the unit vector normal to edge #1 of the plate reflection occurs from
VP	Array which includes the unit vector normal to the plate reflection occurs from
VSOURC	X,Y,Z components defining unit vectors of the source coordinate system axes in RCS
VX,VY,VZ	X,Y,Z components of axis under transformation in RCS

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
MP	F.P.
V	/GEOPLA/
VN	/GEOPLA/

	VP	/GEOPLA/
	VSOURC	F.P.
B.	OUTPUT	LOCATION
	VIMAG	F.P.

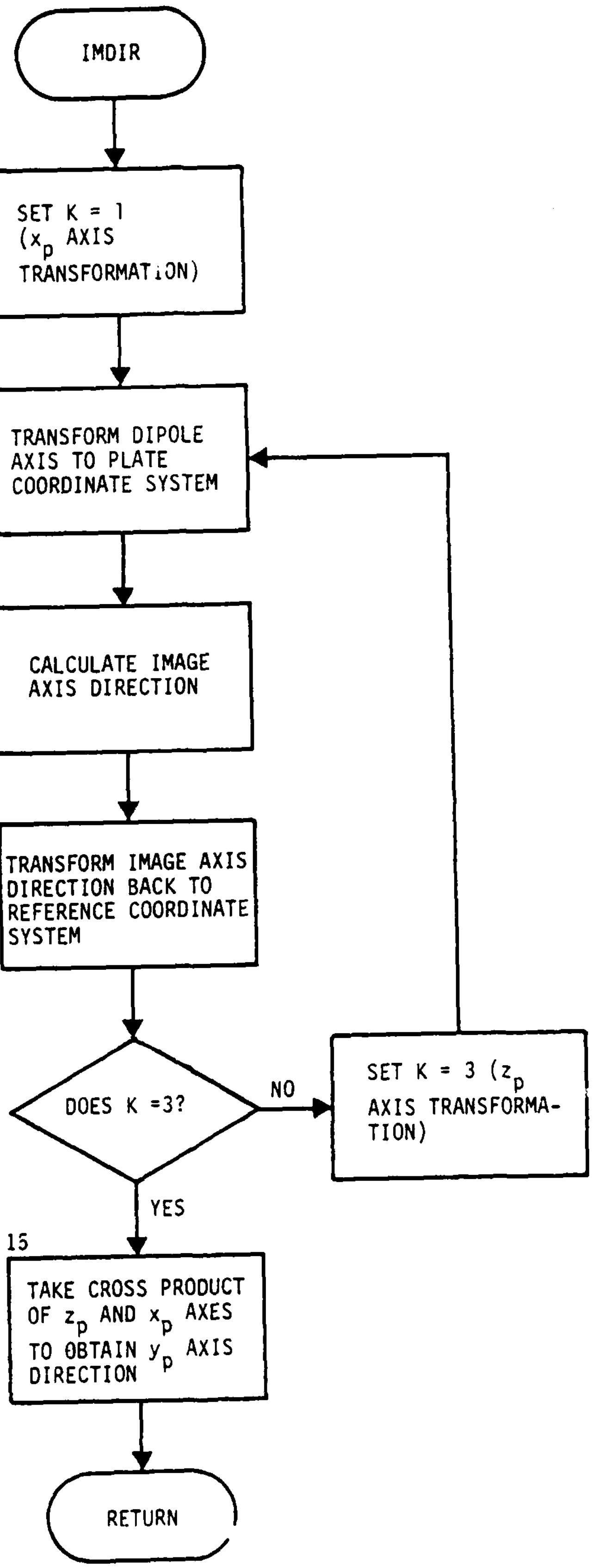
6. CALLING ROUTINES:

GEOM

RPLRPL

7. CALLED ROUTINE:

None


IMDIR
SET K = 1 (x_p AXIS TRANSFORMATION)
TRANSFORM DIPOLE AXIS TO PLATE COORDINATE SYSTEM
CALCULATE IMAGE AXIS DIRECTION
TRANSFORM IMAGE AXIS DIRECTION BACK TO REFERENCE COORDINATE SYSTEM
DOES K =3?
NO
SET K = 3 (z_p AXIS TRANSFORMATION)
YES
15
TAKE CROSS PRODUCT OF z_p AND x_p AXES TO OBTAIN y_p AXIS DIRECTION
RETURN

1. NAME: INCFLD (GTD)

2. PURPOSE: To calculate the unobstructed electric field transmitted by a unit source in a given direction.

3. METHOD: INCFLD is the driver routine which directs all the ray tracing, physics, and field calculations for determining the electric field from a unit source in a given far-field direction or at a near-field point.

The ray path is checked first for obstructions. For the far-field path this is an infinite distance from the source location in the given far-field direction. For near field, only the path between the source and the observation field point is checked. If the ray is obstructed by a plate or cylinder, the theta and phi components of the field are set to zero and no other calculations, except for debug functions if requested, are performed in this routine. If the ray path is unobstructed, the source field pattern factor is calculated by calling subroutine SOURCE. Then the phase factor is computed. For far field this will refer the field back to the origin of the reference coordinate system. Figure 1 shows the geometry involved. For near field, the phase factor includes the spherical wave spread factor. Now the theta and phi components of the electric field can be computed. The electric field is given by:

$$\bar{E} = (\underbrace{ETH\ \hat{\theta}}\ +\ \underbrace{EPH\ \hat{\phi}})\ \underbrace{e^{j2\pi(\overline{XS}\cdot\hat{D})}}, \text{ for far field}$$

from subroutine SOURCE - theta component of source factor	from subroutine SOURCE - phi component of source factor	PH - phase factor

and

$$\bar{E} = (\underbrace{ETH\ \hat{\theta}}\ +\ \underbrace{EPH\ \hat{\phi}})\ \underbrace{\frac{e^{-j2\pi SNF}}{SNF}}, \text{ for near field.}$$

from subroutine SOURCE - theta component of source factor	from subroutine SOURCE - phi component of source factor	PH - phase factor where $SNF = \lvert\overline{FLDPT} - \overline{XS}\rvert$

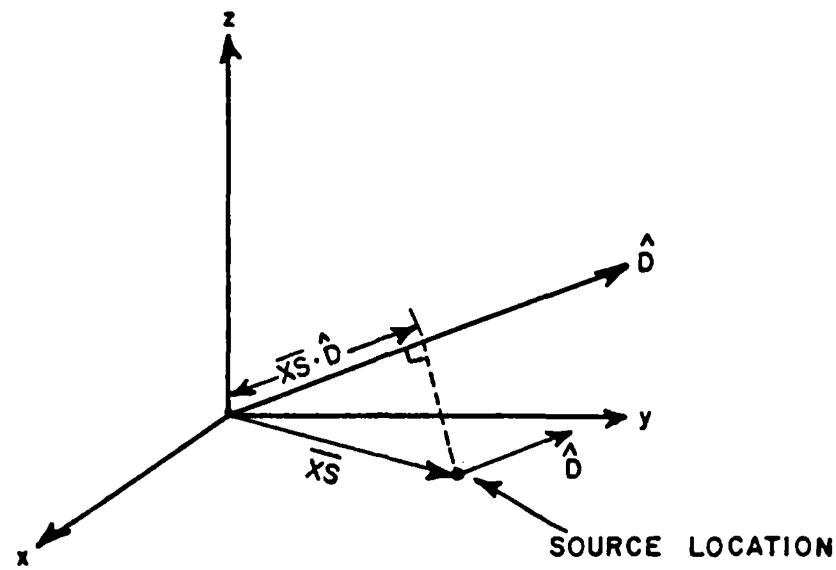


Figure 1. Geometry Showing Source Far-Field Phase Referred to the Origin of the Reference Coordinate System (RCS)

Subroutine XYZFLD is called to compute the x,y,z components of the E-field and to accumulate them in common block /FLDXYZ/.

If the debug capabilities have been requested, the field magnitude is computed. The magnitude, theta and phi complex components are printed on file LUPRNT.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CJ	Complex number j=(0.,1.)
D	Unit vector x,y,z components of ray propagation direction in RCS
DHT	Distance between source and field point
DHIT	Distance between source and nearest hit point (determined in PLAINT and CYLINT)
EPH	E-phi component of source field
ETH	E-theta component of source field

EX,EY,EZ	Source field pattern factor in x,y,z components
FLDMAG	Field magnitude
FLDPT	The x,y,z components of the field point
LDEBUG	Logical variable set true if debug was requested
LHIT	Set true if ray hits plate or cylinder
LNRFLD	Flag to indicate if far-field (LNRFLD=0) or near-field (LNRFLD=1) calculations were requested
LSOR	Logical variable set true if plates and cylinders are to be ignored, however LSOR is set FALSE in subroutine GTDDRV
LUPRNT	Output file number
N	DO loop variable for x,y,z components
PH	Complex phase constant
PHSR	Phi angle of propagation direction in RCS
SNF	Distance between source and field point
THSR	Theta angle of propagation direction in RCS
TPI	2π
VXS	A 3x3 matrix which defines the source axes in the reference coordinate system
XS	The x,y,z components of the source location in RCS
XS1	The same as XS

5. I/O VARIABLES:

A. INPUT	LOCATION
D	/DIR/
FLDPT	/NEAR/

LDEBUG	/TEST/
LNRFLD	/NEAR/
LSOR	F.P.
LUPRNT	/ADEBUG/
PHSR	/DIR/
THSR	/DIR/
TPI	/PIS/
VXS	/SORINF/
XS	/SORINF/
B. OUTPUT	LOCATION
EPH	F.P.
ETH	F.P.

6. CALLING ROUTINE:

GTDDRV

7. CALLED ROUTINES:

ASSIGN

BEXP

CYLINT

NFD

PLAINT

SMAGNF

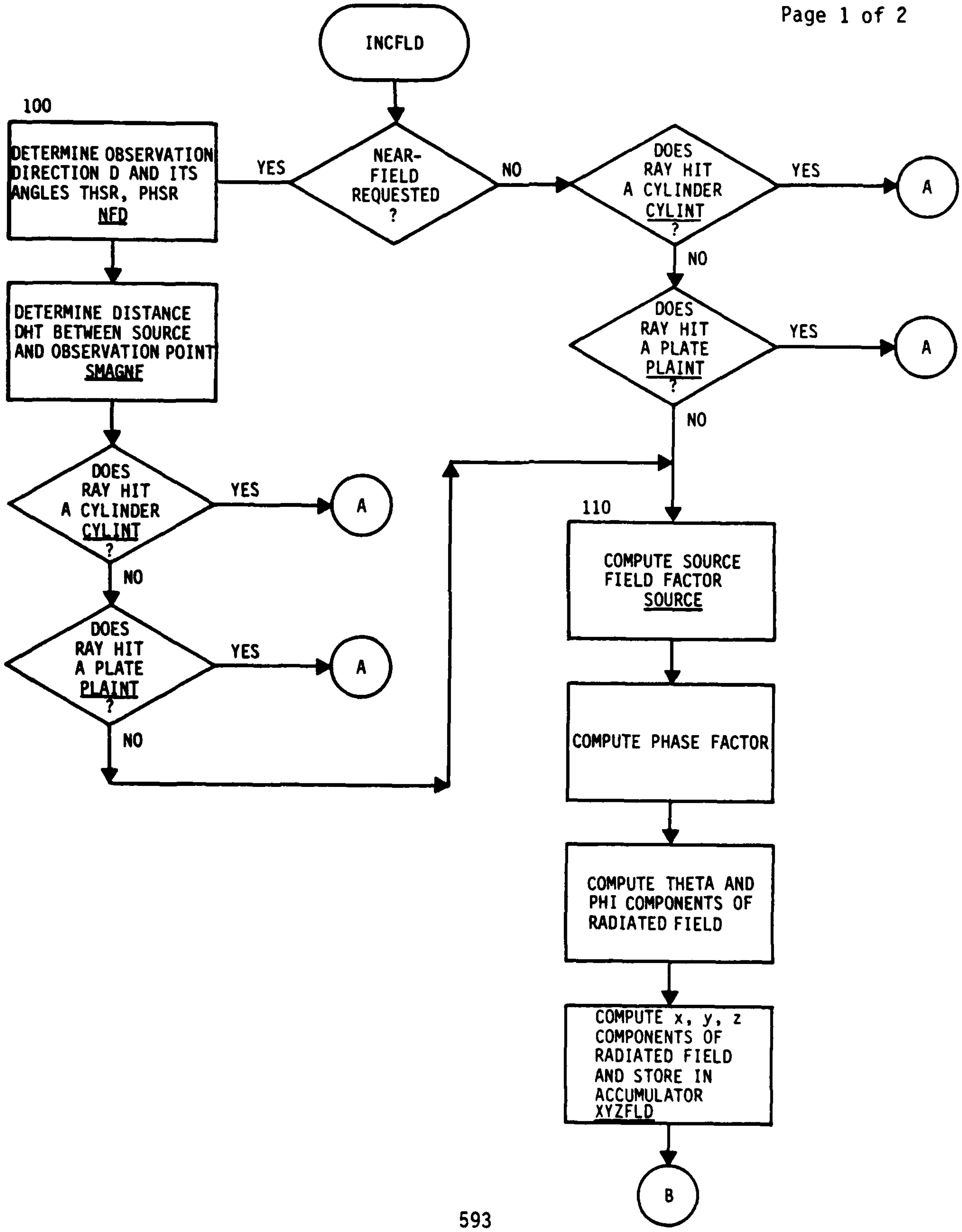
SOURCE

STATIN

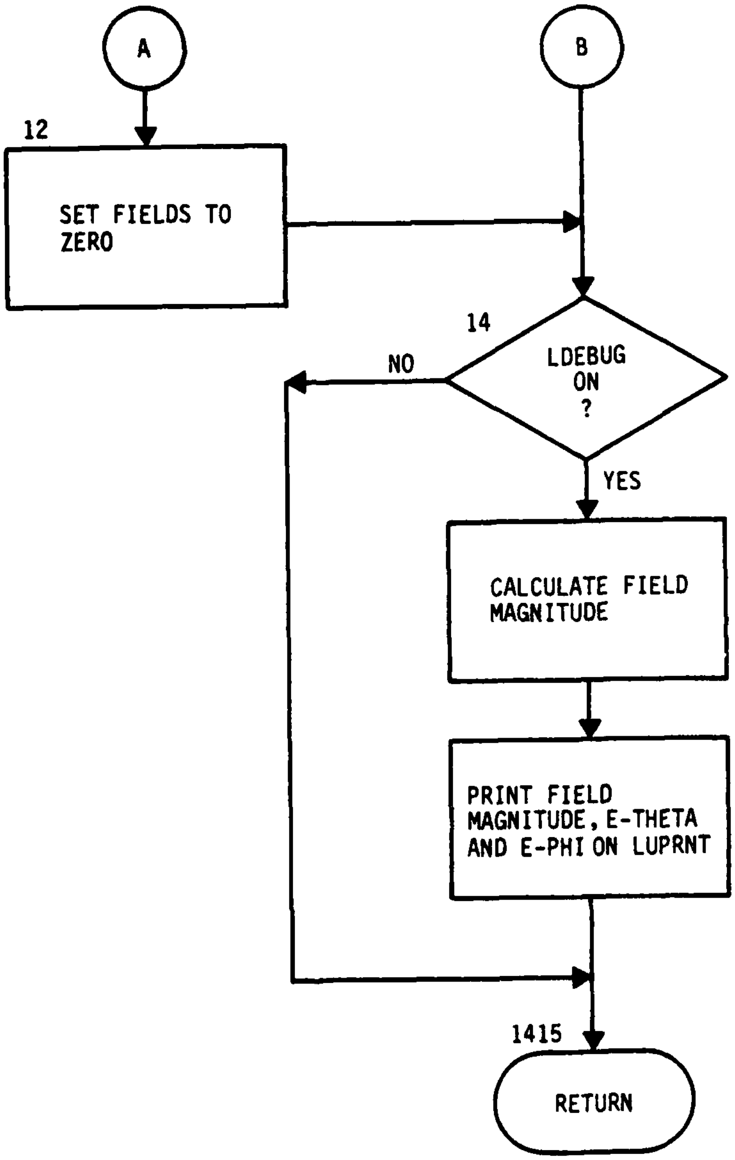
STATOT

WLKBCK

XYZFLD



INCFLD
NEAR-FIELD REQUESTED ?
YES
NO
100
DETERMINE OBSERVATION DIRECTION D AND ITS ANGLES THSR, PHSR
NFD
DETERMINE DISTANCE DHT BETWEEN SOURCE AND OBSERVATION POINT
SMAGNF
DOES RAY HIT A CYLINDER
CYLINT
?
YES
A
NO
DOES RAY HIT A PLATE
PLAINT
?
YES
A
NO
110
COMPUTE SOURCE FIELD FACTOR
SOURCE
COMPUTE PHASE FACTOR
COMPUTE THETA AND PHI COMPONENTS OF RADIATED FIELD
COMPUTE x, y, z COMPONENTS OF RADIATED FIELD AND STORE IN ACCUMULATOR
XYZFLD
B


A
B
12
SET FIELDS TO ZERO
14
LDEBUG ON ?
NO
YES
CALCULATE FIELD MAGNITUDE
PRINT FIELD MAGNITUDE, E-THETA AND E-PHI ON LUPRNT
1415
RETURN

1. NAME: INPDRV (INPUT)

2. PURPOSE: This is the driver routine for the Input Language Processor.

3. METHOD: Subroutine INPDRV sets up initial flags and table pointers. It calls SCAN to input each command text to the scan tables. Then PARSE is called to parse the command and put needed arguments in the argument list. If an RSTART command is parsed, the RESTRT routine is called to read in common blocks and peripheral files. When the END command is parsed and there have been no errors, PRESCN is called which wipes out specified commands. Finally, POSTIP is called to print out tables if the debug for ILP is turned on. If no error is encountered, control is returned normally to GEMACS.

4. INTERNAL VARIABLE:

VARIABLE	DEFINITION
NLOOP	Index to loop table

5. I/O VARIABLES:

A.

INPUT	LOCATION
ISOFF	/ADEBUG/
ISON	/ADEBUG/
KOLCNT	/PARTAB/
LUPRNT	/ADEBUG/
NARGMX	/PARTAB/
NCARD	/SCNPAR/
NPEARG	/INPERR/
NPELIT	/INPERR/
NPELNF	/INPERR/
NPELOO	/INPERR/
NPESCN	/INPERR/
NPESYM	/INPERR/

	NPETSK	/INPERR/
	NVALMX	/SCNPAR/
B.	OUTPUT	LOCATION
	ICALL	/ADEBUG/
	IGNORE	/SCNPAR/
	NARGTB	/PARTAB/
	NCARD	/SCNPAR/
	NCARDS	/SCNPAR/
	NCODE	/SCNPAR/
	NFINCD	/SCNPAR/
	NLOOPS	/PARTAB/
	NOGOFG	/ADEBUG/
	NOPCOD	/ADEBUG/
	NPARGL	/PARTAB/
	NPDATA	/PARTAB/
	NPLITN	/PARTAB/
	NLOOP	/PARTAB/
	NPLOOP	/PARTAB/
	NPRSER	/SCNPAR/
	NPTASK	/PARTAB/
	NRDCDF	/SCNPAR/
	NRESTF	/SCNPAR/
	NSCNER	/SCNPAR/
	NTSKTB	/PARTAB/
	NVAL	/SCNPAR/
	RSTART	/SYSFIL/

6. CALLING ROUTINE:

GEMACS

7. CALLED ROUTINES:

ASSIGN

PARSE

POSTIP

PRESCN

RESTRT

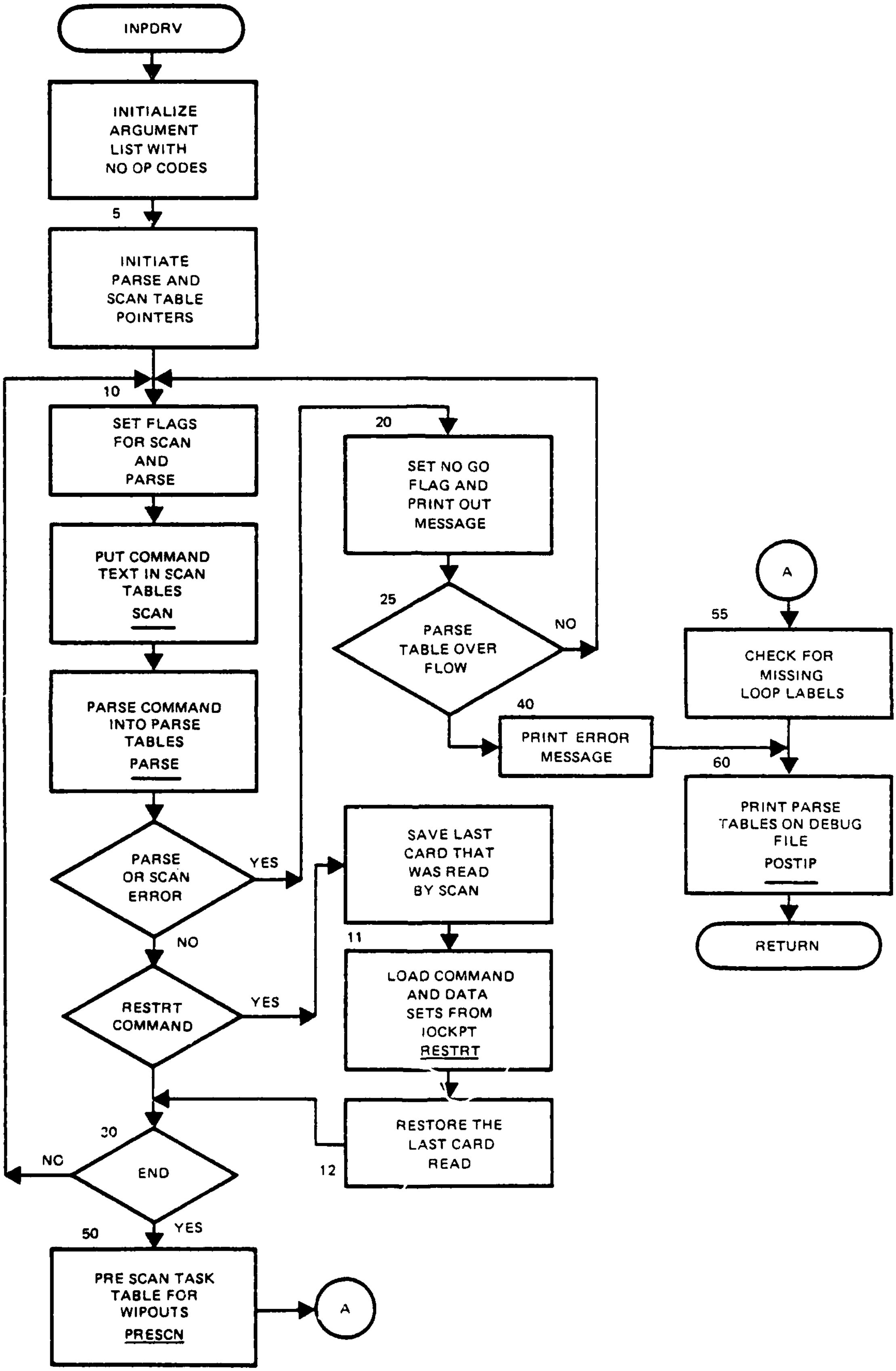
SCAN

STATIN

STATOT

WLKBCK

INPDRV (INPUT)



1. NAME: INTPLT (GTD)

2. PURPOSE: Compute at a field point the contributions of all basis functions whose domains include the source segment.

3. METHOD: Each wire basis function spans three wire segments, as shown in figure 1. Hence, the total current on any one segment is composed of parts of several basis functions (figure 2). Each basis function has the form

$$I_j(s) = A_j + B_j \sin k(s-s_j) + C_j \cos k(s-s_j)$$

with (A_j, B_j, C_j) chosen so that

$$I_j(s_i) = 1$$

$$I_j(s_l) = 0$$

$$I_j(s_k) = 0$$

The coefficients are directly related to the lengths of the segments spanned by the basis function ($\delta_{i\ell}$ and δ_{ik} of figure 1), so in general (A, B, C) will be different for each basis function.

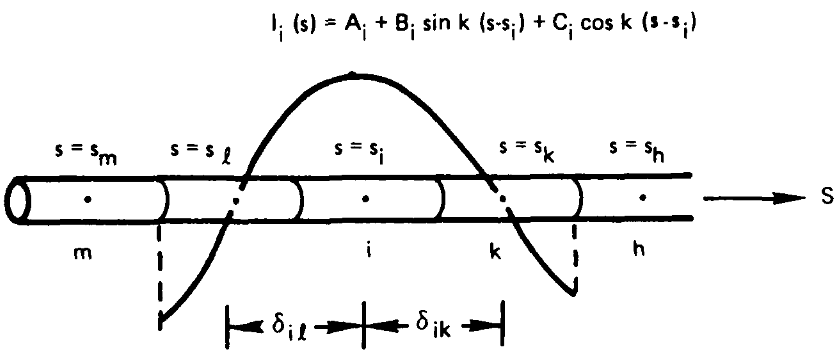
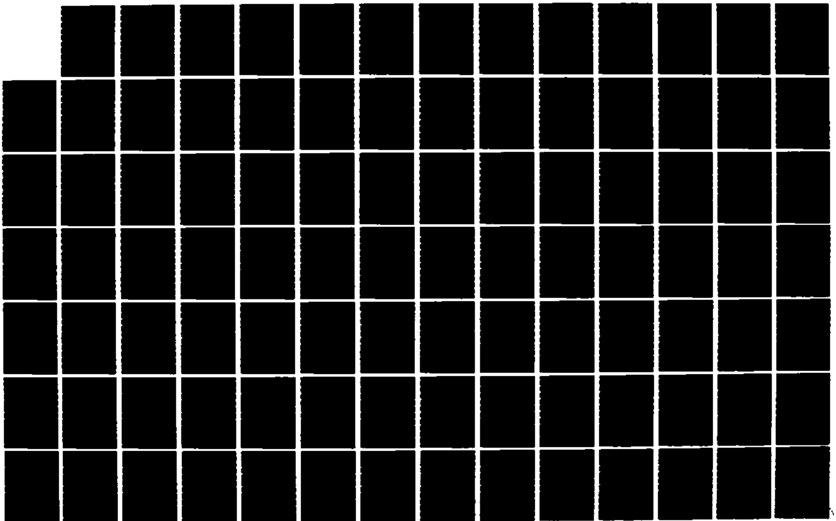
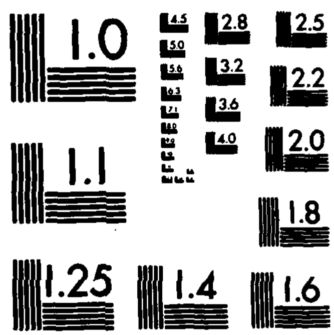


Figure 1. Basis Function $I_i(s)$ Spans Segments ℓ, i, and k

AD-A137 464 GENERAL ELECTROMAGNETIC OODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS..(U) BDM CORP ALBUQUERQU NM D L KADLEC ET AL. SEP 83 BDM/A-83-020-TR-VOL-3-PT-2 3/4

UNCLASSIFIED RADC-TR-83-217-VOL-3-PT-2 F30602-81-C-0084 F/G 20/14 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

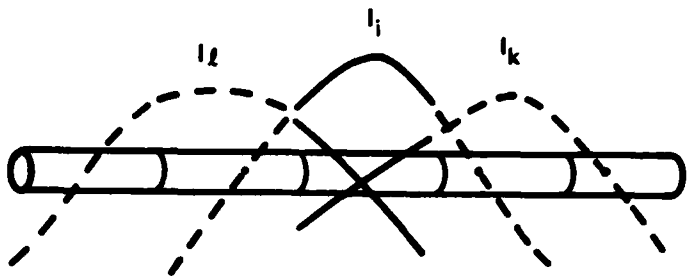


Figure 2. Illustrating How Three Basis Functions Contribute to the Total Current on a Segment

This subroutine computes the contribution of all basis functions spanning segment i by shifting the origins from $s=s_i$ to $s=s_l$ or $s=s_k$. The trigonometric identities used for the shift determine the coefficient weights for the basis functions.

When more than one segment is attached to an end of the source segment, the average of all δ_{ij}s is used as interpolating length.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CK	$k\delta_{ik}$
CL	$k\delta_{il}$
CONS	SINL + SINK - SILK
COSK	$\cos k\delta_{ik}$
COSL	$\cos k\delta_{il}$
FT	Tangential field due to basis functions spanning segment i
FTC	Tangential field from a unit cosine current on segment i
FTK	Tangential field from a unit pulse current on segment i

FTS	Tangential field from a unit sine current on segment i
SILK	$\sin k(\delta_{il} + \delta_{ik})$
SINK	$\sin k(\delta_{ik})$
SINL	$\sin k(\delta_{il})$

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
ANUMK	/ANUM/
ANUML	/ANUM/
DIK	/AMPZIJ/
DIL	/AMPZIJ/
FTC	F.P.
FTK	F.P.
FTS	F.P.
WAVNUM	/AMPZIJ/

B. OUTPUT

OUTPUT	LOCATION
FT	F.P.

6. CALLING ROUTINE:

ZGTDRV

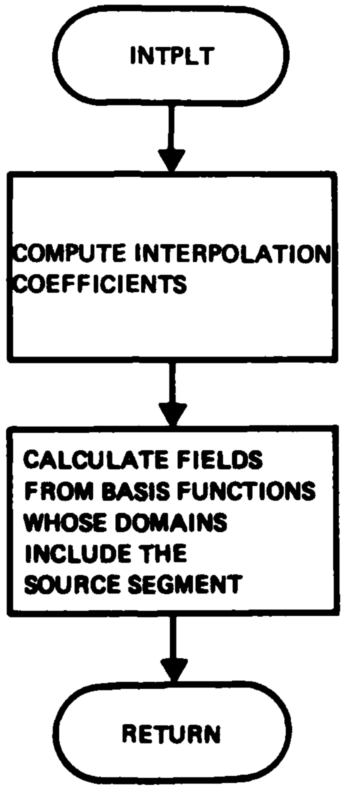
7. CALLED ROUTINES:

ASSIGN

STATIN

STATOT

WLKBCK


INTPLT
COMPUTE INTERPOLATION COEFFICIENTS
CALCULATE FIELDS FROM BASIS FUNCTIONS WHOSE DOMAINS INCLUDE THE SOURCE SEGMENT
RETURN

1. **NAME:** JCTION (INPUT)

2. **PURPOSE:** This subroutine locates all the wire segments which are connected at a junction, and all segments that are connected to patches.

3. **METHOD:** All wire segment end points are checked and those end points that are within the sum of the segment radii are considered to be connected. Three or more segments connected at the same point constitute a multiple junction. Both ends of each wire segment are checked to see if either end is connected to a patch. If a segment end is within the radius of the wire segment to the patch, it is considered to be connected.

4. **INTERNAL VARIABLES:**

VARIABLE	DEFINITION
DX	The difference in x between segment end points
DY	The difference in y between segment end points
DZ	The difference in z between end points
IBLK	Data block index for wire segment I
ICNECT	Connection word for segment I
IJCONT	Flag set to 1 for segment I connected to segment J
ILIM	Location of last wire segment in each data block
IMINUS	Connection data for negative end of segment I
INPBLK	Data block where the first patch segment is located
IPBLK	Data block index for patch I
IPLIM	Location of last patch in each data block
IPLOW	Location of first patch segment in each data block

IPLUS	Connection data for positive end of segment I
IPTAG	Tag number for patch I
IS	Segment number for segment I
ISAV	Saved value for segment I
ISP	Segment number for patch I
ISW	Segment number for wire
ITAG	Segment tag information
IWBLK	Data block index for wires
IWLIM	Location of last wire segment in each data block
JBLK	Data block index for wire segment J
JCBIAS	Bias flag to indicate segment connected to multiple junction
JCNECT	Connection word for segment J
JCT	Junction number
JCTCON	Junction number for a wire connected to a patch
JCTNEG	Multiple junction number for negative end of segment J
JCTPOS	Multiple junction number for positive end of segment J
JLIM	Location of last wire segment in each data block
JMINUS	Connection data for negative end of segment J
JP	First patch to be searched for a duplicate patch after patch I
JPBLK	Data block index for patch J

JPLIM	Location of last patch in each data block
JPLUS	Connection data for positive end of segment J
JPTAG	Tag number for patch J
JS	Segment number for segment J
JSAV	Saved value of segment J
JTAG	Tag identification for segment J
J1	First segment to be searched after segment I
MAXSEG	Maximum number of segments in each data block
MLTJCT	Biased multiple junction identification number
MXBLKW	Number of data blocks containing wire segments
NDXBLK	Data block currently being used
NPATCH	Number of patch segments
NUMCON	Number of wire segments connected to a patch
NUMNEG	Number of segments connected at negative end of segment I
NUMPOS	Number of segments connected at positive end of segment I
NWIRE	Total number of wire segments
RI	Radius of segment I
RJ	Radius of segment J
RMIN	Minimum distance specifying connection criterion
RSQ	Radius squared

RW	Radius of wire used to determine a wire to patch connection
XCP,YCP,ZCP	X,Y, and Z coordinates of patch I center point
XCPJ,YCPJ,ZCPJ	X,Y, and Z coordinates of patch J center point
XMI,YMI,ZMI	X,Y, and Z coordinates for minus end of wire segment I
XMJ,YMJ,ZMJ	X,Y, and Z coordinates for minus end of wire segment J
XMW,YMW,ZMW	X,Y, and Z coordinates of the negative end of the wire used for a wire to patch connection
XPI,YPI,ZPI	X,Y, and Z coordinates of the positive end of wire segment I
XPJ,YPJ,ZPJ	X,Y, and Z coordinates of the positive end of wire segment J
XPW,YPW,ZPW	X,Y, and Z coordinates of the positive end of the wire used for a wire to patch connection
ZEROSQ	Connection separation distance criterion

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
IP217	/GEODAT/
ISGTBL	/SEGMNT/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
JCBIAS	/SEGMNT/
LUPRNT	/ADEBUG/
MAXSEG	/SEGMNT/

NDXBLK	/SEGMNT/
NPATCH	/SEGMNT/
NWIRE	/SEGMNT/
SEGTBL	/SEGMNT/
ZERO	/ADEBUG/

B. OUTPUT LOCATION

ISGTBL	/SEGMNT/
MLTJCT	/SEGMNT/
NOGOFG	/ADEBUG/
UPDBLK	/SEGMNT/

6. CALLING ROUTINE:

GEODRV

7. CALLED ROUTINES:

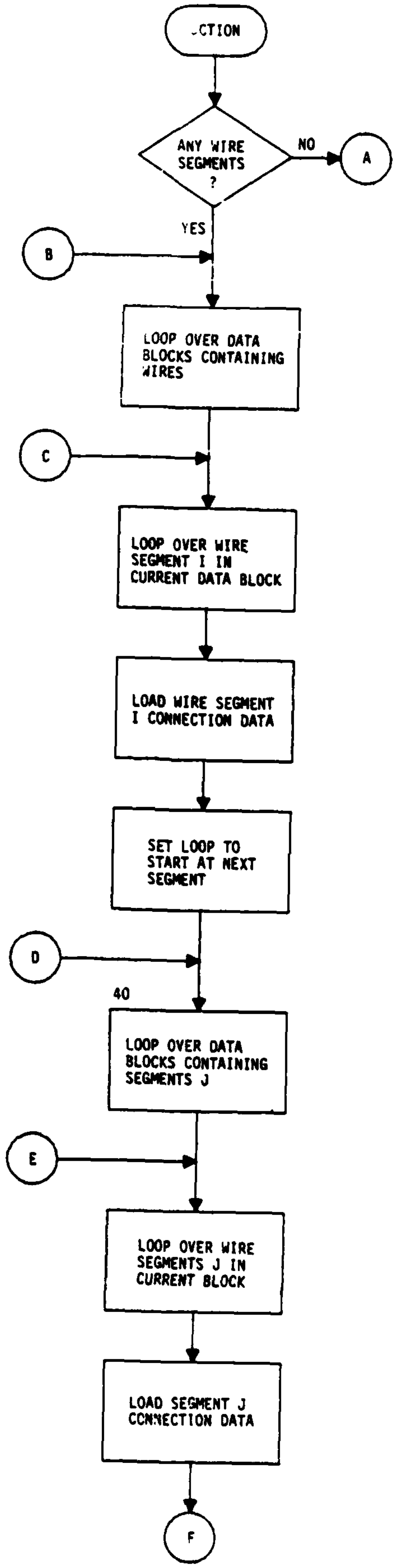
ASSIGN

GETSEG

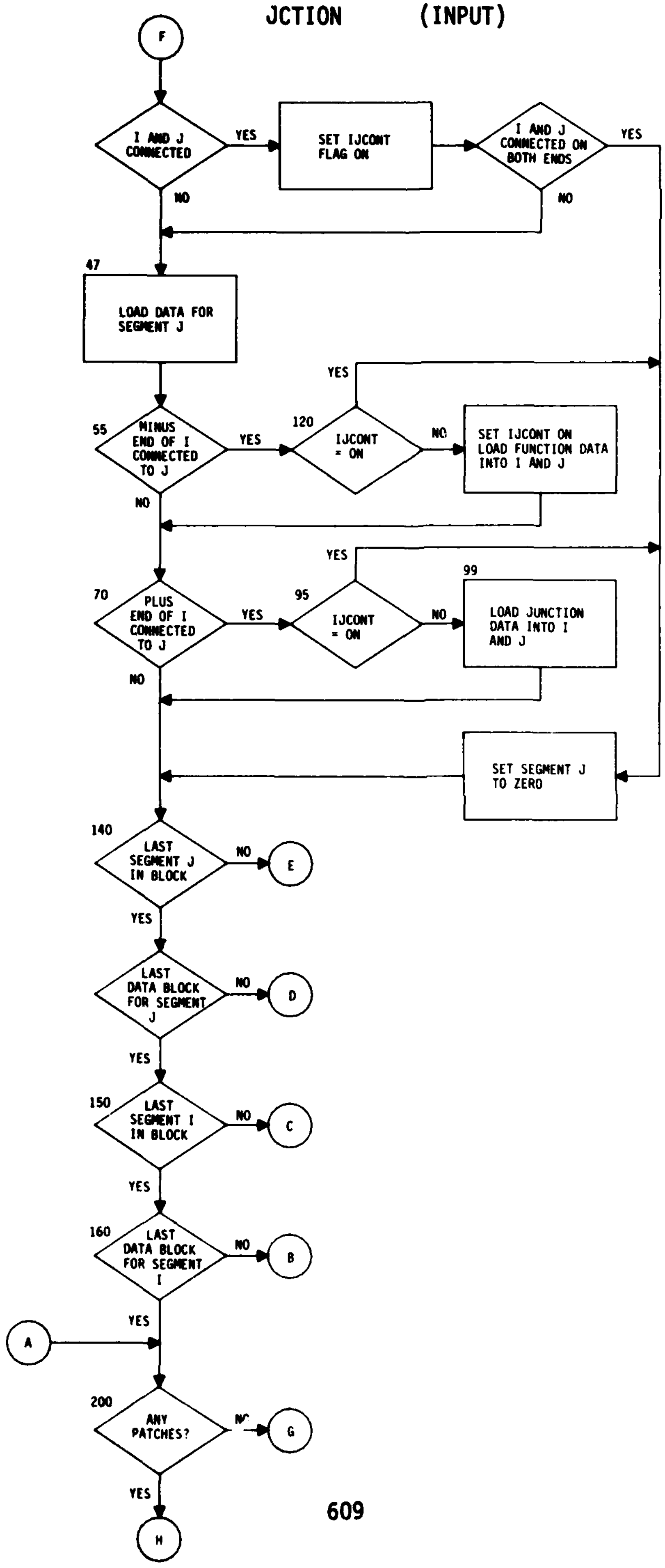
STATIN

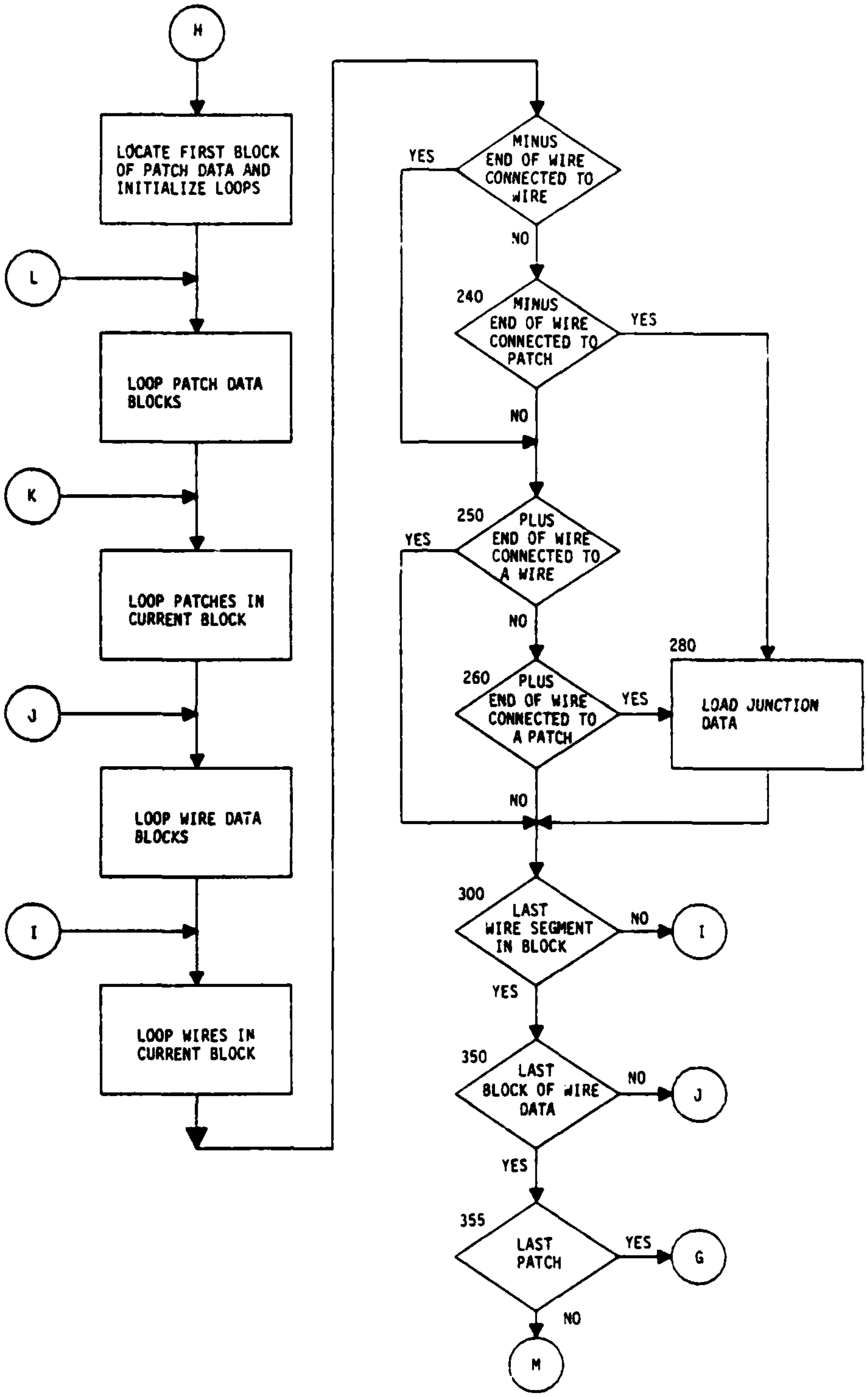
STATOT

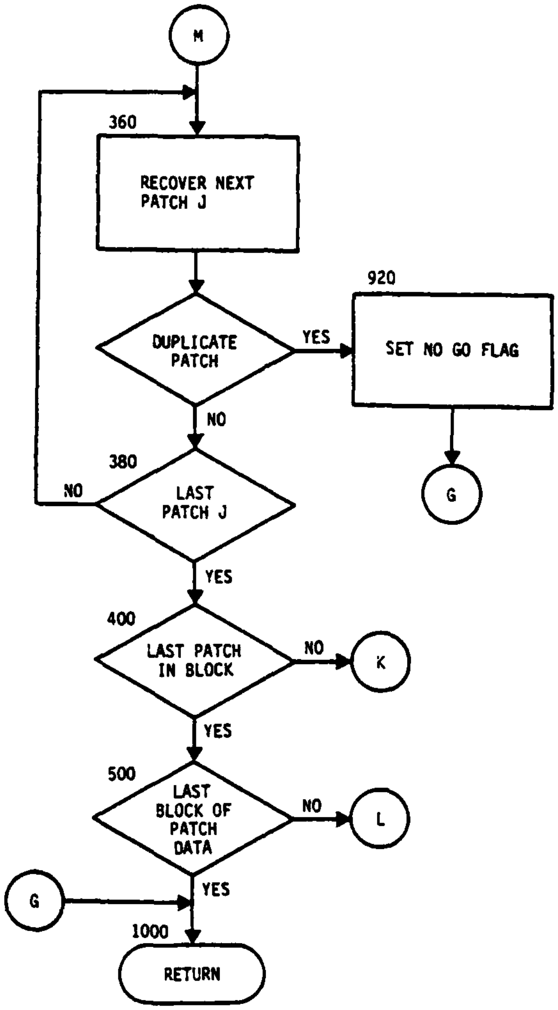
WLKBCK



JCTION
ANY WIRE SEGMENTS ?
NO
A
YES
B
LOOP OVER DATA BLOCKS CONTAINING WIRES
C
LOOP OVER WIRE SEGMENT I IN CURRENT DATA BLOCK
LOAD WIRE SEGMENT I CONNECTION DATA
SET LOOP TO START AT NEXT SEGMENT
D
40
LOOP OVER DATA BLOCKS CONTAINING SEGMENTS J
E
LOOP OVER WIRE SEGMENTS J IN CURRENT BLOCK
LOAD SEGMENT J CONNECTION DATA
F


F
I AND J CONNECTED
YES
SET IJCONT FLAG ON
I AND J CONNECTED ON BOTH ENDS
YES
NO
NO
47
LOAD DATA FOR SEGMENT J
YES
55
MINUS END OF I CONNECTED TO J
YES
120
IJCONT = ON
NO
SET IJCONT ON LOAD FUNCTION DATA INTO I AND J
NO
YES
99
70
PLUS END OF I CONNECTED TO J
YES
95
IJCONT = ON
NO
LOAD JUNCTION DATA INTO I AND J
NO
SET SEGMENT J TO ZERO
140
LAST SEGMENT J IN BLOCK
NO
E
YES
LAST DATA BLOCK FOR SEGMENT J
NO
D
YES
150
LAST SEGMENT I IN BLOCK
NO
C
YES
160
LAST DATA BLOCK FOR SEGMENT I
NO
B
YES
A
200
ANY PATCHES?
G
YES
H


H
LOCATE FIRST BLOCK OF PATCH DATA AND INITIALIZE LOOPS
L
LOOP PATCH DATA BLOCKS
K
LOOP PATCHES IN CURRENT BLOCK
J
LOOP WIRE DATA BLOCKS
I
LOOP WIRES IN CURRENT BLOCK
MINUS END OF WIRE CONNECTED TO WIRE
YES
NO
240
MINUS END OF WIRE CONNECTED TO PATCH
YES
NO
250
PLUS END OF WIRE CONNECTED TO A WIRE
YES
NO
260
PLUS END OF WIRE CONNECTED TO A PATCH
YES
NO
280
LOAD JUNCTION DATA
300
LAST WIRE SEGMENT IN BLOCK
NO
I
YES
350
LAST BLOCK OF WIRE DATA
NO
J
YES
355
LAST PATCH
YES
G
NO
M


M
360
RECOVER NEXT PATCH J
920
DUPLICATE PATCH
YES
SET NO GO FLAG
NO
G
380
NO
LAST PATCH J
YES
400
LAST PATCH IN BLOCK
NO
K
YES
500
LAST BLOCK OF PATCH DATA
NO
L
G
YES
1000
RETURN

1. NAME: JNCSUM (GTD, MOM)

2. PURPOSE: To add matrix element contributions due to segments at multiple junctions to the appropriate matrix locations.

3. METHOD: JNCSUM is called each time that a multiple segment junction occurs on either end of the source segment J. The matrix element contributions, which are ETR(1), ETI(1) if the multiple junction is on end 1 or ETR(3), ETI(3) if the multiple junction is on end 2, are added to the matrix rows of all segments connected to the junction if their directions are the same as that of segment J, and subtracted if their directions oppose that of segment J.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CM	Segment interaction matrix
ETI	Imaginary part of matrix element contribution
ETR	Real part of matrix element contribution
I	Index of matrix column into which element contributions are entered
IPR	Input index of a matrix column
J	DO loop index
JM	Array of indices of matrix rows from which ETR and ETI are subtracted
JMJ	JM(J)
JP	Array of indices of matrix rows to which ETR and ETI are added
JPJ	JP(J)
MAXCON	Not used
NCM	Number of entries in JM array
NCOL	Number of columns in matrix
NCP	Number of entries in JP array
NROW	Number of rows in matrix

5. I/O VARIABLES:

A. INPUT — LOCATION

INPUT	LOCATION
ETI	F.P.
ETR	F.P.
IPR	F.P.
JM	F.P.
JP	F.P.
MAXCON	F.P.
NCM	F.P.
NCOL	F.P.
NCP	F.P.
NROW	F.P.

B. OUTPUT — LOCATION

OUTPUT	LOCATION
CM	F.P.

6. CALLING ROUTINES*:

ZGTDRV (2)

ZIJSET (3)

7. CALLED ROUTINES:

ASSIGN

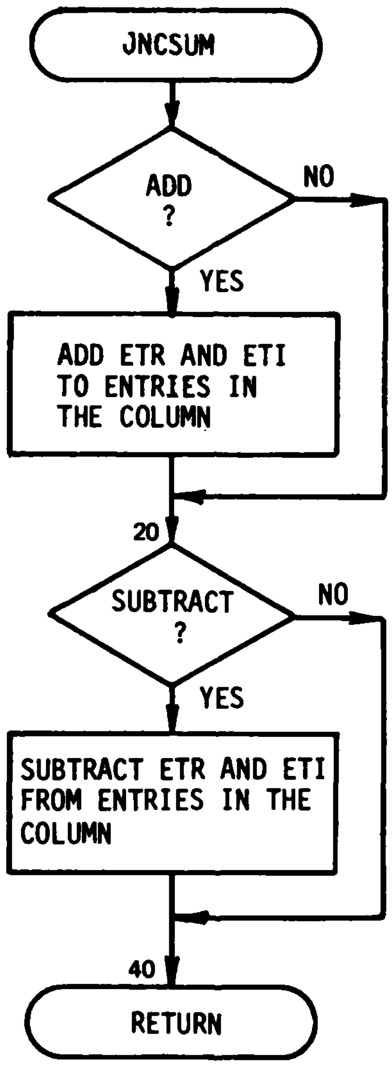
STATIN

STATOT

WLKBCK

*2-GTD
3-MOM

JNCSUM (GTD,MOM)



1. NAME: LITSCH (INPUT)

2. PURPOSE: To search for or insert a number into the literal table and return an index from the table.

3. METHOD: The current scan table entry NVAL is checked for a monadic sign. If found, the scan table pointer NTAB is incremented by 1 and the next scan entry is assigned the monadic sign. The literal table is then searched for a matched entry. If found, the index to the literal table is returned to the calling argument. If it is not found, a new entry is made and the new entry index is returned.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IEND	Pointer to the last filled location of the literal table
IFLG	Flag indicating double precision number
INDEX	Index to the literal table
NSIGN	Algebraic sign of the literal

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
IMINUS	/SCNPAR/
INDEX	F.P.
IPLUS	/SCNPAR/
ISOFF	/ADEBUG/
KOLCOD	/PARTAB/
KOLVAL	/PARTAB/
LITNMX	/PARTAB/
NCODE	/SCNPAR/
NOPCOD	/ADEBUG/
NPELIT	/INPERR/

NPENUM	/INPERR/
NPLITN	/PARTAB/
NTAB	/SCNPAR/
NTALPH	/ADEBUG/
NTERR	/ADEBUG/
NTFLPT	/ADEBUG/
NTINT	/ADEBUG/
NTKEYW	/ADEBUG/
NTSYMB	/ADEBUG/
NVAL	/SCNPAR/
VAL	/SCNPAR/

B. OUTPUT LOCATION

INDEX	F.P.
LITNUM	/PARTAB/
NCODE	/SCNPAR/
NPLITN	/PARTAB/
NPRSER	/SCNPAR/
NTAB	/SCNPAR/
NVAL	/SCNPAR/
VAL	/SCNPAR/

6. CALLING ROUTINES:

FNDARG

PARSE

SYMLIT

7. CALLED ROUTINES:

ASSIGN

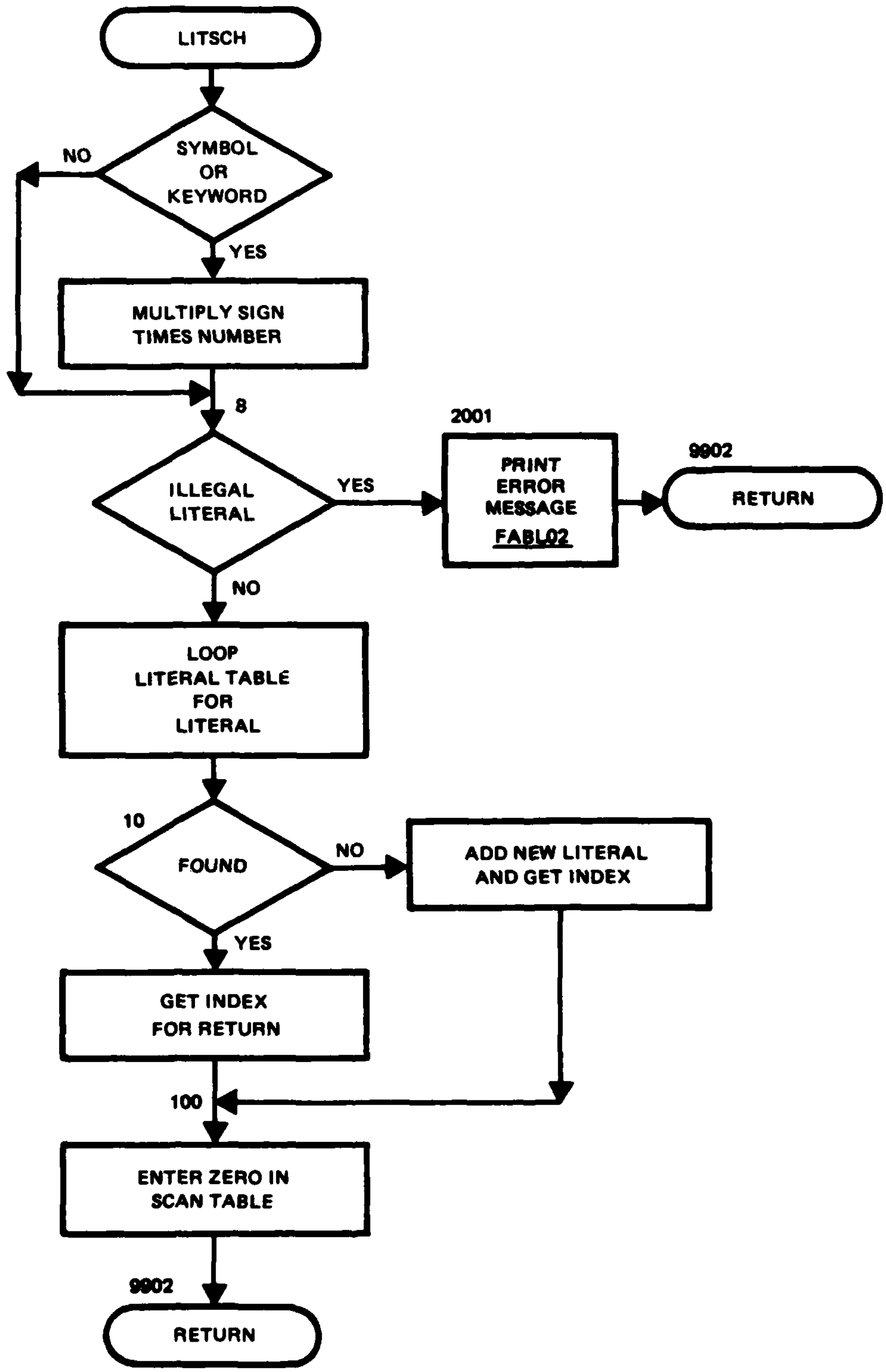
FABLO2

STATIN

STATOT

WLKBCK

LITSCH (INPUT)



1. **NAME:** LNKGTD (INPUT)

2. **PURPOSE:** To link together connected GTD geometry objects by placing linking data in appropriate SEGTBL locations, and to link MOM segments connected to GTD plates.

3. **METHOD:** Since the geometry data set is divided into blocks and resides on a data file, extensive use is made of the in-core array PNTTBL to store intermediate linkage data. This approach makes it possible to link all GTD geometries with only two sequential passes through the geometry data set.

   The purpose of the first pass (Loop 110) is to place in PNTTBL additional data that will be needed to link the geometries:

   Plates: No data required

   Cylinders: Segment number of cylinder into cylinder PNTTBL entry

   End caps: Segment numbers of cylinder and end cap into end cap PNTTBL entry. Segment number of end cap into cylinder PNTTBL entry

   The second pass utilizes the PNTTBL data to place the segment numbers of adjacent GTD geometries into the SEGTBL entry of the GTD geometry. The contents of SEGTBL after pass two are:

	CYLINDER	END CAP	PLATE
1	ITAG/IS	ITAG/IS	
2	LOCEC1/LOCEC2*	LOCECO/LOCCYL*	(No
3	AA (m)	ZLOC*	change)
4	BB (m)	THETA (radians)	
5	FLEN (m)	PHI (radians)	
6	XTRL (m)	XTRL* (m)	
7	YTRL (m)	YTRL* (m)	
8	ZTRL (m)	ZTRL* (m)	
9	THETA (radians)	THETAR* (radians)	
10	PHI (radians)	PHIR* (radians)	
11	cylinder number/coord sys#	end cap number	

The starred entries are the changes made by LNKGTD.

If there are wire segments in the geometry, CNVGTD is called to identify segments connected to GTD plates.

4. INTERNAL VARIABLES

VARIABLE	DESCRIPTION
D	Dummy variable
ERRFLG	Internal flag to indicate a linking error (integer)
FLEN	Length of cylinder (m)
F1,F2,...F6	Arguments for GETPNT and PUTPNT
ICSYS	Coordinate system number
ICYLPT	PNTTBL point number for a cylinder
IECPT	PNTTBL point number for an end cap
IECOPT	PNTTBL point number for "other" end cap on this end cap's cylinder
IEC1	Top of cylinder end cap number
IEC2	Bottom of cylinder end cap number
I1,I2,...I6	Arguments for GETPNT and PUTPNT (equivalenced to F1-F6)
IG	Index to SEGTBL entry
IGBLK	Geometry block in SEGTBL
IGLIM	Last GTD entry in SEGTBL
IGLOW	First GTD entry in SEGTBL
INPBLK	First geometry block containing GTD entries
ISGN	Top (plus) or bottom (minus) of cylinder flag
ITAG	GTD geometry tag number
ITP	Index over GTD geometry type loop

ITYPE	Tag identifier for GTD object being considered
LOCCYL	Segment number of cylinder
LOCEC	Segment number of end cap
LOCECO	Segment number of "other" end cap on cylinder
LOCEC1	Segment number of top end cap
LOCEC2	Segment number of bottom end cap
NCYL	User-assigned cylinder number
NEC	User-assigned end cap number
NECO	User-assigned end cap number for "other" end cap
THETAR, PHIR	Angles defining rotation of cylinder axis (radians)
XTRL, YTRL, ZTRL	Translation of cylinder origin
ZLOC	Location of end cap center (cylinder coordinate system)

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
ICYTAG	/GTDDAT/
IECTAG	/GTDDAT/
IP217	/GEODAT/
ISGTBL	/SEGMNT/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
ITAGID	/GTDDAT/
LUPRNT	/ADEBUG/

MAXBLK	/SEGMNT/
MAXSEG	/SEGMNT/
NPATCH	/SEGMNT/
NTPGTD	/GTDDAT/
NUMSEG	/SEGMNT/
NWIRE	/SEGMNT/
SCALE	/SEGMNT/
SEGTBL	/SEGMNT/
B. OUTPUT	LOCATION
ISGTBL	/SEGMNT/
NOGOFG	/SCNPAR/
SEGTBL	/SEGMNT/
UPDBLK	/SEGMNT/

6. CALLING ROUTINE:

GEODRV

7. CALLED ROUTINES:

ASSIGN

CNVGTD

GETPNT

GETSEG

GTDCS

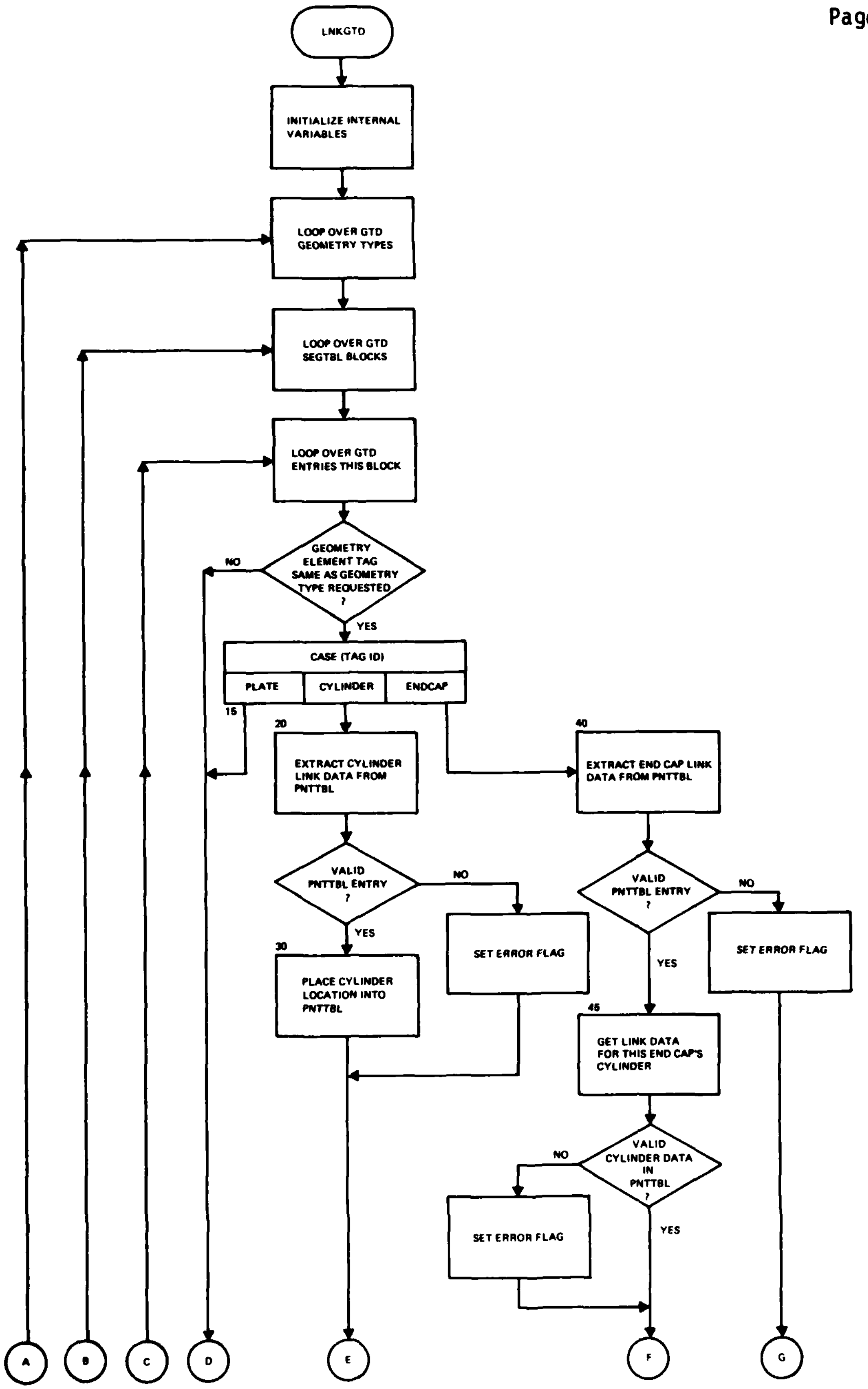
PUTPNT

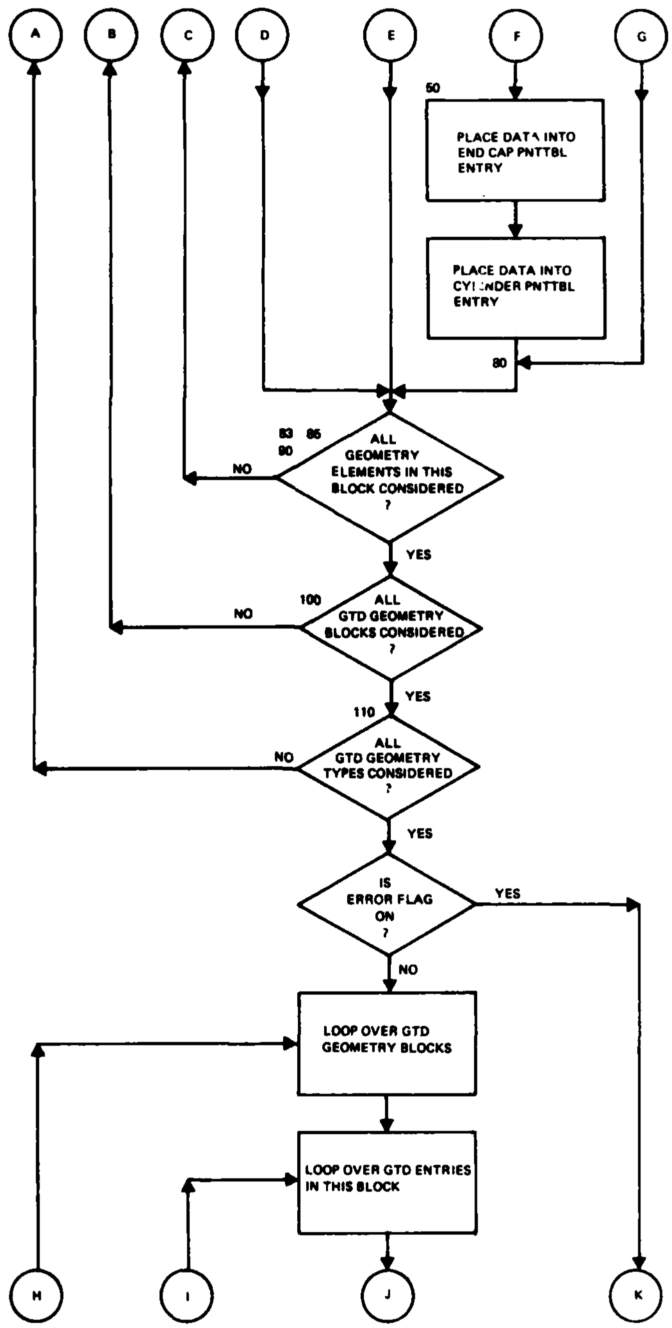
STATIN

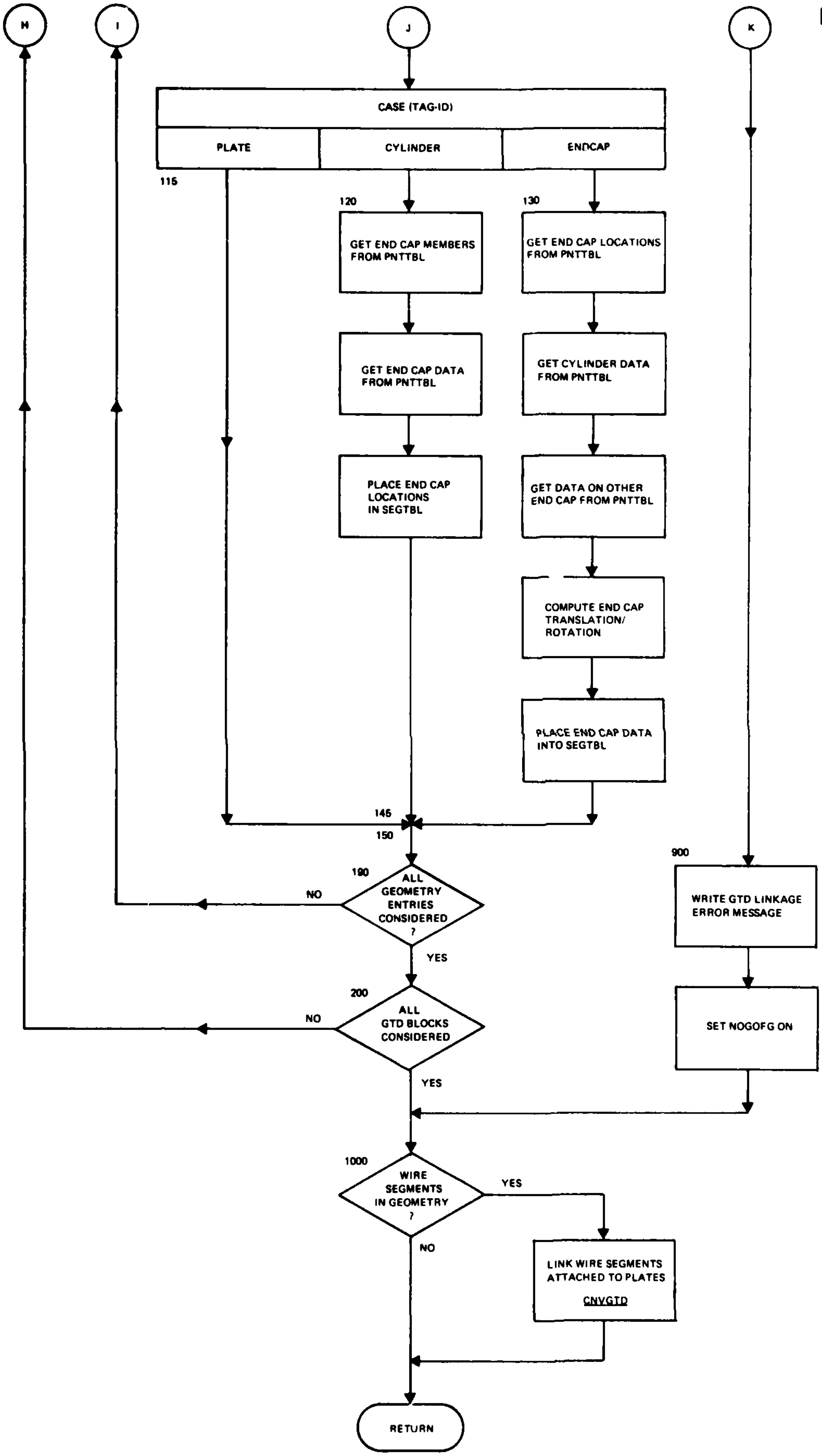
STATOT

WLKBCK

LNKGTD (INPUT)




A
B
C
D
E
F
G
50
PLACE DATA INTO END CAP PNTTBL ENTRY
PLACE DATA INTO CYLINDER PNTTBL ENTRY
80
83 86 90
ALL GEOMETRY ELEMENTS IN THIS BLOCK CONSIDERED ?
NO
YES
100
ALL GTD GEOMETRY BLOCKS CONSIDERED ?
NO
YES
110
ALL GTD GEOMETRY TYPES CONSIDERED ?
NO
YES
IS ERROR FLAG ON ?
YES
NO
LOOP OVER GTD GEOMETRY BLOCKS
LOOP OVER GTD ENTRIES IN THIS BLOCK
H
I
J
K


H
I
J
K
CASE (TAG-ID)
PLATE
CYLINDER
ENDCAP
115
120
GET END CAP MEMBERS FROM PNTTBL
GET END CAP DATA FROM PNTTBL
PLACE END CAP LOCATIONS IN SEGTBL
130
GET END CAP LOCATIONS FROM PNTTBL
GET CYLINDER DATA FROM PNTTBL
GET DATA ON OTHER END CAP FROM PNTTBL
COMPUTE END CAP TRANSLATION/ ROTATION
PLACE END CAP DATA INTO SEGTBL
145
150
190
ALL GEOMETRY ENTRIES CONSIDERED ?
NO
YES
200
ALL GTD BLOCKS CONSIDERED
NO
YES
900
WRITE GTD LINKAGE ERROR MESSAGE
SET NOGOFG ON
1000
WIRE SEGMENTS IN GEOMETRY ?
YES
NO
LINK WIRE SEGMENTS ATTACHED TO PLATES
CNVGTD
RETURN

1. NAME: LNKJCT (INPUT)

2. PURPOSE: This subroutine sets the circular linked list for multiple junctions.

3. METHOD: All segments having the same multiple junction are scanned in ascending sequence with the segment pointing to the correct end of the following segment which is connected at the given junction. The last segment found connected to the junction is set to point back to the first segment found at the given junction.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IFIRST	Flag indicating first segment of junction found
IJCT	Flag set to the segment number connected to the junction. Flag is positive if the positive end of the segment is connected. Flag is negative if the negative end of the segment is connected
ILAST	Last segment connected to a given junction
IOD	Index to output array for junction information
IS	Segment number in SEGTBL
ISG	Segment number being analyzed
IWORDS	Junction array
JBIAS1	Integer to bias connection data to end one of the segment
JBIAS2	Integer to bias connection data to end two of the segment
JBIAS3	Integer to bias connection data to a patch
JCBIAS	Integer to indicate a multiple junction connection
JCT	Word indicating end and number of segment connected to multiple junction

JCTND1	Junction information for end 1 of segment
JCTND2	Junction information for end 2 of segment
JCTNUM	Biased number of multiple junctions being considered
JCTWRD	Junction word for segment entry in SEGTBL
LIMSEG	Number of segment entries in requested data block
LSTBLK	Index to data block of last segment connected to junction
LSTIOD	Maximum number of entries in IWORDS array
LSTNDX	Index to SEGTBL entry of last segment connected to junction
MAXSEG	Maximum number of segments per data block
NDXBLK	Index to data block currently in use
NOGOFG	No go flag
NUMJCT	Number of junctions

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
IP217	/GEODAT/
ISGTBL	/SEGMNT/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
JBIAS1	/SEGMNT/
JBIAS2	/SEGMNT/
JBIAS3	/SEGMNT/
JCBIAS	/SEGMNT/

LSTIOD	/ADEBUG/
LUPRNT	/ADEBUG/
MAXSEG	/SEGMNT/
MLTJCT	/SEGMNT/
NPATCH	/SEGMNT/
NWIRE	/SEGMNT/
SEGTBL	/SEGMNT/
B. OUTPUT	LOCATION
ISGTBL	/SEGMNT/
NOGOFG	/ADEBUG/
SEGTBL	/SEGMNT/
UPDBLK	/SEGMNT/

6. CALLING ROUTINE:

GEODRV

7. CALLED ROUTINES:

ASSIGN

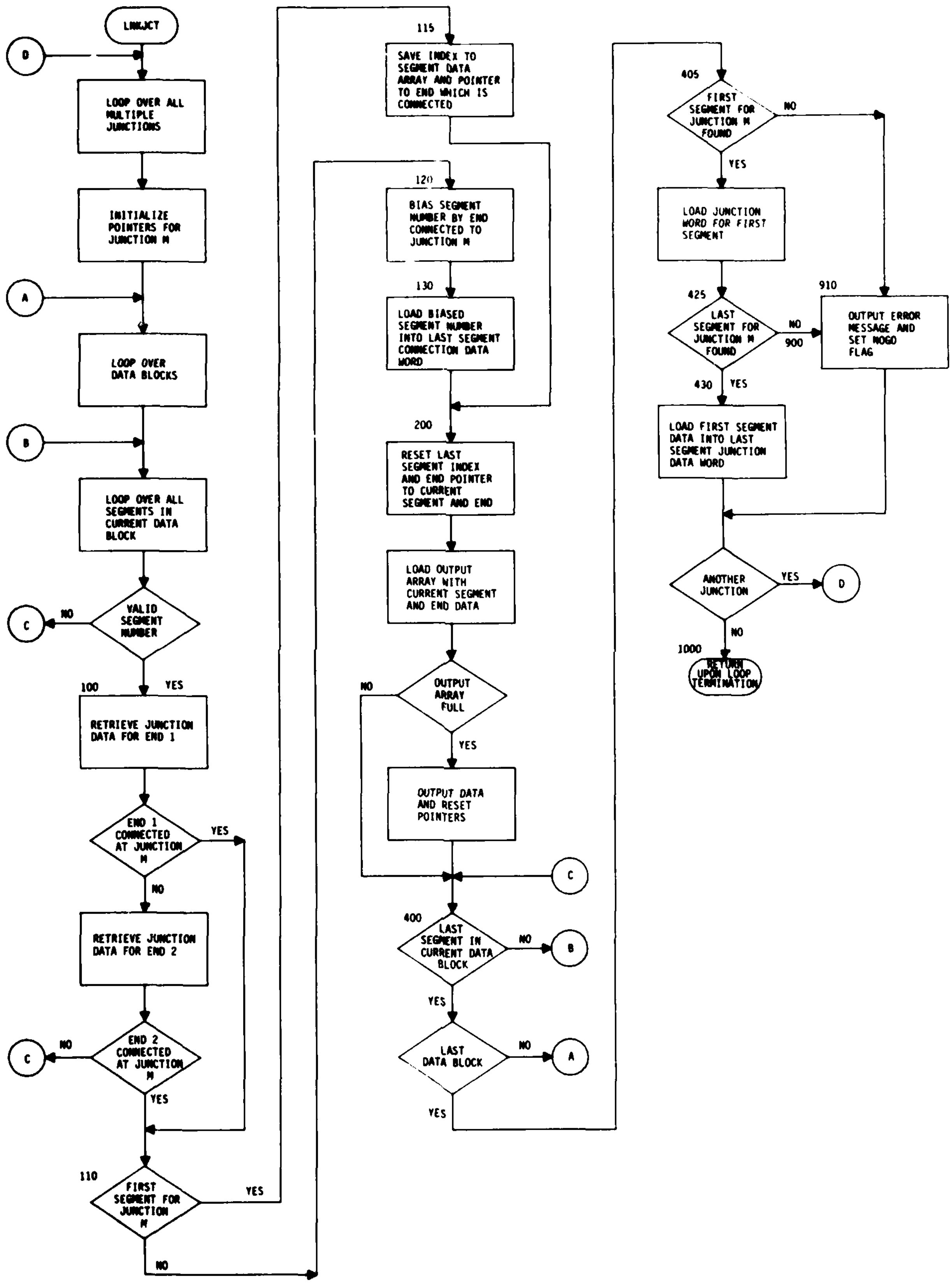
GETSEG

STATIN

STATOT

WLKBCK

LNKJCT (INPUT)



1. **NAME:** LODDRV (MOM)

2. **PURPOSE:** To generate structure loading

3. **METHOD:** Subroutine LODDRV retrieves the user specified geometry data set and the electrical parameters on the command card. If the frequency is the same as on the previous entry, the load data set will be appended to any previously specified load data. Otherwise, the data set will be reinitialized before new load data are computed. The impedance elements are calculated based on the type of load requested and are entered into the data set specified at the index identified by the segments loaded.

4. **INTERNAL VARIABLES:**

VARIABLE	DEFINITION
CAPS	User input capacitance
FRQSAV	Internal variable for frequency
HNRY	User input inductance
IBLK	Data block containing current segment number
IDTYPE	Flag for segments or tags to be loaded
IP	Array pointer
IS	Segment number
ISEG1	First segment loaded
ISEG2	Last segment loaded
ITAG	Tag index
ITAG1	First tag ID loaded
ITAG2	Last tag ID loaded
JSEG	Segment number
KWLDTP	Keyword indicating load type
LNKLS	Pointer to geometry data set
LOAD	Output header array

LOCI	Location of imaginary component
LOCR	Location of real component
LODTYP	Integer indicating type of load
NAMGOM	Geometry data set name
NAMLDS	Load data set name
NBITLS	Attribute word for load data set
NDXGM	Index to geometry data set
NDXGOM	Index to default geometry name in NCODES array
NDXLS	Index to load data set
NOTTAG	Logical .TRUE. for segment specification
NPRROW	Number of words per row
NROWS	Number of rows
NXTARG	Next argument in INTARG array
OHMS	User supplied resistance
RI	Output array of imaginary part of segment impedance
RL	Output array of real part of segment impedance
SEGR	Segment radius
ZC	Capacitive impedance
ZI	Imaginary component of impedance
ZIMP	Internal impedance of a circular wire
ZL	Inductive impedance
ZM	Magnitude of impedance
ZP	Phase of impedance
ZR	Real component of impedance

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
CLITE	/AMPZIJ/
DGTORD	/GEODAT/
FLTARG	/ARGCOM/
FRQMHZ	/AMPZIJ/
INTARG	/ARGCOM/
IPASS	/ARGCOM/
IP217	/GEODAT/
ISEG	/SEGMNT/
ISGTBL	/SEGMNT/
ISON	/ADEBUG/
KBCPLX	/PARTAB/
KBLOAD	/PARTAB/
KOLCOL	/PARTAB/
KOLLNK	/PARTAB/
KOLNAM	/PARTAB/
KOLROW	/PARTAB/
KWCOND	/PARTAB/
KWPRLC	/PARTAB/
KWSEGS	/PARTAB/
KWSRLC	/PARTAB/
KWTAGS	/PARTAB/
KWZIMP	/PARTAB/
LUPRNT	/ADEBUG/
MAXSEG	/SEGMNT/

NCODES	/PARTAB/
NDATBL	/PARTAB/
NDXBLK	/SEGMNT/
NOPCOD	/ADEBUG/
NPDATA	/PARTAB/
NUMARG	/ARGCOM/
SEGTBL	/SEGMNT/
TWOPI	/AMPZIJ/
UPDBLK	/SEGMNT/
ZERO	/ADEBUG/

B. OUTPUT LOCATION

FRQMHZ	/AMPZIJ/
IERRF	/ADEBUG/
MAXBLK	/SEGMNT/
NAMSEG	/SEGMNT/
NOGOFG	/ADEBUG/
NTFLPT	/ADEBUG/
NUMSEG	/SEGMNT/
TEMP	/TEMP01/
WAVLGH	/AMPZIJ/
WAVNUM	/AMPZIJ/

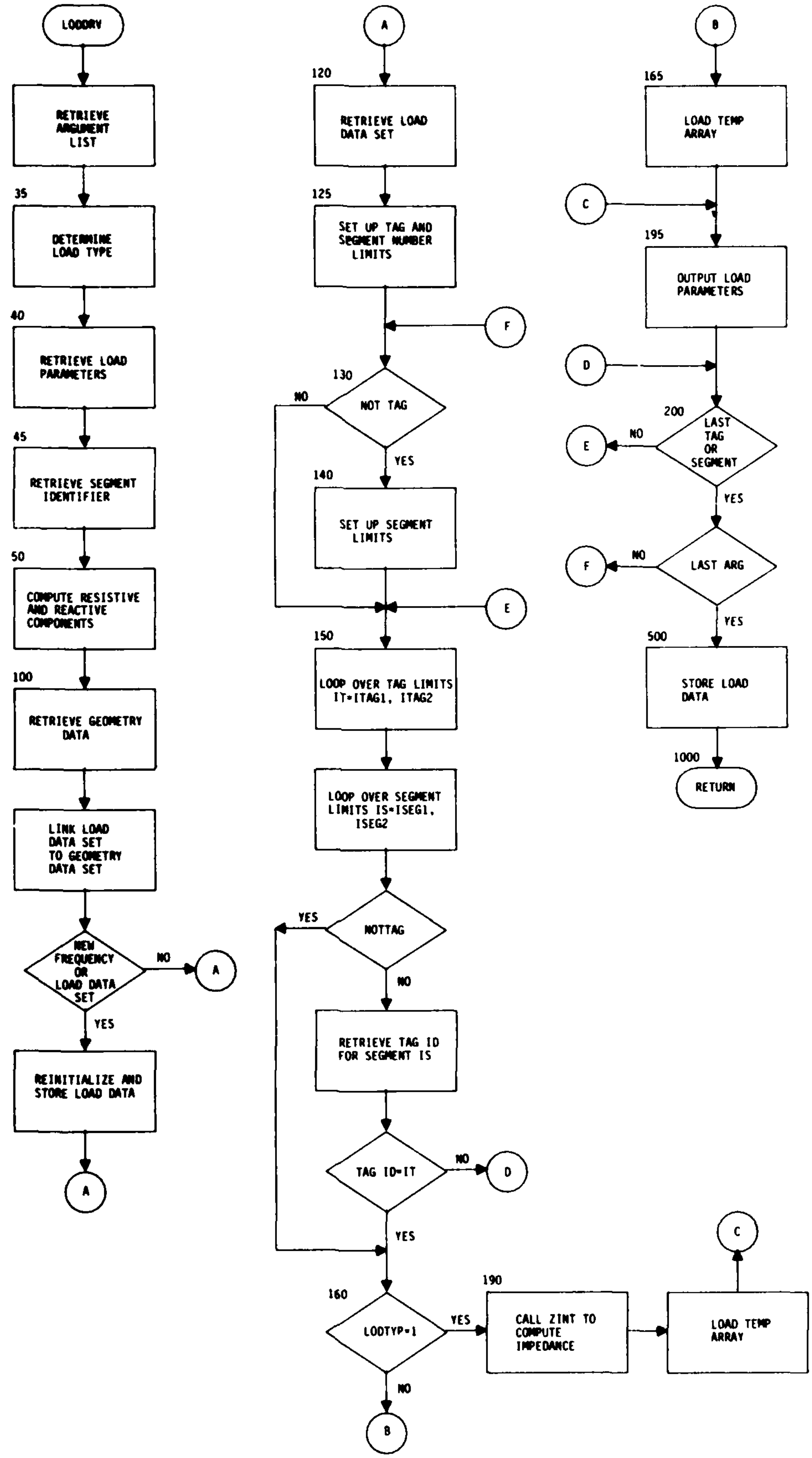
6. CALLING ROUTINE:

TSKXQT

7. CALLED ROUTINES:

ASSIGN	PUTSYM
CONVRT	STATIN
ERROR	STATOT
GETARG	SYMDEF
GETSEG	SYMUPD
GETSYM	WLKBCK
	ZINT

LODDRV (MOM)



LODDRV
RETRIEVE ARGUMENT LIST
35
DETERMINE LOAD TYPE
40
RETRIEVE LOAD PARAMETERS
45
RETRIEVE SEGMENT IDENTIFIER
50
COMPUTE RESISTIVE AND REACTIVE COMPONENTS
100
RETRIEVE GEOMETRY DATA
LINK LOAD DATA SET TO GEOMETRY DATA SET
NEW FREQUENCY OR LOAD DATA SET
NO
A
YES
REINITIALIZE AND STORE LOAD DATA
A
A
120
RETRIEVE LOAD DATA SET
125
SET UP TAG AND SEGMENT NUMBER LIMITS
F
130
NOT TAG
NO
YES
140
SET UP SEGMENT LIMITS
E
150
LOOP OVER TAG LIMITS IT=ITAG1, ITAG2
LOOP OVER SEGMENT LIMITS IS=ISEG1, ISEG2
YES
NOTTAG
NO
RETRIEVE TAG ID FOR SEGMENT IS
TAG ID=IT
NO
D
YES
160
LODTYP=1
YES
190
CALL ZINT TO COMPUTE IMPEDANCE
LOAD TEMP ARRAY
C
NO
B
B
165
LOAD TEMP ARRAY
C
195
OUTPUT LOAD PARAMETERS
D
200
LAST TAG OR SEGMENT
NO
E
YES
LAST ARG
NO
F
YES
500
STORE LOAD DATA
1000
RETURN

1. NAME: LODSYM (MOM)

2. PURPOSE: Determine if structure loads have same symmetry as structure.

3. METHOD: Directly compares loads on symmetrical structure elements.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
KSYM	Symmetry flag
LOADSM	Load symmetry flag
NPRCEL	Number of elements per symmetry cell
NSYM	Symmetry variable NSYM > 0: rotational symmetry NSYM = 0: no operation NSYM < 0: plane symmetry
NZ	Dimension of load vector
Z	Input load vector
ZD	Load difference between elements
ZDMAX	Maximum difference allowable for load symmetry
ZDSQ	ZD^2
ZN	Load on n^{th} element

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
ISOFF	/ADEBUG/
ISON	/ADEBUG/
NSYM	F.P.
NZ	F.P.
Z	F.P.

B. OUTPUT LOCATION

OUTPUT	LOCATION
LOADSM	F.P.

6. CALLING ROUTINE:

   ZIJDRV

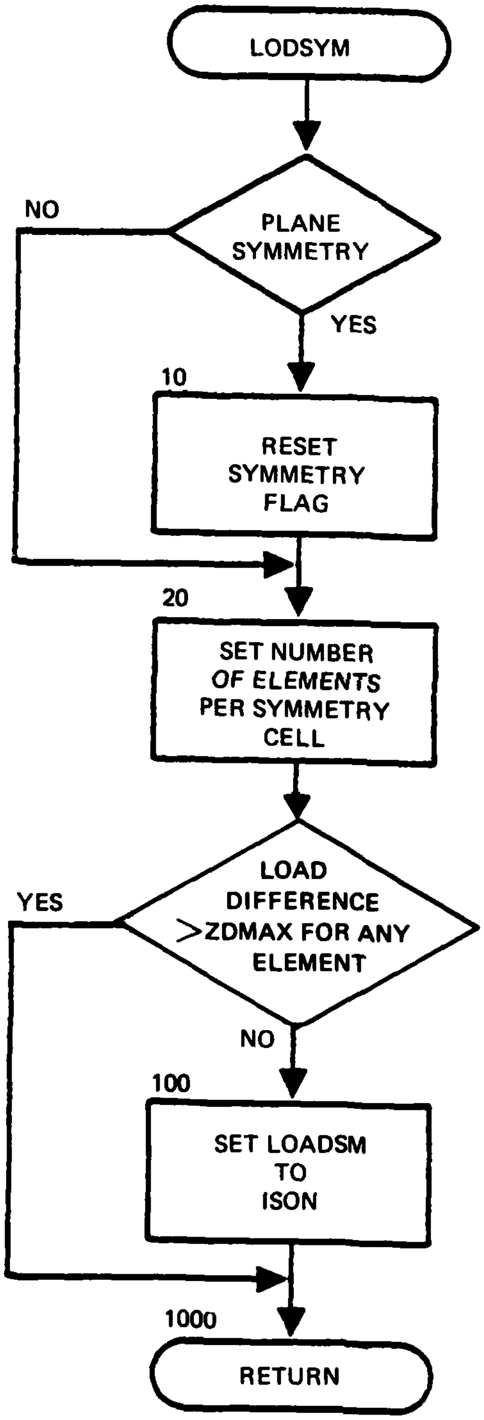
7. CALLED ROUTINES:

   ASSIGN

   STATIN

   STATOT

   WLKBCK


LODSYM
PLANE SYMMETRY
NO
YES
10
RESET SYMMETRY FLAG
20
SET NUMBER OF ELEMENTS PER SYMMETRY CELL
LOAD DIFFERENCE >ZDMAX FOR ANY ELEMENT
YES
NO
100
SET LOADSM TO ISON
1000
RETURN

1. NAME: LUDDRV (MOM)

2. FUNCTION: Subroutine to execute DECOMP command.

3. METHOD: Retrieve arguments on the command and determine if decomposed matrix name is the same as the matrix to be decomposed; if not, copy the matrix to be decomposed into the matrix that is the result of the operation. Determine if the matrix will fit in core. If not, set up variables logically linked to the matrix to be decomposed for the lower and upper triangular decomposed matrices. Call subroutine DECOMP to execute the lower-upper triangular decomposition by rows.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IBAND	Flag indicating if matrix to be decomposed is banded
ILOWER	Integer representation of right justified BCD characters LWR
INDX	Index to the NCODES array for the pivot keyword specified
INDXA	Index to the location of the matrix to be decomposed in the NDATBL array
INDXB	Index to the decomposed matrix in the NDATBL array
IOBL	Logical unit to contain the lower triangular matrix
IOBU	Logical unit to contain the upper triangular matrix
IOS1	Logical unit designation for scratch file 1
IOS2	Logical unit designation for scratch file 2
IPVIT	Pivot argument on command
IREC1	First record to be retrieved
IREC2	Last record to be retrieved
IUPPER	Integer representation of right justified BCD characters UPR

LINKA	Logical linkage for the decomposed matrix
LOCA	Logical unit designation for matrix to be decomposed
LOCAIJ	Location of first element of matrix to be decomposed
MORE	Used for bits that may be set
MXBAND	The half bandwidth plus 1 of the matrix to be decomposed
NAMEA	The symbolic name of the matrix to be decomposed
NAMEB	The symbolic name of the decomposed matrix
NAMEP	Pivoting keyword name
NAMLWR	Symbolic name of the lower triangular matrix when the matrix will not reside in core
NAMSYM	Symbolic name
NAMUPR	Symbolic name of the upper triangular matrix
NBITA	Attribute bit set word of matrix to be decomposed
NBITL	Attribute bit set word for lower triangular matrix
NBITU	Attribute bit set word for upper triangular matrix
NCOLA	Number of columns in matrix to be decomposed
NCOLS	The number of columns of the original matrix from which the matrix to be decomposed was generated
NCORE	The number of columns of the matrix to be decomposed which will reside in core simultaneously

NDXL	Index to the symbol table for the lower triangle of a matrix
NDXU	Index to the symbol table for the upper triangle of a matrix
NMAT	Number of a submatrix
NPRCOL	The number of computer words per column of the matrix to be decomposed
NPRELM	The number of computer words per element of the matrix to be decomposed
NPRGET	The number of records per call when copying the matrix to be decomposed into the symbol of the decomposed matrix
NROWA	The number of rows in the matrix to be decomposed
NROWS	The number of rows in the matrix from which the matrix to be decomposed was derived
NSHIFT	Bit shifter
NSYMBL	Number of entries in the symbol table
NUMMAT	Number of submatrices in original matrix
NUPPER	The number of computer words per row of the upper triangular matrix
N1	First submatrix to be decomposed

5. I/O VARIABLES

A. INPUT

INPUT	LOCATION
INTARG	/ARGCOM/
IOFILE	/IOFLES/
IOSCR1	/SYSFIL/
IOSCR2	/SYSFIL/
IPASS	/ARGCOM/

ISOFF	/ADEBUG/
ISON	/ADEBUG/
KBBAND	/PARTAB/
KBCPLX	/PARTAB/
KBDPRE	/PARTAB/
KBLWRT	/PARTAB/
KBUPRT	/PARTAB/
KOLAST	/PARTAB/
KOLBIT	/PARTAB/
KOLCOL	/PARTAB/
KOLFST	/PARTAB/
KOLLNK	/PARTAB/
KOLLOC	/PARTAB/
KOLNAM	/PARTAB/
KOLROW	/PARTAB/
KWNAME	/PARTAB/
LUPRNT	/ADEBUG/
MAXSEG	/SEGMNT/
NBYTSZ	/ADEBUG/
NCODES	/PARTAB/
NDATBL	/PARTAB/
NDXBLK	/SEGMNT/
NOPCOD	/ADEBUG/
NPDATA	/PARTAB/
NPRSEG	/SEGMNT/

NTEMPS	/TEMPO1/
NUMSEG	/SEGMNT/
RSTART	/SYSFIL/
SEGTBL	/SEGMNT/

B. OUTPUT LOCATION

IERRF	/ADEBUG/
ISGTBL	/SEGMNT/
NDATBL	/PARTAB/
SEGTBL	/SEGMNT/
TEMP	/TEMPO1/
UPDBLK	/SEGMNT/

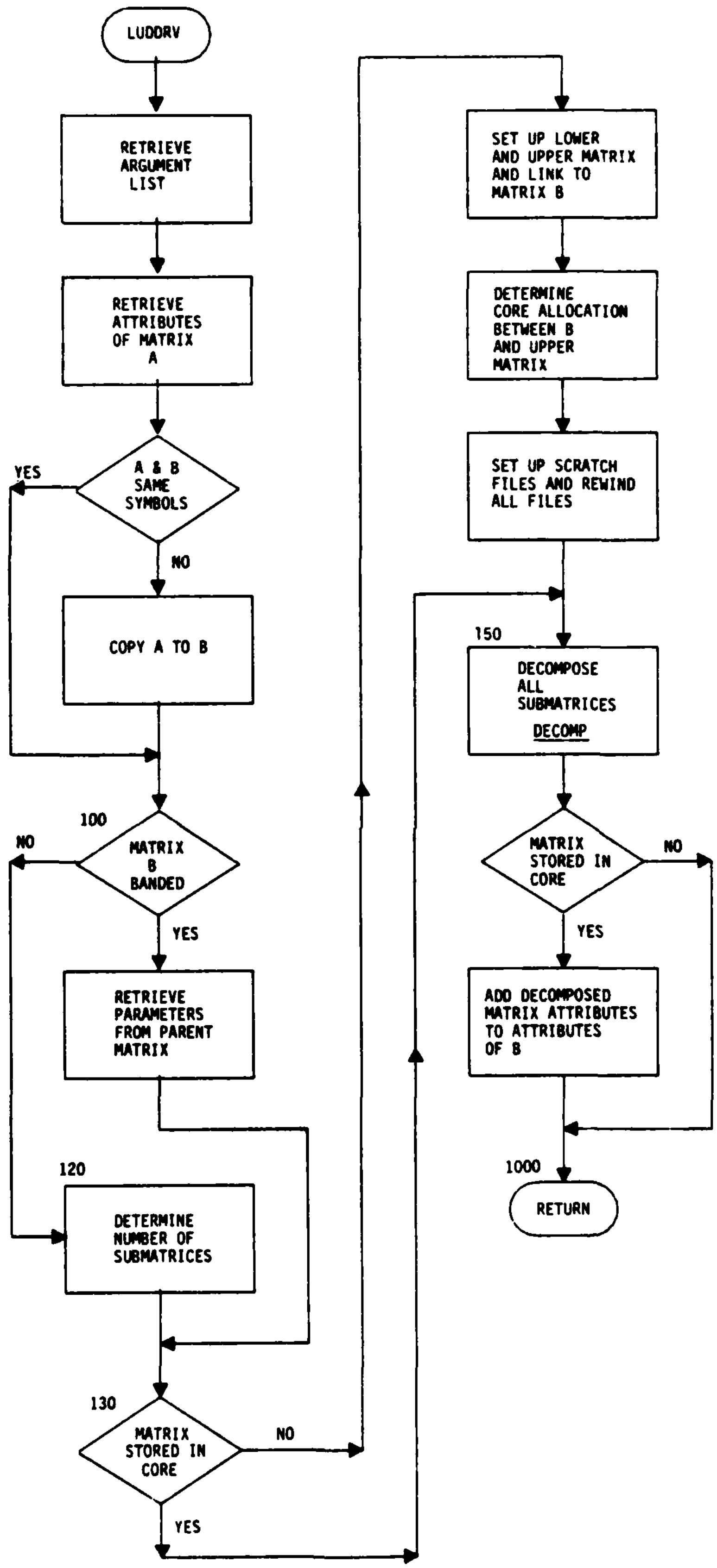
6. CALLING ROUTINE:

TSKXQT

7. CALLED ROUTINES:

ASSIGN	PUTSYM
CONVRT	STATIN
DECOMP	STATOT
ERROR	SYMDEF
GETSEG	SYMUPD
GETSYM	WLKBCK
IBITCK	

LUDDRV (MOM)


LUDDRV
RETRIEVE ARGUMENT LIST
RETRIEVE ATTRIBUTES OF MATRIX A
A & B SAME SYMBOLS
YES
NO
COPY A TO B
100
MATRIX B BANDED
NO
YES
RETRIEVE PARAMETERS FROM PARENT MATRIX
120
DETERMINE NUMBER OF SUBMATRICES
130
MATRIX STORED IN CORE
NO
YES
SET UP LOWER AND UPPER MATRIX AND LINK TO MATRIX B
DETERMINE CORE ALLOCATION BETWEEN B AND UPPER MATRIX
SET UP SCRATCH FILES AND REWIND ALL FILES
150
DECOMPOSE ALL SUBMATRICES
DECOMP
MATRIX STORED IN CORE
NO
YES
ADD DECOMPOSED MATRIX ATTRIBUTES TO ATTRIBUTES OF B
1000
RETURN

1. NAME: LUSTAT (INPUT)

2. PURPOSE: To check a peripheral file for an EOF.

3. METHOD: Reads each command on the user input data deck. On next call after last command an end-of-file (EOF) is returned.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
LSTCOL	Last column of the command to be read
LUTASK	Logical unit on which the command resides
NCARD	Command array
NDFILE	Flag indicating end-of-file status on LUTASK

5. I/O VARIABLES:

A. INPUT	LOCATION
LSTCOL	F.P.
LUTASK	F.P.
NCARD	F.P.

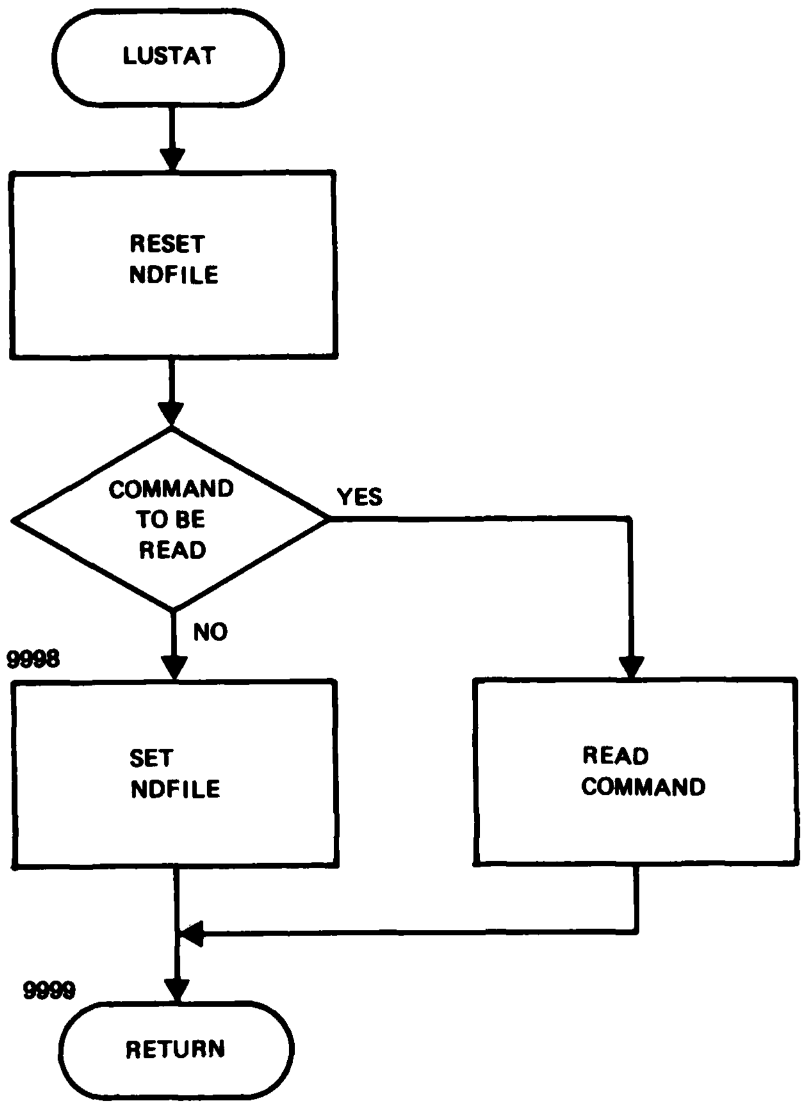
B. OUTPUT	LOCATION
NDFILE	F.P.

6. CALLING ROUTINE:

SCAN

7. CALLED ROUTINES:

NONE


LUSTAT
RESET NDFILE
COMMAND TO BE READ
YES
NO
9998
SET NDFILE
READ COMMAND
9999
RETURN

1. NAME: MOVFIL (GTD, INPUT, MOM, OUTPUT)

2. FUNCTION: To position the peripheral files.

3. METHOD: The file requested is positioned for the number of words requested. If the file is to be moved backwards, it is rewound and then read forward to the correct word. If the file is to be positioned forward, it simply reads the correct number of words to achieve the position desired.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IFILE	Logical unit to be positioned
IFLE	Internal variable for IFILE
MOVE	Internal variable for number of words file is to be moved
MOVWRD	Argument specifying number of words file is to be moved
NBUFS	Number of reads required to move file
NPRBUF	The number of words per read for the file

5. I/O VARIABLES:

A.

INPUT	LOCATION
DBGPRT	/ADEBUG/
IFILE	F.P.
IOFILE	/IOFLES/
ISON	/ADEBUG/
LUPRNT	/ADEBUG/
MOVWRD	F.P.
NPRBUF	F.P.
NTEMPS	/TEMP01/

B. OUTPUT	LOCATION
IERRF	/ADEBUG/
IOFILE	/IOFLES/
TEMP	/TEMPO1/

6. CALLING ROUTINES*:

BUBBLE (1)

DECOMP (3)

GEODRV (1)

GETSYM (1,2,3,4)

PUTSYM (1,2,3,4)

SOLDRV (3)

STRTUP (2,3,4)

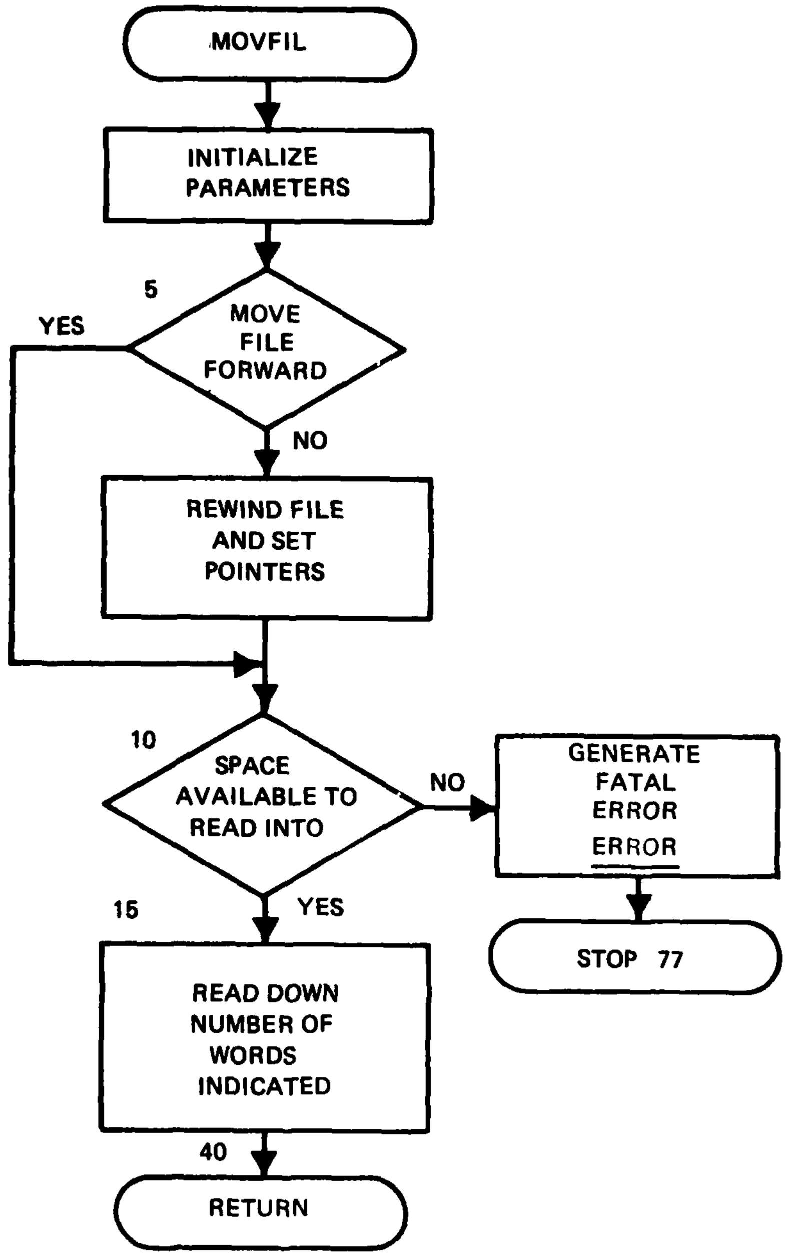
SUBPAT (1)

7. CALLED ROUTINES:

ASSIGN	STATIN
ERROR	STATOT
RDEFIL	WLKBCK

*1-INPUT
2-GTD
3-MOM
4-OUTPUT

MOVFIL (GTD, INPUT, MOM, OUTPUT)



1. NAME: NANDB (GTD)

2. PURPOSE: To calculate the unit vectors for rays normal and tangent to the elliptical cylinder at a given point $\overline{XC}$ (in x-y plane) defined by elliptic angle VR.

3. METHOD: For the point on the cylinder defined by the elliptic angle VR, the unit normal vector is given as:

$$\hat{UN} = \frac{\hat{x}\ B\ \cos(VR) + \hat{y}\ A\ \sin(VR)}{\sqrt{B^2\ \cos^2(VR) + A^2\ \sin^2(VR)}}$$

and the unit tangent vector is given by:

$$\hat{UT} = \frac{-\hat{x}\ A\ \sin\ (VR) + \hat{y}\ B\ \cos\ (VR)}{\sqrt{B^2\ \cos^2(VR) + A^2\ \sin^2(VR)}}$$

as shown in figure 1.

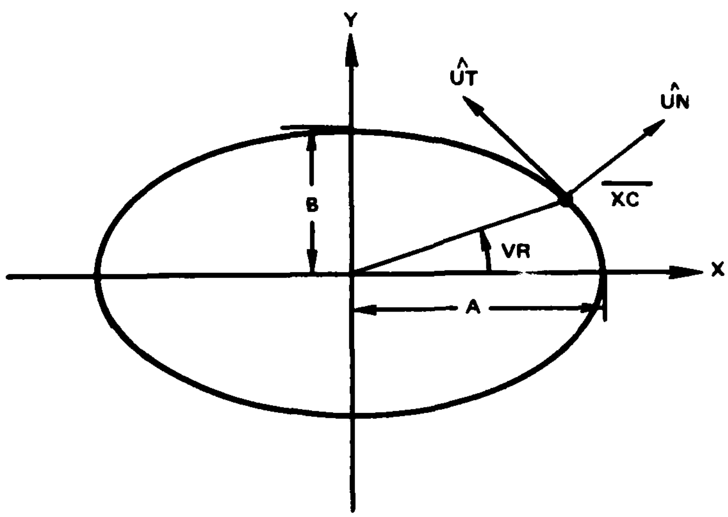


Figure 1. Illustration of Unit Vectors Tangent and Normal to the Cylinder

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	Cylinder radius along x axis.
B	Cylinder radius along y axis.
DN	Magnitude of tangent and normal vectors.
UN	X and Y components of unit vector normal to cylinder in reference coordinate system (RCS).
UT	X and Y components of unit vector tangent to cylinder in reference coordinate system (RCS).
VR	Elliptical angle in RCS x-y plane defining the point on cylinder for which normal and tangent unit vectors are to be calculated.

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
A	/GEOMEL/
B	/GEOMEL/
VR	F.P.

B. OUTPUT

OUTPUT	LOCATION
UN	F.P.
UT	F.P.

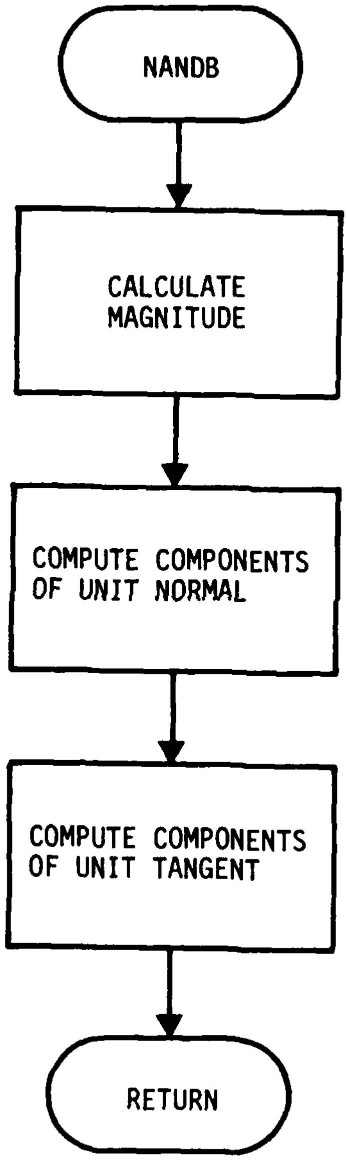
6. CALLING ROUTINES:

DPLRCL	RCLRPL	RPLSCL
ENDIF	REFCYL	SCLRPL
RCLDPL	RPLRCL	SCTCYL

7. CALLED ROUTINES:

None

NANDB (GTD)



NANDB
CALCULATE MAGNITUDE
COMPUTE COMPONENTS OF UNIT NORMAL
COMPUTE COMPONENTS OF UNIT TANGENT
RETURN

1. NAME: NERFLD (MOM)

2. PURPOSE: To compute the x, y, and z components of the electric field of a structure with known currents at any point in space, particularly in the near-field region of the structure. Ground interaction is also taken into account using the reflection coefficient method.

3. METHOD: The electric field of each structure wire segment is calculated by the subroutine TNEFLD where the current is assumed to be

$$\frac{I_i(t)}{\lambda} = I + \sin 2\pi t + \cos 2\pi t$$

The elements of the field corresponding to the sine, cosine, and constant terms are then modified in this routine to reflect the actual current component amplitudes which have been calculated, where the actual current is given by

$$\frac{I_i(t)}{\lambda} = A_i + B_i \sin 2\pi t + C_i \cos 2\pi t$$

The electric field components calculated by TNEFLD are the ρ and z components in a cylindrical coordinate system centered and aligned with the source segment. These components are resolved into the x, y, z coordinate system of the structure, and the contribution from each segment is added. If a ground is present, the contribution of the image of each segment is also calculated and summed. The modification of the image field by the appropriate reflection coefficients is done in subroutine GNDREF.

The coordinates of the observation point must be calculated in the cylindrical coordinate system (ρ, ϕ, z) which subroutine TNEFLD assumes. If the observation point is given by $\bar{r}_o$ and the source segment by $\bar{r}_i$, and the orientation of the segment and cylindrical coordinate system is given by $\hat{u}_i$, then letting

$$\bar{r}_d = \bar{r}_o - \bar{r}_i$$

PREVIOUS PAGE IS BLANK

the location of the observation point is given by

$$\bar{z}_{oi} = (\bar{r}_d \cdot \hat{u}_i)\, \hat{u}_i$$

$$\bar{\rho}_{oi} = \bar{r}_d - \bar{z}_{oi}$$

The electric field due to current density J on each structure patch of area A is calculated by:

$$\bar{E}(\bar{r}_o) = \frac{-j\eta A}{4\pi k}\left[\left(\frac{k^2R^2 - 1 - jkR}{R^3}\right)\bar{J} + \left(\frac{3 - k^2R^2 + j3kR}{R^5}\right)\left(\bar{J}\cdot\bar{R}\right)\bar{R}\right]e^{-jkR}$$

where: $k = 2\pi/\lambda$

$\bar{R}$ = source to field vector

η = 376 ohms (free space impedance)

The total field is found by summing the contribution for each patch and wire object.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AII	Imaginary part of A_i
AIR	Real part of A_i
ARG	ηA in field calculation due to patch source
B	Radius of wire segment
BII	Imaginary part of B_i
BIR	Real part of B_i

CII	Imaginary part of C_i
CIR	Real part of C_i
CTH	Cos(θ) for reflected ray
ERHO	E_ρ^i with actual current amplitude
ERIC,ERIK,ERIS	Imaginary part of (E_ρ^i) for cosine, constant, and sine unit currents, respectively
ERRC,ERRK,ERRS	Real part of (E_ρ^i) for cosine, constant and sine unit currents, respectively.
ETAK	Complex value for ETA
EX,EY,EZ	X,Y,Z components of the electric field from the wire segment at the observation point
EXS,EYS,EZS	Same as EX, EY, EZ for a patch source
EZIC,EZIK,EZIS	Imaginary part of (E_z^i) in the cylindrical coordinate system for cosine, constant, and sine unit current amplitudes, respectively
EZP	E_z^i in cylindrical coordinates for actual segment current
EZRC,EZRK,EZRS	Real part of (E_z^i) in the cylindrical coordinate system for cosine, constant and sine unit current amplitudes, respectively
INCORE	Logical .TRUE. when A,B,C of current expansion are stored in core
IPATCH	Flag indicating patch observation point
IWIRE	Flag indicating local cylindrical coordinate system is used
JX,JY,JZ	Surface current components
LAI	Location of imaginary part of A_i
LAR	Location of real part of A_i
LBI	Location of imaginary part of B_i
LBR	Location of real part of B_i

LCI	Location of imaginary part of C_i
LCR	Location of real part of C_i
PATCH	Logical flag indicating patch source
R	Patch to field point vector
RFL	Variable used for sign changes
RH	ρ
RH2	ρ^2
RMAG	r_d
RS	r_d^2
R3	R^3
R5	R^5
S	Length of wire segment
XD	$(r_d)_x$
XOB	x_o
XYMAG	Intermediate variable
YD	$(r_d)_y$
YOB	y_o
ZD	$(r_d)_z$
ZOB	z_o
ZP	z_{oi}
ZRSIN	Intermediate variable

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
ETA	/AMPZIJ/
FJ	/AMPZIJ/

NERFLD (MOM)

INCORE	F.P.
IOSCR1	/SYSFIL/
IPERF	/AMPZIJ/
ISEG	/SEGMNT/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
KSYMP	/AMPZIJ/
LOCAII	/FLDCOM/
LOCAIR	/FLDCOM/
LOCBII	/FLDCOM/
LOCBIR	/FLDCOM/
LOCCII	/FLDCOM/
LOCCIR	/FLDCOM/
LUPRNT	/ADEBUG/
MAXBLK	/SEGMNT/
NWIRE	/SEGMNT/
SEGTBL	/SEGMNT/
TEMP	/TEMP01/
TWOPI	/AMPZIJ/
WAVNUM	/AMPZIJ/
XOB	F.P.
YOB	F.P.
ZOB	F.P.
ZRATI	/AMPZIJ/

B. OUTPUT	LOCATION
CABJ	/AMPZIJ/
EX	F.P.
EY	F.P.
EZ	F.P.
PX	/AMPZIJ/
PY	/AMPZIJ/
REFH	/AMPZIJ/
REFV	/AMPZIJ/
RHOX	/AMPZIJ/
RHOY	/AMPZIJ/
RHOZ	/AMPZIJ/
S	/AMPZIJ/
SABJ	/AMPZIJ/
SALPR	/AMPZIJ/

6. CALLING ROUTINE:

FLDDRV

7. CALLED ROUTINES:

ASSIGN

GETSEG

GNDREF

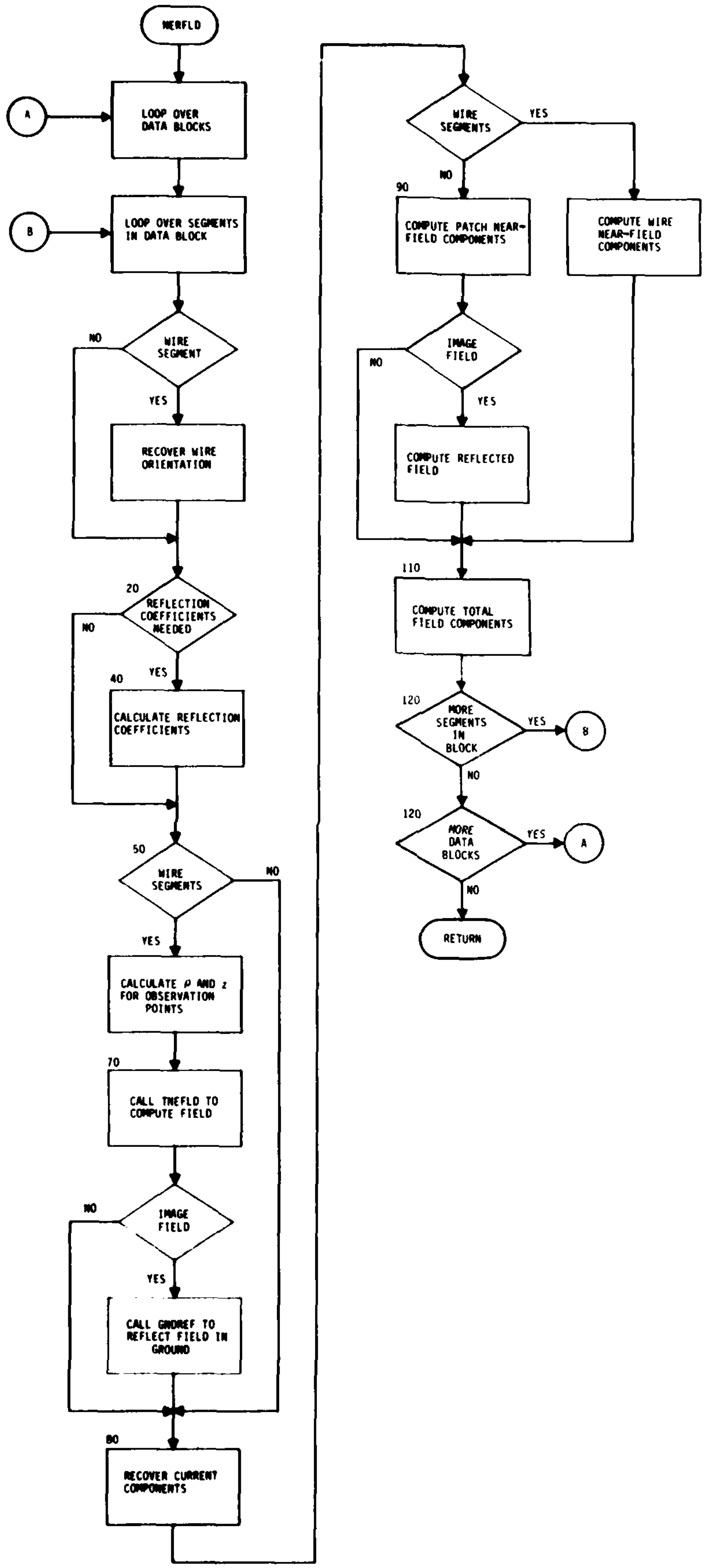
STATIN

STATOT

TNEFLD

WLKBCK

NERFLD (MOM)


NERFLD
A
LOOP OVER DATA BLOCKS
B
LOOP OVER SEGMENTS IN DATA BLOCK
WIRE SEGMENT
NO
YES
RECOVER WIRE ORIENTATION
20
REFLECTION COEFFICIENTS NEEDED
NO
YES
40
CALCULATE REFLECTION COEFFICIENTS
50
WIRE SEGMENTS
NO
YES
CALCULATE ρ AND z FOR OBSERVATION POINTS
70
CALL THEFLD TO COMPUTE FIELD
IMAGE FIELD
NO
YES
CALL GNDREF TO REFLECT FIELD IN GROUND
80
RECOVER CURRENT COMPONENTS
WIRE SEGMENTS
YES
NO
90
COMPUTE PATCH NEAR-FIELD COMPONENTS
COMPUTE WIRE NEAR-FIELD COMPONENTS
IMAGE FIELD
NO
YES
COMPUTE REFLECTED FIELD
110
COMPUTE TOTAL FIELD COMPONENTS
120
MORE SEGMENTS IN BLOCK
YES
B
NO
120
MORE DATA BLOCKS
YES
A
NO
RETURN

1. NAME: NFD (GTD)

2. PURPOSE: To compute the unit vector between two points and the theta and phi angles of the vector in the reference coordinate system (RCS). This is used in the near-field calculations.

3. METHOD: Vector algebra is used to compute the unit vector (D) between points FLDPT and X, and to compute the theta (THSR) and phi (PHSR) angles. Figure 1 shows an illustration of these quantities.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
D	The unit vector between point X and point FLDPT. The x, y, z components are stored in D(1), D(2), D(3) respectively
FLDPT	Ending point of vector
FPTMAG	Magnitude of vector between point X and point FLDPT
PHSR	Phi angle of unit vector D
THSR	Theta angle of unit vector D
X	Starting point of vector

5. I/O VARIABLES:

A. INPUT	LOCATION
FLDPT	F.P.
X	F.P.

B. OUTPUT	LOCATION
D	F.P.
PHSR	F.P.
THSR	F.P.

NFD (GTD)

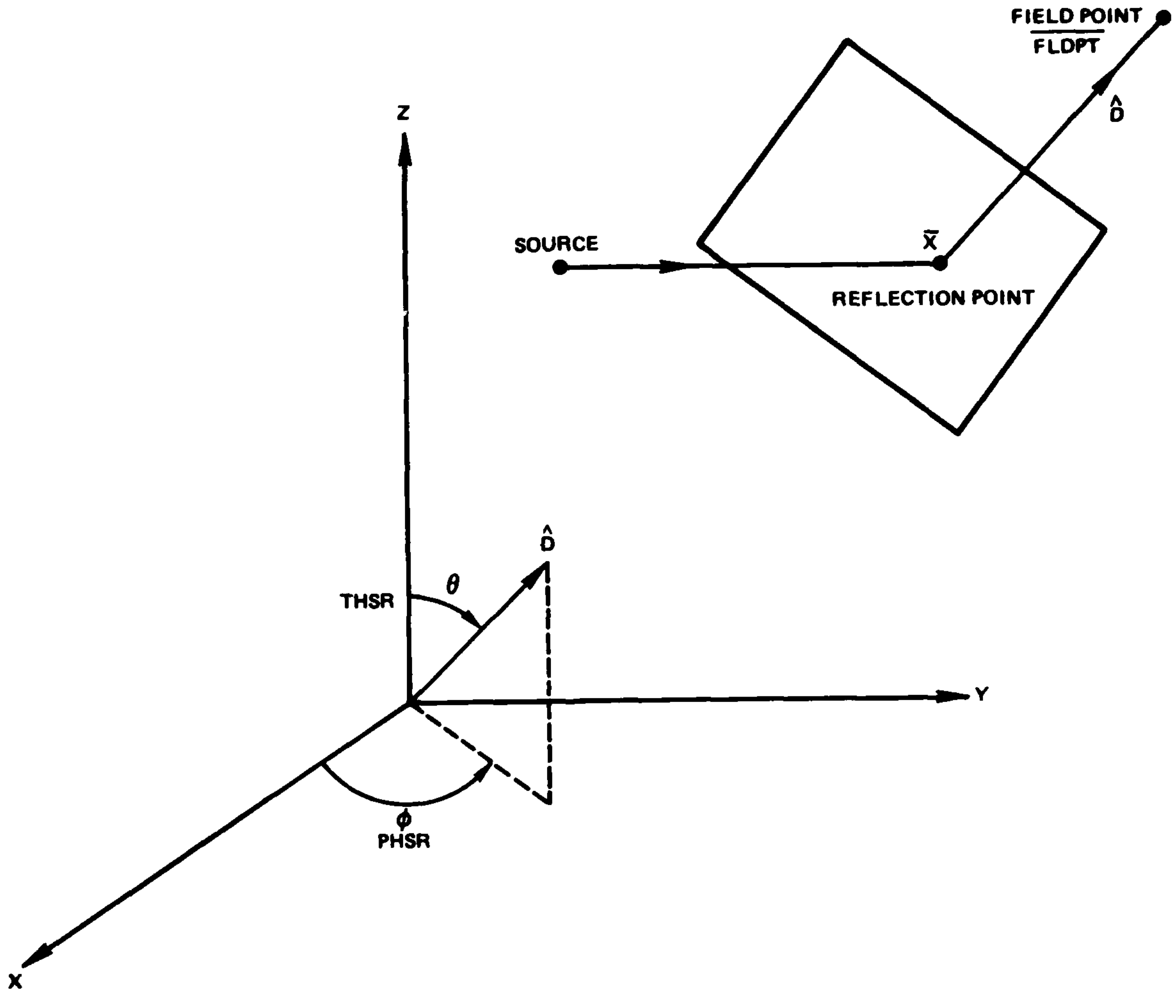


Figure 1. Illustration of the Reference Coordinate System Angles (THSR, PHSR) for the Near-Field Distance of a Ray Propagating from a Reflection Point on a Plate to a Field Point

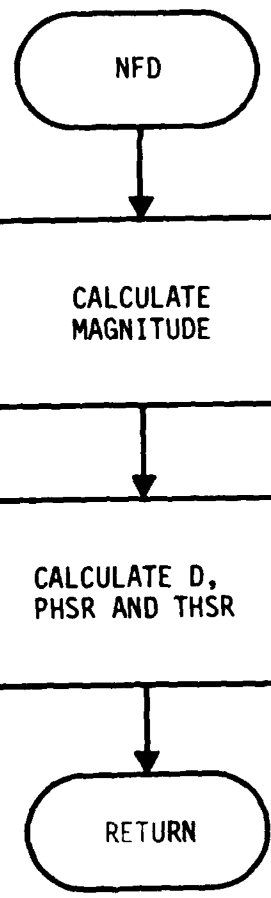
6. CALLING ROUTINES:

DFPTCL	INCFLD	RPLDPL
DFPTWD	RCLDPL	RPLRCL
DIFPLT	RCLRPL	RPLRPL
DPLRCL	REFCAP	RPLSCL
DPLRPL	REFCYL	SCLRPL
ENDIF	REFPLA	SCTCYL
GTDDRV	RFDFPT	

7. CALLED ROUTINE:

NONE

NFD (GTD)



1. NAME: NTGRAN (GTD, MOM)

2. PURPOSE: To compute the value of the integrand for numerical integration.

3. METHOD: The real and imaginary part of the integrand is computed. When a wire is located at an observation point, the integrand

$$I_w = e^{-jkr}/kr$$

is evaluated, and for a patch located at an observation point, the integrand

$$I_p = \left[\frac{1}{(kr)^3} + \frac{j}{(kr)^2}\right] e^{-jkr}$$

is evaluated.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CO	Real part of the integrand
CRK	cos (kr)
IJ	Flag for numerical integration
IPATCH	Flag indicating a patch observation point
RHK	$k\rho$
RK	$k\left[\rho'^2 + z'^2\right]^{1/2}$ for wire observation point; kr for patch observation point
RKB2	$k\rho'$
RK2	k^2r^2
RRK2	$1/k^2r^2$
RRK3	$1/k^3r^3$
SI	Imaginary part of the integrand

SRK	sin (kr)
ZDK	The wavenumber times the difference in polar z coordinates
ZK	Wavenumber times the z coordinate of the source point
ZPK	Wavenumber times the polar z coordinate of the observation point

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
IJ	/TMI/
IPATCH	/TMI/
ISON	/ADEBUG/
RHK	/TMI/
RKB2	/TMI/
ZK	F.P.
ZPK	/TMI/

B. OUTPUT

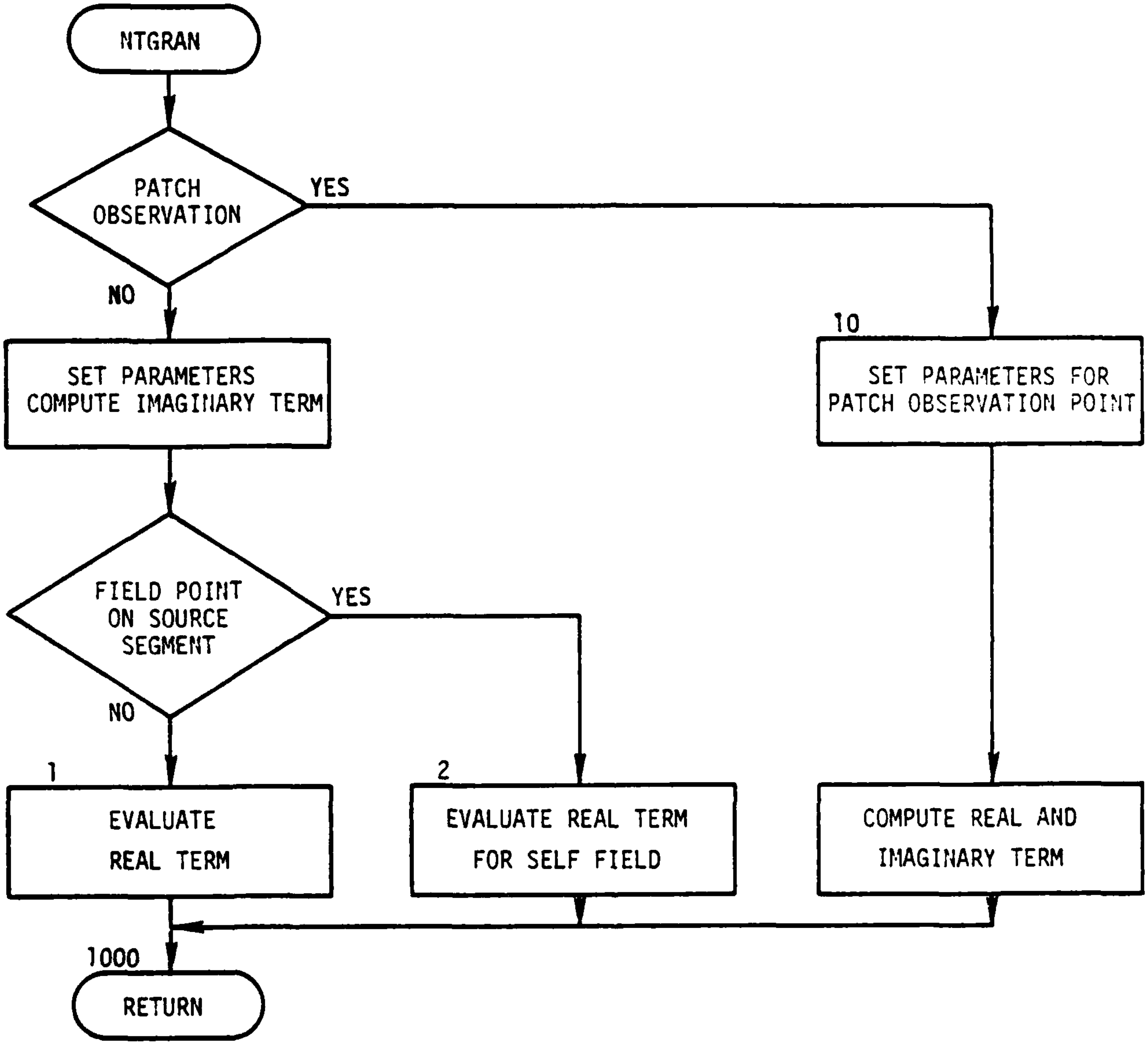
OUTPUT	LOCATION
CO	F.P.
SI	F.P.

6. CALLING ROUTINE:

ROMBNT

7. CALLED ROUTINES:

NONE


NTGRAN
PATCH OBSERVATION
YES
NO
10
SET PARAMETERS COMPUTE IMAGINARY TERM
SET PARAMETERS FOR PATCH OBSERVATION POINT
FIELD POINT ON SOURCE SEGMENT
YES
NO
1
2
EVALUATE REAL TERM
EVALUATE REAL TERM FOR SELF FIELD
COMPUTE REAL AND IMAGINARY TERM
1000
RETURN

1. NAME: NTRPLT (MOM)

2. PURPOSE: To compute the contributions to the matrix elements due to the current on a single segment.

3. METHOD: Subroutine NTRPLT is called by subroutine ZIJSET to compute the fields tangent to segment i or patch i produced by the three components of the current basis function on segment j. The current basis function on segment j is

$$I_j(s) = A_j + B_j \sin k (s - s_j) + C_j \cos k (s - s_j)$$

Expressions for A_j, B_j, and C_j in terms of the currents at the center of segment j and at the center of adjacent segments are given in the discussion of subroutine CABC. When filling the matrix, the currents at the segment centers are unknown and hence are factored out, putting the current in the form

$$I_j(s) = I_\ell X_j(s) + I_j Y_j(s) + I_k Z_j(s)$$

where

$$X_j(s) = \frac{1}{\Delta} \Big[\sin k \delta_{jk} + (\cos k \delta_{jk} - 1) \sin k (s - s_j) - \sin k \delta_{jk} \cdot \cos k (s - s_j)\Big]$$

$$Y_j(s) = \frac{1}{\Delta} \Big[-\sin k (\delta_{j\ell} + \delta_{jk}) + (\cos k \delta_{j\ell} - \cos k \delta_{jk}) \sin k (s - s_j) + (\sin k \delta_{j\ell} + \sin k \delta_{jk}) \cos k (s - s_j)\Big]$$

$$Z_j(s) = \frac{1}{\Delta} \Big[\sin k \delta_{j\ell} + (1 - \cos k \delta_{j\ell}) \sin k (s - s_j) - \sin k \delta_{j\ell} \cdot \cos k (s - s_j)\Big]$$

PREVIOUS PAGE IS BLANK

where the symbols are as defined for subroutine CABC. The electric or magnetic field incident on observation segment i computed from this current distribution is

$$\bar{E} = I_\ell \bar{E}_1 + I_j \bar{E}_2 + I_k \bar{E}_3$$

$$\bar{E}_1 = \frac{1}{\Delta} \left[\bar{E}_k \sin k\, \delta_{jk} + \bar{E}_s (\cos k\, \delta_{jk} - 1) - \bar{E}_c \sin k\, \delta_{jk} \right]$$

$$\bar{E}_2 = \frac{1}{\Delta} \left[-\bar{E}_k \sin k\, (\delta_{j\ell} + \delta_{jk}) + \bar{E}_s (\cos k\, \delta_{j\ell} - \cos k\, \delta_{jk}) + \bar{E}_c (\sin k\, \delta_{j\ell} + \sin k\, \delta_{jk}) \right]$$

$$\bar{E}_3 = \frac{1}{\Delta} \left[\bar{E}_k \sin k\, \delta_{j\ell} + \bar{E}_s (1 - \cos k\, \delta_{j\ell}) - \bar{E}_c \sin k\, \delta_{j\ell} \right]$$

where

$\bar{E}_k$ = field due to constant current on segment j

$\bar{E}_s$ = field due to current sin k $(s - s_j)$ on segment j

$\bar{E}_c$ = field due to current cos k $(s - s_j)$ on segment j

The term $\bar{E}_2 \cdot \hat{i}$ is the contribution to the matrix element G_{ij}. Here $\hat{i}$ is either the unit vector in the direction of segment i or the unit vector in the direction of $\hat{t}_1$ or $\hat{t}_2$ of the observation patch i. $\bar{E}_1 \cdot \hat{i}$ and $\bar{E}_3 \cdot \hat{i}$ contribute to elements in row i with column indices of the segments that connect end one and end two of segment j, respectively. Subroutine TNEFLD is called to compute the electric fields $\bar{E}_k$, $\bar{E}_s$, and $\bar{E}_c$ which are returned in vector components ρ' and z' defined in a cylindrical coordinate system with origin at the center of segment j and $\hat{z}' = \hat{j}$, where $\hat{j}$ is the unit vector in the direction of segment j. TNHFLD is called to compute the magnetic fields which are returned as the ϕ component of the cylindrical coordinate system defined with respect to the origin of segment j and $\hat{z}' = \hat{j}$.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CABJ	The unit vector in the x direction of the wire source
CK	$k\delta_{jk}$
CL	$k\delta_{j\ell}$
CONS	Δ = SINL + SINK - SILK
COSK	$\cos(k\delta_{jk})$
COSL	$\cos(k\delta_{j\ell})$
DIJ	$\hat{i} \cdot \hat{j}$
DIK	δ_{jk} - the interpolation length between segment j and adjacent segment k
DIL	$\delta_{j\ell}$ - the interpolation length between segment j and adjacent segment ℓ
DIR	$\hat{i} \cdot \hat{\rho}'$
ERIC	Imaginary part of $\hat{\rho}'$ component of $\bar{E}_c$
ERIK	Imaginary part of $\hat{\rho}'$ component of $\bar{E}_k$
ERIS	Imaginary part of $\hat{\rho}'$ component of $\bar{E}_s$
ERRC	Real part of $\hat{\rho}'$ component of $\bar{E}_c$
ERRK	Real part of $\hat{\rho}'$ component of $\bar{E}_k$
ERRS	Real part of $\hat{\rho}'$ component of $\bar{E}_s$
ETI	Array containing imaginary parts of $\bar{E}_1 \cdot \hat{i}$, $E_2 \cdot \hat{i}$, and $E_3 \cdot \hat{i}$
ETIC	Imaginary part of $\bar{E}_c \cdot \hat{i}$, $-\bar{H}_c \cdot \hat{t}_1$, or $-H_c \cdot \hat{t}_2$
ETIK	Imaginary part of $\bar{E}_k \cdot \hat{i}$, $-\bar{H}_k \cdot \hat{t}_1$, or $-H_k \cdot \hat{t}_2$

ETIS	Imaginary part of $\bar{E}_s \cdot \hat{i}$, $-\bar{H}_s \cdot \hat{t}_1$, or $-H_s \cdot \hat{t}_2$
ETR	Array containing real parts of $\bar{E}_1 \cdot \hat{i}$, $E_2 \cdot \hat{i}$, and $E_3 \cdot \hat{i}$
ETRC	Real part of $\bar{E}_c \cdot \hat{i}$, $-\bar{H}_c \cdot \hat{t}_1$, or $-H_c \cdot \hat{t}_2$
ETRK	Real part of $\bar{E}_k \cdot \hat{i}$, $-\bar{H}_k \cdot \hat{t}_1$, or $-H_k \cdot \hat{t}_2$
ETRS	Real part of $\bar{E}_s \cdot \hat{i}$, $-\bar{H}_s \cdot \hat{t}_1$, or $-H_s \cdot \hat{t}_2$
EZIC	Imaginary part of $\hat{z}'$ component of $\bar{E}_c$
EZIK	Imaginary part of $\hat{z}'$ component of $\bar{E}_k$
EZIS	Imaginary part of $\hat{z}'$ component of $\bar{E}_s$
EZRC	Real part of $\hat{z}'$ component of $\bar{E}_c$
EZRK	Real part of $\hat{z}'$ component of $\bar{E}_k$
EZRS	Real part of $\hat{z}'$ component of $\bar{E}_s$
FZI	Variable set to zero in calling GNDREF when using a cylindrical coordinate system
FZR	Variable set to zero in calling GNDREF when using a cylindrical coordinate system
HPIC,HPRC	The imaginary and real parts of the magnitude of the ϕ component of magnetic field due to a cosine current, H_c
HPIK,HPRK	The imaginary and real parts of the magnitude of the ϕ component of the magnetic field due to a constant current, H_k
HPIS,HPRS	The imaginary and real parts of the magnitude of the ϕ component of the magnetic field due to a sine current, H_s
HXIC,HXRC	The imaginary and real parts of $\bar{H}_c$ in the x direction
HXIK,HXRK	The imaginary and real parts of $\bar{H}_k$ in the x direction

HXIS,HXRS	The imaginary and real parts of $\bar{H}_s$ in the x direction
HYIC,HYRC	The imaginary and real parts of $\bar{H}_c$ in the y direction
HYIK,HYRK	The imaginary and real parts of $\bar{H}_k$ in the y direction
HYIS,HYRS	The imaginary and real parts of $\bar{H}_s$ in the y direction
HZIC,HZRC	The imaginary and real parts of $\bar{H}_c$ in the z direction
HZIK,HZRK	The imaginary and real parts of $\bar{H}_k$ in the z direction
HZIS,HZRS	The imaginary and real parts of $\bar{H}_s$ in the z direction
IJ	(i - j) flag for wires or odd/even flag for patches
IP	When IP = 2, indicates the field is reflected from ground plane.
IPATCH	A flag indicating a patch observation point
IWIRE	A flag indicating a wire source segment and that a local cylindrical coordinate system is used
PHX,PHY,PHZ	The components of $\hat{\phi}$, where $\hat{\phi} = (\hat{j} \times \hat{\rho})$
RH	ρ coordinate of segment i in $\hat{\rho}'$ and $\hat{z}'$ coordinates
RHOX,RHOY,RHOZ	The x, y, and z components of $\hat{\rho}$
SABJ	The unit vector in the y direction of the source segment
SALPR	The unit vector in the z direction of the source segment
SILK	$\sin\left[k\ (\delta_{j\ell} + \delta_{jk})\right]$
SINK	$\sin\ (k\ \delta_{jk})$

SINL	sin $(k\ \delta_{j\ell})$
T1XI,T1YI,T1ZI	$\hat{t}_1$ unit vector of observation patch
T2XI,T2YI,T2ZI	$\hat{t}_2$ unit vector of observation patch
WAVNUM	$2\pi/\lambda$
ZP	z coordinate of segment i in $\hat{\rho}'$ and $\hat{z}'$ coordinates

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
CABJ	/AMPZIJ/
DIJ	F.P.
DIK	/AMPZIJ/
DIL	/AMPZIJ/
DIR	F.P.
IJ	F.P.
IP	F.P.
IPATCH	F.P.
ISOFF	/ADEBUG/
ISON	/ADEBUG/
RH	F.P.
RHOX,RHOY,RHOZ	/AMPZIJ/
SABJ	/AMPZIJ/
SALPR	/AMPZIJ/
T1XI,T1YI,T1ZI	/AMPZIJ/
T2XI,T2YI,T2ZI	/AMPZIJ/
WAVNUM	/AMPZIJ/
ZP	F.P.

B. OUTPUT	LOCATION
ANUMK	/ANUM/
ANUML	/ANUM/
ETI	F.P.
ETR	F.P.

6. CALLING ROUTINE:

ZIJSET

7. CALLED ROUTINES:

ASSIGN

GNDREF

STATIN

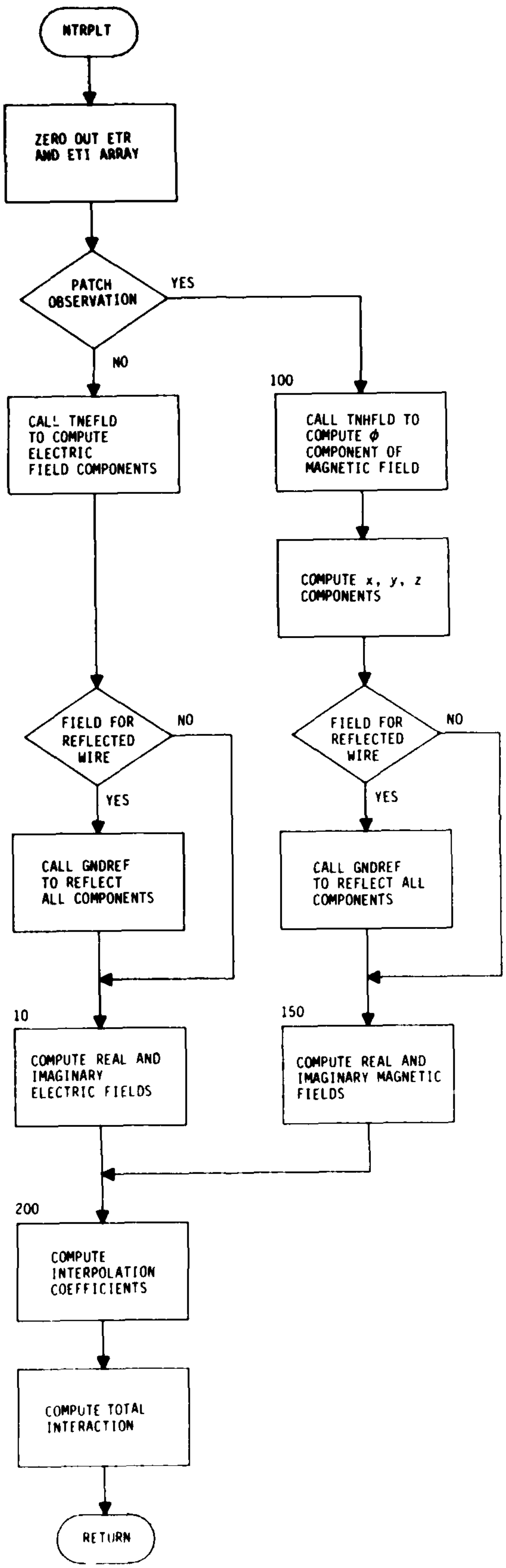
STATOT

TNEFLD

TNHFLD

WLKBCK

NTRPLT (MOM)


NTRPLT
ZERO OUT ETR
AND ETI ARRAY
PATCH
OBSERVATION
YES
NO
100
CALL TNEFLD
TO COMPUTE
ELECTRIC
FIELD COMPONENTS
CALL TNHFLD TO
COMPUTE ϕ
COMPONENT OF
MAGNETIC FIELD
COMPUTE x, y, z
COMPONENTS
FIELD FOR
REFLECTED
WIRE
NO
YES
FIELD FOR
REFLECTED
WIRE
NO
YES
CALL GNDREF
TO REFLECT
ALL COMPONENTS
CALL GNDREF
TO REFLECT ALL
COMPONENTS
10
150
COMPUTE REAL AND
IMAGINARY
ELECTRIC FIELDS
COMPUTE REAL AND
IMAGINARY MAGNETIC
FIELDS
200
COMPUTE
INTERPOLATION
COEFFICIENTS
COMPUTE TOTAL
INTERACTION
RETURN

1. NAME: NTRPLU (MOM)

2. PURPOSE: To compute the complex field components of the interaction matrix due to unit currents on a patch.

3. METHOD: This subroutine is called by the subroutine ZIJSET to compute the tangential electric field along the observation segment and to compute the magnetic field at the observation patch.

The electric field due to a surface patch is computed by calling the routine UNEFLD, and the component of the field tangent to the wire is computed as a dot product with the unit vector in the direction of the observation segment. This result is stored in the arrays ETR and ETI. For the case where the observation segment is connected to the surface (indicated by ICON not equal to zero), the electric field due to the four subpatches around the base of the segment is calculated by the subroutine WYRPAT.

For source patch j and observation patch i, NTRPLU calls UNHFLD to obtain the H-fields at the center of patch i due to unit currents in the directions $\hat{t}_{1j}$ and $\hat{t}_{2j}$ on patch j. The matrix elements are computed as components of the H-field in the directions $\hat{t}_{1i}$ and $\hat{t}_{2i}$ and they are stored in the arrays ETR and ETI. These matrix elements represent the third terms in equation 57 and the negative of equation 58 in the Engineering Manual. When the source and observation patches coincide, contributions of $\pm 1/2$ from the second term in equation 57 and the negative of equation 58 of the Engineering Manual are added to the interaction matrix.

When a ground is present the field of the image of the patch is computed, multiplied by the reflection coefficients, and added to the direct field.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CABI	X component of the observation segment unit vector
ETI, ETR	An array containing the imaginary and real parts of the interaction terms
EWPI, EWPR	An array containing the imaginary and real parts of the interaction terms for a wire connected to a patch
EXIT1, EYIT1, EZIT1 EXRT1, EYRT1, EZRT1	The x,y, and z components of the imaginary and real part of the electric or magnetic

	field at the observation segment due to a current in the $\hat{t}_1$ direction on the source patch
EXIT2, EYIT2, EZIT2 EXRT2, EYRT2, EZRT2	The x,y, and z components of the imaginary and real part of the electric or magnetic field at the observation segment due to a current in $\hat{t}_2$ direction on the source patch
ICON	A flag indicating that a wire observation segment is connected to the patch source
IPATCH	A flag indicating that the observation point is a patch
IWIRE	A flag indicating a wire source segment and that a local cylindrical coordinate system is used
KP	A flag which when equal to two indicates reflection from the ground plane
SABI	Y component of the observation segment unit vector
SALPI	Z component of the observation segment unit vector
T1XI, T1YI, T1ZI	X,Y, and Z components of $\hat{t}_1$ for the observation patch
T2XI, T2YI, T2ZI	X,Y, and Z components of $\hat{t}_2$ for the observation patch
XIJ, YIJ, ZIJ	X,Y, and Z components of the vector from the source patch to the observation segment

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
CABI	/AMPZIJ/
EXIT1, EYIT1, EZIT1	/AMPZIJ/
EXIT2, EYIT2, EZIT2	/AMPZIJ/
EXRT1, EYRT1, EZRT1	/AMPZIJ/
EXRT2, EYRT2, EZRT2	/AMPZIJ/

ICON	F.P.
IPATCH	F.P.
ISOFF	/ADEBUG/
ISON	/ADEBUG/
KP	F.P.
SABI	/AMPZIJ/
SALPI	/AMPZIJ/
T1XI, T1YI, T1ZI	/AMPZIJ/
T2XI, T2YI, T2ZI	/AMPZIJ/
XIJ, YIJ, ZIJ	F.P.
B. OUTPUT	LOCATION
ETI, ETR	F.P.
EWPI, EWPR	F.P.

6. *CALLING ROUTINE:*

ZIJSET

7. CALLED ROUTINES:

ASSIGN

GNDREF

STATIN

STATOT

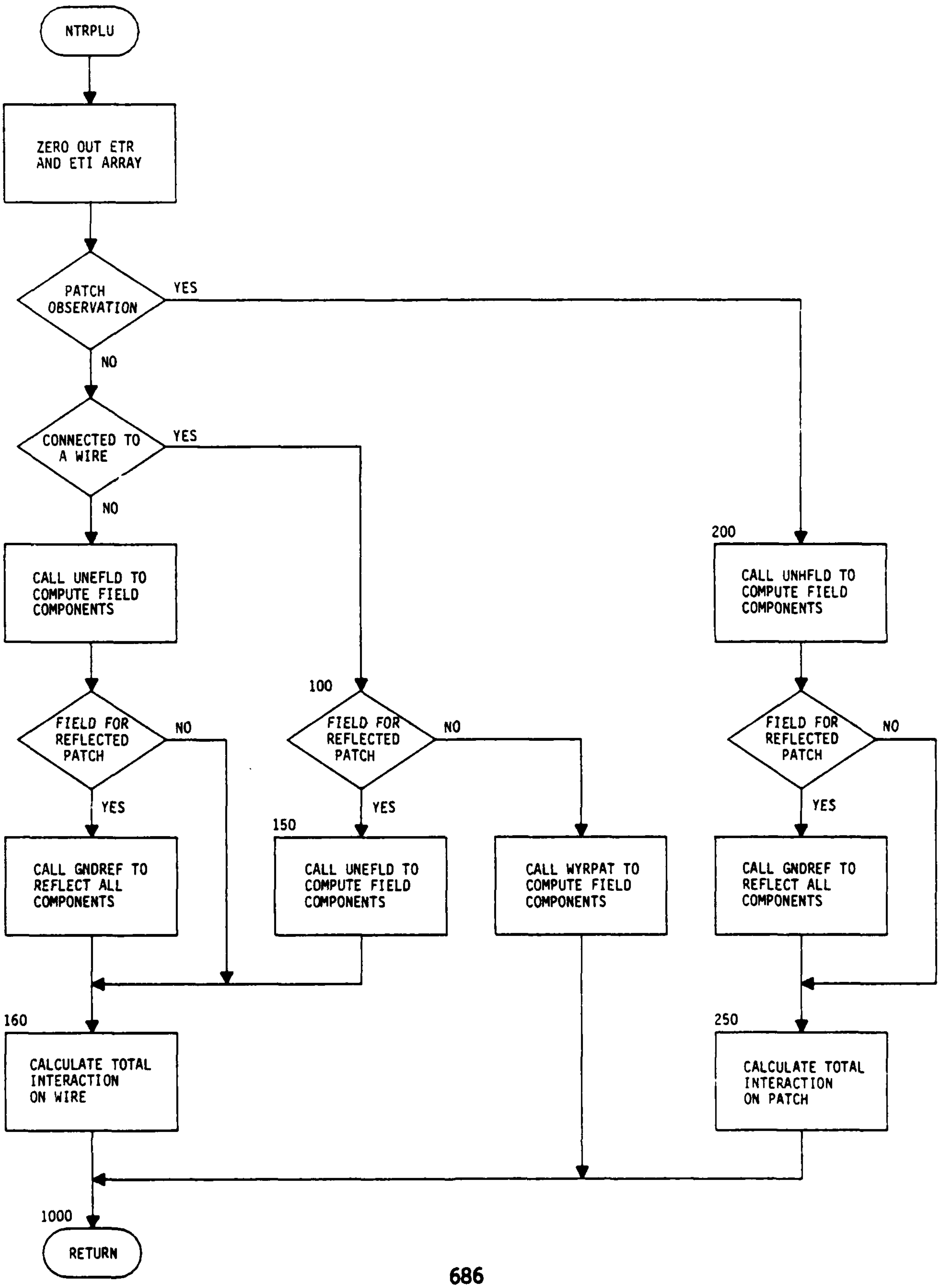
UNEFLD

UNHFLD

WLKBCK

WYRPAT

NTRPLU (MOM)


NTRPLU
ZERO OUT ETR AND ETI ARRAY
PATCH OBSERVATION
YES
NO
CONNECTED TO A WIRE
YES
NO
CALL UNEFLD TO COMPUTE FIELD COMPONENTS
200
CALL UNHFLD TO COMPUTE FIELD COMPONENTS
100
FIELD FOR REFLECTED PATCH
NO
FIELD FOR REFLECTED PATCH
NO
FIELD FOR REFLECTED PATCH
NO
YES
YES
YES
150
CALL GNDREF TO REFLECT ALL COMPONENTS
CALL UNEFLD TO COMPUTE FIELD COMPONENTS
CALL WYRPAT TO COMPUTE FIELD COMPONENTS
CALL GNDREF TO REFLECT ALL COMPONENTS
160
CALCULATE TOTAL INTERACTION ON WIRE
250
CALCULATE TOTAL INTERACTION ON PATCH
1000
RETURN

1. NAME: OPNFIL (GTD, INPUT, MOM, OUTPUT)

2. PURPOSE: Subroutine to open peripheral files.

3. METHOD: Subroutine opens peripheral files by setting internal variables. If the file is already open, a fatal error condition is initiated. If the file is not open, the file is rewound and the file pointer flags are initialized to zero.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ISTAT	File pointer
LUFILE	Logical unit file designator
NUMTSK	File number to be opened

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
IOFILE	/IOFLES/
ISON	/ADEBUG/
LUPRNT	/ADEBUG/
NUMTSK	F.P.

B. OUTPUT

OUTPUT	LOCATION
IERRF	/ADEBUG/
IOFILE	/IOFLES/

OPNFIL (GTD, INPUT, MOM, OUTPUT)

6. CALLING ROUTINES*:

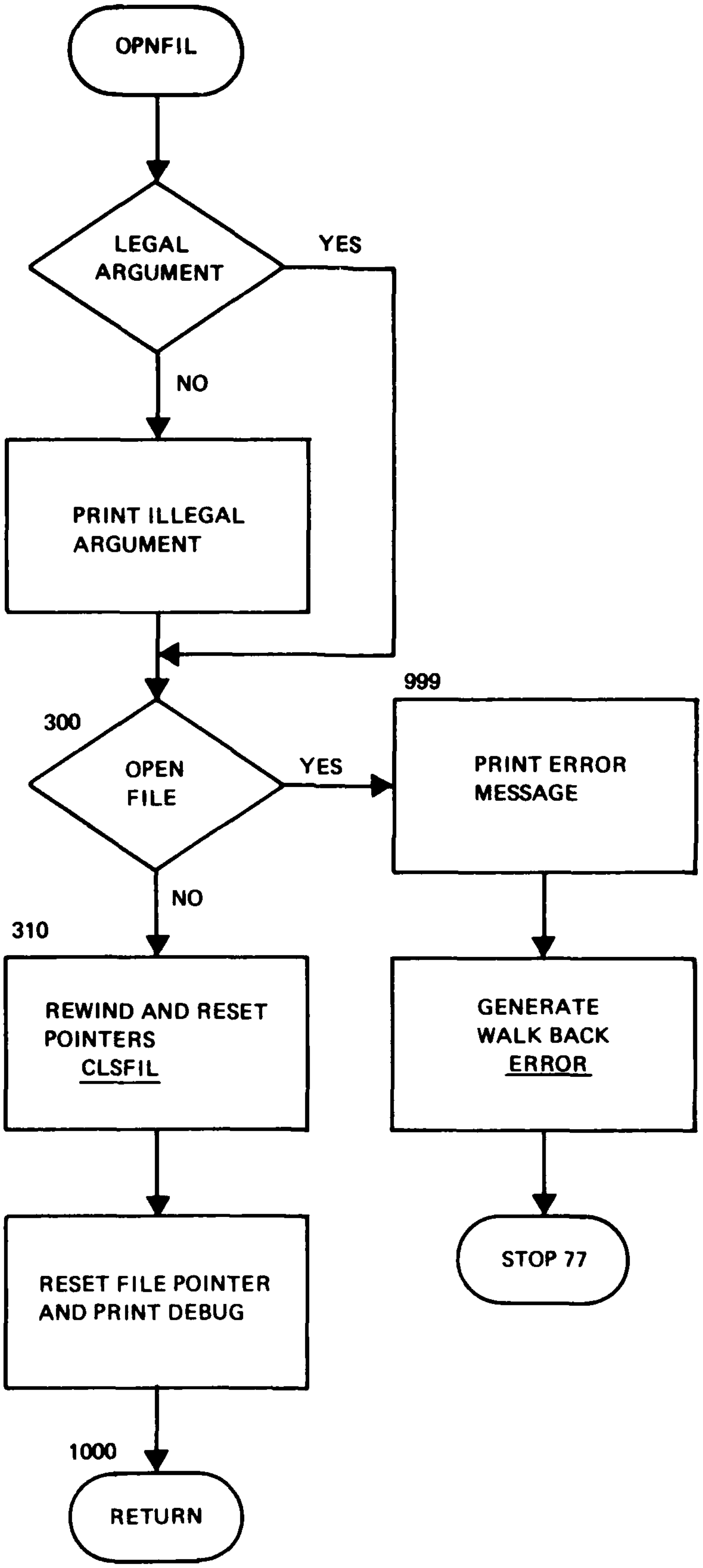
BUBBLE (1)	SOLDRV (3)
CABC (3)	STATFN (1,2,3,4)
DECOMP (3)	SUBPAT (1)
GEODRV (1)	SYMDEF (1,2,3,4)
PRTSYM (3)	TSKXQT (3)
PUTSYM (1,2,3,4)	WRTCHK (1,2,3,4)
RWFILS (1,2,3,4)	

7. CALLED ROUTINES:

ASSIGN	ERROR	STATOT
CLSFIL	STATIN	WLKBCK

*1-INPUT
2-GTD
3-MOM
4-OUTPUT

OPNFIL (GTD, INPUT, MOM, OUTPUT)



1. NAME: PAGPLT (MOM, OUTPUT)

2. PURPOSE: Plot a column from the electric field plot matrix.

3. METHOD: Plot all x, y points in a plot frame specified by ITYPE.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DCHR	Character width in current scale on the x axis
DCINV	Interval of each character on the x axis
DLINV	Interval of each character on the y axis
DLYN	Character height in current scale on the y axis
DX	Distance between scaling marks on the x axis
DY	Distance between the scale marks on the y axis
IS	Variable for checking the sign
ISBLNK	A blank
ISDASH	A dash
ISDOT	A dot
ISPLUS	A plus
ISSTAR	A star
ISYM	An array for building a plot line
ITYPE	Plot type LINLIN, LINLOG, LOGLIN, and LOGLOG
IY	Flag for printing a scale value
LASTI	Check for the last line
LINE	Line counter

LPRPGE	Maximum number of lines to be printed per page
N	Input number of points
NCHLIN	Number of characters on a line
NEXTI	Next scanning point from which to look for points to print on the current line
NI	Number of scale marks plus one
NMOD	The interval size on the y axis
NP	Number of points to be plotted
NXINT	Interval size on the x axis
NXVAL	Number of scale marks on the x axis
NYINT	Interval size on the y axis
NYV	Value of the scale to be printed on the y axis
NYVAL	Number of scale marks on the y axis
X	Temporary location for sorting the x array
XMAX	Maximum value in the x array
XMAXP	The rescaled maximum value
XMIN	Minimum value in the x array
XMINP	The rescaled minimum value
XP	Input x array
XVAL	Array containing the scale values for the x axis
Y	Temporary location for sorting the y array
YMAX	Maximum value in the y array
XMAXP	Maximum rescaled y value
YMIN	Minimum value in the y array

YMINP	Minimum rescaled y value
YP	Input y array
YVAL	Array containing the scale values for the y axis

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
ITYPE	F.P.
LUPRNT	/ADEBUG/
N	F.P.
XP	F.P.
YP	F.P.

B. OUTPUT:

NONE

6. CALLING ROUTINE:

FLDOUT

7. CALLED ROUTINES:

ASSIGN

SCALE2

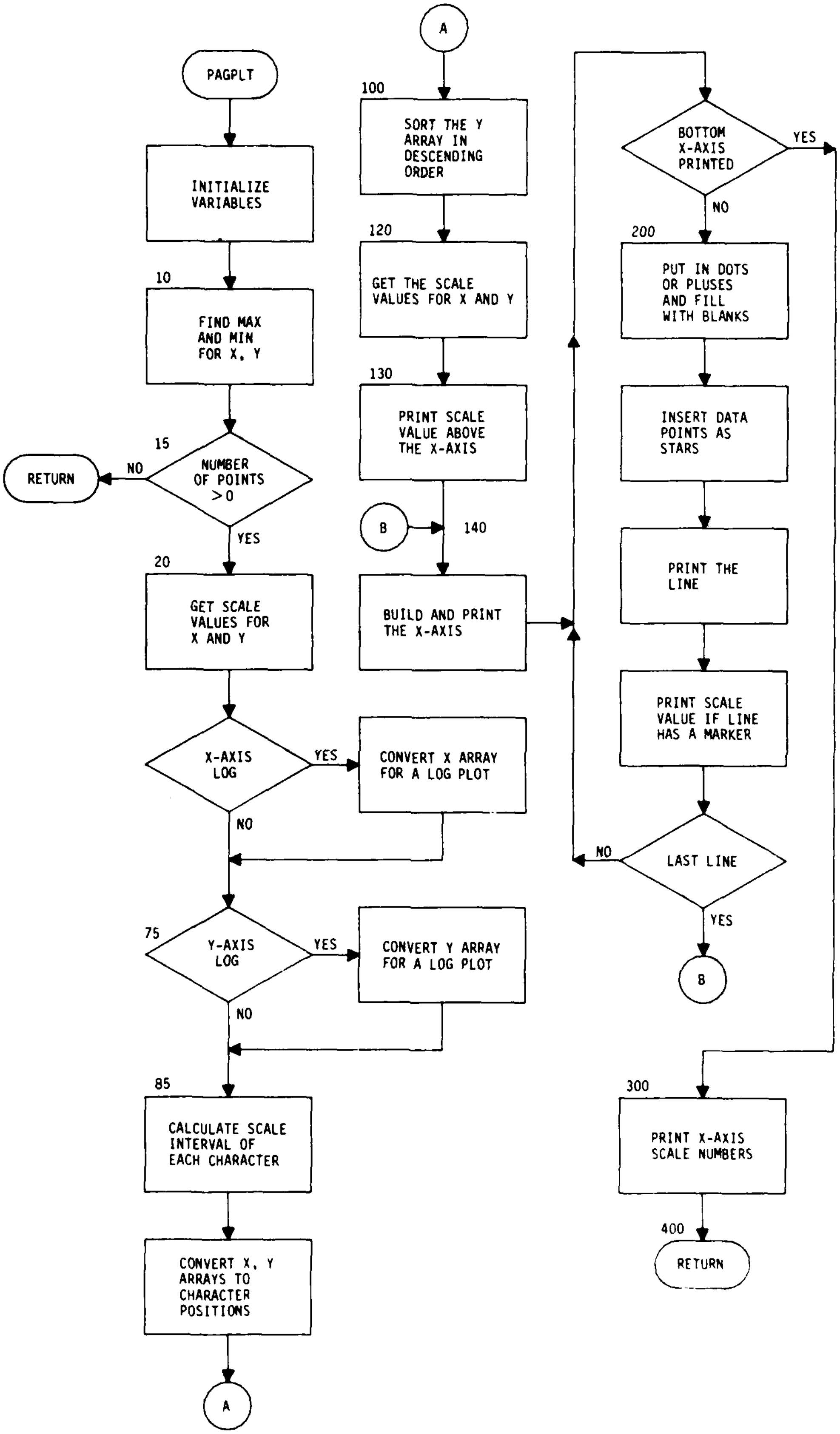
SCALE3

STATIN

STATOT

WLKBCK

PAGPLT (MOM, OUTPUT)



1. NAME: PARSE (INPUT)

2. PURPOSE: Subroutine to parse the command text and generate an argument list to be used for execution.

3. METHOD: There are two types of input that PARSE can handle, tasks and expressions. A task contains a keyword that points to a task in the task format table. This table is used to tell PARSE what arguments need to be put in the argument list. The expression is put directly into the argument list. If, after parsing, there is any leftover information, a fatal error is generated.

A. TASKS

PARSE scans the NCODE array for the first keyword that has a nonzero entry in the KWFMTP column of the keyword table. The following example illustrates how it works:

B = BAND(A), BNDW = 10

The keyword BAND is the second entry in the following table.

TABLE 1. KEYWORD TABLE

NKEYW	KWNAME	KWFMTP	KWARG
1	107	1	0
2	73	6	-2
3	213	0	-3
4	109	15	0
5	110	0	0

The KWFMTP column has a pointer to the following table.

TABLE 2. TASK FORMAT TABLE

NTFMTP	NTSFMT	DESCRIPTION OF TABLE ENTRY
1	4	Number of entries in this table for BACSUB.
2	2	Task number for BACSUB.
3	-2	Defined symbol.
4	-1	Symbol.
5	-2	Defined symbol.
6	4	Number of entries in this table for BANDIT.
7	3	Task number for BANDIT.
8	-1	Symbol.
9	-2	Defined symbol.
10	41	Keyword number for BNDW.
11	0	
12	0	

Now PARSE knows what to look for in the scan tables to build an argument list. The argument list is built by stepping through the NTSFMT table and doing the following:

(1) Put the task number in the argument list (NARGTB) and the argument list pointer (NPARGL) in the task table (NTSKTB). Diagram 1 should help illustrate this.

NPTASK	NTSKTB		NPARGL	NARGTB
4	11	⟶	11	3
5	0		12	-999999

Diagram 1. Task Table and Argument List

(2) The negative entries found after the task number in the NTSFMT table are argument types which will be sent to FNDARG. This routine will scan the scan table for the first argument that has not been used and will put the needed information in the current location of the argument list. If the argument is not the right type, an error will be generated. In the example command,

B = BAND(A), BNDW = 10

if A is defined as the first location in the symbol table prior to this command, B would be put in the second location of the symbol table (NDATBL). The indices from the symbol table would be put in the argument list as shown below:

NPTASK	NTSKTB		NPARGL	NARGTB
4	11	⟶	11	3
5	0		12	2←B
			13	1←A
			14	-99999

Diagram 2. Task Table and Argument List

(3) The positive entries found after the task number in the NTSFMT table are keywords. If the keyword is less than 256, PARSE scans for this keyword. If it is not found, a NOPCOD is left, or else the KWARG column in the keyword table is referenced for the argument type needed for this keyword. This argument type is sent to FNDARG and will yield the following for the sample command:

NPTASK	NTSKTB		NPARGL	NARGTB
4	11	⟶	11	3
5	0		12	2
			13	1
			14	10←BNDW
			15	-99999

Diagram 3. Task Table and Argument List

If the keyword is greater than 256, it is a multiple choice keyword. This means that there may be two or three keywords packed in the first 24 bits of a word using radix 256. These keywords are unpacked into an array. Then PARSE searches for each keyword until one is found. If not, a NOPCOD is left. The following example will show this:

DEBUG ON

NPTASK	NTSKTB	NPARGL	NARGTB
4	11	13	1
5	15 →	14	10
		→ 15	8 DEBUG TASK NUMBER
		16	64←ON
		17	-99999

Diagram 4. Task Table and Argument List

The keyword number will be put in the argument list.

(4) A list of arguments must be the last set of arguments in a command text because anything other than the end of the statement will be an error. The following illustrates a list of arguments:

DEBUG ON, BACSUB, LUD

NPTASK	NTSKTB	NPARGL	NARGTB
4	11	14	10
5	15 →	→ 15	8
6	0	16	64
		17	2
		18	9
		19	-9999

Diagram 5. Task Table and Argument List

B. DIRECT MANIPULATION

In the direct manipulation process, all symbols are put on the symbol table and the rest of the arguments are put on the literal table. The following example will illustrate this:

FRQ = 300.

NPTASK	NPSKTB	NPARGL	NARGTB
4	11	18	9
5	15	19	44
6	19	20	-1
		21	-2
		22	-3

Diagram 6. Task Table and Argument List

All the arguments were put in the literal table.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
INDEX	Index to literal or symbol table
IPSARG	Internal variable for NPARGL
IPSAR1	IPSARG + 1
KWARGT	Passes argument type to FNDARG
KWDNM	Keyword number
KWNUM	Index to a keyword in multiple keyword array (KWDM)
MLTKWD	First temporary location to unpack a multiple keyword
NCDNTB	NCODE (NTAB)
NKEYW	Index for keyword table
NM	Index to unpack multiple keywords

NMKWDS	Index to a keyword in the KWDM array
NMLTKW	Second temporary location to unpack a multiple keyword
NRSTRT	RESTRT task number
NSTEP	Index to step through the task format table
NTFMTP	Task pointer to the task format table
NTFP	Used to point at task number and the first argument type
NTFTLM	Points to the last argument type in the task format table

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
ISOFF	/ADEBUG/
ISON	/ADEBUG/
KWARG	/PARTAB/
KWDM	/PARTAB/
KWFMTP	/PARTAB/
KWLMT	/PARTAB/
MKMX	/PARTAB/
NARGMX	/PARTAB/
NCODE	/SCNPAR/
NDTASK	/SCNPAR/
NOPCOD	/ADEBUG/
NPARGL	/PARTAB/
NPDATA	/PARTAB/
NPEARG	/INPERR/

NPEIFO	/INPERR/
NDESCN	/INPERR/
NPETSK	/INPERR/
NPLITN	/PARTAB/
NPLOOP	/PARTAB/
NPTASK	/PARTAB/
NTALPH	/ADEBUG/
NTASK	/SCNPAR/
NTDM	/PARTAB/
NTEND	/ADEBUG/
NTFLPT	/ADEBUG/
NTINT	/ADEBUG/
NTKEYW	/ADEBUG/
NTSFMT	/PARTAB/
NTSKMX	/PARTAB/
NTSYMB	/ADEBUG/
NVAL	/SCNPAR/

B. OUTPUT	LOCATION
IPSDAT	/SCNPAR/
IPSLIT	/SCNPAR/
IPSLOO	/SCNPAR/
IPSTSK	/SCNPAR/
MATCH	/SCNPAR/
NARGTB	/PARTAB/
NCODE	/SCNPAR/

NFINCD	/SCNPAR/
NOMTCH	/SCNPAR/
NPARGL	/PARTAB/
NPDATA	/PARTAB/
NPLITN	/PARTAB/
NPLOOP	/PARTAB/
NPRSER	/SCNPAR/
NPTASK	/PARTAB/
NRESTF	/SCNPAR/
NTAB	/SCNPAR/
NTSKTB	/PARTAB/
NUMWRD	/ADEBUG/

6. CALLING ROUTINE:

   INPDRV

7. CALLED ROUTINES:

   ASSIGN

   FABLO2

   FNDARG

   INC (intrinsic)

   LITSCH

   POSTPR

   PREPAR

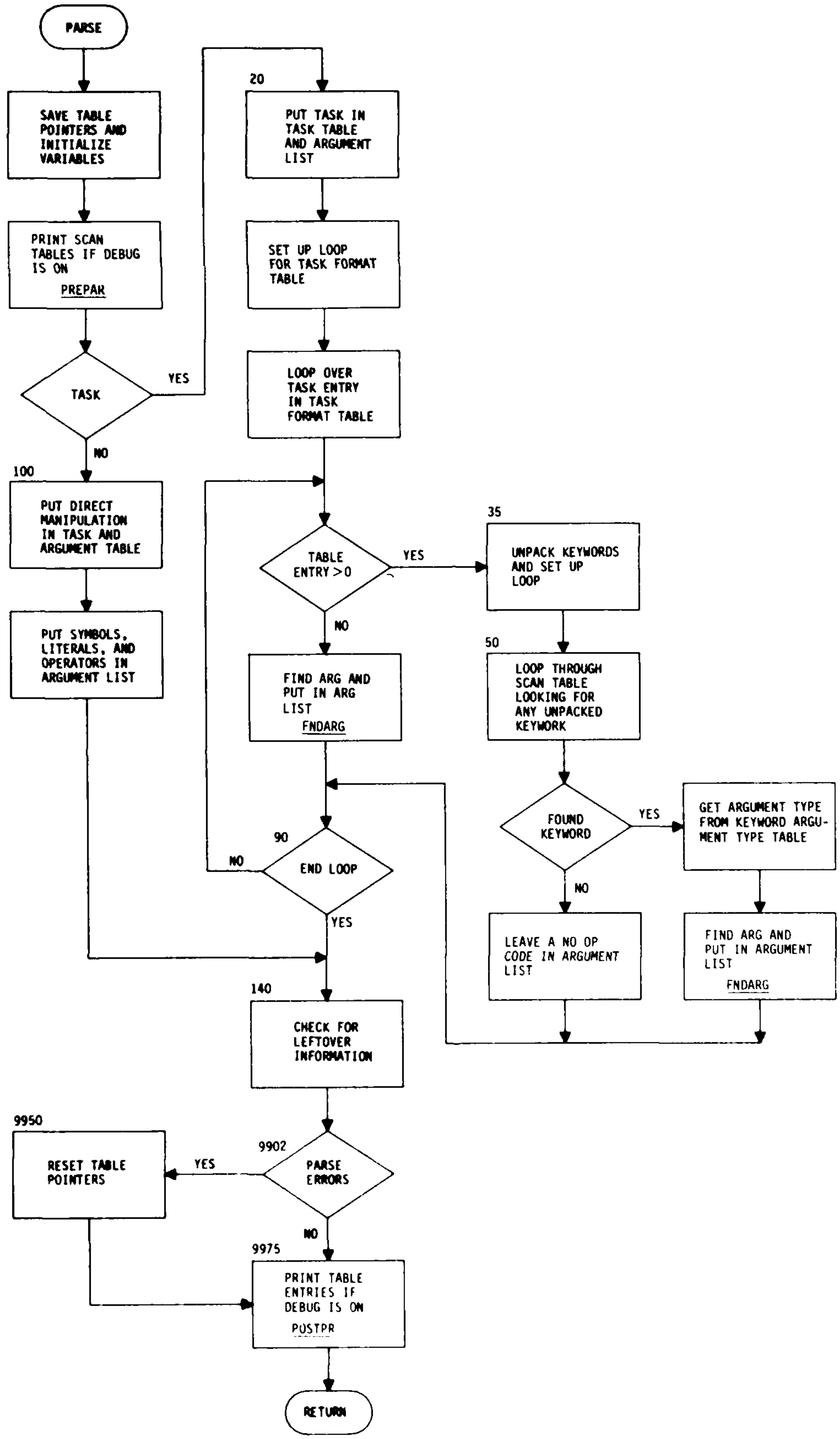
   STATIN

   STATOT

   SYMLIT

   WLKBCK

PARSE (INPUT)


PARSE
SAVE TABLE POINTERS AND INITIALIZE VARIABLES
PRINT SCAN TABLES IF DEBUG IS ON
PREPAR
TASK
YES
NO
100
PUT DIRECT MANIPULATION IN TASK AND ARGUMENT TABLE
PUT SYMBOLS, LITERALS, AND OPERATORS IN ARGUMENT LIST
20
PUT TASK IN TASK TABLE AND ARGUMENT LIST
SET UP LOOP FOR TASK FORMAT TABLE
LOOP OVER TASK ENTRY IN TASK FORMAT TABLE
TABLE ENTRY > 0
YES
NO
35
UNPACK KEYWORDS AND SET UP LOOP
50
LOOP THROUGH SCAN TABLE LOOKING FOR ANY UNPACKED KEYWORK
FOUND KEYWORD
YES
NO
GET ARGUMENT TYPE FROM KEYWORD ARGUMENT TYPE TABLE
LEAVE A NO OP CODE IN ARGUMENT LIST
FIND ARG AND PUT IN ARGUMENT LIST
FNDARG
FIND ARG AND PUT IN ARG LIST
FNDARG
90
END LOOP
NO
YES
140
CHECK FOR LEFTOVER INFORMATION
9902
PARSE ERRORS
YES
NO
9950
RESET TABLE POINTERS
9975
PRINT TABLE ENTRIES IF DEBUG IS ON
POSTPR
RETURN

1. NAME: PATCH (INPUT)

2. PURPOSE: PATCH stores the geometry data of a surface patch in the segment table.

3. METHOD: This subroutine distinguishes whether the user has specified that a patch is to be placed at a certain point or all the patch geometry data have been specified. If a point location is given, the appropriate data point is retrieved and the patch is located at that position. Then all the patch geometry data are stored in the segment table. In order to distinguish patch data from wire data, the patch area is stored as a negative number. The location of the patch data in the segment table is:

   SEGTBL (I,1) - segment table location and patch tag number

   SEGTBL (I,2) - x coordinate of patch center

   SEGTBL (I,3) - y coordinate of patch center

   SEGTBL (I,4) - z coordinate of patch center

   SEGTBL (I,5) - x component of patch normal vector

   SEGTBL (I,6) - y component of patch normal vector

   SEGTBL (I,7) - z component of patch normal vector

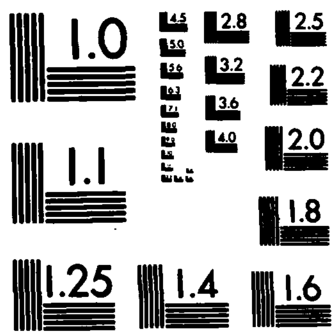
   SEGTBL (I,8) - patch area

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AREA	Patch surface area
ATTACH	Set .FALSE. to preclude translation of unit vector
CP	Cos (PHI)
CT	Cos (THETA)
DGTORD	Degrees to radians conversion factor
ICS	Index to a coordinate system in which a given operation is desired

AD-A137 464 GENERAL ELECTROMAGNETIC OODEL FOR THE ANALYSIS OF COMPLEX SYSTEMS (GEMACS..(U) BDM CORP ALBUQUERQU NM D L KADLEC ET AL. SEP 83 BDM/A-83-020-TR-VOL-3-PT-2 4/4

UNCLASSIFIED RADC-TR-83-217-VOL-3-PT-2 F30602-81-C-0084 F/G 20/14 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ICSSAV	Saves the ICS index
IPT	Point number where the patch is to be attached
IS	Input segment number
ITAG	Tag number
NOGOFG	No go flag
NORM	Set .TRUE. when patch normal has been calculated
NTALPH	An identifier of an alpha field or character
NTINT	An identifier of an integer field
NU	Location in the NVAL array
PHI	ϕ in spherical coordinate system
SCALE	A factor to change the scale of all dimensions. It is used here to preserve values of unit normal vectors when placed in the SEGTBL array.
SP	Sin (PHI)
ST	Sin (THETA)
THETA	θ in spherical coordinate system
XCPA, YCPA, ZCPA	X,Y, and Z coordinates of the patch center
XNPA, YNPA, ZNPA	X,Y, and Z coordinates of the patch unit normal vector
XP, YP, ZP	Coordinates of a specified point
X12, Y12, Z12	Distance between points 1 and 2
X13, Y13, Z13	Distance between points 1 and 3
X2, Y2, Z2	Dummy variables in call to GETPNT

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DGTORD	/GEODAT/
ICSSAV	F.P.
ISOFF	/ADEBUG/
ISON	/ADEBUG/
LUPRNT	/ADEBUG/
NCODE	/SCNPAR/
NTALPH	/ADEBUG/
NTINT	/ADEBUG/
NVAL	/SCNPAR/
SCALE	/SEGMNT/
VAL	/SCNPAR/

B. OUTPUT

OUTPUT	LOCATION
NOGOFG	/ADEBUG/
SEGTBL	/SEGMNT/

6. CALLING ROUTINE:

WYRDRV

7. CALLED ROUTINES:

ASSIGN

COORDS

GETPNT

PUTSEG

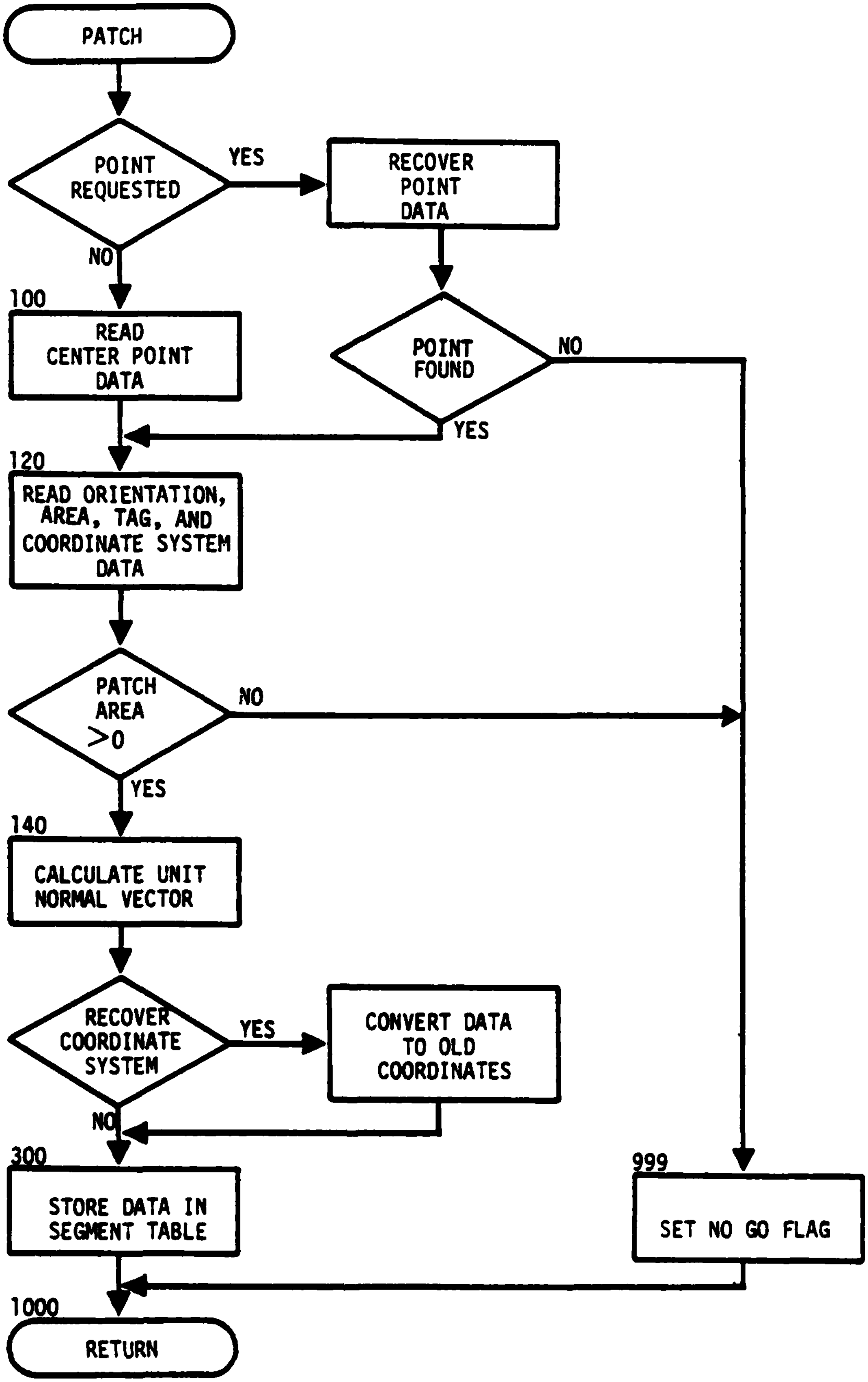
ROTATE

STATIN

STATOT

WLKBCK

PATCH (INPUT)


PATCH
POINT REQUESTED
YES
RECOVER POINT DATA
NO
100
READ CENTER POINT DATA
POINT FOUND
NO
YES
120
READ ORIENTATION, AREA, TAG, AND COORDINATE SYSTEM DATA
PATCH AREA >0
NO
YES
140
CALCULATE UNIT NORMAL VECTOR
RECOVER COORDINATE SYSTEM
YES
CONVERT DATA TO OLD COORDINATES
NO
300
STORE DATA IN SEGMENT TABLE
999
SET NO GO FLAG
1000
RETURN

1. NAME: PFUN (GTD)

2. PURPOSE: This function computes the p* function for the cylinder's acoustically soft diffraction coefficient.

3. METHOD: The p* function is defined as (see references A and B):

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} + \hat{P}_s(x)\ e^{j\pi/4}$$

where

$$\hat{P}_s(x) = \frac{e^{-j\pi/4}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{V(\tau)}{w_2(\tau)}\ e^{-jx\tau}\ d\tau\ ,$$

and $V(\tau)$ and $w_2(\tau)$ are Fock type Airy functions. The p* function is computed as follows:

1) for $x \leq -3$

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} + \frac{1}{2}\sqrt{|x|}\left(1 + j\frac{2}{x^3}\right) e^{j\frac{x^3}{12}}\ e^{j\pi/4}$$

2) for $-3 < x < 2$

$$p^*(x) = p^*(x_i) + \frac{(x-x_i)}{(x_{i+1}-x_i)}\,(p^*(x_{i+1})-p^*(x_i)),$$

where the $p^*(x_i)$ are tabulated values (see references A and B) and $x_{i+1}-x_i=0.1$ with $x_i \leq x \leq x_{i+1}$.

3) For $x \geq 2$

$$p^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{e^{j\pi/6}}{2\sqrt{\pi}} \sum_{n=1}^{5} \frac{e^{xq_n e^{-j\frac{5\pi}{6}}}}{[A'_i(-q_n)]^2}$$

where $A'_i(\tau)$ is the derivative of the Miller type Airy function.

4. INTERNAL VARIABLES:

VARIABLES	DEFINITION
AMC	-0.5*CEXP(J*PI/6)/SQRT(PI)
AQ	Derivative of Miller type Airy function at Q
C	0.5/SQRT(PI)
EXC	CEXP(-5*PI/6)
I	Smallest integer closest to 10*X
PFUN	P function
PI	π
PJ	Imaginary part of tabulated P function
PR	Real part of tabulated P function
Q	Zeroes of Miller type Airy function
X	Argument of P function
XI	Real number representation of I

5. I/O VARIABLES:

A. INPUT	LOCATION
PI	/PIS/
X	F.P.

B. OUTPUT	LOCATION
PFUN	FUNCTION

6. CALLING ROUTINES:

RPLSCL

SCLRPL

SCTCYL

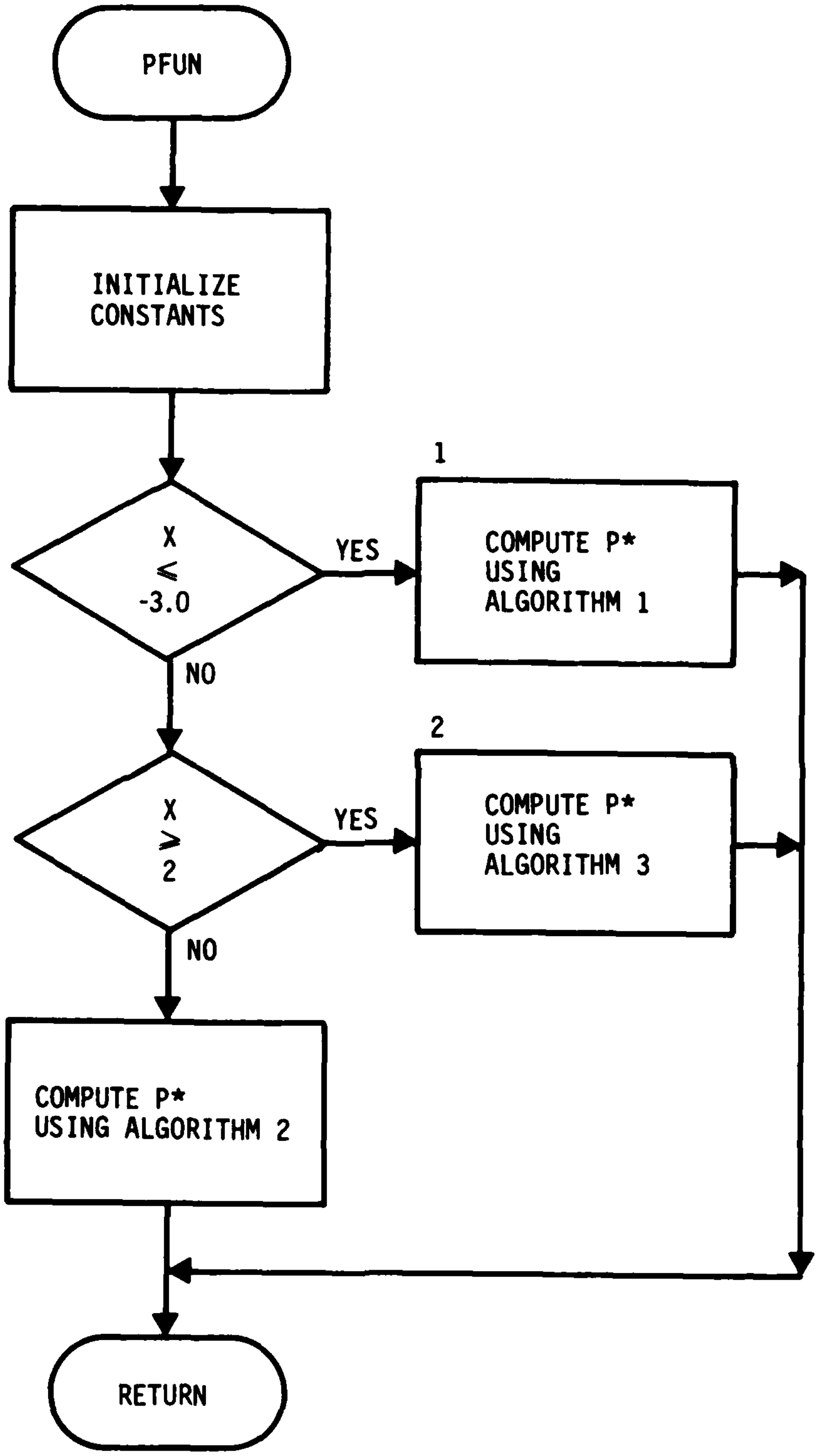
7. CALLED ROUTINE:

BEXP

8. REFERENCES:

A. P. H. Pathak, "An Asymptotic Analysis of the Scattering of Plane Waves by a Smooth Convex Cylinder," paper to appear in J. Radio Science. (Also The Ohio State University Electro-Science Laboratory Technical Report 784583-3, March 1978).

B. N. A. Logan, "General Research in Diffraction Theory," Vol. I, LMSD-288087; and Vol. II, LMSD-288088, Missiles and Space Division, Lockheed Aircraft Corp., 1959.


PFUN
INITIALIZE CONSTANTS
X ≤ -3.0
YES
NO
1
COMPUTE P* USING ALGORITHM 1
X ≥ 2
YES
NO
2
COMPUTE P* USING ALGORITHM 3
COMPUTE P* USING ALGORITHM 2
RETURN

1. NAME: PLAINT (GTD)

2. PURPOSE: To determine if a ray traveling from a given source location in a given direction will intersect a given plate (or set of plates).

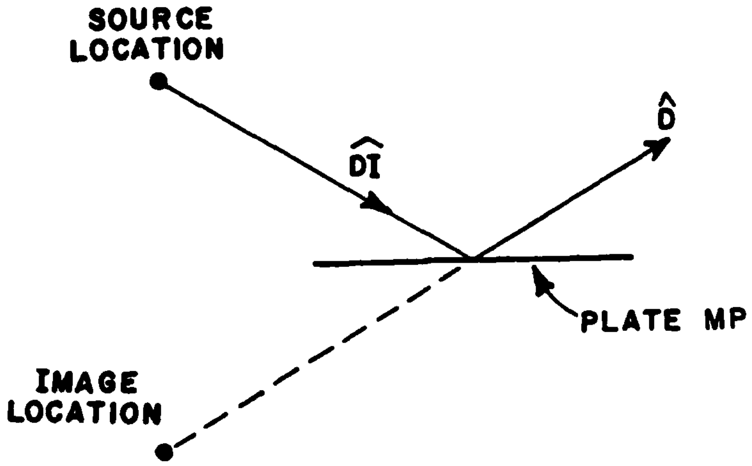
3. METHOD: This subroutine is used for a number of functions:

(a) To determine if a source ray reflection from plate MP occurs. If a ray traveling from the source image location in the reflected ray direction passes through plate MP, the reflection will occur (see figure 1). The routine only checks plate MP (set MH=-MP). Note that the hit point (which is returned as a formal parameter) is the reflection point, and is used in the shadowing tests.

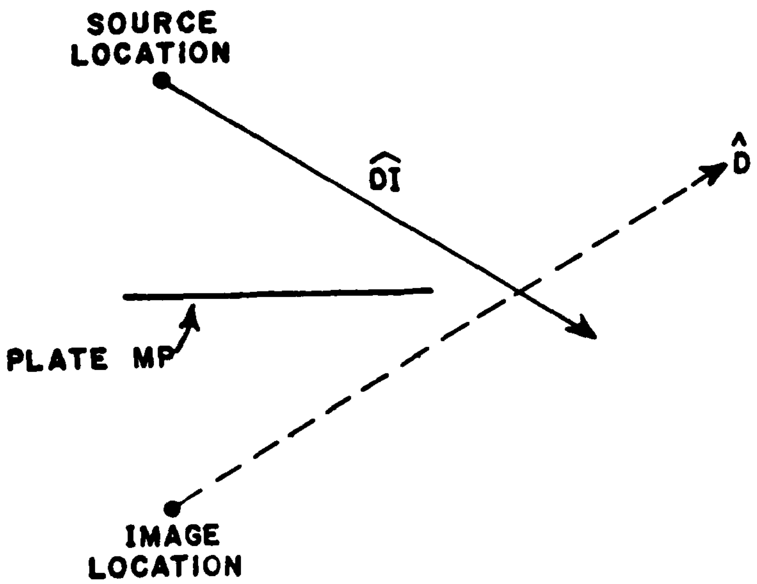
(b) To test to see if a ray is shadowed between scatter points (or between the source and a scatter point). The routine checks all plates (set MH=0) and records the distance from the first scatter (or source) position to the nearest hit (if the ray hits any of the plates). If the distance to the nearest hit is shorter than the distance between scatter points (or between the source and scatter point), the ray is shadowed, and the GTD term being computed is set to zero. Otherwise, the ray is not disturbed and computations are carried out. Note that if the first scatter point is a reflection or diffraction point on a plate, all plates except that plate are checked (set MH=MP).

(c) To determine if a ray after the final scatter point (or source ray) is shadowed. If the final scatter point is a cylinder (or if the source field is being computed) all plates are checked. If the final scatter point is on the plate MP, all plates except plate MP are checked. For far-field calculations if the ray hits a plate (LHIT=TRUE), the ray is shadowed and the GTD term is set to zero. For near-field calculations if a plate is hit (LHIT=TRUE), the routine calling PLAINT compares the distance between the final scatter point and the observation point to the distance between the scatter point and the hit point, for the ray may reach the observation point first and therefore not be shadowed. If LHIT=FALSE, the ray is not shadowed and propagates undisturbed (see figure 2).

(d) To determine if any one plate totally shadows plate MP from the source (referred to as the "total shadowing algorithm"). The routine checks all plates except plate MP (set MH = MP) and remembers plates which shadow the ray every time the routine is called (see section f of subroutine GEOM). The total shadowing algorithm is activated when LSTS is set TRUE.

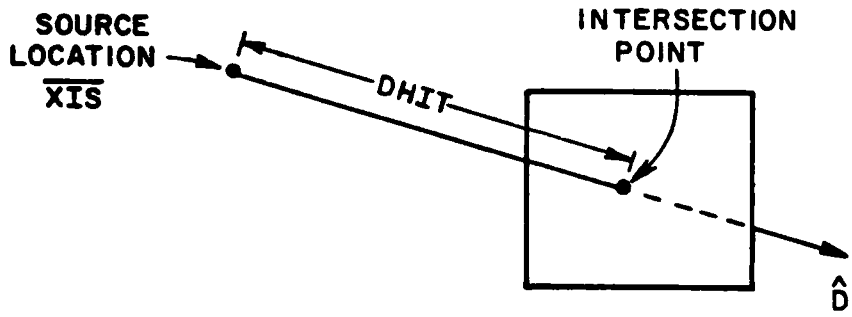


a. Reflection occurs (ray from image source hits plate MP)

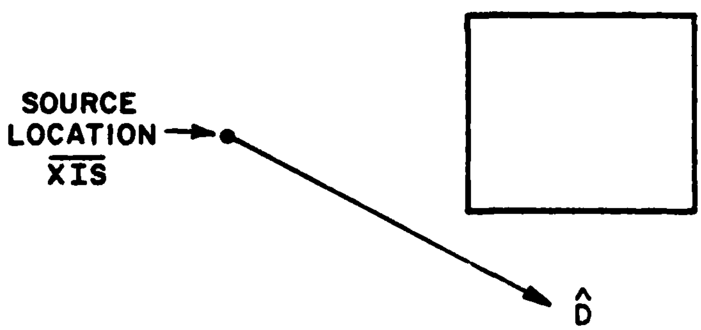


b. Reflection does not occur (ray from image source does not hit plate MP)

Figure 1. Geometry for Determining if Reflection from a Given Plate Occurs



a. Ray hits plate, LHIT = .TRUE



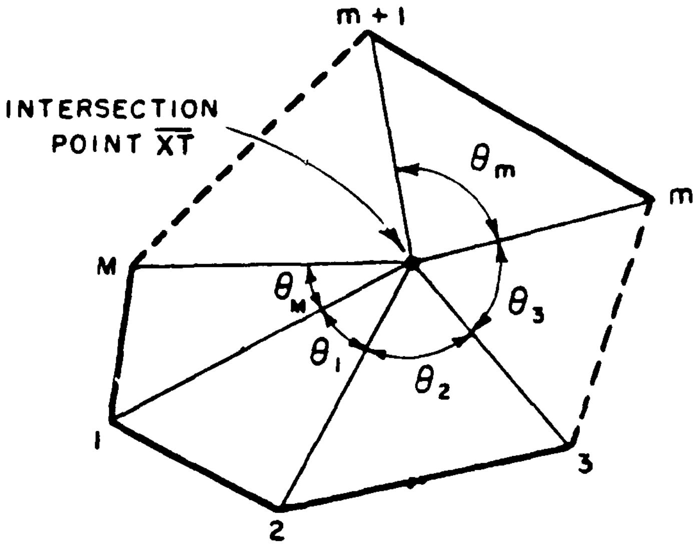
b. Ray does not hit plate, LHIT = .FALSE

Figure 2. Geometry for Determining if a Ray Does or Does Not Hit Plate

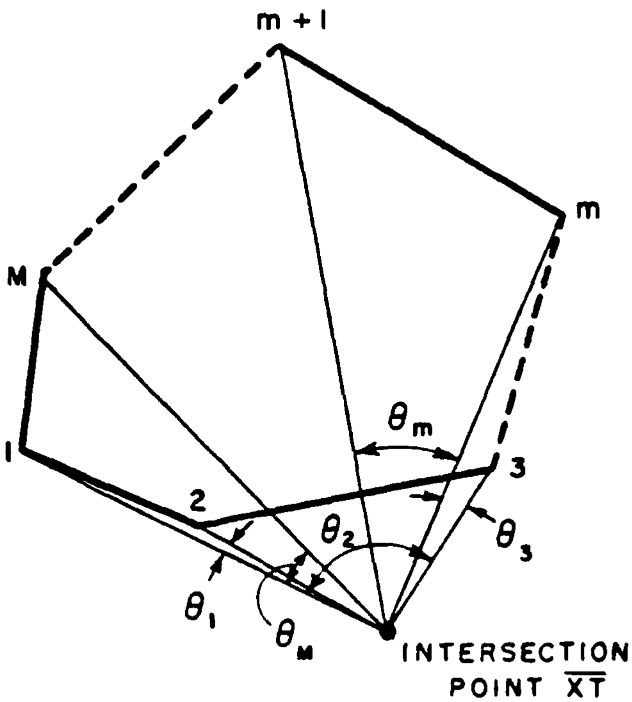
The hit algorithm first tests to see if a ray in the scatter direction will intersect the plane which the plate lies in by comparing the signs of the dot product of the scatter direction and the plate normal and the dot product of the vector from the source to a corner of the plate and the plate normal. If a hit is possible, the intersection point on the plate plane is determined. Whether the intersection point lies within the bounds of the plate is tested by summing the angles formed by the vectors from the intersection point to the various corners of the plate as shown in figure 3. If the sum is zero, the intersection point does not fall within the bounds of the plate. If the sum is 2π, the intersection point does fall within the bounds and the ray hits the plate. (See pp. 38-41, reference A.)

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN	Dot product of vector from edge 1 of plate MP to source and plate unit normal
CP	Computational variable
D	X,Y, and Z components of propagation direction in reference coordinate system
DBI	Computational variable
DBT	Computational variable
DHIT	Distance from source to nearest hit point
DHT	Distance from source to hit point
DN	Dot product of propagation direction unit vector and plate unit normal
LGRND	Logical variable presently set .FALSE. (It could be used to indicate the presence of a ground plane if that was allowed in the geometry.)
LHIT	Logical variable (set true if ray hits at least one plate)
LPLA	Logical variable set true if plates are present in geometry
LSTD	Set true if plate MP totally shadows plate MH from the source



a. Ray hits plate



b. Ray does not hit plate

Figure 3. Geometry for Deciding Whether Ray Which Hits Plate Plane Hits Finite Plate

LSTS	Set true if total shadowing routine is being used
ME	DO loop variable
MEP	Array which defines the number of edges on each plate
MEX	Number of edges on plate MP
MH	Shows which plates are to be checked: MH=-MP only plate MP is checked MH=0 all plates are checked MH=MP all plates except MP are checked
MME	Indicator for second corner on an edge
MP	Index variable (number of plate being checked)
MPH	Index variable
MPP	DO loop variable
N	DO loop variable
PI	π
RD	Computational variable
VN	Array which includes the unit normal vector for each plate
X	Array which includes corner locations for all plates
XIS	X,Y,Z components of source location in reference coordinate system (entering routine) X,Y,Z components of hit position (leaving routine)
XT	X,Y,Z components of point where ray intersects plate plane

5. I/O VARIABLES:

A. INPUT — LOCATION

INPUT	LOCATION
D	F.P.
LPLA	/LPLCY/
LSTD	/LSHDP/
LSTS	/LSHDP/
MEP	/GEOPLA/
MH	F.P.
PI	/PIS/
VN	/GEOPLA/
X	/GEOPLA/
XIS	F.P.

B. OUTPUT — LOCATION

OUTPUT	LOCATION
DHIT	F.P.
LHIT	F.P.
LSTD	/LSHDP/
XIS	F.P.

6. CALLING ROUTINES:

DIFPLT

DPLRCL

DPLRPL

ENDIF

GEOM

GTDDRV

INCFLD

RCLDPL

RCLRPL

REFCAP

REFCYL

REFPLA

RPLDPL

RPLRCL

RPLRPL

RPLSCL

SCLRPL

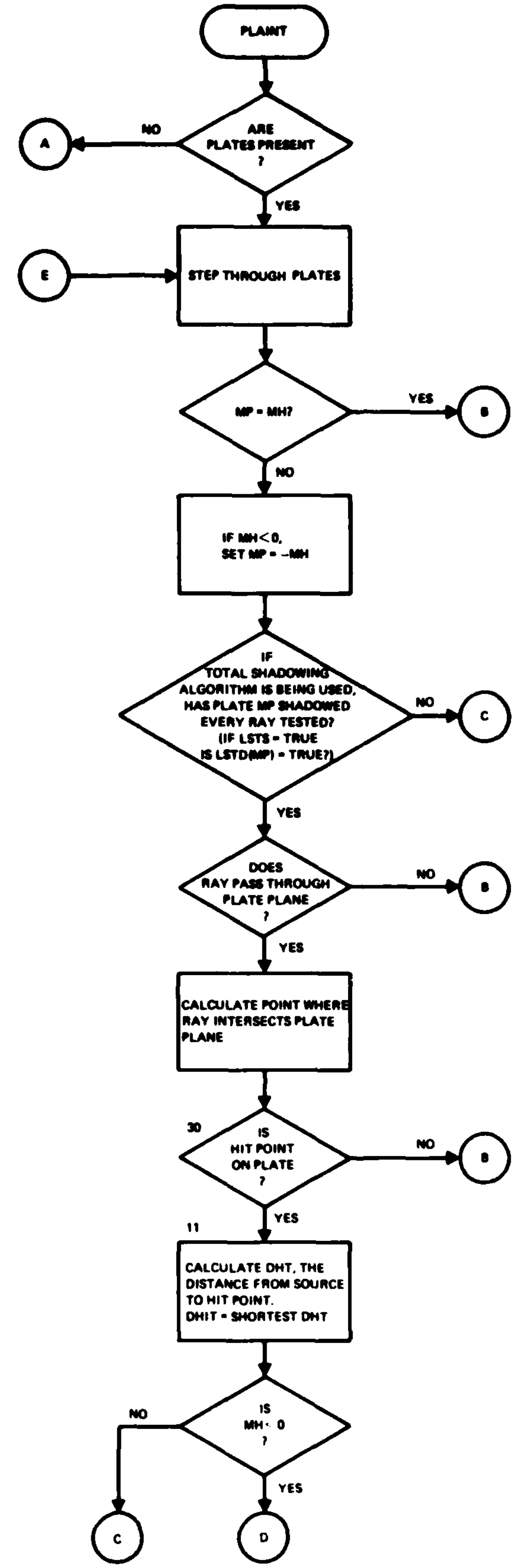
SCTCYL

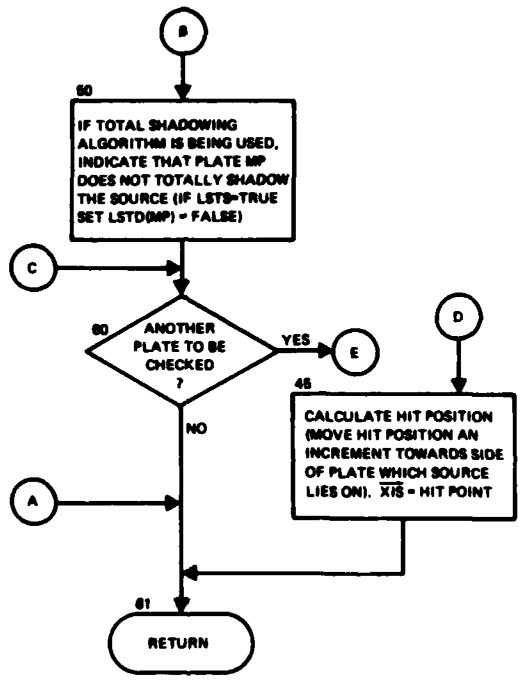
7. CALLED ROUTINE:

BTAN2

8. REFERENCE:

A. R. J. Marhefka, "Analysis of Aircraft Wing-Mounted Antenna Patterns," Report 2902-25, June 1976, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Grant No. NGL 36-008-138 for National Aeronautics and Space Administration.


PLAINT
ARE PLATES PRESENT ?
NO
A
YES
E
STEP THROUGH PLATES
MP = MH?
YES
B
NO
IF MH<0, SET MP = –MH
IF TOTAL SHADOWING ALGORITHM IS BEING USED, HAS PLATE MP SHADOWED EVERY RAY TESTED? (IF LSTS = TRUE IS LSTD(MP) = TRUE?)
NO
C
YES
DOES RAY PASS THROUGH PLATE PLANE ?
NO
B
YES
CALCULATE POINT WHERE RAY INTERSECTS PLATE PLANE
30
IS HIT POINT ON PLATE ?
NO
B
YES
11
CALCULATE DHT, THE DISTANCE FROM SOURCE TO HIT POINT. DHIT = SHORTEST DHT
IS MH < 0 ?
NO
C
YES
D


B
50
IF TOTAL SHADOWING ALGORITHM IS BEING USED, INDICATE THAT PLATE MP DOES NOT TOTALLY SHADOW THE SOURCE (IF LSTS=TRUE SET LSTD(MP) = FALSE)
C
60
ANOTHER PLATE TO BE CHECKED ?
YES
E
NO
D
45
CALCULATE HIT POSITION (MOVE HIT POSITION AN INCREMENT TOWARDS SIDE OF PLATE WHICH SOURCE LIES ON). XIS = HIT POINT
A
61
RETURN

1. NAME: PLATE (INPUT)

2. PURPOSE: To store raw GTD plate geometry data into the segment table.

3. METHOD: WYRDRV calls PLATE whenever a PL command is encountered in the geometry data. PLATE interprets the command items as scanned by SCAN and extracts values for plate number and coordinates. This information is stored in one or more sequential SEGTBL entries.

There are two possible formats for the PL command: (1) coordinates specified by PNTTBL point numbers; or (2) coordinates specified as x,y,z ordered triples. The latter format is assumed if there are more than eight fields on a PL command. The first format is

PL	nn	p1	p2	p3	[p4]	[p5]	[p6]	
1	2	3	4	5	6	7	8	NVAL/VAL index

nn = user-assigned plate number

p1,p2,... = point numbers defining corners of the plate

The second format is:

PL nn x1 y1 z1 x2 y2 z2 x3 y3 z3 [...]

nn = user-assigned plate numbers

x1,y1,z1 = coordinates of corner one

x2,y2,z2 = coordinates of corner two

x3,y3,z3 = coordinates of corner three

[...] = coordinates of corners four-six

The two formats may not be mixed on the same PL command.

PLATE places the following values in SEGTBL:

	First SEGTBL entry	Second SEGTBL entry
1	ITAG/ISG	ITAG/ISG
2	LINK #	LINK #

3	x1	x4
4	y1	y4
5	z1	z4
6	x2	x5
7	y2	y5
8	z2	z5
9	x3	x6
10	y3	y6
11	z3	z6

Plates having only three corners require just the first entry. Should plates with more than six corners be implemented in the future, more SEGTBL entries will be required. The routine can accommodate plates with more corners: (1) by changing the dimensions of x, y, and z; and (2) by changing the value of MXPLAR (maximum number of PL command arguments) in BLKDAT.

The sequential SEGTBL entries are linked together by LINK #. LINK # is a packed word containing (100 times the user plate number plus the number of corners) in the left-half word and the location of the next SEGTBL entry for this plate in the right-half word. To identify the first of a series of entries, the number of corners is set to zero in the LINK # for all but the first entry.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CLNK	Equivalent to NCLNK
D	Dummy argument
ERRFLG	Internal flag to indicate command error (integer)
IERF	Flat plate error flag
IPT	PNTTBL point number of plate corner
ISG	Plate segment number (assigned by SEGTBL)
ITAG	Plate tag number
ITPARG	Array of variable types (real, integer) for argument field

LNKNUM	Location of next SEGTBL entry for this plate
LNKSV	Location of first SEGTBL entry for this plate
MLIM	Number of coordinates stored in a SEGTBL entry
NC	Number of corners on plate
NCLNK	Link word containing plate number, number of corners, and LNKNUM
NCORN	Number of corners used in SEGTBL entry
NSGTBL	Number of SEGTBL entries for this plate
NUMPL	User-assigned plate number
X,Y,Z	Arrays of corner coordinates
XX,YY,ZZ	Arrays of corner coordinates placed in a SEGTBL entry

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
IPLTAG	/GTDDAT/
IP217	/GEODAT/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
LUPRNT	/ADEBUG/
MXPLAR	/GTDDAT/
NARGS	/SCNPAR/
NCODE	/SCNPAR/
NTFLPT	/ADEBUG/
NTINT	/ADEBUG/

NUMSEG	/SEGMNT/
NVAL	/SCNPAR/
SCALE	/SEGMNT/
VAL	/SCNPAR/

B. OUTPUT LOCATION

NOGOFG	/SCNPAR/

6. CALLING ROUTINE:

   WYRDRV

7. CALLED ROUTINES:

   ASSIGN

   FLTPLT

   GETPNT

   PUTSEG

   STATIN

   STATOT

   WLKBCK

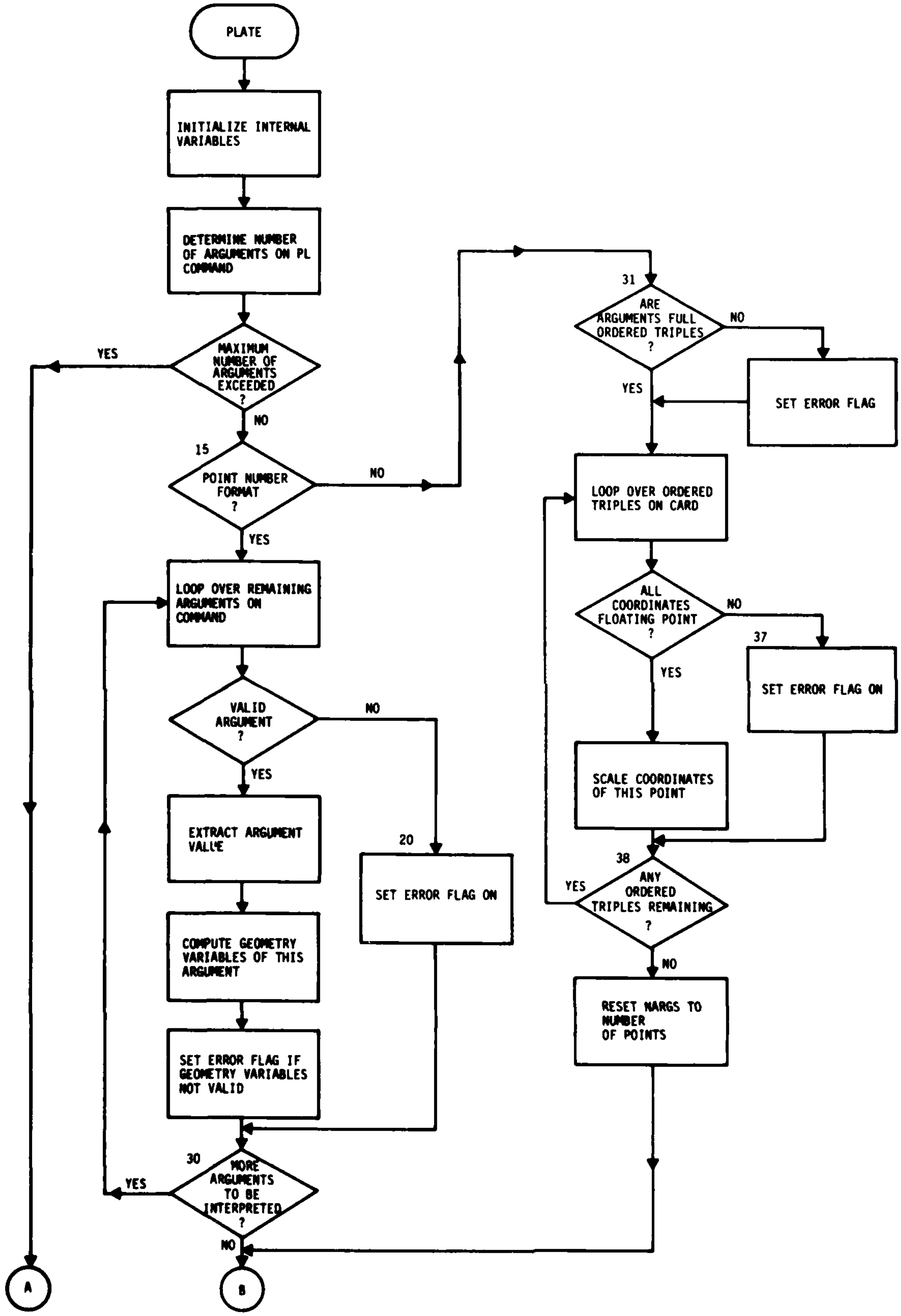
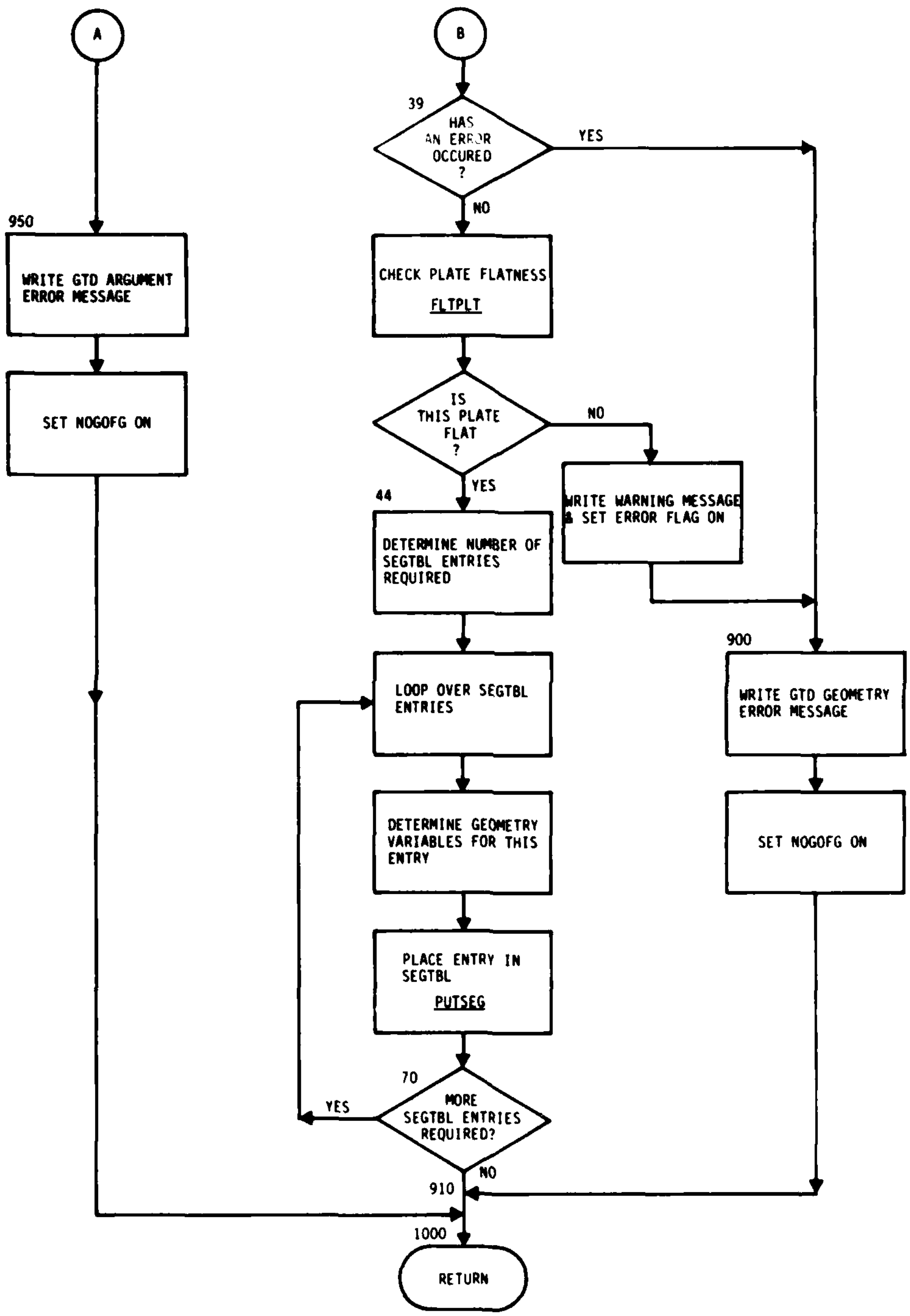


PLATE
INITIALIZE INTERNAL VARIABLES
DETERMINE NUMBER OF ARGUMENTS ON PL COMMAND
MAXIMUM NUMBER OF ARGUMENTS EXCEEDED ?
YES
NO
15
POINT NUMBER FORMAT ?
NO
YES
LOOP OVER REMAINING ARGUMENTS ON COMMAND
VALID ARGUMENT ?
NO
YES
EXTRACT ARGUMENT VALUE
20
SET ERROR FLAG ON
COMPUTE GEOMETRY VARIABLES OF THIS ARGUMENT
SET ERROR FLAG IF GEOMETRY VARIABLES NOT VALID
30
MORE ARGUMENTS TO BE INTERPRETED ?
YES
NO
A
B
31
ARE ARGUMENTS FULL ORDERED TRIPLES ?
NO
YES
SET ERROR FLAG
LOOP OVER ORDERED TRIPLES ON CARD
ALL COORDINATES FLOATING POINT ?
NO
37
SET ERROR FLAG ON
YES
SCALE COORDINATES OF THIS POINT
38
ANY ORDERED TRIPLES REMAINING ?
YES
NO
RESET NARGS TO NUMBER OF POINTS


A
B
39
HAS AN ERROR OCCURED ?
YES
NO
950
WRITE GTD ARGUMENT ERROR MESSAGE
CHECK PLATE FLATNESS
FLTPLT
SET NOGOFG ON
IS THIS PLATE FLAT ?
NO
YES
44
WRITE WARNING MESSAGE & SET ERROR FLAG ON
DETERMINE NUMBER OF SEGTBL ENTRIES REQUIRED
900
LOOP OVER SEGTBL ENTRIES
WRITE GTD GEOMETRY ERROR MESSAGE
DETERMINE GEOMETRY VARIABLES FOR THIS ENTRY
SET NOGOFG ON
PLACE ENTRY IN SEGTBL
PUTSEG
70
YES
MORE SEGTBL ENTRIES REQUIRED?
NO
910
1000
RETURN

1. **NAME:** PLIST (INPUT)

2. **PURPOSE:** Parse a list of integers, symbols, or keywords.

3. **METHOD:** The list of arguments must be at the end of the command text. If not, an error will be generated. The routines will parse a list of integers, symbols, or keywords into the parse table argument list.

4. **INTERNAL VARIABLES:**

VARIABLE	DEFINITION
INC	Intrinsic function
INDEX	Location in NVAL array
ITEMCD	Used to define the type of argument.
KWNDEX	Keyword index.
KWTKNM	Keyword task number.
NPARAM	Type of list to be printed

5. **I/O VARIABLES:**

A. INPUT

INPUT	LOCATION
IMINUS	/SCNPAR/
ISOFF	/ADEBUG/
KWFMTP	/PARTAB/
KWILP	/PARTAB/
LSTDAT	/SCNPAR/
LSTINT	/SCNPAR/
NARGMX	/PARTAB/
NCODE	/SCNPAR/
NPARAM	F.P.
NPARGL	/PARTAB/

NPEARG	/INPERR/
NPELST	/INPERR/
NPENOI	/INPERR/
NPERGE	/INPERR/
NTAB	/SCNPAR/
NTALPH	/ADEBUG/
NTEND	/ADEBUG/
NTERR	/ADEBUG/
NTINT	/ADEBUG/
NTKEYW	/ADEBUG/
NTSFMT	/PARTAB/
NTSYMB	/ADEBUG/
NVAL	/SCNPAR/
NVALMX	/SCNPAR/

B. OUTPUT LOCATION

NARGTB	/PARTAB/
NCODE	/SCNPAR/
NPARGL	/PARTAB/
NPRSER	/SCNPAR/
NTAB	/SCNPAR/
NUMWRD	/ADEBUG/
NVAL	/SCNPAR/

6. CALLING ROUTINE:

FNDARG

7. CALLED ROUTINES:

ASSIGN

FABLO2

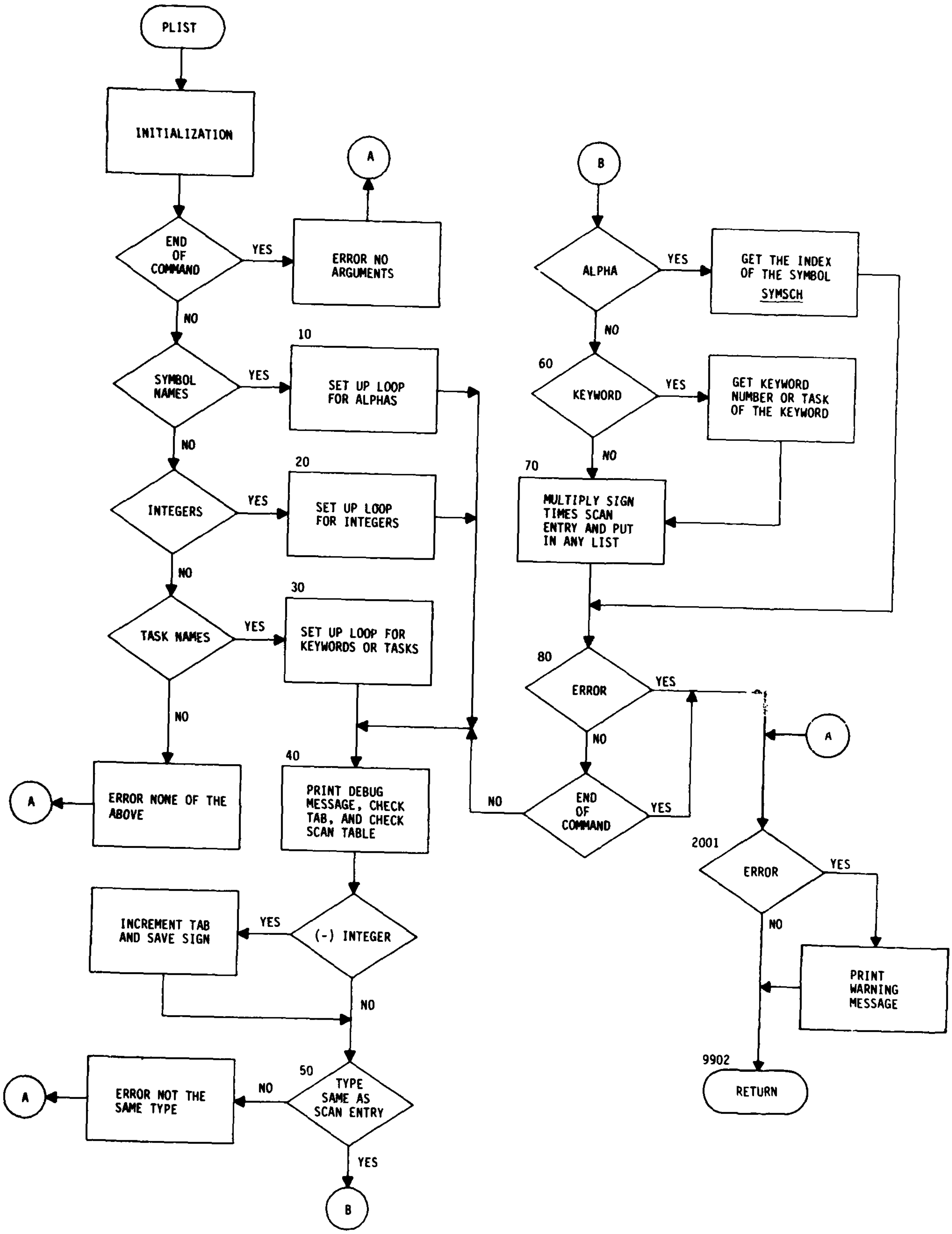
STATIN

STATOT

SYMSCH

WLKBCK

PLIST (INPUT)


PLIST
INITIALIZATION
END OF COMMAND
YES
ERROR NO ARGUMENTS
A
NO
10
SYMBOL NAMES
SET UP LOOP FOR ALPHAS
20
INTEGERS
SET UP LOOP FOR INTEGERS
30
TASK NAMES
SET UP LOOP FOR KEYWORDS OR TASKS
ERROR NONE OF THE ABOVE
40
PRINT DEBUG MESSAGE, CHECK TAB, AND CHECK SCAN TABLE
INCREMENT TAB AND SAVE SIGN
(-) INTEGER
50
TYPE SAME AS SCAN ENTRY
ERROR NOT THE SAME TYPE
B
ALPHA
GET THE INDEX OF THE SYMBOL SYMSCH
60
KEYWORD
GET KEYWORD NUMBER OR TASK OF THE KEYWORD
70
MULTIPLY SIGN TIMES SCAN ENTRY AND PUT IN ANY LIST
80
ERROR
2001
PRINT WARNING MESSAGE
9902
RETURN

1. NAME: PLTSEG (INPUT)

2. PURPOSE: Identify the plate (if any) to which a point is adjacent.

3. METHOD: For each plate in the geometry data set, it is determined (1) whether or not the point lies in the plane of the plate and (2) whether or not the point lies within the boundaries of the plate. If the point satisfies both conditions the plate number is returned. Otherwise zero is returned.

To determine if the point is in the plane of the plate, the unit vector of the ray to the point from a plate corner is compared to the unit normal of the plate (figure 1). If the dot product is zero, the point is in the plate plane.

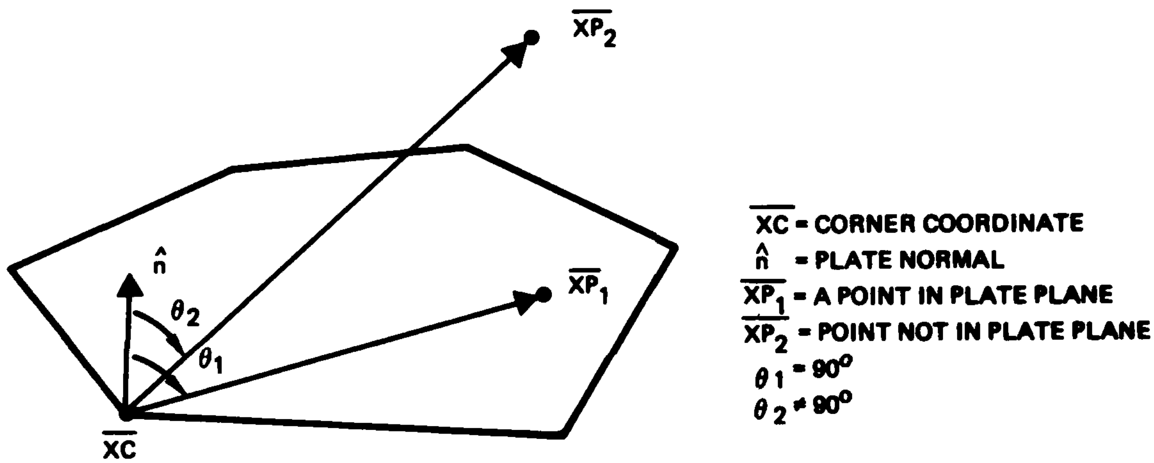
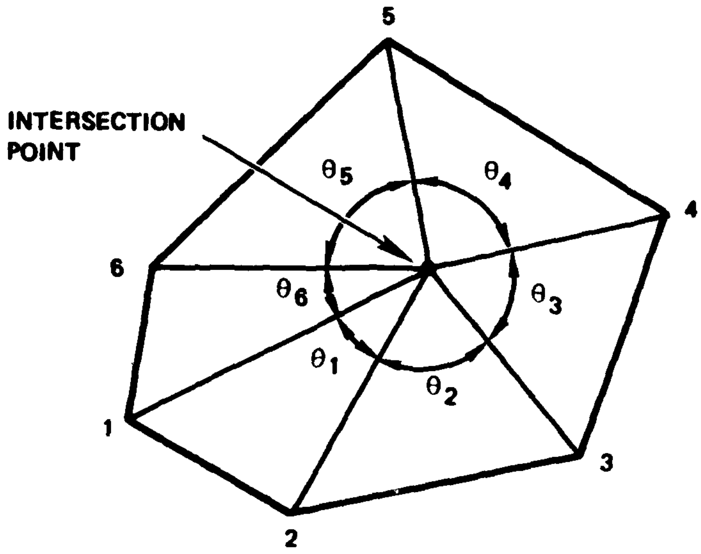


Figure 1. Determining if a Point Lies in the Plane of the Plate

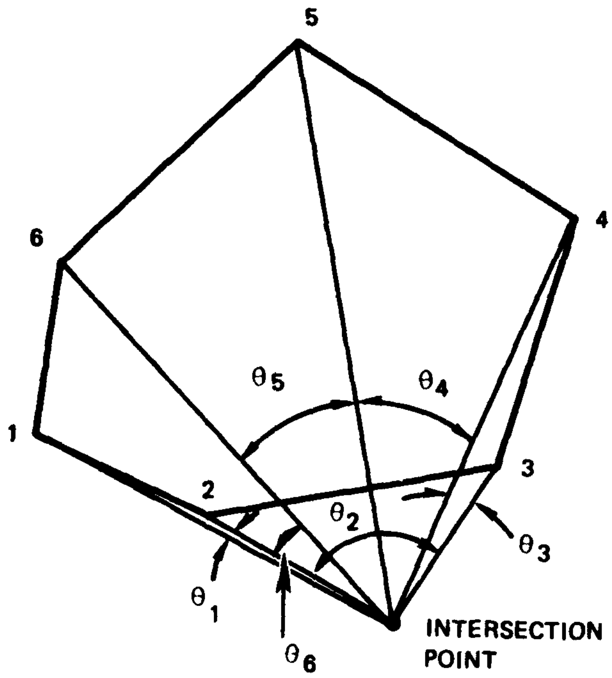
To find if the point is within the boundaries of the plate, the angles formed by the vectors from the point to the various corners are summed as shown in figure 2. If the sum is zero, the point is not on the plate; if the sum is 2π, the point is on the plate.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN	Dot product of plate unit normal and unit vector of ray from corner to point
CP	Dot product of edge vector and cross product of vectors to point from adjacent corners



(A) POINT ON PLATE ($\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 + \theta_6 = 2\pi$)



(B) POINT NOT ON PLATE ($\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 + \theta_6 = 0$)

Figure 2. Geometry for Deciding Whether a Point Is in the Plate Plane on the Plate

DBI	Angle at point subtended by edge
DBT	Sum of all angles subtended by edge
EN	Edge unit normal vector
ENM	Magnitude of normal vector
ENX,ENY,ENZ	A normal vector
E1X,E1Y,E1Z	Vector direction of edge one
E2X,E2Y,E2Z	Vector direction of edge two
I,IP	Index of geometry block in SEGTBL
IG	Index of SEGTBL entry
IG1	Index of SEGTBL entry when second plate entry is in next geometry block
IGBLK1	Geometry block when second plate entry is in next geometry block
IGLIM	Last SEGTBL entry with GTD geometries
IGLIM1	Last GTD SEGTBL entry when second plate entry is in next geometry block
IGLOW	First SEGTBL entry with GTD geometries
IGLOW1	First GTD SEGTBL entry when second plate entry is in next geometry block
IHITPL	Plate number where point lies (zero if no intersection)
IPLOW	First geometry block with GTD entries
IPMAX	Last geometry block with GTD entries
ITAG	Geometry element tag number
ME	Corner number of first corner of edge
MM	Corner index
MME	Corner number of second corner of edge

MMM	Coordinate index in SEGTBL
NCORN	Number of plate corners
NDXSAV	Original block of geometry in SEGTBL
NTRY	Number of SEGTBL entries for plate
NUMPL	User-assigned plate number
RD	Cross product of vectors to point from adjacent corners
RP	Distance from corner to point
X	Array of corner coordinates
XP,YP,ZP	Vector from corner to point
XS,YS,ZS	Point coordinates
XT	Point coordinates

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
IPLTAG	/GTDDAT/
IP217	/GEODAT/
ISGTBL	/SEGMNT/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
MAXBLK	/SEGMNT/
MAXSEG	/SEGMNT/
NDXBLK	/SEGMNT/
NPATCH	/SEGMNT/
NUMPLT	/GTDDAT/
NUMSEG	/SEGMNT/

NWIRE	/SEGMNT/
SEGTBL	/SEGMNT/
XS	F.P.
YS	F.P.
ZERO	/ADEBUG/
ZS	F.P.

B. OUTPUT — LOCATION

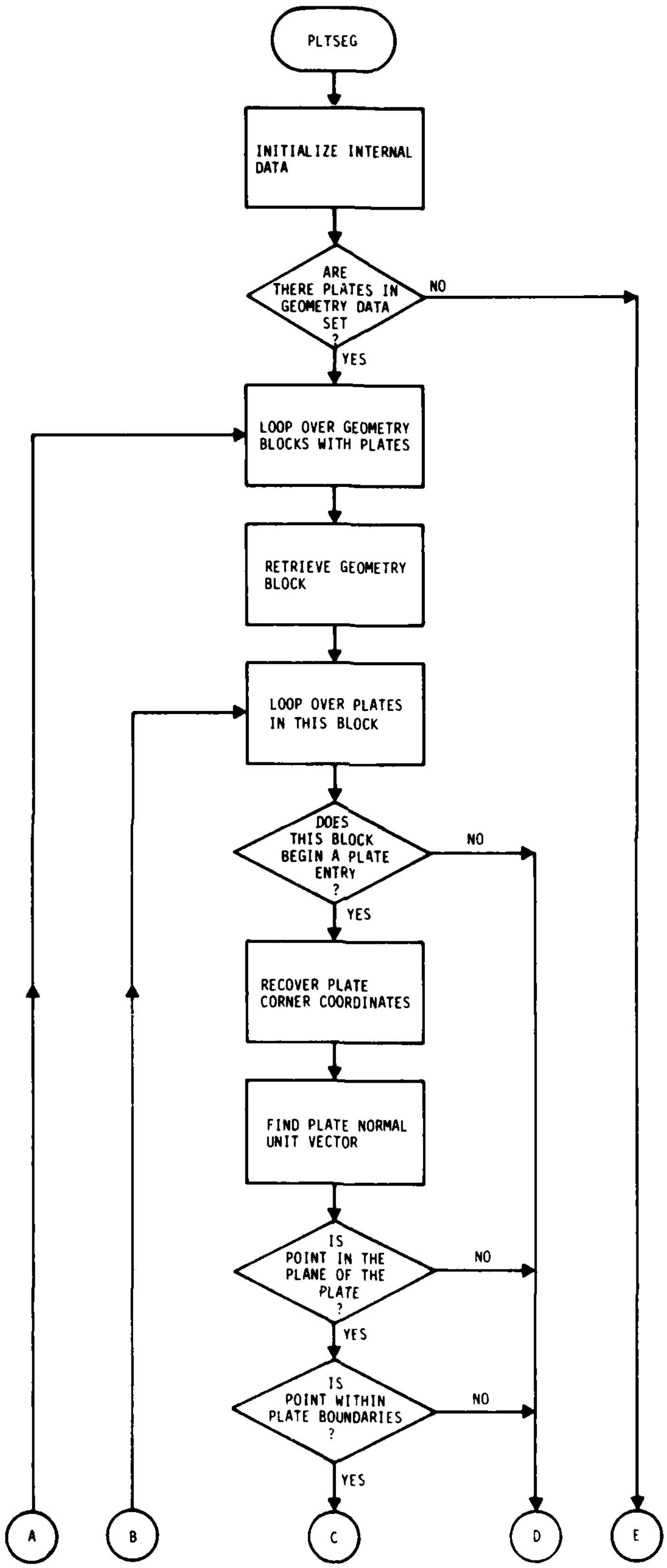
IHITPL	F.P.

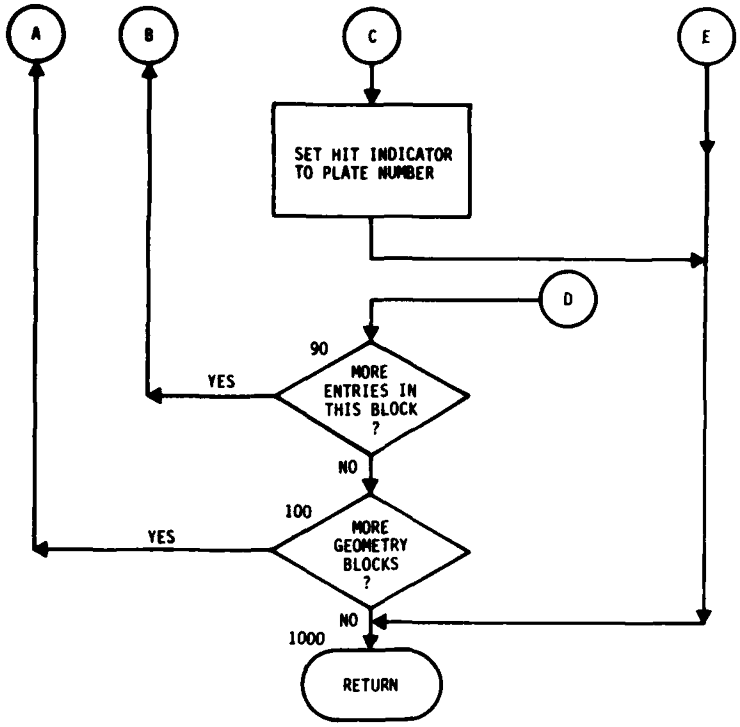
6. CALLING ROUTINE:

CNVGTD

7. CALLED ROUTINES:

ASSIGN	STATOT
GETSEG	WLKBCK
STATIN	


PLTSEG
INITIALIZE INTERNAL DATA
ARE THERE PLATES IN GEOMETRY DATA SET ?
NO
YES
LOOP OVER GEOMETRY BLOCKS WITH PLATES
RETRIEVE GEOMETRY BLOCK
LOOP OVER PLATES IN THIS BLOCK
DOES THIS BLOCK BEGIN A PLATE ENTRY ?
NO
YES
RECOVER PLATE CORNER COORDINATES
FIND PLATE NORMAL UNIT VECTOR
IS POINT IN THE PLANE OF THE PLATE ?
NO
YES
IS POINT WITHIN PLATE BOUNDARIES ?
NO
YES
A
B
C
D
E


A
B
C
E
SET HIT INDICATOR TO PLATE NUMBER
D
90
YES
MORE ENTRIES IN THIS BLOCK ?
NO
100
YES
MORE GEOMETRY BLOCKS ?
NO
1000
RETURN

1. **NAME:** POLYRT (GTD)

2. **PURPOSE:** To solve an Mth order polynomial equation.

3. **METHOD:** This subroutine solves for the roots of an Mth order polynomial,

$$C_M Z^M + C_{M-1} Z^{M-1} + \ldots C_1 Z^1 + C_o = 0.$$

The roots of the polynomial are found using the Newton-Raphson method of iterated synthetic division (see reference A). The coefficients are stored such that C_M = CC(M+1), C_O = CC(1), etc.

4. **INTERNAL VARIABLES:**

VARIABLE	DEFINITION
C	Working array of polynomial coefficients
CC	A complex array containing the polynomial coefficients
CMAX	Magnitude of largest coefficient
CNEW	Array containing coefficients of polynomial left after the prospective root has been factored out
CNNW	Array containing coefficients of polynomial left after the prospective root has been factored out twice
EPS	Small number (relative to largest coefficient)
ICONJ	Index for trying the conjugate of the previous root as a guess
ICCUNT	Index on the number of times the iteration procedure searches for a root
IFLAG	Flag used to indicate if all possible starting values have been tried
ISTART	Index for starting values

LIMIT	Maximum number of iterations used to search for the root
M	Order of polynomial being worked on
MI	Computational variable
MM	Order of the equation
MMP1	MM plus one
MN	Order of once-factored polynomial being worked on
Q	Magnitude of polynomial coefficients
R	A complex array containing the roots of the equation
RJ	Remainder left after prospective root has been factored out
RJP	Remainder left after prospective root has been factored out twice
RT	Prospective root being iterated
SR	Square root of (C(2)*C(2)-4*C(1)*C(3))
START	Array containing initial guess of root locations
TEST	Bound used to determine if the prospective root has converged
XI	Imaginary part of CC
XR	Real part of CC

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
CC	F.P.
MM	F.P.
LUPRNT	/ADEBUG/

B. OUTPUT — LOCATION

OUTPUT	LOCATION
R	F.P.

6. CALLING ROUTINES:

   DFPTCL

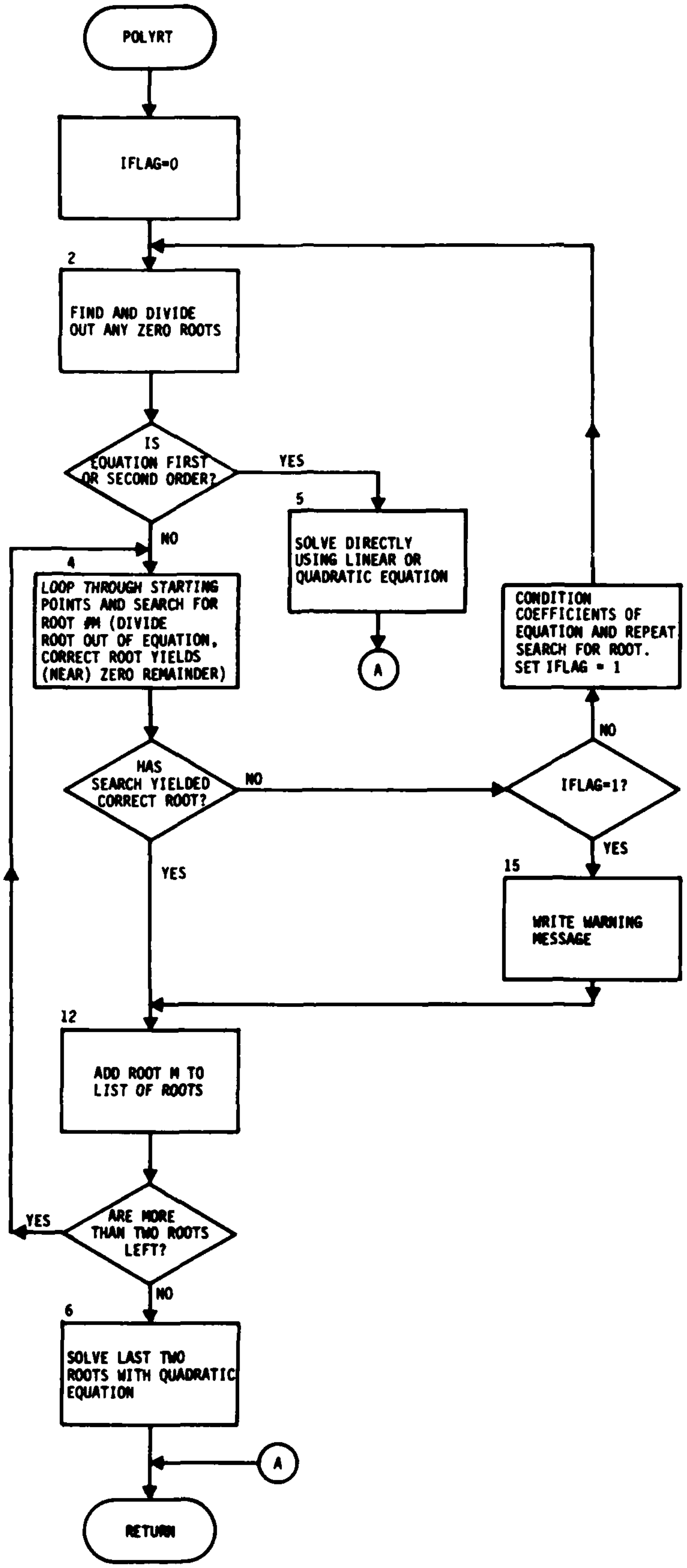
   RFDFIN

7. CALLED ROUTINE:

   BABS

8. REFERENCE:

A. A. Ralston, A First Course in Numerical Analysis, McGraw-Hill Book Co., New York, 1965, pp. 371-373.


POLYRT
IFLAG=0
2
FIND AND DIVIDE OUT ANY ZERO ROOTS
IS EQUATION FIRST OR SECOND ORDER?
YES
5
SOLVE DIRECTLY USING LINEAR OR QUADRATIC EQUATION
A
NO
4
LOOP THROUGH STARTING POINTS AND SEARCH FOR ROOT #M (DIVIDE ROOT OUT OF EQUATION, CORRECT ROOT YIELDS (NEAR) ZERO REMAINDER)
CONDITION COEFFICIENTS OF EQUATION AND REPEAT SEARCH FOR ROOT. SET IFLAG = 1
NO
HAS SEARCH YIELDED CORRECT ROOT?
NO
IFLAG=1?
YES
15
WRITE WARNING MESSAGE
YES
12
ADD ROOT M TO LIST OF ROOTS
ARE MORE THAN TWO ROOTS LEFT?
YES
NO
6
SOLVE LAST TWO ROOTS WITH QUADRATIC EQUATION
A
RETURN

1. NAME: POSTIP (INPUT)

2. PURPOSE: Subroutine to print parse table entries after input processing.

3. METHOD: If the output file is other than zero, the parse table, argument list table, loop table, and literal table entries are output to the file specified.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IDATA	Temporary location for Hollerith name.
ITASK	Task number from argument list.
NCDNDX	Task name index to NCODES.
NDXTSK	Index of task number in argument list.
NMARGS	Pointer to argument list.
NMLITN	Pointer to literal table.
NMLOOP	Pointer to loop table.
NMSYMB	Pointer to symbol table.
NMTASK	Pointer to task table.
NXTARG	Index of task entries in argument list.

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
ISYMBL	/SCNPAR/
KWNAME	/PARTAB/
LITNUM	/PARTAB/
LUDBUG	/ADEBUG/
LUPRNT	/ADEBUG/
NAMTSK	/PARTAB/

NARGTB	/PARTAB/
NCODES	/PARTAB/
NDATBL	/PARTAB/
NLOOPS	/PARTAB/
NPARGL	/PARTAB/
NPDATA	/PARTAB/
NPLITN	/PARTAB/
NPLOOP	/PARTAB/
NPTASK	/PARTAB/
NTALPH	/ADEBUG/
NTINT	/ADEBUG/
NTKEYW	/ADEBUG/
NTSKTB	/PARTAB/
NTSYMB	/ADEBUG/

B. OUTPUT LOCATION

NONE

6. CALLING ROUTINES:

INPDRV

RESTRT

7. CALLED ROUTINES:

ASSIGN

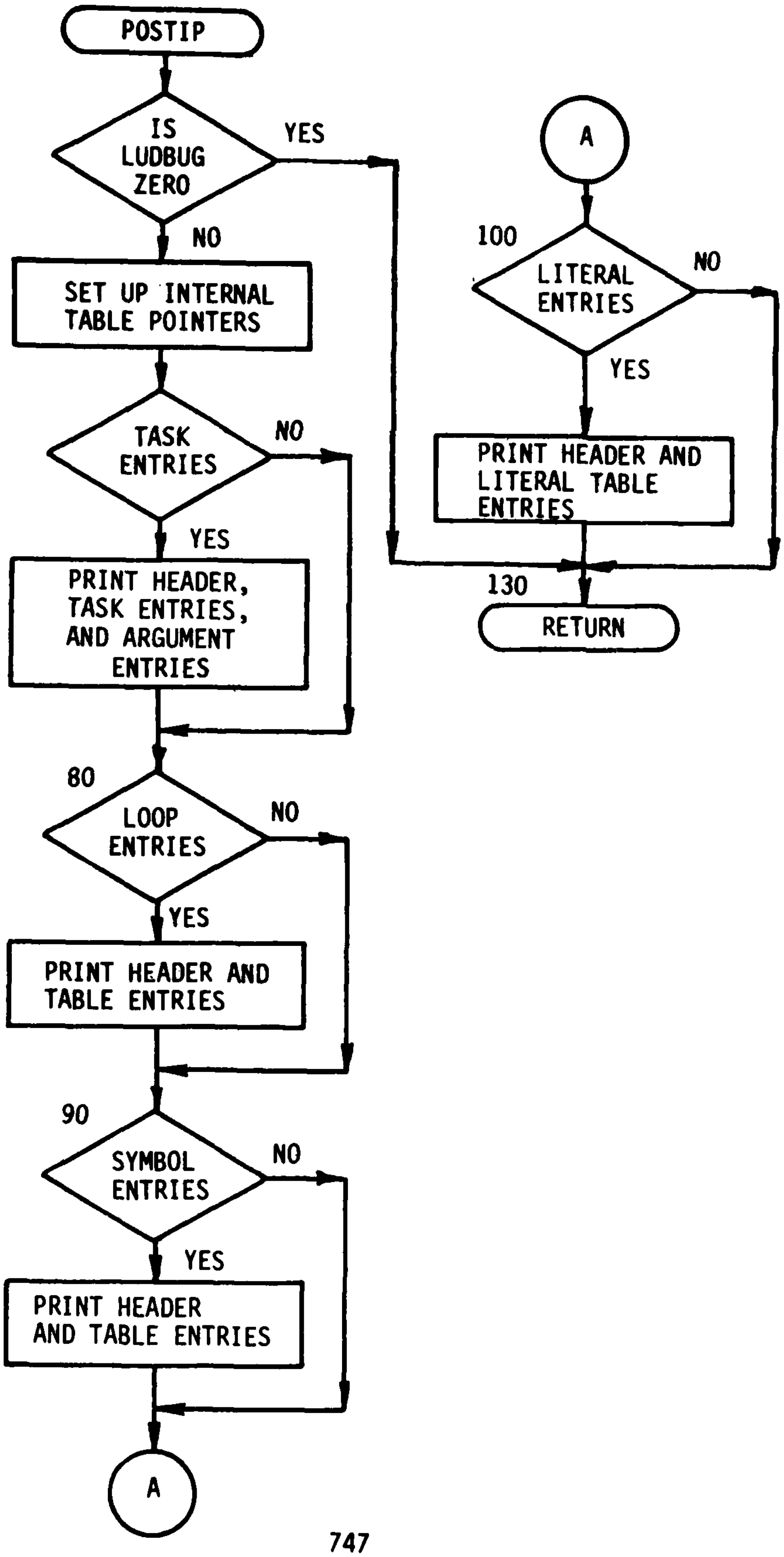
CONVRT

STATIN

STATOT

WLKBCK

POSTIP (INPUT)



1. NAME: POSTPR (INPUT)

2. PURPOSE: Post-processor for the PARSE subroutine

3. METHOD: Subroutine prints out the entries into the task table, argument list table, symbol table, loop table, and literal table made during the parsing of the last command entered.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IDATA	Temporary location for a Hollerith name
I2	Index to first argument of command
I3	Index to first symbol for current command
I5	Index to first literal for current command
NI1	Number of task table entries for command
NI2	Number of argument table entries for command
NI3	Number of symbol table entries for command
NI4	Number of loop table entries for command
NI5	Number of literal table entries for command
N1	Last entry index for task table
N2	Last entry index for argument list
N3	Last entry index for symbol table
N4	Last entry index for loop table
N5	Last entry index for literal table

5. I/O VARIABLES:

A. INPUT	LOCATION
IPSARG	/SCNPAR/
IPSDAT	/SCNPAR/

IPSLIT	/SCNPAR/
IPSLOO	/SCNPAR/
IPSTSK	/SCNPAR/
ISON	/ADEBUG/
ISYMBL	/SCNPAR/
KWNAME	/PARTAB/
LITNUM	/PARTAB/
LUDBUG	/ADEBUG/
NARGTB	/PARTAB/
NCODES	/PARTAB/
NDATBL	/PARTAB/
NDEBUG	/SCNPAR/
NLOOPS	/PARTAB/
NPARGL	/PARTAB/
NPDATA	/PARTAB/
NPLITN	/PARTAB/
NPLOOP	/PARTAB/
NPTASK	/PARTAB/
NTALPH	/ADEBUG/
NTINT	/ADEBUG/
NTKEYW	/ADEBUG/
NTSKTB	/PARTAB/
NTSYMB	/ADEBUG/

B. OUTPUT LOCATION

NONE

6. CALLING ROUTINE:

PARSE

7. CALLED ROUTINES:

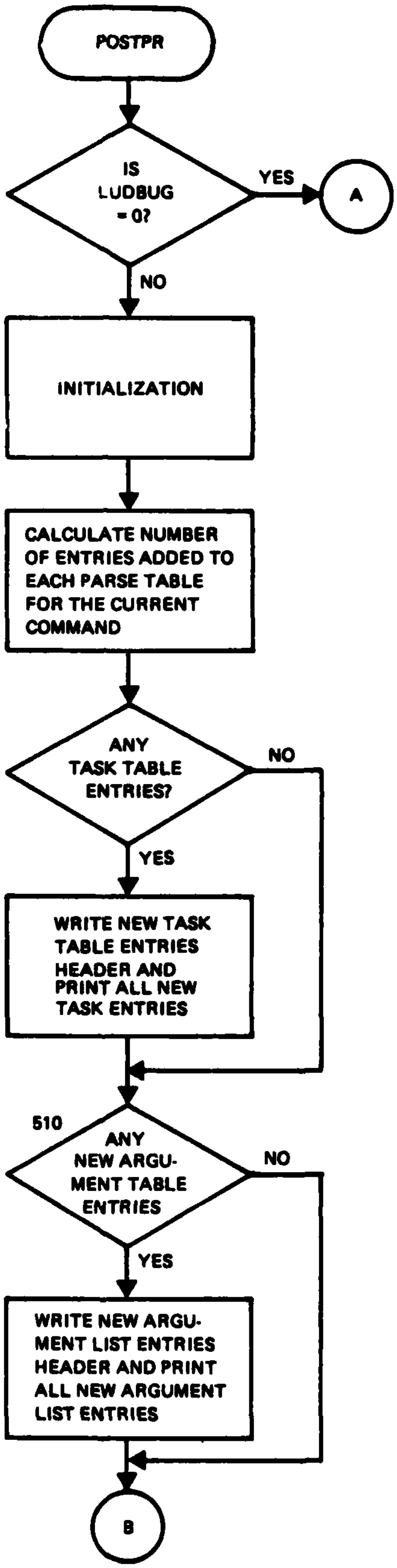
ASSIGN

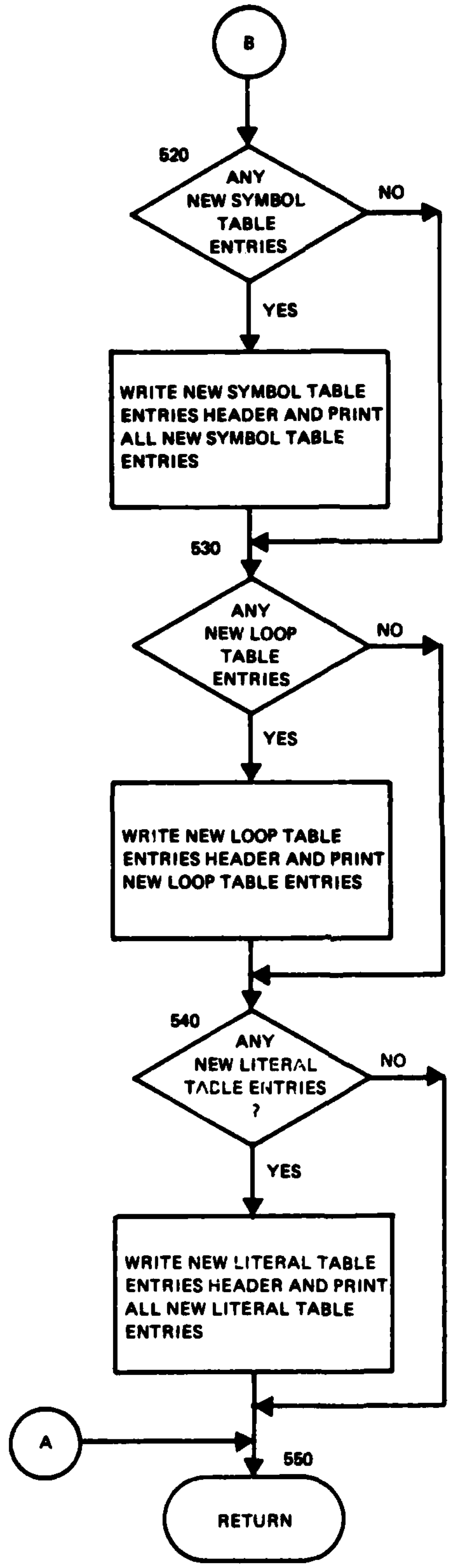
CONVRT

STATIN

STATOT

WLKBCK


POSTPR
IS LUDBUG = 0?
YES
A
NO
INITIALIZATION
CALCULATE NUMBER OF ENTRIES ADDED TO EACH PARSE TABLE FOR THE CURRENT COMMAND
ANY TASK TABLE ENTRIES?
NO
YES
WRITE NEW TASK TABLE ENTRIES HEADER AND PRINT ALL NEW TASK ENTRIES
510
ANY NEW ARGUMENT TABLE ENTRIES
NO
YES
WRITE NEW ARGUMENT LIST ENTRIES HEADER AND PRINT ALL NEW ARGUMENT LIST ENTRIES
B


B
520
ANY NEW SYMBOL TABLE ENTRIES
NO
YES
WRITE NEW SYMBOL TABLE ENTRIES HEADER AND PRINT ALL NEW SYMBOL TABLE ENTRIES
530
ANY NEW LOOP TABLE ENTRIES
NO
YES
WRITE NEW LOOP TABLE ENTRIES HEADER AND PRINT NEW LOOP TABLE ENTRIES
540
ANY NEW LITERAL TACLE ENTRIES ?
NO
YES
WRITE NEW LITERAL TABLE ENTRIES HEADER AND PRINT ALL NEW LITERAL TABLE ENTRIES
A
550
RETURN

1. NAME: PREPAR (INPUT)

2. PURPOSE: Print out scan tables before command is parsed.

3. METHOD: If the output file is nonzero, write out the scan table (NCODE and NVAL) for the current command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IDATA	Temporary location for a Hollerith name.
KODE	Current value of NCODE

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
ISON	/ADEBUG/
ISYMBL	/SCNPAR/
KWNAME	/PARTAB/
LUDBUG	/ADEBUG/
NAMTSK	/PARTAB/
NCODE	/SCNPAR/
NCODES	/PARTAB/
NDEBUG	/SCNPAR/
NTALPH	/ADEBUG/
NTEND	/ADEBUG/
NTERR	/ADEBUG/
NTINT	/ADEBUG/
NTKEYW	/ADEBUG/
NTSYMB	/ADEBUG/
NTTASK	/ADEBUG/

NVAL	/SCNPAR/
NVALMX	/SCNPAR/
VAL	/SCNPAR/
B. OUTPUT	LOCATION
NONE	

6. CALLING ROUTINE:

   PARSE

7. CALLED ROUTINES:

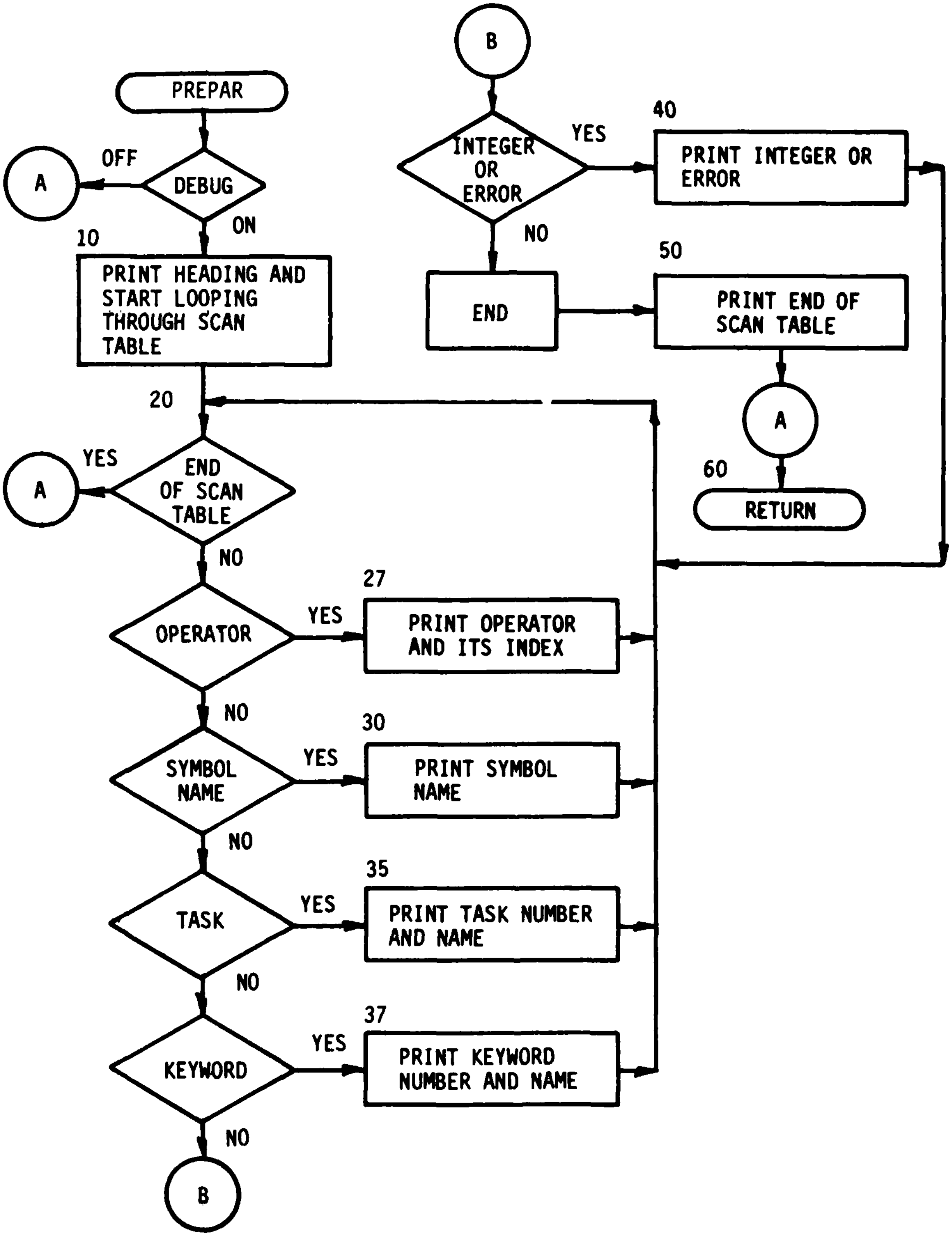
   ASSIGN

   CONVRT

   STATIN

   STATOT

   WLKBCK



PREPAR
DEBUG
OFF
A
ON
10
PRINT HEADING AND START LOOPING THROUGH SCAN TABLE
20
END OF SCAN TABLE
YES
A
NO
OPERATOR
YES
27
PRINT OPERATOR AND ITS INDEX
NO
SYMBOL NAME
YES
30
PRINT SYMBOL NAME
NO
TASK
YES
35
PRINT TASK NUMBER AND NAME
NO
KEYWORD
YES
37
PRINT KEYWORD NUMBER AND NAME
NO
B
B
INTEGER OR ERROR
YES
40
PRINT INTEGER OR ERROR
NO
END
50
PRINT END OF SCAN TABLE
A
60
RETURN

1. PRESCN (INPUT)

2. PURPOSE: Subroutine to prescan the task table and eliminate those tasks which have been affected by a WIPOUT command.

3. METHOD: Before return to the main routine, the task table is scanned for the presence of a WIPOUT command. If the command is found, the arguments of the command are retrieved from the argument list table and the appropriate commands in the task table are set to ON or OFF depending on their previous state.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICLEAR	Code for total wipe out of all commands
ICPTSK	Code for checkpoint
ILCNXT	Index to the next location in the task table
INTLIT	Code for integer in the literal table
ITASK	Index to task table
ITYPE	Entry in the task table
I1	One
JSTRT	First task to be wiped out
JTASK	Index to argument list
KTASK	Index to task table minus one
LOCARG	Index to argument list
LOCNXT	Index to the next argument in the list
LOCPN	Argument pointer for one task
LOCTSK	Task number for wipe out
NMARG	Number of arguments for a task
NM1	N minus one
NRSTRT	Code for a restart from checkpoint
NTSK	ITASK - 1

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
ISON	/ADEBUG/
LUPRNT	/ADEBUG/
NARGTB	/PARTAB/
NPTASK	/PARTAB/
NTSKTB	/PARTAB/
NUMWIP	/PARTAB/

B. OUTPUT

OUTPUT	LOCATION
CHKPNT	/SYSFIL/
IERRF	/ADEBUG/
NTSKTB	/PARTAB/

6. CALLING ROUTINE:

INPDRV

7. CALLED ROUTINES:

ASSIGN

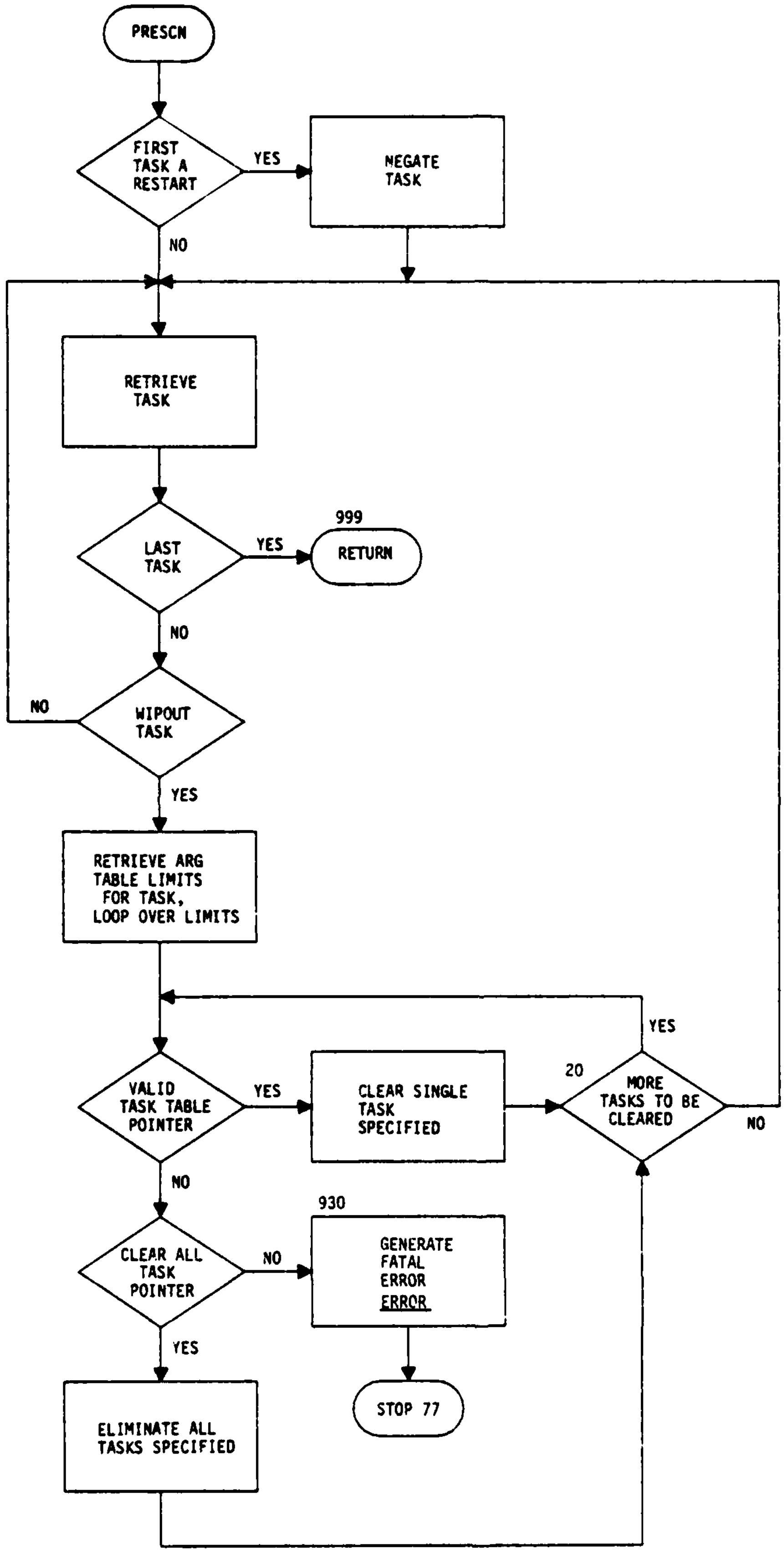
ERROR

STATIN

STATOT

WLKBCK

PRESCN (INPUT)



1. NAME: PRTGTD (INPUT)

2. PURPOSE: To print the descriptions of all GTD geometry objects: plates, cylinders, and end caps.

3. METHOD: For each geometry type a header is printed and the geometry data set is searched for all entries of that type. Then the SEGTBL data for an entry are used to calculate those parameters which describe the geometry object:

PLATES	CYLINDERS	END CAPS
Plate number	Cylinder number	End cap number
Coordinates of corners	End cap numbers	Cylinder number
	Cylinder length	Normal direction
	x axis radius	
	y axis radius	
	Coordinates of center	
	Rotation angles of axis	

Note that it is assumed that plate data appear in contiguous entries in SEGTBL. The GEMACS user may examine this printout to ensure the correctness of his input and geometry processing.

4. INTERNAL VARIABLES:

VARIABLE	DESCRIPTION
D	Dummy variable
F1,F2,F3	Arguments for GETPNT
ICYL	Cylinder number of end cap entry
ICYLPT	PNTTBL entry for cylinder
IECPT	PNTTBL entry for end cap
IEC1	User-assigned number of top end cap
IEC2	User-assigned number of bottom end cap
IG	SEGTBL entry number
IG1	SEGTBL entry number if multiple plate entry crosses SEGTBL page boundary

IGBLK	Geometry block number
IGBLK1	Geometry block number if multiple plate entry crosses SEGTBL page boundary
IGLIM	Last SEGTBL entry containing a GTD geometry
IGLIM1	Last SEGTBL entry containing a GTD geometry if multiple plate entry crosses SEGTBL page boundary
IGLOW	First SEGTBL entry containing a GTD geometry
IGLOW1	First SEGTBL entry containing a GTD geometry if multiple plate entry crosses SEGTBL page boundary
INEED	Number of lines required for page header
INPBLK	First block of geometry containing a GTD geometry
ITAG	Geometry entry tag number
ITP	Index over geometry types
ITYPE	Tag number of geometry type being tabulated
I1,I2,I3	Arguments for GETPNT (equivalent to F1,F2,F3)
LNLEFT	Number of lines left on page
LNPAGE	Number of printable lines per page
MMAX	Number of coordinates in this plate SEGTBL entry
N	Plate SEGTBL entry being printed
NCORN	Number of corners on GTD plate
NEC	User-assigned end cap number
NTRY	Number of SEGTBL entries for a plate geometry
NTYPE	Number of plates, cylinders, or end caps already printed

NUMCY	User-assigned cylinder number
NUMPL	User-assigned plate number
THETAD,PHID	Angles describing rotation of cylinder axis (cylinder) or orientation of end cap normal (end cap) in degrees

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DGTORD	/GEODAT/
ICYTAG	/GTDDAT/
IECTAG	/GTDDAT/
IP217	/GEODAT/
IPLTAG	/GTDDAT/
ISGTBL	/SEGMNT/
ITAGID	/GTDDAT/
LUPRNT	/ADEBUG/
MAXBLK	/SEGMNT/
MAXSEG	/SEGMNT/
NDXBLK	/SEGMNT/
NPATCH	/SEGMNT/
NTPGTD	/GTDDAT/
NUMCYL	/GTDDAT/
NUMECP	/GTDDAT/
NUMPLT	/GTDDAT/
NUMSEG	/SEGMNT/
NWIRE	/SEGMNT/
SEGTBL	/SEGMNT/

B. OUTPUT LOCATION

NONE

6. CALLING ROUTINE:

GEODRV

7. CALLED ROUTINES:

ASSIGN

GETPNT

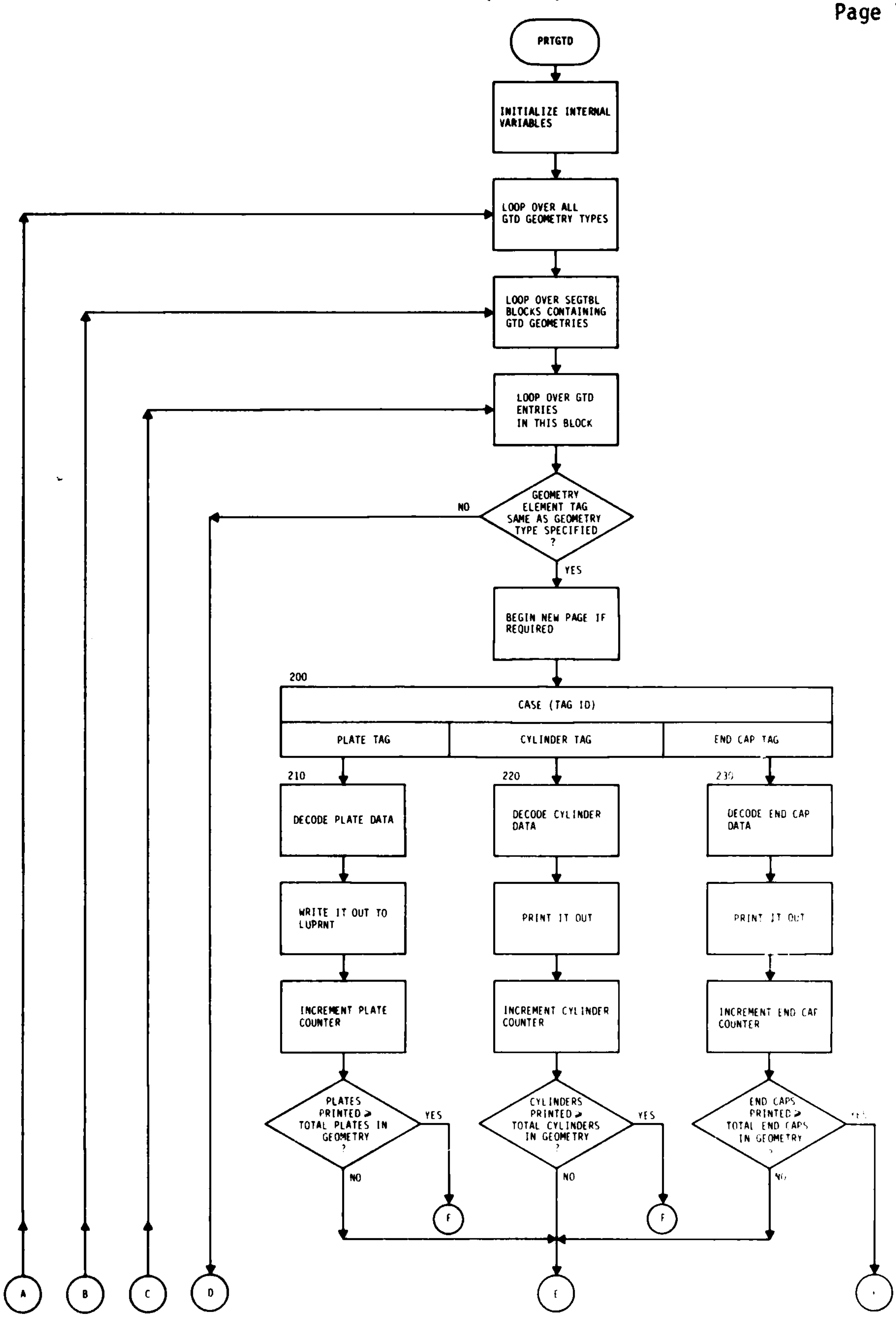
GETSEG

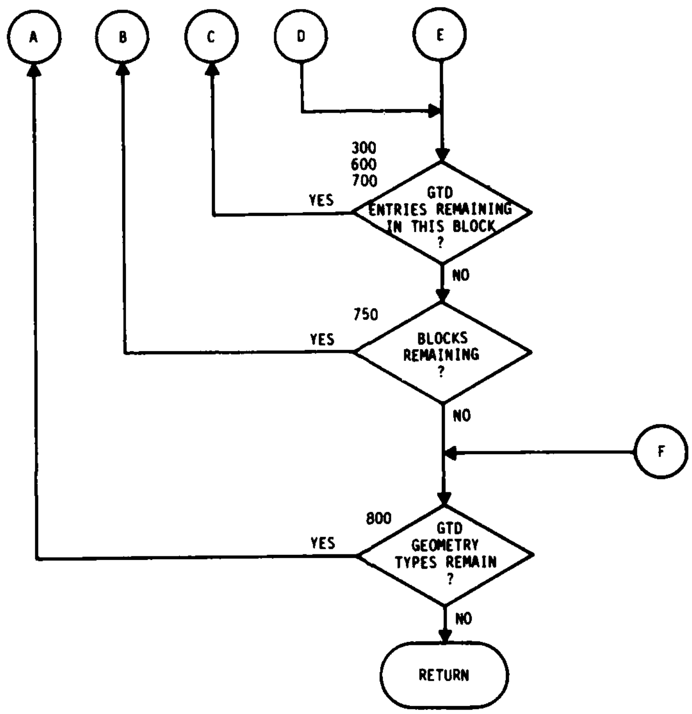
STATIN

STATOT

WLKBCK

PRTGTD (INPUT)




A
B
C
D
E
300
600
700
YES
GTD ENTRIES REMAINING IN THIS BLOCK ?
NO
750
YES
BLOCKS REMAINING ?
NO
F
800
YES
GTD GEOMETRY TYPES REMAIN ?
NO
RETURN

1. NAME: PRTKJ (GTD, MOM)

2. PURPOSE: To print a list of the (K,J) interactions set by the SETINT command.

3. METHOD: The interaction array is searched for (K,J) values contained in ISETTB. When a match is found, the keyword associated with the interaction is placed in a list. When the search is complete the list is printed.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Loop index
II	Pointer to end of keyword name list
IKW	Keyword number
ISET	Row of ISETTB being examined
(K,J)	GTD or MOM interaction indices
KLST	List of keyword names
KN	NCODES pointer for keyword
NAME	Keyword name in NCODES

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
IP217	/GEODAT/
ISETTB	/INTMAT/
KJINT	/INTMAT/
KWNAME	/PARTAB/
LUPRNT	/ADEBUG/
NCODES	/PARTAB/

B. OUTPUT

NONE

6. CALLING ROUTINES:*

   EXCDRV (2,3)

   FLDDRV (2)

   ZIJDRV (2,3)

7. CALLED ROUTINES:

   ASSIGN

   CONVRT

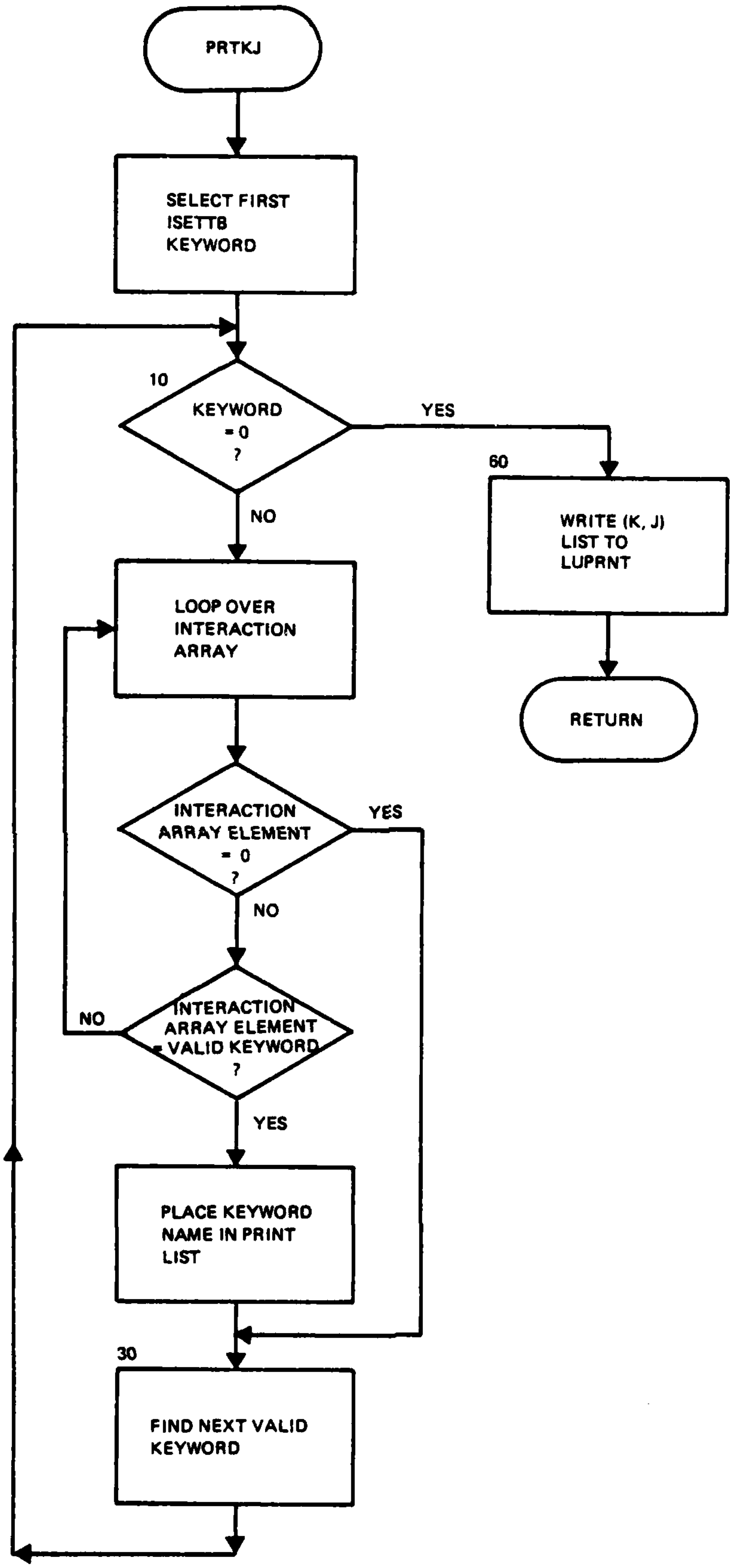
   STATIN

   STATOT

   WLKBCK

*2-GTD
3-MOM

PRTKJ (GTD, MOM)



1. NAME: PRTSYM (MOM)

2. PURPOSE: Subroutine to execute the PRINT and WRITE commands.

3. METHOD: The task number is recovered and the input arguments extracted from the argument list. If a WRITE task, the row and column indices are recovered from the argument list. If a PRINT task, the row and column indices are recovered from NDATBL. The data are retrieved into the TEMP array and written to either the LUPRNT logical unit or the logical unit specified on the WRITE command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ANGLE	Phase for complex data
COLHDR	Column header
CPART	Complex part of data
HEADR	Print variable
IALPHA	Flag indicating an alpha text symbol
IBAND	Flag indicating a banded matrix
IBW	Interval variable for bit set attributes
ICALL	Type of printout required of FABL04
ICMPLX	Variable indicating complex data
IJMOD	Modulus function
ILWR	Variable indicating a lower triangular matrix
INTGER	Variable indicating integer data set
ISYMTY	Variable indicating symmetrical data set
ITRANP	Variable indicating transposed data set
ITRBUT	Bit set attribute word
ITYPE	Variable indicating type of data set

IUPR	Variable indicating upper triangular matrix
LOC1	First cell of TEMP for internal storage
LSTREC	Last record required
LUFILE	Logical file number of the data set
NAMSYM	Name of the data set
NC1	First element index
NC1M1	NC1 minus one
NC2	Last element index
NDXARG	Index to argument list
NDXPNT	Index to parent data set
NDXSYM	Index to symbol being retrieved
NPRELM	Number of words per data set element
NPRLIN	Number of data set words per line
NPRPRT	Number of words per output
NPRREC	Number of words per data set record
NP1	Loop control
NP2	Loop control
NR1	First record index
NR2	Last record index
NUMELM	Number of elements per record
NUMREC	Number of records per data set
RDTODG	Conversion from radians to degrees factor
ROWHDR	Row header data
RPART	Real part of complex data
SQRTRC	Magniturde of complex data

WRITE	Logical .TRUE. for WRITE command
WRTTSK	Task number for WRITE command

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
INTARG	/ARGCOM/
IPASS	/ARGCOM/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
KBBAND	/PARTAB/
KBCPLX	/PARTAB/
KBGEOM	/PARTAB/
KBLWRT	/PARTAB/
KBORDR	/PARTAB/
KBREAL	/PARTAB/
KBSOLN	/PARTAB/
KBSYM	/PARTAB/
KBTEXT	/PARTAB/
KBUPRT	/PARTAB/
KOLBIT	/PARTAB/
KOLCOL	/PARTAB/
KOLLNK	/PARTAB/
KOLNAM	/PARTAB/
KOLROW	/PARTAB/
LSTARG	/ARGCOM/
LSTIOD	/ADEBUG/

LUPRNT	/ADEBUG/
NDATBL	/PARTAB/
NOPCOD	/ADEBUG/
NTALPH	/ADEBUG/
NUMARG	/ARGCOM/
NUMWRD	/ADEBUG/
ZERO	/ADEBUG/

B. OUTPUT

NONE

6. CALLING ROUTINE:

TSKXQT

7. CALLED ROUTINES:

ASSIGN

CONVRT

FABLO4

GETARG

GETSYM

IBITCK

OPNFIL

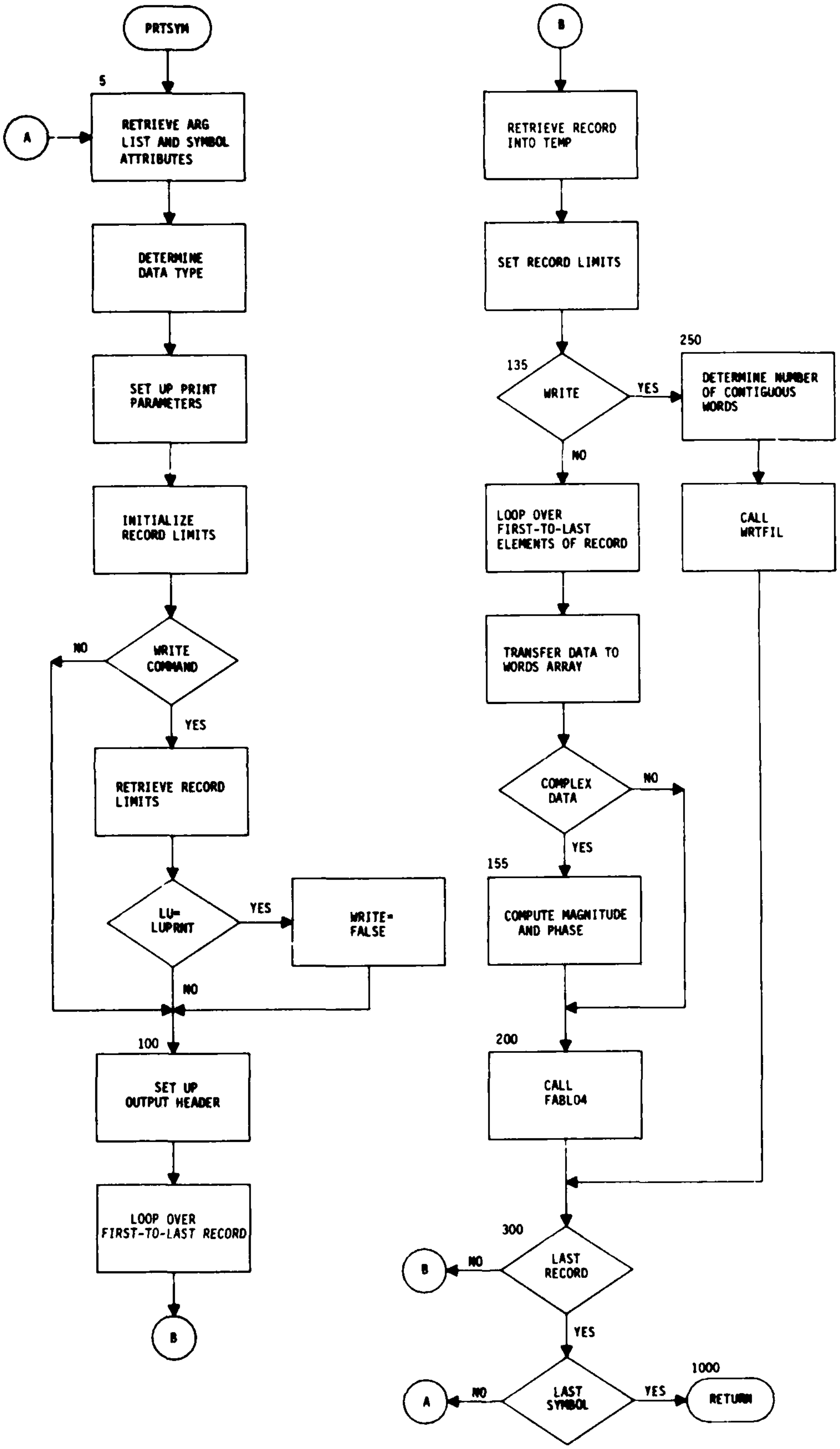
STATIN

STATOT

WLKBCK

WRTFIL

PRTSYM (MOM)


PRTSYM
5
A
RETRIEVE ARG LIST AND SYMBOL ATTRIBUTES
DETERMINE DATA TYPE
SET UP PRINT PARAMETERS
INITIALIZE RECORD LIMITS
WRITE COMMAND
NO
YES
RETRIEVE RECORD LIMITS
LU= LUPRNT
YES
WRITE= FALSE
NO
100
SET UP OUTPUT HEADER
LOOP OVER FIRST-TO-LAST RECORD
B
B
RETRIEVE RECORD INTO TEMP
SET RECORD LIMITS
135
WRITE
YES
250
DETERMINE NUMBER OF CONTIGUOUS WORDS
CALL WRTFIL
NO
LOOP OVER FIRST-TO-LAST ELEMENTS OF RECORD
TRANSFER DATA TO WORDS ARRAY
COMPLEX DATA
NO
YES
155
COMPUTE MAGNITUDE AND PHASE
200
CALL FABL04
300
LAST RECORD
NO
B
YES
LAST SYMBOL
NO
A
YES
1000
RETURN

1. NAME: PUTKWV (GTD, INPUT, MOM, OUTPUT)

2. PURPOSE: Put value in common keyword variable.

3. METHOD: The calling routine will pass through a value to be put in the keyword variable.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
KW	Keyword number
NDX	Keyword index to NCODES array
NDXKWD	Keyword number
VALUKW	Keyword value to be put in keyword variable

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
CLITE	/AMPZIJ/
ISON	/ADEBUG/
KWCOND	/PARTAB/
KWEPSR	/PARTAB/
KWFRQ	/PARTAB/
KWNAME	/PARTAB/
KWNMFL	/PARTAB/
KWTIME	/PARTAB/
LUPRNT	/ADEBUG/
NCODES	/PARTAB/
NDXKWD	F.P.
TWOPI	/AMPZIJ/
VALUKW	F.P.
ZERO	/ADEBUG/

B. OUTPUT — LOCATION

OUTPUT	LOCATION
EPSR	/AMPZIJ/
FRQMHZ	/AMPZIJ/
IERRF	/ADEBUG/
IPERF	/AMPZIJ/
KSYMP	/AMPZIJ/
NFILES	/IOFLES/
SIGMA	/AMPZIJ/
TIMTGO	/SYSFIL/
WAVLGH	/AMPZIJ/
WAVNUM	/AMPZIJ/

6. CALLING ROUTINE:

DMPDRV

7. CALLED ROUTINES:

ASSIGN

CONVRT

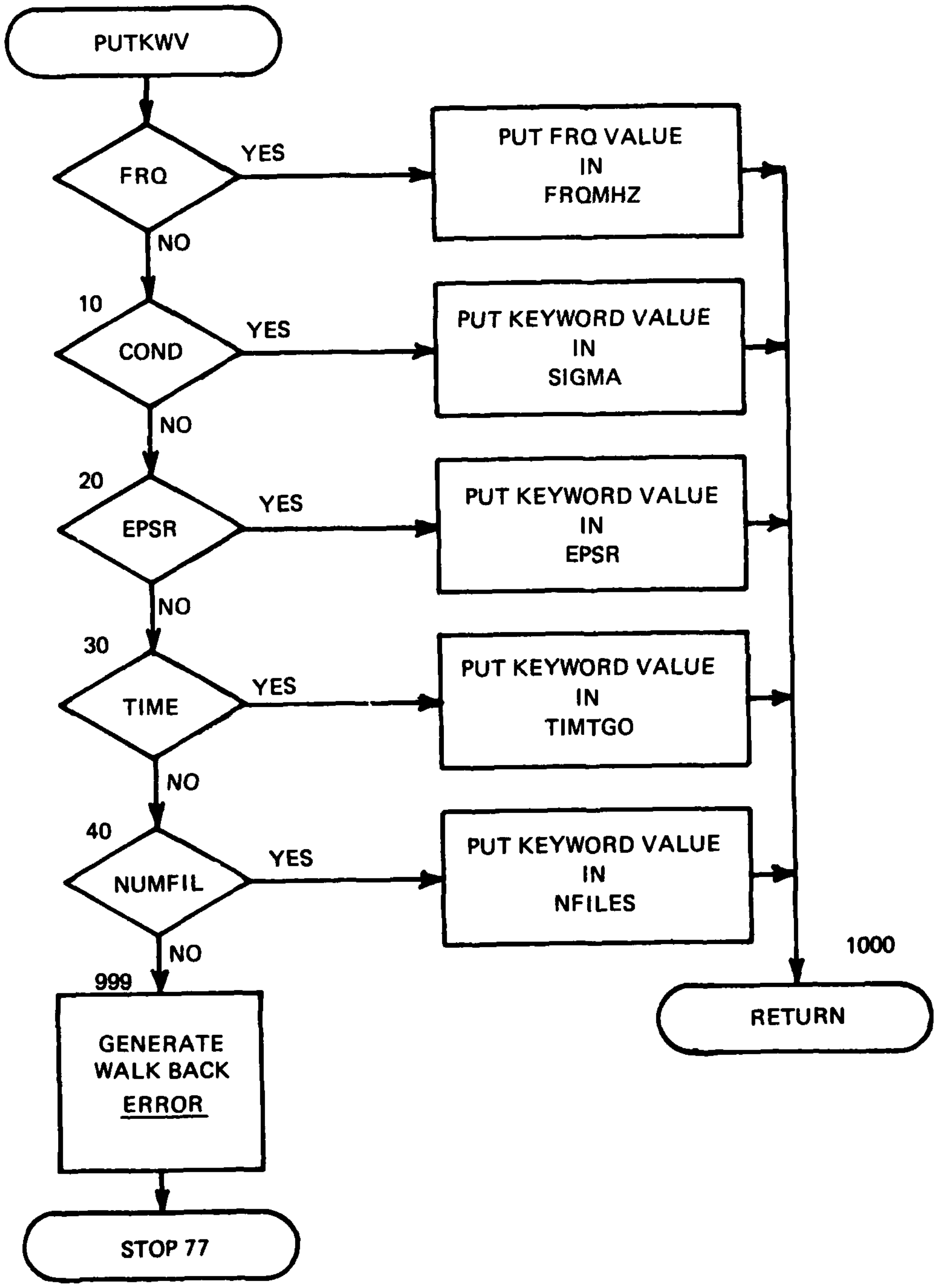
ERROR

STATIN

STATOT

WLKBCK

PUTKWV (GTD, INPUT, MOM, OUTPUT)



1. NAME: PUTPNT (INPUT)

2. PURPOSE: To retrieve and store point data for the geometry processor.

3. METHOD: If I is positive, the input arguments are stored in array PTTBLE. If I is negative, a search is made to find the original data. New data are stored if the point is found. If not found, an error message is printed.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Point number, if negative, point is to be retrieved, if positive, point is to be stored.
IPT	Negated point number
XP	X coordinate of point
YP	Y coordinate of point
ZP	Z coordinate of point

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
I	F.P.
ISON	/ADEBUG/
LUPRNT	/ADEBUG/
MAXPTS	/PNTTBL/
SCALE	/SEGMNT/
XP	F.P.
YP	F.P.
ZP	F.P.

B. OUTPUT

OUTPUT	LOCATION
I	F.P.

IERRF	/ADEBUG/
IPTS	/PNTTBL/
IPTTBL	/PNTTBL/
NUMPTS	/PNTTBL/
PTTBLE	/PNTTBL/

6. CALLING ROUTINE:

   WYRDRV

7. CALLED ROUTINES:

   ASSIGN

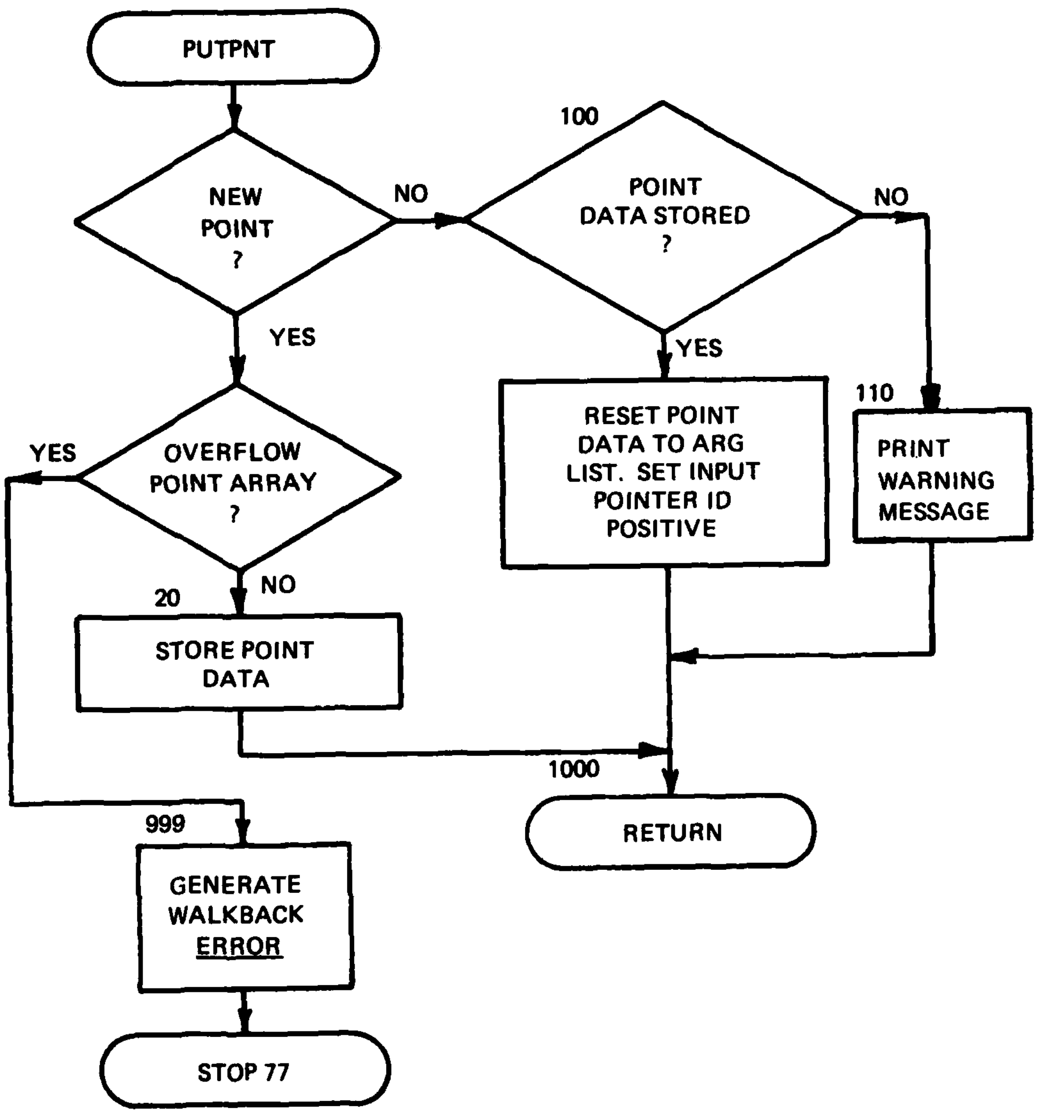
   ERROR

   STATIN

   STATOT

   WLKBCK

PUTPNT (INPUT)



1. NAME: PUTSEG (INPUT)

2. PURPOSE: To retrieve and enter data into the SEGTBL array for the geometry processor.

3. METHOD: The argument list parameters are either stored in the SEGTBL array or retrieved from the SEGTBL array depending upon the value for the ITAG and the IS parameters.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICASE	Flag indicating the various cases: for retrieval by tag number ICASE equals 1; for retrieval by segment number ICASE equals 2; to replace data for a segment already stored, ICASE equals 3
IS	Input segment number
ISEG	Pointer to array SEGTBL
ISG	Internal variable for segment number
ITAG	Tag number of segment
ITG	Type of segment
JSG	Value of segment number for which data are to be replaced
JTG	Negated tag number
LIMSEG	Number of segments in current block of data
NDXBLK	Index to data block currently in use
NTRBLK	NDXBLK-1 or MAXBLK
NUMBLK	Number of geometry data blocks
R	Segment radius or patch area
X1	X coordinate of end 1 of wire
X2	X coordinate of end 2 of wire
X3	Ninth SEGTBL entry

Y1	Y coordinate of end 1 of wire
Y2	Y coordinate of end 2 of wire
Y3	Tenth SEGTBL entry
Z1	Z coordinate of end 1 of wire
Z2	Z coordinate of end 2 of wire
Z3	Eleventh SEGTBL entry

5. I/O VARIABLES:

A. INPUT

INPUT	LOCATION
DBGPRT	/ADEBUG/
IP217	/GEODAT/
IPLTAG	/GTDDAT/
IS	F.P.
ISOFF	/ADEBUG/
ITAG	F.P.
KOLCOL	/PARTAB/
LUPRNT	/ADEBUG/
MAXBLK	/SEGMNT/
MAXSEG	/SEGMNT/
NAMSEG	/SEGMNT/
NDXBLK	/SEGMNT/
NPRSEG	/SEGMNT/
NUMSEG	/SEGMNT/
R	F.P.
SCALE	/SEGMNT/
SEGTBL	/SEGMNT/

PUTSEG (INPUT)

X1	F.P.
X2	F.P.
X3	F.P.
Y1	F.P.
Y2	F.P.
Y3	F.P.
Z1	F.P.
Z2	F.P.
Z3	F.P.

B. OUTPUT — LOCATION

OUTPUT	LOCATION
IS	F.P.
ISEG	/SEGMNT/
ISGTBL	/SEGMNT/
ITAG	F.P.
NDXBLK	/SEGMNT/
NPATCH	/SEGMNT/
NUMCYL	/GTDDAT/
NUMECP	/GTDDAT/
NUMGTD	/GTDDAT/
NUMPLT	/GTDDAT/
NUMSEG	/SEGMNT/
NWIRE	/SEGMNT/
R	F.P.
SEGTBL	/SEGMNT/
UPDBLK	/SEGMNT/

X1	F.P.
X2	F.P.
X3	F.P.
Y1	F.P.
Y2	F.P.
Y3	F.P.
Z1	F.P.
Z2	F.P.
Z3	F.P.

6. CALLING ROUTINES:

CYLNDR

ENDCAP

PATCH

PLATE

WYRDRV

7. CALLED ROUTINES:

ASSIGN

GETSEG

GETSYM

PUTSYM

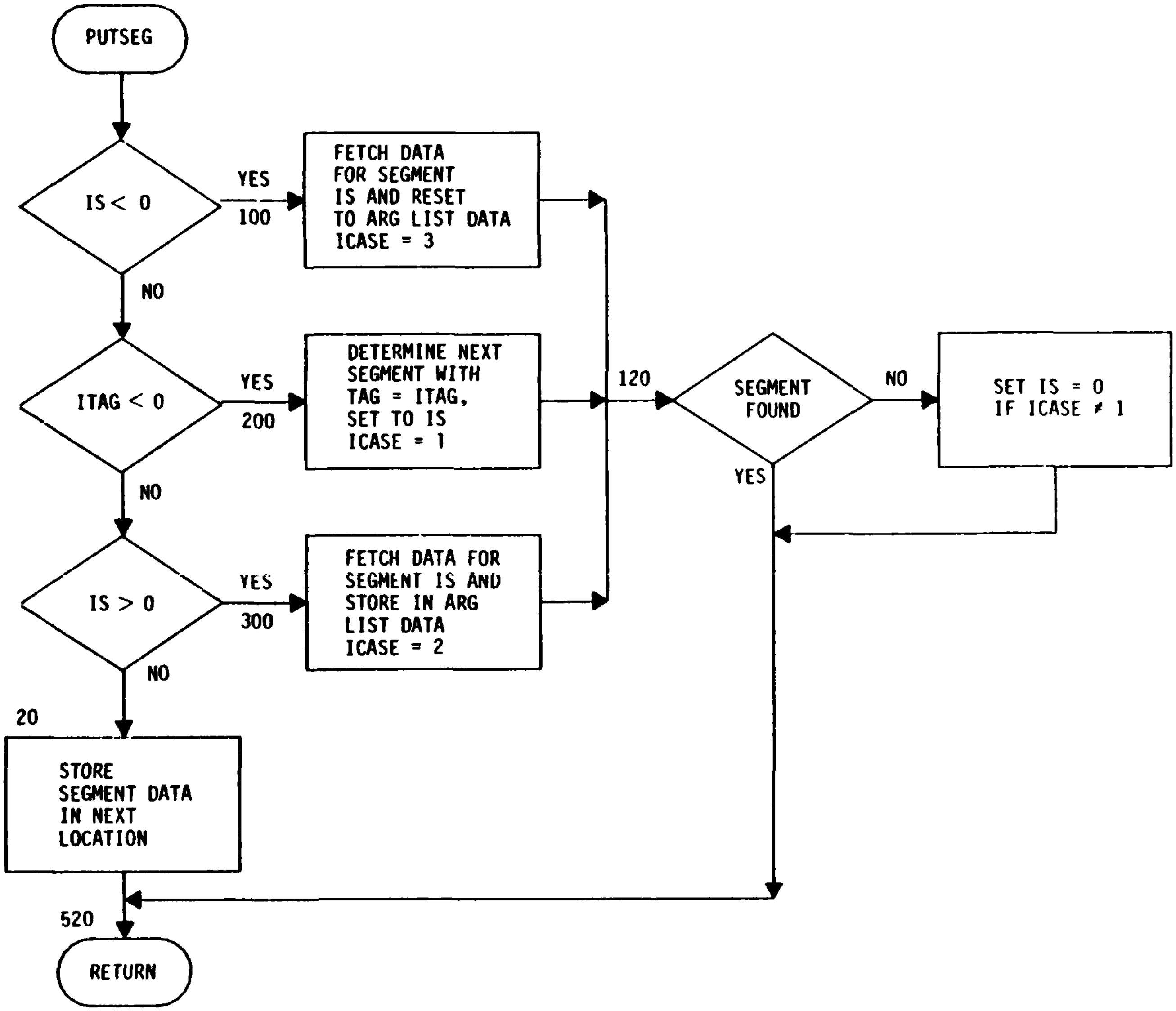
STATIN

STATOT

SYMUPD

WLKBCK

PUTSEG (INPUT)



1. NAME: PUTSYM (GTD, INPUT, MOM, OUTPUT)

2. PURPOSE: To transfer in-core data to a file or an array associated with a specified symbolic name.

3. METHOD: Calculate the record length and symbol attributes. Determine the file number and issue calls to WRTFIL to store the data in the records specified. If the data to be stored replace an existing record, the process occurs in steps:

   (a) All records before the first replacement record are transfered from the symbol data file to a scratch file.

   (b) The replacement records are written onto the scratch file.

   (c) All records after the last replacement record are transferred to the scratch file.

   (d) The unit numbers of the scratch file and symbol file are switched instead of copying the scratch file back into the symbol file.

Any GEMACS file may contain more than one edition of a data set. For example, when a data set is generated in a loop, there will be as many editions of the data set as there are passes through the loop. All editions are contained sequentially on the same file. The format of a multiple edition data set is shown in figure 1. LOCFST, the global location of the first word of the edition being accessed (the "present" edition), is stored in NDATBL column KOLFST.

To access the correct records of the present edition, PUTSYM calls FNDREC, which returns the number of the first global record of the present edition. This value is used as an offset so that global record numbers may be computed from edition record numbers. For example, suppose for the file in figure 1, the present edition is edition #3 (LOCFST = 79) and PUTSYM was called to store record numbers 5 through 8 of the edition. FNDREC returns the global record number for LOCFST as 19. Hence the global record numbers would be

IREC1 = 5 + 19 - 1 = 23
IREC2 = 8 + 19 - 1 = 26

The basic process of file storage is shown in figure 2. If a record is not to be replaced, the record is read from IF1 into ITEMP, then written to IF2. If the record is to be replaced, IF1 is read into ITEMP to advance IF1, but the data for IF2 are taken from TMPBUF. TMPBUF may point to the same storage location as TEMP, since the contents of TEMP are saved prior to file I/O and restored prior to exiting PUTSYM.

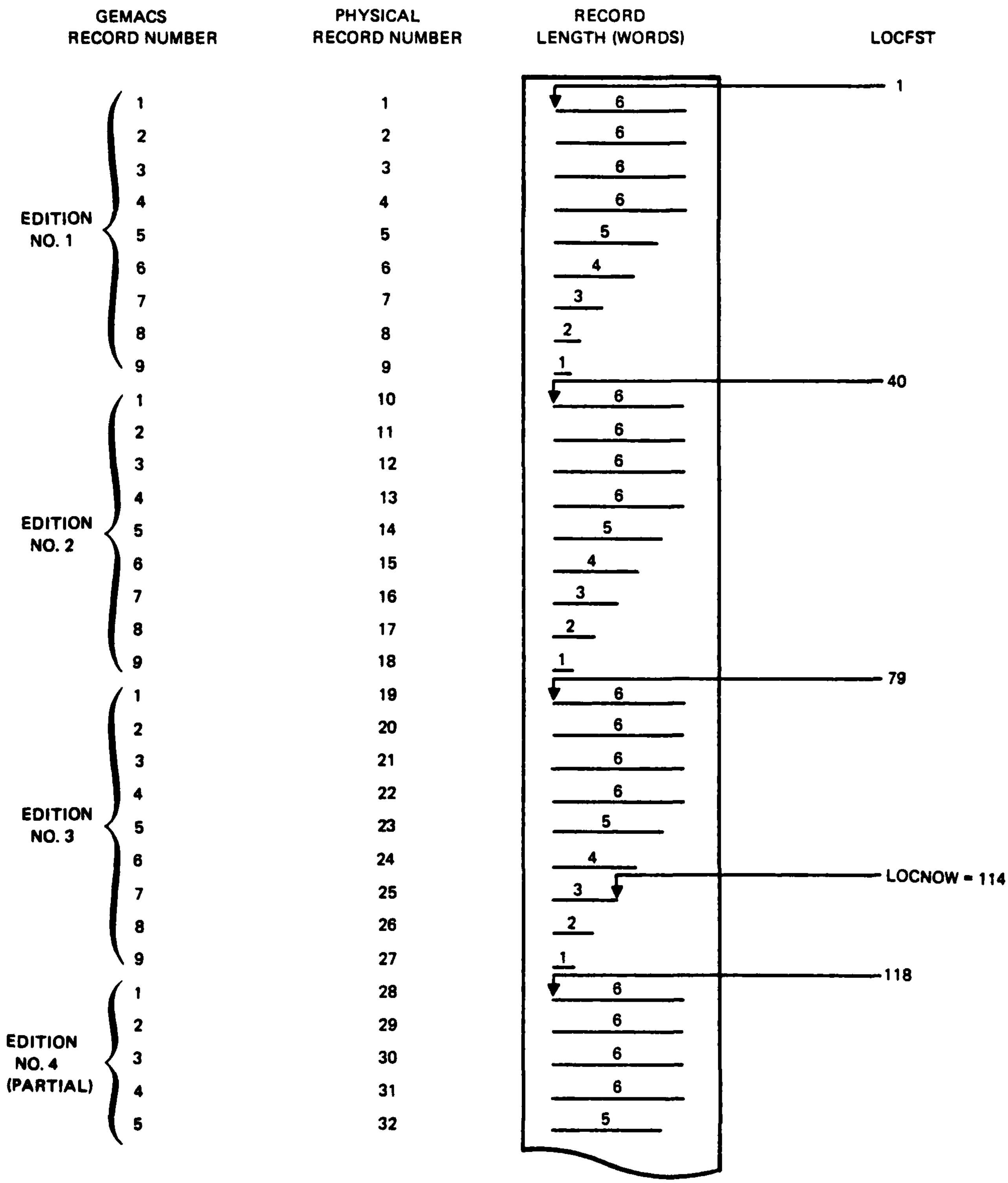


Figure 1. A Multiple Edition Data Set with Three Complete and One Partial Editions. (The matrix stored is a banded, upper triangular, real matrix, derived from a 9 x 9 real matrix.)

PUTSYM (GTD, INPUT, MOM, OUTPUT)

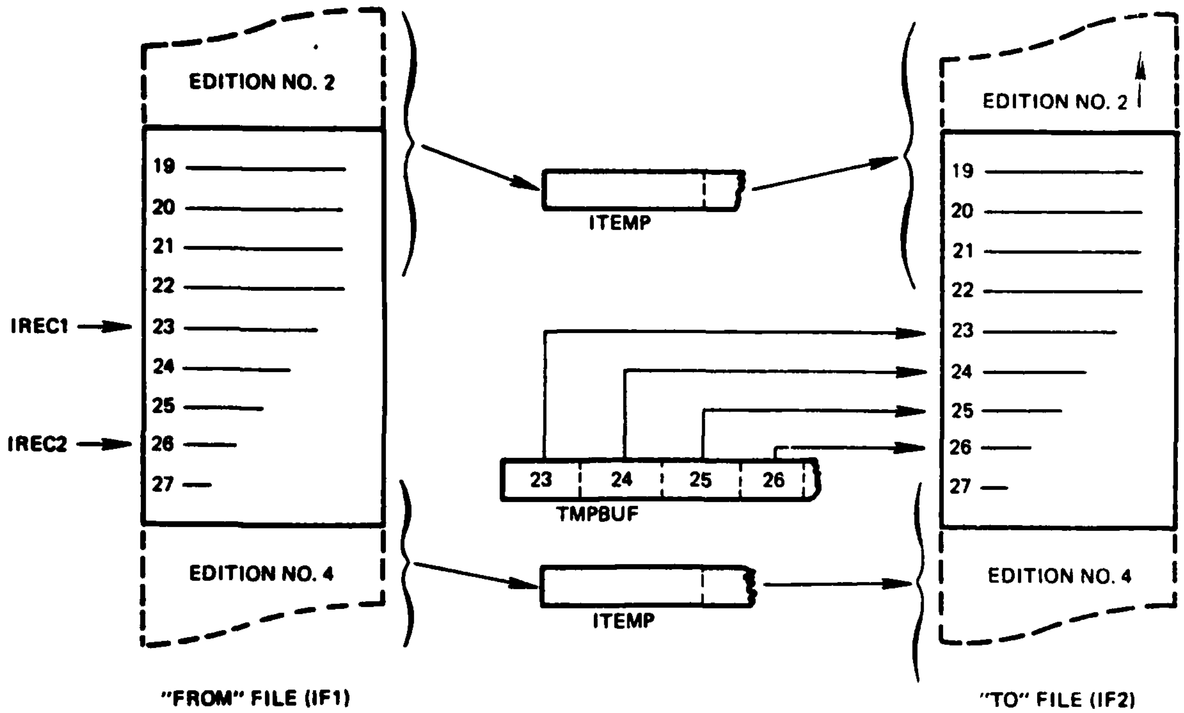


Figure 2. The Basic Data Transfer Process of PUTSYM

PUTSYM is written to take advantage of four special cases: (1) Adding to the end of the file; (2) Replacing the entire file; (3) Writing the entire file to the checkpoint file; and (4) Retrieving the entire file from the checkpoint tape. The file names and record limits for the special cases are shown in figure 3.

	FROM FILE	TO FILE	FILE TRANSFER LIMITS		REPLACEMENT RECORD LIMITS	
SPECIAL CASE	IF1	IF2	N1	N2	IREC1	IREC2
1. ADD TO END OF FILE	0	IFILE	IREC1	IREC2	IREC1	IREC2
2. REPLACE ENTIRE FILE	0	IFILE	IREC1	IREC2	IREC1	IREC2
3. CHECKPOINT	IFILE	(IOCKPT, MODCHK*)	1	IRECND	1	0
4. RESTART	(IOCKPT, MODCHK*)	IFILE	1	IRECND	1	0
5. GENERAL PUTSYM COMMAND	IFILE	IOSCRT	1	IRECND	IREC1	IREC2

(*MODCHK is used only for the end-of-module checkpoint.)

Figure 3. File IDs and Record Limits for Special Cases of PUTSYM

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IBAND	Flag indicating a banded data set
IFILE	The logical unit designation for the symbol
IF1	"From" file number
IF2	"To" file number
ILOWER	Flag indicating a lower triangular matrix
IORDER	Flag indicating a transposed matrix
IR	Record number used to restore file to its position before checkpoint
IRC1	First local record of MATNAM
IRC2	Last local record of MATNAM
IRECND	Record number of last word on file
IRECNW	Record number of present file position
IRECFS	Record number of beginning of this edition
IREC1	Global record number corresponding to IRC1
IREC2	Global record number corresponding to IRC2
IROWM1	First record minus one to be retrieved on file
IR1	Internal variable equal to IRC1
IR2	Internal variable equal to IRC2
ISV	Pointer to symbol name in symbol table
ISW	Flag indicating that TEMP array must be saved
IT	Pointer to TMPBUF array for in-core storage transfer
IUPPER	Flag indicating an upper triangular matrix

IWRD1	First word address for present edition of data set
KBAND	Flag indicating a banded parent data set
KLINK	Pointer to data set linked to parent
LINK	Pointer to parent data set in symbol table
LOCEND	Location of the end of the data file
LOCFST	Location of first word for this edition of symbol
LOCLST	Location of last word for this edition of symbol
LOCNOW	Current location of logical unit containing data set
LOCSTR	Location in the TMPBUF array
LSTWRD	Last word of data set to be retrieved
LWRUPR	Flag indicating a triangular matrix
MATNAM	Input argument designating symbol name to be retrieved
MAXWRD	Total number of words to be retrieved
MORE	Flag set if specified attribute is present in symbol
MOVWRD	Number of words that data file is moved
NO	Dummy variable
N1	Lower transfer record limit
N2	Upper transfer record limit
NA	Alpha format of symbol name
NAMSAV	Saved name of MATNAM for future calls to PUTSYM
NBITWD	The bit set attribute word for the data set

NPRELM	Number of computer words per element of the data set
NPRPRT	Number of rows in parent
NPRREC	Number of rows in this record
NRECS	Number of records in this edition
NUMROW	Number of rows in symbol

5. I/O VARIABLES:

A.

INPUT	LOCATION
DBGPRT	/ADEBUG/
FLTSYM	/SYMSTR/
IMDCHK	/ADEBUG/
IOCKPT	/SYSFIL/
IOFILE	/FILES/
IRC1	F.P.
IRC2	F.P.
IOSCR1,IOSCR2	/SYSFIL/
IRSTRT	/ADEBUG/
ISOFF	/ADEBUG/
ISON	/ADEBUG/
IWRTCK	/ADEBUG/
KBBAND	/PARTAB/
KBCPLX	/PARTAB/
KBDPRE	/PARTAB/
KBLWRT	/PARTAB/
KBORDR	/PARTAB/

PUTSYM (GTD, INPUT, MOM, OUTPUT)

KBUPRT	/PARTAB/
KOLAST	/PARTAB/
KOLBIT	/PARTAB/
KOLFST	/PARTAB/
KOLLNK	/PARTAB/
KOLLOC	/PARTAB/
KOLNAM	/PARTAB/
KOLROW	/PARTAB/
LUPRNT	/ADEBUG/
MATNAM	F.P.
MAXSTR	/SYMSTR/
MODCHK	/SYSFIL/
NDATBL	/PARTAB/
NDFILE	/IOFLES/
NPDATA	/PARTAB/
NTEMPS	/TEMP01/
TEMP	/TEMP01/
TMPBUF	F.P.

B. OUTPUT LOCATION

IERRF	/ADEBUG/
IOSCR1	/SYSFIL/
IOSCR2	/SYSFIL/

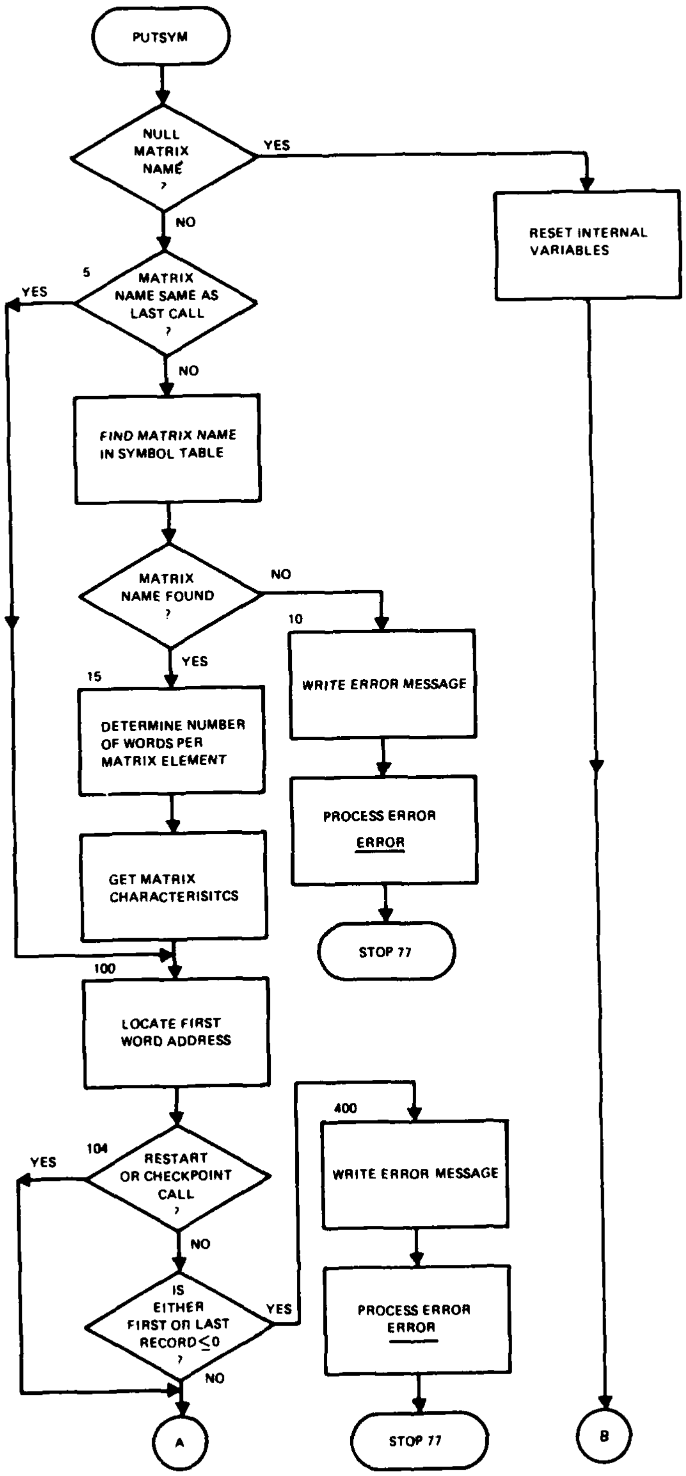
6. CALLING ROUTINES:*

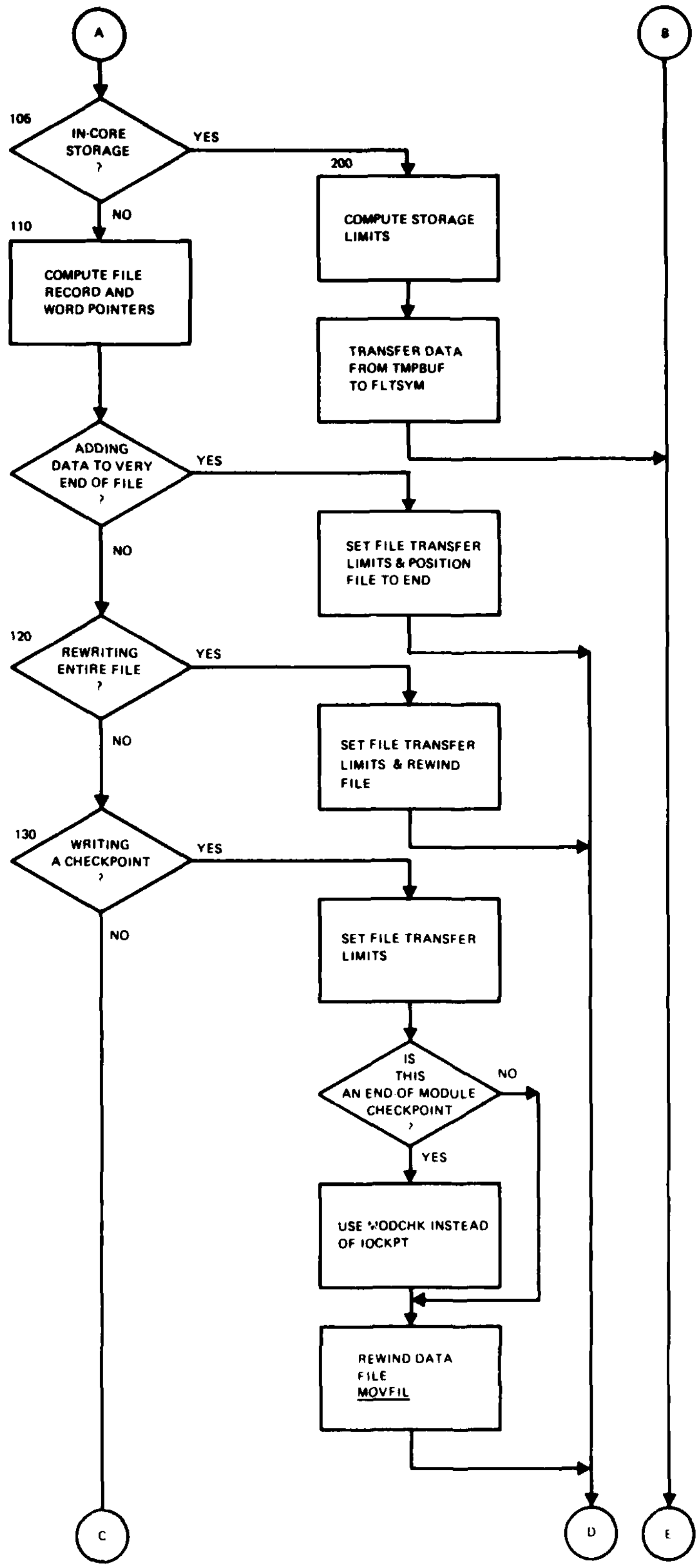
BANDIT (3)	LODDRV (3)	SOLDRV (3)
DMPDRV (1,2,3,4)	LUDDRV (3)	STRTUP (2,3,4)
EGFMAT (3)	PUTSEG (1,2,3)	SUBPAT (1)
EXCDRV (2,3)	REBLCK (3)	WRTCHK (1,2,3,4)
FLDDRV (2,3,4)	RESTRT (1)	ZIJDRV (2,3)
GEODRV (1)	RWFILS (1,2,3,4)	
GETSEG (1,2,3)	SETDRV (3)	

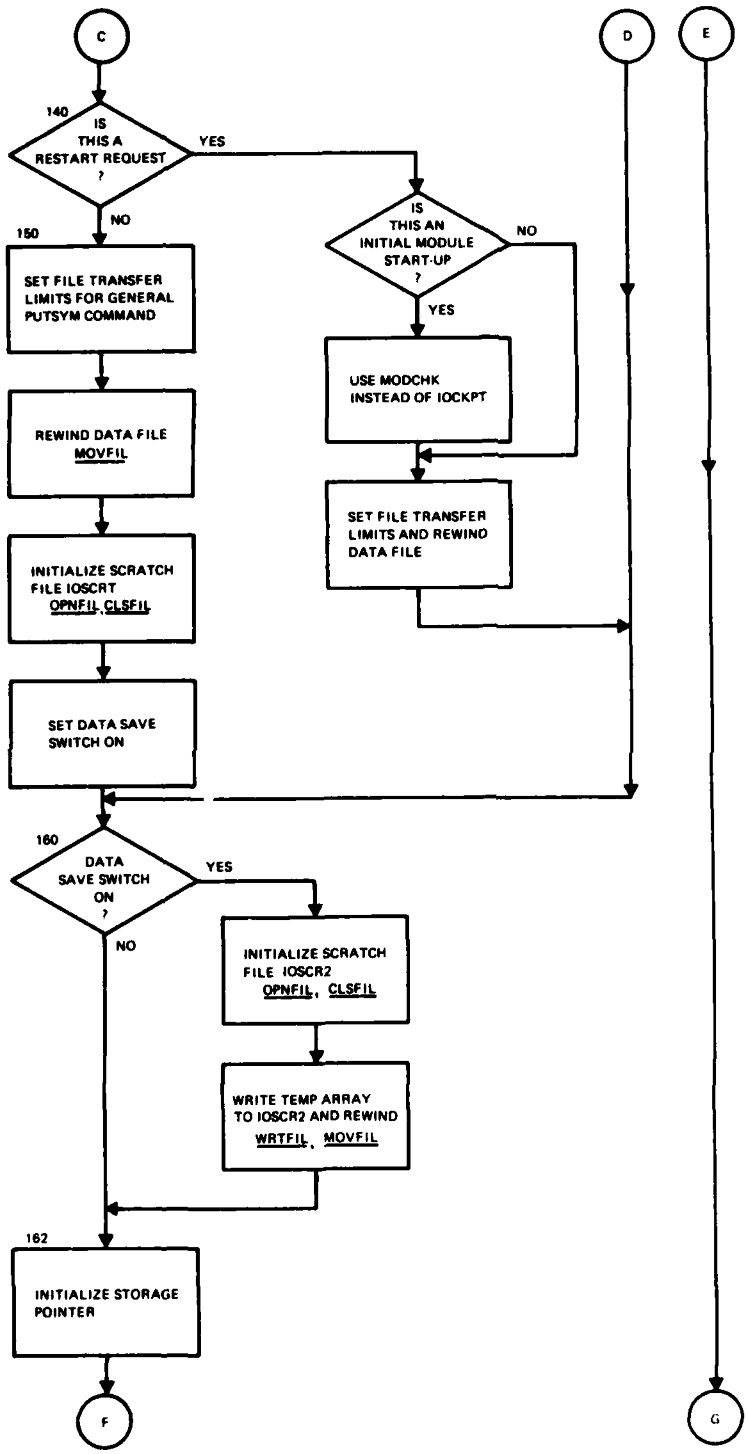
7. CALLED ROUTINES:

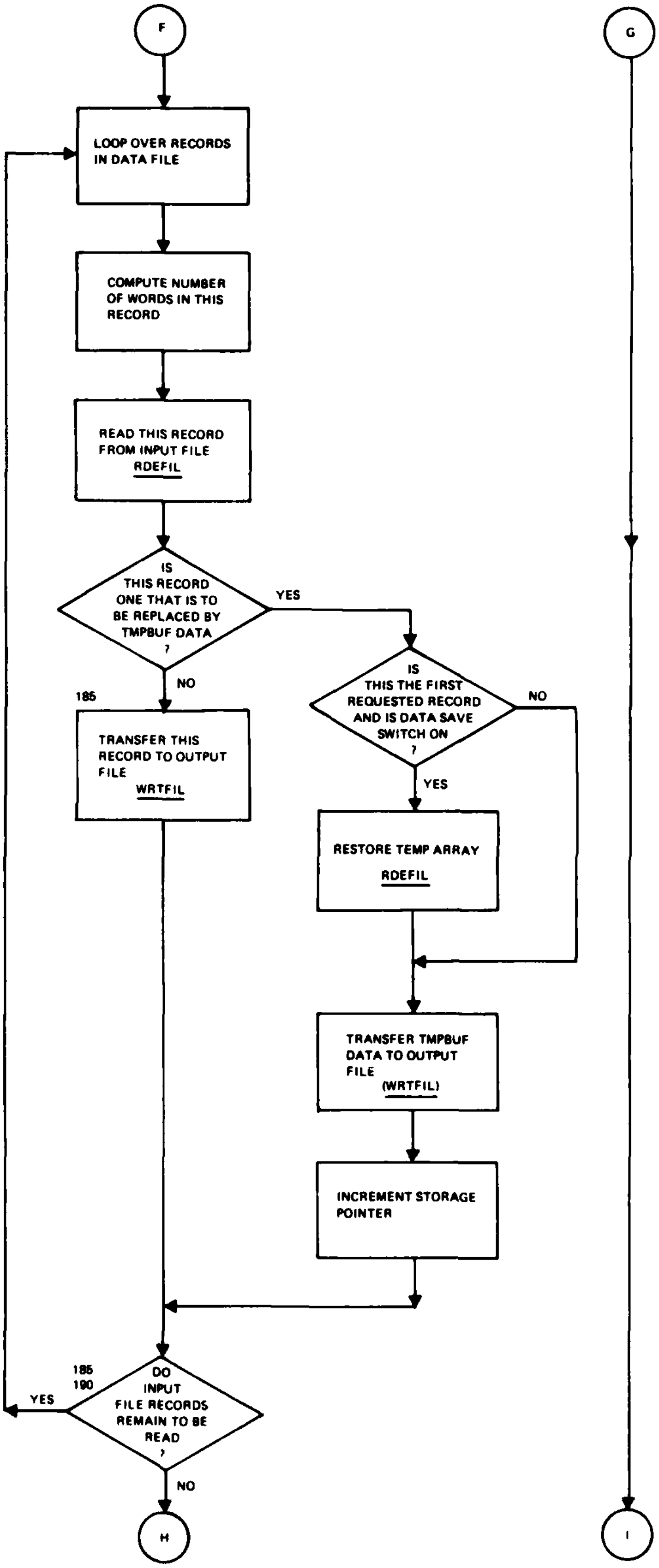
ASSIGN	IBITCK	STATOT
CLSFIL	IJMOD (INTRINSIC)	SYMDEF
CONVRT	MOVFIL	WLKBCK
ERROR	OPNFIL	WRTFIL
FNDREC	RDEFIL	
GETSYM	STATIN	

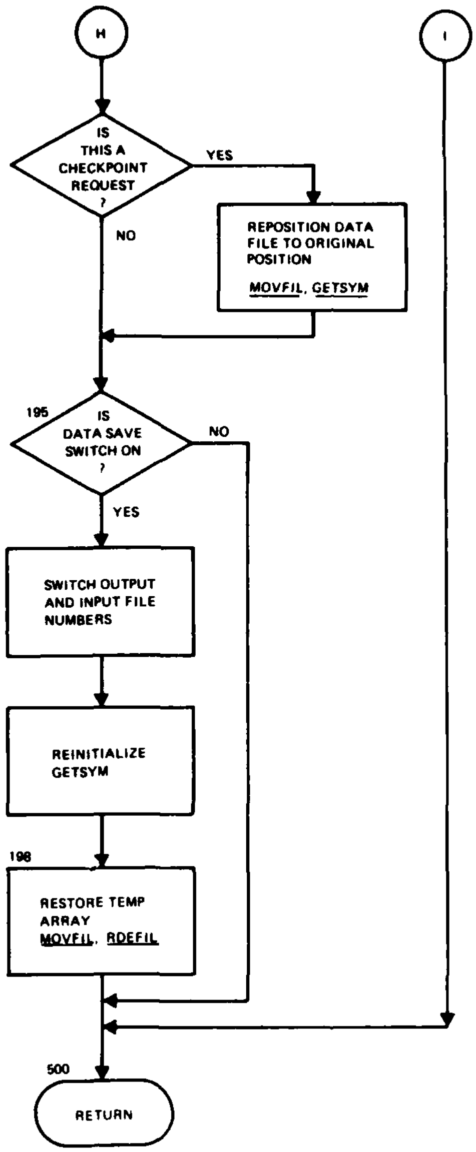
*1 - INPUT
2 - GTD
3 - MOM
4 - OUTPUT


PUTSYM
NULL MATRIX NAME ?
YES
NO
RESET INTERNAL VARIABLES
5
MATRIX NAME SAME AS LAST CALL ?
YES
NO
FIND MATRIX NAME IN SYMBOL TABLE
MATRIX NAME FOUND ?
NO
YES
10
WRITE ERROR MESSAGE
PROCESS ERROR
ERROR
STOP 77
15
DETERMINE NUMBER OF WORDS PER MATRIX ELEMENT
GET MATRIX CHARACTERISITCS
100
LOCATE FIRST WORD ADDRESS
104
RESTART OR CHECKPOINT CALL ?
YES
NO
400
WRITE ERROR MESSAGE
IS EITHER FIRST OR LAST RECORD ≤ 0 ?
YES
NO
PROCESS ERROR
ERROR
STOP 77
A
B


A
B
106
IN-CORE STORAGE ?
YES
200
COMPUTE STORAGE LIMITS
NO
110
COMPUTE FILE RECORD AND WORD POINTERS
TRANSFER DATA FROM TMPBUF TO FLTSYM
ADDING DATA TO VERY END OF FILE ?
YES
SET FILE TRANSFER LIMITS & POSITION FILE TO END
NO
120
REWRITING ENTIRE FILE ?
YES
SET FILE TRANSFER LIMITS & REWIND FILE
NO
130
WRITING A CHECKPOINT ?
YES
SET FILE TRANSFER LIMITS
NO
IS THIS AN END OF MODULE CHECKPOINT ?
NO
YES
OF IOCKPT
REWIND DATA FILE
MOVFIL
C
D
E


C
D
E
140
IS THIS A RESTART REQUEST ?
YES
NO
IS THIS AN INITIAL MODULE START-UP ?
NO
YES
150
SET FILE TRANSFER LIMITS FOR GENERAL PUTSYM COMMAND
USE MODCHK INSTEAD OF IOCKPT
REWIND DATA FILE MOVFIL
SET FILE TRANSFER LIMITS AND REWIND DATA FILE
INITIALIZE SCRATCH FILE IOSCRT OPNFIL,CLSFIL
SET DATA SAVE SWITCH ON
160
DATA SAVE SWITCH ON ?
YES
NO
INITIALIZE SCRATCH FILE IOSCR2 OPNFIL, CLSFIL
WRITE TEMP ARRAY TO IOSCR2 AND REWIND WRTFIL, MOVFIL
162
INITIALIZE STORAGE POINTER
F
G


F
G
LOOP OVER RECORDS IN DATA FILE
COMPUTE NUMBER OF WORDS IN THIS RECORD
READ THIS RECORD FROM INPUT FILE RDEFIL
IS THIS RECORD ONE THAT IS TO BE REPLACED BY TMPBUF DATA ?
YES
NO
185
TRANSFER THIS RECORD TO OUTPUT FILE WRTFIL
IS THIS THE FIRST REQUESTED RECORD AND IS DATA SAVE SWITCH ON ?
NO
YES
RESTORE TEMP ARRAY RDEFIL
TRANSFER TMPBUF DATA TO OUTPUT FILE (WRTFIL)
INCREMENT STORAGE POINTER
185
190
YES
DO INPUT FILE RECORDS REMAIN TO BE READ ?
NO
H
I


H
I
IS THIS A CHECKPOINT REQUEST ?
YES
NO
REPOSITION DATA FILE TO ORIGINAL POSITION
MOVFIL, GETSYM
195
IS DATA SAVE SWITCH ON ?
NO
YES
SWITCH OUTPUT AND INPUT FILE NUMBERS
REINITIALIZE GETSYM
198
RESTORE TEMP ARRAY
MOVFIL, RDEFIL
500
RETURN

1. NAME: QFUN (GTD)

2. PURPOSE: To compute the q^* function for the cylinder's acoustically hard diffraction coefficient.

3. METHOD: The q^* function is defined as (see references A and B):

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} + \hat{P}_h(x)\, e^{j\pi/4}$$

where

$$\hat{P}_h(x) = \frac{e^{-j\pi/4}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{QV(\tau)}{Qw_2(\tau)}\, e^{-jx\tau}\, d\tau ,$$

and $V(\tau)$ and $w_2(\tau)$ are Fock type Airy functions, and $Q = \frac{\partial}{\partial \tau}$. The q^* function is computed as follows:

1) for $x \leq -3$

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{1}{2}\sqrt{|x|}\left(1 - j\,\frac{2}{x^3}\right) e^{j\frac{x^3}{12}}\; e^{j\pi/4},$$

2) for $-3 < x < 2$

$$q^*(x) = q^*(x_i) + \frac{(x - x_i)}{(x_{i+1} - x_i)}\left(q^*(x_{i+1}) - q^*(x_i)\right),$$

where the $q^*(x_i)$ are tabulated values (see references A and B) and $x_{i+1} - x_i = 0.1$ with $x_i \leq x \leq x_{i+1}$.

3) For $x \geq 2$

$$q^*(x) = \frac{1}{2\sqrt{\pi x}} - \frac{e^{j\pi/6}}{2\sqrt{\pi}} \sum_{n=1}^{5} \frac{e^{x\bar{q}_n e^{-j\frac{5\pi}{6}}}}{\bar{q}_n[A_i(-\bar{q}_n)]}$$

where $A_i(\tau)$ is the Miller type Airy function.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AMC	-0.5*CEXP(J*PI/6)/SQRT(PI)
AQ	Miller type Airy function at Q
C	0.5/SQRT(PI)
EXC	CEXP(-5*PI/6)
I	Smallest integer closest to 10*X
PI	π
Q	Zeroes of derivative of Miller type Airy function
QFUN	Q function
QI	Imaginary part of tabulated Q function
QR	Real part of tabulated Q function
X	Argument of Q function
XI	Real number representation of I

5. I/O VARIABLES:

A. INPUT	LOCATION
PI	/PIS/
X	F.P.

B. OUTPUT	LOCATION
QFUN	FUNCTION

6. CALLING ROUTINES:

RPLSCL

SCLRPL

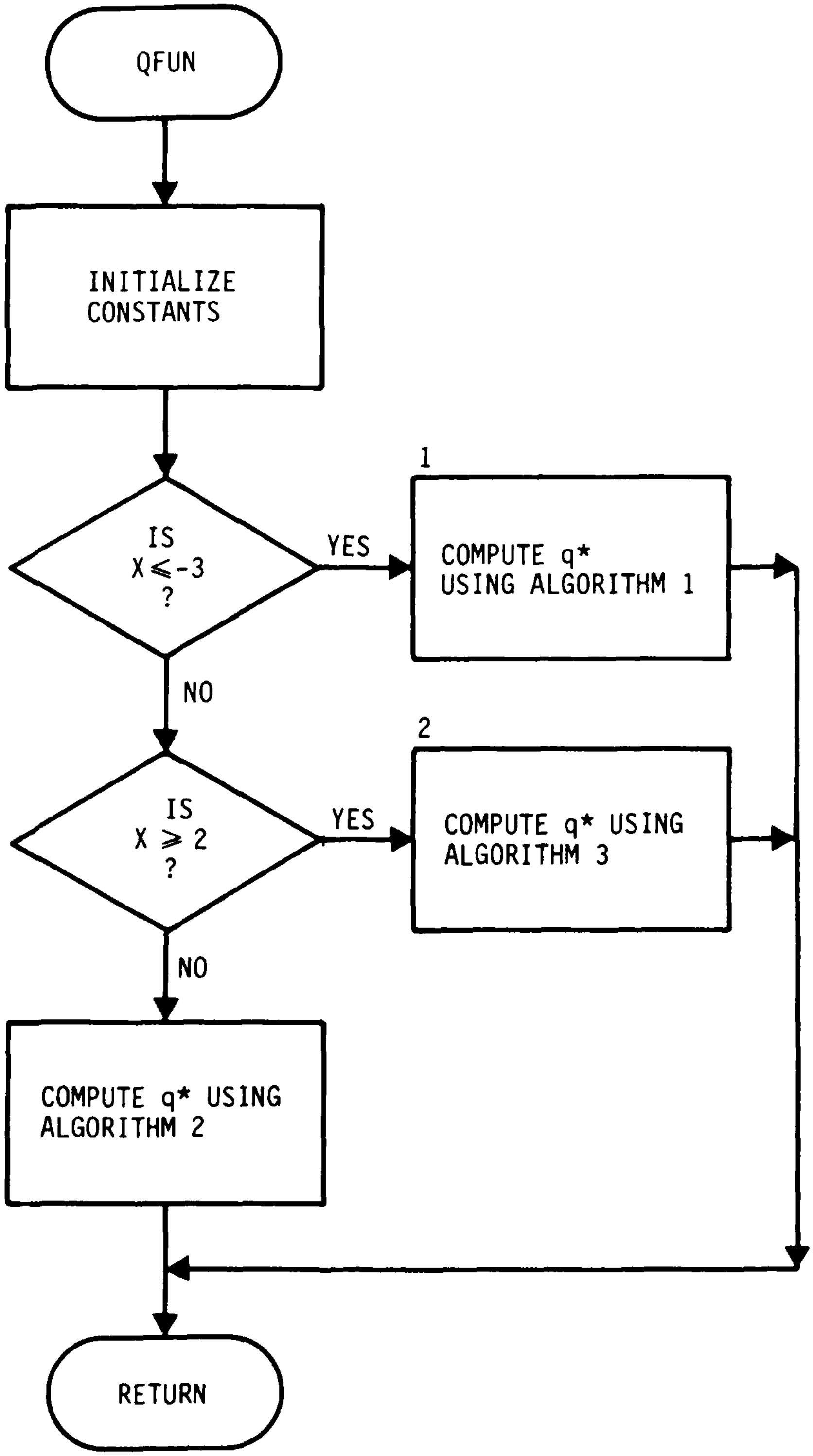
SCTCYL

7. CALLED ROUTINE: BEXP

8. REFERENCES:

A. P. H. Pathak, "An Asymptotic Analysis of the Scattering of Plane Waves by a Smooth Convex Cylinder," paper to appear in J. Radio Science. (Also The Ohio State University Electro-Science Laboratory Technical Report 784583-3, March 1978).

B. N. A. Logan, "General Research in Diffraction Theory," Vol. I, LMSD-288087; and Vol. II, LMSD-288088, Missiles and Space Division, Lockheed Aircraft Corp., 1959.


QFUN
INITIALIZE CONSTANTS
IS X ⩽ -3 ?
YES
1
COMPUTE q* USING ALGORITHM 1
NO
IS X ⩾ 2 ?
YES
2
COMPUTE q* USING ALGORITHM 3
NO
COMPUTE q* USING ALGORITHM 2
RETURN

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

END

FILMED

02 — 84

DTIC