



FACILITY FORM 602	N 69-14869	
	(ACCESSION NUMBER)	(THRU)
	<u>545</u>	<u>1</u>
	(PAGES)	(CODE)
	<u>NASA-CR # 94313</u>	<u>23</u>
	(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

This work was performed for the Jet Propulsion **RE-ORDER NO. 68-499**
Laboratory, California Institute of Technology,
sponsored by the National Aeronautics and Space
Administration under Contract NAS7-100 **Supersedes Re-order No. 68-113**
Vol. III

TR-67-700-10-2

FORTRAN OPTICAL LENS
DESIGN PROGRAM
Volume III

Program Organization and Description

Paul J. Firnett
Leslie A. Wilson

1 October 1968

INFORMATICS INC.
2000 Riverside Drive
Los Angeles, California 90039

PRECEDING PAGE BLANK NOT FILLED.

TR-67-700-10-2

Page iii

Volume III

Page Revised 10/1/68

TABLE OF CONTENTS (Continued)

		<u>Page</u>
3.1	Link 1	3-24
	3.1.1	OPUNCZ 3-25
	3.1.2	READSZ 3-27
	3.1.3	RERE/Z 3-34
	3.1.4	STAR9Z 3-36
	3.1.5	STARIZ 3-39
	3.1.6	MTFPNZ 3-41
	3.1.7	UNBCD. 3-42
	3.1.8	VCNVR. 3-43
3.2	Link 2	3-47
	3.2.1	ARPTSZ 3-48
	3.2.2	CELAIZ 3-50
	3.2.3	CLOSSZ 3-56
	3.2.4	CPLOTZ 3-60
	3.2.5	CROSSZ 3-63
	3.2.6	ENDPTZ 3-67
	3.2.7	EIRGCZ 3-73
	3.2.8	PFILEZ 3-75
	3.2.9	PROFLZ 3-80
	3.2.10	RANGEZ 3-82
	3.2.11	ZYCUTZ 3-87
3.3	Link 3	3-89
	3.3.1	TWNR4Z 3-90
	3.3.2	TWNR5Z 3-95
3.4	Link 4	3-99
	3.4.1	LATTZZ 3-100
	3.4.2	MERIDZ 3-102
	3.4.3	SMPOPZ 3-104
3.5	Link 5	3-111
	3.5.1	BNDCHZ 3-112
	3.5.2	CORDEZ 3-119
	3.5.3	CYCLEZ 3-121
	3.5.4	DOTPZ 3-129
	3.5.5	DPRNTZ 3-131

TABLE OF CONTENTS (Continued)

		<u>Page</u>	
	3.5.6	GETRHZ	3-134
	3.5.7	INCRMZ	3-136
	3.5.8	JURYZ	3-139
	3.5.9	MEVCTZ	3-142
	3.4.10	MODDLZ	3-147
	3.5.11	MWVCTZ	3-149
	3.5.12	STAR2Z	3-153
	3.5.13	MATINV	3-158
3.6	Link 6		3-161
	3.6.1	PARAXZ	3-162
	3.6.2	STAR3Z	3-164
3.7	Link 7		3-170
	3.7.1	GETLCZ	3-171
	3.7.2	STAR7Z	3-173
3.8	Link 8		3-184
	3.8.1	STAR8Z	3-185

PROGRAM MODIFICATION

Section 1	ADDING CAPABILITY	1-1
Section 2	ADDING NEW INPUT	2-1
Section 3	CHANGING THE DIMENSION OF \vec{E}	3-1
Section 4	CONVERSION PROBLEMS	4-1
	4.1 Input/Output	4-1
	4.2 Significance	4-2

STORAGE MAP III-1

PROGRAM LISTING III-0-1

Appendix A SC4020 PLOT PACKAGE A-1

INTRODUCTION

1. PURPOSE

This document contains a detailed description of the program organization and structure. Its primary purpose is to aid the programmer in developing an understanding of how the model described in volumes I and II was implemented into a computer program.

2. SCOPE

The section on program structure presents a complete picture of the organization starting with the overlay diagram and proceeding through a discussion of the characteristics of each subroutine. Because of the importance of labeled common in the communication of parameters between modules it is discussed in detail.

The section on program modification gives some indication, by means of practical examples, of the difficulties one might encounter in attempting to change the program.

The storage map produced on the JPL direct couple system is essentially an overlay diagram with storage requirement information and thus acts as a supplement to figure 1.

The source program listing and appendix A are self explanatory.

TR-67-700-10-2
Page III-1-1
Program Structure

PROGRAM STRUCTURE

Section 1

GENERAL DISCUSSION

The program consists of 49 subroutines arranged into an overlay structure comprising a main link (link 0) and 8 dependent links. This structure is schematically illustrated in Figure 1 where vertical lines represent the links into which the program is divided, and horizontal lines (from which the vertical lines proceed) represent the logical origins of the links. Included in the diagram are the deck names of the subroutines assigned to each of the program links.

The functional responsibilities of each link are discussed in section 3 which includes a detailed description (with flow charts) of each subroutine.

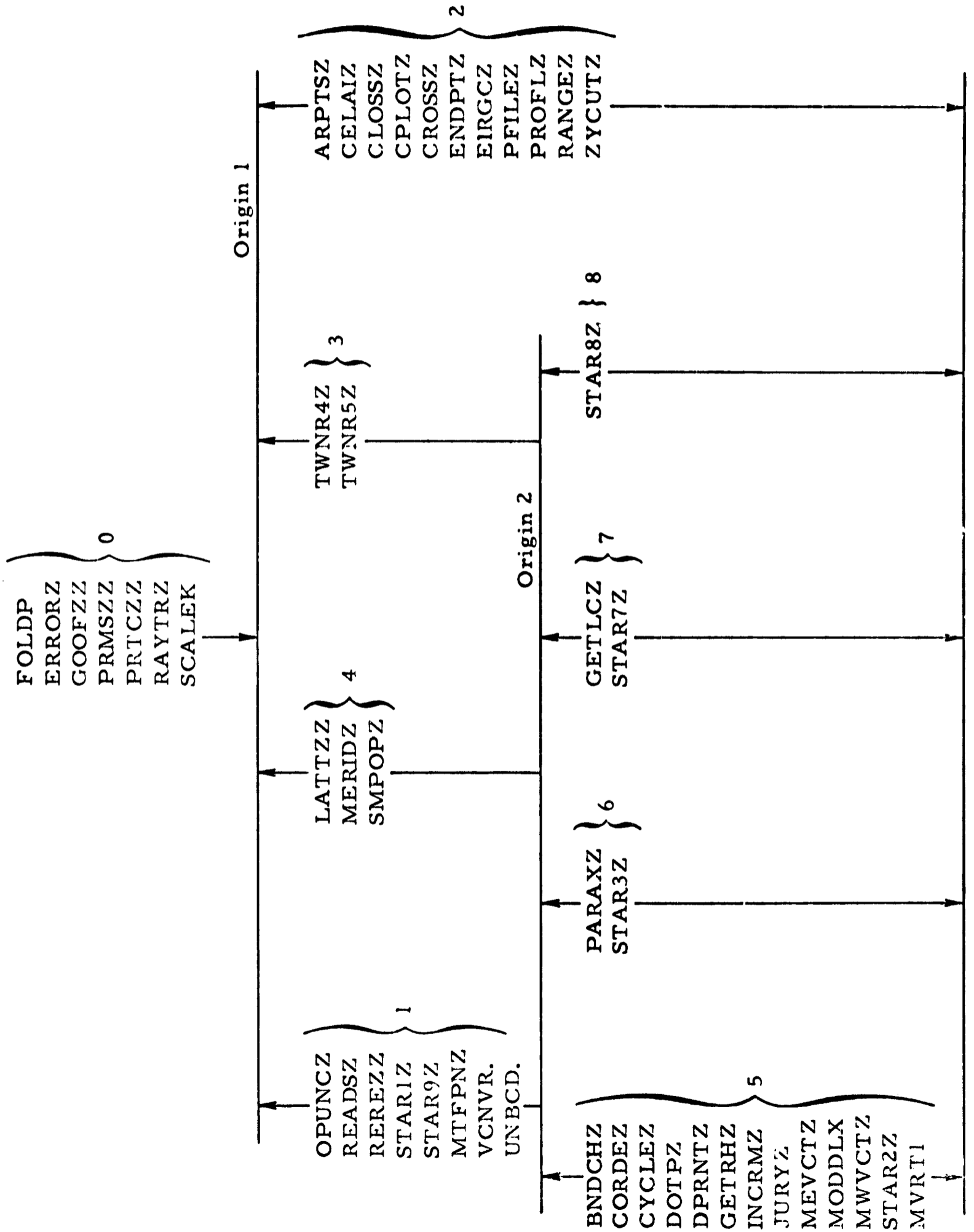


Figure 1. FOLDP Overlay Structure

Section 2

LABELED COMMON

The program employs several labeled common blocks for communicating information between modules. Two types of labeled common can be identified:

- a) Inter link common contains information which is pertinent to at least two parallel links and must therefore be preserved during the overlay process.
- b) Intra link common contains information which is pertinent only within a link or chain of links and need not be preserved during overlay.

2.1 INTERLINK COMMON

The 6 different interlink labeled common blocks are listed (by name) in Figure 2 which is a two entry table: the first entry being the labeled common name and the second entry being a list of deck names of those subroutines which require the block in question.

<u>COMMON BLOCK</u>	<u>DECKS</u>
DATA	FOLDP, PRMSZZ, PRTCZZ, RAYTRZ, OPUNCZ, RE ADSZ, STAR1Z, STAR9Z, ARPTSZ, CELAIZ, CLOSSZ, CPLOTZ, CROSSZ, ENDPTZ, PFILEZ, RANGEZ, ZYCUTZ, LATTZZ, MERIDZ, SMPOPZ, TWNR4Z, TWNR5Z, BNDCHZ, CYCLEZ, DPRNTZ, GETRHZ, INCRMZ, JURYZ, MEVCTZ, MODDLX, MWVCTZ, STAR2Z, PARAXZ, STAR3Z, STAR7Z, STAR8Z, MIFPNZ
PERTB	FOLDP, READSZ, STARIZ, STAR8Z
PLOTG	FOLDP, READSZ, STARIZ, PROFLZ, CPLOTZ, PFILEZ
SYMBLS	FOLDP, READSZ, STAR9Z, STARIZ
AZOBJ	RAYTRZ, PFILEZ, TWNR4Z, TWNR5Z, SMPOPZ, MERIDZ, GETRHZ, MEVCTZ, BNDCHZ, PARAXZ, STAR3Z, STAR7Z, STAR8Z
PRNT	PRTCZZ, READSZ, STAR9Z, STARIZ, PROFLZ, CELAIZ, TWNR4Z, TWNR5Z, DPRNTZ, STAR2Z, BNDCHZ, CYCLEZ, STAR3Z, STAR7Z, STAR8Z

Figure 2. Decks Requiring Interlink Common

The data items associated with each common block are presented in the form of a table containing three entries:

- a) Symbol - This is the symbol employed by the program to represent the variable(s) listed under "Parameter". If the variable is a vector or matrix the dimensions are included.
- b) Type - Each variable in the program is of a certain type in the FORTRAN sense. The following abbreviations are used:
 - I = Integer
 - R = Real
 - IV = Integer Vector
 - RV = Real Vector
 - IM = Integer Matrix
 - RM = Real Matrix
- c) Parameter - This describes the physical data or parameter(s) associated with "Symbol".

2.1.1 DATA

Associated with DATA are all of the input parameters except sensitivity and profile plot.

COMMON/DATA/

<u>Symbol</u>	<u>Type</u>	<u>Parameter</u>
NCNTRL	I	No. of options
CONTRL(10)	IV	Input options
TITLE(12)	RV	Input title
DATE(3)	IV	Month, Day, Year
PUNCID	R	Punch identification
DATA(3486)	RV	Geometry, Design, etc. data

The various NAMELIST input symbols are defined in the input description section of volume II. Each namelist parameter is equivalenced to some component of the real vector DATA, e.g., $ODIST = DATA(31)$. The DATA position establishes a unique location for each parameter which is of paramount importance when processing symbol cards. The following table lists the equivalent DATA index for each NAMELIST parameter and includes a reference with the abbreviations:

- G = GEOM
- D = DESIGN
- B = BNDRY
- S = SUBSTN

DATA Vector

<u>Symbol</u>	<u>Type</u>	<u>DATA Index</u>	<u>NAMELIST Ref.</u>
LMODE	I	1	G
NRAYS	I	2	G
NSLCS	I	3	G
NCLRS	I	4	G
NJAIL	I	5	D
NSUBT	I	6	S
NSUBP	I	7	S
NIPLN	I	8	G
IMODE	I	9	G
NSPLN	I	10	G
NOBJH	I	11	G
NSURF	I	12	G

<u>Symbol</u>	<u>Type</u>	<u>DATA Index</u>	<u>NAMelist Ref.</u>
ITNPRT	I	13	D
DELY	R	14	G
FNUMB	R	15	D
FLNGH	R	16	D
WFLGH	R	17	D
ZETA	R	18	L
HEXPP	R	19	D
DEXPP	R	20	D
WEXPP	R	21	D
DLPLN	R	22	G
OMGA2	R	23	G
OMGA1	R	24	G
DELD	R	25	G
EPRAD	R	26	G
PSCAL	R	27	G
OMGAF	R	28	D
SPFEA	R	29	D
DUMIN	R	30	D
ODIST	R	31	G
HO	R	32	G
DELH	R	33	G
SYSMX	R	34	D
WXDIR	R	35	D
WYDIR	R	36	D
ROTAN	R	37	G
NDSGN(1)	IV	38	D

<u>Symbol</u>	<u>Type</u>	<u>DATA Index</u>	<u>NAMelist Ref.</u>
NDSGV(1)	IV	42	D
WOBJH(1)	RV	53	D
WCLRH(1)	RV	60	D
WIMHT(1)	RV	67	D
EIMHT(1)	RV	74	D
CIMPT(1)	RV	81	G
WCLRS(1)	RV	88	D
LATTC(1, 1)	RM	94	G
SURFC(1, 1)	RM	182	G
DESGN(1, 1)	RM	2182	D
SUBST(1)	RV	2682	S
BOUNDS(1, 1)	RM	3182	B
NCOND	I	3483	B
ATRGR	I	3484	D
GAUSS	R	3485	D

2.1.2 PERTB

PERTB common contains the sensitivity input parameters as described in volume II.

COMMON/PERTB/

<u>Symbol</u>	<u>Type</u>	<u>Parameter</u>
PERTB(30, 4)	RM	Perturbation Matrix
NPRTB	I	Mp No. of Parameters
REFOCS	I	Refocus Option

2.1.3 PLOTC

PLOTC common contains the profile plot input parameters as described in volume II.

<u>COMMON/PLOTC/</u>		
<u>Symbol</u>	<u>Type</u>	<u>Parameter</u>
YMAXX	R	y_{\max} Maximum Aperture
DZMIN	R	ΔZ_{\min} Minimum Thickness
NPTS	I	No. of points per arc

2.1.4 SYMBLS

SYMBLS common contains the symbols from design, substitution, and boundary condition symbol (input) cards. The symbols are preserved for printout purposes only.

<u>COMMON/SYMBLS/</u>		
<u>Symbol</u>	<u>Type</u>	<u>Parameter</u>
SDESN(12, 50)	IM	Design Symbols
SUBSYM(2, 250)	IM	Substitution Symbols
BDYSYM(2, 100)	IM	Boundary Condition Symbols

2.1.5 AZOBJ

AZOBJ common contains the object azimuth and its sine and cosine which are needed whenever a ray is to be traced.

<u>COMMON/AZOBJ/</u>		
<u>Symbol</u>	<u>Type</u>	<u>Parameter</u>
THTR	R	θ in radians (Azimuth)
SNTHTR	R	$\sin \theta$
CSTHTR	R	$\cos \theta$

2.1.6 PRNT

PRNT common contains the current page and line numbers. It is used in connection with the print control subroutine (PRTCTL) to format the output.

<u>COMMON/PRNT/</u>		
<u>Symbol</u>	<u>Type</u>	<u>Parameter</u>
LINE	I	Current Line No.
PAGE	I	Current Page No.

2.2 INTRALINK COMMON

Figure 3 lists each of the intralink labeled common blocks along with the decks which reference them.

<u>COMMON BLOCK</u>	<u>DECKS</u>
HEADER	FOLDP, PRTCZZ
TMTWIN	TWNR4Z, TWNR5Z
TMPATT	SMPOPZ, MEVCTZ, STAR3Z, STAR7Z, STAR8Z
TMDESN	DOTPZ, DPRNTZ, JURYZ, MWVCTZ, STAR2 Z, MEVCTZ, BNDCHZ, CYCLEZ
ELMAIR	PROFLZ, CPLOTZ, PFILEZ, ENDPTZ, RANGEZ, CROSSZ, CLOSSZ, EIRGCZ, CELAIZ
CODRNG	PROFLZ, ZYCUTZ, CPLOTZ, PFILEZ, ENDPTZ, RANGEZ, EIRGCZ

Figure 3. Decks Requiring Intralink Common

The following subsections describe the intralink labeled common blocks in detail.

2.2.1 HEADER

HEADER common communicates the current option and the current substitution set number to the print control subroutine, PRTCTL.

COMMON/HEADER/

<u>Symbol</u>	<u>Type</u>	<u>Parameter</u>
JTEMP	I	Current Option No.
JSUBST	I	Current Substitution Set

2.2.2 TMTWIN

TMTWIN common consists of the single real vector TEMPS (1200) which is used to store position and direction data at each surface for a specified pair of rays.

2.2.3 TMPATT

TMPATT common contains all of the data associated with tracing a ray pattern from a single object point (and a single color) to an image plane.

COMMON/TMPATT/

<u>Symbol</u>	<u>Type</u>	<u>Parameter</u>
NMISS		N_M No. of Misses
NRFLCT	I	N_{RF} No. of Reflections
NVIGN	I	N_v No. of Vignets
NRAYSB	I	N_R No. of Successes
XBAR	R	\bar{x} Average x
YBAR	R	\bar{y} Average y
XSTAR	R	x^* RMS x
YSTAR	R	y^* RMS y
RSTAR	R	R^* Spot Size
SJSTAR	R	S^* Image Plane Position
XCURL(201)	RV	$\left. \begin{matrix} \tilde{x}_i \\ \tilde{y}_i \end{matrix} \right\}$ Image Coordinates
YCURL(201)	RV	
MXBAR	R	m_x See volume I section 6
KXBAR	R	k_x See volume I section 6
MYBAR	R	m_y See volume I section 6
KYBAR	R	k_y See volume I section 6
AX	R	A_x See volume I section 6
BX	R	B_x See volume I section 6
CX	R	C_x See volume I section 6
AY	R	A_y See volume I section 6
BY	R	B_y See volume I section 6
CY	R	C_y See volume I section 6
ACAP	R	A See volume I section 6
BCAP	R	B See volume I section 6
CCAP	R	C See volume I section 6

2.2.4 TMDESN

TMDESN common is used as a repository for various design parameters including the weight vector \vec{W} and the error vector \vec{E} .

<u>COMMON/TMDESN/</u>			
<u>Symbol</u>	<u>Type</u>		<u>Parameter</u>
PERIOD	I	P	Incrementation Period
PCOUNT	I	P_c	No. of Failures
NOFFND	I	N_{OF}	No. of Offenders
JAIL(10,10)	IM	H	Jail Matrix
SENTN(10)	IV	G	Sentence Vector
DIMVCT	I	M	Dimension of \vec{E}
CORD(6)	IV	l_i	$1 \leq i \leq N_c$ Color Order
SIZE	R	ξ_K	Scale Factor
DETERM	R	$ I $	Determinant of I
E0LNG	R	$ \vec{E} $	Old Length
ESLNG	R	$ \vec{E}^* $	New Length
IMPRV	R	ϵ	Relative Improvement
VCTSTR(6000)	RV	$\vec{W}, \vec{E}, \partial \vec{E} / \partial u_1, \dots, \partial \vec{E} / \partial u_n$	
VINDX(11)	IV	$i \cdot M + 1 \quad 1 \leq i \leq 11$	
DJOLD(7)	RV		Temporary storage for d_j

2.2.5 ELMAIR

ELMAIR common contains the edited element and air space matrices and the vertex coordinates relative to the entrance pupil.

<u>Symbol</u>	<u>COMMON/ELMAIR/</u>		<u>Parameter</u>
	<u>Type</u>		
NELMT	I	N_E	No. of Elements
NAIRS	I	N_A	No. of Air Spaces
ELMMTX(2,100)	IM	E	Element Matrix
AIRMTX(2,100)	IM	A	Air Space Matrix
VTXCRD(100)	RV	δ_i	Vertex Coordinates

2.2.6 CODRNG

CODRNG common contains the range and code matrices.

<u>Symbol</u>	<u>COMMON/CODRNG/</u>		<u>Parameter</u>
	<u>Type</u>		
ZHAT(2,100)	RM	\hat{Z}	Range Matrix
YHAT(2,100)	RM	\hat{Y}	Range Matrix
CHAT(2,100)	IM	\hat{C}	Code Matrix

Section 3

LINK 0 THROUGH LINK 8

3.0 LINK 0

Link 0 is the main or control link of the program. It consists of the main program (FOLDP) and those subroutines which are required by one or more lower level links starting at logical origin 1.

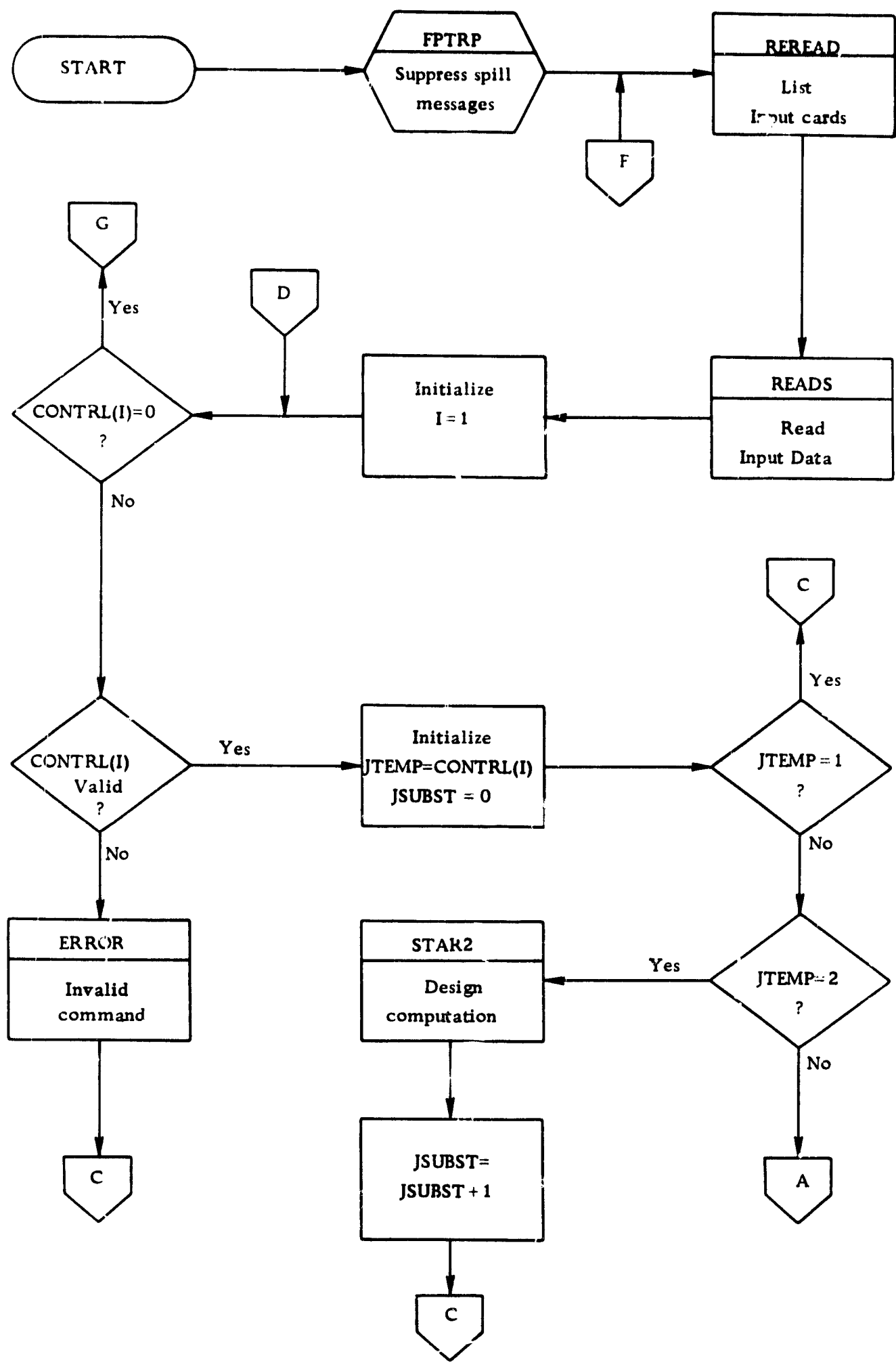
3.0.1 FOLDP

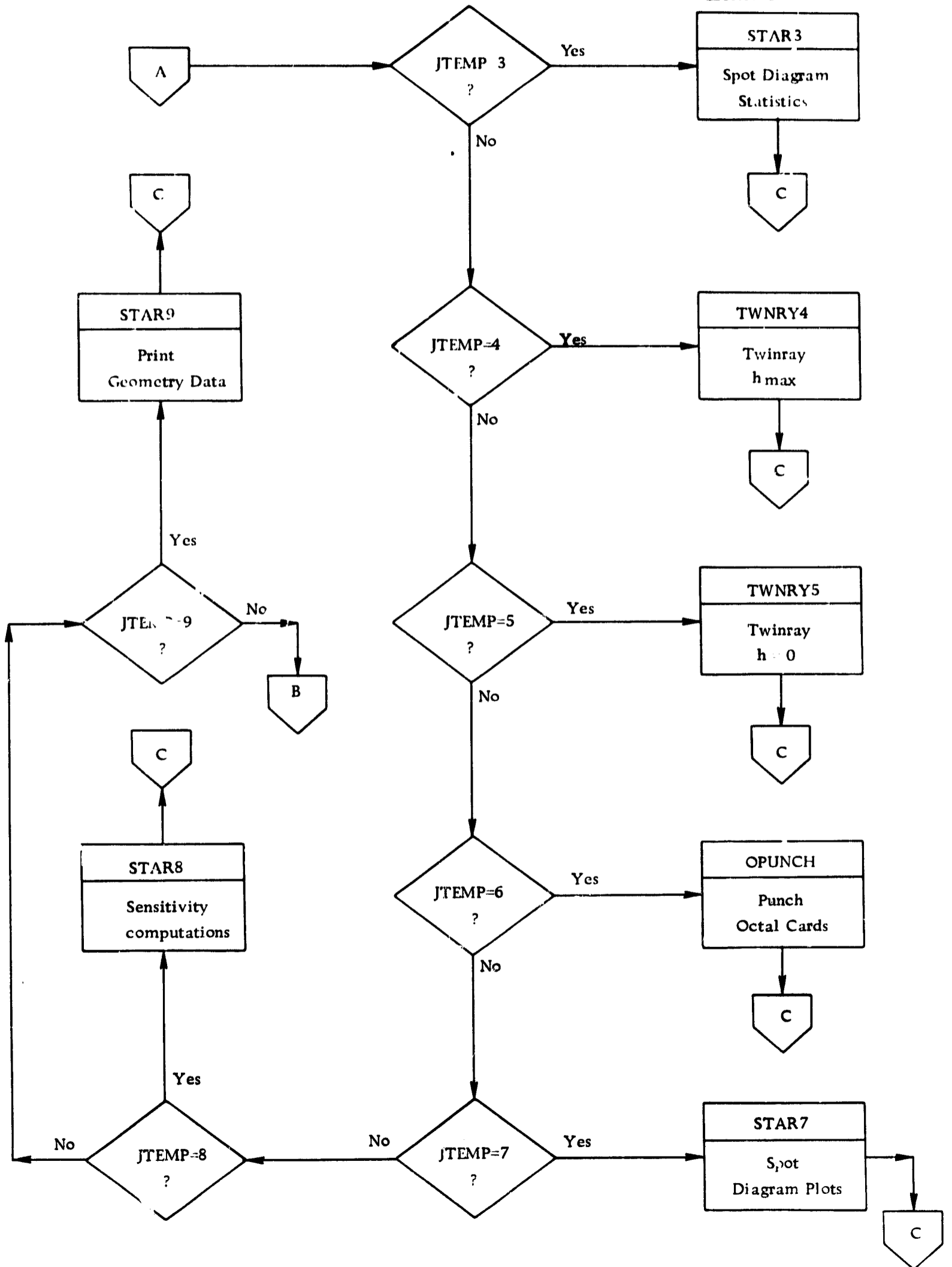
FOLDP is the main program which calls upon the input link and selected subroutines depending upon the options being executed.

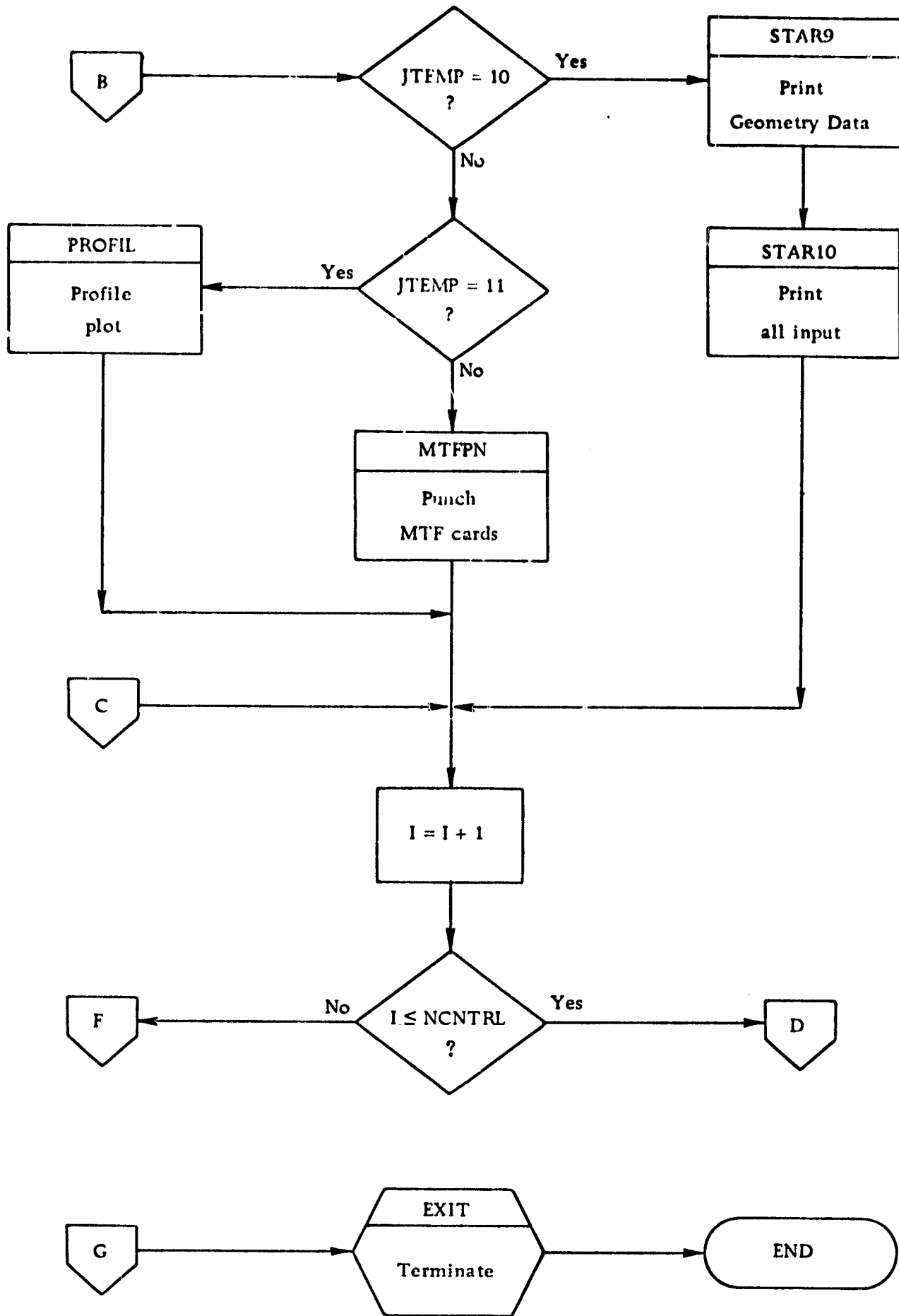
Utility Routines and Common References

COMMON/DATA/	STAR3
COMMON/PERTB/	PRMSUB
COMMON/SYMBOLS/	TWNR4
COMMON/HEADER/	TWNR5
COMMON/PLOT/	OPUNCH
FPTRP*	STAR7
REREAD	STAR8
READS	STAR9
ERROR	STAR10
STAR2	PROFIL
EXIT	MTFPN

* This is a system routine which processes floating point spills. It is called by FOLDP to suppress message printing regarding spills.







3.0.2 ERRORZ (ERROR, ERROR1, ERROR2, ERROR3, ERROR4)

ERRORZ is a general purpose message printer with 5 entry points which handle a variety of message types and formats.

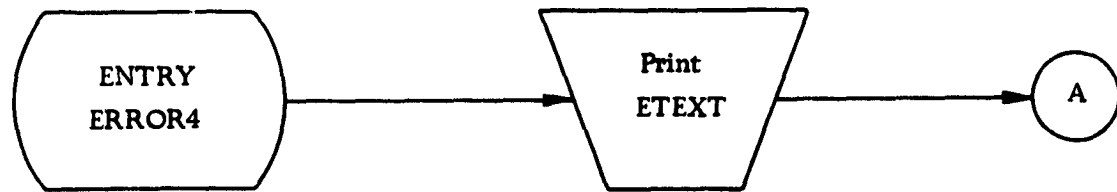
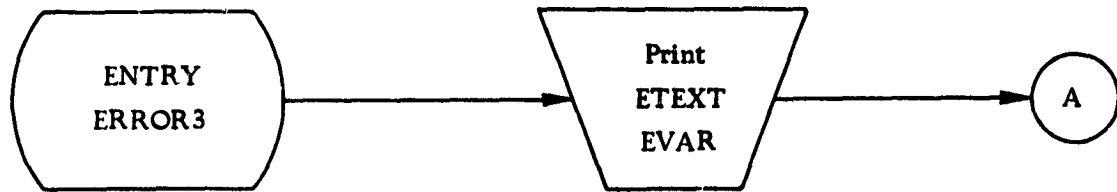
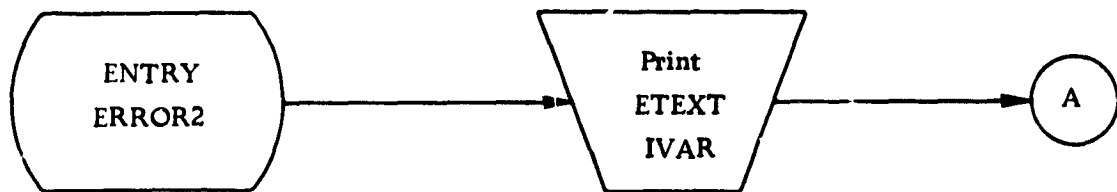
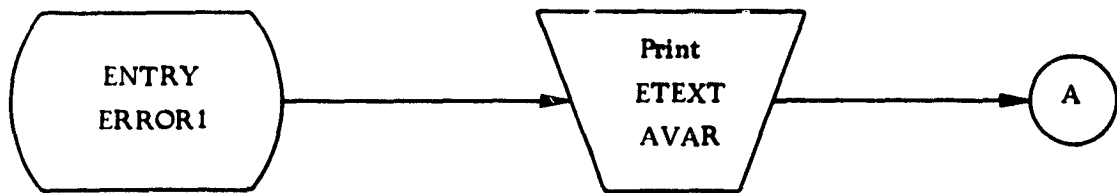
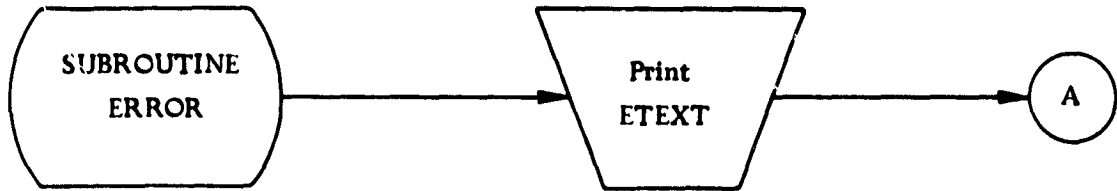
Calling Sequence

CALL ERROR (ETEXT)
CALL ERROR1 (ETEXT, AVAR)
CALL ERROR2 (ETEXT, IVAR)
CALL ERROR3 (ETEXT, EVAR)
CALL ERROR4 (ETEXT)

where ETEXT is a 6 word (36 character) message to be printed along with (AVAR) a BCD word, (IVAR) an integer word, or (EVAR) a real word. The initial concept was the cause job termination for all except ERROR4; however, the current mode is to always affect a normal return.

Utility Routines and Common References

COMMON	-	None
Utility	-	None



3.0.3 GOOFZZ (GOOF)

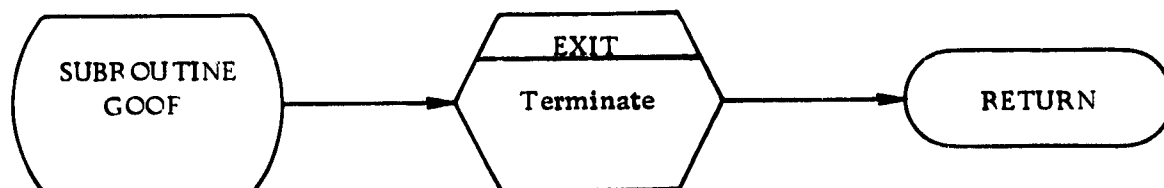
GOOFZZ is a job terminator which does nothing except to call the system subroutine EXIT.

Calling Sequence

CALL GOOF

Utility Routines and Common References

COMMON - None
EXIT



3.0.4 PRMSZZ (PRMSUB)

PRMSZZ performs substitution and restoration of the N_s substitution sets. If $N_s = 0$ it makes an immediate and normal return to the calling program. If $N_s > 0$ then the j th call in a sequence of $N_s + 1$ calls causes the j th set to be substituted and PRMSZZ takes the alternate return. The last call ($N_s + 1$) restores the nominal system and resets PRMSUB so that the sequence can be restarted.

Calling Sequence

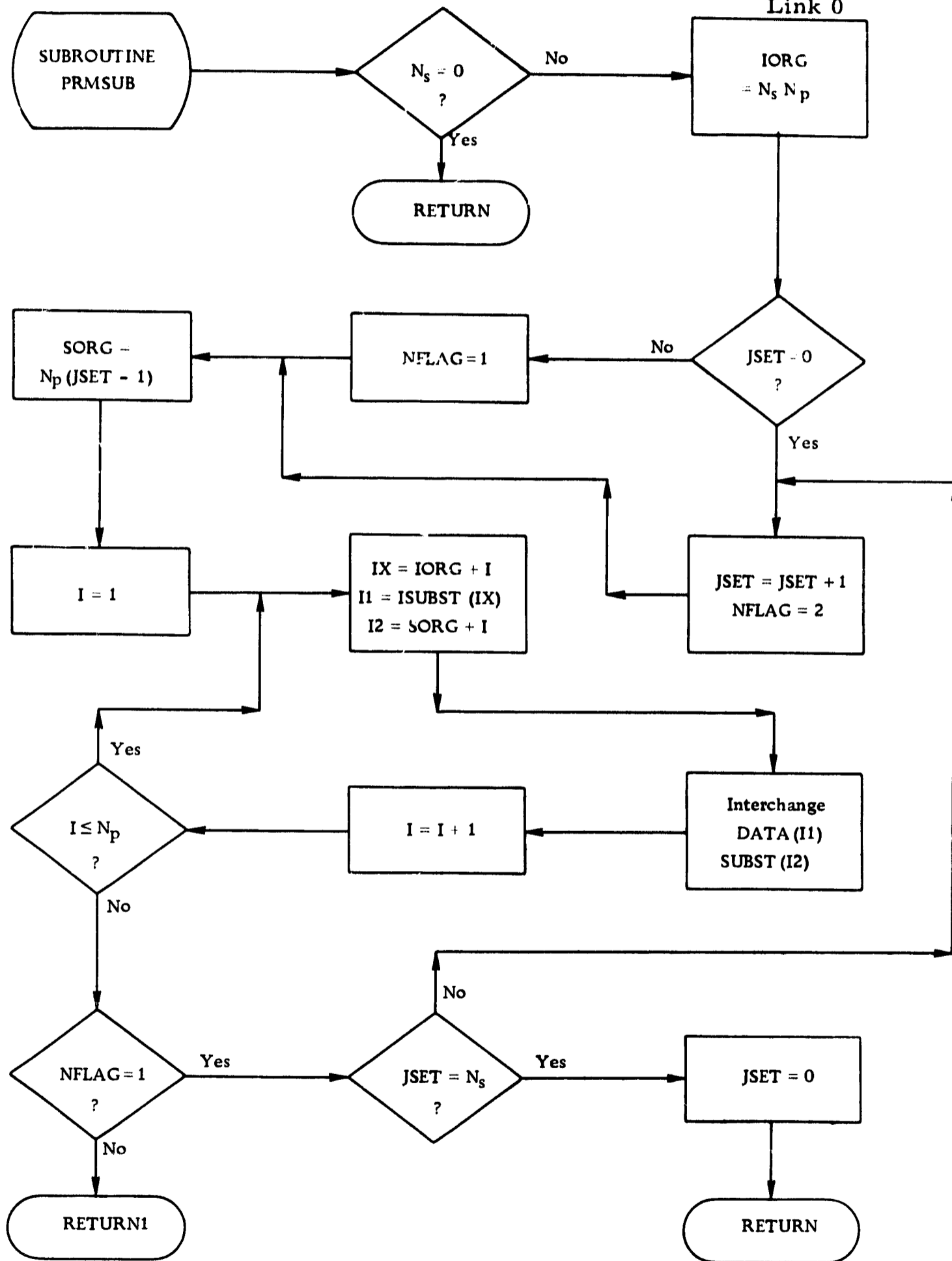
CALL PRMSUB(\$ALT)

where ALT is the statement number to which PRMSUB returns for each new substitution set.

Utility Routines and Common References

COMMON/DATA/

Utility - None



3.0.5 PRTCZZ (PRTCTL)

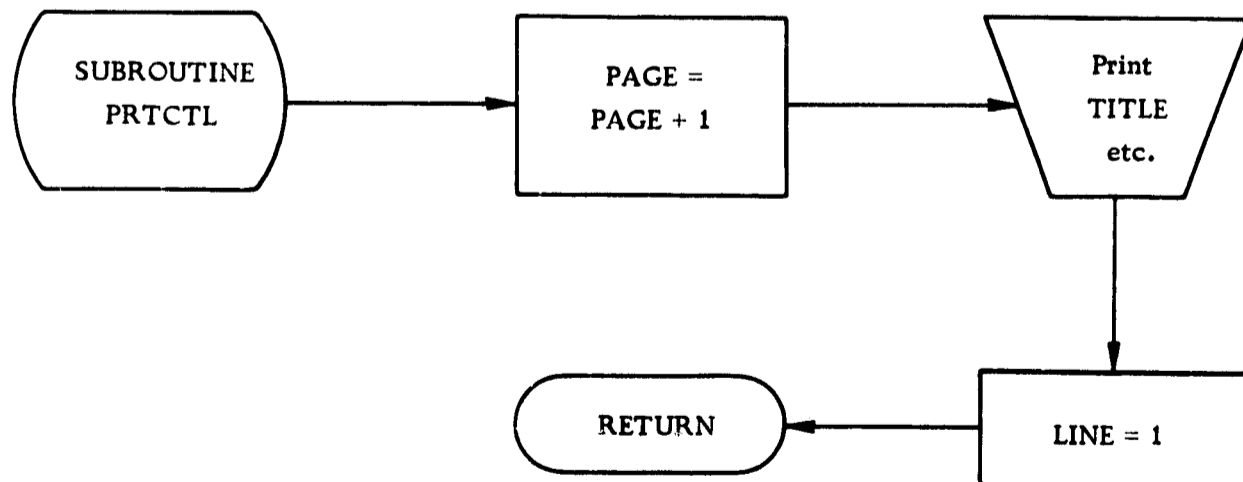
PRTCZZ is a print control routine which causes page ejection and main title generation including title, date, option, substitution set and page number.

Calling Sequence

CALL PRTCTL

Utility Routines and Common References

COMMON/DATA/
COMMON/PRNT/
COMMON/HEADER/
Utility - None



3.0.6 RAYTRZ (RAYTR)

RAYTRZ traces a ray of color i from the object point (h, θ) with unit radius entrance pupil coordinates (\hat{x}_o, \hat{y}_o) to a point (x_N, y_N, z_N) on σ_N . Alternate returns are provided to handle misses, internal reflections and vignets. RAYTRZ will optionally preserve the position (x, y, z) and direction (Q_x, Q_y, Q_z) of the ray at each surface in the system.

Calling Sequence

CALL RAYTR(XOHAT, YOHAT, HEIGHT, COLOR, XVCT,
YVCT, SMISS, \$NM, \$NR, \$NV, DFLAG, XMTRX,
QMTRX)

XOHAT: \hat{x}_o x coordinate on unit radius entrance pupil

YOHAT: \hat{y}_o y coordinate on unit radius entrance pupil

HEIGHT: h Object height

COLOR: i Color index

XVCT: Vector for storing (x_N, y_N, z_N)

QVCT: Vector for storing (Q_{xN}, Q_{yN}, Q_{zN})

SMISS: Set to surface number if miss, reflect, or vignet

\$NM: Alternate return if ray misses

\$NR: Alternate return if internal reflection

\$NV: Alternate return if ray vignets

DFLAG: Input flag to control saving position and direction at each surface.

XMTRX: Matrix for storing (x_i, y_i, z_i) $0 \leq i \leq N$

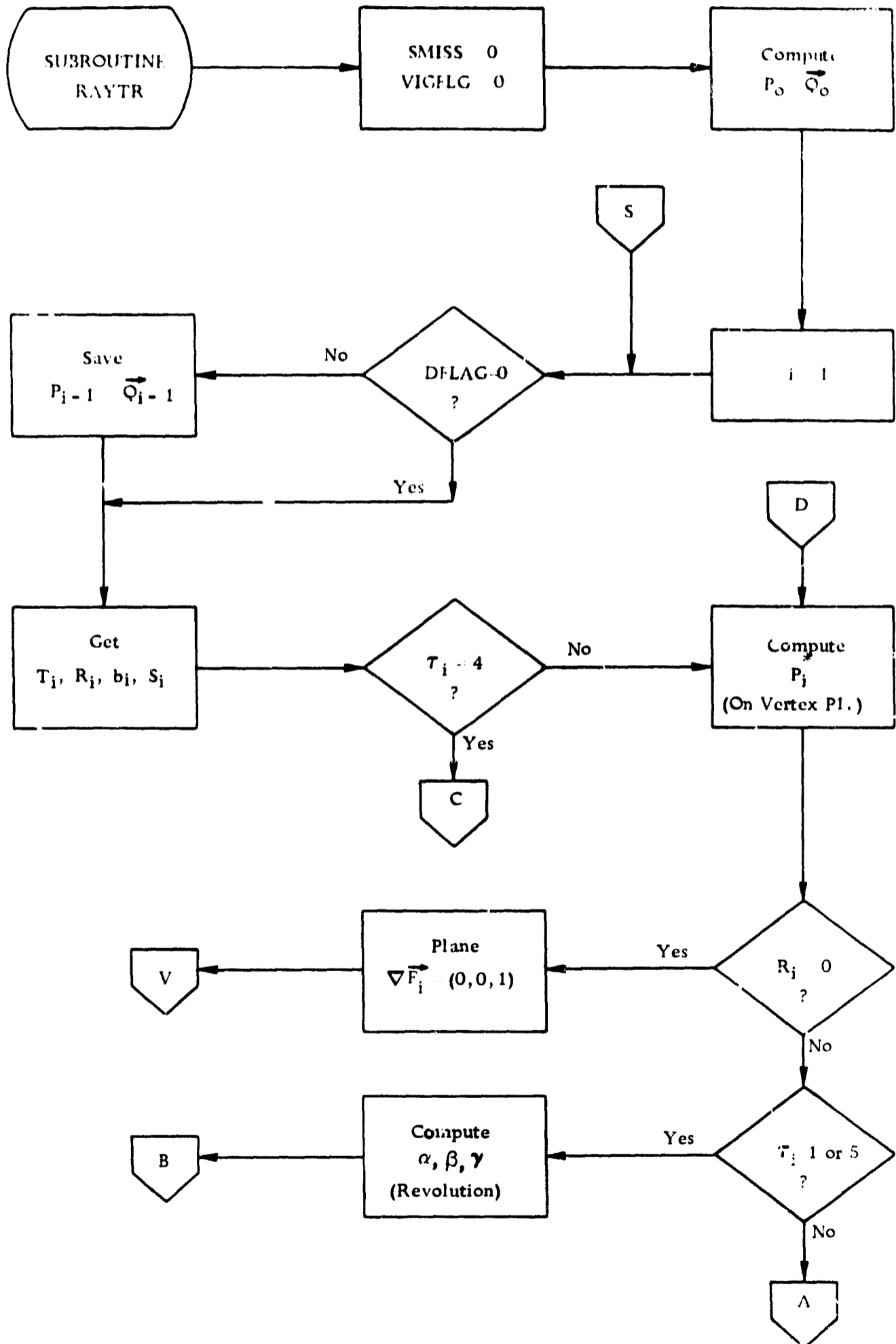
QMTRX: Matrix for storing (Q_{xi}, Q_{yi}, Q_{zi}) $0 \leq i \leq N$

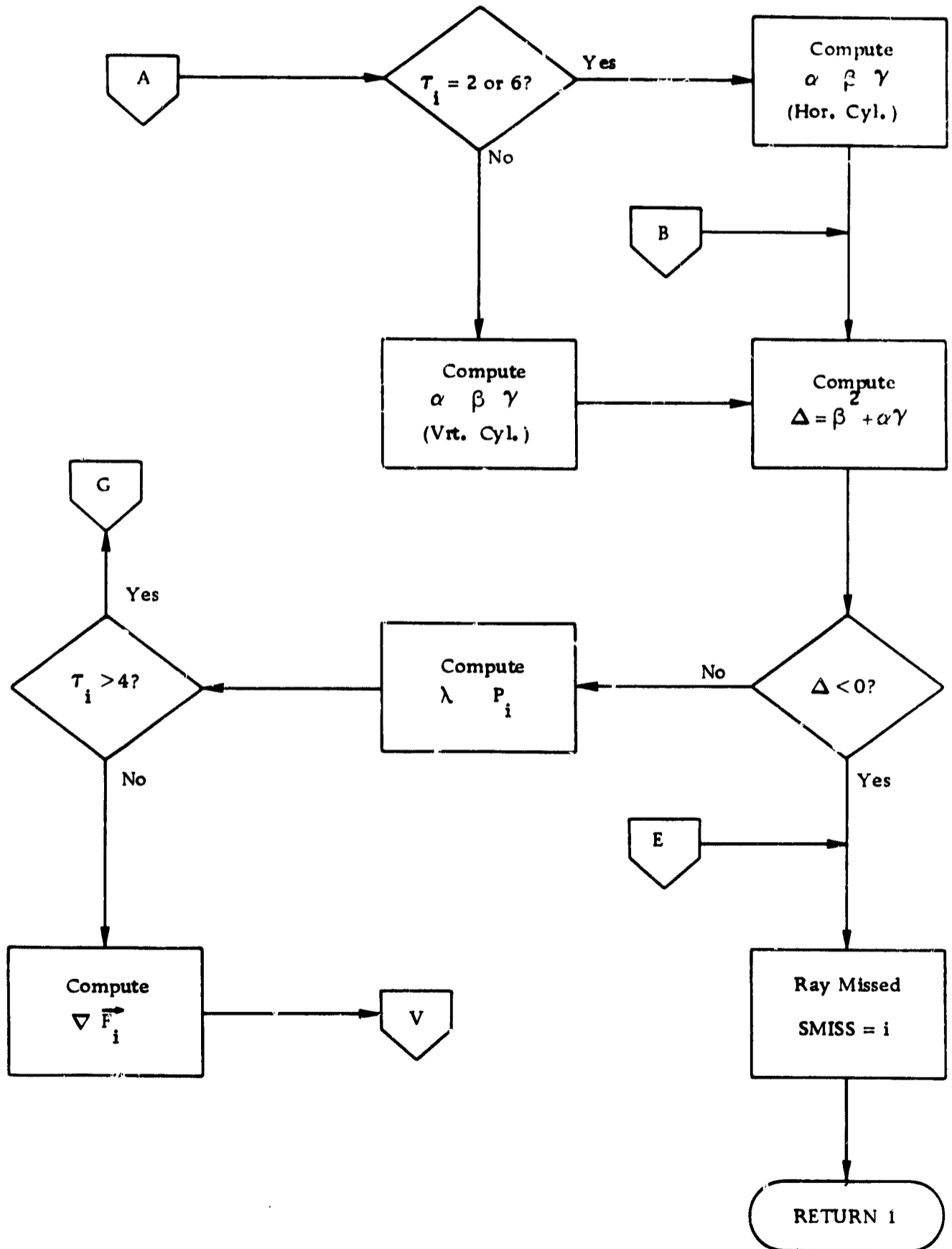
TR-67-700-10-2
Page III-3-13
Program Structure
Link 3

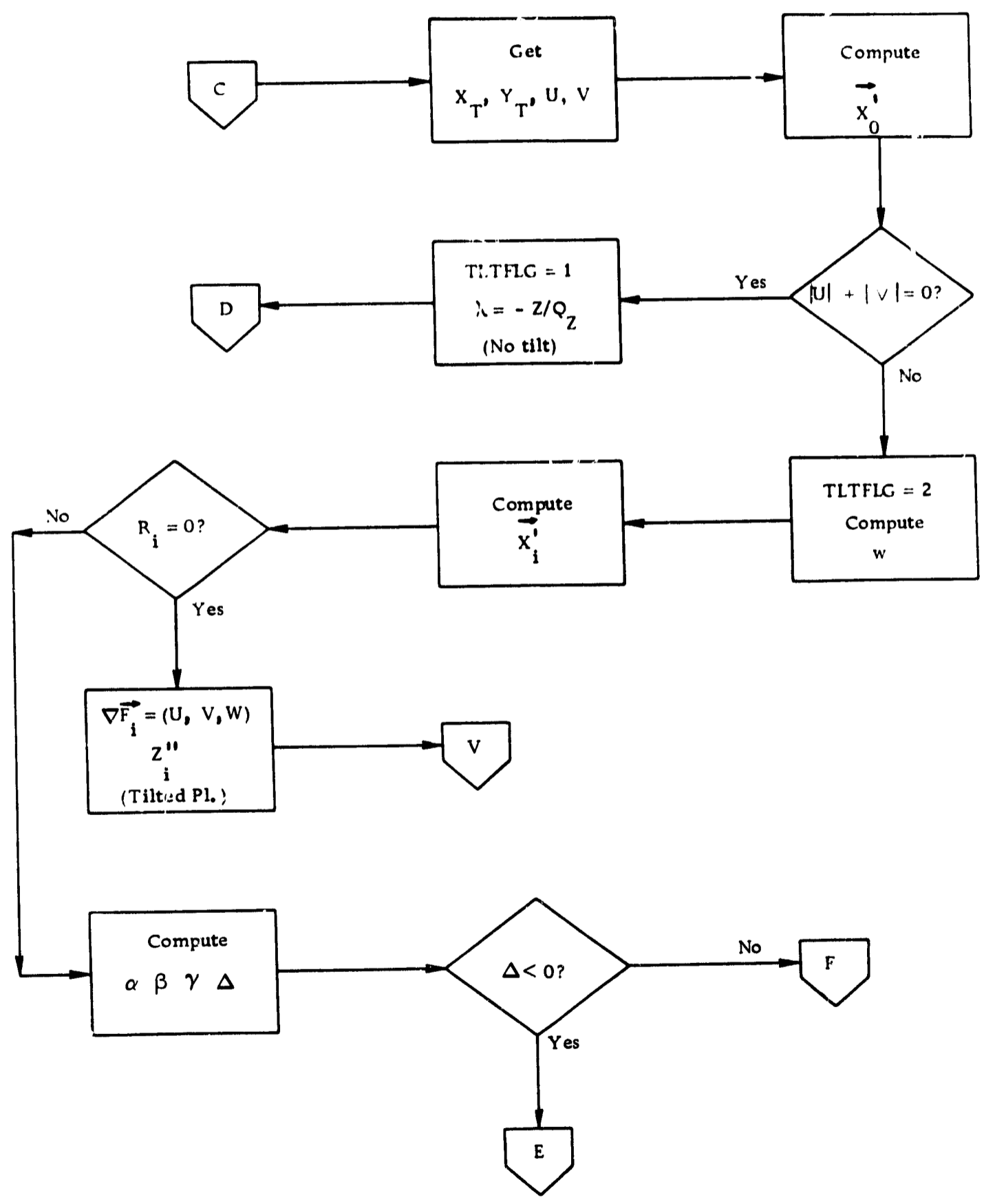
The azimuth of the object (θ) as well as $\sin\theta$ and $\cos\theta$ must be entered into AZOBJ common before calling RAYTR.

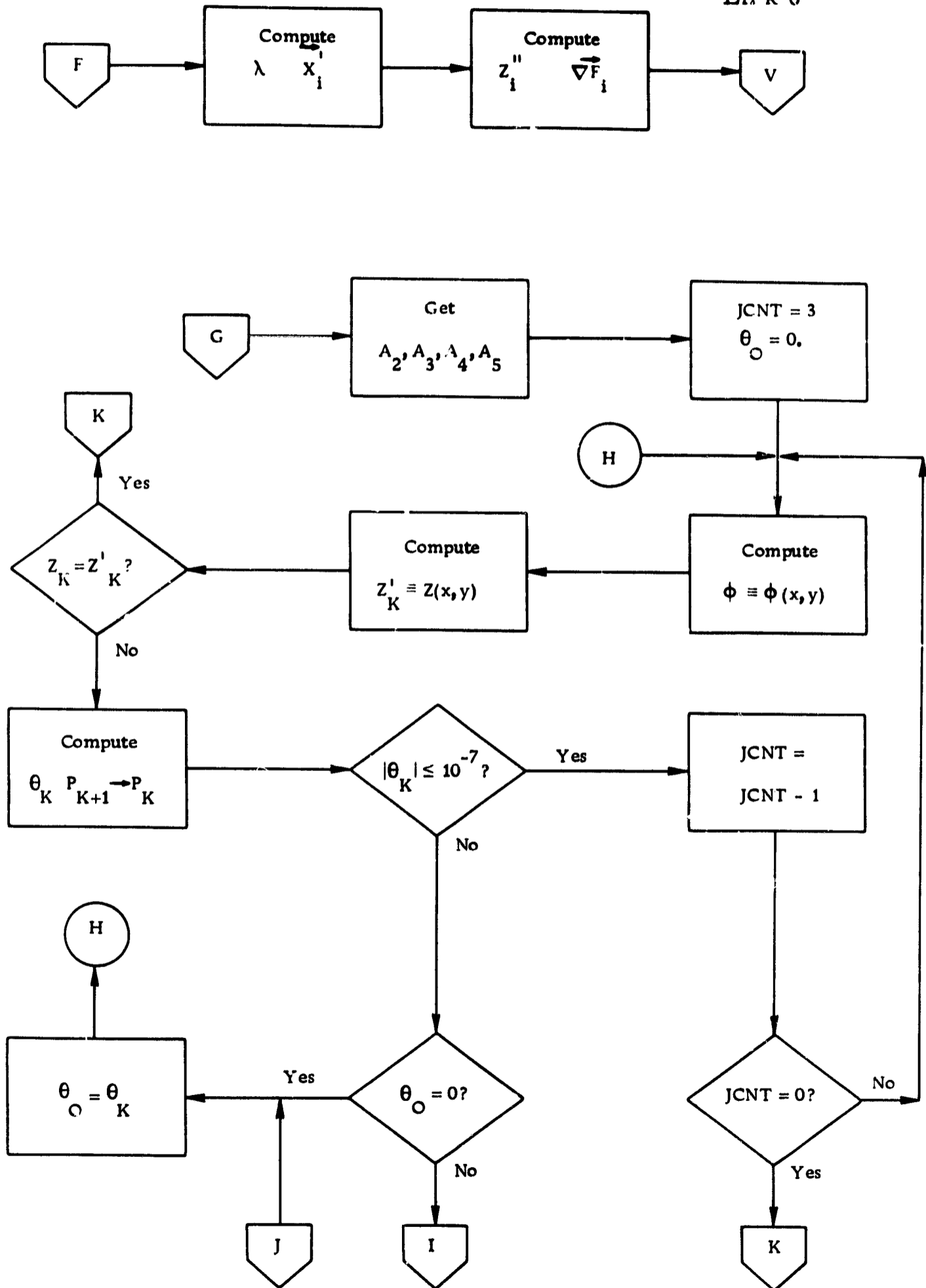
Utility Routines and Common References

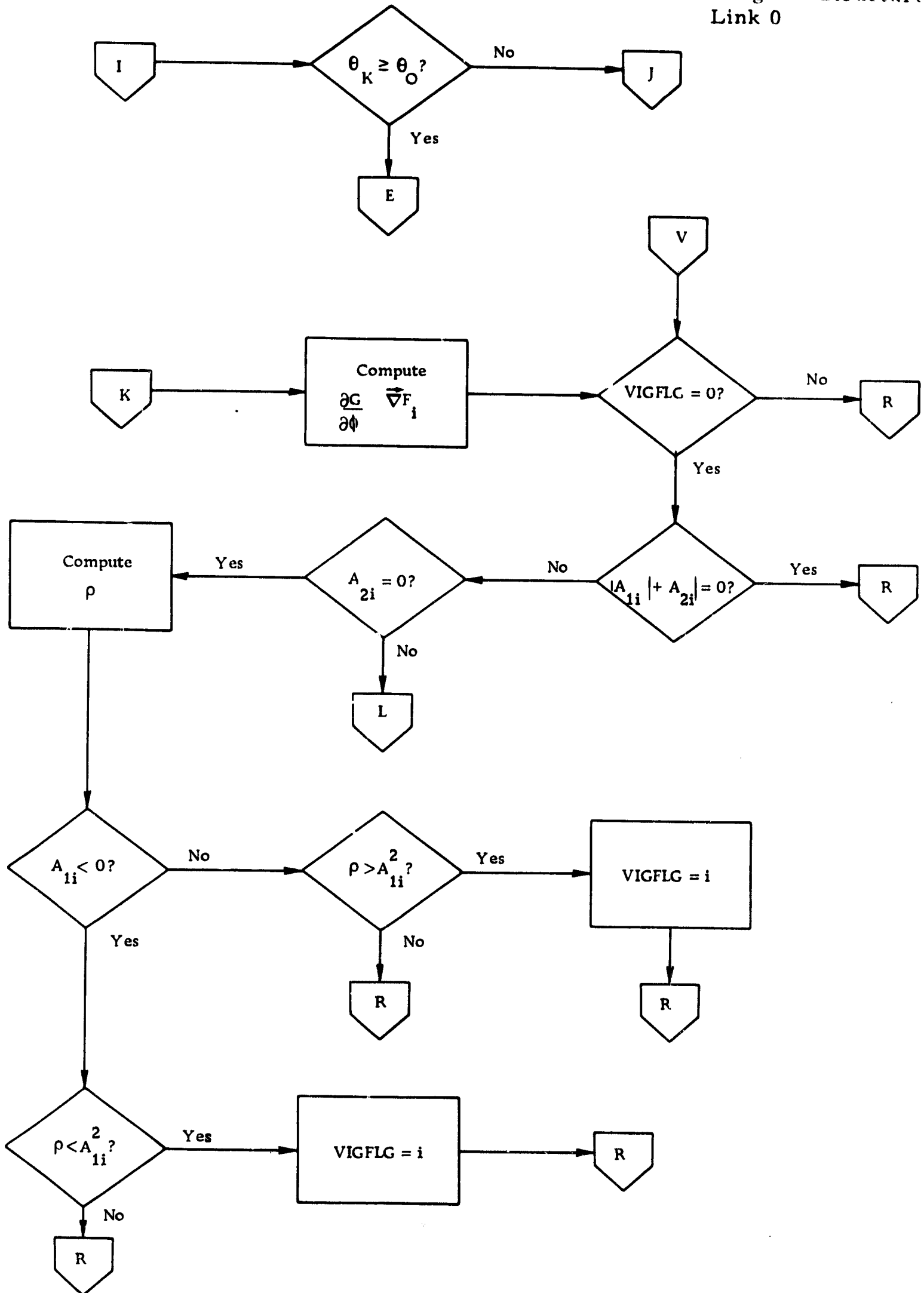
COMMON/DATA/
COMMON/AZOBJ/
GOOF

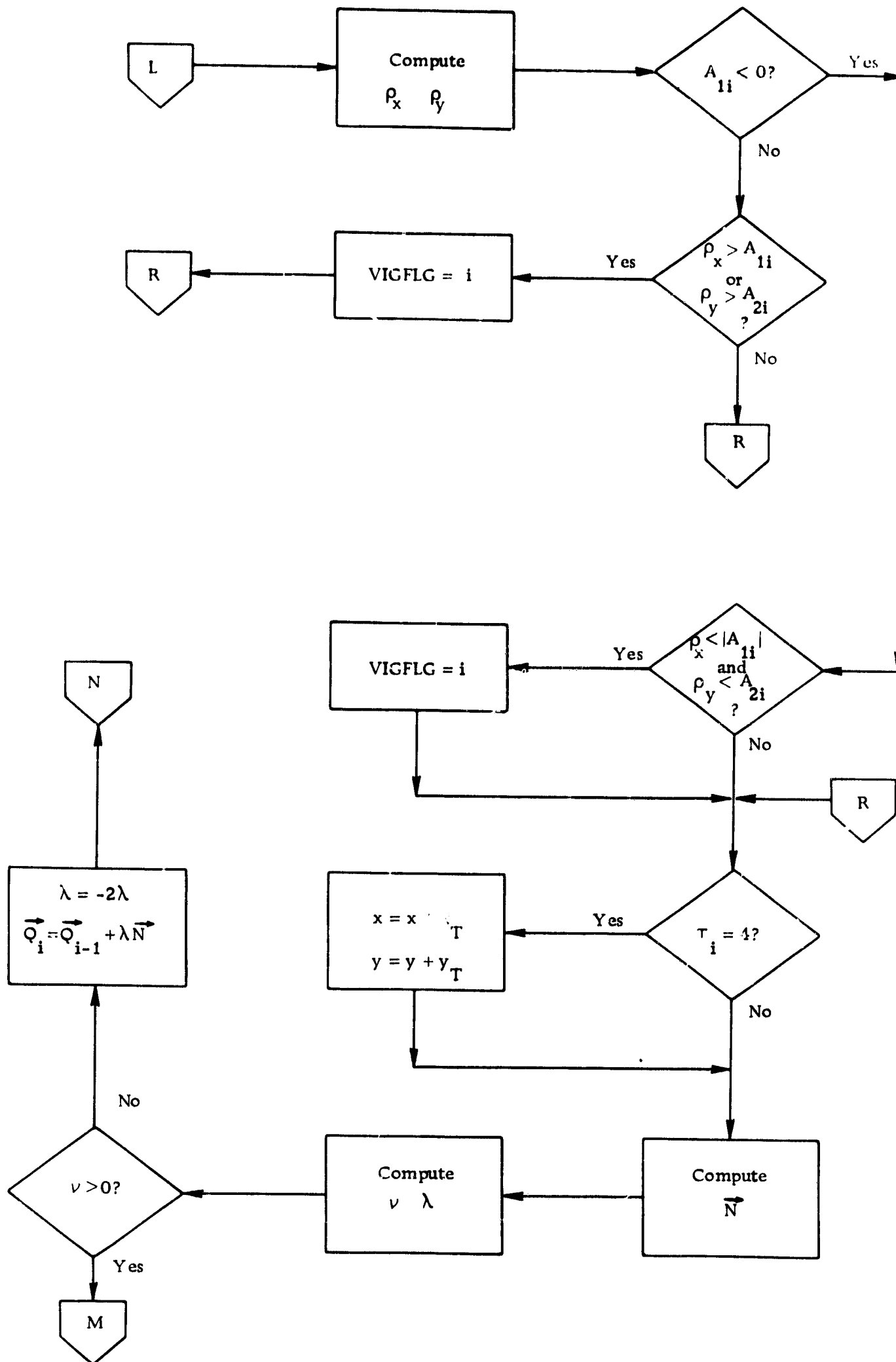


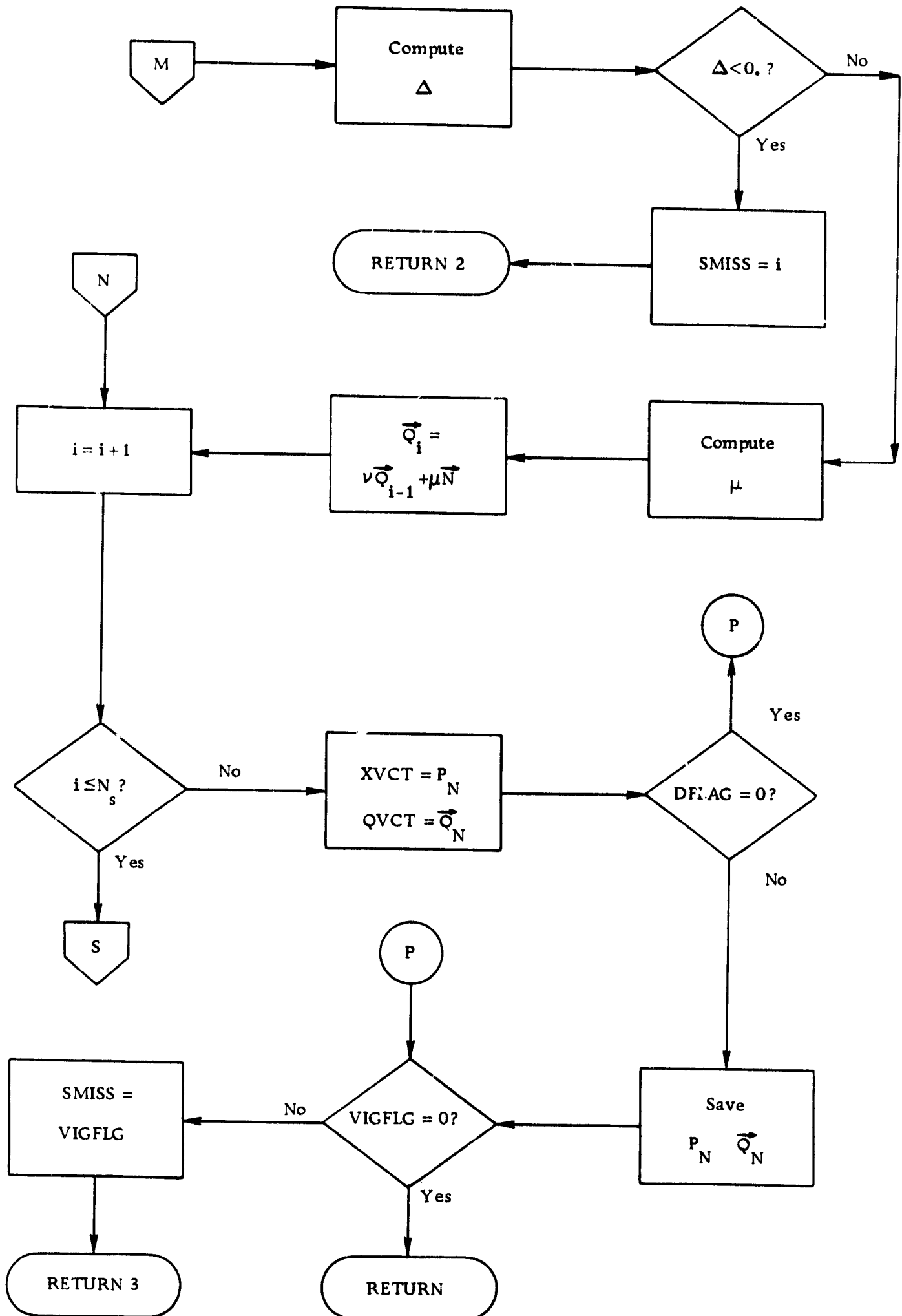












3.0.7 SCALEK (SCALEK)

SCALEK is a standard JPL library subroutine whose purpose and usage are described in the attached writeup. A flow chart is not included in this instance.

SCALEK/PLOTTER SCALING ROUTINE

C. L. Lawson, Jet Propulsion Laboratory, 1965 July 13
Program language: FORTRAN IV for IBM 7094

Purpose To compute four numbers which define a "pleasant" grid for plotting a variable for which upper and lower bounds are known.

Method The numbers C,D, KMAJØR and KMINØR are chosen to minimize D-C subject to the following conditions:

1. $C \leq A \leq D, C \leq B \leq D, C < D.$
2. KMAJØR = 3,4,5,6,7,8,9, or 10.
3. KMINØR = 2,5, or 10.
4. The major scale unit, $u = (D-C)/KMAJØR$, is to be of the form $KMINØR * 10^p$ for some integer p.
5. If zero lies in the closed interval between A and B, then both C and D will be integer multiples of u so that zero will lie on a major grid line. Otherwise, C and D may differ from a multiple of u by a multiple of 10^p .

If $A = B = 0$ the algorithm proceeds as though $A = -1., B = 1.$
If $A = B \neq 0$ the algorithm proceeds as though $A = .99*A, B = 1.01*B$

Usage CALL SCALEK(A,B,C,D,KMAJØR,KMINØR)

The numbers A and B are supplied by the user and represent bounds on some variable, say x, which the user desires to plot. Either $A \leq B$ or $A \geq B$ is permissible. SCALEK will compute numbers C,D,KMAJØR, and KMINØR with $C < D$ such that the interval $C \leq x \leq D$ subdivided into KMAJØR major subintervals each subdivided into KMINØR minor subintervals provides a "pleasant" grid for the variable x.

Example The following example illustrates the use of SCALEK along with appropriate JPL plotting subroutines to produce a "pleasant" grid with labels on the major grid lines:

```
REAL BUF(290),XYGRID(4),SCGRID(4)
DATA SCGRID/110.,940.,1020.,30./,NBUF/290/
XMAX = ...
XMIN = ...
YMAX = ...
YMIN = ...
(over)
```

```
CALL SCALEK(XMIN,XMAX,XYGRID(1),XYGRID(3),KMAJX,KMINX)
CALL SCALEK(YMIN,YMAX,XYGRID(2),XYGRID(4),KMAJY,KMINY)
CALL SDINIT(BUF,NBUF,XYGRID,SCGRID,18)
CALL BGRID(BUF,1000*KMAJX + KMINX*KMAJX,1000*KMAJY +
KMINY*KMAJY,1)
CALL PLAB(BUF,KMAJX,6H(F9.4),KMAJY,6H(F9.4))
```

Efficiency of Graph Space Used The fraction of available graph space used is given by the ratio $r = |B-A|/(D-C)$. In a sample of twenty-five instances of the use of SCALEK in actual applications this ratio was between .9 and 1. eleven times, between .8 and .9 eleven times and between .7 and .8 three times. It is conjectured that the lowest possible value for r is .67 as long as $|B-A| > 10^{-7} * \max(|A|, |B|)$.

Subroutines Used ALØG10,EXP

Storage 5318 = 345₁₀

3.1 LINK 1

Link 1 is the input processor link and contains modules which read and validate the numerical input and list all or a subset of the data region. Link 1 is entered from FOLDP whenever a new case is to be processed.

3.1.1 OPUNCZ (OPUNCH)

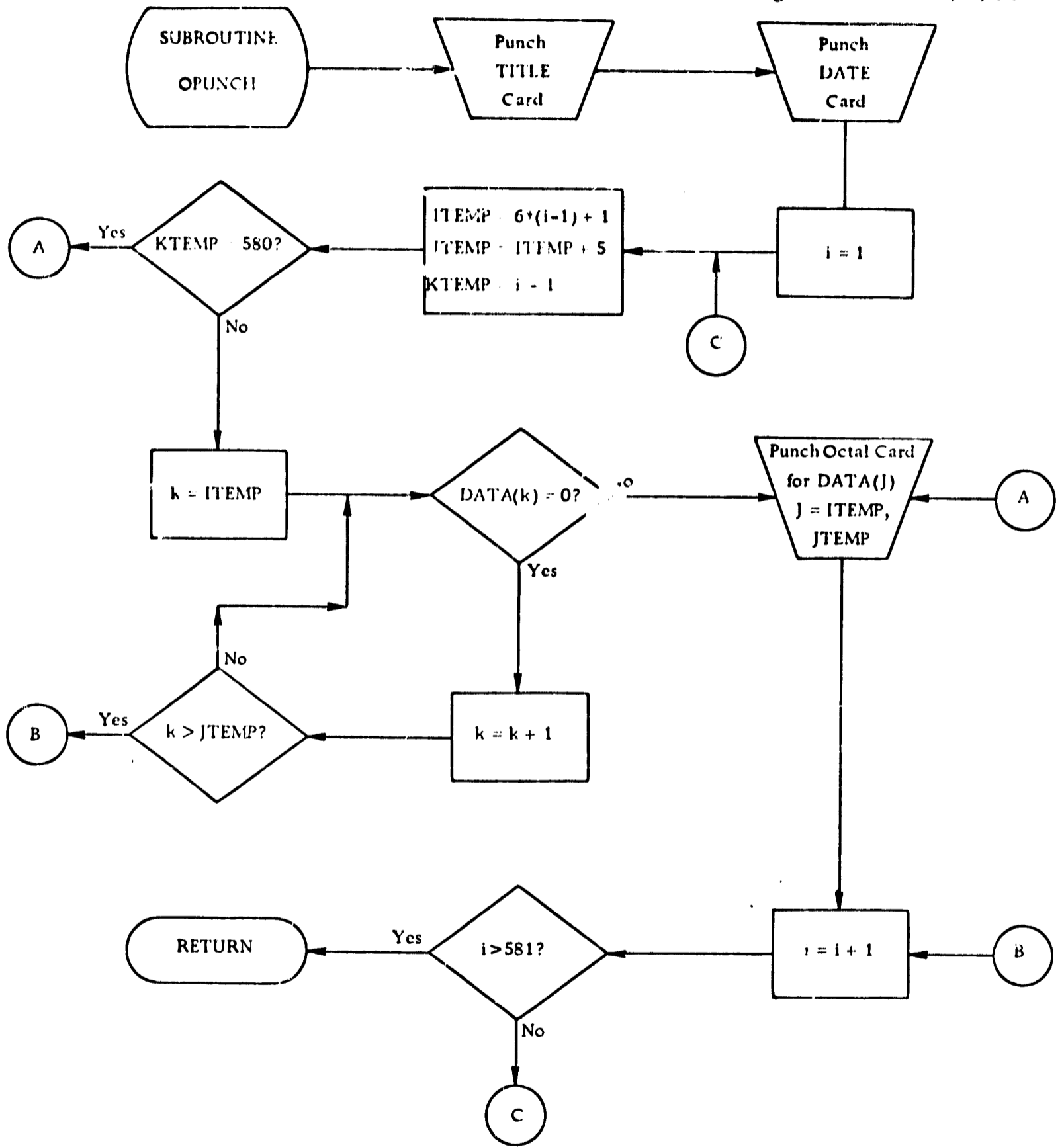
OPUNCZ punches a sequenced deck of not more than 583 cards containing the non-zero members of the DATA vector. The first two cards contain identification information and should be discarded. The remaining cards are punched in a 6012 format. Octal card 580 is always punched, and serves as an end of case card for the octal deck.

Calling Sequence

CALL OPUNCH

Utility Routines and Common References

COMMON/DATA/
Utility - None



3.1.2 READSZ (READS)

READSZ processes the input data cards for a single case.

The following functions are performed:

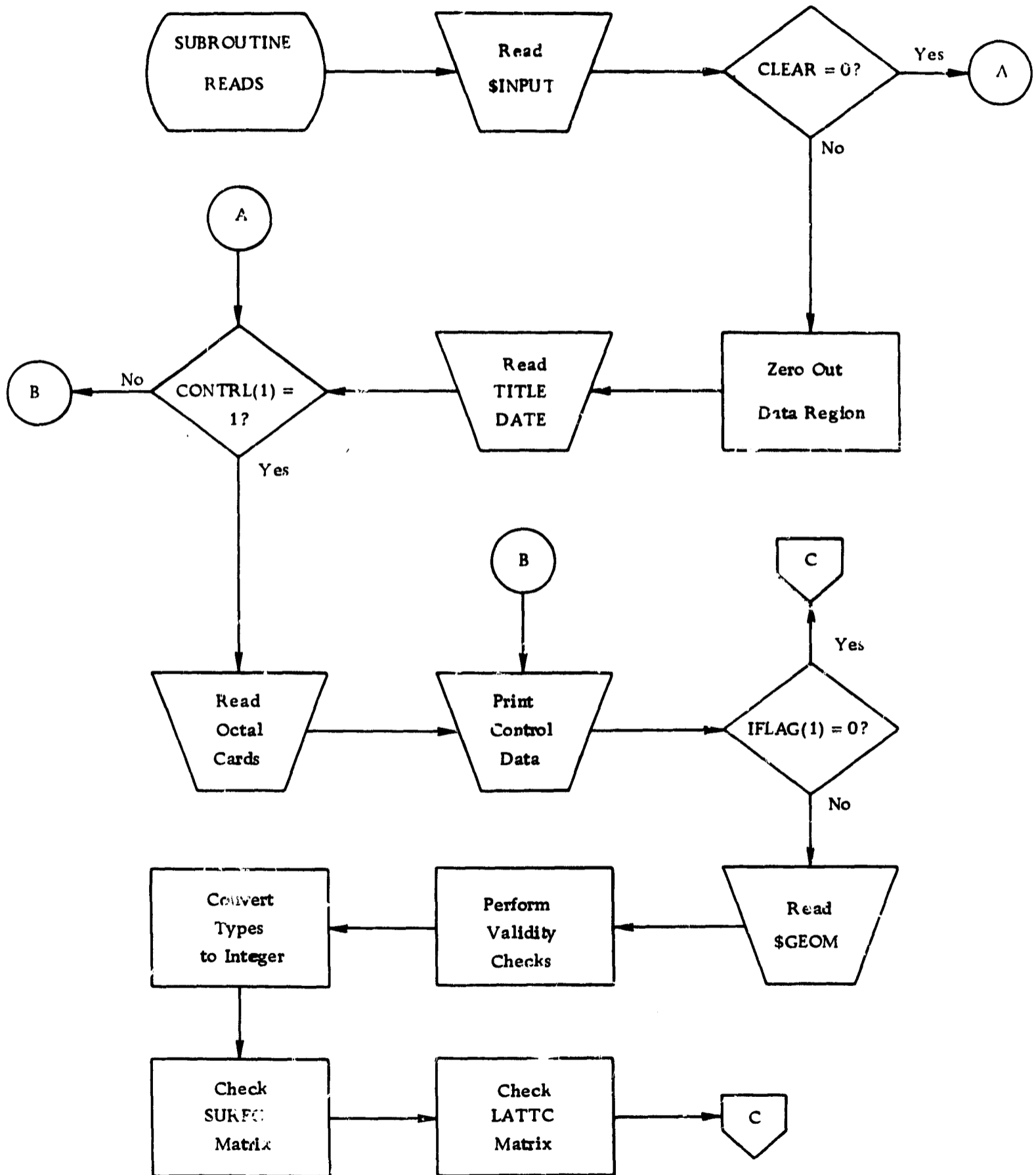
- a) Reads any combination of NAMELIST data including \$INPUT, \$GEOM, \$DESIGN, \$BNDRY, \$SUBSTN, \$SENST and \$CRPLOT.
- b) Executes a limited number of validity checks and issues error messages.
- c) Performs a number of type conversions from real to integer.
- d) Processes symbol cards by calculating and preserving the DATA index for each valid symbol which is read.

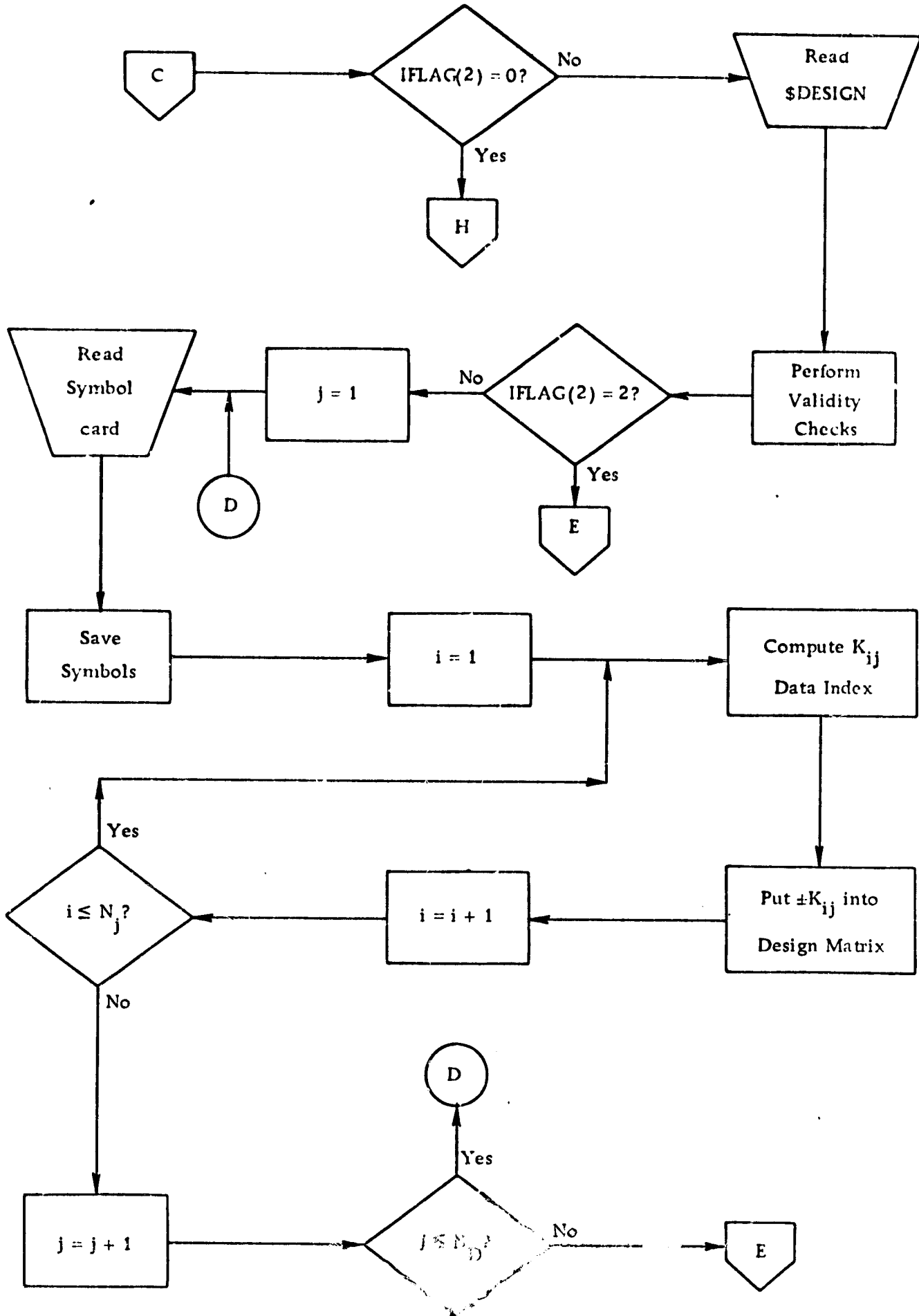
Calling Sequence

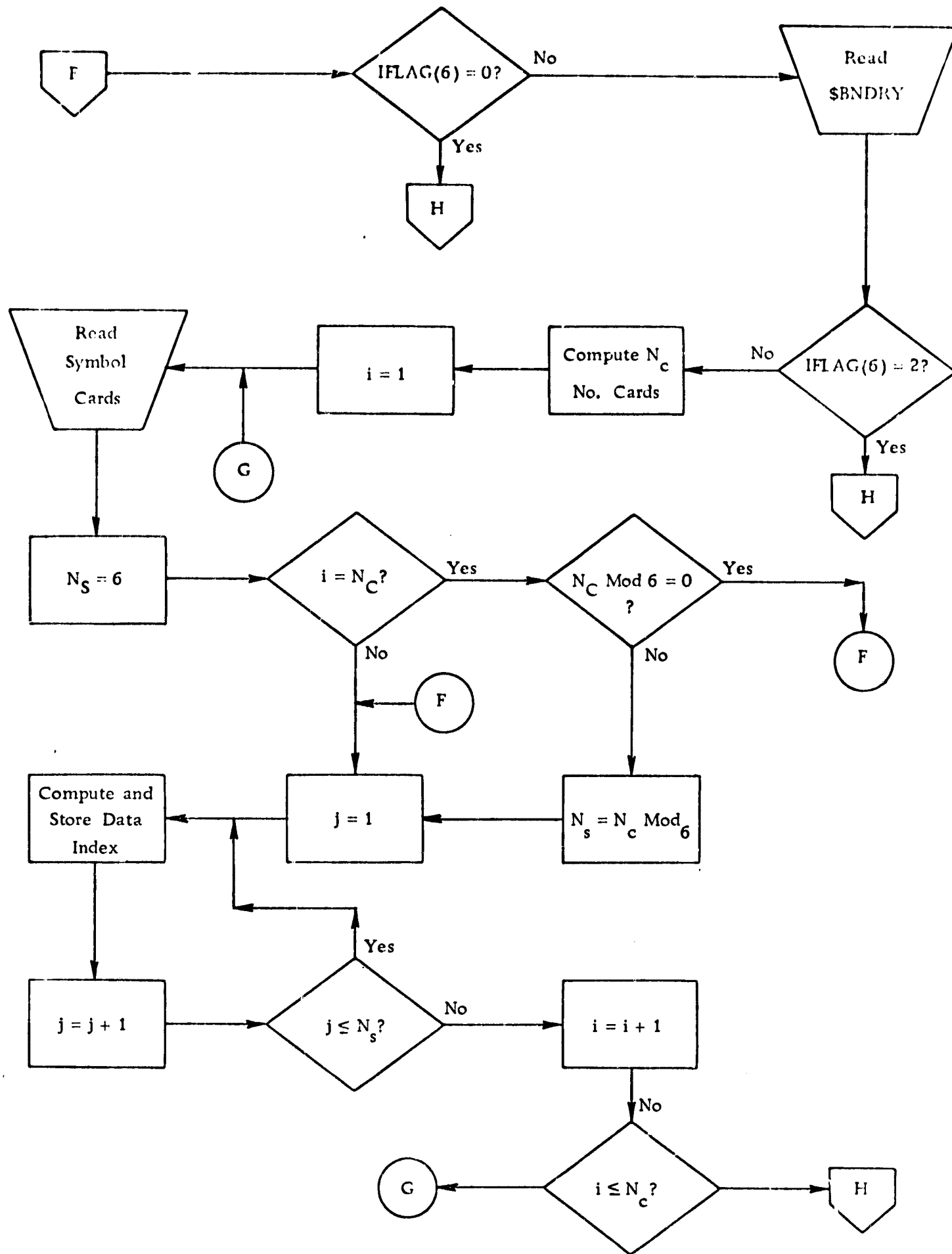
CALL READS

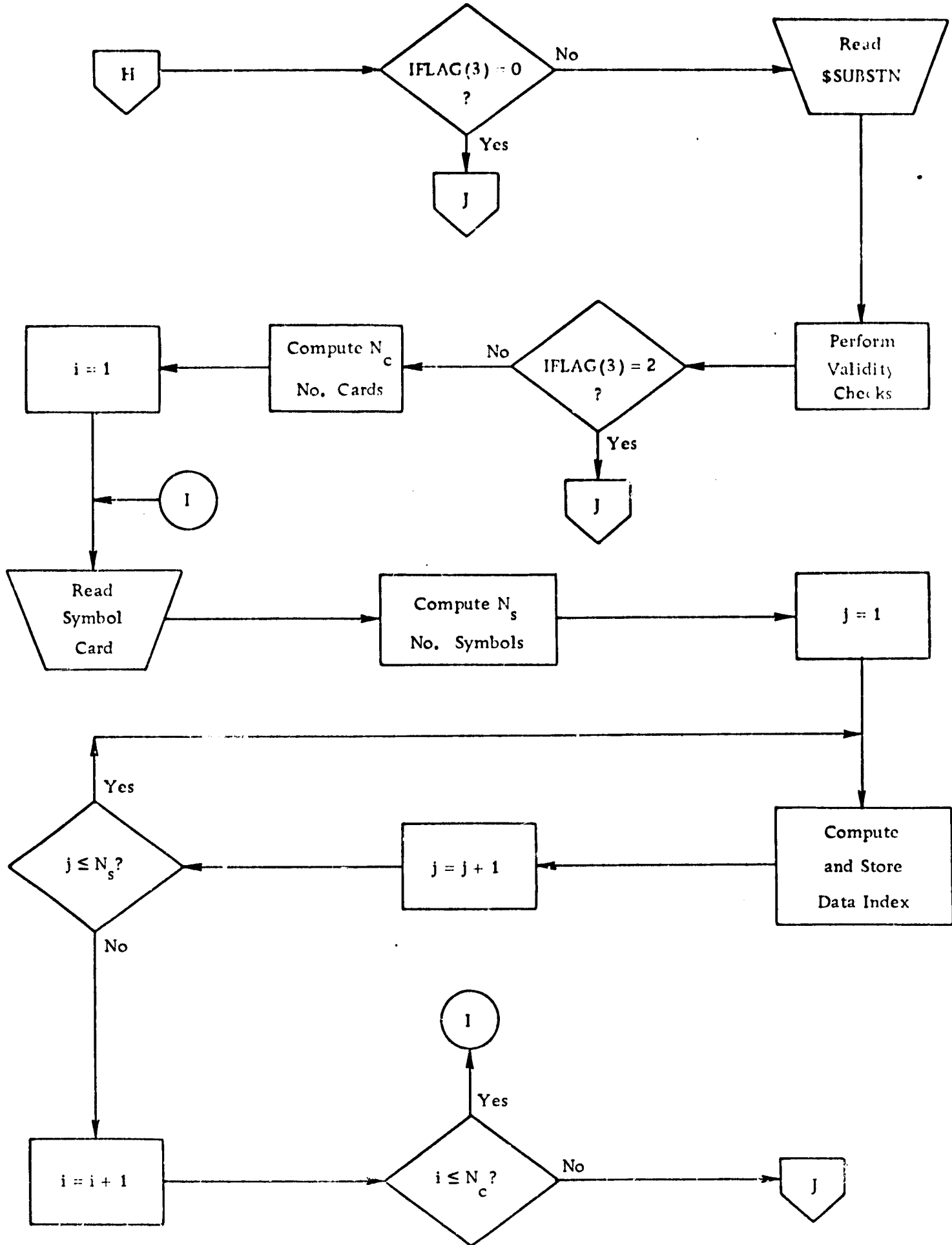
Utility Routines and Common References

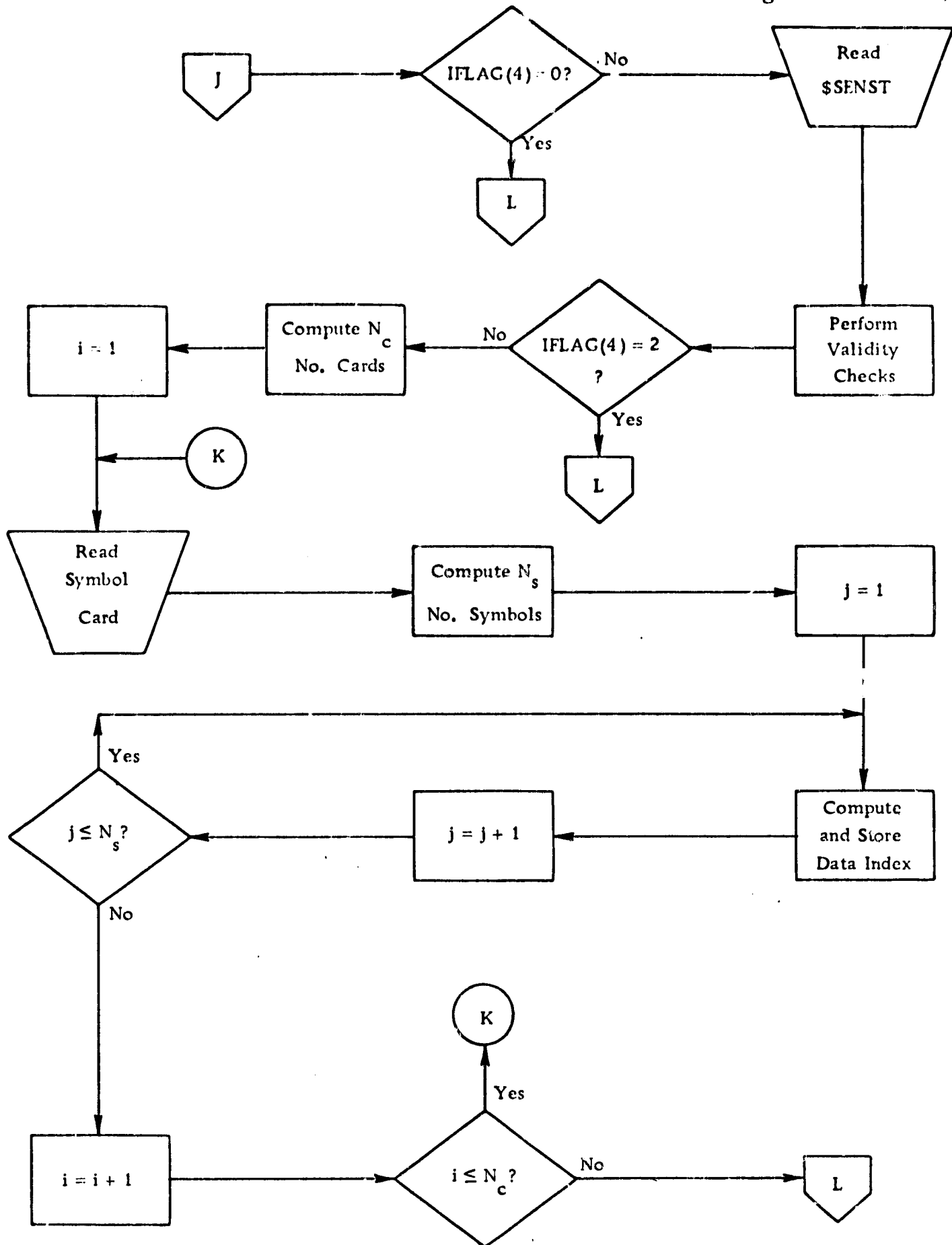
COMMON/DATA/	VCNVRT
COMMON/SYMBLS/	ERROR
COMMON/PRNT/	ERROR1
COMMON/PERTB/	ERROR2
COMMON/FLOTC/	ERROR3
PRTCTL	ERROR4
MCNVRT	

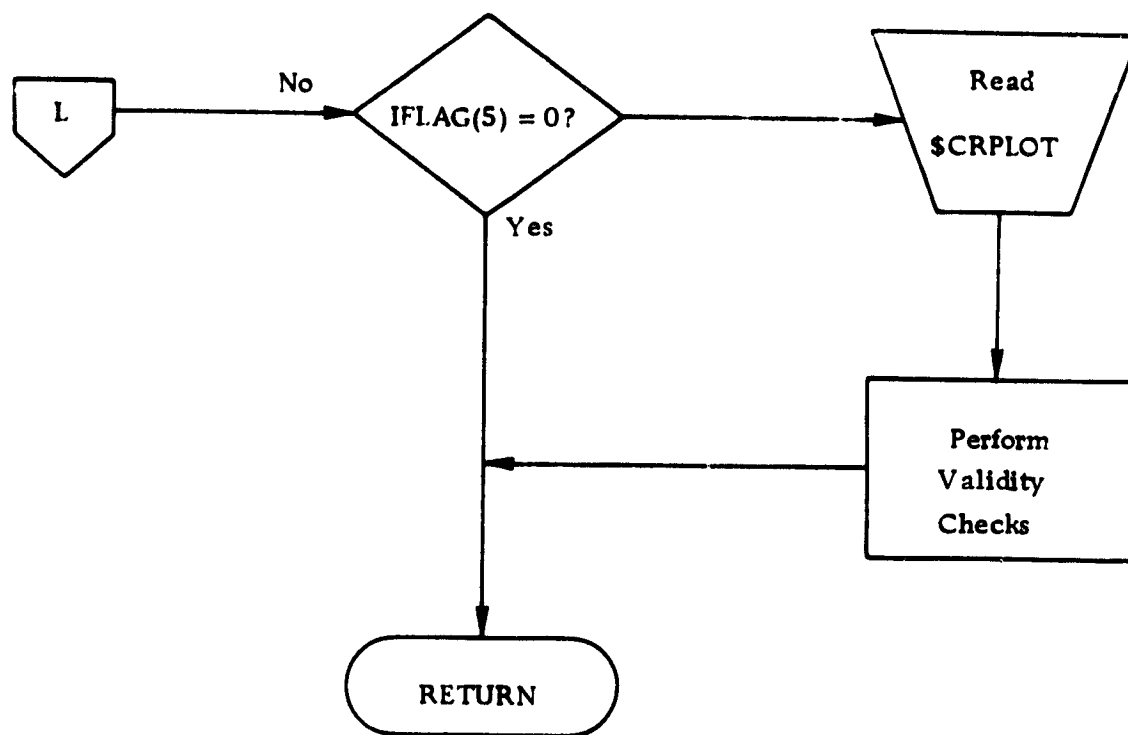












3.1.3 REREZZ (REREAD)

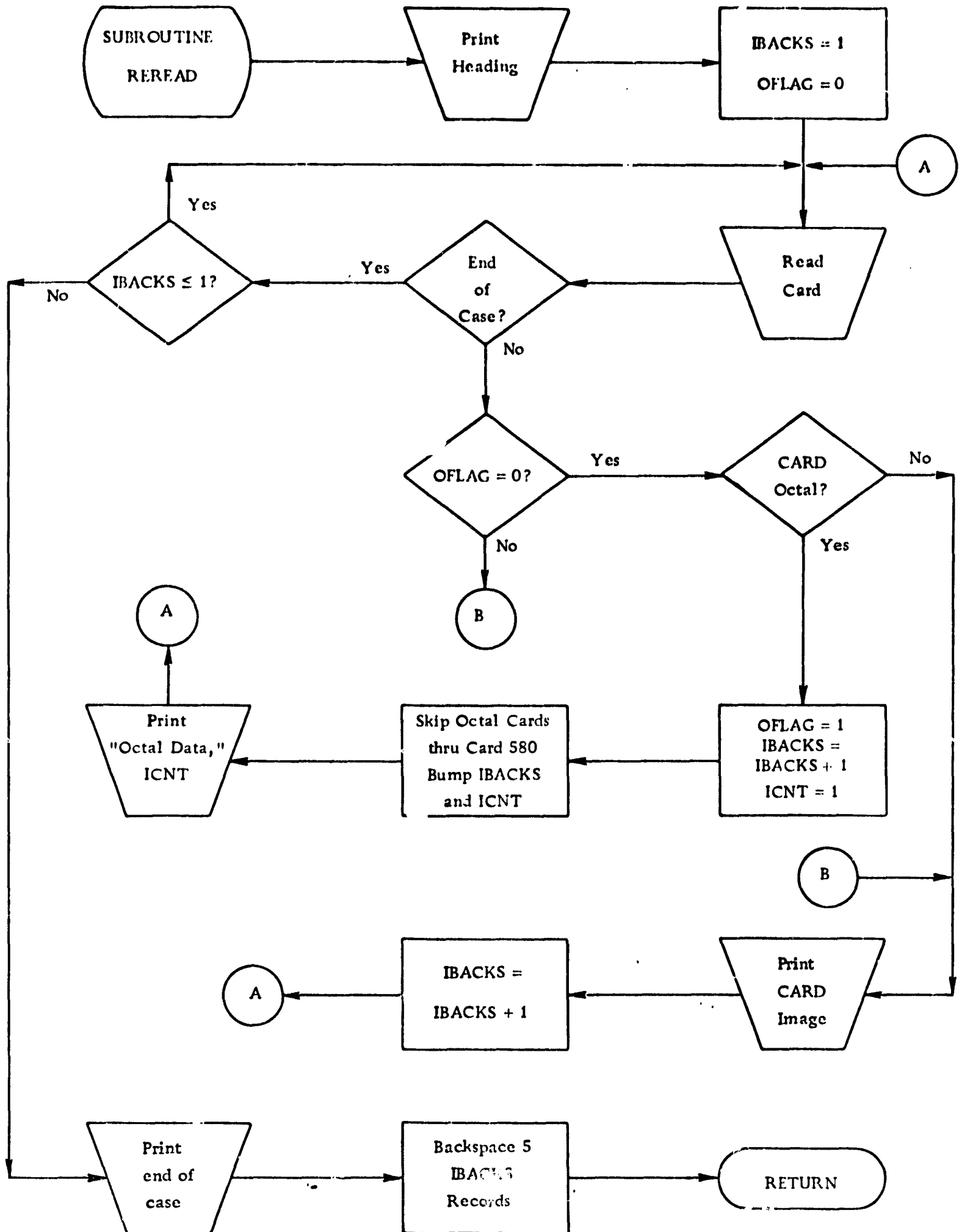
REREZZ reads card images from the system input file (logical 5) in an 80A1 format until an END OF CASE card is encountered. Each card (except octal) which is read is listed on the system output file. A count of the number of cards is kept so that REREZZ can reposition the input file by executing the appropriate number of BACKSPACE commands. If octal input is encountered, the octal cards are not listed, but the number of cards in the octal deck is printed. The purpose of REREZZ is to produce a card image list of the current case deck.

Calling Sequence

CALL REREAD

Utility Routines and Common References

None



3.1.4 STAR9Z (STAR9)

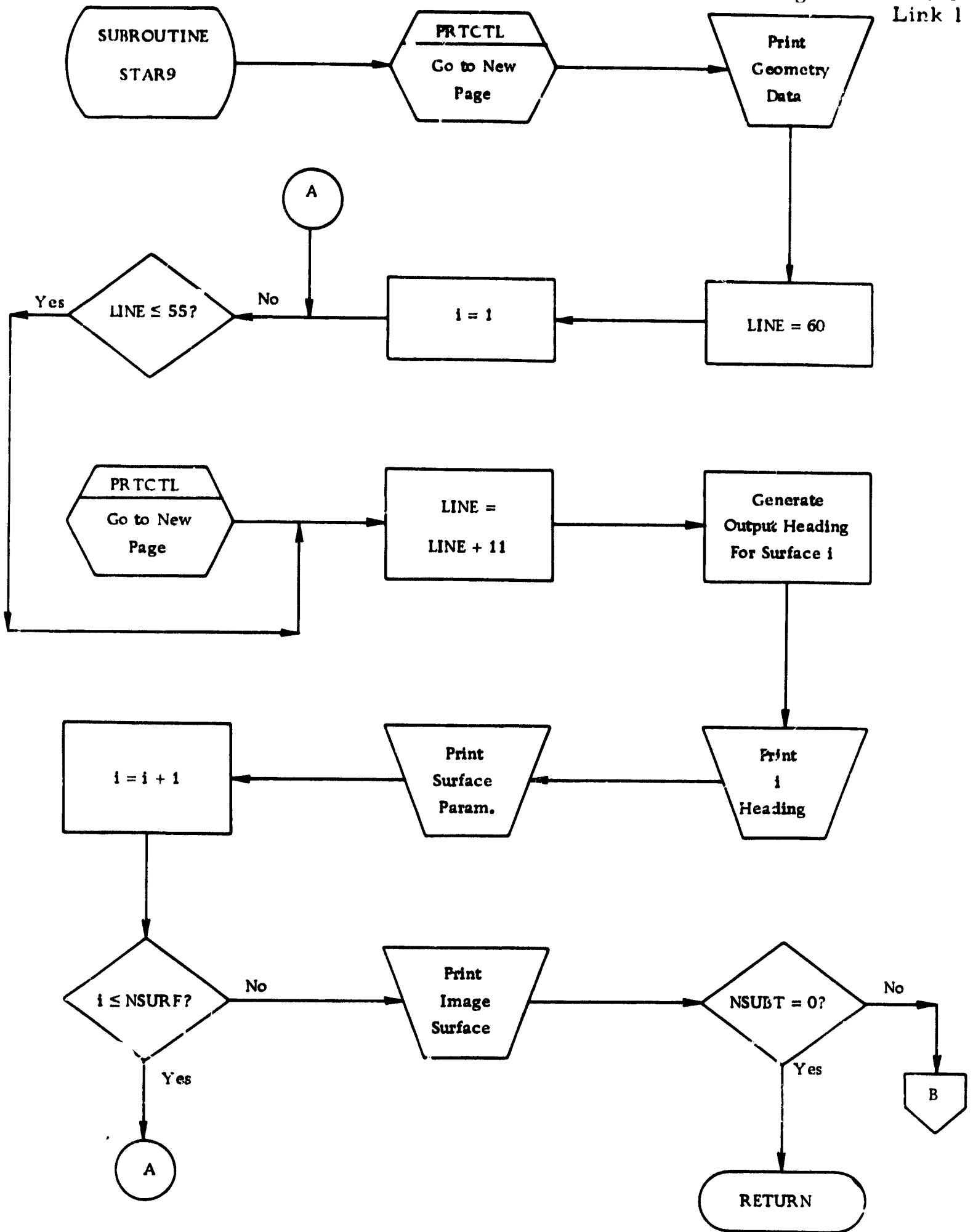
STAR9Z lists all of the geometry data with descriptive headings and if NSUBT \neq 0 it also lists the substitution data by parameter and by set.

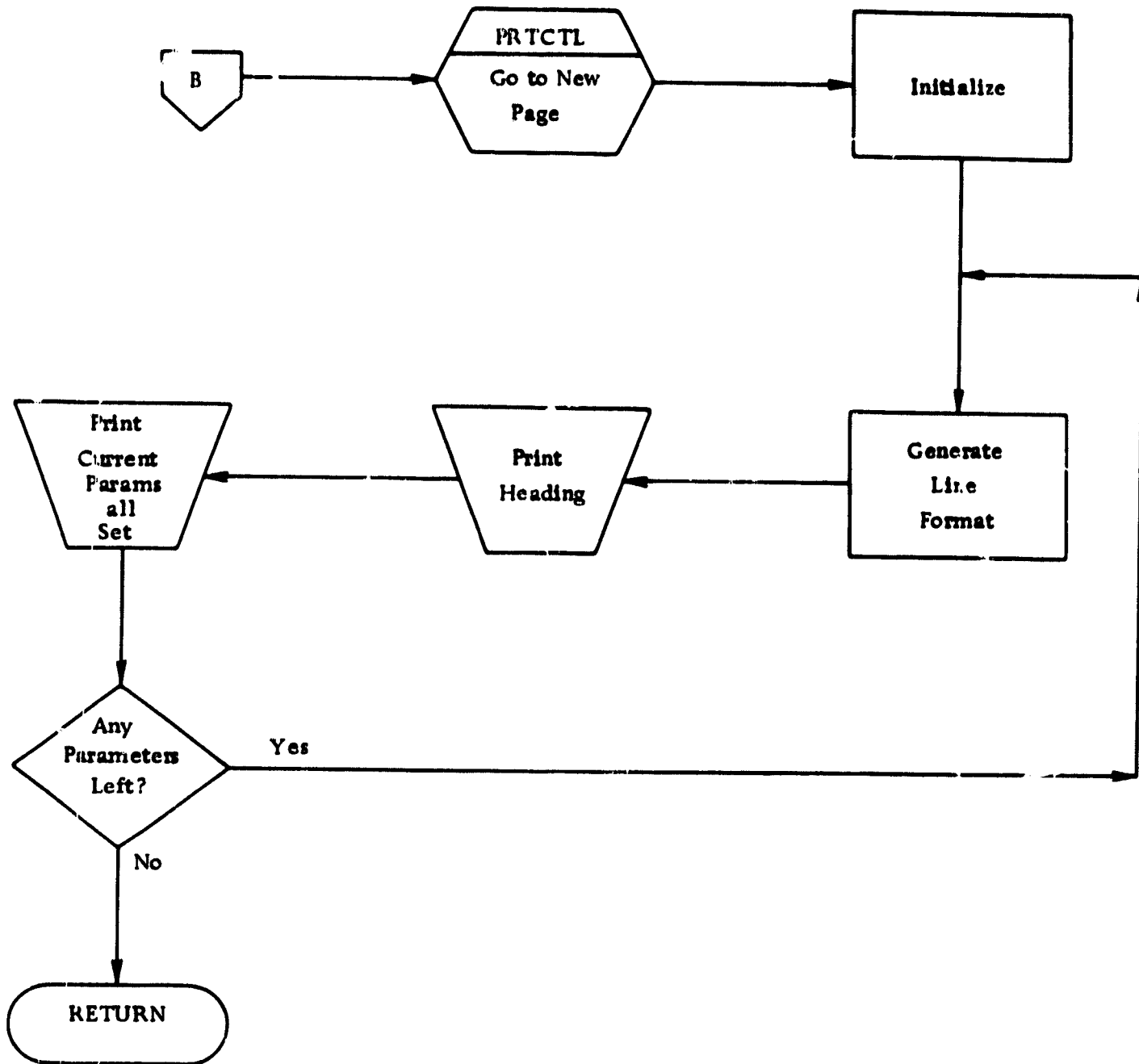
Calling Sequence

CALL STAR9

Utility Routines and Common References

COMMON/DATA/
COMMON/SYMBLS/
COMMON/PRNT/
PRTCTL
ERROR2





3.1.5 STAR1Z (STAR10)

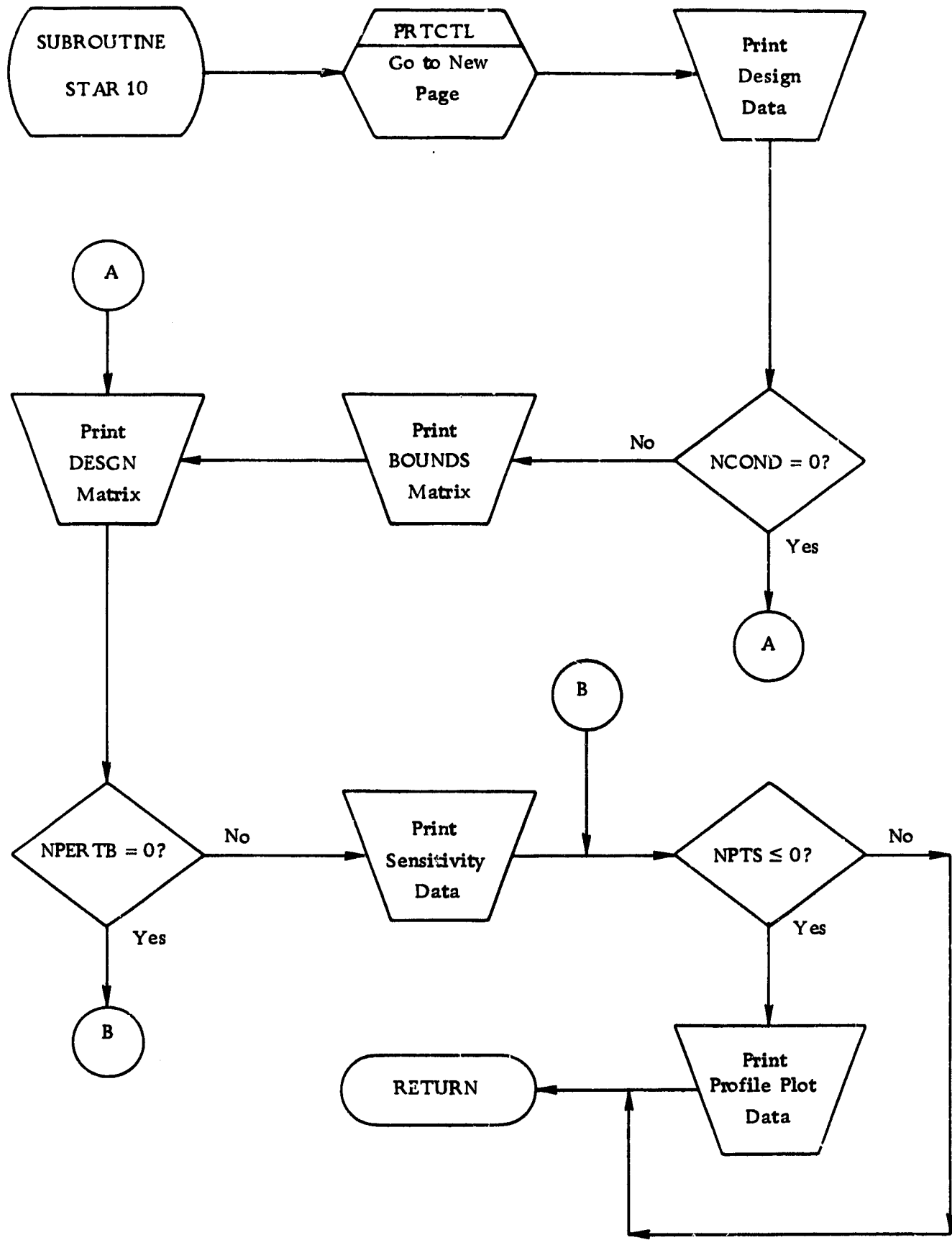
STAR1Z lists all of the design data with descriptive headings as well as the boundary condition matrix (if NCOND > 0) sensitivity data (if NPERTB > 0), and profile plot data (if NPTS > 0).

Calling Sequence

CALL STAR10

Utility Routines and Common References

COMMON/DATA/
COMMON/SYMBLS/
COMMON/PRNT/
COMMON/PERTB/
COMMON/PLOTCL/
PRTCTL



TR-67-700-10-2

Page III-3-41

Program Structure

Link 1

Page Revised 10/1/68

3.1.6 MTFPNZ (MTFPN)

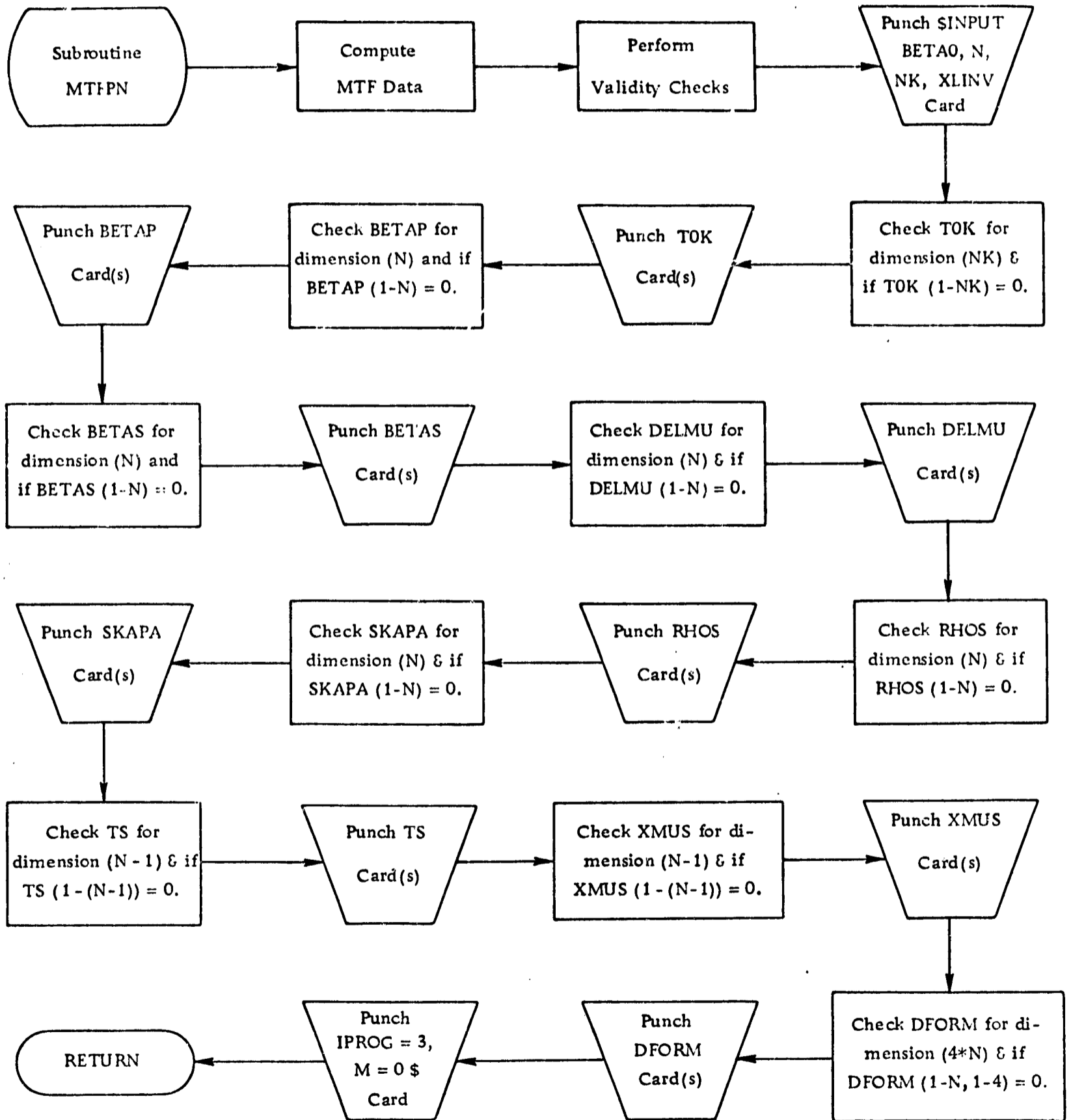
MTFPNZ punches a deck of cards in NAMELIST format which, when input to optical system analysis program PAGOS, provides data necessary for the Modulation Transfer Function (MTF) of that program.

Calling Sequence

CALL MTFPN

Utility Routines and Common References

COMMON/DATA/
ERROR2



3.1.7 UNBCD. (UNPBCD)

UNBCD. uses the DECODE/ENCODE capability of Fortran to unpack a 6 character word into 6 words of 1 character each.

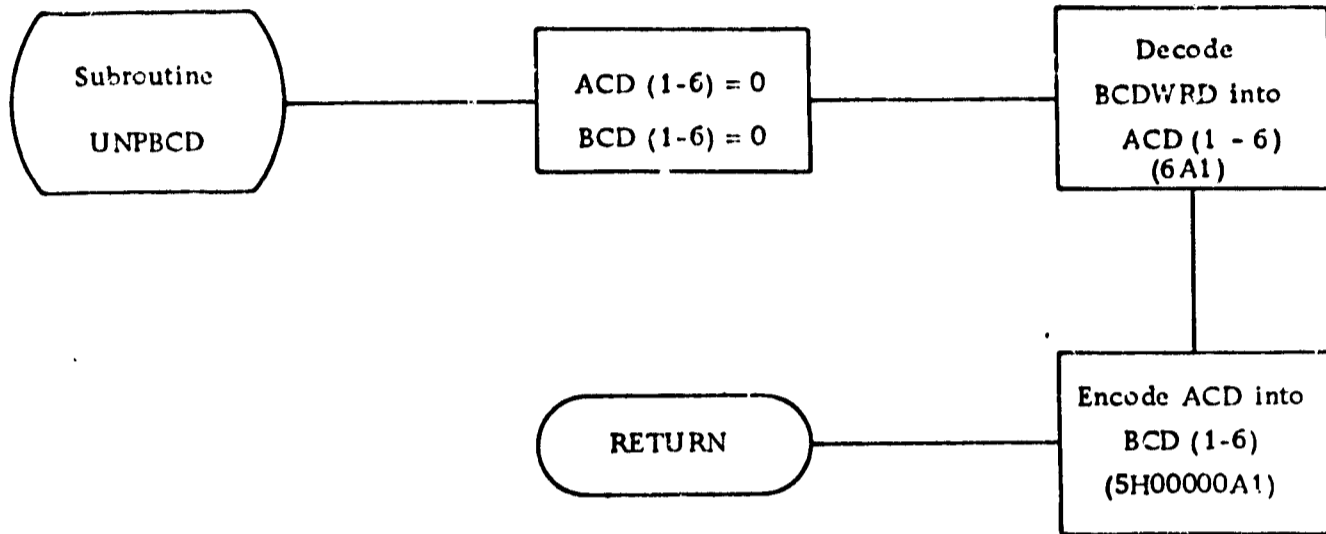
Calling Sequence

CALL UNPBCD (BCDWRD, BCD)

where BCDWRD contains the word to be unpacked and BCD is the first of 6 words for storing the 6 characters. The characters are right justified with leading zeros.

Utility Routines and Common References

None



3.1.8 VCNVR, (VCNVRT, MCVNVRT)

VCNVR. is used by READS when processing symbol cards in order to convert vector and matrix subscripts from BCD to integer so that the corresponding DATA index can be determined.

Calling Sequence

CALL VCNVRT (BCDWRD, ISUB, \$ERR)

where BCDWRD contains a BCD vector subscript of the form X)bbbb, XX)bbb, or XXX)bb and ISUB is the location where the integer subscript is to be stored. If VCNVRT detects an error the alternate return \$ERR is invoked.

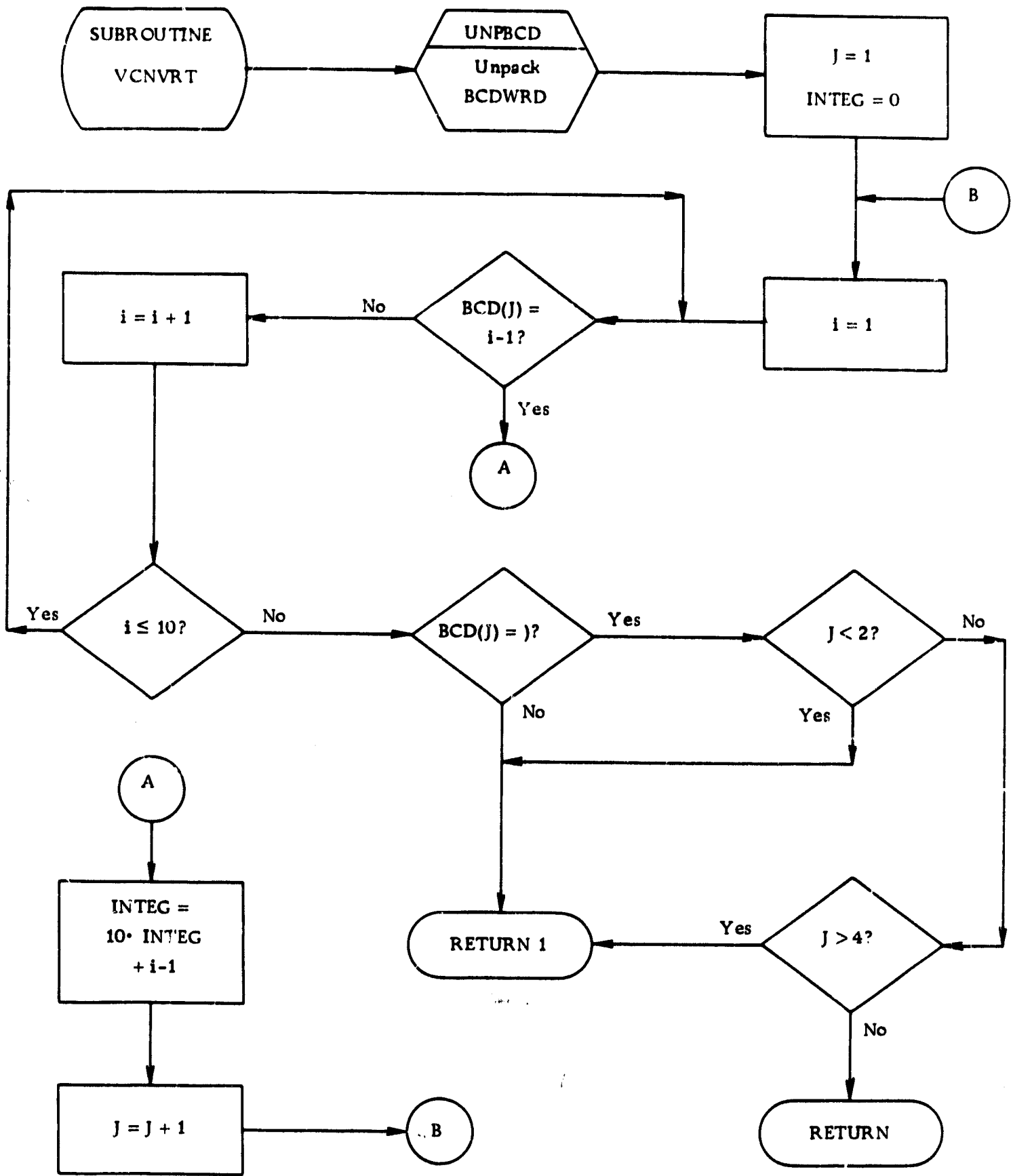
CALL MCVNVRT (BCDWRD, ISUB, JSUB, \$ERR)

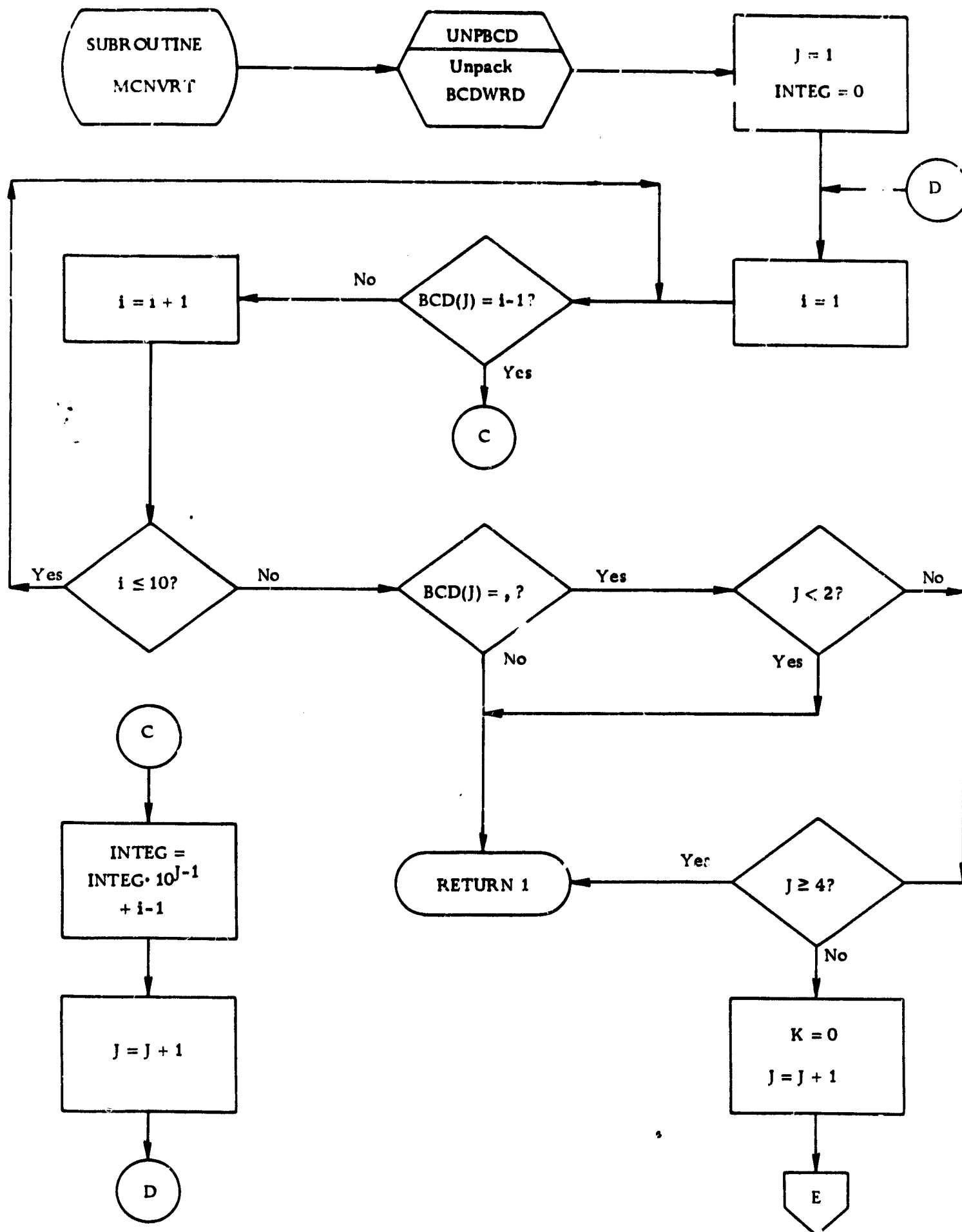
where BCDWRD contains a BCD matrix subscript of the form X, Y)bb, X, YY)b, XX, Y)b, or XX, YY), ISUB is the location where the integer row subscript is to be stored, and JSUB is the location where the integer column subscript is to be stored. The alternate return \$ERR is invoked if an error is detected.

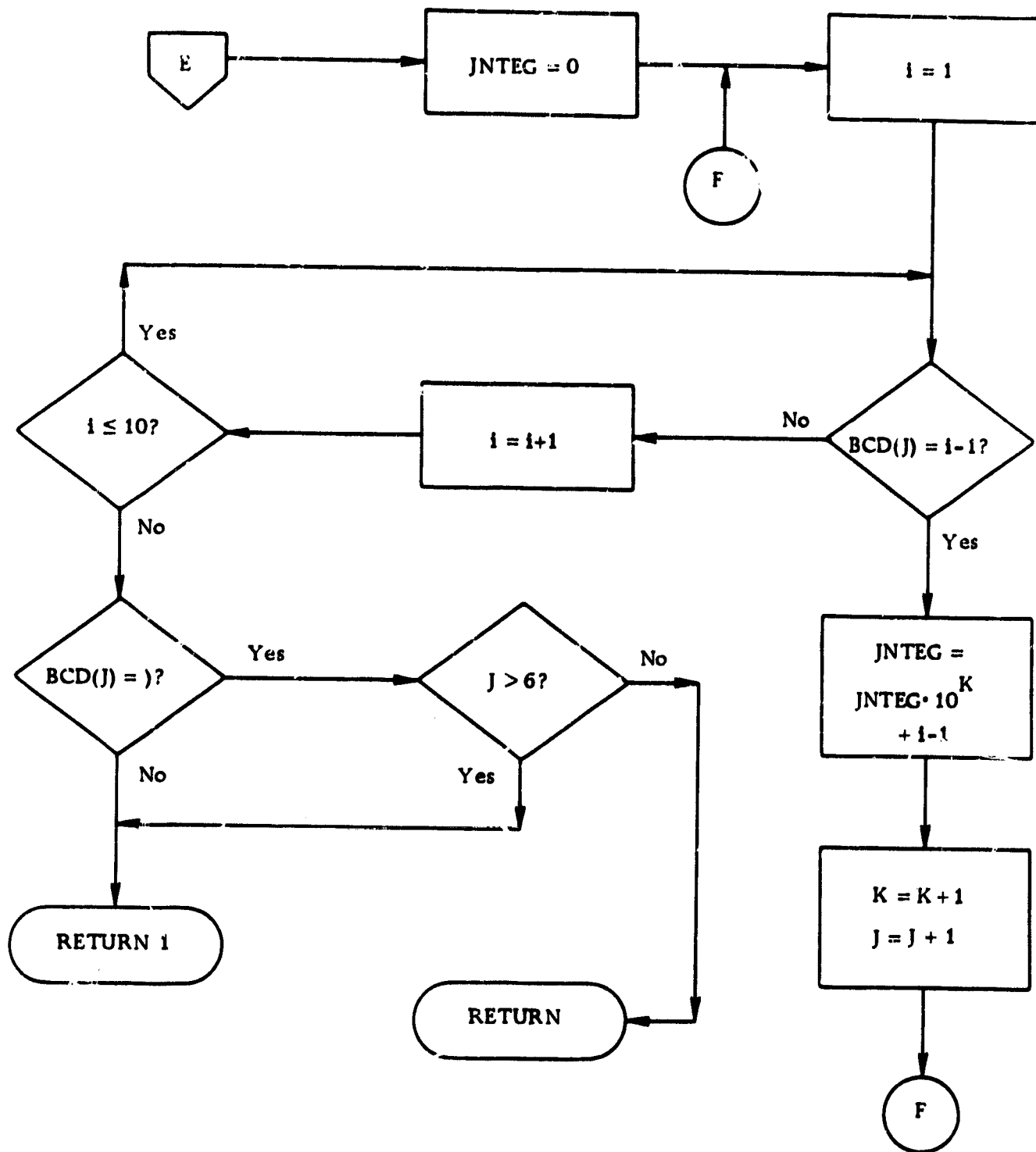
Utility Routines and Common References

COMMON - None

UNPBCD







3.2 LINK 2

Link 2 has the responsibility of producing a profile plot of the lens system on the SC4020 plotter. By definition a profile plot is a two dimensional picture which results when the system is "cut" by the z - y plane. Each element is represented as a (scaled) closed curve making it necessary to have the ability to find the point of intersection of two plane curves. Because of the problems entailed with translated, tilted, and aspheric elements the capability is currently limited to standard conics only.

All references to the plotter are contained in subroutine CPLOT so that adaptation to a different plotting device (such as CALCOMP) would involve a minimum of program alteration.

3.2.1 ARPTSZ (ARPTS)

ARPTSZ generates a set of N points which lie on a plane conic $y = f(z)$ for $z_A \leq z \leq z_B$ and are equally spaced in the z direction by an amount $z = (z_B - z_A)/(N - 1)$.

Calling Sequence

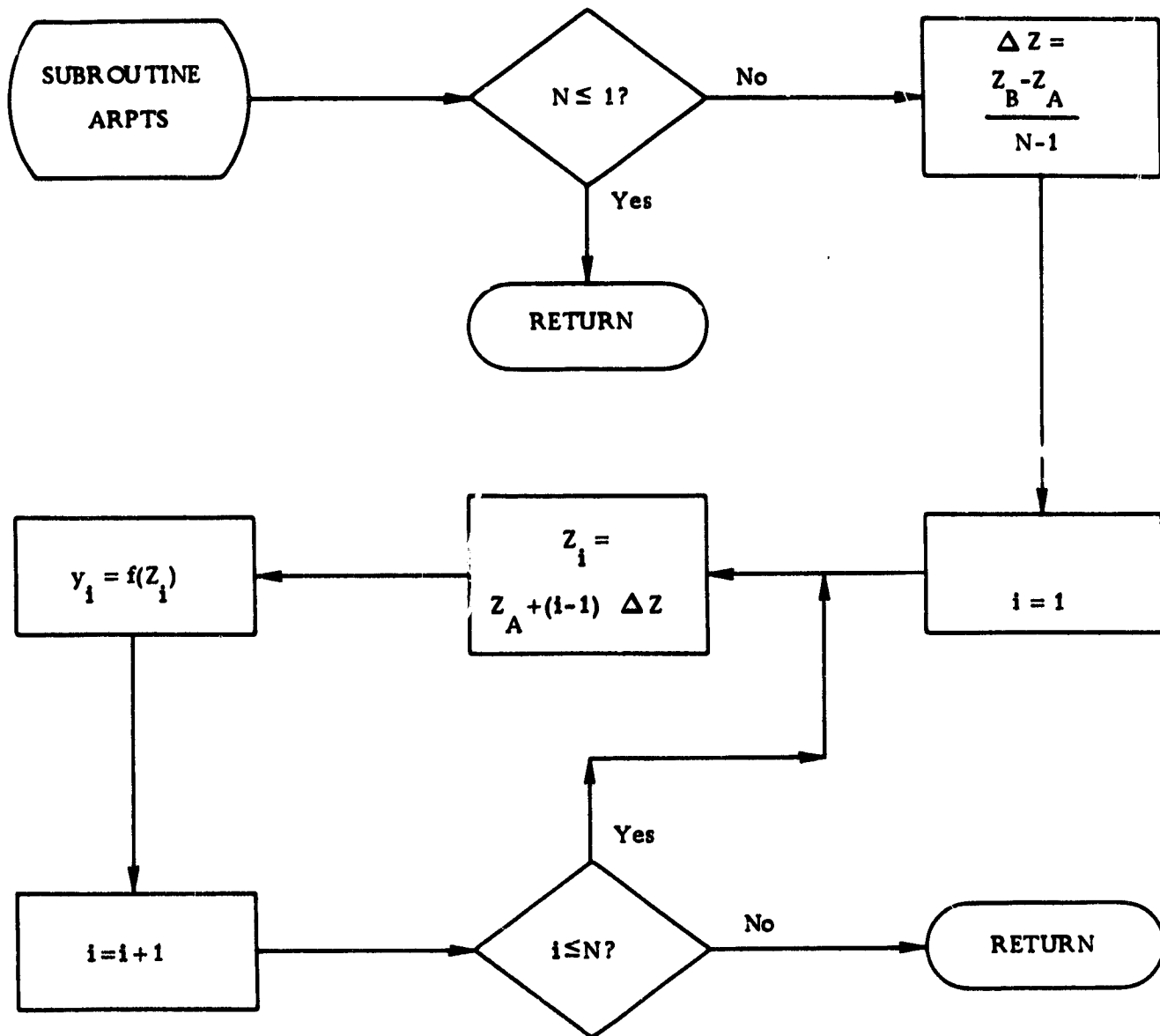
CALL ARPTS (IS, ZA, YA, ZB, YB, N, ZLST, YLST)

IS: The surface number of the conic
ZA: z_A
YA: $y_A = f(z_A)$
ZB: z_B
YB: $y_B = f(z_B)$
N: Number of points to insert in (z_A, z_B)
ZLST: Vector for storing z_i $1 \leq i \leq N$
YLST: Vector for storing y_i $1 \leq i \leq N$

Utility Routines and Common References

COMMON/DATA/

Utility - None



3.2.2 CELAIZ (CELAIR)

CELAIZ determines the vertex coordinate δ_i of each surface relative to the entrance pupil. It also determines the element matrix E and the air space matrix A which are edited by a process which effectively deletes any repeated element or air space.

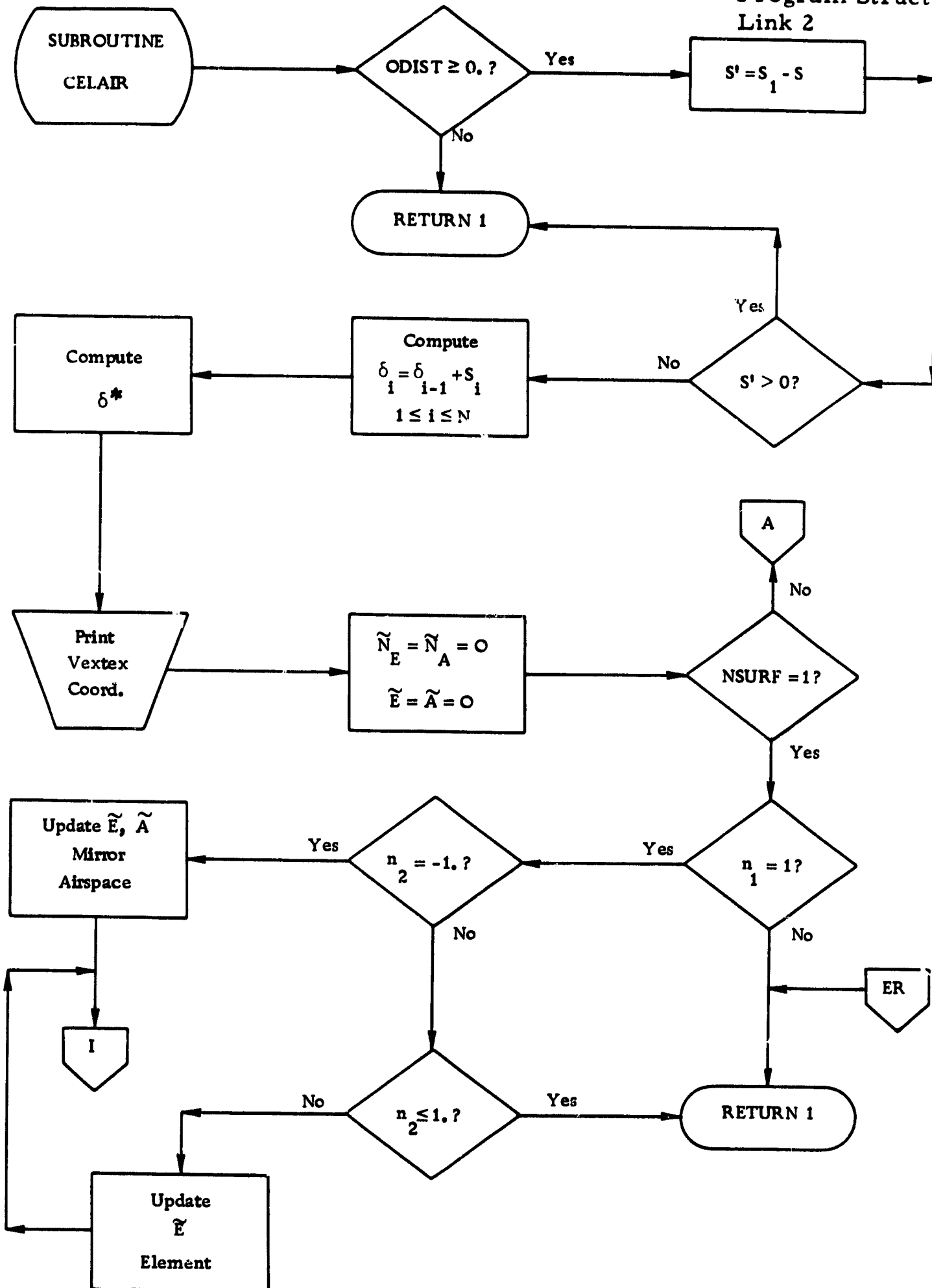
Calling Sequence

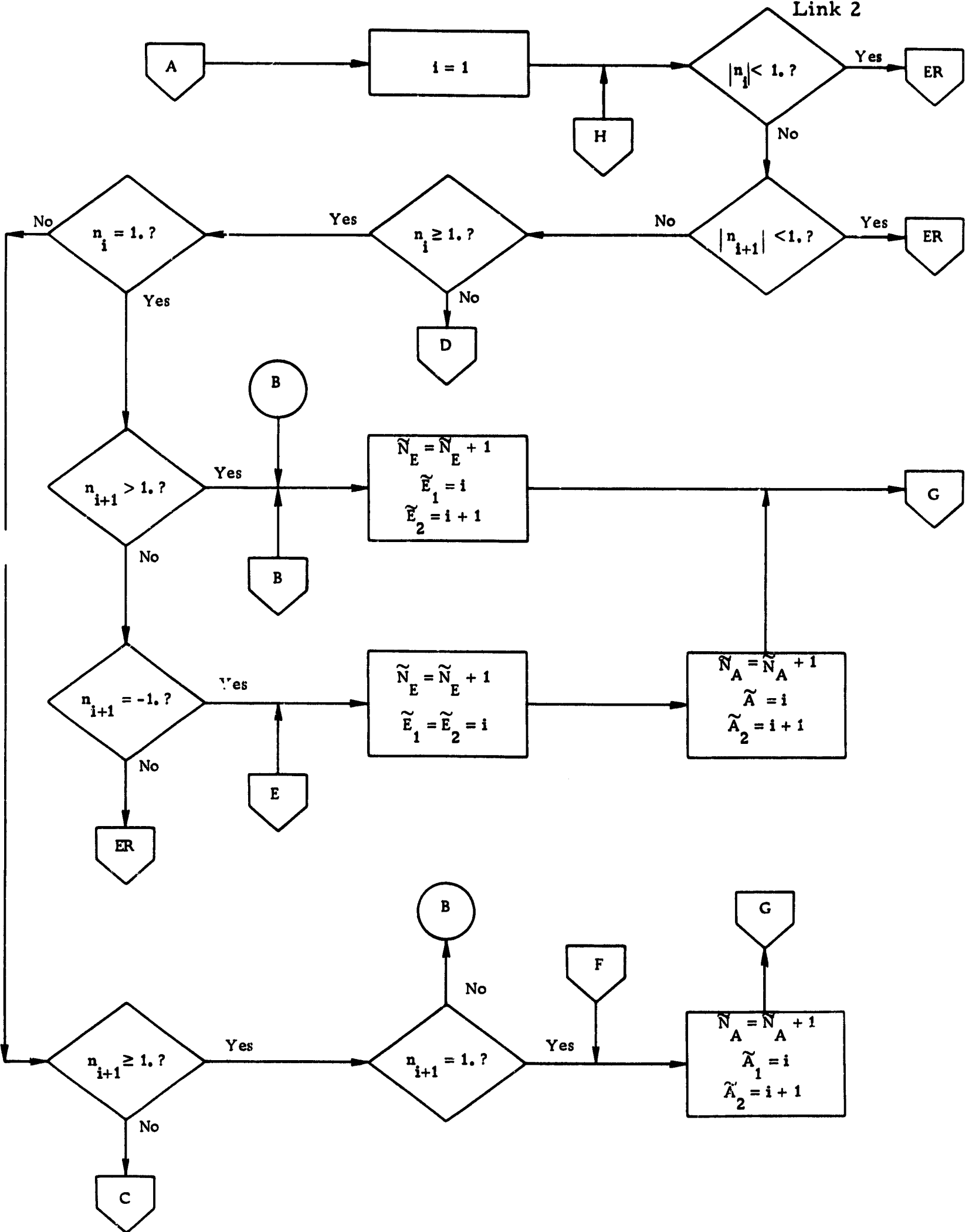
CALL CELAIR(\$ALT)

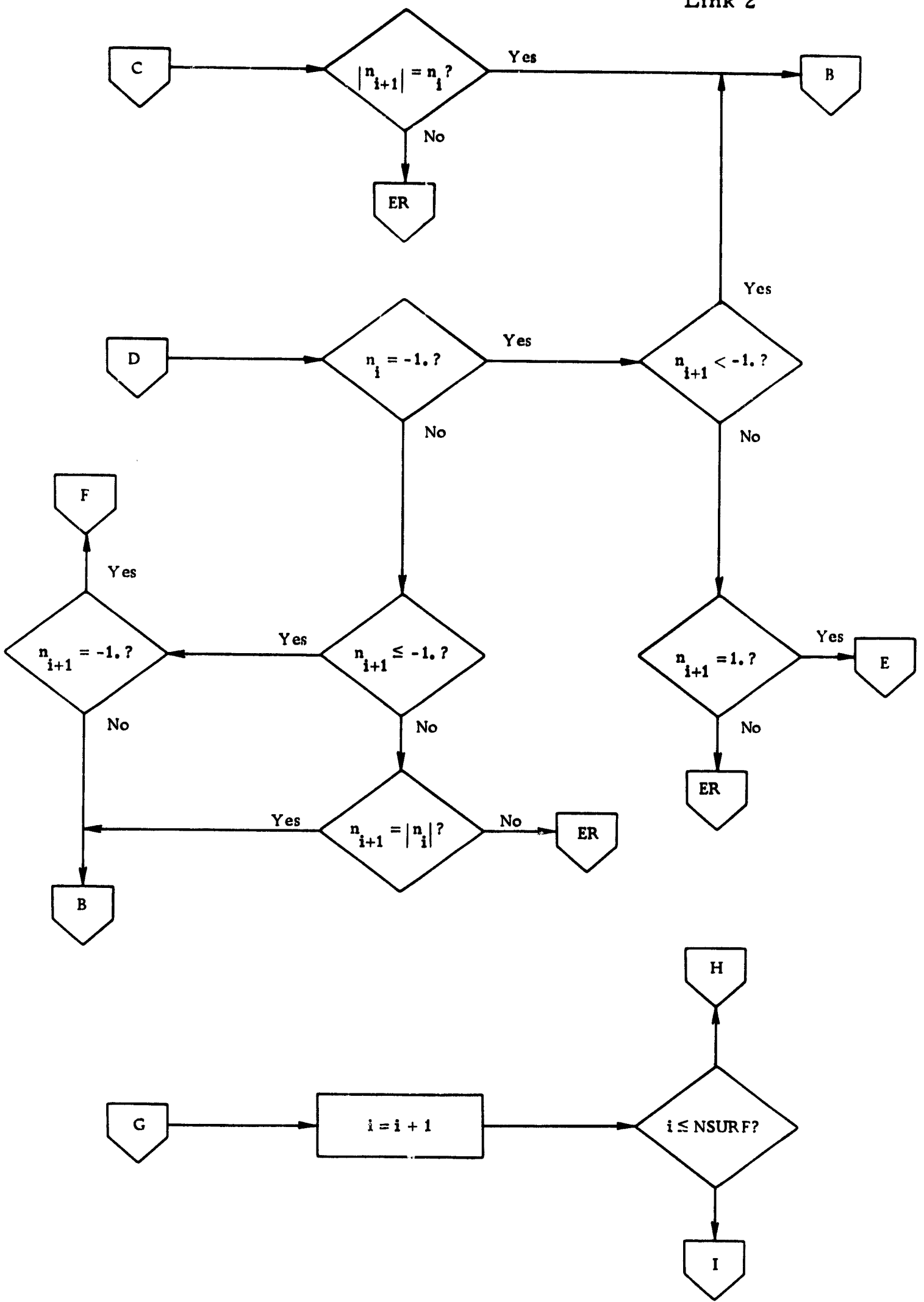
The alternate return is invoked if CELAIR determines that a profile plot cannot be produced.

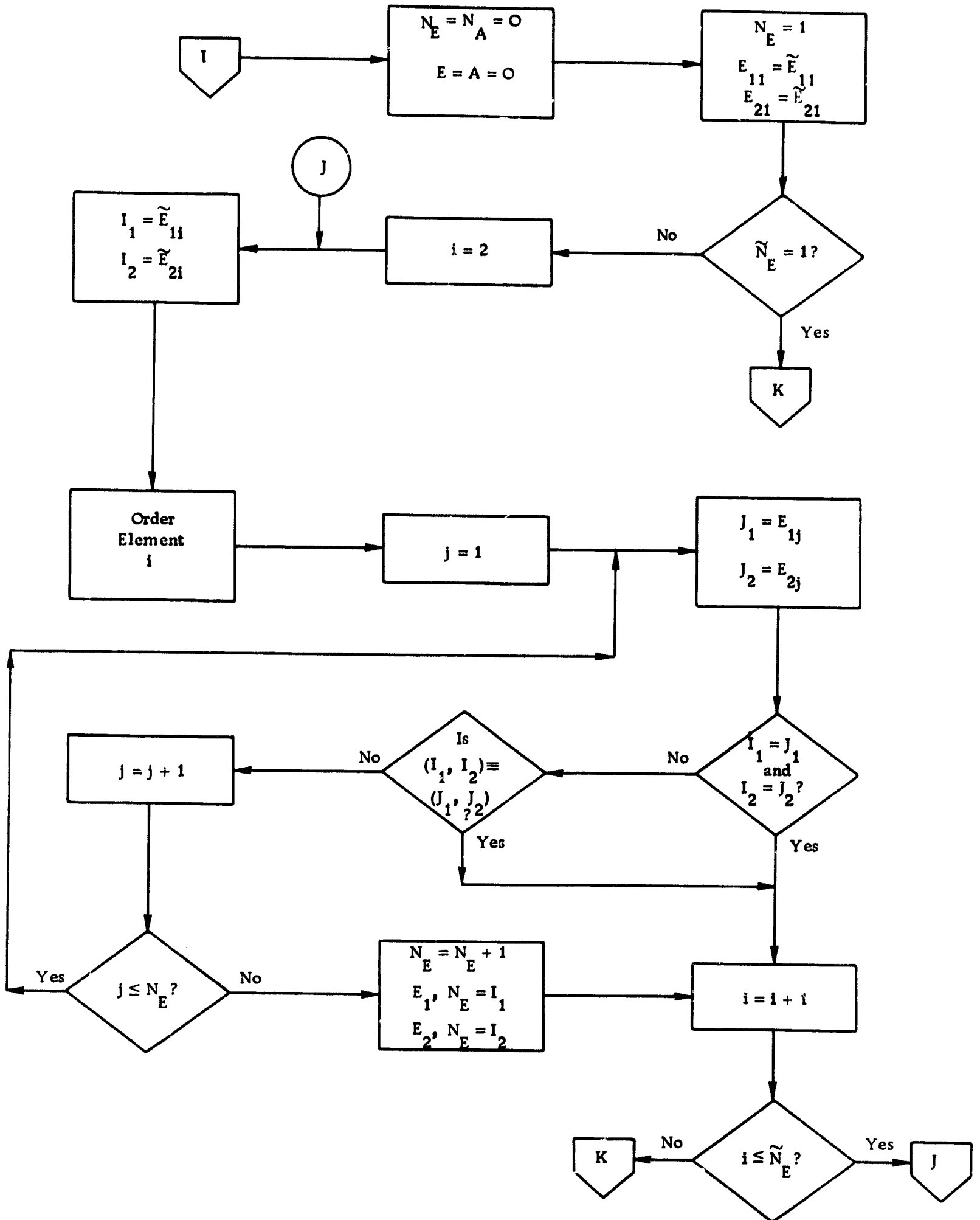
Utility Routines and Common References

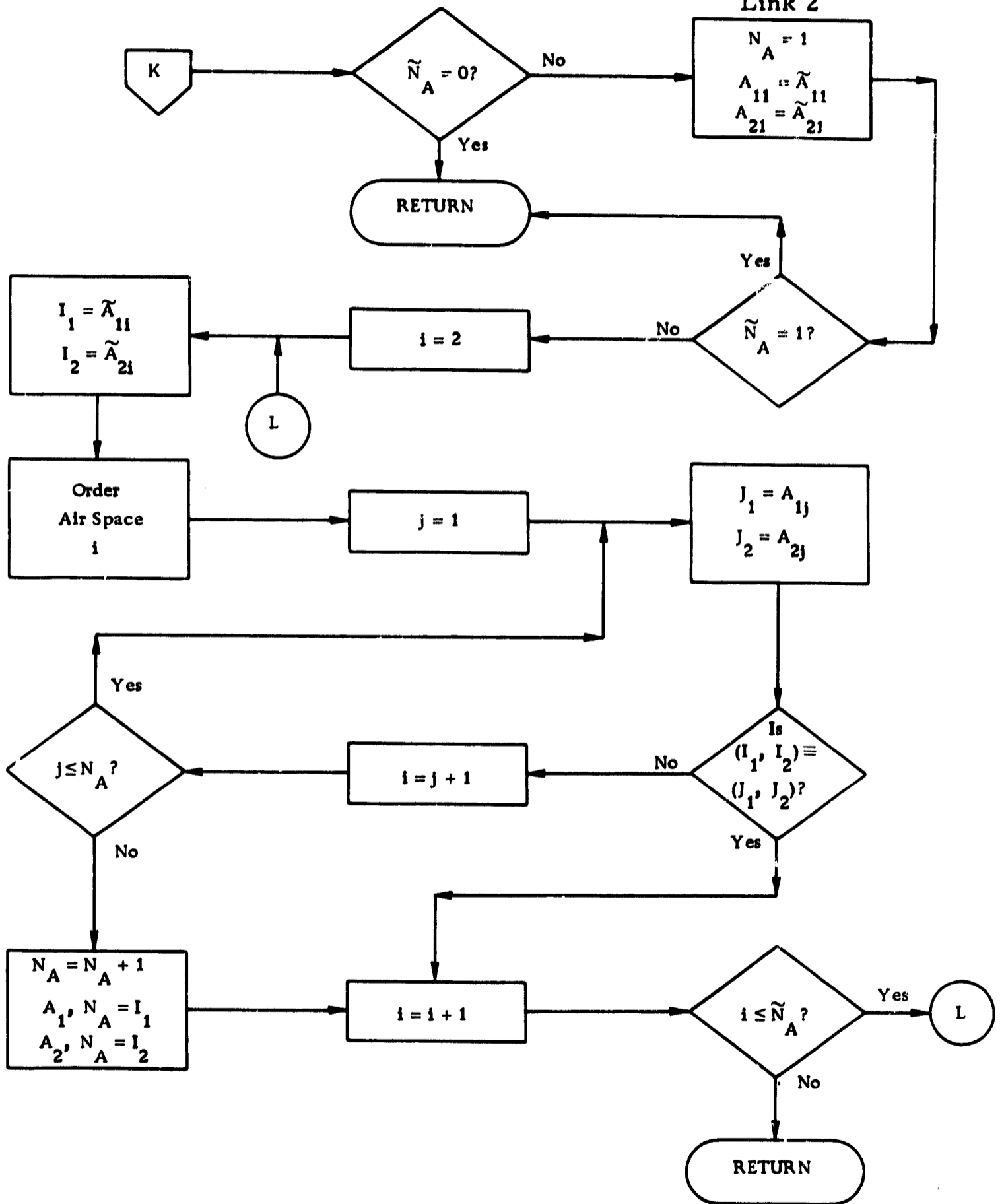
COMMON/DATA/
COMMON/PRNT/
COMMON/ELMAIR/
PRTCTL





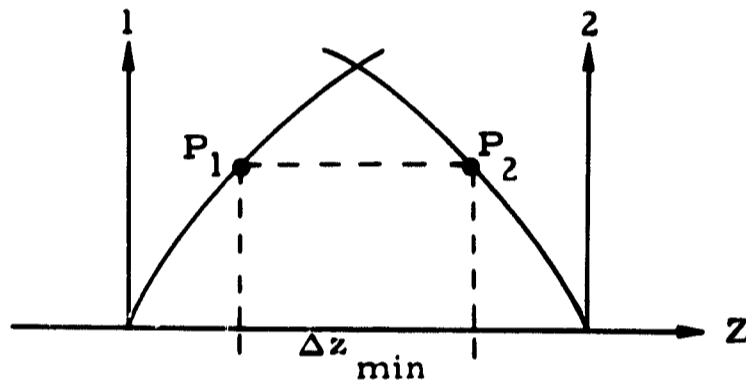






3.2.3 CLOSSZ (CLOSS)

Given curve 1 of the form $y = f_1(z)$ for $z_{1, \min} < z < z_{1, \max}$ and curve 2 of the form $y = f_2(z)$ for $z_{2, \min} < z < z_{2, \max}$, CLOSSZ determines if they intersect at some point. If there is a point of intersection, CLOSSZ determines $P_1: (z_1, y_1)$ lying on curve 1 and $P_2: (z_2, y_1)$ lying on curve 2 such that $z_2 - z_1 = \Delta z_{\min}$.



Calling Sequence

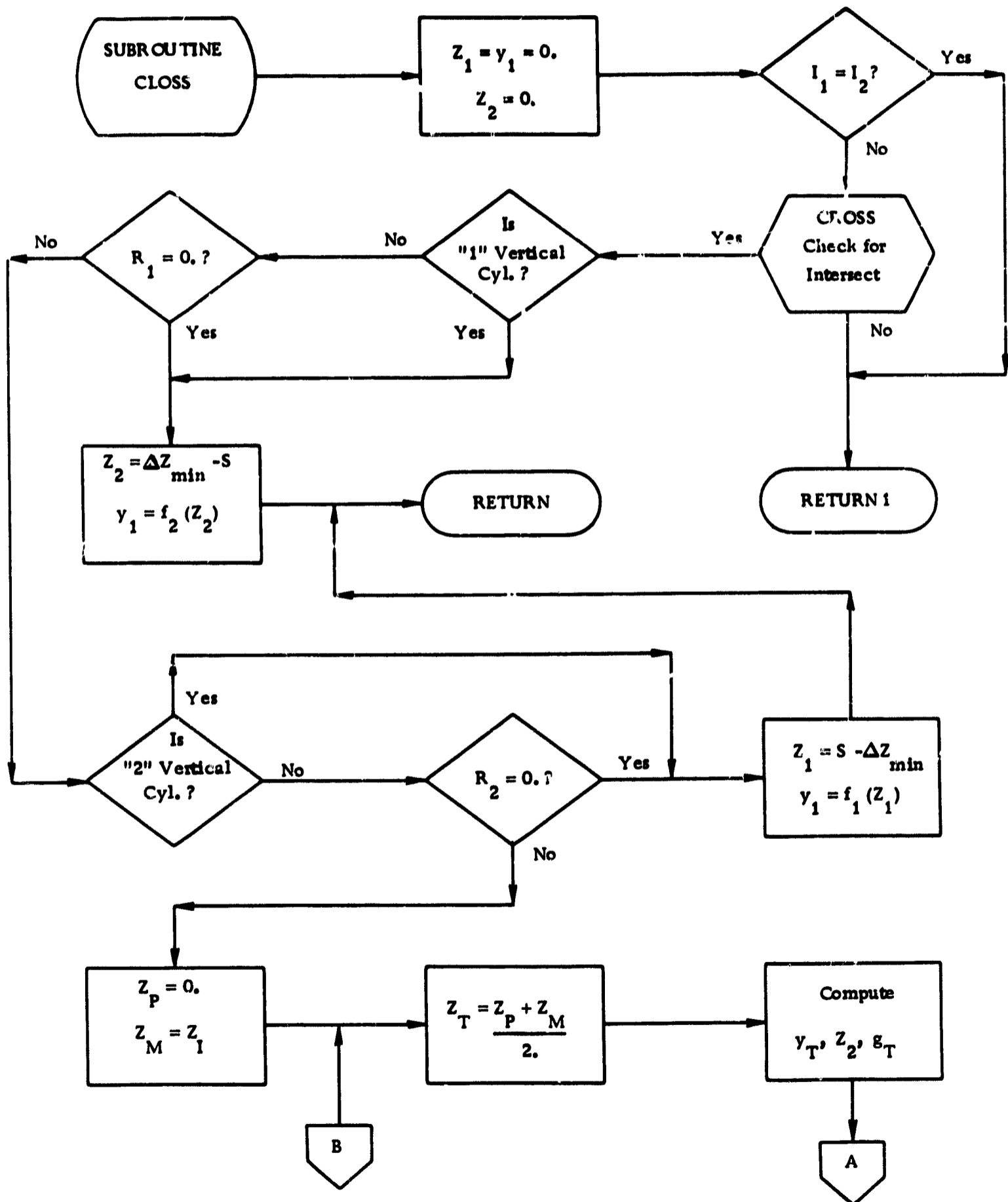
CALL CLOSS (I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX,
 DZMIN, Z1CLOS, Y1CLOS, Z2CLOS, \$ALT)

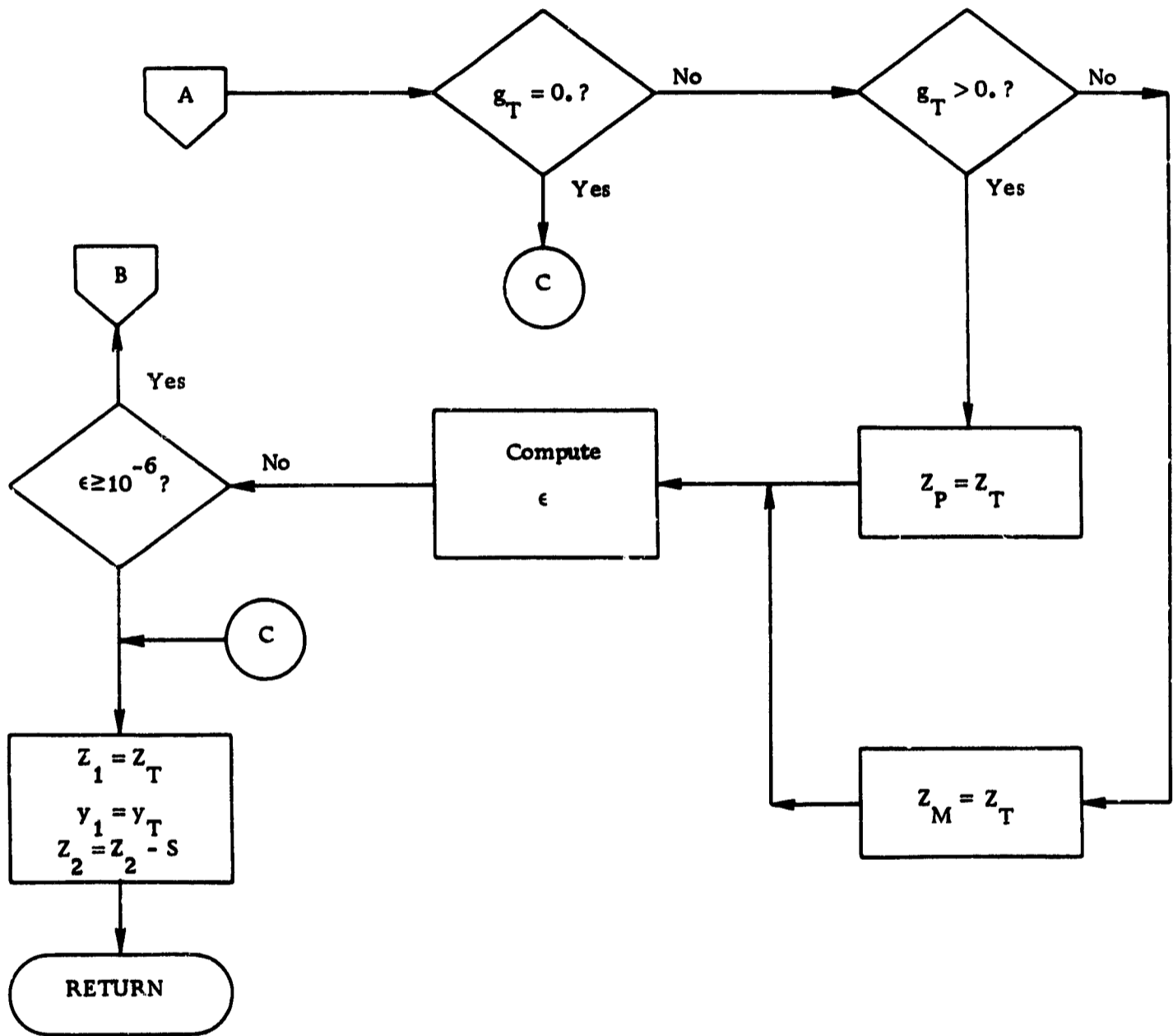
I1: Surface number of $y = f_1(z)$
 Z1MIN: $z_{1, \min}$
 Z1MAX: $z_{1, \max}$
 I2: Surface number of $y = f_2(z)$
 Z2MIN: $z_{2, \min}$
 Z2MAX: $z_{2, \max}$
 DZMIN: Δz_{\min}
 Z1CLOS: z_1
 Y1CLOS: y_1
 } P_1
 \$ALT: Alternate return if no intersection

TR-67-700-10-2
Page III-3-57
Program Structure
Link 2

Utility Routines and Common References

COMMON/DATA/
COMMON/ELMAIR/
CROSS





3.2.4 CPLLOTZ (CPLLOT)

CPLLOTZ employs the JPL FORTRAN IV compatible SPLOT* package to generate a binary file (logical 18) containing data to be plotted on the SC4020, in particular, it produces a scaled cross sectional picture of the lens system.

Calling Sequence

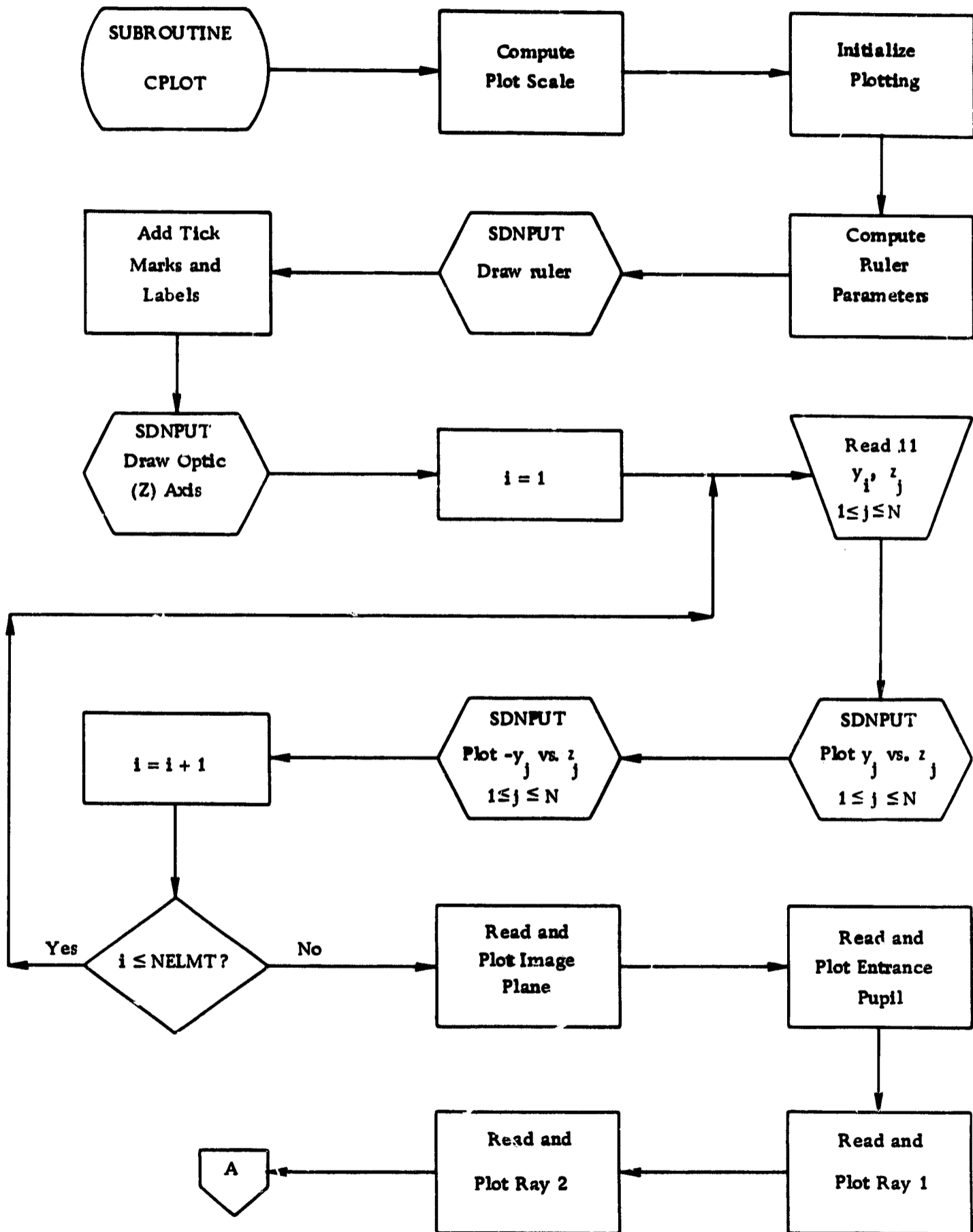
CALL CPLLOT (ZMIN, ZMAX)

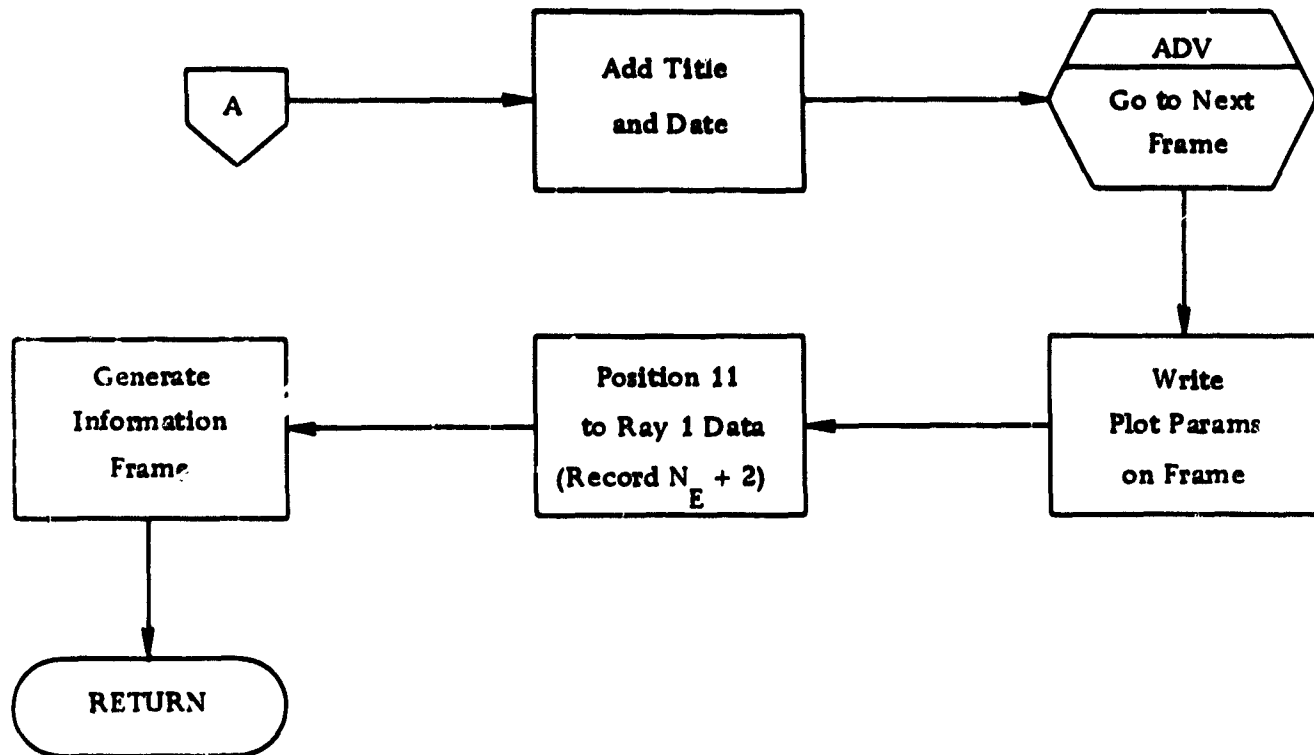
where ZMIN is the minimum value of z relative to the entrance pupil and ZMAX is the maximum value relative to the entrance pupil.

Utility Routines and Common References

COMMON/DATA/	
COMMON/ELMAIR/	
COMMON/CODRNG/	
COMMON/PLOT/	STERM*
CAMERA*	SPRNTA*
SLABEL*	SETPL*
SDINIT*	RESPL*
SDNPUT*	ADV*

* See Appendix A





3.2.5 CROSSZ (CROSS)

Given $y = f_1(z)$ for $z \in [z_{1, \min}, z_{1, \max}]$ relative to the vertex or surface 1 and $y = f_2(z)$ for $z \in [z_{2, \min}, z_{2, \max}]$ relative to the vertex of surface 2, CROSSZ determines the point of intersection (z_I, y_I) where z_I is relative to the vertex of surface 1.

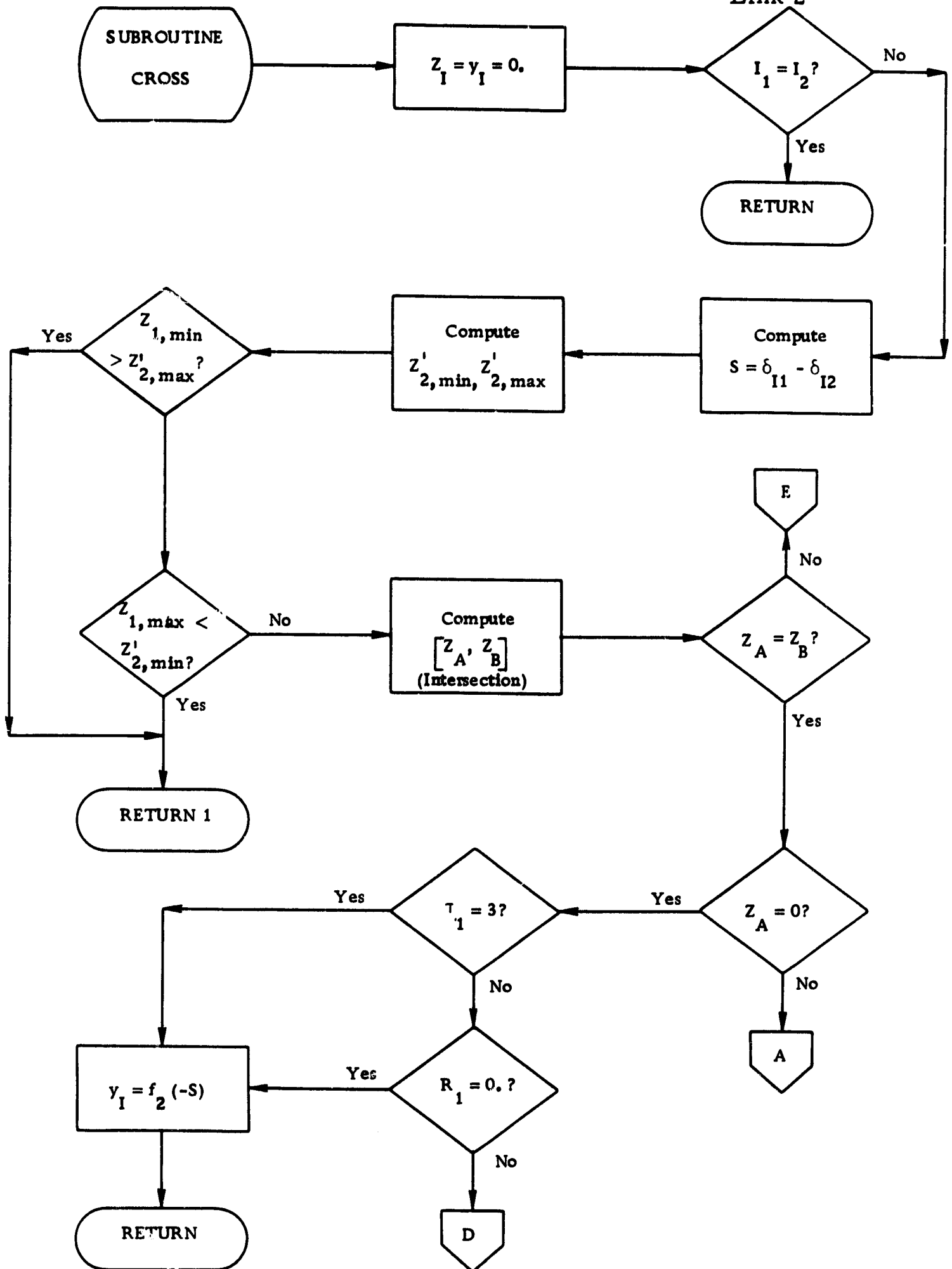
Calling Sequence

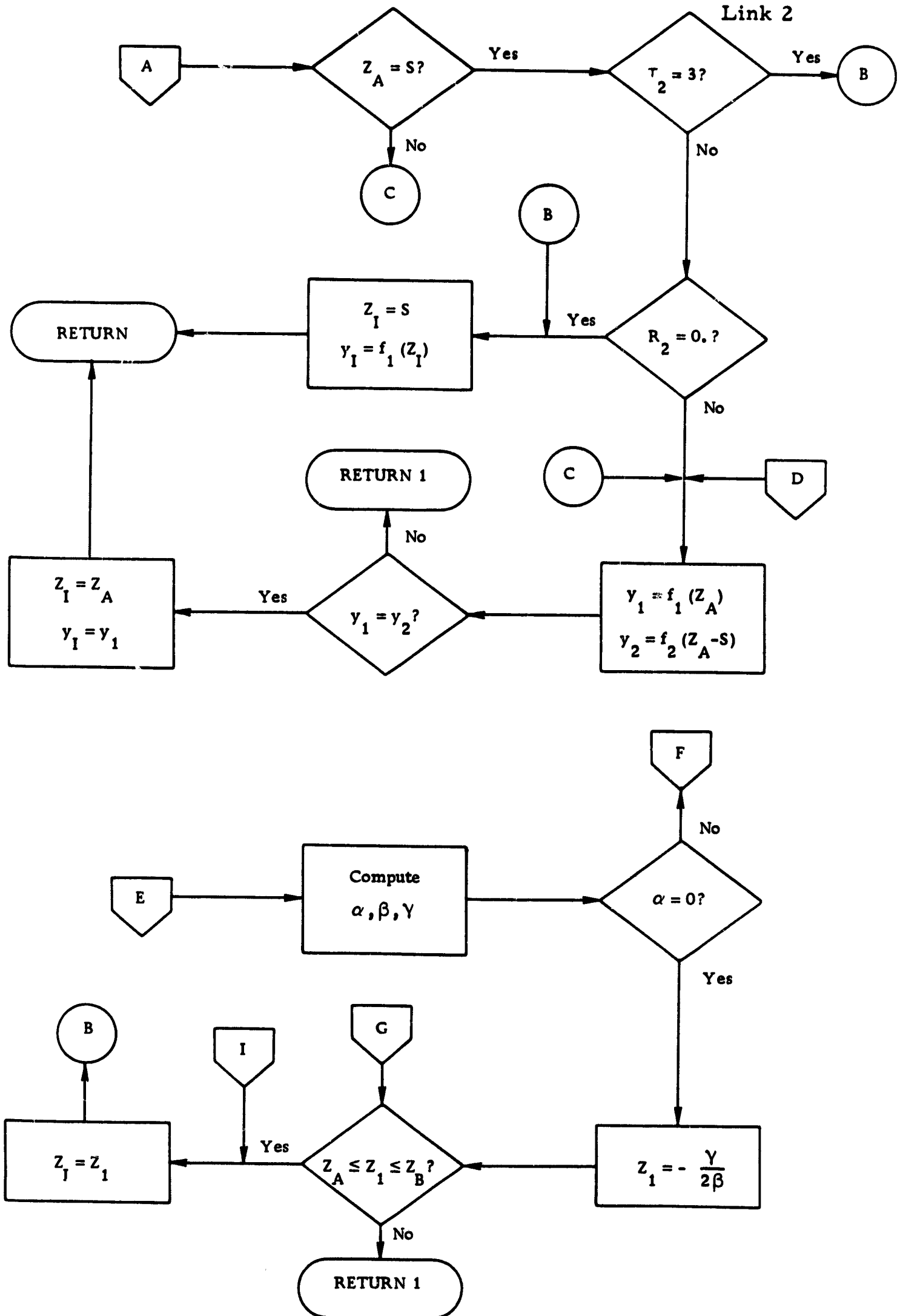
CALL CROSS(I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX,
ZCROSS, YCROSS, \$ALT)

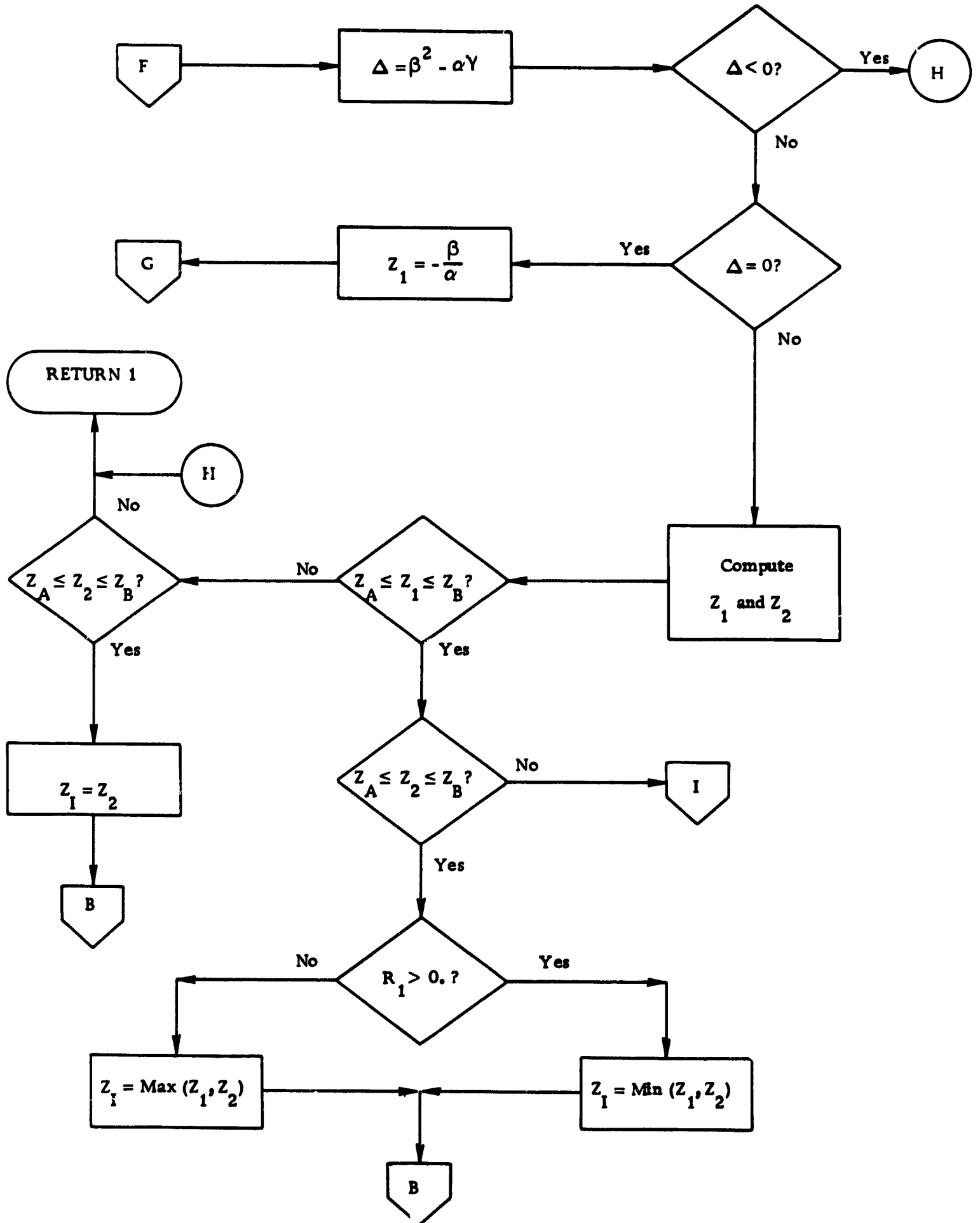
I1: Surface number of $y = f_1(z)$
Z1MIN: $z_{1, \min}$
Z1MAX: $z_{1, \max}$
I2: Surface number of $y = f_2(z)$
Z2MIN: $z_{2, \min}$
Z2MAX: $z_{2, \max}$
ZCROSS: z_I
YCROSS: y_I
\$ALT: Alternate return if no intersection

Utility Routines and Common References

COMMON/DATA/
COMMON/ELMAIR/







3.2.6 ENDPTZ (ENDPTS)

ENDPTZ generates vector NARCS and the 3 dimensional array ARCS. The Ith layer of ARCS contains the end points of the arcs which constitute the upper half of element I as follows:

$$\text{ARCS} (\quad , \quad , I) = \begin{bmatrix} z_1 & z_2 & \cdot & \cdot & \cdot & z_N \\ y_1 & y_2 & \cdot & \cdot & \cdot & y_N \end{bmatrix}$$

where NARCS(I) = N. Element I is composed of N - 1 arcs where $2 \leq N \leq 5$; the end points of the Kth arc being (z_K, y_K) and (z_{K+1}, y_{K+1}) . Each arc is either a horizontal line ($y_K = y_{K+1}$), a vertical line ($z_K = z_{K+1}$), or a portion of a plane conic. If I_1 and I_2 are the surface numbers of the curves which form element I then all z values are referenced to the vertex of I_1 .

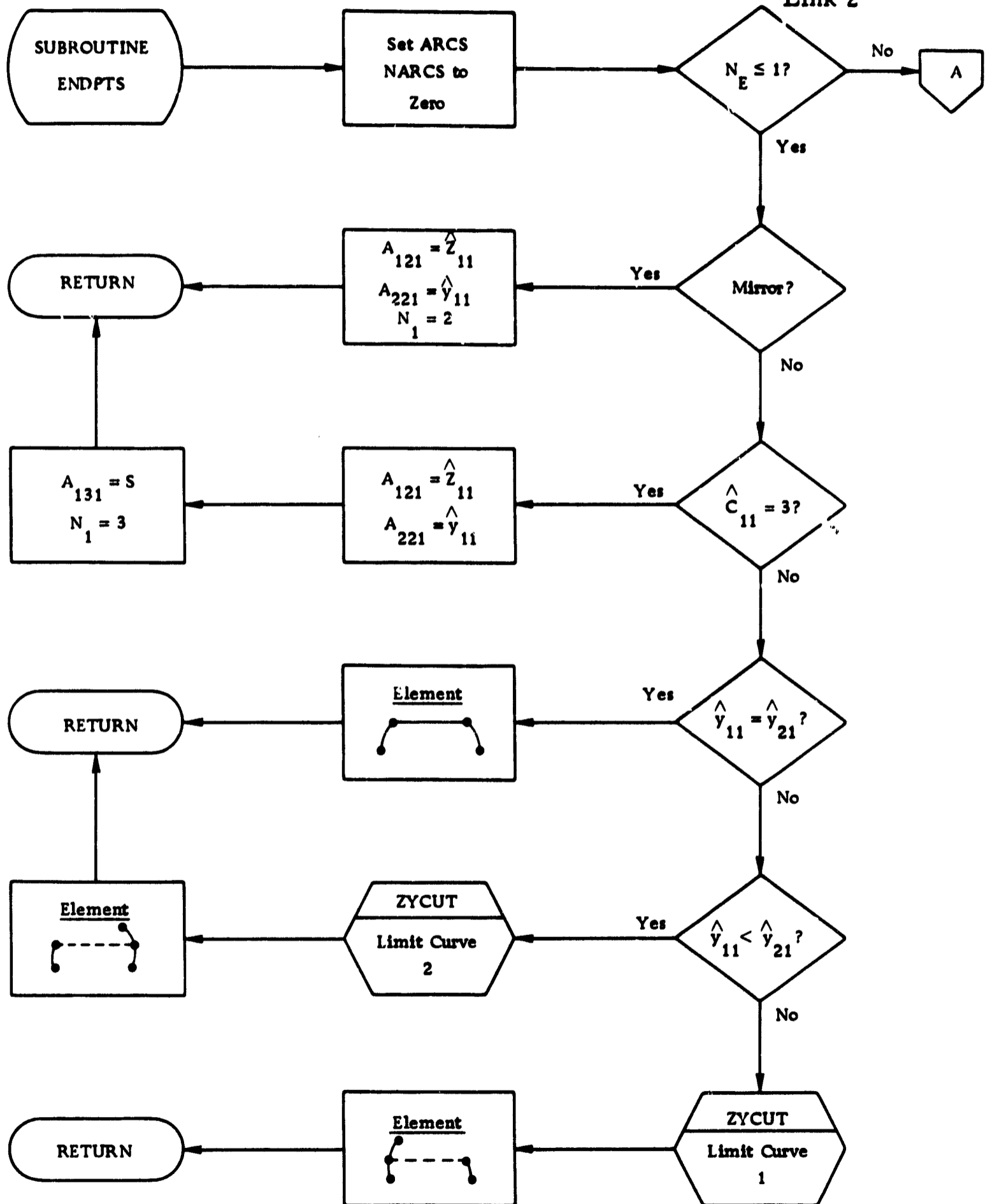
Calling Sequence

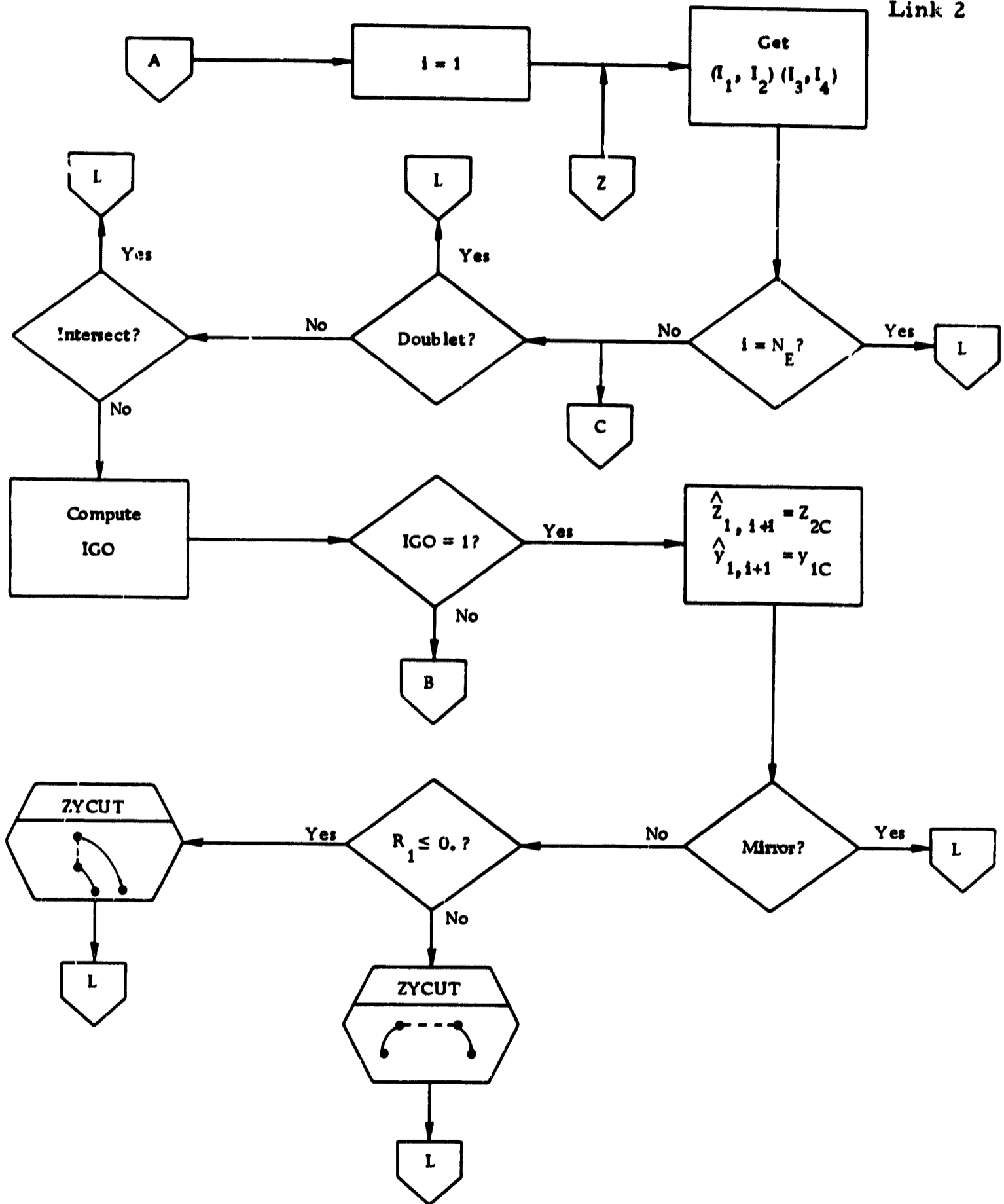
CALL ENDPTS (DZMIN, ARCS, NARCS)

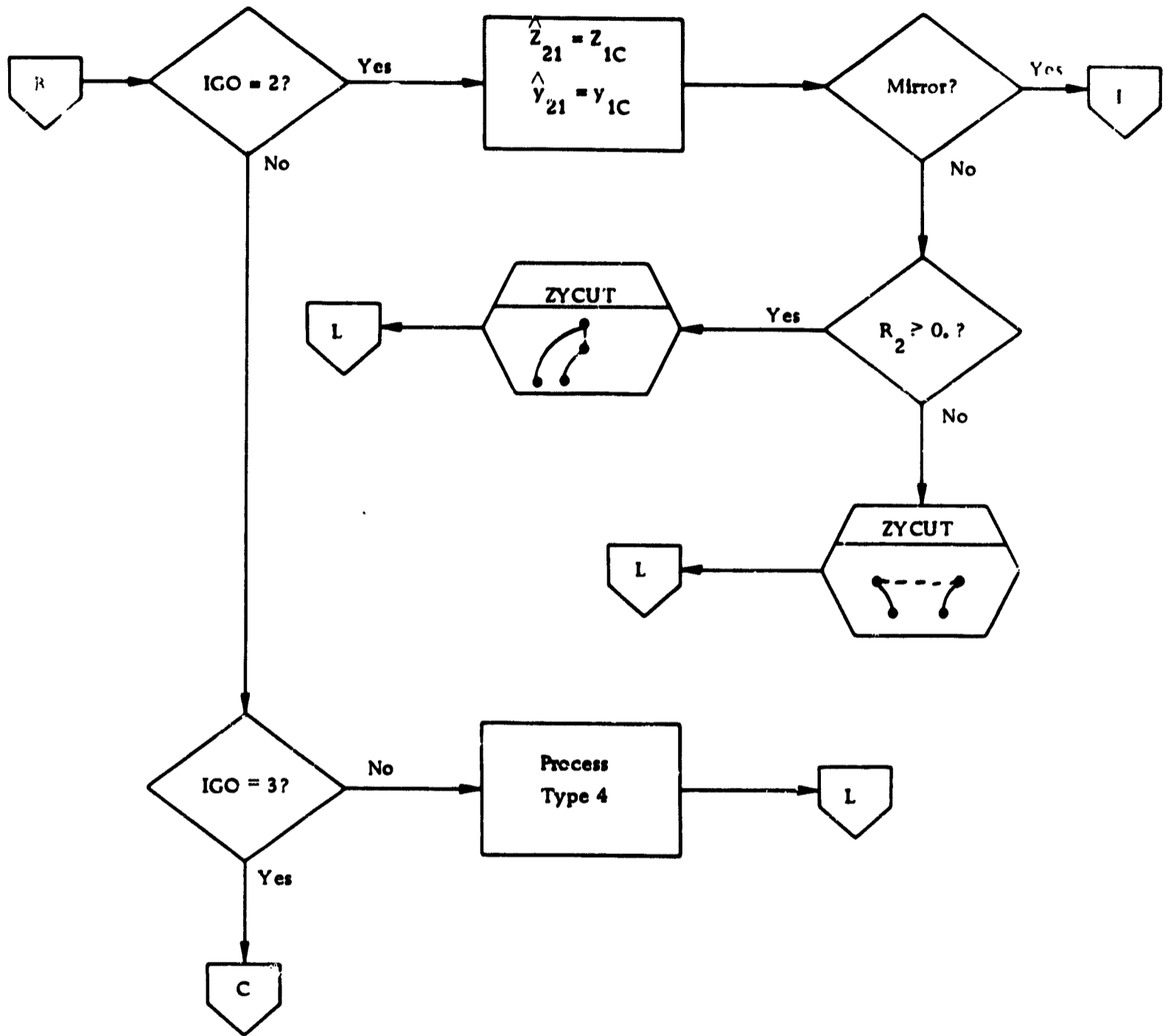
where DZMIN is the minimum permissible air space thickness, ARCS is the 2 x 5 x 100 array described above, and NARCS is a vector containing the arc count for each element.

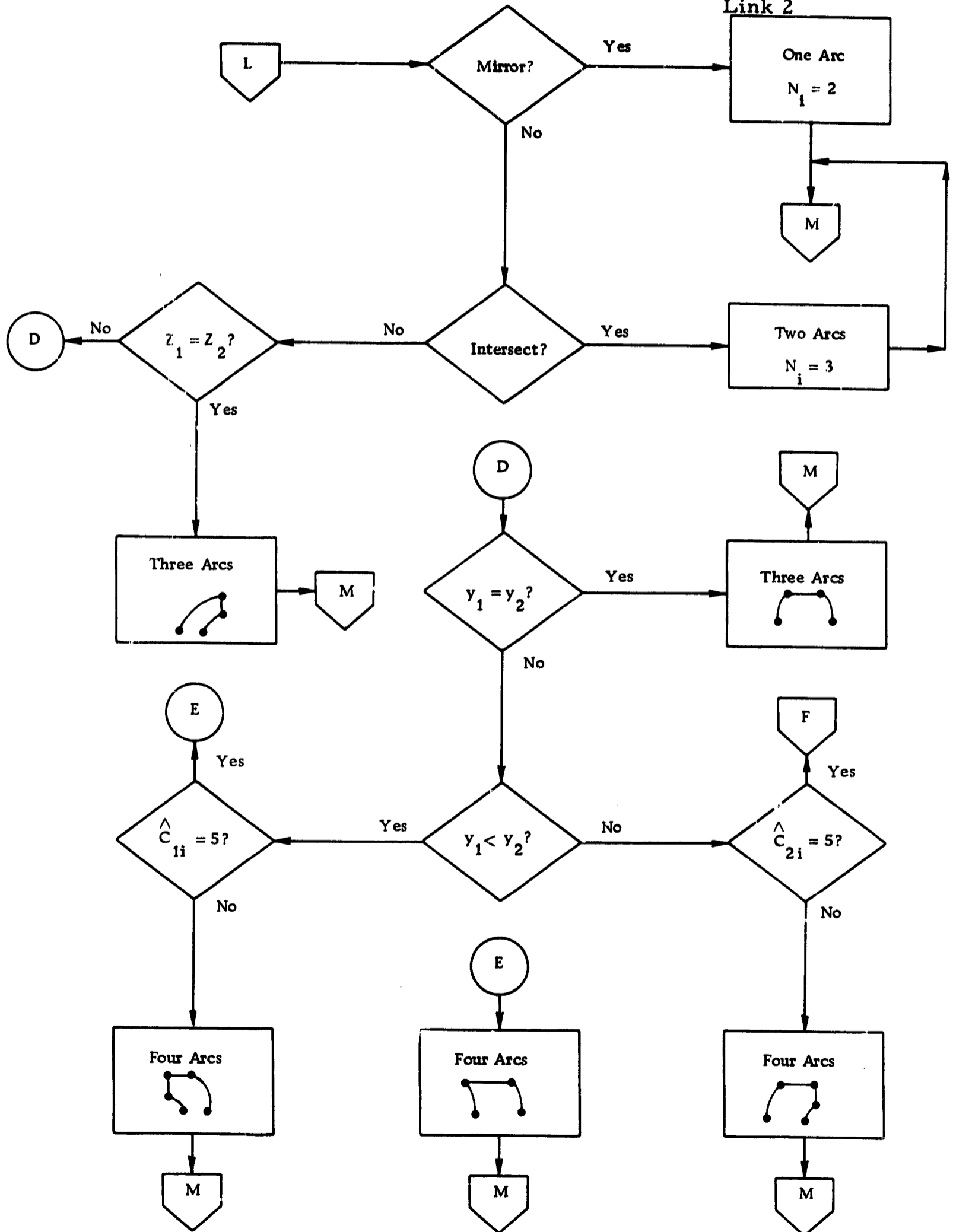
Utility Routines and Common References

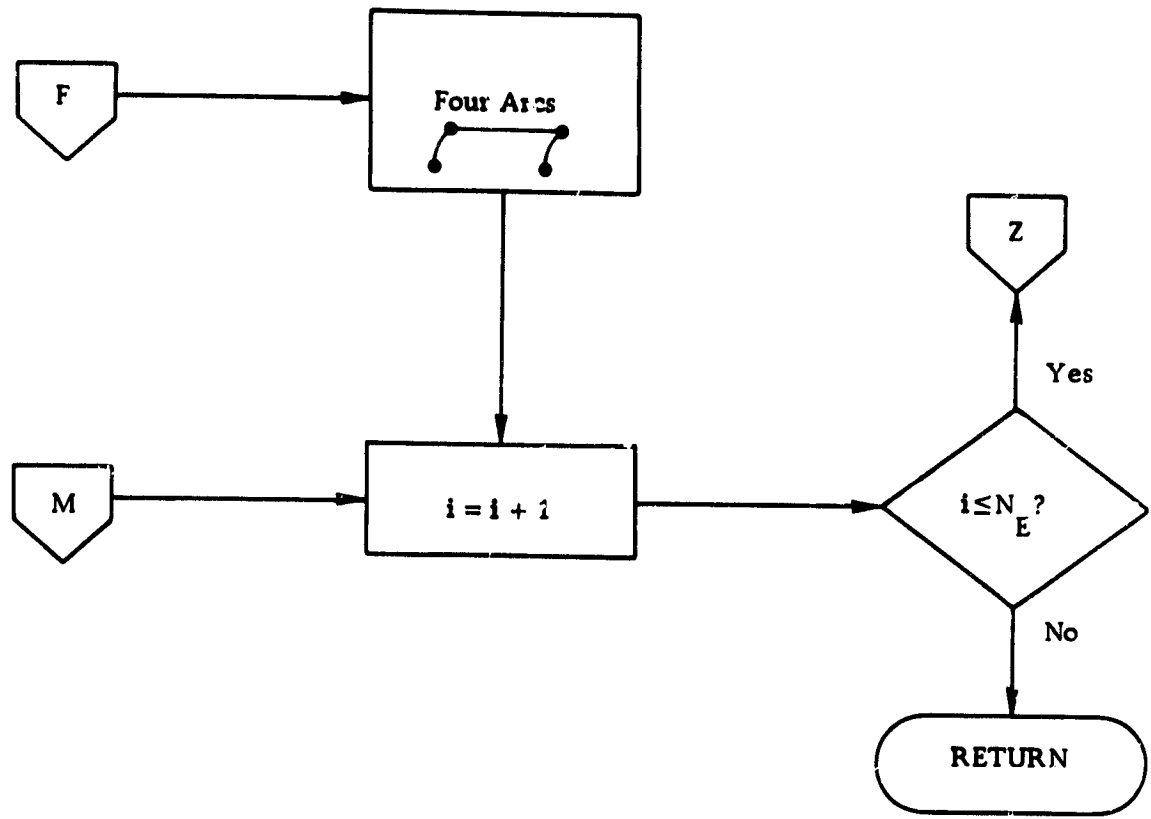
COMMON/DATA
COMMON/ELMAIR/
COMMON/CODRNG/
ZYCUT
CLOSS
GOOF











3.2.7 EIRGCZ (EIRGCD)

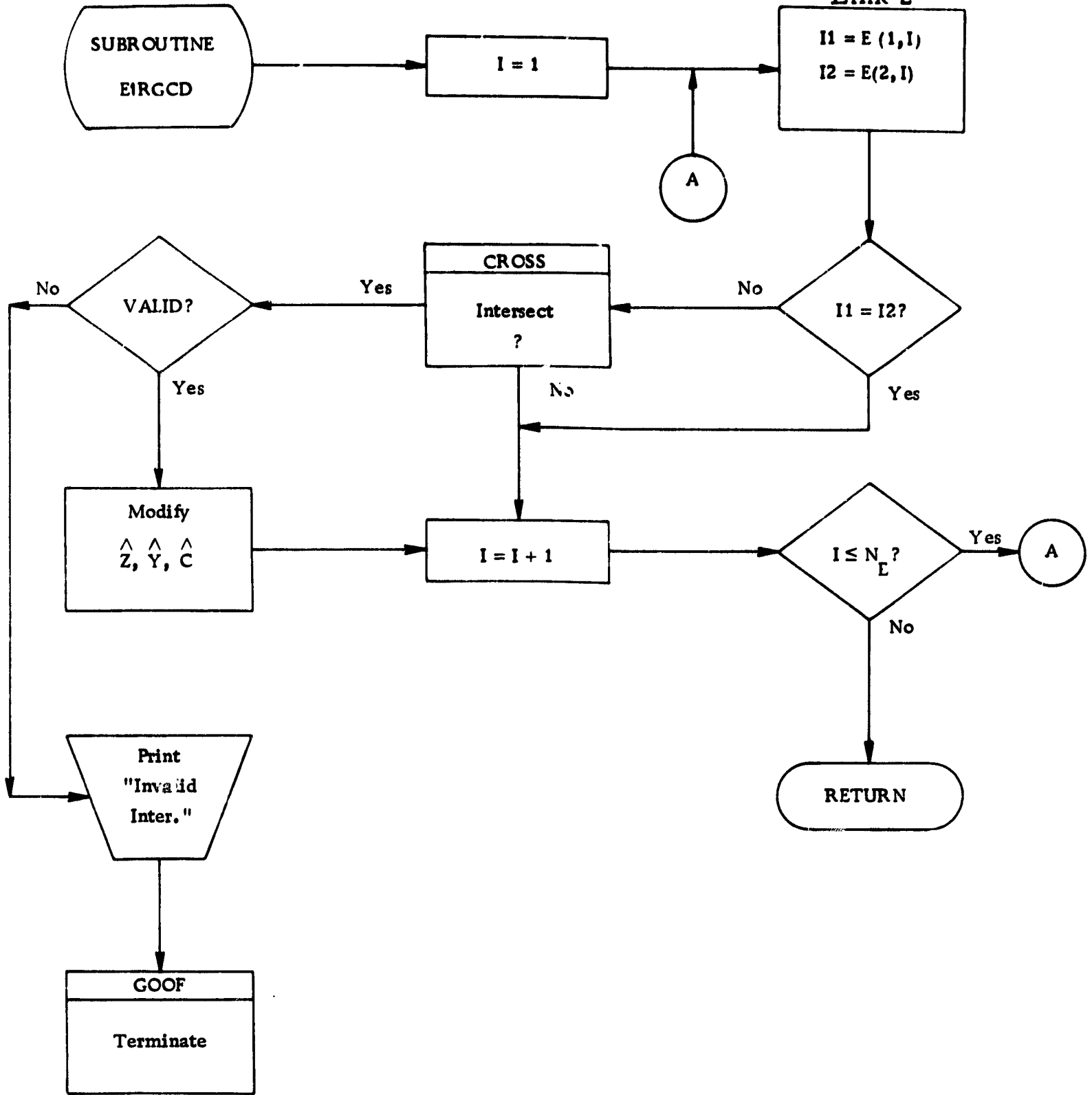
EIRGCZ employs subroutine CROSS to edit the range and code matrices by subjecting each element to intersection constraints.

Calling Sequence

CALL EIRGCD

Utility Routines and Common References

COMMON/ELMAIR/
COMMON/CODRNG/
CROSS
GOOF



3.2.8 PFILEZ (PFILE)

PFILEZ creates a binary file (logical 11) containing $N_E + 4$ logical records each one of which is 601 words long where N_E is the number of elements ($N_E \geq 1$). Each record consists of N , z_i ($1 \leq i \leq 300$), and y_i ($1 \leq i \leq 300$) where $1 \leq N \leq 300$ is the number of points distributed along the upper half ($y \geq 0.$) of the i th element for $z_{\min} \leq z \leq z_{\max}$.

The first N_E records describe the N_E elements and:

Record $N_E + 1$:	Image Plane
Record $N_E + 2$:	Entrance Pupil
Record $N_E + 3$:	Ray 1
Record $N_E + 4$:	Ray 2

Calling Sequence

CALL PFILE (ARCS, NARCS, ZMIN, ZMAX)

ARCS: Three dimensional array determined by ENDPTS

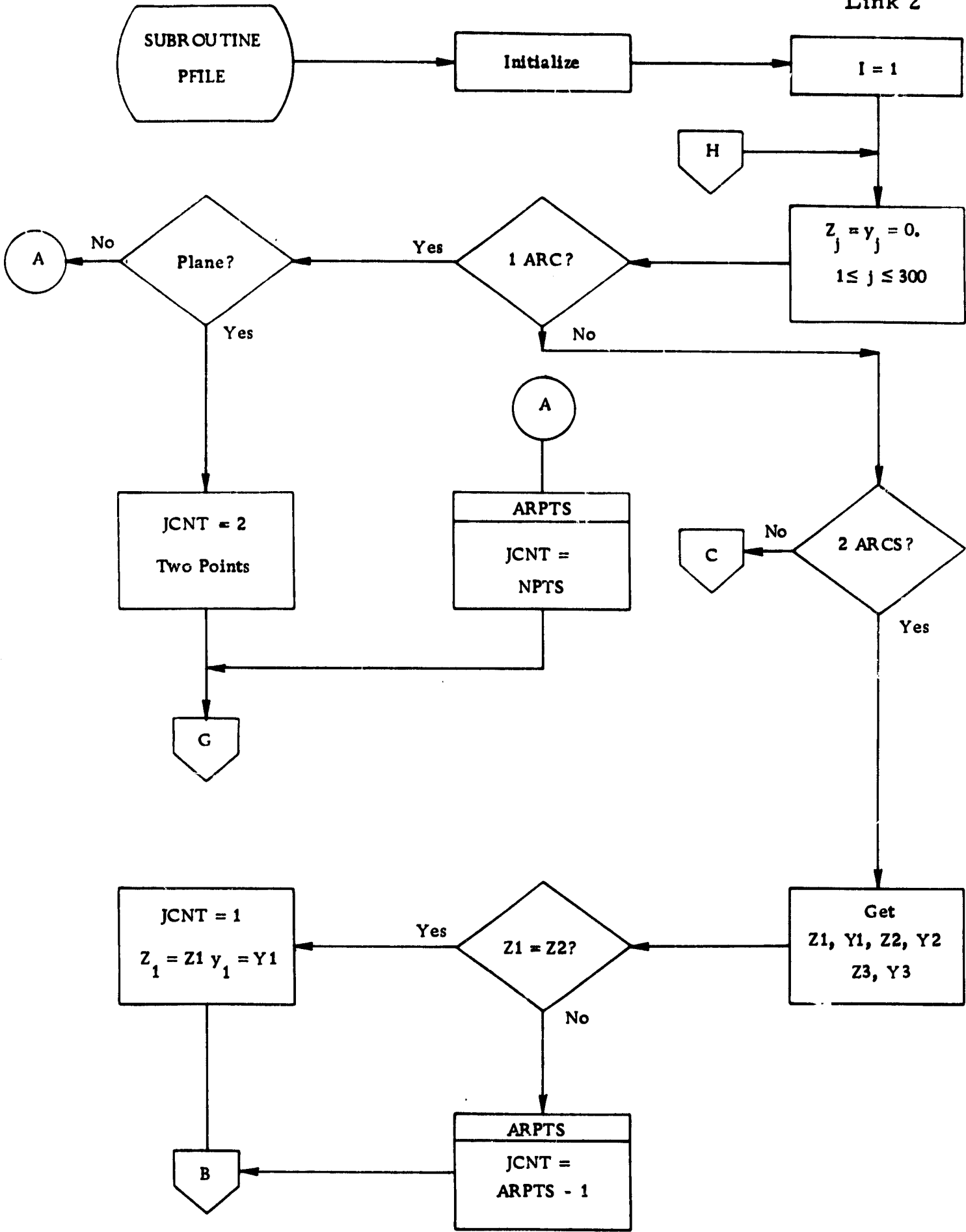
NARCS: Vector determined by ENDPTS

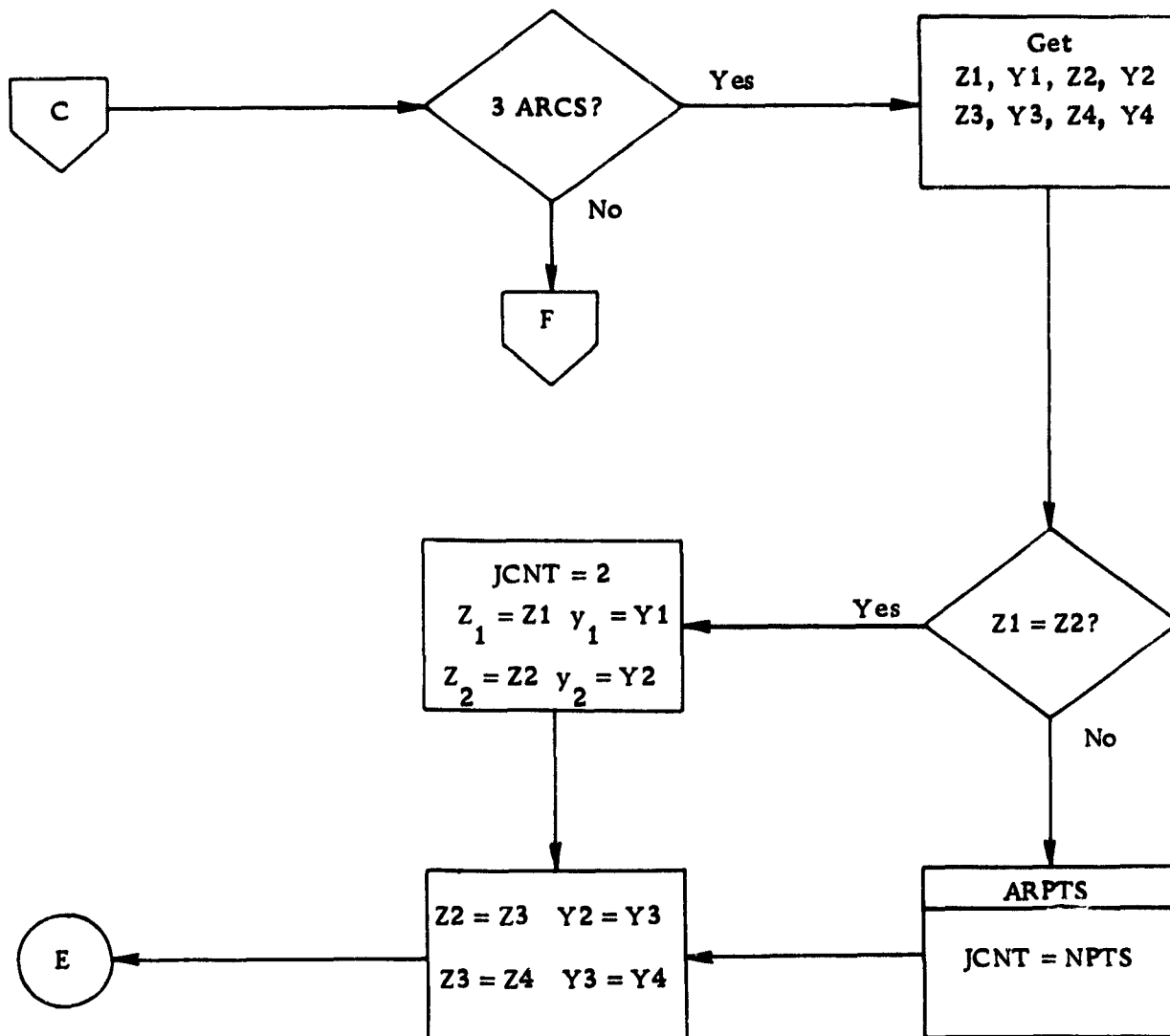
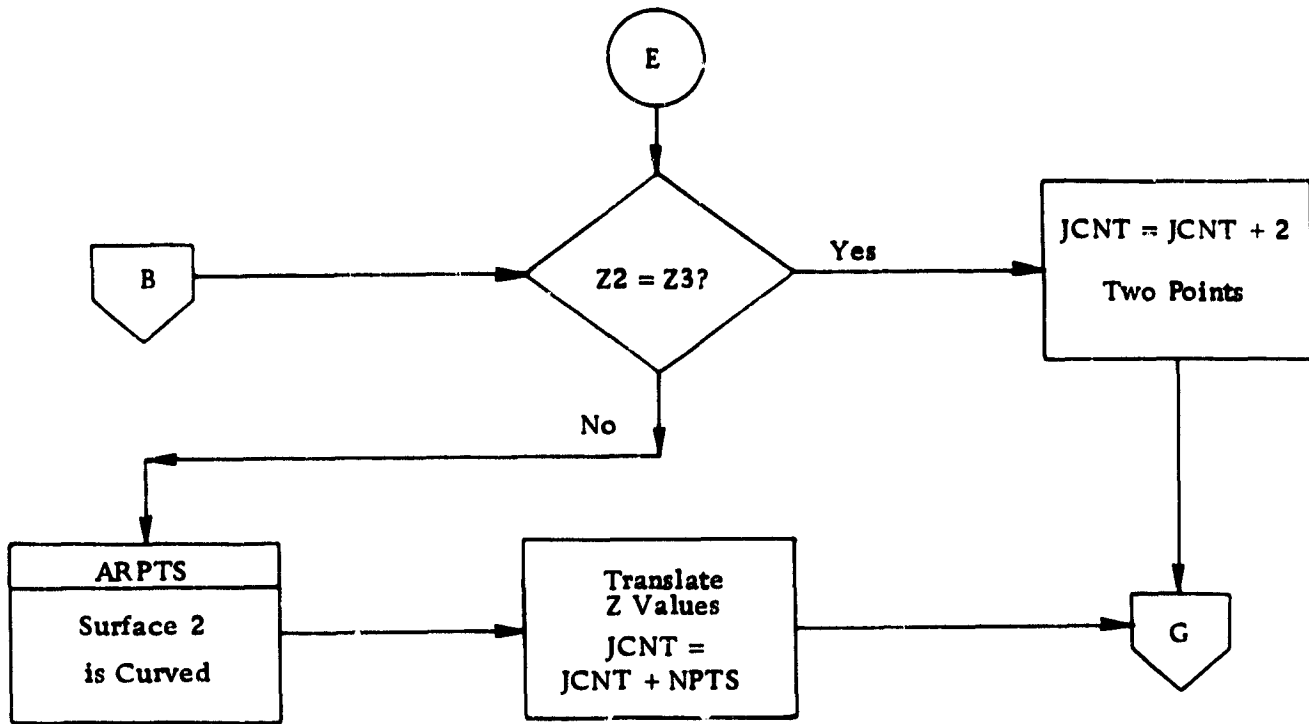
ZMIN: Minimum z relative to entrance pupil.

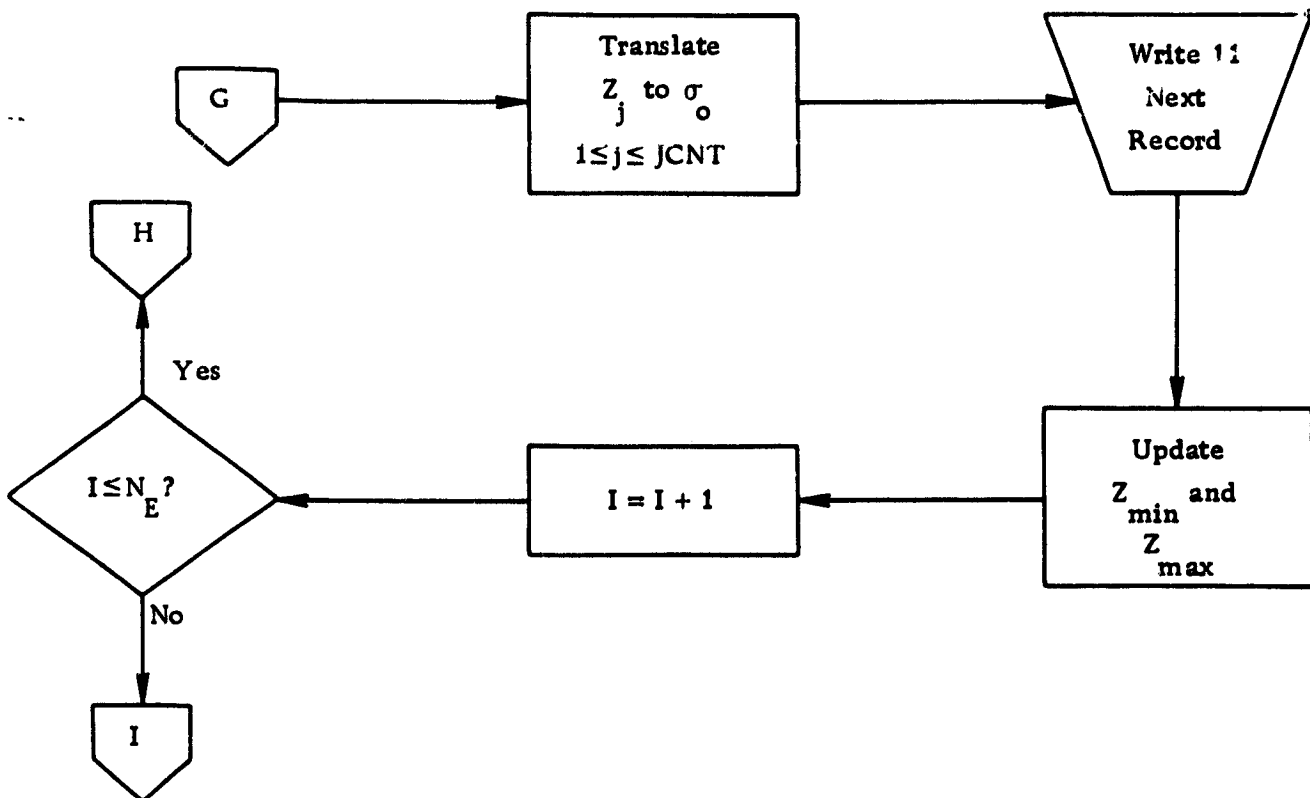
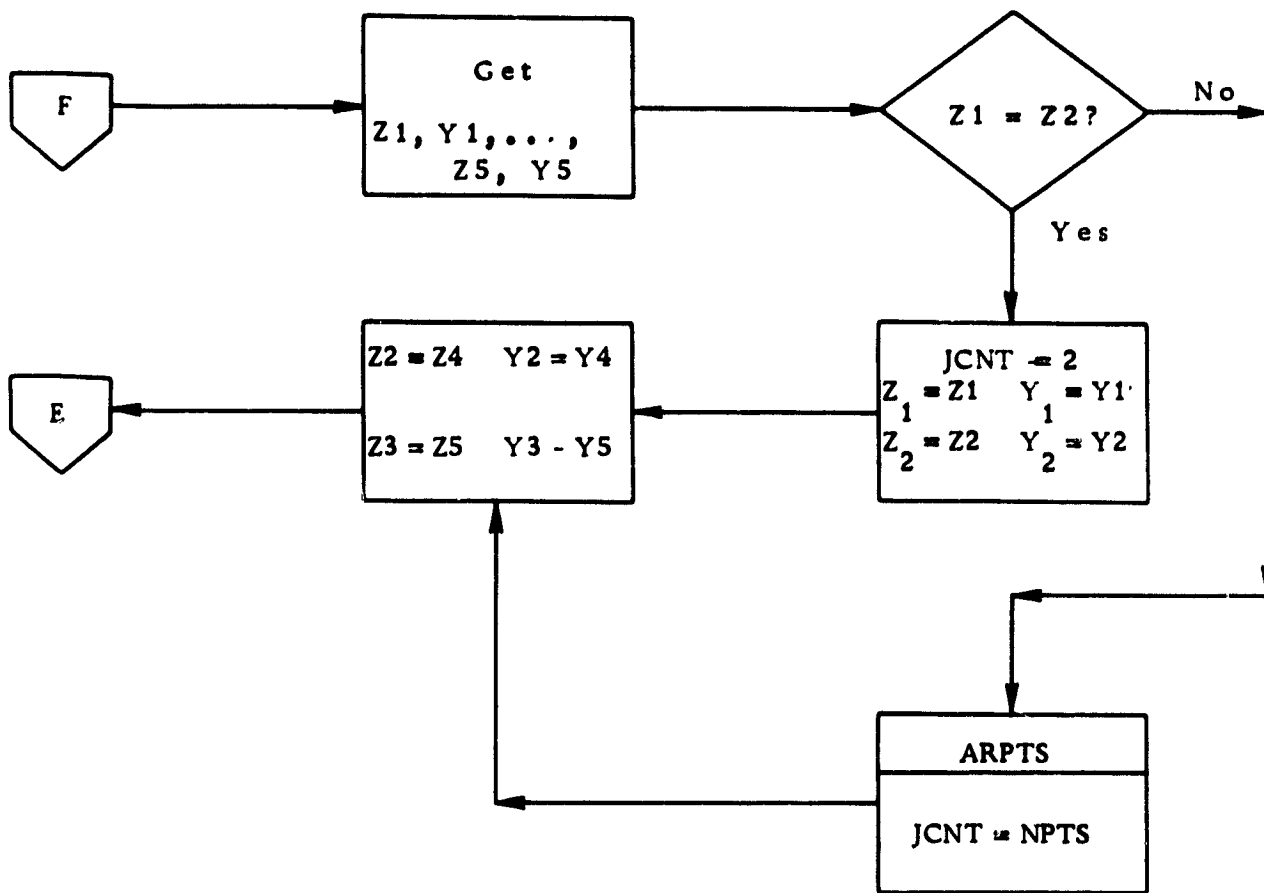
ZMAX: Maximum z relative to entrance pupil

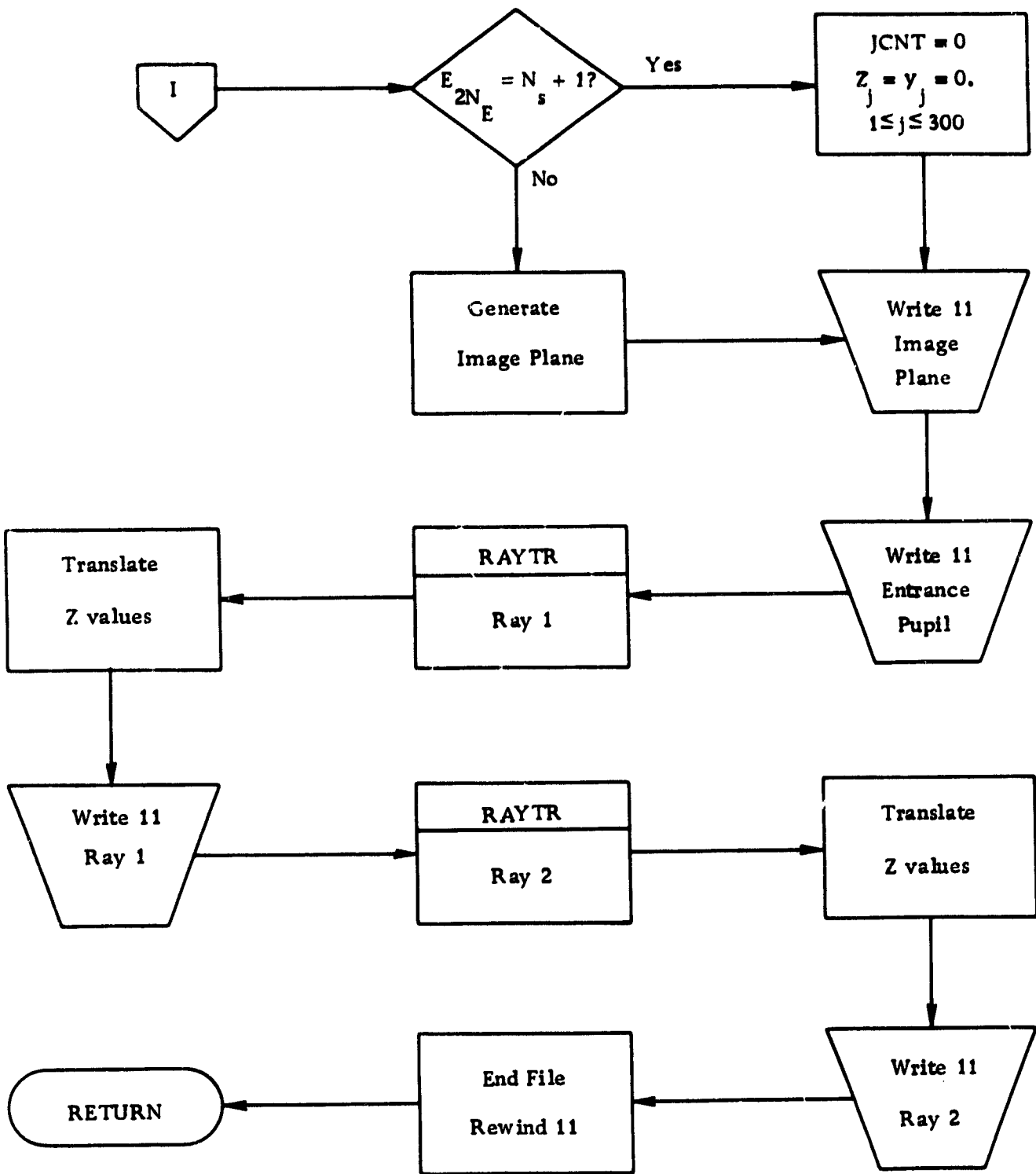
Utility Routines and Common References

COMMON/DATA/
COMMON/ELMAIR/
COMMON/CODRNG/
COMMON/PLOTG/
COMMON/AZOBJ/
ARPTS
RAYTR









3.2.9 PROFLZ (PROFIL)

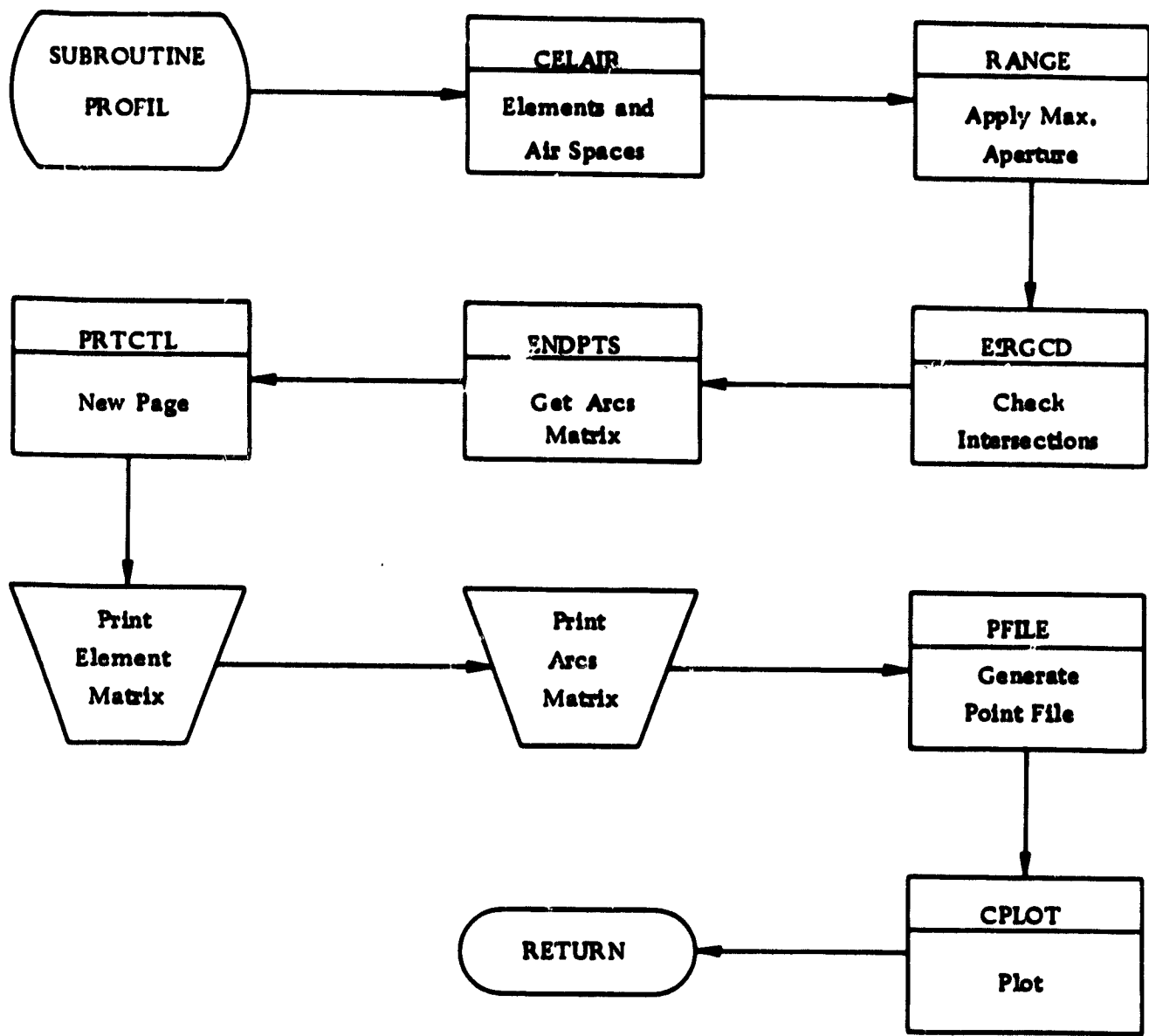
PROFLZ is the main or control routine of the profile plot link. It is called by the main program (FOLDP) whenever a profile plot is to be generated.

Calling Sequence

CALL PROFIL

Utility Routines and Common References

COMMON/ELMAIR/
COMMON/CODRNG/
COMMON/PLOTTC/
COMMON/PRNT/
CELAIR
RANGE
EIRGCD
ENDPTS
PRTCTL
PFILE
CPLOT



3.2.10 RANGEZ (RANGE)

RANGEZ determines the unconstrained range matrices \hat{Z} and \hat{Y} and the code matrix \hat{C} . Range is computed for each curve as if no other curves were present. The only constraints are the presence of a peak (circle or ellipse) and the maximum aperture $y = y_{\max}$. The elements of \hat{Z} (z coordinates) are always referenced to the vertex of the curve in question.

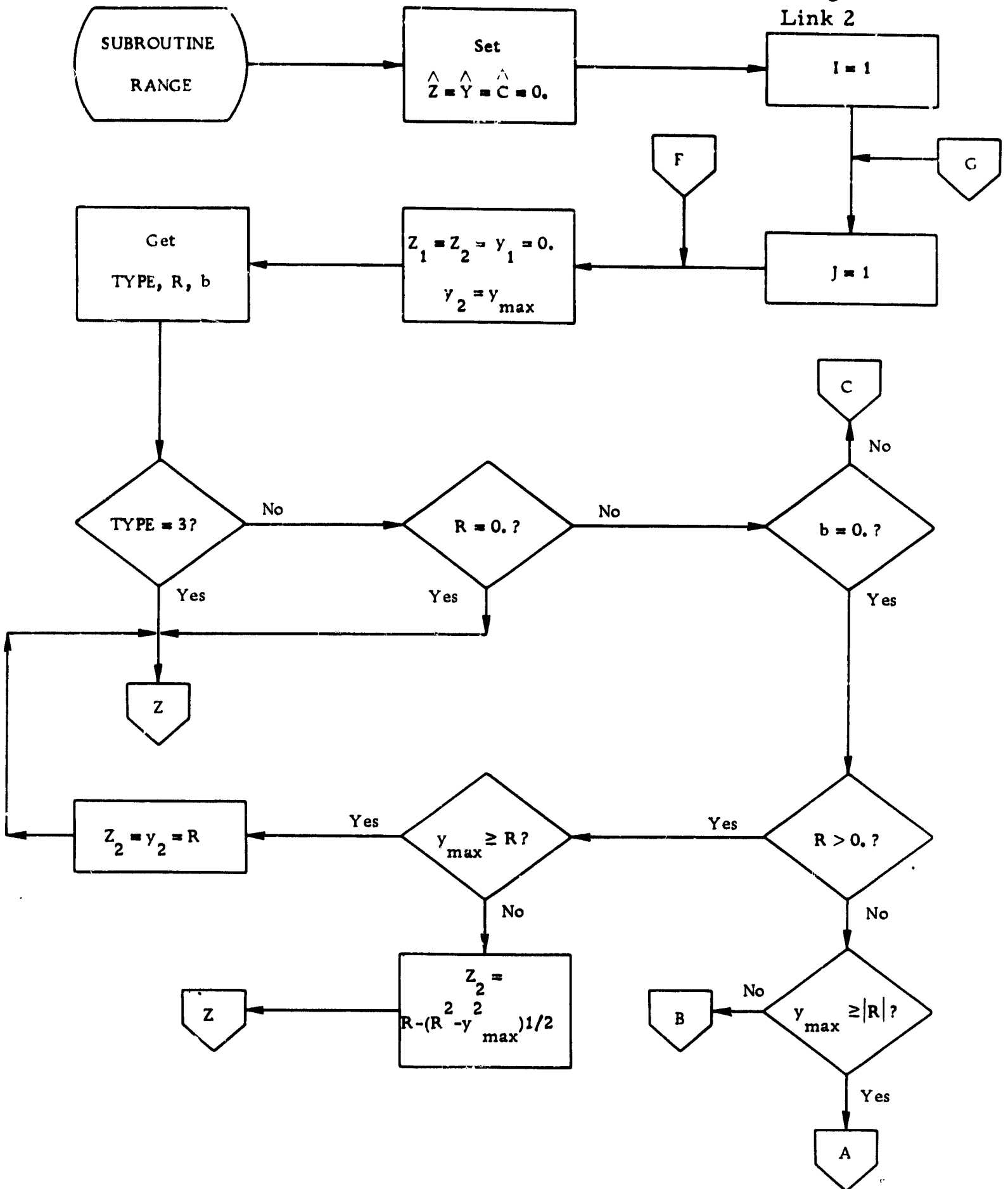
Calling Sequence

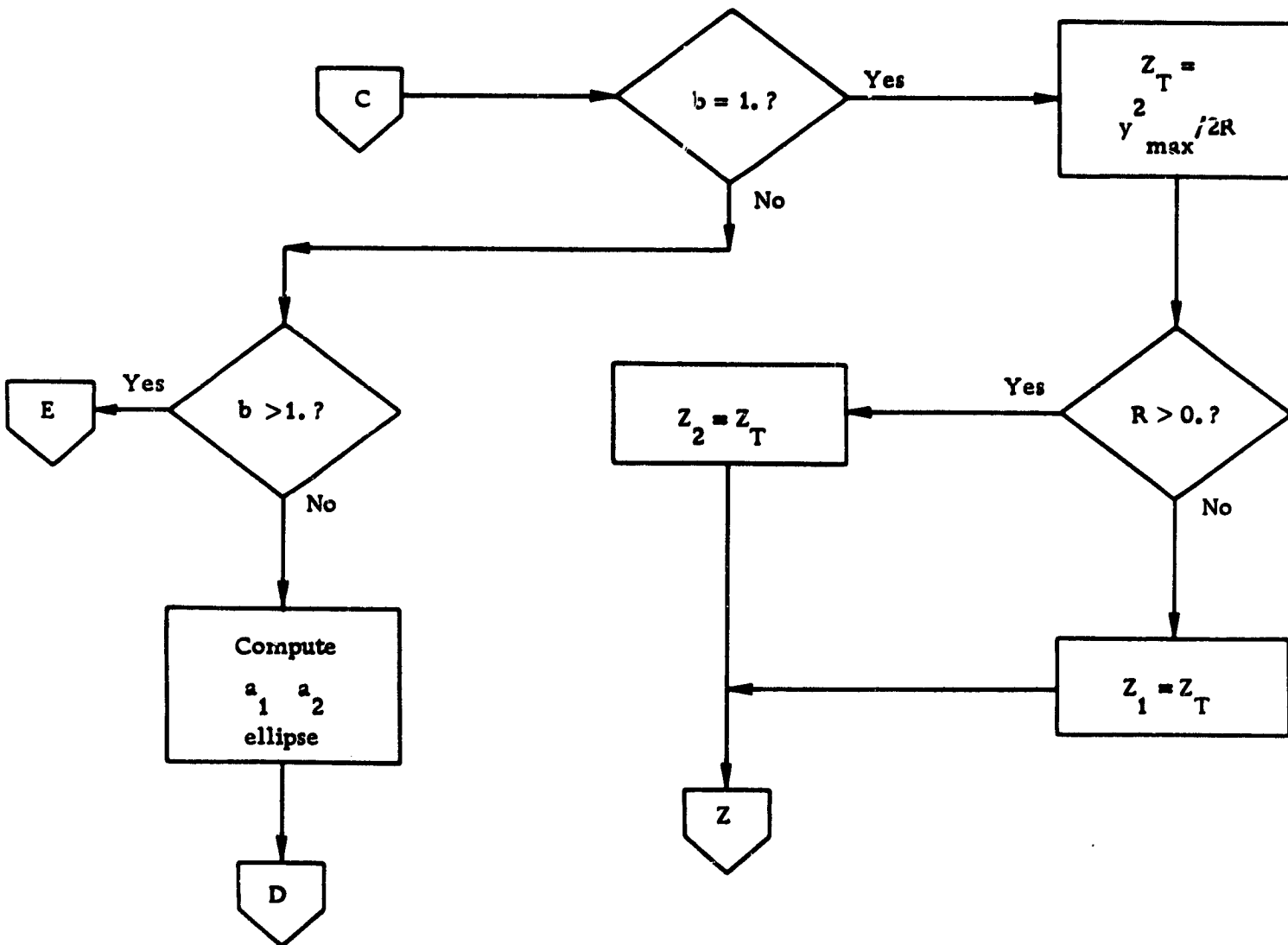
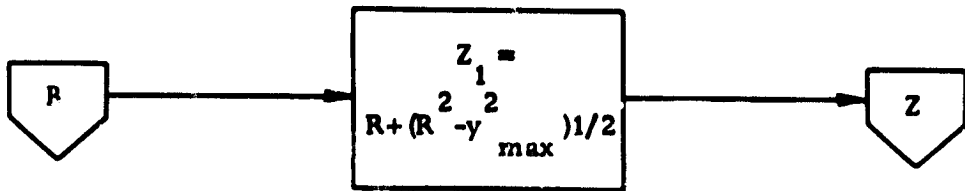
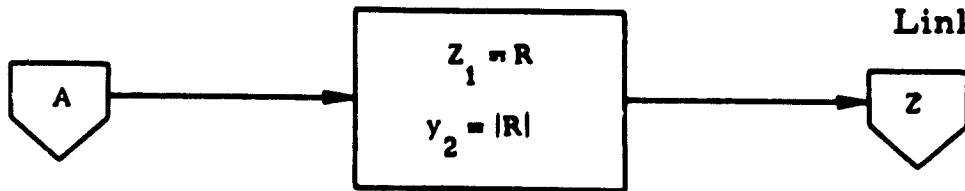
CALL RANGE (YMAXIM)

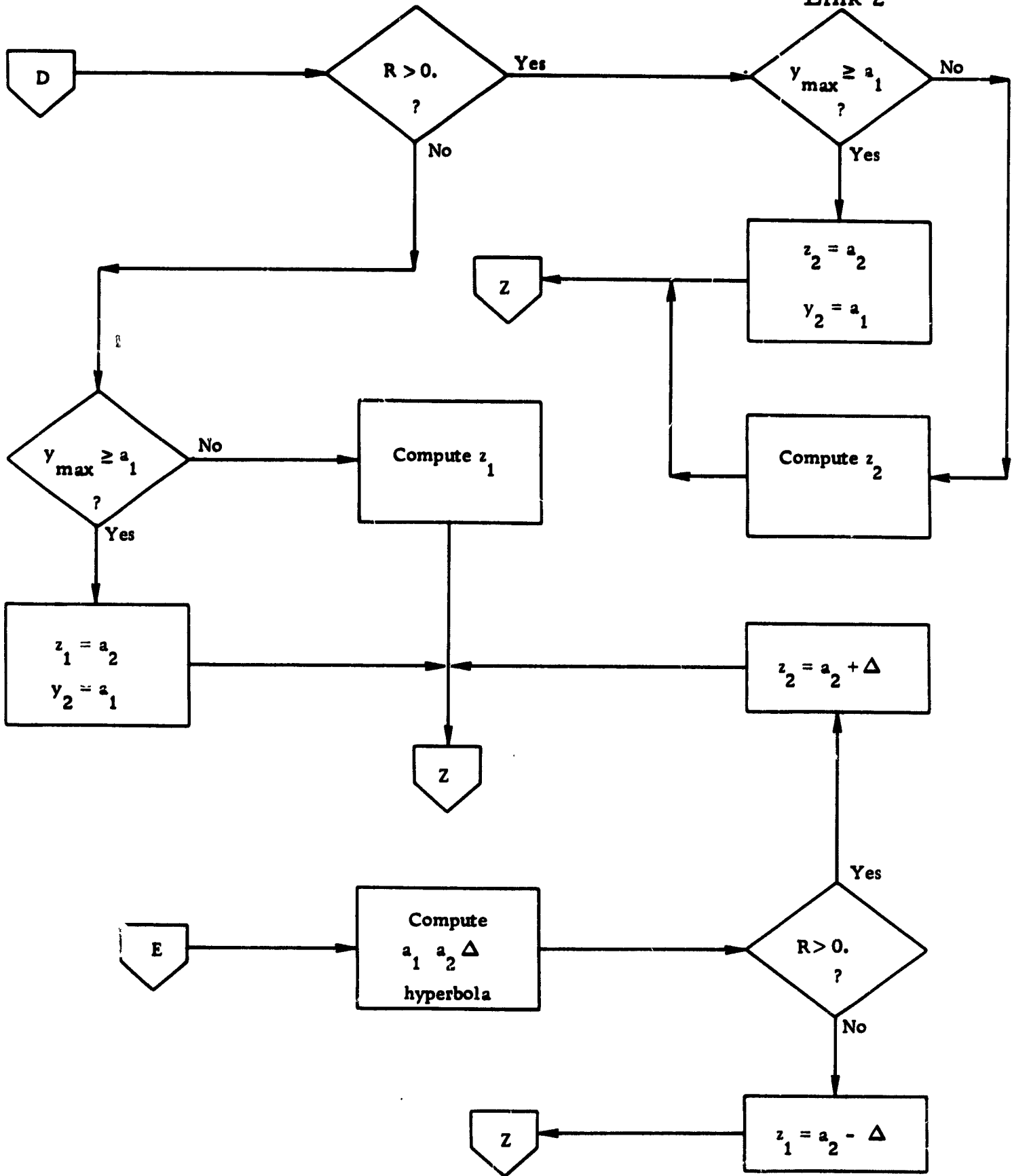
YMAXIM: y_{\max}

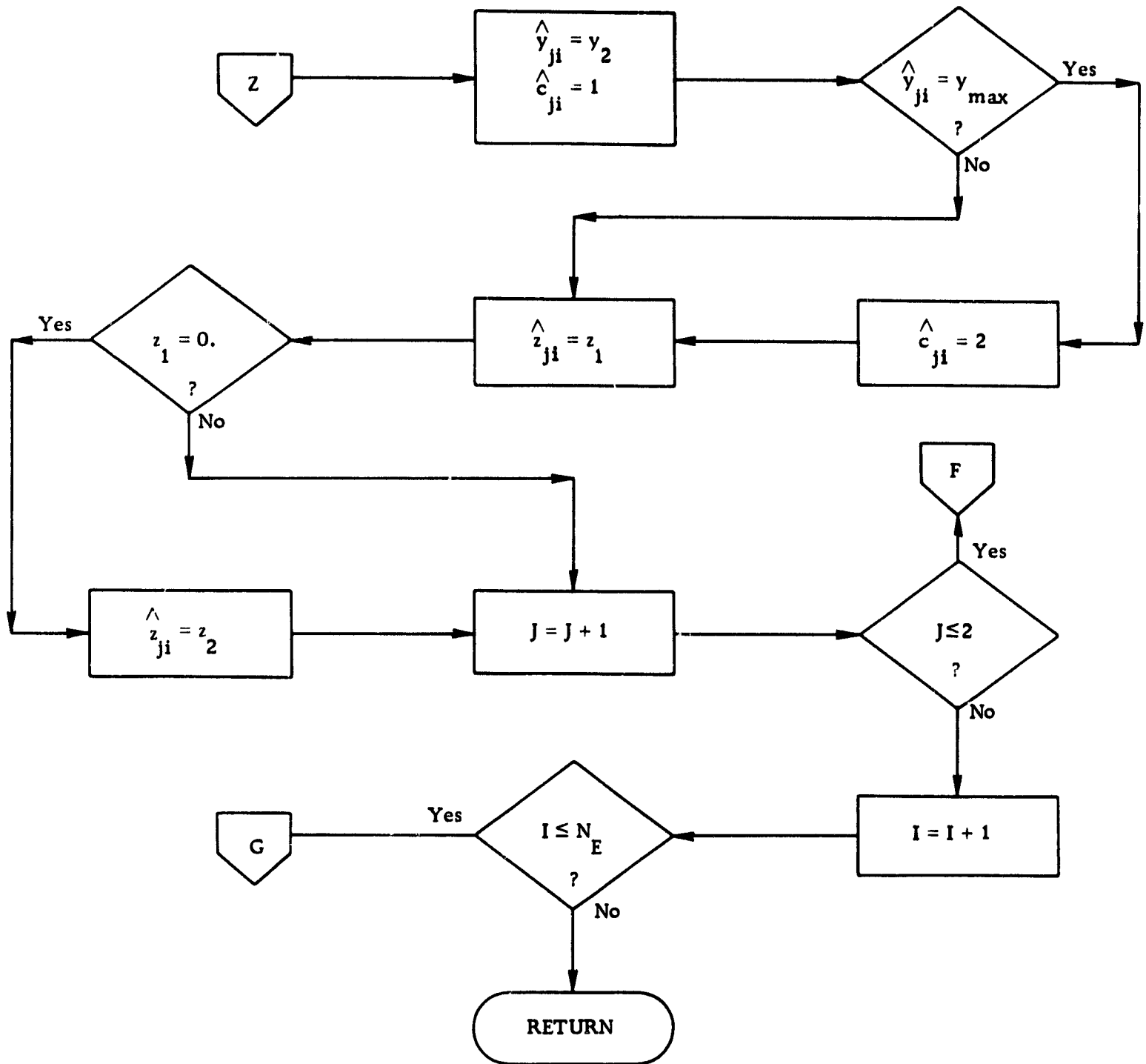
Utility Routines and Common References

COMMON/DATA/
COMMON/ELMAIR/
COMMON/CODRNG/
Utility - None









3.2.11 ZYCUTZ (ZYCUT)

ZYCUTZ determines the point of intersection of the line $y = \alpha$ or the line $z = \alpha$ with a curve of the form $f(z, y) = 0$ or, equivalently, $y = g(z)$.

Calling Sequence

CALL ZYCUT (VHCODE, ALFA, IS, BETA, *, ELNO, SURNO)

VHCODE: 1 Line is $y = \alpha$ (Horizontal)
 2 Line is $z = \alpha$ (Vertical)

ALFA: α

IS: Surface number of curve $f(z, y) = 0$.

BETA: Point of intersection of the form $z = \beta$ if
 VHCODE = 1 and $y = \beta$ if VHCODE = 2.

*: Alternate return if no intersection.

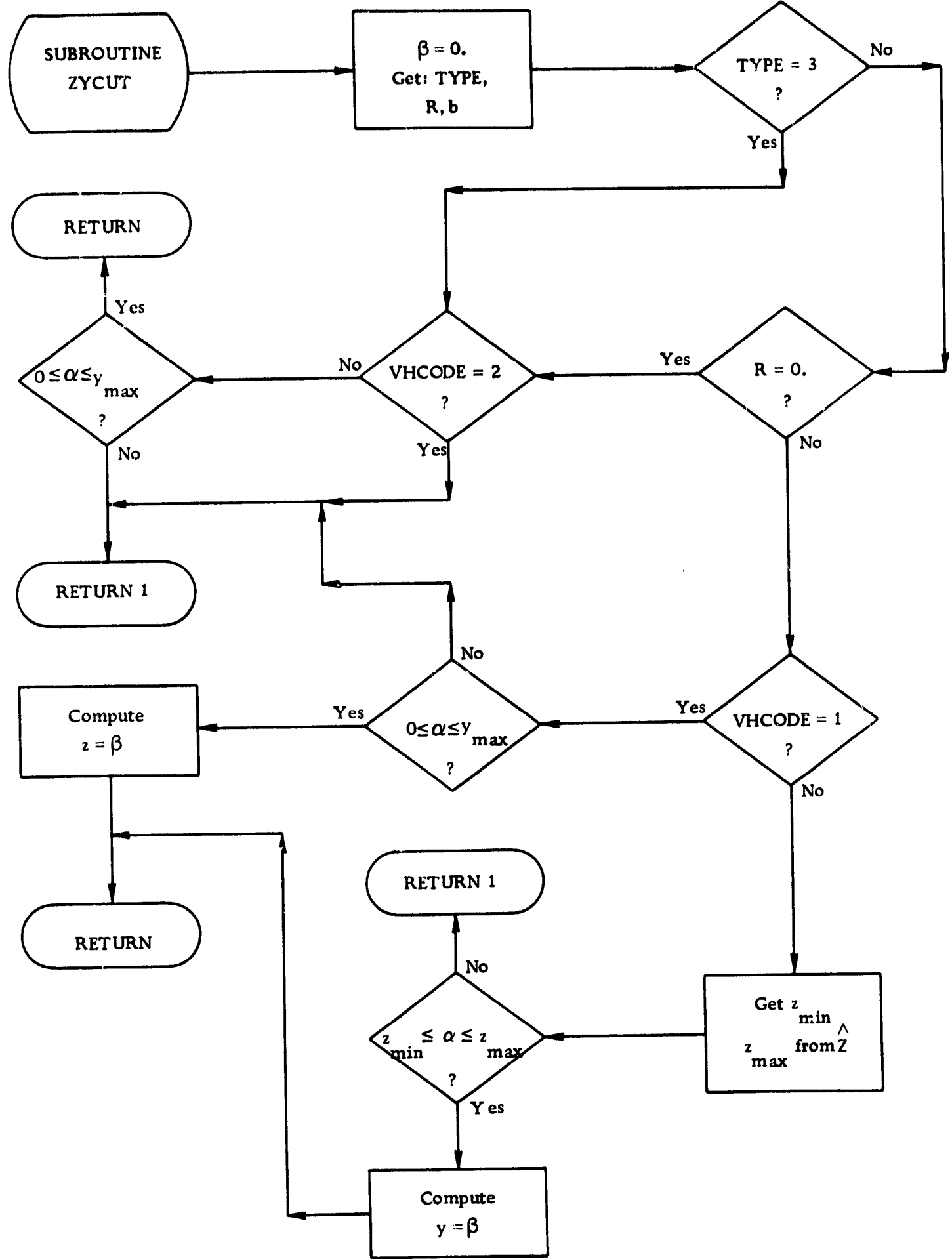
ELNO: Element number of surface in question.

SURNO: 1 Left surface of ELNO
 2 Right surface of ELNO

The parameters ELNO and SURNO are used to obtain bounds from matrices \hat{Z} and \hat{Y} which describe the interval of definition of $f(z, y) = 0$, i. e., $z_{\min} \leq z \leq z_{\max}$ and $0 \leq y \leq y_{\max}$ where

$$y_{\max} = \begin{cases} g(z_{\max}) & z_{\min} = 0 \\ g(z_{\min}) & z_{\min} < 0 \end{cases}$$

If α is outside the interval of definition, ZYCUT invokes the alternate return. If α is in the interval of definition then ZYCUT computes β .



3.3 LINK 3

Link 3 is called upon when the program is executing options 4 and 5 in order to perform twinray diagnostic computations.

3.3.1 TWNR4Z (TWNRY4)

TWNR4Z is called upon by the main program to trace 2 rays from the object point $(h, \theta) = (h_{\max}, 0.)$ where $h_{\max} = h_o + (N_H - 1) \Delta h$. Ray 1 is traced through $(\hat{x}_o, \hat{y}_o) = (0., -1)$ on the unit radius entrance pupil and ray 2 through $(\hat{x}_o, \hat{y}_o) = (0., 1.)$. If the rays are denoted by R_1 and R_2 then TWNR4Z employs RAYTR to compute and save the position and direction of the rays at each element in the system, that is,

$$\{(\vec{P}_i^k, \vec{Q}_i^k); 0 \leq i \leq N_k\} \quad k = 1, 2$$

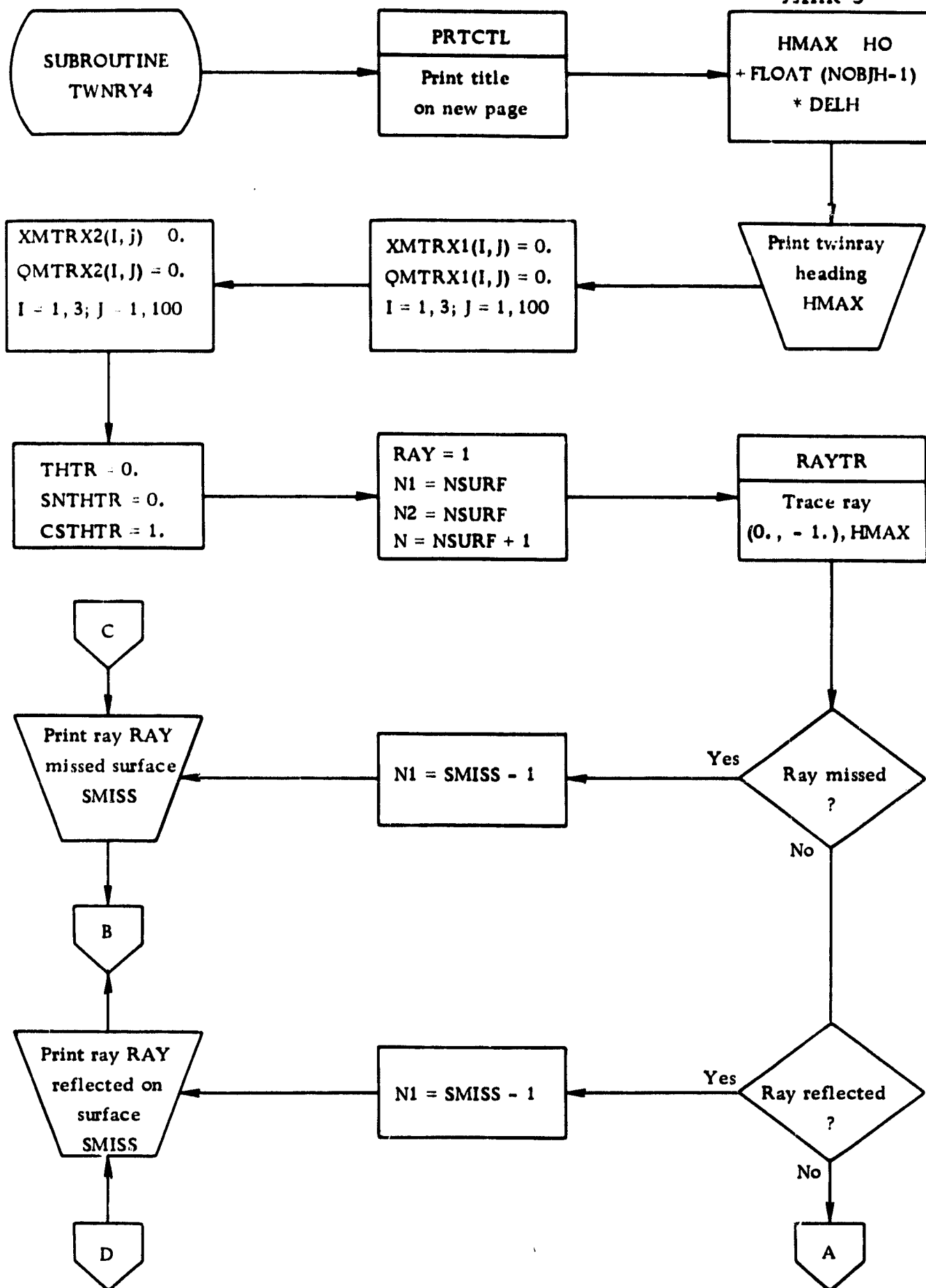
where \vec{P} represents position and \vec{Q} direction. If there is a ray failure (miss or reflect) then $N_k < N$ and a message is printed. The position and direction data are used to compute thickness, λ_i^k , aperture radius, A_{Ri} , and aperture location, A_{Li} , these quantities being listed in tabular form.

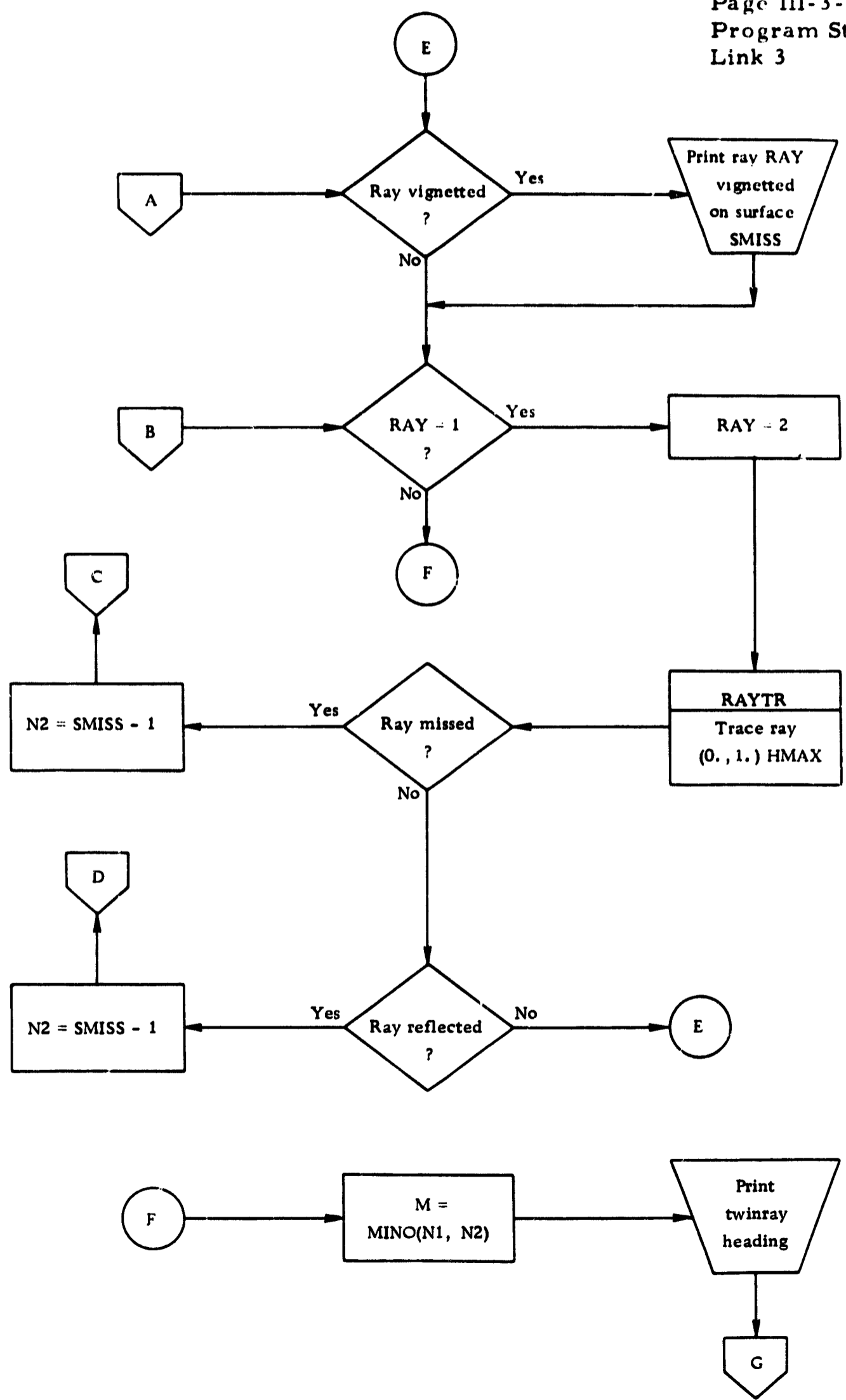
Calling Sequence

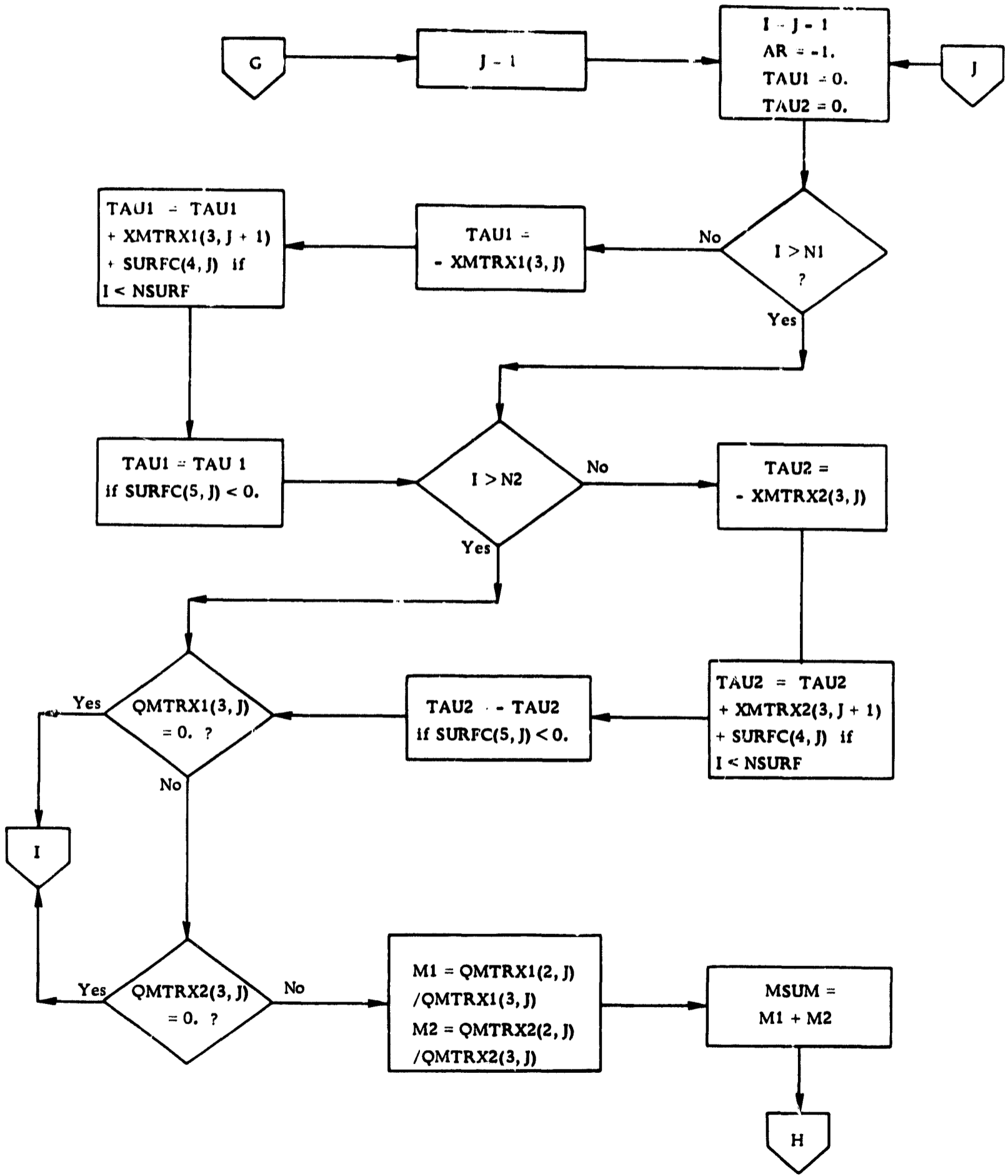
CALL TWNRY4

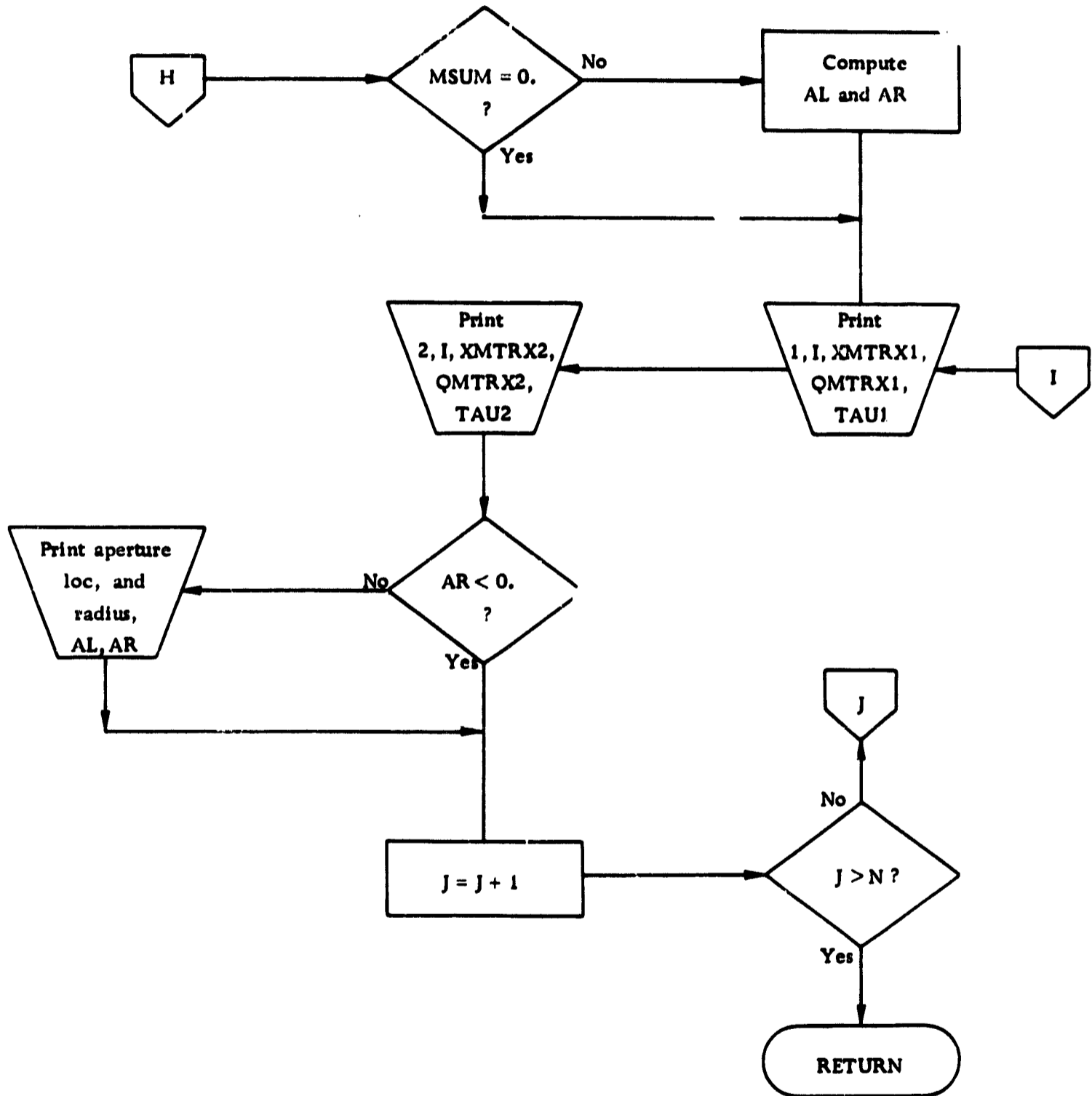
Utility Routines and Common References

COMMON/DATA/
COMMON/AZOBJ/
COMMON/PRNT/
COMMON/TMTWIN/
PRTCTL
RAYTR









3.3.2 TWNR5Z (TWNRY5)

TWNR5Z is called upon by the main program to trace 2 rays from object point $(h, \theta) = (0., 0.)$. Ray 1 is traced through $(\hat{x}_o, \hat{y}_o) = (0., \Omega_2)$ on the entrance pupil and ray 2 is traced through $(\hat{x}_o, \hat{y}_o) = (0., 1.)$. RAYTR is used to compute and save the position and direction of the rays at each element in the system, that is,

$$\{(\vec{P}_i^k, \vec{Q}_i^k); 0 \leq i \leq N_k, k = 1, 2\}$$

where \vec{P} is the position and \vec{Q} is the direction. If a ray misses or reflects then $N_k < N$ and a message is printed. If a ray vignets a message is printed, but $N_k = N$.

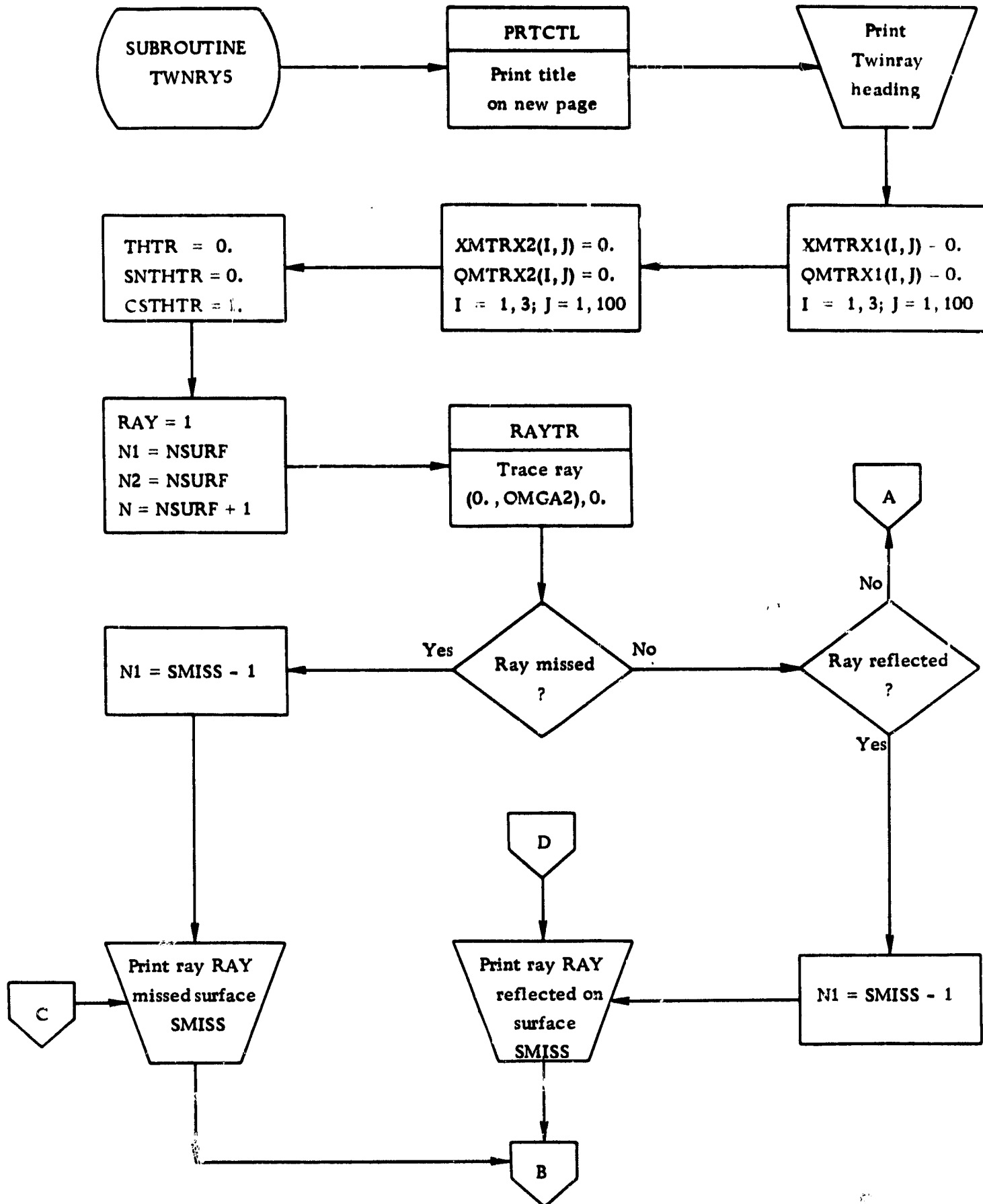
The data for position is stored in arrays XMTRX1 and XMTRX2. The direction data is contained in arrays QMTRX1 and QMTRX2. The thickness is also computed and printed in tabular form along with the position and direction values for each of the two rays at each surface. TWNR5Z must be called additional times if there are substitution sets.

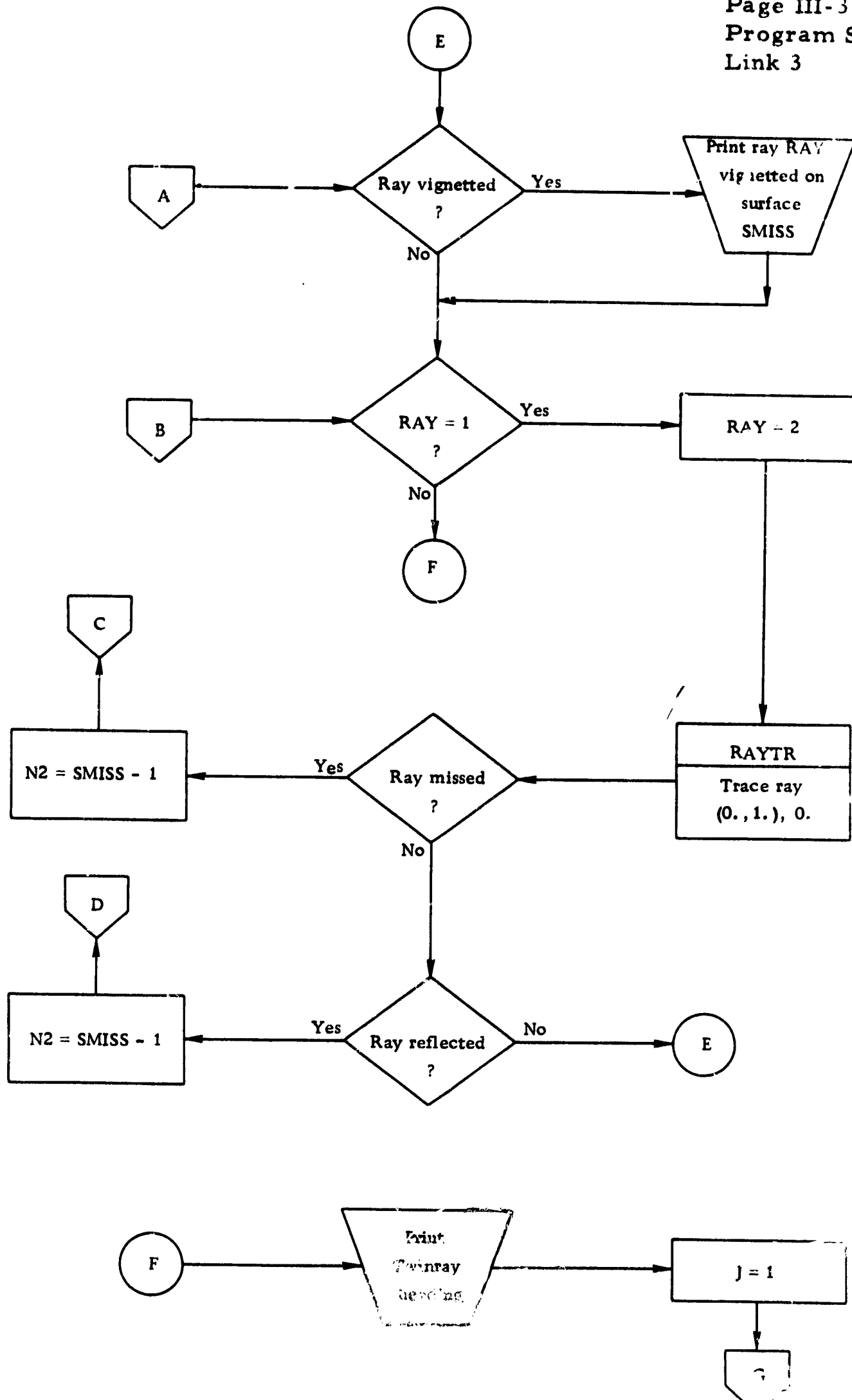
Calling Sequence

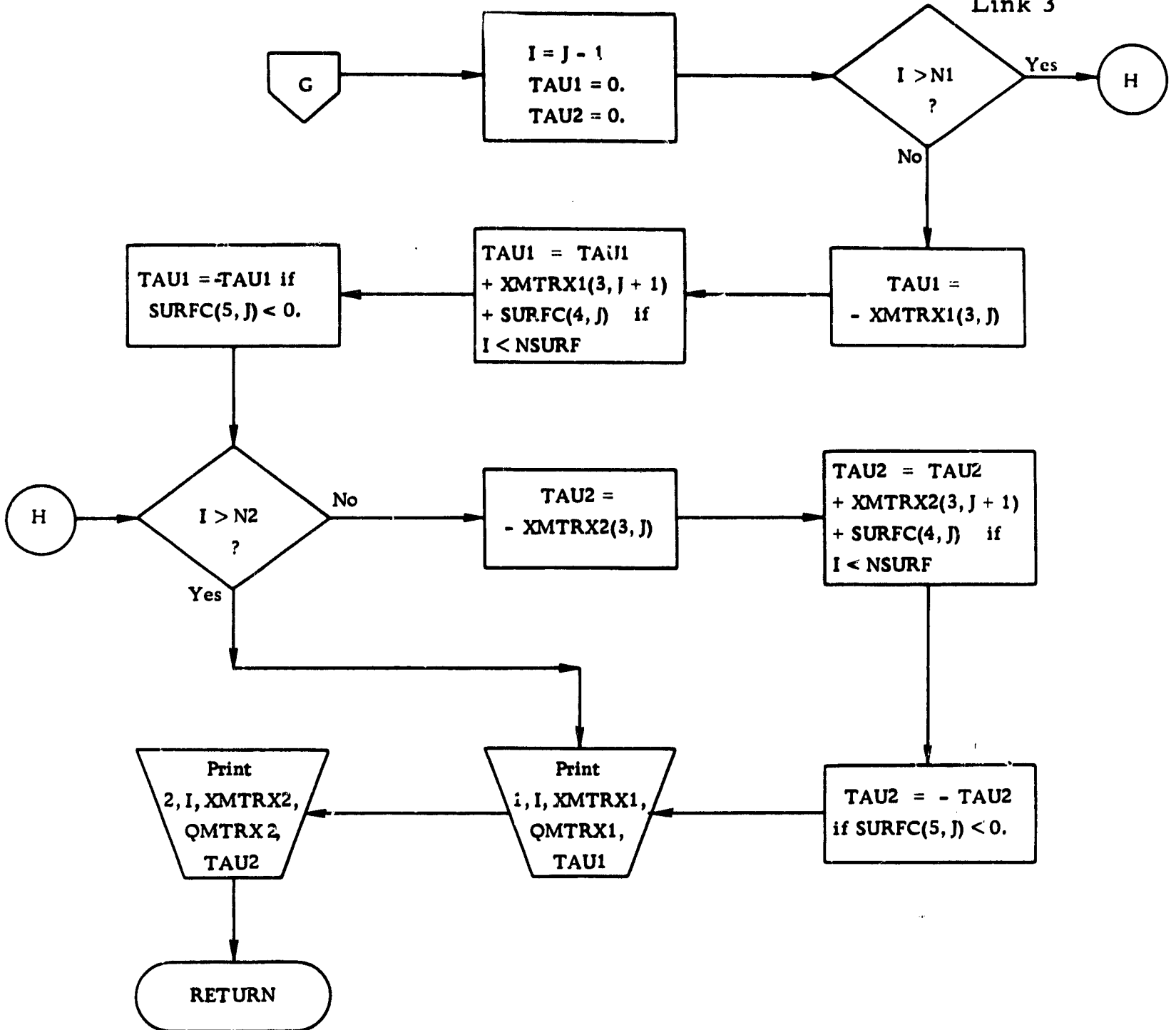
CALL TWNRY5

Utility Routines and Common References

COMMON/DATA/
COMMON/AZOBJ/
COMMON/PRNT/
COMMON/TMTWIN/
RAYTR
PRTCTL







3.4 LINK 4

Link 4 is essentially an interface link between the several lower level links and the ray trace subroutine (RAYTR). By appropriate communication with subroutine SMPOP the complete pattern of rays is traced and a variety of statistical information returned to the calling program including such things as spot size, rms x, rms y etc. This link is employed by options 2, 3, 7, and 8.

3.4.1 LATTZZ (LATT)

LATTZZ is used to generate the ray pattern by producing 1 ray for each entry. The coordinates are stored in (x_o, y_o) . MRAYS, which must be set to zero by the calling program prior to the first entry to LATTZZ, is incremented by 1 for each entry and contains the current count of the number of rays which have been generated. To generate the entire pattern LATTZZ must be entered NRAYS times. It is assumed that y_o does not change between calls.

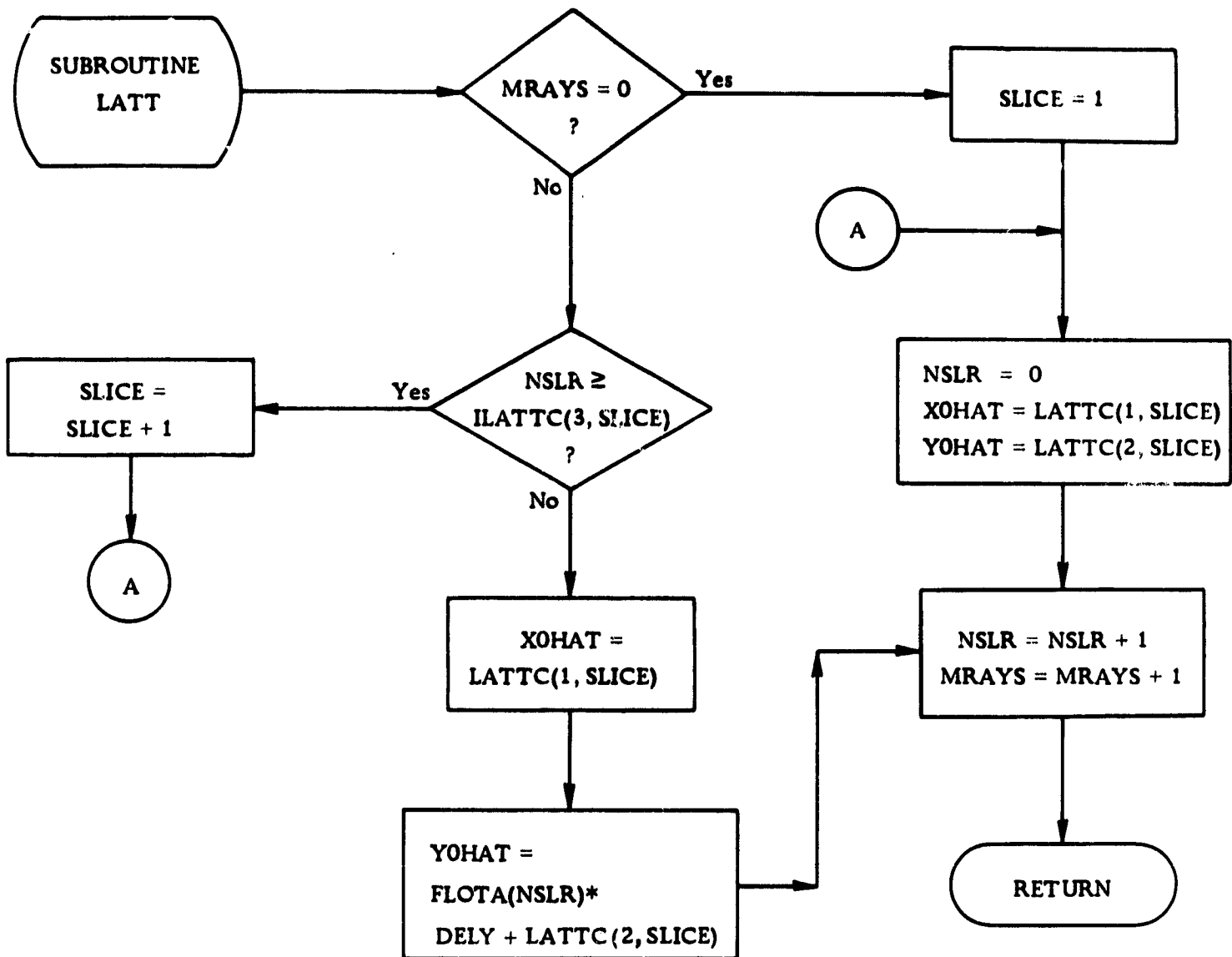
Calling Sequence

CALL LATT (XOHAT, YOCHAT, MRAYS)

XOHAT: Storage for x_o .
YOCHAT: Storage for y_o .
MRAYS: Ray number being generated.
SLICE: Current slice number.
NSLR: Ray number within current slice.

Utility Routines and Common References

COMMON/DATA/



3.4.2 MERIDZ (MERID)

MERIDZ traces a ray from the object point $(0., 0.)$ through $(x_o, y_o) = (0., \Omega_1)$ on the entrance pupil to $P_1 = (x_1, y_1, z_1)$ on σ_N with direction $\vec{Q}_1 = (Q_{x1}, Q_{y1}, Q_{z1})$. Using P_1 and \vec{Q}_1 , MERIDZ computes B_F and $f/$. If MERIDZ returns a value of zero for either B_F and $f/$ this should be interpreted as $B_F = \infty$ or $f/ = \infty$. The run is terminated if the ray misses or reflects.

Calling Sequence

CALL MERID(COLOR, BFOCUS, FSNUMB)

COLOR: Color number currently used.

BFOCUS: Storage for back focus, B_F .

FSNUMB: Storage for f/number, $f/$.

THTR: 0. (θ)

SNTHTR: 0. ($\sin \theta$)

CSTHTR: 1. ($\cos \theta$)

} must be set by calling program

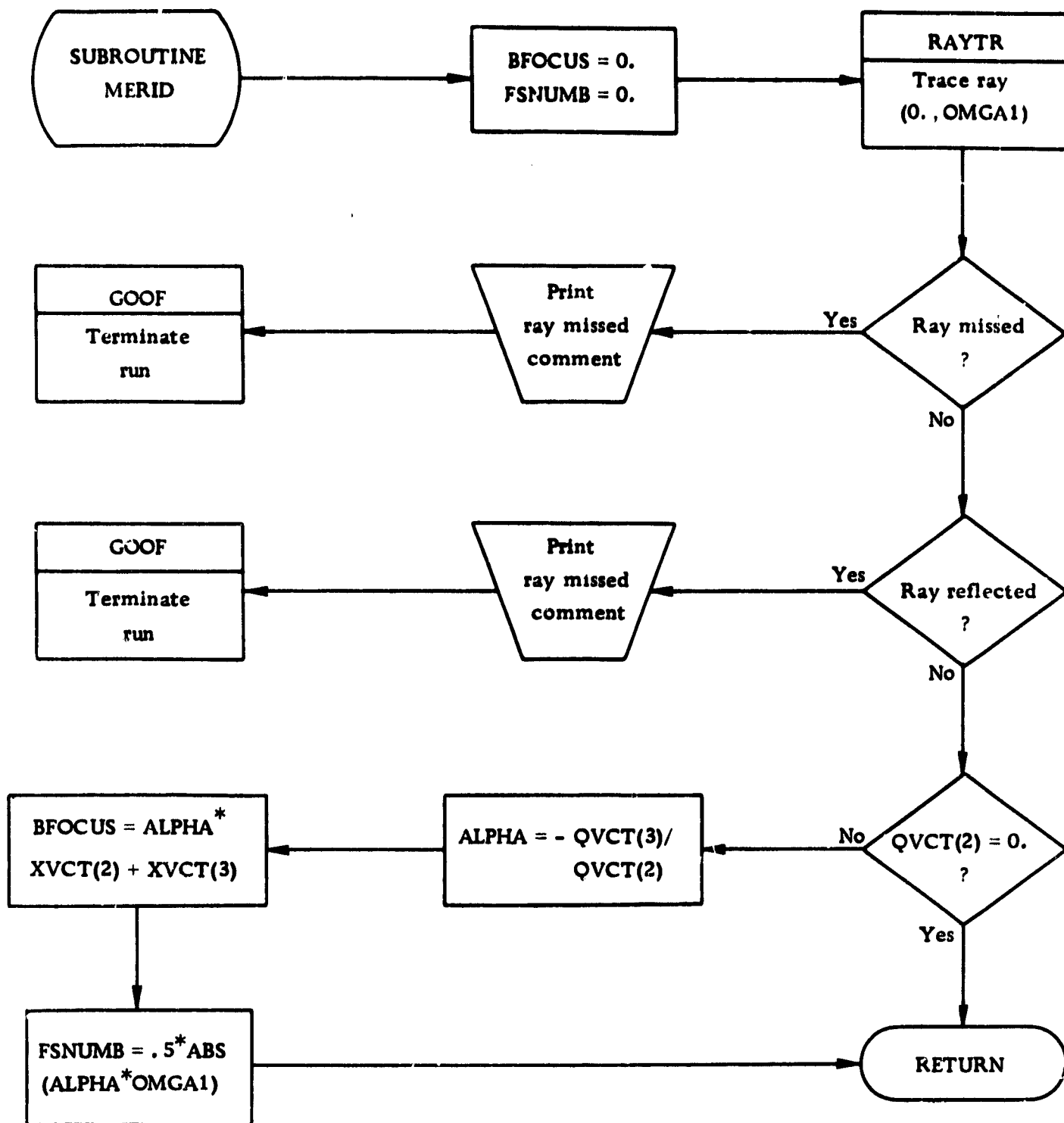
Utility Routines and Common References

COMMON/DATA/

COMMON/AZOBJ/

RAYTR

GOOF



3.4.3 SMPOPZ (SMPOP)

For a specified combination of object point (h_j, θ) and color i , SMPOPZ traces the entire pattern of rays and returns the information listed below, the various parameters being defined in Volume I, Section 6.

S_j^* Image Plane Position
 N_M Number of misses
 N_{RF} Number of reflections
 N_V Number of vignets
 \bar{N}_R Number of successes

A vignettted ray is considered to be a failure in all instances except the design option (CMFLAG = 2 or 5). The remaining outputs from SMPOP depend upon the value assigned to the input trigger, CMFLAG, as follows:

1) CMFLAG = 1

$\bar{m}_x, \bar{k}_x, \bar{m}_y, \bar{k}_y$
 A_x, B_x, \dots, B, C
 \tilde{m}_{xi} (XCURL) $1 \leq i \leq \bar{N}_R$
 \tilde{m}_{yi} (XCURL) " "
 \tilde{k}_{xi} (TEMPS) " "
 \tilde{k}_{yi} (TEMPS(200)) " "

2) CMFLAG = 2

$\bar{x}(S_j^*), \bar{y}(S_j^*)$
 $\tilde{x}_i(S_j^*)$ (XCURL) $1 \leq i \leq \bar{N}_R$
 $\tilde{y}_i(S_j^*)$ (YCURL) " "

3) CMFLAG = 3

$$\begin{aligned} & \bar{x}(S_j^*), \bar{y}(S_j^*) \\ & \tilde{x}_i(S_j^*) \quad (\text{XCURL}) \quad 1 \leq i \leq \bar{N}_R \\ & \tilde{y}_i(S_j^*) \quad (\text{YCURL}) \quad " \quad " \end{aligned}$$

$$\text{TEMPS}(i) = \begin{cases} 0. & \text{Ray } i \text{ is ok.} \\ 1. & \text{Ray } i \text{ is missed.} \\ 2. & \text{Ray } i \text{ is reflected.} \\ 3. & \text{Ray } i \text{ is vignetted.} \end{cases}$$

4) CMFLAG = 4

$$\begin{aligned} & \bar{x}(S_j^*), \bar{y}(S_j^*) \\ & x^*(S_j^*), y^*(S_j^*), R^*(S_j^*) \\ & \tilde{x}_i(S_j^*) \quad (\text{XCURL}) \quad 1 \leq i \leq \bar{N}_R \\ & \tilde{y}_i(S_j^*) \quad (\text{YCURL}) \quad " \quad " \end{aligned}$$

5) CMFLAG = 5

$$\begin{aligned} & \bar{m}_x, \bar{k}_x, \bar{m}_y, \bar{k}_y \\ & \tilde{m}_{xi} \quad (\text{XCURL}) \quad 1 \leq i \leq \bar{N}_R \\ & \tilde{k}_{xi} \quad (\text{YCURL}) \quad " \quad " \\ & \tilde{m}_{yi} \quad (\text{TEMPS}) \quad " \quad " \\ & \tilde{k}_{yi} \quad (\text{TEMPS}(200)) \quad " \quad " \end{aligned}$$

Calling Sequence

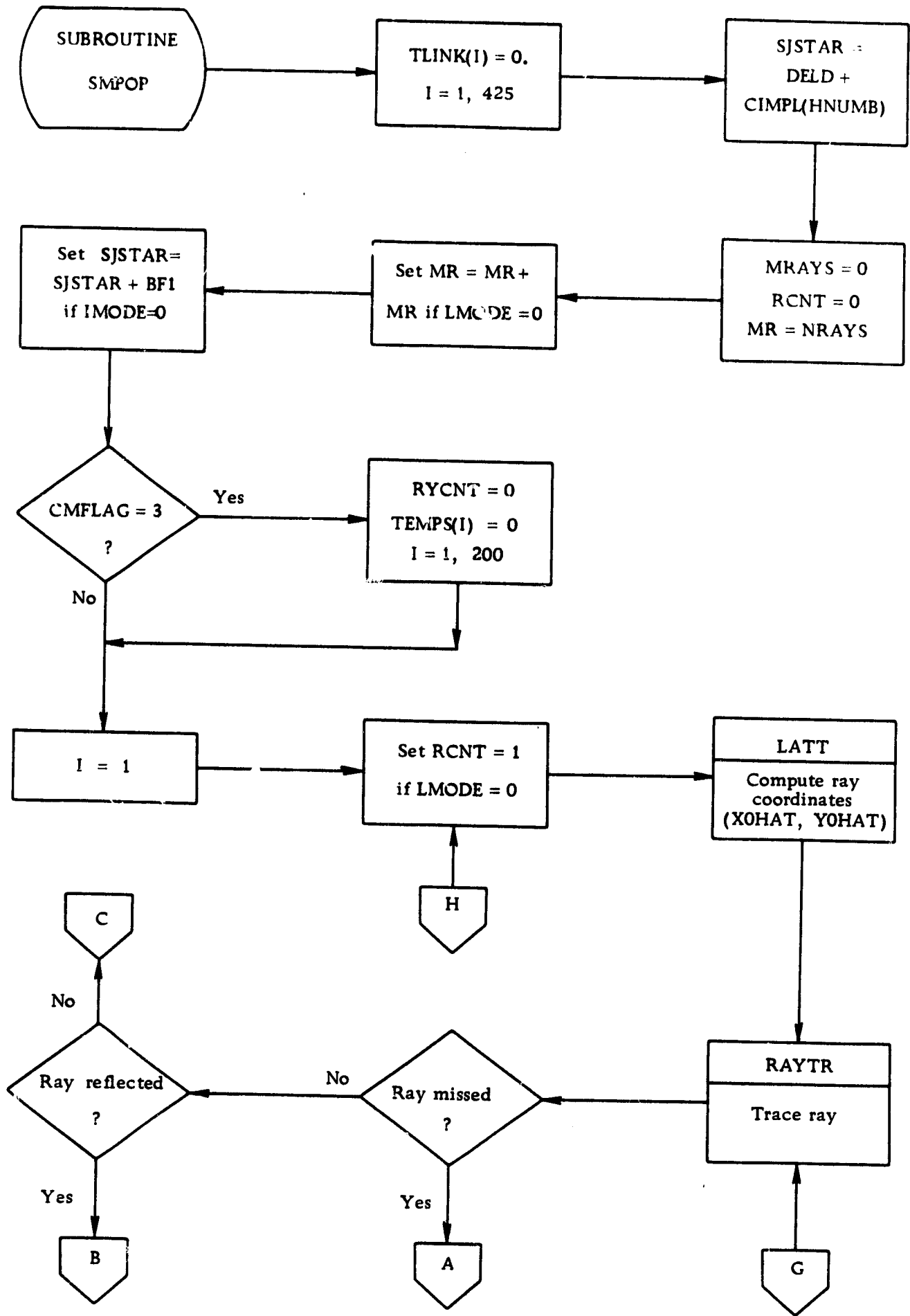
CALL SMPOP (COLOR, HNUMB, HEIGHT, BF1, CMFLAG, TEMPS

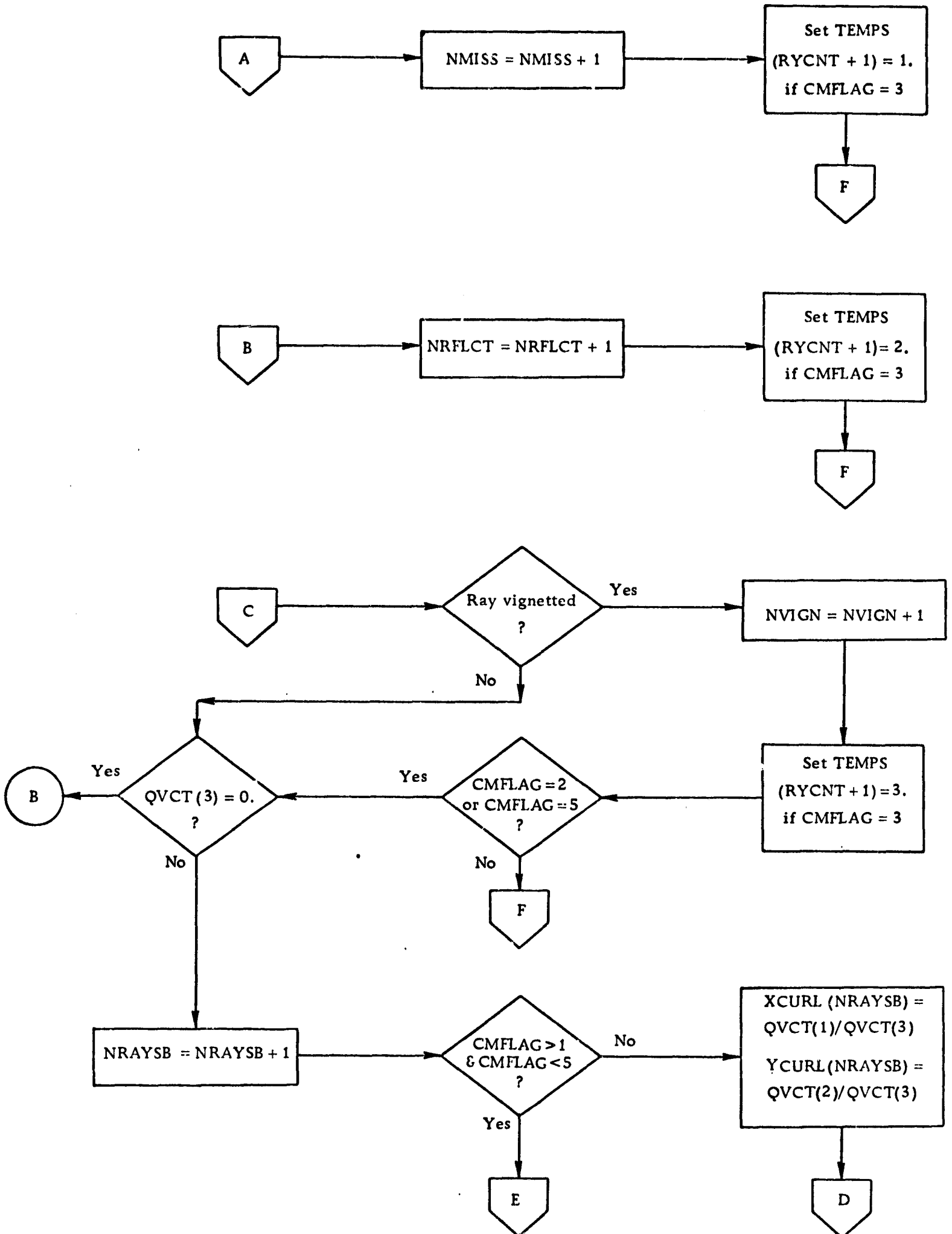
COLOR : Color index i
HNUMB : Object point index j
HEIGHT : Object height h_j
BF1 : Back focus color l
CMFLAG : Computation flag
TEMPS : Used if CMFLAG = 1, 3, or 5

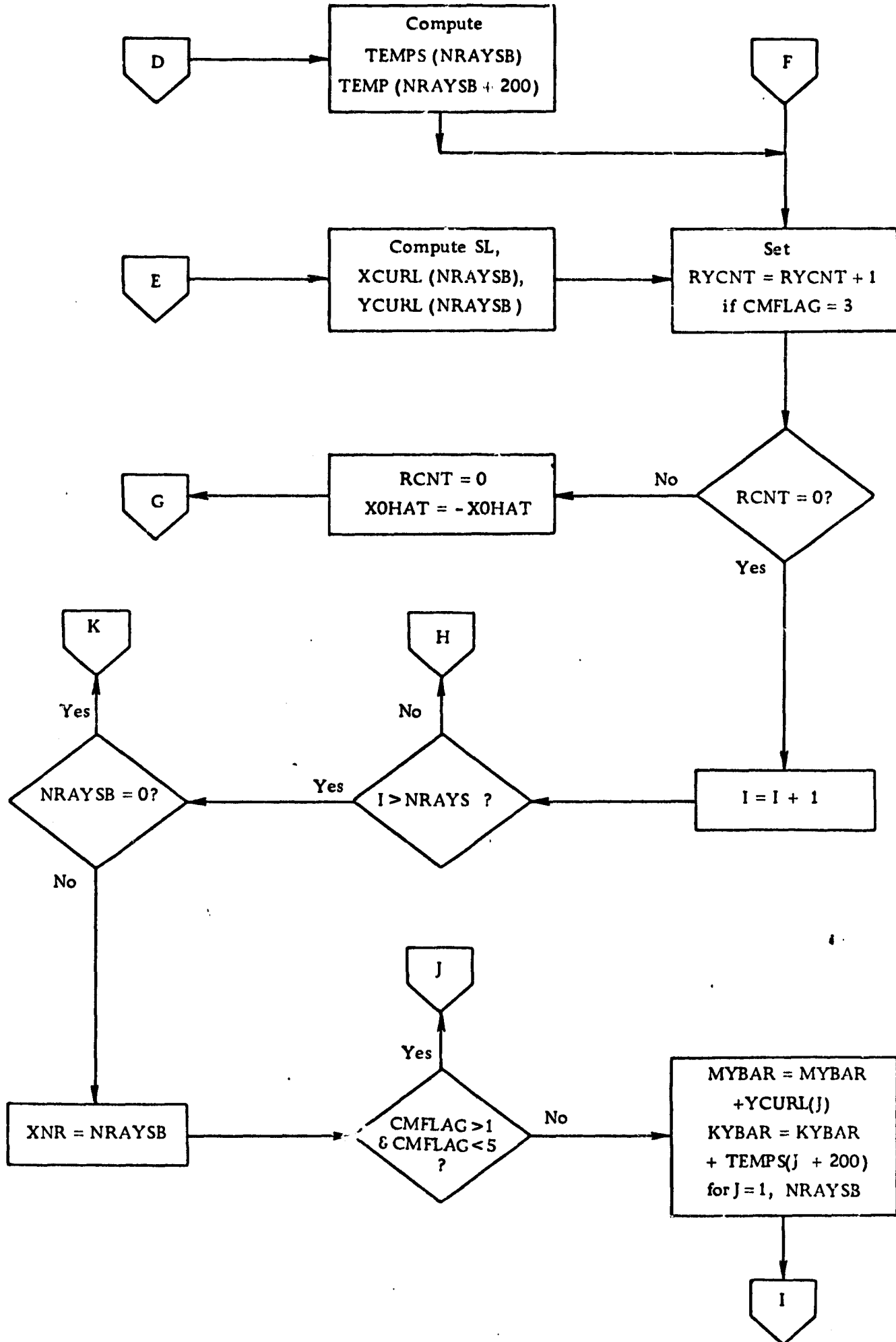
The remaining variables are transmitted through TMPATT COMMON as seen on page III-2-10.

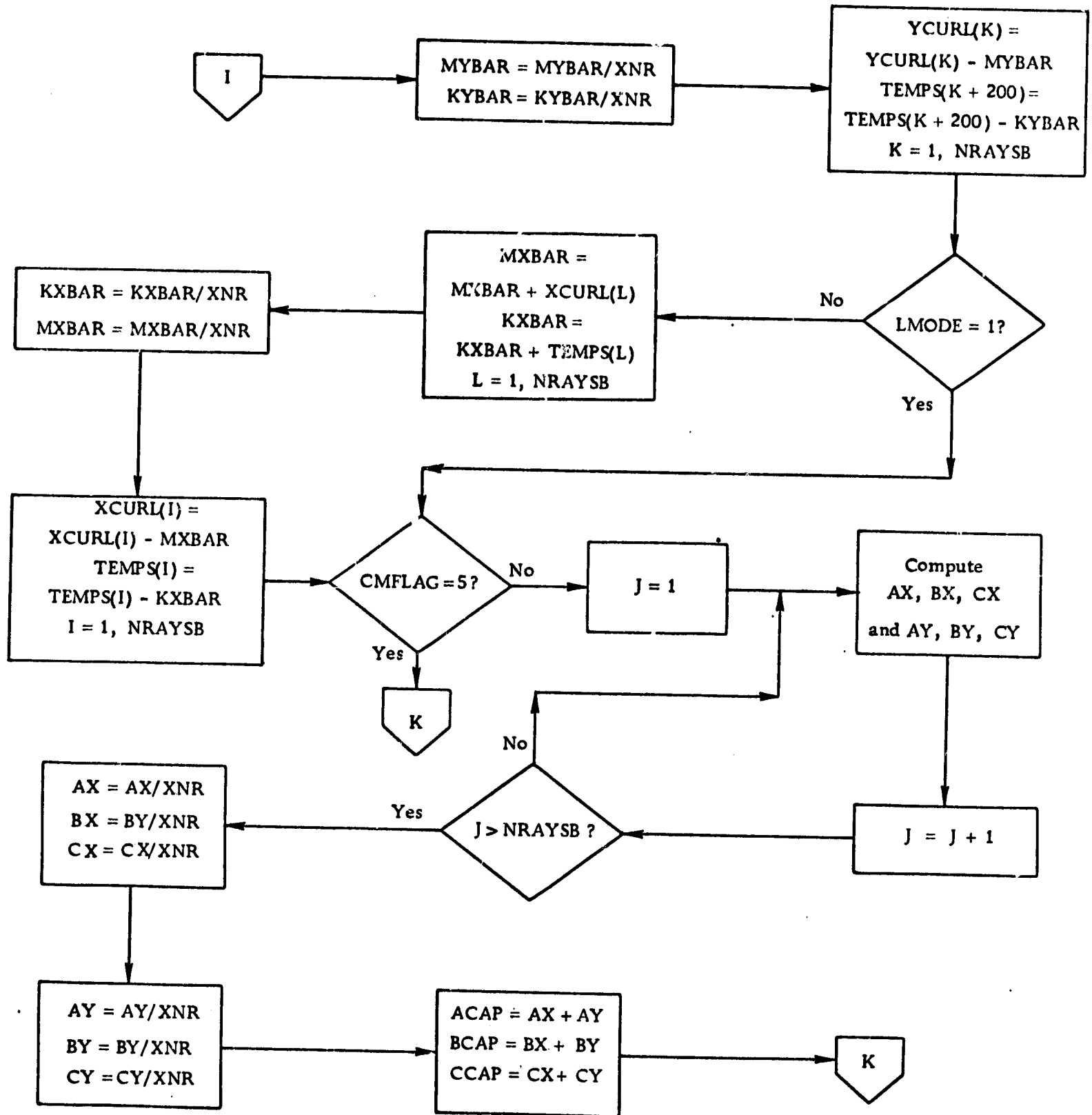
Utility Routines and Common References

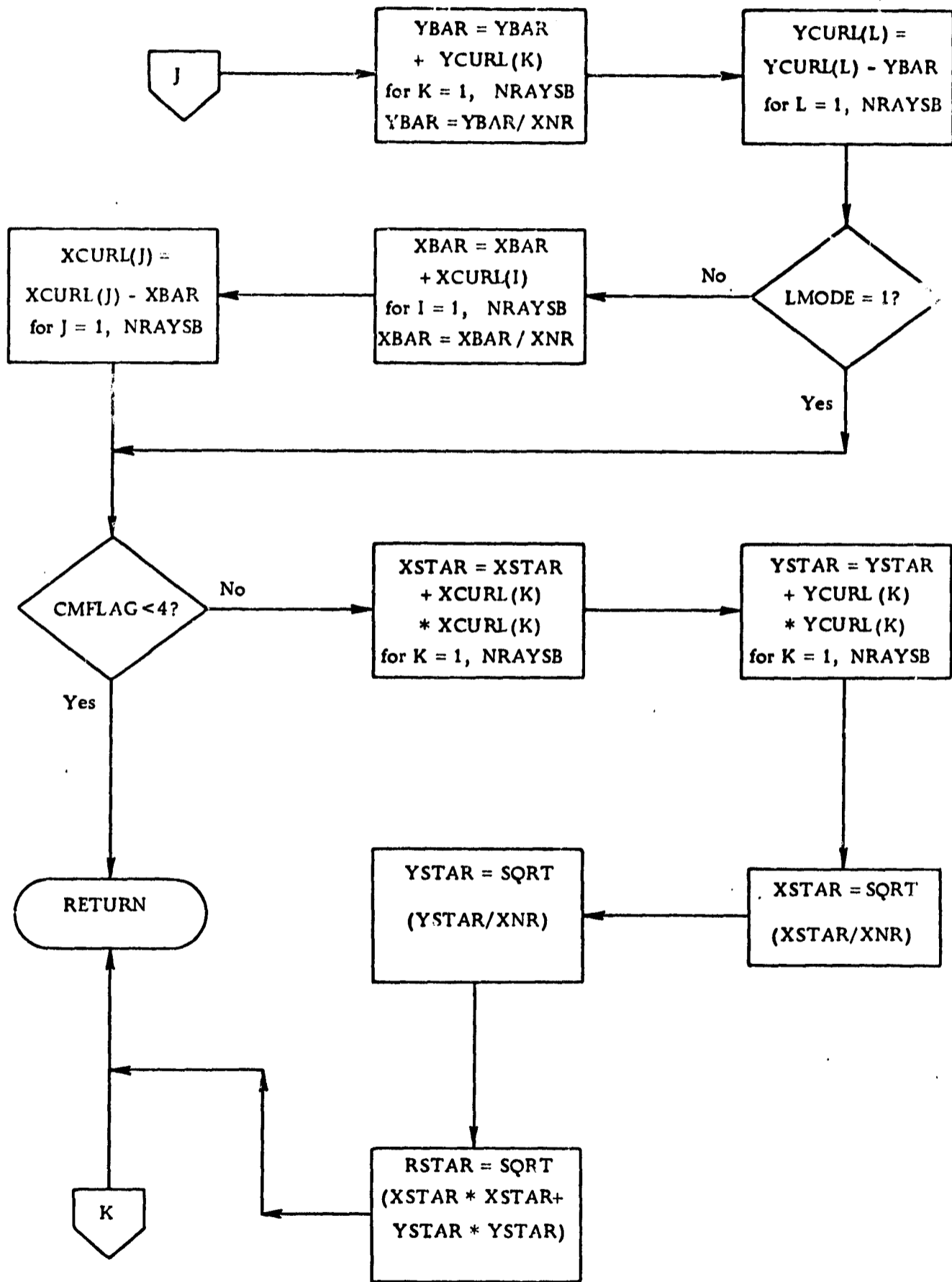
COMMON/DATA/
COMMON/AZOBJ/
COMMON/TMPATT/
RAYTR
LATT











3.5 LINK 5

Link 5 is responsible for performing design computations and is entered whenever option 2 is specified. The downward communication is directly from FOLDP to STAR2 which is the design control subroutine. Link 5 communicates back up to subroutine RAYTR by going through link 4. Control is returned to FOLDP upon completion of the design process or abandonment due to unexpected failure.

3.5.1 BNDCHZ (BNDCHK)

BNDCHZ checks the "improved" lens system for boundary violations using the parameters of the current design variable. The checking includes those constraints defined by the boundary condition matrix, BOUNDS, provided NCOND > 0. The lens length is checked if SYSMX is non zero, and the system is checked for feathering.

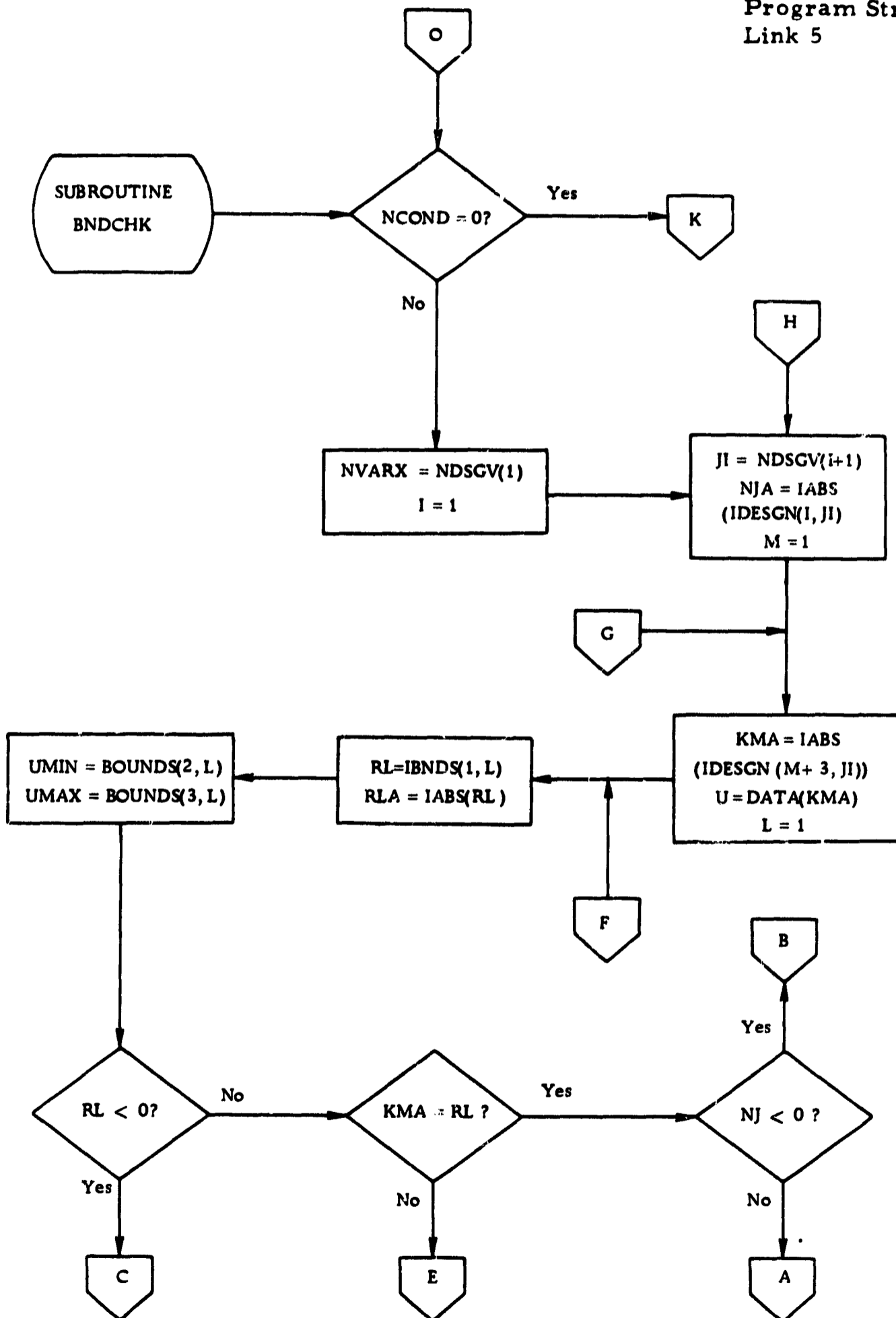
When a boundary violation has occurred an appropriate message is printed, and BNDCHK executes the alternate return signifying the detection of an error condition. The error message print out can be suppressed by setting IFLAG to a non-zero value when calling BNDCHK.

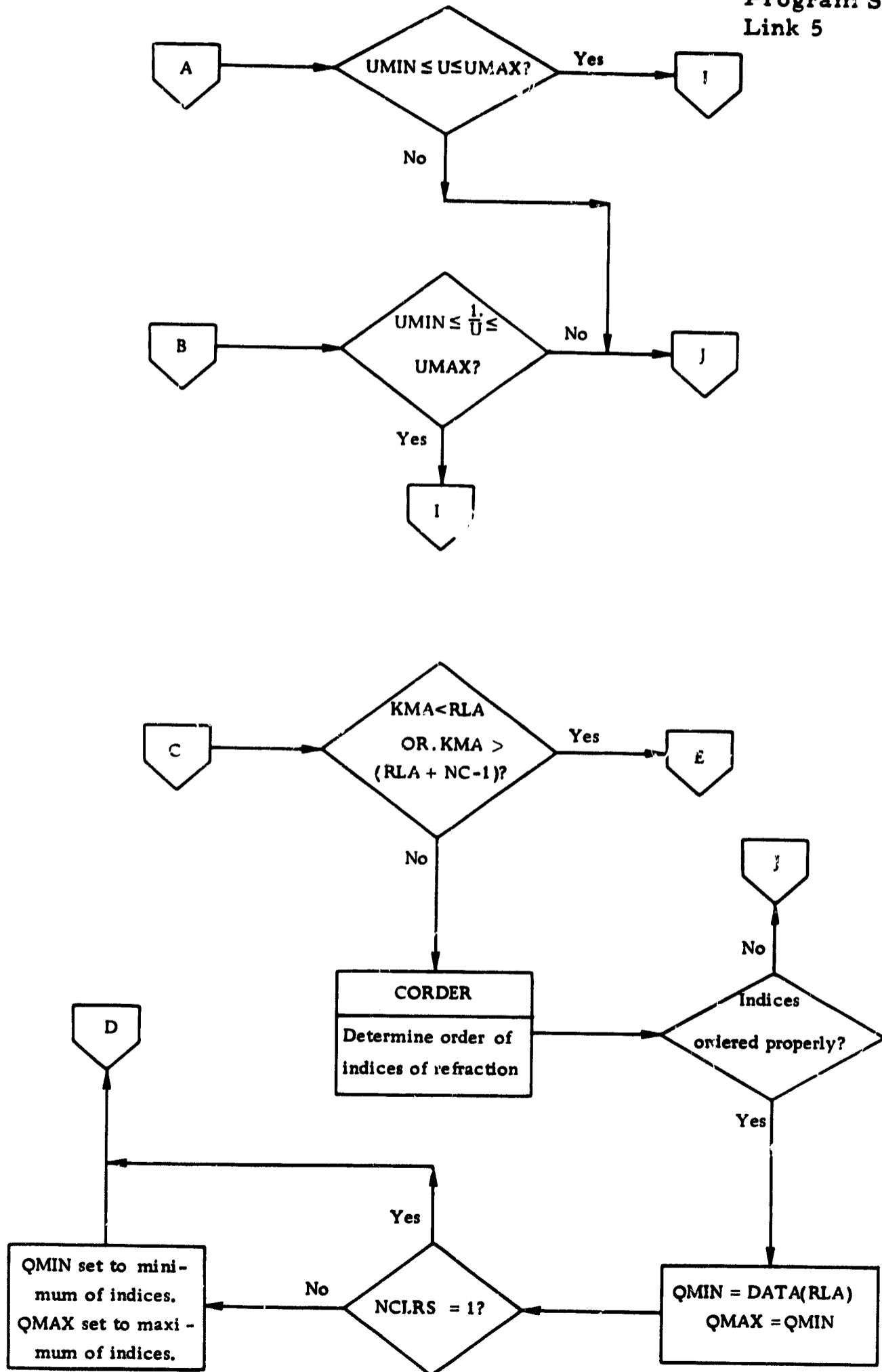
Calling Sequence

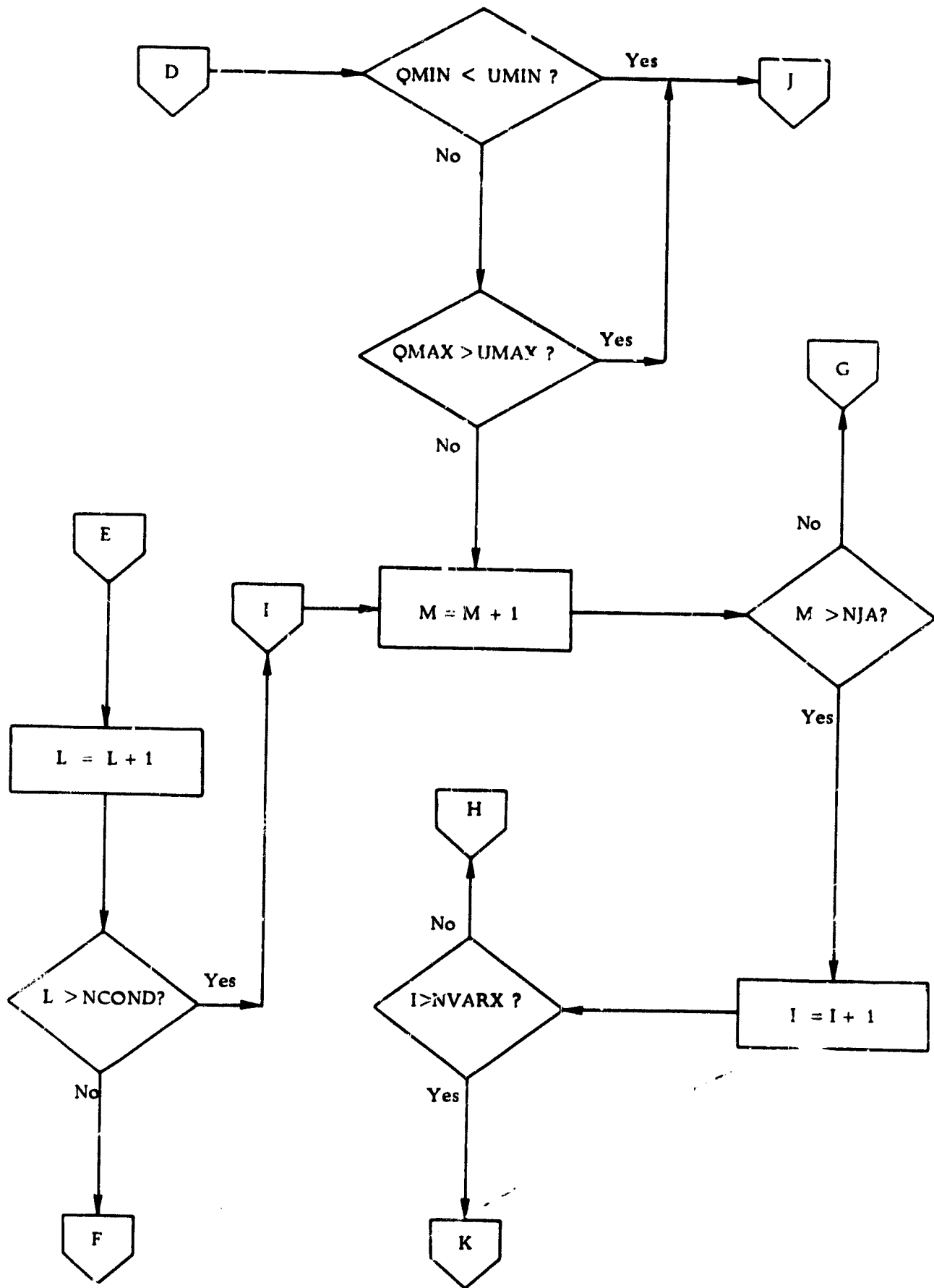
CALL BNDCHK (*, IFLAG)
* Return executed on detection of a boundary violation.
IFLAG: Flag set to non-zero if suppression of error message printout is desired.

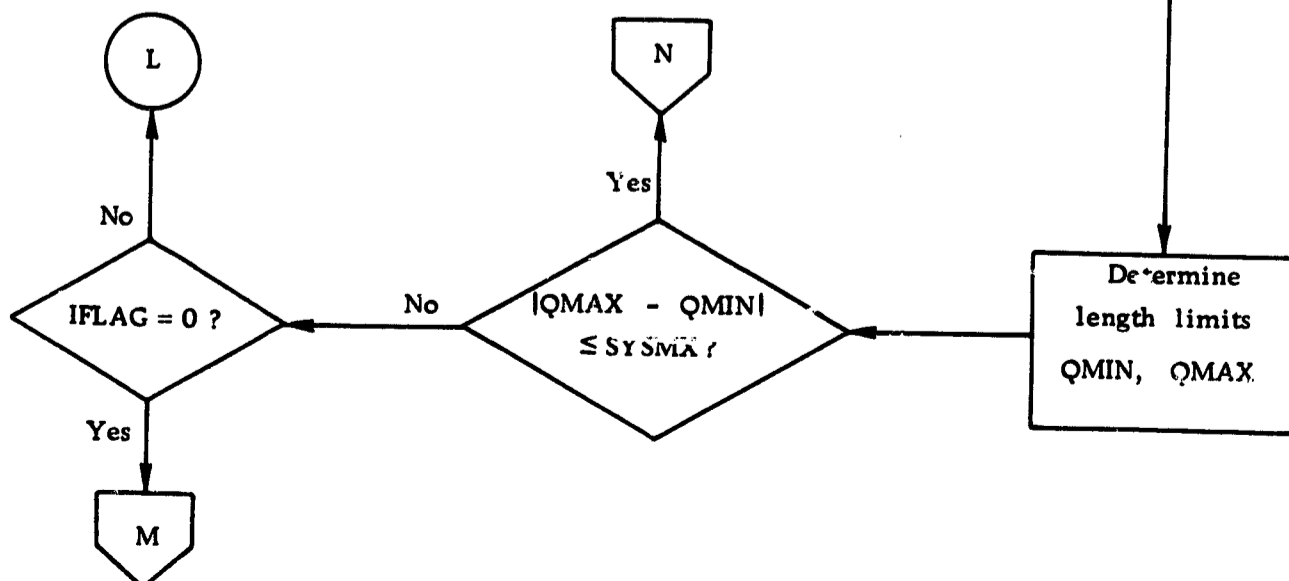
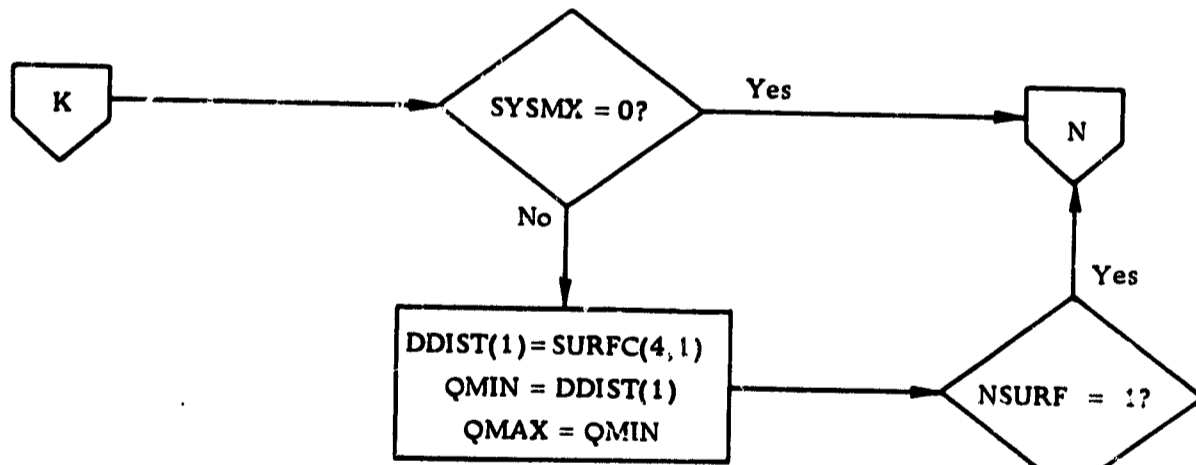
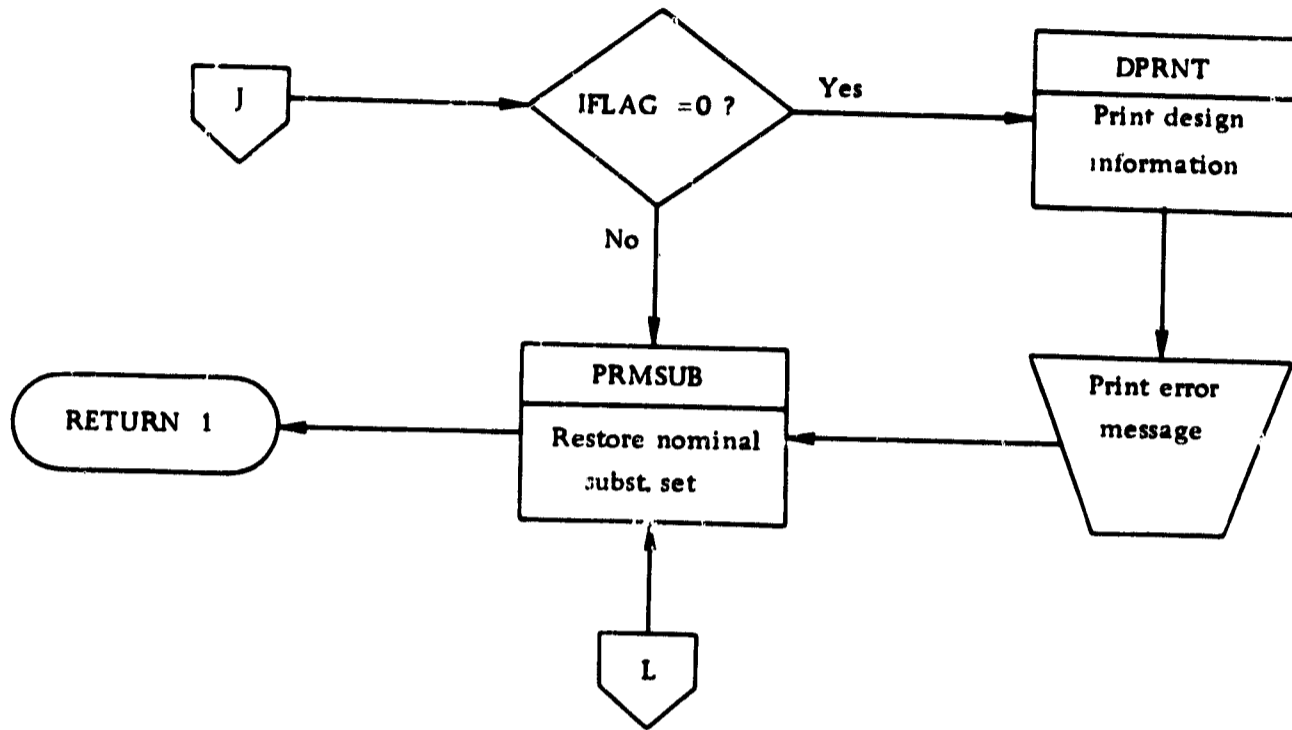
Utility Routines and Common References

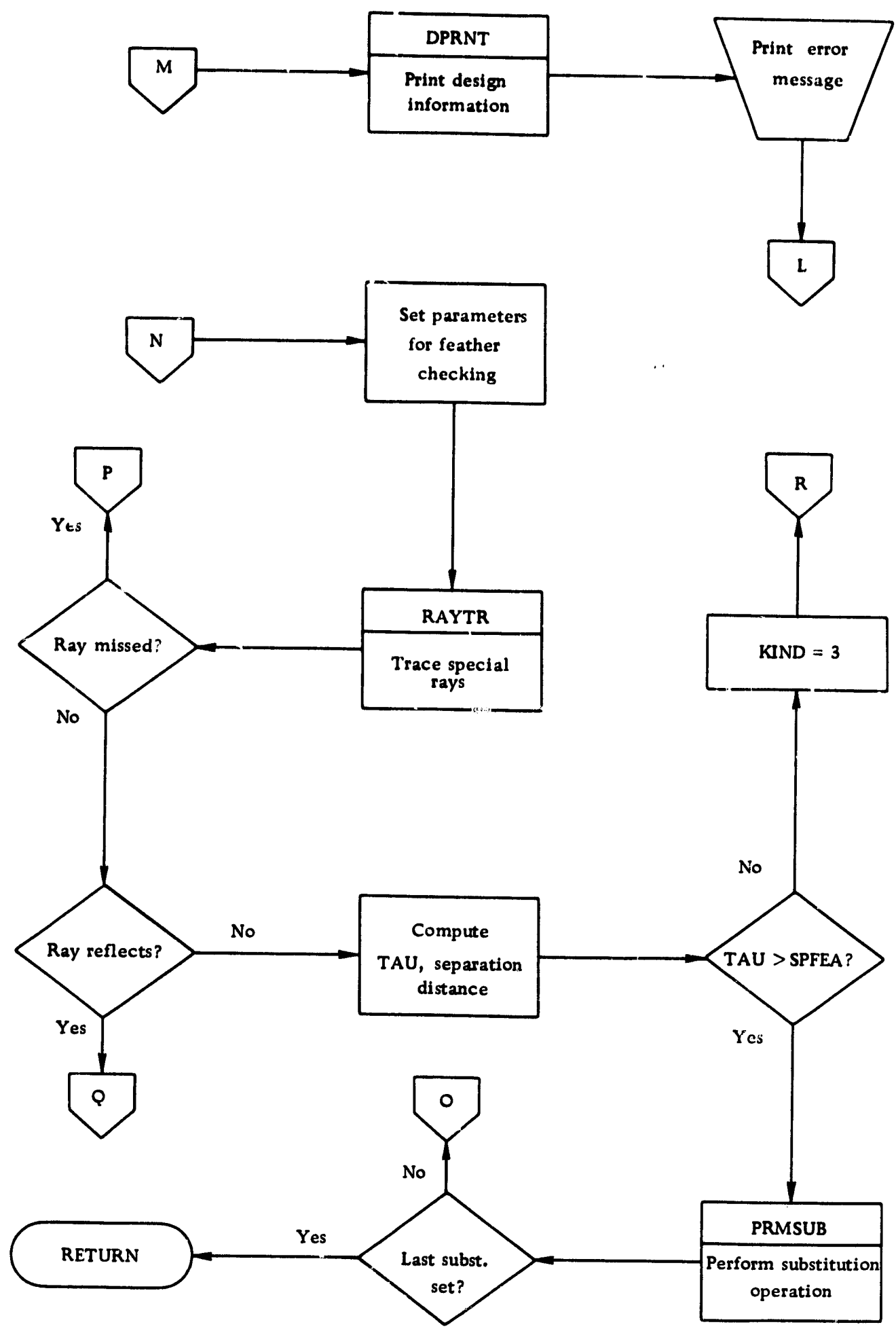
COMMON/DATA/
COMMON/PRNT/
COMMON/AZOBJ/
COMMON/TMDESN/
CORDER
DPRNT
PRMSUB
RAYTR

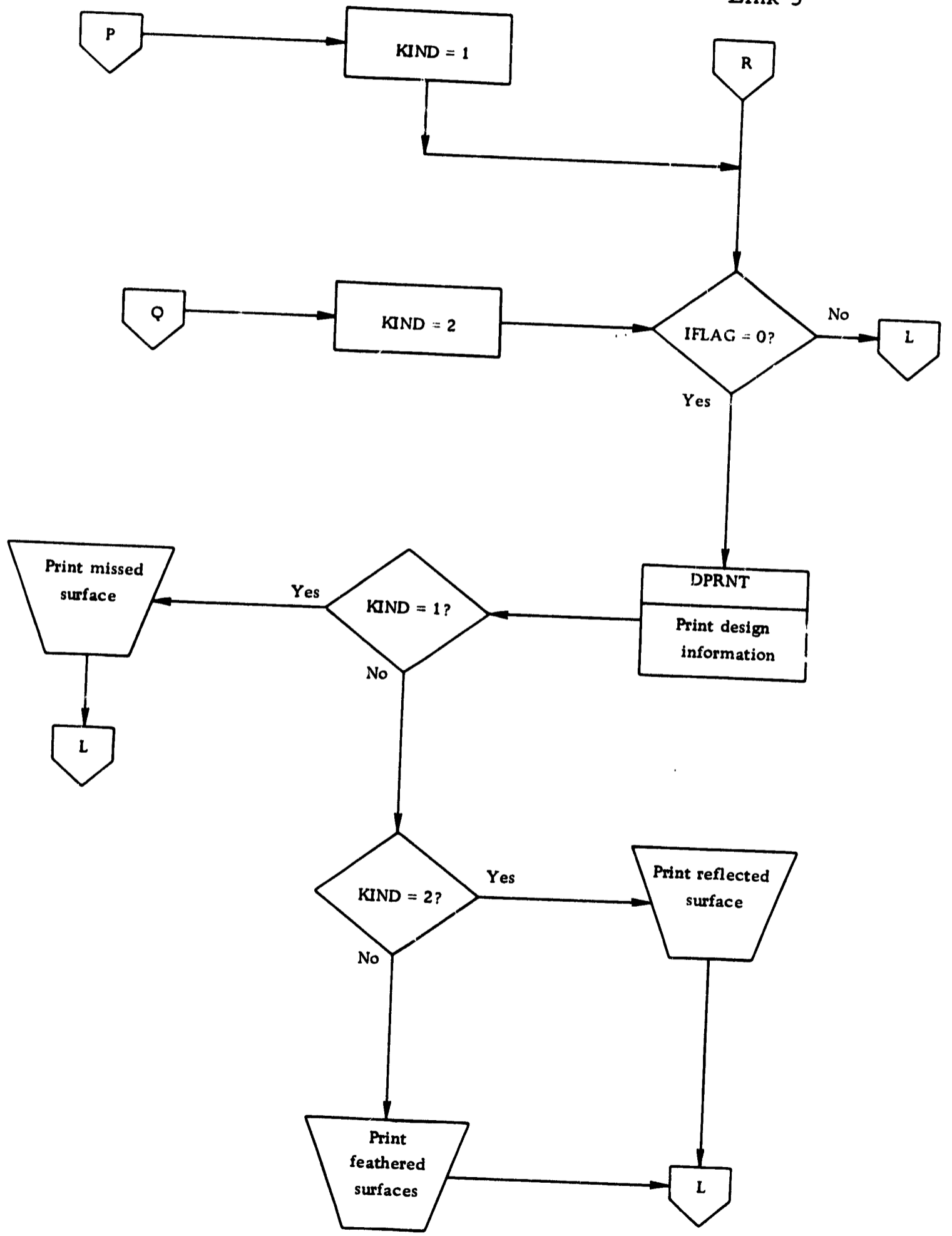












3.5.2 CORDEZ (CORDER)

CORDEZ is used to determine the relative ascending order of a set of N indices of refraction $1 \leq N \leq 6$. An integer vector corresponding to the set of indices is calculated such that each element of the vector is the ordered position of the corresponding index within the set.

Calling Sequence

CALL CORDER (N, DVCT, OVCT)

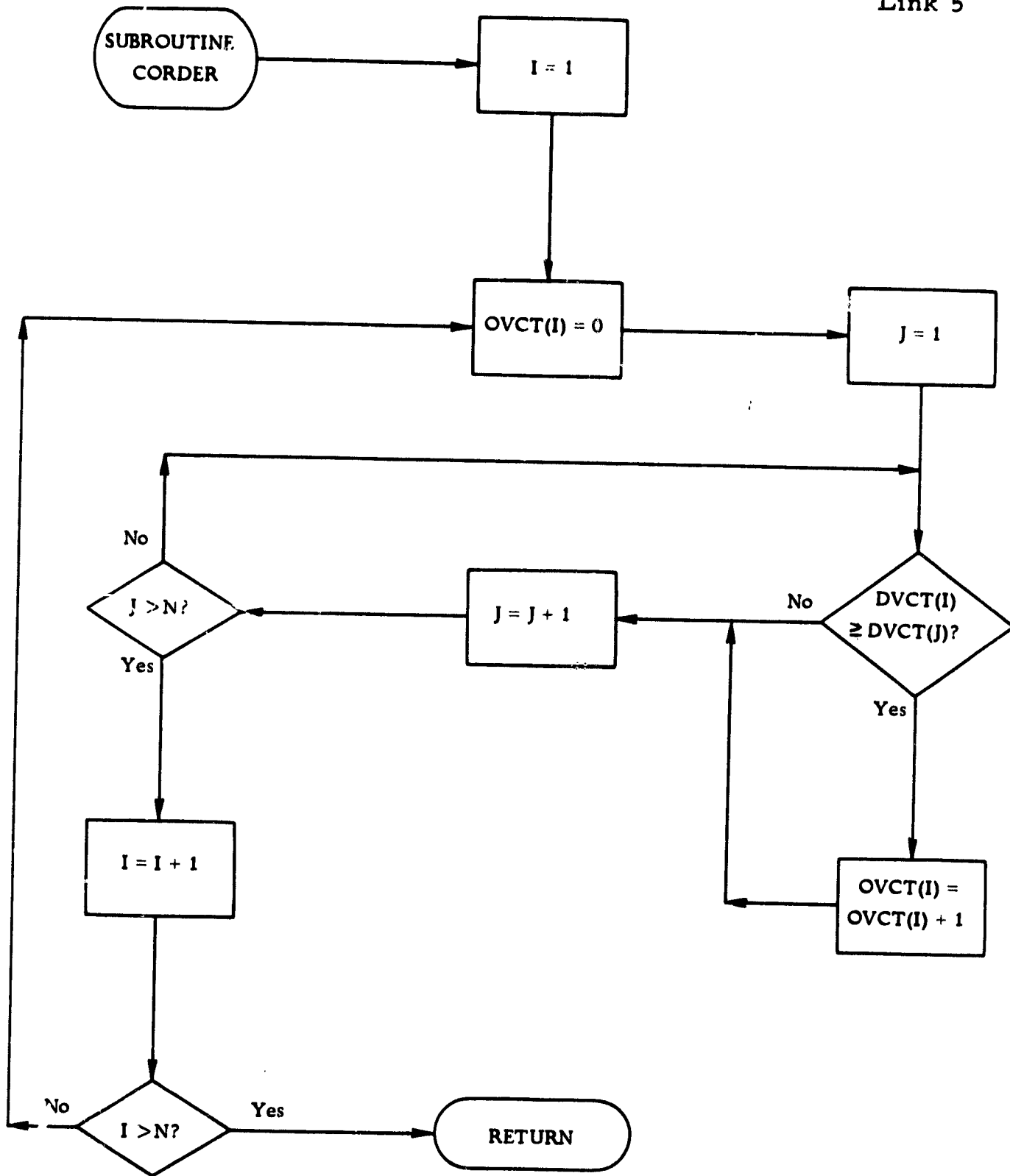
N: Number of colors (indices of refraction).

DVCT: Vector containing the N indices of refraction.

OVCT: Integer vector containing the relative positions of the indices in DVCT.

Utility Routines and Common References

None



3.5.3 CYCLEZ (CYCLE)

CYCLEZ uses the current set of design variables, the parameter vector \vec{U}_0 , and $\vec{E}_0 = \vec{E}(\vec{U}_0)$ to attempt to compute \vec{U}^* such that $|\vec{E}^*| < |\vec{E}_0|$. During this process failures may occur in the form of a singularity in the solution matrix, ray failure, boundary violations which are unrecoverable, and negative improvement.

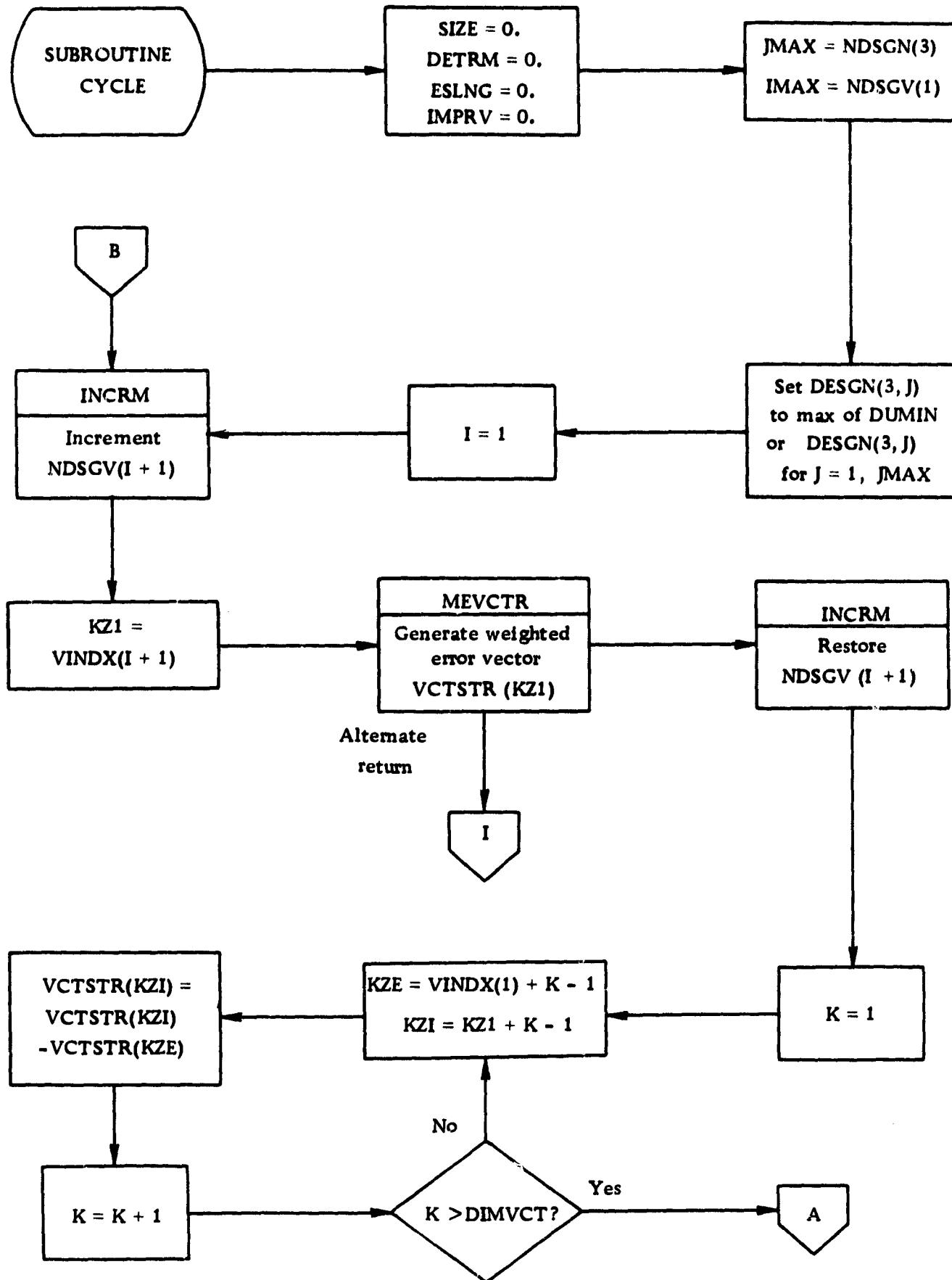
In the case of too many successive failures CYCLEZ executes the alternate return and the design is abandoned. When an improvement has been made the normal return is used. If a negative improvement has been made the parameters are restored so that the design may proceed on the next step.

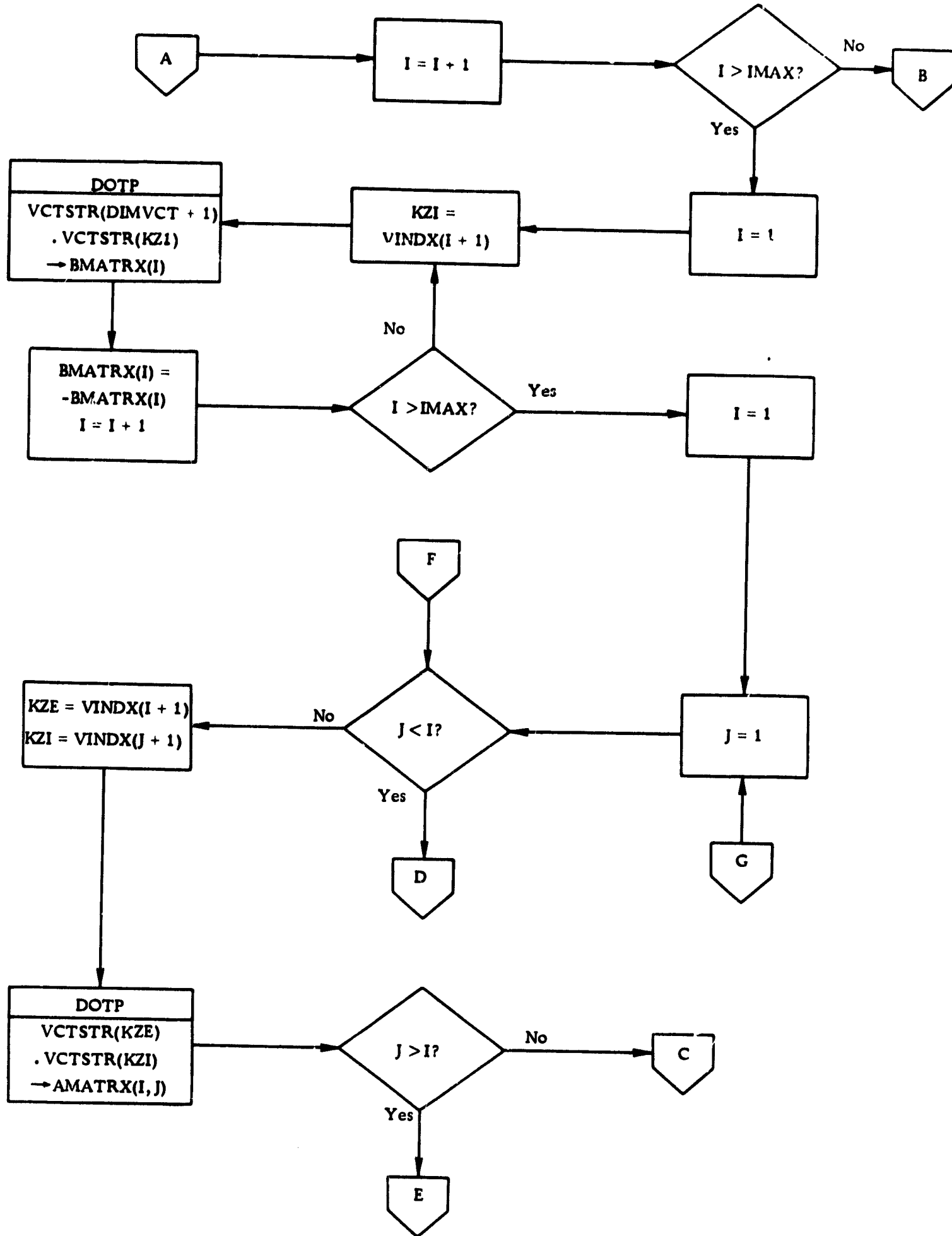
Calling Sequence

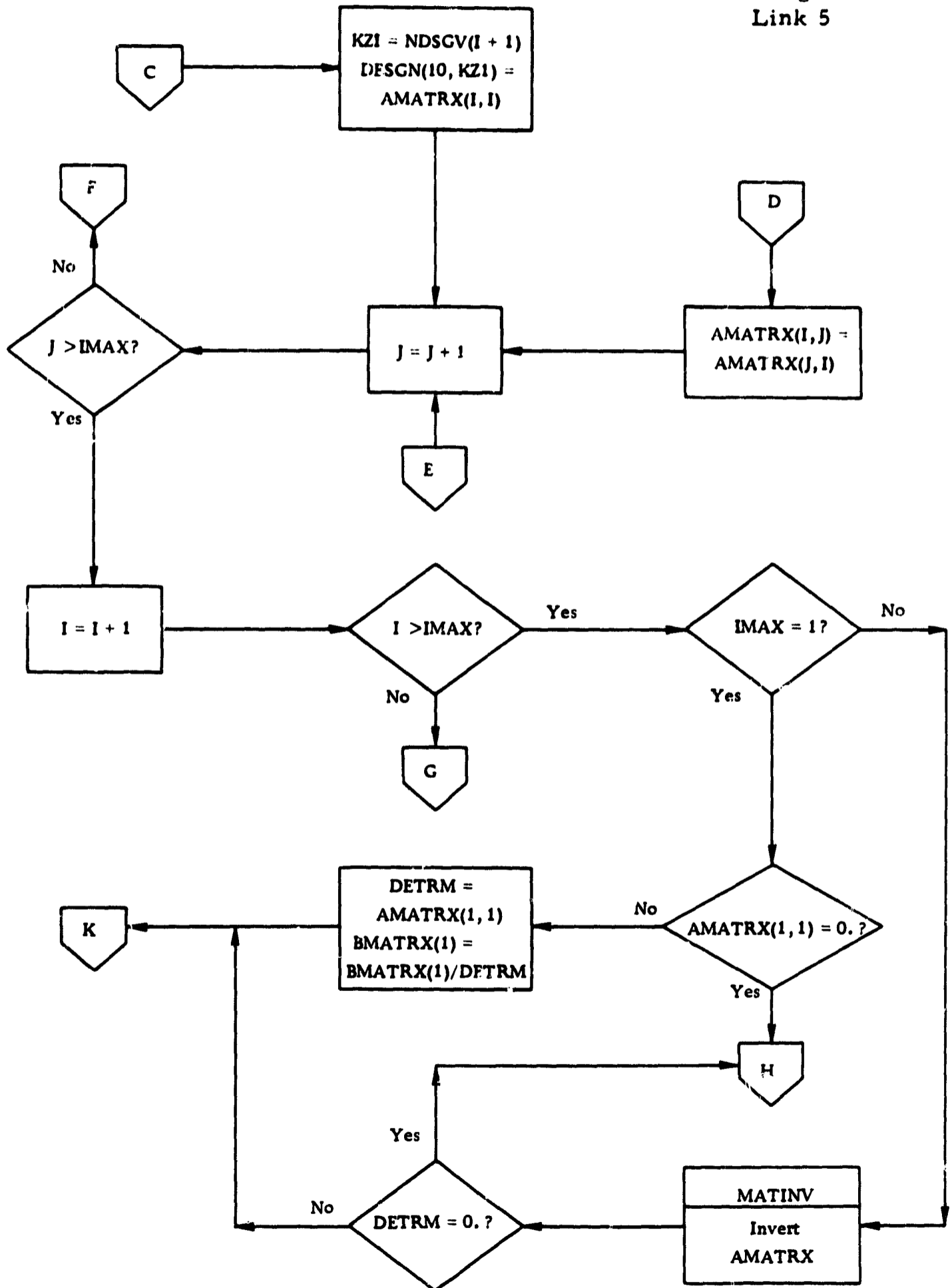
CALL CYCLE (*)
* Alternate return executed when too many
 successive failures have occurred.

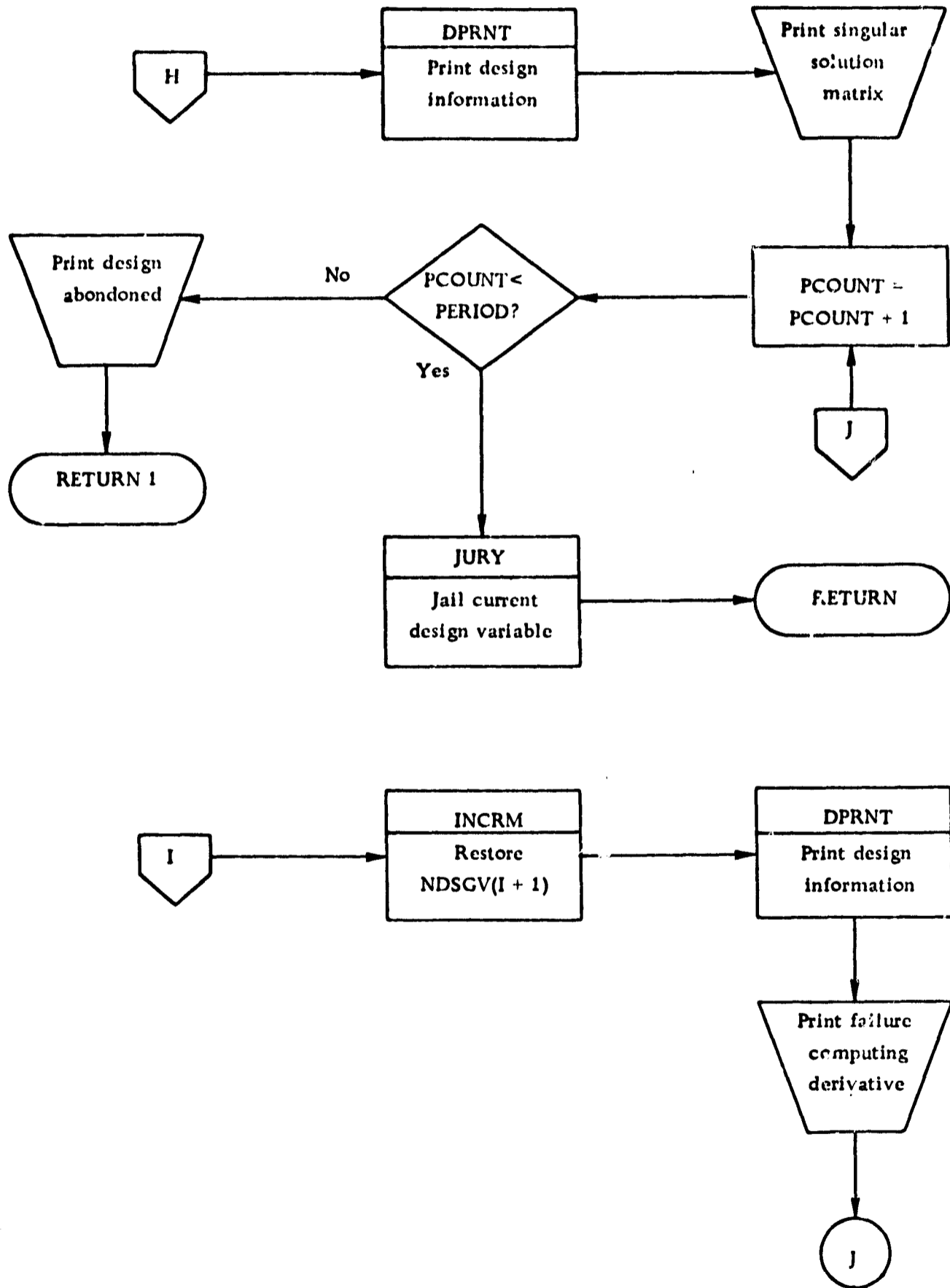
Utility Routines and Common References

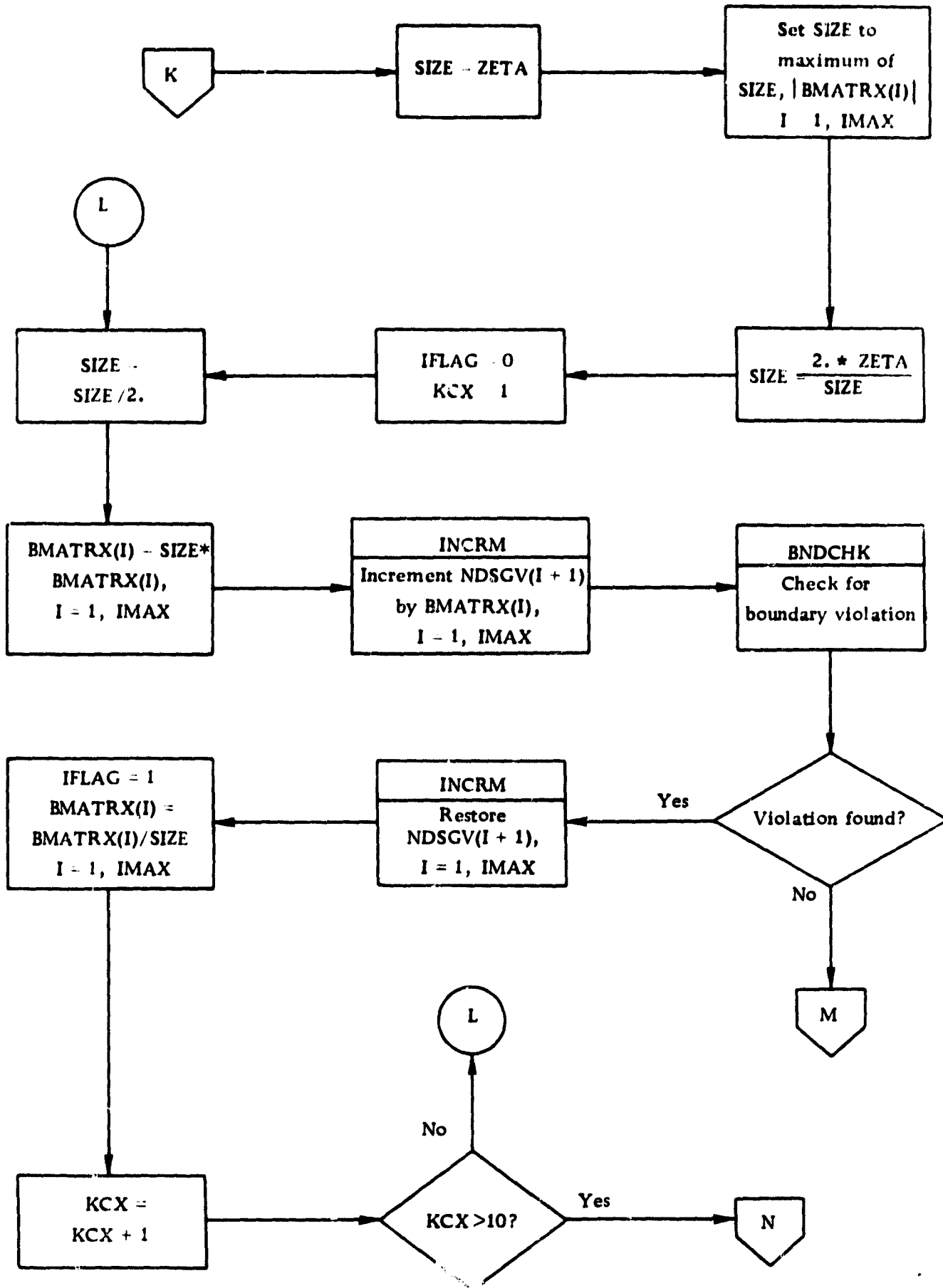
COMMON/DATA/
COMMON/PRNT/
COMMON/TMDESN/
BNDCHK
DOTP
DPRNT (EPRNT)
GETRHO
GOOF
INCRM
JURY
MATINV
MEVCTR
MODDL
PRMSUB



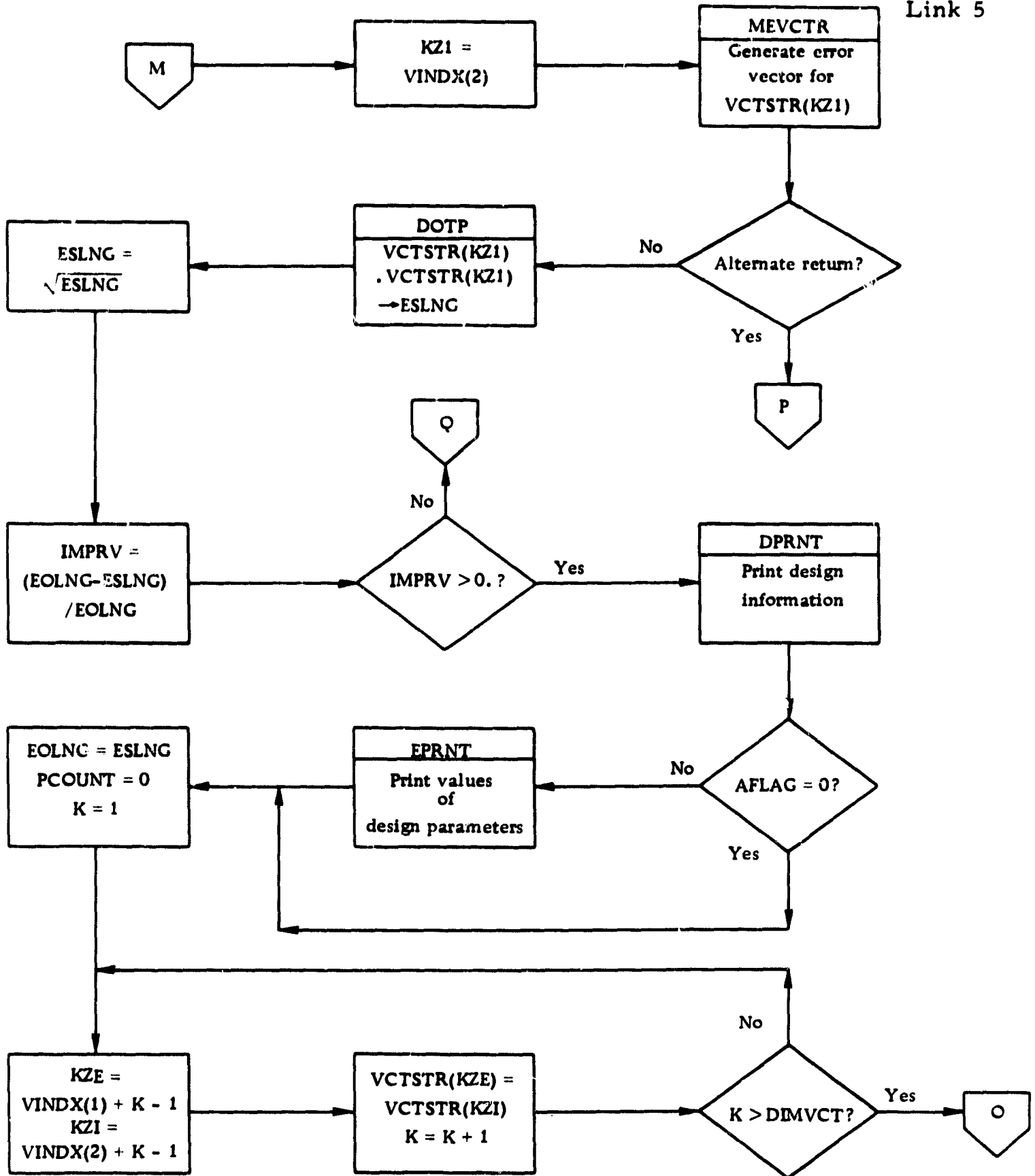


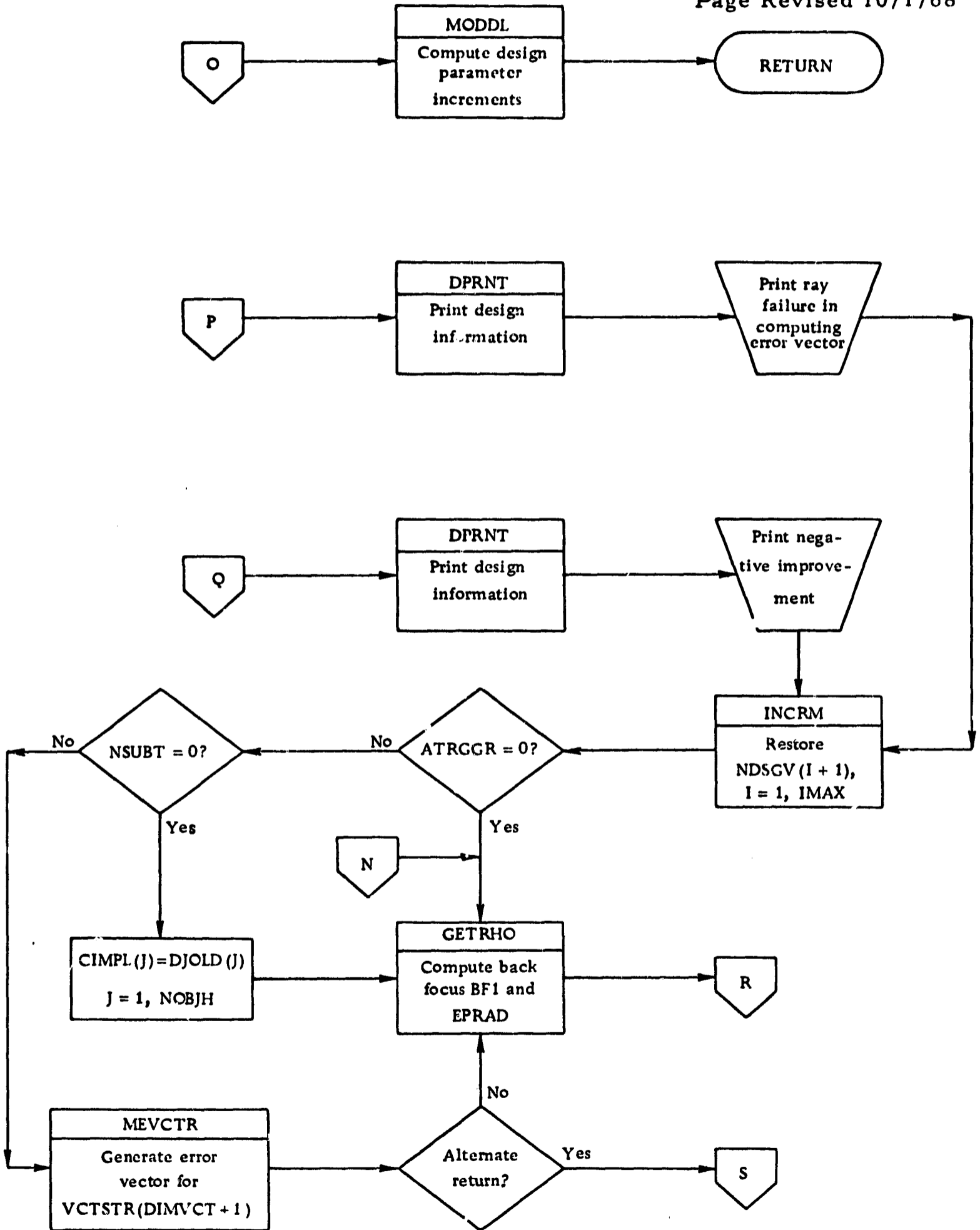


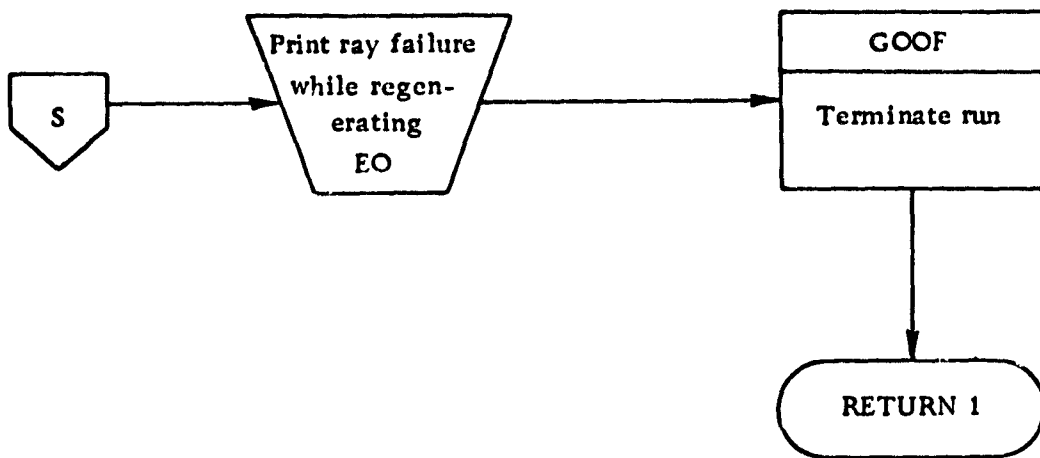
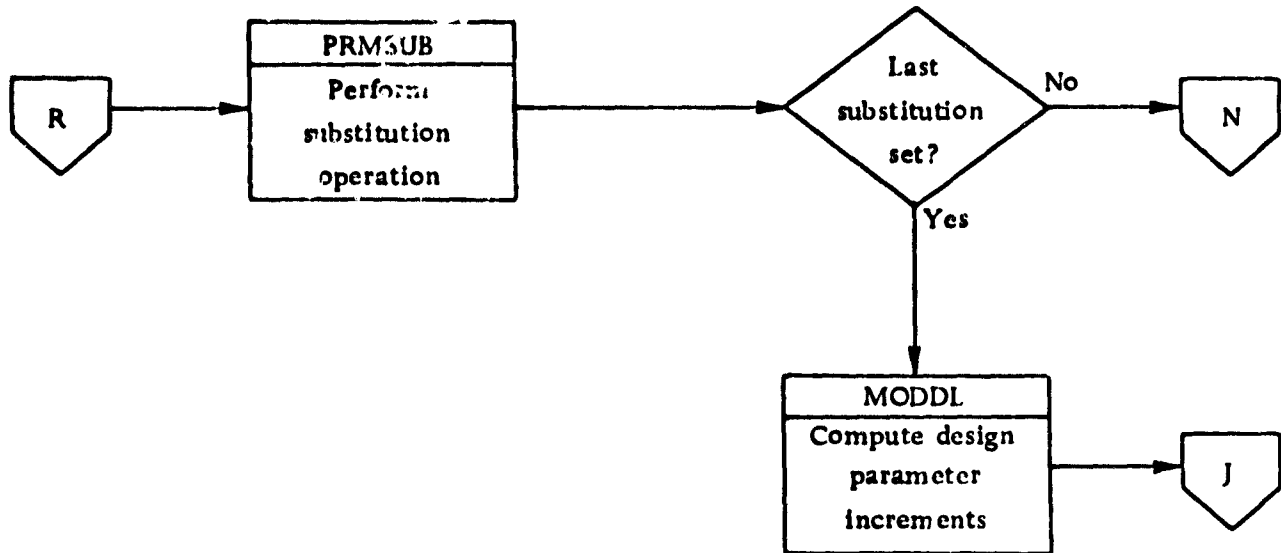




Link 5







3.5.4 DOTPZ (DOTP)

DOTPZ is used to calculate the inner product of \vec{V}_1 and \vec{V}_2 where \vec{V}_1 and \vec{V}_2 are vectors of dimension M. The result is stored in argument PROD.

Calling Sequence

CALL DOTP (VCT1, VCT2, PROD)

VCT1: The first of M cells containing the M-dimensional vector \vec{V}_1 .

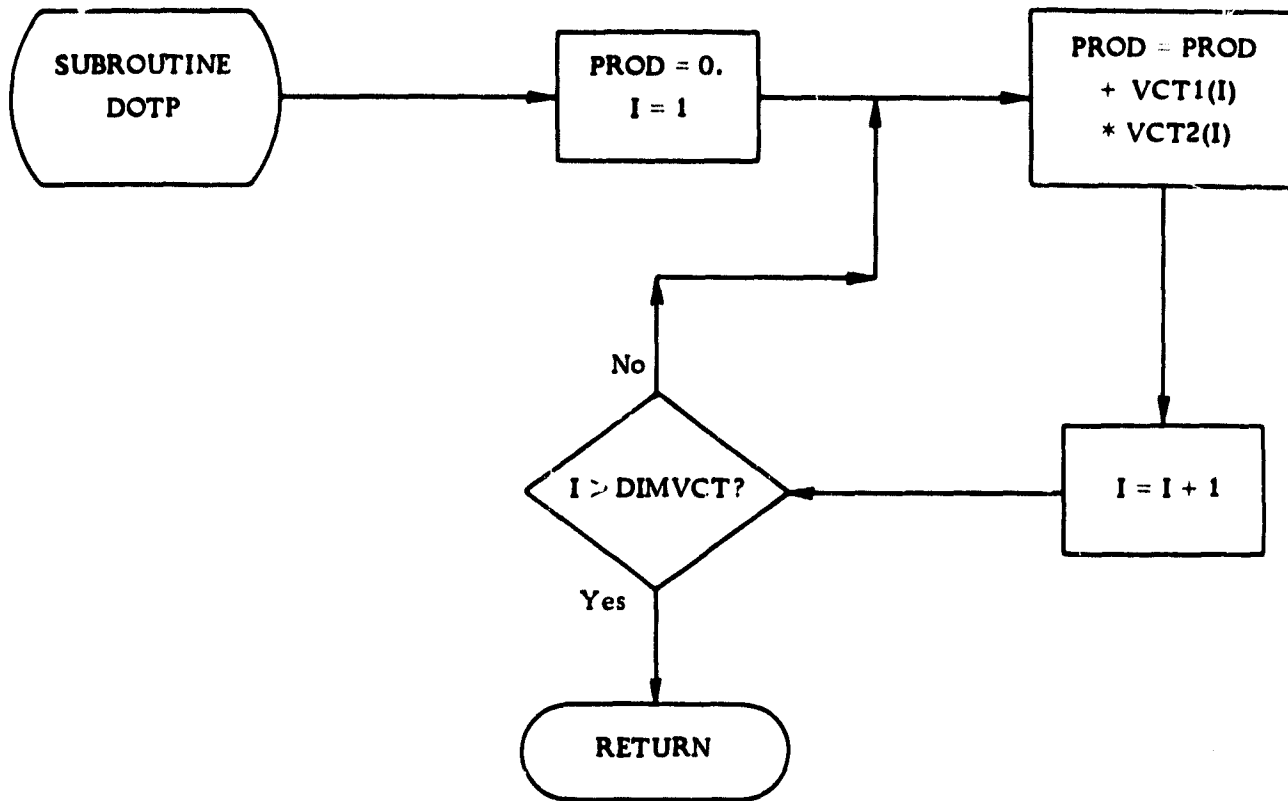
VCT2: The first of M cells containing the M-dimensional vector \vec{V}_2 .

PROD: Result of dot product $\vec{V}_1 \cdot \vec{V}_2$.

DIMVCT: M, dimension of vectors. (TMDESN common).

Utility Routines and Common References

COMMON/TMDESN/



3.5.5 DPRNTZ (DPRNT)

DPRNTZ consists of two entry points, DPRNT and EPRNT, and is used to print out the results of each design step.

In DPRNT the design variables, incremental factor, determinant value, old and new error vector lengths and improvement are printed on a single line.

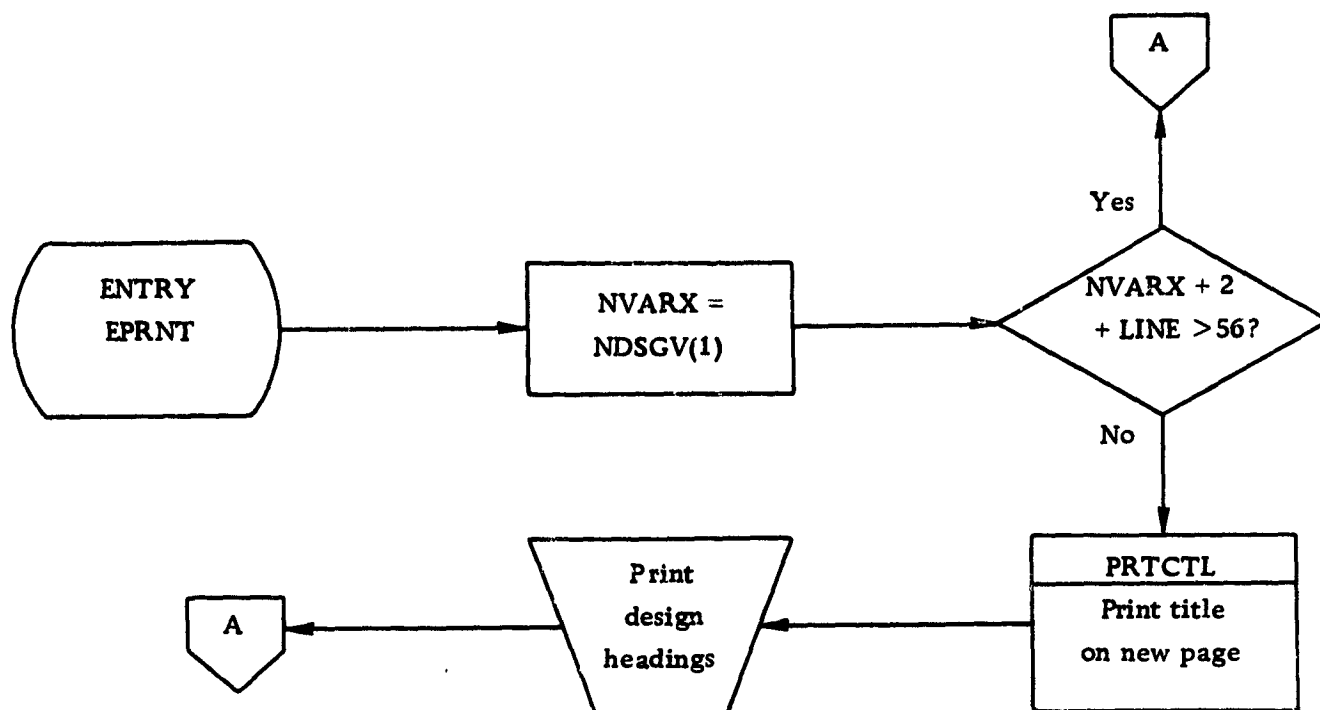
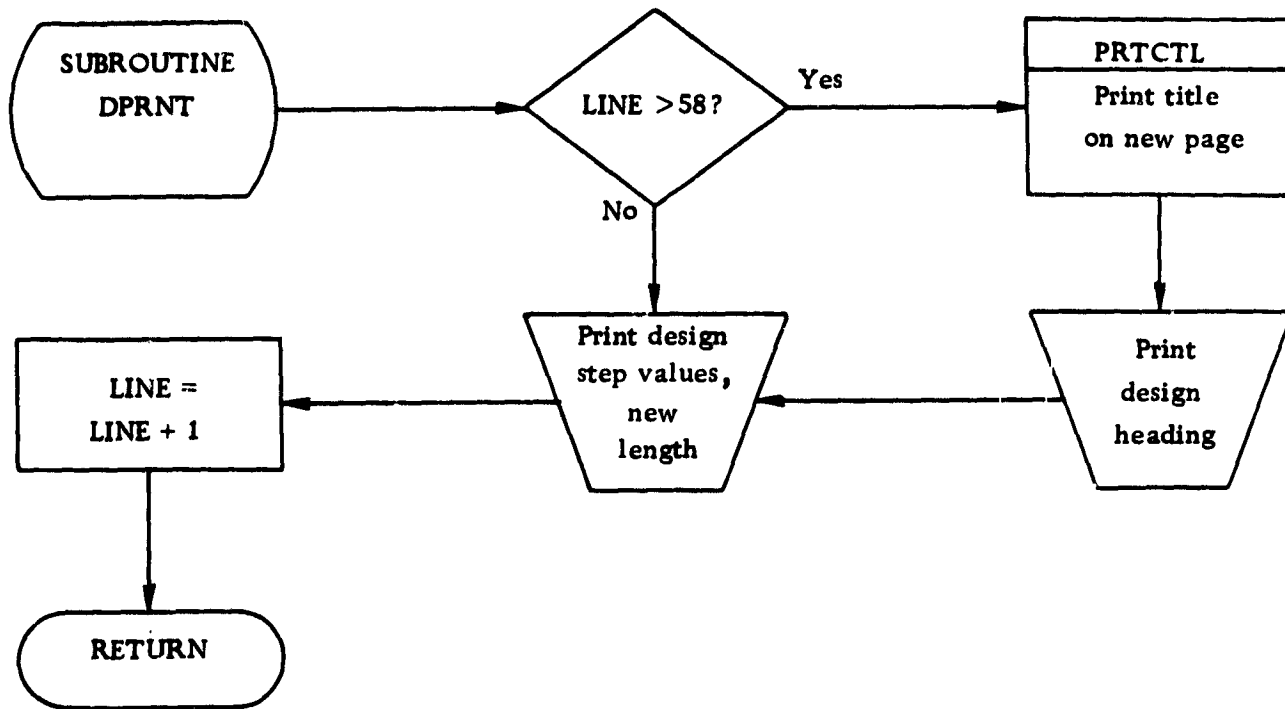
In EPRNT the current values of all the design parameters for the current design step are printed. Up to six parameters for each design variable may be printed per line with one line for each design variable.

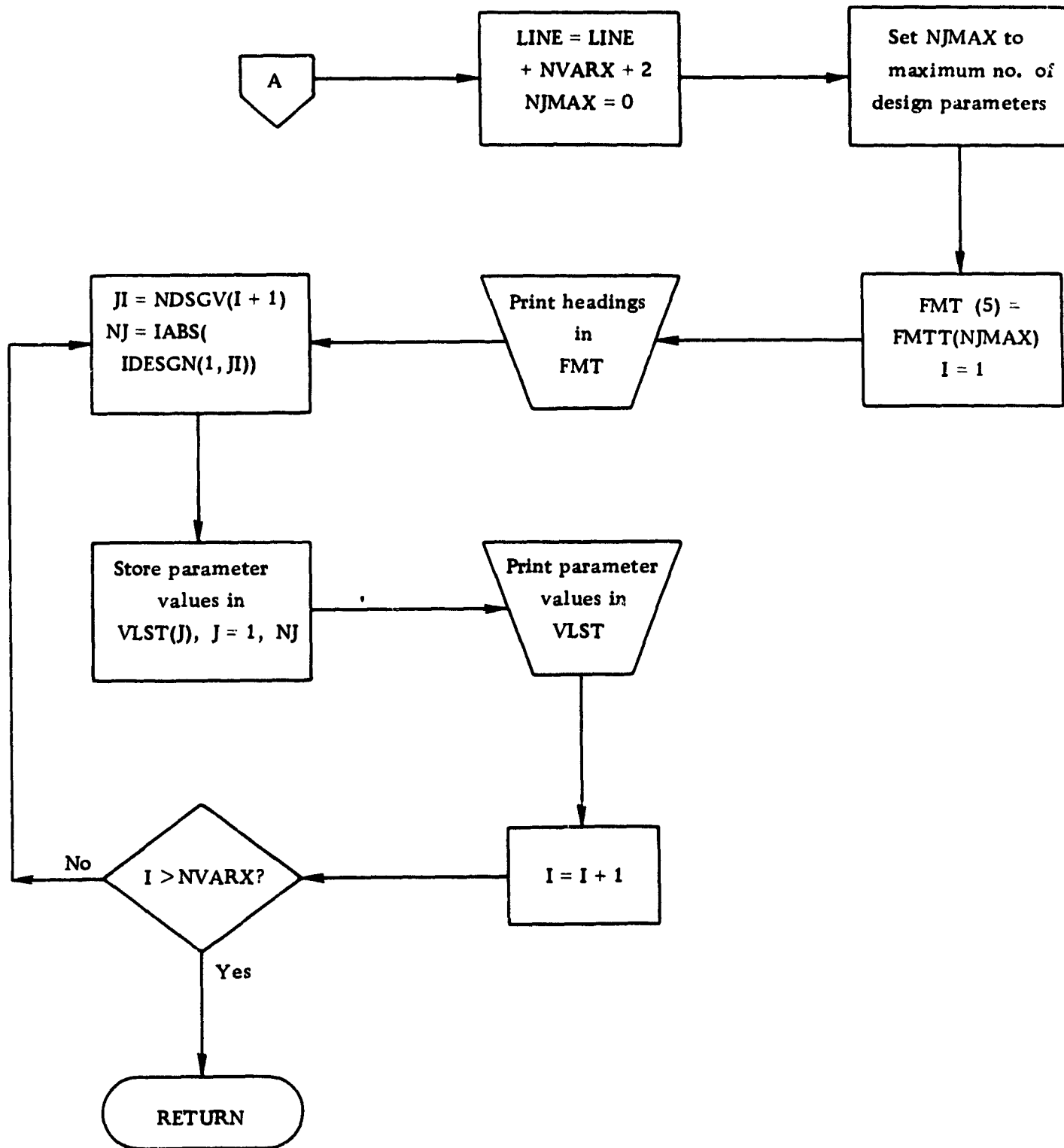
Calling Sequence

CALL DPRNT
CALL EPRNT

Utility Routines and Common References

COMMON/DATA
COMMON/PRNT/
COMMON/TMDESN/
PRTCTL





3.5.6 GETRHZ (GETRHO)

GETRHZ performs two functions. The first is to compute the back focus for color 1 using the initial value of ρ_o (EPRAD). The back focus is stored in BF1.

If the f number, $f(R)/$, in FNUMB is not zero then the GETRHZ replaces ρ_o with $\rho_o^{(3)}$ which is computed as follows. Let $\rho_o^{(0)}$ be the initial value of ρ_o and $f^{(1)}/$ be the corresponding f/number. Then:

$$\rho_o^{(i)} = \left| \rho_o^{(i-1)} \frac{f^{(i)}/}{f^{(R)}/} \right| \quad \text{for } i = 1, 2, 3$$

where $f^{(i)}/$ is the f/number with $\rho_o = \rho_o^{(i-1)}$.

Subroutine MERID is utilized in calculating the f/number.

Calling Sequence

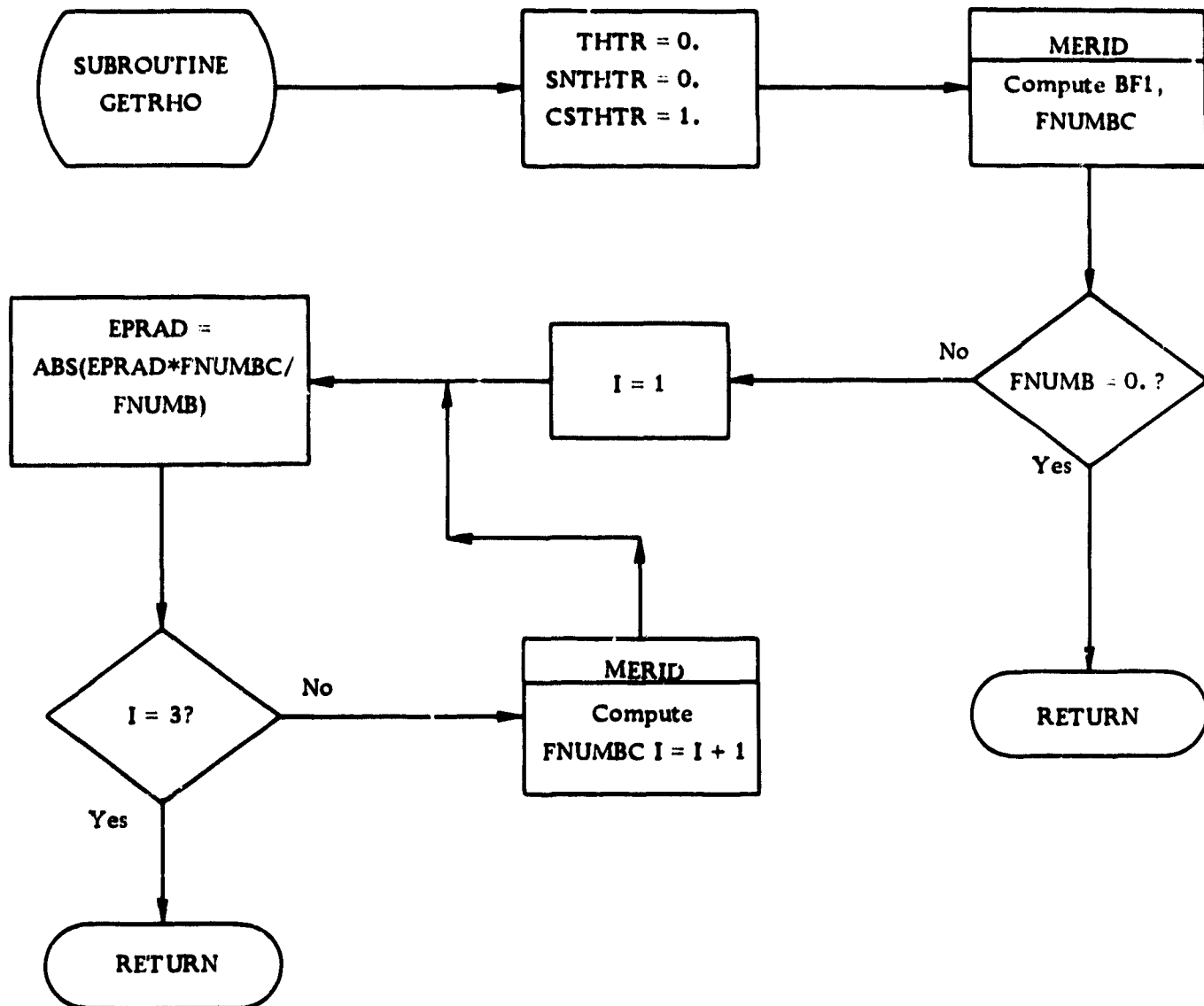
CALL GETRHO (BF1)

BF1: Computed back focus for color 1.

EPRAD: Recomputed value for the radius of the entrance pupil provided FNUMB is non-zero.

Utility Routines and Common References

COMMON/DATA/
COMMON/AZOBJ/
MERID



3.5.7 INCRMZ (iNCRM)

INCRMZ is used to increment each of the parameters associated with the design variable defined by the first argument of the calling sequence. This argument, J, points to the Jth column of the design matrix, \tilde{V}_j , as shown below:

$$\tilde{V}_j = \begin{pmatrix} N_j \\ \Delta_j^{(R)} \\ \Delta V_j \\ K_{1j} \\ K_{2j} \\ \cdot \\ \cdot \\ \cdot \\ KN_{jj} \end{pmatrix}$$

The second argument FACTOR, contains a multiplier λ which is applied to ΔV_j before incrementation. Thus each parameter in \tilde{V}_j is incremented by $\lambda \Delta V_j$. Depending on the associated sign of a parameter the increment is added or subtracted. Also if the parameters are radii of curvature then the increment is applied to the reciprocal.

Calling Sequence

CALL INCRM (J, FACTOR)

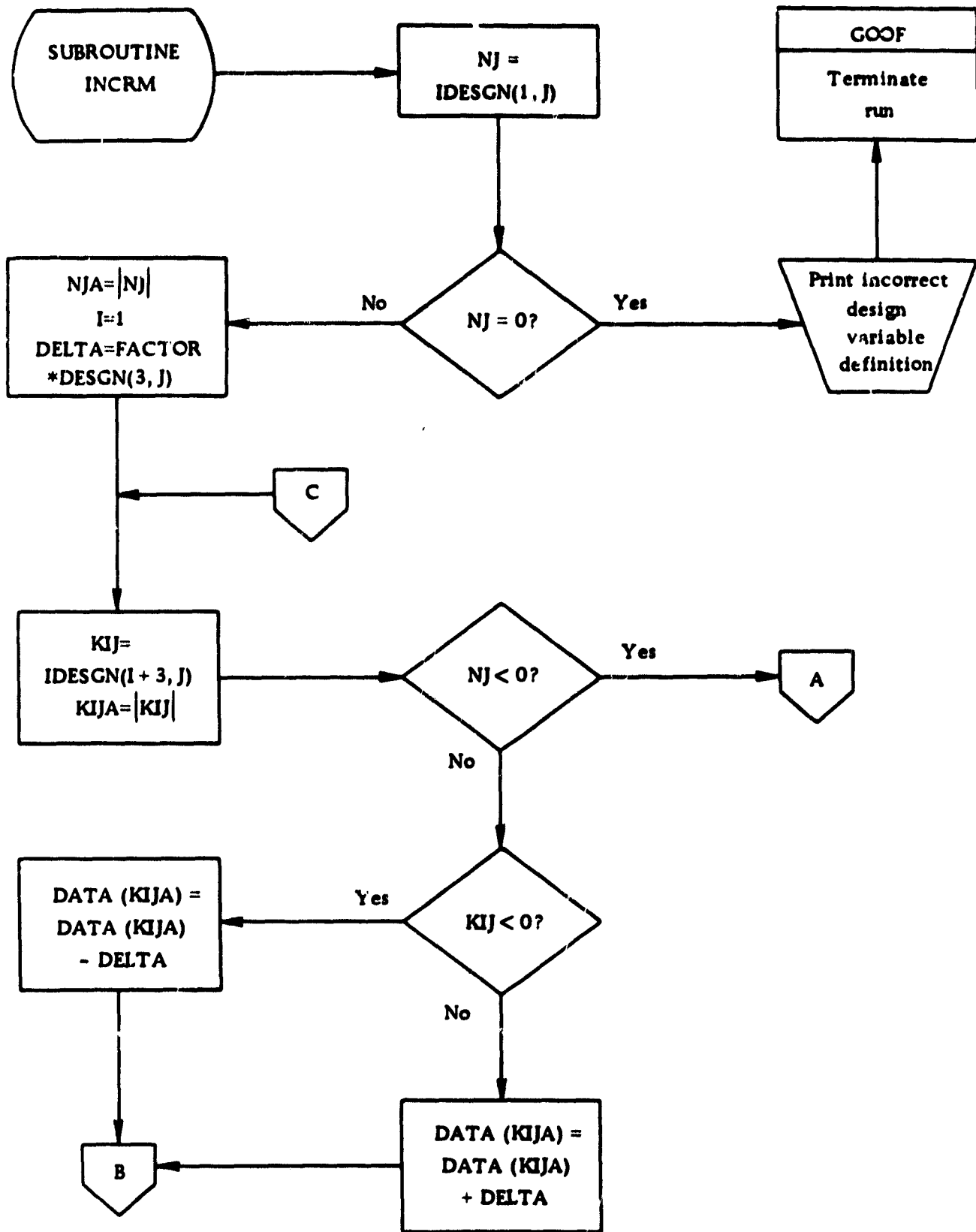
J: Design variable number specifying the design variable to be incremented.

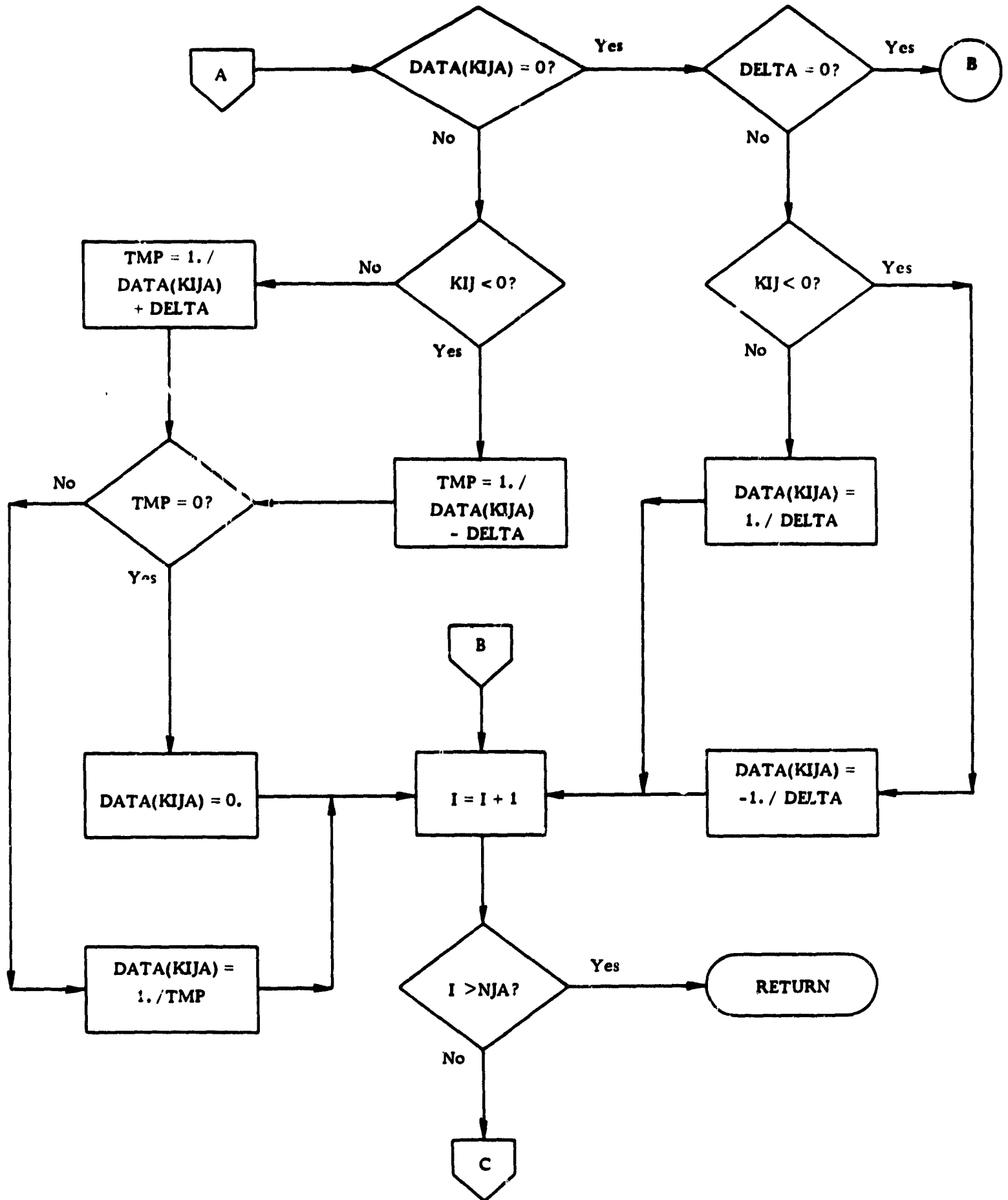
FACTOR: The multiplier λ which is applied to ΔV_j before incrementation.

Utility Routines and Common References

COMMON/DATA/

GOOF





3.5.8 JURYZ (JURY)

JURYZ consists of two entry points, JURY and JUDGE. It is used to handle the logic to restrict a combination of design variables which has resulted in failure while designing from appearing in subsequent design steps.

JURY is entered when a combination of design variables has failed. If there is sufficient storage the combination is jailed (restricted from appearing in a design step until its sentence has been served).

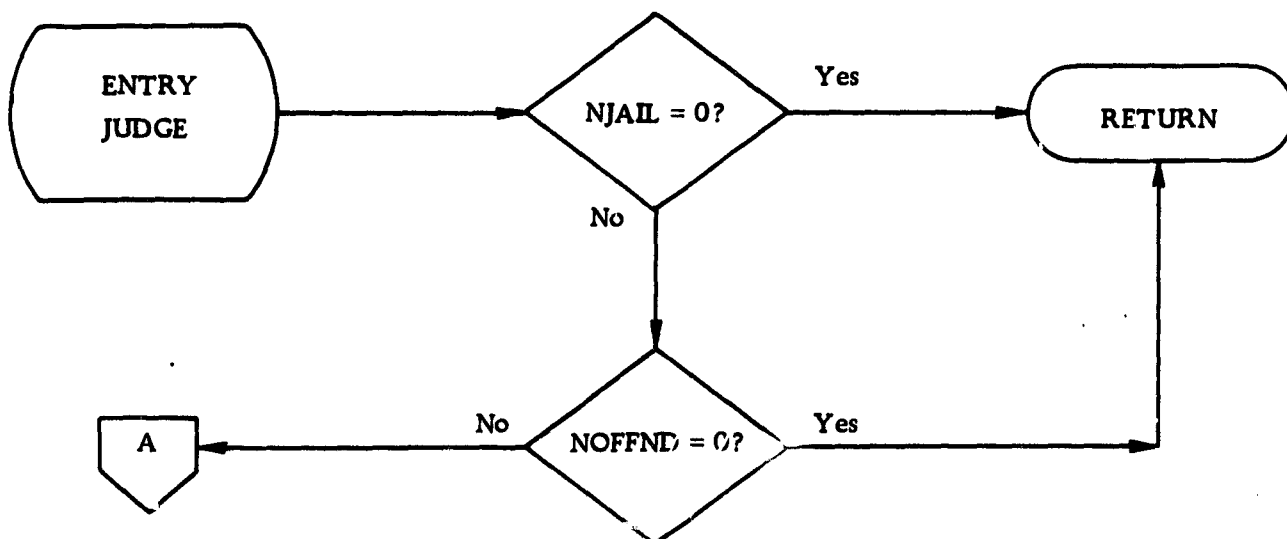
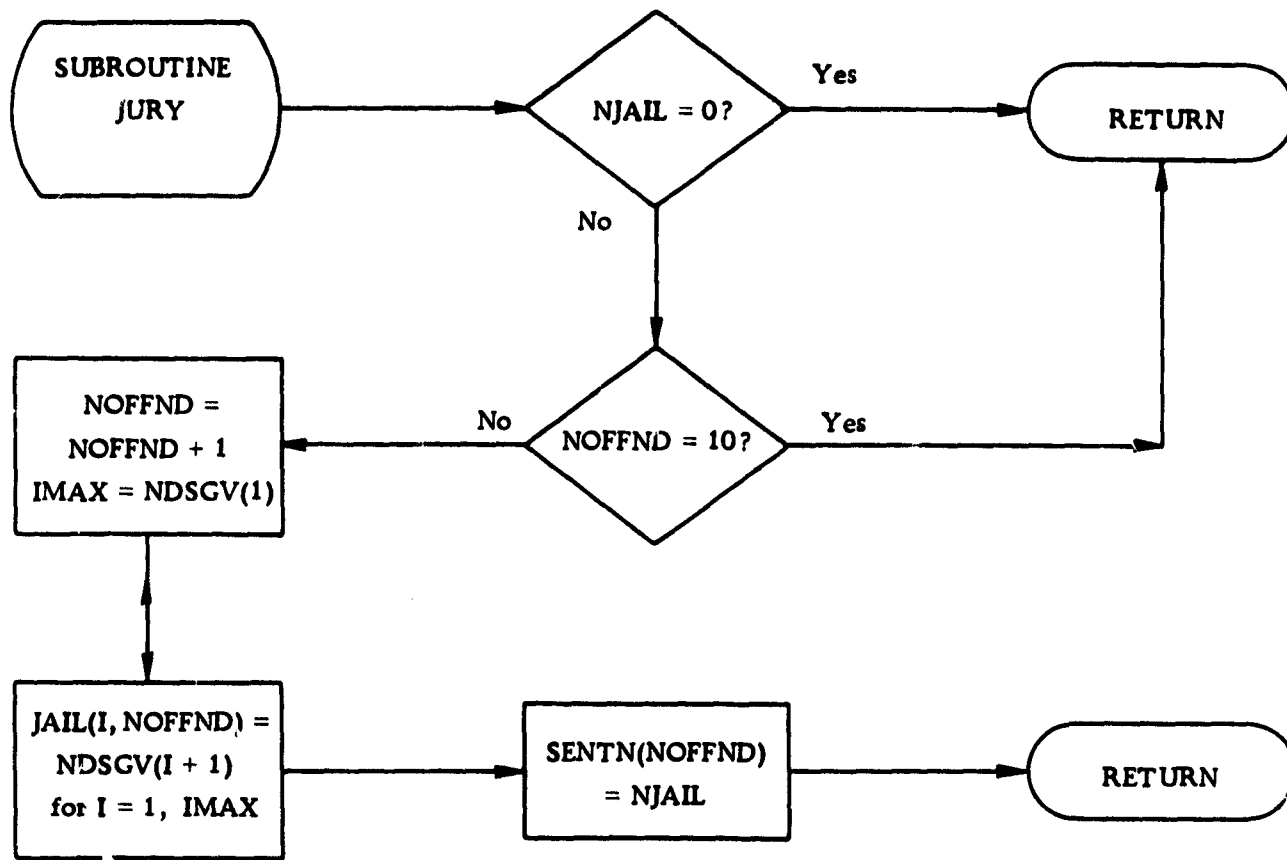
JUDGE is used to test the current design variable combination to determine if it is in jail. If the design variable combination is not in jail, JUDGE makes a normal return. Otherwise the sentence of the combination is reduced by one. If the sentence is now zero the design variable is released from jail and a normal return is made. JUDGE uses the alternate return if the sentence is greater than zero.

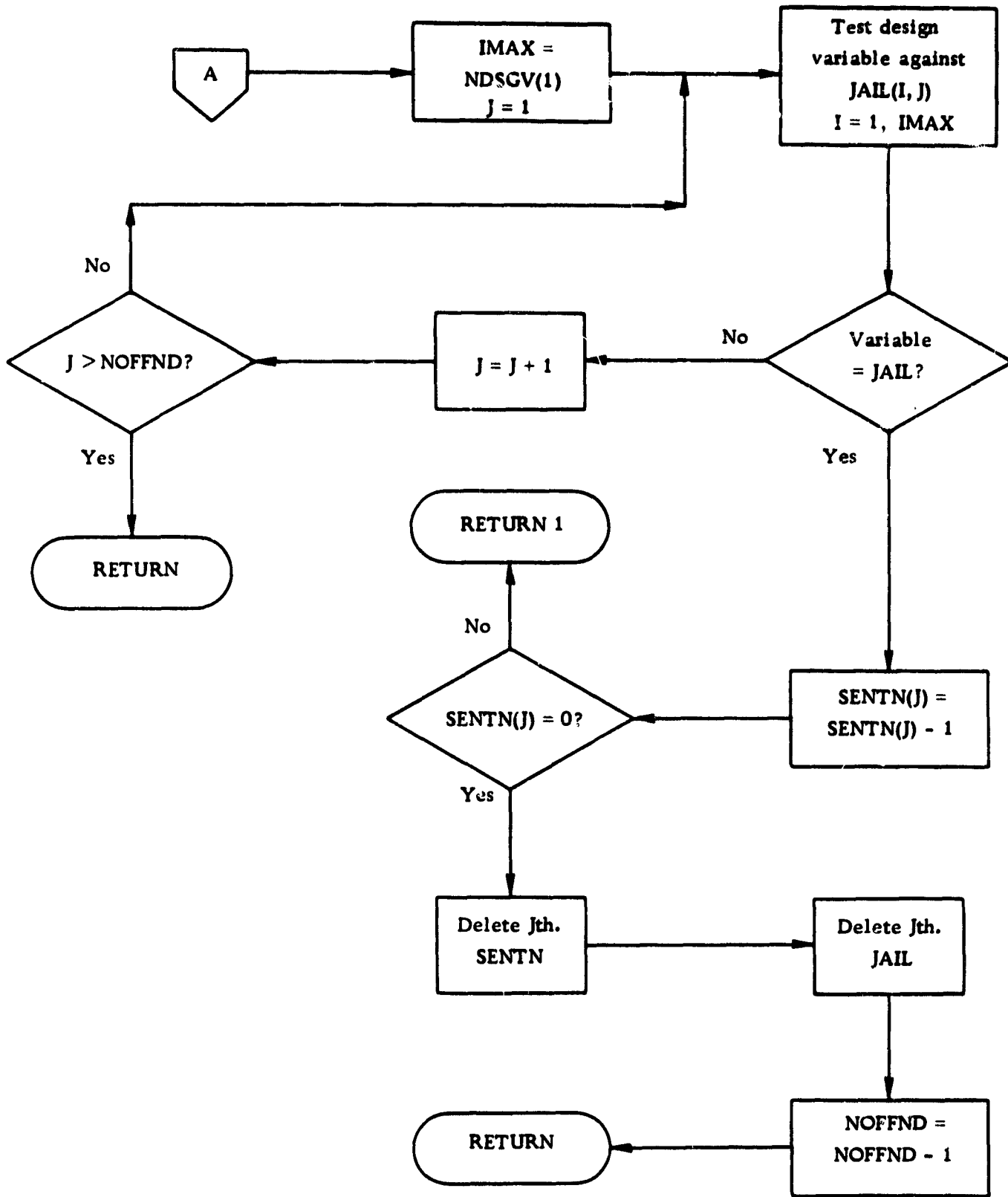
Calling Sequence

CALL	JURY
CALL	JUDGE (*)
*	Is the alternate return taken if the sentence of a jailed design variable combination is greater than zero.

Utility Routines and Common References

COMMON/DATA/
COMMON/TMDESN/





3.5.9 MEVCTZ (MEVCTR)

MEVCTZ is used to compute the M-component weighted error vector \vec{E} . The vector is stored as follows:

$$\begin{aligned} \text{EVCTOR}(1) &= e_1 \\ \text{EVCTOR}(2) &= e_2 \\ &\cdot \\ &\cdot \\ &\cdot \\ \text{EVCTOR}(M) &= e_M \end{aligned}$$

The weight vector, \vec{W} , which was generated by MWVCTR, is utilized in the computations. If any of the rays required in the generation of \vec{E} miss or reflect, then MEVCTZ invokes the alternate return (*). If the value of M does not agree with DIMVCT the run will be terminated by calling GOOF.

Calling Sequence

CALL MEVCTR (EVCTOR, *, KFLAG1, KFLAG3)

EVCTOR: Vector containing the M-component weighted error vector.

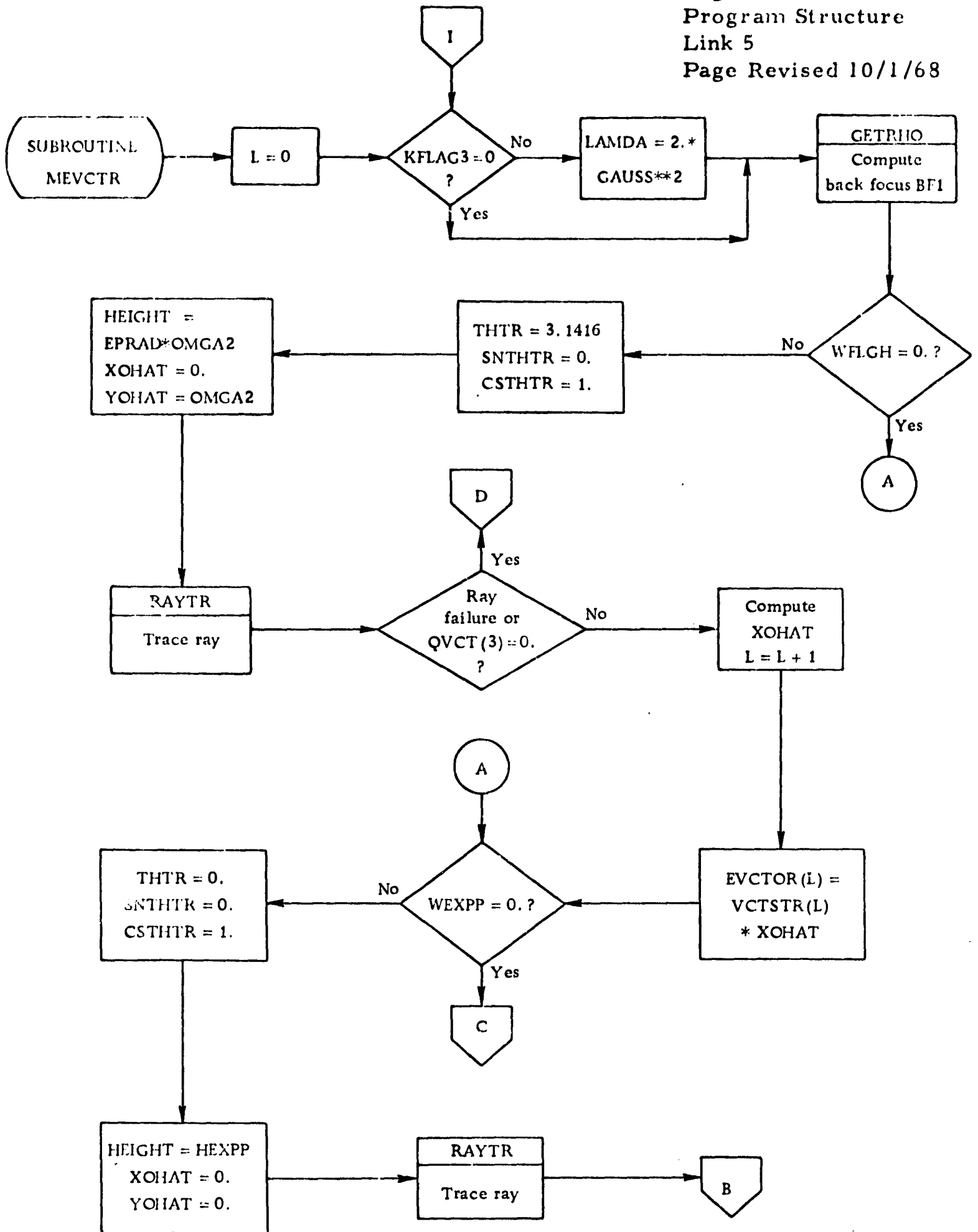
*: Alternate return utilized in case of ray failure.

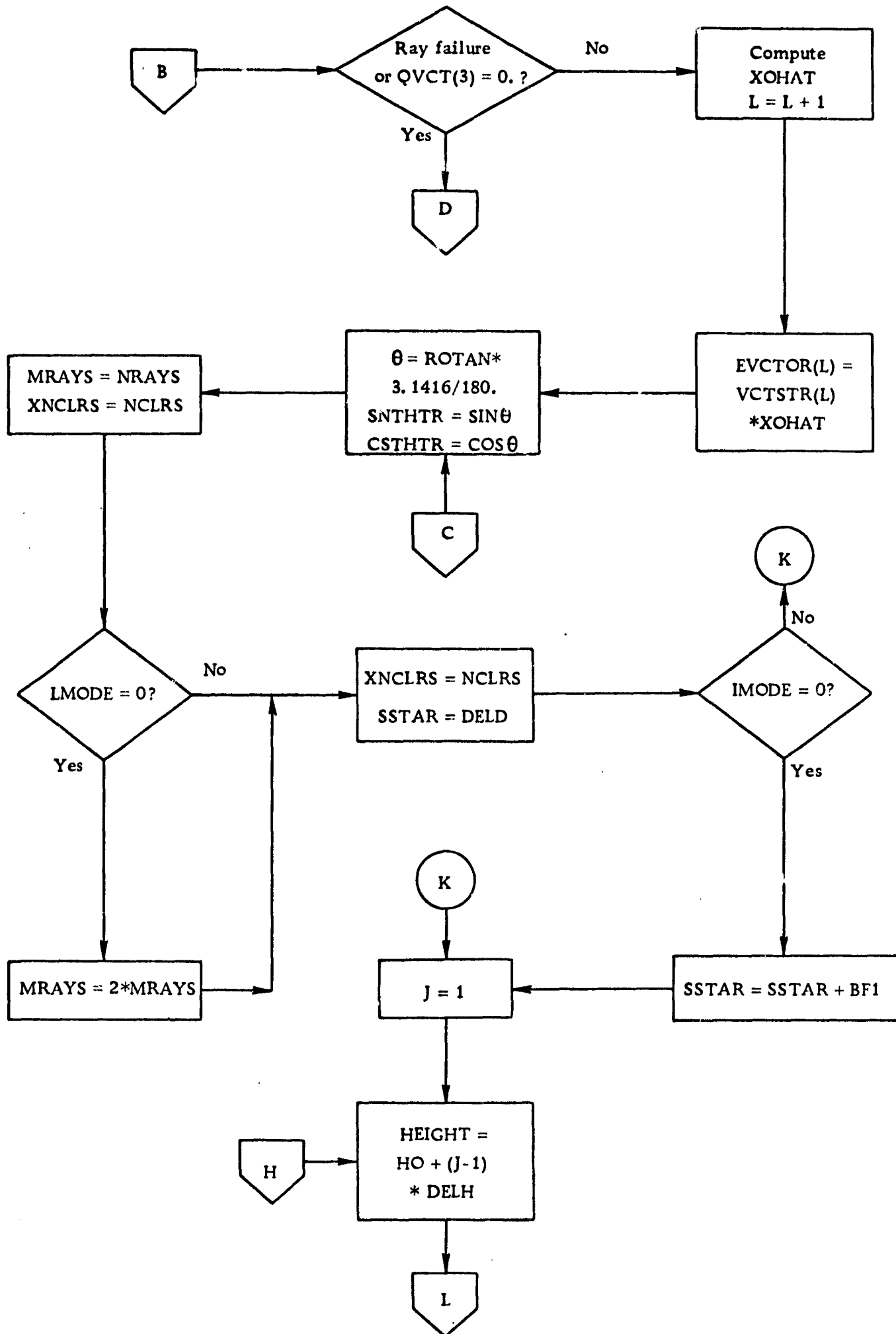
KFLAG1: Curved image surface option flag.

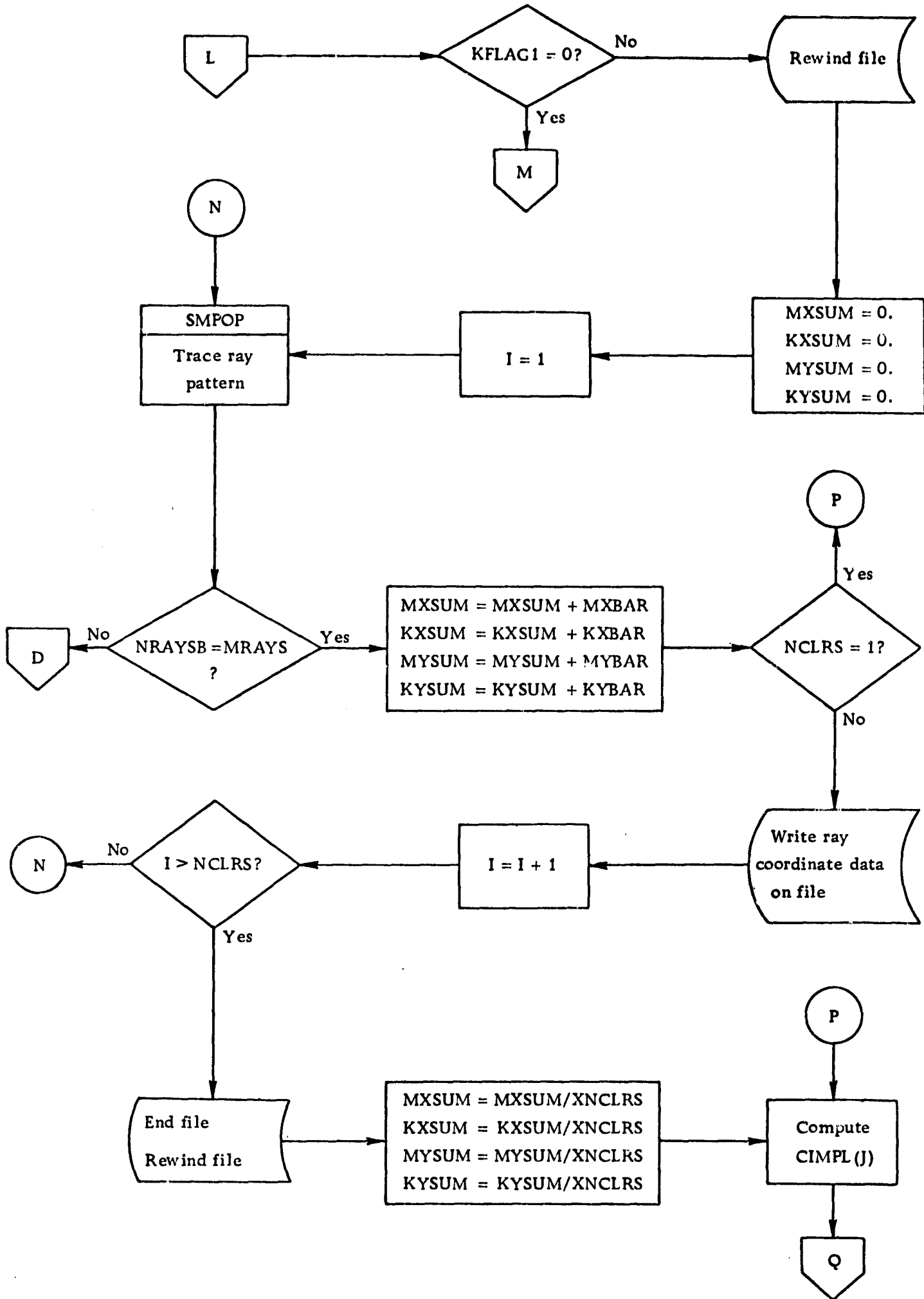
KFLAG3: Gaussian weighting option flag.

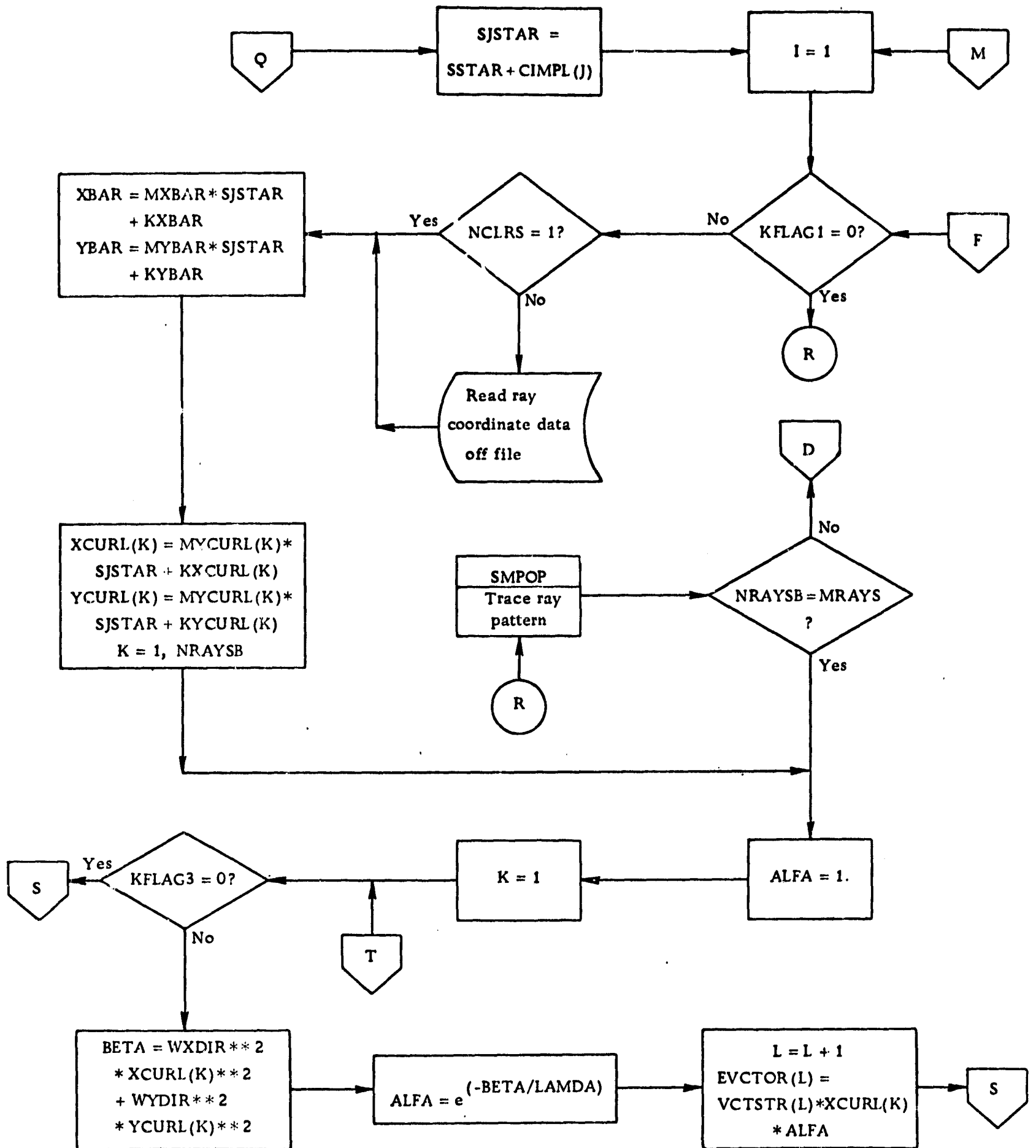
Utility Routines and Common References

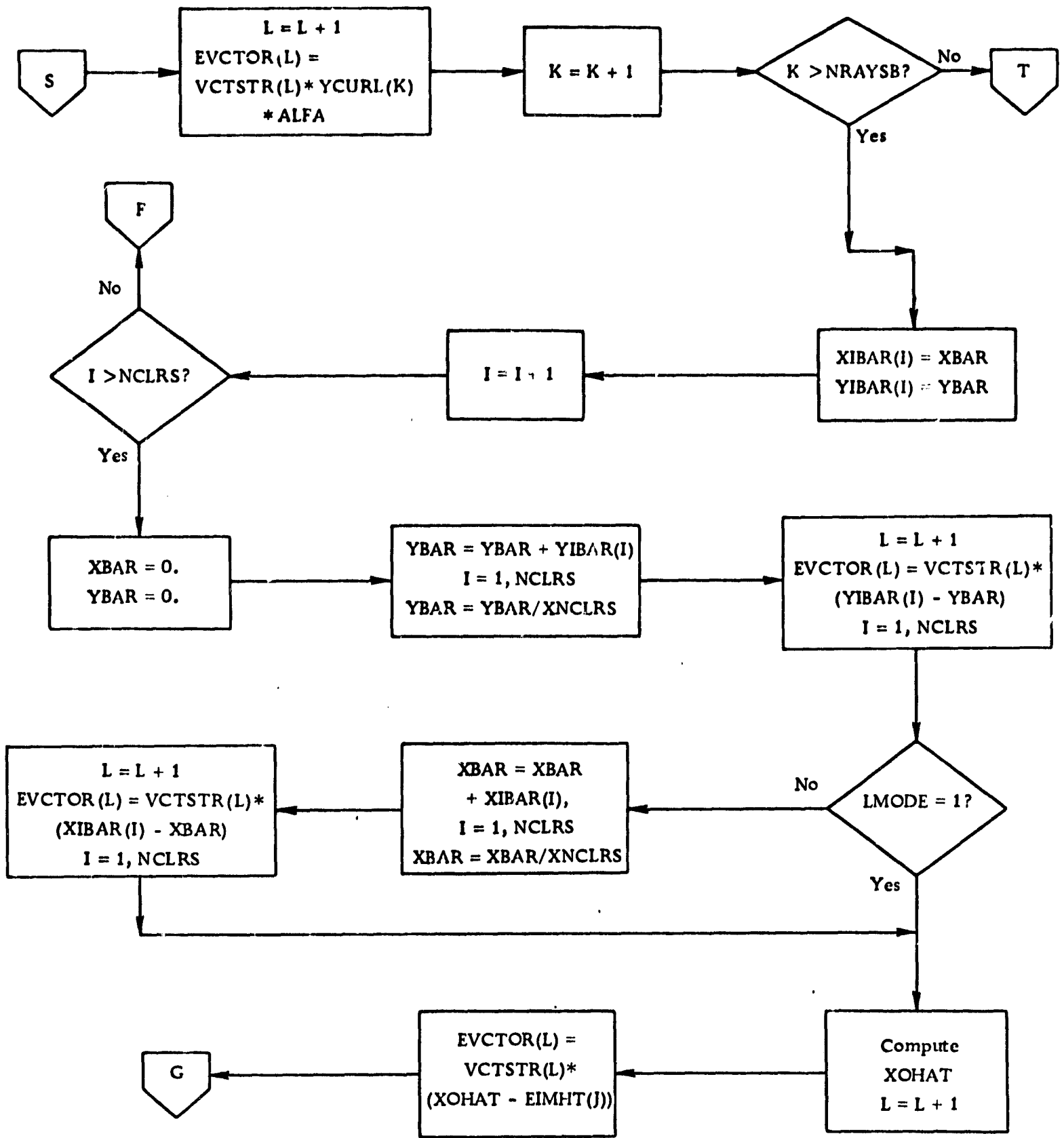
COMMON/DATA/
COMMON/AZOBJ/
COMMON/TMPATT/
SMPOP
RAYTR
PRMSUB
GETRHO

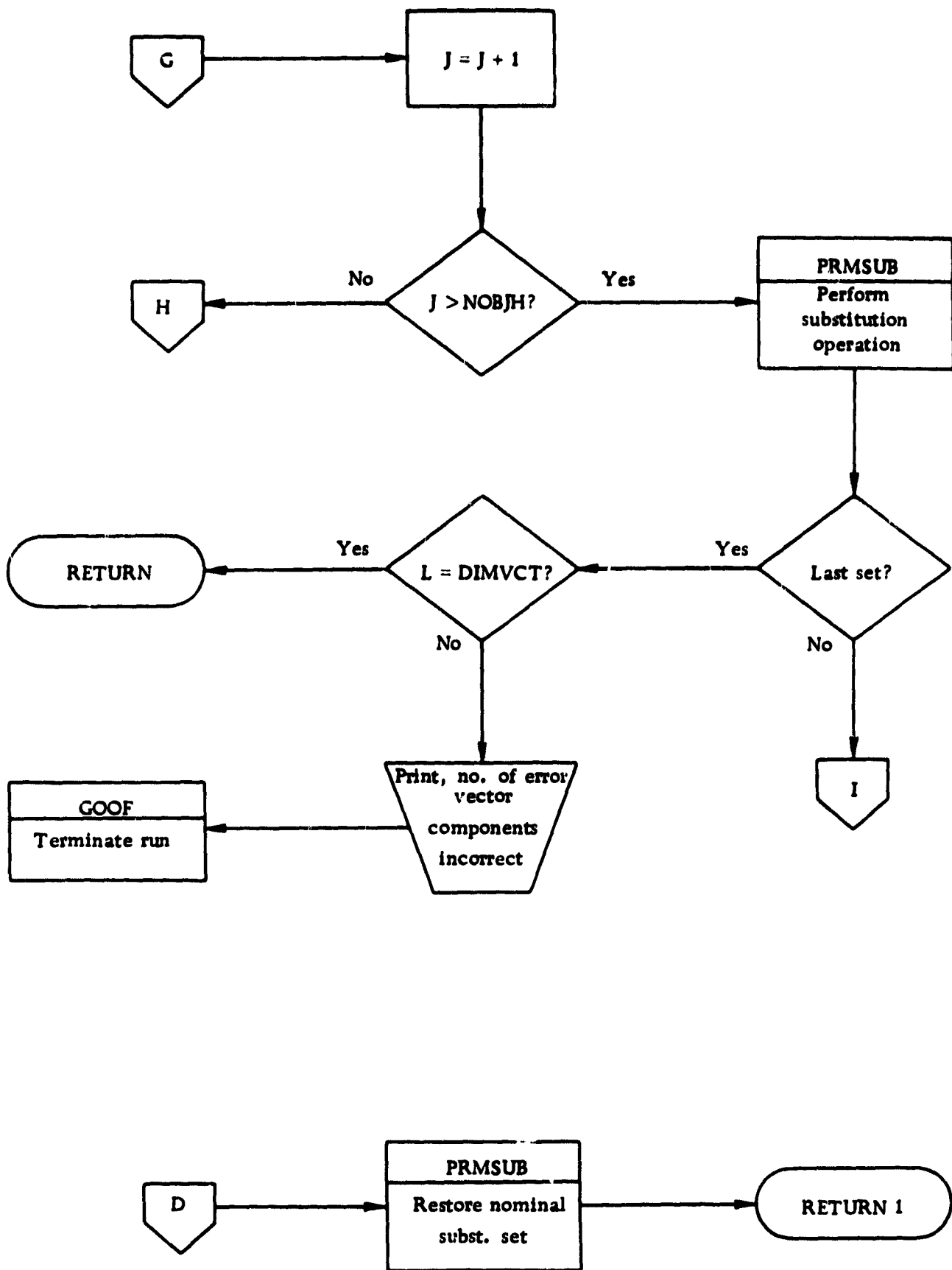












3.5.10 MODDLZ (MODDL)

MODDLZ is used to compute the increments which are applied to the design parameters of each design variable of a design step. For the set $J = \{j_1, j_2, \dots, j_n\}$, representing the design variables presnet, if $\Delta_{j_1}^{(C)}$, i. e. (DESIGN(10, I)), is 0, then ΔV_{j_i} (DESGN(3, I)) is unchanged. Otherwise:

$$\Delta V_{j_i} = \left| \Delta V_{j_i} \cdot \left(\frac{\Delta_{j_i}^{(R)}}{\Delta_{j_i}^{(C)}} \right)^{1/2} \right| \quad \text{where } \Delta_{j_i}^{(R)} \text{ is DESIGN(2, I)}$$

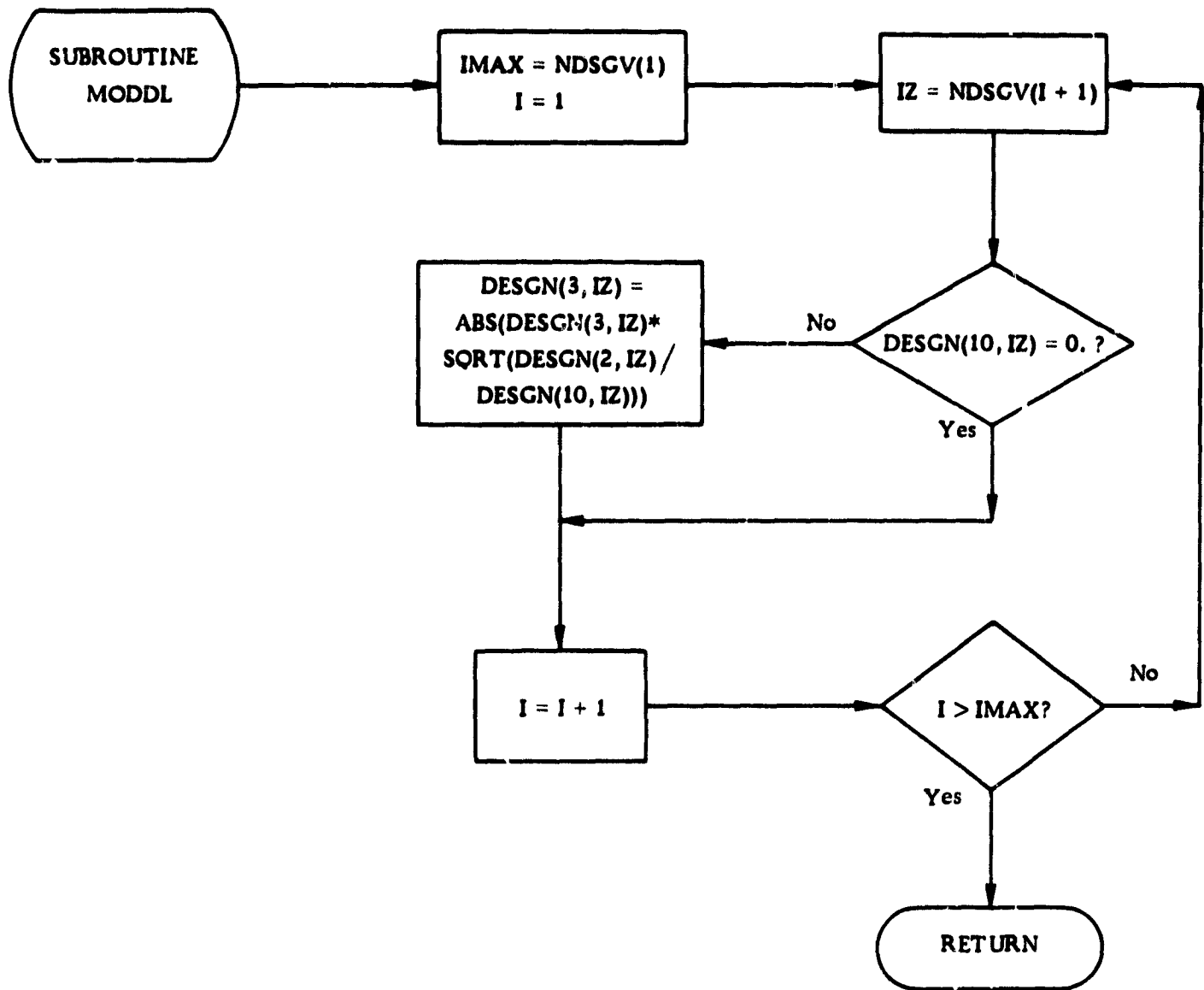
Here $1 \leq I \leq \text{NDSGV}(1)$, the number of design variables per design step.

Calling Sequence

CALL MODDL

Utility Routines and Common References

COMMON/DATA/



3.5.11 MWVCTZ (MWVCTR)

MWVCTR is used to generate the M-component weight vector \vec{W} which is stored in VCTSTR(1) through VCTSTR(M), where $M = \text{DIMVCT}$. All of the weights used to calculate VCTSTR are taken from DATA common. If the number of components in VCTSTR is unequal to DIMVCT the run is terminated by a call to GOOF.

Calling Sequence

CALL MWVCTR

WFLGH: Focal length deviation weight.

WEXPP: Exit pupil position weight.

WXDIR: X coordinate weight.

WYDIR: Y coordinate weight.

WOBJH: Vector of object height weights.

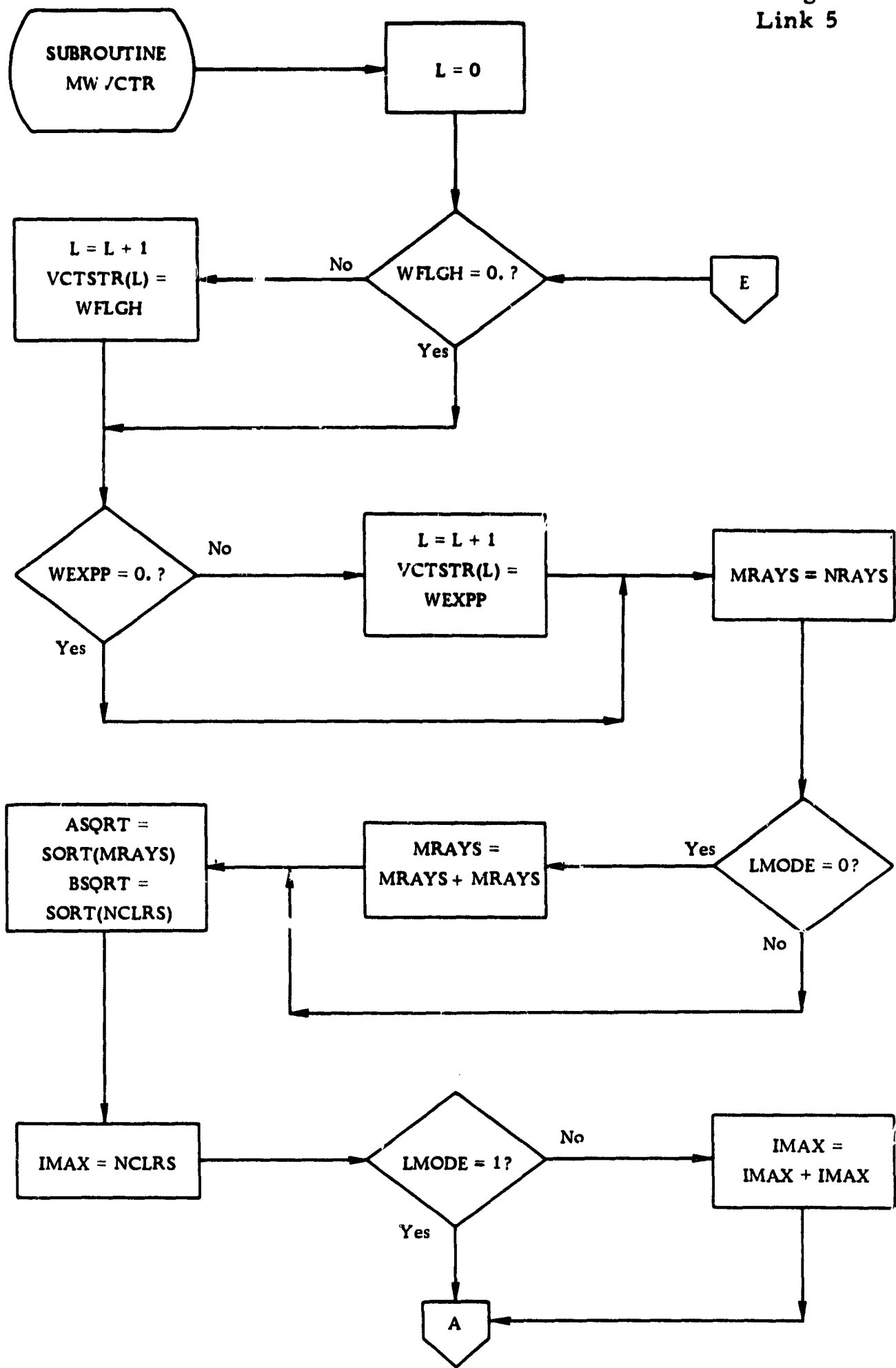
WCLRS: Vector of color weights.

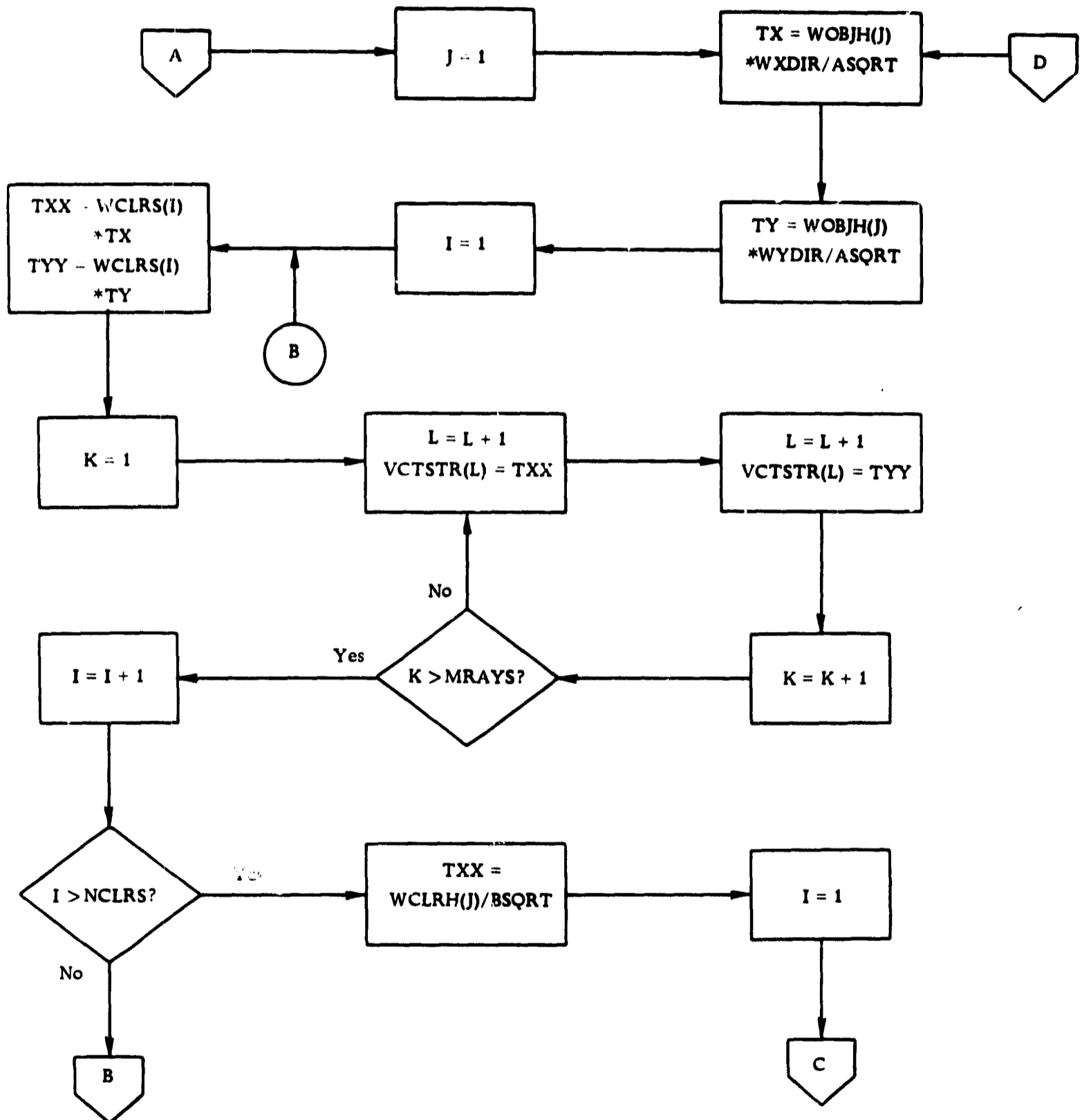
WCLRH: Vector of lateral chromatic aberration weights.

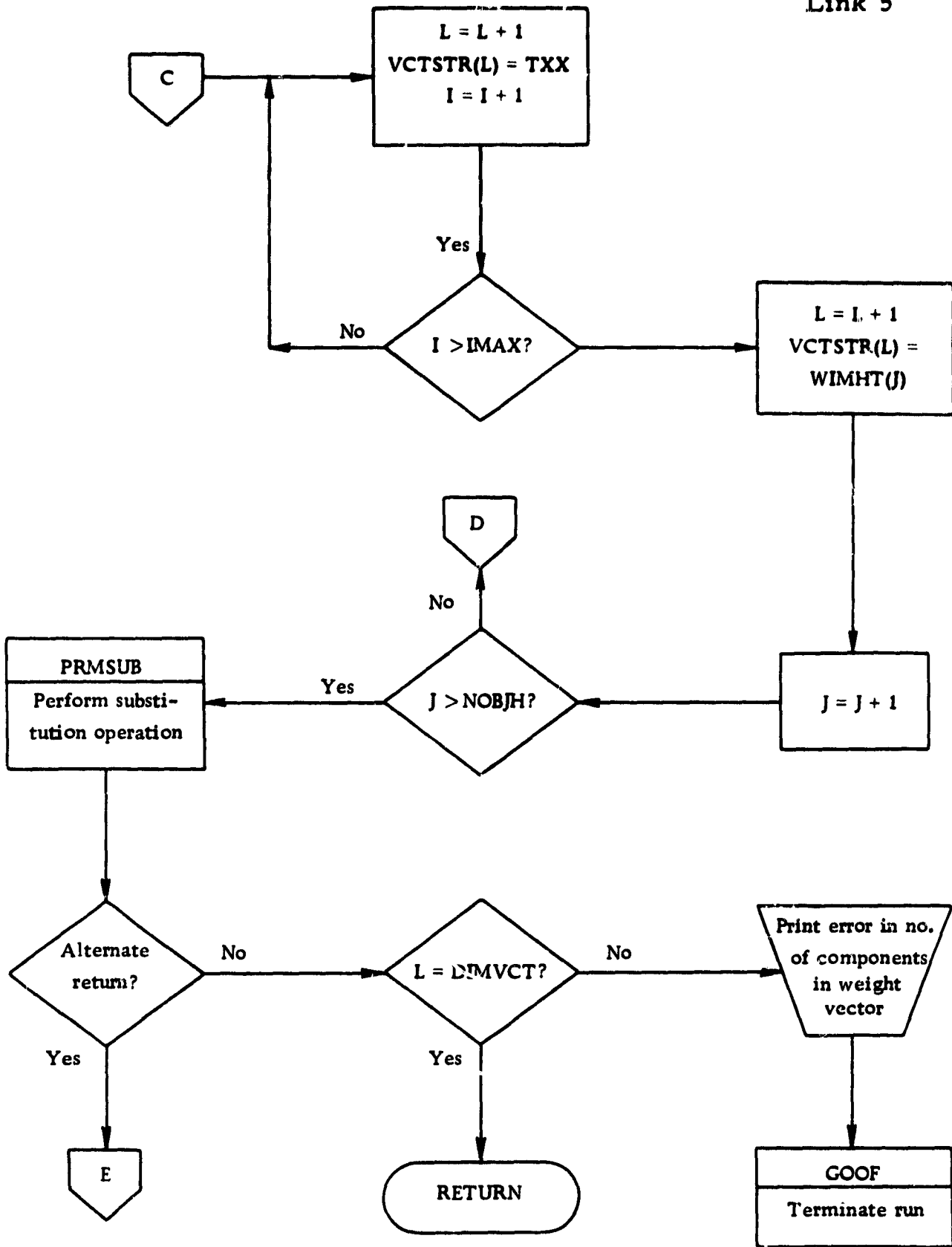
WIMHT: Vector of image height deviation weights.

Utility Routines and Common References

COMMON/DATA/
COMMON/TMDESN/
PRMSUB
GOOF







3.5.12 STAR2Z (STAR2)

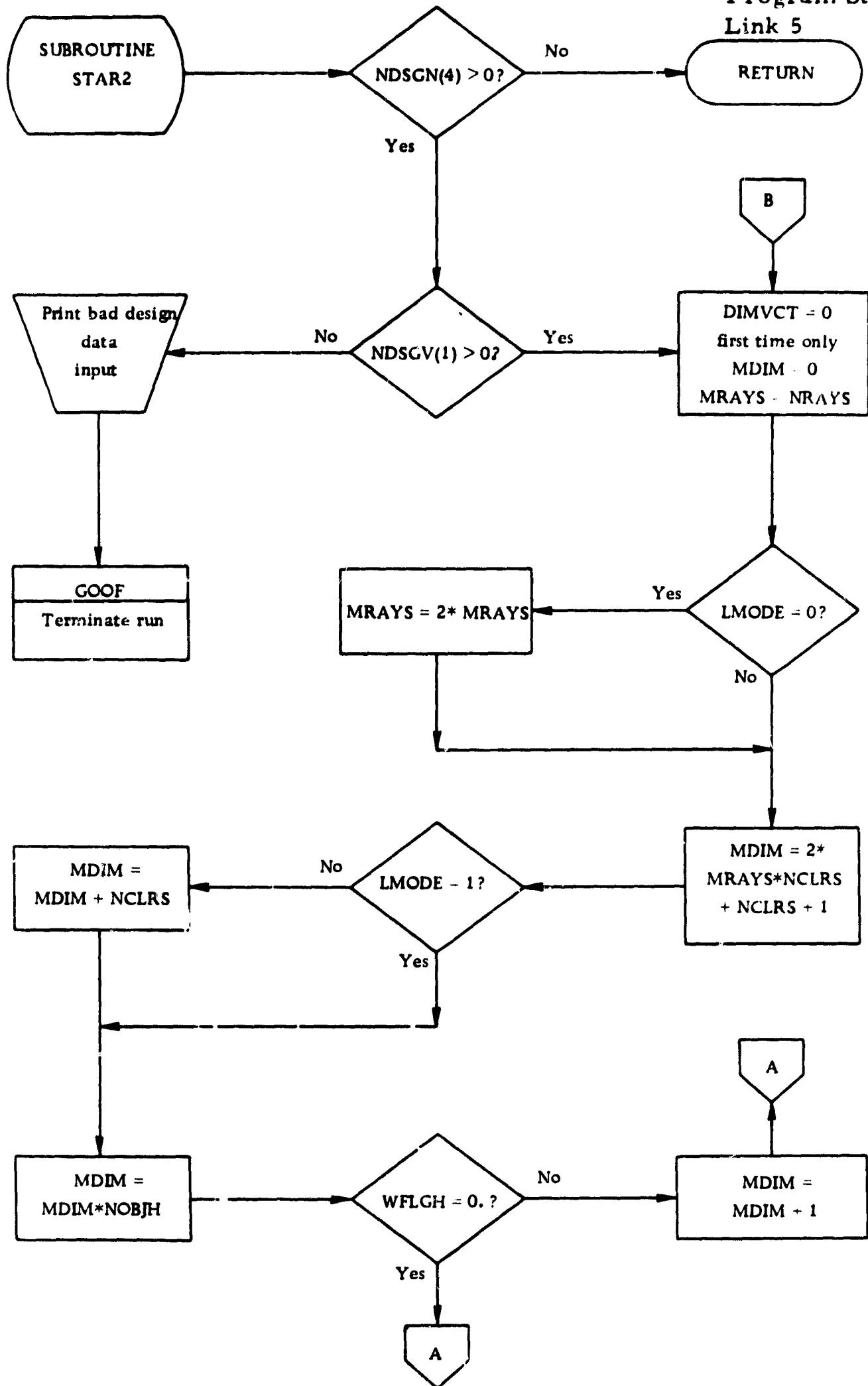
STAR2Z is the control routine for the design logic of the program. Several tests are made to determine if the design specifications are valid. If the design problem is too large for the available computer storage, the run must be terminated. After NDSGV(4) successful iterations have been made the program makes a normal return.

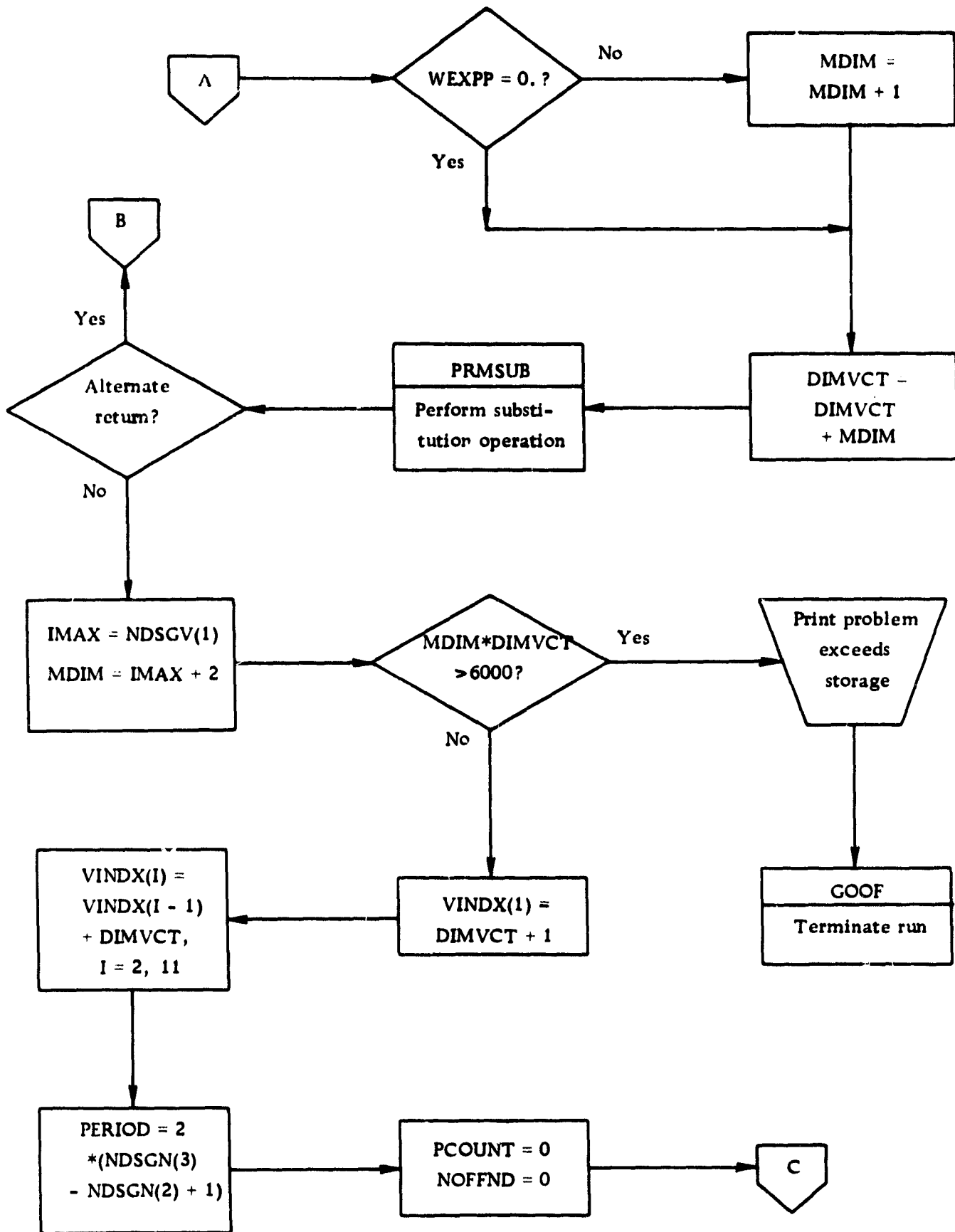
Calling Sequence

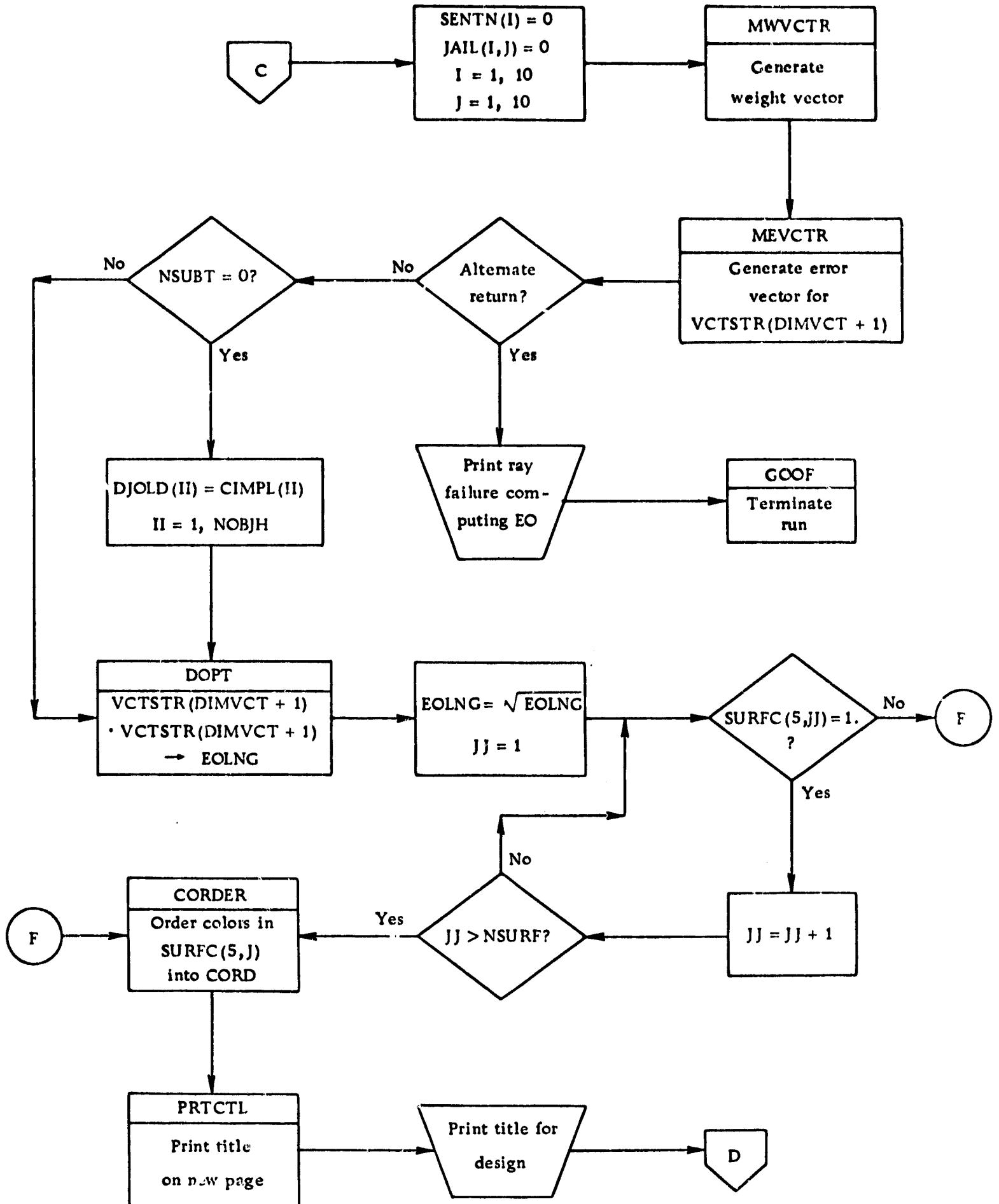
CALL STAR2

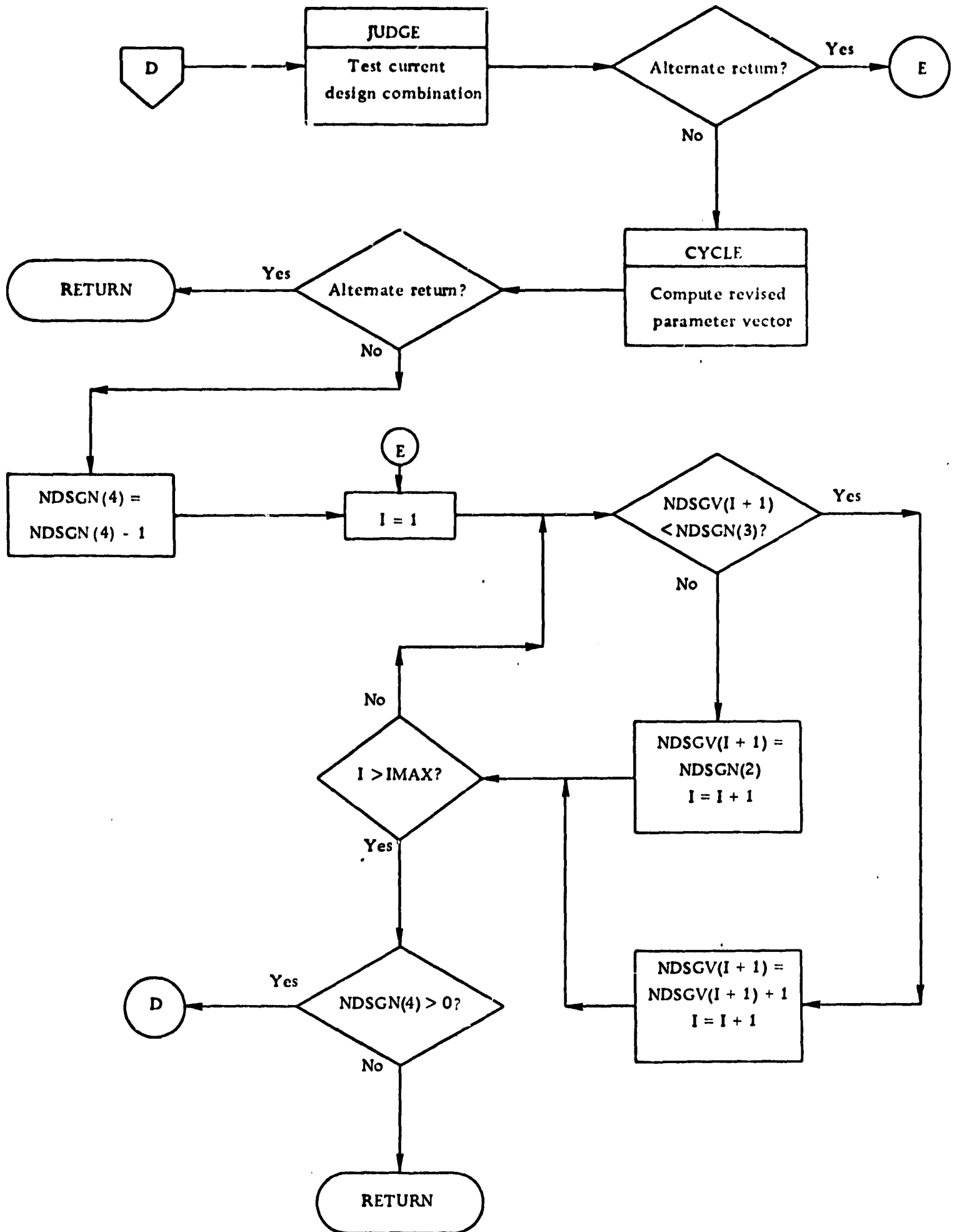
Utility Routines and Common References

COMMON/DATA/
COMMON/PRNT/
COMMON/TMDESN/
CORDER
CYCLE
DOTP
GOOF
JUDGE
MEVCTR
MWVCTR
PRMSUB
PRTCTL









3.5.13 MATINV (MATINV)

MATINV is a standard JPL library subroutine whose purpose and usage are described in the attached writeup. A flow chart has not been included in this instance.

IDENTIFICATION

MATINV/Matrix Inversion with Accompanying Solution of Linear Equations

PURPOSE

FORTRAN IV Subroutine solves the matrix equation $AX = B$, where A is a square coefficient matrix and B is a matrix of constant vectors. A^{-1} is also obtained; indeed, inversion may be the sole aim in a particular usage. Finally, the determinate of A is available; other possibly useful information is available to the user in the arrays 'S1' and 'S2'.

METHOD

Jordan's method is used to reduce a matrix A to the identity matrix I through a succession of elementary transformations; $t_n t_{n-1} \dots t_1 A = I$. If these transformations are simultaneously applied to I and to a matrix B of constant vectors, the result is A^{-1} and X where $AX = B$.

USAGE

Entrance is made via the FORTRAN statement in the calling program:

```
CALL MATINV (NDIM,A,N,B,M,DETERM,S1,S2)
```

- where
1. NDIM is the dimension of the arrays A,B. That is, the dimension specifications for these arrays are assumed to be: $A(NDIM,N')$, $B(NDIM,M')$; $N' \geq N$, $M' \geq M$.
 2. N is the order of A; $N \geq 1$.
 3. M is the number of column vectors in B.
 4. DETERM is the location in which the determinant is to be placed.
 5. S1 must be dimensioned at least $3N$, and S2 must be dimensioned at least as large as N .
 6. A,B,DETERM,S1,S2 are REAL names; NDIM,N,M are INTEGER names.

Suitable variable names may replace the dummy variables listed above at the users' discretion.

Notice that since the dimension information is provided by the calling subprogram, reassembly of MATINV is not required for different matrix sizes.

At the return to the calling program, A^{-1} is stored in A and X is stored in B.

Entering $M = 0$ or negative signals that the routine is to be used solely for inversion; note, however, that in the CALL statement an entry corresponding to B must still be present.

CODING INFORMATION

1. MATINV is the entry point name for deck MVRT1.
2. Storage required: $(720)_8 = (464)_{10}$ locations plus user supplied arrays S1 and S2.
3. Running time is approximately equal to $N^2 - 4N + 6$ milliseconds.

The following times were obtained for inversion of ill-conditioned matrices:

<u>ORDER(N)</u>	<u>ACTUAL TIME</u>	<u>$N^2 - 4N + 6$</u>
2	2.33ms	2
3	2.93	3
5	11.33	11
6	17.33	13
7	25.50	27
8	37.33	33
10	67.50	66

4. Array 'S1' is ordered as follows:
 - a. S1(1) to S1(N) is an array to prevent duplicate pivotings on any single row.
 - b. S1(N+1) to S1(3N) is a 2-column array which records consecutive row interchanges.
5. Array 'S2' is the array of pivot elements used in the inversion.

LINK 6

Link 6 is called upon whenever option 3 is to be executed, the communication being from FOLDP to STAR3. STAR3 employs subroutine SMPOP (Link 4) to produce average x, average y, rms x, rms y, image height, and spot size on the principal image plane and a specified set of alternate planes.

3.6.1 PARAXZ (PARAX)

PARAXZ traces a ray from the object point (ρ_o, Ω_2, π) through $(\hat{x}_o, \hat{y}_o) = (0., \Omega_2)$ on the entrance pupil to $P_1 = (x_1, z_1, z_1)$ on σ_N with direction $\vec{Q}_1 = (Q_{x1}, Q_{y1}, Q_{z1})$. Using P_1 and \vec{Q}_1 , PARAXZ computes F_L and F_P .

If the values for F_L and F_P are 0., this indicates a value of infinity for each. If the ray misses or reflects, the run is terminated.

Calling Sequence

CALL PARAX (COLOR, FOCALL, FOCALP)

COLOR: The color number currently used.

FOCALL: Computed focal length, F_L .

FOCALP: Computed focal point, F_p .

THTR: π (θ)

SNTHTR: 0. ($\sin \theta$)

CSTHTR: -1. ($\cos \theta$)

} Must be set by calling program

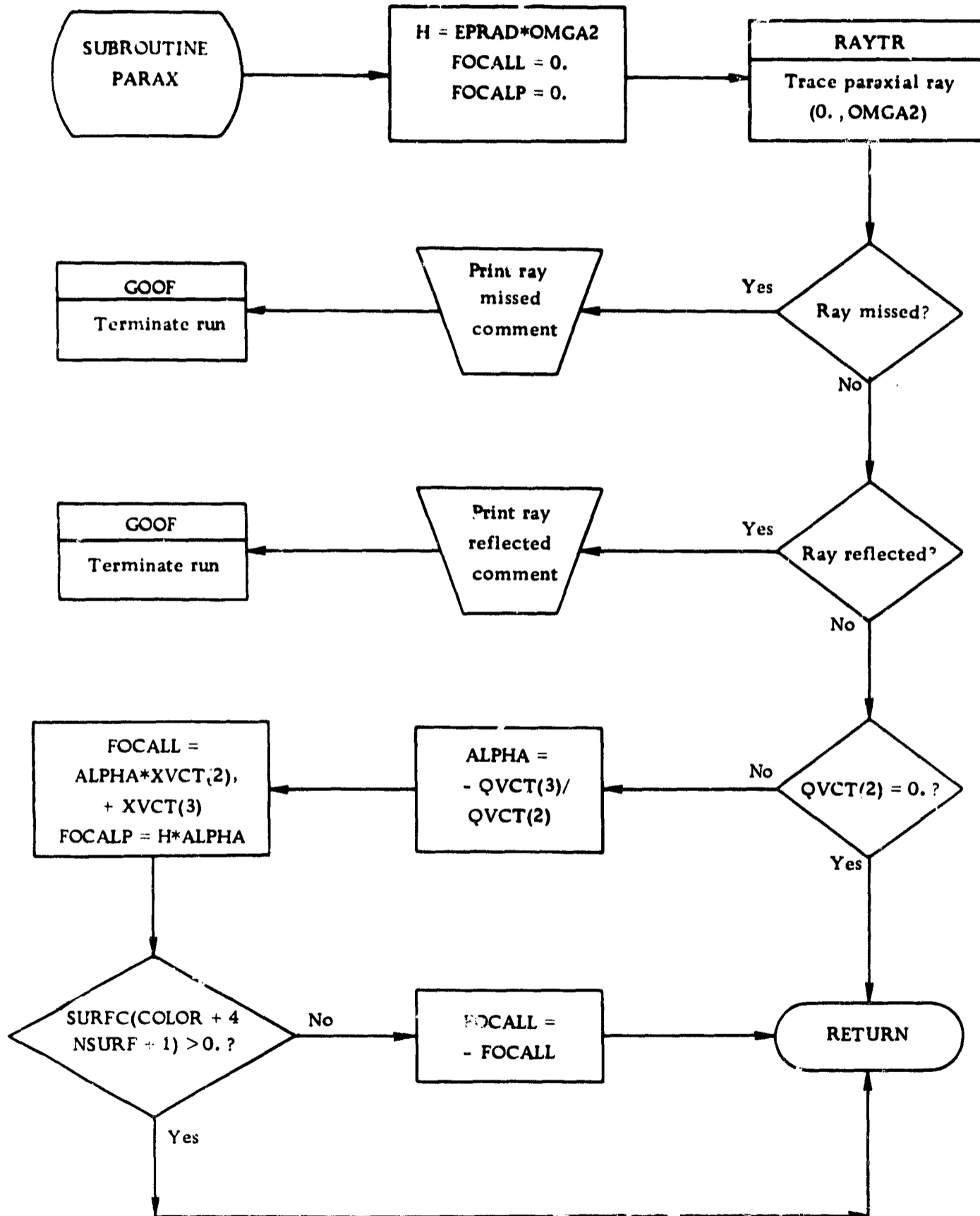
Utility Routines and Common References

COMMON/DATA/

COMMON/AZOBJ/

RAYTR

GOOF



3.6.2 STAR3Z (STAR3)

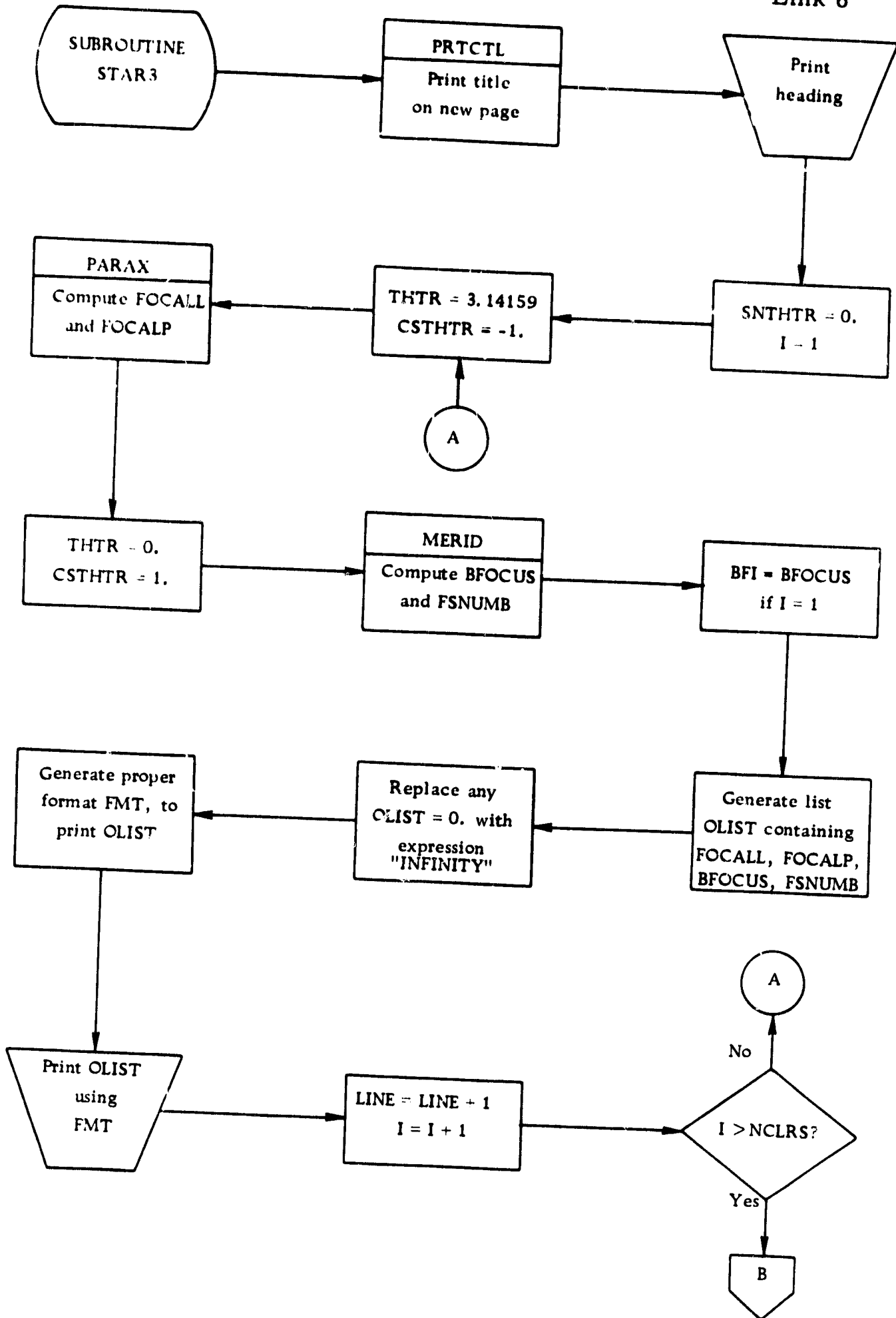
STAR3Z is the controlling subroutine for option 3 which causes optics diagnostic calculations. A table of focal length, focal point, back focus, and f number for each color is produced. A table of exit pupil positions for every combination of color and non-zero object height is output. For every specified color-object point combination the ray statistics are computed. Also for each combination the average x, average y, and spot size are calculated for all specified planes and three special planes chosen to minimize rms x, rms y, and spot size. Intermediate information is also output for assistance in further statistical computations. The values of M_R , N_M , N_{RF} , N_V , and \bar{N}_R are included in the ray statistics for each color-object point combination.

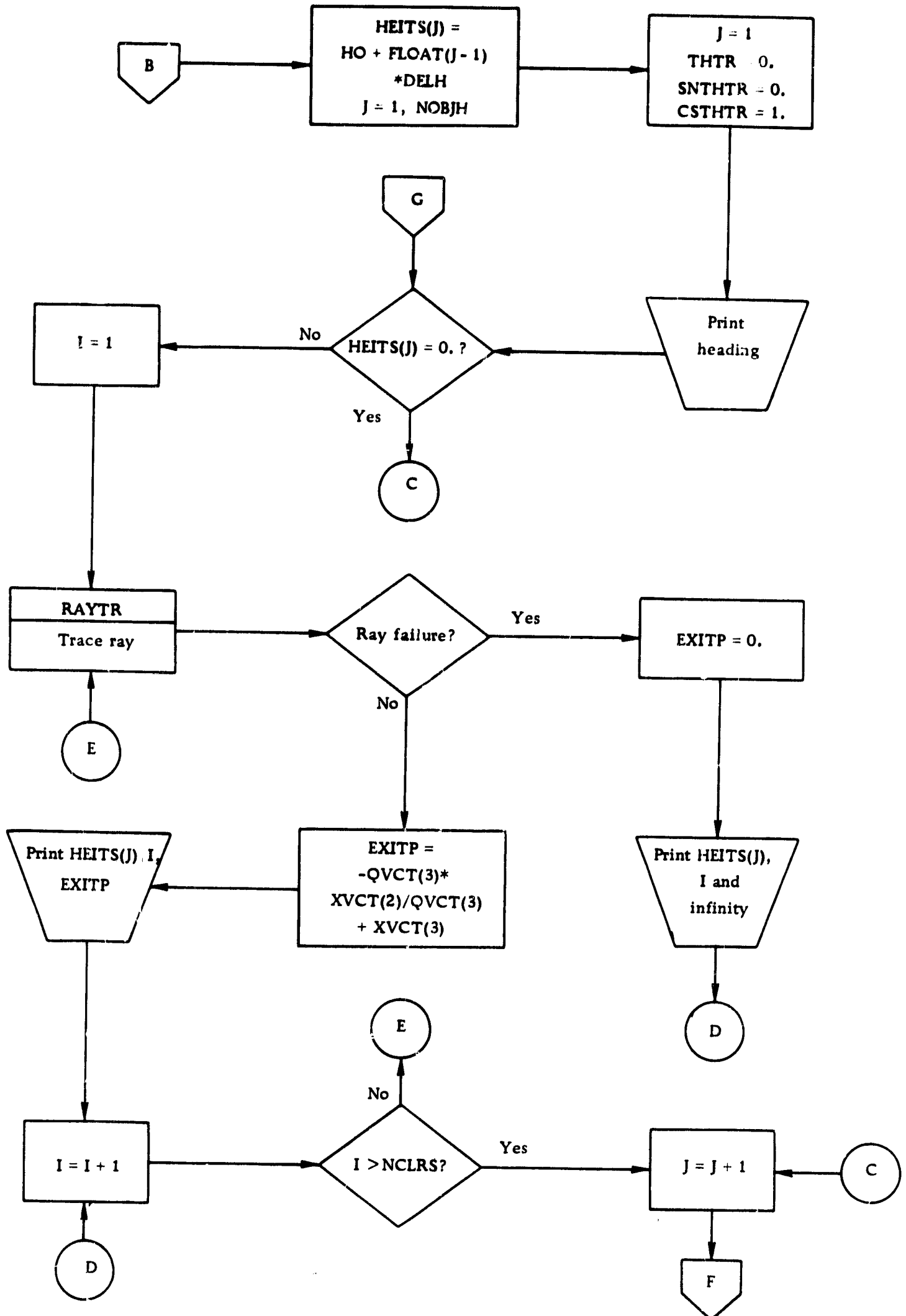
Calling Sequence

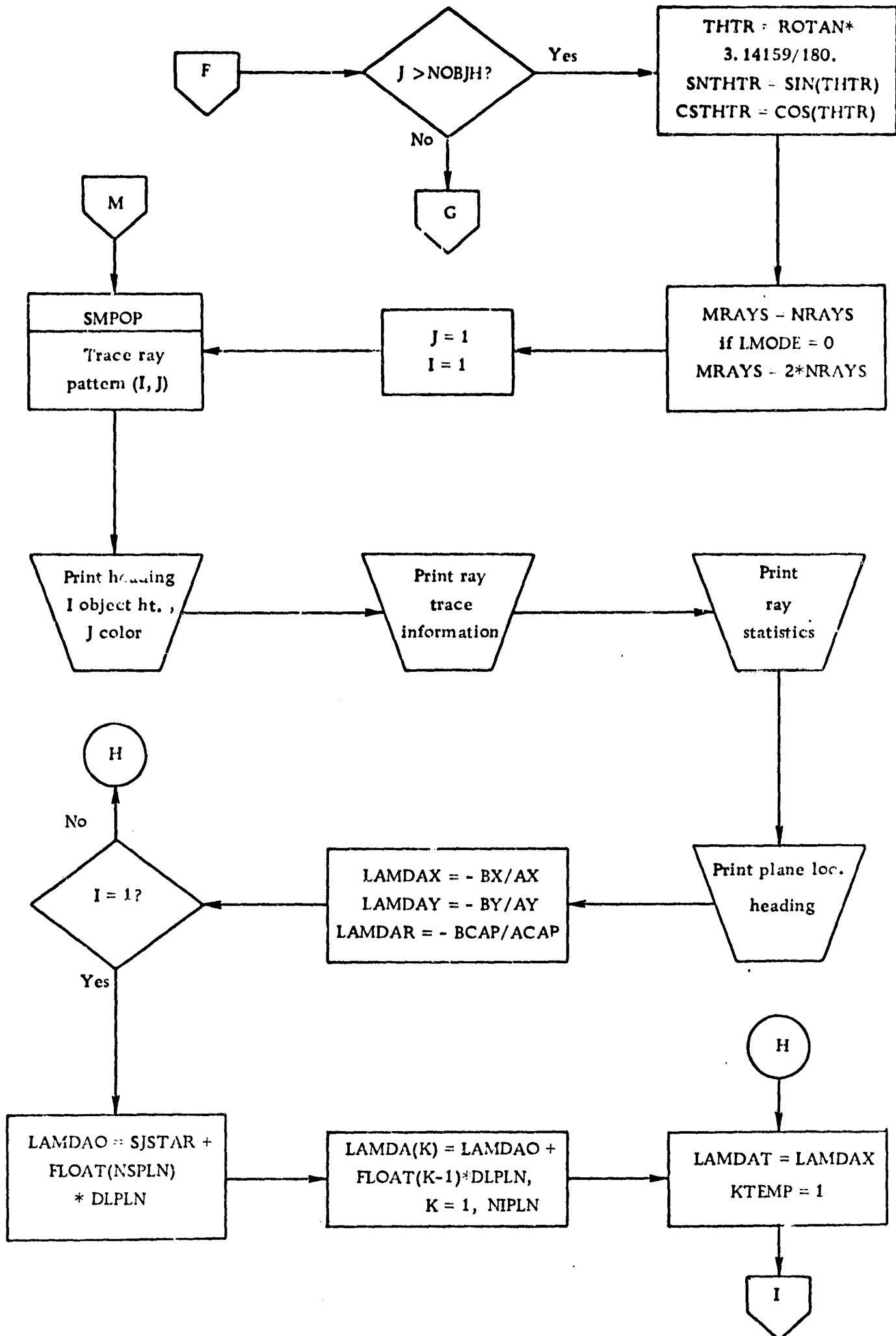
CALL STAR3

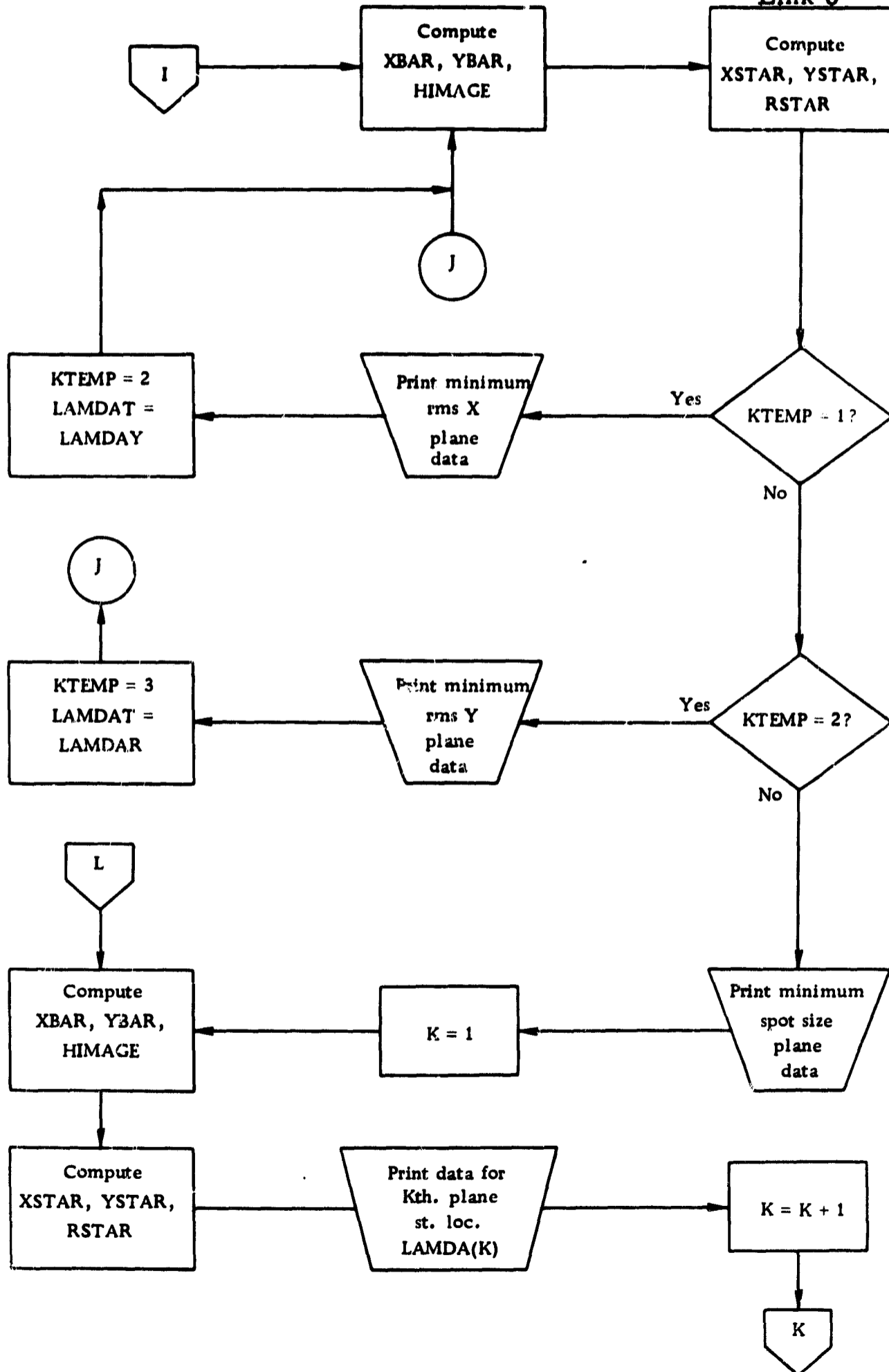
Utility Routines and Common References

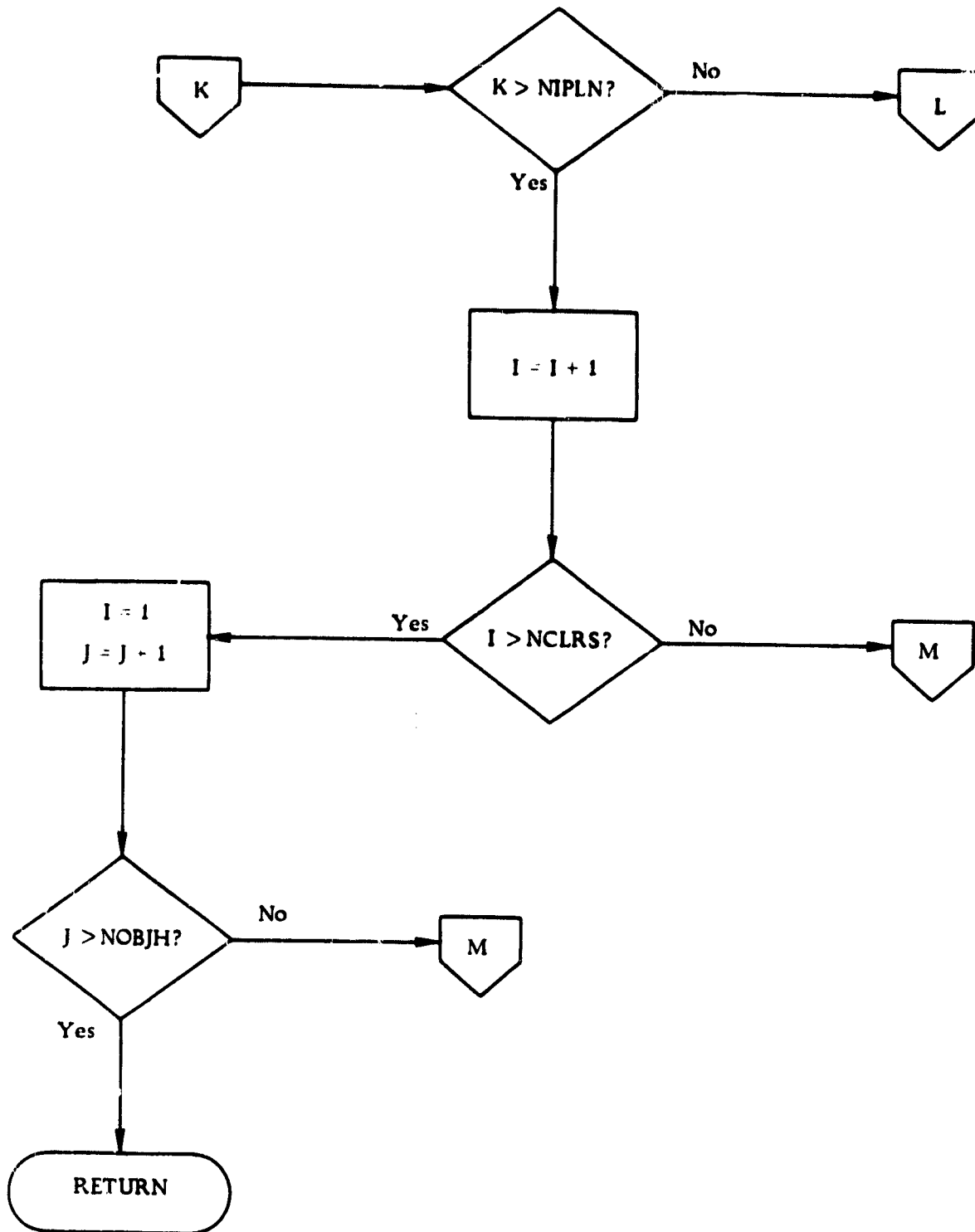
COMMON/DATA/
COMMON/TMPATT/
COMMON/AZOBJ/
COMMON/PRNT/
RAYTR
SMPOP
MERID
PARAX
PRTCTL











3.7 LINK 7

Link 7 is responsible for producing scaled spot diagram points plots on the printer-one plot for each object point-color combination. Subroutine STAR7 which is the control routine is entered directly from FOLDP whenever option 7 is to be executed. Subroutine SMPOP (Link 4) is employed to produce image coordinates on the principal plane which are transformed to printer coordinates by subroutine GETLC.

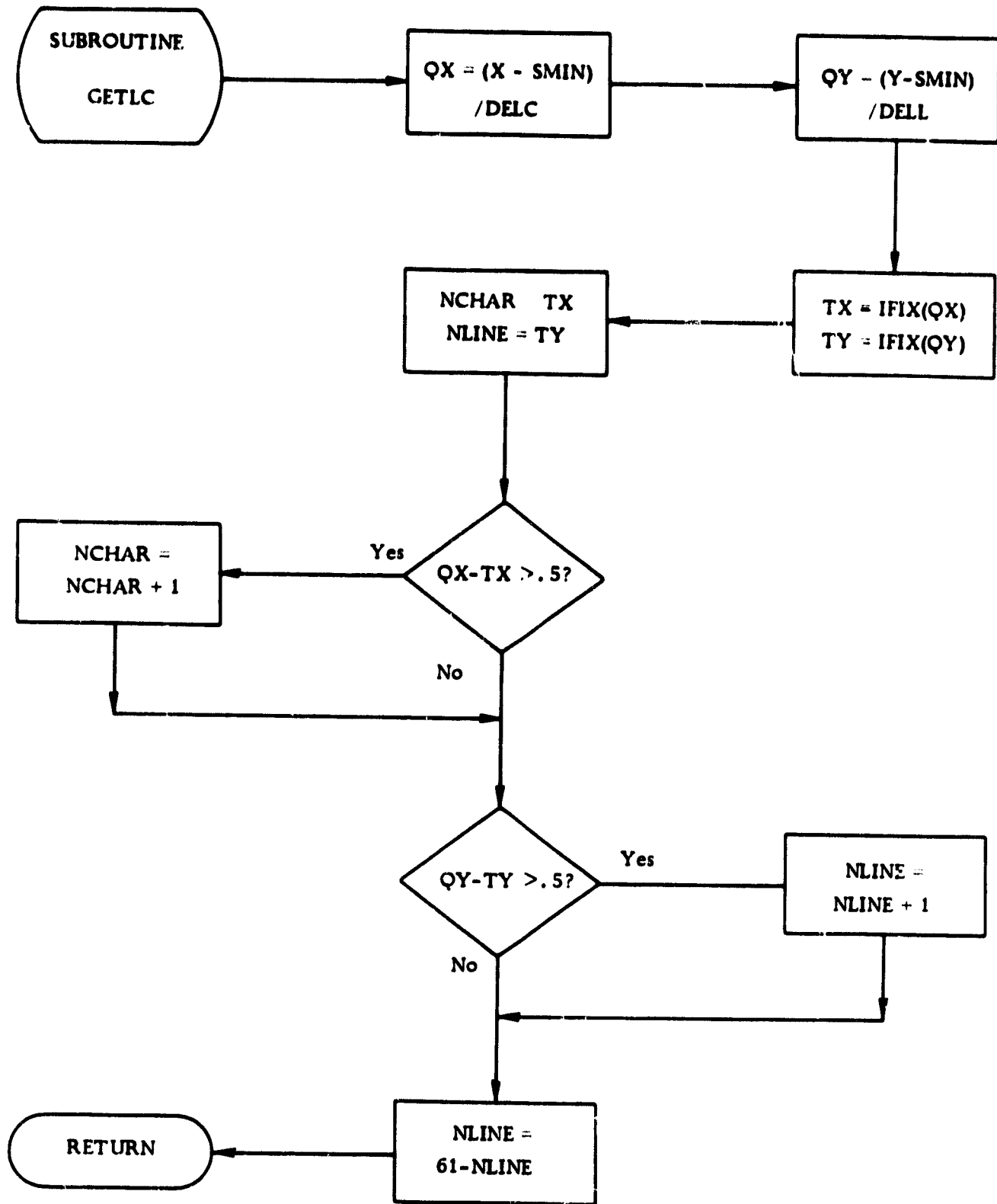
3.7.1 GETLCZ (GETLC)

GETLCZ is used to compute the character and line position of a given point. The point is then output according to its character and line position on the spot diagram point plots.

Calling Sequence

CALL GETLC (X, Y, SMIN, DELL, DELC, NCHAR, NLINE)

X: The X coordinate of the point.
Y: The Y coordinate of the point.
SMIN: Value assigned to lower left corner of grid.
DELL: Increment in the Y direction (line).
DELC: Increment in the X direction (character).
NCHAR: Character position of point on grid.
NLINE: Line position of point on grid.



3.7.2 STAR7Z (STAR7)

STAR7Z is used to produce the spot diagram point plots required for an option 7. The coordinates of the specified ray pattern on the unit radius entrance pupil are printed along with the line-character coordinates, and the plot character for each ray. A scaled point plot of the entrance pupil pattern is then produced on the printer. If there are 35 or fewer rays, each ray is assigned a unique plot character. Otherwise x's are used if the ray number exceeds 35.

For each combination of object point and color the ray pattern traced to the image plane. A tabular listing of each ray of a given object point and color is printed followed by its spot diagram point plot. The plot scale is the same for all point plots and is printed on each plot.

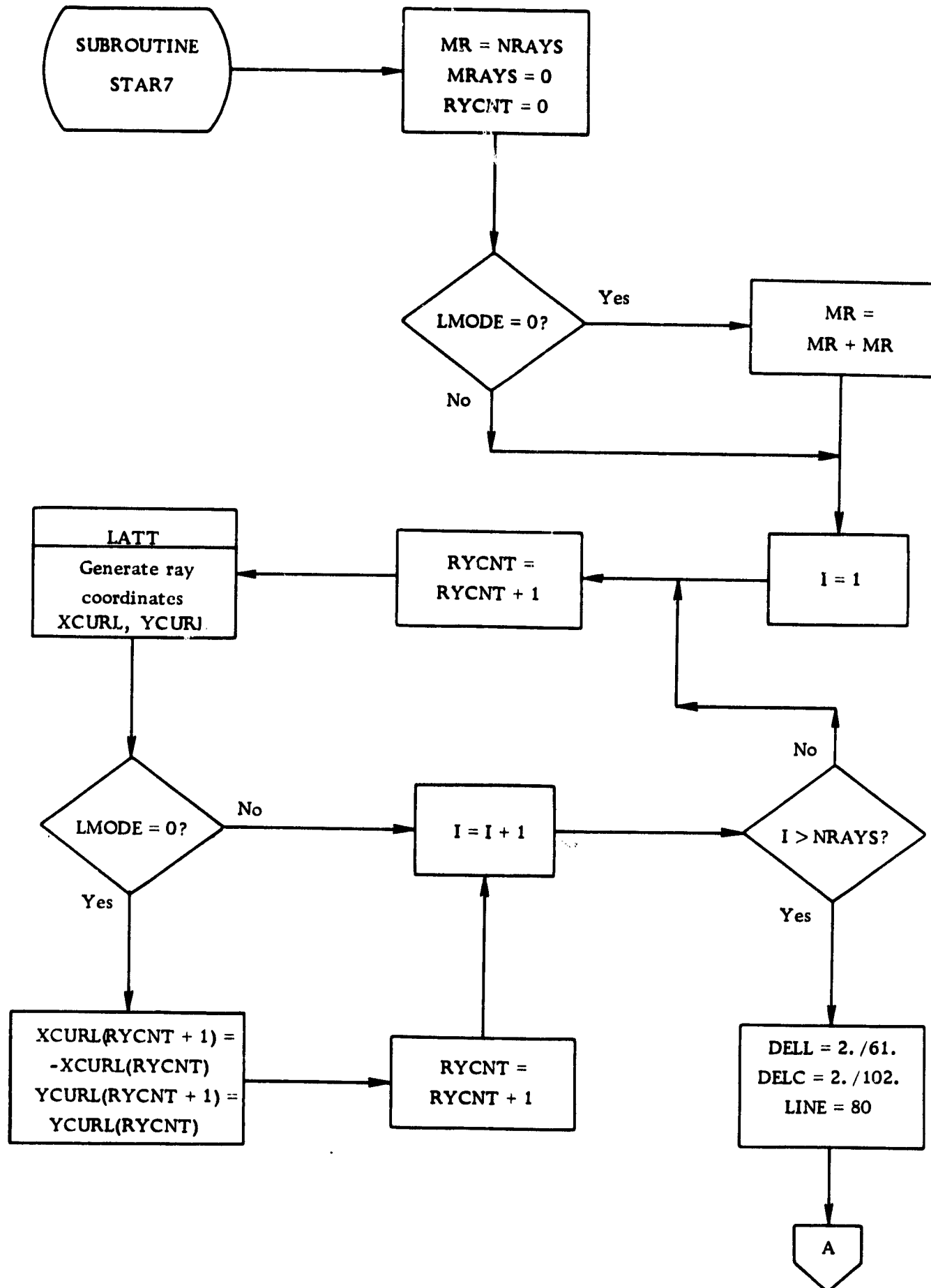
If any ray has failed (missed, reflected, or vignetted) an appropriate comment is printed in the tabular listing and the respective point is deleted from the plot.

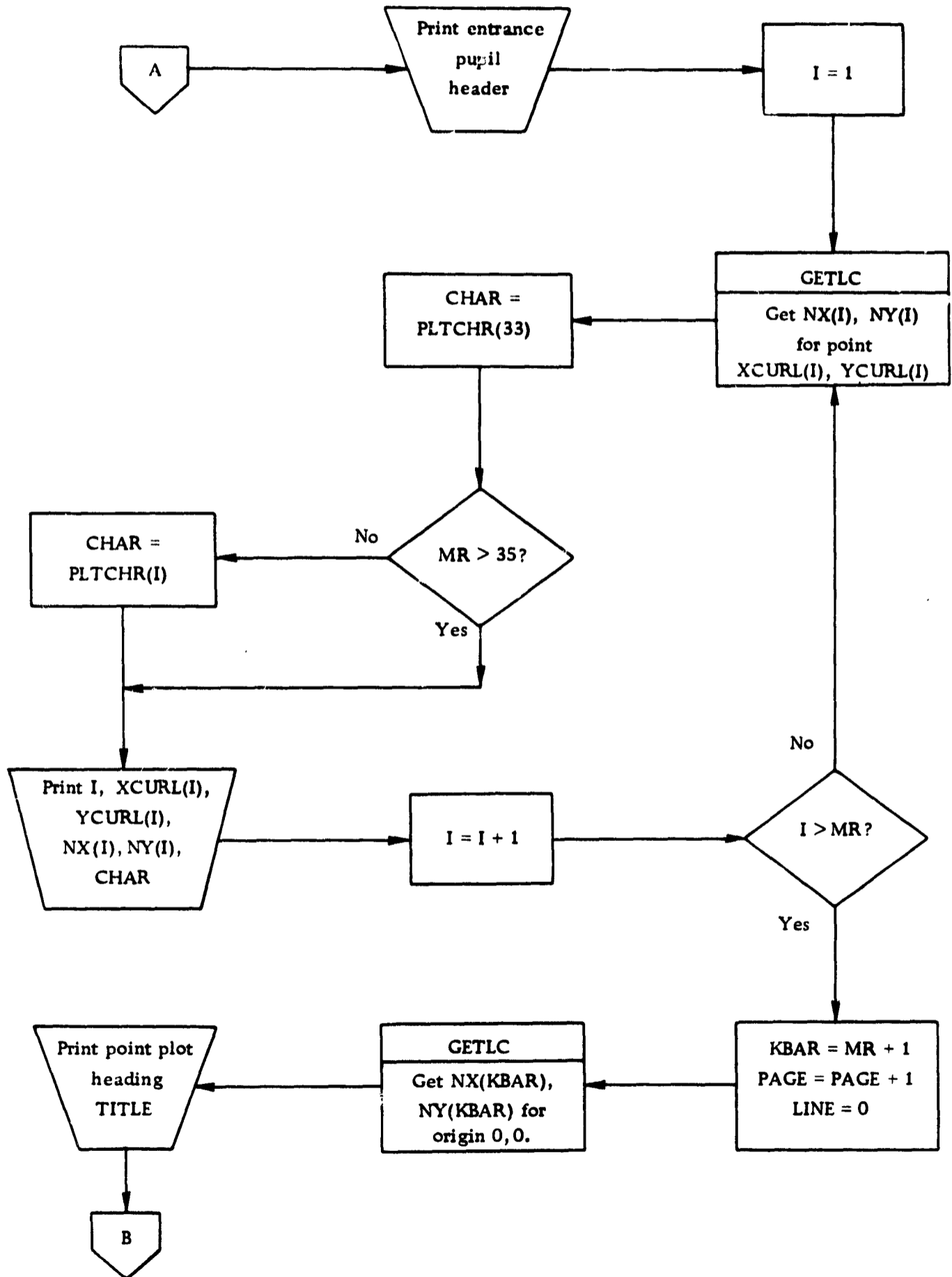
Calling Sequence

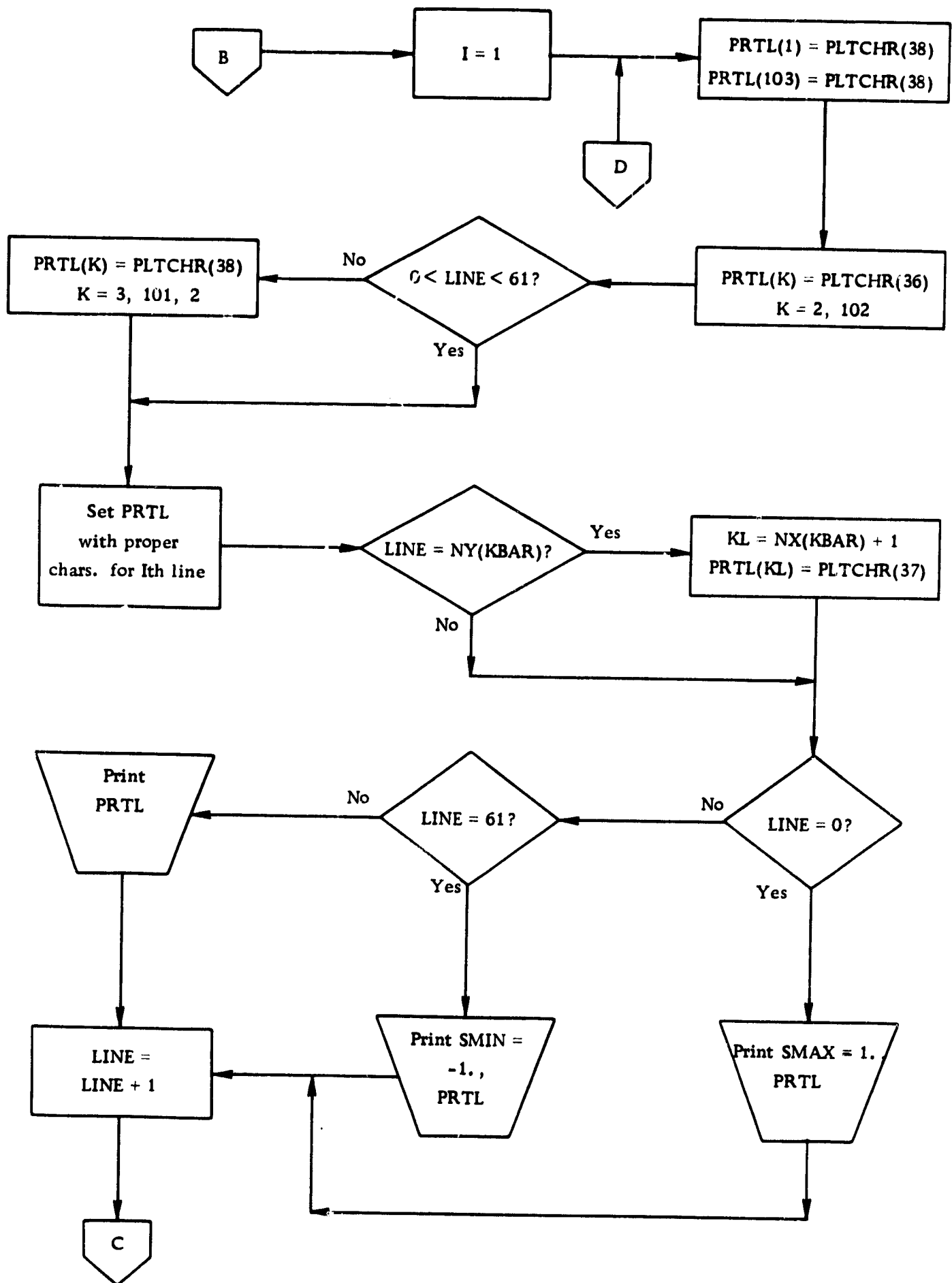
CALL STAR7

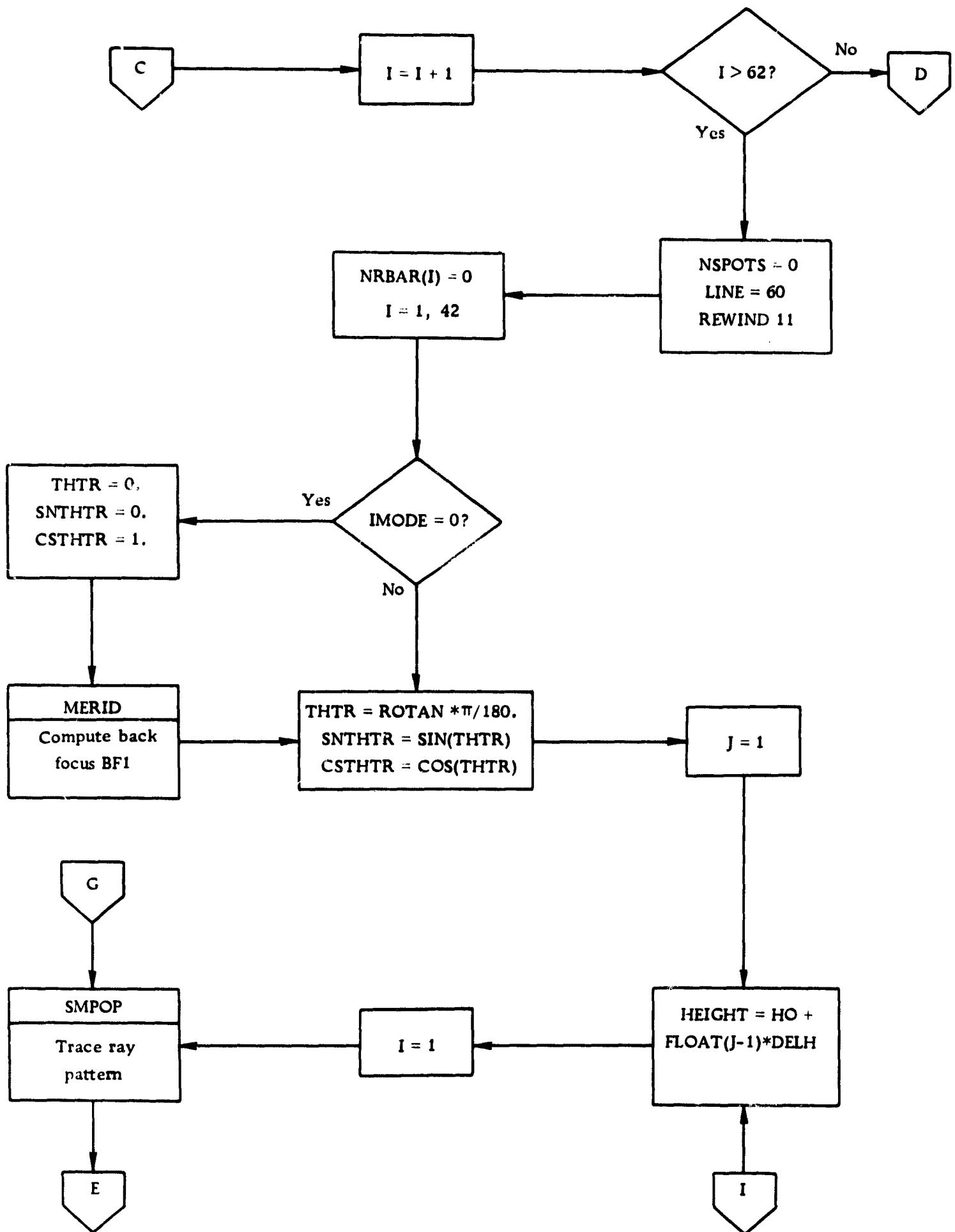
Utility Routines and Common References

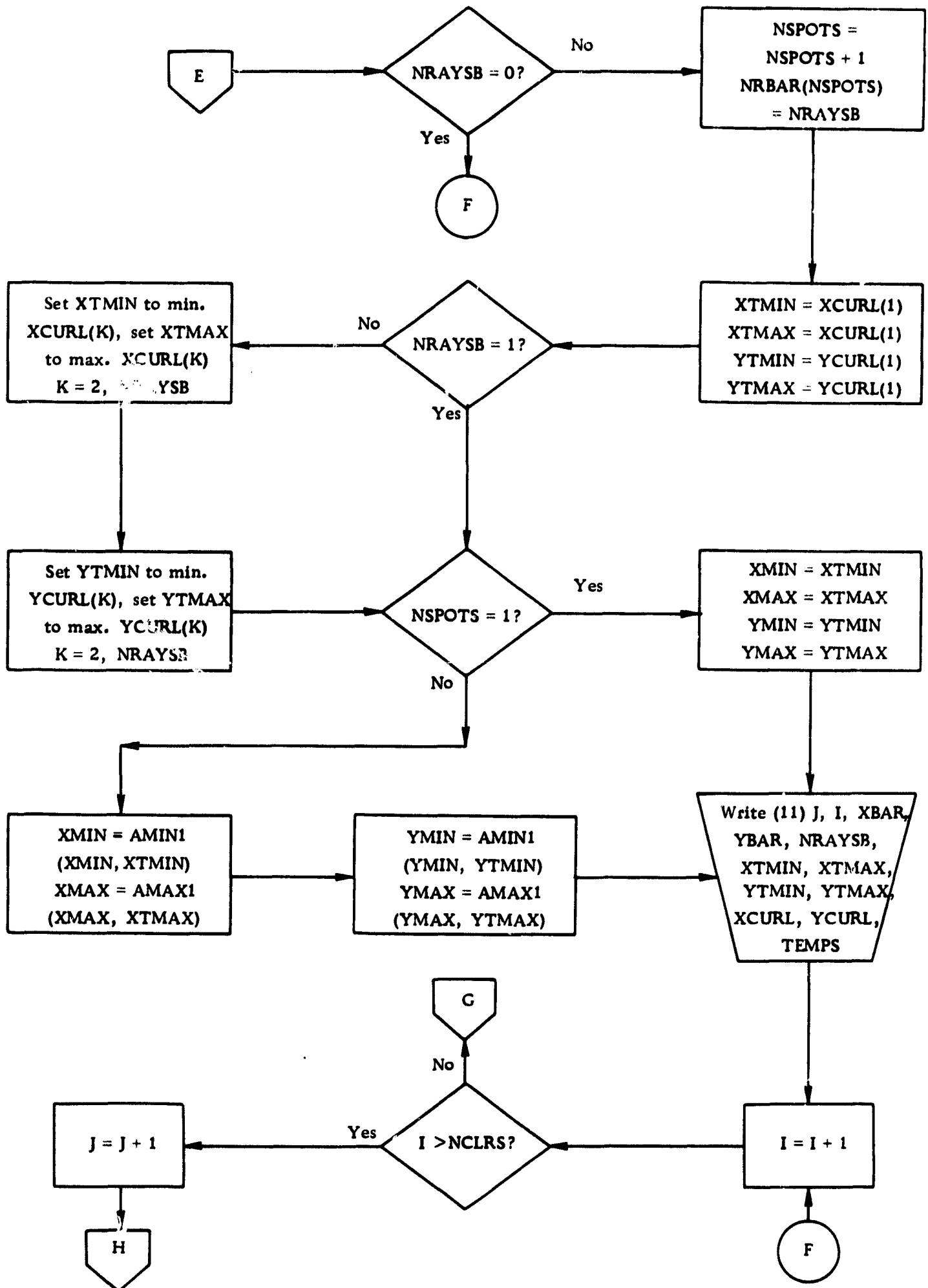
COMMON/DATA/	MERID
COMMON/TMPATT/	SMPOP
COMMON/PRNT/	GETLC
COMMON/AZOBJ/	SCALEK
LATT	PRTCTL

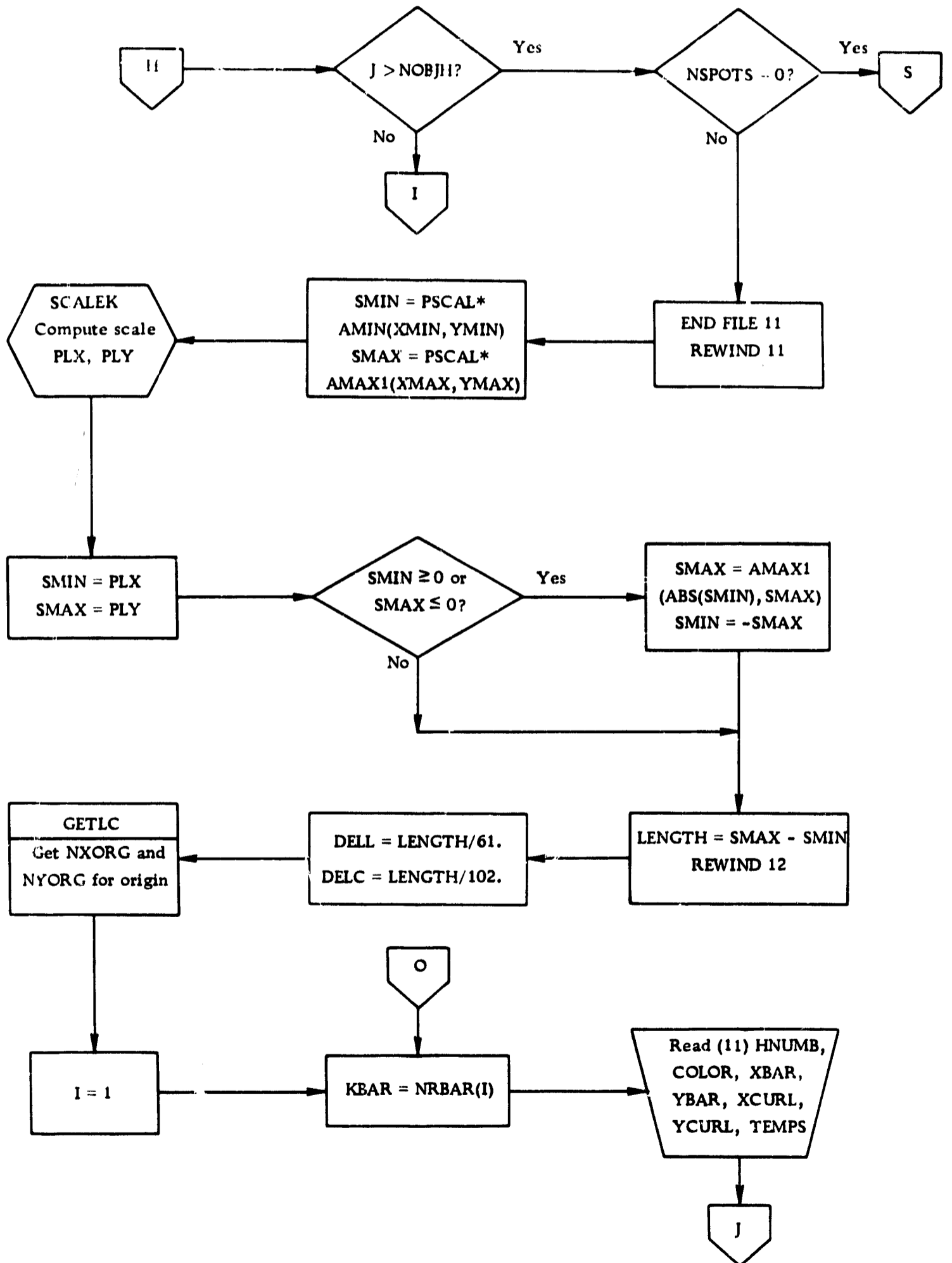


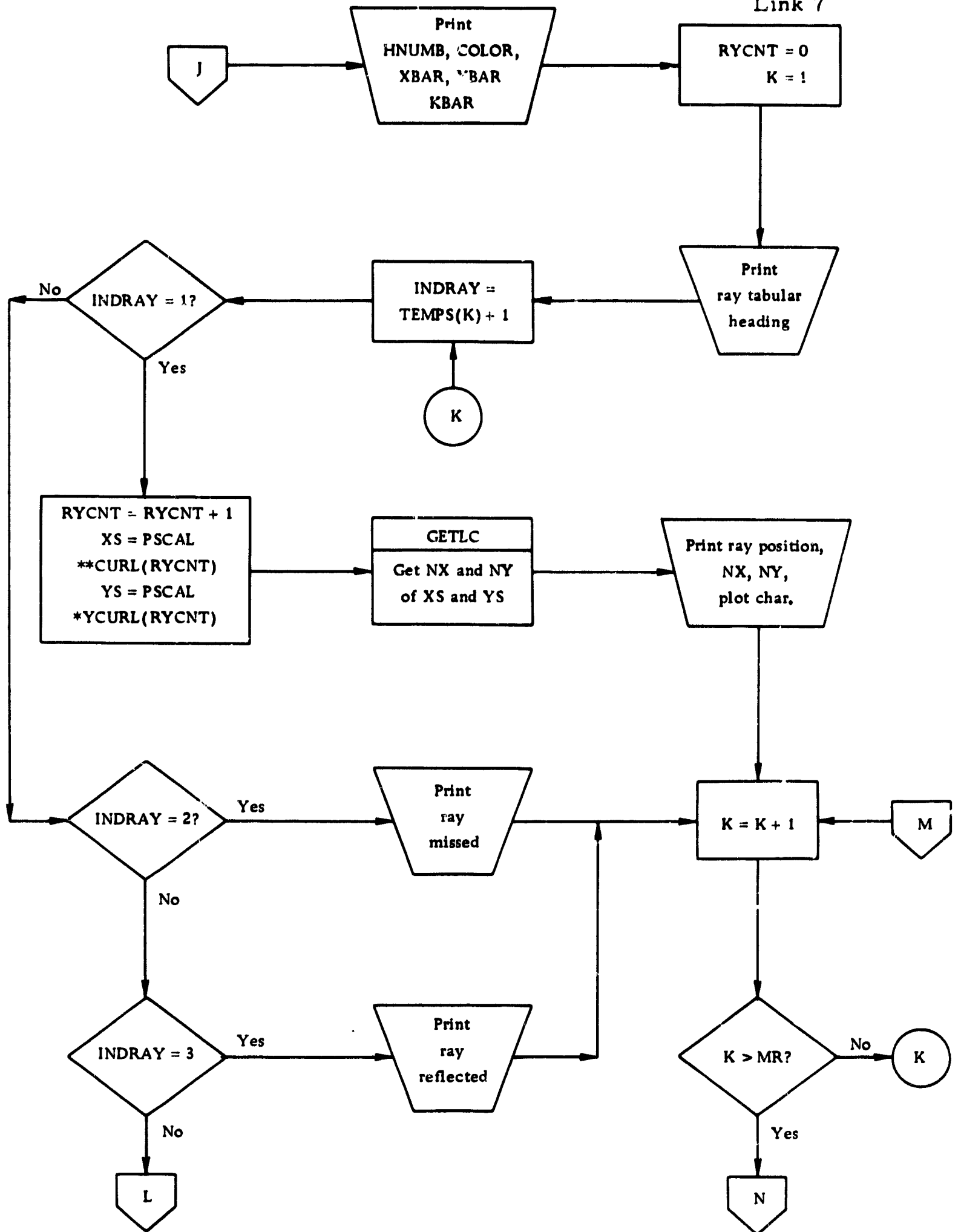


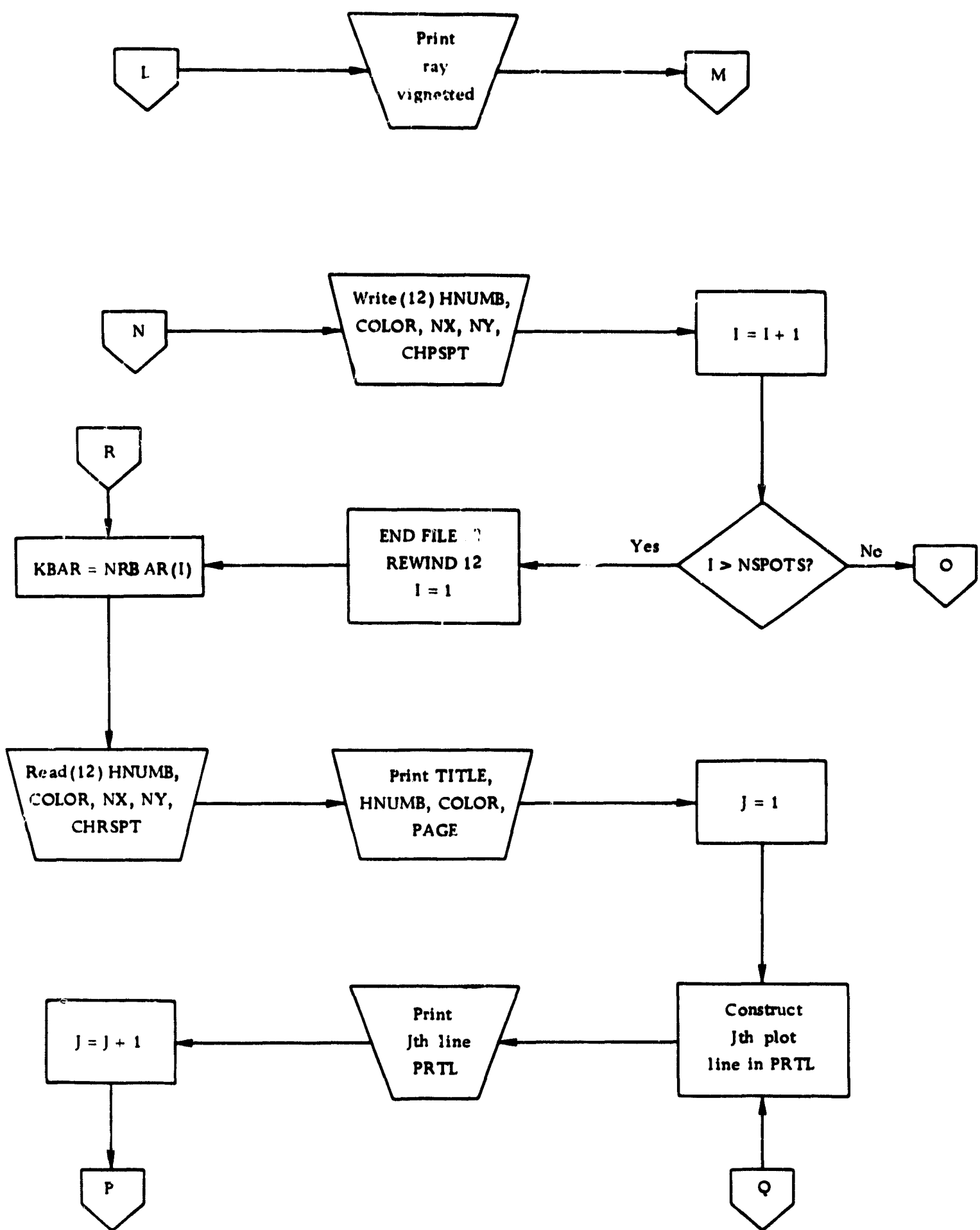


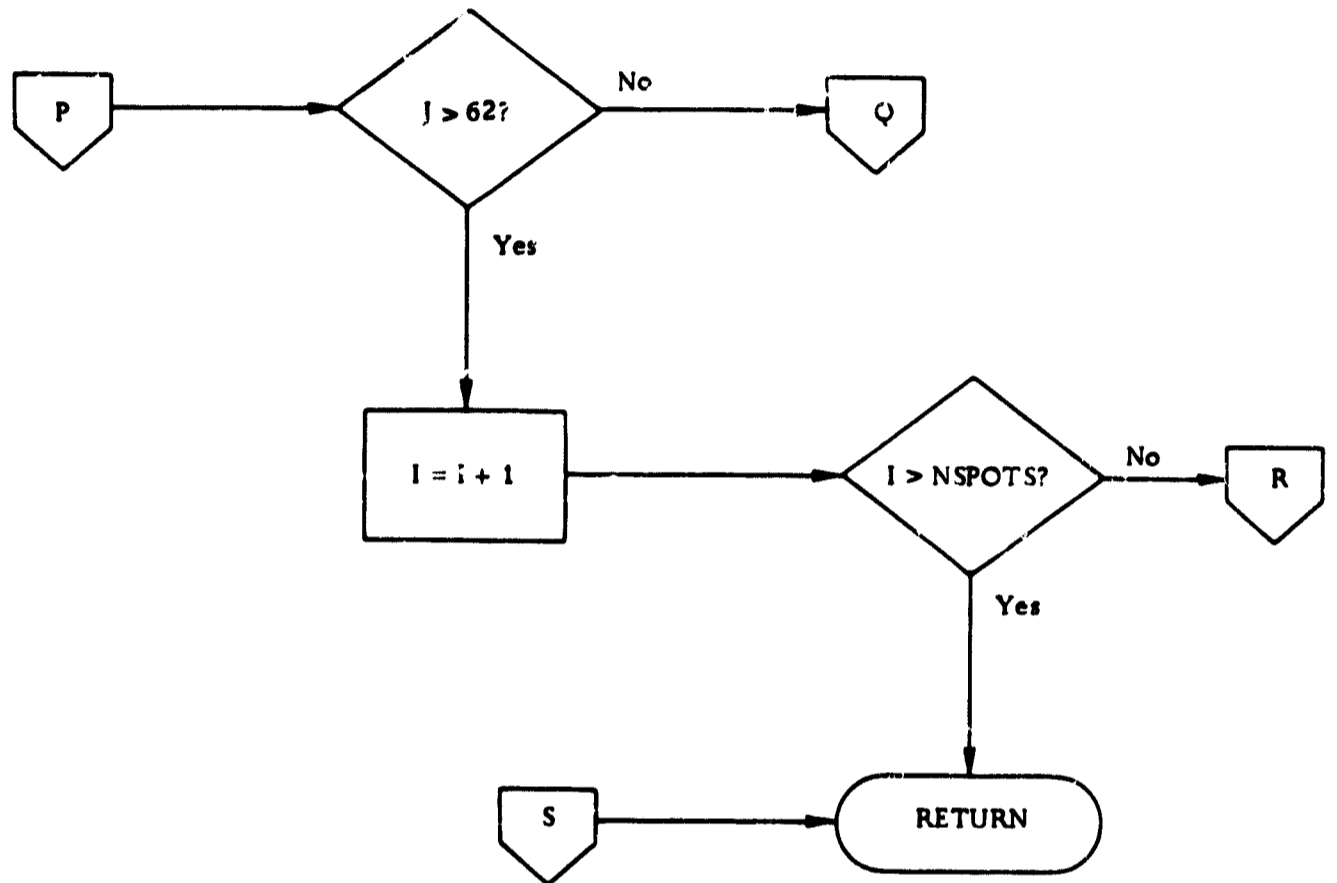












3.8 LINK 8

Link 8 performs a specified sensitivity analysis on the lens system under the control of option 8. One of the primary purposes of this type of analysis is to study the tolerance (sensitivity) of the design with respect to geometric fabrication parameters such as curvature and element spacing. Subroutine STAR8 employs SMPOP (Link 4) to evaluate the nominal and perturbed systems in order to produce sensitivity coefficients.

3.8.1 STAR8Z (STAR8)

STAR8Z performs the computations for the sensitivity analysis which is used to determine the effect of perturbing certain parameters. Statistical data for each object point and color for the nominal system is calculated and saved. Then each parameter specified by PERTB(I, 2) is perturbed by the percentage given in PERTB(I, 1) and the system is recalculated for each object point and color. If there were no ray failures the values for average x, average y, rms x, rms y, and spot size are printed for each object point and color combination. If the refocus option, REFOCS, has been set non-zero, values are also printed for the minimum rms x, rms y, or spot size plane location for all colors taken together. When all combinations of color and object point have been completed the perturbed parameter is restored, and the process is repeated using the next parameter in PERTB until all NPERTB parameters have been perturbed.

Calling Sequence

CALL STAR8

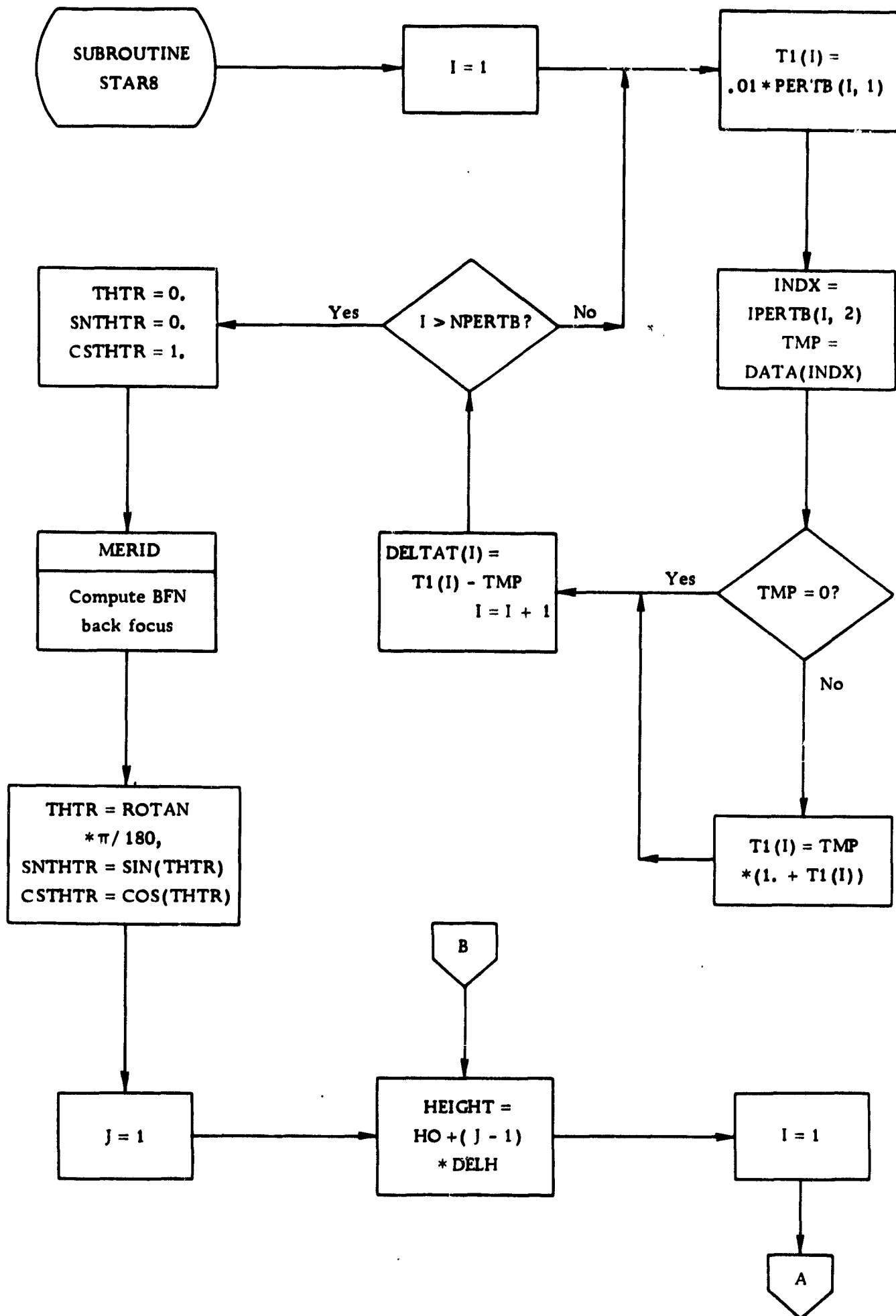
NPERTB: Number of parameters to be perturbed, M_p .

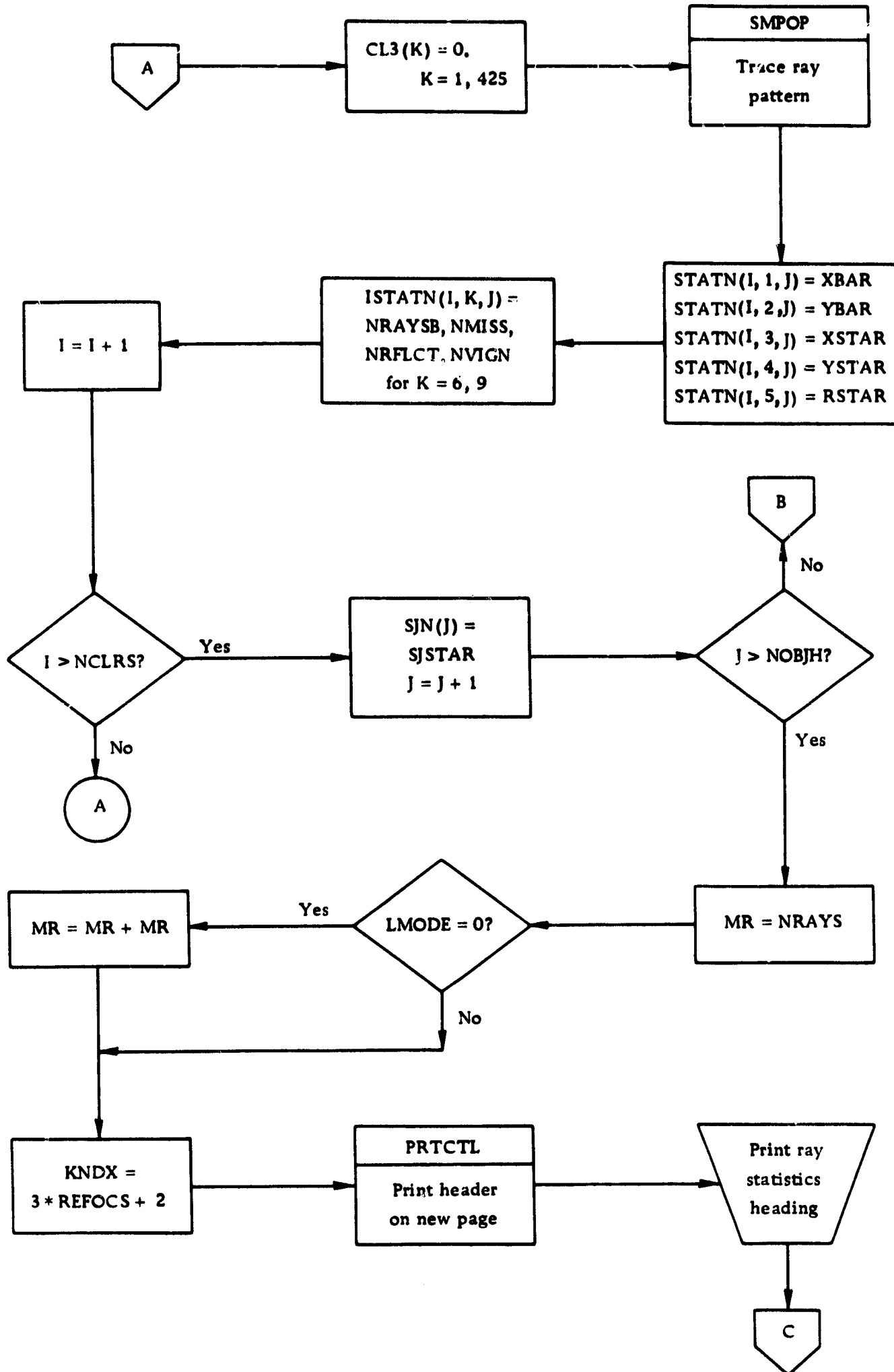
PERTB: Perturbation matrix each row of which contains the perturbation percent, index of parameter, and parameter symbol (2 words of 6 characters each).

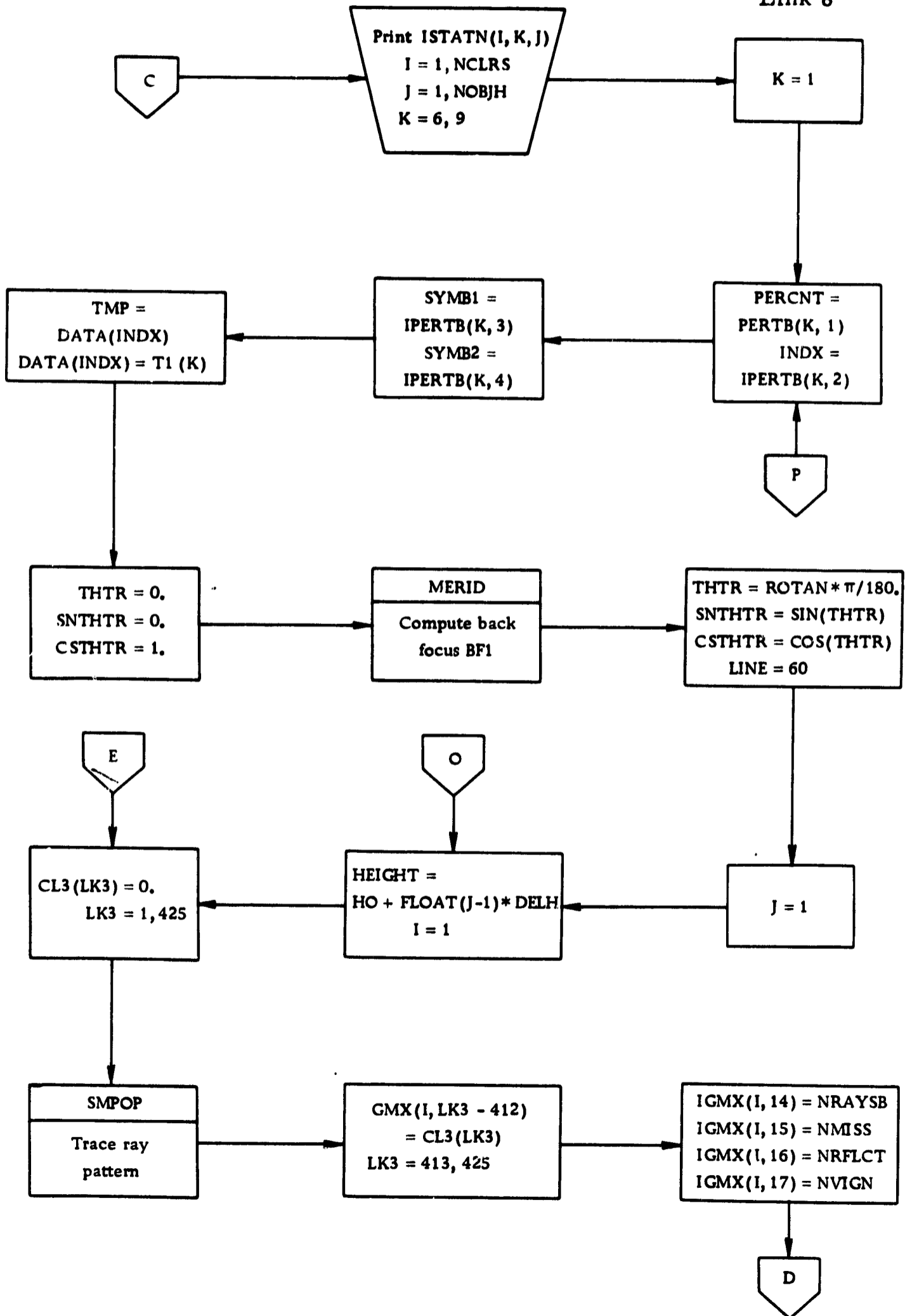
REFOCS: Flag which is set according to the type of lens refocusing desired.

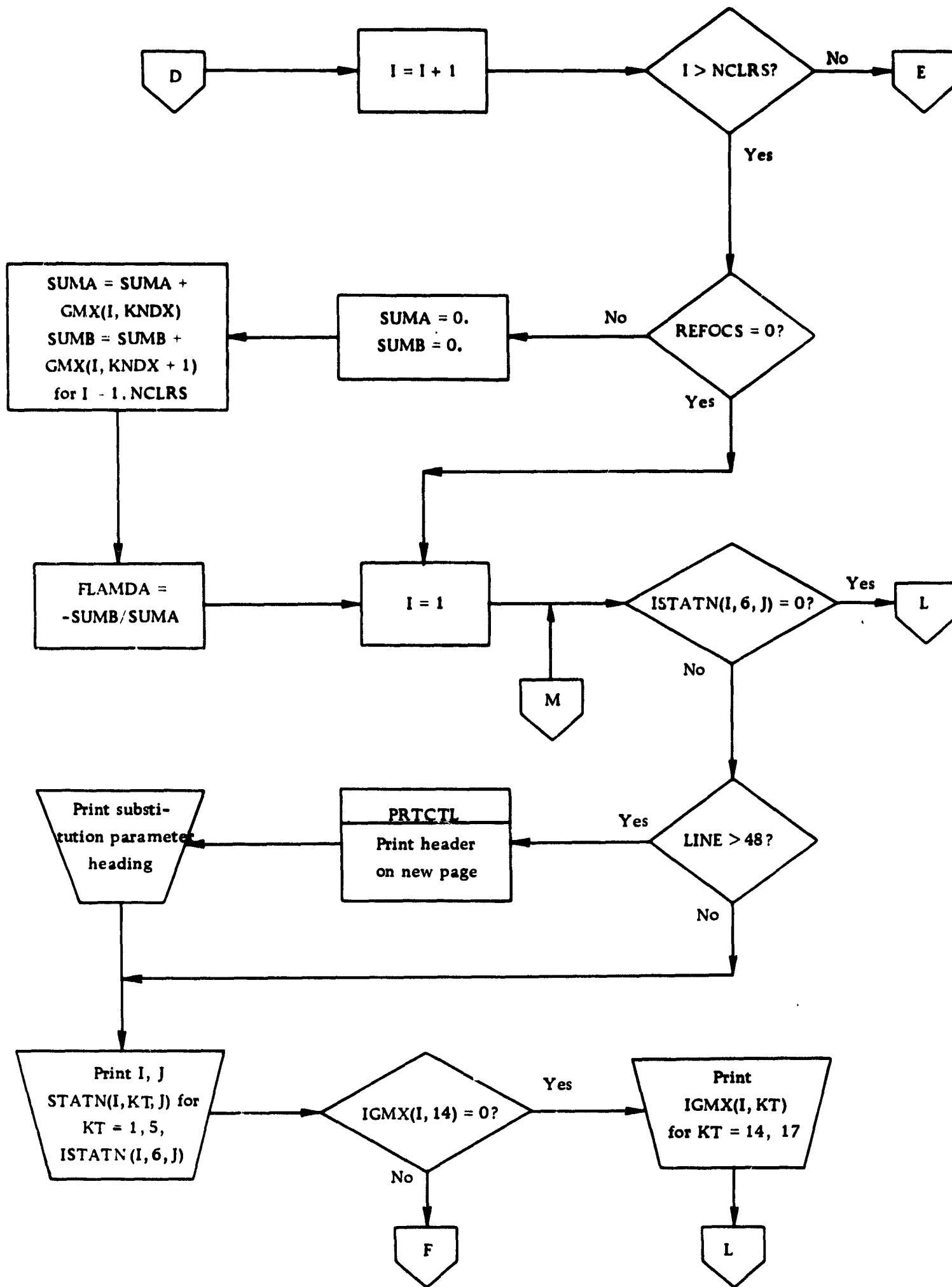
Utility Routines and Common References

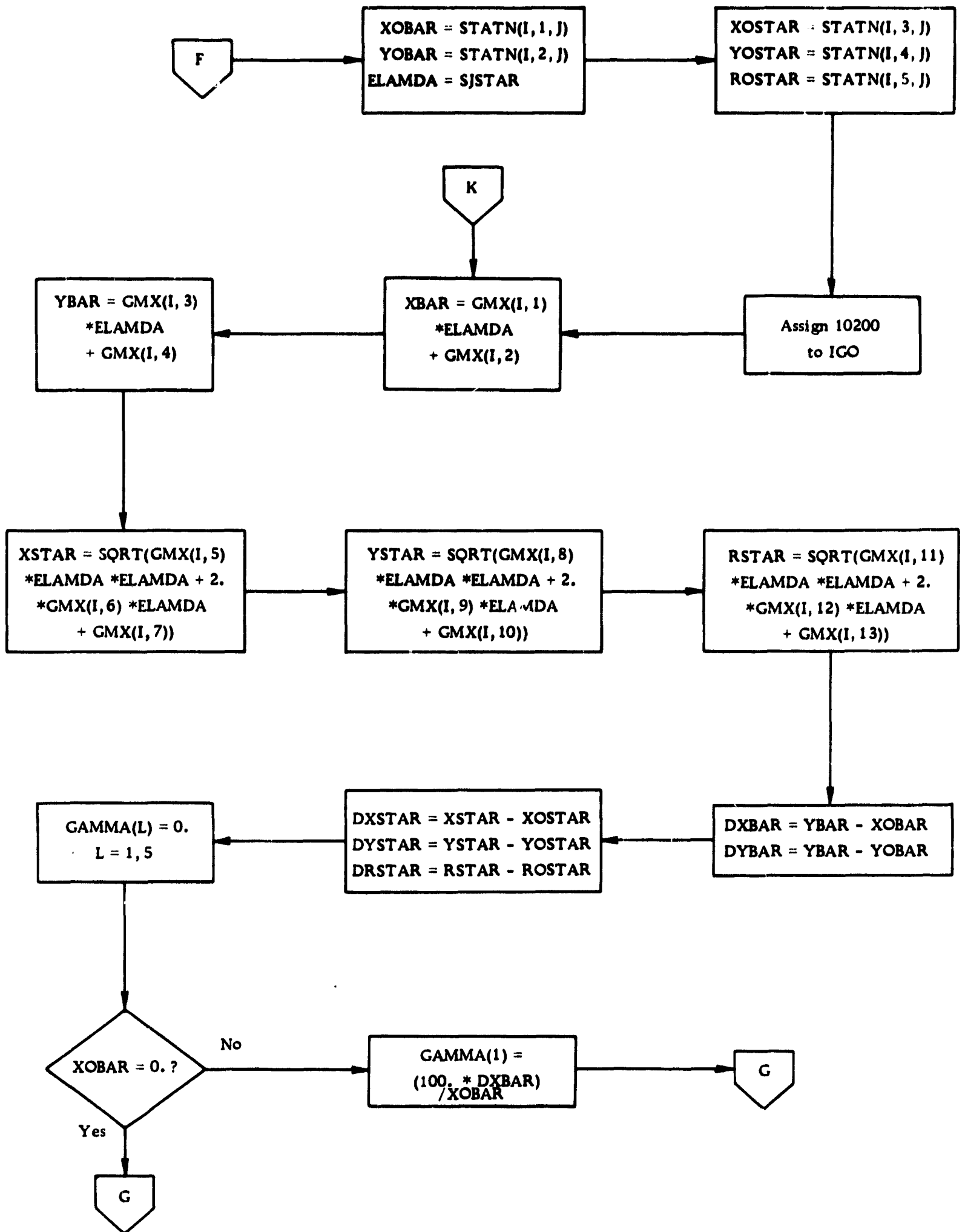
COMMON/DATA/
COMMON/TMPATT/
COMMON/PERTB/
COMMON/AZOBJ/
COMMON/PRNT/
SMPOP
MERID
PRTCTL

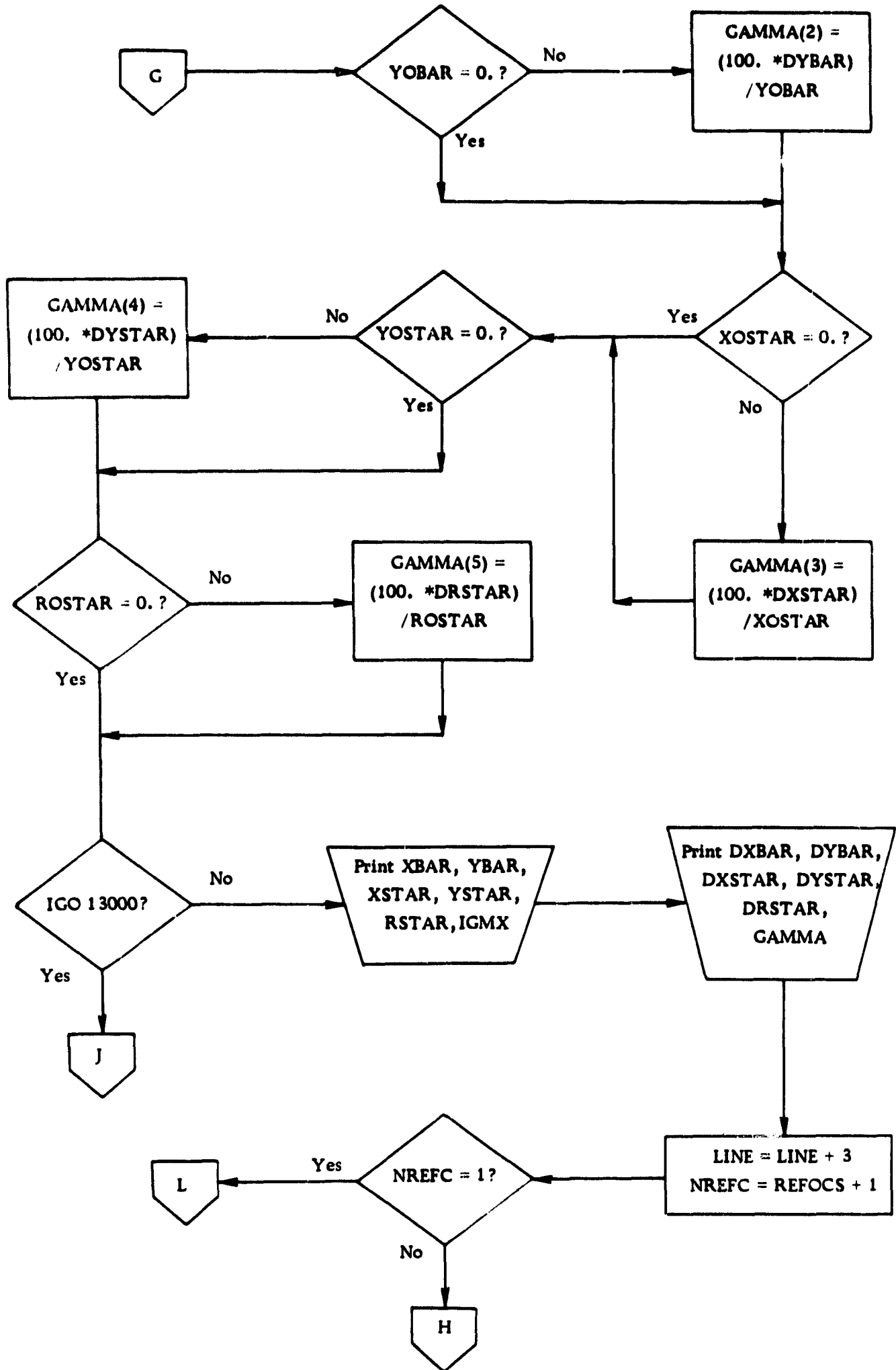


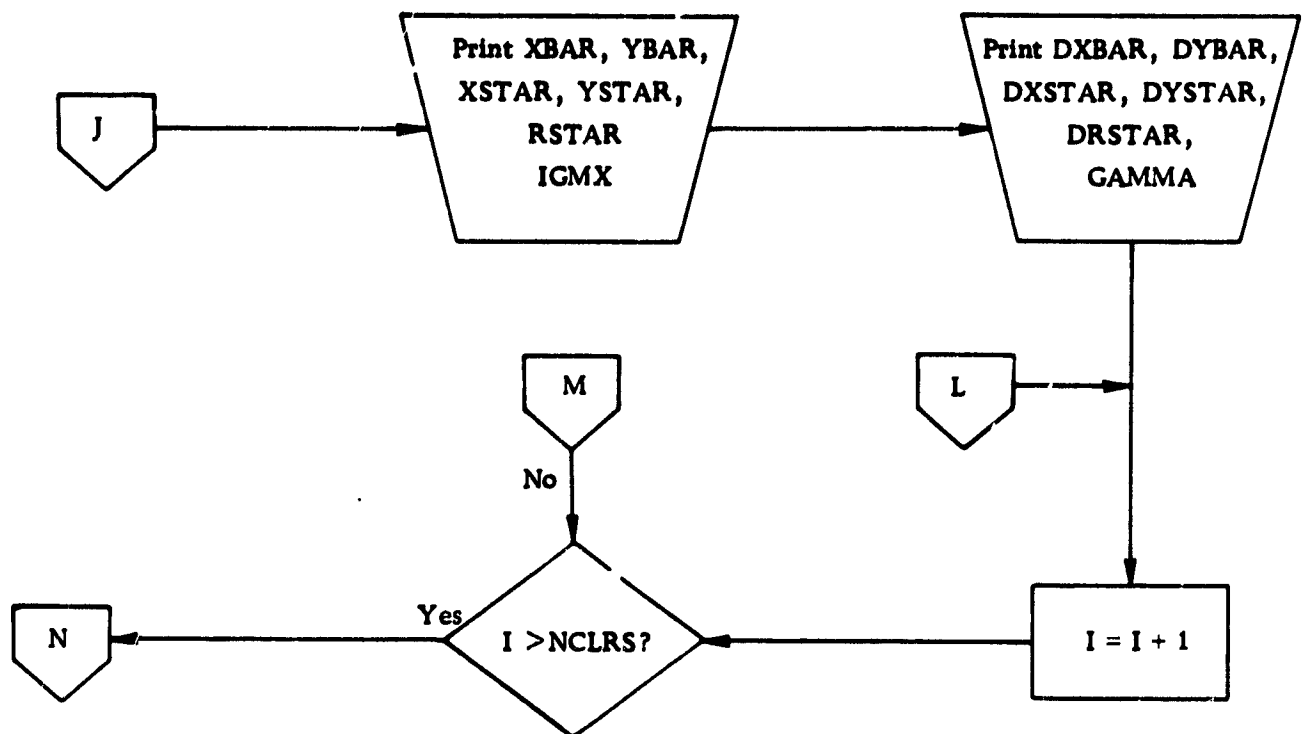
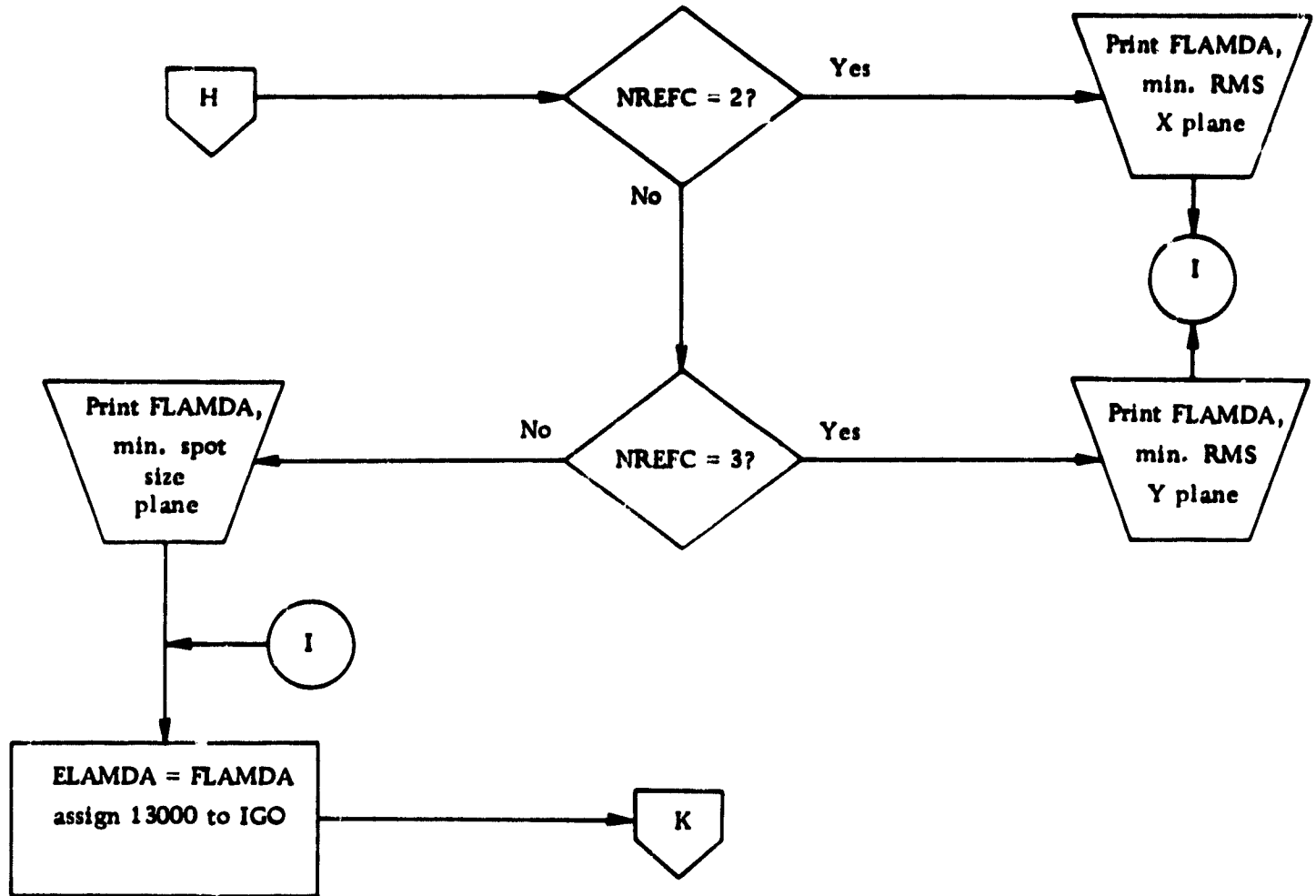


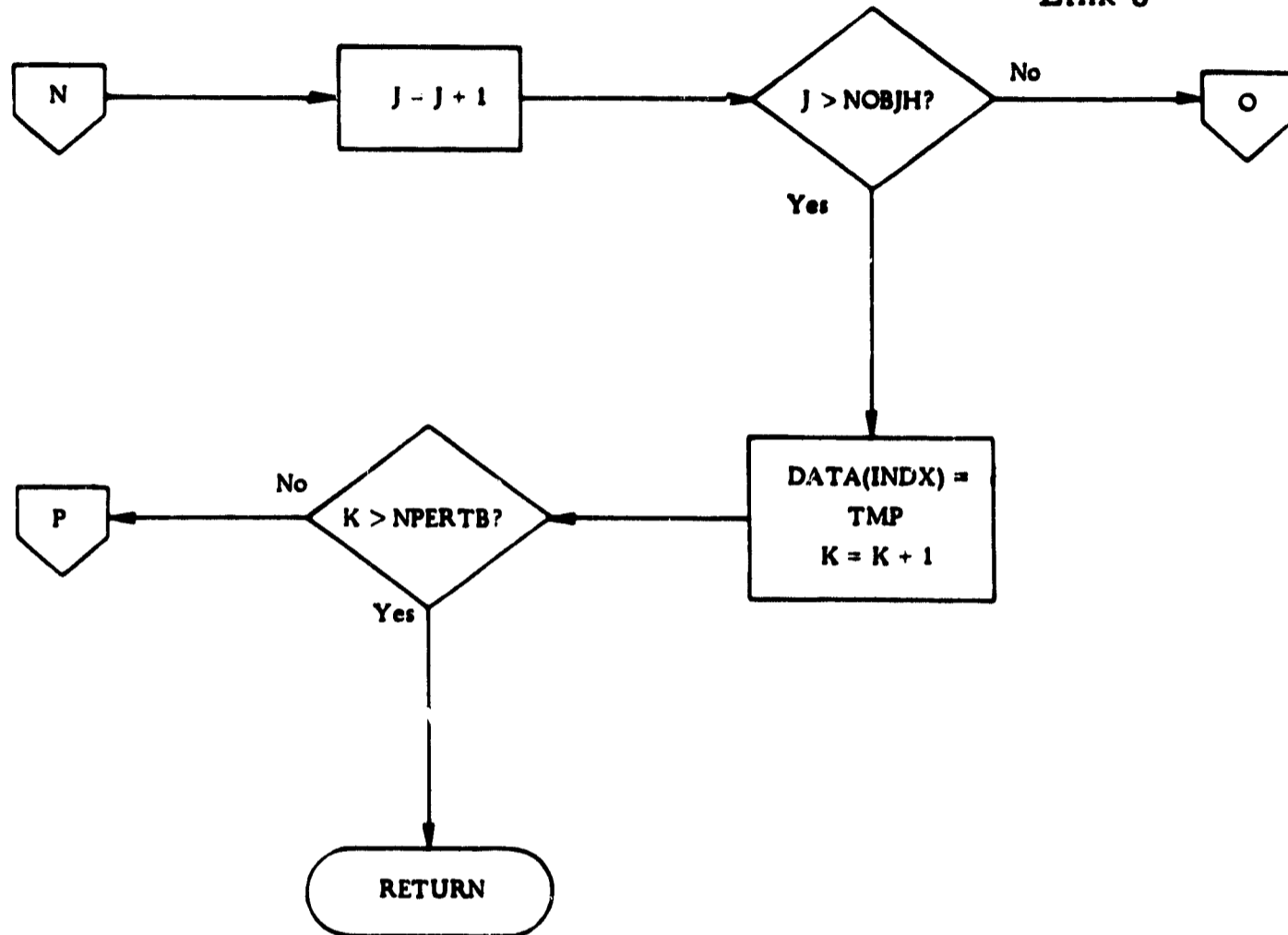












PROGRAM MODIFICATION

Program modification ranges from the simple task of altering a single source program statement to the complicated problem of implementing a new capability. Conversion of the program to some computing system other than the IBM 7094 involves, in general, a number of hardware dependent difficulties related to such considerations as word size and character size. The programming language (FORTRAN IV), the modularity concept (48 subroutines), and the program structure (overlay links) were selected and developed not only to simplify implementation and checkout but also to make modification less complicated.

Section 1

ADDING CAPABILITY

Consider first, the problem of adding some new capability to the program and assume for the moment that no new input parameters or symbols are involved.

An examination of figure 1 (Overlay Structure) reveals that there are two levels (origins) in the overlay arrangement. Those links (namely 5, 6, 7, and 8) which require the specified entrance pupil pattern to be traced must interface with subroutine SMPOP and hence start at origin 2. All other links start at origin 1.

If the capability to be added is a modification of, or an addition to, an existent capability the new subroutine(s) should be added to the appropriate link and interfaced as required. As an example, suppose that a new definition for the merit function has been devised based upon some criteria other than rms spot size. The error vector \vec{E} being defined differently is evaluated by a new subroutine, say ME2CTR. This subroutine would be included in link 5 and appropriate alterations made to deck CYCLEZ in order to optionally call ME2CTR instead of MEVCTR. Deck ME2CTR would contain any of the common blocks needed such as DATA and TMDESN.

Suppose then that the alteration represents a completely new capability to be included as a separate link (link 9) into the structure. If this link requires the entrance pupil pattern to be traced it must be started at origin 2 (parallel to 5, 6 etc.) and there must be one or more calls to subroutine SMPOP (link 4). If the entrance pupil pattern is not required then link 9 should start at origin 1, which gives it access to the print control routine, PRTCTL, and the ray trace routine, RAYTR.

Whenever it is necessary to change one of the labeled common blocks, in particular a change affecting size, all decks containing the labeled common in question must be altered in the same manner. Figures 2 and 3 of section 2. (PROGRAM STRUCTURE) will aid one in doing this.

Section 2 •

ADDING NEW INPUT

Each statement in the program is uniquely identified by its sequence number which appears in the right hand margin of the compilation listing and is punched in columns 73 through 80 of the corresponding source card. For example, statement RD401340 of READS is "L = I + 1".

Suppose there is a requirement to input the parameter γ whose FORTRAN name (symbol) is GAMMA. All input processing is handled by link 1, in particular, subroutine READS is responsible for reading and performing validity checks on all input parameters. It may be possible to include GAMMA in one of the existent NAMELIST statements such as GEOM, (RD007000, RD007100, and RD007200), in which case the modification is almost trivial.

If none of the existent NAMELIST statements are adequate a new one, such as NAMELIST/NEW/GAMMA, should be inserted into READS, as well as the appropriate read statement, READ (5, NEW). Since the presence of "NEW" data might be optional it may be necessary to introduce a sixth IFLAG into the INPUT NAMELIST data (see RD006500 through RD006510).

The normal method of communicating the value of GAMMA from READS to other program modules is by means of labeled common,

for example, inserting the statement, COMMON/NDATA/GAMMA, into READS, STAR10, FOLDP and all other modules which reference GAMMA is sufficient to effect the communication. *

A more complicated situation results when GAMMA is to be included in the DATA vector (see DT001100). In general, this occurs when either (or both) of the following circumstances prevail:

- a) GAMMA is to be included in the octal input/output, that is, included in the octal punch of the data region.
- b) GAMMA is to be an input symbol.

Assume that both conditions are to be satisfied and, further, that GAMMA is a real vector with dimension 20. The following modifications should be made to READS and all other decks containing COMMON/DATA/:

- a) Change the dimension of DATA from 3483 to 3503 (DT001100).
- b) Add the statements "DIMENSION GAMMA(20)" and "EQUIVALENCE (DATA(3484), GAMMA(1))".

In addition, READS must be altered as follows:

- a) Change the dimension of SYMBL4 to (2, 8). (RD002000).
- b) Change the value of NSYMB4 to 8. (RD002500).
- c) Add the statement "DATA SYMBL4 (1, 8), SYMBL4(2, 8) /6HGAMMA(, 3484/".

* It must appear in FOLDP in order not to be destroyed during overlay.

This is true of all interlink common variables.

- d) Add GAMMA to an existent or new NAMELIST statement.
- e) Change the upper limit of the DO statement, RD102600, from 3483 to 3503.
- f) Change the upper limit of the DO statement, RD105100, from 581 to 584.
- g) If necessary, add statements to read and check the validity of GAMMA.

It should be noted that any parameter vector or matrix which is to be included as a new input symbol must consist of exactly 5 characters be it integer or real.

The inclusion of GAMMA in the input DATA vector also makes it necessary to modify several other subroutines of link 1, in particular,

- a) In subroutine OPUNCH change the upper limit of the DO statement, OP005200, from 581 to 584.
- b) In subroutine REREAD change the upper limit of the DO statement, RR103600, from 580 to 583.
- c) Include a statement or statements in subroutine STAR9 or subroutine STAR10 to list GAMMA with appropriate headings using a format which is compatible with the current output scheme.

Section 3

CHANGING THE DIMENSION OF \vec{E}

One of the primary reasons for employing overlay is to maximize the amount of scratch storage available to the design link in order to permit the dimension of the error vector \vec{E} to be as large as possible. In the interests of speed and reliability the program is organized to perform all design computations in core which imposes a limitation on the "size" of the problem which can be handled.

Although M , the dimension of \vec{E} , can theoretically be as large as 16,893*, in practice it is restricted to being considerably less than this in order for the program to be operable on a 32K 7094.

If n represents the number of design variables where $1 \leq n \leq 10$, then the $n + 2$ vectors \vec{W} , \vec{E} , \vec{I}_1 , \vec{I}_2 , . . . , \vec{I}_n , each of dimension M , must be simultaneously available in core storage, requiring $M(n + 2)$ locations. These vectors are stored one behind the other starting in VCTSTR(1) which is located in TMDESN common. If NDIM is the dimension assigned to VCTSTR (currently 6000) then M must be such that:

$$M \leq \frac{\text{NDIM}}{n + 2} \quad (1)$$

* See volume I section 8 pages I-8-6 and I-8-7.

At the outset of a design computation option FOLDP calculates M and checks the inequality expressed by equation (1) using $NDIM = 6000$. If M is too large the run is aborted with an appropriate comment at which time the user can decrease M by selectively decreasing such things as the number of rays, the number of object points or the number of colors.

It is clear from equation (1) that M can take on larger values when n is small the maximum value occurring when $n = 1$ so that $M \leq (NDIM/3)$. Although vectors \vec{W} and \vec{E} must be preserved during a complete design cycle, the derivative vectors $\vec{I}_1, \vec{I}_2, \dots, \vec{I}_n$ are not needed following the generation of matrix I . The program utilizes this fact by permitting subroutine BNDCHK to employ the last 619 locations of VCTSTR as scratch storage, the assumption being that $NDIM$ is large enough so that the final 619 locations will not overlap vectors \vec{W} or \vec{E} . This will be the case provided $M \geq 619$ or

$$NDIM \geq 1857, \quad (2)$$

which is lower bound on acceptable values for $NDIM$ within the framework of the program. The upper bound on $NDIM$ is determined by the amount of storage available when links 0, 4, and 5 have been loaded.

Suppose that it is desirable to change the value of $NDIM$ from 6000 to some other legitimate value and call it $NDIM$. The following modifications are necessary:

- a) Change the dimension of VCTSTR from 6000 to $NDIM$ in TMDESN labeled common. (L7002000)
- b) Change 6000 to $NDIM$ in the IF statement S2005600 of subroutine STAR2.

- c) Change EQUIVALENCE statement items in subroutine BNDCHK from the form (A, VCTSTR (6000 - j)) where $0 < j < 618$ to the form (A, VCTSTR(NDIM - j)).

It should be noted that a re-design of the design link utilizing utility storage (disk or tape) would make it possible to consider more complex systems as (presumably) it would not be necessary to hold all of the M-dimensional vectors in storage at the same time.

Section 4

CONVERSION PROBLEMS

Conversion is used here in a general sense to include not only re-writing the program for a different computing system, but also the adaptation of the program to a 7094 whose monitor or peripheral hardware differs from the JPL 7044/7094 direct couple system. An example would be the re-design of subroutine CPLOT to generate plots on some device other than the SC4020.

4.1 INPUT/OUTPUT

There are a variety of problems which arise in connection with the input/output activity of the program. As an example, the octal format (6012) employed by subroutines OPUNCH and READS to punch and read "re-start" data cards is not defined on a hexadecimal computing system such as IBM 360. Some sort of hexadecimal format is required.

Subroutine REREAD which produces a card image list of the current case deck backspaces the input file which is not valid on some systems, e. g., when the input file resides in a card reader. A possible solution would be to have REREAD copy the card images onto a new file as they are read and make all subsequent input references to the new file.

The spot diagram point plotting which is handled by link 7 (STAR7 and GETLC) produces printer plots scaled to certain size (11" x 15") paper in such a manner that a grid of 62 lines and 102 characters is essentially square. This logic would have to be altered for a different paper size.

A major difficulty in going to another computer is the dependency of symbol manipulation logic on the 7094 word and character size. On the 7094 each word accomodates six 6 bit characters whereas (for example) on the 360 each word accomodates four 8 bit characters; consequently, a 6 character symbol requires one word on the 7094 and two words on the 360.

Subroutines VCNVRT and UNPBCD are employed in the conversion of vector and matrix subscripts from BCD to integer and they would be completely inadequate on a 360 and probably any other system. UNPBCD uses integer arithmetic to unpack a word of six characters into 6 words of 1 character each and thus depends upon word size, character size and the characteristics of integer arithmetic.

4.2 SIGNIFICANCE

There are two known areas of the program in which numerical results are rather sensitive to the basic accuracy of the computer.

The computation of vector \vec{I}_j^* involves differencing the components of two vectors which are nearly the same, that is,

$$\vec{I}_j = \vec{E} (\vec{U}_o + \Delta\vec{U}_j) - \vec{E} (\vec{U}_o)$$

where $|\Delta U_j|$ is small in order to have an accurate approximation of the partial derivative. On the 7094 with a basic accuracy of 8 decimal digits

* See volume I section 8.2.2.

only the first several digits of \vec{I}_j are meaningful, the remainder being round off noise. As a consequence, on a different computer or computing \vec{I}_j in a different manner the results may only agree (with FOLDP) to 2 or 3 significant digits. In some cases involving 4 or more design variables per iteration the design step, although valid, may be totally different from that produced by FOLDP. This situation was actually experienced during the checkout of FOLDP when results were compared with the original Lehman program.

During ray tracing and, in particular, when the program is attempting to find the point of intersection of a ray with an aspheric surface, the iteration algorithm involves the generation of a sequence of values $\theta_1, \theta_2, \dots, \theta_K$ where

$$\theta_K = \frac{Z'_K - Z_K}{Q_Z}$$

with Z_K lying on the ray and Z'_K on the aspheric.*

Convergence occurs when K is such that $|\theta_K| \leq \epsilon$. Again, because of round off noise in differencing two values which are nearly the same, only the first few digits of θ_K are meaningful. The θ sequence is assumed to be ordered, namely, $|\theta_K| > |\theta_{K+1}|$ for all K and a violation of this monotonicity property constitutes a miss. The value $\epsilon = 10^{-7}$ is currently being used on the 7094. When $\epsilon = 10^{-8}$ was tried the θ sequence occasionally became non-ordered giving rise to a "false"

* See volume I section 3.1.3

miss: the reason being that with ϵ so small the θ values were essentially nothing but noise and hence not necessarily ordered.

In order to prevent false misses due to round off noise choose ϵ to be equal to or greater than 10^{-S+1} where S is the number of significant decimal digits of accuracy of which the computer is capable, e. g., on the IBM 360 $\epsilon = 10^{-6}$ is satisfactory.

TR-67-700-10-2
Page III-1
Storage Map

STORAGE MAP


```

$JOB INF,NO10001,44903-0,77387,AM IC1,10,10000 FARNETT INFORMATICS      A 06*24**35*** 10/01/68
SIBJOB          MAP,NOGO
SIBLDR FOLDF          09/19/68      FOLD0000
SIBLDR ERRORZ        09/19/68      ERRO0000
SIBLDR GDFZZ         09/19/68      GDF0000
SIBLDR PRMSZZ        09/19/68      PRMS0000
SIBLDR PRTCZZ        09/19/68      PRTC0000
SIBLDR RAYTRZ        09/19/68      RAYT0000
$ORIGIN LINK1  ORG1      (READSZ,STAR9Z,STAR1Z,REREZZ,OPUNCH,MTFFN2)
SIBLDR OPUNCZ        09/19/68      OPUN0000
SIBLDR READSZ        09/19/68      READ0000
SIBLDR REREZZ        09/19/68      RERE0000
SIBLDR STAR1Z        09/19/68      STAR0000
SIBLDR STAR9Z        09/19/68      STAR0000
SIBLDR UNBCD.        09/19/68      UNBC0000
SIBLDR VCNVR.        09/19/68      VCNV0000
SIBLDR MTFPNZ        09/25/68      MTFP0000
$ORIGIN LINK2  ORG1      (PROF1Z,CELA1Z,RANGCZ,ENDPTZ,PFILEZ,CPL0TZ)
SINCLUDE          .ACF.          (LINK 2)
SINCLUDE          OUT.           (LINK 2)
SINCLUDE          GETJB.         (LINK 2)
SINCLUDE          SDINI.         (LINK 2)
SINCLUDE          SDNP.         (LINK 2)
SINCLUDE          SETBN         (LINK 2)
SINCLUDE          SINTR.         (LINK 2)
SINCLUDE          SLABL.         (LINK 2)
SINCLUDE          SFRT.         (LINK 2)
SINCLUDE          STERM.         (LINK 2)
SINCLUDE          STORE.         (LINK 2)
SINCLUDE          XXNDT.         (LINK 2)
SIBLDR ARPTSZ        09/20/68      ARPT0000
SIBLDR CELA1Z        09/20/68      CELA0000
SIBLDR CLOSSZ        09/20/68      CLOS0000
SIBLDR CPLOTZ        09/20/68      CPLO0000
SIBLDR CROSSZ        09/20/68      CROS0000
SIBLDR ENDPTZ        09/20/68      ENDP0000
SIBLDR EIRGCZ        09/20/68      EIRG0000
SIBLDR PFILEZ        09/20/68      PFIL0000
SIBLDR PROFLZ        09/20/68      PROF0000
SIBLDR RANGEZ        09/20/68      RANG0000
SIBLDR ZYCUTZ        09/20/68      ZYCU0000
$ORIGIN LINK3  ORG1      (TWR4Z,TWR5Z)
SIBLDR TWR4Z        09/20/68      TWR0000
SIBLDR TWR5Z        09/20/68      TWR0000
$ORIGIN LINK4  ORG1      (SMPOPZ,MERIDZ,LATTZZ)
SIBLDR LATTZZ        09/20/68      LATT0000
SIBLDR MERIDZ        09/20/68      MERID000
SIBLDR SMOFOPZ        09/20/68      SMOF0000
$ORIGIN LINK5  ORG2      (STAR2Z,CYCLEZ,BNDCHZ,JURYZ,GETRHZ,MWVCTZ,MEVCTZ)
SINCLUDE          MATINV         (LINK 5)
SIBLDR BNDCHZ        09/24/68      BND0000
SIBLDR CORCEZ        09/24/68      COR0000
SIBLDR CYCLEZ        09/24/68      CYCL0000
SIBLDR DOTFZ        09/24/68      DOTP0000
SIBLDR DFRATZ        09/24/68      DFRN0000
SIBLDR GETRHZ        09/24/68      GETR0000
SIBLDR INCR4Z        09/24/68      INCR0000
SIBLDR JURYZ        09/24/68      JURY0000

```

INF,NO10001,44903-0,77387,AM IC 18JOB L10HKC 7094 A 10/01/68 PAGE 3

SIDLCR MEVCTZ			09/24/68	MEV00000
SIDLCR M00DLZ			09/24/68	M0000000
SIDLCR MWVCTZ			09/24/68	MWV00000
SIDLCR STAR3Z			09/24/68	STAR0000
ORIGIN LINK6	OR02	(STAR3Z,PARAXZ)		
SIDLCR PARAXZ			09/24/68	PARA0000
SIDLCR STAR3Z			09/24/68	STAR0000
ORIGIN LINK7	OR02	(STAR7Z,GETLCZ)		
SIDLCR GETLCZ			09/24/68	GETL0000
SIDLCR STAR7Z			09/24/68	STAR0000
ORIGIN LINK8	OR02	(STAR0Z)		
SIDLCR STAR0Z			09/24/68	STAR0000

IBLDA INF,NO10001,44903-0,77387,AM IC IBJOB L1GMKC 7094 A 10/01/68 PAGE 2

OVERLAY ORIGIN CARDS AND ASSIGNED LINK NUMBERS

ORIGIN LINK1	ORG1	(READSZ,STAR9Z,STIS LINK	1, PARENT LINK IS	0
ORIGIN LINK2	ORG1	(PROF1Z,CELA1Z,RAIS LINK	2, PARENT LINK IS	0
	INCLUDE	.ACF.	(LINK 2)	
	INCLUDE	OUT.	(LINK 2)	
	INCLUDE	GETJB.	(LINK 2)	
	INCLUDE	SDINI.	(LINK 2)	
	INCLUDE	SDNPT	(LINK 2)	
	INCLUDE	SETBN	(LINK 2)	
	INCLUDE	SINTR.	(LINK 2)	
	INCLUDE	SLABL.	(LINK 2)	
	INCLUDE	SPRT.	(LINK 2)	
	INCLUDE	STERM.	(LINK 2)	
	INCLUDE	STORE.	(LINK 2)	
	INCLUDE	XXNDT.	(LINK 2)	
ORIGIN LINK3	ORG1	(TWR4Z,TWR5Z) IS LINK	3, PARENT LINK IS	0
ORIGIN LINK4	ORG1	(SMPOFZ,MERIDZ,LAIS LINK	4, PARENT LINK IS	0
ORIGIN LINK5	ORG2	(STAR2Z,CYCLEZ,BNIS LINK	5, PARENT LINK IS	4
	INCLUDE	MATINV	(LINK 5)	
ORIGIN LINK6	ORG2	(STAR3Z,PARAXZ) IS LINK	6, PARENT LINK IS	4
ORIGIN LINK7	ORG2	(STAR7Z,GETLCZ) IS LINK	7, PARENT LINK IS	4
ORIGIN LINK8	ORG2	(STAR8Z) IS LINK	8, PARENT LINK IS	4

IBLCR INF,NO10001,44933-D,77387,AM 1C IBJOB LISMKC 7594 A 10/01/68 PAGE 4

* MEMORY MAP *

SYSTEM 00000 THRU 02717
FILE BLOCK ORIGIN 02720
FILES 1. UNIT05
2. UNIT06
3. UNIT07
4. UNIT11
5. UNIT12
6. UNIT18

FILE LIST ORIGIN 03030
PRE-EXECUTION INITIALIZATION 03044
CALL ON OBJECT PROGRAM 03103
OBJECT PROGRAM 03110 THRU 72023

LINK	DECK	ORIGIN	CONTROL SECTIONS (/NAME/=NON D LENGTH, (LOC)=DELETED, *=NOT REFERENCED)					
D	FOLDF	03110	/DATA / 03112	EVEN 03111	/PERTB / 12004	EVEN 12003	/SYMBOLS/ 12176	
			/HEADER/ 14622	/FLOTG / 14624	EVEN 14627 15116 *		
	ERRORZ	15132	EVEN 15133	ERROR 15274	ERROR1 15304	ERROR2 15316	ERROR3 15330	
			ERRCR4 15342					
	GOOFZ	15352	EVEN 15353	GOOF 15366				
	PRMSZ	15400	/DATA / (03112)	EVEN 15401	PRMSUB 15541			
	PRTCZ	15566	/DATA / (13112)	/PRNT / 15570	EVEN 15567	/HEADER/ (14622)	PRTCTL 15662	
	RAYTR	15674	/AZOBJ / 15676	EVEN 15675	/DATA / (03112)	EVEN 15701	RAYTR 20050	
	.LINK	20167	/.LDT / 20167	/.LRECT/ 20200	/.LVEC / 20304			
	.LXCON	20336	.LXSTR 20336	.LXSTP 20341	.LXOUT 20407	.LXERR 20416	.LXCAL 20421 *	
			.LXRTRN 20421	IDEXIT 20421 *	.DBCLS 20604 *	.LXARG 20762	.LD 21005 *	
			.CLSE 21013	.LFBL 21014 *	.LUND 21015	.DFOUT 21016		
	.TODEF	21057	.DEFIN 21057	.ATTAC 21063 *	.CLOSE 21065	.OPEN 21067	.READ 21071	
			.WRITE 21073	.BSR 21103	.READR 21113	.RELES 21115 *	.LAREA 21126	
			.LFBLK 21144	.LTSX 21147 *	.RLMLD 21157 *	.AREA1 21162	.LUNBL 21170	
			.ENTRY 21174	.GOA 21231	.GO 21235	.DERR 21251	.NDPXL 21252	
			.COMXI 21254	.EX34 21276				
	.LOVRY	21303	.LOVRY (21303)	.LDT (20167)	.LRECT (20200)	.LVEC (20304)		
	.LXSL	22010	.LXSEL 22010	.LXSL1 22011	.LXST 22014 *	.LXOVL 22054 *	.LXMOD 22116 *	
			.LXIND 22142	.LXDIS 22145	.LXFLG 22146	.LTCH 22147		
	.FFTRP	22155	.FFFT. 22155 *	.FWFL. 22331 *	.FFOUT 22342	.FFARG 22350	/.COUNT/ 22352	
			OVFLOW 22417 *	FFTRP 22422				
	.ERAS.	22461	E.1 22461	E.2 22462	E.3 22463	E.4 22464		
	.XCC.	22465	CC.1 22465	CC.2 22466	CC.3 22467	CC.4 22470		
	XIT	22471	EXIT 22471	.EXIT. 22471				
	FXEM	22472	.FXEM. 22472	FXEM (22472)	.FXOUT 23041	.FXARG 23047	/.OPTW./ 23123	
			/.OPDS./ 23127 *					
	FOUT	23137	.FOUT. 23137					
	FCNV	23172	.FCON. 23172	.FCNV. 23215	.ENDFS 23227	.CNVSW 23231	.FDX1 23235	
			.FDX2 23236	.DBC 23240	.DBC10 23404	.DBC20 23432	.DDSW 23450	
			.DDFIX 23454	.FIXSW 23455	.DDBC 23522	.DDRS1 23762	.DDRS2 23764	
			.D1 23767	.D2 23771	.FERR2 24056 *	.ANPT 24112	.ONPT 24122	
			.LNTP 24215	.AOUT 24264	.DFLT 24302	.FLT 24436	.EXFON 24524	
			.DEXFN 24526	.FXD 24527	.MOUT 24702	.INTG 24752	.LOUT 25100	
			.ODUT 25117	.XCF 25150	ASTERN 25252 *	EVEN 25565	.TEST 25712	
			.KOUNT 25732	.LIST 25735	.DONE 25746	.OUTDF 26012	.BUF 26042	
			.OSTO 26043	.WIDTH 26044	.GAIN 26045	.GAIN1 26046	.FBDBF 26056	
			EVEN 26067	.DDDFL 26103	.DDFLG 26104	.MOD 26111	.FEX 26112	
			.FEXP 26113	.DIG 26114				

F105	26134	.F105.	26134	.FCMT	26237	.FBLT.	26335	.FBDT.	26355	.FRLB.	26401
		.FRLR.	(26401)	.FWLR.	26445	.FWLR.	(26445)	.FD1BF	26505	.FR1TE	26577
F105	26605	.F105.	26605	.FSEL.	26745	.FILR.	26751	.FR1B.	26760	.FR1D.	26765
		.FILL.	26770	.FCLS	26772 *	.FOPN	26776 *	REOF	27002 *	.TOUT.	27145
		.REED	27153 *	.BIN	27154 *	.FCT	27155	.FCKSZ	27157		
F10H	27241	.F10H1	27241	.F10H.	27271	.FF1L.	30054	.CODE.	30075	.FR1N.	30102
FARD	30300	.ENC0.	30300	.DECD.	30302						
FWRD	30666	.FWRD.	30666	.4020.	30715						
FWRB	30726	.FWRB.	30726								
FRDD	31001	.FRDD.	31001								
FRDB	31027	.FRDB.	31027								
FPUN	31053	.FPUN.	31053								
FRDU	31270	.FRDU.	31270								
FOUT	31346	SETPL	31346	RESPL	31350	.FOUT.	31351	.OUT2.	31456 *	.OUT3.	31501
FNOUT	31505	NMOUT	31505 *	NMIN	31507 *	.NMSW.	31511				
UND5	31512	.UND5.	31512								
UND6	31513	.UND6.	31513	.BUFSZ	31514 *						
UND7	31517	.UND7.	31517								
UN11	31520	.UN11.	31520								
UN12	31521	.UN12.	31521								
UN18	31522	.UN18.	31522								
F10U	31523	.F10U.	31523	.CTU10	32235	.NMLST	32260	.NAME.	33716	.INTAP	33717
FLOG	34347	ALOG10	34347	ALOG	34350 *						
FXPFF	34553	EXP	34553	SIN	-34675						
F3CN	34674	COS	34674								
F3QR	35070	BQRT	35070								
FXP1	35143	.XP1.	35143								
FXP2	35257	.XP2.	35257								
FBST	35375	.FBST.	35375								
FEFT	35612	.FEFT.	35612								
FRWT	35712	.FRWT.	35712								
FSLD1	36017	.FSL1.	36035	.FSD1.	36043 *						
FSLB1	36054	.FBL1.	36072	.FBD1.	36100 *						
FSLI	36112	.SL1.	36112	.SL11.	36117	.SD1.	36125	.SD11.	36133		
FSLDO	36146	.FSL0.	36164	.FSD0.	36172 *						
FSLBO	36203	.FBLO.	36221	.FBD0.	36227 *						
FSLO	36241	.SLO.	36241	.SLO2.	36247	.SDO.	36254	.SDO2.	36263		
FRET	36275	.FRET.	36275								
IT	36376	TICK	36376 *	STOPW	36401 *	ALARM	36401 *	CLOCK	36404	FIVE	36546 *
SCALEK	36564	SCALEK	(36564)	EVEN	37225						
ERR113	37316	ERR113	(37316)								
.10CS	37351	.L(D)	37351	.MONSW	37371	.TEOR	37440	.DEF1.	37520	.JOINX	37564 *
		.CLOS.	37603	.ATTC.	37616	.SH1	40030 *	.SH9	40072 *	.OPEN.	40113
		.OP4	40141 *	.OP7	40172 *	.OP9.2	40206 *	.RLSE.	40260	.RER2.	40260
		.READ.	40261	.RER1.	40304	.WRIT.	40306	.MNT1A	40476 *	.EOFEX	40557 *
		.FEE1T	40627	.GTIOX	40650	.RW7	40766 *	.RE7	41411 *	.ENDTR	42052
		.SEL59	42054 *	.BSR.	42473	.EOTOF	42620	.ETOF3	42626 *	.SWITC	42655
		.TCHEX	43162	.BASIO	43165 *						
.10CSM	43170										
OPUNCF	43170	/DATA /	(03112)	EVEN	43171	OPUNCH	43354				
READS2	43372	/DATA /	(03112)	/SYMBLS/	(12176)	/PRNT /	(15570)	/PERTB /	(12004)	/PLOT /	(14624)
		EVEN	43373	READS	50515						
REN22	50542	EVEN	50543	REREAD	51176						
STAR12	51214	/DATA /	(03112)	/SYMBLS/	(12176)	/PRNT /	(15570)	/PERTB /	(12004)	/PLOT /	(14624)
		EVEN	51215	STAR10	53260						

IBLDR INF,NO10001,44903-D,77387,AM IC IBJOB L10MKC 7094 A 10/01/68 PAGE 6

STAR9Z	53303	/DATA / (03112)	/SYMBOLS/ (12176)	/PRNT / (15570)	STAR9	55720		
UNBCD.	55741	UNPCCD 56023						
VCHVR.	56046	EVEN 56047	VCHVRT 56360	MCHVRT 56375				
MTFFNZ	56416	/DATA / (03112)	EVEN 56417	MTFFN 62307				
2 .ACF.	43170	EVEN 43171	ADV 43272	CAMERA 43275	FORM	43302 *	EXIMG	43305 *
		RDIMG 43310 *						
OUT.	43313	OUT 43313	EVEN 43423					
GETJD.	43432	GETID 43432						
SDINI.	43515	SDINIT 43515						
SDNFT	43602	SDNFUT 43604						
SETBN	45227	RESBN 45227						
SINTR.	45230	SINTRF 45230	EVEN 45261	EVEN 45265				
SLABL.	45272	SLABEL 45272						
SPRT.	45365	SPRNTA 45365	SPRINT 45372 *					
STERM.	46167	STERM 46167						
STORE.	46227	STOREP 46227	EVEN 46313					
XXNDT.	46321	XXNDT 46462						
ARPTSZ	46556	/DATA / (03112)	EVEN 46557	ARPTS 46706				
CELAIZ	46750	/DATA / (03112)	/PRNT / (15570)	/ELMAIR/ 46752	EVEN	46751	CELAIR	52371
CLOSSZ	52425	/DATA / (03112)	/ELMAIR/ (46752)	CLOSS 53006				
CPLOTZ	53114	/DATA / (03112)	/ELMAIR/ (46752)	/CODRNG/ 53116	EVEN	53115	/PLOTZ / (14624)	
		CPLOT 57432						
CROSSZ	57472	/DATA / (03112)	/ELMAIR/ (46752)	EVEN 57473	CROSS	60252		
ENDPTZ	60345	/DATA / (03112)	/ELMAIR/ (46752)	/CODRNG/ (53116)	ENDPTS	62137		
EIRG CZ	62321	/ELMAIR/ (46752)	/CODRNG/ (53116)	EIRG CD 62604				
FFILEZ	62622	/DATA / (03112)	/ELMAIR/ (46752)	/CODRNG/ (53116)	/FLOTZ / (14624)		/AZOBJ / (15676)	
		EVEN 62623	FFILE 64043					
PROFLZ	64134	/ELMAIR/ (46752)	/CODRNG/ (53116)	/FLOTZ / (14624)	/PRNT / (15570)		EVEN	64135
		PROFIL 66521						
RANGEZ	66544	/DATA / (03112)	/ELMAIR/ (46752)	/CODRNG/ (53116)	EVEN	66545	RANGE	67320
ZYCUTZ	67364	/DATA / (03112)	/CODRNG/ (53116)	EVEN 67365	ZYCUT	67617		
3 TWRN4Z	43170	/DATA / (03112)	/AZOBJ / (15676)	/PRNT / (15570)	/TMTWIN/ 43172		EVEN	43171
		TWRN4 46431						
TWRN5Z	46454	/DATA / (03112)	/AZOBJ / (15676)	/PRNT / (15570)	/TMTWIN/ (43172)		EVEN	46455
		TWRN5 47272						
4 LATTZZ	43170	/DATA / (03112)	EVEN 43171	LATT 43267				
MERIDZ	43315	/DATA / (03112)	/AZOBJ / (15676)	MERID 43462				
SMFOPZ	43504	/DATA / (03112)	/AZOBJ / (15676)	/TMPATT/ 43506	EVEN	43505	EVEN	44357
		SMFOP 45274						
5 MATINV	45363	MATINV (45363)	EVEN 46257					
BNDCHZ	46314	/DATA / (03112)	/PRNT / (15570)	/AZOBJ / (15676)	/TMDESN/ 46316		EVEN	46315
		EVEN 62315	BNDCHK 63376					
CORDEZ	63440	EVEN 63441	CORDER 63510					
CYCLEZ	63543	/DATA / (03112)	/PRNT / (15570)	/TMDESN/ (46316)	CYCLE	65151		
DOTPZ	65203	/TMDESN/ (46316)	DOTP 65226					
DPRNTZ	65256	/DATA / (03112)	/PRNT / (15570)	/TMDESN/ (46316)	EVEN	65257	DPRNT	65656
		EPRNT 65661						
GETRHZ	65664	/DATA / (03112)	/AZOBJ / (15676)	EVEN 65665	GETRHO	65747		
INCRMZ	65765	/DATA / (03112)	INCRM 66240					
JURYZ	66263	/DATA / (03112)	/TMDESN/ (46316)	JURY 66475	JUDGE	66500		
MEVCTZ	66503	/DATA / (03112)	/TMPATT/ (43506)	/AZOBJ / (15676)	/TMDESN/ (46316)		MEVCTR	70620
MODDLZ	70672	/DATA / (03112)	EVEN 70673	MODDL 70754				

IBLDR INF,NO10001,44903-D,77387,AM 1C IBJOB L10MKC 7094 A 10/01/68 PAGE 7

MWVCTZ	70770	/DATA	/(03112)	/TMDESH/(46316)	EVEN	70771	MWVCTR	71301			
STAR2Z	71322	/DATA	/(03112)	/FRNT/(15570)	/TMDESH/(46316)		EVEN	71323	STAR2	72003	
6	PARAXZ	45363	/DATA	/(03112)	/AZOBJ/(15676)	PARAX	45540				
	STAR3Z	45567	/DATA	/(03112)	/TMPATT/(43506)	/AZOBJ/(15676)	/FRNT	/(15570)	STAR3	51400	
7	GETLCZ	45363	GETLC	45476							
	STAR7Z	45535	/DATA	/(03112)	/TMPATT/(43506)	/FRNT	/(15570)	/AZOBJ	/(15676)	STAR7	52062
8	STAR8Z	45363	/DATA	/(03112)	/TMPATT/(43506)	/FRNT	/(15570)	/PER1B	/(12004)	/AZOBJ	/(15676)
			STAR8	51104							

I/O BUFFERS 72024 THRU 77772

UNUSED CORE 77773 THRU 77777

292 LINES OUTPUT.

IBSYS
 RETURNING TO IBSYS.

TR-67-700-10-2
Page III-0-1
Program Listing
Link 0

PROGRAM LISTING

Section 0

LINK 0

TR-67-700-10-2

Page III-0-2

Program Listing

Link 0

Page Revised 10/1/68

INF,NO10001,44903-0,77387,AM IC

18JOB L10KRF 7094 A 02/19/65

10

BTITLE

FORTRAN OPTICAL LENS DESIGN PROGRAM

BTITLE FOLDF

M94,XR7

LINK 0 (MAIN PROGRAM)

MNO01000

FORTRAN OPTICAL LENS DESIGN PROGRAM
 FOLDF - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE 1

```

C
C *** DESCRIPTION ***
C FOLDF PERFORMS AUTOMATIC DESIGNING OF OPTICAL SYSTEMS BY A CONTROLLED LEAST SQUARES PROCEDURE. THE MAIN PROGRAM IS A CONTROLLED LEAST SQUARES PROCEDURE. THE MAIN PROGRAM IS A CONTROLLED LEAST SQUARES PROCEDURE. THE MAIN PROGRAM IS A CONTROLLED LEAST SQUARES PROCEDURE.
C ROUTINE FOR PROCESSING A SINGLE CASE, WHICH INVOLVES READING INPUT DATA AND EXECUTING THE SET OF USER SPECIFIED OPTIONS.
C
C THE FOLLOWING OPTIONS ARE AVAILABLE ...
C
C 0 TERMINATE PROCESSING
C 1 INPUT DATA ON OCTAL CARDS
C 2 DESIGN COMPUTATIONS
C 3 SPOT DIAGRAM COMPUTATIONS
C 4 TWINRAY DIAGNOSTIC FROM MAXIMUM OBJECT HEIGHT
C 5 TWINRAY DIAGNOSTIC FROM ZERO OBJECT HEIGHT
C 6 FINCH OCTAL CARDS
C 7 PLOT SPOT DIAGRAMS
C 8 SENSITIVITY COMPUTATIONS
C 9 PRINT GEOMETRY DATA
C 10 PRINT ALL INPUT DATA
C 11 PROFILE PLOT
C 12 FINCH CARDS FOR FAGOS MTF
C
C COMMON / DATA / CONTROL, CONTEL(10), TITLE(12), DATE(3), FUNCID,
1 DATA(3485)
DIMENSION WOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7),
1 WCLRS(6), NDSGN(4), NDSGV(11), LATTC(3,26), BOUNDS(3,100),
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATTC(3,26),
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100)
EQUIVALENCE ( DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3),
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),
2 NSUBT ), ( DATA(7), NSUDF ), ( DATA(8), NIPLN ), ( DATA(9),
3 LMODE ), ( DATA(10), NSPLN ), ( DATA(11), NOBJH ), ( DATA(12),
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15),
5 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18),
6 ZETA ), ( DATA(19), HEXFF ), ( DATA(20), DEXFF ), ( DATA(21),
7 WEXFF ), ( DATA(22), DLPLN ), ( DATA(23), OMGA2 ), ( DATA(24),
8 OMGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27),
9 PSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30),
A DUMIN ), ( DATA(31), ODIST ), ( DATA(32), HD ), ( DATA(33),
B DELH ), ( DATA(34), SYSMX ), ( DATA(35), WXDIR ), ( DATA(36),
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(39),
D NDSGV(1) ), ( DATA(40), WOBJH(1) ), ( DATA(41), WCLRH(1) ),
E ( DATA(42), WIMHT(1) ), ( DATA(43), EIMHT(1) ), ( DATA(44),
F CIMPL(1) ), ( DATA(45), WCLRS(1) ),
G ( DATA(46), LATTC(1,1), ILATTC(1,1) ), ( DATA(47), SURFC(1,1),
H ISURFC(1,1), ( DATA(48), DESGN(1,1), IDESGN(1,1) ),
I ( DATA(49), SUBST(1), ISUBST(1) ), ( DATA(50), BOUNDS(1,1),
J IBNDS(1,1), ( DATA(51), NCOND ),
EQUIVALENCE ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )
INTEGER CONTROL, AFLAG, DATE, ATRGGR
REAL LATTC
C
C THIS BLOCK OF COMMON CONTAINS PERTURBATION VALUES
COMMON / PERTS / PERTS(30,4), NPPTS, REFOCS
  
```

0001010
 0001020
 0001030
 0001040
 0001050
 0001060
 0001070
 0001080
 0001090
 0001100
 0001110
 0001120
 0001130
 0001140
 0001150
 0001160
 0001170
 0001180
 0001190
 0001200
 0001210
 0001215
 0001220
 0001000
 0001100
 0002000
 0002100
 0002200
 0002300
 0001000
 0001100
 0001200
 0001300
 0001400
 0001500
 0001600
 0001700
 0001800
 0001900
 0002000
 0002100
 0002200
 0002300
 0002400
 0002500
 0002600
 0002700
 0002800
 0002900
 0003000
 0001000
 0002000
 0001000
 0002000

FORTRAN OPTICAL LENS DESIGN PROGRAM
FOLDF - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE 2

	EQUIVALENCE (PERTD(1,1), IPERTD(1,1))	FB003000	
	DIMENSION IPERTD(30,4)	FB004000	
	INTEGER REFOCS	FB005000	
C			
C	THIS BLOCK OF COMMON CONTAINS THE SYMBOLS USED IN DESIGN	SY001000	
C	AND SUBSTITUTION - UTILIZED FOR PRINT OUT	SY002000	
	COMMON / SYMCLS / SDESH(12,50), SUBSYM(2,250), BDYSYM(2,100)	SY003000	
	INTEGER SDESN, SUBSYM, BDYSYM	SY004000	
C			
C	COMMON / HEADER / JTEMP, JSUBST	MN100500	
C			
C	COMMON / PLOT / YMAXX, DZMIN, NPTS	PC001000	
C			
	CALL FPTRP(0,0)	MN100700	2
1000	CALL REREAD	MN101000	4
	CALL READS	MN101100	
C			
C	PROCESS PROGRAM CONTROL COMMANDS	MN101200	6
	DO 20000 I = 1, NCNTRL	MN102000	
	IF(CNTRL(I) .EQ. 0) GO TO 32000	MN102200	
C			
C	TEST FOR LEGAL PROGRAM CONTROL COMMAND	MN103000	
	IF(CNTRL(I) .GE. 1 .OR. CNTRL(I) .LE. 12) GO TO 1500	MN103200	
	CALL ERROR2(36H ILLEGAL PROGRAM CONTROL COMMAND , CNTRL(I))	MN103500	21
	GO TO 20000	MN103600	
1500	JTEMP = CNTRL(I)	MN104000	
	JSUBST = 0	MN104100	
	GO TO (20000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000,	MN104200	
	1 10000, 11000, 12000), JTEMP	MN104400	
C			
C	DESIGN CALCULATION	MN105000	
2000	CALL STAR2	MN105200	20
	JSUBST = JSUBST + 1	MN105300	
	GO TO 20000	MN105400	
C			
C	OPTICS DIAGNOSTIC CALCULATION	MN106000	
3000	CALL STAR3	MN106200	32
	JSUBST = JSUBST + 1	MN106300	
	CALL PRMSUD(\$3000)	MN106400	35
	GO TO 20000	MN106600	
C			
C	TWIN-RAY DIAGNOSTIC CALCULATION	MN107000	
4000	CALL TWRNY4	MN107200	39
	JSUBST = JSUBST + 1	MN107300	
	CALL PRMSUD(\$4000)	MN107400	42
	GO TO 20000	MN107600	
C			
C	TWIN-RAY DIAGNOSTIC CALCULATION	MN108000	
5000	CALL TWRNY5	MN108400	46
	JSUBST = JSUBST + 1	MN108500	
	CALL PRMSUD(\$5000)	MN108600	49
	GO TO 20000	MN109000	
C			
C	PUNCH OPTAL CARDS	MN201000	
6000	CALL OPUNCH	MN201200	53
	GO TO 20000	MN201400	

FORTRAN OPTICAL LENS DESIGN PROGRAM
 FOLDP - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

C			
C	SPOT DIAGRAM PLOT	MN202000	
	7000 CALL STAR7	MN202200	56
	JSUBST = JSUBST + 1	MN202300	
	CALL PRMSUB(87000)	MN202400	59
	GO TO 20000	MN202600	
C			
C	PERTURBATION CALCULATION	MN203000	
	8000 CALL STAR8	MN203200	63
	JSUBST = JSUBST + 1	MN203300	
	CALL PRMSUB(88000)	MN203400	66
	GO TO 20000	MN203600	
C			
C	PRINT LENS PRESCRIPTION	MN204000	
	9000 CALL STAR9	MN204200	70
	GO TO 20000	MN204400	
C			
C	PRINT DESIGN INFORMATION	MN205000	
	10000 CALL STAR9	MN205100	73
	CALL STAR10	MN205200	75
	GO TO 20000	MN205400	
C			
C	CROSS SECTION PLOT	MN206000	
	11000 CALL PROFIL	MN206200	78
	JSUBST = JSUBST + 1	MN206400	
	CALL PRMSUB(91000)	MN206600	81
	GO TO 20000	MN206800	
C			
C	PUNCH MTF CARDS	MN206850	
	12000 CALL MTFPN	MN206900	
	20000 CONTINUE	MN206920	85
		MN402000	
C			
C	RETURN TO READS SUBROUTINE TO PROCESS NEXT DATA SET	MN501000	
	GO TO 1000	MN502000	
	32000 CALL EXIT	MN503000	91
	STOP	MN504000	
	END	MN505000	

FORTRAN OPTICAL LENS DESIGN PROGRAM
 FOLDF

09/19/68

PAGE 4

STORAGE MAP

MAIN PROGRAM
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	FUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIFLN	00042	I	IMODE	00043	I
NSFLN	00044	I	NDBJH	00045	I	NSURP	00046	I
AFLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXFF	00055	R	DEXFF	00056	R	WEXFF	00057	R
DLFLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	FSCAL	00065	R
OMGAF	00060	R	SFFEA	00067	R	DUMIN	00070	R
CO1ST	00071	R	WD	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WDBJH	00117	R	WCLRH	00126	R	WIMHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I	ATRCCR	06666	I
GAUSS	06667	R						

COMMON BLOCK			PERTB	ORIGIN	06672	LENGTH	00172	
PERTB	00000	R	NPERTB	00170	I	REFOCS	00171	I
FERTB	00000	I						

COMMON BLOCK			SYMBLS	ORIGIN	07064	LENGTH	02424	
SDSN	00000	I	SUBSYM	01130	I	BDYSYM	02114	I

COMMON BLOCK			HEADER	ORIGIN	11510	LENGTH	00002	
JTEMP	00000	I	JSUBST	00001	I			

COMMON BLOCK			PLOTG	ORIGIN	11512	LENGTH	00003	
YMAXX	00000	R	DZMIN	00001	R	NPTS	00002	I

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
I	11515	I						

ENTRY POINTS

FORTRAN OPTICAL LENS DESIGN PROGRAM
 FOLDP

09/19/68

PAGE 5

STORAGE MAP

SUBROUTINES CALLED

FPTRP	SECTION	14
ERROR2	SECTION	17
PRMSUB	SECTION	20
OPUNCH	SECTION	23
STAR9	SECTION	26
MTFPN	SECTION	29
.F.MEM.	SECTION	32

REREAD	SECTION	15
STAR2	SECTION	18
TWNR4	SECTION	21
STAR7	SECTION	24
STAR10	SECTION	27
EXIT	SECTION	30
SYSLOC	SECTION	33
EFN	IFN	CORRESPONDENCE

READS	SECTION
STAR3	SECTION
TWNR5	SECTION
STAR8	SECTION
PROFIL	SECTION
.EXIT.	SECTION

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	3A	11541	20000	86A	11762	32000	90A	11763
1500	23A	11603	2000	27A	11627	3000	31A	11630
4000	38A	11651	5000	45A	11664	6000	52A	11670
7000	55A	11703	8000	62A	11716	9000	69A	11730
10000	72A	11735	11000	77A	11744	12000	84A	11750

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12017.

TR-67-700-10-2
Page III-0-8
Program Listing
Link 0

INF.N010001.44000-0.77307.AM TC IDJOB V13NJO 7094 A 10/31/67 PAGE 3

80
81 TITLE ERROR MESSAGE PRINTER
82B7 TC ERRORZ M04.XR7 LINK 0 (ERROR,ERROR1,....,ERROR4) ER001000

ERROR MESSAGE PRINTER
 ERRORZ - EPN SOURCE STATEMENT - IPN(8) -

10/31/67

PAGE 1

SUBROUTINE ERROR(ETEXT)	ER001100	
DIMENSION ETEXT(8)	ER001200	
C		
C THIS SUBROUTINE IS USED TO PRINT OUT AN ERROR MESSAGE AND	ER002000	
C PRINT ONE VALUE UNDER SEVERAL FORMAT TYPES	ER002100	
C ERROR - TEXT ONLY	ER002200	
C ERROR1 - TEXT AND BCD	ER002300	
C ERROR2 - TEXT AND INTEGER	ER002400	
C ERROR3 - TEXT AND REAL	ER002500	
C ERROR4 - TEXT WITH RETURN	ER002600	
C		
WRITE(6.32100) ETEXT	ER102000	1
32100 FORMAT(1HD, 6A6)	ER102100	
GO TO 32000	ER-TEMP1	
C CALL GOOF	ER102200	
ENTRY ERROR1(ETEXT, AVAR)	ER103000	
WRITE(6.32110) ETEXT, AVAR	ER103100	4
32110 FORMAT(1HD, 6A6, 2X, A6)	ER103200	
GO TO 32000	ER-TEMP2	
C CALL GOOF	ER103300	
ENTRY ERROR2(ETEXT, IVAR)	ER104000	
WRITE(6.32120) ETEXT, IVAR	ER104100	7
32120 FORMAT(1HD, 6A6, 2X, I6)	ER104200	
GO TO 32000	ER-TEMP3	
C CALL GOOF	ER104300	
ENTRY ERROR3(ETEXT, EVAR)	ER105000	
WRITE(6.32130) ETEXT, EVAR	ER105100	10
32130 FORMAT(1HD, 6A6, 2X, E14.6)	ER105200	
C CALL GOOF	ER105300	
GO TO 32000	ER-TEMP4	
ENTRY ERROR4(ETEXT)	ER106000	
WRITE(6.32100)	ER106100	13
32000 RETURN	ER201000	
END	ER202000	

ERROR MESSAGE PRINTED
 ERROR2

10/31/67

PAGE 2

STORAGE MAP

SUBROUTINE ERROR
 ENTRY POINTS

ERROR SECTION 3
 ERRORS SECTION 6

ERROR1 SECTION 4
 ERRORS SECTION 7

ERROR2 SECTION 9

SUBROUTINES CALLED

.FWRD. SECTION 8
 .FFIL. SECTION 11

.FSLO. SECTION 9
 .PCNV. SECTION 12
 EFN IFN CORRESPONDENCE

.UNDB. SECTION 10
 SYSLOC SECTION 13

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32100	FORMAT	00011	32000	14A	00123	32110	FORMAT	00013
32120	FORMAT	00017	32130	FORMAT	00023			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00217.

TR-67-700-10-2
Page III-0-11
Program Listing
Link 0

INF,NO10001.44800-0.77387.AM IC

IBJOB V13MJD 7004 A 10/31/67

PAGE 10

00
STITLE ABNORMAL JOB TERMINATION ROUTINE
SIBFC GOOP 22 M04,MR7 LINK 0 (GOOP)

67001000

TR-67-700-10-2
Page III-0-12
Program Listing
Link 0

ABNORMAL JOB TERMINATION ROUTINE
GOOP22 - EFN SOURCE STATEMENT - IPN(8) -

10/31/67

PAGE 1

```
C SUBROUTINE GOOP                                GF002000
C                                     GF003000
  **** THIS IS A DUMMY SUBROUTINE USED TO EXIT FROM A RUN ****
CALL EXIT                                     GF004000      2
RETURN                                       GF005000
END                                           GF006000
```

ABNORMAL JOB TERMINATION ROUTINE
 GOOP 22

10/31/67

PAGE 2

STORAGE MAP

SUBROUTINE GOOP
 ENTRY POINTS

GOOP SECTION 3

SUBROUTINES CALLED

EXIT SECTION 4

SYBLOC SECTION 9
 EPN IPN CORRESPONDENCE

EPN	IPN	LOCATION	EPN	IPN	LOCATION	EPN	IPN	LOCATION
-----	-----	----------	-----	-----	----------	-----	-----	----------

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 0005.

TR-67-700-10-2
Page III-0-14
Program Listing
Link 0

IMP .NO10001.44800-0.77387.AM IC

ISJOB V13HJD 7094 A 10/31/67

PAGE 6

BTITLE SUBROUTINE TO PERFORM PARAMETER SUBSTITUTIONS
BTITLE PRHSZZ M94.XRT LINK 0 (PRHSUB)

88001000

SUBROUTINE TO PERFORM PARAMETER SUBSTITUTIONS
PRMS77 - EFM SOURCE STATEMENT - IFN(8) -

10/31/67

PAGE 1

```

SUBROUTINE PRMSUB ( * )                                88001200
C
C   *** INPUTS ***                                    88001002
C NSUBT  NUMBER OF SUBSTITUTION SETS                  (DATA COMMON) 88001004
C NSUBP  NUMBER OF SUBSTITUTION PARAMETERS PER SET   (DATA COMMON) 88001006
C SUBST  SUBSTITUTION VECTOR                          (DATA COMMON) 88001008
C
C   *** OUTPUTS ***                                    88001010
C
C   THE OUTPUT IS A MODIFIED DATA VECTOR CONTAINING THE SUBSTITUTED 88001012
C   I.E. VALUES.                                       88001014
C * ALTERNATE RETURN USED UNTIL THERE ARE NO MORE SETS 88001016
C
C   *** DESCRIPTION ***                                88001018
C
C   PRMSUB IS DESIGNED TO CYCLE THROUGH THE NS SUBSTITUTION SETS. 88001020
C   TO THIS END IT MUST BE CALLED NS+1 TIMES IN SUCCESSION. AT THE 88001022
C   JTH CALL SET J IS SUBSTITUTED INTO THE DATA REGION AND THE 88001024
C   ALTERNATE RETURN IS USED. ON THE NS+1 ENTRY THE NOMINAL SYSTEMS 88001026
C   IS RESTORED FOLLOWED BY A NORMAL RETURN. IF CALLED AGAIN THE 88001028
C   SEQUENCE IS RESTARTED.                               88001030
C
COMMON/DATA/DUMMY(27),DATA(3483)                       88001032
EQUIVALENCE (NSUBT,DATA(6)),(NSUBP,DATA(7)),(SUBST(1),ISUBST(1)), 88001034
1DATA(2682))                                           88001036
DATA JSET / 0 /                                       88001400
INTEGER SORG                                         88001600
DIMENSION SUBST(SORG),ISUBST(SORG)                   88001800
C
C   START OF PRMSUB SUBROUTINE                          88002000
C
C
C   IS PARAMETER SUBSTITUTION DESIRED                    88002100
IF ( NSUBT .LE. 0 ) RETURN                            88002200
ISORG = NSUBT * NSUBP                                 88002300
C
C   FIRST TIME CALLED FOR SUBSTITUTION PROCESS           88002400
IF ( JSET .GT. 0 ) GO TO 3000                          88002500
C
C   INCREMENT SUBSTITUTION SET INDEX AND SET FLAG FOR 88002600
REPLACEMENT OF NOMIAL SET BY CURRENT SUBST. SET      88002700
2000 JSET = JSET + 1                                    88002800
NFLAG = 2                                             88002900
GO TO 4000                                             88003000
3000 NFLAG = 1                                         88003100
4000 SORG = ( JSET - 1 ) * NSUBP                       88003200
C
C   THIS LOOP REPLACES NOMIAL SET WITH SUBJ. IF NFLAG = 2 AND 88003300
RESTORES THE NOMIAL SET WHEN NFLAG = 1               88003400
5000 DO 6000 I = 1, NSUBP                              88003500
IX = ISORG + I                                        88003600
C   COMPUTE INDEX TO POINTER TO POSITION OF PARAMETER IN DATA 88003700
I1 = IABS( ISUBST( IX ) )                             88003800
C   COMPUTE INDEX TO SUBSTITUTION VALUE IN SUBST VECTOR 88003900
I2 = SORG + I                                         88004000
STEMP = DATA( I1 )                                   88004100
DATA( I1 ) = SUBST( I2 )                              88004200
6000 SUBST( I2 ) = STEMP                              88004300
C

```

SUBROUTINE TO PERFORM PARAMETER SUBSTITUTIONS
PRMSZZ - EFN SOURCE STATEMENT - IPN(S) -

10/31/67

PAGE 2

C	RETURN IF SUBSTITUTION SET HAS REPLACED THE NOMIAL SET	88007200
	IF (NFLAG .EQ. 2) RETURN 1	88007300
C		
C	HAVE ALL THE SUBSTITUTION SETS BEEN USED	88007500
	IF (JSET .LT. NSUBT) GO TO 8000	88007600
C		
C	SUBSTITUTION PROCESS COMPLETE, RESET SET INDEX	88008000
	JSET = 0	88008100
	RETURN	88009000
	END	88009200

SUBROUTINE TO PERFORM PARAMETER SUBSTITUTIONS
 PRMSZZ STORAGE MAP

10/31/67

PAGE 3

SUBROUTINE PRMSUB
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08668		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
DUMMY	00000	R	DATA	00033	R	NSUBT	00040	I	
NSUBP	00041	I	SUBST	05224	R	ISUBST	05224	I	
UNDIMENSIONED PROGRAM VARIABLES									
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
SDRG	08667	I	JORG	08670	I	JSET	08671	I	
MFLAG	08672	I	I	08673	I	IX	08674	I	
II	08675	I	I2	08676	I	STEMP	08677	R	

ENTRY POINTS

PRMSUB SECTION 3

SUBROUTINES CALLED

SYBLOC SECTION 6

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
3000	11A	08732	2000	8A	08724	4000	12A	08734
3000	13A	08741	6000	24A	07000			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07053.

"REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR"

TR-67-700-10-2
Page III-0-18
Program Listing
Link 0

INF .NO10001.44000-0.77387.AM IC TB JOB V13MJD 7094 A 10/31/67 PAGE 3

00
STITLE PRINT CONTROL ROUTINE PT001000
SIDFIC PRTCZZ H04.X07 LINK 0 (PRTCTL)

PRINT CONTROL ROUTINE
 PRTCTZ - EFM SOURCE STATEMENT - (FN18) -

10/31/67

PAGE 1

```

SUBROUTINE PRTCTL                                PT001500
C
C *** INPUTS ***                                PT001002
C TITLE A 72 CHARACTER BCD TITLE                PT001004
C DATE CURRENT DATE MONTH, DAY, YEAR           PT001008
C LINE CURRENT LINE NUMBER                     PT001008
C PAGE CURRENT PAGE NUMBER                     PT001010
C JTEMP OPTION NUMBER                          PT001010
C JSUBST SUBSTITUTION SET NUMBER              PT001014
C
C *** OUTPUTS ***                                PT001016
C PRTCTL INCREMENTS PAGE AND INITIALIZES LINE PT001018
C
C *** DESCRIPTION ***                            PT001020
C PRTCTL IS CALLED WHENEVER A NEW PAGE OF OUTPUT IS TO BE GENERATED. IT PAGE EJECTS AND PRINTS ONE LINE (AT THE TOP OF THE NEW PAGE) WHICH INCLUDES THE TITLE, DATE, OPTION NUMBER, SUBSTITUTION SET NUMBER, AND PAGE NUMBER.
C
C COMMON / DATA / DUMMY(11), TITLE(12), DATE(3), DATA(3404) PT002000
C INTEGER DATE PT002200
C COMMON / PRNT / LINE, PAGE PT002500
C INTEGER PAGE PT002700
C COMMON / HEADER / JTEMP, JSUBST PT003000
C
C THIS SUBROUTINE PRINTS TITLE, DATE, AND PAGE NO., AND CONTROL THE PRINT INDEXES PT004000
C
C PAGE = PAGE + 1 PT004100
C WRITE ( 6, 32100 ) TITLE, DATE, JTEMP, JSUBST, PAGE PT005000
32100 FORMAT( 1H1, 12AS, 3X, 12, 1H/, 12, 1H/, 12, 3X, 7HCONTROL, 13, PT006200
1 3X, 16HSUBSTITUTION SET, 13, 3X, 4HPAGE, 14 ) PT006400
C LINE = 1 PT007000
C RETURN PT009000
C END PT009200
  
```

2

PRINT CONTROL ROUTINE
 PRTC22

11/31/67

PAGE 2

STORAGE MAP

SUBROUTINE PRTC22
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
DUMMY1	00000	R	TITLE	00013	R	DATE	00027	I	
DATA	00032	R							
COMMON BLOCK			PRINT	ORIGIN	08667	LENGTH	00002		
LINE	00000	I	PAGE	00001	I				
COMMON BLOCK			HEADER	ORIGIN	08671	LENGTH	00002		
JTEMP	00000	I	JSUBST	00001	I				

ENTRY POINTS

PRTC22 SECTION 9

SUBROUTINES CALLED

.FWRD.	SECTION	10	.FSLO.	SECTION	11	.UNDB.	SECTION	12
.FFIL.	SECTION	13	.FCNV.	SECTION	14	SYSLOC	SECTION	15

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
3E100	FORMAT	08703						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 08775.

TR-67-700-10-2
Page III-0-21
Program Listing
Link 0

INF .NO10001.44800-0.77387.AM IC

IBJOB V13WJD 7094 A 10/31/67

PAGE 4

80
STITLE RAY TRACE ROUTINE
SIBF IC RAYTRZ M94.XR7 LINK 0 (RAYTR) RY001000

RAY TRACE ROUTINE
RAYTRZ - EFN SOURCE STATEMENT - IFN(S) -

10/31/67

PAGE 1

```

SUBROUTINE RAYTR(XDHAT,YDHAT,HEIGHT,COLOR,XVCT,QVCT,SMISS,*,*,*,
1DFLAG,XMTRX,QMTRX) RY001500
C RY001600
C RY002000
C XDHAT X COORDINATE OF RAY ON UNIT RADIUS ENTRANCE PUPIL RY002100
C YDHAT Y COORDINATE OF RAY ON UNIT RADIUS ENTRANCE PUPIL RY002200
C HEIGHT OBJECT HEIGHT RY002300
C COLOR COLOR NUMBER RY002400
C XVCT VECTOR FOR STORING XN,YN,ZN RY002500
C QVCT VECTOR FOR STORING QXN,QYN,QZN RY002600
C SMISS SET BY RAYTR TO SURFACE NO. IF MISS, REFLECT, OR VIGNET RY002700
C * RETURN 1 IF RAY MISSES A SURFACE RY002800
C * RETURN 2 IF RAY IS INTERNALLY REFLECTED RY002900
C * RETURN 3 IF RAY VIGNETS RY003000
C DFLAG SET NON-ZERO BY CALLING PROGRAM IF DATA TO BE SAVED AT ALL SURF. RY003100
C XMTRX MATRIX FOR STORING (XI,UI,ZI) I=0,1,...,N RY003200
C QMTRX MATRIX FOR STORING (QXI,QYI,QZI) I=0,1,...,N RY003300
C RY003400
COMMON/AZOBJ/THTR,SNTHTR,CSTHTR RY004000
C RY004200
C THTR THETA AZIMUTH OF OBJECT IN RADIAN RY004300
C SNTHTR SIN(THETA) RY004400
C CSTHTR COS(THETA) RY004500
C RY005000
C RAYTR TRACES A RAY FROM THE OBJECT POINT (HEIGHT,THTR) WITH UNIT RAD- RY005100
C IUS ENTRANCE PUPIL COORDINATES (XDHAT,YDHAT) TO A POINT (XN,YN,ZN) ON RY005200
C THE FINAL LENS SURFACE. RY005300
C RY005400
COMMON/DATA/DUMMY(27),DATA(3483) RY006000
DIMENSION XVCT(3),QVCT(3),XMTRX(3,1),QMTRX(3,1),SURFC(20,100), RY006500
1SURFC(20,100) RY006600
EQUIVALENCE (NSURF,DATA(12)),(RHOD,DATA(26)),(S,DATA(31)), RY007000
1(SURFC(1,1),1SURFC(1,1),DATA(182)) RY007100
INTEGER COLOR,SMISS,DFLAG,TYPE,VIGFLG,TLFLG RY007500
REAL LAMDA,NX,NY,NZ,KAPPA,NU,MU RY008000
C INITIALIZE SMISS AND VIGFLG ... RY101000
SMISS=0 RY101100
VIGFLG=0 RY101200
C RY101300
COMPUTE PD AND QD ... RY101400
C RY101500
X=RHOD*XDHAT RY101600
Y=RHOD*YDHAT RY101700
Z=0. RY101800
QX=X+HEIGHT*SNTHTR RY101900
QY=Y+HEIGHT*CSTHTR RY102000
QZ=S-SURFC(4,1) RY102100
DELTA=SQRT(QX*QX+QY*QY+QZ*QZ) RY102200
QX=QX/DELTA RY102300
QY=QY/DELTA RY102400
QZ=QZ/DELTA RY102500
C RY103000
C DO LOOP TO TRACE FROM I-1 TO I WHERE I=1,2,...,NSURF RY103100
C RY103200
DO 850 I=1,NSURF RY103300
C RY103400

```

RAY TRACE ROUTINE
RAYTRZ - EFN SOURCE STATEMENT - IFN(S) -

10/31/67

PAGE 2

```

C IF DFLAG.NE.0 SAVE P(I-1) AND Q(I-1) ...
C
  IF (DFLAG.EQ.0) GO TO 60
  XMTRX(1,I)=X
  XMTRX(2,I)=Y
  XMTRX(3,I)=Z
  QMTRX(1,I)=QX
  QMTRX(2,I)=QY
  QMTRX(3,I)=QZ
60  IF (QZ.EQ.0.) GO TO 190
C
C GET PARAMETERS FOR ITH SURFACE ...
C
  TYPE=ISURFC(1,I)
  R=SURFC(11,I)
  B=SURFC(12,I)
  SI=SURFC(4,I)
  IF (TYPE.EQ.4) GO TO 280
C
C SURFACE IS STANDARD CONIC OR ASPHERIC ...
C GET POINT ON VERTEX PLANE ...
C
  LAMDA=(SI-Z./QZ
90  X=X+LAMDA*QX
  Y=Y+LAMDA*QY
  Z=0.
  IF (R.NE.0.) GO TO 130
C
C SURFACE IS A PLANE ...
C
  IF (TYPE.GT.4) GO TO 430
  NX=0.
  NY=0.
  NZ=1.
  GO TO 550
130  GO TO (140,150,160,140,140,150,160),TYPE
C
C SURFACE OF REVOLUTION ...
C
140  ALFA=B*QZ*QZ-1.
  BETA=X*QX+Y*QY-R*QZ
  GAMMA=X*X+Y*Y
  GO TO 170
C
C HORIZONTAL CYLINDER ...
C
150  ALFA=(B-1.)*QZ*QZ-QY*QY
  BETA=Y*QY-R*QZ
  GAMMA=Y*Y
  GO TO 170
C
C VERTICAL CYLINDER ...
C
160  ALFA=(B-1.)*QZ*QZ-QX*QX
  BETA=X*QX-R*QZ
  GAMMA=X*X
RY103900
RY103600
RY103700
RY103800
RY103900
RY104000
RY104100
RY104200
RY104300
RY104400
RY105000
RY105100
RY105200
RY105300
RY105400
RY105500
RY105600
RY105700
RY106000
RY106100
RY106200
RY106300
RY106400
RY106500
RY106600
RY106700
RY106800
RY107000
RY107100
RY107200
RY107300
RY107400
RY107500
RY107600
RY107700
RY107800
RY109000
RY109100
RY109200
RY109300
RY109400
RY109500
RY109600
RY201000
RY201100
RY201200
RY201300
RY201400
RY201500
RY201600
RY202000
RY202100
RY202200
RY202300
RY202400
RY202500

```

RAY TRACE ROUTINE
RAYTRZ - EFN SOURCE STATEMENT - IFN(8) -

10/31/67

PAGE 3

```

170 DELTA=BETA*BETA+ALFA*GAMA
    IF (DELTA)190,200,210
C
C RAY MISSED SURFACE I ...
C
190 MISS=I
    RETURN I
200 LAMDA=-GAMA/BETA
    GO TO 220
210 LAMDA=-GAMA/(BETA+SIGN(SQRT(DELTA),BETA))
C
C COMPUTE PI ...
C
220 X=X+LAMDA*GX
    Y=Y+LAMDA*GY
    Z=LAMDA*GZ
C
C COMPUTE NI IF NOT ASPHERIC ...
C
    GO TO (240,250,260,240,430,430,430),TYPE
240 NX=X
    NY=Y
    GO TO 270
250 NX=0.
    NY=Y
    GO TO 270
260 NX=X
    NY=0.
270 NZ=(1.-B)*Z-R
    GO TO 550
C
C SURFACE IS NON-STANDARD CONIC ...
C
280 XT=SURFC(13,I)
    YT=SURFC(14,I)
    U=SURFC(15,I)
    V=SURFC(16,I)
C
C COMPUTE X0 PRIME ...
C
    X=X-XT
    Y=Y-YT
    Z=Z-ST
    IF (ABS(U)+ABS(V).NE.0.)GO TO 310
C
C SURFACE IS TRANSLATED ONLY ...
C
    TLTFLG=1
    LAMDA=-Z/GZ
    GO TO 90
C
C SURFACE IS ALSO TILTED ...
C
310 TLTFLG=2
    W=SQRT(1.-(U*U+V*V))
C
RY202600
RY202700
RY203000
RY203100
RY203200
RY203300
RY203400
RY203500
RY203600
RY203700
RY204000
RY204100
RY204200
RY204300
RY204400
RY204500
RY204600
RY205000
RY205100
RY205200
RY205300
RY205400
RY205500
RY205600
RY205700
RY205800
RY205900
RY206000
RY206100
RY206200
RY207000
RY207100
RY207200
RY207300
RY207400
RY207500
RY207600
RY208000
RY208100
RY208200
RY208300
RY208400
RY208500
RY208600
RY209000
RY209100
RY209200
RY209300
RY209400
RY209500
RY301000
RY301100
RY301200
RY301300
RY301400
RY302000

```

RAY TRACE ROUTINE
 RAYTRZ - EFM SOURCE STATEMENT - IY(8) -

10/31/67

PAGE 4

C COMPUTE XI PRIME ...

C
 ZPP=U0X+V0Y+W0Z
 QZPP=U00X+V00Y+W00Z
 LAMDA=-ZPP/QZPP
 X=X+LAMDA*QX
 Y=Y+LAMDA*QY
 Z=Z+LAMDA*QZ
 F(R.NE.O.)GO TO 380

C
 C TILTED PLANE ...

C
 NX=U
 NY=V
 NZ=W
 ZPP=U0X+V0Y+W0Z
 GO TO 550

C
 C TILTED CONIC ...

C
 380 ALFA=B+QZPP-1.
 BETA=X+QX+Y+QY+Z+QZ-R+QZPP
 GAMMA=X+X+Y+Y+Z+Z
 DELTA=BETA+BETA+ALFA+GAMA
 IF (DELTA) 190,380,390
 390 LAMDA=-GAMA/BETA
 GO TO 400
 390 LAMDA=-GAMA/(BETA+SIGN(SQRT(DELTA),BETA))

C COMPUTE XI PRIME ...

C
 400 X=X+LAMDA*QX
 Y=Y+LAMDA*QY
 Z=Z+LAMDA*QZ

C COMPUTE NI ...

C
 ZPP=U0X+V0Y+W0Z
 KAPPA=-R-B+ZPP
 NX=X+KAPPA*U
 NY=Y+KAPPA*V
 NZ=Z+KAPPA*W
 GO TO 550

C SURFACE IS ASPHERIC ...

C
 430 A2=SURFC(13,I)
 A3=SURFC(14,I)
 A4=SURFC(15,I)
 A5=SURFC(16,I)
 B2=2.*A2
 B3=3.*A3
 B4=4.*A4
 B5=5.*A5
 T01=1.-B
 T02=R+R

RY302100
 RY302200 71
 RY302300
 RY302400
 RY302500
 RY302600
 RY302700
 RY302800
 RY302900
 RY303000
 RY303100
 RY303200
 RY303300
 RY303400
 RY303500
 RY303600
 RY303700
 RY304000
 RY304100
 RY304200
 RY304300
 RY304400
 RY304500
 RY304600
 RY304700
 RY304800
 RY304900
 RY305000
 RY306000
 RY306100 81
 RY306200
 RY306300
 RY306400
 RY306500
 RY307000
 RY307100
 RY307200
 RY307300
 RY307400
 RY307500
 RY307600
 RY307700
 RY307800
 RY308000
 RY308100
 RY308200
 RY308300
 RY308400
 RY308500
 RY308600
 RY308700
 RY308800
 RY308900
 RY309000
 RY309100
 RY309200

RAY TRACE ROUTINE
 RAYTRZ - E/M SOURCE STATEMENT - IFN(8) -

10/31/67

PAGE 9

JCNT= 3	RY309300
THOLD=0.	RY309400
C	RY401000
C THE FOLLOWING IS AN ITERATION LOOP WHICH CONVERGES TO THE REQUIRED	RY401100
C ROOT.	RY401200
C	RY401300
440 GO TO (4401,4401,4401,4401,450,460,470),TYPE	RY401400
4401 CALL GOOF	RY401500
C	
C SURFACE OF REVOLUTION...	RY401600
C	
450 PHI=X*X+Y*Y	RY401700
GO TO 460	RY401800
C	
C HORIZONTAL CYLINDER...	RY401900
C	
460 PHI=Y*Y	RY402000
GO TO 460	RY402100
C	
C VERTICAL CYLINDER...	RY402200
C	
470 PHI=X*X	RY402300
C	
COMPUTE ZKP...	RY402400
C	
480 ZP=(((A5*PHI+A4)*PHI+A3)*PHI+A2)*PHI*PHI	RY402500
IF (R.EQ.D.) GO TO 531	RY402600
DELTA1=T02-T01*PHI	RY402700
IF (DELTA1.LT.D.) GO TO 190	RY402800
DELTA1=SIGN(SQRT(DELTA1),R)	RY402900
ZP=ZP+PHI/(R+DELTA1)	RY403000
C	
COMPUTE THETA...	RY403100
C	
531 LAMDA=ZP-Z	RY403200
IF (LAMDA.EQ.D.) GO TO 540	RY403300
LAMDA=LAMDA/02	RY403400
C	
COMPUTE P (K+1)...	RY403500
C	
X=X+LAMDA*0X	RY403600
Y=Y+LAMDA*0Y	RY403700
Z=ZP	RY403800
LAMDA=ABS(LAMDA)	RY403900
IF (LAMDA.GT.1.E-7) GO TO 539	RY404000
JCNT=JCNT-1	RY404100
IF (JCNT.GT.D) GO TO 440	RY404200
GO TO 540	RY404300
C	
C MAKE SURE THETA IS DECREASING...	RY404400
C	
539 IF (THOLD.EQ.D.) GO TO 5311	RY404500
IF (LAMDA.GE.THOLD) GO TO 190	RY404600
5311 THOLD=LAMDA	RY404700
GO TO 440	RY404800
C	

92

105

RAY TRACE ROUTINE
RAYTRZ - EFM SOURCE STATEMENT - IFN(8) -

10/31/67

PAGE 6

```

COMPUTE NORMAL VECTOR...
C
540  GPHI=((B5*PHI+B4)*PHI+B3)*PHI+B2)*PHI
      IF(R.EQ.0.) GO TO 543
      GPHI=GPHI*(1./(2.*DELTA1))
543  GO TO (4401,4401,4401,4401,544,545,546), TYPE
544  PHIX=2.*X
      PHIY=2.*Y
      GO TO 547
545  PHIX=0.
      PHIY=2.*Y
      GO TO 547
546  PHIX=2.*X
      PHIY=0.
547  NX=-GPHI*PHIX
      NY=-GPHI*PHIY
      NZ=1.
C
C CHECK FOR VIGNETTING ...
C
550  IF(VIGFLG.NE.0)GO TO 710
      A1I=SURFC(2,I)
      A2I=SURFC(3,I)
      IF(ABS(A1I)+ABS(A2I).EQ.0.)GO TO 710
      IF(A2I.NE.0.)GO TO 650
C
C CIRCULAR APERTURE ...
C
      RHO=X*X+Y*Y
      IF(TYPE.EQ.4 .AND. TLTFLG.EQ.2)RHO=RHO+Z*Z-Z*PP*Z*PP
      IF(A1I.LT.0.)GO TO 640
      IF(RHO.GT.A1I*A1I)VIGFLG=I
      GO TO 710
640  IF(RHO.LT.A1I*A1I)VIGFLG=I
      GO TO 710
C
C RECTANGULAR APERTURE ...
C
650  RHOX=ABS(X)
      RHOY=ABS(Y)
      IF(TYPE.NE.4 .OR. (TYPE.EQ.4 .AND. TLTFLG.NE.2))GO TO 680
      G1=(1.-W)/(U*U+V*V)
      A1Z=-G1*U*V
      RHOX=ABS((1.-G1*U*U)*X+A1Z*Y-U*Z)
      RHOY=ABS(A1Z*X+(1.-G1*V*V)*Y-V*Z)
680  IF(A1I.LT.0.)GO TO 700
      IF(RHOX.GT.A1I .OR. RHOY.GT.A2I)VIGFLG=I
      GO TO 710
700  IF(RHOX.LT.ABS(A1I) .AND. RHOY.LT.A2I)VIGFLG=I
710  IF(TYPE.NE.4)GO TO 730
      X=X*XT
      Y=Y*YT
730  DELTA=SQRT(NX*NX+NY*NY+NZ*NZ)
      NX=NX/DELTA
      NY=NY/DELTA
      NZ=NZ/DELTA

```

RY404900
RY405000
RY405100
RY405200
RY405300
RY405400
RY405500
RY405600
RY405700
RY405800
RY405900
RY406000
RY406100
RY406200
RY406300
RY406400
RY407500
RY407600
RY407700
RY407800
RY407900
RY408000
RY408100
RY408200
RY408500
RY409000
RY409100
RY409200
RY409300
RY409400
RY409500
RY501000
RY501100
RY501200
RY502000
RY502100
RY502200
RY502300
RY502400
RY502500
RY502600
RY502700
RY502800
RY502900
RY503000
RY503100
RY503200
RY503300
RY503400
RY503500
RY503600
RY503700
RY503800
RY503900
RY504000

RAY TRACE ROUTINE
 RAYTRZ - EFN SOURCE STATEMENT - IFN(8) -

10/31/67

PAGE 7

C		RY601000
C		RY601100
C REFRACTION ...		RY601200
C		RY601300
C		RY601400
740 NU=SURFC(COLOR+4,I)/SURFC(COLOR+4,I+1)		RY601500
LAMDA=NX*QX+NY*QY+NZ*QZ		RY601600
IF (NU.GT.D.)GO TO 780		RY601700
C		RY602000
C REFLECTION ...		RY602100
C		RY602200
LAMDA = -2. * LAMDA		RY602300
QX=QX+LA/DA*NX		RY602400
QY=QY+LA/DA*NY		RY602500
QZ=QZ+LA/DA*NZ		RY602600
GO TO 850		RY602700
780 DELTA=(NU*NU) - ((NU*NU) - 1.) / (LAMDA*LAMDA)		RY602800
IF (DELTA) 800,810,820		RY602900
C		RY603000
C INTERNAL REFLECTION ...		RY603100
C		RY603200
800 SMISS=1		RY603300
RETURN 2		RY603400
810 MU=-NU*LAMDA		RY603600
GO TO 830		RY603700
820 MU=(-NU+SQRT(DELTA))*LAMDA	192	RY605000
830 QX=NU*QX+MU*NX		RY605100
QY=NU*QY+MU*NY		RY605200
QZ=NU*QZ+MU*NZ		RY605300
850 CONTINUE		RY605400
XVCT(1)=X		RY605500
XVCT(2)=Y		RY605600
XVCT(3)=Z		RY605700
QVCT(1)=QX		RY605800
QVCT(2)=QY		RY605900
QVCT(3)=QZ		RY606000
IF (DFLAG.EQ.0)GO TO 890		RY606100
DO 880 J=1,3		RY606200
XMTRX(J,NSURF+1)=XVCT(J)		RY606300
880 QMTRX(J,NSURF+1)=QVCT(J)		RY606400
860 IF (VIGFLG.EQ.0) RETURN		RY607000
SMISS=VIGFLG		RY607100
RETURN 3		RY608000
END		RY609000

RAY TRACE ROUTINE
RAYTRZ

10/31/67

PAGE 8

STORAGE MAP

SUBROUTINE RAYTR
COMMON VARIABLES

COMMON BLOCK			ATJOB	ORIGIN	00001	LENGTH	00003	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R
COMMON BLOCK			DATA	ORIGIN	00004	LENGTH	00006	
DUMMY	00000	R	DATA	00033	R	NSURF	00046	I
RHO	00084	R	B	00071	R	SHRFC	00320	R
TSURFC	00320	I						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
TIME	08672	I	VIGFLG	08673	I	TLIFLG	08674	I
LAMDA	08675	R	NK	08676	R	NY	08677	R
NZ	08700	R	KAPPA	08701	R	NU	08702	R
MU	08703	R	X	08704	R	Y	08705	R
Z	08708	R	QX	08707	R	QY	08710	R
QZ	08711	R	DELTA	08712	R	I	08713	I
R	08714	R	B	08715	R	SI	08716	R
ALFA	08717	R	BETA	08720	R	GAMA	08721	R
XT	08722	R	YT	08723	R	U	08724	R
V	08725	R	W	08726	R	ZPP	08727	R
QZPP	08730	R	A2	08731	R	A3	08732	R
A4	08733	R	A5	08734	R	B2	08735	R
B3	08736	R	B4	08737	R	B5	08740	R
T01	08741	R	T02	08742	R	JCNT	08743	I
THOLD	08744	R	PHI	08745	R	ZP	08746	R
DELTA1	08747	R	GPHT	08750	R	PHIX	08751	R
PHIY	08752	R	A11	08753	R	A21	08754	R
RHO	08755	R	RHOX	08756	R	RHOY	08757	R
G1	08760	R	A12	08761	R			

ENTRY POINTS

RAYTR SECTION 7

SUBROUTINES CALLED

SORT SECTION 8
SYSLOC SECTION 11

GOOF SECTION 9

.FREM. SECTION 10

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
80	194A	0755	80	17A	07117	190	44A	07326
280	59A	07432	90	28A	07143	130	36A	07172

TR-67-700-10-2
 Page III-0-30
 Program Listing
 Link 0

RAY TRACE ROUTINE RAYTRZ			STORAGE MAP			10/31/67	PAGE 9	
430	84A	07760	550	137A	10307	140	37A	07206
150	39A	07237	160	41A	07266	170	42A	07314
200	46A	07333	210	48A	07341	220	50A	07360
240	52A	07407	250	54A	07414	260	56A	07420
270	57A	07423	310	69A	07474	360	76A	07611
380	78A	07662	390	80A	07670	400	82A	07707
440	90A	10015	4401	91A	10031	450	93A	10034
460	95A	10044	470	97A	10050	480	98A	10033
531	106A	10130	540	126A	10211	539	118A	10175
5311	124A	10206	543	130A	10241	544	131A	10255
545	133A	10264	546	135A	10271	547	136A	10275
710	173A	10556	650	160A	10406	640	156A	10376
680	164A	10523	700	170A	10543	730	176A	10570
740	178A	10621	780	185A	10665	800	187A	10706
810	189A	10713	820	191A	10720	830	193A	10730
890	210A	11020	860	205A	11014			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 11157.

TR-67-700-10-2
Page III-1-1
Program Listing
Link 1

Section 1

LINK 1

TR-67-700-10-2
Page III-1-2
Program Listing
Link 1
Page Revised 10/1/68

INF,NO10001,44903-0,77387,AM IC

10J05 L10KF 7094 A 09/19/68

PAGE

10

TITLE OCTAL CARD PUNCH ROUTINE
SIDFC OFUNCZ M94,XR7 LINK 1 (OFUNC)

OF001000

OCTAL CARD PUNCH ROUTINE
OPUNCZ

09/19/68

PAGE 2

STORAGE MAP

SUBROUTINE OPUNCH
COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
DUMMY	00005	R	TITLE	00013	R	DATE	00027	I	
FUNCID	00032	R	DATA	00033	R	IDATA	00033	I	
UNDIMENSIONED PROGRAM VARIABLES									
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
I	06672	I	ITEMP	06673	I	JTEMP	06674	I	
KTEMP	06675	I	K	06676	I				

ENTRY POINTS

OPUNCH SECTION 5

SUBROUTINES CALLED

.FPUN. SECTION 6
SYSLOC SECTION 9

.FFIL. SECTION 7

.FCNV. SECTION 8

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	FORMAT	06712	32010	FORMAT	06714	1000	38A	07047
750	33A	07026	500	30A	07023	32020	FORMAT	06732

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07072.

TR-67-700-10-2
Page III-1-5
Program Listing
Link 1
Page Revised 10/1/68

INF,NO10001,44903-0,77367,AM IC IDJOB L10MKE 7094 A 09/19/68 PAGE

BT
BTITLE INPUT ROUTINE (CASE DECK CARD PROCESSOR)
SIDFTC READSZ M94,XR7 LINK 1 (READS)

R0001000

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
 READS2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

```

SUBROUTINE READS                                R0001100
C
C *** DESCRIPTION ***                            R0001001
C READS READS AND PROCESSES ALL OF THE INPUT CARDS WHICH COMPRISE R0001002
C THE CURRENT CASE DECK. EVERY CASE DECK BEGINS WITH CONTROL R0001003
C NAMELIST DATA ($INPUT) AND ENDS WITH AND END OF CASE CARD. R0001004
C THE CASE DECK IS (IN GENERAL) COMPOSED OF 10 SETS OF CARDS AS R0001005
C FOLLOWS ..... R0001006
C
C 1. CONTROL INPUT (MANDATORY) ONE OR MORE CARDS R0001007
C R0001008
C $INPUT CLEAR,IFLAG,NCNTRL,CONTRL,FUNCID $ R0001009
C R0001010
C 2. TITLE AND DATE (ONLY IF CLEAR=1) TWO CARDS R0001011
C R0001012
C TITLE (12A6) R0001013
C MONTH,DAY,YEAR (315) R0001014
C R0001015
C R0001016
C 3. OCTAL CARDS (ONLY IF CONTRL(1)=1) TWO OR MORE CARDS R0001017
C R0001018
C OCTAL CARDS (SEQUENCED) (6012,5X13) R0001019
C R0001020
C 4. GEOMETRY DATA (ONLY IF IFLAG(1)=1) ONE OR MORE CARDS R0001021
C R0001022
C $GEOM LMODE,NRAYS,DELY,NSLCS,LATTC, R0001023
C NCLRS,NIFLN,IMODE,CIMPL,NSPLN, R0001024
C DLPLN,OMGA2,OMGA1,DELD,EPRAD, R0001025
C PSCAL,ODIST,HD,DELH,NDBJH, R0001026
C ROTAN,NSURF,SURFC $ R0001027
C R0001028
C 5. DESIGN DATA (ONLY IF IFLAG(2)=1 OR 2) ONE OR MORE CARDS R0001029
C R0001030
C $DESIGN NDSGN,NDSGV,NJAIL,FNUMB,FLNGH, R0001031
C WFLGH,ZETA,HEXFF,DEXFF,WEXFF, R0001032
C WDBJH,WCLRS,WXDIR,WYDIR,WCLRH, R0001033
C EIMHT,WIMHT,OMGAF,SPFEA,DUMIN, R0001034
C SYSMX,ITPRNT,NCOND,DESIGN,ATRGR,GAUSS $ R0001035
C R0001036
C DESIGN SYMBOL CARDS (ONLY IF IFLAG(2)=1) (6(A1,2A6)) R0001037
C R0001038
C 6. BOUNDARY COND. DATA (ONLY IF IFLAG(6)=1 OR 2) ONE OR MORE CARDS R0001039
C R0001040
C $BNDRY BOUNDS $ R0001041
C R0001042
C BOUNDARY COND. SYMBOL CARDS (ONLY IF IFLAG(6)=1) (6(1X,2A6)) R0001043
C R0001044
C 7. SUBSTITUTION SET DATA (ONLY IF IFLAG(3)=1 OR 2) ONE OR MORE CDS. R0001045
C R0001046
C $SUBSTN NSUBT,NSUCF,SUBST $ R0001047
C R0001048
C SUBSTITUTION SYMBOL CARDS (ONLY IF IFLAG(3)=1) (6(1X,2A6)) R0001049
C R0001050
C 8. SENSITIVITY DATA (ONLY IF IFLAG(4)=1 OR 2) ONE OR MORE CARDS R0001051
C R0001052
C $SENST NPERTD,REFOCS,PERTB $ R0001053
C

```

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
 READS2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

C		RD001054
C		RD001055
C		RD001056
C	SENSITIVITY SYMBOL CARDS (ONLY IF IFLAG(4)=1) (5(1X,2A5))	RD001057
C	9. PROFILE PLOT DATA (ONLY IF IFLAG(5)=1) ONE OR MORE CARDS	RD001058
C		RD001059
C	SCRFPLOT YMAXX,DZMIN,NPTS 8	RD001060
C		RD001061
C	10.END OF CASE CARD (MANDATORY) ONE CARD	RD001062
C		RD001063
C	END OF CASE (8DA1)	
C		DT001000
C	COMMON / DATA / NCNTRL, CONTRL(15), TITLE(12), DATE(3), FUNCID,	DT001100
C	1 DATA(3486)	DT002000
C	DIMENSION WOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7),	DT002100
C	1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100),	DT002200
C	2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26),	DT002300
C	3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100)	DT101000
C	EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3),	DT101100
C	1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6),	DT101200
C	2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9),	DT101300
C	3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12),	DT101400
C	4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15),	DT101500
C	5 FNUMD), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18),	DT101600
C	6 ZETA), (DATA(19), HEXPF), (DATA(20), DEXPF), (DATA(21),	DT101700
C	7 WEXPF), (DATA(22), DLFLN), (DATA(23), OMGA2), (DATA(24),	DT101800
C	8 OMGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27),	DT101900
C	9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30),	DT102000
C	A DUMIN), (DATA(31), ODIST), (DATA(32), HD), (DATA(33),	DT102100
C	B DELH), (DATA(34), SYSMX), (DATA(35), WXDIR), (DATA(36),	DT102200
C	C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42),	DT102300
C	D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLRH(1)),	DT102400
C	E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81),	DT102500
C	F CIMPL(1)), (DATA(88), WCLRS(1)),	DT102600
C	G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(182), SURFC(1,1),	DT102700
C	H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)),	DT102800
C	I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1),	DT102900
C	J IBNDS(1,1)), (DATA(3483), NCOND)	DT103000
C	EQUIVALENCE (DATA(3484), ATRGGR), (DATA(3485), GAUSS)	DT201000
C	INTEGER CONTRL, AFLAG, DATE, ATRGGR	DT202000
C	REAL LATT	
C		SY001000
C	THIS BLOCK OF COMMON CONTAINS THE SYMBOLS USED IN DESIGN	SY002000
C	AND SUBSTITUTION - UTILIZED FOR PRINT OUT	SY003000
C	COMMON / SYMOLS / SDESN(12,50), SUBSYM(2,250), BDYSYM(1,100)	SY004000
C	INTEGER SDESN, SUBSYM, BDYSYM	
C		PT002500
C	COMMON/PRNT/LINE,PAGE	PT002700
C	INTEGER PAGE	
C		FB001000
C	THIS BLOCK OF COMMON CONTAINS PERTURBATION VALUES	FB002000
C	COMMON / PERTB / PERTB(30,4), NPRTB, REFOCS	FB003000
C	EQUIVALENCE (PERTB(1,1), IPERTB(1,1))	FB004000
C	DIMENSION IPERTB(30,4)	FB005000
C	INTEGER REFOCS	
C		FC001000
C	COMMON / PLOT / YMAXX, DZMIN, NPTS	

INPUT ROUTINE (CASE DECK CARD PROCESSOR)

READS2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

```

C
C DIMENSION IFLAG(6), ISIGN(6)                                R0001200
C
C INTEGER SYM1(2,13), SYM2(2,25), SYM3(2,2), SYM4(2,7),    R0002000
1 SYM5(2,3), CLEAR, BUFFER(12), BLANK                      R0002100
C DATA NSYMB1 / 13 /, NSYMB2 / 25 /, NSYMB3 / 2 /, NSYMB4 / 7 /, R0002500
1 NSYMB5 / 3 /                                             R0002600
C
C DATA SYM1 / 6HLMODE , 1, 6HNRAYS , 2, 6HNSLCS , 3, 6HNCLRS , 4, R0003000
1 6HNJAIL , 5, 6HNSUDT , 6, 6HNSUDF , 7, 6HNIFLN , 6, 6HIMODE , 9, R0003100
2 6HNSPLN , 10, 6HNOBJH , 11, 6HNSURF , 12 , 6HATRCOR, 3484 / R0003200
C
C DATA SYM2 / 6HDELY , 14, 6HFNUMB , 15, 6HFLNGH , 16, 6HWFLGH , R0003500
1 17, 6HZETA , 18, 6HHEXFF , 19, 6HDEXFF , 20, 6HWEXFF , 21, R0003600
2 6HDLFLN , 22, 6HOMGA2 , 23, 6HOMGA1 , 24, 6HDELD , 25, 6HEPRAD , R0003700
3 26, 6HFSCAL , 27, 6HOMGAF , 28, 6HSFFEA , 29, 6HDUMIN , 30, R0003800
4 6HODIST , 31, 6HND , 32, 6HDELH , 33, 6HSYSMX , 34, 6HWXDIR , R0003900
5 35, 6HWYDIR , 36, 6HROTAN , 37, 6HGAUSS , 3485 / R0004000
C
C DATA SYM3 / 6HNSGN(, 38, 6HNSGV(, 42 / R0004500
C DATA SYM4 / 6HNOBJH(, 53, 6HWCLR(, 60, 6HWIMHT(, 67, 6HEIMHT(, R0005000
1 74, 6HCIMPL(, 81, 6HWCLRS(, 88, 6HSUBST(, 2682 / R0005100
C DATA SYM5 / 6HLATTC(, 94, 6HSURFC(, 182, 6HDESGN(, 2182 / R0005500
C
C NAMELIST / INPUT / IFLAG, NCTRL, CONTRL, CLEAR, FUNCID R0006500
C
C IFLAG(5) NAMELIST DATA PRESENT IF VALUE(S) IS 1. R0006502
C NCTRL NUMBER OF CONTROLS TO BE EXECUTED. R0006504
C CONTRL(10) CONTROLS (OPTIONS) TO BE EXECUTED. R0006506
C CLEAR CLEAR DATA REGION IF NON-ZERO. R0006508
C FUNCID OCTAL DECK ID. R0006510
C
C NAMELIST / GEOM / LMODE, NRAYS, DELY, NSLCS, LATTC, NCLRS, R0007000
1 NIPLN, IMODE, CIMPL, NSPLN, DLPLN, OMGA2, OMGA1, DELD, EPRAD, R0007100
2 FSCAL, ODIST, HD, DELH, NOBJH, SURFC, NSURF, ROTAN R0007200
C
C LMODE LATTICE MODE (0,1, OR 2) DETERMINES PATTERN. R0007202
C NRAYS NUMBER OF RAYS TO BE GENERATED. R0007204
C DELY Y SPACING INCREMENT. R0007206
C NSLCS NUMBER OF SLICES (TRIFLETS) ON ENTRANCE PUPIL. R0007208
C LATTC (X1,Y1,N1) I=1,2,...,NSLCS RAY PATTERN MATRIX. R0007210
C NCLRS NUMBER OF COLORS. R0007212
C NIPLN NUMBER OF PLANES (OPTICS ONLY). R0007214
C IMODE PLANE SETTING MODE (0 OR 1). R0007216
C CIMPL(7) Z DISTANCES FOR CURVATURE OF FIELD. R0007218
C NSPLN LOCATION FOR FIRST PLANE TABULATED. R0007220
C DLPLN INCREMENT FOR PLANE SPACING. R0007222
C OMGA2 APERTURE SCALER FOR F.L. (.00001 TYP.). R0007224
C OMGA1 APERTURE SCALER FOR D.F. (.707 TYP.). R0007226
C DELD BACK FOCUS INCREMENT. R0007228
C EPRAD RADIUS OF ENTRANCE PUPIL. R0007230
C FSCAL PLOT SCALE FACTOR. R0007232
C ODIST DISTANCE FROM OBJECT TO FIRST SURFACE. R0007234
C HD INITIAL OBJECT HEIGHT. R0007236
C DELH OBJECT HEIGHT INCREMENT. R0007238
C NOBJH NUMBER OF OBJECT POINTS. R0007240

```

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
 READS2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

C NSURF	NUMBER OF LENS SURFACES.			R0007242
C ROTAN	AZIMUTH OF OBJECT (DEGREES).			R0007244
C SURFC(20,100)	SURFACE MATRIX... COLUMN J DESCRIBES JTH SURFACE AS			R0007246
CC	FOLLOWS			R0007248
CC				R0007250
C	SURFC(1, J)	SURFACE TYPE		R0007252
C	SURFC(2, J)	A1) APERTURE PARAMETER		R0007254
C	SURFC(3, J)	A2) APERTURE PARAMETER		R0007256
C	SURFC(4, J)	DISTANCE FROM PREVIOUS SURFACE		R0007258
C	SURFC(5, J)	REFRACTIVE INDEX COLOR 1		R0007260
C	SURFC(6, J)	REFRACTIVE INDEX COLOR 2		R0007262
C	.	.		R0007264
C	.	.		R0007266
C	.	.		R0007268
C	SURFC(10, J)	REFRACTIVE INDEX COLOR 6		R0007270
C	SURFC(11, J)	RADIUS OF CURVATURE		R0007272
C	SURFC(12, J)	ECCENTRICITY PARAMETER		R0007274
C				R0007276
C	TILTED/TRANSLATED			R0007278
C				R0007280
C	SURFC(13, J)	XT HOR. AXIS DISPLACEMENT		R0007282
C	SURFC(14, J)	YT VRT. AXIS DISPLACEMENT		R0007284
C	SURFC(15, J)	U X DIRECTION COSINE		R0007286
C	SURFC(16, J)	V Y DIRECTION COSINE		R0007288
C				R0007290
C	ASPHERIC (POLYNOMIAL) ...			R0007292
C				R0007294
C	SURFC(13, J)	A2 DEFORMATION COEFFICIENT		R0007296
C	SURFC(14, J)	A3 DEFORMATION COEFFICIENT		R0007298
C	SURFC(15, J)	A4 DEFORMATION COEFFICIENT		R0007300
C	SURFC(16, J)	A5 DEFORMATION COEFFICIENT		R0007302
C				
C	NAMelist / DESIGN / DESGN, NDSGN, NDSGV, NJAIL, FNUMB, FLNGH,			R0008000
C	1 WFLGH, ZETA, HEXFF, DEXFF, WEXFF, WODJH, WCLRH, WIMHT, EIMHT,			R0008100
C	2 DMGAF, SPFEA, DUMIN, WCLRS, WDIR, WYDIR, NCOND, SYSMX, ITAPRT			R0008200
C	3 , ATROGR, GAUSS			R0008201
C				
C	NDSGN(1)	NUMBER OF DESIGN VARIABLES.		R0008202
C	NDSGN(2)	MINIMUM PERMISSIBLE DESIGN VARIABLE NUMBER.		R0008204
C	NDSGN(3)	MAXIMUM PERMISSIBLE DESIGN VARIABLE NUMBER.		R0008206
C	NDSGN(4)	NUMBER OF DESIGN ITERATIONS.		R0008208
C	NDSGV(1)	NUMBER OF DESIGN VARIABLES PER ITERATION.		R0008210
C	NDSGV(2,...)	INITIAL SET OF DESIGN VARIABLE NUMBERS.		R0008212
C	NJAIL	JAIL SENTENCE.		R0008214
C	FNUMB	REQUIRED F/ NUMBER.		R0008216
C	FLNGH	REQUIRED FOCAL LENGTH.		R0008218
C	WFLGH	WEIGHT ON FOCAL LENGTH.		R0008220
C	ZETA	MAXIMUM ALLOWABLE INCREMENT FACTOR.		R0008222
C	HEXFF	OBJECT HEIGHT FOR EXIT PUPIL.		R0008224
C	DEXFF	EXIT PUPIL DISTANCE FROM LAST SURFACE.		R0008226
C	WEXFF	WEIGHT ON EXIT PUPIL DISTANCE.		R0008228
C	WODJH(7)	SET OF (OBJECT HEIGHT DEPENDENT) WEIGHTS ON SPOT SIZE.		R0008230
C	WCLRH(7)	SET OF WEIGHTS ON LATERAL CHROMATIC ABERRATION.		R0008232
C	WIMHT(7)	SET OF WEIGHTS ON EXPECTED X BARS.		R0008234
C	EIMHT(7)	SET OF EXPECTED Y BARS.		R0008236
C	DMGAF	FEATHER APERTURE SCALER (+1, TYP.).		R0008238

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
READS7 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

C SPFEA	MINIMUM PERMISSIBLE FEATHER THICKNESS.	R0008240
C DUMIN	MINIMUM INCREMENT VALUE.	R0008242
C WCLRS(6)	SET OF (COLOR DEPENDENT) WEIGHTS ON SPOT SIZE.	R0008244
C WXDIR	X DIRECTIONAL WEIGHT.	R0008246
C WYDIR	Y DIRECTIONAL WEIGHT.	R0008248
C NCOND	NUMBER OF BOUNDARY CONDITIONS.	R0008250
C BYSMX	MAXIMUM PERMISSIBLE LENS LENGTH.	R0008252
C ITRNT	SUBSIDIARY DESIGN PRINT TRIGGER.	R0008254
C ATRCCR	OPTION FLAG TO RE-COMPUTE OR NOT RE-COMPUTE D(J).	R0008255
C GAUSS	STANDARD DEVIATION FOR GAUSSIAN WEIGHTING.	R0008255
C DESGN(10,50)	DESIGN MATRIX... COLUMN J DEFINES JTH COMPOSITE DESIGN	R0008256
C	VARIABLE AS FOLLOWS	R0008258
C		R0008260
C	DESGN(1,J) NUMBER OF PARAMETERS	R0008262
C	DESGN(2,J) IJ,IJ REQUIRED	R0008264
C	DESGN(3,J) DELTA VJ	R0008266
C	DESGN(4,J) K1J (DATA INDEX PARAMETER 1)	R0008268
C	DESGN(5,J) K2J (DATA INDEX PARAMETER 2)	R0008270
C	.	R0008272
C	.	R0008274
C	.	R0008276
C	DESGN(9,J) K3J (DATA INDEX PARAMETER 6)	R0008278
C	DESGN(10,J) IJ,IJ COMPUTED	R0008280
C		
C	NAMelist / SUBSTN / SUBST, NSUBT, NSUBF	R0008500
C		
C NSUBT	NUMBER OF SUBSTITUTION SETS.	R0008502
C NSUBF	NUMBER OF PARAMETERS PER SET.	R0008504
C SUBST(500)	SUBSTITUTION VECTOR.	R0008506
C		
C	NAMelist / SENST / NPRTB, PERTB, REFOCS	R0008600
C		
C NPRTB	NUMBER OF PARAMETERS TO BE PERTURBED.	R0008602
C REFOCS	REFOCUS FLAG (0,1,2, OR 3).	R0008604
C PERTB(30,4)	PERTURBATION MATRIX... ROW J DEFINES THE JTH PERTURBATION	R0008606
C	PARAMETER AS FOLLOWS.....	R0008608
C		R0008610
C	PERTB(J,1) PER CENT BY WHICH PARAMETER IS CHANG.	R0008612
C	PERTB(J,2) DATA INDEX	R0008614
C	PERTB(J,3) FIRST 6 CHARACTERS OF SYMBOl	R0008616
C	PERTB(J,4) LAST 6 CHARACTERS OF SYMBOl	R0008618
C		
C	NAMelist / BNDRY / BOUNDS	R0008700
C		
C BOUNDS(3,100)	BOUNDARY CONDITION MATRIX... COLUMN J DESCRIBES THE JTH	R0008702
C	BOUNDARY CONDITION AS FOLLOWS.....	R0008704
C		R0008706
C	BOUNDS(1,J) DATA INDEX	R0008708
C	BOUNDS(2,J) MINIMUM PERMISSIBLE VALUE	R0008710
C	BOUNDS(3,J) MAXIMUM PERMISSIBLE VALUE	R0008712
C		
C	NAMelist / CRPLOT / YMAXX, DZMIN, NPTS	R0008800
C		
C YMAXX	MAXIMUM APERTURE RADIUS	R0008802
C DZMIN	MINIMUM AIR SPACE THICKNESS	R0008804
C NPTS	NUMBER OF POINTS PER ARC	R0008806

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
READSZ - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE 1

C	DATA BLANK / 6H /, MINUS / 6H- /	RD008900	
	DIMENSION BUFFRR(6)	RD009100	
	EQUIVALENCE (AFLAG, ITHPRT)	RD009400	
	INTEGER SYHED(4,2)	RD009410	
	DATA SYHED(1,1)/19H WITH SYMBOLE CARDS /,	RD009420	
1	SYHED(1,2)/21H WITHOUT SYMBOLE CARDS/	RD009430	
C			
C	START OF SUBROUTINE READS	RD101000	
C			
C	THIS SUBROUTINE HANDLES ALL OF THE INPUT TO THE PROGRAM	RD101100	
C	READ IN PROGRAM CONTROL DATA	RD102000	
	READ (5, INPUT)	RD102100	1
	IF(CLEAR .EQ. 0) GO TO 1500	RD102200	
	PAGE=0	RD102300	
C	ZERO PROGRAM DATA AREA AND TITLE	RD102500	
	DO 1200 I = 1,3400	RD102600	
1200	DATA(I) = 0.0	RD102700	
C	ZERO PERTURBATION DATA	RD102800	
	DO 1400 J = 1, 30	RD102900	
	DO 1400 K = 1, 4	RD103000	
1400	PERTD(J,K) = 0.0	RD103100	
	NPERTD = 0	RD103200	
	REFOCS = 0	RD103300	
C	ZERO PLOTTING PARAMETERS	RD103310	
	YMAXX = 0.	RD103320	
	DZMIN = 0.	RD103330	
	NPTS = 0	RD103340	
C			
C	READ NEW TITLE	RD103400	
	READ (5,32100) (TITLE(I), I = 1,12)	RD103500	26
32100	FORMAT(12A6)	RD103600	
C	READ NEW DATE	RD103700	
	READ (5,32110) (DATE(I), I = 1,3)	RD103800	33
32110	FORMAT(3(3X, 12))	RD103900	
1500	IF(CONTRL(1) .NE. 1) GO TO 2000	RD104500	
C	READ PROGRAM DATA FROM OCTAL CARDS	RD105000	
	1600 READ (5,32115) (BUFFRR(J), J = 1,6), KTEMP	RD105200	43
32115	FORMAT(6012, 5X, 13)	RD105400	
	ITEMP = 6*KTEMP + 1	RD105600	
	JTEMP = ITEMP + 5	RD105800	
	J = 0	RD106000	
	DO 1800 I = ITEMP, JTEMP	RD106200	
	J = J + 1	RD106400	
1800	DATA(I) = BUFFRR(J)	RD106600	
	IF(KTEMP .LT. 580) GO TO 1600	RD106800	
C			
C	PRINT PROGRAM CONTROL INFORMATION	RD107000	
	2000 CALL PRCTCL	RD107100	66
	WRITE(6, 32050)	RD107200	67
32050	FORMAT(1HD, 10X, 28H PROGRAM CONTROL INFORMATION)	RD107300	
	IF(CLEAR .NE. 0) GO TO 2005	RD107500	
	WRITE (6, 32055)	RD107600	70
32055	FORMAT(1HD, 10H CLEAR = 0, 10X, 24H DATA REGION NOT CLEARED)	RD107700	
	GO TO 2010	RD107800	
2005	WRITE (6, 32060)	RD107900	72

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
READS2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE 7

32060	FORMAT(1HD, 10H CLEAR = 1, 10X, 24H DATA REGION WAS CLEARED)	RD108000	
2010	IF(IFLAG(1) .NE. 0) GO TO 2015	RD108200	
	WRITE (6, 32065)	RD108300	75
32065	FORMAT(1HD, 13H IFLAG(1) = 0, 7X, 17H NO GEOMETRY DATA)	RD108400	
	GO TO 2020	RD108500	
2015	WRITE (6, 32070)	RD108600	77
32070	FORMAT(1HD, 13H IFLAG(1) = 1, 7X, 22H GEOMETRY DATA FOLLOWS)	RD108650	
2020	IF(IFLAG(2) .NE. 0) GO TO 2025	RD108700	
	WRITE (6, 32075)	RD108750	80
32075	FORMAT(1HD, 13H IFLAG(2) = 0, 7X, 15H NO DESIGN DATA)	RD108800	
	GO TO 2030	RD108850	
2025	IDX=IFLAG(2)	RD108900	
	WRITE (6, 32080) IDX, (SYHED(1,IDX), I=1,4)	RD108910	83
32080	FORMAT(12HD IFLAG(2) = 12, 8X 19H DESIGN DATA FOLLOWS 4A6)	RD108920	
2030	IF(IFLAG(3) .NE. 0) GO TO 2035	RD109000	
	WRITE (6, 32085)	RD109050	92
32085	FORMAT(1HD, 13H IFLAG(3) = 0, 7X, 21H NO SUBSTITUTION DATA)	RD109100	
	GO TO 2040	RD109150	
2035	IDX=IFLAG(3)	RD109200	
	WRITE (6, 32090) IDX, (SYHED(1,IDX), I=1,4)	RD109210	95
32090	FORMAT(12HD IFLAG(3) = 12, 8X 25H SUBSTITUTION DATA FOLLOWS 4A6)	RD109220	
2040	IF(IFLAG(4) .EQ. 0) GO TO 2045	RD109300	
	IDX=IFLAG(4)	RD109370	
	WRITE (6, 32091) IDX, (SYHED(1,IDX), I=1,4)	RD109380	105
32091	FORMAT(12HD IFLAG(4) = 12, 8X 24H SENSITIVITY DATA FOLLOWS 4A6)	RD109390	
	GO TO 2050	RD109420	
2045	WRITE (6, 32092)	RD109450	113
32092	FORMAT(1HD, 13H IFLAG(4) = 0, 7X, 20H NO SENSITIVITY DATA)	RD109470	
2050	IF(IFLAG(5) .EQ. 0) GO TO 2055	RD109480	
	WRITE (6, 32093)	RD109482	116
32093	FORMAT(1HD, 13H IFLAG(5) = 1, 7X, 18H PLOT DATA FOLLOWS)	RD109484	
	GO TO 2200	RD109486	
2055	WRITE (6, 32094)	RD109488	118
32094	FORMAT(1HD, 13H IFLAG(5) = 0, 7X, 13H NO PLOT DATA)	RD109490	
2200	IF(IFLAG(6) .NE. 0) GO TO 2210	RD109500	
	WRITE (6, 32200)	RD109505	121
32200	FORMAT(1HD, 13H IFLAG(6) = 0, 7X, 27H NO BOUNDARY CONDITION DATA)	RD109510	
	GO TO 2290	RD109515	
2210	IDX=IFLAG(6)	RD109520	
	WRITE (6, 32210) IDX, (SYHED(1,IDX), I=1,4)	RD109525	124
32210	FORMAT(12HD IFLAG(6) = 12, 8X 31H BOUNDARY CONDITION DATA FOLLOWS 4A6)	RD109530	
2290	WRITE (6, 32095) NCNTRL, (CNTRL(1), I=1, NCNTRL)	RD109535	131
32095	FORMAT(1HD, 9H NCNTRL =, 12, 9X, 35H NUMBER OF PROGRAM CONTROL CO	RD109550	
	MMANDS // 10H CNTRL =, 1015)	RD109600	
	WRITE(6, 32096) PUNCID	RD109650	138
32096	FORMAT(1HD, 9H PUNCID =, 1X, A6)	RD109700	
	IF(IFLAG(7) .EQ. 0) GO TO 3000	RD109800	
C			
C	READ GEOMETRY DATA	RD201000	
C			
C	READ (5, GEOM)	RD201200	
C	TEST GEOMETRY INPUT DATA FOR CORRECT VALUES	RD201500	141
	IF(LMODE .LT. 0 .OR. LMODE .GT. 2) CALL ERROR2(36H INCORRECT	RD201600	
	IT VALUE FOR LMODE , LMODE)	RD201700	144
	IF(NRAYS .LT. 1 .OR. NRAYS .GT. 200) CALL ERROR2(36H NUMBER	RD202000	
	1 OF RAYS EXCEEDS LIMITS , NRAYS)	RD202100	147

INPUT ROUTINE (CASE DECK CARD PROCESSOR)

09/19/68

PAGE 8

READS7 - EFN SOURCE STATEMENT - IFN(S) -

```

IF( NSLCS .LT. 1 .OR. NSLCS .GT. 26 ) CALL ERROR2( 36H NUMBER RD202400
1OF E.F. SLICES EXCEEDS LIMIT, NSLCS ) RD202500 150
IF( NCLRS .LT. 1 .OR. NCLRS .GT. 6 ) CALL ERROR2( 36H NUMBER ORD202600
1F COLORS EXCEEDS LIMITS , NCLRS ) RD202700 153
IF( IMODE .LT. 0 .OR. IMODE .GT. 1 ) CALL ERROR2( 36H INCORRECT RD203400
1T VALUE FOR IMODE , IMODE ) RD203500 156
IF( NDBJH .LT. 1 .OR. NDBJH .GT. 7 ) CALL ERROR2( 36H NUMBER ORD203600
1F OBJECT HEIGHTS IS ILLEGAL, NDBJH ) RD203700 159
IF( NSURF .LT. 1 .OR. NSURF .GT. 98 ) CALL ERROR2( 36H NUMBER RD204000
1OF SURFACES EXCEEDS LIMITS , NSURF ) RD204100 162
IF( NIFLN .LE. 0 ) CALL ERROR2( 36H NUMBER OF IMAGE PLANES IS INRD204500
1CORRECT, NIFLN ) RD204600 165
IF( OMGA1 .EQ. 0.0 .OR. OMGA2 .EQ. 0.0 ) CALL ERROR( 36H SCALERD205000
1FACTOR IS 0.0 ) RD205100 168
IF( EFRAD .LE. 0.0 ) CALL ERROR3( 36H ENTRANCE PUPIL RADIUS IS 1RD205400
1LLEGAL , EFRAD ) RD205500

C
C CONVERT SURFACE TYPE TO INTEGER RD208000 171
2500 JTEMP = NSURF + 1 RD208050
DO 2525 I = 1, JTEMP RD208100
IF( ISURFC(I,1) .GT. 10000 ) ISURFC(I,1) = SURFC(I,1) RD208200
2525 CONTINUE RD208300
C TEST FOR VALID SURFACE DATA RD301000
DO 2650 I = 1, NSURF RD301100
IF( ISURFC(I,1) .LT. 1 .OR. ISURFC(I,1) .GT. 7 ) GO TO 2700 RD301200
DO 2550 J = 1, NCLRS RD302000
IF( SURFC(J+4,1) .EQ. 0.0 ) GO TO 2750 RD302100
2550 CONTINUE RD302200
IF( SURFC(3,1) .GE. 0.0 ) GO TO 2600 RD302400
CALL ERROR2( 36H SECOND APERTURE VALUE IS NEGATIVE , I ) RD302500 206
2600 IF( ISURFC(I,1) .NE. 4 .OR. ( SURFC(15,1) ** 2 + SURFC(16,1) ** 2) RD302600
1 .LE. 1.0 ) GO TO 2650 RD302700
CALL ERROR2( 36H INCORRECT DEFN. OF TILTED AXIS , I ) RD302900 215
2650 CONTINUE RD303000
GO TO 2800 RD303200
2700 CALL ERROR2( 36H INCORRECT SURFACE TYPE NUMBER GIVEN, I ) RD304000 221
2750 CALL ERROR2( 36H INDEX OF REFRACTION = 0.0 FOR SURFC, I ) RD304200 223
2800 KTEMP = 0 RD304900

C
C CONVERT NUMBER OF RAYS TO INTEGER RD305000
DO 2850 I = 1, NSLCS RD305100
IF( ILATTC(3,I) .GT. 10000 ) ILATTC(3,I) = LATTC(3,I) RD305200
C TEST NUMBER OF RAYS IN EACH SLICE RD305300
IF( ILATTC(3,I) .LT. 1 .OR. ILATTC(3,I) .GT. 200 ) CALL ERROR2 RD305400
1 ( 36H INCORRECT NUMBER OF RAYS IN SLICE , I ) RD305500 237
KTEMP = KTEMP + ILATTC(3,I) RD305600
2850 CONTINUE RD305700

C
C TEST TOTAL NUMBER OF RAYS SPECIFIED IN LATTC, TOTAL MUST RD306000
C EQUAL OR EXCEED NRAYS RD306100
IF( NRAYS .GT. KTEMP ) CALL ERROR2( 36H NRAYS .GT. TOTAL NO. OF RD306200
1AYS IN LATT, KTEMP ) RD306300 244
3000 IF( IFLAG(2) .EQ. 0 ) GO TO 5000 RD307000

C
C READ DESIGN INPUT RD401000
C

```

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
 READS2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

READ (5, DESIGN)	RD401200	
C		
IF(ZETA .EQ. 0.0) CALL ERROR(36H DESIGN SCALE FACTOR IS 0.0	RD401210	248
1)	RD401220	251
IF(DUMIN .EQ. 0.0) CALL ERROR(36H DUMIN. MINIMUM DELTA U, IS	RD401230	
10.0)	RD401240	254
IF(NDSGN(1) .LT. 0 .OR. NDSGN(1) .GT. 50 .OR. NDSGN(2) .LE. 0	RD401250	
1 .OR. NDSGN(3) .GT. NDSGN(1) .OR. NDSGN(4) .LE. 0) CALL ERROR	RD401260	
2 (36H A VALUE IN NDSGN IS INCORRECT)	RD401270	257
IF(NDSGV(1) .LT. 1 .OR. NDSGV(1) .GT. 10) GO TO 3040	RD401280	
ITEMP = NDSGV(1) + 1	RD401290	
DO 3020 I = 2, ITEMP	RD401300	
IF(NDSGV(1) .LT. NDSGN(2) .OR. NDSGV(1) .GT. NDSGN(3)) GO TO	RD401310	
1 3040	RD401320	
IF(I .EQ. ITEMP) GO TO 3090	RD401330	
L = I + 1	RD401340	
DO 3020 J = L, ITEMP	RD401350	
IF(NDSGV(1) .EQ. NDSGV(J)) GO TO 3040	RD401360	
3020 CONTINUE	RD401370	
GO TO 3090	RD401400	
3040 CALL ERROR(36H A VALUE IN NDSGV IS INCORRECT)	RD401410	288
3090 IF(NCOND .LT. 0 .OR. NCOND .GT. 100) CALL ERROR2(36H NUMBER	RD401450	
1 OF BNDRY. CONDITIONS ILLEGAL , NCOND)	RD401460	291
ITEMP = NDSGN(1)	RD401500	
C CONVERT NUMBER OF DESIGN PARAMETERS TO INTEGER	RD401600	
DO 3125 I=1,ITEMP	RD401700	
IF(IABS(IDESGN(1,I)) .GT. 10000) IDESGN(1,I) = DESGN(1,I)	RD401800	
C CHECK FOR SYMBOL CARDS	RD401810	
IF(IFLAG(2) .EQ. 2) GO TO 3125	RD401815	
DO 3100 J = 1, 12	RD401820	
SDESN(J,I) = BLANK	RD401850	
3100 CONTINUE	RD401900	
3125 CONTINUE	RD401910	
IF(IFLAG(2) .EQ. 2) GO TO 3802	RD401915	
C READ PARAMETER SYMBOLS TO BE MODIFIED DURING DESIGNING	RD402500	
DO 3800 I = 1, ITEMP	RD402600	
JTEMP = IABS(IDESGN(1,I))	RD402700	
KTEMP = 2 * JTEMP	RD402750	
READ (5, 3210) ISIGN(1), BUFFER(1), BUFFER(2), ISIGN(2),	RD402800	
1 BUFFER(3), BUFFER(4), ISIGN(3), BUFFER(5), BUFFER(6), ISIGN(4),	RD402820	
2 BUFFER(7), BUFFER(8), ISIGN(5), BUFFER(9), BUFFER(10), ISIGN(5),	RD402840	
3 BUFFER(11), BUFFER(12)	RD402860	321
3210 FORMAT(6(A1, 2A6))	RD402950	
C STORE DESIGN SYMBOLS FOR PRINT OUT	RD403000	
DO 3150 M = 1, KTEMP	RD403100	
3150 SDEN(M,I) = BUFFER(M)	RD403200	
DO 3800 J = 1, JTEMP	RD403500	
C TEST FOR INTEGER VARIABLE	RD403700	
L = 2 * (J - 1) + 1	RD403600	
DO 3200 K = 1, NSYMB1	RD403800	
IF(BUFFER(L) .NE. SYMBL1(1,K)) GO TO 3200	RD403900	
IDESGN(J+3,I) = SYMBL1(2,K)	RD404400	
GO TO 3800	RD404500	
3200 CONTINUE	RD404600	
C TEST FOR REAL VARIABLE	RD405000	
DO 3300 K = 1, NSYMB2	RD405100	

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
READSZ - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

```

IF( BUFFER(L) .NE. SYMBL2(1,K) ) GO TO 3300          RD405200
IDESGN(J+3,1) = SYMBL2(2,K)                          RD405300
GO TO 3600                                           RD405400
3300 CONTINUE                                         RD405500
C TEST FOR INTEGER VECTOR                            RD405800
DO 3325 K = 1, NSYMB3                                RD405900
IF( BUFFER(L) .NE. SYMBL3(1,K) ) GO TO 3325          RD406000
CALL VCNVRT( BUFFER(L+1), IDEX, $3350 )              RD406100    362
IDESGN(J+3,1) = SYMBL3(2,K) + IDEX - 1              RD406200
GO TO 3600                                           RD406300
3325 CONTINUE                                         RD406400
GO TO 3450                                           RD406500

C PRINT ERROR MESSAGE AND SUBSCRIPT IF THE CORRECT SUBSCRIPT
C CAN NOT BE EVALUATED                               RD406700
3350 CALL ERROR1( 36H ILLEGAL SUBSCRIPT IN DESIGN DATA , BUFFER(L+1)) RD406800
C TEST FOR REAL VECTOR                               RD406900
3450 DO 3500 K = 1, NSYMB4                            RD407000    373
IF( BUFFER(L) .NE. SYMBL4(1,K) ) GO TO 3500          RD407100
CALL VCNVRT( BUFFER(L+1), IDEX, $3350 )              RD407200
IDESGN(J+3,1) = SYMBL4(2,K) + IDEX - 1              RD407300    383
GO TO 3600                                           RD407400
3500 CONTINUE                                         RD407500
C TEST FOR REAL MATRIX                               RD407600
DO 3600 K = 1, NSYMB5                                RD408000
IF( BUFFER(L) .NE. SYMBL5(1,K) ) GO TO 3600          RD408100
CALL MCVNVRT( BUFFER(L+1), IDEX, JDEX, $3350 )       RD408200
LTEMP = 3                                             RD408300    399
IF( K .EQ. 2 ) LTEMP = 20                            RD408400
IF( K .EQ. 3 ) LTEMP = 10                            RD408500
IDESGN(J+3,1) = SYMBL5(2,K) + LTEMP * ( JDEX - 1 ) + IDEX - 1 RD408600
GO TO 3600                                           RD408700
3600 CONTINUE                                         RD408800
C DESIGN SYMBOL HAS NOT BEEN IDENTIFIED              RD409200
CALL ERROR1( 36H ILLEGAL SYMBOL IN DESIGN DATA , BUFFER(L) ) RD409300
C MINUS INDEX INDICATES NEGATIVE DESIGN INCREMENT  RD409400    413
3800 IF( ISIGN(J) .EQ. MINUS ) IDESGN(J+3,1) = - IDESGN(J+3,1) RD409500
3802 IF( IFLAG(6) .EQ. 0 ) GO TO 5000                RD451000

C READ BOUNDARY INPUT                                RD451200
C READ ( 5, BNDRY )                                  RD451400
C CHECK FOR SYMBOL CARDS                              RD451410    426
IF( IFLAG(6) .EQ. 2 ) GO TO 5000                    RD451415
ITEMP = NCOND / 6                                    RD451600
IF( MOD( NCOND, 6 ) .NE. 0 ) ITEMPT = ITEMPT + 1    RD451800
MTEMP = 0                                            RD452000
DO 4800 I = 1, ITEMPT                                RD452200
READ ( 5, 32135 ) ( BUFFER(J), J = 1, 12 )          RD452400    437
JTEMP = 12                                           RD452600
IF( I .EQ. ITEMPT .AND. MOD( NCOND, 6 ) .NE. 0 ) JTEMP = 2 * RD452800
MOD( NCOND, 6 )                                       RD453000
DO 4800 J = 1, JTEMP, 2                               RD453200
MTEMP = MTEMP + 1                                    RD453400

```

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
READS2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE 1

C	STORE BOUNDARY SYMBOL FOR PRINT OUT	RD453600	
	BDYSYM(1,MTEMP) = BUFFER(J)	RD453800	
	BDYSYM(2,MTEMP) = BUFFER(J+1)	RD454000	
C	TEST FOR REAL VARIABLE	RD454200	
	DO 4000 K = 1, NSYMB2	RD454400	
	IF(BUFFER(J) .NE. SYMBL2(1,K)) GO TO 4000	RD454600	
	IBNDS(1,MTEMP) = SYMBL2(2,K)	RD454800	
	GO TO 4800	RD455000	
4000	CONTINUE	RD455200	
C	TEST FOR REAL VECTOR	RD455400	
	DO 4200 K = 1, NSYMB4	RD455500	
	IF(BUFFER(J) .NE. SYMBL4(1,K)) GO TO 4200	RD455600	
	CALL VCNVRT(BUFFER(J+1), IDEX, \$4500)	RD455700	471
	IBNDS(1,MTEMP) = SYMBL4(2,K) + IDEX - 1	RD455800	
	GO TO 4800	RD455900	
4200	CONTINUE	RD456000	
C	TEST FOR REAL MATRIX	RD456200	
	DO 4400 K = 1, NSYMB5	RD456300	
	IF(BUFFER(J) .NE. SYMBL5(1,K)) GO TO 4400	RD456400	
	CALL MCVRT(BUFFER(J+1), IDEX, JDEX, \$4500)	RD456500	488
	LTEMP = 3	RD456600	
	IF(K .EQ. 3) LTEMP = 30	RD456700	
C			
C	TEST FOR INDEX OF REFRACTION	RD456800	
	IF(K .NE. 2) GO TO 4300	RD456900	
	LTEMP = 20	RD457000	
	IF(IDEX .LT. 5 .OR. IDEX .GT. 10) GO TO 4300	RD457100	
C	PARAMETER IS AN INDEX OF REFRACTION, USE NEGATIVE INDEX	RD457400	
	IF(IDEX .NE. 5) CALL ERROR4(36H ILLEGAL INDEX OF REFR. IN BOUNDARY)	RD457500	503
	IBNDS(1,MTEMP) = - (SYMBL5(2,K) + LTEMP * (JDEX-1) + 4)	RD457600	
	GO TO 4800	RD457800	
4300	IBNDS(1,MTEMP) = SYMBL5(2,K) + LTEMP * (JDEX-1) + IDEX - 1	RD457900	
	GO TO 4800	RD458000	
4400	CONTINUE	RD458200	
C			
C	BOUNDARY SYMBOL HAS NOT BEEN IDENTIFIED	RD458400	
	CALL ERROR1(36H ILLEGAL SYMBOL IN BOUNDARY DATA , BUFFER(J))	RD458600	515
4500	CALL ERROR1(36H ILLEGAL SUBSCRIPT IN BOUNDARY DATA , BUFFER(J+1))	RD459000	518
4800	CONTINUE	RD459200	
5000	IF(IFLAG(3) .EQ. 0) GO TO 8000	RD501000	
C			
C	READ SUBSTITUTION INPUT	RD501200	
C			
	READ (5, SUBSTN)	RD501250	525
	IF((NSUBT+1) * NSUDP .GT. 500)	RD501260	
	1 CALL ERROR2(36H SUBSTITUTION LIMITS ARE INCORRECT , NSUBT)	RD501270	
C	CHECK FOR SYMBOL CARDS	RD501275	528
	IF(IFLAG(3) .EQ. 2) GO TO 8000	RD501280	
	ITEMP = NSUDP / 6	RD501300	
	KTEMP = NSUDP ÷ NSUBT + 1	RD501400	
	MTEMP = 1	RD501450	
	IF(MOD(NSUDP, 6) .NE. 0) ITEMP = ITEMP + 1	RD501500	
	DO 6000 I = 1, ITEMP	RD502000	
	READ(5, 32130) (BUFFER(J), J = 1, 12)	RD502200	540
32130	FORMAT(6(1X, 2A))	RD502250	

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
 READS2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

JTEMP = 12	RD502300	
IF (1 .EQ. ITEMP .AND. MOD(NSUBP, 6) .NE. 0) JTEMP = 2 *	RD502400	
1 MOD(NSUBP, 6)	RD502500	
DO 6000 J = 1, JTEMP, 2	RD502600	
C STORE SUBSTITUTION SYMBOLS FOR PRINT OUT	RD502700	
SUBSYM(1,MTEMP) = BUFFER(J)	RD502750	
SUBSYM(2,MTEMP) = BUFFER(J+1)	RD502800	
MTEMP = MTEMP + 1	RD502850	
C TEST FOR INTEGER VARIABLE	RD503000	
DO 5200 K = 1, NSYMB1	RD503100	
IF (BUFFER(J) .NE. SYMBL1(1,K)) GO TO 5200	RD503200	
ISUBST(KTEMP) = - SYMBL1(2,K)	RD503300	
GO TO 6000	RD503400	
5200 CONTINUE	RD503500	
C TEST FOR REAL VARIABLE	RD504000	
DO 5300 K = 1, NSYMB2	RD504100	
IF (BUFFER(J) .NE. SYMBL2(1,K)) GO TO 5300	RD504200	
ISUBST(KTEMP) = SYMBL2(2,K)	RD504300	
GO TO 6000	RD504400	
5300 CONTINUE	RD504500	
C TEST FOR INTEGER VECTOR	RD505000	
DO 5350 K = 1, NSYMB3	RD505100	
IF (BUFFER(J) .NE. SYMBL3(1,K)) GO TO 5350	RD505200	
CALL VCNVRT(BUFFER(J+1), IDEX, \$5400)	RD505300	586
ISUBST(KTEMP) = - SYMBL3(2,K) - IDEX + 1	RD505400	
GO TO 6000	RD505500	
5350 CONTINUE	RD505600	
GO TO 5550	RD505700	
C PRINT ERROR MESSAGE AND SUBSCRIPT IF THE CORRECT SUBSCRIPT	RD506000	
C CAN NOT BE EVALUATED	RD506100	
5400 CALL ERROR1(36H ILLEGAL SUBSCRIPT IN SUBSTITN. DATA, BUFFER(J+1)	RD506400	
C TEST FOR REAL VECTOR	RD506500	598
5550 DO 5600 K = 1, NSYMB4	RD506600	
IF (BUFFER(J) .NE. SYMBL4(1,K)) GO TO 5600	RD506700	
CALL VCNVRT(BUFFER(J+1), IDEX, \$5400)	RD506800	608
ISUBST(KTEMP) = SYMBL4(2,K) + IDEX - 1	RD507000	
GO TO 6000	RD507100	
5600 CONTINUE	RD507200	
C TEST FOR REAL MATRIX	RD507500	
DO 5700 K = 1, NSYMB5	RD507600	
IF (BUFFER(J) .NE. SYMBL5(1,K)) GO TO 5700	RD507800	
CALL MCVNVRT(BUFFER(J+1), IDEX, JDEX, \$5400)	RD507900	624
LTEMP = 3	RD508000	
IF (K .EQ. 2) LTEMP = 20	RD508100	
IF (K .EQ. 3) LTEMP = 10	RD508200	
ISUBST(KTEMP) = SYMBL5(2,K) + LTEMP * (JDEX - 1) + IDEX - 1	RD508500	
C CONVERT NO. RAYS AND SURFACE TYPE TO INTEGER NEG. INDEX	RD508520	
IF ((K .EQ. 1 .AND. IDEX .EQ. 3) .OR. (K .EQ. 2 .AND. IDEX	RD508540	
1 .EQ. 1)) ISUBST(KTEMP) = - ISUBST(KTEMP)	RD508560	
GO TO 6000	RD508600	
5700 CONTINUE	RD508700	
C SUBSTITUTION SYMBOL HAS NOT BEEN IDENTIFIED	RD509000	
CALL ERROR1(36H ILLEGAL SYMBOL IN SUBSTITUTION DATA, BUFFER(J))	RD509100	644
6000 KTEMP = KTEMP + 1	RD509200	

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
 READS2 - EFN SOURCE STATEMENT - IFN(5) -

09/19/68

PAGE

```

C
C          CONVERT SUBSTITUTION VALUES TO INTEGER
C          KTEMP = NSUCF + NSUDT + 1
C          DO 6200 I = 1, NSUCF
C          IF( ISUBST( KTEMP ) .GT. 0 ) GO TO 6200
C          ITEMP = I
C          DO 6100 J = 1, NSUDT
C          IF( ISUBST(ITEMP) .GT. 10000 ) ISUBST(ITEMP) = SUBST(ITEMP)
6100 ITEMP = ITEMP + NSUCF
6200 KTEMP = KTEMP + 1
8000 IF( IFLAG(4) .EQ. 0 ) GO TO 12000
C
C          READ PERTURBATION DATA
C
C          READ ( 5, SENST)
C          IF( NPERTB .LT. 1 .OR. NPERTB .GT. 30 ) CALL ERROR2( 36 H NUMBER
C          1R OF PERTURBATIONS IS ILLEGAL , NPERTB )
C CHECK FOR SYMBOL CARDS
C          IF( IFLAG(4) .EQ. 2 ) GO TO 12000
C          ITEMP = NPERTB / 6
C          MTEMP = 1
C          IF( MOD( NPERTB, 6 ) .NE. 0 ) ITEMP = ITEMP + 1
C
C          READ ITEM DATA CARDS CONTAINING SYMBOLS TO
C          BE PERTURBED
C          DO 10000 I = 1, ITEMP
C          READ ( 5, 32130 ) ( BUFFER(J), J = 1, 12 )
C          JTEMP = 12
C          IF( I .EQ. ITEMP .AND. MOD( NPERTB, 6 ) .NE. 0 ) JTEMP = 12 *
C          1 MOD( NPERTB, 6 )
C          DO 10000 J = 1, JTEMP, 2
C
C          STORE PERTURBATION SYMBOLS FOR PRINT OUT
C          IPERTB(MTEMP,3) = BUFFER(J)
C          IPERTB(MTEMP,4) = BUFFER(J+1)
C
C          TEST FOR REAL VARIABLE
C          DO 8200 K = 1, NSYMB2
C          IF( BUFFER(J) .NE. SYMBL2(1,K) ) GO TO 8200
C          IPERTB(MTEMP,2) = SYMBL2(2,K)
C          GO TO 10000
8200 CONTINUE
C
C          TEST FOR REAL VECTOR
C          DO 8400 K = 1, NSYMB4
C          IF( BUFFER(J) .NE. SYMBL4(1,K) ) GO TO 8400
C          CALL VCNVRT( BUFFER(J+1), IDEX, 89000 )
C          IPERTB(MTEMP,2) = SYMBL4(2,K) + IDEX - 1
C          GO TO 10000
8400 CONTINUE
C
C          TEST FOR REAL MATRIX
C          DO 8600 K = 1, NSYMB5
C          IF( BUFFER(J) .NE. SYMBL5(1,K) ) GO TO 8600
C          CALL MCNVRT( BUFFER(J+1), IDEX, JDEX, 89000 )
C          LTEMP = 3

```

RD509350
 RD509400
 RD509450
 RD509500
 RD509550
 RD509600
 RD509650
 RD509700
 RD509750
 RD601000

RD601500

RD602000 675

RD602200

RD602400

RD602410 678

RD602420

RD602600

RD602800

RD603000

RD603200

RD603400

RD603600

RD603800 689

RD604000

RD604200

RD604400

RD605000

RD605200

RD605400

RD605600

RD606000

RD606200

RD606400

RD606600

RD606800

RD607000

RD607400

RD607500

RD607600

RD607700 722

RD607800

RD607900

RD608000

RD608400

RD608500

RD608600

RD608700 738

RD608800

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
 READS2

09/19/68

PAGE 1

STORAGE MAP

SUBROUTINE READS
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
CONTROL	00000	I	CONTROL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCI0	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSIDP	00041	I	NIFLN	00042	I	IMODE	00043	I
NS:LN	00044	I	NDDJH	00045	I	NSURF	00046	I
AFLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXFF	00055	R	DEXFF	00056	R	WEXFF	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CD1ST	00071	R	ND	00072	R	DELH	00073	R
BSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WDDJH	00117	R	WCLRH	00126	R	WIMHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IDNS	06210	I	NCOND	06665	I	ATRCCR	06666	I
GAUSS	06667	R	ITNPT	00047	I			

COMMON BLOCK			SYMBOLS	ORIGIN	06672	LENGTH	02424	
SCEN	00000	I	SUBSYM	01130	I	BDYSYM	02114	I
COMMON BLOCK			FRNT	ORIGIN	11316	LENGTH	00002	
LINE	00000	I	PAGE	00001	I			
COMMON BLOCK			PERTB	ORIGIN	11320	LENGTH	00172	
PERTB	00000	R	NPERTB	00170	I	REFOCS	00171	I
IPERTB	00000	I						
COMMON BLOCK			PLOTG	ORIGIN	11512	LENGTH	00003	
MAXX	00000	R	DZMIN	00001	R	NPTG	00002	I

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
FLAG	11515	I	ISGN	11523	I	SYMBL1	11531	I
SYMBL2	11563	I	SYMBL3	11645	I	SYMBL4	11651	I
SYMBL5	11667	I	BUFFER	11675	I	BUFFRR	11711	R
SYHED	11717	I						

UNDIMENSIONED PROGRAM VARIABLES

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
READS?

09/19/68

PAGE 16

SYMBOL	LOCATION	TYPE
CLEAR	11727	I
J	11732	I
JTEMP	11735	I
L	11740	I
NSYMB3	11743	I
NSYMB5	11746	I
MINUS	11751	I

STORAGE MAP

SYMBOL	LOCATION	TYPE
BLANK	11730	I
K	11733	I
JTEMP	11736	I
NSYMB1	11741	I
IDEX	11744	I
JDEX	11747	I
MTEMP	11752	I

SYMBOL	LOCATION	TYPE
I	11731	I
KTEMP	11734	I
IDX	11737	I
NSYMB2	11742	I
NSYMB4	11745	I
LTEMP	11750	I

ENTRY POINTS

READS SECTION 13

.FRDU.	SECTION	14
.FWRD.	SECTION	17
ERROR3	SECTION	20
MCONVRT	SECTION	23
.FRTN.	SECTION	26
.FFIL.	SECTION	29
CC.3	SECTION	32

SUBROUTINES CALLED

.FRDD.	SECTION	15
ERROR2	SECTION	18
VCONVRT	SECTION	21
ERROR4	SECTION	24
.FCNV.	SECTION	27
CC.1	SECTION	30
CC.4	SECTION	33
EFN	IFN	CORRESPONDENCE

PRTCTL	SECTION	1
ERROR	SECTION	1
ERROR1	SECTION	2
.UNGS.	SECTION	2
.UNGS6.	SECTION	2
CC.2	SECTION	3
SYSLOC	SECTION	3

EFN	IFN	LOCATION
1500	40A	13171
32100	FORMAT	12403
1600	43A	13175
32050	FORMAT	12412
2010	73A	13300
32065	FORMAT	12446
2025	82A	13334
32080	FORMAT	12502
2040	102A	13417
32091	FORMAT	12535
2055	118A	13471
32094	FORMAT	12570
2290	131A	13536
32096	FORMAT	12644
2525	163A	14007
2550	200A	14051
2800	224A	14125
3040	287A	14345
3125	310A	14431
3800	415A	15074
3200	341A	14634
3350	372A	14730
3600	411A	15061
4000	462A	15262
4400	512A	15440
6000	445A	16136
5350	593A	15754

EFN	IFN	LOCATION
1200	10A	13115
32110	FORMAT	12404
32115	FORMAT	12407
2005	72A	13272
32060	FORMAT	12434
2020	78A	13321
32075	FORMAT	12471
2035	94A	13373
32090	FORMAT	12524
2050	114A	13456
32093	FORMAT	12557
2210	123A	13512
32210	FORMAT	12612
3000	246A	14202
2650	216A	14107
2750	222A	14120
2850	240A	14166
3020	282A	14340
3100	308A	14427
32120	FORMAT	12651
3300	353A	14663
3450	375A	14740
4800	519A	15463
4200	478A	15322
4300	508A	15426
5200	565A	15672
5400	596A	15760

EFN	IFN	LOCATION
1400	19A	13130
2000	65A	13246
1800	57A	13232
32055	FORMAT	12422
2015	77A	13313
32070	FORMAT	12457
2030	90A	13360
32085	FORMAT	12512
2045	113A	13450
32092	FORMAT	12546
2200	119A	13477
32200	FORMAT	12600
32095	FORMAT	12624
2500	173A	13766
2700	220A	14113
2600	207A	14065
5000	523A	15467
3090	289A	14351
3802	424A	15100
3150	324A	14561
3325	369A	14724
3500	390A	14776
32130	FORMAT	12654
4500	516A	15453
8000	673A	16230
5300	577A	15716
5550	599A	15770

INPUT ROUTINE (CASE DECK CARD PROCESSOR)
 READS2

09/19/68

PAGE

				STORAGE	MAP			
9500	615A	16023	5700	641A	16123	6200	609A	1622
6100	656A	16216	12000	762A	16563	10000	757A	1655
8200	713A	16413	8400	729A	16451	9000	754A	1654
8000	750A	16531	32000	775A	16633			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 16653.

TR-67-700-10-2
Page III-1-22
Program Listing
Link 1
Page Revised 10/1/68

INF,NO10001,44903-0,77307,AM IC 10000 L10MSF 7094 A 09/19/68 PAGE

84
BTITLE CARD IMAGE LIST ROUTINE
SIDFTC REREZZ M94,XR7 LINK 1 (REREAD) RR001000

CARD IMAGE LIST ROUTINE
REREZZ - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

```

SUBROUTINE REREAD                                RR001500
C
C *** DESCRIPTION ***                            RR001502
C THE PURPOSE OF REREAD IS TO PRODUCE A CARD IMAGE LIST OF THE RR001504
C CURRENT CASE DECK. REREAD READS CARD IMAGES FROM THE SYSTEM RR001506
C INPUT FILE (LOGICAL 5) IN AN 80A1 FORMAT UNTIL AN END OF CASE RR001508
C CARD IS ENCOUNTERED. ALL CARD IMAGES (EXCEPT OCTAL) ARE PRINT RR001510
C ED---ONE CARD PER LINE USING AN 80A1 FORMAT. REREAD RE-POSITION RR001512
C NS THE INPUT FILE BY BACKSPACING OVER THE CARD RECORDS WHICH RR001514
C WERE READ.                                     RR001516
C
C
C DIMENSION BUFFER(80), TEST(11), CHAR(8), OEND(3) RR002000
C INTEGER OFLAG                                   RR002200
C
C DATA TEST / 6HE , 6HN , 6HD , 6H , 6HD , RR002400
C 1 6HF , 6H , 6HC , 6HA , 6HS , 6HE / RR002500
C
C DATA CHAR / 6HG , 6H1 , 6H2 , 6H3 , 6H4 , RR002600
C 1 6H5 , 6H6 , 6H7 / RR002800
C
C DATA OEND / 6H5 , 6H8 , 6H9 / RR003000
C
C START OF SUBROUTINE REREAD                      RR003500
C
C WRITE ( 6, 32000 )                               RR004000 1
32000 FORMAT( 1H1, 5X, 34HLENS DESIGN PROGRAM INPUT LISTING ) RR004500
C IBACKS = 1                                       RR005000
C OFLAG = 0                                         RR005200
C 1000 READ ( 5, 32020 ) ( BUFFER(J), J = 1, 80 ) RR005500 3
32020 FORMAT( 80A1 )                                RR006000
C DO 1500 I = 1, 11                                RR007000
C IF( BUFFER(I) .NE. TEST(I) ) GO TO 2000          RR007200
1500 CONTINUE                                       RR007400
C
C TEST FOR END OF CASE FOR PREVIOUS DATA CASE. RR008000
C IF( IBACKS .LE. 1 ) GO TO 1000                  RR008500
C GO TO 3000                                       RR009000
2000 IF( OFLAG .NE. 0 ) GO TO 2800                RR101000
C
C TEST FOR OCTAL DATA BY COMPARING INPUT WITH OCTAL CHARS. RR101100
C DO 2200 L = 1, 72                                RR101200
C DO 2100 M = 1, 8                                  RR101400
C IF( BUFFER(L) .EQ. CHAR(M) ) GO TO 2200          RR101600
2100 CONTINUE                                       RR101800
C GO TO 2800                                       RR102000
2200 CONTINUE                                       RR102200
C
C OCTAL DATA PRESENT                              RR103000
C OFLAG = 1                                         RR103200
C IBACKS = IBACKS + 1                              RR103250
C ICHT = 1                                          RR103300
2300 DO 2350 K = 1, 3                               RR103350
C IF( BUFFER( K + 77 ) .NE. OEND(K) ) GO TO 2400 RR103400
2350 CONTINUE                                       RR103450

```

CARD IMAGE LIST ROUTINE
 REREZ2 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE 2

GO TO 2500	RR103500	
2400 READ (5, 32020) (BUFFER(J), J = 1, 80)	RR103600	59
IBACKS = IBACKS + 1	RR103800	
ICNT = ICNT + 1	RR104000	
GO TO 2300	RR104200	
2500 WRITE(6, 32060) CNT	RR104400	60
32060 FORMAT(1H0, 5X, 20 **** OCTAL DATA INPUT ****, 14, 19H CARDS IN C	RR104600	
1ECC ****)	RR104700	
GO TO 1000	RR104800	
C		
C PRINT DATA CARD	RR201000	
2800 WRITE (6, 32040) (BUFFER(J), J = 1, 80)	RR201500	70
32040 FORMAT(1H0, 10X, 80A1)	RR202000	
IBACKS = IBACKS + 1	RR202500	
GO TO 1000	RR203000	
C		
C PRINT END OF CASE CARD	RR203500	
3000 WRITE (6, 32040) (BUFFER(J), J = 1, 11)	RR204000	
C		
C BACKSPACE DATA INPUT	RR204500	78
DO 4000 I = 1, IBACKS	RR205000	
BACKSPACE 5	RR205500	87
4000 CONTINUE	RR206000	
C		
C END OF REREAD SUBROUTINE	RR206500	
RETURN	RR208000	
END	RR208500	

CARD IMAGE LIST ROUTINE
 REREZ2

09/19/68

PAGE 3

STORAGE MAP

SUBROUTINE REREAD
 DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE
BUFLR	00001	R
GENC	00144	R

SYMBOL	LOCATION	TYPE
TEST	00121	R

SYMBOL	LOCATION	TYPE
CHAR	00134	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE
OFLAG	00147	I
L	00152	I
K	00155	I

SYMBOL	LOCATION	TYPE
IBACKS	00150	I
M	00153	I

SYMBOL	LOCATION	TYPE
I	00151	I
ICNT	00154	I

ENTRY POINTS

REREAD	SECTION	
	3	

SUBROUTINES CALLED

	SECTION	
.FWRD.	4	
.UN06.	7	
.UN05.	10	

	SECTION	
.FRDD.	5	
.FFIL.	8	
.FRTN.	11	
EFN	IFN	CORRESPONDENCE

	SECTION	
.FBST.	6	
.FCNV.	9	
SYSLOC	1	

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	FORMAT	00165	1000	3A	00230	32020	FORMAT	00176
1500	18A	00252	2000	25A	00262	3000	78A	00403
2000	70A	00364	2200	41A	00306	2100	38A	00303
2300	46A	00317	2350	55A	00326	2400	59A	00331
2500	68A	00353	32060	FORMAT	00200	32040	FORMAT	00214
4000	88A	00426						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00451.

TR-67-700-10-2
Page III-1-26
Program Listing
Link 1

INF.ND10001.44800-0.77307.AM IC

ISJOB V13HJD 7094 A 11/08/67

PAGE 17

04
BTITLE GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
SIBPTC STAR9Z M94.XR7 LINK 1 (STAR9)

89001000

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
 STARDZ - EFM SOURCE STATEMENT - IFN(8) -

11/08/67

PAGE 1

```

SUBROUTINE STARDZ                                     89001200
C
C *** DESCRIPTION ***                                89001202
C STARDZ LISTS ALL OF THE GEOMETRY INPUT DATA WITH DESCRIPTIVE 89001204
C TITLES. IF NSUBST.NE.O THE SUBSTITUTION INPUT DATA IS ALSO 89001208
C LISTED IN A TABULAR FORM BY PARAMETER AND BY SET.      89001209
C
COMMON / DATA / MCNTRL, CNTRL(100), TITLE(12), DATE(3), PUNCID,  DT001000
1 DATA(3483)                                           DT001100
DIMENSION MOBJN(7), WCLRH(7), WINHT(7), EIMHT(7), CIMPL(7),  DT002000
1 WCLR8(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100),  DT002100
2 SURFC(20,100), DESGN(10,500), SUBST(500), ILATT(3,26),  DT002200
3 ISURFC(20,100), IDESGN(10,500), ISUBST(500), IBNDS(3,100)  DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3),  DT101000
1 NSLCS), (DATA(4), NCLCS), (DATA(5), NJAIL), (DATA(6),  DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9),  DT101200
3 INODE), (DATA(10), NSPLN), (DATA(11), NOBJN), (DATA(12),  DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15),  DT101400
5 FNUMB), (DATA(16), FLNEN), (DATA(17), WFLGH), (DATA(18),  DT101500
6 ZETA), (DATA(19), MEXPP), (DATA(20), DEXPP), (DATA(21),  DT101600
7 MEXPP), (DATA(22), DLPLN), (DATA(23), OMSA2), (DATA(24),  DT101700
8 OMSA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27),  DT101800
9 PSCAL), (DATA(28), OMSAF), (DATA(29), SPFEA), (DATA(30),  DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), MO), (DATA(33),  DT102000
B DELN), (DATA(34), SYMXX), (DATA(35), WDIR), (DATA(36),  DT102100
C WDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42),  DT102200
D NDSGV(1)), (DATA(53), MOBJN(1)), (DATA(60), WCLRH(1)),  DT102300
E (DATA(67), WINHT(1)), (DATA(74), EIMHT(1)), (DATA(81),  DT102400
F CIMPL(1)), (DATA(88), WCLR8(1)),  DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1),  DT102600
H ISURFC(1,1)), (DATA(2102), DESGN(1,1), IDESGN(1,1)),  DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1),  DT102800
J IBNDS(1,1)), (DATA(3483), NCOND)  DT102900
INTEGER CNTRL, AFLAG, DATE  DT201000
REAL LATTC  DT202000
C
C THIS BLOCK OF COMMON CONTAINS THE SYMBOLS USED IN DESIGN 8Y001000
C AND SUBSTITUTION - UTILIZED FOR PRINT OUT 8Y002000
COMMON / SYMBS / SDESN(12,50), SUBSYN(2,250), BDYSYN(2,100) 8Y003000
INTEGER SDESN, SUBSYN, BDYSYN 8Y004000
C
COMMON / PRNT / LIME, PAGE  PT002500
INTEGER PAGE  PT002700
C
DATA SURFA / SHPLANE, SHSPHERE, SHPARABO, SHLOID, SHELLIPS, 89002000
1 SHOID, SHHYPERB, SHOLOID, SHHORIZO, SHNTAL, SHVERTIC, 89002100
2 SHAL, SHCIRCUL, SHAR CYL, SHINDER, SHPARABO, SHLIC CY, 89002200
3 SHLINDER, SHELLIPT, SHIC CYL, SHINDER, SHHYPERB, SHLIC C, 89002300
4 SHYLINDE, SHR, SHTRANSL, SHATED/T, SHILTED, SHASPHER, 89002400
5 SHIC OF, SHREVOLU, SHTION, SHHORIZO, SHNTAL A, SHSPHERI, 89002500
6 SHC CYLI, SHNDER, SHVERTIC, SHAL ASP, SHERIC, SHCYLING, 89002600
7 SHER, SHIDFOR, SHMED PL, SHANE), SHIDFOR, SHMED SP, 89002700
8 SHHERE), SHIDFOR, SHMED PA, SHRABOLO, SHID), SHIDFOR, 89002800
9 SHMED EL, SHLIPSOI, SHD), SHIDFOR, SHMED HY, SHPERBOL, 89002900
A SHOID), SHIDFOR, SHMED CI, SHRCLE), SHIDFOR, SHMED PA, 89003000

```

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
STAR92 - EPN SOURCE STATEMENT - IPN(8) -

11/09/67

PAGE 8

```

B GHRABOLA, GH)      , GHIDEPOR, GHMED EL, GHLIPSE), GHIDEPOR,      89003100
C GHMED HY, GHPERDOL, GHA) /                                       89003200
DIMENSION SURFA(74), PLINE(9), PNT(9)                               89004000
DATA BLANK / GH /                                                  89006000
DATA PNT(1) / GH(1H .I / , PNT(2) / GH(10.3X, / , PNT(9) / GH I2 ) / 89008500
DATA IPNT / GH(17.7, / , IPNT / GH(17.10X / , SPNT / GH I2 ) /    89008000
REAL IPNT                                                            89008100

C
C      START OF STAR9 SUBROUTINE                                     89101000
C
CALL PRCTCL                                                           89101200
WRITE ( 6, 32000 ) MO, DELH, NOBJH, ROTAN, ODIST, NCLRS, INODE,      89101500
1   DELD, ( CIMPL(I), I=1, NOBJH )                                   89101600
32000 FORMAT( 1HD, 53H OBJECT HEIGHT..... MO89102000
1, 6X, E14.7 / 53H OBJECT HEIGHT INCREMENT..... 89102100
2DELH, 4X, E14.7 / 54H NUMBER OF OBJECT POINTS..... 89102200
3... NOBJH, 15 / 54H AZINUTH OF OBJECT..... 89102300
4. ROTAN, E17.7 / 54H DISTANCE OBJECT TO SURFACE 1..... 89102400
5... ODIST, E17.7 // 54H NUMBER OF COLORS..... 89102500
6... NCLRS, 15 // 54H MODE NUMBER FOR PRINCIPAL PLANE..... 89102600
7... INODE, 15 // 53H INCREMENTS TO SPECIFY CURVED IMAGE..... 89102700
9DELD, E18.7 / 54H INCREMENTS TO SPECIFY CURVED IMAGE..... 89102900
ACTMPL, 1X, 4E18.7 / 55X, 3E18.7 )                                  89103000

C
WRITE ( 6, 32010 ) NIPLN, NSPLN, DLPLN, LMODE, NRAYS, DELY,        89103500
1 NSLCS                                                            89103600
32010 FORMAT( 1H , 53HNUMBER OF IMAGE PLANES..... NI89104000
1PLN, 15 / 54H POSITION OF FIRST IMAGE PLANE..... NSPL89104100
2N, 15 / 54H SPACING OF IMAGE PLANES..... DLPLN,89104200
3 E17.7 // 54H LATTICE MODE..... LMO89104300
4E, 15 / 54H NUMBER OF RAYS..... NRAYS,89104400
5 15 / 53H VERTICAL INCREMENT..... DELY, 89104500
6 E18.7 / 54H NUMBER OF SLICES..... NSLCS89104600
7, 15 / 54H RAY PATTERN..... LATTC 89104700
8 // 9X, 5HSLICE, 9X, 1HX, 15X, 1HY, 9X, 11HNO. OF RAYS )        89104800

C
DO 1000 I = 1, NSLCS                                                89105000
WRITE ( 6, 32020 ) I, ( LATTC(J,I), J = 1, 3 )                       89105100
32020 FORMAT( 1H , 112, E17.7, E18.7, 18 )                          89105200
1000 CONTINUE                                                        89105300

C
WRITE ( 6, 32030 ) NSUBT, NSUBP, OMG2, OMG1, EPRAD, PSCAL, NSURF89105500
32030 FORMAT( 1HD, 53HNUMBER OF SUBSTITUTION SETS..... NS89107000
1UBT, 16 / 54H NUMBER OF SUBSTITUTION PARAMETERS..... NSUB89107100
2P, 16 / 54H PARAXIAL SCALE FACTOR (FOCAL LENGTH)..... OMG2, 89107200
3 E17.7 / 54H MERIDIONAL SCALE FACTOR (BACK FOCUS)..... OMG189107300
4, E17.7 / 54H RADIUS OF ENTRANCE PUPIL..... EPR89107400
5D, E17.7 / 54H SPOT DIAGRAM SCALE FACTOR..... PSC89107500
6AL, E17.7 // 54H NUMBER OF SURFACES..... NS89107600
7SURF, 18 )                                                         89107700

C
C      SET LINE COUNT TO EJECT PAGE, PRINT TITLE AND DATE        89108000
C      LINE = 60                                                  89108100

C
C      PRINT OUT SURFACE INFORMATION                                89201000
C

```

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
STAR92 - EFM SOURCE STATEMENT - (FN(8) -

11/09/67

PAGE 3

DO 11000 I = 1, NSURF	89201800
IF (LINE .GT. 99) CALL PRCTCL	89201400 88
LINE = LINE + 11	89201600
C	
C BLANK OUT FIRST LINE OF SURFACE PRINT	89201600
DO 1800 J = 1, 9	89201700
1800 PLINE(J) = BLANK	89201800
IT = ISURFC(1, J)	89202000
R = SURFC(11, J)	89202100
B = SURFC(12, J)	89202200
C	
C DETERMINE SURFACE TYPE	89202800
GO TO (2000, 3000, 3500, 5000, 6000, 7000, 8000), IT	89202900
C	
C CONIC OF REVOLUTION	89203000
2000 IF (R .NE. 0.0) GO TO 2100	89203100
PLINE(1) = SURFA(1)	89203200
GO TO 10000	89203300
2100 IF (B .NE. 0.0) GO TO 2200	89203400
PLINE(1) = SURFA(2)	89203500
GO TO 10000	89203600
2200 IF (B .NE. 1.0) GO TO 2300	89203800
PLINE(1) = SURFA(3)	89203900
PLINE(2) = SURFA(4)	89204000
GO TO 10000	89204100
2300 IF (B .LT. 1.0) GO TO 2400	89204200
PLINE(1) = SURFA(7)	89204300
PLINE(2) = SURFA(8)	89204400
GO TO 10000	89204500
2400 IF (B .LE. - 1.0) GO TO 3000	89205000
PLINE(1) = SURFA(5)	89205100
PLINE(2) = SURFA(6)	89205200
GO TO 10000	89205300
C	
C HORIZONTAL CYLINDER	89206000
3000 PLINE(1) = SURFA(9)	89206100
PLINE(2) = SURFA(10)	89206200
GO TO 4000	89206300
C	
C VERTICAL CYLINDER	89206500
3500 PLINE(1) = SURFA(11)	89206600
PLINE(2) = SURFA(12)	89206700
4000 IF (R .EQ. 0.0) GO TO 3000	89206800
IF (B .NE. 0.0) GO TO 4100	89206900
JTEMP = 12	89207000
GO TO 4600	89207100
4100 IF (B .NE. 1.0) GO TO 4200	89207200
JTEMP = 15	89207300
GO TO 4600	89207400
4200 IF (R .LT. 1.0) GO TO 4500	89207500
DO 4300 J = 1, 4	89207600
4300 PLINE(J+2) = SURFA(J+21)	89207700
GO TO 10000	89207800
4500 IF (B .LE. - 1.0) GO TO 3000	89208000
JTEMP = 18	89208100
4600 DO 4700 J = 1, 3	89208200

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
 STAR92 - EFN SOURCE STATEMENT - IPN(8) -

11/09/67

PAGE 4

KTEMP = JTEMP + J	89209300
4700 PLINE(J+2) = SURFA(KTEMP)	89209400
GO TO 10000	89209500
C	
NONSTANDARD CONIC	89301000
8000 DO 8100 J = 1, 2	89301200
8100 PLINE(J) = SURFA(J+28)	89301300
IF (R .NE. 0.0) GO TO 8200	89301500
PLINE(4) = SURFA(1)	89301600
GO TO 10000	89301700
8200 IF (B .NE. 0.0) GO TO 8300	89301800
PLINE(4) = SURFA(2)	89301900
GO TO 10000	89302000
8300 IF (B .NE. 1.0) GO TO 8400	89302100
PLINE(4) = SURFA(3)	89302200
PLINE(5) = SURFA(4)	89302300
GO TO 10000	89302400
8400 IF (B .LT. 1.0) GO TO 8500	89302500
PLINE(4) = SURFA(7)	89302600
PLINE(5) = SURFA(8)	89302700
GO TO 10000	89302800
8500 IF (B .LE. - 1.0) GO TO 9000	89302900
PLINE(4) = SURFA(5)	89303000
PLINE(5) = SURFA(6)	89303100
GO TO 10000	89303200
89303300	
89303400	
89303500	
89303600	
89303700	
89303800	
C	
ASPHERIC OF REVOLUTION	89304000
9000 DO 9100 J = 1, 4	89304100
9100 PLINE(J) = SURFA(J+28)	89304200
ITEMP = 4	89304300
IF (R .NE. 0.0) GO TO 9200	89304400
JTEMP = 42	89304500
ITEMP = 3	89304600
GO TO 9600	89304700
9200 IF (B .NE. 0.0) GO TO 9300	89304800
JTEMP = 45	89304900
ITEMP = 3	89305000
GO TO 9600	89305100
9300 IF (B .NE. 1.0) GO TO 9400	89305200
JTEMP = 48	89305300
GO TO 9600	89305400
9400 IF (B .LT. 1.0) GO TO 9500	89305500
JTEMP = 56	89305600
GO TO 9600	89305700
9500 IF (B .LE. - 1.0) GO TO 9000	89305800
JTEMP = 52	89305900
89306000	
89306100	
89306200	
89306300	
89306400	
89306500	
89306600	
C	
HORIZONTAL ASPHERIC CYLINDER	89307000
7000 DO 7100 J = 1, 8	89307100
7100 PLINE(J) = SURFA(J+32)	89307200
GO TO 8200	89307300
C	

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
 STAR8Z - EFM SOURCE STATEMENT - IFM(8) -

11/08/67

PAGE 9

```

C          VERTICAL ASPHERIC CYLINDER
0000 DO 0100 J = 1, 8
0100 PLINE(J) = SURFA(J+37)
0200 ITEMP = 3
      IF ( B .NE. 0.0 ) GO TO 0300
      JTEMP = 60
      GO TO 0600
0300 IF ( B .NE. 1.0 ) GO TO 0400
      JTEMP = 63
      ITEMP = 4
      GO TO 0600
0400 IF ( B .LT. 1.0 ) GO TO 0500
      ITEMP = 4
      JTEMP = 70
      GO TO 0600
0500 IF ( B .LE. - 1.0 ) GO TO 0600
      JTEMP = 67
0600 DO 0700 J = 1, ITEMP
      KTEMP = JTEMP + J
0700 PLINE(J+8) = SURFA.KTEMP
      GO TO 10000
C          SURFACE ECCENTRICITY VALUE IS INCORRECT
0900 CALL ERROR( 36H ECCENTRICITY VALUE WRONG - SURFACE , I )
C
C          PRINT SURFACE TYPE AND NUMBER
10000 WRITE ( 6, 32050 ) I, ( PLINE(J), J = 1, 8 )
32050 FOR / ( 1HD, 7NSURFACE, 13, 2X, 9A6 )
      WRITE ( 6, 32060 ) ISURFC(1,I), ( SURFC(J,I), J = 2, 10 )
32060 FORMAT( 1HD, 12X, 4HTYPE, 12X, 19HAPERTURE PARAMETERS, 9X,
1 1 8HDISTANCE / 116, 7X, 3E15.7 // 43X, 21HINDICES OF REFRA
2ACTION / 8X, 6E15.7 )
      IF ( ISURFC(1,I) .GT. 3 ) GO TO 10200
      WRITE ( 6, 32070 ) SURFC(11,I), SURFC(12,I)
32070 FORMAT( 1HD, 15X, 1HR, 14X, 1HD / 8X, 2E15.7 )
      GO TO 11000
10200 IF ( ISURFC(1,I) .NE. 4 ) GO TO 10400
      WRITE ( 6, 32080 ) ( SURFC(J,I), J = 11, 16 )
32080 FORMAT( 1HD, 16X, 1HR, 14X, 1HD, 13X, 2HXT, 13X, 2HYT, 14X, 1HU,
1 14X, 1HV / 8X, 6E15.7 )
      GO TO 11000
10400 WRITE ( 6, 32090 ) ( SURFC(J,I), J = 11, 16 )
32090 FORMAT( 1HD, 16X, 1HR, 14X, 1HD, 13X, 2HAB, 13X, 2HAB, 13X, 2HA4,
1 13X, 2HA5 / 8X, 6E15.7 )
C
C          PRINT OUT OF SURFACE HAS BEEN COMPLETED
11000 CONTINUE
      IF ( LINE .GT. 55 ) CALL PRCTCL
C
C          PRINT OUT IMAGE SURFACE
      ITEMP = NSURF + 1
      WRITE ( 6, 32100 )
32100 FORMAT( 1HD, 13HIMAGE SURFACE )
      WRITE ( 6, 32110 ) ISURFC(1,ITEMP), ( SURFC(J,ITEMP), J = 1, 10 )
32110 FORMAT( 1HD, 12X, 4HTYPE, 41X, 21HINDICES OF REFRACTION / 1.6,
1 7X, 6E15.7 )
      IF ( APLA6 .NE. 1 ) GO TO 12000
  
```

89307500
 89307600
 89307700
 89307800
 89308000
 89308100
 89308200
 89308300
 89308400
 89308500
 89308600
 89308700
 89308800
 89308900
 89309000
 89309100
 89309200
 89309400
 89309500
 89309600
 89309700
 89401000
 89401500
 89402000 201
 89402100 202
 89402200
 89402500 207
 89402600
 89402700
 89402800
 89403000
 89403200 216
 89403300
 89403500
 89403600
 89404000 223
 89404100
 89404200
 89404300
 89405000 229
 89405100
 89405200
 89405500
 89405600
 89405800
 89406100 238
 89406200
 89406300 240
 89406400
 89407000 241
 89407100
 89407200
 89407500

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
START97 - EFM SOURCE STATEMENT - (FN(8) -

11/09/67

PAGE 6

WRITE (0, 32120) SUBFC(11,ITEMP), SUBFC(12,ITEMP)	89507000	291
32120 FORMAT(1ND, 10X, 1NR, 14X, 1ND / 6X, 2E15.7)	89507700	
12000 IF(NSUBT .EQ. 0) GO TO 20000	89501000	
CALL PRCTCL	89501100	
C		
C PRINT OUT SUBSTITUTION INFORMATION	89501200	
C		297
WRITE(0, 32130)	89501400	298
32130 FORMAT(1ND, 10X, 24NSUBSTITUTION INFORMATION)	89501500	
LINE = LINE + 2	89501600	
C		
C SET POINTERS - I REFERENCES ISUBST() AND JTEMP	89502000	
C REFERENCES SUBST() AND SUBSYN()	89502100	
I = NSUBP + NSUBT + 1	89502200	
JTEMP = 1	89502300	
C		
C LOOP TO CONSTRUCT VARIABLE FORMAT	89502500	
12500 DO 13500 K = 1, 6	89502600	
C ITEMP SETS THE NUMBER OF SUBSTITUTION VALUES	89502700	
C PRINTED PER LINE	89502800	
ITEMP = K	89502900	
IF(JTEMP .LE. NSUBP) GO TO 13000	89503000	
ITEMP = ITEMP - 1	89503100	
C		
C LAST FORMAT (CONSISTING OF ITEMP VALUES) HAS BEEN	89503200	
C CONSTRUCTED	89503300	
FMT(ITEMP+3) = MFMT	89503400	
GO TO 14000	89503500	
13000 FMT(ITEMP+2) = MFMT	89503600	
IF(ISUBST(I) .LT. 0) FMT(ITEMP+2) = IFMT	89503700	
JTEMP = JTEMP + 1	89503800	
13500 I = I + 1	89503900	
14000 KTEMP = JTEMP - ITEMP	89504000	
LTEMP = JTEMP - 1	89504100	
MEMP = KTEMP	89504200	
NTEMP = LTEMP	89504300	
J = 0	89504400	
C		
C PRINT OUT SUBSTITUTION SYMBOL HEADING	89505000	
14500 WRITE (0, 32160) ((SUBSYN(L,K), L=1,2), K=KTEMP,LTEMP)	89505200	299
32160 FORMAT(1ND, 7X, 3NSBT, 6X, 6(2A6, 5X))	89505300	
LINE = LINE + 2	89505700	
C		
C PRINT OUT SUBSTITUTION VALUES FOR ALL SETS FOR THE FIRST	89506000	
C GROUP OF PARAMETERS	89506100	
15000 J = J + 1	89506200	
C FMT IS (1H .11D,5X, AND E17.7, OR 17.11X DEPENDING ON	89506300	
C THE TYPE OF THE SUBSTITUTION PARAMETER	89506400	
WRITE(0, FMT) J, (SUBST(K), K= NTEMP, NTEMP)	89506500	300
NTEMP = NTEMP + NSUBP	89506600	
NTEMP = NTEMP + NSUBP	89506700	
LINE = LINE + 1	89506800	
IF(LINE .LT. 99) GO TO 15500	89507000	
C		
C START NEW PAGE WITH TITLE AND SUBSTITUTION SYMBOL HEADING	89507200	
CALL PRCTCL	89507300	312

TR-67-700-10-2
Page III-1-33
Program Listing
Link 1

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
STAR9Z - EPN SOURCE STATEMENT - IPN(8) -

11/08/67

PAGE 7

GO TO 14500	89507400
15500 IF (J .LT. NSUBT) GO TO 15000	89507500
C	
C VALUES FOR ALL SETS HAVE BEEN PRINTED	89508000
IF (JTEMP .LE. NSUBP) GO TO 12500	89508200
C	
C ALL SUBSTITUTION PARAMETERS HAVE BEEN PROCESSED	89508500
C	
20000 RETURN	89509000
END	89509500

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
 STARR2 STORAGE MAP

11/08/67

PAGE 8

SUBROUTINE STARR
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00019	R
DATE	00027	I	PUNCD	00032	R	DATA	00033	R
UNOGE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
PLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXPP	00055	R	DEKPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMTN	00070	R
CDIST	00071	R	HO	00072	R	DELH	00073	R
SYBNK	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NOBGN	00100	I	NOBGN	00104	I
NOBJH	00117	R	WCLRH	00126	R	WIMHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESIGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I			

COMMON BLOCK			SYMBLS	ORIGIN	06667	LENGTH	02424	
DESIGN	00000	I	SUBSYM	01130	I	BDYSYM	02114	I

COMMON BLOCK			PRNT	ORIGIN	11313	LENGTH	00002	
LINE	00000	I	PAGE	00001	I			

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SURFA	11315	R	PLINE	11427	R	FHT	11440	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IFMT	11451	R	I	11452	I	J	11453	I
BLANK	11454	R	IT	11455	I	R	11456	R
B	11457	R	JTEMP	11460	I	KTEMP	11461	I
ITEMP	11462	I	K	11463	I	SFMT	11464	R
IFMT	11465	R	LTEMP	11466	I	MTEMP	11467	I
NTEMP	11470	I						

ENTRY POINTS

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
 STARDZ STORAGE MAP

11/08/67

PAGE 9

STARD	SECTION	9
PRTCTL	SECTION	10
.PXEM.	SECTION	13
.PCNV.	SECTION	16

SUBROUTINES CALLED

.FWRD.	SECTION	11	ERROR2	SECTION	12
.UNDB.	SECTION	14	.FFIL.	SECTION	15
SYSLOC	SECTION	17			
EFN	IFN	CORRESPONDENCE			

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	FORMAT	11535	32010	FORMAT	11705	1000	10A	12457
32020	FORMAT	12046	32030	FORMAT	12053	11000	234A	13339
1800	31A	12531	2000	39A	12557	3000	59A	12632
3500	61A	12637	5000	94A	12725	6000	121A	13005
7000	158A	13105	8000	166A	13113	2100	43A	12966
10000	202A	13211	2200	47A	12575	2300	51A	12608
2400	55A	12620	9000	200A	13204	4000	62A	12643
4100	69A	12656	4600	66A	12707	4200	73A	12665
4500	83A	12700	4300	78A	12675	4700	89A	12715
5100	97A	12726	5200	105A	12741	5300	109A	12750
5400	113A	12761	5900	117A	12773	6100	124A	13008
6200	134A	13025	6600	150A	13064	6300	139A	13036
6400	143A	13045	6500	147A	13055	6700	153A	13075
7100	161A	13106	6200	173A	13120	6100	169A	13114
6300	179A	13131	6600	192A	13163	6400	184A	13142
6500	189A	13154	6700	195A	13174	32050	FORMAT	12171
32060	FORMAT	12177	10200	220A	13270	32070	FORMAT	12250
10400	229A	13315	32080	FORMAT	12237	32090	FORMAT	12255
32100	FORMAT	12274	32110	FORMAT	12301	12000	254A	13426
32120	FORMAT	12316	20000	320A	13654	32150	FORMAT	12325
12500	261A	13455	13500	276A	13522	13000	272A	13500
14000	282A	13527	14500	287A	13542	32160	FORMAT	12334
15000	297A	13575	15500	314A	13642			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 13705.

TR-67-700-10-2

Page III-1-36

Program Listing

Link 1

Page Revised 10/1/68

INF,NG10001,44903-0,77387,AM IC

10J00 110MF

7094 A

09/19/68

PAGE

8-

BTITLE

DESIGN, SENSITIVITY, AND PROFILE PLOT PRINT ROUTINE

SDCTC STAR12 M94.XR7

LINK 1 (STAR10)

81001000

DESIGN, SENSITIVITY, AND PROFILE PLOT PRINT ROUTINE
STAR12 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

```

SUBROUTINE STAR1D                                     $1001200
C
C *** DESCRIPTION ***                                $1001202
C STAR1D LISTS ALL OF THE DESIGN INPUT DATA WITH DESCRIPTIVE $1001204
C TITLES. IN ADDITION IT LISTS ...                   $1001206
C                                                    $1001208
C BOUNDARY CONDITION MATRIX (IF NCOND.NE.0)         $1001210
C SENSITIVITY DATA (IF NPERTB.NE.0)                $1001212
C PROFILE PLOT DATA (IF NPTS.NE.0)                 $1001214
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), FUNCID, DT001000
1 DATA(3486)                                         DT001100
DIMENSION WOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3), DT101000
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIFLN ), ( DATA(9), DT101200
3 IMODE ), ( DATA(10), NSPLN ), ( DATA(11), NOBJH ), ( DATA(12), DT101300
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
5 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18), DT101500
6 ZETA ), ( DATA(19), HEXFF ), ( DATA(20), DEXFF ), ( DATA(21), DT101600
7 WEXFF ), ( DATA(22), DLFLN ), ( DATA(23), OMGA2 ), ( DATA(24), DT101700
8 OMGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
9 PSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
A DUMIN ), ( DATA(31), ODIST ), ( DATA(32), HD ), ( DATA(33), DT102000
B DELH ), ( DATA(34), SYSMX ), ( DATA(35), WXDIR ), ( DATA(36), DT102100
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
D NDSGV(1) ), ( DATA(53), WOBJH(1) ), (DATA(60), WCLRH(1) ), DT102300
E ( DATA(67), WIMHT(1) ), (DATA(74), EIMHT(1) ), ( DATA(81), DT102400
F CIMPL(1) ), ( DATA(88), WCLRS(1) ), DT102500
G ( DATA(94), LATT(1,1), ILATT(1,1) ), ( DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1) ), ( DATA(2182), DESGN(1,1), IDESGN(1,1) ), DT102700
I ( DATA(2682), SUBST(1), ISUBST(1) ), ( DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1) ), ( DATA(3483), NCOND ) DT102900
EQUIVALENCE ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS ) DT103000
INTEGER CNTRL, AFLAG, DATE, ATRGGR DT201000
REAL LATTC DT202000
C
C THIS BLOCK OF COMMON CONTAINS THE SYMBOLS USED IN DESIGN SY001000
C AND SUBSTITUTION - UTILIZED FOR PRINT OUT SY002000
COMMON / SYMBLS / SDESN(12,50), SUBSYM(2,250), BDYSYM(2,100) SY003000
INTEGER SDESN, SUBSYM, BDYSYM SY004000
C
COMMON / PRNT / LINE, PAGE FT002500
INTEGER PAGE FT002700
C
C THIS BLOCK OF COMMON CONTAINS PERTURBATION VALUES FB001000
COMMON / PERTB / PERTB(30,4), NPERTB, REFOCS FB002000
EQUIVALENCE ( PERTB(1,1), IPERTB(1,1) ) FB003000
DIMENSION IPERTB(30,4) FB004000
INTEGER REFOCS FB005000
C
COMMON / PLOT / YMAXX, DZMIN, NPTS FC001000

```

DESIGN, SENSITIVITY, AND PROFILE PLOT PRINT ROUTINE
STAR1Z - EFN SOURCE STATEMENT - IFN(S) -

09/1 /68

PAGE

C	DATA BLANK / 6H / , MINUS / 6H- /	\$1002000	.
	INTEGER BLANK	\$1002200	
	DIMENSION ISIGN(6)	\$1002400	
C	START OF STAR10 SUBROUTINE	\$1003000	
C	CALL PRCTL	\$1003200	2
	WRITE (6, 32000) FNUMB, FLNGH, WFLGH, ZETA, HEXFF, DEXFF,	\$1004000	
	1 WEXFF, (1, WOBJH(1), WCLRH(1), WIMHT(1), EIMHT(1), I = 1, NOBJH)	\$1004100	3
32000	FORMAT(1H0, 53HREQUIRED F/ NUMBER.....)	FNS1004500	
	1UMB, E19.7 / 54H REQUIRED FOCAL LENGTH.....)	FS1004600	
	2LNGH, E19.7 / 54H WEIGHT ON FOCAL LENGTH.....)	\$1004700	
	3WFLGH, E19.7 / 53H CORRECTION VECTOR SCALE FACTOR.....)	\$1004800	
	4 ZETA, E20.7 / 54H OBJECT HEIGHT FOR EXIT PUPIL PLANE.....)	\$1004900	
	5 HEXFF, E19.7 / 54H REQUIRED POSITION EXIT PUPIL PLANE.....)	\$1005000	
	6. DEXFF, E19.7 / 54H WEIGHT ON EXIT PUPIL PLANE POSITION.....)	\$1005100	
	7.. WEXFF, E19.7 // 21X, 44H ***** OBJECT HEIGHT DEPENDENT WEIGHTS	\$1005200	
	8***** // 4X, 14H OBJECT HEIGHT, 7X, 5HWOBJH, 11X, 5HWCLRH, 11X,	\$1005300	
	9 5HWIMHT, 11X, 5HEIMHT / (112, 6X, 4E16.7))	\$1005400	
C	WRITE (6, 32010) (1, WCLRS(1), I = 1, NCLRS)	\$1006000	13
32010	FORMAT(1H0, 23X, 36H ***** COLOR DEPENDENT WEIGHTS ***** // 31X,	\$1006500	
	1 6H COLOR, 6X, 6H WCLRS / (135, 4X, E14.7))	\$1006600	
C	WRITE (6, 32020) WDIR, WYDIR, SPFEA, OMGAF, DUMIN, SYSMX, NCONDS	\$1007000	20
32020	FORMAT(1H0, 53HX DIRECTION WEIGHT.....)	WXS1008000	
	1DIR, E19.7 / 54H Y DIRECTION WEIGHT.....)	WS1008100	
	2YDIR, E19.7 // 54H MINIMUM THICKNESS TO AVOID FEATHERING.....)	\$1008200	
	3 SPFEA, E19.7 / 54H SCALE FACTOR FOR FEATHER CHECK RAY.....)	\$1008300	
	4. OMGAF, E19.7 / 54H MINIMUM PARAMETER INCREMENT.....)	\$1008400	
	5. DUMIN, E19.7 / 54H MAXIMUM LENGTH OF SYSTEM.....)	\$1008500	
	6.. SYSMX, E19.7 / 54H NUMBER OF BOUNDARY CONDITIONS.....)	\$1008600	
	7.. NCOND, 19)	\$1008700	
	IF(NCOND .LE. 0) GO TO 2500	\$1041000	
C	BOUNDARY CONDITIONS ARE PRESENT	\$1041200	
C	WRITE (6, 32030)	\$1041400	23
32030	FORMAT(1H0, 22X, 38H ***** BOUNDARY CONDITION MATRIX *****)	\$1041600	
	WRITE (6, 32035)	\$1042000	24
32035	FORMAT(1H0, 17X, 9HPARAMETER, 10X, 7HMINIMUM, 10X, 7HMAXIMUM)	\$1042200	
	LINE = LINE + NOBJH + NCLRS + 29	\$1042400	
	DO 2000 I = 1, NCOND	\$1043000	
	WRITE(6, 32040) BDYSYM(1,I), BDYSYM(2,I), BOUNDS(2,I),	\$1043200	
	1 BOUNDS(3,I)	\$1043300	28
32040	FORMAT(1H , 17X, 2AG, 2E17.7)	\$1043400	
	LINE = LINE + 1	\$1043600	
	IF(LINE .LT. 55) GO TO 2000	\$1043800	
C	BEGIN NEW PAGE - PRINT TITLE	\$1044000	
C	CALL PRCTL	\$1044200	37
	WRITE (6, 32035)	\$1044500	38
	LINE = LINE + 2	\$1044600	
	2000 CONTINUE	\$1045000	
	2500 IF(LINE .GT. 45) CALL PRCTL	\$1045500	
C			44

DESIGN, SENSITIVITY, AND PROFILE PLOT PRINT ROUTINE
STAR1Z - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE

```

WRITE ( 6, 32050 ) ( NDSGN(I), I=1,4 ), NDSGV(1)          S1046000   45
32050 FORMAT( 1HD, 56HTOTAL NUMBER OF DESIGN VARIABLES..... NDS1046500
1SGN(1), 14 / 57H MINIMUM PERMISSIBLE DESIGN VARIABLE NUMBER..... NS1046500
2DSGN(2), 14 / 57H MAXIMUM PERMISSIBLE DESIGN VARIABLE NUMBER..... S1046700
3NDSGN(3), 14 / 57H NUMBER OF DESIGN ITERATIONS.....S1046800
4 NDSGN(4), 14 / 57H NUMBER OF DESIGN VARIABLES PER ITERATION.....S1046900
5. NDSGV(1), 14 )          S1047000
ITEMP = NDSGV(1) + 1      S1047200

C
WRITE ( 6, 32060 ) ( NDSGV(I), I = 2, ITEMP )          S1047500   52
32060 FORMAT( 1H , 56HDESIGN VARIABLE NUMBERS FIRST-ITERATION..... NDS1048000
1SGV(2), 1X, 1G13 )          S1048100
WRITE ( 6, 32070 ) NJAIL, AFLAG          S1048500   59
32070 FORMAT( 1H , 53HNUMBER OF SKIPS FOR BAD COMBINATION..... NJS1048700
1AIL, 17 / 55H SPECIAL DESIGN ITERATION PRINTOUT FLAG..... ITNFS1048800
2RT, 16 )          S1048900
WRITE ( 6, 32072 ) ATRGCR          S1048910   60
32072 FORMAT(1H , 54HOPTION FLAG TO RE-COMPUTE D(J)..... ATRS1048912
1GCR, 16 )          S1048914
WRITE ( 6, 32076 ) GAUSS          S1048924   61
32076 FORMAT(1H , 53HSTANDARD DEVIATION FOR GAUSSIAN WEIGHTING..... GAUS1048926
1SS, E19.7 )          S1048928
LINE = LINE + 1          S1049000
IF( NDSGN(1) .EQ. 0 ) GO TO 5000          S1049100
IF( LINE .GT. 40 ) CALL PRCTL          S1049200

C
C          PRINT OUT DESIGN INFORMATION          S1061000
C
WRITE ( 6, 32080 )          S1061500   68
32080 FORMAT( 1HD, 21X, 30H***** DESIGN INFORMATION ***** )          S1062000
WRITE ( 6, 32090 )          S1062500   69
32090 FORMAT( 1HD, 4H ID., 4H NO., 32X, 18H DESIGN PARAMETERS, 35X,
1 11ACTUAL DET., 2X, 9HREQ. DET., 3X, 9HINCREMENT )          S1063000
LINE = LINE + 4          S1063100
ITEMP = NDSGN(1)          S1063500
S1063600

C
DO 4000 I = 1, ITEMP          S1064000
KK = IABS( IDESGN(1,I) )          S1064100
DO 2900 K = 1, 6          S1064200
2900 ISIGN(K) = BLANK          S1064300
DO 3000 K = 1, KK          S1064400
3000 IF( IDESGN(K+3,I) .LE. 0 ) ISIGN(K) = MINUS          S1064500
WRITE ( 6, 32100 ) I, IDESGN(1,I), ISIGN(1), SDESN(1,I),          S1064600
1 SDESN(2,I), ISIGN(2), SDESN(3,I), SDESN(4,I), ISIGN(3),          S1064700
2 SDESN(5,I), SDESN(6,I), ISIGN(4), SDESN(7,I), SDESN(8,I),          S1064800
3 ISIGN(5), SDESN(9,I), SDESN(10,I), ISIGN(6), SDESN(11,I),          S1064900
4 SDESN(12,I), DESGN(10,I), DESGN(2,I), DESGN(3,I)          S1065000   89
32100 FORMAT( 1H , 12, 14, 1X, 6( A1, 2A6, 1X ), 3E12.4 )          S1065200
LINE = LINE + 1          S1065500
IF( LINE .LT. 55 ) GO TO 4000          S1065600

C
C          BEGIN NEW PAGE - PRINT TITLE          S1066000
CALL PRCTL          S1066200   110
WRITE ( 6, 32090 )          S1066500   111
LINE = LINE + 2          S1066700
4000 CONTINUE          S1067000

```

DESIGN, SENSITIVITY, AND PROFILE PLOT PRINT ROUTINE
STAR12 - EFN SOURCE STATEMENT - IFN(S) -

09/19/68

PAGE 4

```

5000 IF( NPRTD .EQ. 0 ) GO TO 6000          $1070000
C
C      PRINT OUT PERTURBATION INFORMATION    $1071000
C
      LINE = LINE + NPRTD + 7                $1071100
      IF( LINE .GT. 54 ) CALL PRCTCL         $1071200   120
      WRITE ( 6, 32110 )                    $1071600   121
32110 FORMAT( 1HD, 18X, 35H***** SENSITIVITY INFORMATION ***** ) $1071800
      WRITE ( 6, 32112 ) NPRTD              122
32112 FORMAT( 1HD, 54HNUMBER OF PARAMETERS..... NP
      IERTD, 16 )
      NREFC = REFOCS + 1
C
      GO TO ( 5100, 5150, 5200, 5250 ), NREFC
5100 WRITE ( 6, 32114 ) REFOCS              124
32114 FORMAT( 1H , 54HREFOCUS OPTION..... RE
      IFOCS, 5X, 11, 3X, 14HDO NOT REFOCUS )
      GO TO 5300
5150 WRITE ( 6, 32115 ) REFOCS              126
32115 FORMAT( 1H , 54HREFOCUS OPTION..... RE
      IFOCS, 5X, 11, 3X, 22HREFOCUS FOR MIN. RMS X )
      GO TO 5300
5200 WRITE ( 6, 32116 ) REFOCS              128
32116 FORMAT( 1H , 54HREFOCUS OPTION..... RE
      IFOCS, 5X, 11, 3X, 22HREFOCUS FOR MIN. RMS Y )
      GO TO 5300
5250 WRITE ( 6, 32117 ) REFOCS              130
32117 FORMAT( 1H , 54HREFOCUS OPTION..... RE
      IFOCS, 5X, 11, 3X, 26HREFOCUS FOR MIN. SPOT SIZE )
C
5300 WRITE ( 6, 32118 )                    131
32118 FORMAT( 1HD, 2X, 3HND., 10X, 9HPARAMETER, 10X, 8HPER CENT )
      DO 5500 I = 1, NPRTD                  $1073000
      WRITE ( 6, 32120 ) I, PERTB(1,3), PERTB(1,4), PERTB(1,1) $1073400   134
32120 FORMAT( 1H , 14, 10X, 2AS, 10X, E14.7 ) $1073600
5500 CONTINUE                              $1073800
6000 IF( NPRTS .LE. 0 ) GO TO 10000        $1081000
C
C      PRINT OUT CROSS SECTION PLOT INFORMATION $1081200
C
      LINE = LINE + 6                        $1081400
      IF( LINE .GT. 54 ) CALL PRCTCL         $1081600   146
      WRITE ( 6, 32150 )                    $1082000   147
32150 FORMAT( 1HD, 15X, 42H***** CROSS SECTION PLOT INFORMATION ***** ) $1082200
      WRITE ( 6, 32155 ) YMAXX, DZMIN, NPRTS $1083000   148
32155 FORMAT( 1HD, 53HDIAMETER OF LENS PLOT..... YMS1083200
      1AXX, E19.7 / 54H MINUMIM AIR SPACE SEPARATION..... DS1083400
      2ZMIN, E19.7 / 53H NUMBER OF SURFACE POINTS PER ELEMENT..... $1083600
      3NPRTS, 18 )                          $1083800
C
C      END OF STAR12 SUBROUTINE              $1091000
10000 RETURN                                $1093000
      END                                    $1095000

```

DESIGN, SENSITIVITY, AND PROFILE PLOT PRINT ROUTINE
 STAR12 STORAGE MAP

09/19/68

PAGE 5

SUBROUTINE STAR10
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
WNTREL	00000	I	CONTRL	00001	I	TITLE	00013	R	
DATE	00027	I	PUNCI0	00032	R	DATA	00033	R	
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I	
WCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I	
ISUBP	00041	I	NIFLN	00042	I	IMODE	00043	I	
ISFLN	00044	I	NDBJH	00045	I	NSURF	00046	I	
AFLAG	00047	I	DELY	00050	R	FNUMB	00051	R	
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R	
HXPFF	00055	R	DEXFF	00056	R	WEXFF	00057	R	
CLFLN	00060	R	OMGA2	00061	R	OMGA1	00062	R	
DELD	00063	R	EPRAD	00064	R	FSCAL	00065	R	
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R	
ODIST	00071	R	HD	00072	R	DELH	00073	R	
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R	
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I	
WDBJH	00117	R	WCLRH	00126	R	WIMHT	00135	R	
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R	
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R	
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I	
SUBST	05224	R	ISUBST	05224	I	BUNDS	06210	R	
IDNDS	06210	I	NCOND	06665	I	ATRCGR	06666	I	
GAUSS	06667	R							

COMMON BLOCK			SYMBLS	ORIGIN	06672	LENGTH	02424		
SDESN	00000	I	SUBSYM	01130	I	BDYSYM	02114	I	

COMMON BLOCK			PRNT	ORIGIN	11316	LENGTH	00002		
LINE	00000	I	PAGE	00001	I				

COMMON BLOCK			PERTB	ORIGIN	11320	LENGTH	00172		
PERTB	00000	R	NPERTB	00170	I	REFOCS	00171	I	

COMMON BLOCK			FLOTG	ORIGIN	11512	LENGTH	00003		
WMAXX	00000	R	DZMIN	00001	R	NFTS	00002	I	

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ISIGN	11515	I						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
BLANK	11523	I	I	11524	I	ITEMP	11525	I
KK	11526	I	MINUS	11527	I	NREFC	11530	I

DESIGN, SENSITIVITY, AND PROFILE PLOT PRINT ROUTINE
 STAR12 STORAGE MAP

09/19/68

PAGE 6

ENTRY POINTS

STAR10 SECTION 13

SUBROUTINES CALLED

PRCTL SECTION 14
 .UN06. SECTION 17
 SYSLOC SECTION 20

.FWRD. SECTION 15
 .FFIL. SECTION 18

.FXEM. SECTION 1
 .FCNV. SECTION 1

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	FORMAT	11554	32010	FORMAT	11726	32020	FORMAT	11750
2500	42A	13021	32030	FORMAT	12067	32035	FORMAT	12101
2000	39A	13015	32040	FORMAT	12113	32050	FORMAT	12120
32060	FORMAT	12211	32070	FORMAT	12226	32072	FORMAT	12255
32076	FORMAT	12271	5000	115A	13336	32080	FORMAT	12305
32090	FORMAT	12315	4000	112A	13332	2900	79A	13175
3000	84A	13212	32100	FORMAT	12340	6000	141A	13506
32110	FORMAT	12350	32112	FORMAT	12361	5100	124A	13407
5150	126A	13420	5200	128A	13431	5250	130A	13442
32114	FORMAT	12375	5300	131A	13452	32115	FORMAT	12416
32116	FORMAT	12440	32117	FORMAT	12462	32118	FORMAT	12505
5500	138A	13504	32120	FORMAT	12516	10000	149A	13547
32150	FORMAT	12524	32155	FORMAT	12536			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 13602.

TR-67-700-10-2
Page III-1-43
Program Listing
Link 1
Page Revised 10/1/68

INF.ND10001.44903-5.77387.AM 10

10J00 L10KF 7094 A 09/19/68

PAGE

25
BTITLE SUBROUTINE TO UNPACK A BCD WORD
BTDFTC UNDCD. M94,XR7 LINK 1 (UNBCD)

UDC00010

SUBROUTINE TO UNPACK A BCD WORD
 UNBCD.

09/19/68

PAGE 2

STORAGE MAP

SUBROUTINE UNBCD
 DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE
ACD	00001	1

SYMBOL	LOCATION	TYPE
--------	----------	------

SYMBOL	LOCATION	TYPE
--------	----------	------

ENTRY POINTS

UNBCD	SECTION	3
-------	---------	---

SUBROUTINES CALLED

.DECD.	SECTION	4
.FSLO.	SECTION	7
.FFIL.	SECTION	10

.FSLI.	SECTION	5
.FRTN.	SECTION	8
SYSLOC	SECTION	11
EFN	IFN	CORRESPONDENCE

.ENCD.	SECTION	1
.FCHV.	SECTION	

EFN	IFN	LOCATION
10	6A	00024

EFN	IFN	LOCATION
1000	FORMAT	00017

EFN	IFN	LOCATION
1010	FORMAT	00020

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00105.

TR-67-700-10-2
Page III-1-46
Program Listing
Link 1

1 JOB INF.N010001.44800-0.77387.AN IC1,3.5000 L. WILSON INFORMATICS A 13'51''29''' 11/08/67 *****V13NJD*

STB JOB NOGO

BT

BTITLE VECTOR-MATRIX BCD TO INTEGER CONVERT ROUTINE

BTBFC VCNVR. M/4.XH? LINK 1 (VCNVRT.MCNVRT) VCO01030

VECTOR-MATRIX BCD TO INTEGER CONVERT ROUTINE
VCNVR. - EFN SOURCE STATEMENT - IFN(8) -

11/08/67

PAGE 1

```

SUBROUTINE VCNVRT( BCDWRD, INTEG, * )                                VC001100
C                                                                 VC001102
C *** INPUTS ***                                                VC001104
C BCDWRD LOCATION CONTAINING A BCD VECTOR SUBSCRIPT OF THE FORM X) ,VC001106
C      XX) , XXX)                                                VC001108
C                                                                 VC001110
C *** OUTPUTS ***                                               VC001112
C INTEG LOCATION WHERE VCNVRT IS TO STORE INTEGER SUBSCRIPT     VC001114
C * ALTERNATE RETURN IF BCDWRD IS INVALID                        VC001116
C                                                                 VC001118
C *** DESCRIPTION ***                                           VC001120
C VCNVRT CONVERTS A GIVEN VECTOR SUBSCRIPT FROM BCD TO INTEGER. VC001122
C USED BY READS WHEN PROCESSING SYMBOL CARDS.
C
DIMENSION IDIGT(10), IBCD(6)                                       VC001200
INTEGER BCDWRD, RTPRN, COMMA                                       VC001300
DATA IDIGT / 6H000000, 6H000001, 6H000002, 6H000003, 6H000004, VC001400
1 6H000005, 6H000006, 6H000007, 6H000008, 6H000009 /, RTPRN, COMMA VC001500
2 / 6H000000), 6H000000, /                                       VC001600
C                                                                 VC001700
CALL UNPBCD( BCDWRD, IBCD )                                       VC002100
J = 1                                                                VC002200
INTEG = 0                                                            VC002300
100 DO 200 I = 1, 10                                               VC002400
IF( IBCD(J) .EQ. IDIGT(I) ) GO TO 300                             VC002500
200 CONTINUE                                                       VC002600
IF( IBCD(J) .NE. RTPRN .OR. J.LT.2 .OR. J.GT.4) RETURN 1       VC002700
RETURN                                                             VC002800
300 INTEG=10*INTEG+I-1                                           VC003000
J = J + 1                                                         VC003100
GO TO 100                                                         VC003200
C                                                                 VC003300
ENTRY MCVNVRT( BCDWRD, INTEG, JNTEG, * )                          VC003700
C
C *** INPUTS ***                                                VC003702
C BCDWRD LOCATION CONTAINING A BCD MATRIX SUBSCRIPT OF THE FORM X,Y) ,VC003704
C      X,YY) , XX,Y) , XX,YY)                                     VC003706
C                                                                 VC003708
C *** OUTPUTS ***                                               VC003710
C INTEG LOCATION WHERE INTEGER ROW SUBSCRIPT IS TO BE STORED     VC003712
C JNTEG LOCATION WHERE INTEGER COLUMN SUBSCRIPT IS TO BE STORED VC003714
C * ALTERNATE RETURN IF BCDWRD IS INVALID                        VC003716
C                                                                 VC003718
C *** DESCRIPTION ***                                           VC003720
C MCVNVRT CONVERTS A GIVEN BCD MATRIX SUBSCRIPT PAIR TO INTEGER. VC003722
C USED BY READS WHEN PROCESSING SYMBOL CARDS.                    VC003724
C
CALL UNPBCD( BCDWRD, IBCD )                                       VC003800
C DETERMINE FIRST SUBSCRIPT, SAVE IN INTEG                        VC003900
J = 1                                                                VC004000
INTEG = 0                                                            VC004100
500 DO 600 I = 1, 10                                               VC004200
IF( IBCD(J) .EQ. IDIGT(I) ) GO TO 700                             VC004300
600 CONTINUE                                                       VC004400
IF( IBCD(J) .NE. COMMA .OR. J.LT.2 .OR. J.GE.4) RETURN 1       VC004500

```

VECTOR-MATRIX BCD TO INTEGER CONVERT ROUTINE
VCNVR. - EFN SOURCE STATEMENT - IFN(8) -

11/08/67

PAGE 2

```

      K = 0
      J = J + 1
      GO TO 800
700  INTEG = 10**(J-1) * INTEG + I - 1
      J = J + 1
      GO TO 800
C      DETERMINE SECOND SUBSCRIPT, SAVE IN JNTEG
800  JNTEG = 0
900  DO 1000 I = 1, 10
      IF ( IBCD(J) .EQ. IDIGT(I) ) GO TO 1100
1000 CONTINUE
      IF ( IBCD(J) .NE. RTRN .OR. J .GT. 8 ) RETURN 1
      RETURN
1100 JNTEG = 10**K * JNTEG + I - 1
      K = K + 1
      J = J + 1
      GO TO 800
      END
VC004600
VC004700
VC004800
VC004900      45
VC005000
VC005100
VC005200
VC005300
VC005400
VC005500
VC005600
VC005700
VC005800
VC005900      65
VC006000
VC006100
VC006200
VC006300
```

VECTOR-MATRIX BCD TO INTEGER CONVERT ROUTINE
 VCNVR.

11/09/67

PAGE 3

STORAGE MAP

SUBROUTINE VCNVRT
 DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IDIST	00001	I	IBCD	00013	I			
			UNDIMENSIONED PROGRAM VARIABLES					
RTPRN	00021	I	COMMA	00022	I	J	00023	I
I	00024	I	K	00025	I			

ENTRY POINTS

VCNVRT SECTION 3

MCNVRT SECTION 4
 SUBROUTINES CALLED

UNPBCD SECTION 5

.XP1. SECTION 6
 EFN IFN CORRESPONDENCE

SYSLOC SECTION 7

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
100	5A	00050	200	14A	00082	300	20A	00104
500	27A	00126	600	36A	00140	700	44A	00166
800	48A	00210	900	49A	00211	1000	58A	00223
1100	64A	00241						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00347.

TR-67-700-10-2

Page III-1-50

Program Listing

Link 1

Page Revised 10/1/68

INF,ND10001,44903-0,77367,AM IC

BJOB L10MKC 7094 A

09/25/68

PAGE

34

BTITLE

MTF CARD PUNCH ROUTINE

81DFTC MTFPNZ M94,XR7

LINK 1 (MTFFH)

MT000000

MTF CARD PUNCH ROUTINE
 MTFPHZ - EFN SOURCE STATEMENT - IFN(S) -

09/25/68

PAGE

```

SUBROUTINE MTFPH
C
C *** DESCRIPTION ***
C MTFPH, UTILIZING FOLDP DATA, PUNCHES FAGOS NAMELIST INPUT
C CARDS FOR FAGOS MTF CAPABILITY.
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), FUNCID,
1 DATA(3486)
DIMENSION MDSJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7),
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100),
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26),
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IDNDS(3,100)
EQUIVALENCE ( DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3),
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),
2 NSUBT ), ( DATA(7), NSURF ), ( DATA(8), NIPLN ), ( DATA(9),
3 INODE ), ( DATA(10), NSPLN ), ( DATA(11), MDSJH ), ( DATA(12),
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15),
5 FIMXB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18),
6 ZETA ), ( DATA(19), HEXPP ), ( DATA(20), DEXPP ), ( DATA(21),
7 WEXPP ), ( DATA(22), DLPLN ), ( DATA(23), OMGA2 ), ( DATA(24),
8 OMGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27),
9 FSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30),
A CURIN ), ( DATA(31), ODIST ), ( DATA(32), HD ), ( DATA(33),
B DELH ), ( DATA(34), SYSMX ), ( DATA(35), WDIR ), ( DATA(36),
C WDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42),
D NDSGV(1) ), ( DATA(53), MDSJH(1) ), ( DATA(60), WCLRH(1) ),
E ( DATA(67), WIMHT(1) ), ( DATA(74), EIMHT(1) ), ( DATA(81),
F CIMPL(1) ), ( DATA(88), WCLRS(1) ),
G ( DATA(94), LATT(1,1), ILATT(1,1) ), ( DATA(102), SURFC(1,1),
H ISURFC(1,1) ), ( DATA(2182), DESGN(1,1), IDESGN(1,1) ),
I ( DATA(2682), SUBST(1), ISUBST(1) ), ( DATA(3182), BOUNDS(1,1),
J IDNDS(1,1) ), ( DATA(3483), NCOND )
EQUIVALENCE ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )
INTEGER CNTRL, AFLAG, DATE, ATRGGR
REAL LATT

C
DIMENSION BETAP(100), BETAS(100), DELMU(100), DFORM(4,100),
1 RMDS(100), SKAPA(100), TOK(10), TS(100), XMUS(100)

C
C START OF MTFPH SUBROUTINE
C
C COMPUTE MTF INPUT FROM FOLDP PARAMETERS
BETAO = EPRAD
N = NSURF + 1
NR = MDSJH
DO 1200 K = 1, NR
1200 TOK(K) = SURFC(4,1)
XLINV = 1. / ( ODIST - SURFC(4,1) )
DO 1000 J = 1, N
IF( SURFC(2,J) .NE. 0. ) GO TO 100
BETAP(J) = 0.
BETAS(J) = 0.
GO TO 500
100 IF( SURFC(3,J) .EQ. 0. ) GO TO 200
CALL ERROR2( 36H MTF ABORTED--RECTANGULAR APERTURE , J )

```

MT000000
 MT001001
 MT001002
 MT001003
 MT001004
 MT001005
 DT001000
 DT001100
 DT002000
 DT002100
 DT002200
 DT002300
 DT101000
 DT101100
 DT101200
 DT101300
 DT101400
 DT101500
 DT101600
 DT101700
 DT101800
 DT101900
 DT102000
 DT102100
 DT102200
 DT102300
 DT102400
 DT102500
 DT102600
 DT102700
 DT102800
 DT102900
 DT103000
 DT201000
 DT202000
 MT001006
 MT001007
 MT001008
 MT001009
 MT001010
 MT001011
 MT001012
 MT001013
 MT001014
 MT001015
 MT001016
 MT001017
 MT001018
 MT001019
 MT001100
 MT001101
 MT001102
 MT001103
 MT001104
 MT001105
 MT001106
 MT001107
 MT001108
 MT001109
 MT001110
 MT001111
 MT001112
 MT001113
 MT001114
 MT001115
 MT001116
 MT001117
 MT001118
 MT001119
 MT001120
 MT001121
 MT001122
 MT001123
 MT001124
 MT001125
 MT001126
 MT001127
 MT001128
 MT001129
 MT001130

MTF CARD PUNCH ROUTINE
 MTFPHZ - EFN SOURCE STATEMENT - IPN(8) -

09/25/68

PAGE

RETURN	MT001135
200 IF(SURFC(2,J) .LT. 0.) GO TO 300	MT001140
BETAP(J) = 0.	MT001145
BETAS(J) = SURFC(2,J)	MT001150
GO TO 500	MT001155
300 BETAP(J) = SURFC(2,J)	MT001160
BETAS(J) = 0.	MT001165
500 NCP4 = NCLRS + 4	MT001170
DELHU(J) = SURFC(NCP4,J) - SURFC(5,J)	MT001175
IF(SURFC(1,J) .NE. 4.) GO TO 600	MT001180
CALL ERROR2(36H MTF ABORTED-TRANSLATED/TILTED CONIC, J)	MT001185 48
RETURN	MT001190
600 IFLG = 0	MT001191
IF(SURFC(1,J) .LE. 3.) IFLG = 1	MT001195
DO 700 I = 1,4	MT001200
IF(IFLG .EQ. 0) GO TO 650	MT001205
DFORM(1,J) = 0.	MT001206
GO TO 700	MT001207
650 IP = 1 + 12	MT001210
DFORM(1,J) = SURFC(IP,J)	MT001212
700 CONTINUE	MT001214
RHOS(J) = SURFC(11,J)	MT001218
SKAPA(J) = 1. - SURFC(12,J)**2	MT001220
IF(J .EQ. N) GO TO 1000	MT001222
JP1 = J + 1	MT001225 ---
TS(J) = SURFC(4,JP1)	MT001226
XMUS(J) = SURFC(5,JP1)	MT001230
1000 CONTINUE	MT001235
C	MT001300
C PUNCH PAGOS NAMELIST \$INPUT CARDS	MT001305
PUNCH 10, BETA0, N, NK, XLINV	MT001310 83
10 FORMAT(16H \$INPUT BETA0=, E15.8, 4H, N=, 12, 5H, NK=, 12,	MT001315
1 8H, XLINV=, E15.8, 1H,)	MT001320
IF(TOK(1) .EQ. 0.) GO TO 1450	MT001325
L = 4	MT001330
IF(NK .LT. 4) L = NK	MT001335
PUNCH 20, (TOK(N1), N1=1,L)	MT001340 89
20 FORMAT(8H TOK(1)=, 4(E15.8, 1H,))	MT001345
IF(NK .LE. 4) GO TO 1500	MT001350
PUNCH 25, (TOK(N1), N1=5,NK)	MT001355 98
25 FORMAT(4(1X, E15.8, 1H,))	MT001360
GO TO 1500	MT001365
1450 PUNCH 30, NK	MT001370 106
30 FORMAT(8H TOK(1)=, 11, 4H#0.,)	MT001375
1500 DCNT1 = 0.	MT001380
DCNT2 = 0.	MT001385
DCNT3 = 0.	MT001390
DCNT4 = 0.	MT001395
DCNT5 = 0.	MT001400
DO 1525 I = 1, N	MT001420
DCNT1 = DCNT1 + ABS(BETAP(I))	MT001425
DCNT2 = DCNT2 + ABS(BETAS(I))	MT001430
DCNT3 = DCNT3 + ABS(DELHU(I))	MT001435
DCNT4 = DCNT4 + ABS(RHOS(I))	MT001440
1525 DCNT5 = DCNT5 + ABS(SKAPA(I))	MT001445
L = 4	MT001450

MTF CARD PUNCH ROUTINE MTFPNZ - EFN SOURCE STATEMENT - IFN(8) -	09/25/68	PAGE
IF(N .LT. 4) L = N	MT001455	
IF(DCNT1 .EQ. 0.) GO TO 1550	MT001460	
PUNCH 35, (BETAP(N2), N2=1,L)	MT001465	124
35 FORMAT(10H BETAP(1)=, 4(E15.0, 1H.))	MT001470	
IF(N .LE. 4) GO TO 1600	MT001475	
PUNCH 25, (BETAP(N2), N2=5,N)	MT001480	133
GO TO 1600	MT001485	
1550 PUNCH 40, N	MT001490	141
40 FORMAT(10H BETAP(1)=, 12, 4H#0.,)	MT001495	
1600 IF(DCNT2 .EQ. 0.) GO TO 1650	MT001500	
PUNCH 45, (BETAS(N3), N3=1,L)	MT001505	144
45 FORMAT(10H BETAS(1)=, 4(E15.0, 1H.))	MT001510	
IF(N .LE. 4) GO TO 1700	MT001515	
PUNCH 25, (BETAS(N3), N3=5,N)	MT001520	153
GO TO 1700	MT001525	
1650 PUNCH 50, N	MT001530	161
50 FORMAT(10H BETAS(1)=, 12, 4H#0.,)	MT001535	
1700 IF(DCNT3 .EQ. 0.) GO TO 1750	MT001540	
PUNCH 55, (DELMU(N4), N4=1,L)	MT001545	164
55 FORMAT(10H DELMU(1)=, 4(E15.0, 1H.))	MT001550	
IF(N .LE. 4) GO TO 1800	MT001555	
PUNCH 25, (DELMU(N4), N4=5,N)	MT001560	173
GO TO 1800	MT001565	
1750 PUNCH 60, N	MT001570	181
60 FORMAT(10H DELMU(1)=, 12, 4H#0.,)	MT001575	
1800 IF(DCNT4 .EQ. 0.) GO TO 1850	MT001580	
PUNCH 65, (RHOS(N5), N5=1,L)	MT001585	184
65 FORMAT(9H RHOS(1)=, 4(E15.0, 1H.))	MT001590	
IF(N .LE. 4) GO TO 1900	MT001595	
PUNCH 25, (RHOS(N5), N5=5,N)	MT001600	193
GO TO 1900	MT001605	
1850 PUNCH 70, N	MT001610	201
70 FORMAT(9H RHOS(1)=, 12, 4H#0.,)	MT001615	
1900 IF(DCNT5 .EQ. 0.) GO TO 1950	MT001620	
PUNCH 75, (SKAPA(N6), N6=1,L)	MT001625	204
75 FORMAT(10H SKAPA(1)=, 4(E15.0, 1H.))	MT001630	
IF(N .LE. 4) GO TO 2000	MT001635	
PUNCH 25, (SKAPA(N6), N6=5,N)	MT001640	213
GO TO 2000	MT001645	
1950 PUNCH 80, N	MT001650	221
80 FORMAT(10H SKAPA(1)=, 12, 4H#0.,)	MT001655	
2000 DCNT6 = 0.	MT001660	
DCNT7 = 0.	MT001665	
NM1 = N - 1	MT001670	
DO 2025 I = 1, NM1	MT001675	
DCNT6 = DCNT6 + ABS(TS(I))	MT001680	
2025 DCNT7 = DCNT7 + ABS(XMUS(I))	MT001385	
L = 4	MT001690	
IF(N .LT. 5) L = N - 1	MT001695	
IF(DCNT6 .EQ. 0.) GO TO 2050	MT001700	
PUNCH 85, (TS(N7), N7=1,L)	MT001705	237
85 FORMAT(7H TS(1)=, 4(E15.0, 1H.))	MT001710	
IF(N .LE. 5) GO TO 2100	MT001715	
PUNCH 25, (TS(N7), N7=5,NM1)	MT001720	246
GO TO 2100	MT001725	
2050 PUNCH 90, NM1	MT001730	254

MTF CARD PUNCH ROUTINE
 MTFPHZ - EFN SOURCE STATEMENT - IFN(S) -

09/25/68

PAGE

90	FORMAT(7H TC(1)=, 12, 4H'D.,)	MT001735
2100	IF(DCNT7 .EQ. 0.) GO TO 2150	MT001740
	PUNCH 95, (XMUS(10), N8=1,L)	MT001745 257
95	FORMAT(9H XMUS(1)=, 4(E15.8, 1H,))	MT001750
	IF(N .LE. 5) GO TO 2200	MT001755
	PUNCH 25, (XMUS(N8), N8=5, NH1)	MT001760 266
	GO TO 2200	MT001765
2150	PUNCH 105, NH1	MT001770 274
105	FORMAT(9H XMUS(1)=, 12, 4H'D.,)	MT001775
2200	DFCNT = 0.	MT001780
	DO 2225 J = 1, N	MT001785
	DO 2225 I = 1, 4	MT001790
2225	DFCNT = DFCNT + ADS(DFORM(I,J))	MT001795
	L = 2	MT001800
	IF(N .EQ. 1) L = 1	MT001805
	IF(DFCNT .EQ. 0.) GO TO 2250	MT001810
	PUNCH 110, ((DFORM(I,J), I=1,4), J=1,L)	MT001815 291
110	FORMAT(12H DFORM(1,1)=, 4(E15.8, 1H,))	MT001820
	IF(N .LE. 2) GO TO 2300	MT001825
	PUNCH 25, ((DFORM(I,J), I=1,4), J=3,N)	MT001830 303
	GO TO 2300	MT001835
2250	NT4 = 4*N	MT001837
	PUNCH 115, NT4	MT001840 315
115	FORMAT(12H DFORM(1,1)=, 13, 4H'D.,)	MT001845
2300	PUNCH 120	MT001850 316
120	FORMAT(16H IFROG=3, M=0 8)	MT001855
C		MT001960
C	END OF SUBROUTINE MTFPHZ	MT001965
	RETURN	MT001870
	END	MT001875

MTF CARD PUNCH ROUTINE
 MTFPNZ

09/25/68

PAGE 5

STORAGE MAP

SUBROUTINE MTFPN
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671		
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MENTAL	00000	I		CONTRL	00001	I	TITLE	00013	R
BATE	00027	I		PUNCID	00032	R	DATA	00033	R
LMODE	00033	I		NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I		NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I		NIFLN	00042	I	IMODE	00043	I
ISPLN	00044	I		NDDJH	00045	I	NSURF	00046	I
AFLAG	00047	I		DELY	00050	R	FIUMD	00051	R
FLNGH	00052	R		WFLGH	00053	R	ZETA	00054	R
WEXFP	00055	R		DEXPP	00056	R	WEXFP	00057	R
DLPLN	00060	R		OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R		EPFAD	00064	R	PSCAL	00065	R
OMGAF	00066	R		SPFEA	00067	R	DUMIN	00070	R
ODIST	00071	R		MO	00072	R	DELH	00073	R
BSMX	00074	R		WXDIR	00075	R	WYDIR	00076	R
NOTAN	00077	R		NDSGN	00100	I	NDSGV	00104	I
WDDJH	00117	R		WCLRH	00126	R	WIHHT	00135	R
EIMHT	00144	R		CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R		ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I		DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R		ISUBT	05224	I	BOUND5	06210	R
IDNDS	06210	I		NCONU	06665	I	ATRGR	06666	I
GAUSS	06667	R							

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
BETAP	06672	R	BETAS	07036	R	DELHU	07202	R
DFORM	07346	R	RHOS	10166	R	SKAPA	10332	R
TK	10476	R	TS	10510	R	XMUS	10654	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
BETA0	11020	R	N	11021	I	NK	11022	I
XLINV	11023	R	J	11024	I	NCP4	11025	I
IFLG	11026	I	I	11027	I	IP	11030	I
JP1	11031	I	L	11032	I	DCNT1	11033	R
DCNT2	11034	R	DCNT3	11035	R	DCNT4	11036	R
DCNT5	11037	R	DCNT6	11040	R	DCNT7	11041	R
IM:	11042	I	DFCNT	11043	R	NT4	11044	I

ENTRY POINTS

MTFPN SECTION 5

MTF CARD PUNCH ROUTINE
 MTFPNZ

STORAGE MAP

09/25/68

PAGE 6

SUBROUTINES CALLED

ERROR2 SECTION 6			.FPUN. SECTION 7			.FFIL. SECTION 8		
.PCNV. SECTION 9			SYSLOC SECTION 10					
EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1200	7A	11313	1000	61A	11510	100	21A	11343
900	39A	11371	200	27A	11355	300	35A	11366
600	30A	11414	700	66A	11455	650	62A	11442
10	FORMAT	11106	1450	106A	11604	20	FORMAT	11125
1500	107A	11613	25	FORMAT	11133	30	FORMAT	11137
1525	115A	11654	1550	141A	11740	35	FORMAT	11144
1600	142A	11747	40	FORMAT	11152	1650	161A	12013
45	FORMAT	11157	1700	162A	12022	50	FORMAT	11165
1750	181A	12066	55	FORMAT	11172	1800	182A	12075
60	FORMAT	11200	1850	201A	12141	65	FORMAT	11205
1900	202A	12150	70	FORMAT	11213	1950	221A	12214
75	FORMAT	11220	2000	222A	12223	80	FORMAT	11226
2025	226A	12242	2050	254A	12327	85	FORMAT	11233
2100	255A	12336	90	FORMAT	11340	2150	274A	12402
16	FORMAT	11245	2200	275A	12411	105	FORMAT	11253
2225	281A	12426	2250	314A	12537	110	FORMAT	11260
2300	316A	12551	115	FORMAT	11266	120	FORMAT	11274

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12602.

TR-67-700-10-2
Page III-2-1
Program Listing
Link 2

Section 2

LINK 2

TR-67-700-10-2
Page III-2-2
Program Listing
Link 2

3JOB INF.N010001.44800-0.77387.AM IC1.6.8000 WILSON - INFORMATICS A 08:17:58*** 02/18/68 *****LINK101
818JOB NO60
80
81TITLE ROUTINE TO GENERATE POINTS ON A PLANE CONIC ARC
818FTC ARPT87 M86.XR7 LINK 2 (ARPT8) AR001000

ROUTINE TO GENERATE POINTS ON A PLANE CONIC ARC
 ARPTSZ - EPN SOURCE STATEMENT - IPN(S) -

```

SUBROUTINE ARPTS( IS, ZA, YA, ZB, YB, N, ZLST, YLST )      AR001000
C
C *** INPUTS ***                                         AR002000
C IS SURFACE NUMBER SPECIFIES CURVE Y = F(Z)           AR003000
C ZA Z COORDINATE POINT A                               AR003200
C YA Y COORDINATE POINT A                               AR003400
C ZB Z COORDINATE POINT B                               AR003700
C YB Y COORDINATE POINT B                               AR003800
C N NUMBER OF POINTS TO GENERATE                       AR004000
C
C *** OUTPUTS ***                                       AR004100
C ZLST ZI I=1,2,...,N ZI=ZA ZN=ZB                       AR004200
C YLST YI I=1,2,...,N YI=YA YN=YB                       AR004400
C
C *** DESCRIPTION ***                                   AR004800
C IS DEFINES A CURVE Y=F(Z) FOR Z BETWEEN POINTS A AND B. ARPTS GENERATAR005000
C ES A SET OF POINTS (ZI,YI) I=1,2,...,N SUCH THAT ...  AR005200
C 1. (ZI,YI) SATISFIES Y=F(Z) FOR ALL I.                AR005800
C 2. THE ZI ARE EQUALLY SPACED.                         AR006200
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNC10, DT001000
1 DATA(3483)                                             DT001100
DIMENSION WOBJH(7), WCLRH(7), WINHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLR8(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3), DT101000
1 NSLCS ), ( DATA(4), NCLR8 ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9), DT101200
3 IMODE ), ( DATA(10), NSPLN ), ( DATA(11), NOBJH ), ( DATA(12), DT101300
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
5 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WPLGH ), ( DATA(18), DT101500
6 ZETA ), ( DATA(19), HEXPP ), ( DATA(20), DEXPP ), ( DATA(21), DT101600
7 WEXPP ), ( DATA(22), DLPLN ), ( DATA(23), OMC2 ), ( DATA(24), DT101700
8 OMC1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
9 PSCAL ), ( DATA(28), OMC2F ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
A DUMIN ), ( DATA(31), ODIST ), ( DATA(32), HO ), ( DATA(33), DT102000
B DELH ), ( DATA(34), SYSMX ), ( DATA(35), WDIR ), ( DATA(36), DT102100
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
D NDSGV(1) ), ( DATA(53), WOBJH(1) ), ( DATA(60), WCLRH(1) ), DT102300
E ( DATA(67), WINHT(1) ), ( DATA(74), EIMHT(1) ), ( DATA(81), DT102400
F CIMPL(1) ), ( DATA(88), WCLR8(1) ), DT102500
G ( DATA(94), LATT(1,1), ILATT(1,1) ), ( DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1) ), ( DATA(2102), DESGN(1,1), IDESGN(1,1) ), DT102700
I ( DATA(2602), SUBST(1), ISUBST(1) ), ( DATA(3102), BOUNDS(1,1), DT102800
J IBNDS(1,1) ), ( DATA(3483), NCOND ) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
DIMENSION ZLST(1), YLST(1)                               AR006500
C
IF( N.LE. 1 ) RETURN                                     AR007000
DELTAZ = ( ZB - ZA ) / FLOAT( N-1 )                     AR007200
RC = 2. * SURFC(11,18)                                  AR007400
BC = SURFC(12,18) - 1.                                  AR007600

```

ROUTINE TO GENERATE POINTS ON A PLANE CONIC ARC
ARPTSZ - EPN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 2

DO 10 I = 1, N
ZLST(I) = ZA + FLOAT(I-1) * DELTAZ
10 YLST(I) = SORT(ABS(BC * ZLST(I) * ZLST(I) + RC * ZLST(I)))
C
RETURN
END

AR007800
AR008000
AR008200

14

AR009000
AR009200

ROUTINE TO GENERATE POINTS ON A PLANE CONIC ARC
 ARPTS STORAGE MAP

62/10/66

PAGE 3

SUBROUTINE ARPTS
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	00000	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSGTY	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGAE	00061	R	OMGAI	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	00071	R	HO	00072	R	DELH	00073	R
SYBNK	00074	R	WDIR	00075	R	WDIR	00076	R
NDTAN	00077	R	NDGCH	00100	I	NDGCV	00104	I
NOBJH	00117	R	WCLRH	00126	R	WINHT	00133	R
EINHT	00144	R	CIMPL	00133	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBGT	05224	R	ISUBST	05224	I	BOUNDS	08210	R
IBNDS	08210	I	NCOND	08665	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DELTAZ	08667	R	RC	08670	R	BC	08671	R
I	08672	I						

ENTRY POINTS

ARPTS SECTION 5

SUBROUTINES CALLED

SQRT	SECTION 6	E.1	SECTION 7	E.2	SECTION 8
E.3	SECTION 9	E.4	SECTION 10	CC.1	SECTION 11
CC.2	SECTION 12	CC.3	SECTION 13	CC.4	SECTION 14
SYBLOC	SECTION 15				

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	10A	08767						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07057.

TR-67-700-10-2
Page III-2-6
Program Listing
Link 2

INF.N010001.44800-0.77387.AM IC

ISJOB L10M1 7004 A 02/18/68

PAGE 4

BT
BTITLE ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
SIBPTC CELAIZ M94.XR7 LINK 2 (CELAIR) EADD1000

ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
CELAIR - EFM SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 1

```

SUBROUTINE CELAIR( * )                                EA001200
C
C *** INPUTS ***                                     EA001300
C NSURF NUMBER OF LENS SURFACES.                     (DATA COMMON) EA001600
C SURFC ARRAY OF SURFACE PARAMETERS.                 (DATA COMMON) EA001700
C
C *** OUTPUTS ***                                     EA002000
C NELMT NUMBER OF EDITED ELEMENTS.                   (ELMAIR COMMON) EA002100
C NAIRS NUMBER OF EDITED AIR SPACES.                  (ELMAIR COMMON) EA002200
C ELMNMX MATRIX OF EDITED SURFACE ELEMENTS.          (ELMAIR COMMON) EA002300
C AIRNMX MATRIX OF EDITED AIR SPACES.                (ELMAIR COMMON) EA002400
C VTXCRD VECTOR OF VERTEX COORDINATES OF SURFACES   (ELMAIR COMMON) EA002500
C REFERENCED TO ENTRANCE PUPIL.                      EA002600
C * ALTERNATE RETURN IF ILLEGAL SYSTEM.              EA002700
C
C *** DESCRIPTION ***                                 EA003000
C THIS SUBROUTINE GENERATES THE MATRIX OF EDITED SURFACE ELEMENTS EA003100
C AND THE MATRIX OF EDITED AIR SPACES. THE VECTOR OF VERTEX EA003200
C COORDINATES OF EACH SURFACE REFERENCED TO THE ENTRANCE PUPIL EA003300
C IS ALSO COMPUTED.                                  EA003400
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483)                                         DT001100
DIMENSION MOBJH(7), WCLR(7), WINHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLR(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATTC(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 INODE), (DATA(10), NSPLN), (DATA(11), MOBJH), (DATA(12), DT101300
, NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMG2), (DATA(24), DT101700
8 OMG1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), MO), (DATA(33), DT102000
B DELM), (DATA(34), SYSMX), (DATA(35), WXDIR), (DATA(36), DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), MOBJH(1)), (DATA(60), WCLR(1)), DT102300
E DATA(67), WINHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLR(1)), DT102500
G (DATA(94), LATT(1,1), ILATTC(1,1)), (DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
COMMON / PRT / LINE, PAGE PT002500
INTEGER PAGE PT002700
C
COMMON / ELMAIR / NELMT, NAIRS, ELMNMX(2,100), AIRNMX(2,100), EL001000
1 VTXCRD(100) EL002000

```

ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
CELAIR - EFN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 2

C	INTEGER ELMNTH, AIRMTH	EL003000	
C	DIMENSION ECURL(2,100), ACURL(2,100)	EA004000	
C	INTEGER ECURL, ACURL	EA004500	
C	START OF CELAIR SUBROUTINE	EA005000	
C	IF(ODIST .GE. 0.) GO TO 1000	EA005500	
	WRITE (6, 32010)	EA005600	4
32010	FORMAT(1H0, 35HOBJECT IS TO THE RIGHT OF SURFACE 1)	EA005700	
	RETURN 1	EA005800	
C	1000 SPRIME = SURFC(4,1) .. ODIST	EA006000	
	IF(SPRIME .LE. 0.) GO TO 1500	EA006100	
	WRITE (6, 32020)	EA006200	6
32020	FORMAT(1H0, 44HOBJECT IS TO THE RIGHT OF THE ENTRANCE PUPIL)	EA006300	
	RETURN 1	EA006400	
C	1500 VTXCRD(1) = SURFC(4,1)	EA006600	
	IF(ISURFC(1,1) .LE. 3) GO TO 2500	EA006700	
	I = 1	EA006800	
	2000 WRITE (6, 32030) I, ISURFC(1,1)	EA006900	13
32030	FORMAT(1H0, 7HSURFACE, 13, 7HIS TYPE, 13, 14HNOT ACCEPTABLE)	EA007000	
	RETURN 1	EA007100	
C	2500 IF(NSURF .LE. 1) GO TO 3500	EA101000	
	DO 3000 I = 2, NSURF	EA101100	
	IF(ISURFC(1,I) .GT. 3) GO TO 2000	EA101200	
	VTXCRD(I) = VTXCRD(I-1) + SURFC(4,I)	EA101300	
	IF(VTXCRD(I) .GE. SPRIME) GO TO 3000	EA101400	
	WRITE (6, 32040) I	EA101500	31
32040	FORMAT(1H0, 33HOBJECT IS TO THE RIGHT OF SURFACE, 13)	EA101600	
	RETURN 1	EA101700	
	3000 CONTINUE	EA101800	
3500	VTXCRD(NSURF+1) = VTXCRD(NSURF) + DELD	EA101900	
	CALL PRTCTL	EA102000	38
	WRITE (6, 32050)	EA102100	39
32050	FORMAT(1H0, 47HVERTEX COORDINATES REFERENCED TO ENTRANCE PUPIL //	EA102200	
	1 2X, 7HSURFACE, 10X, 1H2)	EA102300	
	LINE = LINE + 4	EA102400	
	ZERO = 0.	EA102500	
	WRITE (6, 32060) ZERO	EA102600	41
32060	FORMAT(1H , 1X, 14HENTRANCE PUPIL, E18.7)	EA102700	
	LINE = LINE + 1	EA102800	
	DO 4500 I = 1, NSURF	EA102900	
	IF(LINE .LE. 56) GO TO 4000	EA103000	
C	BEGIN NEW PAGE	EA103100	
	CALL PRTCTL	EA103200	49
	WRITE (6, 32070)	EA103300	50
	LINE = LINE + 4	EA103400	
4000	WRITE (6, 32070) I, VTXCRD(I)	EA103500	51
32070	FORMAT(1H , 13, E20.7)	EA103600	
	4500 LINE = LINE + 1	EA103700	
	WRITE (6, 32075) VTXCRD(NSURF+1)	EA103800	56
32075	FORMAT(1H , 1X, 11HIMAGE PLANE, E21.7)	EA103900	

ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
 CELAIZ - EFM SOURCE STATEMENT - EFM(S) -

02/16/66

PAGE 3

```

C
NRC = 0
NAC = 0
DO 8000 I = 1, 2
DO 9000 J = 1, 100
ECURL(I,J) = 0
9000 ACURL(I,J) = 0
IF ( NSURF .GT. 1 ) GO TO 7500
I = 1
X1 = SURFC(S,I)
X2 = SURFC(S,2)
IF ( X1 .EQ. 1. ) GO TO 6000
9900 WRITE ( 6, 32000 ) I
32000 FORMAT( 1HD, 15HINDER 1 SURFACE, 13, 13H IS NOT VALID )
RETURN 1

C
6000 IF ( X2 .NE. -1. ) GO TO 6500
NEC = NEC + 1
ECURL(1,NEC) = I
ECURL(2,NEC) = I
NAC = NAC + 1
ACURL(1,NAC) = I
ACURL(2,NAC) = I+1
GO TO 13900
6500 IF ( X2 .GT. 1. ) GO TO 7000
I = I+1
GO TO 5900
7000 NEC = NEC + 1
ECURL(1,NEC) = I
ECURL(2,NEC) = I+1
GO TO 13900

C
7500 DO 10000 I = 1, NSURF
X1 = SURFC(S,I)
X2 = SURFC(S,I+1)
IF ( ABS( X1 ) .LT. 1. ) GO TO 8500
IF ( ABS( X2 ) .GE. 1. ) GO TO 8000
I = I+1
GO TO 8500
8000 IF ( X1 .LT. 1. ) GO TO 8000
IF ( X1 .NE. 1. ) GO TO 8500
IF ( X2 .LE. 1. ) GO TO 8200
9100 NEC = NEC + 1
ECURL(1,NEC) = I
ECURL(2,NEC) = I+1
GO TO 10000
8200 IF ( X2 .EQ. -1. ) GO TO 8400
IF ( X2 .EQ. 1. ) GO TO 8210
I = I+1
GO TO 8500
8210 IF ( NAC .NE. 0 ) ACURL(2,NAC) = I+1
GO TO 10000
8400 NEC = NEC + 1
ECURL(1,NEC) = I
ECURL(2,NEC) = I
NAC = NAC + 1
    
```

EA107000
 EA107200
 EA107400
 EA107600
 EA107800
 EA108000
 EA108200
 EA108400
 EA108600
 EA108800
 EA109000
 EA109200
 EA109400
 EA109600
 EA109800
 EA201000
 EA201200
 EA201400

70

EA201600
 EA201800
 EA202000
 EA202200
 EA202400
 EA202600
 EA202800
 EA203000
 EA203200
 EA203400
 EA203600
 EA203800
 EA204000
 EA204200
 EA204400
 EA204600

EA204800
 EA205000
 EA205200
 EA205400
 EA205600
 EA205800
 EA206000
 EA206200
 EA206400
 EA206600
 EA206800
 EA207000
 EA207200
 EA207400
 EA207600
 EA207800
 EA208000
 EA208100
 EA208200
 EA208400
 EA208500
 EA208600
 EA208700
 EA208800
 EA208900
 EA209000
 EA209200

ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
CELA12 - EPN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 4

```

ACURL(1,NAC) = I
ACURL(2,NAC) = I+1
GO TO 10000
0800 IF ( X2 .LT. 1. ) GO TO 0800
IF ( X2.NE.1. ) GO TO 0710
0700 NAC = NAC + 1
ACURL(1,NAC) = I
ACURL(2,NAC) = I+1
GO TO 10000
0710 IF ( X2.NE.X1 ) GO TO 0100
ECURL(2,NEC) = I+1
GO TO 10000
0600 IF ( ABS( X2 ) .EQ. X1 ) GO TO 0100
I = I+1
GO TO 0500
0500 IF ( X1 .NE. -1. ) GO TO 0200
IF ( X2 .LT. -1. ) GO TO 0100
IF ( X2 .EQ. 1. ) GO TO 0400
IF ( X2.EQ.-1. ) GO TO 0210
I = I+1
GO TO 0500
0200 IF ( X2 .GT. -1. ) GO TO 0400
IF ( X2.NE.-1. ) GO TO 0710
GO TO 0700
0400 IF ( X2 .EQ. ABS( X1 ) ) GO TO 0100
I = I+1
GO TO 0500
10000 CONTINUE
C
13500 NELNT = 0
NAIRS = 0
DO 14000 I = 1, 2
DO 14100 J = 1, 100
ELMNTX(I,J) = 0
14000 AIRMNTX(I,J) = 0
NELNT = I
ELMNTX(1,1) = ECURL(1,1)
ELMNTX(2,1) = ECURL(2,1)
IF ( NEC .EQ. 1 ) GO TO 16000
DO 15500 I = 2, NEC
I1 = ECURL(1,I)
I2 = ECURL(2,I)
VX1 = VTXCRD(I1)
VX2 = VTXCRD(I2)
IF ( VX1 .LE. VX2 ) GO TO 14500
IT = I1
I1 = I2
I2 = IT
VXT = VX1
VX1 = VX2
VX2 = VXT
14500 DO 15000 J = 1, NELNT
J1 = ELMNTX(1,J)
J2 = ELMNTX(2,J)
VY1 = VTXCRD(J1)
VY2 = VTXCRD(J2)
EA800400
EA800600
EA800800
EA801000
EA801200
EA801400
EA801600
EA801800
EA802000
EA802200
EA802400
EA802600
EA802800
EA803000
EA803200
EA803400
EA803600
EA803800
EA804000
EA804200
EA804400
EA804600
EA804800
EA805000
EA805200
EA805400
EA805600
EA805800
EA806000
EA401000
EA401200
EA401400
EA401600
EA401800
EA402000
EA402200
EA402400
EA402600
EA402800
EA403000
EA403200
EA403400
EA403600
EA403800
EA404000
EA404200
EA404400
EA404600
EA404800
EA405000
EA405200
EA405400
EA405600
EA405800
EA406000
EA406200
EA406400
EA406600
EA406800
EA407000
EA407200

```


ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
 CELAIR - EPN SOURCE STATEMENT - (PN18) -

02/16/68

PAGE 3

IF (I1 .NE. J1) GO TO 14700	E4407400
IF (I2 .EQ. J2) GO TO 18900	E4407600
14700 IF (ABS(VX1-VY1).GT.1.E-8)GO TO 18000	E4407800
IF (ISURFC(1,I1) .NE. ISURFC(1,J1)) GO TO 18000	E4408000
IF (SURFC(11,I1) .NE. SURFC(11,J1)) GO TO 18000	E4408200
IF (SURFC(12,I1) .NE. SURFC(12,J1)) GO TO 18000	E4408400
IF (ABS(VX2-VY2).GT.1.E-8)GO TO 18000	E4408600
IF (ISURFC(1,I2) .NE. ISURFC(1,J2)) GO TO 18000	E4408800
IF (SURFC(11,I2) .NE. SURFC(11,J2)) GO TO 18000	E4409000
IF (SURFC(12,I2) .EQ. SURFC(12,J2)) GO TO 18800	E4409200
18000 CONTINUE	E4409400
NELMT = NELMT + 1	E4501000
ELM1TX(1,NELMT) = I1	E4501200
ELM1TX(2,NELMT) = I2	E4501400
18800 CONTINUE	E4501600
C	
19000 IF (NAC .EQ. 0) GO TO 32000	E4502000
NAIRS = 1	E4502200
AIRMTX(1,1) = ACURL(1,1)	E4502400
AIRMTX(2,1) = ACURL(2,1)	E4502600
IF (NAC .EQ. 1) GO TO 32000	E4502800
DO 17800 I = 2, NAC	E4503000
I1 = ACURL(1,I)	E4503200
I2 = ACURL(2,I)	E4503400
VX1 = VTXCRD(I1)	E4503600
VX2 = VTXCRD(I2)	E4503800
IF (VX1 .LE. VX2) GO TO 16800	E4504000
IT = I1	E4504200
I1 = I2	E4504400
I2 = IT	E4504600
VX1 = VX2	E4504800
VX2 = VX1	E4505000
VX2 = VX1	E4505200
17800 DO 17000 J = 1, NAIRS	E4505400
J1 = AIRMTX(1,J)	E4505600
J2 = AIRMTX(2,J)	E4505800
VY1 = VTXCRD(J1)	E4506000
VY2 = VTXCRD(J2)	E4506200
IF (I1 .NE. J1) GO TO 16700	E4506400
IF (I2 .EQ. J2) GO TO 17800	E4506600
16700 IF (ABS(VX1-VY1).GT.1.E-8)GO TO 17000	E4506800
IF (ABS(VX2-VY2).LE.1.E-8)GO TO 17800	E4507000
17000 CONTINUE	E4507200
NAIRS = NAIRS + 1	E4507400
AIRMTX(1,NAIRS) = I1	E4507600
AIRMTX(2,NAIRS) = I2	E4507800
17800 CONTINUE	E4508000
C	
C END OF CELAIR SUBROUTINE	E4609000
C	
32000 RETURN	E4609200
END	E4609400

ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
 CELAIR STORAGE MAP

08/16/68

PAGE 6

SUBROUTINE CELAIR
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCTHL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00007	I	PUNCID	00032	R	DATA	00033	R
LN DOE	00033	I	NRAYS	00034	I	NSLCS	00038	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	INODE	00043	I
NSPLN	00044	I	NOSJH	00045	I	NSOPE	00046	I
PLAS	00047	I	DELY	00080	R	PNUMB	00081	R
PLNSM	00082	R	WFLGH	00083	R	ZETA	00084	R
WEXPP	00085	R	DEXPP	00086	R	WEXPP	00087	R
D P LN	00080	R	OMGAR	00081	R	OMGAI	00082	R
DELD	00083	R	EPRAD	00084	R	PSCAL	00085	R
OMGAF	00086	R	SPFEA	00087	R	DUMIN	00070	R
OMIST	00071	R	MO	00072	R	DELM	00073	R
SYBHX	00074	R	WXDIR	00075	R	WYDIR	00076	R
NOTAN	00077	R	NOSGN	00100	I	NOSGV	00104	I
WOBJH	00117	R	WCLRH	00126	R	WINHT	00138	R
EIMHT	00144	R	CIMPL	00153	R	WCLR8	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IOESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUND8	08210	R
IBND8	08210	I	NCOND	08665	I			
COMMON BLOCK			PRNT	ORIGIN	08667	LENGTH	00008	
LINE	07000	I	PAGE	00001	I			
COMMON BLOCK			ELNAIR	ORIGIN	08671	LENGTH	00766	
NELMT	00000	I	NAIRS	00001	I	ELHMTX	00002	I
AIRMTX	00312	I	VTXCRD	00822	R			
DIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ECUNL	07667	I	ACURL	10167	I			
UNDIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SPRIME	10477	R	I	10500	I	ZERO	10501	R
NEC	10502	I	NAC	10503	I	J	10504	I
X1	10505	R	X2	10506	R	I1	10507	I
I2	10510	I	VX1	10511	R	VX2	10512	R
IT	10513	I	VY1	10514	R	J1	10515	I
J2	10516	I	VY2	10517	R	VY2	10520	R
ENTRY POINTS								

ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
 CELAIZ STORAGE MAP

02/16/68

PAGE 7

CELAIZ SECTION 9

SUBROUTINES CALLED

.FWRD. SECTION 10
 .FFIL. SECTION 13
 .FRET. SECTION 16

PRTCTL SECTION 11
 .FCNV. SECTION 14

.UNDS. SECTION 12
 SYSLOC SECTION 15

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	5A	10873	32010	FORMAT	10840	1500	9A	10714
32020	FORMAT	10850	2500	15A	10745	2000	13A	10725
32030	FORMAT	10862	3500	35A	11014	3000	32A	11011
32040	FORMAT	10874	32050	FORMAT	10805	32060	FORMAT	10824
4500	53A	11111	4000	51A	11077	32070	FORMAT	10833
32075	FORMAT	10836	5000	66A	11145	7500	98A	11264
6000	79A	11205	9500	78A	11172	32080	FORMAT	10844
6500	89A	11237	13500	192A	11575	7000	93A	11250
10000	189A	11572	8000	111A	11317	9000	165A	11523
8000	145A	11443	8200	125A	11355	8100	120A	11335
8400	137A	11405	8210	132A	11371	8800	161A	11511
8710	156A	11474	8700	151A	11454	9200	178A	11547
9400	185A	11561	14000	201A	11613	16000	284A	12116
15500	281A	12113	14500	225A	11701	15000	276A	12072
14700	240A	11762	32000	335A	12305	17500	332A	12302
16500	308A	12202	17000	327A	12261	16700	321A	12237

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12344.

TR-67-700-10-2
Page III-2-14
Program Listing
Link 2

INF.N010001.44800-0.77387.AM IC

IBJOB LINK1 7094 A

02/16/68

PAGE 8

*
STITLE ROUTINE TO FIND POINT OF CLOSEST APPROACH OF TWO CURVES
SIBFTC CLOSSZ M84.XR7 LINK 2 (CLOSS) CLO01000

ROUTINE TO FIND POINT OF CLOSEST APPROACH OF TWO CURVES
CLOSSZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 1

```

SUBROUTINE CLOSS( I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX, DZMIN,
1 Z1CLOS, Y1CLOS, Z2CLOS, * )
C
C *** INPUTS ***
C I1 SURFACE NUMBER OF FIRST SURFACE.
C Z1MIN MINIMUM Z VALUE OF FIRST SURFACE.
C Z1MAX MAXIMUM Z VALUE OF FIRST SURFACE.
C I2 SURFACE NUMBER OF SECOND SURFACE.
C Z2MIN MINIMUM Z VALUE OF SECOND SURFACE.
C Z2MAX MAXIMUM Z VALUE OF SECOND SURFACE.
C DZMIN MINIMUM Z SURFACE SEPARATION.
C
C *** OUTPUTS ***
C Z1CLOS REVISED Z VALUE ON FIRST SURFACE.
C Y1CLOS Y VALUE CORRESPONDING TO Z1CLOS AND Z2CLOS.
C Z2CLOS REVISED Z VALUE ON SECOND SURFACE.
C * ALTERNATE RETURN FOR NON-INTERSECTING SURFACES.
C
C *** DESCRIPTION ***
C THIS SUBROUTINE DETERMINES THE POINTS ( Z1CLOS, Y1CLOS ) AND
C ( Z2CLOS, Y1CLOS ) ON INTERSECTING SURFACES I1 AND I2 SUCH
C THAT THE DISTANCE BETWEEN THEM IS DZMIN.
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID,
1 DATA(3483)
DIMENSION MOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7),
1 WCLRS(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100),
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26),
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100)
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAY), (DATA(3),
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6),
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9),
3 IMODE), (DATA(10), NSPLN), (DATA(11), MOBJH), (DATA(12),
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15),
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18),
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21),
7 WEXPP), (DATA(22), DLPLN), (DATA(23), ONGA2), (DATA(24),
8 ONGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27),
9 PSCAL), (DATA(28), ONGAF), (DATA(29), SPFEA), (DATA(30),
A DUMIN), (DATA(31), ODIST), (DATA(32), HD), (DATA(33),
B DELH), (DATA(34), SYSX), (DATA(35), WDIR), (DATA(36),
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42),
D NDSGV(1)), (DATA(53), MOBJH(1)), (DATA(60), WCLRH(1)),
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81),
F CIMPL(1)), (DATA(86), WCLRS(1)),
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1),
H ISURFC(1,1)), (DATA(2102), DESGN(1,1), IDESGN(1,1)),
I (DATA(2602), SUBST(1), ISUBST(1)), (DATA(3102), BOUNDS(1,1),
J IBNDS(1,1)), (DATA(3483), NCOND)
INTEGER CNTRL, AFLAG, DATE
REAL LATT
C
COMMON/ELMAIR/NELMT,NAIRS,ELMNTX(2,100),AIRMTX(2,100),VTXCRD(100)
INTEGER ELMNTX, AIRMTX

```

CLO01500
CLO02000
CLO02100
CLO02200
CLO02300
CLO02400
CLO02500
CLO02600
CLO02700
CLO02800
CLO03000
CLO03100
CLO03200
CLO03300
LC003400
CLO03500
CLO03600
CLO03700
CLO03800
DT001000
DT001100
DT002000
DT002100
DT002200
DT002300
DT101000
DT101100
DT101200
DT101300
DT101400
DT101500
DT101600
DT101700
DT101800
DT101900
DT102000
DT102100
DT102200
DT102300
DT102400
DT002500
DT102600
DT102700
DT102800
DT102900
DT201000
DT202000
EL001000
EL002000

ROUTINE TO FIND POINT OF CLOSEST APPROACH OF TWO CURVES
CLOSSZ - EFN SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 2

C	START OF CLOSS SUBROUTINE	CL004000	
C			
	Z1CLOS = 0.	CL005000	
	Y1CLOS = 0.	CL005200	
	Z2CLOS = 0.	CL005400	
	IF(I1 .NE. I2) GO TO 2500	CL005600	
C			
C	SURFACES ARE IDENTICAL, NO INTERSECTION POSSIBLE	CL005700	
	2000 RETURN 1	CL005800	
C	CHECK FOR INTERSECTION	CL006000	
	2500 CALL CROSS(I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX, ZI, YI, \$2000)	CL006200	7
	S = VTXCRD(I2) - VTXCRD(I1)	CL006400	
	IT1 = 1SURFC(1,I1)	CL006600	
	R1 = SURFC(1,I1)	CL006800	
	B1 = SURFC(12,I1)	CL007000	
	AL1 = B1 - 1.	CL007200	
	IT2 = 1SURFC(1,I2)	CL008000	
	R2 = SURFC(11,I2)	CL008200	
	B2 = SURFC(12,I2)	CL008400	
	AL2 = B2 - 1.	CL008600	
C			
	IF(IT1 .EQ. 3) GO TO 4000	CL201000	
	IF(R1 .NE. 0.) GO TO 5000	CL201500	
	4000 Z2CLOS = DZMIN	CL202000	
	Z = Z2CLOS - S	CL202200	
	Y1CLOS = SQRT(AL2 * Z * Z + 2. * R2 * Z)	CL202400	27
	GO TO 32000	CL202600	
	5000 IF(IT2 .EQ. 3) GO TO 6000	CL203000	
	IF(R2 .NE. 0.) GO TO 7000	CL203200	
	6000 Z2CLOS = S	CL204000	
	Z1CLOS = S - DZMIN	CL214200	
	Y1CLOS = SQRT(AL1 * Z1CLOS * Z1CLOS + 2. * R1 * Z1CLOS)	CL204400	36
	GO TO 32000	CL204600	
	7000 ZP = 0.	CL205000	
	ZM = ZI	CL205200	
	7500 ZT = (ZP + ZM) / 2.	CL205400	
	YT = SQRT(AL1 * ZT * ZT + 2. * R1 * ZT)	CL206000	40
	TEMP = SQRT(R2 * R2 + AL2 * YT * YT)	CL206200	41
	IF(R2 .LT. 0.) TEMP = - TEMP	CL206400	
	Z2 = S + (YT * YT) / (R2 + TEMP)	CL206600	
	GT = Z2 - ZT - DZMIN	CL206800	
	IF(GT) 9000, 11000, 8000	CL207000	
	8000 ZP = ZT	CL207200	
	GO TO 10000	CL207400	
	9000 ZM = ZT	CL208000	
	10000 ECURL = ABS(ZP - ZM)	CL208200	
	IF(ZM .NE. 0.) ECURL = ECURL / ABS(ZM)	CL208400	
	IF(ECURL .GE. 1.E-6) GO TO 7500	CL208600	
	11000 Z1CLOS = ZT	CL209000	
	Y1CLOS = YT	CL209200	
	Z2CLOS = Z2	CL209400	
	32000 Z2CLOS = Z2CLOS - S	CL401000	
C			
C	END OF CLOSS SUBROUTINE	CL402000	
	RETURN	CL403000	
	END	CL404000	

ROUTINE TO FIND POINT OF CLOSEST APPROACH OF TWO CURVES
CLOSSZ STORAGE MAP

02/18/68

PAGE 3

SUBROUTINE CLOSS
COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	INDOE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSUBT	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HO	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
NOBJH	00117	R	WCLRH	00126	R	WIMWT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
TSURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I			

COMMON BLOCK			ELMAIR	ORIGIN	08667	LENGTH	00766	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RELMT	00000	I	NAIRS	00001	I	ELMNTX	00002	I
AIRMTX	00312	I	VTXCRD	00822	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ZI	07655	R	YI	07656	R	S	07657	R
IT1	07660	I	R1	07661	R	B1	07662	R
AL1	07663	R	IT2	07664	I	R2	07665	R
BE	07666	R	AL2	07667	R	Z	07670	R
ZP	07671	R	ZM	07672	R	ZT	07673	R
YT	07674	R	TEMP	07675	R	Z2	07676	R
GT	07677	R	ECURL	07700	R			

ENTRY POINTS

CLOSS SECTION 7

SUBROUTINES CALLED

CROSS SECTION 8
.FRET. SECTION 11

SORT SECTION 9

SYSLOC SECTION 10

ROUTINE TO FIND POINT OF CLOSEST APPROACH OF TWO CURVES
CLOSSZ STORAGE MAP

02/16/68

PAGE 4

			EFN	IFN	CORRESPONDENCE			
EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
2000	6A	07726	2000	5A	07723	4000	26A	10007
3000	29A	10035	32000	55A	10227	6000	35A	10045
7000	38A	10073	7500	39A	10076	9000	47A	10173
11000	54A	10221	8000	45A	10170	10000	48A	10179

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 10343.

TR-67-700-10-2
Page III-2-19
Program Listing
Link 2

INF.N010001.44000-0.77307.AM IC

18JOB LINK1 7094 A 02/10/68

PAGE 3

00
STITLE ROUTINE TO PLOT Z-Y PROFILE ON SC4020 CP001000
SIBP TC CLOTZ M94,XR7 LINK 2 (CLOT)

ROUTINE TO PLOT Z-Y PROFILE ON 8402B
CPL0T2 - EFM SOURCE STATEMENT - (FN.8) -

02/16/68

PAGE 1

```

SUBROUTINE CPL0T( ZMIN, ZMAX )                                C P001100
C
C *** INPUTS ***                                             C P001200
C ZMIN  MINIMUM Z OF SYSTEM RELATIVE TO ENTRANCE PUPIL.    C P001250
C ZMAX  MAXIMUM Z OF SYSTEM RELATIVE TO ENTRANCE PUPIL.    C P001300
C ZLST  Z COORDINATES OF POINTS ON CURRENT ELEMENT READ FROM BINARY C P001350
C       TAPE ( LOGICAL 11 ).                                C P001400
C YLST  Y COORDINATES OF POINTS ON CURRENT ELEMENT READ FROM BINARY C P001450
C       TAPE ( LOGICAL 11 ).                                C P001500
C NPTS  NUMBER OF POINTS FOR CURRENT ELEMENT, TAPE LOGICAL 11. C P001550
C
C *** OUTPUTS ***                                           C P001600
C PLOT FRAME CONTAINING Z-Y CROSS SECTION OF LENS SYSTEM FROM C P001650
C THE ENTRANCE PUPIL TO THE IMAGE SURFACE WITH OPTIC AXIS SHOWN. C P001700
C THE SCALE IS THE SAME IN THE Z AND Y DIRECTIONS WITH THE Y    C P001750
C RANGE FROM YMAXX TO -YMAXX. A SCALED RULER IS POSITIONED      C P001800
C BELOW THE LENS WITH ASSOCIATED LABELS. BELOW THIS THE TITLE  C P001850
C AND DATE ARE PLOTTED. TWO SPECIAL RAYS AT THE LIMITS OF THE  C P001900
C ENTRANCE PUPIL ARE ALSO TRACED ON THE LENS PLOT.            C P001950
C A SECOND FRAME CONTAINING THE PLOT PARAMETERS AND THE VALUES C P002000
C FOR SURFACE RADIUS, SURFACE DISTANCE, SURFACE ECCENTRICITY,  C P002050
C AND THE INTERSECTIONS OF THE SPECIAL RAYS WITH EACH SURFACE  C P002100
C IS ALSO DRAWN.                                             C P002150
C
COMMON / DATA / NCNTRL, CNTRL(100), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483) DT001100
DIMENSION WOBJH(7), WCLRH(7), WIMHT(7), EINHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IOESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NS'BP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMC2), (DATA(24), DT101700
8 OMC1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMCAP), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), HO), (DATA(33), DT102000
B DELH), (DATA(34), SYSRX), (DATA(35), WDIR), (DATA(36), DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLRH(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EINHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(202), DESGN(1,1), IOESGN(1,1)), DT102700
I (DATA(202), SUBST(1), ISUBST(1)), (DATA(302), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
COMMON/ELHATR/NELMT,NAIRS,ELMNTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
INTEGER ELMNTX, AIRMTX EL002000

```

ROUTINE TO PLOT Z-Y PROFILE ON SC4020
CPL0T2 - EFM SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 2

```

C      COMMON / COORNS / ZHAT(2,100), YHAT(2,100), CHAT(2,100)      CP001000
      INTEGER CHAT      CD002000

C      COMMON / PLOTG / YMAXX, DZMIN, NPOINT      PC001000

C      EQUIVALENCE ( ZHAT(1,1), ZLST(1) ), ( YHAT(1,51), YLST(1) )      CP002400

C      DIMENSION SC(4), CD(4), BUF(500), ZLST(300), YLST(300), NUM(9,10), CP002500
      BUFR(100), SCR(4), CDR(4)      CP002600

C      DATA NB / 500 /, SC / 70., 953., 1023., 0. /, SCR / 70., 1023., CP003000
      1 1023., 0. /, NBR / 100 /      CF 003080

C      DATA NZERO / 3H0.0 /,      CP003100
      1 NUM(1,1) / 3H0.1 /,      CP003190
      2(NUM(1,2), I=1,2)      CP003200
      3 / 3H0.2, 3H0.4 /,
      3(NUM(1,3), I=1,3)      CP003250
      C / 3H0.5, 3H0.6, 3H0.9 /,
      4(NUM(1,4), I=1,4)      CP003300
      D / 3H0.4, 3H0.8, 3H1.2, 3H1.6 /,
      5(NUM(1,5), I=1,5)      CP003350
      E / 3H0.5, 3H1.0, 3H1.5, 3H2.0, 3H2.5 /,
      6(NUM(1,6), I=1,6)      CP003400
      F / 3H0.6, 3H1.2, 3H1.8, 3H2.4, 3H3.0, 3H3.6 /,
      7(NUM(1,7), I=1,7)      CP003450
      G / 3H0.7, 3H1.4, 3H2.1, 3H2.8, 3H3.5, 3H4.2, 3H4.9 /,
      8(NUM(1,8), I=1,8)      CP003500
      H / 3H0.8, 3H1.6, 3H2.4, 3H3.2, 3H4.0, 3H4.8, 3H5.6, 3H6.4 /,
      9(NUM(1,9), I=1,9)      CP003550
      I / 3H0.9, 3H1.8, 3H2.7, 3H3.6, 3H4.5, 3H5.4, 3H6.3, 3H7.2, 3H8.1 /,
      X(NUM(1,10), I=1,10)      CP003600
      J / 3H1.0, 3H2.0, 3H3.0, 3H4.0, 3H5.0, 3H6.0, 3H7.0, 3H8.0 /

C      START OF CPLOT SUBROUTINE      CP004000

C      DETERMINE SCALE OF PLOT      CP004500
      ALENG1 = 2. * YMAXX      CP004600
      ALENG2 = ZMAX - ZMIN      CP004800
      IF ( ALENG1 .GT. ALENG2 ) GO TO 1000      CP005200

C      LENS LENGTH GREATER THAN HEIGHT      CP006000
      CD(1) = ZMIN      CP006200
      CD(2) = - ALENG2 / 2.      CP006400
      CD(3) = ZMAX      CP006600
      CD(4) = ALENG2 / 2.      CP006800
      GO TO 1500      CP007000

C      LENS HEIGHT GREATER THAN LENGTH      CP008000
      1000 CD(1) = ZMIN - ( ALENG1 - ALENG2 ) / 2.      CP008200
      CD(2) = - YMAXX      CP008400
      CD(3) = ZMAX + ( ALENG1 - ALENG2 ) / 2.      CP008600
      CD(4) = YMAXX      CP008800
      1500 RSCALE = ( CD(3) - CD(1) ) / 953.      CP009000
      DO 1600 I = 1, 4      CP009200

```

ROUTINE TO PLOT Z-Y PROFILE ON SC4020
CPLTZ - EFM SOURCE STATEMENT - (FN18) -

02/16/68

PAGE 3

1600	CDR(1) = CD(1)	CP009400	
	CDR(2) = CDR(2) - RSCALE * 70.	CP009400	
C			
C	INITIALIZE FOR PLOTTING	CP101000	
	CALL CAMERA(NT, 2)	CP101200	16
	CALL SLABEL	CP101400	18
	CALL SDINIT(BUFR, NBR, CDR, SCR, NT, BUF, NB, CD, SC, NT)	CP101600	
C			
C	COMPUTE RULER FOR LENS PLOT	CP102000	20
	FNLENG = ALOG10(CD(3) - ZMIN)	CP102100	22
	NLENG = FNLENG	CP102200	
	IF(FNLENG .LT. 0.) NLENG = NLENG - 1	CP102220	
	FNORM = (CD(3) - ZMIN) / 10. ** NLENG	CP102240	26
	NORM = FNORM	CP102260	
C			
C	DRAW RULER LENGTH AT 960 RASTERS FROM ZMIN TO RULX2	CP102400	
	RULX2 = ZMIN + (FLOAT(NORM)/FNORM) * (CD(3) - ZMIN)	CP102450	
	RULY = CDR(2) - RSCALE * 7.	CP102500	
C			
	CALL SDNPUT(ZMIN, RULY, BUFR, 0)	CP102550	28
	CALL SDNPUT(RULX2, RULY, BUFR, 0)	CP102600	30
	CALL STERM(NT, BUFR)	CP102650	
C			
C	DRAW 11 TICK MARKS ALONG RULER	CP102700	32
	RULYT = RULY - RSCALE * 10.	CP102750	
	RULXTI = (RULX2 - ZMIN) / 10.	CP102800	
	RULYTS = RULY - RSCALE * 9.	CP102820	
C			
C	LABEL TICK MARKS WITH APPROPRIATE INCREMENTS	CP102850	
	DO 1700 I = 1, 11	CP103000	
	XTICK = ZMIN + FLOAT(I-1) * RULXTI	CP103050	
	CALL SDNPUT(XTICK, RULY, BUFR, 0)	CP103100	37
	CALL SDNPUT(XTICK, RULY, BUFR, 0)	CP103150	39
	LABEL = NUM(I-1, NORM)	CP103200	
	IF(I .EQ. 1) LABEL = NZERO	CP103250	
	IF(I .GT. NORM+1) LABEL = NUM(NORM, I-1)	CP103300	
	XLABEL = XTICK * BUFR(11) * BUFR(12) - 10.	CP103350	
	CALL SPRTA(NT, XLABEL, 965, +8, LABEL, 1)	CP103400	43
	CALL STERM(NT, BUFR)	CP103405	50
	IF(I .EQ. 11) GO TO, 1700	CP103410	
C			
C	DRAW 4 TICK MARKS, 3 RASTERS LONG, BETWEEN MAJOR TICKS	CP103415	
	DO 1650 J = 1, 4	CP103420	
	XTICKS = XTICK + FLOAT(J) * (RULXTI / 5.)	CP103430	
	CALL SDNPUT(XTICKS, RULY, BUFR, 0)	CP103440	57
	CALL SDNPUT(XTICKS, RULY, BUFR, 0)	CP103450	59
	CALL STERM(NT, BUFR)	CP103455	61
	1650 CONTINUE	CP103460	
	1700 CONTINUE	CP103500	
C			
C	DRAW OPTIC AXIS	CP104000	
	CALL SDNPUT(CD(1), 0., BUF, 0)	CP104200	67
	CALL SDNPUT(CD(3), 0., BUF, 0)	CP104400	69
	CALL STERM(NT, BUF)	CP104600	
C			
C	BEGIN LOOP TO PLOT ELEMENTS	CP105000	71

ROUTINE TO PLOT Z-Y PROFILE ON SC4020
 CPLOTZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 4

DO 4000 I = 1, NELMT	CP105200	
READ (11) NPTS, ZLST, YLST	CP105300	
C		
C PLOT UPPER HALF OF ITH. ELEMENT	CP105600	78
DO 2000 J = 1, NPTS	CP105800	
2000 CALL SDMPUT(ZLST(J), YLST(J), BUF, 0)	CP106000	84
CALL STERN(NT, BUF)	CP106200	
C		
C PLOT LOWER HALF OF ITH. ELEMENT	CP106400	87
DO 3000 J = 1, NPTS	CP106600	
3000 CALL SDMPUT(ZLST(J), -YLST(J), BUF, 0)	CP106800	94
CALL STERN(NT, BUF)	CP107000	97
4000 CONTINUE	CP107200	
C		
C IMAGE PLANE	CP108000	
READ (11) NPTS, ZLST, YLST	CP108200	100
IF(NPTS .LE. 0) GO TO 6000	CP108400	
DO 5000 J = 1, NPTS	CP108600	
5000 CALL SDMPUT(ZLST(J), YLST(J), BUF, 0)	CP108800	113
CALL STERN(NT, BUF)	CP201000	117
DO 5500 J = 1, NPTS	CP201200	
5500 CALL SDMPUT(ZLST(J), -YLST(J), BUF, 0)	CP201400	126
CALL STERN(NT, BUF)	CP201600	
C		
C ENTRANCE PUPIL	CP202000	130
6000 READ (11) NPTS, ZLST, YLST	CP202200	131
CALL SDMPUT(ZLST(1), YLST(1), BUF, 0)	CP202400	139
CALL SDMPUT(ZLST(2), YLST(2), BUF, 0)	CP202600	137
CALL STERN(NT, BUF)	CP202800	139
CALL SDMPUT(ZLST(1), -YLST(1), BUF, 0)	CP203000	141
CALL SDMPUT(ZLST(2), -YLST(2), BUF, 0)	CP203200	143
CALL STERN(NT, BUF)	CP203400	
C		
C RAY 1 TRACE	CP204000	145
READ (11) NPTS, ZLST, YLST	CP204200	146
IF(NPTS .LE. 0) GO TO 7500	CP204300	
DO 7000 J = 1, NPTS	CP204400	
7000 CALL SDMPUT(ZLST(J), YLST(J), BUF, 0)	CP204600	159
CALL STERN(NT, BUF)	CP204800	
C		
C RAY 2 TRACE	CP205000	163
7500 READ (11) NPTS, ZLST, YLST	CP205200	164
IF(NPTS .LE. 0) GO TO 8000	CP205300	
DO 8000 J = 1, NPTS	CP205400	
8000 CALL SDMPUT(ZLST(J), YLST(J), BUF, 0)	CP205600	
C		
C LABEL PLOT FRAME WITH TITLE AND DATE	CP205800	177
9000 CALL SETPL(990)	CP206000	182
WRITE (10, 32100) (TITLE(I), I=1,12), (DATE(J), J=1,3)	CP207000	183
32100 FORMAT(20X, 12A6, 6X, 3I4)	CP207200	
CALL STERN(NT, BUF)	CP207400	196
CALL ADV(NT)	CP207600	196
CALL RESPL	CP207800	
C		
C DRAW INFORMATION PLOT FRAME	CP301000	
C LABEL PLOT FRAME WITH TITLE AND DATE	CP301200	200

ROUTINE TO PLOT Z-Y PROFILE ON SC4020
CPLOTZ - EFM SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 5

CALL SETPL(990)	CP301400	202
WRITE (18, 32100) (TITLE(I), I=1,12), (DATE(J), J=1,3)	CP301600	203
CALL SETPL(9)	CP302000	216
WRITE (18, 32150) YMAXX, DZMIN, NPOINT, NLENG	CP302200	217
32150 FORMAT(5X, 13HTUBE RADIUS =, F5.2, 5X, 20MINIMUM SEPARATION =,	CP302400	
1 F6.3, 5X, 17H ELEMENT POINTS =, I4, 5X, 16HPLOT SCALE = 10E, I2)	CP302600	
WRITE (18, 32160)	CP303000	218
32160 FORMAT(10D, 2X, 7HSURFACE, 12X, 1HR, 17X, 1HD, 17X, 1HD, 16X,	CP303200	
1 2HM1, 16X, 2HR2)	CP303400	
C		
REWIND 11	CP304000	219
NREAD = NELMT + 2	CP304200	
DO 15000 I = 1, NREAD	CP304400	
15000 READ (11) DUMMY	CP304600	
C		
READ RAY 1 DATA	CP305000	224
READ (11) NPTS, ZLST, ZLST	CP305200	
C		
BEGIN LOOP TO PLOT SURFACE PARAMETERS	CP306000	226
LRAY = 2	CP306200	
DO 16000 I = 1, NELMT	CP306400	
DO 16000 J = 1, 2	CP306600	
NEL = ELMNMX(J,I)	CP306800	
NELN = ELMNMX(2,I)	CP307000	
IF(J .EQ. 2) NELN = ELMNMX(1,I+1)	CP307200	
DTEMP = SURFC(4,NELN)	CP307300	
IF(NEL .EQ. NSURF) DTEMP = DELD	CP307400	
WRITE (18, 32170) NEL, SURFC(11,NEL), DTEMP,	CP307500	
1 SURFC(12,NEL), ZLST(NEL+1), YLST(NEL+1)	CP307600	245
32170 FORMAT(5X, I2, 5X, E17.7, 4E18.7)	CP307800	
IF(NEL .EQ. NSURF) GO TO 18000	CP308000	
16000 LRAY = LRAY + 1	CP401000	
IF(LRAY .LE. 51) GO TO 16000	CP401200	
C		
BEGIN NEW PLOT FRAME	CP401400	
CALL STERN(NT, BUF)	CP401600	259
CALL ADV(NT)	CP401800	261
CALL RESPL	CP402000	263
CALL SETPL(990)	CP402400	265
WRITE (18, 32100) (TITLE(K), K=1,12), (DATE(K), K=1,3)	CP402600	266
CALL SETPL(20)	CP403000	275
WRITE (18, 32160)	CP403200	276
18000 CONTINUE	CP404000	
CALL STERN(NT, BUF)	CP404200	280
CALL ADV(NT)	CP404400	282
CALL RESPL	CP404600	
C		
END OF CPLOT SUBROUTINE	CP408000	284
RETURN	CP408200	
END	CP408400	

ROUTINE TO PLOT Z-Y PROFILE ON 8C4020
 CPLOTZ

STORAGE MAP

02/16/68

PAGE 6

SUBROUTINE CPLOT
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	TMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
HEKPP	00055	R	DEKPP	00056	R	WEXPP	00057	R
OLPLN	00058	R	ONGA2	00061	R	ONGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
ONGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	00071	R	HD	00072	R	DELH	00073	R
SYMX	00074	R	WDIR	00075	R	WDIR	00076	R
ROTAN	00077	R	NDSCN	00100	I	NDJGV	00104	I
WOB JH	00117	R	WCLRH	00126	R	WTMNT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	08210	R
IBND8	08210	I	NCOND	08665	I			
COMMON BLOCK			ELMATR	ORIGIN	08667	LENGTH	00766	
MELMT	00000	I	NAIRS	00001	I	ELMNTX	00002	I
AIRMTX	00312	I	VTXCRD	00622	R			
COMMON BLOCK			CODRNG	ORIGIN	07655	LENGTH	01130	
ZHAT	00000	R	YHAT	00310	R	CHAT	00620	I
ZLST	00000	R	YLST	00454	R			
COMMON BLOCK			PLOTG	ORIGIN	11005	LENGTH	00003	
YMAXX	00000	R	DZMIN	00001	R	NPOINT	00002	I
DIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SC	11010	R	CD	11014	R	BUF	11020	R
NUM	12004	I	BUFR	12136	R	SCR	12302	R
CDR	12308	R						
UNDIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ALENG1	12312	R	ALENG2	12313	R	RSCALE	12314	R
I	12315	I	NT	12316	I	NBR	12317	I
NB	12320	I	FNLENG	12321	R	NLENG	12322	I

ROUTINE TO PLOT Z-Y PROFILE ON RC4020
CPL0T2

02/16/68

PAGE 7

FNORM	12323	P
NULY	12326	R
NULYTS	12331	R
NZERO	12334	I
XTICKS	12337	R
DUMMY	12342	R
NELN	12348	I

NORM	12324	I
NULYT	12327	R
XTICK	12332	R
XLABEL	12339	R
NPTS	12340	I
LRAY	12343	I
DTEMP	12346	R

RULK2	12329	R
RULKTI	12330	R
LABEL	12333	I
J	12336	I
NREAD	12341	I
NEL	12344	I

STORAGE MAP

ENTRY POINTS

CPL0T	SECTION	11
-------	---------	----

SUBROUTINES CALLED

CAMERA	SECTION	12
ALOG10	SECTION	15
STEM	SECTION	18
.FBLI.	SECTION	21
ADV	SECTION	24
.UN11.	SECTION	27
.FBLT.	SECTION	30
.FCNV.	SECTION	33
E.3	SECTION	36
CC.2	SECTION	39
SYSLOC	SECTION	42

SLABEL	SECTION	13
.XP2.	SECTION	16
SPRNTA	SECTION	19
SETPL	SECTION	22
RESPL	SECTION	25
.FRLR.	SECTION	28
.UN10.	SECTION	31
E.1	SECTION	34
E.4	SECTION	37
CC.3	SECTION	40

SDINIT	SECTION	14
SDNPUT	SECTION	17
.FNDS.	SECTION	20
.FWRD.	SECTION	23
.FRMT.	SECTION	26
.FFIL.	SECTION	29
E.2	SECTION	32
CC.1	SECTION	35
CC.4	SECTION	41

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION
1000	6A	1250B
1700	64A	1310B
2000	81A	13167
5000	110A	13275
7000	156A	13475
32100	FORMAT	12408
15000	284A	13721
32170	FORMAT	12451

EFN	IFN	LOCATION
1500	7A	12527
1650	62A	13104
3000	91A	13217
5500	123A	13325
9000	181A	13572
32150	FORMAT	12407
18000	277A	14153

EFN	IFN	LOCATION
1600	11A	12534
4000	98A	13242
6000	131A	13350
7500	164A	13521
8000	174A	13553
32160	FORMAT	12434
16060	253A	14087

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 14234.

TR-67-700-10-2
Page III-2-27
Program Listing
Link 2

INF.ND10001.44000-0.77307.AM IC IDJOB LINK1 7004 A 02/16/68 PAGE 8

80
STITLE ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS
STEP TC CROSS2 M04.XR7 LINK 2 (CROSS) CROD1000

ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS
CROSSZ .. EFM SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 1

```

SUBROUTINE CROSS( I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX, ZCROSS,
1 YCROSS, * )
C
C   *** INPUTS ***
C I1   SURFACE NUMBER OF FIRST SURFACE.
C Z1MIN MINIMUM Z VALUE OF FIRST SURFACE.
C Z1MAX MAXIMUM Z VALUE OF FIRST SURFACE.
C I2   SURFACE NUMBER OF SECOND SURFACE.
C Z2MIN MINIMUM Z VALUE OF SECOND SURFACE.
C Z2MAX MAXIMUM Z VALUE OF SECOND SURFACE.
C
C   *** OUTPUTS ***
C ZCROSS Z VALUE OF INTERSECTION.
C YCROSS Y VALUE OF INTERSECTION.
C *     ALTERNATE RETURN FOR NON-INTERSECTING SURFACES.
C
C   *** DESCRIPTION ***
C THIS SUBROUTINE DETERMINES THE INTERSECTION POINT, ( YCROSS,
C ZCROSS ), OF SURFACE I1 AND SURFACE I2.
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID,
1 DATA(3483)
DIMENSION WOBJH(7), WCLR(7), WINHT(7), EIMHT(7), CIMPL(7),
1 WCLRS(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100),
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26),
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100)
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3),
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6),
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9),
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12),
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15),
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18),
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21),
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMGA2), (DATA(24),
8 OMGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27),
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30),
A DUMIN), (DATA(31), ODIST), (DATA(32), HO), (DATA(33),
B DELH), (DATA(34), SYSHX), (DATA(35), WXDIR), (DATA(36),
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42),
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLR(1)),
E (DATA(67), WINHT(1)), (DATA(74), EIMHT(1)), (DATA(81),
F CIMPL(1)), (DATA(88), WCLRS(1)),
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(182), SURFC(1,1),
H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)),
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1),
J IBNDS(1,1)), (DATA(3483), NCOND)
INTEGER CNTRL, AFLAG, DATE
REAL LATT
C
COMMON/ELMAIR/NELMT,NAIRS,ELMHTX(2,100),AIRMTX(2,100),VTXCRD(100)
INTEGER ELMHTX, AIRMTX
C
C   START OF CROSS SUBROUTINE
C
ZCROSS = 0.

```

CR001500
CR001600
CR002000
CR002100
CR002200
CR002300
CR002400
CR002500
CR002600
CR003000
CR003100
CR003200
CR003300
CR003500
CR003600
CR003700
DT001000
DT001100
DT002000
DT002100
DT002200
DT002300
DT101000
DT101100
DT101200
DT101300
DT101400
DT101500
DT101600
DT101700
DT101800
DT101900
DT102000
DT102100
DT102200
DT102300
DT102400
DT102500
DT102600
DT102700
DT102800
DT102900
DT201000
DT202000
EL001000
EL002000
CR004000
CR004200

ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS
CROSSZ - EFN SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 2

YCROSS = 0.	CR004400	
IT1 = ISURFC(1,11)	CR004600	
R1 = SURFC(11,11)	CR004800	
B1 = SURFC(12,11)	CR005000	
IT2 = ISURFC(1,12)	CR005200	
R2 = SURFC(11,12)	CR005400	
B2 = SURFC(12,12)	CR005600	
IF(I1 .EQ. I2) GO TO 32000	CR005800	
DEL1 = VTXCRD(I1)	CR006000	
DEL2 = VTXCRD(I2)	CR006200	
S = DEL2 - DEL1	CR006400	
Z2MINP = Z2MIN + S	CR006600	
Z2MAXP = Z2MAX + S	CR006800	
ZA = 0.	CR007000	
ZB = 0.	CR007200	
	CR007400	
C		
C		
CHECK FOR INTERSECTION	CR100800	
IF(Z1MIN .GT. Z2MAXP) GO TO 2700	CR100800	
IF(Z1MAX .LT. Z2MINP) GO TO 2700	CR100700	
C		
C		
INTERSECTION EXISTS	CR100800	
ZA = Z2MINP	CR100900	
ZB = Z2MAXP	CR101000	
IF(Z1MIN .GT. Z2MINP) ZA = Z1MIN	CR101100	
IF(Z1MAX .LT. Z2MAXP) ZB = Z1MAX	CR101200	
IF(ZA .NE. ZB) GO TO 3500	CR101400	
IF(ZA .NE. 0.) GO TO 1500	CR101600	
IF(IT1 .EQ. 3) GO TO 1000	CR101800	
IF(R1 .NE. 0.) GO TO 2500	CR102000	
1000 YCROSS = SQRT((B2-1.) * S * S - 2. * R2 * S)	CR102200	38
GO TO 32000	CR102400	
1500 IF(ZA .NE. 3) GO TO 2500	CR103000	
IF(IT2 .EQ. 3) GO TO 2000	CR103200	
IF(R2 .NE. 0.) GO TO 2500	CR103400	
2000 ZCROSS = S	CR104000	
2200 YCROSS = SQRT((B1-1.) * ZCROSS * ZCROSS + 2. * R1 * ZCROSS)	CR104200	51
GO TO 32000	CR104400	
2500 Y1 = SQRT((B1-1.) * ZA * ZA + 2. * R1 * ZA)	CR105000	54
Y2 = SQRT((B2-1.) * (ZA-S) * (ZA-S) + 2. * R2 * (ZA-S))	CR105200	55
IF(Y1 .EQ. Y2) GO TO 3000	CR105400	
C		
C		
INTERSECTION DOES NOT EXIST	CR105500	
2700 RETURN 1	CR105600	
C		
3000 ZCROSS = ZA	CR106000	
YCROSS = Y1	CR106200	
GO TO 32000	CR106400	
3500 ALPHA = B1 - B2	CR106600	
BETA = R1 - R2 + S * (B2-1.)	CR106800	
GAMMA = 2. * R2 * S - S * S * (B2-1.)	CR107000	
IF(ALPHA .NE. 0.) GO TO 4000	CR107200	
Z1 = - GAMMA / (2. * BETA)	CR107400	
3800 IF(Z1 .LT. ZA .OR. Z1 .GT. ZB) GO TO 2700	CR107600	
3900 ZCROSS = Z1	CR107800	
GO TO 2200	CR108000	
C		

ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS
 CROSSZ - EFM SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 3

4000 DELTA = BETA * BETA - ALPHA * GAMMA	CR108000
IF (DELTA .LT. 0.) GO TO 2700	CR108400
IF (DELTA .GT. 0.) GO TO 4800	CR108800
Z1 = - BETA / ALPHA	CR109200
GO TO 3800	CR109600
4800 Z1 = - BETA / ALPHA + SRT(DELTA) / ALPHA	CR201000 80
Z2 = - BETA / ALPHA - SRT(DELTA) / ALPHA	CR201200 81
IF (Z1 .GE. ZA .AND. Z1 .LE. ZB) GO TO 5000	CR201400
IF (Z2 .LT. ZA .OR. Z2 .GT. ZB) GO TO 2700	CR201600
ZCROSS = Z2	CR201800
GO TO 2200	CR202000
C	
5000 IF (Z2 .LT. ZA .OR. Z2 .GT. ZB) GO TO 3900	CR202200
ZCROSS = AMIN(Z1, Z2)	CR203000
IF (R1 .LT. 0.) ZCROSS = AMAX(Z1, Z2)	CR203200
GO TO 2200	CR203400
C	
C	END OF CROSS SUBROUTINE
3800 RETURN	CR204000
END	CR205000
	CR206000

ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS
 CROSS2 STORAGE MAP

02/16/68

PAGE 4

SUBROUTINE CROSS
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENG" #	08668	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00019	R
LMODE	00027	I	PUNCID	00032	R	DATA	00039	R
NCLRS	00033	I	NRAYS	00034	I	NSLCS	00039	I
NSUBP	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSPLN	00041	I	NIPLN	00042	I	INODE	00043	I
PLAS	00044	I	NOBJH	00048	I	NSURF	00046	I
FLNSH	00047	I	DELY	00050	R	FNAMB	00051	R
HEXPP	00052	R	WFLGH	00053	R	ZETA	00054	R
DLPLN	00055	R	DEXPP	00056	R	WEYPP	00057	R
DELD	00058	R	ONGA2	00051	R	ONGA1	00052	R
ONGAP	00059	R	EPHAD	00054	R	PSCAL	00055	R
ODIST	00060	R	SPFEA	00057	R	DUMIN	00070	R
CYSM	00071	R	HO	00072	R	DELH	00073	R
OTAM	00074	R	WDIR	00075	R	WDIR	00076	R
NOBJH	00117	R	NSGN	00100	I	NSGV	00104	I
EINH	00144	R	WLRH	00126	R	WINHT	00139	R
LATTC	00170	R	CIMPL	00153	R	WCLRS	00162	R
ISURF	00320	I	ILATTC	00170	I	SURFC	00320	I
SUBST	05224	R	DESGN	04240	R	IDESGN	04240	I
BNDS	06210	I	ISUBST	05224	I	BOUNDS	06210	R
			NCOND	06665	I			

COMMON BLOCK			ELHAIR	ORIGIN	08667	LENGT	00768	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NELMT	00000	I	HAIRS	00001	I	ELMNTX	00002	I
AIRLTH	00312	I	VTXCRD	00822	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IT1	07655	I	R1	07656	R	B1	07657	R
IT2	07660	I	R2	07661	R	B2	07662	R
DEL1	07663	R	DEL2	07664	R	S	07665	R
ZMINP	07666	R	Z2MAXP	07667	R	Z1	07670	R
Z1	07671	R	Y1	07672	R	Y2	07673	R
ALPHA	07674	R	BETA	07675	R	GAMMA	07676	R
Z1	07677	R	DELTA	07700	R	Z2	07701	R

ENTRY POINTS

CROSS SECTION 7

SUBROUTINES CALLED

SORT SECTION 8

SYSLCC SECTION 9

.FRET. SECTION 10

ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS
 CROSSZ STORAGE MAP

02/16/68

PAGE 9

			EPM IFM CORRESPONDENCE					
EPM	IFM	LOCATION	EPM	IFM	LOCATION	EPM	IFM	LOCATION
32000	89A	10430	2700	89A	10203	3500	82A	10213
1900	40A	10084	1000	37A	10040	2900	83A	10129
2000	49A	10100	2200	80A	10102	3000	80A	10208
4000	71A	10273	3800	88A	10257	3900	89A	10270
4800	79A	10322	9000	89A	10377			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 10226.

TR-67-700-10-2
Page III-2-33
Program Listing
Link 2

INF.N010001.44800-0.97397.AM IC ISJOB L10M1 7004 A 02/16/68 PAGE 6

00
BTITLE ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEMENT
BTPTC ENDPTZ M04.KR7 LINK 2 (ENDPTS) EP001000

ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME
ENDPTZ - EPN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 1

```

SUBROUTINE ENDPTS( DZMIN, ARCS, NARCS )
EP001800
C
C *** INPUTS ***
EP001800
C DZMIN MINIMUM ALLOWABLE Z SURFACE SEPARATION.
EP001700
C ELMNTR MATRIX OF EDITED SURFACE ELEMENTS. (ELMAIR COMMON) EP001800
C VTXCRD VECTOR OF VERTEX COORDINATES OF THE SURFACES. (ELMAIR COMMON) EP001900
C ZMAT MATRIX OF RANGE OF Z DEFINITION OF EACH SURFACE(COORNG COMMON) EP002000
C YMAT MATRIX OF RANGE OF Y DEFINITION OF EACH SURFACE(COORNG COMMON) EP002100
C CMAT MATRIX OF RANGE CODES FOR EACH SURFACE. (COORNG COMMON) EP002200
C
C *** OUTPUTS ***
EP002400
C ARCS 3-D ARRAY CONTAINING THE Z AND Y COORDINATES OF THE SEGMENTS
EP002500
C DEFINING EACH ELEMENT FOR EVERY ELEMENT OF THE LENS SYSTEM.
EP002600
C NARCS VECTOR WHICH CONTAINS THE NUMBER OF POINTS NEEDED TO DEFINE
EP002700
C THE SEGMENTS FOR EACH ELEMENT. BETWEEN 2 AND 3 POINTS ARE
EP002800
C NECESSARY FOR EACH ELEMENT.
EP002900
C
C *** DESCRIPTION ***
EP003000
C THIS SUBROUTINE GENERATES THE ARCS ARRAY WHICH CONSISTS OF
EP003100
C THE Z AND Y COORDINATES OF THE SEGMENTS WHICH DEFINE EACH
EP003200
C ELEMENT. THE SUBSCRIPTS OF ARCS ARE IN ORDER -
EP003300
C 1) THE Z COORDINATE FOLLOWED BY THE Y COORDINATE.
EP003400
C 2) THE SEGMENT DESIGNATION
EP003500
C 3) THE ELEMENT DESIGNATION
EP003600
C THE COORDINATES ARE REFERENCED FROM THE VERTEX OF THE FIRST
EP003700
C SURFACE OF EACH ELEMENT. THE NARCS VECTOR CONTAINS THE NUMBER
EP003800
C OF POINTS DEFINING EVERY ELEMENT. THUS THE NUMBER OF SEGMENTS
EP003900
C IN THE ITH. ELEMENT IS NARCS(I)-1.
EP004000
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID,
DT001000
1 DATA(3483)
DT001100
DIMENSION WOBJH(7), WCLR(7), WIMHT(7), EIMHT(7), CIMPL(7),
DT002000
1 WCLR(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100),
DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26),
DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100)
DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NWAYS), (DATA(3),
DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJATL), (DATA(6),
DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9),
DT101200
3 INODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12),
DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15),
DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18),
DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21),
DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), ONGA2), (DATA(24),
DT101700
8 ONGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27),
DT101800
9 PSCAL), (DATA(28), ONGAF), (DATA(29), SPFEA), (DATA(30),
DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), HO), (DATA(33),
DT102000
B DELH), (DATA(34), SYSHX), (DATA(35), WDIR), (DATA(36),
DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42),
DT102200
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLR(1)),
DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81),
DT102400
F CIMPL(1)), (DATA(88), WCLR(1)),
DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1),
DT102600
H ISURFC(1,1)), (DATA(2102), DESGN(1,1), IDESGN(1,1)),
DT102700
I (DATA(2602), SUBST(1), ISUBST(1)), (DATA(3102), BOUNDS(1,1),
DT102800
J IBNDS(1,1)), (DATA(3483), NCOND)
DT102900

```


ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME
ENDPTZ - EFN SOURCE STATEMENT - IPN(S) -

02/18/68

PAGE 2

```

INTEGER CONTRL, AFLAG, DATE          DT201000
RTAL LATTG                            DT202000
C
COMMON/ELHAIR/NELMT,NAIRS,ELMNTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
INTEGER ELMNTX, AIRMTX                EL002000
C
COMMON / CODRNG / ZHAT(2,100), YHAT(2,100), CHAT(2,100)          CD001000
INTEGER CHAT                          CD002000
C
DIMENSION ARCS(2,3,100), NARCS(100)  EP004000
C
C          START OF ENDPTS SUBROUTINE  EP004100
C
DO 2000 K = 1, 100                    EP004200
DO 1000 I = 1, 2                       EP004400
DO 1000 J = 1, 3                       EP004600
1000 ARCS(I,J,K) = 0.                   EP004800
2000 NARCS(K) = 0                       EP005000
IF ( NELMT .GT. 1 ) GO TO 7000         EP005200
IF ( ELMNTX(1,1) .NE. ELMNTX(2,1) ) GO TO 3000 EP005400
C
C          MIRROR                      EP006210
ARCS(1,2,1) = ZHAT(1,1)               EP006400
ARCS(2,2,1) = YHAT(1,1)               EP006600
NARCS(1) = 2                           EP006800
GO TO 32000                             EP007000
3000 I1 = ELMNTX(1,1)                  EP007200
I2 = ELMNTX(2,1)                       EP007400
S = VTXCRD(I2) - VTXCRD(I1)            EP007600
IF ( CHAT(1,1) .NE. 3 ) GO TO 4000     EP007800
ARCS(1,2,1) = ZHAT(1,1)               EP008000
ARCS(2,2,1) = YHAT(1,1)               EP008200
ARCS(1,3,1) = S                       EP008400
ARCS(2,3,1) = 0.                      EP008600
NARCS(1) = 3                           EP008800
GO TO 32000                             EP009000
4000 IF ( YHAT(1,1) .NE. YHAT(2,1) ) GO TO 5000 EP101000
ARCS(1,2,1) = ZHAT(1,1)               EP101200
ARCS(2,2,1) = YHAT(1,1)               EP101400
ARCS(1,3,1) = ZHAT(2,1) + S           EP101600
ARCS(2,3,1) = YHAT(1,1)               EP101800
ARCS(1,4,1) = S                       EP102000
NARCS(1) = 4                           EP102200
GO TO 32000                             EP102400
5000 IF ( YHAT(1,1) .GE. YHAT(2,1) ) GO TO 5500 EP103000
C
C          LIMIT CURVE 2                EP103200
CALL ZYCUT( 1, YHAT(1,1), I2, BETA, 36000, 1, 2 ) EP103400
ARCS(1,2,1) = ZHAT(1,1)               EP103600
ARCS(2,2,1) = YHAT(1,1)               EP103800
ARCS(1,3,1) = BETA + S                 EP104000
ARCS(2,3,1) = YHAT(1,1)               EP104200
ARCS(1,4,1) = S                       EP104400
NARCS(1) = 4                           EP104600
GO TO 32000                             EP104800
C

```

ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME
ENDPTZ - EFM SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 3

C	LIMIT CURVE 1	EP108000	
	5500 CALL ZYCUT(1, YHAT(2,1), I1, BETA, 8000, 1, 1)	EP108200	48
	ARCS(1,2,1) = BETA	EP108400	
	ARCS(2,2,1) = YHAT(2,1)	EP108600	
	ARCS(1,3,1) = ZHAT(2,1) + 8	EP108800	
	ARCS(2,3,1) = YHAT(2,1)	EP109000	
	ARCS(1,4,1) = 8	EP109200	
	NARCS(1) = 4	EP109400	
	GO TO 32000	EP109600	
C			
	6000 WRITE (6, 32010)	EP107000	49
	32010 FORMAT(1ND, 2PARAMETERED ERROR USING ZYCUT)	EP107200	
	CALL 600P	EP107400	50
	7000 DO 18000 I = 1, NELMT	EP107600	
	I1 = ELNMTX(1,I)	EP107800	
	I2 = ELNMTX(2,I)	EP108000	
	I3 = ELNMTX(1,I+1)	EP108200	
	I4 = ELNMTX(2,I+1)	EP108400	
	IF(I .EQ. NELMT .OR. I2 .EQ. I3) GO TO 14000	EP108600	
	Z1M = ANIN1(0., ZHAT(2,I))	EP108800	
	Z1X = ANAX1(0., ZHAT(2,I))	EP109000	
	Z2M = ANIN1(0., ZHAT(1,I+1))	EP109200	
	Z2X = ANAX1(0., ZHAT(1,I+1))	EP109400	
C			
C	CHECK AIR SPACE CONSTRAINT	EP201000	
	CALL CLOSS(I2, Z1M, Z1X, I3, Z2M, Z2X, DZMIN, Z1C, Y1C, Z2C,	EP201200	
	1 814000)	EP201400	70
	I2C = CHAT(2,I)	EP201600	
	I3C = CHAT(1,I+1)	EP201800	
	I60 = 4	EP202000	
	IF(I2C .NE. 3) GO TO 8000	EP202200	
	I60 = 1	EP202400	
	IF(I3C .EQ. 3) I60 = 3	EP202600	
	GO TO 8400	EP202800	
	8000 IF(I3C .EQ. 3) I60 = 2	EP203000	
	8400 GO TO (8500, 8600, 10400, 11500), I60	EP203200	
	8500 ZHAT(1,I+1) = Z2C	EP203400	
	YHAT(1,I+1) = Y1C	EP203600	
	S = VTXCRD(I4) - VTXCRD(I3)	EP203800	
	IF(I3 .EQ. I4) GO TO 14000	EP204000	
	IF(ISURFC(1,I2) .EQ. 3) GO TO 8800	EP204200	
	IF(SURFC(11,I2) .GT. 0.) GO TO 9000	EP204400	
	8800 CHAT(1,I+1) = 4	EP204600	
	CALL ZYCUT(2, Z2C-S, I4, BETA, 814000 I+1, 2)	EP204800	
	8900 ZHAT(2,I+1) = Z2C - S	EP205000	107
	YHAT(2,I+1) = BETA	EP205200	
	CHAT(2,I+1) = 1	EP205400	
	GO TO 14000	EP205600	
	9000 CHAT(1,I+1) = 5	EP205800	
	CALL ZYCUT(1, Y1C, I4, BETA, 814000, I+1, 2)	EP206000	116
	9200 ZHAT(2,I+1) = BETA	EP206200	
	YHAT(2,I+1) = Y1C	EP206400	
	CHAT(2,I+1) = 1	EP206600	
	GO TO 14000	EP206800	
	9300 ZHAT(2,I) = Z1C	EP207000	
	YHAT(2,I) = Y1C	EP207200	
		EP207400	
		EP207600	

ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEMENT
ENDPTZ - EFM SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 4

8 = VTXCRD(I2) - VTXCRD(I1)	EP207800	
IF (I1 .EQ. I2) GO TO 14000	EP208000	
IF (ISURFC(1,I3) .EQ. 3) GO TO 9800	EP208200	
IF (SURFC(11,I3) .LT. 0.) GO TO 10000	EP301000	
9800 CHAT(2,I) = 4	EP301200	
CALL ZYCUT(2, ZIC+8, I1, BETA, \$14000, I, 1)	EP301400	141
ZHAT(1,I) = ZIC + 8	EP301600	
YHAT(1,I) = BETA	EP301800	
GO TO 14000	EP302000	
10000 CHAT(2,I) = 8	EP302200	
CALL ZYCUT(1, YIC, I1, BETA, \$14000, I, 1)	EP302400	149
ZHAT(1,I) = BETA	EP302600	
YHAT(1,I) = YIC	EP302800	
GO TO 14000	EP303000	
10400 ZHAT(1,I+1) = ZIC	EP303200	
YHAT(1,I+1) = YIC	EP303400	
8 = VTXCRD(I4) - VTXCRD(I3)	EP303600	
IF (ISURFC(1,I2) .EQ. 3) GO TO 10800	EP303800	
IF (SURFC(11,I2) .GT. 0.) GO TO 11000	EP304000	
10800 JCD = 48	EP304200	
CHAT(1,I+1) = 4	EP304400	
CALL ZYCUT(2, ZIC-8, I4, BETA, \$10800, I+1, 2)	EP304600	171
GO TO 8900	EP304800	
C		
10800 WRITE (6, 32020) JCD	EP305000	174
32020 FORMAT(1HD, 10HEXERRR CODE, I3, 12H USING ZYCUT)	EP305200	
CALL GOCP	EP305400	175
11000 JCD = 52	EP305600	
CHAT(1,I+1) = 8	EP305800	
CALL ZYCUT(1, YIC, I4, BETA, \$10800, I+1, 2)	EP306000	179
GO TO 9200	EP306200	
11500 IF (ISURFC(1,I2) .EQ. 3) GO TO 11800	EP306400	
IF (SURFC(11,I2) .GT. 0.) GO TO 12000	EP306600	
11800 ZHAT(1,I+1) = ZIC	EP306800	
YHAT(1,I+1) = YIC	EP307000	
8 = VTXCRD(I4) - VTXCRD(I3)	EP307200	
IF (I3 .EQ. I4) GO TO 14000	EP307400	
GO TO 8900	EP307600	
12000 IF (ISURFC(1,I3) .EQ. 3) GO TO 12200	EP307800	
IF (SURFC(11,I3) .LT. 0.) GO TO 12500	EP308000	
12200 ZHAT(2,I) = ZIC	EP308200	
YHAT(2,I) = YIC	EP401000	
8 = VTXCRD(I2) - VTXCRD(I1)	EP401200	
IF (I1 .EQ. I2) GO TO 14000	EP401400	
GO TO 9800	EP401600	
12500 ZHAT(2,I) = ZIC	EP401800	
YHAT(2,I) = YIC	EP402000	
ZHAT(1,I+1) = ZIC	EP402200	
YHAT(1,I+1) = YIC	EP402400	
CHAT(2,I) = 4	EP402600	
CHAT(1,I+1) = 4	EP402800	
IF (I1 .EQ. I2) GO TO 13000	EP403000	
8 = VTXCRD(I2) - VTXCRD(I1)	EP403200	
CALL ZYCUT(2, ZIC+8, I1, BETA, \$13000, I, 1)	EP403400	
ZHAT(1,I) = ZIC + 8	EP403600	
YHAT(1,I) = BETA	EP403800	
	EP404000	228
	EP404200	
	EP404400	

ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME
 ENDP12 - EFN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 8

13000 IF (I3 .EQ. I4) GO TO 14000	EP405000	
S = VTXCRD(I4) - VTXCRD(I3)	EP405200	
CALL ZYCUT(2, Z2C-S, I4, BETA, \$14000, I+1, 2)	EP405400	230
ZHA (2,I+1) = Z2C - S	EP405600	
YHA (2,I+1) = BETA	EP405800	
CHAT (1,I+1) = 1	EP406000	
C		
14000 S = VTXCRD(I2) - VTXCRD(I1)	EP406200	
IF (I1 .NE. I2) GO TO 14400	EP406400	
ARCS(1,2,I) = ZHAT(1,I)	EP406600	
ARCS(2,2,I) = YHAT(1,I)	EP406800	
NARCS(I) = 2	EP407000	
GO TO 18000	EP407200	
14400 IF (CHAT(1,I) .NE. 3) GO TO 15000	EP407400	
ARCS(1,2,I) = ZHAT(1,I)	EP407600	
ARCS(2,2,I) = YHAT(1,I)	EP407800	
ARCS(1,3,I) = S	EP408000	
NARCS(I) = 3	EP408200	
GO TO 18000	EP408400	
15000 Z1 = ZHAT(1,I)	EP501000	
Y1 = YHAT(1,I)	EP501200	
Z2 = ZHAT(2,I) + S	EP501400	
Y2 = YHAT(2,I)	EP501600	
IF (ABS(Z1-Z2) .GE. 1.E-5) GO TO 15500	EP501800	
ARCS(1,2,I) = Z1	EP502000	
ARCS(2,2,I) = Y1	EP502200	
ARCS(1,3,I) = Z1	EP502400	
ARCS(2,3,I) = Y2	EP502600	
ARCS(1,4,I) = S	EP502800	
NARCS(I) = 4	EP503000	
GO TO 18000	EP503200	
15500 IF (ABS(Y1-Y2) .GE. 1.E-5) GO TO 16000	EP503400	
ARCS(1,2,I) = Z1	EP503600	
ARCS(2,2,I) = Y1	EP503800	
ARCS(1,3,I) = Z2	EP504000	
ARCS(2,3,I) = Y1	EP504200	
ARCS(1,4,I) = S	EP504400	
NARCS(I) = 4	EP504600	
GO TO 18000	EP504800	
16000 IF (Y1 .GT. Y2) GO TO 17000	EP505000	
IF (CHAT(1,I) .EQ. 5) GO TO 16500	EP505200	
ARCS(1,2,I) = Z1	EP505400	
ARCS(2,2,I) = Y1	EP505600	
ARCS(1,3,I) = Z1	EP505800	
ARCS(2,3,I) = Y2	EP506000	
ARCS(1,4,I) = Z2	EP506200	
ARCS(2,4,I) = Y2	EP506400	
ARCS(1,5,I) = S	EP506600	
NARCS(I) = 5	EP506800	
GO TO 18000	EP507000	
16500 JCD = 92	EP507200	
CALL ZYCUT(1, Y1, I2, BETA, \$10000, I, 2)	EP507400	
ARCS(1,2,I) = Z1	EP507600	
ARCS(2,2,I) = Y1	EP507800	
ARCS(1,3,I) = BETA + S	EP508000	
ARCS(2,3,I) = Y1	EP508200	
	EP508400	
	EP508600	310
	EP508800	
	EP509000	
	EP509100	
	EP509200	

ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME
ENDPTZ - EPN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 6

```
ARCS(1,4,I) = 8
NARCS(I) = 4
GO TO 18000
17000 IF ( CHAT(2,I) .EQ. 5 ) GO TO 17500
ARCS(1,2,I) = Z1
ARCS(2,2,I) = Y1
ARCS(1,3,I) = Z2
ARCS(2,3,I) = Y1
ARCS(1,4,I) = Z2
ARCS(2,4,I) = Y2
ARCS(1,5,I) = 8
NARCS(I) = 5
GO TO 18000
17500 JCD = 98
CALL ZYCUT( 1, Y2, I1, BETA, $10800, I, 1 )
ARCS(1,2,I) = BETA
ARCS(2,2,I) = Y2
ARCS(1,3,I) = Z2
ARCS(2,3,I) = Y2
ARCS(1,4,I) = 8
NARCS(I) = 4
18000 CONTINUE
C
C          END OF ENDPTS SUBROUTINE
32700 RETURN
END
```

EP509400
EP509600
EP509800
EP601000
EP601200
EP601400
EP601600
EP601800
EP602000
EP602200
EP602400
EP602600
EP602800
EP603000
EP603200 339
EP603400
EP603600
EP603800
EP604000
EP604200
EP604400
EP605000

EP606000
EP606500
EP607000

ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME
 ENOPTZ STORAGE MAP

02/16/68

PAGE 7

SUBROUTINE ENOPTS
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00039	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00049	I	NSURF	00046	I
AFLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXPP	00055	R	WEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMG A2	00061	R	OMG A1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMG AF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	00071	R	HO	00072	R	DELH	00073	R
SYSHK	00074	R	WXDIR	00075	R	WYDIR	00076	R
MDTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WOB JH	00117	R	WCLRH	00126	R	WINHT	00135	R
EINHT	00144	R	CIMPL	00193	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESIGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I			

COMMON BLOCK			ELMAIR	ORIGIN	08667	LENGTH	00766	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NELMT	00000	I	NAIRS	00001	I	ELMHTX	00002	I
AIRMTX	00312	I	VTXCRD	00822	R			

COMMON BLOCK			CODRNG	ORIGIN	07655	LENGTH	01130	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ZHAT	00000	R	YHAT	00310	R	CHAT	00120	I

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
K	11008	I	I	11008	I	I1	11007	I
IB	11010	I	S	11011	R	BETA	11012	R
IS	11013	I	I4	11014	I	Z1H	11015	R
Z1X	11016	R	Z2H	11017	R	Z2X	11020	R
Z1C	11021	R	Y1C	11022	R	Z2C	11023	R
IBC	11024	I	I3C	11025	I	ICO	11026	I
JCD	11027	I	Z1	11030	R	Y1	11031	R
Z2	11032	R	Y2	11033	R			

ENTRY POINTS

ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME
 ENDPTZ STORAGE MAP

02/16/68

PAGE 8

SUBROUTINES CALLED

ZYCUT SECTION 10
 CLOS8 SECTION 13
 .FFIL. SECTION 16

.FWRD. SECTION 11
 .FXEM. SECTION 14
 .FCNV. SECTION 17

GOOF SECTION 12
 .UNDB. SECTION 13
 SYSLOC SECTION 18

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
200	13A	11125	1000	9A	11117	7000	91A	11317
3001	24A	11150	32000	347A	12566	4000	33A	11177
5000	37A	11221	5500	44A	11256	6000	49A	11308
32010	FORMAT	11082	18000	344A	12562	14000	244A	12273
8000	85A	11444	8400	88A	11451	8500	89A	11462
9500	123A	11608	10400	159A	11720	11500	182A	12042
8800	105A	11526	9000	114A	11960	8900	109A	11550
9200	118A	11577	9800	139A	11652	10000	147A	11677
10800	168A	11760	11000	178A	12020	10800	174A	12008
32020	FORMAT	11071	11800	190A	12085	12000	199A	12107
12200	207A	12132	12500	216A	12154	13000	233A	12227
14400	256A	12317	15000	267A	12334	15500	262A	12373
16000	292A	12421	17000	320A	12505	16500	308A	12453
17500	333A	12532						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12760.

TR-67-700-10-2
Page III-2-42
Program Listing
Link 2

IMP.N010001.44800-0.77367.AM IC

ISJOB LINK1 7094 A 02/18/68

PAGE 0

00
BTITLE ROUTINE TO CHECK FOR INTERSECTIONS
SIBPTC E1R6CZ M04.X87 LINK 2 (E1R6CD)

E1001000

ROUTINE TO CHECK FOR INTERSECTIONS
EIR6CZ - EFN SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 1

```

SUBROUTINE EIR6CD                                E1001200
C
C *** INPUTS ***                                E1001300
C ELMNT4 MATRIX OF EDITED SURFACE ELEMENTS.      (ELMAIR COMMON) E1001600
C VTXCRD VECTOR OF VERTEX COORDINATES OF SURFACES. (ELMAIR COMMON) E1001700
C ZHAT MATRIX OF RANGE OF Z FOR EACH SURFACE.    (CODRNG COMMON) E1001800
C YHAT MATRIX OF RANGE OF Y FOR EACH SURFACE.    (CODRNG COMMON) E1001900
C
C *** OUTPUTS ***                                E1002000
C ZHAT EDITED MATRIX OF RANGE OF Z DEFINITION.  (CODRNG COMMON) E1002100
C YHAT EDITED MATRIX OF RANGE OF Y DEFINITION.  (CODRNG COMMON) E1002200
C CHAT EDITED CODE MATRIX, = 3 IF TWO ADJOINING (CODRNG COMMON) E1002300
C SURFACES INTERSECT.                            E1002400
C
C *** DESCRIPTION ***                             E1002500
C THIS SUBROUTINE EDITS THE Z AND Y RANGE MATRICES AND THE CODE E1002600
C MATRIX BASED ON THE SURFACE INTERSECTION CONSTRAINTS. POINTS E1002700
C OF INTERSECTION OF ADJOINING SURFACES REPLACE PREVIOUSLY E1002800
C DETERMINED RANGES.                             E1002900
C
COMMON/ELMAIR/NEIHT,NAIR8,ELMNTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
INTEGER ELMNTX, AIRMTX                                EL002000
C
COMMON / CODRNG / ZHAT(2,100), YHAT(2,100), CHAT(2,100)    CD001000
INTEGER CHAT                                          CD002000
C
C START OF EIR6CD SUBROUTINE                        E1003000
DO 2000 I = 1, NEIHT                                E1003500
  I1 = ELMNTX(1,I)                                  E1004000
  I2 = ELMNTX(2,I)                                  E1004200
  IF ( I1 .EQ. I2 ) GO TO 2000                       E1004400
  Z1MIN = AMIN1( 0., ZHAT(1,I) )                    E1004600
  Z1MAX = AMAX1( 0., ZHAT(1,I) )                    E1004800
  Z2MIN = AMIN1( 0., ZHAT(2,I) )                    E1005000
  Z2MAX = AMAX1( 0., ZHAT(2,I) )                    E1005200
C
C CHECK FOR INTERSECTION                            E1006000
CALL CROSS( I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX, ZI, YI, $2000 ) E1006200 16
ZIPRM = ZI - ( VTXCRD(I2) - VTXCRD(I1) )            E1006400
IF ( ZI .LT. Z1MIN .OR. ZI .GT. Z1MAX ) GO TO 4000 E1006600
IF ( ZIPRM .LT. Z2MIN .OR. ZIPRM .GT. Z2MAX ) GO TO 4000 E1006800
IF ( YI .LT. 0. .OR. YI .GT. AMIN1( YHAT(1,I), YHAT(2,I) ) ) E1007000
I GO TO 4000                                          E1007200
C
C LENS ELEMENT IS CLOSED                            E1007300
YHAT(1,I) = YI                                       E1007400
YHAT(2,I) = YI                                       E1007600
CHAT(1,I) = 3                                         E1007800
CHAT(2,I) = 3                                         E1008000
ZHAT(1,I) = ZI                                       E1008200
ZHAT(2,I) = ZIPRM                                     E1008400
2000 CONTINUE                                         E1008600
GO TO 6000                                           E1201000
C
C ILLEGAL INTERSECTION POINT DETERMINED            E1201400

```

ROUTINE TO CHECK FOR INTERSECTIONS
EIRGCD - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 2

4000 WRITE (6, 32100) ZI, YI, II, IS
32100 FORMAT (1HD, 27HTHE POINT OF INTERSECTION (, E14.7, 1H, E14.7,
1 10H) SURFACES, IS, 4H AND, IS, 13H IS NOT VALID)
CALL GOOF

E1201500 42
E1202000
E1202200
E1204000

C
C END OF EIRGCD SUBROUTINE
6000 RETURN
END

E1204000 43
E1204500
E1205000

ROUTINE TO CHECK FOR INTERSECTIONS
 EIRSCZ

02/16/68

PAGE 3

STORAGE MAP

SUBROUTINE EIRSCZ
 COMMON VARIABLES

COMMON BLOCK			ELM1R	ORIGIN	00001	LENGTH	00000		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
NELMT	00000	I	NAIRS	00001	I	ELM1R	00000	I	
AIRMTX	00312	I	VTKCRD	00002	R				
COMMON BLOCK			COORDS	ORIGIN	00007	LENGTH	01130		
CHAT	00000	R	YHAT	00010	R	CHAT	00000	I	

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
I	02117	I	I1	02120	I	I2	02121	I
Z1MIN	02122	R	Z1MAX	02123	R	Z2MIN	02124	R
Z2MAX	02125	R	Z1	02126	R	Y1	02127	R
Z1PRN	02130	R						

ENTRY POINTS

EIRSCD SECTION 7

SUBROUTINES CALLED

CROSS SECTION 8
 .UNDB. SECTION 11
 SYBLOC SECTION 14

.FWRD. SECTION 9
 .FFIL. SECTION 12

GOOP SECTION 10
 .FCNV. SECTION 13

EPN IFN CORRESPONDENCE

EPN	IFN	LOCATION	EPN	IFN	LOCATION	EPN	IFN	LOCATION
2C00	38A	02351	4000	42A	02358	6000	44A	02378
3E100	FORMAT	02143						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 02417.

TR-67-700-10-2
Page III-2-46
Program Listing
Link 2

INF.N010001.44000-D.77307.AM IC

ISJOB LINE: 7004 A 02/16/68

PAGE 4

00
BTITLE ROUTINE TO GENERATE BINARY POINT FILE
BTITLE PFILE2 N04.XR7 LINE 2 (PFILE)

PL001608

ROUTINE TO GENERATE BINARY POINT FILE
PFILEZ - EPN SOURCE STATEMENT - IFN(8) -

02/16/88

PAGE 1

```

SUBROUTINE PFILE( ARCS, NARCS, ZMIN, ZMAX )
C
C *** INPUTS ***
C NPTS NUMBER OF POINTS PER CURVED ARC
C ARCS 3D MATRIX WHERE ITH LAYER CONTAINS END POINTS OF THE SEVERAL
C ARCS WHICH COMPRISE ELEMENT I.
C NARCS VECTOR WHOSE ITH COMPONENT SPECIFIES THE NUMBER OF POINTS IN
C LAYER I OF MATRIX ARCS.
C YMAXX MAXIMUM Y ALLOWED (TUBE BOUNDARY).
C
C *** OUTPUTS ***
C ZMIN MINIMUM Z RELATIVE TO ENTRANCE PUPIL.
C ZMAX MAXIMUM Z RELATIVE TO ENTRANCE PUPIL.
C
C *** DESCRIPTION ***
C PFILE CREATES A BINARY FILE (LOGICAL 11) OF NELMT+4 RECORDS EACH ONE
C OF WHICH IS 601 WORDS LONG AND CONTAINING ...
C
C N,ZLIST,YLIST
C
C N=NO. OF POINTS
C ZLIST=VECTOR CONTAINING ZI I=1,2,...,N
C YLIST=VECTOR CONTAINING YI I=1,2,...,N
C
C THE POINTS (ZI,YI) I=1,2,...,N ARE CONSECUTIVE POINTS ALONG THE
C UPPER HALF OF A CURVE (YI,G.E.O.) DEFINED BY Y=F(Z). THIS CURVE WILL
C BE PLOTTED.
C
C RECORDS 1 THRU NELMT CONTAIN DATA FOR THE NELMT ELEMENTS.
C RECORD NELMT+1 CONTAINS DATA FOR THE IMAGE PLANE
C RECORD NELMT+2 CONTAINS DATA FOR THE ENTRANCE PUPIL
C RECORD NELMT+3 CONTAINS DATA FOR RAY 1 (H,TH)=(HMAX,0.) (X0,Y0)=(0,-1)
C RECORD NELMT+4 CONTAINS DATA FOR RAY 2 (H,TH)=(HMAX,0.) (X0,Y0)=(0,-1)
C
C COMMON / DATA / NCHTRL, CONTRL(10), TITLE(12), DATE(3), PUNCID,
1 DATA(3483)
DIMENSION WOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7),
1 WCLRS(6), NDSGN(4), NDSGV(11), LATTC(3,26), BOUNDS(3,100),
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATTC(3,26),
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100)
EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAY ), ( DATA(3),
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),
2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9),
3 IMODE ), ( DATA(10), NSPLN ), ( DATA(11), WOBJH ), ( DATA(12),
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15),
5 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18),
6 ZETA ), ( DATA(19), HEXPP ), ( DATA(20), DEXPP ), ( DATA(21),
7 WIXPP ), ( DATA(22), DLPLN ), ( DATA(23), OMC2 ), ( DATA(24),
8 OMC1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27),
9 PSCAL ), ( DATA(28), OMC4F ), ( DATA(29), SFEA ), ( DATA(30),
A DUMIN ), ( DATA(31), ODIST ), ( DATA(32), HO ), ( DATA(33),
B DELH ), ( DATA(34), SYSWX ), ( DATA(35), WXDIR ), ( DATA(36),
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42),

```

PL001800
PL002000
PL002100
PL002200
PL002400
PL002600
PL002800
PL003000
PL003200
PL003400
PL003500
PL003600
PL004000
PL004200
PL004400
PL004500
PL004600
PL005000
PL005200
PL005400
PL005600
PL005800
PL006000
PL006200
PL006400
PL006600
PL006800
PL007000
PL007200
PL007400
PL007600
PL007800
PL008000
PL008200
PL008400
PL008600
DT001000
DT001100
DT002000
DT002100
DT002200
DT002300
DT101000
DT101100
DT101200
DT101300
DT101400
DT101500
DT101600
DT101700
DT101800
DT101900
DT102000
DT102100
DT102200

ROUTINE TO GENERATE BINARY POINT FILE
PPFILEZ - EFM SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 3

GO TO 310	PL202800	
70 CALL ARPTS(I1,Z1,Y1,Z2,Y2,NPTS,ZLIST(1),YLIST(1))	PL202800	38
JCNT=NPTS	PL203000	
GO TO 310	PL203200	
C	PL203400	
C 2 ARCS ...	PL203600	
80 IF (Z1.NE.Z2) GO TO 110	PL203800	
JCNT=1	PL204000	
ZLIST(1)=Z1	PL204200	
YLIST(1)=Y1	PL204400	
GO TO 120	PL204600	
110 CALL ARPTS(I1,Z1,Y1,Z2,Y2,NPTS,ZLIST(1),YLIST(1))	PL204800	44
JCNT=NPTS-1	PL205000	
120 IF (Z2.NE.Z3) GO TO 140	PL205200	
ZLIST(JCNT+1)=Z2	PL205400	
YLIST(JCNT+1)=Y2	PL205600	
ZLIST(JCNT+2)=Z3	PL205800	
YLIST(JCNT+2)=Y3	PL206000	
JCNT=JCNT+2	PL206200	
GO TO 310	PL206400	
140 Z2=Z2-S	PL206600	
Z3=Z3-S	PL206800	
CALL ARPTS(I2,Z2,Y2,Z3,Y3,NPTS,ZLIST(JCNT+1),YLIST(JCNT+1))	PL207000	55
DO 190 J=1,NPTS	PL207200	
K=JCNT+J	PL207400	
190 ZLIST(K)=ZLIST(K)+S	PL207600	
JCNT=JCNT+NPTS	PL207800	
GO TO 310	PL208000	
C	PL301000	
C 3 ARCS ...	PL301200	
210 IF (Z1.NE.Z2) GO TO 240	PL301400	
JCNT=2	PL301600	
ZLIST(1)=Z1	PL301800	
YLIST(1)=Y1	PL302000	
ZLIST(2)=Z2	PL302200	
YLIST(2)=Y2	PL302400	
GO TO 250	PL302600	
240 CALL ARPTS(I1,Z1,Y1,Z2,Y2,NPTS,ZLIST(1),YLIST(1))	PL302800	72
JCNT=NPTS	PL303000	
250 Z2=Z3	PL303200	
Y2=Y3	PL303400	
Z3=Z4	PL303600	
Y3=Y4	PL303800	
GO TO 120	PL304000	
C	PL304200	
C 4 ARCS ...	PL305000	
260 IF (Z1.NE.Z2) GO TO 290	PL305200	
JCNT=2	PL305400	
ZLIST(1)=Z1	PL305600	
YLIST(1)=Y1	PL305800	
ZLIST(2)=Z2	PL306000	
YLIST(2)=Y2	PL306200	
GO TO 300	PL306400	
290 CALL ARPTS(I1,Z1,Y1,Z2,Y2,NPTS,ZLIST(1),YLIST(1))	PL306600	82
JCNT=NPTS	PL306800	
300 JCNT=JCNT+1	PL307000	

ROUTINE TO GENERATE BINARY POINT FILE
PFILEZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 4

```

ZLIST(JCNT)=Z3
YLIST(JCNT)=Y3
Z2=Z4
Y2=Y4
Z3=Z8
Y3=Y8
GO TO 120

C
C TRANSLATE DATA TO CENTER OF ENTRANCE PUPIL ...
C
310 S=VTXCRD(I1)
DO 390 J=1,JCNT
390 ZLIST(J)=ZLIST(J)+S
C
C WRITE DATA AS NEXT LOGICAL RECORD ON 11 ...
C
WRITE (11) JCNT,ZLIST,YLIST
C
C GET CURRENT MAX AND MIN ...
DO 400 J=1,JCNT
ZMIN=AMIN1(ZMIN,ZLIST(J))
400 ZMAX=AMAX1(ZMAX,ZLIST(J))
410 CONTINUE
C
C PUT IMAGE PLANE DATA ON 11 ...
C
IF (ELMNTX(2,NZLMT).NE.NSURF+1) GO TO 450
JCNT=0
DO 440 J=1,300
ZLIST(J)=0.
440 YLIST(J)=0.
GO TO 460
450 ZLIST(1)=VTXCRD(NSURF+1)
YLIST(1)=0.
ZLIST(2)=ZLIST(1)
YLIST(2)=YMAXX
ZMIN=AMIN1(ZMIN,ZLIST(1))
ZMAX=AMAX1(ZMAX,ZLIST(1))
JCNT=2
460 WRITE (11) JCNT,ZLIST,YLIST
C
C PUT ENTRANCE PUPIL DATA ON TAPE ...
C
JCNT=2
ZLIST(1)=0.
ZLIST(2)=0.
YLIST(1)=EPRAD
YLIST(2)=YMAXX
WRITE (11) JCNT,ZLIST,YLIST
C
C TRACE 2 SPECIAL RAYS ...
C
HMAX=HO+FLOAT(NOBJN-1)*DELH
THTR=0.
SNTHTR=0.
CSTHTR=1.

```

```

PL307200
PL307400
PL307600
PL307800
PL308000
PL308200
PL308400
PL401000
PL401200
PL401400
PL401600
PL401800
PL402000
PL402200
PL402400
PL402600
PL402800
PL403000
PL403200
PL403400
PL403600
PL403800
PL404000
PL404200
PL404400
PL404600
PL404800
PL405000
PL405200
PL405400
PL405600
PL405800
PL406000
PL406200
PL406400
PL406600
PL406800
PL407000
PL407200
PL407400
PL407600
PL407800
PL408000
PL408200
PL408400
PL408600
PL408800
PL409000
PL409200
PL501000
PL501200
PL501400
PL501600
PL501800
PL502000
PL502200

```

98

124

128

ROUTINE TO GENERATE BINARY POINT FILE
PPFILEZ - EPN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 3

	ASSIGN 680 TO IGO	PL502300	
	CALL RAYTR(0.,-1.,HMAX,1,XVCT,QVCT,SMISS,\$520,\$520,\$510,1,ZLIST,	PL502400	700000
	1 YLIST)	PL502600	132
510	GO TO 530	PL502800	
520	N1 = 0	PL503000	
	GO TO 678	PL503100	
530	N1=NSURF+1	PL503200	
540	DO 570 I=1,N1	PL503600	
	J=3+I	PL503800	
	YLIST(J-2)=ZLIST(J-2)	PL504000	
	YLIST(J-1)=ZLIST(J-1)	PL504200	
570	YLIST(J)=ZLIST(J)	PL504400	
	DO 600 I=1,N1	PL504600	
	J=3+I	PL504800	
	ZLIST(I)=YLIST(J)	PL505000	
	I1=N1+I	PL505200	
600	ZLIST(I1)=YLIST(J-1)	PL505400	
	DO 650 I=1,N1	PL505600	
	I1=N1+I	PL505800	
650	YLIST(I)=ZLIST(I1)	PL506000	
	DO 670 I = 2, N1	PL506150	
670	ZLIST(I) = ZLIST(I) + VTXCRD(I-1)	PL506100	
C			
C	COMPUTE RAY POSITION ON IMAGE SURFACE	PL506120	
	N1 = N1 + 1	PL506140	
	ZLIST(N1) = VTXCRD(NSURF+1)	PL506160	
	YLIST(N1) = (QVCT(2) / QVCT(3)) * (DELD - XVCT(3)) + XVCT(2)	PL506180	
678	WRITE (11) N1, ZLIST, YLIST	PL506200	184
	GO TO IGO, (680,710)	PL506400	
680	ASSIGN 710 TO IGO	PL506600	
	N1 = NSURF + 1	PL506700	
	CALL RAYTR(0.,1.,HMAX,1,XVCT,QVCT,SMISS,\$520,\$520,\$700,1,ZLIST,	PL506800	
	1 YLIST)	PL507000	191
700	GO TO 540	PL507200	
710	END FILE 11	PL509000	194
	REWIND 11	PL509200	195
	RETURN	PL509400	
	END	PL509600	

ROUTINE TO GENERATE BINARY POINT FILE
 PFILEZ

STORAGE MAP

02/16/68

PAGE 6

SUBROUTINE PFILE
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
NCNTRL	00000	I	CONTR	00001	I	TITLE	00013	R	
DATE	00027	I	PUNCI0	00032	R	DATA	00035	R	
LMODE	00033	I	NRAYS	00034	I	NSLCS	00039	I	
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I	
NSLBP	00041	I	NIPLN	00042	I	IMODE	00043	I	
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I	
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R	
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R	
HEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R	
DLPLN	00060	R	ONGA2	00061	R	ONGA1	00062	R	
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R	
ONGAP	00066	R	SPFEA	00067	R	DUMIN	00070	R	
ODIST	00071	R	NO	00072	R	DELH	00073	R	
YSMX	00074	R	WDIR	00075	R	WYDIR	00076	R	
NDTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I	
WOB JH	00117	R	WCLRH	00126	R	WIMHT	00135	R	
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R	
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R	
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I	
SUBST	05224	R	ISUBST	05224	I	BOUNDS	08210	R	
IBNDS	08210	I	NCOND	08665	I				
COMMON BLOCK			ELMAIR	ORIGIN	08667	LENGTH	00766		
NELMT	00000	I	NAIRS	00001	I	ELMHTX	00002	I	
AIRMTX	00312	I	VTXCRD	00822	R				
COMMON BLOCK			CODRNG	ORIGIN	07655	LENGTH	01130		
ZHAT	00000	R	YHAT	00310	R	CHAT	00620	I	
ZLIST	00000	R	YLIST	00494	R				
COMMON BLOCK			PLOTG	ORIGIN	11005	LENGTH	00003		
YMAXX	00000	R	DZMIN	00001	R	NPTS	00002	I	
COMMON BLOCK			AZOBJ	ORIGIN	11010	LENGTH	00003		
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R	
DIMENSIONED PROGRAM VARIABLES									
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
XVCT	11013	R	GVCT	11016	R				
UNDIMENSIONED PROGRAM VARIABLES									
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
I	11021	I	J	11022	I	IGO	11023	I	
Z1	11024	R	Y1	11025	R	Z2	11026	R	

ROUTINE TO GENERATE BINARY POINT FILE
 PFILE2

02/16/68

PAGE 7

ROUTINE TO GENERATE BINARY POINT FILE PFILE2			STORAGE MAP					
Y2	11027	R	Z3	11030	R	Y3	11031	R
Z4	11032	R	Y4	11033	R	Z3	11034	R
Y3	11038	R	I1	11036	I	I2	11037	I
S	11040	R	JCNT	11041	I	K	11042	I
HMAX	11043	R	SMISS	11044	R	N1	11048	I

ENTRY POINTS

PFILE SECTION IS

SUBROUTINES CALLED

.PRMT.	SECTION	14
.PBLO.	SECTION	17
.PKEM.	SECTION	20
.PBLT.	SECTION	23
E.2	SECTION	26
CC.1	SECTION	29
CC.4	SECTION	32

ARPT8	SECTION	18
RAYTR	SECTION	19
.UN11.	SECTION	21
.FBDT.	SECTION	24
E.3	SECTION	27
CC.2	SECTION	30
SYBLOC	SECTION	33
EPN	IFN	CORRESPONDENCE

.PWRB.	SECTION	16
.FEPT.	SECTION	19
.PWL.R.	SECTION	22
E.1	SECTION	25
E.4	SECTION	28
CC.3	SECTION	31

EPN	IFN	LOCATION	EPN	IFN	LOCATION	EPN	IFN	LOCATION
410	108A	11878	30	8A	11108	30	30A	11169
60	38A	11222	210	66A	11362	260	76A	11427
70	34A	11204	310	69A	11511	110	43A	11235
120	46A	11253	140	54A	11308	190	61A	11348
240	71A	11401	250	74A	11416	290	61A	11446
300	64A	11463	350	93A	11522	400	102A	11564
450	122A	11620	440	117A	11614	460	124A	11649
680	189A	12163	520	135A	11767	510	134A	11766
530	137A	11771	675	184A	12141	540	138A	11774
570	147A	12023	600	156A	12056	650	166A	12101
670	174A	12114	710	194A	12212	700	193A	12211

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12323.

TR-67-700-10-2
Page III-2-54
Program Listing
Link 2

IMP.ND1000; .44800-0.77387.AM IC

ISJOB LINK1 7096 A 02/16/68

PAGE 8

00
STITLE CONTROL ROUTINE OF PROFILE PLINK
SIBPTC PROFLZ M94.XR7 LINK 2 (PROFIL)

PF001000

CONTROL ROUTINE OF PROFILE PLOT LINE
PROFLZ - EFM SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 1

```

SUBROUTINE PROFIL                                PF001800
C
C *** INPUTS ***                                PF002000
C YMAXX MAXIMUM ALLOWABLE Y VALUE FROM OPTIC AXIS. (PLOT) PF002100
C DZMIN MINIMUM ALLOWABLE Z SURFACE SEPARATION. (PLOT) PF002200
C NPTS NUMBER OF POINTS TO GENERATE ON EACH SURFACE. (PLOT) PF002300
C
C *** OUTPUTS ***                                PF002400
C CROSS SECTION PLOT FRAME AND PLOT PARAMETERS ON FOLLOWING FRAME PF002500
C
C *** DESCRIPTION ***                             PF002600
C THIS SUBROUTINE CONTAINS THE ENTIRE LOGIC TO GENERATE THE PF002700
C CROSS SECTION PLOT OF THE LENS SYSTEM AND THE PLOT PARAMETER PF002800
C DATA FRAME. SEVERAL SUPPLEMENTARY SUBROUTINES PERFORM VARIOUS PF002900
C LOGICAL FUNCTIONS IN THIS PROCESS.             PF003000
C
COMMON/ELMAIR/RELMT,MAIRS,ELMNTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
INTEGER ELMNTX, AIRMTX                               EL002000
C
COMMON / CODRNS / ZHAT(2,100), YHAT(2,100), CHAT(2,100)          CD001000
INTEGER CHAT                                          CD002000
C
COMMON / PLOT / YMAXX, DZMIN, NPTS                      PC001000
C
COMMON / PRINT / LINE, PAGE                             PT001000
INTEGER PAGE                                          PT002000
C
DIMENSION ARCS(2,5,100), NARCS(100)                   PF004000
C
C GET ELEMENTS AND AIR SPACES ...                       PF005000
CALL CELAIR( 81000 )                                   PF005200
C
C GET RANGES FOR ALL ELEMENTS ...                       PF005400 2
CALL RANGE( YMAXX )                                   PF005600
C
C GET INTERSECTIONS ...                                 PF005800 3
CALL EIRGCD                                           PF005900
C
C GET ARC LIST EACH ELEMENT ...                         PF006000 7
CALL ENDPTS( DZMIN, ARCS, NARCS )                     PF006400
C
C SECTION TO PRINT OUT LENS SYSTEM                      PF151000 9
CALL PRCTL                                           PF151200 11
WRITE ( 6, 32010 )                                     PF151300 12
32010 FORMAT( 1H0, 5X, 33H CROSS SECTIONAL LENS SYSTEM PLOT // 6X, PF152000
1 85H Z REFERENCED FROM VERTEX OF SURFACE 1 FOR EACH ELEMENT, Y REF PF152200
ERENCED FROM OPTIC AXIS )                             PF152400
LINE = LINE + 4                                       PF152600
DO 6000 I = 1, NELMT                                  PF153000
WRITE ( 6, 32020 ) I, ELMNTX(1,I), ELMNTX(2,I)       PF153200 16
32020 FORMAT( 1H0, 5X, 7HELEMENT, 13, 5X, 13H( SURFACE 1 =, 13, 3X, PF153400
1 11HSURFACE 2 =, 13, 2H ) // 14X, 1HZ, 15X, 1HY )   PF153600
NP = NARCS(I)                                         PF154000
DO 5000 J = 1, NP                                     PF154200
WRITE( 6, 32030 ) ARCS(1,J,I), ARCS(2,J,I)           PF154400 22

```

CONTROL ROUTINE OF PROFILE PLOT LINK
PROFLZ - EFM SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 2

5000 CONTINUE	PF194800
32030 FORMAT(IN , SX, SE16.7)	PF194800
LINE = LINE + NP + 4	PF195000
IF(LINE .GT. 48) CALL PRCTL	PF195200 29
6000 CONTINUE	PF195400
C	
C GENERATE PLOT POINTS ...	PF201000
CALL PFILE(ARCS, MARCS, ZMIN, ZMAX)	PF201500
C	
C PLOT LENS SYSTEM ...	PF202000 34
CALL CPLOT(ZMIN, ZMAX)	PF202500
C	
10000 RETURN	PF203000 38
END	PF203400

CONTROL ROUTINE OF PROFILE PLOT LINE
 PROFLT

08/16/68

PAGE 3

STORAGE MAP

SUBROUTINE PROFIL
 COMMON VARIABLES

COMMON BLOCK			ELNAIR	ORIGIN	00001	LENGTH	00000	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NELMT	00000	I	NAIRB	00001	I	ELNTR	00000	I
AIRMTX	00310	I	VTRCD	00002	R			
COMMON BLOCK			COJRS	ORIGIN	00067	LENGTH	01130	
ZMAT	00000	R	YMAT	00310	R	CMAT	00000	I
COMMON BLOCK			PLOT	ORIGIN	00117	LENGTH	00000	
WAXX	00000	R	DZM	00001	R	NPTS	00000	I
COMMON BLOCK			PR	ORIGIN	00122	LENGTH	00000	
LINE	00000	I	PAGE	00001	I			

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ARCS	00124	R	NARCS	04074	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
I	04240	I	NP	04241	I	ZPTN	04242	R
ZMAX	04243	R						

ENTRY POINTS

PROFIL SECTION 11

SUBROUTINES CALLED

CELAIR	SECTION	12	RANGE	SECTION	13	E1R6CD	SECTION	14
ENDPTS	SECTION	15	PRCTL	SECTION	16	.FWRD.	SECTION	17
PPIL	SECTION	18	CPLOT	SECTION	19	.UNDB.	SECTION	20
.PPIL.	SECTION	21	.PCNV.	SECTION	22	SYBLOC	SECTION	23

EPN	IFN	LOCATION	EPN	IFN	LOCATION	EPN	IFN	LOCATION
10000	37A	04504	32010	FORMAT	04250	6000	31A	04460
30000	FORMAT	04310	8000	28A	04447	32030	FORMAT	04333

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 04332.

TR-67-700-10-2
Page III-2-58
Program Listing
Link 2

IMP.ND:0001.44000-0.77307.AM IC

ISJOB LION: 7004 A 02/18/68

PAGE 4

00
BTITLE ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES
STPTC RANGE? M04.MRY LINK 2 (RANGE) R5001000

ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES
RANGEZ - EPN SOURCE STATEMENT - IPN(3) -

02/16/68

PAGE 1

```

SUBROUTINE RANGE( YMAXIM )                                RG001200
C
C *** INPUTS ***                                         RG001300
C YMAXIM MAXIMUM ALLOWABLE Y FROM OPTIC AXIS.           RG001400
C ELMNTX MATRIX OF EDITED SURFACE ELEMENTS.           (ELMAIR COMMON) RG001700
C SURFC ARRAY OF SURFACE PARAMETERS.                  (DATA COMMON)  RG001800
C
C *** OUTPUTS ***                                       RG002000
C ZHAT MATRIX OF RANGE OF Z DEFINITION FOR EACH      (CODRNG COMMON) RG002100
C SURFACE REFERENCED FROM ITS VERTEX.                 RG002200
C YHAT MATRIX OF RANGE OF Y DEFINITION FOR EACH      (CODRNG COMMON) RG002300
C SURFACE CORRESPONDING TO Z DEFINITION.             RG002400
C CHAT MATRIX OF RANGE CODES. SET = 1 INITIALLY.     (CODRNG COMMON) RG002500
C RESET = 2 IF Y OF SURFACE EXCEEDS YMAXIM.         RG002600
C
C *** DESCRIPTION ***                                   RG002700
C THIS SUBROUTINE COMPUTES THE UNCONSTRAINED RANGE IN Z, ZHAT,
C AND THE RANGE IN Y, YHAT, FOR EVERY SURFACE. THE Z VALUE IS
C REFERENCED FROM THE VERTEX OF THAT SURFACE. THE CODE MATRIX
C VALUE = 2 WHEN A SURFACE EXTENDS BEYOND YMAXIM, OTHERWISE = 1. RG002800
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483)                                           DT001100
DIMENSION WOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLR8(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRRAS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMG2), (DATA(24), DT101700
8 OMG1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PBC/L), (DATA(28), OMGF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), HO), (DATA(33), DT102000
B DELH), (DATA(34), SYSHX), (DATA(35), WDIR), (DATA(36), DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLRH(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLR8(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2102), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2602), SUBST(1), ISUBST(1)), (DATA(3102), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOMP) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
COMMON/ELMAIR/NELMT,NAIRS,ELMNTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
INTEGER ELMNTX, AIRMTX EL002000
C
COMMON / CODRNG / ZHAT(2,100), YHAT(2,100), CHAT(2,100) C0001000
INTEGER CHAT C0002000
    
```

ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES
RANGEZ - EFN SOURCE STATEMENT - IPN(S) -

02/16/68

PAGE 2

```

C
C   DIMENSION ZRANGE(2), YRANGE(2)                                RC 002900
C
C   INTEGER SURFNO                                               RC 003000
C
C       START OF RANGE SUBROUTINE                                RC 003100
C
C   DO 500 I = 1, 2                                              RC 003200
C   DO 500 J = 1, 100                                           RC 003400
C   ZHAT(I,J) = 0.0                                             RC 003600
C   YHAT(I,J) = 0.0                                             RC 003800
C   500 CHAT(I,J) = 0                                           RC 004000
C   DO 20000 I = 1, NELMT                                       RC 004200
C   DO 20000 J = 1, 2                                           RC 004400
C
C       THIS SECTION DETERMINES THE RANGE IN Z OF THE SURFACE  RC 004600
C       GIVEN IN SURFNO. THE RANGE IN Y, YRANGE, CORRESPONDING RC 004700
C       TO ZRANGE IS ALSO COMPUTED                               RC 004800
C
C   SURFNO = ELMTX(J,I)                                         RC 004900
C   ZRANGE(1) = 0.                                             RC 005000
C   ZRANGE(2) = 0.                                             RC 005200
C   YRANGE(1) = 0.                                             RC 005400
C   YRANGE(2) = YMAXIM                                         RC 005600
C   ITYPE = ISURFC(1,SURFNO)                                   RC 005800
C   R = SURFC(11,SURFNO)                                       RC 006000
C   B = SURFC(12,SURFNO)                                       RC 006200
C   IF ( ITYPE .EQ. 3 ) GO TO 12000                             RC 006400
C   IF ( R .EQ. 0. ) GO TO 12000                                RC 006600
C   IF ( B .NE. 0. ) GO TO 4000                                 RC 006800
C   IF ( R .LT. 0. ) GO TO 2000                                 RC 007000
C
C       CONVEX CIRCLE                                           RC 008000
C   IF ( YMAXIM .LT. R ) GO TO 1000                             RC 008200
C   ZRANGE(2) = R                                               RC 008400
C   YRANGE(2) = R                                               RC 008600
C   GO TO 12000                                                 RC 008800
C   1000 ZRANGE(2) = R - SQRT( R*R - YMAXIM*YMAXIM )           RC 009000   42
C   GO TO 12000                                                 RC 009200
C
C       CONCAVE CIRCLE                                          RC 101000
C   2000 IF ( YMAXIM .LT. ABS( R ) ) GO TO 3000                RC 101200
C   ZRANGE(1) = R                                               RC 101400
C   YRANGE(2) = ABS( R )                                         RC 101600
C   GO TO 12000                                                 RC 101800
C   3000 ZRANGE(1) = R + SQRT( R*R - YMAXIM*YMAXIM )           RC 102000   48
C   GO TO 12000                                                 RC 102200
C   4000 IF ( B .NE. 1. ) GO TO 5000                             RC 102400
C
C       PARABOLA                                               RC 102600
C   4002 ZT=YMAXIM*YMAXIM/(2.*R)                                RC 102800
C   K = 1                                                         RC 103000
C   IF ( R .GT. 0. ) K = 2                                       RC 103200
C   ZRANGE(K) = ZT                                               RC 103400
C   GO TO 12000                                                 RC 103600
C   5000 IF ( B .GT. 1. ) GO TO 9000                             RC 104000
C
C

```

ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES
RANGEZ - EFM SOURCE STATEMENT - IPN(8) -

02/16/88

PAGE 3

C	ELLIPSE	RG174200	
	A1 = ABS(R) / SQRT(1. - B)	RG174400	63
	A2 = R / (1.-B)	RG174600	
	IF(R .LT. 0.) GO TO 7000	RG174800	
C	CONVEX ELLIPSE	RG175000	
	IF(YMAXIM .LT. A1) GO TO 8000	RG175200	
	ZRANGE(2) = A2	RG175400	
	YRANGE(2) = A1	RG175600	
	GO TO 12000	RG175800	
	8000 ZRANGE(2) = A2 - SQRT((A1 * A1 - YMAXIM * YMAXIM) / (1.-B))	RG176000	72
	GO TO 12000	RG176200	
C	CONCAVE ELLIPSE	RG176400	
	7000 IF(YMAXIM .LT. A1) GO TO 8000	RG176600	
	ZRANGE(1) = A2	RG176800	
	YRANGE(2) = A1	RG177000	
	GO TO 12000	RG177200	
	8000 ZRANGE(1) = A2 + SQRT((A1*A1 - YMAXIM*YMAXIM) / (1.-B))	RG177400	79
	GO TO 12000	RG177600	
	9000 IF((B-1.) .LT. 1.E-5) GO TO 4000	RG177800	
	A2 = -R / (B-1.)	RG178000	
	A1 = ABS(R) / SQRT(B-1.)	RG178200	84
	DELTA = SQRT((A1*A1 + YMAXIM*YMAXIM) / (B-1.))	RG178400	85
	IF(R .GT. 0.) GO TO 10000	RG201000	
C	CONCAVE HYPERBOLA	RG201200	
	ZRANGE(1) = A2 - DELTA	RG201400	
	GO TO 12000	RG201600	
C	CONVEX HYPERBOLA	RG202000	
	10000 ZRANGE(2) = A2 + DELTA	RG202200	
	12000 YHAT(J,I) = YRANGE(2)	RG202400	
	CHAT(J,I) = 1	RG202600	
	IF(YHAT(J,I) .EQ. YMAXIM) CHAT(J,I) = 2	RG202800	
	ZHAT(J,I) = ZRANGE(1)	RG203000	
	IF(ZRANGE(1) .EQ. 0.) ZHAT(J,I) = ZRANGE(2)	RG203200	
	20000 CONTINUE	RG203400	
C	END OF RANGE SUBROUTINE	RG203600	
	RETURN	RG203800	
	END	RG204000	

ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES
RANGEZ STORAGE MAP

02/16/68

PAGE 4

SUBROUTINE RANGE
COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	00000	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00039	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NTPLN	00042	I	INODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
HEKPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGAS	00061	R	OMGAS	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	00071	R	HO	00072	R	DELH	00073	R
SYBMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
NOBJH	00117	R	WCLRH	00126	R	WINHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUND8	05210	R
IBND8	05210	I	NCOND	05665	I			

COMMON BLOCK			ELMAIR	ORIGIN	05667	LENGTH	00766	
NELMT	00000	I	NAIRS	00001	I	ELMHTX	00002	I
AIRMTX	00312	I	VTXCRD	00822	R			

COMMON BLOCK			CODRNG	ORIGIN	07655	LENGTH	01130	
ZHAT	00000	R	YHAT	00310	R	CHAT	00620	I

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ZRANGE	11005	R	YRANGE	11007	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SURFNO	11011	I	I	11012	I	ITYPE	11013	I
R	11014	R	B	11015	R	ZT	11016	R
K	11017	I	A1	11020	R	A2	11021	R
DELTA	11022	R						

ENTRY POINTS

ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES
 RANGEZ STORAGE MAP

02/16/68

PAGE 8

SUBROUTINES CALLED

SORT			SYSLOC			SECTION II		
EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
800	9A	11081	20000	102A	11950	12000	91A	11930
4000	81A	11292	2000	44A	11220	1000	41A	11201
3000	48A	11234	8000	60A	11303	4002	94A	11296
6000	81A	11436	7000	74A	11400	6000	71A	11393
8000	78A	11412	10000	90A	11929			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 11023.

TR-67-700-10-2
Page III-2-64
Program Listing
Link 2

INF.NOJ0001.44800-0.77387.AM IC ISJOB L10MR: 7084 A 02/16/68 PAGE 6

BT
BTITLE ROUTINE TO FIND INTERSECTION OF CONIC AND ST. LINE
BTBPTC ZYCUTZ M94,XRY LINK 2 (ZYCUT) ZY001000

02/16/68

PAGE 1

ROUTINE TO FIND INTERSECTION OF CONIC AND ST. LINE
ZYCUTZ - EPN SOURCE STATEMENT - I/PN(8) -

```

SUBROUTINE ZYCUT( VHCODE, ALFA, IS, BETA, *, ELNO, SURNO )      ZY001900
C
C   *** INPUTS ***                                           ZY002000
C VHCODE FLAG FOR VERTICAL, =2, OR HORIZONTAL, =1, LINE.      ZY002100
C ALFA   Y OR Z VALUE OF LINE.                                ZY002200
C IS     SURFACE NUMBER TO BE INTERSECTED BY LINE.           ZY002300
C ELNO   ELEMENT NUMBER REFERENCED TO THAT OR ZHAT.          ZY002400
C SURNO  VALUE SPECIFYING FIRST OR SECOND SURFACE OF GIVEN ELEMENT. ZY002500
C
C   *** OUTPUTS ***                                           ZY003000
C BETA   Y OR Z VALUE OF COMPUTED INTERSECTION.              ZY003100
C *      ALTERNATE RETURN FOR NO INTERSECTION.                ZY003200
C
C   *** DESCRIPTION ***                                       ZY003500
C THIS SUBROUTINE DETERMINES THE INTERSECTION, BETA, OF THE ZY003600
C LINE THROUGH ALFA AND CURVE NUMBER IS. IF VHCODE=1, THE ZY003700
C LINE IS HORIZONTAL, IF VHCODE=2, THE LINE IS VERTICAL.      ZY003800
C THE ALTERNATE RETURN IS USED WHEN THERE IS NO INTERSECTION. ZY003900
C
COMMON / DATA / MCNTRL, CONTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483) DT001100
DIMENSION MOBJH(7), WCLRH(7), WINHT(7), EINHT(7), CIMPL(7), DT002000
1 WCLR8(8), NDSGN(4), NDSGV(11), LATTC(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATTC(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAY8 ), ( DATA(3), DT101000
1 NBLCS ), ( DATA(4), NCLR8 ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
2 NSUBT ), ( DATA(7), NSURP ), ( DATA(8), NIFLN ), ( DATA(9), DT101200
3 IMODE ), ( DATA(10), NSPLN ), ( DATA(11), MOBJH ), ( DATA(12), DT101300
4 NSURP ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
5 PNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18), DT101500
6 ZETA ), ( DATA(19), HEXPP ), ( DATA(20), DEXPP ), ( DATA(21), DT101600
7 WEXPP ), ( DATA(22), DLPLN ), ( DATA(23), OMGAS ), ( DATA(24), DT101700
8 OMGAS ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
9 PSCAL ), ( DATA(28), OMGAP ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
A JUMIN ), ( DATA(31), ODIST ), ( DATA(32), HO ), ( DATA(33), DT102000
B DELH ), ( DATA(34), SYMOK ), ( DATA(35), WKDIR ), ( DATA(36), DT102100
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
D NDSGV(1) ), ( DATA(39), MOBJH(1) ), ( DATA(40), WCLRH(1) ), DT102300
E ( DATA(47), WINHT(1) ), ( DATA(74), EINHT(1) ), ( DATA(81), DT102400
F CIMPL(1) ), ( DATA(88), WCLR8(1) ), DT102500
G ( DATA(94), LATTC(1,1), ILATTC(1,1) ), ( DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1) ), ( DATA(2182), DESGN(1,1), IDESGN(1,1) ), DT102700
I ( DATA(2682), SUBST(1), ISUBST(1) ), ( DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1) ), ( DATA(3483), NCOND ) DT102900
INTEGER CONTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
COMMON / CODRNG / ZHAT(2,100), YHAT(2,100), CHAT(2,100) CD001000
INTEGER CHAT CD002000
C
INTEGER VHCODE, ELNO, SURNO ZY004000
C
C START OF ZYCUT SUBROUTINE ZY004100
C
S

```

ROUTINE TO FIND INTERSECTION OF CONIC AND ST. LINE
 ZYCUTZ - EPN SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 2

BETA = 0.	ZY004200	
IT = ISUMFC(1,18)	ZY004400	
R = SUMFC(11,18)	ZY004600	
B = SUMFC(12,18)	ZY004800	
IF(IT .EQ. 3) GO TO 2000	ZY005000	
IF(R .NE. 0.) GO TO 4000	ZY005200	
2000 IF(VHCODE .EQ. 2) RETURN ;	ZY005400	
IF(ALFA .LT. 0. .OR. ALFA .GT. YHAT(SURNO,ELND)) RETURN ;	ZY005600	
GO TO 32000	ZY005800	
4000 IF(VHCODE .NE. 1) GO TO 6000	ZY006000	
IF(ALFA .LT. 0. .OR. ALFA .GT. YHAT(SURNO,ELND)) RETURN ;	ZY006200	
BETA = ALFA*ALFA / (R+SIGN(SRT(R*R - (1.-B)*ALFA*ALFA), R))	ZY006400	29
GO TO 32000	ZY006600	
6000 ZMIN = AMIN(0., ZHAT(SURNO,ELND))	ZY006800	
ZMAX = AMAX(0., ZHAT(SURNO,ELND))	ZY007000	
IF(ALFA .LT. ZMIN .OR. ALFA .GT. ZMAX) RETURN ;	ZY007200	
BETA = SRT((B-1.)*ALFA*ALFA + 2.*R*ALFA)	ZY007400	
C		
END OF ZYCUT SUBROUTINE	ZY008000	32
32000 RETURN	ZY008200	
END	ZY008400	

ROUTINE TO FIND INTERSECTION OF CONIC AND ST. LINE
 ZYCUTZ STORAGE MAP

02/18/68

PAGE 3

SUBROUTINE ZYCUT
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	00000	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00007	I	PUNCTD	00032	R	DATA	00033	R
UNODE	00033	I	NRAYS	00034	I	NSLCS	00039	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00049	I	NSURF	00046	I
FLAG	00047	I	DELY	00080	R	FRUMB	00081	R
FLNGH	00082	R	WFLSH	00083	R	ZETA	00084	R
HEXPP	00085	R	DEKPP	00086	R	WEXPP	00087	R
DLPLH	00080	R	OMGAE	00081	R	OMGAI	00082	R
DELD	00083	R	EPRAD	00084	R	PSCAL	00085	R
OMGAF	00086	R	SPFEA	00087	R	DUMIN	00070	R
CDIST	00071	R	HD	00072	R	DELH	00073	R
SYBHX	00074	R	WDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NOBGN	00100	I	NOSEV	00104	I
NOBJH	00117	R	WCLRH	00126	R	WINHT	00139	R
EINHNT	00144	R	CINPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	00224	R	ISUBST	00224	I	BOUND8	00210	R
IBND8	00210	I	NOOND	00665	I			

COMMON BLOCK			00000	ORIGIN	00007	LENGTH	01130	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ZHAT	00000	R	YHAT	00310	R	CHAT	00820	I

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IT	10017	I	R	10020	R	B	10021	R
ZMIN	10022	R	ZMAX	10023	R			

ENTRY POINTS

ZYCUT SECTION 7

SUBROUTINES CALLED

SORT SECTION 8

SYSLOC SECTION 9
 EFN IFN CORRESPONDENCE

.FRET. SECTION 10

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
2000	12A	10083	4000	19A	10107	32000	33A	10241
6000	27A	10162						

TR-67-700-10-2
Page III-2-68
Program Listing
Link 2

ROUTINE TO FIND INTERSECTION OF CONIC AND ST. LINE
ZYCUTZ STORAGE MAP

08/16/68

PAGE 4

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 10330.

TR-67-700-10-2
Page III-3-1
Program Listing
Link 3

Section 1

LINK 3

TR-67-700 1-2
Page III-3-2
Program Listing
Link 3

INF.NO10001.44800-0.77387.AM IC 18JOB LINK1 7094 A 02/18/69 PAGE 3

00
BTITLE SUBROUTINE TO COMPUTE TWINRAY 4 DIAGNOSTIC INFORMATION TR401000
STEPIC TWNR4Z MD4.HR7 LINK 3 (TWNRY4)

SUBROUTINE TO COMPUTE TWINRAY & DIAGNOSTIC INFORMATION
 TWINRAY - EFN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 1

```

SUBROUTINE TWINRAY                                     TR401200
C
C *** OUTPUTS ***                                     TR401300
C XMTRX1 MATRIX OF RAY 1 POSITIONS AT EACH SURFACE. TR401310
C XMTRX2 MATRIX OF RAY 2 POSITIONS AT EACH SURFACE. TR401320
C QMTRX1 MATRIX OF RAY 1 DIRECTIONS AT EACH SURFACE. TR401330
C QMTRX2 MATRIX OF RAY 2 DIRECTIONS AT EACH SURFACE. TR401340
C TAU1 THICKNESS FOR RAY 1.                          TR401350
C TAU2 THICKNESS FOR RAY 2.                          TR401360
C AL APERTURE LOCATION.                              TR401370
C AR APERTURE RADIUS.                                TR401380
C
C *** DESCRIPTION ***                                 TR401400
C THIS SUBROUTINE TRACES 2 SPECIAL RAYS AND COMPUTES THEIR TR401410
C POSITION AND DIRECTION AT EACH SURFACE. THICKNESS, APERTURE TR401420
C RADIUS AND APERTURE LOCATION ARE ALSO DETERMINED. TR401430
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(5), PUNCID, DT001000
1 DATA(3483)                                          DT001100
DIMENSION WOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(5,2), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAY5), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMCOE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMGA2), (DATA(24), DT101700
8 OMGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), MO), (DATA(33), DT102000
B DELH), (DATA(34), SYSMX), (DATA(35), WXDYR), (DATA(36), DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLRH(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
INTEGER CNTRL, AFLAG, DATE DT?01000
REAL LATTC DT202000
C
COMMON/AZOBJ/THTR,SNTHTR,C8THTR                       RY004000
C
COMMON / PRNT / LINE, PAGE                             PT002500
INTEGER PAGE                                           PT002700
C
COMMON / TMTWIN / TEMPS(1200)                          L2001000
EQUIVALENCE (TEMPS(1), XMTRX1(1,1)), (TEMPS(301), QMTRX1(1,1)) L2002000
1, (TEMPS(601), XMTRX2(1,1)), (TEMPS(901), QMTRX2(1,1)) L2003000
DIMENSION XMTRX1(3,100), QMTRX1(3,100), XMTRX2(3,100), L2004000

```

SUBROUTINE TO COMPUTE TWINRAY 4 DIAGNOSTIC INFORMATION
TWR4Z - EFM SOURCE STATEMENT - IFN(S) -

02/18/68

PAGE 2

	1 ENTER2(3,100)	L2008000	
C	INTEGER RAY, SMISS	TR401800	
C	REAL N1, N2, NSURF	TR402000	
	DIMENSION XVCT(3), QVCT(3)	TR402500	
C			
C	TWR4Z MUST BE CALLED FROM MAIN PROGRAM FOR EACH	TR403000	
C	SUBSTITUTION SET AS FOLLOWS -	TR403200	
C			
C	100 PRINT OUT	TR403400	
C	CALL TWR4Z	TR403600	
C	CALL PRMSUB(8100)	TR403800	
C	DONE	TR404000	
C			
C	START OF TWR4Z SUBROUTINE	TR405000	
C	CALL PRCTL	TR405200	2
	HMAX = MD + FLOAT(NOBJH - 1) * DELH	TR405400	
	WRITE (6, 32000) HMAX	TR405600	4
32000	FORMAT(1HD, 34HTWYNRAY FOR COLOR 1 OBJECT POINT (, E14.7, 4H,D.,)	TR405800	
	LINE = LINE + 2	TR406000	
	DO 500 I = 1, 3	TR406200	
	DO 500 J = 1, 100	TR406400	
	XNTRX1(I,J) = 0.0	TR406600	
	QNTRX1(I,J) = 0.0	TR406800	
	XNTRX2(I,J) = 0.0	TR407000	
500	QNTRX2(I,J) = 0.0	TR407200	
	YNTR = 0.0	TR407400	
	SNTR = 0.0	TR407600	
	CSNTR = 1.0	TR407800	
	N1 = NSURF	TR408000	
	N2 = NSURF	TR408200	
	N = NSURF + 1	TR408400	
C			
C	TRACE RAY 1	TR421000	
	RAY = 1	TR421200	
	CALL RAYTR(0., -1., HMAX, 1, XVCT, QVCT, SMISS, 8000, 8900,	TR421400	
	1 8100, 1, XNTRX1, QNTRX1)	TR421600	23
	GO TO 1100	TR421800	
C			
C	RAY 1 MISSED SURFACE NO. SMISS	TR422000	
800	N1 = SMISS - 1	TR422200	
850	WRITE (6, 32010) RAY, SMISS	TR422400	27
32010	FORMAT(1HD, 3HRAY, I2, 15H MISSED SURFACE, I3)	TR422600	
	GO TO 1050	TR422800	
C			
C	RAY 1 REFLECTED ON SURFACE NO. SMISS	TR423000	
900	N1 = SMISS - 1	TR423200	
950	WRITE (6, 32020) RAY, SMISS	TR423400	30
32020	FORMAT(1HD, 3HRAY, I2, 21H REFLECTED ON SURFACE, I3)	TR423600	
	GO TO 1050	TR423800	
C			
C	RAY 1 OR 2 VIGNETTED ON SURFACE NO. SMISS	TR424000	
1000	WRITE (6, 32030) RAY, SMISS	TR424200	32
32030	FORMAT(1HD, 3HRAY, I2, 21H VIGNETTED ON SURFACE, I3)	TR424400	

SUBROUTINE TO COMPUTE TWINRAY & DIAGNOSTIC INFORMATION
TWR4Z - EPN SOURCE STATEMENT - IPN(8) -

02/18/68

PAGE 3

1090 LINE = LINE + 2	TR424900	
1100 IF (RAY .NE. 1) GO TO 1800	TR424700	
C		
C TRACE RAY 2	TR425000	
RAY = 2	TR425200	
CALL RAYTR(D., 1., HMAX, 1, XVCT, OVCT, SMISS, 81300, 81400,	TR425500	
1 81000, 1, XMTRX2, QMTRX2)	TR425600	38
GO TO 1800	TR426000	
C		
C RAY 2 MISSED SURFACE NO. SMISS	TR427000	
1300 N2 = SMISS - 1	TR427200	
GO TO 850	TR427400	
C		
C RAY 2 REFLECTED ON SURFACE NO. SMISS	TR428000	
1400 N2 = SMISS - 1	TR428200	
GO TO 850	TR428400	
1600 N = MIND(N1, N2)	TR429000	
C		
C PRINT TWINRAY HEADING	TR441000	
WRITE (6, 32040)	TR441500	46
32040 FORMAT(1HD, 12HRAY SURFACE, 9X, 1HX, 15X, 1HY, 15X, 1HZ, 14X,	TR441700	
1 2HQX, 14X, 2HQY, 14X, 2HQZ, 11X, 9HTHICKNESS)	TR441900	
LINE = LINE + 2	TR442000	
DO 5000 J = 1, N	TR442500	
I = J - 1	TR442800	
TAU1 = 0.	TR443000	
TAU2 = 0.	TR443200	
AR = -1.	TR443400	
IF(I .GT. N1) GO TO 475	TR443600	
TAU1 = - XMTRX1(3,J)	TR443800	
IF(I .LT. NSURF) TAU1 = TAU1 + XMTRX1(3,J+1) + SURFC(4,J)	TR444000	
IF(SURFC(5,J) .LT. 0.) TAU1 = - TAU1	TR444200	
475 IF(I .GT. N2) GO TO 477	TR445000	
TAU2 = - XMTRX2(3,J)	TR445200	
IF(I .LT. NSURF) TAU2 = TAU2 + XMTRX2(3,J+1) + SURFC(4,J)	TR445400	
IF(SURFC(5,J) .LT. 0.) TAU2 = - TAU2	TR445600	
477 CONTINUE	TR445800	
IF(QMTRX1(3,J) .EQ. 0.0 .OR. QMTRX2(3,J) .EQ. 0.0) GO TO 3000	TR446000	
M1 = QMTRX1(2,J) / QMTRX1(3,J)	TR446200	
M2 = QMTRX2(2,J) / QMTRX2(3,J)	TR446400	
MSUM = M1 + M2	TR446500	
IF(MSUM .EQ. 0.0) GO TO 3000	TR446800	
AL = (M1 * XMTRX1(3,J) - XMTRX1(2,J) + M2 * XMTRX2(3,J) -	TR447000	
1 XMTRX2(2,J)) / MSUM	TR447200	
AR = ABS(M1 * (AL - XMTRX1(3,J)) + XMTRX1(2,J))	TR448000	
3000 IF(LINE .LE. 52) GO TO 3500	TR448200	
C		
C BEGIN NEW PAGE - PRINT TITLE	TR448500	
CALL PRCTL	TR448600	96
WRITE (6, 32040)	TR448800	97
LINE = LINE + 2	TR449000	
C		
C PRINT OUT TWINRAY INFORMATION	TR461000	
3500 ITEM = 1	TR461200	
WRITE (6, 32045)	TR461300	99
32045 FORMAT(1H)	TR461350	

SUBROUTINE TO COMPUTE TWINRAY 4 DIAGNOSTIC INFORMATION
 TWR42 - EFM SOURCE STATEMENT - IFN(8) -

02/16/58

PAGE 4

WRITE (6, 32050) ITENP, 1, (XNTRX1(K,J), R=1,3),	TR461400	
1 (QNTRX1(K,J), R=1,3), TAU1	TR461500	100
ITENP = 2	TR462000	
WRITE (6, 32050) ITENP, 1, (XNTRX2(K,J), R=1,3),	TR462200	
1 (QNTRX2(K,J), R=1,3), TAU2	TR462300	100
32050 FORMAT(1H , 12, 17, 3X, 7E16.7)	TR462900	
LINE = LINE + 3	TR462700	
IF (AR .LT. 0.0) GO TO 5000	TR463000	
C		
PRINT OUT APERTURE INFORMATION	TR463500	
C	TR463800	121
WRITE (6, 32050) AL, AR	TR464000	
32050 FORMAT(1H0, 14X, 18APERTURE LOCATION=, E14.7, 18H APERTURE RADITR464200	TR464200	
SUS=, E14.7)	TR465000	
LINE = LINE + 2	TR466000	
5000 CONTINUE		
C		
END OF TWR42 SUBROUTINE	TR461700	
C	TR462000	
RETURN	TR463000	
END		

SUBROUTINE TO COMPUTE TWINRAY-4 DIAGNOSTIC INFORMATION
TWR42 STORAGE MAP

02/16/68

PAGE 8

SUBROUTINE TWR44
COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	00000		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R	
DATE	00007	I	PUNCI0	00002	R	DATA	00033	R	
LMODE	00033	I	NRAYS	00004	I	NSLCS	00035	I	
NCLRS	00036	I	NJAIL	00007	I	NSUBT	00040	I	
NSUBP	00041	I	NJPLN	00012	I	TMODE	00043	I	
NSPLN	00044	I	NBJH	00015	I	NSURF	00046	I	
FLAG	00047	I	DELY	00020	R	FNUMB	00051	R	
FLNSH	00052	R	WFLSH	00023	R	ZETA	00054	R	
MXPP	00055	R	DEXPP	00026	R	WEXPP	00057	R	
DLPLN	00058	R	ONGA2	00031	R	ONGA1	00062	R	
DELD	00063	R	EPRAD	00034	R	PSCAL	00065	R	
ONGAP	00066	R	SPFEA	00037	R	DUMIN	00070	R	
ODIST	00071	R	HO	00072	R	DELH	00073	R	
SYNCR	00074	R	WDJR	00075	R	WYJR	00076	R	
NDTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I	
NBJH	00117	R	WCLR	00125	R	WINHT	00135	R	
EIMNT	00144	R	CIMPL	00153	R	WCLRS	00162	R	
LATTC	00170	R	ILATTC	00170	I	SURFC	00220	R	
ISURFC	00220	I	DESGN	00240	R	IDESGN	00240	I	
SURST	00224	R	ISURST	00224	I	BOUND8	00210	R	
IBND8	00210	I	NCOND	00225	I				

COMMON BLOCK			AZOBJ	ORIGIN	00007	LENGTH	00003
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002
COMMON BLOCK			PRNT	ORIGIN	00072	LENGTH	00002
LINE	00000	I	PAGE	00001	I		
COMMON BLOCK			TMTWIN	ORIGIN	00074	LENGTH	00200
TEMP8	00000	R	XNTRX1	00000	R	GNTRX1	00454
XNTRX2	01130	R	GNTRX2	01604	R		

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
XVCT	11134	R	GVCT	11157	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RAY	11162	I	SNISS	11163	I	M1	11164	R
ME	11165	R	NSUN	11166	R	MMAX	11167	R
I	11170	I	J	11171	I	N1	11172	I
NE	11173	I	N	11174	I	M	11175	I
TAU1	11176	R	TAU2	11177	R	AR	11200	R
AL	11201	R	ITEMP	11202	I			

SUBROUTINE TO COMPUTE TWINRAY 4 DIAGNOSTIC INFORMATION
 TWMR4Z STORAGE MAP

02/18/68

PAGE 8

ENTRY POINTS

TWMR4 SECTION 11

SUBROUTINES CALLED

PRTCTL SECTION 12
 .UNNS SECTION 18
 E.1 SECTION 19
 E.4 SECTION 21
 CC.3 SECTION 24

.FWRD. SECTION 13
 .FFIL. SECTION 16
 E.2 SECTION 19
 CC.1 SECTION 22
 CC.4 SECTION 25
 EFN IFN CORRESPONDENCE

RAYTR SECTION 14
 .FCNV. SECTION 17
 E.3 SECTION 20
 CC.2 SECTION 23
 SYSLOC SECTION 26

EFN	IFN	LOCATION
32000	FORMAT	11220
800	29A	11473
80	27A	11480
250	30A	11476
3200	45A	11568
32040	FORMAT	11261
477	74A	11675
32045	FORMAT	11303

EFN	IFN	LOCATION
500	14A	11407
1000	32A	11511
32010	FORMAT	11232
32020	FORMAT	11241
1300	41A	11595
9000	122A	12124
3000	93A	11744
32050	FORMAT	11308

EFN	IFN	LOCATION
800	26A	11455
1100	34A	11526
1080	33A	11523
32030	FORMAT	11251
1400	43A	11561
475	63A	11650
3500	98A	11765
32050	FORMAT	11312

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12156.

TR-67-700-10-2
Page III-3-9
Program Listing
Link 3

IMP.MD10001.44000-0.77307.AM IC

10JOB L10MR1 7004 A

02/16/68

PAGE 7

00

BTITLE SUBROUTINE TO COMPUTE TWINRAY S DIAGNOSTIC INFORMATION

SIBFTC TWR5Z M04.XR7

LINK 3 (TWRYS)

TR501000

SUBROUTINE TO COMPUTE TWINRAY S DIAGNOSTIC INFORMATION
 TWINRS2 - EPN SOURCE STATEMENT - IPN(S) -

02/18/68

PAGE 1

```

SUBROUTINE TWINRS2                                     TR501200
C
C *** OUTPUTS ***                                     TR501300
C XNTRX1 MATRIX OF RAY 1 POSITIONS AT EACH SURFACE.   TR501310
C XNTRX2 MATRIX OF RAY 2 POSITIONS AT EACH SURFACE.   TR501320
C QNTRX1 MATRIX OF RAY 1 DIRECTIONS AT EACH SURFACE.  TR501330
C QNTRX2 MATRIX OF RAY 2 DIRECTIONS AT EACH SURFACE.  TR501340
C TAU1 THICKNESS FOR RAY 1.                            TR501350
C TAU2 THICKNESS FOR RAY 2.                            TR501360
C
C *** DESCRIPTION ***                                   TR501600
C THIS SUBROUTINE TRACES 2 SPECIAL RAYS AND COMPUTES THEIR TR501610
C POSITION AND DIRECTION AT EACH SURFACE. THICKNESS IS ALSO TR501620
C DETERMINED FOR EACH RAY.                             TR501630
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3485)                                           DT001100
DIMENSION NOBJH(7), WCLRH(7), WINHT(7), EINHT(7), CIMPL(7), DT002000
1 WCLR8(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE ( DATA(1), LMODE ), ( DATA(2), NRRYS ), ( DATA(3), DT101000
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9), DT101200
3 IMODE ), ( DATA(10), NSPLN ), ( DATA(11), NOBJH ), ( DATA(12), DT101300
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
5 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLC ), ( DATA(18), DT101500
6 ZETA ), ( DATA(19), WEXPP ), ( DATA(20), DEXF ), ( DATA(21), DT101600
7 WEXPP ), ( DATA(22), DLPLH ), ( DATA(23), ONGA2 ), ( DATA(24), DT101700
8 ONGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
9 PSICAL ), ( DATA(28), ONGAF ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
A DUMIN ), ( DATA(31), ODIST ), ( DATA(32), HO ), ( DATA(33), DT102000
B DELH ), ( DATA(34), SYBCK ), ( DATA(35), WDIR ), ( DATA(36), DT102100
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
D NDSGV(1) ), ( DATA(53), NOBJH(7) ), ( DATA(60), WCLRH(1) ), DT102300
E ( DATA(67), WINHT(1) ), ( DATA(74), EINHT(1) ), ( DATA(81), DT102400
F CIMPL(1) ), ( DATA(88), WCLR8(1) ), DT102500
G ( DATA(94), LATT(1,1), ILATT(1,1) ), ( DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1) ), ( DATA(2102), DESGN(1,1), IDESGN(1,1) ), DT102700
I ( DATA(2602), SUBST(1), ISUBST(1) ), ( DATA(3102), BOUNDS(1,1), DT102800
J IBNDS(1,1) ), ( DATA(3485), NCOND ) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
COMMON / A2OBJ / THTR, SNTHTR, CSTHTR RY004000
C
COMMON / PRINT / LINE, PAGE PT002500
INTEGER PAGE PT002700
C
COMMON / TMTWIN / TEMPS(1200) L2001000
EQUIVALENCE ( TEMPS(1), XNTRX1(1,1) ), ( TEMPS(301), QNTRX1(1,1) ) L2002000
1, ( TEMPS(601), XNTRX2(1,1) ), ( TEMPS(901), QNTRX2(1,1) ) L2003000
DIMENSION XNTRX1(3,100), QNTRX1(3,100), XNTRX2(3,100), L2004000
1 QNTRX2(3,100) L2005000
C

```

02/16/68

PAGE 2

SUBROUTINE TO COMPUTE TWIRAY'S DIAGNOSTIC INFORMATION
TWNRSZ - EFN SOURCE STATEMENT - IPN(8) -

DIMENSION XVCT(3), QVCT(3)	TR002500	
INTEGER RAY, SMISS	TR003000	
C		
C TWNRYS MUST BE CALLED FROM MAIN PROGRAM EACH SUBSTITUTION	TR003500	
C SET AS FOLLOWS -	TR003100	
C 100 PRINT OUT	TR003200	
C CALL TWNRYS	TR003300	
C CALL PRMSUB(3100)	TR003400	
C DONE	TR003500	
C		
C START OF TWNRYS SUBROUTINE	TR003500	
C		
CALL PRCTCL	TR003500	2
WRITE (6, 32000)	TR101000	3
32000 FORMAT(1H0, 41HTWINRAY FOR COLOR 1 OBJECT POINT (0.0.0))	TR101200	
LINE = LINE + 2	TR101300	
DO 900 I = 1, 3	TR101500	
DO 900 J = 1, 100	TR101600	
XNTRX1(I,J) = 0.0	TR101700	
QNTRX1(I,J) = 0.0	TR101800	
XNTRX2(I,J) = 0.0	TR101900	
900 QNTRX2(I,J) = 0.0	TR102000	
THTR = 0.0	TR102500	
SNTHTR = 0.0	TR102600	
CBTHTR = 1.0	TR102700	
N1 = NSURF	TR103000	
N2 = NSURF	TR103100	
N = NSURF + 1	TR103200	
C		
C TRACE RAY 1	TR104000	
RAY = 1	TR104200	
CALL RAYTR(0., OMCAR, 0., 1, XVCT, QVCT, SMISS, \$700, \$800,	TR104500	
1 \$900, 1, XNTRX1, QNTRX1)	TR104600	22
GO TO 1000	TR104700	
C		
C RAY 1 MISSED SURFACE NO. SMISS	TR105000	
700 N1 = SMISS - 1	TR105200	
750 WRITE (6, 32010) RAY, SMISS	TR105500	23
32010 FORMAT(1H0, 3HRAY, 12, 15H MISSED SURFACE, 13)	TR105600	
LINE = LINE + 2	TR105800	
GO TO 1000	TR105900	
C		
C RAY 1 REFLECTED ON SURFACE NO. SMISS	TR106000	
800 N1 = SMISS - 1	TR106200	
850 WRITE (6, 32020) RAY, SMISS	TR106500	29
32020 FORMAT(1H0, 3HRAY, 12, 21H REFLECTED ON SURFACE, 13)	TR106600	
LINE = LINE + 2	TR106800	
GO TO 1000	TR106900	
C		
C RAY 1 OR 2 VIGNETTED ON SURFACE NO. SMISS	TR107000	
900 WRITE (6, 32030) RAY, SMISS	TR107200	31
32030 FORMAT(1H0, 3HRAY, 12, 21H VIGNETTED ON SURFACE, 13)	TR107500	
LINE = LINE + 2	TR108000	
1000 IF (RAY .NE. 1) GO TO 1500	TR107000	
C		
C TRACE RAY 2	TR201000	

SUBROUTINE TO COMPUTE TWINRAY S DIAGNOSTIC INFORMATION
TWNRSZ - EPN SOURCE STATEMENT - IPN(S) -

02/16/68

PAGE 3

RAY = 2	TR201500	
CALL RAYTR(D., 1., D., 1, XVCT, OVCT, SMISS, S1200, S1300,	TR202000	
1 3000, 1, XMTRX2, QMTRX2)	TR202100	36
GO TO 1500	TR202500	
C		
RAY 2 MISSED SURFACE NO. SMISS	TR203000	
1200 N2 = SMISS - 1	TR203200	
GO TO 750	TR203500	
C		
RAY 2 REFLECTED ON SURFACE NO. SMISS	TR204000	
1300 N2 = SMISS - 1	TR204200	
GO TO 650	TR204500	
C		
PRINT TWINRAY HEADINGS	TR205000	
1500 WRITE (6, 32040)	TR205200	43
32040 FORMAT(1H0,12HRAY SURFACE, 5X, 1HX, 15X, 1HY, 15X, 1HZ, 14X,	TR205400	
1 2HQX, 14X, 2HQY, 14X, 2HQZ, 11X, 9HTHICKNESS)	TR205500	
LINE = LINE + 2	TR205700	
DO 5000 J = 1, N	TR206000	
I = J - 1	TR206200	
TAU1 = 0.	TR206400	
IF(I .GT. N1) GO TO 475	TR206600	
TAU1 = -XMTRX1(3,J)	TR206800	
IF(I .LT. NSURF) TAU1 = TAU1 + XMTRX1(3,J+1) + SURFC(4,J)	TR207000	
IF(SURFC(3,J) .LT. 0.) TAU1 = -TAU1	TR207200	
475 TAU2 = 0.	TR207400	
IF(I .GT. N2) GO TO 477	TR207600	
TAU2 = -XMTRX2(3,J)	TR207800	
IF(I .LT. NSURF) TAU2 = TAU2 + XMTRX2(3,J+1) + SURFC(4,J)	TR208000	
IF(SURFC(3,J) .LT. 0.) TAU2 = -TAU2	TR208200	
477 CONTINUE	TR208400	
IF(LINE .LE. 52) GO TO 3000	TR301800	
C		
BEGIN NEW PAGE - PRINT TITLE	TR302000	
CALL PRCTL	TR302500	76
WRITE (6, 32040)	TR303000	77
LINE = LINE + 2	TR303200	
C		
PRINT OUT TWINRAY INFORMATION	TR303500	
3000 ITEMP = 1	TR303700	
WRITE (6, 32080)	TR303800	79
32080 FORMAT(1H)	TR303900	
WRITE (6, 32050) ITEMP, I, (XMTRX1(K,J), K=1,3),	TR304000	
1 (QMTRX1(K,J), K=1,3), TAU1	TR304200	80
ITEMP = 2	TR305000	
WRITE (6, 32050) ITEMP, I, (XMTRX2(K,J), K=1,3),	TR305200	
1 (QMTRX2(K,J), K=1,3), TAU2	TR305300	80
32050 FORMAT(1H, 12, 17, 3X, 7E16.7)	TR305500	
LINE = LINE + 3	TR306000	
5000 CONTINUE	TR306500	
C		
END OF TWNRYS SUBROUTINE	TR309000	
RETURN	TR309500	
END	TR309600	

SUBROUTINE TO COMPUTE TWINRAY'S DIAGNOSTIC INFORMATION
TWNRSZ STORAGE MAP

02/16/68

PAGE 4

SUBROUTINE TWNRYS
COMMON VARIABLE 8

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00038	I
NCLRS	00038	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00060	R	FNUMB	00061	R
FLNGH	00062	R	WFLGH	00063	R	ZETA	00064	R
HEKPP	00065	R	DEXPP	00066	R	WEXPP	00067	R
DLPLN	00068	R	OMGAS	00081	R	OMGAS	00082	R
DELD	00083	R	EPRAD	00084	R	PSCAL	00085	R
OMGAP	00086	R	SPFEA	00087	R	DUMIN	00070	R
CDIST	00071	R	HD	00072	R	DELH	00073	R
SYBKH	00074	R	WKDIR	00075	R	WYDIR	00076	R
NDTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
NOBJH	00117	R	WCLRH	00120	R	WINHT	00135	R
EINHT	00144	R	CIMPL	00143	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	08210	R
BNDS	08210	I	NCOND	08665	I			
COMMON BLOCK			AZOBJ	ORIGIN	08667	LENGTH	00003	
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R
COMMON BLOCK			PRNT	ORIGIN	08672	LENGTH	00002	
LINE	00000	I	PAGE	00001	I			
COMMON BLOCK			TMTWIN	ORIGIN	08674	LENGTH	08260	
TEMP8	00000	R	XNTRX1	00000	R	GNTRX1	00454	R
XNTRX2	01130	R	GNTRX2	01604	R			
DIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
XVCT	11154	R	QVCT	11157	R			
UNDIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RAY	11162	I	SMISS	11163	I	I	11164	I
J	11165	I	N1	11166	I	NR	11167	I
N	11170	I	TAU1	11171	R	TAU2	11172	R
ITEMP	11173	I						

SUBROUTINE TO COMPUTE TWINRAY'S DIAGNOSTIC INFORMATION
 TWRBZ

02/18/68

PAGE 9

STORAGE MAP

ENTRY POINTS

TWRBZ SECTION 11

SUBROUTINES CALLED

PRCTYL SECTION 12
 .UNDB. SECTION 13
 SYSLOC SECTION 14

.FWRD. SECTION 15
 .FFIL. SECTION 16

RATTE SECTION 17
 .FCWV. SECTION 18

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
3E000	FORMAT	11210	800	13A	11341	700	29A	11408
800	28A	11428	900	31A	11447	1000	32A	11484
780	28A	11410	3E010	FORMAT	11221	850	29A	11431
120E0	FORMAT	11230	3E0E0	FORMAT	11240	1900	43A	11523
1200	39A	11513	1300	41A	11517	3E040	FORMAT	11250
9000	98A	11761	478	60A	11575	477	72A	11623
3000	78A	11644	3E0E0	FORMAT	11272	3E0E0	FORMAT	11274

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 1E913.

TR-67-700-10-2
Page III-4-1
Program Listing
Link 4

Section 1

LINK 4

TR-67-700-10-2
Page III-4-2
Program Listing
Link 4

INF.M010001.44800-0.77367.AM IC 18JOB LINK1 7094 A 02/16/68 PAGE 6

BTITLE SUBROUTINE TO COMPUTE ENTRANCE PUPIL COORDINATES
SIBPTC LATTZ M94.XR7 LINK 4 (LATT)

LT001000

SUBROUTINE TO COMPUTE ENTRANCE PUPIL COORDINATES
 LATTZZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 1

```

SUBROUTINE LATT( XOHAT, YOHAT, MRAYS )                                LT001200
C
C *** INPUTS ***                                                    LT001300
C LATTC MATRIX SPECIFYING ENTRANCE PUPIL COORDINATES. (DATA COMMON) LT001310
C MRAYS RAY NUMBER BEING GENERATED.                                LT001320
C
C *** OUTPUTS ***                                                  LT001600
C XOHAT X COORDINATE OF CURRENT RAY.                                LT001610
C YOHAT Y COORDINATE OF CURRENT RAY.                                LT001620
C
C *** DESCRIPTION ***                                              LT001700
C THIS SUBROUTINE COMPUTES THE ENTRANCE PUPIL COORDINATES OF ONE LT001710
C RAY PER CALL. THE PROPER USAGE TO GENERATE THE ENTIRE PATTERN LT001720
C IS SHOWN BELOW.                                                 LT001730
C
C MRAYS = 0                                                         LT003000
C RCNT = 0                                                         LT003100
C DO 20 I = 1, MRAYS                                               LT003200
C IF( LMODE .EQ. 0 ) RCNT = 1                                       LT003300
C . . . . .                                                         LT003400
C CALL LATT( XOHAT, YOHAT, MRAYS )                                  LT003500
C 10 . . . . .                                                         LT003600
C CALL RAYTR( XOHAT, YOHAT, ... )                                  LT003700
C . . . . .                                                         LT003800
C IF( RCNT .EQ. 0 ) GO TO 20                                         LT003900
C RCNT = 0                                                         LT004000
C XOHAT = - XOHAT                                                  LT004100
C GO TO 10                                                         LT004200
C 20 CONTINUE                                                       LT004300
C
COMMON / DATA / DUMMY(27), DATA(3483)                            LT005000
EQUIVALENCE ( DATA(8), LATTC(1,1), ILATTC(1,1) ), ( DATA(14), LT005100
1 DELY )                                                            LT005150
DIMENSION LATTC(3,26), ILATTC(3,26)                                LT005200
REAL LATTC                                                          LT005300
C
INTEGER SLICE                                                       LT005400
C
C START OF LATT SUBROUTINE                                          LT005500
C
IF( MRAYS .NE. 0 ) GO TO 1000                                       LT006000
C
C FIRST RAY OF PATTERN                                             LT006200
SLICE = 1                                                            LT006400
C
C BEGIN NEW SLICE                                                  LT006600
500 NSLR = 0                                                         LT006800
XOHAT = LATTC(1,SLICE)                                              LT007000
YOHAT = LATTC(2,SLICE)                                              LT007200
GO TO 2000                                                           LT007400
1000 IF( NSLR .GE. ILATTC(3,SLICE) ) GO TO 1500                    LT008000
C
C GENERATE NEXT RAY OF CURRENT SLICE                                LT008100
XOHAT = LATTC(1,SLICE)                                              LT008200
YOHAT = FLOAY( NSLR ) * DELY + LATTC(2,SLICE)                       LT008400

```

SUBROUTINE TO COMPUTE ENTRANCE PUPIL COORDINATES
LATTZZ - EPN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 2

GO TO 2000	LT000800
C	
C	BEGIN RAY GENERATION ON NEXT SLICE
1900 SLICE = SLICE + 1	LT000900
GO TO 500	LT000910
	LT000920
C	
C	INCREMENT RAY COUNTERS
2000 NSLR = NSLR + 1	LT000930
MRAYS = MRAYS + 1	LT000940
	LT000950
C	
C	END OF LATT SUBROUTINE
RETURN	LT000960
END	LT000970
	LT000980

SUBROUTINE TO COMPUTE ENTRANCE PUPIL COORDINATES
 LATTZZ STORAGE MAP

02/16/68

PAGE 3

SUBROUTINE LATT
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08668		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
DUMMY	00000	R	DATA	00093	R	LATTC	00170	R	
LATTC	00170	I	DELY	00090	R				

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SLICE	08667	I	NSLR	08670	I			

ENTRY POINTS

LATT SECTION 9

SUBROUTINES CALLED

E.1 SECTION 6
 E.4 SECTION 9
 CC.3 SECTION 12

E.2 SECTION 7
 CC.1 SECTION 10
 CC.4 SECTION 13
 EFN IFN CORRESPONDENCE

E.3 SECTION 8
 CC.2 SECTION 11
 SYSLOC SECTION 14

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	10A	08716	500	5A	08705	2000	19A	08753
1500	17A	08747						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07012.

TR-67-700-10-2
Page III-4-6
Program Listing
Link 4

INF.N510001.44800-0.77387.AM IC

ISJOB L10MR1 7094 A 02/18/68

PAGE 4

00
STITLE SUBROUTINE TO COMPUTE BACK FOCUS AND F/ NUMBER
SIBP TC MERIDZ M94,MM7 LINK 4 (MERID)

NR000100

SUBROUTINE TO COMPUTE BACK FOCUS AND F/ NUMBER
 MERIDZ - EPN SOURCE STATEMENT - IPN(S) -

02/18/68

PAGE 1

```

SUBROUTINE MERID( COLOR, BFOCUS, FSNUMB )
C
C   *** INPUTS ***
C   COLOR NUMBER OF COLOR TO BE USED.
C   ONSA1 MERIDIONAL SCALE FACTOR FOR BACK FOCUS.      (DATA COMMON)
C
C   *** OUTPUTS ***
C   BFOCUS BACK FOCUS VALUE.
C   FSNUMB F/ NUMBER.
C
C   *** DESCRIPTION ***
C   THIS SUBROUTINE COMPUTES THE BACK FOCUS AND F/ NUMBER FOR THE
C   PROVIDED COLOR NUMBER. IF THEY ARE INFINITE, THEN BFOCUS AND
C   FSNUMB ARE SET TO ZERO.
C
COMMON / DATA / DUMMY1(27), DATA(3483)
EQUIVALENCE ( DATA(24), ONSA1 )
C
COMMON/AZOBJ/TNTR,SNTHTR,CSTHTR
C
INTEGER COLOR
DIMENSION XVCT(3), QVCT(3)
C
C           START OF MERID SUBROUTINE
C
BFOCUS = 0.0
FSNUMB = 0.0
CALL RAYTR( D., ONSA1, D., COLOR, XVCT, QVCT, SMISS, $400, $500,
1 $600, 0, DUMMY, DUMMY )
GO TO 600
400 WRITE ( 6, 32000 ) SMISS
32000 FORMAT( 1H0, 29MERIDIONAL RAY MISSED SURFACE, I4 )
CALL GOOP
500 WRITE ( 6, 32010 ) SMISS
32010 FORMAT( 1H0, 35MERIDIONAL RAY REFLECTED AT SURFACE, I4 )
CALL GOOP
600 IF( QVCT(2) .EQ. 0.0 ) GO TO 1000
ALPHA = - QVCT(3) / QVCT(2)
BFOCUS = ALPHA * XVCT(2) + XVCT(3)
FSNUMB = .5 * ABS( ALPHA * ONSA1 )
C
1000 RETURN
END

```

NR000200
 NR000300
 NR000400
 NR000700
 NR000800
 NR000900
 NR001000
 NR001100
 NR001200
 NR001300
 NR001400
 NR001600
 NR001700
 AZ004000
 NR001800
 NR001900
 NR002400
 NR002600
 NR002800
 NR003000
 NR003200
 NR003400
 NR004000
 NR004200
 NR004400
 NR005000
 NR005200
 NR005400
 NR006000
 NR006200
 NR006400
 NR006600
 NR009000
 NR009200

2
3
6
7
8

SUBROUTINE TO COMPUTE BACK FOCUS AND F/ NUMBER
 MERIDZ

STORAGE MAP

02/16/68

PAGE 2

SUBROUTINE MERID
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	07001	LENGTH	08666		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
DUMMY1	0000	R	DATA	0003	R	OMGA1	0002	R	
COMMON BLOCK			AZOBJ	ORIGIN	08667	LENGTH	00003		
THTR	0000	R	SNTHTR	0001	R	CSTHTR	0002	R	
DIMENSIONED PROGRAM VARIABLES									
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
KVCT	08672	R	QVCT	08673	R				
UNDIMENSIONED PROGRAM VARIABLES									
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
SN188	08700	R	DUMMY	08701	R	ALPHA	08702	R	

ENTRY POINTS

MERID SECTION 7

SUBROUTINES CALLED

RAYTR SECTION 8
 .UNDB. SECTION 11
 SYSLOC SECTION 14

.FWRD. SECTION 9
 .FFIL. SECTION 12

GOOF SECTION 10
 .FCNV. SECTION 13

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
400	5A	08760	500	7A	08773	600	9A	07008
32000	FORMAT	08714	32010	FORMAT	08724	1000	12A	07033

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07080.

TR-67-700-10-2
Page III-4-9
Program Listing
Link 4
Page Revised 10/1/63

INF,ND10001,44953-0,77387,AM IC

IBJOB L10MKC 7094 A 09/20/68

PAGE 3

84
TITLE SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION
SICFIC SMFOF Z M94,XR7 LINK 4 (SMFOF)

SM001000

SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION
SMPOFZ - EFN SOURCE STATEMENT - IFN(5) -

09/29/68

PAGE 3

```

SUBROUTINE SMPOF( COLOR, HNUMB, HEIGHT, BFL, CMFLAG, TEMPS )      SM001200
C
C   *** INPUTS ***
C   COLOR NUMBER OF COLOR TO BE USED.                            SM001500
C   HNUMB OBJECT POINT INDEX TO BE USED.                          SM001510
C   HEIGHT OBJECT HEIGHT.                                         SM001520
C   BFL COMPUTED BACK FOCUS FOR COLOR NUMBER 1.                   SM001530
C   CMFLAG FLAG FOR TYPE OF COMPUTATIONS TO BE PERFORMED AS FOLLOWS - SM001540
C       1) OPTICS DIAGNOSTIC                                       SM001550
C       2) DESIGN                                                    SM001560
C       3) PLOT                                                       SM001570
C       4) SENSITIVITY                                               SM001580
C
C   *** OUTPUTS ***
C   NMIS NUMBER OF RAYS WHICH MISSED. (TMPATT COMMON)            SM001600
C   NREFLCT NUMBER OF RAYS WHICH REFLECTED. (TMPATT COMMON)      SM001610
C   NVIGN NUMBER OF RAYS WHICH VIGNETTED. (TMPATT COMMON)        SM001620
C   NRSUC NUMBER OF RAYS WHICH ARE SUCCESSFUL. (TMPATT COMMON)    SM001630
C   XSJSTAR POSITION OF PRINCIPAL IMAGE PLANE. (TMPATT COMMON)     SM001640
C   XDAR AVERAGE X (TMPATT COMMON)                               SM001650
C   YDAR AVERAGE Y (TMPATT COMMON)                               SM001660
C   XSTAR RMS X. (TMPATT COMMON)                                 SM001670
C   YSTAR RMS Y. (TMPATT COMMON)                                 SM001680
C   RSTAR SPOT SIZE. (TMPATT COMMON)                             SM001690
C   XCURL VECTOR OF X COORDINATES OF RAYS ON IMAGE PLANE. (TMPATT COMMON) SM001700
C   YCURL VECTOR OF Y COORDINATES OF RAYS ON IMAGE PLANE. (TMPATT COMMON) SM001710
C
C   *** DESCRIPTION ***
C   THIS SUBROUTINE TRACES THE SPECIFIED RAY PATTERN AND COMPUTES SM001800
C   VARIOUS PARAMETERS DEPENDENT UPON THE SETTING OF CMFLAG.     SM001810
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3486) DT001100
2 DIMENSION WDBJH(7), WCLR(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
3 WCLRS(6), NDSGN(4), NDSGV(11), LATTC(3,26), BOUNDS(3,100), DT002100
4 SURFC(20,100), DESGN(10,50), SUBST(500), ILATTC(3,26), DT002200
5 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
6 EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3), DT101000
7 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
8 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9), DT101200
9 LMODE ), ( DATA(10), NSFLN ), ( DATA(11), WDBJH ), ( DATA(12), DT101300
10 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
11 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18), DT101500
12 ZETA ), ( DATA(19), HEXPF ), ( DATA(20), DEXPF ), ( DATA(21), DT101600
13 WEXPF ), ( DATA(22), DLPLN ), ( DATA(23), OMGA2 ), ( DATA(24), DT101700
14 OMGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
15 FSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
16 DUMIN ), ( DATA(31), OD1ST ), ( DATA(32), HD ), ( DATA(33), DT102000
17 DELH ), ( DATA(34), SYSMX ), ( DATA(35), WXD1R ), ( DATA(36), DT102100
18 WYDIR ), ( DATA(37), RDTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
19 NDSGV(1) ), ( DATA(53), WDBJH(1) ), ( DATA(60), WCLR(1) ), DT102300
20 E ( DATA(67), WIMHT(1) ), ( DATA(74), EIMHT(1) ), ( DATA(81), DT102400
21 CIMPL(1) ), ( DATA(88), WCLRS(1) ), DT102500
22 F ( DATA(94), LATTC(1,1), ILATTC(1,1) ), ( DATA(102), SURFC(1,1), DT102600
23 H ISURFC(1,1) ), ( DATA(2182), DESGN(1,1), IDESGN(1,1) ), DT102700

```

SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION
SMFOPZ - EFN SOURCE STATEMENT - IFN(S) -

09/20/68

PAGE 2

```

I ( DATA(2682), SUBST(1), ISUBST(1) ), ( DATA(3182), BOUNDS(1,1), DT102800
J BOUNDS(1,1) ), ( DATA(3483), NCOND ) DT102900
EQUIVALENCE ( DATA(3484), ATRGCR ), ( DATA(3485), GAUSS ) DT103000
INTEGER CONTRL, AFLAG, DATE, ATRGCR DT201000
REAL LATTC DT202000
C
COMMON/AZOBJ/THTR,SNTHTR,CSTHTR A2004000
C
COMMON / TMPATT / NMISS, NREFLCT, NVIGN, NRAYS, XBAR, YBAR, XSTAR, L3001000
1 YSTAR, RSTAR, SJSTAR, XCURL(200), YCURL(200), MXBAR, KXBAR, L3001200
2 MYBAR, KYBAR, AX, BX, CX, AY, BY, CY, ACAF, BCAP, CCAF L3001400
DIMENSION MXCURL(200), MYCURL(200), L3001600
DIMENSION TLINK(425) L3001700
EQUIVALENCE ( XCURL(1), MXCURL(1) ), ( YCURL(1), MYCURL(1) ) L3001800
EQUIVALENCE ( NMISS, TLINK(1) ) L3001900
REAL MXBAR, KXBAR, MYBAR, KYBAR, MXCURL, MYCURL L5002000
C
INTEGER COLOR, HNUMB, CMFLAG, RCNT, SMISS, RYCNT SM002500
C
DIMENSION TEMPS(1), XVCT(3), QVCT(3) SM003000
C
START OF SMFOP SUBROUTINE SM005000
C
ZERO OUT LINK3 COMMON BLOCK SM005500
DO 1000 I = 1, 425 SM005600
1000 TLINK(I) = 0.0 SM005800
SJSTAR = DELD + CIMPL( HNUMB ) SM006000
IF( LMODE .EQ. 0 ) SJSTAR = SJSTAR + DELD SM006200
NRAYS = 0 SM006500
RCNT = 0 SM007000
MR = NRAYS SM007200
IF( LMODE .EQ. 0 ) MR = MR + MR SM007400
IF( CMFLAG .NE. 3 ) GO TO 1300 SM007600
RYCNT = 0 SM007800
DO 1200 I = 1, 200 SM008000
1200 TEMPS(I) = 0. SM008200
C
LOOP TO GENERATE AND TRACE EACH RAY FOR COLOR = COLOR AND SM101000
OBJECT HEIGHT = HNUMB SM101100
1300 DO 8000 I = 1, NRAYS SM101500
C SET FLAG TO INDICATE RAY ( -XGHAT, YGHAT ) IS TO BE TRACED SM101800
IF( LMODE .EQ. 0 ) RCNT = 1 SM102000
C GENERATE CURRENT RAY SM102500
CALL LATT( XGHAT, YGHAT, MRAYS ) SM102700
C
TRACE CURRENT RAY SM103000
1400 CALL RAYTR( XGHAT, YGHAT, HEIGHT, COLOR, XVCT, QVCT, SMISS, SM103200
1 32000, 33000, 34000, 0, DUMMY, DUMMY ) SM103400
36
1500 IF( QVCT(3) .NE. 0.0 ) GO TO 5000 SM103600
GO TO 3000 SM103800
C RAY HAS MISSED SURFACE SM104000
2000 NMISS = NMISS + 1 SM104200
IF( CMFLAG .EQ. 3 ) TEMPS( RYCNT+1 ) = 1. SM104300
GO TO 7000 SM104400
C RAY HAS BEEN REFLECTED SM105000
3000 NREFLCT = NREFLCT + 1 SM105200

```

SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION
SMPOP2 - EFN SOURCE STATEMENT - IFN(5) -

09/20/68

PAGE 3

IF(CMFLAG .EQ. 3) TEMPS(RYCNT+1) = 2.	SM105300
GO TO 7000	SM105400
C RAY HAS VIGNETTED	SM106000
4000 NVIGN = NVIGN + 1	SM106200
IF(CMFLAG .EQ. 3) TEMPS(RYCNT+1) = 3.	SM106300
IF (CMFLAG .EQ. 2 .OR. CMFLAG .EQ. 5) GO TO 1500	SM106500 700000
GO TO 7000	SM106700
5000 NRAYSB = NRAYSB + 1	SM107000
IF(CMFLAG .GT. 1 .AND. CMFLAG .LT. 5) GO TO 6000	SM107200
C	
C COMPUTE SPECIAL VALUES USED IN OPTIC DIAGNOSTICS	SM108000
XCURL(NRAYSB) = QVCT(1) / QVCT(3)	SM108200
YCURL(NRAYSB) = QVCT(2) / QVCT(3)	SM108400
TEMPS(NRAYSB) = XVCT(1) - XCURL(NRAYSB) * XVCT(3)	SM108600
TEMPS(NRAYSB+200) = XVCT(2) - YCURL(NRAYSB) * XVCT(3)	SM108800
GO TO 7000	SM109000
6000 SL = (SJSTAR - XVCT(3)) / QVCT(3)	SM201000
XCURL(NRAYSB) = QVCT(1) * SL + XVCT(1)	SM201200
YCURL(NRAYSB) = QVCT(2) * SL + XVCT(2)	SM201400
7000 IF(CMFLAG .EQ. 3) RYCNT = RYCNT + 1	SM201800
IF(RCNT .EQ. 0) GO TO 8000	SM202000
C	
C TRACE MIRRORED RAY	SM202200
RCNT = 0	SM202400
XDHAT = - XDHAT	SM202600
GO TO 1400	SM203000
8000 CONTINUE	SM203200
C	
C TEST FOR ANY SUCESSFUL RAYS	SM203500
IF(NRAYSB .EQ. 0) GO TO 32000	SM203800
XNR = NRAYSB	SM204000
IF (CMFLAG .GT. 1 .AND. CMFLAG .LT. 5) GO TO 15000	SM204200
DO 9000 J = 1, NRAYSB	SM204500
MYBAR = MYBAR + YCURL(J)	SM204600
9000 KYBAR = KYBAR + TEMPS(J+200)	SM204700
MYBAR = MYBAR / XNR	SM205000
KYBAR = KYBAR / XNR	SM205200
DO 10000 K = 1, NRAYSB	SM205500
YCURL(K) = YCURL(K) - MYBAR	SM205600
10000 TEMPS(K+200) = TEMPS(K+200) - KYBAR	SM205700
IF(LMODE .EQ. 1) GO TO 13000	SM206000
DO 11000 L = 1, NRAYSB	SM206500
MXBAR = MXBAR + XCURL(L)	SM206600
11000 KXBAR = KXBAR + TEMPS(L)	SM206700
MXBAR = MXBAR / XNR	SM207000
KXBAR = KXBAR / XNR	SM207200
DO 12000 I = 1, NRAYSB	SM207500
XCURL(I) = XCURL(I) - MXBAR	SM207600
12000 TEMPS(I) = TEMPS(I) - KXBAR	SM207700
13000 IF (CMFLAG .EQ. 5) RETURN	SM207800
DO 14000 J = 1, NRAYSB	SM208000
AX = AX + XCURL(J) * XCURL(J)	SM208100
BX = BX + XCURL(J) * TEMPS(J)	SM208200
CX = CX + TEMPS(J) * TEMPS(J)	SM208300
AY = AY + YCURL(J) * YCURL(J)	SM208500
BY = BY + YCURL(J) * TEMPS(J+200)	SM208600

SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION
 SMPOPZ - EFN SOURCE STATEMENT - IFN(S) -

09/20/68

PAGE

14000	CY = CY + TEMPS(J*200) * TEMPS(J*200)	SM208700	
	AX = AX / XNR	SM209000	
	BX = BX / XNR	SM209100	
	CX = CX / XNR	SM209200	
	AY = AY / XNR	SM209300	
	BY = BY / XNR	SM209400	
	CY = CY / XNR	SM209500	
	ACAP = AX + AY	SM209600	
	BCAP = BX + BY	SM209700	
	CCAP = CX + CY	SM209800	
	GO TO 32000	SM209900	
C			
15000	DO 15500 K = 1, NRAYSB	SM301000	
15500	YBAR = YBAR + YCURL(K)	SM301200	
	YBAR = YBAR / XNR	SM301400	
	DO 16000 L = 1, NRAYSB	SM302000	
16000	YCURL(L) = YCURL(L) - YBAR	SM302200	
	IF (LMODE .EQ. 1) GO TO 18000	SM302400	
	DO 16500 I = 1, NRAYSB	SM303000	
16500	XBAR = XBAR + XCU(L(I))	SM303200	
	XBAR = XBAR / XNR	SM303400	
	DO 17000 J = 1, NRAYSB	SM304000	
17000	XCURL(J) = XCURL(J) - XBAR	SM304200	
18000	IF (CMFLAG .LT. 4) GO TO 32000	SM304500	
	DO 19000 K = 1, NRAYSB	SM305000	
	XSTAR = XSTAR + XCURL(K) * XCURL(K)	SM305200	
19000	YSTAR = YSTAR + YCURL(K) * YCURL(K)	SM305400	
	XSTAR = SQRT(XSTAR / XNR)	SM306000	203
	YSTAR = SQRT(YSTAR / XNR)	SM306200	204
	RSTAR = SQRT(XSTAR * XSTAR + YSTAR * YSTAR)	SM306400	
C			
C	END OF SMPOP SUBROUTINE	SM901000	205
32000	RETURN	SM902000	
	END	SM903000	

SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION
SMPOPZ

STORAGE MAP

09/20/68

PAGE 5

SUBROUTINE SMOF
COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	FUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIFLN	00042	I	IMODE	00043	I
NSFLN	00044	I	NODJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXFF	00055	R	DEXFF	00056	R	WEXFF	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SFFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HO	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDEGV	00104	I
WOBJH	00117	R	WCLRH	00126	R	WIMHT	00130	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IDNDS	06210	I	NCOND	06665	I	ATRCR	06666	I
GAUSS	06667	R						

COMMON BLOCK			AZOBJ	ORIGIN	06672	LENGTH	00003	
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R

COMMON BLOCK			THPATT	ORIGIN	06675	LENGTH	06651	
NMISS	00000	I	NRFICT	00001	I	NVIGN	00002	I
NRAYSB	00003	I	XBAR	00004	R	YBAR	00005	R
XSTAR	00006	R	YSTAR	00007	R	RSTAR	00010	R
SJS' R	00011	R	XCURL	00012	R	YCURL	00023	R
KXBAR	00634	R	KXBAR	00635	R	MYBAR	00636	R
KYBAR	00637	R	AX	00640	R	BX	00641	R
CX	00642	R	AY	00643	R	BY	00644	R
CY	00645	R	ACAP	00646	R	BCAP	00647	R
CCAP	00650	R	MXCURL	00012	R	MYCURL	00323	R
TLINK	00000	R						

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
XVCT	07546	R	QVCT	07551	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
--------	----------	------	--------	----------	------	--------	----------	------

SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION
 SMPDF 2

09/20/68

PAGE 6

NAME	ADDRESS	MODE	NAME	ADDRESS	MODE	NAME	ADDRESS	MODE
PCNT	07554	I	SMISS	07555	I	RYCNT	07556	I
MRAYS	07557	I	MR	07560	I	XDHAT	07561	R
YDHAT	07562	R	DUMMY	07563	R	SL	07564	R
XDR	07565	R						

ENTRY POINTS

SMPDF SECTION 9

SUBROUTINES CALLED

NAME	SECTION	ADDRESS	NAME	SECTION	ADDRESS	NAME	SECTION	ADDRESS
LATT	SECTION	10	RAYTR	SECTION	11	SORT	SECTION	12
E.1	SECTION	13	E.2	SECTION	14	E.3	SECTION	15
E.4	SECTION	16	CC.1	SECTION	17	CC.2	SECTION	18
CC.3	SECTION	19	CC.4	SECTION	20	BYSLOC	SECTION	21

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	5A	07604	1300	28A	07644	1200	24A	07641
2000	86A	10075	1400	35A	07663	2000	42A	07710
3000	48A	07724	4000	54A	07740	1500	38A	07703
5000	63A	07763	7000	78A	10057	6000	75A	10035
32000	206A	10457	15000	156A	10322	9000	99A	10133
10000	108A	10155	13000	134A	10224	11000	120A	10175
12000	129A	10217	14000	150A	10260	15500	160A	10326
16000	167A	10342	18000	190A	10400	16500	178A	10357
17000	185A	10373	19000	198A	10415			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 10551.

TR-67-700-10-2
Page III-5-1
Program Listing
Link 5

Section 5

LINK 5

TR-67-700-10-2
Page III-5-2
Program Listing
Link 5

1739 INF .ND10001.44800-0.77387.AM IC1.8.8000 WILSON - INFORMATICS A 17-28-59 02/16/68 *****LINK1:
JOB NO60
BT
BTITLE SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN
BTBFC BNOCHZ M94.XR7 LINK 5 (BNOCHK) BC001000

SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN
 BNDCHK - EFN SOURCE STATEMENT - IPN/C.

02/16/68

PAGE 1

104*****

```

SUBROUTINE BNDCHK ( *, IFLAG )                                BC001500
C
C *** INPUTS ***                                           BC001600
C IFLAG FLAG TO SUPPRESS PRINTOUT WHEN SET NON-ZERO.      BC001610
C BOUNDS MATRIX CONTAINING BOUNDARY INFORMATION.          (DATA COMMON) BC001620
C
C *** OUTPUTS ***                                           BC001630
C * ALTERNATE RETURN USED IF A BOUNDARY VIOLATION OCCURS. BC001640
C
C *** DESCRIPTION ***                                       BC001650
C BNDCHK IS USED TO CHECK THE IMPROVED LENS SYSTEM FOR BOUNDARY BC001660
C VIOLATIONS USING THE PARAMETERS IN THE CURRENT DESIGN VARIABLE. BC001670
C TESTS ARE MADE ON THE MATRIX, BOUNDS, IF NCOND .GT. 0 AND BC001680
C OTHER PARAMETERS. THE ALTERNATE RETURN IS TAKEN IF A VIOLATION BC001690
C IS DETECTED.                                             BC001700
C
COMMON / DATA / NCNTRL, CONTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483)                                             DT001100
DIMENSION WOBJH(7), WCLRH(7), WINHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3), DT101000
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9), UT101200
3 IMODE ), ( DATA(10), NSPLN ), ( DATA(11), NOBJH ), ( DATA(12), DT101300
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
5 PNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18), DT101500
6 ZETA ), ( DATA(19), HEXPP ), ( DATA(20), DEXPP ), ( DATA(21), DT101600
7 WEXPP ), ( DATA(22), OLPLN ), ( DATA(23), OMGA2 ), ( DATA(24), DT101700
8 OMGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
9 PSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
A DUMIN ), ( DATA(31), ODIST ), ( DATA(32), HD ), ( DATA(33), DT102000
B DELH ), ( DATA(34), SYMX ), ( DATA(35), WXDIR ), ( DATA(36), DT102100
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
D NDSGV(1) ), ( DATA(53), WOBJH(1) ), ( DATA(60), WCLRH(1) ), DT102300
E ( DATA(67), WINHT(1) ), ( DATA(74), EIMHT(1) ), ( DATA(81), DT102400
F CIMPL(1) ), ( DATA(88), WCLRS(1) ), DT102500
G ( DATA(94), LATT(1,1), ILATT(1,1) ), ( DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1) ), ( DATA(2102), DESGN(1,1), IDESGN(1,1) ), DT102700
I ( DATA(2602), SUBST(1), ISUBST(1) ), ( DATA(3102), BOUNDS(1,1), DT102800
J IBNDS(1,1) ), ( DATA(3483), NCOND ) DT102900
INTEGER CONTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
COMMON/PRNT/LINE,PAGE PT002500
INTEGER PAGE PT002700
C
COMMON/AZOBJ/THTR,SNTHTR,CSTHTR AZ004000
C
COMMON / INDESN / PERIOD, PCOUNT, NOFFND, JAIL(10,10), SENTN(10), L7001000
1 DIMVCT, CORD(6), SIZE, DETRN, EDLNG, ESLNG, IMPRV, VCTSTR(6000), L7002000
2 VINDX(11) L7002100
REAL IMPRV L7003000
INTEGER PERIOD, PCOUNT, SENTN, DIMVCT, CORD, VINDX L7004000

```

SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN
 BNDCHK - EFN SOURCE STATEMENT - IFN(S) -

02/16/68

PAGE 2

```

C
EQUIVALENCE ( NVARX, VCTSTR(6000)), ( JI, VCTSTR(5999)), ( NJ, VCTSTR(5998)), ( NJA, VCTSTR(5997)), ( KMA, VCTSTR(5996)), ( U, VCTSTR(5995)), ( RL, VCTSTR(5994)), ( RLA, VCTSTR(5993)), ( UMIN, VCTSTR(5992)), ( UMAX, VCTSTR(5991)), ( ICOLOR(1), VCTSTR(5990)), ( QMBC(1), VCTSTR(5989)), ( QMBC(2), VCTSTR(5988)), ( QMAX, VCTSTR(5987)), ( INDX, VCTSTR(5986)), ( DDIST(1), VCTSTR(5985)), ( ALENG, VCTSTR(5984)), ( XMTRX(1,1), VCTSTR(5983)), ( QMTRX(1,1), VCTSTR(5982)), ( YOHAT(1), VCTSTR(5981)), ( HTAB(1), VCTSTR(5980)), ( SMISS, VCTSTR(5979)), ( XVCT(1), VCTSTR(5978)), ( QVCT(1), VCTSTR(5977)), ( TAU, VCTSTR(5976)), ( NUPPER, VCTSTR(5975)), ( KIND, VCTSTR(5974))
BC002000
BC002100
BC002200
BC002300
BC002400
BC002500
BC002600
BC002700
BC002800
BC002900
BC003000
BC003100
BC003200
BC003300
BC003400
BC003500
BC003600
BC003700
BC003800
BC003900
BC004000

C
INTEGER SMISS, RL, RLA
DIMENSION ICOLOR(6), DDIST(100), XMTRX(3,100), QMTRX(3,100),
1 YOHAT(4), HTAB(4), XVCT(3), QVCT(3)
BC005000
BC006000
BC006200

C
C
C
C
START OF BNDCHK SUBROUTINE
BC101000

100 IF ( NCOND .EQ. 0 ) GO TO 3000
NVARX = NMSGV(1)
DO 3000 I = 1, NVARX
JI = NMSGV(I+1)
NJ = IOESGN(1,JI)
NJA = IABS(NJ)
DO 3000 M = 1, NJA
KMA = IABS( IOESGN(M+3,JI) )
U = DATA(KMA)
DO 2500 L = 1, NCOND
RL = IBNDS( 1,L )
RLA = IABS(RL)
UMIN = BOUNDS( 2,L )
UMAX = BOUNDS( 3,L )
IF ( RL .LT. 0 ) GO TO 1000
IF ( KMA .NE. RL ) GO TO 2500
IF ( NJ .LT. 0 ) GO TO 500
IF ( UMIN .LE. U .AND. U .LE. UMAX ) GO TO 3000
GO TO 3500
500 IF ( UMIN .LE. (1./U) .AND. (1./U) .LE. UMAX ) GO TO 3000
GO TO 3500

1000 IF ( KMA .LT. RLA .OR. KMA .GT. (RLA + NC - 1) ) GO TO 2500
CALL CORDER ( NCLRS, DATA(RLA), ICOLOR )
DO 1500 J = 1, NCLRS
IF ( CORD(J) .NE. ICOLOR(J) ) GO TO 3500
1500 CONTINUE
QMIN = DATA(RLA)
QMAX = QMIN
IF ( NCLRS .EQ. 1 ) GO TO 2200
DO 2000 J = 2, NCLRS
JNDX = RLA + J - 1
QMIN = AMIN1( QMIN, DATA(JNDX) )
2000 QMAX = AMAX1( QMAX, DATA(JNDX) )
2200 IF ( QMIN .LT. UMIN ) GO TO 3500
IF ( QMAX .GT. UMAX ) GO TO 3500
GO TO 3000
2500 CONTINUE
BC101200
BC101400
BC101600
BC101800
BC102000
BC102200
BC102400
BC102600
BC102800
BC103000
BC103200
BC103400
BC103600
BC103800
BC104000
BC104200
BC104400
BC104600
BC104800
BC105000
BC105200

BC106000
BC106200
BC106400
BC106600
BC106800
BC107000
BC107200
BC107400
BC107600
BC107800
BC108000
BC108200
BC108400
BC108600
BC108800
BC109000

```

49

SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN
BNDCHZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 3

3000 CONTINUE	BC109200	
GO TO 3000	BC109400	
C		
C BOUNDARY CONDITION HAS BEEN VIOLATED	BC201000	
3500 IF (IFLAG .NE. 0) GO TO 4000	BC201200	
CALL DPRNT	BC201300	99
WRITE (6, 32010) JI, N, U	BC201400	08
32010 FORMAT(1HD, 9X, 15HDESIGN VARIABLE, 13, 10H PARAMETER, 12, 32H VIOLATED A BOUNDARY CONDITION (, E14.7, 1H))	BC201800	
LINE = LINE + 2	BC202000	
C		
4000 CALL PRMSUB(34000)	BC202200	88
RETURN 1	BC202400	
5000 IF (SYSMX .EQ. 0.) GO TO 6000	BC202600	
DDIST(1) = SURFC(4,1)	BC202800	
QMIN = DDIST(1)	BC203000	
QMAX = QMIN	BC203200	
IF (NSURF .EQ. 1) GO TO 6000	BC203400	
DO 5500 I = 2, NSURF	BC203600	
DDIST(I) = DDIST(I-1) + SURFC(4,I)	BC203800	
QMIN = AMIN1(QMIN, DDIST(I))	BC204000	
5500 QMAX = AMAX1(QMAX, DDIST(I))	BC204200	
ALENG = ABS(QMAX - QMIN)	BC205000	
IF (ALENG .LE. SYSMX) GO TO 6000	BC205200	
C		
C MAXIMUM LENGTH EXCEEDED	BC205400	
IF (IFLAG .NE. 0) GO TO 4000	BC205600	
CALL DPRNT	BC205700	114
WRITE (6, 32020) ALENG	BC205800	115
32020 FORMAT(1HD, 9X, 18HTOTAL LENS LENGTH=, E14.7, 29H VIOLATES BOUNDARY CONDITION)	BC206000	
LINE = LINE + 2	BC206200	
GO TO 4000	BC206400	
6000 THTR = 0.	BC206600	
SMTNTR = 0.	BC206800	
CSTNTR = 1.	BC207000	
YOHAT(1) = - OHGAF	BC207200	
YOHAT(2) = OHGAF	BC207400	
YOHAT(3) = OHGAF	BC207600	
YOHAT(4) = OHGAF	BC207800	
HTAB(1) = HO + FLOAT(NOBJN-1) * DELH	BC208000	
HTAB(2) = HTAB(1)	BC208200	
HTAB(3) = 0.	BC208400	
HTAB(4) = 0.	BC208600	
NUPPER = NSURF - 1	BC208800	
IF (NSURF .LE. 1) GO TO 7000	BC209000	
DO 7000 I = 1, 4	BC209200	
CALL RAYTR(0., YOHAT(I), HTAB(I), 1, KVCT, QVCT, MISS, 3000, 1 30500, 30500, 1, XMTRX, QMTRX)	BC301000	
	BC301200	
	BC301400	127
	BC301600	
6500 DO 7000 J = 1, NUPPER	BC301800	
TAU = XMTRX(3, J+2) - XMTRX(3, J+1) + SURFC(4, J+1)	BC302000	
IF (SURFC(9, J+1) .LT. 0.) TAU = - TAU	BC302200	
IF (TAU .GT. SPFEA) GO TO 7000	BC302400	
KIND = 3	BC302600	
GO TO 9000	BC302800	
7000 CONTINUE	BC302800	

SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN
BNDCHZ - EFM SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 4

7500 CALL PRMSUB(8100)	BC303000	147
RETURN	BC303200	
8000 KIND = 1	BC303400	
GO TO 9000	BC303600	
8500 KIND = 2	BC303800	
9000 IF (IFLAG .NE. 0) GO TO 4000	BC304000	
CALL DPRNT	BC304100	156
GO TO (9200, 9400, 9600), KIND	BC304200	
9200 WRITE (6, 32030) I, SMISS	BC304400	150
32030 FORMAT(1H0, 9X, 17HFEATHER CHECK RAY, 12, 13H MISSED SURFACE, 13)	BC304600	
LINE = LINE + 2	BC304800	
GO TO 4000	BC305000	
9400 WRITE (6, 32040) I, SMISS	BC305200	160
32040 FORMAT(1H0, 9X, 17HFEATHER CHECK RAY, 12, 21H REFLECTED AT SURFACE	BC305400	
1E, 13)	BC305600	
LINE = LINE + 2	BC305800	
GO TO 4000	BC306000	
9600 K = J + 1	BC307000	
WRITE (6, 32050) J, K, TAU, I	BC307200	
32050 FORMAT(1H0, 9X, 13HLENS SURFACES, 13, 4H AND, 13, 22H FEATHERED	BC307400	163
1THICKNESS=, E14.7, 6H) RAY, 12)	BC307600	
LINE = LINE + 2	BC307800	
GO TO 4000	BC308000	
END	BC308200	
	BC309000	

SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN
SNOCHZ STORAGE MAP

02/18/68

PAGE 5

SUBROUTINE SNOCHK
COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	00000	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ICENTRL	00000	I	CONTROL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
UNODE	00033	I	NRAYS	00034	I	NSLCS	00038	I
ICLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00049	I	NSURF	00046	I
FLAG	00047	I	DELY	00080	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
EXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
OLFIN	00080	R	OMGAS	00081	R	OMGAL	00082	R
OXID	00083	R	EPRAD	00084	R	PSCAL	00085	R
OMGAF	00086	R	SPFE	00087	R	DUMIN	00070	R
ODIST	00071	R	NO	00072	R	DELH	00073	R
SYSHX	00074	R	WXDIR	00075	R	WYDIR	00076	R
NDTAN	00077	R	NOBGN	00100	I	NOBGT	00104	I
NOBJH	00117	R	WCLRH	00126	R	WIMVT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
BMDS	06210	I	NCOND	06663	I			
COMMON BLOCK			PRNT	ORIGIN	00007	LENGTH	00002	
LINE	00000	I	PAGE	00001	I			
COMMON BLOCK			AZOBJ	ORIGIN	00071	LENGTH	00003	
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R
COMMON BLOCK			TMDESN	ORIGIN	13074	LENGTH	13770	
PERIOD	00000	I	FCOUNT	00001	I	HOFFND	00002	I
JAIL	00003	I	SENTN	00147	I	DIMVCT	00161	I
ORJ	00162	I	SIZE	00170	R	DETRM	00171	R
ESLNG	00172	R	ESLNG	00173	R	IMPRV	00174	R
VCTSTR	00175	R	VINDX	13750	I	NVARX	13754	I
J1	13755	I	NJ	13752	I	NJA	13751	I
MA	13750	I	U	13747	R	RL	13746	I
LA	13748	I	UNITN	13744	R	UNAX	13743	R
ICOLOR	13735	I	QMTN	13734	R	QMAX	13733	R
INDX	13732	I	ODIST	13444	R	ALENG	13731	R
INTRX	13300	R	QINTRX	12624	R	YONAT	12620	R
HTAB	12614	R	SHISS	12613	I	XVCT	12610	R
QVCT	12609	R	TAU	12604	R	NUPPER	12603	I
KIND	12602	I						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
--------	----------	------	--------	----------	------	--------	----------	------

SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN
 BNDCHK STORAGE MAP

02/16/68

PAGE 6

I	22664	I	M	22669	I	L	22668	I
NC	22667	I	J	22670	I	JNOX	22671	I
K	22672	I						

ENTRY POINTS

BNDCHK SECTION 11

SUBROUTINES CALLED

CORDER	SECTION	12	DPINT	SECTION	13	.FWRD.	SECTION	14
PRMSUB	SECTION	15	RAYTR	SECTION	16	.FXEM.	SECTION	17
.UNDB.	SECTION	18	.FFIL.	SECTION	19	.FCNV.	SECTION	20
E.1	SECTION	21	E.2	SECTION	22	E.3	SECTION	23
E.4	SECTION	24	CC.1	SECTION	25	CC.2	SECTION	26
CC.3	SECTION	27	CC.4	SECTION	28	SYSLOC	SECTION	29
.FRET.	SECTION	30						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
100	1A	23017	5000	91A	23356	3000	77A	23314
2500	75A	23311	1000	42A	23162	500	38A	23144
3500	82A	23321	1500	55A	23220	2200	68A	23276
2000	65A	23262	4000	87A	23347	32010	FORMAT	22712
6000	117A	23464	5500	104A	23414	32020	FORMAT	22734
7500	146A	23623	7000	142A	23615	8000	150A	23630
8500	152A	23633	6500	129A	23563	9000	153A	23635
9200	158A	23634	9400	160A	23672	9600	162A	23710
32030	FORMAT	22751	32040	FORMAT	22763	32050	FORMAT	22777

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 24008.

TR-67-700-10-2
Page III-5-9
Program Listing
Link 5

INF .M910001.44800-0.77387.AM IC ISJOB LINK1 7094 A 02/18/68 PAGE 7

BT
BTITLE SUBROUTINE TO ORDER A SET OF INDICES OF REFRACTION CO001000
SIBFTC CORDEZ M94.XR7 LINK 5 (CORDEZ)

SUBROUTINE TO ORDER A SET OF INDICES OF REFRACTION
 CORDER STORAGE MAP

02/16/68

PAGE 2

SUBROUTINE CORDER
 UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
I	0001	I						
			ENTRY POINTS					
			SUBROUTINES CALLED					
CORDER	SECTION	3						
SYBLOC	SECTION	4						
			EFN IFN CORRESPONDENCE					
EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	BA	00031						
THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00102.								

TR-67-700-10-2
Page III-5-12
Program Listing
Link 5
Page Revised 10/1/68

INF,NO10001,44903-0,77387,AM IC

18JOB L10MKC 7094 A 09/24/68

PAGE

84
BTITLE SUBROUTINE TO CALCULATE REVISED ERROR VECTOR
BTFC CYCLEZ M94,XR7 LINK 5 (CYCLE)

CY001000

SUBROUTINE TO CALCULATE REVISED ERROR VECTOR
CYCLEZ - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

```

SUBROUTINE CYCLE(9)                                CY001200
C
C *** DESCRIPTION ***                               CY001400
C CYCLE INCREMENTS THE DESIGN PARAMETERS, CALCULATES THE WEIGHTED CY001410
C ERROR VECTOR, COMPUTES THE SOLUTION VECTOR, TESTS FOR BOUNDARY CY001420
C VIOLATIONS, AND DETERMINES THE NEW ERROR VECTOR. IN CASES CY001430
C WHERE A NEGATIVE IMPROVEMENT OCCURS THE PARAMETERS ARE RESET. CY001440
C FOR SUCCESSIVE FAILURES THE ALTERNATE RETURN IS USED. CY001450
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), FUNCID, DT001000
1 DATA(3485)                                       DT001100
DIMENSION WDBJH(7), WCLR(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NDBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMD), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXFF), (DATA(20), DEXFF), (DATA(21), DT101600
7 WEXFF), (DATA(22), DLPLN), (DATA(23), OMGA2), (DATA(24), DT101700
8 OMGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SFPEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), HD), (DATA(33), DT102000
B DELH), (DATA(34), SYSMX), (DATA(35), WDIR), (DATA(36), DT102100
C WDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WDBJH(1)), (DATA(60), WCLR(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
EQUIVALENCE (DATA(3484), ATRGGR), (DATA(3485), GAUSS) DT103000
INTEGER CNTRL, AFLAG, DATE, ATRGGR DT201000
REAL LATT DT202000
C
COMMON / PRNT / LINE, PAGE                          PT002500
INTEGER PAGE                                       PT002700
C
COMMON / IMDES / PERIOD, PCOUNT, NFFND, JAIL(10,10), SENTN(10), L7001000
1 DIMVCT, CORD(6), SIZE, DETRM, EGLNG, ESLNG, IMPRV, VCTSTR(6000), L7002000
2 VINDX(11), DJOLO(7)                             L7022100
REAL IMPRV                                         L7003000
INTEGER PERIOD, PCOUNT, SENTN, DIMVCT, CORD, VINDX L7004000
C
DIMENSION BMATRX(10), AMATRX(10,10), S1(30), S2(10) CY001500
C
C INITIALIZE ...                                   CY002000
SIZE=0.                                           CY002200
DETRM=0.                                          CY002400
ESLNG=0.                                          CY002600
IMPRV=0.                                          CY002800

```

SUBROUTINE TO CALCULATE REVISED ERROR VECTOR
CYCLEZ - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

JMAX=NDSCN(3)	CY003000	
IMAX=NDSCV(1)	CY003200	
C		
C SET DELTA VJ= MAX(ABS(DELTA VJ),DUMIN) J=1,2,...,JMAX ...	CY004000	
DO 50 J=1,JMAX	CY004200	
50 DESGN(3,J)=AMAX1(ABS(DESGN(3,J)),DUMIN)	CY004400	
C	CY004600	
COMPUTE IK K=1,2,...,IMAX ...	CY004800	
C	CY005000	
DO 170 I=1,IMAX	CY005200	
C INCREMENT VARIABLE JI ...	CY005400	
CALL INCR(INDSGV(I+1),1.)	CY005600	16
KZ1=VINDX(I+1)	CY005800	
COMPUTE E(U0+DELTA UJ) ...	CY006000	
CALL MEVCTR(VCTSTR(KZ1),\$47D,0,GAUSS)	CY006200	20
CALL INCR(INDSGV(I+1),-1.)	CY006400	
COMPUTE IJ=EJ-EG ...	CY006600	25
DO 150 K=1,DIMVCT	CY006800	
KZE=VINDX(1)+K-1	CY007000	
KZ1=KZ1+K-1	CY007200	
150 VCTSTR(KZ1)=VCTSTR(KZ1)-VCTSTR(KZE)	CY007400	
170 CONTINUE	CY007600	
C	CY007800	
COMPUTE ED.IJ J=1,2,...,IMAX ...	CY008000	
C	CY008200	
DO 230 I=1,IMAX	CY008400	
KZ1=VINDX(I+1)	CY008600	
CALL DOTP(VCTSTR(DINVCT+1),VCTSTR(KZ1),BMATRIX(I))	CY008800	44
230 BMATRIX(I)=-BMATRIX(I)	CY009000	
C	CY101000	
COMPUTE MATRIX I (IMAX,IMAX) WITH ELEMENTS IJ.IK J,K=1,2,...,IMAX ...	CY101200	
C	CY101400	
DO 350 I=1,IMAX	CY101600	
DO 350 J=1,IMAX	CY101800	
IF(J.LT.1)GO TO 310	CY102000	
KZE=VINDX(I+1)	CY102200	
KZ1=VINDX(J+1)	CY102400	
CALL DOTP(VCTSTR(KZE),VCTSTR(KZ1),AMATRIX(I,J))	CY102600	65
IF(J.GT.1)GO TO 350	CY102800	
KZ1=NDSCV(I+1)	CY103000	
DESGN(10,KZ1)=AMATRIX(I,I)	CY103200	
GO TO 350	CY103400	
310 AMATRIX(I,J)=AMATRIX(J,I)	CY103600	
350 CONTINUE	CY103800	
C	CY104000	
C INVERT LINEAR SYSTEM ...	CY104200	
C	CY104400	
IF(IMAX.NE.1)GO TO 390	CY104600	
IF(AMATRIX(1,1).EQ.D.)GO TO 410	CY104800	
DETRM=AMATRIX(1,1)	CY105000	
BMATRIX(1)=BMATRIX(1)/DETRM	CY105200	
GO TO 500	CY105400	
390 CALL MATINV(10,AMATRIX,IMAX,BMATRIX,1,DETRM,S1,S2)	CY105600	92
IF(DETRM.NE.D.)GO TO 500	CY105800	
C	CY106000	
MATRIX IS SINGULAR ...	CY106200	

SUBROUTINE TO CALCULATE REVISED ERROR VECTOR
CYCLEZ - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

C		CY106400	
410	CALL DFRT	CY106600	98
	WRITE (6,4100)	CY106800	99
4100	FORMAT(10X27HSOLUTION MATRIX IS SINGULAR)	CY107000	
430	LINE=LINE+1	CY107200	
	PCOUNT=PCOUNT+1	CY107400	
	IF(PCOUNT.LT.PERIOD)GO TO 460	CY107600	
	WRITE (6,4101)PCOUNT	CY107800	105
4101	FORMAT(10X16HDESIGN ABANDONED13,20H SUCCESSIVE FAILURES)	CY108000	
	RETURN	CY108200	
460	CALL JURY	CY108400	107
	RETURN	CY108600	
470	CALL INCRM(NDSGV(I+1),-1.)	CY108800	111
	CALL DFRT	CY109000	113
	WRITE (6,4102)NDSGV(I+1)	CY109200	114
4102	FORMAT(10X75HRAY FAILURE WHILE COMPUTING DERIVATIVE OF E WITH RESP	CY109400	
	ECT TO DESIGN VARIABLE13)	CY109600	
	GO TO 430	CY109800	
C		CY201000	
C	SCALE THE SOLUTION VECTOR ...	CY201200	
C		CY201400	
500	SIZE=ZETA	CY201600	
	DO 540 I=1,IMAX	CY201800	
540	SIZE=AMAX1(SIZE,ABS(BMATRX(I)))	CY202000	
	SIZE=(2.*ZETA)/SIZE		
	IFLAG = 0		
	DO 8000 KX=1,10		
	SIZE=SIZE/2.		
	DO 590 I=1,IMAX		
590	BMATRX(I)=SIZE*BMATRX(I)		
	DO 630 I=1,IMAX		
630	CALL INCRM(NDSGV(I+1),BMATRX(I))		140
	CALL BNDCHK(\$7900, IFLAG)		143
	GO TO 8002		
7900	DO 7902 I=1,IMAX		
7902	CALL INCRM(NDSGV(I+1),-BMATRX(I))		152
	IFLAG = 1		
	DO 7904 I=1,IMAX		
7904	BMATRX(I)=BMATRX(I)/SIZE		
8000	CONTINUE		
	GO TO 870		
8002	CONTINUE		
	COMPUTE ESTAR=E(USTAR) ...	CY205200	
C		CY205400	
	KZ1=VINDX(2)	CY205600	
	CALL MEVCTR(VCTSTR(KZ1), \$790, ATRCCR, GAUSS)	CY205800	169
	CALL DOTF(VCTSTR(KZ1), VCTSTR(KZ1), ESLNG)	CY206000	174
	ESLNG=SQRT(ESLNG)	CY206200	176
	IMPRV=(EDLNG-ESLNG)/EDLNG	CY206400	
	IF(IMPRV.LE.0.1)GO TO 810	CY206600	
	CALL DFRT	CY206800	180
	IF(AFLAG.NE.0) CALL EPRNT	CY206900	183
	EDLNG=ESLNG	CY207000	
	PCOUNT=0	CY207200	
	DO 770 K=1,CIMVCT	CY207400	
	KZE=VINDX(1)+K-1	CY207600	

SUBROUTINE TO CALCULATE REVISED ERROR VECTOR		09/24/68	PAGE
CYCLEZ - EFN SOURCE STATEMENT - IFN(S) -			
	KZ1=VINCK(2)+K-1	CY207000	
770	VCTSTR(KZE)=VCTSTR(KZ1)	CY208000	
	CALL MODDL	CY208200	195
	RETURN	CY208400	
C		CY301000	
C	RAY FAILURE ...	CY301200	
C		CY301400	
790	CALL DPRNT	CY301600	198
	WRITE (6,4103)	CY301800	199
4103	FORMAT(10X49H'RAY FAILURE WHILE COMPUTING IMPROVED ERROR VECTOR)	CY302000	
	GO TO 830	CY302200	
810	CALL DPRNT	CY302400	202
	WRITE (6,4104)	CY302600	203
4104	FORMAT(10X20H'NEGATIVE IMPROVEMENT)	CY302800	
830	DO 860 I=1,IMAX	CY303000	
850	CALL INCRM(INDSGV(I+1),-DMATRX(I))	CY303200	211
	IF(ATRCCR.EQ.0)GO TO 870		
	IF(NSUBT.NE.0)GO TO 8610		
	DO 8600 J=1,NDBJH		
8600	C(1PL(J)=D(JOLD(J)		
	GO TO 870		
8610	CALL MEVCTR(VCTSTR(DIMVCT+1),\$8620,ATRCCR,GAUSS)		232
870	CALL GETRHD(BF1)	CY303400	235
	CALL PRMSUB(\$870)	CY303600	237
	CALL MODDL	CY303800	240
	GO TO 430	CY304000	
C			
8620	WRITE (6,8622)		242
8622	FORMAT(10X,59H'UNEXPLAINABLE RAY FAILURE WHILE ATTEMPTING TO REGENE		
	STATE EO)		
	CALL GOOF		243
	RETURN		
	END	CY305000	

SUBROUTINE TO CALCULATE REVISED ERROR VECTOR
 CYCLEZ

STORAGE MAP

09/24/68

PAGE 1

SUBROUTINE CYCLE
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	FUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIFLN	00042	I	IMODE	00043	I
NSFLN	00044	I	NDBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXFP	00055	R	DEXFP	00056	R	WEXFP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
OD1ST	00071	R	MO	00072	R	DELH	00073	R
SYSHX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WDBJH	00117	R	WCLRH	00126	R	WIMHT	00135	R
EINH1	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I	ATRCR	06666	I
GAUSS	06667	R						

LINE	COMMON BLOCK	FRNT	ORIGIN	06672	LENGTH	00002
	00000	I	00001	I		

COMMON BLOCK			TMDESH	ORIGIN	06674	LENGTH	13777	
PERIOD	00000	I	PCOUNT	00001	I	NDFND	00002	I
JAIL	00003	I	SENTN	00147	I	DIMVCT	00161	I
CORD	00162	I	SIZE	00170	R	DETRM	00171	R
EOLNG	00172	R	ESLNG	00173	R	IMPRV	00174	R
VCTS1R	00175	R	VINDX	13755	I	DJOLD	13770	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DMATRX	22673	R	AMATRX	22705	R	S1	23051	R
S2	23107	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
JMAX	23121	I	IMAX	23122	I	J	23123	I
I	23124	I	KZ1	23125	I	K	23126	I
KZE	23127	I	KZ1	23130	I	IFLAG	23131	I
KCX	23132	I	BF1	23133	R			

SUBROUTINE TO CALCULATE REVISED ERROR VECTOR
 CYCLE2

09/24/68

PAGE 6

STORAGE MAP

ENTRY POINTS

CYCLE	SECTION	9
INCRH	SECTION	10
MATINV	SECTION	13
JURY	SECTION	16
EFRT	SECTIC	19
FRMSUB	SECTION	22
.FFIL.	SECTION	25
.FRET.	SECTION	28

SUBROUTINES CALLED

MEVCTR	SECTION	11	DOTP	SECTION	12
DFRNT	SECTION	14	.FWRD.	SECTION	15
BNDCHK	SECTION	17	SQRT	SECTION	16
WDDL	SECTION	20	GETIND	SECTION	21
GOOF	SECTION	23	.JUNG.	SECTION	24
.FCNV.	SECTION	26	SYSLOC	SECTION	27

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
50	7A	23261	170	35A	23374	470	109A	23640
150	30A	23360	230	47A	23423	350	80A	23541
310	77A	23537	390	91A	23564	410	97A	23603
500	117A	23675	4100	FORMAT	23153	430	100A	23614
460	106A	23642	4101	FORMAT	23161	4102	FORMAT	23172
540	121A	23703	8000	162A	24025	590	131A	23747
630	137A	23755	7900	146A	24000	8002	166A	24030
7902	149A	24001	7904	158A	24020	870	234A	24246
790	197A	24147	810	201A	24161	770	190A	24131
4103	FORMAT	23211	830	204A	24172	4104	FORMAT	23223
850	208A	24176	8610	230A	24233	8600	224A	24226
8620	242A	24262	8622	FORMAT	23230			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 24332.

TR-67-700-10-2
Page III-5-19
Program Listing
Link 5

IMP.M010001.44800-0.77387.AM IC

ISJOB LINK1 7094 A 02/18/58

PAGE 7

00
TITLE SUBROUTINE TO CALCULATE VECTOR INNER PRODUCT
SIBFTC DOTPZ MS4.KR7 LINK 5 (DOTP)

00001060

SUBROUTINE TO CALCULATE VECTOR INNER PRODUCT
 DOTPZ

STORAGE MAP

02/16/68

PAGE 2

SUBROUTINE DOTP
 COMMON VARIABLES

COMMON BLOCK			TMDESN	ORIGIN	00001	LENGTH	13770	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
PERIOD	00000	I	PCOUNT	00001	I	NOFFND	00002	I
JAIL	00003	I	SENTN	00147	I	DINVCT	00161	I
ORD	00162	I	SIZE	00170	R	DETRM	00171	R
EOLNG	00172	R	ESLNG	00173	R	IMPRV	00174	R
VCTSTR	00175	R	VINDY	13755	I			

ENTRY POINTS

DOTP SECTION 5

SUBROUTINES CALLED

SYSLOC SECTION 6

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	5A	1400E						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 14043.

TR-67-700-10-2
Page III-5-22
Program Listing
Link 5

INF.ND100D1.44800-0.77387.AM IC

IBJOB LINK1 7094 A

02/16/68

PAGE 3

30
TITLE SUBROUTINE TO PRINT OUT DESIGN INFORMATION
JIBFC DPRINTZ M94.XR7 LINK 5 (DPRINT,EPRT)

DP001000

SUBROUTINE TO PRINT OUT DESIGN INFORMATION
DPRNTZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 1

```

SUBROUTINE DPRNT                                DP001200
C
C *** DESCRIPTION ***                            DP001400
C DPRNT AND EPRNT PRINT OUT VARIOUS PARAMETERS DURING THE DESIGN DP001410
C OPERATION. DPRNT PRINTS VALUES ASSOCIATED WITH DESIGN STATUS DP001420
C AT EACH STEP. EPRNT PRINTS THE PARAMETER VALUES IN EACH STEP. DP001430
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483)                                     DT001100
DIMENSION WOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMG2), (DATA(24), DT101700
8 OMG1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUNIN), (DATA(31), ODIST), (DATA(32), HO), (DATA(33), DT102000
B DELW), (DATA(34), SYSHX), (DATA(35), WXDIR), (DATA(36), DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLRH(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2182), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
COMMON / PRT / LINE, PAGE PT002500
INTEGER PAGE PT002700
C
COMMON / TDES / PERIOD, PCOUNT, NOFFND, JAIL(10,10), SENTN(10), L7001000
1 DIMVCT, CORD(6), SIZE, DETRM, EDLNG, ESLNG, IMPRV, VCTSTR(6000), L7002000
2 VINDX(11) L7002100
REAL IMPRV L7003000
INTEGER PERIOD, PCOUNT, SENTN, DIMVCT, CORD, VINDX L7004000
C
DIMENSION VLST(6), FNT(8), FMT(6) DP001500
C
DATA FMT(1)/6H(16H00)/, FMT(2)/6HESGN /, FMT(3)/6HVARIAB/, DP001600
1 FMT(4)/6HLE. /, FMT(6)/6H6X9HPA/, FMT(7)/6HRAMETE/, DP001700
2 FMT(8)/6HR.12)/ DP001800
DATA FMT / 2H1/, 2H2/, 2H3/, 2H4/, 2H5/, 2H6/ / DP001900
C
C START OF DPRNT SUBROUTINE DP002000
C IF ( LINE .LE. 58 ) GO TO 1000 DP002200
C BEGIN NEW PAGE DP003000

```

SUBROUTINE TO PRINT OUT DESIGN INFORMATION
 DPRINTZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 2

CALL PRTCTL	DP003200	5
WRITE (6, 32010)	DP003400	6
32010 FORMAT(1H0, 3X, 23HDESIGN VARIABLE NUMBERS, 6X, 12HINCR. FACTOR,	DP003600	
1 3X, 11HDETERMINANT, 5X, 10HOLD LENGTH, 5X, 10HNEW LENGTH, 4X,	DP003800	
2 11HIMPROVEMENT)	DP004000	
LINE = LINE + 2	DP004200	
1000 WRITE (6, 32020) (NDSGV(I), I = 2, 11), SIZE, DETRN, EOLNG,	DP005000	
1 ESLNG, IMPRV	DP005200	7
32020 FORMAT(1H , I2, 9I3, 5E15.6)	DP005400	
LINE = LINE + 1	DP005600	
RETURN	DP006000	
C		
ENTRY EPRNT	DP101000	
NVARX = NDSGV(1)	DP101200	
IF (NVARX+2+LINE .LE. 56) GO TO 2000	DP101400	
BEGIN NEW PAGE	DP101600	
C		
CALL PRTCTL	DP101800	20
WRITE (6, 32010)	DP102000	21
LINE = LINE + 2	DP102200	
2000 LINE = LINE + NVARX + 2	DP102400	
NJMAX = 0	DP102600	
DO 3000 I = 1, NVARX	DP103000	
JI = NDSGV(I+1)	DP103200	
NJMAX = MAX0(NJMAX, IABS(IDESGN(1,JI)))	DP103400	
3000 CONTINUE	DP103600	
FMT(5) = FMTT(NJMAX)	DP103700	
WRITE (6, FMT) (I, I = 1, NJMAX)	DP103800	35
DO 5000 I = 1, NVARX	DP104200	
JI = NDSGV(I+1)	DP104400	
NJ = IABS(IDESGN(1,JI))	DP104600	
DO 4000 J = 1, NJ	DP105000	
KDX = IABS(IDESGN(J+3,JI))	DP105200	
4000 VLST(J) = DATA(KDX)	DP105400	
WRITE (6, 32050) JI, (VLST(J), J=1,NJ)	DP105600	54
32050 FORMAT(6X I3, 9X 6E17.8)	DP106000	
5000 CONTINUE	DP106200	
C		
RETURN	DP109000	
END	DP109200	

SUBROUTINE TO PRINT OUT DESIGN INFORMATION
 DPRINTZ

02/16/68

PAGE 3

STORAGE MAP

SUBROUTINE DPRINT
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ICNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
UMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
AFLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
HEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PUMCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HO	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WOBXH	00117	R	WCLRH	00126	R	WINHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I			
COMMON BLOCK			PRNT	ORIGIN	06667	LENGTH	00002	
LINE	00000	I	PAGE	00001	I			
COMMON BLOCK			TMDESN	ORIGIN	06671	LENGTH	13770	
PERIOD	00000	I	PCOUNT	00001	I	NOFFND	00002	I
JAIL	00003	I	SENTN	00147	I	DINVCT	00161	I
CORD	00162	I	SIZE	00170	R	DETRM	00171	R
EOLNG	00172	R	ESLNG	00173	R	IMPRV	00174	R
VCTSTR	00175	R	VINDX	13755	I			
DIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
VLST	22661	R	FMT	22667	R	FMTT	22677	R
UNDIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
I	22705	I	NVARK	22706	I	NJMAX	22707	I
J1	22710	I	NJ	22711	I	KDX	22712	I
ENTRY POINTS								

SUBROUTINE TO PRINT OUT DESIGN INFORMATION
 DPRINTZ

STORAGE MAP

02/16/68

PAGE 4

DPRINTZ			STORAGE MAP			SUBROUTINES CALLED		
EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
100J	7A	23011	32010	FORMAT	22731	32020	FORMAT	22761
2000	22A	23087	3000	31A	23121	5000	59A	23235
4000	51A	23208	32080	FORMAT	22765			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 23265.

TR-67-700-10-2
Page III-5-27
Program Listing
Link 5

INF.M010L..1.64800-0.77387.AM IC ISJOB LINK1 7094 A 02/18/68 PAGE 5

34
BTITLE SUBROUTINE TO COMPUTE BACK FOCUS AND ENTR. PUPIL RADIUS
31BPTC GETRHZ M94.XRT LINK 5 (GETRND) 6R001000

SUBROUTINE TO COMPUTE BACK FOCUS AND ENTR. PUPIL RADIUS
GETRHZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 1

```

SUBROUTINE GETRHZ( BF1 )                                GR001200
C
C *** INPUTS ***                                       GR001600
C EPRAD INITIAL ENTRANCE PUPIL RADIUS.                (DATA COMMON) GR001610
C FNUMB F / NUMBER.                                   (DATA COMMON) GR001620
C
C *** OUTPUTS ***                                       GR001630
C BF1 COMPUTED BACK FOCUS FOR COLOR 1.                GR001640
C EPRAD RECOMPUTED ENTRANCE PUPIL RADIUS IF FNUMB IS NON-ZERO. GR001650
C
C *** DESCRIPTION ***                                    GR001660
C GETRHZ COMPUTES THE BACK FOCUS FOR COLOR 1. ALSO IF FNUMB .NE. GR001670
C 0. A REVISED EPRAD IS CALCULATED.                  GR001680
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483)                                          DT001100
DIMENSION WOBJH(7), WCLRH(7), WINHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 INODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMGA2), (DATA(24), DT101700
8 OMGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUNIN), (DATA(31), ODIST), (DATA(32), HD), (DATA(33), DT102000
B DELH), (DATA(34), SYSMX), (DATA(35), WDIR), (DATA(36), DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLRH(1)), DT102300
E (DATA(67), WINHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2102), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
INTEGER CNTRL, AFLAG, DATE QT201000
REAL LATTC DT202000
C
COMMON / AZOBJH / THTR, SNTHTR, CSTHTR AZ004000
C
C START OF GETRHZ SUBROUTINE GR002000
C
C THTR = 0. GR002400
C SNTHTR = 0. GR002600
C CSTHTR = 1. GR002800
CALL MERID( 1, BF1, FNUMBC ) GR003000
IF( FNUMB .EQ. 0.0 ) GO TO 32000 GR003200
DO 2000 I = 1, 3 GR003400
EPRAD = ABS( EPRAD * FNUMBC / FNUMB ) GR003600
IF( I .EQ. 3 ) GO TO 32000 GR003800

```

TR-67-700-10-2
Page III-5-29
Program Listing
Link 5

SUBROUTINE TO COMPUTE BACK FOCUS AND ENTR. PUPIL RADIUS
GETRNZ - EPN SOURCE STATEMENT - IPN(8) -

02/18/68

PAGE 2

CALL HERID(1, DUMMY, PNUMBC)
2000 CONTINUE
C
32000 RETURN
END

CR004000 13
CR004500
CR005000
CR006000

SUBROUTINE TO COMPUTE BACK FOCUS AND ENTR. PUPIL RADIUS
 GETRHZ STORAGE MAP

02/16/68

PAGE 3

SUBROUTINE GETRHO
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R	
DATE	00027	I	PUNCID	00032	R	DATA	00033	R	
LMODE	00033	I	NRAYS	00034	I	NSLCS	00039	I	
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I	
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I	
NSPLN	00044	I	NOBJN	00049	I	NSURF	00046	I	
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R	
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R	
WEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R	
DLPLN	00060	R	OMGAP	00061	R	OMGAI	00062	R	
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R	
OMGAP	00066	R	SPFEA	00067	R	DUMIN	00070	R	
ODIST	00071	R	HO	00072	R	DELH	00073	R	
SYSNK	00074	R	WXDIR	00079	R	WYDIR	00076	R	
NOTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I	
WOB JH	00117	R	WCLRH	00126	R	WIMHT	00135	R	
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R	
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R	
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I	
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R	
BNDS	06210	I	NCOND	06665	I				

COMMON BLOCK			AZOBJ	ORIGIN	08667	LENGTH	00003		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R	

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
FNUMBC	08672	R	I	08673	I	DUMMY	08674	R

ENTRY POINTS

GETRHO SECTION 7

SUBROUTINES CALLED

MERID SECTION 8

SYSLOC SECTION 9
 EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	174	08750	2000	14A	08746			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 08771.

TR-67-700-10-2
Page III-5-31
Program Listing
Link 5

INF.N010001.44000.0.77307.AM IC

IBJOB L10M1 7094 A

02/16/68

PAGE 4

70
BTITLE SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE
SIBFTC INCRIZ H94.XR7 LINK 5 (INCRM)

!N001000

SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE
INCR - EFN SOURCE STATEMENT - IFN(S) -

02/16/68

PAGE 1

```

SUBROUTINE INCRM( J, FACTOR )                                IN001400
C
C *** INPUTS ***                                           IN001600
C J THE CURRENT DESIGN VARIABLE IN DESGN.                  IN001610
C FACTOR THE MULTIPLIER APPLIED TO THE INCREMENTAL VALUE BEFORE IN001620
C INCREMENTATION OF EACH PARAMETER.                        IN001630
C
C *** OUTPUTS ***                                          IN001640
C DATA THE INCREMENTED DESIGN PARAMETERS ARE SET DIRECTLY IN DATA. IN001650
C
C *** DESCRIPTION ***                                       IN001660
C INCRM INCREMENTS THE CURRENT DESIGN VARIABLE PARAMETERS IN001670
C ACCORDING TO THE INCREMENTAL FACTOR AND THE PARAMETER TYPE. IN001680
C THE NEW VALUE IS PLACED IN THE PROPER LOCATION OF DATA VECTOR. IN001690
C
COMMON / DATA / NCNTRL, CONTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483)                                               DT001100
DIMENSION WOBJH(7), WCLR(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,107), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3), DT101000
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9), DT101200
3 IMODE ), ( DATA(10), NSPLN ), ( DATA(11), NOBJH ), ( DATA(12), DT101300
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
5 FNUMB ), ( DATA(16), FLNCH ), ( DATA(17), WFLGH ), ( DATA(18), DT101500
6 ZETA ), ( DATA(19), HEXP ), ( DATA(20), DEXPP ), ( DATA(21), DT101600
7 WEXPP ), ( DATA(22), DLPLN ), ( DATA(23), OMGA2 ), ( DATA(24), DT101700
8 OMGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
9 PSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
A DUMIN ), ( DATA(31), ODIST ), ( DATA(32), HO ), ( DATA(33), DT102000
B DELH ), ( DATA(34), SYSMX ), ( DATA(35), WXDIR ), ( DATA(36), DT102100
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
D NDSGV(1) ), ( DATA(53), WOBJH(1) ), ( DATA(60), WCLR(1) ), DT102300
E ( DATA(67), WIMHT(1) ), ( DATA(74), EIMHT(1) ), ( DATA(81), DT102400
F CIMPL(1) ), ( DATA(88), WCLRS(1) ), DT102500
G ( DATA(94), LATT(1,1), ILATT(1,1) ), ( DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1) ), ( DATA(2182), DESGN(1,1), IDESGN(1,1) ), DT102700
I ( DATA(2682), SUBST(1), ISUBST(1) ), ( DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1) ), ( DATA(3483), NCOND ) DT102900
INTEGER CONTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
C START OF INCRM SUBROUTINE                                IN002400
C
C NJ = IDESGN(1,J)                                          IN003000
IF( NJ .NE. 0 ) GO TO 1000 IN003200
WRITE ( 6, 32010 ) J IN003400
32010 FORMAT( 1H0, 9X, 15HDESIGN VARIABLE, 13, 23H IS INCORRECTLY DEFINE IN003600
1D ) IN003800
CALL GOOF IN004000
1000 NJA = IABS( NJ ) IN005000
DELTA = FACTOR * DESGN(3,J) IN005200
DO 6000 I = 1, NJA IN005400

```

SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE
INCRMZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 2

```
      KIJ = IDESGN(I+3,J)
      KIJA = IABS(KIJ)
      IF( NJ .LT. 0 ) GO TO 2500
      IF( KIJ .LT. 0 ) GO TO 2000
      DATA(KIJA) = DATA(KIJA) + DELTA
      GO TO 6000
2000 DATA(KIJA) = DATA(KIJA) - DELTA
      GO TO 6000
2500 IF( DATA(KIJA) .NE. 0. ) GO TO 3500
      IF( DELTA .EQ. 0. ) GO TO 6000
      IF( KIJ .LT. 0 ) GO TO 3000
      DATA(KIJA) = 1. / DELTA
      GO TO 6000
3000 DATA(KIJA) = - 1. / DELTA
      GO TO 6000
3500 IF( KIJ .LT. 0 ) GO TO 4000
      TMP = 1. / DATA(KIJA) + DELTA
      GO TO 4500
4000 TMP = 1. / DATA(KIJA) - DELTA
4500 IF( TMP .NE. 0. ) GO TO 5000
      DATA(KIJA) = 0.
      GO TO 6000
5000 DATA(KIJA) = 1. / TMP
6000 CONTINUE
C
      RETURN
      END
```

```
IN005600
IN005800
IN006000
IN006200
IN006400
IN006600
IN006800
IN007000
IN007200
IN007400
IN007600
IN007800
IN008000
IN008200
IN008400
IN101000
IN101200
IN101400
IN101600
IN101800
IN102000
IN102200
IN102400
IN103000

IN104000
IN105000
```


SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE
 INCRMZ

02/16/68

PAGE 3

STORAGE MAP

SUBROUTINE INCRM
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666		
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R	
DATE	00027	I	PUNCID	00032	R	DATA	00033	R	
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I	
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I	
NSUBP	00041	I	N7PLN	00042	I	IMODE	00043	I	
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I	
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R	
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R	
HEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R	
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R	
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R	
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R	
QDIST	00071	R	HD	00072	R	DELH	00073	R	
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R	
RD TAN	00077	R	NDSGN	00100	I	NDSGV	00104	I	
WOB JH	00117	R	WCLRH	00126	R	WIMHT	00139	R	
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R	
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R	
ISURFC	00320	I	DESIGN	04240	R	IDESGN	04240	I	
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R	
IBHDS	06210	I	NCOND	06665	I				

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NJ	06667	I	NJA	06670	I	DELTA	06671	R
KIJ	06672	I	KIJA	06673	I	TMP	06674	R

ENTRY POINTS

INCRM SECTION 5

SUBROUTINES CALLED

.FWRD. SECTION 6
 .FFIL. SECTION 9

GOOF SECTION 7
 .FCNV. SECTION 10
 EFN IFN CORRESPONDENCE

.UNDB. SECTION 8
 SYSLOC SECTION 11

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	AA	06746	32010	FORMAT	06707	6000	59A	07134
2500	AA	07027	2000	26A	07017	3500	45A	07067
3000	AA	07056	4000	50A	07105	4500	52A	07115
5000	57A	07126						

TR-67-700-10-2
Page III-5-35
Program Listing
Link 5

SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE
INCRMZ STORAGE MAP

02/16/68

PAGE 4

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07164.

TR-67-700-10-2
Page III-5-36
Program Listing
Link 5

INF.N010001.44800-0.77387.AM IC

ISJOB L10M1 7094 A

02/16/68

PAGE 3

00
JTITLE SUBROUTINE TO TEST AND RESTRICT DESIGN VARIABLE COMBS. JUDD1000
JTBFC JURYZ M94.XR7 LINK 5 (JURY, JUDGE)

SUBROUTINE TO TEST AND RESTRICT DESIGN VARIABLE COMBS.
JURYZ - EFN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 1

```

SUBROUTINE JURY                                JU001500
C
C *** INPUTS ***                               JU001600
C INSGV VECTOR OF DESIGN VARIABLE COMBINATION. (DATA COMMON) JU001610
C NOFFND NUMBER OF COMBINATIONS CURRENTLY JAILED. (TMDESN COMMON) JU001620
C NJAIL JAIL FLAG EQUAL TO CYCLES JAILED WHEN NON-ZERO. (DATA COMMON) JU001630
C
C *** OUTPUTS ***                               JU001640
C JAIL MATRIX CONTAINING THE JAILED DESIGN COMBINATIONS. JU001650
C SENTN VECTOR CONTAINING THE SENTENCES OF THE JAILED COMBINATIONS. JU001660
C * ALTERNATE RETURN FROM JUDGE IF COMBINATION IS IN JAIL. JU001670
C
C *** DESCRIPTION ***                           JU001680
C ENTRY JURY IS USED TO PLACE A DESIGN COMBINATION WHICH HAS JU001690
C RESULTED IN FAILURE, IN JAIL. THE SENTENCE IS SET AT NJAIL. JU001700
C ENTR/ JUDGE TESTS THE CURRENT COMBINATION TO DETERMINE IF IT JU001710
C IS IN JAIL. ALTERNATE RETURN TAKEN IF COMBINATION REMAINS IN JU001720
C JAIL.                                         JU001730
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483) DT001100
DIMENSION WOBJH(7), WCLR(7), WIMHT(7), EINHT(7), CIMPL(7), DT002000
1 WCLRS(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), ONGA2), (DATA(24), DT101700
8 ONGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), ONGAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), HO), (DATA(33), DT102000
B DELH), (DATA(34), SYSHX), (DATA(35), WDIR), (DATA(36), DT102100
C WDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLR(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EINHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2102), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3102), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATT DT202000
C
COMMON / TMDESN / PERIOD, PCOUNT, NOFFND, JAIL(10,10), SENTN(10), L7001000
1 DINVCT, CORD(6), SIZE, DETRM, EDLNG, ESLNG, IMPRV, VCTSTR(6000), L7002000
2 VINDX(11) L7002100
REAL IMPRV L7003000
INTEGER PERIOD, PCOUNT, SENTN, DINVCT, CORD, VINDX L7004000
C
C START OF JURY SUBROUTINE JU002000

```

SUBROUTINE TO TEST AND RESTRICT DESIGN VARIABLE COMBS.
 JURY2 - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 2

C	IF(NJAIL .EQ. 0) RETURN	JU002200
	IF(NOFFND .EQ. 10) RETURN	JU002400
	NOFFND = NOFFND + 1	JU002600
	IMAX = NDSGV(1)	JU002800
	DO 1000 I = 1, IMAX	JU003000
1000	JAIL(I,NOFFND) = NDSGV(I+1)	JU003200
	SENTN(NOFFND) = NJAIL	JU003400
	RETURN	JU003600
C		
C	START OF JUDGE SUBROUTINE	JU101000
C		
	ENTRY JUDGE(*)	JU101200
	IF(NJAIL .EQ. 0) RETURN	JU101400
	IF(NOFFND .EQ. 0) RETURN	JU101600
	IMAX = NDSGV(1)	JU101800
	DO 2500 J = 1, NOFFND	JU102000
	DO 2000 I = 1, IMAX	JU102200
	IF(NDSGV(I+1) .NE. JAIL(I,J)) GO TO 2500	JU102400
2000	CONTINUE	JU102600
	GO TO 3000	JU102800
2500	CONTINUE	JU103000
	RETURN	JU103200
3000	SENTN(J) = SENTN(J) - 1	JU103400
	IF(SENTN(J) .NE. 0) RETURN 1	JU103600
	DO 4500 J = 1, 9	JU104000
	SENTN(J) = SENTN(J+1)	JU104100
	DO 4000 I = 1, IMAX	JU104200
4000	JAIL(I,J) = JAIL(I,J+1)	JU104400
4500	CONTINUE	JU104600
	NOFFND = NOFFND - 1	JU104700
C		
C	END OF JUDGE SUBROUTINE	JU104800
	RETURN	JU105000
	END	JU106000

SUBROUTINE TO TEST AND RESTRICT DESIGN VARIABLE COMBS.
 JURYZ STORAGE MAP

02/16/68

PAGE 3

SUBROUTINE JURY
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
CENTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCI0	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCB	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
FLAG	00047	I	QELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	FSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HO	00072	R	DELH	00073	R
SYSHK	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WOB JH	00117	R	WCLRH	00126	R	WIMHT	00139	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATYC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I			
COMMON BLOCK			THDESN	ORIGIN	08667	LENGTH	13770	
PERIOD	00000	I	PCOUNT	00001	I	NOFFND	00002	I
JAIL	00003	I	SENTN	00147	I	DIMVCT	00161	I
ORD	00162	I	SIZE	00170	R	DETRM	00171	R
EDLNG	00172	R	ESLNG	00173	R	IMPRV	00174	R
VCTSTR	00175	R	VINDX	13755	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DNAX	22657	I	I	22660	I	J	22661	I

ENTRY POINTS

JURY SECTION 7

JUDGE SECTION 8
 SUBROUTINES CALLED

SYSLOC SECTION 9

.FRET. SECTION 10
 EFN IFN CORRESPONDENCE

TR-67-700-10-2
Page III-5-40
Program Listing
Link 5

SUBROUTINE TO TEST AND RESTRICT DESIGN VARIABLE COMBS.
JURYZ STORAGE MAP

02/18/68

PAGE 4

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	11A	22717	2500	36A	22771	2000	33A	22768
3000	39A	22774	4500	56A	23032	4000	52A	23026

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 23076.

TR-67-700-10-2
Page III-5-41
Program Listing
Link 5
Page Revised 10/1/68

INF,ND10001,44903-0,77387,AM 10

18JDD L10MKE 7094 A 09/24/68

PAGE 3

BTITLE SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR
SIDF10 MEVCTZ M94,XR7 LINK 5 (MEVCTR)

EV001000

SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR
MEVCTR - EFN SOURCE STATEMENT - IFN(5) -

09/24/68

PAGE

```

SUBROUTINE MEVCTR( FVCTOR, KFLAG1, KFLAG3 )          EV001500
C
C   *** OUTPUTS ***                                EV001600
C   VECTOR THE WEIGHTED ERROR VECTOR OF LENGTH DIMVCT. EV001610
C   ALTERNATE RETURN USED WHEN RAY TRACING FAILS.  EV001620
C
C   *** DESCRIPTION ***                             EV001630
C   MEVCTR USES THE WEIGHT VECTOR, VCTSTR, AND OTHER PARAMETERS TO EV001640
C   COMPUTE THE WEIGHTED ERROR VECTOR OF LENGTH DIMVCT. IF ANY EV001650
C   RAY MISSES OR REFLECTS DURING RAY TRACING THE ALTERNATE RETURN EV001660
C   IS MADE.                                         EV001670
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), FUNCID, DT001000
1 DATA(3486)                                       DT001100
DIMENSION WDBJH(7), WCLRHH(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3), DT101000
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9), DT101200
3 LMODE ), ( DATA(10), NSPLN ), ( DATA(11), WDBJH ), ( DATA(12), DT101300
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
5 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18), DT101500
6 ZETA ), ( DATA(19), HEXFF ), ( DATA(20), DEXFF ), ( DATA(21), DT101600
7 WEXFF ), ( DATA(22), DLPLN ), ( DATA(23), OMGA2 ), ( DATA(24), DT101700
8 OMGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
9 FSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
A DUMIN ), ( DATA(31), OD1ST ), ( DATA(32), HD ), ( DATA(33), DT102000
B DELH ), ( DATA(34), SYSMX ), ( DATA(35), WXD1R ), ( DATA(36), DT102100
C WYD1R ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
D NDSGV(1) ), ( DATA(53), WDBJH(1) ), ( DATA(60), WCLRHH(1) ), DT102300
E ( DATA(67), WIMHT(1) ), ( DATA(74), EIMHT(1) ), ( DATA(81), DT102400
F CIMPL(1) ), ( DATA(88), WCLRS(1) ), DT102500
G ( DATA(94), LATT(1,1), ILATT(1,1) ), ( DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1) ), ( DATA(2182), DESGN(1,1), IDESGN(1,1) ), DT102700
I ( DATA(2682), SUBST(1), ISUBST(1) ), ( DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1) ), ( DATA(3483), NCOND ) DT102900
EQUIVALENCE ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS ) DT103000
INTEGER CNTRL, AFLAG, DATE, ATRGGR DT201000
REAL LATTC DT202000
C
COMMON / IMFATT / NMISS, NRFLCT, NVIGN, NRAYSB, XBAR, YBAR, XSTAR, L3001000
1 YSTAR, RSTAR, SJSTAR, XCURL(20), YCURL(20), MXBAR, KYBAR, L3001200
2 MYBAR, KYBAR, AX, BX, CX, AY, BY, CY, ACAP, BCAP, CCAF L3001400
DIMENSION MXCURL(200), MYCURL(200) L3001600
EQUIVALENCE ( XCURL(1), MXCURL(1) ), ( YCURL(1), MYCURL(1) ) L3001800
REAL MXBAR, KYBAR, MYBAR, KYBAR, MXCURL, MYCURL L3002000
C
COMMON / AZOBJ / THTR, SHTHTR, CSTHTR A2004000
C
COMMON / INDESH / PERIOD, PCOUNT, NOFFND, JAIL(13,10), SENTN(10), L7001000
1 DIMVCT, CORD(6), SIZE, DETRM, ERLNG, ESLNG, IMPRV, VCTSTR(6000), L7002000
2 VINDX(11), DJOLD(7) L7022100
REAL IMPRV L7003000

```

```

SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR
MEVCTZ - EFN SOURCE STATEMENT - IFN(S) -
                                         09/24/68
INTEGER PERIOD, PCOUNT, SENTH, DIMVCT, CO3D, VINDX
                                         L7004000
C
DIMENSION QVCT(3), XVCT(3), EVCTOR(3), XIDAR(6), YIDAR(6)
1 ,TEMPS(400), KXCURL(200), KYCURL(200)
EQUIVALENCE ( TEMPS(1), KXCURL(1) ), ( TEMPS(201), KYCURL(1) )
INTEGER SMISS
REAL KXCURL, KYCURL, MXSUM, KXSUM, MYSUM, KYSUM, LAMDA
                                         EV002000
                                         EV002200
                                         EV002300
                                         EV002400
                                         EV002600
C
START OF SUBROUTINE MEVCTR
                                         EV003000
C
L = 0
                                         EV003200
200 IF (KFLAG3 .NE. 0 ) LAMDA = 2. * GAUSS ** 2
                                         EV003300
CALL GETRID(BF1)
                                         EV003400
IF (WF.CH .EQ. 0. ) GO TO 1000
                                         EV003600
THTR = 3.1415926
                                         EV003800
SINTHTR = 0.
                                         EV004000
CSTHTR = -1.
                                         EV004200
HEIGHT = EPRAD * OMGA2
                                         EV004400
XOHAT = 0.
                                         EV004600
YOHAT = OMGA2
                                         EV005000
C
TRACE RAY
                                         EV005200
CALL RAYTR ( XOHAT, YOHAT, HEIGHT, 1, XVCT, QVCT, SMISS, $5000,
1 $5000, $500, 0, DUMMY, DUMMY )
                                         EV005400
IF ( QVCT(3) .EQ. 0. ) GO TO 5000
                                         EV005600
XOHAT = QVCT(2) / ( QVCT(3) * HEIGHT )
                                         EV005800
IF ( SURFC(5,NSURF+1) .GT. 0. ) XOHAT = -XOHAT
                                         EV006000
IF ( FLNGH .NE. 0. ) XOHAT = XOHAT - (1. / FLNGH )
                                         EV006200
L = L + 1
                                         EV006400
EVCTOR(L) = VCTSTR(L) * XOHAT
                                         EV006600
1000 IF ( WEXFP .EQ. 0. ) GO TO 1500
                                         EV006800
THTR = 0.
                                         EV007000
SINTHTR = 0.
                                         EV007200
CSTHTR = 1.
                                         EV007400
HEIGHT = HEXFP
                                         EV007600
XOHAT = 0.
                                         EV007800
YOHAT = 0.
                                         EV008000
                                         EV008200
C
TRACE RAY
                                         EV201000
CALL RAYTR ( XOHAT, YOHAT, HEIGHT, 1, XVCT, QVCT, SMISS, $5000,
1 $5000, $1200, 0, DUMMY, DUMMY )
                                         EV201200
IF ( QVCT(3) .EQ. 0. ) GO TO 5000
                                         EV201400
XOHAT = XVCT(2) + ( QVCT(2) * ( DEXFP - XVCT(3) ) ) / QVCT(3)
                                         EV201600
L=L+1
                                         EV201800
EVCTOR(L) = VCTSTR(L) * XOHAT
                                         EV202000
1500 THTR = ROTAN * 3.1415926 / 180.
                                         EV202200
SINTHTR = SIN(THTR)
                                         EV202400
CSTHTR = COS(THTR)
                                         EV202600
MRAYS = NRAYS
                                         EV202800
IF ( LMODE .EQ. 0 ) MRAYS = 2 * MRAYS
                                         EV203000
XNCLRS = NCLRS
                                         EV203200
SSTAR = DELD
                                         EV203400
IF ( IMODE .EQ. 0 ) SSTAR = SSTAR + DF1
                                         EV203500
DO 4300 J = 1, NOBJH
                                         EV203600
HEIGHT = HD + FLOAT( J-1 ) * DELH
                                         EV203800
IF ( KFLAG1 .EQ. 0 ) GO TO 1800
                                         EV203820

```

SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR
MEVCTZ - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

C			
C	RE-COMPUTE D(J) WHILE DESIGNING	EV203830	
	REWIND 11	EV203840	47
	MXSUM = 0.	EV203860	
	KXSUM = 0.	EV203880	
	MYSUM = 0.	EV203900	
	KYSUM = 0.	EV203920	
	DO 1600 I = 1, NCLRS	EV204000	
C			
C	TRACE PATTERN	EV205000	
	CALL SMPOP (I, J, HEIGHT, BF1, 5, TEMPS)	EV205200	50
	IF (NRAYSB .NE. MRAYS) GO TO 5000	EV205400	
	MXSUM = MXSUM + MXBAR	EV205405	
	KXSUM = KXSUM + KXBAR	EV205410	
	MYSUM = MYSUM + MYBAR	EV205415	
	KYSUM = KYSUM + KYBAR	EV205420	
	IF (NCLRS .EQ. 1) GO TO 1700	EV205425	
C			
C	SAVE COORDINATE DATA	EV205427	
	1600 WRITE (11) MXBAR, KXBAR, MYBAR, KYBAR, (MXCURL(IX), KXCURL(IX),	EV205430	
	MYCURL(IX), KYCURL(IX), IX = 1, NRAYSB)	EV205435	58
	END FILE 11	EV205440	66
	REWIND 11	EV205445	67
	MXSUM = MXSUM / XNCLRS	EV205450	
	KXSUM = KXSUM / XNCLRS	EV205455	
	MYSUM = MYSUM / XNCLRS	EV205460	
	KYSUM = KYSUM / XNCLRS	EV205465	
	1700 AJ = MXSUM * SNTHTR + MYSUM * CSTHTR	EV205470	
	BJ = KXSUM * SNTHTR + KYSUM * CSTHTR	EV205475	
	DLTA = BJ + AJ * SSTAR	EV205480	
	ALFA = SURFC(12, NSURF + 1) - 1. - AJ ** 2	EV205485	
	BETA = AJ * DLTA - SURFC(11, NSURF + 1)	EV205490	
	GAMA = DLTA ** 2	EV205495	
	DLTA = BETA ** 2 + ALFA * GAMA	EV205500	
	IF (DLTA) 1775, 1725, 1750	EV205502	
	1725 CIMPL(J) = - GAMA / BETA	EV205504	
	GO TO 1775	EV205506	
	1750 CIMPL(J) = - GAMA / (BETA + SIGN(SQRT(DLTA), BETA))	EV205508	76
	1775 SJSTAR = SSTAR + CIMPL(J)	EV205510	
	1800 CONTINUE	EV205512	
	DO 2500 I = 1, NCLRS	EV205514	
	IF (KFLAG1 .EQ. 0) GO TO 1875	EV205516	
	IF (NCLRS .EQ. 1) GO TO 1825	EV205518	
C			
C	READ STORED COORDINATE DATA	EV205519	
	READ (11) MXBAR, KXBAR, MYBAR, KYBAR, (MXCURL(IX), KXCURL(IX),	EV205520	
	MYCURL(IX), KYCURL(IX), IX = 1, NRAYSB)	EV205522	86
	1825 XBAR = MXBAR * SJSTAR + KXBAR	EV205524	
	YBAR = MYBAR * SJSTAR + KYBAR	EV205526	
	DO 1850 K = 1, NRAYSB	EV205528	
	XCURL(K) = MXCURL(K) * SJSTAR + KXCURL(K)	EV205530	
	1850 YCURL(K) = MYCURL(K) * SJSTAR + KYCURL(K)	EV205532	
	GO TO 1900	EV205534	
C			
C	TRACE PATTERN--DO NOT RE-COMPUTE D(J)	EV205535	
	1875 CALL SMPOP (I, J, HEIGHT, BF1, 2, DUMMY)	EV205536	109

SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR
MEVCTZ - FOR SOURCE STATEMENT - IFN(3) -

09/24/68

PAGE

```

IF ( NRAYSB .NE. MRAYS ) GO TO 5000          EV205530
C
C      COMPUTE WEIGHTED SPOTS                EV205541
1900 ALFA = 1.                                EV205550
      DO 2000 K = 1, NRAYSB                   EV205600
      IF ( KFLAG3 .EQ. 0 ) GO TO 1975         EV205650
C
C      USE GAUSSIAN WEIGHTING SCHEME         EV205660
      BETA = WXDIR ** 2 * XCURL(K) ** 2 + WYDIR ** 2 * YCURL(K) ** 2 EV205700
      ALFA = EXP( - BETA / LAMDA )            EV205750      124
1975 L = L + 1                                EV205800
      EVCTOR(L) = VCTSTR(L) * XCURL(K) * ALFA EV206000
      L = L + 1                                EV206200
2000 EVCTOR(L) = VCTSTR(L) * YCURL(K) * ALFA EV206400
      XIBAR(1) = XBAR                          EV206600
2500 YIBAR(1) = YBAR                          EV206800
      XBAR = 0.                                EV207000
      YBAR = 0.                                EV207200
      DO 2800 I = 1, NCLRS                      EV207400
2800 YBAR = YBAR + YIBAR(I)                   EV207600
      YBAR = YBAR / XNCLRS                     EV207800
      DO 3000 I = 1, NCLRS                      EV208000
      L = L + 1                                EV208200
3000 EVCTOR(L) = VCTSTR(L) * ( YIBAR(I) - YBAR ) EV208400
      IF ( LMODE .EQ. 1 ) GO TO 4000           EV208600
      DO 3500 I = 1, NCLRS                      EV208800
3500 XBAR = XBAR + XIBAR(I)                   EV209000
      XBAR = XBAR / XNCLRS                     EV401000
      DO 3800 I = 1, NCLRS                      EV401200
      L = L + 1                                EV401400
3800 EVCTOR(L) = VCTSTR(L) * ( XIBAR(I) - XBAR ) EV401600
C
C      COMPUTE WEIGHTED IMAGE HEIGHT DEVIATIONS EV402000
4000 XGHAT = XBAR * SNTHTR + YBAR * CSTHTR    EV402200
      L = L + 1                                EV402400
4300 EVCTOR(L) = VCTSTR(L) * ( XGHAT - EIMHT(J) ) EV402600
      CALL PRMSUB ( $200 )                      EV402800      173
      IF ( L .EQ. DIMVCT ) RETURN              EV403000
      WRITE ( 6 , 32010 ) L, DIMVCT           EV403200      177
32010 FORMAT ( 1H0, 9X, 36HNUMBER OF COMPONENTS IN ERROR VECTOR, 15,29H EV403400
1DOES NOT AGREE WITH DIMVCT =, 15)          EV403600
      CALL GOOF                                  EV403800      178
5000 CALL PRMSUR ( $5000 )                    EV404000
C
C      RETURN 1                                EV405000
      END                                        EV406000

```

SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR
 MEVCTR

STORAGE MAP

09-24/68

PAGE 5

SUBROUTINE MEVCTR
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	00671	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCLD	00032	R	DATA	00033	R
UMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUDP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NDBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
HXFP	00055	R	DEXFP	00056	R	WEXFP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
FLD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HD	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAH	00077	R	NDSGN	00100	I	NDSGV	00104	I
WDBJH	00117	R	WCLRH	00120	R	WIMHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	BOUNDS	04240	I
SUBST	05224	R	ISUBST	05224	I	IDESGN	06210	R
IDNDS	06210	I	NCOND	06665	I	ATRGGR	06666	I
GAUSS	06667	R						

COMMON BLOCK			THPATT	ORIGIN	06672	LENGTH	06651	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NMISS	00000	I	NRFLCT	00001	I	NVIGN	00002	I
NRAYS0	00003	I	XBAR	00004	R	YBAR	00005	R
XSTAR	00006	R	YSTAR	00007	R	RSTAR	00010	R
SJSTAR	00011	R	XCURL	00012	R	YCURL	00323	R
KXBAR	00634	R	KXBAR	00635	R	MYBAR	00636	R
KYBAR	00637	R	AX	00640	R	BX	00641	R
CX	00642	R	AY	00643	R	BY	00644	R
CY	00645	R	ACAP	00646	R	BCAP	00647	R
CCAP	00650	R	MXCURL	00012	R	MYCURL	00323	R

COMMON BLOCK			AZOBJ	ORIGIN	07543	LENGTH	00003	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
THTR	00000	R	SNTHTR	00001	R	ESTHTR	00002	R

COMMON BLOCK			TMDESN	ORIGIN	07546	LENGTH	13777	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
PERIOD	00000	I	PCOUNT	00001	I	NOFFND	00002	I
JAIL	00003	I	SENTN	00147	I	DIMVCT	00161	I
ORD	00162	I	SIZE	00170	R	DETRM	00171	R
EOLNG	00172	R	ESLNG	00173	R	IMPRV	00174	R
VCTSTR	00175	R	VINDX	13755	I	DJOLD	13770	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
--------	----------	------	--------	----------	------	--------	----------	------

SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR
MEVCTZ

09/24/68

PAGE 6

STORAGE MAP

OVCT	24365	R	XVCT	24370	R	XIBAR	24373	R
YIBAR	24401	R	TEMPS	23545	R	KXCURL	23545	R
KXCURL	24655	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MISS	24407	I	MXSUM	24410	R	KXSUM	24411	R
MYSUM	24412	R	KYSUM	24413	R	LAMDA	24414	R
L	24415	I	BF1	24416	R	HEIGHT	24417	R
XDHAT	24420	R	YDHAT	24421	R	DUMMY	24422	R
MRAYS	24421	I	XNCLRS	24424	R	SSTAR	24425	R
J	24426	I	I	24427	I	AJ	24430	R
BJ	24431	R	DLTA	24432	R	ALFA	24433	R
DELTA	24434	R	GAMA	24435	R			

ENTRY POINTS

MEVCTR SECTION 11

SUBROUTINES CALLED

GETRHD	SECTION	12	RAYTR	SECTION	13	SIN	SECTION	1
COS	SECTION	15	.FRWT.	SECTION	16	SMPOP	SECTION	1
.FWRD.	SECTION	18	.FEFT.	SECTION	19	SQRT	SECTION	2
.FRDB.	SECTION	21	EXP	SECTION	22	FRMSUB	SECTION	2
.FWRD.	SECTION	24	GOOF	SECTION	25	.UN11.	SECTION	2
.FWLR.	SECTION	27	.FBLT.	SECTION	28	.FBDT.	SECTION	2
.FRLR.	SECTION	30	.UN06.	SECTION	31	.FFIL.	SECTION	3
.FCNV.	SECTION	33	E.1	SECTION	34	E.2	SECTION	3
E.3	SECTION	36	E.4	SECTION	37	CC.1	SECTION	3
CC.2	SECTION	39	CC.3	SECTION	40	CC.4	SECTION	4
BYSLOC	SECTION	42	.FRET.	SECTION	43			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
200	2A	24500	1000	23A	24616	5000	179A	25650
500	12A	24553	1500	34A	24675	1200	28A	24652
4300	167A	25606	1800	80A	25240	1600	58A	25046
1700	68A	25127	1775	78A	25235	1725	72A	25210
1750	75A	25216	2500	136A	25462	1875	108A	25340
1825	56A	25311	1850	102A	25331	1900	114A	25355
2000	130A	25443	1975	125A	25422	2800	141A	25474
3000	140A	25513	4000	160A	25574	3500	156A	25540
3800	161A	25557	32010	FORMAT	24457			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 25733.

TR-67-700-10-2
Page III-5-47
Program Listing
Link 5

IMP.M010001.44800-0.77387.AM IC IBJOB L10MK1 7094 A 02/16/68 PAGE 6

BT
BTITLE SUBROUTINE TO COMPUTE DESIGN PARAMETER INCREMENTS M0001000
BTBFC M000LZ M94.XR7 LINK 5 (M000L)

SUBROUTINE TO COMPUTE DESIGN PARAMETER INCREMENTS
 MODDLZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 1

```

SUBROUTINE MODDL                                MOD01500
C
C *** INPUTS ***                                MOD01600
C NDSGV VECTOR CONTAINING CURRENT DESIGN VARIABLE NUMBERS.(DATA COMMON)MOD01610
C DESGN MATRIX OF DESIGN VALUES.              (DATA COMMON)MOD01620
C
C *** OUTPUTS ***                                MOD01630
C DESGN THE INCREMENT IN DESGN(I,J), FOR I A DESIGN VARIABLE NUMBER MOD01640
C CONTAINED IN NDSGV.                          (DATA COMMON) MOD01650
C
C *** DESCRIPTION ***                             MOD01660
C MODDLZ USES DESGN(I,J), I=2,3,10 TO CALCULATE A NEW INCREMENT MOD01670
C STORED IN DESGN(I,J). THIS PROCESS IS REPEATED FOR EACH J MOD01680
C DESIGN VARIABLE CONTAINED IN NDSGV.          MOD01690
C
COMMON / DATA / NCNTRL, CONTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483)                                     DT001100
DIMENSION WOBXH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATTC(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATTC(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMGA2), (DATA(24), DT101700
8 OMGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), HO), (DATA(33), DT102000
B DELH), (DATA(34), SYSMX), (DATA(35), WXDIR), (DATA(36), DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WOBXH(1)), (DATA(60), WCLRH(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATTC(1,1), ILATTC(1,1)), (DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
INTEGER CONTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
C START OF SUBROUTINE MODDL                                MOD02000
C
C
C IMAX = NDSGV(1)                                          MOD02200
DO 500 I = 1, IMAX                                       MOD02400
IZ = NDSGV(I+1)                                          MOD02600
IF ( DESGN(10,IZ) .NE. 0.) DESGN(3,IZ) = ABS( DESGN(3,IZ) * SQRT(MOD02800
1 DESGN(2,IZ) / DESGN(10,IZ) ) ) MOD03000
500 CONTINUE                                            MOD03200
RETURN                                                  MOD04000
END                                                    MOD04200

```


SUBROUTINE TO COMPUTE DESIGN PARAMETER INCREMENTS
 MODDLZ STORAGE MAP

02/18/68

PAGE 2

SUBROUTINE MODDL
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
CONTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00039	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	N:PLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJM	00045	I	NSURF	00046	I
AFLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HO	00072	R	DELH	00073	R
SYBNX	00074	R	WXDIR	00075	R	WYDIR	00076	R
RTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WOBJM	00117	R	WCLRH	00126	R	WINHT	00139	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	08210	R
IBNDS	08210	I	NCOND	08665	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MAX	08667	I	IZ	08670	I			

ENTRY POINTS

MODDL SECTION 5

SUBROUTINES CALLED

SGRT SECTION 6

SYSLOC SECTION 7
 EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
500	16A	08742						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 08763.

TR-67-700-10-2
Page III-5-50
Program Listing
Link 5

.ND10001.44800-0.77387.AM IC

18JOB LINK1 7094 A 02/16/68

PAGE 3

30
BTITLE SUBROUTINE TO GENERATE WEIGHT VECTOR
SISFIC MMVCTZ N94.XR7 LINK 5 (MMVCTN)

WV001000

SUBROUTINE TO GENERATE WEIGHT VECTOR
MVCTZ - EFN SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 1

```

SUBROUTINE MVCTR                                WV001200
C
C *** INPUTS ***                                WV001600
C   VARIOUS DESIGN WEIGHTS.                    (DATA COMMON) WV001610
C
C *** OUTPUTS ***                                WV001620
C   VCTSTR WEIGHT VECTOR OF LENGTH DIMVCT.      (TMDESN COMMON) WV001630
C
C *** DESCRIPTION ***                            WV001640
C   MVCTR IS USED TO GENERATE THE WEIGHT VECTOR IN VCTSTR, WHOSE WV001650
C   LENGTH IS DIMVCT, FROM THE DESIGN WEIGHTS IN DATA COMMON. WV001660
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483)                                       DT001100
DIMENSION NOBJH(7), WCLR(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLR(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMGA2), (DATA(24), DT101700
8 OMGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), MO), (DATA(33), DT102000
B DELH), (DATA(34), SYSMX), (DATA(35), WDIR), (DATA(36), DT102100
C WDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), NOBJH(1)), (DATA(60), WCLR(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLR(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2192), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(382), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATTC DT202000
C
COMMON / TMDESN / PERIOD, PCOUNT, NOFFND, JAIL(10,10), SENTN(10), L7001000
1 DIMVCT, CORD(6), SIZE, DETRM, EDLNG, ESLNG, IMPRV, VCTSTR(6000), L7002000
2 VINDX(11) L7002100
REAL IMPRV L7003000
INTEGER PERIOD, PCOUNT, SENTN, DIMVCT, CORD, VINDX L7004000
C
C   START OF MVCTR                                WV002000
C
C   L = 0                                           WV002200
1000 IF (WFLGH .EQ. 0.0) GO TO 2000 WV002400
L = L + 1 WV002600
VCTSTR(L) = WFLGH WV002800
2000 IF (WEXPP .EQ. 0.0) GO TO 2500 WV003000
L = L + 1 WV003200

```

SUBROUTINE TO GENERATE WEIGHT VECTOR
 MWVCTR - EFN SOURCE STATEMENT - IFN(B) -

02/16/68

PAGE 2

VCTSTR(L) = WEXPP	WV003400	
2500 MRAYS = MRAYS	WV003600	
IF (LMODE .EQ. 0) MRAYS = MRAYS + MRAYS	WV003800	
ASQRT = SQRT(FLOAT(MRAYS))	WV004000	16
BSQRT = SQRT(FLOAT(NCLRS))	WV004200	17
IMAX = NCLRS	WV004400	
IF (LMODE .NE. 1) IMAX = IMAX + IMAX	WV004600	
C		
BEGIN DO LOOP ON OBJECT HEIGHTS	WV005000	
C		
DO 4000 J = 1, NOBJH	WV005200	
TX = WOBJH(J) * WXDIR / ASQRT	WV005400	
TY = WOBJH(J) * WYDIR / ASQRT	WV005600	
C		
BEGIN DO LOOP ON NUMBER OF COLORS	WV006000	
C		
DO 3000 I = 1, NCLRS	WV006200	
TXX = WCLRS(I) * TX	WV006400	
TYX = WCLRS(I) * TY	WV006600	
C		
BEGIN DO LOOP ON RAY NUMBER	WV008000	
C		
DO 3000 K = 1, MRAYS	WV007000	
L = L + 1	WV007200	
VCTSTR(L) = TXX	WV007400	
L = L + 1	WV007600	
VCTSTR(L) = TYX	WV007800	
L = L + 1	WV008000	
3000 CONTINUE	WV008200	
TXX = WCLRH(J) / BSQRT	WV008400	
DO 3500 I = 1, IMAX	WV008600	
L = L + 1	WV008800	
3500 VCTSTR(L) = TXX	WV009000	
L = L + 1	WV009200	
VCTSTR(L) = WINHT(J)	WV009400	
4000 CONTINUE		
C		
CALL PRMSUB(1000)	WV101000	51
IF (L .EQ. DIMVCT) GO TO 32000	WV101200	
C		
ERROR - TERMINATE RUN	WV102000	
C		
WRITE (6, 32000) L, DIMVCT	WV102200	56
32100 FORMAT(1H0, 37HNUMBER OF COMPONENTS IN WEIGHT VECTOR, 15, 29H DOES	WV102400	
15 NOT AGREE WITH DIMVCT =, 15)	WV102600	
CALL GOOP	WV103000	
C		
END OF MWVCTR SUBROUTINE	WV103800	57
C		
32000 RETURN	WV106000	
END	WV107000	

SUBROUTINE TO GENERATE WEIGHT VECTOR
 MVCTR

02/16/68

PAGE 3

STORAGE MAP

SUBROUTINE MVCTR
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ICNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRJ	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
#FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
ICXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGAR	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAP	00066	R	SPFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HO	00072	R	DELH	00073	R
SYMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
NDYAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WOBJH	00117	R	WCLRH	00126	R	WINHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00120	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I			

COMMON BLOCK			THDEEN	ORIGIN	08667	LENGTH	13770	
PERIOD	LOCATION	TYPE	PCOUNT	LOCATION	TYPE	NOFFND	LOCATION	TYPE
JAIL	00003	I	SENTN	00147	I	DIMVCT	00161	I
ORD	00162	I	SIZE	00170	R	DETRN	00171	R
EDLNG	00172	R	ESLNG	00173	R	IMPRV	00174	R
VCTSTR	00175	R	VINDX	13755	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
L	22657	I	NRAYS	22660	I	ASQRT	22661	R
BSQRT	22662	R	IMAX	22663	I	J	22664	I
TX	22665	R	TY	22666	R	I	22667	I
TXX	22670	R	TYT	22671	R			

ENTRY POINTS

MVCTR SECTION 7

SUBROUTINES CALLED

SQRT	SECTION	8	PRMSUB	SECTION	9	.FWRD.	SECTION	10
GOOF	SECTION	11	.UNDS.	SECTION	12	.FFIL.	SECTION	13

SUBROUTINE TO GENERATE WEIGHT VECTOR
 MMVCTZ

02/16/68

PAGE 4

STORAGE MAP

.FCNV. SECTION 14
 E.3 SECTION 17
 CC.2 SECTION 20
 SYSLOC SECTION 23

E.1 SECTION 15
 E.4 SECTION 18
 CC.3 SECTION 21

E.2 SECTION 16
 CC.1 SECTION 19
 CC.4 SECTION 22

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	2A	22723	2000	7A	22738	2900	12A	22747
4000	40A	23134	3000	36A	23102	3500	42A	23119
32000	90A	23163	32100	FORMAT	22702			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 23207.

TR-67-700-10-2
Page III-5-55
Program Listing
Link 5
Page Revised 10/1/68

INF,NO10001,44903-0,77367,AM IC

18JOB L10MKC 7094 A

09/24/68

PAGE

82

BTITLE

SUBROUTINE TO CONTROL DESIGN LOGIC

31DFTC STAR22

M94,XR7

LINK 5 (STAR2)

92001000

SUBROUTINE TO CONTROL DESIGN LOGIC
 STAR2Z - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

```

SUBROUTINE STAR2                                S2001200
C
C *** INPUTS ***                                S2001600
C NDSGN VECTOR DESCRIBING DESIGN ITERATIONS.   (DATA COMMON) S2001610
C NDSGV VECTOR DEFINING DESIGN VARIABLES.      (DATA COMMON) S2001620
C NJAIL JAIL SENTENCE.                        (DATA COMMON) S2001630
C DJOLD TEMPORARY STORAGE FOR D(J)           (TMDESN COMMON) S2001635
C
C *** DESCRIPTION ***                            S2001640
C STAR2 TESTS THE DESIGN DATA FOR VALIDITY, SETS UP INITIAL S2001650
C VALUES, AND CALCULATES THE WEIGHT VECTOR. THE CONTROL LOOP IS S2001660
C EXECUTED FOR EACH ITERATION AND CONSISTS OF A CALL TO JUDGE, S2001670
C A CALL TO CYCLE, AND THE SETUP OF THE NEXT DESIGN VARIABLE SET. S2001680
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), FUNCID, DT001000
1 DATA(3485)                                     DT001100
DIMENSION WOBJH(7), WCLR(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3), DT101000
1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6), DT101100
2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9), DT101200
3 IMODE ), ( DATA(10), NSFLN ), ( DATA(11), NOBJH ), ( DATA(12), DT101300
4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15), DT101400
5 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ), ( DATA(18), DT101500
6 ZETA ), ( DATA(19), HEXFF ), ( DATA(20), DEXFF ), ( DATA(21), DT101600
7 WEXFF ), ( DATA(22), DLPLN ), ( DATA(23), OMGA2 ), ( DATA(24), DT101700
8 OMGA1 ), ( DATA(25), DELD ), ( DATA(26), EPRAD ), ( DATA(27), DT101800
9 PSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30), DT101900
A DELH ), ( DATA(31), ODIST ), ( DATA(32), HD ), ( DATA(33), DT102000
B DUM ), ( DATA(34), SYSMX ), ( DATA(35), WYDIR ), ( DATA(36), DT102100
C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42), DT102200
D NDSGV(1) ), ( DATA(53), WOBJH(1) ), ( DATA(60), WCLR(1) ), DT102300
E ( DATA(67), WIMHT(1) ), ( DATA(74), EIMHT(1) ), ( DATA(81), DT102400
F CIMPL(1) ), ( DATA(88), WCLRS(1) ), DT102500
G ( DATA(94), LATT(1,1), ILATT(1,1) ), ( DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1) ), ( DATA(2182), DESGN(1,1), IDESGN(1,1) ), DT102700
I ( DATA(2682), SUBST(1), ISUBST(1) ), ( DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1) ), ( DATA(3483), NCOND ) DT102900
EQUIVALENCE ( DATA(3484), ATRGCR ), ( DATA(3485), GAUSS ) DT103000
INTEGER CNTRL, AFLAG, DATE, ATRGCR DT201000
REAL LATT DT202000
C
COMMON/PRNT/LINE,PAGE                             FT002500
INTEGER PAGE FT002700
C
COMMON / TMDESN / PERIOD, PCOUNT, NOFFND, JAIL(10,10), SENTN(10), L7001000
1 DIMVCT, CORD(6), SIZE, DETRM, EGLNG, ESLNG, IMPRV, VCTSTR(6000), L7002000
2 VINDX(11), DJOLD(7)                             L7022100
REAL IMPRV                                         L7003000
INTEGER PERIOD, PCOUNT, SENTN, DIMVCT, CORD, VINDX L7004000
C
IF(NDSGN(4).LE.0) RETURN                          S2002000
IF(NDSGV(1).GT.0) GO TO 40                        S2002200

```


SUBROUTINE TO CONTROL DESIGN LOGIC		09/24/68	PAGE
STAR2Z - EFN SOURCE STATEMENT - IFN(5) -			
	WRITE (6,600)	\$2002400	6
600	FORMAT(2PHIBAD DESIGN DATA INPUT)	\$2002600	
	CALL GOOF	\$2002800	7
40	DIMVCT=0	\$2003000	
50	MDIM=0	\$2003200	
	NRAYS=NRAYS	\$2003400	
	IF (LMODE.EQ.0) NRAYS=NRAYS+NRAYS	\$2003600	
	MDIM=2*NRAYS+NCLRS+NCLRS+1	\$2003800	
	IF (LMODE.NE.1) MDIM=MDIM+NCLRS	\$2004000	
	MDIM=MDIM+NOBJH	\$2004200	
	IF (WFLGH.NE.0.) MDIM=MDIM+1	\$2004400	
	IF (WEXPP.NE.0.) MDIM=MDIM+1	\$2004600	
	DIMVCT=DIMVCT+MDIM	\$2004800	
	CALL PRMSUB(\$50)	\$2005000	23
	IMAX=NDSGV(1)	\$2005200	
	MDIM=IMAX+2	\$2005400	
	IF (MDIM#DIMVCT.LE.6000) GO TO 170	\$2005600	
	WRITE (6,601) DIMVCT	\$2005800	30
601	FORMAT(0HIDIMVCT=110,28H PROBLEM TOO BIG FOR MACHINE)	\$2006000	
	CALL GOOF	\$2006200	31
170	VINDX(1)=DIMVCT+1	\$2006400	
	DO 210 I=2,11	\$2006600	
210	VINDX(I)=VINDX(I-1)+DIMVCT	\$2006800	
	PERIOD=2*(NDSGN(3)-NDSGN(2)+1)	\$2007000	
	PCOUNT=0	\$2007200	
	NDFND=0	\$2007400	
	DO 220 I=1,10	\$2007600	
	SENTN(I)=0	\$2007800	
	DO 220 J=1,10	\$2008000	
220	JAIL(I,J)=0	\$2008200	
	CALL MWVCTR	\$2008400	55
	CALL MEVCTR(VCTSTR(DIMVCT+1), \$260, ATRGGR, GAUSS)	\$2008600	58
	IF (NSUBT .NE. 0) GO TO 270	\$2008800	
	DO 230 II = 1, NOBJH	\$2009000	
230	DJOLD(II)=CIMPL(II)	\$2009200	
	GO TO 270	\$2009400	
250	WRITE (6,602)	\$2021000	73
602	FORMAT(1H136HRAY FAILURE DURING COMPUTATION OF ED)	\$2021200	
	CALL GOOF	\$2021400	74
270	CALL DOTF(VCTSTR(DIMVCT+1), VCTSTR(DIMVCT+1), EDLNG)	\$2021600	78
	EDLNG=SQRT(EDLNG)	\$2021800	80
	DO 280 JJ = 1, NSURF	\$2021900	
	IF (SURFC(5,JJ) .NE. 1.) GO TO 290	\$2021920	
280	CONTINUE	\$2021940	
290	CALL CORDER(NCLRS, SURFC(5,JJ), CORD)	\$2022000	93
	CALL PRCTL	\$2022200	95
	WRITE (6,603)	\$2022400	96
603	FORMAT(1H33X23HDESIGN VARIABLE NUMBERS6X12HINCR. FACTOR3X11HDETERM	\$2022600	
	1INANTSX10HOLD LENGTHSX10HNEW LENGTH4X11HIMPROVEMENT)	\$2022800	
	LINE=LINE+2	\$2023000	
320	CALL JUDGE(\$350)	\$2023200	98
	CALL CYCLE(\$420)	\$2023400	101
	NDSGN(4)=NDSGN(4)-1	\$2023600	
350	DO 400 I=1,IMAX	\$2023800	
	IF (NDSGV(1+I) .LT. 0) GO TO 380	\$2024000	
	NDSGV(1+I)=NDSGN(2)	\$2024200	

SUBROUTINE TO CONTROL DESIGN LOGIC		09/24/68	PAGE
STAR2Z - EFN SOURCE STATEMENT - IFN(5) -			
	GO TO 400	\$2024400	
380	NDSGV(1+1)=NDSGV(1+1)+1	\$2024600	
400	CONTINUE	\$2024800	
	IF (NDSGN(4) .GT. 0) GO TO 320	\$2025000	
C			
420	RETURN	\$2025200	
	END	\$2026000	

TR-67-700-10-2
Page III-5-58
Program Listing
Link 5
Page Revised 10/1/68

SUBROUTINE TO CONTROL DESIGN LOGIC
STAR2Z - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

	GO TO 400	\$2024400
300	NDSGV(I+1)=NDSGV(I+1)+1	\$2024600
400	CONTINUE	\$2024800
	IF (NDSGN(4).GT.0)GO TO 320	\$2025000
C		
420	RETURN	\$2025200
	END	\$2026000

SUBROUTINE TO CONTROL DESIGN LOGIC
STAR2Z

STORAGE MAP

09/24/68

PAGE 4

SUBROUTINE STAR2
COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUDF	00041	I	NIFLN	00042	I	IMODE	00043	I
NSFLN	00044	I	NBJH	00045	I	NSURF	00046	I
AFLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
HEXFP	00055	R	DEXFP	00056	R	WEXFP	00057	R
DLFLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	FSCAL	00065	R
OMGAF	00066	R	SFFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HO	00072	R	DELH	00073	R
YSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
NOBJH	00117	R	WCLRH	00126	R	WIMHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IGNDS	06210	I	NCOND	06665	I	ATRCCR	06666	I
GAUSS	06667	R						

LINE	COMMON BLOCK	PRNT	ORIGIN	06672	LENGTH	00002		
	00000	I	PAGE	00001	I			
PERIOD	COMMON BLOCK	TMDESN	ORIGIN	06674	LENGTH	13777		
	00000	I	PCOUNT	00001	I	NDFND	00002	I
JAIL	00003	I	SENTN	00147	I	DIMVCT	00161	I
COND	00162	I	SIZE	00170	R	DETRM	00171	R
ESLNG	00172	R	ESLNG	00173	R	IMPRV	00174	R
VCTSTR	00175	R	VINDX	13755	I	DJOLD	13770	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MDIM	22673	I	MRAYS	22674	I	IMAX	22675	I
I	22676	I	JJ	22677	I			

ENTRY POINTS

STAR2 SECTION 9

SUBROUTINES CALLED

SUBROUTINE TO CONTROL DESIGN LOGIC
 STAR2Z

09/24/68

PAGE 5

STORAGE MAP

.FWRD. SECTION 10
 MWVCTR SECTION 13
 SORT SECTION 16
 JUDGE SECTION 19
 .FFIL. SECTION 22

GOODF SECTION 11
 MEVCTR SECTION 14
 CORDER SECTION 17
 CYCLE SECTION 20
 .FCNV. SECTION 23
 EFN IFN CORRESPONDENCE

PRMSUB SECTION 11
 DDTF SECTION 11
 PRCTL SECTION 11
 .JUN06. SECTION 2
 SYSLOC SECTION 2

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
40	8A	23006	600	FORMAT	22713	50	9A	23007
170	32A	23112	601	FORMAT	22720	210	36A	23116
220	50A	23144	260	73A	23205	270	75A	23216
230	67A	23200	602	FORMAT	22730	280	88A	23253
290	91A	23256	603	FORMAT	22740	320	97A	23306
350	104A	23321	420	123A	23347	400	117A	23340
360	114A	23335						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 23375.

TR-67-700-10-2
Page III-5-61
Program Listing
Link 5

IMP.N010001.44800-0.77387.AM IC

ISJOB LINK: 7004 A 02/15/68

PAGE 6

30
STITLE JPL MATRIX INVERSION LIBRARY SUBROUTINE
SIBPTC HVRT: M94.XR7 LINK 5 (MATINV)

JPL MATRIX INVERSION LIBRARY SUBROUTINE
HVRT1 - EFN SOURCE STATEMENT - IPN(?) -

02/16/68

PAGE 1

```

SUBROUTINE MATINV (NDIM,A,NORDER,B,MORDER,DETERM,S1,S2)
C
C   MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
C   DIMENSION A(NDIM,NDIM),B(NDIM,2)
C
C   PIVOT=S2
C   IPIVOT=S1
C   INDEX (N,1) =S1((N+1)-(2N))
C   DIMENSION S1(2),S2(2)
C   INTEGER Z1,Z2,S1
C   EQUIVALENCE (IROW,JROW), (ICOLUMN,JCOLUMN), (AMAX, T, SWAP)
C
C   INITIALIZATION
C
10 DETERM=1.0
   N=NORDER
   M=MORDER
15 DO 20 J=1,N
20   S1(J)=0
30 DO 50 I=1,N
31   Z1=I+N
32   Z2=Z1+N
C
C   SEARCH FOR PIVOT ELEMENT
C
40 AMAX=0.0
45 DO 105 J=1,N
50   IF (S1(J).EQ.1) GO TO 105
60 DO 100 K=1,N
70   IF (S1(K)-1) 80,100,740
80   IF (ABS(AMAX)-ABS(A(J,K))) 85, 100, 100
85   IROW=J
90   ICOLUMN=K
95   AMAX=A(J,K)
100 CONTINUE
105 CONTINUE
110   S1(ICOLUMN)=S1(ICOLUMN)+1
C
C   INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
130   IF (IROW.EQ.ICOLUMN) GO TO 260
140   DETERM=-DETERM
150   DO 200 L=1,N
160   SWAP=A(IROW,L)
170   A(IROW,L)=A(ICOLUMN,L)
200   A(ICOLUMN,L)=SWAP
205   IF (M.LE.0) GO TO 260
210   DO 250 L=1,M
220   SWAP=B(IROW,L)
230   B(IROW,L)=B(ICOLUMN,L)
250   B(ICOLUMN,L)=SWAP
260   S1(Z1)=IROW
270   S1(Z2)=ICOLUMN
310   S2(I)=A(ICOLUMN,ICOLUMN)

```

JPL MATRIX INVERSION LIBRARY SUBROUTINE
MVRT1 - EPH SOURCE STATEMENT - IPN(8) -

08/16/68

PAGE 2

```
320 DETERM=DETERM08(I)
C
C   DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
330 A(I COLUMN, I COLUMN)=1.0
340 DO 350 L=1, N
350 A(I COLUMN, L)=A(I COLUMN, L)/B2(I)
360 IF (N.LE.0) GO TO 380
370 DO 370 L=1, N
370 B(I COLUMN, L)=B(I COLUMN, L)/B2(I)
C
C   REDUCE NON-PIVOT ROWS
C
380 DO 550 L=1, N
390 IF (L1.EQ. I COLUMN) GO TO 550
400 T=A(L1, I COLUMN)
420 A(L1, I COLUMN)=0.0
430 DO 450 L=1, N
450 A(L1, L)=A(L1, L)-A(I COLUMN, L)*T
460 IF (N.LE.0) GO TO 550
480 DO 500 L=1, N
500 B(L1, L)=B(L1, L)-B(I COLUMN, L)*T
550 CONTINUE
C
C   INTERCHANGE COLUMNS
C
600 DO 710 I=1, N
610 L=N+1-I
611 L1=L+N
612 L2=L1+N
620 IF (B1(L1).EQ.B1(L2)) GO TO 710
630 JROW=B1(L1)
640 JCOLUMN=B1(L2)
650 DO 705 K=1, N
660 SWAP=A(K, JROW)
670 A(K, JROW)=A(K, JCOLUMN)
700 A(K, JCOLUMN)=SWAP
705 CONTINUE
710 CONTINUE
740 RETURN
END
```

JPL MATRIX INVERSION LIBRARY SUBROUTINE
 NVRTI

STORAGE MAP

02/10/68

PAGE 3

SUBROUTINE MATINV
 UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
Z1	00004	I	Z2	00008	I	IRCM	00001	I
JROW	00001	I	JCOLM	00002	I	JCOLM	00002	I
ANAX	00003	R	T	00003	R	SWAP	00003	I
N	00004	I	M	00007	I	J	00010	I
I	00011	I	K	00012	I	L	00013	I
LI	00014	I	LP	00015	I			

ENTRY POINTS

MATINV SECTION 3

SUBROUTINES CALLED

SYBLOC SECTION 4

EPN IFN CORRESPONDENCE

EPN	IFN	LOCATION	EPN	IFN	LOCATION	EPN	IFN	LOCATION
10	1A	00031	15	3A	00037	20	7A	00043
30	11A	00046	50	133A	00048	31	19A	00129
32	16A	00130	40	17A	00133	45	18A	00134
105	40A	00203	30	21A	00137	60	25A	00143
100	36A	00200	70	29A	00153	80	31A	00157
740	100A	00271	85	34A	00172	90	35A	00174
80	36A	00176	110	42A	00208	130	45A	00214
850	74A	00275	140	48A	00220	150	49A	00222
200	57A	00240	160	52A	00234	170	54A	00236
205	60A	00245	217	63A	00252	250	71A	00270
220	66A	00264	230	68A	00266	270	6A	00302
310	76A	00307	320	61A	00317	330	63A	00322
340	85A	00332	350	66A	00343	355	63A	00351
380	104A	00375	360	66A	00356	370	69A	00367
390	107A	00415	400	110A	00421	420	112A	00423
430	114A	00424	450	117A	00432	455	122A	00442
460	125A	00447	500	129A	00455	600	137A	00471
710	165A	00567	610	141A	00502	611	142A	00506
612	143A	00511	620	144A	00514	630	149A	00526
640	151A	00533	650	153A	00540	705	163A	00565
650	156A	00557	670	159A	00561	700	161A	00563

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00753.

TR-67-700-10-2
Page III-6-1
Program Listing
Link 6

Section 6

LINK 6

TR-67-700-10-2
Page III-6-2
Program Listing
Link 6

INF.N010001.44800-0.77387.AM IC ISJOB LINK1 7084 A 02/18/68 PAGE 4

30
BTITLE SUBROUTINE TO COMPUTE FOCAL LENGTH AND FOCAL POINT
SIBPTC PARAXZ H94.XR7 LINK 6 (PARAX) PRO01000

SUBROUTINE TO COMPUTE FOCAL LENGTH AND FOCAL POINT
PARAXZ - EFM SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 1

```

SUBROUTINE PARAX( COLOR, FOCALL, FOCALP )
C
C *** INPUTS ***
C COLOR NUMBER OF COLOR TO BE USED.
C
C *** OUTPUTS ***
C FOCALL FOCAL LENGTH VALUE.
C FOCALP FOCAL POINT VALUE.
C
C *** DESCRIPTION ***
C THIS SUBROUTINE COMPUTES THE FOCAL LENGTH AND FOCAL POINT
C FOR THE SPECIFIED COLOR. IF THEY ARE INFINITE, THEN FOCALL
C AND FOCALP ARE SET TO ZERO.
C
COMMON / DATA / DUMMY1(27), DATA(3483)
EQUIVALENCE ( DATA(23), OMCAR ), ( DATA(26), EPRAD ), ( DATA(12),
1 NSURF ), ( DATA(182), SURFC(1,1) )
DIMENSION SURFC(20,100)
C
COMMON/AZOB1/THTR,SNTHTR,C8THTR
C
INTEGER COLOR
DIMENSION XVCT(3), QVCT(3)
C
C START OF PARAX SUBROUTINE
C
H = EPRAD * OMCAR
FOCALL = 0.0
FOCALP = 0.0
CALL RAYTR( D., OMCAR, H, COLOR, XVCT, QVCT, SWISS, $400, $500,
1 $600, 0, DUMMY, DUMMY )
GO TO 600
400 WRITE ( 6, 32000 ) SWISS
32000 FORMAT( 1H0, 27MPARAXIAL RAY MISSED SURFACE, I4 )
CALL GOOF
500 WRITE ( 6, 32010 ) SWISS
32010 FORMAT( 1H0, 33MPARAXIAL RAY REFLECTED AT SURFACE, I4 )
CALL GOOF
600 IF( QVCT(2) .EQ. 0.0 ) GO TO 1000
ALPHA = - QVCT(3) / QVCT(2)
FOCALP = ALPHA * XVCT(2) + XVCT(3)
FOCALL = ALPHA * H
IF( SURFC( COLOR+4, NSURF+1 ) .LE. 0.0 ) FOCALL = - FOCALL
C
1000 RETURN
END

```

PRO01100

PRO01200

PRO01210

PRO01300

PRO01310

PRO01320

PRO01400

PRO01410

PRO01420

PRO01430

PRO01500

PRO01700

PRO01800

PRO02000

AZ004000

PRO02000

PRO02200

PRO03000

PRO03500

PRO03700

PRO03900

PRO04000

PRO04100

2

PRO04500

PRO05000

3

PRO05200

PRO05400

6

PRO06000

7

PRO06200

PRO06400

8

PRO07000

PRO07200

PRO07400

PRO07600

PRO08000

PRO09000

PRO09200

SUBROUTINE TO COMPUTE FOCAL LENGTH AND FOCAL POINT
 PARAXZ STORAGE MAP

08/18/68

PAGE 2

SUBROUTINE PARAX
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	08666		
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DUMMY1	00000	R		DATA	00033	R	OMSA2	00081	R
EPHAD	00084	R		NSURF	00046	I	SURFC	00320	R
COMMON BLOCK			A20BJ	ORIGIN	08667	LENGTH	00003		
THTR	00000	R		ENTHTR	00001	R	CSTHTR	00002	R
DIMENSIONED PROGRAM VARIABLES									
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
XVCT	08672	R		GVCT	08678	R			
UNDIMENSIONED PROGRAM VARIABLES									
SYMBOL	LOCATION	TYPE		SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
H	08700	R		SHISS	08701	R	DUMMY	08702	R
ALPHA	08703	R							

ENTRY POINTS

PARAX	SECTION	7
RAYTR	SECTION	8
.UNDS.	SECTION	11
SYLOC	SECTION	14

SUBROUTINES CALLED

.FWRD.	SECTION	9	GOOF	SECTION	10
.FF/L.	SECTION	12	.FCNV.	SECTION	13

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
400	5A	08763	500	7A	08776	600	9A	07011
32000	FORMAT	08714	32010	FORMAT	08724	1000	15A	07043

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07075.

TR-67-700-10-2
Page III-6-5
Program Listing
Link 6
Page Revised 10/1/68

INF,ND10001,44903-D,77387,AM IC 1BJOB L10MKC 7094 A 09/24/68 PAGE

BT
BTITLE SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS
SIDFTC STARSZ M04,XR7 LINK 6 (STARS) 83001000

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS
STAR3Z - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

```

SUBROUTINE STARS                                S3001200
C
C *** DESCRIPTION ***                            S3001500
C THIS SUBROUTINE IS USED TO CALCULATE AND PRINT CERTAIN OPTIC S3001510
C DIAGNOSTICS. AVERAGE X, Y, AND SPOT SIZE ARE COMPUTED FOR ALL S3001520
C SPECIFIED PLANES AND FOR THREE SPECIAL ONES CHOSEN TO MINIMIZE S3001530
C RMS X, RMS Y, AND SPOT SIZE.                  S3001540
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), FUNCID, DT001000
1 DATA(3486)                                     DT001100
DIMENSION WDBJH(7), WCLR(7), WIMHT(7), EIMHT(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIFLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSFLN), (DATA(11), NDBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXFP), (DATA(20), DEXFP), (DATA(21), DT101600
7 WEXFP), (DATA(22), DLFLN), (DATA(23), OMGA2), (DATA(24), DT101700
8 OMGA1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), HD), (DATA(33), DT102000
B DELH), (DATA(34), SYSMX), (DATA(35), WDIR), (DATA(36), DT102100
C WDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WDBJH(1)), (DATA(60), WCLR(1)), DT102300
E (DATA(67), WIMHT(1)), (DATA(74), EIMHT(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCOND) DT102900
EQUIVALENCE (DATA(3484), ATRGGR), (DATA(3485), GAUSS) DT103000
INTEGER CNTRL, AFLAG, DATE, ATRGGR DT201000
REAL LATTC DT202000
C
COMMON / TMPATT / NMISS, NREFLCT, NVIGN, NRAYSB, XBAR, YBAR, XSTAR, L3001000
1 YSTAR, RSTAR, SJSTAR, XCURL(201), YCURL(201), MXBAR, KXBAR, L3001200
2 MYBAR, KYBAR, AX, BX, CX, AY, BY, CY, ACAP, BCAP, CCAP L3001400
REAL MXBAR, KXBAR, MYBAR, KYBAR L3002000
C
COMMON / AZOBJ / THTR, SNTHTR, CSTHTR A2004000
C
COMMON / FRNT / LINE, PAGE FT002500
INTEGER PAGE FT002700
C
DIMENSION XVCT(3), OVCT(3), HEITS(7), LAMDA(100), FMT(7), S3002000
1 OLIST(8), TEMPS(400) S3002200
C
DATA FMT(1) / 6H(1H ,? /, FMT(2) / 6H3,2X, /, FMT(7) / 6H 12 ) /, S3003000
1 EFRMT / 6H16.7, /, AFRMT / 6H4X2A6, /, FINTY1 / 6H INFIN /, S3003200
2 FINTY2 / 6HITY / S3003400
C

```

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS
STAR32 - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

REAL LAMDAX, LAMDY, LAMDZ, LAMDA, LAMDA, LAMDA	\$3004000	
INTEGER SMISS	\$3005000	
EQUIVALENCE (MXCURL(1),XCURL(1)),(KXCURL(1),TEMP5(1)),	\$3005010	
1 (MYCURL(1),YCURL(1)),(KXCURL(1),TEMP5(201))	\$3005020	
REAL MXCURL(200),KXCURL(200),MYCURL(200),KYCURL(200),XCORD(200),	\$3005030	
1 YCORD(200)	\$3005040	
C		
C START OF STAR3 SUBROUTINE	\$3007000	
C		
CALL PRCTL	\$3007500	2
WRITE (6, 32010)	\$3008000	3
32010 FORMAT(1H3, 5HCOLOR, 3X, 12HFOCAL LENGTH, 5X, 11HFOCAL POINT, 5X,	\$3008200	
1 10HBACK FOCUS, 7X, 9HF/ NUMBER)	\$3008400	
LINE = LINE + 2	\$3008600	
BNTHR = 0.0	\$3008800	
DO 7000 I = 1, NCLRS	\$3101000	
THTR = 3.14159265	\$3101200	
CSTHR = - 1.0	\$3101400	
C		
C COMPUTE FOCAL LENGTH AND FOCAL POINT	\$3102000	
CALL PARAX(1, FOCALL, FOCALP)	\$3102200	6
THTR = 0.0	\$3102400	
CSTHR = 1.0	\$3102600	
C		
C COMPUTE BACK FOCUS AND F NUMBER	\$3103000	
CALL MERID(1, BFOCUS, FSNUMB)	\$3103200	10
IF(1 .EQ. 1) DF1 = BFOCUS	\$3103400	
C		
C GENERATE OUTPUT LIST AND FORMAT FOR FOCAL LENGHT, FOCAL	\$3104000	
C POINT, BACK FOCUS, AND F/ NUMBER	\$3104200	
C IF ANY VALUE IS 0.0 THIS INDICATES INFINITY SHOULD BE	\$3104400	
C PRINTED OUT	\$3104600	
DO 2000 J = 3, 6	\$3105000	
2000 FMT(J) = EFRMT	\$3105200	
ITEMP = 1	\$3105400	
IF(FOCALL .EQ. 0.0) GO TO 2500	\$3105600	
OLIST(1) = FOCALL	\$3105800	
ITEMP = 2	\$3106000	
GO TO 3000	\$3106200	
2500 OLIST(1) = FINTY1	\$3106600	
OLIST(2) = FINTY2	\$3106800	
FMT(3) = AFRMT	\$3107000	
ITEMP = 3	\$3107200	
3000 IF(FOCALP .EQ. 0.0) GO TO 3500	\$3107400	
OLIST(ITEMP) = FOCALP	\$3107600	
ITEMP = ITEM + 1	\$3107800	
GO TO 4000	\$3108000	
3500 OLIST(ITEMP) = FINTY1	\$3108600	
OLIST(ITEMP+1) = FINTY2	\$3108800	
FMT(4) = AFRMT	\$3109000	
ITEMP = ITEM + 2	\$3109200	
4000 IF(BFOCUS .EQ. 0.0) GO TO 4500	\$3201000	
OLIST(ITEMP) = BFOCUS	\$3201200	
ITEMP = ITEM + 1	\$3201400	
GO TO 5000	\$3201600	
4500 OLIST(ITEMP) = FINTY1	\$3202000	

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS
STAR32 - EFN SOURCE STATEMENT - IFN(8) -

09/24/68

PAGE

OLIST(ITEMP+1) = FINTY2	83202200	
FMT(5) = AFRMY	83202400	
ITEMP = ITEM + 2	83202600	
8000 IF (FSNUMB .EQ. 0.0) GO TO 8500	83203000	
OLIST(ITEMP) = FSNUMB	83203200	
GO TO 6000	83203400	
8500 OLIST(ITEMP) = FINTY1	83203600	
OLIST(ITEMP+1) = FINTY2	83204000	
ITEMP = ITEM + 1	83204200	
FMT(6) = AFRMT	83204400	
C		
C FMT IS (1H, 13.2X, AND E16.7 OF 4X2A6, DEPENDING ON	83205000	
C WHETHER THE VALUES ARE ZERO OR NOT	83205200	
C		
C PRINT OUT VALUES OR COMMENT INFINITY	83205400	
6000 WRITE (6, FMT) 1, (OLIST(J), J=1, ITEM)	83205600	50
LINE = LINE + 1	83206000	
7000 CONTINUE	83207000	
C		
C COMPUTE OBJECT HEIGHTS	83208000	
DO 8000 J = 1, NDBJH	83208200	
8000 HEITS(J) = HD + FLOAT(J-1) * DELH	83208400	
THTR = 0.0	83209000	
8NTHTR = 0.0	83209200	
8BTHTR = 1.0	83209400	
C		
C PRINT HEADING FOR EXIT PUPIL	83301000	
WRITE (6, 32020)	83301200	65
32020 FORMAT(1H, 4X, 6HHEIGHT, 6X, 5HCOLOR, 4X, 10HEXIT PUPIL)	83301400	
LINE = LINE + 2	83301600	
DO 13600 J = 1, NDBJH	83302000	
IF (HEITS(J) .EQ. 0.0) GO TO 13600	83302400	
DO 13500 I = 1, NCLRS	83303000	
.ALL RAYTR(0., 0., HEITS(J), 1, XVCT, QVCT, SMISS, \$10200,	83303400	
1 \$10200, \$10000, 0, DUMMY, DUMMY)	83303600	76
10000 IF (QVCT(2) .NE. 0.0) GO TO 10500	83304000	
10200 EXITP = 0.0	83304200	
GO TO 11000	83304400	
10500 EXITP = - QVCT(3) * XVCT(2) / QVCT(2) + XVCT(3)	83304600	
11000 IF (LINE .LE. 54) GO TO 11500	83305000	
C		
C BEGIN NEW PAGE, PRINT TITLE	83305200	
CALL PRCTL	83305400	88
WRITE (6, 32020)	83305600	89
LINE = LINE + 2	83305800	
11500 IF (EXITP .NE. 0.0) GO TO 12000	83306000	
C PRINT INFINITY FOR VALUE OF EXIT PUPIL	83306400	
WRITE (6, 32030) HEITS(J), 1	83306600	92
32030 FORMAT(1H, E14.7, 4X, 11, 6X, 9H INFINITY)	83306800	
GO TO 13000	83307000	
C		
C PRINT EXIT PUPIL VALUE	83307600	
12000 WRITE (6, 32040) HEITS(J), 1, EXITP	83307800	95
32040 FORMAT(1H, E14.7, 4X, 11, E18.7)	83308000	
13000 LINE = LINE + 1	83308200	
13500 CONTINUE	83308400	

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS		09/24/68	PAGE
STAR32 - EFN SOURCE STATEMENT - IFN(S) -			
13600	CONTINUE	83308500	
	THTR = ROTAN * 3.14159265 / 180.	83309000	
	SINTHTR = SIN(THTR)	83309200	103
	CSTHTR = COS(THTR)	83309400	104
	MRAYS = NRAYS	83309600	
	IF(LMODE .EQ. 0) MRAYS = 2 * NRAYS	83309800	
C			
	MTEMP = 0	83400500	
	DO 20000 J = 1, NOBJM	83401000	
	DO 20000 I = 1, NCLRS	83402000	
C			
	MTEMP = MTEMP + 1	83402200	
C	COMPUTE STATISTICAL INFORMATION	83402400	
	CALL SMOPOP(I, J, HEITS(J), DF1, 1, TEMPS)	83402600	115
	CALL PRICTL	83402800	118
	WRITE (6, 32050) MTEMP, J, I	83403000	119
32050	FORMAT(1HD, //13,14H, OBJECT POINT, 12, 1X, 6H COLOR, 12, 21H ...	83403200	
	1.....)	83403400	
C			
C	PRINT OUT RAY TRACE INFORMATION	83404000	
	WRITE (6, 32060) HEITS(J), THTR, MRAYS, NMISS, NRFLCT, NVIGN,	83404200	
	1 NRAYSB	83404400	120
32060	FORMAT(1HD, 7X, 6HHEIGHT, 10X, 7HAZIMUTH / 2X, 2E16.7 // 6X,	83405000	
	1 52HRAYS TRACED MISSED REFLECTIONS VIGNETS SUCCESSES / 9X, 13,	83405200	
	2 8X, 13, 8X, 13, 8X, 13, 7X, 13)	83405400	
	LINE = LINE + 10	83405600	
C			
C	PRINT OUT RAY STATISTICS	83406000	
	WRITE (6, 32070) MXBAR, KXBAR, MYBAR, KYBAR, AX, BX, CX, AY,	83406200	
	1 BY, CY, ACAP, BCAP, CCAP	83406400	122
32070	FORMAT(1HD, 8X, 5HMXBAR, 11X, 5HKXBAR, 11X, 5HMYBAR, 11X,	83406800	
	1 5HKYBAR / 2X, 4E16.7 // 10X, 2HAX, 14X, 2HBY, 14X, 2HCX / 2X,	83407000	
	2 3E16.7 // 10X, 2HAY, 14X, 2HBY, 14X, 2HCY / 2X, 3E16.7 // 11X,	83407200	
	3 1HA, 15X, 1HD, 15X, 1HC / 2X, 3E16.7)	83407400	
	LINE = LINE + 12	83407600	
	WRITE (6, 32080)	83408000	123
32080	FORMAT(1HD, 20X, 13HLOC. OF PLANE, 5X, 9HAVERAGE X, 7X, 9HAVERAGE	83408400	
	1 Y, 5X, 12HIMAGE HEIGHT, 8X, 5HRMS X, 11X, 5HRMS Y, 9X, 9HSFOT SIZE)	83408600	
	LINE = LINE + 2	83408800	
	LAMDAX = - BX / AX	83409000	
	LAMDAY = - BY / AY	83409200	
	LAMDAR = - BCAP / ACAP	83409400	
	IF(1 .NE. 1) GO TO 14500	83501000	
	LAMDAD = S. STAR + FLOAT(NSPLN) * DLPLN	83501200	
	DO 14000 K = 1, NIPLN	83501600	
14000	LAMDA(K) = LAMDAD + FLOAT(K-1) * DLPLN	83501800	
14500	LAMDAT = LAMDAX	83502000	
	KTEMP = 1	83502400	
	FNR=NRAYSB	83502402	
15000	XBAR = MXBAR * LAMDAT + KXBAR	83502600	
	YBAR = MYBAR * LAMDAT + KYBAR	83502800	
	HIMAGE = XBAR * SINTHTR + YBAR * CSTHTR	83503000	
	XSTAR=0.	83503010	
	YSTAR=0.	83503012	
	DO 15010 K=1, NRAYSB	83503014	
	XCORD(K) = MXCURL(K) * LAMDAT + KXCURL(K)	83503016	

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS
STARSZ - EFN SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

15010	YCORD(K)=MYCURL(K)*LAMDAT+KYCURL(K)	83503018	
	DO 15012 K=1,NRAYSB	83503020	
	XSTAR=XSTAR+XCORD(K)*XCORD(K)	83503022	
15012	YSTAR=YSTAR+YCORD(K)*YCORD(K)	83503024	
	XSTAR=SQRT(XSTAR/FNR)	83503026	154
	YSTAR=SQRT(YSTAR/FNR)	83503028	155
	RSTAR=SQRT(XSTAR*XSTAR+YSTAR*YSTAR)	83503030	156
	GO TO (15500, 16000, 16500), KTEMP	83504000	
C			
C	PRINT MINIMUM RMS X	83505000	
15500	WRITE (6, 32090) LAMDAX, XBAR, YBAR, HIMAGE, XSTAR, YSTAR, RSTAR	83505200	158
32090	FORMAT(1H, 3X, 10HMIN, RMS X, 4X, 7E16.7)	83505400	
	KTEMP = 2	83506000	
	LAMDAT = LAMDAY	83506200	
	GO TO 15000	83506400	
C			
C	PRINT MINIMUM RMS Y	83507000	
16000	WRITE (6, 32100) LAMDAY, XBAR, YBAR, HIMAGE, XSTAR, YSTAR, RSTAR	83507200	161
32100	FORMAT(1H, 3X, 10HMIN, RMS Y, 4X, 7E16.7)	83507400	
	KTEMP = 3	83508000	
	LAMDAT = LAMDAR	83508200	
	GO TO 15000	83508400	
C			
C	PRINT MINIMUM SPOT SIZE	83509000	
16500	WRITE (6, 32110) LAMDAR, XBAR, YBAR, HIMAGE, XSTAR, YSTAR, RSTAR	83509200	164
32110	FORMAT(1H, 3X, 17HIN, SPOT SIZE, 7E16.7)	83509400	
	LINE = LINE + 3	83509600	
	DO 16000 K = 1, NIPLN	83601000	
	IF(LINE .LE. 94) GO TO 17000	83601200	
C	BEGIN NEW PAGE - PRINT HEADINGS	83601600	
	CALL PRICTL	83601800	171
	WRITE (6, 32080)	83602000	172
	LINE = LINE + 2	83602200	
C			
17000	XBAR = MXBAR * LAMDA(K) + KXBAR	83603000	
	YBAR = MYBAR * LAMDA(K) + KYBAR	83603200	
	HIMAGE = XBAR * SINTHR + YBAR * CSTHR	83603400	
	XSTAR=D.	83603410	
	YSTAR=D.	83603412	
	DO 17010 ITER=1,NRAYSB	83603414	
	XCORD(ITER)=MXCURL(ITER)*LAMDA(K)+KXCURL(ITER)	83613416	
17010	YCORD(ITER)=MYCURL(ITER)*LAMDA(K)+KYCURL(ITER)	83603418	
	DO 17012 ITER=1,NRAYSB	83603420	
	XSTAR=XSTAR+XCORD(ITER)*XCORD(ITER)	83603422	
17012	YSTAR=YSTAR+YCORD(ITER)*YCORD(ITER)	83603424	
	XSTAR=SQRT(XSTAR/FNR)	83603426	196
	YSTAR=SQRT(YSTAR/FNR)	83603428	197
	RSTAR=SQRT(XSTAR*XSTAR+YSTAR*YSTAR)	83603430	198
	WRITE (6,32120) K, LAMDA(K), XBAR, YBAR, HIMAGE, XSTAR, YSTAR, RSTAR	83604600	199
32120	FORMAT(4X, 5HPLANE, 14, 5X, 7E16.7)	83605400	
	LINE = LINE + 1	83605800	
18000	CONTINUE	83607000	
20000	CONTINUE		
C			
C	END OF STARS SUBROUTINE	83901000	
	RETURN	83902000	

TR-67-700-10-2
Page III-6-11
Program Listing
Link 6
Page Revised 10/1/68

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS
STAR32 - EFM SOURCE STATEMENT - IFN(S) -

09/24/68

PAGE

END

83903000

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS 09/24/68 PAGE
 STARS2 STORAGE MAP

SUBROUTINE STARS
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06671	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUCP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUME	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXFP	00055	R	DEXFP	00056	R	WEXFP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	EPFEA	00067	R	DUMIN	00070	R
COIST	00071	R	HO	00072	R	DELH	00073	R
SYBHX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	WDSGV	00104	I
WDBJH	00117	R	WCLRH	00126	R	WINHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
BUDST	05224	R	ISUDST	05224	I	BOUND9	06210	R
IDNDS	06210	I	NCOND	06665	I	ATRCR	06666	I
GAUSS	06667	R						

COMMON BLOCK			TMPATT	ORIGIN	06672	LENGTH	00651	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NRFLCT	00000	I	NRFLCT	00001	I	NVIGN	00002	I
XBAR	00003	I	XBAR	00004	R	YBAR	00005	R
YSTAR	00005	R	YSTAR	00007	R	RSTAR	00010	R
XSTAR	00011	R	XCURL	00012	R	YCURL	00323	R
KYBAR	00634	R	KYBAR	00635	R	MYBAR	00636	R
AX	00640	R	AX	00640	R	BX	00641	R
AY	00642	R	AY	00643	R	BY	00644	R
ACAP	00645	R	ACAP	00646	R	BCAP	00647	R
MXCURL	00650	R	MXCURL	00012	R	MYCURL	00323	R

COMMON BLOCK			AZOBJ	ORIGIN	07543	LENGTH	00003	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
INTHR	00000	R	SINTHR	00001	R	CSINTR	00002	R

COMMON BLOCK			PRNT	ORIGIN	07546	LENGTH	00002	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LINE	00000	I	PAGE	00001	I			

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
KVCT	10370	R	OVCT	10373	R	HEITS	10376	R
LAMDA	10405	R	FMT	10551	R	OLIST	10560	R
TEMP9	07550	R	KXCURL	07550	R	KYCURL	10660	R
KCORD	10570	R	YCORD	11100	R			

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS
 STAR3Z STORAGE MAP

09/24/68

PAGE 8

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LAMDAX	11410	R	LAMDAY	11411	R	LAMDAR	11412	R
LAMDAD	11413	R	LAMDAT	11414	R	SMISS	11415	I
I	11416	I	FOCALL	11417	R	FOCALP	11420	R
BFOCUS	11421	R	FSNUMB	11422	R	BF1	11423	R
J	11424	I	EFRMT	11425	R	ITEMP	11426	I
FINTY1	11427	R	FINTY2	11430	R	AFRMT	11431	P
DUMMY	11432	R	EXITP	11433	R	HRAYS	11434	I
MTEMP	11435	I	K	11436	I	KTEMP	11437	I
FNR	11440	F	MIMAGE	11441	R			

ENTRY POINTS

STAR3 SECTION 11

SUBROUTINES CALLED

PRCTL SECTION 12
 MERID SECTION 15
 COS SECTION 18
 .FXEM. SECTION 21
 .FCNV. SECTION 24
 E.3 SECTION 27
 CC.2 SECTION 30
 SYSLOC SECTION --- 33

.FWRD. SECTION 13
 RAYTR SECTION 16
 SMFOP SECTION 19
 .UNDG. SECTION 22
 E.1 SECTION 25
 E.4 SECTION 28
 CC.3 SECTION 31

PARAX SECTION
 SIN SECTION
 SQRT SECTION
 .FFIL. SECTION
 E.2 SECTION
 CC.1 SECTION
 CC.4 SECTION

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32010	FORMAT	11463	7000	55A	12174	2000	16A	12004
2500	24A	12023	3000	25A	12033	3500	30A	12050
4000	33A	12066	4500	36A	12103	5000	41A	12121
5500	46A	12133	6000	50A	12151	6000	61A	12204
32020	FORMAT	11503	13600	100A	12404	13500	98A	12402
10200	82A	12312	10000	79A	12306	10500	84A	12314
11000	85A	12323	11500	90A	12344	12000	95A	12363
32030	FORMAT	11514	13000	97A	12377	32040	FORMAT	11523
20000	203A	13344	32050	FORMAT	11530	32060	FORMAT	11546
32070	FORMAT	11601	32080	FORMAT	11650	14500	133A	12677
14000	130A	12657	15000	135A	12713	15010	141A	12744
15012	150A	12762	15500	158A	13034	16000	161A	13065
16500	164A	13116	32090	FORMAT	11673	32100	FORMAT	11704
32110	FORMAT	11713	18000	201A	13342	17000	173A	13174
17010	182A	13231	17012	192A	13247	32120	FORMAT	11722

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 13403.

TR-67-700-10-2
Page III-7-1
Program Listing
Link 7

Section 7

LINK 7

TR-67-700-10-2
Page III-7-2
Program Listing
Link 7

INF.ND10001-44000-0.77387.AM IC ISJOB LINK1 7094 A 02/10/68 PAGE 8

30
TITLE SUBROUTINE TO COMPUTE LINE AND CHARACTER POSITION OF POINT
JOBTC GETLCZ MS6.M7 LINK 7 (GETLC) 6T001000

SUBROUTINE TO COMPUTE LINE AND CHARACTER POSITION OF POINT
 GETLCZ STORAGE MAP

08/16/68

PAGE 2

SUBROUTINE GETLC
 UNDIMENSIONED PROGRAM VARIABLES

SYMBOL OF TY	LOCATION 00001 00004	TYPE R R	SYMBOL BY	LOCATION 00002	TYPE R	SYMBOL TX	LOCATION 00003	TYPE R
ENTRY POINTS								
GETLC	SECTION	3						
SUBROUTINES CALLED								
E.1	SECTION	4	E.2	SECTION	5	E.3	SECTION	6
E.4	SECTION	7	CC.1	SECTION	8	CC.2	SECTION	9
CC.3	SECTION	10	CC.4	SECTION	11	SYSLOC	SECTION	12
EFN IFN CORRESPONDENCE								
EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00152.

TR-67-700-10-2
Page III-7-5
Program Listing
Link 7

INF.N010001.44000-0.77307.AM IC

ISJOB LICNR: 7004 A 02/16/68

PAGE 3

BTITLE SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
JIBFTC STAR72 H04.HR7 LINK 7 (STAR7)

87001000

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
STAR7Z - EFN SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 1

```

SUBROUTINE STAR7                                DT001500
C
C *** DESCRIPTION ***                          DT002000
C STAR7 LISTS THE ENTRANCE PUPIL PATTERN AND THE IMAGE PLANE DT002100
C COORDINATES FOR EACH COMBINATION OF COLOR AND OBJECT POINT. DT002200
C SPOT DIAGRAM POINT PLOTS ARE PRINTED FOR THE ENTRANCE PUPIL DT002300
C AND EACH COMBINATION UNLESS A RAY FAILURE OCCURS. DT002400
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID, DT001000
1 DATA(3483) DT001100
DIMENSION WOBJH(7), WCLRH(7), WINHT(7), EINH(7), CIMPL(7), DT002000
1 WCLRS(6), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100), DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26), DT002200
3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100) DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRAYS), (DATA(3), DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6), DT101100
2 NSUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9), DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12), DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15), DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18), DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21), DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMG2), (DATA(24), DT101700
8 OMG1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27), DT101800
9 PSCAL), (DATA(28), OMCAF), (DATA(29), SPFEA), (DATA(30), DT101900
A DUMIN), (DATA(31), ODIST), (DATA(32), HO), (DATA(33), DT102000
B DELH), (DATA(34), SYSMX), (DATA(35), WDIR), (DATA(36), DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42), DT102200
D NDSGV(1)), (DATA(53), WOBJH(1)), (DATA(60), WCLRH(1)), DT102300
E (DATA(67), WINHT(1)), (DATA(74), EINH(1)), (DATA(81), DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)), DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(182), SURFC(1,1), DT102600
H ISURFC(1,1)), (DATA(2182), DESGN(1,1), IDESGN(1,1)), DT102700
I (DATA(2682), SUBST(1), ISUBST(1)), (DATA(3182), BOUNDS(1,1), DT102800
J IBNDS(1,1)), (DATA(3483), NCNTRL) DT102900
INTEGER CNTRL, AFLAG, DATE DT201000
REAL LATT DT202000
C
COMMON / TMPATT / NMISS, NRFLCT, NVIGN, NRAYSB, XBAR, YBAR, XBAR, L3001000
1 YSTAR, RSTAR, SJSTAR, XCURL(201), YCURL(201), MXBAR, KXBAR, L3001200
2 NYBAR, KYBAR, AX, BX, CX, AY, BY, CY, ACAP, BCAP, CCAP L3001400
REAL MXBAR, KXBAR, NYBAR, KYBAR L3002000
C
COMMON / PRNT / LINE, PAGE PT002500
INTEGER PAGE PT002700
C
COMMON / AZOBJ / THTR, STHTR, CSTNR AZ004000
C
DIMENSION NRBAR(42), NX(201), NY(201), PRTL(103), TEMPS(200), DT002500
1 PLTCHR(38), CHRSP(200) DT002600
C
INTEGER COLOR, MNUMB, RYCNT DT003000
REAL LENGTH DT003500
C
DATA PLTCHR(1) / 1H1 /, PLTCHR(2) / 1H2 /, PLTCHR(3) / 1H3 /, DT004000
1 PLTCHR(4) / 1H4 /, PLTCHR(5) / 1H5 /, PLTCHR(6) / 1H6 /, DT004100

```

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
STAFFZ - EFN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 2

	2 PLTCHR(7) / 1H7 /, PLTCHR(8) / 1H8 /, PLTCHR(9) / 1H9 /,	87004200	
	3 PLTCHR(10) / 1HA /, PLTCHR(11) / 1HB /, PLTCHR(12) / 1HC /,	87004300	
	4 PLTCHR(13) / 1HD /, PLTCHR(14) / 1HE /, PLTCHR(15) / 1HF /,	87004400	
	5 PLTCHR(16) / 1HG /, PLTCHR(17) / 1HH /, PLTCHR(18) / 1HI /,	87004500	
	6 PLTCHR(19) / 1HJ /, PLTCHR(20) / 1HK /, PLTCHR(21) / 1HL /,	87004600	
	7 PLTCHR(22) / 1HM /, PLTCHR(23) / 1HN /, PLTCHR(24) / 1HO /,	87004700	
	8 PLTCHR(25) / 1HP /, PLTCHR(26) / 1HQ /, PLTCHR(27) / 1HR /,	87004800	
	9 PLTCHR(28) / 1HS /, PLTCHR(29) / 1HT /, PLTCHR(30) / 1HU /,	87004900	
	A PLTCHR(31) / 1HV /, PLTCHR(32) / 1HW /, PLTCHR(33) / 1HX /,	87005000	
	B PLTCHR(34) / 1HY /, PLTCHR(35) / 1HZ /, PLTCHR(36) / 1H /,	87005100	
	C PLTCHR(37) / 1HD /, PLTCHR(38) / 1HO /	87005200	
C			
C	START OF START SUBROUTINE	87021000	
C			
	MR = NRAYS	87021500	
	IF (LMODE .EQ. 0) MR = MR + MR	87021600	
	NRAYS = 0	87022000	
	RYCNT = 0	87022200	
	DO 2000 I = 1, NRAYS	87022400	
	RYCNT = RYCNT + 1	87022600	
	CALL LATT(XCURL(RYCNT), YCURL(RYCNT), NRAYS)	87022800	11
	IF (LMODE .NE. 0) GO TO 2000	87023000	
	XCURL(RYCNT+1) = - XCURL(RYCNT)	87023200	
	YCURL(RYCNT+1) = YCURL(RYCNT)	87023400	
	RYCNT = RYCNT + 1	87023600	
	2000 CONTINUE	87023800	
	DELL = 2. / 61.	87024000	
	DELC = 2. / 102.	87024200	
	LINE = 80	87024400	
C			
	DO 5000 I = 1, MR	87025000	
	IF (LINE .LE. 60) GO TO 4000	87025200	
C	BEGIN NEW PAGE	87025400	
	CALL PRCTL	87025600	32
	WRITE (6, 32010)	87025800	33
	32010 FORMAT(1H, 45HRAY COORDINATES ON UNIT RADIUS ENTRANCE PUPIL /	87026000	
	14H RAY, 6X, 1H, 15X, 1HY, 6X, 26HLINE CHAR PLOT CHARACTER)	87026200	
	LINE = LINE + 4	87026400	
C			
C	COMPUTE LINE AND CHARACTER	87026600	
	4000 CALL GETLC(XCURL(I), YCURL(I), -1., DELL, DELC, NX(I), NY(I))	87026800	39
	CHAR = PLTCHR(33)	87027000	
	IF (MR .LE. 53) CHAR = PLTCHR(I)	87027200	
	WRITE (6, 32020) I, XCURL(I), YCURL(I), NY(I), NX(I), CHAR	87027400	43
	32020 FORMAT(1H, 15, 2E16.7, 15, 16, 10X, A1)	87027600	
	LINE = LINE + 1	87027800	
	5000 CONTINUE	87028000	
	KBAR = MR + 1	87028200	
C			
C	CONVERT ORIGIN	87028400	
	CALL GETLC(0., 0., -1., DELL, DELC, NX(KBAR), NY(KBAR))	87028600	34
	PAGE = PAGE + 1	87028800	
	LINE = 0	87029000	
C			
C	PRINT FIRST LINE	87041300	
	WRITE (6, 32030) (TITLE(L), L=1,12), PAGE	87041200	37

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
 STARYZ - EFM SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 3

32030	FORMAT(1H1, 12A6, 3X, 29HRAY PATTERN ON ENTRANCE PUPIL, 10X, 1 4MPAGE, 14)	87041400	
	DO 8500 I = 1, 62	87041600	
	PRTL(I) = PLTCHR(38)	87041800	
	PRTL(103) = PLTCHR(38)	87042000	
	DO 6000 K = 2, 102	87042200	
6000	PRTL(K) = PLTCHR(38)	87042400	
	IF(LINE .NE. 0 .AND. LINE .NE. 61) GO TO 7000	87042600	
	DO 6500 R = 3, 101, 2	87042800	
6500	PRTL(R) = PLTCHR(38)	87043000	
7000	DO 7500 K = 1, NR	87043200	
	IF(LINE .NE. NY(K)) GO TO 7500	87043400	
	KL = NX(K) + 1	87043600	
	PRTL(KL) = PLTCHR(K)	87043800	
	IF(NR .GT. 35) PRTL(KL) = PLTCHR(33)	87044000	
7500	CONTINUE	87044200	
	IF(LINE .NE. NY(KBAR)) GO TO 8000	87044400	
	KL = NX(KBAR) + 1	87044600	
	PRTL(KL) = PLTCHR(37)	87044800	
8000	IF(LINE .EQ. 0) GO TO 8200	87045000	
	IF(LINE .EQ. 61) GO TO 8400	87045200	
	WRITE (6, 32040) PRTL	87045400	
32040	FORMAT(20X, 103A1)	87045600	108
	GO TO 8500	87045800	
		87046000	
		87046200	
		87046400	
C		87046600	
C	PRINT UPPER LINE	87046800	
8200	SHAX = 1.	87047000	112
	WRITE (6, 32050) SHAX, PRTL	87047200	
32050	FORMAT(6X, E11.4, 3X, 103A1)	87047400	
	GO TO 8500		
C		87047600	
C	PRINT LOWER LINE	87047800	
8400	SHIN = - 1.	87048000	116
	WRITE (6, 32050) SHIN, PRTL	87048200	
8500	LINE = LINE + 1		
C		87051000	
C	BEGIN REGULAR PLOTS	87051200	
	NSPOTS = 0	87051400	
	LINE = 60	87051600	
	REWIND 11		
C		87051800	123
	DO 9000 I = 1, 42	87052000	
9000	NRBAR(I) = 0	87052200	
	IF(IMODE .NE. 0) GO TO 10000	87052400	
	THTR = 0.	87052600	
	SINTHTR = 0.	87052800	
	CSITHTR = 1.	87053000	134
	CALL MERID(1, BF1, DUMMY)	87053200	
10000	THTR = ROTAN * 3.14159265 / 180.	87053400	138
	SINTHTR = SIN(THTR)	87053600	
	CSITHTR = COS(THTR)		
C		87053800	137
C	BEGIN DO LOOP ON OBJECT HEIGHT	87054000	
	DO 12500 J = 1, NOBJH	87054200	
	HEIGHT = HO + FLOAT(J-1) * DELH		
C			

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
STARTZ - EPN SOURCE STATEMENT - IFN(B) -

02/16/68

PAGE 4

C	BEGIN LOOP ON COLORS	87084400	
	DO 12500 I = 1, NCLR8	87084600	
C	TRACE RAY PATTERN	87084800	
	CALL SHPOP(I, J, HEIGHT, BP1, 3, TEMPS)	87085000	143
	IF(NRAYSB .EQ. 0) GO TO 12500	87085200	
	NSPOTS = NSPOTS + 1	87085400	
	NRBAR(NSPOTS) = NRAYSB	87085600	
	XTHIN = XCURL(I)	87085800	
	XTHAX = XCURL(I)	87086000	
	YTHIN = YCURL(I)	87086200	
	YTHAX = YCURL(I)	87086400	
	IF(NRAYSB .EQ. 1) GO TO 11200	87086600	
C			
	DO 11000 K = 2, NRAYSB	87086800	
	XTHIN = AMIN1(XTHIN, XCURL(K))	87087000	
	XTHAX = AMAX1(XTHAX, XCURL(K))	87087200	
	YTHIN = AMIN1(YTHIN, YCURL(K))	87087400	
11000	YTHAX = AMAX1(YTHAX, YCURL(K))	87087600	
11200	IF(NSPOTS.NE. 1) GO TO 11500	87087800	
	XMIN = XTHIN	87088000	
	XMAX = XTHAX	87088200	
	YMIN = YTHIN	87088400	
	YMAX = YTHAX	87088600	
	GO TO 12000	87088800	
11500	XMIN = AMIN1(XMIN, XTHIN)	87089000	
	XMAX = AMAX1(XMAX, XTHAX)	87089200	
	YMIN = AMIN1(YMIN, YTHIN)	87089400	
	YMAX = AMAX1(YMAX, YTHAX)	87089600	
C			
C	WRITE BINARY TAPE 11 - 1 RECORD	87071000	
12000	WRITE (11) J, I, XBAR, YBAR, NRAYSB, XTHIN, XTHAX, YTHIN,	87071200	
	1 YTHAX, (XCURL(K), K=1, NRAYSB), (YCURL(K), K=1, NRAYSB),	87071400	
	2 (TEMPS(K), K=1, NR)	87071600	146
12500	CONTINUE	87071800	
	IF(NSPOTS .EQ. 0) GO TO 32000	87072000	
C			
	END FILE 11	87072200	145
	REWIND 11	87072400	148
	SMIN = PSCAL * AMIN1(XMIN, YMIN)	87072600	
	SMAX = PSCAL * AMAX1(XMAX, YMAX)	87072800	
C			
C	COMPUTE SCALE	87073000	
	CALL SCALER(SMIN, SMAX, PLX, PLY, XMIN, XMAX)	87073200	147
	SMIN = PLX	87073400	
	SMAX = PLY	87073600	
	IF(SMIN .GE. 0. .OR. SMAX .LE. 0.) GO TO 13000	87073800	
	SMAX = AMAX1(ABS(SMIN), SMAX)	87074000	
	SMIN = - SMAX	87074200	
13000	LENGTH = SMAX - SMIN	87074400	
	DELL = LENGTH / 61.	87074600	
	DELC = LENGTH / 102.	87074800	
	REWIND 12	87075000	
C			
C	CONVERT ORIGIN	87075200	143
	CALL GETLC(0., 0., SMIN, DELL, DELC, NXORG, NYORG)	87075400	144
	DO 14000 I = 1, NSPOTS	87075600	

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
 STAR7Z - EFM SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 9

KBAR = MRBAR(1)	87075800	
C		
C READ BINARY TAPE 11	87076000	
READ (11) MNUMB, COLOR, XBAR, YBAR, DUMMY, DUMMY, DUMMY, DUMMY,	87076200	
1 DUMMY, (XCURL(K), K=1, KBAR), (YCURL(K), K=1, KBAR),	87076400	
2 (TEMPS(K), K=1, MR)	87076600	200
IF (LINE .GT. 80) CALL PRCTCL	87076800	
C		216
WRITE (6, 32060) MNUMB, COLOR, XBAR, YBAR, KBAR	87077000	217
32060 FORMAT(1H0, 12HOBJECT POINT, I2, 2X, 5HCOLOR, I2 / 13X, 9HAVERAGE	87077200	
1E X, 7X, 9HAVERAGE Y, 4X, 14HNUMBER OF RAYS / 6X, 2E16.7, I11 //	87077400	
2 3X, 3HRAY, 9X, 1HX, 19X, 1HY, 10X, 10HX (SCALED), 6X, 10HY (SCALE	87077600	
3D), 4X, 4HLINE, 2X, 4HCHAR, 2X, 14HPLOT CHARACTER)	87077800	
LINE = LINE + 7	87078000	
RYCNT = 0	87078200	
C		
DO 15000 K = 1, MR	87078400	
IF (LINE .LE. 99) GO TO 14000	87078600	
C BEGIN NEW PAGE	87078800	
CALL PRCTCL	87079000	226
WRITE (6, 32070) MNUMB, COLOR	87079200	228
32070 FORMAT(1H0, 12HOBJECT POINT, I2, 2X, 5HCOLOR, I2 // 5X, 3HRAY,	87081000	
1 9X, 1HX, 19X, 1HY, 10X, 10HX (SCALED), 6X, 10HY (SCALE), 4X,	87081200	
2 4HLINE, 2X, 4HCHAR, 2X, 14HPLOT CHARACTER)	87081400	
LINE = LINE + 4	87081600	
14000 INDRAY = TEMPS(K) + 1.	87081800	
GO TO (14200, 14400, 14600, 14800), INDRAY	87082000	
C		
C SUCCESSFUL RAY	87082200	
14200 RYCNT = RYCNT + 1	87082400	
X8 = PSCAL * XCURL(RYCNT)	87082600	
Y2 = PSCAL * YCURL(RYCNT)	87082800	
C		
C GET LINE AND CHARACTER COORDINATES	87083000	
CALL GETLC(X8, Y8, 8MIN, DELL, DELC, NX(RYCNT), NY(RYCNT))	87083200	
C		
C GET PLOT CHARACTER	87083400	234
CHRSP(RYCNT) = PLTCHR(K)	87083600	
IF (MR .GT. 39) CHRSP(RYCNT) = ' ' (39)	87083800	
WRITE (6, 32080) K, XCURL(RYCNT), YCURL(RYCNT), X8, Y8,	87084000	
1 NY(RYCNT), NX(RYCNT), CHRSP(RYCNT)	87084200	242
32080 FORMAT(1H , 4X, I3, 4E16.7, I5, I6, I6X, A1)	87084400	
GO TO 15000	87084600	
C		
C MISSED RAY	87084800	
14400 WRITE (6, 32090) K	87085000	249
32090 FORMAT(1H , 4X, I3, 6X, 6HMISSED)	87085200	
GO TO 15000	87085400	
C		
C REFLECTED RAY	87085600	
14600 WRITE (6, 32100) K	87085800	251
32100 FORMAT(1H , 4X, I3, 6X, 6HREFLECTED)	87086000	
GO TO 15000	87086200	
C		
C VIGNETTED RAY	87086400	
14800 WRITE (6, 32110) K	87086600	253

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
 STARTZ - EPN SOURCE STATEMENT - I7N(8) -

02/16/68

PAGE 6

32110	FORMAT(1H , 4X, 13, 8X, 8HVISMETTED)	87086800	
19000	LINE = LINE + 1	87087000	
	WRITE (12) INUMB, COLOR, (NX(K), K=1,KBAR), (NY(K), K=1,KBAR	87087200	
	1), (CHRSPY(K), K=1,KBAR)	87087400	286
18000	CONTINUE	87087600	
	END FILE 12	87087800	271
	REWIND 12	87088000	
C			272
	DO 20000 I = 1, NSPOTS	87088200	
	KBAR = NRBAR(I)	87088400	
	READ (12) INUMB, COLOR, (NX(K), K=1,KBAR), (NY(K), K=1,KBAR	87088600	
	1), (CHRSPY(K), K=1,KBAR)	87088800	277
	PAGE = PAGE + 1	87089000	
	LINE = 0	87089200	
	WRITE (6, 32120) (TITLE(L), L=1,12), INUMB, COLOR, PAGE	87091000	293
32120	FORMAT(1H1, 12A6, 3X, 13HHEIGHT NUMBER, 12, 2X, 12HCOLOR NUMBER,	87091200	
	1 12, 8X, 4HPAGE, 14)	87091400	
C	CONSTRUCT JTH. LINE OF PLOT	87091600	
C			
	DO 19000 J = 1, 62	87091800	
	PRTL(1) = PLTCHR(30)	87092000	
	PRTL(103) = PLTCHR(38)	87092200	
	DO 16500 K = 2, 102	87092400	
16500	PRTL(K) = PLTCHR(36)	87092600	
	IF (LINE .NE. 0 .AND. LINE .NE. 61) GO TO 17200	87092800	
	DO 17000 K = 3, 101, 2	87093000	
17000	PRTL(K) = PLTCHR(38)	87093200	
17200	DO 17500 K = 1, KBAR	87093400	
	IF (LINE .NE. NY(K)) GO TO 17300	87093600	
	KL = NX(K) + 1	87093800	
	PRTL(KL) = CHRSPY(K)	87094000	
17500	CONTINUE	87094200	
	IF (LINE .NE. NYORB) GO TO 18000	87094400	
	KL = NKORB + 1	87094600	
	PRTL(KL) = PLTCHR(37)	87094800	
18000	IF (LINE .EQ. 0) GO TO 18200	87095000	
	IF (LINE .EQ. 61) GO TO 18400	87095200	
	WRITE (6, 32040) PRTL	87095400	337
	GO TO 18500	87095600	
18200	WRITE (6, 32050) SMAX, PRTL	87095800	340
	GO TO 18500	87096000	
18400	WRITE (6, 32050) SMIN, PRTL	87096200	343
18500	LINE = LINE + 1	87096400	
19000	CONTINUE	87096600	
20000	CONTINUE	87096800	
C			
32000	RETURN	87098000	
	END	87098400	

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
STARTZ STORAGE MAP

09/18/68

PAGE 7

SUBROUTINE START
COMMON VARIABLES

COMMON BLOCK			DATA	ORIGIN	00001	LENGTH	06666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NCNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
UNODE	00033	I	NRAYS	00034	I	NSLCS	00038	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIFLN	00042	I	THODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
WEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMG1P	00066	R	SPFEA	00067	R	QUMIN	00070	R
ODIST	00071	R	HO	00072	N	DELH	00073	R
SYSNX	00074	R	WXDIR	00075	R	WYDIR	00076	R
NDTAN	00077	R	NDSGN	00100	I	NDSCV	00104	I
WOBJH	00117	R	WCLRH	00126	R	WIMHT	00135	R
EIMHT	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I			

COMMON BLOCK			TRIPATT	ORIGIN	06667	LENGTH	00651	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NRFLCT	00000	I	NRFLCT	00001	I	NVIGN	00002	I
XBAR	00003	I	XBAR	00004	R	YBAR	00005	R
YSTAR	00006	R	YSTAR	00007	R	KSTAR	00010	R
SJSTAR	00011	R	XCURL	00012	R	YCURL	00323	R
KXBAR	00634	R	KXBAR	00635	R	MYBAR	00636	R
KYBAR	00637	R	AX	00640	R	BX	00641	R
OX	00642	R	AY	00643	R	BY	00644	R
CY	00645	R	ACAP	00646	R	BCAP	00647	R
CCAP	00650	R						

COMMON BLOCK			PRNT	ORIGIN	07540	LENGTH	00002	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LINE	00000	I	PAGE	00001	I			

COMMON BLOCK			AZOBJ	ORIGIN	07542	LENGTH	00003	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
THTR	00000	R	SNTHTR	00001	R	CSHTR	00002	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NRBAR	07545	I	NX	07617	I	NY	10130	I
PRTL	10441	R	TEMP8	10610	R	PLTCHR	11180	R
CHRSP	11186	R						

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
 STARTZ

STORAGE MAP

02/16/68

PAGE 8

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
COLOR	11476	I	NRUMB	11477	I	RYCNT	11500	I
LENGTH	11501	R	NR	11502	I	NRAYS	11503	I
I	11504	I	DELL	11505	R	DELC	11506	R
CHAR	11507	R	KBAR	11510	I	K	11511	I
KL	11512	I	SMAX	11513	R	SMIN	11514	R
NSPOTS	11515	I	BF1	11516	R	DUMMY	11517	R
J	11520	I	HEIGHT	11521	R	XTHIN	11522	R
XTHAX	11523	R	YTHIN	11524	R	YTHAX	11525	R
YMIN	11526	R	XMAX	11527	R	YMIN	11530	R
MAX	11531	R	PLX	11532	R	PLY	11533	R
NYORG	11534	I	NYORG	11535	I	INDRAY	11536	I
XB	11537	R	YB	11540	R			

ENTRY POINTS

START SECTION 11

SUBROUTINES CALLED

LATT	SECTION	12
GETLC	SECTION	15
MERID	SECTION	18
SMPOP	SECTION	21
SCALEK	SECTION	24
.UNDB.	SECTION	27
.UN11.	SECTION	30
.FBDT.	SECTION	33
E.1	SECTION	36
E.4	SECTION	39
CC.3	SECTION	42

PRCTL	SECTION	13
.FSLO.	SECTION	16
SIN	SECTION	19
.FWRB.	SECTION	22
.FRDB.	SECTION	25
.FFIL.	SECTION	28
.FWLR.	SECTION	31
.UN12.	SECTION	34
S-2	SECTION	37
CC.1	SECTION	40
CC.4	SECTION	43

.FWRD.	SECTION	14
.FRWT.	SECTION	17
COS	SECTION	20
.FEFT.	SECTION	23
.FXEN.	SECTION	26
.FCNV.	SECTION	29
.FBLT.	SECTION	32
.FRLR.	SECTION	35
E.3	SECTION	38
CC.2	SECTION	41
SYSLOC	SECTION	44

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION
2000	22A	12110
32010	FORMAT	11573
6000	118A	12475
6500	77A	12344
8200	111A	12436
32050	FORMAT	11646
12500	179A	13032
11500	165A	12730
13000	192A	13131
15000	254A	13527
14200	230A	13354
14800	253A	13517
32100	FORMAT	11774
32120	FORMAT	12010
17200	314A	13740
18000	332A	13773

EFN	IFN	LOCATION
5000	48A	12236
32020	FORMAT	11620
6000	69A	12330
7500	94A	12374
8400	115A	12456
9000	127A	12512
11200	161A	12713
12000	166A	12756
16000	269A	13575
14000	227A	13335
14400	249A	13475
32080	FORMAT	11760
32110	FORMAT	12002
19000	346A	14055
17000	311A	13734
18200	340A	14017

EFN	IFN	LOCATION
4000	34A	12151
32030	FORMAT	11626
7000	80A	12350
8000	103A	12412
32040	FORMAT	11643
10000	135A	12533
11000	158A	12702
32000	351A	14061
32080	FORMAT	11652
32070	FORMAT	11723
14600	251A	13506
32090	FORMAT	11767
20000	348A	14057
16900	303A	13720
17900	325A	13755
18400	343A	14055

TR-67-700-10-2
Page III-7-14
Program Listing
Link 7

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
START

STORAGE NAME

02/16/68

PAGE 9

18500 349A 14282
THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 14112.

TR-67-700-10-2
Page III-8-1
Program Listing
Link 8

Section 8

LINK 8

TR-67-700-10-2
Page III-8-2
Program Listing
Link 8

INF.ND10501.44800-0.77387.AM IC

ISJOB LINK1 7094 A 02/16/68

PAGE 10

00
STITLE SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
SIMP TC STAR02 H96.KR7 LINK 8 (STAR0)

80001000

SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
 STARRZ - EFN SOURCE STATEMENT - IFN(8) -

02/16/68

PAGE 1

```

SUBROUTINE STARRZ                                88001800
C
C   *** INPUTS ***                                88001800
C NPRTB NUMBER OF PARAMETERS TO BE PERTURBED.    (PERTB COMMON) 88001810
C PERTB PERTURBATION MATRIX CONTAINING PARAMETER INDEX (PERTB COMMON) 88001820
C   AND PERCENT AND THE PARAMETER SYMBOL.        88001830
C REFOCS FLAG DEFINING THE TYPE OF REFOCUSING DESIRED. (PERTB COMMON) 88001840
C
C   *** OUTPUTS ***                                88001850
C   VARIOUS RAY STATISTICS FOR NOMIAL AND PERTURBED SYSTEMS. 88001860
C
C   *** DESCRIPTION ***                            88001870
C STARRZ PERFORMS THE COMPUTATIONS FOR THE SENSITIVITY ANALYSIS. 88001880
C STATISTICAL DATA FOR THE NOMIAL SYSTEM IS CALCULATED FOR EACH 88001890
C COLOR AND OBJECT POINT. THE CALCULATIONS ARE THEN REPEATED 88001700
C FOR EACH PERTURBED PARAMETER. A REFOCUS OPTION MAY BE 88001710
C SPECIFIED TO OBTAIN THE PLANE LOCATION FOR MINIMUM RMS X. 88001720
C RMS Y, OR SPOT SIZE.                            88001730
C
COMMON / DATA / NCNTRL, CNTRL(10), TITLE(12), DATE(3), PUNCID,  DT001000
1 DATA(3403)                                       DT001100
DIMENSION WOBJH(7), WCLR(7), WINHT(7), EINH(7), CIMPL(7),  DT002000
1 WCLRS(8), NDSGN(4), NDSGV(11), LATT(3,26), BOUNDS(3,100),  DT002100
2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATT(3,26),  DT002200
3 ICURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100)  DT002300
EQUIVALENCE (DATA(1), LMODE), (DATA(2), NRRAYS), (DATA(3),  DT101000
1 NSLCS), (DATA(4), NCLRS), (DATA(5), NJAIL), (DATA(6),  DT101100
2 ISUBT), (DATA(7), NSUBP), (DATA(8), NIPLN), (DATA(9),  DT101200
3 IMODE), (DATA(10), NSPLN), (DATA(11), NOBJH), (DATA(12),  DT101300
4 NSURF), (DATA(13), AFLAG), (DATA(14), DELY), (DATA(15),  DT101400
5 FNUMB), (DATA(16), FLNGH), (DATA(17), WFLGH), (DATA(18),  DT101500
6 ZETA), (DATA(19), HEXPP), (DATA(20), DEXPP), (DATA(21),  DT101600
7 WEXPP), (DATA(22), DLPLN), (DATA(23), OMG2), (DATA(24),  DT101700
8 OMG1), (DATA(25), DELD), (DATA(26), EPRAD), (DATA(27),  DT101800
9 PSCAL), (DATA(28), OMGAF), (DATA(29), SPFEA), (DATA(30),  DT101900
A DUMIN), (DATA(31), C'IST), (DATA(32), HO), (DATA(33),  DT102000
B DELH), (DATA(34), SYSX), (DATA(35), WXDIR), (DATA(36),  DT102100
C WYDIR), (DATA(37), ROTAN), (DATA(38), NDSGN(1)), (DATA(42),  DT102200
D NDSGV(1)), (DATA(53), NOBJH(1)), (DATA(60), WCLR(1)),  DT102300
E (DATA(67), WINHT(1)), (DATA(74), EINH(1)), (DATA(81),  DT102400
F CIMPL(1)), (DATA(88), WCLRS(1)),  DT102500
G (DATA(94), LATT(1,1), ILATT(1,1)), (DATA(102), SURFC(1,1),  DT102600
H ISURFC(1,1)), (DATA(2102), DESGN(1,1), IDESGN(1,1)),  DT102700
I (DATA(2602), SUBST(1), ISUBST(1)), (DATA(3102), BOUNDS(1,1),  DT102800
J IBNDS(1,1)), (DATA(3403), NCOND)  DT102900
INTEGER CNTRL, AFLAG, DATE  DT201000
REAL LATT  DT202000
C
COMMON / TMPATT / NHISS, NRFLCT, NVIGN, NRRAYS, XBAR, YBAR, XSTAR, L3001000
1 YSTAR, RATAR, SJSTAR, XCURL(201), YCURL(201), MXBAR, KYBAR,  L3001200
2 NYBAR, KYBAR, AX, BX, CX, AY, BY, CY, ACAP, BCAP, CCAP  L3001400
REAL MXBAR, KYBAR, NYBAR, KYBAR  L3002000
C
COMMON / PRT / LINZ, PAGE  PT002500
INTEGER PAGE  PT002700

```

SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
STAR8Z - EPN SOURCE STATEMENT - IFN(8) -

02/18/68

PAGE 2

C			
C	THIS BLOCK OF COMMON CONTAINS PERTURBATION VALUES	PS001000	
	COMMON / PERTB / PERTB(3D.4), NPERTB, REFOCS	PS002000	
	EQUIVALENCE (PERTB(1.1), IPERTB(1.1))	PS003000	
	DIMENSION IPERTB(3D.4)	PS004000	
	INTEGER REFOCS	PS005000	
C			
	COMMON/AZOBJ/THTR,SNTHTR,CSTHTR	AZ004000	
C			
	DIMENSION STATN(6.9.7), TEMPS(400), GMX(6.17), CL3(425),	S8002000	
	1 GAMMA(5), SJN(7), T1(30), DELTAT(30), IGMX(6.17), ISTATN(6.9.7)	S8002200	
C			
	INTEGER SYMB1, SYMB2	S8003000	
	EQUIVALENCE (NMISS, CL3(1)), (STATN(1.1.1), ISTATN(1.1.1)),	S8003600	
	1 (GMX(1.1), IGMX(1.1))	S8003700	
C			
C	START OF STAR8 SUBROUTINE	S8006000	
C			
	DO 2000 I = 1, NPERTB	S8007000	
	INDX = IPERTB(I,2)	S8007200	
	T1(I) = .01 * PERTB(I,1)	S8007400	
	TMP = DATA(INDX)	S8007600	
	IF (TMP .NE. 0.0) T1(I) = TMP * (1. + T1(I))	S8007800	
	2000 DELTAT(I) = T1(I) - TMP	S8008000	
	THTR = 0.	S8008600	
	SNTHTR = 0.	S8008800	
	CSTHTR = 1.	S8009000	
C			
C	COMPUTE NOMINAL BACK FOCUS	S8021000	
	CALL MERID(1, BFN, DUMMY)	S8021200	
C			
C	COMPUTE NOMINAL OBJECT AZIMUTH	S8021600	19
	THTR = ROTAN * 3.14159265 / 180.	S8021800	
	SNTHTR = SIN(THTR)	S8022000	21
	CSTHTR = COS(THTR)	S8022200	22
	DO 3000 J = 1, NOBJH	S8023000	
	HEIGHT = HO + FLOAT(J-1) * DELH	S8023200	
	DO 4000 I = 1, NCLRS	S8023400	
C			
C	ZERO OUT LINK3 COMMON USING EQUIVALENCED ARRAY	S8023600	
	DO 3000 K = 1, 425	S8023800	
	3000 CL3(K) = 0.0	S8024000	
C			
C	TRACE RAY PATTERN	S8024400	
	CALL SHPOP(I, J, HEIGHT, BFN, 4, DUMMY)	S8024600	33
	STATN(I,1,J) = XBAR	S8024800	
	STATN(I,2,J) = YBAR	S8025000	
	STATN(I,3,J) = XSTAR	S8025200	
	STATN(I,4,J) = YSTAR	S8025400	
	STATN(I,5,J) = RSTAR	S8025600	
	ISTATN(I,6,J) = NRAYS8	S8025800	
	ISTATN(I,7,J) = NMIS8	S8026000	
	ISTATN(I,8,J) = NRFLCT	S8026200	
	ISTATN(I,9,J) = NVIS8	S8026300	
	4000 CONTINUE	S8026400	
	RJN(J) = SJSTAR	S8027000	

SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
STAR92 - EFM SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 3

5000 CONTINUE	58027200	
NR = NRAY8	58028000	
IF (LMODE .EQ. 0) NR = NR + NR	58028200	
RNDX = 3 + REFOCS + 2	58028400	
CALL PRCTCL	58028600	
C		54
WRITE (6, 32010)	58028800	55
32010 FORMAT(1H0, 13X, 45H***** UNPERTURBED SYSTEM RAY STATISTICS *****	58029000	
1 //74H OBJECT POINT COLOR RAYS TRACED SUCCESSES MISSES REFLECTI	58029200	
CTIONS VIGNETS)	58029400	
C		
DO 6000 J = 1, NOBJH	58041000	
DO 6000 I = 1, NCLRS	58041200	
WRITE (6, 32020) J, I, NR, (ISTATN(I,K,J), K = 6, 9)	58041400	60
32020 FORMAT(1H0, 17, 110, 6X, 13, 6X, 13, 6X, 13, 6X, 13, 6X, 13)	58041600	
6000 CONTINUE	58041800	
C		
DO 15000 K = 1, NPERTB	58042000	
PERCNT = PERTB(K,1)	58042200	
INDX = IPERTB(K,2)	58042400	
SYMB1 = IPERTB(K,3)	58042600	
SYMB2 = IPERTB(K,4)	58042800	
C		
SAVE NOMINAL VALUE OF KTH. PARAMETER	58043000	
TMP = DATA(INDX)	58043200	
C		
PERTURB KTH. PARAMETER	58043600	
DATA(INDX) = T1(K)	58043800	
C		
COMPUTE BACK FOCUS - COLOR 1	58044000	
THTR = 0.	58044200	
SNTHTR = 0.	58044400	
CSTHTR = 1.	58044600	
CALL MERID(1, BF1, DUMMY)	58045000	
C		62
SET AZIMUTH OF OBJECT	58045200	
THTR = ROTAN * 3.14159265 / 180.	58045400	
SNTHTR = SIN(THTR)	58045600	64
CSTHTR = COS(THTR)	58045800	65
LINE = 60	58046000	
C		
DO 14000 J = 1, NOBJH	58046200	
HEIGHT = HO + FLOAT(J-1) * DELH	58046400	
DO 8000 I = 1, NCLRS	58047000	
C		
ZERO LINKS COMMON	58047200	
DO 7000 LK3 = 1, 425	58047400	
7000 CL3(LK3) = 0.0	58047600	
CALL SMPOP(I, J, HEIGHT, BF1, .1, TEMPS)	58047800	
C		66
LOAD GHX FROM LINKS COMMON	58048000	
DO 7500 LK3 = 413, 425	58048200	
7500 GHX(I,LK3-412) = CL3(LK3)	58048400	
IGHX(I,14) = NRAY8B	58048600	
IGHX(I,15) = NMISS	58048800	
IGHX(I,16) = NREFLCT	58049000	
IGHX(I,17) = NVIGN	58049200	
8000 CONTINUE	58049400	
IF (REFOCS .EQ. 0) GO TO 8700	58051000	
SUM8 = 0.	58051200	
SUM9 = 0.	58051400	

SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
STAR8Z - EFN SOURCE STATEMENT - IPN(8) -

02/16/68

PAGE 4

```

DO 8500 I = 1, NCLR8                                80081600
SUMA = SUMA + GMX(I,KNDX)                            80081800
8500 SUMB = SUMB + GMX(I,KNDX+1)                    80082000
FLAMDA = - SUMB / SUMA                              80082200
8700 DO 14000 I = 1, NCLR8                          80082400
IF ( YSTATN(I,6,J) .EQ. 0 ) GO TO 14000            80082600
IF ( LINE .LE. 48 ) GO TO 9000                    80082800
C          BEGIN NEW PAGE                          80083000
CALL PRCTCL                                        80083200
C
WRITE ( 6, 32030 ) K, SYMB1, SYMB2, TMP, T1(K), DELTAT(K), PERCENTS 80083400
32030 FORMAT( 1HD, 42X, 9HPARAMETER, I3, 1X, 2A6 // 24X, 30HNOMINAL VALUS 80083600
1E PERTURBED VALUE, 4X, 10HDIFFERENCE, 4X, 15HPER CENT CHANGE /
2 21X, 2E16.7, 2E17.7 )                          80083800
LINE = 5                                           80084000
9000 WRITE ( 6, 32040 ) J, I, ( STATN(I,KT,J), KT = 1,5 ), 80085000
1 ISTATN(I,6,J)                                    80085200
32040 FORMAT( 1HD, 12HOBJECT POINT, I2, 2X, 5HCOLOR, I2 // 34X, 4HXBAR, 80085400
1 12X, 4HYBAR, 12X, 5HRMS X, 11X, 5HRMS Y, 8X, 9HSPOT SIZE, 5X,
2 4HKAYS / 9X, 18HNOMINAL VALUE....., 5E16.7, I6 ) 80085600
LINE = LINE + 5                                    80086000
IF ( IGMX(I,14) .NE. 0 ) GO TO 9500              80086200
C
WRITE ( 6, 32050 ) ( IGMX(I,KT), KT = 14, 17 )    80086400
32050 FORMAT( 1HD, 31X, 39HSUCCESSSES MISSES REFLECTIONS VIGNETS / 80086600
1 35X, I3, 6X, I3, 6X, I3, 6X, I3 )              80086800
LINE = LINE + 3                                    80087000
GO TO 14000                                        80087200
9500 X0BAR = STATN(I,1,J)                          80088000
Y0BAR = STATN(I,2,J)                              80088200
X0STAR = STATN(I,3,J)                             80088400
Y0STAR = STATN(I,4,J)                             80088600
R0STAR = STATN(I,5,J)                             80088800
ELAMDA = 8JSTAR                                    80089000
ASSIGN 10200 TO IGO                                80089200
C
C          SPECIAL LOGIC SECTION                    80081000
C
9800 XBAR = GMX(I,1) * ELAMDA + GMX(I,2)           80081200
YBAR = GMX(I,3) * ELAMDA + GMX(I,4)               80081400
XSTAR = SQRT( GMX(I,5) * ELAMDA * ELAMDA + 2. * GMX(I,6) * ELAMDA 80081600
1 + GMX(I,7) )                                     80081800
YSTAR = SQRT( GMX(I,8) * ELAMDA * ELAMDA + 2. * GMX(I,9) * ELAMDA 80082000
1 + GMX(I,10) )                                    80082200
RSTAR = SQRT( GMX(I,11) * ELAMDA * ELAMDA + 2. * GMX(I,12) * 80082400
1 ELAMDA + GMX(I,13) )                             80082600
DXBAR = XBAR - X0BAR                               80082800
DYBAR = YBAR - Y0BAR                               80083000
DXSTAR = XSTAR - X0STAR                           80083200
DYSTAR = YSTAR - Y0STAR                           80083400
DRSTAR = RSTAR - R0STAR                           80083600
DO 10000 L = 1, 9                                  80083800
10000 GAMMA(L) = 0.                                80084000
IF ( X0BAR .NE. 0. ) GAMMA(1) = ( 100. * DXBAR ) / X0BAR 80084200
IF ( Y0BAR .NE. 0. ) GAMMA(2) = ( 100. * DYBAR ) / Y0BAR 80084400
IF ( X0STAR .NE. 0. ) GAMMA(3) = ( 100. * DXSTAR ) / X0STAR 80084600

```

```

SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
STAR8Z - EPN SOURCE STATEMENT - IPN(8) -
                                02/16/68
                                PAGE 9

IF ( YOSTAR .NE. 0. ) GAMMA(4) = ( 100. + DYSTAR ) / YOSTAR      80084000
IF ( ROSTAR .NE. 0. ) GAMMA(5) = ( 100. + DRSTAR ) / ROSTAR      80085000
C
C      END OF SPECIAL SECTION      80086000
C
GO TO IGO.( 10200, 13000 )      80086200
10200 WRITE ( 6, 32080 ) XBAR, YBAR, XSTAR, YSTAR, RSTAR, IGMX(I,14), 80091000
      1 DXBAR, DYBAR, DXSTAR, DYSTAR, DRSTAR, ( GAMMA(L), L = 1, 5 ) 80091200      191
32080 FORMAT( 1H , 8X, 18PERTURBED VALUE..., 5E16.7, 16 / 8X, 18DIFFER 80091400
IENCE....., 5E16.7 / 8X, 18SENSITIVITY....., 5E16.7 )
      LINE = LINE + 3      80091600
      NREFC = REFOCS + 1      80092000
C
GO TO ( 14000, 10300, 11000, 11500 ), NREFC      80092200
10300 WRITE ( 6, 32070 ) FLANDA      80092400      199
32070 FORMAT( 1H , 8X, 26MIN. RMS X PLANE LOCATION=, E14.7 )
      GO TO 12000      80092800
11000 WRITE ( 6, 32080 ) FLANDA      80093000      201
32080 FORMAT( 1H , 8X, 26MIN. RMS Y PLANE LOCATION=, E14.7 )
      GO TO 12000      80093400
11500 WRITE ( 6, 32090 ) FLANDA      80093600      203
32090 FORMAT( 1H , 8X, 30MIN. SPOT SIZE PLANE LOCATION=, E14.7 )
12000 ELANDA = FLANDA      80094000
      ASSIGN 13000 TO IGO      80094200
      GO TO 9800      80094400
C
13000 WRITE ( 6, 32050 ) XBAR, YBAR, XSTAR, YSTAR, RSTAR, IGMX(I,14), 80095000
      1 DXBAR, DYBAR, DXSTAR, DYSTAR, DRSTAR, ( GAMMA(L), L = 1, 5 ) 80095200      207
      LINE = LINE + 4      80095400
14000 CONTINUE      80095600
C
C      RESTORE PARAMETER TO NOMINAL VALUE      80096000
DATA(INOX) = TMP      80096200
15000 CONTINUE      80097000
C
RETURN      80098000
END      80099500

```

SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
 STARR2

STORAGE MAP

02/16/68

PAGE 8

SUBROUTINE STARR
 COMMON VARIABLES

COMMON BLOCK			DATA	ORIG IN	00001	LENGTH	08666	
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ICNTRL	00000	I	CONTRL	00001	I	TITLE	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
UNODE	00033	I	NRAYS	00034	I	NSLCS	00039	I
ICLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00049	I	NSURF	00046	I
FLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
HEXPP	00055	R	DEXPP	00056	R	WEXPP	00057	R
DLPLN	00060	R	ONGAR	00061	R	ONGA1	00062	R
DELD	00063	R	EPRAD	00064	R	PSCAL	00065	R
ONGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
ODIST	00071	R	HO	00072	R	DELH	00073	R
SYBKE	00074	R	WXDIR	00075	R	WYDIR	00076	R
RTAN	00077	R	NDSCN	00100	I	NDSCV	00104	I
NOBJH	00117	R	WCLRH	00126	R	WINHT	00139	R
EINH7	00144	R	CIMPL	00153	R	WCLRS	00162	R
LATTC	00170	R	ILATTC	00170	I	SURFC	00320	R
ISURFC	00320	I	DESGN	04240	R	IDESGN	04240	I
SUBST	05224	R	ISUBST	05224	I	BOUNDS	08210	R
IBNDS	08210	I	NCOND	08665	I			
COMMON BLOCK			THPATT	ORIG IN	08667	LENGTH	00851	
NRAYS	00003	I	NRFLCT	00001	I	NVIGN	00002	I
XSTAR	00006	R	XBAR	00004	R	YBAR	00005	R
SJSTAR	00011	R	YSTAR	00007	R	RSTAR	00010	R
KXBAR	00634	R	XCURL	00012	R	YCURL	00323	R
KYBAR	00637	R	KXBAR	00635	R	HYBAR	00636	R
OX	00642	R	AX	07640	R	OX	00641	R
CY	00645	R	AY	00643	R	BY	00644	R
CCAP	00650	R	ACAP	00646	R	BCAP	00647	R
			CLS	00000	R			
COMMON BLOCK			PRNT	ORIG IN	07540	LENGTH	00002	
LINE	00000	I	PAGE	00001	I			
COMMON BLOCK			PERTB	ORIG IN	07542	LENGTH	00172	
PERTB	00000	R	NPERTB	00170	I	REFOCS	00171	I
PERTB	00000	I						
COMMON BLOCK			AZOBJ	ORIG IN	07734	LENGTH	00000	
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00000	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
STATN	07737	R	TEMPS	10577	R	GNX	10531	R

SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
 STARSZ

02/16/68

PAGE 7

STORAGE MAP

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
GAMMA	11817	R	SJN	11824	R	T1	11833	R
DELTA	11871	R	IGMX	10931	I	ISTATN	07737	I
UNDIMENSIONED PROGRAM VARIABLES								
SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SYMB1	11827	I	SYMB2	11830	I	I	11831	I
INDX	11832	I	TMP	11833	R	BFN	11834	R
DUMMY	11838	R	J	11838	I	HEIGHT	11839	R
K	11840	I	MR	11841	I	KNDX	11842	I
PERCNT	11843	R	BF1	11844	R	SUMA	11848	R
SUM8	11848	R	FLANDA	11847	R	XOBR	11850	R
YOBAR	11851	R	XOSTAR	11852	R	YOSTAR	11853	R
ROSTAR	11854	R	ELANDA	11855	R	IC	11856	I
OKBAR	11857	R	DYBAR	11860	R	DX	11861	R
OYSTAR	11862	R	DRSTAR	11863	R	NREFC	11864	I

ENTRY POINTS

STAR8 SECTION 13

SUBROUTINES CALLED

HERIO	SECTION	14
SNDOP	SECTION	17
SQRT	SECTION	20
.FFIL.	SECTION	23
E.2	SECTION	26
CC.1	SECTION	29
CC.4	SECTION	32

SIN	SECTION	15
PRTCTL	SECTION	18
.FXEM.	SECTION	21
.FCNV.	SECTION	24
E.3	SECTION	27
CC.2	SECTION	30
SYSLOC	SECTION	33

CS8	SECTION	16
.FWRD.	SECTION	19
.UNDB.	SECTION	22
E.1	SECTION	25
E.4	SECTION	28
CC.3	SECTION	31

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
2000	14A	12172	5000	47A	12352	4000	44A	12346
3000	30A	12310	32010	FORMAT	11714	6000	65A	12447
32020	FORMAT	11744	15000	217A	13445	14000	213A	13434
8000	108A	12641	7000	93A	12603	7500	100A	12624
8000	119A	12672	8500	118A	12660	9000	134A	12763
32030	FORMAT	11756	32040	FORMAT	12007	9500	150A	13047
32050	FORMAT	12045	10200	191A	13250	9800	157A	13086
10000	178A	13201	13000	207A	13750	32060	FORMAT	12085
10500	199A	13330	11000	201A	13741	11500	203A	13352
32070	FORMAT	12111	12000	204A	13362	32080	FORMAT	12122
32090	FORMAT	12133						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 13504.

TR-67-700-10-2
Page A-1

APPENDIX A

SC4020 PLOT PACKAGE

SPLOT/S-C 4020 FORTRAN IV PLOT ROUTINES

The following set of FORTRAN IV plot routines are very similar to the FORTRAN - II Version III routines. The following subroutines have been changed:

1. SINTRP no longer has a variable length calling sequence.
2. WRITE no longer uses a dummy tape number.
3. In many of the routines the user is no longer allowed to use either integers or floating point numbers. He is restricted to the type specified in the calling sequence.
4. Provisions have been made in SDINIT to use the EXPAND image mode or REDUCE image mode of the SC 4020.

Although the tape number NT appears in the calling sequence of many subroutines, it is not used. It was left for compatibility purposes. All plot output is put on SYSPL1 (unit 18).

IDENTIFICATION

SPLIT IV SUBROUTINES / S-C 4020 plotting and control for FORTRAN IV

Robert Jirka (JPL)

Lowell Smith (JPL)

Chuck Lawson (JPL)

PURPOSE

SPLIT is a basic set of subroutines to

1. prepare decimal, Hollerith or curve data and write a binary tape for use on the S-C 4020.
2. provide for control of the S-C 4020.

While preparing curve data, linear interpolation can be used to fill in between the points of the curve. Provisions are made for points of the curve which are off the plotting surface to be saved and used for interpolation purposes but not plotted.

METHOD

The S-C 4020 plotting surface is square with coordinates of (0,0) for the upper left hand corner and (1023, 1023) for the lower right hand corner

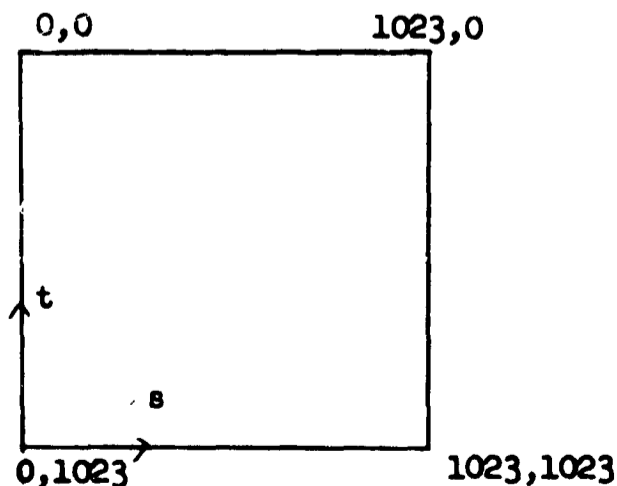


figure 1. S-C 4020 Plotting Surface

For any labelling or printing the S-C coordinate system is used.

For plotting curves another coordinate system specified by the user is considered. The bounds of the surface are specified by giving the coordinates (x_0, y_0) of the lower left corner and the coordinates (x_m, y_m) of the upper right corner.

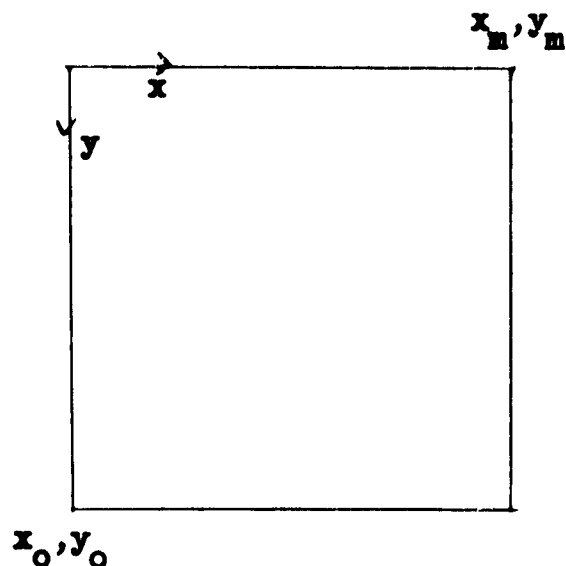


figure 2. Users Plotting Surface

For plotting curves everything will be relative to the users plotting surface.

To plot a curve the user supplies

1. The coordinates (x_0, y_0) and (x_m, y_m) for the corners of the plotting surface.
2. The points (s_0, t_0) and (s_m, t_m) on the S-C 4020 plotting surface at which to locate (x_0, y_0) and (x_m, y_m) respectively.
3. A one dimensional array where the S-C 4020 instructions generated by the subroutines are to be stored. This array will subsequently be called a buffer area.
4. If desired, a grid to superimpose over the graph.
5. Points of the curve (one at a time).

6. When the last point has been supplied, a signal to the subroutines to dump the S-C 4020 instructions onto a specified tape unit.

The following FORTRAN statements are used to plot a curve

1. CALL SDINIT (BUF,NB,X,S,NT,...)
2. CALL GRID (NT,NX,NY,XO,YO,...) or CALL BGRID (BUF,NX,NY,K)
3. CALL PLAB (BUF,NDX,FMT1,NDY,FMT2)
4. CALL SCALE (K,A,B,C,D) $\left\{ \begin{array}{l} \text{SCALEK HAS BEEN ADDED TO} \\ \text{LIBRARY -- SEE SEPARATE WRITEUP} \\ \text{67-41-01} \end{array} \right.$
5. CALL SDNPUT (X,Y,BUF,L)
6. CALL PPL~~OT~~ (X,Y,BUF,1H*)
7. CALL SINTRP (BUF)
8. CALL STERM (NT, BUF,...)

The following statements are used for output of Hollerith/decimal data.

9. CALL SETPL (N) - CALL RESPL $\left\{ \begin{array}{l} \text{NO LONGER REQUIRED IF} \\ \text{UNIT IN THE WRITE STATEMENT} \\ \text{REFERS TO UNIT 18} \end{array} \right.$
10. WRITE (UNIT,FMT) 'List'
11. CALL SPRINT (NF,NXL,NYL, +NS,NHX,...X_n,...)
12. CALL SPRNTA (NF,NXL,NYL,+NS,A,N,...)
13. CALL SLABEL

The following statements are used to control the S-C 4020.

14. CALL CAMERA (NT,NO)
15. CALL ADV (NT)
16. CALL F~~ORM~~ (NT)
- ~~17. CALL EXPOSE (NT,NO) NOT AVAILABLE~~

USAGE

In any of the following FORTRAN statements the restrictions imposed by FORTRAN in naming variables will be used and the type of variable (fixed or floating) will not always be mentioned.

It is assumed the user is familiar with the techniques of using Hollerith text in the parameter list of a subroutine.

Furthermore it is assumed the user has familiarized himself with the S-C 4020 by reading the S-C 4020 PROGRAMMING MANUAL.

Before curves are plotted certain control routines have to be used:

1. CAMERA to select the cameras for the desired output.
2. SLABEL to label the user's job for identification.

1. SDINIT: The FORTRAN statement

CALL SDINIT (BUF,NB,X,S,NT,...) will set up the buffer area BUF for the subroutines SDNPUT,PLAB,PPL~~OT~~,SINTRP,STERM

where:

BUF is the name of a buffer area.

NB is the number of locations in the buffer area and the sign of NB is used to indicate the image mode. A minus (-) NB will use the expanded image mode. Positive (+) NB will use the reduced (standard) image code.

X is a one dimensional array containing the coordinates the user wishes the curve plotting surface to have.

$$X(1) = x_0$$

$$X(2) = y_0$$

$$X(3) = x_m$$

$$X(4) = y_m$$

S is a one dimensional array containing the coordinates of points on the S-C plotting surface at which to locate the points (x_0, y_0) and (x_m, y_m) .

(x_0, y_0) will be at the point (S(1), S(2))

(x_m, y_m) will be at the point (S(3), S(4))

NT is a FORTRAN tape number used to dump the buffer if it should become filled. This is not used in FORTRAN IV.

The parameter list is variable and the 5-tuple (BUF,NB,X,S,NT) may be repeated using other variable names or constants.

SDINIT uses 20 locations in BUF to store information which will later be used by SDNPUT and other subroutines.

Each buffer area to be used for plotting must appear in the parameter list of a CALL SDINIT (List) statement.

Once a buffer area BUF has been associated with a coordinate system by using SDINIT, none of the other SPLOT routines will destroy this association unless SDINIT is used again.

S(1) must be less than S(3) and S(2) greater than S(4) because of the upside down character of the S-C plotting surface (see figure 1).

Any element in the S-array must be ≥ 0.0 and ≤ 1023.0

The only restrictions on the X-array are that $x(1) \neq x(3)$ and $x(2) \neq x(4)$.

If $x_0 = x_m$ or $y_0 = y_m$, the diagnostic

ERROR IN COORDINATES DEFINING PLOT SURFACE

will be output on A3 and return is made to the user.

Depending on the sign of NB, a S-C 4020 instruction to either EXPAND image or REDUCE image will be inserted in BUF as the first instruction.

2.1 GRID: The FORTRAN statement

CALL GRID (NT,NX,NY,XO,YO) will draw a grid using the S-C 4020 instructions GXA and GYA. (see page 24 S-C 4020 Programming Manual.)

where:

NT is a FORTRAN tape number.

NX is the number of intervals for the x-axis.

NY is the number of intervals for the y-axis.

XO is the x-coordinate of the lower left corner of the grid desired by the user.

YO is the y-coordinate of the lower left corner of the grid desired by the user.

The parameter list is variable and if any parameter (which may be fixed or floating) is omitted, all following parameters must be omitted and their values are assumed to be

NT = 18

NX = 20

NY = 20

XO = 80

YO = 944

Thus, if a CALL GRID (NT,NX) statement were used then NY,XO,YO would be assumed to be 20,80 and 944 respectively.

All output is on SYSPL1 (FORTRAN 18)

2.2 BGRID: The Fortran statement

CALL BGRID (BUF,NX,NY,K) where

BUF is a buffer region previously initialized by SDINIT.

NX is an integer of the form $m_x + 1000n_x$ indicating first that the x-axis is to be divided into n_x intervals and then into m_x intervals.

NY is an integer of the form $m_y + 1000n_y$ indicating first that the y-axis is to be divided into n_y intervals and then into m_y intervals.

K is an integer indicator which is either 0 or positive. The last grid line parallel to the y-axis will not be drawn if $K=0$.

will generate and store in BUF the S-C 4020 instructions for generating a grid pattern on the plotting surface defined previously by a CALL SDINIT.

The grid will have $n_x + 1$ grid lines parallel to the y-axis and then $m_x + 1$ grid lines drawn over these. m_x may be zero in which case a second set of grid lines will not be drawn. m_x is intended for darkening certain major grid lines.

m_y and n_y dictate the grids parallel to the x-axis in the same manner as m_x and n_x for the grids parallel to the y-axis.

If $K=0$ then the last grid line parallel to the y-axis is not generated. This would probably be used when the S-C 4020 is operating in the EXPAND IMAGE MODE and the user is making a continuous strip chart.

If $K \neq 0$ then all grid lines will be drawn.

3. PLAB: The FORTRAN statement

CALL PLAB (BUF,NDX,FMT1,NDY,FMT2) where

BUF is the name of a buffer area previously initialized by
SDINIT.

NDX is the number of equal divisions considered on the
x-axis. NDX+1 labels will be generated. If NDX = 0, no
x-axis labels will be generated.

FMT1 is a hollerith array containing a format statement for
the conversion of the labels for the x-axis. Only F or E
type conversions are allowed.

NDY is the number of divisions considered on the y-axis.
NDY + 1 labels will be generated. If NDY = 0, then no y-axis
labels will be done.

FMT2 is a hollerith array containing a format statement for
the conversion of the labels for the y-axis. Only F or E
type conversions are allowed.

will label a grid with the information taken from the array BUF.
The information used by PLAB is inserted in BUF by SDINIT. The
generated label information is put back into BUF. To put this
information onto tape, STEPM must be used. The labels are centered
on the grid lines.

Example:

CALL PLAB (BUF,10,6H(F6.2),10,6H(F6.3)) would put 11 labels
on the x-axis with 2 places after the decimal point and 11
labels on the y-axis with 3 places after the decimal point.

4. SCALE

PURPOSE

To compute upper and lower limits for a variable which is to be plotted so that the plot will be compatible with the scale lines of the graph paper.

METHOD

The variable, say X , is specified as lying between A and B and the X scale on the graph paper is specified as having a range of K major scale units, i.e., $K+1$ major scale lines.

X -values of C and D are to be assigned to the first and last major scale lines respectively, so that the total range $(D-C)$ will be as small as possible, subject to the following conditions.

1. $C \leq \min(A, B) < \max(A, B) \leq D$.

(a tolerance of about $1/1000$ of a major scale unit (U) is allowed in this condition to preclude the possibility of round off error causing an unnecessary increase in $(D-C)$.)

2. The major scale unit, which will be $U = \frac{D-C}{K}$, is to be of the form 10^p , $2 \cdot 10^p$, or $5 \cdot 10^p$ for some integer p .

3. Each major scale line is to correspond to an X value which is an integer multiple of 10^p , where this p is the same as in condition 2. The stronger, and more desirable, condition that each major scale line should correspond to an integer multiple of U will be used if it does not require an increase in $(D-C)$.

4. If $A*B \leq 0$, then one of the major scale lines is to correspond to $X=0$.

RESTRICTIONS

If $K \leq 0$ or $A=B$ then SCALE sets $C=\min(A,B)$ and $D=\max(A,B)$ and makes the normal return.

The combination of $K=1$ and $A*B < 0$ is not illegal but in this case conditions 1. and 4. are incompatible. SCALE treats this case as though $k=10$, which effectively preserves condition 1. but violates condition 4.

USAGE

CALL SCALE (K,A,B,C,D)

where

For INPUT

K = Number of major scale units. $K \geq 1$.

A,B = Limits on the range of the variable to be plotted.

Either $A < B$ or $A > B$ is acceptable.

For OUTPUT

C,D = Desirable lower and upper limits, respectively, for use in plotting.

EXAMPLE OF USAGE

Suppose a curve $y = f(X)$ is to be plotted on graph paper which has 100x150 minor scale units and 10x15 major scale units.

Suppose X min, Y min, X max, and Y max are stored respectively in X(I), I = 1,2,3,4. The statement CALL DUNIT (BUF, 1000, X, NT) will, for general values of X min, etc., lead to a graph which bears no simple relation to the scale lines on the graph paper and is, therefore, difficult to read.

This can be remedied by inserting before the CALL DUNIT the two statements

```
CALL SCALE (15,X(1),X(3),X(1),X(3))
```

```
CALL SCALE (10,X(2),X(4),X(2),X(4))
```

As a second example, suppose it is desired to leave a one inch margin around the plot, say for labeling. The following statements will provide the desired positioning of the plot.

```
CALL SCALE (13,X(1),X(3),X(1),X(3))
```

```
CALL SCALE (8,X(2),X(4),X(2),X(4))
```

```
UNIT = (X(3)-X(1))/FLOAT (13)
```

```
X(1) = X(1)-UNIT
```

```
X(3) = X(3)+ UNIT
```

```
UNIT = (X(4)-X(2))/FLOAT (8)
```

```
X(2) = X(2)-UNIT
```

```
X(4) = X( )+UNIT
```

5. SDNPUT: The FORTRAN statement

CALL SDNPUT (X,Y,BUF,L) will take the coordinate pair (X,Y) and do one of the following:

- a. if it is the first point it will be stored in BUF
- b. if it is not the first point and linear interpolation is requested, SDNPUT will do the linear interpolation between the previous point and (x,y), generate the S-C instructions (draw vectors) and store them in BUF.
- c. if it is the second point and quadratic interpolation is requested then (x,y) is stored in BUF along with the first point.
- d. if it is the third point and quadratic interpolation is requested, then a quadratic is fit through the 3 points and extra points are generated to give a smooth looking curve from the first point to the third point. The S-C instructions are formed and stored in BUF.

where:

BUF is a buffer area previously set up by SDINIT.

X is a value of the independent variable of the curve the user wishes to plot.

Y is the value of the dependent variable of the curve corresponding to X.

L is 0 if linear interpolation is desired and 1 if the user wishes quadratic interpolation between successive points.

If the buffer area should become full while using SDNFUT, it is automatically dumped on the tape unit SYSPL1.

If the point (x,y) is off the plotting surface, interpolation is carried only to the boundary. Points outside the plotting surface are saved and will be used for interpolation should any subsequent points be within the limits of the plotting surface.

6. P~~P~~LOT: The FORTRAN statement

CALL P~~P~~LOT (X,Y,BUF,1H*) will plot at the point (x,y) the single Hollerith character *. The character * can be replaced by any acceptable Hollerith character.

where:

X is the x-coordinate of a point (x,y) in the users coordinate system associated with BUF.

Y is the y-coordinate.

BUF is the name of a buffer area previously initialized by SDINIT.

* is any Hollerith character.

7. **SINTRP: The FORTRAN statement**

CALL SINTRP (BUF) will reset the buffer area such that the next time **BUF** is used in the parameter list of **SDRPUT** the corresponding point will be considered as a first point.

where:

BUF is a buffer area.

SINTRP can be used for plotting discontinuous functions.

8. **STERM**: The FORTRAN statement

CALL STERM (NT,BUF,...) will write any S-C 4020 instructions in **BUF** onto tape **SYSPL1**. In addition it resets certain control parameters in **BUF** and leaves the buffer as if **SDINIT** were used again.

where:

NT is not used in the F-IV version

BUF is a buffer area set up by **SDINIT** and used by **SDNPUT**.

STERM is usually used after the last point of a curve has been put into **BUF** by using **SDNPUT**.

The parameter list is variable and the parameter **BUF** may be repeated any number of times for other buffer areas.

*THIS SUBROUTINE IS NO LONGER
REQUIRED IF THE UNIT IN
THE WRITE STATEMENT REFERS
TO UNIT 18*

9. SETPL-RESPL: The FORTRAN statement

CALL SETPL(NY) will set a counter (COUNT) in the output routine equal to NY and cause all subsequent WRITE(UNIT,FMT)'List' to output on UNIT in a SC 4020 format.

where:

NY is the y-coordinate of a point on the S-C plotting surface.

CALL RESPL will nullify the effect of SETPL and all subsequent WRITE statements will take their normal action.

10. **OUTPUT:** The FORTRAN statement

WRITE (UNIT,FMT) 'List' will output the list of parameters under control of the format statement **FMT**.

where:

UNIT is a FORTRAN-IV logical tape unit.

FMT is a fixed point integer corresponding to a **FORMAT** statement.

'List' denotes all the parameters the user wishes to output.

The printed characters occupy a width of 8 units on the S-C 4020 plotting surface and the shortest distance between two succeeding lines of print (without using **SWT**) is 16 units.

All lines of printing start at (8,x) on the S-C 4020 plotting surface, where x is fixed either by **OUTPUT** or **SET**. The first character of the line will have its center at (8,x) and will proceed horizontally from there.

The first character of the line governs vertical placement on the page as follows:

- a. 1. will advance the film, set a line counter (**COUNT**) in the subroutine to zero and start printing at (8,0) in the S-C coordinate system.
- b. N,N = 2,3,4,5,6,7, will add to the line counter (**COUNT**) the amount $1024/N$ and start the line of print at the point (8,COUNT) on the S-C coordinate system. If **COUNT** exceeds 1023, the film will be advanced and printing will be resumed at (8,0) of the new frame.

- c. 0 will add 32 to the line counter and start the line of print at the point (8,COUNT).
- d. "blank" will add 16 to the line counter and start the line of print at the point (8,COUNT).

The first character will be replaced by a blank when outputted.

Only 120 characters are permitted on a line.

Before any WRITE (UNIT,FMT) 'List' statements can be used for SC-4020 output, the subroutine SREPL must be used.

11. SPRINT: The FORTRAN statement

CALL SPRINT (NF,NXL,NYL,⁺NS,NHXL..Xn,...) will put on tape NF the S-C instructions to plot the N Hollerith characters X1..Xn starting at the point (NX,NY) and being NS units high. The line of characters will be plotted either horizontally or vertically depending on the sign of NS.

where:

NF is not used in FORTRAN IV.

NXL is a fixed (or floating) point number indicating the X component of a point on the S-C plotting surface.

NYL is a fixed (or floating) point number indicating the Y component of a point on the S-C plotting surface.

NS is a fixed (or floating) point number indicating the number of units high to print the characters. +NS indicates printing is to be done horizontally and -NS indicates vertical.

NHXL..Xn is the standard way of carrying Hollerith text in a parameter list.

The parameter list is variable and the quadruple (NXL,NYL,⁺NS, NHXL..Xn) may be repeated using different names.

The point (NXL,NYL) will be at the lower left edge of the first character.

The ratio of the height to the width of the characters is 7 to 5. The distance between the characters (center to center) in the horizontal direction is the height of the characters and in the vertical direction an extra 1/3 of the height is used.

12. **SPRINTA:** The FORTRAN statement

CALL SPRINTA (NF,NXL,NYL,⁺NS,A,N,...) will do the same thing as **SPRINT** except the Hollerith text is taken from the array **A** starting at **A(1)** and proceeding to **A(N)**.

where:

NF,NXL,NYL,⁺NS are the same as for **PRINT**.

A is a 1-dimensional array containing Hollerith characters.

N indicates the number of words in **A** to be printed.

A user desiring to set up the array **A** using conversion under format control should see Appendix I.

13. SLABEL: The FORTRAN statement

CALL SLABEL will put on tape SYSPL1 (18) the S-C instructions to do in order the following:

- a. Advance the film.
- b. Plot the \$JOB card of the current program
- c. Advance the film.

SLABEL should be used before any other plotting or writing is done.

14. CAMERA: The FORTRAN statement

CALL CAMERA (NT,NO) will write on tape SYSPL1 the S-C instruction to

- a. select camera #1 if NO = 1
- b. select camera #2 if NO = 2
- c. select cameras #1 and #2 if NO = 3

where:

NT is not used and may be any dummy number.

NO is either 1, 2 or 3.

Camera #1 will be a 35mm camera and requires photographic processing before the plots are available.

Camera #2 will be used in conjunction with an F-80 which is a fast copy device.

CAMERA must be used before any plotting or writing is done (including SLABEL).

15. ADV: The FORTRAN statement

CALL ADV (NT) will output on tape SYSPL1 the S-C 4020 instruction
to advance the film one frame in the camera(s) previously selected.
where:

NT is not used and may be any dummy number.

16. **F_{FORM}**: The FORTRAN statement

CALL F_{FORM} (NF) will write on tape SYSPL1 the S-C instruction to superimpose the information on the form slide (see S-C 4020 Manual) to the film of the previously selected camera(s).

where:

NF is not used and may be any dummy number.

~~17. EXPOSE: The PLOT/N statement~~ *NOT AVAILABLE*

CALL EXPOSE (NT,NO) will write on tape SYSPL1 the S-C instruction to plot all succeeding information with light or heavy intensity depending on whether NO=1 or 2 respectively.

where:

NT is not used and may be any dummy number.

NO is 1 or 2.

GRID uses light intensity and cannot be varied. The subroutine SPRINT,SPRINTA,SDNPUT use heavy intensity and cannot be varied. The reason being that the S-C 4020 instructions "draw grid" and "draw vector" have their intensities fixed. EXPOSE will effect PLOT and WRITE.

APPENDIX I

BUFFER INITIALIZATION

Once a buffer BUF has been initialized by SDINIT, the first 20 locations are used for control information. The locations and uses are:

BUF (1) = X_o	}	users coordinate system and set by SDINIT
BUF (2) = Y_o		
BUF (3) = X_m		
BUF (4) = Y_m		
BUF (5) = S_o	}	S-C 4020 coordinates and set by SDINIT
BUF (6) = t_o		
BUF (7) = S_m		
BUF (8) = t_m		
BUF (9) = S_{n-1}	}	last S-C 4020 point used by SDNPUT
BUF (10) = t_{n-1}		
BUF (11) = A	}	defines transformation of x variable to S-C 4020 units, i.e., $S=A \cdot x+B$
BUF (12) = B		
BUF (13) = A_1	}	defines transformation of y variable to S-C 4020 units, i.e., $T=A_1 \cdot y+B$
BUF (14) = B_1		
BUF (15) =		number of S-C 4020 instructions input into BUF. The first instruction is at BUF (21). BUF (15) is changed by SDINIT, STORE, OUT, STERM.
BUF (16) =		the size of the array BUF and set by SDINIT. This is checked by STORE
BUF (17) =		not used
BUF (18) = S_{n-2}	}	next to last S-C 4020 point used by SDNPUT
BUF (19) = t_{n-2}		
BUF (20) = NP		is a flag used SDNPUT.

APPENDIX II

When drawing a curve on the S-C 4020, interpolation is needed between the points supplied by the user. The S-C 4020 can not draw a line between two points if they are greater than 64 units apart. The user is provided two options for generating points in between.

1. Linear Interpolation:

Let (X_n, Y_n) be the last available point and (X_{n-1}, Y_{n-1}) , (X_{n-2}, Y_{n-2}) be the two points previous to it. Then interpolated points (X_p, Y_p) are generated from the formulas;

$$Q = \max \left[\left| \frac{X_n - X_{n-1}}{60} \right|, \left| \frac{Y_n - Y_{n-1}}{60} \right| \right] + 1$$

$$X_p = X_{n-1} + \frac{P}{Q} (X_n - X_{n-1})$$

$$Y_p = Y_{n-1} + \frac{P}{Q} (Y_n - Y_{n-1}) \text{ for } P = 1, \dots, Q$$

which guarantees that no two successive points will be greater than 64 units apart.

2. Quadratic Interpolation:

Let the sequence of points be given as above. Then interpolated points $(X(s), Y(s))$ are generated from the formulas;

$$S_1 = \left[(X_{n-2} - X_{n-1})^2 + (Y_{n-2} - Y_{n-1})^2 \right]^{\frac{1}{2}}$$

$$S_2 = \left[(X_{n-1} - X_n)^2 + (Y_{n-1} - Y_n)^2 \right]^{\frac{1}{2}}$$

$$DS = \frac{S_1 + S_2}{15}$$

$$X(S) = X_{n-1} + A_x S + b_x S^2$$

$$Y(S) = Y_{n-1} + A_y S + b_y S^2$$

where S runs from $-S_1$ to S_2 in steps of DS and A_x, b_x, A_y, b_y are determined from the linear equations formed by letting

$$X(-S_1) = X_{n-2} \quad Y(-S_1) = Y_{n-2}$$

$$X(S_2) = X_n \quad Y(S_2) = Y_n$$

The above equations guarantee that no two successive points will be greater than 15 units apart.

The parameter S is roughly equivalent to the arc length.

The formulas have the advantages of:

- a. uniform vector lengths along the arc length
- b. capable of handling the order in which points are supplied i.e. handle closed curves.

They have the disadvantage of giving bad results for very sparse data.

APPENDIX III

Several low level utility routines were developed which are of general use to the SPLØT subroutines

A. LØGIC: This is a set of FUNCTION type subprograms used for packing and unpacking SC-4020 instructions. The individual programs are:

- a. ARS (N,A)-will shift A right N places
- b. ALS (N,A)-will shift A left N places
- c. ØRA (A,B)-will OR together A and B
- c. ANA (A,B)-will AND together A and B

B. STØRE: The statement

CALL STØRE (ØP,BUF) where

ØP is a SC-4020 instruction

BUF is a buffer region initialized by SDINIT,

will store ØP in BUF. If ØP should fill the buffer region, all the operations currently in BUF will be output on SYSPLI (unit 18). The subroutine ØUT is used for dumping BUF.

C. ØUT: The Statement

CALL ØUT (BUF) where

BUF is a buffer region

will output any data stored in BUF and reset all control words in BUF.

D. SETPR-RESPR: The statement

CALL SETPR (A,N) where

A is an array for storage of an output line image,

N is an integer for storage of the number of words

in the line image

will cause any subsequent

PRINT fmt, 'List'

statements to put the formatted line image into the array

A and the number of words of the line image into N. No

physical unit action will take place.

CALL RESPR

will nullify the action of CALL SETPR (A,N).

E. SETRD-RESRD: The statement

CALL SETRD (A) where

A is an array containing an input card image

will cause any subsequent

READ fmt, 'List'

statements to obtain the input card image from the array A.

CALL RESRD

will nullify the action of SETRD (A).

PROGRAM SIZES (IASYS V/2)
FORTRAN WRITTEN

Program	Locations ₁₀	Subroutines Used	Comments
1. XKNDT	146		This routine has the same calling sequence as SDINIT but is not variable length.
2. SINTRP	24		
3. SCALE	371		
4. STORE	57	OUT	
5. PLAB	455	ORA, STORE, RESRD, ALS, SETRD, SETPR, RESPR	
6. ADV	36		
7. CAMERA	40		
8. FPRM	36		
9. PPLDT	134	ORA, ANA, ALS, ARS, STORE	
10. OUT	87		
11. OUTPUT			This is the standard FORTRAN IV I/O package.
12. SLABEL	59	GETJOB	
13. EXPOSE	40		
14. BGRID	368		

MAP WRITTEN

1. GRID	124		
2. STERM	28	OUT	
3. ALS (LOGIC) ANA ARS ORA	20		
4. SDINIT	43	XKNDT	
5. SPRNTA SPRINT	340	SINTRP, SDNPUT, STERM	
6. SDNPUT	787	STORE	
7. SETPL RESPL	16	FWRDP	
8. FWRDP	89		An alternate FWRD subr.
9. SETPR RESPR	22	FPRNA	
10. FPRNA	22		An alternate FPRN subr.
11. SETRD RESRD	19	FRDA	
12. FRDA	19		An alternate FRCD subr.