FACILITY FORM 602

N69-14869
(ACCESSION NUMBER)

(THRU)
1

545
(PAGES)

(CODE)
23

NASA-CR #94313
(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 Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration under Contract NAS7-100

RE-ORDER NO. 68-499

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

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

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 proceding 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.

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.

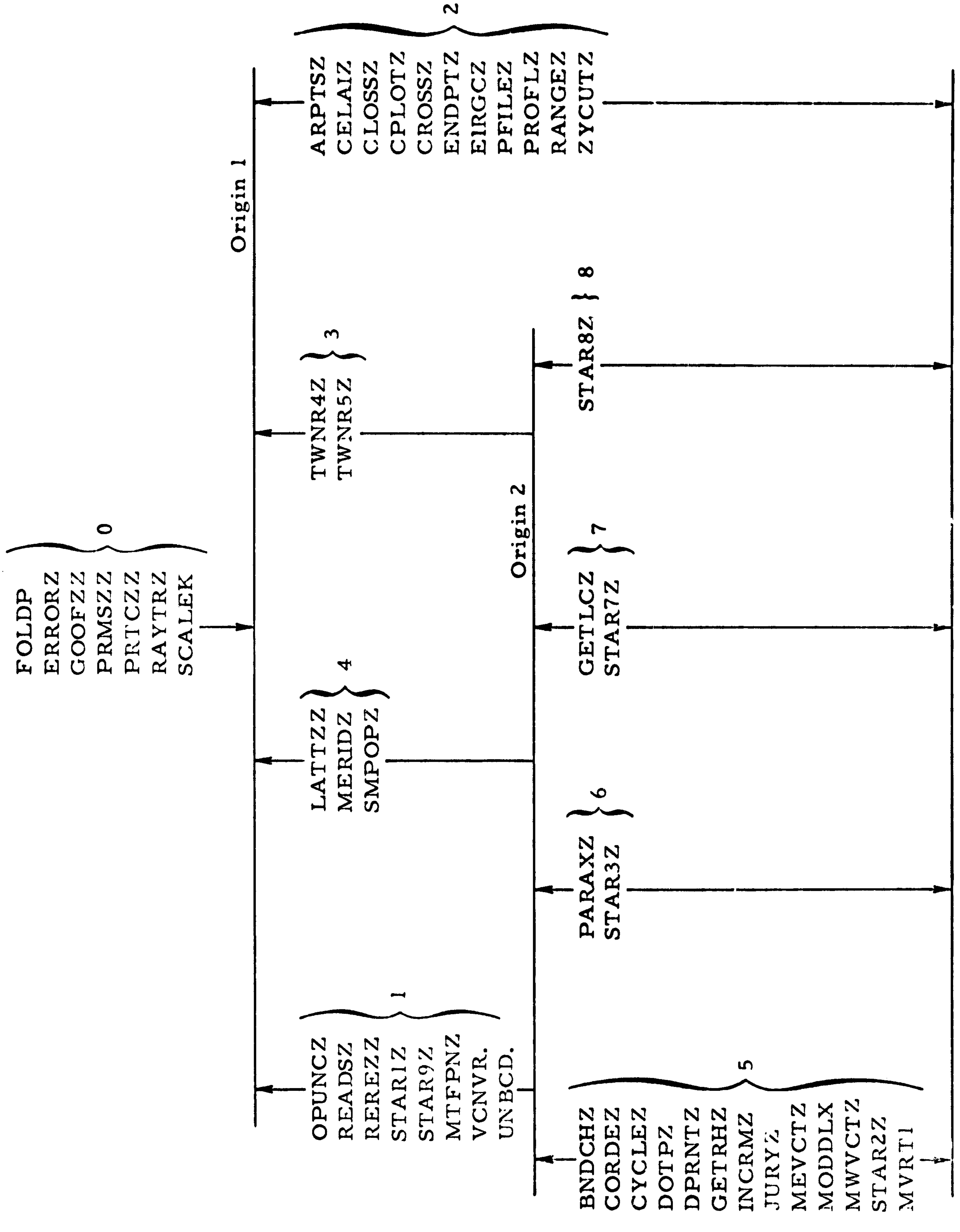


Figure 1. FOLDP Overlay Structure

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.

COMMON BLOCK	DECKS
DATA	FOLDP, PRMSZZ, PRTCZZ, RAYTRZ, OPUNCZ, READSZ, 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, MTFPNZ
PERTB	FOLDP, READSZ, STARIZ, STAR8Z
PLOTC	FOLDP, READSZ, STAR1Z, PROFLZ, CPLOTZ, PFILEZ
SYMBLS	FOLDP, READSZ, STAR9Z, STAR1Z
AZOBJ	RAYTRZ, PFILEZ, TWNR4Z, TWNR5Z, SMPOPZ MERIDZ, GETRHZ, MEVCTZ, BNDCHZ, PARAXZ, STAR3Z, STAR7Z, STAR8Z
PRNT	PRTCZZ, READSZ, STAR9Z, STAR1Z, 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/

Symbol	Type	Parameter
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

Symbol	Type	DATA Index	NAMELIST Ref.
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

Symbol	Type	DATA Index	NAMELIST Ref.
ITNPRT	I	13	D
DELY	R	14	G
FNUMB	R	15	D
FLNGH	R	16	D
WFLGH	R	17	D
ZETA	R	18	D
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

Symbol	Type	DATA Index	NAMELIST Ref.
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
CIMPL(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
ATRGGR	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/

Symbol	Type	Parameter
PERTB(30, 4)	RM	Perturbation Matrix
NPERTB	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.

COMMON/PLOTC/

Symbol	Type	Parameter
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.

COMMON/SYMBLS/

Symbol	Type	Parameter
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.

COMMON/AZOBJ/

Symbol	Type	Parameter
THTR	R	Θ in radians (Azimuth)
SNTHTR	R	SinΘ
CSTHTR	R	CosΘ

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.

COMMON/PRNT/

Symbol	Type	Parameter
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.

COMMON BLOCK	DECKS
HEADER	FOLDP, PRTCZZ
TMTWIN	TWNR4Z, TWNR5Z
TMPATT	SMPOPZ, MEVCTZ, STAR3Z, STAR7Z, STAR8Z
TMDESN	DOTPZ, DPRNTZ, JURYZ, MWVCTZ, STAR2Z, 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/

Symbol	Type	Parameter
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/

Symbol	Type	Parameter
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	$\tilde{x}_i$ Image Coordinates
YCURL(201)	RV	$\tilde{y}_i$ Image Coordinates
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 $\vec{E}$.

COMMON/TMDESN/

Symbol	Type		Parameter
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 $\overline{E}$
CORD(6)	IV	l_i	$1 \leq i \leq N_c$ Color Order
SIZE	R	ξ_K	Scale Factor
DETERM	R	$\lvert I \rvert$	Determinant of I
E0LNG	R	$\lvert \vec{E}_o \rvert$	Old Length
ESLNG	R	$\lvert \vec{E}^* \rvert$	New Length
IMPRV	R	ϵ	Relative Improvement
VCTSTR(6000)	RV	$\vec{W}, \vec{E}, \partial\vec{E}/\partial u_1, \ldots, \partial\vec{E}/\partial u_n$	
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.

COMMON/ELMAIR/

Symbol	Type		Parameter
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.

COMMON/CODRNG/

Symbol	Type		Parameter
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.

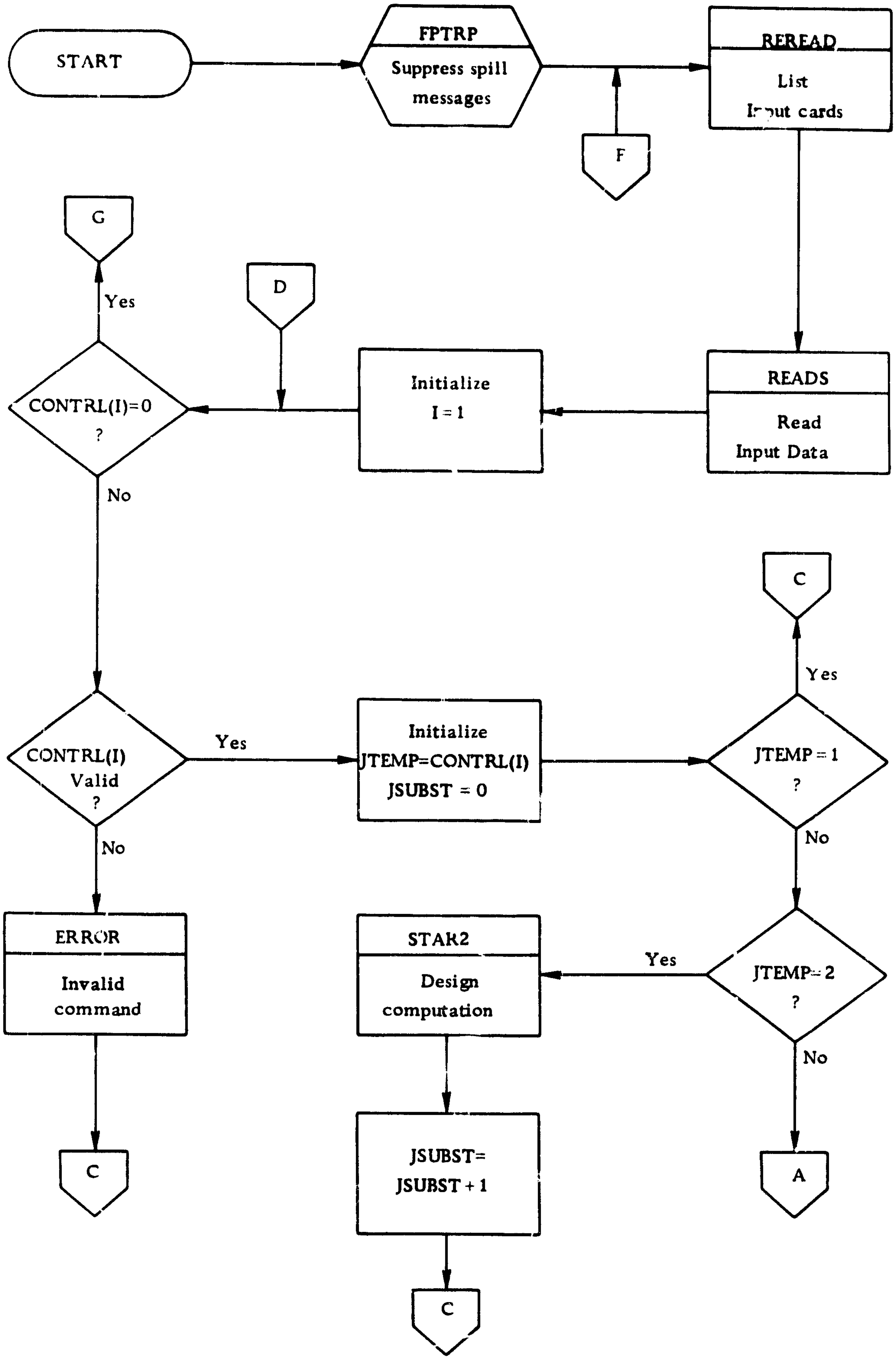
3.0.1 FOLDP

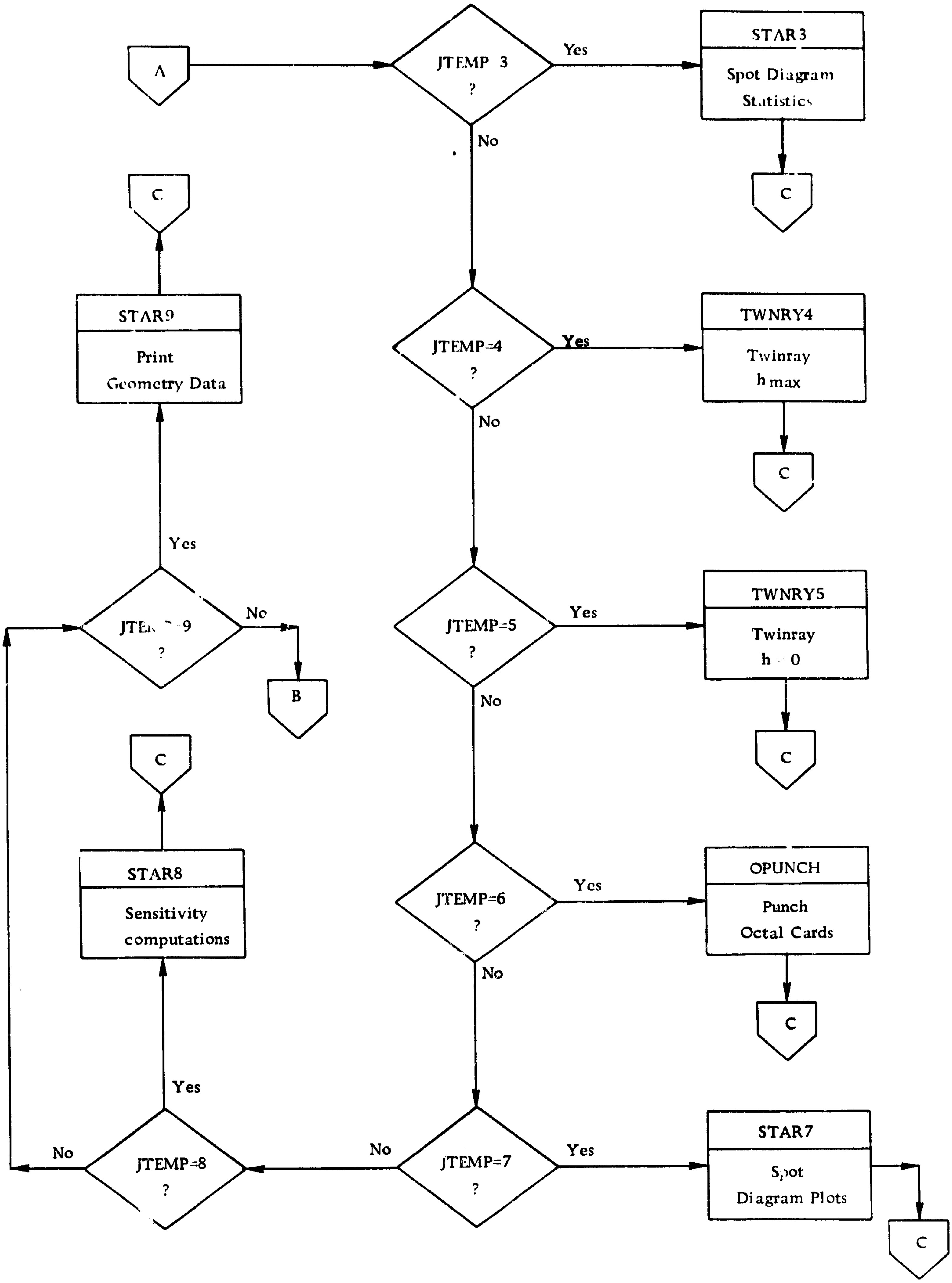
FOLDP is the main program which calls upon the input link and selected subroutines depending upon the options being executed.

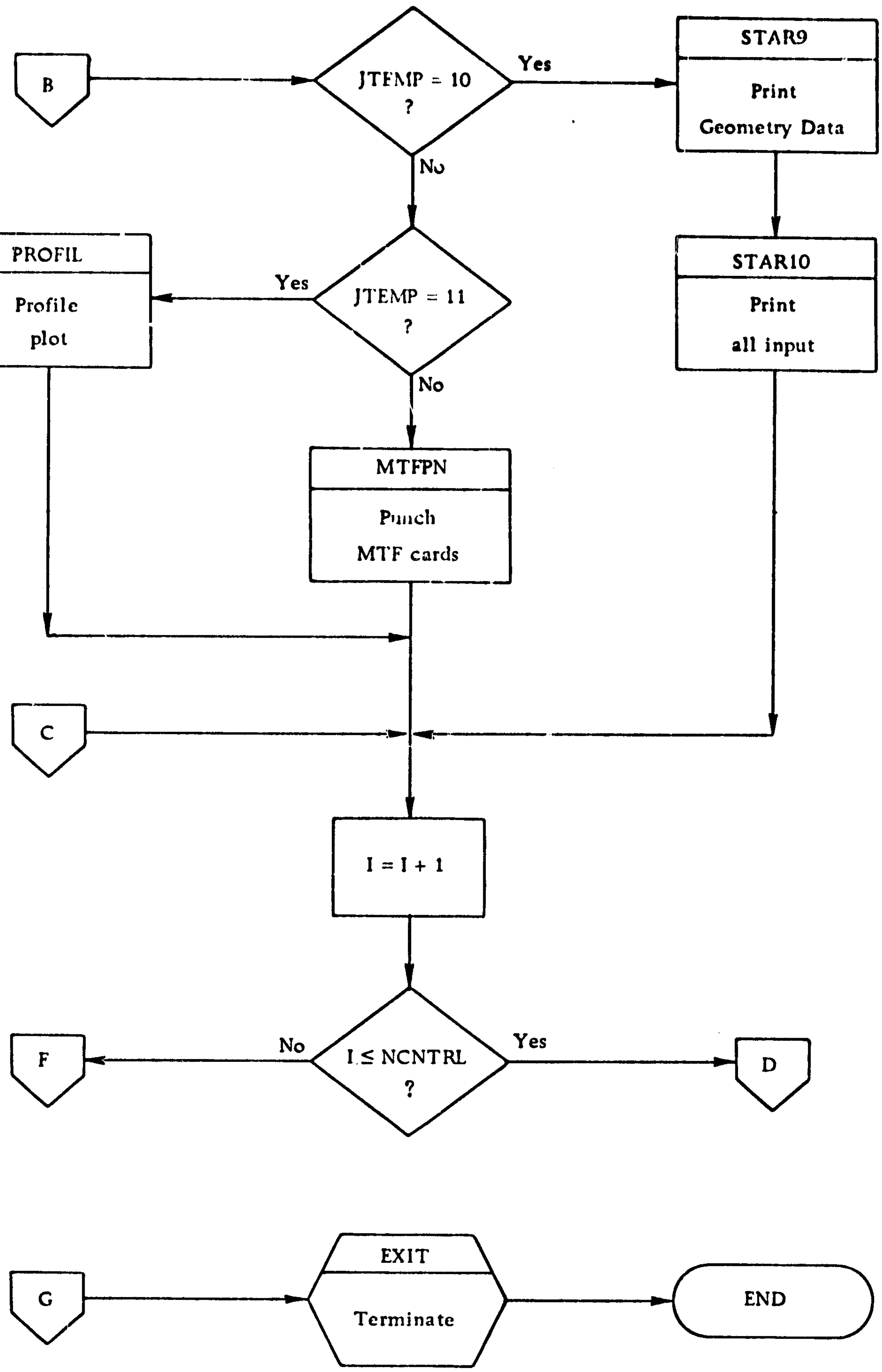
Utility Routines and Common References

COMMON/DATA/	STAR3
COMMON/PERTB/	PRMSUB
COMMON/SYMBLS/	TWNRY4
COMMON/HEADER/	TWNRY5
COMMON/PLOTC/	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.


START
FPTRP
Suppress spill messages
F
REREAD
List
Input cards
G
Yes
D
CONTRL(I)=0
?
Initialize
I = 1
READS
Read
Input Data
No
C
Yes
CONTRL(I)
Valid
?
Yes
Initialize
JTEMP=CONTRL(I)
JSUBST = 0
JTEMP = 1
?
No
No
ERROR
Invalid
command
STAR2
Design
computation
Yes
JTEMP=2
?
No
C
JSUBST=
JSUBST + 1
A
C


A
JTEMP 3
?
Yes
No
STAR3
Spot Diagram
Statistics
C
JTEMP=4
?
TWNRY4
Twinray
h max
JTEMP=5
?
TWNRY5
Twinray
h 0
JTEMP=6
?
OPUNCH
Punch
Octal Cards
JTEMP=7
?
STAR7
Spot
Diagram Plots
JTEMP=8
?
STAR8
Sensitivity
computations
JTEMP=9
?
STAR9
Print
Geometry Data
B


B
JTEMP = 10
?
Yes
STAR9
Print
Geometry Data
No
PROFIL
Profile
plot
Yes
JTEMP = 11
?
STAR10
Print
all input
No
MTFPN
Punch
MTF cards
C
I = I + 1
F
No
I ≤ NCNTRL
?
Yes
D
G
EXIT
Terminate
END

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.

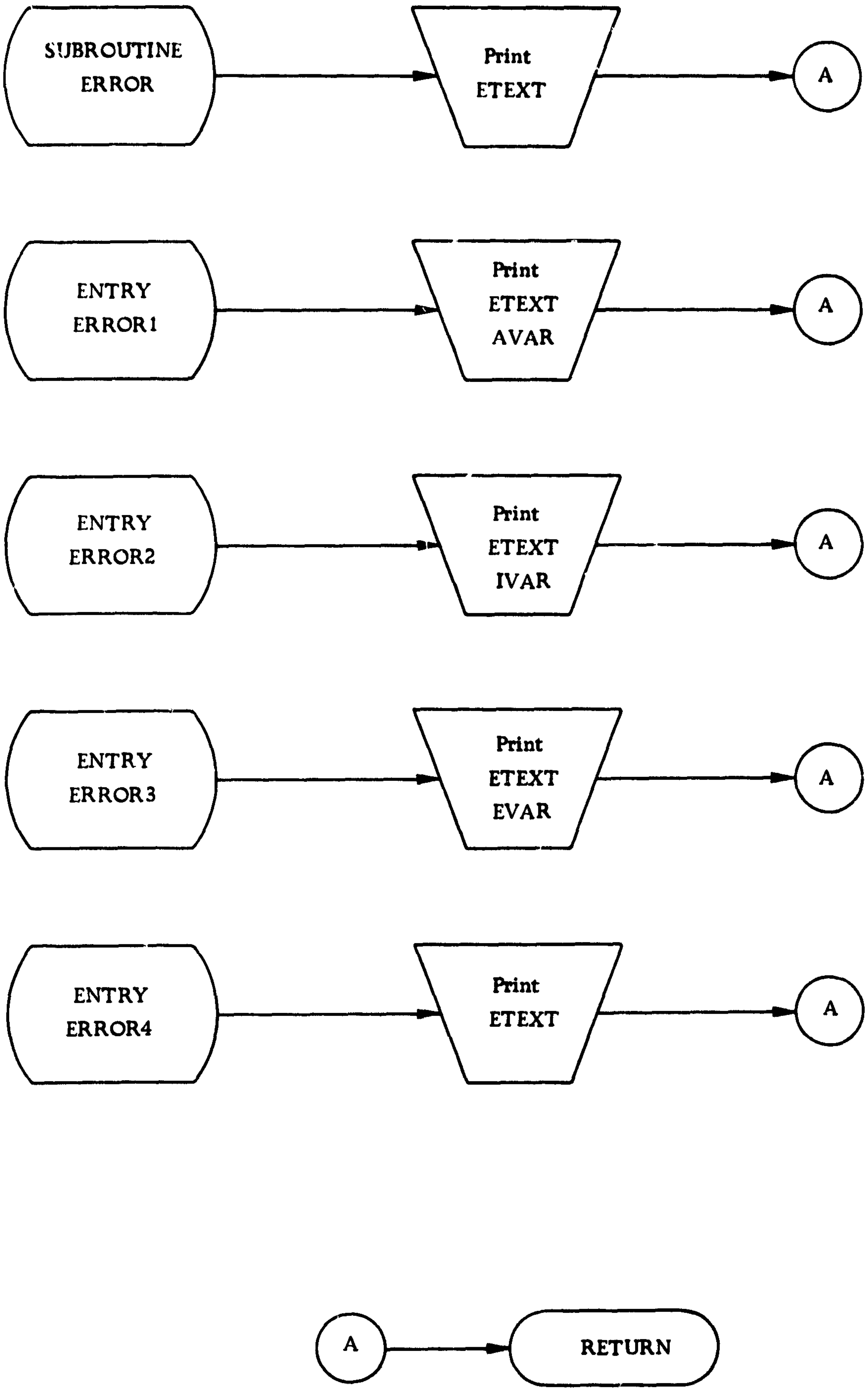
Calling Sequence

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


SUBROUTINE
ERROR
Print
ETEXT
A
ENTRY
ERROR1
Print
ETEXT
AVAR
A
ENTRY
ERROR2
Print
ETEXT
IVAR
A
ENTRY
ERROR3
Print
ETEXT
EVAR
A
ENTRY
ERROR4
Print
ETEXT
A
A
RETURN

3.0.3 GOOFZZ (GOOF)

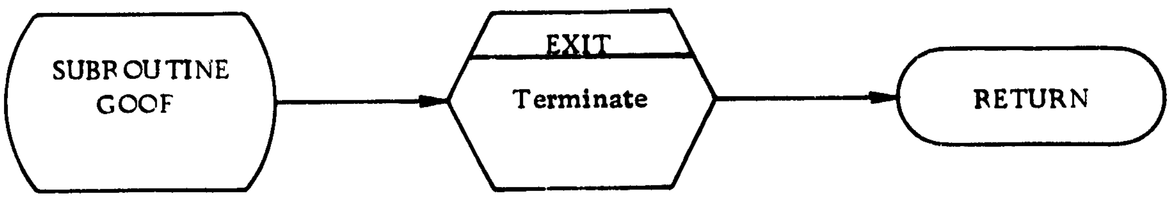
GOOFZZ is a job terminator which does nothing except to call the system subroutine EXIT.

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 jth call in a sequence of $N_s + 1$ calls causes the jth 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

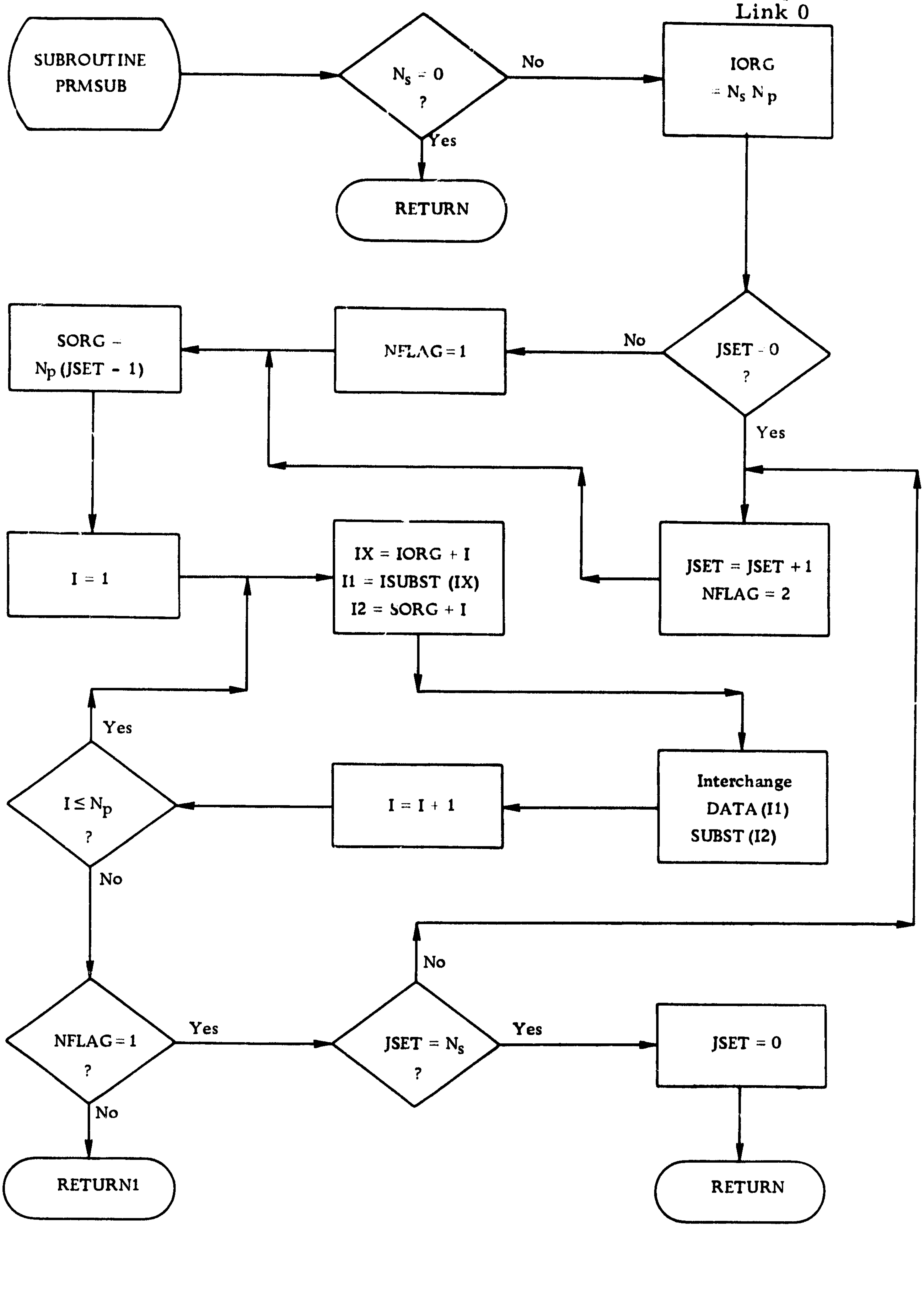
CALL PRMSUB($ALT)

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

Utility Routines and Common References

COMMON/DATA/

Utility - None


SUBROUTINE PRMSUB
$N_s = 0$?
No
Yes
RETURN
IORG = $N_s N_p$
JSET = 0 ?
No
Yes
NFLAG = 1
SORG = N_p (JSET - 1)
I = 1
IX = IORG + I
I1 = ISUBST (IX)
I2 = SORG + I
JSET = JSET + 1
NFLAG = 2
Interchange
DATA (I1)
SUBST (I2)
I = I + 1
$I \le N_p$?
Yes
No
NFLAG = 1 ?
Yes
No
JSET = N_s ?
Yes
No
JSET = 0
RETURN1
RETURN

3.0.5 PRTCZZ (PRTCTL)

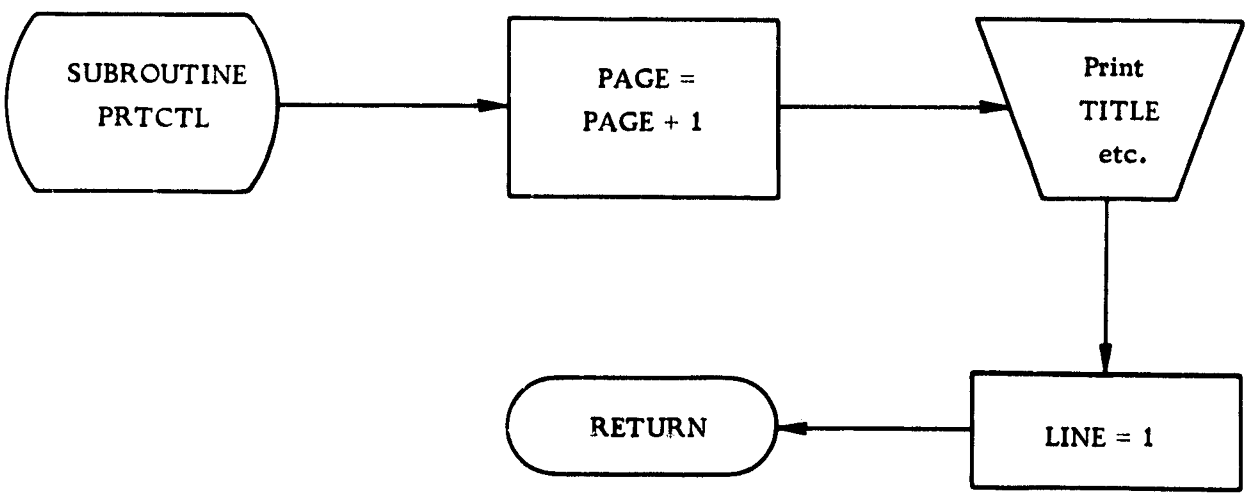
PRTCZZ is a print control routine which causes page ejection and main title generation including title, date, option, substitution set and page number.

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

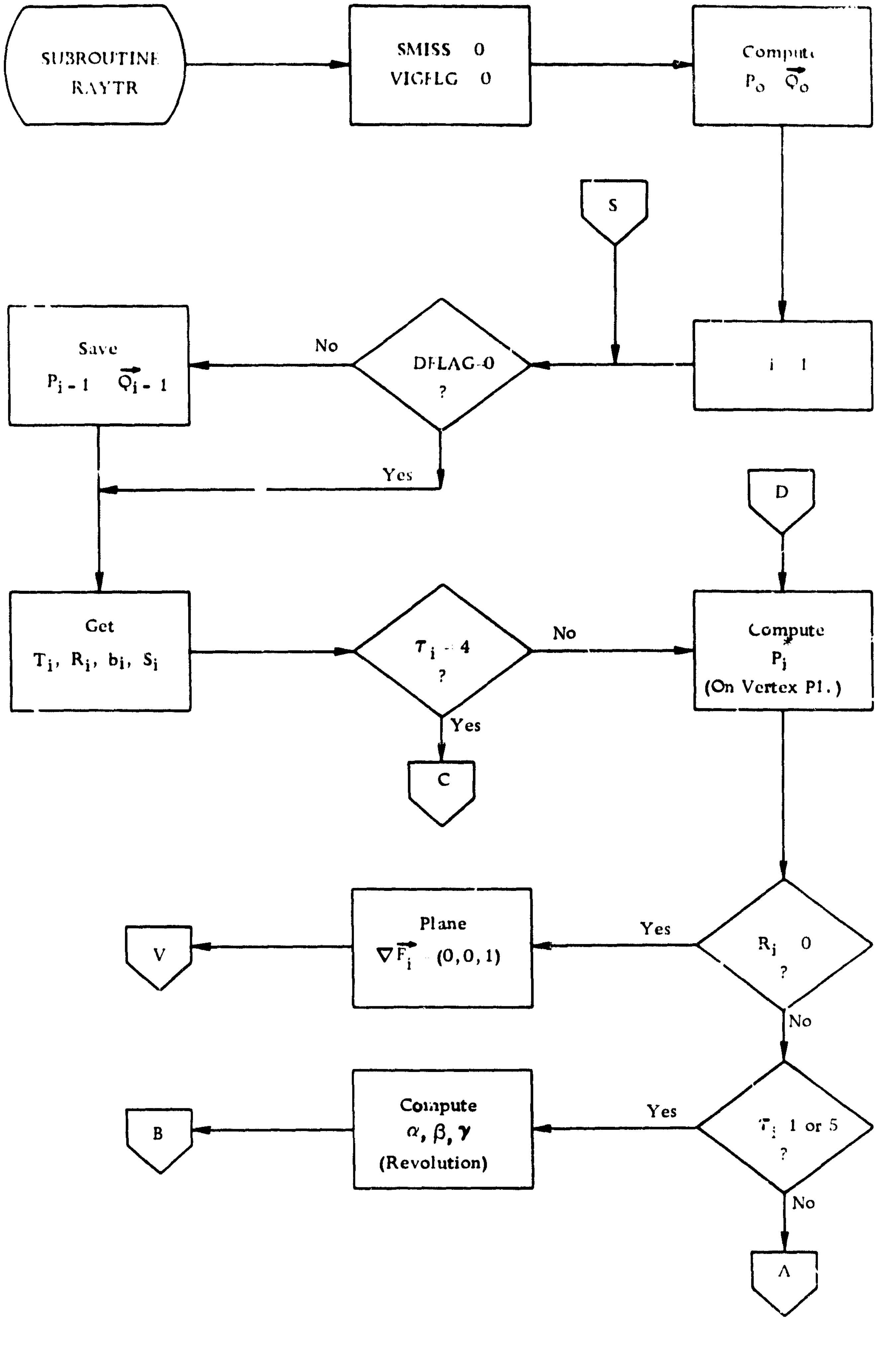
The azimuth of the object (θ) as well as $\sin\theta$ and $\cos\theta$ must be entered into AZOBJ common before calling RAYTR.

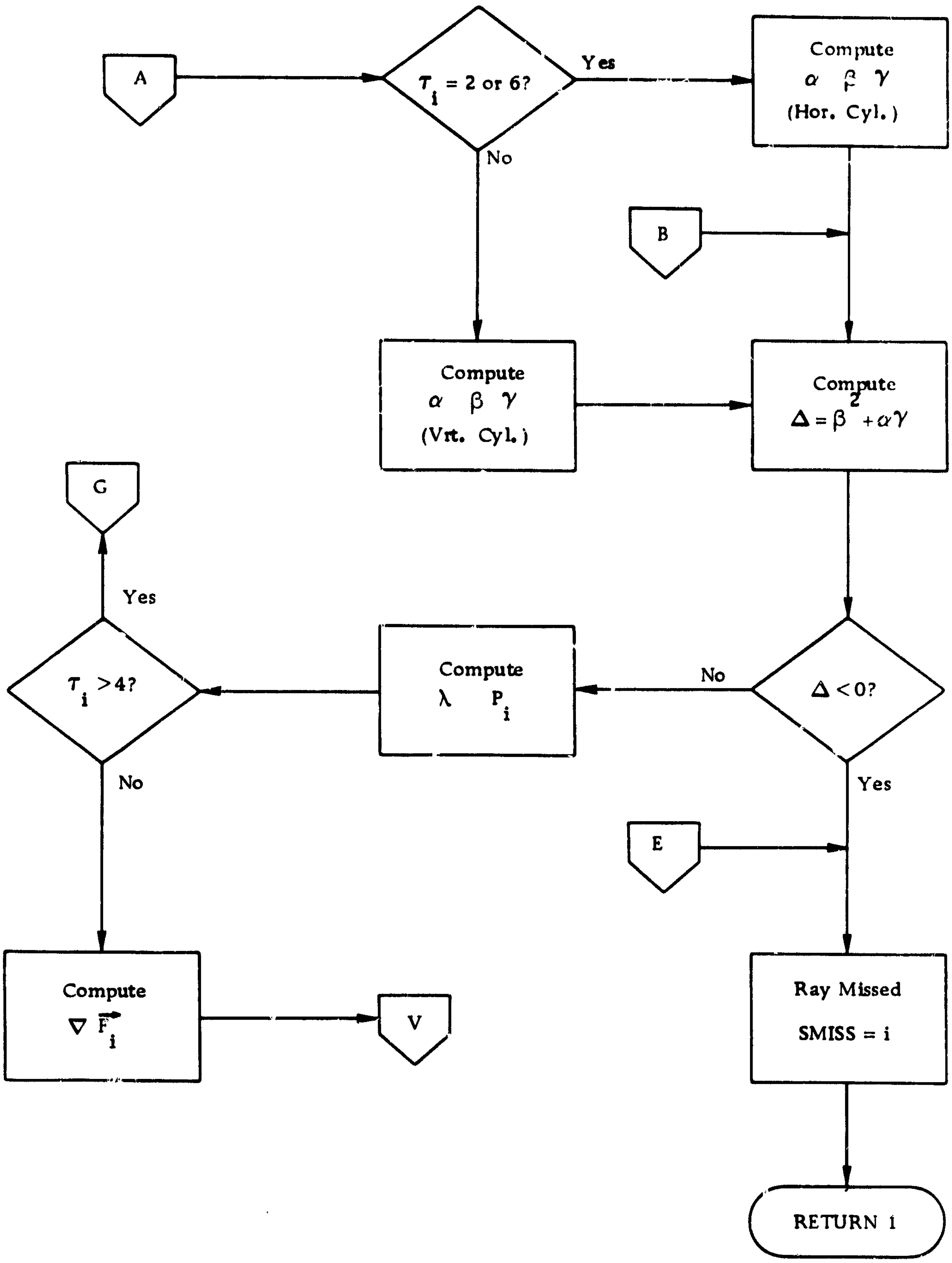
Utility Routines and Common References

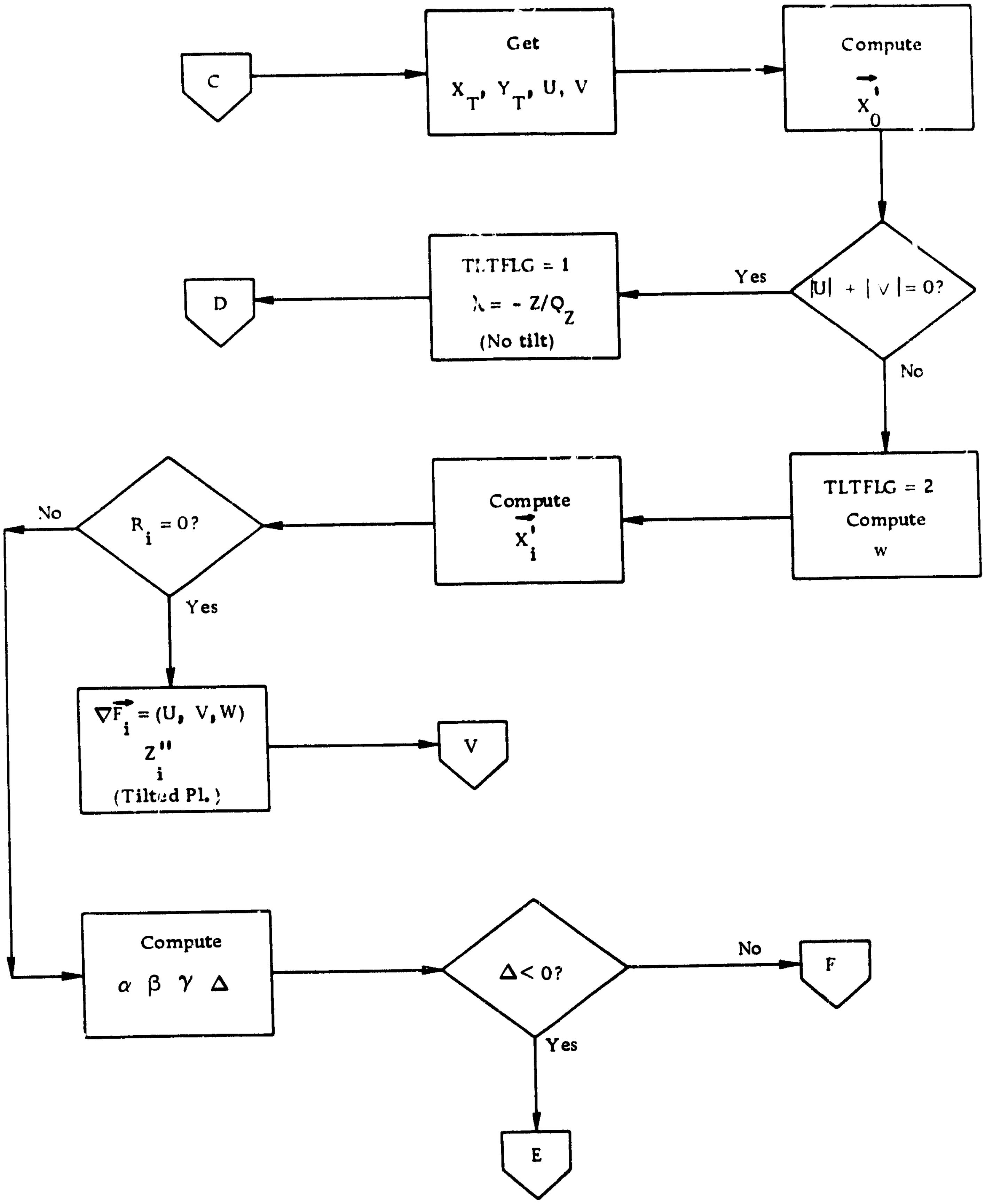
COMMON/DATA/

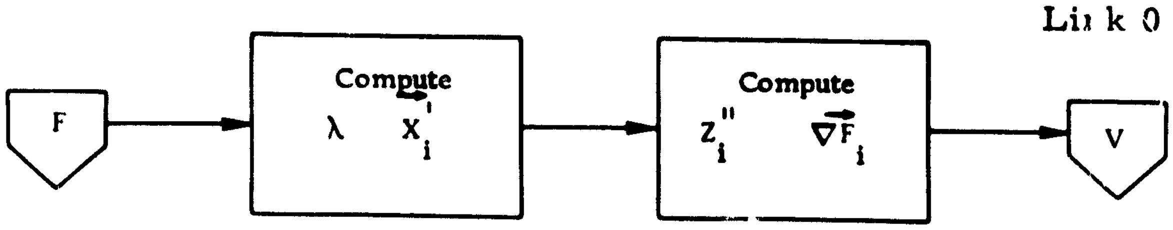
COMMON/AZOBJ/

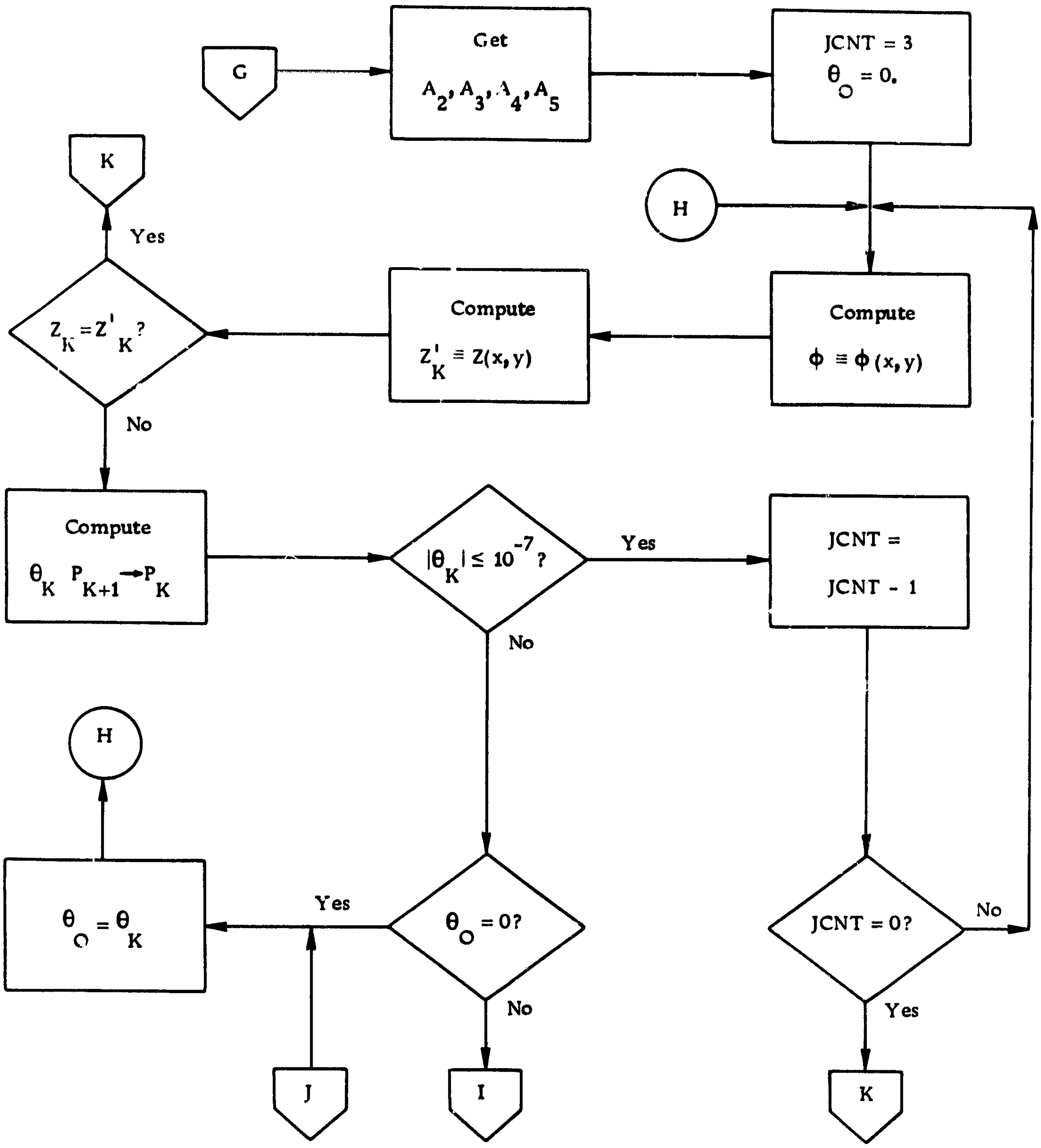
GOOF

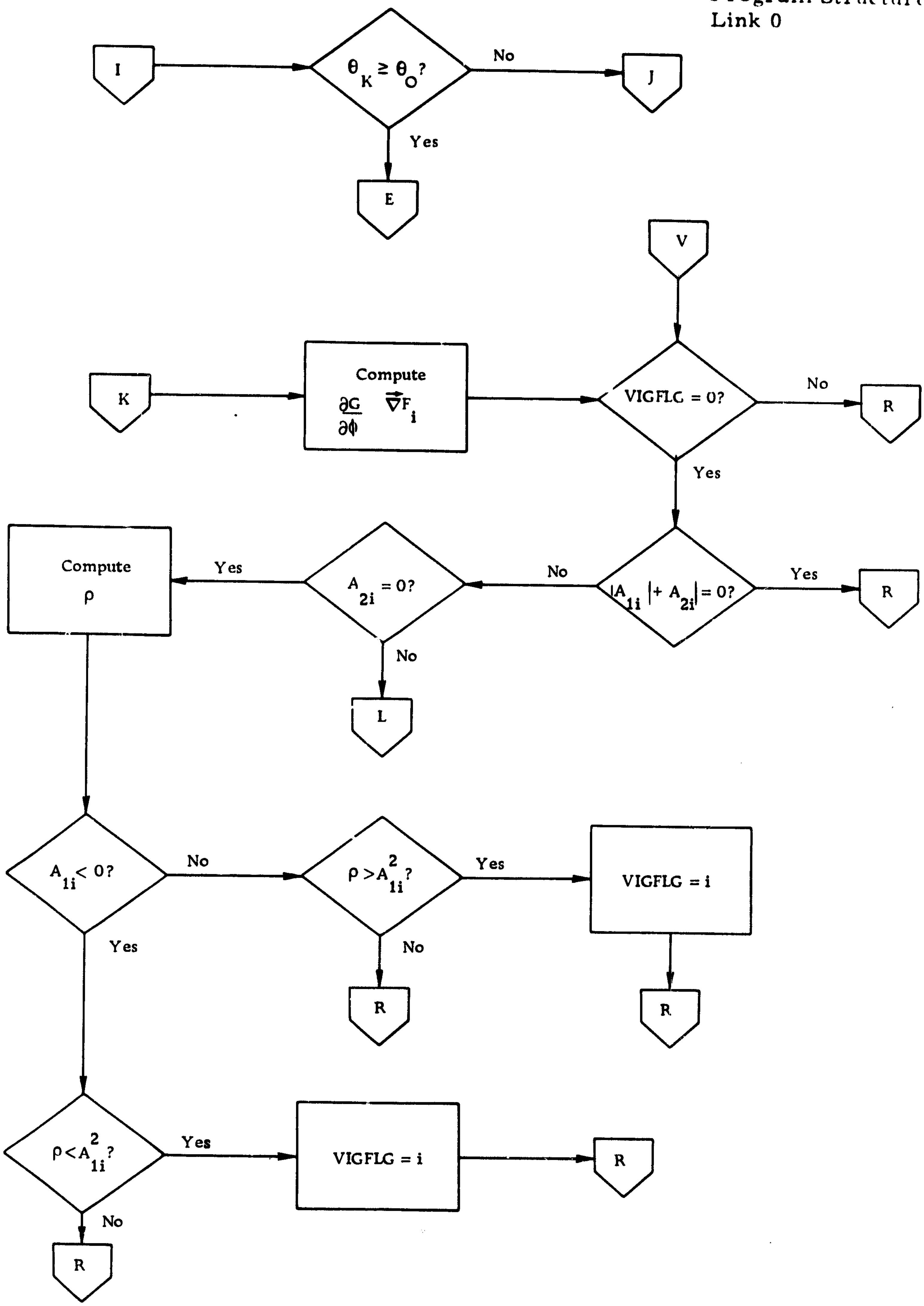

SUBROUTINE RAYTR
SMISS 0
VIGFLG 0
Compute
P_o $\vec{Q_o}$
S
No
DFLAG 0
?
i 1
Save
P_{i-1} $\vec{Q}_{i-1}$
Yes
D
Get
T_i, R_i, b_i, S_i
τ_i 4
?
No
Compute
P_j^*
(On Vertex P1.)
Yes
C
Plane
$\nabla \vec{F_i}$ (0,0,1)
Yes
R_j 0
?
V
No
Compute
α, β, γ
(Revolution)
Yes
τ_i 1 or 5
?
B
No
A

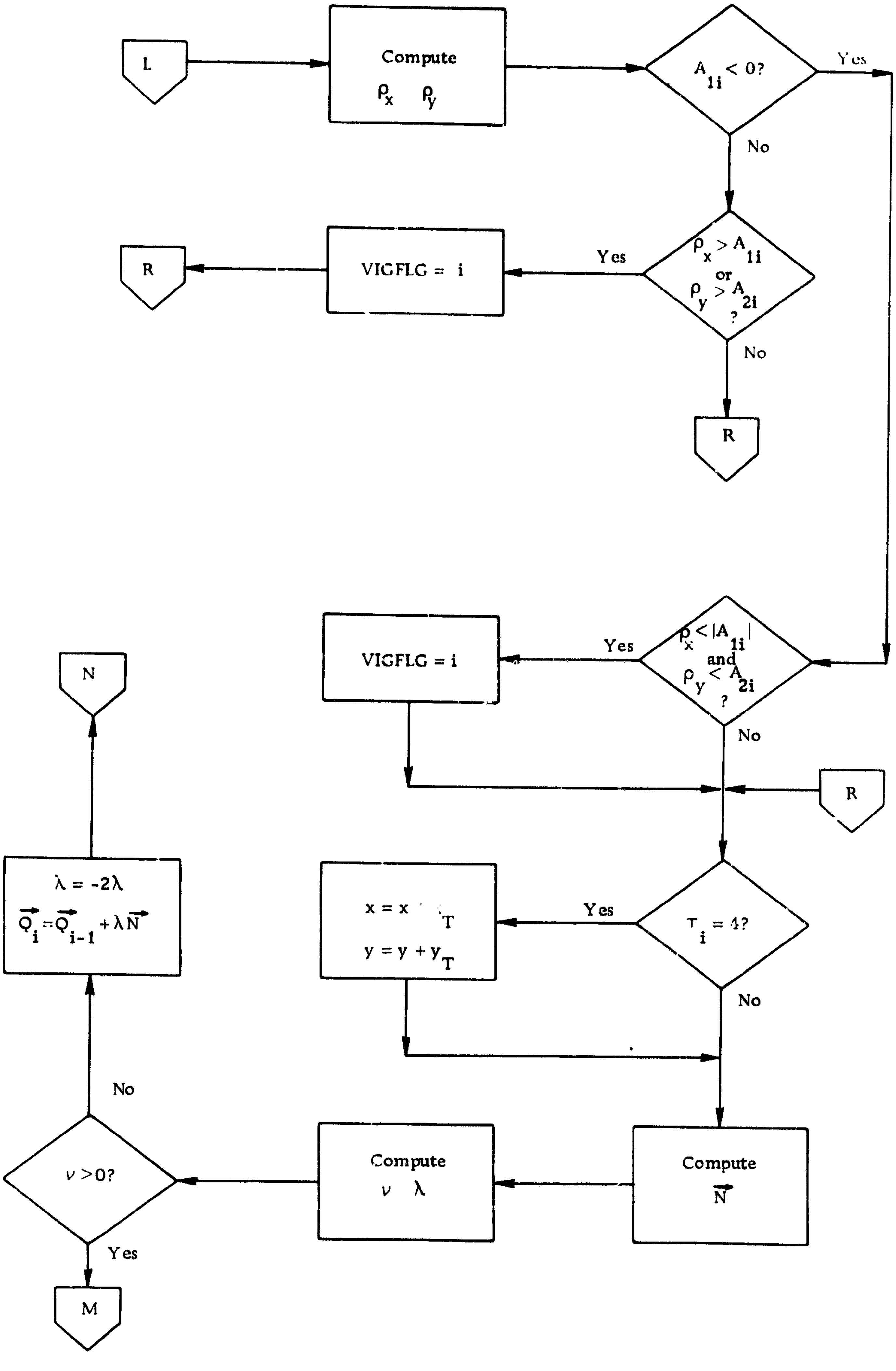

A
τ_i = 2 or 6?
Yes
No
Compute
α β γ
(Hor. Cyl.)
B
Compute
α β γ
(Vrt. Cyl.)
Compute
$\Delta = \beta^2 + \alpha\gamma$
G
Yes
$\tau_i > 4$?
Compute
λ P_i
No
$\Delta < 0$?
No
Yes
E
Compute
$\nabla \vec{F}_i$
V
Ray Missed
SMISS = i
RETURN 1

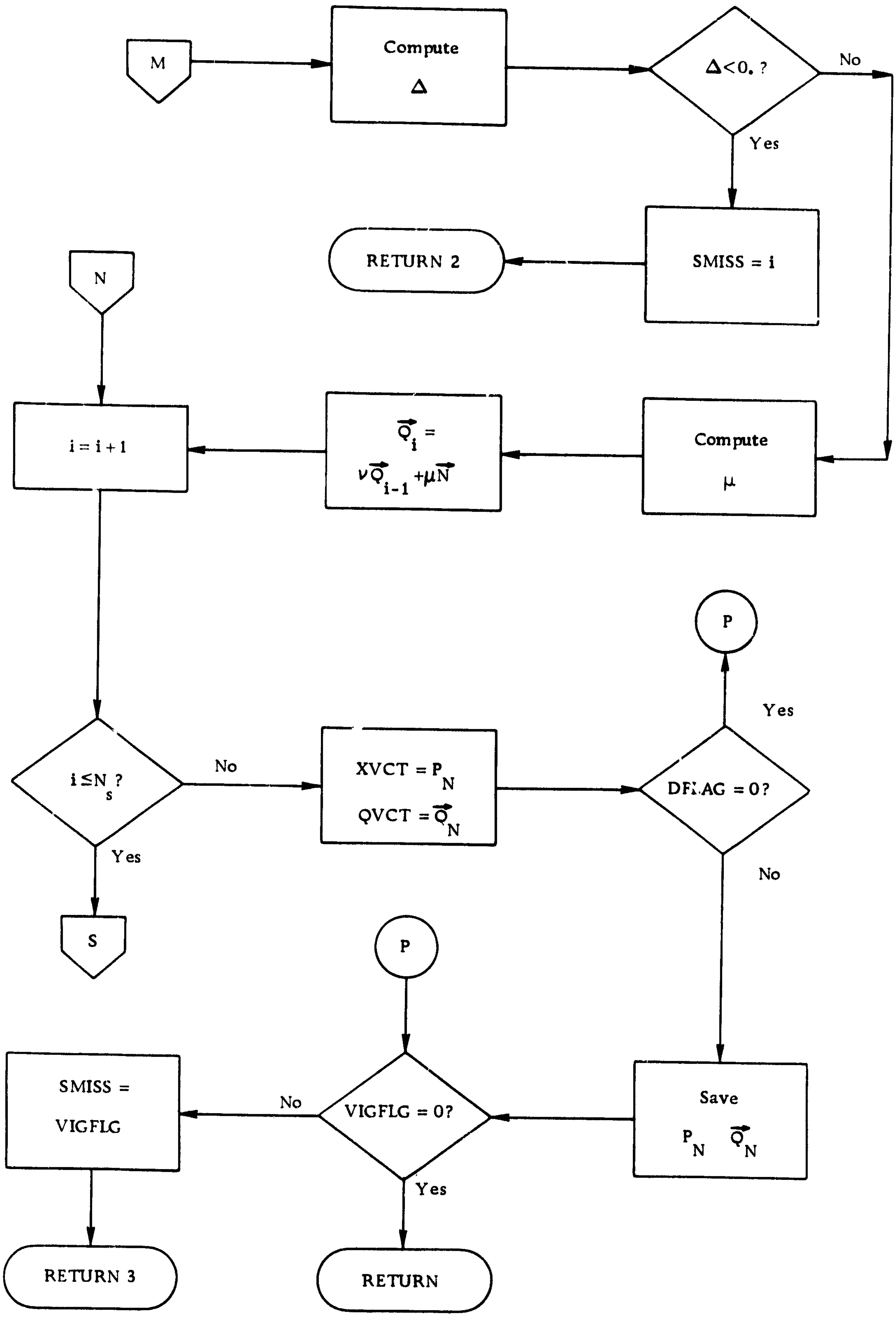

C
Get
X_T, Y_T, U, V
Compute
$\vec{X}'_0$
$|U| + |V| = 0?$
Yes
TLTFLG = 1
$\lambda = -Z/Q_Z$
(No tilt)
D
No
TLTFLG = 2
Compute
w
Compute
$\vec{X}'_i$
$R_i = 0?$
No
Yes
$\nabla\vec{F}_i = (U, V, W)$
Z''_i
(Tilted Pl.)
V
Compute
α β γ Δ
$\Delta < 0?$
No
F
Yes
E


F
Compute
λ $\vec{X}'_i$
Compute
Z''_i $\nabla\vec{F}_i$
V


G
Get
A_2, A_3, A_4, A_5
JCNT = 3
$\theta_O = 0.$
H
Compute
$\phi \equiv \phi(x, y)$
Compute
$Z'_K \equiv Z(x, y)$
$Z_K = Z'_K$?
Yes
K
No
Compute
θ_K $P_{K+1} \rightarrow P_K$
$|\theta_K| \leq 10^{-7}$?
Yes
JCNT =
JCNT - 1
No
JCNT = 0?
No
Yes
K
$\theta_O = 0$?
Yes
No
$\theta_O = \theta_K$
H
J
I


I
$\theta_K \geq \theta_O$?
No
J
Yes
E
V
K
Compute
$\frac{\partial G}{\partial \phi}$ $\vec{\nabla} F_i$
VIGFLG = 0?
No
R
Yes
Compute
ρ
Yes
$A_{2i} = 0$?
No
$|A_{1i}| + |A_{2i}| = 0$?
Yes
R
No
L
$A_{1i} < 0$?
No
$\rho > A_{1i}^2$?
Yes
VIGFLG = i
Yes
No
R
R
$\rho < A_{1i}^2$?
Yes
VIGFLG = i
R
No
R


L
Compute
ρ_x ρ_y
$A_{1i} < 0$?
Yes
No
$\rho_x > A_{1i}$ or $\rho_y > A_{2i}$?
Yes
VIGFLG = i
R
No
R
$\rho_x < |A_{1i}|$ and $\rho_y < A_{2i}$?
Yes
VIGFLG = i
No
R
N
$\tau_i = 4$?
Yes
$x = x + x_T$
$y = y + y_T$
No
$\lambda = -2\lambda$
$\vec{Q}_i = \vec{Q}_{i-1} + \lambda \vec{N}$
No
Compute
$\vec{N}$
Compute
ν λ
$\nu > 0$?
Yes
M


M
Compute
Δ
Δ<0. ?
No
Yes
SMISS = i
RETURN 2
N
i = i + 1
$\vec{Q}_i$ =
$\nu\vec{Q}_{i-1} + \mu\vec{N}$
Compute
μ
P
Yes
i ≤ N_s ?
No
XVCT = P_N
QVCT = $\vec{Q}_N$
DFLAG = 0?
Yes
No
S
P
SMISS =
VIGFLG
No
VIGFLG = 0?
Save
P_N $\vec{Q}_N$
Yes
RETURN 3
RETURN

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. $C \le A \le D$, $C \le B \le D$, $C < D$.
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 ≠ 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 $A \le B$ or $A \ge B$ is permissible. SCALEK will compute numbers C,D,KMAJØR, and KMINØR with $C < D$ such that the interval $C \le x \le D$ 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 Use 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 $531_8 = 345_{10}$

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 6O12 format. Octal card 580 is always punched, and serves as an end of case card for the octal deck.

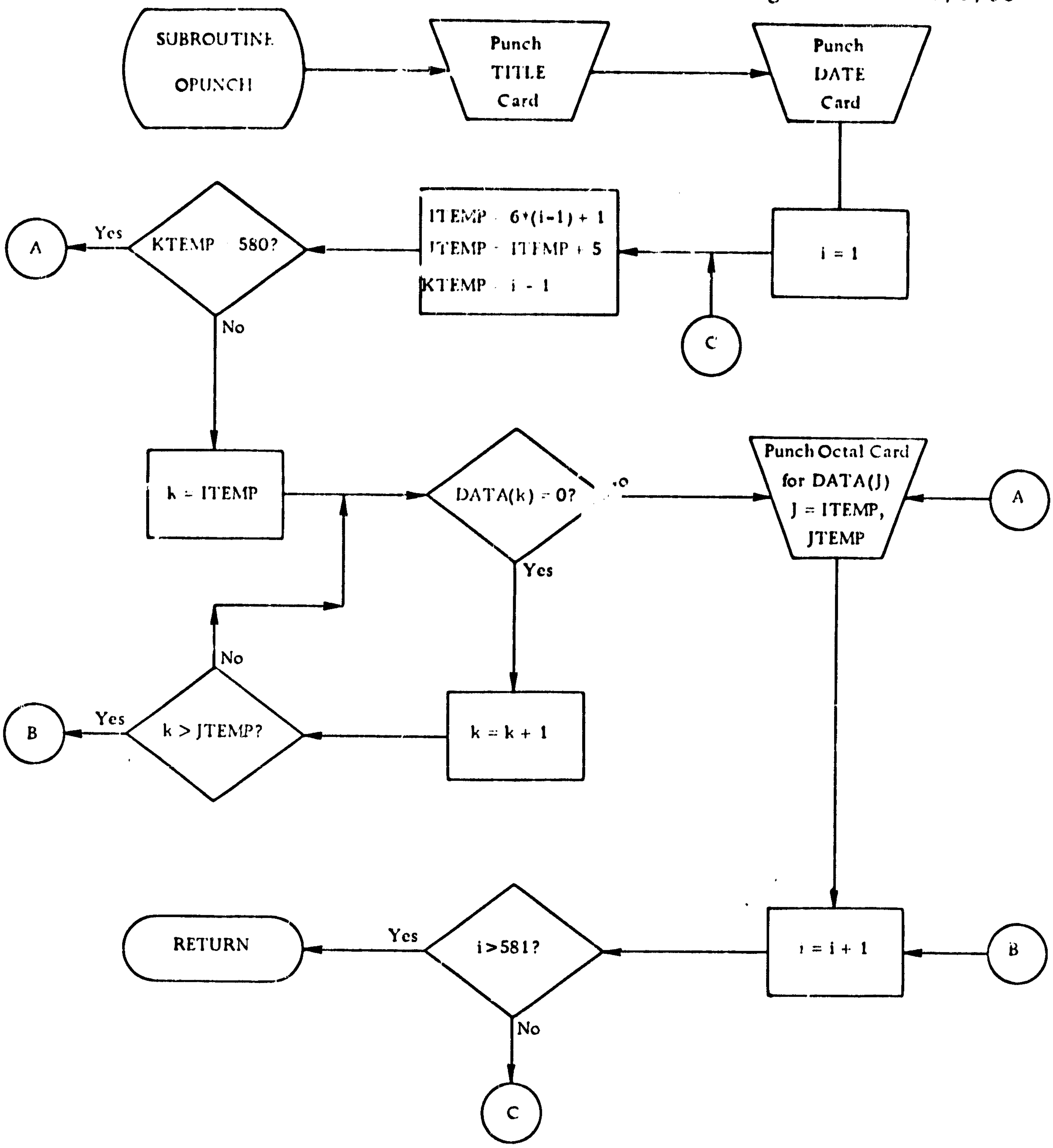
Calling Sequence

CALL OPUNCH

Utility Routines and Common References

COMMON/DATA/

Utility - None


SUBROUTINE OPUNCH
Punch TITLE Card
Punch DATE Card
i = 1
ITEMP 6*(i-1) + 1
JTEMP ITEMP + 5
KTEMP i - 1
KTEMP 580?
Yes
No
A
C
k = ITEMP
DATA(k) = 0?
Yes
Punch Octal Card for DATA(J) J = ITEMP, JTEMP
k = k + 1
k > JTEMP?
No
Yes
B
i = i + 1
i>581?
RETURN

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 numb. : of validity checks and issues error messages.

c) Performs a number of type conversions from real to integer.

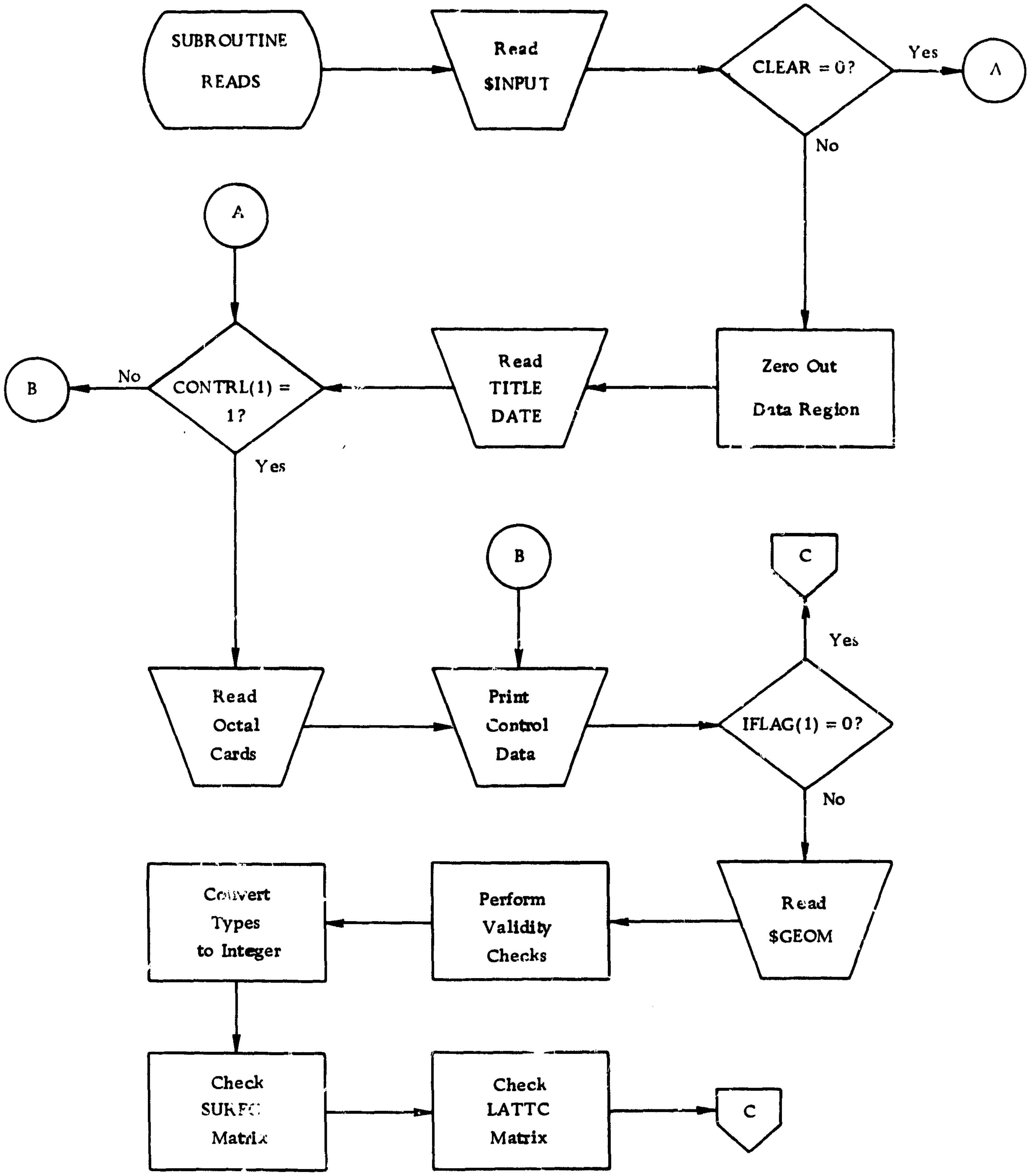
d) Processes symbol cards by calculating and preserving the DATA index for each valid symbol which is read.

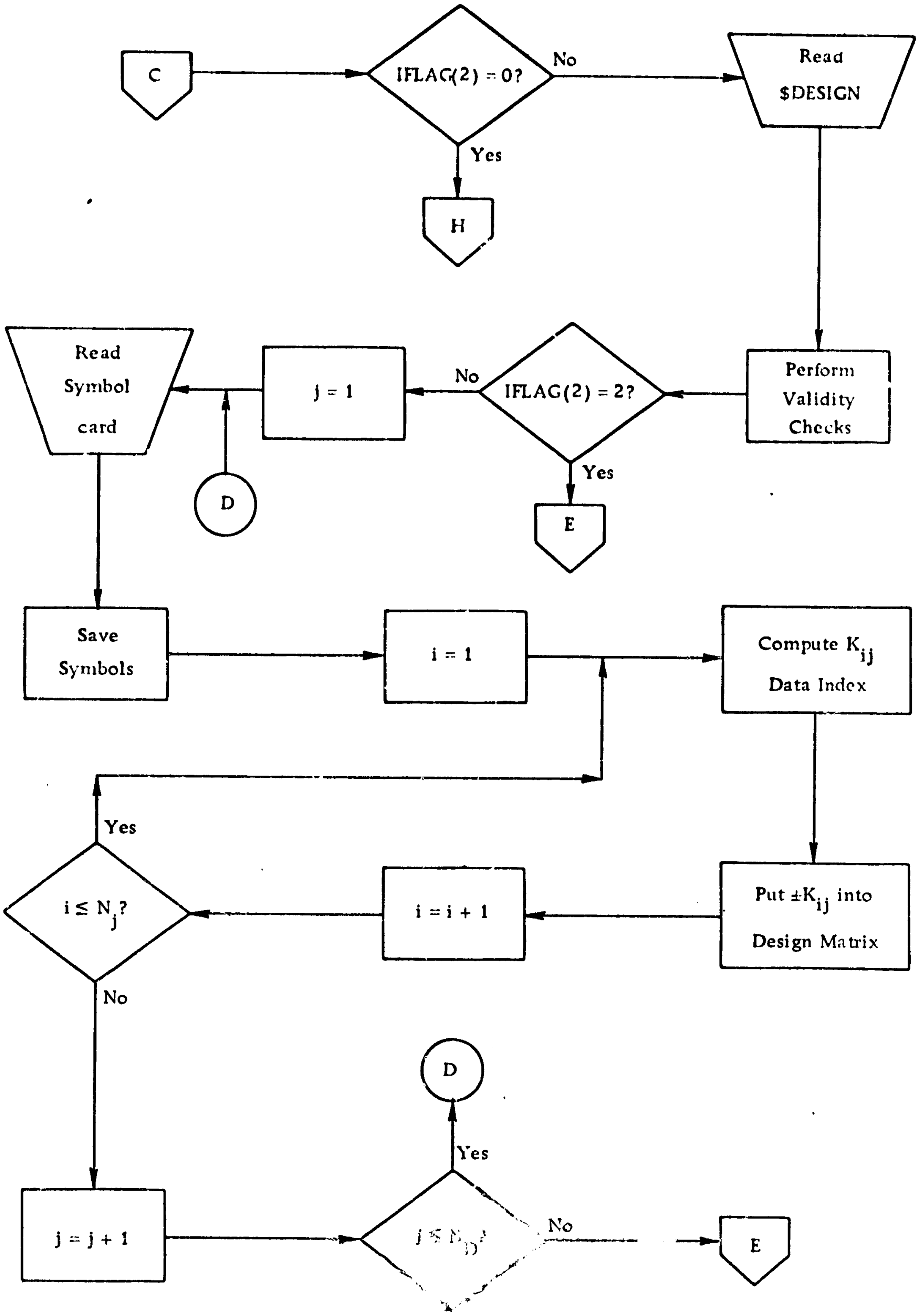
Calling Sequence

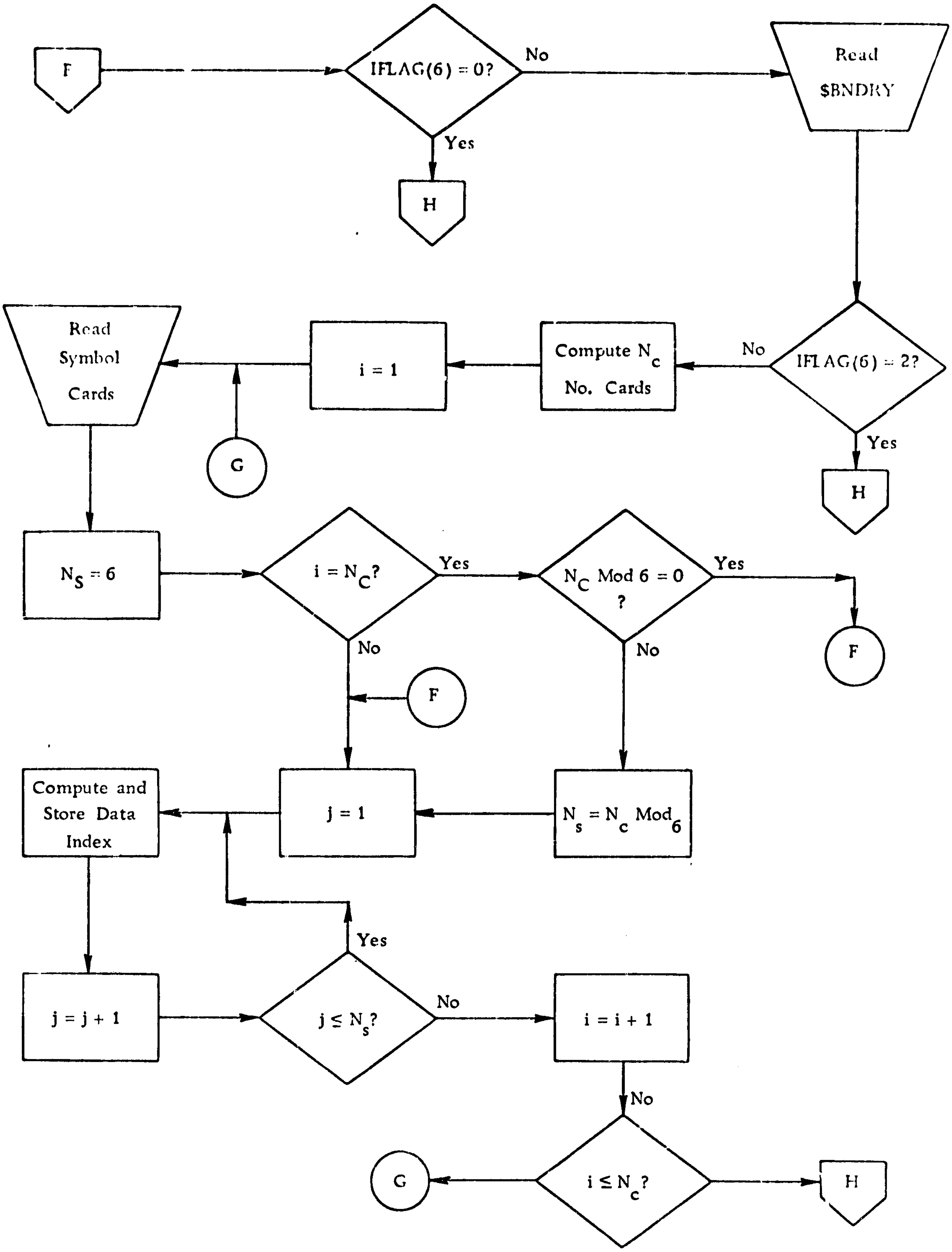
CALL READS

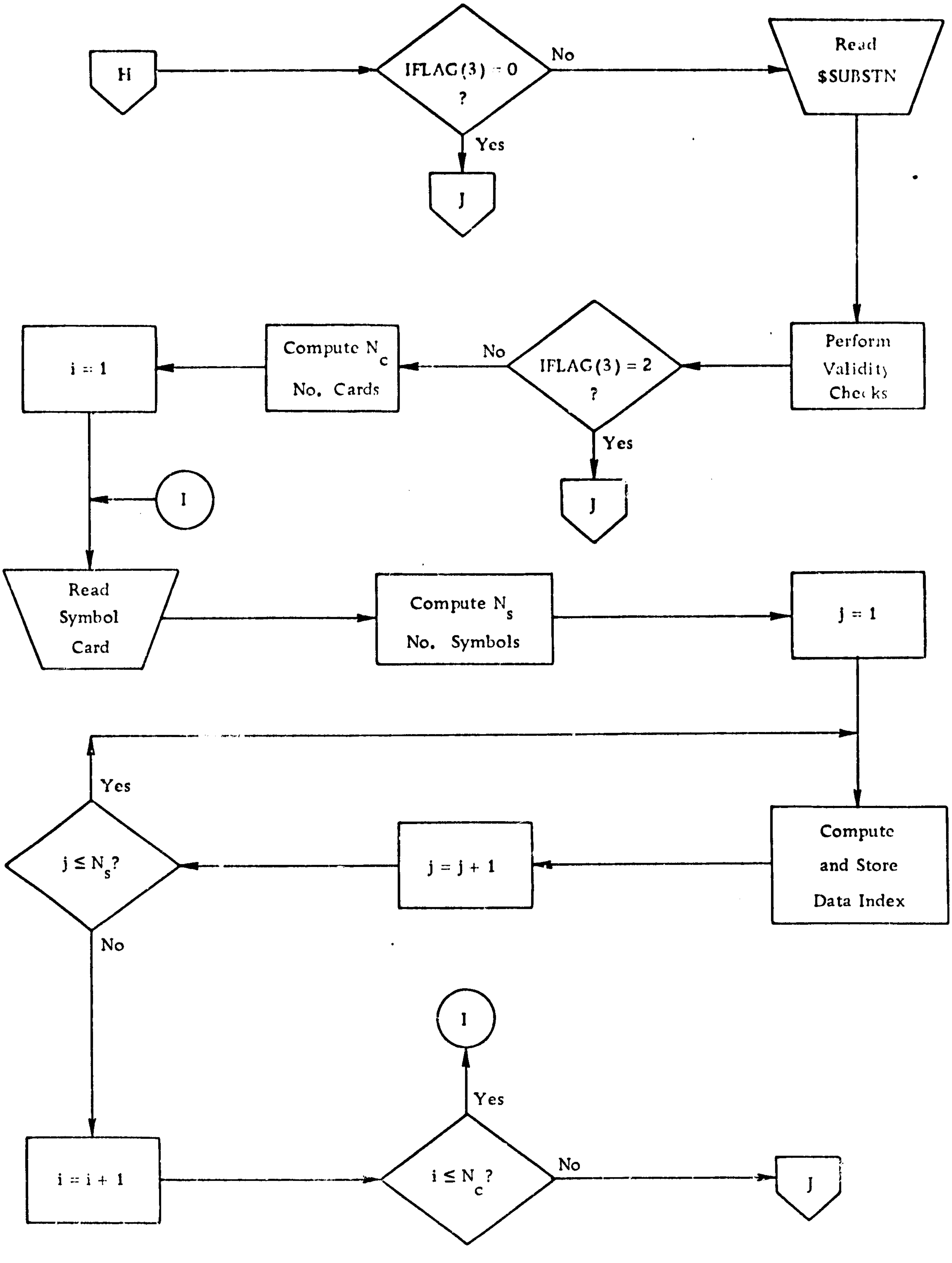
Utility Routines and Common References

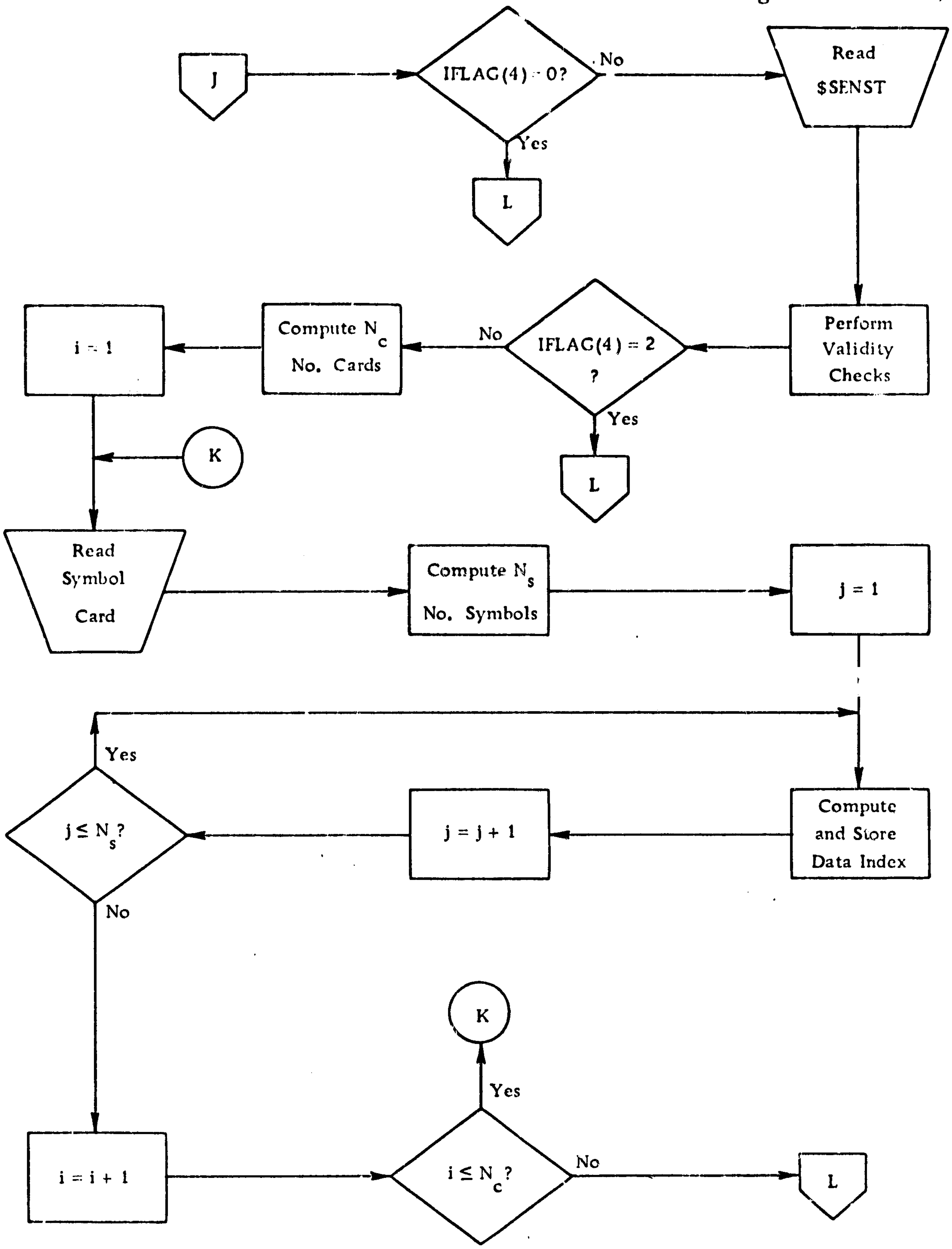
COMMON/DATA/	VCNVRT
COMMON/SYMBLS/	ERROR
COMMON/PRNT/	ERROR1
COMMON/PERTB/	ERROR2
COMMON/PLOTC/	ERROR3
PRTCTL	ERROR4
MCNVRT	

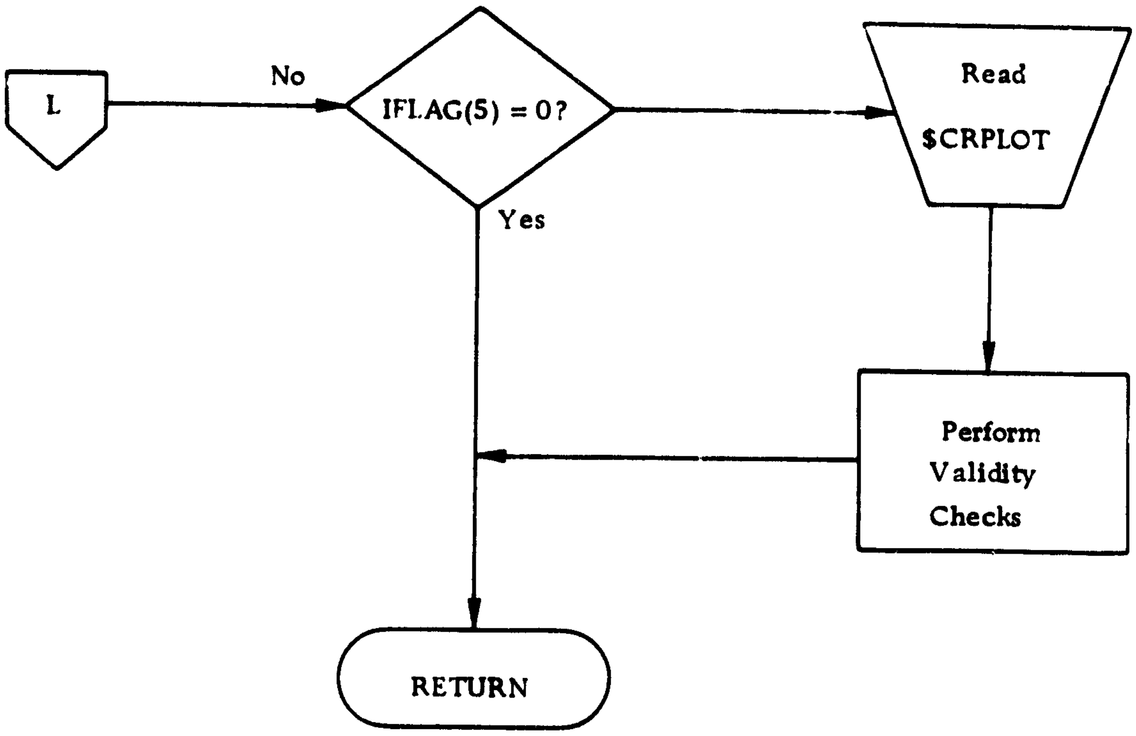

SUBROUTINE READS
Read $INPUT
CLEAR = 0?
Yes
A
No
A
Zero Out Data Region
Read TITLE DATE
CONTRL(1) = 1?
No
B
Yes
B
C
Yes
Read Octal Cards
Print Control Data
IFLAG(1) = 0?
No
Read $GEOM
Perform Validity Checks
Convert Types to Integer
Check SURFC Matrix
Check LATTC Matrix
C


C
IFLAG(2) = 0?
No
Yes
H
Read
$DESIGN
Perform
Validity
Checks
IFLAG(2) = 2?
No
Yes
E
j = 1
Read
Symbol
card
D
Save
Symbols
i = 1
Compute K_{ij}
Data Index
Put $\pm K_{ij}$ into
Design Matrix
i = i + 1
$i \le N_j$?
Yes
No
j = j + 1
$j \le N_D$?
Yes
No
D
E


F
IFLAG(6) = 0?
No
Yes
H
Read
$BNDRY
IFLAG(6) = 2?
No
Yes
H
Compute N
c
No. Cards
i = 1
Read
Symbol
Cards
G
N
S
= 6
i = N
C
?
Yes
No
N
C
Mod 6 = 0
?
Yes
No
F
F
j = 1
N
s
= N
c
Mod
6
Compute and
Store Data
Index
j = j + 1
j ≤ N
s
?
Yes
No
i = i + 1
No
i ≤ N
c
?
G
H


H
IFLAG(3) = 0 ?
No
Yes
J
Read $SUBSTN
Perform Validity Checks
IFLAG(3) = 2 ?
No
Yes
J
Compute N_c No. Cards
i = 1
I
Read Symbol Card
Compute N_s No. Symbols
j = 1
Compute and Store Data Index
j = j + 1
$j \leq N_s$?
Yes
No
i = i + 1
$i \leq N_c$?
Yes
I
No
J


J
IFLAG(4) = 0?
No
Yes
L
Read
$SENST
Perform
Validity
Checks
IFLAG(4) = 2
?
No
Yes
L
Compute N c
No. Cards
i = 1
K
Read
Symbol
Card
Compute N s
No. Symbols
j = 1
Compute
and Store
Data Index
j = j + 1
j ≤ N s ?
Yes
No
i = i + 1
i ≤ N c ?
Yes
K
No
L


L
No
IFLAG(5) = 0?
Yes
Read
$CRPLOT
Perform
Validity
Checks
RETURN

3.1.3 RERZZ (REREAD)

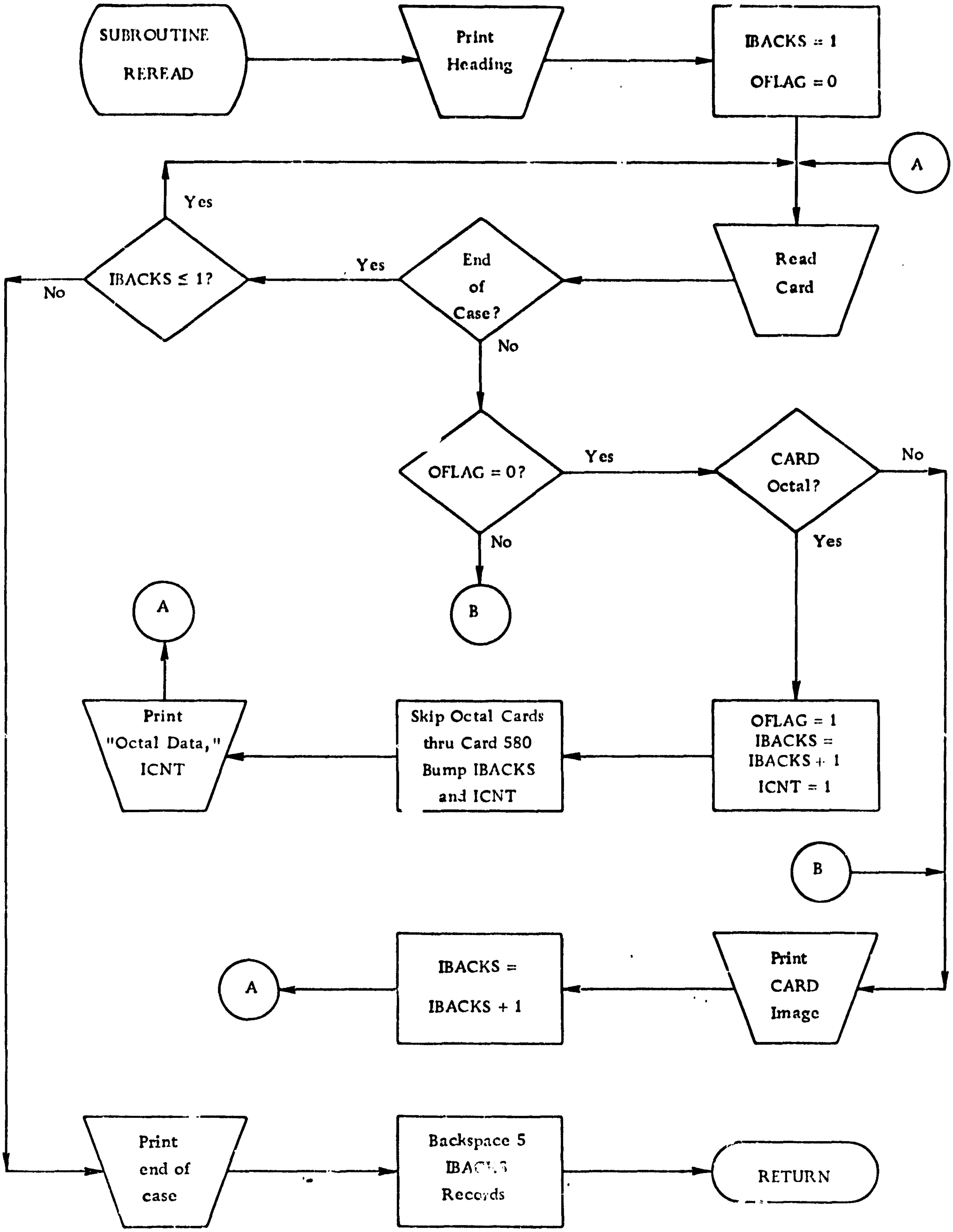
REREZZ reads card images from the system input file (logical 5) in an 80A1 format until an END OF CASE card is encountered. Each card (except octal) which is read is listed on the system output file. A count of the number of cards is kept so that REREZZ can reposition the input file by executing the 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


SUBROUTINE REREAD
Print Heading
IBACKS = 1
OFLAG = 0
A
Read Card
End of Case?
Yes
No
IBACKS ≤ 1?
Yes
No
OFLAG = 0?
Yes
No
CARD Octal?
Yes
No
B
OFLAG = 1
IBACKS = IBACKS + 1
ICNT = 1
Skip Octal Cards thru Card 580 Bump IBACKS and ICNT
Print "Octal Data," ICNT
A
B
Print CARD Image
IBACKS = IBACKS + 1
A
Print end of case
Backspace 5 IBACKS Records
RETURN

3.1.4 STAR9Z (STAR9)

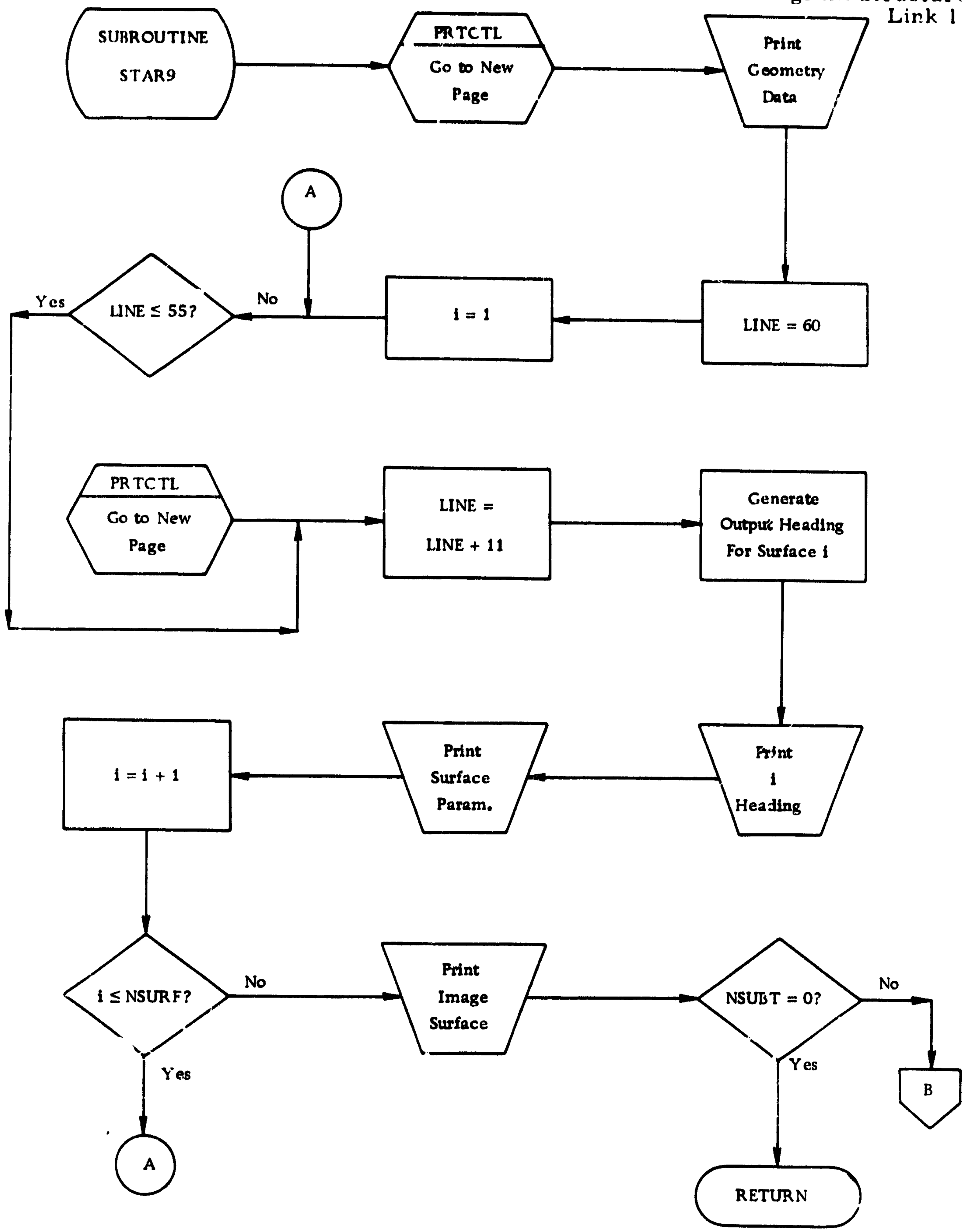
STAR9Z lists all of the geometry data with descriptive headings and if NSUBT $\neq 0$ it also lists the substitution data by parameter and by set.

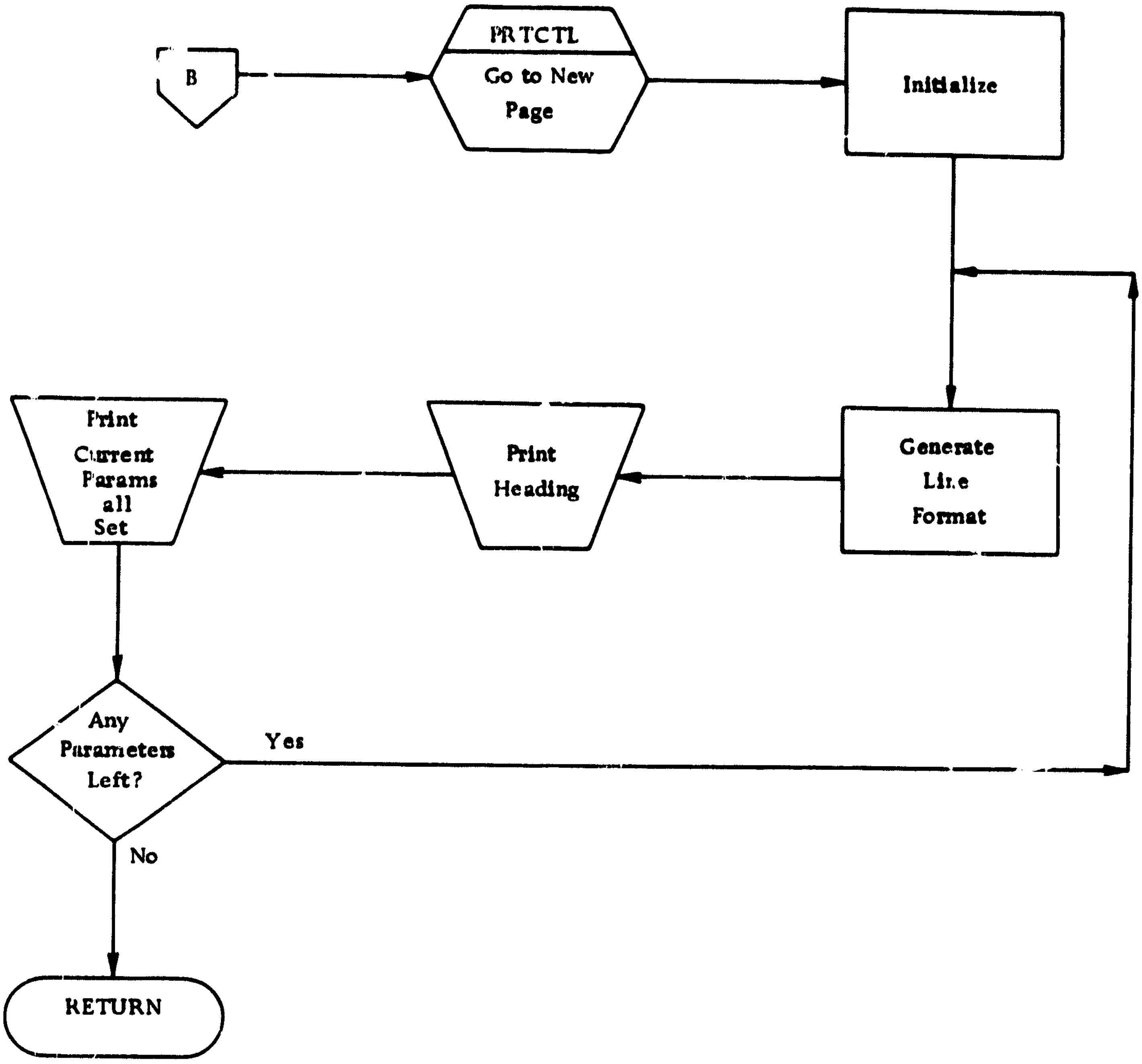
Calling Sequence

CALL STAR9

Utility Routines and Common References

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


SUBROUTINE
STAR9
PRTCTL
Go to New
Page
Print
Geometry
Data
A
Yes
LINE ≤ 55?
No
i = 1
LINE = 60
PRTCTL
Go to New
Page
LINE =
LINE + 11
Generate
Output Heading
For Surface i
i = i + 1
Print
Surface
Param.
Print
i
Heading
i ≤ NSURF?
No
Yes
A
Print
Image
Surface
NSUBT = 0?
No
Yes
B
RETURN


B
PRTCTL
Go to New Page
Initialize
Generate Line Format
Print Heading
Print Current Params all Set
Any Parameters Left?
Yes
No
RETURN

3.1.5 STAR1Z (STAR10)

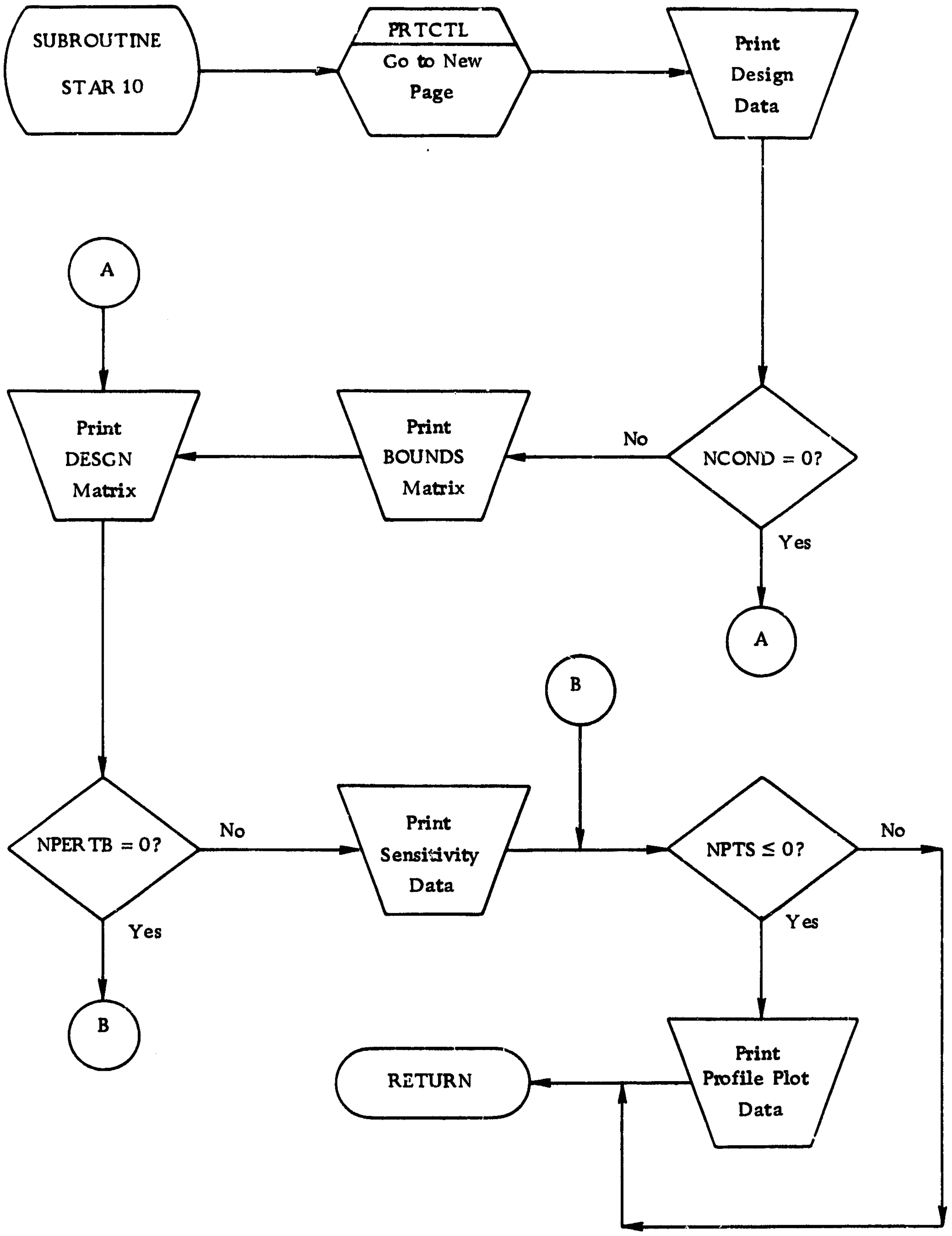
STAR1Z lists all of the design data with descriptive headings as well as the boundary condition matrix (if NCOND > 0) sensitivity data (if NPERTB > 0), and profile plot data (if NPTS > 0).

Calling Sequence

CALL STAR10

Utility Routines and Common References

COMMON/DATA/
COMMON/SYMBLS/
COMMON/PRNT/
COMMON/PERTB/
COMMON/PLOTC/
PRTCTL


SUBROUTINE
STAR 10
PRTCTL
Go to New Page
Print Design Data
A
Print DESGN Matrix
Print BOUNDS Matrix
No
NCOND = 0?
Yes
A
B
NPERTB = 0?
No
Yes
B
Print Sensitivity Data
NPTS ≤ 0?
No
Yes
Print Profile Plot Data
RETURN

3.1.6 MTFPNZ (MTFPN)

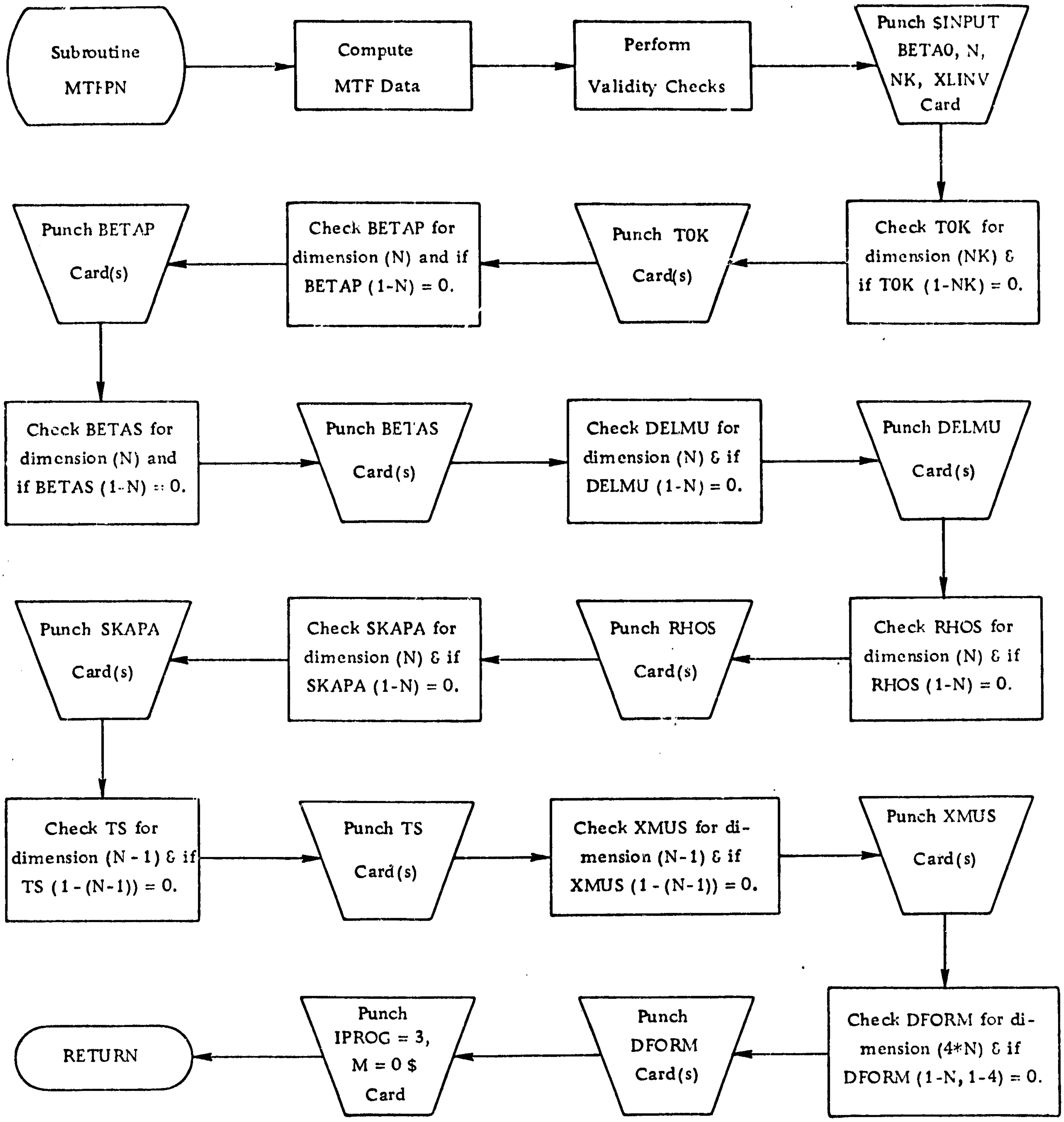
MTFPNZ punches a deck of cards in NAMELIST format which, when input to optical system analysis program PAGOS, provides data necessary for the Modulation Transfer Function (MTF) of that program.

Calling Sequence

CALL MTFPN

Utility Routines and Common References

COMMON/DATA/
ERROR2


Subroutine MTFPN
Compute MTF Data
Perform Validity Checks
Punch $INPUT BETA0, N, NK, XLINV Card
Check TOK for dimension (NK) & if TOK (1-NK) = 0.
Punch TOK Card(s)
Check BETAP for dimension (N) and if BETAP (1-N) = 0.
Punch BETAP Card(s)
Check BETAS for dimension (N) and if BETAS (1-N) = 0.
Punch BETAS Card(s)
Check DELMU for dimension (N) & if DELMU (1-N) = 0.
Punch DELMU Card(s)
Check RHOS for dimension (N) & if RHOS (1-N) = 0.
Punch RHOS Card(s)
Check SKAPA for dimension (N) & if SKAPA (1-N) = 0.
Punch SKAPA Card(s)
Check TS for dimension (N - 1) & if TS (1 - (N-1)) = 0.
Punch TS Card(s)
Check XMUS for dimension (N-1) & if XMUS (1 - (N-1)) = 0.
Punch XMUS Card(s)
Check DFORM for dimension (4*N) & if DFORM (1-N, 1-4) = 0.
Punch DFORM Card(s)
Punch IPROG = 3, M = 0 $ Card
RETURN

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.

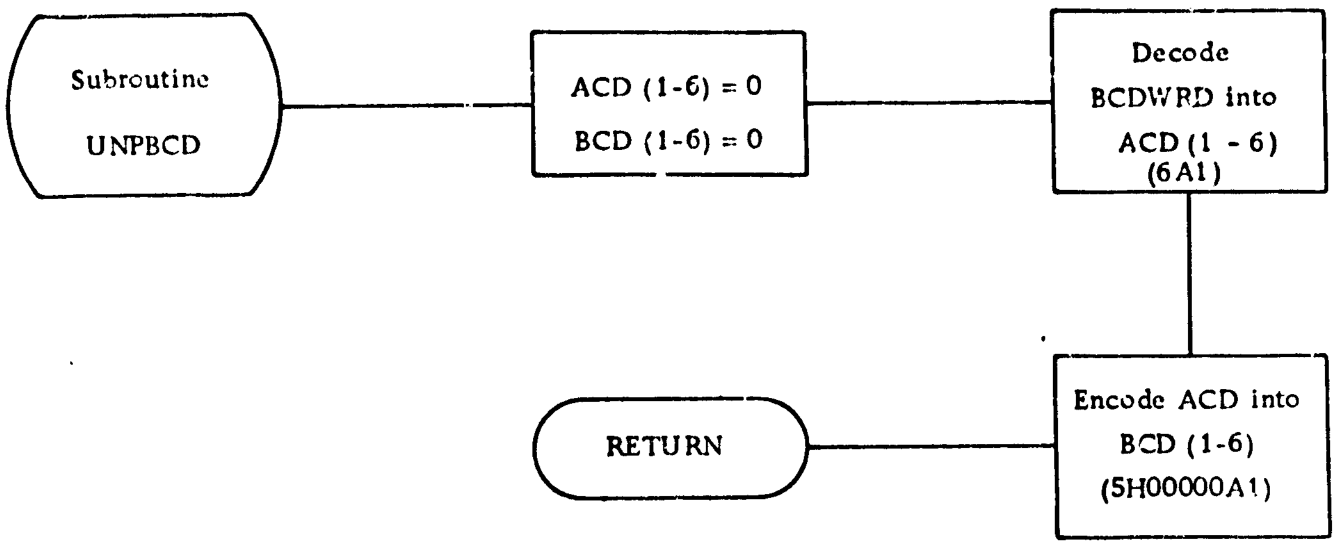
Calling Sequence

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


Subroutine
UNPBCD
ACD (1-6) = 0
BCD (1-6) = 0
Decode
BCDWRD into
ACD (1 - 6)
(6A1)
Encode ACD into
BCD (1-6)
(5H00000A1)
RETURN

3.1.8 VCNVR. (VCNVRT, MCNVRT)

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)

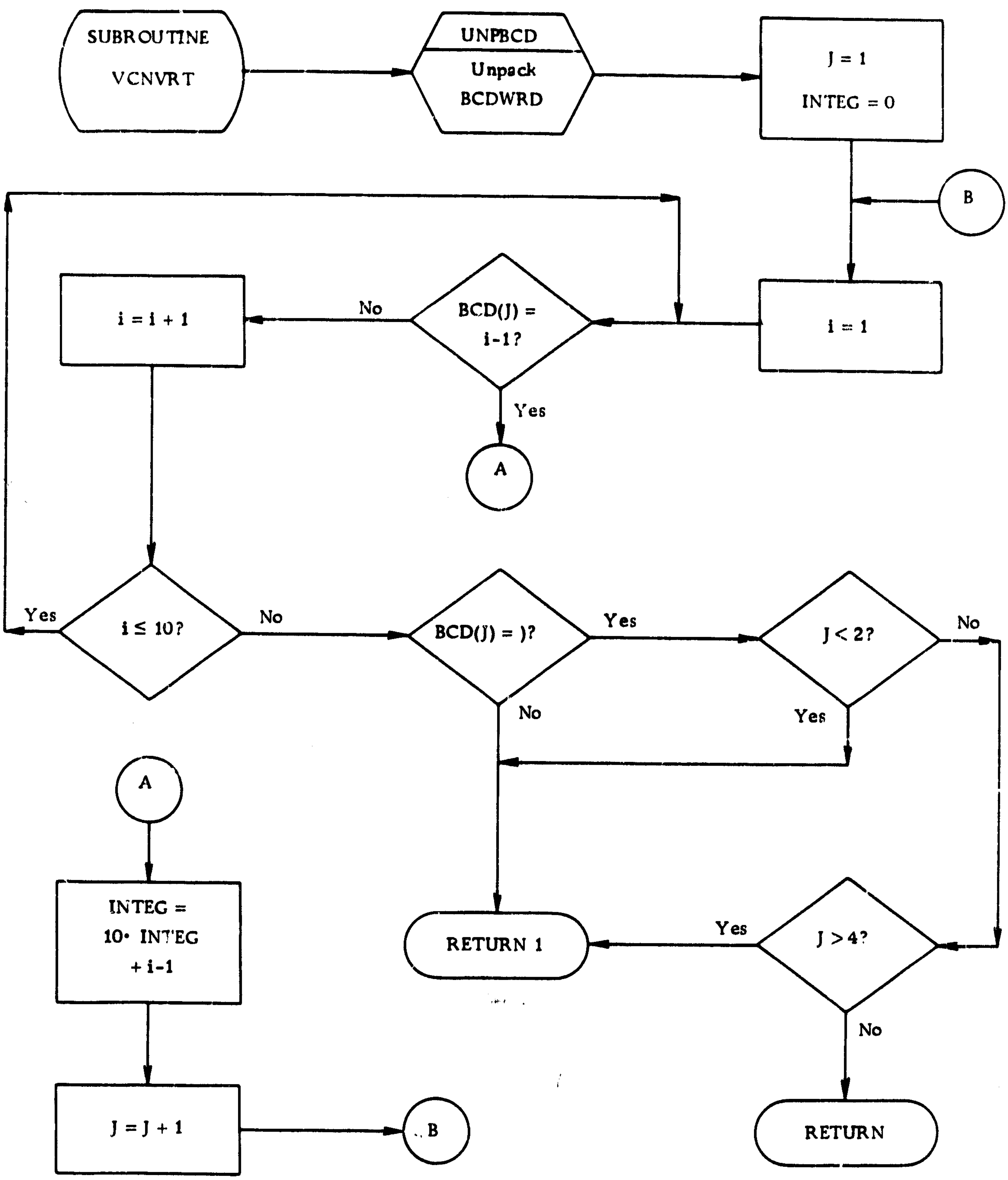
where BCDWRD contains a BCD vector subscript of the form X)bbbb. XX)bbb, or XXX)bb and ISUB is the location where the integer subscript is to be stored. If VCNVRT detects an error the alternate return $ERR is invoked.

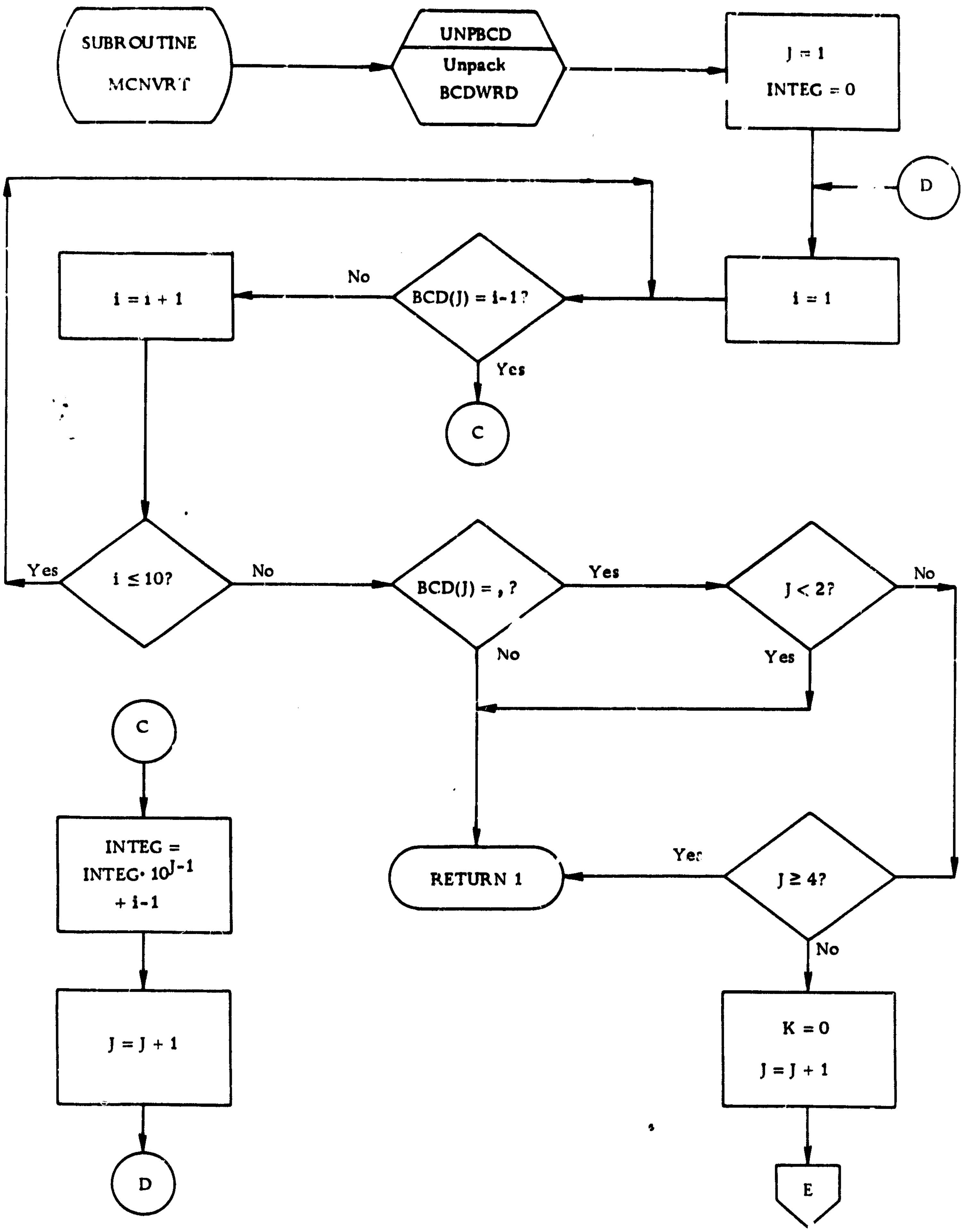
CALL MCNVRT (BCDWRD, ISUB, JSUB, $ERR)

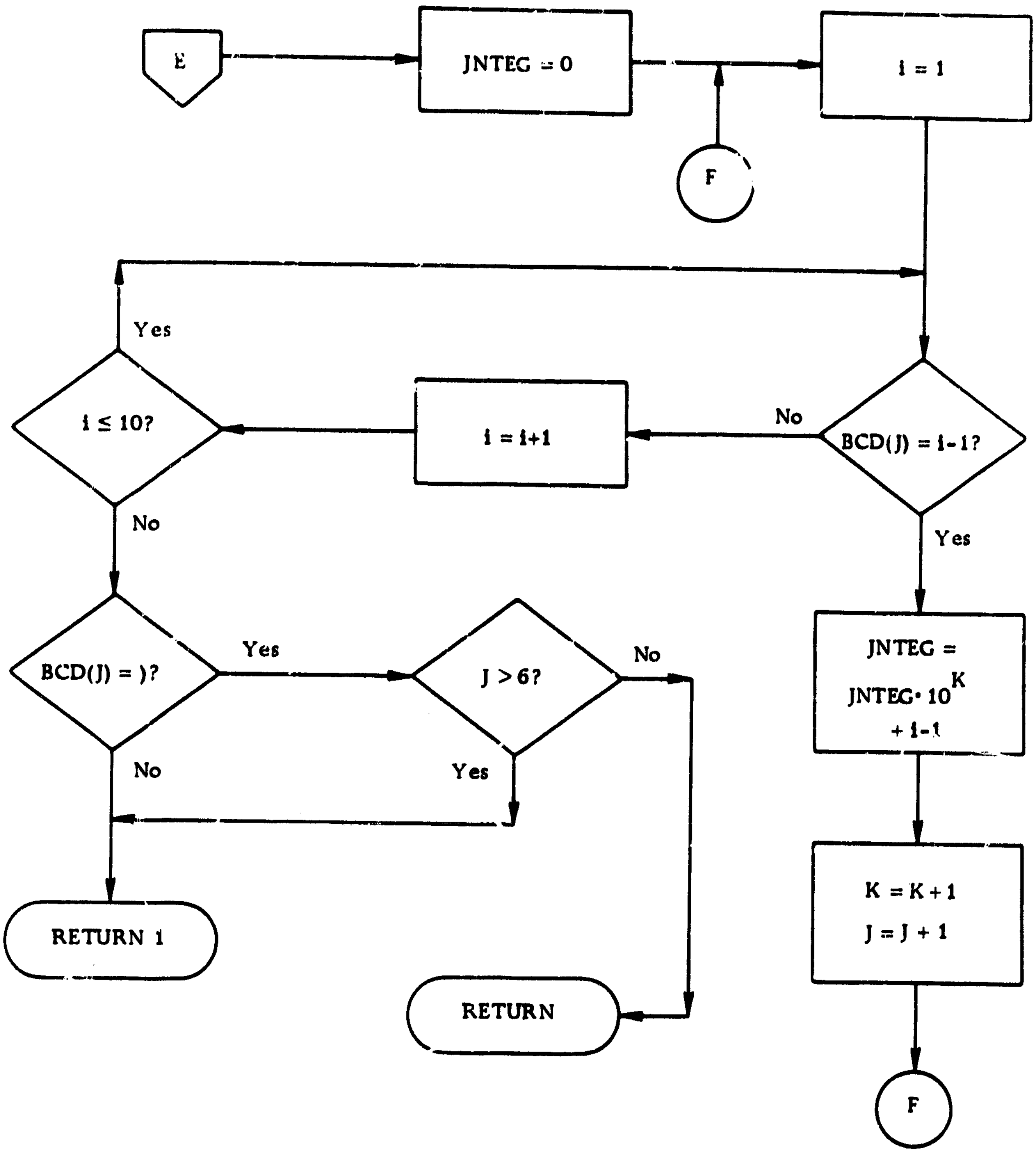
where BCDWRD contains a BCD matrix subscript of the form X, Y)bb, X, YY)b, XX, Y)b, or XX, YY), ISUB is the location where the integer row subscript is to be stored, and JSUB is the location where the integer column subscript is to be stored. The alternate return $ERR is invoked if an error is detected.

Utility Routines and Common References

COMMON - None
UNPBCD


SUBROUTINE VCNVRT
UNPBCD
Unpack BCDWRD
J = 1
INTEG = 0
B
i = 1
BCD(J) = i-1?
No
Yes
A
i = i + 1
i ≤ 10?
Yes
No
BCD(J) =)?
Yes
No
J < 2?
No
Yes
A
INTEG = 10· INTEG + i-1
RETURN 1
Yes
J > 4?
No
J = J + 1
B
RETURN


SUBROUTINE MCNVRT
UNPBCD
Unpack BCDWRD
J = 1
INTEG = 0
D
i = 1
BCD(J) = i-1?
No
Yes
i = i + 1
C
i ≤ 10?
Yes
No
BCD(J) = , ?
Yes
No
J < 2?
No
Yes
C
INTEG = INTEG· 10^J-1 + i-1
J = J + 1
D
RETURN 1
Yes
J ≥ 4?
No
K = 0
J = J + 1
E


E
JNTEG = 0
i = 1
F
Yes
No
i ≤ 10?
i = i+1
BCD(J) = i-1?
No
Yes
JNTEG =
JNTEG· 10^K
+ i-1
BCD(J) =)?
Yes
J > 6?
No
No
Yes
K = K + 1
J = J + 1
RETURN 1
RETURN
F

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

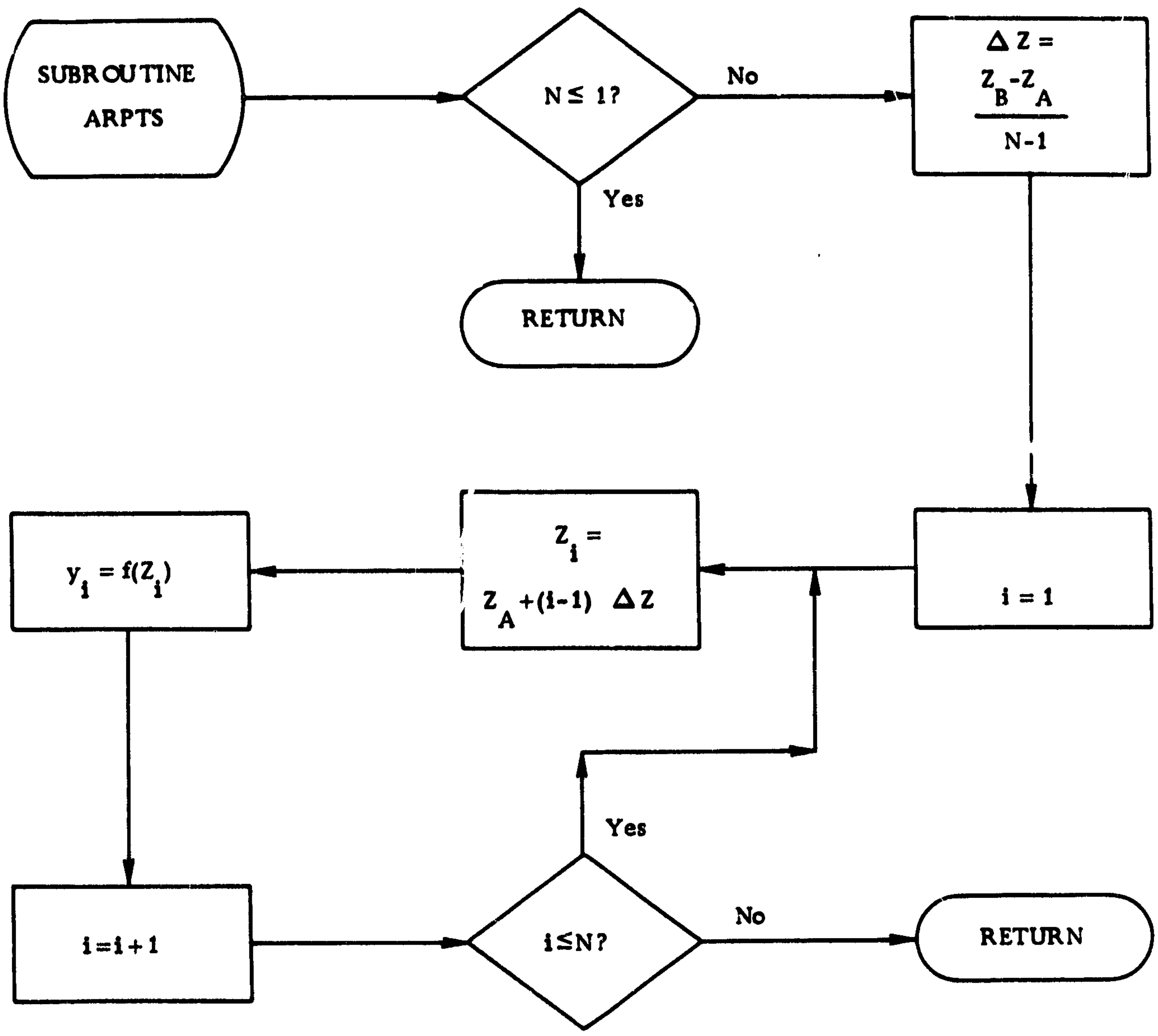
CALL ARPTS (IS, ZA, YA, ZB, YB, N, ZLST, YLST)

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


SUBROUTINE ARPTS
N ≤ 1?
No
Yes
ΔZ = (Z_B - Z_A)/(N-1)
RETURN
i = 1
Z_i = Z_A + (i-1) ΔZ
y_i = f(Z_i)
i = i + 1
i ≤ N?
Yes
No
RETURN

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.

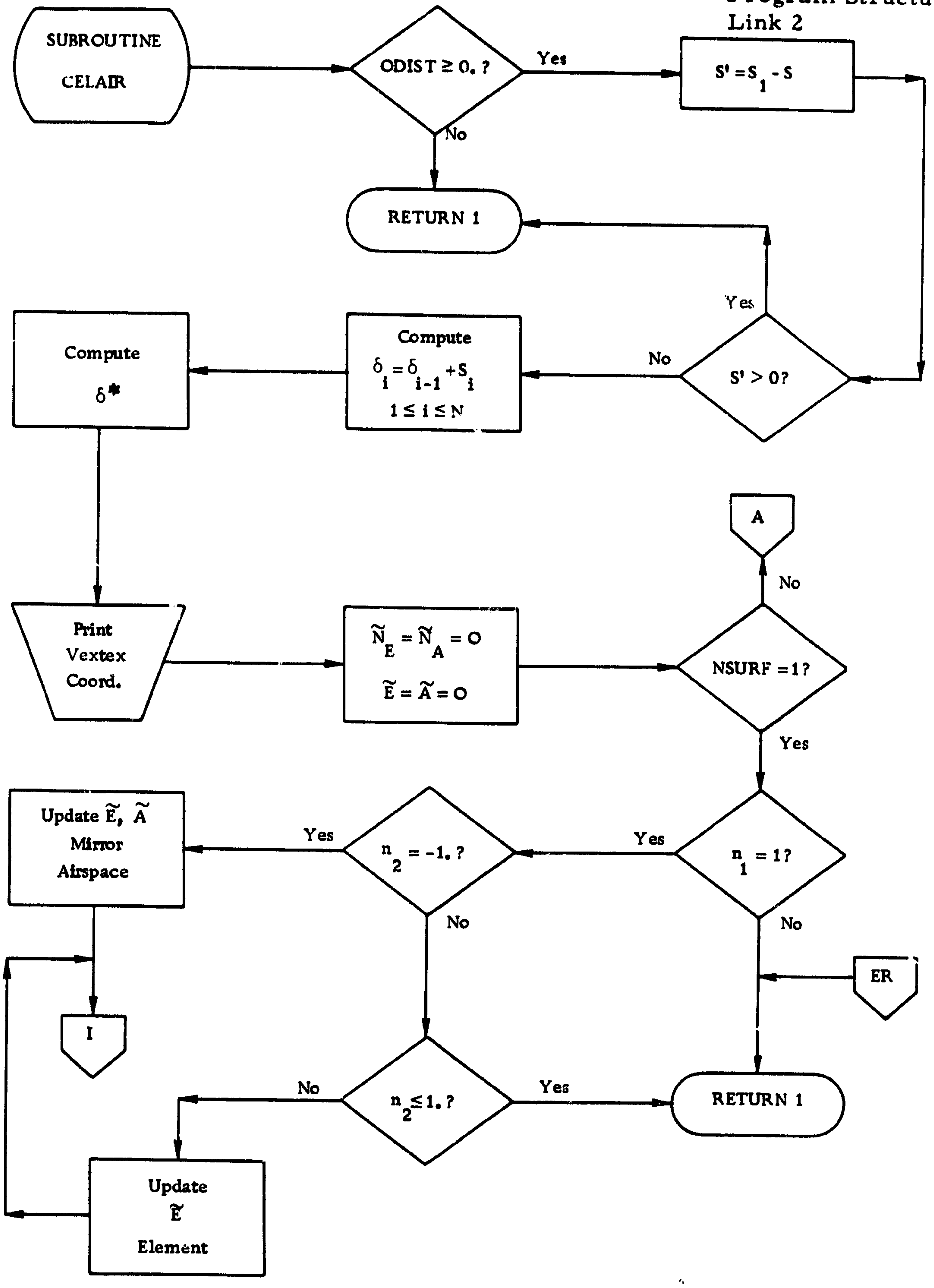
Calling Sequence

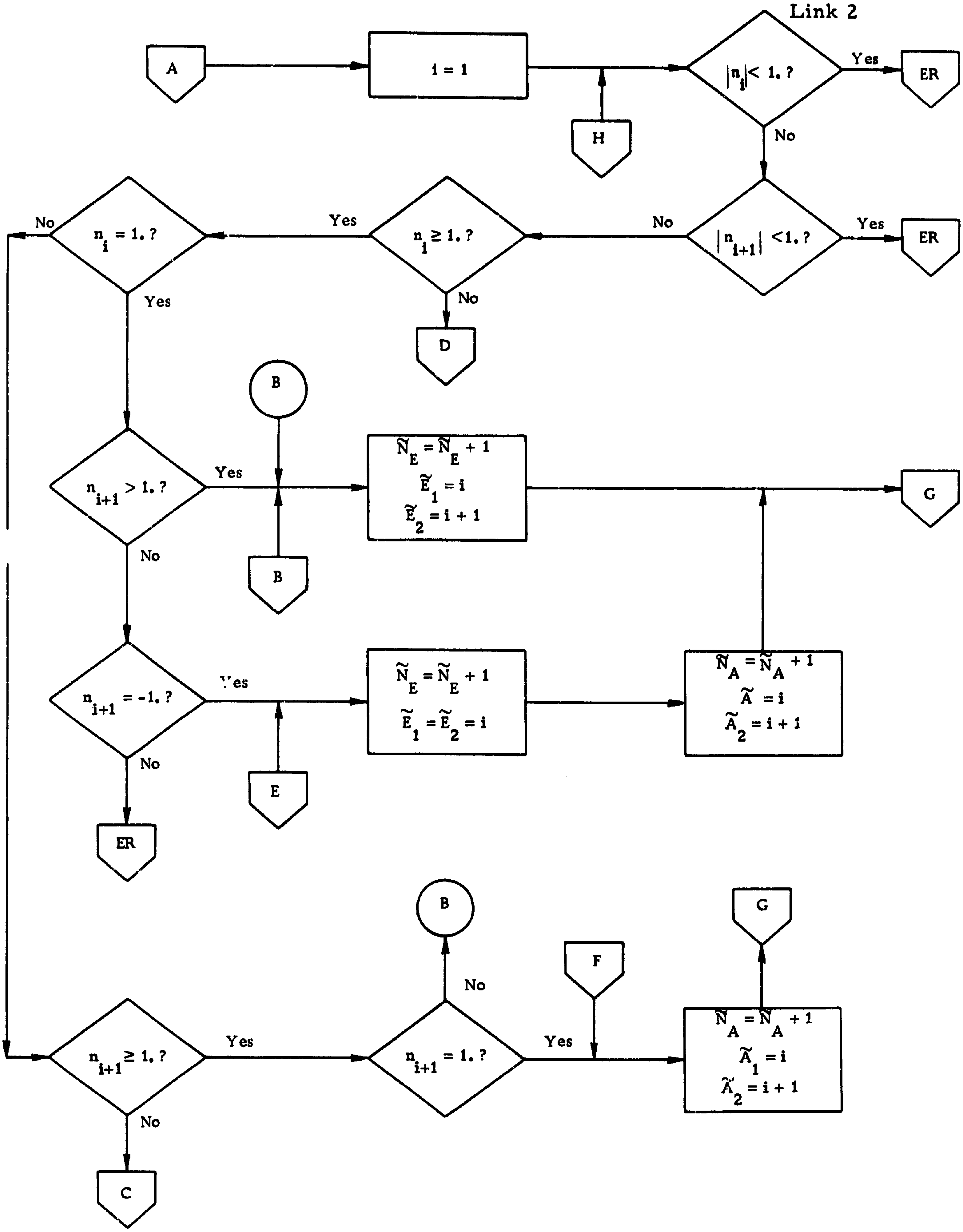
CALL CELAIR($ALT)

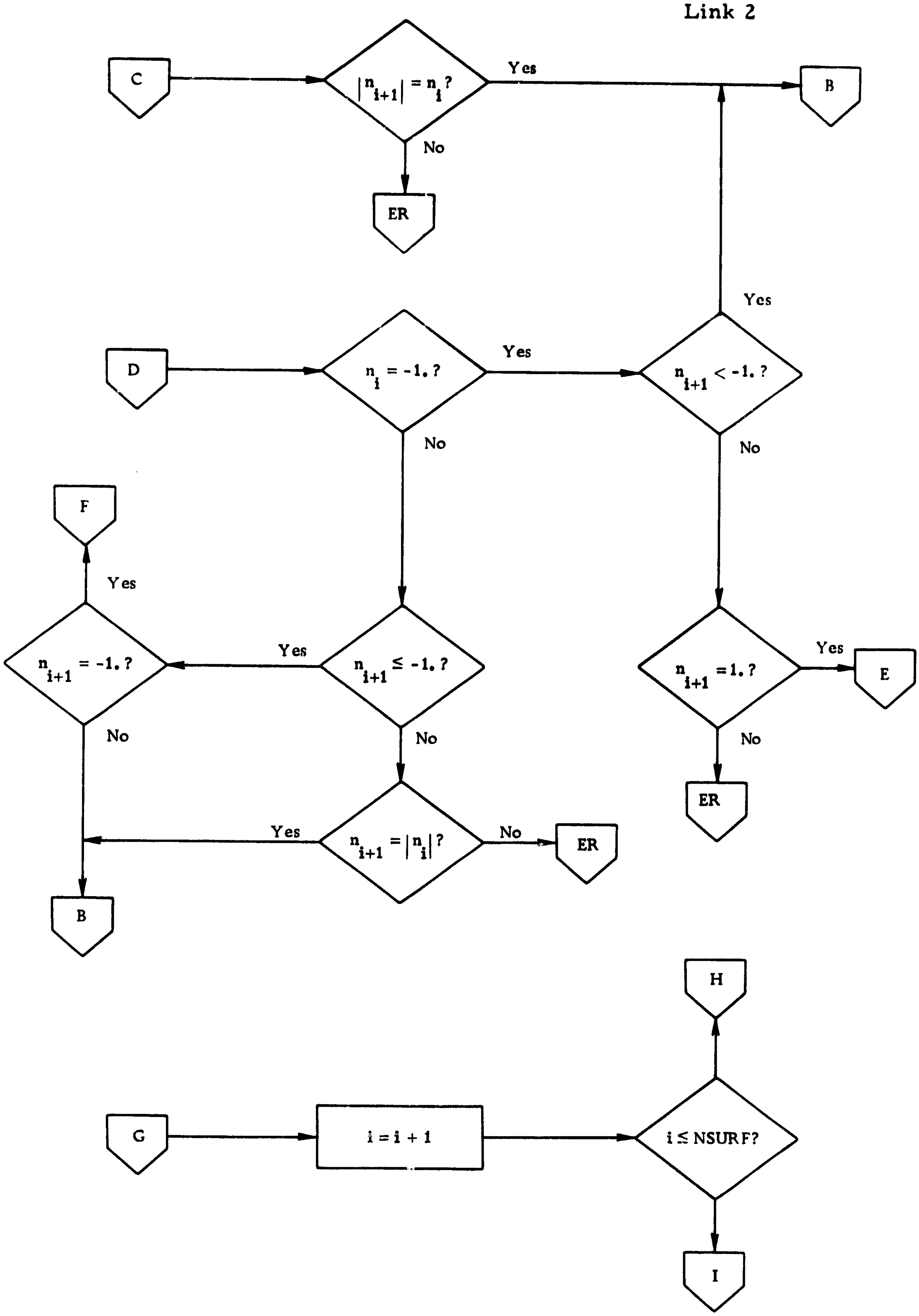
The alternate return is invoked if CELAIR determines that a profile plot cannot be produced.

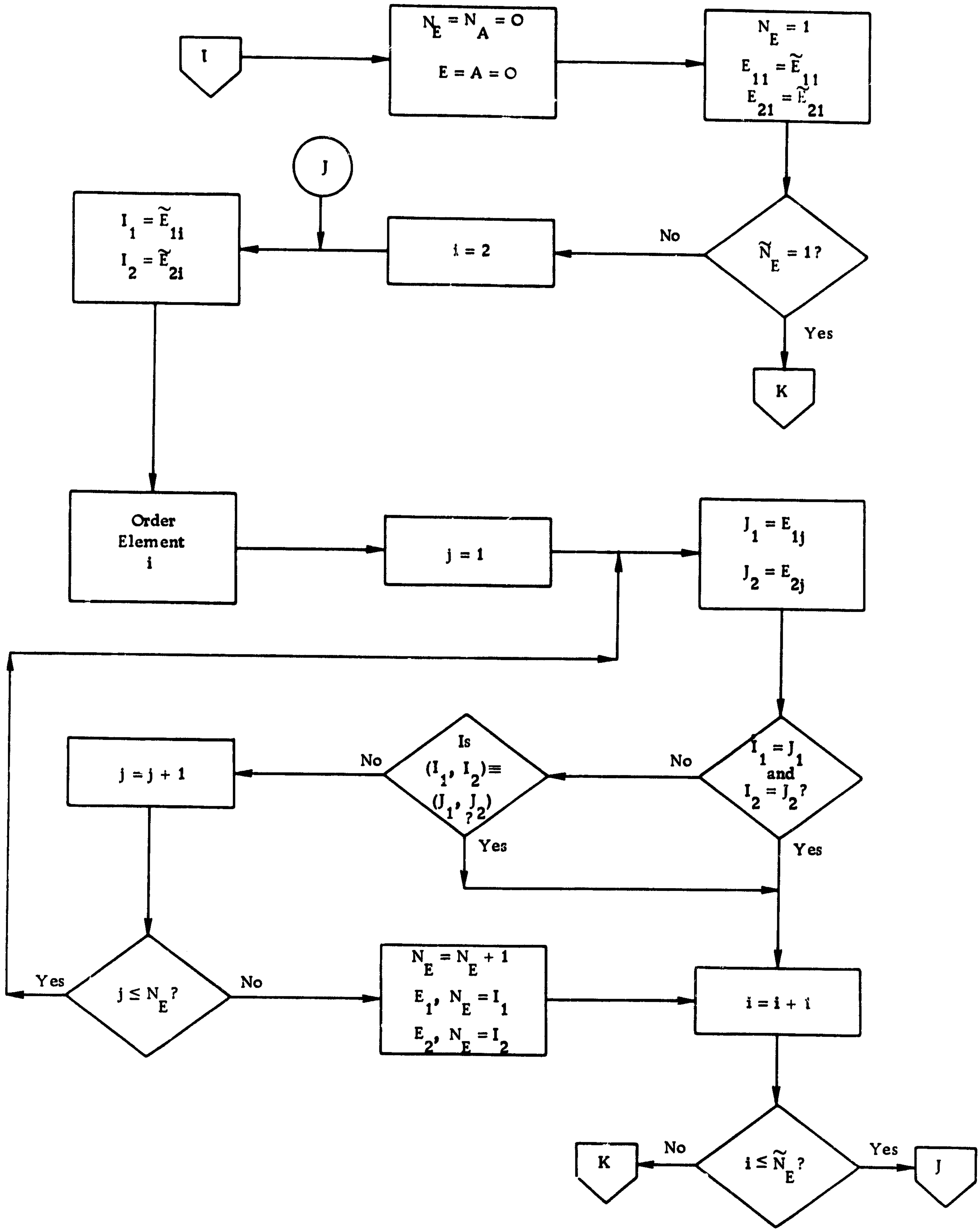
Utility Routines and Common References

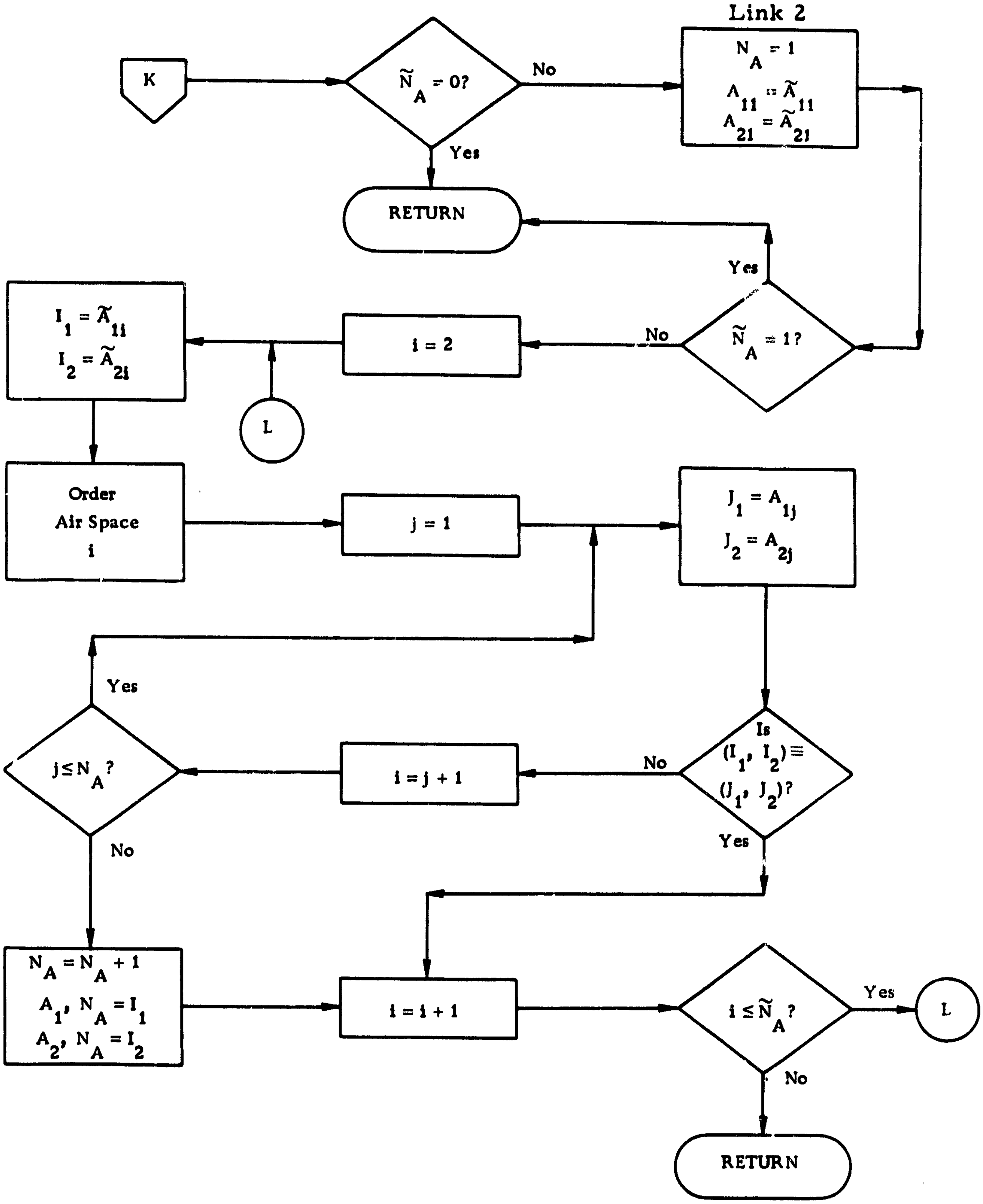
COMMON/DATA/
COMMON/PRNT/
COMMON/ELMAIR/
PRTCTL


SUBROUTINE CELAIR
ODIST ≥ 0. ?
Yes
No
S' = S1 - S
RETURN 1
Yes
No
S' > 0?
Compute δi = δi-1 + Si 1 ≤ i ≤ N
Compute δ*
Print Vextex Coord.
ÑE = ÑA = O
Ẽ = Ã = O
NSURF = 1?
No
A
Yes
n1 = 1?
Yes
No
n2 = -1. ?
Yes
No
Update Ẽ, Ã Mirror Airspace
I
ER
n2 ≤ 1. ?
No
Yes
RETURN 1
Update Ẽ Element


A
$i = 1$
H
$|n_i| < 1.$?
Yes
ER
No
$|n_{i+1}| < 1.$?
Yes
ER
No
$n_i \geq 1.$?
Yes
No
D
$n_i = 1.$?
No
Yes
B
$n_{i+1} > 1.$?
Yes
B
$\tilde{N}_E = \tilde{N}_E + 1$
$\tilde{E}_1 = i$
$\tilde{E}_2 = i + 1$
G
No
$n_{i+1} = -1.$?
Yes
E
$\tilde{N}_E = \tilde{N}_E + 1$
$\tilde{E}_1 = \tilde{E}_2 = i$
$\tilde{N}_A = \tilde{N}_A + 1$
$\tilde{A} = i$
$\tilde{A}_2 = i + 1$
No
ER
B
G
F
No
$n_{i+1} \geq 1.$?
Yes
$n_{i+1} = 1.$?
Yes
$\tilde{N}_A = \tilde{N}_A + 1$
$\tilde{A}_1 = i$
$\tilde{A}_2 = i + 1$
No
C

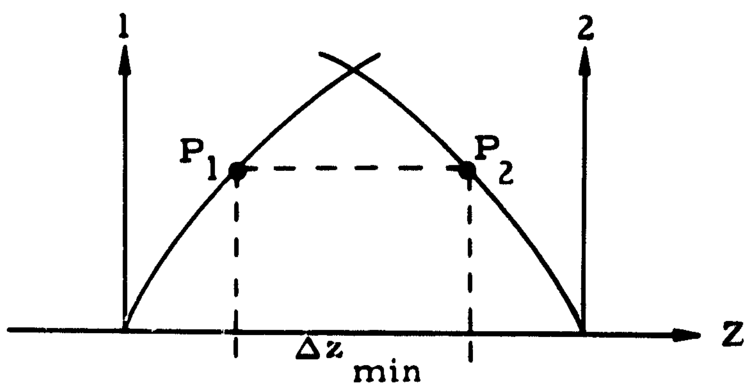

C
|n i+1| = n i ?
Yes
No
B
ER
D
n i = -1. ?
Yes
No
n i+1 < -1. ?
Yes
No
F
Yes
n i+1 = -1. ?
No
Yes
n i+1 ≤ -1. ?
No
n i+1 = 1. ?
Yes
No
E
ER
Yes
n i+1 = |n i| ?
No
ER
B
H
G
i = i + 1
i ≤ NSURF?
I


I
$N_E = N_A = 0$
$E = A = 0$
$N_E = 1$
$E_{11} = \tilde{E}_{11}$
$E_{21} = \tilde{E}_{21}$
J
$I_1 = \tilde{E}_{1i}$
$I_2 = \tilde{E}_{2i}$
$i = 2$
No
$\tilde{N}_E = 1?$
Yes
K
Order Element i
$j = 1$
$J_1 = E_{1j}$
$J_2 = E_{2j}$
$j = j + 1$
No
Is $(I_1, I_2) \equiv (J_1, J_2)$?
No
$I_1 = J_1$ and $I_2 = J_2$?
Yes
Yes
Yes
$j \le N_E$?
No
$N_E = N_E + 1$
$E_1, N_E = I_1$
$E_2, N_E = I_2$
$i = i + 1$
K
No
$i \le \tilde{N}_E$?
Yes
J


K
$\tilde{N}_A = 0?$
No
Yes
$N_A = 1$
$A_{11} = \tilde{A}_{11}$
$A_{21} = \tilde{A}_{21}$
RETURN
Yes
$\tilde{N}_A = 1?$
No
$i = 2$
$I_1 = \tilde{A}_{1i}$
$I_2 = \tilde{A}_{2i}$
L
Order
Air Space
i
$j = 1$
$J_1 = A_{1j}$
$J_2 = A_{2j}$
Yes
$j \leq N_A?$
No
$i = j + 1$
No
Is
$(I_1, I_2) \equiv (J_1, J_2)?$
Yes
$N_A = N_A + 1$
$A_1, N_A = I_1$
$A_2, N_A = I_2$
$i = i + 1$
$i \leq \tilde{N}_A?$
Yes
L
No
RETURN

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)

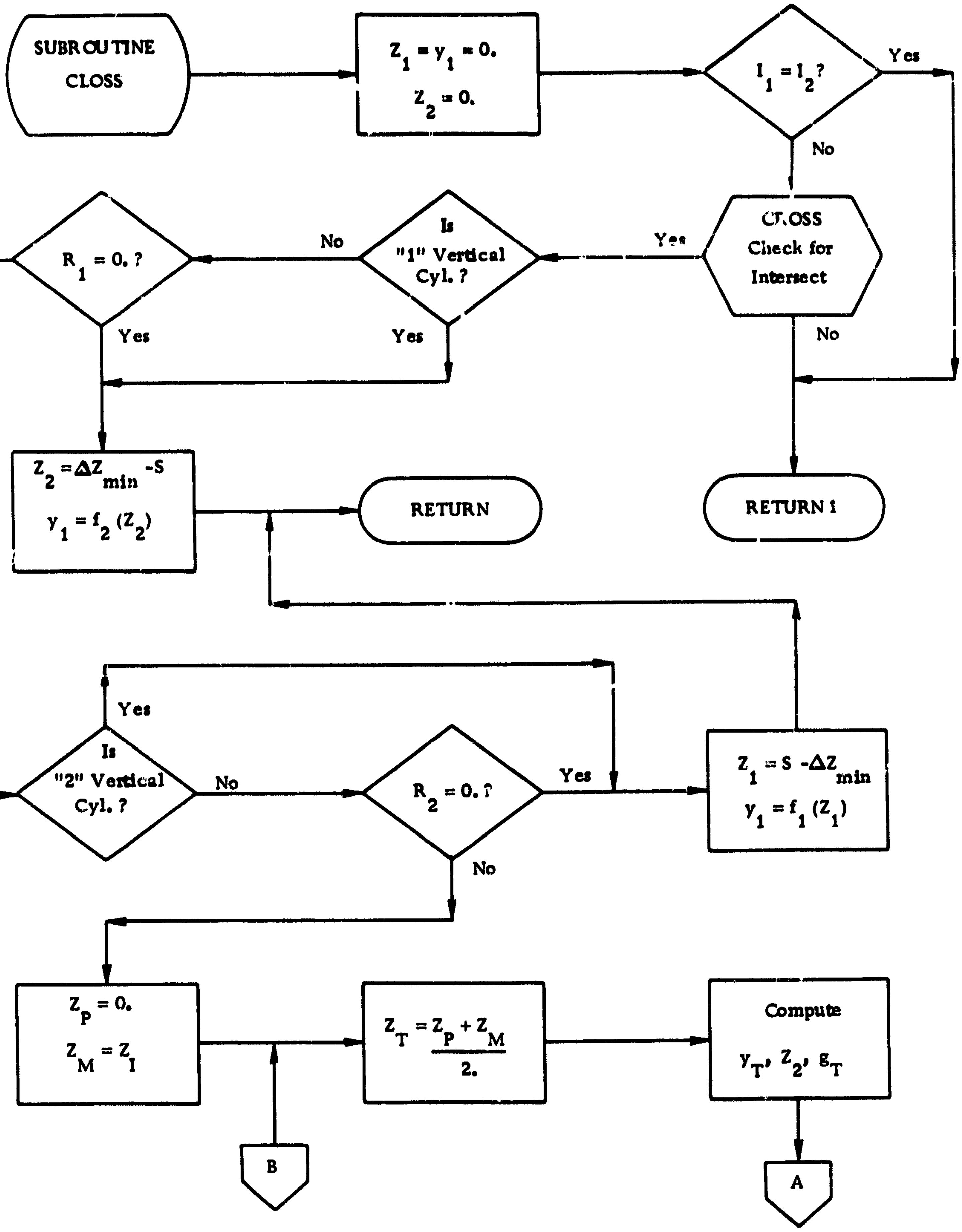
I1:	Surface number of $y = f_1(z)$
Z1MIN:	$z_{1,min}$
Z1MAX:	$z_{1,max}$
I2:	Surface number of $y = f_2(z)$
Z2MIN:	$z_{2,min}$
Z2MAX:	$z_{2,max}$
DZMIN:	Δz_{min}
Z1CLOS:	z_1 } P_1
Y1CLOS:	y_1 } P_1
$ALT:	Alternate return if no intersection

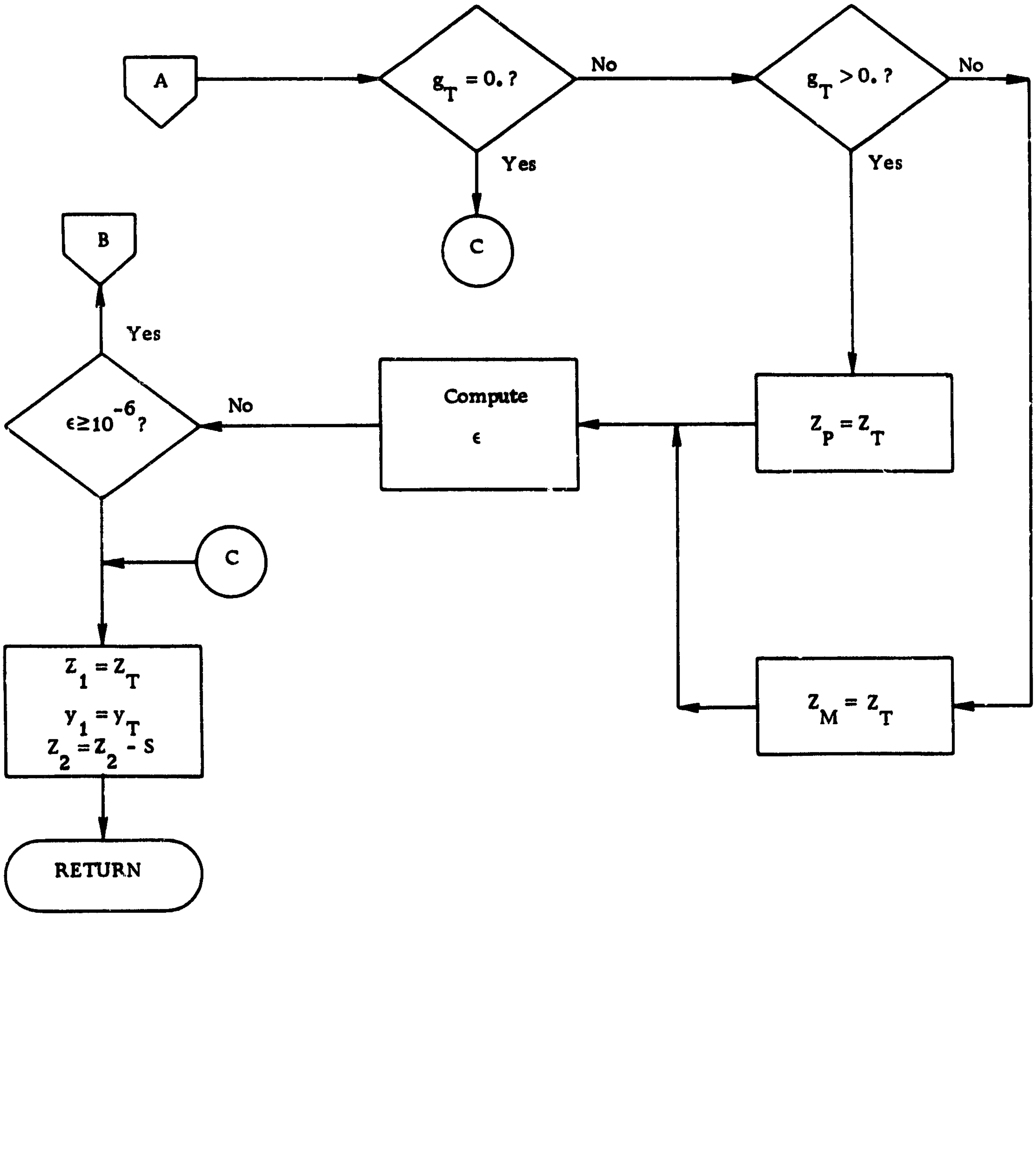
Utility Routines and Common References

COMMON/DATA/

COMMON/ELMAIR/

CROSS


SUBROUTINE CLOSS
$Z_1 = y_1 = 0.$
$Z_2 = 0.$
$I_1 = I_2$?
Yes
No
CLOSS Check for Intersect
Yes
No
Is "1" Vertical Cyl. ?
No
Yes
$R_1 = 0.$?
No
Yes
$Z_2 = \Delta Z_{min} - S$
$y_1 = f_2 (Z_2)$
RETURN
RETURN 1
Is "2" Vertical Cyl. ?
Yes
No
$R_2 = 0.$?
Yes
No
$Z_1 = S - \Delta Z_{min}$
$y_1 = f_1 (Z_1)$
$Z_P = 0.$
$Z_M = Z_I$
$Z_T = \frac{Z_P + Z_M}{2.}$
Compute
y_T, Z_2, g_T
B
A


A
$g_T = 0.?$
No
Yes
C
$g_T > 0.?$
No
Yes
B
Yes
$\epsilon \geq 10^{-6}?$
No
Compute ϵ
$Z_P = Z_T$
C
$Z_1 = Z_T$
$y_1 = y_T$
$Z_2 = Z_2 - S$
$Z_M = Z_T$
RETURN

3.2.4 CPLOTZ (CPLOT)

CPLOTZ 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

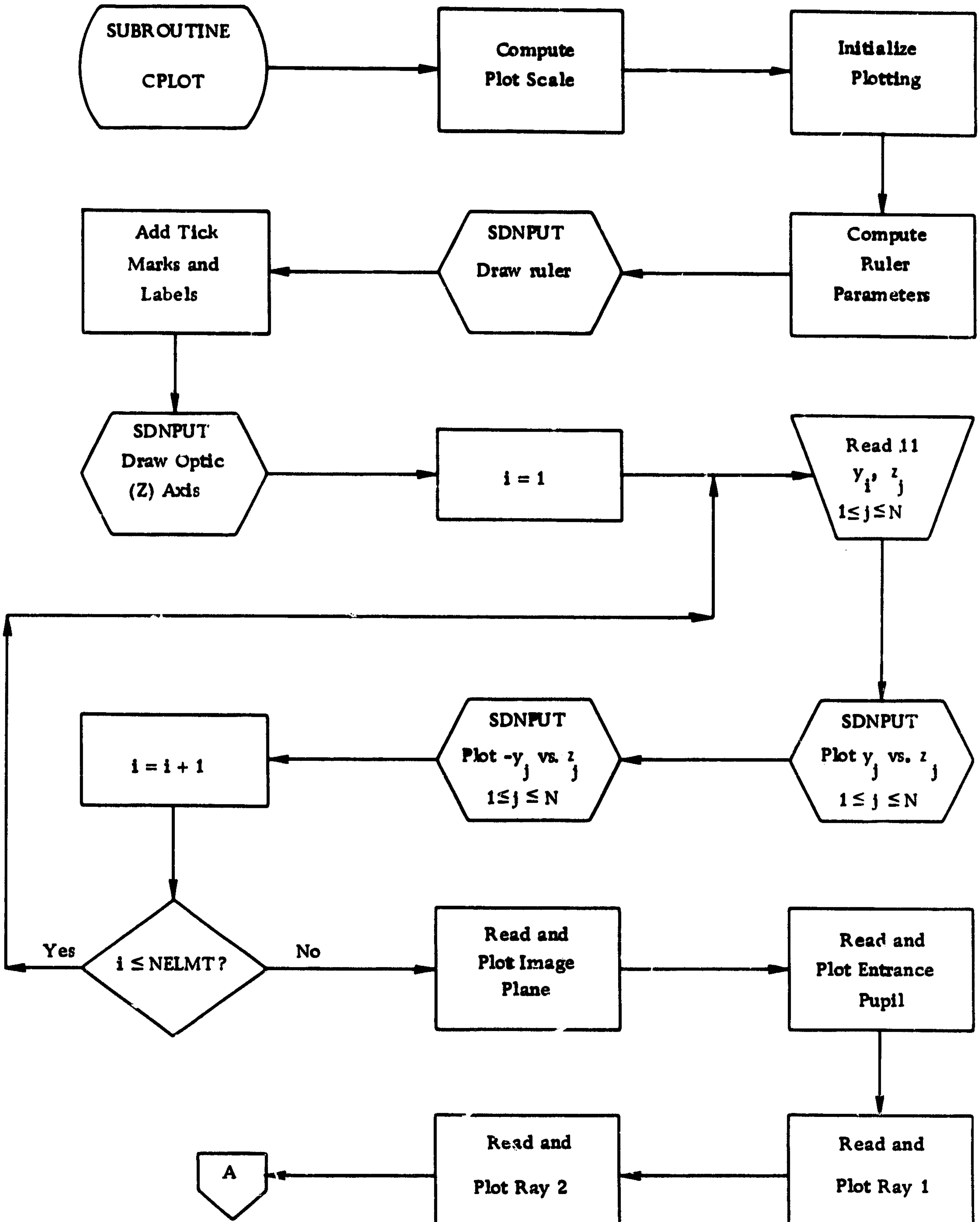
CALL CPLOT (ZMIN, ZMAX)

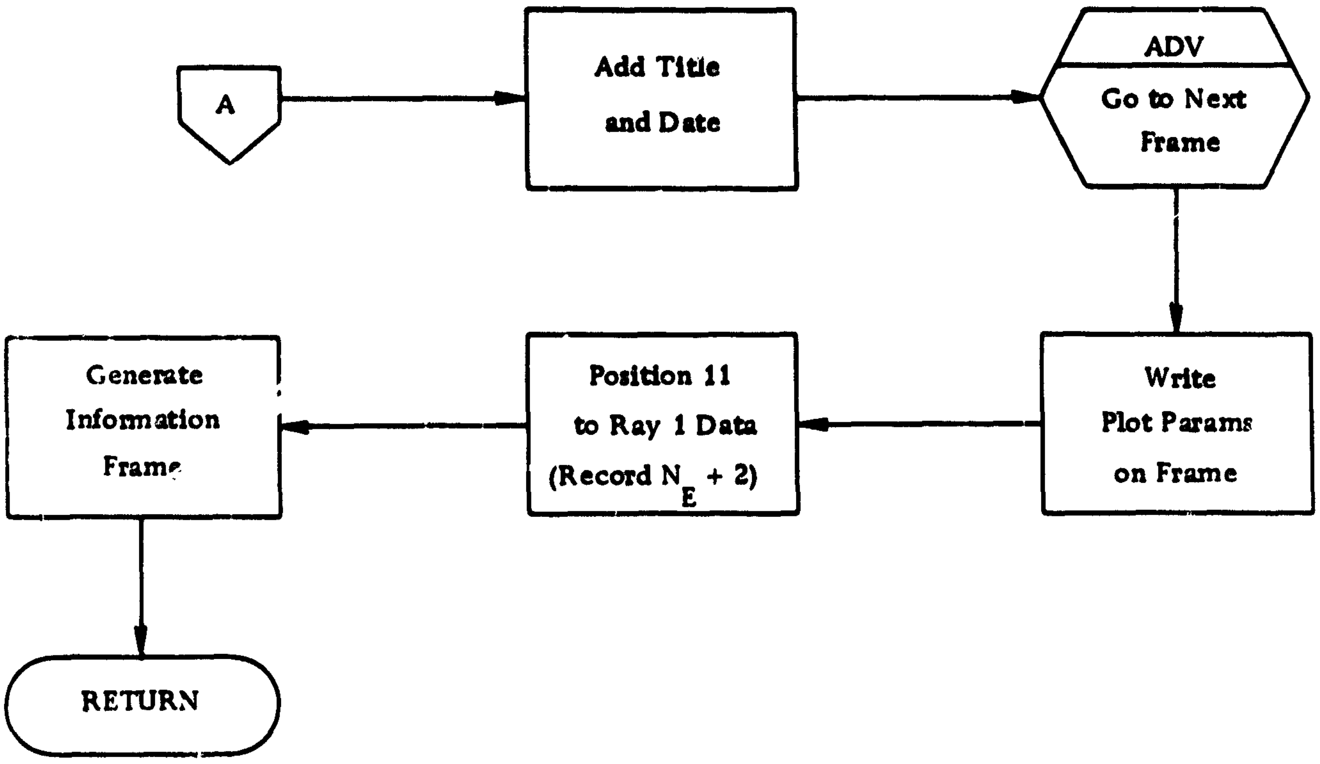
where ZMIN is the minimum value of z relative to the entrance pupil and ZMAX is the maximum value relative to the entrance pupil.

Utility Routines and Common References

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

* See Appendix A


SUBROUTINE CPLOT
Compute Plot Scale
Initialize Plotting
Compute Ruler Parameters
SDNPUT Draw ruler
Add Tick Marks and Labels
SDNPUT Draw Optic (Z) Axis
i = 1
Read 11 y_i, z_j $1 \le j \le N$
SDNPUT Plot y_j vs. z_j $1 \le j \le N$
SDNPUT Plot $-y_j$ vs. z_j $1 \le j \le N$
i = i + 1
$i \le$ NELMT?
Yes
No
Read and Plot Image Plane
Read and Plot Entrance Pupil
Read and Plot Ray 1
Read and Plot Ray 2
A


A
Add Title and Date
ADV
Go to Next Frame
Write Plot Params on Frame
Position 11 to Ray 1 Data (Record N_E + 2)
Generate Information Frame
RETURN

3.2.5 CROSSZ (CROSS)

Given $y = f_1(z)$ for $z \epsilon [z_{1,min}, z_{1,max}]$ relative to the vertex or surface 1 and $y = f_2(z)$ for $z \epsilon [z_{2,min}, z_{2,max}]$ 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

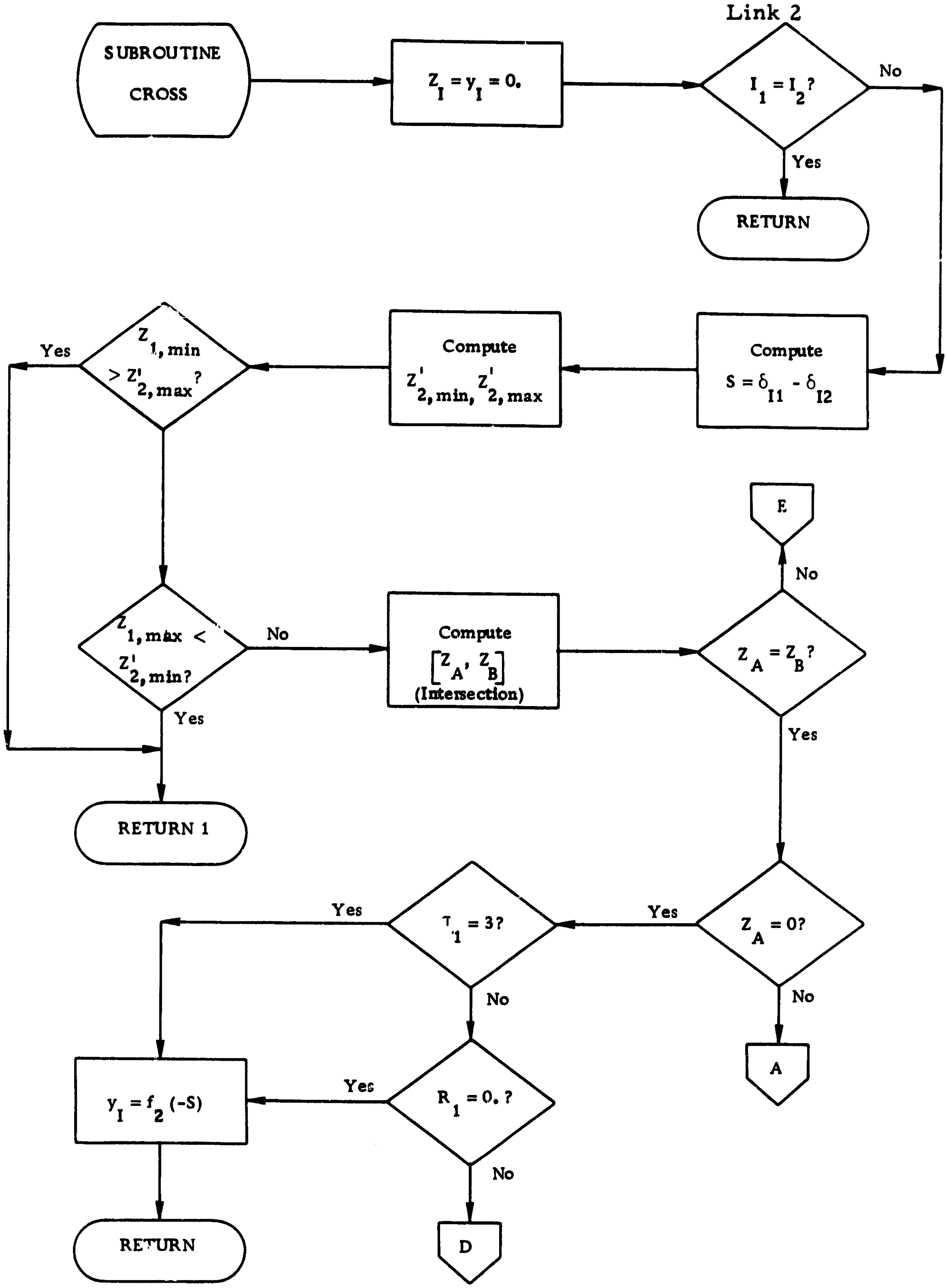
```
CALL   CROSS(I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX,
       ZCROSS, YCROSS, $ALT)
```

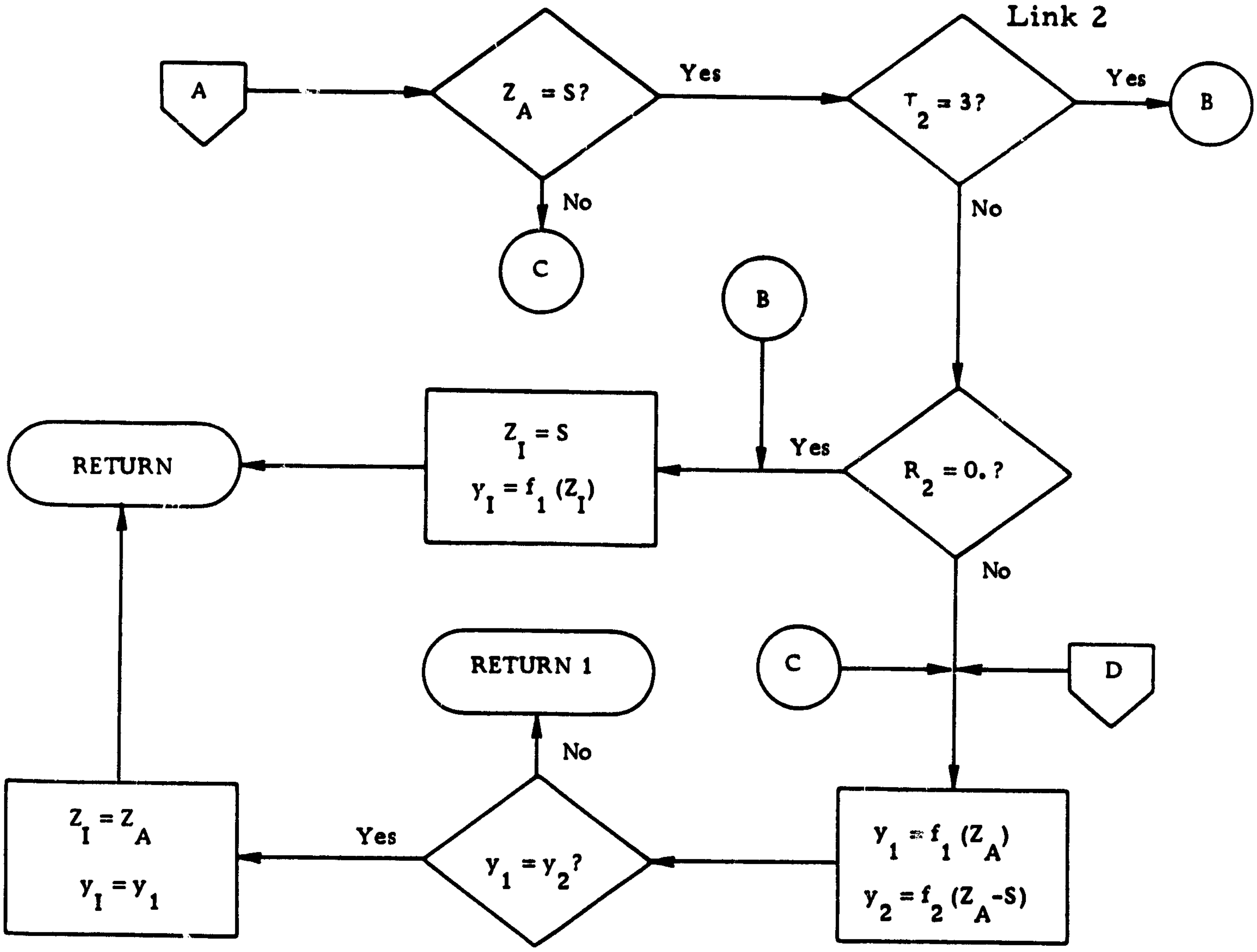
I1:	Surface number of $y = f_1(z)$
Z1MIN:	$z_{1,min}$
Z1MAX:	$z_{1,max}$
I2:	Surface number of $y = f_2(z)$
Z2MIN:	$z_{2,min}$
Z2MAX:	$z_{2,max}$
ZCROSS:	z_I
YCROSS:	y_I
$ALT:	Alternate return if no intersection

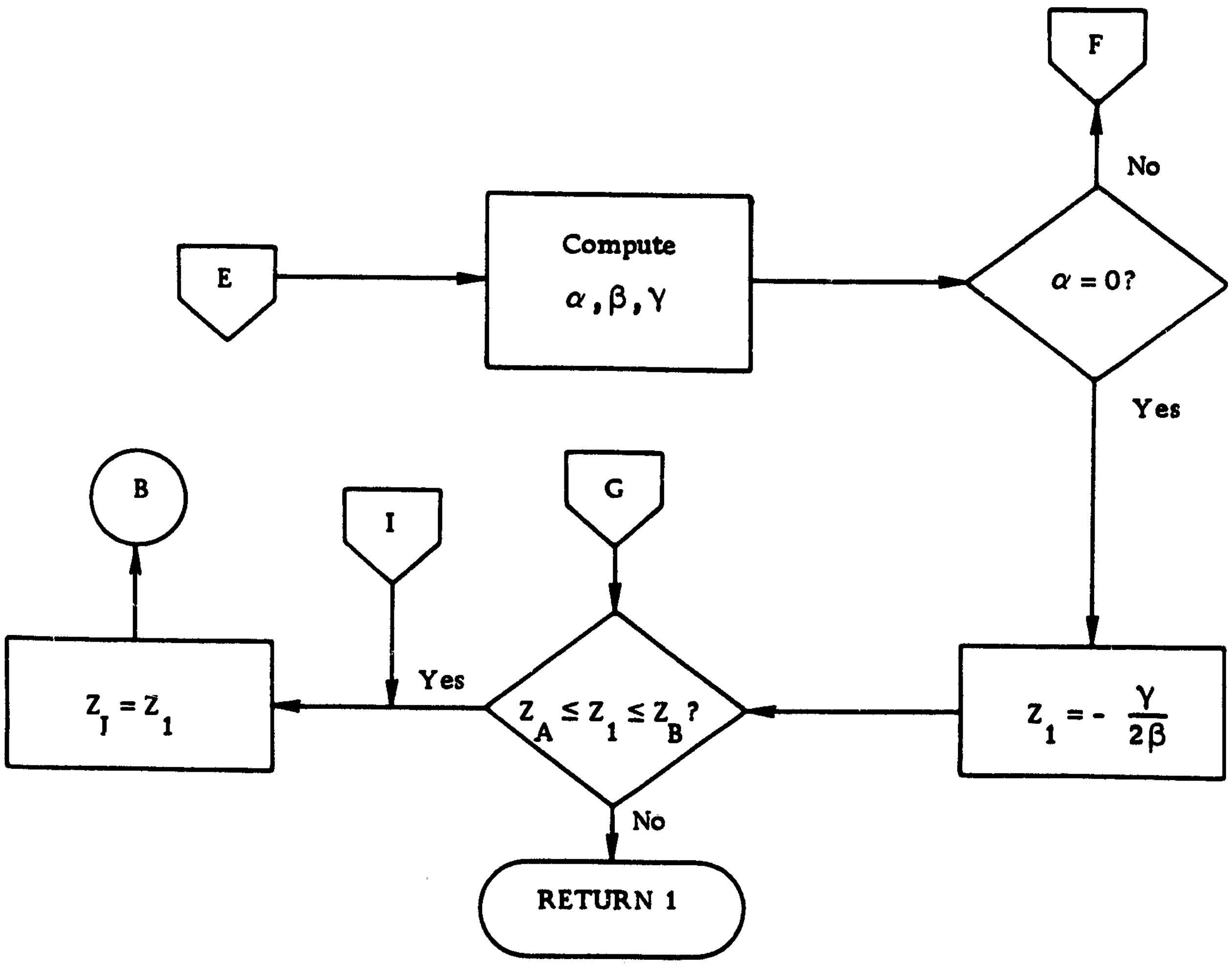
Utility Routines and Common References

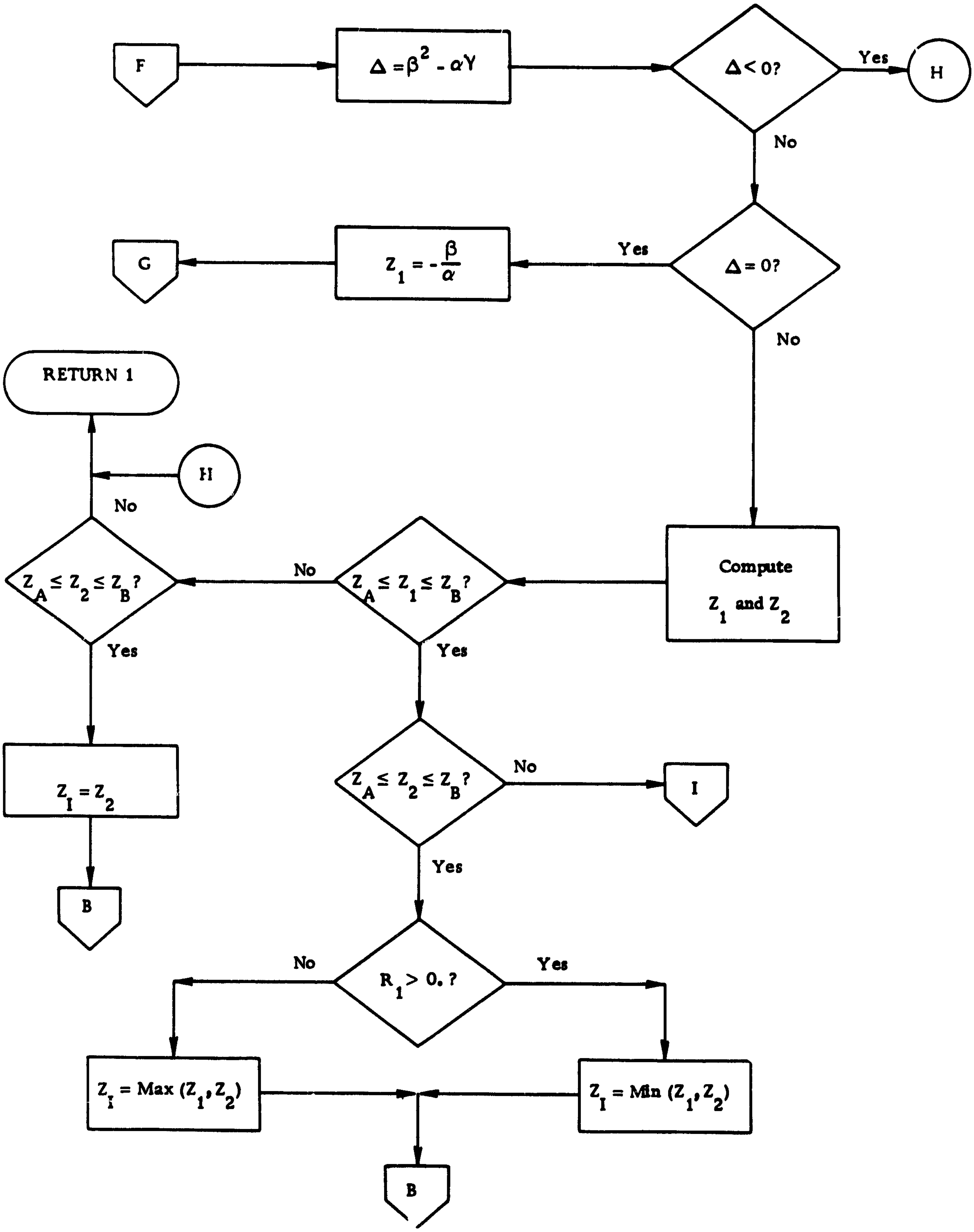
COMMON/DATA/

COMMON/ELMAIR/


SUBROUTINE CROSS
$Z_I = y_I = 0.$
$I_1 = I_2$?
No
Yes
RETURN
Compute
$S = \delta_{I1} - \delta_{I2}$
Compute
$Z'_{2,min}, Z'_{2,max}$
$Z_{1,min} > Z'_{2,max}$?
Yes
$Z_{1,max} < Z'_{2,min}$?
No
Yes
RETURN 1
Compute
$[Z_A, Z_B]$
(Intersection)
$Z_A = Z_B$?
No
E
Yes
$Z_A = 0$?
Yes
No
A
$T_1 = 3$?
Yes
No
$R_1 = 0.$?
Yes
No
D
$y_I = f_2(-S)$
RETURN


A
$Z_A = S$?
Yes
No
C
$T_2 = 3$?
Yes
B
No
B
$Z_I = S$
$y_I = f_1(Z_I)$
Yes
$R_2 = 0.$?
No
RETURN
C
D
RETURN 1
No
$Z_I = Z_A$
$y_I = y_1$
Yes
$y_1 = y_2$?
$y_1 = f_1(Z_A)$
$y_2 = f_2(Z_A - S)$


F
No
E
Compute
α, β, γ
$\alpha = 0$?
Yes
B
I
G
$Z_J = Z_1$
Yes
$Z_A \leq Z_1 \leq Z_B$?
$Z_1 = -\frac{\gamma}{2\beta}$
No
RETURN 1


F
Δ = β² - αγ
Δ < 0?
Yes
H
No
Δ = 0?
Yes
Z₁ = -β/α
G
No
RETURN 1
H
No
Compute Z₁ and Z₂
Z_A ≤ Z₁ ≤ Z_B ?
No
Z_A ≤ Z₂ ≤ Z_B ?
Yes
Z_I = Z₂
B
Yes
Z_A ≤ Z₂ ≤ Z_B ?
No
I
Yes
R₁ > 0. ?
No
Yes
Z_I = Max (Z₁, Z₂)
Z_I = Min (Z₁, Z₂)
B

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:

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

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.

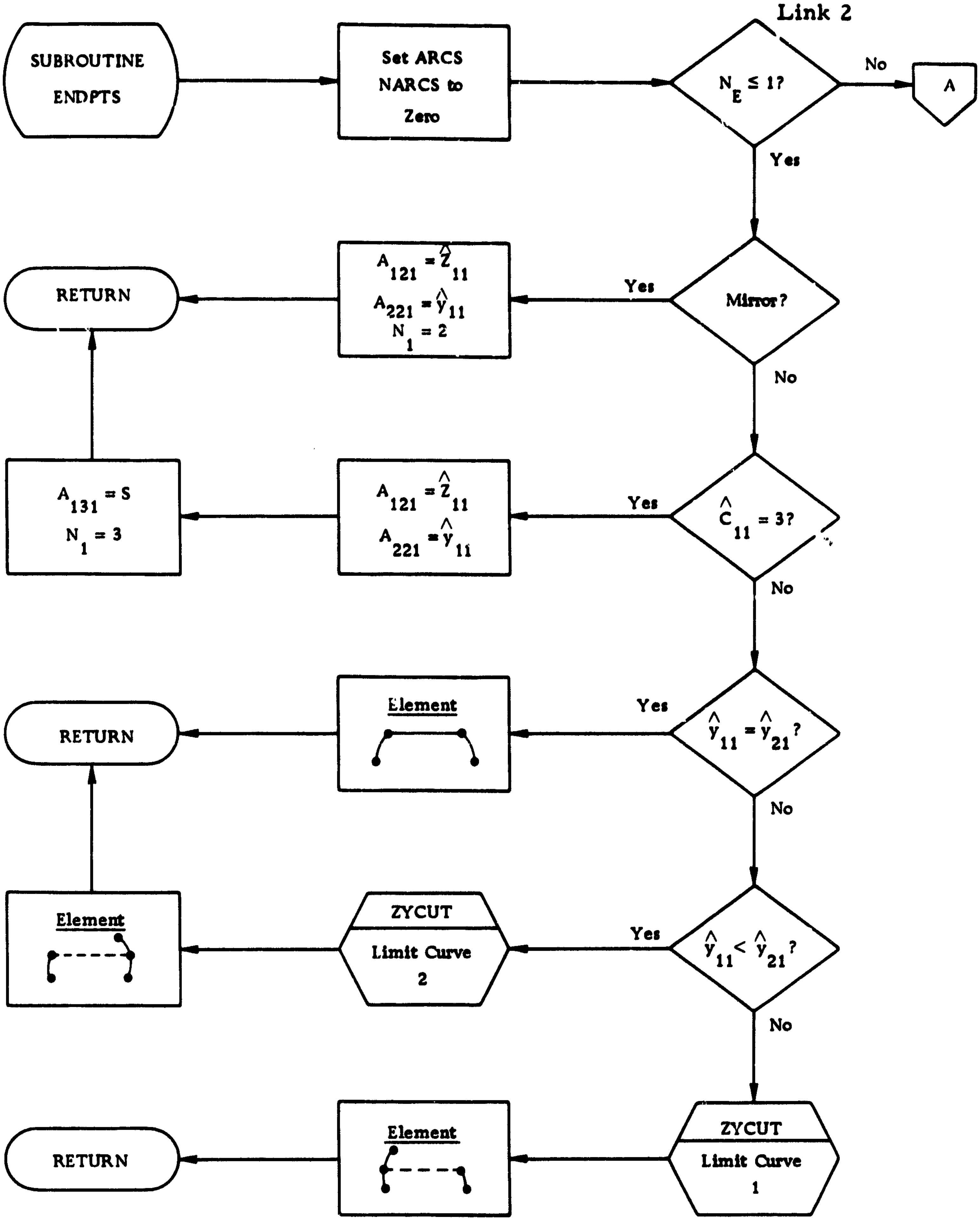
Calling Sequence

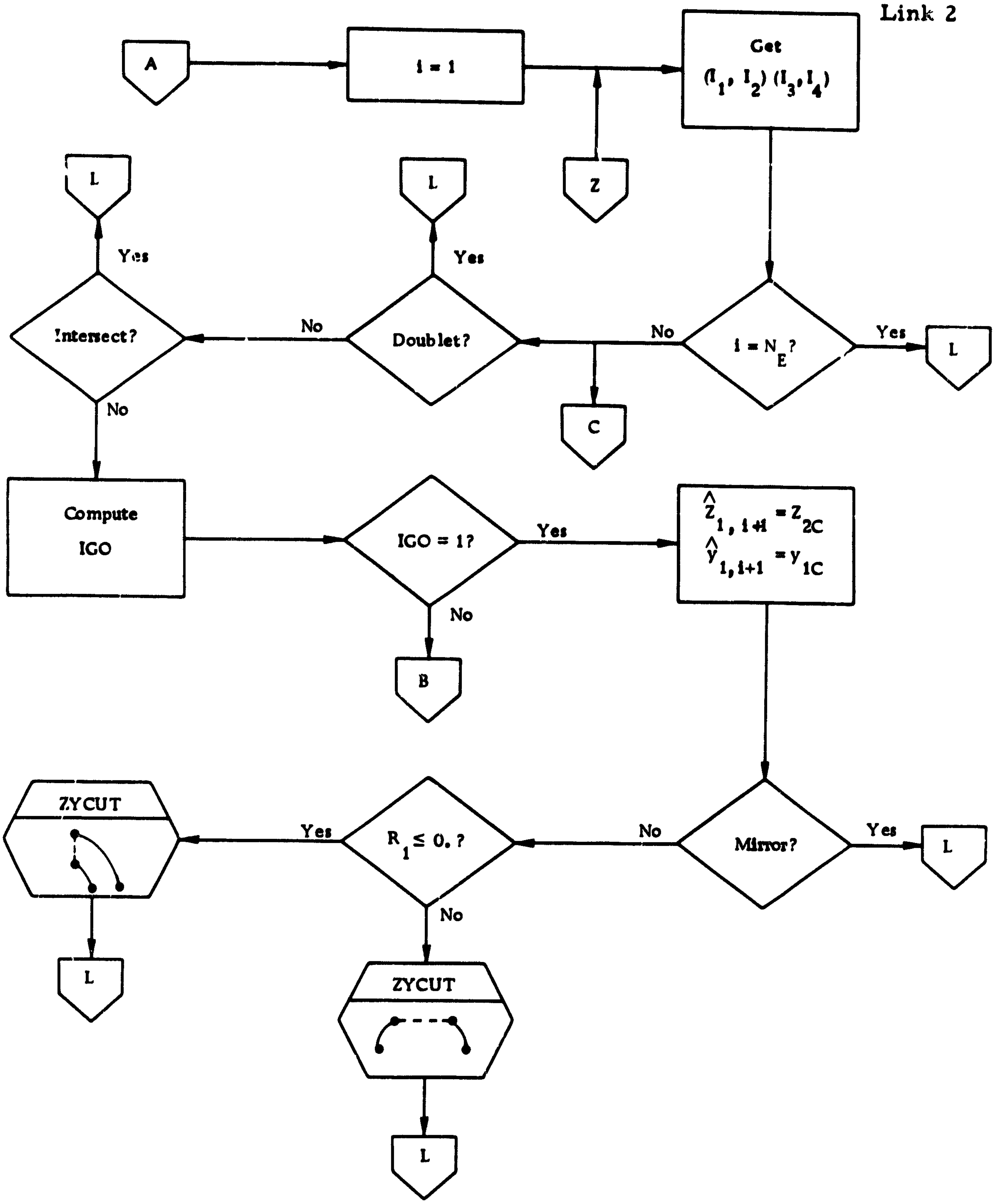
CALL ENDPTS (DZMIN, ARCS, NARCS)

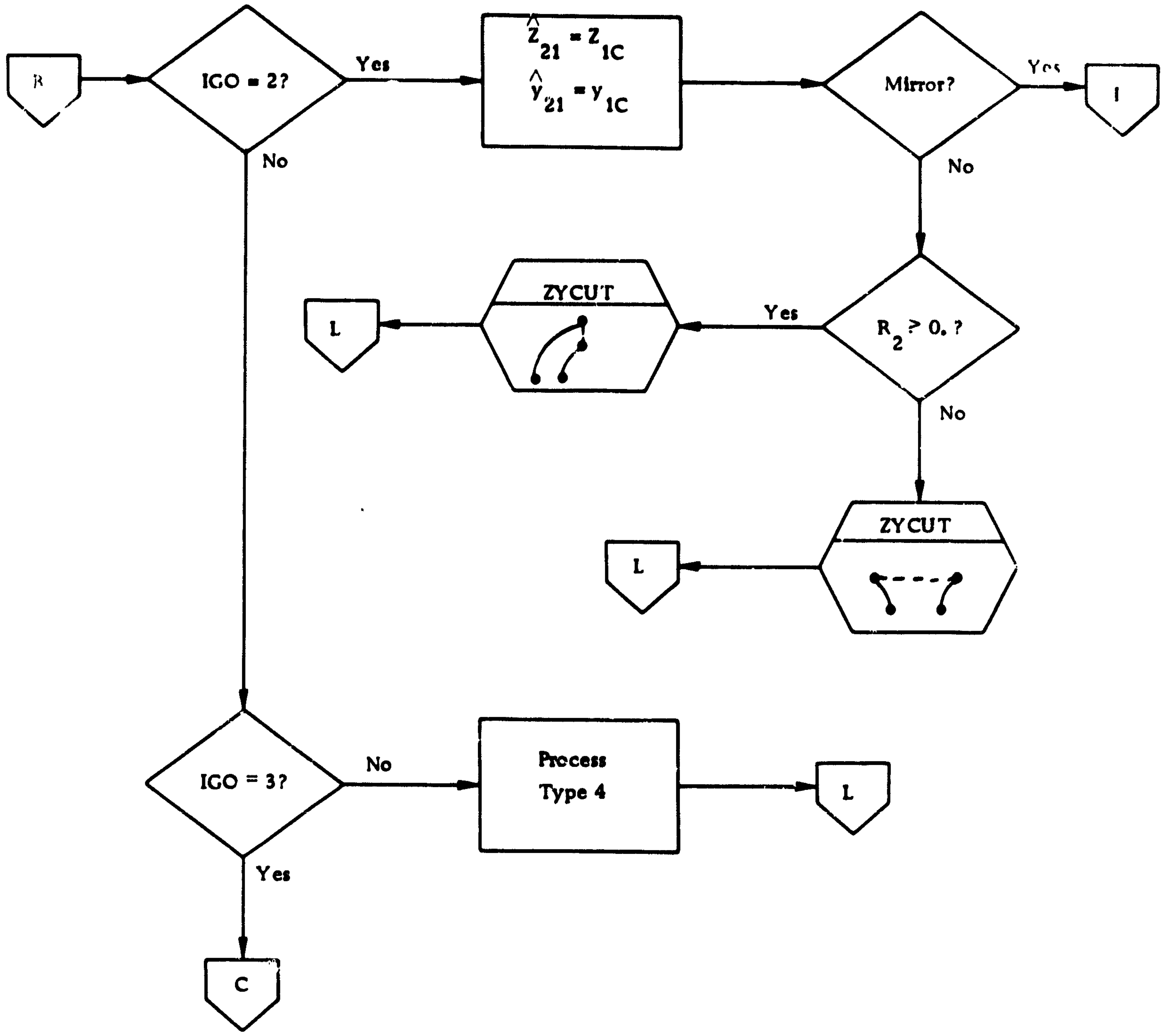
where DZMIN is the minimum permissible air space thickness, ARCS is the 2 x 5 x 100 array described above, and NARCS is a vector containing the arc count for each element.

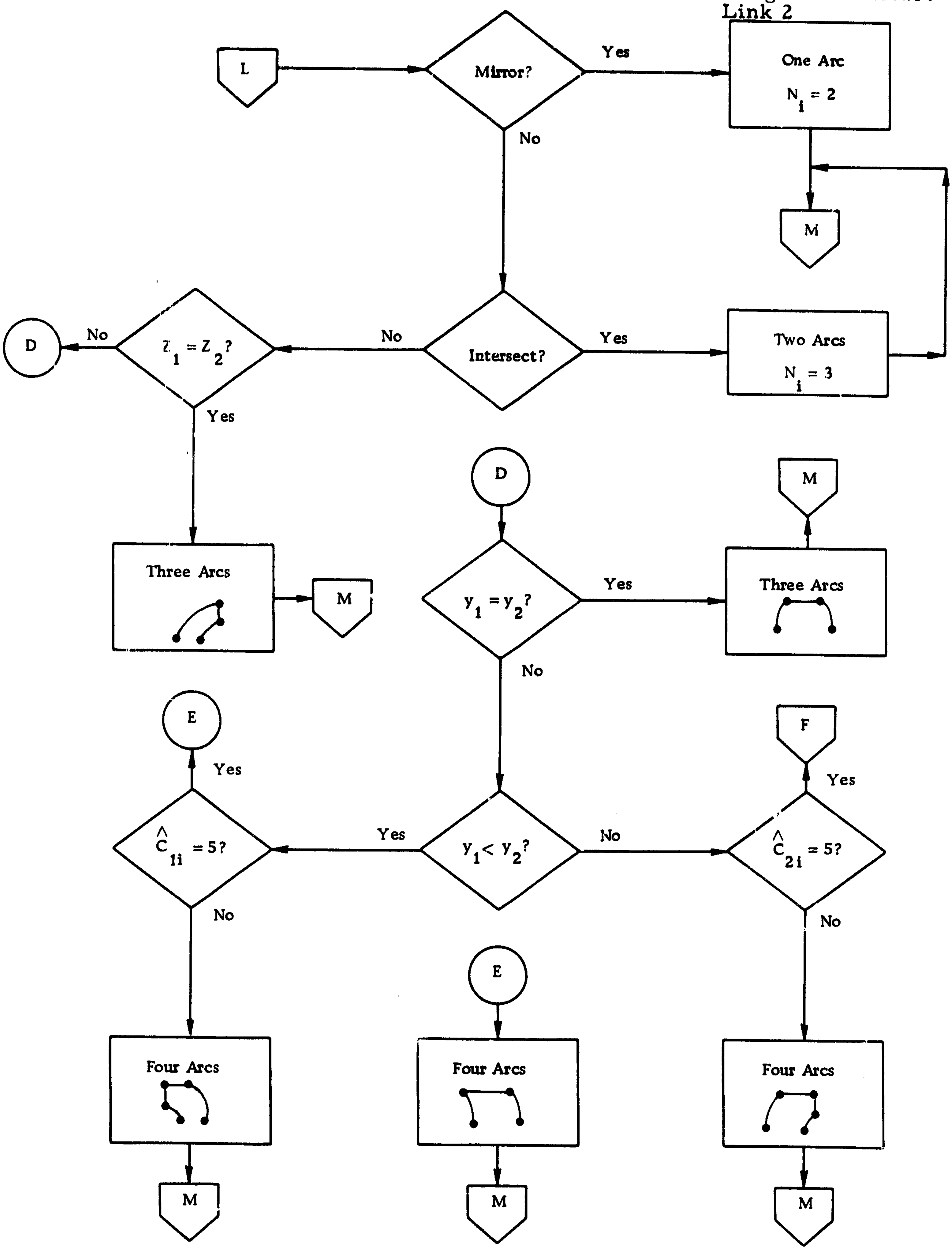
Utility Routines and Common References

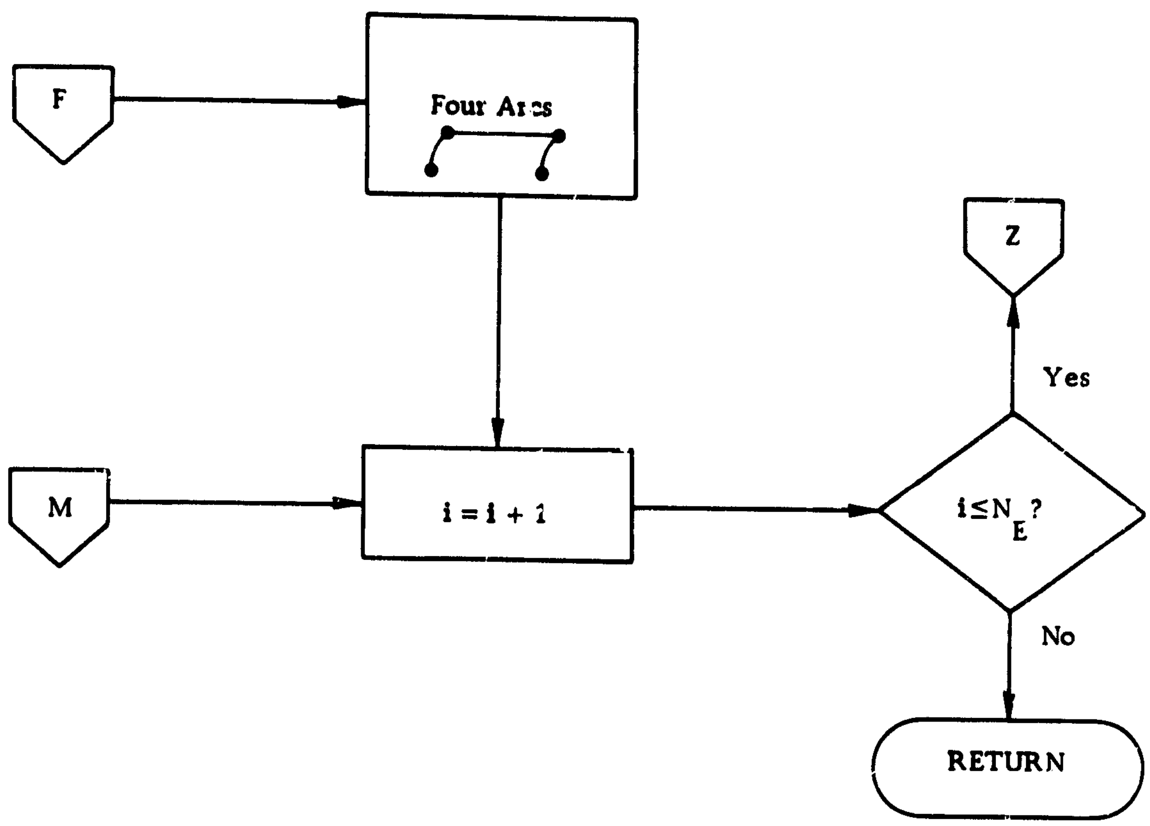
COMMON/DATA
COMMON/ELMAIR/
COMMON/CODRNG/
ZYCUT
CLOSS
GOOF


SUBROUTINE ENDPTS
Set ARCS NARCS to Zero
$N_E \leq 1$?
No
A
Yes
Mirror?
Yes
$A_{121} = \hat{Z}_{11}$
$A_{221} = \hat{y}_{11}$
$N_1 = 2$
RETURN
No
$\hat{C}_{11} = 3$?
Yes
$A_{121} = \hat{Z}_{11}$
$A_{221} = \hat{y}_{11}$
$A_{131} = S$
$N_1 = 3$
No
$\hat{y}_{11} = \hat{y}_{21}$?
Yes
Element
RETURN
No
$\hat{y}_{11} < \hat{y}_{21}$?
Yes
ZYCUT
Limit Curve 2
Element
No
ZYCUT
Limit Curve 1
Element
RETURN


A
i = 1
Get
(I_1, I_2) (I_3, I_4)
Z
L
Yes
Intersect?
No
L
Yes
Doublet?
No
$i = N_E$?
Yes
L
C
No
Compute
IGO
IGO = 1?
Yes
No
B
$\hat{z}_{1,i+1} = z_{2C}$
$\hat{y}_{1,i+1} = y_{1C}$
ZYCUT
Yes
$R_1 \leq 0.$?
No
Mirror?
Yes
L
No
L
ZYCUT
L


R
IGO = 2?
Yes
No
$\hat{Z}_{21} = Z_{1C}$
$\hat{y}_{21} = y_{1C}$
Mirror?
Yes
No
I
$R_2 \geq 0.$?
Yes
No
ZYCUT
L
ZYCUT
L
IGO = 3?
No
Yes
Process
Type 4
L
C


L
Mirror?
Yes
No
One Arc
$N_i = 2$
M
D
No
$Z_1 = Z_2$?
No
Intersect?
Yes
Two Arcs
$N_i = 3$
Yes
D
M
Three Arcs
M
$y_1 = y_2$?
Yes
Three Arcs
No
E
F
Yes
Yes
$\hat{C}_{1i} = 5$?
Yes
$y_1 < y_2$?
No
$\hat{C}_{2i} = 5$?
No
No
E
Four Arcs
Four Arcs
Four Arcs
M
M
M


F
Four Arcs
M
i = i + 1
Z
Yes
$i \leq N_E$?
No
RETURN

3.2.7 E1RGCZ (E1RGCD)

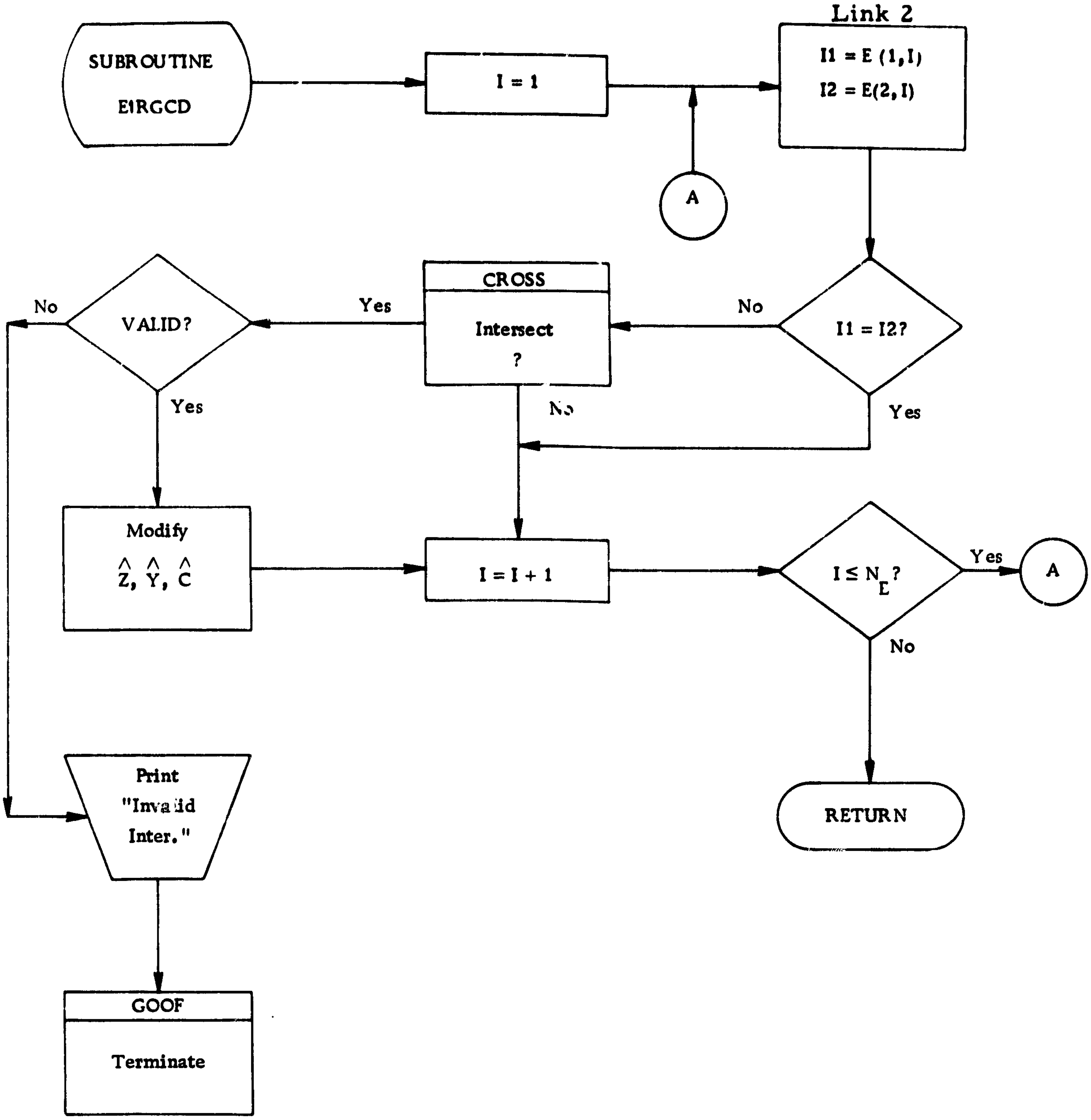
E1RGCZ employs subroutine CROSS to edit the range and code matrices by subjecting each element to intersection constraints.

Calling Sequence

CALL E1RGCD

Utility Routines and Common References

COMMON/ELMAIR/
COMMON/CODRNG/
CROSS
GOOF


SUBROUTINE
E1RGCD
I = 1
I1 = E (1,I)
I2 = E(2,I)
A
I1 = I2?
No
Yes
CROSS
Intersect
?
Yes
No
VALID?
No
Yes
Modify
$\hat{Z}$, $\hat{Y}$, $\hat{C}$
I = I + 1
I ≤ N_E?
Yes
A
No
RETURN
Print
"Invalid
Inter."
GOOF
Terminate

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 ith 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/

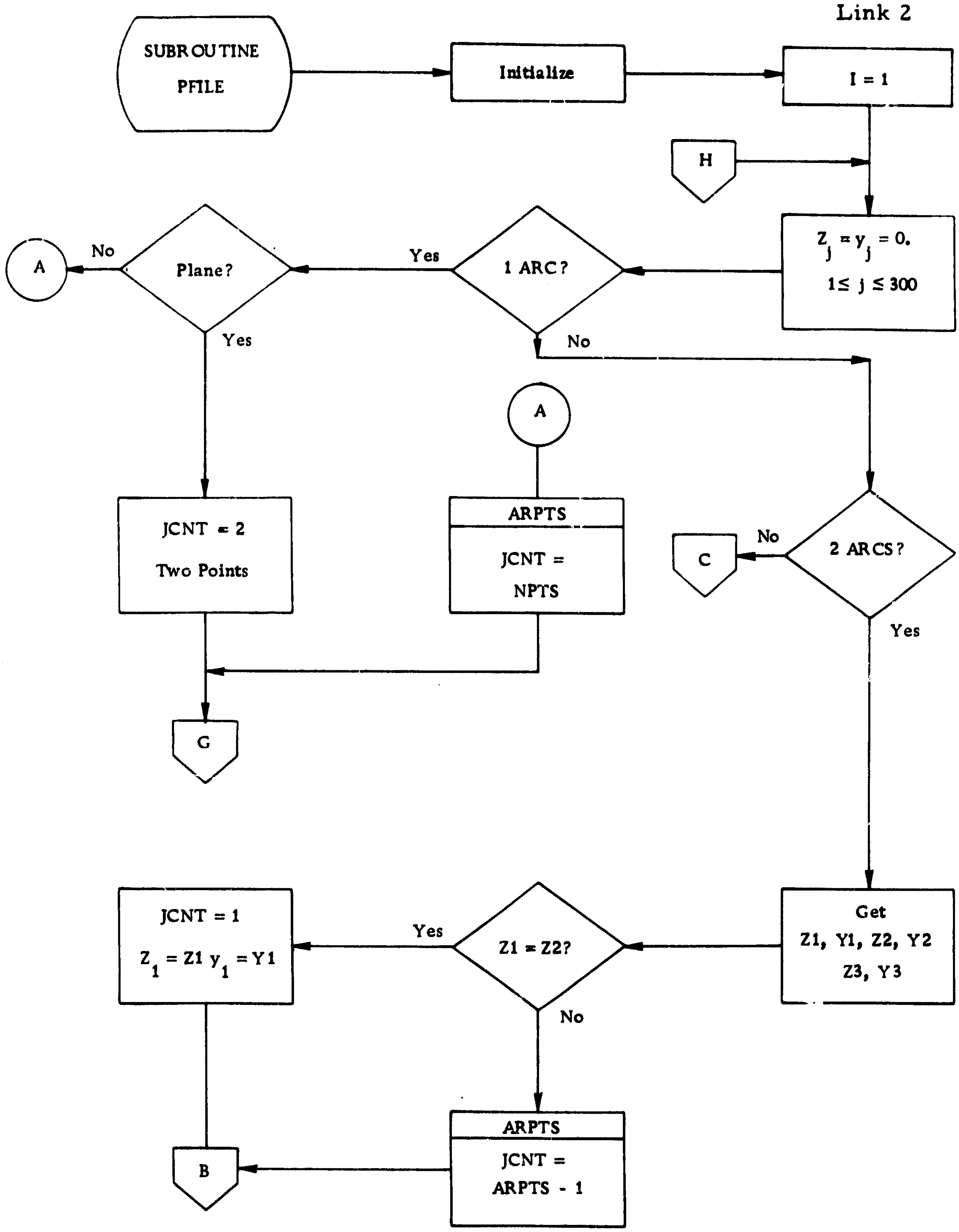
COMMON/CODRNG/

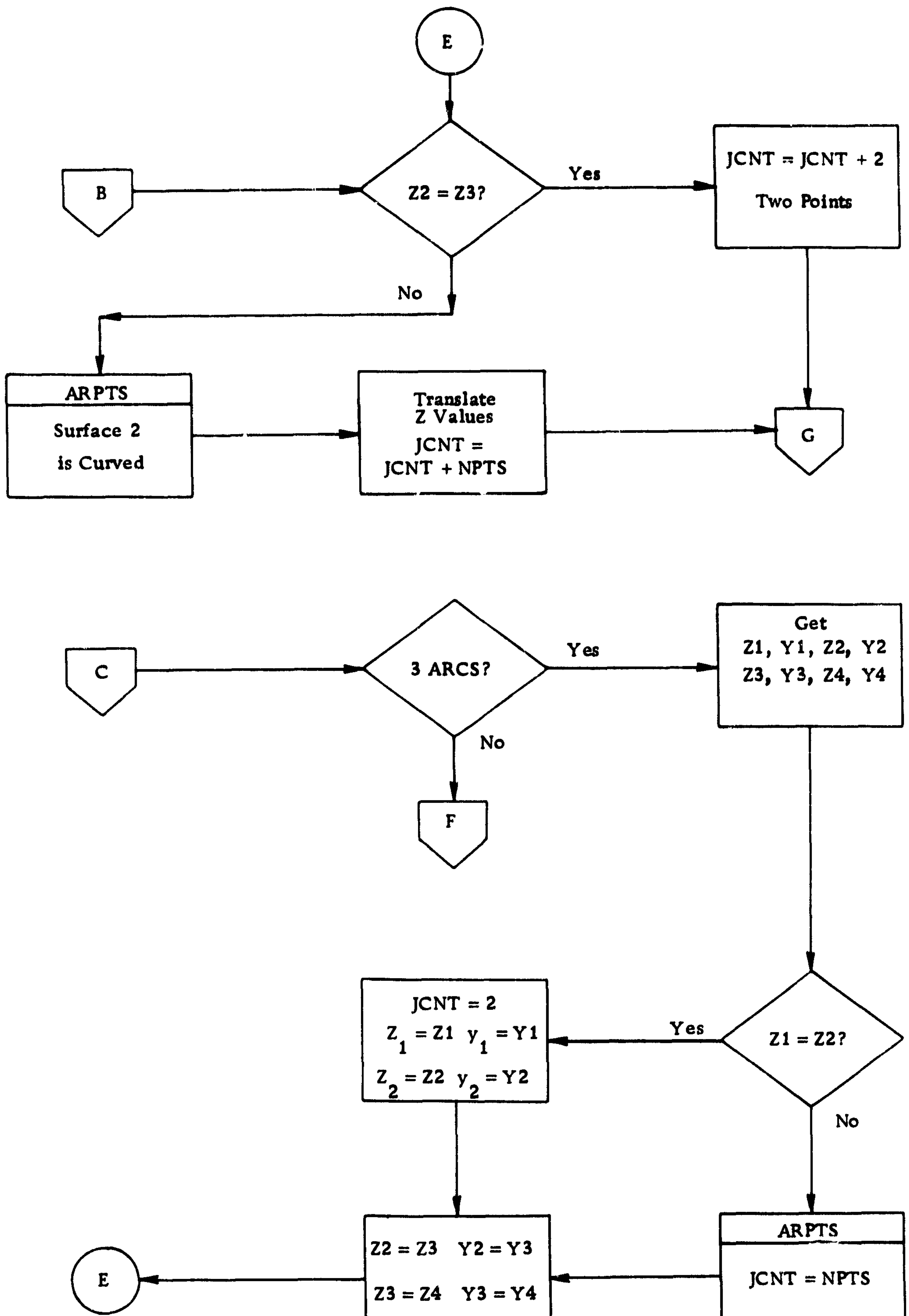
COMMON/PLOTC/

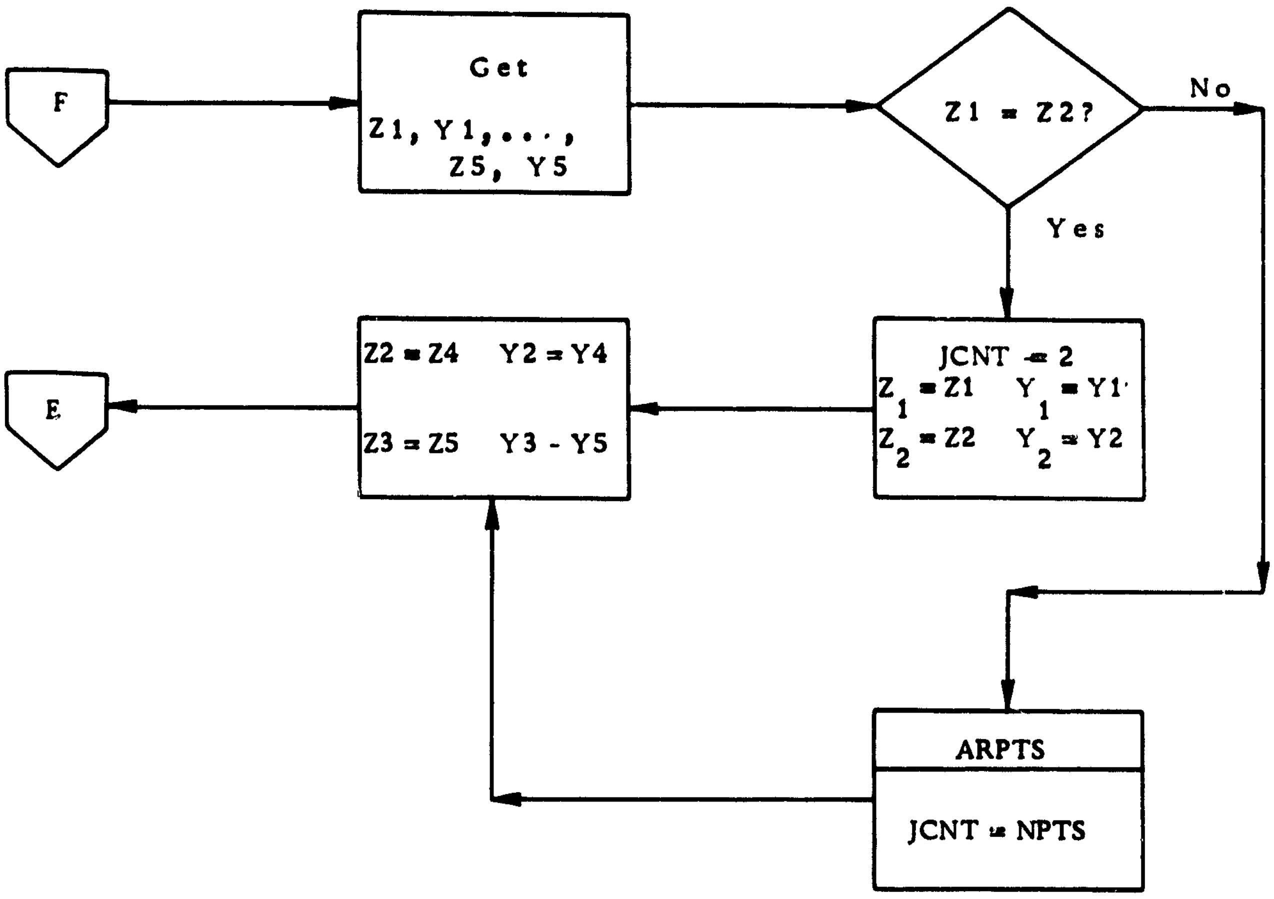
COMMON/AZOBJ/

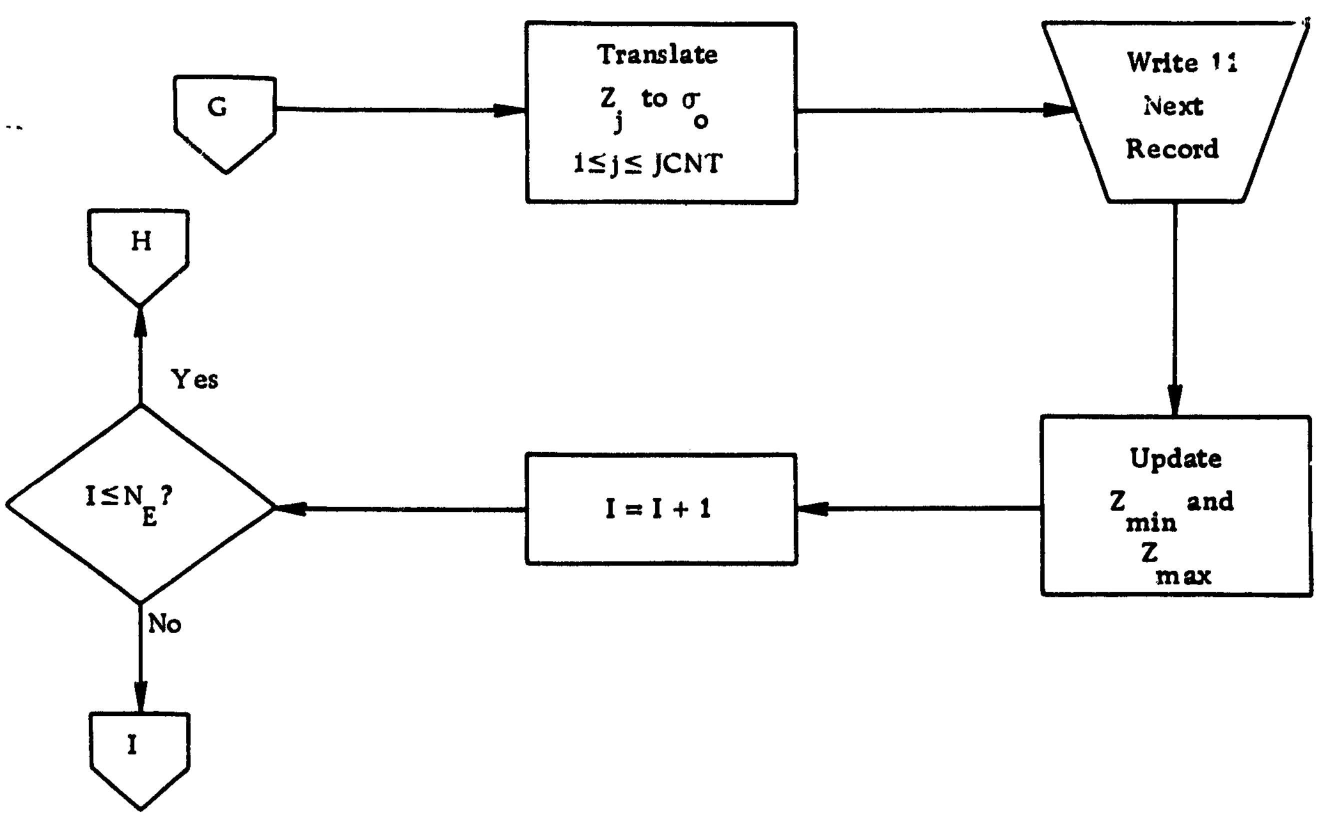
ARPTS

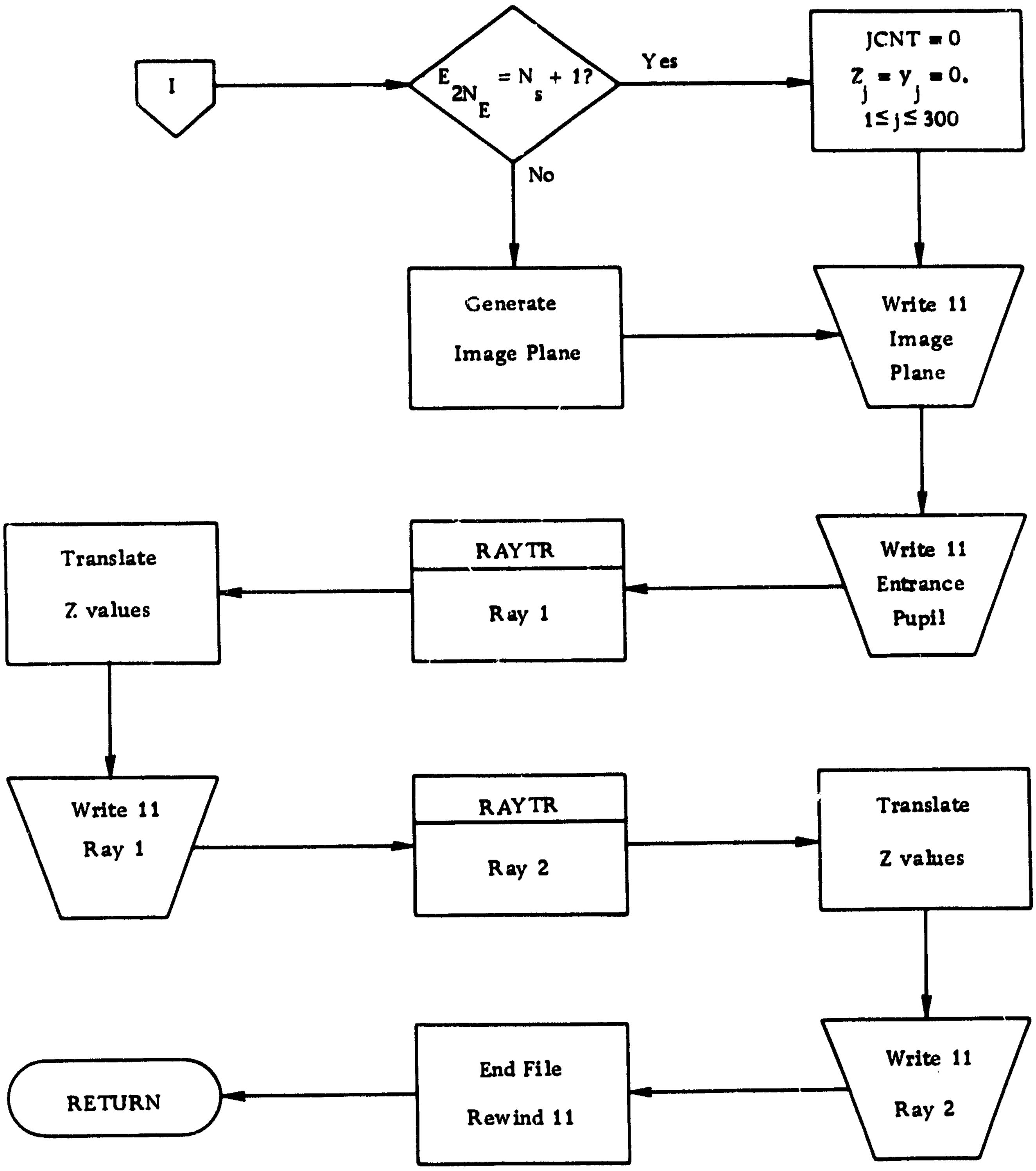
RAYTR


SUBROUTINE PFILE
Initialize
I = 1
H
Z_j = y_j = 0.
1≤ j ≤ 300
1 ARC?
Yes
No
Plane?
No
Yes
A
A
JCNT = 2
Two Points
ARPTS
JCNT = NPTS
G
2 ARCS?
No
C
Yes
Get
Z1, Y1, Z2, Y2
Z3, Y3
Z1 = Z2?
Yes
No
JCNT = 1
Z_1 = Z1 y_1 = Y1
ARPTS
JCNT = ARPTS - 1
B


E
B
Z2 = Z3?
Yes
JCNT = JCNT + 2
Two Points
No
ARPTS
Surface 2
is Curved
Translate
Z Values
JCNT =
JCNT + NPTS
G
C
3 ARCS?
Yes
Get
Z1, Y1, Z2, Y2
Z3, Y3, Z4, Y4
No
F
JCNT = 2
Z_1 = Z1 y_1 = Y1
Z_2 = Z2 y_2 = Y2
Yes
Z1 = Z2?
No
Z2 = Z3 Y2 = Y3
Z3 = Z4 Y3 = Y4
ARPTS
JCNT = NPTS
E


F
Get
Z1, Y1, . . . ,
Z5, Y5
Z1 = Z2?
No
Yes
JCNT = 2
Z_1 = Z1 Y_1 = Y1
Z_2 = Z2 Y_2 = Y2
Z2 = Z4   Y2 = Y4
Z3 = Z5   Y3 - Y5
E
ARPTS
JCNT = NPTS


G
Translate
Z_j to σ_o
1≤j≤ JCNT
Write 11
Next
Record
Update
Z_{min} and
Z_{max}
I = I + 1
I≤N_E?
Yes
H
No
I


I
$E_{2N_E} = N_s + 1$?
Yes
No
JCNT = 0
$Z_j = y_j = 0.$
$1 \le j \le 300$
Generate
Image Plane
Write 11
Image
Plane
Write 11
Entrance
Pupil
RAYTR
Ray 1
Translate
Z values
Write 11
Ray 1
RAYTR
Ray 2
Translate
Z values
Write 11
Ray 2
End File
Rewind 11
RETURN

3.2.9 PROFLZ (PROFIL)

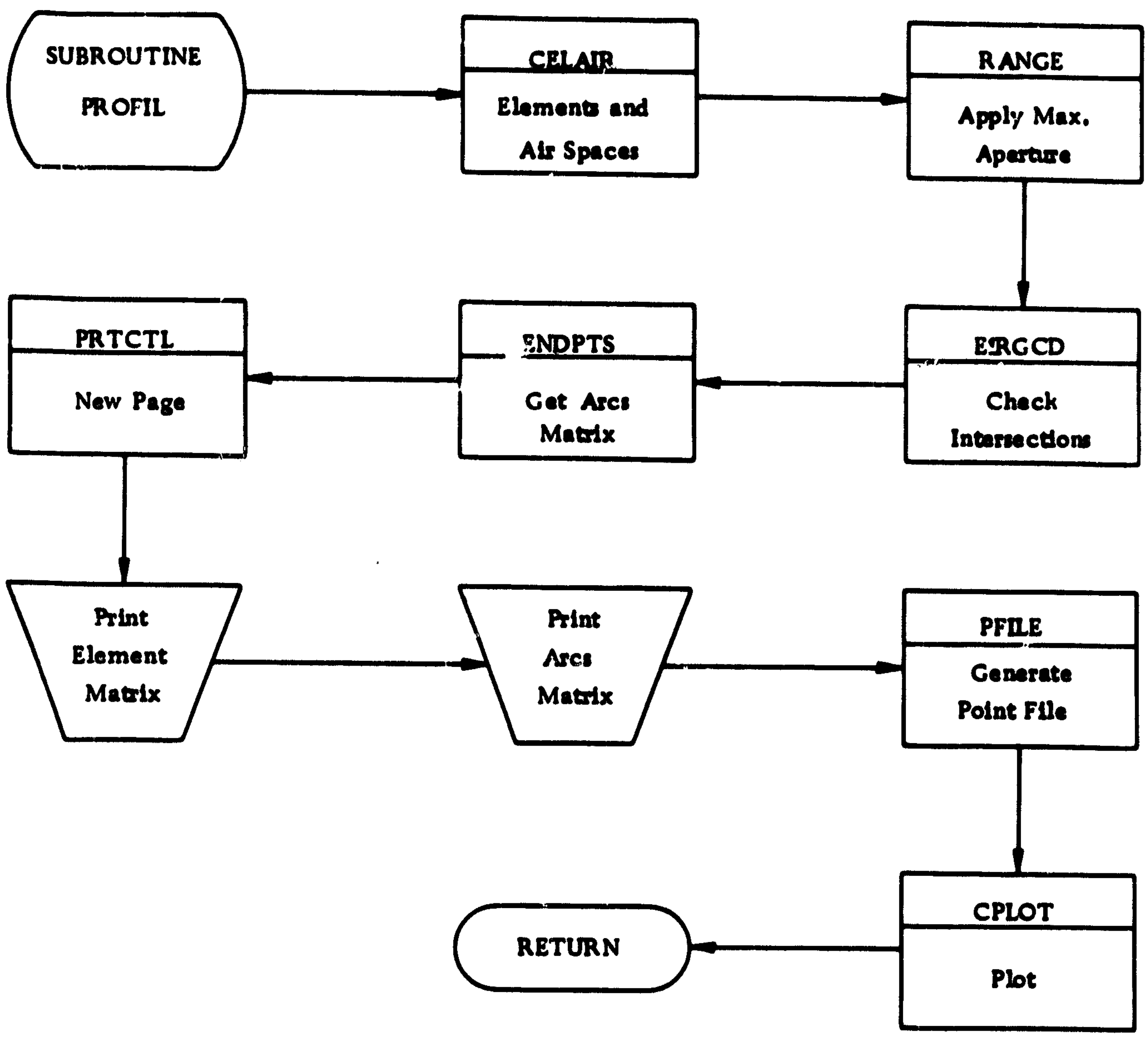
PROFLZ is the main or control routine of the profile plot link. It is called by the main program (FOLDP) whenever a profile plot is to be generated.

Calling Sequence

CALL PROFIL

Utility Routines and Common References

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


SUBROUTINE
PROFIL
CELAIR
Elements and
Air Spaces
RANGE
Apply Max.
Aperture
ERGCD
Check
Intersections
ENDPTS
Get Arcs
Matrix
PRTCTL
New Page
Print
Element
Matrix
Print
Arcs
Matrix
PFILE
Generate
Point File
CPLOT
Plot
RETURN

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.

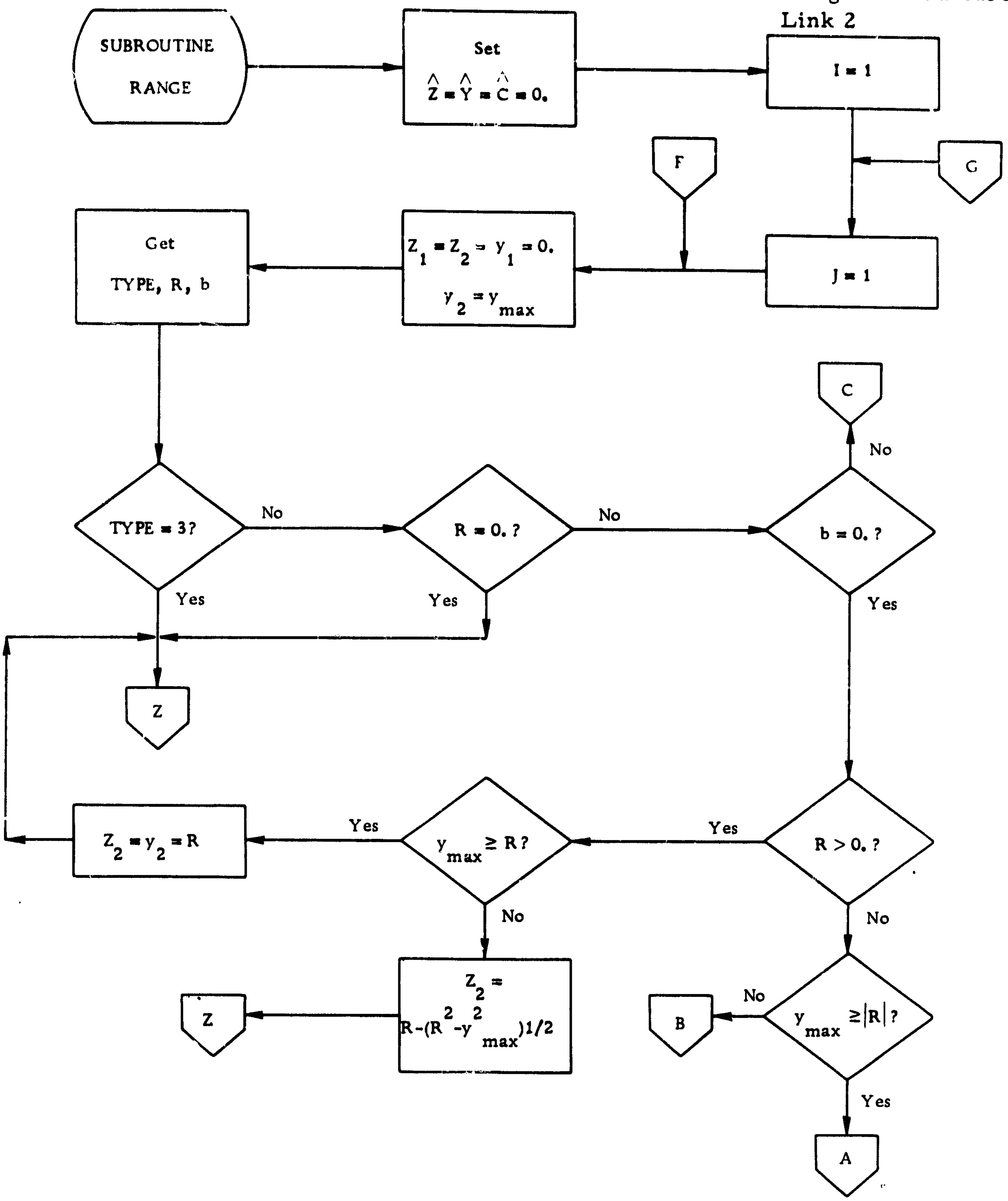
Calling Sequence

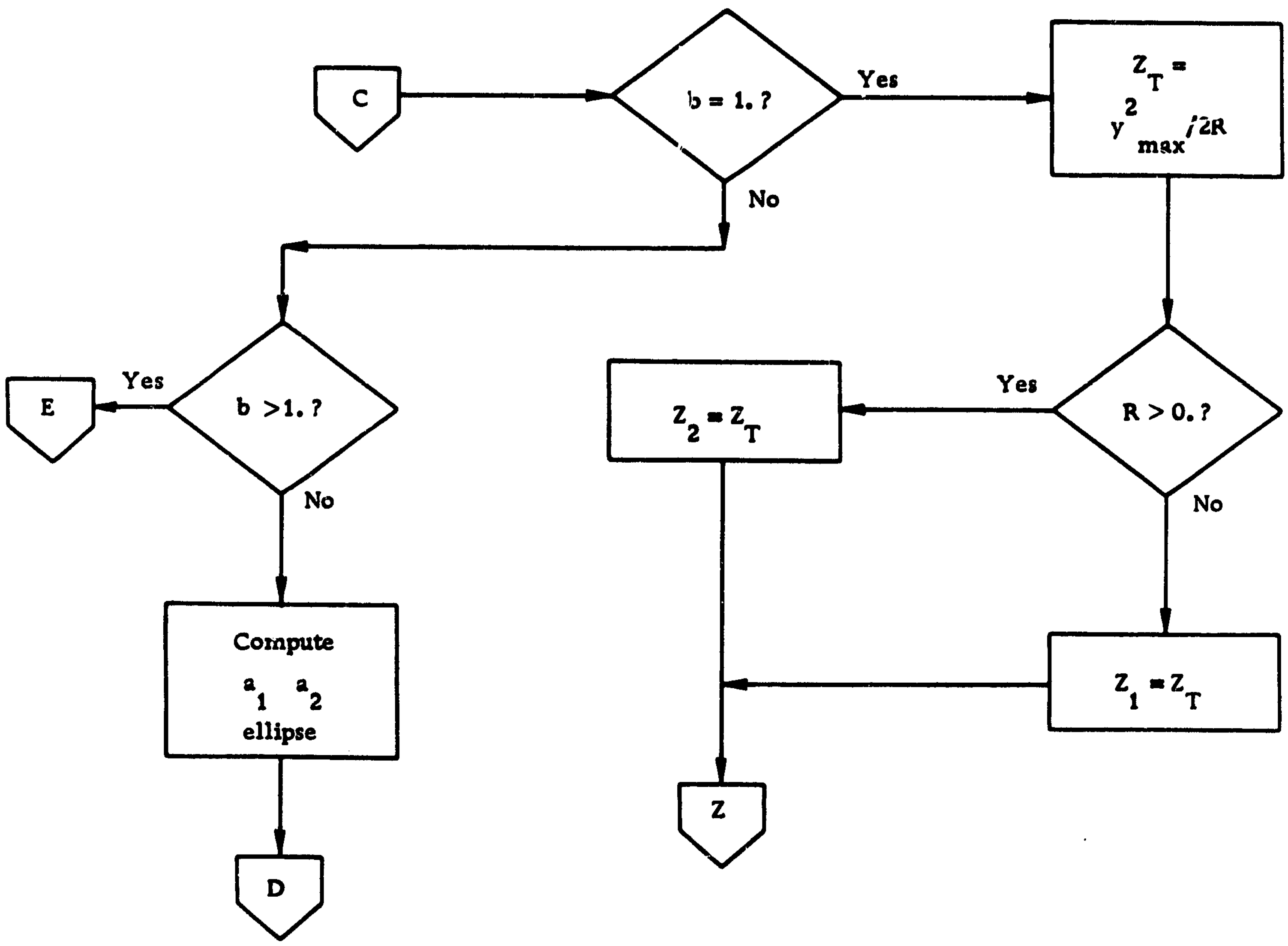
CALL RANGE (YMAXIM)

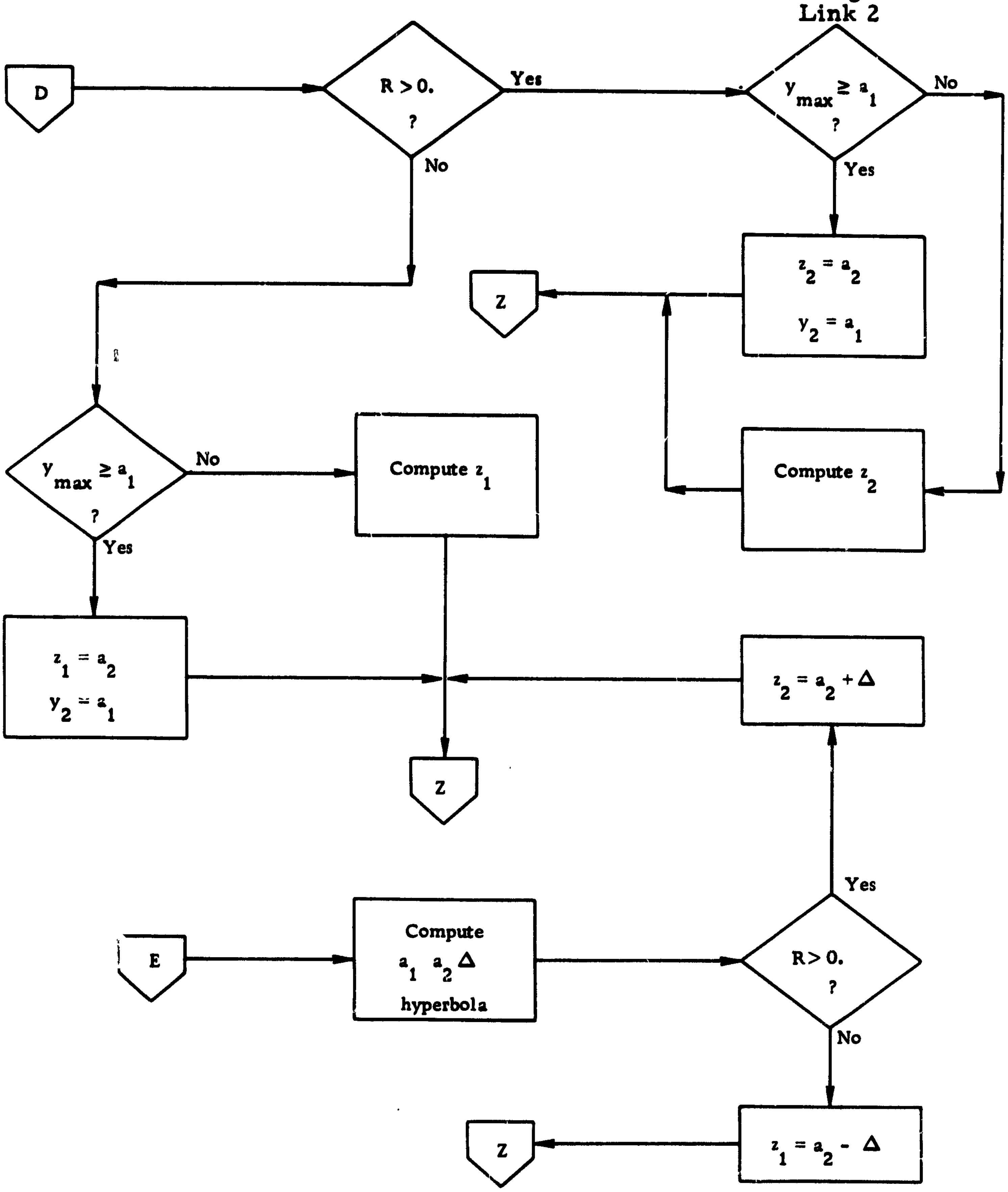
YMAXIM: y_{max}

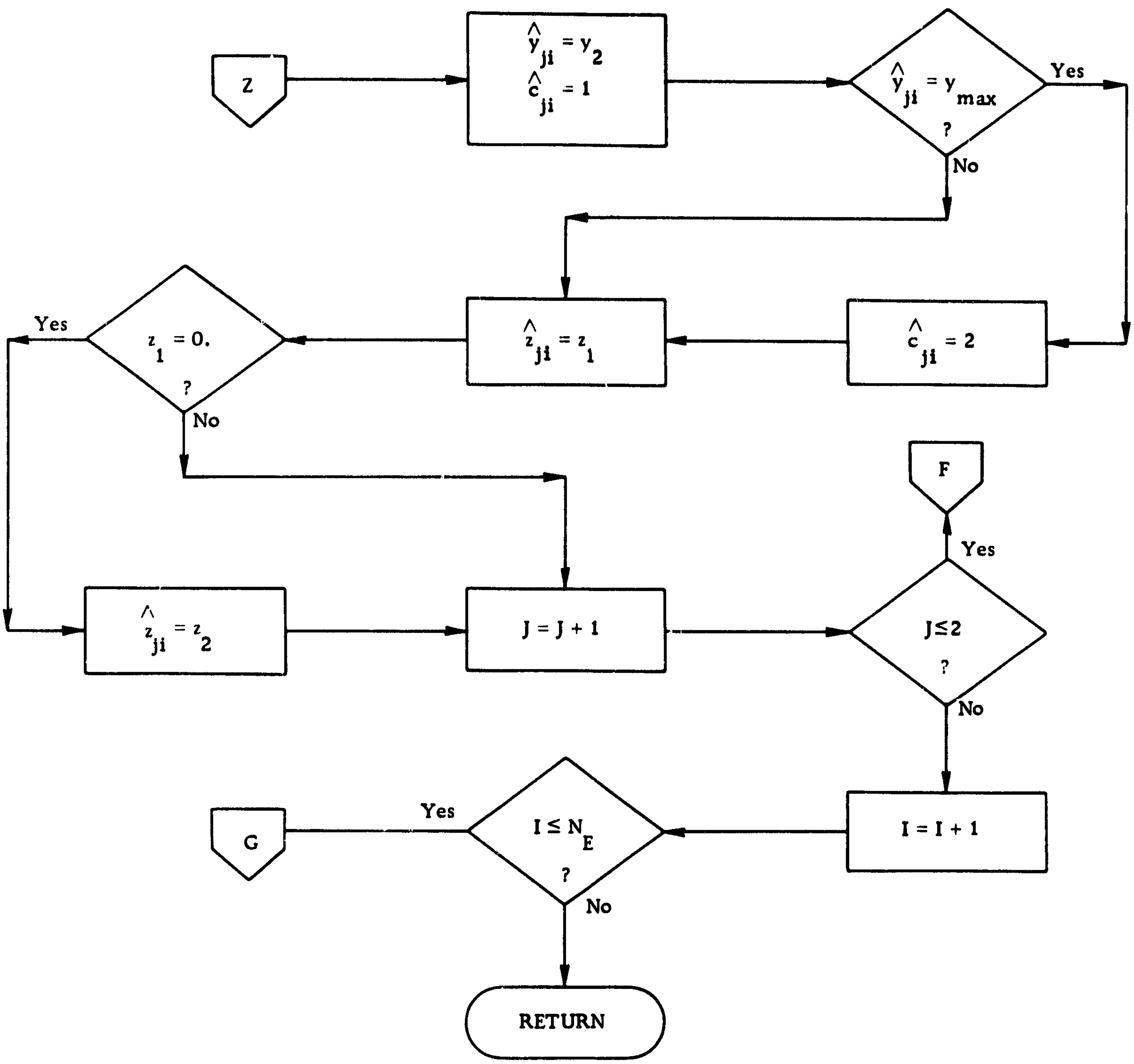
Utility Routines and Common References

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


SUBROUTINE RANGE
Set
$\hat{Z} = \hat{Y} = \hat{C} = 0.$
I = 1
G
F
J = 1
$Z_1 = Z_2 = y_1 = 0.$
$y_2 = y_{max}$
Get
TYPE, R, b
TYPE = 3?
No
Yes
R = 0. ?
No
Yes
b = 0. ?
No
Yes
C
Z
R > 0. ?
Yes
No
$y_{max} \geq R$?
Yes
No
$Z_2 = y_2 = R$
$Z_2 = R-(R^2-y_{max}^2)1/2$
Z
$y_{max} \geq |R|$?
No
Yes
B
A


A
$Z_1 = R$
$y_2 = |R|$
Z
B
$Z_1 =$
$R+(R^2-y^2_{max})1/2$
Z
C
b = 1. ?
Yes
$Z_T =$
$y^2_{max}/2R$
No
Yes
E
b >1. ?
No
Compute
a_1 a_2
ellipse
D
$Z_2 = Z_T$
Yes
R > 0. ?
No
$Z_1 = Z_T$
Z


D
R > 0.
?
Yes
No
$y_{max} \geq a_1$
?
No
Yes
$z_2 = a_2$
$y_2 = a_1$
Z
$y_{max} \geq a_1$
?
No
Yes
Compute z_1
Compute z_2
$z_1 = a_2$
$y_2 = a_1$
$z_2 = a_2 + \Delta$
Z
E
Compute
a_1 a_2 Δ
hyperbola
R> 0.
?
Yes
No
Z
$z_1 = a_2 - \Delta$


Z
$\hat{y}_{ji} = y_2$
$\hat{c}_{ji} = 1$
$\hat{y}_{ji} = y_{max}$?
Yes
No
$\hat{c}_{ji} = 2$
$\hat{z}_{ji} = z_1$
$z_1 = 0.$?
Yes
No
$\hat{z}_{ji} = z_2$
J = J + 1
J≤2 ?
Yes
F
No
I = I + 1
$I \leq N_E$?
Yes
G
No
RETURN

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

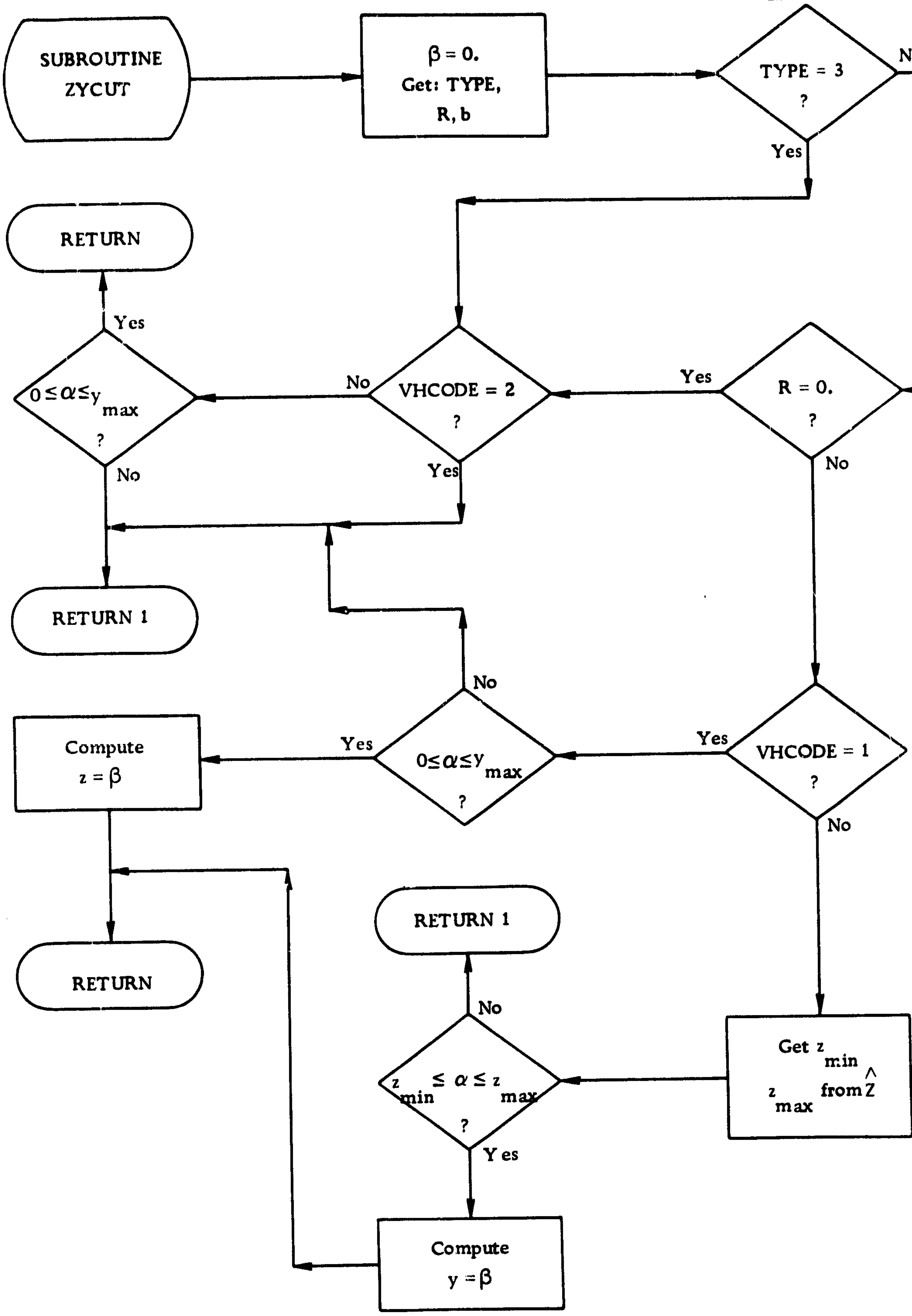
CALL ZYCUT(VHCODE, ALFA, IS, BETA, *, ELNO, SURNO)

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.


SUBROUTINE ZYCUT
β = 0.
Get: TYPE, R, b
TYPE = 3 ?
No
Yes
RETURN
Yes
0 ≤ α ≤ y max ?
No
VHCODE = 2 ?
Yes
R = 0. ?
No
Yes
No
RETURN 1
No
Compute z = β
Yes
0 ≤ α ≤ y max ?
Yes
VHCODE = 1 ?
No
RETURN 1
RETURN
No
z min ≤ α ≤ z max ?
Get z min z max from Ẑ
Yes
Compute y = 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,

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

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/

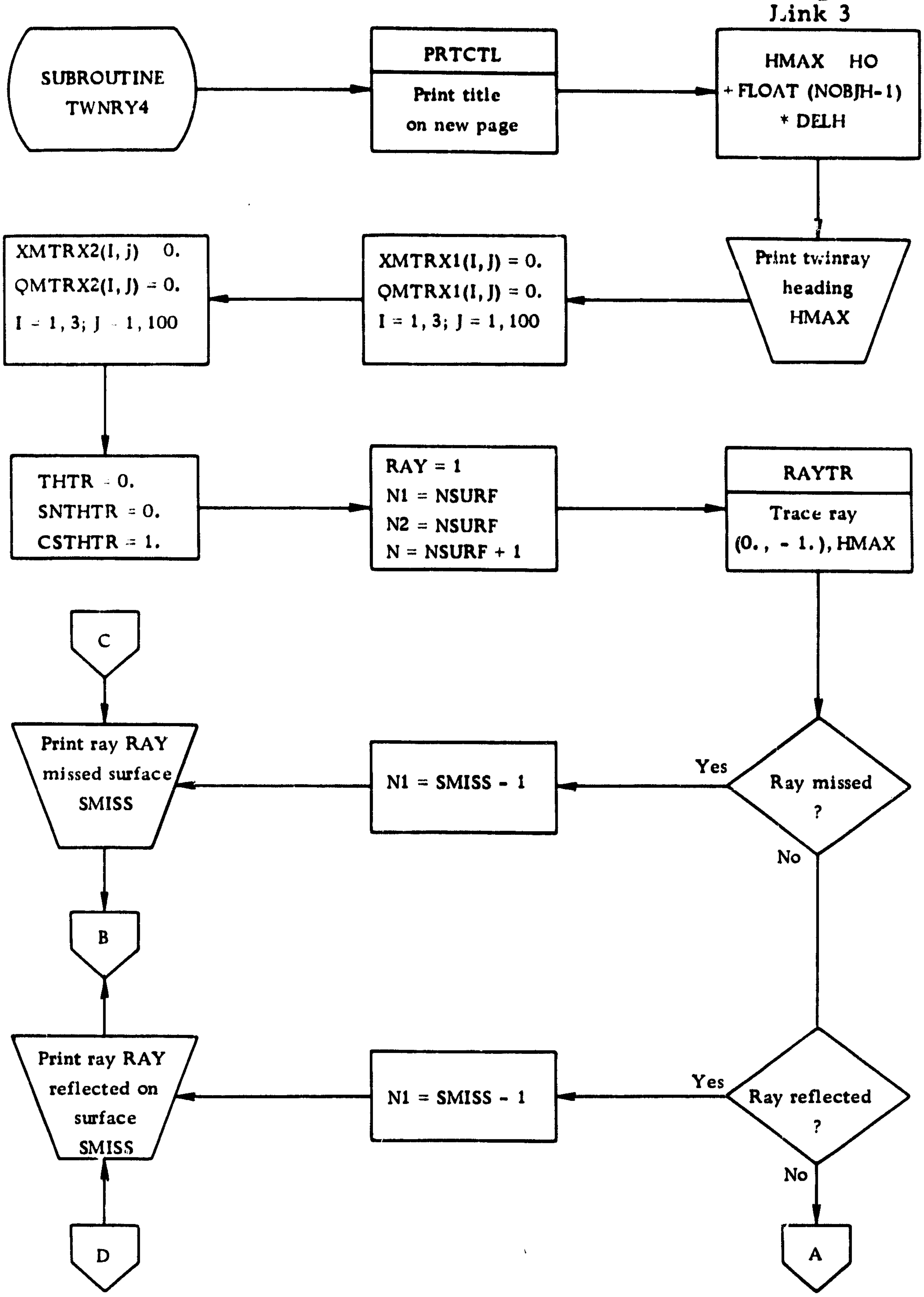
COMMON/AZOBJ/

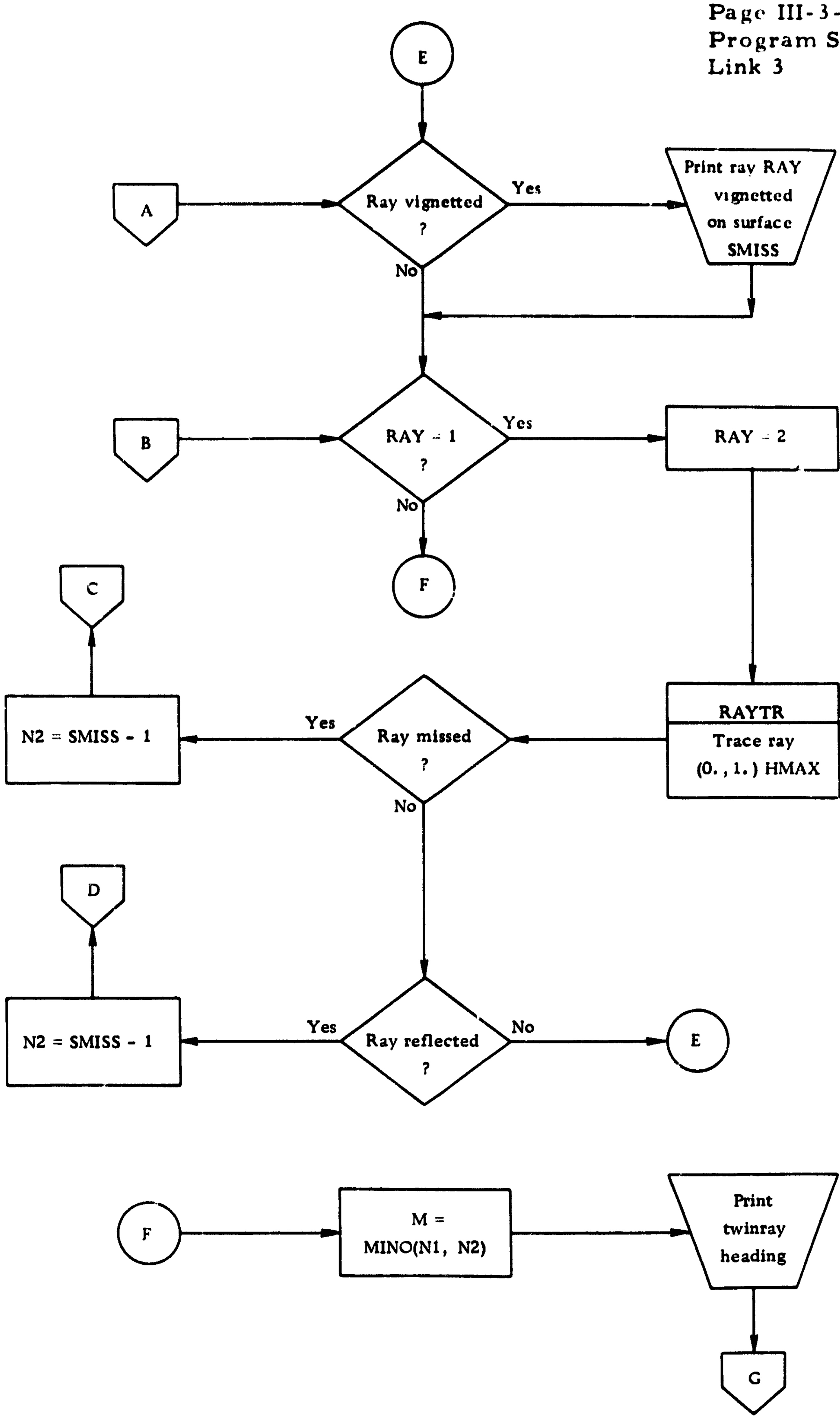
COMMON/PRNT/

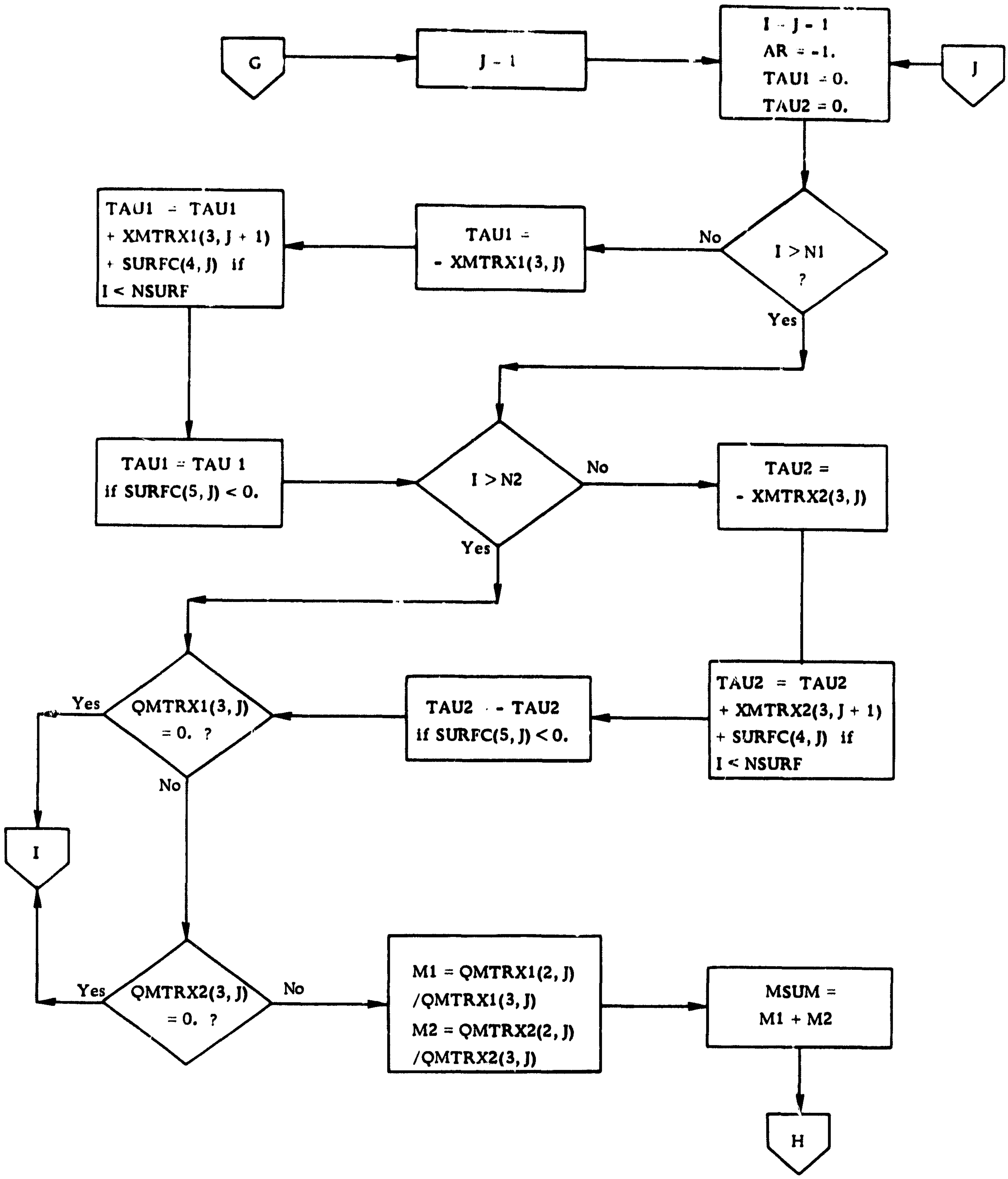
COMMON/TMTWIN/

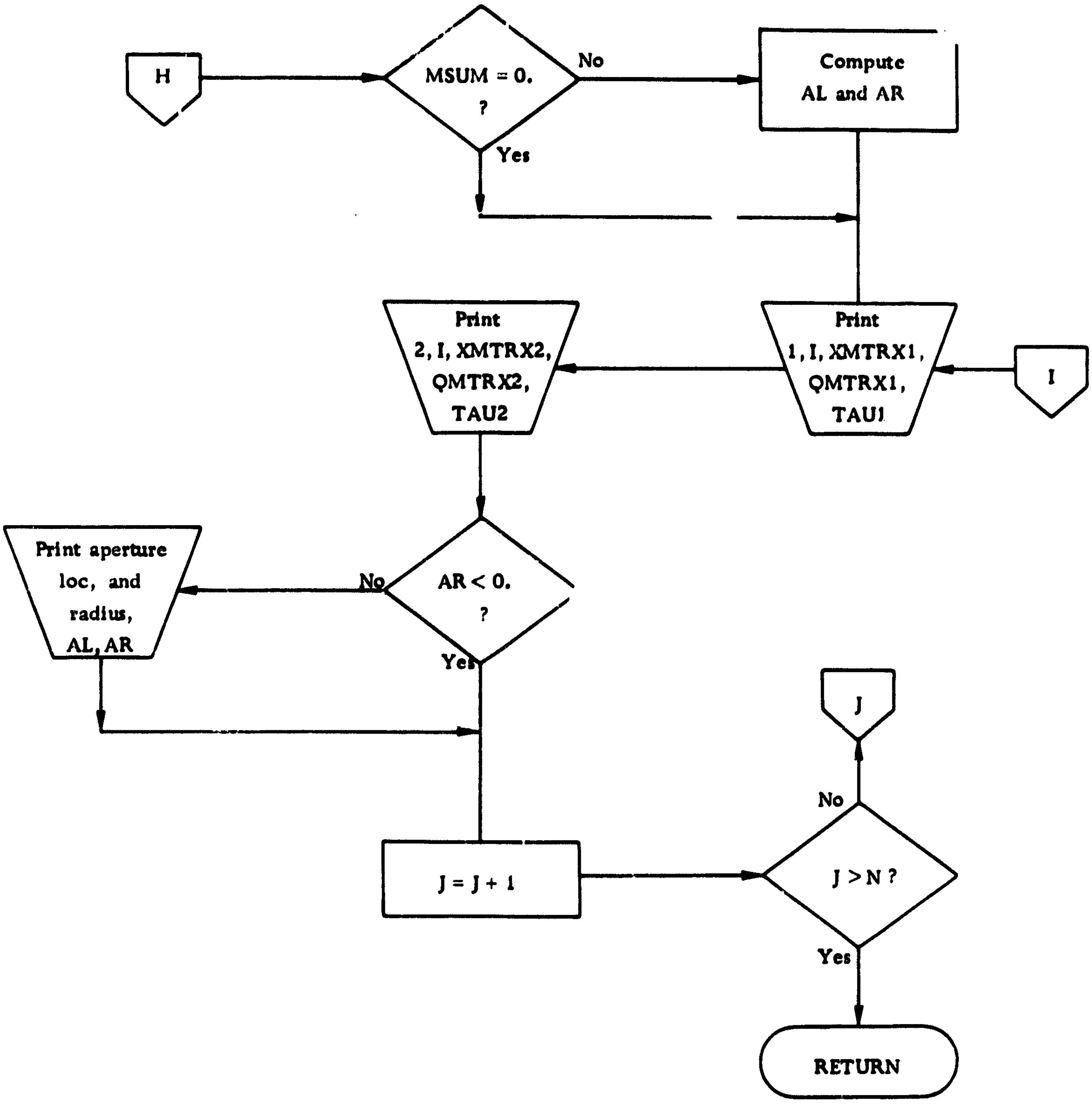
PRTCTL

RAYTR


SUBROUTINE TWNRY4
PRTCTL
Print title on new page
HMAX HO + FLOAT (NOBJH-1) * DELH
Print twinray heading HMAX
XMTRX1(I, J) = 0.
QMTRX1(I, J) = 0.
I = 1, 3; J = 1, 100
XMTRX2(I, j) 0.
QMTRX2(I, J) = 0.
I = 1, 3; J = 1, 100
THTR = 0.
SNTHTR = 0.
CSTHTR = 1.
RAY = 1
N1 = NSURF
N2 = NSURF
N = NSURF + 1
RAYTR
Trace ray (0., - 1.), HMAX
C
Print ray RAY missed surface SMISS
Yes
N1 = SMISS - 1
Ray missed ?
No
B
Print ray RAY reflected on surface SMISS
Yes
N1 = SMISS - 1
Ray reflected ?
No
D
A


E
Ray vignetted ?
Yes
No
A
Print ray RAY vignetted on surface SMISS
B
RAY = 1 ?
Yes
No
RAY = 2
F
C
RAYTR
Trace ray (0., 1.) HMAX
Ray missed ?
Yes
No
N2 = SMISS - 1
D
Ray reflected ?
Yes
No
N2 = SMISS - 1
E
F
M = MINO(N1, N2)
Print twinray heading
G


G
J = 1
I = J - 1
AR = -1.
TAU1 = 0.
TAU2 = 0.
J
I > N1 ?
No
Yes
TAU1 =
- XMTRX1(3, J)
TAU1 = TAU1
+ XMTRX1(3, J + 1)
+ SURFC(4, J) if
I < NSURF
TAU1 = TAU 1
if SURFC(5, J) < 0.
I > N2
No
Yes
TAU2 =
- XMTRX2(3, J)
TAU2 = TAU2
+ XMTRX2(3, J + 1)
+ SURFC(4, J) if
I < NSURF
TAU2 = - TAU2
if SURFC(5, J) < 0.
QMTRX1(3, J)
= 0. ?
Yes
No
I
QMTRX2(3, J)
= 0. ?
Yes
No
M1 = QMTRX1(2, J)
/QMTRX1(3, J)
M2 = QMTRX2(2, J)
/QMTRX2(3, J)
MSUM =
M1 + M2
H


H
MSUM = 0. ?
No
Yes
Compute AL and AR
Print 2, I, XMTRX2, QMTRX2, TAU2
Print 1, I, XMTRX1, QMTRX1, TAU1
I
Print aperture loc, and radius, AL, AR
AR < 0. ?
No
Yes
J
No
J = J + 1
J > N ?
Yes
RETURN

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,

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

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 $N_k = N$.

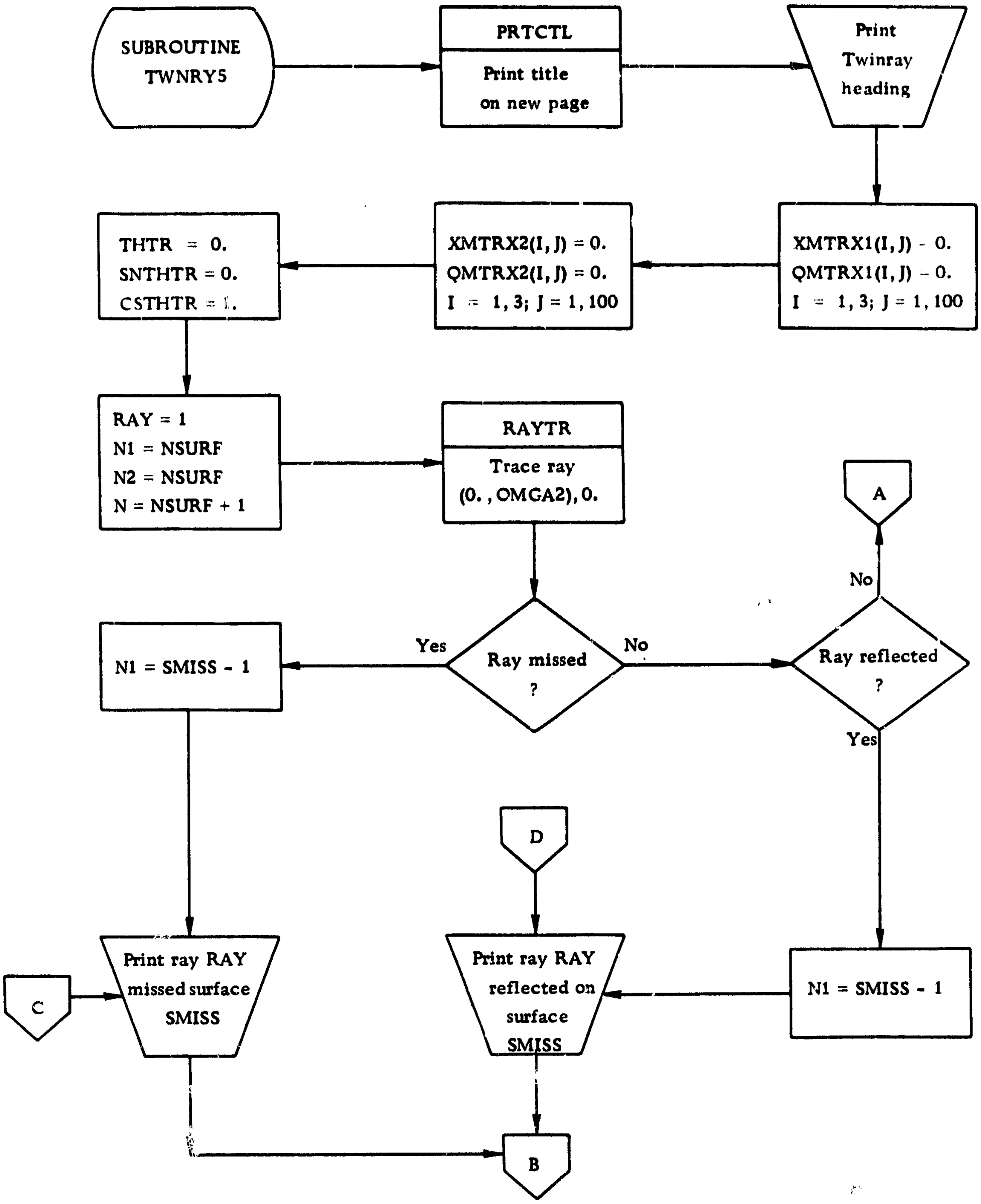
The data for position is stored in arrays XMTRX1 and XMTRX2. The direction data is contained in arrays QMTRX1 and QMTRX2. The thickness is also computed and printed in tabular form along with the position and direction values for each of the two rays at each surface. TWNR5Z must be called additional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

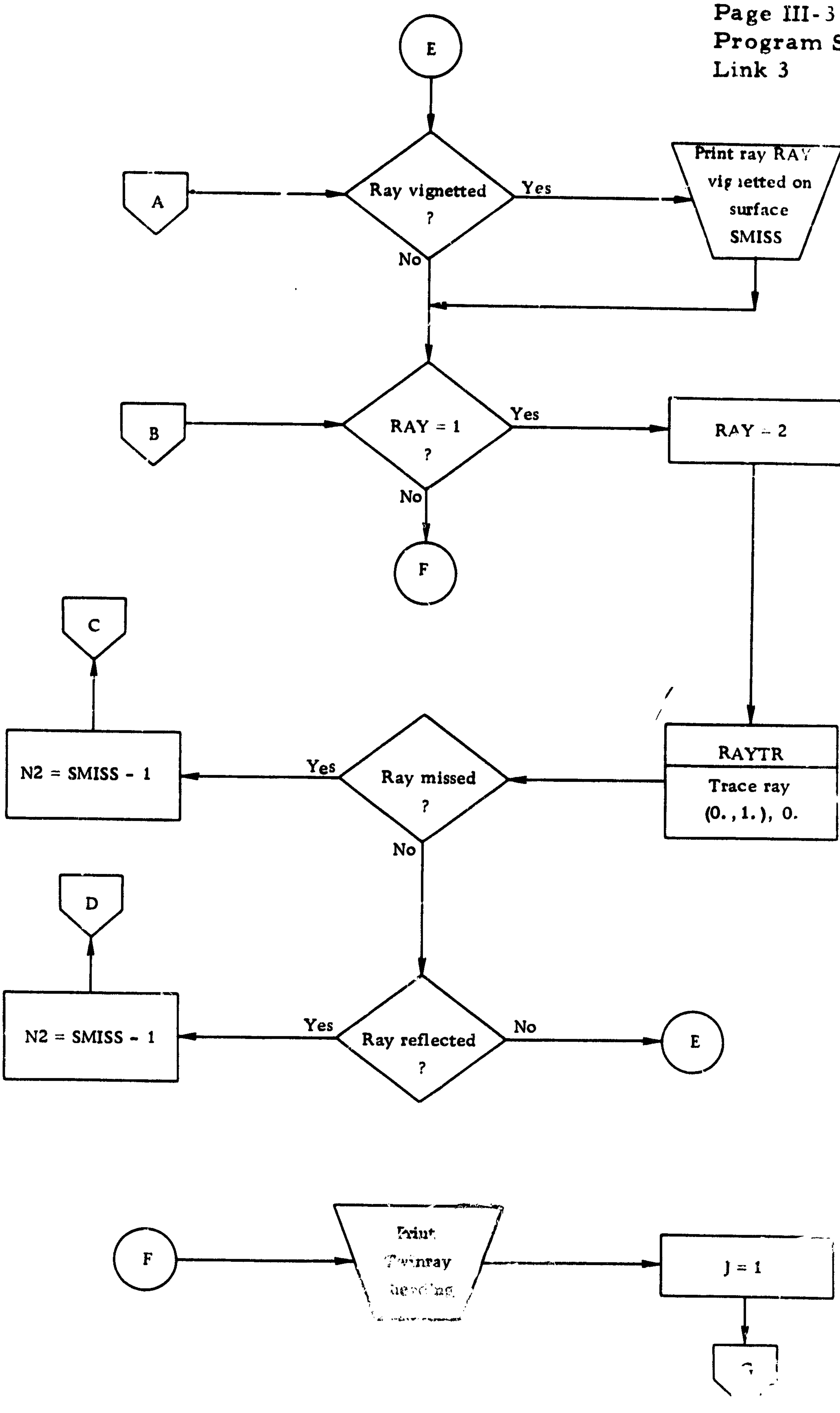
Calling Sequence

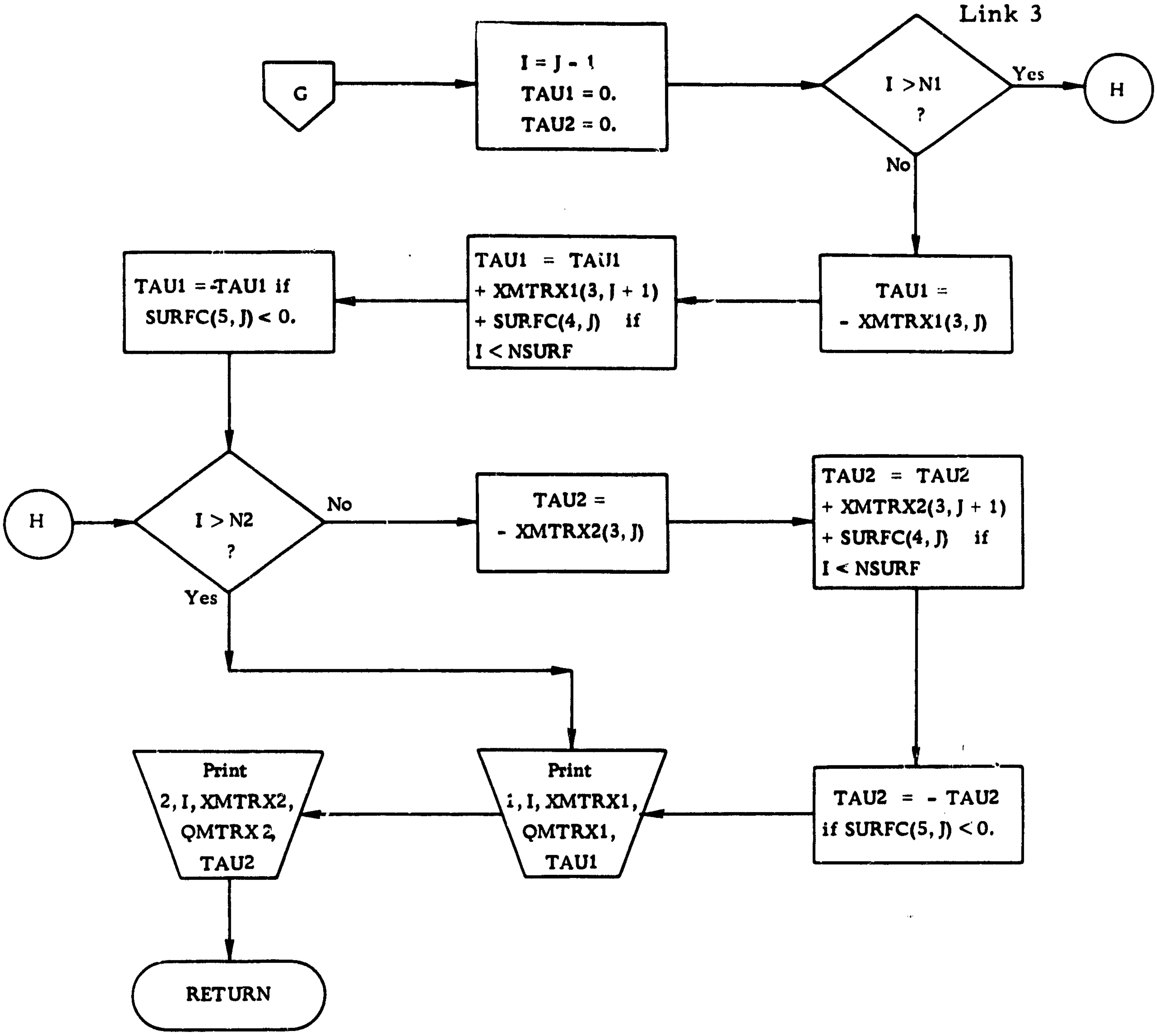
CALL TWNRY5

Utility Routines and Common References

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


SUBROUTINE TWNRY5
PRTCTL
Print title on new page
Print Twinray heading
XMTRX1(I,J) = 0.
QMTRX1(I,J) = 0.
I = 1, 3; J = 1, 100
XMTRX2(I,J) = 0.
QMTRX2(I,J) = 0.
I = 1, 3; J = 1, 100
THTR = 0.
SNTHTR = 0.
CSTHTR = 1.
RAY = 1
N1 = NSURF
N2 = NSURF
N = NSURF + 1
RAYTR
Trace ray (0., OMGA2), 0.
Ray missed ?
Yes
No
N1 = SMISS - 1
Ray reflected ?
No
Yes
A
D
Print ray RAY missed surface SMISS
C
Print ray RAY reflected on surface SMISS
N1 = SMISS - 1
B


E
A
Ray vignetted ?
Yes
No
Print ray RAY vignetted on surface SMISS
B
RAY = 1 ?
Yes
No
RAY = 2
F
C
RAYTR
Trace ray (0., 1.), 0.
Ray missed ?
Yes
No
N2 = SMISS - 1
D
Ray reflected ?
Yes
No
N2 = SMISS - 1
E
F
Print Twinray heading
J = 1


G
I = J - 1
TAU1 = 0.
TAU2 = 0.
I > N1
?
Yes
H
No
TAU1 =
- XMTRX1(3, J)
TAU1 = TAU1
+ XMTRX1(3, J + 1)
+ SURFC(4, J) if
I < NSURF
TAU1 = -TAU1 if
SURFC(5, J) < 0.
H
I > N2
?
No
TAU2 =
- XMTRX2(3, J)
TAU2 = TAU2
+ XMTRX2(3, J + 1)
+ SURFC(4, J) if
I < NSURF
Yes
TAU2 = - TAU2
if SURFC(5, J) < 0.
Print
1, I, XMTRX1,
QMTRX1,
TAU1
Print
2, I, XMTRX2,
QMTRX2,
TAU2
RETURN

3.4 LINK 4

Link 4 is essentially and 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.

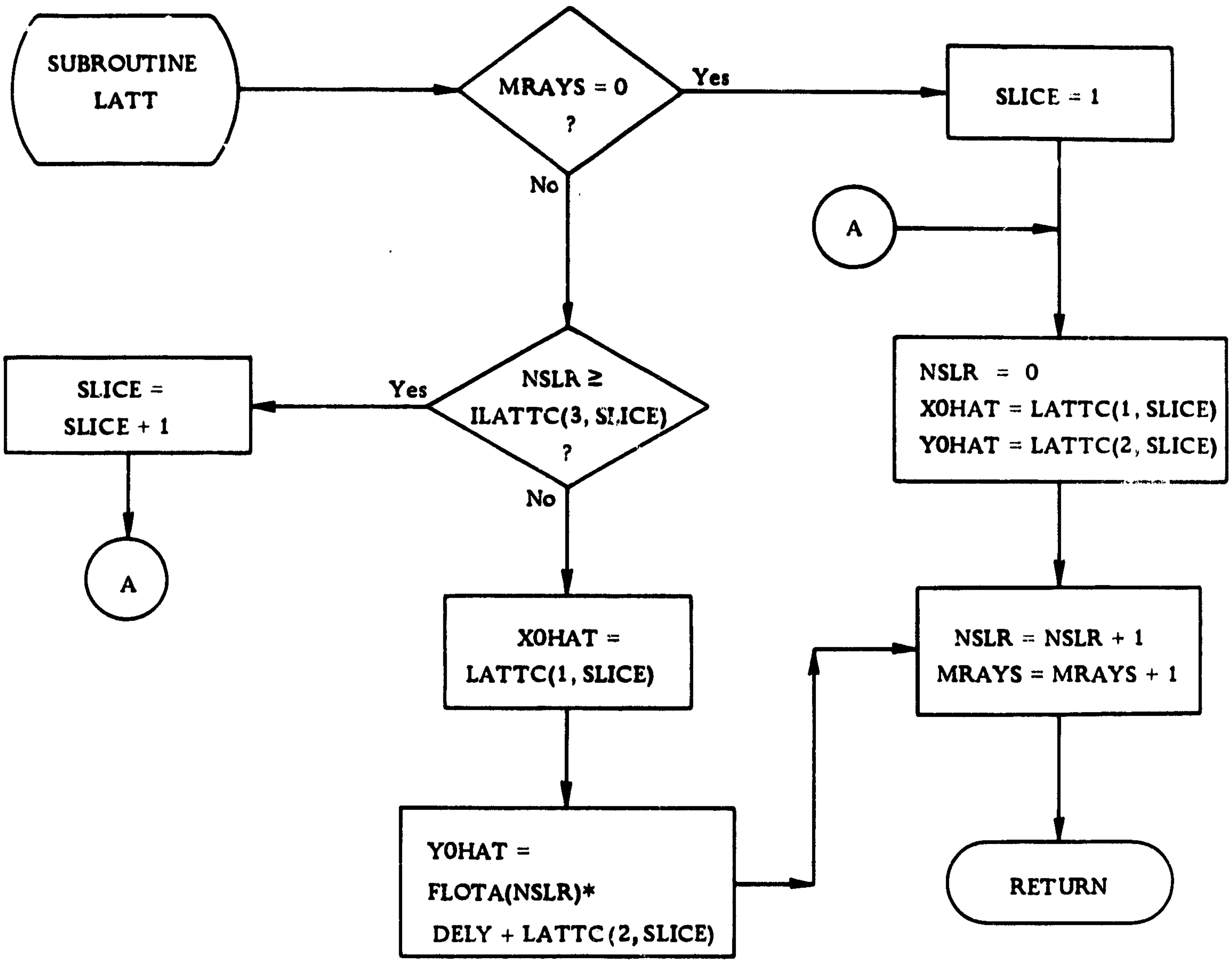
Calling Sequence

CALL LATT (XOHAT, YOHAT, MRAYS)

XOHAT:	Storage for x_o.
YOHAT:	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/


SUBROUTINE LATT
MRAYS = 0 ?
Yes
No
SLICE = 1
A
NSLR = 0
X0HAT = LATTC(1, SLICE)
Y0HAT = LATTC(2, SLICE)
NSLR ≥ ILATTC(3, SLICE) ?
Yes
No
SLICE = SLICE + 1
A
X0HAT = LATTC(1, SLICE)
NSLR = NSLR + 1
MRAYS = MRAYS + 1
Y0HAT = FLOTA(NSLR)* DELY + LATTC(2, SLICE)
RETURN

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)	must be set by calling program
SNTHTR:	0.	(Sin θ)	must be set by calling program
CSTHTR:	1.	(Cos θ)	must be set by calling program

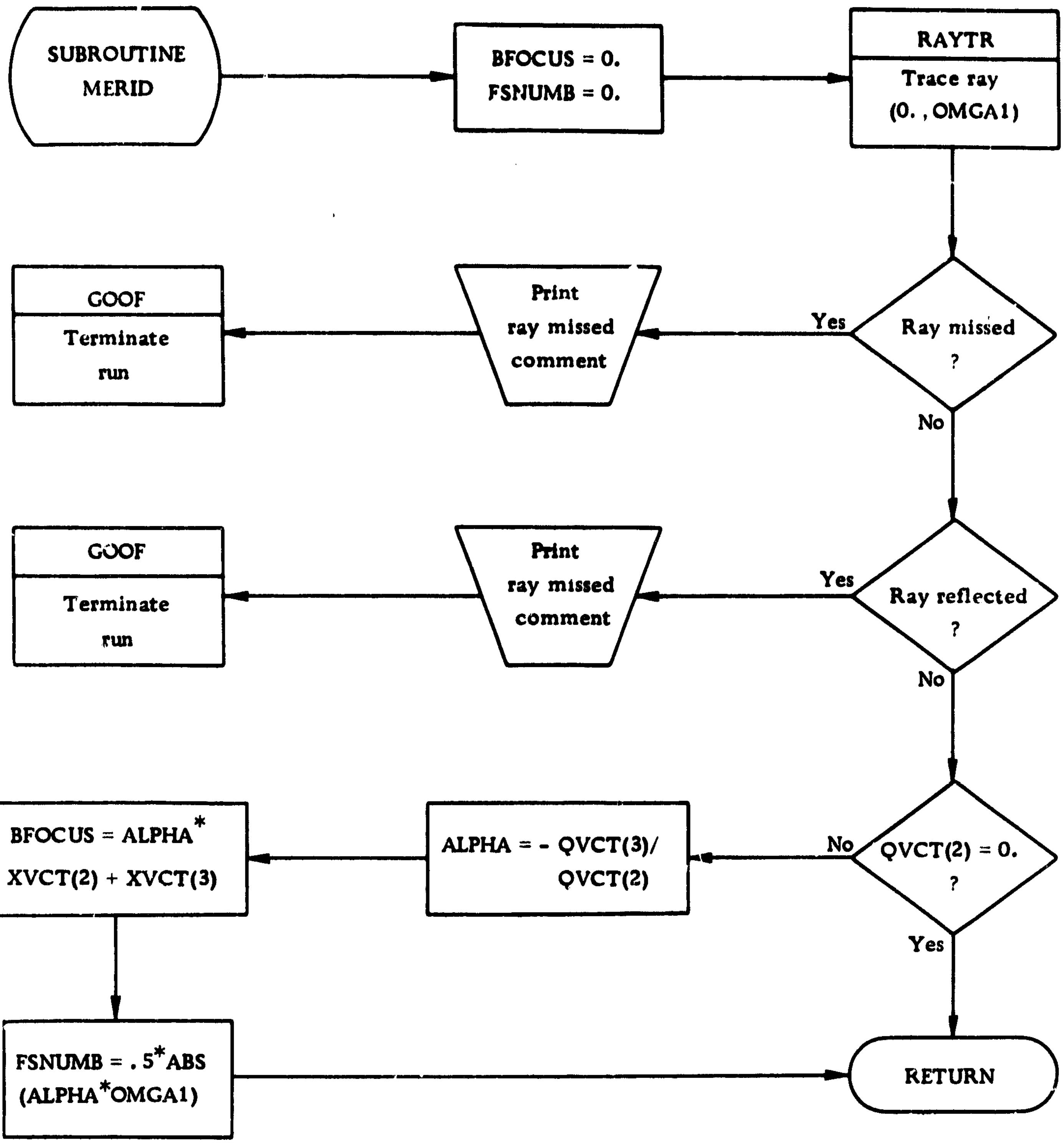
Utility Routines and Common References

COMMON/DATA/

COMMON/AZOBJ/

RAYTR

GOOF


SUBROUTINE MERID
BFOCUS = 0.
FSNUMB = 0.
RAYTR
Trace ray (0., OMGA1)
Ray missed ?
Yes
No
Print ray missed comment
GOOF
Terminate run
Ray reflected ?
Yes
No
Print ray missed comment
GOOF
Terminate run
QVCT(2) = 0. ?
No
Yes
ALPHA = - QVCT(3)/ QVCT(2)
BFOCUS = ALPHA* XVCT(2) + XVCT(3)
FSNUMB = .5*ABS (ALPHA*OMGA1)
RETURN

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
$\overline{N}_R$	Number of successes

A vignetted 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

$\overline{m}_x$, $\overline{k}_x$, $\overline{m}_y$, $\overline{k}_y$

A_x, B_x, . . . , B, C

$\tilde{m}_{xi}$	(XCURL)	$1 \le i \le \overline{N}_R$
$\tilde{m}_{yi}$	(XCURL)	" "
$\tilde{k}_{xi}$	(TEMPS)	" "
$\tilde{k}_{yi}$	(TEMPS (200))	" "

2) CMFLAG = 2

$\overline{x}(S_j^*)$, $\overline{y}(S_j^*)$

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

3) CMFLAG = 3

$\bar{x}(S_j^*)$, $\bar{y}(S_j^*)$

$\tilde{x}_i(S_j^*)$ (XCURL) $1 \le i \le \bar{N}_R$

$\tilde{y}_i(S_j^*)$ (YCURL) " "

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

4) CMFLAG = 4

$\bar{x}(S_j^*)$, $\bar{y}(S_j^*)$

$x^*(S_j^*)$, $y^*(S_j^*)$, $R^*(S_j^*)$

$\tilde{x}_i(S_j^*)$ (XCURL) $1 \le i \le \bar{N}_R$

$\tilde{y}_i(S_j^*)$ (YCURL) " "

5) CMFLAG = 5

$\bar{m}_x$, $\bar{k}_x$, $\bar{m}_y$, $\bar{k}_y$

$\tilde{m}_{xi}$ (XCURL) $1 \le i \le \bar{N}_R$

$\tilde{k}_{xi}$ (YCURL) " "

$\tilde{m}_{yi}$ (TEMPS) " "

$\tilde{k}_{yi}$ (TEMPS (200)) " "

Calling Sequence

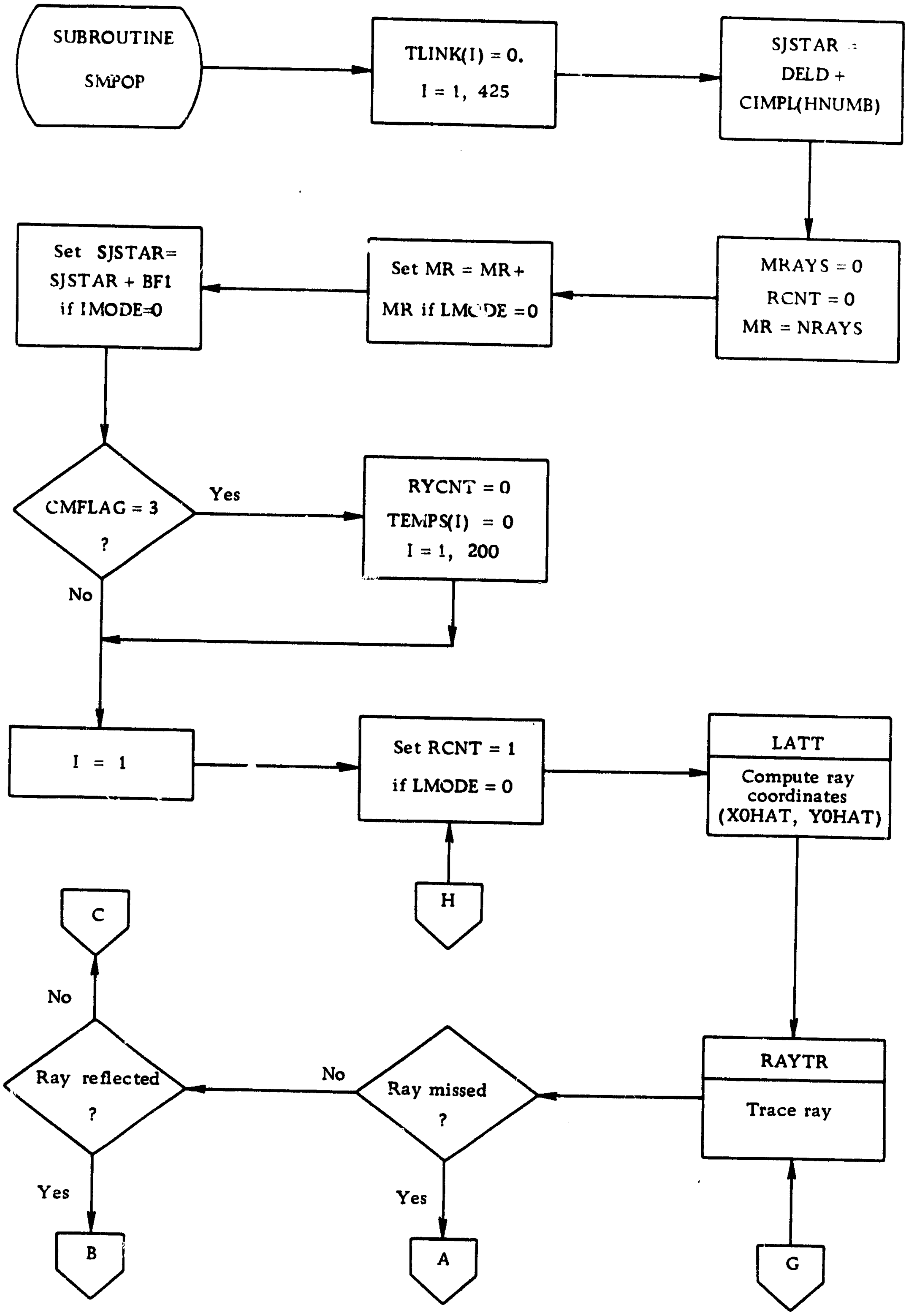
CALL SMPOP(COLOR, HNUMB, HEIGHT, BF1,CMFLAG, TEMPS

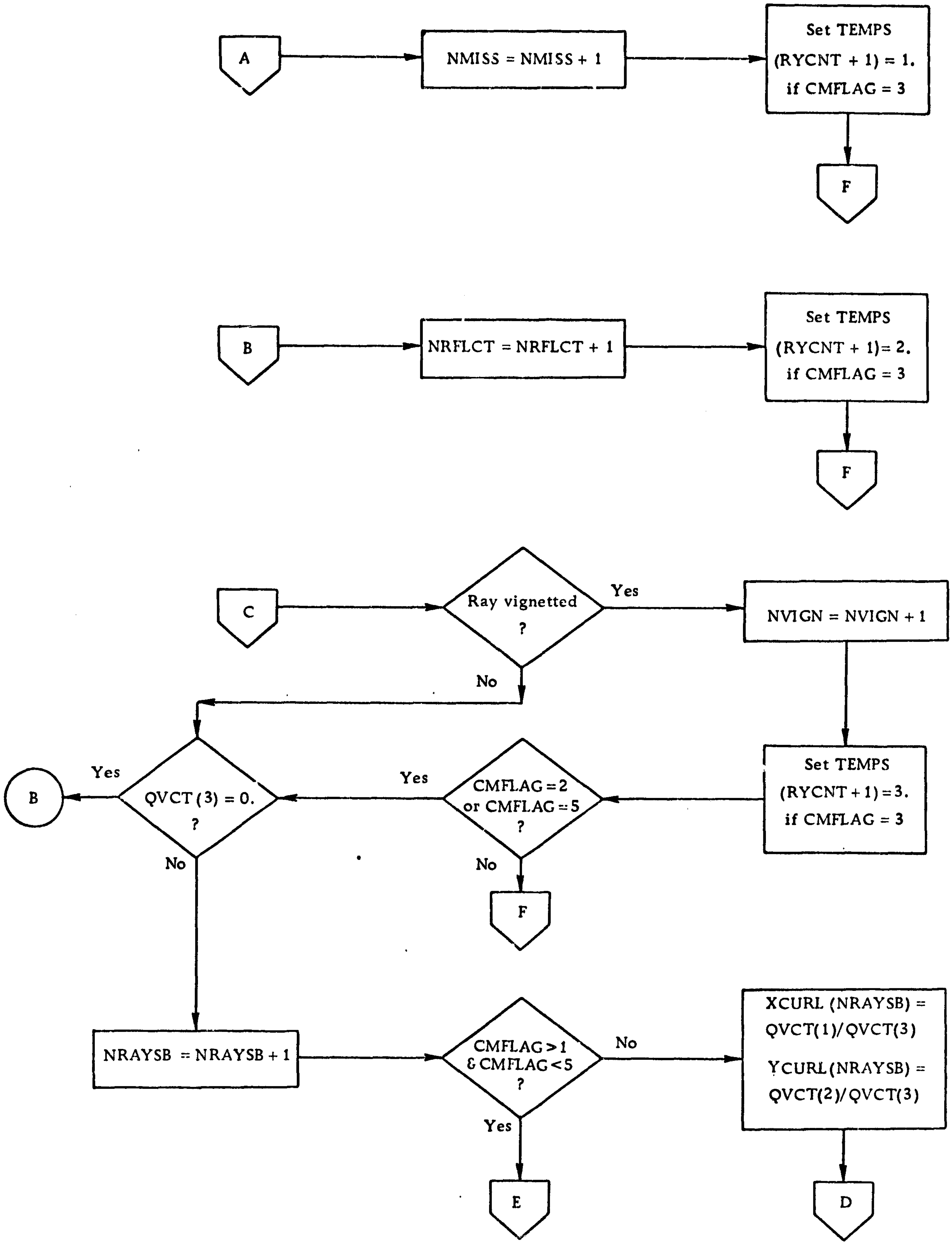
COLOR	:	Color index i
HNUMB	:	Object point index j
HEIGHT	:	Object height h_j
BF1	:	Back focus color 1
CMFLAG	:	Computation flag
TEMPS	:	Used if CMFLAG = 1, 3, or 5

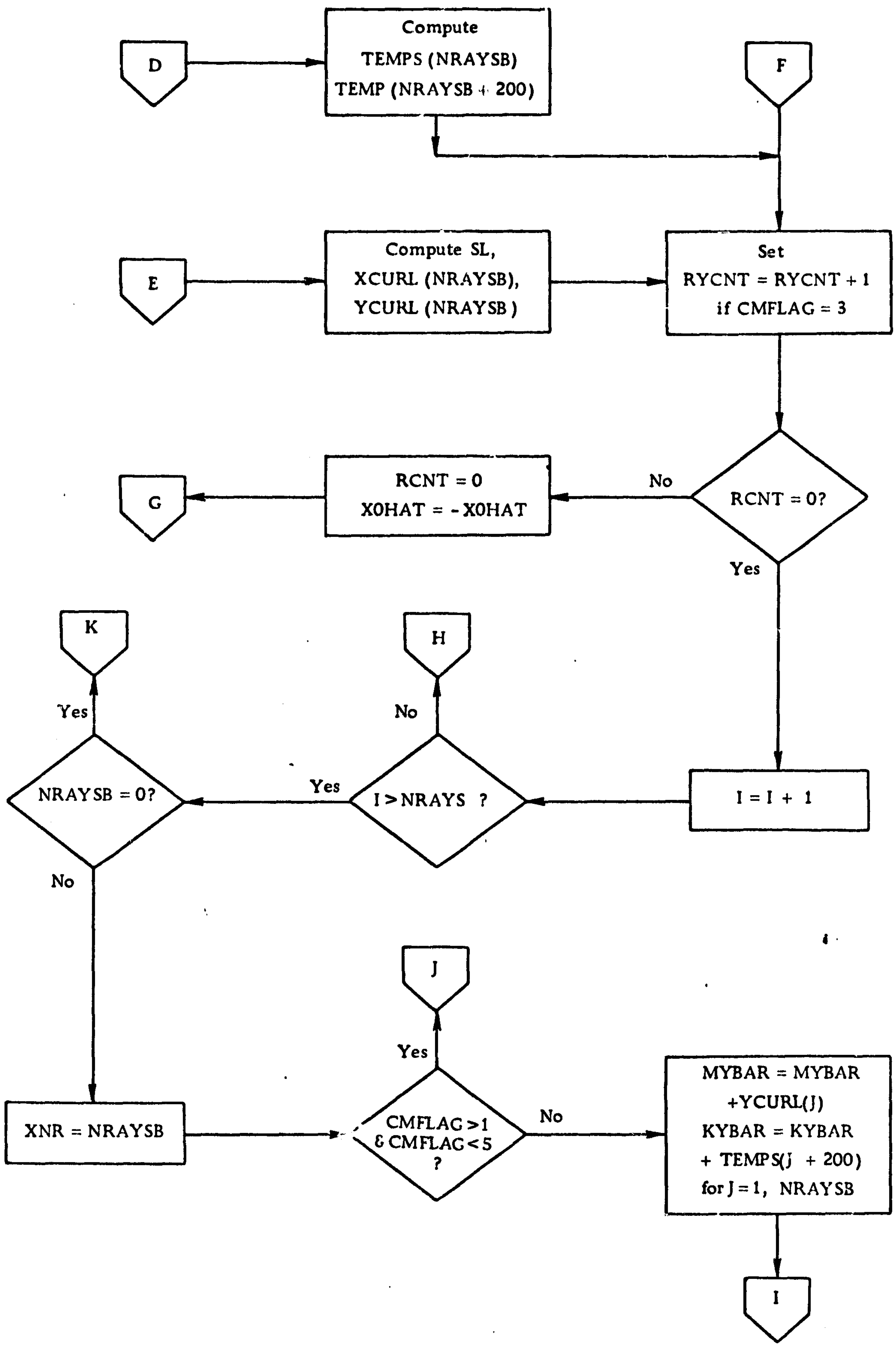
The remaining variables are transmitted through TMPATT COMMON as seen on page III-2-10.

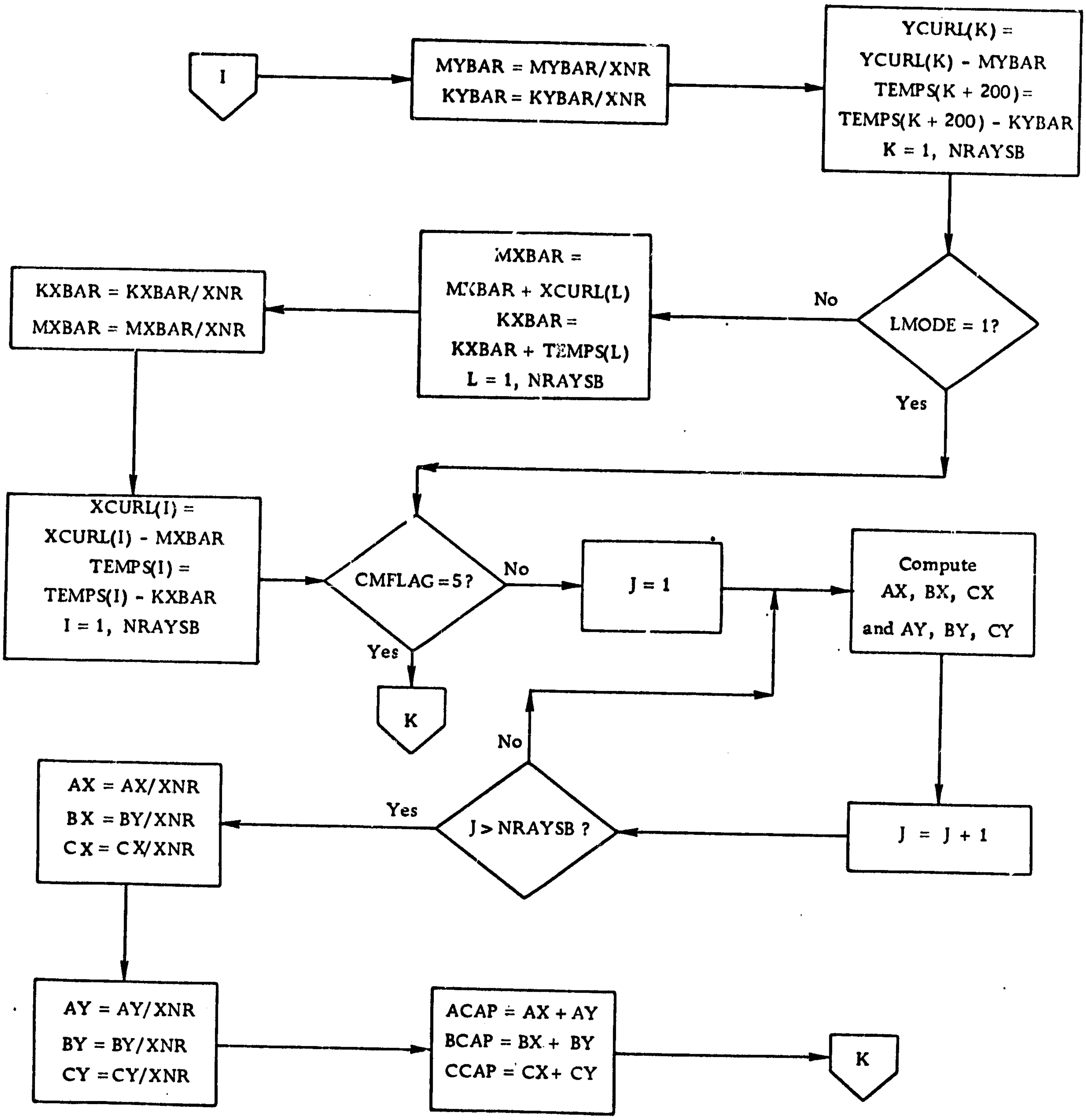
Utility Routines and Common References

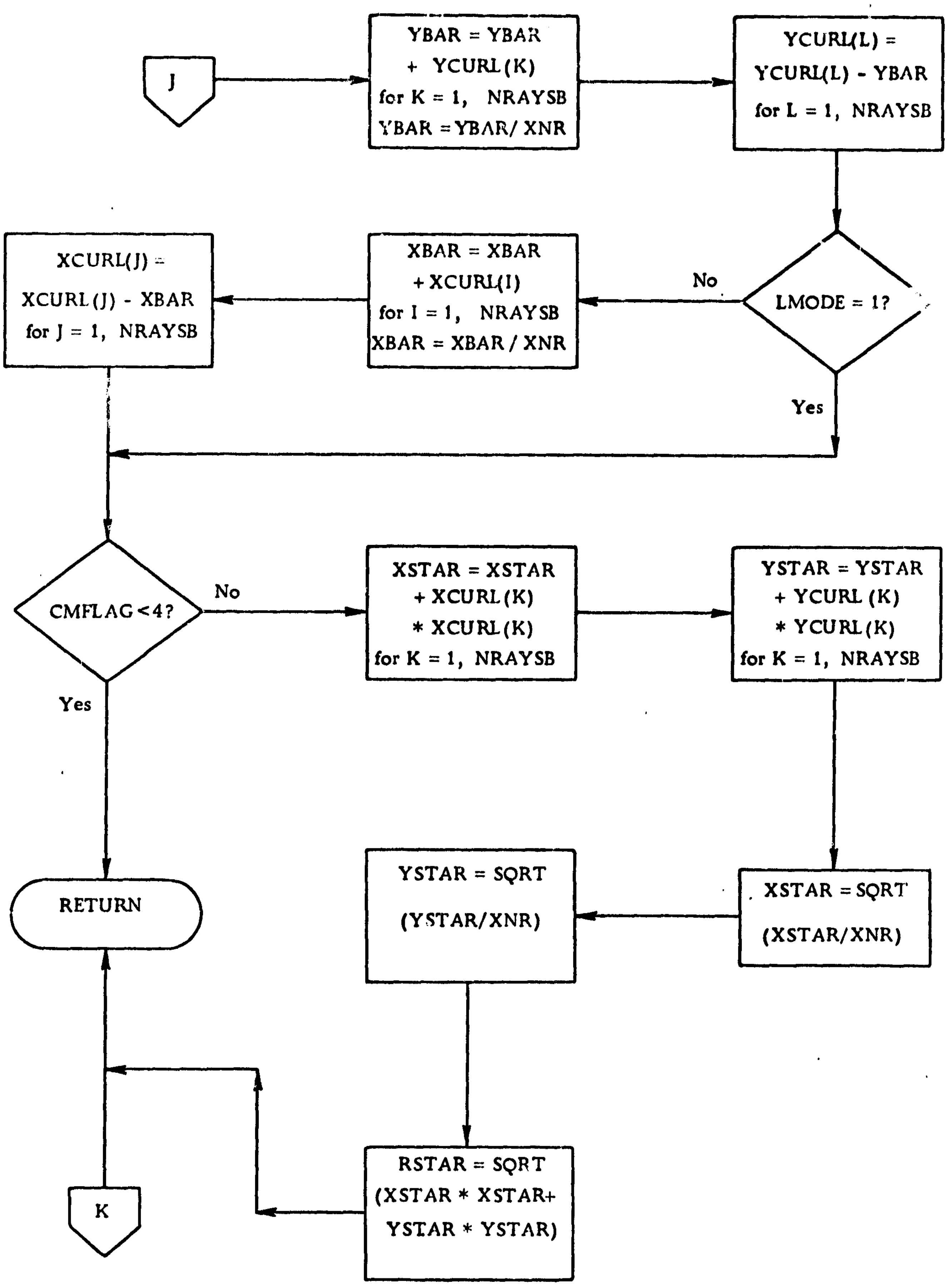
COMMON/DATA/
COMMON/AZOBJ/
COMMON/TMPATT/
RAYTR
LATT


SUBROUTINE SMPOP
TLINK(I) = 0.
I = 1, 425
SJSTAR =
DELD +
CIMPL(HNUMB)
MRAYS = 0
RCNT = 0
MR = NRAYS
Set MR = MR +
MR if LMODE = 0
Set SJSTAR=
SJSTAR + BF1
if IMODE=0
CMFLAG = 3
?
Yes
No
RYCNT = 0
TEMPS(I) = 0
I = 1, 200
I = 1
Set RCNT = 1
if LMODE = 0
LATT
Compute ray
coordinates
(X0HAT, Y0HAT)
H
C
No
Ray reflected
?
Yes
B
No
Ray missed
?
Yes
A
RAYTR
Trace ray
G


A
NMISS = NMISS + 1
Set TEMPS
(RYCNT + 1) = 1.
if CMFLAG = 3
F
B
NRFLCT = NRFLCT + 1
Set TEMPS
(RYCNT + 1)= 2.
if CMFLAG = 3
F
C
Ray vignetted
?
Yes
NVIGN = NVIGN + 1
No
Set TEMPS
(RYCNT + 1) = 3.
if CMFLAG = 3
CMFLAG = 2
or CMFLAG = 5
?
Yes
No
F
QVCT (3) = 0.
?
Yes
B
No
NRAYSB = NRAYSB + 1
CMFLAG > 1
& CMFLAG < 5
?
No
Yes
E
XCURL (NRAYSB) =
QVCT(1)/QVCT(3)
YCURL (NRAYSB) =
QVCT(2)/QVCT(3)
D


D
Compute
TEMPS (NRAYSB)
TEMP (NRAYSB + 200)
F
E
Compute SL,
XCURL (NRAYSB),
YCURL (NRAYSB)
Set
RYCNT = RYCNT + 1
if CMFLAG = 3
G
RCNT = 0
XOHAT = - XOHAT
No
RCNT = 0?
Yes
K
H
Yes
No
NRAYSB = 0?
Yes
I > NRAYS ?
I = I + 1
No
J
Yes
XNR = NRAYSB
CMFLAG > 1
& CMFLAG < 5
?
No
MYBAR = MYBAR
+YCURL(J)
KYBAR = KYBAR
+ TEMPS(J + 200)
for J = 1, NRAYSB
I


I
MYBAR = MYBAR/XNR
KYBAR = KYBAR/XNR
YCURL(K) = YCURL(K) - MYBAR
TEMPS(K + 200) = TEMPS(K + 200) - KYBAR
K = 1, NRAYSB
LMODE = 1?
No
Yes
MXBAR = MXBAR + XCURL(L)
KXBAR = KXBAR + TEMPS(L)
L = 1, NRAYSB
KXBAR = KXBAR/XNR
MXBAR = MXBAR/XNR
XCURL(I) = XCURL(I) - MXBAR
TEMPS(I) = TEMPS(I) - KXBAR
I = 1, NRAYSB
CMFLAG = 5?
No
Yes
K
J = 1
Compute AX, BX, CX and AY, BY, CY
J = J + 1
J > NRAYSB ?
No
Yes
AX = AX/XNR
BX = BY/XNR
CX = CX/XNR
AY = AY/XNR
BY = BY/XNR
CY = CY/XNR
ACAP = AX + AY
BCAP = BX + BY
CCAP = CX + CY
K


J
YBAR = YBAR + YCURL(K) for K = 1, NRAYSB YBAR = YBAR/ XNR
YCURL(L) = YCURL(L) - YBAR for L = 1, NRAYSB
LMODE = 1?
No
Yes
XBAR = XBAR + XCURL(I) for I = 1, NRAYSB XBAR = XBAR / XNR
XCURL(J) = XCURL(J) - XBAR for J = 1, NRAYSB
CMFLAG < 4?
No
Yes
XSTAR = XSTAR + XCURL(K) * XCURL(K) for K = 1, NRAYSB
YSTAR = YSTAR + YCURL (K) * YCURL(K) for K = 1, NRAYSB
XSTAR = SQRT (XSTAR/XNR)
YSTAR = SQRT (YSTAR/XNR)
RSTAR = SQRT (XSTAR * XSTAR+ YSTAR * YSTAR)
RETURN
K

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.

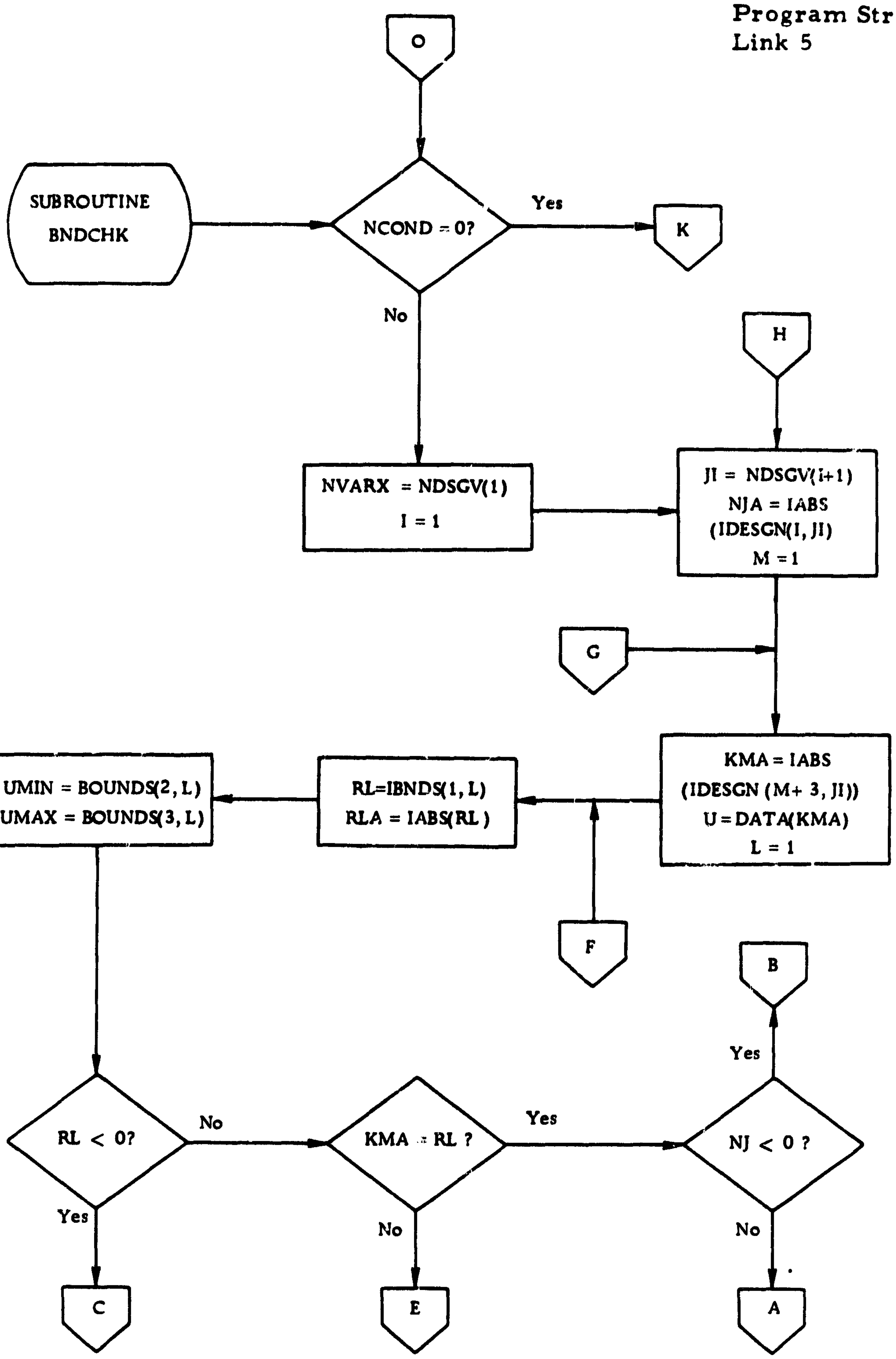
When a boundary violation has occured an appropriate message is printed, and BNDCHK executes the alternate return signifying the detection of an error condition. The error message print out can be supressed by setting IFLAG to a non-zero value when calling BNDCHK.

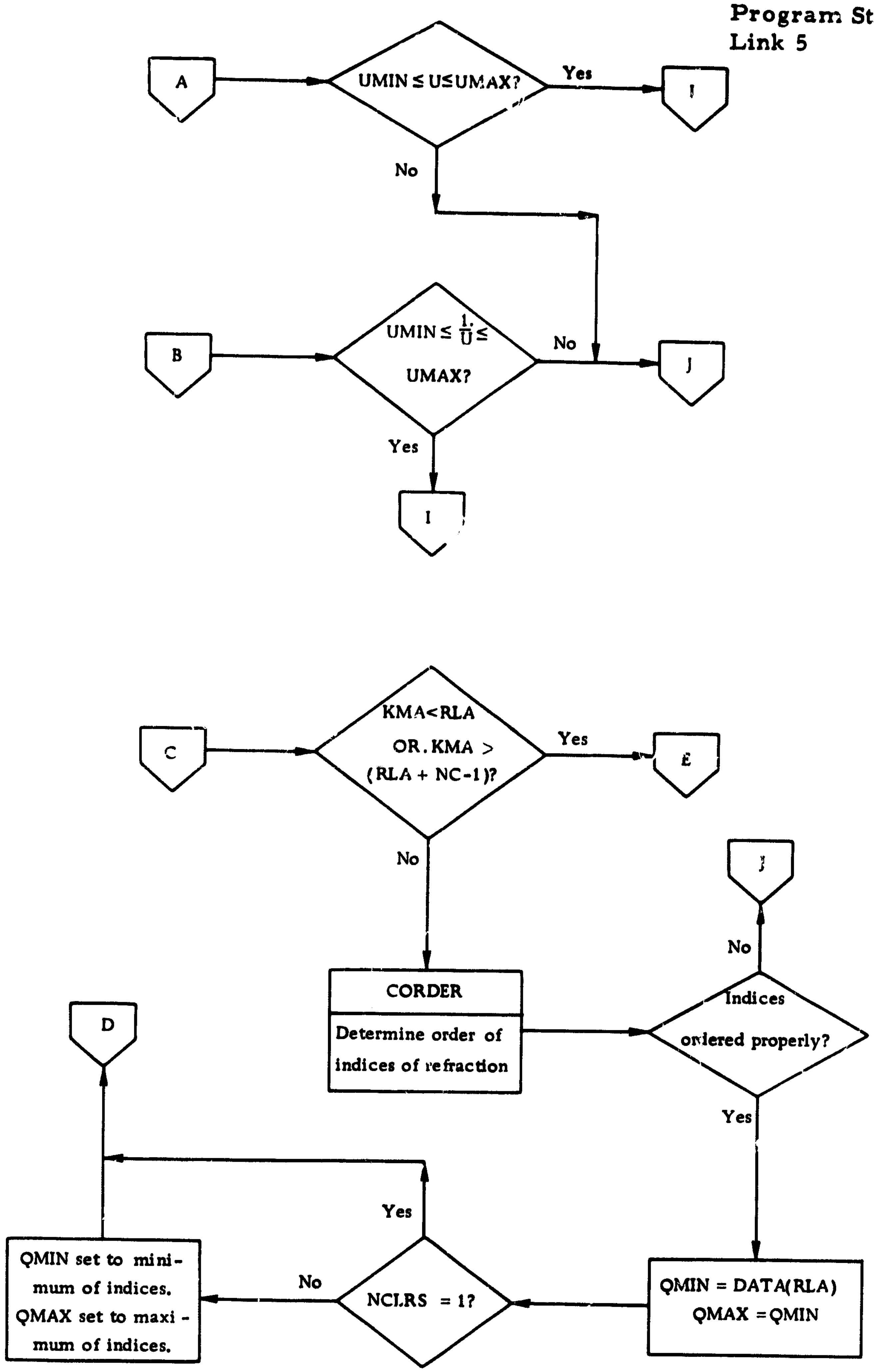
Calling Sequence

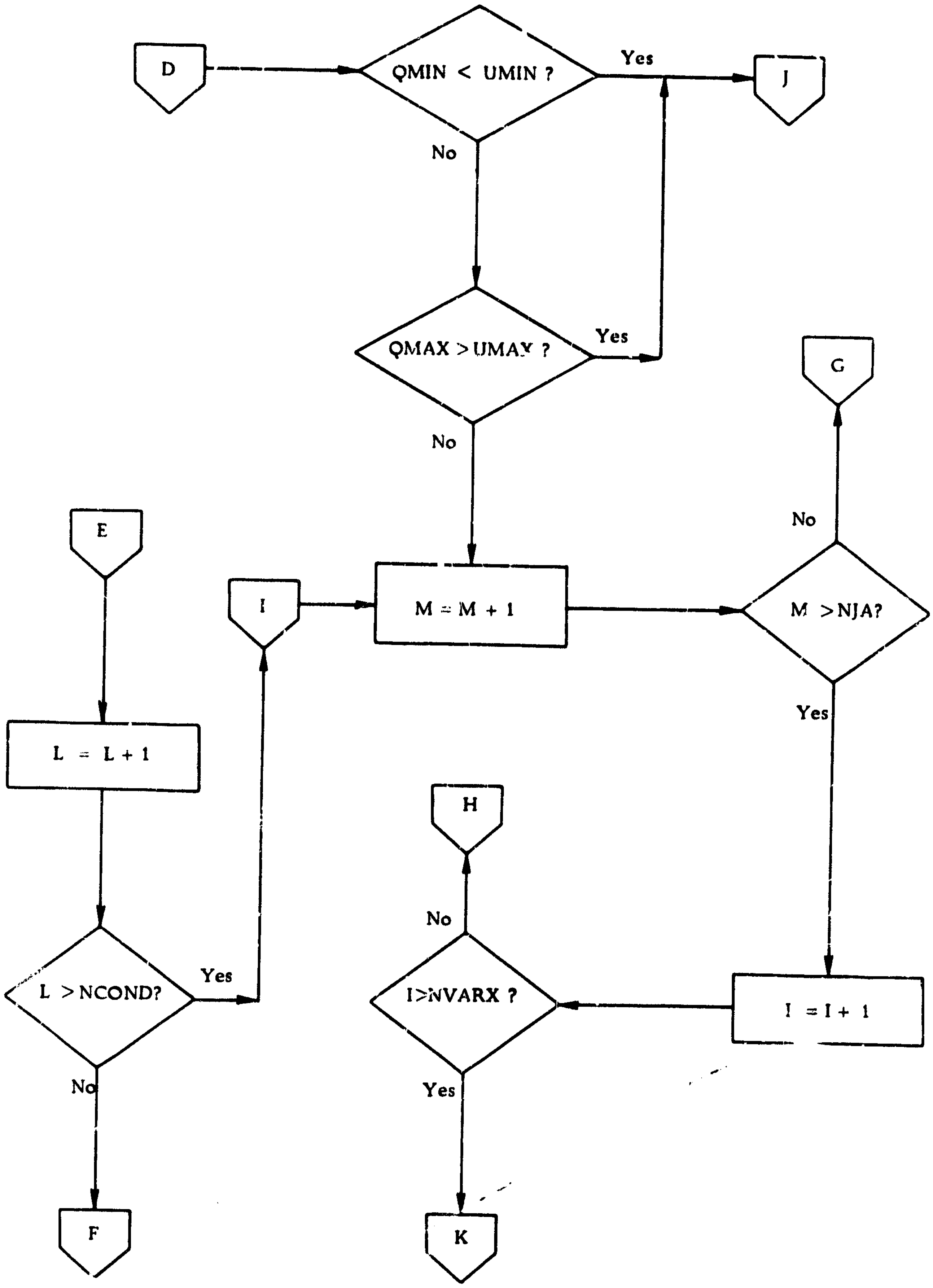
CALL	BNDCHK (*, IFLAG)
*	Return executed on detection of a boundary violation.
IFLAG:	Flag set to non-zero if supression of error message printout is desired.

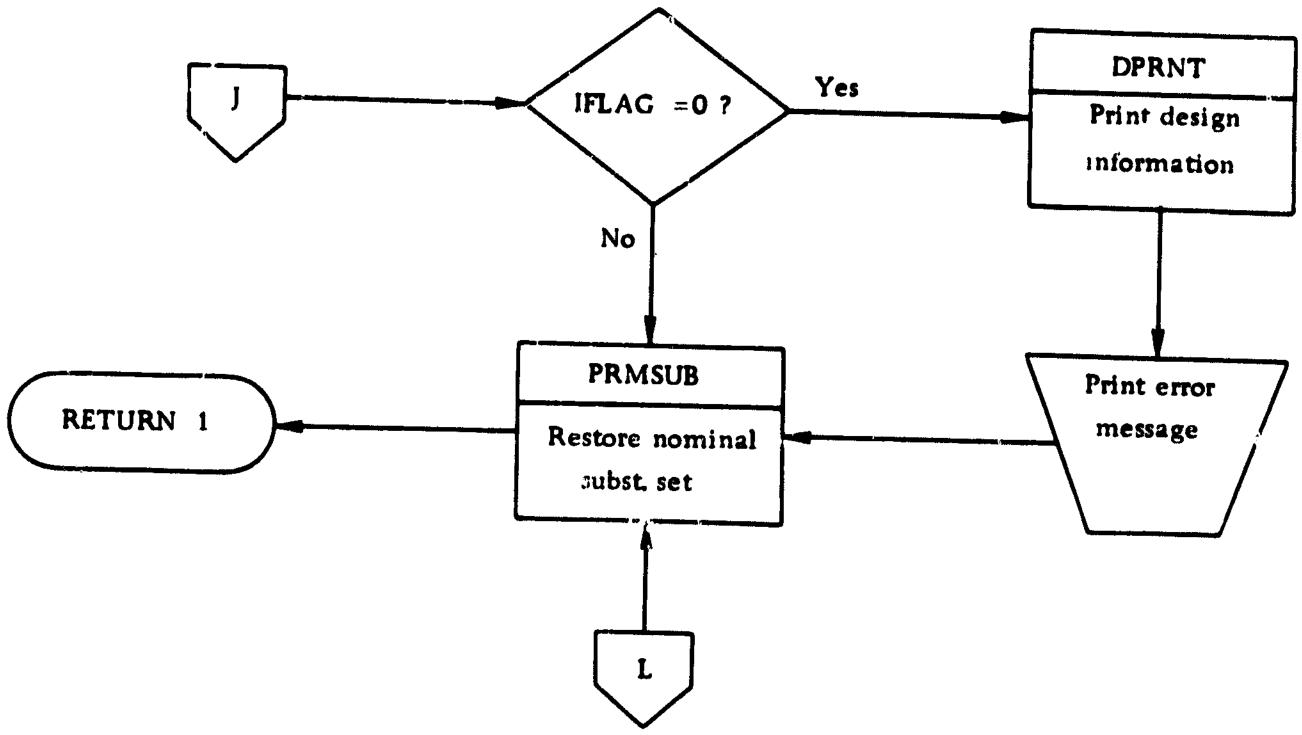
Utility Routines and Common References

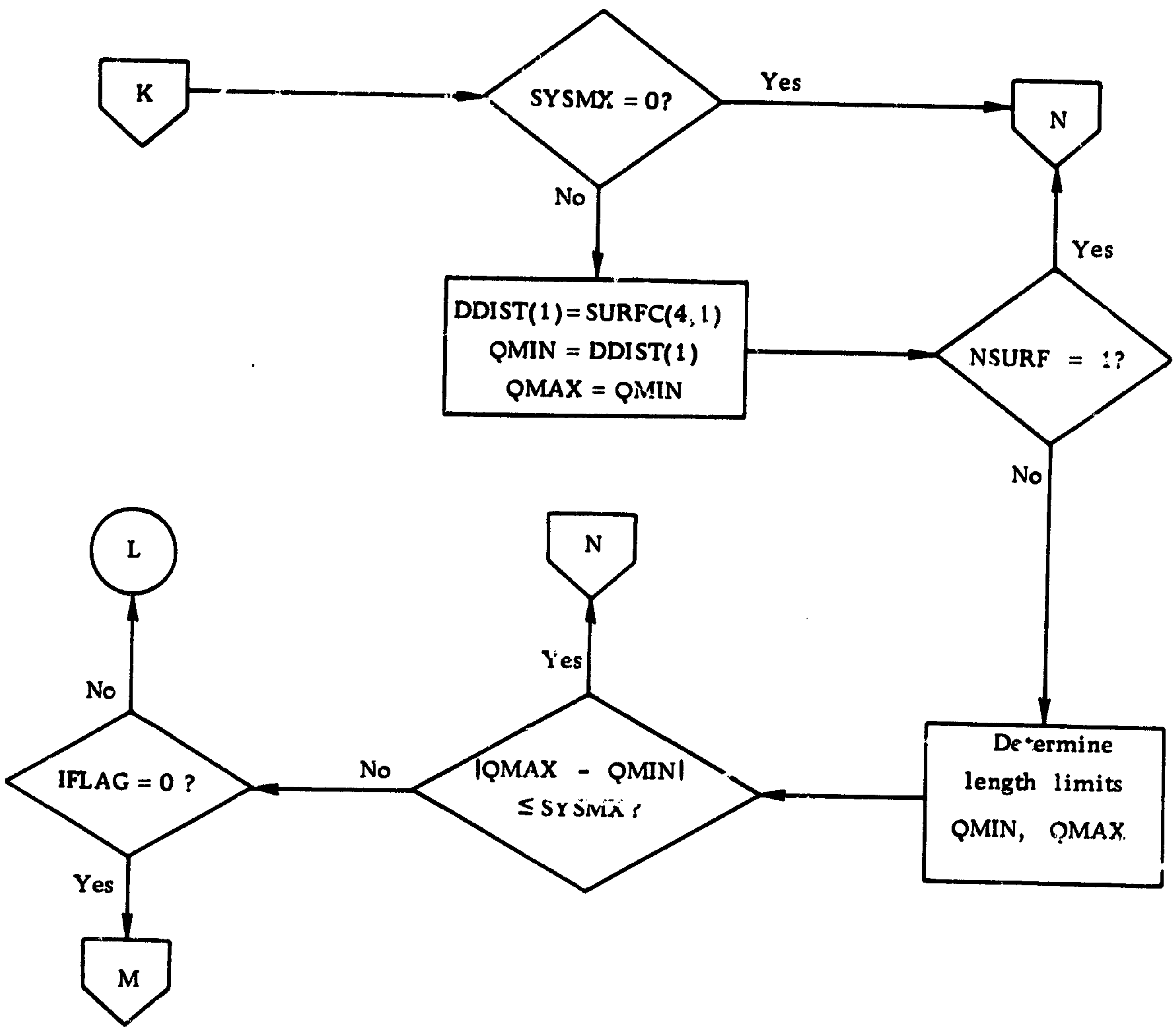
COMMON/DATA/
COMMON/PRNT/
COMMON/AZOBJ/
COMMON/TMDESN/
CORDER
DPRNT
PRMSUB
RAYTR

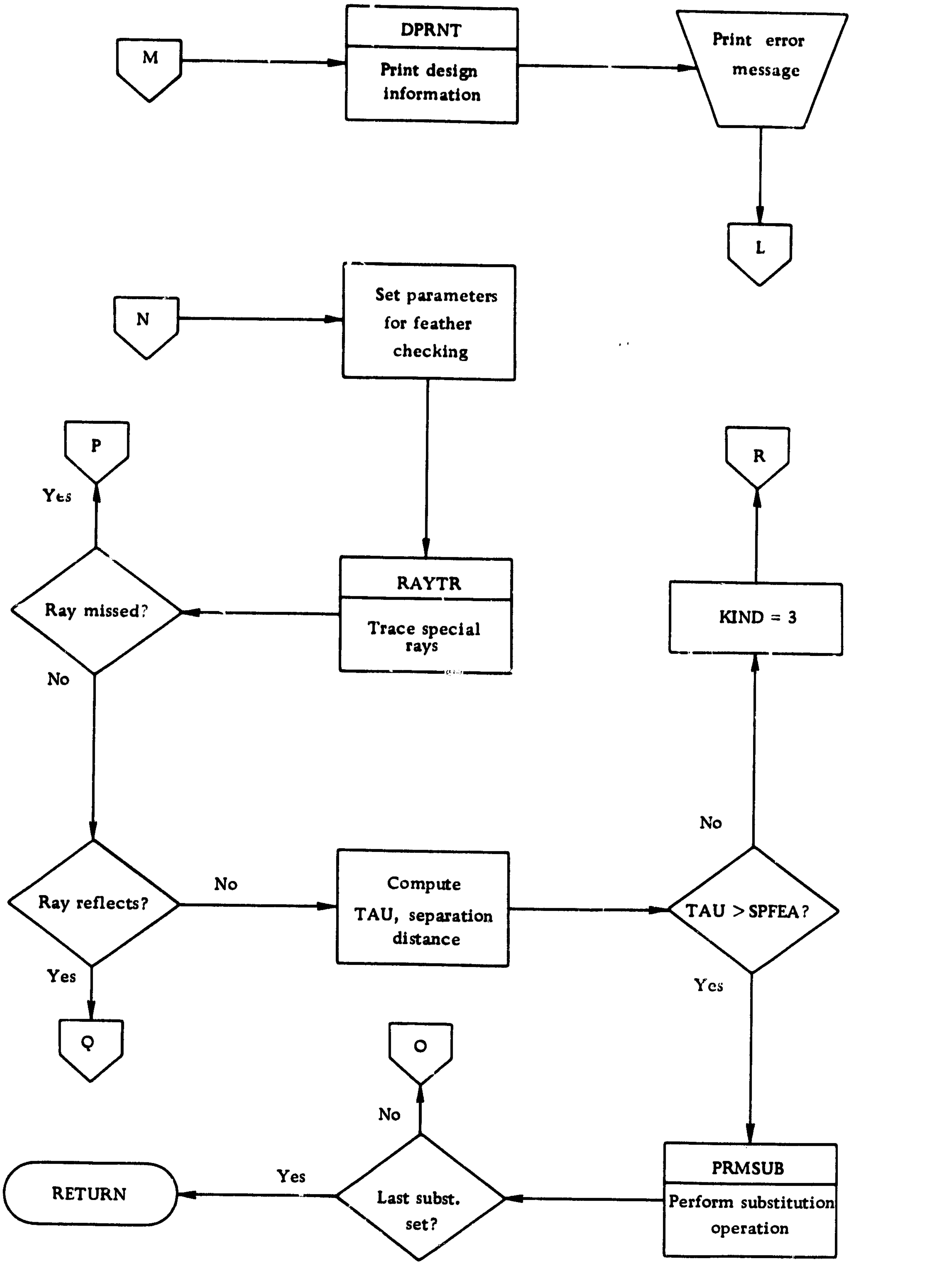

O
SUBROUTINE BNDCHK
NCOND = 0?
Yes
K
No
H
NVARX = NDSGV(1)
I = 1
JI = NDSGV(I+1)
NJA = IABS
(IDESGN(1, JI)
M = 1
G
KMA = IABS
(IDESGN (M+ 3, JI))
U = DATA(KMA)
L = 1
RL=IBNDS(1, L)
RLA = IABS(RL)
UMIN = BOUNDS(2, L)
UMAX = BOUNDS(3, L)
F
B
Yes
RL < 0?
No
KMA = RL ?
Yes
NJ < 0 ?
Yes
No
No
C
E
A

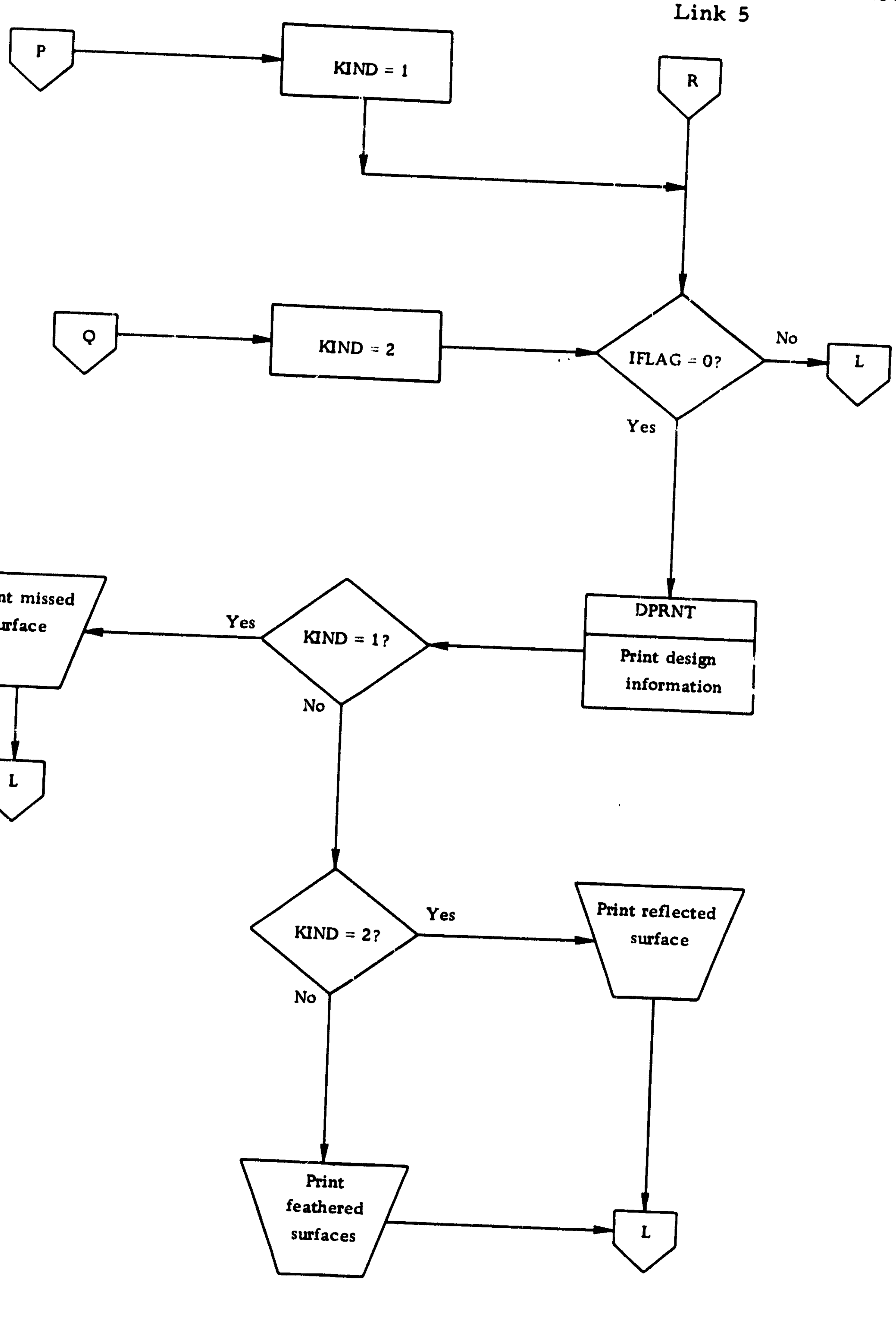

A
UMIN ≤ U≤UMAX?
Yes
I
No
B
UMIN ≤ 1/U ≤ UMAX?
No
J
Yes
I
C
KMA<RLA OR.KMA > (RLA + NC-1)?
Yes
E
No
CORDER
Determine order of indices of refraction
Indices ordered properly?
No
J
Yes
QMIN = DATA(RLA)
QMAX =QMIN
NCLRS = 1?
Yes
No
QMIN set to minimum of indices.
QMAX set to maximum of indices.
D


D
QMIN < UMIN ?
Yes
J
No
QMAX > UMAX ?
Yes
G
No
No
E
I
M = M + 1
M > NJA?
Yes
L = L + 1
H
No
L > NCOND?
Yes
I>NVARX ?
I = I + 1
No
Yes
F
K


J
IFLAG =0 ?
Yes
DPRNT
Print design information
No
Print error message
PRMSUB
Restore nominal subst. set
RETURN 1
L


K
SYSMX = 0?
Yes
N
No
DDIST(1)=SURFC(4,1)
QMIN = DDIST(1)
QMAX = QMIN
Yes
NSURF = 1?
No
Determine length limits QMIN, QMAX
|QMAX - QMIN| ≤ SYSMX ?
Yes
N
No
IFLAG = 0 ?
No
L
Yes
M


M
DPRNT
Print design information
Print error message
L
N
Set parameters for feather checking
P
Yes
R
Ray missed?
RAYTR
Trace special rays
KIND = 3
No
No
Ray reflects?
No
Compute TAU, separation distance
TAU > SPFEA?
Yes
Yes
Q
O
No
RETURN
Yes
Last subst. set?
PRMSUB
Perform substitution operation


P
KIND = 1
R
Q
KIND = 2
IFLAG = 0?
No
L
Yes
DPRNT
Print design information
KIND = 1?
Yes
Print missed surface
L
No
KIND = 2?
Yes
Print reflected surface
No
Print feathered surfaces
L

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)

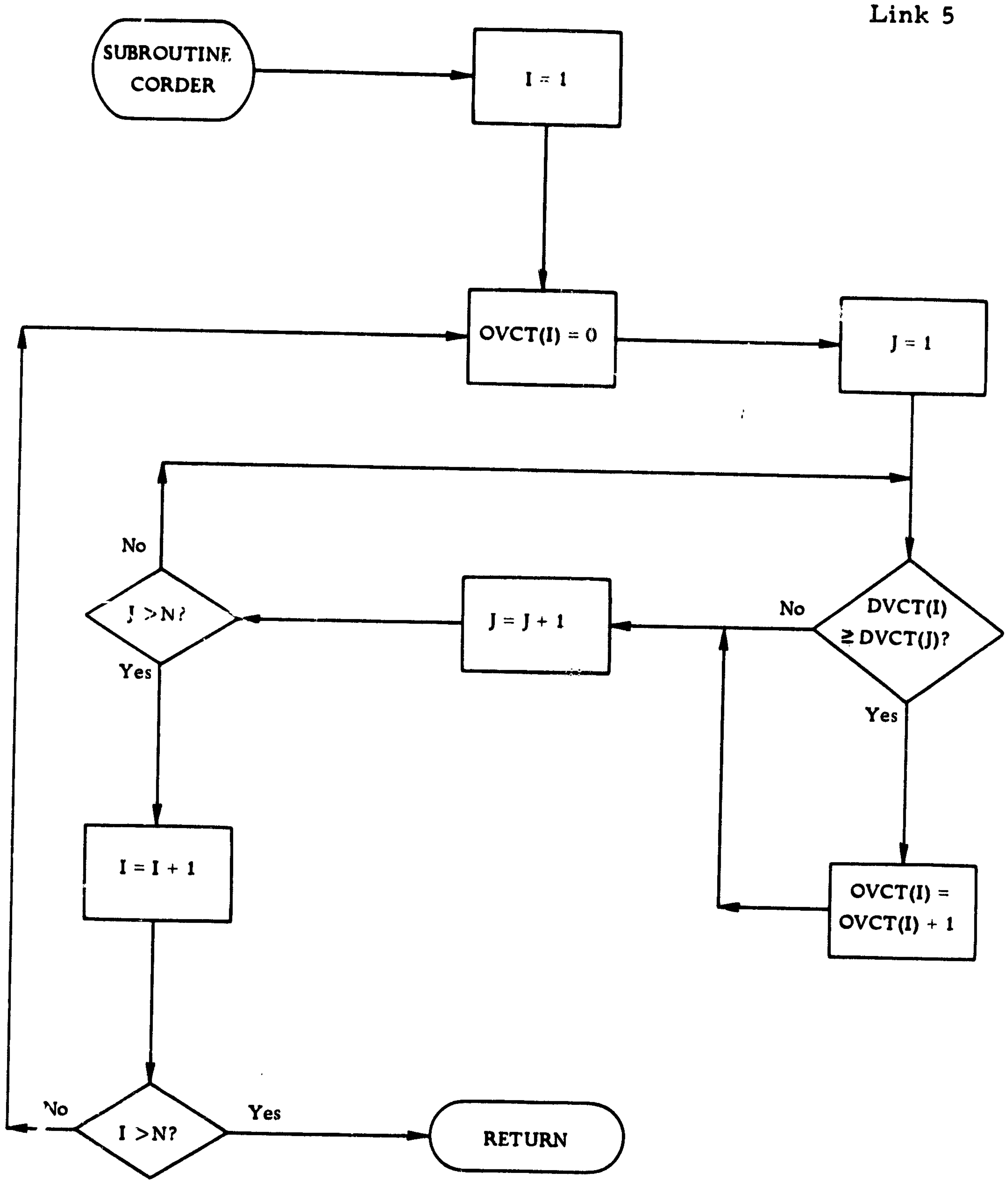
N: Number of colors (indices of refraction).

DVCT: Vector containing the N indices of refraction.

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

Utility Routines and Common References

None


SUBROUTINE CORDER
I = 1
OVCT(I) = 0
J = 1
DVCT(I) ≥ DVCT(J)?
No
Yes
OVCT(I) = OVCT(I) + 1
J = J + 1
J > N?
No
Yes
I = I + 1
I > N?
No
Yes
RETURN

3.5.3 CYCLEZ (CYCLE)

CYCLEZ uses the current set of design variables, the parameter vector $\vec{U}_o$, and $\vec{E}_o = \vec{E}(\vec{U}_o)$ to attempt to compute $\vec{U}^*$ such that $|\vec{E}^*| < |\vec{E}_o|$. 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.

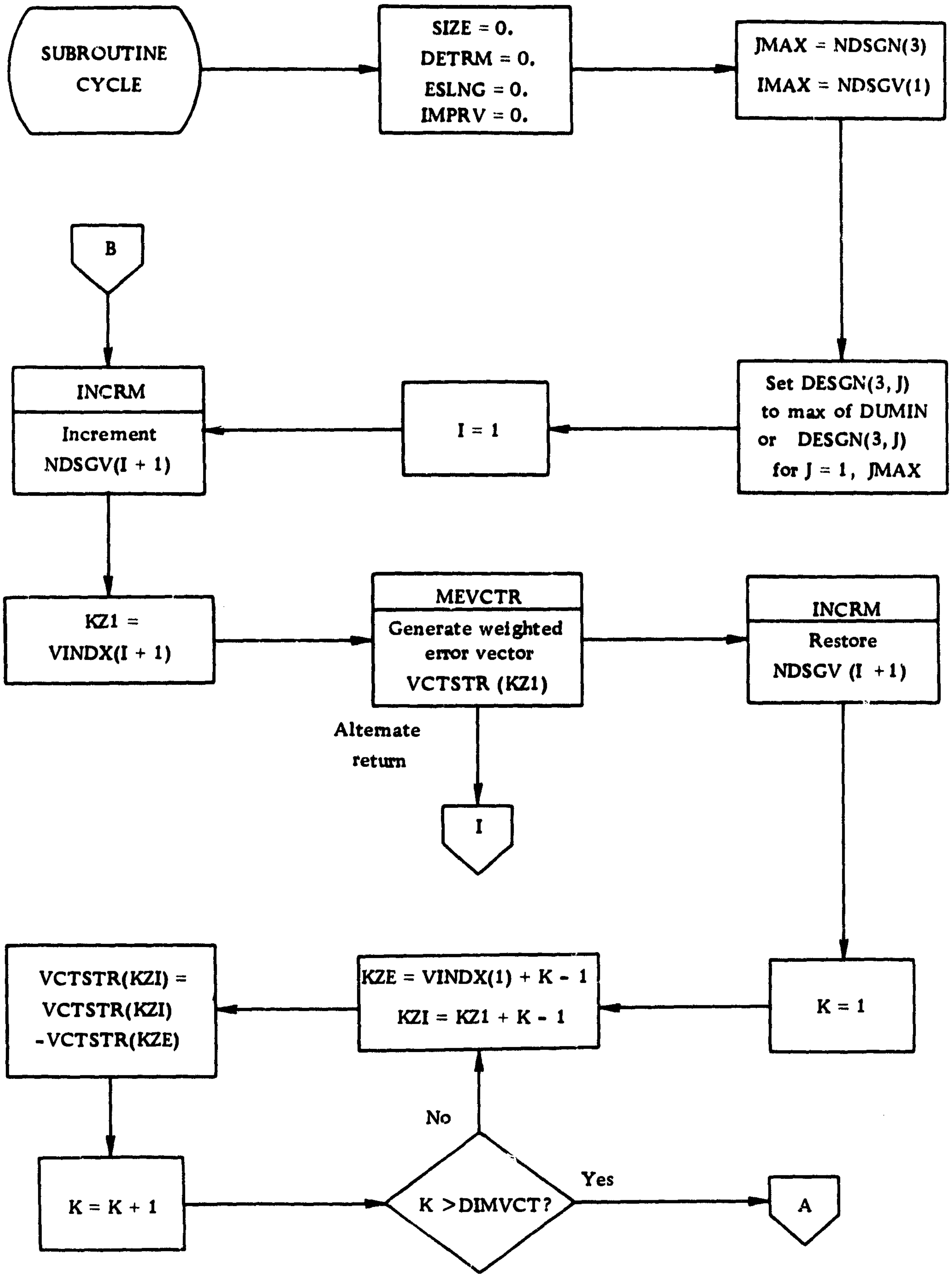
In the case of too many successive failures CYCLEZ executes the alternate return and the design is abandoned. When an improvement has been made the normal return is used. If a negative improvement has been made the parameters are restored so that the design may proceed on the next step.

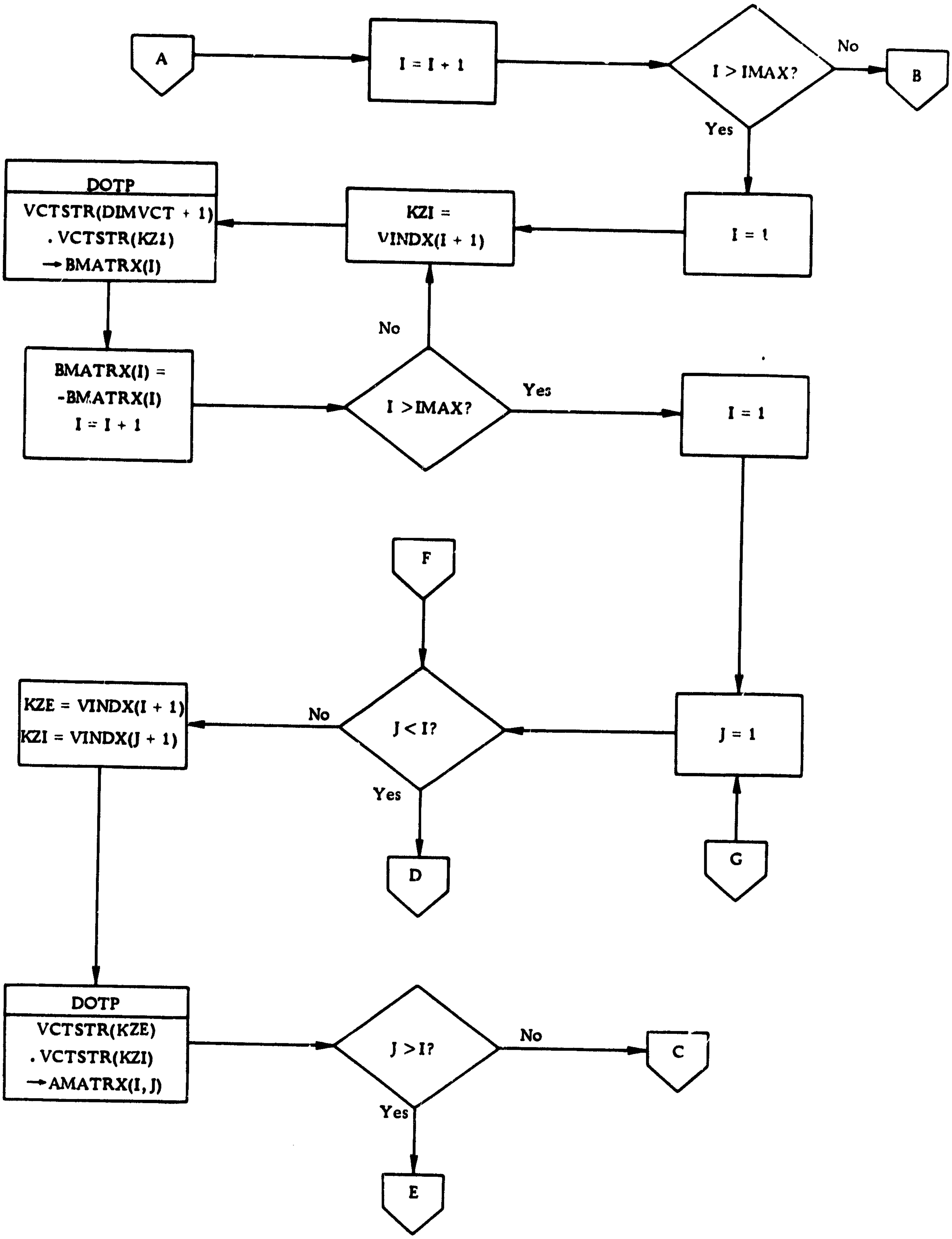
Calling Sequence

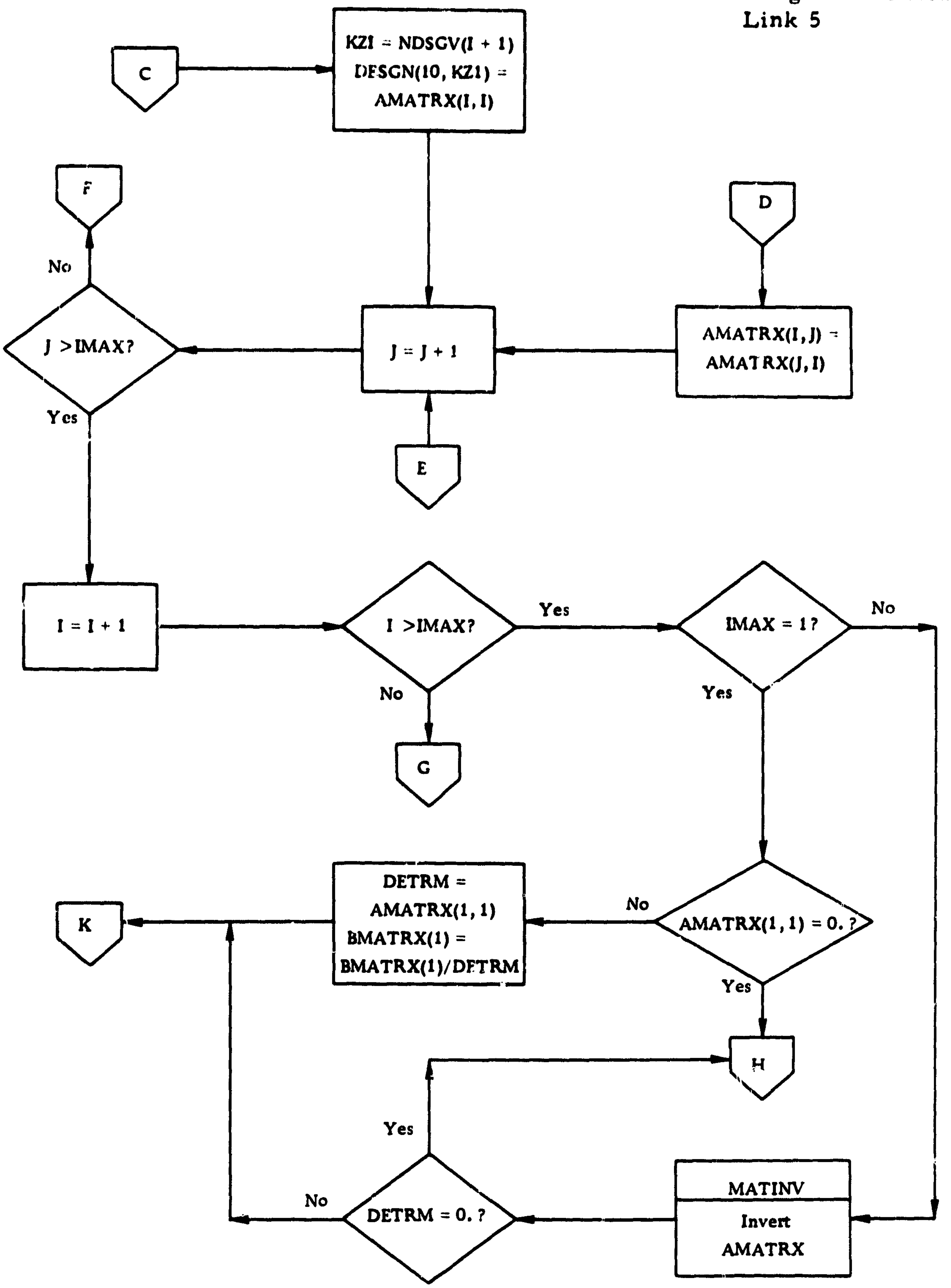
CALL	CYCLE (*)
*	Alternate return executed when too many successive failures have occurred.

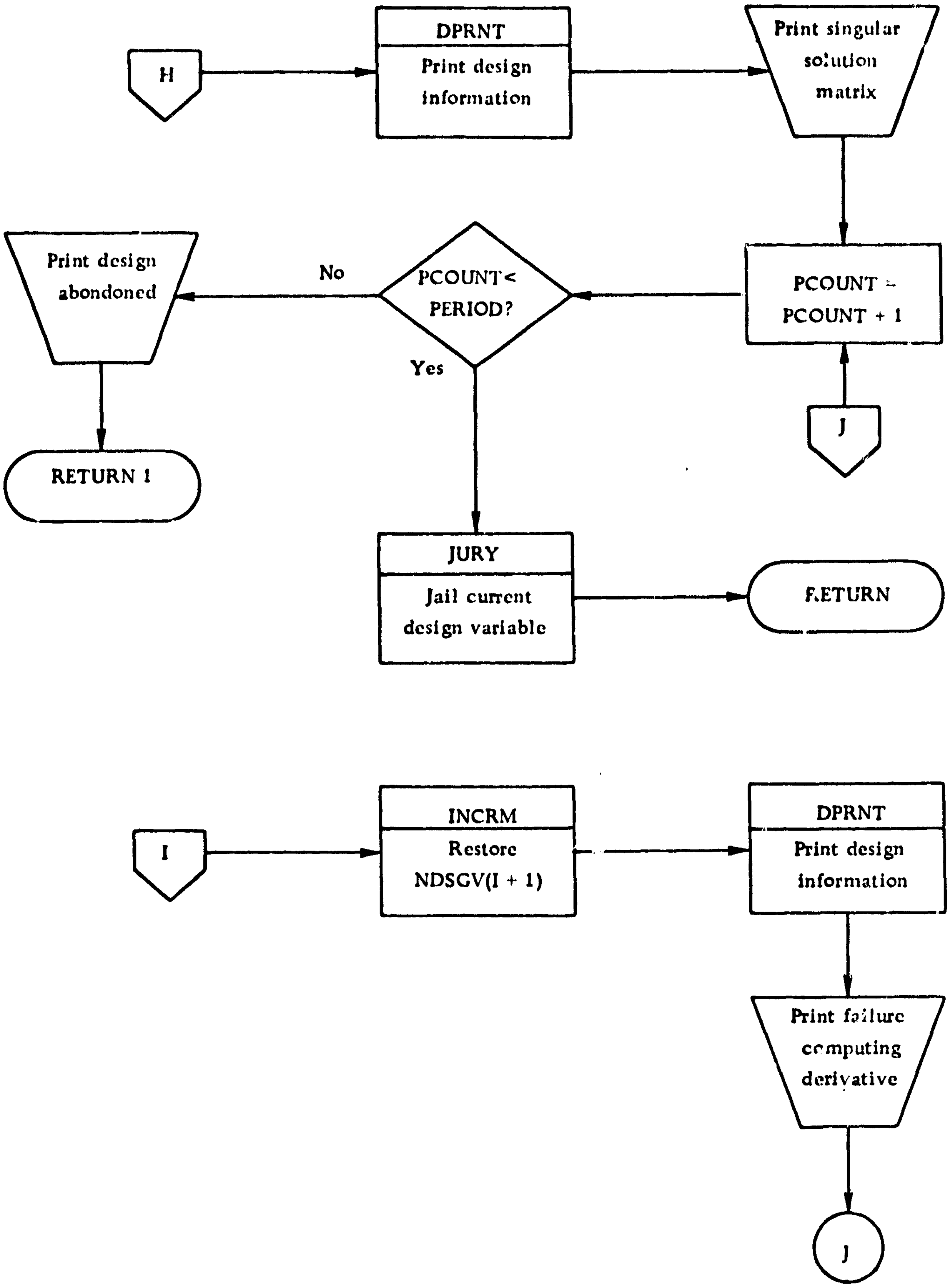
Utility Routines and Common References

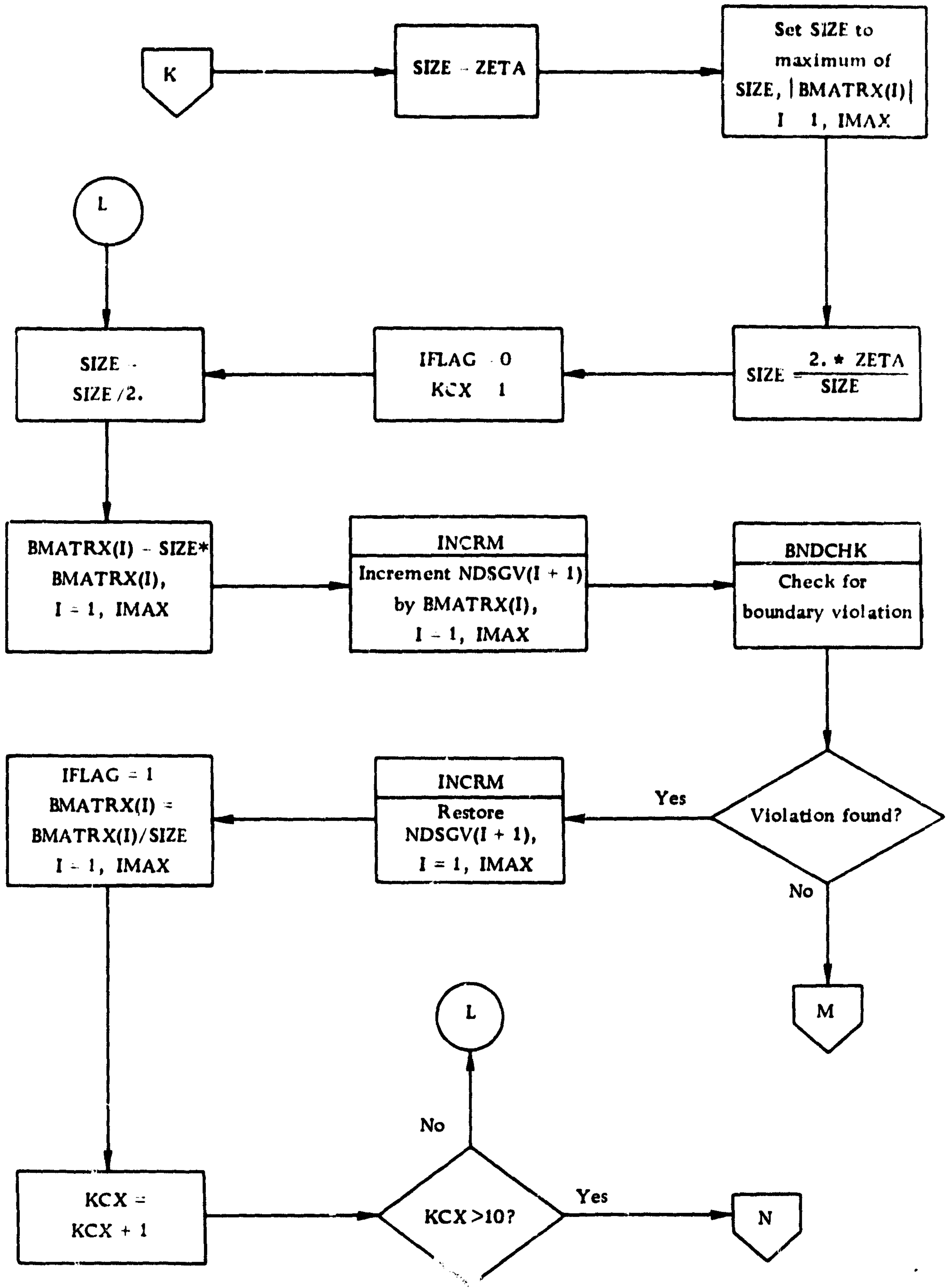
COMMON/DATA/
COMMON/PRNT/
COMMON/TMDESN/
BNDCHK
DOTP
DPRNT (EPRNT)
GETRHO
GOOF
INCRM
JURY
MATINV
MEVCTR
MODDL
PRMSUB

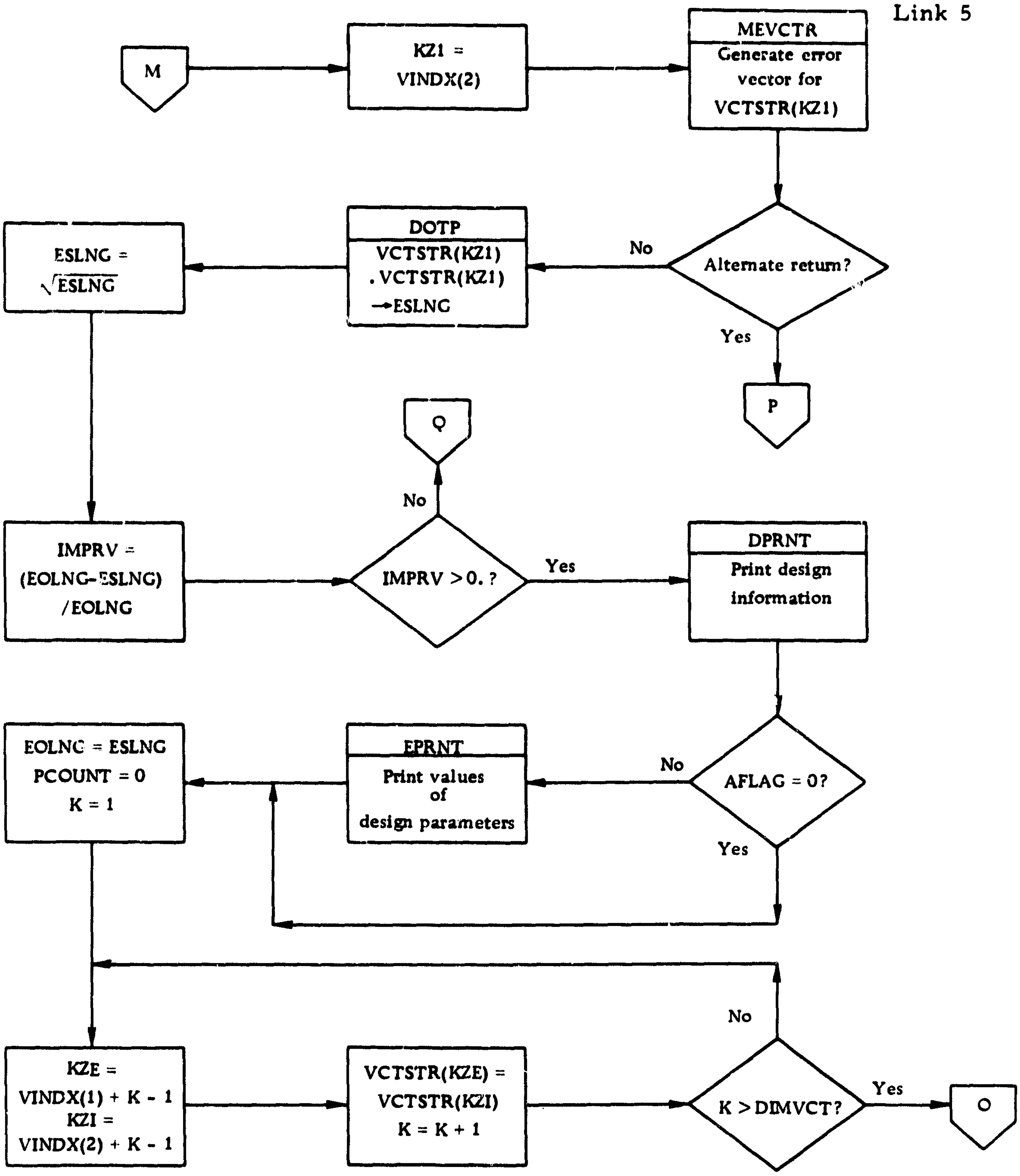

SUBROUTINE CYCLE
SIZE = 0.
DETRM = 0.
ESLNG = 0.
IMPRV = 0.
JMAX = NDSGN(3)
IMAX = NDSGV(1)
B
Set DESGN(3, J) to max of DUMIN or DESGN(3, J) for J = 1, JMAX
I = 1
INCRM
Increment NDSGV(I + 1)
KZ1 = VINDX(I + 1)
MEVCTR
Generate weighted error vector VCTSTR (KZ1)
INCRM
Restore NDSGV (I +1)
Alternate return
I
K = 1
KZE = VINDX(1) + K - 1
KZI = KZ1 + K - 1
VCTSTR(KZI) = VCTSTR(KZI) -VCTSTR(KZE)
K = K + 1
No
K >DIMVCT?
Yes
A

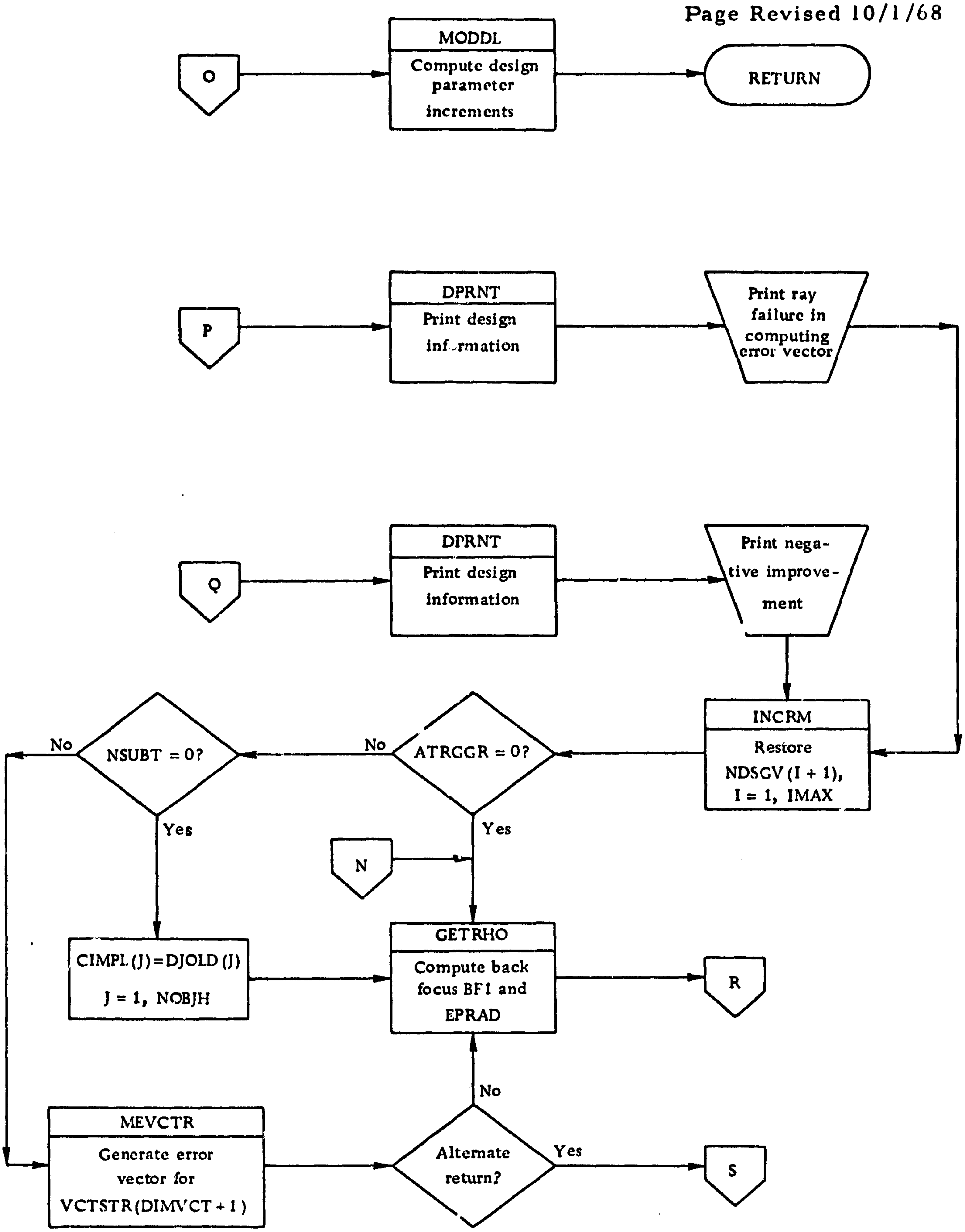

A
I = I + 1
I > IMAX?
No
B
Yes
I = 1
KZI =
VINDX(I + 1)
DOTP
VCTSTR(DIMVCT + 1)
. VCTSTR(KZI)
→ BMATRX(I)
BMATRX(I) =
-BMATRX(I)
I = I + 1
I > IMAX?
No
Yes
I = 1
F
J < I?
No
Yes
D
J = 1
G
KZE = VINDX(I + 1)
KZI = VINDX(J + 1)
DOTP
VCTSTR(KZE)
. VCTSTR(KZI)
→ AMATRX(I, J)
J > I?
No
C
Yes
E

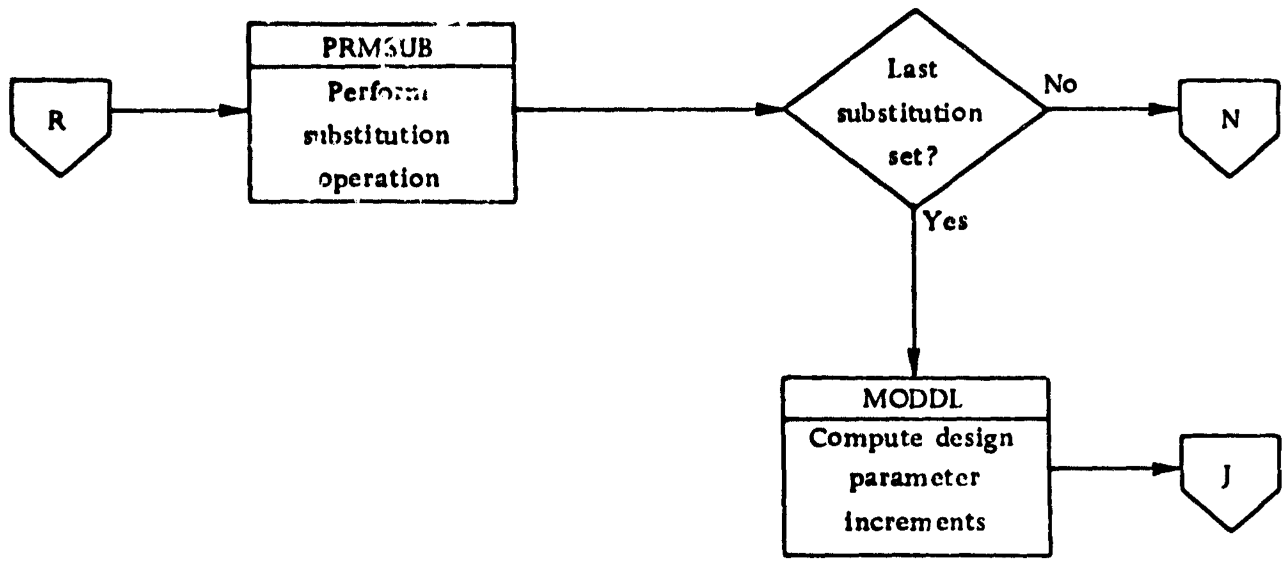

C
KZ1 = NDSGV(I + 1)
DESGN(10, KZ1) =
AMATRX(I, I)
F
No
J >IMAX?
Yes
J = J + 1
D
AMATRX(I, J) =
AMATRX(J, I)
E
I = I + 1
I >IMAX?
Yes
No
G
IMAX = 1?
No
Yes
K
DETRM =
AMATRX(1, 1)
BMATRX(1) =
BMATRX(1)/DETRM
No
AMATRX(1, 1) = 0. ?
Yes
H
Yes
No
DETRM = 0. ?
MATINV
Invert
AMATRX

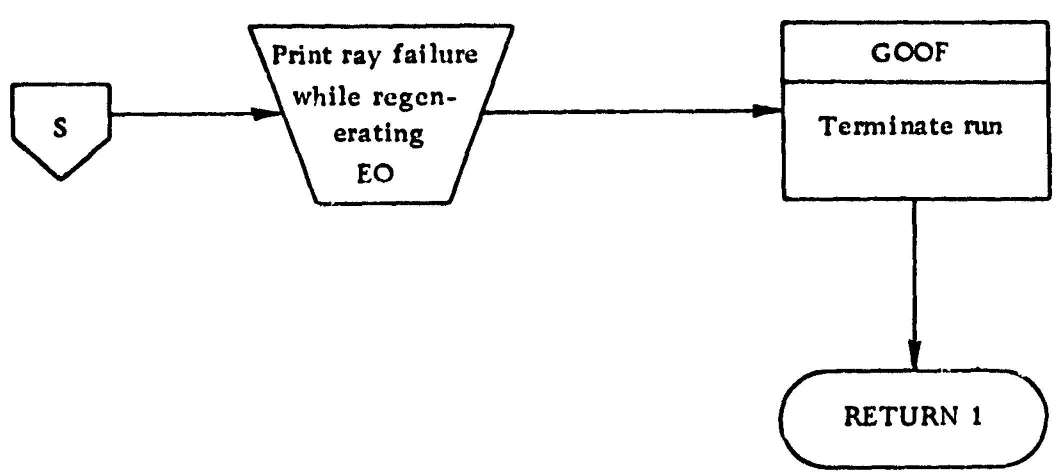

H
DPRNT
Print design information
Print singular solution matrix
PCOUNT = PCOUNT + 1
J
PCOUNT< PERIOD?
No
Yes
Print design abondoned
RETURN 1
JURY
Jail current design variable
RETURN
I
INCRM
Restore NDSGV(I + 1)
DPRNT
Print design information
Print failure computing derivative
J


K
SIZE = ZETA
Set SIZE to maximum of SIZE, |BMATRX(I)| I = 1, IMAX
L
SIZE = 2. * ZETA / SIZE
IFLAG = 0
KCX = 1
SIZE = SIZE/2.
BMATRX(I) = SIZE* BMATRX(I), I = 1, IMAX
INCRM
Increment NDSGV(I + 1) by BMATRX(I), I = 1, IMAX
BNDCHK
Check for boundary violation
Violation found?
Yes
No
M
INCRM
Restore NDSGV(I + 1), I = 1, IMAX
IFLAG = 1
BMATRX(I) = BMATRX(I)/SIZE
I = 1, IMAX
KCX = KCX + 1
KCX >10?
Yes
N
No
L


M
KZ1 = VINDX(2)
MEVCTR
Generate error vector for VCTSTR(KZ1)
Alternate return?
No
Yes
P
DOTP
VCTSTR(KZ1) . VCTSTR(KZ1) →ESLNG
ESLNG = √ESLNG
IMPRV = (EOLNG-ESLNG) /EOLNG
IMPRV > 0. ?
No
Q
Yes
DPRNT
Print design information
AFLAG = 0?
No
Yes
EPRNT
Print values of design parameters
EOLNG = ESLNG
PCOUNT = 0
K = 1
KZE = VINDX(1) + K - 1
KZI = VINDX(2) + K - 1
VCTSTR(KZE) = VCTSTR(KZI)
K = K + 1
K > DIMVCT?
No
Yes
O


O
MODDL
Compute design parameter increments
RETURN
P
DPRNT
Print design information
Print ray failure in computing error vector
Q
DPRNT
Print design information
Print negative improvement
INCRM
Restore NDSGV (I + 1), I = 1, IMAX
ATRGGR = 0?
No
Yes
NSUBT = 0?
No
Yes
N
GETRHO
Compute back focus BF1 and EPRAD
R
CIMPL (J) = DJOLD (J)
J = 1, NOBJH
MEVCTR
Generate error vector for VCTSTR (DIMVCT + 1)
Alternate return?
Yes
No
S


R
PRMSUB
Perform substitution operation
Last substitution set?
No
N
Yes
MODDL
Compute design parameter increments
J


S
Print ray failure while regen- erating EO
GOOF
Terminate run
RETURN 1

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.

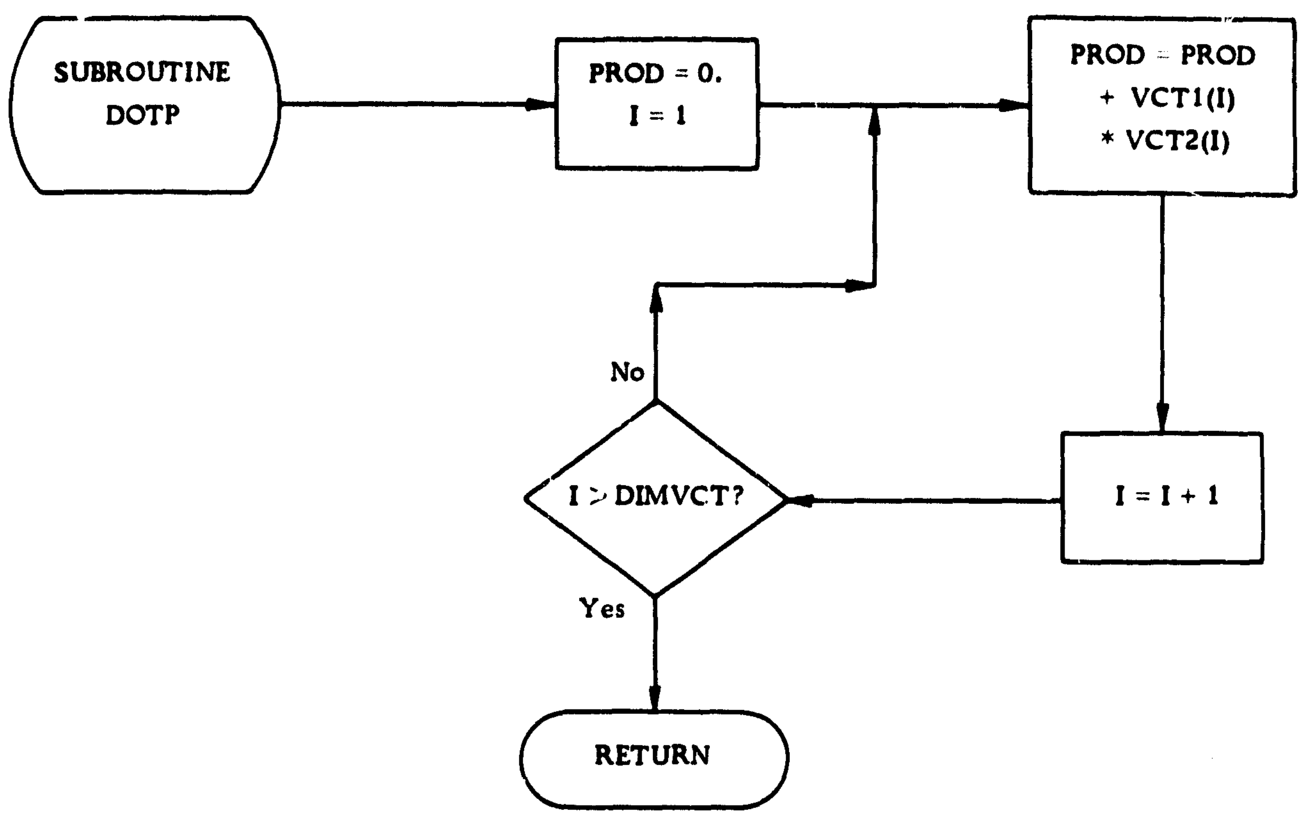
Calling Sequence

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/


SUBROUTINE DOTP
PROD = 0.
I = 1
PROD = PROD
+ VCT1(I)
* VCT2(I)
No
I > DIMVCT?
I = I + 1
Yes
RETURN

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.

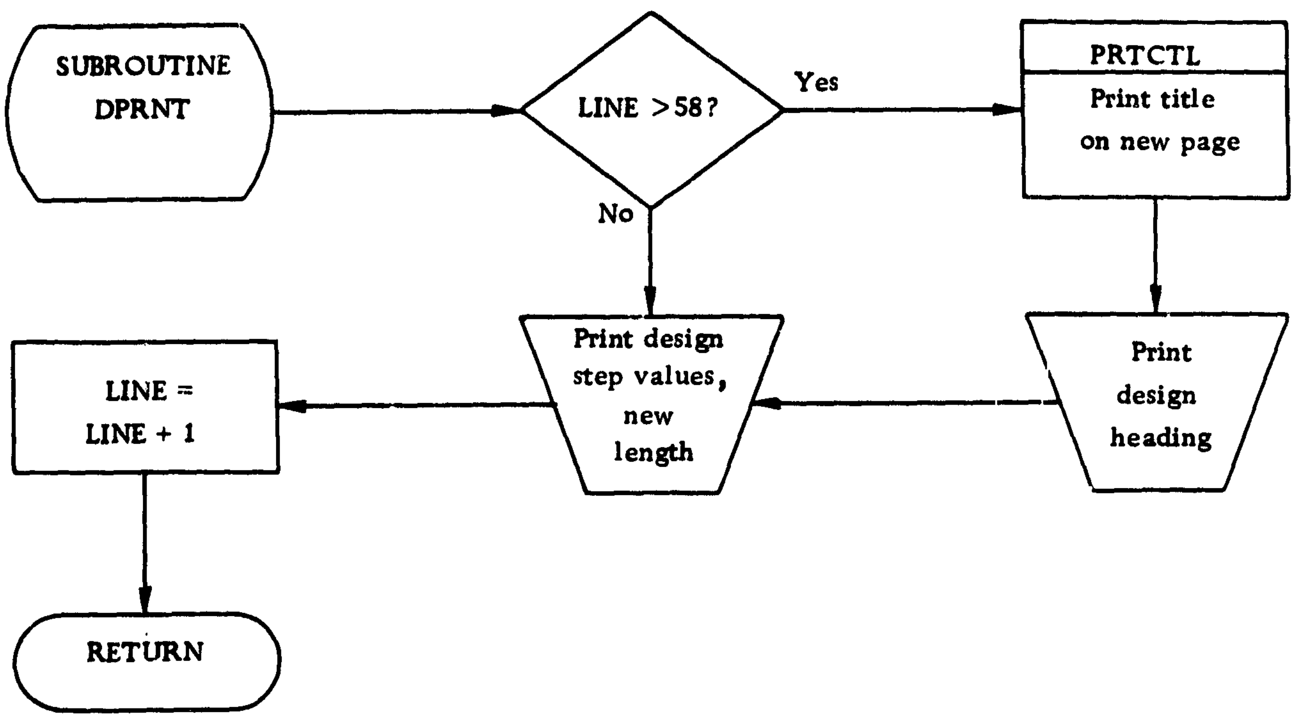
In EPRNT the current values of all the design parameters for the current design step are printed. Up to six parameters for each design variable may be printed per line with one line for each design variable.

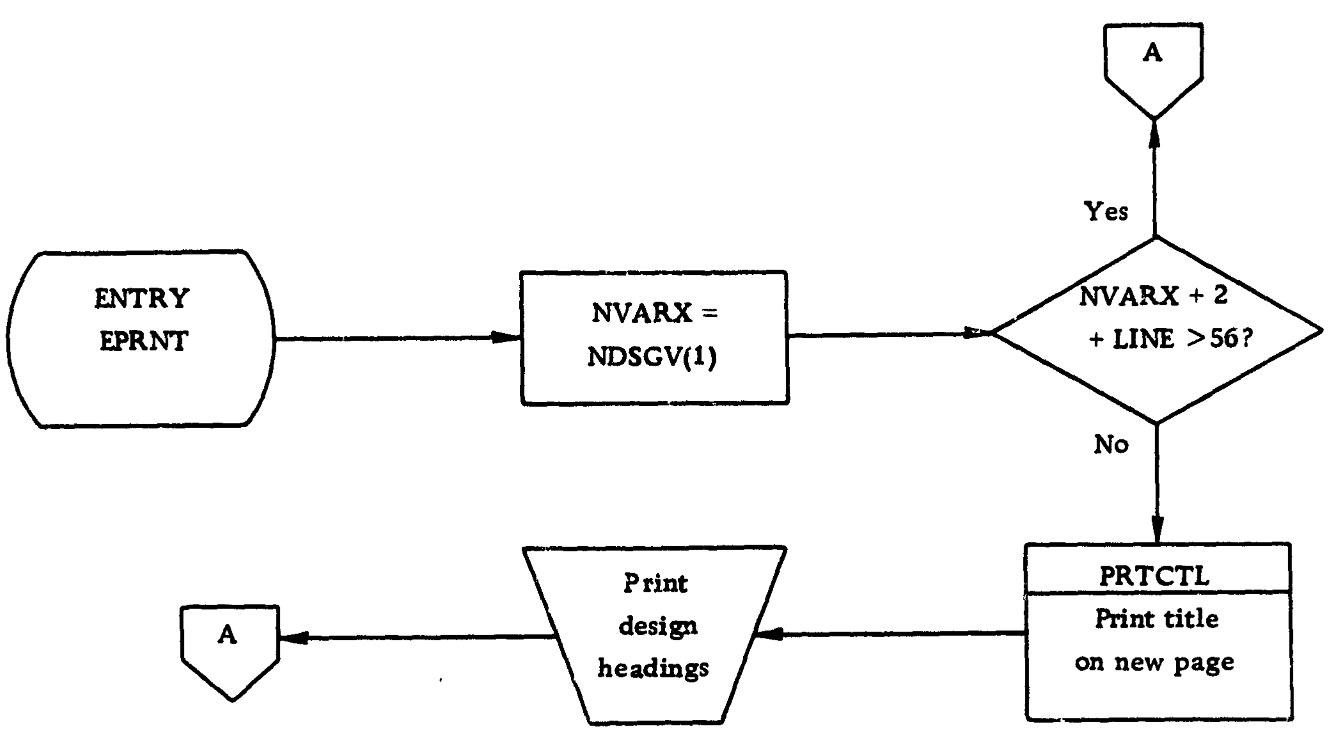
Calling Sequence

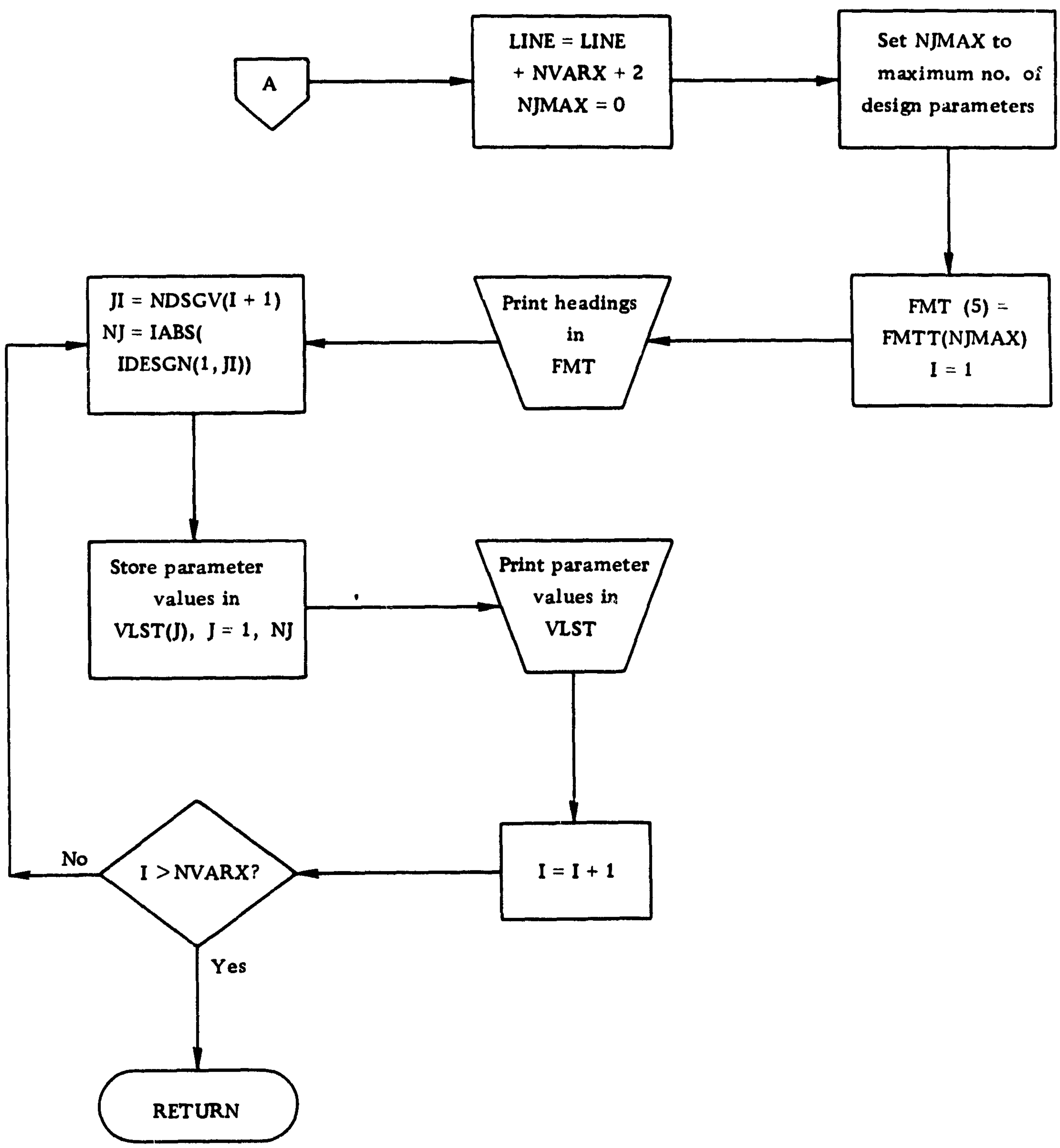
CALL DPRNT
CALL EPRNT

Utility Routines and Common References

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


SUBROUTINE DPRNT
LINE >58?
Yes
No
PRTCTL
Print title on new page
Print design heading
Print design step values, new length
LINE = LINE + 1
RETURN


ENTRY EPRNT
NVARX = NDSGV(1)
NVARX + 2 + LINE >56?
Yes
A
No
PRTCTL
Print title on new page
Print design headings
A


A
LINE = LINE + NVARX + 2
NJMAX = 0
Set NJMAX to maximum no. of design parameters
FMT (5) = FMTT(NJMAX)
I = 1
Print headings in FMT
JI = NDSGV(I + 1)
NJ = IABS(IDESGN(1, JI))
Store parameter values in VLST(J), J = 1, NJ
Print parameter values in VLST
I = I + 1
I > NVARX?
No
Yes
RETURN

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:

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

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.

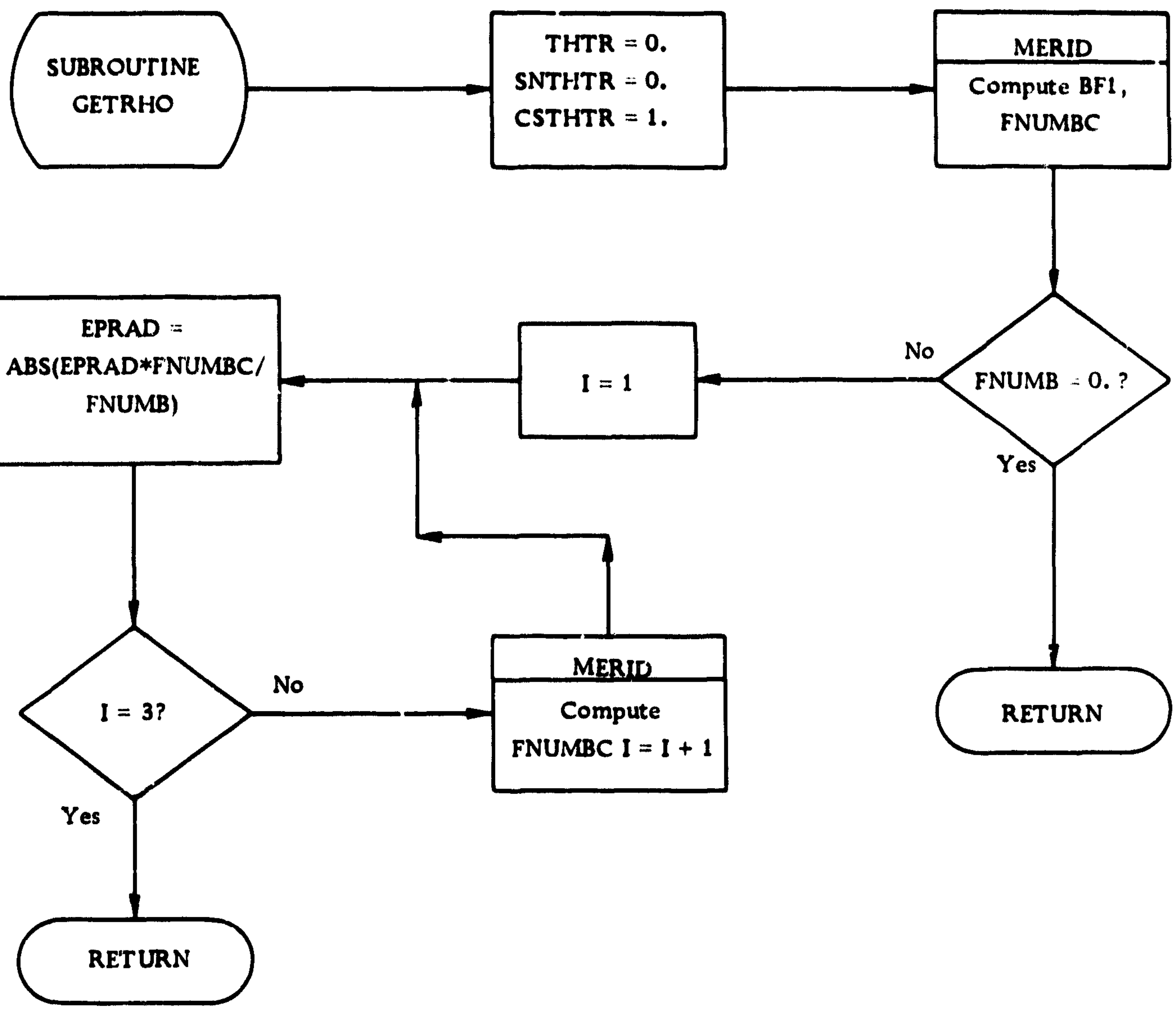
EPRAD: Recomputed value for the radius of the entrance pupil provided FNUMB is non-zero.

Utility Routines and Common References

COMMON/DATA/

COMMON/AZOBJ/

MERID


SUBROUTINE GETRHO
THTR = 0.
SNTHTR = 0.
CSTHTR = 1.
MERID
Compute BF1, FNUMBC
FNUMB = 0. ?
No
Yes
RETURN
I = 1
EPRAD = ABS(EPRAD*FNUMBC/ FNUMB)
I = 3?
No
Yes
MERID
Compute FNUMBC I = I + 1
RETURN

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:

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

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)

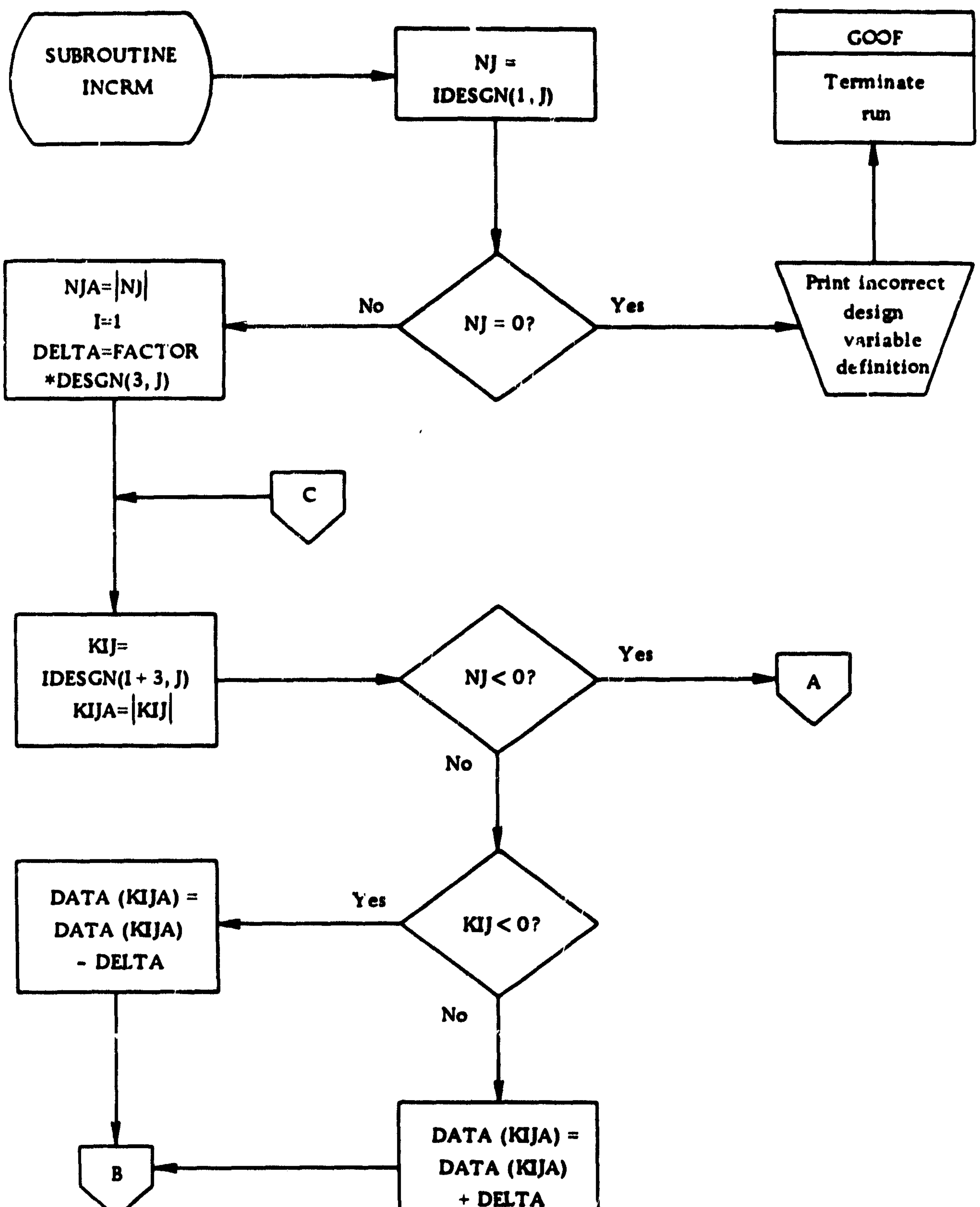
J: Design variable number specifying the design variable to be incremented.

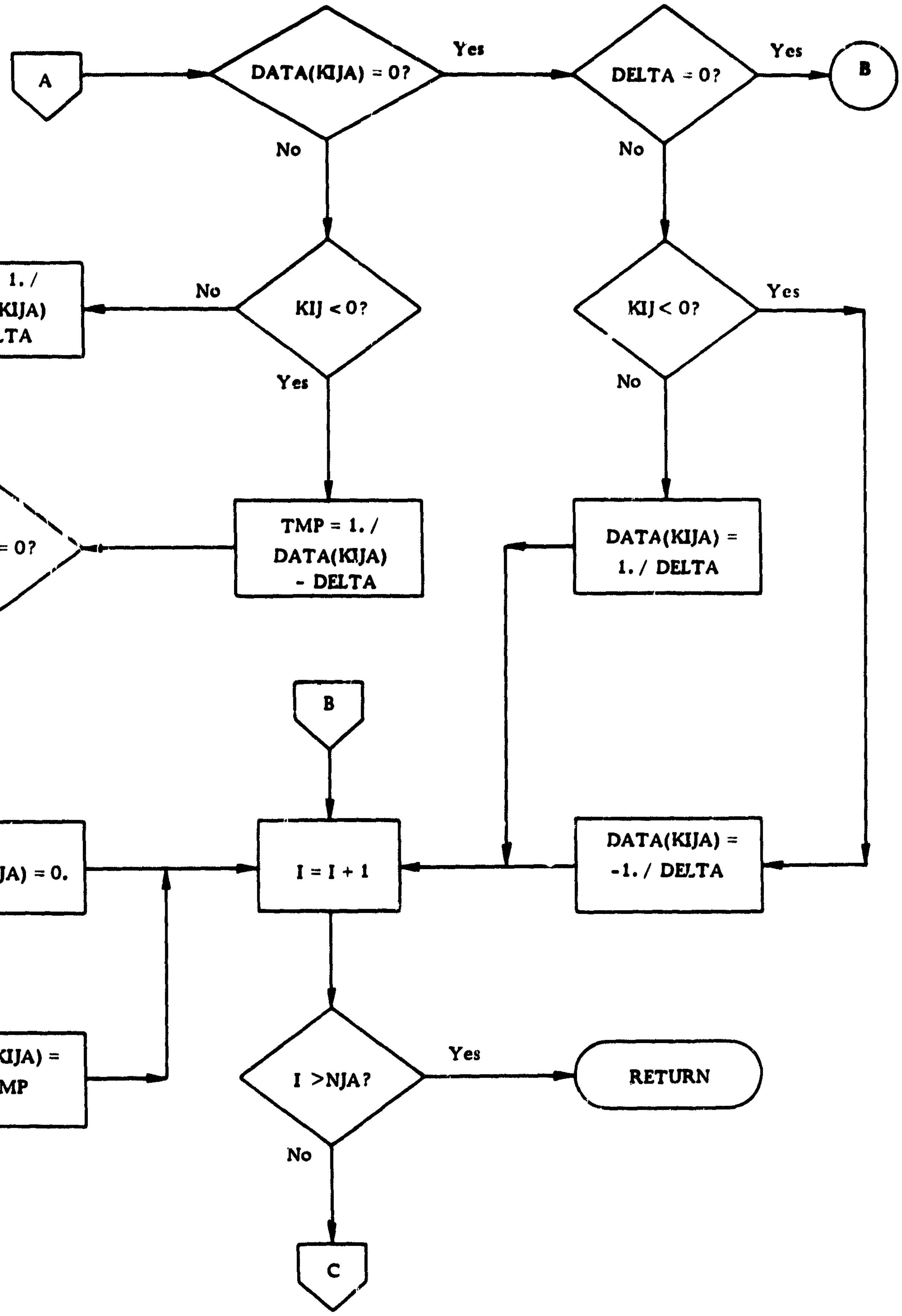
FACTOR: The multiplier λ which is applied to ΔV_j before incrementation.

Utility Routines and Common References

COMMON/DATA/

GOOF


SUBROUTINE INCRM
NJ = IDESGN(1, J)
GOOF
Terminate run
NJA=|NJ|
I=1
DELTA=FACTOR *DESGN(3, J)
No
NJ = 0?
Yes
Print incorrect design variable definition
C
KIJ= IDESGN(I + 3, J)
KIJA=|KIJ|
NJ < 0?
Yes
A
No
DATA (KIJA) = DATA (KIJA) - DELTA
Yes
KIJ < 0?
No
B
DATA (KIJA) = DATA (KIJA) + DELTA


A
DATA(KIJA) = 0?
Yes
DELTA = 0?
Yes
B
No
No
TMP = 1./
DATA(KIJA)
+ DELTA
No
KIJ < 0?
Yes
KIJ < 0?
Yes
No
No
TMP = 0?
TMP = 1./
DATA(KIJA)
- DELTA
DATA(KIJA) =
1./ DELTA
Yes
B
DATA(KIJA) = 0.
I = I + 1
DATA(KIJA) =
-1./ DELTA
DATA(KIJA) =
1./TMP
I >NJA?
Yes
RETURN
No
C

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 mades 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.

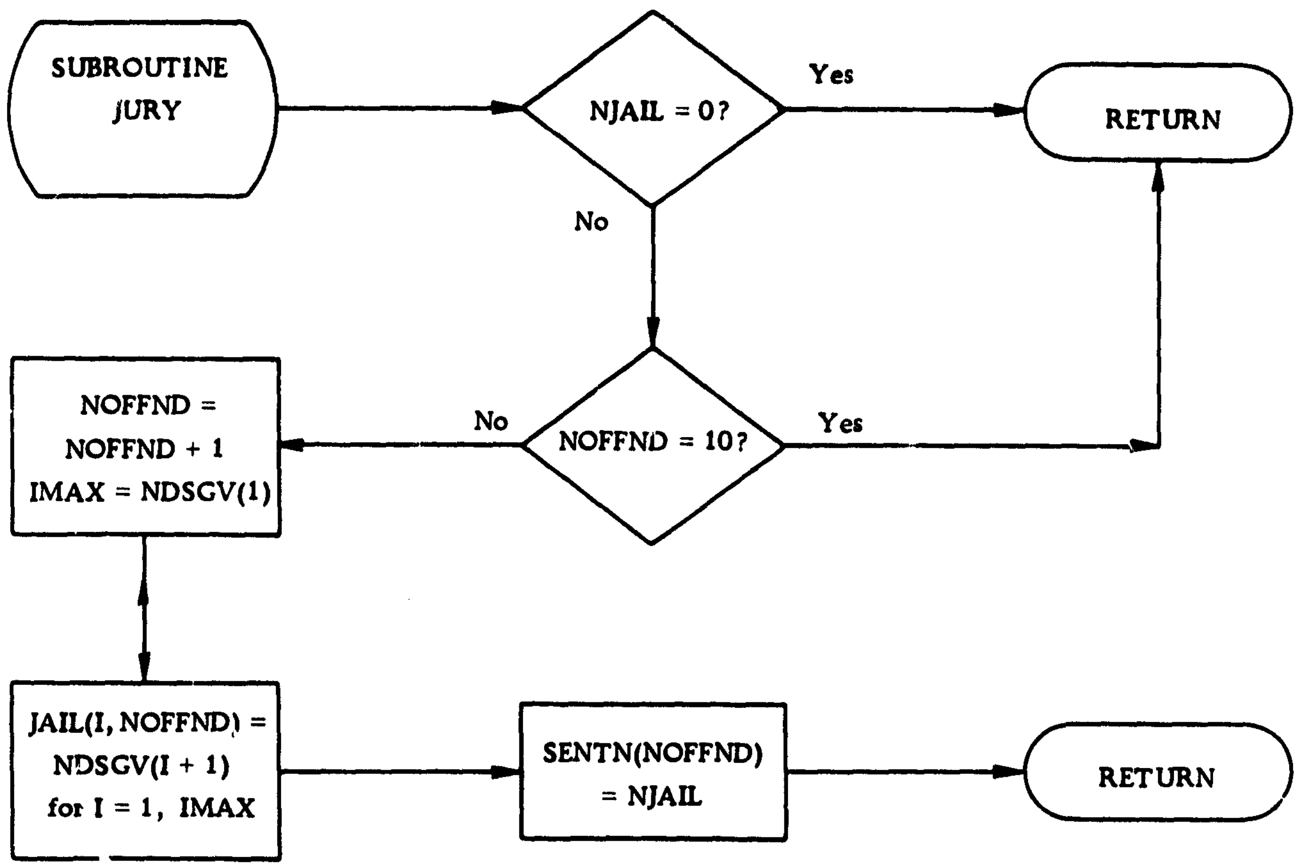
Calling Sequence

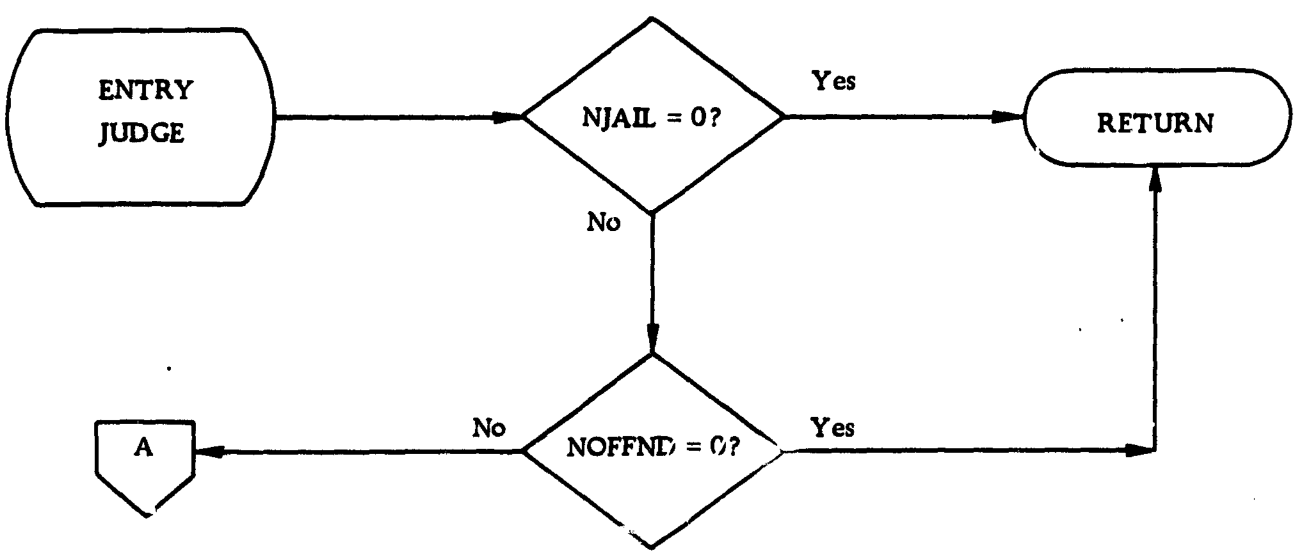
CALL	JURY
CALL	JUDGE (*)
*	Is the alternate return taken if the sentence of a jailed design variable combination is greater than zero.

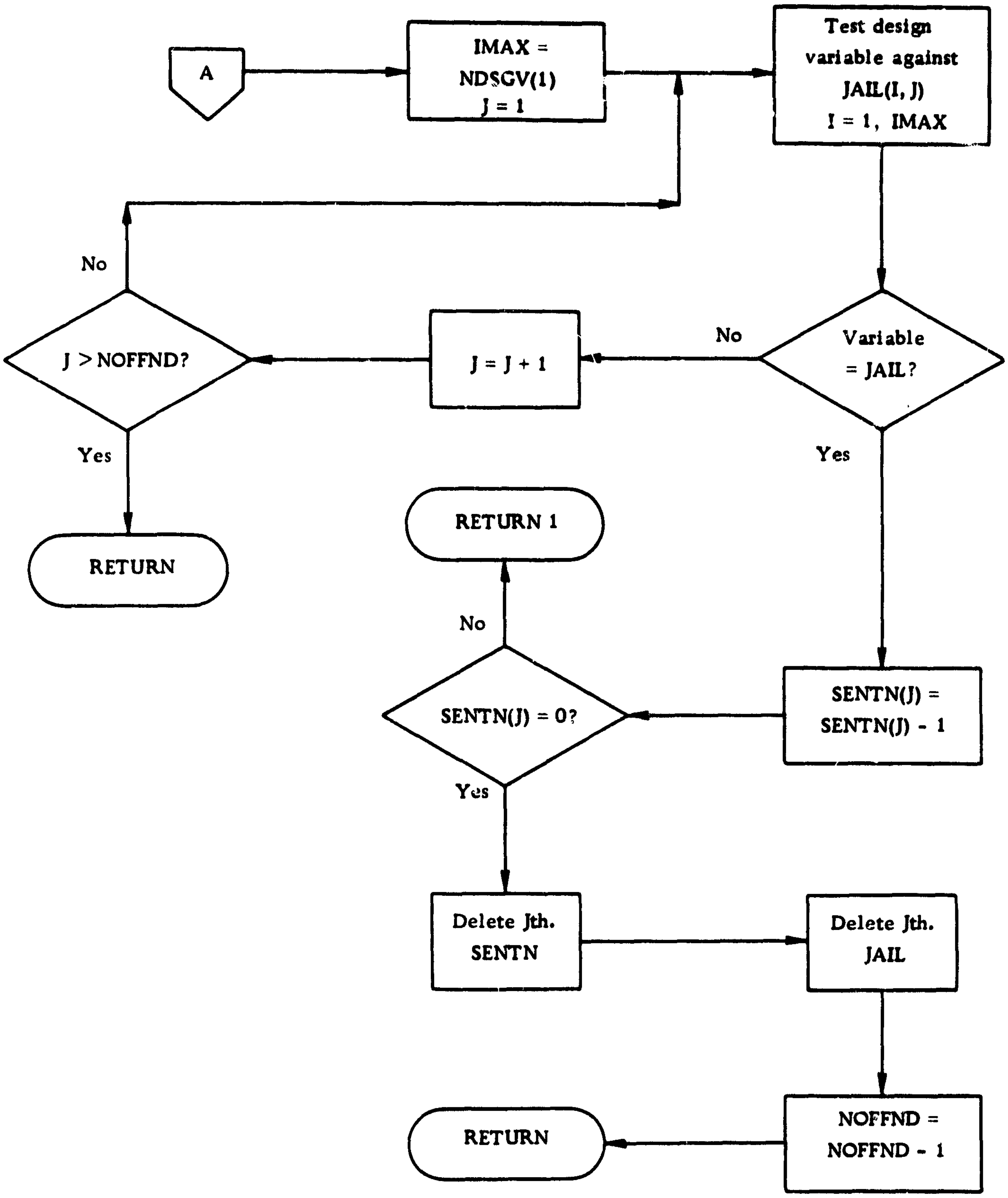
Utility Routines and Common References

COMMON/DATA/

COMMON/TMDESN/


SUBROUTINE JURY
NJAIL = 0?
Yes
RETURN
No
NOFFND = 10?
Yes
No
NOFFND = NOFFND + 1
IMAX = NDSGV(1)
JAIL(I, NOFFND) = NDSGV(I + 1) for I = 1, IMAX
SENTN(NOFFND) = NJAIL
RETURN


ENTRY JUDGE
NJAIL = 0?
Yes
RETURN
No
NOFFND = 0?
Yes
No
A


A
IMAX =
NDSGV(1)
J = 1
Test design
variable against
JAIL(I, J)
I = 1, IMAX
No
J > NOFFND?
J = J + 1
No
Variable
= JAIL?
Yes
Yes
RETURN
RETURN 1
No
SENTN(J) = 0?
SENTN(J) =
SENTN(J) - 1
Yes
Delete Jth.
SENTN
Delete Jth.
JAIL
RETURN
NOFFND =
NOFFND - 1

3.5.9 MEVCTZ (MEVCTR)

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

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

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.

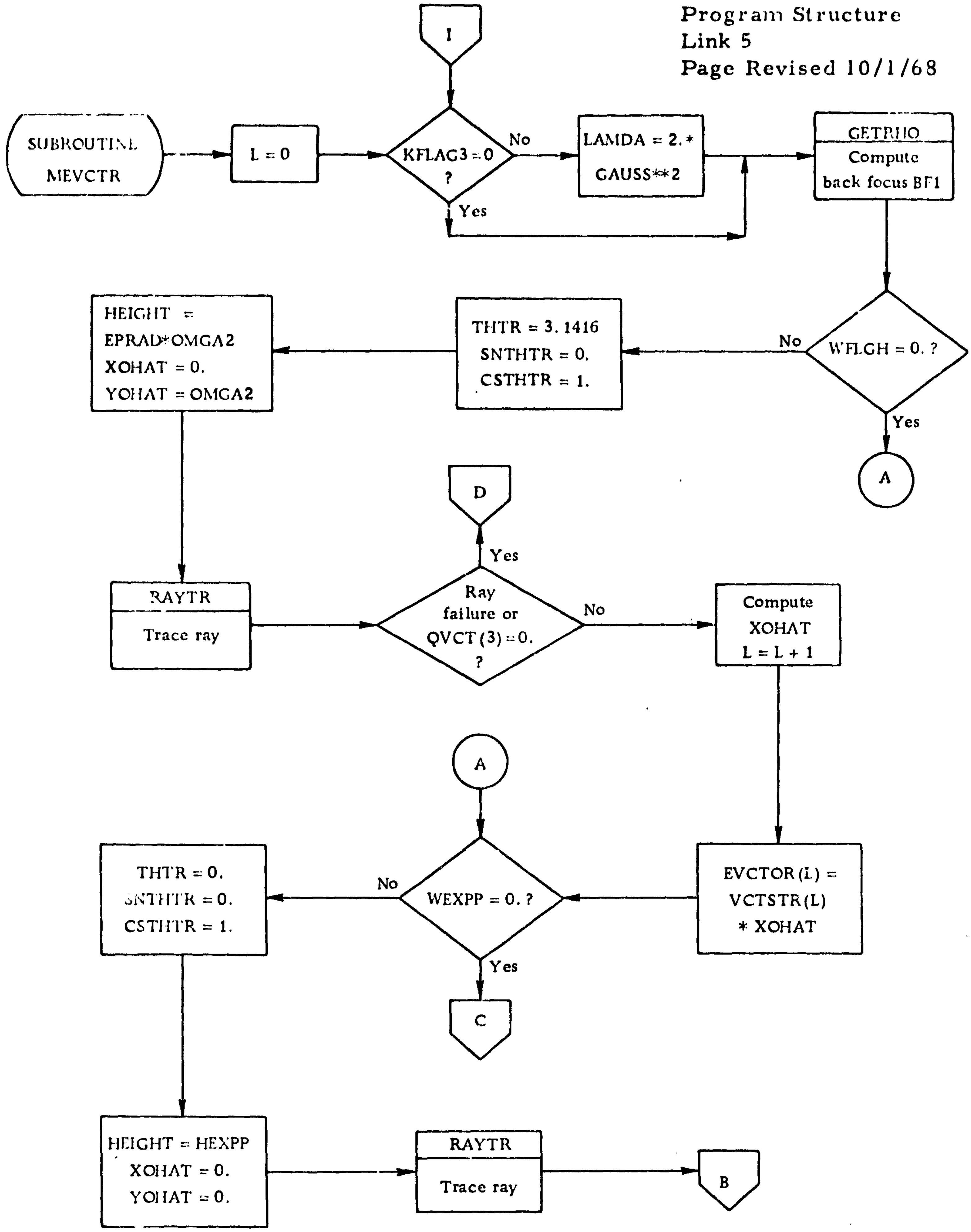
*: Alternate return utilized in case of ray failure.

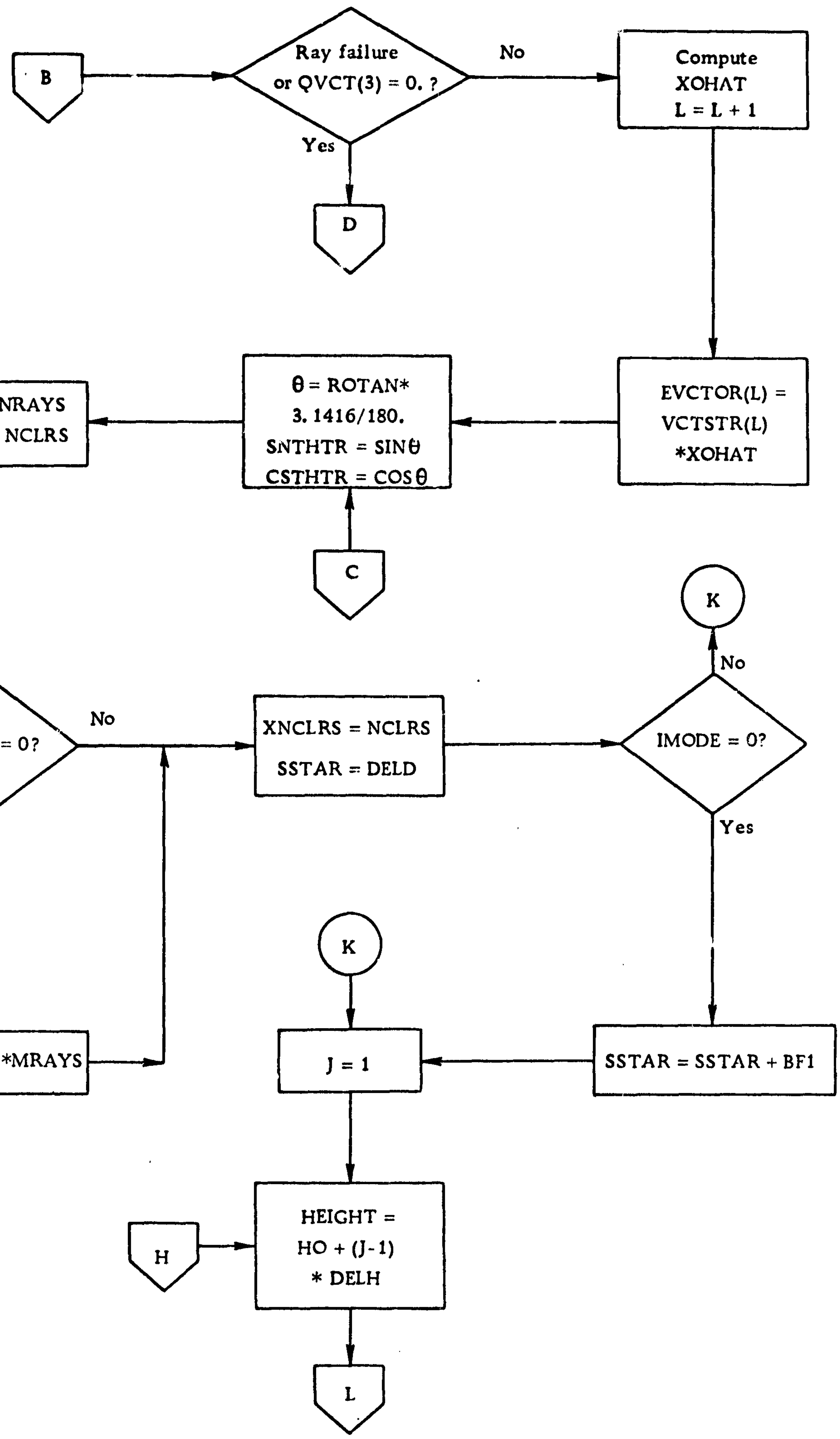
KFLAG1: Curved image surface option flag.

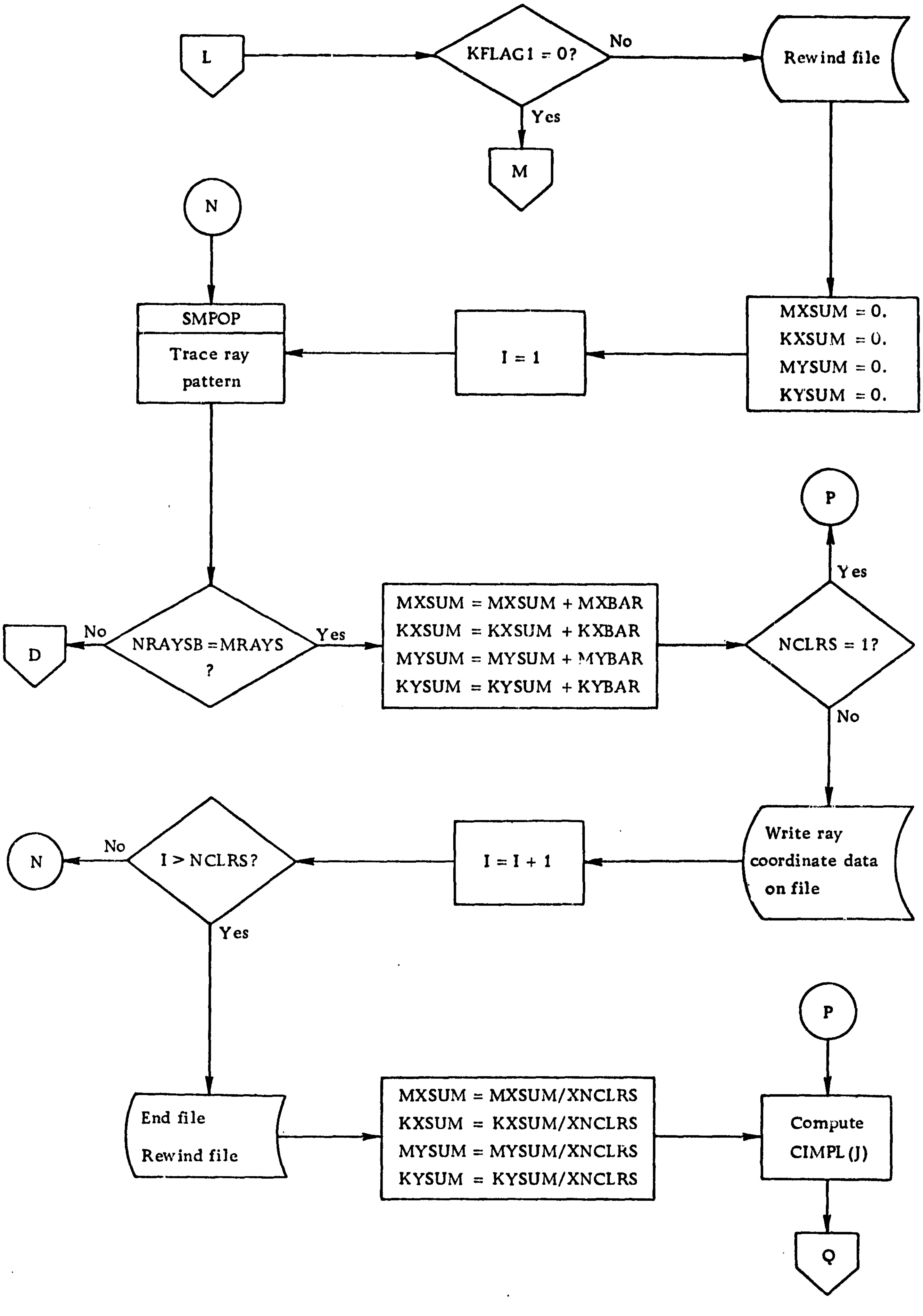
KFLAG3: Gaussian weighting option flag.

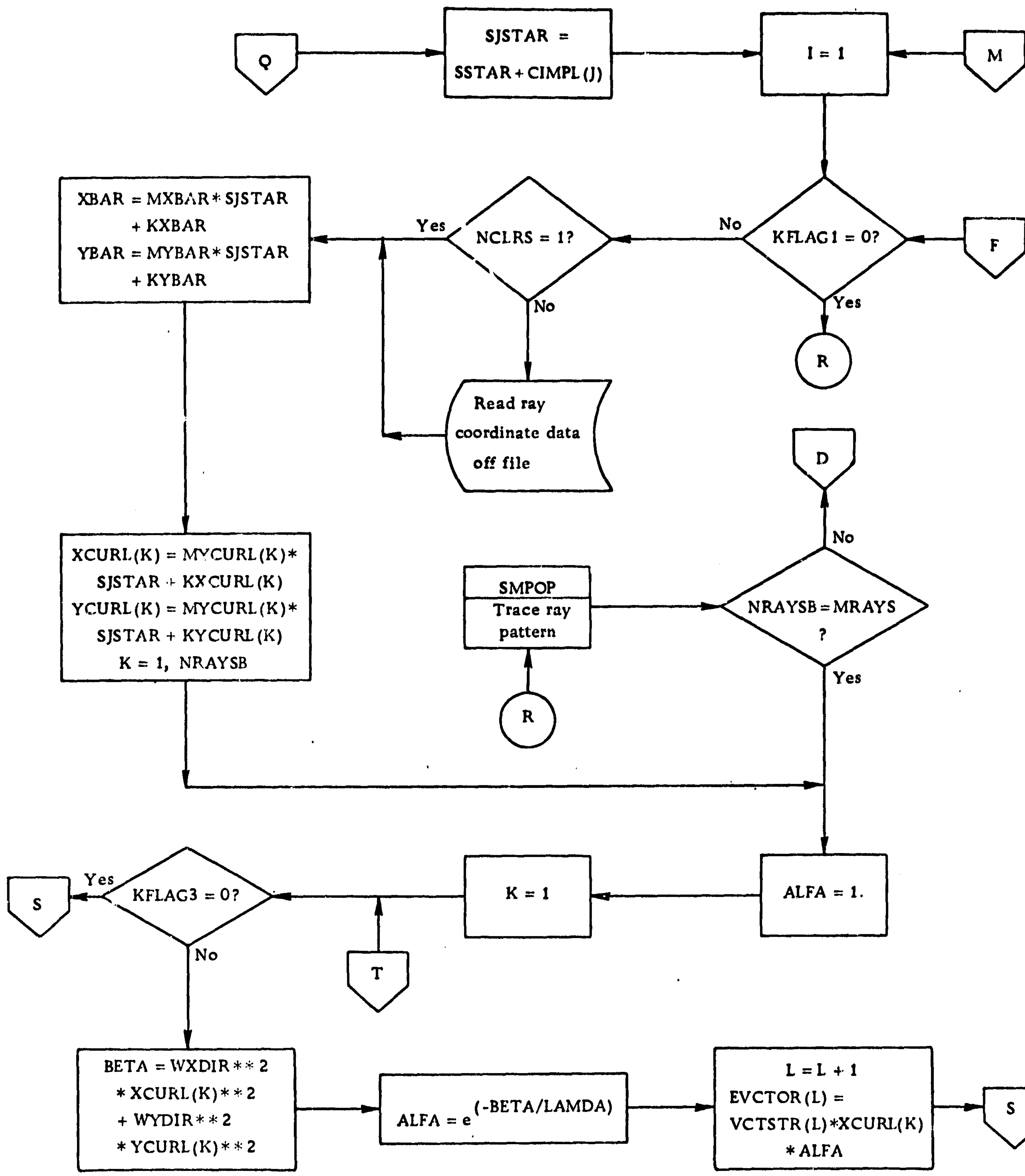
Utility Routines and Common References

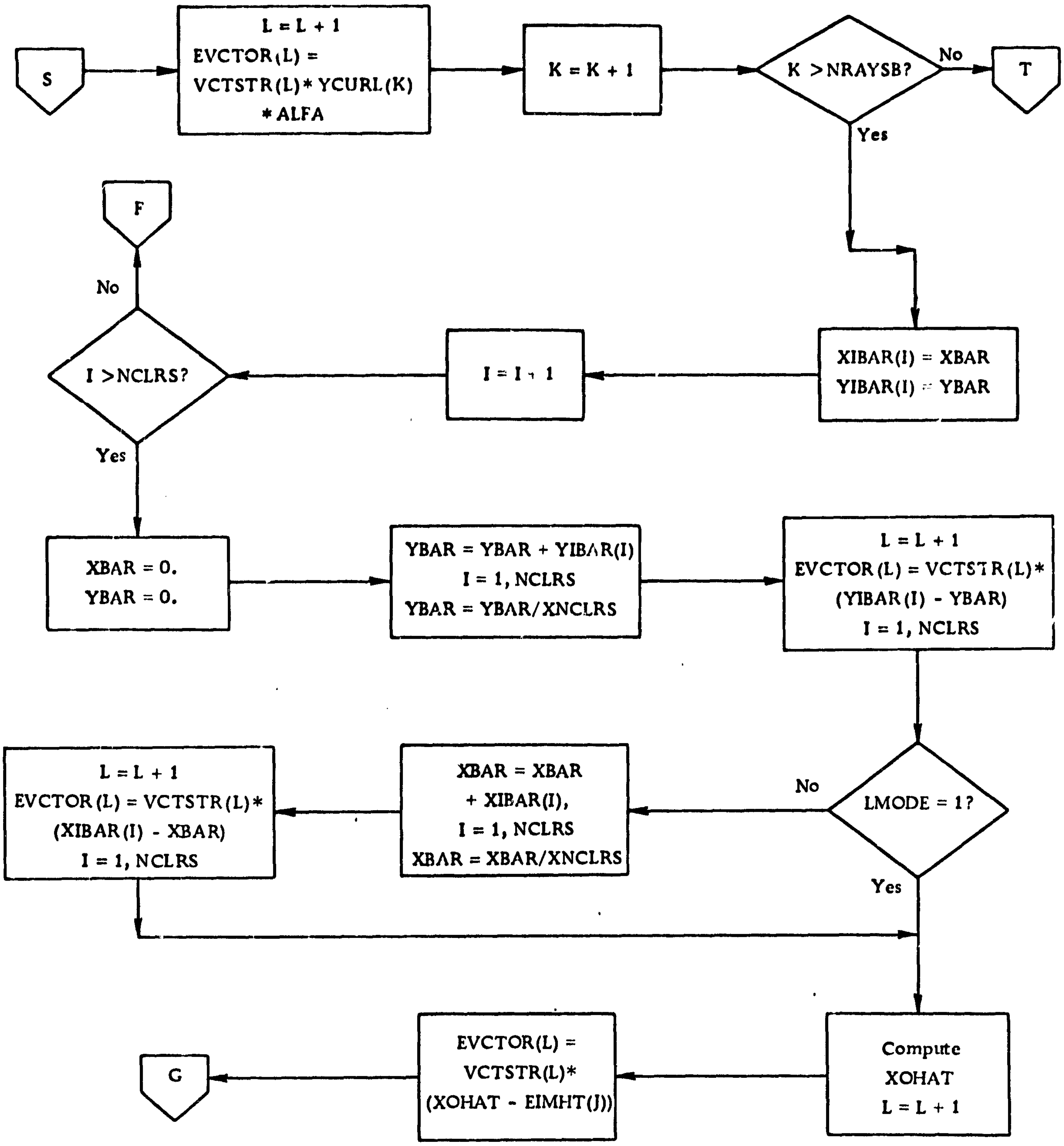
COMMON/DATA/
COMMON/AZOBJ/
COMMON/TMPATT/
SMPOP
RAYTR
PRMSUB
GETRHO

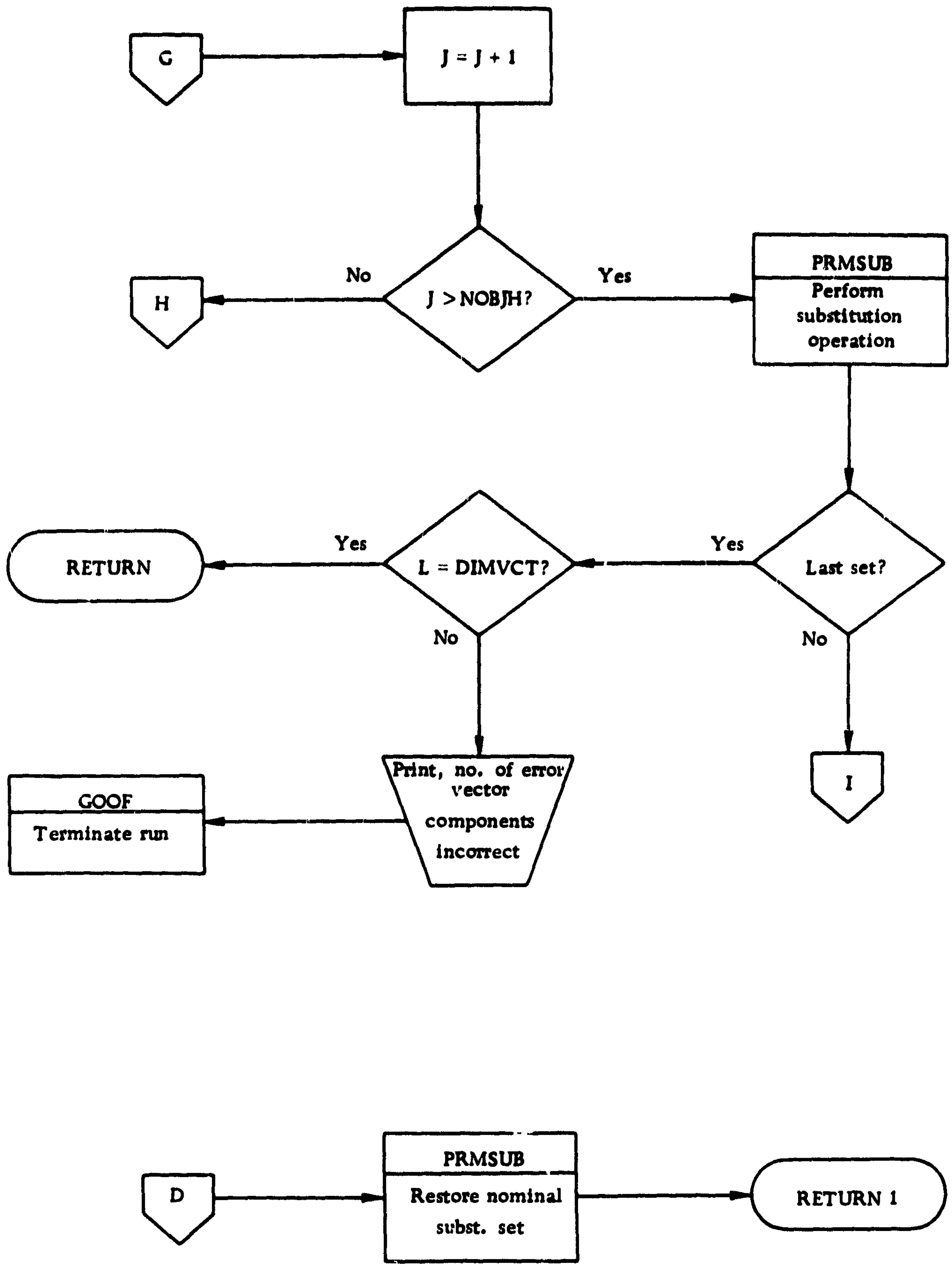

I
SUBROUTINE MEVCTR
L = 0
KFLAG3 = 0 ?
No
Yes
LAMDA = 2.* GAUSS**2
GETRHO
Compute back focus BF1
WFLGH = 0. ?
No
Yes
A
THTR = 3.1416
SNTHTR = 0.
CSTHTR = 1.
HEIGHT = EPRAD*OMGA2
XOHAT = 0.
YOHAT = OMGA2
D
Yes
RAYTR
Trace ray
Ray failure or QVCT(3) = 0. ?
No
Compute XOHAT
L = L + 1
A
EVCTOR(L) = VCTSTR(L) * XOHAT
WEXPP = 0. ?
No
Yes
THTR = 0.
SNTHTR = 0.
CSTHTR = 1.
C
HEIGHT = HEXPP
XOHAT = 0.
YOHAT = 0.
RAYTR
Trace ray
B


B
Ray failure or QVCT(3) = 0. ?
No
Yes
D
Compute XOHAT L = L + 1
EVCTOR(L) = VCTSTR(L) *XOHAT
θ = ROTAN* 3. 1416/180. SNTHTR = SINθ CSTHTR = COSθ
C
MRAYS = NRAYS XNCLRS = NCLRS
LMODE = 0?
No
Yes
XNCLRS = NCLRS SSTAR = DELD
IMODE = 0?
No
Yes
K
K
MRAYS = 2*MRAYS
J = 1
SSTAR = SSTAR + BF1
H
HEIGHT = HO + (J-1) * DELH
L


L
KFLAG1 = 0?
No
Yes
M
Rewind file
N
SMPOP
Trace ray pattern
I = 1
MXSUM = 0.
KXSUM = 0.
MYSUM = 0.
KYSUM = 0.
P
Yes
NRAYSB = MRAYS ?
No
D
Yes
MXSUM = MXSUM + MXBAR
KXSUM = KXSUM + KXBAR
MYSUM = MYSUM + MYBAR
KYSUM = KYSUM + KYBAR
NCLRS = 1?
No
Write ray coordinate data on file
I = I + 1
I > NCLRS?
No
N
Yes
P
End file
Rewind file
MXSUM = MXSUM/XNCLRS
KXSUM = KXSUM/XNCLRS
MYSUM = MYSUM/XNCLRS
KYSUM = KYSUM/XNCLRS
Compute CIMPL(J)
Q


Q
SJSTAR = SSTAR + CIMPL (J)
I = 1
M
KFLAG1 = 0?
No
Yes
F
R
NCLRS = 1?
Yes
No
XBAR = MXBAR * SJSTAR + KXBAR
YBAR = MYBAR * SJSTAR + KYBAR
Read ray coordinate data off file
D
No
XCURL (K) = MYCURL (K)* SJSTAR + KXCURL (K)
YCURL (K) = MYCURL (K)* SJSTAR + KYCURL (K)
K = 1, NRAYSB
SMPOP
Trace ray pattern
R
NRAYSB = MRAYS ?
Yes
ALFA = 1.
K = 1
KFLAG3 = 0?
Yes
S
No
T
BETA = WXDIR ** 2 * XCURL (K) ** 2 + WYDIR ** 2 * YCURL (K) ** 2
ALFA = e (-BETA/LAMDA)
L = L + 1
EVCTOR (L) = VCTSTR (L) * XCURL (K) * ALFA
S


S
L = L + 1
EVCTOR(L) =
VCTSTR(L)* YCURL(K)
* ALFA
K = K + 1
K > NRAYSB?
No
T
Yes
F
No
I > NCLRS?
Yes
I = I + 1
XIBAR(I) = XBAR
YIBAR(I) = YBAR
XBAR = 0.
YBAR = 0.
YBAR = YBAR + YIBAR(I)
I = 1, NCLRS
YBAR = YBAR/XNCLRS
L = L + 1
EVCTOR(L) = VCTSTR(L)*
(YIBAR(I) - YBAR)
I = 1, NCLRS
L = L + 1
EVCTOR(L) = VCTSTR(L)*
(XIBAR(I) - XBAR)
I = 1, NCLRS
XBAR = XBAR
+ XIBAR(I),
I = 1, NCLRS
XBAR = XBAR/XNCLRS
No
LMODE = 1?
Yes
G
EVCTOR(L) =
VCTSTR(L)*
(XOHAT - EIMHT(J))
Compute
XOHAT
L = L + 1


G
J = J + 1
J > NOBJH?
No
H
Yes
PRMSUB
Perform substitution operation
Last set?
Yes
No
I
L = DIMVCT?
Yes
RETURN
No
Print, no. of error vector components incorrect
GOOF
Terminate run
D
PRMSUB
Restore nominal subst. set
RETURN 1

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 $J = \{j_1, j_2, \ldots j_n\}$, representing the design variables presnet, if $\Delta_{j_i}^{(C)}$, i.e. (DESIGN(10, I)). is 0, then ΔV_{j_i} (DESGN(3, I)) is unchanged. Otherwise:

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

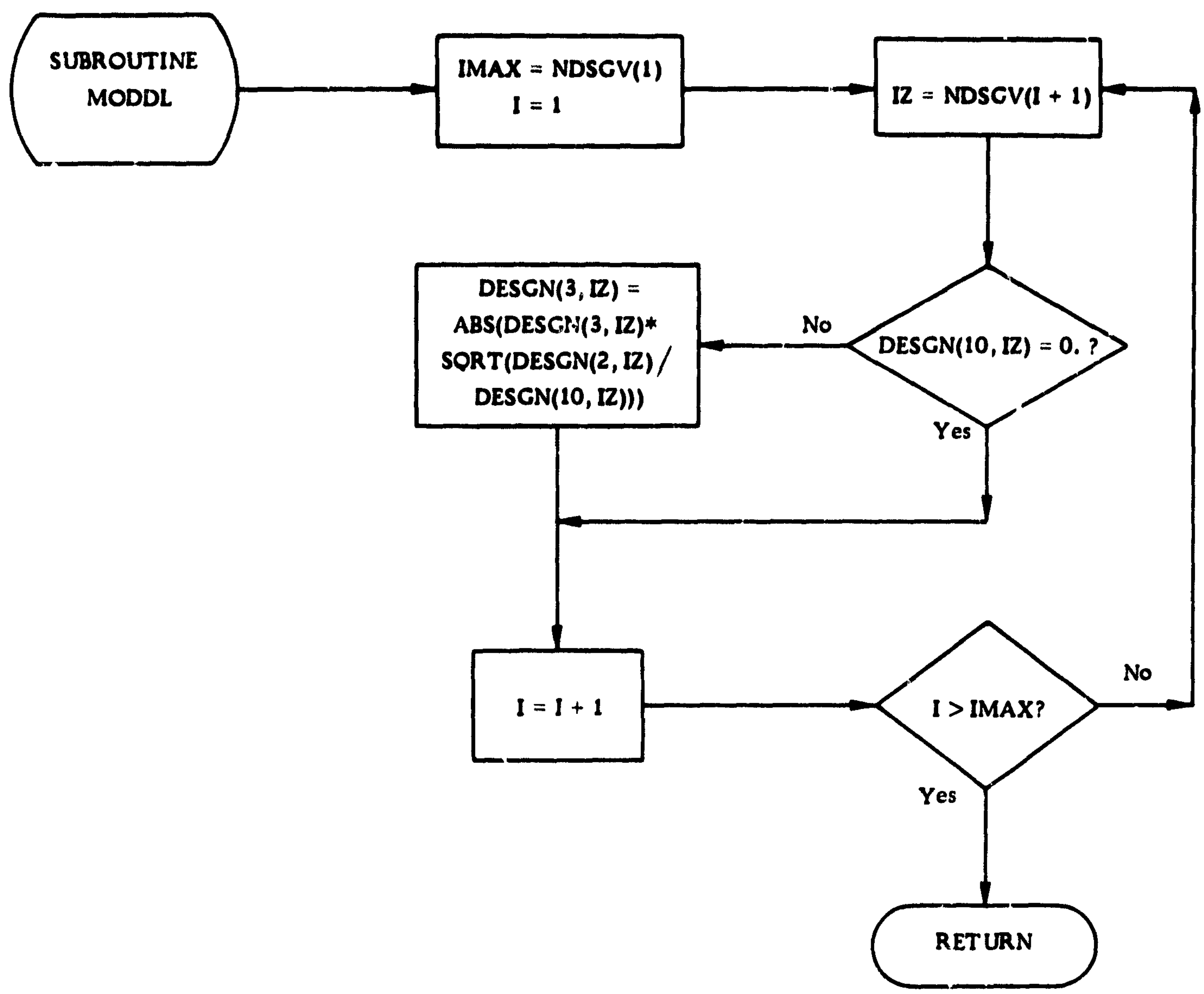
Here $1 \leq I \leq$ NDSGV(1), the number of design variables per design step.

Calling Sequence

CALL MODDL

Utility Routines and Common References

COMMON/DATA/


SUBROUTINE MODDL
IMAX = NDSGV(1)
I = 1
IZ = NDSGV(I + 1)
DESGN(10, IZ) = 0. ?
No
Yes
DESGN(3, IZ) = ABS(DESGN(3, IZ)* SQRT(DESGN(2, IZ)/ DESGN(10, IZ)))
I = I + 1
I > IMAX?
No
Yes
RETURN

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 = 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 wieght.

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.

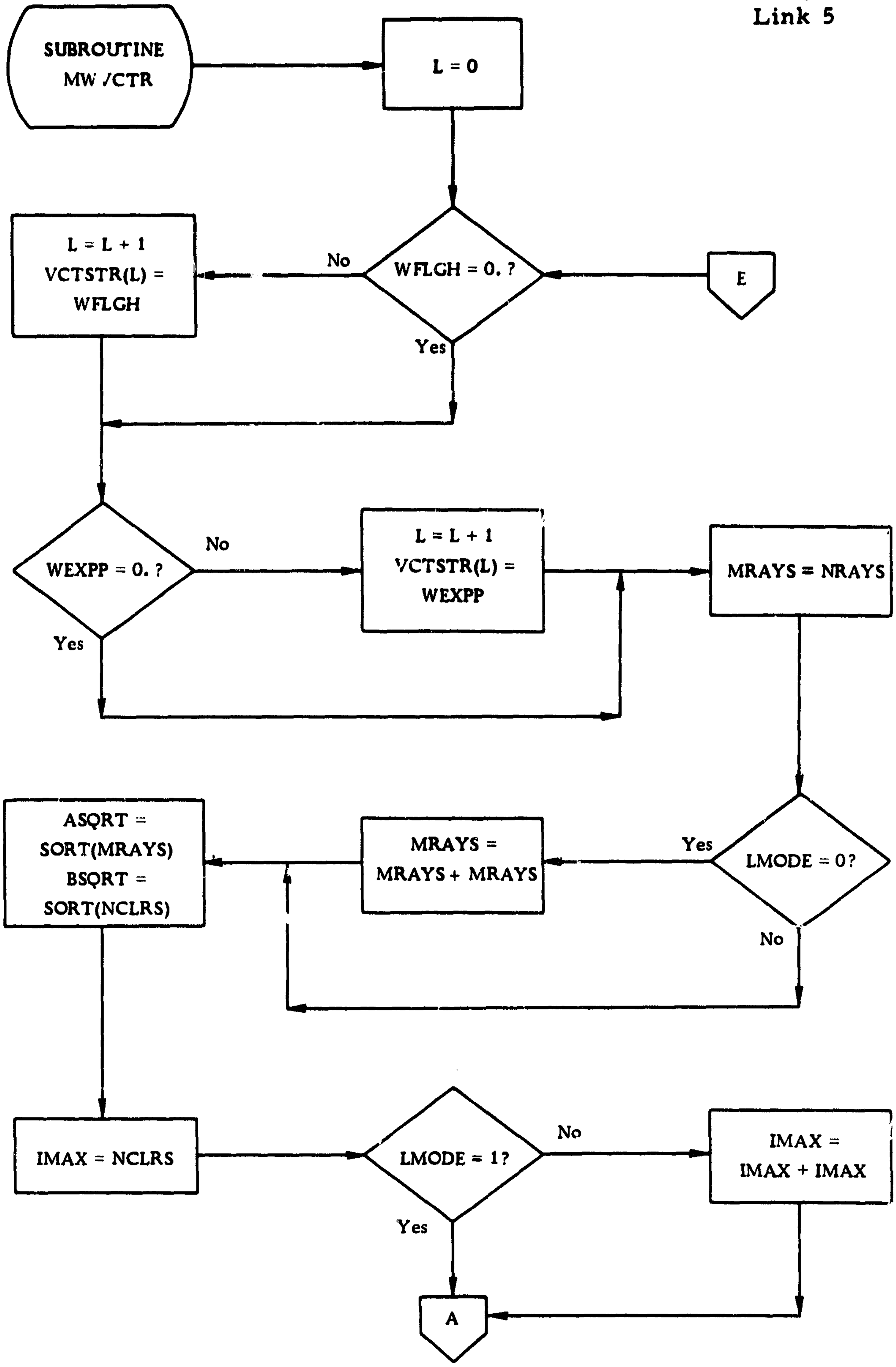
Utility Routines and Common References

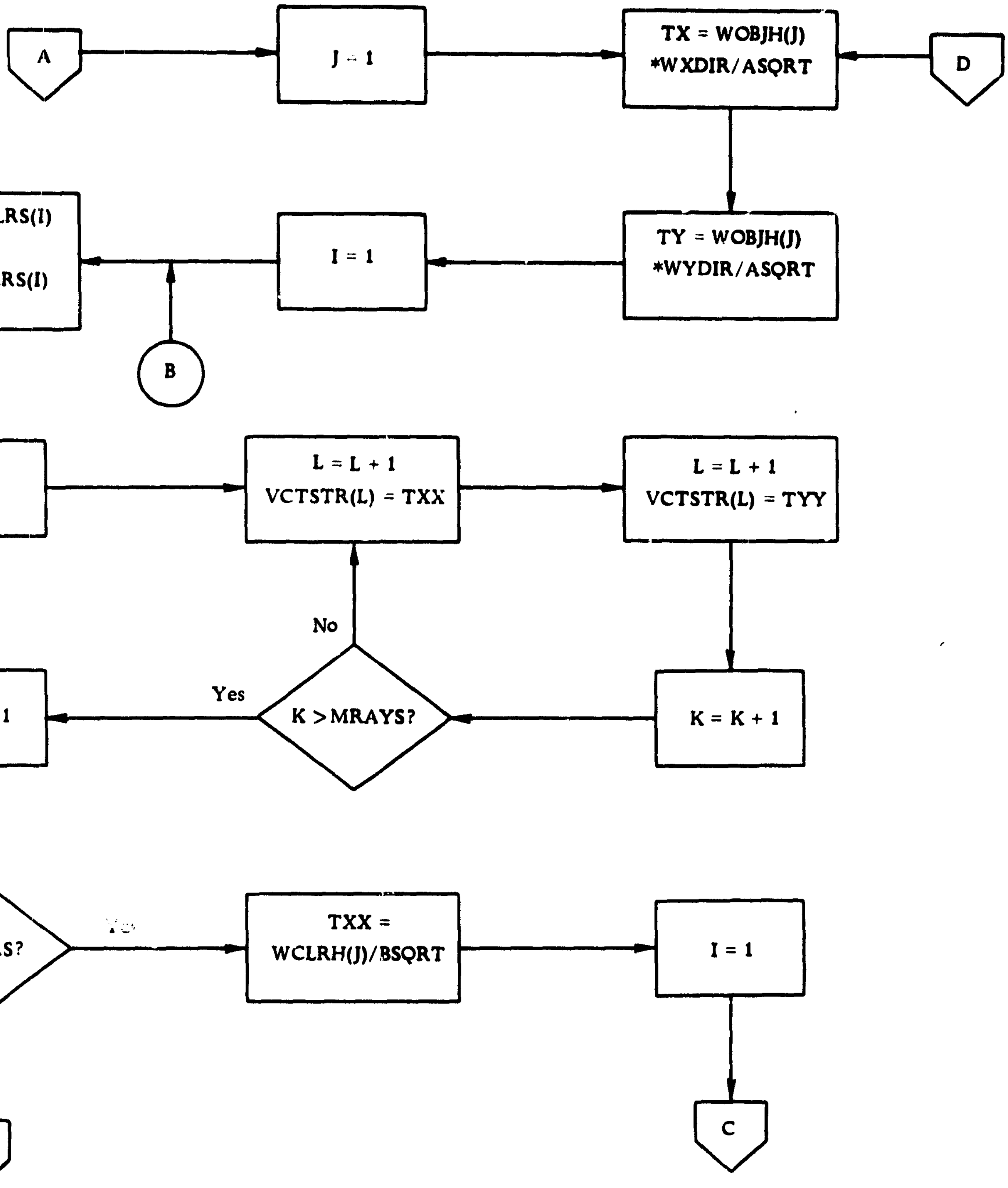
COMMON/DATA/

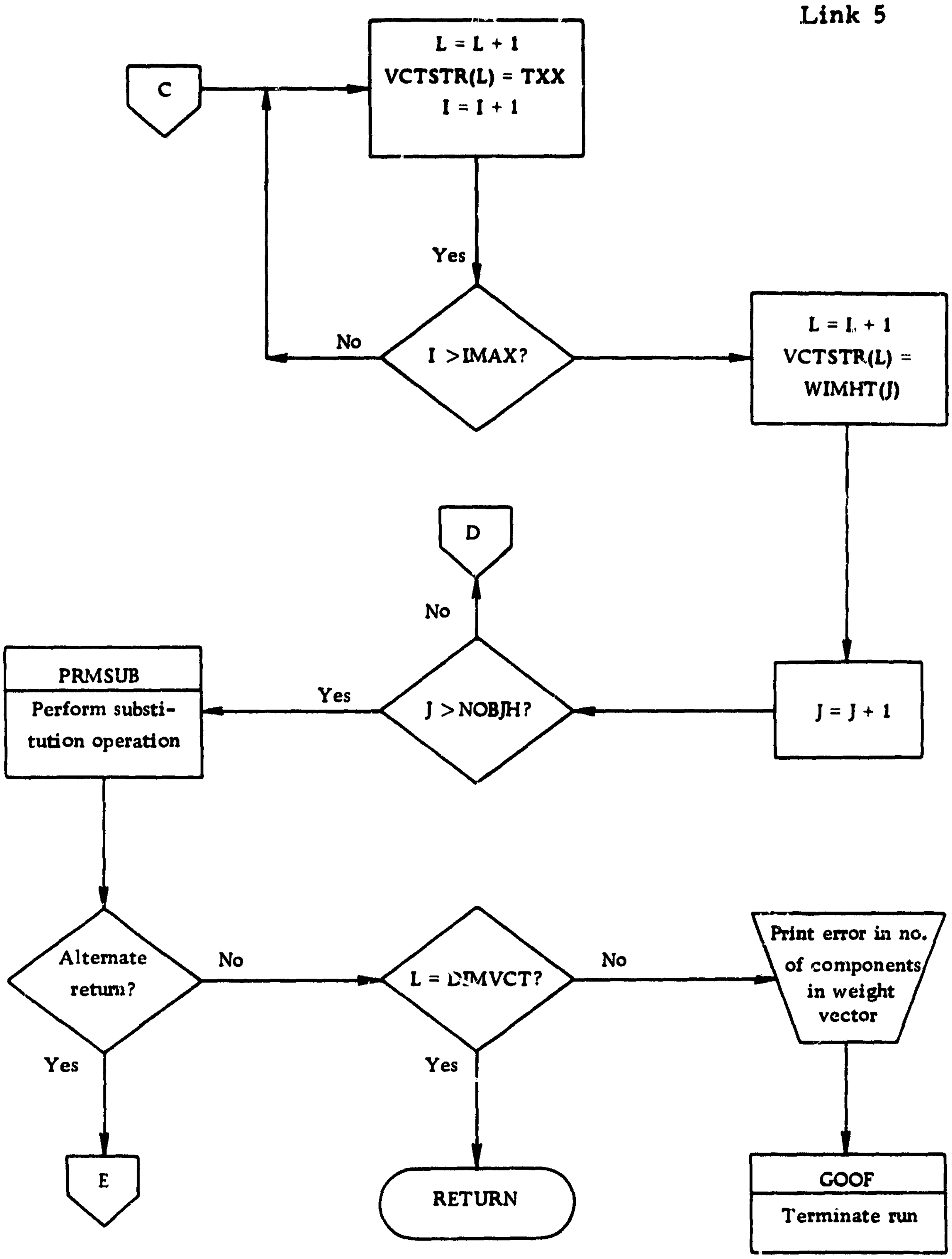
COMMON/TMDESN/

PRMSUB

GOOF


SUBROUTINE
MW /CTR
L = 0
WFLGH = 0. ?
No
Yes
E
L = L + 1
VCTSTR(L) =
WFLGH
WEXPP = 0. ?
No
Yes
L = L + 1
VCTSTR(L) =
WEXPP
MRAYS = NRAYS
LMODE = 0?
Yes
No
MRAYS =
MRAYS + MRAYS
ASQRT =
SORT(MRAYS)
BSQRT =
SORT(NCLRS)
IMAX = NCLRS
LMODE = 1?
No
Yes
IMAX =
IMAX + IMAX
A


A
J = 1
TX = WOBJH(J)
*WXDIR/ASQRT
D
TY = WOBJH(J)
*WYDIR/ASQRT
I = 1
TXX = WCLRS(I)
*TX
TYY = WCLRS(I)
*TY
B
K = 1
L = L + 1
VCTSTR(L) = TXX
L = L + 1
VCTSTR(L) = TYY
No
Yes
K >MRAYS?
I = I + 1
K = K + 1
I >NCLRS?
Yes
No
TXX =
WCLRH(J)/BSQRT
I = 1
B
C


C
L = L + 1
VCTSTR(L) = TXX
I = I + 1
Yes
No
I >IMAX?
L = L + 1
VCTSTR(L) =
WIMHT(J)
D
No
PRMSUB
Perform substitution operation
Yes
J >NOBJH?
J = J + 1
Alternate return?
No
L = DIMVCT?
No
Print error in no. of components in weight vector
Yes
Yes
E
RETURN
GOOF
Terminate run

3.5.12 STAR2Z (STAR2)

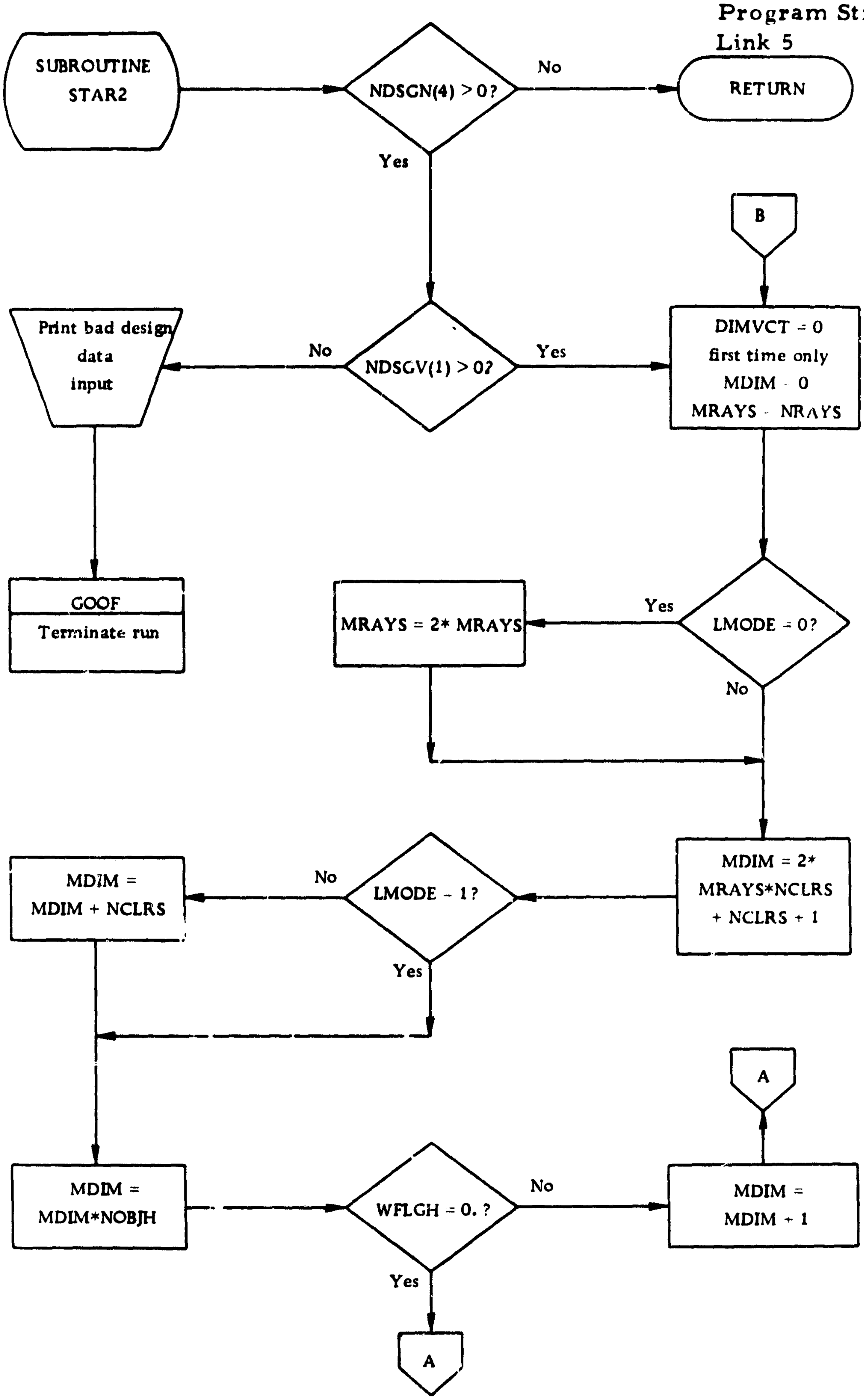
STAR2Z is the control routine for the design logic of the program. Several tests are made to determine if the design specifications are valid. If the design problem is too large for the available computer storage, the run must be terminated. After NDSGV(4) successful iterations have been made the program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

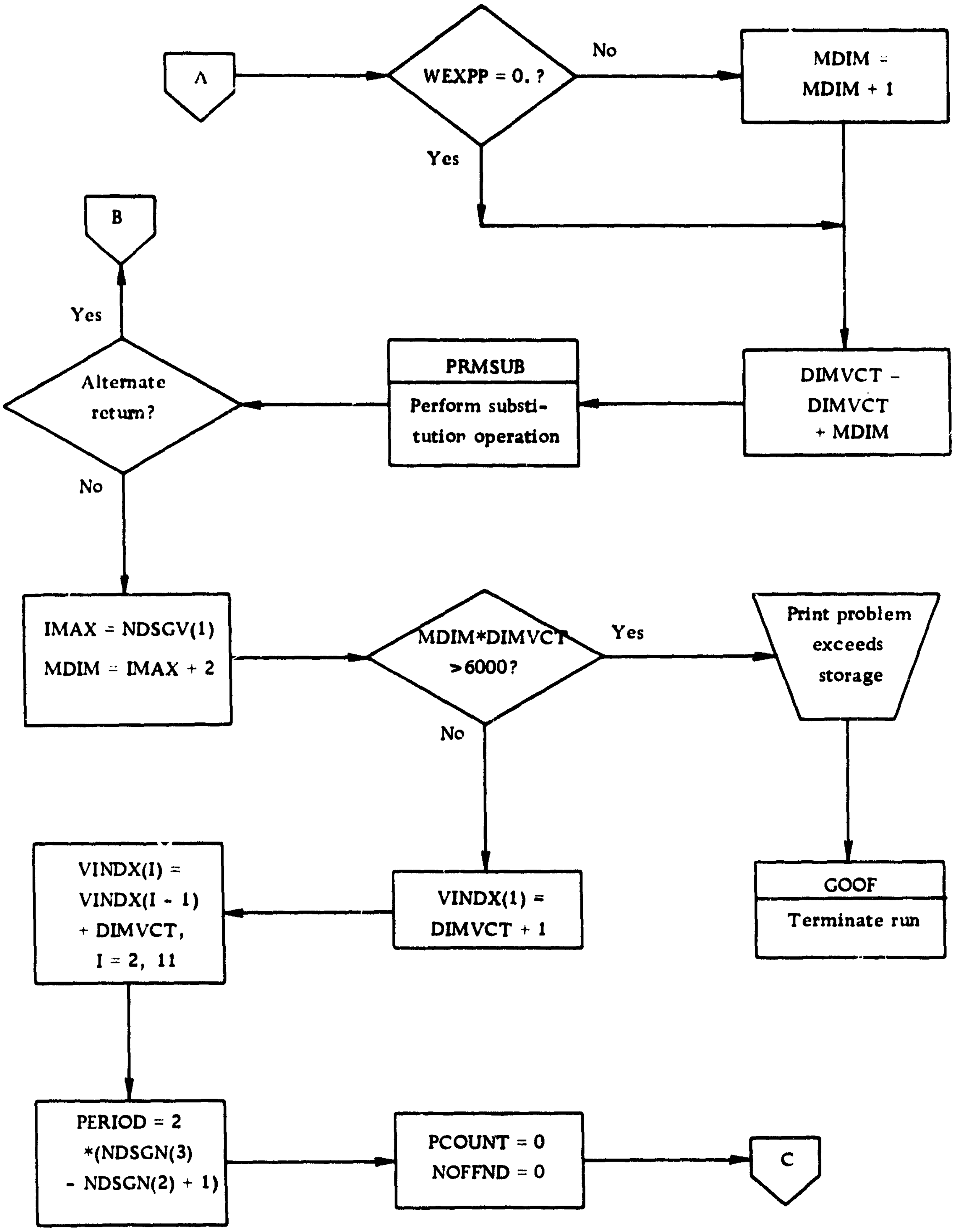
Calling Sequence

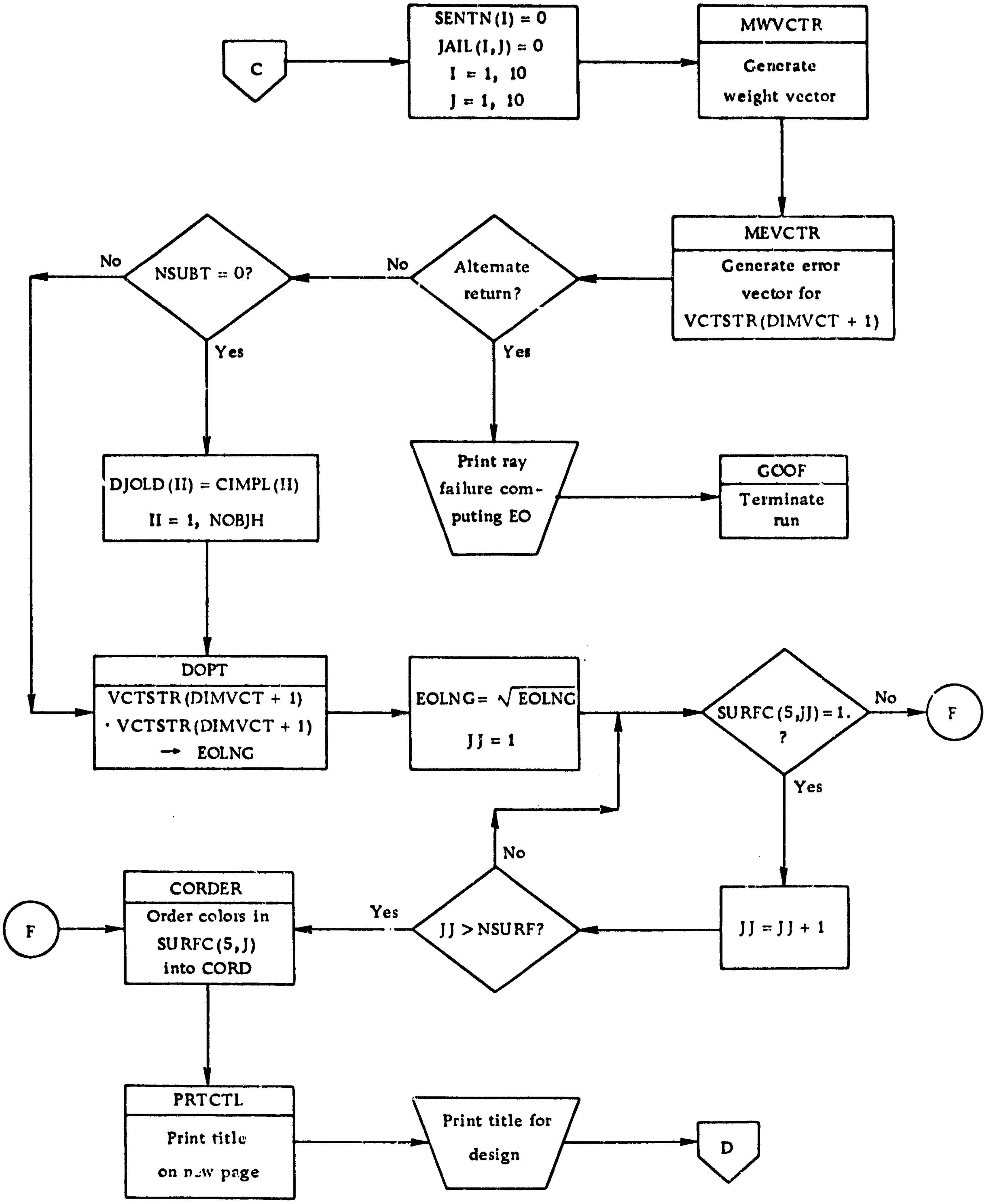
CALL STAR2

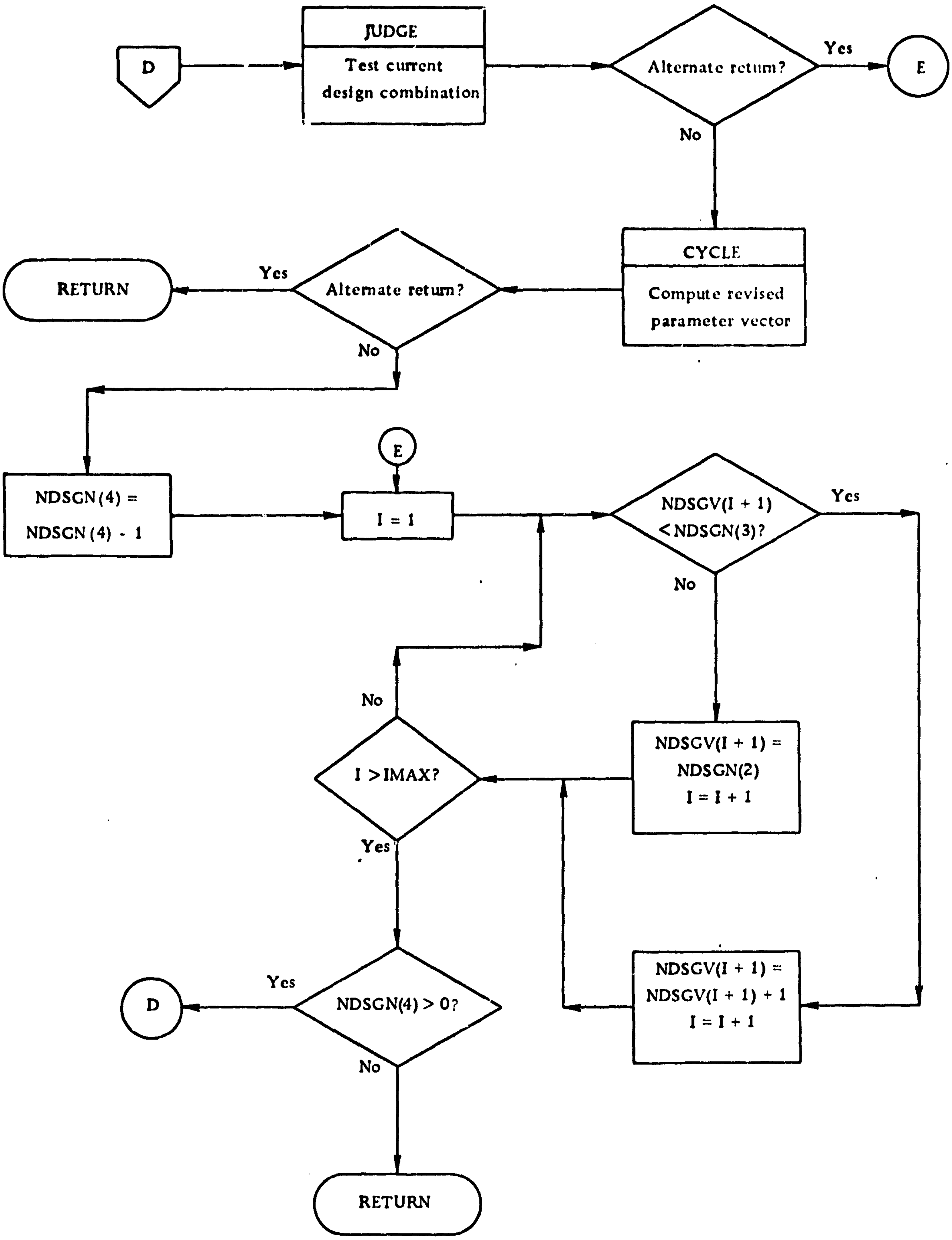
Utility Routines and Common References

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


SUBROUTINE STAR2
NDSGN(4) > 0?
No
RETURN
Yes
B
Print bad design data input
No
NDSGV(1) > 0?
Yes
DIMVCT = 0 first time only
MDIM = 0
MRAYS = NRAYS
GOOF
Terminate run
MRAYS = 2* MRAYS
Yes
LMODE = 0?
No
MDIM = MDIM + NCLRS
No
LMODE = 1?
MDIM = 2* MRAYS*NCLRS + NCLRS + 1
Yes
A
MDIM = MDIM*NOBJH
WFLGH = 0. ?
No
MDIM = MDIM + 1
Yes
A


A
WEXPP = 0. ?
No
Yes
MDIM = MDIM + 1
B
Yes
Alternate return?
No
PRMSUB
Perform substitution operation
DIMVCT = DIMVCT + MDIM
IMAX = NDSGV(1)
MDIM = IMAX + 2
MDIM*DIMVCT >6000?
Yes
No
Print problem exceeds storage
VINDX(I) = VINDX(I - 1) + DIMVCT, I = 2, 11
VINDX(1) = DIMVCT + 1
GOOF
Terminate run
PERIOD = 2 *(NDSGN(3) - NDSGN(2) + 1)
PCOUNT = 0
NOFFND = 0
C


C
SENTN(I) = 0
JAIL(I,J) = 0
I = 1, 10
J = 1, 10
MWVCTR
Generate weight vector
MEVCTR
Generate error vector for VCTSTR(DIMVCT + 1)
Alternate return?
No
Yes
NSUBT = 0?
No
Yes
Print ray failure computing EO
GOOF
Terminate run
DJOLD(II) = CIMPL(II)
II = 1, NOBJH
DOPT
VCTSTR(DIMVCT + 1)
· VCTSTR(DIMVCT + 1)
→ EOLNG
EOLNG= √EOLNG
JJ = 1
SURFC(5,JJ) = 1.
?
No
Yes
F
No
JJ > NSURF?
Yes
JJ = JJ + 1
F
CORDER
Order colors in SURFC(5,J) into CORD
PRTCTL
Print title on new page
Print title for design
D


D
JUDGE
Test current design combination
Alternate return?
Yes
E
No
CYCLE
Compute revised parameter vector
Alternate return?
Yes
RETURN
No
NDSGN(4) = NDSGN(4) - 1
E
I = 1
NDSGV(I + 1) < NDSGN(3)?
Yes
No
No
NDSGV(I + 1) = NDSGN(2)
I = I + 1
I > IMAX?
Yes
NDSGV(I + 1) = NDSGV(I + 1) + 1
I = I + 1
NDSGN(4) > 0?
Yes
D
No
RETURN

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; $t_n\, t_{n-1} \cdots t_1\, A = I$. 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'≥N, M'≥M.
2. N is the order of A;N≥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 atleast 3N, and S2 must be dimensioned atleast 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} if 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:

ORDER(N)	ACTUAL TIME	N^2-4N+6
2	2.33ms	2
3	2.83	3
5	11.33	11
6	17.33	18
7	25.50	27
8	37.33	38
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.

3.6 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 prinipal image plane and a specified set of alternate planes.

3.6.1 PARAXZ (PARAX)

PARAXZ traces a ray from the object point $(\rho_o \Omega_2, \pi)$ 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)	Must be set by calling program
SNTHTR:	0.	$(\sin \theta)$	Must be set by calling program
CSTHTR:	-1.	$(\cos \theta)$	Must be set by calling program

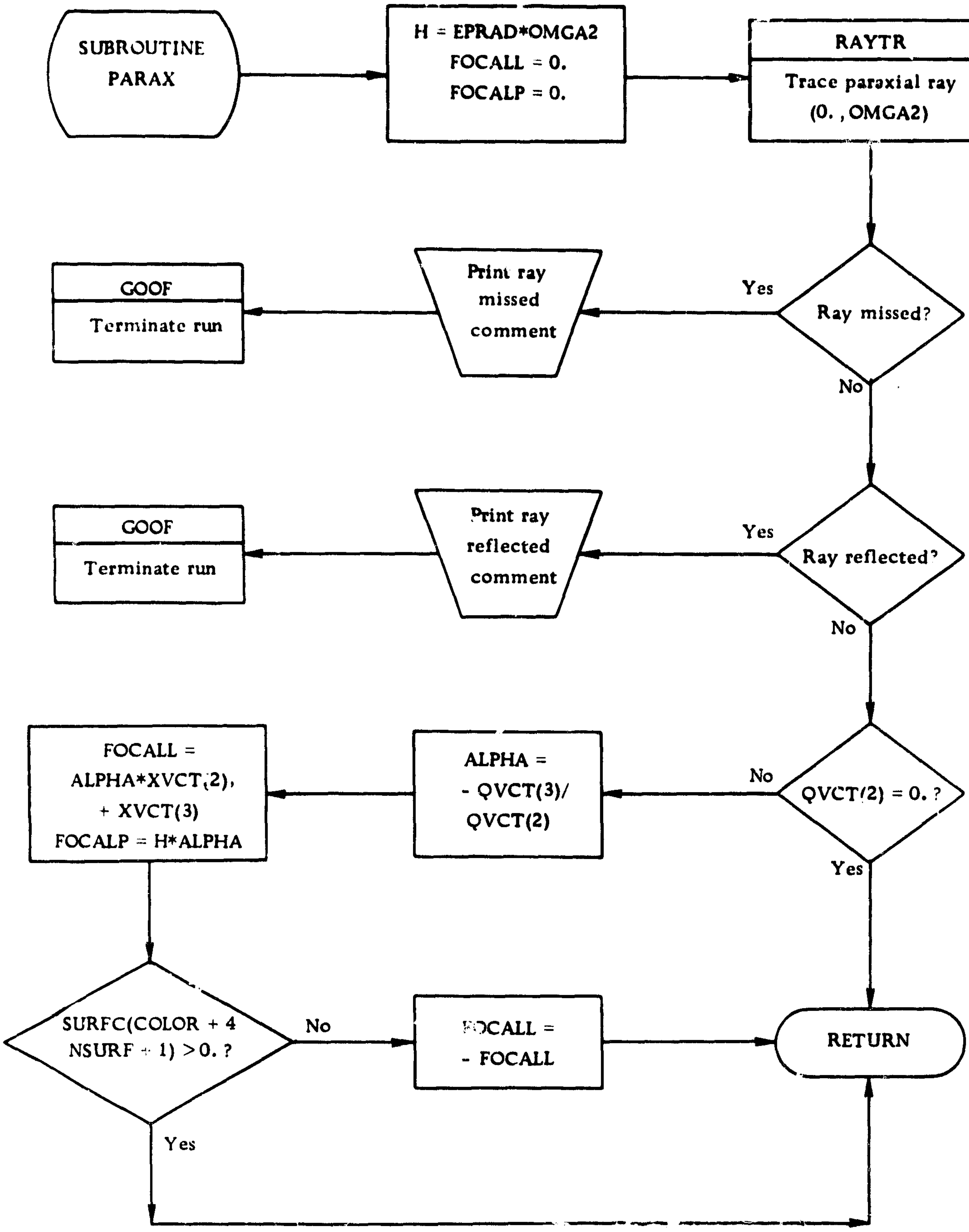
Utility Routines and Common References

COMMON/DATA/

COMMON/AZOBJ/

RAYTR

GOOF


SUBROUTINE PARAX
H = EPRAD*OMGA2
FOCALL = 0.
FOCALP = 0.
RAYTR
Trace paraxial ray (0. , OMGA2)
Ray missed?
Yes
No
Print ray missed comment
GOOF
Terminate run
Ray reflected?
Yes
No
Print ray reflected comment
GOOF
Terminate run
QVCT(2) = 0. ?
No
Yes
ALPHA = - QVCT(3)/ QVCT(2)
FOCALL = ALPHA*XVCT(2), + XVCT(3)
FOCALP = H*ALPHA
SURFC(COLOR + 4 NSURF + 1) >0. ?
No
Yes
FOCALL = - FOCALL
RETURN

3.6.2 STAR3Z (STAR3)

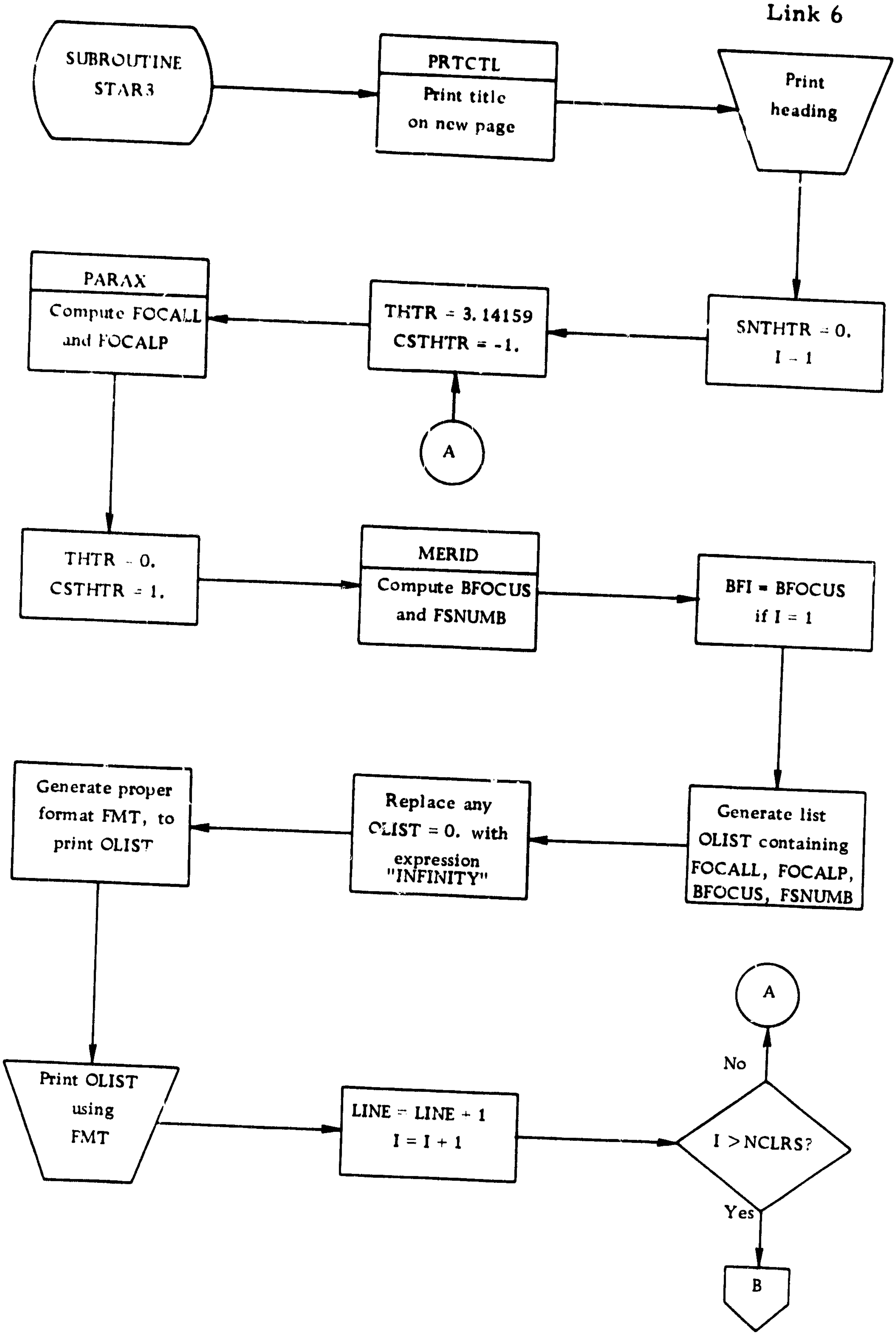
STAR3Z is the controlling subroutine for option 3 which causes optics diagnostic calculations. A table of focal length focal point, back focus, and f number for each color is produced. A table of exit pupil positions for every combination of color and non-zero object height is output. For every specified 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 $\overline{N}_R$ are included in the ray statistics for each color-object point combination.

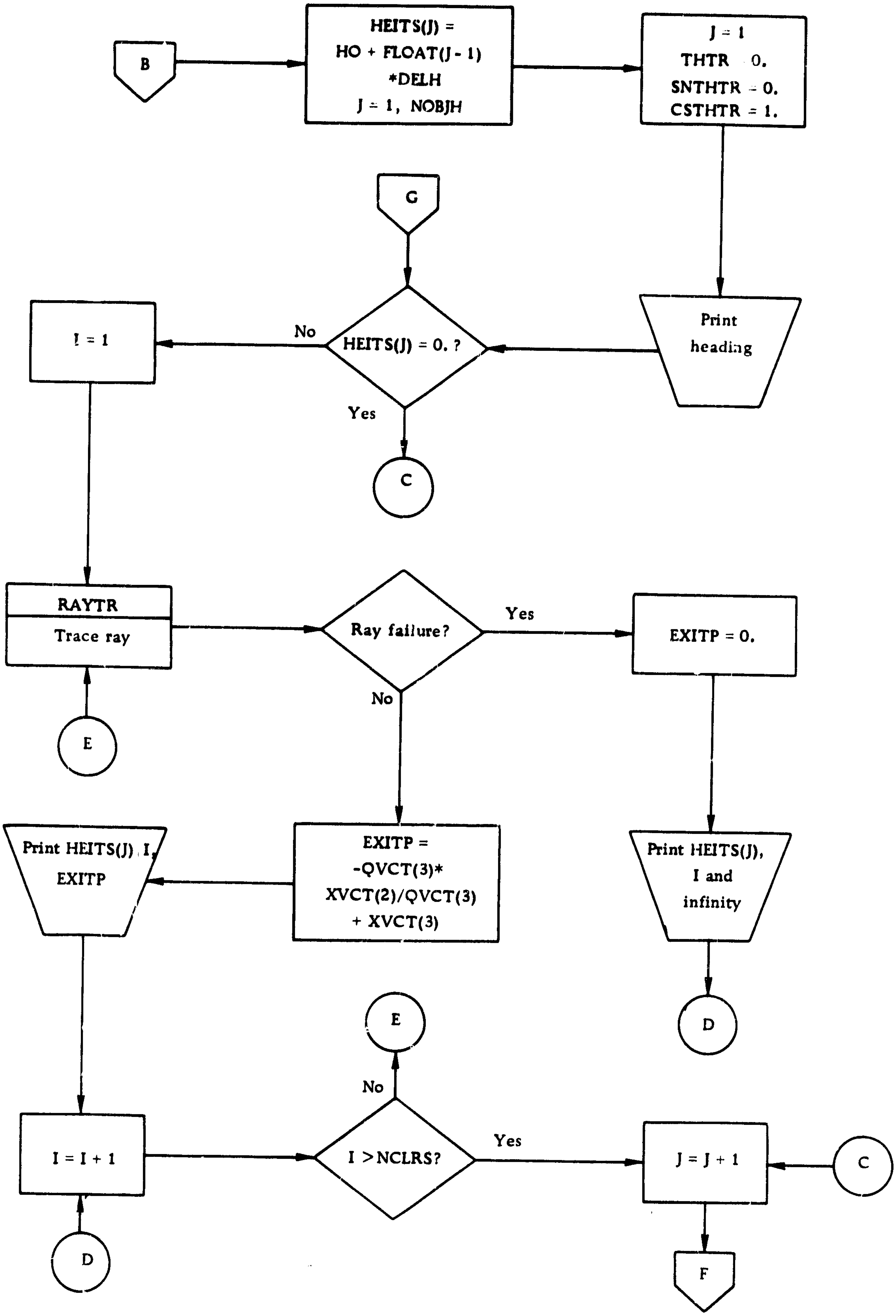
Calling Sequence

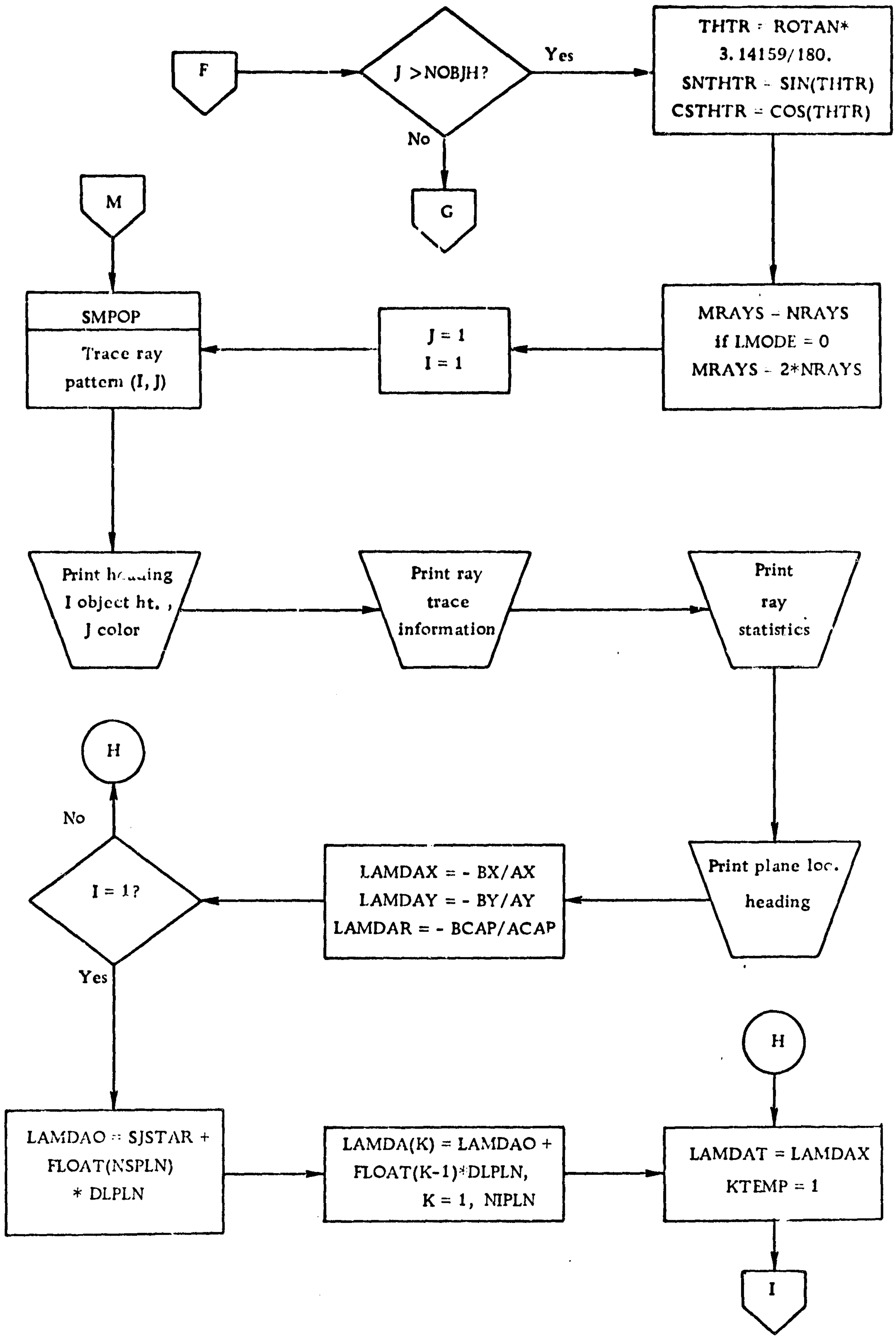
CALL STAR3

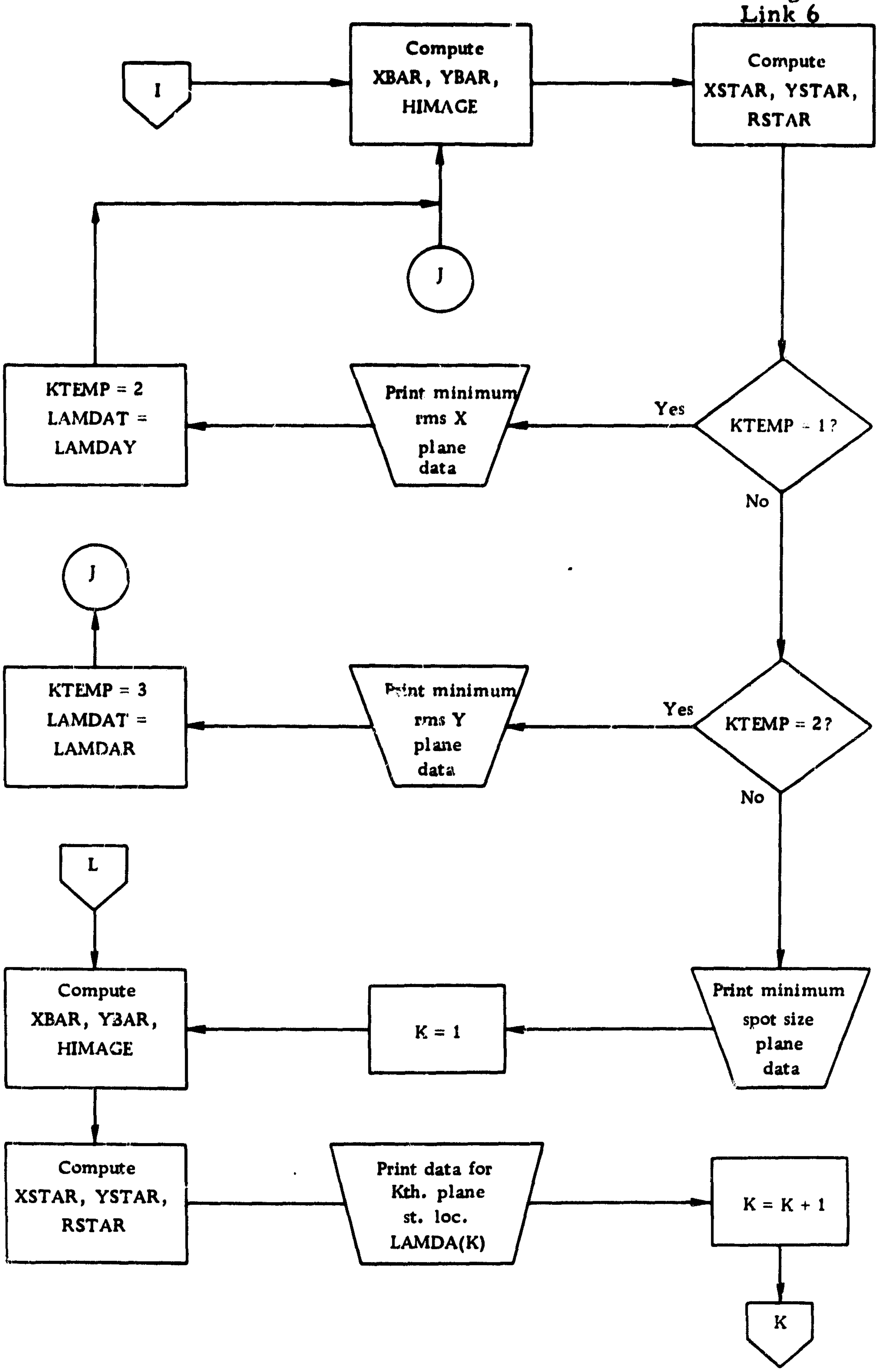
Utility Routines and Common References

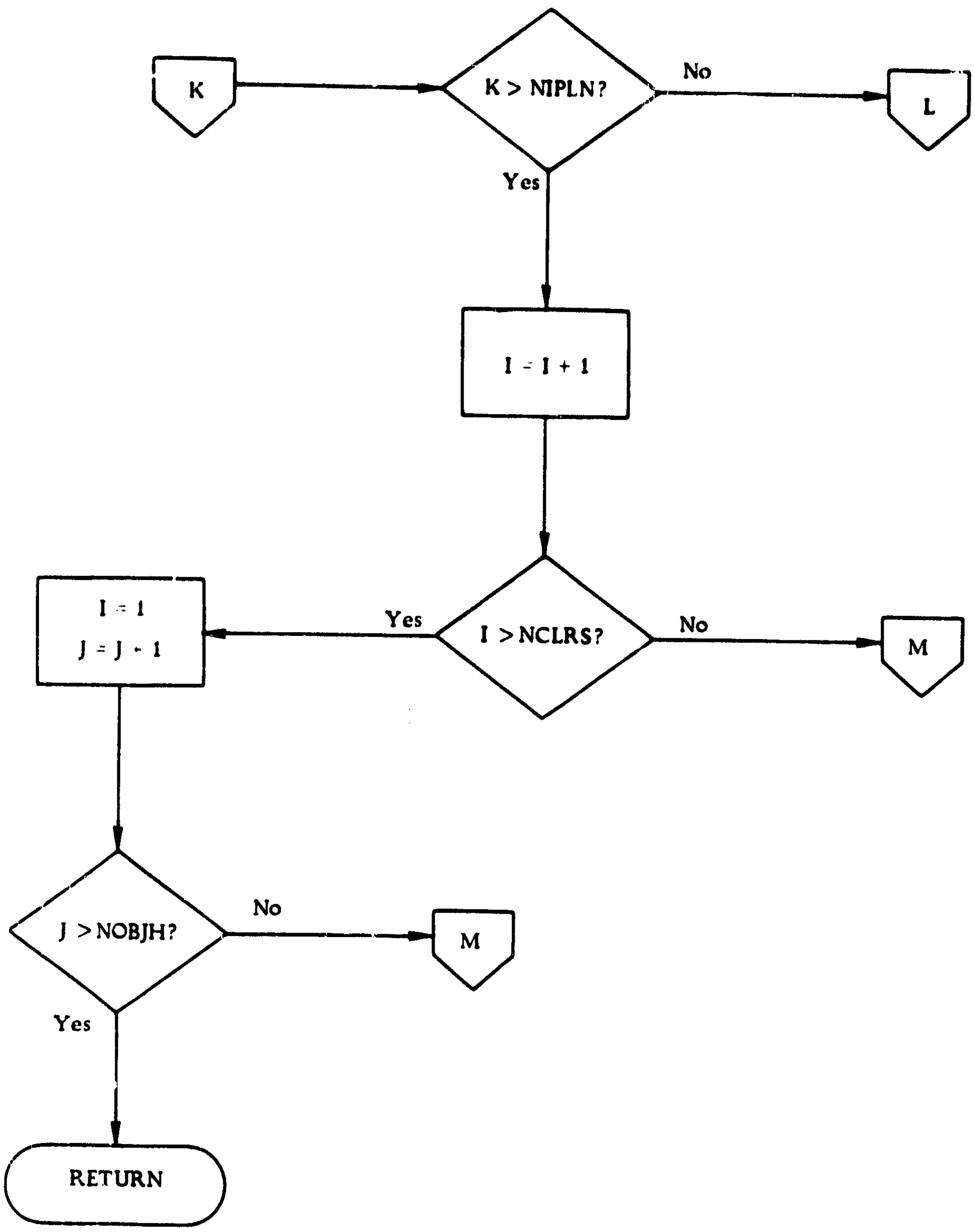
COMMON/DATA/
COMMON/TMPATT/
COMMON/AZOBJ/
COMMON/PRNT/
RAYTR
SMPOP
MERID
PARAX
PRTCTL


SUBROUTINE STAR3
PRTCTL
Print title on new page
Print heading
SNTHTR = 0.
I = 1
THTR = 3.14159
CSTHTR = -1.
A
PARAX
Compute FOCALL and FOCALP
THTR = 0.
CSTHTR = 1.
MERID
Compute BFOCUS and FSNUMB
BFI = BFOCUS if I = 1
Generate list OLIST containing FOCALL, FOCALP, BFOCUS, FSNUMB
Replace any OLIST = 0. with expression "INFINITY"
Generate proper format FMT, to print OLIST
Print OLIST using FMT
LINE = LINE + 1
I = I + 1
I >NCLRS?
No
A
Yes
B


B
HEITS(J) =
HO + FLOAT(J - 1)
*DELH
J = 1, NOBJH
J = 1
THTR = 0.
SNTHTR = 0.
CSTHTR = 1.
G
Print
heading
HEITS(J) = 0. ?
No
Yes
I = 1
C
RAYTR
Trace ray
Ray failure?
Yes
No
EXITP = 0.
E
Print HEITS(J), I,
EXITP
EXITP =
-QVCT(3)*
XVCT(2)/QVCT(3)
+ XVCT(3)
Print HEITS(J),
I and
infinity
E
D
No
I = I + 1
I >NCLRS?
Yes
J = J + 1
C
D
F


F
J >NOBJH?
Yes
No
G
THTR = ROTAN*
3.14159/180.
SNTHTR = SIN(THTR)
CSTHTR = COS(THTR)
M
SMPOP
Trace ray
pattern (I, J)
J = 1
I = 1
MRAYS = NRAYS
if LMODE = 0
MRAYS = 2*NRAYS
Print heading
I object ht.,
J color
Print ray
trace
information
Print
ray
statistics
H
No
I = 1?
Yes
LAMDAX = - BX/AX
LAMDAY = - BY/AY
LAMDAR = - BCAP/ACAP
Print plane loc.
heading
H
LAMDAO = SJSTAR +
FLOAT(NSPLN)
* DLPLN
LAMDA(K) = LAMDAO +
FLOAT(K-1)*DLPLN,
K = 1, NIPLN
LAMDAT = LAMDAX
KTEMP = 1
I


I
Compute
XBAR, YBAR,
HIMAGE
Compute
XSTAR, YSTAR,
RSTAR
J
KTEMP = 2
LAMDAT =
LAMDAY
Print minimum
rms X
plane
data
Yes
KTEMP = 1 ?
No
J
KTEMP = 3
LAMDAT =
LAMDAR
Print minimum
rms Y
plane
data
Yes
KTEMP = 2?
No
L
Compute
XBAR, YBAR,
HIMAGE
K = 1
Print minimum
spot size
plane
data
Compute
XSTAR, YSTAR,
RSTAR
Print data for
Kth. plane
st. loc.
LAMDA(K)
K = K + 1
K


K
K > NIPLN?
No
L
Yes
I = I + 1
I = 1
J = J + 1
Yes
I > NCLRS?
No
M
J > NOBJH?
No
M
Yes
RETURN

3.7 LINK 7

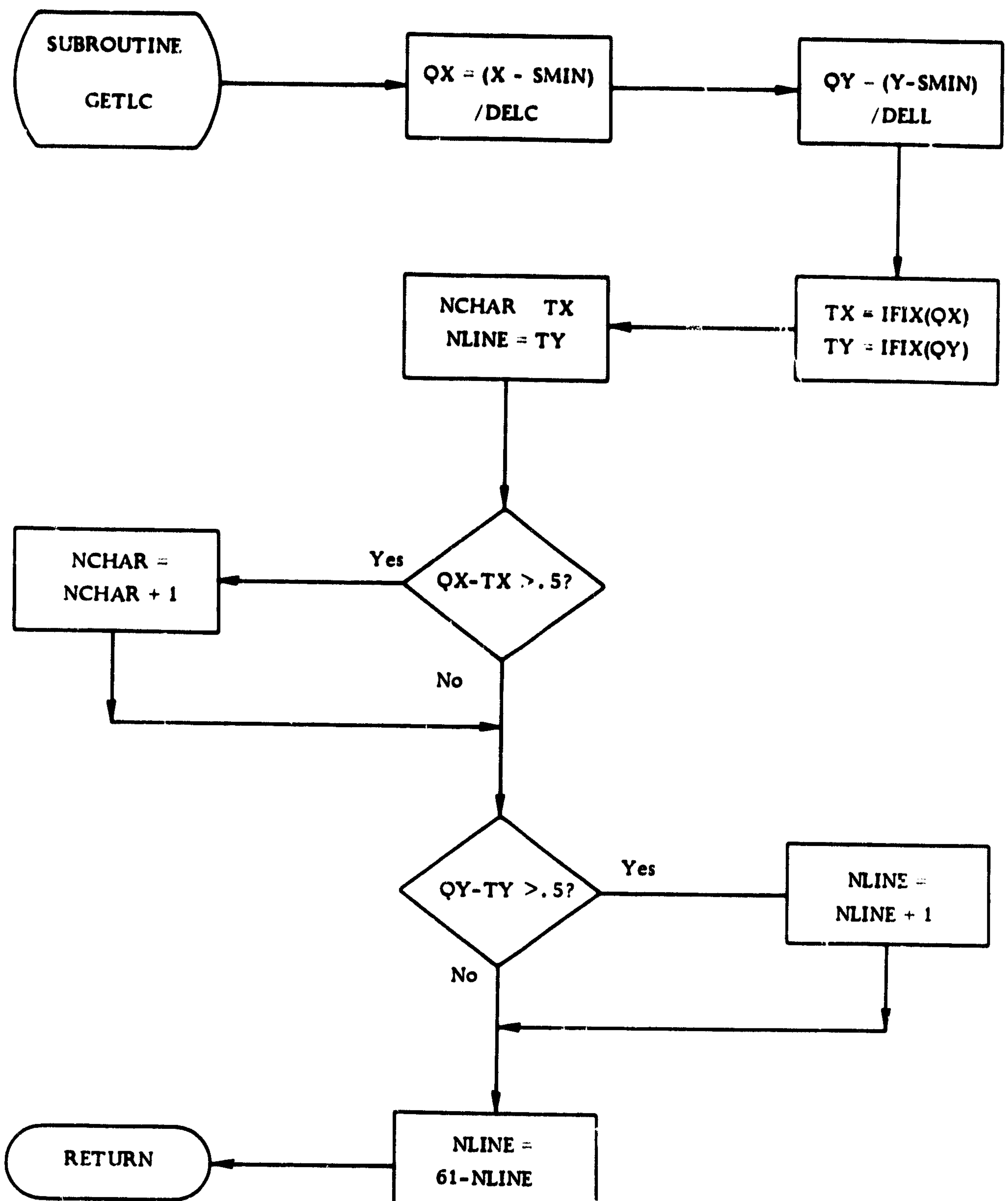
Link 7 is responsible for producing scaled spot diagram points plots on the printer-one plot for each object point-color combination. Subroutine STAR7 which is the control routine is entered directly from FOLDP whenever option 7 is to be executed. Subroutine SMPOP (Link 4) is employed to produce image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


SUBROUTINE GETLC
QX = (X - SMIN) /DELC
QY = (Y-SMIN) /DELL
TX = IFIX(QX)
TY = IFIX(QY)
NCHAR = TX
NLINE = TY
QX-TX >.5?
Yes
No
NCHAR = NCHAR + 1
QY-TY >.5?
Yes
No
NLINE = NLINE + 1
NLINE = 61-NLINE
RETURN

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.

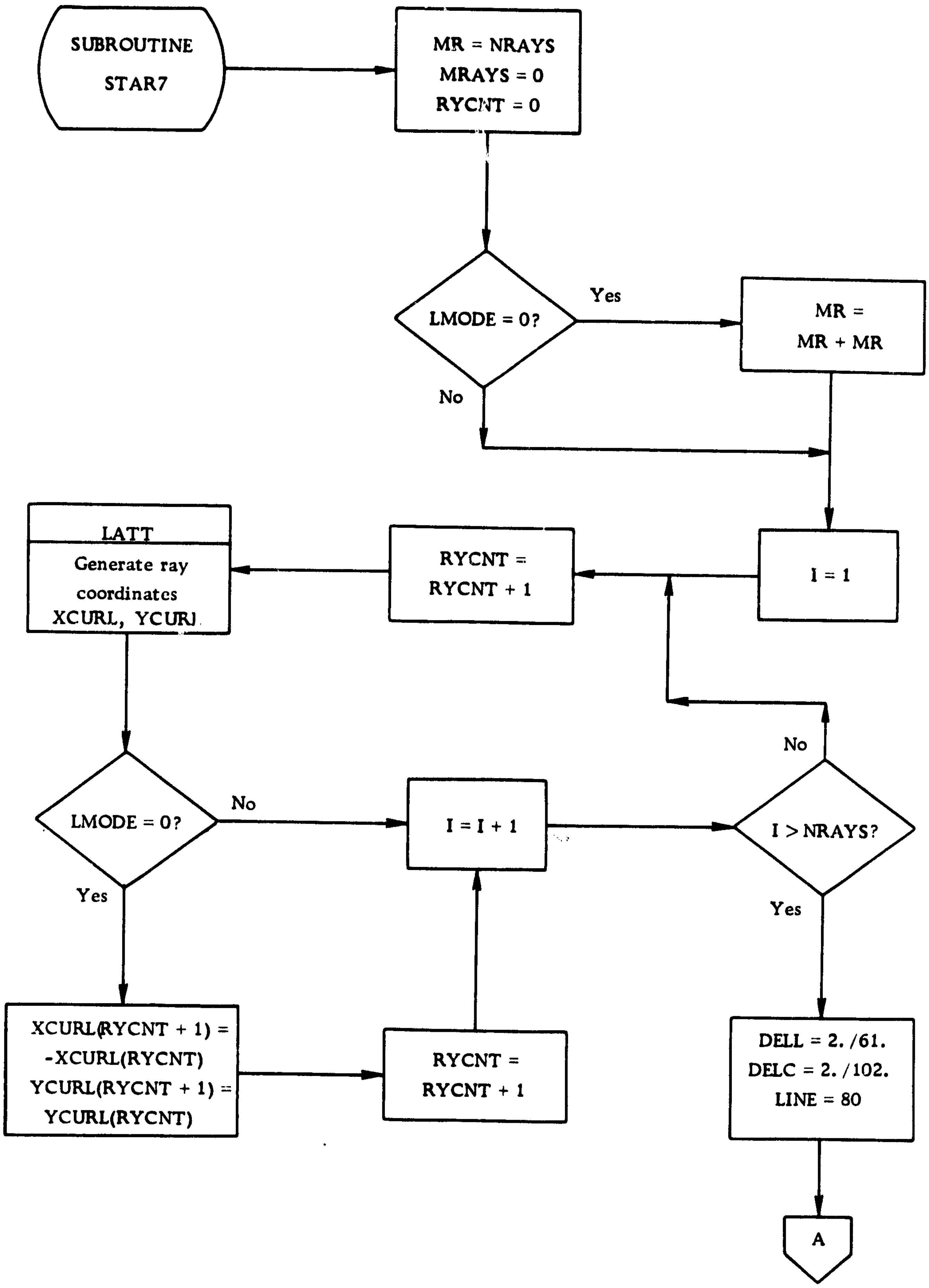
If any ray has failed (missed, reflected, or vignetted) an appropriate comment is printed in the tabular listing and the respective point is deleted from the plot.

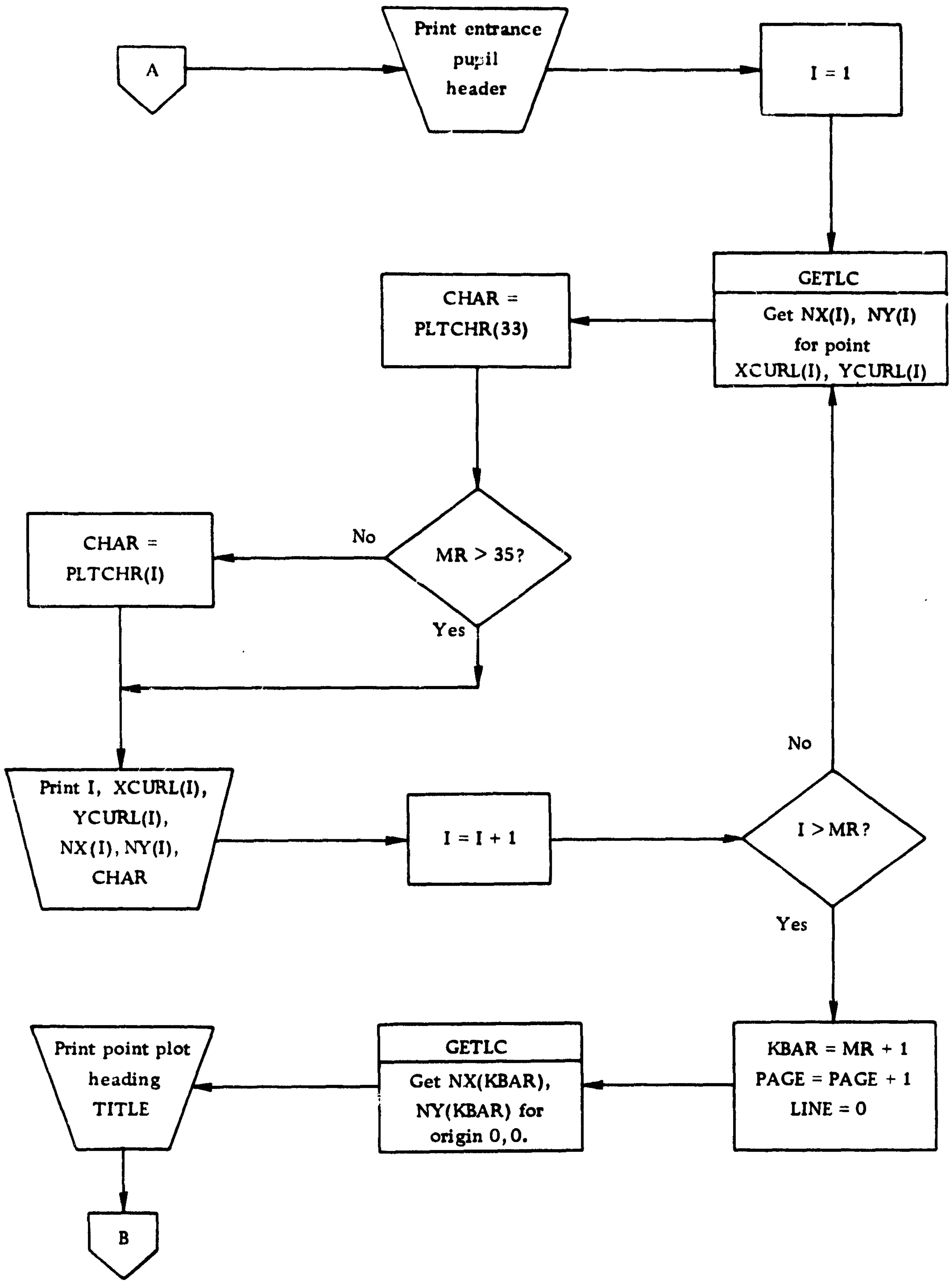
Calling Sequence

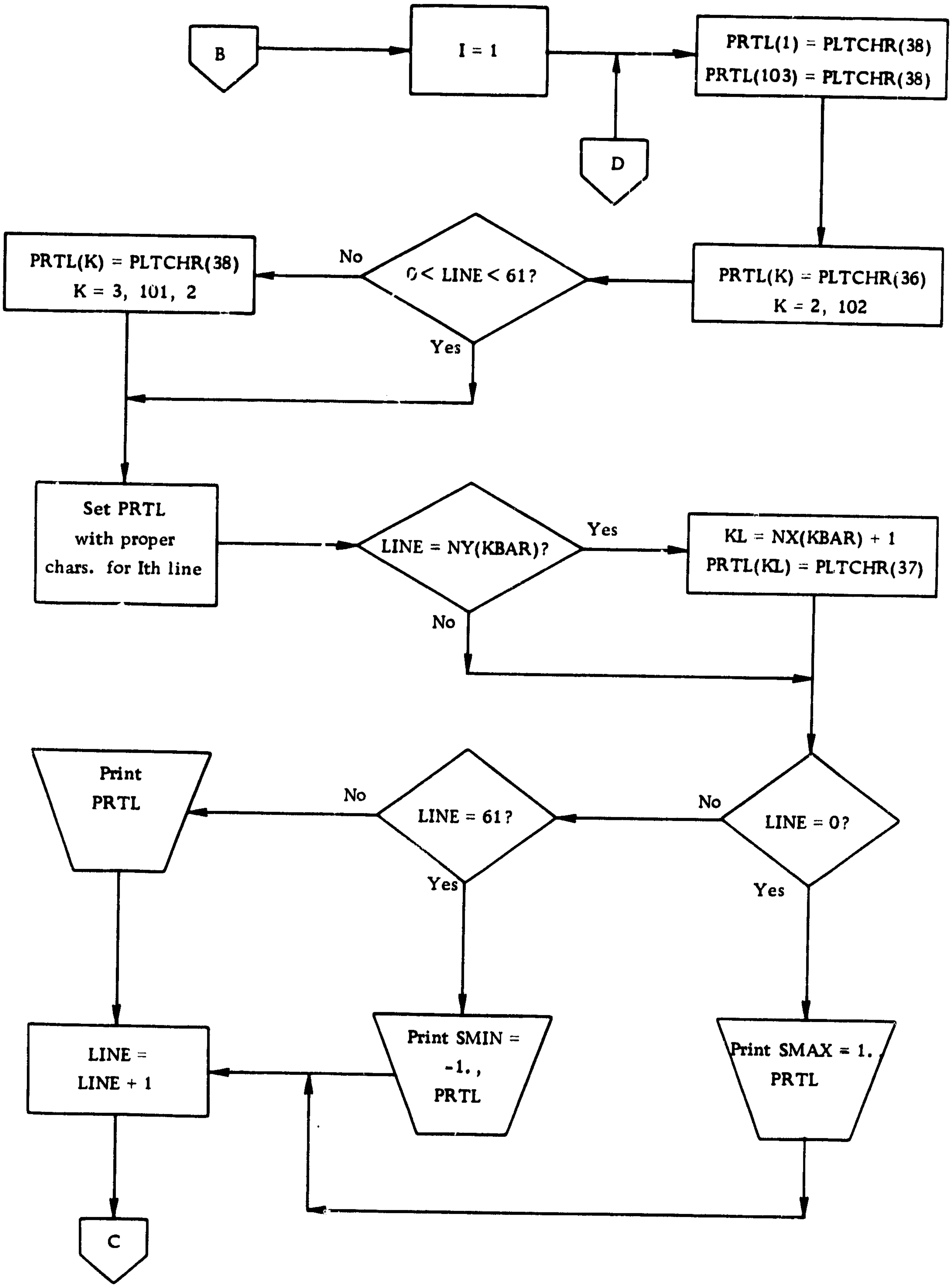
CALL STAR7

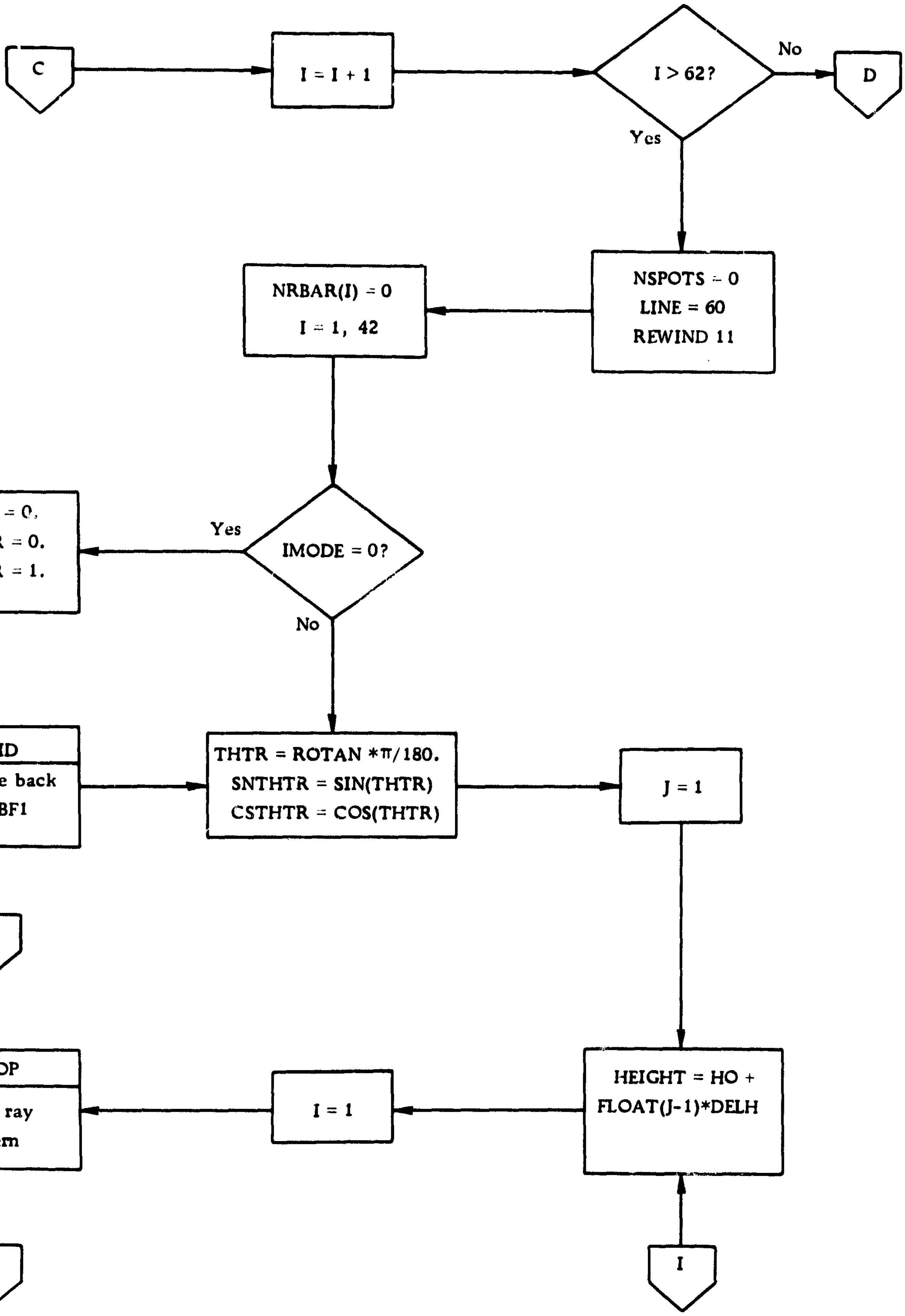
Utility Routines and Common References

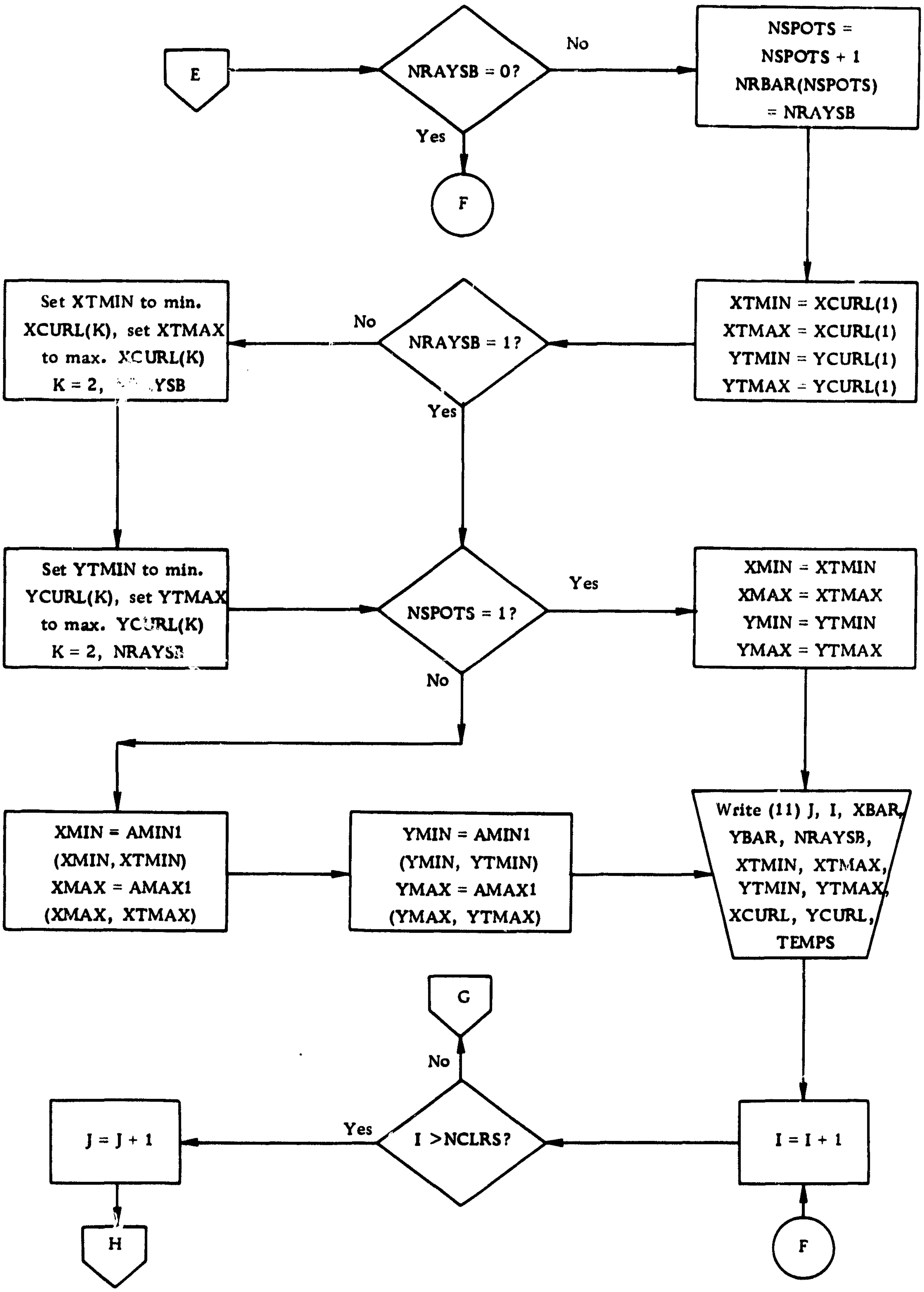
COMMON/DATA/	MERID
COMMON/TMPATT/	SMPOP
COMMON/PRNT/	GETLC
COMMON/AZOBJ/	SCALEK
LATT	PRTCTL

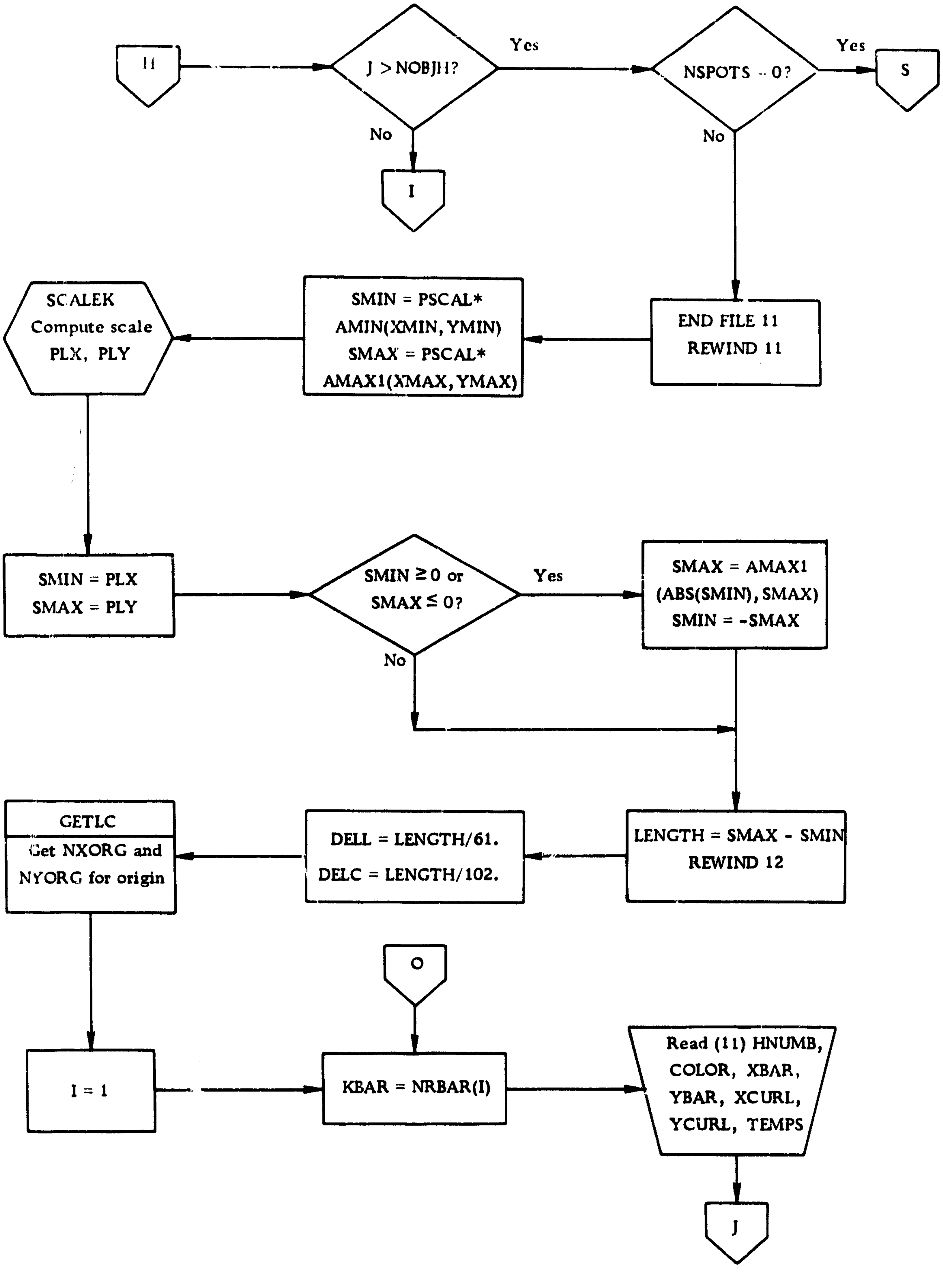

SUBROUTINE STAR7
MR = NRAYS
MRAYS = 0
RYCNT = 0
LMODE = 0?
Yes
MR = MR + MR
No
I = 1
RYCNT = RYCNT + 1
LATT
Generate ray coordinates XCURL, YCURL
LMODE = 0?
No
I = I + 1
I > NRAYS?
No
Yes
Yes
XCURL(RYCNT + 1) = -XCURL(RYCNT)
YCURL(RYCNT + 1) = YCURL(RYCNT)
RYCNT = RYCNT + 1
DELL = 2./61.
DELC = 2./102.
LINE = 80
A

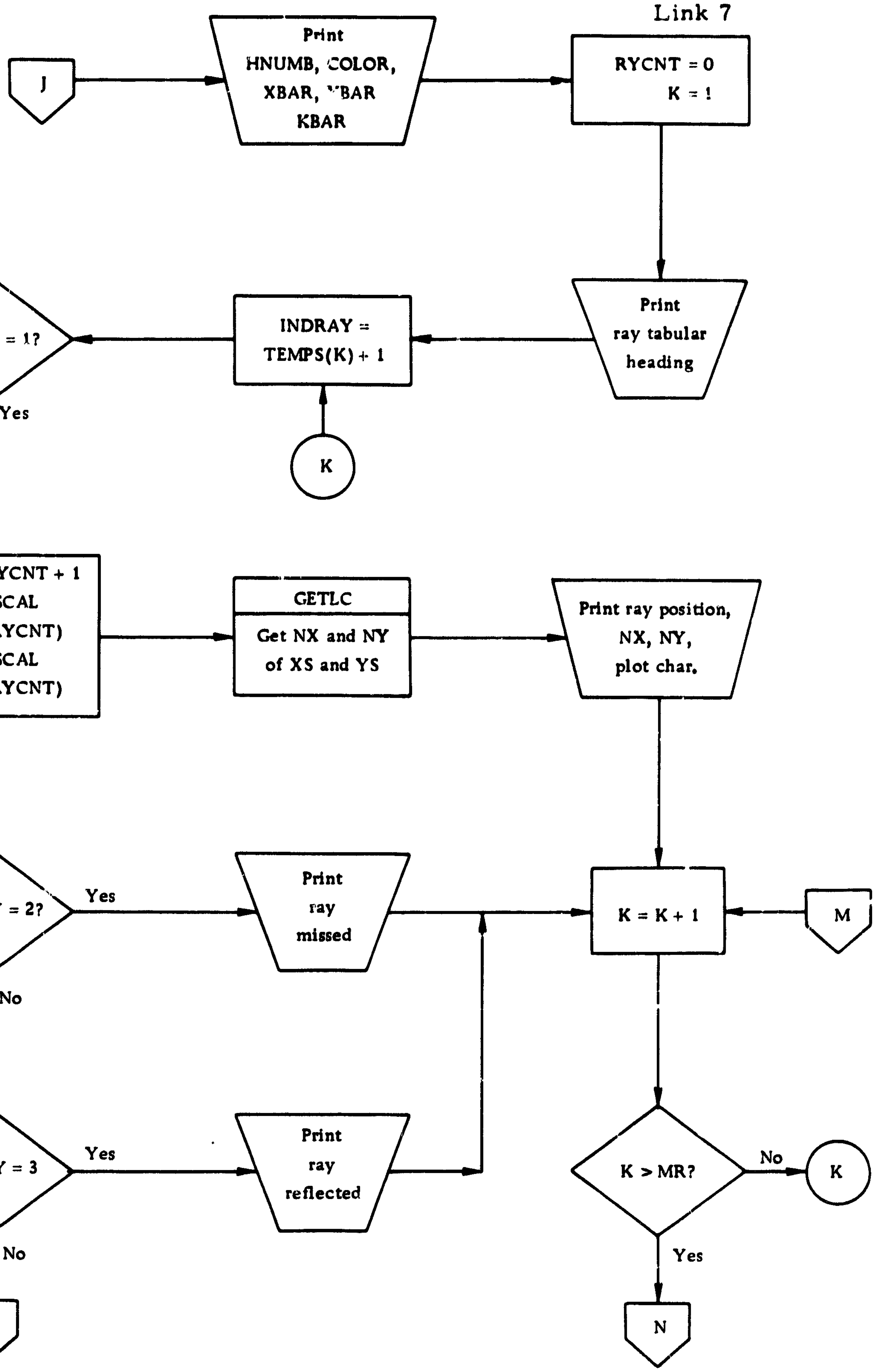

A
Print entrance pupil header
I = 1
GETLC
Get NX(I), NY(I) for point XCURL(I), YCURL(I)
CHAR = PLTCHR(33)
MR > 35?
No
Yes
CHAR = PLTCHR(I)
Print I, XCURL(I), YCURL(I), NX(I), NY(I), CHAR
I = I + 1
I > MR?
No
Yes
KBAR = MR + 1
PAGE = PAGE + 1
LINE = 0
GETLC
Get NX(KBAR), NY(KBAR) for origin 0, 0.
Print point plot heading TITLE
B

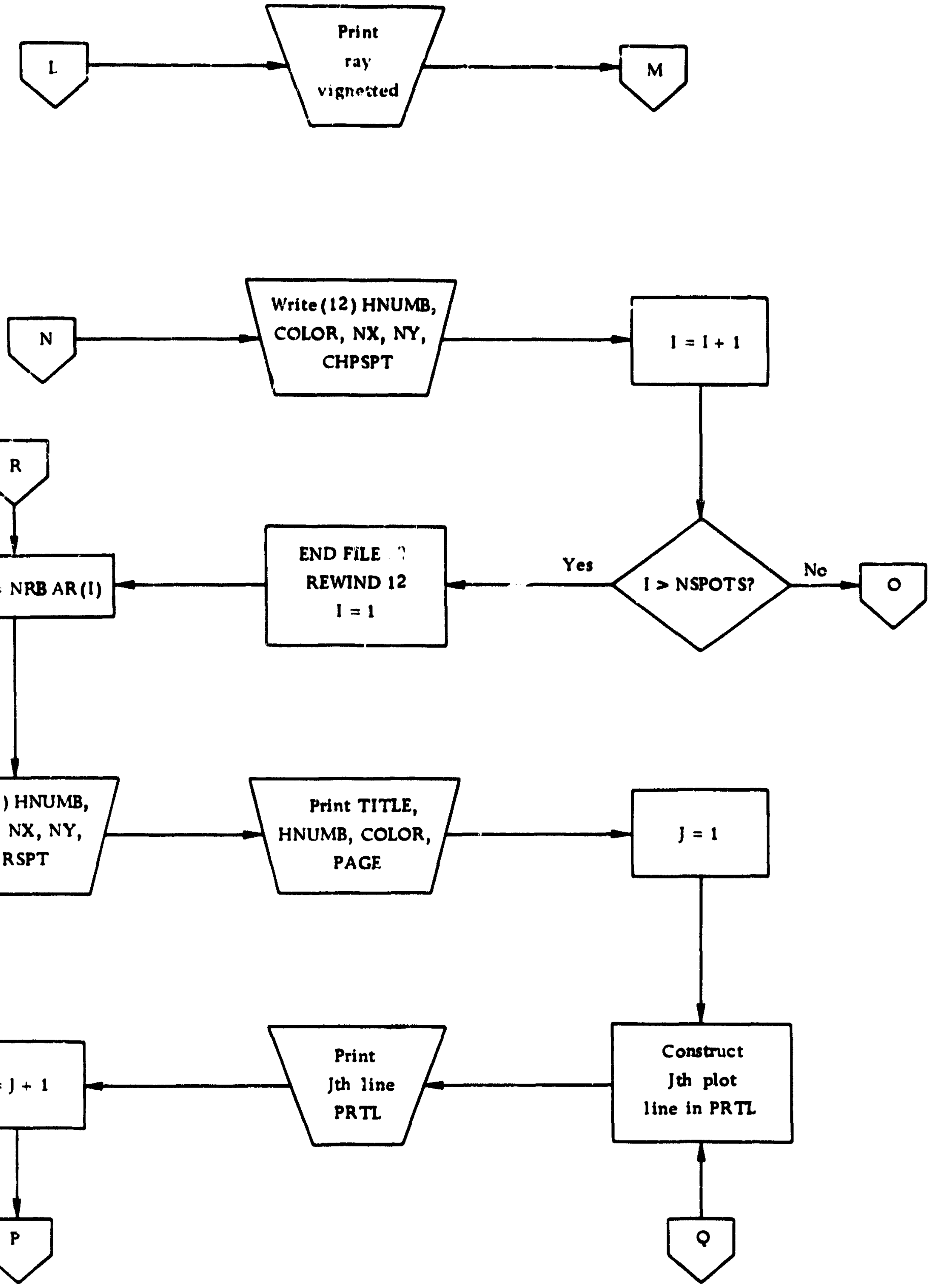

B
I = 1
PRTL(1) = PLTCHR(38)
PRTL(103) = PLTCHR(38)
D
PRTL(K) = PLTCHR(38)
K = 3, 101, 2
No
0 < LINE < 61?
Yes
PRTL(K) = PLTCHR(36)
K = 2, 102
Set PRTL with proper chars. for Ith line
LINE = NY(KBAR)?
Yes
No
KL = NX(KBAR) + 1
PRTL(KL) = PLTCHR(37)
Print PRTL
No
LINE = 61?
Yes
No
LINE = 0?
Yes
LINE = LINE + 1
Print SMIN = -1., PRTL
Print SMAX = 1., PRTL
C

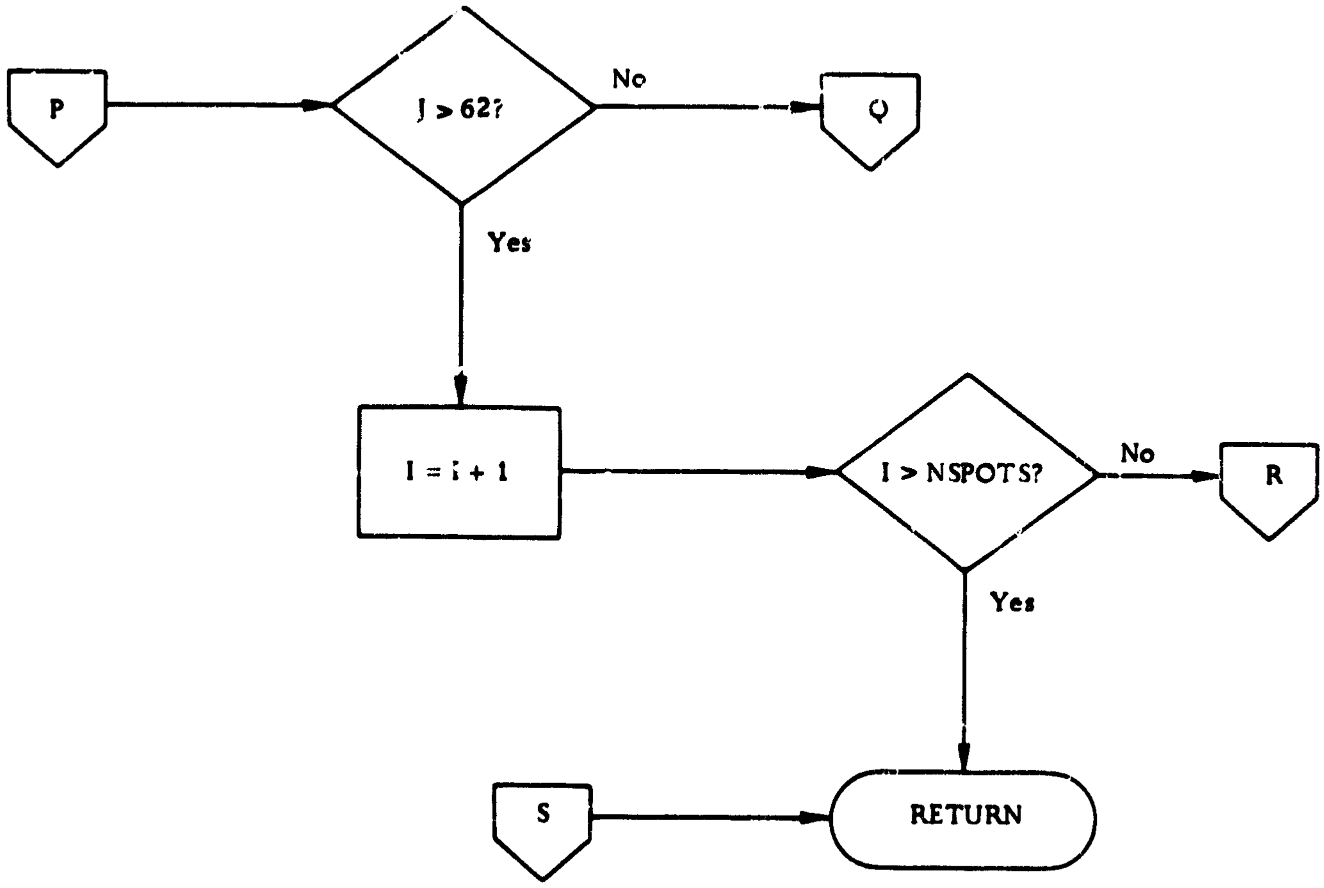

C
I = I + 1
I > 62?
No
D
Yes
NSPOTS = 0
LINE = 60
REWIND 11
NRBAR(I) = 0
I = 1, 42
IMODE = 0?
Yes
No
THTR = 0.
SNTHTR = 0.
CSTHTR = 1.
MERID
Compute back
focus BF1
THTR = ROTAN *π/180.
SNTHTR = SIN(THTR)
CSTHTR = COS(THTR)
J = 1
G
SMPOP
Trace ray
pattern
E
I = 1
HEIGHT = HO +
FLOAT(J-1)*DELH
I


E
NRAYSB = 0?
No
Yes
F
NSPOTS = NSPOTS + 1
NRBAR(NSPOTS) = NRAYSB
XTMIN = XCURL(1)
XTMAX = XCURL(1)
YTMIN = YCURL(1)
YTMAX = YCURL(1)
NRAYSB = 1?
No
Yes
Set XTMIN to min. XCURL(K), set XTMAX to max. XCURL(K) K = 2,
YSB
Set YTMIN to min. YCURL(K), set YTMAX to max. YCURL(K) K = 2, NRAYSB
NSPOTS = 1?
Yes
No
XMIN = XTMIN
XMAX = XTMAX
YMIN = YTMIN
YMAX = YTMAX
XMIN = AMIN1 (XMIN, XTMIN)
XMAX = AMAX1 (XMAX, XTMAX)
YMIN = AMIN1 (YMIN, YTMIN)
YMAX = AMAX1 (YMAX, YTMAX)
Write (11) J, I, XBAR, YBAR, NRAYSB, XTMIN, XTMAX, YTMIN, YTMAX, XCURL, YCURL, TEMPS
G
No
Yes
I >NCLRS?
I = I + 1
J = J + 1
H
F


H
J > NOBJH?
Yes
No
I
NSPOTS = 0?
Yes
S
No
END FILE 11
REWIND 11
SMIN = PSCAL*
AMIN(XMIN, YMIN)
SMAX = PSCAL*
AMAX1(XMAX, YMAX)
SCALEK
Compute scale
PLX, PLY
SMIN = PLX
SMAX = PLY
SMIN ≥ 0 or
SMAX ≤ 0?
Yes
No
SMAX = AMAX1
(ABS(SMIN), SMAX)
SMIN = -SMAX
LENGTH = SMAX - SMIN
REWIND 12
DELL = LENGTH/61.
DELC = LENGTH/102.
GETLC
Get NXORG and
NYORG for origin
O
I = 1
KBAR = NRBAR(I)
Read (11) HNUMB,
COLOR, XBAR,
YBAR, XCURL,
YCURL, TEMPS
J


J
Print
HNUMB, COLOR,
XBAR, YBAR
KBAR
RYCNT = 0
K = 1
Print
ray tabular
heading
INDRAY =
TEMPS(K) + 1
K
INDRAY = 1?
No
Yes
RYCNT = RYCNT + 1
XS = PSCAL
**CURL(RYCNT)
YS = PSCAL
*YCURL(RYCNT)
GETLC
Get NX and NY
of XS and YS
Print ray position,
NX, NY,
plot char.
INDRAY = 2?
Yes
No
Print
ray
missed
K = K + 1
M
INDRAY = 3
Yes
No
Print
ray
reflected
L
K > MR?
No
K
Yes
N


L
Print
ray
vignetted
M
N
Write (12) HNUMB,
COLOR, NX, NY,
CHPSPT
I = I + 1
R
KBAR = NRBAR(I)
END FILE 12
REWIND 12
I = 1
Yes
I > NSPOTS?
No
O
Read (12) HNUMB,
COLOR, NX, NY,
CHRSPT
Print TITLE,
HNUMB, COLOR,
PAGE
J = 1
J = J + 1
Print
Jth line
PRTL
Construct
Jth plot
line in PRTL
P
Q


P
J > 62?
No
Q
Yes
I = i + 1
I > NSPOTS?
No
R
Yes
S
RETURN

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/

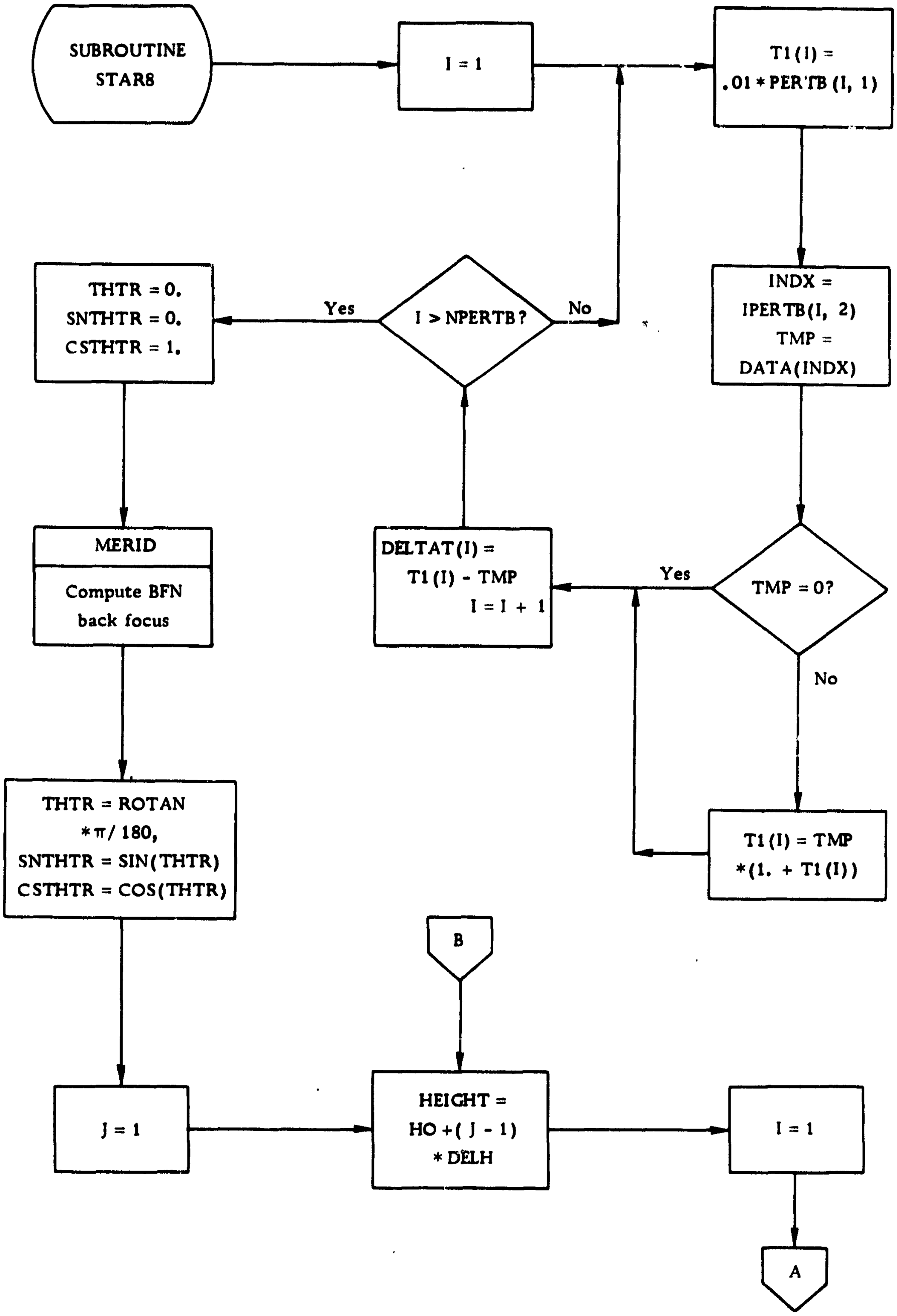
COMMON/AZOBJ/

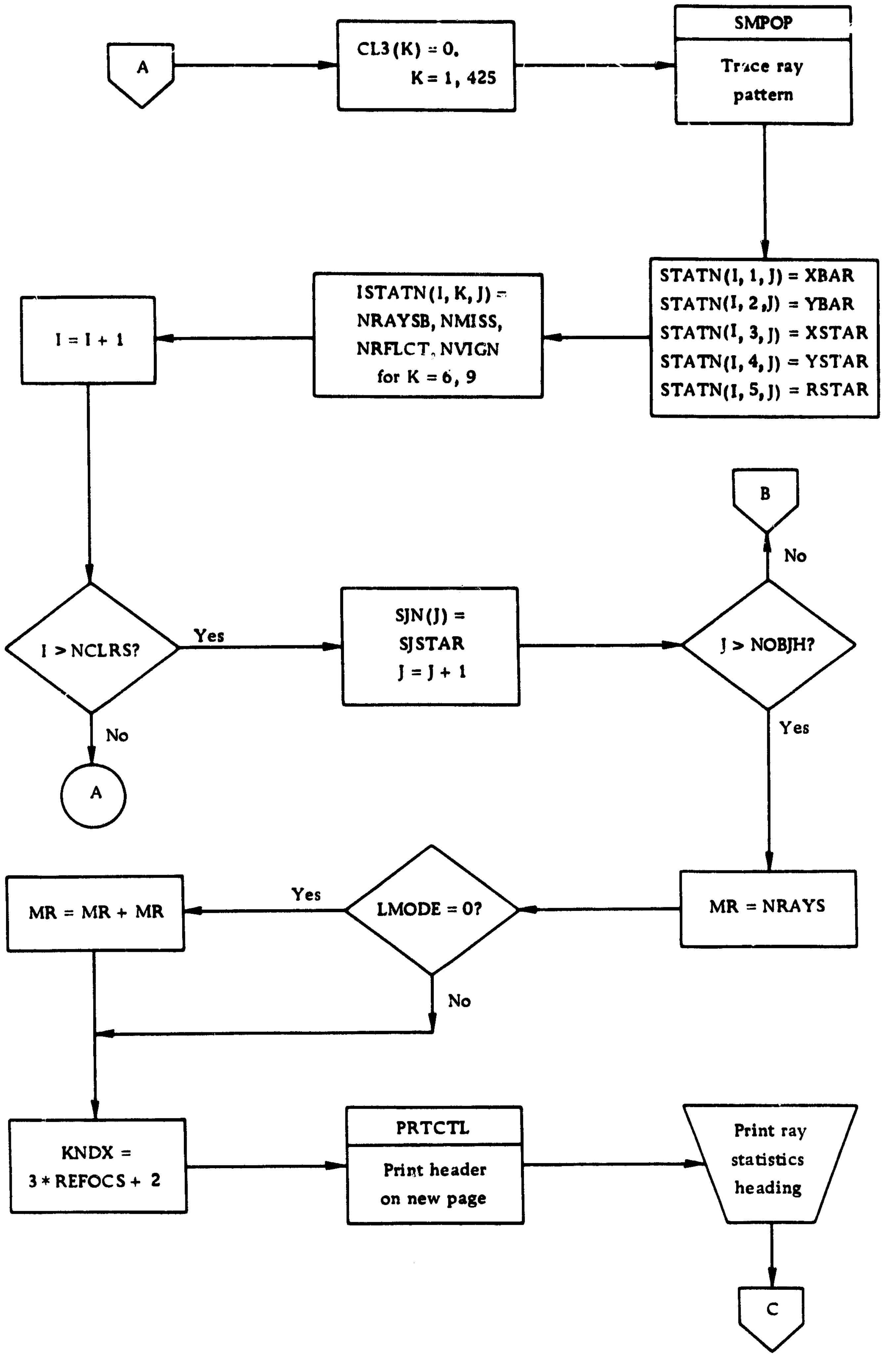
COMMON/PRNT/

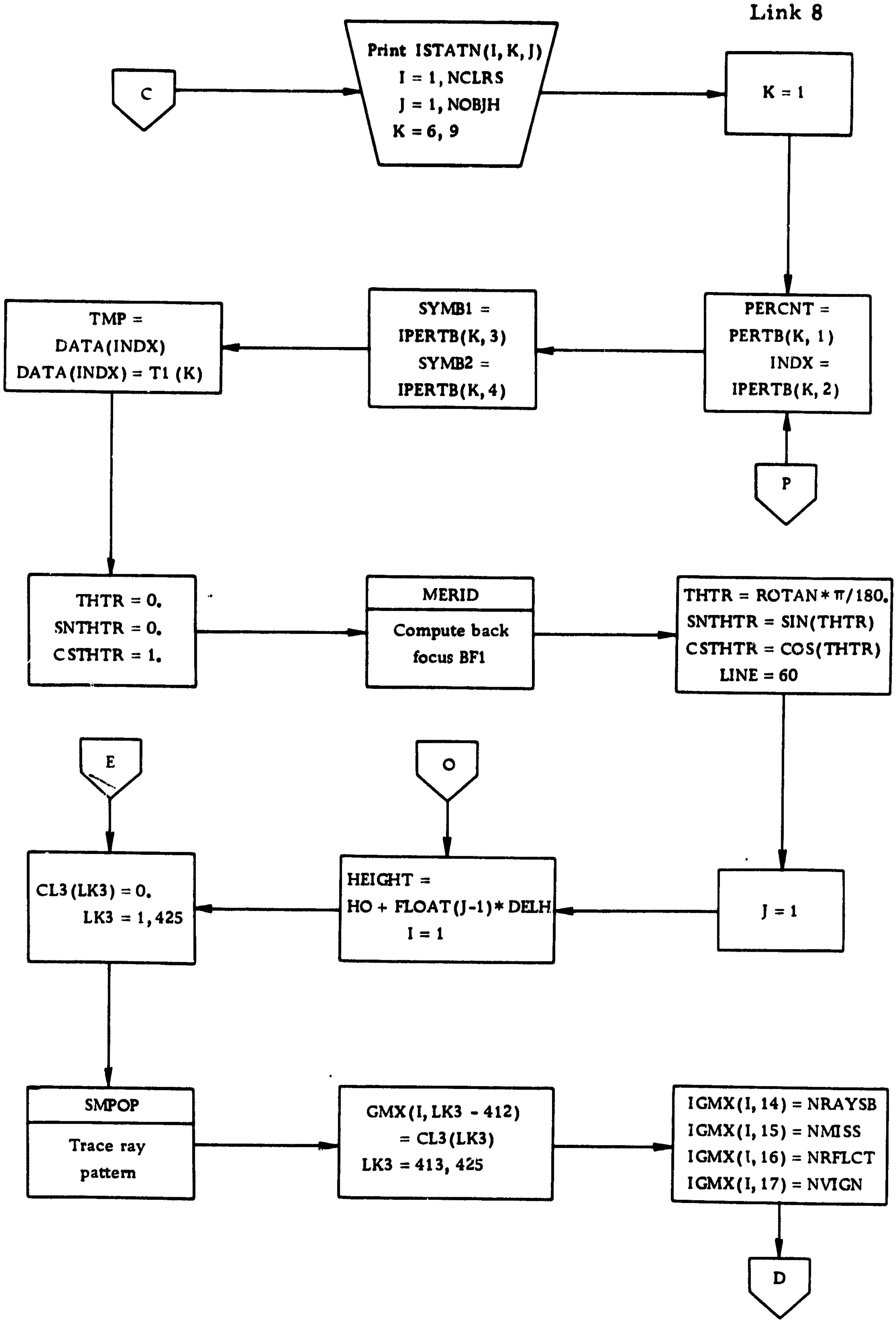
SMPOP

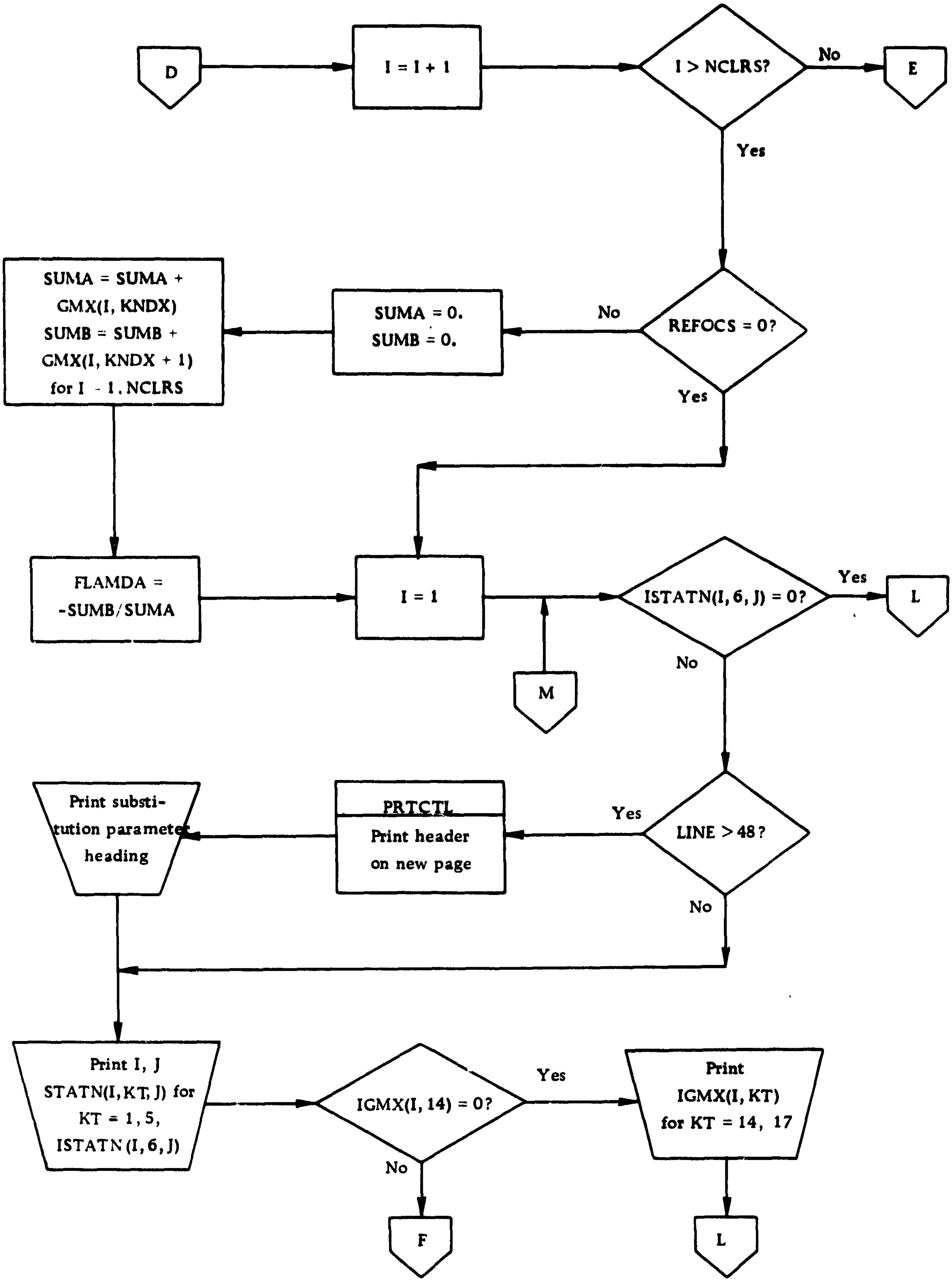
MERID

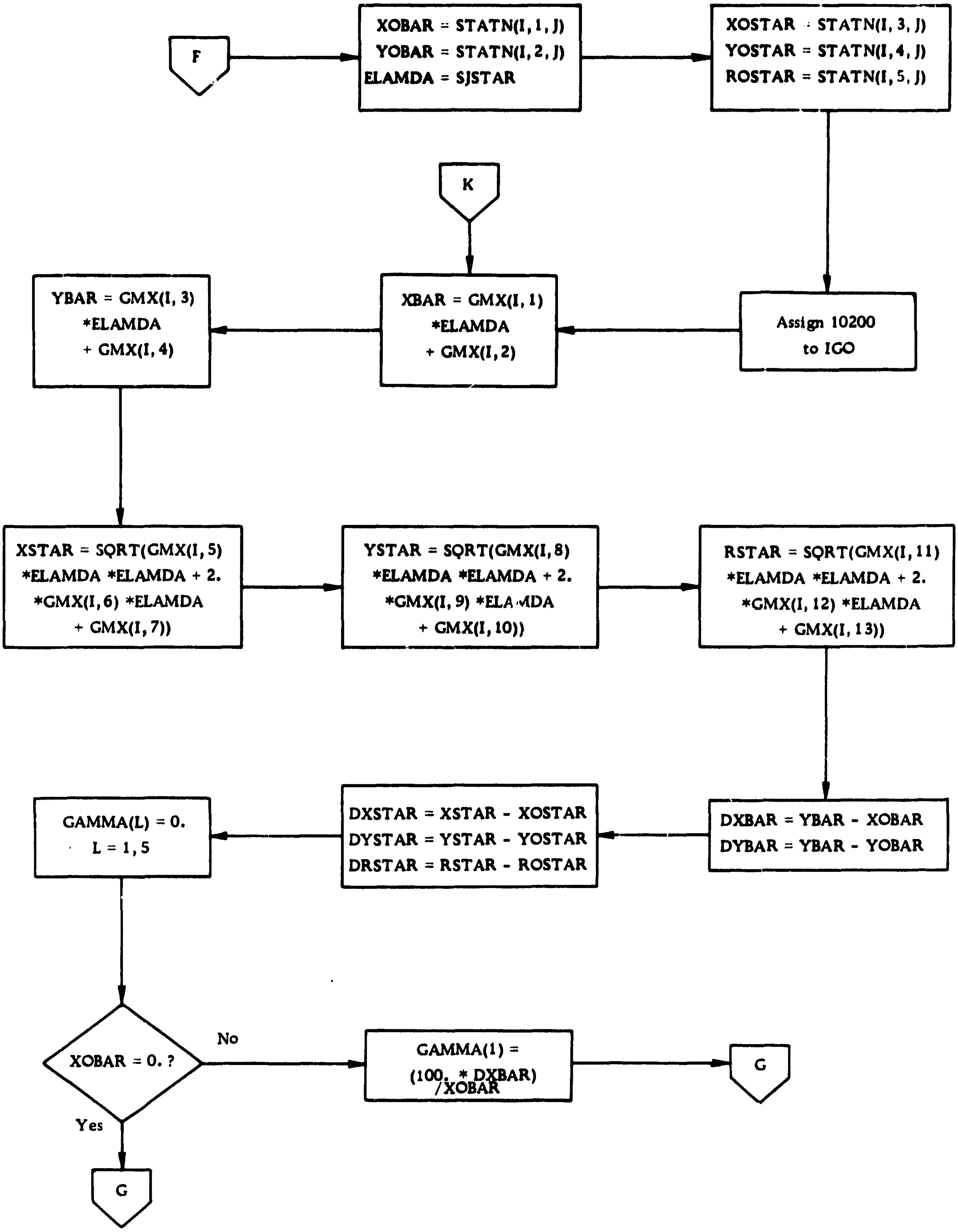
PRTCTL

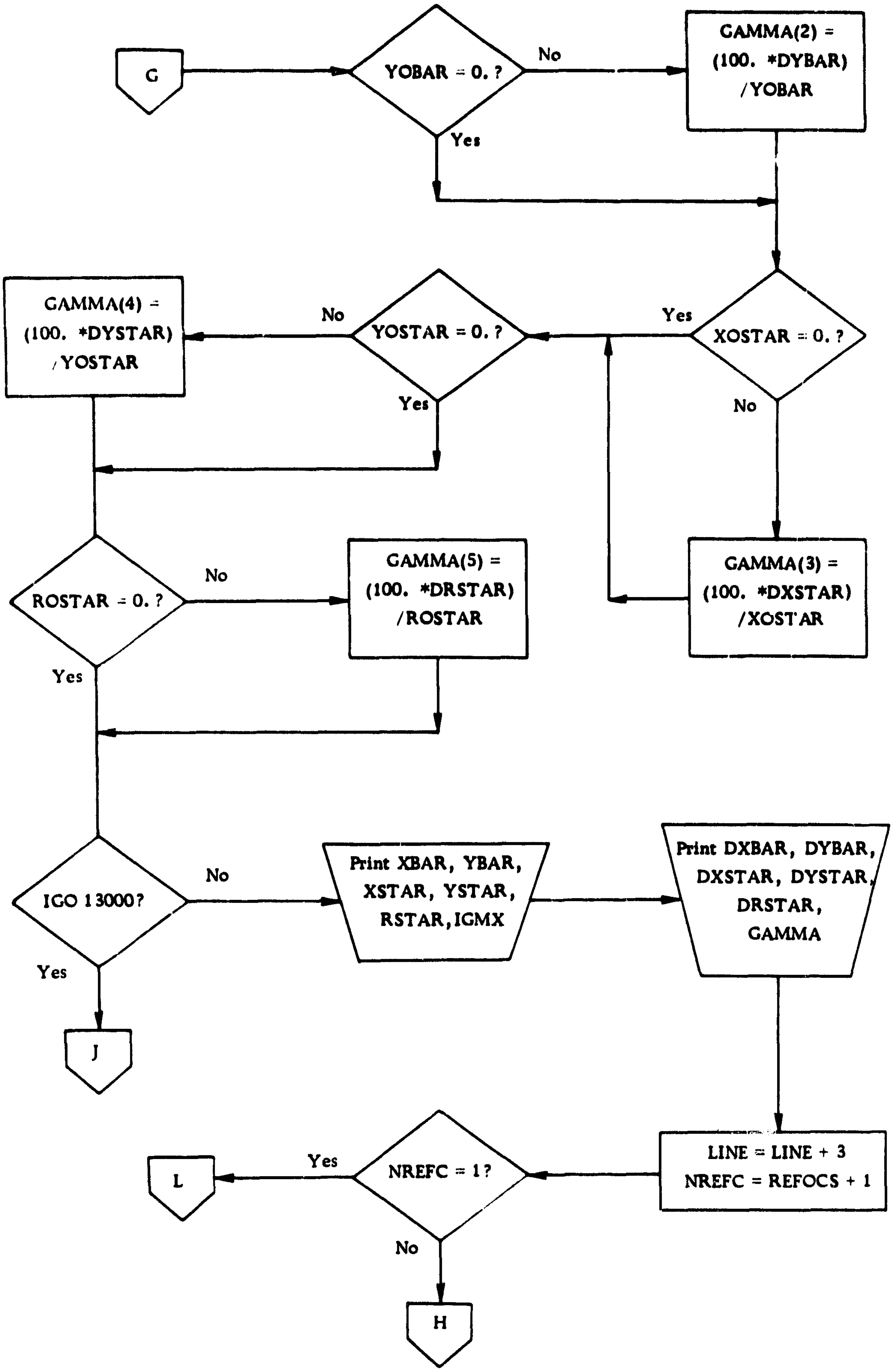

SUBROUTINE STAR8
I = 1
T1(I) = .01 * PERTB(I, 1)
INDX = IPERTB(I, 2)
TMP = DATA(INDX)
TMP = 0?
Yes
No
T1(I) = TMP *(1. + T1(I))
DELTAT(I) = T1(I) - TMP
I = I + 1
I > NPERTB?
Yes
No
THTR = 0.
SNTHTR = 0.
CSTHTR = 1.
MERID
Compute BFN back focus
THTR = ROTAN *π/ 180,
SNTHTR = SIN(THTR)
CSTHTR = COS(THTR)
J = 1
B
HEIGHT = HO +(J - 1) * DELH
I = 1
A

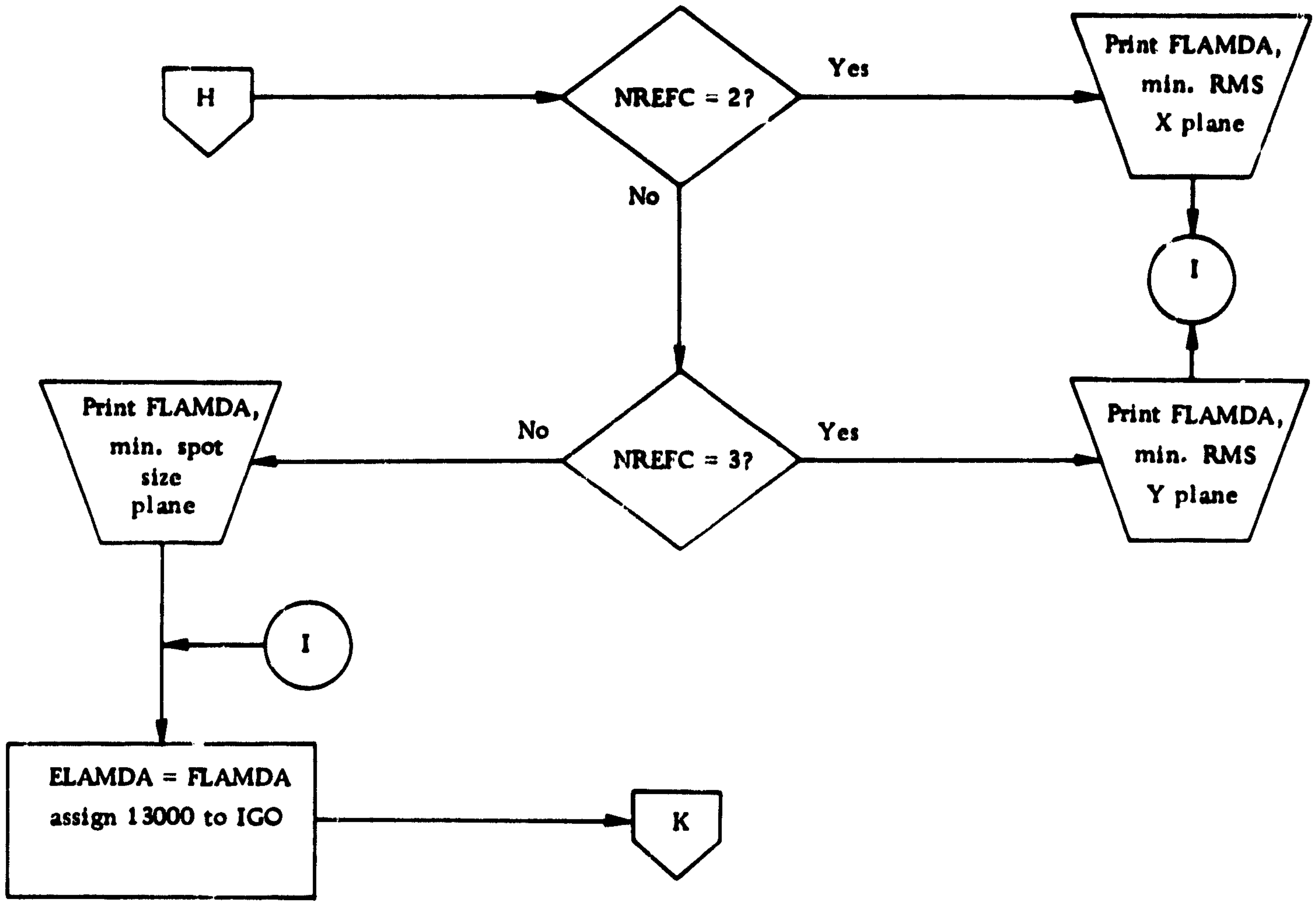

A
CL3(K) = 0,
K = 1, 425
SMPOP
Trace ray
pattern
STATN(I, 1, J) = XBAR
STATN(I, 2,J) = YBAR
STATN(I, 3, J) = XSTAR
STATN(I, 4, J) = YSTAR
STATN(I, 5, J) = RSTAR
ISTATN(I, K, J) =
NRAYSB, NMISS,
NRFLCT, NVIGN
for K = 6, 9
I = I + 1
I > NCLRS?
Yes
No
A
SJN(J) =
SJSTAR
J = J + 1
J > NOBJH?
No
B
Yes
MR = NRAYS
LMODE = 0?
Yes
No
MR = MR + MR
KNDX =
3 * REFOCS + 2
PRTCTL
Print header
on new page
Print ray
statistics
heading
C

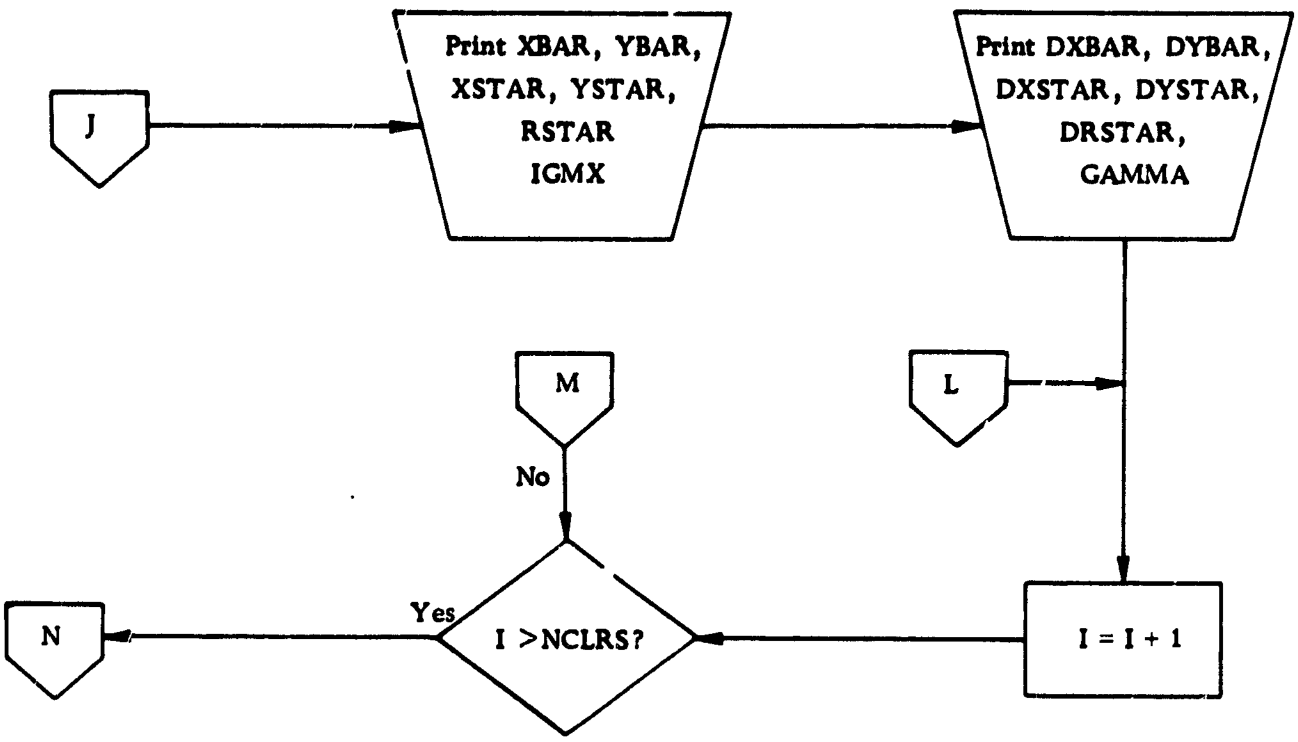

C
Print ISTATN(I, K, J)
I = 1, NCLRS
J = 1, NOBJH
K = 6, 9
K = 1
PERCNT =
PERTB(K, 1)
INDX =
IPERTB(K, 2)
P
SYMB1 =
IPERTB(K, 3)
SYMB2 =
IPERTB(K, 4)
TMP =
DATA(INDX)
DATA(INDX) = T1 (K)
THTR = 0.
SNTHTR = 0.
CSTHTR = 1.
MERID
Compute back
focus BF1
THTR = ROTAN * π/180.
SNTHTR = SIN(THTR)
CSTHTR = COS(THTR)
LINE = 60
E
O
J = 1
HEIGHT =
HO + FLOAT(J-1) * DELH
I = 1
CL3(LK3) = 0.
LK3 = 1, 425
SMPOP
Trace ray
pattern
GMX(I, LK3 - 412)
= CL3(LK3)
LK3 = 413, 425
IGMX(I, 14) = NRAYSB
IGMX(I, 15) = NMISS
IGMX(I, 16) = NRFLCT
IGMX(I, 17) = NVIGN
D

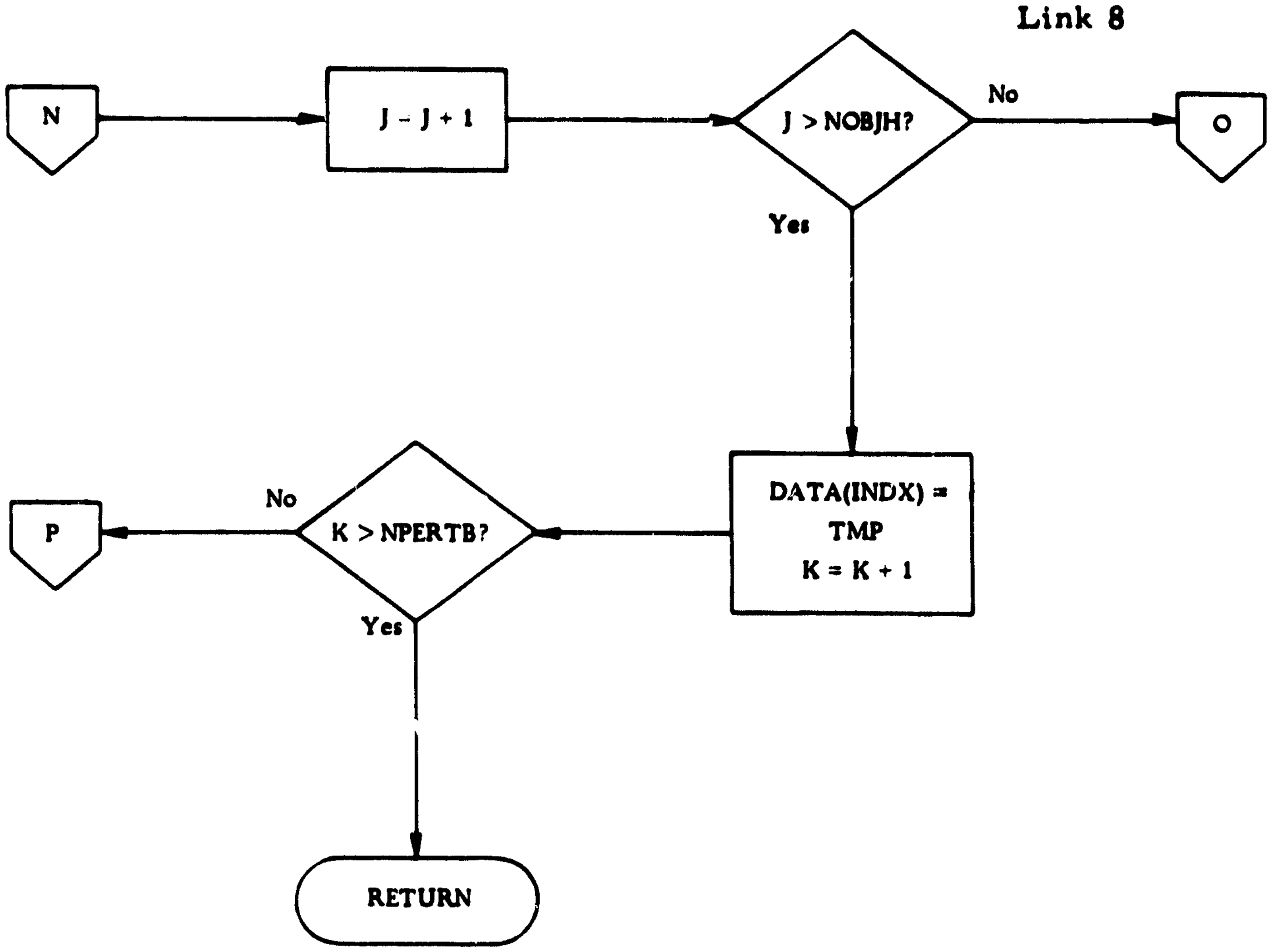

D
I = I + 1
I > NCLRS?
No
E
Yes
REFOCS = 0?
No
SUMA = 0.
SUMB = 0.
SUMA = SUMA +
GMX(I, KNDX)
SUMB = SUMB +
GMX(I, KNDX + 1)
for I - 1, NCLRS
Yes
FLAMDA =
-SUMB/SUMA
I = 1
ISTATN(I, 6, J) = 0?
Yes
L
M
No
LINE > 48?
Yes
PRTCTL
Print header
on new page
Print substi-
tution parameter
heading
No
Print I, J
STATN(I, KT, J) for
KT = 1, 5,
ISTATN (I, 6, J)
IGMX(I, 14) = 0?
Yes
Print
IGMX(I, KT)
for KT = 14, 17
No
F
L


F
XOBAR = STATN(I, 1, J)
YOBAR = STATN(I, 2, J)
ELAMDA = SJSTAR
XOSTAR = STATN(I, 3, J)
YOSTAR = STATN(I, 4, J)
ROSTAR = STATN(I, 5, J)
K
Assign 10200 to IGO
XBAR = GMX(I, 1) *ELAMDA + GMX(I, 2)
YBAR = GMX(I, 3) *ELAMDA + GMX(I, 4)
XSTAR = SQRT(GMX(I, 5) *ELAMDA *ELAMDA + 2. *GMX(I, 6) *ELAMDA + GMX(I, 7))
YSTAR = SQRT(GMX(I, 8) *ELAMDA *ELAMDA + 2. *GMX(I, 9) *ELAMDA + GMX(I, 10))
RSTAR = SQRT(GMX(I, 11) *ELAMDA *ELAMDA + 2. *GMX(I, 12) *ELAMDA + GMX(I, 13))
DXBAR = YBAR - XOBAR
DYBAR = YBAR - YOBAR
DXSTAR = XSTAR - XOSTAR
DYSTAR = YSTAR - YOSTAR
DRSTAR = RSTAR - ROSTAR
GAMMA(L) = 0.
L = 1, 5
XOBAR = 0. ?
No
GAMMA(1) = (100. * DXBAR) /XOBAR
G
Yes
G


G
YOBAR = 0. ?
No
Yes
GAMMA(2) = (100. *DYBAR) /YOBAR
XOSTAR = 0. ?
Yes
No
GAMMA(3) = (100. *DXSTAR) /XOSTAR
YOSTAR = 0. ?
No
Yes
GAMMA(4) = (100. *DYSTAR) /YOSTAR
ROSTAR = 0. ?
No
Yes
GAMMA(5) = (100. *DRSTAR) /ROSTAR
IGO 13000?
No
Yes
J
Print XBAR, YBAR, XSTAR, YSTAR, RSTAR, IGMX
Print DXBAR, DYBAR, DXSTAR, DYSTAR, DRSTAR, GAMMA
LINE = LINE + 3
NREFC = REFOCS + 1
NREFC = 1?
Yes
L
No
H


H
NREFC = 2?
Yes
No
Print FLAMDA, min. RMS X plane
I
NREFC = 3?
No
Yes
Print FLAMDA, min. spot size plane
Print FLAMDA, min. RMS Y plane
I
ELAMDA = FLAMDA assign 13000 to IGO
K


J
Print XBAR, YBAR, XSTAR, YSTAR, RSTAR IGMX
Print DXBAR, DYBAR, DXSTAR, DYSTAR, DRSTAR, GAMMA
M
L
No
N
Yes
I >NCLRS?
I = I + 1


N
J = J + 1
J > NOBJH?
No
O
Yes
DATA(INDX) =
TMP
K = K + 1
K > NPERTB?
No
P
Yes
RETURN

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 <u>not</u> 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 <u>in core</u> 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:

$$M \leq \frac{NDIM}{n+2}. \tag{1}$$

---

* 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 occuring 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 $\vec{I}_1, \vec{I}_2, \ldots, \vec{I}_n$ 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

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

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 (6O12) 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 $\theta_1, \theta_2, \ldots, \theta_K$ 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.

STORAGE MAP

```
      $JOB   INF,N010001,44903-0,77387,AM IC1,10,10000  FIRNETT INFORMATICS          A 08"24""35"" 10/01/68
$IBJOB           MAP,NOGO
$IBLDR FOLDF                                                            09/19/68    FOLD0000
$IBLDR ERRORZ                                                           09/19/68    ERRO0000
$IBLDR GOOFZZ                                                           09/19/68    GOOF0000
$IBLDR PRMSZZ                                                           09/19/68    PRMS0000
$IBLDR PRTCZZ                                                           09/19/68    PRTC0000
$IBLDR RAYTRZ                                                           09/19/68    RAYT0000
$ORIGIN LINK1   ORG1      (READSZ,STAR9Z,STAR1Z,REREZZ,OPUNCH,MTFFNZ)
$IBLDR OPUNCZ                                                           09/19/68    OPUN0000
$IBLDR READSZ                                                           09/19/68    READ0000
$IBLDR REREZZ                                                           09/19/68    RERE0000
$IBLDR STAR1Z                                                           09/19/68    STAR0000
$IBLDR STAR9Z                                                           09/19/68    STAR0000
$IBLDR UNBCD.                                                           09/19/68    UNBC0000
$IBLDR VCNVR.                                                           09/19/68    VCNV0000
$IBLDR MTFFNZ                                                           09/25/68    MTFF0000
$ORIGIN LINK2   ORG1      (PROFIZ,CELAIZ,RANGEZ,ENDPTZ,PFILEZ,CPLOTZ)
$INCLUDE        .ACF.            (LINK 2)
$INCLUDE        OUT.             (LINK 2)
$INCLUDE        GETJB.           (LINK 2)
$INCLUDE        SDINI.           (LINK 2)
$INCLUDE        SDNP.            (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)
$IBLDR ARPTSZ                                                           09/20/68    ARPT0000
$IBLDR CELAIZ                                                           09/20/68    CELA0000
$IBLDR CLOSSZ                                                           09/20/68    CLOS0000
$IBLDR CPLOTZ                                                           09/20/68    CPLO0000
$IBLDR CROSSZ                                                           09/20/68    CROS0000
$IBLDR ENDPTZ                                                           09/20/68    ENDP0000
$IBLDR E1RGCZ                                                           09/20/68    E1RG0000
$IBLDR PFILEZ                                                           09/20/68    PFIL0000
$IBLDR PROFLZ                                                           09/20/68    PROF0000
$IBLDR RANGEZ                                                           09/20/68    RANG0000
$IBLDR ZYCUTZ                                                           09/20/68    ZYCU0000
$ORIGIN LINK3   ORG1      (TWNR4Z,TWNR5Z)
$IBLDR TWNR4Z                                                           09/20/68    TWNR0000
$IBLDR TWNR5Z                                                           09/20/68    TWNR0000
$ORIGIN LINK4   ORG1      (SMPOPZ,MERIDZ,LATTZZ)
$IBLDR LATTZZ                                                           09/20/68    LATT0000
$IBLDR MERIDZ                                                           09/20/68    MERI0000
$IBLDR SMPOPZ                                                           09/20/68    SMPO0000
$ORIGIN LINK5   ORG2      (STAR2Z,CYCLEZ,BNDCHZ,JURYZ,GETRHZ,MWVCTZ,MEVCTZ)
$INCLUDE        MATINV           (LINK 5)
$IBLDR BNDCHZ                                                           09/24/68    BNDC0000
$IBLDR CORDEZ                                                           09/24/68    CORD0000
$IBLDR CYCLEZ                                                           09/24/68    CYCL0000
$IBLDR DOTPZ                                                            09/24/68    DOTP0000
$IBLDR DPRNTZ                                                           09/24/68    DPRN0000
$IBLDR GETRHZ                                                           09/24/68    GETR0000
$IBLDR INCRMZ                                                           09/24/68    INCR0000
$IBLDR JURYZ                                                            09/24/68    JURY0000
```

```
                 $NF,N010001,44903-5,77387,AM IC     IBJOB L10HKC  7094 A    10/01/68          PAGE  1
......
      $IBLDR MEVCTZ                                                    09/24/68    MEVC0000
      $IBLDR MODDLZ                                                    09/24/68    MODD0000
      $IBLDR MWVCTZ                                                    09/24/68    MWVC0000
      $IBLDR STAR2Z                                                    09/24/68    STAR0000
      $ORIGIN LINK6  ORG2      (STAR3Z,PARAXZ)
      $IBLDR PARAXZ                                                    09/24/68    PARA0000
      $IBLDR STAR3Z                                                    09/24/68    STAR0000
      $ORIGIN LINK7  ORG2      (STAR7Z,GETLCZ)
      $IBLDR GETLCZ                                                    09/24/68    GETL0000
      $IBLDR STAR7Z                                                    09/24/68    STAR0000
      $ORIGIN LINK8  ORG2      (STAR8Z)
      $IBLDR STAR8Z                                                    09/24/68    STAR0000
```

```
          IBLDR      INF,N010501,44953-0,77387,AN IC      IBJOB LIGNKC  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       (PROFIZ,CELAIZ,RAIS LINK   2, PARENT LINK IS   0
          INCLUDE       .ACF.                 (LINK 2)
          INCLUDE       OUT.                  (LINK 2)
          INCLUDE       GETJB.                (LINK 2)
          INCLUDE       $DINI.                (LINK 2)
          INCLUDE       $DNPT                 (LINK 2)
          INCLUDE       SETBN                 (LINK 2)
          INCLUDE       $INTR.                (LINK 2)
          INCLUDE       $LABL.                (LINK 2)
          INCLUDE       $PRT.                 (LINK 2)
          INCLUDE       $TERM.                (LINK 2)
          INCLUDE       STORE.                (LINK 2)
          INCLUDE       XXNDT.                (LINK 2)

$ORIGIN LINK3  ORG1       (TWNR4Z,TWNR5Z)  IS LINK   3, PARENT LINK IS   0

$ORIGIN LINK4  ORG1       (SMPOPZ,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
```

```
          IBLDR      INF,N010001,44933-0,77387,AM 1C      IBJOB LIOMKC  7094 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 0 LENGTH, (LOC)=DELETED, *=NOT REFERENCED)

 0  FOLDP   03110      /DATA  / 03112    EVEN    03111    /PERTB / 12004    EVEN    12003   /SYMBLS/ 12176
                       /HEADER/ 14622    /PLOTC / 14624    EVEN    14627    ......  15116 *
    ERRORZ  15132       EVEN    15133    ERROR   15274    ERROR1  15304    ERROR2  15316    ERROR3  15330
                        ERROR4  15342
    GOOFZZ  15352       EVEN    15353    GOOF    15366
    PRMSZZ  15400      /DATA  /(03112)   EVEN    15401    PRMSUD  15541
    PRTCZZ  15566      /DATA  /(03112)   /PRNT  / 15570    EVEN    15567   /HEADER/(14622)  PRTCTL  15662
    RAYTRZ  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 *
                        .LXRTN  20421    IBEXIT  20421 *  .DBCLS  20604 *  .LXARG  20762    .LO     21005 *
                        .CLSE   21013    .LFBL   21014 *  .LUND   21015    .DFOUT  21016
    .IODEF  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 *  .RLHLD  21157 *  .AREA1  21162    .LUNBL  21170
                        .ENTRY  21174    .GOA    21231    .GO     21235    .DERR   21251    .NOPXI  21252
                        .COMXI  21254    .EX34   21276
    .LOVRY  21303       .LOVRY (21303)   .LDT   (20167)   .LRECT (20200)   .LVEC  (20304)
    .LXSL   22010       .LXSEL  22010    .LXSL1  22011    .LXTST  22014 *  .LXOVL  22054 *  .LXMOD  22116 *
                        .LXIND  22142    .LXDIS  22145    .LXFLG  22146    .LTCH   22147
    .FPTRP  22155       .FFPT.  22155 *  .FWFL.  22331 *  .FPOUT  22342    .FPARG  22350   /.COUNT/ 22352
                        OVFLOW  22417 *  FPTRP   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   24127
                        .LNTP   24215    .AOUT   24264    .DFLT   24302    .FLT    24436    .EXFON  24524
                        .DEXFN  24526    .FXD    24527    .HOUT   24702    .INTG   24752    .LOUT   25100
                        .OOUT   25117    .XCF    25150    ASTERN  25252 *  EVEN    25565    .TEST   25712
                        .KOUNT  25732    .LIST   25735    .DONE   25746    .OUTBF  26012    .BUF    26042
                        .QSTO   26043    .WIDTH  26044    .GAIN   26045    .GAIN1  26046    .FBDBF  26056
                        EVEN    26067    .DDDFL  26103    .DDFLG  26104    .MOD    26111    .FEX    26112
                        .FEXP   26113    .DIG    26114
```

```
           IBLDR     INF,N010001,44903-0,77387,AM IC     IBJOB L10MKC  7094 A    10/01/68          PAGE  5

  FIOB     26134       .FIOB.  26134     .FCNT   26237     .FBLT.  26335     .FBDT.  26355     .FRLB.  26401
                       .FRLR. (26401)    .FWLR.  26445     .FWLR. (26445)    .FDIBF  26505     .FRITE  26577
  FIOS     26605       .FIOS.  26605     .FSEL.  26745     .FILR.  26751     .FRTB.  26760     .FRTD.  26765
                       .FILL.  26770     .FCLS   26772 *   .FOPN   26776 *   REOF    27002 *   .TOUT.  27145
                       .REED   27153 *   .BIN    27154 *   .FCT    27155     .FCKSZ  27157
  FIOH     27241       .FIOH1  27241     .FIOH.  27271     .FFIL.  30054     .CODE.  30075     .FRTN.  30102
  FRRD     30300       .ENCD.  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
  UN05     31512       .UN05.  31512
  UN06     31513       .UN06.  31513     .BUFSZ  31514 *
  UN07     31517       .UN07.  31517
  UN11     31520       .UN11.  31520
  UN12     31521       .UN12.  31521
  UN18     31522       .UN18.  31522
  FIOU     31523       .FIOU.  31523     .CTUIO  32235     .NMLST  32260     .NAME.  33716     .INTAP  33717
  FLOG     34347       ALOG10  34347     ALOG    34350 *
  FXPF     34553       EXP     34553
  FSCN     34674       COS     34674     SIN    -34675
  FSQR     35070       SQRT    35070
  FXP1     35143       .XP1.   35143
  FXP2     35257       .XP2.   35257
  FBST     35375       .FBST.  35375
  FEFT     35612       .FEFT.  35612
  FRWT     35712       .FRWT.  35712
  FSLDI    36017       .FSLI.  36035     .FSDI.  36043 *
  FSLBI    36054       .FBLI.  36072     .FBDI.  36100 *
  FSLI     36112       .SLI.   36112     .SLI1.  36117     .SDI.   36125     .SDI1.  36133
  FSLDO    36146       .FSLO.  36164     .FSDO.  36172 *
  FSLBO    36203       .FBLO.  36221     .FBDO.  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)
  .IOCS    37351       .L(0)   37351     .MONSW  37371     .TEOR   37440     .DEFI.  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 *
                       .FEEIT  40627     .GTIOX  40650     .RW7    40766 *   .RE7    41411 *   .ENDTR  42052
                       .SEL59  42054 *   .BSR.   42473     .EOTOF  42620     .ETOF3  42626 *   .SWITC  42655
                       .TCHEX  43162     .BASIO  43165 *
  .IOCSM   43170

1 OPUNCZ   43170      /DATA  /(03112)    EVEN    43171     OPUNCH  43354
  READSZ   43372      /DATA  /(03112)   /SYMBLS/(12176)   /PRNT  /(15570)   /PERTB /(12004)   /FLOTC /(14624)
                       EVEN    43373     READS   50515
  REREZZ   50542       EVEN    50543     REREAD  51176
  STARIZ   51214      /DATA  /(03112)   /SYMBLS/(12176)   /PRNT  /(15570)   /PERTB /(12004)   /FLOTC /(14624)
                       EVEN    51215     STAR10  53260
```

```
        IBLDR      INF,N010001,44903-0,77387,AM IC      IBJOB L10MKC  7094 A     10/01/68          PAGE  6

  STAR9Z  53303      /DATA  /(03112)  /SYMBLS/(12176)  /PRNT  /(15570)   STAR9   55720
  UNBCD.  55741       UNPBCD  56023
  VCNVR.  56046       EVEN    56047    VCNVRT  56360    MCNVRT  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
  GETJB.  43432       GETID   43432
  SDINI.  43515       SDINIT  43515
  SDNPT   43602       SDNPUT  43604
  SETBN   45227       RESBN   45227
  SINTR.  45230       SINTRP  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)  /COORNG/ 53116    EVEN    53115    /PLOTC /(14624)
                      CPLOT   57432
  CROSSZ  57472      /DATA  /(03112)  /ELMAIR/(46752)   EVEN    57473    CROSS   60252
  ENDPTZ  60345      /DATA  /(03112)  /ELMAIR/(46752)  /COORNG/(53116)   ENDPTS  62137
  E1RGCZ  62321      /ELMAIR/(46752)  /COORNG/(53116)   E1RGCD  62604
  PFILEZ  62622      /DATA  /(03112)  /ELMAIR/(46752)  /COORNG/(53116)  /PLOTC /(14624)   /AZOBJ /(15676)
                      EVEN    62623    PFILE   64043
  PROFLZ  64134      /ELMAIR/(46752)  /COORNG/(53116)  /PLOTC /(14624)  /PRNT  /(15570)   EVEN    64135
                      PROFIL  66521
  RANGEZ  66544      /DATA  /(03112)  /ELMAIR/(46752)  /COORNG/(53116)   EVEN    66545     RANGE   67320
  ZYCUTZ  67364      /DATA  /(03112)  /COORNG/(53116)   EVEN    67365    ZYCUT   67617

3 TWNR4Z  43170      /DATA  /(03112)  /AZOBJ /(15676)  /PRNT  /(15570)  /TMTWIN/ 43172    EVEN    43171
                      TWNRY4  46431
  TWNR5Z  46454      /DATA  /(03112)  /AZOBJ /(15676)  /PRNT  /(15570)  /TMTWIN/(43172)    EVEN    46455
                      TWNRY5  47272

4 LATTZZ  43170      /DATA  /(03112)   EVEN    43171    LATT    43267
  MERIDZ  43315      /DATA  /(03112)  /AZOBJ /(15676)   MERID   43462
  SMPOFZ  43504      /DATA  /(03112)  /AZOBJ /(15676)  /TMPATT/ 43506    EVEN    43505     EVEN    44357
                      SMPOF   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  70020
  MODDLZ  70672      /DATA  /(03112)   EVEN    70673    MODDL   70754
```

```
         IBLDR      INF,N010001,44903-0,77387,AM IC     IBJOB L10MKC  7094 A    10/01/68          PAGE  7

   MWVCTZ  70770    /DATA  /(03112)   /THMDESN/(46316)   EVEN    70771    MWVCTR  71301
   STAR2Z  71322    /DATA  /(03112)   /PRNT  /(15570)   /THMDESN/(46316)  EVEN    71323    STAR2   72003

 6 PARAXZ  45363    /DATA  /(03112)   /AZOBJ /(15676)   PARAX   45540
   STAR3Z  45567    /DATA  /(03112)   /TMPATT/(43506)   /AZOBJ /(15676)  /PRNT  /(15570)   STAR3   51400

 7 GETLCZ  45363    GETLC   45476
   STAR7Z  45535    /DATA  /(03112)   /TMPATT/(43506)   /PRNT  /(15570)  /AZOBJ /(15676)   STAR7   52062

 8 STAR8Z  45363    /DATA  /(03112)   /TMPATT/(43506)   /PRNT  /(15570)  /PERTB /(12004)   /AZOBJ /(15676)
                    STAR8   51104

I/O BUFFERS                   72024 THRU  77772

UNUSED CORE                   77773 THRU  77777

        292 LINES OUTPUT.
$IBSYS
 RETURNING TO IBSYS.
```

PROGRAM LISTING

Section 0

LINK 0

```
                         INF,N010001,44903-0,77387,AM IC      $JOB L10MKF  7094 A   09/19/66

$*
$TITLE          FORTRAN OPTICAL LENS DESIGN PROGRAM
$IBFTC FOLDF    M94,XR7             LINK 0    (MAIN PROGRAM)              MN001000
```

```
C                                                                      WN001010
C     *** DESCRIPTION ***                                              WN001020
C         FOLDP PERFORMS AUTOMATIC DESIGNING OF OPTICAL SYSTEMS BY A LINEWN001030
C         ARIZED LEAST SQUARES PROCEDURE.  THE MAIN PROGRAM IS A CONTROL WN001040
C         ROUTINE FOR PROCESSING A SINGLE CASE, WHICH INVOLVES READING   WN001050
C         INPUT DATA AND EXECUTING THE SET OF USER SPECIFIED OPTIONS.    WN001060
C                                                                      WN001070
C         THE FOLLOWING OPTIONS ARE AVAILABLE ...                        WN001080
C                                                                      WN001090
C                  0      TERMINATE PROCESSING                         WN001100
C                  1      INPUT DATA ON OCTAL CARDS                    WN001110
C                  2      DESIGN COMPUTATIONS                          WN001120
C                  3      SPOT DIAGRAM COMPUTATIONS                    WN001130
C                  4      TWINRAY DIAGNOSTIC FROM MAXIMUM OBJECT HEIGHT WN001140
C                  5      TWINRAY DIAGNOSTIC FROM ZERO OBJECT HEIGHT   WN001150
C                  6      PUNCH OCTAL CARDS                            WN001160
C                  7      PLOT SPOT DIAGRAMS                           WN001170
C                  8      SENSITIVITY COMPUTATIONS                     WN001180
C                  9      PRINT GEOMETRY DATA                          WN001190
C                 10      PRINT ALL INPUT DATA                         WN001200
C                 11      PROFILE PLOT                                 WN001210
C                 12      PUNCH CARDS FOR PAGOS MTF                    WN001215
C                                                                      WN001220
      COMMON / DATA / NCNTRL, CONTRL(10), TITLE(12), DATE(3), PUNCID,  DT001000
     1 DATA(3465)                                                      DT001100
      DIMENSION  WOBJH(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), 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), SPPEA ), ( 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), 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
      EQUIVALENCE  ( DATA(3484), ATRGGR ), ( DATA(3465), GAUSS )       DT103000
      INTEGER  CONTRL, AFLAG, DATE, ATRGGR                             DT201000
      REAL  LATTC                                                      DT202000
C
C          THIS BLOCK OF COMMON CONTAINS PERTURBATION VALUES          PB001000
      COMMON / PERTB /  PERTB(30,4), NPERTB, REFOCS                    PB002000
```

```
      EQUIVALENCE  ( FERTD(1,1), IFERTD(1,1) )                          FD003000
      DIMENSION  IFERTD(30,4)                                           FD004000
      INTEGER  REFOCS                                                   FD005000
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 / HEADER /  JTEMP, JSUBST                                  MN100500
C
      COMMON  / PLOTC /  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( CONTRL(I) .EQ. 0 )  GO TO 32000                               MN102200
C
C            TEST FOR LEGAL PROGRAM CONTROL COMMAND                     MN103000
      IF( CONTRL(I) .GE. 1  .OR.  CONTRL(I) .LE. 12 )  GO TO 1500       MN103200
      CALL  ERROR2( 36H ILLEGAL PROGRAM CONTROL COMMAND    , CONTRL(I) )MN103500    21
      GO TO 20000                                                       MN103600
 1500 JTEMP = CONTRL(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    28
      JSUBST = JSUBST + 1                                               MN105300
      GO TO 20000                                                       MN105400
C
C            OPTICS DIAGNOSTIC CALCULATION                              MN106000
 3000 CALL  STAR3                                                       MN106200    32
      JSUBST = JSUBST + 1                                               MN106300
      CALL  PRMSUB( $3000 )                                             MN106400    35
      GO TO 20000                                                       MN106600
C
C            TWIN-RAY DIAGNOSTIC CALCULATION                            MN107000
 4000 CALL  TWNRY4                                                      MN107200    39
      JSUBST = JSUBST + 1                                               MN107300
      CALL  PRMSUB( $4000 )                                             MN107400    42
      GO TO 20000                                                       MN107600
C
C            TWIN-RAY DIAGNOSTIC CALCULATION                            MN108000
 5000 CALL TWNRY5                                                       MN108400    46
      JSUBST = JSUBST + 1                                               MN108500
      CALL  PRMSUB( $5000 )                                             MN108600    49
      GO TO 20000                                                       MN109000
C
C            PUNCH OCTAL CARDS                                          MN201000
 6000 CALL OPUNCH                                                       MN201200    53
      GO TO 20000                                                       MN201400
```

```
      FORTRAN OPTICAL LENS DESIGN PROGRAM                                        09/19/68                PAGE
         FOLDP        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C
C              SPOT DIAGRAM PLOT                                          MN202000
 7000 CALL   START7                                                       MN202200    56
      JSUBST = JSUBST + 1                                                 MN202300
      CALL   PRMSUB( $7000 )                                              MN202400    59
      GO TO 20000                                                         MN202600
C
C              PERTURBATION CALCULATION                                   MN203000
 8000 CALL   STAR8                                                        MN203200    63
      JSUBST = JSUBST + 1                                                 MN203300
      CALL   PRMSUB( $8000 )                                              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( $11000 )                                             MN206600    81
      GO TO 20000                                                         MN206800
C                                                                         MN206850
C              PUNCH MTF CARDS                                            MN206900
12000 CALL MTFPN                                                          MN206920    85
20000 CONTINUE                                                            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 09/19/68 PAGE 4
FOLDP 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	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	PSCAL	00065	R
OMGAF	0006C	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	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
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	ATRGGR	06666	I
GAUSS	06667	R						

COMMON BLOCK PERTB ORIGIN 06672 LENGTH 00172

PERTB	00000	R	NPERTB	00170	I	REFOCS	00171	I
IPERTB	00000	I						

COMMON BLOCK SYMBLS ORIGIN 07064 LENGTH 02424

SDESN	00000	I	SUBSYM	01130	I	BDYSYM	02114	I

COMMON BLOCK HEADER ORIGIN 11510 LENGTH 00002

JTEMP	00000	I	JSUBST	00001	I

COMMON BLOCK PLOTC 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

...... SECTION 13

STORAGE MAP

SUBROUTINES CALLED

FPTRP	SECTION	14	REREAD	SECTION	15	READS	SECTION
ERROR2	SECTION	17	STAR2	SECTION	18	STAR3	SECTION
PRMSUB	SECTION	20	TWNRY4	SECTION	21	TWNRY5	SECTION
OPUNCH	SECTION	23	STAR7	SECTION	24	STAR8	SECTION
STAR9	SECTION	26	STAR10	SECTION	27	PROFIL	SECTION
MTFPN	SECTION	29	EXIT	SECTION	30	.EXIT.	SECTION
.FMEM.	SECTION	32	SYSLOC	SECTION	33		

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCAT
1000	3A	11541	20000	86A	11762	32000	90A	1176
1500	23A	11603	2000	27A	11627	3000	31A	1163
4000	38A	11651	5000	45A	11664	6000	52A	1167
7000	55A	11703	8000	62A	11716	9000	69A	1173
10000	72A	11735	11000	77A	11744	12000	84A	1175

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12017.

```
$*
$TITLE         ERROR MESSAGE PRINTER
$IBFTC ERRORZ  M94,XR7             LINK 0   (ERROR,ERROR1,....,ERROR4)   ER001000
```

```
        ERROR MESSAGE PRINTER                                          10/31/67                   PAGE  1
           ERRORZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  ERROR( ETEXT )                                       ER001100
      DIMENSION  ETEXT(6)                                              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( 1H0, 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( 1H0, 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( 1H0, 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( 1H0, 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 PRINTER                                                    10/31/67                 PAGE  2
  ERROR2                                    STORAGE  MAP

                                      SUBROUTINE  ERROR
                                            ENTRY POINTS

    ERROR      SECTION     3          ERROR1     SECTION     4          ERROR2     SECTION     5
    ERROR3     SECTION     6          ERROR4     SECTION     7
                                            SUBROUTINES CALLED

    .FWRD.     SECTION     8          .FSLO.     SECTION     9          .UN06.     SECTION    10
    .FFIL.     SECTION    11          .FCNV.     SECTION    12          SYSLOC     SECTION    13
                                      EFN    IFN    CORRESPONDENCE

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.
```

```
                         INF,N010001,44800-0,77387,AN IC     IBJOB V13NJO  7094 A    10/31/67          PAGE  10

$*
$TITLE          ABNORMAL  JOB TERMINATION ROUTINE
$IBFTC GOOF22  M94,XR7              LINK 0    (GOOF)                      GF001000
```

```
   ABNORMAL JOB TERMINATION ROUTINE
     GOOF 22      - EFN   SOURCE STATEMENT  -  IFN(S)  -                10/31/67                     PAGE  1

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

```
ABNORMAL JOB TERMINATION ROUTINE                                      10/31/67                     PAGE 2
   GOOF22                                       STORAGE MAP

                                      SUBROUTINE GOOF
                                             ENTRY POINTS

  GOOF        SECTION     3
                                             SUBROUTINES CALLED

  EXIT        SECTION     4              SYSLOC      SECTION     5
                                         EFN    IFN  CORRESPONDENCE

EFN        IFN        LOCATION       EFN        IFN        LOCATION       EFN        IFN        LOCATION

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00025.
```

```
                    INF,N010001,44800-0,77387,AN IC     $IBJOB V13MJO  7094 A    10/31/67          PAGE  6

$*
$TITLE         SUBROUTINE TO PERFORM PARAMETER SUBSTITUTIONS
$IBFTC PRMSZZ  M94,XR7             LINK 0   (PRMSUB)                    8B001000
```

```
      SUBROUTINE TO PERFORM PARAMETER SUBSTITUTIONS                      10/31/67                       PAGE  1
        PRMSZZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  PRMSUB( * )                                          SB001200
C                                                                      
C     *** INPUTS ***                                                   SB001002
C NSUBT    NUMBER OF SUBSTITUTION SETS                  (DATA COMMON) SB001004
C NSUBP    NUMBER OF SUBSTITUTION PARAMETERS PER SET    (DATA COMMON) SB001006
C SUBST    SUBSTITUTION VECTOR                          (DATA COMMON) SB001008
C                                                                      SB001010
C     *** OUTPUTS ***                                                  SB001012
C          THE OUTPUT IS A MODIFIED DATA VECTOR CONTAINING THE SUBSTITUTABSB001014
C          LE VALUES.                                                  SB001016
C *        ALTERNATE RETURN USED UNTIL THERE ARE NO MORE SETS          SB001018
C                                                                      SB001020
C     *** DESCRIPTION ***                                              SB001022
C          PRMSUB IS DESIGNED TO CYCLE THROUGH THE NS SUBSTITUTION SETS.  SB001024
C          TO THIS END IT MUST BE CALLED NS+1 TIMES IN SUCCESSION.  AT THESB001026
C          JTH CALL SET J IS SUBSTITUTED INTO THE DATA REGION AND THE     SB001028
C          ALTERNATE RETURN IS USED.  ON THE NS+1 ENTRY THE NOMINAL SYSTEMSB001030
C          IS RESTORED FOLLOWED BY A NORMAL RETURN.  IF CALLED AGAIN THE  SB001032
C          SEQUENCE IS RESTARTED.                                      SB001034
C                                                                      SB001036
      COMMON/DATA/DUMMY(27),DATA(3483)                                 SB001400
      EQUIVALENCE (NSUBT,DATA(6)),(NSUBP,DATA(7)),(SUBST(1),ISUBST(1), SB001600
     1DATA(2682))                                                      SB001800
      DATA  JSET / 0 /                                                 SB002000
      INTEGER  SORG                                                    SB002100
      DIMENSION SUBST(500),ISUBST(500)                                 SB002200
C              START OF PRMSUB SUBROUTINE                              SB003000
C
C              IS PARAMETER SUBSTITUTION DESIRED                       SB003200
      IF( NSUBT .LE. 0 )  RETURN                                       SB003300
      IORG = NSUBT * NSUBP                                             SB003400
C
C              FIRST TIME CALLED FOR SUBSTITUTION PROCESS              SB003500
      IF( JSET .GT. 0 )  GO TO 3000                                    SB003600
C
C              INCREMENT SUBSTITUTION SET INDEX AND SET FLAG FOR       SB004000
C              REPLACEMENT OF NOMIAL SET BY CURRENT SUBST. SET         SB004100
 2000 JSET = JSET + 1                                                  SB004200
      NFLAG = 2                                                        SB004300
      GO TO 4000                                                       SB004500
 3000 NFLAG = 1                                                        SB005000
 4000 SORG = ( JSET - 1 ) * NSUBP                                      SB005200
C
C              THIS LOOP REPLACES NOMIAL SET WITH SUB. IF NFLAG = 2 AND  SB005500
C              RESTORES THE NOMIAL SET WHEN NFLAG = 1                  SB005600
 5000 DO  6000  I = 1, NSUBP                                           SB005700
      IX = IORG + I                                                    SB005800
C              COMPUTE INDEX TO POINTER TO POSITION OF PARAMETER IN DATA SB006000
      I1 = IABS( ISUBST( IX ) )                                        SB006100
C              COMPUTE INDEX TO SUBSTITUTION VALUE IN SUBST VECTOR     SB006500
      I2 = SORG + I                                                    SB006600
      STEMP = DATA( I1 )                                               SB006700
      DATA( I1 ) = SUBST( I2 )                                         SB006800
 6000 SUBST( I2 ) = STEMP                                              SB007000
C
```

```
  SUBROUTINE TO PERFORM PARAMETER SUBSTITUTIONS                          10/31/67                    PAGE  2
     PRMSZZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C             RETURN IF SUBSTITUTION SET HAS REPLACED THE NOMIAL SET    SB007200
      IF( NFLAG .EQ. 2 )  RETURN 1                                      SB007300
C
C             HAVE ALL THE SUBSTITUTION SETS BEEN USED                  SB007500
      IF( JSET .LT. NSUBT )  GO TO 2000                                 SB007600
C
C             SUBSTITUTION PROCESS COMPLETE, RESET SET INDEX            SB008000
      JSET = 0                                                          SB008100
      RETURN                                                            SB009000
      END                                                               SB009200
```

SUBROUTINE TO PERFORM PARAMETER SUBSTITUTIONS 10/31/67 PAGE 3
PRMS22 STORAGE MAP

SUBROUTINE PRMSUB
COMMON VARIABLES

COMMON BLOCK DATA ORIGIN 00001 LENGTH 06666

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
IORG	06667	I	IORG	06670	I	JSET	06671	I
NFLAG	06672	I	I	06673	I	IX	06674	I
I1	06675	I	I2	06676	I	STEMP	06677	R

ENTRY POINTS

PRMSUB SECTION 5

SUBROUTINES CALLED

SYSLOC SECTION 6

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
3000	11A	06732	2000	8A	06724	4000	12A	06734
5000	13A	06741	6000	24A	07000			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07053.

```
                         INF,ND10001.44000-0.77367,AM IC     IBJOB V13HJO  7094 A    10/31/67          PAGE  3

$*
$TITLE         PRINT CONTROL ROUTINE
$IBFTC PRTCZZ  M94,XR7            LINK 0   (PRTCTL)                    PT001000
```

```
      PRINT CONTROL ROUTINE                                              10/31/67                PAGE  1
        PRTC22      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  PRTCTL                                                 PT001500
C
C     *** INPUTS ***                                                     PT001002
C TITLE    A 72 CHARACTER BCD TITLE                                      PT001004
C DATE     CURRENT DATE  MONTH,DAY,YEAR                                  PT001006
C LINE     CURRENT LINE NUMBER                                           PT001008
C PAGE     CURRENT PAGE NUMBER                                           PT001010
C JTEMP    OPTION NUMBER                                                 PT001012
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 GENERATPT001022
C         ED.  IT PAGE EJECTS AND PRINTS ONE LINE (AT THE TOP OF THE NEW PT001024
C         PAGE)WHICH INCLUDES THE TITLE, DATE, OPTION NUMBER, SUBSTITUTIOPT001026
C         N SET NUMBER, AND PAGE NUMBER.                                 PT001028
C
      COMMON / DATA / DUMMY1(11), TITLE(12), DATE(3), DATA(3484)         PT002000
      INTEGER  DATE                                                      PT002200
      COMMON / PRNT / LINE, PAGE                                         PT002500
      INTEGER  PAGE                                                      PT002700
      COMMON / HEADER /  JTEMP, JSUBST                                   PT003000
C
C             THIS SUBROUTINE PRINTS TITLE, DATE, AND PAGE NO., AND      PT004000
C             CONTROL THE PRINT INDEXES                                  PT004100
      PAGE = PAGE + 1                                                    PT005000
      WRITE ( 6, 32100 )  TITLE, DATE, JTEMP, JSUBST, PAGE               PT006000     2
32100 FORMAT( 1H1, 12A6, 3X, I2, 1H/, I2, 1H/, I2, 3X, 7HCONTROL, I3,    PT006200
     1 3X, 16HSUBSTITUTION SET, I3, 3X, 4HPAGE, I4 )                     PT006400
      LINE = 1                                                           PT007000
      RETURN                                                             PT009000
      END                                                                PT009200
```

```
        PRINT CONTROL ROUTINE                                                        1../31/67                 PAGE  2
          PRTC22                                  STORAGE  MAP

                                     SUBROUTINE  PRTCTL
                                          COMMON   VARIABLES

                  COMMON BLOCK      DATA          ORIGIN       00001          LENGTH       06666

SYMBOL      LOCATION      TYPE         SYMBOL      LOCATION      TYPE         SYMBOL      LOCATION      TYPE
DUMMY1       00000          R          TITLE        00013          R          DATE         00027          I
DATA         00032          R

                  COMMON BLOCK      PRNT          ORIGIN       06667          LENGTH       00002
LINE         00000          I          PAGE         00001          I

                  COMMON BLOCK      HEADER        ORIGIN       06671          LENGTH       00002
JTEMP        00000          I          JSUBST       00001          I
                                                  ENTRY POINTS

        PRTCTL      SECTION      9
                                                  SUBROUTINES CALLED

        .ZWRD.      SECTION     10          .FSLO.      SECTION     11          .UNDS.      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       06703
 THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 06775.
```

```
$*
$TITLE          RAY TRACE ROUTINE
$IBFTC RAYTRZ  M94,XR7            LINK 0    (RAYTR)                    RY001000
```

```
      RAY TRACE ROUTINE                                                   10/31/67                     PAGE  1
         RAYTRZ      - EFN    SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE RAYTR(XOHAT,YOHAT,HEIGHT,COLOR,XVCT,QVCT,SMISS,*,*,*,  RY001500
     1DFLAG,XMTRX,QMTRX)                                                 RY001600
C                                                                        RY002000
C XOHAT   X COORDINATE OF RAY ON UNIT RADIUS ENTRANCE PUPIL              RY002100
C YOHAT   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.003100
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 RADIANS                            RY004301
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 (XOHAT,YOHAT) 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
     1ISURFC(20,100)                                                     RY006600
      EQUIVALENCE (NSURF,DATA(12)),(RHOO,DATA(26)),(S,DATA(31)),         RY007000
     1(SURFC(1,1),ISURFC(1,1),DATA(182))                                 RY007100
      INTEGER COLOR,SMISS,DFLAG,TYPE,VIGFLG,TLTFLG                       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 PO AND QO ...                                                    RY101400
C                                                                        RY101500
      X=RHOO*XOHAT                                                       RY101600
      Y=RHOO*YOHAT                                                       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      4
      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
```

```
C IF DFLAG.NE.0 SAVE P(I-1) AND Q(I-1) ...                         RY103500
C                                                                  RY103600
      IF(DFLAG.EQ.0)GO TO 60                                       RY103700
      XMTRX(1,I)=X                                                 RY103800
      XMTRX(2,I)=Y                                                 RY103900
      XMTRX(3,I)=Z                                                 RY104000
      QMTRX(1,I)=QX                                                RY104100
      QMTRX(2,I)=QY                                                RY104200
      QMTRX(3,I)=QZ                                                RY104300
60    IF(QZ.EQ.0.)GO TO 190                                        RY104400
C                                                                  RY105000
C GET PARAMETERS FOR ITH SURFACE ...                               RY105100
C                                                                  RY105200
      TYPE=ISURFC(1,I)                                             RY105300
      R=SURFC(11,I)                                                RY105400
      B=SURFC(12,I)                                                RY105500
      SI=SURFC(4,I)                                                RY105600
      IF(TYPE.EQ.4) GO TO 280                                      RY105700
C                                                                  RY106000
C SURFACE IS STANDARD CONIC OR ASPHERIC ...                        RY106100
C GET POINT ON VERTEX PLANE ...                                    RY106200
C                                                                  RY106300
      LAMDA=(SI-Z)/QZ                                              RY106400
90    X=X+LAMDA*QX                                                 RY106500
      Y=Y+LAMDA*QY                                                 RY106600
      Z=0.                                                         RY106700
      IF(R.NE.0.)GO TO 130                                         RY106800
C                                                                  RY107000
C SURFACE IS A PLANE ...                                           RY107100
C                                                                  RY107200
      IF(TYPE.GT.4)GO TO 430                                       RY107300
      NX=0.                                                        RY107400
      NY=0.                                                        RY107500
      NZ=1.                                                        RY107600
      GO TO 550                                                    RY107700
130   GO TO (140,150,160,140,140,150,160),TYPE                     RY107800
C                                                                  RY109000
C SURFACE OF REVOLUTION ...                                        RY109100
C                                                                  RY109200
140   ALFA=B*QZ*QZ-1.                                              RY109300
      BETA=X*QX+Y*QY-R*QZ                                          RY109400
      GAMA=X*X+Y*Y                                                 RY109500
      GO TO 170                                                    RY109600
C                                                                  RY201000
C HORIZONTAL CYLINDER ...                                          RY201100
C                                                                  RY201200
150   ALFA=(B-1.)*QZ*QZ-QY*QY                                      RY201300
      BETA=Y*QY-R*QZ                                               RY201400
      GAMA=Y*Y                                                     RY201500
      GO TO 170                                                    RY201600
C                                                                  RY202000
C VERTICAL CYLINDER ...                                            RY202100
C                                                                  RY202200
160   ALFA=(B-1.)*QZ*QZ-QX*QX                                      RY202300
      BETA=X*QX-R*QZ                                               RY202400
      GAMA=X*X                                                     RY202500
```

```
        RAY TRACE ROUTINE                                         10/31/67                  PAGE  3
           RAYTR2      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

170    DELTA=BETA*BETA+ALFA*GAMA                                  RY202600
       IF(DELTA)190,200,210                                       RY202700
C                                                                 RY203000
C RAY MISSED SURFACE I ...                                        RY203100
C                                                                 RY203200
190    SMISS=I                                                    RY203300
       RETURN 1                                                   RY203400
200    LAMDA=-GAMA/BETA                                           RY203500
       GO TO 220                                                  RY203600
210    LAMDA=-GAMA/(BETA+SIGN(SQRT(DELTA),BETA))                  RY203700
C                                                                 RY204000
C COMPUTE PI ...                                                  RY204100
C                                                                 RY204200    49
220    X=X+LAMDA*QX                                               RY204300
       Y=Y+LAMDA*QY                                               RY204400
       Z=LAMDA*QZ                                                 RY204500
C                                                                 RY204600
C COMPUTE NI IF NOT ASPHERIC ...                                  RY205000
C                                                                 RY205100
       GO TO (240,250,260,240,430,430,430),TYPE                   RY205200
240    NX=X                                                       RY205300
       NY=Y                                                       RY205400
       GO TO 270                                                  RY205500
250    NX=0.                                                      RY205600
       NY=Y                                                       RY205700
       GO TO 270                                                  RY205800
260    NX=X                                                       RY205900
       NY=0.                                                      RY206000
270    NZ=(1.-B)*Z-R                                              RY206100
       GO TO 550                                                  RY206200
C                                                                 RY207000
C SURFACE IS NON-STANDARD CONIC ...                               RY207100
C                                                                 RY207200
430    XT=SURFC(13,I)                                             RY207300
       YT=SURFC(14,I)                                             RY207400
       U=SURFC(15,I)                                              RY207500
       V=SURFC(16,I)                                              RY207600
C                                                                 RY208000
C COMPUTE XO PRIME ...                                            RY208100
C                                                                 RY208200
       X=X-XT                                                     RY208300
       Y=Y-YT                                                     RY208400
       Z=Z-SI                                                     RY208500
       IF(ABS(U)+ABS(V).NE.0.)GO TO 310                           RY208600
C                                                                 RY209000
C SURFACE IS TRANSLATED ONLY ...                                  RY209100
C                                                                 RY209200
       TLTFLG=1                                                   RY209300
       LAMDA=-Z/QZ                                                RY209400
       GO TO 90                                                   RY209500
C                                                                 RY301000
C SURFACE IS ALSO TILTED ...                                      RY301100
C                                                                 RY301200
310    TLTFLG=2                                                   RY301300
       W=SQRT(1.-(U*U+V*V))                                       RY301400
C                                                                 RY302000
```

```
C COMPUTE XI PRIME ...                                                   RY302100
C                                                                        RY302200    71
      ZPP=U*X+V*Y+W*Z                                                    RY302300
      QZPP=U*QX+V*QY+W*QZ                                                RY302400
      LAMDA=-ZPP/QZPP                                                    RY302500
      X=X+LAMDA*QX                                                       RY302600
      Y=Y+LAMDA*QY                                                       RY302700
      Z=Z+LAMDA*QZ                                                       RY302800
      IF(R.NE.0.)GO TO 380                                               RY302900
C                                                                        RY303000
C TILTED PLANE ...                                                       RY303100
C                                                                        RY303200
      NX=U                                                               RY303300
      NY=V                                                               RY303400
      NZ=W                                                               RY303500
      ZPP=U*X+V*Y+W*Z                                                    RY303600
      GO TO 550                                                          RY303700
C                                                                        RY304000
C TILTED CONIC ...                                                       RY304100
C                                                                        RY304200
380   ALFA=B*QZPP-1.                                                     RY304300
      BETA=X*QX+Y*QY+Z*QZ-R*QZPP                                         RY304400
      GAMA=X*X+Y*Y+Z*Z                                                   RY304500
      DELTA=BETA*BETA+ALFA*GAMA                                          RY304600
      IF(DELTA)190,380,390                                               RY304700
380   LAMDA=-GAMA/BETA                                                   RY304800
      GO TO 400                                                          RY304900
390   LAMDA=-GAMA/(BETA+SIGN(SQRT(DELTA),BETA))                          RY305000
C                                                                        RY306000
C COMPUTE XI PRIME ...                                                   RY306100
C                                                                        RY306200    81
400   X=X+LAMDA*QX                                                       RY306300
      Y=Y+LAMDA*QY                                                       RY306400
      Z=Z+LAMDA*QZ                                                       RY306500
C                                                                        RY307000
C COMPUTE NI ...                                                         RY307100
C                                                                        RY307200
      ZPP=U*X+V*Y+W*Z                                                    RY307300
      KAPPA=-R-B*ZPP                                                     RY307400
      NX=X+KAPPA*U                                                       RY307500
      NY=Y+KAPPA*V                                                       RY307600
      NZ=Z+KAPPA*W                                                       RY307700
      GO TO 550                                                          RY307800
C                                                                        RY308000
C SURFACE IS ASPHERIC ...                                                RY308100
C                                                                        RY308200
430   A2=SURFC(13,I)                                                     RY308300
      A3=SURFC(14,I)                                                     RY308400
      A4=SURFC(15,I)                                                     RY308500
      A5=SURFC(16,I)                                                     RY308600
      B2=2.*A2                                                           RY308700
      B3=3.*A3                                                           RY308800
      B4=4.*A4                                                           RY308900
      B5=5.*A5                                                           RY309000
      TQ1=1.-B                                                           RY309100
      TQ2=R*R                                                            RY309200
```

```
   RAY TRACE ROUTINE                                            10/31/67                    PAGE  5
      RAYTRZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      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                                                                        92
450   PHI=X*X+Y*Y                                               RY401700
      GO TO 480                                                 RY401800
C
C HORIZONTAL CYLINDER...                                        RY401900
C
460   PHI=Y*Y                                                   RY402000
      GO TO 480                                                 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.0.) GO TO 531                                     RY402600
      DELTA1=TQ2-TQ1*PHI                                        RY402700
      IF(DELTA1.LT.0.)  GO TO 190                               RY402800
      DELTA1=SIGN(SQRT(DELTA1),R)                               RY402900   105
      ZP=ZP+PHI/(R+DELTA1)                                      RY403000
C
COMPUTE THETAK...                                               RY403100
C
531   LAMDA=ZP-Z                                                RY403200
      IF(LAMDA.EQ.0.)  GO TO 540                                RY403300
      LAMDA=LAMDA/QZ                                            RY403400
C
COMPUTE P(K+1)...                                               RY403500
C
      X=X+LAMDA*QX                                              RY403600
      Y=Y+LAMDA*QY                                              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.0)  GO TO 440                                  RY404200
      GO TO 540                                                 RY404300
C
C MAKE SURE THETA IS DECREASING...                              RY404400
C
539   IF(THOLD.EQ.0.)  GO TO 5311                               RY404500
      IF(LAMDA.GE.THOLD)  GO TO 190                             RY404600
5311  THOLD=LAMDA                                               RY404700
      GO TO 440                                                 RY404800
C
```

```
COMPUTE NORMAL VECTOR...                                          RY404900
C
540   GPHI=(((B5*PHI+B4)*PHI+B3)*PHI+B2)*PHI                      RY405000
      IF(R.EQ.0.)  GO TO 543                                      RY405100
      GPHI=GPHI+(1./(2.*DELTA1))                                  RY405200
543   GO TO (4401,4401,4401,4401,544,545,546), TYPE               RY405300
544   PHIX=2.*X                                                   RY405400
      PHIY=2.*Y                                                   RY405500
      GO TO 547                                                   RY405600
545   PHIX=0.                                                     RY405700
      PHIY=2.*Y                                                   RY405800
      GO TO 547                                                   RY405900
546   PHIX=2.*X                                                   RY406000
      PHIY=0.                                                     RY406100
547   NX=-GPHI*PHIX                                               RY406200
      NY=-GPHI*PHIY                                               RY406300
      NZ=1.                                                       RY406400
C                                                                 RY407500
C CHECK FOR VIGNETTING ...                                        RY407600
C                                                                 RY407700
550   IF(VIGFLG.NE.0)GO TO 710                                    RY407800
      A1I=SURFC(2,I)                                              RY407900
      A2I=SURFC(3,I)                                              RY408000
      IF(ABS(A1I)+ABS(A2I).EQ.0.)GO TO 710                        RY408100
      IF(A2I.NE.0.)GO TO 650                                      RY408200
C                                                                 RY408300
CIRCULAR APERTURE ...                                             RY409000
C                                                                 RY409100
      RHO=X*X+Y*Y                                                 RY409200
      IF(TYPE.EQ.4 .AND. TLTFLG.EQ.2)RHO=RHO+Z*Z-ZPP*ZPP          RY409300
      IF(A1I.LT.0.)GO TO 640                                      RY409400
      IF(RHO.GT.A1I*A1I)VIGFLG=I                                  RY409500
      GO TO 710                                                   RY501000
640   IF(RHO.LT.A1I*A1I)VIGFLG=I                                  RY501100
      GO TO 710                                                   RY501200
C                                                                 RY502000
C RECTANGULAR APERTURE ...                                        RY502100
C                                                                 RY502200
650   RHOX=ABS(X)                                                 RY502300
      RHOY=ABS(Y)                                                 RY502400
      IF(TYPE.NE.4 .OR. (TYPE.EQ.4 .AND. TLTFLG.NE.2))GO TO 680   RY502500
      G1=(1.-W)/(U*U+V*V)                                         RY502600
      A12=-G1*U*V                                                 RY502700
      RHOX=ABS((1.-G1*U*U)*X+A12*Y-U*Z)                           RY502800
      RHOY=ABS(A12*X+(1.-G1*V*V)*Y-V*Z)                           RY502900
680   IF(A1I.LT.0.)GO TO 700                                      RY503000
      IF(RHOX.GT.A1I .OR. RHOY.GT.A2I)VIGFLG=I                    RY503100
      GO TO 710                                                   RY503200
700   IF(RHOX.LT.ABS(A1I) .AND. RHOY.LT.A2I)VIGFLG=I              RY503300
710   IF(TYPE.NE.4)GO TO 730                                      RY503400
      X=X+XT                                                      RY503500
      Y=Y+YT                                                      RY503600
730   DELTA=SQRT(NX*NX+NY*NY+NZ*NZ)                               RY503700    177
      NX=NX/DELTA                                                 RY503800
      NY=NY/DELTA                                                 RY503900
      NZ=NZ/DELTA                                                 RY504000
```

```
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.0.)GO TO 780                                   RY601700
C                                                              RY602000
C REFLECTION ...                                               RY602100
C                                                              RY602200
       LAMDA = -2. * LAMDA                                     RY602350
       QX=QX+LAMDA*NX                                          RY602400
       QY=QY+LAMDA*NY                                          RY602300
       QZ=QZ+LAMDA*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=I                                                 RY603300
       RETURN 2                                                RY603400
810    MU=-NU*LAMDA                                            RY603600
       GO TO 830                                               RY603700
820    MU=(-NU+SQRT(DELTA))*LAMDA                              RY605000   192
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
890    IF(VIGFLG.EQ.0)RETURN                                   RY607000
       SMISS=VIGFLG                                            RY607100
       RETURN 3                                                RY608000
       END                                                     RY609000
```

STORAGE MAP

SUBROUTINE RAYTR

COMMON VARIABLES

COMMON BLOCK ATOBJ 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 08666

DUMMY	00000	R	DATA	00033	R	NSURF	00046	I
RHOO	00084	R	S	00071	R	SURFC	00320	R
ISURFC	00320	I						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
TYPE	08672	I	VIGFLG	08673	I	TLTFLG	08674	I
LAMDA	08675	R	NX	08676	R	NY	08677	R
NZ	08700	R	KAPPA	08701	R	NU	08702	R
MU	08703	R	X	08704	R	Y	08705	R
Z	08706	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
TQ1	08741	R	TQ2	08742	R	JCNT	08743	I
THOLD	08744	R	PHI	08745	R	ZP	08746	R
DELTA1	08747	R	GPHI	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

SQRT	SECTION 8	GOOF	SECTION 9	.FXEM.	SECTION 10
SYSLOC	SECTION 11				

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
850	194A	10755	60	17A	07117	190	44A	07326
280	59A	07432	90	28A	07143	130	36A	07172

RAYTRZ

STORAGE MAP

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	10053
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
690	210A	11020	860	205A	11014			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 11157.

Section 1

LINK 1

```
$*
$TITLE          OCTAL CARD PUNCH ROUTINE
$IBFTC OPUNCZ  M94,XR7              LINK 1    (OPUNCH)                      OP001000
```

```
   OCTAL CARD PUNCH ROUTINE                                                09/19/68                    PAGE
      OPUNCZ      - EFN   SOURCE STATEMENT  - IFN(S)  -

      SUBROUTINE  OPUNCH                                                 OP001500
C
      COMMON  / DATA / DUMMY(11), TITLE(12), DATE(3), PUNCID, DATA(3486)OP001600
      DIMENSION  IDATA(3486)                                             OP001650
      EQUIVALENCE  ( DATA(1), IDATA(1) )                                 OP001700
      INTEGER  DATE                                                      OP001750
C
C             START OF OPUNCH SUBROUTINE                                 OP002000
C
C             THIS SUBROUTINE PUNCHES OUT THE ENTIRE PROGRAM DATA AREA   OP002100
C             ( THE DATA ARRAY ) ON CARDS IN OCTAL FORMAT                OP002200
C
C             PUNCH OUT TITLE AND DATE INFORMATION                       OP003000
      PUNCH  32000,  ( TITLE(I), I = 1, 12 ), PUNCID                     OP003200      1
32000 FORMAT( 12A6, A5 )                                                 OP003400
      PUNCH  32010,  ( DATE(I), I = 1, 3 ), PUNCID                       OP003600      8
32010 FORMAT( 38H FORTRAN IV LENS DESIGN PROGRAM RUN OF, I2, 1H-, I2,    OP003800
     1 1H-, I2, 27X, A5 )                                                OP004000
C
C             PUNCH OUT DATA INFORMATION AND CARD SEQUENCE NUMBER        OP005000
      DO  1000 I = 1, 581                                                OP005200
      ITEMP = 6 * ( I - 1 ) + 1                                          OP005400
      JTEMP = ITEMP + 5                                                  OP005600
      KTEMP = I - 1                                                      OP005800
      IF( KTEMP .EQ. 580 )  GO TO 750                                    OP005900
      DO  500 K = ITEMP, JTEMP                                           OP006000
      IF( IDATA(K) .NE. 0 )  GO TO 750                                   OP006100
  500 CONTINUE                                                           OP006200
      GO TO 1000                                                         OP006300
  750 PUNCH  32020,  ( DATA(J), J = ITEMP, JTEMP ), PUNCID, KTEMP        OP006400     33
32020 FORMAT( 6O12, A5, I3 )                                             OP007000
 1000 CONTINUE                                                           OP007200
C
C             END OF OPUNCH SUBROUTINE                                   OP009000
      RETURN                                                             OP009400
      END                                                                OP009800
```

STORAGE MAP

SUBROUTINE OPUNCH

COMMON VARIABLES

COMMON BLOCK DATA ORIGIN 00001 LENGTH 06671

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DUMMY	00000	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		.FFIL.	SECTION 7		.FCNV.	SECTION (
SYSLOC	SECTION 9						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIC
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.

```
$*
$TITLE          INPUT ROUTINE (CASE DECK CARD PROCESSOR)
$IBFTC READSZ  M94,XR7              LINK 1    (READS)                    RD001000
```

```
      INPUT ROUTINE (CASE DECK CARD PROCESSOR)                                   09/19/68                 PAGE
        READSZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  READS                                                 RD001100
C                                                                       
C     *** DESCRIPTION ***                                               RD001001
C         READS READS AND PROCESSES ALL OF THE INPUT CARDS WHICH COMPRISERD001002
C         THE CURRENT CASE DECK.  EVERY CASE DECK BEGINS WITH CONTROL   RD001003
C         NAMELIST DATA ($INPUT) AND ENDS WITH AND END OF CASE CARD.    RD001004
C         THE CASE DECK IS (IN GENERAL) COMPOSED OF 10 SETS OF CARDS AS RD001005
C         FOLLOWS ..........                                            RD001006
C                                                                       RD001007
C     1. CONTROL INPUT (MANDATORY) ONE OR MORE CARDS                    RD001008
C                                                                       RD001009
C         $INPUT  CLEAR,IFLAG,NCNTRL,CONTRL,FUNCID  $                   RD001010
C                                                                       RD001011
C     2. TITLE AND DATE (ONLY IF CLEAR=1) TWO CARDS                     RD001012
C                                                                       RD001013
C         TITLE      (12A6)                                             RD001014
C         MONTH,DAY,YEAR    (3I5)                                       RD001015
C                                                                       RD001016
C     3. OCTAL CARDS (ONLY IF CONTRL(1)=1) TWO OR MORE CARDS            RD001017
C                                                                       RD001018
C         OCTAL CARDS (SEQUENCED)    (6O12,5X13)                        RD001019
C                                                                       RD001020
C     4. GEOMETRY DATA (ONLY IF IFLAG(1)=1) ONE OR MORE CARDS           RD001021
C                                                                       RD001022
C         $GEOM  LMODE,NRAYS,DELY,NSLCS,LATTC,                          RD001023
C                NCLRS,NIPLN,IMODE,CIMPL,NSPLN,                         RD001024
C                DLPLN,OMGA2,OMGA1,DELD,EPRAD,                          RD001025
C                PSCAL,ODIST,HO,DELH,NOBJH,                             RD001026
C                ROTAN,NSURF,SURFC  $                                   RD001027
C                                                                       RD001028
C     5. DESIGN DATA (ONLY IF IFLAG(2)=1 OR 2) ONE OR MORE CARDS        RD001029
C                                                                       RD001030
C         $DESIGN  NDSGN,NDSGV,NJAIL,FNUMB,FLNGH,                       RD001031
C                  WFLGH,ZETA,HEXPP,DEXPP,WEXPP,                        RD001032
C                  WOBJH,WCLRS,WXDIR,WYDIR,WCLRH,                       RD001033
C                  EIMHT,WIMHT,OMGAF,SPPEA,DUMIN,                       RD001034
C                  SYSMX,ITPRNT,NCOND,DESGN,ATRGGR,GAUSS  $             RD001035
C                                                                       RD001036
C         DESIGN SYMBOL CARDS (ONLY IF IFLAG(2)=1)     (6(A1,2A6))      RD001037
C                                                                       RD001038
C     6. BOUNDARY COND. DATA (ONLY IF IFLAG(6)=1 OR 2) ONE OR MORE CARDS RD001039
C                                                                       RD001040
C         $BNDRY  BOUNDS  $                                             RD001041
C                                                                       RD001042
C         BOUNDARY COND. SYMBOL CARDS (ONLY IF IFLAG(6)=1)    (6(1X,2A6))RD001043
C                                                                       RD001044
C     7. SUBSTITUTION SET DATA (ONLY IF IFLAG(3)=1 OR 2) ONE OR MORE CDS.RD001045
C                                                                       RD001046
C         $SUBSTN  NSUBT,NSUBP,SUBST  $                                 RD001047
C                                                                       RD001048
C         SUBSTITUTION SYMBOL CARDS (ONLY IF IFLAG(3)=1)    (6(1X,2A6)) RD001049
C                                                                       RD001050
C     8. SENSITIVITY DATA (ONLY IF IFLAG(4)=1 OR 2) ONE OR MORE CARDS   RD001051
C                                                                       RD001052
C         $SENST  NPERTC,REFOCS,PERTB  $                                RD001053
```

```
C                                                                            RD001054
C         SENSITIVITY SYMBOL CARDS (ONLY IF IFLAG(4)=1)      (5(1X,2A5))     RD001055
C                                                                            RD001056
C     9. PROFILE PLOT DATA (ONLY IF IFLAG(5)=1) ONE OR MORE CARDS            RD001057
C                                                                            RD001058
C         $CRPLOT  YMAXX,DZMIN,NPTS  $                                       RD001059
C                                                                            RD001060
C     10.END OF CASE CARD (MANDATORY) ONE CARD                               RD001061
C                                                                            RD001062
C         END OF CASE   (8OA1)                                               RD001063
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), NRAYS ), ( DATA(3),         DT101000
     1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),           DT101100
     2 NSUBT ), ( DATA(7), NSUBF ), ( DATA(8), NIFLN ), ( 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), HEXPF ), ( DATA(20), DEXPF ), ( DATA(21),        DT101600
     7 WEXPF ), ( 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), 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), 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
      EQUIVALENCE  ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )             DT103000
      INTEGER  CONTRL, 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                                                  PT002500
      INTEGER PAGE                                                           PT002700
C
C          THIS BLOCK OF COMMON CONTAINS PERTURBATION VALUES                 PB001000
      COMMON / PERTB /  PERTB(30,4), NPERTB, REFOCS                          PB002000
      EQUIVALENCE  ( PERTB(1,1), IPERTB(1,1) )                               PB003000
      DIMENSION  IPERTB(30,4)                                                PB004000
      INTEGER  REFOCS                                                        PB005000
C
      COMMON  / PLOTC /  YMAXX, DZMIN, NPTS                                  PC001000
```

```
        INPUT ROUTINE (CASE DECK CARD PROCESSOR)                                        09/19/68                          PAGE
           READSZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C
      DIMENSION IFLAG(6),ISIGN(6)                                              RD001200
C
      INTEGER  SYMBL1(2,13), SYMBL2(2,25), SYMBL3(2,2), SYMBL4(2,7),           RD002000
     1 SYMBL5(2,3), CLEAR, BUFFER(12), BLANK                                   RD002100
      DATA  NSYMB1 / 13 /, NSYMB2 / 25 /, NSYMB3 / 2 /, NSYMB4 / 7 /,          RD002500
     1 NSYMB5 / 3 /                                                            RD002600
C
      DATA  SYMBL1 / 6HLMODE , 1, 6HNRAYS , 2, 6HNSLCS , 3, 6HNCLRS , 4,RD003000
     1 6HNJAIL , 5, 6HNSUBT , 6, 6HNSUBF , 7, 6HNIFLN , 8, 6HIMODE , 9, RD003100
     2 6HNSFLN , 10, 6HNOBJH , 11, 6HNSURF , 12  , 6HATRGGR, 3484 /     RD003200
C
      DATA  SYMBL2 / 6HDELY  , 14, 6HFNUMB , 15, 6HFLNGH , 16, 6HWFLGH ,RD003500
     1 17, 6HZETA  , 18, 6HHEXFF , 19, 6HDEXFF , 20, 6HWEXFF , 21,        RD003600
     2 6HDLFLN , 22, 6HOMGA2 , 23, 6HOMGA1 , 24, 6HDELD  , 25, 6HEFRAD ,RD003700
     3 26, 6HFSCAL , 27, 6HOMGAF , 28, 6HSFFEA , 29, 6HDUMIN , 30,        RD003800
     4 6HODIST , 31, 6HHO    , 32, 6HDELH  , 33, 6HSYSMX , 34, 6HWXDIR ,RD003900
     5 35, 6HWYDIR , 36, 6HROTAN , 37, 6HGAUSS , 3485 /                   RD004000
C
      DATA  SYMBL3 / 6HNDSGN(, 38, 6HNDSGV(, 42 /                             RD004500
      DATA  SYMBL4 / 6HWOBJH(, 53, 6HWCLRH(, 60, 6HWIMHT(, 67, 6HEIMHT(,RD005000
     1 74, 6HCIMPL(, 81, 6HWCLRS(, 88, 6HSUBST(, 2682 /                    RD005100
      DATA  SYMBL5 / 6HLATTC(, 94,  6HSURFC(, 182, 6HDESGN(, 2182 /        RD005500
C
      NAMELIST / INPUT / IFLAG, NCNTRL, CONTRL, CLEAR, FUNCID                RD006500
C
C IFLAG(5)       NAMELIST DATA PRESENT IF VALUE(S) IS 1.                        RD006502
C NCNTRL         NUMBER OF CONTROLS TO BE EXECUTED.                             RD006504
C CONTRL(10)     CONTROLS (OPTIONS) TO BE EXECUTED.                             RD006506
C CLEAR          CLEAR DATA REGION IF NON-ZERO.                                 RD006508
C FUNCID         OCTAL DECK ID.                                                 RD006510
C
      NAMELIST / GEOM / LMODE, NRAYS, DELY, NSLCS, LATTC, NCLRS,              RD007000
     1 NIFLN, IMODE, CIMPL, NSFLN, DLFLN, OMGA2, OMGA1, DELD, EFRAD,          RD007100
     2 FSCAL, ODIST, HO, DELH, NOBJH, SURFC, NSURF, ROTAN                      RD007200
C
C LMODE          LATTICE MODE (0,1, OR 2) DETERMINES PATTERN.                   RD007202
C NRAYS          NUMBER OF RAYS TO BE GENERATED.                                RD007204
C DELY           Y SPACING INCREMENT.                                           RD007206
C NSLCS          NUMBER OF SLICES (TRIPLETS) ON ENTRANCE PUPIL.                 RD007208
C LATTC          (XI,YI,NI) I=1,2,...,NSLCS  RAY PATTERN MATRIX.                RD007210
C NCLRS          NUMBER OF COLORS.                                              RD007212
C NIFLN          NUMBER OF PLANES (OPTICS ONLY).                                RD007214
C IMODE          PLANE SETTING MODE (0 OR 1).                                   RD007216
C CIMPL(7)       Z DISTANCES FOR CURVATURE OF FIELD.                            RD007218
C NSFLN          LOCATION FOR FIRST PLANE TABULATED.                            RD007220
C DLFLN          INCREMENT FOR PLANE SPACING.                                   RD007222
C OMGA2          APERTURE SCALER FOR F.L. (.00001 TYP.).                        RD007224
C OMGA1          APERTURE SCALER FOR B.F. (.707 TYP.).                          RD007226
C DELD           BACK FOCUS INCREMENT.                                          RD007228
C EFRAD          RADIUS OF ENTRANCE PUPIL.                                      RD007230
C FSCAL          PLOT SCALE FACTOR.                                             RD007232
C ODIST          DISTANCE FROM OBJECT TO FIRST SURFACE.                         RD007234
C HO       INITIAL OBJECT HEIGHT.                                               RD007236
C DELH           OBJECT HEIGHT INCREMENT.                                       RD007238
C NOBJH          NUMBER OF OBJECT POINTS.                                       RD007240
```

```
      INPUT ROUTINE (CASE DECK CARD PROCESSOR)                      09/19/68          PAGE
         READSZ     - EFN   SOURCE STATEMENT  -  IFN(S)  -

C NSURF          NUMBER OF LENS SURFACES.                                      RD007242
C ROTAN          AZIMUTH OF OBJECT (DEGREES).                                  RD007244
C SURFC(20,100)SURFACE MATRIX... COLUMN J DESCRIBES JTH SURFACE AS             RD007246
CC               FOLLOWS ..........                                            RD007248
CC                                                                             RD007250
C                    SURFC(1,J)      SURFACE TYPE                              RD007252
C                    SURFC(2,J)     A1J  APERTURE PARAMETER                    RD007254
C                    SURFC(3,J)     A2Z  APERTURE PARAMETER                    RD007256
C                    SURFC(4,J)     DISTANCE FROM PREVIOUS SURFACE             RD007258
C                    SURFC(5,J)      REFRACTIVE INDEX COLOR 1                  RD007260
C                    SURFC(6,J)      REFRACTIVE INDEX COLOR 2                  RD007262
C                        .                  .                                  RD007264
C                        .                  .                                  RD007266
C                        .                  .                                  RD007268
C                    SURFC(10,J)     REFRACTIVE INDEX COLOR 6                  RD007270
C                    SURFC(11,J)     RADIUS OF CURVATURE                       RD007272
C                    SURFC(12,J)     ECCENTRICITY PARAMETER                    RD007274
C                                                                              RD007276
C                TILTED/TRANSLATED .....                                       RD007278
C                                                                              RD007280
C                    SURFC(13,J)    XT    HOR. AXIS DISPLACEMENT               RD007282
C                    SURFC(14,J)    YT    VRT. AXIS DISPLACEMENT               RD007284
C                    SURFC(15,J)    U     X DIRECTION COSINE                   RD007286
C                    SURFC(16,J)    V     Y DIRECTION COSINE                   RD007288
C                                                                              RD007290
C                ASPHERIC (POLYNOMIAL) ...                                     RD007292
C                                                                              RD007294
C                    SURFC(13,J)     A2    DEFORMATION COEFFICIENT             RD007296
C                    SURFC(14,J)     A3    DEFORMATION COEFFICIENT             RD007298
C                    SURFC(15,J)     A4    DEFORMATION COEFFICIENT             RD007300
C                    SURFC(16,J)     A5    DEFORMATION COEFFICIENT             RD007302
C
      NAMELIST / DESIGN / DESGN, NDSGN, NDSGV, NJAIL, FNUMB, FLNGH,             RD008000
     1 WFLGH, ZETA, HEXPP, DEXPP, WEXPP, WOBJH, WCLRH, WIMHT, EIMHT,            RD008100
     2 OMGAF, SPFEA, DUMIN, WCLRS, WXDIR, WYDIR, NCOND, SYSMX, ITSTRT            RD008200
     3 , ATFOCR, GAUSS                                                          RD008201
C
C NDSGN(1)       NUMBER OF DESIGN VARIABLES.                                    RD008202
C NDSGN(2)       MINIMUM PERMISSIBLE DESIGN VARIABLE NUMBER.                    RD008204
C NDSGN(3)       MAXIMUM PERMISSIBLE DESIGN VARIABLE NUMBER.                    RD008206
C NDSGN(4)       NUMBER OF DESIGN ITERATIONS.                                   RD008208
C NDSGV(1)       NUMBER OF DESIGN VARIABLES PER ITERATION.                      RD008210
C NDSGV(2,...)   INITIAL SET OF DESIGN VARIABLE NUMBERS.                        RD008212
C NJAIL          JAIL SENTENCE.                                                 RD008214
C FNUMB          REQUIRED F/ NUMBER.                                            RD008216
C FLNGH          REQUIRED FOCAL LENGTH.                                         RD008218
C WFLGH          WEIGHT ON FOCAL LENGTH.                                        RD008220
C ZETA           MAXIMUM ALLOWABLE INCREMENT FACTOR.                            RD008222
C HEXPP          OBJECT HEIGHT FOR EXIT PUPIL.                                  RD008224
C DEXPP          EXIT PUPIL DISTANCE FROM LAST SURFACE.                         RD008226
C WEXPP          WEIGHT ON EXIT PUPIL DISTANCE.                                 RD008228
C WOBJH(7)       SET OF (OBJECT HEIGHT DEPENDENT) WEIGHTS ON SPOT SIZE.         RD008230
C WCLRH(7)       SET OF WEIGHTS ON LATERAL CHROMATIC ABERRATION.                RD008232
C WIMHT(7)       SET OF WEIGHTS ON EXPECTED Y BARS.                             RD008234
C EIMHT(7)       SET OF EXPECTED Y BARS.                                        RD008236
C OMGAF          FEATHER APERTURE SCALER (+1. TYP.).                            RD008238
```

```
      INPUT ROUTINE (CASE DECK CARD PROCESSOR)                                09/19/68            PAGE
         READS7      - EFN   SOURCE STATEMENT  -  IFN(S)  -

C SPFEA         MINIMUM PERMISSIBLE FEATHER THICKNESS.                         RD008240
C DUMIN         MINIMUM INCREMENT VALUE.                                       RD008242
C WCLRS(6)      SET OF (COLOR DEPENDENT) WEIGHTS ON SPOT SIZE.                 RD008244
C WXDIR         X DIRECTIONAL WEIGHT.                                          RD008246
C WYDIR         Y DIRECTIONAL WEIGHT.                                          RD008248
C NCOND         NUMBER OF BOUNDARY CONDITIONS.                                 RD008250
C SYSMX         MAXIMUM PERMISSIBLE LENS LENGTH.                               RD008252
C ITPRNT        SUBSIDIARY DESIGN PRINT TRIGGER.                               RD008254
C ATRGGR        OPTION FLAG TO RE-COMPUTE OR NOT RE-COMPUTE D(J).              RD008255
C GAUSS         STANDARD DEVIATION FOR GAUSSIAN WEIGHTING.                     RD048255
C DESGN(10,50) DESIGN MATRIX...  COLUMN J DEFINES JTH COMPOSITE DESIGN         RD008256
C               VARIABLE AS FOLLOWS ..........                                 RD008258
C                                                                              RD008260
C                     DESGN(1,J)     NUMBER OF PARAMETERS                      RD008262
C                     DESGN(2,J)     IJ.IJ  REQUIRED                           RD008264
C                     DESGN(3,J)     DELTA VJ                                  RD008266
C                     DESGN(4,J)     K1J  (DATA INDEX  PARAMETER 1)            RD008268
C                     DESGN(5,J)     K2J  (DATA INDEX  PARAMETER 2)            RD008270
C                         .                      .                             RD008272
C                         .                      .                             RD008274
C                         .                      .                             RD008276
C                     DESGN(9,J)     K3J  (DATA INDEX  PARAMETER 6)            RD008278
C                     DESGN(10,J)    IJ.IJ  COMPUTED                           RD008280
C
      NAMELIST / SUBSTN / SUBST, NSUBT, NSUBP                                  RD008500
C
C NSUBT         NUMBER OF SUBSTITUTION SETS.                                   RD008502
C NSUBP         NUMBER OF PARAMETERS PER SET.                                  RD008504
C SUBST(500)    SUBSTITUTION VECTOR.                                           RD008506
C
      NAMELIST / SENST /  NPERTB, PERTB, REFOCS                                RD008600
C
C NPERTB        NUMBER OF PARAMETERS TO BE PERTURBED.                          RD008602
C REFOCS        REFOCUS FLAG (0,1,2, OR 3).                                    RD008604
C PERTB(30,4)   PERTURBATION MATRIX...  ROW J DEFINESTHE JTH PERTURBATIONRD008606
C               PARAMETER AS FOLLOWS..........                                 RD008608
C                                                                              RD008610
C                     PERTB(J,1)     PER CENT BY WHICH PARAMETER IS CHANG.RD008612
C                     PERTB(J,2)     DATA INDEX                                RD008614
C                     PERTB(J,3)     FIRST 6 CHARACTERS OF SYMBOL              RD008616
C                     PERTB(J,4)     LAST 6 CHARACTERS OF SYMBOL               RD008618
C
      NAMELIST / BNDRY /  BOUNDS                                               RD008700
C
C BOUNDS(3,100)BOUNDARY CONDITION MATRIX...  COLUMN J DESCRIBES THE JTH RD008702
C               BOUNDARY CONDITION AS FOLLOWS..........                        RD008704
C                                                                              RD008706
C                     BOUNDS(1,J)    DATA INDEX                                RD008708
C                     BOUNDS(2,J)    MINIMUM PERMISSIBLE VALUE                 RD008710
C                     BOUNDS(3,J)    MAXIMUM PERMISSIBLE VALUE                 RD008712
C
      NAMELIST  / CRPLOT /  YMAXX, DZMIN, NPTS                                 RD008800
C
C YMAXX         MAXIMUM APERTURE RADIUS                                        RD008802
C DZMIN         MINIMUM AIR SPACE THICKNESS                                    RD008804
C NPTS          NUMBER OF POINTS PER ARC                                       RD008806
```

```
C
      DATA  BLANK / 6H         /, MINUS / 6H-       /                  RD008900
      DIMENSION  BUFFRR(6)                                              RD009100
      EQUIVALENCE ( AFLAG, ITNPRT )                                     RD009400
      INTEGER SYHED(4,2)                                                RD009410
      DATA SYHED(1,1)/19H WITH SYMBOL CARDS /,                          RD009420
     1     SYHED(1,2)/21H WITHOUT SYMBOL 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 = 1,3406                                                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, I2 ) )                                             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( 6O12, 5X, I3 )                                            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  PRTCTL                                                      RD107100    66
      WRITE( 6, 32050 )                                                 RD107200    67
32050 FORMAT( 1H0, 10X, 28H PROGRAM CONTROL INFORMATION   )             RD107300
      IF( CLEAR .NE. 0)   GO TO 2005                                    RD107500
      WRITE ( 6, 32055 )                                                RD107600    70
32055 FORMAT( 1H0, 10H CLEAR = 0,  10X, 24H DATA REGION NOT CLEARED )   RD107700
      GO TO 2010                                                        RD107800
 2005 WRITE ( 6, 32060 )                                                RD107900    72
```

```
32060 FORMAT( 1H0, 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( 1H0, 13H IFLAG(1) = 0, 7X,17H NO GEOMETRY DATA  )           RD108400
      GO TO 2020                                                          RD108500
 2015 WRITE ( 6, 32070 )                                                  RD108600    77
32070 FORMAT( 1H0, 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( 1H0, 13H IFLAG(2) = 0, 7X, 15H NO DESIGN DATA )             RD108800
      GO TO 2030                                                          RD108850
2025  IDX=IFLAG(2)                                                        RD108900
      WRITE (6,32080)IDX,(SYHED(I,IDX),I=1,4)                             RD108910    83
32080 FORMAT(12H0 IFLAG(2) =I2,8X19HDESIGN DATA FOLLOWS4A6)               RD108920
 2030 IF( IFLAG(3) .NE. 0)  GO TO 2035                                    RD109000
      WRITE ( 6, 32085 )                                                  RD109050    92
32085 FORMAT( 1H0, 13H IFLAG(3) = 0, 7X, 21H NO SUBSTITUTION DATA  )      RD109100
      GO TO 2040                                                          RD109150
2035  IDX=IFLAG(3)                                                        RD109200
      WRITE (6,32090)IDX,(SYHED(I,IDX),I=1,4)                             RD109210    95
32090 FORMAT(12H0 IFLAG(3) =I2,8X25HSUBSTITUTION DATA FOLLOWS4A6)         RD109220
 2040 IF( IFLAG(4) .EQ. 0 )  GO TO 2045                                   RD109350
      IDX=IFLAG(4)                                                        RD109370
      WRITE (6,32091)IDX,(SYHED(I,IDX),I=1,4)                             RD109380    105
32091 FORMAT(12H0 IFLAG(4) =I2,8X24HSENSITIVITY DATA FOLLOWS4A6)          RD109390
      GO TO 2050                                                          RD109420
 2045 WRITE ( 6, 32092 )                                                  RD109450    113
32092 FORMAT( 1H0, 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( 1H0, 13H IFLAG(5) = 1, 7X, 18H PLOT DATA FOLLOWS )          RD109484
      GO TO 2200                                                          RD109486
 2055 WRITE ( 6, 32094 )                                                  RD109488    118
32094 FORMAT( 1H0, 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(1H0,13H IFLAG(6) = 0,7X,27H NO BOUNDARY CONDITION DATA)      RD109510
      GO TO 2290                                                          RD109515
2210  IDX=IFLAG(6)                                                        RD109520
      WRITE (6,32210)IDX,(SYHED(I,IDX),I=1,4)                             RD109525    124
32210 FORMAT(12H0 IFLAG(6) =I2,8X31HBOUNDARY CONDITION DATA FOLLOWS4A6) RD109530
2290  WRITE (6,32095)NCNTRL, (CONTRL(I),I=1,NCNTRL)                       RD109535    131
32095 FORMAT( 1H0, 9H NCNTRL =, I2, 9X, 35H NUMBER OF PROGRAM CONTROL CORD109550
     1MMANDS  // 10H  CONTRL =, 10I5 )                                    RD109600
      WRITE( 6, 32096 )  PUNCID                                           RD109650    138
32096 FORMAT( 1H0, 9H PUNCID =, 1X, A6 )                                  RD109700
      IF( IFLAG(1) .EQ. 0 )  GO TO 3000                                   RD109800
C
C             READ GEOMETRY DATA                                          RD201000
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 INCORRECRD201600
     1T VALUE FOR LMODE            , LMODE )                              RD201700    144
      IF( NRAYS .LT. 1  .OR.  NRAYS .GT. 200 )  CALL  ERROR2( 36H NUMBERRD202000
     1 OF RAYS EXCEEDS LIMITS       , NRAYS )                             RD202100    147
```

```
      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 INCORRECRD203400
     1T VALUE FOR IMODE           , IMODE )                            RD203500     156
      IF( NOBJH .LT. 1  .OR.  NOBJH .GT. 7 )  CALL  ERROR2( 36H NUMBER ORD203600
     1F OBJECT HEIGHTS IS ILLEGAL, NOBJH )                             RD203700     159
      IF( NSURF .LT. 1  .OR.  NSURF .GT. 98 )  CALL  ERROR2( 36H NUMBER RD204000
     1OF SURFACES EXCEEDS LIMITS  , NSURF )                            RD204100     162
      IF( NIPLN .LE. 0 )  CALL  ERROR2( 36H NUMBER OF IMAGE PLANES IS INRD204500
     1CORRECT, NIPLN )                                                 RD204600     165
      IF( OMGA1 .EQ. 0.0  .OR.  OMGA2 .EQ. 0.0 )  CALL  ERROR( 36H SCALERD205000
     1 FACTOR IS 0.0               )                                   RD205100     168
      IF( EPRAD .LE. 0.0 )  CALL  ERROR3( 36H ENTRANCE PUPIL RADIUS IS 1RD205400
     1LLEGAL    , EPRAD )                                              RD205500
C
C            CONVERT SURFACE TYPE TO INTEGER                           RD208000     171
 2500 JTEMP = NSURF + 1                                                RD208050
      DO  2525  I = 1, JTEMP                                           RD208100
      IF( ISURFC(1,I) .GT. 10000 )  ISURFC(1,I) = SURFC(1,I)           RD208200
 2525 CONTINUE                                                         RD208300
C            TEST FOR VALID SURFACE DATA                               RD301000
      DO  2650  I = 1, NSURF                                           RD301100
      IF( ISURFC(1,I) .LT. 1 .OR. ISURFC(1,I) .GT. 7 )  GO TO 2700     RD301200
      DO  2550  J = 1, NCLRS                                           RD302000
      IF( SURFC(J+4,I) .EQ. 0.0 )  GO TO 2750                          RD302100
 2550 CONTINUE                                                         RD302200
      IF( SURFC(3,I) .GE. 0.0 )  GO TO 2600                            RD302400
      CALL  ERROR2( 36H SECOND APERTURE VALUE IS NEGATIVE   , I )      RD302500     206
 2600 IF( ISURFC(1,I) .NE. 4 .OR. ( SURFC(15,I) ** 2  + SURFC(16,I) **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 ( 36HINCORRECT 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( 36HNRAYS .GT. TOTAL NO. OF RRD306200
     1AYS IN LATT, KTEMP )                                             RD306300     244
 3000 IF( IFLAG(2) .EQ. 0 )  GO TO 5000                                RD307000
C
C             READ DESIGN INPUT                                        RD401000
C
```

```
      READ ( 5, DESIGN )                                                RD401200
C                                                                                248
      IF( ZETA .EQ. 0.0 )  CALL  ERROR( 36H DESIGN SCALE FACTOR IS 0.0  RD401210
     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(I) .LT. NDSGN(2)  .OR.  NDSGV(I) .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(I) .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 NUMBERRD401450
     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, 32120 )  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(6), RD402840
     3 BUFFER(11), BUFFER(12)                                           RD402860   321
32120 FORMAT( 6( A1, 2A6 ) )                                            RD402950
C              STORE DESIGN SYMBOLS FOR PRINT OUT                       RD403000
      DO  3150  M = 1, KTEMP                                            RD403100
 3150 SDESN(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
```

```
      IF( BUFFER(L) .NE. SYMBL2(1,K) )  GO TO 3300                        RD405200
      IDESGN(J+3,I) = SYMBL2(2,K)                                          RD405300
      GO TO 3800                                                           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,I) = SYMBL3(2,K) + IDEX - 1                               RD406200
      GO TO 3800                                                           RD406300
 3325 CONTINUE                                                             RD406400
      GO TO 3450                                                           RD406500
C
C             PRINT ERROR MESSAGE AND SUBSCRIPT IF THE CORRECT SUBSCRIPTRD406700
C             CAN NOT BE EVALUATED                                         RD406800
 3350 CALL ERROR1( 36H ILLEGAL SUBSCRIPT IN DESIGN DATA    , BUFFER(L+1))RD406900
C             TEST FOR REAL VECTOR                                         RD407000    373
 3450 DO  3500  K = 1, NSYMB4                                              RD407100
      IF( BUFFER(L) .NE. SYMBL4(1,K) )  GO TO 3500                         RD407200
      CALL  VCNVRT( BUFFER(L+1), IDEX, $3350 )                             RD407300    383
      IDESGN(J+3,I) = SYMBL4(2,K) + IDEX - 1                               RD407400
      GO TO 3800                                                           RD407500
 3500 CONTINUE                                                             RD407600
C             TEST FOR REAL MATRIX                                         RD408000
      DO  3600  K = 1, NSYMB5                                              RD408100
      IF( BUFFER(L) .NE. SYMBL5(1,K) )  GO TO 3600                         RD408200
      CALL  MCNVRT( BUFFER(L+1), IDEX, JDEX, $3350 )                       RD408300    399
      LTEMP = 3                                                            RD408400
      IF( K .EQ. 2 )  LTEMP = 20                                           RD408500
      IF( K .EQ. 3 )  LTEMP = 10                                           RD408600
      IDESGN(J+3,I) = SYMBL5(2,K) + LTEMP * ( JDEX - 1 ) + IDEX - 1        RD408700
      GO TO 3800                                                           RD408800
 3600 CONTINUE                                                             RD409000
C
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,I) = - IDESGN(J+3,I)           RD409500
 3802 IF(IFLAG(6).EQ.0)GO TO 5000                                          RD451000
C
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 )  ITEMP = ITEMP + 1                      RD451800
      MTEMP = 0                                                            RD452000
      DO  4800  I = 1, ITEMP                                               RD452200
      READ ( 5, 32130 )  ( BUFFER(J), J = 1, 12 )                          RD452400    437
      JTEMP = 12                                                           RD452600
      IF( I .EQ. ITEMP  .AND.  MOD( NCOND, 6 ) .NE. 0 )  JTEMP = 2 *       RD452800
     1 MOD( NCOND, 6 )                                                     RD453000
      DO  4800  J = 1, JTEMP, 2                                            RD453200
      MTEMP = MTEMP + 1                                                    RD453400
C
```

```
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  MCNVRT( 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. 15 )  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 BOURD457500
     1NDARY  )                                                        RD457600    503
      IBNDS(1,MTEMP) = - ( SYMBL5(2,K) + LTEMP * ( JDEX-1 ) + 4 )     RD457700
      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 ) * NSUBP  .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 = NSUBP / 6                                               RD501300
      KTEMP = NSUBP * NSUBT + 1                                       RD501400
      MTEMP = 1                                                       RD501450
      IF( MOD( NSUBP, 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, 2A6 ) )                                          RD502250
```

```
   INPUT ROUTINE (CASE DECK CARD PROCESSOR)                                        09/19/68              PAGE
      READSZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      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
C             PRINT ERROR MESSAGE AND SUBSCRIPT IF THE CORRECT SUBSCRIPTRD506000
C             CAN NOT BE EVALUATED                                       RD506100
 5400 CALL ERROR1( 36H ILLEGAL SUBSCRIPT IN SUBSTUTN. 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  MCNVRT( 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
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
```

```
C
C           CONVERT SUBSTITUTION VALUES TO INTEGER                      RD509350
      KTEMP = NSUDP + NSUDT + 1                                         RD509400
      DO  6200  I = 1, NSUDP                                            RD509450
      IF( ISUBST( KTEMP ) .GT. 0 )  GO TO 6200                          RD509500
      ITEMP = I                                                         RD509550
      DO  6100  J = 1, NSUDT                                            RD509600
      IF( ISUDST(ITEMP) .GT. 10000 )  ISUBST(ITEMP) = SUBST(ITEMP)      RD509650
 6100 ITEMP = ITEMP + NSUDP                                             RD509700
 6200 KTEMP = KTEMP + 1                                                 RD509750
 8000 IF( IFLAG(4) .EQ. 0 )  GO TO 12000                                RD601000
C
C           READ PERTURBATION DATA                                      RD601500
C
      READ ( 5, SENST)                                                  RD602000     675
      IF( NPERTD .LT. 1  .OR.  NPERTD .GT. 30 )  CALL  ERROR2( 36H NUMBERD602200
     1R OF PERTURBATIONS IS ILLEGAL , NPERTD )                          RD602400
C CHECK FOR SYMBOL CARDS                                                RD602410     678
      IF(IFLAG(4).EQ.2)GO TO 12000                                      RD602420
      ITEMP = NPERTD / 6                                                RD602600
      MTEMP = 1                                                         RD602800
      IF( MOD( NPERTD, 6 ) .NE. 0 )  ITEMP = ITEMP + 1                  RD603000
C
C           READ ITEMP DATA CARDS CONTAINING SYMBOLS TO                 RD603200
C           BE PERTURBED                                                RD603400
      DO  10000  I = 1, ITEMP                                           RD603600
      READ ( 5, 32130 ) ( BUFFER(J), J = 1, 12 )                        RD603800     689
      JTEMP = 12                                                        RD604000
      IF( I .EQ. ITEMP  .AND. MOD( NPERTD, 6 ) .NE. 0 )  JTEMP = 2 *    RD604200
     1 MOD( NPERTD, 6 )                                                 RD604400
      DO  10000  J = 1, JTEMP, 2                                        RD605000
C
C           STORE PERTURBATION SYMBOLS FOR PRINT OUT                    RD605200
      IPERTD(MTEMP,3) = BUFFER(J)                                       RD605400
      IPERTD(MTEMP,4) = BUFFER(J+1)                                     RD605600
C
C           TEST FOR REAL VARIABLE                                      RD606000
      DO  8200  K = 1, NSYMB2                                           RD606200
      IF( BUFFER(J) .NE. SYMBL2(1,K) )  GO TO 8200                      RD606400
      IPERTD(MTEMP,2) = SYMBL2(2,K)                                     RD606600
      GO TO 10000                                                       RD606800
 8200 CONTINUE                                                          RD607000
C
C           TEST FOR REAL VECTOR                                        RD607400
      DO  8400  K = 1, NSYMB4                                           RD607500
      IF( BUFFER(J) .NE. SYMBL4(1,K) )  GO TO 8400                      RD607600
      CALL  VCNVRT( BUFFER(J+1), IDEX, $9000 )                          RD607700     722
      IPERTD(MTEMP,2) = SYMBL4(2,K) + IDEX - 1                          RD607800
      GO TO 10000                                                       RD607900
 8400 CONTINUE                                                          RD608000
C
C           TEST FOR REAL MATRIX                                        RD608400
      DO  8600  K = 1, NSYMB5                                           RD608500
      IF( BUFFER(J) .NE. SYMBL5(1,K)) GO TO 8600                        RD608600
      CALL MCNVRT( BUFFER(J+1), IDEX, JDEX, $9000 )                     RD608700     738
      LTEMP = 3                                                         RD608800
```

```
      IF( K .EQ. 2 )  LTEMP = 20                                           RD608900
      IF( K .EQ. 3 )  LTEMP = 10                                           RD609000
      IPERTB(MTEMP,2) = SYMBLS(2,K) + LTEMP * ( JDEX-1 ) + IDEX - 1        RD609100
      GO TO 10000                                                          RD609200
 8600 CONTINUE                                                             RD609300
C
C             PERTURBATION SYMBOL HAS NOT BEEN IDENTIFIED                  RD609500
      CALL  ERROR1( 36H ILLEGAL SYMBOL IN SENSITIVITY DATA , BUFFER(J) )RD609600
C
C             SUBSCRIPT CANNOT BE EVALUATED CORRECTLY                      RD609700     753
 9000 CALL  ERROR1( 36H ILLEGAL SUBSCRIPT IN SENST. DATA   ,BUFFER(J+1))RD609800     756
10000 MTEMP = MTEMP + 1                                                    RD609900
12000 IF( IFLAG(5) .EQ. 0 )  GO TO 32000                                   RD701000
C
C             READ PLOTTING DATA                                           RD701500
C
      READ ( 5, CRPLOT )                                                   RD702000
C
C             TEST PLOT INPUT FOR CORRECT VALUES                           RD702200     764
      IF( YMAXX .LE. 0. )  CALL  ERROR3( 36H TUBE DIAMETER FOR PLOT IS IRD702400
     1NCORRECT, YMAXX )                                                    RD702600     767
      IF( DZMIN .LE. 0. )  CALL  ERROR3( 36H MINIMUM AIR SPACE SEPARATIORD703000
     1N WRONG , DZMIN )                                                    RD703200     770
      IF( NPTS .LT. 2  .OR.  NPTS .GT. 149 )  CALL  ERROR2( 36H NUMBER ORD703400
     1F PLOT POINTS EXCEEDS LIMIT, NPTS )                                  RD703600
C                                                                                       773
32000 RETURN                                                               RD902000
      END                                                                  RD905000
```

SUBROUTINE READS
COMMON VARIABLES

	COMMON BLOCK		DATA	ORIGIN	00001		LENGTH	06[illegible]
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
NSLCF	00041	I	NIFLN	00042	I	IMODE	00043	I
NS LN	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
HEXPF	00055	R	DEXPF	00056	R	WEXPF	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
CDIST	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
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	ATRGGR	06666	I
GAUSS	06667	R	ITNPRT	00047	I			

	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
IPERTB	00000	I						

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

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYP
IFLAG	11515	I	ISIGN	11523	I	SYMBL1	11531	I
SYMBL2	11563	I	SYMBL3	11645	I	SYMBL4	11651	I
SYMBL5	11667	I	BUFFER	11675	I	BUFFRR	11711	R
SYNED	11717	I						

UNDIMENSIONED PROGRAM VARIABLES

READS2

STORAGE MAP

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
CLEAR	11727	I	BLANK	11730	I	I	11731	I
J	11732	I	K	11733	I	KTEMP	11734	I
ITEMP	11735	I	JTEMP	11736	I	IDX	11737	I
L	11740	I	NSYMB1	11741	I	NSYMB2	11742	I
NSYMB3	11743	I	IDEX	11744	I	NSYMB4	11745	I
NSYMB5	11746	I	JDEX	11747	I	LTEMP	11750	I
MINUS	11751	I	MTEMP	11752	I			

ENTRY POINTS

READS	SECTION	13

SUBROUTINES CALLED

.FRDU.	SECTION	14	.FRDD.	SECTION	15	PRTCTL	SECTION	1
.FWRD.	SECTION	17	ERROR2	SECTION	18	ERROR	SECTION	1
ERROR3	SECTION	20	VCNVRT	SECTION	21	ERROR1	SECTION	2
HCNVRT	SECTION	23	ERROR4	SECTION	24	.UN05.	SECTION	2
.FRTN.	SECTION	26	.FCNV.	SECTION	27	.UN06.	SECTION	2
.FFIL.	SECTION	29	CC.1	SECTION	30	CC.2	SECTION	3
CC.3	SECTION	32	CC.4	SECTION	33	SYSLOC	SECTION	3

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIO
1500	40A	13171	1200	10A	13115	1400	19A	13130
32100	FORMAT	12403	32110	FORMAT	12404	2000	65A	13246
1600	43A	13175	32115	FORMAT	12407	1800	57A	13232
32050	FORMAT	12412	2005	72A	13272	32055	FORMAT	12422
2010	73A	13300	32060	FORMAT	12434	2015	77A	13313
32065	FORMAT	12446	2020	78A	13321	32070	FORMAT	12457
2025	82A	13334	32075	FORMAT	12471	2030	90A	13360
32080	FORMAT	12502	2035	94A	13373	32085	FORMAT	12512
2040	102A	13417	32090	FORMAT	12524	2045	113A	13450
32091	FORMAT	12535	2050	114A	13456	32092	FORMAT	12546
2055	118A	13471	32093	FORMAT	12557	2200	119A	13477
32094	FORMAT	12570	2210	123A	13512	32200	FORMAT	12600
2290	131A	13536	32210	FORMAT	12612	32095	FORMAT	12624
32096	FORMAT	12644	3000	246A	14202	2500	173A	13766
2525	183A	14007	2650	216A	14107	2700	220A	14113
2550	200A	14051	2750	222A	14120	2600	207A	14065
2800	224A	14125	2850	240A	14166	5000	523A	15467
3040	287A	14345	3020	282A	14340	3090	289A	14351
3125	310A	14431	3100	308A	14427	3802	424A	15106
3800	415A	15074	32120	FORMAT	12651	3150	324A	14561
3200	341A	14634	3300	353A	14663	3325	369A	14724
3350	372A	14730	3450	375A	14740	3500	390A	14776
3600	411A	15061	4800	519A	15463	32130	FORMAT	12654
4000	462A	15262	4200	478A	15322	4500	516A	15453
4400	512A	15440	4300	508A	15426	8000	673A	16230
6000	645A	16136	5200	565A	15672	5300	577A	15716
5350	593A	15754	5400	596A	15760	5550	599A	15770

INPUT ROUTINE (CASE DECK CARD PROCESSOR) 09/19/68 PAGE

READS2 STORAGE MAP

5600	615A	16023	5700	641A	16123	6200	609A	1622
6100	666A	16216	12000	762A	15563	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.

```
$*
$TITLE          CARD IMAGE LIST ROUTINE
$IBFTC REREZZ  M94,XR7             LINK 1    (REREAD)                    RR001000
```

```
      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-POSITIORR001512
C        NS THE INPUT FILE BY BACKSPACING OVER THE CARD RECORDS WHICH    RR001514
C        WERE READ.                                                      RR001516
C
C
      DIMENSION  BUFFER(80), TEST(11), CHAR(8), OEND(3)                  RR002000
      INTEGER  OFLAG                                                     RR002200
C
      DATA  TEST    / 6HE      , 6HN      , 6HD      , 6H       , 6HD     , RR002400
     1 6HF     , 6H      , 6HC      , 6HA      , 6HS      , 6HE       /  RR002500
C
      DATA  CHAR    / 6H0      , 6H1      , 6H2      , 6H3      , 6H4     , RR002600
     1 6H5     , 6H6      , 6H7       /                                  RR002800
C
      DATA  OEND    / 6H5      , 6H8      , 6H0       /                  RR003000
C
C              START OF SUBROUTINE REREAD                                RR003500
C
      WRITE ( 6, 32000 )                                                 RR004000    1
32000 FORMAT( 1H1, 5X, 34HLENS DESIGN PROGRAM INPUT LISTING   )          RR004500
      IBACKS = 1                                                         RR005000
      OFLAG = 0                                                          RR005200
 1000 READ ( 5, 32020 )  ( BUFFER(J), J = 1, 80 )                        RR005500    3
32020 FORMAT( 80A1 )                                                     RR006000
      DO  1500  I = 1, 11                                                RR007000
      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
      IF( IBACKS .LE. 1 )  GO TO 1000                                    RR008500
      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
      DO  2200  L = 1, 72                                                RR101200
      DO  2100  M = 1, 8                                                 RR101400
      IF( BUFFER(L) .EQ. CHAR(M) )  GO TO 2200                           RR101600
 2100 CONTINUE                                                           RR101800
      GO TO 2800                                                         RR102000
 2200 CONTINUE                                                           RR102200
C
C              OCTAL DATA PRESENT                                        RR103000
      OFLAG = 1                                                          RR103200
      IBACKS = IBACKS + 1                                                RR103250
      ICNT = 1                                                           RR103300
 2300 DO  2350  K = 1, 3                                                 RR103350
      IF( BUFFER( K + 77 ) .NE. OEND(K) )  GO TO 2400                    RR103400
 2350 CONTINUE                                                           RR103450
```

```
      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    62
32060 FORMAT( 1H0, 5X, 20( **** OCTAL DATA INPUT ****, I4, 19H CARDS IN DRR104600
     1ECK **** )                                                        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 09/19/68 PAGE 3
REREZZ

STORAGE MAP

SUBROUTINE REREAD

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
BUFFER	00001	R	TEST	00121	R	CHAR	00134	R
OEND	00144	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
OFLAG	00147	I	IBACKS	00150	I	I	00151	I
L	00152	I	M	00153	I	ICNT	00154	I
K	00155	I						

ENTRY POINTS

REREAD SECTION 3

SUBROUTINES CALLED

.FWRD.	SECTION	4	.FRDD.	SECTION	5	.FBST.	SECTION	6
.UN06.	SECTION	7	.FFIL.	SECTION	8	.FCNV.	SECTION	9
.UN05.	SECTION	10	.FRTN.	SECTION	11	SYSLOC	SECTION	1

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIO
32000	FORMAT	00165	1000	3A	00235	32020	FORMAT	00176
1500	18A	00252	2000	25A	00262	3000	78A	00403
2300	70A	00364	2200	41A	00306	2100	38A	00303
2350	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.

```
$*
$TITLE          GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE
$IBFTC STAR92  M94,XR7              LINK 1    (STAR9)                  89001000
```

```
      SUBROUTINE STAR9                                                  S9001200
C
C     *** DESCRIPTION ***                                               S9001202
C         STAR9 LISTS ALL OF THE GEOMETRY INPUT DATA WITH DESCRIPTIVE   S9001204
C         TITLES.  IF NSUBT.NE.0 THE SUBSTITUTION INPUT DATA IS ALSO    S9001206
C         LISTED IN A TABULAR FORM BY PARAMETER AND BY SET.             S9001208
C
      COMMON / DATA / NCNTRL, CONTRL(160), 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), 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), 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), 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), 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), 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             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                                        PT002500
      INTEGER  PAGE                                                     PT002700
C
      DATA  SURFA / 6HPLANE , 6HSPHERE, 6HPARABO, 6HLOID  , 6HELLIPS,   S9002000
     1 6HOID   , 6HHYPERB, 6HOLOID , 6HHORIZO, 6HNTAL  , 6HVERTIC,      S9002100
     2 6HAL    , 6HCIRCUL, 6HAR CYL, 6HINDER , 6HPARABO, 6HLIC CY,      S9002200
     3 6HLINDER, 6HELLIPT, 6HIC CYL, 6HINDER , 6HHYPERB, 6HOLIC C,      S9002300
     4 6HYLINDE, 6HR     , 6HTRANSL, 6HATED/T, 6HILTED , 6HASPHER,      S9002400
     5 6HIC OF , 6HREVOLU, 6HTION  , 6HHORIZO, 6HNTAL A, 6HSPHERI,      S9002500
     6 6HC CYLI, 6HNDER  , 6HVERTIC, 6HAL ASP, 6HHERIC , 6HCYLIND,      S9002600
     7 6HER    , 6H(DEFOR, 6HMED PL, 6HANE)  , 6H(DEFOR, 6HMED SP,      S9002700
     8 6HHERE) , 6H(DEFOR, 6HMED PA, 6HRABOLO, 6HID)   , 6H(DEFOR,      S9002800
     9 6HMED EL, 6HLIPSOI, 6HD)    , 6H(DEFOR, 6HMED HY, 6HPERBOL,      S9002900
     A 6HOID)  , 6H(DEFOR, 6HMED CI, 6HRCLE) , 6H(DEFOR, 6HMED PA,      S9003000
```

```
     B 6HHABOLA, 6H)      , 6H(DEFOR, 6HMED EL, 6HLIPSE), 6H(DEFOR,     S9003100
     C 6HMED HY, 6HPERBOL, 6HA)     /                                   S9003200
      DIMENSION  SURFA(74), PLINE(9), FMT(9)                            S9004000
      DATA  BLANK / 6H       /                                          S9005000
      DATA  FMT(1) / 6H(1H ,I /, FMT(2) / 6H10.3X, /, FMT(9) / 6H I2  )/S9005500
      DATA  RFMT / 6HE17.7, /, IFMT / 6HI7,10X /, SFMT / 6H I2  ) /     S9008000
      REAL IFMT                                                         S9008100
C
C            START OF STAR9 SUBROUTINE                                  S9101000
C
      CALL  PRTCTL                                                      S9101200    2
      WRITE ( 6, 32000 )  HO, DELH, NOBJH, ROTAN, ODIST, NCLRS, IMODE,  S9101500
     1         DELD, ( CIMPL(I), I=1, NOBJH )                           S9101600    3
32000 FORMAT( 1H0, 50HINITIAL OBJECT HEIGHT......................... HOS9102000
     1, 6X, E14.7 / 53H OBJECT HEIGHT INCREMENT........................ S9102100
     2DELH, 4X, E14.7 / 54H NUMBER OF OBJECT POINTS.......................S9102200
     3... NOBJH, I5 / 54H AZIMUTH OF OBJECT...............................S9102300
     4. ROTAN, E17.7 / 54H DISTANCE OBJECT TO SURFACE 1...................S9102400
     5.. ODIST, E17.7 // 54H NUMBER OF COLORS.............................S9102500
     6.... NCLRS, I5 // 54H MODE NUMBER FOR PRINCIPAL PLANE...............S9102600
     7... IMODE,I5/ 53H DISTANCE FOR PRINCIPAL PLANE.................... S9102700
     9DELD, E18.7 / 54H INCREMENTS TO SPECIFY CURVED IMAGE............. S9102900
     ACIMPL, 1X, 4E16.7 / 55X, 3E16.7 )                                 S9103000
C
      WRITE ( 6, 32010 )  NIPLN, NSPLN, DLPLN, LMODE, NRAYS, DELY,      S9103500
     1 NSLCS                                                            S9103600    10
32010 FORMAT( 1H , 53HNUMBER OF IMAGE PLANES......................... NIS9104000
     1PLN, I5 / 54H POSITION OF FIRST IMAGE PLANE.................. NSPLS9104100
     2N, I5 / 54H SPACING OF IMAGE PLANES....................... DLPLN,S9104200
     3 E17.7 // 54H LATTICE MODE................................... LMODS9104300
     4E, I5 / 54H NUMBER OF RAYS................................. NRAYS,S9104400
     5 I5 / 53H VERTICAL INCREMENT............................. DELY,  S9104500
     6 E18.7 / 54H NUMBER OF SLICES................................ NSLCSS9104600
     7, I5 / 54H RAY PATTERN....................................... LATTC  S9104700
     8 // 9X, 5HSLICE, 9X, 1HX, 15X, 1HY, 8X, 11HNO. OF RAYS )          S9104800
C
      DO  1000  I = 1, NSLCS                                            S9106000
      WRITE ( 6, 32020 )  I, ( LATTC(J,I), J = 1, 3 )                   S9106100    13
32020 FORMAT( 1H , I12, E17.7, E16.7, I8 )                              S9106200
 1000 CONTINUE                                                          S9106300
C
      WRITE ( 6, 32030 )  NSUBT, NSUBP, OMGA2, OMGA1, EPRAD, PSCAL,NSURFS9106500    20
32030 FORMAT( 1H0, 53HNUMBER OF SUBSTITUTION SETS.................... NSS9107000
     1UBT, I6 / 54H NUMBER OF SUBSTITUTION PARAMETERS.............. NSUBS9107100
     2P, I6/ 54H PARAXIAL SCALE FACTOR (FOCAL LENGTH)........... OMGA2, S9107200
     3 E17.7 / 54H MERIDIONAL SCALE FACTOR (BACK FOCUS)........... OMGA1S9107300
     4, E17.7 / 54H RADIUS OF ENTRANCE PUPIL...................... EPRAS9107400
     5D, E17.7 / 54H SPOT DIAGRAM SCALE FACTOR...................... PSCS9107500
     6AL, E17.7 // 54H NUMBER OF SURFACES............................ NS9107600
     7SURF, I5 )                                                        S9107700
C
C             SET LINE COUNT TO EJECT PAGE, PRINT TITLE AND DATE        S9108000
      LINE = 60                                                         S9108100
C
C             PRINT OUT SURFACE INFORMATION                             S9201000
C
```

```
       GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE                        11/08/67                     PAGE  3
            STAR92      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      DO  11000  I = 1, NSURF                                              89201200
      IF( LINE .GT. 55 )  CALL  PRTCTL                                     89201400    26
      LINE = LINE + 11                                                     89201500
C
C               BLANK OUT FIRST LINE OF SURFACE PRINT                      89201600
      DO  1800  J = 1, 9                                                   89201700
 1800 PLINE(J) = BLANK                                                     89201800
      IT = ISURFC(1,I)                                                     89202000
      R  = SURFC(11,I)                                                     89202100
      B  = SURFC(12,I)                                                     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 9000                                       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 9000                                         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 9000                                       89208000
      JTEMP = 18                                                           89208100
 4600 DO  4700  J = 1, 3                                                   89208200
```

```
      KTEMP = JTEMP + J                                              89208300
 4700 PLINE(J+2) = SURFA(KTEMP)                                      89208400
      GO TO 10000                                                    89208500
C
C                   NONSTANDARD CONIC                                89301000
 5000 DO 5100  J = 1, 3                                              89301200
 5100 PLINE(J) = SURFA(J+28)                                         89301300
      IF( R .NE. 0.0 )  GO TO 5200                                   89301500
      PLINE(4) = SURFA(1)                                            89301600
      GO TO 10000                                                    89301700
 5200 IF( B .NE. 0.0 )  GO TO 5300                                   89301800
      PLINE(4) = SURFA(2)                                            89301900
      GO TO 10000                                                    89302000
 5300 IF( B .NE. 1.0 )  GO TO 5400                                   89302100
      PLINE(4) = SURFA(3)                                            89302500
      PLINE(5) = SURFA(4)                                            89302600
      GO TO 10000                                                    89302700
 5400 IF( B .LT. 1.0 )  GO TO 5500                                   89302800
      PLINE(4) = SURFA(7)                                            89302900
      PLINE(5) = SURFA(8)                                            89303000
      GO TO 10000                                                    89303100
 5500 IF( B .LE. - 1.0 )  GO TO 9000                                 89303500
      PLINE(4) = SURFA(5)                                            89303600
      PLINE(5) = SURFA(6)                                            89303700
      GO TO 10000                                                    89303800
C
C                   ASPHERIC OF REVOLUTION                           89304000
 6000 DO 6100  J = 1, 4                                              89304100
 6100 PLINE(J) = SURFA(J+28)                                         89304200
      ITEMP = 4                                                      89304300
      IF( R .NE. 0.0 )  GO TO 6200                                   89304400
      JTEMP = 42                                                     89304500
      ITEMP = 3                                                      89304600
      GO TO 6600                                                     89304700
 6200 IF( B .NE. 0.0 )  GO TO 6300                                   89304800
      JTEMP = 45                                                     89304900
      ITEMP = 3                                                      89305000
      GO TO 6600                                                     89305200
 6300 IF( B .NE. 1.0 )  GO TO 6400                                   89305300
      JTEMP = 48                                                     89305400
      GO TO 6600                                                     89305500
 6400 IF( B .LT. 1.0 )  GO TO 6500                                   89305600
      JTEMP = 56                                                     89305700
      GO TO 6600                                                     89305800
 6500 IF( B .LE. - 1.0 )  GO TO 9000                                 89305900
      JTEMP = 52                                                     89306000
 6600 DO 6700  J = 1, ITEMP                                          89306100
      KTEMP = JTEMP + J                                              89306400
 6700 PLINE(J+4) = SURFA(KTEMP)                                      89306500
      GO TO 10000                                                    89306600
C
C                   HORIZONTAL ASPHERIC CYLINDER                     89307000
 7000 DO 7100  J = 1, 5                                              89307100
 7100 PLINE(J) = SURFA(J+32)                                         89307200
      GO TO 8200                                                     89307300
C
```

```
           STAR02      - EFN   SOURCE STATEMENT  -  IFN(S)  -

C               VERTICAL ASPHERIC CYLINDER                          89307500
 8000 DO  8100  J = 1, 5                                            89307600
 8100 PLINE(J) = SURFA(J+37)                                        89307700
 8200 ITEMP = 3                                                     89307800
      IF( B .NE. 0.0 )  GO TO 8300                                  89308000
      JTEMP = 60                                                    89308100
      GO TO 8600                                                    89308200
 8300 IF( B .NE. 1.0 )  GO TO 8400                                  89308300
      JTEMP = 63                                                    89308400
      ITEMP = 4                                                     89308500
      GO TO 8600                                                    89308600
 8400 IF( B .LT. 1.0 )  GO TO 8500                                  89308700
      ITEMP = 4                                                     89308800
      JTEMP = 70                                                    89308900
      GO TO 8600                                                    89309000
 8500 IF( B .LE. - 1.0 )  GO TO 9000                                89309100
      JTEMP = 67                                                    89309200
 8600 DO  8700  J = 1, ITEMP                                        89309400
      KTEMP = JTEMP + J                                             89309500
 8700 PLINE(J+5) = SURFA(KTEMP)                                     89309600
      GO TO 10000                                                   89309700
C               SURFACE ECCENTRICITY VALUE IS INCORRECT             89401000
 9000 CALL  ERROR2( 36H ECCENTRICITY VALUE WRONG - SURFACE , I )    89401500
C
C               PRINT SURFACE TYPE AND NUMBER                       89402000    201
10000 WRITE  6, 32050 )  I, ( PLINE(J), J = 1, 9 )                  89402100    202
32050 FORMAT( 1H0, 7HSURFACE, I3, 2X, 9A8 )                         89402200
      WRITE ( 6, 32060 )  ISURFC(1,I), ( SURFC(J,I), J = 2, 10 )    89402500    207
32060 FORMAT( 1H0, 12X, 4HTYPE, 12X, 19HAPERTURE PARAMETERS, 9X,    89402600
    1 8HDISTANCE / I16, 7X, 3E15.7         / /  43X, 21HINDICES OF REFR89402700
    2ACTION  / 8X, 6E15.7 )                                         89402800
      IF( ISURFC(1,I) .GT. 3 )  GO TO 10200                         89403000
      WRITE ( 6, 32070 )  SURFC(11,I), SURFC(12,I)                  89403200    216
32070 FORMAT( 1H0, 15X, 1HR, 14X, 1HB / 8X, 2E15.7 )                89403300
      GO TO 11000                                                   89403500
10200 IF( ISURFC(1,I) .NE. 4 )  GO TO 10400                         89403600
      WRITE ( 6, 32080 )  ( SURFC(J,I), J = 11, 16 )                89404000    223
32080 FORMAT( 1H0, 16X, 1HR, 14X, 1HB, 13X, 2HXT, 13X, 2HYT, 14X, 1HU,  89404100
    1 14X, 1HV / 8X, 6E15.7 )                                       89404200
      GO TO 11000                                                   89404300
10400 WRITE ( 6, 32090 )  ( SURFC(J,I), J = 11, 16 )                89405000    229
32090 FORMAT( 1H0, 16X, 1HR, 14X, 1HB, 13X, 2HA2, 13X, 2HA3, 13X, 2HA4, 89405100
    1 13X, 2HA5 / 8X, 6E15.7 )                                      89405200
C
C               PRINT OUT OF SURFACE HAS BEEN COMPLETED             89405500
11000 CONTINUE                                                      89405600
      IF( LINE .GT. 55 )  CALL  PRTCTL                              89406000
C
C               PRINT OUT IMAGE SURFACE                             89406100    238
      ITEMP = NSURF + 1                                             89406200
      WRITE ( 6, 32100 )                                            89406300    240
32100 FORMAT( 1H0, 13HIMAGE SURFACE )                               89406400
      WRITE ( 6, 32110 )  ISURFC(1,ITEMP), ( SURFC(J,ITEMP), J   ), 10 )89407000    241
32110 FORMAT( 1H0, 12X, 4HTYPE, 41X, 21HINDICES OF REFRACTION / ..6,   89407100
    1 7X, 6E15.7 )                                                  89407200
      IF( AFLAG .NE. 1 )  GO TO 12000                               89407500
```

```
      WRITE ( 6, 32120 )  SURFC(11,ITEMP), SURFC(12,ITEMP)
32120 FORMAT( 1H0, 10X, 1HR, 14X, 1HB / 6X, 2E15.7 )
12000 IF( NSUBT .EQ. 0 )  GO TO 20000
      CALL  PRTCTL
C
C              PRINT OUT SUBSTITUTION INFORMATION
C
      WRITE( 6, 32150 )
32150 FORMAT( 1H0, 10X, 24HSUBSTITUTION INFORMATION  )
      LINE = LINE + 2
C
C              SET POINTERS  -  I REFERENCES ISUBST( ) AND JTEMP
C              REFERENCES SUBST( ) AND SUBSYM( )
      I = NSUBP + NSUBT + 1
      JTEMP = 1
C
C              LOOP TO CONSTRUCT VARIABLE FORMAT
12500 DO  13500  K = 1, 6
C              ITEMP SETS THE NUMBER OF SUBSTITUTION VALUES
C              PRINTED PER LINE
      ITEMP = K
      IF( JTEMP .LE. NSUBP )  GO TO 13000
      ITEMP = ITEMP - 1
C
C              LAST FORMAT ( CONSISTING OF ITEMP VALUES ) HAS BEEN
C              CONSTRUCTED
      FMT(ITEMP+3) = BFMT
      GO TO 14000
13000 FMT(ITEMP+2) = RFMT
      IF( ISUBST( I ) .LT. 0 )  FMT(ITEMP+2) = IFMT
      JTEMP = JTEMP + 1
13500 I = I + 1
14000 KTEMP = JTEMP - ITEMP
      LTEMP = JTEMP - 1
      M EMP = KTEMP
      NTEMP = LTEMP
      J = 0
C
C              PRINT OUT SUBSTITUTION SYMBOL HEADING
14500 WRITE ( 6, 32160 ) ( ( SUBSYM(L,K), L=1,2 ), K=KTEMP,LTEMP )
32160 FORMAT( 1H0, 7X, 3HSET, 8X, 6( 2A6, 5X ) )
      LINE = LINE + 2
C
C              PRINT OUT SUBSTITUTION VALUES FOR ALL SETS FOR THE FIRST
C              GROUP OF PARAMETERS
15000 J = J + 1
C              FMT IS  (1H ,I10,5X, AND E17.7, OR I7,11X DEPENDING ON
C              THE TYPE OF THE SUBSTITUTION PARAMETER
      WRITE( 6, FMT )  J, ( SUBST(K), K= MTEMP, NTEMP )
      NTEMP = NTEMP + NSUBP
      MTEMP = MTEMP + NSUBP
      LINE = LINE + 1
      IF( LINE .LT. 55 )  GO TO 15500
C
C              START NEW PAGE WITH TITLE AND SUBSTITUTION SYMBOL HEADING
      CALL PRTCTL
```

```
      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                                                           89902000
      END                                                              89905000
```

SUBROUTINE STAR9
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
LMODE	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
IFLAG	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
CDIST	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
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 SYMBLS ORIGIN 06667 LENGTH 02424

SDESN	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	FMT	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
RFMT	11465	R	LTEMP	11466	I	MTEMP	11467	I
NTEMP	11470	I						

ENTRY POINTS

GEOMETRY AND SUBSTITUTION INPUT PRINT ROUTINE 11/08/67 PAGE 9

STAR9Z STORAGE MAP

STAR9	SECTION	9

SUBROUTINES CALLED

PRTCTL	SECTION	10	.FWRD.	SECTION	11	ERROR2	SECTION	12
.FXEM.	SECTION	13	.UNDS.	SECTION	14	.FFIL.	SECTION	15
.FCNV.	SECTION	16	SYSLOC	SECTION	17			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	FORMAT	11535	32010	FORMAT	11705	1000	18A	12457
32020	FORMAT	12046	32030	FORMAT	12053	11000	234A	13335
1800	31A	12531	2000	39A	12557	3000	59A	12632
3500	81A	12637	5000	94A	12725	6000	121A	13005
7000	158A	13105	8000	166A	13113	2100	43A	12566
10000	202A	13211	2200	47A	12575	2300	51A	12606
2400	55A	12620	9000	200A	13204	4000	62A	12643
4100	69A	12656	4600	86A	12707	4200	73A	12665
4500	83A	12700	4300	78A	12673	4700	89A	[illegible]35
5100	97A	12726	5200	105A	12741	5300	109A	[illegible]2750
5400	113A	12761	5500	117A	12773	6100	124A	13008
62[illegible]	134A	13025	6600	150A	13064	6300	139A	13036
6400	143A	13045	6500	147A	13055	6700	153A	13075
7100	161A	13106	8200	173A	13120	8100	169A	13114
8300	179A	13131	8600	192A	13163	8400	184A	13142
8500	189A	13154	8700	195A	13174	32050	FORMAT	12171
32060	FORMAT	12177	10200	220A	13270	32070	FORMAT	12230
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	278A	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.

```
$*
$TITLE          DESIGN, SENSITIVITY, AND PROFILE PLOT PRINT ROUTINE
$IBFTC STAR12  M94,XR7              LINK 1    (STAR10)                    $1001000
```

```
      SUBROUTINE  STAR1D                                              S1001200
C                                                                     
C     *** DESCRIPTION ***                                             S1001202
C         STAR1D LISTS ALL OF THE DESIGN INPUT DATA WITH DESCRIPTIVE  S1001204
C         TITLES.  IN ADDITION IT LISTS ...                           S1001206
C                                                                     S1001208
C                  BOUNDARY CONDITION MATRIX  (IF NCOND.NE.0)         S1001210
C                  SENSITIVITY DATA           (IF NPERTB.NE.0)        S1001212
C                  PROFILE PLOT DATA          (IF NPTS.NE.0)          S1001214
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), SPPEA ), ( 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), 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
      EQUIVALENCE  ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )      DT103000
      INTEGER  CONTRL, 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                                           PT002500
      INTEGER PAGE                                                    PT002700
C
C           THIS BLOCK OF COMMON CONTAINS PERTURBATION VALUES         PB001000
      COMMON / PERTB /  PERTB(30,4), NPERTB, REFOCS                   PB002000
      EQUIVALENCE  ( PERTB(1,1), IPERTB(1,1) )                        PB003000
      DIMENSION  IPERTB(30,4)                                         PB004000
      INTEGER  REFOCS                                                 PB005000
C
      COMMON  / PLOTC /  YMAXX, DZMIN, NPTS                           PC001000
```

```
C
      DATA  BLANK / 6H       /, MINUS / 6H-      /                    S1002000
      INTEGER  BLANK                                                  S1002200
      DIMENSION  ISIGN(6)                                             S1002400
C
C           START OF STAR1D SUBROUTINE                                S1003000
C
      CALL PRTCTL                                                     S1003200    2
      WRITE ( 6, 32000 )  FNUMB, FLNGH, WFLGH, ZETA, HEXPP, DEXPP,    S1004000
     1 WEXPP, ( I, WOBJH(I), WCLRH(I), WIMHT(I), EIMHT(I), I = 1,NOBJH )S1004100    3
32000 FORMAT( 1H0, 53HREQUIRED F/ NUMBER........................... FNS1004500
     1UMB, E19.7 / 54H REQUIRED FOCAL LENGTH.......................... FS1004600
     2LNGH, E19.7 / 54H WEIGHT ON FOCAL LENGTH......................... S1004700
     3WFLGH, E19.7 / 53H CORRECTION VECTOR SCALE FACTOR.................S1004800
     4 ZETA, E20.7 / 54H OBJECT HEIGHT FOR EXIT PUPIL PLANE.............S1004900
     5 HEXPP, E19.7 / 54H REQUIRED POSITION EXIT PUPIL PLANE............S1005000
     6. DEXPP, E19.7 / 54H WEIGHT ON EXIT PUPIL PLANE POSITION..........S1005100
     7.. WEXPP, E19.7 // 21X, 44H ***** OBJECT HEIGHT DEPENDENT WEIGHTS S1005200
     8***** // 4X, 14H OBJECT HEIGHT, 7X, 5HWOBJH, 11X, 5HWCLRH, 11X,   S1005300
     9 5HWIMHT, 11X, 5HEIMHT / ( I12, 6X, 4E16.7 ) )                  S1005400
C
      WRITE ( 6, 32010 )  ( I, WCLRS(I), I = 1, NCLRS )               S1006000   13
32010 FORMAT( 1H0, 23X, 36H ***** COLOR DEPENDENT WEIGHTS ***** // 31X, S1006500
     1 6H COLOR, 6X, 6H WCLRS / ( I35, 4X, E14.7 ) )                  S1006600
C
      WRITE ( 6, 32020 )  WXDIR, WYDIR, SPFEA, OMGAF, DUMIN, SYSMX,NCONDS1007000   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..........S1008200
     3 SPFEA, E19.7 / 54H SCALE FACTOR FOR FEATHER CHECK RAY............S1008300
     4. OMGAF, E19.7/ 54H MINIMUM PARAMETER INCREMENT...................S1008400
     5. DUMIN, E19.7 / 54H MAXIMUM LENGTH OF SYSTEM.....................S1008500
     6.. SYSMX, E19.7 /54H NUMBER OF BOUNDARY CONDITIONS................S1008600
     7.. NCOND, I9 )                                                  S1008700
      IF( NCOND .LE. 0 ) GO TO 2500                                   S1041000
C
C           BOUNDARY CONDITIONS ARE PRESENT                           S1041200
      WRITE ( 6, 32030 )                                              S1041400   23
32030 FORMAT( 1H0, 22X, 38H ***** BOUNDARY CONDITION MATRIX ***** )   S1041600
      WRITE ( 6, 32035 )                                              S1042000   24
32035 FORMAT( 1H0, 17X, 9HPARAMETER, 10X, 7HMINIMUM, 10X, 7HMAXIMUM )  S1042200
      LINE = LINE + NOBJH + NCLRS + 29                                S1042400
      DO  2000  I = 1, NCOND                                          S1043000
      WRITE( 6, 32040 )  BDYSYM(1,I), BDYSYM(2,I), BOUNDS(2,I),       S1043200
     1 BOUNDS(3,I)                                                    S1043300   28
32040 FORMAT( 1H , 17X, 2A6, 2E17.7 )                                 S1043400
      LINE = LINE + 1                                                 S1043600
      IF( LINE .LT. 55 )  GO TO 2000                                  S1043800
C
C           BEGIN NEW PAGE  -  PRINT TITLE                            S1044000
      CALL  PRTCTL                                                    S1044200   37
      WRITE ( 6, 32035 )                                              S1044500   38
      LINE = LINE + 2                                                 S1044600
 2000 CONTINUE                                                        S1045000
 2500 IF( LINE .GT. 45 )  CALL  PRTCTL                                S1045500
C                                                                                44
```

```
      WRITE ( 6, 32050 )  ( NDSGN(I), I=1,4 ), NDSGV(1)                 S1046000    45
32050 FORMAT( 1H0, 56HTOTAL NUMBER OF DESIGN VARIABLES.............. NDS1046500
     1SGN(1), I4 / 57H MINIMUM PERMISSIBLE DESIGN VARIABLE NUMBER..... NS1046600
     2DSGN(2), I4 / 57H MAXIMUM PERMISSIBLE DESIGN VARIABLE NUMBER..... S1046700
     3NDSGN(3), I4 / 57H NUMBER OF DESIGN ITERATIONS....................S1046800
     4 NDSGN(4), I4 / 57H NUMBER OF DESIGN VARIABLES PER ITERATION......S1046900
     5, NDSGV(1), I4 )                                                  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, 10I3 )                                                S1048100
      WRITE ( 6, 32070 )  NJAIL, AFLAG                                  S1048500    59
32070 FORMAT( 1H , 53HNUMBER OF SKIPS FOR BAD COMBINATION............ NJS1048700
     1AIL, I7 / 55H SPECIAL DESIGN ITERATION PRINTOUT FLAG......... ITNPS1048800
     2RT, I6 )                                                          S1048900
      WRITE ( 6, 32072 )  ATRGGR                                        S1048910    60
32072 FORMAT(1H , 54HOPTION FLAG TO RE-COMPUTE D(J)................ ATRS1048912
     1GGR, I6 )                                                         S1048914
      WRITE ( 6, 32076 )  GAUSS                                         S1048924    61
32076 FORMAT(1H , 53HSTANDARD DEVIATION FOR GAUSSIAN WEIGHTING...... GAUS1048926
     1S5, E19.7 )                                                       S1048928
      LINE = LINE + 11                                                  S1049000
      IF( NDSGN(1) .EQ. 0 )  GO TO 5000                                 S1049100
      IF( LINE .GT. 40 )  CALL  PRTCTL                                  S1049200
C
C              PRINT OUT DESIGN INFORMATION                             S1061000
C                                                                                   67
      WRITE ( 6, 32080 )                                                S1061500    68
32080 FORMAT( 1H0, 21X, 30H***** DESIGN INFORMATION ***** )             S1062000
      WRITE ( 6, 32090 )                                                S1062500    69
32090 FORMAT( 1H0, 4H ID., 4H NO., 32X, 18H DESIGN PARAMETERS, 35X,     S1063000
     1 11HACTUAL DET., 2X, 9HREQ. DET., 3X,  9HINCREMENT )              S1063100
      LINE = LINE + 4                                                   S1063500
      ITEMP = NDSGN(1)                                                  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 , I2, I4, 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  PRTCTL                                                      S1066200    110
      WRITE ( 6, 32090 )                                                S1066500    111
      LINE = LINE + 2                                                   S1066700
 4000 CONTINUE                                                          S1067000
```

```
5000 IF( NPERTB .EQ. 0 )  GO TO 6000                                     S1070000
C
C            PRINT OUT PERTURBATION INFORMATION                            S1071000
C
      LINE = LINE + NPERTB + 7                                             S1071100
      IF( LINE .GT. 54 )  CALL  PRTCTL                                     S1071200    120
      WRITE ( 6, 32110 )                                                   S1071600    121
32110 FORMAT( 1H0, 18X, 35H***** SENSITIVITY INFORMATION ***** )          S1071800
      WRITE ( 6, 32112 )  NPERTB                                                       122
32112 FORMAT( 1H0, 54HNUMBER OF PARAMETERS............................ NP
     1ERTB, I6 )
      NREFC = REFOCS + 1
C
      GO TO ( 5100, 5150, 5200, 5250 ),  NREFC
 5100 WRITE ( 6, 32114 )  REFOCS                                                       124
32114 FORMAT( 1H , 54HREFOCUS OPTION.................................. RE
     1FOCS, 5X, I1, 3X, 14HDO NOT REFOCUS  )
      GO TO 5300
 5150 WRITE ( 6, 32115 )  REFOCS                                                       126
32115 FORMAT( 1H , 54HREFOCUS OPTION.................................. RE
     1FOCS, 5X, I1, 3X, 22HREFOCUS FOR MIN. RMS X  )
      GO TO 5300
 5200 WRITE ( 6, 32116 )  REFOCS                                                       128
32116 FORMAT( 1H , 54HREFOCUS OPTION.................................. RE
     1FOCS, 5X, I1, 3X, 22HREFOCUS FOR MIN. RMS Y  )
      GO TO 5300
 5250 WRITE ( 6, 32117 )  REFOCS                                                       130
32117 FORMAT( 1H , 54HREFOCUS OPTION.................................. RE
     1FOCS, 5X, I1, 3X, 26HREFOCUS FOR MIN. SPOT SIZE  )
C
 5300 WRITE ( 6, 32118 )                                                               131
32118 FORMAT( 1H0, 2X, 3HNO., 10X, 9HPARAMETER, 10X, 8HPER CENT )
      DO  5500  I = 1, NPERTB                                              S1073000
      WRITE ( 6, 32120 )  I, PERTB(1,3), PERTB(1,4), PERTB(1,1)           S1073400    134
32120 FORMAT( 1H , I4, 10X, 2A6, 10X, E14.7 )                              S1073600
 5500 CONTINUE                                                             S1073800
 6000 IF( NPTS .LE. 0 )  GO TO 10000                                       S1081000
C
C            PRINT OUT CROSS SECTION PLOT INFORMATION                      S1081200
      LINE = LINE + 6                                                      S1081400
      IF( LINE .GT. 54 )  CALL PRTCTL                                      S1081600    146
      WRITE ( 6, 32150 )                                                   S1082000    147
32150 FORMAT( 1H0, 15X, 42H***** CROSS SECTION PLOT INFORMATION ***** ) S1082200
      WRITE ( 6, 32155 )  YMAXX, DZMIN, NPTS                               S1083000    148
32155 FORMAT( 1H0, 53HDIAMETER OF LENS PLOT......................... YMS1083200
     1AXX, E19.7 / 54H MINUMIM AIR SPACE SEPARATION................... DS1083400
     2ZMIN, E19.7 / 53H NUMBER OF SURFACE POINTS PER ELEMENT........... S1083600
     3NPTS, I8 )                                                           S1083800
C
C            END OF STAR10 SUBROUTINE                                      S1091000
10000 RETURN                                                               S1093000
      END                                                                  S1095000
```

SUBROUTINE STAR10
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
NSUBF	00041	I	NIPLN	00042	I	IMODE	00043	I
NSFLN	00044	I	NOBJH	00045	I	NSURF	00046	I
IFLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
HEXPF	00055	R	DEXPF	00056	R	WEXPF	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
DDIST	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
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	ATRGGR	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
IPERTB	00000	I						

COMMON BLOCK			PLOTC	ORIGIN	11512	LENGTH	00003	
YMAXX	00000	R	DZMIN	00001	R	NPTS	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

STAR1Z STORAGE MAP

ENTRY POINTS

STAR10 SECTION 13

SUBROUTINES CALLED

PRTCTL	SECTION	14	.FWRD.	SECTION	15	.FXEM.	SECTION	1
.UN06.	SECTION	17	.FFIL.	SECTION	18	.FCNV.	SECTION	1
SYSLOC	SECTION	20						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIC
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.

```
$>
$TITLE          SUBROUTINE TO UNPACK A BCD WORD
$IBFTC UNBCD.   M94,XR7              LINK 1    (UNPBCD)                  UDCD0010
```

```
     SUBROUTINE TO UNPACK A BCD WORD                                    09/19/68                 PAGE
        UNBCD.      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  UNPBCD( BCDWRD, BCD )                                UBCD0020
C                                                                      
C     *** INPUTS ***                                                   UBCD0821
C BCDWRD  CONTAINS A 6 CHARACTER WORD TO BE UNPACKED INTO 6 WORDS.     UBCD0822
C                                                                      UBCD0823
C     *** OUTPUTS ***                                                  UBCD0824
C BCD      FIRST OF 6 WORDS WHICH CONTAIN THE 6 CHARACTERS--RIGHT ADJUST UBCD0825
C          ED WITH LEADING ZEROS.                                      UBCD0826
C                                                                      UBCD0827
C     *** DESCRIPTION ***                                              UBCD0828
C        UNPBCD UNPACKS THE SYMBOL IN BCDWRD INTO 6 CHARACTERS WHICH IT UBCD0829
C        STORES RIGHT ADJUSTED WITH LEADING ZEROS INTO BCD(1) THRU BCD(6UBCD0830
C        ), RESPECTIVELY.  THE DECODE/ENCODE CAPABILITY OF FORTRAN IS   UBCD0831
C        USED TO PERFORM THE UNPACKING.                                UBCD0832
C
      INTEGER BCDWRD,ACD(6),BCD(6)                                     UBCD0050
      DO 10 I=1,6                                                      UBCD0100
      ACD(I)=0                                                         UBCD0110
10    BCD(I)=0                                                         UBCD0120
      DECODE (6,1000,BCDWRD) ACD                                       UBCD0130      9
1000  FORMAT(6A1)                                                      UBCD0140
      ENCODE (6,1010,BCD) ACD                                          UBCD0150     11
1010  FORMAT(5H00000A1)                                                UBCD0160
      RETURN                                                           UBCD0170
      END                                                              UBCD0180
```

STORAGE MAP

SUBROUTINE UNPBCD
DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ACD	00001	1						

ENTRY POINTS

UNPBCD	SECTION	3

SUBROUTINES CALLED

.DECD.	SECTION	4	.FSLI.	SECTION	5	.ENCD.	SECTION	
.FSLO.	SECTION	7	.FRTN.	SECTION	8	.FCNV.	SECTION	
.FFIL.	SECTION	10	SYSLOC	SECTION	11			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATI
10	6A	00024	1000	FORMAT	00017	1010	FORMAT	00020

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00105.

```
    $JOB   INF,N010001,44800-0,77387,AM IC1,3,5000    L. WILSON INFORMATICS       A 13*51**29*** 11/08/67 *****VI3HJO*
$IBJOB         NOGO
$*
$TITLE         VECTOR-MATRIX BCD TO INTEGER CONVERT ROUTINE
$IBFTC VCNVR.  M94,XR7               LINK 1    (VCNVRT,MCNVRT)              VC001030
```

```
        VECTOR-MATRIX BCD TO INTEGER CONVERT ROUTINE                    11/08/67                PAGE  1
          VCNVR.     - EFN   SOURCE STATEMENT  -  IFN(S)  -
******

      SUBROUTINE  VCNVRT( BCDWRD, INTEG, * )                            VC001100
C                                                                        
C     *** INPUTS ***                                                    VC001102
C BCDWRD  LOCATION CONTAINING A BCD VECTOR SUBSCRIPT OF THE FORM X)    ,VC001104
C         XX)    , XXX)                                                 VC001106
C                                                                       VC001108
C     *** OUTPUTS ***                                                   VC001110
C INTEG   LOCATION WHERE VCNVRT IS TO STORE INTEGER SUBSCRIPT           VC001112
C *        ALTERNATE RETURNIF BCDWRD IS INVALID                         VC001114
C                                                                       VC001116
C     *** DESCRIPTION ***                                               VC001118
C        VCNVRT CONVERTS A GIVEN VECTOR SUBSCRIPT FROM BCD TO INTEGER.  VC001120
C        USED BY READS WHEN PROCESSING SYMBOL CARDS.                    VC001122
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 / 6H00000), 6H00000, /                                           VC001600
C                                                                       VC001700
      CALL  UNPBCD( BCDWRD, IBCD )                                      VC002100     2
      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  MCNVRT( 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     * . D.SCRIPTION ***                                               VC003720
C        .. 'RT 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    24
      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                      11/08/67                    PAGE  2
      VCNVR.     - EFN   SOURCE STATEMENT  -  IFN(S)  -

      K = 0                                                          VC004600
      J = J + 1                                                      VC004700
      GO TO 800                                                      VC004800
  700 INTEG = 10**(J-1) * INTEG + I - 1                              VC004900   45
      J = J + 1                                                      VC005000
      GO TO 900                                                      VC005100
C             DETERMINE SECOND SUBSCRIPT, SAVE IN JNTEG              VC005200
  800 JNTEG = 0                                                      VC005300
  900 DO  1000 I = 1, 10                                             VC005400
      IF( IBCD(J) .EQ. IDIGT(I) )  GO TO 1100                        VC005500
 1000 CONTINUE                                                       VC005600
      IF( IBCD(J) .NE. RTPRN  .OR.  J .GT. 6 )  RETURN 1             VC005700
      RETURN                                                         VC005800
 1100 JNTEG = 10**K * JNTEG + I - 1                                  VC005900   65
      K = K + 1                                                      VC006000
      J = J + 1                                                      VC006100
      GO TO 900                                                      VC006200
      END                                                            VC006300
```

VCNVR. STORAGE MAP

SUBROUTINE VCNVRT

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IDIGT	00001	I	IBCD	00013	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RTPRN	00021	I	COMMA	00022	I	J	00023	I
I	00024	I	K	00025	I			

ENTRY POINTS

VCNVRT	SECTION	3	HCNVRT	SECTION	4

SUBROUTINES CALLED

UNPBCD	SECTION	5	.XP1.	SECTION	6	SYSLOC	SECTION	7

EFN IFN CORRESPONDENCE

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.

```
                    $INF,ND10001,44903-0,77387,AM IC      $BJOB L10MKC  7094 A    09/25/68         PAGE

$*
$TITLE          MTF CARD PUNCH ROUTINE
$IBFTC MTFFNZ  M94,XR7               LINK 1    (MTFFN)                      MT000000
```

```
  MTF CARD PUNCH ROUTINE                                                          09/25/68                    PAGE
     MTFPNZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

       SUBROUTINE  MTFPN                                                    MT100000
                                                                            MT001001
C                                                                           MT001002
C      *** DESCRIPTION ***                                                  MT001003
C         MTFPN, UTILIZING FOLDP DATA, PUNCHES FACOS NAMELIST $INPUT         MT001004
C         CARDS FOR FACOS MTF CAPABILITY.                                   MT001005
C                                                                           DT001000
       COMMON / DATA / NCNTRL, CONTRL(10), TITLE(12), DATE(3), PUNCID,      DT001100
     1 DATA(3486)                                                           DT002000
       DIMENSION  WOBJH(7), WCLRH(7), WIMHT(7), EIMHT(7), CIMPL(7),        DT002100
     1 WCLRS(6), NDSGN(4), NDSGV(11), LATTC(3,26), BOUNDS(3,100),           DT002200
     2 SURFC(20,100), DESGN(10,50), SUBST(500), ILATTC(3,26),               DT002300
     3 ISURFC(20,100), IDESGN(10,50), ISUBST(500), IBNDS(3,100)             DT101000
       EQUIVALENCE  ( DATA(1), LMODE ), ( DATA(2), NRAYS ), ( DATA(3),      DT101100
     1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),          DT101200
     2 NSUBT ), ( DATA(7), NSUBP ), ( DATA(8), NIPLN ), ( DATA(9),          DT101300
     3 IMODE ), ( DATA(10), NSPLN ), ( DATA(11), NOBJH ), ( DATA(12),       DT101400
     4 NSURF ), ( DATA(13), AFLAG ), ( DATA(14), DELY ), ( DATA(15),        DT101500
     5 FNUMB ), ( DATA(16), FLNGH ), ( DATA(17), WFLGH ),( DATA(18),        DT101600
     6 ZETA ),  ( DATA(19), HEXPP ), ( DATA(20), DEXPP ), ( DATA(21),       DT101700
     7 WEXPP ), ( DATA(22), DLPLN ), ( DATA(23), OMGA2 ), ( DATA(24),       DT101800
     8 OMGA1 ), ( DATA(25), DELD  ), ( DATA(26), EFRAD ), ( DATA(27),       DT101900
     9 FSCAL ), ( DATA(28), OMGAF ), ( DATA(29), SPFEA ), ( DATA(30),       DT102000
     A DUMIN ), ( DATA(31), ODIST ), ( DATA(32), HO    ), ( DATA(33),       DT102100
     B DELH  ), ( DATA(34), SYSMX ), ( DATA(35), WXDIR ), ( DATA(36),       DT102200
     C WYDIR ), ( DATA(37), ROTAN ), ( DATA(38), NDSGN(1) ), ( DATA(42),DT102300
     D NDSGV(1) ), ( DATA(53), WOBJH(1) ), (DATA(60), WCLRH(1) ),           DT102400
     E ( DATA(67), WIMHT(1) ), (DATA(74), EIMHT(1) ), ( DATA(81),           DT102500
     F CIMPL(1) ), ( DATA(88), WCLRS(1) ),                                  DT102600
     G ( DATA(94), LATTC(1,1), ILATTC(1,1) ), ( DATA(182), SURFC(1,1),      DT102700
     H ISURFC(1,1) ), ( DATA(2182), DESGN(1,1), IDESGN(1,1) ),              DT102800
     I ( DATA(2682), SUBST(1), ISUBST(1) ), ( DATA(3182), BOUNDS(1,1),      DT102900
     J IBNDS(1,1) ), ( DATA(3483), NCOND )                                  DT103000
       EQUIVALENCE  ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )           DT201000
       INTEGER  CONTRL, AFLAG, DATE, ATRGGR                                 DT202000
       REAL  LATTC                                                          MT001005
C                                                                           MT001007
       DIMENSION  BETAP(100), BETAS(100), DELMU(100), DFORM(4,100),         MT001008
     1 RHOS(100), SKAPA(100), TOK(10), TS(100), XMUS(100)                   MT001020
C                                                                           MT001021
C         START OF MTFPN SUBROUTINE                                         MT001022
C                                                                           MT001023
C         COMPUTE MTF INPUT FROM FOLDP PARAMETERS                           MT001030
       BETAO = EFRAD                                                        MT001035
       N = NSURF + 1                                                        MT001040
       NK = NOBJH                                                           MT001045
       DO 1200 K = 1, NK                                                    MT001050
  1200 TOK(K) = SURFC(4,1)                                                  MT001055
       XLINV = 1./( ODIST - SURFC(4,1) )                                    MT001100
       DO 1000 J = 1,N                                                      MT001105
       IF( SURFC(2,J) .NE. 0. )  GO TO 100                                  MT001110
       BETAP(J) = 0.                                                        MT001115
       BETAS(J) = 0.                                                        MT001120
       GO TO 500                                                            MT001125
   100 IF( SURFC(3,J) .EQ. 0. )  GO TO 200                                  MT001130
       CALL  ERROR2( 36H MTF ABORTED--RECTANGULAR APERTURE  , J )           MT001135    25
```

```
      MTF CARD PUNCH ROUTINE                                               09/25/68              PAGE
         MTFPNZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      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
      DELMU(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(I,J) = 0.                                                  MT001206
      GO TO 700                                                        MT001207
  650 IP = I + 12                                                      MT001210
      DFORM(I,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,  BETAO, N, NK, XLINV                                   MT001310    83
   10 FORMAT( 16H $INPUT   BETAO=, E15.8, 4H, N=, I2, 5H, NK=, I2,     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)=, I1, 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( DELMU(I) )                                  MT001435
      DCNT4 = DCNT4 + ABS( RHOS(I) )                                   MT001440
 1525 DCNT5 = DCNT5 + ABS( SKAPA(I) )                                  MT001445
      L = 4                                                            MT001450
```

```
      IF( N .LT. 4 )  L = N                                MT001455
      IF( DCNT1 .EQ. O. )  GO TO 1550                      MT001460
      PUNCH 35, ( BETAP(N2), N2=1,L )                      MT001465    124
   35 FORMAT( 10H BETAP(1)=, 4( E15.8, 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)=, I2, 4H*D., )                  MT001495
 1600 IF( DCNT2 .EQ. O. )  GO TO 1650                      MT001500
      PUNCH 45, ( BETAS(N3), N3=1,L )                      MT001505    144
   45 FORMAT( 10H BETAS(1)=, 4( E15.8, 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)=, I2, 4H*D., )                  MT001535
 1700 IF( DCNT3 .EQ. O. )  GO TO 1750                      MT001540
      PUNCH 55, ( DELMU(N4), N4=1,L )                      MT001545    164
   55 FORMAT( 10H DELMU(1)=, 4( E15.8, 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)=, I2, 4H*D., )                  MT001575
 1800 IF( DCNT4 .EQ. O. )  GO TO 1850                      MT001580
      PUNCH 65, ( RHOS(N5), N5=1,L )                       MT001585    184
   65 FORMAT( 9H RHOS(1)=, 4( E15.8, 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)=, I2, 4H*D., )                    MT001615
 1900 IF( DCNT5 .EQ. O. )  GO TO 1950                      MT001620
      PUNCH 75, ( SKAPA(N6), N6=1,L )                      MT001625    204
   75 FORMAT( 10H SKAPA(1)=, 4( E15.8, 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)=, I2, 4H*D., )                  MT001655
 2000 DCNT6 = O.                                           MT001660
      DCNT7 = O.                                           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. O. )  GO TO 2050                      MT001700
      PUNCH 85, ( TS(N7), N7=1,L )                         MT001705    237
   85 FORMAT( 7H TS(1)=, 4( E15.8, 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                                              09/25/68          PAGE
          MTFPN2      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

   90 FORMAT( 7H TE(1)=, I2, 4H*0., )                                  MT001735
 2100 IF( DCNT7 .EQ. 0. )  GO TO 2150                                  MT001740
      PUNCH 95, ( XMUS(N8), 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, NM1 )                                MT001760    266
      GO TO 2200                                                       MT001765
 2150 PUNCH 105, NM1                                                   MT001770    274
  105 FORMAT( 9H XMUS(1)=, I2, 4H*0., )                                MT001775
 2200 DFCNT = 0.                                                       MT001780
      DO 2225 J = 1, N                                                 MT001785
      DO 2225 I = 1, 4                                                 MT001790
 2225 DFCNT = DFCNT + ABS( 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)=, I3, 4H*0., )                            MT001845
 2300 PUNCH 120                                                        MT001850    316
  120 FORMAT( 16H IPROG=3, M=0  $ )                                    MT001855
C                                                                      MT001960
C        END OF SUBROUTINE MTFPN                                       MT001965
      RETURN                                                           MT001870
      END                                                              MT001875
```

MTF CARD PUNCH ROUTINE 09/25/68 PAGE 5
MTFPNZ

STORAGE MAP

SUBROUTINE MTFPN
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
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
ISPLN	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	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	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
WOBJH	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	ISUBS	05224	I	BOUNDS	06210	R
IBNDS	06210	I	NCOND	06665	I	ATRGGR	06666	I
GAUSS	06667	R						

DIMENSIONED PROGRAM VARIABLES

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

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
BETAO	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
NM1	11042	I	DFCNT	11043	R	NT4	11044	I

ENTRY POINTS

MTFPN SECTION 5

MTF CARD PUNCH ROUTINE   09/25/68   PAGE 6
MTFPNZ
STORAGE MAP

SUBROUTINES CALLED

ERROR2	SECTION	6	.FPUN.	SECTION	7	.FFIL.	SECTION	8
.FCNV.	SECTION	9	SYSLOC	SECTION	10			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1200	7A	11313	1000	61A	11510	100	21A	11343
500	39A	11371	200	27A	11355	300	35A	11366
600	50A	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	11240	2150	274A	12402
95	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.

Section 2

LINK 2

```
    $JOB   INF,N010001,44800-0,77387,AM IC1,6,6000    WILSON   -   INFORMATICS      A 08*17**56*** 02/16/68 ******L1DMK1#
$IBJOB         NOGO
$*
$TITLE         ROUTINE TO GENERATE POINTS ON A PLANE CONIC ARC
$IBFTC ARPTS7  M94,XR7               LINK 2   (ARPTS)                AR001000
```

```
      SUBROUTINE  ARPTS( IS, ZA, YA, ZB, YB, N, ZLST, YLST )              AR001900
C
C     *** INPUTS ***                                                      AR002800
C IS    SURFACE NUMBER SPECIFIES CURVE Y = F(Z)                           AR003000
C ZA    Z COORDINATE POINT A                                              AR003200
C YA    Y COORDINATE POINT A                                              AR003600
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   Z1=ZA  ZN=ZB                                   AR004200
C YLST YI   I=1,2,....,N   Y1=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.                      AR006000
C            2.  THE ZI ARE EQUALLY SPACED.                               AR006200
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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 WZXPP ), ( DATA(22), DLPLN ), ( DATA(23), OMGA2 ), ( DATA(24),     DT101700
     8 OMGA1 ), ( DATA(25), DELO  ), ( 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), 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
      DIMENSION  ZLST(1), YLST(1)                                         AR006500
C
      IF( N .LE. 1 )  RETURN                                              AR007000
      DELTAZ = ( ZB - ZA ) / FLOAT( N-1 )                                 AR007200
      RC = 2. * SURFC(11,IS)                                              AR007400
      BC = SURFC(12,IS) - 1.                                              AR007600
```

```
      DO  10  I = 1, N                                                      AR007800
      ZLST(I) = ZA + FLOAT(I-1) * DELTAZ                                    AR008000
   10 YLST(I) = SQRT( ABS( BC * ZLST(I) * ZLST(I) + RC * ZLST(I) ) )        AR008200
C                                                                                   14
      RETURN                                                                AR009000
      END                                                                   AR009200
```

ROUTINE TO GENERATE POINTS ON A PLANE CONIC ARC 02/16/66 PAGE 3
ARPTSZ STORAGE MAP

SUBROUTINE ARPTS
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
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NS[illegible]T	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	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	HO	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
RDTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WOBJH	00117	R	WCLRH	00126	R	WINHT	00135	R
EINHT	00144	R	CI[illegible]L	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			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DELTAZ	06667	R	RC	06670	R	BC	06671	R
I	06672	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
SYSLOC	SECTION	15						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	10A	06767						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07057.

```
$*
$TITLE          ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES
$IBFTC CELAIZ  M94,XR7              LINK 2    (CELAIR)                      EA001000
```

```
      SUBROUTINE  CELAIR( * )                                           EA001200
C                                                                       
C     *** INPUTS ***                                                    EA001500
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 ELMMTX MATRIX OF EDITED SURFACE ELEMENTS.           (ELMAIR COMMON) EA002300
C AIRMTX 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 ELEMENTSEA003100
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, CONTRL(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), 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), 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), 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), 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                                                                       
      COMMON/PRNT/LINE,PAGE                                             PT002500
      INTEGER PAGE                                                      PT002700
C                                                                       
      COMMON  / ELMAIR /  NELMT, NAIRS, ELMMTX(2,100), AIRMTX(2,100),   EL001000
     1 VTXCRD(100)                                                      EL002000
```

```
        ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES              02/16/68                    PAGE  2
           CELAIZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      INTEGER  ELMMTX, AIRMTX                                          EL003000
C
      DIMENSION  ECURL(2,100), ACURL(2,100)                            EA004000
C
      INTEGER  ECURL, ACURL                                            EA004500
C
C             START OF CELAIR SUBROUTINE                               EA005000
C
      IF( ODIST .GE. 0. )  GO TO 1000                                  EA005500
      WRITE ( 6, 32010 )                                               EA006000     4
32010 FORMAT( 1H0, 35HOBJECT IS TO THE RIGHT OF SURFACE 1 )            EA006200
      RETURN 1                                                         EA006400
C
 1000 SPRIME = SURFC(4,1) - ODIST                                      EA006600
      IF( SPRIME .LE. 0. )  GO TO 1500                                 EA006800
      WRITE ( 6, 32020 )                                               EA007000     6
32020 FORMAT( 1H0, 44HOBJECT IS TO THE RIGHT OF THE ENTRANCE PUPIL )   EA007200
      RETURN 1                                                         EA007400
C
 1500 VTXCRD(1) = SURFC(4,1)                                           EA007600
      IF( ISURFC(1,1) .LE. 3 )  GO TO 2500                             EA007800
      I = 1                                                            EA008000
 2000 WRITE ( 6, 32030 )  I, ISURFC(1,I)                               EA008200    13
32030 FORMAT( 1H0, 7HSURFACE, I3, 7HIS TYPE, I3, 14HNOT ACCEPTABLE )   EA008400
      RETURN 1                                                         EA008600
C
 2500 IF( NSURF .LE. 1 )  GO TO 3500                                   EA101000
      DO  3000  I = 2, NSURF                                           EA101200
      IF( ISURFC(1,I) .GT. 3 )  GO TO 2000                             EA101400
      VTXCRD(I) = VTXCRD(I-1) + SURFC(4,I)                             EA101600
      IF( VTXCRD(I) .GE. SPRIME )  GO TO 3000                          EA101800
      WRITE ( 6, 32040 )  I                                            EA102000    31
32040 FORMAT( 1H0, 33HOBJECT IS TO THE RIGHT OF SURFACE, I3 )          EA102200
      RETURN 1                                                         EA102400
 3000 CONTINUE                                                         EA102600
3500  VTXCRD(NSURF+1)=VTXCRD(NSURF)+DELD                               EA103000
      CALL  PRTCTL                                                     EA103200    38
      WRITE ( 6, 32050 )                                               EA103400    39
32050 FORMAT( 1H0, 47HVERTEX COORDINATES REFERENCED TO ENTRANCE PUPIL //EA103600
     1 2X, 7HSURFACE, 18X, 1HZ )                                       EA103800
      LINE = LINE + 4                                                  EA104000
      ZERO = 0.                                                        EA104200
      WRITE (6,32060) ZERO                                             EA104400    41
32060 FORMAT( 1H , 1X, 14HENTRANCE PUPIL, E18.7 )                      EA104600
      LINE = LINE + 1                                                  EA105000
      DO  4500  I = 1, NSURF                                           EA105200
      IF( LINE .LE. 56 )  GO TO 4000                                   EA105400
C             BEGIN NEW PAGE                                           EA105600
      CALL  PRTCTL                                                     EA105800    49
      WRITE ( 6, 32050 )                                               EA106000    50
      LINE = LINE + 4                                                  EA106200
 4000 WRITE ( 6, 32070 )  I, VTXCRD(I)                                 EA106400    51
32070 FORMAT( 1H , I5, E28.7 )                                         EA106600
 4500 LINE = LINE + 1                                                  EA106800
      WRITE ( 6, 32075 )  VTXCRD(NSURF+1)                              EA106900    56
32075 FORMAT( 1H , 1X, 11HIMAGE PLANE, E21.7 )                         EA106950
```

```
C
      NEC = 0                                                          EA107000
      NAC = 0                                                          EA107200
      DO  5000  I = 1, 2                                               EA107400
      DO  5000  J = 1, 100                                             EA107600
      ECURL(I,J) = 0                                                   EA107800
 5000 ACURL(I,J) = 0                                                   EA108000
      IF( NSURF .GT. 1 )  GO TO 7500                                   EA108200
      I = 1                                                            EA108400
      X1 = SURFC(5,1)                                                  EA108600
      X2 = SURFC(5,2)                                                  EA108800
      IF( X1 .EQ. 1. )  GO TO 6000                                     EA109000
 5500 WRITE ( 6, 32080 )  I                                            EA201000    78
32080 FORMAT( 1H0, 15HINDEX 1 SURFACE, I3, 13H IS NOT VALID )          EA201200
      RETURN 1                                                         EA201400
C
 6000 IF( X2 .NE. -1. )  GO TO 6500                                    EA201600
      NEC = NEC + 1                                                    EA201800
      ECURL(1,NEC) = I                                                 EA202000
      ECURL(2,NEC) = I                                                 EA202200
      NAC = NAC + 1                                                    EA202400
      ACURL(1,NAC) = I                                                 EA202600
      ACURL(2,NAC) = I+1                                               EA202800
      GO TO 13500                                                      EA203000
 6500 IF( X2 .GT. 1. )  GO TO 7000                                     EA203200
      I = I+1                                                          EA203400
      GO TO 5500                                                       EA203600
 7000 NEC = NEC + 1                                                    EA204000
      ECURL(1,NEC) = I                                                 EA204200
      ECURL(2,NEC) = I+1                                               EA204400
      GO TO 13500                                                      EA204600
C
 7500 DO  10000  I = 1, NSURF                                          EA204800
      X1 = SURFC(5,I)                                                  EA205000
      X2 = SURFC(5,I+1)                                                EA205200
      IF( ABS( X1 ) .LT. 1. )  GO TO 5500                              EA205400
      IF( ABS( X2 ) .GE. 1. )  GO TO 8000                              EA205600
      I = I+1                                                          EA205800
      GO TO 5500                                                       EA206000
 8000 IF( X1 .LT. 1. )  GO TO 9000                                     EA206200
      IF( X1 .NE. 1. )  GO TO 8100                                     EA206400
      IF( X2 .LE. 1. )  GO TO 8200                                     EA206600
 8100 NEC = NEC + 1                                                    EA207000
      ECURL(1,NEC) = I                                                 EA207200
      ECURL(2,NEC) = I+1                                               EA207400
      GO TO 10000                                                      EA207600
 8200 IF( X2 .EQ. -1. )  GO TO 8400                                    EA208000
      IF(X2.EQ.1.)GO TO 8210                                           EA208100
      I = I+1                                                          EA208200
      GO TO 5500                                                       EA208400
 8210 IF(NAC.NE.0)ACURL(2,NAC)=I+1                                     EA208500
      GO TO 10000                                                      EA208550
 8400 NEC = NEC + 1                                                    EA208600
      ECURL(1,NEC) = I                                                 FA208800
      ECURL(2,NEC) = I                                                 EA209000
      NAC = NAC + 1                                                    EA209200
```

```
      ACURL(1,NAC) = I                                    EA209400
      ACURL(2,NAC) = I+1                                  EA209600
      GO TO 10000                                         EA209800
 8600 IF( X2 .LT. 1. )  GO TO 8800                        EA301000
      IF(X2.NE.1.)GO TO 8710                              EA301200
 8700 NAC = NAC + 1                                       EA301400
      ACURL(1,NAC) = I                                    EA301600
      ACURL(2,NAC) = I+1                                  EA301800
      GO TO 10000                                         EA302000
 8710 IF(X2.NE.X1)GO TO 8100                              EA302050
      ECURL(2,NEC)=I+1                                    EA302100
      GO TO 10000                                         EA302150
 8800 IF( ABS( X2 ) .EQ. X1 )  GO TO 8100                 EA302200
      I = I+1                                             EA302400
      GO TO 5500                                          EA302600
 9000 IF( X1 .NE. -1. )  GO TO 9200                       EA302800
      IF( X2 .LT. -1. )  GO TO 8100                       EA303000
      IF( X2 .EQ. 1. )  GO TO 8400                        EA303200
      IF(X2.EQ.-1.)GO TO 8210                             EA303300
      I = I+1                                             EA303400
      GO TO 5500                                          EA303600
 9200 IF( X2 .GT. -1. )  GO TO 9400                       EA304000
      IF(X2.NE.-1.)GO TO 8710                             EA304200
      GO TO 8700                                          EA304400
 9400 IF( X2 .EQ. ABS( X1 ) )  GO TO 8100                 EA305000
      I = I+1                                             EA305200
      GO TO 5500                                          EA305400
10000 CONTINUE                                            EA306000
C
13500 NELMT = 0                                           EA401000
      NAIRS = 0                                           EA401200
      DO  14000  I = 1, 2                                 EA401400
      DO  14000  J = 1, 100                               EA401600
      ELMNTX(I,J) = 0                                     EA401800
14000 AIRMTX(I,J) = 0                                     EA402000
      NELMT = 1                                           EA403000
      ELMNTX(1,1) = ECURL(1,1)                            EA403200
      ELMNTX(2,1) = ECURL(2,1)                            EA403400
      IF( NEC .EQ. 1 )  GO TO 16000                       EA403600
      DO  15500  I = 2, NEC                               EA404000
      I1 = ECURL(1,I)                                     EA404200
      I2 = ECURL(2,I)                                     EA404400
      VX1 = VTXCRD(I1)                                    EA404600
      VX2 = VTXCRD(I2)                                    EA404800
      IF( VX1 .LE. VX2 )  GO TO 14500                     EA405000
      IT = I1                                             EA405200
      I1 = I2                                             EA405400
      I2 = IT                                             EA405600
      VXT = VX1                                           EA405800
      VX1 = VX2                                           EA406000
      VX2 = VXT                                           EA406200
14500 DO  15000  J = 1, NELMT                             EA406400
      J1 = ELMNTX(1,J)                                    EA406600
      J2 = ELMNTX(2,J)                                    EA406800
      VY1 = VTXCRD(J1)                                    EA407000
      VY2 = VTXCRD(J2)                                    EA407200
```

```
      IF( I1 .NE. J1 )  GO TO 14700                                   EA407400
      IF( I2 .EQ. J2 )  GO TO 15500                                   EA407600
14700 IF(ABS(VX1-VY1).GT.1.E-5)GO TO 15000                            EA407800
      IF( ISURFC(1,I1) .NE. ISURFC(1,J1) )  GO TO 15000               EA408000
      IF( SURFC(11,I1) .NE. SURFC(11,J1) )  GO TO 15000               EA408200
      IF( SURFC(12,I1) .NE. SURFC(12,J1) )  GO TO 15000               EA408400
      IF(ABS(VX2-VY2).GT.1.E-5)GO TO 15000                            EA408600
      IF( ISURFC(1,I2) .NE. ISURFC(1,J2) )  GO TO 15000               EA408800
      IF( SURFC(11,I2) .NE. SURFC(11,J2) )  GO TO 15000               EA409000
      IF( SURFC(12,I2) .EQ. SURFC(12,J2) )  GO TO 15500               EA409200
15000 CONTINUE                                                        EA409400
      NELMT = NELMT + 1                                               EA501000
      ELMMTX(1,NELMT) = I1                                            EA501200
      ELMMTX(2,NELMT) = I2                                            EA501400
15500 CONTINUE                                                        EA501600
C
16000 IF( NAC .EQ. 0 )  GO TO 32000                                   EA502000
      NAIRS = 1                                                       EA502200
      AIRMTX(1,1) = ACURL(1,1)                                        EA502400
      AIRMTX(2,1) = ACURL(2,1)                                        EA502600
      IF( NAC .EQ. 1 )  GO TO 32000                                   EA502800
      DO  17500  I = 2, NAC                                           EA503000
      I1 = ACURL(1,I)                                                 EA503200
      I2 = ACURL(2,I)                                                 EA503400
      VX1 = VTXCRD(I1)                                                EA503600
      VX2 = VTXCRD(I2)                                                EA503800
      IF( VX1 .LE. VX2 )  GO TO 16500                                 EA504000
      IT = I1                                                         EA504200
      I1 = I2                                                         EA504400
      I2 = IT                                                         EA504600
      VXT = VX1                                                       EA504800
      VX1 = VX2                                                       EA505000
      VX2 = VXT                                                       EA505200
16500 DO  17000  J = 1, NAIRS                                         EA505400
      J1 = AIRMTX(1,J)                                                EA505600
      J2 = AIRMTX(2,J)                                                EA505800
      VY1 = VTXCRD(J1)                                                EA506000
      VY2 = VTXCRD(J2)                                                EA506200
      IF( I1 .NE. J1 )  GO TO 16700                                   EA506400
      IF( I2 .EQ. J2 )  GO TO 17500                                   EA506600
16700 IF(ABS(VX1-VY1).GT.1.E-5)GO TO 17000                            EA506800
      IF(ABS(VX2-VY2).LE.1.E-5)GO TO 17500                            EA507000
17000 CONTINUE                                                        EA507200
      NAIRS = NAIRS + 1                                               EA507400
      AIRMTX(1,NAIRS) = I1                                            EA507600
      AIRMTX(2,NAIRS) = I2                                            EA507800
17500 CONTINUE                                                        EA508000
C
C             END OF CELAIR SUBROUTINE                                EA609000
C
32000 RETURN                                                          EA609200
      END                                                             EA609400
```

ROUTINE TO GENERATE EDITED ELEMENT AND AIR SPACE MATRICES 08/16/68 PAGE 6
CELAIR STORAGE MAP

SUBROUTINE CELAIR
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	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
U[illegible]ODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	INODE	00043	I
NSPLN	00044	I	NOBJN	00045	I	NSURF	00046	I
IFLAG	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
D[illegible]LN	00060	R	OMGA2	00061	R	OMGA1	00062	R
DE[illegible]D	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	00071	R	HD	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
HDTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
HOBJH	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	08665	I			

COMMON BLOCK PRNT ORIGIN 08667 LENGTH 00002

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LINE	00000	I	PAGE	00001	I			

COMMON BLOCK ELNAIR ORIGIN 08671 LENGTH 00766

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NELMT	00000	I	NAIRS	00001	I	ELMMTX	00002	I
AIRMTX	00312	I	VTXCRD	00622	R			

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ECUNL	07657	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	VXT	10514	R	J1	10515	I
J2	10516	I	VY1	10517	R	VY2	10520	R

ENTRY POINTS

CELAIR SECTION 9

SUBROUTINES CALLED

.FWRD.	SECTION	10	PRTCTL	SECTION	11	.UNDS.	SECTION	12
.FFIL.	SECTION	13	.FCNV.	SECTION	14	SYSLOC	SECTION	15
.FRET.	SECTION	16						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	5A	10873	32010	FORMAT	10540	1500	9A	10714
32020	FORMAT	10550	2500	15A	10745	2000	13A	10725
32030	FORMAT	10562	3500	35A	11014	3000	32A	11011
32040	FORMAT	10574	32050	FORMAT	10605	32060	FORMAT	10624
4500	53A	11.11	4000	51A	11077	32070	FORMAT	10633
32075	FORMAT	10636	5000	66A	11145	7500	98A	11264
6000	79A	11205	5500	78A	11172	32080	FORMAT	10644
6500	89A	11237	13500	192A	11575	7000	93A	11250
10000	189A	11572	8000	111A	11317	9000	165A	11522
8600	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	306A	12202	17000	327A	12261	16700	321A	12237

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12344.

```
$*
$TITLE          ROUTINE TO FIND POINT OF CLOSEST APPROACH OF TWO CURVES
$IBFTC CLOSSZ  M94,XR7              LINK 2    (CLOSS)                  CL001000
```

```
      SUBROUTINE  CLOSS( I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX, DZMIN,      CL001500
     1 Z1CLOS, Y1CLOS, Z2CLOS, * )                                       CL002000
C
C     *** INPUTS ***                                                     CL002100
C I1       SURFACE NUMBER OF FIRST SURFACE.                              CL002200
C Z1MIN   MINIMUM Z VALUE OF FIRST SURFACE.                              CL002300
C Z1MAX   MAXIMUM Z VALUE OF FIRST SURFACE.                              CL002400
C I2       SURFACE NUMBER OF SECOND SURFACE.                             CL002500
C Z2MIN   MINIMUM Z VALUE OF SECOND SURFACE.                             CL002600
C Z2MAX   MAXIMUM Z VALUE OF SECOND SURFACE.                             CL002700
C DZMIN   MINIMUM Z SURFACE SEPARATION.                                  CL002800
C
C     *** OUTPUTS ***                                                    CL003000
C Z1CLOS REVISED Z VALUE ON FIRST SURFACE.                               CL003100
C Y1CLOS Y VALUE CORRESPONDING TO Z1CLOS AND Z2CLOS.                     CL003200
C Z2CLOS REVISED Z VALUE ON SECOND SURFACE.                              CL003300
C *       ALTERNATE RETURN FOR NON-INTERSECTING SURFACES.                LC003400
C
C     *** DESCRIPTION ***                                                CL003500
C         THIS SUBROUTINE DETERMINES THE POINTS ( Z1CLOS, Y1CLOS ) AND   CL003600
C         ( Z2CLOS, Y1CLOS ) ON INTERSECTING SURFACES I1 AND I2 SUCH     CL003700
C         THAT THE DISTANCE BETWEEN THEM IS DZMIN.                       CL003800
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), 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), 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
      COMMON/ELMAIR/NELMT,NAIRS,ELMMTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
      INTEGER  ELMMTX, AIRMTX                                            EL002000
C
```

```
          ROUTINE TO FIND POINT OF CLOSEST APPROACH OF TWO CURVES          02/16/68                    PAGE  2
             CLOSSZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

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 = ISURFC(1,I1)                                                   CL006600
      R1 = SURFC(11,I1)                                                    CL006800
      B1 = SURFC(12,I1)                                                    CL007000
      AL1 = B1 - 1.                                                        CL007200
      IT2 = ISURFC(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                                                   CL204200
      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 02/16/68 PAGE 3
CLOSSZ STORAGE MAP

SUBROUTINE CLOSS
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
LNODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	INODE	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	[illegible]	R
DELD	00063	R	EPRAD	00064	R	PSCAL	[illegible]	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	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
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 ELMAIR ORIGIN 06667 LENGTH 00766

NELMT	00000	I	NAIRS	00001	I	ELMMTX	00002	I
AIRMTX	00312	I	VTXCRD	00622	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
B2	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	SQRT	SECTION	9	SYSLOC	SECTION	10
.FRET.	SECTION	11						

CLOSSZ STORAGE MAP

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
2500	6A	07726	2000	5A	07723	4000	26A	10007
5000	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	10175

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 10343.

```
$*
$TITLE          ROUTINE TO PLOT Z-Y PROFILE ON SC4020
$IBFTC CPLOTZ  M94,XR7           LINK 2    (CPLOT)                  CP001000
```

```
      SUBROUTINE  CPLOT( ZMIN, ZMAX )                                   CP001100
C
C     *** INPUTS ***                                                    CP001200
C ZMIN    MINIMUM Z OF SYSTEM RELATIVE TO ENTRANCE PUPIL.               CP001250
C ZMAX    MAXIMUM Z OF SYSTEM RELATIVE TO ENTRANCE PUPIL.               CP001300
C ZLST    Z COORDINATES OF POINTS ON CURRENT ELEMENT READ FROM BINARY   CP001350
C         TAPE ( LOGICAL 11 ).                                          CP001400
C YLST    Y COORDINATES OF POINTS ON CURRENT ELEMENT READ FROM BINARY   CP001450
C         TAPE ( LOGICAL 11 ).                                          CP001500
C NPTS    NUMBER OF POINTS FOR CURRENT ELEMENT, TAPE LOGICAL 11.        CP001550
C
C     *** OUTPUTS ***                                                   CP001600
C         PLOT FRAME CONTAINING Z-Y CROSS SECTION OF LENS SYSTEM FROM   CP001650
C         THE ENTRANCE PUPIL TO THE IMAGE SURFACE WITH OPTIC AXIS SHOWN. CP001700
C         THE SCALE IS THE SAME IN THE Z AND Y DIRECTIONS WITH THE Y    CP001750
C         RANGE FROM YMAXX TO -YMAXX.  A SCALED RULER IS POSITIONED     CP001800
C         BELOW THE LENS WITH ASSOCIATED LABELS.  BELOW THIS THE TITLE  CP001850
C         AND DATE ARE PLOTTED.  TWO SPECIAL RAYS AT THE LIMITS OF THE  CP001900
C         ENTRANCE PUPIL ARE ALSO TRACED ON THE LENS PLOT.              CP001950
C         A SECOND FRAME CONTAINING THE PLOT PARAMETERS AND THE VALUES  CP002000
C         FOR SURFACE RADIUS, SURFACE DISTANCE, SURFACE ECCENTRICITY,   CP002050
C         AND THE INTERSECTIONS OF THE SPECIAL RAYS WITH EACH SURFACE   CP002100
C         IS ALSO DRAWN.                                                CP002150
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), NRAYS ), ( 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), 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), 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
      COMMON/ELMAIR/NELMT,NAIRS,ELMNTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
      INTEGER  ELMNTX, AIRMTX                                           EL002000
```

```
     ROUTINE TO PLOT Z-Y PROFILE ON SC4020                           02/16/68                PAGE  2
        CPLOTZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C
      COMMON  / COORNG /  ZHAT(2,100), YHAT(2,100), CHAT(2,100)          CD001000
      INTEGER  CHAT                                                      CD002000
C
      COMMON  / PLOTC /  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
     1            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 /                                          CP003050
C
      DATA NZERO / 3H0.0 /,                                              CP003100
     1 NUM(1,1) / 3H0.1 /,                                               CP003150
     2(NUM(I,2), I=1,2 )                                                 CP003200
     B / 3H0.2, 3H0.4 /,
     3(NUM(I,3), I=1,3 )                                                 CP003250
     C / 3H0.3, 3H0.6, 3H0.9 /,
     4(NUM(I,4), I=1,4 )                                                 CP003300
     D / 3H0.4, 3H0.8, 3H1.2, 3H1.6 /,
     5(NUM(I,5), I=1,5 )                                                 CP003350
     E / 3H0.5, 3H1.0, 3H1.5, 3H2.0, 3H2.5 /,
     6(NUM(I,6), I=1,6 )                                                 CP003400
     F / 3H0.6, 3H1.2, 3H1.8, 3H2.4, 3H3.0, 3H3.6 /,
     7(NUM(I,7), I=1,7 )                                                 CP003450
     G / 3H0.7, 3H1.4, 3H2.1, 3H2.8, 3H3.5, 3H4.2, 3H4.9 /,
     8(NUM(I,8), I=1,8 )                                                 CP003500
     H / 3H0.8, 3H1.6, 3H2.4, 3H3.2, 3H4.0, 3H4.8, 3H5.6, 3H6.4 /,
     9(NUM(I,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(I,10),I=1,9 )                                                 CP003600
     J / 3H1.0, 3H2.0, 3H3.0, 3H4.0, 3H5.0, 3H6.0, 3H7.0, 3H8.0, 3H9.0 /
C
C             START OF CPLOT SUBROUTINE                                  CP004000
C
C             DETERMINE SCALE OF PLOT                                    CP004500
      ALENG1 = 2. * YMAXX                                                CP004800
      ALENG2 = ZMAX - ZMIN                                               CP005000
      IF( ALENG1 .GT. ALENG2 )  GO TO 1000                               CP005200
C
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
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                                02/18/68                  PAGE  3
            CPLOTZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

 1600 CDR(I) = CD(I)                                                   CP009400
      CDR(2) = CDR(2) - RSCALE * 70.                                   CP009800
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 980 RASTERS FROM ZMIN TO RULX2     CP102400
      RULX2 = ZMIN + ( FLOAT(NORM)/FNORM ) * ( CD(3) - ZMIN )          CP102450
      RULY=CD(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
      RULYT5 = RULY - RSCALE * 5.                                      CP102320
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, RULYT, 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 SPRNTA( NT, XLABEL, 985,   +8, LABEL, 1 )                   CP103400    48
      CALL  STERM( NT, BUFR)                                           CP103405    50
      IF( I .EQ. 11 )  GO TO 1700                                      CP103410
C
C              DRAW 4 TICK MARKS, 5 RASTERS LONG, BETWEEN MAJOR TICKS  CP103415
      DO  1650  J = 1, 4                                               CP103420
      XTICK5 = XTICK + FLOAT(J) * ( RULXTI / 5. )                      CP103430
      CALL  SDNPUT( XTICK5, RULY, BUFR, 0 )                            CP103440    57
      CALL  SDNPUT( XTICK5, RULYT5, 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
```

```
      DO  4000  I = 1, NELMT                                          CP105200
      READ  ( 11 )  NPTS, ZLST, YLST                                  CP105400
C
C            PLOT UPPER HALF OF ITH. ELEMENT                          CP105600    78
      DO  2000  J = 1, NPTS                                           CP105800
 2000 CALL  SDNPUT( ZLST(J), YLST(J), BUF, 0 )                        CP106000    84
      CALL  STERM( NT, BUF )                                          CP106200
C
C            PLOT LOWER HALF OF ITH. ELEMENT                          CP106400    87
      DO  3000  J = 1, NPTS                                           CP106600
 3000 CALL  SDNPUT( ZLST(J), -YLST(J), BUF, 0 )                       CP106800    94
      CALL  STERM( NT, BUF )                                          CP107000    97
 4000 CONTINUE                                                        CP107200
C
C            IMAGE PLANE                                              CP108000
      READ  ( 11 )  NPTS, ZLST, YLST                                  CP108200   100
      IF( NPTS .EQ. 0 )  GO TO 6000                                   CP108400
      DO  5000  J = 1, NPTS                                           CP108600
 5000 CALL  SDNPUT( ZLST(J), YLST(J), BUF, 0 )                        CP108800   113
      CALL  STERM( NT, BUF )                                          CP201000   117
      DO  5500  J = 1, NPTS                                           CP201200
 5500 CALL  SDNPUT( ZLST(J), -YLST(J), BUF, 0 )                       CP201400   126
      CALL  STERM( NT, BUF )                                          CP201600
C
C            ENTRANCE PUPIL                                           CP202000   130
 6000 READ  ( 11 )  NPTS, ZLST, YLST                                  CP202200   131
      CALL  SDNPUT( ZLST(1), YLST(1), BUF, 0 )                        CP202400   135
      CALL  SDNPUT( ZLST(2), YLST(2), BUF, 0 )                        CP202600   137
      CALL  STERM( NT, BUF )                                          CP202800   139
      CALL  SDNPUT( ZLST(1), -YLST(1), BUF, 0 )                       CP203000   141
      CALL  SDNPUT( ZLST(2), -YLST(2), BUF, 0 )                       CP203200   143
      CALL  STERM( 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  SDNPUT( ZLST(J), YLST(J), BUF, 0 )                        CP204600   159
      CALL  STERM( NT, BUF )                                          CP204800
C
C            RAY 2 TRACE                                              CP205000   163
 7500 READ ( 11 )  NPTS, ZLST, YLST                                   CP205200   164
      IF( NPTS .LE. 0 )  GO TO 9000                                   CP205300
      DO  8000  J = 1, NPTS                                           CP205400
 8000 CALL  SDNPUT( 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  STERM( NT, BUF )                                          CP207400   196
      CALL  ADV( NT )                                                 CP207600   198
      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                                  02/16/68                    PAGE  5
         CPLOTZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      CALL 3ETPL( 990 )                                                 CP301400    202
      WRITE ( 18, 32100 )   ( TITLE(I), I=1,12 ), ( DATE(J), J=1,3 )    CP301600    203
      CALL  SETPL( 5 )                                                  CP302000    216
      WRITE ( 18, 32150 )  YMAXX, DZMIN, NPOINT, NLENG                  CP302200    217
32150 FORMAT(  5X, 13HTUBE RADIUS =, F5.2;  5X, 20HMINIMUM SEPARATION =,CP302400
     1 F6.3,  5X, 17H ELEMENT POINTS =, I4, 5X, 16HPLOT SCALE = 1(E, I2)CP302600
      WRITE ( 18, 32160 )                                               CP303000    218
32160 FORMAT( 1HO, 2X, 7HSURFACE, 12X, 1HR, 17X, 1HD, 17X, 1HB, 16X,    CP303200
     1 2HR1, 16X, 2HR2 )                                                CP303400
C
      REWIND  11                                                        CP304000    219
      NREAD = NELMT + 2                                                 CP304200
      DO  15000  I = 1, NREAD                                           CP304400
15000 READ ( 11 )  DUMMY                                                CP304600
C
C            READ RAY 1 DATA                                            CP305000    224
      READ ( 11 )  NPTS, ZLST, ZLST                                     CP305200
C
C            BEGIN LOOP  TO PLOT SURFACE PARAMETERS                     CP306000    226
      LRAY = 2                                                          CP306200
      DO  18000  I = 1, NELMT                                           CP306400
      DO  16000  J = 1, 2                                               CP306600
      NEL  = ELMMTX(J,I)                                                CP306800
      NELN = ELMMTX(2,I)                                                CP307000
      IF( J .EQ. 2 )  NELN = ELMMTX(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
C            BEGIN NEW PLOT FRAME                                       CP401400
      CALL  STERM( 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  STERM( NT, BUF )                                            CP404200    280
      CALL  ADV( NT )                                                   CP404400    282
      CALL  RESPL                                                       CP404600
C
C            END OF CPLOT SUBROUTINE                                    CP408000    284
      RETURN                                                            CP408200
      END                                                               CP408400
```

SUBROUTINE CPLOT
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	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	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
OLPLN	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
CDIST	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
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	08210	R
IBNDS	08210	I	NCOND	08665	I			

COMMON BLOCK ELMAIR ORIGIN 08667 LENGTH 00768

NELMT	00000	I	NAIRS	00001	I	ELMMTX	00002	I
AIRMTX	00312	I	VTXCRD	00822	R			

COMMON BLOCK CODRNG ORIGIN 07655 LENGTH 01130

ZHAT	00000	R	YHAT	00310	R	CHAT	00820	I
ZLST	00000	R	YLST	00454	R			

COMMON BLOCK PLOTC 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	12306	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 SC4020 02/16/68 PAGE 7

CPLOTZ

STORAGE MAP

FNORM	12323	R	NORM	12324	I	MULX2	12325	R
MULY	12326	R	MULYT	12327	R	MULXT1	12330	R
MULYTS	12331	R	XTICK	12332	R	LABEL	12333	I
NZERO	12334	I	XLABEL	12335	R	J	12336	I
XTICKS	12337	R	NPTS	12340	I	NREAD	12341	I
DUMMY	12342	R	LRAY	12343	I	NEL	12344	I
NELM	12345	I	DTEMP	12346	R			

ENTRY POINTS

CPLOT	SECTION	11

SUBROUTINES CALLED

CAMERA	SECTION	12	SLABEL	SECTION	13	SDINIT	SECTION	14
ALOG10	SECTION	15	.XP2.	SECTION	16	SDNPUT	SECTION	17
STERM	SECTION	18	SPRNTA	SECTION	19	.FRDB.	SECTION	20
.FBLI.	SECTION	21	SETPL	SECTION	22	.FWRD.	SECTION	23
ADV	SECTION	24	RESPL	SECTION	25	.FRMT.	SECTION	26
.UN11.	SECTION	27	.FRLR.	SECTION	28	.FBLT.	SECTION	29
.FBDT.	SECTION	30	.UN18.	SECTION	31	.FFIL.	SECTION	32
.FCNV.	SECTION	33	E.1	SECTION	34	E.2	SECTION	35
E.3	SECTION	36	E.4	SECTION	37	CC.1	SECTION	38
CC.2	SECTION	39	CC.3	SECTION	40	CC.4	SECTION	41
SYSLOC	SL.TION	42						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	6A	12506	1500	7A	12527	1600	11A	12534
1700	64A	13106	1650	62A	13104	4000	98A	13242
2000	81A	13167	3000	91A	13217	6000	131A	13350
5000	110A	13275	5500	123A	13325	7500	164A	13521
7000	156A	13475	9000	181A	13572	8000	174A	13553
32100	FORMAT	12400	32150	FORMAT	12407	32160	FORMAT	12434
15000	224A	13721	18000	277A	14153	16000	253A	14067
32170	FORMAT	12451						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 14234.

```
$*
$TITLE         ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS
$IBFTC CROSSZ  M94,XR7              LINK 2   (CROSS)                    CR001000
```

```
      SUBROUTINE  CROSS( I1, Z1MIN, Z1MAX, I2, Z2MIN, Z2MAX, ZCROSS,    CR001500
     1 YCROSS, * )                                                      CR001600
C
C     *** INPUTS ***                                                    CR002000
C I1      SURFACE NUMBER OF FIRST SURFACE.                              CR002100
C Z1MIN   MINIMUM Z VALUE OF FIRST SURFACE.                             CR002200
C Z1MAX   MAXIMUM Z VALUE OF FIRST SURFACE.                             CR002300
C I2      SURFACE NUMBER OF SECOND SURFACE.                             CR002400
C Z2MIN   MINIMUM Z VALUE OF SECOND SURFACE.                            CR002500
C Z2MAX   MAXIMUM Z VALUE OF SECOND SURFACE.                            CR002600
C
C     *** OUTPUTS ***                                                   CR003000
C ZCROSS Z VALUE OF INTERSECTION.                                       CR003100
C YCROSS Y VALUE OF INTERSECTION.                                       CR003200
C *       ALTERNATE RETURN FOR NON-INTERSECTING SURFACES.               CR003300
C
C     *** DESCRIPTION ***                                               CR003500
C         THIS SUBROUTINE DETERMINES THE INTERSECTION POINT, ( YCROSS,  CR003600
C         ZCROSS ), OF SURFACE I1 AND SURFACE I2.                       CR003700
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), 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), 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
      COMMON/ELMAIR/NELMT,NAIRS,ELMMTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
      INTEGER  ELMMTX, AIRMTX                                           EL002000
C
C             START OF CROSS SUBROUTINE                                 CR004000
C
      ZCROSS = 0.                                                       CR004200
```

```
        ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS          02/18/68                    PAGE  2
          CROSSZ     -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      YCROSS = 0.                                                       CR004400
      IT1 = ISURFC(1,I1)                                                CR004600
      R1  = SURFC(11,I1)                                                CR004800
      B1  = SURFC(12,I1)                                                CR005000
      IT2 = ISURFC(1,I2)                                                CR005200
      R2  = SURFC(11,I2)                                                CR005400
      B2  = SURFC(12,I2)                                                CR005600
      IF( I1 .EQ. I2 )  GO TO 32000                                     CR006000
      DEL1 = VTXCRD(I1)                                                 CR006200
      DEL2 = VTXCRD(I2)                                                 CR006400
      S = DEL2 - DEL1                                                   CR006600
      Z2MINP = Z2MIN + S                                                CR006800
      Z2MAXP = Z2MAX + S                                                CR007000
      ZA = 0.                                                           CR007200
      ZB = 0.                                                           CR007400
C
C             CHECK FOR INTERSECTION                                    CR100500
      IF( Z1MIN .GT. Z2MAXP )  GO TO 2700                               CR100600
      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. S )  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          08/16/68              PAGE 3
         CROSS2      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

 4000 DELTA = BETA * BETA - ALPHA * GAMMA                            CR108200
      IF( DELTA .LT. 0. )  GO TO 2700                                CR108400
      IF( DELTA .GT. 0. )  GO TO 4500                                CR108600
      Z1 = - BETA / ALPHA                                            CR108800
      GO TO 3800                                                     CR109000
 4500 Z1 = - BETA / ALPHA + SQRT( DELTA ) / ALPHA                    CR201000    80
      Z2 = - BETA / ALPHA - SQRT( 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 = AMIN1( Z1, Z2 )                                       CR203000
      IF( R1 .LT. 0. )  ZCROSS = AMAX1( Z1, Z2 )                     CR203200
      GO TO 2200                                                     CR203400
C
C               END OF CROSS SUBROUTINE                              CR204000
32000 RETURN                                                         CR205000
      END                                                            CR206000
```

ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS 02/16/68 PAGE 4
CROSS2 STORAGE MAP

SUBROUTINE CROSS
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
LMODE	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	NOBJM	00045	I	NSURF	00046	I
IFLAG	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
CDIST	00071	R	HD	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ADTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WOBJM	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 ELMAIR ORIGIN 06667 LENGTH 00766

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NELMT	00000	I	NAIRS	00001	I	ELMMTX	00002	I
AIRMTX	00312	I	VTXCRD	00622	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
Z2MINP	07666	R	Z2MAXP	07667	R	ZA	07670	R
ZB	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

SQRT SECTION 8 SYSLOC SECTION 9 .FRET. SECTION 10

ROUTINE TO FIND POINT OF INTERSECTION OF TWO PLANE CONICS 02/16/68 PAGE 9
CROSS2 STORAGE MAP

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	93A	10430	2700	59A	10203	3500	62A	10213
1500	40A	10084	1000	37A	10040	2500	53A	10126
2000	49A	10100	2200	50A	10102	3000	60A	10208
4000	71A	10273	3800	66A	10257	3900	69A	10270
4500	79A	10322	5000	89A	10377			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 10826.

```
$*
$TITLE          ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEMENT
$IBFTC ENDPTZ  M94,XR7              LINK 2    (ENDPTS)                   EP001000
```

```
      SUBROUTINE  ENDPTS( DZMIN, ARCS, NARCS )                          EP001500
C
C     *** INPUTS ***                                                    EP001600
C DZMIN  MINIMUM ALLOWABLE Z SURFACE SEPARATION.                        EP001700
C ELMMTX MATRIX OF EDITED SURFACE ELEMENTS.            (ELMAIR COMMON) EP001800
C VTXCRD VECTOR OF VERTEX COORDINATES OF THE SURFACES.  (ELMAIR COMMON) EP001900
C ZHAT   MATRIX OF RANGE OF Z DEFINITION OF EACH SURFACE(COORNG COMMON) EP002000
C YHAT   MATRIX OF RANGE OF Y DEFINITION OF EACH SURFACE(COORNG COMMON) EP002100
C CHAT   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 5 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, CONTRL(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), 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), 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), 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), 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
```

```
        ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME              02/18/68                PAGE  2
           ENDPTS       - EFN    SOURCE STATEMENT  -  IFN(S)  -

      INTEGER  CONTRL, AFLAG, DATE                                             DT201000
      REAL  LATTC                                                              DT202000
C
      COMMON/ELHAIR/NELMT,NAIRS,ELMMTX(2,100),AIRMTX(2,100),VTXCRD(100)      EL001000
      INTEGER  ELMMTX, AIRMTX                                                  EL002000
C
      COMMON  / COORNG /  ZHAT(2,100), YHAT(2,100), CHAT(2,100)               CD001000
      INTEGER  CHAT                                                            CD002000
C
      DIMENSION  ARCS(2,5,100), NARCS(100)                                     EP004080
C
C              START OF ENDPTS SUBROUTINE                                      EP004100
C
      DO  2000  K = 1, 100                                                     EP004200
      DO  1000  I = 1, 2                                                       EP004400
      DO  1000  J = 1, 5                                                       EP004600
 1000 ARCS(I,J,K) = 0.                                                         EP004800
 2000 NARCS(K) = 0                                                             EP005000
      IF( NELMT .GT. 1 )  GO TO 7000                                           EP005200
      IF( ELMMTX(1,1) .NE. ELMMTX(2,1) )  GO TO 3000                           EP006000
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 = ELMMTX(1,1)                                                         EP007200
      I2 = ELMMTX(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, S6000, 1, 2 )                       EP103400    40
      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            02/16/68                  PAGE  3
          ENDPTZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

C              LIMIT CURVE 1                                              EP105000
 5500 CALL  ZYCUT( 1, YHAT(2,1), I1, BETA, $6000, 1, 1 )                  EP105200    48
      ARCS(1,2,1) = BETA                                                  EP105400
      ARCS(2,2,1) = YHAT(2,1)                                             EP105600
      ARCS(1,3,1) = ZHAT(2,1) + S                                         EP105800
      ARCS(2,3,1) = YHAT(2,1)                                             EP106000
      ARCS(1,4,1) = S                                                     EP106200
      NARCS(1) = 4                                                        EP106400
      GO TO 32000                                                         EP106600
C
 6000 WRITE ( 6, 32010 )                                                  EP107000    49
32010 FORMAT( 1H0, 28HUNEXPECTED ERROR USING ZYCUT )                      EP107200
      CALL  GOOF                                                          EP107400    50
 7000 DO  18000  I = 1, NELMT                                             EP107600
      I1 = ELMNTX(1,I)                                                    EP107800
      I2 = ELMNTX(2,I)                                                    EP108000
      I3 = ELMNTX(1,I+1)                                                  EP108200
      I4 = ELMNTX(2,I+1)                                                  EP108400
      IF( I .EQ. NELMT  .OR.  I2 .EQ. I3 )  GO TO 14000                   EP108600
      Z1N = AMIN1( 0., ZHAT(2,I) )                                        EP108800
      Z1X = AMAX1( 0., ZHAT(2,I) )                                        EP109000
      Z2N = AMIN1( 0., ZHAT(1,I+1) )                                      EP109200
      Z2X = AMAX1( 0., ZHAT(1,I+1) )                                      EP109400
C
C              CHECK AIR SPACE CONSTRAINT                                 EP201000
      CALL  CLOSS( I2, Z1N, Z1X, I3, Z2N, Z2X, DZMIN, Z1C, Y1C, Z2C,      EP201200
     1 $14000 )                                                           EP201400    70
      I2C = CHAT(2,I)                                                     EP201600
      I3C = CHAT(1,I+1)                                                   EP201800
      IGO = 4                                                             EP202000
      IF( I2C .NE. 3 )  GO TO 8000                                        EP202200
      IGO = 1                                                             EP202400
      IF( I3C .EQ. 3 )  IGO = 3                                           EP202600
      GO TO 8400                                                          EP202800
 8000 IF( I3C .EQ. 3 )  IGO = 2                                           EP203000
 8400 GO TO ( 8500, 9500, 10400, 11500 ),  IGO                            EP203600
 8500 ZHAT(1,I+1) = Z2C                                                   EP203800
      YHAT(1,I+1) = Y1C                                                   EP204000
      S = VTXCRD(I4) - VTXCRD(I3)                                         EP204200
      IF( I3 .EQ. I4 )  GO TO 14000                                       EP204400
      IF( ISURFC(1,I2) .EQ. 3 )  GO TO 8800                               EP204600
      IF( SURFC(11,I2) .GT. 0. )  GO TO 9000                              EP204800
 8800 CHAT(1,I+1) = 4                                                     EP205000
      CALL  ZYCUT( 2, Z2C-S, I4, BETA, $14000 I+1, 2 )                    EP205200    107
 8900 ZHAT(2,I+1) = Z2C - S                                               EP205400
      YHAT(2,I+1) = BETA                                                  EP205600
      CHAT(2,I+1) = 1                                                     EP205800
      GO TO 14000                                                         EP206000
 9000 CHAT(1,I+1) = 5                                                     EP206200
      CALL  ZYCUT( 1, Y1C, I4, BETA, $14000,I+1, 2 )                      EP206400    116
 9200 ZHAT(2,I+1) = BETA                                                  EP206600
      YHAT(2,I+1) = Y1C                                                   EP206800
      CHAT(2,I+1) = 1                                                     EP207000
      GO TO 14000                                                         EP207200
 9500 ZHAT(2,I) = Z1C                                                     EP207400
      YHAT(2,I) = Y1C                                                     EP207600
```

```
      S = 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, Z1C+S, I1, BETA, $14000, I, 1 )                   EP301400    141
      ZHAT(1,I) = Z1C + S                                               EP301600
      YHAT(1,I) = BETA                                                  EP301800
      GO TO 14000                                                       EP302000
10000 CHAT(2,I) = 5                                                     EP302200
      CALL  ZYCUT( 1, Y1C, I1, BETA, $14000, I, 1 )                     EP302400    149
      ZHAT(1,I) = BETA                                                  EP302600
      YHAT(1,I) = Y1C                                                   EP302800
      GO TO 14000                                                       EP303000
10400 ZHAT(1,I+1) = Z2C                                                 EP303200
      YHAT(1,I+1) = Y1C                                                 EP303400
      S = 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 = 49                                                          EP304200
      CHAT(1,I+1) = 4                                                   EP304400
      CALL  ZYCUT( 2, Z2C-S, I4, BETA, $10800, I+1, 2 )                 EP304600    171
      GO TO 8900                                                        EP304800
C
10800 WRITE ( 6, 32020 )  JCD                                           EP305000    174
32020 FORMAT( 1H0, 10HERROR CODE, I3, 12H USING ZYCUT )                 EP305200
      CALL  GOOF                                                        EP305400    175
11000 JCD = 52                                                          EP306000
      CHAT(1,I+1) = 5                                                   EP306200
      CALL  ZYCUT( 1, Y1C, I4, BETA, $10800, I+1, 2 )                   EP306400    179
      GO TO 9200                                                        EP306600
11500 IF( ISURFC(1,I2) .EQ. 3 )  GO TO 11800                            EP307000
      IF( SURFC(11,I2) .GT. 0. ) GO TO 12000                            EP307200
11800 ZHAT(1,I+1) = Z2C                                                 EP307400
      YHAT(1,I+1) = Y1C                                                 EP307600
      S = VTXCRD(I4) - VTXCRD(I3)                                       EP307800
      IF( I3 .EQ. I4 )  GO TO 14000                                     EP308000
      GO TO 8800                                                        EP308200
12000 IF( ISURFC(1,I3) .EQ. 3 )  GO TO 12200                            EP401000
      IF( SURFC(11,I3) .LT. 0. ) GO TO 12500                            EP401200
12200 ZHAT(2,I) = Z1C                                                   EP401400
      YHAT(2,I) = Y1C                                                   EP401600
      S = VTXCRD(I2) - VTXCRD(I1)                                       EP401800
      IF( I1 .EQ. I2 )  GO TO 14000                                     EP402000
      GO TO 9800                                                        EP402200
12500 ZHAT(2,I) = Z1C                                                   EP402400
      YHAT(2,I) = Y1C                                                   EP402600
      ZHAT(1,I+1) = Z2C                                                 EP402800
      YHAT(1,I+1) = Y1C                                                 EP403000
      CHAT(2,I) = 4                                                     EP403200
      CHAT(1,I+1) = 4                                                   EP403400
      IF( I1 .EQ. I2 )  GO TO 13000                                     EP403600
      S = VTXCRD(I2) - VTXCRD(I1)                                       EP403800
      CALL  ZYCUT( 2, Z1C+S, I1, BETA, $13000, I, 1 )                   EP404000    228
      ZHAT(1,I) = Z1C + S                                               EP404200
      YHAT(1,I) = BETA                                                  EP404400
```

```
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   238
      ZHA (2,I+1) = Z2C - S                                           EP405600
      YHA  2,I+1) = BETA                                              EP405800
      CHAT  ),I+1) = 1                                                EP406000
C
14000 S = \ 'XCRD(I2) - VTXCRD(I1)                                    EP406200
      IF( 1   .NE. I2 )  GO TO 14400                                  EP406400
      ARCS( , 2,I) = ZHAT(1,I)                                        EP406600
      ARCS.  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                                   EP506000
      IF( CHAT(1,I) .EQ. 5 )  GO TO 16500                             EP506200
      ARCS(1,2,I) = Z1                                                EP506400
      ARCS(2,2,I) = Y1                                                EP506600
      ARCS(1,3,I) = Z1                                                EP506800
      ARCS(2,3,I) = Y2                                                EP507000
      ARCS(1,4,I) = Z2                                                EP507200
      ARCS(2,4,I) = Y2                                                EP507400
      ARCS(1,5,I) = S                                                 EP507600
      NARCS(I) = 5                                                    EP508000
      GO TO 18000                                                     EP508200
16500 JCD = 92                                                        EP508400
      CALL  ZYCUT( 1, Y1, I2, BETA, $10800, I, 2 )                    EP508600   310
      ARCS(1,2,I) = Z1                                                EP508800
      ARCS(2,2,I) = Y1                                                EP509000
      ARCS(1,3,I) = BETA + S                                          EP509100
      ARCS(2,3,I) = Y1                                                EP509200
```

```
      ARCS(1,4,I) = 8                                     EP509400
      NARCS(I) = 4                                        EP509600
      GO TO 18000                                         EP509800
17000 IF( CHAT(2,I) .EQ. 5 )  GO TO 17500                 EP601000
      ARCS(1,2,I) = Z1                                    EP601200
      ARCS(2,2,I) = Y1                                    EP601400
      ARCS(1,3,I) = Z2                                    EP601600
      ARCS(2,3,I) = Y1                                    EP601800
      ARCS(1,4,I) = Z2                                    EP602000
      ARCS(2,4,I) = Y2                                    EP602200
      ARCS(1,5,I) = 8                                     EP602400
      NARCS(I) = 5                                        EP602600
      GO TO 18000                                         EP602800
17500 JCD = 96                                            EP603000
      CALL  ZYCUT( 1, Y2, Z1, BETA, $10800, I, 1 )        EP603200    335
      ARCS(1,2,I) = BETA                                  EP603400
      ARCS(2,2,I) = Y2                                    EP603600
      ARCS(1,3,I) = Z2                                    EP603800
      ARCS(2,3,I) = Y2                                    EP604000
      ARCS(1,4,I) = 8                                     EP604200
      NARCS(I) = 4                                        EP604400
18000 CONTINUE                                            EP605000
C
C             END OF ENDPTS SUBROUTINE                    EP606000
32000 RETURN                                              EP606500
      END                                                 EP607000
```

ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME 02/16/68 PAGE 7
ENDPTZ STORAGE MAP

SUBROUTINE ENDPTS
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	00013	R
DATE	00027	I	PUNCID	00032	R	DATA	00033	R
LNODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	INODE	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	00080	R	OMGA2	00081	R	OMGA1	00082	R
DELD	00083	R	EPRAD	00084	R	PSCAL	00085	R
OMGAF	00086	R	SPFEA	00087	R	DUMIN	00070	R
CDIST	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
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	08210	R
IBNDS	08210	I	NCOND	08665	I			

COMMON BLOCK ELMAIR ORIGIN 08667 LENGTH 00766

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NELMT	00000	I	NAIRS	00001	I	ELMMTX	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	00320	I

UNDIMENSIONEC PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
K	11005	I	I	11006	I	I1	11007	I
I2	11010	I	S	11011	R	BETA	11012	R
I3	11013	I	I4	11014	I	Z1M	11015	R
Z1X	11016	R	Z2M	11017	R	Z2X	11020	R
Z1C	11021	R	Y1C	11022	R	Z2C	11023	R
I2C	11024	I	I3C	11025	I	IGO	11026	I
JCD	11027	I	Z1	11030	R	Y1	11031	R
Z2	11032	R	Y2	11033	R			

ENTRY POINTS

ENDPTS SECTION 9

```
ROUTINE TO GENERATE THE END POINTS OF ARCS FOR EACH ELEME            08/16/68                PAGE  8
   ENDPTZ                                          STORAGE  MAP

                                              SUBROUTINES CALLED

ZYCUT      SECTION    10              .FWRD.     SECTION    11               GOOF       SECTION    12
CLOSS      SECTION    13              .FXEM.     SECTION    14               .UNDS.     SECTION    13
.FFIL.     SECTION    16              .FCNV.     SECTION    17               SYSLOC     SECTION    18
                                      EFN   IFN   CORRESPONDENCE
```

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
2001	13A	11125	1000	9A	11117	7000	51A	11317
3001	24A	11150	32000	347A	12566	4000	33A	11177
5000	37A	11221	5500	44A	11258	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	155A	11720	11500	182A	12042
8800	108A	11528	9000	114A	11560	8900	109A	11550
9200	118A	11577	9800	139A	11652	10000	147A	11677
10600	168A	11760	11000	176A	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	282A	12373
16000	292A	12421	17000	320A	12505	16500	308A	12453
17500	333A	12532						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12760.

```
$0
$TITLE          ROUTINE TO CHECK FOR INTERSECTIONS
$IBFTC E1RGCZ  M94,XR7            LINK 2   (E1RGCD)              E1001000
```

```
         ROUTINE TO CHECK FOR INTERSECTIONS                                          02/20/68                    PAGE  1
           E1RGCZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  E1RGCD                                                     E1001200
C                                                                            
C     *** INPUTS ***                                                         E1001500
C ELMNTX 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/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
C             START OF E1RGCD SUBROUTINE                                     E1003000
      DO  2000  I = 1, NELMT                                                 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
     1 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
```

```
 4000 WRITE ( 6, 32100 )  Z1, Y1, I1, I2                         E1201800   42
32100 FORMAT( 1H0, 27HTHE POINT OF INTERSECTION (, E14.7, 1H, E14.7,   E1202000
     1 10H) SURFACES, I3, 4H AND, I3, 13H IS NOT VALID )          E1202200
      CALL GOOF                                                   E1204000
C                                                                 
C                END OF E1RGCD SUBROUTINE                         E1204000   43
 6000 RETURN                                                      E1204500
      END                                                         E1205000
```

```
ROUTINE TO CHECK FOR INTERSECTIONS                                   08/16/68                    PAGE  3
   E1RGCD                                   STORAGE  MAP

                                      SUBROUTINE  E1RGCD
                                          COMMON   VARIABLES

                COMMON BLOCK      ELMAIR            ORIGIN       00001            LENGTH      00766

SYMBOL      LOCATION     TYPE              SYMBOL    LOCATION     TYPE            SYMBOL     LOCATION     TYPE
NELMT        00000        I                NAIRS      00001        I              ELMNTX      00002        I
AIRMTX       00312        I                VTXCRD     00522        R

                COMMON BLOCK      COORNS            ORIGIN       00767            LENGTH      01130
XHAT         00000        R                YHAT       00510        R              CHAT        00820        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                ZI         02126        R              YI          02127        R
Z1PRM        02130        R
                                            ENTRY POINTS

     E1RGCD        SECTION      7
                                            SUBROUTINES CALLED

     CROSS         SECTION      8           .FWRD.      SECTION     9            GOOF        SECTION     10
     .UNDS.        SECTION     11           .FFIL.      SECTION    12            .FCNV.      SECTION     13
     SYSLOC        SECTION     14
                                        EFN   IFN   CORRESPONDENCE

EFN         IFN         LOCATION           EFN         IFN        LOCATION        EFN         IFN        LOCATION
2000         38A         02351             4000         42A         02355          6000         44A         02376
32100       FORMAT       02143
 THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 02417.
```

```
$*
$TITLE          ROUTINE TO GENERATE BINARY POINT FILE
$IBFTC PFILE2  N94,XR7              LINK 2    (PFILE)                    PL001600
```

```
     ROUTINE TO GENERATE BINARY POINT FILE                          02/16/68                  PAGE   1
        PFILEZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  PFILE( ARCS, NARCS, ZMIN, ZMAX )                     PL001900
C                                                                      PL002000
C     *** INPUTS ***                                                   PL002100
C NPTS     NUMBER OF POINTS PER CURVED ARC                             PL002200
C ARCS     3D MATRIX WHERE ITH LAYER CONTAINS END POINTS OF THE SEVERAL PL0024~~
C          ARCS WHICH COMPRISE ELEMENT I.                              PL002600
C NARCS    VECTOR WHOSE ITH COMPONENT SPECIFIES THE NUMBER OF POINTS IN PL002800
C          LAYER I OF MATRIX ARCS.                                     PL003000
C YMAXX    MAXIMUM Y ALLOWED (TUBE BOUNDARY).                          PL003200
C                                                                      PL003400
C     *** OUTPUTS ***                                                  PL003500
C ZMIN     MINIMUM Z RELATIVE TO ENTRANCE PUPIL.                       PL003800
C ZMAX     MAXIMUM Z RELATIVE TO ENTRANCE PUPIL.                       PL004000
C                                                                      PL004200
C                                                                      PL004400
C     *** DESCRIPTION ***                                              PL004500
C PFILE CREATES A BINARY FILE (LOGICAL 11) OF NELMT+4 RECORDS EACH ONE PL004800
C OF WHICH IS 601 WORDS LONG AND CONTAINING ...                        PL005000
C                                                                      PL005200
C                         N,ZLIST,YLIST                                PL005400
C                                                                      PL005600
C   N=NO. OF POINTS                                                    PL005800
C   ZLIST=VECTOR CONTAINING ZI  I=1,2,....,N                           PL006000
C   YLIST=VECTOR CONTAINING YI  I=1,2,....,N                           PL006200
C                                                                      PL006400
C THE POINTS (ZI,YI)  I=1,2,....,N ARE CONSECUTIVE POINTS ALONG THE    PL006600
C UPPER HALF OF A CURVE (YI.GE.0.) DEFINED BY Y=F(Z).  THIS CURVE WILL PL006800
C BE PLOTTED.                                                          PL007000
C                                                                      PL007200
C RECORDS 1 THRU NELMT CONTAIN DATA FOR THE NELMT ELEMENTS.            PL007400
C RECORD NELMT+1 CONTAINS DATA FOR THE IMAGE PLANE                     PL007600
C RECORD NELMT+2 CONTAINS DATA FOR THE ENTRANCE PUPIL                  PL007800
C RECORD NELMT+3 CONTAINS DATA FOR RAY 1  (H,TH)=(HMAX,0.) (X0,Y0)=(0,-1)PL008000
C RECORD NELMT+4 CONTAINS DATA FOR RAY 2  (H,TH)=(HMAX,0.) (X0,Y0)=(0,-1)PL008200
C                                                                      PL008400
C                                                                      PL008600
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), 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 NOSGV(1) ), ( DATA(53), WOBJH(1) ), (DATA(60), WCLRH(1) ),        DT102300
      E ( DATA(67), WINHT(1) ), (DATA(74), EINHT(1) ), ( DATA(81),         DT102400
      F CINPL(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
       RE    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)           CD001000
       INTEGER  CHAT                                                       CD002000
C
       COMMON  / PLOTC /  YMAXX, DZMIN, NPTS                               PC001000
C
       COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                     AZ004000
C                                                                          PL009000
       DIMENSION ARCS(2,5,100),NARCS(100),XVCT(3),QVCT(3),ZLIST(300),     PL101000
      1 YLIST(300)                                                         PL101200
       EQUIVALENCE  (ZLIST(1),ZHAT(1,1)),(YLIST(1),YHAT(1,51))             PL102000
       REWIND 11                                                           PL103000    1
       ZMIN=0.                                                             PL103200
       YMAX=0.                                                             PL103400
C                                                                          PL104000
C LOOP TO GENERATE POINTS ON EACH ELEMENT ...                              PL104200
C                                                                          PL104400
       DO 410 I=1,NELMT                                                    PL104600
       DO 30 J=1,300                                                       PL104800
       ZLIST(J)=0.                                                         PL105000
 30    YLIST(J)=0.                                                         PL105200
       IGO=NARCS(I)-1                                                      PL105400
       Z1=ARCS(1,1,I)                                                      PL105600
       Y1=ARCS(2,1,I)                                                      PL105800
       Z2=ARCS(1,2,I)                                                      PL106000
       Y2=ARCS(2,2,I)                                                      PL106200
       Z3=ARCS(1,3,I)                                                      PL106400
       Y3=ARCS(2,3,I)                                                      PL106600
       Z4=ARCS(1,4,I)                                                      PL106800
       Y4=ARCS(2,4,I)                                                      PL107000
       Z5=ARCS(1,5,I)                                                      PL107200
       Y5=ARCS(2,5,I)                                                      PL107400
       I1=ELMNTX(1,I)                                                      PL107600
       I2=ELMNTX(2,I)                                                      PL107800
       S=VTXCRD(I2)-VTXCRD(I1)                                             PL108000
       GO TO (50,80,210,260),IGO                                           PL108200
C                                                                          PL201000
C  1 ARC ...                                                               PL201200
 50    IF(Z1.NE.Z2)GO TO 70                                                PL201400
       ZLIST(1)=Z1                                                         PL201600
       ZLIST(2)=Z1                                                         PL201800
       YLIST(1)=Y1                                                         PL202000
       YLIST(2)=Y2                                                         PL202200
       JCNT=2                                                              PL202400
```

```
      ROUTINE TO GENERATE BINARY POINT FILE                                02/16/68                PAGE  3
        PFILE2      - EFN   SOURCE STATEMENT  -  IFN(S)  -

         GO TO 310                                                   PL202600
 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
 280     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
```

```
      ZLIST(JCNT)=Z3                                            PL307200
      YLIST(JCNT)=Y3                                            PL307400
      Z2=Z4                                                     PL307600
      Y2=Y4                                                     PL307800
      Z3=Z5                                                     PL308000
      Y3=Y5                                                     PL308200
      GO TO 120                                                 PL308400
C                                                               PL401000
C TRANSLATE DATA TO CENTER OF ENTRANCE PUPIL ...                PL401200
C                                                               PL401400
310   S=VTXCRD(I1)                                              PL401600
      DO 350 J=1,JCNT                                           PL401800
350   ZLIST(J)=ZLIST(J)+S                                       PL402000
C                                                               PL402200
C WRITE DATA AS NEXT LOGICAL RECORD ON 11 ...                    PL402400
C                                                               PL402600
      WRITE (11) JCNT,ZLIST,YLIST                               PL402800
C                                                               PL403000
C GET CURRENT MAX AND MIN ...                                    PL403200    98
      DO 400 J=1,JCNT                                           PL403400
      ZMIN=AMIN1(ZMIN,ZLIST(J))                                 PL403600
400   ZMAX=AMAX1(ZMAX,ZLIST(J))                                 PL403800
410   CONTINUE                                                  PL404000
C                                                               PL404200
C PUT IMAGE PLANE DATA ON 11 ...                                 PL404400
C                                                               PL404600
      IF(ELMMTX(2,NELMT).NE.NSURF+1)GO TO 450                   PL404800
      JCNT=0                                                    PL405000
      DO 440 J=1,300                                            PL405200
      ZLIST(J)=0.                                               PL405400
440   YLIST(J)=0.                                               PL405600
      GO TO 460                                                 PL405800
450   ZLIST(1)=VTXCRD(NSURF+1)                                  PL406000
      YLIST(1)=0.                                               PL406200
      ZLIST(2)=ZLIST(1)                                         PL406400
      YLIST(2)=YMAXX                                            PL406600
      ZMIN=AMIN1(ZMIN,ZLIST(1))                                 PL406800
      ZMAX=AMAX1(ZMAX,ZLIST(1))                                 PL407000
      JCNT=2                                                    PL407200
460   WRITE (11) JCNT,ZLIST,YLIST                               PL407400
C                                                               PL407600
C PUT ENTRANCE PUPIL DATA ON TAPE ...                            PL407800
C                                                               PL408000    124
      JCNT=2                                                    PL408200
      ZLIST(1)=0.                                               PL408400
      ZLIST(2)=0.                                               PL408600
      YLIST(1)=EPRAD                                            PL408800
      YLIST(2)=YMAXX                                            PL409000
      WRITE (11) JCNT,ZLIST,YLIST                               PL409200
C                                                               PL501000
C TRACE 2 SPECIAL RAYS ...                                       PL501200
C                                                               PL501400    128
      HMAX=HO+FLOAT(NOBJH-1)*DELH                               PL501600
      THTR=0.                                                   PL501800
      SNTHTR=0.                                                 PL502000
      CSTHTR=1.                                                 PL502200
```

```
  ROUTINE TO GENERATE BINARY POINT FILE                                    02/16/68         PAGE  5
     PFILE2      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

       ASSIGN 680 TO IGO                                                   PL502300
       CALL RAYTR(0.,-1.,HMAX,1,XVCT,QVCT,SMISS,$520,$520,$510,1,ZLIST,    PL502400   100000
      1 YLIST)                                                             PL502600     132
 510   GO TO 530                                                           PL502800
 520   N1 = 0                                                              PL503000
       GO TO 675                                                           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                                                  PL506050
 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
 675   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
```

STORAGE MAP

SUBROUTINE PFILE
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	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	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	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	00071	R	HD	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
HOTAN	00077	R	NDSGN	00100	I	NDSGV	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	08665	I			
	COMMON BLOCK		ELMAIR	ORIGIN	08667	LENGTH	00766	
NELMT	00000	I	NAIRS	00001	I	ELMMTX	00002	I
AIRMTX	00312	I	VTXCRD	00622	R			
	COMMON BLOCK		CODRNG	ORIGIN	07655	LENGTH	01130	
ZHAT	00000	R	YHAT	00310	R	CHAT	00620	I
ZLIST	00000	R	YLIST	00454	R			
	COMMON BLOCK		PLOTC	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	QVCT	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 02/16/68 PAGE 7

PFILEZ

STORAGE MAP

Y2	11027	R	Z3	11030	R	Y3	11031	R
Z4	11032	R	Y4	11033	R	Z5	11034	R
Y5	11035	R	I1	11036	I	I2	11037	I
S	11040	R	JCNT	11041	I	K	11042	I
NMAX	11043	R	SMISS	11044	R	N1	11045	I

ENTRY POINTS

PFILE SECTION 13

SUBROUTINES CALLED

.FRWT.	SECTION	14	ARPTS	SECTION	15	.FWRB.	SECTION	16
.FBLO.	SECTION	17	RAYTR	SECTION	18	.FEFT.	SECTION	19
.FXEM.	SECTION	20	.UN11.	SECTION	21	.FWLR.	SECTION	22
.FBLT.	SECTION	23	.FBDT.	SECTION	24	E.1	SECTION	25
E.2	SECTION	26	E.3	SECTION	27	E.4	SECTION	28
CC.1	SECTION	29	CC.2	SECTION	30	CC.3	SECTION	31
CC.4	SECTION	32	SYSLOC	SECTION	33			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
410	108A	11575	30	8A	11103	50	30A	11165
80	38A	11222	210	66A	11362	260	70A	11427
70	34A	11204	310	89A	11511	110	43A	11235
120	46A	11253	140	54A	11306	190	61A	11345
240	71A	11401	250	74A	11416	290	81A	11446
300	84A	11463	350	93A	11522	400	102A	11564
450	122A	11620	440	117A	11614	460	124A	11645
680	189A	12163	520	135A	11767	510	134A	11766
530	137A	11771	675	184A	12141	540	138A	11774
570	147A	12023	600	158A	12056	650	166A	12101
670	174A	12114	710	194A	12212	700	193A	12211

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12323.

```
$*
$TITLE          CONTROL ROUTINE OF PROFILE PL  LINK
$IBFTC PROFLZ  M94.XR7              LINK 2    (PROFIL)                PF001000
```

```
     CONTROL ROUTINE OF PROFILE PLOT LINK                                  02/16/68                PAGE  1
        PROFLZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

       SUBROUTINE  PROFIL                                           PF001500
C                                                                 
C     *** INPUTS ***                                                PF002000
C YMAXX  MAXIMUM ALLOWABLE Y VALUE FROM OPTIC AXIS.         (PLOTC) PF002100
C DZMIN  MINIMUM ALLOWABLE Z SURFACE SEPARATION.            (PLOTC) PF002200
C NPTS   NUMBER OF POINTS TO GENERATE ON EACH SURFACE.      (PLOTC) PF002300
C
C     *** OUTPUTS ***                                               PF002400
C        CROSS SECTION PLOT FRAME AND PLOT PARAMETERS ON FOLLOWING FRAMEPF002500
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/NELMT,NAIRS,ELMMTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
      INTEGER  ELMMTX, AIRMTX                                       EL002000
C
      COMMON  / COORNS /  ZHAT(2,100), YHAT(2,100), CHAT(2,100)     CD001000
      INTEGER  CHAT                                                 CD002000
C
      COMMON  / PLOTC /  YMAXX, DZMIN, NPTS                         PC001000
C
      COMMON / PRNT / LINE, PAGE                                    PT001000
      INTEGER  PAGE                                                 PT002000
C
      DIMENSION  ARCS(2,3,100), NARCS(100)                          PF004000
C
C GET ELEMENTS AND AIR SPACES ...                                   PF005000
      CALL CELAIR( $10000 )                                         PF005200
C
C GET RANGES FOR ALL ELEMENTS ...                                   PF005400      2
      CALL  RANGE( YMAXX )                                          PF005600
C
C GET INTERSECTIONS ...                                             PF006200      5
      CALL EIRGCD                                                   PF006500
C
C              GET ARC LIST EACH ELEMENT ...                        PF008000      7
      CALL  ENDPTS( DZMIN, ARCS, NARCS )                            PF008400
C
C              SECTION TO PRINT OUT LENS SYSTEM                     PF151000      9
      CALL PRTCTL                                                   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 REFPF152200
     2ERENCED FROM OPTIC AXIS )                                     PF152400
      LINE = LINE + 4                                               PF152600
      DO  6000  I = 1, NELMT                                        PF153000
      WRITE ( 6, 32020 )  I, ELMMTX(1,I), ELMMTX(2,I)               PF153200     16
32020 FORMAT( 1H0, 5X, 7HELEMENT, I3, 3X, 13H( SURFACE 1 =, I3, 3X,  PF153400
     1 11HSURFACE 2 =, I3, 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                                  08/16/68                    PAGE 2
          PROFL2       - EFN   SOURCE STATEMENT  -  IFN(S)  -

 5000 CONTINUE                                                     PF194600
32030 FORMAT( 1H , 9X, 8E16.7 )                                    PF194800
      LINE = LINE + NP + 4                                         PF195000
      IF( LINE .GT. 48 )  CALL  PRTCTL                             PF195200    29
 6000 CONTINUE                                                     PF195400
C
C     GENERATE PLOT POINTS ...                                     PF201000
      CALL  PFILE( ARCS, NARCS, ZMIN, ZMAX )                       PF201500
C
C     PLOT LENS SYSTEM ...                                         PF202000    34
      CALL  CPLOT( ZMIN, ZMAX )                                    PF202500
C                                                                              36
10000 RETURN                                                       PF209000
      END                                                          PF209400
```

```
CONTROL ROUTINE OF PROFILE PLOT LINK                                      08/16/68                     PAGE  3
   PROFLZ                                          STORAGE  MAP

                                         SUBROUTINE  PROFIL
                                             COMMON   VARIABLES

                  COMMON BLOCK      ELMAIR          ORIGIN      00001               LENGTH      [illegible]

SYMBOL     LOCATION     TYPE            SYMBOL     LOCATION     TYPE            SYMBOL     LOCATION     TYPE
NELMT       00000        I              NAIRS       00001        I              ELMMTX      00002        I
AIRMTX      00312        I              VTXCRD      00822        R

                  COMMON BLOCK      COJ[illegible]  ORIGIN      00767               LENGTH      01130
ZHAT        00000        R              YHAT        00310        R              CHAT        00820        I

                  COMMON BLOCK      PLOTC           ORIGIN      02117               LENGTH      00005
YMAXX       00000        R              DZMIN       00001        R              NPTS        00002        I

                  COMMON BLOCK      P[illegible]    ORIGIN      02122               LENGTH      00002
LINE        00000        I              PAGE        00001        I
                                             DIMENSIONED PROGRAM VARIABLES

SYMBOL     LOCATION     TYPE            SYMBOL     LOCATION     TYPE            SYMBOL     LOCATION     TYPE
ARCS        02124        R              NARCS       04074        I
                                           UNDIMENSIONED PROGRAM VARIABLES

SYMBOL     LOCATION     TYPE            SYMBOL     LOCATION     TYPE            SYMBOL     LOCATION     TYPE
I           04240        I              NP          04241        I              ZMIN        04242        R
ZMAX        04243        R
                                                ENTRY POINTS

     PROFIL       SECTION      11
                                                SUBROUTINES CALLED

     CELAIR       SECTION      12              RANGE       SECTION      13              E1R6CD      SECTION      14
     ENDPTS       SECTION      15              PRTCTL      SECTION      16              .FWRD.      SECTION      17
     PFILE        SECTION      18              CPLOT       SECTION      19              .UNDS.      SECTION      20
     .PFIL.       SECTION      21              .FCNV.      SECTION      22              SYSLOC      SECTION      23
                                               EFN    IFN    CORRESPONDENCE

EFN         IFN         LOCATION            EFN         IFN         LOCATION            EFN         IFN         LOCATION
10000       37A          04504              32010       FORMAT       04256              6000        31A          04465
32020       FORMAT       04310              5000        25A          04447              32030       FORMAT       04333

 THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 04532.
```

```
$0
$TITLE          ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES
$IBFTC RANGEZ  M94,XR7              LINK 2   (RANGE)              RG001000
```

```
      SUBROUTINE  RANGE( YMAXIM )                                      RG001200
C                                                                      
C     *** INPUTS ***                                                   RG001500
C YMAXIM MAXIMUM ALLOWABLE Y FROM OPTIC AXIS.                          RG001600
C ELMMTX 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,  RG002750
C        AND THE RANGE IN Y, YHAT,FOR EVERY SURFACE.  THE Z VALUE IS   RG002800
C        REFERENCED FROM THE VERTEX OF THAT SURFACE.  THE CODE MATRIX  RG002850
C        VALUE = 2 WHEN A SURFACE EXTENDS BEYOND YMAXIM, OTHERWISE = 1. RG002900
C                                                                      
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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 PSC/L ), ( 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), 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), NCONT )                             DT102900
      INTEGER  CONTRL, AFLAG, DATE                                     DT201000
      REAL  LATTC                                                      DT202000
C                                                                      
      COMMON/ELMAIR/NELMT,NAIRS,ELMMTX(2,100),AIRMTX(2,100),VTXCRD(100) EL001000
      INTEGER  ELMMTX, AIRMTX                                          EL002000
C                                                                      
      COMMON  / CODRNG /  ZHAT(2,100), YHAT(2,100), CHAT(2,100)        CD001000
      INTEGER  CHAT                                                    CD002000
```

```
       RANGEZ     - EFN   SOURCE STATEMENT  -  IFN(S)  -

C
      DIMENSION  ZRANGE(2), YRANGE(2)                             RG002990
C
      INTEGER  SURFNO                                             RG003000
C
C             START OF RANGE SUBROUTINE                           RG003100
C
      DO  500 I = 1, 2                                            RG003200
      DO  500  J = 1, 100                                         RG003400
      ZHAT(I,J) = 0.0                                             RG003600
      YHAT(I,J) = 0.0                                             RG003800
  500 CHAT(I,J) = 0                                               RG004000
      DO 20000  I = 1, NELMT                                      RG004200
      DO 20000  J = 1, 2                                          RG004400
C
C             THIS SECTION DETERMINES THE RANGE IN Z OF THE SURFACE  RG004600
C             GIVEN IN SURFNO. THE RANGE IN Y, YRANGE, CORRESPONDING RG004700
C             TO ZRANGE IS ALSO COMPUTED                          RG004800
      SURFNO = ELMMTX(J,I)                                        RG004900
      ZRANGE(1) = 0.                                              RG005000
      ZRANGE(2) = 0.                                              RG005200
      YRANGE(1) = 0.                                              RG005400
      YRANGE(2) = YMAXIM                                          RG005600
      ITYPE = ISURFC(1,SURFNO)                                    RG005800
      R = SURFC(11,SURFNO)                                        RG006000
      B = SURFC(12,SURFNO)                                        RG006200
      IF( ITYPE .EQ. 3 )  GO TO 12000                             RG006400
      IF( R .EQ. 0. )  GO TO 12000                                RG006600
      IF( B .NE. 0. )  GO TO 4000                                 RG006800
      IF( R .LT. 0. )  GO TO 2000                                 RG007000
C
C             CONVEX CIRCLE                                       RG008000
      IF( YMAXIM .LT. R )  GO TO 1000                             RG008200
      ZRANGE(2) = R                                               RG008400
      YRANGE(2) = R                                               RG008600
      GO TO 12000                                                 RG008800
 1000 ZRANGE(2) = R - SQRT( R*R - YMAXIM*YMAXIM )                 RG009000    42
      GO TO 12000                                                 RG009200
C
C             CONCAVE CIRCLE                                      RG101000
 2000 IF( YMAXIM .LT. ABS( R ) )  GO TO 3000                      RG101200
      ZRANGE(1) = R                                               RG101400
      YRANGE(2) = ABS( R )                                        RG101600
      GO TO 12000                                                 RG101800
 3000 ZRANGE(1) = R + SQRT( R*R - YMAXIM*YMAXIM )                 RG102000    49
      GO TO 12000                                                 RG102200
 4000 IF( B .NE. 1. )  GO TO 5000                                 RG102400
C
C             PARABOLA                                            RG102600
4002  ZT=YMAXIM*YMAXIM/(2.*R)                                     RG102800
      K = 1                                                       RG103000
      IF( R .GT. 0. )  K = 2                                      RG103200
      ZRANGE(K) = ZT                                              RG103400
      GO TO 12000                                                 RG103600
 5000 IF( B .GT. 1. )  GO TO 9000                                 RG104000
C
```

```
       ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES                    08/16/68                PAGE  3
          RANGEZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

C               ELLIPSE                                                 RG104200
      A1 = ABS( R ) / SQRT( 1. - B )                                    RG104400    63
      A2 = R / ( 1.-B )                                                 RG104600
      IF( R .LT. 0. )  GO TO 7000                                       RG104800
C
C               CONVEX ELLIPSE                                          RG105000
      IF( YMAXIM .LT. A1 )  GO TO 6000                                  RG105200
      ZRANGE(2) = A2                                                    RG105400
      YRANGE(2) = A1                                                    RG105600
      GO TO 12000                                                       RG105800
 6000 ZRANGE(2) = A2 - SQRT((A1 * A1 - YMAXIM * YMAXIM ) / ( 1.-B ) )   RG106000    72
      GO TO 12000                                                       RG106200
C
C               CONCAVE ELLIPSE                                         RG106400
 7000 IF( YMAXIM .LT. A1 )  GO TO 8000                                  RG106600
      ZRANGE(1) = A2                                                    RG106800
      YRANGE(2) = A1                                                    RG107000
      GO TO 12000                                                       RG107200
 8000 ZRANGE(1) = A2 + SQRT( ( A1*A1 - YMAXIM*YMAXIM ) / ( 1.-B ) )     RG107400    79
      GO TO 12000                                                       RG107600
9000  IF((B-1.).LT.1.E-5)GO TO 4002                                     RG107800
      A2=-R/(B-1.)                                                      RG107900
      A1 = ABS( R ) / SQRT( B-1. )                                      RG108000    84
      DELTA = SQRT( ( A1*A1 + YMAXIM*YMAXIM ) / ( B-1. ) )              RG108200    85
      IF( R .GT. 0. )  GO TO 10000                                      RG201000
C
C               CONCAVE HYPERBOLA                                       RG201200
      ZRANGE(1) = A2 - DELTA                                            RG201400
      GO TO 12000                                                       RG201600
C
C               CONVEX HYPERBOLA                                        RG202000
10000 ZRANGE(2) = A2 + DELTA                                            RG202200
12000 YHAT(J,I) = YRANGE(2)                                             RG205600
      CHAT(J,I) = 1                                                     RG205800
      IF( YHAT(J,I) .EQ. YMAXIM )  CHAT(J,I) = 2                        RG206000
      ZHAT(J,I) = ZRANGE(1)                                             RG206200
      IF( ZRANGE(1) .EQ. 0. )  ZHAT(J,I) = ZRANGE(2)                    RG206400
20000 CONTINUE                                                          RG206600
C
C               END OF RANGE SUBROUTINE                                 RG209000
      RETURN                                                            RG209200
      END                                                               RG209400
```

ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES 02/16/68 PAGE 4
RANGEZ STORAGE MAP

SUBROUTINE RANGE
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	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	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	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	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
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	08210	R
IBNDS	08210	I	NCOND	08665	I			

COMMON BLOCK ELMAIR ORIGIN 08667 LENGTH 00766

NELMT	00000	I	NAIRS	00001	I	ELMMTX	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

RANGE SECTION 9

```
ROUTINE TO DETERMINE UNCONSTRAINED RANGE MATRICES                02/16/68                PAGE 8
   RANGE2                                      STORAGE  MAP

                                               SUBROUTINES CALLED

     SQRT          SECTION      10                SYSLOC      SECTION     11
                                                  EFN   IFN   CORRESPONDENCE
```

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
500	9A	11081	20000	102A	11550	12000	91A	1530
4000	51A	11252	2000	44A	11220	1000	41A	11201
3000	48A	11234	5000	60A	11303	4002	54A	11256
8000	81A	11436	7000	74A	11400	6000	71A	11353
8000	78A	11412	10000	90A	11525			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 11623.

```
$*
$TITLE          ROUTINE TO FIND INTERSECTION OF CONIC AND ST. LINE
$IBFTC ZYCUTZ  M94,XR7              LINK 2    (ZYCUT)                    ZY001000
```

```
      SUBROUTINE  ZYCUT( VHCODE, ALFA, IS, BETA, *, ELNO, SURNO )      ZY001500
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 YHAT 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 INTERSECTIONZY003900
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), NRAYS ), ( DATA(3),   DT101000
     1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),     DT101100
     2 NSUBT ), ( DATA(7), NSURF ), ( 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), 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), 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
      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
```

```
      BETA = 0.                                                          ZY004200
      IT = ISURFC(1,IS)                                                  ZY004400
      R = SURFC(11,IS)                                                   ZY004600
      B = SURFC(12,IS)                                                   ZY004800
      IF( IT .EQ. 3 )  GO TO 2000                                        ZY005000
      IF( R .NE. 0. )  GO TO 4000                                        ZY005200
 2000 IF( VHCODE .EQ. 2 )  RETURN 1                                      ZY005400
      IF( ALFA .LT. 0.  .OR. ALFA .GT. YHAT(SURNO,ELNO) )  RETURN 1      ZY006000
      GO TO 32000                                                        ZY006200
 4000 IF( VHCODE .NE. 1 )  GO TO 6000                                    ZY006400
      IF( ALFA .LT. 0.  .OR.  ALFA .GT. YHAT(SURNO,ELNO) )  RETURN 1     ZY006600
      BETA = ALFA*ALFA / ( R+SIGN( SQRT(R*R - (1.-B)*ALFA*ALFA), R ))   ZY006800    28
      GO TO 32000                                                        ZY007000
 6000 ZMIN = AMIN1( 0., ZHAT(SURNO,ELNO) )                               ZY008000
      ZMAX = AMAX1( 0., ZHAT(SURNO,ELNO) )                               ZY008200
      IF( ALFA .LT. ZMIN  .OR.  ALFA .GT. ZMAX )  RETURN 1               ZY008400
      BETA = SQRT( (B-1.)*ALFA*ALFA + 2.*R*ALFA )                        ZY008600
C
C                END OF ZYCUT SUBROUTINE                                 ZY009000    32
32000 RETURN                                                             ZY009200
      END                                                                ZY009400
```

ROUTINE TO FIND INTERSECTION OF CONIC AND ST. LINE 02/18/68 PAGE 3
ZYCUT2 STORAGE MAP

SUBROUTINE ZYCUT
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	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	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
DLPLH	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
CDIST	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
WOBJH	00117	R	WCLRH	00126	R	WINHT	00135	R
EINHT	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 CODING ORIGIN 08667 LENGTH 01130

ZHAT	00000	R	YHAT	00310	R	CHAT	00620	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

SQRT SECTION 8 SYSLOC SECTION 9 .FRET. SECTION 10

EFN IFN CORRESPONDENCE

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

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 10330.

Section 1

LINK 3

```
$*
$TITLE         SUBROUTINE TO COMPUTE TWINRAY 4 DIAGNOSTIC INFORMATION
$IBFTC TWNR4Z  M94,XR7              LINK 3   (TWNRY4)                    TR401000
```

```
      SUBROUTINE  TWNRY4                                                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      APERATURE LOCATION.                                           TR401370
C AR      APERATURE 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, APERATURE TR401420
C         RADIUS AND APERATURE LOCATION ARE ALSO DETERMINED.            TR401430
C                                                                       
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), 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), 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                                                                       
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                   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              02/18/68                    PAGE  2
          TWNR4Z      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      1 QMTRX2(3,100)                                                       L2008000
C
        INTEGER  RAY, SMISS                                                 TR401500
C
        REAL  N1, N2, NSUM                                                  TR402000
        DIMENSION  XVCT(3), QVCT(3)                                         TR402500
C
C               TWNRY4 MUST BE CALLED FROM MAIN PROGRAM FOR EACH            TR403000
C               SUBSTITUTION SET AS FOLLOWS -                               TR403200
C
C                           100 PRINT OUT                                   TR403400
C                               CALL  TWNRY4                                TR403600
C                               CALL  PRNSUB( $100 )                        TR403800
C                               DONE                                        TR404000
C
C               START OF TWNRY4 SUBROUTINE                                  TR405000
C
        CALL  PRTCTL                                                        TR405200      2
        HMAX = HO + FLOAT( NOBJH - 1 ) * DELH                               TR405400
        WRITE ( 6, 32000 )  HMAX                                            TR406000      4
32000 FORMAT( 1H0, 34HTWINRAY FOR COLOR 1 OBJECT POINT (, E14.7, 4H,0.,)TR406100
        LINE = LINE + 2                                                     TR406500
        DO  500  I = 1, 3                                                   TR407000
        DO  500  J = 1, 100                                                 TR407200
        XMTRX1(I,J) = 0.0                                                   TR407400
        QMTRX1(I,J) = 0.0                                                   TR407600
        XMTRX2(I,J) = 0.0                                                   TR407800
  500 QMTRX2(I,J) = 0.0                                                     TR408000
        THTR = 0.0                                                          TR408500
        SNTHTR = 0.0                                                        TR408700
        CSTHTR = 1.0                                                        TR408900
        N1 = NSURF                                                          TR409000
        N2 = NSURF                                                          TR409200
        M = NSURF + 1                                                       TR409400
C
C               TRACE RAY 1                                                 TR421000
        RAY = 1                                                             TR421200
        CALL  RAYTR( 0., -1., HMAX, 1, XVCT, QVCT, SMISS, $800, $900,      TR421500
      1 $1000, 1, XMTRX1, QMTRX1 )                                          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                                        TR422300     27
32010 FORMAT( 1H0, 3HRAY, I2, 15H MISSED SURFACE, I3 )                      TR422400
        GO TO 1050                                                          TR422600
C
C               RAY 1 REFLECTED ON SURFACE NO. SMISS                        TR423000
  900 N1 = SMISS - 1                                                        TR423200
  950 WRITE ( 6, 32020 )  RAY, SMISS                                        TR423300     30
32020 FORMAT( 1H0, 3HRAY, I2, 21H REFLECTED ON SURFACE, I3 )                TR423400
        GO TO 1050                                                          TR423600
C
C               RAY 1 OR 2 VIGNETTED ON SURFACE NO. SMISS                   TR424000
 1000 WRITE ( 6, 32030 )  RAY, SMISS                                        TR424200     32
32030 FORMAT( 1H0, 3HRAY, I2, 21H VIGNETTED ON SURFACE, I3 )                TR424300
```

```
      SUBROUTINE TO COMPUTE TWINRAY 4 DIAGNOSTIC INFORMATION                02/18/68                PAGE  3
            TWNR4Z       - EFN   SOURCE STATEMENT  -  IFN(S)  -

 1050 LINE = LINE + 2                                                  TR424500
 1100 IF( RAY .NE. 1 )  GO TO 1600                                     TR424700
C
C              TRACE RAY 2                                             TR425000
      RAY = 2                                                          TR425200
      CALL  RAYTR( 0., 1., NMAX, 1, XVCT, QVCT, SMISS, $1300, $1400,   TR425500
     1 $1000, 1, XMTRX2, QMTRX2 )                                      TR425600    38
      GO TO 1600                                                       TR426000
C
C              RAY 2 MISSED SURFACE NO. SMISS                          TR427000
 1300 N2 = SMISS - 1                                                   TR427200
      GO TO 950                                                        TR427400
C
C              RAY 2 REFLECTED ON SURFACE NO. SMISS                    TR428000
 1400 N2 = SMISS - 1                                                   TR428200
      GO TO 950                                                        TR428400
 1600 N = MIN0( N1, N2 )                                               TR429000
C
C              PRINT TWINRAY HEADING                                   TR441000
      WRITE ( 6, 32040 )                                               TR441500    46
32040 FORMAT( 1H0, 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 3000TR446000
      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  PRTCTL                                                     TR448600    96
      WRITE ( 6, 32040 )                                               TR448800    97
      LINE = LINE + 2                                                  TR449000
C
C              PRINT OUT [illegible]NRAY INFORMATION                   TR461000
 3500 ITEMP = 1                                                        TR461200
      WRITE ( 6, 32045 )                                               TR461300    99
32045 FORMAT( 1H  )                                                    TR461350
```

```
        SUBROUTINE TO COMPUTE TWINRAY 4 DIAGNOSTIC INFORMATION              02/16/68                  PAGE  4
             TWNR4Z      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      WRITE ( 6, 32030 )  ITEMP, I, ( XNTRX1(K,J), K=1,3 ),              TR461400
     1 ( ONTRX1(K,J), K=1,3 ), TAU1                                      TR461500     100
      ITEMP = 2                                                          TR462000
      WRITE ( 6, 32030 )  ITEMP, I, ( XNTRX2(K,J), K=1,3 ),              TR462200
     1 ( ONTRX2(K,J), K=1,3 ), TAU2                                      TR462300     109
32030 FORMAT( 1H , I2, I7, 3X, 7E16.7 )                                  TR462500
      LINE = LINE + 3                                                    TR462700
      IF( AR .LT. 0.0 )  GO TO 5000                                      TR463000
C
C              PRINT OUT APERTURE INFORMATION                            TR463500
      WRITE ( 6, 32060 )  AL, AR                                         TR463800     121
32060 FORMAT( 1H0, 14X, 18HAPERTURE LOCATION=, E14.7, 18H  APERTURE RADITR464000
     1US=, E14.7  )                                                      TR464200
      LINE = LINE + 2                                                    TR465000
 5000 CONTINUE                                                           TR466000
C
C              END OF TWNRY4 SUBROUTINE                                  TR491000
      RETURN                                                             TR492000
      END                                                                TR493000
```

SUBROUTINE TO COMPUTE TWINRAY-4 DIAGNOSTIC INFORMATION 02/16/68 PAGE 9
TWNR4Z STORAGE MAP

SUBROUTINE TWNRY4
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
LMODE	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
NBPLN	00044	I	NOBJN	00045	I	NSURF	00046	I
IFLAG	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
OLPLN	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
CDIST	00071	R	HO	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
RDTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
WOBJH	00117	R	WCLRH	00125	R	WINHT	00135	R
EINHT	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 AZOBJ ORIGIN 06667 LENGTH 00003

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R

COMMON BLOCK PRNT ORIGIN 06672 LENGTH 00002

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LINE	00000	I	PAGE	00001	I			

COMMON BLOCK TWTWIN ORIGIN 06674 LENGTH 02260

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
TEMPS	00000	R	XMTRX1	00000	R	QMTRX1	00454	R
XMTRX2	01130	R	QMTRX2	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	H1	11164	R
H2	11165	R	HSUM	11166	R	HMAX	11167	R
I	11170	I	J	11171	I	N1	11172	I
N2	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 08/18/68 PAGE 6
TWNR4Z STORAGE MAP

ENTRY POINTS

TWNRY4	SECTION	11

SUBROUTINES CALLED

PRTCTL	SECTION	12	.FWRD.	SECTION	13	RAYTR	SECTION	14
.UNOB.	SECTION	15	.FFIL.	SECTION	16	.FCNV.	SECTION	17
E.1	SECTION	18	E.2	SECTION	19	E.3	SECTION	20
E.4	SECTION	21	CC.1	SECTION	22	CC.2	SECTION	23
CC.3	SECTION	24	CC.4	SECTION	25	SYSLOC	SECTION	26

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	FORMAT	11220	500	14A	11407	800	26A	11455
900	29A	11473	1000	32A	11511	1100	34A	11526
[illegible]0	27A	11460	32010	FORMAT	11232	1050	33A	11523
[illegible]0	30A	11476	32020	FORMAT	11241	32030	FORMAT	11251
[illegible]00	45A	11565	1300	41A	11555	1400	43A	11561
32040	FORMAT	11261	5000	122A	12124	475	63A	11650
477	74A	11675	3000	93A	11744	3500	98A	11765
32045	FORMAT	11303	32050	FORMAT	11306	32060	FORMAT	11312

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12156.

```
$*
$TITLE          SUBROUTINE TO COMPUTE TWINRAY S DIAGNOSTIC INFORMATION
$IBFTC TWNRSZ  M94,XR7              LINK 3   (TWNRYS)                  TRS01000
```

```
      SUBROUTINE  TWNRYS                                                TR501200
C
C    *** OUTPUTS ***                                                    TR501500
C XNTRX1 MATRIX OF RAY 1 POSITIONS AT EACH SURFACE.                     TR501510
C XNTRX2 MATRIX OF RAY 2 POSITIONS AT EACH SURFACE.                     TR501520
C QNTRX1 MATRIX OF RAY 1 DIRECTIONS AT EACH SURFACE.                    TR501530
C QNTRX2 MATRIX OF RAY 2 DIRECTIONS AT EACH SURFACE.                    TR501540
C TAU1    THICKNESS FOR RAY 1.                                          TR501550
C TAU2    THICKNESS FOR RAY 2.                                          TR501560
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, CONTRL(10), TITLE(12), DATE(3), PUNCID,   DT001000
     1 DATA(3488)                                                       DT001100
      DIMENSION  NOBJH(7), WCLRH(7), WINHT(7), EINHT(7), CINPL(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), 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), DIPLN ), ( 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), NOBJH(1) ), (DATA(60), WCLRH(1) ),       DT102300
     E ( DATA(67), WINHT(1) ), (DATA(74), EINHT(1) ), ( DATA(81),       DT102400
     F CINPL(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
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                   RY004000
C
      COMMON / PRNT / LINE, PAGE                                        PT002500
      INTEGER  PAGE                                                     PT002700
C
      COMMON / TNTWIN / 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
```

```
     SUBROUTINE TO COMPUTE TWINRAY S DIAGNOSTIC INFORMATION             08/16/68          PAGE  2
          TWNRSZ     - EFN   SOURCE STATEMENT  -  IFN(S)  -

      DIMENSION  XVCT(3), QVCT(3)                                       TR302500
      INTEGER  RAY, SMISS                                               TR003000
C
C             TWNRYS MUST BE CALLED FROM MAIN PROGRAM EACH SUBSTITUTION TR008000
C             SET AS FOLLOWS -                                          TR008100
C                        100 PRINT OUT                                  TR008200
C                            CALL  TWNRYS                               TR008300
C                            CALL  PRNSUB( $100 )                       TR008400
C                            DONE                                       TR008500
C
C             START OF TWNRYS SUBROUTINE                                TR008000
C
      CALL  PRTCTL                                                      TR008500    2
      WRITE ( 6, 32000 )                                                TR101000    3
32000 FORMAT( 1HO, 41HTWINRAY FOR COLOR 1  OBJECT POINT (0.,0.)   )     TR101200
      LINE = LINE +2                                                    TR101300
      DO  500  I = 1, 3                                                 TR101500
      DO  500  J = 1, 100                                               TR101600
      XMTRX1(I,J) = 0.0                                                 TR101700
      QMTRX1(I,J) = 0.0                                                 TR101800
      XMTRX2(I,J) = 0.0                                                 TR101900
  500 QMTRX2(I,J) = 0.0                                                 TR102000
      THTR = 0.0                                                        TR102500
      SNTHTR = 0.0                                                      TR102600
      CSTHTR = 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., OMGA2,  0., 1, XVCT, QVCT, SMISS, $700, $800,    TR104500
     1 $900, 1, XMTRX1, QMTRX1 )                                        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   26
32010 FORMAT( 1HO, 3HRAY, I2, 15H MISSED SURFACE, I3 )                  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( 1HO, 3HRAY, I2, 21H REFLECTED ON SURFACE, I3 )            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( 1HO, 3HRAY, I2, 21H VIGNETTED ON SURFACE, I3 )            TR107500
      LINE = LINE + 2                                                   TR108000
 1000 IF( RAY .NE. 1 )  GO TO 1500                                      TR109000
C
C             TRACE RAY 2                                               TR201000
```

```
      SUBROUTINE TO COMPUTE TWINRAY 2 DIAGNOSTIC INFORMATION          02/16/68           PAGE  3
          TWNRS2      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      RAY = 2                                                          TR201900
      CALL  RAYTR( 0., 1., 0., 1, XVCT, QVCT, SMISS, $1200, $1300,     TR202000
     1  $900, 1, XMTRX2, QMTRX2 )                                      TR202100    36
      GO TO 1500                                                       TR202500
C
C             RAY 2 MISSED SURFACE NO. SMISS                           TR203000
 1200 N2 = SMISS - 1                                                   TR203200
      GO TO 750                                                        TR203500
C
C             RAY 2 REFLECTED ON SURFACE NO. SMISS                     TR204000
 1300 N2 = SMISS - 1                                                   TR204200
      GO TO 850                                                        TR204500
C
C             PRINT TWINRAY HEADING                                    TR205000
 1500 WRITE ( 6, 32040 )                                               TR205200    43
32040 FORMAT( 1H0,12HRAY  SURFACE, 9X, 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(5,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(5,J).LT. 0. )  TAU2 = -TAU2                            TR208200
  477 CONTINUE                                                         TR208400
      IF( LINE .LE. 52 )  GO TO 3000                                   TR301800
C
C             BEGIN NEW PAGE  -  PRINT TITLE                           TR302000
      CALL  PRTCTL                                                     TR302500    76
      WRITE ( 6, 32040 )                                               TR303000    77
      LINE = LINE + 2                                                  TR303200
C
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    89
32050 FORMAT( 1H , I2, I7, 3X, 7E16.7 )                                TR305500
      LINE = LINE + 3                                                  TR306000
 5000 CONTINUE                                                         TR306500
C
C             END OF TWNRYS SUBROUTINE                                 TR309000
      RETURN                                                           TR309500
      END                                                              TR309600
```

SUBROUTINE TO COMPUTE TWINRAY'S DIAGNOSTIC INFORMATION 08/18/68 PAGE 4
TWNRSZ STORAGE MAP

SUBROUTINE TWNRYS
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	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	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	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	00071	R	HO	00072	R	DELH	00073	R
SYSHX	00074	R	WXDIR	00075	R	WYDIR	00076	R
RDTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
NOBJH	00117	R	WCLRH	00126	R	WINHT	00135	R
EINHT	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 AZOBJ ORIGIN 08667 LENGTH 00003

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R

COMMON BLOCK PRNT ORIGIN 08672 LENGTH 00002

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LINE	00000	I	PAGE	00001	I

COMMON BLOCK TNTWIN ORIGIN 08674 LENGTH 02260

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
TEMPS	00000	R	XMTRX1	00000	R	QMTRX1	00454	R
XMTRX2	01130	R	QMTRX2	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	N2	11167	I
N	11170	I	TAU1	11171	R	TAU2	11172	R
ITEMP	11173	I						

SUBROUTINE TO COMPUTE TWINRAY S DIAGNOSTIC INFORMATION 08/18/68 PAGE 9

TWNRS2 STORAGE MAP

ENTRY POINTS

TWNRYS SECTION 11

SUBROUTINES CALLED

PRTCTL	SECTION	12	.FWRD.	SECTION	13	RAYTR	SECTION	14
.UN06.	SECTION	15	.FFIL.	SECTION	16	.FCNV.	SECTION	17
SYSLOC	SECTION	18						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
32000	FORMAT	11210	580	13A	11341	700	25A	11408
800	28A	11426	900	31A	11447	1000	32A	11464
750	26A	11410	32010	FORMAT	11221	850	29A	11431
32020	FORMAT	11230	32030	FORMAT	11240	1500	43A	11523
1200	39A	11513	1300	41A	11517	32040	FORMAT	11250
5000	98A	11761	475	60A	11575	477	72A	11623
3000	78A	11644	32050	FORMAT	11272	32060	FORMAT	11274

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 12013.

Section 1

LINK 4

```
$*
$TITLE          SUBROUTINE TO COMPUTE ENTRANCE PUPIL COORDINATES
$IBFTC LATTZZ  M94,XR7               LINK 4    (LATT)                    LT001000
```

```
      SUBROUTINE  LATT( XOHAT, YOHAT, NRAYS )                           LT001200
C
C     *** INPUTS ***                                                    LT001500
C LATTC  MATRIX SPECIFYING ENTRANCE PUPIL COORDINATES.   (DATA COMMON)  LT001510
C NRAYS  RAY NUMBER BEING GENERATED.                                    LT001520
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      NRAYS = 0                                                        LT003000
C      RCNT  = 0                                                        LT003100
C      DO  20  I = 1, NRAYS                                             LT003200
C      IF( LMODE .EQ. 0 )  RCNT = 1                                     LT003300
C       .       .       .                                               LT003400
C      CALL    LATT( XOHAT, YOHAT, NRAYS )                              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(81), LATTC(1,1), ILATTC(1,1) ), ( DATA(14),   LT005100
     1 DELY )                                                           LT005150
      DIMENSION  LATTC(3,28), ILATTC(3,28)                              LT005200
      REAL  LATTC                                                       LT005300
C
      INTEGER  SLICE                                                    LT005400
C
C            START OF LATT SUBROUTINE                                   LT005500
C
      IF( NRAYS .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 = FLOAT( NSLR ) * DELY + LATTC(2,SLICE)                     LT008400
```

```
      GO TO 2000                                               LT008800
C
C              BEGIN RAY GENERATION ON NEXT SLICE              LT009000
 1500 SLICE = SLICE + 1                                        LT009100
      GO TO 500                                                LT009200
C
C              INCREMENT RAY COUNTERS                          LT009300
 2000 NSLR = NSLR + 1                                          LT009400
      NRAYS = NRAYS + 1                                        LT009500
C
C              END OF LATT SUBROUTINE                          LT009600
      RETURN                                                   LT009700
      END                                                      LT009800
```

```
SUBROUTINE TO COMPUTE ENTRANCE PUPIL COORDINATES                     02/16/68                  PAGE  3
    LATTZZ                                          STORAGE  MAP

                                       SUBROUTINE  LATT
                                           COMMON   VARIABLES

                COMMON BLOCK     DATA               ORIGIN     00001              LENGTH     08666

SYMBOL      LOCATION     TYPE            SYMBOL     LOCATION     TYPE             SYMBOL     LOCATION     TYPE
DUMMY        00000        R              DATA        00033        R               LATTC       00170        R
ILATTC       00170        I              DELY        00050        R
                                            UNDIMENSIONED PROGRAM VARIABLES

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

     LATT         SECTION     5
                                                    SUBROUTINES CALLED

     E.1          SECTION     6               E.2         SECTION     7                E.3        SECTION     8
     E.4          SECTION     9               CC.1        SECTION    10                CC.2       SECTION    11
     CC.3         SECTION    12               CC.4        SECTION    13                SYSLOC     SECTION    14
                                              EFN    IFN    CORRESPONDENCE

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.
```

```
$*
$TITLE        SUBROUTINE TO COMPUTE BACK FOCUS AND F/ NUMBER
$IBFTC MERIDZ  M94,XR7              LINK 4    (MERID)                 MR000100
```

```
        SUBROUTINE TO COMPUTE BACK FOCUS AND F/ NUMBER                            02/10/68                    PAGE  1
            MERIDZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  MERID( COLOR, BFOCUS, FSNUMB )                        MR000200
C
C     *** INPUTS ***                                                    MR000500
C COLOR   NUMBER OF COLOR TO BE USED.                                   MR000600
C ONGA1   MERIDIONAL SCALE FACTOR FOR BACK FOCUS.         (DATA COMMON) MR000700
C
C     *** OUTPUTS ***                                                   MR000800
C BFOCUS BACK FOCUS VALUE.                                              MR000900
C FSNUMB F/ NUMBER.                                                     MR001000
C
C     *** DESCRIPTION ***                                               MR001100
C         THIS SUBROUTINE COMPUTES THE BACK FOCUS AND F/ NUMBER FOR THE MR001200
C         PROVIDED COLOR NUMBER.  IF THEY ARE INFINITE, THEN BFOCUS AND MR001300
C         FSNUMB ARE SET TO ZERO.                                       MR001400
C
      COMMON  / DATA / DUMMY1(27), DATA(3483)                           MR001600
      EQUIVALENCE  ( DATA(24), ONGA1 )                                  MR001700
C
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                   AZ004000
C
      INTEGER  COLOR                                                    MR001800
      DIMENSION  XVCT(3), QVCT(3)                                       MR001900
C
C               START OF MERID SUBROUTINE                               MR002400
C
      BFOCUS = 0.0                                                      MR002600
      FSNUMB = 0.0                                                      MR002800
      CALL  RAYTR( 0., ONGA1, 0., COLOR, XVCT, QVCT, SMISS, $400, $500, MR003000
     1 $600, 0, DUMMY, DUMMY )                                          MR003200    2
      GO TO 600                                                         MR003400
  400 WRITE ( 6, 32000 )  SMISS                                         MR004000    5
32000 FORMAT( 1H0, 29HMERIDIONAL RAY MISSED SURFACE, I4 )               MR004200
      CALL  GOOF                                                        MR004400    6
  500 WRITE ( 6, 32010 )  SMISS                                         MR005000    7
32010 FORMAT( 1H0, 35HMERIDIONAL RAY REFLECTED AT SURFACE, I4 )         MR005200
      CALL  GOOF                                                        MR005400    8
  600 IF( QVCT(2) .EQ. 0.0 )  GO TO 1000                                MR008000
      ALPHA = - QVCT(3) / QVCT(2)                                       MR008200
      BFOCUS = ALPHA * XVCT(2) + XVCT(3)                                MR008400
      FSNUMB = .5 * ABS( ALPHA * ONGA1 )                                MR008600
C
 1000 RETURN                                                            MR009000
      END                                                               MR009200
```

SUBROUTINE TO COMPUTE BACK FOCUS AND F/ NUMBER 02/18/68 PAGE 2
MERIDZ STORAGE MAP

SUBROUTINE MERID
COMMON VARIABLES

COMMON BLOCK DATA ORIGIN 03001 LENGTH 08666

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DUMMY1	00000	R	DATA	00033	R	OMGA1	00082	R

COMMON BLOCK AZOBJ ORIGIN 08667 LENGTH 00003

THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
XVCT	08672	R	QVCT	08675	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SMISS	08700	R	DUMMY	08701	R	ALPHA	08702	R

ENTRY POINTS

MERID SECTION 7

SUBROUTINES CALLED

RAYTR	SECTION	8	.FWRD.	SECTION	9	GOOF	SECTION	10
.UNDS.	SECTION	11	.FFIL.	SECTION	12	.FCNV.	SECTION	13
SYSLOC	SECTION	14						

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.

```
$*
$TITLE        SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION
$IBFTC SMFOFZ  M94,XR7              LINK 4    (SMFOF)                 SM001010
```

```
      SUBROUTINE  SMFOF( COLOR, NNUMB, HEIGHT, BF1, CMFLAG, TEMPS )       SM001200
C
C     *** INPUTS ***                                                       SM001500
C COLOR  NUMBER OF COLOR TO BE USED.                                       SM001510
C NNUMB  OBJECT POINT INDEX TO BE USED.                                    SM001520
C HEIGHT OBJECT HEIGHT.                                                    SM001530
C BF1    COMPUTED BACK FOCUS FOR COLOR NUMBER 1.                           SM001540
C CMFLAG FLAG FOR TYPE OF COMPUTATIONS TO BE PERFORMED AS FOLLOWS -        SM001550
C               1) OPTICS DIAGNOSTIC                                       SM001560
C               2) DESIGN                                                  SM001570
C               3) PLOT                                                    SM001580
C               4) SENSITIVITY                                             SM001590
C
C     *** OUTPUTS ***                                                      SM001600
C NMISS  NUMBER OF RAYS WHICH MISSED.                   (TMPATT COMMON)    SM001610
C NRFLCT NUMBER OF RAYS WHICH REFLECTED.                (TMPATT COMMON)    SM001620
C NVIGN  NUMBER OF RAYS WHICH VIGNETTED.                (TMPATT COMMON)    SM001630
C NRAYSD NUMBER OF RAYS WHICH ARE SUCCESSFUL.           (TMPATT COMMON)    SM001640
C SJSTAR POSITION OF PRINCIPAL IMAGE PLANE.             (TMPATT COMMON)    SM001650
C XBAR   AVERAGE X                                      (TMPATT COMMON)    SM001660
C YBAR   AVERAGE Y                                      (TMPATT COMMON)    SM001670
C XSTAR  RMS X.                                         (TMPATT COMMON)    SM001680
C YSTAR  RMS Y.                                         (TMPATT COMMON)    SM001690
C RSTAR  SPOT SIZE.                                     (TMPATT COMMON)    SM001700
C XCURL  VECTOR OF X COORDINATES OF RAYS ON IMAGE PLANE.(TMPATT COMMON)    SM001710
C YCURL  VECTOR OF Y COORDINATES OF RAYS ON IMAGE PLANE.(TMPATT COMMON)    SM001720
C
C     *** DESCRIPTION ***                                                  SM001800
C        THIS SUBROUTINE TRACES THE SPECIFIED RAY PATTERN AND COMPUTES     SM001810
C        VARIOUS PARAMETERS DEPENDENT UPON THE SETTING OF CMFLAG.          SM001820
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), NRAYS ), ( DATA(3),      DT101000
     1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),         DT101100
     2 NSUBT ), ( DATA(7), NSUBF ), ( DATA(8), NIFLN ), ( 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), HEXPP ), ( DATA(20), DEXPP ), ( DATA(21),      DT101600
     7 WEXPP ), ( DATA(22), DLFLN ), ( DATA(23), OMGA2 ), ( DATA(24),      DT101700
     8 OMGA1 ), ( DATA(25), DELD  ), ( DATA(26), EFRAD ), ( DATA(27),      DT101800
     9 FSCAL ), ( 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), 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
```

```
      SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION                        09/20/68          PAGE  2
         SMFOPZ       - EFN    SOURCE STATEMENT  -  IFN(S)  -

     1 ( DATA(2082), 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  CONTRL, PFLAG, DATE, ATRGGR                                DT201000
      REAL  LATTC                                                         DT202000
C
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                     AZ004000
C
      COMMON / THPATT / NMISS, NREFLCT, NVIGN, NRAYSD, 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
      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                    L3002000
C
      INTEGER  COLOR, HNUMB, CMFLAG, RCNT, SMISS, RYCNT                    SM002500
C
      DIMENSION  TEMPS(1), XVCT(3), QVCT(3)                               SM003000
C
C              START OF SMPOP SUBROUTINE                                  SM005000
C
               ZERO OUT LINK3 COMMON BLOCK                                SM005500
      DO  1000  I = 1, 425                                                SM005600
 1000 TLINK(I) = 0.0                                                      SM005800
      SJSTAR = DELD + CINPL( HNUMB )                                      SM006000
      IF( IMODE .EQ. 0 )  SJSTAR = SJSTAR + DF1                           SM006200
      MRAYS = 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
C              LOOP TO GENERATE AND TRACE EACH RAY FOR COLOR = COLOR AND  SM101000
C              OBJECT HEIGHT = HNUMB                                      SM101100
 1300 DO  8000  I = 1, NRAYS                                              SM101500
C              SET FLAG TO INDICATE RAY ( -XOHAT, YOHAT ) IS TO BE TRACEDSM101800
      IF( LMODE .EQ. 0 )  RCNT = 1                                        SM102000
C              GENERATE CURRENT RAY                                       SM102500
      CALL  LATT( XOHAT, YOHAT, MRAYS )                                   SM102700
C
C              TRACE CURRENT RAY                                          SM103000      34
 1400 CALL  RAYTR( XOHAT, YOHAT, HEIGHT, COLOR, XVCT, QVCT, SMISS,        SM103200
     1 $2000, $3000, $4000, 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 NRFLCT = NRF   T + 1                                                SM105200
```

```
      SUBROUTINE TO COMPUTE SPOT SIZE INFORMATION                          09/20/68                PAGE  3
          SMFOP2      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      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   100000
      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
      XOHAT = - XOHAT                                                     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
```

```
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 + XCURL(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 SMFOP SUBROUTINE                            SM901000   205
32000 RETURN                                                   SM902000
      END                                                      SM903000
```

SUBROUTINE SMFOP
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	NOBJH	00045	I	NSURF	00046	I
JFLAG	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
CLPLN	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
CDIST	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
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	ATRGGR	06666	I
GAUSS	06667	R						

COMMON BLOCK AZOBJ ORIGIN 06672 LENGTH 00003

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R

COMMON BLOCK TMPATT ORIGIN 06675 LENGTH 00651

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NMISS	00000	I	NRFLCT	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	00323	R
MXBAR	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 09/20/68 PAGE 6

SMPOP2 STORAGE MAP

FCNT	07554	I	SMISS	07555	I	RYCNT	07556	I
NRAYS	07557	I	MR	07560	I	XOHAT	07561	R
YOHAT	07562	R	DUMMY	07563	R	SL	07564	R
XNR	07565	R						

ENTRY POINTS

SMPOP SECTION 9

SUBROUTINES CALLED

LATT	SECTION	10	RAYTR	SECTION	11	SORT	SECTION	1
E.1	SECTION	13	E.2	SECTION	14	E.3	SECTION	1
E.4	SECTION	16	CC.1	SECTION	17	CC.2	SECTION	1
CC.3	SECTION	19	CC.4	SECTION	20	SYSLOC	SECTION	2

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIO
1000	5A	07604	1300	28A	07644	1200	24A	07641
8000	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.

Section 5

LINK 5

```
     $JOB  INF,N010001,44800-0,77387,AM IC1,8,6000   WILSON   -  INFORMATICS      A 17*28**59*** 02/16/68 *****$LIONKI(
$IBJOB        NOGO
$*
$TITLE        SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN
$IBFTC BNDCHZ  M94,XR7           LINK 5    (BNDCHK)                 BC001000
```

```
      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 VIOLATIONBC001690
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), 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), 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), 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), 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                                                                       
      COMMON/PRNT/LINE,PAGE                                             PT002500
      INTEGER PAGE                                                      PT002700
C                                                                       
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                   AZ004000
C                                                                       
      COMMON / THDESN /  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
      EQUIVALENCE  ( NVARX, VCTSTR(6000)), ( JI, VCTSTR(5999)), ( NJ, VCBC002200
     1TSTR(5998)), ( NJA, VCTSTR(5997)), ( KMA, VCTSTR(5996)), ( U, VCTSBC002400
     2TR(5995)), ( RL, VCTSTR(5994)), ( RLA, VCTSTR(5993)), ( UMIN, VCTSBC002600
     3TR(5992)), ( UMAX, VCTSTR(5991)), ( ICOLOR(1), VCTSTR(5985)), ( QMBC002800
     4IN, VCTSTR(5984)), ( QMAX, VCTSTR(5983)), ( INDX, VCTSTR(5982)),  BC003000
     5 ( DDIST(1), VCTSTR(5800)), ( ALENG, VCTSTR(5981)), ( XMTRX(1,1), BC003200
     6VCTSTR(5700)), ( QMTRX(1,1), VCTSTR(5400)), ( YOHAT(1), VCTSTR(539BC003400
     76)), ( HTAB(1), VCTSTR(5392)), ( SMISS, VCTSTR(5391)), ( XVCT(1), BC003600
     8VCTSTR(5388)), ( QVCT(1), VCTSTR(5385)), ( TAU, VCTSTR(5384)),    BC003800
     9 ( NUMBER, VCTSTR(5383)), ( KIND, VCTSTR(5382))                   BC004000
C
      INTEGER  SMISS, RL, RLA                                           BC005000
      DIMENSION  ICOLOR(6), DDIST(100), XMTRX(3,100), QMTRX(3,100),     BC006000
     1 YOHAT(4), HTAB(4), XVCT(3), QVCT(3)                              BC006200
C
C              START OF BNDCHK SUBROUTINE                               BC101000
C
  100 IF( NCOND .EQ. 0 )  GO TO 5000                                    BC101200
      NVARX = NDSGV(1)                                                  BC101400
      DO  3000  I = 1, NVARX                                            BC101600
      JI = NDSGV(I+1)                                                   BC101800
      NJ = IDESGN(1,JI)                                                 BC102000
      NJA = IABS(NJ)                                                    BC102200
      DO  3000 M = 1, NJA                                               BC102400
      KMA = IABS( IDESGN(M+3,JI) )                                      BC102600
      U = DATA(KMA)                                                     BC102800
      DO  2500  L = 1, NCOND                                            BC103000
      RL = IBNDS( 1,L )                                                 BC103200
      RLA = IABS(RL)                                                    BC103400
      UMIN = BOUNDS( 2,L )                                              BC103600
      UMAX = BOUNDS( 3,L )                                              BC103800
      IF ( RL .LT. 0 )  GO TO 1000                                      BC104000
      IF ( KMA .NE. RL )  GO TO 2500                                    BC104200
      IF ( NJ .LT. 0 )  GO TO 500                                       BC104400
      IF ( UMIN .LE. U .AND. U .LE. UMAX )  GO TO 3000                  BC104600
      GO TO 3500                                                        BC104800
  500 IF ( UMIN .LE. (1./ U) .AND. (1./ U) .LE. UMAX ) GO TO 3000       BC105000
      GO TO 3500                                                        BC105200
C
 1000 IF ( KMA .LT. RLA .OR. KMA .GT. (RLA + NC-1 ) )  GO TO 2500       BC106000
      CALL CORDER ( NCLRS, DATA(RLA), ICOLOR )                          BC106200     45
      DO 1500 J = 1, NCLRS                                              BC106400
      IF ( CORD(J) .NE. ICOLOR(J) ) GO TO 3500                          BC106600
 1500 CONTINUE                                                          BC106800
      QMIN = DATA(RLA)                                                  BC107000
      QMAX = QMIN                                                       BC107200
      IF ( NCLRS .EQ. 1 ) GO TO 2200                                    BC107400
      DO  2000  J = 2, NCLRS                                            BC107600
      JNDX = RLA + J - 1                                                BC107800
      QMIN = AMIN1( QMIN, DATA(JNDX) )                                  BC108000
 2000 QMAX = AMAX1( QMAX, DATA(JNDX) )                                  BC108200
 2200 IF( QMIN .LT. UMIN )  GO TO 3500                                  BC108400
      IF( QMAX .GT. UMAX )  GO TO 3500                                  BC108600
      GO TO 3000                                                        BC108800
 2500 CONTINUE                                                          BC109000
```

```
      SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN          02/16/68              PAGE   3
         BNDCHZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

 3000 CONTINUE                                                          BC109200
      GO TO 5000                                                        BC109400
C
C              BOUNDARY CONDITION HAS BEEN VIOLATED                     BC201000
 3500 IF( IFLAG .NE. 0 )  GO TO 4000                                    BC201200
      CALL DPRNT                                                        BC201300      85
      WRITE ( 6, 32010 )  JI, N, U                                      BC201400      86
32010 FORMAT( 1H0, 9X, 15HDESIGN VARIABLE, I3, 10H PARAMETER, I2, 32H VIBC201600
     1OLATED A BOUNDARY CONDITION (, E14.7, 1H) )                       BC201800
      LINE = LINE + 2                                                   BC202000
C
 4000 CALL  PRMSUB( $4000 )                                             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( 1H0, 9X, 18HTOTAL LENS LENGTH=, E14.7, 29H  VIOLATES BOUNDBC206000
     1ARY CONDITION )                                                   BC206200
      LINE = LINE + 2                                                   BC206400
      GO TO 4000                                                        BC206600
 6000 THTR = 0.                                                         BC206800
      SNTHTR = 0.                                                       BC207000
      CSTHTR = 1.                                                       BC207200
      YOHAT(1) = - OMGAF                                                BC207400
      YOHAT(2) = OMGAF                                                  BC207600
      YOHAT(3) = OMGA2                                                  BC207800
      YOHAT(4) = OMGAF                                                  BC208000
      HTAB(1) = HO + FLOAT( NOBJH-1 ) * DELH                            BC208200
      HTAB(2) = HTAB(1)                                                 BC208400
      HTAB(3) = 0.                                                      BC208600
      HTAB(4) = 0.                                                      BC208800
      NUPPER = NSURF - 1                                                BC209000
      IF( NSURF .LE. 1 )  GO TO 7500                                    BC209200
      DO  7000  I = 1, 4                                                BC301000
      CALL  RAYTR( 0., YOHAT(I), HTAB(I), 1, XVCT, QVCT, SMISS, $8000,  BC301200
     1 $8500, $8500, 1, XMTRX, QMTRX )                                  BC301400      127
 6500 DO  7000  J = 1, NUPPER                                           BC301600
      TAU = XMTRX(3,J+2) - XMTRX(3,J+1) + SURFC(4,J+1)                  BC301800
      IF( SURFC(5,J+1) .LT. 0. )  TAU = - TAU                           BC302000
      IF( TAU .GT. SPFEA )  GO TO 7000                                  BC302200
      KIND = 3                                                          BC302400
      GO TO 9000                                                        BC302600
 7000 CONTINUE                                                          BC302800
```

```
        SUBROUTINE TO CHECK FOR BOUNDARY VIOLATIONS DURING DESIGN            02/16/68                    PAGE  4
          BNDCHZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

 7500 CALL  PRNSUB( 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                                BC305000
      CALL OPRNT                                                    BC305100     156
      GO TO ( 9200, 9400, 9600 ),  KIND                             BC305200
 9200 WRITE ( 6, 32030 )  I, SMISS                                  BC30540C     158
32030 FORMAT( 1H0, 9X, 17HFEATHER CHECK RAY, I2, 13H MISSED SURFACE, I3)BC305600
      LINE = LINE + 2                                               BC305800
      GO TO 4000                                                    BC306000
 9400 WRITE ( 6, 32040 )  I, SMISS                                  BC306200     160
32040 FORMAT( 1H0, 9X, 17HFEATHER CHECK RAY, I2, 21H REFLECTED AT SURFACBC306400
     1E, I3 )                                                       BC306600
      LINE = LINE + 2                                               BC306800
      GO TO 4000                                                    BC307000
 9600 K = J + 1                                                     BC307200
      WRITE ( 6, 32050 )  J, K, TAU, I                              BC307400     163
32050 FORMAT( 1H0, 9X, 13HLENS SURFACES, I3, 4H AND, I3, 22H FEATHERED (BC307600
     1THICKNESS=, E14.7, 6H)  RAY, I2 )                             BC307800
      LINE = LINE + 2                                               BC308000
      GO TO 4000                                                    BC308200
      END                                                           BC309000
```

SUBROUTINE BNDCHK
COMMON VARIABLES

COMMON BLOCK DATA ORIGIN 00001 LENGTH 06666

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
MCNTRL	00000	I	CONTRL	00001	I	TITLE	00015	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	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
IFLAG	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
DLC'N	00060	R	OMGA2	00061	R	OMGA1	00062	R
DE(D	00063	R	EPRAD	00064	R	PSCAL	00065	R
OMGAF	00066	R	SPFE.	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	00103	I	NDSGV	00104	I
HOBJH	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	06663	I			

COMMON BLOCK PRNT ORIGIN 06667 LENGTH 00002

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LINE	00000	I	PAGE	00001	I

COMMON BLOCK AZOBJ ORIGIN 06671 LENGTH 00003

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R

COMMON BLOCK THDESN ORIGIN L06?4 LENGTH 13770

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
PERIOD	00000	I	PCOUNT	00001	I	HOFFND	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
VCTSTR	00175	R	VINDX	13755	I	NVARX	13754	I
JI	13753	I	NJ	13752	I	NJA	13751	I
KNA	13750	I	U	13747	R	RL	13746	I
RLA	13745	I	UMIN	13744	R	UMAX	13743	R
ICOLOR	13735	I	QMIN	13734	R	QMAX	13733	R
INDX	13732	I	DDIST	13444	R	ALENG	13731	R
XMTRX	13300	R	QMTRX	12624	R	YDHAT	12620	R
HTAB	12614	R	SMISS	12613	I	XVCT	12610	R
QVCT	12605	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 02/16/68 PAGE 6

BNDCHZ STORAGE MAP

I	22664	I	M	22665	I	L	22666	I
NC	22667	I	J	22670	I	JNOX	22671	I
K	22672	I						

ENTRY POINTS

BNDCHK SECTION 11

SUBROUTINES CALLED

CORDER	SECTION	12	DPRNT	SECTION	13	.FWRD.	SECTION	14
PRMSUB	SECTION	15	RAYTR	SECTION	16	.FXEM.	SECTION	17
.UNDS.	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	23654	9400	160A	23672	9600	162A	23710
32030	FORMAT	22751	32040	FORMAT	22763	32050	FORMAT	22777

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 24006.

```
$*
$TITLE          SUBROUTINE TO ORDER A SET OF INDICES OF REFRACTION
$IBFTC CORDEZ  M94,XR7            LINK 5   (CORDER)                CO001000
```

```
      SUBROUTINE  CORDER( N, DVCT, OVCT )                               CO001200
C
C    *** INPUTS ***                                                     CO001300
C N       NUMBER OF COLORS (INDICES OF REFRACTION).                     CO001310
C DVCT    VECTOR CONTAINING N INDICES OF REFRACTION.                    CO001320
C
C    *** OUTPUTS ***                                                    CO001330
C OVCT    VECTOR CONTAINING RELATIVE POSITIONS OF INDICES IN DVCT.      CO001340
C
C    *** DESCRIPTION ***                                                CO001350
C         CORDER IS USED TO DETERMINE THE RELATIVE ASCENDING ORDER OF   CO001360
C         THE SET OF N INDICES OF REFRACTION CONTAINED IN DVCT.         CO001370
C         THE RELATIVE ORDER IS STORED IN OVCT.                         CO001380
C
      DIMENSION  DVCT(1), OVCT(1)                                       CO001400
      INTEGER  OVCT                                                     CO002000
C
C              START OF CORDER SUBROUTINE                               CO003000
C
      DO  1000  I = 1, N                                                CO003200
      OVCT(I) = 0                                                       CO003400
      DO  1000  J = 1, N                                                CO003600
 1000 IF( DVCT(I) .GE. DVCT(J) )  OVCT(I) = OVCT(I) + 1                 CO004000
C
      RETURN                                                            CO004200
      END                                                               CO004400
```

```
        SUBROUTINE TO ORDER A SET OF INDICES OF REFRACTION                    02/16/68                  PAGE  2
          CORDEZ                                        STORAGE  MAP

                                               SUBROUTINE  CORDER
                                                  UNDIMENSIONED PROGRAM VARIABLES

SYMBO       LOCATION     TYPE           SYMBOL      LOCATION     TYPE           SYMBOL      LOCATION     TYPE
I           00001        I
                                                    ENTRY POINTS

     CORDER       SECTION    3
                                                    SUBROUTINES CALLED

     SYSLOC       SECTION    4
                                              EFN   IFN   CORRESPONDENCE

EFN         IFN          LOCATION       EFN         IFN          LOCATION       EFN         IFN          LOCATION
1000        6A           00031
 THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00102.
```

```
$*
$TITLE         SUBROUTINE TO CALCULATE REVISED ERROR VECTOR
$IBFTC CYCLEZ  M94,XR7              LINK 5   (CYCLE)                   CY001000
```

```
      SUBROUTINE CYCLE(*)                                               CY001200
C                                                                        
C     *** DESCRIPTION ***                                               CY001400
C        CYCLE INCREMENTS THE DESIGN PARAMETERS, CALCULATES THE WEIGHTEDCY001410
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, CONTRL(10), TITLE(12), DATE(3), FUNCID,   DT001000
     1 DATA(3485)                                                       DT001100
      DIMENSION  WOBJH(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), NRAYS ), ( DATA(3),    DT101000
     1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),      DT101100
     2 NSUBT ), ( DATA(7), NSUBF ), ( 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 FSCAL ), ( 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), 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
      EQUIVALENCE  ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )        DT103000
      INTEGER  CONTRL, AFLAG, DATE, ATRGGR                              DT201000
      REAL  LATTC                                                       DT202000
C
      COMMON/PRNT/LINE,PAGE                                             PT002500
      INTEGER PAGE                                                      PT002700
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), DJOLD(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
```

```
      CYCLEZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      JMAX=NDSGN(3)                                                    CY003000
      IMAX=NDSGV(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 INCRM(NDSGV(I+1),1.)                                        CY005600    16
      KZ1=VINDX(I+1)                                                   CY005800
COMPUTE E(U0+DELTA UJ) ...                                             CY006000
      CALL MEVCTR(VCTSTR(KZ1),$470,0,GAUSS)                            CY006200    20
      CALL INCRM(NDSGV(I+1),-1.)                                       CY006400
COMPUTE IJ=EJ-E0 ...                                                   CY006600    25
      DO 150 K=1,DIMVCT                                                CY006800
      KZE=VINDX(1)+K-1                                                 CY007000
      KZI=KZ1+K-1                                                      CY007200
150   VCTSTR(KZI)=VCTSTR(KZI)-VCTSTR(KZE)                              CY007400
170   CONTINUE                                                         CY007600
C                                                                      CY007800
COMPUTE  E0.IJ  J=1,2,...,IMAX ...                                     CY008000
C                                                                      CY008200
      DO 230 I=1,IMAX                                                  CY008400
      KZ1=VINDX(I+1)                                                   CY008600
      CALL DOTF(VCTSTR(DIMVCT+1),VCTSTR(KZ1),BMATRX(I))                CY008800    44
230   BMATRX(I)=-BMATRX(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.I)GO TO 310                                              CY102000
      KZE=VINDX(I+1)                                                   CY102200
      KZI=VINDX(J+1)                                                   CY102400
      CALL DOTF(VCTSTR(KZE),VCTSTR(KZI),AMATRX(I,J))                   CY102600    65
      IF(J.GT.I)GO TO 350                                              CY102800
      KZ1=NDSGV(I+1)                                                   CY103000
      DESGN(10,KZ1)=AMATRX(I,I)                                        CY103200
      GO TO 350                                                        CY103400
310   AMATRX(I,J)=AMATRX(J,I)                                          CY103600
350   CONTINUE                                                         CY103800
C                                                                      CY104000
C INVERT LINEAR SYSTEM ...                                             CY104200
C                                                                      CY104400
      IF(IMAX.NE.1)GO TO 390                                           CY104600
      IF(AMATRX(1,1).EQ.0.)GO TO 410                                   CY104800
      DETRM=AMATRX(1,1)                                                CY105000
      BMATRX(1)=BMATRX(1)/DETRM                                        CY105200
      GO TO 500  -                                                     CY105400
390   CALL MATINV(10,AMATRX,IMAX,BMATRX,1,DETRM,S1,S2)                 CY105600    92
      IF(DETRM.NE.0.)GO TO 500                                         CY105600
C                                                                      CY106000
CMATRIX IS SINGULAR ...                                                CY106200
```

```
     SUBROUTINE TO CALCULATE REVISED ERROR VECTOR                      09/24/68          PAGE
        CYCLEZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C                                                                   CY106400
410   CALL DFRNT                                                    CY106600    98
      WRITE (6,4100)                                                CY106800    99
4100  FORMAT(10X27HSOLUTION MATRIX IS SINGULAR)                     CY107000
430   LINE=LINE+1                                                   CY107200
      FCOUNT=FCOUNT+1                                               CY107400
      IF(FCOUNT.LT.PERIOD)GO TO 460                                 CY107600
      WRITE (6,4101)FCOUNT                                          CY107800   105
4101  FORMAT(10X16HDESIGN ABANDONEDI3,20H SUCCESSIVE FAILURES)      CY108000
      RETURN 1                                                      CY108200
460   CALL JURY                                                     CY108400   107
      RETURN                                                        CY108600
470   CALL INCRM(NDSGV(I+1),-1.)                                    CY108800   111
      CALL DFRNT                                                    CY109000   113
      WRITE (6,4102)NDSGV(I+1)                                      CY109200   114
4102  FORMAT(10X75HRAY FAILURE WHILE COMPUTING DERIVATIVE OF E WITH RESPCY109400
     1ECT TO DESIGN VARIABLEI3)                                     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 KCX=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,ATRGGR,GAUSS)                    CY205800   169
      CALL DOTF(VCTSTR(KZ1),VCTSTR(KZ1),ESLNG)                      CY206000   174
      ESLNG=SQRT(ESLNG)                                             CY206200   176
      IMPRV=(EOLNG-ESLNG)/EOLNG                                     CY206400
      IF(IMPRV.LE.0.)GO TO 810                                      CY206600
      CALL DFRNT                                                    CY206800   180
      IF( AFLAG .NE. 0 )  CALL  EPRNT                               CY206900   183
      EOLNG=ESLNG                                                   CY207000
      FCOUNT=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=VINDX(2)+K-1                                                   CY207800
770   VCTSTR(KZE)=VCTSTR(KZ1)                                            CY208000
      CALL MODDL                                                         CY208200     195
      RETURN                                                             CY208400
C                                                                        CY301000
C RAY FAILURE ...                                                        CY301200
C                                                                        CY301400
790   CALL DFRNT                                                         CY301600     198
      WRITE (6,4103)                                                     CY301800     199
4103  FORMAT(10X49HRAY FAILURE WHILE COMPUTING IMPROVED ERROR VECTOR)    CY302000
      GO TO 830                                                          CY302200
810   CALL DFRNT                                                         CY302400     202
      WRITE (6,4104)                                                     CY302600     203
4104  FORMAT(10X20HNEGATIVE IMPROVEMENT)                                 CY302800
830   DO 860 I=1,IMAX                                                    CY303000
860   CALL INCRM(NDSGV(I+1),-BMATRX(I))                                  CY303200     211
      IF(ATRGGR.EQ.0)GO TO 870
      IF(NSUBT.NE.0)GO TO 8610
      DO 8600 J=1,NOBJH
8600  CIMPL(J)=DJOLD(J)
      GO TO 870
8610  CALL MEVCTR(VCTSTR(DIMVCT+1),$8620,ATRGGR,GAUSS)                                232
870   CALL GETRHO(DF1)                                                   CY303400     235
      CALL PRMSUB($870)                                                  CY303600     237
      CALL MODDL                                                         CY303800     240
      GO TO 430                                                          CY304000
C
8620  WRITE (6,8622)                                                                  242
8622  FORMAT(10X,59HUNEXPLAINABLE RAY FAILURE WHILE ATTEMPTING TO REGENE
     1RATE EO)
      CALL GOOF                                                                       243
      RETURN 1
      END                                                                CY305000
```

SUBROUTINE TO CALCULATE REVISED ERROR VECTOR 09/24/68 PAGE
CYCLEZ STORAGE MAP

SUBROUTINE CYCLE
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
NSUBF	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	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	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
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	ATRGGR	06666	I
GAUSS	06667	R						

COMMON BLOCK PRNT ORIGIN 06672 LENGTH 00002

LINE	00000	I	PAGE	00001	I

COMMON BLOCK THDESN ORIGIN 06674 LENGTH 13777

PERIOD	00000	I	PCOUNT	00001	I	NOFFND	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
VCTSTR	00175	R	VINDX	13755	I	DJOLD	13770	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYP
DMATRX	22673	R	AMATRX	22705	R	S1	23551	R
S2	23107	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYP
JMAX	23121	I	IMAX	23122	I	J	23123	I
I	23124	I	KZ1	23125	I	K	23126	I
KZE	23127	I	KZI	23130	I	IFLAG	23131	I
KCX	23132	I	BF1	23133	R			

CYCLE2

STORAGE MAP

ENTRY POINTS

CYCLE	SECTION	9

SUBROUTINES CALLED

INCRM	SECTION	10	MEVCTR	SECTION	11	DOTP	SECTION	12
MATINV	SECTION	13	DFRNT	SECTION	14	.FWRD.	SECTION	1
JURY	SECTION	16	BNDCHK	SECTION	17	SQRT	SECTION	1
EFRNT	SECTIC	19	MODDL	SECTION	20	GETRND	SECTION	2
FRMSUB	SECTION	22	GODF	SECTION	23	.UN06.	SECTION	2
.FFIL.	SECTION	25	.FCNV.	SECTION	26	SYSLOC	SECTION	2
.FRET.	SECTION	28						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIO
50	7A	23261	170	35A	23374	470	109A	23646
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.

```
                    INF.M010001.44800-0.77387.AM IC      IBJOB LIDNK1  7094 A    02/16/58          PAGE  7

$0
$*ITLE         SUBROUTINE TO CALCULATE VECTOR INNER PRODUCT
$IBFTC DOTPZ   M94,XR7              LINK 5    (DOTP)                    00001060
```

```
SUBROUTINE TO CALCULATE VECTOR INNER PRODUCT                          02/16/68           PAGE  1
   DOTP2        - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  DOTP( VCT1, VCT2, PROD )                                 DO001200
C
C     *** INPUTS ***                                                       DO001600
C VCT1    FIRST VECTOR OF DIMENSION DIMVCT.                                DO001610
C VCT2    SECOND VECTOR OF DIMENSION DIMVCT.                               DO001620
C DIMVCT DIMENSION OF VECTORS DEFINING INNER PRODUCT.    (DATA COMMON) DO001630
C
C     *** OUTPUTS ***                                                      DO001640
C PROD    RESULTANT INNER PRODUCT OF VCT1 AND VCT2.                        DO001650
C
C     *** DESCRIPTION ***                                                  DO001660
C         DOTP CALCULATES THE INNER PRODUCT OF VECTORS VCT1 AND VCT2       DO001670
C         AND STORES THE RESULT IN PROD.                                   DO001680
C
      COMMON / THDESN /  PERIOD, PCOUNT, NOFFND, JAIL(10,10), SENTN(10),L7001000
     1 DIMVCT, CORD(8), 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  VCT1(1), VCT2(1)                                          DO002000
C
C             START OF SUBROUTINE DOTP                                     DO003000
C
      PROD = 0.                                                            DO003200
      DO  1000  I = 1, DIMVCT                                              DO003400
 1000 PROD = PROD + VCT1(I) * VCT2(I)                                      DO003600
      RETURN                                                               DO004000
      END                                                                  DO004200
```

SUBROUTINE TO CALCULATE VECTOR INNER PRODUCT 02/16/68 PAGE 2

DOTP Z STORAGE MAP

SUBROUTINE DOTP

COMMON VARIABLES

COMMON BLOCK THDESN 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	DIMVCT	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			

ENTRY POINTS

DOTP SECTION 5

SUBROUTINES CALLED

SYSLOC SECTION 6

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	5A	14002						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 14043.

```
                    INF.N010001.44800-0.77387.AM IC     IBJOB L1ONK1  7094 A    02/16/68          PAGE  3

$*
$TITLE          SUBROUTINE TO PRINT OUT DESIGN INFORMATION
$IBFTC DPRNTZ  M94.XR7              LINK 5    (DPRNT.EPRNT)                DP001000
```

```
      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, CONTRL(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), 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), 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), 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), 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
      COMMON/PRNT/LINE,PAGE                                              PT002500
      INTEGER PAGE                                                       PT002700
C
      COMMON / THDESN /  PERIOD, PCOUNT, NOFFND, JAIL(10,10), SENTN(10),L7001000
     1 DINVCT, CORD(6), SIZE, DETRM, EOLNG, ESLNG, IMPRV, VCTSTR(6000), L7002000
     2 VINDX(11)                                                         L7002100
      REAL  IMPRV                                                        L7003000
      INTEGER  PERIOD, PCOUNT, SENTN, DINVCT, CORD, VINDX                L7004000
C
      DIMENSION  VLST(6), FMT(8), FMTT(6)                                DP001500
C
      DATA  FMT(1)/6H(1HOD/, FMT(2)/6HESIGN /, FMT(3)/6HVARIAB/,        DP001600
     1 FMT(4)/6HLE,    /, FMT(6)/6H6X9HPA/, FMT(7)/6HRAMETE/,            DP001700
     2 FMT(8)/6HR,I2))/                                                  DP001800
      DATA  FMTT / 2H1(, 2H2(, 2H3(, 2H4(, 2H5(, 2H6( /                 DP001900
C
C            START OF DPRNT SUBROUTINE                                   DP002000
      IF( LINE .LE. 58 )  GO TO 1000                                     DP002200
C            BEGIN NEW PAGE                                              DP003000
```

```
      SUBROUTINE TO PRINT OUT DESIGN INFORMATION                          02/16/68                    PAGE  2
         DPRNTZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      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, DETRM, 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
C              BEGIN NEW PAGE                                            DP101600
      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, 32060 )  JI, ( VLST(J), J=1,NJ )                        DP105600    54
32060 FORMAT( 6XI3,9X6E17.8 )                                            DP106000
 5000 CONTINUE                                                           DP106200
C
      RETURN                                                             DP109000
      END                                                                DP109200
```

SUBROUTINE TO PRINT OUT DESIGN INFORMATION 02/16/68 PAGE 3
DPRNTZ STORAGE MAP

SUBROUTINE DPRNT
COMMON VARIABLES

COMMON BLOCK DATA ORIGIN 00001 LENGTH 06666

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	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
AFLAG	00047	I	DELY	00050	R	FNUMB	000[illegible]	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
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
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 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	NVARX	22706	I	NJMAX	22707	I
JI	22710	I	NJ	22711	I	KDX	22712	I

ENTRY POINTS

```
  SUBROUTINE TO PRINT OUT DESIGN INFORMATION                              02/16/68                        PAGE  4
     DPRNTZ                                          STORAGE  MAP

     DPRNT       SECTION     9                 EPRNT      SECTION    10
                                                     SUBROUTINES CALLED

     PRTCTL      SECTION    11                 .FWRD.     SECTION    12              .UNDS.     SECTION    13
     .FFIL.      SECTION    14                 .FCNV.     SECTION    15              SYSLOC     SECTION    16
                                               EFN   IFN  CORRESPONDENCE

EFN         IFN        LOCATION       EFN        IFN       LOCATION        EFN        IFN       LOCATION
1003        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.
```

```
$*
$TITLE          SUBROUTINE TO COMPUTE BACK FOCUS AND ENTR. PUPIL RADIUS
$IBFTC GETRHZ  M94,XR7              LINK 5    (GETRHO)                     GR001000
```

```
  SUBROUTINE TO COMPUTE BACK FOCUS AND ENTR. PUPIL RADIUS            02/16/68                         PAGE   1
    GETRHZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  GETRHO( BF1 )                                                GR001200
C
C     *** INPUTS ***                                                           GR0016C0
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         GETRHO COMPUTES THE BACK FOCUS FOR COLOR 1.  ALSO IF FNUMB .NE.GR001670
C         0. A REVISED EPRAD IS CALCULATED.                                    GR001680
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), 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), 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
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                        AZ004000
C
C            START OF GETRHO SUBROUTINE                                      GR002000
C
      THTR = 0.                                                              GR002400
      SNTHTR = 0.                                                            GR002600
      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
```

```
      CALL  HERID( 1, DUMMY, FNUMBC )                                 GR004000    13
 2000 CONTINUE                                                        GR004500
C
32000 RETURN                                                          GR005000
      END                                                             GR006000
```

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	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NBPLN	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	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	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
HOBJH	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	08665	I			

COMMON BLOCK AZOBJ ORIGIN 08667 LENGTH 00003

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	17A	08750	2000	14A	08746			

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 08771.

```
                     INF.N010001.44800-0.77387.AM IC      IBJOB LIONKI  7094 A    02/16/68          PAGE  4

*
$TITLE        SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE
$IBFTC INCRMZ  M94,XR7          LINK 5    (INCRM)                    INC01000
```

```
      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 PARAMENTERS 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), 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), 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), 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), 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 INCRM SUBROUTINE                                IN002400
C
      NJ = IDESGN(1,J)                                                  IN003000
      IF( NJ .NE. 0 )  GO TO 1000                                       IN003200
      WRITE ( 6, 32010 )  J                                             IN003400    6
32010 FORMAT( 1H0, 9X, 15HDESIGN VARIABLE, I3, 23H IS INCORRECTLY DEFINEIN003600
     1D )                                                               IN003800
      CALL  GOOF                                                        IN004000    7
 1000 NJA = IABS( NJ )                                                  IN005000
      DELTA = FACTOR * DESGN(3,J)                                       IN005200
      DO  6000  I = 1, NJA                                              IN005400
```

```
      SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE                        02/16/68        PAGE  2
         INCRMZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      KIJ = IDESGN(I+3,J)                                                IN005600
      KIJA = IABS( KIJ )                                                 IN005800
      IF( NJ .LT. 0 )  GO TO 2500                                        IN006000
      IF( KIJ .LT. 0 )  GO TO 2000                                       IN006200
      DATA(KIJA) = DATA(KIJA) + DELTA                                    IN006400
      GO TO 6000                                                         IN006600
 2000 DATA(KIJA) = DATA(KIJA) - DELTA                                    IN006800
      GO TO 6000                                                         IN007000
 2500 IF( DATA(KIJA) .NE. 0. )  GO TO 3500                               IN007200
      IF( DELTA .EQ. 0. )  GO TO 6000                                    IN007400
      IF( KIJ .LT. 0 )  GO TO 3000                                       IN007600
      DATA(KIJA) = 1. / DELTA                                            IN007800
      GO TO 6000                                                         IN008000
 3000 DATA(KIJA) = - 1. / DELTA                                          IN008200
      GO TO 6000                                                         IN008400
 3500 IF( KIJ .LT. 0 )  GO TO 4000                                       IN101000
      TMP = 1. / DATA(KIJA) + DELTA                                      IN101200
      GO TO 4500                                                         IN101400
 4000 TMP = 1. / DATA(KIJA) - DELTA                                      IN101600
 4500 IF( TMP .NE. 0. )  GO TO 5000                                      IN101800
      DATA(KIJA) = 0.                                                    IN102000
      GO TO 6000                                                         IN102200
 5000 DATA(KIJA) = 1. / TMP                                              IN102400
 6000 CONTINUE                                                           IN103000
C
      RETURN                                                             IN104000
      END                                                                IN105000
```

SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE 02/16/68 PAGE 3
INCRMZ STORAGE MAP

SUBROUTINE INCRM
COMMON VARIABLES

COMMON BLOCK DATA ORIGIN 00001 LENGTH 06666

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	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	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	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
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			

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	GOOF	SECTION	7	.UNDB.	SECTION	8
.FFIL.	SECTION	9	.FCNV.	SECTION	10	SYSLOC	SECTION	11

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	[illegible]	06746	32010	FORMAT	06707	6000	59A	07134
2500	[illegible]A	07027	2000	26A	07017	3500	45A	07067
3000	[illegible]A	07056	4000	50A	07105	4500	52A	07115
5000	57A	07126						

```
     SUBROUTINE TO INCREMENT CURRENT DESIGN VARIABLE                02/16/68                PAGE  4
        INCRMZ                                  STORAGE  MAP

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07164.
```

```
                    INF.N010001.44800-0.77387.AM IC      IBJOB L1OMK1  7094 A    02/16/68            PAGE  5

$*
$TITLE          SUBROUTINE TO TEST AND RESTRICT DESIGN VARIABLE COMBS.
$IBFTC JURYZ    M94,XR7             LINK 5    (JURY,JUDGE)               JU001000
```

```
      SUBROUTINE  JURY                                                  JU001500
C
C     *** INPUTS ***                                                    JU001600
C NDSGV   VECTOR OF DESIGN VARIABLE COMBINATION.          (DATA COMMON) JU001610
C NOFFND  NUMBER OF COMBINATIONS CURRENTLY JAILED.      (THDESN 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         ENTRY 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, CONTRL(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), 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), 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), HO    ), ( DATA(33),   DT102000
     B DELH  ), ( 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), 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
       COMMON / THDESN /  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 JURY SUBROUTINE                                JU002000
```

```
C
      IF( NJAIL .EQ. 0 )  RETURN                                        JU002200
      IF( NOFFND .EQ. 10 )  RETURN                                      JU002400
      NOFFND = HOFFND + 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. 02/16/68 PAGE 3
JURYZ STORAGE MAP

SUBROUTINE JURY
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	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
OLPLN	00080	R	OMGA2	00081	R	OMGA1	00082	R
DELD	00083	R	EPRAD	00084	R	FSCAL	00085	R
OMGAF	00086	R	SPFEA	00087	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
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	08210	R
IBNDS	08210	I	NCOND	08665	I			

COMMON BLOCK THDESN ORIGIN 08667 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	DIMVCT	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			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IMAX	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

SUBROUTINE TO TEST AND RESTRICT DESIGN VARIABLE COMBS. 02/16/68 PAGE 4

JURYZ STORAGE MAP

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1000	11A	22717	2500	36A	22771	2000	33A	22766
3000	39A	22774	4500	56A	23032	4000	52A	23026

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 23076.

```
                 $INF,N010001,44903-0,77387,AM IC      IDJOB LIGMKC  7094 A    09/24/68          PAGE  5

$*
$TITLE         SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR
$IBFTC MEVCTZ  M94,XR7             LINK 5   (MEVCTR)                  EV001000
```

```
      SUBROUTINE  MEVCTR( EVCTOR,*,KFLAG1,KFLAG3 )                       EV001500
C
C     *** OUTPUTS ***                                                    EV001600
C EVCTOR 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, CONTRL(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), 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), NRAYS ), ( DATA(3),    DT101000
     1 NSLCS ), ( DATA(4), NCLRS ), ( DATA(5), NJAIL ), ( DATA(6),      DT101100
     2 NSUBT ), ( DATA(7), NSUBF ), ( 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), 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 FSCAL ), ( 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), 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
      EQUIVALENCE  ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )        DT103000
      INTEGER  CONTRL, AFLAG, DATE, ATRGGR                              DT201000
      REAL  LATTC                                                        DT202000
C
      COMMON / TMPATT / NMISS, NRFLCT, NVIGN, NRAYSD, 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
      DIMENSION  MXCURL(200), MYCURL(200)                                L3001600
      EQUIVALENCE  ( XCURL(1), MXCURL(1) ), ( YCURL(1), MYCURL(1) )     L3001800
      REAL  MXBAR, KXBAR, MYBAR, KYBAR, MXCURL, MYCURL                  L3002000
C
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                    AZ004000
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
```

```
      SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR                          09/24/68                 PAGE
          MEVCTR       - EFN    SOURCE STATEMENT  -  IFN(S)  -

      INTEGER  PERIOD, PCOUNT, SENTN, DIMVCT, CORD, VINDX              LT004000
C
      DIMENSION  QVCT(3), XVCT(3), EVCTOR(1), XIDAR(6), YIDAR(6)        EV002000
     1 ,TEMPS(400), KXCURL(200), KYCURL(200)                           EV002200
      EQUIVALENCE  ( TEMPS(1), KXCURL(1) ), ( TEMPS(201), KYCURL(1) )   EV002300
      INTEGER  SMISS                                                    EV002400
      REAL  KXCURL, KYCURL, MXSUM, KXSUM, MYSUM, KYSUM, LAMDA            EV002600
C
C              START OF SUBROUTINE MEVCTR                               EV003000
C
      L = 0                                                             EV003200
  200 IF  (KFLAGS .NE. 0 )  LAMDA = 2. * GAUSS ** 2                     EV003300
      CALL  GETRHO(DF1)                                                 EV003400    5
      IF( WF.GH .EQ. 0. ) GO TO 1000                                    EV003600
      THTR   = 3.1415926                                                EV003800
      SNTHTR = 0.                                                       EV004000
      CSTHTR = -1.                                                      EV004200
      HEIGHT = EPRAD * OMGA2                                            EV004400
      XOHAT  =  0.                                                      EV004600
      YOHAT  = OMGA2                                                    EV005000
C
C     TRACE RAY                                                         EV005200
      CALL RAYTR ( XOHAT, YOHAT, HEIGHT, 1, XVCT, QVCT, SMISS, $5000,   EV005400
     1 $5000, $500,  0, DUMMY, DUMMY )                                  EV005600    10
  500 IF ( QVCT(3) .EQ. 0.)  GO TO 5000                                 EV005800
      XOHAT = QVCT(2  / ( QVCT(3) * HEIGHT )                            EV006000
      IF ( SURFC(5,NSURF+1) .GT. 0.)  XOHAT = -XOHAT                    EV006200
      IF ( FLNGH .NE. 0. )   XOHAT = XOHAT - (1. / FLNGH )              EV006400
      L = L + 1                                                         EV006600
      EVCTOR(L) = VCTSTR(L) * XOHAT                                     EV006800
 1000 IF ( WEXFP .EQ. 0. )  GO TO 1500                                  EV007000
      THTR = 0.                                                         EV007200
      SNTHTR = 0.                                                       EV007400
      CSTHTR = 1.                                                       EV007600
      HEIGHT = HEXFP                                                    EV007800
      XOHAT = 0.                                                        EV008000
      YOHAT = 0.                                                        EV008200
C
C              TRACE RAY                                                EV201000
      CALL RAYTR ( XOHAT, YOHAT, HEIGHT, 1, XVCT, QVCT, SMISS, $5000,   EV201200
     1$5000, $1200, 0, DUMMY, DUMMY )                                   EV201400    26
 1200 IF ( QVCT(3) .EQ. 0. ) GO TO 5000                                 EV201600
      XOHAT = XVCT(2) + ( QVCT(2) *( DEXPP - XVCT(3) ) ) / QVCT(3)      EV201800
      L=L+1                                                             EV202000
      EVCTOR(L) = VCTSTR(L) * XOHAT                                     EV202200
 1500 THTR = ROTAN * 3.1415926 / 180.                                   EV202400
      SNTHTR = SIN(THTR)                                                EV202600    35
      CSTHTR = COS(THTR)                                                EV202800    36
      MRAYS = NRAYS                                                     EV203000
      IF ( LMODE .EQ. 0 )  MRAYS = 2 * MRAYS                            EV203200
      XNCLRS = NCLRS                                                    EV203400
      SSTAR = DELD                                                      EV203500
      IF  ( IMODE .EQ. 0 )  SSTAR = SSTAR + DF1                         EV203520
      DO  4300  J = 1, NOBJH                                            EV203600
      HEIGHT = HO + FLOAT( J-1 ) * DELH                                 EV203800
      IF  ( KFLAG1 .EQ. 0 )  GO TO 1800                                 EV203820
```

```
      SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR                        09/24/68                    PAGE
         MEVCTZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

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
     1 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
     1 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
```

```
      IF ( NRAYSD .NE. MRAYS ) GO TO 5000                               EV205538
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(I) = XBAR                                                    EV206600
 2500 YIBAR(I) = 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 XBHAT = XBAR * SNTHTR + YBAR * CSTHTR                             EV402200
      L = L + 1                                                          EV402400
 4300 EVCTOR(L) = VCTSTR(L) * ( XBHAT - 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, I5,29H EV403400
     1DOES NOT AGREE WITH DIMVCT =, I5)                                  EV403600
      CALL GOOF                                                          EV403800     178
 5000 CALL PRMSUB ($5000 )                                               EV404000
C                                                                                     180
      RETURN 1                                                           EV405000
      END                                                                EV406000
```

SUBROUTINE MEVCTR
COMMON VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
COMMON BLOCK	DATA		ORIGIN	00001		LENGTH	06671	
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
NSUBF	00041	I	NIFLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
NFLAG	00047	I	DELY	00050	R	FNUMB	00051	R
FLNGH	00052	R	WFLGH	00053	R	ZETA	00054	R
HEXPF	00055	R	DEXPF	00056	R	WEXPF	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
CDIST	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
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	ATRGGR	06666	I
GAUSS	06667	R						
COMMON BLOCK	THPATT		ORIGIN	06672		LENGTH	00651	
NMISS	00000	I	NRFLCT	00001	I	NVIGN	00002	I
NRAYSD	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
MXBAR	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	
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R
COMMON BLOCK	THDESN		ORIGIN	07546		LENGTH	13777	
PERIOD	00000	I	PCOUNT	00001	I	NOFFND	00002	I
JAIL	00003	I	SENTN	00147	I	DIMVCT	00161	I
CORD	00162	I	SIZE	00170	R	DETRM	00171	R
EDLNG	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	TY

SUBROUTINE TO COMPUTE WEIGHTED ERROR VECTOR 09/24/68 PAGE 6

MEVCTZ

STORAGE MAP

QVCT	24365	R	XVCT	24370	R	XIBAR	24373	R
YIBAR	24401	R	TEMPS	23545	R	RXCURL	23545	R
RYCURL	24055	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SMISS	24407	I	MXSUM	24410	R	RXSUM	24411	R
MYSUM	24412	R	RYSUM	24413	R	LAMDA	24414	R
L	24415	I	BF1	24416	R	HEIGHT	24417	R
XOHAT	24420	R	YOHAT	24421	R	DUMMY	24422	R
MRAYS	24423	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
BETA	24434	R	GAMA	24435	R			

ENTRY POINTS

MEVCTR	SECTION	11

SUBROUTINES CALLED

GETRHO	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
.FRDD.	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	[illegible]
.FCNV.	SECTION	33	E.1	SECTION	34	E.2	SECTION	[illegible]
E.3	SECTION	36	E.4	SECTION	37	CC.1	SECTION	[illegible]
CC.2	SECTION	39	CC.3	SECTION	40	CC.4	SECTION	[illegible]
SYSLOC	SECTION	42	.FRET.	SECTION	43			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIC
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	96A	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.

```
$*
$TITLE          SUBROUTINE TO COMPUTE DESIGN PARAMETER INCREMENTS
$IBFTC MODDLZ  M94,XR7              LINK 5    (MODDL)                M0001000
```

```
      SUBROUTINE MODDL                                                  MO001500
C
C     *** INPUTS ***                                                    MO001600
C NDSGV   VECTOR CONTAINING CURRENT DESIGN VARIABLE NUMBERS.(DATA COMMON)MO001610
C DESGN   MATRIX OF DESIGN VALUES.                        (DATA COMMON)MO001620
C
C     *** OUTPUTS ***                                                   MO001630
C DESGN   THE INCREMENT IN DESGN(3,I), FOR I A DESIGN VARIABLE NUMBER   MO001640
C         CONTAINED IN NDSGV.                             (DATA COMMON) MO001650
C
C     *** DESCRIPTION ***                                               MO001660
C         MODDLZ USES DESGN(I,J), I=2,3,10 TO CALCULATE A NEW INCREMENT MO001670
C         STORED IN DESGN(3,J).  THIS PROCESS IS REPEATED FOR EACH J    MO001680
C         DESIGN VARIABLE CONTAINED IN NDSGV.                           MO001690
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), 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), 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                                 MO002000
C
      IMAX = NDSGV(1)                                                   MO002200
      DO 500 I = 1, IMAX                                                MO002400
      IZ = NDSGV(I+1)                                                   MO002600
      IF ( DESGN(10,IZ) .NE. 0.)  DESGN(3,IZ) = ABS( DESGN(3,IZ) * SQRT(MO002800
     1 DESGN(2,IZ) / DESGN(10,IZ) ) )                                   MO003000      12
  500 CONTINUE                                                          MO003200
      RETURN                                                            MO004000
      END                                                               MO004200
```

SUBROUTINE TO COMPUTE DESIGN PARAMETER INCREMENTS 02/16/68 PAGE 2
MODDLZ STORAGE MAP

SUBROUTINE MODDL
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	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	PSCAL	00065	R
OMGAF	00066	R	SPFEA	00067	R	DUMIN	00070	R
CDIST	00071	R	HD	00072	R	DELH	00073	R
SYSNX	00074	R	WXDIR	00075	R	WYDIR	00076	R
ROTAN	00077	R	NDSGN	00100	I	NDSGV	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	08210	R
IBNDS	08210	I	NCOND	08665	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IMAX	08667	I	IZ	08670	I			

ENTRY POINTS

MODDL SECTION 5

SUBROUTINES CALLED

SQRT 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.

```
                              .N010001.44800-0.77387.AM IC      IBJOB L1DMK1  7094 A    02/16/68          PAGE  3

$*
$TITLE          SUBROUTINE TO GENERATE WEIGHT VECTOR
$IBFTC HWVCTZ  M94,XR7              LINK 5    (HWVCTR)                    WV001000
```

```
      SUBROUTINE HWVCTR                                                  WV001200
C
C     *** INPUTS ***                                                     WV001600
C         VARIOUS DESIGN WEIGHTS.                      (DATA COMMON) WV001610
C
C     *** OUTPUTS ***                                                    WV001620
C VCTSTR WEIGHT VECTOR OF LENGTH DIMVCT.             (THDESN COMMON) WV001630
C
C     *** DESCRIPTION ***                                                WV001640
C         HWVCTR 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, CONTRL(10), TITLE(12), DATE(3), PUNCID,   DT001000
     1 DATA(3483)                                                        DT001100
      DIMENSION  WOBJH(7), WCLRH(7), WIHHT(7), EIHHT(7), CIHPL(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), 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), 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), WCLRH(1) ),       DT102300
     E ( DATA(67), WIHHT(1) ), (DATA(74), EIHHT(1) ), ( DATA(81),       DT102400
     F CIHPL(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
      COMMON / THDESN /  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 HWVCTR                                           WV002000
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
```

```
      VCTSTR(L) = WEXPP                                                 WV003400
 2500 NRAYS = NRAYS                                                     WV003600
      IF( LMODE .EQ. 0 )  NRAYS = NRAYS + NRAYS                         WV003800
      ASQRT = SQRT( FLOAT( NRAYS ) )                                    WV004000     16
      BSQRT = SQRT( FLOAT( NCLRS ) )                                    WV004200     17
      IMAX  = NCLRS                                                     WV004400
      IF( LMODE .NE. 1 )  IMAX = IMAX + IMAX                            WV004600
C
C            BEGIN DO LOOP ON OBJECT HEIGHTS                            WV005000
      DO  4000  J = 1, NOBJH                                            WV005200
      TX = WOBJH(J) * WXDIR / ASQRT                                     WV005400
      TY = WOBJH(J) * WYDIR / ASQRT                                     WV005600
C
C            BEGIN DO LOOP ON NUMBER OF COLORS                          WV006000
      DO  3000  I = 1, NCLRS                                            WV006200
      TXX = WCLRS(I) * TX                                               WV006400
      TYY = WCLRS(I) * TY                                               WV006600
C
C            BEGIN DO LOOP ON RAY NUMBER                                WV006800
      DO  3000  K = 1, NRAYS                                            WV007000
      L = L + 1                                                         WV007200
      VCTSTR(L) = TXX                                                   WV007400
      L = L + 1                                                         WV007600
      VCTSTR(L) = TYY                                                   WV007800
 3000 CONTINUE                                                          WV008000
      TXX = WCLRH(J) / BSQRT                                            WV008200
      DO  3500  I = 1, IMAX                                             WV008400
      L = L + 1                                                         WV008600
 3500 VCTSTR(L) = TXX                                                   WV008800
      L = L + 1                                                         WV009000
      VCTSTR(L) = WIMHT(J)                                              WV009200
 4000 CONTINUE                                                          WV009400
C
      CALL  PRMSUB( $1000 )                                             WV101000     51
      IF( L .EQ. DIMVCT )  GO TO 32000                                  WV101200
C
C             ERROR  -  TERMINATE RUN                                   WV102000
      WRITE ( 6, 32100 )  L, DIMVCT                                     WV102200     56
32100 FORMAT( 1H0, 37HNUMBER OF COMPONENTS IN WEIGHT VECTOR, I5, 29H DOEWV102400
     1S NOT AGREE WITH DIMVCT =, I5 )                                   WV102600
      CALL  GOOF                                                        WV103000
C
C             END OF HWVCTR SUBROUTINE                                  WV105800     57
32000 RETURN                                                            WV106000
      END                                                               WV107000
```

SUBROUTINE TO GENERATE WEIGHT VECTOR 02/16/68 PAGE 3

HWVCTZ STORAGE MAP

SUBROUTINE HWVCTR

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
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
IFLAG	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
DELO	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
HOBJH	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 THDESN ORIGIN 06667 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	DIMVCT	00161	I
CORD	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
L	22657	I	MRAYS	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	TYY	22671	R			

ENTRY POINTS

HWVCTR	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                             02/16/68                    PAGE  4
       HWVCTZ                                    STORAGE  MAP

     .FCNV.      SECTION    14          E.1        SECTION    15          E.2       SECTION    16
     E.3         SECTION    17          E.4        SECTION    18          CC.1      SECTION    19
     CC.2        SECTION    20          CC.3       SECTION    21          CC.4      SECTION    22
     SYSLOC      SECTION    23
                                        EFN    IFN    CORRESPONDENCE

EFN        IFN        LOCATION       EFN        IFN        LOCATION       EFN        IFN        LOCATION
1000        2A         22723         2000        7A         22735         2500       12A         22747
4000       48A         23134         3000       36A         23102         3500       42A         23115
32000      58A         23163         32100     FORMAT       22702
 THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 23207.
```

```
$*
$TITLE          SUBROUTINE TO CONTROL DESIGN LOGIC
$IBFTC STAR2Z  M94,XR7              LINK 5    (STAR2)                S2001000
```

```
      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, CONTRL(10), TITLE(12), DATE(3), FUNCID,   DT001000
     1 DATA(3485)                                                       DT001100
      DIMENSION  WOBJH(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), 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), 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), 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
      EQUIVALENCE  ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )        DT103000
      INTEGER  CONTRL, AFLAG, DATE, ATRGGR                              DT201000
      REAL  LATTC                                                       DT202000
C
      COMMON/PRNT/LINE,PAGE                                             PT002500
      INTEGER PAGE                                                      PT002700
C
      COMMON / TMDESN /  PERIOD, PCOUNT, NOFFND, JAIL(10,10), SENTN(10),L7001000
     1 DIMVCT, CORD(6), SIZE, DETRM, EOLNG, 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
         STAR22       - EFN   SOURCE STATEMENT  -  IFN(S)  -

      WRITE (6,600)                                                     $2002400   6
600   FORMAT(22H1BAD DESIGN DATA INPUT)                                 $2002600
      CALL GOOF                                                         $2002800   7
40    DIMVCT=0                                                          $2003000
50    MDIM=0                                                            $2003200
      MRAYS=NRAYS                                                       $2003400
      IF(LMODE.EQ.0)MRAYS=MRAYS+MRAYS                                   $2003500
      MDIM=2*MRAYS*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(8H1DIMVCT=I10,28H 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
      NOFFND=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                                    $2008650
      DO 230 II = 1, NOBJH                                              $2008700
  230 DJOLD(II)=CIMPL(II)                                               $2008750
      GO TO 270                                                         $2008800
260   WRITE (6,602)                                                     $2021000   73
602   FORMAT(1H136HRAY FAILURE DURING COMPUTATION OF ED)                $2021200
      CALL GOOF                                                         $2021400   74
270   CALL DOTP(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 PRTCTL                                                       $2022200   95
      WRITE (6,603)                                                     $2022400   96
603   FORMAT(1H03X23HDESIGN VARIABLE NUMBERS6X12HINCR. FACTOR3X11HDETERM$2022600
     1INANT5X10HOLD LENGTH5X10HNEW 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(I+1).LT.NDSGN(3))GO TO 380                               $2024000
      NDSGV(I+1)=NDSGN(2)                                               $2024200
```



```
      SUBROUTINE TO CONTROL DESIGN LOGIC                                09/24/68                PAGE
         STAR22       - EFN   SOURCE STATEMENT  -  IFN(S)  -

      GO TO 400                                                         $2024400
380   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
```

```
      GO TO 400                                              $2024400
380   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 09/24/68 PAGE

STAR2Z STORAGE MAP

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	FUNCID	00032	R	DATA	00033	R
LMODE	00033	I	NRAYS	00034	I	NSLCS	00035	I
NCLRS	00036	I	NJAIL	00037	I	NSUBT	00040	I
NSUBF	00041	I	NIPLN	00042	I	IMODE	00043	I
NSFLN	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	FSCAL	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	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	ATRGGR	06666	I
GAUSS	06667	R						

COMMON BLOCK PRNT ORIGIN 06672 LENGTH 00002

LINE	00000	I	PAGE	00001	I

COMMON BLOCK TMDESN ORIGIN 06674 LENGTH 13777

PERIOD	00000	I	PCOUNT	00001	I	NOFFND	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 09/24/68 PAGE 5

STAR22 STORAGE MAP

.FWRD.	SECTION	10	GOOF	SECTION	11	PRMSUB	SECTION	1[illegible]
MWVCTR	SECTION	13	MEVCTR	SECTION	14	DOTF	SECTION	1[illegible]
SORT	SECTION	16	CORDER	SECTION	17	PRTCTL	SECTION	1[illegible]
JUDGE	SECTION	19	CYCLE	SECTION	20	.UNGS.	SECTION	2[illegible]
.FFIL.	SECTION	22	.FCNV.	SECTION	23	SYSLOC	SECTION	2[illegible]

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIO[illegible]
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
380	114A	23335						

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 23373.

```
                    INF.N010001.44800-0.77387.AM IC      IBJOB LINKS  7094 A    02/16/68          PAGE  6

$*
$TITLE        JPL MATRIX INVERSION LIBRARY SUBROUTINE
$IBFTC MVRTS   M94,XR7              LINK 5   (MATINV)
```

```
      SUBROUTINE MATINV (NDIM,A,NORDER,B,MORDER,DETERM,S1,S2)
C
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
      DIMENSION A(NDIM,NDIM),B(NDIM,2)
C
C     PIVOT=S2
C     IPIVOT=S1
C     INDEX  (N,1) =S1((N+1)-(2N))
      DIMENSION S1(2),S2(2)
      INTEGER Z1,Z2,S1
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (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 550 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 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
110   S1(ICOLUM)=S1(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
130   IF (IROW.EQ.ICOLUM) GO TO 260
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,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(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
260   S1(Z1)=IROW
270   S1(Z2)=ICOLUM
310   S2(I)=A(ICOLUM,ICOLUM)
```

```
320    DETERM=DETERM*S2(I)
C
C      DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
350    A(ICOLUM,L)=A(ICOLUM,L)/S2(I)
355    IF (M.LE.0) GO TO 380
  360 DO 370 L=1,M
370    B(ICOLUM,L)=B(ICOLUM,L)/S2(I)
C
C      REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
390    IF (L1.EQ.ICOLUM) GO TO 550
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
455    IF (M.LE.0) GO TO 550
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,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 (S1(L1).EQ.S1(L2)) GO TO 710
630    JROW=S1(L1)
640    JCOLUM=S1(L2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
```

JPL MATRIX INVERSION LIBRARY SUBROUTINE 02/10/63 PAGE 3
NVRT1 STORAGE MAP

SUBROUTINE MATINV
UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
Z1	00004	I	Z2	00005	I	IROW	00001	I
JROW	00001	I	ICOLUM	00002	I	JCOLUM	00002	I
AMAX	00003	R	T	00003	R	SWAP	00003	
N	00006	I	M	00007	I	J	00010	I
I	00011	I	K	00012	I	L	00013	I
L1	00014	I	L2	00015	I			

ENTRY POINTS

MATINV SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	1A	00031	15	3A	00037	20	7A	00043
30	11A	00046	550	133A	00465	31	15A	00125
32	16A	00130	40	17A	00133	45	18A	00134
105	40A	00203	50	21A	00137	60	25A	00143
100	38A	00200	70	28A	00153	80	31A	00157
740	168A	00671	85	34A	00172	90	35A	00174
95	36A	00176	110	42A	00205	130	45A	00214
180	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	76A	00302
310	78A	00307	320	81A	00317	330	83A	00322
340	85A	00332	350	88A	00343	355	93A	00351
380	104A	00375	360	96A	00356	370	99A	00367
390	107A	00415	400	110A	00421	420	112A	00423
430	114A	00424	450	117A	00432	455	122A	00442
460	125A	00447	500	128A	00455	600	137A	00471
710	165A	00567	610	141A	00502	611	142A	00505
612	143A	00511	620	144A	00514	630	149A	00526
640	151A	00533	650	153A	00540	705	163A	00565
660	156A	00557	670	158A	00561	700	161A	00563

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 00753.

Section 6

LINK 6

```
$*
$TITLE        SUBROUTINE TO COMPUTE FOCAL LENGTH AND FOCAL POINT
$IBFTC PARAXZ  M94,XR7              LINK 6    (PARAX)                PR001000
```

```
      SUBROUTINE  PARAX( COLOR, FOCALL, FOCALP )                        PR001100
C
C    *** INPUTS ***                                                     PR001200
C COLOR  NUMBER OF COLOR TO BE USED.                                    PR001210
C
C    *** OUTPUTS ***                                                    PR001300
C FOCALL FOCAL LENGTH VALUE.                                            PR001310
C FOCALP FOCAL POINT VALUE.                                             PR001320
C
C    *** DESCRIPTION ***                                                PR001400
C        THIS SUBROUTINE COMPUTES THE FOCAL LENGTH AND FOCAL POINT      PR001410
C        FOR THE SPECIFIED COLOR.  IF THEY ARE INFINITE, THEN FOCALL    PR001420
C        AND FOCALP ARE SET TO ZERO.                                    PR001430
C
      COMMON  / DATA / DUMMY1(27), DATA(3483)                           PR001500
      EQUIVALENCE  ( DATA(23), OMGA2 ), ( DATA(26), EPRAD ), ( DATA(12),PR001700
     1 NSURF ), ( DATA(182), SURFC(1,1) )                               PR001800
      DIMENSION  SURFC(20,100)                                          PR002000
C
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                   AZ004000
C
      INTEGER  COLOR                                                    PR002000
      DIMENSION  XVCT(3), QVCT(3)                                       PR002200
C
C              START OF PARAX SUBROUTINE                                PR003000
C
      H = EPRAD * OMGA2                                                 PR003500
      FOCALL = 0.0                                                      PR003700
      FOCALP = 0.0                                                      PR003900
      CALL  RAYTR( 0., OMGA2, H, COLOR, XVCT, QVCT, SMISS, $400, $500,  PR004000
     1 $600, 0, DUMMY, DUMMY )                                          PR004100    2
      GO TO 600                                                         PR004500
  400 WRITE ( 6, 32000 )  SMISS                                         PR005000    5
32000 FORMAT( 1H0, 27HPARAXIAL RAY MISSED SURFACE, I4 )                 PR005200
      CALL  GOOF                                                        PR005400    6
  500 WRITE ( 6, 32010 )  SMISS                                         PR006000    7
32010 FORMAT( 1H0, 33HPARAXIAL RAY REFLECTED AT SURFACE, I4 )           PR006200
      CALL GOOF                                                         PR006400    8
  600 IF( QVCT(2) .EQ. 0.0 )  GO TO 1000                                PR007000
      ALPHA = - QVCT(3) / QVCT(2)                                       PR007200
      FOCALP = ALPHA * XVCT(2) + XVCT(3)                                PR007400
      FOCALL = ALPHA * H                                                PR007600
      IF( SURFC( COLOR+4,NSURF+1 ) .LE. 0.0 )  FOCALL = - FOCALL        PR008000
C
 1000 RETURN                                                            PR009000
      END                                                               PR009200
```

```
SUBROUTINE TO COMPUTE FOCAL LENGTH AND FOCAL POINT                    08/16/68                         PAGE  2
    PARAX2                                              STORAGE  MAP

                                                  SUBROUTINE  PARAX
                                                       COMMON   VARIABLES

                COMMON BLOCK      DATA              ORIGIN      00001               LENGTH      06666

SYMBOL      LOCATION     TYPE          SYMBOL      LOCATION     TYPE                SYMBOL      LOCATION     TYPE
DUMMY1       00000        R            DATA         00033        R                  OMGA2        00081        R
EPRAD        00084        R            NSURF        00046        I                  SURFC        00320        R

                COMMON BLOCK      AZOBJ             ORIGIN      06667               LENGTH      00003
THTR         00000        R            SNTHTR       00001        R                  CSTHTR       00002        R
                                                     DIMENSIONED PROGRAM VARIABLES

SYMBOL      LOCATION     TYPE          SYMBOL      LOCATION     TYPE                SYMBOL      LOCATION     TYPE
XVCT         06672        R            QVCT         06675        R
                                                   UNDIMENSIONED PROGRAM VARIABLES

SYMBOL      LOCATION     TYPE          SYMBOL      LOCATION     TYPE                SYMBOL      LOCATION     TYPE
H            06700        R            SMISS        06701        R                  DUMMY        06702        R
ALPHA        06703        R
                                                      ENTRY POINTS

     PARAX        SECTION      7

                                                      SUBROUTINES CALLED

     RAYTR        SECTION      8               .FWRD.      SECTION      9                 GOOF        SECTION      10
     .UNDS.       SECTION      11              .FFIL.      SECTION      12                .FCNV.      SECTION      13
     SYSLOC       SECTION      14
                                               EFN    IFN    CORRESPONDENCE

EFN          IFN          LOCATION             EFN          IFN          LOCATION         EFN          IFN          LOCATION
400          5A           06763                500          7A           06776            600          9A           07011
32000        FORMAT       06714                32010        FORMAT       06724            1000         15A          07043

 THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 07075.
```

```
                     $NF,N01000,44903-0,77387,AM IC      $BJOB L1DMKC  7094 A   09/24/68        PAGE

$)
$TITLE        SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS
$IBFTC STARSZ  M94,XR7              LINK 6   (STARS)                    $3001000
```

```
        SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS                 09/24/68           PAGE
             STARSZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

       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 FLANES AND FOR THREE SPECIAL ONES CHOSEN TO MINIMIZE S3001530
C         RMS X, RMS Y, AND SPOT SIZE.                                     S3001540
C
       COMMON / DATA / NCNTRL, CONTRL(10), TITLE(12), DATE(3), FUNCID,     DT001000
      1 DATA(3485)                                                          DT001100
       DIMENSION  WOBJH(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), 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), 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), 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
       EQUIVALENCE  ( DATA(3484), ATRGGR ), ( DATA(3485), GAUSS )          DT103000
       INTEGER  CONTRL, AFLAG, DATE, ATRGGR                                DT201000
       REAL  LATTC                                                          DT202000
C
       COMMON / TMPATT / NMISS, NRFLCT, NVIGN, NRAYSD, 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                                     AZ004000
C
       COMMON/PRNT/LINE,PAGE                                               PT002500
       INTEGER PAGE                                                        PT002700
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 ,I /, FMT(2) / 6H3,2X,  /,  FMT(7)/6H  I2 ) /,S3003000
      1 EFRMT / 6HE16.7, /, AFRMT / 6H4X2A6, /, FINTY1 / 6H INFIN /,       S3003200
      2 FINTY2 / 6HITY    /                                                S3003400
C
```

```
      REAL  LAMDAX, LAMDAY, LAMDAR, LAMDAD, LAMDA, LAMDAT                 S3004000
      INTEGER  SMISS                                                      S3005000
      EQUIVALENCE (MXCURL(1),XCURL(1)),(KXCURL(1),TEMPS(1)),              S3005010
     1            (MYCURL(1),YCURL(1)),(KYCURL(1),TEMPS(201))             S3005020
      REAL MXCURL(200),KXCURL(200),MYCURL(200),KYCURL(200),XCORD(200),    S3005030
     1     YCORD(200)                                                     S3005040
C                                                                         
C              START OF STAR3 SUBROUTINE                                  S3007000
C                                                                         
      CALL  PRTCTL                                                        S3007500   2
      WRITE ( 6, 32010 )                                                  S3008000   3
32010 FORMAT( 1H0, 5HCOLOR, 3X, 12HFOCAL LENGTH, 5X, 11HFOCAL POINT, 5X,S3008200
     1 10HBACK FOCUS, 7X, 9HF/ NUMBER )                                   S3008400
      LINE = LINE + 2                                                     S3008600
      SNTHTR = 0.0                                                        S3008800
      DO 7000  I = 1, NCLRS                                               S3101000
      THTR = 3.14159265                                                   S3101200
      CSTHTR = - 1.0                                                      S3101400
C                                                                         
C              COMPUTE FOCAL LENGTH AND FOCAL POINT                       S3102000
      CALL  PARAX( I, FOCALL, FOCALP )                                    S3102200   8
      THTR = 0.0                                                          S3102400
      CSTHTR = 1.0                                                        S3102600
C                                                                         
C              COMPUTE BACK FOCUS AND F NUMBER                            S3103000
      CALL  MERID( I, BFOCUS, FSNUMB )                                    S3103200   10
      IF( I .EQ. 1 )  BF1 = BFOCUS                                        S3103400
C                                                                         
C              GENERATE OUTPUT LIST AND FORMAT FOR FOCAL LENGHT, FOCAL   S3104000
C              POINT, BACK FOCUS, AND F/ NUMBER                           S3104200
C              IF ANY VALUE IS 0.0 THIS INDICATES INFINITY SHOULD BE      S3104400
C              PRINTED OUT                                                S3104600
      DO  2000  J = 3, 6                                                  S3105000
 2000 FMT(J) = EFRMT                                                      S3105200
      ITEMP = 1                                                           S3105400
      IF( FOCALL .EQ. 0.0 )  GO TO 2500                                   S3105600
      OLIST(1) = FOCALL                                                   S3105800
      ITEMP = 2                                                           S3106000
      GO TO 3000                                                          S3106200
 2500 OLIST(1) = FINTY1                                                   S3106600
      OLIST(2) = FINTY2                                                   S3106800
      FMT(3) = AFRMT                                                      S3107000
      ITEMP = 3                                                           S3107200
 3000 IF( FOCALP .EQ. 0.0 )  GO TO 3500                                   S3107400
      OLIST(ITEMP) = FOCALP                                               S3107600
      ITEMP = ITEMP + 1                                                   S3107800
      GO TO 4000                                                          S3108000
 3500 OLIST(ITEMP) = FINTY1                                               S3108600
      OLIST(ITEMP+1) = FINTY2                                             S3108800
      FMT(4) = AFRMT                                                      S3109000
      ITEMP = ITEMP + 2                                                   S3109200
 4000 IF( BFOCUS .EQ. 0.0 )  GO TO 4500                                   S3201000
      OLIST(ITEMP) = BFOCUS                                               S3201200
      ITEMP = ITEMP + 1                                                   S3201400
      GO TO 5000                                                          S3201600
 4500 OLIST(ITEMP) = FINTY1                                               S3202000
```

```
      SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS                    09/24/68                PAGE
          STAR32       -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      OLIST(ITEMP+1) = FINTY2                                                $3202200
      FMT(5) = AFRMT                                                         $3202400
      ITEMP = ITEMP + 2                                                      $3202600
 5000 IF( FSNUMB .EQ. 0.0 )  GO TO 5500                                      $3203000
      OLIST(ITEMP) = FSNUMB                                                  $3203200
      GO TO 6000                                                             $3203400
 5500 OLIST(ITEMP) = FINTY1                                                  $3203800
      OLIST(ITEMP+1) = FINTY2                                                $3204000
      ITEMP = ITEMP + 1                                                      $3204200
      FMT(6) = AFRMT                                                         $3204400
C
C             FMT IS (1H ,I3,2X, AND E16.7 OF 4X2A6, DEPENDING ON            $3205000
C             WHETHER THE VALUES ARE ZERO OR NOT                             $3205200
C
C             PRINT OUT VALUES OR COMMENT INFINITY                           $3205400
 6000 WRITE ( 6, FMT )  I, ( OLIST(J), J=1,ITEMP )                           $3205600   50
      LINE = LINE + 1                                                        $3206000
 7000 CONTINUE                                                               $3207000
C
C             COMPUTE OBJECT HEIGHTS                                         $3208000
      DO  8000  J = 1, NOBJH                                                 $3208200
 8000 HEITS(J) = HO + FLOAT( J-1 ) * DELH                                    $3208400
      THTR = 0.0                                                             $3209000
      SNTHTR = 0.0                                                           $3209200
      CSTHTR = 1.0                                                           $3209400
C
C             PRINT HEADING FOR EXIT PUPIL                                   $3301000
      WRITE ( 6, 32020 )                                                     $3301200   65
32020 FORMAT( 1H0, 4X, 6HHEIGHT, 6X, 5HCOLOR, 4X, 10HEXIT PUPIL )            $3301400
      LINE = LINE + 2                                                        $3301600
      DO  13600  J = 1, NOBJH                                                $3302000
      IF( HEITS(J) .EQ. 0.0 )  GO TO 13600                                   $3302400
      DO  13500  I = 1, NCLRS                                                $3303000
      .ALL  RAYTR( 0., 0., HEITS(J), I, XVCT, QVCT, SMISS, $10200,           $3303400
     1 $10200, $10000, 0, DUMMY, DUMMY )                                     $3303600   76
10000 IF( QVCT(2) .NE. 0.0 )  GO TO 10500                                    $3304000
10200 EXITP = 0.0                                                            $3304200
      GO TO 11000                                                            $3304400
10500 EXITP = - QVCT(3) * XVCT(2) / QVCT(2) + XVCT(3)                        $3304600
11000 IF( LINE .LE. 54 )  GO TO 11500                                        $3305000
C
C             BEGIN NEW PAGE,  PRINT TITLE                                   $3305200
      CALL  PRTCTL                                                           $3305400   88
      WRITE ( 6, 32020 )                                                     $3305600   89
      LINE = LINE + 2                                                        $3305800
11500 IF( EXITP .NE. 0.0 )  GO TO 12000                                      $3306000
C             PRINT INFINITY FOR VALUE OF EXIT PUPIL                         $3306400
      WRITE ( 6, 32030 )  HEITS(J), I                                        $3306600   92
32030 FORMAT( 1H , E14.7, 4X, I1, 6X, 9H INFINITY )                          $3306800
      GO TO 13000                                                            $3307000
C
C             PRINT EXIT PUPIL VALUE                                         $3307600
12000 WRITE ( 6, 32040 )  HEITS(J), I, EXITP                                 $3307800   95
32040 FORMAT( 1H , E14.7, 4X, I1, E18.7 )                                    $3308000
13000 LINE = LINE + 1                                                        $3308200
13500 CONTINUE                                                               $3308400
```

```
      SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS            09/24/68             PAGE
          STAR32      - EFN   SOURCE STATEMENT  -  IFN(S)  -

13600  CONTINUE                                                        $3308500
      THTR = ROTAN * 3.14159265 / 180.                                 $3309000
      SNTHTR = SIN( THTR )                                             $3309200    103
      CSTHTR = COS( THTR )                                             $3309400    104
      MRAYS = NRAYS                                                    $3309600
      IF( LMODE .EQ. 0 )  MRAYS = 2 * MRAYS                            $3309800
C
      MTEMP = 0                                                        $3400500
      DO  20000  J = 1, NOBJH                                          $3401000
      DO  20000  I = 1, NCLRS                                          $3402000
C
      MTEMP = MTEMP + 1                                                $3402200
C             COMPUTE STATISTICAL INFORMATION                          $3402400
      CALL  SMPOP( I, J, HEITS(J), BF1, 1, TEMPS )                     $3402600    115
      CALL  PRTCTL                                                     $3402800    118
      WRITE ( 6, 32050 )  MTEMP, J, I                                  $3403000    119
32050 FORMAT( 1H0, //I3,14H. OBJECT POINT, I2, 1X, 6H COLOR, I2, 21H ...$3403200
     1.................. )                                             $3403400
C
C             PRINT OUT RAY TRACE INFORMATION                          $3404000
      WRITE ( 6, 32060 )  HEITS(J), THTR, MRAYS, NMISS, NRFLCT, NVIGN,  $3404200
     1 NRAYSB                                                          $3404400    120
32060 FORMAT( 1H0, 7X, 6HHEIGHT, 10X, 7HAZIMUTH / 2X, 2E16.7 // 6X,    $3405000
     1 52HRAYS TRACED  MISSED  REFLECTIONS  VIGNETS  SUCCESSES / 9X, I3,$3405200
     2 8X, I3, 8X, I3, 8X, I3, 7X, I3 )                                $3405400
      LINE = LINE + 10                                                 $3405600
C
C             PRINT OUT RAY STATISTICS                                 $3406000
      WRITE ( 6, 32070 )  MXBAR, KXBAR, MYBAR, KYBAR, AX, BX, CX, AY,  $3406200
     1 BY, CY, ACAP, BCAP, CCAP                                        $3406400    122
32070 FORMAT( 1H0, 8X, 5HMXBAR, 11X, 5HKXBAR, 11X, 5HMYBAR, 11X,       $3406800
     1 5HKYBAR / 2X, 4E16.7 // 10X, 2HAX, 14X, 2HBX, 14X, 2HCX / 2X,   $3407000
     2 3E16.7 // 10X, 2HAY, 14X, 2HBY, 14X, 2HCY / 2X, 3E16.7 // 11X,  $3407200
     3 1HA, 15X, 1HB, 15X, 1HC / 2X, 3E16.7 )                          $3407400
      LINE = LINE + 12                                                 $3407600
      WRITE ( 6, 32080 )                                               $3408000    123
32080 FORMAT( 1H0, 20X, 13HLOC. OF PLANE, 5X, 9HAVERAGE X, 7X, 9HAVERAGE$3408400
     1 Y, 5X, 12HIMAGE HEIGHT,8X,5HRMS X, 11X,5HRMS Y,9X,9HSPOT SIZE  ) $3408600
      LINE = LINE + 2                                                  $3408800
      LAMDAX = - BX / AX                                               $3409000
      LAMDAY = - BY / AY                                               $3409200
      LAMDAR = - BCAP / ACAP                                           $3409400
      IF( I .NE. 1 )  GO TO 14500                                      $3501000
      LAMDA0 = SSTAR + FLOAT( NSPLN ) * DLPLN                          $3501200
      DO  14000  K = 1, NIPLN                                          $3501600
14000 LAMDA( K ) = LAMDA0 + FLOAT( K-1 ) * DLPLN                       $3501800
14500 LAMDAT = LAMDAX                                                  $3502000
      KTEMP = 1                                                        $3502400
      FNR=NRAYSB                                                       $3502402
15000 XBAR = MXBAR * LAMDAT + KXBAR                                    $3502600
      YBAR = MYBAR * LAMDAT + KYBAR                                    $3502800
      HIMAGE = XBAR * SNTHTR + YBAR * CSTHTR                           $3503000
      XSTAR=0.                                                         $3503010
      YSTAR=0.                                                         $3503012
      DO 15010 K=1,NRAYSB                                              $3503014
      XCORD(K)=MXCURL(K)*LAMDAT+KXCURL(K)                              $3503016
```

```
          SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS                 09/24/68                  PAGE
               STARS2      - EFN   SOURCE STATEMENT  -  IFN(S)  -

15010 YCORD(K)=MYCURL(K)*LAMDAT+KYCURL(K)                            $3503018
      DO 15012 K=1,NRAYSD                                            $3503020
      XSTAR=XSTAR+XCORD(K)*XCORD(K)                                  $3503022
15012 YSTAR=YSTAR+YCORD(K)*YCORD(K)                                  $3503024
      XSTAR=SQRT(XSTAR/FNR)                                          $3503026    154
      YSTAR=SQRT(YSTAR/FNR)                                          $3503028    155
      RSTAR=SQRT(XSTAR*XSTAR+YSTAR*YSTAR)                            $3503030    156
      GO TO ( 15500, 16000, 16500 ),  KTEMP                          $3504000
C
C             PRINT MINIMUM RMS X                                    $3505000
15500 WRITE ( 6, 32090 ) LAMDAX, XBAR, YBAR, HIMAGE, XSTAR, YSTAR, RSTAR$3505200    158
32090 FORMAT( 1H , 3X, 10HMIN. RMS X, 4X, 7E16.7 )                   $3505400
      KTEMP = 2                                                      $3506000
      LAMDAT = LAMDAY                                                $3506200
      GO TO 15000                                                    $3506400
C
C             PRINT MINIMUM RMS Y                                    $3507000
16000 WRITE ( 6, 32100 ) LAMDAY, XBAR, YBAR, HIMAGE, XSTAR, YSTAR, RSTAR$3507200    161
32100 FORMAT( 1H , 3X, 10HMIN. RMS Y, 4X, 7E16.7 )                   $3507400
      KTEMP = 3                                                      $3508000
      LAMDAT = LAMDAR                                                $3508200
      GO TO 15000                                                    $3508400
C
C             PRINT MINIMUM SPOT SIZE                                $3509000
16500 WRITE ( 6, 32110 ) LAMDAR, XBAR, YBAR, HIMAGE, XSTAR, YSTAR, RSTAR$3509200    164
32110 FORMAT( 1H , 3X, 1[illegible]IN. SPOT SIZE, 7E16.7 )                  $3509400
      LINE = LINE + 3                                                $3509600
      DO  18000  K = 1, NIPLN                                        $3601000
      IF( LINE .LE. 54 )  GO TO 17000                                $3601200
C             BEGIN NEW PAGE  -  PRINT HEADINGS                      $3601600
      CALL  PRTCTL                                                   $3601800    171
      WRITE ( 6, 32080 )                                             $3602000    172
      LINE = LINE + 2                                                $3602200
C
17000 XBAR = MXBAR * LAMDA( K ) + KXBAR                              $3603000
      YBAR = MYBAR * LAMDA( K ) + KYBAR                              $3603200
      HIMAGE = XBAR * SNTHTR + YBAR * CSTHTR                         $3603400
      XSTAR=0.                                                       $3603410
      YSTAR=0.                                                       $3603412
      DO 17010 ITER=1,NRAYSD                                         $3603414
      XCORD(ITER)=MXCURL(ITER)*LAMDA(K)+KXCURL(ITER)                 $3613416
17010 YCORD(ITER)=MYCURL(ITER)*LAMDA(K)+KYCURL(ITER)                 $3603418
      DO 17012 ITER=1,NRAYSB                                         $3603420
      XSTAR=XSTAR+XCORD(ITER)*XCORD(ITER)                            $3603422
17012 YSTAR=YSTAR+YCORD(ITER)*YCORD(ITER)                            $3603424
      XSTAR=SQRT(XSTAR/FNR)                                          $3603426    195
      YSTAR=SQRT(YSTAR/FNR)                                          $3603428    197
      RSTAR=SQRT(XSTAR*XSTAR+YSTAR*YSTAR)                            $3603430    198
      WRITE (6,32120) K, LAMDA(K), XBAR, YBAR, HIMAGE, XSTAR,YSTAR,RSTAR         199
32120 FORMAT( 4X, 5HPLANE, I4, 5X, 7E16.7 )                          $3604600
      LINE = LINE + 1                                                $3605400
18000 CONTINUE                                                       $3605800
20000 CONTINUE                                                       $3607000
C
C             END OF STARS SUBROUTINE                                $3901000
      RETURN                                                         $3902000
```

```
      END                                                        S3903000
```

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS . 09/24/68 PAGE
STAR32 STORAGE MAP

SUBROUTINE STARS
COMMON VARIABLES

COMMON BLOCK DATA ORIGIN 00001 LENGTH 06671

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYP
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
NSUBP	00041	I	NIPLN	00042	I	IMODE	00043	I
NSPLN	00044	I	NOBJH	00045	I	NSURF	00046	I
IFLAG	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
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
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	ATRGGR	06666	I
GAUSS	06667	R						

COMMON BLOCK TMPATT ORIGIN 06672 LENGTH 00651

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYP
NMISS	00000	I	NRFLCT	00001	I	NVIGN	00002	I
NRAYSB	00003	I	XBAR	00004	R	YBAR	00005	R
XSTAR	00005	R	YSTAR	00007	R	RSTAR	00010	R
SJSTAR	00011	R	XCURL	00012	R	YCURL	00323	R
MXBAR	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	TYP
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R

COMMON BLOCK PRNT ORIGIN 07546 LENGTH 00002

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LINE	00000	I	PAGE	00001	I

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TY
XVCT	10370	R	OVCT	10373	R	HEITS	10376	R
LAMDA	10405	R	FMT	10551	R	OLIST	10560	R
TEMPS	07550	R	KXCURL	07550	R	KYCURL	10060	R
XCORD	10510	R	YCORD	11100	R			

SUBROUTINE TO PERFORM OPTIC DIAGNOSTIC CALCULATIONS 09/24/68 PAGE 8

STAR3Z STORAGE MAP

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	F
DUMMY	11432	R	EXITP	11433	R	MRAYS	11434	I
MTEMP	11435	I	K	11436	I	KTEMP	11437	I
FNR	11440	F	HIMAGE	11441	R			

ENTRY POINTS

STA[illegible] SECTION 11

SUBROUTINES CALLED

FRTCTL	SECTION	12	.FWRD.	SECTION	13	PARAX	SECTION	
MERID	SECTION	15	RAYTR	SECTION	16	SIN	SECTION	
COS	SECTION	18	SMFOP	SECTION	19	SQRT	SECTION	
.FXEM.	SECTION	21	.UNO6.	SECTION	22	.FFIL.	SECTION	
.FCNV.	SECTION	24	E.1	SECTION	25	E.2	SECTION	
E.3	SECTION	27	E.4	SECTION	28	CC.1	SECTION	
CC.2	SECTION	30	CC.3	SECTION	31	CC.4	SECTION	
SYSLOC	SECTION --- 33							

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATI
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	8000	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.

Section 7

LINK 7

```
                         INF.N010001.44800-0.77387.AM IC      IBJOB LIDHK1  7094 A    02/16/68          PAGE  8
$0
$TITLE          SUBROUTINE TO COMPUTE LINE AND CHARACTER POSITION OF POINT
$IBFTC GETLCZ  M94,XR7                LINK 7    (GETLC)                        6T001000
```

```
     SUBROUTINE TO COMPUTE LINE AND CHARACTER POSITION OF POINT          08/18/68          PAGE  1
          GETLC2      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  GETLC( X, Y, SMIN, DELL, DELC, NCHAR, NLINE )        GT001200
C
C     *** INPUTS ***                                                    GT002000
C X        X COORDINATE OF POINT.                                       GT002010
C Y        Y COORDINATE OF POINT.                                       GT002020
C SMIN     VALUE ASSIGNED TO LOWER LEFT CORNER OF GRID.                 GT002030
C DELL     INCREMENT IN Y DIRECTION (LINE).                             GT002040
C DELC     INCREMENT IN X DIRECTION (CHARACTER).                        GT002050
C
C     *** OUTPUTS ***                                                   GT002200
C NCHAR    CHARACTER POSITION OF POINT ON GRID.                         GT002210
C NLINE    LINE POSITION OF POINT ON GRID.                              GT002220
C
C     *** DESCRIPTION ***                                               GT002400
C          GETLC COMPUTES THE LINE AND CHARACTER POSITION OF AN INPUT   GT002410
C          POINT WHICH IS THEN OUTPUT ON A SPOT DIAGRAM POINT PLOT.     GT002420
C
      QX = ( X - SMIN ) / DELC                                          GT005000
      QY = ( Y - SMIN ) / DELL                                          GT005200
      TX = IFIX( QX )                                                   GT005400
      TY = IFIX( QY )                                                   GT005600
      NCHAR = TX                                                        GT006000
      NLINE = TY                                                        GT006200
      IF( QX - TX .GT. 0.5 )  NCHAR = NCHAR + 1                         GT006400
      IF( QY - TY .GT. 0.5 )  NLINE = NLINE + 1                         GT006600
      NLINE = 61 - NLINE                                                GT006800
C
C               END OF GETLC SUBROUTINE                                 GT008000
      RETURN                                                            GT008200
      END                                                               GT008400
```

```
SUBROUTINE TO COMPUTE LINE AND CHARACTER POSITION OF POINT          02/16/68                    PAGE 2
    GETLC2                                              STORAGE MAP

                                              SUBROUTINE GETLC
                                                 UNDIMENSIONED PROGRAM VARIABLES

SYMBOL      LOCATION     TYPE          SYMBOL      LOCATION     TYPE          SYMBOL     LOCATION     TYPE
GX           00001        R            GY           00002        R            TX          00003        R
TY           00004        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.
```

```
                    INF.N010001.44800-0.77387.AM IC      IBJOB LIONK1  7094 A    02/18/68          PAGE  3

$*
$TITLE        SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
$IBFTC START7  M94,XR7              LINK 7   (START7)                   S7001000
```

```
      SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS                        02/16/68                    PAGE   1
           START7       - EFN    SOURCE STATEMENT  -  IFN(S)   -

       SUBROUTINE  START7                                                   S7001500
C
C     *** DESCRIPTION ***                                                   S7002000
C         START7 LISTS THE ENTRANCE PUPIL PATTERN AND THE IMAGE PLANE       S7002100
C         COORDINATES FOR EACH COMBINATION OF COLOR AND OBJECT POINT.       S7002200
C         SPOT DIAGRAM POINT PLOTS ARE PRINTED FOR THE ENTRANCE PUPIL       S7002300
C         AND EACH COMBINATION UNLESS A RAY FAILURE OCCURS.                 S7002400
C
       COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), 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), 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
       COMMON / THPATT / NMISS, NRFLCT, 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/PRNT/LINE,PAGE                                                PT002500
       INTEGER PAGE                                                         PT002700
C
       COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                      AZ0040
C
       DIMENSION  NRBAR(42), NX(201), NY(201), PRTL(103), TEMPS(200),       S7002500
      1 PLTCHR(38), CHRSPT(200)                                             S7002600
C
       INTEGER  COLOR, MNUMB, RYCNT                                         S7003000
       REAL  LENGTH                                                         S7003500
C
       DATA  PLTCHR(1) / 1H1 /, PLTCHR(2) / 1H2 /, PLTCHR(3) / 1H3 /,       S7004000
      1 PLTCHR(4) / 1H4 /, PLTCHR(5) / 1H5 /, PLTCHR(6) / 1H6 /,            S7004100
```

```
     SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS                           08/10/68                PAGE  2
          STAR7        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

       2 PLTCHR(7)  / 1H7 /, PLTCHR(8)  / 1H8 /, PLTCHR(9)  / 1H9 /,        S7004200
       3 PLTCHR(10) / 1HA /, PLTCHR(11) / 1HB /, PLTCHR(12) / 1HC /,        S7004300
       4 PLTCHR(13) / 1HD /, PLTCHR(14) / 1HE /, PLTCHR(15) / 1HF /,        S7004400
       5 PLTCHR(16) / 1HG /, PLTCHR(17) / 1HH /, PLTCHR(18) / 1HI /,        S7004500
       6 PLTCHR(19) / 1HJ /, PLTCHR(20) / 1HK /, PLTCHR(21) / 1HL /,        S7004600
       7 PLTCHR(22) / 1HM /, PLTCHR(23) / 1HN /, PLTCHR(24) / 1HO /,        S7004700
       8 PLTCHR(25) / 1HP /, PLTCHR(26) / 1HQ /, PLTCHR(27) / 1HR /,        S7004800
       9 PLTCHR(28) / 1HS /, PLTCHR(29) / 1HT /, PLTCHR(30) / 1HU /,        S7004900
       A PLTCHR(31) / 1HV /, PLTCHR(32) / 1HW /, PLTCHR(33) / 1HX /,        S7005000
       B PLTCHR(34) / 1HY /, PLTCHR(35) / 1HZ /, PLTCHR(36) / 1H  /,        S7005100
       C PLTCHR(37) / 1H0 /, PLTCHR(38) / 1H* /                             S7005200
C
C             START OF STAR7 SUBROUTINE                                     S7021000
C
        MR = NRAYS                                                          S7021500
        IF( LMODE .EQ. 0 )  MR = MR + MR                                    S7021600
        NRAYS = 0                                                           S7022000
        RYCNT = 0                                                           S7022200
        DO  2000  I = 1, NRAYS                                              S7022400
        RYCNT = RYCNT + 1                                                   S7022600
        CALL  LATT( XCURL(RYCNT), YCURL(RYCNT), NRAYS )                     S7022800    11
        IF( LMODE .NE. 0 )  GO TO 2000                                      S7023000
        XCURL( RYCNT+1 ) = - XCURL( RYCNT )                                 S7023200
        YCURL( RYCNT+1 ) = YCURL( RYCNT )                                   S7023400
        RYCNT = RYCNT + 1                                                   S7023600
 2000 CONTINUE                                                              S7023800
        DELL = 2. / 61.                                                     S7024000
        DELC = 2. / 102.                                                    S7024200
        LINE = 80                                                           S7024400
C
        DO 5000  I = 1, MR                                                  S7025000
        IF( LINE .LE. 60 )  GO TO 4000                                      S7025200
C             BEGIN NEW PAGE                                                S7025400
        CALL  PRTCTL                                                        S7025600    32
        WRITE ( 6, 32010 )                                                  S7025800    33
32010 FORMAT( 1H0, 45HRAY COORDINATES ON UNIT RADIUS ENTRANCE PUPIL  /      S7026000
     14H RAY, 8X, 1HX, 15X, 1HY, 8X, 26HLINE  CHAR  PLOT CHARACTER )        S7026200
        LINE = LINE + 4                                                     S7026400
C
C             COMPUTE LINE AND CHARACTER                                    S7026600
 4000 CALL  GETLC( XCURL(I), YCURL(I),-1., DELL, DELC, NX(I), NY(I) )       S7026800    39
        CHAR = PLTCHR(33)                                                   S7027000
        IF( MR .LE. 35 )  CHAR = PLTCHR(I)                                  S7027200
        WRITE ( 6, 32020 )  I, XCURL(I), YCURL(I), NY(I), NX(I), CHAR       S7027400    43
32020 FORMAT( 1H , I3, 2E16.7, I5, I6, 10X, A1 )                            S7027600
        LINE = LINE + 1                                                     S7027800
 5000 CONTINUE                                                              S7028000
        KBAR = MR + 1                                                       S7028200
C
C             CONVERT ORIGIN                                                S7028400
        CALL  GETLC( 0., 0.,-1., DELL, DELC, NX(KBAR), NY(KBAR) )           S7028600    54
        PAGE = PAGE + 1                                                     S7028800
        LINE = 0                                                            S7029000
C
C             PRINT FIRST LINE                                              S7041000
        WRITE ( 6, 32030 )  ( TITLE(L), L=1,12 ), PAGE                      S7041200    57
```

```
      SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS                         02/16/68                PAGE  3
            START2      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

32030 FORMAT( 1H1, 12A6, 3X, 29HRAY PATTERN ON ENTRANCE PUPIL, 10X,      S7041400
     1 4HPAGE, I4 )                                                      S7041600
      DO  8500  I = 1, 62                                                S7041800
      PRTL(1) = PLTCHR(38)                                               S7042000
      PRTL(103) = PLTCHR(38)                                             S7042200
      DO  6000  K = 2, 102                                               S7042400
 6000 PRTL(K) = PLTCHR(36)                                               S7042600
      IF( LINE .NE. 0  .AND.  LINE .NE. 61 )  GO TO 7000                 S7042800
      DO  6500  K = 3, 101, 2                                            S7043000
 6500 PRTL(K) = PLTCHR(38)                                               S7043200
 7000 DO  7500  K = 1, NR                                                S7043400
      IF( LINE .NE. NY(K) )  GO TO 7500                                  S7043600
      KL = NX(K) + 1                                                     S7043800
      PRTL(KL) = PLTCHR(K)                                               S7044000
      IF( NR .GT. 35 )  PRTL(KL) = PLTCHR(33)                            S7044200
 7500 CONTINUE                                                           S7044400
      IF( LINE .NE. NY(KBAR) )  GO TO 8000                               S7045000
      KL = NX(KBAR) + 1                                                  S7045200
      PRTL(KL) = PLTCHR(37)                                              S7045400
 8000 IF( LINE .EQ. 0 )  GO TO 8200                                      S7045600
      IF( LINE .EQ. 61 )  GO TO 8400                                     S7045800
      WRITE ( 6, 32040 )  PRTL                                           S7046000     108
32040 FORMAT( 20X, 103A1 )                                               S7046200
      GO TO 8500                                                         S7046400
C
C               PRINT UPPER LINE                                         S7046600
 8200 SMAX = 1.                                                          S7046800
      WRITE ( 6, 32050 )  SMAX, PRTL                                     S7047000     112
32050 FORMAT( 6X, E11.4, 3X, 103A1 )                                     S7047200
      GO TO 8500                                                         S7047400
C
C               PRINT LOWER LINE                                         S7047600
 8400 SMIN = - 1.                                                        S7047800
      WRITE ( 6, 32050 )  SMIN, PRTL                                     S7048000     116
 8500 LINE = LINE + 1                                                    S7048200
C
C               BEGIN REGULAR PLOTS                                      S7081000
      NSPOTS = 0                                                         S7081200
      LINE = 60                                                          S7081400
      REWIND  11                                                         S7081600
C                                                                                     123
      DO  9000  I = 1, 42                                                S7081800
 9000 NRBAR(I) = 0                                                       S7082000
      IF( IMODE .NE. 0 )  GO TO 10000                                    S7082200
      THTR = 0.                                                          S7082400
      SNTHTR = 0.                                                        S7082600
      CSTHTR = 1.                                                        S7082800
      CALL  MERID( 1, BF1, DUMMY )                                       S7083000     134
10000 THTR = ROTAN * 3.14159265 / 180.                                   S7083200
      SNTHTR = SIN( THTR )                                               S7083400     136
      CSTHTR = COS( THTR )                                               S7083600
C
C               BEGIN DO LOOP ON OBJECT HEIGHT                           S7083800     137
      DO  12500  J = 1, NOBJH                                            S7084000
      HEIGHT = HO + FLOAT( J-1 ) * DELH                                  S7084200
C
```

```
      SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS                          08/18/88                PAGE  4
          START2      - EFN   SOURCE STATEMENT  -  IFN(S)  -

C              BEGIN LOOP ON COLORS                                 87084400
      DO  12500  I = 1, NCLRS                                       87084600
C              TRACE RAY PATTERN                                    87084800
      CALL  SHPOP( I, J, HEIGHT, BF1, 3, TEMPS )                    87085000    143
      IF( NRAYSB .EQ. 0 )  GO TO 12500                              87085200
      NSPOTS = NSPOTS + 1                                           87085400
      NRBAR(NSPOTS) = NRAYSB                                        87085600
      XTMIN = XCURL(1)                                              87085800
      XTMAX = XCURL(1)                                              87086000
      YTMIN = YCURL(1)                                              87086200
      YTMAX = YCURL(1)                                              87086400
      IF( NRAYSB .EQ. 1 )  GO TO 11200                              87086600
C
      DO  11000  K = 2, NRAYSB                                      87086800
      XTMIN = AMIN1( XTMIN, XCURL(K) )                              87087000
      XTMAX = AMAX1( XTMAX, XCURL(K) )                              87087200
      YTMIN = AMIN1( YTMIN, YCURL(K) )                              87087400
11000 YTMAX = AMAX1( YTMAX, YCURL(K) )                              87087600
11200 IF( NSPOTS.NE. 1 )  GO TO 11500                               87087800
      XMIN = XTMIN                                                  87088000
      XMAX = XTMAX                                                  87088200
      YMIN = YTMIN                                                  87088400
      YMAX = YTMAX                                                  87088600
      GO TO 12000                                                   87088800
11500 XMIN = AMIN1( XMIN, XTMIN )                                   87089000
      XMAX = AMAX1( XMAX, XTMAX )                                   87089200
      YMIN = AMIN1( YMIN, YTMIN )                                   87089400
      YMAX = AMAX1( YMAX, YTMAX )                                   87089600
C
C              WRITE BINARY TAPE 11  -  1 RECORD                    87071000
12000 WRITE ( 11 )  J, I, XBAR, YBAR, NRAYSB, XTMIN, XTMAX, YTMIN,   87071200
     1 YTMAX, ( XCURL(K), K=1,NRAYSB ), ( YCURL(K), K=1,NRAYSB ),   87071400
     2 ( TEMPS(K), K=1,MR )                                         87071600    166
12500 CONTINUE                                                      87071800
      IF( NSPOTS .EQ. 0 )  GO TO 32000                              87072000
C
      END FILE  11                                                  87072200    185
      REWIND  11                                                    87072400    186
      SMIN = PSCAL * AMIN1( XMIN, YMIN )                            87072600
      SMAX = PSCAL * AMAX1( XMAX, YMAX )                            87072800
C
C              COMPUTE SCALE                                        87073000
      CALL  SCALEK( SMIN, SMAX, PLX, PLY, XMIN, XMAX )              87073200    187
      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    193
      CALL  SETLC( 0., 0., SMIN, DELL, DELC, NXORG, NYORG )         87075400    194
      DO  16000  I = 1, NSPOTS                                      87075600
```

```
      KBAR = NRBAR(I)                                                   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,NR )                                             87076600     200
      IF( LINE .GT. 50 )  CALL  PRTCTL                                  87076800
C                                                                                    216
      WRITE ( 6, 32060 )  MNUMB, COLOR, XBAR, YBAR, KBAR                87077000     217
32060 FORMAT( 1H0, 12HOBJECT POINT, I2, 2X, 5HCOLOR, I2 / 13X,  9HAVERAG87077200
     1E X, 7X, 9HAVERAGE Y, 4X, 14HNUMBER OF RAYS / 8X, 2E16.7, I11 //  87077400
     2 5X, 3HRAY, 9X, 1HX, 15X, 1HY, 10X, 10HX (SCALED), 6X, 10HY (SCALE87077600
     3D), 4X, 4HLINE, 2X, 4HCHAR, 2X, 14HPLOT CHARACTER )               87077800
      LINE = LINE + 7                                                   87078000
      RYCNT = 0                                                         87078200
C
      DO  15000  K = 1, NR                                              87078400
      IF( LINE .LE. 59 )  GO TO 14000                                   87078600
C              BEGIN NEW PAGE                                           87078800
      CALL  PRTCTL                                                      87078900     225
      WRITE ( 6, 32070 )  MNUMB, COLOR                                  87079000     226
32070 FORMAT( 1H0, 12HOBJECT POINT, I2, 2X, 5HCOLOR, I2 // 5X, 3HRAY,   87081000
     1 9X, 1HX, 15X, 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
      XS = PSCAL * XCURL(RYCNT)                                         87082600
      YS = PSCAL * YCURL(RYCNT)                                         87082800
C
C              GET LINE AND CHARACTER COORDINATES                       87083000
      CALL  GETLC( XS, YS, SMIN, DELL, DELC, NX(RYCNT), NY(RYCNT) )     87083200
C
C              GET PLOT CHARACTER                                       87083400     234
      CHRSPT(RYCNT) = PLTCHR(K)                                         87083600
      IF( NR .GT. 35 )  CHRSPT(RYCNT) = '[illegible])                   87083800
      WRITE ( 6, 32080 )  K, XCURL(RYCNT), [illegible]YCNT), XS, YS,    87084000
     1 NY(RYCNT), NX(RYCNT), CHRSPT(RYCNT)                              87084200     242
32080 FORMAT( 1H , 4X, I3, 4E16.7, I5, I6, 10X, 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, 5X, 9HREFLECTED )                            87086000
      GO TO 15000                                                       87086200
C
C              VIGNETTED RAY                                            87086400
14800 WRITE ( 6, 32110 )  K                                             87086600     253
```

```
        SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS                          02/18/68                PAGE  6
             START2      - EFN   SOURCE STATEMENT  -  IFN(S)  -

32110 FORMAT( 1H , 4X, I3, 5X, 9HVIGNETTED )                            87086800
15000 LINE = LINE + 1                                                   87087000
      WRITE ( 12 )  HNUMB, COLOR, ( NX(K), K=1,KBAR ), ( NY(K), K=1,KBAR87087200
     1 ), ( CHRSPT(K), K=1,KBAR )                                       87087400    266
16000 CONTINUE                                                          87087600
      END FILE  12                                                      87087800    271
      REWIND  12                                                        87088000
C                                                                                   272
      DO  20000  I = 1, NSPOTS                                          87088200
      KBAR = NRBAR(I)                                                   87088400
      READ ( 12 )  HNUMB, COLOR, ( NX(K), K=1,KBAR ), ( NY(K), K=1,KBAR 87088600
     1 ), ( CHRSPT(K), K=1,KBAR )                                       87088800    277
      PAGE = PAGE + 1                                                   87089000
      LINE = 0                                                          87089200
      WRITE ( 6, 32120 )  ( TITLE(L), L=1,12 ), HNUMB, COLOR, PAGE      87091000    283
32120 FORMAT( 1H1, 12A6, 3X, 13HHEIGHT NUMBER, I2, 2X, 12HCOLOR NUMBER, 87091200
     1 I2, 8X, 4HPAGE, I4 )                                             87091400
C               CONSTRUCT JTH. LINE OF PLOT                             87091600
C
      DO  19000  J = 1, 62                                              87091800
      PRTL(1) = PLTCHR(38)                                              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 17500                                87093600
      KL = NX(K) + 1                                                    87093800
      PRTL(KL) = CHRSPT(K)                                              87094000
17500 CONTINUE                                                          87094200
      IF( LINE .NE. NYORG )  GO TO 18000                                87094400
      KL = NXORG + 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                                                          87097000
C
32000 RETURN                                                            87098000
      END                                                               87098400
```

```
SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS                              09/18/68                    PAGE  7
     START7                                          STORAGE  MAP

                                             SUBROUTINE  START
                                                   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        00013          R
DATE         00027          I           PUNCID       00032          R           DATA         00033          R
LMODE        00033          I           NRAYS        00034          I           NSLCS        00038          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
DFLD         00063          R           EPRAD        00064          R           PSCAL        00065          R
OMG/F        00066          R           SPFEA        00067          R           DUMIN        00070          R
ODIST        00071          R           HO           00072          R           DELH         00073          R
SYSNX        00074          R           WXDIR        00075          R           WYDIR        00076          R
RDTAN        00077          R           NDSGN        00100          I           NDSGV        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        08665          I

                    COMMON BLOCK       TMPATT            ORIGIN        08667              LENGTH       00851
NMISS        00000          I           NRFLCT       00001          I           NVIGN        00002          I
NRAYSB       00003          I           XBAR         00004          R           YBAR         00005          R
XSTAR        00006          R           YSTAR        00007          R           KSTAR        00010          R
SJSTAR       00011          R           XCURL        00012          R           YCURL        00323          R
MXBAR        00834          R           KXBAR        00835          R           MYBAR        00836          R
KYBAR        00837          R           AX           00840          R           BX           00841          R
CX           00842          R           AY           00843          R           BY           00844          R
CY           00845          R           ACAP         00846          R           BCAP         00847          R
CCAP         00850          R

                    COMMON BLOCK       PRNT              ORIGIN        07540              LENGTH       00002
LINE         00000          I           PAGE         00001          I

                    COMMON BLOCK       AZOBJ             ORIGIN        07542              LENGTH       00003
THTR         00000          R           SNTHTR       00001          R           CSTHTR       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           TEMPS        10810          R           PLTCHR       11120          R
CHRSPT       11166          R
```

SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS 02/16/68 PAGE 8

START2

STORAGE MAP

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
COLOR	11476	I	NNUMB	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	XTMIN	11522	R
XTMAX	11523	R	YTMIN	11524	R	YTMAX	11525	R
XMIN	11526	R	XMAX	11527	R	YMIN	11530	R
YMAX	11531	R	PLX	11532	R	PLY	11533	R
NXORG	11534	I	NYORG	11535	I	INDRAY	11536	I
XS	11537	R	YS	11540	R			

ENTRY POINTS

START	SECTION	11

SUBROUTINES CALLED

LATT	SECTION	12	PRTCTL	SECTION	13	.FWRD.	SECTION	14
GETLC	SECTION	15	.FSLO.	SECTION	16	.FRWT.	SECTION	17
MERID	SECTION	18	SIN	SECTION	19	COS	SECTION	20
SHPOP	SECTION	21	.FWRB.	SECTION	22	.FEFT.	SECTION	23
SCALEK	SECTION	24	.FRDB.	SECTION	25	.FXEM.	SECTION	26
.UN06.	SECTION	27	.FFIL.	SECTION	28	.FCNV.	SECTION	29
.UN11.	SECTION	30	.FWLR.	SECTION	31	.FBLT.	SECTION	32
.FBDT.	SECTION	33	.UN12.	SECTION	34	.FRLR.	SECTION	35
E.1	SECTION	36	E.2	SECTION	37	E.3	SECTION	38
E.4	SECTION	39	CC.1	SECTION	40	CC.2	SECTION	41
CC.3	SECTION	42	CC.4	SECTION	43	SYSLOC	SECTION	44

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
2000	22A	12110	5000	48A	12236	4000	34A	12151
32010	FORMAT	11573	32020	FORMAT	11620	32030	FORMAT	11626
8500	118A	12475	6000	69A	12330	7000	80A	12350
6500	77A	12344	7500	94A	12374	8000	103A	12412
8200	111A	12436	8400	115A	12456	32040	FORMAT	11643
32050	FORMAT	11646	9000	127A	12512	10000	135A	12533
12500	179A	13032	11200	161A	12713	11000	158A	12702
11500	165A	12730	12000	166A	12756	32000	351A	14061
13000	192A	13131	16000	269A	13575	32060	FORMAT	11652
15000	254A	13527	14000	227A	13335	32070	FORMAT	11723
14200	230A	13354	14400	249A	13475	14600	251A	13506
14800	253A	13517	32080	FORMAT	11760	32090	FORMAT	11767
32100	FORMAT	11774	32110	FORMAT	12002	20000	348A	14057
32120	FORMAT	12010	19000	346A	14055	16500	303A	13720
17200	314A	13740	17000	311A	13734	17500	325A	13755
18000	332A	13773	18200	340A	14017	18400	343A	14035

```
        SUBROUTINE TO PRODUCE SPOT DIAGRAM POINT PLOTS
            START                                                       02/16/68                        PAGE  9
                                                     STORAGE  MAP
18500          345A          14082
 THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 14112.
```

Section 8

LINK 8

```
$*
$TITLE          SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS
$IBFTC STAR8Z  M94,XR7              LINK 8   (STAR8)                  S8001000
```

```
      SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS                    08/18/68         PAGE   1
          STARBZ      - EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE  STARS                                                   S8001500
C                                                                        
C     *** INPUTS ***                                                      S8001600
C NPERTS NUMBER OF PARAMETERS TO BE PERTURBED.         (PERTS COMMON)    S8001610
C PERTS  PERTURBATION MATRIX CONTAINING PARAMETER INDEX  (PERTS COMMON)  S8001620
C        AND PERCENT AND THE PARAMETER SYMBOL.                           S8001630
C REFOCS FLAG DEFINING THE TYPE OF REFOCSING DESIRED.    (PERTS COMMON)  S8001640
C
C     *** OUTPUTS ***                                                    S8001650
C        VARIOUS RAY STATISTICS FOR NOMIAL AND PERTURBED SYSTEMS.        S8001660
C
C     *** DESCRIPTION ***                                                S8001670
C        STARS PERFORMS THE COMPUTATIONS FOR THE SENSITIVITY ANALYSIS.   S8001680
C        STATISTICAL DATA FOR THE NOMINAL SYSTEM IS CALCULATED FOR EACH  S8001690
C        COLOR AND OBJECT POINT.  THE CALCULATIONS ARE THEN REPEATED     S8001700
C        FOR EACH PERTURBED PARAMETER.  A REFOCUS OPTION MAY BE          S8001710
C        SPECIFIED TO OBTAIN THE PLANE LOCATION FOR MINIMUM RMS X,       S8001720
C        RMS Y, OR SPOT SIZE.                                            S8001730
C
      COMMON / DATA / NCNTRL, CONTRL(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), 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), 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), CJIST ), ( 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), 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
      COMMON / THPATT / NMISS, NRFLCT, 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/PRNT/LINZ,PAGE                                              PT002500
      INTEGER PAGE                                                       PT002700
```

```
C
C          THIS BLOCK OF COMMON CONTAINS PERTURBATION VALUES              PB001000
      COMMON / PERTB / PERTB(30,4), NPERTB, REFOCS                        PB002000
      EQUIVALENCE ( PERTB(1,1), IPERTB(1,1) )                             PB003000
      DIMENSION IPERTB(30,4)                                              PB004000
      INTEGER REFOCS                                                      PB005000
C
      COMMON/AZOBJ/THTR,SNTHTR,CSTHTR                                     AZ004000
C
      DIMENSION STATN(6,9,7), TEMPS(400), GHX(6,17), CL3(425),            S8002000
     1 GAMMA(5), SJN(7), T1(30), DELTAT(30), IGHX(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 ( GHX(1,1), IGHX(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 5000 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) = NRAYSB                                              S8025800
      ISTATN(I,7,J) = NMISS                                               S8026000
      ISTATN(I,8,J) = NRFLCT                                              S8026200
      ISTATN(I,9,J) = NVIGN                                               S8026300
 4000 CONTINUE                                                            S8026400
      SJN(J) = SJSTAR                                                     S8027000
```

```
 5000 CONTINUE                                                          S8027200
      NR = NRAYS                                                        S8028000
      IF( LMODE .EQ. 0 )  NR = NR + NR                                  S8028200
      KNOX = 3 * REFOCS + 2                                             S8028400
      CALL  PRTCTL                                                      S8028600
C                                                                                  54
      WRITE ( 6, 32010 )                                                S8028800   55
32010 FORMAT( 1H0, 13X, 45H***** UNPERTURBED SYSTEM RAY STATISTICS *****S8029000
     1 //74H OBJECT POINT  COLOR  RAYS TRACED  SUCCESSES  MISSES  REFLECS8029200
     2TIONS  VIGNETS )                                                  S8029400
C
      DO  6000  J = 1, NOBJH                                            S8041000
      DO  6000  I = 1, NCLRS                                            S8041200
      WRITE( 6, 32020 )  J, I, NR, ( ISTATN(I,K,J), K = 6, 9 )          S8041400   60
32020 FORMAT( 1H0, I7, I10, 8X, I3, 9X, I3, 6X, I3, 8X, I3, 8X, I3 )    S8041600
 6000 CONTINUE                                                          S8041800
C
      DO 15000  K = 1, NPERTB                                           S8042000
      PERCNT = PERTB(K,1)                                               S8042200
      INDX = IPERTB(K,2)                                                S8042400
      SYMB1 = IPERTB(K,3)                                               S8042600
      SYMB2 = IPERTB(K,4)                                               S8042800
C             SAVE NOMINAL VALUE OF KTH. PARAMETER                      S8043000
      TMP = DATA( INDX )                                                S8043200
C             PERTURB KTH. PARAMETER                                    S8043600
      DATA(INDX) = T1(K)                                                S8043800
C             COMPUTE BACK FOCUS - COLOR 1                              S8044000
      THTR = 0.                                                         S8044200
      SNTHTR = 0.                                                       S8044400
      CSTHTR = 1.                                                       S8044600
      CALL  MERID( 1, BF1, DUMMY )                                      S8045000
C             SET AZIMUTH OF OBJECT                                     S8045200   82
      THTR = ROTAN * 3.14159265 / 180.                                  S8045400
      SNTHTR = SIN( THTR )                                              S8045600   84
      CSTHTR = COS( THTR )                                              S8045800   85
      LINE = 60                                                         S8046000
C
      DO  14000 J = 1, NOBJH                                            S8046200
      HEIGHT = HO + FLOAT( J-1 ) * DELH                                 S8046400
      DO  8000  I = 1, NCLRS                                            S8047000
C
C             ZERO LINKS COMMON                                         S8047200
      DO  7000  LK3 = 1, 425                                            S8047400
 7000 CL3(LK3) = 0.0                                                    S8047600
      CALL  SMPOP( I, J, HEIGHT, BF1, 1, TEMPS )                        S8047800
C             LOAD GMX FROM LINKS COMMON                                S8048000   98
      DO  7500  LK3 = 413, 425                                          S8048200
 7500 GMX(I,LK3-412) = CL3(LK3)                                         S8048400
      IGMX(I,14) = NRAYSB                                               S8048600
      IGMX(I,15) = NMISS                                                S8048800
      IGMX(I,16) = NRFLCT                                               S8049000
      IGMX(I,17) = NVIGN                                                S8049200
 8000 CONTINUE                                                          S8049400
      IF( REFOCS .EQ. 0 )  GO TO 8700                                   S8051000
      SUMA = 0.                                                         S8051200
      SUMB = 0.                                                         S8051400
```

```
      SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS                          02/16/68                PAGE  4
           STAR8Z      - EFN   SOURCE STATEMENT  -  IFN(S)  -

       DO  8500  I = 1, NCLRS                                            S8081600
       SUMA = SUMA + GMX(I,KNDX)                                         S8081800
  8500 SUMB = SUMB + GMX(I,KNDX+1)                                       S8082000
       FLAMDA = - SUMB / SUMA                                            S8082200
  8700 DO  14000  I = 1, NCLRS                                           S8082400
       IF( ISTATN(I,6,J) .EQ. 0 )  GO TO 14000                           S8082600
       IF( LINE .LE. 48 )  GO TO 9000                                    S8082800
C              BEGIN NEW PAGE                                            S8083000
       CALL  PRTCTL                                                      S8083200
C                                                                                      130
       WRITE ( 6, 32030 )  K, SYMB1, SYMB2, TMP, T1(K), DELTAT(K), PERCNTS8083400      131
 32030 FORMAT( 1H0, 42X, 9HPARAMETER, I3, 1X, 2A6 // 24X, 30HNOMINAL VALUS8083800
      1E  PERTURBED VALUE, 4X, 10HDIFFERENCE, 4X, 15HPER CENT CHANGE /   S8083300
      2 21X, 2E16.7, 2E17.7 )                                            S8084000
       LINE = 5                                                          S8084200
  9000 WRITE ( 6, 32040 )  J, I, ( STATN(I,KT,J), KT = 1,5 ),            S8085000
      1 ISTATN(I,6,J)                                                    S8085200      134
 32040 FORMAT( 1H0, 12HOBJECT POINT, I2, 2X, 5HCOLOR, I2 // 34X, 4HXBAR, S8085400
      1 12X, 4HYBAR, 12X, 5HRMS X, 11X, 5HRMS Y, 8X, 9HSPOT SIZE, 5X,    S8085600
      2 4HRAYS / 9X, 18HNOMINAL VALUE....., 5E16.7, I6 )                 S8085800
       LINE = LINE + 5                                                   S8086000
       IF( IGMX(I,14) .NE. 0 )  GO TO 9500                               S8086200
C
       WRITE ( 6, 32050 )  ( IGMX(I,KT), KT = 14, 17 )                   S8086400      144
 32050 FORMAT( 1H0, 31X, 39HSUCCESSES  MISSES  REFLECTIONS  VIGNETS /    S8086400
      1 35X, I3, 6X, I3, 8X, I3, 8X, I3 )                                S8086600
       LINE = LINE + 3                                                   S8086800
       GO TO 14000                                                       S8087000
  9500 XOBAR = STATN(I,1,J)                                              S8088000
       YOBAR = STATN(I,2,J)                                              S8088200
       XOSTAR = STATN(I,3,J)                                             S8088400
       YOSTAR = STATN(I,4,J)                                             S8088600
       ROSTAR = STATN(I,5,J)                                             S8088800
       ELAMDA = SJSTAR                                                   S8089000
       ASSIGN 10200 TO IGO                                               S8089200
C
C               SPECIAL LOGIC SECTION                                    S8081000
C
  9800 XBAR = GMX(I,1) * ELAMDA + GMX(I,2)                               S8081200
       YBAR = GMX(I,3) * ELAMDA + GMX(I,4)                               S8081400
       XSTAR = SQRT( GMX(I,5) * ELAMDA * ELAMDA + 2. * GMX(I,6) * ELAMDA S8081600
      1 + GMX(I,7) )                                                     S8081800      165
       YSTAR = SQRT( GMX(I,8) * ELAMDA * ELAMDA + 2. * GMX(I,9) * ELAMDA S8082000
      1 + GMX(I,10) )                                                    S8082200      169
       RSTAR = SQRT( GMX(I,11) * ELAMDA * ELAMDA + 2. * GMX(I,12) *      S8082400
      1 ELAMDA + GMX(I,13) )                                             S8092600      173
       DXBAR = XBAR - XOBAR                                              S8082800
       DYBAR = YBAR - YOBAR                                              S8083000
       DXSTAR = XSTAR - XOSTAR                                           S8083200
       DYSTAR = YSTAR - YOSTAR                                           S8083400
       DRSTAR = RSTAR - ROSTAR                                           S8083600
       DO  10000  L = 1, 5                                               S8083800
 10000 GAMMA(L) = 0.                                                     S808 000
       IF( XOBAR .NE. 0. )  GAMMA(1) = ( 100. * DXBAR ) / XOBAR          S8084200
       IF( YOBAR .NE. 0. )  GAMMA(2) = ( 100. * DYBAR ) / YOBAR          S8084400
       IF( XOSTAR .NE. 0. )  GAMMA(3) = ( 100. * DXSTAR ) / XOSTAR       S8084600
```

```
      SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS                         02/16/68                    PAGE  5
             STAR8Z      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      IF( YOSTAR .NE. 0. )  GAMMA(4) = ( 100. * DYSTAR ) / YOSTAR       S8084800
      IF( ROSTAR .NE. 0. )  GAMMA(5) = ( 100. * DRSTAR ) / ROSTAR       S8085000
C
C             END OF SPECIAL SECTION                                    S8086000
C
      GO TO IGO,( 10200, 13000 )                                        S8086200
10200 WRITE ( 6, 32060 )  XBAR, YBAR, XSTAR, YSTAR, RSTAR, IGMX(I,14),  S8091000
     1 DXBAR, DYBAR, DXSTAR, DYSTAR, DRSTAR, ( GAMMA(L), L = 1, 5 )     S8091200      191
32060 FORMAT( 1H , 9X, 18HPERTURBED VALUE...., 5E16.7, I6 / 9X, 18HDIFFERS8091400
     1ENCE........., 5E16.7 / 9X, 18HSENSITIVITY......., 5E16.7 )       S8091600
      LINE = LINE + 3                                                   S8091800
      NREFC = REFOCS + 1                                                S8092000
C
      GO TO ( 14000, 10500, 11000, 11500 ), NREFC                       S8092200
10500 WRITE ( 6, 32070 )  FLAMDA                                        S8092400      199
32070 FORMAT( 1H , 9X, 26HMIN. RMS X PLANE LOCATION=, E14.7 )           S8092600
      GO TO 12000                                                       S8092800
11000 WRITE ( 6, 32080 )  FLAMDA                                        S8093000      201
32080 FORMAT( 1H , 9X, 26HMIN. RMS Y PLANE LOCATION=, E14.7 )           S8093200
      GO TO 12000                                                       S8093400
11500 WRITE ( 6, 32090 )  FLAMDA                                        S8093600      203
32090 FORMAT( 1H , 9X, 30HMIN. SPOT SIZE PLANE LOCATION=, E14.7 )       S8093800
12000 ELAMDA = FLAMDA                                                   S8094000
      ASSIGN 13000 TO IGO                                               S8094200
      GO TO 9800                                                        S8094400
C
13000 WRITE ( 6, 32060 )  XBAR, YBAR, XSTAR, YSTAR, RSTAR, IGMX(I,14),  S8095000
     1 DXBAR, DYBAR, DXSTAR, DYSTAR, DRSTAR, ( GAMMA(L), L = 1, 5 )     S8095200      207
      LINE = LINE + 4                                                   S8095400
14000 CONTINUE                                                          S8095600
C
C              RESTORE PARAMETER TO NOMINAL VALUE                       S8096000
      DATA(INDX) = TMP                                                  S8096200
15000 CONTINUE                                                          S8097000
C
      RETURN                                                            S8099000
      END                                                               S8099500
```

SUBROUTINE STAR8
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
LMODE	00033		NRAYS	00034	I	NSLCS	00035	I
ICLRS	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
IFLAG	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
ODIST	00071	R	HO	00072	R	DELH	00073	R
SYSMX	00074	R	WXDIR	00075	R	WYDIR	00076	R
HOTAN	00077	R	NDSGN	00100	I	NDSGV	00104	I
HOBJH	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 THPATT   ORIGIN 06667   LENGTH 00651

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NMISS	00000	I	NRFLCT	00001	I	NVIGN	00002	I
NRAYSB	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
HXBAR	00634	R	KXBAR	00635	R	HYBAR	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	CLS	00000	R			

COMMON BLOCK PRNT   ORIGIN 07540   LENGTH 00002

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
LINE	00000	I	PAGE	00001	I

COMMON BLOCK PERTB   ORIGIN 07542   LENGTH 00172

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
PERTB	00000	R	NPERTB	00170	I	REFOCS	00171	I
IPERTB	00000	I						

COMMON BLOCK AZOBJ   ORIGIN 07734   LENGTH 00003

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
THTR	00000	R	SNTHTR	00001	R	CSTHTR	00002	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
STATN	07737	R	TEMPS	10677	R	GMX	10531	R

SUBROUTINE TO PERFORM SENSITIVITY ANALYSIS 02/16/68 PAGE 7

STAR8Z STORAGE MAP

GAMMA	11517	R	SJN	11524	R	T1	11533	R
DELTAT	11571	R	IGMX	10531	I	ISTATN	07737	I

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SYMB1	11627	I	SYMB2	11630	I	I	11631	I
INDX	11632	I	TMP	11633	R	BFN	11634	R
DUMMY	11635	R	J	11636	I	HEIGHT	11637	R
K	11640	I	MR	11641	I	KNDX	11642	I
PERCNT	11643	R	BF1	11644	R	SUMA	11645	R
SUMB	11646	R	FLAMDA	11647	R	XOBAR	11650	R
YOBAR	11651	R	XOSTAR	11652	R	YOSTAR	11653	R
ROSTAR	11654	R	ELAMDA	11655	R	IC[illegible]	11656	I
DXBAR	11657	R	DYBAR	11660	R	DX[illegible]	11661	R
DYSTAR	11662	R	DRSTAR	11663	R	NREFC	11664	I

ENTRY POINTS

STAR8 SECTION 13

SUBROUTINES CALLED

MERID	SECTION	14	SIN	SECTION	15	COS	SECTION	16
SMPOP	SECTION	17	PRTCTL	SECTION	18	.FWRD.	SECTION	19
SQRT	SECTION	20	.FXEM.	SECTION	21	.UNDS.	SECTION	22
.FFIL.	SECTION	23	.FCNV.	SECTION	24	E.1	SECTION	25
E.2	SECTION	26	E.3	SECTION	27	E.4	SECTION	28
CC.1	SECTION	29	CC.2	SECTION	30	CC.3	SECTION	31
CC.4	SECTION	32	SYSLOC	SECTION	33			

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
[illegible]00	119A	12672	8500	116A	12660	9000	134A	12763
32030	FORMAT	11756	32040	FORMAT	12007	9500	150A	13047
32050	FORMAT	12045	10200	191A	13250	9800	157A	13066
10000	176A	13201	13000	207A	13[illegible]0	32060	FORMAT	12085
10500	199A	13330	11000	201A	13[illegible]41	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.

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

SPLØT IV SUBROUTINES / S-C 4020 plotting and control for FORTRAN IV

Robert Jirka (JPL)

Lowell Smith (JPL)

Chuck Lawson (JPL)

PURPOSE

SPLOT 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

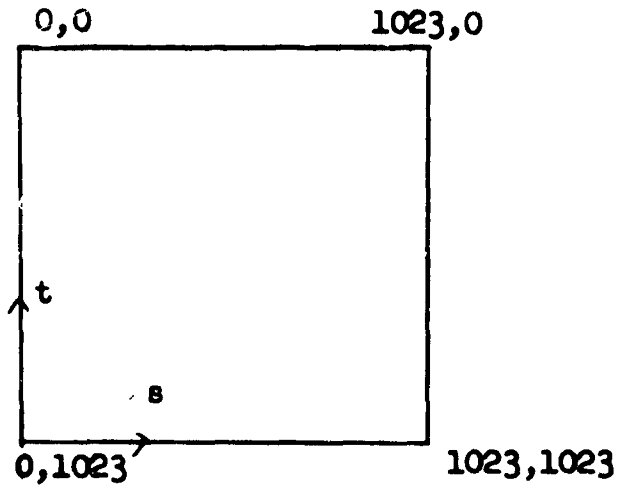


figure 1. S-C 4020 Plotting Surface

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_o, y_o) of the lower left corner and the coordinates (x_m, y_m) of the upper right corner.

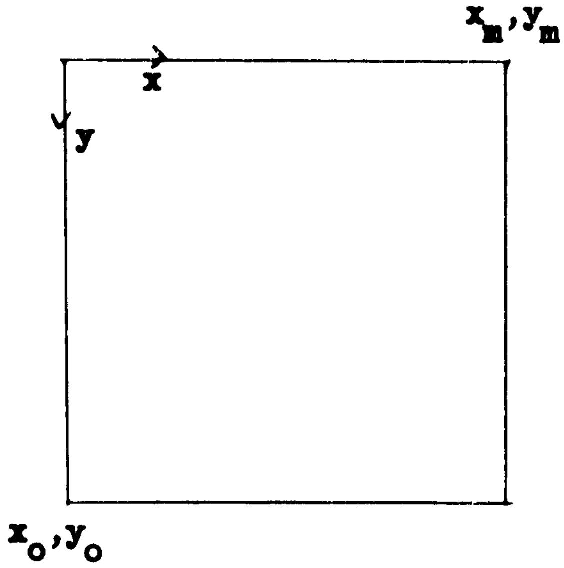


figure 2. Users Plotting Surface

For plotting curves everything will be relative to the users plotting surface.

To plot a curve the user supplies

1. The coordinates (x_o, y_o) and (x_m, y_m) for the corners of the plotting surface.
2. The points (s_o, t_o) and (s_m, t_m) on the S-C 4020 plotting surface at which to locate (x_o, y_o) 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) { SCALEK HAS BEEN ADDED TO LIBRARY -- SEE SEPARATE WRITEUP 67-41-01
5. CALL SDNPUT (X,Y,BUF,L)
6. CALL PPLØT (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 { NO LONGER REQUIRED IF UNIT IN THE WRITE STATEMENT REFERS TO UNIT 18
10. WRITE (UNIT,FMT) 'List'
11. CALL SPRINT (NT,NXL,NYL, $\pm$NS,NHX,...X_n,...)
12. CALL SPRNTA (NT,NXL,NYL,$\pm$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 FØRM (NT)
17. ~~CALL EXPØSE (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,PPLØT,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.

$X(1) = x_o$

$X(2) = y_o$

$X(3) = x_m$

$X(4) = y_m$

S is a one dimensional array containing the coordinates of points on the S-C plotting surface at which to locate the points (x_o,y_o) and (x_m,y_m).

(x_o,y_o) 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 FØRTRAN 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 <u>less than</u> 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 $x_o = x_m$ or $y_o = y_m$, the diagnostic

ERRØR IN CØØRDINATES DEFINING PLØT 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.

21 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 valves 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 (FØRTRAN 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 FØRTRAN 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, STERM 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, $2*10^p$, or $5*10^p$ 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 DINIT (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 DINIT 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 SDMPUT, 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. PPLØT: The FORTRAN statement

CALL PPLØT (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 SDNPUT 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 FØRTRAN 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. ØUTPUT: 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 SET) is 16 units.

All lines of printing start at (8,x) on the S-C 4020 plotting surface, where x is fixed either by ØUTPUT 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,CØUNT).

d. "blank" will add 16 to the line counter and start the line of print at the point (8,CØUNT).

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 SETPL must be used.

11. SPRINT: The FORTRAN statement

CALL SPRINT (NT,NXL,NYL,±NS,NHX1..Xn,...) will put on tape NT 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:

NT 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.

NHX1..Xn is the standard way of carrying Hollerith text in a parameter list.

The parameter list is variable and the quadruple (NXL,NYL,±NS, NHX1..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. SPRNTA: The FORTRAN statement

CALL SPRNTA (NT,NXL,NYL,$\pm$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:

NT,NXL,NYL,$\pm$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 FØRTRAN 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 $JØB 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. FØRM: The FORTRAN statement

CALL FØRM (NT) 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:

NT is not used and may be any dummy number.

~~17. EXPØSE: The FORTRAN statement~~ NOT AVAILABLE

CALL EXPØSE (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,SPRNTA,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. EXPØSE will effect PPLØT 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 BUF (2) = Y_o BUF (3) = X_m BUF (4) = Y_m	users coordinate system and set by SDINIT
BUF (5) = S_o BUF (6) = t_o BUF (7) = S_m BUF (8) = t_m	S-C 4020 coordinates and set by SDINIT
BUF (9) = S_n-1 BUF (10) = t_n-1	last S-C 4020 point used by SDNPUT
BUF (11) = A BUF (12) = B	defines transformation of x variable to S-C 4020 units, i.e., $S=A\cdot x+B$
BUF (13) = A_1 BUF (14) = B_1	defines transformation of y variable to S-C 4020 units, i.e., $T=A_1\cdot y+B$
BUF (15) =	number of S-C 4020 instructions input into BUF. The first instruction is at BUF (21). BUF (15) is changed by SDINIT, STØRE, ØUT, STERM.
BUF (16) =	the size of the array BUF and set by SDINIT. This is checked by STØRE
BUF (17) =	not used
BUF (18) = S_{n-2} BUF (19) = t_{n-2}	next to last S-C 4020 point used by SDNPUT
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;

$$Q = \max \left[\left| \frac{X_n - X_{n-1}}{60} \right| , \left| \frac{Y_n - Y_{n-1}}{60} \right| \right] + 1$$

$$X_p = X_{n-1} + \frac{P}{Q} (X_n - X_{n-1})$$

$$Y_p = Y_{n-1} + \frac{P}{Q} (Y_n - Y_{n-1}) \quad \text{for } P = 1, \ldots, Q$$

   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 liniar equations formed by letting

$$X(-S_1) = X_{n-2} \qquad Y(-S_1) = Y_{n-2}$$

$$X(S_2) = X_n \qquad Y(S_2) = Y_n$$

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 ie. 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. It 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 (IBSYS V/2)

FORTRAN WRITTEN

	Program	Locations$_{10}$	Subroutines Used	Comments
1.	XXNDT	146		This routine has the same calling sequence as SDINIT but is not variable length.
2.	SINTRP	24		
3.	SCALE	371		
4.	STØRE	57	ØUT	
5.	PLAB	455	ØRA, STØRE, RESRD, ALS, SETRD, SETPR, RESPR	
6.	ADV	36		
7.	CAMERA	40		
8.	FØRM	36		
9.	PPLØT	134	ØRA, ANA, ALS, ARS, STØRE	
10.	ØUT	87		
11.	ØUTPUT			This is the standard FORTRAN IV I/O package.
12.	SLABEL	59	GETJØB	
13.	EXPØSE	40		
14.	BGRID	368		

MAP WRITTEN

	Program	Locations$_{10}$	Subroutines Used	Comments
1.	GRID	124		
2.	STERM	28	ØUT	
3.	ALS (LØGIC) ANA ARS ØRA	20		
4.	SDINIT	43	XXNDT	
5.	SPRNTA SPRINT	340	SINTRP, SDNPUT, STERM	
6.	SDNPUT	787	STØRE	
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	FRCDA	
12.	FRCDA	19		An alternate FRCD subr.