NPS69-77-004

NAVAL POSTGRADUATE SCHOOL

Monterey, California

LASTOP - A COMPUTER CODE FOR LASER TURRET

OPTIMIZATION OF SMALL PERTURBATION TURRETS

IN SUBSONIC OR SUPERSONIC FLOW

by

G. N. VANDERPLAATS

A. E. FUHS

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Isham Linder
Superintendent

J. R. Borsting
Provost

LASTOP - A COMPUTER CODE FOR LASER TURRET OPTIMIZATION OF SMELL PERTURBATION TURRET IN SUBSONIC OR SUPERSONIC FLOW

A program has been developed which calculates optical path length and phase distortion arising from the density field surrounding a laser turret. Further, the program finds the optimum turret shape yielding minimum phase distortion. The aerodynamic model is briefly described; however, the optimization and control codes are thoroughly presented. Sample data input and sample output are given. The program is listed. The material is presented in detail so that this report constitutes a user's manual.

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS69-77-004	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* LASTOP - A Computer Code for Laser Turret Optimization of Small Perturbation Turrets in Subsonic or Supersonic Flow.		5. TYPE OF REPORT & PERIOD COVERED Final Report 1976-77
		6. PERFORMING ORG. REPORT NUMBER NPS69-77-004
7. AUTHOR(s) Garret N. Vanderplaats and Allen E. Fuhs		8. CONTRACT OR GRANT NUMBER(s) AF185 Project Order No. 77-050
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Lt. Col. K. Gilbert AFWL/LRO Kirtland AFB, NM 87117		12. REPORT DATE 20 December 1977
		13. NUMBER OF PAGES 231
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Laser turrets. Phase Distortion. Optimized turret shape. Aero-optics. Laser aerodynamics.

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

A program has been developed which calculates optical path length and phase distortion arising from the density field surrounding a laser turret. Further, the program finds the optimum turret shape yielding minimum phase distortion. The aerodynamic model is briefly described; however, the optimization and control codes are thoroughly presented. Sample data input and sample output are given. The program is listed. The material is presented in detail so that this report constitutes a user's manual.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

ABSTRACT

A program has been developed which calculates optical path length and phase distortion arising from the density field surrounding a laser turret. Further, the program finds the optimum turret shape yielding minimum phase distortion. The aerodynamic model is briefly described; however, the optimization and control codes are thoroughly presented. Sample data input and sample output are given. The program is listed. The material is presented in detail so that this report constitutes a user's manual.

TABLE OF CONTENTS

LIST OF FIGURES ---4

NOMENCLATURE-- 5

I. INTRODUCTION-- 6

II. PROGRAM ORGANIZATION-------------------------------------11

III. DATA TRANSFER--13

IV. PROGRAM DATA--16

A. COPES - A CONTROL PROGRAM FOR ENGINEERING SYNTHESES-----17

B. LASER TURRET ANALYSIS-------------------------------------39

V. SAMPLE DATA--54

A. GEOMETRY -- 54

B. AERODYNAMICS -- 55

C. MIRROR -- 55

D. BEAM ORIENTATIONS ------------------------------------- 55

E. PHASE DISTORTION -------------------------------------- 56

F. COPES DATA -- 56

G. TURRET ANALYSIS DATA ---------------------------------- 62

VI. SAMPLE OUTPUT--66

LIST OF REFERENCES--92

APPENDIX A - PROGRAM FLOW CHARTS AND FORTRAN VARIABLES------------93

APPENDIX B - PROGRAM LISTING------------------------------------103

APPENDIX C - DATA FORMS---231

LIST OF FIGURES

1. Small Perturbation Laser Turret on a Cylindrical Fuselage---------6

2. Turret Geometry --7

3. Laser Beam Orientation--8

4. Phase Distortion Calculation Within the Laser Beam----------------9

5. Program Organization---11

NOMENCLATURE

a_k	coefficients for the turret shape polynomial in x-direction
b_p	coefficients for the turret shape polynomial in the θ-direction
ℓ	extent of turret in x-direction; for $\lvert x \rvert > \ell$ radius of fuselage is R_0.
L	the distance 2L is separation between turrets
OPL	optical path length
PD	phase distortion; nondimensional
r	radial distance
R_D	fuselage radius
W_i	weighting factor for i-th beam direction
x	axial distance in cylindrical coordinates
X_M	axial location of mirror center
z	reference direction to measure angles within beam cross section
β^2	shorthand notation for $1 - M_\infty^2$
γ	beam elevation angle
ε	nondimensional turret height; R_0 is reference length
ε_M	radial location of mirror center
η	polar coordinate used to locate points or rays within the beam
θ	variable in cylindrical coordinates used to describe turret shape
ϕ	perturbation potential function; also, beam azimuth angle.

I. INTRODUCTION

A computer program is described here which obtains the optimum shape of a laser turret to minimuze optical distortion of a laser beam. The analysis and optimization procedure on which the program is based are described in detail in Ref. 1.

The turret is assumed to be situated on a cylindrical fuselage, as shown in Figure 1. The details of the turret geometry are shown in Figure 2. The shape of the turret is defined by the product of two polynomials, so that

$$r = \varepsilon f(x)\ f(\theta) \tag{1}$$

where

$$f(x) = 1 + \bar{a}_1\ x + \bar{a}_2\ x^2 + \ldots + \bar{a}_k x^k \tag{2}$$

and

$$f(\theta) = 1 + \bar{b}_2\ \theta^2 + \ldots + \bar{b}_p \theta^p \tag{3}$$

where p is the sequence of even numbers 2, 4, 6 . . . .

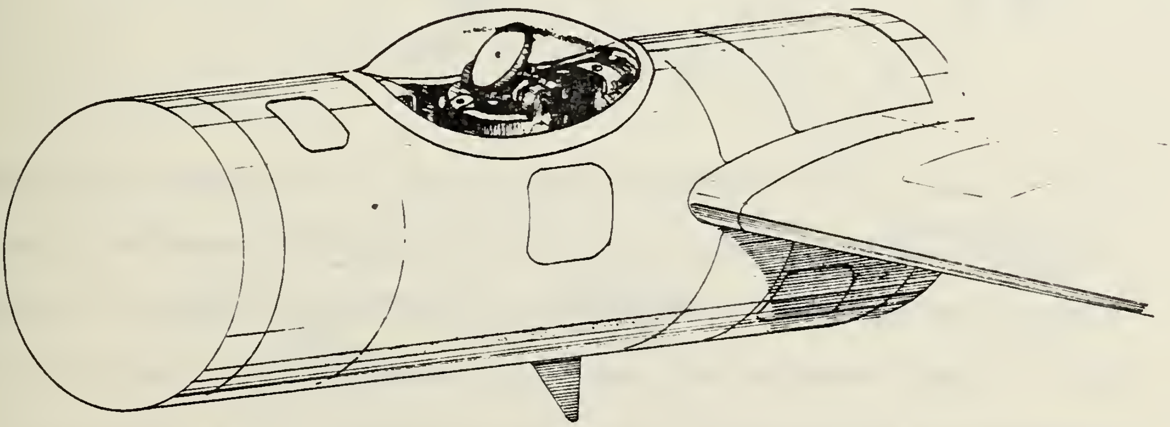


Figure 1. Small Perturbation Laser Turret on a Cylindrical Fuselage.

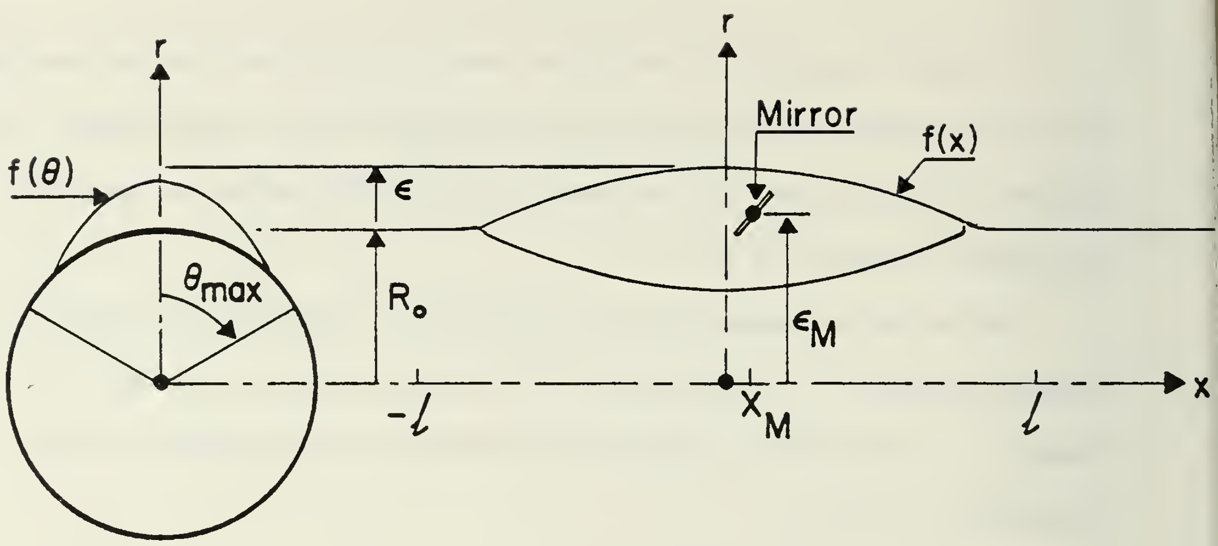


Figure 2. Turret Geometry.

Optional distortion is introduced into a laser beam propagating through the flow field surrounding the turret; see Ref. 2. For purposes of this analysis, the flow is assumed to be compressible and inviscid and is governed by the small perturbation equation

$$\pm\beta^2 \phi_{xx} + \phi_{rr} + \frac{\phi_r}{r} + \frac{\phi_{\theta\theta}}{r^2} = 0 \qquad (4)$$

The (+) sign applies to subsonic flow and the (-) sign applies to supersonic flow. The solution of equation (4) is valid for small perturbation subsonic and supersonic flow. For transonic flow the analysis is nonlinear, even for small perturbations, and is not considered here. Reference 2 discusses the formulation of the aerodynamics model for a variety of geometrical shapes and flow regimes.

From the solution of the potential equation, the perturbation velocities, u and v, may be calculated anywhere in the flow field. From knowledge of the flow field the optical path length on any ray of a laser beam is calculated. The laser beam is propagated through the flow field as shown in Figure 3. Taking the center of the beam as the reference ray, the difference in optical path lengths, OPL, between a specified ray and the ray on the beam center is calculated as

$$\Delta OPL = OPL_j - OPL_i \qquad (5)$$

where the subscript i corresponds to the reference ray and j corresponds to the particular ray being considered. The phase distortion, PD, is defined as $\Delta OPL/\lambda$ where λ is the wave length of radiation; Refs. 3 and 4 discuss OPL and PD in more detail.

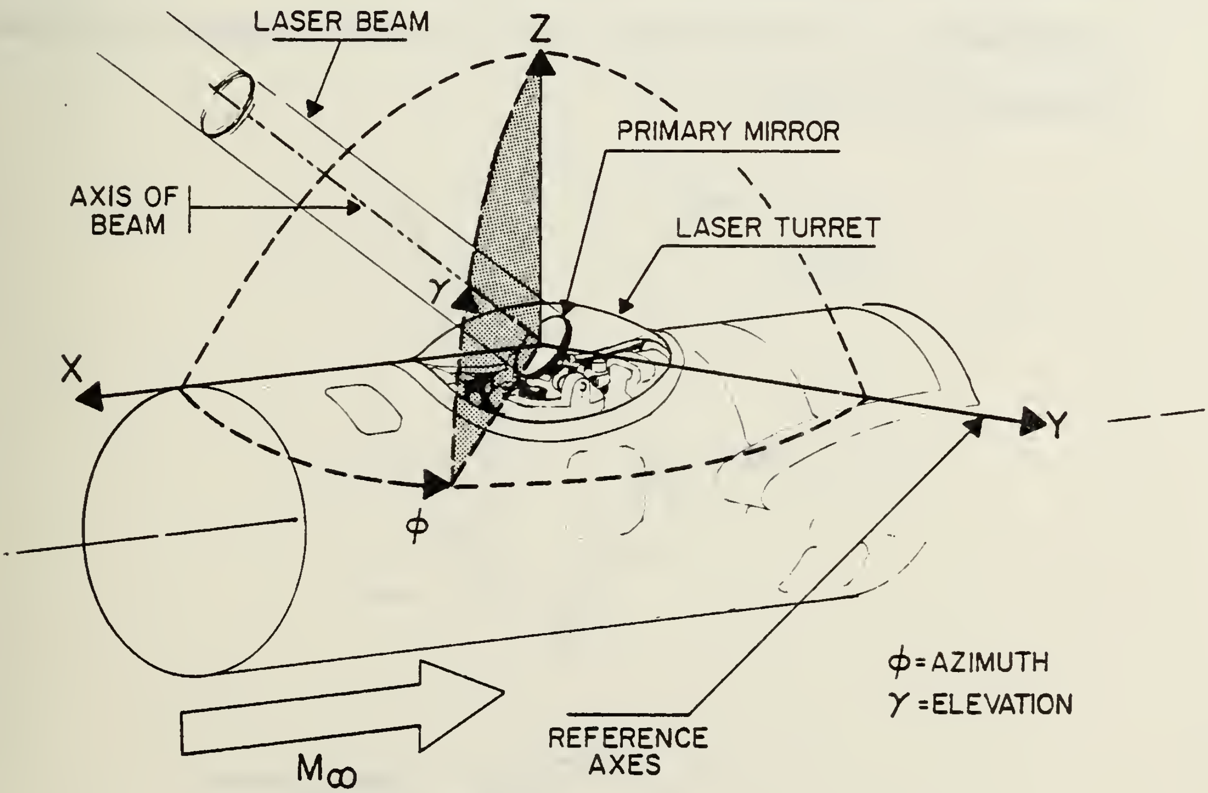


Figure 3. Laser Beam Orientation

Phase distortion, PD, is calculated numerically at several radial and angular locations within the beam as shown in Figure 4. The sum of $(PD)^2$ over all calculation points for several beam orientations is considered to provide a measure of the "goodness" of the turret design. The coefficients of the turret shape functions of equations 2 and 3 are then determined to minimize

$$SUMPD = \sum_{\text{orientations}} W_i \sum_{\text{radii}} \sum_{\text{angles}} (PD)^2 \quad (6)$$

where W_i is a weighting factor applied to the i-th beam orientation. The COPES/CONMIN optimization program (Ref.5) is used to provide the turret optimization capability.

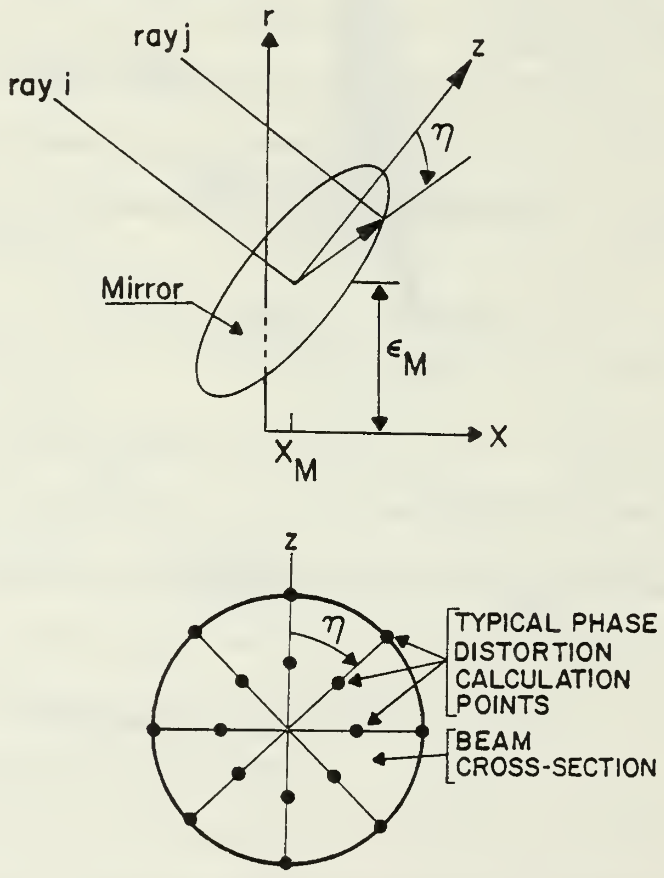


Figure 4. Phase Distortion Calculation Within the Laser Beam.

Finally, the optical aberrations are calculated for each beam orientation in terms of Zernicke coefficients. This provides a measure of the turret design in terms familiar to optical design specialists; see Ref. 6.

In the following sections, the program organization, data transfer mechanism, input data and output are described. Test cases are provided to help in making the program operational. Additional program details and a FORTRAN listing are included in the Appendices.

II. PROGRAM ORGANIZATION

The basic program organization is shown in block diagram form in Figure 5. The COPES program is the main driver which calls the optimization program, CONMIN, and the turret analysis program; COPES is an accronym for COntrol Program for Engineering Synthesis, and CONMIN is an accronym for CONstrained function MINimization. Both are general purpose programs which may be applied to a wide variety of engineering design problems (Ref. 7). If only the analysis of a specific turret shape is desired, this may be done without COPES/CONMIN by using a very simple main program. Alternatively, COPES/CONMIN may be used for a single analysis by specifying the proper value of a single control parameter in the input data.

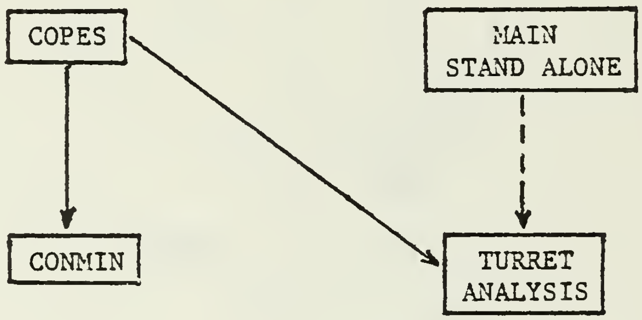


Figure 5. Program Organization.

The combined program containing COPES, CONMIN and laser turret analysis is referred to by the accronym LASTOP, for LASer Turret Optimizatio

The entire program is written in FORTRAN IV and has been executed, without modification, on IBM 360/67 and CDC 7600 computers. The program executes in approximately 50^k octal words of storage on a CDC computer.

The program reads from unit 5 and writes on unit 6. Units 20 and 40 are scratch files. (These file numbers may be changed by changing two cards at the beginning of the COPES program.) Execution times on a CDC 7600 computer are approximately 0.3 and 1.0 CPU seconds for subsonic and supersonic flow respectively for the <u>analysis</u> of <u>one</u> beam orientation. In a typical design optimization run, fifteen beam orientations may be considered. Assuming fifty candidate designs are analyzed before the optimum is obtained, the total CPU time is from 200 to 750 seconds.

To execute the turret optimization program, the user must be familiar with the mechanism by which data are transferred between analysis and design programs. This is the subject of the following section.

III. DATA TRANSFER

To couple the analysis and optimization programs for automated design optimization, pertinent data must be transferred between programs. This is done by means of a single labeled common block. To execute the program, it is necessary for the user to know what information is transferred and the location in common of that information. This section defines the data to be transferred and identifies their location within the common block.

The variables contained in the "GLOBAL" common block are listed below; the terms have the following meaning:

LOCATION - The physical location of the variable in the common block. For example, the polynomial coefficient $\bar{a}_1$ is in location 2 while $\bar{a}_2$ is in location 3. The usual design objective (phase distortion), SUMPD2, is in global location 169.

TYPE - The purpose of the variable in design optimization. D = design variable, S = sensitivity variable, O = objective function and C = constraint function. Note that a sensitivity variable may be a design variable if this is meaningful. For example, the direction of minimum phase distortion may be found by considering only one beam orientation and treating the azimuth angle (location 108) and elevation angle (location 78) as design variables. Similarly, objective and constraint functions are interchangeable. For example, the minimum turret half-length (location 21) may be found with an upper bound on phase distortio (location 169). Under special circumstances, the objective function may also be a design variable. For example, the maximum turret height (location 76) may be sought, subject to a constraint on maximum phase distortion (location 169). Because the turret height is intended as a design variable, it must also be a design variable here because it only appears on the right-hand side of equations in the program.

FORTRAN - The FORTRAN name of the variable used in the program.

MATH - The mathematical symbol for the variable (used in Ref. 1).

DEFINITION - Physical meaning of the variable.

LOCATION	TYPE	FORTRAN	MATH	DEFINITION
1-20	D	ABAR(20)	$\bar{a}_i$	Polynomial coefficients on f(x),
21	S	ACL	L	Turret half-spacing for Fourier analysis.
22	S	AKPRIM	k'	Constant in phase distortion calculations.
23	D,S	AL	ℓ	Turret half length divided by RFUS
24-53	S	AMACHI(30)	M_∞	Mach number associated with i-th beam orientation.
54-73	D	BBAR(20)	$\bar{b}_i$	Polynomial coefficients on f(θ).
74	S	DENRTO	ρ/ρ_{SL}	Density of air divided by density of air at sea level.
75	S	DENGAM	γ	Exponent in pressure-density relationship.
76	D,S	EPS	ε	Turret height divided by RFUS.
77	S	EPSM	ε_m	Mirror center height divided by RFUS.
78-107	S	GAMMAI(30)	γ	Elevation angle of i-th beam orientation.
108-137	S	PHII(30)	ϕ	Azimuth angle of i-th beam orientation.
138	S	RFUS	R_0	Fuselage radius (meters).
139-168	C	SLOPEX(30)	f'(x)	Slope of turret surface in streamwise direction.
169	O	SUMPD2	$\Sigma(PD)^2$	Sum of squares of all calculated phase distortions.
170	D,S	TDENRT	ρ/ρ_{SL}	Density of air inside canopy divided by density of air at sea level.
171	D,S	THMAX	θ_{MAX}	Half angle of turret (degrees).
172	S	WAVEL	λ	Wave length of radiation (meters).
173-202	S	WGHTI(30)	W_i	Weighting factor on i-th beam orientation.
203	S	XM	X_M	X-coordinate of center of mirror.

IV. PROGRAM DATA

The data for laser turret analysis and optimization are separated into two parts. First are the control program (COPES) data which control the analysis and design operations. These are followed by the turret analysis data.

When the program is being made operational or when only analysis is desired, the turret analysis program may be run, stand-alone using a simple driver program given in the subsection on laser turret analysis. In this case, the COPES data are omitted, and only the turret analysis data are provided.

Appendix C contains convenient data forms for both the COPES and the turret analysis data. The reader may want to copy these forms for use in preparing a problem.

A. COPES - A CONTROL PROGRAM FOR ENGINEERING SYNTHESES

The COPES program is a general purpose program to aid in design optimization and is not limited to the specific application for which it is used here. The user must provide an analysis program in subroutine form, which in this case is the analysis of a laser turret in subsonic and supersonic flow. The principal requirements are that the analysis program be coded in FORTRAN and be segmented into input, execution and output and that all design information be stored in a single labeled common block called GLOBCM.

The COPES program provides four specific capabilities:

1. Simple analysis - just as if COPES was not used.
2. Optimization - minimization or maximization of one calculated function with limits imposed on other functions.
3. Sensitivity analysis - the effect of changing one or more design variables on one or more calculated functions.
4. Two-variable function space - analysis for all specified combinations of two design variables.

COPES utilizes the general purpose optimization program CONMIN (Ref. 2) for optimization, and this is the capability of primary interest here. Data requirements for options 3 and 4 are included for completeness.

To better understand the COPES data requirements, the following definitions are useful:

Design Variables - Design variables are those parameters which the optimization program is allowed to change in order to improve the design. Design variables appear only on the right-hand side of an equation in the analysis program. COPES considers two types of design variables, independent and dependent. If two or more variables are always required to have the same value or be in a constant ratio, one is the independent variable while the remaining are dependent variables. For example, if the turret shape polynomials are required to be the same in both the x and θ directions, the coefficients $\bar{a}_i$ may be independent variables, and the $\bar{b}_i$ may be dependent variables. In this example, the total number of design variables will then be twice the number of independent design variables.

Objective Function - The parameter which is to be minimized or maximized during optimization is an objective function. Included are parameters calculated as a function of specified design variables during a sensitivity or two-variable function space study. Objective functions always occur on the left side of an equation unless the objective function is also a design variable.(The turret height may be maximized as an objective function if it is also a design variable. In this way, the maximum height is found for which no constraints are violated.) An objective function may be linear or non-linear and implicit or explicit but must be a continuous function of the design variables to be meaningful.

Constraint - Any parameter which must not exceed specified bounds for the design to be acceptable is a constraint. Constraint functions always appear on the left side of an equation. Just as for objective functions, constraints may be linear or non-linear and implicit or explicit but must be continuous functions of the design variables.

The COPES program reads from unit 5 and writes output on unit 6. Units 20 and 40 are used as scratch files. The scratch file numbers may be changed by changing two cards at the beginning of the COPES program.

The data required to run the COPES program are now defined. All GLOBAL LOCATION NUMBERS refer to the location of the specified variable in the labeled common block, GLOBCM. The pertinent variables and their global locations are listed in the section entitled DATA TRANSFER.

The data are segmented into "blocks" for convenience. All formats are alphanumeric for TITLE, END, and STOP cards; F10 for real data; and I10 for integer data. Comment cards may be inserted anywhere in the data stack prior to the END card and are identified by a dollar sign ($) in Column 1. The COPES data stack must terminate with an end card containing the word "END" in Columns 1-3.

Data coding forms are provided in Appendix C.

COPES

DATA BLOCK A

DESCRIPTION: Title Card

FORMAT AND EXAMPLE

1 2 3 4 5 6 7 8	FORMAT
TITLE	20A4
LASER TURRET OPTIMIZATION	

FIELD	CONTENTS
1-8	Any 80 character title

REMARKS

1) Program is terminated by the word 'STOP' in columns 1-4.

COPES

DATA BLOCK B

DESCRIPTION : Program Control Parameters

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NCALC	NDV	NSV	N2VAR	IPNPUT	IPSENS	IP2ZAR		7I10
2	4	3	2	0	0	0		

FIELD		CONTENTS
1	NCALC:	Calculation control
	0 -	Read input and stop. Data of blocks A-B ar required. Remaining data are optional.
	1 -	One cycle through program. Data of blocks A-B are required. Remaining data are optio
	2 -	Optimization. Data of blocks A-I are requi Remaining data are optional.
	3 -	Sensitivity analysis. Data of blocks A-B ai J-K are required. Remaining data are optioi
	4 -	Two variable function space. Data of blocks and L-O are required. Remaining data are optional.
2	NDV:	Number of independent design variables in optimization.
3	NSV:	Number of variables on which sensitivity analysis will be performed.
4	N2VAR:	Number of objective functions in a two variable function space study.
5	IPNPUT:	Input print control
	0 -	Print card images plus formated print of inp
	1 -	Formated print of input only.
	2 -	No print of input.
6	IPSENS:	Print control for sensitivity analysis. If IPSENS.GT.0 detailed print will be called fo at each step in the sensitivity analysis. DEFAULT = No print.
7	IP2VAR:	Print control for two variable function spac study. If IP2VAR.GT.0 detailed print will b called for at each step (each X-Y combinatio DEFAULT = No print.

REMARKS

1) Field 1 determines program execution.

2) Fields 2-4 identify which information will be read in subsequent data blocks.

DATA BLOCK C Omit if NDV = 0 in Block A

DESCRIPTION : Integer Optimization Control Parameters

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
IPRINT	ITMAX	ICNDIR	NSCAL	ITRM	LINOBJ	NACMX1	NFDG	8I10
5	0	0	5	0	0	0	0	

FIELD	CONTENTS	
1	IPRINT:	Print control used in optimization program, CONMIN.
	0 -	No print during optimization.
	1 -	Print initial and final optimization information.
	2 -	Print above plus function value and design variable values at each iteration.
	3 -	Print above plus constraint values, direction vector and move parameter at each iteration.
	4 -	Print above plus gradient information.
	5 -	Print above plus each proposed design vector, objective function and constraints during the one-dimensional search.
2	ITMAX:	Maximum number of optimization iterations allowed. DEFAULT = 20.
3	ICNDIR:	Conjugate direction restart parameter. DEFAULT = NDV+1.
4	NSCAL:	Scaling parameter. GT.0 - Scale design variables to order of magnitude one every NSCAL iterations. LT.0 - Scale design variables according to scaling values input. DEFAULT = No scaling.
5	ITRM:	Number of subsequent iterations which must satisfy relative or absolute convergence criterion before optimization process is terminated. DEFAULT = 3.
6	LINOBJ:	Linear objective function identifier. If the optimization objective is known to be a linear function of the design variables, set LINOBJ = 1. DEFAULT = Non-Linear.
7	NACMX1:	One plus the maximum number of active constraints anticipated. DEFAULT = NDV+2.

FIELD	CONTENTS	
8	NFDG:	Finite difference gradient identifier.
	0 -	All gradient information is computed by finite difference.
	1 -	Gradient of objective is computed analytically. Gradients of constraints are computed by finite difference.
	2 -	All gradient information is computed analytically.

REMARKS

1) For LASER TURRET OPTIMIZATION, the value of LINOBJ and NFDG should always be zero. The value of NSCAL = 5 is suggested and ITRM = NACMX1 = 0 should be used. The value of IPRINT may be reduced when the user is familiar with the optimization output.

DATA BLOCK D Omit if NDV = 0 in Block A

DESCRIPTION : Floating Point Optimization Program Parameters

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
FDCH	FDCHM	CT	CTMIN	CTL	CTLMIN	THETA	PHI	8F10
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
DELFUN	DABFUN							2F10
0.0	0.0							

Note: Two cards of data are read here.

FIELD	CONTENTS	
1	FDCH:	Relative change in design variables in calculating finite difference gradients. DEFAULT = 0.01.
2	FDCHM:	Minimum absolute step in finite difference gradient calculations. DEFAULT = 0.001.
3	CT:	Constraint thickness parameter. DEFAULT = -0.05.
4	CTMIN:	Minimum absolute value of CT considered in the optimization process. DEFAULT = 0.004.
5	CTL:	Constraint thickness parameter for linear and side constraints. DEFAULT = -0.01.
6	CTLMIN:	Minimum absolute value of CTL considered in the optimization process. DEFAULT = 0.001.
7	THETA:	Mean value of push-off factor in the method of feasible directions. DEFAULT = 1.0.
8	PHI:	Participation coefficient, used if one or more constraints are violated. DEFAULT = 5.0.
1	DELFUN:	Minimum relative change in objective function to indicate convergence of optimization process. DEFAULT = 0.001.
2	DABFUN:	Minimum absolute change in objective function to incate convergence of the optimization process. DEFAULT = 0.001 times the initial objective value.

REMARKS

1) For LASER TURRET OPTIMIZATION default values of these parameters usually work well.

DATA BLOCK E Omit if NDV = 0 in Block A

DESCRIPTION: Total Number of Design Variables, Design Objective Identification and Sign on Design Objective.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NDVTOT	IOBJ	SGNOBJ						2I10,F10
0	163	-1.0						

FIELD	CONTENTS	
1	NDVTOT:	Total number of variables linked to the design variables. NDVTOT must be greater than or equal to NDV. This option allows two or more parameters to be assigned to a single design variable. The value of each parameter is the value of the design variable times a multiplier which may be different for each parameter. DEFAULT = NDV.
2	IOBJ:	Global variable number associated with objective function in optimization.
3	SGNOPT:	Sign used on objective of optimization to identify whether function is to be maximized or minimized. +1.0 indicates maximization. -1.0 indicates minimization. DEFAULT = -1.0.

REMARKS

1) For LASER TURRET OPTIMIZATION, the numbers used in this example are correct if phase distortion is to be minimized. If phase distortion is to be maximized set SGNOPT = +1.0.

DATA BLOCK F Omit if NDV = 0 in Block A

DESCRIPTION : Design variable bounds, initial values and scaling factors.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
VLB	VUB	X	SCAL					4F10
-3.0	3.0	0.0	0.0					

Note: Read one card for each of the NDV independent design variables.

FIELD	CONTENTS	
1	VLB:	Lower bound on the design variable.
2	VUB:	Upper bound on the design variable.
3	X:	Initial value of the design variable. If X is non-zero, this will supercede the value initialized by subroutine ANALIZ.
4	SCAL:	Design variable scale factor. Not used if NSCAL.GE.0 in Block C.

REMARKS

1) For LASER TURRET OPTIMIZATION, the values used in this example are suggested.

COPES

DATA BLOCK G Omit if NDV = 0 in Block A.

DESCRIPTION: Design Variable Identification

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NDSGN	IDSGN	AMULT						2I10,F10
1	26	1.0						

Note: Read one card for each of the NDVTOT Design Variables.

FIELD	CONTENTS
1	NDSGN: Design variable number associated with the variable.
2	IDSGN: Global variable number associated with the variable.
3	AMULT: Constant multipliter on the variable. The value of the variable will be the value of the design variable, NDSGN times AMULT. DEFAULT = 1.0.

COPES

DATA BLOCK H Omit if NDV = 0 in Block A

DESCRIPTION : Number of sets of constrained parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NCONS								I10
1								

FIELD CONTENTS

1 NCONS: Number of constraint sets in the optimization problem.

REMARKS

1) If two or more adjacent parameters in the Global common block have the same limits imposed, these are part of the same constraint set.

DATA BLOCK I Omit if NDV = 0 in Block A or if NCONS = 0 in Block M.

DESCRIPTION : Constraint Identification and Bounds.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
ICON	JCON	LCON						3I10
224	234	1						
BL	SCAL1	BU	SCAL2					4F10
-.3	.3	.3	.3					

Note: Read two cards for each of the NCONS constraint sets.

FIELD	CONTENTS	
1	ICON:	First Global number corresponding to the constraint set.
2	ICON:	Last Global number corresponding to the constraint set. DEFAULT = ICON.
3	LCON:	Linear constraint identifier for this set of constrained variables. LCON = 1 indicates linear constraints. DEFAULT = 0 = Nonlinear constraint.
1	BL:	Lower bound on the constrained variables. Value less than -1.0E+15 is assumed unbounded.
2	SCAL1:	Normalization factor on lower bound. DEFAULT = Max of ABS(BL), 0.1.
3	BU:	Upper bound on the constrained variables. Value greater than 1.0E+15 is assumed unbounded.
4	SCAL2:	Normalization factor on upper bound . DEFAULT = Max of ABS(BU), 0.1.

REMARKS

1) The normalization factors should usually be defaulted.

COPES

DATA BLOCK J Omit if NSV = 0 in Block A

DESCRIPTION : Sensitivity Objectives.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NSOBJ								I10
4								
NSN1	NSN2	NSN3	NSN4					8I10
26	27	41	42					

Note: Two or more cards are read here.

FIELD		CONTENTS
1	NSOBJ:	Number of separate objective functions to be calculated as functions of the sensitivity variables.
1-8	NSNI:	Global variable number associated with the sensitivity objective functions.

REMARKS

1) More than eight sensitivity objectives are allowed. Add data cards as required to contain data.

DATA BLOCK K Omit if NSV = 0 in Block A

DESCRIPTION : Sensitivity Variables

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
ISENS	NSENS							2I10
26	4							
SNS1	SNS2	SNS3	SNS4	. . .	. . .			8F10
2.0	1.0	3.0	4.0					

Note: Read one set of data for each of the NSV sensitivity variables.

Note: Two or more cards are read here.

FIELD	CONTENTS	
1	ISENS:	Global variable number associated wit[h] the sensitivity variable.
2	NSENS:	Number of values of the sensitivi[ty] variable to be considered.
1-8	SNSI:	Values of the sensitivity variable, f[or] J = 1, NSENS. J = 1 corresponds to nominal value.

REMARKS

1) More than eight values of the sensitivity variable are allowed. Add data cards as required to contain data.

DATA BLOCK L Omit if N2VAR = 0 in Block A

DESCRIPTION : Two variable function space control parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
N2VX	M2VX	N2VY	M2VY					4I10
26	5	27	5					

FIELD	CONTENTS
1	N2VX: Global location of X-variable in two-variable function space.
2	M2VX: Number of values of X-variable to be considered.
3	N2VY: Global location of Y-variable in two-variable function space.
4	M2VY: Number of values of Y-variable to be considered.

DATA BLOCK M Omit if N2VAR = 0 in Block A

DESCRIPTION : Objective Functions of Two-variable Function Space Study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NZ1	NZ2	NZ3	NZ4	. . .	. . .			8I10
7	4	21	67					

FIELD	CONTENTS
1-8	NZI: Global variable location corresponding to ITH function of X and Y in two variable function space.
	I = 1, NZVAR, where NZVAR is read in Bl

REMARKS

1) More than eight objective functions are allowed. Add data cards as required to contain data.

COPES

DATA BLOCK N Omit if N2VAR = 0 in Block A

DESCRIPTION : Values of X-variable in Two-variable Function Space Study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
X1	X2	X3	X4	. . .	. . .			8F10
0.5	1.0	1.5	2.0					

FIELD	CONTENTS
1-8	XI: Values of X-variable to be considered in two-variable function space. I = 1, MZVX, where MZVX is read in Block L.

REMARKS

1) More than eight X-values are allowed. Add data cards as required to contain data.

DATA BLOCK O Omit if N2VAR = 0 in Block A

DESCRIPTION : Values of Y-variable in two-variable Function Space Study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
Y1	Y2	Y3	. .	. .	. .			8F10
0.0	-1.0	1.0						

FIELD	CONTENTS
1-8	YI: Values of Y-variable to be considered in two-variable function space. I = 1, MZVY, where MZVY is read in Block.

REMARKS

1) More than eight Y-values are allowed. Add data cards as required to contain data.

COPES

DATA BLOCK P

DESCRIPTION : Copes data 'END' card.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
END								3A1
END								

FIELD	CONTENTS
1	The word 'END' in columns 1-3.

REMARKS

1) This card must appear at the end of the COPES data.

2) This ends the COPES input data.

B. LASER TURRET ANALYSIS

Data for the laser turret analysis follow the COPES data. If the general design capability of COPES is not needed, the analysis program can be run by itself by using the following simple main program.

```
C    MAIN PROGRAM FOR STAND ALONE LASER TURRET ANALYSIS.
C
C -  INPUT
     ICALC = 1
     CALL ANALIZ(ICALC)
C
C -  EXECUTION AND OUTPUT.
     ICALC = 3
     CALL ANALIZ(ICALC)
     STOP
     END
```

If this main progam is used, the COPES and CONMIN routines are omitted, and the COPES data are not read. This provides simple analysis of a specified turret and allows the turret analysis program to be tested independently.

The turret analysis program reads from unit 5 and writes the output on unit 6.

The input data are segmented into blocks for convenience, just as for the COPES data.

Comment cards are not allowed in the turret analysis data.

Data coding forms are provided in Appendix C.

TURRET

DATA BLOCK A

DESCRIPTION: Title Card.

FORMAT AND EXAMPLE

1 2 3 4 5 6 7 8	FORMAT
TITLE	20A4
LASER TURRET ANALYSIS	

FIELD	CONTENTS
1-8	Title : Any 80 character title.

TURRET

DATA BLOCK B

DESCRIPTION : Aerodynamics, Optics constants

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
AMACH	DENRTO	TDENRT	DENGAM	AKPRIM	WAVEL			6F10
1.25	.25	.25	1.405	.00023	3.4-6			

FIELD	CONTENTS
1	AMACH: Freestream Mach number.
2	DENRTO: Freestream air density/sea level density
3	TDENRT: Air density inside turret/sea level density
4	DENGAM: Exponent in pressure-density relationship $\frac{p}{p_o} = (\frac{\rho}{\rho_o})^{\gamma}$
5	AKPRIM: Phase distortion constant, k'
6	WAVEL: Wave length of radiation, λ (meters)

REMARKS

1) AMACH is the freestream MACH number for all beam orientations unless specified otherwise in data Block N.

TURRET

DATA BLOCK C

DESCRIPTION: Turret Geometry

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
RFUS	AL	THMAX	ACL	EPS				5F10
2.5	2.0	60.	10.	0.3				

FIELD	CONTENTS	
1	RFUS:	Fuselage Radius (meters)
2	AL:	Turret half length divided by RFUS.
3	THMAX:	Half angle subtended by turret (deg.)
4	ACL:	Half spacing between turrets divided by RFUS, for Fourier Series calculations.
5	EPS:	Turret height divided by RFUS at x = r = 0.

REMARKS

1) ACL must be much larger for supersonic flow then for subsonic flow to avoid interference between turrets. ACL = 5. is adequate for subsonic flow calculations.

TURRET

DATA BLOCK D

DESCRIPTION: Turret Geometry

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
MAXK	MAXP	NXBC	NTHBC					4I10
6	6	2	1					

FIELD / CONTENTS

1 MAXK: Order of x-polynomial shape function.
$f(x) = 1 + \bar{a}_1 x + .. \bar{a}_{maxk} x^{maxk}$

2 MAXP: Order of polynomial shape function.
$f(\theta) = 1 + \bar{b}_1 \theta + .. \bar{b}_{maxp} \theta^{maxp}$

3 NXBC: Number of sets of y and y' boundary conditions in x-direction, externally imposed.

4 NTHBC: Number of sets of θ and θ' boundary conditions in θ-direction, externally imposed.

REMARKS

1) The order plus one of each polynomial must be at least as great as the actual number of externally imposed boundary conditions.

TURRET

DATA BLOCK E

DESCRIPTION : Polynomial coefficients in x-direction.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
1	ABAR1	. .	. -			ABAR6	. . .	8F10
1.	0.	-.61111	0.	-.18056	0.	-.006944		

FIELD	CONTENTS	
1-8	ABARI	Coefficient of x-polynomial shape function, $f(x) = 1 + \bar{a}_1 x + \ldots \bar{a}_{maxk} x^{maxk}$

REMARKS

1) The total number of coefficients equals 1 + MAXK. Additional data cards are used as required to contain the data.

TURRET

DATA BLOCK F

DESCRIPTION : Geometric boundary conditions in x-direction.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
X	YBC	YPBC						3F10
-1.	0.	0.						

Note: NXBC cards are required.

FIELD	CONTENTS	
1	X:	X-location as fraction of turret half-length, AL, where boundary conditions is imposed.
2	YBC:	Required value of f(x) at x.
3	YPBC:	Required value of f'(x) at x.

REMARKS

1) The boundary condition that $f(x,\theta) = EPS$ at $x = \theta = 0$ is automatically imposed.

2) If YBC or YPBC is input greater than or equal 200., the corresponding bounda condition is omitted, i.e., if YPBC = 200., no boundary condition is imposed on f'(x).

TURRET

DATA BLOCK G

DESCRIPTION: Polynomial coefficients in θ-directions.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
1	BBAR1	BBAR2	. . .	. . .	. . .	BBAR6	...	8F10
1.	0.	-.61111	0.	-.18056	0.	-.006944		

FIELD — CONTENTS

1-8 — BBARI: Coefficient of θ polynomial shape function, $f(\theta) = 1 + \bar{b}_1\theta + \bar{b}_{maxp}\,\theta^{maxp}$

REMARKS

1) The total number of coefficients equals 1 + MAXP. Additional data cards are used as required to contain the data.

TURRET

DATA BLOCK H

DESCRIPTION : Geometric boundary conditions in θ-direction.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
THETA	YBC	YPBC						3F10
1.	0.	0.						

Note: NTHBC cards are required.

FIELD	CONTENTS
1	THETA: θ-location divided by turret half angle, THMAX, where the boundary condition is imposed.
2	YBC: Required value of f(θ) at THETA.
3	YPBC: Required value of f'(θ) at THETA.

REMARKS

1) The boundary condition that f(x,θ) = EPS at x = θ = 0 is automatically imposed.
2) If YBC or YPBC is input greater than or equal 200., the corresponding boundary condition is omitted, i.e., if YPBC = 200., no boundary condition is imposed on f'(θ).
3) Symmetry about θ = 0 is automatically imposed.

TURRET

DATA BLOCK I

DESCRIPTION : Mirror location.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
EPSM	XM							2F10
1.15	0.							

FIELD	CONTENTS	
1	EPSM:	Distance from fuselage axis to mirror center divided by RFUS.
2	XM:	x-coordinate of mirror center divided by RFUS.

REMARKS

1) Mirror is along fuselage centerline, $\theta = 0$.

TURRET

DATA BLOCK J

DESCRIPTION: Number of angular and radial locations on beam where phase distortion is to be calculated.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NETAI	NRBI							2I10
8	2							

FIELD	CONTENTS
1	NETAI: Number of angular points at which phase distortion is calculated.
2	NRBI: Number of radial points at which phase distortion is calculated.

TURRET

DATA BLOCK K

DESCRIPTION: Angles around beam at which phase distortion is calculated.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
ETA1	ETA2	ETA3	. . .	. . .	ETA6	. . .	. . .	8F10
0.	45.	90.			225.			

FIELD	CONTENTS
1-8	ETAI: Angle at which phase distortion is calculated in the laser beam.

REMARKS

1) If more than eight angular locations are considered, use additional data cards to contain the data.
2) Phase distortion is calculated at each combination of angular and radial locations.

TURRET

DATA BLOCK L

DESCRIPTION: Radial locations in beam at which phase distortion is calculated.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
RB1	RB2	. . .	RB4	. .	. .	. .	. .	8F10
0.025	0.05		0.1					

FIELD CONTENTS

1-8 RBI: Radial location in laser beam at which phase distortion is calculated.

REMARKS

1) If more than eight radial locations are considered, use additional data cards to contain the data.

2) Phase distortion is calculated at each combination of angular and radial locations.

TURRET

DATA BLOCK M

DESCRIPTION: Number of separate beam orientations to be analyzed.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NBEAM								I10
10								

FIELD	CONTENTS
1	NBEAM: Number of beam orientations considered in the analysis.

TURRET

DATA BLOCK N

DESCRIPTION: Beam orientation information.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
PHI	GAMMA	AMACHI	WGHT					4F10
30.	45.	1.4	1.					

Note: NBEAM cards are required.

FIELD	CONTENTS	
1	PHI:	Beam Azimuth angle. Measured from aircraft nose positive to the right. (degrees)
2	GAMMA:	Beam elevation angle. Measured from the horizontal plane, positive upward (degrees)
3	AMACHI:	Flight Mach number for this beam orientation. May be different then AMACH read in DATA Block B. DEFAULT = AMACH.
4	WGHT:	Weighting factor which multiplies the phase distortion for this beam orientation. Measure of relative importance to the design objective. DEFAULT = 1.0.

REMARKS

1) If AMACHI is read as zero, it is set equal to AMACH read in DATA Block B.

2) If WGHT is read as zero, it is set to 1.0.

3) This ends the input data for laser turret analysis.

V. SAMPLE DATA

Assume the turret shown in Figure 1 is to be analyzed or designed. The initial geometry of the turret, together with the aircraft flight and beam orientation information, is listed here.

A. GEOMETRY

R_0 = 2.5 meters = fuselage radius.

ε = 0.3 = turret height relative to R_0.

ℓ = 2.0 = turret half-length relative to R_0.

L = 10.0 = turret half-spacing for Fourier series approx.

θ_{max} = 60.0 degrees = turret half angle.

$f(x) = 1.0 - 0.6111\,1x^2 - 0.180\,56x^4 - 0.006944\,x^6$ = shape function in X.

$f(\theta) = f(x)$ = shape function in θ [initially the same as f(x)].

Boundary conditions imposed in this example are that $f(x) = f'(x) = 0$ at $x/\ell = \pm 1.0$, and $f(\theta) = f'(\theta) = 0$ at $\theta/\theta_{max} = 1.0$. The boundary condition that $f(x,\theta) = 0.3 = \varepsilon$ at $x = \theta = 0$ is automatically imposed by the program.

A total of five boundary conditions is imposed on f(x) so that $\bar{a}_0 - \bar{a}_4$ are computed by the analysis program and may not be design variables. Six boundary conditions are imposed on $f(\theta)$ (including symmetry requirements) so that only $\bar{b}_6$ may be treated as a design variable. The three design variables available for optimization in this example are

Variable	Global Location
$\bar{a}_5$	6
$\bar{a}_6$	7
$\bar{b}_6$	60

Because the aerodynamic analysis is based on small perturbation theory, it is only valid if the slope of the turret in the x-direction is small.

Therefore, constraints are imposed on the design so that the turret shape contained in vector SLOPEX is less than 0.3 in magnitude. That is

$$-0.3 \leq \text{SLOPE}(\bar{I}) \leq 0.3 \qquad I = 1,30$$

SLOPEX is stored in global locations 139 - 168 inclusive.

B. AERODYNAMICS

The aircraft is assumed to fly at sea level, and the turret is not pressurized so that

DENRTO = TDENRT = 1.0

The aerodynamic and optical constants are

DENGRAM = 1.405

AKPRIM = 0.00023

WAVEL = 3.4×10^{-6} infrared radiation

AMACH = 0.7 nominal. Mach number

C. MIRROR

The mirror is situated at

XM = 0.0

EPSM = 1.15

D. BEAM ORIENTATIONS

Three orientations are considered as follows:

Beam	Azimuth (PHI)	Elevation (GAMMA)
1	0.	50.
2	45.	30.
3	90.	10.

For brevity only three beam orientations are considered here. Typically fifteen orientations are used for optimization.

E. PHASE DISTORTION

The phase distortion is calculated at all combinations of two radial and eight angular positions.

R = o.05, 0.10 relative to R_0.

η = o, 45, 90, 135, 180, 225, 270, 315 degrees

Note that since the maximum value of R is 0.10, this is the assumed radius of the mirror.

F. COPES DATA

Based on the above requirements, the COPES data are listed here on a data sheet reproduced from APPENDIX C. These data are for a complete optimization. If only a simple analysis is desired, these data may be run by changing NCALC in DATA BLOCK B to 1 instead of NCALC = 2 given here.

COPES DATA

DATA BLOCK A

	TITLE	FORMAT
*	LASER TURRET OPTIMIZATION AT M=0.7	20A4

DATA BLOCK B

+	$ BLOCK B – CONTROL PARAMETERS								COMMENT
	NCALC	NDV	NSV	N2VAR	IPNPUT	IPSENS	IP2VAR		FORMAT
*	2	3							8I10

DATA BLOCK C - OMIT IF NDV = 0

+	$ BLOCK C - CONMIN INTEGER PARAMETERS								COMMENT
	IPRINT	ITMAX	ICNDIR	NSCAL	ITRM	LINOBJ	NACMX1	NFDG	FORMAT
*	5								8I10

DATA BLOCK D - OMIT IF NDV = 0

+	$ BLOCK D - CONMIN REAL PARAMETERS								COMMENT
	FDCH	FDCHM	CT	CTMIN	CTL	CTLMIN	THETA	PHI	FORMAT
*	0.								8F10
	DELFUN	DABFUN							FORMAT
*	0.								2F10

DATA BLOCK E - OMIT IF NDV = 0

+	$ BLOCK E - MINIMIZE PHASE DISTORTION				COMMENT
	NDVTOT	IOBJ	SGNOBJ		FORMAT
*	3	169	-1.0		2I10,F10

DATA BLOCK F - OMIT IF NDV = 0

+	$ BLOCK F - DESIGN VARIABLE LIMITS					COMMENT
	VLB	VUB	X	SCAL		FORMAT
*	$ COEF	FICIEN	T A-5			4F10
	-3.0	3.0				
	$ COEF	FICIEN	T A-6			
	-3.0	3.0				
	$ COEF	FICIEN	T B-6			
	-3.0	3.0				

COPES DATA CONT.

DATA BLOCK G - OMIT IF NDV = 0

+	$ BLOCK G - DESIGN VARIABLE IDENTIFICATION				COMMENT
	NDSGN	IDSGN	AMULT		FORMAT
*	$ COEFFICIENT A-5				2I10,F10
	1	6	1.0		
	$ COEFFICIENT A-6				
	2	7	1.0		
	$ COEFFICIENT B-6				
	3	60	1.0		

DATA BLOCK H - OMIT IF NDV = 0

+	$ BLOCK H - CONSTRAINTS		COMMENT
	NCONS		FORMAT
*	1		I10

DATA BLOCK I - OMIT IF NDV = 0 OR NCONS = 0

+	$ CONSTRAINT ON SLOPE.					COMMENT
	ICON	JCON	LCON			FORMAT
*	139	268	1			3I10
+	$ LIMITED TO SMALL PERTURBATION THEORY					COMMENT
	BL	SCAL1	BU	SCAL2		FORMAT
*	-0.3	0.3	0.3	0.3		4F10

DATA BLOCK I — CONT.

DATA BLOCK J — OMIT IF NSV = 0

+	$								COMMENT
	NSOBJ								FORMAT
*									I10
+	$								COMMENT
	NSN1	NSN2	NSN3	NSN4	NSN5	NSN6	NSN7	NSN8	FORMAT
*									8I10

DATA BLOCK K — OMIT IF NSV = 0

+	$								COMMENT
	ISENS	NSENS							FORMAT
*									2I10
+	$								COMMENT
	SNS1	SNS2	SNS3	SNS4	SNS5	SNS6	SNS7	SNS8	FORMAT
*									8F10

+	$								COMMENT
	ISENS	NSENS							FORMAT
*									2I10
+	$								COMMENT
	SNS1	SNS2	SNS3	SNS4	SNS5	SNS6	SNS7	SNS8	FORMAT
*									8F10

COPES DATA - CONT.

DATA BLOCK K - CONT.

+	$								COMMENT
	ISENS	NSENS							FORMAT
*									2I10
+	$								COMMENT
	SNS1	SNS2	SNS3	SNS4	SNS5	SNS6	SNS7	SNS8	FORMAT
*									8F10

+	$								COMMENT
	ISENS	NSENS							FORMAT
*									2I10
+	$								COMMENT
	SNS1	SNS2	SNS3	SNS4	SNS5	SNS6	SNS7	SNS8	FORMAT
*									8F10

DATA BLOCK L - OMIT IF N2VAR = 0

+	$					COMMENT
	N2VX	M2VX	N2VY	M2VY		FORMAT
*						4I10

DATA BLOCK M - OMIT IF N2VAR = 0

+	$								COMMENT
	NZ1	NZ2	NZ3	NZ4	NZ5	NZ6	NZ7	NZ8	FORMAT
*									8I10

DATA BLOCK N - OMIT IF N2VAR = 0

+	$								COMMENT
	X1	X2	X3	X4	X5	X6	X7	X8	FORMAT
*									8F10

DATA BLOCK O - OMIT IF N2VAR = 0

+	$								COMMENT
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	FORMAT
*									8F10

DATA BLOCK P

	END		FORMAT
*	END		3A1

SUMMARY OF COPES DATA

LISTING OF DATA AS IT APPEARS ON PUNCHED CARDS

```
COL.→ 1        10        20        30        40        50 ....
      LASER TURRET OPTIMIZATION AT M = 0.7
      $ BLOCK B - CONTROL PARAMETERS
               2         3
      $ BLOCK C - CONMIN INTEGER PARAMETERS
               5
      $ BLOCK D - CONMIN REAL PARAMETERS. USE ALL DEFAULTS.
      0.
      0.
      $ BLOCK E - MINIMIZE PHASE DISTORTION
               3       169  -1.0
      $ BLOCK F - DESIGN VARIABLE LIMITS
      $ COEFFICIENT A - 5
      -3.0      3.0
      $ COEFFICIENT A - 6
      -3.0      3.0
      $ COEFFICIENT B - 6
      -3.0      3.0
      $ BLOCK G - DESIGN VARIABLE IDENTIFICATION
      $ COEFFICIENT A - 5
               1         6 1.0
      $ COEFFICIENT A - 6
               2         7 1.0
      $ COEFFICIENT B - 6
               3        60 1.0
      $ BLOCK H - CONSTRAINTS
               1
      $ BLOCK I - CONSTRAINT ON SLOPE
             139       168         1
      $ LIMITED TO SMALL PERTURBATION THEORY
      -0.3      0.3       0.3       0.3
      $ BLOCK P - END OF COPES DATA
      END
```

G. TURRET ANALYSIS DATA

The data required to analyze the laser turret described above are listed here on a data sheet reproduced from APPENDIX C. Note that the Mach number for each beam orientation (BLOCK N) is read as zero so that all beam orientations will be analyzed at the nominal Mach number of 0.7. If another run is desired at a different Mach number, only AMACH (BLOCK B) need be changed. If certain beam orientations are to be analyzed at different Mach numbers, the appropriate value should be read in BLOCK N.

LASER TURRET ANALYSIS DATA

DATA BLOCK A

	TITLE	FORMAT
*	SUBSONIC LASER TURRET AT SEA LEVEL	20A4

DATA BLOCK B

	AMACH	DENRTO	TDENRT	DENGAM	AKPRIM	WAVEL		FORMAT
*	0.7	1.0	1.0	1.405	.00023	3.4 -6		6F10

DATA BLOCK C

	RFUS	AL	THMAX	ACL	EPS		FORMAT
*	2.5	2.0	60.	5.	0.3		5F10

DATA BLOCK D

	MAXK	MAXP	NXBC	NTHBC		FORMAT
*	6	6	2	1		4I10

DATA BLOCK E

	ABAR0	ABAR1	ABAR2	ABAR3	ABAR4	ABAR5	ABAR6	ABAR7	FORMAT
*	1.0	0.	-0.61111	0.	-0.18056	0.	-0.006944		8F10

DATA BLOCK F

	X	YBC	YPBC		FORMAT
*	-1.0	0.	0.		3F10
	1.0	0.	0.		

DATA BLOCK G

	BBAR0	BBAR 1	BBAR2	BBAR3	BBAR4	BBAR5	BBAR6	BBAR7	FORMAT
*	1.0	0.	-0.61111	0.	-0.18056	0.	-0.006944		8F10

DATA BLOCK H

	THETA	YBC	YPBC		FORMAT
*	1.0	0.	0.		3F10

LASER TURRET ANALYSIS DATA - CONT.

DATA BLOCK I

	EPSM	XM		FORMAT
*	1.15	0.		2F10

DATA BLOCK J

	NETAI	NRBI		FORMAT
*	8	2		2I10

DATA BLOCK K

	ETA1	ETA2	ETA3	ETA4	ETA5	ETA6	ETA7	ETA8	FORMAT
*	0.	45.	90.	135.	180.	225.	270.	315.	8F10

DATA BLOCK L

	RB1	RB2	RB3	RB4	RB5	RB6	RB7	RB8	FORMAT
*	0.05	0.1							8F10

DATA BLOCK M

	NBEAM		FORMAT
*	3		I10

DATA BLOCK N

	PHI	GAMMA	AMACHI	WGHT		FORMAT
*	0.	50.				4F10
	45.	30.				
	90.	10.				

SUMMARY OF TURRET ANALYSIS DATA

LISTING OF DATA AS IT APPEARS ON PUNCHED CARDS

BLOCK \ COL.→	1	10	20	30	40	50	60 ....
A	SUBSONIC LASER TURRET AT SEA LEVEL						
B	0.7	1.0	1.0	1.405	0.00023		3.4-6
C	2.5	2.0	60.0	10.0	0.3		
D		6	6	2	1		
E	1.0	0.	-.61111	0.	-.18056	0.	-.006944
F	-1.0	0.	0.				
F	1.0	0.	0.				
G	1.0	0.	-.61111	0.	-.18056	0.	-.006944
H	1.0	0.	0.				
I	1.15	0.					
J		8	2				
K	0.	45.	90.	135.	180.	225.	270.
						(end of card)	315.
L	0.05	0.1					
M		3					
N	0.	50.					
N	45.	30.					
N	90.	10.					
A	STOP ← (New COPES Data Title Card To Terminate Program After This Run)						

VI. SAMPLE OUTPUT

```
CCCCCCC   0000000   PPPPPPP   EEEEEEE   SSSSSSS
C         0     0   P     P   E         S
C         0     0   P     P   E         S
C         0     0   PPPPPPP   EEEE      SSSSSSS
C         0     0   P         E               S
C         0     0   P         E               S
CCCCCCC   0000000   P         EEEEEEE   SSSSSSS
```

N A S A - A M E S

C O N T R O L   P R O G R A M

F O R

E N G I N E E R I N G   S Y N T H E S I S

T I T L E

LASER TURRET OPTIMIZATION AT M = 0.7

CARD IMAGES OF CONTROL DATA

```
CARD                        IMAGE

  1)  LASER TURRET OPTIMIZATION AT M = 0.7
  2)  $ BLOCK B - CONTROL PARAMETERS
  3)          2         3
  4)  $ BLOCK C - CONMIN INTEGER PARAMETERS
  5)          5
  6)  $ BLOCK D - CONMIN REAL PARAMETERS, USE ALL DEFAULTS.
  7)  0.
  8)  0.
  9)  $ BLOCK E - MINIMIZE PHASE DISTORTION
 10)          3        169  -1.0
 11)  $ BLOCK F - DESIGN VARIABLE LIMITS
 12)  $ COEFFICIENT A - 5
 13)  -3.0       3.0
 14)  $ COEFFICIENT A - 6
 15)  -3.0       3.0
 16)  $ COEFFICIENT B - 6
 17)  -3.0       3.0
 18)  $ BLOCK G - DESIGN VARIABLE IDENTIFICATION
 19)  $ COEFFICIENT A - 5
 20)          1          6 1.0
 21)  $ COEFFICIENT A - 6
 22)          2          7 1.0
 23)  $      COEFFICIENT B - 6
 24)          3         60 1.0
 25)  $ BLOCK H - CONSTRAINTS
 26)          1
 27)  $ BLOCK I - CONSTRAINT ON SLOPE
 28)        139       168          1
 29)  $ LIMITED TO SMALL PERTURBATION THEORY
 30)  -0.3       0.3       0.3       0.3
 31)  $ BLOCK P - END OF COPES DATA
 32)  END
```

CARD IMAGES OF CONTROL DATA

```
CARD                          IMAGE

  1)   LASER TURRET OPTIMIZATION AT M = 0.7
  2)   $ BLOCK B - CONTROL PARAMETERS
  3)             2          3
  4)   $ BLOCK C - CONMIN INTEGER PARAMETERS
  5)             5
  6)   $ BLOCK D - CONMIN REAL PARAMETERS, USE ALL DEFAULTS.
  7)   0.
  8)   0.
  9)   $ BLOCK E - MINIMIZE PHASE DISTORTION
 10)             3        169  -1.0
 11)   $ BLOCK F - DESIGN VARIABLE LIMITS
 12)   $ COEFFICIENT A - 5
 13)   -3.0      3.0
 14)   $ COEFFICIENT Ā - 6
 15)   -3.0      3.0
 16)   $ COEFFICIENT B - 6
 17)   -3.0      3.0
 18)   $ BLOCK G - DESIGN VARIABLE IDENTIFICATION
 19)   $ COEFFICIENT A - 5
 20)             1          6 1.0
 21)   $ COEFFICIENT Ā - 6
 22)             2          7 1.0
 23)   $      COEFFICIENT B - 6
 24)             3         60 1.0
 25)   $ BLOCK H - CONSTRAINTS
 26)             1
 27)   $ BLOCK I - CONSTRAINT ON SLOPE
 28)           139        168         1
 29)   $ LIMITED TO SMALL PERTURBATION THEORY
 30)   -0.3      0.3        0.3        0.3
 31)   $ BLOCK P - END OF COPES DATA
 32)   END
```

TITLE:
LASER TURRET OPTIMIZATION AT M = 0.7

CONTROL PARAMETERS:

CALCULATION CONTROL,	NCALC	=	2
NUMBER OF GLOBAL DESIGN VARIABLES,	NDV	=	3
NUMBER OF SENSITIVITY VARIABLES,	NSV	=	-0
NUMBER OF FUNCTIONS IN TWO-SPACE,	N2VAR	=	-0
INPUT INFORMATION PRINT CODE,	IPNPUT	=	-0
SENSITIVITY PRINT CODE,	IPSENS	=	-0
TWO-SPACE PRINT CODE,	IP2VAR	=	-0
DEBUG PRINT CODE,	IPDBG	=	-0

CALCULATION CONTROL, NCALC

VALUE	MEANING
1	SINGLE ANALYSIS
2	OPTIMIZATION
3	SENSITIVITY
4	TWO-VARIABLE FUNCTION SPACE

GLOBAL VARIABLE NUMBER OF OBJECTIVE = 169
MULTIPLIER (NEGATIVE INDICATES MINIMIZATION) = -.1000E+01

CONMIN PARAMETERS (IF ZERO, CONMIN DEFAULT WILL OVER-RIDE)

IPRINT	ITMAX	ICNDIR	NSCAL	ITRM	LINOBJ	NACMX1	NFDG
5	-0	-0	-0	-0	-0	5	-0

FDCH	FDCHM	CT	CTMIN
0.	-0.	-0.	-0.

CTL	CTLMIN	THETA	PHI
-0.	-0.	-0.	-0.

DELFUN	DABFUN
0.	-0.

DESIGN VARIABLE INFORMATION
NON-ZERO INITIAL VALUE WILL OVER-RIDE MODULE INPUT

D. V. NO.	LOWER BOUND	UPPER BOUND	INITIAL VALUE	SCALE
1	-.30000E+01	.30000E+01	-0.	-0.
2	-.30000E+01	.30000E+01	-0.	-0.
3	-.30000E+01	.30000E+01	-0.	-0.

DESIGN VARIABLES

ID	D. V. NO.	GLOBAL VAR. NO.	MULTIPLYING FACTOR
1	1	6	.10000E+01
2	2	7	.10000E+01
3	3	60	.10000E+01

CONSTRAINT INFORMATION

THERE ARE 1 CONSTRAINT SETS

ID	GLOBAL VAR. 1	GLOBAL VAR. 2	LINEAR ID	LOWER BOUND	NORMALIZATION FACTOR	UPPER BOUND	NORMALIZATION FACTOR
1	139	168	1	-.30000E+00	.30000E+00	.30000E+00	.30000E+00

TOTAL NUMBER OF CONSTRAINED PARAMETERS = 30

DATA STORAGE REQUIREMENTS

REAL			INTEGER		
INPUT	EXECUTION	AVAILABLE	INPUT	EXECUTION	AVAILABLE
144	407	5000	103	118	1000

```
TURRET ANALYSIS INPUT

TITLE
SUBSONIC LASER TURRET AT SEA LEVEL


AERO-OPTICS
MACH NUMBER, AMACH                      =  .700
EXTERNAL DENSITY RATION, DENRTO         = 1.000
INTERNAL DENSITY RATIO, TDENRT          = 1.000
PRESSURE-DENSITY EXPONENT, DENGAM       = 1.405
PHASE DISTORTION CONSTANT, AKPRIM       =  .2300E-03
WAVELENGTH, WAVEL                       =  .3400E-05


GEOMETRY
FUSELAGE RADIUS, RFUS       =  2.500
TURRET HALF-LENGTH,         =  2.000
TURRET HALF-ANGLE, THMAX    = 60.000  DEGREES
TURRET HEIGHT FACTOR, EPS =     .300
TURRET HALF-SPACING, ACL    = 10.000


TURRET POLYNOMIAL SHAPE COEFICIENTS
X-DIRECTION,      ORDER =     6
COEFICIENTS
  .10000E+01  0.              -.61111E+00  0.              -.18056E+00
 0.              -.69440E-02

BOUNDARY CONDITIONS
 X/L        Y     Y-PRIME
0.000      .300  200.000
-1.000   -0.000   -0.000
1.000    -0.000   -0.000

THETA-DIRECTION, ORDER =     6
COEFICIENTS
  .10000E+01  0.              -.61111E+00  0.              -.18056E+00
 0.              -.69440E-02

BOUNDARY CONDITIONS
THETA/THMAX     Y     Y-PRIME
    0.000      .300  200.000
    1.000    -0.000   -0.000


LOCATION OF CENTER OF MIRROR
XM    = -0.000      EPSM =  1.150


PHASE DISTORTION CALCULATION POINTS
ANGLES
      0.000     45.000     90.000    135.000    180.000
    225.000    270.000    315.000

RADII
       .050       .100


BEAM ORIENTATIONS
```

BEAM	PHI	GAMMA	MACH	WEIGHT
1	0.00	50.00	.700	1.000
2	45.00	30.00	.700	1.000
3	90.00	10.00	.700	1.000

PHASE DISTORTION CALCULATIONS

BEAM ORIENTATION NUMBER = 1
AZMUTH ANGLE = 0.00 DEGREES
ELEVATION ANGLE = 50.00 DEGREES
MACH NUMBER = .70

R	ETA	X	Y	A	N
0.	0.00	0.	0.	.1920E+00	0.
.5000E-01	0.00	0.	.5000E-01	.1523E+00	.7546E+00
.5000E-01	45.00	.3536E-01	.3536E-01	.1629E+00	.5477E+00
.5000E-01	90.00	.5000E-01	.6675E-09	.1898E+00	.1850E-01
.5000E-01	135.00	.3536E-01	-.3536E-01	.2174E+00	-.5351E+00
.5000E-01	180.00	.1335E-08	-.5000E-01	.2291E+00	-.7745E+00
.5000E-01	225.00	-.3536E-01	-.3536E-01	.2174E+00	-.5351E+00
.5000E-01	270.00	-.5000E-01	-.2002E-08	.1898E+00	.1850E-01
.5000E-01	315.00	-.3536E-01	.3536E-01	.1629E+00	.5477E+00
.1000E+00	0.00	0.	.1000E+00	.1109E+00	.1461E+01
.1000E+00	45.00	.7071E-01	.7071E-01	.1307E+00	.1089E+01
.1000E+00	90.00	.1000E+00	.1335E-08	.1829E+00	.7426E-01
.1000E+00	135.00	.7071E-01	-.7071E-01	.2396E+00	-.1070E+01
.1000E+00	180.00	.2670E-08	-.1000E+00	.2645E+00	-.1581E+01
.1000E+00	225.00	-.7071E-01	-.7071E-01	.2396E+00	-.1070E+01
.1000E+00	270.00	-.1000E+00	-.4005E-08	.1829E+00	.7426E-01
.1000E+00	315.00	-.7071E-01	.7071E-01	.1307E+00	.1089E+01

ZERNICKE COEFFICIENTS/
AVERAGE = .10883E-02
TILT, X = .12854E+00 Y = -.10536E-02
FOCUS = .30067E-03
ASTIG = -.20378E-02 .12277E-04
COMA = -.13262E-03 -.21463E-06 -.48326E-03 .25331E-02

PHASE DISTORTION CALCULATIONS

BEAM ORIENTATION NUMBER = 2
AZMUTH ANGLE = 45.00 DEGREES
ELEVATION ANGLE = 30.00 DEGREES
MACH NUMBER = .70

R	ETA	X	Y	A	N
0.	0.00	0.	0.	.2532E+00	0.
.5000E-01	0.00	0.	.5000E-01	.1943E+00	.7872E+00
.5000E-01	45.00	.3536E-01	.3536E-01	.2166E+00	.2368E+00
.5000E-01	90.00	.5000E-01	.6675E-09	.2601E+00	-.4696E+00
.5000E-01	135.00	.3536E-01	-.3536E-01	.2977E+00	-.8906E+00
.5000E-01	180.00	.1335E-08	-.5000E-01	.3062E+00	-.7552E+00
.5000E-01	225.00	-.3536E-01	-.3536E-01	.2833E+00	-.1882E+00
.5000E-01	270.00	-.5000E-01	-.2002E-08	.2434E+00	.4563E+00
.5000E-01	315.00	-.3536E-01	.3536E-01	.2073E+00	.8422E+00
.1000E+00	0.00	0.	.1000E+00	.1181E+00	.1800E+01

.1000E+00	45.00	.7071E-01	.7071E-01	-.1668E+00	.6386E+00
.1000E+00	90.00	.1000E+00	.1335E-08	.2640E+00	-.9474E+00
.1000E+00	135.00	.7071E-01	-.7071E-01	.3411E+00	-.1821E+01
.1000E+00	180.00	.2670E-08	-.1000E+00	.3547E+00	-.1486E+01
.1000E+00	225.00	-.7071E-01	-.7071E-01	.3077E+00	-.3232E+00
.1000E+00	270.00	-.1000E+00	-.4005E-08	.2309E+00	.8960E+00
.1000E+00	315.00	-.7071E-01	.7071E-01	.1530E+00	.1759E+01

ZERNICKE COEFFICIENTS/
AVERAGE = .49776E-02
TILT, X = .13398E+00 Y = -.78917E-01
FOCUS = .37912E-02
ASTIG = .26635E-02 .58705E-02
COMA = .85058E-03 .77995E-04 .32978E-02 .27808E-02

PHASE DISTORTION CALCULATIONS

BEAM ORIENTATION NUMBER = 3
AZMUTH ANGLE = 90.00 DEGREES
ELEVATION ANGLE = 10.00 DEGREES
MACH NUMBER = .70

R	ETA	X	Y	A	N
0.	0.00	0.	0.	.3440E+00	0.
.5000E-01	0.00	0.	.5000E-01	.2708E+00	.7185E+00
.5000E-01	45.00	.3536E-01	.3536E-01	.2928E+00	.4686E+00
.5000E-01	90.00	.5000E-01	.6675E-09	.3434E+00	-.1814E-01
.5000E-01	135.00	.3536E-01	-.3536E-01	.3901E+00	-.2612E+00
.5000E-01	180.00	.1335E-08	-.5000E-01	.4066E+00	-.4199E+00
.5000E-01	225.00	-.3536E-01	-.3536E-01	.3901E+00	-.2612E+00
.5000E-01	270.00	-.5000E-01	-.2002E-08	.3434E+00	-.1814E-01
.5000E-01	315.00	-.3536E-01	.3536E-01	.2928E+00	.4686E+00
.1000E+00	0.00	0.	.1000E+00	.1783E+00	.1692E+01
.1000E+00	45.00	.7071E-01	.7071E-01	.2361E+00	.9818E+00
.1000E+00	90.00	.1000E+00	.1335E-08	.3413E+00	-.9792E-01
.1000E+00	135.00	.7071E-01	-.7071E-01	.4249E+00	-.6617E+00
.1000E+00	180.00	.2670E-08	-.1000E+00	.4514E+00	-.8736E+00
.1000E+00	225.00	-.7071E-01	-.7071E-01	.4249E+00	-.6617E+00
.1000E+00	270.00	-.1000E+00	-.4005E-08	.3413E+00	-.9792E-01
.1000E+00	315.00	-.7071E-01	.7071E-01	.2361E+00	.9818E+00

ZERNICKE COEFFICIENTS/
AVERAGE = .19155E-01
TILT, X = .99613E-01 Y = -.74724E-03
FOCUS = .62398E-02
ASTIG = .81849E-02 -.72295E-04
COMA = .30317E-02 .10055E-04 .29599E-02 .17965E-02

FLOW FIELD FOR THETA = 0.000 DEGREES

MACH NUMBER = .700

X	R	PHI	U	V	CP
-.4000E+01	.1000E+01	-.4757E-02	-.7515E-02	.1878E-02	.1503E-01
-.3600E+01	.1000E+01	-.7151E-02	-.3386E-02	.6275E-02	.6733E-02
-.3200E+01	.1000E+01	-.7146E-02	.2194E-02	.7948E-03	-.4389E-02
-.2800E+01	.1000E+01	-.7275E-02	-.6934E-02	-.1104E-01	.1375E-01
-.2400E+01	.1000E+01	-.1601E-01	-.4028E-01	-.6405E-02	.8052E-01
-.2000E+01	.1000E+01	-.4092E-01	-.8283E-01	.3960E-01	.1641E+00
-.1600E+01	.1028E+01	-.7512E-01	-.9061E-01	.1169E+00	.1675E+00
-.1200E+01	.1103E+01	-.9141E-01	-.3756E-01	.1593E+00	.4976E-01
-.8000E+00	.1197E+01	-.7710E-01	.3703E-01	.1370E+00	-.9283E-01
-.4000E+00	.1272E+01	-.4235E-01	.9030E-01	.7510E-01	-.1862E+00
.4974E-13	.1300E+01	.5391E-14	.1084E+00	-.9533E-14	-.2168E+00
.4000E+00	.1272E+01	.4235E-01	.9030E-01	-.7510E-01	-.1862E+00
.8000E+00	.1197E+01	.7710E-01	.3703E-01	-.1370E+00	-.9283E-01
.1200E+01	.1103E+01	.9141E-01	-.3756E-01	-.1593E+00	.4976E-01
.1600E+01	.1028E+01	.7512E-01	-.9061E-01	-.1169E+00	.1675E+00
.2000E+01	.1000E+01	.4092E-01	-.8283E-01	-.3960E-01	.1641E+00
.2400E+01	.1000E+01	.1601E-01	-.4028E-01	.6405E-02	.8052E-01
.2800E+01	.1000E+01	.7275E-02	-.6934E-02	.1104E-01	.1375E-01
.3200E+01	.1000E+01	.7146E-02	.2194E-02	-.7948E-03	-.4389E-02
.3600E+01	.1000E+01	.7151E-02	-.3386E-02	-.6275E-02	.6733E-02
.4000E+01	.1000E+01	.4757E-02	-.7515E-02	-.1878E-02	.1503E-01

CRITICAL PRESSURE COFFFICIENT ON SURFACE = 41.76395

SURFACE DEFINITION (EPS = .300)
POLYNOMIAL COEFICIENTS (A(I), I=0,MAXK) IN X-DIRECTION
.10000E+01 0. -.61110E+00 0. .11805E+00
0. -.69440E-02

POLYNOMIAL COEFICIENTS (B(I), I=0,MAXP) IN THETA-DIRECTION
.10000E+01 0. -.18321E+01 0. .84677E+00
0. -.69440E-02

COORDINATES

X	Z	Z-PRIME
-2.200	0.0000	0.0000
-2.000	.0000	0.0000
-1.800	.0069	.0700
-1.600	.0278	.1375
-1.400	.0610	.1918
-1.200	.1032	.2263
-1.000	.1500	.2375
-.800	.1966	.2249
-.600	.2385	.1904
-.400	.2716	.1377
-.200	.2927	.0722
.000	.3000	-.0000
.200	.2927	-.0722
.400	.2716	-.1377
.600	.2385	-.1904
.800	.1966	-.2249
1.000	.1500	-.2375
1.200	.1032	-.2263
1.400	.0610	-.1918
1.600	.0278	-.1375
1.800	.0069	-.0700
2.000	.0000	0.0000

2.200	0.0000	0.0000	

THETA RADIANS	DEGREES	Z	Z-PRIME
-1.152	-66.0000	0.0000	0.0000
-1.047	-60.0000	.0000	-.0000
-.942	-54.0000	.0107	.1947
-.838	-48.0000	.0387	.3286
-.733	-42.0000	.0777	.4082
-.628	-36.0000	.1225	.4399
-.524	-30.0000	.1684	.4302
-.419	-24.0000	.2114	.3859
-.314	-18.0000	.2482	.3139
-.209	-12.0000	.2764	.2209
-.105	-6.0000	.2940	.1139
.000	.0000	.3000	-.0000
.105	6.0000	.2940	-.1139
.209	12.0000	.2764	-.2209
.314	18.0000	.2482	-.3139
.419	24.0000	.2114	-.3859
.524	30.0000	.1684	-.4302
.628	36.0000	.1225	-.4399
.733	42.0000	.0777	-.4082
.838	48.0000	.0387	-.3286
.942	54.0000	.0107	-.1947
1.047	60.0000	.0000	.0000
1.152	66.0000	0.0000	0.0000

SUM OF SQUARES OF PHASE DISTORTION = .36648E+02

```
* * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                   *
*                    C O N M I N                    *
*                                                   *
*                FORTRAN PROGRAM FOR                *
*                                                   *
*         CONSTRAINED FUNCTION MINIMIZATION         *
*                                                   *
* NASA/AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.   *
*                                                   *
*            VERSION II      JULY, 1975             *
*                                                   *
* * * * * * * * * * * * * * * * * * * * * * * * * * *


CONSTRAINED FUNCTION MINIMIZATION

CONTROL PARAMETERS

IPRINT   NDV     ITMAX     NCON     NSIDE   ICNDIR    NSCAL    NFDG
  5       3       20        60        1        4        -0       -0

LINOBJ   ITRM     N1        N2       N3       N4       N5
 -0       3       5         66       5        5        10

    CT              CTMIN             CTL              CTLMIN
 -.10000E+00      .40000E-02      -.10000E-01        .10000E-02

   THETA             PHI             DELFUN           DABFUN
 .10000E+01       .50000E+01       .10000E-03        .36648E-01

   FDCH             FDCHM
 .10000E-01       .10000E-01

LOWER BOUNDS ON DECISION VARIABLES (VLB)
  1)    -.30000E+01  -.30000E+01  -.30000E+01

UPPER BOUNDS ON DECISION VARIABLES (VUB)
  1)     .30000E+01   .30000E+01   .30000E+01

ALL CONSTRAINTS ARE LINEAR


INITIAL FUNCTION INFORMATION

OBJ =     .366482E+02

DECISION VARIABLES (X-VECTOR)
  1)    0.           -.69440E-02  -.69440E-02

CONSTRAINT VALUES (G-VECTOR)
  1)    -.10000E+01  -.10000E+01  -.11598E+01  -.84021E+00  -.13218E+01  -.67816E+00
  7)    -.14725E+01  -.52753E+00  -.16011E+01  -.39889E+00  -.17001E+01  -.29994E+00
 13)    -.17642E+01  -.23576E+00  -.17910E+01  -.20905E+00  -.17796E+01  -.22039E+00
```

```
 19)    -.17315E+01  -.26850E+00  -.16495E+01  -.35047E+00  -.15380E+01  -.46200E+00
 25)    -.14023E+01  -.59771E+00  -.12487E+01  -.75130E+00  -.10841E+01  -.91586E+00
 31)    -.91586E+00  -.10841E+01  -.75130E+00  -.12487E+01  -.59771E+00  -.14023E+01
 37)    -.46200E+00  -.15380E+01  -.35047E+00  -.16495E+01  -.26850E+00  -.17315E+01
 43)    -.22039E+00  -.17796E+01  -.20905E+00  -.17910E+01  -.23576E+00  -.17642E+01
 49)    -.29994E+00  -.17001E+01  -.39889E+00  -.16011E+01  -.52753E+00  -.14725E+01
 55)    -.67816E+00  -.13218E+01  -.84021E+00  -.11598E+01  -.10000E+01  -.10000E+01


  BEGIN ITERATION NUMBER    1

  CT =   -.10000E+00      CTL =   -.10000E-01      PHI =     .50000E+01

  THERE ARE    0 ACTIVE CONSTRAINTS

  THERE ARE    0 VIOLATED CONSTRAINTS

  THERE ARE    0 ACTIVE SIDE CONSTRAINTS

  GRADIENT OF OBJ
    1)     .53485E+03  -.70436E+03   .19787E+02

  SEARCH DIRECTION (S-VECTOR)
    1)    -.75935E+00   .10000E+01  -.28092E-01

  ONE-DIMENSIONAL SEARCH
  INITIAL SLOPE =  -.1111E+04  PROPOSED ALPHA =   .3298E-02


* * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATION * * *


  PROPOSED DESIGN
  ALPHA =  .32985E-02
  X-VECTOR
-.2505E-02  -.3646E-02  -.7037E-02

  OBJ =    .33029E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1219E+01  -.7805E+00  -.1407E+01  -.5927E+00  -.1559E+01  -.440
-.1674E+01  -.3263E+00  -.1749E+01  -.2510E+00  -.1786E+01  -.2144E+00  -.1785E+01  -.214
-.1750E+01  -.2503E+00  -.1683E+01  -.3171E+00  -.1589E+01  -.4113E+00  -.1472E+01  -.528
-.1337E+01  -.6630E+00  -.1190E+01  -.8097E+00  -.1037E+01  -.9629E+00  -.8833E+00  -.111
-.7347E+00  -.1265E+01  -.5968E+00  -.1403E+01  -.4749E+00  -.1525E+01  -.3738E+00  -.162
-.2978E+00  -.1702E+01  -.2506E+00  -.1749E+01  -.2347E+00  -.1765E+01  -.2521E+00  -.174
-.3034E+00  -.1697E+01  -.3881E+00  -.1612E+01  -.5043E+00  -.1496E+01  -.6486E+00  -.135
-.8161E+00  -.1184E+01  -.1000E+01  -.1000E+01

  TWO-POINT INTERPOLATION


  PROPOSED DESIGN
  ALPHA =  .16492E-01
```

```
  X-VECTOR
-.1252E-01   .9548E-02  -.7407E-02

  OBJ =    .20655E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1458E+01  -.5418E+00  -.1749E+01  -.2511E+00  -.1907E+01  -.9258E-01
-.1964E+01  -.3596E-01  -.1945E+01  -.5527E-01  -.1871E+01  -.1287E+00  -.1762E+01  -.2384E+00
-.1630E+01  -.3698E+00  -.1489E+01  -.5115E+00  -.1345E+01  -.6548E+00  -.1206E+01  -.7937E+00
-.1076E+01  -.9241E+00  -.9565E+00  -.1043E+01  -.8490E+00  -.1151E+01  -.7531E+00  -.1247E+01
-.6682E+00  -.1332E+01  -.5930E+00  -.1407E+01  -.5263E+00  -.1474E+01  -.4671E+00  -.1533E+01
-.4152E+00  -.1585E+01  -.3713E+00  -.1629E+01  -.3375E+00  -.1662E+01  -.3177E+00  -.1682E+01
-.3175E+00  -.1683E+01  -.3451E+00  -.1655E+01  -.4115E+00  -.1588E+01  -.5306E+00  -.1469E+01
-.7197E+00  -.1280E+01  -.1000E+01  -.1000E+01

  THREE-POINT INTERPOLATION


  PROPOSED DESIGN
  ALPHA =   .18127E-01
  X-VECTOR
-.1376E-01   .1118E-01  -.7453E-02

  OBJ =    .19478E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1488E+01  -.5122E+00  -.1791E+01  -.2087E+00  -.1951E+01  -.4948E-01
-.2000E+01   .4145E-13  -.1969E+01  -.3103E-01  -.1882E+01  -.1181E+00  -.1759E+01  -.2413E+00
-.1615E+01  -.3846E+00  -.1464E+01  -.5355E+00  -.1315E+01  -.6850E+00  -.1173E+01  -.8266E+00
-.1044E+01  -.9564E+00  -.9276E+00  -.1072E+01  -.8257E+00  -.1174E+01  -.7370E+00  -.1263E+01
-.6600E+00  -.1340E+01  -.5925E+00  -.1407E+01  -.5326E+00  -.1467E+01  -.4786E+00  -.1521E+01
-.4297E+00  -.1570E+01  -.3863E+00  -.1614E+01  -.3503E+00  -.1650E+01  -.3258E+00  -.1674E+01
-.3192E+00  -.1681F+01  -.3398E+00  -.1660E+01  -.4000E+00  -.1600E+01  -.5160E+00  -.1484E+01
-.7078E+00  -.1292E+01  -.1000E+01  -.1000E+01

  * * * END OF ONE-DIMENSIONAL SEARCH

  CALCULATED ALPHA =     .18127E-01

  OBJ =     .194784E+02

  DECISION VARIABLES (X-VECTOR)
    1)     -.13765E-01   .11183E-01  -.74532E-02

  CONSTRAINT VALUES (G-VECTOR)
    1)     -.10000E+01  -.10000E+01  -.14878E+01  -.51220E+00  -.17913E+01  -.20873E+00
    7)     -.19505E+01  -.49484E-01  -.20000E+01   .41448E-13  -.19690E+01  -.31029E-01
   13)     -.18819E+01  -.11814E+00  -.17587E+01  -.24133E+00  -.16154E+01  -.38459E+00
   19)     -.14645E+01  -.53553E+00  -.13150E+01  -.68499E+00  -.11734E+01  -.82660E+00
   25)     -.10436E+01  -.95640E+00  -.92757E+00  -.10724E+01  -.82566E+00  -.11743E+01
   31)     -.73701E+00  -.12630E+01  -.65999E+00  -.13400E+01  -.59255E+00  -.14075E+01
   37)     -.53264E+00  -.14674E+01  -.47864E+00  -.15214E+01  -.42971E+00  -.15703E+01
   43)     -.38626E+00  -.16137E+01  -.35028E+00  -.16497E+01  -.32578E+00  -.16742E+01
   49)     -.31919E+00  -.16808E+01  -.33978E+00  -.16602E+01  -.40000E+00  -.16000E+01
   55)     -.51597E+00  -.14840E+01  -.70778E+00  -.12922E+01  -.10000E+01  -.10000E+01


  BEGIN ITERATION NUMBER    2
```

```
CT =   -.10000E+00     CTL =   -.10000E-01     PHI =    .50000E+01

THERE ARE    1 ACTIVE CONSTRAINTS
CONSTRAINT NUMBERS ARE
   10

THERE ARE    0 VIOLATED CONSTRAINTS

THERE ARE    0 ACTIVE SIDE CONSTRAINTS

GRADIENT OF OBJ
  1)     .12385E+03  -.58717E+03   .80411E+01

GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS
CONSTRAINT NUMBER   10
  1)    -.12342E+02   .12633E+02  0.


PUSH-OFF FACTORS, (THETA(I), I=1,NAC)
  1)    0.

CONSTRAINT PARAMETER, BETA =     .74996E+00

SEARCH DIRECTION (S-VECTOR)
  1)     .10000E+01   .97698E+00  -.34940E-01

ONE-DIMENSIONAL SEARCH
INITIAL SLOPE =  -.4501E+03  PROPOSED ALPHA =   .8655E-02



* * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATION * * *


  PROPOSED DESIGN
  ALPHA =  .86554E-02
  X-VECTOR
-.5109E-02   .1964E-01  -.7756E-02

  OBJ =   .15172E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1534E+01  -.4663E+00  -.1842E+01  -.1580E+00  -.1982E+01  -.184
-.2000E+01   .5329E-13  -.1937E+01  -.6340E-01  -.1823E+01  -.1770E+00  -.1684E+01  -.316
-.1538E+01  -.4621E+00  -.1398E+01  -.6022E+00  -.1272E+01  -.7280E+00  -.1164E+01  -.835
-.1076E+01  -.9245E+00  -.1004E+01  -.9964E+00  -.9446E+00  -.1055E+01  -.8931E+00  -.110
-.8433E+00  -.1157E+01  -.7894E+00  -.1211E+01  -.7266E+00  -.1273E+01  -.6514E+00  -.134
-.5629E+00  -.1437E+01  -.4627E+00  -.1537E+01  -.3565E+00  -.1644E+01  -.2541E+00  -.174
-.1704E+00  -.1830E+01  -.1261E+00  -.1874E+01  -.1485E+00  -.1851E+01  -.2720E+00  -.172
-.5389E+00  -.1461E+01  -.1000E+01  -.1000E+01

  TWO-POINT INTERPOLATION


  PROPOSED DESIGN
  ALPHA =  .13765E-01
  X-VECTOR
```

```
-.3525E-14   .2463E-01  -.7934E-02

  OBJ =   .13638E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1561E+01  -.4392E+00  -.1872E+01  -.1281E+00  -.2000E+01  -.1179E-0
-.2000E+01   .7698E-13  -.1917E+01  -.8250E-01  -.1788E+01  -.2117E+00  -.1640E+01  -.3602E+0
-.1492E+01  -.5079E+00  -.1359E+01  -.6415E+00  -.1247E+01  -.7534E+00  -.1159E+01  -.8411E+0
-.1094E+01  -.9056E+00  -.1049E+01  -.9515E+00  -.1015E+01  -.9852E+00  -.9852E+00  -.1015E+0
-.9515E+00  -.1049E+01  -.9056E+00  -.1094E+01  -.8411E+00  -.1159E+01  -.7534E+00  -.1247E+0
-.6415E+00  -.1359E+01  -.5079E+00  -.1492E+01  -.3602E+00  -.1640E+01  -.2117E+00  -.1788E+0
-.8250E-01  -.1917E+01  0.          -.2000E+01  -.1179E-03  -.2000E+01  -.1281E+00  -.1872E+0
-.4392E+00  -.1561E+01  -.1000E+01  -.1000E+01

  * * * END OF ONE-DIMENSIONAL SEARCH

  CALCULATED ALPHA =    .13765E-01

  OBJ =    .136384E+02

  DECISION VARIABLES (X-VECTOR)
    1)    -.35250E-14   .24631E-01  -.79342E-02

  CONSTRAINT VALUES (G-VECTOR)
    1)    -.10000E+01  -.10000E+01  -.15608E+01  -.43920E+00  -.18719E+01  -.12806E+00
    7)    -.19999E+01  -.11790E-03  -.20000E+01   .76975E-13  -.19175E+01  -.82503E-01
   13)    -.17883E+01  -.21172E+00  -.16398E+01  -.36016E+00  -.14921E+01  -.50786E+00
   19)    -.13585E+01  -.64148E+00  -.12466E+01  -.75345E+00  -.11589E+01  -.84107E+00
   25)    -.10944E+01  -.90561E+00  -.10485E+01  -.95146E+00  -.10148E+01  -.98522E+00
   31)    -.98522E+00  -.10148E+01  -.95146E+00  -.10485E+01  -.90561E+00  -.10944E+01
   37)    -.84107E+00  -.11589E+01  -.75345E+00  -.12466E+01  -.64148E+00  -.13585E+01
   43)    -.50786E+00  -.14921E+01  -.36016E+00  -.16398E+01  -.21172E+00  -.17883E+01
   49)    -.82503E-01  -.19175E+01  0.           -.20000E+01  -.11790E-03  -.19999E+01
   55)    -.12806E+00  -.18719E+01  -.43920E+00  -.15608E+01  -.10000E+01  -.10000E+01


  BEGIN ITERATION NUMBER    3

  CT =  -.10000E+00     CTL =   -.10000E-01     PHI =    .50000E+01

  THERE ARE    4 ACTIVE CONSTRAINTS
  CONSTRAINT NUMBERS ARE
      8   10   51   53

  THERE ARE    0 VIOLATED CONSTRAINTS

  THERE ARE    0 ACTIVE SIDE CONSTRAINTS

  GRADIENT OF OBJ
    1)     .39138E+03   .66056E+03   .67398E+01

  GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS
  CONSTRAINT NUMBER    8
    1)    -.12733E+02   .16704E+02  0.

  CONSTRAINT NUMBER   10
    1)    -.12342E+02   .12633E+02  0.
```

```
CONSTRAINT NUMBER    51
  1)      .12342E+02   .12633E+02  0.

CONSTRAINT NUMBER    53
  1)      .12733E+02   .16704E+02  0.


PUSH-OFF FACTORS, (THETA(I), I=1,NAC).
  1)    0.            0.            0.            0.

CONSTRAINT PARAMETER, BETA =     .11624E+01

SEARCH DIRECTION (S-VECTOR)
  1)    -.59251E+00  -.10000E+01  -.10203E-01

ONE-DIMENSIONAL SEARCH
INITIAL SLOPE =  -.8925E+03  PROPOSED ALPHA =    .3056E-02



* * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATION * * *


  PROPOSED DESIGN
  ALPHA =  .30562E-02
  X-VECTOR
-.1811E-02   .2157E-01  -.7965E-02

  OBJ =    .14934E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1535E+01  -.4652E+00  -.1839E+01  -.1611E+00  -.1972E+01  -.2811E
-.1984E+01  -.1626E-01  -.1915E+01  -.8459E-01  -.1800E+01  -.2004E+00  -.1662E+01  -.3384E
-.1520E+01  -.4799E+00  -.1388E+01  -.6123E+00  -.1272E+01  -.7280E+00  -.1176E+01  -.8237E
-.1100E+01  -.8997E+00  -.1041E+01  -.9592E+00  -.9927E+00  -.1007E+01  -.9497E+00  -.1050E
-.9050E+00  -.1095E+01  -.8519E+00  -.1148E+01  -.7850E+00  -.1215E+01  -.7009E+00  -.1299E
-.5984E+00  -.1402E+01  -.4801E+00  -.1520E+01  -.3527E+00  -.1647E+01  -.2277E+00  -.1772E
-.1225E+00  -.1877E+01  -.6096E-01  -.1939E+01  -.7422E-01  -.1926E+01  -.2015E+00  -.1798E
-.4909E+00  -.1509E+01  -.1000E+01  -.1000E+01

  TWO-POINT INTERPOLATION


  PROPOSED DESIGN
  ALPHA =  .10361E-02
  X-VECTOR
-.6139E-03   .2359E-01  -.7945E-02

  OBJ =    .14034E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1552E+01  -.4480E+00  -.1861E+01  -.1393E+00  -.1990E+01  -.9608E
-.1994E+01  -.5512E-02  -.1917E+01  -.8321E-01  -.1792E+01  -.2079E+00  -.1647E+01  -.3528E
-.1502E+01  -.4984E+00  -.1368E+01  -.6316E+00  -.1255E+01  -.7448E+00  -.1165E+01  -.8352E
-.1096E+01  -.9036E+00  -.1046E+01  -.9541E+00  -.1007E+01  -.9927E+00  -.9732E+00  -.1027E
-.9357E+00  -.1064E+01  -.8874E+00  -.1113E+01  -.8221E+00  -.1178E+01  -.7356E+00  -.1264E
-.6269E+00  -.1373E+01  -.4985E+00  -.1502E+01  -.3576E+00  -.1642E+01  -.2171E+00  -.1783E
-.9606E-01  -.1904E+01  -.2067E-01  -.1979E+01  -.2524E-01  -.1975E+01  -.1530E+00  -.1847E
```

```
-.4567E+00  -.1543E+01  -.1000E+01  -.1000E+01

 THREE-POINT INTERPOLATION


 PROPOSED DESIGN
 ALPHA =   .30615E-03
 X-VECTOR
-.1814E-03   .2432E-01  -.7937E-02

 OBJ =    .13750E+02

 CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1558E+01  -.4418E+00  -.1869E+01  -.1314E+00  -.1997E+01  -.2922E-02
-.1998E+01  -.1629E-02  -.1917E+01  -.8271E-01  -.1789E+01  -.2106E+00  -.1642E+01  -.3580E+00
-.1495E+01  -.5051E+00  -.1361E+01  -.6386E+00  -.1249E+01  -.7509E+00  -.1161E+01  -.8393E+00
-.1095E+01  -.9050E+00  -.1048E+01  -.9522E+00  -.1013E+01  -.9874E+00  -.9817E+00  -.1018E+01
-.9468E+00  -.1053E+01  -.9002E+00  -.1100E+01  -.8355E+00  -.1165E+01  -.7482E+00  -.1252E+01
-.6372E+00  -.1363E+01  -.5051E+00  -.1495E+01  -.3594E+00  -.1641E+01  -.2133E+00  -.1787E+01
-.8651E-01  -.1913E+01  -.6107E-02  -.1994E+01  -.7541E-02  -.1992E+01  -.1354E+00  -.1865E+01
-.4444E+00  -.1556E+01  -.1000E+01  -.1000E+01

 * * * END OF ONE-DIMENSIONAL SEARCH

 CALCULATED ALPHA =      .69389E-17

 OBJ =     .136384E+02       NO CHANGE ON OBJ

 DECISION VARIABLES (X-VECTOR)
   1)     -.35258E-14   .24631E-01  -.79342E-02

 CONSTRAINT VALUES (G-VECTOR)
   1)     -.10000E+01  -.10000E+01  -.15608E+01  -.43920E+00  -.18719E+01  -.12806E+00
   7)     -.19999E+01  -.11790E-03  -.20000E+01   .76975E-13  -.19175E+01  -.82503E-01
  13)     -.17883E+01  -.21172E+00  -.16398E+01  -.36016E+00  -.14921E+01  -.50786E+00
  19)     -.13585E+01  -.64148E+00  -.12466E+01  -.75345E+00  -.11589E+01  -.84107E+00
  25)     -.10944E+01  -.90561E+00  -.10485E+01  -.95146E+00  -.10148E+01  -.98522E+00
  31)     -.98522E+00  -.10148E+01  -.95146E+00  -.10485E+01  -.90561E+00  -.10944E+01
  37)     -.84107E+00  -.11589E+01  -.75345E+00  -.12466E+01  -.64148E+00  -.13585E+01
  43)     -.50786E+00  -.14921E+01  -.36016E+00  -.16398E+01  -.21172E+00  -.17883E+01
  49)     -.82503E-01  -.19175E+01  0.           -.20000E+01  -.11790E-03  -.19999E+01
  55)     -.12806E+00  -.18719E+01  -.43920E+00  -.15608E+01  -.10000E+01  -.10000E+01


 BEGIN ITERATION NUMBER    4

 CT =    -.34200E-01      CTL =   -.46416E-02      PHI =     .50000E+01

 THERE ARE     4 ACTIVE CONSTRAINTS
 CONSTRAINT NUMBERS ARE
     8    10    51    53

 THERE ARE     0 VIOLATED CONSTRAINTS

 THERE ARE     0 ACTIVE SIDE CONSTRAINTS

 GRADIENT OF OBJ
   1)     .39138E+03   .66056E+03   .67398E+01
```

```
GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS
CONSTRAINT NUMBER     8
  1)     -.60623E+00   .79529E+00  0.

CONSTRAINT NUMBER    10
  1)     -.69883E+00   .71529E+00  0.

CONSTRAINT NUMBER    51
  1)      .69883E+00   .71529E+00  0.

CONSTRAINT NUMBER    53
  1)      .60623E+00   .79529E+00  0.


PUSH-OFF FACTORS, (THETA(I), I=1,NAC)
  1)    0.              0.             0.             0.

CONSTRAINT PARAMETER, BETA =     .11624E+01

SEARCH DIRECTION (S-VECTOR)
  1)     -.59251E+00  -.10000E+01  -.10203E-01

ONE-DIMENSIONAL SEARCH
INITIAL SLOPE =  -.8925E+03   PROPOSED ALPHA =    .2291E-02



* * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATION * * *


  PROPOSED DESIGN
  ALPHA =   .22907E-02
  X-VECTOR
-.1357E-02    .2234E-01   -.7958E-02

  OBJ =    .14579E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1541E+01  -.4587E+00  -.1847E+01  -.1528E+00  -.1979E+01  -.2110E
-.1988E+01  -.1219E-01  -.1916E+01  -.8407E-01  -.1797E+01  -.2032E+00  -.1656E+01  -.3438E
-.1513E+01  -.4869E+00  -.1380E+01  -.6196E+00  -.1266E+01  -.7344E+00  -.1172E+01  -.8281E
-.1099E+01  -.9012E+00  -.1043E+01  -.9573E+00  -.9983E+00  -.1002E+01  -.9586E+00  -.1041E
-.9166E+00  -.1083E+01  -.8653E+00  -.1135E+01  -.7991E+00  -.1201E+01  -.7140E+00  -.1286E
-.6092E+00  -.1391E+01  -.4871E+00  -.1513E+01  -.3546E+00  -.1645E+01  -.2237E+00  -.1776E
-.1125E+00  -.1888E+01  -.4569E-01  -.1954E+01  -.5566E-01  -.1944E+01  -.1831E+00  -.1817E
-.4779E+00  -.1522E+01  -.1000E+01  -.1000E+01

  TWO-POINT INTERPOLATION


  PROPOSED DESIGN
  ALPHA =   .78455E-03
  X-VECTOR
-.4649E-03    .2385E-01   -.7942E-02

  OBJ =    .13977E+02
```

```
 CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1554E+01  -.4459E+00  -.1863E+01  -.1365E+00  -.1993E+01  -.7304E-02
-.1996E+01  -.4174E-02  -.1917E+01  -.8304E-01  -.1791E+01  -.2088E+00  -.1645E+01  -.3546E+00
-.1499E+01  -.5007E+00  -.1366E+01  -.6340E+00  -.1253E+01  -.7469E+00  -.1163E+01  -.8366E+00
-.1096E+01  -.9041E+00  -.1047E+01  -.9535E+00  -.1009E+01  -.9909E+00  -.9761E+00  -.1024E+01
-.9395E+00  -.1060E+01  -.8918E+00  -.1108E+01  -.8267E+00  -.1173E+01  -.7399E+00  -.1260E+01
-.6304E+00  -.1370E+01  -.5007E+00  -.1499E+01  -.3582E+00  -.1642E+01  -.2158E+00  -.1784E+01
-.9277E-01  -.1907E+01  -.1565E-01  -.1984E+01  -.1914E-01  -.1981E+01  -.1469E+00  -.1853E+01
-.4525E+00  -.1548E+01  -.1000E+01  -.1000E+01

 THREE-POINT INTERPOLATION


 PROPOSED DESIGN
 ALPHA =  .22044E-03
 X-VECTOR
-.1306E-03   .2441E-01  -.7936E-02

 OBJ =    .13716E+02

 CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1559E+01  -.4411E+00  -.1870E+01  -.1304E+00  -.1998E+01  -.2137E-02
-.1999E+01  -.1173E-02  -.1917E+01  -.8265E-01  -.1789E+01  -.2109E+00  -.1641E+01  -.3586E+00
-.1494E+01  -.5058E+00  -.1361E+01  -.6394E+00  -.1248E+01  -.7516E+00  -.1160E+01  -.8398E+00
-.1095E+01  -.9052E+00  -.1048E+01  -.9520E+00  -.1013E+01  -.9868E+00  -.9827E+00  -.1017E+01
-.9481E+00  -.1052E+01  -.9017E+00  -.1098E+01  -.8370E+00  -.1163E+01  -.7497E+00  -.1250E+01
-.6384E+00  -.1362E+01  -.5059E+00  -.1494E+01  -.3596E+00  -.1640E+01  -.2129E+00  -.1787E+01
-.8539E-01  -.1915E+01  -.4397E-02  -.1996E+01  -.5463E-02  -.1995E+01  -.1334E+00  -.1867E+01
-.4429E+00  -.1557E+01  -.1000E+01  -.1000E+01

 * * * END OF ONE-DIMENSIONAL SEARCH

 CALCULATED ALPHA =     .43368E-17

 OBJ =     .136384E+02      NO CHANGE ON OBJ

 DECISION VARIABLES (X-VECTOR)
   1)     -.35267E-14   .24631E-01  -.79342E-02

 CONSTRAINT VALUES (G-VECTOR)
   1)     -.10000E+01  -.10000E+01  -.15608E+01  -.43920E+00  -.18719E+01  -.12806E+00
   7)     -.19999E+01  -.11790E-03  -.20000E+01   .76975E-13  -.19175E+01  -.82503E-01
  13)     -.17883E+01  -.21172E+00  -.16398E+01  -.36016E+00  -.14921E+01  -.50786E+00
  19)     -.13585E+01  -.64148E+00  -.12466E+01  -.75345E+00  -.11589E+01  -.84107E+00
  25)     -.10944E+01  -.90561E+00  -.10485E+01  -.95146E+00  -.10148E+01  -.98522E+00
  31)     -.98522E+00  -.10148E+01  -.95146E+00  -.10485E+01  -.90561E+00  -.10944E+01
  37)     -.84107E+00  -.11589E+01  -.75345E+00  -.12466E+01  -.64148E+00  -.13585E+01
  43)     -.50786E+00  -.14921E+01  -.36016E+00  -.16398E+01  -.21172E+00  -.17883E+01
  49)     -.82503E-01  -.19175E+01  0.           -.20000E+01  -.11790E-03  -.19999E+01
  55)     -.12806E+00  -.18719E+01  -.43920E+00  -.15608E+01  -.10000E+01  -.10000E+01



 BEGIN ITERATION NUMBER     5

 CT =   -.11696E-01     CTL =   -.21544E-02      PHI =     .50000E+01

 THERE ARE    4 ACTIVE CONSTRAINTS
 CONSTRAINT NUMBERS ARE
```

```
     8   10   51   53

  THERE ARE     0 VIOLATED CONSTRAINTS

  THERE ARE     0 ACTIVE SIDE CONSTRAINTS

  GRADIENT OF OBJ
    1)      .39138E+03   .66056E+03   .67398E+01

  GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS
  CONSTRAINT NUMBER    8
    1)     -.12733E+02   .16704E+02  -.11842E-11

  CONSTRAINT NUMBER   10
    1)     -.12342E+02   .12633E+02  0.

  CONSTRAINT NUMBER   51
    1)      .12342E+02   .12633E+02  -.11842E-11

  CONSTRAINT NUMBER   53
    1)      .12733E+02   .16704E+02  -.29606E-11


  PUSH-OFF FACTORS, (THETA(I), I=1,NAC)
    1)    0.            0.            0.            0.

  CONSTRAINT PARAMETER, BETA =    .11624E+01

  SEARCH DIRECTION (S-VECTOR)
    1)     -.59251E+00  -.10000E+01  -.10203E-01

  ONE-DIMENSIONAL SEARCH
  INITIAL SLOPE =  -.8925E+03  PROPOSED ALPHA =   .1528E-05



* * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATION * * *


  PROPOSED DESIGN
  ALPHA =  .15281E-05
  X-VECTOR
-.9054E-06   .2463E-01  -.7934E-02

  OBJ =   .13639E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1561E+01  -.4392E+00  -.1872E+01  -.1281E+00  -.2000E+01  -.1319E
-.2000E+01  -.8130E-05  -.1917E+01  -.8250E-01  -.1788E+01  -.2117E+00  -.1640E+01  -.3602E
-.1492E+01  -.5078E+00  -.1359E+01  -.6415E+00  -.1247E+01  -.7534E+00  -.1159E+01  -.8411E
-.1094E+01  -.9056E+00  -.1049E+01  -.9515E+00  -.1015E+01  -.9852E+00  -.9852E+00  -.1015E
-.9514E+00  -.1049E+01  -.9056E+00  -.1094E+01  -.8410E+00  -.1159E+01  -.7534E+00  -.1247E
-.6415E+00  -.1359E+01  -.5078E+00  -.1492E+01  -.3602E+00  -.1640E+01  -.2117E+00  -.1788E
-.8252E-01  -.1917E+01  -.3048E-04  -.2000E+01  -.1550E-03  -.2000E+01  -.1281E+00  -.1872E
-.4392E+00  -.1561E+01  -.1000E+01  -.1000E+01

  TWO-POINT INTERPOLATION
```

```
  PROPOSED DESIGN
  ALPHA =  .53016E-06
  X-VECTOR
-.3141E-06   .2463E-01  -.7934E-02

  OBJ =    .13639E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1561E+01  -.4392E+00  -.1872E+01  -.1281E+00  -.2000E+01  -.1228E-03
-.2000E+01  -.2821E-05  -.1917E+01  -.8250E-01  -.1788E+01  -.2117E+00  -.1640E+01  -.3602E+00
-.1492E+01  -.5079E+00  -.1359E+01  -.6415E+00  -.1247E+01  -.7534E+00  -.1159E+01  -.8411E+00
-.1094E+01  -.9056E+00  -.1049E+01  -.9515E+00  -.1015E+01  -.9852E+00  -.9852E+00  -.1015E+01
-.9515E+00  -.1049E+01  -.9056E+00  -.1094E+01  -.8411E+00  -.1159E+01  -.7534E+00  -.1247E+01
-.6415E+00  -.1359E+01  -.5079E+00  -.1492E+01  -.3602E+00  -.1640E+01  -.2117E+00  -.1788E+01
-.8251E-01  -.1917E+01  -.1057E-04  -.2000E+01  -.1308E-03  -.2000E+01  -.1281E+00  -.1872E+01
-.4392E+00  -.1561E+01  -.1000E+01  -.1000E+01

  THREE-POINT INTERPOLATION


  PROPOSED DESIGN
  ALPHA =  .15380E-06
  X-VECTOR
-.9113E-07   .2463E-01  -.7934E-02

  OBJ =    .13638E+02

  CONSTRAINT VALUES
-.1000E+01  -.1000E+01  -.1561E+01  -.4392E+00  -.1872E+01  -.1281E+00  -.2000E+01  -.1193E-03
-.2000E+01  -.8183E-06  -.1917E+01  -.8250E-01  -.1788E+01  -.2117E+00  -.1640E+01  -.3602E+00
-.1492E+01  -.5079E+00  -.1359E+01  -.6415E+00  -.1247E+01  -.7534E+00  -.1159E+01  -.8411E+00
-.1094E+01  -.9056E+00  -.1049E+01  -.9515E+00  -.1015E+01  -.9852E+00  -.9852E+00  -.1015E+01
-.9515E+00  -.1049E+01  -.9056E+00  -.1094E+01  -.8411E+00  -.1159E+01  -.7534E+00  -.1247E+01
-.6415E+00  -.1359E+01  -.5079E+00  -.1492E+01  -.3602E+00  -.1640E+01  -.2117E+00  -.1788E+01
-.8250E-01  -.1917E+01  -.3068E-05  -.2000E+01  -.1216E-03  -.2000E+01  -.1281E+00  -.1872E+01
-.4392E+00  -.1561E+01  -.1000E+01  -.1000E+01

  * * * END OF ONE-DIMENSIONAL SEARCH

  CALCULATED ALPHA =      .25411E-20

  OBJ =      .136384E+02      NO CHANGE ON OBJ

  DECISION VARIABLES (X-VECTOR)
    1)      -.35267E-14    .24631E-01   -.79342E-02

  CONSTRAINT VALUES (G-VECTOR)
    1)      -.10000E+01   -.10000E+01   -.15608E+01   -.43920E+00   -.18719E+01   -.12806E+00
    7)      -.19999E+01   -.11790E-03   -.20000E+01    .76975E-13   -.19175E+01   -.82503E-01
   13)      -.17883E+01   -.21172E+00   -.16398E+01   -.36016E+00   -.14921E+01   -.50786E+00
   19)      -.13585E+01   -.64148E+00   -.12466E+01   -.75345E+00   -.11589E+01   -.84107E+00
   25)      -.10944E+01   -.90561E+00   -.10485E+01   -.95146E+00   -.10148E+01   -.98522E+00
   31)      -.98522E+00   -.10148E+01   -.95146E+00   -.10485E+01   -.90561E+00   -.10944E+01
   37)      -.84107E+00   -.11589E+01   -.75345E+00   -.12466E+01   -.64148E+00   -.13585E+01
   43)      -.50786E+00   -.14921E+01   -.36016E+00   -.16398E+01   -.21172E+00   -.17883E+01
   49)      -.82503E-01   -.19175E+01   0.            -.20000E+01   -.11790E-03   -.19999E+01
   55)      -.12806E+00   -.18719E+01   -.43920E+00   -.15608E+01   -.10000E+01   -.10000E+01
```

```
FINAL OPTIMIZATION INFORMATION

 OBJ =     .136384E+02

 DECISION VARIABLES (X-VECTOR)
   1)     -.35267E-14   .24631E-01  -.79342E-02

 CONSTRAINT VALUES (G-VECTOR)
   1)     -.10000E+01  -.10000E+01  -.15608E+01  -.43920E+00  -.18719E+01  -.12806E+00
   7)     -.19999E+01  -.11790E-03  -.20000E+01   .76975E-13  -.19175E+01  -.82503E-01
  13)     -.17883E+01  -.21172E+00  -.16398E+01  -.36016E+00  -.14921E+01  -.50786E+00
  19)     -.13585E+01  -.64148E+00  -.12466E+01  -.75345E+00  -.11589E+01  -.84107E+00
  25)     -.10944E+01  -.90561E+00  -.10485E+01  -.95146E+00  -.10148E+01  -.98522E+00
  31)     -.98522E+00  -.10148E+01  -.95146E+00  -.10485E+01  -.90561E+00  -.10944E+01
  37)     -.84107E+00  -.11589E+01  -.75345E+00  -.12466E+01  -.64148E+00  -.13585E+01
  43)     -.50786E+00  -.14921E+01  -.36016E+00  -.16398E+01  -.21172E+00  -.17883E+01
  49)     -.82503E-01  -.19175E+01  0.           -.20000E+01  -.11790E-03  -.19999E+01
  55)     -.12806E+00  -.18719E+01  -.43920E+00  -.15608E+01  -.10000E+01  -.10000E+01

 THERE ARE    4 ACTIVE CONSTRAINTS
 CONSTRAINT NUMBERS ARE
     8   10   51   53

 THERE ARE    0 VIOLATED CONSTRAINTS

 THERE ARE    0 ACTIVE SIDE CONSTRAINTS

 TERMINATION CRITERION
      ABS(1-OBJ(I-1)/OBJ(I)) LESS THAN DELFUN FOR  3 ITERATIONS
      ABS(OBJ(I)-OBJ(I-1))   LESS THAN DABFUN FOR  3 ITERATIONS

 NUMBER OF ITERATIONS =     5

 OBJECTIVE FUNCTION WAS EVALUATED          27  TIMES

 CONSTRAINT FUNCTIONS WERE EVALUATED       27  TIMES


 PHASE DISTORTION CALCULATIONS

 BEAM ORIENTATION NUMBER =      1
 AZMUTH ANGLE            =      0.00 DEGREES
 ELEVATION ANGLE         =     50.00 DEGREES
 MACH NUMBER             =       .70
      R            ETA        X             Y              A            N
 0.               0.00   0.            0.              .1951E+00   0.
  .5000E-01       0.00   0.             .5000E-01      .1536E+00    .4060E+00
  .5000E-01      45.00    .3536E-01     .3536E-01      .1646E+00    .2946E+00
  .5000E-01      90.00    .5000E-01     .6675E-09      .1928E+00    .1179E-01
  .5000E-01     135.00    .3536E-01   -.3536E-01      .2230E+00   -.2882E+00
  .5000E-01     180.00    .1335E-08   -.5000E-01      .2361E+00   -.4180E+00
  .5000E-01     225.00   -.3536E-01   -.3536E-01      .2230E+00   -.2882E+00
  .5000E-01     270.00   -.5000E-01   -.2002E-08      .1928E+00    .1179E-01
  .5000E-01     315.00   -.3536E-01    .3536E-01      .1646E+00    .2946E+00
  .1000E+00       0.00   0.             .1000E+00      .1117E+00    .8038E+00
```

.1000E+00	45.00	.7071E-01	.7071E-01	.1316E+00	.5981E+00
.1000E+00	90.00	.1000E+00	.1335E-08	.1858E+00	.4740E-01
.1000E+00	135.00	.7071E-01	-.7071E-01	.2483E+00	-.5606E+00
.1000E+00	180.00	.2670E-08	-.1000E+00	.2766E+00	-.8476E+00
.1000E+00	225.00	-.7071E-01	-.7071E-01	.2483E+00	-.5606E+00
.1000E+00	270.00	-.1000E+00	-.4005E-08	.1858E+00	.4740E-01
.1000E+00	315.00	-.7071E-01	.7071E-01	.1316E+00	.5981E+00

ZERNICKE COEFFICIENTS/
AVERAGE = .13222E-02
TILT, X = .69462E-01 Y = -.56737E-03
FOCUS = .74027E-03
ASTIG = -.10778E-02 .76826E-05
COMA = .42196E-04 -.20757E-07 -.16585E-03 .13640E-02

PHASE DISTORTION CALCULATIONS

BEAM ORIENTATION NUMBER = 2
AZMUTH ANGLE = 45.00 DEGREES
ELEVATION ANGLE = 30.00 DEGREES
MACH NUMBER = .70

R	ETA	X	Y	A	N
0.	0.00	0.	0.	.2591E+00	0.
.5000E-01	0.00	0.	.5000E-01	.1973E+00	.4980E+00
.5000E-01	45.00	.3536E-01	.3536E-01	.2219E+00	.2349E+00
.5000E-01	90.00	.5000E-01	.6675E-09	.2695E+00	-.1833E+00
.5000E-01	135.00	.3536E-01	-.3536E-01	.3090E+00	-.4811E+00
.5000E-01	180.00	.1335E-08	-.5000E-01	.3157E+00	-.4716E+00
.5000E-01	225.00	-.3536E-01	-.3536E-01	.2894E+00	-.1923E+00
.5000E-01	270.00	-.5000E-01	-.2002E-08	.2466E+00	.1797E+00
.5000E-01	315.00	-.3536E-01	.3536E-01	.2093E+00	.4600E+00
.1000E+00	0.00	0.	.1000E+00	.1195E+00	.1151E+01
.1000E+00	45.00	.7071E-01	.7071E-01	.1724E+00	.5715E+00
.1000E+00	90.00	.1000E+00	.1335E-08	.2777E+00	-.3700E+00
.1000E+00	135.00	.7071E-01	-.7071E-01	.3590E+00	-.9916E+00
.1000E+00	180.00	.2670E-08	-.1000E+00	.3676E+00	-.9181E+00
.1000E+00	225.00	-.7071E-01	-.7071E-01	.3134E+00	-.3451E+00
.1000E+00	270.00	-.1000E+00	-.4005E-08	.2323E+00	.3574E+00
.1000E+00	315.00	-.7071E-01	.7071E-01	.1538E+00	.9660E+00

ZERNICKE COEFFICIENTS/
AVERAGE = .43769E-02
TILT, X = .84736E-01 Y = -.31425E-01
FOCUS = .29497E-02
ASTIG = .17874E-02 .39176E-02
COMA = .38912E-03 .92258E-04 .17111E-02 .16210E-02

PHASE DISTORTION CALCULATIONS

BEAM ORIENTATION NUMBER = 3
AZMUTH ANGLE = 90.00 DEGREES
ELEVATION ANGLE = 10.00 DEGREES

MACH NUMBER = .70

R	ETA	X	Y	A	N
0.	0.00	0.	0.	.3440E+00	0.
.5000E-01	0.00	0.	.5000E-01	.2707E+00	.5017E+00
.5000E-01	45.00	.3536E-01	.3536E-01	.2931E+00	.3247E+00
.5000E-01	90.00	.5000E-01	.6675E-09	.3439E+00	-.1160E-02
.5000E-01	135.00	.3536E-01	-.3536E-01	.3903E+00	-.1608E+00
.5000E-01	180.00	.1335E-08	-.5000E-01	.4066E+00	-.2919E+00
.5000E-01	225.00	-.3536E-01	-.3536E-01	.3903E+00	-.1608E+00
.5000E-01	270.00	-.5000E-01	-.2002E-08	.3439E+00	-.1160E-02
.5000E-01	315.00	-.3536E-01	.3536E-01	.2931E+00	.3247E+00
.1000E+00	0.00	0.	.1000E+00	.1783E+00	.1295E+01
.1000E+00	45.00	.7071E-01	.7071E-01	.2374E+00	.8055E+00
.1000E+00	90.00	.1000E+00	.1335E-08	.3435E+00	-.4875E-02
.1000E+00	135.00	.7071E-01-	-.7071E-01	.4256E+00	-.4397E+00
.1000E+00	180.00	.2670E-08	-.1000E+00	.4514E+00	-.6368E+00
.1000E+00	225.00	-.7071E-01	-.7071E-01	.4256E+00	-.4397E+00
.1000E+00	270.00	-.1000E+00	-.4005E-08	.3435E+00	-.4875E-02
.1000E+00	315.00	-.7071E-01	.7071E-01	.2374E+00	.8055E+00

```
ZERNICKE COEFFICIENTS/
AVERAGE =   .18672E-01
TILT, X =   .73271E-01            Y =  -.50955E-03
FOCUS   =   .78043E-02
ASTIG   =   .53594E-02   -.45822E-04
COMA    =   .23076E-02    .10126E-04    .41076E-02    .12251E-02
```

FLOW FIELD FOR THETA = 0.000 DEGREES

MACH NUMBER = .700

X	R	PHI	U	V	CP
-.4000E+01	.1000E+01	-.1254E-01	-.1820E-01	.2255E-01	.3588E-01
-.3600E+01	.1000E+01	-.1319E-01	.1550E-01	.2079E-01	-.3144E-01
-.3200E+01	.1000E+01	-.3774E-02	.2254E-01	-.1946E-01	-.4547E-01
-.2800E+01	.1000E+01	-.3857E-02	-.3198E-01	-.4196E-01	.6221E-01
-.2400E+01	.1000E+01	-.3242E-01	-.1075E+00	.1389E-01	.2147E+00
-.2000E+01	.1000E+01	-.8165E-01	-.1236E+00	.1328E+00	.2295E+00
-.1600E+01	.1078E+01	-.1032E+00	-.4248E-01	.1824E+00	.5167E-01
-.1200E+01	.1193E+01	-.8471E-01	.3381E-01	.1370E+00	-.8639E-01
-.8000E+00	.1265E+01	-.5571E-01	.6501E-01	.7915E-01	-.1363E+00
-.4000E+00	.1294E+01	-.2681E-01	.6833E-01	.3319E-01	-.1378E+00
.4974E-13	.1300E+01	.7817E-14	.6565E-01	-.1226E-13	-.1313E+00
.4000E+00	.1294E+01	.2681E-01	.6833E-01	-.3319E-01	-.1378E+00
.8000E+00	.1265E+01	.5571E-01	.6501E-01	-.7915E-01	-.1363E+00
.1200E+01	.1193E+01	.8471E-01	.3381E-01	-.1370E+00	-.8639E-01
.1600E+01	.1078E+01	.1032E+00	-.4248E-01	-.1824E+00	.5167E-01
.2000E+01	.1000E+01	.8165E-01	-.1236E+00	-.1328E+00	.2295E+00
.2400E+01	.1000E+01	.3242E-01	-.1075E+00	-.1389E-01	.2147E+00
.2800E+01	.1000E+01	.3857E-02	-.3198E-01	.4196E-01	.6221E-01
.3200E+01	.1000E+01	.3774E-02	.2254E-01	.1946E-01	-.4547E-01
.3600E+01	.1000E+01	.1319E-01	.1550E-01	-.2079E-01	-.3144E-01
.4000E+01	.1000E+01	.1254E-01	-.1820E-01	-.2255E-01	.3588E-01

CRITICAL PRESSURE COEFFICIENT ON SURFACE = 41.76395

SURFACE DEFINITION (EPS = .300)
POLYNOMIAL COEFICIENTS (A(I), I=0,MAXK) IN X-DIRECTION
.10000E+01 -.56843E-13 -.10591E+00 .28422E-13 -.13454E+00
-.35267E-14 .24631E-01

POLYNOMIAL COEFICIENTS (B(I), I=0,MAXP) IN THETA-DIRECTION
.10000E+01 0. -.18333E+01 0. .84895E+00
0. -.79342E-02

COORDINATES

X	Z	Z-PRIME
-2.200	0.0000	0.0000
-2.000	.0000	-.0000
-1.800	.0247	.2182
-1.600	.0781	.2981
-1.400	.1383	.2935
-1.200	.1926	.2449
-1.000	.2353	.1807
-.800	.2651	.1190
-.600	.2837	.0696
-.400	.2939	.0353
-.200	.2987	.0140
.000	.3000	-.0000
.200	.2987	-.0140
.400	.2939	-.0353
.600	.2837	-.0696
.800	.2651	-.1190
1.000	.2353	-.1807
1.200	.1926	-.2449
1.400	.1383	-.2935
1.600	.0781	-.2981
1.800	.0247	-.2182
2.000	.0000	.0000
2.200	0.0000	0.0000

THETA RADIANS	THETA DEGREES	Z	Z-PRIME
-1.152	-66.0000	0.0000	0.0000
-1.047	-60.0000	.0000	-.0000
-.942	-54.0000	.0107	.1945
-.838	-48.0000	.0386	.3284
-.733	-42.0000	.0776	.4081
-.628	-36.0000	.1224	.4398
-.524	-30.0000	.1683	.4303
-.419	-24.0000	.2113	.3861
-.314	-18.0000	.2482	.3140
-.209	-12.0000	.2764	.2210
-.105	-6.0000	.2940	.1140
.000	.0000	.3000	-.0000
.105	6.0000	.2940	-.1140
.209	12.0000	.2764	-.2210
.314	18.0000	.2482	-.3140
.419	24.0000	.2113	-.3861
.524	30.0000	.1683	-.4303
.628	36.0000	.1224	-.4398
.733	42.0000	.0776	-.4081
.838	48.0000	.0386	-.3284

.942	54.0000	.0107	-.1945
1.047	60.0000	.0000	.0000
1.152	66.0000	0.0000	0.0000

SUM OF SQUARES OF PHASE DISTORTION = .13638E+02

<u>LIST OF REFERENCES</u>

1. Garret N. Vanderplaats,"CONMIN - A FORTRAN Program for Constrained Function Minimization," NASA TM X-62, 282, August 1973.

2. A. E. Fuhs and S. E. Fuhs, "Phase Distortion Due to Airflow Over a Hemispherical Laser Turret," Naval Postgraduate School Report, NPS-69Fu 76101, Sept. 1976.

3. A. E. Fuhs and S. E. Fuhs, "Phase Distortion at High Subsonic Mach Numbers for a Small Perturbation Laser Turret," Proceedings of the Electro-Optics/Laser Conference - 1976, pp 9 - 19. Proceedings published by Industrial and Scientific Conference Management, Inc., Chicago, 1976.

4. A. E. Fuhs, "Distortion of Laser Turret Optics Due to Aircraft Mainstream Flow," Journal of the Optical Society of America, 66, p 1137, October 1976.

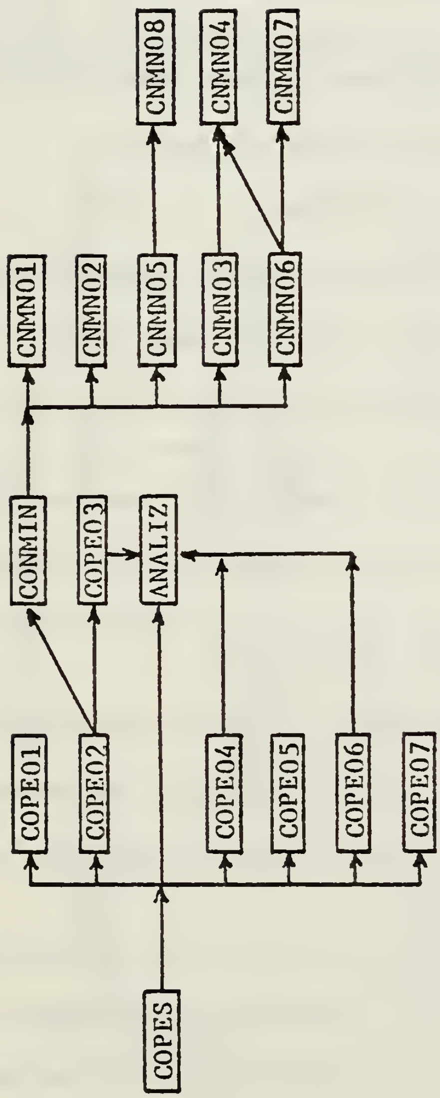
5. Garret N. Vanderplaats, "Inviscid Flow Over Turrets; Optimum Turret Shape," Lecture 5A, Short Course on Laser Aerodynamics presented at AFWL, April, 1977.

6. C. Barry Hogge and R. Russell Butts, "Frequency Spectra for the Geometrical Representation of Wavefront Distortions Due to Atmospheric Turbulence," IEEE Transactions on Antennas and Propagation, AP-24, 2, pp 144-154, 1976.

7. Garret N. Vanderplaats, "The Computer for Design and Optimization," <u>Computing in Applied Mechanics</u>, AMD Vol. 18, ASME, Dec. 1976.

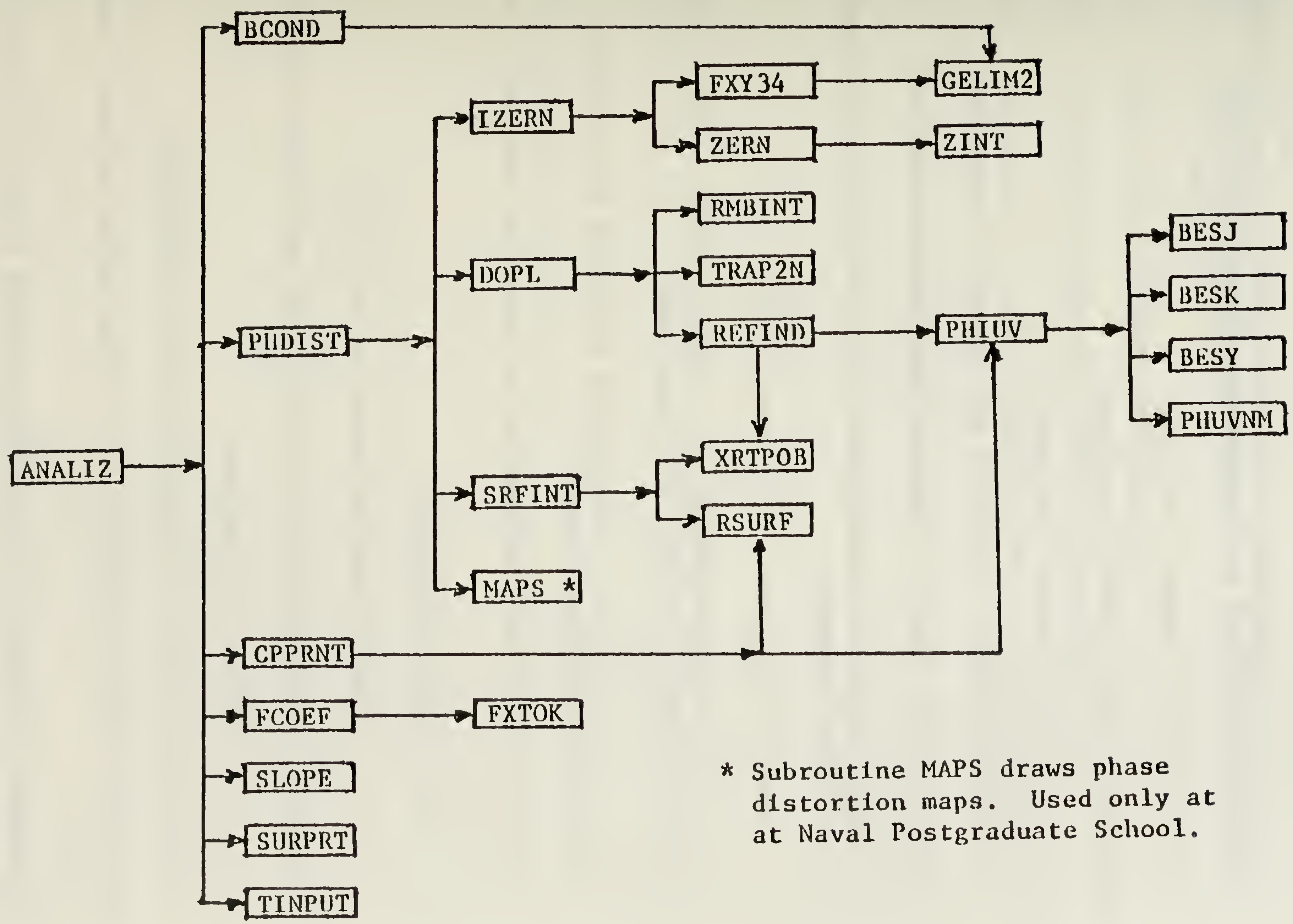
APPENDIX A

PROGRAM FLOW CHARTS AND FORTRAN VARIABLES


COPES
COPE01
COPE02
COPE04
COPE05
COPE06
COPE07
CONMIN
COPE03
ANALIZ
CNMN01
CNMN02
CNMN05
CNMN03
CNMN06
CNMN08
CNMN04
CNMN07

COPES/CONMIN FORTRAN ROUTINES

FORTRAN ROUTINE	PURPOSE
COPES	CONtrol Program for Engineering Synthesis. This is the main program which organizes all design and analysis operations.
COPE01	Reads COPES input data.
COPE02	Controls optimization process.
COPE03	Calculates objective and constraint functions in the form required by CONMIN and performs data transfer operations.
COPE04	Controls sensitivity analysis process.
COPE05	Prints sensitivity results.
COPE06	Controls two-variable function space analysis process.
COPE07	Prints two-variable function space analysis results.
ANALIZ	User supplied subroutine for analysis of the problem under consideration.
CONMIN	Control routine for CONstrained function MINimization.
CNMN01	Calculates gradients by first forward finite difference.
CNMN02	Calculates search direction by Fletcher-Reeves Conjugate Direction Method.
CNMN03	Solves one-dimensional search for unconstrained problems.
CNMN04	Finds the minimum of a function by polynomial interpolation.
CNMN05	Calculates search direction by Zoutendijk's Method of Feasible Directions.
CNMN06	Solves one-dimensional search for constrained problems.
CNMN07	Finds the zero of a function by polynomial interpolation.
CNMN08	Solves the direction-finding sub-problem in Zoutendijk's Method of Feasible Directions.



* Subroutine MAPS draws phase distortion maps. Used only at at Naval Postgraduate School.

LASER TURRET ANALYSIS FORTRAN ROUTINES

FORTRAN ROUTINE	PURPOSE
ANALIZ	Control routine for turret analysis.
BCOND	Determines the dependent coefficients of the polynomial shape functions to satisfy the geometric boundary conditions.
BESJ	Calculates the J Bessel functions.
BESK	Calculates the K Bessel functions.
BESY	Calculates the Y Bessel functions.
CPPRNT	Prints perturbation velocities and pressure coefficient.
DOPL	Calculates the change in optical path length along a ray.
FCOEF	Calculates and saves coefficients for Fourier Series approximation to the turret geometry.
FXTOK	Calculates the Fourier coefficients of X^k.
FXY34	Fits a surface approximation to a three or four cornered segment of phase distortion within the laser beam.
GELIM2	Solves a set of linear simultaneous equations using Gausian elimination with pivot search.
IZERN	Calculates Zernicke functions for a prescribed section of the laser beam.
PHDIST	Calculates phase distortion.
PHIUV	Calculates potential and perturbation velocities.
PHUVNM	Calculates n,m component of potential and perturbation velocities.
REFIND	Calculates index of refraction.
RMBINT	Romberg improvement of trapezoidal rule integration.
RSURF	Calculates radial coordinate, R, of the turret surface, given X and θ.
SLOPE	Calculates the slope of the turret surface in the streamwise direction.

LASER TURRET ANALYSIS FORTRAN ROUTINES - CONCLD.

FORTRAN ROUTINE	PURPOSE
SRFINT	Calculates the distance along a ray from the mirror to the turret surface.
SURPRT	Prints the coordinates defined by the geometric shape functions, f(X) and f(Θ).
TINPUT	Reads laser turret analysis input.
TRAP2N	Numerical integration using trapezoidal rule.
XRTPOB	Calculates the polar coordinates, X, R and Θ of a given point on a ray.
ZERN	Calculates the definite integral of the Zernicke coefficients.
ZINT	Calculates the indefinite integral of the Zernicke coefficients.

FORTRAN VARIABLES COMMONLY USED IN LASER TURRET ANALYSIS PROGRAM

TURRET

ABAR(I)	I-1 coefficient of polynomial in x-direction.
ACL	Half spacing of periodic turret for Fourier series approximation
AL	Turret half length.
AMX(I,m)	Fourier a-sub-m coefficient on I-1 power of x.
ANT(I,J)	Fourier a-sub-n coefficient (J=n+1) on I-1 power of x.
BMX(I,m)	Fourier b-sub-m coefficient on I-1 power of x.
BBAR(I)	I-1 coefficient of polynomial in θ-direction.
EPS	Turret height relative to fuselage radius at $x = \theta = 0$.
MMAX	Maximum number of m-terms in Fourier expansion.
NMAX	Maximum number of n-terms in Fourier expansion.
NTHBC	Number of f and f' pairs of boundary conditions imposed on geometry in θ-direction.
NXBC	Number of f and f' pairs of boundary conditions imposed on geometry in x-direction.
R	Radial coordinate measured from centerline of fuselage.
RFUS	Fuselage radius (meters).
SLOPEX(I)	Turret slope at various x-locations for $\theta = 0$.
THETA	Angular coordinate measured from the vertical axis.
THMAX	Turret half angle.
X	Coordinate along fuselage centerline.
YYPXBC(I,J)	f and f' boundary conditions in x-direction. J = 1 is x location, J = 2 is f boundary condition and J = 3 is f' boundary condition.
YYPTBC(I,J)	f and f' boundary conditions in θ-direction.

MIRROR

GAMMA	Elevation angle measured from horizontal plane.
GAMMAI(I)	Angle GAMMA for I-th beam orientation.
PHI	Azimuth angle measured from negative x-axis.
PHII(I)	Angle PHI for I-th beam orientation.

BEAM

A	Intercept of a ray with the turret surface.
B	Upper limit for phase distortion calculations along a ray.
ETA	Angular point from local z-axis to a point on the beam.
ETAI(I)	ETA for I-th beam element.
NBEAM	Total number of beam orientations considered.
NETAI	Number of values of ETA used in phase distortion calculations.
NRBI	Number of values of RB used in phase distortion calculations.
RB	Radial distance from beam centerline.
RBI(I)	RB for I-th beam element.
WGHTI(I)	Weighting factor for importance of the I-th beam orientation.
Y	Y-coordinate of a point on the beam.
Z	Z-coordinate of a point on the beam.

AERO - OPTICS

AKPRIM	k' in phase distortion relationship.
AMACH	Mach number.
AMACHI(I)	Mach number for I-th beam orientation.
BETA	ABS(1 - AMACH**2)
CP	Pressure coefficient.
DENGAM	Exponent in pressure-density relationship.
DENTRO	Ratio of external air density to sea level air density.
PDISTI(I)	Phase distortion if I-th ray.
PHIPP	Potential function.
RINDEX	Index of refraction.
SUMPD2	Sum of squares of phase distortion.
T(I)	Trapezoidal rule or Romberg integration for phase distortion.
TDENRT	Ratio of internal turret air density to sea level air density.
U	Axial perturbation velocity.
V	Radial perturbation velocity.
WAVEL	Wavelength of laser beam.

ARRAYS USED IN LASER TURRET ANALYSIS PROGRAM AND THEIR REQUIRED DIMENSIONS

ARRAY AND REQUIRED DIMENSION(S)	ACTUAL DIMENSION(S) IN PROGRAM
ABAR(MAXK+1)	ABAR(20)
AMACHI(NBEAM)	AMACHI(30)
AMX(MAXK+1,MMAX)	AMX(10,15)
AN(MAXK+1)	AN(10)
ANT(MAXP+1,NMAX+1)	ANT(10,15)
BBAR(MAXP+1)	BBAR(20)
BMX(MAXK+1,MMAX)	BMX(10,15)
BN(MAXK+1)	BN(10)
ETAI(NETAI)	ETAI(16)
GAMMAI(NBEAM)	GAMMAI(30)
PDISTI(NRBI*NETAI)	PDISTI(200)
PHII(NBEAM)	PHII(30)
RBI(NRBI)	RBI(10)
SLOPEX(30)	SLOPEX(30)
T(KTRAP+1)	T(10)
TITLE(20)	TITLE(20)
WGHTI(NBEAM)	WGHTI(30)
YYPTBC(NTHBC,3)	YYPTBC(10,3)
YYPXBC(NXBC,3)	YYPXBC(10,3)

APPENDIX B

PROGRAM LISTING

```
C     ***********************************************************************
C           COPES - CONTROL PROGRAM FOR ENGINEERING SYNTHESIS.
C     ***********************************************************************
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /COPES1/ ATITLE(20)
      COMMON /COPES2/ RA(5000),IA(1000)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     *1,ISCR2
      COMMON /GLOBCM/ ARRAY(1500)
C     BY G. N. VANDERPLAATS                         OCT., 1974.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     NCALC OPTIONS:
C         0. READ ALL INPUT AND STOP.
C         1. SINGLE PASS ANALYSIS.
C         2. OPTIMIZATION.
C         3. SENSITIVITY - Z = F(X).
C         4. TWO VARIABLE FUNCTION SPACE - Z = F(X,Y).
C     -----------------------------------------------------------------------
C     **********                    INPUT                          **********
C     -----------------------------------------------------------------------
C     DIMENSIONS OF ARRAYS ARRAY, RA AND IA.
      NARRAY=1500
      NDRA=5000
      NDIA=1000
C     READ GENERAL SYNTHESIS CONTROL INPUT.
10    CONTINUE
C     SCRATCH TAPE NUMBERS.
      ISCR1=20
      ISCR2=40
      CALL COPE01 (RA,IA,NDRA,NDIA)
      IF (NCALC.LT.0) GO TO 140
C     CHECK TO INSURE STORAGE REQUIREMENTS DO NOT EXCEED
C     DIMENSIONED SIZES OF ARRAYS RA AND IA.
      NDRA1=LOCR(25)
      NDIA1=LOCI(25)
      IF (NDRA1.LE.NDRA.AND.NDIA1.LE.NDIA) GO TO 20
      WRITE (6,150) NDRA,NDRA1,NDIA,NDIA1
      GO TO 140
20    CONTINUE
C     READ USER INPUT.
      ICALC=1
      CALL ANALIZ(ICALC)
      IF (NCALC.LE.0) GO TO 10
C     -----------------------------------------------------------------------
C     **********                  EXECUTION                       **********
C     -----------------------------------------------------------------------
      IF(NCALC.NE.2) GO TO 50
```

```
C     -----------------------------------------------------------------
C     IF ABS(X(I)).GT.0 OVER-RIDE USER INPUT OF DECISION VARIABLES FOR
C     OPTIMIZATION.
C     -----------------------------------------------------------------
      DO 40 I=1,NDV
      XX=ABS(RA(I))
      IF (XX.LT.1.0E-10) GO TO 40
      N5=LOCR(5)
      M2=LOCI(2)
      DO 30 J=1,NDVTOT
      NN1=IA(M2)
      M2=M2+1
      IF (NN1.NE.I) GO TO 30
      NN1=IA(J)
      ARRAY(NN1)=RA(I)*RA(N5)
30    N5=N5+1
40    CONTINUE
50    CONTINUE
      IF(NCALC.NE.3) GO TO 70
C     -----------------------------------------------------------------
C     TRANSFER NOMINAL VALUES OF SENSITIVITY VARIABLES TO ARRAY.
C     -----------------------------------------------------------------
      M6=LOCI(6)
      M7=LOCI(7)
      DO 60 I=1,NSV
      N=IA(M7)
      M7=M7+1
      NN=IA(M6)
      M6=M6+1
60    ARRAY(NN)=RA(N)
70    CONTINUE
      IF(NCALC.GT.4) GO TO 140
      GO TO (80,90,120,130),NCALC
C     -----------------------------------------------------------------
C                                 ONE ANALYSIS
C     -----------------------------------------------------------------
80    ICALC=2
      CALL ANALIZ(ICALC)
      ICALC=3
      CALL ANALIZ(ICALC)
      GO TO 10
C     -----------------------------------------------------------------
C                                 OPTIMIZATION
C     -----------------------------------------------------------------
90    CONTINUE
      N2=LOCR(2)
      N3=LOCR(3)
      N4=LOCR(4)
      DO 100 I=1,NDV
C     X-VECTOR.
```

```
      M2=LOCI(2)
      DO 91 J=1,NDVTOT
      N=IA(M2)
      M2=M2+1
      IF(N.NE.I) GO TO 91
      N5=LOCR(5)+J-1
      N=IA(J)
      RA(I)=ARRAY(N)/RA(N5)
      GO TO 92
91    CONTINUE
92    CONTINUE
      N2=N2+1
      N3=N3+1
100   N4=N4+1
C     INITIAL ANALYSIS.
C     DESIGN VARIABLE VALUES.
      M2=LOCI(2)
      N5=LOCR(5)
      DO 111 I=1,NDVTOT
      N=IA(M2)
      M=IA(I)
      ARRAY(M)=RA(N)*RA(N5)
      N5=N5+1
111   M2=M2+1
C     ANALIZE INITIAL DESIGN.
      ICALC=2
      CALL ANALIZ(ICALC)
C     OUTPUT INITIAL DESIGN.
      ICALC=3
      CALL ANALIZ(ICALC)
C     OPTIMIZATION.
      CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
C     OUTPUT FINAL DESIGN.
      ICALC=3
      CALL ANALIZ(ICALC)
      GO TO 10
C     ------------------------------------------------------------------
C                           SENSITIVITY ANALYSIS
C     ------------------------------------------------------------------
120   CALL COPE04 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
C     OUTPUT RESULTS.
      CALL COPE05 (RA,IA,NDRA,NDIA)
      GO TO 10
130   CONTINUE
C     ------------------------------------------------------------------
C                         TWO VARIABLE FUNCTION SPACE
C     ------------------------------------------------------------------
      CALL COPE06 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
C     OUTPUT RESULTS.
      CALL COPE07 (RA,IA,NDRA,NDIA)
```

```
      GO TO 10
140   CONTINUE
      REWIND ISCR1
      REWIND ISCR2
      STOP
C     ------------------------------------------------------------
C                               FORMATS
C     ------------------------------------------------------------
150   FORMAT (//5X,60HREQUIRED STORAGE FOR ARRAY RA OR IA EXCEEDS DIMENS
     1IONED SIZE/5X,5HARRAY,2X,9HDIMENSION,2X,8HREQUIRED/7X,2HRA,I8,6X,I
     25/7X,2HIA,I8,6X,I5//5X,22H* * PROGRAM TERMINATED)
      END
```

```
      SUBROUTINE COPE01 (RA,IA,NORA,NOIA)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /COPES1/ ATITLE(20)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPOBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     *1,ISCR2
      DIMENSION RA(NDRA),IA(NDIA),CC(10),TITLE(20)
      DATA STOP1/1HS/,STOP2/1HT/,STOP3/1HO/,STOP4/1HP/,STOP5/4HSTOP/
      DATA END1/1HE/,END2/1HN/,END3/1HD/
      DATA COM/1H$/,COMMA/1H,/,BLANK/1H /,ZERO/1H0/
C     ******************************************************************
C     ROUTINE TO READ CONTROL INPUT FOR COPES.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                          MAR., 1973.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     ------------------------------------------------------------------
C     READ CARD IMAGES AND STORE ON UNIT ISCR2.  STORE ON UNIT ISCR1
C     WITHOUT COMMENT CARDS
C     ------------------------------------------------------------------
      REWIND ISCR1
      REWIND ISCR2
      NCARDS=0
      LOCI(25)=0
      NCOM=0
2     FORMAT(80A1)
      ICARD=0
10    READ(5,2)(RA(I),I=1,80)
      ICARD=ICARD+1
      IFORM=0
C     IS THIS THE TITLE CARD OR A COMMENT CARD?
      IF(RA(1).EQ.COM.OR.NCOM.EQ.0) GO TO 27
      IF(RA(1).EQ.END1.AND.(RA(2).EQ.END2.AND.RA(3).EQ.END3)) GO TO 27
C     UNFORMATTED INPUT CHECK.  USE RA FOR TEMP. STORAGE.
C     CHECK FOR FORMATTED INPUT.
      DO 25 J=1,80
      IF(RA(J).EQ.COMMA) GO TO 26
      IF(RA(J).EQ.COM) GO TO 27
25    CONTINUE
27    CONTINUE
      IFORM=1
      IF(RA(1).NE.COM) NCOM=1
C     NO COMMA FOUND.  THIS DATA IS ALREADY FORMATTED.
      DO 21 J=1,80
21    RA(J+80)=RA(J)
      GO TO 18
26    CONTINUE
      ICARD=ICARD+1
C     BLANK B-VECTOR.
```

```
      DO 11 I=1,80
11    RA(I+80)=BLANK
C     CONVERT UNFORMATTED TO FORMATTED.
      I2=10
      LI=1
      DO 12 I=1,8
C     BLANK WORKING VECTOR, CC.
      DO 13 J=1,10
13    CC(J)=BLANK
C     PUT FIELD I IN CC.
      K=0
      NFLG=0
      DO 14 J=LI,80
      JJ=J
C     IGNORE LEADING BLANKS.
      IF(RA(J).EQ.BLANK.AND.K.LT.1) GO TO 14
C     CHECK FOR COMMA.
      IF(RA(J).EQ.COMMA) GO TO 16
C     CHECK FOR COMMENT.
      IF(RA(J).EQ.COM) GO TO 17
      K=K+1
      IF(K.LE.10) GO TO 29
      K=K-1
      IF(NFLG.GT.0) GO TO 14
      WRITE(6,28)(RA(L),L=1,80),I,(CC(L),L=1,10)
28    FORMAT(/5X,37H* * INPUT FIELD EXCEEDS 10 CHARACTERS/5X,
     * 13HCARD INPUT IS/5X,80A1/5X,17HERROR IS IN FIELD,I5/5X,
     * 45HFIRST 10 NON-BLANK CHARACTERS ARE RETAINED AS,2X,10A1/5X,
     * 24HRESULTS MAY NOT BE VALID)
      NFLG=1
      GO TO 14
29    CC(K)=RA(J)
14    CONTINUE
      GO TO 18
17    CONTINUE
C     COMMENT FOUND.  STORE BEGINNING IN FIELD I OR IN ACTUAL LOCATION,
C     WHICHEVER IS GREATER.
      I1=I2-10
      IF(I1.LT.JJ) I1=JJ
      I1=I1+1
      DO 19 J=JJ,79
      IF(I1.GT.80) GO TO 18
      RA(I1+80)=RA(J+1)
19    I1=I1+1
      GO TO 18
16    CONTINUE
C     STORE CONTENTS OF CC IN B, RIGHT JUSTIFIED.
      LI=JJ+1
      J1=I2+80
      DO 22 J=1,10
```

```
      IF(K.EQ.0) GO TO 23
      IF(CC(K).EQ.BLANK) CC(K)=ZERO
      RA(J1)=CC(K)
      J1=J1-1
22    K=K-1
23    CONTINUE
      I2=I2+10
12    CONTINUE
C     CHECK TO SEE IF NORE THAN 8 FIELDS OF INPUT ARE CONTAINED ON THIS
C     CARD.  IF YES, PRINT ERROR MESSAGE.
      IF(LI.GT.80) GO TO 18
      DO 32 J=LI,80
      IF(RA(J).EQ.COMMA) GO TO 33
      IF(RA(J).EQ.COM) GO TO 18
32    CONTINUE
      GO TO 18
33    WRITE(6,34)(RA(J),J=1,80)
34    FORMAT(/5X,51H* * INPUT DATA CARD CONTAINS MORE THAN EIGHT FIELDS/
     * 5X,13HCARD INPUT IS/5X,80A1/5X,24HRESULTS MAY NOT BE VALID)
18    CONTINUE
      IF(RA(1).NE.COM) WRITE(ISCR1,2)(RA(I),I=81,160)
      NCARDS=NCARDS+1
      IF((RA(1).EQ.STOP1.AND.RA(2).EQ.STOP2).AND.(RA(3).EQ.STOP3.AND.
     * RA(4).EQ.STOP4)) GO TO 20
      WRITE(ISCR2,41)NCARDS,(RA(I),I=1,80)
      IF(IFORM.EQ.0) WRITE(ISCR2,41)NCARDS,(RA(I),I=81,60)
41    FORMAT(I5/80A1)
      IF(RA(1).EQ.END1.AND.(RA(2).EQ.END2.AND.RA(3).EQ.END3)) GO TO 20
      GO TO 10
20    REWIND ISCR1
      REWIND ISCR2
C     ----------------------------------------------------------------
C                     GENERAL SYNTHESIS INFORMATION
C     ----------------------------------------------------------------
C     TITLE.
C ---- DATA BLOCK A.
      READ (ISCR1,750) (ATITLE(I),I=1,20)
      NCALC=-1
      IF(ATITLE(1).EQ.STOPS) RETURN
C     CONTROL PARAMETERS.
C ---- DATA BLOCK B.
      READ (ISCR1,770) NCALC,NOV,NSV,N2VAR,IPNPUT,IPSENS,IP2VAR,IPDBG
      IF (NCALC.LT.0) RETURN
      IF (IPNPUT.GT.1) GO TO 50
      WRITE (6,540)
      WRITE (6,550)
      WRITE (6,560) (ATITLE(I),I=1,20)
C     ----------------------------------------------------------------
C                           CARD IMAGE PRINT
C     ----------------------------------------------------------------
```

```
      IF (IPNPUT.GT.0) GO TO 40
      WRITE (6,430)
      WRITE (6,440)
      DO 30 I=1,ICARD
      READ(ISCR2,41)NCARDS,(RA(J),J=1,80)
30    WRITE(6,450)NCARDS,(RA(J),J=1,80)
      REWIND ISCR2
40    CONTINUE
      WRITE (6,570) (ATITLE(I),I=1,20)
      WRITE (6,580) NCALC,NDV,NSV,N2VAR,IPNPUT,IPSENS,IP2VAR,IPDBG
      WRITE (6,480)
50    NACMX1=0
      NDVTOT=0
      NCONA=0
      NACMX2=0
      IF (NDV.LE.0) GO TO 200
C     ------------------------------------------------------------------
C                          OPTIMIZATION INFORMATION
C     ------------------------------------------------------------------
C     OPTIMIZATION CONTROL VARIABLES. - CONMIN DEPENDENT.
C ---- DATA BLOCK C.
      READ (ISCR1,770) IPRINT,ITMAX,ICNDIR,NSCAL,ITRM,LINOBJ,NACMX1,NFDG
C ---- DATA BLOCK D.
      READ (ISCR1,780) FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,D
     1ABFUN
C ---- DATA BLOCK E.
C     TOTAL NO. OF D. V., OBJECTIVE GLOBAL NUMBER, SIGN
C     ON OPTIMIZATION OBJECTIVE.
      READ (ISCR1,490) NDVTOT,IOBJ,SGNOPT
      IF (NDVTOT.LT.NDV) NDVTOT=NDV
      IF(NACMX1.LE.0) NACMX1=NDV+2
      IF (IPNPUT.GE.2) GO TO 60
      IF (ABS(SGNOPT).LT.1.0E-10) SGNOPT=-1.
      WRITE (6,630) IOBJ,SGNOPT
      WRITE (6,310) IPRINT,ITMAX,ICNDIR,NSCAL,ITRM,LINOBJ,NACMX1,NFDG
      WRITE (6,320) FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,DABF
     1UN
60    N2=NDV+3
      N3=N2+NDV+2
      N4=N3+NDV+2
C ---- DATA BLOCK F.
C     DESIGN VARIABLE INFORMATION, LB, UB, INITIAL VALUE, SCAL.
      IF (IPNPUT.LT.2) WRITE (6,640)
      N5=N4+NDV+2
      IF (N5.LE.NDRA) GO TO 70
      WRITE (6,330)
      WRITE (6,340)
      LOCR(25)=N5
      GO TO 300
70    CONTINUE
```

```
      NSIDE=0
      DO 80 I=1,NDV
      READ (ISCR1,620) RA(N2),RA(N3),RA(I),RA(N4),(TITLE(J),J=1,5)
      IF(RA(N2).GT.-1.0E+15.OR.RA(N3).LT.1.0E+15) NSIDE=1
      IF(RA(N2).LE.-1.0E+15) RA(N2)=-1.1E+15
      IF(RA(N3).GE.1.0E+15) RA(N3)=1.1E+15
      IF (IPNPUT.LT.2) WRITE (6,650) I,RA(N2),RA(N3),RA(I),RA(N4),(TITLE
     1(J),J=1,5)
      N2=N2+1
      N3=N3+1
      N4=N4+1
80    CONTINUE
C ---- DATA BLOCK G.
C     D. V. NO., GLOBAL LOCATION, MULTIPLYING FACTOR.
      IF (IPNPUT.LT.2) WRITE (6,500)
      N5=4*NDV+9
      M2=NDVTOT+1
      N6=N5+NDVTOT
      M3=M2+NDVTOT
      IF (N6.LE.NDRA) GO TO 90
      WRITE (6,330)
      WRITE (6,350)
      LOCR(25)=N5
      GO TO 300
90    CONTINUE
      IF (M3.LE.NDIA) GO TO 100
      WRITE (6,360)
      WRITE (6,350)
      LOCI(25)=M3
      GO TO 300
100   CONTINUE
      DO 110 I=1,NDVTOT
      READ (ISCR1,490) IA(M2),IA(I),RA(N5)
      IF(ABS(RA(N5)).LT.1.0E-20) RA(N5)=1.0
      IF (IPNPUT.LT.2) WRITE (6,510) I,IA(M2),IA(I),RA(N5)
      M2=M2+1
      N5=N5+1
110   CONTINUE
      NCON=0
C ---- DATA BLOCK H.
C     NUMBER OF CONSTRAINT SETS.
      READ (ISCR1,490) NCONS
      IF (IPNPUT.LT.2) WRITE (6,670)
      IF (IPNPUT.LT.2) WRITE (6,680) NCONS
      IF (NCONS.EQ.0) GO TO 200
      IF (IPNPUT.LT.2) WRITE (6,690)
      N6=4*NDV+NDVTOT+9
      M3=2*NDVTOT+1
      M4=2*NDVTOT+NCONS
      M4A=M4+1
```

```
      L=1
C ---- DATA BLOCK I.
      NCONA=0
      DO 170 I=1,NCONS
      NNN=N6+3
      IF (NNN.GT.NDRA) GO TO 180
C     GLOBAL NO. 1, GLOBAL NO. 2, LINEAR CONSTRAINT ID.
      READ(ISCR1,770) ICONI,JCONI,LCONI
C     LB, NORM, UB, NORM.
      READ(ISCR1,780)(RA(J),J=N6,NNN)
      IF(RA(N6).LE.-1.0E+15) RA(N6)=-1.1E+15
      IF(RA(N6+2).GE.1.0E+15) RA(N6+2)=1.1E+15
      IF(RA(N6+1).LT.1.0E-20) RA(N6+1)=ABS(RA(N6))
      IF(RA(N6+1).LT.0.1) RA(N6+1)=0.1
      IF(RA(N6+3).LT.1.0E-20) RA(N6+3)=ABS(RA(N6+2))
      IF(RA(N6+3).LT.0.1) RA(N6+3)=0.1
C     NUMBER OF VARIABLES IN THIS SET.
      NVAR=JCONI-ICONI+1
      IF (NVAR.LT.1) NVAR=1
      NCONA=NCONA+NVAR
C     HOW MANY CONSTRAINTS?
      J1=0
      IF (RA(N6).GE.-1.0E+15) J1=1
      IF(RA(N6+2).LT.1.0E+15) J1=J1+1
      NCONI=J1*NVAR
      NCON=NCON+NCONI
      IF (J1.EQ.0) GO TO 130
C     ADD LINEAR CONSTRAINT IDENTIFIERS TO ISC.
      DO 120 J=1,NCONI
      M4=M4+1
      MMM=M4
      IF (MMM.GT.NDIA) GO TO 190
120   IA(M4)=LCONI
130   CONTINUE
C     ADD LB, UB AND SCAL TO BLU IF NVAR.GT.1.
      IF (NVAR.EQ.1) GO TO 150
      NVAR1=NVAR-1
      DO 140 J=1,NVAR1
      NNN=N6+7
      IF (NNN.GT.NDRA) GO TO 180
      RA(N6+4)=RA(N6)
      RA(N6+5)=RA(N6+1)
      RA(N6+6)=RA(N6+2)
      RA(N6+7)=RA(N6+3)
      N6=N6+4
140   CONTINUE
150   CONTINUE
C     ADD CONSTRAINED VARIABLE GLOBAL IDENTIFIERS TO ICON.
      ICON1=ICONI
      DO 160 J=1,NVAR
```

```
      MMM=M3
      IF (MMM.GT.NDIA) GO TO 190
      IA(M3)=ICON1
      ICON1=ICON1+1
      IF(J.EQ.1) GO TO 160
C     SHIFT ISC VECTOR.
      L1=M4+1
      L2=M4
      DO 165 K=M4A,M4
      IA(L1)=IA(L2)
      L1=L1-1
  165 L2=L2-1
      M4=M4+1
      M4A=M4A+1
160   M3=M3+1
      IF (IPNPUT.LT.2) WRITE (6,660) L,ICONI,JCONI,LCONI,RA(N6),RA(N6+1)
     1,RA(N6+2),RA(N6+3)
      N6=N6+4
      L=NCON+1
170   CONTINUE
      IF (IPNPUT.LT.2) WRITE (6,470) NCONA
      GO TO 200
180   WRITE (6,330)
      WRITE (6,370)
      LOCR(25)=NNN
      GO TO 300
190   WRITE (6,360)
      WRITE (6,370)
      LOCI(25)=MMM
      GO TO 300
200   CONTINUE
      NSOBJ=0
      NSVTOT=0
C     STARTING LOCATIONS FOR SENSITIVITY INFORMATION.
      NSVR=4*NDV+NDVTOT+4*NCONA+9
      NSVI=2*(NDV+NCONA)+2*NDVTOT+NCONA+1
      IF (NSV.LE.0) GO TO 240
C     -----------------------------------------------------------------
C                            SENSITIVITY INFORMATION
C     -----------------------------------------------------------------
      IF (IPNPUT.LT.2) WRITE (6,590)
C ---- DATA BLOCK J, PART 1.
C     NSOBJ.
      READ (ISCR1,770) NSOBJ
C ---- DATA BLOCK J, PART 2.
C     NSENSZ.
      M5=NSVI
      MM5=M5+NSOBJ-1
      IF (MM5.LE.NDIA) GO TO 210
      WRITE (6,360)
```

```
      WRITE (6,380)
      LOCI(25)=MM5
      GO TO 300
210   CONTINUE
      READ (ISCR1,770) (IA(I),I=M5,MM5)
      IF (IPNPUT.LT.2) WRITE (6,530) NSOBJ
      IF (IPNPUT.LT.2) WRITE (6,520) (IA(I),I=M5,MM5)
      IF (IPNPUT.LT.2) WRITE (6,600)
      N7=NSVR
      M6=NSVI+NSOBJ
      M7=M6+NSV
      DO 230 I=1,NSV
C ---- DATA BLOCK K, PART 1.
C     ISENS, NSENS.
      READ(ISCR1,770)IA(M6),NN1
      NN7=N7+NN1-1
      IF (NN7.LE.NDRA) GO TO 220
      WRITE (6,330)
      WRITE (6,390)
      LOCR(25)=NN7
      GO TO 300
220   CONTINUE
C ---- DATA BLOCK K, PART 2.
C     SENS.
      READ (ISCR1,780) (RA(J),J=N7,NN7)
      IF(IPNPUT.GE.2) GO TO 225
      JJ=N7+5
      IF(JJ.GT.NN7) JJ=NN7
      WRITE(6,610)I,IA(M6),(RA(J),J=N7,JJ)
      JJ=JJ+1
      IF(JJ.LE.N7) WRITE(6,615)(RA(J),J=JJ,N7)
225   CONTINUE
      NSVTOT=NSVTOT+NN1
      IA(M7)=N7
      N7=NN7+1
      M6=M6+1
      M7=M7+1
230   CONTINUE
240   CONTINUE
      M2VX=0
      M2VY=0
      IF (N2VAR.LE.0) GO TO 270
C     -----------------------------------------------------------------
C                 TWO-VARIABLE FUNCTION SPACE INFORMATION
C     -----------------------------------------------------------------
C ---- DATA BLOCK L.
C     VARIABLE NUMBERS AND NUMBER OF VALUES OF X AND Y.
      READ (ISCR1,770) N2VX,M2VX,N2VY,M2VY
      N8=NSVR+NSVTOT
      M8=NSVI+NSOBJ+2*NSV
```

```
      MM8=M8+N2VAR-1
      IF (MM8.LE.NDIA) GO TO 250
      WRITE (6,360)
      WRITE (6,400)
      LOCI(25)=MM8
      GO TO 300
250   CONTINUE
C ---- DATA BLOCK M.
C     GLOBAL VARIABLE NUMBERS CORRESPONDING TO FUNCTIONS OF X AND Y.
      READ (ISCR1,770) (IA(I),I=M8,MM8)
      IF (IPNPUT.LT.2) WRITE (6,730)
      IF (IPNPUT.LT.2) WRITE (6,740) (IA(I),I=M8,MM8)
C ---- DATA BLOCK N.
C     VALUES OF X COMPONENTS.
      NN8=N8+M2VX-1
      IF (NN8.GT.NDRA) GO TO 260
      READ (ISCR1,780) (RA(I),I=N8,NN8)
      IF (IPNPUT.LT.2) WRITE (6,700) N2VX
      IF (IPNPUT.LT.2) WRITE (6,720) (RA(I),I=N8,NN8)
C ---- DATA BLOCK O.
C     VALUES OF Y COMPONENTS.
      N9=N8+M2VX
      NN9=N9+M2VY-1
      NN8=NN9
      READ (ISCR1,780) (RA(I),I=N9,NN9)
      IF (IPNPUT.LT.2) WRITE (6,710) N2VY
      IF (IPNPUT.LT.2) WRITE (6,720) (RA(I),I=N9,NN9)
      GO TO 270
260   WRITE (6,330)
      WRITE (6,400)
      LOCR(25)=NN8
      GO TO 300
270   CONTINUE
C     ------------------------------------------------------------------
C                       DYNAMIC STORAGE ALLOCATION
C     ------------------------------------------------------------------
      NDV2=NDV+2
C     REAL VARIABLES.
C     X.
      LOCR(1)=1
C     VLB.
      LOCR(2)=NDV+3
C     VUB.
      LOCR(3)=LOCR(2)+NDV2
C     SCAL.
      LOCR(4)=LOCR(3)+NDV2
C     AMULT.
      LOCR(5)=LOCR(4)+NDV2
C     BLU.
      LOCR(6)=LOCR(5)+NDVTOT
```

```
C       SENS.
        LOCR(7)=LOCR(6)+4*NCONA
C       XM2V.
        LOCR(8)=LOCR(7)+NSVTOT
C       YM2V.
        LOCR(9)=LOCR(8)+M2VX
C       EXECUTION LEVEL ARRAYS.
        LOCR(10)=LOCR(9)+M2VY
        DO 280 I=11,25
280     LOCR(I)=LOCR(10)
C       INTEGER VARIABLES.
C       IDSGN.
        LOCI(1)=1
C       NDSGN.
        LOCI(2)=NDVTOT+1
C       ICON.
        LOCI(3)=LOCI(2)+NDVTOT
C       ISC.
        LOCI(4)=LOCI(3)+NCONA
C       NSENSZ
        LOCI(5)=LOCI(4)+2*(NDV+NCONA)
C       ISENS.
        LOCI(6)=LOCI(5)+NSOBJ
C       NSENS.
        LOCI(7)=LOCI(6)+NSV
C       N2VZ.
        LOCI(8)=LOCI(7)+NSV
C       EXECUTION LEVEL ARRAYS.
        LOCI(9)=LOCI(8)+N2VAR
        DO 290 I=10,25
290     LOCI(I)=LOCI(9)
C       STORAGE FOR CONMIN ARRAYS.
        IF(NCALC.NE.2) GO TO 295
        NRI=NDV
        IF (NACMX1.GT.NRI) NRI=NACMX1
        NR=3*NCON+8*NDV+NACMX1*(NDV2+NACMX1)+NRI+4
        NI=NACMX1+2*NRI
        LOCR(25)=LOCR(10)+NR
        LOCI(25)=LOCI(9)+NI
        GO TO 300
295     NR=NSV
        IF(NSOBJ.GT.NR) NR=NSOBJ
        IF(NCALC.EQ.3) LOCR(25)=LOCR(10)+NR
        IF(NCALC.EQ.4) LOCR(25)=LOCR(10)+N2VAR
300     CONTINUE
        IF(IPNPUT.LT.2) WRITE(6,410)LOCR(10),LOCR(25),NDRA,LOCI(9),LOCI(25
       *),NDIA
        RETURN
C       ----------------------------------------------------------------
C                                FORMATS
```

```
C     ------------------------------------------------------------------
310   FORMAT (/5X,58HCONMIN PARAMETERS (IF ZERO, CONMIN DEFAULT WILL OVE
     1R-RIDE)//5X,6HIPRINT,2X,5HITMAX,3X,6HICNDIR,3X,5HNSCAL,3X,4HITRM,3
     2X,6HLINOBJ,2X,6HNACMX1,3X,4HNFDG/8I8)
320   FORMAT (/6X,4HFDCH,12X,5HFDCHM,11X,2HCT,14X,5HCTMIN/1X,4(2X,E14.5)
     *//6X,3HCTL,13X,6HCTLMIN,10X,5HTHETA,11X,3HPHI/1X,4(2X,E14.5)//
     * 6X,6HDELFUN,10X,6HDABFUN/1X,2(2X,E14.5))
330   FORMAT (//5X,54HREQUIRED STORAGE IN ARRAY RA EXCEEDS AVAILABLE STO
     1RAGE)
340   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK F)
350   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK G)
360   FORMAT (//5X,54HREQUIRED STORAGE IN ARRAY IA EXCEEDS AVAILABLE STO
     1RAGE)
370   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK I)
380   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK J)
390   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK K)
400   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK L)
410   FORMAT(//5X,25HDATA STORAGE REQUIREMENTS//17X,4HREAL,26X,
     * 7HINTEGER/7X,27HINPUT  EXECUTION  AVAILABLE,5X,
     * 27HINPUT  EXECUTION  AVAILABLE/1X,3I10,2X,3I10)
420   FORMAT (A1,A2,A1,19A4)
430   FORMAT (1H1,4X,27HCARD IMAGES OF CONTROL DATA///5X,4HCARD,20X,5HIM
     1AGE)
440   FORMAT (1H0)
450   FORMAT(I8,1H),2X,80A1)
470   FORMAT (/5X,40HTOTAL NUMBER OF CONSTRAINED PARAMETERS =,I5)
480   FORMAT (//5X,26HCALCULATION CONTROL, NCALC/5X,5HVALUE,3X,7HMEANING
     1/7X,1H1,5X,15HSINGLE ANALYSIS/7X,1H2,5X,12HOPTIMIZATION/7X,1H3,5X,
     2 11HSENSITIVITY/7X,1H4,5X,27HTWO-VARIABLE FUNCTION SPACE)
490   FORMAT(2I10,F10.2)
500   FORMAT (//5X,16HDESIGN VARIABLES/11X,5HD. V.,5X,6HGLOBAL,4X,11HMUL
     1TIPLYING/5X,2HID,5X,3HNO.,5X,8HVAR. NO.,5X,6HFACTOR)
510   FORMAT (2I7,5X,I5,6X,E12.5)
520   FORMAT (5X,16I5)
530   FORMAT (/5X,34HNUMBER OF SENSITIVITY OBJECTIVES =,I5/5X,53HGLOBAL
     1NUMBERS ASSOCIATED WITH SENSITIVITY OBJECTIVES)
540   FORMAT (1H1,//////,5X,47HCCCCCCC   OOOOOOO   PPPPPPP   EEEEEEE   S
     1SSSSSS/5X,47HC          O     O   P     P   E         S       /5X,47
     2HC         O     O   P     P   E         S        /5X,47HC        O
     3     O   PPPPPPP   EEEE      SSSSSSS/5X,47HC         O     O   P
     4       E               S/5X,47HC         O     O   P         E
     5         S/5X,47HCCCCCCC   OOOOOOO   P         EEEEEEE   SSSSSSS
     6)
550   FORMAT (/////,18X,19HN A S A  -  A M E S//14X,29HC O N T R O L   P
     1 R O G R A M//26X,5HF O R//8X,41HE N G I N E E R I N G   S Y N T H
     2 E S I S)
560   FORMAT (/////24X,9HT I T L E//5X,20A4)
570   FORMAT (1H1,4X,6HTITLE:/5X,20A4)
580   FORMAT (///5X,19HCONTROL PARAMETERS:/5X,42HCALCULATION CONTROL,
     1                NCALC =,I5/5X,42HNUMBER OF GLOBAL DESIGN VARIABLES,
```

```
     2NDV =,I5/5X,42HNUMBER OF SENSITIVITY VARIABLES,      NSV =,I5/5X,42
     3HNUMBER OF FUNCTIONS IN TWO-SPACE,  N2VAR =,I5/5X,42HINPUT INFORMA
     4TION PRINT CODE,      IPNPUT =,I5/5X,42HSENSITIVITY PRINT CODE,
     5      IPSENS =,I5/5X,42HTWO-SPACE PRINT CODE,                IP2VAR
     6=,I5/5X,42HDEBUG PRINT CODE,                     IPDBG =,I5)
590   FORMAT (///5X,27H* * SENSITIVITY INFORMATION)
600   FORMAT (/14X,6HGLOBAL,4X,7HNOMINAL/5X,6HNUMBER,2X,8HVARIABLE,4X,5H
     1VALUE,6X,18HOFF-NOMINAL VALUES)
610   FORMAT (5X,I4,I8,5X,E12.5,1X,5E11.4)
615   FORMAT(35X,5E11.4)
620   FORMAT (4F10.2,10A4)
630   FORMAT(/5X,35HGLOBAL VARIABLE NUMBER OF OBJECTIVE,10X,1H=,I5/5X,
     1 46HMULTIPLIER (NEGATIVE INDICATES MINIMIZATION) =,E12.4)
640   FORMAT (/5X,27HDESIGN VARIABLE INFORMATION/5X,50HNON-ZERO INITIAL
     1VALUE WILL OVER-RIDE MODULE INPUT/5X,5HD. V.,5X,5HLOWER,10X,5HUPPE
     2R,9X,7HINITIAL/5X,3HNO.,7X,5HBOUND,10X,5HBOUND,10X,5HVALUE,10X,5HS
     3CALE)
650   FORMAT (I8,4X,E12.5,3X,E12.5,3X,E12.5,3X,E12.5,5A4)
660   FORMAT (I8,I7,2I8,5X,E12.5,3X,E12.5,3X,E12.5,3X,E12.5)
670   FORMAT (//5X,22HCONSTRAINT INFORMATION)
680   FORMAT (/5X,9HTHERE ARE,I3,16H CONSTRAINT SETS)
690   FORMAT (11X,6HGLOBAL,2X,6HGLOBAL,2X,6HLINEAR,6X,5HLOWER,6X,
     * 13HNORMALIZATION,7X,5HUPPER,6X,13HNORMALIZATION/6X,2HID,3X,
     * 6HVAR. 1,2X,6HVAR. 2,4X,2HID,8X,5HBOUND,9X,6HFACTOR,10X,
     * 5HBOUND,9X,6HFACTOR)
700   FORMAT (//5X,49HGLOBAL VARIABLE NUMBER CORRESPONDING TO X, N2VX =,
     1I5//5X,20HVALUES OF X-VARIABLE)
710   FORMAT (//5X,49HGLOBAL VARIABLE NUMBER CORRESPONDING TO Y, N2VY =,
     1I5//5X,20HVALUES OF Y-VARIABLE)
720   FORMAT (3X,5E12.4)
730   FORMAT (///5X,51H* TWO-VARIABLE FUNCTION SPACE MAPPING INFORMATI
     1ON//5X,52HGLOBAL VARIABLE NUMBERS ASSOCIATED WITH F(X,Y), M2VZ)
740   FORMAT (5X,10I5)
750   FORMAT (20A4)
770   FORMAT(8I10)
780   FORMAT (8F10.2)
      END
```

```
      SUBROUTINE COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /COPES1/ ATITLE(20)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     *1,ISCR2
      DIMENSION ARRAY(NARRAY),RA(NDRA),IA(NDIA)
C     ******************************************************************
C     ROUTINE TO CONTROL OPTIMIZATION.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                                MAR., 1973.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     ------------------------------------------------------------------
C                             ARRAY DIMENSIONS
C     ------------------------------------------------------------------
      NN1=NDV+2
      NN2=2*NDV+NCON
      NN3=NACMX1
      NN4=NN3
      IF (NDV.GT.NN4) NN4=NDV
      NN5=2*NN4
C     ------------------------------------------------------------------
C                           ARRAY STARTING LOCATIONS
C     ------------------------------------------------------------------
C     X, VLB, VUB, DF, A, S, G1, G2, C, B, SCAL, ISC, IC, MS1
      NX=1
      NVLB=LOCR(2)
      NVUB=LOCR(3)
      NNSCAL=LOCR(4)
      NDF=LOCR(10)
      NG=NDF+NN1
      NA=NG+NN2
      NS=NA+NN1*NN3
      NG1=NS+NN1
      NG2=NG1+NN2
      NC=NG2+NN2
      NB=NC+NN4
      NISC=LOCI(4)
      NIC=LOCI(10)
      NMS1=NIC+NN3
C     ------------------------------------------------------------------
C                              OPTIMIZATION
C     ------------------------------------------------------------------
      IGOTO=0
C     CALL CONMIN (X,DF,G,ISC,IC,A,S,G1,G2,C,MS1,B,VLB,VUB,
C    *SCAL,N1,N2,N3,N4,N5)
50    CONTINUE
      CALL CONMIN (RA(NX),RA(NDF),RA(NG),IA(NISC),IA(NIC),RA(NA),RA(NS),
```

```
     1RA(NG1),RA(NG2),RA(NC),IA(NMS1),RA(NB),RA(NVLB),RA(NVUB),RA(NNSCAL
     2),NN1,NN2,NN3,NN4,NN5)
C     ANALIZE.
      CALL COPE03 (ARRAY,NARRAY,RA(NX),RA(NDF),RA(NG),IA(NIC),RA(NA),NN1
     1,NN2,NN3,RA,IA,NDRA,NDIA)
      IF(IGOTO.GT.0) GO TO 50
      RETURN
      END
```

```
      SUBROUTINE COPE03 (ARRAY,NARRAY,X,DF,G,IC,A,NN1,NN2,NN3,RA,IA,NDRA
     1,NDIA)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPOBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     *1,ISCR2
      DIMENSION ARRAY(NARRAY),RA(NDRA),IA(NDIA)
      DIMENSION X(NN1),DF(NN1),G(NN2),IC(NN3),A(NN3,NN1)
C     ******************************************************************
C     BUFFER BETWEEN CONMIN AND COPES FUNCTION EVALUATION.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                             MAR., 1973.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     INITIAL ANALYSIS HAS BEEN DONE.  IF ITER = 0, GO EVALUATE
C     OBJECTIVE AND CONSTRAINTS.
      IF(ITER.LT.1) GO TO 25
C     ------------------------------------------------------------------
C               TRANSFER DESIGN VARIABLE VALUES TO USER ARRAY
C     ------------------------------------------------------------------
      N5=LOCR(5)
      M2=LOCI(2)
      DO 20 I=1,NDVTOT
      N=IA(M2)
      M=IA(I)
      ARRAY(M)=RA(N)*RA(N5)
      N5=N5+1
      M2=M2+1
      M9=M9+1
20    CONTINUE
C     ------------------------------------------------------------------
C                                   ANALIZE
C     ------------------------------------------------------------------
      ICALC=2
      CALL ANALIZ(ICALC)
C     ------------------------------------------------------------------
C                                  OBJECTIVE
C     ------------------------------------------------------------------
25    CONTINUE
      OBJ=-SGNOPT*ARRAY(IOBJ)
      IF (NCON.EQ.0) RETURN
C     ------------------------------------------------------------------
C                              CONSTRAINT VALUES
C     ------------------------------------------------------------------
      M3=LOCI(3)
      N6=LOCR(6)
      N=0
      DO 40 I=1,NCONA
C     PARAMETER IDENTIFIER.
```

```
      NN=IA(M3)
      CC=ARRAY(NN)
C     LOWER BOUND.
      BB=RA(N6)
      IF (BB.LT.-1.0E+15) GO TO 30
C     NORMALIZATION FACTOR.
      C1=RA(N6+1)
C     CONSTRAINT VALUE.
      N=N+1
      G(N)=(BB-CC)/C1
C     UPPER BOUND.
30    BB=RA(N6+2)
C     NORMALIZATION FACTOR.
      C1=RA(N6+3)
      N6=N6+4
      M3=M3+1
      IF (BB.GT.1.0E+15) GO TO 40
C     CONSTRAINT VALUE.
      N=N+1
      G(N)=(CC-BB)/C1
40    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COPE04 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /COPES1/ ATITLE(20)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     *1,ISCR2
      DIMENSION ARRAY(NARRAY),RA(NDRA),IA(NDIA)
C     ******************************************************************
C     ROUTINE TO PROVIDE SENSITIVITY INFORMATION WITH RESPECT TO
C     A PRESCRIBED SET OF DESIGN VARIABLES.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                            MAR., 1973.
C     STORE OUTPUT ON UNIT ISCR1.
      REWIND ISCR1
C     ------------------------------------------------------------------
C                    WRITE BASIC INFORMATION ON UNIT ISCR1
C     ------------------------------------------------------------------
C     TITLE.
      WRITE (ISCR1,330) (ATITLE(I),I=1,20)
C     NCALC, NSV, NSOBJ
      WRITE (ISCR1,340) NCALC,NSV,NSOBJ
C     ISENS(I),I=1,NSV.
      M6=LOCI(6)
      M7=M6+NSV-1
      WRITE (ISCR1,340) (IA(I),I=M6,M7)
C     NSENSZ(I),I=1,NSOBJ.
      M5=LOCI(5)
      M6=M5+NSOBJ-1
      WRITE (ISCR1,340) (IA(I),I=M5,M6)
      JCALC=3
      ICALC=2
C     ------------------------------------------------------------------
C     **********                  NOMINAL                    **********
C     ------------------------------------------------------------------
      CALL ANALIZ(ICALC)
      IF (IPSENS.GT.0) CALL ANALIZ (JCALC)
C     ------------------------------------------------------------------
C                    WRITE NOMINAL RESULTS ON UNIT ISCR1
C     ------------------------------------------------------------------
C     SENS(I,1).
      M7=LOCI(7)
      N10=LOCR(10)
      N11=N10
      DO 160 I=1,NSV
      N=M7+I-1
      N=IA(N)
      RA(N11)=RA(N)
160   N11=N11+1
```

```
      N11=N10+NSV-1
      WRITE(ISCR1,350)(RA(I),I=N10,N11)
C     SENSITIVITY OBJECTIVES, OBJZ.
      M5=LOCI(5)
      N10=LOCR(10)
      N11=N10
      DO 170 I=1,NSOBJ
      M=M5+I-1
      M=IA(M)
      RA(N11)=ARRAY(M)
170   N11=N11+1
      N11=N10+NSOBJ-1
      WRITE(ISCR1,350)(RA(I),I=N10,N11)
C     ------------------------------------------------------------------
C     **********                SENSITIVITIES                 **********
C     ------------------------------------------------------------------
      NSVAL=LOCR(8)-LOCR(7)-NSV
      NSVAL1=0
      DO 320 II=1,NSV
C     SENSITIVITY VARIABLE NUMBER.
      M6=LOCI(6)+II-1
      ISENS=IA(M6)
C     STARTING LOCATION OF SENSITIVITY VALUES IN RA (M7).
      M7=LOCI(7)+II-1
      M8=IA(M7+1)
      M7=IA(M7)
C     NUMBER OF SENSITIVITY VARIABLES, NSENS.
      NSENS=M8-M7
      IF (II.EQ.NSV) NSENS=NSVAL-NSVAL1+1
      IF (NSENS.LE.1) GO TO 320
C     WRITE ISENS AND NSENS ON UNIT ISCR1.
      NSENSI=NSENS-1
      WRITE (ISCR1,340) ISENS,NSENSI
C     ------------------------------------------------------------------
C               VARY THE VALUE OF THE SENSITIVITY PARAMETER
C     ------------------------------------------------------------------
      DO 310 JJ=2,NSENS
      NSVAL1=NSVAL1+1
      K=M7+JJ-1
      ARRAY(ISENS)=RA(K)
C     WRITE SENS(I,J) ON UNIT ISCR1.
      WRITE (ISCR1,350) ARRAY(ISENS)
C     ANALIZE.
      CALL ANALIZ(ICALC)
      IF(IPSENS.GT.0) CALL ANALIZ(JCALC)
C     ------------------------------------------------------------------
C               WRITE SENSITIVITY RESULTS ON UNIT ISCR1
C     ------------------------------------------------------------------
C     OBJZ.
      M5=LOCI(5)
```

```
      N10=LOCR(10)
      N11=N10
      DO 300 I=1,NSOBJ
      M=M5+I-1
      M=IA(M)
      RA(N11)=ARRAY(M)
300   N11=N11+1
      N11=N10+NSOBJ-1
      WRITE(ISCR1,350)(RA(I),I=N10,N11)
310   CONTINUE
      ARRAY(ISENS)=RA(M7)
320   CONTINUE
      RETURN
C     -----------------------------------------------------------------
C                               FORMATS
C     -----------------------------------------------------------------
330   FORMAT (20A4)
340   FORMAT (16I5)
350   FORMAT (5F15.8)
      END
```

```
      SUBROUTINE COPE05 (RA,IA,NDRA,NDIA)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /COPES1/ ATITLE(20)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     *1,ISCR2
      DIMENSION RA(NDRA),IA(NDIA)
C     ******************************************************************
C     ROUTINE TO PRINT SENSITIVITY INFORMATION STORED ON UNIT ISCR1.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                          JULY, 1974.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
      REWIND ISCR1
C     ------------------------------------------------------------------
C                            GENERAL INFORMATION
C     ------------------------------------------------------------------
C     TITLE.
      READ (ISCR1,60) (ATITLE(I),I=1,20)
C     NCALC, NSV, NSOBJ
      READ (ISCR1,70) NCALC,NSV,NSOBJ
      IF(NCALC.NE.3) RETURN
      WRITE(6,80)
      WRITE (6,50) (ATITLE(I),I=1,20)
      WRITE (6,90) NSV,NSOBJ
C     ISENS(I),I=1,NSV.
      READ (ISCR1,70) (IA(I),I=1,NSV)
      WRITE (6,110)
      WRITE (6,120) (IA(I),I=1,NSV)
C     NSENSZ(I),I=1,NSOBJ.
      READ (ISCR1,70) (IA(I),I=1,NSOBJ)
      WRITE (6,130)
      WRITE (6,120) (IA(I),I=1,NSOBJ)
C     -----------------------------------------------------------------
C                            NOMINAL INFORMATION
C     -----------------------------------------------------------------
C     SENS(I),I=1,NSV.
      READ (ISCR1,140) (RA(I),I=1,NSV)
      WRITE (6,150)
      WRITE (6,160) (RA(I),I=1,NSV)
C     OBJZ(I),I=1,NSOBJ.
      READ (ISCR1,140) (RA(I),I=1,NSOBJ)
      WRITE (6,170)
      WRITE (6,160) (RA(I),I=1,NSOBJ)
C     ------------------------------------------------------------------
C     **********             SENSITIVITY INFORMATION            **********
C     ------------------------------------------------------------------
      WRITE (6,180)
      DO 40 ISENS=1,NSV
```

```
C       ISENSI, NSENSI
        READ (ISCR1,70) ISENSI,NSENSI
        WRITE (6,190) ISENSI
        IF (NSENSI.EQ.0) WRITE (6,200)
        IF (NSENSI.EQ.0) GO TO 40
        DO 30 JJ=1,NSENSI
C       SENS(I,J).
        READ (ISCR1,140) SENSIJ
C       OBJZ(I),I=1,NSOBJ.
        READ (ISCR1,140) (RA(I),I=1,NSOBJ)
        N=MIN0(4,NSOBJ)
        WRITE (6,210) SENSIJ,(RA(I),I=1,N)
        N=(NSOBJ-1)/4
        IF (N.LT.1) GO TO 20
        L1=5
        DO 10 I=1,N
        L2=L1+3
        L2=MIN0(L2,NSOBJ)
        WRITE (6,220) (RA(J),J=L1,L2)
10      L1=L1+4
20      CONTINUE
30      CONTINUE
40      CONTINUE
        RETURN
C       ------------------------------------------------------------------
C                                  FORMATS
C       ------------------------------------------------------------------
50      FORMAT (//5X,5HTITLE/5X,20A4)
60      FORMAT (20A4)
70      FORMAT (16I5)
80      FORMAT (1H1,4X,47HSTANDARD SENSITIVITY ANALYSIS RESULTS (NCALC=3))
90      FORMAT (//5X,36HNUMBER OF SENSITIVITY VARIABLES, NSV,9X,1H=,I5/5X,
       139HNUMBER OF SENSITIVITY OBJECTIVES, NSOBJ,6X,1H=,I5)
110     FORMAT (//5X,52HGLOBAL NUMBERS ASSOCIATED WITH SENSITIVITY VARIABL
       1ES)
120     FORMAT (5X,10I5)
130     FORMAT (//5X,53HGLOBAL NUMBERS ASSOCIATED WITH SENSITIVITY OBJECTI
       1VES)
140     FORMAT (5E15.8)
150     FORMAT (////5X,26HNOMINAL DESIGN INFORMATION//5X,31HVALUES OF SENS
       1ITIVITY VARIABLES)
160     FORMAT (5X,5E13.5)
170     FORMAT (//5X,41HVALUES OF SENSITIVITY OBJECTIVE FUNCTIONS)
180     FORMAT (////5X,28HSENSITIVITY ANALYSIS RESULTS)
190     FORMAT (//5X,15HGLOBAL VARIABLE,I5//10X,1HX,20X,4HF(X))
200     FORMAT (/5X,35HTHE NOMINAL VALUE IS THE ONLY VALUE/5X,27HSPECIFIED
       1 FOR THIS VARIABLE)
210     FORMAT (/3X,E12.4,3X,4E13.4)
220     FORMAT (18X,4E13.4)
        END
```

```
      SUBROUTINE COPE06 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /COPES1/ ATITLE(20)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPOBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     *1,ISCR2
      DIMENSION ARRAY(NARRAY),RA(NDRA),IA(NDIA)
C     ******************************************************************
C     ROUTINE TO CALCULATE FUNCTIONS OF TWO DESIGN VARIABLES FOR ALL
C     COMBINATIONS OF A SET OF PRESCRIBED VALUES OF THESE VARIABLES.
C     ******************************************************************
C     WRITE OUTPUT INFORMATION ON SCRATCH UNIT ISCR1.
C     BY G. N. VANDERPLAATS                           AUG., 1974.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
      REWIND ISCR1
C     ------------------------------------------------------------------
C                                UNIT ISCR1 WRITE
C     ------------------------------------------------------------------
      WRITE (ISCR1,160) (ATITLE(I),I=1,20)
      WRITE (ISCR1,170) NCALC,N2VAR,M2VX,N2VX,M2VY,N2VY
C     N2VZ.
      M8=LOCI(8)
      M9=LOCI(9)-1
      WRITE (ISCR1,170) (IA(I),I=M8,M9)
C     ------------------------------------------------------------------
C                             TWO-VARIABLE FUNCTION SPACE
C     ------------------------------------------------------------------
      ICALC=2
      KCALC=3
      ISIGN=1
      N8=LOCR(8)
      N9=LOCR(9)-1
      DO 150 I=1,M2VX
      ARRAY(N2VX)=RA(N8)
      DO 140 J=1,M2VY
      N9=N9+ISIGN
      ARRAY(N2VY)=RA(N9)
C     ANALIZE.
110   CALL ANALIZ(ICALC)
120   CONTINUE
      IF(IP2VAR.GT.0) CALL ANALIZ(KCALC)
C     ------------------------------------------------------------------
C                                   UNIT ISCR1 WRITE
C     ------------------------------------------------------------------
C     WRITE X, Y.
      WRITE (ISCR1,180) RA(N8),RA(N9)
C     F(X,Y) VALUES.
      N10=LOCR(10)
```

```
      N11=N10
      M8=LOCI(8)
      DO 130 K=1,N2VAR
      N=IA(M8)
      RA(N11)=ARRAY(N)
      N11=N11+1
      M8=M8+1
130   CONTINUE
      N11=N10+N2VAR-1
      WRITE(ISCR1,180)(RA(K),K=N10,N11)
140   CONTINUE
      N9=N9+ISIGN
      N8=N8+1
      ISIGN=-ISIGN
150   CONTINUE
      RETURN
C     ------------------------------------------------------------------
C                                FORMATS
C     ------------------------------------------------------------------
160   FORMAT (20A4)
170   FORMAT (16I5)
180   FORMAT (5F15.8)
      END
```

```
      SUBROUTINE COPE07 (RA,IA,NDRA,NDIA)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /COPES1/ ATITLE(20)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     *1,ISCR2
      DIMENSION RA(NDRA),IA(NDIA)
C     ******************************************************************
C     ROUTINE TO PRINT TWO VARIABLE FUNCTION SPACE INFORMATION STORED ON
C     UNIT ISCR1.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                          AUG., 1974.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
      REWIND ISCR1
C     ------------------------------------------------------------------
C                           GENERAL INFORMATION
C     ------------------------------------------------------------------
C     TITLE.
      READ (ISCR1,80) (ATITLE(I),I=1,20)
      READ (ISCR1,90) NCALC,N2VAR,M2VX,N2VX,M2VY,N2VY
      IF(NCALC.NE.4) RETURN
C     N2VZ(I),I=1,N2VAR.
      READ (ISCR1,90) (IA(I),I=1,N2VAR)
      WRITE(6,50)
      WRITE (6,40) (ATITLE(I),I=1,20)
C     N2VX, N2VY.
      WRITE (6,140) N2VX,N2VY
C     N2VZ.
      WRITE (6,150)
      WRITE (6,100) (IA(I),I=1,N2VAR)
C     ------------------------------------------------------------------
C                 TWO-VARIABLE FUNCTION SPACE INFORMATION
C     ------------------------------------------------------------------
      DO 30 I=1,M2VX
      WRITE (6,160)
      DO 30 J=1,M2VY
C     X, Y.
      READ (ISCR1,170) XX,YY
C     F(X,Y).
      N10=LOCR(10)
      N11=N10+N2VAR-1
      READ(ISCR1,170)(RA(K),K=N10,N11)
      N=4
      IF (N2VAR.LT.4) N=N2VAR
      N11=N10+N-1
      IF(J.EQ.1) WRITE(6,120)XX,YY,(RA(K),K=N10,N11)
      IF(J.GT.1) WRITE(6,110)YY,(RA(K),K=N10,N11)
      IF (N.LE.N2VAR) GO TO 20
```

```
      M=(N2VAR-1)/4
      N=N10+4
      DO 10 K=1,M
      L=N+3
      IF(L.GT.N11) L=N11
      WRITE (6,130) (RA(KK),KK=N,L)
10    N=L+1
20    CONTINUE
30    CONTINUE
      RETURN
C     ---------------------------------------------------------------
C                                FORMATS
C     ---------------------------------------------------------------
40    FORMAT (//5X,5HTITLE/5X,20A4)
50    FORMAT (1H1,4X,35HTWO-VARIABLE FUNCTION SPACE RESULTS)
80    FORMAT (20A4)
90    FORMAT (16I5)
100   FORMAT (5X,10I5)
110   FORMAT (/15X,E12.4,3X,4E13.4)
120   FORMAT (/3X,2E12.4,3X,4E13.4)
130   FORMAT (32X,4E13.4)
140   FORMAT (///5X,48HGLOBAL NUMBER ASSOCIATED WITH X-VARIABLE, N2VX =
     1,I5//5X,48HGLOBAL NUMBER ASSOCIATED WITH Y-VARIABLE, N2VY =,I5)
150   FORMAT (//5X,37HGLOBAL NUMBERS ASSOCIATED WITH F(X,Y))
160   FORMAT (//10X,1HX,11X,1HY,20X,6HF(X,Y))
170   FORMAT (5F15.8)
      END
```

SUBROUTINE CONMIN - CONSTRAINED FUNCTION MINIMIZATION   SEPT. 77

```
      SUBROUTINE CONMIN (X,DF,G,ISC,IC,A,S,G1,G2,C,MS1,B,VLB,VUB,SCAL,N1
     1,N2,N3,N4,N5)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /CONSAV/DM1,DM2,DM3,DM4,DM5,DM6,DM7,DM8,DM9,DM10,DCT,DCTL,P
     1HI1,ABOBJ,ABOBJ1,ALPHAX,CTA,CTAM,CTBM,OBJ1,SLOPE,DX,DX1,FI,XI,DFTD
     2F1,ALP,FFF,D1(21),RSPACE,IDM1,IDM2,IDM3,JDIR,
     4IOBJ,KOBJ,KCOUNT,NCAL(2),NFEAS,MSCAL,NCOBJ,NVC,ID1(7)
     *,III,NLNC,JGOTO,ISPACE(2)
      DIMENSION X(N1),DF(N1),G(N2),ISC(N2),IC(N3),A(N3,N1),S(N1),G1(N2),
     1G2(N2),C(N4),MS1(N5),B(N3,N3),VLB(N1),VUB(N1),SCAL(N1)
C     ROUTINE TO SOLVE CONSTRAINED OR UNCONSTRAINED FUNCTION
C     MINIMIZATION.
C     BY G. N. VANDERPLAATS                          APRIL, 1972.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     REFERENCE;  CONMIN - A FORTRAN PROGRAM FOR CONSTRAINED FUNCTION
C         MINIMIZATION:  USER'S MANUAL,  BY G. N. VANDERPLAATS,
C         NASA TM X-62,282, AUGUST, 1973.
C     STORAGE REQUIREMENTS:
C         PROGRAM - 7000 DECIMAL WORDS (CDC COMPUTER)
C         ARRAYS  - APPROX. 2*(NDV**2)+26*NDV+4*NCON,
C               WHERE N3 = NDV+2.
C     RE-SCALE VARIABLES IF REQUIRED.
      IF (NSCAL.EQ.0.OR.IGOTO.EQ.0) GO TO 20
      DO 10 I=1,NDV
10    X(I)=C(I)
20    CONTINUE
C     CONSTANTS.
      NDV1=NDV+1
      NDV2=NDV+2
      IF (IGOTO.EQ.0) GO TO 30
      GO TO (150,370,360,650,670), IGOTO
C     ------------------------------------------------------------------
C                          SAVE INPUT CONTROL PARAMETERS
C     ------------------------------------------------------------------
30    CONTINUE
      IF (IPRINT.GT.0) WRITE (6,1230)
      IF (LINOBJ.EQ.0.OR.(NCON.GT.0.OR.NSIDE.GT.0)) GO TO 40
C     TOTALLY UNCONSTRAINED FUNCTION WITH LINEAR OBJECTIVE.
C     SOLUTION IS UNBOUNDED.
      WRITE (6,970) LINOBJ,NCON,NSIDE
      RETURN
40    CONTINUE
      IDM1=ITRM
      IDM2=ITMAX
      IDM3=ICNDIR
      DM1=DELFUN
      DM2=DABFUN
      DM3=CT
```

```
      DM4=CTMIN
      DM5=CTL
      DM6=CTLMIN
      DM7=THETA
      DM8=PHI
      DM9=FDCH
      DM10=FDCHM
C     ------------------------------------------------------------------
C                               DEFAULTS
C     ------------------------------------------------------------------
      IF (ITRM.LE.0) ITRM=3
      IF (ITMAX.LE.0) ITMAX=20
      NDV1=NDV+1
      IF (ICNDIR.EQ.0) ICNDIR=NDV1
      IF (DELFUN.LE.0.) DELFUN=.0001
      CT=-ABS(CT)
      IF (CT.GE.0.) CT=-.1
      CTMIN=ABS(CTMIN)
      IF (CTMIN.LE.0.) CTMIN=.004
      CTL=-ABS(CTL)
      IF (CTL.GE.0.) CTL=-0.01
      CTLMIN=ABS(CTLMIN)
      IF (CTLMIN.LE.0.) CTLMIN=.001
      IF (THETA.LE.0.) THETA=1.
      IF (PHI.LE.0.) PHI=5.
      IF (FDCH.LE.0.) FDCH=.01
      IF (FDCHM.LE.0.) FDCHM=.01
C     ------------------------------------------------------------------
C                     INITIALIZE INTERNAL PARAMETERS
C     ------------------------------------------------------------------
      INFOG=0
      ITER=0
      JDIR=0
      IOBJ=0
      KOBJ=0
      NDV2=NDV+2
      KCOUNT=0
      NCAL(1)=0
      NCAL(2)=0
      NAC=0
      NFEAS=0
      MSCAL=NSCAL
      CT1=ITRM
      CT1=1./CT1
      DCT=(CTMIN/ABS(CT))**CT1
      DCTL=(CTLMIN/ABS(CTL))**CT1
      PHI1=PHI
      ABOBJ=.1
      ABOBJ1=.1
      ALPHAX=.1
```

```
      NCOBJ=0
      CTAM=ABS(CTMIN)
      CTBM=ABS(CTLMIN)
C     CALCULATE NUMBER OF LINEAR CONSTRAINTS, NLNC.
      NLNC=0
      IF (NCON.EQ.0) GO TO 60
      DO 50 I=1,NCON
      IF (ISC(I).GT.0) NLNC=NLNC+1
50    CONTINUE
60    CONTINUE
C     ------------------------------------------------------------------
C          CHECK TO BE SURE THAT SIDE CONSTRAINTS ARE SATISFIED
C     ------------------------------------------------------------------
      IF (NSIDE.EQ.0) GO TO 100
      DO 90 I=1,NDV
      IF (VLB(I).LE.VUB(I)) GO TO 70
      XX=.5*(VLB(I)+VUB(I))
      X(I)=XX
      VLB(I)=XX
      VUB(I)=XX
      WRITE (6,1120) I
70    CONTINUE
      XX=X(I)-VLB(I)
      IF (XX.GE.0.) GO TO 80
C     LOWER BOUND VIOLATED.
      WRITE (6,1130) X(I),VLB(I),I
      X(I)=VLB(I)
      GO TO 90
80    CONTINUE
      XX=VUB(I)-X(I)
      IF (XX.GE.0.) GO TO 90
      WRITE (6,1140) X(I),VUB(I),I
      X(I)=VUB(I)
90    CONTINUE
100   CONTINUE
C     ------------------------------------------------------------------
C                     INITIALIZE SCALING VECTOR, SCAL
C     ------------------------------------------------------------------
      IF (NSCAL.EQ.0) GO TO 140
      IF (NSCAL.LT.0) GO TO 120
      DO 110 I=1,NDV
110   SCAL(I)=1.
      GO TO 140
120   CONTINUE
      DO 130 I=1,NDV
      SI=ABS(SCAL(I))
      IF (SI.LT.1.0E-20) SI=1.0E-5
      SCAL(I)=SI
      SI=1./SI
      X(I)=X(I)*SI
```

```
      IF (NSIDE.EQ.0) GO TO 130
      VLB(I)=VLB(I)*SI
      VUB(I)=VUB(I)*SI
130   CONTINUE
140   CONTINUE
C     ------------------------------------------------------------------
C     ***** CALCULATE INITIAL FUNCTION AND CONSTRAINT VALUES  *****
C     ------------------------------------------------------------------
      INFO=1
      NCAL(1)=1
      IGOTO=1
      GO TO 950
150   CONTINUE
      OBJ1=OBJ
      IF (DABFUN.LE.0.) DABFUN=.001*ABS(OBJ)
      IF (DABFUN.LT.1.0E-10) DABFUN=1.0E-10
      IF (IPRINT.LE.0) GO TO 260
C     ------------------------------------------------------------------
C                      PRINT INITIAL DESIGN INFORMATION
C     ------------------------------------------------------------------
      IF (IPRINT.LE.1) GO TO 220
      IF (NSIDE.EQ.0.AND.NCON.EQ.0) WRITE (6,1300)
      IF (NSIDE.NE.0.OR.NCON.GT.0) WRITE (6,1240)
      WRITE (6,1250) IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,LINOB
     1J,ITRM,N1,N2,N3,N4,N5
      WRITE (6,1270) CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,DABFUN
      WRITE (6,1260) FDCH,FDCHM
      IF (NSIDE.EQ.0) GO TO 180
      WRITE (6,1280)
      DO 160 I=1,NDV,6
      M1=MIN0(NDV,I+5)
160   WRITE (6,1010) I,(VLB(J),J=I,M1)
      WRITE (6,1290)
      DO 170 I=1,NDV,6
      M1=MIN0(NDV,I+5)
170   WRITE (6,1010) I,(VUB(J),J=I,M1)
180   CONTINUE
      IF (NSCAL.GE.0) GO TO 190
      WRITE (6,1310)
      WRITE (6,1470) (SCAL(I),I=1,NDV)
190   CONTINUE
      IF (NCON.EQ.0) GO TO 220
      IF (NLNC.EQ.0.OR.NLNC.EQ.NCON) GO TO 210
      WRITE (6,1020)
      DO 200 I=1,NCON,15
      M1=MIN0(NCON,I+14)
200   WRITE (6,1030) I,(ISC(J),J=I,M1)
      GO TO 220
210   IF (NLNC.EQ.NCON) WRITE (6,1040)
      IF (NLNC.EQ.0) WRITE (6,1050)
```

```
220   CONTINUE
      WRITE (6,1450) OBJ
      WRITE (6,1460)
      DO 230 I=1,NDV
      X1=1.
      IF (NSCAL.NE.0) X1=SCAL(I)
230   G1(I)=X(I)*X1
      DO 240 I=1,NDV,6
      M1=MIN0(NDV,I+5)
240   WRITE (6,1010) I,(G1(J),J=I,M1)
      IF (NCON.EQ.0) GO TO 260
      WRITE (6,1480)
      DO 250 I=1,NCON,6
      M1=MIN0(NCON,I+5)
250   WRITE (6,1010) I,(G(J),J=I,M1)
260   CONTINUE
      IF (IPRINT.GT.1) WRITE (6,1370)
C     ------------------------------------------------------------------
C     *********************  BEGIN MINIMIZATION  ************************
C     ------------------------------------------------------------------
270   CONTINUE
      ITER=ITER+1
      IF (ABOBJ1.LT..0001) ABOBJ1=.0001
      IF (ABOBJ1.GT..2) ABOBJ1=.2
      IF (ALPHAX.GT.1.) ALPHAX=1.
      IF (ALPHAX.LT..01) ALPHAX=.01
      IF (IPRINT.GT.2) WRITE (6,1320) ITER
      NFEAS=NFEAS+1
      IF (NFEAS.GT.10) GO TO 790
      IF (IPRINT.GT.3.AND.NCON.GT.0) WRITE (6,1330) CT,CTL,PHI
      CTA=ABS(CT)
      IF (NCOBJ.EQ.0) GO TO 310
C     ------------------------------------------------------------------
C     NO MOVE ON LAST ITERATION.  DELETE CONSTRAINTS THAT ARE NO
C     LONGER ACTIVE.
C     ------------------------------------------------------------------
      NNAC=NAC
      DO 300 I=1,NNAC
      NIC=IC(I)
      IF (NIC.GT.NCON) NAC=NAC-1
      IF (NIC.GT.NCON) GO TO 300
      CT1=CT
      IF (ISC(NIC).GT.0) CT1=CTL
      IF (G(NIC).GT.CT1) GO TO 300
      NAC=NAC-1
      IF (I.EQ.NNAC) GO TO 300
      IP1=I+1
      DO 290 K=IP1,NNAC
      II=K-1
      DO 280 J=1,NDV2
```

```
280   A(II,J)=A(K,J)
290   IC(II)=IC(K)
300   CONTINUE
      GO TO 400
310   CONTINUE
      IF (MSCAL.LT.NSCAL.OR.NSCAL.EQ.0) GO TO 330
      IF (NSCAL.LT.0.AND.KCOUNT.LT.ICNDIR) GO TO 330
      MSCAL=0
      KCOUNT=0
C     ------------------------------------------------------------------
C                          SCALE VARIABLES
C     ------------------------------------------------------------------
      DO 320 I=1,NDV
      SI=SCAL(I)
      XI=SI*X(I)
      SIB=SI
      IF (NSCAL.GT.0) SI=ABS(XI)
      IF (SI.LT.1.0E-10) GO TO 320
      SCAL(I)=SI
      SI=1./SI
      X(I)=XI*SI
      IF (NSIDE.EQ.0) GO TO 320
      VLB(I)=SIB*SI*VLB(I)
      VUB(I)=SIB*SI*VUB(I)
320   CONTINUE
      IF (IPRINT.LT.4.OR.(NSCAL.LT.0.AND.ITER.GT.1)) GO TO 330
      WRITE (6,1340)
      WRITE (6,1470) (SCAL(I),I=1,NDV)
330   CONTINUE
      MSCAL=MSCAL+1
      NAC=0
C     ------------------------------------------------------------------
C          OBTAIN GRADIENTS OF OBJECTIVE AND ACTIVE CONSTRAINTS
C     ------------------------------------------------------------------
      INFO=2
      NCAL(2)=NCAL(2)+1
      IF (NFDG.LT.2) GO TO 350
      IGOTO=2
      GO TO 950
350   CONTINUE
      JGOTO=0
360   CONTINUE
      CALL CNMN01 (JGOTO,X,DF,G,ISC,IC,A,G1,VLB,VUB,SCAL,C,NCAL,DX,DX1,F
     1I,XI,III,N1,N2,N3,N4)
      IGOTO=3
      IF (JGOTO.GT.0) GO TO 950
370   CONTINUE
      INFO=1
      IF (NAC.GE.N3) GO TO 790
      IF (NSCAL.EQ.0.OR.NFDG.EQ.0) GO TO 400
```

```
C     ----------------------------------------------------------------
C                               SCALE GRADIENTS
C     ----------------------------------------------------------------
C     SCALE GRADIENT OF OBJECTIVE FUNCTION.
      DO 380 I=1,NDV
380   DF(I)=DF(I)*SCAL(I)
      IF (NFDG.EQ.1.OR.NAC.EQ.0) GO TO 400
C     SCALE GRADIENTS OF ACTIVE CONSTRAINTS.
      DO 390 J=1,NDV
      SCJ=SCAL(J)
      DO 390 I=1,NAC
390   A(I,J)=A(I,J)*SCJ
400   CONTINUE
      IF (IPRINT.LT.3.OR.NCON.EQ.0) GO TO 450
C     ----------------------------------------------------------------
C                                    PRINT
C     ----------------------------------------------------------------
C     PRINT ACTIVE AND VIOLATED CONSTRAINT NUMBERS.
      M1=0
      M2=N3
      IF (NAC.EQ.0) GO TO 430
      DO 420 I=1,NAC
      J=IC(I)
      IF (J.GT.NCON) GO TO 420
      GI=G(J)
      C1=CTAM
      IF (ISC(J).GT.0) C1=CTBM
      GI=GI-C1
      IF (GI.GT.0.) GO TO 410
C     ACTIVE CONSTRAINT.
      M1=M1+1
      MS1(M1)=J
      GO TO 420
410   M2=M2+1
C     VIOLATED CONSTRAINT.
      MS1(M2)=J
420   CONTINUE
430   M3=M2-N3
      WRITE (6,1060) M1
      IF (M1.EQ.0) GO TO 440
      WRITE (6,1070)
      WRITE (6,1490) (MS1(I),I=1,M1)
440   WRITE (6,1080) M3
      IF (M3.EQ.0) GO TO 450
      WRITE (6,1070)
      M3=N3+1
      WRITE (6,1490) (MS1(I),I=M3,M2)
450   CONTINUE
C     ----------------------------------------------------------------
C             CALCULATE GRADIENTS OF ACTIVE SIDE CONSTRAINTS
```

```
C     ------------------------------------------------------------------
      IF (NSIDE.EQ.0) GO TO 510
      MCN1=NCON
      M1=0
      DO 490 I=1,NDV
C     LOWER BOUND.
      XI=X(I)
      XID=VLB(I)
      X12=ABS(XID)
      IF (X12.LT.1.) X12=1.
      GI=(XID-XI)/X12
      IF (GI.LT.-1.0E-6) GO TO 470
      M1=M1+1
      MS1(M1)=-I
      NAC=NAC+1
      IF (NAC.GE.N3) GO TO 790
      MCN1=MCN1+1
      DO 460 J=1,NDV
460   A(NAC,J)=0.
      A(NAC,I)=-1.
      IC(NAC)=MCN1
      G(MCN1)=GI
      ISC(MCN1)=1
C     UPPER BOUND.
470   XID=VUB(I)
      X12=ABS(XID)
      IF (X12.LT.1.) X12=1.
      GI=(XI-XID)/X12
      IF (GI.LT.-1.0E-6) GO TO 490
      M1=M1+1
      MS1(M1)=I
      NAC=NAC+1
      IF (NAC.GE.N3) GO TO 790
      MCN1=MCN1+1
      DO 480 J=1,NDV
480   A(NAC,J)=0.
      A(NAC,I)=1.
      IC(NAC)=MCN1
      G(MCN1)=GI
      ISC(MCN1)=1
490   CONTINUE
C     ------------------------------------------------------------------
C                                   PRINT
C     ------------------------------------------------------------------
C     PRINT ACTIVE SIDE CONSTRAINT NUMBERS.
      IF (IPRINT.LT.3) GO TO 510
      WRITE (6,1090) M1
      IF (M1.EQ.0) GO TO 510
      WRITE (6,1100)
      DO 500 I=1,M1,15
```

```
      M2=MIN0(M1,I+14)
500   WRITE (6,1490) (MS1(J),J=1,M2)
510   CONTINUE
C     PRINT GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS.
      IF (IPRINT.LT.4) GO TO 550
      WRITE (6,1350)
      DO 520 I=1,NDV,6
      M1=MIN0(NDV,I+5)
520   WRITE (6,1010) I,(DF(J),J=I,M1)
      IF (NAC.EQ.0) GO TO 550
      WRITE (6,1360)
      DO 540 I=1,NAC
      M1=IC(I)
      M2=M1-NCON
      M3=0
      IF (M2.GT.0) M3=IABS(MS1(M2))
      IF (M2.LE.0) WRITE (6,990) M1
      IF (M2.GT.0) WRITE (6,1000) M3
      DO 530 K=1,NDV,6
      M1=MIN0(NDV,K+5)
530   WRITE (6,1010) K,(A(I,J),J=K,M1)
540   WRITE (6,1370)
550   CONTINUE
C     ------------------------------------------------------------------
C     ******************* DETERMINE SEARCH DIRECTION *******************
C     ------------------------------------------------------------------
      ALP=1.0E+20
      IF (NAC.GT.0) GO TO 560
C     ------------------------------------------------------------------
C                       UNCONSTRAINED FUNCTION
C     ------------------------------------------------------------------
C     FIND DIRECTION OF STEEPEST DESCENT OR CONJUGATE DIRECTION.
      NVC=0
      NFEAS=0
      KCOUNT=KCOUNT+1
C     IF KCOUNT.GT.ICNDIR  RESTART CONJUGATE DIRECTION ALGORITHM.
      IF (KCOUNT.GT.ICNDIR.OR.IOBJ.EQ.2) KCOUNT=1
      IF (KCOUNT.EQ.1) JDIR=0
C     IF JDIR = 0 FIND DIRECTION OF STEEPEST DESCENT.
      CALL CNMN02 (JDIR,SLOPE,DFTDF1,DF,S,N1)
      GO TO 610
560   CONTINUE
C     ------------------------------------------------------------------
C                        CONSTRAINED FUNCTION
C     ------------------------------------------------------------------
C     FIND USABLE-FEASIBLE DIRECTION.
      KCOUNT=0
      JDIR=0
      PHI=10.*PHI
      IF (PHI.GT.1000.) PHI=1000.
```

```
      IF (NFEAS.EQ.1) PHI=PHI1
C     CALCULATE DIRECTION, S.
      CALL CNMN05 (NVC,SLOPE,DF,G,ISC,IC,A,S,C,MS1,B,N1,N2,N3,N4,N5)
C     IF THIS DESIGN IS FEASIBLE AND LAST ITERATION WAS INFEASIBLE,
C     SET ABOBJ1=.05 (5 PERCENT).
      IF (NVC.EQ.0.AND.NFEAS.GT.1) ABOBJ1=.05
      IF (NVC.EQ.0) NFEAS=0
      IF (IPRINT.LT.3) GO TO 580
      WRITE (6,1380)
      DO 570 I=1,NAC,6
      M1=MIN0(NAC,I+5)
570   WRITE (6,1010) I,(A(J,NDV1),J=I,M1)
      WRITE (6,1220) S(NDV1)
580   CONTINUE
C     ------------------------------------------------------------------
C     ******************* ONE-DIMENSIONAL SEARCH ************************
C     ------------------------------------------------------------------
      IF (S(NDV1).LT.1.0E-6.AND.NVC.EQ.0) GO TO 690
C     ------------------------------------------------------------------
C                 FIND ALPHA TO OBTAIN A FEASIBLE DESIGN
C     ------------------------------------------------------------------
      IF (NVC.EQ.0) GO TO 610
      ALP=-1.
      DO 600 I=1,NAC
      NCI=IC(I)
      C1=G(NCI)
      CTC=CTAM
      IF (ISC(NCI).GT.0) CTC=CTBM
      IF (C1.LE.CTC) GO TO 600
      ALP1=0.
      DO 590 J=1,NDV
590   ALP1=ALP1+S(J)*A(I,J)
      ALP1=ALP1*A(I,NDV2)
      IF (ABS(ALP1).LT.1.0E-20) GO TO 600
      ALP1=-C1/ALP1
      IF (ALP1.GT.ALP) ALP=ALP1
600   CONTINUE
610   CONTINUE
C     ------------------------------------------------------------------
C                       LIMIT CHANCE TO ABOBJ1*OBJ
C     ------------------------------------------------------------------
      ALP1=1.0E+20
      SI=ABS(OBJ)
      IF (SI.LT..01) SI=.01
      IF (ABS(SLOPE).GT.1.0E-20) ALP1=ABOBJ1*SI/SLOPE
      ALP1=ABS(ALP1)
      IF (NVC.GT.0) ALP1=10.*ALP1
      IF (ALP1.LT.ALP) ALP=ALP1
C     ------------------------------------------------------------------
C                    LIMIT CHANGE IN VARIABLE TO ALPHAX
```

```
C     ---------------------------------------------------------------------
      ALP11=1.0E+20
      DO 620 I=1,NDV
      SI=ABS(S(I))
      XI=ABS(X(I))
      IF (SI.LT.1.0E-10.OR.XI.LT.0.1) GO TO 620
      ALP1=ALPHAX*XI/SI
      IF (ALP1.LT.ALP11) ALP11=ALP1
620   CONTINUE
      IF (NVC.GT.0) ALP11=10.*ALP11
      IF (ALP11.LT.ALP) ALP=ALP11
      IF (ALP.GT.1.0E+20) ALP=1.0E+20
      IF (ALP.LE.1.0E-20) ALP=1.0E-20
      IF (IPRINT.LT.3) GO TO 640
      WRITE (6,1390)
      DO 630 I=1,NDV,6
      M1=MIN0(NDV,I+5)
630   WRITE (6,1010) I,(S(J),J=I,M1)
      WRITE (6,1110) SLOPE,ALP
640   CONTINUE
      IF (NCON.GT.0.OR.NSIDE.GT.0) GO TO 660
C     ---------------------------------------------------------------------
C            DO ONE-DIMENSIONAL SEARCH FOR UNCONSTRAINED FUNCTION
C     ---------------------------------------------------------------------
      JGOTO=0
650   CONTINUE
      CALL CNMN03 (X,S,DF,G,A,IC,SCAL,C,N1,N2,N3,N4)
      IGOTO=4
      IF (JGOTO.GT.0) GO TO 950
      JDIR=1
C     PROCEED TO CONVERGENCE CHECK.
      GO TO 680
C     ---------------------------------------------------------------------
C       SOLVE ONE-DIMENSIONAL SEARCH PROBLEM FOR CONSTRAINED FUNCTION
C     ---------------------------------------------------------------------
660   CONTINUE
      JGOTO=0
670   CONTINUE
      CALL CNMN06 (X,DF,G,ISC,S,G1,G2,VLB,VUB,SCAL,N1,N2)
      IGOTO=5
      IF (JGOTO.GT.0) GO TO 950
      IF (NAC.EQ.0) JDIR=1
C     ---------------------------------------------------------------------
C     ********************     UPDATE ALPHAX    **************************
C     ---------------------------------------------------------------------
680   CONTINUE
690   CONTINUE
      IF (ALP.GT.1.0E+19) ALP=0.
C     UPDATE ALPHAX TO BE AVERAGE OF MAXIMUM CHANGE IN X(I)
C     AND ALHPAX.
```

```
      ALP11=0.
      DO 700 I=1,NDV
      SI=ABS(S(I))
      XI=ABS(X(I))
      IF (XI.LT.1.0E-10) GO TO 700
      ALP1=ALP*SI/XI
      IF (ALP1.GT.ALP11) ALP11=ALP1
700   CONTINUE
      ALP11=.5*(ALP11+ALPHAX)
      ALP12=5.*ALPHAX
      IF (ALP11.GT.ALP12) ALP11=ALP12
      ALPHAX=ALP11
      NCOBJ=NCOBJ+1
C     ABSOLUTE CHANGE IN OBJECTIVE.
      OBJD=OBJ1-OBJ
      OBJB=ABS(OBJD)
      IF (OBJB.LT.1.0E-10) OBJB=0.
      IF (NAC.EQ.0.OR.OBJB.GT.0.) NCOBJ=0
      IF (NCOBJ.GT.1) NCOBJ=0
C     ------------------------------------------------------------------
C                                  PRINT
C     ------------------------------------------------------------------
C     PRINT MOVE PARAMETER, NEW X-VECTOR AND CONSTRAINTS.
      IF (IPRINT.LT.3) GO TO 710
      WRITE (6,1400) ALP
710   IF (IPRINT.LT.2) GO TO 780
      IF (OBJB.GT.0.) GO TO 720
      IF (IPRINT.EQ.2) WRITE (6,1410) ITER,OBJ
      IF (IPRINT.GT.2) WRITE (6,1420) OBJ
      GO TO 740
720   IF (IPRINT.EQ.2) GO TO 730
      WRITE (6,1430) OBJ
      GO TO 740
730   WRITE (6,1440) ITER,OBJ
740   WRITE (6,1460)
      DO 750 I=1,NDV
      FF1=1.
      IF (NSCAL.NE.0) FF1=SCAL(I)
750   G1(I)=FF1*X(I)
      DO 760 I=1,NDV,6
      M1=MIN0(NDV,I+5)
760   WRITE (6,1010) I,(G1(J),J=I,M1)
      IF (NCON.EQ.0) GO TO 780
      WRITE (6,1480)
      DO 770 I=1,NCON,6
      M1=MIN0(NCON,I+5)
770   WRITE (6,1010) I,(G(J),J=I,M1)
780   CONTINUE
C     ------------------------------------------------------------------
C                           CHECK CONVERGENCE
```

```
C     ------------------------------------------------------------------
C     STOP IF ITER EQUALS ITMAX.
      IF (ITER.GE.ITMAX) GO TO 790
C     ------------------------------------------------------------------
C                     ABSOLUTE CHANGE IN OBJECTIVE
C     ------------------------------------------------------------------
      OBJB=ABS(OBJD)
      KOBJ=KOBJ+1
      IF (OBJB.GE.DABFUN.OR.NFEAS.GT.0) KOBJ=0
C     ------------------------------------------------------------------
C                     RELATIVE CHANGE IN OBJECTIVE
C     ------------------------------------------------------------------
      IF (ABS(OBJ1).GT.1.0E-10) OBJD=OBJD/ABS(OBJ1)
      ABOBJ1=.5*(ABS(ABOBJ)+ABS(OBJD))
      ABOBJ=ABS(OBJD)
      IOBJ=IOBJ+1
      IF (NVC.GT.0.OR.OBJD.GE.DELFUN) IOBJ=0
      IF (IOBJ.GE.ITRM.OR.KOBJ.GE.ITRM) GO TO 790
      OBJ1=OBJ
C     ------------------------------------------------------------------
C            REDUCE CT IF OBJECTIVE FUNCTION IS CHANGING SLOWLY
C     ------------------------------------------------------------------
      IF (IOBJ.LT.1.OR.NAC.EQ.0) GO TO 270
      CT=DCT*CT
      CTL=CTL*DCTL
      IF(ABS(CT).LT.CTMIN) CT=-CTMIN
      IF(ABS(CTL).LT.CTLMIN) CTL=-CTLMIN
C     ------------------------------------------------------------------
C                     CHECK FOR UNBOUNDED SOLUTION
C     ------------------------------------------------------------------
C     STOP IF OBJ IS LESS THAN -1.0E+40.
      IF (OBJ.GT.-1.0E+40) GO TO 270
      WRITE (6,980)
790   CONTINUE
      IF (NAC.GE.N3) WRITE (6,1500)
C     ------------------------------------------------------------------
C     ****************  FINAL FUNCTION INFORMATION  ********************
C     ------------------------------------------------------------------
      IF (NSCAL.EQ.0) GO TO 820
C     UN-SCALE THE DESIGN VARIABLES.
      DO 810 I=1,NDV
      XI=SCAL(I)
      IF (NSIDE.EQ.0) GO TO 810
      VLB(I)=XI*VLB(I)
      VUB(I)=XI*VUB(I)
810   X(I)=XI*X(I)
C     ------------------------------------------------------------------
C                           PRINT FINAL RESULTS
C     ------------------------------------------------------------------
820   IF (IPRINT.EQ.0.OR.NAC.GE.N3) GO TO 940
```

```
      WRITE (6,1510)
      WRITE (6,1430) OBJ
      WRITE (6,1460)
      DO 830 I=1,NDV,6
      M1=MIN0(NDV,I+5)
830   WRITE (6,1010) I,(X(J),J=I,M1)
      IF (NCON.EQ.0) GO TO 890
      WRITE (6,1480)
      DO 840 I=1,NCON,6
      M1=MIN0(NCON,I+5)
840   WRITE (6,1010) I,(G(J),J=I,M1)
C     DETERMINE WHICH CONSTRAINTS ARE ACTIVE AND PRINT.
      NAC=0
      NVC=0
      DO 860 I=1,NCON
      CTA=CTAM
      IF (ISC(I).GT.0) CTA=CTBM
      GI=G(I)
      IF (GI.GT.CTA) GO TO 850
      IF (GI.LT.CT.AND.ISC(I).EQ.0) GO TO 860
      IF (GI.LT.CTL.AND.ISC(I).GT.0) GO TO 860
      NAC=NAC+1
      IC(NAC)=I
      GO TO 860
850   NVC=NVC+1
      MS1(NVC)=I
860   CONTINUE
      WRITE (6,1060) NAC
      IF (NAC.EQ.0) GO TO 870
      WRITE (6,1070)
      WRITE (6,1490) (IC(J),J=1,NAC)
870   WRITE (6,1080) NVC
      IF (NVC.EQ.0) GO TO 880
      WRITE (6,1070)
      WRITE (6,1490) (MS1(J),J=1,NVC)
880   CONTINUE
890   CONTINUE
      IF (NSIDE.EQ.0) GO TO 920
C     DETERMINE WHICH SIDE CONSTRAINTS ARE ACTIVE AND PRINT.
      NAC=0
      DO 910 I=1,NDV
      XI=X(I)
      XID=VLB(I)
      X12=ABS(XID)
      IF (X12.LT.1.) X12=1.
      GI=(XID-XI)/X12
      IF (GI.LT.-1.0E-6) GO TO 900
      NAC=NAC+1
      MS1(NAC)=-I
900   XID=VUB(I)
```

```
      X12=ABS(XID)
      IF (X12.LT.1.) X12=1.
      GI=(XI-XID)/X12
      IF (GI.LT.-1.0E-6) GO TO 910
      NAC=NAC+1
      MS1(NAC)=I
910   CONTINUE
      WRITE (6,1090) NAC
      IF (NAC.EQ.0) GO TO 920
      WRITE (6,1100)
      WRITE (6,1490) (MS1(J),J=1,NAC)
920   CONTINUE
      WRITE (6,1150)
      IF (ITER.GE.ITMAX) WRITE (6,1160)
      IF (NFEAS.GE.10) WRITE (6,1170)
      IF (IOBJ.GE.ITRM) WRITE (6,1190) ITRM
      IF (KOBJ.GE.ITRM) WRITE (6,1200) ITRM
      WRITE (6,1210) ITER
      WRITE (6,1520) NCAL(1)
      IF (NCON.GT.0) WRITE (6,1530) NCAL(1)
      IF (NFDG.NE.0) WRITE (6,1540) NCAL(2)
      IF (NCON.GT.0.AND.NFDG.EQ.2) WRITE (6,1550) NCAL(2)
C     ------------------------------------------------------------------
C                  RE-SET BASIC PARAMETERS TO INPUT VALUES
C     ------------------------------------------------------------------
940   ITRM=IDM1
      ITMAX=IDM2
      ICNDIR=IDM3
      DELFUN=DM1
      DABFUN=DM2
      CT=DM3
      CTMIN=DM4
      CTL=DM5
      CTLMIN=DM6
      THETA=DM7
      PHI=DM8
      FDCH=DM9
      FDCHM=DM10
      IGOTO=0
950   CONTINUE
      IF (NSCAL.EQ.0.OR.IGOTO.EQ.0) RETURN
C     UN-SCALE VARIABLES.
      DO 960 I=1,NDV
      C(I)=X(I)
960   X(I)=X(I)*SCAL(I)
      RETURN
C     ------------------------------------------------------------------
C                                FORMATS
C     ------------------------------------------------------------------
C
```

```
970   FORMAT (///5X,72HA COMPLETELY UNCONSTRAINED FUNCTION WITH A LINEAR
     1 OBJECTIVE IS SPECIFIED//10X,8HLINOBJ =,I5/10X,8HNCON   =,I5/10X,8
     2HNSIDE  =,I5//5X,35HCONTROL RETURNED TO CALLING PROGRAM)
980   FORMAT (///5X,56HCONMIN HAS ACHIEVED A SOLUTION OF OBJ LESS THAN -
     11.0E+40/5X,32HSOLUTION APPEARS TO BE UNBOUNDED/5X,26HOPTIMIZATION
     2IS TERMINATED)
990   FORMAT (5X,17HCONSTRAINT NUMBER,I5)
1000  FORMAT (5X,27HSIDE CONSTRAINT ON VARIABLE,I5)
1010  FORMAT (3X,I5,1H),2X,6E13.5)
1020  FORMAT (/5X,35HLINEAR CONSTRAINT IDENTIFIERS (ISC)/5X,36HNON-ZERO
     1INDICATES LINEAR CONSTRAINT)
1030  FORMAT (3X,I5,1H),2X,15I5)
1040  FORMAT (/5X,26HALL CONSTRAINTS ARE LINEAR)
1050  FORMAT (/5X,30HALL CONSTRAINTS ARE NON-LINEAR)
1060  FORMAT (/5X,9HTHERE ARE,I5,19H ACTIVE CONSTRAINTS)
1070  FORMAT (5X,22HCONSTRAINT NUMBERS ARE)
1080  FORMAT (/5X,9HTHERE ARE,I5,21H VIOLATED CONSTRAINTS)
1090  FORMAT (/5X,9HTHERE ARE,I5,24H ACTIVE SIDE CONSTRAINTS)
1100  FORMAT (5X,43HDECISION VARIABLES AT LOWER OR UPPER BOUNDS,30H (MIN
     1US INDICATES LOWER BOUND))
1110  FORMAT (/5X,22HONE-DIMENSIONAL SEARCH/5X,15HINITIAL SLOPE =,E12.4,
     12X,16HPROPOSED ALPHA =,E12.4)
1120  FORMAT (///5X,35H* * CONMIN DETECTS VLB(I).GT.VUB(I)/5X,57HFIX IS
     1SET X(I)=VLB(I)=VUB(I) = .5*(VLB(I)+VUB(I) FOR I =,I5)
1130  FORMAT (///5X,41H* * CONMIN DETECTS INITIAL X(I).LT.VLB(I)/5X,6HX(
     1I) =,E12.4,2X,8HVLB(I) =,E12.4/5X,35HX(I) IS SET EQUAL TO VLB(I) F
     2OR I =,I5)
1140  FORMAT (///5X,41H* * CONMIN DETECTS INITIAL X(I).GT.VUB(I)/5X,6HX(
     1I) =,E12.4,2X,8HVUB(I) =,E12.4/5X,35HX(I) IS SET EQUAL TO VUB(I) F
     2OR I =,I5)
1150  FORMAT (/5X,21HTERMINATION CRITERION)
1160  FORMAT (10X,17HITER EQUALS ITMAX)
1170  FORMAT (10X,62HTEN CONSECUTIVE ITERATIONS FAILED TO PRODUCE A FEAS
     1IBLE DESIGN)
1190  FORMAT (10X,43HABS(1-OBJ(I-1)/OBJ(I)) LESS THAN DELFUN FOR,I3,11H
     1ITERATIONS)
1200  FORMAT (10X,43HABS(OBJ(I)-OBJ(I-1))   LESS THAN DABFUN FOR,I3,11H
     1ITERATIONS)
1210  FORMAT (/5X,22HNUMBER OF ITERATIONS =,I5)
1220  FORMAT (/5X,28HCONSTRAINT PARAMETER, BETA =,E14.5)
1230  FORMAT (1H1,////12X,27(2H* )/12X,1H*,51X,1H*/12X,1H*,20X,11HC O N
     1M I N,20X,1H*/12X,1H*,51X,1H*/12X,1H*,15X,21H FORTRAN PROGRAM FOR
     2,15X,1H*/12X,1H*,51X,1H*/12X,1H*,9X,33HCONSTRAINED FUNCTION MINIMI
     3ZATION,9X,1H*/12X,1H*,51X,1H*/12X,1H*,2X,48HNASA/AMES RESEARCH CEN
     4TER, MOFFETT FIELD, CALIF.,1X,1H*/12X,1H*,51X,1H*/12X,1H*,13X,25HV
     5ERSION II      JULY, 1975,13X,1H*/12X,1H*,51X,1H*/12X,27(2H* ))
1240  FORMAT (////5X,33HCONSTRAINED FUNCTION MINIMIZATION//5X,18HCONTROL
     1 PARAMETERS)
1250  FORMAT (/5X,60HIPRINT  NDV    ITMAX    NCON    NSIDE  ICNDIR   NSC
     1AL   NFDG/8I8//5X,12HLINOBJ  ITRM,5X,2HN1,6X,2HN2,6X,2HN3,6X,2HN4,
```

```
     26X,2HN5/8I8)
1260 FORMAT (/9X,4HFDCH,12X,5HFDCHM/3X,2E14.5)
1270 FORMAT (/9X,2HCT,14X,5HCTMIN,11X,3HCTL,13X,6HCTLMIN/1X,4(2X,E14.5)
     1//9X,5HTHETA,11X,3HPHI,13X,6HDELFUN,10X,6HDABFUN/1X,4(2X,E14.5))
1280 FORMAT (/5X,40HLOWER BOUNDS ON DECISION VARIABLES (VLB))
1290 FORMAT (/5X,40HUPPER BOUNDS ON DECISION VARIABLES (VUB))
1300 FORMAT (////5X,35HUNCONSTRAINED FUNCTION MINIMIZATION//5X,18HCONTR
     1OL PARAMETERS)
1310 FORMAT (/5X,21HSCALING VECTOR (SCAL))
1320 FORMAT (////5X,22HBEGIN ITERATION NUMBER,I5)
1330 FORMAT (/5X,4HCT =,E14.5,5X,5HCTL =,E14.5,5X,5HPHI =,E14.5)
1340 FORMAT (/5X,25HNEW SCALING VECTOR (SCAL))
1350 FORMAT (/5X,15HGRADIENT OF OBJ)
1360 FORMAT (/5X,44HGRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS)
1370 FORMAT (1H )
1380 FORMAT (/5X,37HPUSH-OFF FACTORS, (THETA(I), I=1,NAC))
1390 FORMAT (/5X,27HSEARCH DIRECTION (S-VECTOR))
1400 FORMAT (/5X,18HCALCULATED ALPHA =,E14.5)
1410 FORMAT (////5X,6HITER =,I5,5X,5HOBJ =,E14.5,5X,16HNO CHANGE IN OBJ
     1)
1420 FORMAT (/5X,5HOBJ =,E15.6,5X,16HNO CHANGE ON OBJ)
1430 FORMAT (/5X,5HOBJ =,E15.6)
1440 FORMAT (////5X,6HITER =,I5,5X,5HOBJ =,E14.5)
1450 FORMAT (//5X,28HINITIAL FUNCTION INFORMATION//5X,5HOBJ =,E15.6)
1460 FORMAT (/5X,29HDECISION VARIABLES (X-VECTOR))
1470 FORMAT (3X,7E13.4)
1480 FORMAT (/5X,28HCONSTRAINT VALUES (G-VECTOR))
1490 FORMAT (5X,15I5)
1500 FORMAT (/5X,59HTHE NUMBER OF ACTIVE AND VIOLATED CONSTRAINTS EXCEE
     1DS N3-1./5X,66HDIMENSIONED SIZE OF MATRICES A AND B AND VECTOR IC
     2IS INSUFFICIENT/5X,61HOPTIMIZATION TERMINATED AND CONTROL RETURNED
     3 TO MAIN PROGRAM.)
1510 FORMAT (1H1,////4X,30HFINAL OPTIMIZATION INFORMATION)
1520 FORMAT (/5X,32HOBJECTIVE FUNCTION WAS EVALUATED,8X,I5,2X,5HTIMES)
1530 FORMAT (/5X,35HCONSTRAINT FUNCTIONS WERE EVALUATED,I10,2X,5HTIMES)
1540 FORMAT (/5X,36HGRADIENT OF OBJECTIVE WAS CALCULATED,I9,2X,5HTIMES)
1550 FORMAT (/5X,40HGRADIENTS OF CONSTRAINTS WERE CALCULATED,I5,2X,5HTI
     1MES)
     END
```

```
      SUBROUTINE CNMN01                                       SEPT. 77

      SUBROUTINE CNMN01 (JGOTO,X,DF,G,ISC,IC,A,G1,VLB,VUB,SCAL,C,NCAL,DX
     1,DX1,FI,XI,III,N1,N2,N3,N4)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      DIMENSION X(N1),DF(N1),G(N2),ISC(N2),IC(N3),A(N3,N1),G1(N2),VLB(N1
     1),VUB(N1),SCAL(N1),NCAL(2),C(N4)
C     ROUTINE TO CALCULATE GRADIENT INFORMATION BY FINITE DIFFERENCE.
C     BY G. N. VANDERPLAATS                         JUNE, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
      IF (JGOTO.EQ.1) GO TO 10
      IF (JGOTO.EQ.2) GO TO 70
      INFOG=0
      INF=INFO
      NAC=0
      IF (LINOBJ.NE.0.AND.ITER.GT.1) GO TO 10
C     ------------------------------------------------------------------
C                       GRADIENT OF LINEAR OBJECTIVE
C     ------------------------------------------------------------------
      IF (NFDG.EQ.1) JGOTO=1
      IF (NFDG.EQ.1) RETURN
10    CONTINUE
      JGOTO=0
      IF (NFDG.EQ.1.AND.NCON.EQ.0) RETURN
      IF (NCON.EQ.0) GO TO 40
C     ------------------------------------------------------------------
C       * * * DETERMINE WHICH CONSTRAINTS ARE ACTIVE OR VIOLATED * * *
C     ------------------------------------------------------------------
      DO 20 I=1,NCON
      IF (G(I).LT.CT) GO TO 20
      IF (ISC(I).GT.0.AND.G(I).LT.CTL) GO TO 20
      NAC=NAC+1
      IF (NAC.GE.N3) RETURN
      IC(NAC)=I
20    CONTINUE
      IF (NFDG.EQ.1.AND.NAC.EQ.0) RETURN
      IF ((LINOBJ.GT.0.AND.ITER.GT.1).AND.NAC.EQ.0) RETURN
C     ------------------------------------------------------------------
C                     STORE VALUES OF CONSTRAINTS IN G1
C     ------------------------------------------------------------------
      DO 30 I=1,NCON
30    G1(I)=G(I)
40    CONTINUE
      JGOTO=0
      IF (NAC.EQ.0.AND.NFDG.EQ.1) RETURN
C     ------------------------------------------------------------------
C                             CALCULATE GRADIENTS
C     ------------------------------------------------------------------
      INFOG=1
      INFO=1
```

```
      FI=OBJ
      III=0
50    III=III+1
      XI=X(III)
      DX=FDCH*XI
      DX=ABS(DX)
      FDCH1=FDCHM
      IF (NSCAL.NE.0) FDCH1=FDCHM/SCAL(III)
      IF (DX.LT.FDCH1) DX=FDCH1
      X1=XI+DX
      IF (NSIDE.EQ.0) GO TO 60
      IF (X1.LT.VLB(III).AND.DX.LT.0.) X1=XI-DX
      IF (X1.GT.VUB(III).AND.DX.GT.0.) X1=XI-DX
60    DX1=1./DX
      X(III)=XI+DX
      NCAL(1)=NCAL(1)+1
C     ------------------------------------------------------------------
C                          FUNCTION EVALUATION
C     ------------------------------------------------------------------
      JGOTO=2
      RETURN
70    CONTINUE
      X(III)=XI
      IF (NFDG.EQ.0) DF(III)=DX1*(OBJ-FI)
      IF (NAC.EQ.0) GO TO 90
C     ------------------------------------------------------------------
C             DETERMINE GRADIENT COMPONENTS OF ACTIVE CONSTRAINTS
C     ------------------------------------------------------------------
      DO 80 J=1,NAC
      I1=IC(J)
80    A(J,III)=DX1*(G(I1)-G1(I1))
90    CONTINUE
      IF (III.LT.NDV) GO TO 50
      INFOG=0
      INFO=INF
      JGOTO=0
      OBJ=FI
      IF (NCON.EQ.0) RETURN
C     ------------------------------------------------------------------
C             STORE CURRENT CONSTRAINT VALUES BACK IN G-VECTOR
C     ------------------------------------------------------------------
      DO 100 I=1,NCON
100   G(I)=G1(I)
      RETURN
      END
```

SUBROUTINE CNMN02 SEPT. 77

```
      SUBROUTINE CNMN02 (NCALC,SLOPE,DFTDF1,DF,S,N1)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      DIMENSION DF(N1),S(N1)
C     ROUTINE TO DETERMINE CONJUGATE DIRECTION VECTOR OR DIRECTION
C     OF STEEPEST DESCENT FOR UNCONSTRAINED FUNCTION MINIMIZATION.
C     BY G. N. VANDERPLAATS                         APRIL, 1972.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     NCALC = CALCULATION CONTROL.
C         NCALC = 0,     S = STEEPEST DESCENT.
C         NCALC = 1,     S = CONJUGATE DIRECTION.
C     CONJUGATE DIRECTION IS FOUND BY FLETCHER-REEVES ALGORITHM.
C     ------------------------------------------------------------------
C                   CALCULATE NORM OF GRADIENT VECTOR
C     ------------------------------------------------------------------
      DFTDF=0.
      DO 10 I=1,NDV
      DFI=DF(I)
10    DFTDF=DFTDF+DFI*DFI
C     ------------------------------------------------------------------
C     **********                FIND DIRECTION S             **********
C     ------------------------------------------------------------------
      IF (NCALC.NE.1) GO TO 30
      IF (DFTDF1.LT.1.0E-20) GO TO 30
C     ------------------------------------------------------------------
C                 FIND FLETCHER-REEVES CONJUGATE DIRECTION
C     ------------------------------------------------------------------
      BETA=DFTDF/DFTDF1
      SLOPE=0.
      DO 20 I=1,NDV
      DFI=DF(I)
      SI=BETA*S(I)-DFI
      SLOPE=SLOPE+SI*DFI
20    S(I)=SI
      GO TO 50
30    CONTINUE
      NCALC=0
C     ------------------------------------------------------------------
C                 CALCULATE DIRECTION OF STEEPEST DESCENT
C     ------------------------------------------------------------------
      DO 40 I=1,NDV
40    S(I)=-DF(I)
      SLOPE=-DFTDF
50    CONTINUE
C     ------------------------------------------------------------------
C                 NORMALIZE S TO MAX ABS VALUE OF UNITY
C     ------------------------------------------------------------------
      S1=0.
      DO 60 I=1,NDV
```

```
      S2=ABS(S(I))
      IF (S2.GT.S1) S1=S2
60    CONTINUE
      IF (S1.LT.1.0E-20) S1=1.0E-20
      S1=1./S1
      DFTDF1=DFTDF*S1
      DO 70 I=1,NDV
70    S(I)=S1*S(I)
      SLOPE=S1*SLOPE
      RETURN
      END
```

SUBROUTINE CNMN03 SEPT. 77

```
      SUBROUTINE CNMN03 (X,S,DF,G,A,IC,SCAL,C,N1,N2,N3,N4)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /CONSAV/ D1(20),SLOPE,D2(3),XI,D3,
     2ALP,FFF,A1,A2,A3,A4,F1,F2,F3,F4,D4(4),APP,
     *D5(8),RSPACE,ID1(6),KCOUNT,NCAL(2),ID2(4),KOUNT,ID3(8),
     *JGOTO,ISPACE(2)
      DIMENSION X(N1),S(N1),DF(N1),G(N2),A(N3,N1),IC(N3),SCAL(N1),C(N4)
C     ROUTINE TO SOLVE ONE-DIMENSIONAL SEARCH IN UNCONSTRAINED
C     MINIMIZATION USING 2-POINT QUADRATIC INTERPOLATION, 3-POINT
C     CUBIC INTERPOLATION AND 4-POINT CUBIC INTERPOLATION.
C     BY G. N. VANDERPLAATS                         APRIL, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
C     ALP = PROPOSED MOVE PARAMETER.
C     SLOPE = INITIAL FUNCTION SLOPE = S-TRANSPOSE TIMES DF.
C     SLOPE MUST BE NEGATIVE.
C     OBJ = INITIAL FUNCTION VALUE.
      ZRO=0.
      IF (JGOTO.EQ.0) GO TO 10
      GO TO (50,80,110,140,180,220,270), JGOTO
C     ------------------------------------------------------------------
C                         INITIAL INFORMATION  (ALPHA=0)
C     ------------------------------------------------------------------
10    IF (SLOPE.LT.0.) GO TO 20
      ALP=0.
      RETURN
20    CONTINUE
      IF (IPRINT.GT.4) WRITE (6,360)
      FFF=OBJ
      AP1=0.
      A1=0.
      F1=OBJ
      A2=ALP
      A3=0.
      F3=0.
      AP=A2
      KOUNT=0
C     ------------------------------------------------------------------
C              MOVE A DISTANCE AP*S AND UPDATE FUNCTION VALUE
C     ------------------------------------------------------------------
30    CONTINUE
      KOUNT=KOUNT+1
      DO 40 I=1,NDV
40    X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) AP
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=1
      RETURN
```

```
50    CONTINUE
      F2=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F2
      IF (F2.LT.F1) GO TO 120
C     ------------------------------------------------------------------------
C                       CHECK FOR ILL-CONDITIONING
C     ------------------------------------------------------------------------
      IF (KOUNT.GT.5) GO TO 60
      FF=2.*ABS(F1)
      IF (F2.LT.FF) GO TO 90
      FF=5.*ABS(F1)
      IF (F2.LT.FF) GO TO 60
      A2=.5*A2
      AP=-A2
      ALP=A2
      GO TO 30
60    F3=F2
      A3=A2
      A2=.5*A2
C     ------------------------------------------------------------------------
C                  UPDATE DESIGN VECTOR AND FUNCTION VALUE
C     ------------------------------------------------------------------------
      AP=A2-ALP
      ALP=A2
      DO 70 I=1,NDV
70    X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) A2
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=2
      RETURN
80    CONTINUE
      F2=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F2
C     PROCEED TO CUBIC INTERPOLATION.
      GO TO 160
90    CONTINUE
C     ------------------------------------------------------------------------
C     **********         2-POINT QUADRATIC INTERPOLATION          **********
C     ------------------------------------------------------------------------
      JJ=1
      II=1
      CALL CNMN04 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,ZRO,ZRO,ZRO,ZRO)
      IF (APP.LT.ZRO.OR.APP.GT.A2) GO TO 120
      F3=F2
      A3=A2
      A2=APP
      JJ=0
C     ------------------------------------------------------------------------
C                   UPDATE DESIGN VECTOR AND FUNCTION VALUE
```

```
C     -----------------------------------------------------------------
      AP=A2-ALP
      ALP=A2
      DO 100 I=1,NDV
100   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) A2
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=3
      RETURN
110   CONTINUE
      F2=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F2
      GO TO 150
120   A3=2.*A2
C     -----------------------------------------------------------------
C                   UPDATE DESIGN VECTOR AND FUNCTION VALUE
C     -----------------------------------------------------------------
      AP=A3-ALP
      ALP=A3
      DO 130 I=1,NDV
130   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) A3
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=4
      RETURN
140   CONTINUE
      F3=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F3
150   CONTINUE
      IF (F3.LT.F2) GO TO 190
160   CONTINUE
C     -----------------------------------------------------------------
C     **********          3-POINT CUBIC INTERPOLATION          **********
C     -----------------------------------------------------------------
      II=3
      CALL CNMN04 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,A3,F3,ZRO,ZRO)
      IF (APP.LT.ZRO.OR.APP.GT.A3) GO TO 190
C     -----------------------------------------------------------------
C     UPDATE DESIGN VECTOR AND FUNCTION VALUE.
C     -----------------------------------------------------------------
      AP1=APP
      AP=APP-ALP
      ALP=APP
      DO 170 I=1,NDV
170   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) ALP
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
```

```
      JGOTO=5
      RETURN
180   CONTINUE
      IF (IPRINT.GT.4) WRITE (6,390) OBJ
C     ------------------------------------------------------------------
C                           CHECK CONVERGENCE
C     ------------------------------------------------------------------
      AA=1.-APP/A2
      AB2=ABS(F2)
      AB3=ABS(OBJ)
      AB=AB2
      IF (AB3.GT.AB) AB=AB3
      IF (AB.LT.1.0E-15) AB=1.0E-15
      AB=(AB2-AB3)/AB
      IF (ABS(AB).LT.1.0E-15.AND.ABS(AA).LT..001) GO TO 330
      A4=A3
      F4=F3
      A3=APP
      F3=OBJ
      IF (A3.GT.A2) GO TO 230
      A3=A2
      F3=F2
      A2=APP
      F2=OBJ
      GO TO 230
190   CONTINUE
C     ------------------------------------------------------------------
C     **********         4-POINT CUBIC INTERPOLATION          **********
C     ------------------------------------------------------------------
200   CONTINUE
      A4=2.*A3
C     UPDATE DESIGN VECTOR AND FUNCTION VALUE.
      AP=A4-ALP
      ALP=A4
      DO 210 I=1,NDV
210   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) ALP
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=6
      RETURN
220   CONTINUE
      F4=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F4
      IF (F4.GT.F3) GO TO 230
      A1=A2
      F1=F2
      A2=A3
      F2=F3
      A3=A4
```

```
      F3=F4
      GO TO 200
230   CONTINUE
      II=4
      CALL CNMN04 (II,APP,A1,A1,F1,SLOPE,A2,F2,A3,F3,A4,F4)
      IF (APP.GT.A1) GO TO 250
      AP=A1-ALP
      ALP=A1
      OBJ=F1
      DO 240 I=1,NDV
240   X(I)=X(I)+AP*S(I)
      GO TO 280
250   CONTINUE
C     ------------------------------------------------------------------
C                 UPDATE DESIGN VECTOR AND FUNCTION VALUE
C     ------------------------------------------------------------------
      AP=APP-ALP
      ALP=APP
      DO 260 I=1,NDV
260   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) ALP
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=7
      RETURN
270   CONTINUE
      IF (IPRINT.GT.4) WRITE (6,390) OBJ
280   CONTINUE
C     ------------------------------------------------------------------
C                     CHECK FOR ILL-CONDITIONING
C     ------------------------------------------------------------------
      IF (OBJ.GT.F2.OR.OBJ.GT.F3) GO TO 290
      IF (OBJ.LE.F1) GO TO 330
      AP=A1-ALP
      ALP=A1
      OBJ=F1
      GO TO 310
290   CONTINUE
      IF (F2.LT.F3) GO TO 300
      OBJ=F3
      AP=A3-ALP
      ALP=A3
      GO TO 310
300   OBJ=F2
      AP=A2-ALP
      ALP=A2
310   CONTINUE
C     ------------------------------------------------------------------
C                        UPDATE DESIGN VECTOR
C     ------------------------------------------------------------------
```

```
      DO 320 I=1,NDV
320   X(I)=X(I)+AP*S(I)
330   CONTINUE
C     ------------------------------------------------------------------
C                    CHECK FOR MULTIPLE MINIMA
C     ------------------------------------------------------------------
      IF (OBJ.LE.FFF) GO TO 350
C     INITIAL FUNCTION IS MINIMUM.
      DO 340 I=1,NDV
340   X(I)=X(I)-ALP*S(I)
      ALP=0.
      OBJ=FFF
350   CONTINUE
      JGOTO=0
      RETURN
C     ------------------------------------------------------------------
C                                  FORMATS
C     ------------------------------------------------------------------
C
360   FORMAT (/////5X,60H* * * UNCONSTRAINED ONE-DIMENSIONAL SEARCH INFO
     1RMATION * * *)
370   FORMAT (/5X,7HALPHA =,E14.5/5X,8HX-VECTOR)
380   FORMAT (5X,6E13.5)
390   FORMAT (/5X,5HOBJ =,E14.5)
      END
```

```
      SUBROUTINE CNMN04 (II,XBAR,EPS,X1,Y1,SLOPE,X2,Y2,X3,Y3,X4,Y4)
C     ROUTINE TO FIND FIRST XBAR.GE.EPS CORRESPONDING TO A MINIMUM
C     OF A ONE-DIMENSIONAL REAL FUNCTION BY POLYNOMIEL INTERPOLATION.
C     BY G. N. VANDERPLAATS                          APRIL, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
C
C     II = CALCULATION CONTROL.
C          1:  2-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, SLOPE,
C              X2 AND Y2.
C          2:  3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2,
C              X3 AND Y3.
C          3:  3-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2, Y2,
C              X3 AND Y3.
C          4:  4-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3,
C              Y3, X4 AND Y4.
C     EPS MAY BE NEGATIVE.
C     IF REQUIRED MINIMUM ON Y DOES NOT EXITS, OR THE FUNCTION IS
C     ILL-CONDITIONED, XBAR = EPS-1.0 WILL BE RETURNED AS AN ERROR
C     INDICATOR.
C     IF DESIRED INTERPOLATION IS ILL-CONDITIONED, A LOWER ORDER
C     INTERPOLATION, CONSISTANT WITH INPUT DATA, WILL BE ATTEMPTED,
C     AND II WILL BE CHANGED ACCORDINGLY.
      XBAR1=EPS-1.
      XBAR=XBAR1
      X21=X2-X1
      IF (ABS(X21).LT.1.0E-20) RETURN
      NSLOP=MOD(II,2)
      GO TO (10,20,40,50), II
10    CONTINUE
C     ------------------------------------------------------------------
C                  II=1: 2-POINT QUADRATIC INTERPOLATION
C     ------------------------------------------------------------------
      II=1
      DX=X1-X2
      IF (ABS(DX).LT.1.0E-20) RETURN
      AA=(SLOPE+(Y2-Y1)/DX)/DX
      IF (AA.LT.1.0E-20) RETURN
      BB=SLOPE-2.*AA*X1
      XBAR=-.5*BB/AA
      IF (XBAR.LT.EPS) XBAR=XBAR1
      RETURN
20    CONTINUE
C     ------------------------------------------------------------------
C                  II=2: 3-POINT QUADRATIC INTERPOLATION
C     ------------------------------------------------------------------
      II=2
      X21=X2-X1
      X31=X3-X1
      X32=X3-X2
      QQ=X21*X31*X32
```

```
      IF (ABS(QQ).LT.1.0E-20) RETURN
      AA=(Y1*X32-Y2*X31+Y3*X21)/QQ
      IF (AA.LT.1.0E-20) GO TO 30
      BB=(Y2-Y1)/X21-AA*(X1+X2)
      XBAR=-.5*BB/AA
      IF (XBAR.LT.EPS) XBAR=XBAR1
      RETURN
30    CONTINUE
      IF (NSLOP.EQ.0) RETURN
      GO TO 10
40    CONTINUE
C     ------------------------------------------------------------------
C                  II=3: 3-POINT CUBIC INTERPOLATION
C     ------------------------------------------------------------------
      II=3
      X21=X2-X1
      X31=X3-X1
      X32=X3-X2
      QQ=X21*X31*X32
      IF (ABS(QQ).LT.1.0E-20) RETURN
      X11=X1*X1
      DNOM=X2*X2*X31-X11*X32-X3*X3*X21
      IF (ABS(DNOM).LT.1.0E-20) GO TO 20
      AA=((X31*X31*(Y2-Y1)-X21*X21*(Y3-Y1))/(X31*X21)-SLOPE*X32)/DNOM
      IF (ABS(AA).LT.1.0E-20) GO TO 20
      BB=((Y2-Y1)/X21-SLOPE-AA*(X2*X2+X1*X2-2.*X11))/X21
      CC=SLOPE-3.*AA*X11-2.*BB*X1
      BAC=BB*BB-3.*AA*CC
      IF (BAC.LT.0.) GO TO 20
      BAC=SQRT(BAC)
      XBAR=(BAC-BB)/(3.*AA)
      IF (XBAR.LT.EPS) XBAR=EPS
      RETURN
50    CONTINUE
C     ------------------------------------------------------------------
C                  II=4: 4-POINT CUBIC INTERPOLATION
C     ------------------------------------------------------------------
      X21=X2-X1
      X31=X3-X1
      X41=X4-X1
      X32=X3-X2
      X42=X4-X2
      X11=X1*X1
      X22=X2*X2
      X33=X3*X3
      X44=X4*X4
      X111=X1*X11
      X222=X2*X22
      Q2=X31*X21*X32
      IF (ABS(Q2).LT.1.0E-30) RETURN
```

```
      SUBROUTINE CNMN04                                          SEPT. 77
      Q1=X111*X32-X222*X31+X3*X33*X21
      Q4=X111*X42-X222*X41+X4*X44*X21
      Q5=X41*X21*X42
      DNOM=Q2*Q4-Q1*Q5
      IF (ABS(DNOM).LT.1.0E-30) GO TO 60
      Q3=Y3*X21-Y2*X31+Y1*X32
      Q6=Y4*X21-Y2*X41+Y1*X42
      AA=(Q2*Q6-Q3*Q5)/DNOM
      BB=(Q3-Q1*AA)/Q2
      CC=(Y2-Y1-AA*(X222-X111))/X21-BB*(X1+X2)
      BAC=BB*BB-3.*AA*CC
      IF (ABS(AA).LT.1.0E-20.OR.BAC.LT.0.) GO TO 60
      BAC=SQRT(BAC)
      XBAR=(BAC-BB)/(3.*AA)
      IF (XBAR.LT.EPS) XBAR=XBAR1
      RETURN
60    CONTINUE
      IF (NSLOP.EQ.1) GO TO 40
      GO TO 20
      END
```

```
      SUBROUTINE CNMNO5 (NVC,SLOPE,DF,G,ISC,IC,A,S,C,MS1,B,N1,N2,N3,N4,N
     15)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      DIMENSION DF(N1),G(N2),ISC(N2),IC(N3),A(N3,N1),S(N1),C(N4),MS1(N5)
     1,B(N3,N3)
C     ROUTINE TO SOLVE DIRECTION FINDING PROBLEM IN MODIFIED METHOD OF
C     FEASIBLE DIRECTIONS.
C     BY G. N. VANDERPLAATS                          MAY, 1972.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     NORM OF S VECTOR USED HERE IS S-TRANSPOSE TIMES S.LE.1.
C     IF NVC = 0 FIND DIRECTION BY ZOUTENDIJK'S METHOD.  OTHERWISE
C     FIND MODIFIED DIRECTION.
C     ------------------------------------------------------------------
C     ***  NORMALIZE GRADIENTS, CALCULATE THETA'S AND DETERMINE NVC  ***
C     ------------------------------------------------------------------
      NDV1=NDV+1
      NDV2=NDV+2
      NAC1=NAC+1
      NVC=0
      THMAX=0.
      CTA=ABS(CT)
      CT1=1./CTA
      CTAM=ABS(CTMIN)
      CTB=ABS(CTL)
      CT2=1./CTB
      CTBM=ABS(CTLMIN)
      A1=1.
      DO 40 I=1,NAC
C     CALCULATE THETA
      NCI=IC(I)
      NCJ=1
      IF (NCI.LE.NCON) NCJ=ISC(NCI)
      C1=G(NCI)
      CTD=CT1
      CTC=CTAM
      IF (NCJ.LE.0) GO TO 10
      CTC=CTBM
      CTD=CT2
10    IF (C1.GT.CTC) NVC=NVC+1
      THT=0.
      GG=1.+CTD*C1
      IF (NCJ.EQ.0.OR.C1.GT.CTC) THT=THETA*GG*GG
      IF (NCJ.GT.0.AND.C1.GT.CTC) THT=THT-3.*THETA
      IF (THT.GT.50.) THT=50.
      IF (THT.GT.THMAX) THMAX=THT
      A(I,NDV1)=THT
C     ------------------------------------------------------------------
C                    NORMALIZE GRADIENTS OF CONSTRAINTS
```

```
C     -----------------------------------------------------------------
      A(I,NDV2)=1.
      IF (NCI.GT.NCON) GO TO 40
      A1=0.
      DO 20 J=1,NDV
      A1=A1+A(I,J)**2
20    CONTINUE
      IF (A1.LT.1.0E-20) A1=1.0E-20
      A1=SQRT(A1)
      A(I,NDV2)=A1
      A1=1./A1
      DO 30 J=1,NDV
30    A(I,J)=A1*A(I,J)
40    CONTINUE
C     -----------------------------------------------------------------
C     NORMALIZE GRADIENT OF OBJECTIVE FUNCTION AND STORE IN NAC+1
C     ROW OF A
C     -----------------------------------------------------------------
      A1=0.
      DO 50 I=1,NDV
      A1=A1+DF(I)**2
50    CONTINUE
      IF (A1.LT.1.0E-20) A1=1.0E-20
      A1=SQRT(A1)
      A1=1./A1
      DO 60 I=1,NDV
60    A(NAC1,I)=A1*DF(I)
C     BUILD C VECTOR.
      IF (NVC.GT.0) GO TO 80
C     -----------------------------------------------------------------
C                   BUILD C FOR CLASSICAL METHOD
C     -----------------------------------------------------------------
      NDB=NAC1
      A(NDB,NDV1)=1.
      DO 70 I=1,NDB
70    C(I)=-A(I,NDV1)
      GO TO 110
80    CONTINUE
C     -----------------------------------------------------------------
C                    BUILD C FOR MODIFIED METHOD
C     -----------------------------------------------------------------
      NDB=NAC
      A(NAC1,NDV1)=-PHI
C     -----------------------------------------------------------------
C           SCALE THETA'S SO THAT MAXIMUM THETA IS UNITY
C     -----------------------------------------------------------------
      IF(THMAX.GT.0.00001) THMAX=1./THMAX
      DO 90 I=1,NDB
      NCI=IC(I)
      C1=CTA
```

```
      IF (ISC(NCI).GT.0) C1=CTB
      A(I,NDV1)=A(I,NDV1)*THMAX
90    CONTINUE
      DO 100 I=1,NDB
      C(I)=0.
      DO 100 J=1,NDV1
100   C(I)=C(I)+A(I,J)*A(NAC1,J)
110   CONTINUE
C     ------------------------------------------------------------------
C                        BUILD B MATRIX
C     ------------------------------------------------------------------
      DO 120 I=1,NDB
      DO 120 J=1,NDB
      B(I,J)=0.
      DO 120 K=1,NDV1
120   B(I,J)=B(I,J)-A(I,K)*A(J,K)
C     ------------------------------------------------------------------
C                     SOLVE SPECIAL L. P. PROBLEM
C     ------------------------------------------------------------------
      CALL CNMN08 (NDB,NER,C,MS1,B,N3,N4,N5)
      IF (IPRINT.GT.1.AND.NER.GT.0) WRITE (6,180)
C     CALCULATE RESULTING DIRECTION VECTOR, S.
      SLOPE=0.
C     ------------------------------------------------------------------
C                     USABLE-FEASIBLE DIRECTION
C     ------------------------------------------------------------------
      DO 140 I=1,NDV
      S1=0.
      IF (NVC.GT.0) S1=-A(NAC1,I)
      DO 130 J=1,NDB
130   S1=S1-A(J,I)*C(J)
      SLOPE=SLOPE+S1*DF(I)
140   S(I)=S1
      S(NDV1)=1.
      IF (NVC.GT.0) S(NDV1)=-A(NAC1,NDV1)
      DO 150 J=1,NDB
150   S(NDV1)=S(NDV1)-A(J,NDV1)*C(J)
C     ------------------------------------------------------------------
C                     NORMALIZE S TO MAX ABS OF UNITY
C     ------------------------------------------------------------------
      S1=0.
      DO 160 I=1,NDV
      A1=ABS(S(I))
      IF (A1.GT.S1) S1=A1
160   CONTINUE
      IF (S1.LT.1.0E-10) S1=1.0E-10
      S1=1./S1
      DO 170 I=1,NDV
170   S(I)=S1*S(I)
      SLOPE=S1*SLOPE
```

```
      S(NDV1)=S1*S(NDV1)
      RETURN
C
180   FORMAT (//5X,46H* * DIRECTION FINDING PROCESS DID NOT CONVERGE/5X,
     129H* * S-VECTOR MAY NOT BE VALID)
      END
```

```
      SUBROUTINE CNMN06 (X,DF,G,ISC,S,G1,G2,VLB,VUB,SCAL,N1,N2)
      COMMON /CNMN1/ IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,FDCH,
     1FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,LINOBJ,ITRM,
     2ITER,INFOG,IGOTO,INFO,OBJ
      COMMON /CONSAV/D1(16),CTA,CTAM,CTBM,D2,SLOPE,D3(3),XI,
     2D4,ALP,D5(2),A2,A3,A4,D6,F2,F3,F4,CV1,CV2,CV3,CV4,D7,ALPCA,A
     3LPFES,ALPLN,ALPMIN,ALPNC,ALPSAV,ALPSID,ALPTOT,RSPACE,ID1(7),
     *NCAL(2),ID2(3),NVC,ID3,ICOUNT,
     5IGOOD1,IGOOD2,IGOOD3,IGOOD4,IBEST,III,NLNC,JGOTO,ISPACE(2)
      DIMENSION X(N1),DF(N1),G(N2),ISC(N2),S(N1),G1(N2),G2(N2),VLB(N1),V
     1UB(N1),SCAL(N1)
C     ROUTINE TO SOLVE ONE-DIMENSIONAL SEARCH PROBLEM FOR CONSTRAINED
C     FUNCTION MINIMIZATION.
C     BY G. N. VANDERPLAATS                          AUG., 1974.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     OBJ = INITIAL AND FINAL FUNCTION VALUE.
C     ALP = MOVE PARAMETER.
C     SLOPE = INITIAL SLOPE.
C
C     ALPSID = MOVE TO SIDE CONSTRAINT.
C     ALPFES = MOVE TO FEASIBLE REGION.
C     ALPNC = MOVE TO NEW NON-LINEAR CONSTRAINT.
C     ALPLN = MOVE TO LINEAR CONSTRAINT.
C     ALPCA = MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT.
C     ALPMIN = MOVE TO MINIMIZE FUNCTION.
C     ALPTOT = TOTAL MOVE PARAMETER.
      ZRO=0.
      IF (JGOTO.EQ.0) GO TO 10
      GO TO (140,310,520), JGOTO
10    IF (IPRINT.GE.5) WRITE (6,730)
      ALPSAV=ALP
      ICOUNT=0
      ALPTOT=0.
C     TOLERANCES.
      CTAM=ABS(CTMIN)
      CTBM=ABS(CTLMIN)
C     PROPOSED MOVE.
20    CONTINUE
C     ------------------------------------------------------------------
C     *****  BEGIN SEARCH OR IMPOSE SIDE CONSTRAINT MODIFICATION  *****
C     ------------------------------------------------------------------
      A2=ALPSAV
      ICOUNT=ICOUNT+1
      ALPSID=1.0E+20
C     INITIAL ALPHA AND OBJ.
      ALP=0.
      F1=OBJ
      KSID=0
      IF (NSIDE.EQ.0) GO TO 70
C     ------------------------------------------------------------------
```

```
C       FIND MOVE TO SIDE CONSTRAINT AND INSURE AGAINST VIOLATION OF
C       SIDE CONSTRAINTS
C       ------------------------------------------------------------------
        DO 60 I=1,NDV
        SI=S(I)
        IF (ABS(SI).GT.1.0E-20) GO TO 30
C                 CALCULATE ALPHA TO MINIMIZE FUNCTION
C       ------------------------------------------------------------------
        II=3
        IF (A2.GT.A3.AND.(IGOOD2.EQ.0.AND.IBEST.EQ.2)) II=2
        CALL CNMN04 (II,ALPMIN,ZRO,ZRO,F1,SLOPE,A2,F2,A3,F3,ZRO,ZRO)
450     CONTINUE
C       ------------------------------------------------------------------
C                           PROPOSED MOVE
C       ------------------------------------------------------------------
C       MOVE AT LEAST ENOUGH TO OVERCOME CONSTRAINT VIOLATIONS.
        A4=ALPFES
C       MOVE TO MINIMIZE FUNCTION.
        IF (ALPMIN.GT.A4) A4=ALPMIN
C       IF A4.LE.0, SET A4 = ALPSID.
        IF (A4.LE.0.) A4=ALPSID
C       LIMIT MOVE TO NEW CONSTRAINT ENCOUNTER.
        IF (A4.GT.ALPLN) A4=ALPLN
        IF (A4.GT.ALPNC) A4=ALPNC
C       LIMIT MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT.
        IF (A4.GT.ALPCA) A4=ALPCA
C       LIMIT A4 TO 5.*A3.
        IF (A4.GT.(5.*A3)) A4=5.*A3
C       UPDATE DESIGN.
        IF (IBEST.NE.3.OR.NCON.EQ.0) GO TO 470
C       STORE CONSTRAINT VALUES IN G2.  F3 IS BEST.  F2 IS NOT.
        DO 460 I=1,NCON
        G2(I)=G(I)
460     CONTINUE
470     CONTINUE
C       IF A4=A3 AND IGOOD1=0 AND IGOOD3=1, SET A4=.9*A3.
        ALP=A4-A3
        IF ((IGOOD1.EQ.0.AND.IGOOD3.EQ.1).AND.ABS(ALP).LT.1.0E-20) A4=.9*A
       13
C       ------------------------------------------------------------------
C                        MOVE A DISTANCE A4*S
C       ------------------------------------------------------------------
        ALP=A4-A3
        ALPTOT=ALPTOT+ALP
        DO 480 I=1,NDV
        X(I)=X(I)+ALP*S(I)
480     CONTINUE
        IF (IPRINT.LT.5) GO TO 510
        WRITE (6,720)
        WRITE (6,740) A4
```

```
      IF (NSCAL.EQ.0) GO TO 500
      DO 490 I=1,NDV
490   G(I)=SCAL(I)*X(I)
      WRITE (6,750) (G(I),I=1,NDV)
      GO TO 510
500   WRITE (6,750) (X(I),I=1,NDV)
510   CONTINUE
C     ------------------------------------------------------------------
C              UPDATE FUNCTION AND CONSTRAINT VALUES
C     ------------------------------------------------------------------
      NCAL(1)=NCAL(1)+1
      JGOTO=3
      RETURN
520   CONTINUE
      F4=OBJ
      IF (IPRINT.GE.5) WRITE (6,760) F4
      IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 530
      WRITE (6,770)
      WRITE (6,750) (G(I),I=1,NCON)
530   CONTINUE
C     DETERMINE ACCAPTABILITY OF F4.
      IGOOD4=0
      CV4=0.
      IF (NCON.EQ.0) GO TO 550
      DO 540 I=1,NCON
      CC=CTAM
      IF (ISC(I).GT.0) CC=CTBM
      C1=G(I)-CC
      IF (C1.GT.CV4) CV4=C1
540   CONTINUE
      IF (CV4.GT.0.) IGOOD4=1
550   CONTINUE
      ALP=A4
      OBJ=F4
C     ------------------------------------------------------------------
C                       DETERMINE BEST DESIGN
C     ------------------------------------------------------------------
      GO TO (560,610,660), IBEST
560   CONTINUE
C     CHOOSE BETWEEN F1 AND F4.
      IF (IGOOD1.EQ.0.AND.IGOOD4.EQ.0) GO TO 570
      IF (CV1.GT.CV4) GO TO 710
      GO TO 580
570   CONTINUE
      IF (F4.LE.F1) GO TO 710
580   CONTINUE
C     F1 IS BEST.
      ALPTOT=ALPTOT-A4
      OBJ=F1
      DO 590 I=1,NDV
```

```
      X(I)=X(I)-A4*S(I)
590   CONTINUE
      IF (NCON.EQ.0) GO TO 710
      DO 600 I=1,NCON
      G(I)=G1(I)
600   CONTINUE
      GO TO 710
610   CONTINUE
C     CHOOSE BETWEEN F2 AND F4.
      IF (IGOOD2.EQ.0.AND.IGOOD4.EQ.0) GO TO 620
      IF (CV2.GT.CV4) GO TO 710
      GO TO 630
620   CONTINUE
      IF (F4.LE.F2) GO TO 710
630   CONTINUE
C     F2 IS BEST.
      OBJ=F2
      A2=A4-A2
      ALPTOT=ALPTOT-A2
      DO 640 I=1,NDV
      X(I)=X(I)-A2*S(I)
640   CONTINUE
      IF (NCON.EQ.0) GO TO 710
      DO 650 I=1,NCON
      G(I)=G2(I)
650   CONTINUE
      GO TO 710
660   CONTINUE
C     CHOOSE BETWEEN F3 AND F4.
      IF (IGOOD3.EQ.0.AND.IGOOD4.EQ.0) GO TO 670
      IF (CV3.GT.CV4) GO TO 710
      GO TO 680
670   CONTINUE
      IF (F4.LE.F3) GO TO 710
680   CONTINUE
C     F3 IS BEST.
      OBJ=F3
      A3=A4-A3
      ALPTOT=ALPTOT-A3
      DO 690 I=1,NDV
      X(I)=X(I)-A3*S(I)
690   CONTINUE
      IF (NCON.EQ.0) GO TO 710
      DO 700 I=1,NCON
      G(I)=G2(I)
700   CONTINUE
710   CONTINUE
      ALP=ALPTOT
      IF (IPRINT.GE.5) WRITE (6,790)
      JGOTO=0
```

```
      RETURN
C     ------------------------------------------------------------------
C                                FORMATS
C     ITH COMPONENT OF S IS SMALL.  SET TO ZERO.
      S(I)=0.
      SLOPE=SLOPE-SI*DF(I)
      GO TO 60
30    CONTINUE
      XI=X(I)
      SI=1./SI
      IF (SI.GT.0.) GO TO 40
C     LOWER BOUND.
      XI2=VLB(I)
      XI1=ABS(XI2)
      IF (XI1.LT.1.) XI1=1.
C     CONSTRAINT VALUE.
      GI=(XI2-XI)/XI1
      IF (GI.GT.-1.0E-6) GO TO 50
C     PROPOSED MOVE TO LOWER BOUND.
      ALPA=(XI2-XI)*SI
      IF (ALPA.LT.ALPSID) ALPSID=ALPA
      GO TO 60
40    CONTINUE
C     UPPER BOUND.
      XI2=VUB(I)
      XI1=ABS(XI2)
      IF (XI1.LT.1.) XI1=1.
C     CONSTRAINT VALUE.
      GI=(XI-XI2)/XI1
      IF (GI.GT.-1.0E-6) GO TO 50
C     PROPOSED MOVE TO UPPER BOUND.
      ALPA=(XI2-XI)*SI
      IF (ALPA.LT.ALPSID) ALPSID=ALPA
      GO TO 60
50    CONTINUE
C     MOVE WILL VIOLATE SIDE CONSTRAINT.  SET S(I)=0.
      SLOPE=SLOPE-S(I)*DF(I)
      S(I)=0.
      KSID=KSID+1
60    CONTINUE
C     ALPSID IS UPPER BOUND ON ALPHA.
      IF (A2.GT.ALPSID) A2=ALPSID
70    CONTINUE
C     ------------------------------------------------------------------
C                    CHECK ILL-CONDITIONING
C     ------------------------------------------------------------------
      IF (KSID.EQ.NDV.OR.ICOUNT.GT.10) GO TO 710
      IF (NVC.EQ.0.AND.SLOPE.GT.0.) GO TO 710
      ALPFES=-1.
      ALPMIN=-1.
```

```
      ALPLN=1.1*ALPSID
      ALPNC=ALPSID
      ALPCA=ALPSID
      IF (NCON.EQ.0) GO TO 90
C     STORE CONSTRAINT VALUES IN G1.
      DO 80 I=1,NCON
      G1(I)=G(I)
80    CONTINUE
90    CONTINUE
C     ------------------------------------------------------------------
C                    MOVE A DISTANCE A2*S
C     ------------------------------------------------------------------
      ALPTOT=ALPTOT+A2
      DO 100 I=1,NDV
      X(I)=X(I)+A2*S(I)
100   CONTINUE
      IF (IPRINT.LT.5) GO TO 130
      WRITE (6,740) A2
      IF (NSCAL.EQ.0) GO TO 120
      DO 110 I=1,NDV
110   G(I)=SCAL(I)*X(I)
      WRITE (6,750) (G(I),I=1,NDV)
      GO TO 130
120   WRITE (6,750) (X(I),I=1,NDV)
C     ------------------------------------------------------------------
C                    UPDATE FUNCTION AND CONSTRAINT VALUES
C     ------------------------------------------------------------------
130   NCAL(1)=NCAL(1)+1
      JGOTO=1
      RETURN
140   CONTINUE
      F2=OBJ
      IF (IPRINT.GE.5) WRITE (6,760) F2
      IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 150
      WRITE (6,770)
      WRITE (6,750) (G(I),I=1,NCON)
150   CONTINUE
C     ------------------------------------------------------------------
C               IDENTIFY ACCAPTABILITY OF DESIGNS F1 AND F2
C     ------------------------------------------------------------------
C     IGOOD = 0 IS ACCAPTABLE.
C     CV = MAXIMUM CONSTRAINT VIOLATION.
      IGOOD1=0
      IGOOD2=0
      CV1=0.
      CV2=0.
      NVC1=0
      IF (NCON.EQ.0) GO TO 170
      DO 160 I=1,NCON
      CC=CTAM
```

```
      IF (ISC(I).GT.0) CC=CTBM
      C1=G1(I)-CC
      C2=G(I)-CC
      IF (C2.GT.0.) NVC1=NVC1+1
      IF (C1.GT.CV1) CV1=C1
      IF (C2.GT.CV2) CV2=C2
160   CONTINUE
      IF (CV1.GT.0.) IGOOD1=1
      IF (CV2.GT.0.) IGOOD2=1
170   CONTINUE
      ALP=A2
      OBJ=F2
C     ------------------------------------------------------------------
C     IF F2 VIOLATES FEWER CONSTRAINTS THAN F1 BUT STILL HAS CONSTRAINT
C     VIOLATIONS RETURN
C     ------------------------------------------------------------------
      IF (NVC1.LT.NVC.AND.NVC1.GT.0) GO TO 710
C     ------------------------------------------------------------------
C              IDENTIFY BEST OF DESIGNS F1 ANF F2
C     ------------------------------------------------------------------
C     IBEST CORRESPONDS TO MINIMUM VALUE DESIGN.
C     IF CONSTRAINTS ARE VIOLATED, IBEST CORRESPONDS TO MINIMUM
C     CONSTRAINT VIOLATION.
      IF (IGOOD1.EQ.0.AND.IGOOD2.EQ.0) GO TO 180
C     VIOLATED CONSTRAINTS.  PICK MINIMUM VIOLATION.
      IBEST=1
      IF (CV1.GE.CV2) IBEST=2
      GO TO 190
180   CONTINUE
C     NO CONSTRAINT VIOLATION.  PICK MINIMUM F.
      IBEST=1
      IF (F2.LE.F1) IBEST=2
190   CONTINUE
      II=1
      IF (NCON.EQ.0) GO TO 230
C     ------------------------------------------------------------------
C     *****                2 - POINT INTERPOLATION                 *****
C     ------------------------------------------------------------------
      III=0
200   III=III+1
      C1=G1(III)
      C2=G(III)
      IF (ISC(III).EQ.0) GO TO 210
C     ------------------------------------------------------------------
C                           LINEAR CONSTRAINT
C     ------------------------------------------------------------------
      IF (C1.GE.1.0E-5.AND.C1.LE.CTBM) GO TO 220
      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,ZRO,ZRO)
      IF (ALP.LE.0.) GO TO 220
      IF (C1.GT.CTBM.AND.ALP.GT.ALPFES) ALPFES=ALP
```

```
      IF (C1.LT.CTL.AND.ALP.LT.ALPLN) ALPLN=ALP
      GO TO 220
210   CONTINUE
C     ------------------------------------------------------------------
C                        NON-LINEAR CONSTRAINT
C     ------------------------------------------------------------------
      IF (C1.GE.1.0E-5.AND.C1.LE.CTAM) GO TO 220
      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,ZRO,ZRO)
      IF (ALP.LE.0.) GO TO 220
      IF (C1.GT.CTAM.AND.ALP.GT.ALPFES) ALPFES=ALP
      IF (C1.LT.CT.AND.ALP.LT.ALPNC) ALPNC=ALP
220   CONTINUE
      IF (III.LT.NCON) GO TO 200
230   CONTINUE
      IF (LINOBJ.GT.0.OR.SLOPE.GE.0.) GO TO 240
C     CALCULATE ALPHA TO MINIMIZE FUNCTION.
      CALL CNMN04 (II,ALPMIN,ZRO,ZRO,F1,SLOPE,A2,F2,ZRO,ZRO,ZRO,ZRO)
240   CONTINUE
C     ------------------------------------------------------------------
C                            PROPOSED MOVE
C     ------------------------------------------------------------------
C     MOVE AT LEAST FAR ENOUGH TO OVERCOME CONSTRAINT VIOLATIONS.
      A3=ALPFES
C     MOVE TO MINIMIZE FUNCTION.
      IF (ALPMIN.GT.A3) A3=ALPMIN
C     IF A3.LE.0, SET A3 = ALPSID.
      IF (A3.LE.0.) A3=ALPSID
C     LIMIT MOVE TO NEW CONSTRAINT ENCOUNTER.
      IF (A3.GT.ALPNC) A3=ALPNC
      IF (A3.GT.ALPLN) A3=ALPLN
C     MAKE A3 NON-ZERO.
      IF (A3.LE.1.0E-20) A3=1.0E-20
C     IF A3=A2=ALPSID AND F2 IS BEST, GO INVOKE SIDE CONSTRAINT
C     MODIFICATION.
      ALPB=1.-A2/A3
      ALPA=1.-ALPSID/A3
      JBEST=0
      IF (ABS(ALPB).LT.1.0E-10.AND.ABS(ALPA).LT.1.0E-10) JBEST=1
      IF (JBEST.EQ.1.AND.IBEST.EQ.2) GO TO 20
C     SIDE CONSTRAINT CHECK NOT SATISFIED.
      IF (NCON.EQ.0) GO TO 260
C     STORE CONSTRAINT VALUES IN G2.
      DO 250 I=1,NCON
      G2(I)=G(I)
250   CONTINUE
260   CONTINUE
C     IF A3=A2, SET A3=.9*A2.
      IF (ABS(ALPB).LT.1.0E-10) A3=.9*A2
C     MOVE AT LEAST .01*A2.
      IF (A3.LT.(.01*A2)) A3=.01*A2
```

```
C     LIMIT MOVE TO 5.*A2.
      IF (A3.GT.(5.*A2)) A3=5.*A2
C     LIMIT MOVE TO ALPSID.
      IF (A3.GT.ALPSID) A3=ALPSID
C     MOVE A DISTANCE A3*S.
      ALP=A3-A2
      ALPTOT=ALPTOT+ALP
      DO 270 I=1,NDV
      X(I)=X(I)+ALP*S(I)
270   CONTINUE
      IF (IPRINT.LT.5) GO TO 300
      WRITE (6,780)
      WRITE (6,740) A3
      IF (NSCAL.EQ.0) GO TO 290
      DO 280 I=1,NDV
280   G(I)=SCAL(I)*X(I)
      WRITE (6,750) (G(I),I=1,NDV)
      GO TO 300
290   WRITE (6,750) (X(I),I=1,NDV)
300   CONTINUE
C     ------------------------------------------------------------------
C              UPDATE FUNCTION AND CONSTRAINT VALUES
C     ------------------------------------------------------------------
      NCAL(1)=NCAL(1)+1
      JGOTO=2
      RETURN
310   CONTINUE
      F3=OBJ
      IF (IPRINT.GE.5) WRITE (6,760) F3
      IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 320
      WRITE (6,770)
      WRITE (6,750) (G(I),I=1,NCON)
320   CONTINUE
C     ------------------------------------------------------------------
C       CALCULATE MAXIMUM CONSTRAINT VIOLATION AND PICK BEST DESIGN
C     ------------------------------------------------------------------
      CV3=0.
      IGOOD3=0
      NVC1=0
      IF (NCON.EQ.0) GO TO 340
      DO 330 I=1,NCON
      CC=CTAM
      IF (ISC(I).GT.0) CC=CTBM
      C1=G(I)-CC
      IF (C1.GT.CV3) CV3=C1
      IF (C1.GT.0.) NVC1=NVC1+1
330   CONTINUE
      IF (CV3.GT.0.) IGOOD3=1
340   CONTINUE
C     DETERMINE BEST DESIGN.
```

```
      IF (IBEST.EQ.2) GO TO 360
C     CHOOSE BETWEEN F1 AND F3.
      IF (IGOOD1.EQ.0.AND.IGOOD3.EQ.0) GO TO 350
      IF (CV1.GE.CV3) IBEST=3
      GO TO 380
350   IF (F3.LE.F1) IBEST=3
      GO TO 380
360   CONTINUE
C     CHOOSE BETWEEN F2 AND F3.
      IF (IGOOD2.EQ.0.AND.IGOOD3.EQ.0) GO TO 370
      IF (CV2.GE.CV3) IBEST=3
      GO TO 380
370   IF (F3.LE.F2) IBEST=3
380   CONTINUE
      ALP=A3
      OBJ=F3
C     IF F3 VIOLATES FEWER CONSTRAINTS THAN F1 RETURN.
      IF (NVC1.LT.NVC) GO TO 710
C     IF OBJECTIVE AND ALL CONSTRAINTS ARE LINEAR, RETURN.
      IF (LINOBJ.NE.0.AND.NLNC.EQ.NCON) GO TO 710
C     IF A3 = ALPLN AND F3 IS BOTH GOOD AND BEST RETURN.
      ALPB=1.-ALPLN/A3
      IF ((ABS(ALPB).LT.1.0E-20.AND.IBEST.EQ.3).AND.(IGOOD3.EQ.0)) GO TO
     1 710
C     IF A3 = ALPSID AND F3 IS BEST, GO INVOKE SIDE CONSTRAINT
C     MODIFICATION.
      ALPA=1.-ALPSID/A3
      IF (ABS(ALPA).LT.1.0E-20.AND.IBEST.EQ.3) GO TO 20
C     ---------------------------------------------------------------
C     **********            3 - POINT INTERPOLATION          *********
C     ---------------------------------------------------------------
      ALPNC=ALPSID
      ALPCA=ALPSID
      ALPFES=-1.
      ALPMIN=-1.
      IF (NCON.EQ.0) GO TO 440
      III=0
390   III=III+1
      C1=G1(III)
      C2=G2(III)
      C3=G(III)
      IF (ISC(III).EQ.0) GO TO 400
C     ---------------------------------------------------------------
C     LINEAR CONSTRAINT.  FIND ALPFES ONLY.  ALPLN SAME AS BEFORE.
C     ---------------------------------------------------------------
      IF (C1.LE.CTBM) GO TO 430
      II=1
      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A3,C3,ZRO,ZRO)
      IF (ALP.GT.ALPFES) ALPFES=ALP
      GO TO 430
```

```
400   CONTINUE
C     ----------------------------------------------------------------
C                      NON-LINEAR CONSTRAINT
C     ----------------------------------------------------------------
      II=2
      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,A3,C3)
      IF (ALP.LE.ZRO) GO TO 430
      IF (C1.GE.CT.AND.C1.LE.0.) GO TO 410
      IF (C1.GT.CTAM.OR.C1.LT.0.) GO TO 420
C     ALP IS MINIMUM MOVE.  UPDATE FOR NEXT CONSTRAINT ENCOUNTER.
410   ALPA=ALP
      CALL CNMN07 (II,ALP,ALPA,ZRO,C1,A2,C2,A3,C3)
      IF (ALP.LT.ALPCA.AND.ALP.GE.ALPA) ALPCA=ALP
      GO TO 430
420   CONTINUE
      IF (ALP.GT.ALPFES.AND.C1.GT.CTAM) ALPFES=ALP
      IF (ALP.LT.ALPNC.AND.C1.LT.0.) ALPNC=ALP
430   CONTINUE
      IF (III.LT.NCON) GO TO 390
440   CONTINUE
      IF (LINOBJ.GT.0.OR.SLOPE.GT.0.) GO TO 450
C     ----------------------------------------------------------------
C     ----------------------------------------------------------------
C
720   FORMAT (/5X,25HTHREE-POINT INTERPOLATION)
730   FORMAT (/////58H* * * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATI
     1ON * * *)
740   FORMAT (//5X,15HPROPOSED DESIGN/5X,7HALPHA =,E12.5/5X,8HX-VECTOR)
750   FORMAT (1X,8E12.4)
760   FORMAT (/5X,5HOBJ =,E13.5)
770   FORMAT (/5X,17HCONSTRAINT VALUES)
780   FORMAT (/5X,23HTWO-POINT INTERPOLATION)
790   FORMAT (/5X,35H* * * END OF ONE-DIMENSIONAL SEARCH)
      END
```

```
      SUBROUTINE CNMN07 (II,XBAR,EPS,X1,Y1,X2,Y2,X3,Y3)
C     ROUTINE TO FIND FIRST XBAR.GE.EPS CORRESPONDING TO A REAL ZERO
C     OF A ONE-DIMENSIONAL FUNCTION BY POLYNOMIEL INTERPOLATION.
C     BY G. N. VANDERPLAATS                         APRIL, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
C     II = CALCULATION CONTROL.
C          1:  2-POINT LINEAR INTERPOLATION, GIVEN X1, Y1, X2 AND Y2.
C          2:  3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2,
C              X3 AND Y3.
C     EPS MAY BE NEGATIVE.
C     IF REQUIRED ZERO ON Y DOES NOT EXITS, OR THE FUNCTION IS
C     ILL-CONDITIONED, XBAR = EPS-1.0 WILL BE RETURNED AS AN ERROR
C     INDICATOR.
C     IF DESIRED INTERPOLATION IS ILL-CONDITIONED, A LOWER ORDER
C     INTERPOLATION, CONSISTANT WITH INPUT DATA, WILL BE ATTEMPTED AND
C     II WILL BE CHANGED ACCORDINGLY.
      XBAR1=EPS-1.
      XBAR=XBAR1
      JJ=0
      X21=X2-X1
      IF (ABS(X21).LT.1.0E-20) RETURN
      IF (II.EQ.2) GO TO 30
C
10    CONTINUE
C     ------------------------------------------------------------------
C                  II=1: 2-POINT LINEAR INTERPOLATION
C     ------------------------------------------------------------------
      II=1
      YY=Y1*Y2
      IF (JJ.EQ.0.OR.YY.LT.0.) GO TO 20
C     INTERPOLATE BETWEEN X2 AND X3.
      DY=Y3-Y2
      IF (ABS(DY).LT.1.0E-20) GO TO 20
      XBAR=X2+Y2*(X2-X3)/DY
      IF (XBAR.LT.EPS) XBAR=XBAR1
      RETURN
20    DY=Y2-Y1
C     INTERPOLATE BETWEEN X1 AND X2.
      IF (ABS(DY).LT.1.0E-20) RETURN
      XBAR=X1+Y1*(X1-X2)/DY
      IF (XBAR.LT.EPS) XBAR=XBAR1
      RETURN
30    CONTINUE
C     ------------------------------------------------------------------
C                 II=2: 3-POINT QUADRATIC INTERPOLATION
C     ------------------------------------------------------------------
      JJ=1
      X31=X3-X1
      X32=X3-X2
      QQ=X21*X31*X32
```

```
IF (ABS(QQ).LT.1.0E-20) RETURN
AA=(Y1*X32-Y2*X31+Y3*X21)/QQ
IF (ABS(AA).LT.1.0E-20) GO TO 10
BB=(Y2-Y1)/X21-AA*(X1+X2)
CC=Y1-X1*(AA*X1+BB)
BAC=BB*BB-4.*AA*CC
IF (BAC.LT.0.) GO TO 10
BAC=SQRT(BAC)
AA=.5/AA
XBAR=AA*(BAC-BB)
XB2=-AA*(BAC+BB)
IF (XBAR.LT.EPS) XBAR=XB2
IF (XB2.LT.XBAR.AND.XB2.GT.EPS) XBAR=XB2
IF (XBAR.LT.EPS) XBAR=XBAR1
RETURN
END
```

```
      SUBROUTINE CNMN08 (NDB,NER,C,MS1,B,N3,N4,N5)
      DIMENSION C(N4),MS1(N5),B(N3,N3)
C     ROUTINE TO SOLVE SPECIAL LINEAR PROBLEM FOR IMPOSING S-TRANSPOSE
C     TIMES S.LE.1 BOUNDS IN THE MODIFIED METHOD OF FEASIBLE DIRECTIONS.
C     BY G. N. VANDERPLAATS                          APRIL, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
C     REF.  'STRUCTURAL OPTIMIZATION BY METHODS OF FEASIBLE DIRECTIONS',
C     G. N. VANDERPLAATS AND F. MOSES, JOURNAL OF COMPUTERS
C     AND STRUCTURES, VOL 3, PP 739-755, 1973.
C     FORM OF L. P. IS BX=C WHERE 1ST NDB COMPONENTS OF X CONTAIN VECTOR
C     U AND LAST NDB COMPONENTS CONTAIN VECTOR V.  CONSTRAINTS ARE
C     U.GE.0, V.GE.0, AND U-TRANSPOSE TIMES V = 0.
C     NER = ERROR FLAG.  IF NER.NE.0 ON RETURN, PROCESS HAS NOT
C     CONVERGED IN 5*NDB ITERATIONS.
C     VECTOR MS1 IDENTIFIES THE SET OF BASIC VARIABLES.
C     ------------------------------------------------------------------
C     CHOOSE INITIAL BASIC VARIABLES AS V, AND INITIALIZE VECTOR MS1
C     ------------------------------------------------------------------
      NER=1
      M2=2*NDB
C     CALCULATE CBMIN AND EPS AND INITIALIZE MS1.
      EPS=-1.0E+10
      CBMIN=0.
      DO 10 I=1,NDB
      BI=B(I,I)
      CBMAX=0.
      IF (BI.LT.-1.0E-6) CBMAX=C(I)/BI
      IF (BI.GT.EPS) EPS=BI
      IF (CBMAX.GT.CBMIN) CBMIN=CBMAX
10    MS1(I)=0
      EPS=.0001*EPS
      IF (EPS.LT.-1.0E-10) EPS=-1.0E-10
      IF (EPS.GT.-.0001) EPS=-.0001
      CBMIN=CBMIN*1.0E-6
      IF (CBMIN.LT.1.0E-10) CBMIN=1.0E-10
      ITER1=0
      NMAX=5*NDB
C     ------------------------------------------------------------------
C     **********              BEGIN NEW ITERATION             **********
C     ------------------------------------------------------------------
20    ITER1=ITER1+1
      IF (ITER1.GT.NMAX) RETURN
C     FIND MAX. C(I)/B(I,I) FOR I=1,NDB.
      CBMAX=.9*CBMIN
      ICHK=0
      DO 30 I=1,NDB
      C1=C(I)
      BI=B(I,I)
      IF (BI.GT.EPS.OR.C1.GT.0.) GO TO 30
      CB=C1/BI
```

```
      IF (CB.LE.CBMAX) GO TO 30
      ICHK=I
      CBMAX=CB
30    CONTINUE
      IF (CBMAX.LT.CBMIN) GO TO 70
      IF (ICHK.EQ.0) GO TO 70
C     UPDATE VECTOR MS1.
      JJ=ICHK
      IF (MS1(JJ).EQ.0) JJ=ICHK+NDB
      KK=JJ+NDB
      IF (KK.GT.M2) KK=JJ-NDB
      MS1(KK)=ICHK
      MS1(JJ)=0
C     ------------------------------------------------------------------
C                      PIVOT OF B(ICHK,ICHK)
C     ------------------------------------------------------------------
      BB=1./B(ICHK,ICHK)
      DO 40 J=1,NDB
40    B(ICHK,J)=BB*B(ICHK,J)
      C(ICHK)=CBMAX
      B(ICHK,ICHK)=BB
C     ELIMINATE COEFICIENTS ON VARIABLE ENTERING BASIS AND STORE
C     COEFICIENTS ON VARIABLE LEAVING BASIS IN THEIR PLACE.
      DO 60 I=1,NDB
      IF (I.EQ.ICHK) GO TO 60
      BB1=B(I,ICHK)
      B(I,ICHK)=0.
      DO 50 J=1,NDB
50    B(I,J)=B(I,J)-BB1*B(ICHK,J)
      C(I)=C(I)-BB1*CBMAX
60    CONTINUE
      GO TO 20
70    CONTINUE
      NER=0
C     ------------------------------------------------------------------
C     STORE ONLY COMPONENTS OF U-VECTOR IN 'C'.  USE B(I,1) FOR
C     TEMPORARY STORAGE
C     ------------------------------------------------------------------
      DO 80 I=1,NDB
      B(I,1)=C(I)
80    CONTINUE
      DO 90 I=1,NDB
      C(I)=0.
      J=MS1(I)
      IF (J.GT.0) C(I)=B(J,1)
      IF (C(I).LT.0.) C(I)=0.
90    CONTINUE
      RETURN
      END
```

SUBROUTINF ANALIZ - LASER TURRET ANALYSIS SEPT. 77

```
      SUBROUTINF ANALIZ(ICALC)
C     ROUTINE TO PERFORM LASER TURRET ANALYSIS IN SUBSONIC AND
C     SUPERSONIC FLOW.
C     BY G. N. VANDERPLAATS                          4ULY, 1977.
C     NAVAL POSTGRADUATS
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
      COMMON /GLOBCM/ ABAR(20),ACL,AKPRIM,AL,AMACHI(30),BBAR(20),DENRTO,
     * DENGAM,EPS,EPSM,GAMMAI(30),PHII(30),RFUS,SLOPEX(30),SUMPD2,
     * TDENRT,THMAX,WAVEL,WGHTI(30),XM
      COMMON /CMLOC/ ETAI(16),MAXK,MAXP,NBEAM,NETAI,NRBI,NTHBC,NXBC,
     * RBI(10),TITLE(20),YYPXBC(10,3),YYPTBC(10,3)
      COMMON /CMLOC2/AMX(10,15),BMX(10,15),ANT(10,15)
      DIMENSION T(10),AN(10),BN(10),PDISTI(200)
C     FOURIER EXPANSION.
      NMAX=10
      MMAX=10
C     OPTICAL PATH LENGTH.
      KTRAP=3
      B=4.
      NPRINT=0
      IF (ICALC.GT.1) GO TO 10
      CALL TINPUT
C     CALCULATE FOURIER COEFFICIENTS.
      CALL FCOEF(AL,ACL,THMAX,AN,BN,MAXK,MAXP,NMAX,MMAX)
      RETURN
   10 CONTINUE
      YYPXBC(1,2)=EPS
      YYPTBC(1,2)=EPS
      IPRINT=0
      IF (ICALC.EQ.3.OR.NPRINT.GT.0) IPRINT=1
      IPLOT=0
      IF (ICALC.EQ.3) IPLOT=1
      SUMPD2=0.
C     BOUNDARY CONDITIONS.
C     X-DIRECTION.
      NSYM=0
      AMULT=EPS*BBAR(1)
      CALL BCOND (NSYM,NXBC,YYPXBC,ABAR,MAXK,AL,AMULT)
C     THETA-DIRECTION.
      NSYM=1
      AMULT=EPS*ABAR(1)
      CALL BCOND (NSYM,NTHBC,YYPTBC,BBAR,MAXP,THMAX,AMULT)
      DO 30 IBEAM=1,NBEAM
      AMACH=AMACHI(IBEAM)
      CALL PHDIST (X,R,THETA,EPSM,XM,PHI,GAMMA,RHO,Y,Z,PHIPP,U,V,CP,ABAR
     1,BBAR,AL,ACL,THMAX,EPS,RINDEX,RB,ETA,AN,BN,MAXK,MAXP,NMAX,MMAX,KTR
     2AP,A,B,T,DELOPL,IBEAM,REFOPL,WAVEL,RFUS,ETAI,RBI,GAMMAI,PHII,NETAI
     3,NRBI,TDENRT,PDISTI,DENGAM,AMACH,DENRTO,AKPRIM,IPRINT,IPLOT)
C     SUM OF SQUARES OF PHASE DISTORTION.
      NN=NRBI*NETAI
```

```
      SMP1=0.
      DO 20 I=1,NN
20    SMP1=SMP1+PDISTI(I)**2
      SUMPD2=SUMPD2+WGHTI(IBEAM)*SMP1
   30 CONTINUE
      THETA=0.
      N=20
      XMAX=2.*AL
      XMIN=-XMAX
      R=0.
      IF(IPRINT.EQ.0) GO TO 50
      DO 60 I=1,NBEAM
      AMACH=AMACHI(I)
      IF(I.EQ.1) GO TO 80
      IM1=I-1
      DO 70 J=1,IM1
      DMACH=AMACHI(J)-AMACH
      IF(ABS(DMACH).LT.0.001) GO TO 60
70    CONTINUE
80    CONTINUE
      CALL CPPRNT(THETA,AMACH,AL,ACL,THMAX,MAXK,MAXP,NMAX,MMAX,ABAR,
     * BBAR,EPS,AN,BN,N,XMIN,XMAX,R,DENGAM)
60    CONTINUE
      CALL SURPRT(ABAR,BBAR,MAXK,MAXP,EPS,AL,THMAX)
      WRITE(6,40)SUMPD2
C     CALCULATE TURRET SLOPE AT 30 POINTS.
50    NVAL=30
      AMULT=EPS*BBAR(1)
      CALL SLOPE (MAXK,ABAR,AL,SLOPEX,NVAL,AMULT)
      RETURN
C
   40 FORMAT (///5X,36HSUM OF SQUARES OF PHASE DISTORTION =,E12.5)
      END
```

SUBROUTINE BCOND SEPT. 77

```
      SUBROUTINE BCOND (NSYM,NBC,YYPBC,ABAR,MAXE,XREF,AMULTS)
      DIMENSION YYPBC(10,3),ABAR(1),A(10,10),B(10)
C     ROUTINE TO IMPOSE POLYNOMIAL BOUNDARY CONDITIONS.
C     THE FIRST NBCT COEFICIENTS OF ABAR ARE CALCULATED WHERE NBCT IS
C     THE TOTAL NUMBER OF B. C. S.
C     TOTAL NUMBER OF BOUNDARY CONDITIONS.
      NBCT=0
      DO 10 I=1,NBC
      IF (ABS(YYPBC(I,2)).LT.100.) NBCT=NBCT+1
      IF (ABS(YYPBC(I,3)).LT.100.) NBCT=NBCT+1
   10 CONTINUE
      IF (NBCT.LE.0) RETURN
      MAXE1=MAXE+1
C     IMPOSE SYMMETRY IF REQUIRED.
      NSYM1=1
      IF (NSYM.EQ.0) GO TO 30
      NSYM1=2
      DO 20 I=2,MAXE1,2
   20 ABAR(I)=0.
   30 CONTINUE
C     NUMBER OF COEFICIENTS ELIMINATED.
      N1=NBCT*NSYM1
C     SET UP COEFFICIENT MATRIX AND RHS.
      N=0
      JJ=NSYM1+1
      DO 70 I=1,NBC
      X=YYPBC(I,1)*XREF
      IF (ABS(YYPBC(I,2)).GE.100.) GO TO 50
C     Y BOUNDARY CONDITION.
      N=N+1
      B(N)=YYPBC(I,2)/AMULTS
      L=1
      AA=1.
      DO 40 J=1,MAXE1,NSYM1
      IF (J.GT.N1) B(N)=B(N)-ABAR(J)*AA
      IF (J.LE.N1) A(N,L)=AA
      L=L+1
      AA=AA*X
      IF (NSYM1.EQ.2) AA=AA*X
   40 CONTINUE
   50 CONTINUE
      IF (ABS(YYPBC(I,3)).GE.100.) GO TO 70
C     Y-PRIME BOUNDARY CONDITION.
      N=N+1
      B(N)=YYPBC(I,3)/AMULTS
      L=2
      A(N,1)=0.
      AA=1.
      IF (NSYM1.EQ.2) AA=X
      DO 60 J=JJ,MAXE1,NSYM1
```

```
      BB=FLOAT(J)-1.
      IF (J.GT.N1) B(N)=B(N)-ABAR(J)*BB*AA
      IF (J.LE.N1) A(N,L)=AA*BB
      L=L+1
      AA=AA*X
      IF (NSYM1.EQ.2) AA=AA*X
   60 CONTINUE
   70 CONTINUE
C     DETERMINE COEFICIENTS.
      M1=10
      M2=10
      M3=10
      M4=1
      NLC=1
      CALL GELIM2 (A,B,N,NLC,M1,M2,M3,M4,NER)
C     STORE RESULTS IN ABAR.
      J=1-NSYM1
      DO 80 I=1,N
      J=J+NSYM1
   80 ABAR(J)=B(I)
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE BESJ
C
C        PURPOSE
C           COMPUTE THE J BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C
C        USAGE
C           CALL BESJ(X,N,BJ,D,IER)
C
C        DESCRIPTION OF PARAMETERS
C           X  -THE ARGUMENT OF THE J BESSEL FUNCTION DESIRED
C           N  -THE ORDER OF THE J BESSEL FUNCTION DESIRED
C           BJ -THE RESULTANT J BESSEL FUNCTION
C           D  -REQUIRED ACCURACY
C           IER-RESULTANT ERROR CODE WHERE
C              IER=0  NO ERROR
C              IER=1  N IS NEGATIVE
C              IER=2  X IS NEGATIVE OR ZERO
C              IER=3  REQUIRED ACCURACY NOT OBTAINED
C              IER=4  RANGE OF N COMPARED TO X NOT CORRECT (SEE REMARKS)
C
C        REMARKS
C           N MUST BE GREATER THAN OR EQUAL TO ZERO, BUT IT MUST BE
C           LESS THAN
C              20+10*X-X** 2/3   FOR X LESS THAN OR EQUAL TO 15
C              90+X/2            FOR X GREATER THAN 15
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           RECURRENCE RELATION TECHNIQUE DESCRIBED BY H. GOLDSTEIN AND
C           R.M. THALER,'RECURRENCE TECHNIQUES FOR THE CALCULATION OF
C           BESSEL FUNCTIONS',M.T.A.C.,V.13,PP.102-108 AND I.A. STEGUN
C           AND M. ABRAMOWITZ,'GENERATION OF BESSEL FUNCTIONS ON HIGH
C           SPEED COMPUTERS',M.T.A.C.,V.11,1957,PP.255-257
C
C     ..................................................................
C
      SUBROUTINE BESJ(X,N,BJ,D,IER)
C
      BJ=.0
      IF(N)10,20,20
   10 IER=1
      RETURN
   20 IF(X)30,30,31
   30 IER=2
      RETURN
```

```
 31 IF(X-15.)32,32,34
 32 NTEST=20.+10.*X-X** 2/3
    GO TO 36
 34 NTEST=90.+X/2.
 36 IF(N-NTEST)40,38,38
 38 IER=4
    RETURN
 40 IER=0
    N1=N+1
    BPREV=.0
C
C   COMPUTE STARTING VALUE OF M
C
    IF(X-5.)50,60,60
 50 MA=X+6.
    GO TO 70
 60 MA=1.4*X+60./X
 70 MB=N+IFIX(X)/4+2
    MZERO=MAX0(MA,MB)
C
C   SET UPPER LIMIT OF M
C
    MMAX=NTEST
100 DO 190 M=MZERO,MMAX,3
C
C   SET F(M),F(M-1)
C
    FM1=1.0E-28
    FM=.0
    ALPHA=.0
    IF(M-(M/2)*2)120,110,120
110 JT=-1
    GO TO 130
120 JT=1
130 M2=M-2
    DO 160 K=1,M2
    MK=M-K
    BMK=2.*FLOAT(MK)*FM1/X-FM
    FM=FM1
    FM1=BMK
    IF(MK-N-1)150,140,150
140 BJ=BMK
150 JT=-JT
    S=1+JT
160 ALPHA=ALPHA+BMK*S
    BMK=2.*FM1/X-FM
    IF(N)180,170,180
170 BJ=BMK
180 ALPHA=ALPHA+BMK
    BJ=BJ/ALPHA
```

```
    IF(ABS(BJ-BPREV)-ABS(D*BJ))200,200,190
190 BPREV=BJ
    IER=3
200 RETURN
    END
```

```
C       SUBROUTINE BESK
C
C          COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C
C       USAGE
C          CALL BESK(X,N,BK,IER)
C
C       DESCRIPTION OF PARAMETERS
C          X   -THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED
C          N   -THE ORDER OF THE K BESSEL FUNCTION DESIRED
C          BK  -THE RESULTANT K BESSEL FUNCTION
C          IER-RESULTANT ERROR CODE WHERE
C             IER=0  NO ERROR
C             IER=1  N IS NEGATIVE
C             IER=2  X IS ZERO OR NEGATIVE
C             IER=3  X .GT. 170, MACHINE RANGE EXCEEDED
C             IER=4  BK .GT. 10**70
C
C       REMARKS
C          N MUST BE GREATER THAN OR EQUAL TO ZERO
C
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C          NONE
C
C       METHOD
C          COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING
C          SERIES APPROXIMATIONS AND THEN COMPUTES N TH ORDER FUNCTION
C          USING RECURRENCE RELATION.
C          RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
C          AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS
C          TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
C          FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON,
C          'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
C          UNIVERSITY PRESS, 1958, P. 62
C
C     ..................................................................
C
      SUBROUTINE BESK (X,N,BK,IER)
      DIMENSION T(12)
      BK=.0
      IF (N) 10,20,20
   10 IER=1
      RETURN
   20 IF (X) 30,30,40
   30 IER=2
      RETURN
   40 IF (X-170.0) 60,60,50
   50 IER=3
      RETURN
   60 IER=0
```

```
      IF (X-1.) 180,180,70
   70 A=EXP(-X)
      B=1./X
      C=SQRT(B)
      T(1)=B
      DO 80 L=2,12
   80 T(L)=T(L-1)*B
      IF (N-1) 90,110,90
C
C     COMPUTE K0 USING POLYNOMIAL APPROXIMATION
C
   90 G0=A*(1.2533141-.1566642*T(1)+.08811128*T(2)-.09139095*T(3)+.13445
     196*T(4)-.2299850*T(5)+.3792410*T(6)-.5247277*T(7)+.5575368*T(8)-.4
     2262633*T(9)+.2184518*T(10)-.06680977*T(11)+.009189383*T(12))*C
      IF (N) 40,100,110
  100 BK=G0
      RETURN
C
C     COMPUTE K1 USING POLYNOMIAL APPROXIMATION
C
  110 G1=A*(1.2533141+.4699927*T(1)-.1468583*T(2)+.1280427*T(3)-.1736432
     1*T(4)+.2847618*T(5)-.4594342*T(6)+.6283381*T(7)-.6632295*T(8)+.505
     20239*T(9)-.2581304*T(10)+.07880001*T(11)-.01082418*T(12))*C
      IF (N-1) 40,120,130
  120 BK=G1
      RETURN
C
C     FROM K0,K1 COMPUTE KN USING RECURRENCE RELATION
C
  130 DO 160 J=2,N
      GJ=2.*(FLOAT(J)-1.)*G1/X+G0
      IF (GJ-1.0E70) 150,150,140
  140 IER=4
      GO TO 170
  150 G0=G1
  160 G1=GJ
  170 BK=GJ
      RETURN
  180 B=X/2.
      A=.5772157+ALOG(B)
      C=B*B
      IF (N-1) 190,220,190
C
C     COMPUTE K0 USING SERIES EXPANSION
C
  190 G0=-A
      X2J=1.
      FACT=1.
      HJ=.0
      DO 200 J=1,6
```

```
      SUBROUTINE BESK

      RJ=1./FLOAT(J)
      X2J=X2J*C
      FACT=FACT*RJ*RJ
      HJ=HJ+RJ
  200 G0=G0+X2J*FACT*(HJ-A)
      IF (N) 220,210,220
  210 BK=G0
      RETURN
C
C     COMPUTE K1 USING SERIES EXPANSION
C
  220 X2J=B
      FACT=1.
      HJ=1.
      G1=1./X+X2J*(.5+A-HJ)
      DO 230 J=2,8
      X2J=X2J*C
      RJ=1./FLOAT(J)
      FACT=FACT*RJ*RJ
      HJ=HJ+RJ
  230 G1=G1+X2J*FACT*(.5+(A-HJ)*FLOAT(J))
      IF (N-1) 130,240,130
  240 BK=G1
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE BESY
C
C        PURPOSE
C           COMPUTE THE Y BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C
C        USAGE
C           CALL BESY(X,N,BY,IER)
C
C        DESCRIPTION OF PARAMETERS
C           X  -THE ARGUMENT OF THE Y BESSEL FUNCTION DESIRED
C           N  -THE ORDER OF THE Y BESSEL FUNCTION DESIRED
C           BY -THE RESULTANT Y BESSEL FUNCTION
C           IER-RESULTANT ERROR CODE WHERE
C              IER=0  NO ERROR
C              IER=1  N IS NEGATIVE
C              IER=2  X IS NEGATIVE OR ZERO
C              IER=3  BY HAS EXCEEDED MAGNITUDE OF 10**70
C
C        REMARKS
C           VERY SMALL VALUES OF X MAY CAUSE THE RANGE OF THE LIBRARY
C           FUNCTION ALOG TO BE EXCEEDED
C           X MUST BE GREATER THAN ZERO
C           N MUST BE GREATER THAN OR EQUAL TO ZERO
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
C           AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS
C           TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
C           FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON,
C           'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
C           UNIVERSITY PRESS, 1958, P. 62
C
C     ..................................................................
C
      SUBROUTINE BESY(X,N,BY,IER)
C
C     CHECK FOR ERRORS IN N AND X
C
      IF(N)180,10,10
   10 IER=0
      IF(X)190,190,20
C
C     BRANCH IF X LESS THAN OR EQUAL 4
C
```

```
   20 IF(X-4.0)40,40,30
C
C        COMPUTE Y0 AND Y1 FOR X GREATER THAN 4
C
   30 T1=4.0/X
      T2=T1*T1
      P0=((((-.0000037043*T2+.0000173565)*T2-.0000487613)*T2
     1  +.00017343)*T2-.001753062)*T2+.3989423
      Q0=((((.0000032312*T2-.0000142078)*T2+.0000342468)*T2
     1  -.0000869791)*T2+.0004564324)*T2-.01246694
      P1=((((.0000042414*T2-.0000200920)*T2+.0000580759)*T2
     1  -.00022320S)*T2+.002921826)*T2+.3989423
      Q1=((((-.0000036594*T2+.00001622)*T2-.0000398708)*T2
     1  +.0001064741)*T2-.0006390400)*T2+.03740084
      A=2.0/SQRT(X)
      B=A*T1
      C=X-.7853982
      Y0=A*P0*SIN(C)+B*Q0*COS(C)
      Y1=-A*P1*COS(C)+B*Q1*SIN(C)
      GO TO 90
C
C        COMPUTE Y0 AND Y1 FOR X LESS THAN OR EQUAL TO 4
C
   40 XX=X/2.
      X2=XX*XX
      T=ALOG(XX)+.5772157
      SUM=0.
      TERM=T
      Y0=T
      DO 70 L=1,15
      IF(L-1)50,60,50
   50 SUM=SUM+1./FLOAT(L-1)
   60 FL=L
      TS=T-SUM
      TERM=(TERM*(-X2)/FL**2)*(1.-1./(FL*TS))
   70 Y0=Y0+TERM
      TERM = XX*(T-.5)
      SUM=0.
      Y1=TERM
      DO 80 L=2,16
      SUM=SUM+1./FLOAT(L-1)
      FL=L
      FL1=FL-1.
      TS=T-SUM
      TERM=(TERM*(-X2)/(FL1*FL))*((TS-.5/FL)/(TS+.5/FL1))
   80 Y1=Y1+TERM
      PI2=.6366198
      Y0=PI2*Y0
      Y1=-PI2/X+PI2*Y1
C
```

```
C     CHECK IF ONLY Y0 OR Y1 IS DESIRED
C
   90 IF(N-1)100,100,130
C
C     RETURN EITHER Y0 OR Y1 AS REQUIRED
C
  100 IF(N)110,120,110
  110 BY=Y1
      GO TO 170
  120 BY=Y0
      GO TO 170
C
C    PERFORM RECURRENCE OPERATIONS TO FIND YN(X)
C
  130 YA=Y0
      YB=Y1
      K=1
  140 T=FLOAT(2*K)/X
      YC=T*YB-YA
      IF(ABS(YC)-1.0E70)145,145,141
  141 IER=3
      RETURN
  145 K=K+1
      IF(K-N)150,160,150
  150 YA=YB
      YB=YC
      GO TO 140
  160 BY=YC
  170 RETURN
  180 IER=1
      RETURN
  190 IER=2
      RETURN
      END
```

```
      SUBROUTINE CPPRNT (THETA,AMACH,AL,ACL,THMAX,MAXK,MAXP,NMAX,MMAX,AB
     1AR,BBAR,EPS,AN,BN,N,XMIN,XMAX,R,DENGAM)
      DIMENSION ABAR(1),BBAR(1),AN(1),BN(1)
C     ROUTINE TO PRINT PHI,UMV,CP AT N+1 LOCATIONS ALONG X FOR SPECIFIED
C     THETA
C     IF R = 0 IS INPUT, R IS CALCULATED AS TURRET SURFACE.
C     IF R.GT.0 IS INPUT, THAT R IS USED IN CALCULATIONS.
      IR=0
      IF(R.GT.0.) IR=1
      WRITE (6,20) THETA,AMACH
      DX=(XMAX-XMIN)/FLOAT(N)
      X=XMIN-DX
      NP1=N+1
      DO 10 I=1,NP1
      X=X+DX
      IF(IR.EQ.0) CALL RSURF(ABAR,BBAR,EPS,MAXK,MAXP,X,THETA,AL,THMAX,R)
      CALL PHIUV (X,THETA,R,AMACH,AL,ACL,THMAX,MAXK,MAXP,NMAX,MMAX,ABAR,
     1BBAR,EPS,AN,BN,PHI,U,V)
      CP=-2.*U-V**2
10    WRITE(6,30)X,R,PHI,U,V,CP
C     CRITICAL PRESSURE COEFFICIENT.
      CPSTAR=2.*(1.+.5*(DENGAM-1.)*AMACH*AMACH)/( DENGAM +1.)
      EX1=DENGAM/(DENGAM-1.)
      CPSTAR=2.*(CPSTAR**EX1-1.)/(DENGAM*AMACH*AMACH)
      WRITE(6,40)CPSTAR
40    FORMAT(/5X,42HCRITICAL PRESSURE COEFFICIENT ON SURFACE =,F10.5)
      RETURN
C
20    FORMAT(////5X,22HFLOW FIELD FOR THETA =,F7.3,8H DEGREES//
     * 5X,22HMACH NUMBER           =,F7.3//10X,1HX,
     110X,1HR,9X,3HPHI,11X,1HU,11X,1HV,10X,2HCP)
30    FORMAT(5X,6E11.4)
      END
```

SUBROUTINE DOPL SEPT. 77

```
      SUBROUTINE DOPL (X,R,THETA,EPSM,XM,PHI,GAMMA,RHO,Y,Z,PHIPP,U,V,CP,
     1ABAR,BBAR,AL,ACL,THMAX,EPS,RINDEX,RB,ETA,AN,BN,MAXK,MAXP,NMAX,MMAX
     2,KTRAP,A,B,T,DELOPL,TDENRT,DENGAM,AMACH,DENRTO,AKPRIM,DELPLA)
      DIMENSION ABAR(1),BBAR(1),AN(1),BN(1),T(1)
C     ROUTINE TO CALCULATE CHANGE IN OPTICAL PATH LENGTH BY INTEGRATING
C     THE INDEX OF REFRACTION - 1.0 FROM 0.0 TO A AND A TO B.
C     BY G. N. VANDERPLAATS                          NOV., 1976.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C
C     INTEGRATE FROM ZERO TO A FOR CONSTANT PRESSURE.  DENSITY
C     RATIO = TDENRT.
      DELOPL=AKPRIM*TDENRT*A
      DELPLA=DELOPL
C     KTRAP = MAX. NUMBER OF TRAPEZOIDAL SOLUTIONS.  MAX NO. OF INTERVAL
C     IS 2**(KTRAP-1)
      N2=1
      DO 30 K=1,KTRAP
      IGOTO=0
   10 CALL TRAP2N (IGOTO,A,B,N2,RHO,RINDEX)
      IF (IGOTO.EQ.0) GO TO 20
C     INDEX OF REFRACTION -1.
      CALL REFIND (X,R,THETA,EPSM,XM,PHI,GAMMA,RHO,Y,Z,PHIPP,U,V,CP,ABAR
     1,BBAR,AL,ACL,THMAX,EPS,RINDEX,RB,ETA,AN,BN,MAXK,MAXP,NMAX,MMAX,DEN
     2GAM,AMACH,DENRTO,AKPRIM)
      GO TO 10
   20 T(K)=RINDEX
   30 N2=2*N2
C     ROMBERG INTEGRATION.
      K1=1
      CALL RMBINT (T,KTRAP,K1)
      DELOPL=DELOPL+T(1)
      RETURN
      END
```

SUBROUTINE FCOEF   SEPT. 77

```
      SUBROUTINE FCOEF(AL,ACL,THMAX,AN,BN,MAXK,MAXP,NMAX,MMAX)
      COMMON /CMLOC2/ AMX(10,15),BMX(10,15),ANT(10,15)
      DIMENSION AN(1),BN(1)
C     ROUTINE TO CALCULATE FOURIER COEFFICIENTS FOR EXPANSION OF
C     POLYNOMIAL SURFACE IN X AND THETA.
C     BY G. N. VANDERPLAATS                              MAY, 1977.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C     COEFFICIENTS ON X.
      MAXKP1=MAXK+1
      DO 10 M=1,MMAX
      CALL FXTOK(M,MAXK,AL,ACL,AN,BN)
      DO 20 I=1,MAXKP1
      AMX(I,M)=AN(I)
20    BMX(I,M)=BN(I)
10    CONTINUE
C     COEFFICIENTS ON THETA.
      MAXPP1=MAXP+1
      PI=3.1415927
      NMAXP1=NMAX+1
      DO 30 NP1=1,NMAXP1
      N=NP1-1
      CALL FXTOK(N,MAXP,THMAX,PI,AN,BN)
      DO 40 I=1,MAXPP1
40    ANT(I,NP1)=AN(I)
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FXTOK (N,K,X1,X2,AN,BN)
      DIMENSION AN(1),BN(1)
C     ROUTINE TO CALCULATE THE NTH FOURIER COEFICIENTS FOR THE
C     EXPANSION OF 1, X, X**2, . . . X**K.
C     FORM OF FOURIER SERIES IS
C         Y = SUM (AN(K+1)*COS(NX) + BN(K+1)*SIN(NX)), N = 0,1,2.. INF.
C     BY G. N. VANDERPLAATS                          OCT. 22, 1976
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C
C  INPUT.
C     N - DESIRED FOURIER COEFICIENT.
C     K - HIGEST ORDER EXPONENT ON X FOR WHICH AN AND BN ARE REQUIRED.
C     X1 - 1/2 INTERVAL OVER WHICH X**K IS EXPANDED.
C     X2 - 1/2 SPACING BETWEEN EXPANSIONS.
C  OUTPUT.
C     AN - VECTOR OF A-COEFFICIENTS FOR FOURIER EXPANSION.  THE
C          THE COEFICIENT FOR X**I IS STORED IN THE I+1 LOCATION OF AN,
C        FOR I=0, 1, 2, . . . K.
C     BN - VECTOR OF B-COEFFICIENTS FOR FOURIER EXPANSION.  THE
C          COEFFICIENT FOR X**I IS STORED IN THE I+1 LOCATION OF BN,
C        FOR I=0, 1, 2, . . . K.
C  NOTE - ALTHOUGH ONLY THE COEFICIENTS FOR X**K MAY BE REQUIRED, THE
C     COEFICIENTS FOR EXPANSION ON 1, X, X**2, . . X**(K-1) ARE
C     ALSO PROVIDED SINCE THESE ARE OBTAINED AS A CONSEQUENCE OF
C     CALCULATING THE REQUIRED INFORMATION.
C
C     CONSTANTS.
      PI=3.1415927
      KMP1=K+1
      IF (N.GT.0) GO TO 20
C  SPECIAL CASE.  N = 0.
C     A(N,K) AND B(N,K) ARE THE FOURIER COEFICIENTS A-SUB-N AND B-SUB-N
C     RESPECTIVELY FOR THE EXPANSION X**K, K = 0, 1, . . .
C     A(0,K) = .5*(X1**(K+1))*(1+(-1)**K)/(X2*(K+1))
C     B(0,K) = 0
      SIGN=-1.
      C1=.5/X2
      DO 10 KP1=1,KMP1
      C1=C1*X1
      AN(KP1)=C1*(1.-SIGN)/FLOAT(KP1)
      SIGN=-SIGN
   10 BN(KP1)=0.
      RETURN
C  GENERAL CASE.  N.GT.0.
C     A(N,K) = (X1**K)*(1+(-1)**K)*SIN(N*PI*X1/X2)/(N*PI) -
C              (K*X2/(N*PI))*B(N,K-1)
C     B(N,K) = (X1**K)*(-1+(-1)**K)*COS(N*PI*X1/X2)/(N*PI) +
C              (K*X2/(N*PI))*A(N,K-1)
C   WHERE A(N,-1) = B(N,-1) = 0
C         PI = 3.1415927
```

```
C     SOLUTION BEGINS WITH K = 0 AND USES THE ABOVE RECURSION FORMULAS
C     TO CALCULATE A(N,K) AND B(N,K).
C
C     CONSTANTS.
   20 ANPI=FLOAT(N)*PI
      ANPIX=ANPI*X1/X2
      SN1=SIN(ANPIX)/ANPI
      CS1=COS(ANPIX)/ANPI
C     K = 0.
      AN(1)=2.*SN1
      BN(1)=0.
      IF (K.EQ.0) RETURN
C     K = 1, 2, . . K
      SIGN=-1.
      CC=X2/ANPI
      C1=1.
      DO 30 KN=2,KMP1
      K=KN-1
      C1=C1*X1
      C2=FLOAT(K)*CC
      AN(KN)=C1*(1.+SIGN)*SN1-C2*BN(K)
      BN(KN)=C1*(SIGN-1.)*CS1+C2*AN(K)
   30 SIGN=-SIGN
      RETURN
      END
```

SUBROUTINE FXY34 SEPT. 77

```
      SUBROUTINE FXY34(N,X,Y,Z,NER)
      DIMENSION X(1),Y(1),Z(1),AA(4,4)
C     ROUTINE TO CALCULATE THE COEFFICIENTS OF A POLYNOMIAL
C     FUNCTION OF Z IN X AND Y.
C     BY G. N. VANDERPLAATS                        MAY, 1977.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C --INPUT.
C     N - NUMBER OF INTERPOLATION POINTS (N = 3 OR 4).
C     X,Y - X AND Y COORDINATES, I=1,N.
C     Z - Z = F(X,Y) = FUNCTION VALUES.
C       Z IS DESTROYED.
C --OUTPUT.
C     Z - POLYNOMIAL COEFFICIENTS.
C       IF N = 3, Y = Z(1) + Z(2)*X + Z(3)*Y.
C       IF N = 4, Y = Z(1) + Z(2)*X + Z(3)*Y + Z(4)*X*Y.
C     NER - ERROR INDICATOR, 0 = NO ERROR.  NER.GT.0 = ERROR DUE TO
C             TWO X,Y POINTS ARE THE SAME OR THREE X,Y POINTS ARE
C             COLINEAR.
C
C     DIMENSION OF AA MATRIX AND NUMBER OF RHS VECTORS FOR EQUATIONS.
      NDIM=4
      NRHS=1
C     INSURE N = 3 OR 4.
      IF(N.LT.3) N=3
      IF(N.GT.4) N=4
C     SET UP COEFFICIENT MATRIX FOR SIMULTANEOUS EQUATION SOLUTION.
      DO 10 I=1,N
      AA(I,1)=1.
      AA(I,2)=X(I)
      AA(I,3)=Y(I)
10    AA(I,4)=X(I)*Y(I)
C     SOLVE EQUATIONS.
      CALL GELIM2(AA,Z,N,NRHS,NDIM,NDIM,NDIM,NRHS,NER)
      IF(N.EQ.3) Z(4)=0.
      RETURN
      END
```

SUBROUTINE GELIM2 SEPT. 77

```
      SUBROUTINE GELIM2 (A,B,N,NLC,M1,M2,M3,M4,NER)
      DIMENSION A(M1,M2),B(M3,M4),K(10)
C     SOLUTION OF SIMULTANEOUS EQUATIONS WITH MULTIPLE CONSTANT VECTORS
C     BY GAUSS ELIMINATION, USING PIVOT SEARCH.
C     BY G. N. VANDERPLAATS,   9-25-70
C     A=COEF. MATRIX    B=MATRIX CONTAINING NLC CONSTANT VECTORS
C     N=NO. OF EQUATIONS      M1 AND M2 ARE DIMENSIONS AS GIVEN ABOVE
C     IF NER=1 ON RETURN, A IS SINGULAR.
      NER=1
      EPS=1.0E-20
C     INITIALIZE K TO ZERO
      DO 10 I=1,N
   10 K(I)=0
C     BEGIN ELIMINATION
      DO 90 J=1,N
C     FIND BEST PIVOT ROW
      AA=0.
      II=0
      DO 20 I=1,N
      IF (K(I).NE.0) GO TO 20
      BB=ABS(A(I,J))
      IF (BB.LE.AA) GO TO 20
      AA=BB
      II=I
   20 CONTINUE
      IF (II.EQ.0.OR.AA.LE.EPS) RETURN
      K(II)=J
C     PIVOT ON POSITION A(II,J)
C     REDUCE A(II,J) TO IDENTITY
      AA=1./A(II,J)
      DO 30 L=J,N
   30 A(II,L)=A(II,L)*AA
      DO 40 L=1,NLC
   40 B(II,L)=B(II,L)*AA
C     ELIM. COEF. OF JTH COL. FOR I.NE.II
      L1=J+1
      DO 80 I=1,N
      IF (I.EQ.II) GO TO 80
      BB=A(I,J)
      IF (ABS(BB).LE.EPS) GO TO 80
      IF (L1.GT.N) GO TO 60
      DO 50 L=L1,N
   50 A(I,L)=A(I,L)-A(II,L)*BB
   60 CONTINUE
      DO 70 L=1,NLC
   70 B(I,L)=B(I,L)-B(II,L)*BB
   80 CONTINUE
   90 CONTINUE
C     RE-ORDER VARIABLES TO ORIGINAL POSITION
C     TEMPORARILY STORE SOLN. MATRIX IN A
```

```
      SUBROUTINE GELIM2
      DO 100 I=1,N
      DO 100 J=1,NLC
  100 A(I,J)=B(I,J)
C     STORE VALUES BACK IN B IN PROPER ORDER
      DO 110 I=1,N
      L=K(I)
      DO 110 J=1,NLC
  110 B(L,J)=A(I,J)
      NER=0
      RETURN
      END
```

```
      SUBROUTINE IZERN(IRB,RBI,IETA,ETAI,NETA,R,PD,A)
      DIMENSION RBI(1),ETAI(1),PD(1),A(1),RI(4),TI(4),PDI(4)
C     ROUTINE TO CALCULATE ZERNICKE FUNCTIONS OF SECTION OF BEAM WITH
C     FIRST NODE IRB, IETA.
C     BY G. N. VANDERPLAATS                        MAY, 1977.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C     IF IRB = 1 AND IETA = 1, THIS IF THE FIRST CALL TO IZERN.
C     THERFFORE ZERO OUT VECTOR A.
      IF(IRB.GT.1.OR.IETA.GT.1) GO TO 10
      DO 20 I=1,10
20    A(I)=0.
10    CONTINUE
C     RADIAL COORDINATES.
      RI(1)=0.
      IRB1=IRB-1
      IF(IRB.GT.1) RI(1)=RBI(IRB1)
      RI(4)=RI(1)
      RI(2)=RBI(IRB)
      RI(3)=RI(2)
C     ETA COORDINATES.
      TI(1)=ETAI(IETA)
      TI(2)=TI(1)
      IETA1=IETA+1
      TI(3)=ETAI(1)+6.2831854
      IF(IETA.LT.NETA) TI(3)=ETAI(IETA1)
      TI(4)=TI(3)
C     PHASE DISTORTION.
      N1=(IRB-2)*NETA+IETA
      N2=N1+NETA
      N3=N2+1
      IF(IETA.EQ.NETA) N3=N3-NETA
      N4=N1+1
      IF(IETA.EQ.NETA) N4=N4-NETA
      PDI(1)=0.
      IF(N1.GT.0) PDI(1)=PD(N1)
      PDI(2)=PD(N2)
      PDI(3)=PD(N3)
      PDI(4)=0.
      IF(N1.GT.0) PDI(4)=PD(N4)
C     CALCULATE INTERPOLATION COEFFICIENTS.
      N=4
      IF(IRB.EQ.1) N=3
      CALL FXY34(N,RI,TI,PDI,NER)
C     INTEGRATION.
      R1=RI(1)
      T2=TI(2)
      R2=RI(2)
      T3=TI(3)
      AZ=PDI(1)
      A1=PDI(2)
```

```
A2=PDI(3)
A3=PDI(4)
CALL ZERN(R,R1,R2,T2,T3,AZ,A1,A2,A3,A)
RETURN
END
```

```
      SUBROUTINE PHDIST (X,R,THETA,EPSM,XM,PHI,GAMMA,RHO,Y,Z,PHIPP,U,V,C
     1P,ABAR,BBAR,AL,ACL,THMAX,EPS,RINDEX,RB,ETA,AN,BN,MAXK,MAXP,NMAX,MM
     2AX,KTRAP,A,B,T,DELOPL,IBEAM,REFOPL,WAVEL,RFUS,ETAI,RBI,GAMMAI,PHII
     3,NETAI,NRBI,TDENRT,PDISTI,DENGAM,AMACH,DENRTO,AKPRIM,IPRINT,IPLOT)
      DIMENSION ABAR(1),BBAR(1),AN(1),BN(1),T(1),ETAI(1),RBI(1),GAMMAI(1
     1),PHII(1),PDISTI(1)
      DIMENSION AI(32),XP(100),YP(100),ZP(100)
C     ROUTINE TO CALCULATE PHASE DISTORTION FOR THE IBEAM TURRET
C     ORIENTATION.
C     BY G. N. VANDERPLAATS                         NOV., 1976
C     NAVAL POST GRADUATE SCHOOL, MONTEREY, CALIF.
C     REFOPL = REFERENCE DELTA PATH LENGTH ALONG CENTER OF BEAM.
      NEXTRA=3
C     BEAM ORIENTATION.
      PHI=PHII(IBEAM)
      GAMMA=GAMMAI(IBEAM)
      A1=57.29578*PHI
      A2=57.29578*GAMMA
      IF(IPRINT.GT.0) WRITE(6,90) IBEAM,A1,A2,AMACH
C     CALCULATE REFERENCE PHASE DISTORTION.
      RB=0.
      ETA=0.
C     TURRET SURFACE INTERCEPT.
      CALL SRFINT (XM,EPSM,PHI,GAMMA,A,RB,ETA,X,R,THETA,ABAR,BBAR,EPS,MA
     1XK,MAXP,AI,THMAX)
      AIREF=A
C     REFERENCE CHANGE IN PATH LENGTH DUE TO DISTORTION.
      CALL DOPL (X,R,THETA,EPSM,XM,PHI,GAMMA,RHO,Y,Z,PHIPP,U,V,CP,ABAR,B
     1BAR,AL,ACL,THMAX,EPS,RINDEX,RB,ETA,AN,BN,MAXK,MAXP,NMAX,MMAX,KTRAP
     2,A,B,T,DELOPL,TDENRT,DENGAM,AMACH,DENRTO,AKPRIM,DELPLA)
      REFOPL=DELOPL*RFUS/WAVEL
      A1=57.29578*ETA
      A2=0.
      XP(1)=0.
      YP(1)=0.
      ZP(1)=0.
      IF(IPRINT.GT.0) WRITE(6,100)RB,A1,A2,A2,A,A2
C     CHANGE IN PATH LENGTH DUE TO DISTORTION FOR SPECIFIED VALUES OF
C     RB AND ETA.
C     INCRIMENT RB.
      NN=0
      MM=1
      DO 60 IRB=1,NRBI
      RB=RBI(IRB)
C     INCREMENT ETA.
      DO 50 IETA=1,NETAI
      ETA=ETAI(IETA)
C     SURFACE INTERCEPT.
      CALL SRFINT (XM,EPSM,PHI,GAMMA,A,RB,ETA,X,R,THETA,ABAR,BBAR,EPS,MA
     1XK,MAXP,AI,THMAX)
```

```
C     CHANGE IN PATH LENGTH DUE TO DISTORTION.
      CALL DOPL (X,R,THETA,EPSM,XM,PHI,GAMMA,RHO,Y,Z,PHIPP,U,V,CP,ABAR,B
     1BAR,AL,ACI,THMAX,EPS,RINDEX,RB,ETA,AN,BN,MAXK,MAXP,NMAX,MMAX,KTRAP
     2,A,B,T,DELOPL,TDENRT,DENGAM,AMACH,DENRTO,AKPRIM,DELPLA)
      DPL=DELOPL*RFUS/WAVEL
      NN=NN+1
      MM=MM+1
      AI(NN)=A
      PDISTI(NN)=DPL-REFOPL
      A1=57.29578*ETA
      XX=RB*SIN(ETA)
      YY=RB*COS(ETA)
      XP(MM)=XX
      YP(MM)=YY
      ZP(MM)=PDISTI(NN)
      IF(IPRINT.GT.0) WRITE(6,100)RB,A1,XX,YY,A,PDISTI(NN)
      IF (IRB.LT.NRBI) GO TO 40
      IF (IETA.GT.1) GO TO 10
      X11=XP(MM)
      Y11=YP(MM)
      DP11=PDISTI(NN)
      ETA11=ETA+6.2831854
      GO TO 40
C     INTERPOLATE FOR MORE BOUNDARY POINTS.
   10 NCOUNT=0
      MM1=MM+NEXTRA
      XP(MM1)=XP(MM)
      YP(MM1)=YP(MM)
      ZP(MM1)=ZP(MM)
      DETA=(ETA-ETA1)/(FLOAT(NEXTRA)+1.)
      DPD=PDISTI(NN)-PDIST1
      DX=XP(MM)-XIM1
      DY=YP(MM)-YIM1
   20 CONTINUE
      IF (ABS(DX).LT.1.0E-10) DX=1.0E-10
      IF (ABS(DY).LT.1.0E-10) DY=1.0E-10
      DO 30 INT=1,NEXTRA
      ETA1=ETA1+DETA
      XX=RB*SIN(ETA1)
      YY=RB*COS(ETA1)
      XP(MM)=XX
      YP(MM)=YY
      ZP(MM)=PDIST1+DPD*(YY-YIM1)/DY
   30 MM=MM+1
      NCOUNT=NCOUNT+1
      IF (IETA.LT.NETA1) GO TO 40
      IF (NCOUNT.GT.1) GO TO 40
      DETA=(ETA11-ETA)/(FLOAT(NEXTRA)+1.)
      PDIST1=PDISTI(NN)
      DPD=DP11-PDIST1
```

```
      ETA1=ETA
      XIM1=XP(MM)
      YIM1=YP(MM)
      DX=X11-XIM1
      DY=Y11-YIM1
      MM=MM+1
      GO TO 20
   40 CONTINUE
      ETA1=ETA
      PDIST1=PDISTI(NN)
      XIM1=XP(MM)
      YIM1=YP(MM)
   50 CONTINUE
   60 CONTINUE
      MM=MM-1
      PHI=57.29578*PHII(IBEAM)
      GAMMA=57.29578*GAMMAI(IBEAM)
      IF (IPLOT.GT.0) CALL MAPS (MM,PHI,GAMMA,NETAI,NRBI,XP,YP,ZP)
      IF (IPRINT.EQ.0) RETURN
C     CALCULATE ZERNICKE COEFFICIENTS.
C     VECTOR ZP IS USED TO STORE ZERNICKE COEFFICIENTS, A.
      RBMAX=RBI(NRBI)
      DO 62 IRB=1,NRBI
      DO 62 IETA=1,NETAI
62    CALL IZERN(IRB,RBI,IETA,ETAI,NETAI,RBMAX,PDISTI,ZP)
      WRITE(6,63)(ZP(I),I=1,10)
63    FORMAT(///5X,22HZERNICKE COEFFICIENTS//5X,9HAVERAGE =,E13.5/5X,
     *9HTILT, X =,E13.5,10X,3HY =,E13.5/5X,9HFOCUS   =,F13.5/5X,
     * 9HASTIG   =,2E13.5/5X,9HCOMA    =,4E13.5)
      RETURN
C
   90 FORMAT (///5X,29HPHASE DISTORTION CALCULATIONS//5X,25HBEAM ORIENTA
     1TION NUMBER =,I5/5X,25HAZMUTH ANGLE            =,F10.2,8H DEGREES/
     * 5X,25HELEVATION ANGLE         =,F10.2,8H DEGREES/5X,
     *11HMACH NUMBER,13X,1H=,F10.2/10X,1HR,9X,3HETA,8X,1HX,11X,1HY,11X,
     *1HA,11X,1HN)
  100 FORMAT (5X,E10.4,2X,F7.2,6E12.4)
      END
```

SUBROUTINE PHIUV SEPT. 77

```
      SUBROUTINE PHIUV (X,THETA,R,AMACH,AL,ACL,THMAX,MAXK,MAXP,NMAX,MMAX
     1,ABAR,BBAR,EPS,AN,BN,PHI,U,V)
      DIMENSION ABAR(1),BBAR(1),AN(1),BN(1)
C     ROUTINE TO CALCULATE POTENTIAL FUNCTION, PHI, AND PERTURBATION
C     VELOCITIES U AND V.
C     BY G. N. VANDERPLAATS                        OCT., 1976
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C
C     CONSTANTS.
      DEL1=1.0E-4
      DEL2=1.0E-4
      BETA=1.-AMACH**2
      BETA=ABS(BETA)
      BETA=SQRT(BETA)
      PI=3.1415927
      BPIL=BETA*PI/ACL
      BPIRL=BPIL*R
      NMAX1=NMAX+1
C     INITIALIZE PHI, U AND V.
      PHI=0.
      U=0.
      V=0.
C     CALCULATE POTENTIAL AND VELOCITIES.
C     M - LOOP.
      DO 40 M=1,MMAX
      AM=FLOAT(M)
      AMPIL=AM*BPIL
      AMPIRL=AM*BPIRL
      IF (AMACH.GT.1.) GO TO 10
C     SUBSONIC.
C     K-BESSEL FUNCTIONS FOR N=-1 AND N=0.
      N=1
      CALL BESK (AMPIRL,N,BKRN,IER)
      CALL BESK (AMPIL,N,BKN,IER)
      N=0
      CALL BESK (AMPIRL,N,BKRNP1,IER)
      CALL BESK (AMPIL,N,BKNP1,IER)
      GO TO 20
   10 CONTINUE
C     SUPERSONIC.
C     J-BESSEL FUNCTIONS FOR N=-1 AND N=0.
      PRECIS=.0001
      N=1
      CALL BESJ(AMPIRL,N,BJRN,PRECIS,IER)
      CALL BESJ(AMPIL,N,BJN,PRECIS,IER)
      BJRN=-BJRN
      BJN=-BJN
      N=0
      CALL BESJ(AMPIRL,N,BJRNP1,PRECIS,IER)
      CALL BESJ(AMPIL,N,BJNP1,PRECIS,IER)
```

```
C     Y-BESSEL FUNCTIONS FOR N=-1 AND N=0.
      N=1
      CALL BESY(AMPIRL,N,BYRN,IER)
      CALL BESY(AMPIL,N,BYN,IER)
      BYRN=-BYRN
      BYN=-BYN
      N=0
      CALL BESY(AMPIRL,N,BYRNP1,IER)
      CALL BESY(AMPIL,N,BYNP1,IER)
   20 CONTINUE
C     N - LOOP.
      DO 30 NP1=1,NMAX1
      N=NP1-1
      IF(AMACH.GT.1.) GO TO 25
C     SUBSONIC.
      BKNM1=BKN
      BKRNM1=BKRN
      BKN=BKNP1
      BKRN=BKRNP1
C     N+1 BESSEL FUNCTIONS BY RECURSION.
      BKNP1=2.*FLOAT(N)*BKN/AMPIL+BKNM1
      BKRNP1=2.*FLOAT(N)*BKRN/AMPIRL+BKRNM1
      GO TO 27
25    CONTINUE
C     SUPERSONIC.
      BYNM1=BYN
      BYRNM1=BYRN
      BYN=BYNP1
      BYRN=BYRNP1
      BJNM1=BJN
      BJRNM1=BJRN
      BJN=BJNP1
      BJRN=BJRNP1
C     N+1 BESSEL FUNCTIONS BY RECURSION.
      BYNP1=2.*FLOAT(N)*BYN/AMPIL-BYNM1
      BYRNP1=2.*FLOAT(N)*BYRN/AMPIRL-BYRNM1
      BJNP1=2.*FLOAT(N)*BJN/AMPIL-BJNM1
      BJRNP1=2.*FLOAT(N)*BJRN/AMPIRL-BJRNM1
27    CONTINUE
C     N,M COMPONENT OF PHI, U AND V.
      CALL PHUVNM (N,M,X,THETA,AMACH,AL,ACL,THMAX,BKNM1,BKNP1,BKRNM1,BKR
     1N,BKRNP1,MAXK,MAXP,ABAR,BBAR,EPS,AN,BN,PHINM,UNM,VNM,
     * BJNM1,BJN,BJNP1,BJRNM1,BJRN,BJRNP1,BYNM1,BYN,BYNP1,BYRNM1,
     * BYRN,BYRNP1)
C     UPDATE PHI, U AND V.
      PHI=PHI+PHINM
      U=U+UNM
      V=V+VNM
C     CHECK CONVERGENCE.
      IF(N.EQ.0) GO TO 30
```

```
      IF(ABS(PHINM).LT.DEL1.AND.(ABS(UNM).LT.DEL1.AND.ABS(VNM).LT.DEL1))
     * GO TO 35
   30 CONTINUE
35    CONTINUE
      IF(M.EQ.1) GO TO 36
      DPHI=ABS(PHI-PHIA)
      DU=ABS(U-UA)
      DV=ABS(V-VA)
      IF(DPHI.LT.DEL2.AND.(DU.LT.DEL2.AND.DV.LT.DEL2)) GO TO 45
36    PHIA=PHI
      UA=U
      VA=V
   40 CONTINUE
45    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PHUVNM (N,M,X,THETA,AMACH,AL,ACL,THMAX,BKNM1,BKNP1,BKRN
     1M1,BKRN,BKRNP1,MAXK,MAXP,ABAR,BBAR,EPS,AN,BN,PHINM,UNM,VNM,
     * BJNM1,BJN,BJNP1,BJRNM1,BJRN,BJRNP1,BYNM1,BYN,BYNP1,BYRNM1,
     * BYRN,BYRNP1)
      COMMON /CMLOC2/AMX(10,15),BMX(10,15),ANT(10,15)
      DIMENSION ABAR(1),BBAR(1),AN(1),BN(1)
C     ROUTINE TO CALCULATE N,M COMPONENTS OF POTENTIAL, PHINM, AND
C     PERTURBATION VELOCITIES UNM AND VNM FOR A TURRET DEFINED BY A
C     DOUBLE POLYNOMIAL.
C     BY G. N. VANDERPLAATS                           OCT., 1976
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C  INPUT
C     N,M      - SUBSCRIPTS ON PHI, U AND V.
C     X        - LONGITUDINAL COORDINATE ALONG TURRET.
C     THETA    - CIRCUMFERENTIAL COORDINATE AROUND TURRET.
C     BETA     - ABS(1.-AMACH**2)
C     AL,ACL   - 1/2 LENGTH OF TURRET AND 1/2 PERIOD BETWEEN TURRETS.
C     THMAX    - 1/2 CIRCUMFERENCE OF FUSELAGE OCCUPIED BY TURRET.
C     BKNM1, BKNP1 - K BESSEL FUNCTIONS AT N-1 AND N+1.
C     BKRNM1, BKRN, BKRNP1 - K BESSEL FUNCTIONS OF R AT N-1, N AND N+1.
C     MAXK, MAXP  - MAX EXPONENT OF X AND THETA POLYNOMIALS.
C     ABAR, BBAR - X AND THETA POLYNOMIAL COEFICIENTS.
C     AN, BN   - DUMMY STORAGE DIMENSIONED MAX(MAXK+1,MAXP+1)
C  OUTPUT
C     PHINM    - PERTURBATION POTENTIAL.
C     UNM      - U PERTURBATION VELOCITY.
C     VNM      - V PERTURBATION VELOCITY.
C
C     CONSTANTS.
      PI=3.1415927
      AMPL=FLOAT(M)*PI/ACL
      BETA=ABS(1.-AMACH**2)
      BETA=SQRT(BETA)
      BMPL=BETA*AMPL
      SM=AMPL*X
      CM=COS(SM)
      SM=SIN(SM)
      SN=FLOAT(N)*THETA
      CN=COS(SN)
      MAXKP1=MAXK+1
      MAXPP1=MAXP+1
C     CALCULATE A-BAR TIMES A-SUB-M AND A-BAR TIMES B-SUB-M.
      AAM=0.
      ABM=0.
      DO 10 I=1,MAXKP1
      AAM=AAM+ABAR(I)*AMX(I,M)
10    ABM=ABM+ABAR(I)*BMX(I,M)
C     CALCULATE B-BAR TIMES A-SUB-N.
      BAN=0.
      BBN=0.
```

```
      NP1=N+1
      DO 20 I=1,MAXPP1
20    BAN=BAN+BBAR(I)*ANT(I,NP1)
C     CALCULATE F-SUB-N OF THETA.
      FN=BAN*CN
      IF(AMACH.GT.1.) GO TO 30
C     SUBSONIC.
C     CALCULATE PHINM.
      C1=AAM*SM-ABM*CM
      C2=BETA*(BKNP1+BKNM1)
      C3=2.*EPS*FN*BKRN
      PHINM=C3*C1/C2
C     CALCULATE UNM.
      UNM=C3*AMPL*(AAM*CM+ABM*SM)/C2
C     CALCULATE VNM.
      VNM=-AMPL*EPS*FN*(BKRNP1+BKRNM1)*C1/(BKNP1+BKNM1)
      RETURN
30    CONTINUE
C     SUPERSONIC.
      ANM=BYNP1-BYNM1+BJNP1-BJNM1
      BNM=BYNP1-BYNM1-BJNP1+BJNM1
      APB=ANM+BNM
      AMB=ANM-BNM
      AB2=ANM**2+BNM**2
      A1=APB*SM-AMB*CM
      A2=AMB*SM+APB*CM
      A3=AAM*BYRN+ABM*BJRN
      A4=AAM*BJRN-ABM*BYRN
      A5=2.*EPS*FN/(AB2*BETA)
C     PHINM.
      PHINM=A5*(A1*A3+A2*A4)
C     UNM.
      UNM=A5*AMPL*(A2*A3-A1*A4)
C     VNM
      VNM=-EPS*FN*AMPL*((A1*AAM-A2*ABM)*(BYRNP1-BYRNM1)+
     $(A1*ABM+A2*AAM)*(BJRNP1-BJRNM1))/AB2
      RETURN
      END
```

```
      SUBROUTINE REFIND                                        SEPT. 77

      SUBROUTINE REFIND (X,R,THETA,EPSM,XM,PHI,GAMMA,RHO,Y,Z,PHIPP,U,V,C
     1P,ABAR,BBAR,AL,ACL,THMAX,EPS,RINDEX,RB,ETA,AN,BN,MAXK,MAXP,NMAX,MM
     2AX,DENGAM,AMACH,DENRTO,AKPRIM)
      DIMENSION ABAR(1),BBAR(1),AN(1),BN(1)
C     ROUTINE TO CALCULATE INDEX OF REFRACTION -1 FOR A SPECIFIED POINT
C     ON A BEAM.
C     BY G. N. VANDERPLAATS                       NOV., 1976.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C     GIVIN AZMUTH, ELEVATION AND DISTANCE ALONG BEAM, CALCULATE
C     X, THETA AND R-COORDINATES.
      CALL XRTPOB (XM,EPSM,PHI,GAMMA,RHO,RB,ETA,X,R,THETA,Y,Z)
C     CALCULATE POTENTIAL AND PERTURBATION VELOCITIES.
      CALL PHIUV (X,THETA,R,AMACH,AL,ACL,THMAX,MAXK,MAXP,NMAX,MMAX,ABAR,
     1BBAR,EPS,AN,BN,PHIPP,U,V)
C     INDEX OF REFRACTION.
      CP=-2.*U-V*V
      C1=1.+.5*DENGAM*AMACH*AMACH*CP
      RINDEX=AKPRIM*DENRTO/(C1**DENGAM)
      RETURN
      END
```

```
      SUBROUTINE RMBINT (T,K,K1)
      DIMENSION T(1)
C     ROUTINE TO PERFORM ROMBERG INTEGRATION.
C     BY G. N. VANDERPLAATS                        NOV., 1976
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C
C  INPUT
C     T  = VECTOR CONTAINING RESULTS OF TRAPEZOIDAL RULE INTEGRATION.
C          IF T(1) CONTAINS TRAP. RULE RESULTS FOR N INTERVALS, T(2)
C          CONTAINS RESULTS FOR 2N INTERVALS, T(3) CONTAINS RESULTS FOR
C          4N INTERVALS AND T(I) CONTAINS RESULTS FOR (2**(I-1))N
C          INTERVALS.
C     K  = NUMBER OF TRAPEZOIDAL RULE RESULTS CONTAINED IN T.
C     K1 = K ON LAST CALL TO RMBINT.  FIRST TIME RMBINT IS CALLED K1=1.
C  OUTPUT.
C     T  = VECTOR CONTAINING LAST ROW OF ROMBERG TABLE IN REVERSE ORDER.
C        THE HIGHEST ORDER APPROXIMATION TO THE INTEGRAL IS IN T(1).
C        T(2) GIVES THE 2ND HIGHEST ORDER APPROXIMATION SO THE
C        DIFFERENCE BETWEEN T(1) AND T(2) IS AN ACCURACY ESTIMATION.
C        T(K) IS THE HIGHEST ORDER TRAP. RULE APPROXIMATION AND IS NOT
C        DESTROYED.
C  NOTES
C     1) IF ACCURACY IS NOT SATISFACTORY, THE NUMBER OF TRAP RULE
C        STATIONS CAN BE DOUBLED AND A NEW SOLUTION STORED IN K+1 OF T.
C        THEN SET K1=K AND K=K+1 AND CALL RMBINT AGAIN FOR NEW SOLUTION.
C     2) ALL INITIAL ENTRIES OF T UP TO K-1 ARE DESTROYED.
C  REFERENCE, CONTE, ELEMENTARY NUMERICAL ANALYSIS, MCGRAW-HILL,
C     1965, PP 126-133.
      IF (K.LE.1) RETURN
      K1P1=K1+1
C     BUILD ROW KK OF ROMBERG TABLE.
      DO 10 KK=K1P1,K
      KM1=KK-1
      A=1.
      I=KK
C     PUT ROW KK IN T(I),I=1,KK IN REVERSE ORDER. T(KK) DOES NOT CHANGE.
      DO 10 II=1,KM1
      I=I-1
      A=4.*A
   10 T(I)=(A*T(I+1)-T(I))/(A-1.)
      RETURN
      END
```

SUBROUTINE RSURF SEPT. 77

```
      SUBROUTINE RSURF (ABAR,BBAR,EPS,MAXK,MAXP,X,THETA,AL,THMAX,R)
      DIMENSION ABAR(1),BBAR(1)
C     ROUTINE TO CALCULATE THE NON-DIMENSIONAL TURRET RADIUS AT
C     X AND THETA.
C     BY G. N. VANDERPLAATS                         NOV., 1976.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C     SPECIAL CASE - THETA OR X NOT ON TURRET, POINT IS ON CYLINDRICAL
C     FUSELAGE.
      R=1.
      IF (ABS(THETA).GE.THMAX.OR.ABS(X).GE.AL) RETURN
C     CONSTANTS.
      MAXKP1=MAXK+1
      MAXPP1=MAXP+1
C     POINT ON TURRET.
C     EVALUATE F(X)
      FX=ABAR(1)
      IF (MAXK.EQ.0) GO TO 20
      XI=1.
      DO 10 IX=2,MAXKP1
      XI=XI*X
      IF(ABS(XI).LT.1.0E-20) GO TO 20
   10 FX=FX+ABAR(IX)*XI
   20 CONTINUE
C     EVALUATE F(THETA)
      FTH=BBAR(1)
      IF (MAXP.EQ.0) GO TO 40
      THI=1.
      DO 30 ITH=2,MAXPP1
      THI=THI*THETA
      IF(ABS(THI).LT.1.0E-20) GO TO 40
   30 FTH=FTH+BBAR(ITH)*THI
   40 CONTINUE
C     R=1.0 + F(X)*F(THETA)*EPS
      R=1.+FX*FTH*EPS
      RETURN
      END
```

```
      SUBROUTINE SLOPE (MAXK,ABAR,AL,SLOPEX,NVAL,AMULTS)
      DIMENSION ABAR(1),SLOPEX(1)
C     ROUTINE TO CALCULATE SLOPE OF A POLYNOMIAL AT NVAL POINTS
C     BETWEEN X = -AL AND X = AL.
      IF (NVAL.LT.2) RETURN
      DX=2.*AL/(FLOAT(NVAL)-1.)
      X=-AL-DX
      MAXK1=MAXK+1
      DO 30 I=1,NVAL
      X=X+DX
      SLOPEX(I)=0.
      IF (MAXK.LT.1) GO TO 30
      SLOPEX(I)=ABAR(2)
      IF (MAXK.EQ.1) GO TO 20
      AMULT=1.
      XI=1.
      DO 10 J=3,MAXK1
      XI=XI*X
      AMULT=AMULT+1.
   10 SLOPEX(I)=SLOPEX(I)+AMULT*ABAR(J)*XI
   20 SLOPEX(I)=AMULTS*SLOPEX(I)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SRFINT (XM,EPSM,PHI,GAMMA,A,RB,ETA,X,R,THETA,ABAR,BBAR,
     1EPS,MAXK,MAXP,AL,THMAX)
      DIMENSION ABAR(1),BBAR(1)
C     ROUTINE TO CALCULATE DISTANCE ALONG BEAM FROM MIRROR TO TURRET
C     SURFACE.
C     BY G. N. VANDERPLAATS                                NOV., 1976
C     NAVAL POST GRADUATE SCHOOL, MONTEREY, CALIF.
C  OUTPUT.
C     A = DISTANCE FROM MIRROR TO TURRET SURFACE.
C       IF A = -1.0E-6 ON RETURN, MORROR SURFACE IS OUTSIDE TURRET
C       SURFACE.
C       IF A = 1.0E-6 ON RETURN, NO INTERCEPT COULD BE FOUND AT A LESS
C       THAN 10.  THIS PROBABLY  RESULTS FROM UNREALISTIC TURRET SHAPE.
C  METHOD.
C     FOR VARIOUS VALUES OF RHO, CALCULATE X, RR AND THETA FOR A POINT
C       ON THE BEAM.  FOR EACH X AND THETA, CALCULATE RS FOR RADIUS TO
C       THE SURFACE.  INTERPOLATE TO GET RR=RS.  THE CORRESPONDING VALUE
C       OF RHO IS A.
      DRHO=.2
C     RADIUS OF BEAM RAY AT POINT ON MIRROR SURFACE.
      RHO=0.
      A1=0.
      RR1=EPSM
      X=XM
      THETA=0.
      IF (RB.GT.1.0E-4) CALL XRTPOB (XM,EPSM,PHI,GAMMA,RHO,RB,ETA,X,RR1,
     1THETA,Y,Z)
C     SURFACE RADIUS OF POINT AT X AND THETA, FOR RHO=0.
      CALL RSURF (ABAR,BBAR,EPS,MAXK,MAXP,X,THETA,AL,THMAX,RS1)
      DR1=RS1-RR1
      A=-1.0E-6
C     IF DR1.LT.0, BASE OF MIRROR IS OUTSIDE TURRET.
      IF (DR1.LT.0.) RETURN
C     PICK ARBITRARY NEW RHO AND INTERPOLATE.
   10 RHO=RHO+DRHO
C     RADIUS OF POINT ON BEAM.
      CALL XRTPOB (XM,EPSM,PHI,GAMMA,RHO,RB,ETA,X,RR2,THETA,Y,Z)
C     TURRET SURFACE.
      CALL RSURF (ABAR,BBAR,EPS,MAXK,MAXP,X,THETA,AL,THMAX,RS2)
      DR2=RS2-RR2
      DDR=DR2-DR1
      IF (ABS(DDR).LT.1.0E-10) DDR=1.0E-10
      A=A1-DRHO*DR1/DDR
      IF (A.LE.RHO.OR.RHO.GT.10.) GO TO 20
C     A IS EXTRAPOLATED POINT, UPDATE AND INTERPOLATE AGAIN.
      RR1=RR2
      RS1=RS2
      DR1=DR2
      A1=RHO
      GO TO 10
```

```
20 CONTINUE
   IF (A.LT.0.) A=1.0E-6
   RETURN
   END
```

```
      SUBROUTINE SURPRT                                            SEPT. 77

      SUBROUTINE SURPRT (ABAR,BBAR,MAXK,MAXP,EPS,AL,THMAX)
      DIMENSION ABAR(1),BBAR(1)
C     ROUTINE TO PRINT SURFACE FUNCTION ORDINATES FOR POLYNOMIAL TURRET.
C     BY G. N. VANDERPLAATS                            NOV., 1976.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C  INPUT.
C     ABAR   = VECTOR OF POLYNOMIAL COEFICIENTS IN X-DIRECTION.  ABAR
C              MUST BE DIMENSIONED AT LEAST MAXK+1 IN CALLING ROUTINE.
C     BBAR   = VECTOR OF POLYNOMIAL COEFICIENTS IN THETA-DIRECTION.
C              BBAR MUST BE DIMENSIONED AT LEAST MAXP+1 IN CALLING
C              ROUTINE.
C     MAXK   = ORDER OF X-POLYNOMIAL.
C     MAXP   = ORDER OF THETA-POLYNOMIAL.
C     EPS    = SCALAR SURFACE MULTIPLIER.  SURFACE = EPS*F(X)*F(THETA).
C     AL     = 1/2 TURRET LENGTH.
C     THMAX  = 1/2 TURRET ANGLE.
C     OUTPUT.
C     POLYNOMIAL FUNCTION COORDINATES IN TERMS OF X AT THETA = 0 AND
C     THETA AT X = 0.
      MAXK1=MAXK+1
      MAXP1=MAXP+1
      WRITE (6,70) EPS
      WRITE (6,80) (ABAR(I),I=1,MAXK1)
      WRITE (6,90)
      WRITE (6,80) (BBAR(I),I=1,MAXP1)
C     X-DIRECTION.
      WRITE (6,100)
      DX=.1*AL
      X=-1.2*AL
      DO 30 I=1,23
      X=X+DX
      Z=ABAR(1)
      AMULT=0.
      ZPRIM=0.
      IF (MAXK.EQ.0) GO TO 20
      XI=1.
      DO 10 J=2,MAXK1
      AMULT=AMULT+1.
      ZPRIM=ZPRIM+AMULT*ABAR(J)*XI
      XI=XI*X
   10 Z=Z+ABAR(J)*XI
   20 CONTINUE
      Z=EPS*BBAR(1)*Z
      ZPRIM=EPS*BBAR(1)*ZPRIM
      IF (I.EQ.1.OR.I.EQ.23) Z=0.
      IF (I.EQ.1.OR.I.EQ.23) ZPRIM=0.
      WRITE (6,110) X,Z,ZPRIM
   30 CONTINUE
C     THETA-DIRECTION.
      WRITE (6,120)
```

```
      DTH=.1*THMAX
      TH=-1.2*THMAX
      DO 60 I=1,23
      TH=TH+DTH
      Z=BBAR(1)
      IF (MAXP.EQ.0) GO TO 50
      THI=1.
      AMULT=0.
      ZPRIM=0.
      DO 40 J=2,MAXP1
      AMULT=AMULT+1.
      ZPRIM=ZPRIM+AMULT*BBAR(J)*THI
      THI=THI*TH
   40 Z=Z+BBAR(J)*THI
   50 CONTINUE
      Z=EPS*ABAR(1)*Z
      ZPRIM=EPS*ABAR(1)*ZPRIM
      IF (I.EQ.1.OR.I.EQ.23) Z=0.
      IF (I.EQ.1.OR.I.EQ.23) ZPRIM=0.
      THR=TH*57.29578
      WRITE (6,110) TH,THR,Z,ZPRIM
   60 CONTINUE
      RETURN
C
   70 FORMAT (///5X,18HSURFACE DEFINITION,5X,6H(EPS =,F7.3,1H)/5X,54HPOL
     1YNOMIAL COEFICIENTS (A(I), I=0,MAXK) IN X-DIRECTION)
   80 FORMAT (5X,5E12.5)
   90 FORMAT (/5X,58HPOLYNOMIAL COEFICIENTS (B(I), I=0,MAXP) IN THETA-DI
     1RECTION)
  100 FORMAT (/5X,11HCOORDINATES/8X,1HX,11X,1HZ,9X,7HZ-PRIME)
  110 FORMAT (5X,F7.3,5X,F8.4,5X,F8.4,5X,F8.4)
  120 FORMAT (/12X,5HTHETA/5X,7HRADIANS,6X,7HDEGREES,8X,1HZ,9X,7HZ-PRIME
     1)
      END
```

```
C     SUBROUTINE TINPUT
C
C     INPUT CARD FORMAT
C
C * TITLE(I),I=1,20                                          FORMAT(20A4)
C         ANYTHING MAY BE TYPED IN COL. 2-80
C
C  AERODYNAMICS - OPTICS
C * AMACH,DENRTO,TDENRT,DENGAM,AKPRIM,WAVEL                  FORMAT(8F10)
C         AMACH  = FREESTREAM MACH NUMBER
C         DENRTO = FLIGHT DENSITY/SEA LEVEL DENSITY
C         TDENRT = DENSITY INSIDE TURRET/SEA LEVEL DENSITY
C         DENGAM = EXPONENT ON PRESSURE-DENSITY RELATIONSHIP
C         AKPRIM = INDEX OF REFRACTION CONSTANT
C         WAVEL  = BEAM WAVELENGTH
C
C  GEOMETRY
C  TURRET
C * RFUS,AL,THMAX,ACL,EPS                                    FORMAT(8F10)
C         RFUS   = FUSELAGE RADIUS
C         AL     = TURRET NON-DIMENSIONAL HALF LENGTH
C         THMAX  = TURRET HALF ANGLE (RAD)
C         ACL    = HALF TURRET SPACING
C         EPS    = TURRET HEIGHT MULTIPLIER
C * MAXK,MAXP,NXBC,NTHBC                                     FORMAT(8I10)
C         MAXK   = ORDER OF X-POLYNOMIAL SHAPE FUNCTION
C         MAXP   = ORDER OF THETA-POLYNOMIAL
C         NXBC   = NUMBER OF SETS OF Y AND Y-PRIME BOUNDARY
C                  CONDITIONS IN X-DIRECTION, EXTERNALLY IMPOSED.
C         NTHBC  = NUMBER OF SETS OF Y AND Y-PRIME BOUNDARY
C                  CONDITIONS IN THETA-DIRECTION, EXTERNALLY IMPOSED.
C                  NOTE. AT X=THETA=0, Y=EPS IS AUTOMATICALLY IMPOSED.
C * ABAR(I),I=1,MAXK+1                                       FORMAT(8F10)
C         ABAR(I) = I-1 COEFICIENT OF X-POLYNOMIAL
C * YYPXBC(I,J), J=1,3                NXBC CARDS             FORMAT(8F10)
C         YYPXBC(I,J) = X, Y AND Y-PRIME BOUNDARY CONDITIONS IN THE
C                       X-DIRECTION.
C * BBAR(I),I=1,MAXP+1                                       FORMAT(8F10)
C         BBAR(I) = I-1 COEFICIENT OF THETA-POLYNOMIAL
C * YYPTBC(I,J), J=1,3                NTHBC CARDS            FORMAT(8F10)
C         YYPTBC(I,J) = X, Y AND Y-PRIME BOUNDARY CONDITIONS IN THE
C                       THETA-DIRECTION.
C
C  MIRROR CENTER
C * EPSM,XM                                                  FORMAT(8F10)
C         EPSM = Z-LOCATION OF CENTER OF MORROR
C         XM   = X-LOCATION OF CENTER OF MIRROR
C
C  PHASE DISTORTION CALCULATION POINTS
C * NETAI,NRBI                                               FORMAT(8I10)
```

```
C          NETAI = NUMBER OF ETA ANGLES
C          NRBI  = NUMBER OF RADIUS POINTS
C * ETAI(I),I=1,NETAI                                           FORMAT(8F10)
C          ETAI(I) = ANGLE (DEGREES)
C * RBI(I),I=1,NRBI                                             FORMAT(8F10)
C          RBI(I)  = RADIUS
C
C  BEAM ORIENTATION
C * NBEAM                                                       FORMAT(8I10)
C          NBEAM = NUMBER OF DIFFERENT BEAM ORIENTATIONS ANALIZED
C * PHII(I),GAMMAI(I),AMACHI(I),WGHTI(I)    NBEAM CARDS   FORMAT(8I10)
C          PHII(I)    = AZMUTH ANGLE (DEGREES)
C          GAMMAI(I) = ELEVATION ANGLE (DEGREES)
C          AMACHI(I) = MACH NUMBER. DEFAULT = AMACH.
C          WGHTI(I) = WEIGHTING COEFFICIENT.  DEFAULT = 1.
      SUBROUTINE TINPUT
      COMMON /GLOBCM/ ABAR(20),ACL,AKPRIM,AL,AMACHI(30),BBAR(20),DENRTO,
     * DENGAM,EPS,EPSM,GAMMAI(30),PHII(30),RFUS,SLOPEX(30),SUMPD2,
     * TDENRT,THMAX,WAVEL,WGHTI(30),XM
      COMMON /CMLOC/ ETAI(16),MAXK,MAXP,NBEAM,NETAI,NRBI,NTHBC,NXBC,
     * RBI(10),TITLE(20),YYPXBC(10,3),YYPTBC(10,3)
C     ROUTINE TO READ INPUT FOR LASER TURRET PHASE DISTORTION ANALYSIS.
C   . BY G. N. VANDERPLAATS                              NOV., 1976
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C
C  TITLE.
      READ (5,70) (TITLE(I),I=1,20)
      IPNPUT=0
      IF (IPNPUT.EQ.0) WRITE (6,140) (TITLE(I),I=1,20)
C  AERO-OPTICS.
      READ (5,80) AMACH,DENRTO,TDENRT,DENGAM,AKPRIM,WAVEL
      IF (IPNPUT.EQ.0) WRITE (6,150) AMACH,DENRTO,TDENRT,DENGAM,AKPRIM,W
     1AVEL
C  GEOMETRY.
C     TURRET.
      READ (5,80) RFUS,AL,THMAX,ACL,EPS
      IF (IPNPUT.EQ.0) WRITE (6,160) RFUS,AL,THMAX,EPS,ACL
      THMAX=THMAX/57.29578
      READ (5,90) MAXK,MAXP,NXBC,NTHBC
      NXBC=NXBC+1
      NTHBC=NTHBC+1
      MAXK1=MAXK+1
      MAXP1=MAXP+1
      IF (IPNPUT.EQ.0) WRITE (6,170) MAXK
      READ (5,80) (ABAR(I),I=1,MAXK1)
      ABAR(1)=1.
      IF (IPNPUT.EQ.0) WRITE (6,180) (ABAR(I),I=1,MAXK1)
      YYPXBC(1,1)=0.
      YYPXBC(1,2)=EPS
      YYPXBC(1,3)=200.
```

```
      IF (NXBC.EQ.1) GO TO 20
      IF (IPNPUT.EQ.0) WRITE (6,100)
      DO 10 I=2,NXBC
      READ (5,80) (YYPXBC(I,J),J=1,3)
   10 CONTINUE
   20 CONTINUE
      IF (IPNPUT.EQ.0) WRITE (6,110) ((YYPXBC(I,J),J=1,3),I=1,NXBC)
      IF (IPNPUT.EQ.0) WRITE (6,190) MAXP
      READ (5,80) (BBAR(I),I=1,MAXP1)
C     IMPOSE BOUNDARY CONDITION R(0)=1.
      BBAR(1)=1.
      IF (IPNPUT.EQ.0) WRITE (6,180) (BBAR(I),I=1,MAXP1)
      YYPTBC(1,1)=0.
      YYPTBC(1,2)=EPS
      YYPTBC(1,3)=200.
      IF (NTHBC.EQ.1) GO TO 40
      IF (IPNPUT.EQ.0) WRITE (6,120)
      DO 30 I=2,NTHBC
      READ (5,80) (YYPTBC(I,J),J=1,3)
   30 CONTINUE
   40 CONTINUE
      IF (IPNPUT.EQ.0) WRITE (6,130) ((YYPTBC(I,J),J=1,3),I=1,NTHBC)
C     MIRROR CENTER.
      READ (5,80) EPSM,XM
      IF (IPNPUT.EQ.0) WRITE (6,200) XM,EPSM
C  PHASE DISTORTION CALCULATION POINTS.
      READ (5,90) NETAI,NRBI
      READ (5,80) (ETAI(I),I=1,NETAI)
      IF (IPNPUT.EQ.0) WRITE (6,230)
      IF (IPNPUT.EQ.0) WRITE (6,240) (ETAI(I),I=1,NETAI)
      DO 50 I=1,NETAI
   50 ETAI(I)=ETAI(I)/57.29578
      READ (5,80) (RBI(I),I=1,NRBI)
      IF (IPNPUT.EQ.0) WRITE (6,250)
      IF (IPNPUT.EQ.0) WRITE (6,240) (RBI(I),I=1,NRBI)
C  BEAM ORIENTATIONS.
      READ (5,90) NBEAM
      IF (IPNPUT.EQ.0) WRITE (6,210)
      DO 60 I=1,NBEAM
      READ(5,80) PHII(I),GAMMAI(I),AMACHI(I),WGHTI(I)
      IF(AMACHI(I).LT.0.001) AMACHI(I)=AMACH
      IF(ABS(WGHTI(I)).LT.0.001) WGHTI(I)=1.
      IF(IPNPUT.EQ.0) WRITE(6,220)I,PHII(I),GAMMAI(I),AMACHI(I),WGHTI(I)
      PHII(I)=PHII(I)/57.29578
      GAMMAI(I)=GAMMAI(I)/57.29578
   60 CONTINUE
      RETURN
C
   70 FORMAT (20A4)
   80 FORMAT (8F10.2)
```

```
 90 FORMAT(8I10)
100 FORMAT (/5X,19HBOUNDARY CONDITIONS/5X,3HX/L,6X,1HY,4X,7HY-PRIME)
110 FORMAT (3F9.3)
120 FORMAT (/5X,19HBOUNDARY CONDITIONS/5X,11HTHETA/THMAX,4X,1HY,4X,7HY
   1-PRIME)
130 FORMAT (5X,3F9.3)
140 FORMAT (1H1,4X,21HTURRET ANALYSIS INPUT//5X,5HTITLE/5X,20A4)
150 FORMAT (//5X,11HAERO-OPTICS/5X,36HMACH NUMBER, AMACH
   1    =,F6.3/5X,36HEXTERNAL DENSITY RATION, DENRTO     =,F6.3/5X,36HIN
   2TERNAL DENSITY RATIO, TDENRT       =,F6.3/5X,36HPRESSURE-DENSITY EXP
   3ONENT, DENGAM  =,F6.3/5X,36HPHASE DISTORTION CONSTANT, AKPRIM  =,E
   411.4/5X,36HWAVELENGTH, WAVEL                        =,E11.4)
160 FORMAT (//5X,8HGEOMETRY/5X,27HFUSELAGE RADIUS, RFUS       =,F7.3/5X,
   127HTURRET HALF-LENGTH,          =,F7.3/5X,27HTURRET HALF-ANGLE, THMAX
   2  =,F7.3,9H  DEGREES/5X,27HTURRET HEIGHT FACTOR, EPS =,F7.3/5X,27H
   3TURRET HALF-SPACING, ACL  =,F7.3)
170 FORMAT (//5X,35HTURRET POLYNOMIAL SHAPE COEFICIENTS/5X,24HX-DIRECT
   1ION,      ORDER =,I5/5X,11HCOEFICIENTS)
180 FORMAT (4X,5E13.5)
190 FORMAT (/5X,24HTHETA-DIRECTION, ORDER =,I5/5X,11HCOEFICIENTS)
200 FORMAT (//5X,28HLOCATION OF CENTER OF MIRROR/5X,6HXM    =,F7.3,5X,6
   1HEPSM =,F7.3)
210   FORMAT(//5X,17HBEAM ORIENTATIONS/5X,18HBEAM    PHI    GAMMA,4X,
   * 12HMACH  WEIGHT)
220   FORMAT(I8,2F8.2,2F8.3)
230 FORMAT (//5X,35HPHASE DISTORTION CALCULATION POINTS/5X,6HANGLES)
240 FORMAT (5X,5F10.3)
250 FORMAT (/5X,5HRADII)
    END
```

```
      SUBROUTINE TRAP2N (IGOTO,A,B,N2,X,FX)
C     ROUTINE TO PERFORM TRAPEZOIDAL RULE INTEGRATION FOR F(X)2N,
C     BEGINING WITH F(X)N.
C     BY G. N. VANDERPLAATS                           NOV., 1976
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C  INPUT
C     IGOTO = CALCULATION PARAMETER.  INITIALLY CALL TRAP2N WITH
C             IGOTO = 0.
C     A     = LOWER BOUND ON INTEGRATION.
C     B     = UPPER BOUND ON INTEGRATION.
C     N2    = NUMBER OF INTERVALS USED IN THIS SOLUTION.  N2 = 1
C             IF INTEGRATION IS JUST BEGINING.  OTHERWISE N2 = 2*N
C             OF PREVIOUS SOLUTION.
C     FX    = F(X)N ON FORST CALL (IGOTO=0) AND F(X) ON SUBSEQUENT CALLS
C             (IGOTO=1).
C  OUTPUT
C     IGOTO = CALCULATION CONTROL.  IF IGOTO.NE.0, CALCULATE F(X) AND
C             CALL AGAIN.  IF IGOTO=0 ON RETURN, INTEGRATION IS COMPLETE
C     X     = X-VALUE FOR NEW FUNCTION EVALUATION (IF IGOTO.NE.0)
C     FX    = F(X)2N IF IGOTO=0.  THIS IS FINAL SOLUTION.
C USAGE       K IS TOTAL NUMBER OF TRAPEZOIDAL SOLUTIONS DESIRED.
C     DO 20 I = 1,K
C     N2=2**(I-1)
C     IGOTO = 0
C 10  CALL TRAP2(IGOTO,A,B,N2,X,FX)
C     IF(IGOTO.EQ.0) GO TO 20
C     FX = F(X)
C     GO TO 10
C 20  CONTINUE
C     SOLUTION IS COMPLETE.
C
      IF (IGOTO-1) 10,20,40
C     CONSTANT.
   10 H=(B-A)/FLOAT(N2)
      FN=0.
      A1=1.
      A2=1.
      IF (N2.GT.1) GO TO 20
C     SPECIAL CASE, 1 INTERVAL.
      A1=H
      A2=.5
      X=A
      IGOTO=1
      RETURN
C     GENERAL CASE, N2.GE.1
   20 FN1=.5*FX*A1
      I=-1
   30 I=I+2
      IF (I.GT.N2) GO TO 50
      X=A+FLOAT(I)*H
```

```
      IGOTO=2
      RETURN
   40 FN=FN+FX
      GO TO 30
   50 FN=A2*FN
      FX=FN1+FN*H
      IGOTO=0
      RETURN
      END
```

```
      SUBROUTINE XRTPOB (XM,EPSM,PHI,GAMMA,RHO,RB,ETA,X,R,THETA,Y,Z)
C     ROUTINE TO CALCULATE COORDINATES, X,R,THETA OF A POINT ON A BEAM.
C     BY G. N. VANDERPLAATS                              NOV., 1976
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C  INPUT.
C     XM      = X-LOCATION OF CENTER OF MIRROR.
C     EPSM    = Z-LOCATION OF CENTER OF MIRROR.
C     PHI     = AZMUTH ANGLE MEASURED FROM POSITIVE X-AXIS.
C     GAMMA   = ELEVATION ANGLE MEASURED FROM X-Y PLANE.
C     RHO     = DISTANCE ALONG BEAM.
C     RB      = RADIAL DISTANCE FROM CENTER OF BEAM.
C     ETA     = ANGULAR LOCATION MEASURED FROM LINE IN THE X-Y PLANE.
C  OUTPUT.
C     X       = X-CYLINDRICAL AND CARTISIAN COORDINATE.
C     Y       = Y-CARTISIAN COORDINATE.
C     Z       = Z-CARTISIAN COORDINATE.
C     R       = RADIAL LOCATION TO POINT FROM X-AXIS.
C     THETA   = CIRCUMFERENTIAL LOCATION OF POINT FROM Z-AXIS.
C  NOTE - ALL ANGLES ARE IN REDIANS.
C
C     CONSTANTS.
      SNP=SIN(PHI)
      CNP=COS(PHI)
      SNG=SIN(GAMMA)
      CNG=COS(GAMMA)
      SNE=SIN(ETA)
      CNE=COS(ETA)
C     CARTISIAN COORDINATES.
C     X = XM - RHO*COS(GAMMA)*COS(PHI) - RB*SIN(ETA)*SIN(PHI) +
C     RB*COS(ETA)*SIN(GAMMA)*COS(PHI)
      X=XM-RHO*CNG*CNP-RB*(SNE*SNP-CNE*SNG*CNP)
C     Y = RHO*COS(GAMMA)*SIN(PHI) - RB*SIN(ETA)*COS(PHI) -
C     RB*COS(ETA)*SIN(GAMMA)*SIN(PHI)
      Y=RHO*CNG*SNP-RB*(SNE*CNP+CNE*SNG*SNP)
C     Z = EPSM + RHO*SIN(GAMMA) + RB*COS(ETA)*COS(GAMMA)
      Z=EPSM+RHO*SNG+RB*CNE*CNG
C
C     POLAR COORDINATES.
C     X = X.
C     R = SQRT(Y**2+Z**2)
      R=SQRT(Y**2+Z**2)
C     THETA = ARCTAN(-Y/Z).
C     GUARD AGAINST ZERO DIVIDE.
      IF (ABS(Z).LT.1.0E-6) Z=1.0E-6
      YZ=ABS(Y/Z)
      THETA=ATAN(YZ)
C     ANGLE GREATER THAN PI/2.
      IF (Z.LT.0.) THETA=3.1415927-THETA
C     NEGATIVE ANGLE.
      IF (Y.GT.0.) THETA=-THETA
```

```
RETURN
END
```

```
      SUBROUTINE ZERN(R,R1,R2,T1,T2,AZ,A1,A2,A3,A)
      DIMENSION A(10),Z(10)
C     ROUTINE TO CALCULATE OPTICAL PROPERTIES OF PHASE DISTORTION IN
C     TERMS OF ZERNICKE POLYNOMIALS.
C     BY G. N. VANDERPLAATS                               MAY, 1977.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C     PHASE DISTORTION IS ASSUMED OF THE FORM AZ + A1*R + A2*T + A3*R*T
C     WHERE R = RADIUS AND T = THETA IN RADIANS.
C --- INPUT.
C     R = BEAM RADIUS.
C     R1, T1 = LOWER LIMITS OF INTEGRATION.
C     R2, T2 = UPPER LIMITS OF INTEGRATION.
C     AZ, A1, A2, A3 = POLYNOMIAL COEFFICIENTS.
C     A = VECTOR OF ZERNICKE COEFFICIENTS.  ON FIRST CALL TO ZERN A MUST
C         BE ZERO.
C ---- OUTPUT.
C     A = UPDATED VECTOR OF ZERNICKE COEFFICIENTS.
C
      DO 20 I=1,4
      GO TO (21,22,23,24),I
21    CALL ZINT(R,R1,T1,AZ,A1,A2,A3,Z)
      SIGN=1.
      GO TO 25
22    CALL ZINT(R,R1,T2,AZ,A1,A2,A3,Z)
      SIGN=-1.
      GO TO 25
23    CALL ZINT(R,R2,T1,AZ,A1,A2,A3,Z)
      SIGN=-1.
      GO TO 25
24    CALL ZINT(R,R2,T2,AZ,A1,A2,A3,Z)
      SIGN=1.
25    CONTINUE
      DO 30 J=1,10
30    A(J)=A(J)+SIGN*Z(J)
20    CONTINUE
      RETURN
      END
```

SUBROUTINE ZINT SEPT. 77

```
      SUBROUTINE ZINT(CAPR,R,THETA,AZ,A1,A2,A3,Z)
      DIMENSION Z(10)
C     ROUTINE TO EVALUATE INTEGRAL OF ZERNICKE POLYNOMIAL TIMES PHASE
C     DISTORTION AT R AND THETA.
C     BY G. N. VANDERPLAATS                           MAY, 1977.
C     NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C
C     CONSTANTS.
      CR2=CAPR*CAPR
      CR3=CR2*CAPR
      CR4=CR2*CR2
      R2=R*R
      R3=R2*R
      R4=R2*R2
      R5=R*R4
      R6=R3*R3
      T2=THETA*THETA
      ST=SIN(THETA)
      ST2=ST*ST
      ST3=ST*ST2
      S2T=SIN(2.*THETA)
      S3T=SIN(3.*THETA)
      CT=COS(THETA)
      CT2=CT*CT
      CT3=CT*CT2
      C2T=COS(2.*THETA)
      C3T=COS(3.*THETA)
      PI=3.1415927
      SQPI=SQRT(PI)
      Z(1)=R2*THETA*(.5*AZ+A1*R/3.+THETA*(.25*A2+A3*R/6.))/(CAPR*SQPI)
C     TILT.
      Z(2)=2.*R3*(ST*(AZ/3.+.25*R*A1)+(CT+THETA*ST)*(A2/3.+.25*A3*R))/
     * (CR2*SQPI)
      Z(3)=-2.*R3*((AZ/3.+.25*A1*R)*CT-(ST-THETA*CT)*(A2/3.+.25*A3*R))/
     * (CR2*SQPI)
C     FOCUS.
      Z(4)=3.4641016*R2*THETA*(.25*(R2-CR2)*(AZ+.5*A2*THETA)+
     * R*(A1+.5*A3*THETA)*(.2*R2-CR2/6.))/(SQPI*CR3)
C     ASTIGMATISM.
      Z(5)=2.4494897*R4*(.5*(.25*AZ+.2*A1*R)*S2T+.5*(.25*A2+.2*A3*R)*
     * (THETA*S2T+.25*C2T))/(SQPI*CR3)
      Z(6)=2.4494897*R4*(ST2*(.25*AZ+.2*A1*R)+(.25*A2+.2*A3*R)*
     * (THETA*ST2-.5*THETA+.25*S2T))/(SQPI*CR3)
C     COMA.
      B1=.2*AZ+A1*R/6.
      B2=.2*A2+A3*R/6.
      Z(7)=2.828427*R5*(B1*(ST*(CT2+2.)/3.-ST3)+B2*(3.*THETA*S3T+C3T+
     * 27.*THETA*ST+3.*CT-36.*THETA*ST3-12.*CT*ST2)/36.)/(SQPI*CR4)
      Z(8)=2.8284271*R5*(B1*(CT3-CT*(ST2+2.)/3.)+B2*(3.*THETA*C3T-S3T-
     * 27.*THETA*CT+3.*ST+36.*THETA*CT3-12.*ST*CT2)/36.)/(SQPI*CR4)
```

APPENDIX C

DATA FORMS

COPES DATA

DATA BLOCK A

	TITLE	FORMAT
*		20A4

DATA BLOCK B

+	$								COMMENT
	NCALC	NDV	NSV	N2VAR	IPNPUT	IPSENS	IP2VAR		FORMAT
*									8I10

DATA BLOCK C - OMIT IF NDV = 0

+	$								COMMENT
	IPRINT	ITMAX	ICNDIR	NSCAL	ITRM	LINOBJ	NACMX1	NFDG	FORMAT
*									8I10

DATA BLOCK D - OMIT IF NDV = 0

+	$								COMMENT
	FDCH	FDCHM	CT	CTMIN	CTL	CTLMIN	THETA	PHI	FORMAT
*									8F10
	DELFUN	DABFUN							FORMAT
*									2F10

DATA BLOCK E - OMIT IF NDV = 0

+	$				COMMENT
	NDVTOT	IOBJ	SGNOBJ		FORMAT
*					2I10,F10

DATA BLOCK F - OMIT IF NDV = 0

+	$					COMMENT
	VLB	VUB	X	SCAL		FORMAT
*						4F10

COPES DATA CONT.

DATA BLOCK G - OMIT IF NDV = 0

+	$				COMMENT
	NDSGN	IDSGN	AMULT		FORMAT
*					2I10,F10

DATA BLOCK H - OMIT IF NDV = 0

+	$		COMMENT
	NCONS		FORMAT
*			I10

DATA BLOCK I - OMIT IF NDV = 0 OR NCONS = 0

+	$					COMMENT
	ICON	JCON	LCON			FORMAT
*						3I10
+	$					COMMENT
	BL	SCAL1	BU	SCAL2		FORMAT
*						4F10

DATA BLOCK I – CONT.

DATA BLOCK J - OMIT IF NSV = 0

+	$								COMMENT
	NSOBJ								FORMAT
*									I10
+	$								COMMENT
	NSN1	NSN2	NSN3	NSN4	NSN5	NSN6	NSN7	NSN8	FORMAT
*									8I10

DATA BLOCK K - OMIT IF NSV = 0

+	$								COMMENT
	ISENS	NSENS							FORMAT
*									2I10
+	$								COMMENT
	SNS1	SNS2	SNS3	SNS4	SNS5	SNS6	SNS7	SNS8	FORMAT
*									8F10

+	$								COMMENT
	ISENS	NSENS							FORMAT
*									2I10
+	$								COMMENT
	SNS1	SNS2	SNS3	SNS4	SNS5	SNS6	SNS7	SNS8	FORMAT
*									8F10

COPES DATA - CONT.

DATA BLOCK K - CONT.

+	$								COMMENT
	ISENS	NSENS							FORMAT
*									2I10
+	$								COMMENT
	SNS1	SNS2	SNS3	SNS4	SNS5	SNS6	SNS7	SNS8	FORMAT
*									8F10

+	$								COMMENT
	ISENS	NSENS							FORMAT
*									2I10
+	$								COMMENT
	SNS1	SNS2	SNS3	SNS4	SNS5	SNS6	SNS7	SNS8	FORMAT
*									8F10

DATA BLOCK L - OMIT IF N2VAR = 0

+	$					COMMENT
	N2VX	M2VX	N2VY	M2VY		FORMAT
*						4I10

DATA BLOCK M - OMIT IF N2VAR = 0

+	$								COMMENT
	NZ1	NZ2	NZ3	NZ4	NZ5	NZ6	NZ7	NZ8	FORMAT
*									8I10

DATA BLOCK N - OMIT IF N2VAR = 0

+	$								COMMENT
	X1	X2	X3	X4	X5	X6	X7	X8	FORMAT
*									8F10

DATA BLOCK O - OMIT IF N2VAR = 0

+	$								COMMENT
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	FORMAT
*									8F10

DATA BLOCK P

	END		FORMAT
*	END		3A1

LASER TURRET ANALYSIS DATA

DATA BLOCK A

	TITLE	FORMAT
*		20A4

DATA BLOCK B

	AMACH	DENRTO	TDENRT	DENGAM	AKPRIM	WAVEL		FORMAT
*								6F10

DATA BLOCK C

	RFUS	AL	THMAX	ACL	EPS		FORMAT
*							5F10

DATA BLOCK D

	MAXK	MAXP	NXBC	NTHBC		FORMAT
*						4I10

DATA BLOCK E

	ABAR0	ABAR1	ABAR2	ABAR3	ABAR4	ABAR5	ABAR6	ABAR7	FORMAT
*									8F10

DATA BLOCK F

	X	YBC	YPBC		FORMAT
*					3F10

DATA BLOCK G

	BBAR0	BBAR 1	BBAR2	BBAR3	BBAR4	BBAR5	BBAR6	BBAR7	FORMAT
*									8F10

DATA BLOCK H

	THETA	YBC	YPBC		FORMAT
*					3F10

LASER TURRET ANALYSIS DATA - CONT.

DATA BLOCK I

	EPSM	XM		FORMAT
*				2F10

DATA BLOCK J

	NETAI	NRBI		FORMAT
*				2I10

DATA BLOCK K

	ETA1	ETA2	ETA3	ETA4	ETA5	ETA6	ETA7	ETA8	FORMAT
*									8F10

DATA BLOCK L

	RB1	RB2	RB3	RB4	RB5	RB6	RB7	RB8	FORMAT
*									8F10

DATA BLOCK M

	NBEAM		FORMAT
*			I10

DATA BLOCK N

	PHI	GAMMA	AMACHI	WGHT		FORMAT
*						4F10

DISTRIBUTION LIST

1.	Lt. Col. K. Gilbert AFWL/LRO Kirtland AFB, NM 87117	5
2.	Library Naval Postgraduate School Monterey, CA 93940	1
3.	Research Administration Naval Postgraduate School Monterey, CA 93940	1
4.	Defense Documentation Center Cameron Station Alexandria, VA 22314	2
5.	Dr. Garret Vanderplaats Code 1613 David Taylor Naval Ship Research and Development Center Bethesda, MD 20084	4
6.	Professor Allen E. Fuhs, Chairman Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93940	4
7.	Dr. B. Hogge AFWL Kirtland AFB, NM 87117	1