NASA CR-66837

UTILIZATION REPORT ON
STATISTICAL TRAJECTORY ESTIMATION PROGRAMS

By William E. Wagner and Arno C. Serold

Prepared under Contract No. NAS1-8500 by
MARTIN MARIETTA CORPORATION
Denver, Colorado

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA CR-66837

UTILIZATION REPORT ON

STATISTICAL TRAJECTORY ESTIMATION PROGRAMS

By William E. Wagner and Arno C. Serold

Prepared under Contract No. NAS1-8500 by
MARTIN MARIETTA CORPORATION
Denver, Colorado

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

FOREWORD

This report describing the utilization of the Statistical Trajectory Estimation Programs is provided in accordance with Part IV.A.4 of NASA Contract NAS1-8500. An additional report, provided in accordance with this same contract, is presented in NASA CR-1482 to describe the formulation of the Statistical Trajectory Estimation Program.

This work was conducted under the direction of Robert J. Mayhue, Technical Monitor; Sherwood Hoffman, Alternate Monitor (both of the Applied Materials and Physics Division); and George B. Boyles, Computer Analyst, Analysis and Computation Division, Langley Research Center, National Aeronautics and Space Administration.

CONTENTS

UTILIZATION REPORT ON

STATISTICAL TRAJECTORY ESTIMATION PROGRAMS

By William E. Wagner and Arno C. Serold
Martin Marietta Corporation

SUMMARY

This volume documents the computer program logic, subroutine arrangement, input and output formats, and other information concerning the Statistical Trajectory Estimation Programs (STEP) of interest to the user and programer. These programs were originally developed and used on the U.S. Air Force Precision Recovery Including Maneuvering Entry (PRIME) program to perform the postflight trajectory reconstruction and analysis of the SV-5D maneuverable lifting reentry vehicle. Subsequently, the programs have been considerably improved under NASA Contract NAS1-8500.

STEP uses the recursive Kalman minimum variance filtering algorithms to fit the equations of motion to trajectory measurement data. The programs are formulated to process position radar tracking and airborne gyro and accelerometer measurements. The equations of motion account for three dimensional trajectories in the vicinity of an oblate rotating planet. Vehicle maneuvers in pitch, roll, and yaw within the atmosphere are acceptable. STEP1 is restricted to nonthrusting vehicles; STEP2 is applicable to any vehicle recording accelerations and inertial angular rates and having at least partial radar coverage.

In addition to postflight reconstruction, the programs are useful for solving preflight trajectory simulation and error analysis problems.

I. INTRODUCTION

The Statistical Trajectory Estimation Programs (STEP1 and STEP2) were originally developed, validated, and successfully used to reconstruct the trajectories of the SV-5D maneuverable lifting reentry vehicles on the U.S. Air Force PRIME program. The original formulation is presented in reference 1.

Subsequently, under NASA Contract NAS1-8500 with Langley Research Center, the original formulation for STEP was modified, and the programs were rebuilt. The modifications consist of changing the translational equations of motion to inertial axes and using quaternions to describe the vehicle attitude. Several additional options to improve utility were also included. The improved formulation is presented in Volume I of this report.

From their first conception, the following six rules have served as guidelines for the development of STEP:

1) Keep all user/program interfaces simple and logical;
2) Use FORTRAN IV or FORTRAN 2.0 coding throughout to facilitate modifications and conversion to various computers;
3) Keep STEP1 and STEP2 as identical as possible;
4) Minimize logic that limits program applications to specific vehicles;
5) Use an executive program structure wherein the main program controls logic flow to and from numerous subordinate subroutines;
6) Be continually concerned with minimizing program execution time and computer core requirements.

As a consequence of these guidelines, the original programs were operated on the IBM 7094/2, GE-635, IBM-360/65, and CDC-6400 computers. The majority of subroutines in the two programs are identical, program input and output are nearly alike, and both programs can interface with the same data tapes.

In the following sections; information pertenent to the user and programer are presented. This includes descriptions of the input and output, checkout procedures, program logic, FORTRAN symbols, flow diagrams, and listings. Reference will be made frequently to the variables, equations, modes of operation, and computational procedures described in Volume I of this report. It is recommended that the reader be familiar with this material before proceeding into this volume--the Utilization Report.

II. PROGRAM INPUT AND OUTPUT

In Volume I of this report, the mathematical techniques and procedures used in STEP are described. Briefly, the concept consists of fitting the equations of motion to trajectory data in a minimum variance sense. The equations of motion in STEP1 require the inertial angular rate data, P_M, Q_M, and R_M; and, in STEP2 the inertial angular rate data and accelerometer data, a_{XM}, a_{YM}, and a_{ZM}. These data are satisfied exactly by the equations of motion, so obviously they must be excluded from the data to which the equations of motion are fit. As a result of the two different ways in which data are used, STEP interfaces with two magnetic tapes. The first tape, the PQR tape, contains the inertial angular rate and accelerometer data required by the equations of motion in STEP2. The second tape, the FIT tape, contains the radar tracking and accelerometer data as well as any additional data to which the equations of motion in STEP1 are to be fit. Note that the PQR tape can be used in STEP1, and the FIT tape in STEP2 if the accelerations are neglected.

STEP also requires a priori estimates, covariances, and correlations of the state variables and model parameters; program controls and numerical integration information; geophysical and gravitational constants; vehicle physical and aerodynamic data; atmospheric data; etc. These data are input by cards. Figure 1 is a schematic of the input and output information for STEP1 and STEP2. The figure shows the data required by the equations of motion entering from the left, the data to which the equations of motion are fit in a minimum variance sense entering from the top, and the output exiting toward the right. Also shown are those data obtained from cards and those from tape. In the following subsections, these inputs and outputs are described.

A. Card Input

The program card inputs have been divided into 20 logical categories. The quantities input in each of these categories are discussed below. The categories should be input in numerical order. They contain data consisting of both fixed and floating point numbers. The fixed point numbers, designated by an asterisk in the following description, must be rightmost justified within the column field specified. Floating point numbers must be rightmost justified only if the E-mode is used, e.g., .2574532E+3.

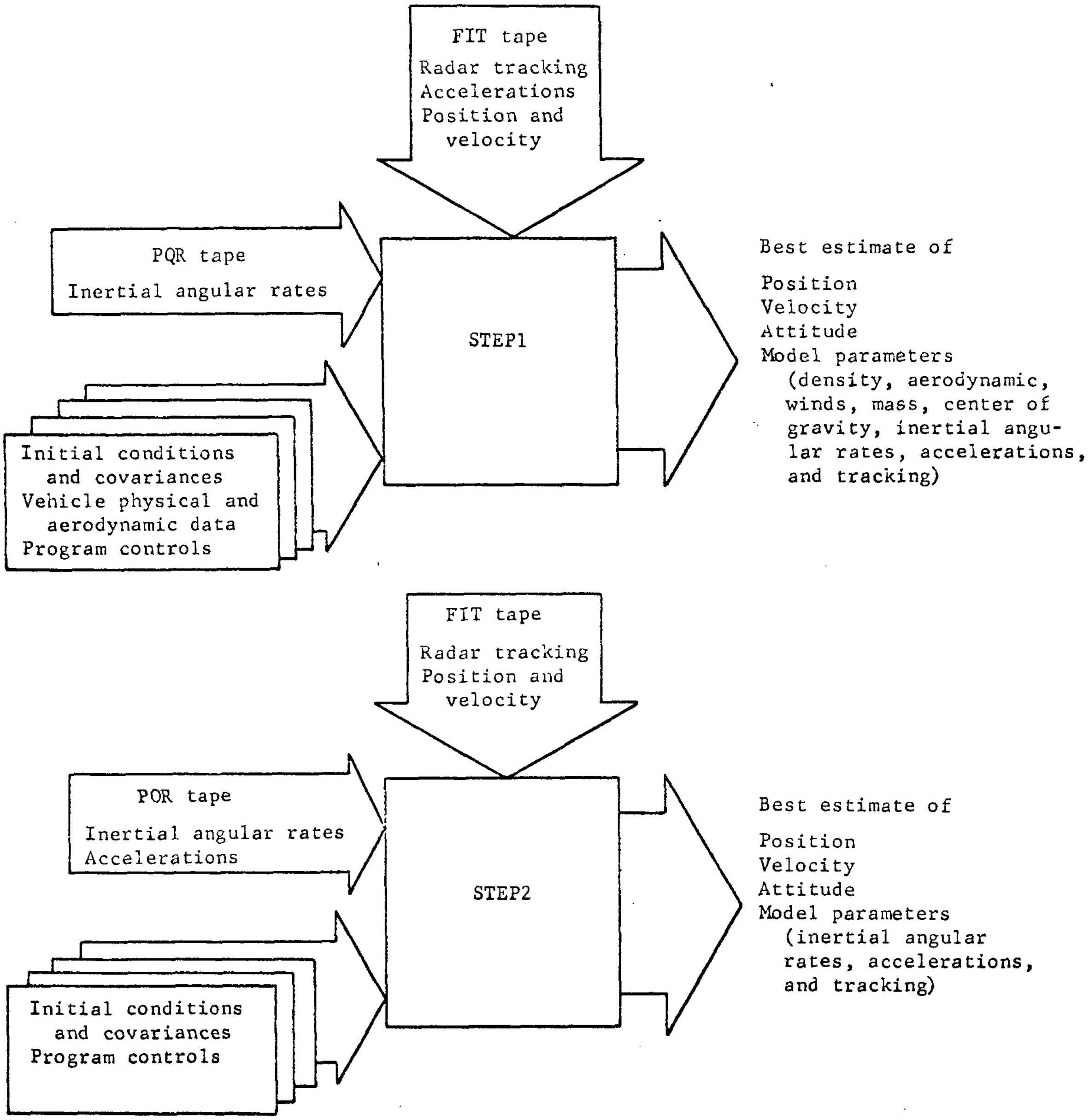


Figure 1.- Schematic of STEP Inputs and Outputs

When floating point numbers are input in the F-mode, e.g., 257.4532, they can lie anywhere within the column field specified. Linear interpolation and extrapolation are used on all tabular input data. Therefore, at least two points must be specified for each table. Curves to be maintained constant at zero need not be input.

1. Input categories.- Those input categories common only to STEP1 will be noted in the description that follows.

Category 1 comment card (STEP1 and STEP2):

Variables	Columns	Description
1*¶	2	Identifies category 1 input
"comment"	3-72	Comment containing alphabetic and numerical characters to describe the problem, date, etc.

As many comment cards as desired may be input, each however, requiring a 1 in column 2 (a 0 or blank will also work). The comment cards will be printed on the tab printout.

Category 2 program controls (STEP1 and STEP2):

Variable	Column	Description
2*	2	Identifies category 2 input
NPC(1)*	6	= 0 filtering, = 1 deterministic, = 2 error analysis
NPC(2)*	10	= 0 metric units, = 1 English units
NPC(3)*	14	= 0, 1, or 2 for input/output in internal variables or types 1 or 2 variables, respectively
NPC(4)*	18	= 0 updated reference, = 1 nonupdated reference
NPC(5)*	22	= 0 print fitting data schedule, = 1 do not print
NPC(6)*	26	= 0 do not print covariance and correlation matrices, = 1 print covariance and correlation matrices during smoothing on last iteration = 2 print covariance and correlation matrices during filtering and smoothing on last iteration
NPC(7)*	30	Number of iterations
NPC(8)*	34	= 0 smooth state only, no residual or loss function calculations = 1 smooth state only, calculate residuals and loss function = 2 smooth state and covariances, no residuals or loss function calculations = 3 smooth state and covariances, calculate residuals and loss function = 4 do not smooth
NPC(9)*	38	= 0 vector process, = 1 scalar process
NPC(10)*	42	= 0 do not write STATE tape during smoothing, = 1 write tape

¶A variable followed by an asterisk indicates a fixed point integer which must be rightmost justified.

These program controls pertain to entire program operation. For more complete definitions, see Subsection A.2. Program controls common to a specific input category are input with that category.

Category 3 state variables and model parameters (STEP1 and STEP2):

Variable	Column	Description
3*	2	Identifies category 3 input
IPC_1^*	4	Variable in X=1, W=2, U=3, V=4, Const=5
IDN*	5-8	Variable identification number, see Subsection A.4, page 21
VALUE	9-20	Mean values of state variable or model parameter, see Subsection A.4 for units
SIG	21-32	Standard deviation of state or model parameter if IPC_1 = 1-4, see Subsection A.4 for units

One category 3 card is required for each state variable and model parameter desired in the problem. Should no card be input for a specific model parameter, it assumes a preset value of 1 or 0 to bypass the error model. For example, in the accelerometer error model, equation (159), $C_{61} = C_{65} = C_{69} = 1$ all other Cs are zero. For uncertain variables, the standard deviations are squared and constitute the variance of the diagonal covariance matrix. Should off-diagonal elements be required, they can be input in category 10. The maximum number of components in X is 10; in W is 20, in U is 5; in V is 5.

Category 4 numerical integration (STEP1 and STEP2):

Variable	Column	Description
4*	2	Identifies category 4 input
TO	9-20	Initial problem time, sec
DET	21-32	Time interval between block print-outs, sec
DCOMP	33-44	Fixed computing interval, sec
TFINAL	45-56	Final problem time, sec

Category 5 geophysical and gravitational (STEP1 and STEP2):

Variable	Column	Description
5*	2	Identifies category 5 input
1*	4	Identifies remaining variables on card as GO, XMU, and XJ2
GO	9-20	Acceleration of gravity at sea level for use in STEP1 atmosphere equations, m/sec^2, ft/sec^2
XMU	21-32	Coefficient of first gravitational harmonic, m^3/sec^2, ft^3/sec^2
XJ2	33-44	Coefficient of second gravitational harmonic
and/or		
5*	2	Identifies category 5 input
2*	4	Identifies remaining variables on card as R_P, R_E, and OMEGA
RPO	9-20	Polar radius of planet, m, ft
REO	21-32	Equatorial radius of planet, m, ft
OMEGA	33-44	Planet rotation rate, rad/sec

Category 5 cards need be input only if a change is desired from the built-in data. The values of the category 5 data that are built-in are from reference 2:

$$GO = g_o = 9.80665 \ m/sec^2 = 32.174048 \ ft/sec^2$$

$$XMU = \mu = 3.985992 \times 10^{14} \ m^3/sec^2 = 1.407639 \times 10^{16} \ ft^3/sec^2$$

$$XJ2 = J_2 = 1082.645 \times 10^{-6}$$

$$RPO = R_P = 6\ 356\ 173 \ m = 20\ 853\ 599 \ ft$$

$$REO = R_E = 6\ 378\ 163 \ m = 20\ 925\ 731 \ ft$$

$$OMEGA = \Omega_P = 7.292116 \times 10^{-5} \ rad/sec$$

Category 6 center-of-gravity/IMU distances (STEP1 and STEP2):

Variable	Column	Description
6*	2	Identifies category 6 input
IPC_2^*	4	Identifies table to follow = 1 $x_p(t)$, = 2 $y_p(t)$, = 3 $z_p(t)$
$NPTS^*$	5-8	Number of points on table

[Read in $x_p(t)$, $y_p(t)$, or $z_p(t)$ (m, ft), depending on IPC_2, using table input format 1 in Subsection A.3; NPTS must be no larger than 20]

A separate category 6 card precedes each of the tables. Zero tables are omitted.

Category 7 fitting data controls (STEP1 and STEP2):

Variable	Column	Description
7*	2	Identifies category 7 input
MTYP*	4	Identifies type of data = 1, 2, 3, 4, 5 tracking stations = 6 airborne radar, = 7 position, = 8 velocity, = 9 accelerometer
TI	5-14	Initial time to process MTYP data, sec
TF	15-24	Final time to process MTYP data, sec
DT	25-34	Minimum time span between MTYP data points, sec
MR(1,MTYP)*	35-36	= 0 do not process component 1, = 1 process component 1
MR(2,MTYP)*	37-38	= 0 do not process component 2, = 1 process component 2
MR(3,MTYP)*	39-40	= 0 do not process component 3, = 1 process component 3
SIG(1,MTYP)	41-50	Incremental standard deviation on component 1, m, ft, rad
SIG(2,MTYP)	51-60	Incremental standard deviation on component 2, m, ft, rad
SIG(3,MTYP)	61-70	Incremental standard deviation on component 3, m, ft, rad

Input one category 7 card for each type (MTYP) of data to be processed. Category 7 cards are used to control the frequency and time span of data on the FIT tape. Specified components of the data triples can be edited out (even though on the FIT tape) by specifying its corresponding MR input equal to 0. The incremental standard deviations are added to the standard deviations on the FIT tape to yield the total standard deviations used in the filtering equations. All angular standard deviations (e.g., A_M and E_M) are in radians.

In error analysis problems, the category 7 cards provide all the fitting data information necessary, i.e., time span, frequency, and standard deviation. Therefore, no FIT tape is used.

Category 8 FIT tape (STEP1 and STEP2):

Variable	Column	Description
8*	2	Identifies category 8 input
IPC_3^*	4	= 0 no print of FIT tape, = 1 print FIT tape
IPC_4^*	8	= 1 input FIT data from cards, = 0 do not enter FIT data from cards

(If $IPC_4 = 1$, read in FIT data from cards using table input format 2 in Subsection A.3)

The category 8 cards permit printing of the FIT tape and/or input of the FIT data from cards. When card input is used, a tape is written from the card inputs and later used in problem execution. If a prepared FIT tape is to be used and no printout of the tape is desired, omit category 8 input.

Category 9 POR tape (STEP1 and STEP2):

Variable	Column	Description
9*	2	Identifies category 9 input
IPC_5^*	4	= 0 no print of PQR tape, = 1 print PQR tape
IPC_6^*	8	= 1 input PQR data from cards, = 0 do not enter PQR data from cards

(If $IPC_6 = 1$, read PQR data from cards using table format 3 in Subsection A.3)

The category 9 cards permit printing of the PQR tape and/or input of the PQR data from cards. When card input is used, a tape is written from the card inputs and later used in problem execution. If a prepared PQR tape is desired, omit category 9 input.

Category 10 covariance and correlation matrices (STEP1 and STEP2):

Variable	Column	Description
10*	1-2	Identifies category 10 input
IPC_7^*	4	Identifies matrix or matrix elements to follow = 1 P, =2 CUZ, =3 CVZ = 4 P(i,j), = 5 CUZ(i,j) = 6 CVZ(i,j)
NPTS*	8	Number of single point cards to follow if $IPC_7 > 3$

[Read in P, CUZ, or CVZ (m, ft, rad) if IPC_7 = 1, 2, or 3, respectively, by means of table format 4 in Subsection A.3]

or input NPTS cards each containing a single point formatted as follows

Variable	Column	Description
i*	5-8	if $IPC_7 = 4$
j*	9-12	
P(i,j)	13-24	

or

Variable	Column	Description
i*	5-8	if $IPC_7 = 5$
j*	9-12	
CUZ(i,j)	13-24	

or

Variable	Column	Description
i*	5-8	if $IPC_7 = 6$
j*	9-12	
CVZ(i,j)	13-24	

The category 10 data must be input after all category 3 cards have been input. This is necessary to allow the program to determine the number of components in the expanded state vector Z, model parameters U, and measurement parameters V. The size of P, CUZ, and CVZ require that these vector sizes be known. Care should be exercised in assuring that the matrices input in category 10 are compatible both in size and the order of variables input in category 3. The category 10 input can be omitted if the diagonal matrix P input in category 3 and null correlation matrices CUZ and CVZ are acceptable.

Category 11 aerodynamic coefficients (STEP1 only):

Variable	Column	Description
11*	1-2	Identifies category 11 input
IPC_8^*	4	Aerodynamic input option indicator $= -1\ C_D,\ C_Y,\ C_L;\ = 0\ C_A,\ C_Y,\ C_N;$ $= 1\ C_A,\ C_{Y_\eta},\ C_{N_\eta}$
SREF	9-21	Reference area for aerodynamic coefficients
XLREF	22-33	Reference length in Reynolds number

(Input all three tables of aerodynamic coefficients by means of table input format 5 in Subsection A.3)

Category 12 mass (STEP1 only):

Variable	Column	Description
12*	1-2	Identifies category 12 input
NPTS*	8	Number of points on table, the mass table to follow
TØNE	9-20	Time t_1 to commence mass error model correction
TTWO	21-32	Time t_2 at which mass error model correction is stopped

[Read in m(t) (kg, slugs) by means of table input format 1 in Subsection A.3]

Category 13 atmospheric winds (STEP1 only):

Variable	Column	Description
13*	1-2	Identifies category 13 input
IPC_{10}^*	4	Identifies wind profile to follow $= 0\ u_{WM}(h_o),\ = 1\ v_{WM}(h_o)$
NPTS*	8	Number of points on table

[Read in $u_{WM}(h)$ or $v_{WM}(h)$ (mps or fps), depending on IPC_{10}, by means of table input format 1 of Subsection A.3]

Category 14 atmosphere (STEP1 only):

Variable	Column	Description
14	1-2	Identifies category 14 input
IPC_{11}^{*}	4	Identifies atmosphere option = 0 1962 U.S. standard, = 1 1959 ARDC, = 2 arbitrary atmosphere to be input as function of geopotential altitude, = 3 arbitrary atmosphere to be input as function of geometric altitude
	If $IPC_{11} \geq 2$	input the following
NPTS*	3-4	Number of base points on attitude profile to be input
APO	5-15	Sea level pressure, nt/m^2 or lb_f/ft^2
AMO	16-26	Mean molecular weight of atmosphere constituents
AR	27-37	Gas constant, J/(°K) (kg-mol) or ft/lb_f (°R) (slug-mol)
AGAM	38-48	Specific heat ratio of atmosphere constituents
ABET	49-59	Sutherlands constants β and S used to calculate atmosphere viscosity, see units on data below
AS	60-70	

[Input HB(i) (m, ft) and TMB(i) (°K, °R), i=1, NPTS by means of table input format 6 in Subsection A.3]

The data entered in category 14 input (if $IPC_{11} \geq 2$) must be dimensionally compatible with that specified in other categories. The values of AMO, APO, AR, AGAM, ABET, and AS built-in the program for use in the 1959 ARDC and 1962 U.S. Standard atmospheres are the following:

APO = 1.013250×10^5 nt/m^2 = 2116.2166 lb$_f$/ft^2

AMO = 28.9644

AR = 8.31432×10^3 J/(°K) (kg-mol) = 49.7192×10^3 ft-lb$_f$/(°R) (slug-mol)

AGAM = 1.40

ABET = 1.458×10^{-6} kg/sec m (°K)$^{\frac{1}{2}}$ = 2.269×10^{-8} slugs/ft-sec (°R)$^{\frac{1}{2}}$

AS = 110.4 °K = 198.72 °R

Category 20 end card:

Variable	Column	Description
20*	1-2	Identifies category 20 card

The category 20 card must always be input last.

2. Program control definitions.- Definitions of the program controls used in the previous section follow:

NPC(1) = 0 Filtering mode of operation;
1 Deterministic mode of operation;
2 Error analysis mode of operation.

NPC(2) = 0 Input and output in metric units (kg, m, sec);
1 Input and output in English units (slug, ft, sec).

NPC(3) Input and output the following state variables, their standard deviations, covariance and correlation matrices;
= 0 u, v, w, h_o, φ_D, θ, e_0, e_1, e_2, e_3;
1 V_A, γ_A, λ_A, h_o, φ_D, θ, $\bar{\psi}$, $\bar{\theta}$, $\bar{\varphi}$;
2 V_A, γ_A, λ_A, h_o, φ_D, θ, σ, β, α.

NPC(4) = 0 Update reference after each measurement processed;
1 Do not update reference until final time is met.

NPC(5) = 0 Print schedule of processed fitting data;
1 Do not print schedule of processed fitting data.

NPC(6) = 0 Do not print covariance and correlation matrices;

1 Print covariance and correlation matrices P, CUZ, and CVZ after each block printout during smoothing on last iteration;

2 Print covariance and correlation matrices P, CUZ, and CVZ after each block printout during filtering and smoothing on last iteration.

NPC(7) = Number of filtering and smoothing iterations.

NPC(8) = 0 Smooth state only, no residuals nor loss function;

1 Smooth state only, calculate residuals and loss function;

2 Smooth state, covariance, and correlation matrices, no residuals nor loss function calculations;

3 Smooth state, covariance, and correlation matrices, calculate residuals, and loss function;

4 Do not smooth.

[Before the last iteration, only the state is smoothed if NPC(8) = 2 or 3]

NPC(9) = 0 Process data as vectors where possible;

1 Process data as scalars at all times.

NPC(10) = 0 Do not write STATE tape;

1 Write STATE tape on backward smoothing of last iteration.

IPC_1 = 1 Variable is a component of the state vector, X;

2 Variable is a model parameter to be estimated in the expanded state vector, Z;

3 Variable is an uncertain model parameter not to be estimated, U;

4 Variable is an uncertain measurement parameter not to be estimated, V;

5 Variable is a model parameter known with absolute certainty, C.

IPC_2 = 1 Indicates the $x_p(t)$ table will follow category 6 card;

2 Indicates the $y_p(t)$ table will follow category 6 card;

	3	Indicates the $z_p(t)$ table will follow category 6 card.
$IPC_3(IPC_5)$	= 0	Do not print out the data on the FIT (PQR) tape;
	1	Print out the data on the FIT (PQR) tape in tabular form before commencing the problem execution.
$IPC_4(IPC_6)$	= 0	Write the FIT (PQR) tape from input cards;
	1	The FIT (PQR) tape has already been prepared and need not be written from cards.
IPC_7	= 1	The covariance matrix P follows the category 10 card;
	2	The covariance matrix CUZ follows the category 10 card;
	3	The covariance matrix CVZ follows the category 10 card;
	4	Single points P(i,j) follows the category 10 card;
	5	Single points CUZ(i,j) follows the category 10 card;
	6	Single points CVZ(i,j) follows the category 10 card.
IPC_8	= -1	Aerodynamic coefficient inputs are $C_D(\alpha, \beta, M)$, $C_Y(\alpha, \beta, M)$ and $C_L(\alpha, \beta, M)$;
	0	Aerodynamic coefficient inputs are $C_A(\alpha,\beta,M)$, $C_Y(\alpha,\beta,M)$ and $C_N(\alpha,\beta,M)$;
	1	Aerodynamic coefficient inputs are $C_A(\xi,\eta,M)$, $C_{Y_\eta}(\xi,\eta,M)$ and $C_{N_\eta}(\xi,\eta,M)$.
IPC_{10}	= 0	Indicates the $u_W(h_o)$ table follows category 13 card;
	1	Indicates the $v_W(h_o)$ table follows category 13 card.
IPC_{11}	= 0	Use a 1962 U.S. Standard atmosphere;
	1	Use a 1959 ARDC atmosphere;
	2	Input an arbitrary atmosphere as a function of geopotential altitude;
	3	Input an arbitrary atmosphere as a function of geometric altitude.

MTYP = 1 Tracking station 1 with components 1, 2, and 3 being R, A, and E, respectively;

2 Tracking station 2 with components 1, 2, and 3 being R, A, and E, respectively;

3 Tracking station 3 with components 1, 2, and 3 being R, A, and E, respectively;

4 Tracking station 4 with components 1, 2, and 3 being R, A, and E, respectively;

5 Tracking station 5 with components 1, 2, and 3 being R, A, and E, respectively;

6 Airborne radar with component 1 being R_R (components 2 and 3 are currently unused);

7 Velocity of vehicle with components 1, 2, and 3 being u, v, and w, respectively;

8 Position of vehicle with components 1, 2, and 3 being h, φ, and θ, respectively;

9 Accelerometer with components 1, 2, and 3 being a_{XM}, a_{YM}, and a_{ZM}, respectively.

3. Table input formats.- The formats described below are used to input tables in STEP.

a. Format 1: For a table, $X(t_i)$, having NPTS discrete values of the independent variable, t_i, and the dependent variable, $X(t_i)$, the following format is used.

Card \ Col	1-12	13-24	25-36	37-48	49-60	61-72
1	t_1	$X(t_1)$	t_2	$X(t_2)$	t_3	$X(t_3)$
2	t_4	$X(t_4)$	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
	t_{NPTS-1}	$X(t_{NPTS-1})$	t_{NPTS}	$X(t_{NPTS})$		

The number of points on the table NPTS is entered on the category card preceding the table.

b. Format 2: This format corresponds to the data on the FIT tape. The data can be input on cards in category 8 using the following format.

Card \ Col	1-2	3-4	5-14	15-24	25-34	35-44	45-54	55-64	65-74
1	K*								
2		MTP(1)*	TYM(1)	DAT(1,1)	DAT(2,1)	DAT(3,1)	SIG(1,1)	SIG(2,1)	SIG(3,1)
3		MTP(2)*	TYM(2)	DAT(1,2)	DAT(2,2)	DAT(3,2)	SIG(1,2)	SIG(2,2)	SIG(3,2)
.									
.									
.									
K+1		MTP(K)*	TYM(K)	DAT(1,K)	DAT(2,K)	DAT(3,K)	SIG(1,K)	SIG(2,K)	SIG(3,K)
	(The last record of input must contain zeros)								

Each record of the FIT tape corresponds to the K points described above. The number of points per record, K, must never exceed 20. As many records (with $K \leq 20$) as desired may be input. The last record must always consist of a single card having $K = 0$.

c. Format 3: This format correponds to the data on the PQR tape. The data can be input on cards in category 9 using the following format.

Card \ Col	1-2	3-12	13-22	23-32	33-42	43-52	53-62	63-72
1	K*							
2		TT(1)	TP(1)	TQ(1)	TR(1)	AX(1)	AY(1)	AZ(1)
.		.	.	.	.	.	.	.
.		.	.	.	.	.	.	.
.		.	.	.	.	.	.	.
K+1		TT(K)	TP(K)	TQ(K)	TR(K)	AX(K)	AY(K)	AZ(K)
	(The last record of input must contain zeros)							

Each record on the PQR tape corresponds to the K points described above. The number of points per record, K, must never exceed 20. As many records (with $K \leq 20$) as desired may be input. The last record must always consist of a single card having $K = 0$.

d. Format 4: This format corresponds to the covariance and correlation matrices input in category 10. These matrices are input by rows using the following format.

Card \ Col	1-12	13-24	25-36	37-48	49-60	61-72
1	P(1,1)	P(1,2)	P(1,3)	P(1,4)	P(1,5)	P(1,6)
.	P(1,7)	.	.	P(1,m)	P(2,1)	P(2,2)
.	.	.	.			
.	.	.	.			
.	.	.	P(m,m)			
1	CUZ(1,1)	CUX(1,2)	CUZ(1,3)	...	CUZ(1,q)	CUZ(2,1)
.	.	.	.			
.	.	.	.			
.	.	.	.			
.	.	.	CUZ(m,q)			
1	CVZ(1,1)	CVZ(1,2)	CVZ(1,3)	...	CVZ(1,r)	CVZ(2,1)
.	.	.	.			
.	.	.	.			
.	.	.	.			
.	.	.	.			
.	.	.	CVZ(m,r)			

The number of components of the expanded state vector, m, uncertain model parameters, q, and measurement parameters, r, are determined by counting the category 3 cards as they are input.

e. Format 5: This format is used to input the tables of aerodynamic coefficients in category 11. Each table commences with the following control card

Card \ Col	1-4	5-8	9-12	13-16	17-20
1	NT*	NV*	N_1*	N_2*	N_3*

where

NT = Table identification number = 1 for C_D or C_A,
= 2 for C_Y or C_{Y_η},
= 3 for C_L, C_N or C_{N_η};

NV = Number of independent variables for the table, $0 \leq NV \leq 3$;

N_1 = Number of points in the first independent variable, α or η;

N_2 = Number of points in the second independent variables, β or ξ;

N_3 = Number of point in the third independent variable, M.

The tabular entries follow. The first three cards list the N_1, α, or η values, the N_2, β, or ξ values and the N_3 M values. This is followed by the aerodynamic coefficients, specified by NT, which will be denoted by C below.

Card \ Col	1-6	7-12	13-18	23-24	25-30	31-36	37-42	43-48	49-54	55-60	61-66	67-72
1	α_1	α_2	α_3	α_4	α_5	.	.	.	α_{N1}			
2	β_1	β_2	.	.	.	β_{N2}						
3	M_1	M_2	M_3	M_4	.	.	.	M_{N3}				
4	$C(\alpha_1\beta_1M_1)$	$C(\alpha_2\beta_1M_1)$	$C(\alpha_3\beta_1M_1)$	.	.	.	$C(\alpha_{N1}\beta_1M_1)$	$C(\alpha_1\beta_2M_1)$	$C(\alpha_2\beta_2M_1)$	.	.	.
.	.	.	$C(\alpha_{N1}\beta_2M_1)$	$C(\alpha_1\beta_1M_2)$	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	$C(\alpha_{N1}\beta_{N2}M_{N3})$					

For NT = 3, replace α and β by η and ξ in the above table. The trivariate table is size limited to $N_1 \times N_2 \times N_3 \leq 512$, which amounts to eight values of α, eight values of β, and eight values of M. By specifying N_1 and/or N_2 and/or N_3 equal to zero, the trivariate table can be collapsed to a bivariate or monovariate table or a point. For bivariate or monovariate tables, the size limitation is still 512. However $N_i \times N_j \leq 512$ for monovariate tables allowing more points per curve to be specified.

f. Format 6: This format is used to input the base geopotential or geometric altitudes and molecular scale temperatures in category 14.

Card \ Col	1-12	13-24	25-36	37-48	49-60	61-72
1	HB(1)	HB(2)	HB(3)	HB(4)	HB(5)	HB(6)
2	HB(7)	.	.	.	HB(NPTS)	
.	TMB(1)	TMB(2)	TMB(3)	TMB(4)	TMB(5)	TMB(6)
.	TMB(7)	.	.	.	TMB(NPTS)	

The number of points in each table, NPTS, is input on the category 14 card. In the above table, HB(i) are the base geopotential attitudes and TMB(i) the base molecular scale temperature. The maximum number of points in these tables is NPTS $\leq$ 23.

<u>4. Variable identification numbers and input/output units</u>.- Variable identification numbers and input/output units are given in table 1.

TABLE 1.- VARIABLE IDENTIFICATION NUMBERS AND INPUT UNITS

IDN	Variable (for NPC(3) = 0)	Variable (for NPC(3) = 1)	Variable (for NPC(3) = 2)	Reference equation from vol. I	Vector component		Input units[a]		Output units[a]	
					STEP1	STEP2	Variable	Standard deviation	Variable	Standard deviation
State										
1	u	V_A	V_A		X	X	mps	mps	mps	mps
2	v	γ_A	γ_A	(127) thru (129)	X	X	mps(deg)	mps(rad)	mps(deg)	mps(rad)
3	w	λ_A	λ_A	(219)	X	X	mps(deg)	mps(rad)	mps(deg)	mps(rad
4	h_o	h_o	h_o	(229)	X	X	m	m	m	m
5	φ_D	φ_D	φ_D		X	X	deg (geod)[b]	rad (geoc)[b]	deg (geod)[b]	rad (geoc)[b]
6	θ	θ	θ		X	X	deg	rad	deg	rad
7	e_o	$\bar{\psi}$	σ		X	X	- (deg)	- (rad)	- (deg)	- (rad)
8	e_1	$\bar{\theta}$	β		X	X	- (deg)	- (rad)	- (deg)	- (rad)
9	e_2	$\bar{\Phi}$	α		X	X	- (deg)	- (rad)	- (deg)	- (rad)
10	e_3				X	X	----	----	----	----
Model parameters										
1	C_1 Aerodynamic			(147)	Z,U,C	↑	----	----	----	----
2	C_2 Error coefficients			↓	Z,U,C		----	----	----	----
3	C_3 ↓			↓	Z,U,C		----	----	----	----
4	C_4 ↓			↓	Z,U,C	Not included	rad^{-2}	rad^{-2}	rad^{-2}	rad^{-2}
5	C_5 ↓			↓	Z,U,C		rad^{-1}	rad^{-1}	rad^{-1}	rad^{-1}
6	C_6 ↓			↓	Z,U,C		rad^{-1}	rad^{-1}	rad^{-1}	rad^{-1}
7	C_7 ↓			↓	Z,U,C		----	----	----	----
8	C_8 ↓			↓	Z,U,C		----	----	----	----
9	C_9 ↓			↓	Z,U,C		----	----	----	----
10	C_{10} ↓			↓	Z,U,C		----	----	----	----
11	C_{11} ↓			↓	Z,U,C		----	----	----	----
12	C_{12} ↓			↓	Z,U,C		----	----	----	----
13	C_{13} Unused									
14	C_{14} ↓									
15	C_{15} ↓									
16	C_{16} Mass error			(149)	Z,U,C		kg	kg	kg	kg
17	C_{17} Coefficients			↓	Z,U,C		kg/sec	kg/sec	kg/sec	kg/sec
18	C_{18} ↓			↓	Z,U,C		kg/sec²	kg/sec²	kg/sec²	kg/sec²
19	C_{19} Unused									
20	C_{20} Unused									
21	C_{21} Density error			(151)	Z,U,C		----	----	----	----
22	C_{22} Coefficients			↓	Z,U,C		kg/m³	kg/m³	kg/m³	kg/m³
23	C_{23} ↓			↓	Z,U,C		1/m	1/m	1/m	1/m
24	C_{24} Unused									
25	C_{25} Unused									
26	C_{26} Atmosphere winds			(152)	Z,U,C		m/sec	m/sec	m/sec	m/sec
27	C_{27} Error coefficient			↓	Z,U,C		m/sec	m/sec	m/sec	m/sec
28	C_{28} Unused									
29	C_{29} Unused									
30	C_{30} Unused					↓				
31	C_{31} Center of gravity			(153)	Z,V,C	Z,U,C	m	m	m	m

[a]Only metric SI units are presented. For English units replace m by ft and kg by slugs.

[b]geod refers to geodetic latitude, geoc refers to geocentric latitude.

TABLE 1- VARIABLE IDENTIFICATION NUMBERS AND IMPUT UNITS - Continued

IDN	Variable	Reference equation from vol. I	Vector component		Input units[a]		Output units[a]	
			STEP1	STEP2	Variable	Standard deviation	Variable	Standard deviation
Model parameters								
32	C_{32} Error coefficients	↓	Z,V,C	Z,U,C	m	m	m	m
33	C_{33} ↓	↓	Z,V,C	Z,U,C	m	m	m	m
34	C_{34} Unused							
35	C_{35} Unused							
36	C_{36} Inertial	(154)	Z,U,C	Z,U,C	----	----	----	----
37	C_{37} Angular rate	↓	Z,U,C	Z,U,C	----	----	----	----
38	C_{38} Error coefficients	↓	Z,U,C	Z,U,C	----	----	----	----
39	C_{39} ↓	↓	Z,U,C	Z,U,C	----	----	----	----
40	C_{40}		Z,U,C	Z,U,C	----	----	----	----
41	C_{41}		Z,U,C	Z,U,C	----	----	----	----
42	C_{42}		Z,U,C	Z,U,C	----	----	----	----
43	C_{43}		Z,U,C	Z,U,C	----	----	----	----
44	C_{44}		Z,U,C	Z,U,C	----	----	----	----
45	C_{45}		Z,U,C	Z,U,C	rad/sec	rad/sec	rad/sec	rad/sec
46	C_{46}		Z,U,C	Z,U,C	rad/sec	rad/sec	rad/sec	rad/sec
47	C_{47}		Z,U,C	Z,U,C	rad/sec	rad/sec	rad/sec	rad/sec
48	C_{48}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
49	C_{49}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
50	C_{50}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
51	C_{51}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
52	C_{52}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
53	C_{53}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
54	C_{54}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
55	C_{55}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
56	C_{56}		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
57	C_{57} ↓		Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
58	C_{58} Unused	↓	Z,U,C	Z,U,C	rad sec/m	rad sec/m	rad sec/m	rad sec/m
59	C_{59} Unused							
60	C_{60} Unused							
61	C_{61} Acceleration	(159)	Z,V,C	Z,U,C	----	----	----	----
62	C_{62} Error coefficients	↓	Z,V,C	Z,U,C	----	----	----	----
63	C_{63} ↓		Z,V,C	Z,U,C	----	----	----	----
64	C_{64}		Z,V,C	Z,U,C	----	----	----	----
65	C_{65}		Z,V,C	Z,U,C	----	----	----	----
66	C_{66}		Z,V,C	Z,U,C	----	----	----	----
67	C_{67}		Z,V,C	Z,U,C	----	----	----	----
68	C_{68}		Z,V,C	Z,U,C	----	----	----	----
69	C_{69}		Z,V,C	Z,U,C	----	----	----	----
70	C_{70}		Z,V,C	Z,U,C	m/sec^2	m/sec^2	m/sec^2	m/sec^2
71	C_{71}		Z,V,C	Z,U,C	m/sec^2	m/sec^2	m/sec^2	m/sec^2
72	C_{72} ↓	↓	Z,V,C	Z,U,C	m/sec^2	m/sec^2	m/sec^2	m/sec^2

[a]Only metric SI units are presented. For English units replace m by ft and kg by slugs.

TABLE 1.- VARIABLE IDENTIFICATION NUMBERS AND INPUT UNITS - Concluded

IDN	Variable		Reference equation from vol. I	Vector component		Input units[a]		Output units[a]	
				STEP1	STEP2	Variable	Standard deviation	Variable	Standard deviation
Model parameters									
73	C_{73}	Unused							
74	C_{74}	Unused							
75	C_{75}	Unused							
76	C_{76}	Tracking	(239)	Z,V,C	Z,V,C	----	----	----	----
77	C_{77}	Station 1	and	Z,V,C	Z,V,C	----	----	----	----
78	C_{78}	Error coefficients	table 11	Z,V,C	Z,V,C	---	---	---	---
79	C_{79}	↓	↓	Z,V,C	Z,V,C	m	m	m	m
80	C_{80}			Z,V,C	Z,V,C	rad	rad	rad	rad
81	C_{81}			Z,V,C	Z,V,C	rad	rad	rad	rad
82	C_{82}			Z,V,C	Z,V,C	sec	sec	sec	sec
83	C_{83}			Z,V,C	Z,V,C	sec	sec	sec	sec
84	C_{84}			Z,V,C	Z,V,C	sec	sec	sec	sec
85	C_{85}			Z,V,C	Z,V,C	ft	ft	ft	ft
86	C_{86}			Z,V,C	Z,V,C	rad	rad	rad	rad
87	C_{87}	Unused							
88	C_{88}	Latitude of station 1		Z,V,C	Z,V,C	deg(geod)[b]	rad(geoc)[b]	rad(geoc)[b]	rad(geoc)[b]
89	C_{89}	Longitude of station 1		Z,V,C	Z,V,C	deg	rad	rad	rad
90	C_{90}	Altitude of station 1	↓	Z,V,C	Z,V,C	m	m	m	m
91	C_{91}	Error coefficients	table 11	Z,V,C	Z,V,C				
.	.	for tracking sta-		.	.				
.	.	tion 2,3,4, & 5		.	.				
.	.	See table 11, vol. I to correspond with C_{76} - C_{90}		.	.				
150	C_{150}								
151	δ_P	Orientation of airborne radar	(269)	C	C	deg	----	----	----
152	δ_Y		and figure 7	C	C	deg	----	----	----

[a]Only metric SI units are presented. For English units replace m by ft and kg by slugs.

[b]geod refers to geodetic latitude, geoc refers to geocentric latitude.

5. Sample input.- The card inputs for STEP1 and STEP2, which correspond to the sample output in Section II.C.5 follow.

a. STEP1 input:

```
1    STEP1 FILTERING CHECK PROBLEM (FEB. 1969)  WAGNER AND SEROLD
2    0   1   2   0    0   2    1   3    0    0                          PROGRAM CONTROLS
3 1    1  16001.2000  100.                                 (SIG=100.    ,RN=.55)
3 1    2 -6.75068000  .035                                 (SIG=2 DEG   ,RN=-.81)
3 1    3  63.0352000  .018                                 (SIG=1 DEG   ,RN=-.33)
3 1    4  100564.000 1000.                                 (SIG=1000.   ,RN=-.96)
3 1    5  16.7583977 .0003                                 (SIG=.017DEG,RN=1.69)
3 1    6  28.0418380 .0003                                 (SIG=.017DEG,RN=.05)
3 1    7  29.8270000 .0530                                 (SIG=3 DEG   ,RN=1.10)
3 1    8 -4.43200000 .0350                                 (SIG=2 DEG   ,RN=-1.72)
3 1    9  19.3930000 .0530                                 (SIG=3 DEG   ,RN=-1.37)
3 2  46  -.000108    .0002                                 (SIG=.0002   ,RN=-.54)
3 2 103 16.6055454   .000375                               (SIG=.0214 DEG) MOD.PARA. - Z
3 2 104 31.0         .000375                               (SIG=.0214 DEG) MOD.PARA. - Z
3 3  45  0.          .0001                                                 MOD.PARA. - U
3 3  47  0.          .0001                                                 MOD.PARA. - U
3 5  31 -1.0                                                               CONSTANT
3 5  33  0.5                                                               CONSTANT
3 5  88 15.09690377                                                        CONSTANT
3 5  89 27.0                                                               CONSTANT
4        70.0          5.0          0.5        90.                         NUMERICAL INTEG.
6 1   2                                                                    CG/IMU
 0.0          1.0          100.0        1.0
6 3   2                                                                    CG/IMU
 0.0         -0.5          100.0       -0.5
5 1       32.174048    1.407690E16 1092.055E-6                             GRAV. DATA
5 2       20855970.    20926430.    .7292115E-4                            GEOPH.DATA
7 1 70.0          82.0       .99        1 1 1  20.0      .00003     .00003    MEAS.CDS.
7 2 78.0          88.0      1.99        1 1 1  20.0      .00003     .00003    MEAS.CDS.
12      3        -1.        400.                                       MASS(T)
       -1.  25.57631            200.  24.8013            400.  24.0262
14 1                                                                   ATMOSPHERE
11-1              5.        15.                                        AERO TABLES
  1    3    8    5    8                                                CD(ALPHA,BETA,M)
   0.     4.     8.    12.    16.    20.    24.    28.
  -8.    -4.     0.     4.     8.
   2.     3.     4.     5.     6.    6.5     7.    30.
 .2247 .2266 .2404 .2602 .2886 .3261 .3730 .4297 .2220 .2265 .2388 .2589
 .2875 .3252 .3724 .4294 .2188 .2252 .2381 .2584 .2871 .3249 .3722 .4293
 .2220 .2265 .2388 .2589 .2875 .3252 .3724 .4294 .2247 .2286 .2404 .2602
 .2886 .3261 .3730 .4297 .2086 .2121 .2228 .2410 .2672 .3020 .3460 .3997
 .2004 .2048 .2170 .2364 .2634 .2989 .3434 .3976 .1915 .2006 .2146 .2347
 .2621 .2978 .3425 .3969 .2004 .2048 .2170 .2364 .2634 .2989 .3434 .3976
 .2086 .2121 .2228 .2410 .2672 .3020 .3460 .3997 .1660 .1700 .1817 .2006
 .2270 .2613 .3043 .3566 .1527 .1585 .1729 .1938 .2214 .2566 .3003 .3531
 .1387 .1522 .1692 .1911 .2193 .2549 .2988 .3519 .1527 .1585 .1729 .1938
 .2214 .2566 .3003 .3531 .1660 .1700 .1817 .2006 .2270 .2613 .3043 .3566
 .1226 .1275 .1410 .1618 .1896 .2250 .2685 .3210 .1058 .1135 .1306 .1537
 .1830 .2194 .2637 .3169 .0886 .1061 .1263 .1507 .1807 .2174 .2621 .3154
 .1058 .1135 .1306 .1537 .1830 .2194 .2637 .3169 .1226 .1275 .1410 .1618
 .1896 .2250 .2685 .3210 .0994 .1051 .1203 .1431 .1728 .2099 .2550 .3090
 .0812 .0901 .1095 .1347 .1660 .2041 .2501 .3048 .0625 .0826 .1051 .1317
 .1636 .2021 .2484 .3033 .0812 .0901 .1095 .1347 .1660 .2041 .2501 .3048
 .0994 .1051 .1203 .1431 .1728 .2099 .2550 .3090 .0969 .1026 .1179 .1408
 .1706 .2078 .2532 .3073 .0785 .0876 .1070 .1324 .1638 .2021 .2482 .3030
 .0597 .0799 .1027 .1293 .1614 .2001 .2465 .3016 .0785 .0876 .1070 .1324
 .1638 .2021 .2482 .3030 .0969 .1026 .1179 .1408 .1706 .2078 .2532 .3073
 .0965 .1023 .1176 .1405 .1703 .2075 .2529 .3070 .0782 .0873 .1067 .1321
 .1635 .2018 .2480 .3028 .0594 .0796 .1024 .1290 .1611 .1998 .2463 .3013
```

```
 .0782 .0873 .1067 .1321 .1635 .2018 .2480 .3028 .0965 .1023 .1176 .1405
 .1703 .2075 .2529 .3070 .0965 .1023 .1176 .1405 .1703 .2075 .2529 .3070
 .0782 .0873 .1067 .1321 .1635 .2018 .2480 .3028 .0594 .0796 .1024 .1290
 .1611 .1998 .2463 .3013 .0782 .0873 .1067 .1321 .1635 .2018 .2480 .3028
 .0965 .1023 .1176 .1405 .1703 .2075 .2529 .3070
   2   3   8   5   8                                       CY(ALPHA,BETA,M)
    0.    4.     8.    12.    16.    20.    24.    28.
   -8.   -4.     0.     4.     8.
    2.    3.     4.     5.     6.    6.5    7.    30.
 .1117 .1129 .1163 .1209 .1262 .1319 .1379 .1442 .0538 .0545 .0566 .0593
 .0622 .0652 .0684 .07160     0     0     0     0     0     0     0
-.0538-.0545-.0566-.0593-.0622-.0652-.0684-.0716-.1117-.1129-.1163-.1209
-.1262-.1319-.1379-.1442 .0575 .0623 .0722 .0826 .0924 .1017 .1108 .1197
 .0106 .0206 .0312 .0386 .0444 .0496 .0544 .05910     0     0     0
0     0     0     0     -.0106-.0206-.0312-.0386-.0444-.0496-.0544-.0591
-.0575-.0623-.0722-.0826-.0924-.1017-.1108-.1197 .0345 .0394 .0497 .0607
 .0713 .0814 .0913 .1011-.0011 .0089 .0198 .0275 .0337 .0393 .0446 .0497
0     0     0     0     0     0     0     0      .0011-.0089-.0198-.0275
-.0337-.0393-.0446-.0497-.0345-.0394-.0497-.0607-.0713-.0814-.0913-.1011
 .0354 .0386 .0461 .0548 .0639 .0733 .0827 .0923 .0077 .0128 .0194 .0250
 .0303 .0353 .0403 .04530     0     0     0     0     0     0     0
-.0077-.0128-.0194-.0250-.0303-.0353-.0403-.0453-.0354-.0386-.0461-.0548
-.0639-.0733-.0827-.0923 .0481 .0495 .0538 .0600 .0674 .0755 .0841 .0931
 .0228 .0228 .0248 .0283 .0323 .0366 .0411 .04570     0     0     0
0     0     0     0     -.0228-.0228-.0248-.0283-.0323-.0366-.0411-.0457
-.0481-.0495-.0538-.0600-.0674-.0755-.0841-.0931 .0484 .0499 .0540 .0602
 .0675 .0756 .0842 .0932 .0232 .0230 .0250 .0284 .0324 .0367 .0411 .0458
0     0     0     0     0     0     0     0     -.0232-.0230-.0250-.0284
-.0324-.0367-.0411-.0458-.0484-.0499-.0540-.0602-.0675-.0756-.0842-.0932
 .0484 .0499 .0540 .0602 .0675 .0756 .0842 .0932 .0232 .0230 .0250 .0284
 .0324 .0367 .0411 .04580     0     0     0     0     0     0     0
-.0232-.0230-.0250-.0284-.0324-.0367-.0411-.0458-.0484-.0499-.0540-.0602
-.0675-.0756-.0842-.0932 .0484 .0499 .0540 .0602 .0675 .0756 .0842 .0932
 .0232 .0230 .0250 .0284 .0324 .0367 .0411 .04580     0     0     0
0     0     0     0     -.0232-.0230-.0250-.0284-.0324-.0367-.0411-.0458
-.0484-.0499-.0540-.0602-.0675-.0756-.0842-.0932
   3   3   8   5   8                                       CL(ALPHA,BETA,M)
    0.    4.     8.    12.    16.    20.    24.    28.
   -8.   -4.     0.     4.     8.
    2.    3.     4.     5.     6.    6.5    7.    30.
0      .0402 .0825 .1275 .1747 .2230 .2709 .31700       .0387 .0802 .1251
 .1726 .2212 .2697 .3163 .0020 .0382 .0793 .1242 .1718 .2206 .2692 .3160
0      .0387 .0802 .1251 .1726 .2212 .2697 .31630       .0402 .0825 .1275
 .1747 .2230 .2709 .31700       .0162 .0409 .0737 .1119 .1536 .1969 .2402
0      .0063 .0323 .0670 .1067 .1495 .1937 .2379-.0288-.0034 .0279 .0643
 .1048 .1480 .1926 .23710       .0063 .0323 .0670 .1067 .1495 .1937 .2379
0      .0162 .0409 .0737 .1119 .1536 .1969 .24020       .0077 .0242 .0492
 .0803 .1158 .1538 .19290      -.0022 .0154 .0423 .0748 .1112 .1502 .1901
-.0282-.0118 .0110 .0395 .0727 .1096 .1489 .18910      -.0022 .0154 .0423
 .0748 .1112 .1502 .19010       .0077 .0242 .0492 .0803 .1158 .1538 .1929
0      .0103 .0262 .0485 .0761 .1078 .1424 .17840       .0049 .0206 .0434
 .0716 .1040 .1392 .1758-.0109 .0003 .0180 .0415 .0700 .1026 .1380 .1749
0      .0049 .0206 .0434 .0716 .1040 .1392 .17580       .0103 .0262 .0485
 .0761 .1078 .1424 .17840       .0173 .0369 .0603 .0879 .1191 .1528 .1879
0      .0165 .0345 .0573 .0847 .1161 .1502 .1857 .0067 .0171 .0338 .0562
 .0836 .1151 .1493 .18500       .0165 .0345 .0573 .0847 .1161 .1502 .1857
0      .0173 .0369 .0603 .0879 .1191 .1528 .18790       .0176 .0375 .0611
 .0888 .1202 .1541 .18920       .0169 .0352 .0581 .0857 .1172 .1515 .1871
 .0071 .0177 .0345 .0571 .0847 .1162 .1506 .18630       .0169 .0352 .0581
 .0857 .1172 .1515 .18710       .0176 .0375 .0611 .0888 .1202 .1541 .1892
0      .0177 .0375 .0611 .0889 .1203 .1542 .18940       .0169 .0352 .0582
 .0858 .1173 .1516 .1872 .0071 .0177 .0345 .0571 .0847 .1163 .1507 .1865
0      .0169 .0352 .0582 .0858 .1173 .1516 .18720       .0177 .0375 .0611
 .0889 .1203 .1542 .18940       .0177 .0375 .0611 .0889 .1203 .1542 .1894
0      .0169 .0352 .0582 .0858 .1173 .1516 .1872 .0071 .0177 .0345 .0571
 .0847 .1163 .1507 .18650       .0169 .0352 .0582 .0858 .1173 .1516 .1872
0      .0177 .0375 .0611 .0889 .1203 .1542 .1894
20                                                          END CARD
-1      END OF JOB
```

b. STEP2 input:

```
1     STEP2 FILTERING CHECK PROBLEM (FEB. 1969)   WAGNER AND SEROLD
2     0   1   2   0   0   2   1   3   0   0                     PROGRAM CONTROLS
3 1   1  16001.2000  100.                      (SIG=100.    ,RN=.55)
3 1   2 -6.75068000  .035                      (SIG=2 DEG   ,RN=-.81)
3 1   3  63.0352000  .018                      (SIG=1 DEG   ,RN=-.33)
3 1   4  100584.000 1000.                      (SIG=1000.   ,RN=-.96)
3 1   5  16.7583977 .0003                      (SIG=.017DEG,RN=1.69)
3 1   6  28.0418380 .0003                      (SIG=.017DEG,RN=.05)
3 1   7  29.8270000 .0530                      (SIG=3 DEG   ,RN=1.10)
3 1   8 -4.43200000 .0350                      (SIG=2 DEG   ,RN=-1.72)
3 1   9  19.3930000 .0530                      (SIG=3 DEG   ,RN=-1.37)
3 2  72  .092200000 .2000                      (SIG=.20     ,RN=-.54)
3 2 103 16.6055454  .000375                    (SIG=.0214 DEG) MOD.PARA. - Z
3 2 104 31.0        .000375                    (SIG=.0214 DEG) MOD.PARA. - Z
3 3  70  0.         .20                                        MOD.PARA. - U
3 3  71  0.         .20                                        MOD.PARA. - U
3 4  76  1.0        .000030                                    MEAS.PARA.- V
3 4  81  0.         .000030
3 5  31 -1.0                                                   CONSTANT
3 5  33  0.5                                                   CONSTANT
3 5  88 15.09690377                                            CONSTANT
3 5  89 27.0                                                   CONSTANT
6 1   2                                                        CG/IMU
 0.0          1.0          100.0        1.0
6 3   2                                                        CG/IMU
 0.0         -0.5          100.0       -0.5
4        70.0          5.0          0.5          90.0
5 1      32.174048     1.407690E16 1092.055E-6                 GRAV. DATA
5 2      20855970.     20926430.   .7292115E-4                 GEOPH.DATA
7 1 70.0       82.0       .99       1 1 1  20.0      .00003    .00003    MEAS.CDS.
7 2 78.0       88.0      1.99       1 1 1  20.0      .00003    .00003    MEAS.CDS.
20                                                             END CARD
-1       END OF JOB
```

B. Tape Input and Output

STEP always requires the PQR tape containing data used by the equations of motion. When operating in the filtering mode, the programs also require a FIT tape containing data to which the equations of motion are fit. Formats for these tapes are presented in the previous section under input format 2 and 3.

On filtering problems, optional capability has been included to write a tape containing the 10-state variables u, v, w, h, φ, θ, e_0, e_1, e_2, and e_3 versus time. This tape, called STATE, is written on the backward smoothing of the last iteration so that it contains the best estimate of the variables after processing all data. Because it is written on the backward smoothing, the first points on the tape are at final time, the last points at initial time. The tape can be used to print the state time history in tabular form, or to prepare STEP2 position and velocity estimates for processing on STEP1. All angular data on the tape are in radians, the remainder in the m, kg, sec or slug, ft, sec systems of units.

FORTRAN name	Logical tape unit	Description
IN	5	Input tape
OUT	6	Output tape
PQR	2	PQR tape
FIT	1	FIT tape
STATE	4	STATE tape
SCRACH	3	Scratch tape used for temporary storage during problem execution

C. Tab Output

The tab printout from STEP is divided into four parts -- Part 1, the printout of input variables and controls; Part 2, the block printout during the forward and backward integration; Part 3, the printout of the measurement schedule; and Part 4, the printout of the residuals and loss function. Each of these printouts will be described. The sample outputs in Subsection C.5 will aid in the discussion.

1. Printout of input data and controls.- As data are input to STEP, they are immediately printed for later verification. This printout, shown on the first page of the sample output, is unlabled and exactly duplicates the data on the input cards. If the user requests printout of the PQR and/or FIT tapes, it occurs during this part of the output. After all input data have been loaded from cards, subroutine SETUP is called to transform and convert units (e.g., degrees to radians) on the data to prepare for the problem execution. From SETUP, a summary of the problem operating mode, options, and controls is printed in the form of phrases and data. The data consist primarily of geophysical, gravitational, and atmosphere constants, state variables, and tracking station locations. The units on these data are consistent with metric or English system specified. All angular data are in degrees.

The components of the expanded state variables transformed to internal units are next tabulated along with uncertain model parameters and their variances. In this tabulation, all angles are in radians.

2. Block printout.- During execution of a problem, labeled block printouts occurs during both the forward filtering and backward smoothing at a frequency specified by input. The variables in the block printout on the forward integration correspond to the reference trajectory. In STEP2, this block printout appears as follows:

```
TIME --

V(A) -- G(A) -- L(A) --  ALT --  LATD --  LON --

U(I) -- V(I) -- W(I) --  PSI --   THE --  PHI --

U(B) -- V(B) -- W(B) --  SIG --   BET --  ALF --

  XP --   YP --   ZP -- V(T) --   XZI --  ETA --

   P --    Q --    R --   AX --    AY --   AZ --

  PM --   QM --   RM --  AXM --   AYM --  AZM --

  EO --   E1 --   E2 --   E3 --  LATC -- A(T) --
```

where

V(A), G(A), L(A) = Atmosphere relative velocity vector V_A, γ_A, and λ_A. In STEP2, there is no atmosphere; therefore, the velocity vector is relative to planet surface. Units are m/sec or ft/sec and deg.

U(I), V(I), W(I) = Inertial velocity vector Cartesian components, u, v, and w. Units are m/sec or ft/sec.

U(B), V(B), W(B) = Components of V_A in body axes directions, u_B, v_B, and w_B. Units are m/sec or ft/sec.

ALT = Altitude above oblate planet surface, h_O. Units are in m or ft.

LATD = Geodetic latitude in degrees.

LON = Longitude in degrees.

PSI, THE, PHI = Euler angles $\bar{\psi}$, $\bar{\theta}$, $\bar{\varphi}$ and in degrees.

SIG, BET, ALF = Roll angle σ about the velocity vector V_A, sideslip angle β and angle of attack α. Units are degrees.

XP, YP, ZP = Distances from center of gravity to the IMU as measured along body axes directions, and corrected for systemmatic error, x_p, y_p, and z_p. Units are in m or ft.

V(T) = Magnitude of the inertial velocity vector in m/sec or ft/sec.

XZI = Steering angle ξ in degrees. See figure 4 of vol. I.

ETA = Steering angle η in degrees. See figure 4 of vol. I.

P, Q, R = Inertial angular rates about the roll, pitch, and yaw axes and corrected for systemmatic error. Units are in rad/sec.

PM, QM, RM = Inertial angular rates about the roll, pitch, and yaw axes as interpolated from the PQR tape. Units are in rad/sec.

AX, AY, AZ = Accelerations a_{XB}, a_{YB}, and a_{ZB}, along the body axes direction and corrected for systematic error. Units are in m/sec^2 or ft/sec^2.

AXM, AYM, AZM = Accelerations, a_{XM}, a_{YM}, and a_{ZM}, along the body axes directions as interpolated from the PQR tape. Units are in m/sec^2 or ft/sec^2.

EO, E1, E2, E3 = Euler parameters e_0, e_1, e_2, and e_3.

LATC = Geocentric latitude in degrees.

A(T) = Total acceleration in m/sec^2 or ft/sec^2.

In STEP1, the last two lines of output are replaced by the following:

PM -- QM -- RM -- C_A -- C_Y -- C_N --

DP -- M -- RHO -- C1 -- C2 -- C3 --

EO -- E1 -- E2 -- E3 -- LATC -- RE --

UW -- VW --

where

C_A, C_Y, C_N = Axial, lateral, and normal aerodynamic coefficients.

C1, C2, C3 = Aerodynamic coefficients interpolated from input tables, either C_A, C_Y, C_N, or C_D, C_Y, C_L, or C_A, C_{Y_η}, C_{N_η}, respectively.

DP = Dynamic pressure q in nt/m^2 or lb_f/ft^2.

M = Mach number, M.

RHO = Atmospheric density ρ in kg/m^3 or $slugs/ft^3$.

RE = Reynolds number, R_e.

UW, VW = Atmospheric winds corrected for systemmatic error, u_W and v_W. Units are in m/sec or ft/sec.

On filtering problems during the forward integration the reference values of the model parameters being estimated are printed in the order that they were input. The print format is as follows:

C(24) -- C(28) -- C(43) -- C(40) -- C(77) -- C(88) --

C(.) -- . . .

The argument of C(.) are the variable identification numbers and their units are as described in Subsection II.A.4.

On nonupdated nominal filtering problems during the forward integration, the accumulated minimum variance corrections, $\hat{z}(t/t)$, to be added to the reference, $Z_{ref}(t)$, are next printed:

Z(1) -- Z(2) -- Z(3) -- Z(4) -- Z(5) -- Z(6) --

Z(7) -- Z(8) -- Z(9) -- Z(24) -- Z(28) -- Z(43) --

Z(88) -- . . .

The first 9 or 10 variables correspond to the transformed state variables specified by NPC(3). The remaining variables correspond to the estimated model parameters whose identification numbers and units are specified in Subsection II.A.4.

The standard deviations of the expanded state vector components are printed next on error analysis and filtering problems according to the following format:

S(1) -- S(2) -- S(3) -- S(4) -- S(5) -- S(6) --

S(7) -- S(8) -- S(9) -- S(23) -- S(19) --.

The first 9 or 10 standard deviations correspond to the basic state variables, transformed as specified by NPC(3). These are followed by the standard deviation of model parameters being estimated in the order that they were input. The argument in S( ) is the identification number of the estimated model parameter which, along with the units for the standard deviations, are shown in Subsection II.A.4. These standard deviations are printed on the backward smoothing on the last iteration only if NPC(8) equals 2 or 3.

On error analysis or filter problems, the entire covariance and correlation matrices P, CUZ, and CVZ, can be printed during the last iteration if NPC(6) is properly specified. These matrices correspond to the output units designated by NPC(3). Units for the covariance and correlation elements are as described for the standard deviations in Subsection II.A.4. Because the covariance matrix P is symmetrical, only its lower left triangle is printed as follows:

```
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
P(1,1)
P(2,1) P(2,2)
P(3,1) P(3,2) P(3,3)
.      .      .      .
.      .      .      .      .
.      .      .      .      .      .
```

The correlation matrices CUZ and CVZ are printed as shown below.

```
CORRELATION MATRIX (CVZ) TRANSPOSE
CUZ(1,1) CUZ(2,1) . . . . . . . . . . . CUZ(12,1)
CUZ(13,1) CUZ(14,1) . . .
CUZ(1,2) CUZ(2,2) . . . . . . . . . . . CUZ(12,2)
CUZ(13,2) CUZ(14,2) . . .
  .
  .
CUZ(1,q) CUZ(2,q) . . . . . . . . . . CUZ(12,q)
CUZ(13,q) CUZ(14,q) . . .
CORRELATION MATRIX (CVZ) TRANSPOSE
CVZ(1,1) CVZ(2,1) . . . . . . . . . .   CVZ(12,1)
CVZ(13,1) CVZ(14,1) . . .
```

CVZ(1,2) CVZ(2,2) . . . . CVZ(12,2)

CVZ(13,2) CVZ(14,2) . . .

.

.

CVZ(1,r) CVZ(2,r) . . . . CVZ(12,r)

CVZ(13,4) CVZ(14,4) . . .

3. Measurement schedule.- Between block printing on the forward integration, a schedule of the measurements processed can be printed if NPC(5) equals 0. The following format is used for this printout:

POINTS	TYPE	TIME	COMP1	COMP2	COMP3	SIG1	SIG2	SIG3
x, x, x	x	xxx	xxx	xxx	xxx	xxx	xxx	xxx
x, x, x	x	xxx	xxx	xxx	xxx	xxx	xxx	xxx

The number of scalar data points processed is counted and printed under the heading POINTS. The three numbers correspond to the three scalar components accounted for on each line of data. If a zero is printed under the POINTS heading, then the component corresponding to the zero was not processed. Under TYPE, the type of data (MTYP) is printed. Definitions of MTYP are presented in Subsection II.A.2. The time of the data is next printed under TIME and followed by the values for the three components. Units for these data are shown below.

. . .	TYP	TIME	COMP1	COMP2	COMP3	SIG1	SIG2	SIG3
Tracking	1-5	(sec)	R(m, ft)	A(rad)	E(rad)	σ_R(m, ft)	σ_A(rad)	σ_E(rad)
Airborne radar	6	(sec)	R_R(m, ft)	unused	unused	σ_{R_R} (m, ft)	unused	unused
Velocity	7	(sec)	u(m/sec ft/sec)	v(m/sec ft/sec)	w(m/sec ft/sec)	σ_u(m/sec ft/sec)	σ_v(m/sec ft/sec)	σ_w(m/sec ft/sec)
Position	8	(sec)	h(m, ft)	φ(rad)	θ(rad)	σ_h(m, ft)	σ_φ(rad)	σ_θ(rad)
Accelerations	9	(sec)	a_x(m/sec^2 ft/sec^2)	a_y(m/sec^2 ft/sec^2)	a_z(m/sec^2 ft/sec^2)	σ_{a_x} (m/sec^2 ft/sec^2)	σ_{a_y} (m/sec^2 ft/sec^2)	σ_{a_z} (m/sec^2 ft/sec^2)

The values for the measurement that are printed under columns COMP1. . COMP3 are the values corresponding to the reference trajectory and not the actual data from the FIT tape. The standard deviations under columns SIG1. . SIG3 are the sum of the standard deviations from the FIT tape and those input in category 7. These are the standard deviations used in the filter equations.

4. Residuals and loss function.- During the backward smoothing on the last iteration, best estimates of the measurements are calculated. The value of the measurement from the FIT tape is then subtracted from that calculated to form a residual. The residual is then divided by the standard deviation to form a weighted residual, and its square accumulated with all previously calculated weighted residuals to form the sum of the weighted residuals squared or loss function. These are tabulated between block prints as follows:

POINT	TYPE	TIME	RES1	RES2	RES3	WGT. RES1	WGT. RES2	WGT. RES3	LOSS FCTN
x, x, x	x	xx	xx	xx	xx	xx	xx	xx	xx
x, x, x	x	xx	xx	xx	xx	xx	xx	xx	xx

The quantities under POINT, TYPE, and TIME are the same as in the measurement schedule in Subsection 3 above. RES1 .. RES3 are the unweighted residuals in m, ft, and rad; WGT. RES1 ... WGT. RES3 are the weighted residuals; and LOSS FCTN is the loss function. Three components of data per line is maintained. Only those components processed on the forward integration are accumulated in the loss function.

5. Sample output.- The sample outputs presented below were obtained on STEP1 and STEP2. They correspond to 20 sec of the check problem between 70 and 90 sec.

a. STEP1 output:

```
DATA ANAL●SIS PROGRAM

INPUT DATA

    1        STEP1 FILTERING CHECK PROBLEM (FEB. 1969)  WAGNER AND SEROLD
    2        0  1  2  0  0  2  1  3  0  0                                  PROGRAM CO     NTROLS
    3     1  1  16001.2000  100.                           (SIG=100.    ,RN=.55)
    3     1  2 -6.75088000  .035                           (SIG=2 DEG   ,RN=-.81)
    3     1  3  63.0352000  .018                           (SIG=1 DEG   ,RN=-.33)
    3     1  4  100584.000 1000.                           (SIG=1000.   ,RN=-.96)
    3     1  5  16.7583977 .0003                           (SIG=.017DEG,RN=1.69)
    3     1  6  28.0418380 .0003                           (SIG=.017DEG,RN=.05)
    3     1  7  29.8270000 .0530                           (SIG=3 DEG   ,RN=1.10)
    3     1  8 -4.43200000 .0350                           (SIG=2 DEG   ,RN=-1.72)
    3     1  9  19.3930000 .0530                           (SIG=3 DEG   ,RN=-1.37)
    3     2  46  -.000108   .0002                          (SIG=.0002   ,RN=-.54)
    3     2 103 16.6055454  .000375                        (SIG=.0214 DEG) MOD.PARA.     - Z
    3     2 104 31.0        .000375                        (SIG=.0214 DEG) MOD.PARA.     - Z
    3     3  45  0.         .0001                                          MOD.PARA.     - U
    3     3  47  0.         .0001                                          MOD.PARA.     - U
    3     5  31 -1.0                                                       CONSTANT
    3     5  33  0.5                                                       CONSTANT
    3     5  88 15.09690377                                                CONSTANT
    3     5  89 27.0                                                       CONSTANT
    ●          70.0        5.0          0.5        90.                     NUMERICAL     INTFG.
    6     1  2                                                             CG/IMU
 0.       1.0000E+00  1.0000E+02  1.0000E+00
    6     3  2                                                             CG/IMU
 0.       -5.0000E-01  1.0000E+02 -5.0000E-01
    5     1       32.174048   1.407690E16 1092.055E-6                      GRAV. DATA
    5     2       20855970.   20926430.   .7292115E-4                      GEOPH.DATA
    7     1 70.0       82.0       .99       1 1 1  20.0     .00003    .00003   M    EAS.CDS.
    7     2 78.0       88.0       1.99      1 1 1  20.0     .00003    .00003   M    EAS.CDS.
   12        2          -1.         400.                                   MASS(T)
   -1.00  2.55763E+01    200.00   2.●8013E+01     400.00   2.40262E+01
   14     1                                                                ATMOSPHERE
   11    -1              5.         15.                                    AERO TABLE    S
 1  3  8  5  8
 0.        4.000000  8.000000 12.000000 16.000000 20.000000 24.000000 28.000000
-8.000000 -4.000000  0.        4.000000  8.000000
 2.000000  3.000000  4.000000  5.000000  6.000000  6.500000  7.000000 30.000000
  .224700   .228600   .240400   .260200   .288600   .326100   .373000   .429700   .222000   .226500   .238800   .258900
  .287500   .325200   .372400   .429400   .218800   .225200   .238100   .258400   .287100   .324900   .372200   .429300
  .222000   .226500   .238800   .258900   .287500   .325200   .372400   .429400   .224700   .228600   .240400   .260200
  .288600   .326100   .373000   .429700   .208600   .212100   .222800   .241000   .267200   .302000   .346000   .399700
  .200400   .204800   .217000   .236400   .263400   .298900   .343400   .397600   .191500   .200600   .214600   .234700
  .262100   .297800   .342500   .396900   .200400   .204800   .217000   .236400   .263400   .298900   .343400   .397600
  .208600   .212100   .222800   .241000   .267200   .302000   .346000   .399700   .166000   .170000   .181700   .200600
  .227000   .261300   .304300   .356600   .152700   .158500   .172900   .193800   .221400   .256600   .300300   .353100
  .138700   .152200   .169200   .191100   .219300   .254900   .298800   .351900   .152700   .158500   .172900   .193800
  .221400   .256600   .300300   .353100   .166000   .170000   .181700   .200600   .227000   .261300   .304300   .356600
  .122600   .127500   .141000   .161800   .189600   .225000   .268500   .321000   .105800   .113500   .130600   .153700
  .183000   .219400   .263700   .316900   .088600   .106100   .126300   .150700   .180700   .217400   .262100   .315400
  .105800   .113500   .130600   .153700   .183000   .219400   .263700   .316900   .122600   .127500   .141000   .161800
  .189600   .225000   .268500   .321000   .099400   .105100   .120300   .143100   .172800   .209900   .255000   .309000
  .081200   .090100   .109500   .134700   .166000   .204100   .250100   .304800   .062500   .082600   .105100   .131700
  .163600   .202100   .248400   .303300   .081200   .090100   .109500   .134700   .166000   .204100   .250100   .304800
  .099400   .105100   .120300   .143100   .172800   .209900   .255000   .309000   .096900   .102600   .117900   .140800
  .170600   .207800   .253200   .307300   .078500   .087600   .107000   .132400   .163800   .202100   .248200   .303000
  .059700   .079900   .102700   .129300   .161400   .200100   .246500   .301600   .078500   .087600   .107000   .132400
  .163800   .202100   .248200   .303000   .096900   .102600   .117900   .140800   .170600   .207800   .253200   .307300
  .096500   .102300   .117600   .140500   .170300   .207500   .252900   .307000   .078200   .087300   .106700   .132100
  .163500   .201800   .248000   .302800   .059400   .079600   .102400   .129000   .161100   .199800   .246300   .301300
  .078200   .087300   .106700   .132100   .163500   .201800   .248000   .302800   .096500   .102300   .117600   .140500
  .170300   .207500   .252900   .307000   .096500   .102300   .117600   .140500   .170300   .207500   .252900   .307000
  .078200   .087300   .106700   .132100   .163500   .201800   .248000   .302800   .059400   .079600   .102400   .129000
  .161100   .199800   .246300   .301300   .078200   .087300   .106700   .132100   .163500   .201800   .248000   .302800
  .096500   .102300   .117600   .140500   .170300   .207500   .252900   .307000
```

2	3	8	5	8							
0.	4.000000	8.000000	12.000000	16.000000	20.000000	24.000000	28.000000				
-8.000000	-4.000000	0.	4.000000	8.000000							
2.000000	3.000000	4.000000	5.000000	6.000000	6.500000	7.000000	30.000000				
.111700	.112900	.116300	.120900	.126200	.131900	.137900	.144200	.053800	.054500	.056600	.059300
.062200	.065200	.068400	.071600	0.	0.	0.	0.	0.	0.	0.	0.
-.053800	-.054500	-.056600	-.059300	-.062200	-.065200	-.068400	-.071600	-.111700	-.112900	-.116300	-.120900
-.126200	-.131900	-.137900	-.144200	.057500	.062300	.072200	.082600	.092400	.101700	.110800	.119700
.010600	.020600	.031200	.038600	.044400	.049600	.054400	.059100	0.	0.	0.	0.
0.	0.	0.	0.	-.010600	-.020600	-.031200	-.038600	-.044400	-.049600	-.054400	-.059100
-.057500	-.062300	-.072200	-.082600	-.092400	-.101700	-.110800	-.119700	.034500	.039400	.049700	.060700
.071300	.081400	.091300	.101100	-.001100	.008900	.019800	.027500	.033700	.039300	.044600	.049700
0.	0.	0.	0.	0.	0.	0.	0.	.001100	-.008900	-.019800	-.027500
-.033700	-.039300	-.044600	-.049700	-.034500	-.039400	-.049700	-.060700	-.071300	-.081400	-.091300	-.101100
.035400	.038600	.046100	.054800	.063900	.073300	.082700	.092300	.007700	.012800	.019400	.025000
.030300	.035300	.040300	.045300	0.	0.	0.	0.	0.	0.	0.	0.
-.007700	-.012800	-.019400	-.025000	-.030300	-.035300	-.040300	-.045300	-.035400	-.038600	-.046100	-.054800
-.063900	-.073300	-.082700	-.092300	.048100	.049500	.053800	.060000	.067400	.075500	.084100	.093100
[illegible]	[illegible]	.024800	.028300	.032300	.036600	.041100	.045700	0.	0.	0.	0.
0.	0.	0.	0.	-.022800	-.022800	-.024800	-.028300	-.032300	-.036600	-.041100	-.045700
-.048100	-.049500	-.053800	-.060000	-.067400	-.075500	-.084100	-.093100	.048400	.049900	.054000	.060200
.067500	.075600	.084200	.093200	.023200	.023000	.025000	.028400	.032400	.036700	.041100	.045800
0.	0.	0.	0.	0.	0.	0.	0.	-.023200	-.023000	-.025000	-.028400
-.032400	-.036700	-.041100	-.045800	-.048400	-.049900	-.054000	-.060200	-.067500	-.075600	-.084200	-.093200
.048400	.049900	.054000	.060200	.067500	.075600	.084200	.093200	.023200	.023000	.025000	.028400
.032400	.036700	.041100	.045800	0.	0.	0.	0.	0.	0.	0.	0.
-.023200	-.023000	-.025000	-.028400	-.032400	-.036700	-.041100	-.045800	-.048400	-.049900	-.054000	-.060200
-.067500	-.075600	-.084200	-.093200	.048400	.049900	.054000	.060200	.067500	.075600	.084200	.093200
.023200	.023000	.025000	.028400	.032400	.036700	.041100	.045800	0.	0.	0.	0.
0.	0.	0.	0.	-.023200	-.023000	-.025000	-.028400	-.032400	-.036700	-.041100	-.045800
-.048400	-.049900	-.054000	-.060200	-.067500	-.075600	-.084200	-.093200				
3	3	8	5	8							
0.	4.000000	8.000000	12.000000	16.000000	20.000000	24.000000	28.000000				
-8.000000	-4.000000	0.	4.000000	8.000000							
2.000000	3.000000	4.000000	5.000000	6.000000	6.500000	7.000000	30.000000				
0.	.040200	.082500	.127500	.174700	.223000	.270900	.317000	0.	.038700	.080200	.125100
.172600	.221200	.269700	.316300	.002000	.038200	.079300	.124200	.171800	.220600	.269200	.316000
0.	.038700	.080200	.125100	.172600	.221200	.269700	.316300	0.	.040200	.082500	.127500
.174700	.223000	.270900	.317000	0.	.016200	.040900	.073700	.111900	.153600	.196900	.240200
0.	.006300	.032300	.067000	.106700	.149500	.193700	.237900	-.028800	-.003400	.027900	.064300
.104800	.148000	.192600	.237100	0.	.006300	.032300	.067000	.106700	.149500	.193700	.237900
0.	.016200	.040900	.073700	.111900	.153600	.196900	.240200	0.	.007700	.024200	.049200
.080300	.115800	.153800	.192900	0.	-.002200	.015400	.042300	.074800	.111200	.150200	.190100
-.028200	-.011800	.011000	.039500	.072700	.109600	.148900	.189100	0.	-.002200	.015400	.042300
.074800	.111200	.150200	.190100	0.	.007700	.024200	.049200	.080300	.115800	.153800	.192900
0.	.010300	.026200	.048500	.076100	.107800	.142400	.178400	0.	.004900	.020600	.043400
[illegible]	.104000	.139200	.175800	-.010900	.000300	.018000	.041500	.070000	.102600	.138000	.174900
0.	.004900	.020600	.043400	.071600	.104000	.139200	.175800	0.	.010300	.026200	.048500
.076100	.107800	.142400	.178400	0.	.017300	.036900	.060300	.087900	.119100	.152800	.187900
0.	.016500	.034500	.057300	.084700	.116100	.150200	.185700	.006700	.017100	.033800	.056200
.083600	.115100	.149300	.185000	0.	.016500	.034500	.057300	.084700	.116100	.150200	.185700
0.	.017300	.036900	.060300	.087900	.119100	.152800	.187900	0.	.017600	.037500	.061100
.088800	.120200	.154100	.189200	0.	.016900	.035200	.058100	.085700	.117200	.151500	.187100
.007100	.017700	.034500	.057100	.084700	.116200	.150600	.186300	0.	.016900	.035200	.058100
.085700	.117200	.151500	.187100	0.	.017600	.037500	.061100	.088800	.120200	.154100	.189200
0.	.017700	.037500	.061100	.088900	.120300	.154200	.189400	0.	.016900	.035200	.058200
.085800	.117300	.151600	.187200	.007100	.017700	.034500	.057100	.084700	.116300	.150700	.186500
0.	.016900	.035200	.058200	.085800	.117300	.151600	.187200	0.	.017700	.037500	.061100
.088900	.120300	.154200	.189400	0.	.017700	.037500	.061100	.088900	.120300	.154200	.18940
0.	.016900	.035200	.058200	.085800	.117300	.151600	.187200	.007100	.017700	.034500	.0571
.084700	.116300	.150700	.186500	0.	.016900	.035200	.058200	.085800	.117300	.151600	.1872
0.	.017700	.037500	.061100	.088900	.120300	.154200	.189400				
20							END CARD				

```
CONTROLS ARE SPECIFIED FOR
   *FILTERING RUN FROM TIME TO=  70.0000 TO TFINAL=  90.0000
   *UPDATED REFERENCE
   *SMOOTH COVARIANCE,CALCULATE RESIDUALS AND LOSS FUNCTION
   *FILTER FOR  1 ITERATIONS
   *VECTOR PROCESS FITTING DATA TRIPLES
   *INPUT AND OUTPUT IN ENGLISH UNITS (FT,FT/SEC,FT/SEC2,LB,SLUG)
   *PLANET PARAMETERS
                 EQUIT.RAD.= 2.09264300E+07  POLAR RAD.= 2.08559700E+07  ROTATION  = 7.29211500E-05
                 MU(GRAV.) = 1.40769000E+16  J2(GRAV. ) = 1.09205500E-03      G(0)  = 3.21740480E+01
   *ATMOSPHERE IS 1959 ARDC
                     MO = 2.89644000E+01    R(STAR) = 4.97192198E+04      CP/CV = 1.40000000E+00
                   P(0) = 2.11621667E+03       BETA = 2.26968110E-08          S = 1.98720000E+02
     HB =     0.              3.60892388E+04   8.20209974E+04   1.54199475E+05   1.73884514E+05   2.59186352E+05
              2.95275591E+05  3.44488189E+05   5.24934383E+05   5.57742782E+05   6.56167979E+05   2.29658793E+06
     TMB =    5.18688000E+02  3.89988000E+02   3.89988000E+02   5.08788000E+02   5.08788000E+02   2.98188000E+02
              2.98188000E+02  4.06188000E+02   2.38618800E+03   2.56618800E+03   2.83618800E+03   5.98618800E+03
     PB =     2.11621667E+03  4.72708961E+02   5.19844194E+01   2.51625139E+00   1.21846395E+00   2.10938433E-02
              2.18258255E-03  1.55768677E-04   7.56734675E-06   5.90259850E-06   2.97981080E-06   2.03048103E-09
   *INITIAL STATE ESTIMATE IS
                 VELOCITY  = 1.60012000E+04  GAMMA     =-6.75088000E+00  LAMBDA    = 6.30352000E+01
                 ALTITUDE  = 1.00584000E+05  GEOD.LAT. = 1.67583977E+01  LONGITUDE = 2.80418380E+01
                 SIGMA     = 2.98270000E+01  BETA      =-4.43200000E+00  ALPHA     = 1.93930000E+01
   *FITTING DATA IS FROM  2 SOURCES
       1. TRACKING STATION 1, GEOD.LAT.=  15.096904  LONGITUDE=  27.000000  ALTITUDE =     0.     ABOVE REF. SURFACE
       2. TRACKING STATION 2, GEOD.LAT.=  16.605545  LONGITUDE=  31.000000  ALTITUDE =     0.     ABOVE REF. SURFACE


STATE VECTOR COMPONENTS
COMP.   ID.NO.     EST.VALUE      VARIANCE
  1       1      7.20532663E+03 6.79079215E+04
  2       2      1.56313582E+04 2.80981835E+04
  3       3      1.88098346E+03 3.09451057E+05
  4       4      9.47712173E+04 1.00000000E+06
  5       5      2.90632098E-01 9.00000000E-08
  6       6      4.89422401E-01 9.00000000E-08
  7       7      7.69645202E-01 2.08169635E-04
  8       8      1.55940474E-01 7.90725687E-04
  9       9      2.08732405E-01 8.18646927E-04
 10      10      5.82889025E-01 2.80457751E-04
MODEL PARAMETERS IN EXPANDED STATE VECTOR
 11      46     -1.08000000E-04 4.00000000E-08
 12     103      2.87979327E-01 1.40625000E-07
 13     104      5.41052068E-01 1.40625000E-07
RANDOM MODEL PARAMETERS NOT BEING ESTIMATED (U-VECTOR)
  1      45      0.             1.00000000E-08
  2      47      0.             1.00000000E-08
```

ITERATION NUMBER 1

```
TIME 7.00000000E+01
V(R) 1.60012000E+04 G(R)-6.75088000E+00 L(R) 6.30352000E+01 ALT  1.00584000E+05 LAT  1.67583977E+01 LON  2.80418380E+01
U(I) 7.20532663E+03 V(I) 1.56313582E+04 W(I) 1.88098346E+03 PSI  7.63702516E+01 THE  8.01936612E+00 PHI  2.92188575E+01
U(B) 1.50482107E+04 V(B)-1.23650674E+03 W(B) 5.29724514E+03 SIG  2.98270000E+01 BET -4.43200000E+00 ALF  1.93930000E+01
XP   0.             YP   0.             ZP   0.             V(T) 5.43964658E+03 XZI -1.31389754E+01 ETA  1.87755893E+01
P   -8.75453186E-04 Q    5.93691552E-03 R    8.82155972E-04 AX  -1.16901583E+02 AY   3.17431021E+01 AZ  -1.35649291E+02
PM  -8.75453186E-04 QM   6.04491552E-03 RM   8.82155972E-04 CX  -1.47996039E-01 CY   4.01863968E-02 CZ  -1.71730420E-01
DP   3.99728017E+03 M    1.59317207E+01 RHO  3.12240675E-05 C1   1.96621603E-01 C2   4.01863968E-02 C3   1.12845514E-01
E0   7.69645202E-01 E1   1.55940474E-01 E2   2.08732405E-01 E3   5.82889025E-01 LATC 1.66519926E+01 RE   2.37465225E+07
UW   0.             VW   0.
C 46-1.08000000E-04 C103 2.87979327E-01 C104 5.41052068E-01 C
S(  1) 1.000000E+02 S(  2) 3.500000E-02 S(  3) 1.800000E-02 S(  4) 1.000000E+03 S(  5) 3.000000E-04 S(  6) 3.000000E-04
S(  7) 5.300000E-02 S(  8) 3.500000E-02 S(  9) 5.300000E-02 S( 46) 0.             S(103) 0.             S(104) 3.750000E-04
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
 1.000E+04
-4.796E-14 1.225E-03
-1.954E-14 2.074E-26 3.240E-04
-2.274E-13 0.        1.388E-17 1.000E+06
 2.168E-19 1.654E-24 6.617E-24 0.        9.000E-08
 0.        0.        0.        0.        0.        9.000E-08
 8.281E-15 0.        0.        0.       -8.272E-25 0.        2.809E-03
 7.317E-15-1.041E-17 0.        0.        0.        0.        3.339E-17 1.225E-03
 7.855E-15-1.843E-17-4.337E-19 0.        3.309E-24 0.        3.567E-17-5.204E-18 2.809E-03
 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        1.406E-07
CORRELATION MATRIX  (CUZ)TRANSPOSE
 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.

    POINTS      TYPE     TIME        COMP1          COMP2          COMP3          SIG1           SIG2           SIG3
   1,   2,   3   1     70.0000     714184.53        .543024        .124359       20.000000        .000030        .000030
   4,   5,   6   1     71.0000     720631.04        .560080        .121736       20.000000        .000030        .000030
   7,   8,   9   1     72.0000     733673.53        .571235        .117737       20.000000        .000030        .000030
  10,  11,  12   1     73.0000     746633.86        .581933        .113474       20.000000        .000030        .000030
  13,  14,  15   1     74.0000     759495.28        .592182        .109513       20.000000        .000030        .000030
  16,  17,  18   1     75.0000     772261.98        .602009        .105841       20.000000        .000030        .000030


TIME 7.50000000E+01
V(R) 1.49660941E+04 G(R)-3.60307847E+00 L(R) 6.45375595E+01 ALT  9.57092856E+04 LAT  1.68297030E+01 LON  2.82380330E+01
U(I) 6.42149491E+03 V(I) 1.49534045E+04 W(I) 9.40531351E+02 PSI  7.71482732E+01 THE  1.62581832E+01 PHI  2.77237382E+01
U(B) 1.37299621E+04 V(B)-5.68174499E+02 W(B) 5.92868369E+03 SIG  2.67560182E+01 BET -2.17570634E+00 ALF  2.33549896E+01
XP   0.             YP   0.             ZP   0.             V(T) 5.95584693E+03 XZI -5.47421418E+00 ETA  2.17003793E+01
P   -9.42614978E-04 Q    7.62858852E-03 R    9.82154065E-04 AX  -1.42289431E+02 AY   1.92525940E+01 AZ  -2.00471913E+02
PM  -9.42614978E-04 QM   7.73459373E-03 RM   9.82154065E-04 CX  -1.62368876E-01 CY   2.19694605E-02 CZ  -2.28761889E-01
DP   4.43132009E+03 M    1.50442366E+01 RHO  3.95681893E-05 C1   2.39752742E-01 C2   2.19694605E-02 C3   1.45651215E-01
E0   7.72529879E-01 E1   9.98223304E-02 E2   2.55214946E-01 E3   5.72798761E-01 LATC 1.67228979E+01 RE   2.85923201E+07
UW   0.             VW   0.
C 46-1.06005212E-04 C103 2.87979327E-01 C104 5.41052068E-01 C
S(  1) 1.663637E+01 S(  2) 7.496498E-04 S(  3) 9.200352E-04 S(  4) 1.794085E+01 S(  5) 8.503153E-07 S(  6) 9.186861E-07
S(  7) 4.771434E-02 S(  8) 2.262646E-02 S(  9) 1.034877E-02 S( 46) 0.             S(103) 0.             S(104) 3.750000E-04
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
```

```
 2.768E+02
-5.954E-03 5.620E-07
-8.856E-04-4.365E-07 8.465E-07
-8.104E+01 9.490E-03-4.393E-03 3.219E+02
 4.808E-06 1.160E-10-5.272E-10 6.334E-07 7.230E-13
 1.000E-05-2.445E-10 2.230E-10-3.270E-06-1.626E-14 8.440E-13
-2.163E-02 0.         0.        -1.325E-01-4.733E-09 1.759E-09 2.277E-03
-5.133E-02 7.556E-06 0.         7.933E-02 5.634E-09-3.897E-09 5.180E-04 5.120E-04
-1.583E-01 4.664E-06-1.277E-06 5.810E-02-1.496E-09-4.909E-09 7.118E-05 9.860E-05 1.071E-04
 1.254E-09 6.114E-10 6.157E-06 6.157E-06-4.268E-13-7.340E-13-4.509E-10 1.302E-09 9.956E-08 0.
 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.         1.406E-07
CORRELATION MATRIX (CUZ)TRANSPOSE
-9.179E-07-6.451E-11 8.775E-12-5.485E-07-4.716E-16-1.870E-14 4.622E-08 1.743E-08 2.666E-09-4.422E-14 0.         0.
-1.108E-06-3.918E-10 7.323E-10-2.585E-06-2.056E-13 8.836E-14 5.053E-09-3.208E-08-1.330E-09-2.372E-13 0.         0.
```

POINTS			TYPE	TIME	COMP1	COMP2	COMP3	SIG1	SIG2	SIG3
19,	20,	21	1	76.0000	784923.14	.611442	.102442	20.000000	.000030	.000030
22,	23,	24	1	77.0000	797463.16	.620511	.099300	20.000000	.000030	.000030
25,	26,	27	1	78.0000	809871.47	.629241	.096400	20.000000	.000030	.000030
28,	29,	30	2	78.0000	938712.82	-1.456932	.077468	20.000000	.000030	.000030
31,	32,	33	1	79.0000	822141.49	.637645	.093739	20.000000	.000030	.000030
34,	35,	36	1	80.0000	834266.85	.645753	.091282	20.000000	.000030	.000030
37,	38,	39	2	80.0000	914525.81	-1.441376	.079653	20.000000	.000030	.000030

```
TIME 8.00000000E+01
V(R) 1.38836925E+04 G(R)-1.60029376E+00 L(R) 6.58255320E+01 ALT  9.25467048E+04 LAT  1.69127035E+01 LON  2.84242341E+01
U(I) 5.68338413E+03 V(I) 1.41280452E+04 W(I) 3.87726600E+02 PSI  7.74209018E+01 THE  1.94219603E+01 PHI  2.48472847E+01
U(B) 1.26924926E+04 V(B)-4.78055747E+02 W(B) 5.60615836E+03 SIG  2.34308556E+01 BET -1.97324971E+00 ALF  2.38306109E+01
[illegible]         YP   0.             ZP   0.             V(T) 5.62650414E+03 XZI -4.87401043E+00 ETA  2.20607403E+01
[illegible]              [illegible]    R    1.01728525E-03 AX  -1.44515592E+02 AY   1.78077358E+01 AZ  -2.08212031E+02
PM  -1.09490564E-03 QM   8.46549342E-03 RM   1.01728525E-03 CX  -1.63793445E-01 CY   2.01832228E-02 CZ  -2.35986757E-01
DP   4.45810372E+03  M   1.40444332E+01 RHO  4.62562257E-05 C1   2.45175750E-01 C2   2.01832228E-02 C3   1.49689324E-01
E0   7.73819629E-01 E1   6.24517880E-02 E2   2.61156707E-01 E3   5.73672494E-01 LATC 1.68054335E+01 RE   3.13326526E+07
UW   0.             VW   0.
C 46-1.02775455E-04 C103 2.87979531E-01 C104 5.41051790E-01 C
S(  1) 6.806076E+00 S(  2) 3.300555E-04 S(  3) 4.771417E-04 S(  4) 1.288265E+01 S(  5) 7.332071E-07 S(  6) 7.759604E-07
S(  7) 4.093697E-02 S(  8) 1.459723E-02 S(  9) 2.853961E-03 S( 46) 0.             S(103) 0.             S(104) 9.207285E-07
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
 4.632E+01
-5.890E-04 1.089E-07
-2.145E-04-1.143E-07 2.277E-07
-1.505E+01 2.982E-03-2.258E-03 1.660E+02
 1.485E-06 7.328E-11-2.544E-10 1.199E-06 5.376E-13
 3.603E-06-8.456E-11 1.121E-10-2.073E-06-5.626E-14 6.021E-13
-8.365E-02 0.         0.        -1.157E-01-2.752E-09-2.992E-09 1.676E-03
-3.227E-02-1.228E-06 0.        -1.933E-02 7.623E-10-2.040E-09 5.704E-04 2.131E-04
-1.603E-02 4.278E-08 9.668E-08 2.819E-03-3.029E-10-9.979E-10 8.261E-05 3.089E-05 8.145E-06
 4.763E-09 2.556E-09 3.913E-05 3.913E-05-3.847E-12-4.821E-12-1.661E-07-6.849E-08 1.723E-07 0.
 3.251E-11-1.437E-10 4.702E-07 4.702E-07 3.862E-13-5.573E-14 6.539E-12 8.623E-10-6.971E-11 0.        0.
-4.287E-11 6.902E-11-1.391E-06-1.391E-06-4.833E-14 4.377E-13-1.059E-09-8.475E-10-4.561E-10 0.        0.         8.477E-13
CORRELATION MATRIX (CUZ)TRANSPOSE
-2.405E-06-1.792E-10 1.069E-10-2.505E-06-3.881E-14-9.020E-14 8.502E-08 3.220E-08 3.723E-09-4.151E-12 1.328E-14-2.894E-14
-3.407E-06-6.545E-10 1.878E-09-3.313E-06-7.820E-13 2.513E-13-1.833E-08-5.619E-08-1.412E-09-1.022E-11-1.284E-13 4.060E-14
```

POINTS			TYPE	TIME	COMP1	COMP2	COMP3	SIG1	SIG2	SIG3
40,	41,	42	1	81.0000	846245.26	.653569	.089040	20.000000	.000030	.000030
43,	44,	45	1	82.0000	858073.13	.661114	.086983	20.000000	.000030	.000030
46,	47,	48	2	82.0000	891213.20	-1.425722	.082262	20.000000	.000030	.000030
49,	50,	51	2	84.0000	868752.44	-1.409972	.085290	20.000000	.000030	.000030

```
TIME 8.50000000E+01
V(R) 1.28487472E+04 G(R) 3.56572674E-01 L(R) 6.71563070E+01 ALT  9.18874943E+04 LAT  1.69857162E+01 LON  2.85987062E+01
U(I) 4.98802448E+03 V(I) 1.33070001E+04 W(I)-7.99619558E+01 PSI  7.87793574E+01 THE  2.16930459E+01 PHI  2.39315365E+01
U(R) 1.17232832E+04 V(R)-5.09210095E+02 W(R) 5.23408451E+03 SIG  2.21462941E+01 BET -2.27129007E+00 ALF  2.40592889E+01
XP   0.             YP   0.             ZP   0.             V(T) 5.25879602E+03 XZI -5.55466636E+00 ETA  2.22585360E+01
P   -1.27096643E-03 Q    7.93022881E-03 R    9.96728523E-04 AX  -1.28680039E+02 AY   1.82683560E+01 AZ  -1.87312983E+02
PM  -1.27096643E-03 QM   8.01514722E-03 RM   9.96728523E-04 CX  -1.64665135E-01 CY   2.33770626E-02 CZ  -2.39694656E-01
DP   3.94557200E+03 M    1.30147285E+01 RHO  4.77989544E-05 C1   2.48078837E-01 C2   2.33770626E-02 C3   1.51739993E-01
F0   7.67306422E-01 F1   4.05479111E-02 F2   2.71491173E-01 E3   5.79559544E-01 LATC 1.68780381E+01 RE   3.00299192E+07
UW   0.             VW   0.
C 46-8.49184054E-05 C103 2.87979609E-01 C104 5.41051854E-01 C
S(  1) 6.740534E+00 S(  2) 3.412897E-04 S(  3) 5.035895E-04 S(  4) 1.800481E+01 S(  5) 1.286662E-06 S(  6) 1.182860E-06
S(  7) 3.450646E-02 S(  8) 1.153834E-02 S(  9) 2.241204E-03 S( 46) 0.           S(103) 0.           S(104) 7.335804E-07
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
 4.543E+01
 1.794E-04 1.165E-07
-1.245E-03-1.360E-07 2.536E-07
 5.300E+00 5.166E-03-5.670E-03 3.242E+02
 4.509E-06 2.043E-10-5.534E-10 8.552E-06 1.656E-12
 6.090E-06-7.612E-11 6.074E-11-4.132E-06 1.341E-13 1.399E-12
-1.013E-01 0.        0.        -3.258E-01-1.463E-08-8.065E-09 1.191E-03
-3.262E-02-2.597E-06 0.        -9.048E-02-2.094E-09-3.323E-09 3.909E-04 1.331E-04
-1.332E-02-1.936E-07 4.504E-07-6.162E-03-1.143E-09-1.450E-09 5.128E-05 1.709E-05 5.023E-06
 1.554E-08 5.073E-09 3.868E-04 3.868E-04-3.211E-11-4.638E-11-5.719E-07-2.107E-07 2.535E-07 0.
 3.240E-11-1.376E-10 1.330E-06 1.330E-06 5.782E-13-2.665E-14-9.795E-10 1.214E-10-1.404E-10 0.        0.
-1.178E-11 2.827E-11-1.023E-06-1.023E-06-1.743E-14 5.240E-13-1.516E-09-6.190E-10-2.390E-10 0.        0.        5.381E-13
CORRELATION MATRIX (CUZ)TRANSPOSE
-4.674E-06-4.841E-10 3.943E-10-1.413E-05-5.152E-13-3.370E-13 1.159E-07 4.375E-08 4.608E-09-1.703E-11-3.266E-14-5.133E-14
-8.273E-06-1.506E-09 5.207E-09-2.339E-05-6.240E-12 1.696E-12-6.365E-08-9.993E-08-3.572E-09-2.284E-11-6.500E-13 7.410E-15

    POINTS      TYPE     TIME         COMP1          COMP2          COMP3          SIG1           SIG2           SIG3
  52,  53,  54   2     86.0000      847113.19      -1.394128       .088741      20.000000        .000030        .000030
  55,  56,  57   2     88.0000      826256.65      -1.378176       .092599      20.000000        .000030        .000030


TIME 9.00000000E+01
V(P) 1.19554237E+04 G(R) 2.06815112E+00 L(R) 6.83780005E+01 ALT  9.32692875E+04 LAT  1.70498550E+01 LON  2.87620795E+01
U(I) 4.40248317E+03 V(I) 1.25727819E+04 W(I)-4.31449824E+02 PSI  8.01089906E+01 THE  2.37172370E+01 PHI  2.35771291E+01
U(R) 1.08836235E+04 V(R)-5.02356181E+02 W(R) 4.92204576E+03 SIG  2.14226721E+01 BET -2.40822629E+00 ALF  2.43345424E+01
XP   0.             YP   0.             ZP   0.             V(T) 4.94761520E+03 XZI -5.82757030E+00 ETA  2.24816576E+01
P   -1.40626441E-03 Q    6.71335986E-03 R    9.56333835E-04 AX  -1.04926882E+02 AY   1.57921351E+01 AZ  -1.54454736E+02
PM  -1.40626441E-03 QM   6.75272832E-03 RM   9.56333835E-04 CX  -1.65981227E-01 CY   2.49811859E-02 CZ  -2.44328108E-01
DP   3.18930794E+03 M    1.20763478E+01 RHO  4.46268781E-05 C1   2.51913384E-01 C2   2.49811859E-02 C3   1.54225935E-01
F0   7.60313690E-01 E1   2.35892890E-02 E2   2.82641664E-01 E3   5.84363181E-01 LATC 1.69418189E+01 RE   2.59682438E+07
UW   0.             VW   0.
C 46-3.93684637E-05 C103 2.87979615E-01 C104 5.41051857E-01 C
S(  1) [illegible] S(  2) [illegible]-04 S(  3) 4.563711E-04 S(  4) 2.192707E+01 S(  5) 1.705938E-06 S(  6) 1.491527E-06
S(  7) 3.620497E-02 S(  8) [illegible] S(  9) 1.955745E-03 S( 46) 0.           S(103) 0.           S(104) 7.252491E-07
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
 4.294E+01
-8.985E-05 1.188E-07
-1.159E-03-1.175E-07 2.083E-07
 6.822E+00 6.515E-03-6.811E-03 4.808E+02
 5.907E-06 2.024E-10-6.362E-10 1.315E-05 2.910E-12
 7.282E-06-6.841E-11-7.031E-12-3.621E-06 4.863E-13 2.226E-12
-4.456E-02 0.        0.        -3.893E-01-1.189E-08-5.486E-09 6.867E-04
-1.329E-02-2.385E-06 0.        -1.176E-01-2.453E-09-2.048E-09 2.247E-04 7.586E-05
-1.178E-02-8.329E-09 3.167E-07-2.636E-03-1.217E-09-1.612E-09 1.890E-05 6.124E-06 3.825E-06
 2.664E-08 1.190E-09 9.588E-04 9.588E-04-4.986E-11-8.796E-11-9.120E-07-3.126E-07 2.735E-07 0.
 9.832E-12-1.279E-10 9.246E-07 9.246E-07 9.135E-13 2.455E-14 9.819E-10 6.185E-10-6.777E-11 0.        0.
 1.191E-11 7.267E-12 1.523E-07 1.523E-07 2.835E-14 6.842E-13-1.456E-09-4.826E-10-8.534E-11 0.        0.        5.260E-13
CORRELATION MATRIX (CUZ)TRANSPOSE
-4.651E-06-7.411E-10 5.259E-10-2.869E-05-6.680E-13-4.007E-13 1.376E-07 5.301E-08 4.996E-09-2.850E-11 6.562E-14-4.483E-14
-6.993E-06-1.867E-09 8.239E-09-1.603E-05-1.240E-11 4.332E-12-1.361E-07-1.497E-07-6.644E-09-2.112E-11-1.392E-13-2.952E-13


********************BEGIN BACKWARDS SMOOTHING*****************************************************************
```

```
TIME 9.00000000E+01
V(R) 1.19554237E+04 G(R) 2.06815112E+00 L(R) 6.83780005E+01 ALT  9.32692875E+04 LAT  1.70498550E+01 LON  2.87620795E+01
U(I) 4.40248317E+03 V(I) 1.25727819E+04 W(I)-4.31449824E+02 PSI  8.01089906E+01 THE  2.37172370E+01 PHI  2.35771291E+01
U(B) 1.08836235E+04 V(B)-5.02356181E+02 W(B) 4.92204576E+03 SIG  2.14226721E+01 BET -2.40822629E+00 ALF  2.43345424E+01
XP   0.             YP   0.             ZP   0.             V(T) 4.94761520E+03 XZI -5.82757030E+00 ETA  2.24816576E+01
P   -1.40626441E-03 Q    6.71335986E-03 R    9.56333835E-04 AX  -1.04926882E+02 AY   1.57921351E+01 AZ  -1.54454736E+02
PM  -1.40626441E-03 QM   6.75272832E-03 RM   9.56333835E-04 CX  -1.65981227E-01 CY   2.49811859E-02 CZ  -2.44328108E-01
DP   3.18930794E+03 M    1.20763478E+01 RHO  4.46268781E-05 C1   2.51913384E-01 C2   2.49811859E-02 C3   1.54225935E-01
E0   7.60313690E-01 E1   2.35892890E-02 E2   2.82641664E-01 E3   5.84363181E-01 LATC 1.69418189E+01 RE   2.59682438E+07
UW   0.             VW   0.
S(  1) 6.552922E+00 S(  2) 3.447435E-04 S(  3) 4.563711E-04 S(  4) 2.192707E+01 S(  5) 1.705938E-06 S(  6) 1.491527E-06
S(  7) 2.620497E-02 S(  8) 8.709750E-03 S(  9) 1.955745E-03 S( 46) 0.           S(103) 0.           S(104) 7.252491E-07
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
 4.294E+01
-8.985E-05 1.188E-07
-1.159E-03-1.175E-07 2.083E-07
 6.822E+00 6.515E-03-6.811E-03 4.808E+02
 5.907E-06 2.024E-10-6.362E-10 1.315E-05 2.910E-12
 7.282E-06-6.861E-11-7.031E-12-3.621E-06 4.863E-13 2.225E-12
-4.456E-02 0.        0.        -3.893E-01-1.189E-08-5.486E-09 6.867E-04
-1.329E-02-2.385E-06 0.        -1.176E-01-2.453E-09-2.048E-09 2.247E-04 7.586E-05
-1.178E-02-8.329E-09 3.167E-07-2.636E-03-1.217E-09-1.612E-09 1.890E-05 6.124E-06 3.825E-06
 2.664E-08 1.190E-09 9.588E-04 9.588E-04-4.986E-11-8.796E-11-9.120E-07-3.126E-07 2.735E-07 0.
 9.832E-12-1.279E-10 9.246E-07 9.246E-07 9.135E-13 2.455E-14 9.819E-10 6.185E-10-6.777E-11 0.        0.
 1.191E-11 7.267E-12 1.523E-07 1.523E-07 2.835E-14 6.842E-13-1.456E-09-4.826E-10-8.534E-11 0.        0.        5.260E-13
CORRELATION MATRIX (CUZ)TRANSPOSE
-4.651E-06-7.411E-10 5.259E-10-2.869E-05-6.680E-13-4.007E-13 1.376E-07 5.301E-08 4.996E-09-2.850E-11 6.562E-14-4.483E-14
-6.993E-06-1.867E-09 8.239E-09-1.603E-05-1.240E-11 4.332E-12-1.361E-07-1.497E-07-6.644E-09-2.112E-11-1.392E-13-2.952E-13

    POINTS     TYPE    TIME       RES1          RES2         RES3        WGT.RES1      WGT.RES2      WGT.RES3      LOSS FCTN
                  2    88.000  8.81716E+00  2.29483E-05 -2.66269E-05  4.40858E-01  7.64943E-01 -8.87564E-01  1.56726E+00
  54,  53,  52    2    86.000  2.51242E+00  6.01273E-06 -2.07197E-05  1.25621E-01  2.00424E-01 -6.90657E-01  2.10022E+00

TIME 8.50000000E+01
V(R) 1.28472466E+04 G(R) 3.58006627E-01 L(R) 6.71738030E+01 ALT  9.18954181E+04 LAT  1.69856578E+01 LON  2.85987195E+01
U(I) 4.98382560E+03 V(I) 1.33071388E+04 W(I)-8.02741411E+01 PSI  7.88483700E+01 THE  2.17031346E+01 PHI  2.34676792E+01
U(B) 1.17190580E+04 V(B)-5.61520723E+02 W(B) 5.23451228E+03 SIG  2.17211533E+01 BET -2.50505193E+00 ALF  2.40687220E+01
XP   0.             YP   0.             ZP   0.             V(T) 5.26454407E+03 XZI -6.12286345E+00 ETA  2.22828348E+01
P   -1.27096643E-03 Q    7.97577875E-03 R    9.96728523E-04 AX  -1.28687875E+02 AY   2.01413029E+01 AZ  -1.87386647E+02
PM  -1.27096643E-03 QM   8.01514722E-03 RM   9.96728523E-04 CX  -1.64778691E-01 CY   2.57899783E-02 CZ  -2.39939670E-01
DP   3.94309308E+03 M    1.30130011E+01 RHO  4.77800832E-05 C1   2.48307423E-01 C2   2.57899783E-02 C3   1.51876547E-01
E0   7.67113067E-01 E1   3.72231288E-02 E2   2.69228703E-01 E3   5.81091978E-01 LATC 1.68779800E+01 RE   3.00137644E+07
UW   0.             VW   0.
S(  1) 3.186541E+00 S(  2) 1.767405E-04 S(  3) 2.400212E-04 S(  4) 1.007031E+01 S(  5) 8.053660E-07 S(  6) 7.548484E-07
S(  7) 2.591052E-02 S(  8) 9.202198E-03 S(  9) 1.494262E-03 S( 46) 0.           S(103) 0.           S(104) 7.252491E-07
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
 1.015E+01
 6.031E-06 3.124E-08
-2.296E-04-2.145E-08 5.761E-08
 2.197E-01 9.328E-04-4.879E-04 1.014E+02
 8.783E-07 7.988E-12-1.371E-10 1.615E-07 6.486E-13
 1.300E-06 3.483E-12 2.862E-11-3.619E-07-2.735E-14 5.698E-13
-2.162E-02 0.        0.        -4.177E-02 2.599E-09-2.909E-09 6.714E-04
-6.995E-03-1.064E-06 0.        -1.243E-02 1.365E-09-1.078E-09 2.358E-04 8.468E-05
-3.266E-03-8.000E-08 7.837E-08-7.836E-04 1.466E-11-1.667E-10 2.617E-05 9.173E-06 2.233E-06
 8.152E-09 2.865E-09-3.738E-05-3.738E-05-8.469E-13 2.107E-11-9.028E-07-3.326E-07 1.328E-07 0.
 9.951E-12-1.062E-10 2.032E-07 2.032E-07 5.185E-13-2.938E-14 9.618E-10 6.687E-10-5.106E-11 0.        0.
 1.513E-12 1.929E-11-3.536E-07-3.536E-07-2.250E-14 4.437E-13-1.439E-09-5.157E-10-1.236E-10 0.        0.        5.260E-13
CORRELATION MATRIX (CUZ)TRANSPOSE
-8.058E-07-1.999E-10-8.729E-12-3.093E-07 1.893E-13-8.682E-14 9.055E-08 3.547E-08 3.646E-09-2.850E-11 6.562E-14-4.483E-14
-8.075E-07 3.813E-11 2.001E-09 3.505E-05 1.077E-12-5.946E-13-1.525E-07-1.152E-07-7.449E-09-2.112E-11-1.392E-13-2.952E-13
```

```
   POINTS    TYPE   TIME      RES1          RES2          RES3          WGT.RES1      WGT.RES2      WGT.RES3      LOSS FCTN
 51, 50,  49 2      84.000 -1.11081E+00 -4.93179E-06 -1.31905E-05 -5.55405E-02 -1.64393E-01 -4.39682E-01  2.32365E+00
 48, 47,  46 2      82.000 -3.15169E+00 -1.02173E-05 -4.68689E-06 -1.57584E-01 -3.40577E-01 -1.56230E-01  2.48889E+00
 45, 44,  43 1      82.000 -4.92528E+00 -3.05396E-07 -4.59258E-06 -2.46264E-01 -1.01799E-02 -1.53086E-01  2.57307E+00
 42, 41,  40 1      81.000 -4.46646E+00  1.42205E-06 -2.28316E-07 -2.23323E-01  4.74017E-02 -7.61052E-03  2.62525E+00
 39, 38,  37 2      80.000 -4.54573E+00 -1.06611E-05  3.96539E-06 -2.27287E-01 -3.55368E-01  1.32180E-01  2.82067E+00
 36, 35,  34 1      80.000 -3.19663E+00  2.55063E-06  4.18142E-06 -1.59832E-01  8.50211E-02  1.39381E-01  2.87287E+00


TIME 8.00000000E+01
V(R) 1.38827931E+04 G(R)-1.58868337E+00 L(R) 6.58464283E+01 ALT  9.25656539E+04 LAT  1.69126709E+01 LON  2.84242584E+01
U(I) 5.67843029E+03 V(I) 1.41293698E+04 W(I) 3.84889372E+02 PSI  7.74884300E+01 THE  1.94403555E+01 PHI  2.33182576E+01
U(B) 1.26889643E+04 V(B)-6.37336577E+02 W(B) 5.59606400E+03 SIG  2.20294477E+01 BET -2.63128139E+00 ALF  2.37983516E+01
XP   0.             YP   0.             ZP   0.             V(T) 5.63224025E+03 XZI -6.49742889E+00 ETA  2.20823630E+01
P   -1.09490564E-03 Q    8.42612495E-03 R    1.01728525E-03 AX  -1.44449586E+02 AY   2.37001913E+01 AZ  -2.07771837E+02
PM  -1.09490564E-03 QM   8.46549342E-03 RM   1.01728525E-03 CX  -1.63894121E-01 CY   2.68905030E-02 CZ  -2.35740259E-01
DP   4.45333027E+03 M    1.40429897E+01 RHO  4.62126843E-05 C1   2.45084080E-01 C2   2.68905030E-02 C3   1.49561178E-01
E0   7.74243129E-01 E1   5.18706093E-02 E2   2.53627120E-01 E3   5.77520823E-01 LATC 1.68054011E+01 RE   3.12991604E+07
UW   0.             VW   0.
S( 1) 2.047869E+00 S( 2) 9.251443E-05 S( 3) 1.815181E-04 S( 4) 7.039200E+00 S( 5) 4.096122E-07 S( 6) 4.495324E-07
S( 7) 2.556625E-02 S( 8) 9.891964E-03 S( 9) 1.413456E-03 S( 46) 0.           S(103) 0.           S(104) 7.252491E-07
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
 4.194E+00
-2.191E-05 8.559E-09
-1.278E-04-4.197E-09 3.295E-08
 8.881E-01 1.812E-04-9.247E-05 4.955E+01
 1.018E-07-3.161E-12-1.863E-11-4.234E-08 1.678E-13
 2.994E-07 3.493E-12 1.472E-11-3.852E-07-2.965E-14 2.021E-13
 1.762E-02 0.        0.        4.462E-02-5.826E-10-3.708E-10 6.536E-04
 7.026E-03 2.278E-07 0.        1.442E-02-3.230E-10-5.081E-11 2.504E-04 9.785E-05
 5.357E-04 3.306E-08-1.563E-07 2.058E-03-2.382E-11 4.170E-11 3.329E-05 1.281E-05 1.998E-06
-4.012E-09 4.032E-09-1.093E-04-1.093E-04 7.236E-12 1.212E-11-8.903E-07-3.596E-07-1.405E-08 0.
 6.333E-12-7.593E-11-3.349E-07-3.349E-07 1.964E-13-4.162E-14 9.359E-10 7.319E-10-3.401E-11 0.        0.
-4.401E-12 2.830E-11-1.780E-07-1.780E-07-3.094E-14 2.218E-13-1.418E-09-5.591E-10-1.652E-10 0.        0.        5.260E-13
CORRELATION MATRIX (CUZ)TRANSPOSE
 2.323E-06 1.236E-10-2.763E-10 7.782E-07-3.630E-13-5.587E-14 4.383E-08 1.727E-08 2.024E-09-2.850E-11 6.562E-14-4.483E-14
-4.960E-06 6.066E-10-6.446E-10 4.135E-06 2.157E-12-5.896E-13-1.692E-07-7.811E-08-8.789E-09-2.112E-11-1.392E-13-2.952E-13


   POINTS    TYPE   TIME      RES1          RES2          RES3          WGT.RES1      WGT.RES2      WGT.RES3      LOSS FCTN
 33, 32,  31 1      79.000 -1.16035E+00  3.22293E-06  8.49005E-06 -5.80173E-02  1.07431E-01  2.83002E-01  2.96787E+00
 30, 29,  28 2      78.000 -6.19738E+00 -7.37602E-06  1.18348E-05 -3.09869E-01 -2.45867E-01  3.94492E-01  3.27996E+00
 27, 26,  25 1      78.000  1.58505E+00  3.60155E-06  1.25346E-05  7.92527E-02  1.20052E-01  4.17819E-01  3.47523E+00
 24, 23,  22 1      77.000  4.96830E+00  3.86681E-06  1.61388E-05  2.48415E-01  1.28894E-01  5.37960E-01  3.84295E+00
 21, 20,  19 1      76.000  8.90346E+00  4.21347E-06  1.91174E-05  4.45173E-01  1.40449E-01  6.37248E-01  4.46694E+00
 18, 17,  16 1      75.000  1.32897E+01  4.84690E-06  2.12810E-05  6.64487E-01  1.61563E-01  7.09367E-01  5.43779E+00


TIME 7.50000000E+01
V(R) 1.49620564E+04 G(R)-3.50931506E+00 L(R) 6.45270147E+01 ALT  9.57506942E+04 LAT  1.68297235E+01 LON  2.82380703E+01
U(I) 6.42289684E+03 V(I) 1.49499584E+04 W(I) 9.15839650E+02 PSI  7.61662630E+01 THE  1.71041299E+01 PHI  2.31785239E+01
U(B) 1.37024886E+04 V(B)-7.23500048E+02 W(B) 5.96502176E+03 SIG  2.23413317E+01 BET -2.77165565E+00 ALF  2.35246470E+01
XP   0.             YP   0.             ZP   0.             V(T) 6.00873838E+03 XZI -6.91564882E+00 ETA  2.18803171E+01
P   -9.42614978E-04 Q    7.69522527E-03 R    9.82154065E-04 AX  -1.42558057E+02 AY   2.45761403E+01 AZ  -2.02428780E+02
PM  -9.42614978E-04 QM   7.73459373E-03 RM   9.82154065E-04 CX  -1.63094206E-01 CY   2.81164473E-02 CZ  -2.31589585E-01
DP   4.41994124E+03 M    1.50389447E+01 RHO  3.94878889E-05 C1   2.41976679E-01 C2   2.81164473E-02 C3   1.47243821E-01
E0   7.80928947E-01 E1   6.60275258E-02 E2   2.37783293E-01 E3   5.73802625E-01 LATC 1.67229182E+01 RE   2.85227348E+07
UW   0.             VW   0.
S( 1) 3.744827E+00 S( 2) 1.340256E-04 S( 3) 2.379396E-04 S( 4) 7.285943E+00 S( 5) 8.280703E-07 S( 6) 3.943657E-07
S( 7) 2.521082E-02 S( 8) 1.068616E-02 S( 9) 1.842794E-03 S( 46) 0.           S(103) 0.           S(104) 7.252491E-07
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
```

```
 1.402E+01
 3.002E-04 1.796E-08
-6.631E-04-2.258E-08 5.662E-08
-9.842E+00-3.982E-04 4.251E-04 5.308E+01
-2.543E-06-7.351E-11 1.553E-10 2.019E-06 6.857E-13
-6.281E-07-2.079E-11 4.141E-11 1.667E-07 1.437E-13 1.555E-13
 8.363E-02 0.        0.        -6.582E-02-1.915E-08-4.794E-09 6.356E-04
 3.535E-02 1.018E-06 0.        -3.076E-02-8.213E-09-1.948E-09 2.663E-04 1.142E-04
 5.737E-03 1.516E-07-3.221E-07-5.235E-03-1.152E-09-1.739E-10 4.077E-05 1.726E-05 3.396E-06
-3.677E-09 5.946E-09 2.686E-04 2.686E-04 1.434E-11-1.375E-11-8.764E-07-3.899E-07-1.726E-07 0.
-1.057E-12-3.860E-11-4.702E-07-4.702E-07-1.537E-14-3.001E-14 9.067E-10 8.014E-10-1.641E-11 0.        0.
-2.829E-12 3.328E-11 2.333E-07 2.333E-07 1.728E-14 6.438E-14-1.395E-09-6.073E-10-2.121E-10 0.        0.        5.260E-13
CORRELATION MATRIX (CUZ)TRANSPOSE
 4.995E-06 2.028E-10-4.140E-10-1.147E-05-1.829E-12-3.880E-13-2.295E-09-2.194E-09 1.389E-10-2.850E-11 6.562E-14-4.483E-14
-2.490E-05-4.975E-10 1.193E-09-1.227E-05 3.186E-12 2.518E-12-1.873E-07-3.660E-08-1.066E-08-2.112E-11-1.392E-13-2.952E-13

    POINTS     TYPE   TIME      RES1         RES2         RES3         WGT.RES1     WGT.RES2     WGT.RES3     LOSS FCTN
  15,  14,  13  1     74.000  1.80118E+01  5.97863E-06  2.24414E-05  9.00589E-01  1.99288E-01  7.48046E-01  6.84814E+00
  12,  11,  10  1     73.000  2.29408E+01  7.82160E-06  2.24174E-05  1.14704E+00  2.60720E-01  7.47246E-01  8.79018E+00
   9,   8,   7  1     72.000  2.79360E+01  1.05855E-05  2.10403E-05  1.39680E+00  3.52849E-01  7.01343E-01  1.13576E+01
   6,   5,   4  1     71.000  3.28470E+01  1.44720E-05  1.81588E-05  1.64235E+00  4.82400E-01  6.05294E-01  1.46540E+01
   3,   2,   1  1     70.000  3.75168E+01  1.96710E-05  1.36433E-05  1.87584E+00  6.55699E-01  4.54777E-01  1.88096E+01


TIME 7.00000000E+01
V(R) 1.59397314E+04 G(R)-5.12193481E+00 L(R) 6.33688559E+01 ALT  1.01564920E+05 LAT  1.67368207E+01 LON  2.80410136E+01
U(I) 7.11637590E+03 V(I) 1.56606403E+04 W(I) 1.42302923E+03 PSI  7.50264993E+01 THE  1.52554851E+01 PHI  2.30996772E+01
U(B) 1.46262669E+04 V(B)-8.07034828E+02 W(B) 6.28458815E+03 SIG  2.26472466E+01 BET -2.90214852E+00 ALF  2.32521428E+01
XP   0.             YP   0.             ZP   0.             V(T) 6.33619393E+03 XZI -7.31758453E+00 ETA  2.16782886E+01
P   -8.75453186E-04 Q    6.00554706E-03 R    8.82155972E-04 AX  -1.21334273E+02 AY   2.18520195E+01 AZ  -1.70094573E+02
PM  -8.75453186E-04 QM   6.04491552E-03 RM   8.82155972E-04 CX  -1.62260392E-01 CY   2.92227180E-02 CZ  -2.27467569E-01
DP   3.78412316E+03 M    1.58403752E+01 RHO  2.97874455E-05 C1   2.38880268E-01 C2   2.92227180E-02 C3   1.44934976E-01
E0   7.86457630E-01 E1   7.82195103E-02 E2   2.24000367E-01 E3   5.70254276E-01 LATC 1.66305368E+01 RE   2.24965894E+07
UW   0.             VW   0.
S(  1) 8.670433E+00 S(  2) 1.820709E-04 S(  3) 3.083126E-04 S(  4) 1.431983E+01 S(  5) 2.099586E-06 S(  6) 1.207621E-06
S(  7) 2.490332E-02 S(  8) 1.144140E-02 S(  9) 2.606191E-03 S( 46) 0.           S(103) 0.           S(104) 7.252491E-07
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY
 7.518E+01
 2.820E-04 3.315E-08
-1.568E-03-3.792E-08 9.506E-08
-4.768E+01-2.095E-03 2.421E-03 2.051E+02
-1.522E-05-2.198E-10 5.204E-10 1.826E-05 4.408E-12
-9.391E-06-5.010E-11 1.554E-10 6.848E-06 1.993E-12 1.458E-12
 1.752E-01 0.        0.        -2.241E-01-4.997E-08-2.474E-08 6.202E-04
 7.957E-02 1.000E-06 0.        -9.609E-02-2.205E-08-1.158E-08 2.805E-04 1.309E-04
 2.153E-02 4.123E-08-4.245E-07-1.194E-02-4.138E-09-2.483E-09 4.870E-05 2.238E-05 6.792E-06
 9.504E-09 9.401E-09-6.632E-05-6.632E-05 8.421E-11 6.859E-11-8.640E-07-4.181E-07-3.428E-07 0.
-1.015E-11 5.171E-13-8.492E-09-8.492E-09-8.473E-14-1.517E-14 8.791E-10 8.666E-10 1.224E-12 0.        0.
 5.360E-12 3.451E-11 1.366E-07 1.366E-07 1.354E-13 1.806E-14-1.375E-09-6.521E-10-2.624E-10 0.        0.        5.260E-13
CORRELATION MATRIX (CUZ)TRANSPOSE
 7.232E-06 1.219E-10-5.978E-10-2.221E-05-4.136E-12-9.635E-13-4.787E-08-2.289E-08-2.011E-09-2.850E-11 6.562E-14-4.483E-14
 6.720E-05-2.872E-09 7.083E-09 9.597E-05 2.080E-11 6.148E-12-2.071E-07 9.780E-09-1.318E-08-2.112E-11-1.392E-13-2.952E-13
```

b. STEP2 output:

```
DATA ANALYSIS PROGRAM
INPUT DATA

    1         STEP2 FILTERING CHECK PROBLEM (FEB. 1969)  WAGNER AND SEROLD
    2         0  1  2  0  0  2  1  3  0  0                          PROGRAM CO     NTROLS
    3       1   1  16001.2000  100.                    (SIG=100.   ,RN=.55)
    3       1   2 -6.75088000  .035                    (SIG=2 DEG  ,RN=-.81)
    3       1   3  63.0352000  .018                    (SIG=1 DEG  ,RN=-.33)
    3       1   4  100584.000 1000.                    (SIG=1000.  ,RN=-.96)
    3       1   5  16.7583977 .0003                    (SIG=.017DEG,RN=1.69)
    3       1   6  28.0418380 .0003                    (SIG=.017DEG,RN=.05)
    3       1   7  29.8270000 .0530                    (SIG=3 DEG  ,RN=1.10)
    3       1   8 -4.43200000 .0350                    (SIG=2 DEG  ,RN=-1.72)
    3       1   9  19.3930000 .0530                    (SIG=3 DEG  ,RN=-1.37)
    3       2  72  .092200000 .2000                    (SIG=.20    ,RN=-.54)
    3       2 103 16.6055454  .00037                   (SIG=.0214 DEG) MOD.PARA.   - Z
    3       2 104 31.0        .000375                  (SIG=.0214 DEG) MOD.PARA.   - Z
    3       3  70  0.         .20                                      MOD.PARA.   - U
    3       3  71  0.         .20                                      MOD.PARA.   - U
    3       4  76  1.0        .000030                                  MEAS.PARA.  - V
    3       4  81  0.         .000030
    3       5  31 -1.0                                                 CONSTANT
    3       5  33  0.5                                                 CONSTANT
    3       5  88 15.09690377                                          CONSTANT
    3       5  89 27.0                                                 CONSTANT
    6       1   2                                                      CG/IMU
0.         1.0000E+00  1.0000E+02  1.00uuE+00
    6       3   2                                                      CG/IMU
0.        -5.0000E-01  1.0000E+02 -5.0000E-01
    4            70.0        5.0         0.5        90.0
    5       1       32.174048    1.407690E16 1092.055E-6               GRAV. DATA
    5       2       20855970.    20926430.   .7292115E-4               GEOPH.DATA
    7       1 70.0        82.0       .99      1 1 1  20.0    .00003    .00003     M    EAS.CDS.
    7       2 78.0        88.0      1.99      1 1 1  20.0    .00003    .00003     M    EAS.CDS.
                                                                       END CARD
```

```
CONTROLS ARE SPECIFIED FOR
   *FILTERING RUN FROM TIME TO=  70.0000 TO TFINAL=  90.0000
   *UPDATED REFERENCE
   *SMOOTH COVARIANCE,CALCULATE RESIDUALS AND LOSS FUNCTION
   *FILTER FOR  1 ITERATIONS
   *VECTOR PROCESS FITTING DATA TRIPLES
   *INPUT AND OUTPUT IN ENGLISH UNITS (FT,FT/SEC,FT/SEC2,LB,SLUG)
   *PLANET PARAMETERS
                 EQUIT.RAD.= 2.09264300E+07  POLAR RAD.= 2.08559700E+07  ROTATION  = 7.29211500E-05
                 MU(GRAV.) = 1.40769000E+16  J2(GRAV.) = 1.09205500E-03
   *INITIAL STATE ESTIMATE IS
                 VELOCITY  = 1.60012000E+04  GAMMA     =-6.75088000E+00  LAMBDA    = 6.30352000E+01
                 ALTITUDE  = 1.00584000E+05  GEOD-LAT. = 1.67583977E+01  LONGITUDE = 2.80418380E+01
                 SIGMA     = 2.98270000E+01  BETA      =-4.43200000E+00  ALPHA     = 1.93930000E+01
   *FITTING DATA IS FROM  2 SOURCES
       1. TRACKING STATION 1, GEOD.LAT.=  15.096904  LONGITUDE=  27.000000  ALTITUDE =    0.    ABOVE REF. SURFACE
       2. TRACKING STATION 2, GEOD.LAT.=  16.605545  LONGITUDE=  31.000000  ALTITUDE =    0.    ABOVE REF. SURFACE


STATE VECTOR COMPONENTS
COMP.   ID.NO.      EST.VALUE      VARIANCE
   1       1      7.20532663E+03 6.79079215E+04
   2       2      1.56313582E+04 2.80981835E+04
   3       3      1.88098346E+03 3.09451057E+05
   4       4      9.47712173E+04 1.00000000E+06
   5       5      2.90632098E-01 9.00000000E-08
   6       6      4.89422401E-01 9.00000000E-08
   7       7      7.69645202E-01 2.08169635E-04
   8       8      1.55940474E-01 7.90725687E-04
   [illegible]       9      2.98[illegible]2405E-01 8.18646927E-04
  10      10      5.82889025E-01 2.80457751E-04
MODEL PARAMETERS IN EXPANDED STATE VECTOR
  11      72      9.22000000E-02 4.00000000E-02
  12     103      2.87979327E-01 1.40625000E-07
  13     104      5.41052068E-01 1.40625000E-07
RANDOM MODEL PARAMETERS NOT BEING ESTIMATED (U-VECTOR)
   1      70      0.             4.00000000E-02
   2      71      0.             4.00000000E-02
RANDOM MEASUREMENT PARAMETERS NOT BEING ESTIMATED (V-VECTOR)
   1      76      1.00000000E+00 9.00000000E-10
   2      81      0.             9.00000000E-10
```

```
TIME 7.00000000E+01
V(A) 1.60012000E+04 G(A)-6.75088000E+00 L(A) 6.30352000E+01 ALT  1.00584000E+05 LATD 1.67583977E+01 LON  2.80418380E+01
U(I) 7.20532663E+03 V(I) 1.56313582E+04 W(I) 1.88098346E+03 PSI  7.63702516E+01 THE  8.01936612E+00 PHI  2.92188575E+01
U(B) 1.50482107E+04 V(B)-1.23650674E+03 W(B) 5.29724514E+03 SIG  2.98270000E+01 BET -4.43200000E+00 ALF  1.93930000E+01
XP   0.             YP   0.             ZP   0.             V(T) 1.73145658E+04 XZI -1.31389754E+01 ETA  1.87755893E+01
P   -8.75453186E-04 Q    6.04491552E-03 R    8.82155972E-04 AX  -1.21200856E+02 AY   7.53823858E+00 AZ  -1.72113726E+02
PM  -8.75453186E-04 QM   6.04491552E-03 RM   8.82155972E-04 AXM -1.21200856E+02 AYM  7.53823858E+00 AZM -1.72205926E+02
E0   7.69645202E-01 E1   1.55940474E-01 E2   2.08732405E-01 E3   5.82889025E-01 LATC 1.66519926E+01 A(T) 2.10640943E+02
C 72 9.22000000E-02 C103 2.87979327E-01 C104 5.41052068E-01 C
S(  1) 1.000000E+02 S(  2) 3.500000E-02 S(  3) 1.800000E-02 S(  4) 1.000000E+03 S(  5) 3.000000E-04 S(  6) 3.000000E-04
S(  7) 5.300000E-02 S(  8) 3.500000E-02 S(  9) 5.300000E-02 S( 72) 2.000000E-01 S(103) 3.750000E-04 S(104) 3.750000E-04
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 1.000E+04
-4.885E-14 1.225E-03
-1.[illegible]E-14 0.      3.240E-04
-2.274E-13 0.         0.         1.000E+06
 0.         0.         0.         0.         9.000E-08
 0.         0.         0.         0.         0.         9.000E-08
 8.809E-15 0.         0.         0.         8.272E-25 0.         2.809E-03
 3.986E-15-1.561E-17 0.         0.        -6.617E-24 0.         1.475E-17 1.225E-03
 7.855E-15-1.171E-17-1.301E-18 0.         3.309E-24 0.         4.510E-17-5.204E-18 2.809E-03
 0.         0.         0.         0.         0.         0.         0.         0.         0.         4.000E-02
 0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         1.406E-07
 0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         1.406E-07
CORRELATION MATRIX (CUZ)TRANSPOSE
 0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.
CORRELATION MATRIX (CVZ)TRANSPOSE
 0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.         0.

    POINTS      TYPE    TIME          COMP1           COMP2          COMP3          SIG1           SIG2           SIG3
   1,   2,   3    1     70.0000       714184.53       .543024        .124359        20.000000      .000030        .000030
   4,   5,   6    1     71.0000       720629.03       .560073        .121768        20.000000      .000030        .000030
   7,   8,   9    1     72.0000       733667.87       .571218        .117803        20.000000      .000030        .000030
  10,  11,  12    1     73.0000       746636.75       .581913        .113572        20.000000      .000030        .000030
  13,  14,  15    1     74.0000       759519.00       .592172        .109626        20.000000      .000030        .000030
  16,  17,  18    1     75.0000       772294.05       .602013        .105952        20.000000      .000030        .000030


TIME 7.50000000E+01
V(A) 1.49754064E+04 G(A)-3.44307998E+00 L(A) 6.45442767E+01 ALT  9.57405776E+04 LATD 1.68297058E+01 LON  2.82380568E+01
U(I) 6.42501306E+03 V(I) 1.49648714E+04 W(I) 8.99376795E+02 PSI  7.86629984E+01 THE  1.30337921E+01 PHI  3.09555216E+01
U(B) 1.39205091E+04 V(B)-9.94630937E+02 W(B) 5.43073966E+03 SIG  3.04737058E+01 BET -3.80825304E+00 ALF  2.13120036E+01
XP  -1.00000000E+00 YP   0.             ZP   5.00000000E-01 V(T) 1.63106421E+04 XZI -1.03786026E+01 ETA  2.02377132E+01
P   -9.42614978E-04 Q    7.73459373E-03 R    9.82154065E-04 AX  -1.42491976E+02 AY   8.87341034E+00 AZ  -2.04646062E+02
PM  -9.42614978E-04 QM   7.73459373E-03 RM   9.82154065E-04 AXM -1.42491786E+02 AYM  8.87341895E+00 AZM -2.04739434E+02
E0   7.59820572E-01 E1   1.35757782E-01 E2   2.52652023E-01 E3   5.83446209E-01 LATC 1.67229007E+01 A(T) 4.82360201E-02
C 72 9.36035097E-02 C103 2.87979327E-01 C104 5.41052068E-01 C
S(  1) 1.062884E+01 S(  2) 9.443782E-04 S(  3) 1.047804E-03 S(  4) 2.965420E+01 S(  5) 1.219798E-06 S(  6) 1.116628E-06
S(  7) 4.249156E-02 S(  8) 3.040992E-02 S(  9) 2.732858E-02 S( 72) 1.999375E-01 S(103) 3.750000E-04 S(104) 3.750000E-04
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 1.130E+02
 6.880E-03 8.919E-07
```

```
4.430E-03-7.117E-08 1.098E-06
6.119E+01 1.289E-02-9.150E-04 8.794E+02
7.726E-07 1.283E-10-5.545E-10 1.039E-06 1.488E-12
6.977E-06 2.203E-10 4.622E-10 1.731E-06 4.334E-13 1.247E-12
2.500E-03 0.        0.        -1.035E-01-8.533E-09 4.112E-09 1.806E-03
-1.401E-02-8.676E-06 0.       -8.744E-02-6.079E-09 2.289E-09-5.996E-04 9.248E-04
-2.564E-01-2.001E-05-1.302E-05-1.951E-01 2.077E-09-1.120E-08 1.779E-05 5.497E-06 7.469E-04
-3.925E-06-1.716E-06-4.415E-02-4.415E-02 1.470E-09 1.180E-09 6.893E-06-3.548E-09-6.074E-05 3.997E-02
0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        1.406E-07
0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        1.406E-07
CORRELATION MATRIX (CUZ)TRANSPOSE
5.498E-02-1.506E-06-3.366E-07-1.774E-02 8.735E-10 1.590E-09-3.962E-07 8.394E-06 1.135E-04-1.233E-05 0.        0.
-3.542E-03-9.435E-07 1.932E-06-1.005E-02-9.453E-10 3.219E-10-1.067E-04-6.311E-05-6.671E-06 1.242E-06 0.        0.
CORRELATION MATRIX (CVZ)TRANSPOSE
-1.338E-05-7.485E-11 4.611E-11-9.832E-05-2.687E-11-1.951E-11 2.496E-09 1.758E-09-1.466E-10-5.361E-10 0.        0.
-3.693E-07-8.119E-10 7.776E-11-6.879E-04 3.787E-12 2.784E-12 1.007E-09 8.710E-10 2.730E-09 4.320E-10 0.        0.
```

POINTS	TYPE	TIME	COMP1	COMP2	COMP3	SIG1	SIG2	SIG3
19, 20, 21	1	76.0000	784951.02	.611460	.102538	20.000000	.000030	.000030
22, 23, 24	1	77.0000	797484.21	.620538	.099380	20.000000	.000030	.000030
25, 26, 27	1	78.0000	809886.84	.629272	.096468	20.000000	.000030	.000030
28, 29, 30	2	78.0000	938702.81	-1.456939	.077501	20.000000	.000030	.000030
31, 32, 33	1	79.0000	822150.95	.637681	.093785	20.000000	.000030	.000030
34, 35, 36	1	80.0000	834275.20	.645785	.091334	20.000000	.000030	.000030
37, 38, 39	2	80.0000	914516.24	-1.441382	.079687	20.000000	.000030	.000030

```
TIME 8.00000000E+01
V(A) 1.38871850E+04 G(A)-1.55003031E+00 L(A) 6.58581919E+01 ALT  9.25678278E+04 LATD 1.69126766E+01 LON  2.84242687E+01
U(I) 5.67773089E+03 V(I) 1.41347759E+04 W(I) 3.75646032E+02 PSI  8.11573098E+01 THE  1.68754609E+01 PHI  3.15467356E+01
U(B) 1.27044889E+04 V(B)-8.99822820E+02 W(B) 5.53535816E+03 SIG  3.02423816E+01 BET -3.71509372E+00 ALF  2.35428038E+01
XP  -1.00000000E+00 YP   0.             ZP   5.00000000E-01 V(I) 1.52371135E+04 XZI -9.23318508E+00 ETA  2.19901799E+01
P   -1.09490564E-03 Q    8.46549342E-03 R    1.01728525E-03 AX  -1.44364893E+02 AY   9.00130871E+00 AZ  -2.09648954E+02
PM  -1.09490564E-03 QM   8.46549342E-03 RM   1.01728525E-03 AXM -1.44364809E+02 AYM  9.00133960E+00 AZM -2.09748555E+02
E0   7.48948638E-01 E1   1.12370823E-01 E2   2.82161245E-01 E3   5.88925944E-01 LATC 1.68054068E+01 A(T) 5.97984088E-03
C 72 9.95896561E-02 C103 2.87979347E-01 C104 5.41052072E-01 C
S(  1) 4.691653E+00 S(  2) 5.074014E-04 S(  3) 5.358290E-04 S(  4) 1.841072E+01 S(  5) 1.190999E-06 S(  6) 1.073813E-06
S(  7) 3.724297E-02 S(  8) 2.624732E-02 S(  9) 6.436773E-03 S( 72) 1.986454E-01 S(103) 1.429951E-06 S(104) 1.183838E-06
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
2.201E+01
1.628E-03 2.575E-07
9.099E-04-3.623E-08 2.871E-07
2.423E-03 4.714E-03-5.926E-04 3.390E+02
3.605E-07 7.410E-11-2.858E-10 1.069E-06 1.418E-12
2.899E-06 9.944E-11 2.137E-10 1.364E-06 5.413E-13 1.153E-12
1.757E-03 0.        0.        -7.971E-03-1.216E-09 6.232E-10 1.387E-03
-3.540E-03-2.497E-06 0.       -2.949E-02-2.409E-09 8.316E-10-9.353E-04 6.889E-04
-2.633E-02-2.524E-06-1.454E-06-3.890E-02 5.965E-10-2.453E-09 2.520E-05-1.446E-05 4.143E-05
-7.726E-06-2.848E-06-1.273E-01-1.273E-01 5.666E-09 6.045E-09 1.221E-05 2.549E-06-6.375E-05 3.946E-02
3.179E-11-1.637E-10 4.611E-07 4.611E-07 1.270E-12 5.972E-13-1.018E-09-9.085E-10 3.381E-10 2.570E-09 2.045E-12
3.964E-11 1.129E-10-2.673E-07-2.673E-07 6.061E-13 9.907E-13 4.922E-10 2.724E-10-1.092E-09 2.706E-09 6.898E-13 1.401E-12
CORRELATION MATRIX (CUZ)TRANSPOSE
8.084E-02-5.645E-06-1.839E-06-9.194E-02 3.832E-09 4.252E-09 9.956E-07 1.342E-05 1.504E-04-3.691E-04 1.745E-09 1.941E-09
-5.186E-03 2.998E-07 1.574E-07 4.442E-03-3.634E-10-2.062E-10-1.470E-04-9.300E-05-7.630E-06 2.396E-05-2.049E-10-7.027E-11
CORRELATION MATRIX (CVZ)TRANSPOSE
-1.237E-05 6.218E-11 1.529E-10-3.481E-05-2.827E-11-2.235E-11 9.172E-10 1.250E-09-1.059E-09 6.508E-09-2.799E-11-2.210E-11
-2.583E-06-5.934E-10 1.359E-10-2.855E-04 3.331E-12 2.636E-12-3.208E-10 4.447E-09 4.482E-09 5.027E-09 3.382E-12 3.550E-12
```

POINTS	TYPE	TIME	COMP1	COMP2	COMP3	SIG1	SIG2	SIG3
40, 41, 42	1	81.0000	846248.87	.653599	.089080	20.000000	.000030	.000030
43, 44, 45	1	82.0000	858076.10	.661143	.087028	20.000000	.000030	.000030

```
 46,  47,  48   2    82.0000      891202.76      -1.425735        .082296      20.000000        .000030        .000030
 49,  50,  51   2    84.0000      868737.84      -1.409996        .085338      20.000000        .000030        .000030


TIME 8.50000000E+01
V(A) 1.28495774E+04 G(A) 3.94523293E-01 L(A) 6.71947614E+01 ALT  9.19217852E+04 LATD 1.69856431E+01 LON  2.85987537E+01
U(I) 4.98037651E+03 V(I) 1.33110620E+04 W(I)-8.84780272E+01 PSI  8.28770173E+01 THE  1.91879217E+01 PHI  3.17612396E+01
U(B) 1.17127844E+04 V(B)-8.56830532E+02 W(B) 5.21422709E+03 SIG  2.98544615E+01 BET -3.82341193E+00 ALF  2.39973852E+01
XP   0.             YP   0.             ZP   0.             V(T) 1.42125420E+04 XZI -9.33175991E+00 ETA  2.23540342E+01
P   -1.27096643E-03 Q    8.01514722E-03 R    9.96728523E-04 AX  -1.28517050E+02 AY   8.02340547E+00 AZ  -1.88684021E+02
PM  -1.27096643E-03 QM   8.01514722E-03 RM   9.96728523E-04 AXM -1.28517050E+02 AYM  8.02340547E+00 AZM -1.88791271E+02
E0   7.41151859E-01 E1   9.61734110E-02 E2   2.98736819E-01 E3   5.93465171E-01 LATC 1.68779654E+01 A(T) 7.03240675E-06
C 72 1.07250206E-01 C103 2.87979360E-01 C104 5.41052036E-01 C
S(  1) 4.033931E+00 S(  2) 4.975702E-04 S(  3) 5.135661E-04 S(  4) 2.365175E+01 S(  5) 1.540886E-06 S(  6) 1.375152E-06
S(  7) 3.609172E-02 S(  8) 2.641984E-02 S(  9) 3.494534E-03 S( 72) 1.951286E-01 S(103) 1.264449E-06 S(104) 1.060568E-06
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 1.627E+01
 1.306E-03 2.476E-07
 4.446E-04-6.113E-08 2.638E-07
 4.893E+01 9.826E-03-1.947E-03 5.594E+02
 5.446E-07 1.765E-10-5.298E-10 6.200E-06 2.374E-12
 3.721E-06 1.575E-10 3.032E-10 5.974E-06 3.505E-13 1.891E-12
 1.018E-02 0.        0.        7.673E-02 5.283E-09-1.139E-09 1.303E-03
-9.686E-03-4.231E-06 0.       -1.029E-01-7.587E-09 1.909E-09-9.407E-04 6.980E-04
-1.148E-02-1.266E-06-6.538E-07-4.933E-02 6.026E-10-2.306E-09 1.409E-05-8.568E-06 1.221E-05
-1.265E-05-7.433E-06-4.774E-01-4.774E-01 2.564E-08 2.141E-08 3.092E-05-2.810E-05-5.634E-05 3.808E-02
 2.907E-11-1.546E-10 1.025E-06 1.025E-06 1.486E-12 6.216E-13-2.695E-11-9.157E-10 2.172E-10 4.582E-09 1.599E-12
 2.572E-11 5.930E-11 4.655E-07 4.655E-07 6.526E-13 1.117E-12 2.891E-10-6.389E-11-3.637E-10 2.565E-09 7.091E-13 1.125E-12
CORRELATION MATRIX (CUZ)TRANSPOSE
 1.386E-01-8.210E-06-4.622E-06-3.192E-01 1.764E-08 1.551E-08 1.655E-05-1.281E-05 1.527E-04-1.363E-03 3.241E-09 1.904E-09
-9.127E-03 1.076E-07 9.333E-07 1.351E-02-1.589E-09-7.951E-10-1.502E-04-8.754E-05-7.833E-06 8.303E-05-2.778E-10-8.897E-11
CORRELATION MATRIX (CVZ)TRANSPOSE
-7.099E-06 4.107E-10 4.235E-10-4.769E-06-2.968E-11-2.368E-11 3.300E-09-1.113E-09-6.161E-09 5.413E-08-2.841E-11-2.275E-11
 3.617E-06 4.348E-10-1.777E-10-1.535E-04 3.892E-12 3.247E-12 1.861E-08-1.683E-08-3.027E-09-5.273E-08 3.465E-12 3.325E-12

   POINTS      TYPE     TIME         COMP1          COMP2           COMP3          SIG1           SIG2           SIG3
 52,  53,  54   2    86.0000      847097.12      -1.394159        .088795      20.000000        .000030        .000030
 55,  56,  57   2    88.0000      826239.40      -1.378211        .092653      20.000000        .000030        .000030


TIME 9.00000000E+01
V(A) 1.19574508E+04 G(A) 2.09463749E+00 L(A) 6.84176801E+01 ALT  9.33074086E+04 LATD 1.70497522E+01 LON  2.87621416E+01
U(I) 4.39546143E+03 V(I) 1.25775287E+04 W(I)-4.37046954E+02 PSI  8.42001745E+01 THE  2.12640452E+01 PHI  3.17535862E+01
U(B) 1.18746202E+04 V(B)-7.83268880E+02 W(B) 4.91016861E+03 SIG  2.93025729E+01 BET -3.75583042E+00 ALF  2.43003910E+01
XP   0.             YP   0.             ZP   0.             V(T) 1.33306158E+04 XZI -9.06344558E+00 ETA  2.25789365E+01
P   -1.40626441E-03 Q    6.75272832E-03 R    9.56333835E-04 AX  -1.04586147E+02 AY   6.53780171E+00 AZ  -1.55212352E+02
PM  -1.40626441E-03 QM   6.75272832E-03 RM   9.56333835E-04 AXM -1.04586147E+02 AYM  6.53780171E+00 AZM -1.55323524E+02
E0   7.35257124E-01 E1   8.05213741E-02 E2   3.11932621E-01 E3   5.96331543E-01 LATC 1.69417167E+01 A(T) 3.45543794E-05
C 72 1.11171555E-01 C103 2.87979338E-01 C104 5.41051991E-01 C
S(  1) 3.427940E+00 S(  2) 4.405705E-04 S(  3) 4.786468E-04 S(  4) 2.645112E+01 S(  5) 1.873405E-06 S(  6) 1.567973E-06
S(  7) 3.382445E-02 S(  8) 2.574473E-02 S(  9) 2.149743E-03 S( 72) 1.864880E-01 S(103) 1.256231E-06 S(104) 1.053415E-06
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 1.175E+01
 7.711E-04 1.941E-07
-2.908E-05-8.050E-08 2.291E-07
 4.116E+01 1.026E-02-2.961E-03 6.997E+02
 1.359E-06 2.047E-10-6.412E-10 9.267E-06 3.510E-12
 3.654E-06 1.082E-10 2.422E-10 6.748E-06 3.733E-13 2.459E-12
 1.799E-02 0.        0.        2.007E-01 1.147E-08-2.208E-09 1.144E-03
-1.520E-02-5.848E-06 0.       -1.846E-01-1.155E-08 2.235E-09-8.659E-04 6.628E-04
```

```
-4.745E-03-6.142E-07-2.540E-07-3.504E-02 4.085E-10-1.378E-09 5.473E-07 3.086E-07 4.621E-06
-1.584E-05-1.181E-05-7.795E-01-7.795E-01 5.381E-08 3.860E-08 1.832E-05-2.902E-05-5.675E-05 3.478E-02
 1.355E-11-1.512E-10 8.381E-07 8.381E-07 1.814E-12 6.424E-13-1.090E-09 1.734E-10 1.668E-10 7.181E-09 1.578E-12
 1.824E-11 2.130E-11 7.001E-07 7.001E-07 7.260E-13 1.272E-12 6.699E-10-5.276E-10-1.169E-10 9.155E-10 7.171E-13 1.110E-12
CORRELATION MATRIX (CUZ)TRANSPOSE
 1.893E-01-8.256E-06-6.220E-06-4.765E-01 3.629E-08 2.995E-08 1.060E-05-1.587E-05 1.538E-04-3.626E-03 5.135E-09 5.637E-10
-1.261E-02-1.050E-06 3.212E-06-7.048E-03-5.088E-09-7.642E-10-1.396E-04-9.347E-05-7.736E-06 2.243E-04-2.337E-10-7.590E-11
CORRELATION MATRIX (CVZ)TRANSPOSE
-3.462E-07 8.337E-10 3.936E-10 3.483E-05-2.998E-11-2.303E-11 7.758E-09-6.153E-09-5.492E-09 1.741E-07-2.849E-11-2.269E-11
 1.140E-05 2.659E-09-6.078E-10 4.663E-05 5.757E-12 4.779E-12 2.500E-08-2.848E-08-9.757E-09-2.340E-07 3.657E-12 3.338E-12


*****************BEGIN BACKWARDS SMOOTHING***************************************************************************


TIME 9.00000000E+01
V(A) 1.19574508E+04 G(A) 2.09463749E+00 L(A) 6.84176801E+01 ALT  9.33074086E+04 LATD 1.70497522E+01 LON  2.87621416E+01
U(I) 4.39546143E+03 V(I) 1.25775287E+04 W(I)-4.37046954E+02 PSI  8.42001745E+01 THE  2.12640452E+01 PHI  3.17535862E+01
U(B) 1.08746202E+04 V(B)-7.83268880E+02 W(B) 4.91016861E+03 SIG  2.93025729E+01 BET -3.75583042E+00 ALF  2.43003910E+01
XP   0.              YP   0.             ZP   0.             V(T) 1.33306158E+04 XZI -9.06344558E+00 ETA  2.25789365E+01
P   -1.40626441E-03 Q    6.75272832E-03 R    9.56333835E-04 AX  -1.04586147E+02 AY   6.53780171E+00 AZ  -1.55212352E+02
PM  -1.40626441E-03 QM   6.75272832E-03 RM   9.56333835E-04 AXM -1.04586147E+02 AYM  6.53780171E+00 AZM -1.55323524E+02
E0   7.35257124E-01 E1   8.05213741E-02 E2   3.11932621E-01 E3   5.96331543E-01 LATC 1.69417167E+01 A(T) 3.45543794E-05
S(  1) 3.427940E+00 S(  2) 4.405705E-04 S(  3) 4.786468E-04 S(  4) 2.645112E+01 S(  5) 1.873405E-06 S(  6) 1.567973E-06
S(  7) 3.382445E-02 S(  8) 2.574473E-02 S(  9) 2.149743E-03 S( 72) 1.864880E-01 S(103) 1.256231E-06 S(104) 1.053415E-06
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 1.175E+01
 7.711E-04 1.941E-07
-2.908E-05-8.050E-08 2.291E-07
 4.116E+01 1.026E-02-2.961E-03 6.997E+02
 1.359E-06 2.047E-10-6.412E-10 9.267E-06 3.510E-12
 3.654E-06 1.082E-10 2.422E-10 6.748E-06 3.733E-13 2.459E-12
 1.799E-02 0.         0.         2.007E-01 1.147E-08-2.208E-09 1.144E-03
-1.520E-02-5.848E-06 0.        -1.846E-01-1.155E-08 2.235E-09-8.659E-04 6.628E-04
-4.745E-03-6.142E-07-2.540E-07-3.504E-02 4.085E-10-1.378E-09 5.473E-07 3.086E-07 4.621E-06
-1.584E-05-1.181E-05-7.795E-01-7.795E-01 5.381E-08 3.860E-08 1.832E-05-2.902E-05-5.675E-05 3.478E-02
 1.355E-11-1.512E-10 8.381E-07 8.381E-07 1.814E-12 6.424E-13-1.090E-09 1.734E-10 1.668E-10 7.181E-09 1.578E-12
 1.824E-11 2.130E-11 7.001E-07 7.001E-07 7.260E-13 1.272E-12 6.699E-10-5.276E-10-1.169E-10 9.155E-10 7.171E-13 1.110E-12
CORRELATION MATRIX (CUZ)TRANSPOSE
 1.893E-01-8.256E-06-6.220E-06-4.765E-01 3.629E-08 2.995E-08 1.060E-05-1.587E-05 1.538E-04-3.626E-03 5.135E-09 5.637E-10
-1.261E-02-1.050E-06 3.212E-06-7.048E-03-5.088E-09-7.642E-10-1.396E-04-9.347E-05-7.736E-06 2.243E-04-2.337E-10-7.590E-11
CORRELATION MATRIX (CVZ)TRANSPOSE
[illegible] 8.337E-10 3.936E-10 3.483E-05-2.998E-11-2.303E-11 7.758E-09-6.153E-09-5.492E-09 1.741E-07-2.849E-11-2.269E-11
 1.140E-05 2.659E-09-6.078E-10 4.663E-05 5.757E-12 4.779E-12 2.500E-08-2.848E-08-9.757E-09-2.340E-07 3.657E-12 3.338E-12

   POINTS     TYPE   TIME      RES1          RES2          RES3          WGT.RES1      WGT.RES2      WGT.RES3      LOSS FCTN
  57,  56,  55  2     88.000 -1.37444E+00  3.15725E-06  8.48743E-06 -6.87221E-02  1.05242E-01  2.82914E-01  9.58391E-02
  54,  53,  52  2     86.000 -9.38659E-01 -1.61044E-06  7.76388E-07 -4.69329E-02 -5.36814E-02  2.58796E-02  1.01593E-01


TIME 8.50000000E+01
V(A) 1.28475913E+04 G(A) 3.77934747E-01 L(A) 6.71955546E+01 ALT  9.19084577E+04 LATD 1.69856348E+01 LON  2.85987338E+01
U(I) 4.97945246E+03 V(I) 1.33093224E+04 W(I)-8.47447400E+01 PSI  8.26846784E+01 THE  1.94174709E+01 PHI  3.15896660E+01
U(B) 1.17046872E+04 V(B)-8.08350670E+02 W(B) 5.23521426E+03 SIG  2.96442413E+01 BET -3.60734503E+00 ALF  2.40977755E+01
XP   0.              YP   0.             ZP   0.             V(T) 1.42105661E+04 XZI -8.77751807E+00 ETA  2.24070859E+01
P   -1.27096643E-03 Q    8.01514722E-03 R    9.96728523E-04 AX  -1.28517050E+02 AY   8.02340547E+00 AZ  -1.88680099E+02
PM  -1.27096643E-03 QM   8.01514722E-03 RM   9.96728523E-04 AXM -1.28517050E+02 AYM  8.02340547E+00 AZM -1.88791271E+02
```

```
E0     7.42403300E-01 E1    9.42395292E-02 E2    2.99054229E-01 E3    5.92049676E-01 LATC 1.68779572E+01 A(T) 2.16550637E-05
S(  1) 2.326724E+00 S(  2) 2.447078E-04 S(  3) 2.529354E-04 S(  4) 1.130819E+01 S(  5) 1.237032E-06 S(  6) 1.065748E-06
S(  7) 3.467664E-02 S(  8) 2.432409E-02 S(  9) 2.058828E-03 S( 72) 1.864880E-01 S(103) 1.256231E-06 S(104) 1.053415E-06
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 5.414E+00
 2.533E-04 5.988E-08
 2.360E-05-9.940E-09 6.398E-08
 2.384E+00 9.990E-04-1.442E-05 1.279E+02
 5.888E-07 1.478E-11-1.411E-10 1.568E-07 1.530E-12
 1.245E-06 1.926E-11 4.326E-11 1.005E-07 6.853E-13 1.136E-12
 5.209E-03 0.        0.       -4.012E-02-2.885E-09 1.258E-09 1.202E-03
-4.672E-03-1.193E-06 0.        2.475E-02 1.206E-09-8.312E-10-8.390E-04 5.917E-04
-2.741E-03-3.256E-07-1.604E-07-4.258E-03 1.432E-10-2.549E-10 1.285E-05-8.387E-06 4.239E-06
-7.987E-06-6.523E-06-7.726E-02-7.726E-02 1.198E-08 4.963E-09 1.948E-05-2.793E-05-6.594E-05 3.478E-02
 9.490E-12-1.117E-10 6.216E-08 6.216E-08 1.413E-12 6.868E-13-1.098E-09 1.739E-10 1.446E-10 7.181E-09 1.578E-12
 9.248E-12 2.408E-11-2.497E-07-2.497E-07 7.023E-13 1.024E-12 6.884E-10-4.987E-10-1.043E-10 9.155E-10 7.171E-13 1.110E-12
CORRELATION MATRIX (CUZ)TRANSPOSE
 [illegible]-5.535E-06-4.423E-06-6.045E-02 8.724E-09 3.070E-09 9.987E-06-1.469E-05 1.506E-04-3.626E-03 5.135E-09 5.637E-10
-6.422E-03 2.390E-07 4.036E-07 1.041E-02 4.246E-12-4.196E-10-1.364E-04-1.012E-04-7.499E-06 2.243E-04-2.337E-10-7.590E-11
CORRELATION MATRIX (CVZ)TRANSPOSE
-3.601E-06 5.135E-10 2.860E-10-6.673E-06-2.885E-11-2.291E-11 8.006E-09-5.849E-09-5.082E-09 1.741E-07-2.849E-11-2.269E-11
 6.629E-06 1.584E-09-2.599E-10-8.533E-05 3.831E-12 3.318E-12 2.602E-08-2.683E-08-8.696E-09-2.340E-07 3.657E-12 3.338E-12


   POINTS      TYPE    TIME      RES1          RES2          RES3          WGT.RES1      WGT.RES2      WGT.RES3      LOSS FCTN
 51,  50,  49  2      84.000 -5.12041E-02 -3.22529E-06 -3.25728E-06 -2.56020E-03 -1.07510E-01 -1.08576E-01  1.24917E-01
 48,  47,  46  2      82.000  7.61788E-01 -2.61999E-06 -4.37501E-06  3.80894E-02 -8.73331E-02 -1.45834E-01  1.55292E-01
 45,  44,  43  1      82.000 -3.85934E+00 -1.43647E-06 -3.85684E-06 -1.92967E-01 -4.78823E-02 -1.28561E-01  2.11349E-01
 42,  41,  40  1      81.000 -3.31897E+00 -2.07174E-06 -3.72869E-06 -1.65948E-01 -6.90580E-02 -1.24290E-01  2.59105E-01
 39,  38,  37  2      80.000  9.56996E-01 -8.31592E-07 -3.42725E-06  4.78498E-02 -2.77197E-02 -1.14242E-01  2.75214E-01
 36,  35,  34  1      80.000 -2.45734E+00 -2.55421E-06 -3.23101E-06 -1.22867E-01 -8.51404E-02 -1.07700E-01  3.09159E-01


TIME 8.00000000E+01
V(A) 1.30834262E+04 G(A)-1.57588294E+00 L(A) 6.58493837E+01 ALT  9.25596200E+04 LATD 1.69126813E+01 LON  2.84242515E+01
U(I) 5.67807109E+03 V(I) 1.41303178E+04 W(I) 3.81806415E+02 PSI  8.10394337E+01 THE  1.73383116E+01 PHI  3.13849178E+01
U(B) 1.26709482E+04 V(B)-8.35880085E+02 W(B) 5.61229911E+03 SIG  2.99915750E+01 BET -3.45169713E+00 ALF  2.38898236E+01
XP   0.             YP   0.             ZP   0.             V(I) 1.52332580E+04 XZI -8.47120207E+00 ETA  2.22300050E+01
P   -1.09490564E-03 Q    8.46549342E-03 R    1.01728525E-03 AX  -1.44364809E+02 AY   9.00133960E+00 AZ  -2.09637384E+02
PM  -1.09490564E-03 QM   8.46549342E-03 RM   1.01728525E-03 AXM -1.44364809E+02 AYM  9.00133960E+00 AZM -2.09748555E+02
E0   7.49974636E-01 E1   1.08980567E-01 E2   2.84032259E-01 E3   5.87355904E-01 LATC 1.68054115E+01 A(T) 8.11571593E-06
S(  1) 1.503867E+00 S(  2) 1.305393E-04 S(  3) 1.434849E-04 S(  4) 1.209780E+01 S(  5) 1.019882E-06 S(  6) 8.614057E-07
S(  7) 3.557583E-02 S(  8) 2.284125E-02 S(  9) 2.011421E-03 S( 72) 1.864830E-01 S(103) 1.256231E-06 S(104) 1.053415E-06
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 2.262E+00
 2.369E-05 1.704E-08
-2.805E-06-3.960E-09 2.059E-08
-2.669E+00-6.170E-04 5.402E-05 1.464E+02
 3.569E-07-3.912E-12-2.625E-11 3.969E-07 1.040E-12
 4.683E-07-3.580E-12 5.696E-12 4.387E-07 6.518E-13 7.420E-13
-3.752E-04 0.        0.       -6.329E-03-5.088E-10-2.660E-10 1.266E-03
-2.926E-04 1.029E-06 0.        1.295E-02 4.525E-10 3.064E-11-8.085E-04 5.217E-04
-8.068E-04-1.239E-07 9.880E-09 9.619E-03 6.268E-11 2.548E-10 2.377E-05-1.473E-05 4.046E-06
-2.021E-06-2.341E-06 2.485E-01 2.485E-01-9.286E-09-1.033E-08 2.064E-05-2.674E-05-7.296E-05 3.478E-02
 2.919E-12-6.977E-11-3.349E-07-3.349E-07 1.099E-12 6.861E-13-1.107E-09 1.724E-10 1.235E-10 7.181E-09 1.578E-12
 1.725E-12 2.469E-11-5.575E-07-5.575E-07 6.897E-13 7.880E-13 7.079E-10-4.684E-10-9.245E-11 9.155E-10 7.171E-13 1.110E-12
CORRELATION MATRIX (CUZ)TRANSPOSE
 1.494E-02-1.934E-06-2.032E-06 1.911E-01-5.844E-09-5.129E-09 9.275E-06-1.354E-05 1.463E-04-3.626E-03 5.135E-09 5.637E-10
-1.371E-03 2.231E-07-1.573E-07-1.032E-02 3.218E-10 4.644E-10-1.324E-04-1.067E-04-7.189E-06 2.243E-04-2.337E-10-7.590E-11
CORRELATION MATRIX (CVZ)TRANSPOSE
-7.117E-06 1.969E-10 1.448E-10-3.044E-05-2.771E-11-2.196E-11 8.274E-09-5.523E-09-4.666E-09 1.741E-07-2.849E-11-2.269E-11
```

```
1.526E-06 5.743E-10 1.846E-12-1.561E-04 3.084E-12 2.495E-12 2.710E-08-2.513E-08-7.683E-09-2.340E-07 3.657E-12 3.338E-12

  POINTS      TYPE   TIME      RES1         RES2         RES3         WGT.RES1     WGT.RES2     WGT.RES3     LOSS FCTN
 33.  32.  31  1     79.000 -1.35924E+00 -2.74801E-06 -2.45315E-06 -6.79622E-02 -9.16004E-02 -8.17716E-02  3.28855E-01
 30.  29.  28  2     78.000  1.35550E-02  1.06657E-06 -1.30329E-06  6.77751E-04  3.55524E-02 -4.34429E-02  3.32007E-01
 27.  26.  25  1     78.000 -1.15783E-01 -2.51061E-06 -1.49191E-06 -5.78913E-03 -8.36871E-02 -4.97303E-02  3.41517E-01
 24.  23.  22  1     77.000  1.17701E+00 -1.69556E-06 -4.49689E-07  5.88505E-02 -5.65187E-02 -1.49896E-02  3.48399E-01
 21.  20.  19  1     76.000  2.41976E+00 -1.55493E-07  5.67658E-07  1.20988E-01 -5.18309E-03  1.89219E-02  3.63422E-01
 18.  17.  16  1     75.000  3.51146E+00  2.25469E-06  1.45315E-06  1.75573E-01  7.51564E-02  4.84383E-02  4.02243E-01

TIME 7.50000000E+01
V(A) 1.49641593E+04 G(A)-3.50509731E+00 L(A) 6.45148380E+01 ALT  9.57341473E+04 LATD 1.68297096E+01 LON  2.82380534E+01
U(I) 6.42669410E+03 V(I) 1.49505475E+04 W(I) 9.14868865E+02 PSI  7.94380945E+01 THE  1.52710090E+01 PHI  3.11967081E+01
U(B) 1.36818354E+04 V(B)-8.64092941E+02 W(B) 5.99889883E+03 SIG  3.03316550E+01 BET -3.31033857E+00 ALF  2.36753928E+01
XP   0.             YP   0.             ZP   0.             V(T) 1.62990261E+04 XZI -8.19661671E+00 ETA  2.20490252E+01
P   -9.42614978E-04 Q    7.73459373E-03 R    9.82154065E-04 AX  -1.42491786E+02 AY   8.87341895E+00 AZ  -2.04628262E+02
PM  -9.42614978E-04 QM   7.73459373E-03 RM   9.82154065E-04 AXM -1.42491786E+02 AYM  8.87341895E+00 AZM -2.04739434E+02
E0   7.57120270E-01 E1   1.23214203E-01 E2   2.68743155E-01 E3   5.82549803E-01 LATC 1.67229044E+01 A(T) 5.56835862E-06
S(  1) 1.630148E+00 S(  2) 1.360471E-04 S(  3) 1.493390E-04 S(  4) 1.528436E+01 S(  5) 9.832247E-07 S(  6) 7.644820E-07
S(  7) 3.640269E-02 S(  8) 2.149621E-02 S(  9) 2.000494E-03 S( 72) 1.864880E-01 S(103) 1.256231E-06 S(104) 1.053415E-06
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 2.657E+00
 2.585E-05 1.851E-08
 4.225E-05-3.703E-09 2.230E-08
 2.770E+00-6.414E-04 1.998E-04 2.336E+02
 3.008E-07-6.374E-12 2.161E-11 1.349E-06 9.667E-13
 1.675E-07 9.013E-12-7.549E-12 1.443E-07 5.513E-13 5.844E-13
 3.221E-03 0.        0.        1.276E-01 9.515E-09-4.978E-09 1.325E-03
-2.020E-03 9.360E-07 0.       -6.806E-02-5.465E-09 2.875E-09-7.787E-04 4.621E-04
 1.021E-03 6.536E-08 1.226E-07 1.131E-02 3.110E-10 8.321E-11 3.315E-05-1.911E-05 4.002E-06
 2.732E-06 9.941E-07 2.126E-01 2.126E-01-1.015E-08-7.262E-09 2.163E-05-2.567E-05-7.851E-05 3.478E-02
-5.428E-12-2.958E-11-1.198E-07-1.198E-07 8.896E-13 6.308E-13-1.117E-09 1.704E-10 1.049E-10 7.181E-09 1.578E-12
-3.572E-12 2.299E-11-2.939E-07-2.939E-07 6.833E-13 5.681E-13 7.260E-10-4.409E-10-8.199E-11 9.155E-10 7.171E-13 1.110E-12
CORRELATION MATRIX (CUZ)TRANSPOSE
-6.479E-02 1.459E-06 2.200E-07 1.955E-01-5.997E-09 1.673E-09 8.471E-06-1.260E-05 1.423E-04-3.626E-03 5.135E-09 5.637E-10
 3.214E-03-2.157E-07 2.667E-08-9.467E-03 6.524E-11 3.389E-10-1.283E-04-1.112E-04-6.920E-06 2.243E-04-2.337E-10-7.590E-11
CORRELATION MATRIX (CVZ)TRANSPOSE
-1.054E-05-7.400E-11-1.270E-11-3.639E-05-2.663E-11-2.005E-11 8.531E-09-5.223E-09-4.287E-09 1.741E-07-2.849E-11-2.269E-11
-3.[illegible]E-06-2.[illegible]E-10 1.412E-10-1.659E-04 3.419E-12 2.534E-12 2.809E-08-2.358E-08-6.789E-09-2.340E-07 3.657E-12 3.338E-12

  POINTS      TYPE   TIME      RES1         RES2         RES3         WGT.RES1     WGT.RES2     WGT.RES3     LOSS FCTN
 15.  14.  13  1     74.000  4.35127E+00  5.67465E-06  2.10114E-06  2.17563E-01  1.89155E-01  7.00379E-02  4.90262E-01
 12.  11.  10  1     73.000  4.84013E+00  1.02356E-05  2.40975E-06  2.42006E-01  3.41186E-01  8.03250E-02  6.71689E-01
  9.   8.   7  1     72.000  4.88237E+00  1.60575E-05  2.28321E-06  2.44118E-01  5.35251E-01  7.61072E-02  1.02357E+00
  6.   5.   4  1     71.000  4.38696E+00  2.32471E-05  1.63398E-06  2.19348E-01  7.74903E-01  5.44660E-02  1.67512E+00
  3.   2.   1  1     70.000  3.26856E+00  3.18956E-05  3.84540E-07  1.63428E-01  1.06319E+00  1.28180E-02  2.83236E+00

TIME 7.00000000E+01
V(A) 1.59437078E+04 G(A)-5.12693675E+00 L(A) 6.33489928E+01 ALT  1.01550176E+05 LATD 1.67367329E+01 LON  2.80409860E+01
U(I) 7.12301619E+03 V(I) 1.56616007E+04 W(I) 1.42477056E+03 PSI  7.80639211E+01 THE  1.34842323E+01 PHI  3.10808113E+01
U(B) 1.46034724E+04 V(B)-8.90445016E+02 W(B) 6.33620712E+03 SIG  3.06531017E+01 BET -3.20159537E+00 ALF  2.34552310E+01
XP   0.             YP   0.             ZP   0.             V(T) 1.72642135E+04 XZI -7.99954935E+00 ETA  2.18663951E+01
P   -8.75453186E-04 Q    6.04491552E-03 R    8.82155972E-04 AX  -1.21200856E+02 AY   7.53823858E+00 AZ  -1.72094754E+02
PM  -8.75453186E-04 QM   6.04491552E-03 RM   8.82155972E-04 AXM -1.21200856E+02 AYM  7.53823858E+00 AZM -1.72205926E+02
E0   7.63029219E-01 E1   1.35447313E-01 E2   2.55417835E-01 E3   5.78102211E-01 LATC 1.66304495E+01 A(T) 1.60671336E-05
S(  1) 2.494760E+00 S(  2) 2.200619E-04 S(  3) 2.084220E-04 S(  4) 2.084416E+01 S(  5) 1.127610E-06 S(  6) 9.169253E-07
```

```
S(  7) 3.705445E-02 S(  8) 2.045858E-02 S(  9) 2.027214E-03 S( 72) 1.864880E-01 S(103) 1.256231E-06 S(104) 1.053415E-06
COVARIANCE MATRIX P (LOWER TRIANGLE ONLY)
 6.224E+00
 2.052E-04 4.843E-08
 1.246E-04-3.287E-09 4.344E-08
-1.392E+00-2.487E-03 1.391E-04 4.345E+02
 3.009E-07-2.478E-12 1.023E-10 2.265E-06 1.272E-12
-1.070E-06-3.376E-11-6.918E-11 1.880E-08 3.152E-13 8.408E-13
 1.430E-02 0.        0.        1.474E-01 1.537E-08-1.021E-08 1.373E-03
-7.694E-03-5.232E-07 0.       -8.602E-02-8.993E-09 5.845E-09-7.545E-04 4.186E-04
 2.652E-03 2.683E-07 1.240E-07-1.635E-04 6.000E-10-6.374E-10 4.139E-05-2.244E-05 4.110E-06
 7.224E-06 3.973E-06-2.161E-01-2.161E-01 9.337E-09 1.237E-08 2.228E-05-2.489E-05-8.364E-05 3.478E-02
-1.392E-11 4.267E-12 8.624E-07 8.624E-07 7.893E-13 5.187E-13-1.126E-09 1.696E-10 8.951E-11 7.181E-09 1.578E-12
-6.656E-12 1.994E-11 3.973E-07 3.973E-07 6.768E-13 3.662E-13 7.405E-10-4.197E-10-7.313E-11 9.155E-10 7.171E-13 1.110E-12
CORRELATION MATRIX (CUZ)TRANSPOSE
-1.552E-01 3.648E-06 1.631E-06-5.451E-02 8.401E-09 2.337E-08 7.690E-06-1.192E-05 1.397E-04-3.626E-03 5.135E-09 5.637E-10
 8.265E-03-6.898E-08-6.970E-07 1.013E-02-1.180E-09-5.081E-10-1.247E-04-1.158E-04-6.755E-06 2.243E-04-2.337E-10-7.590E-11
CORRELATION MATRIX (CVZ)TRANSPOSE
-1.349E-05-2.700E-10-1.583E-10-2.713E-05-2.564E-11-1.714E-11 8.744E-09-4.992E-09-3.983E-09 1.741E-07-2.849E-11-2.269E-11
-7.098E-06-8.871E-10 1.734E-10-1.217E-04 4.577E-12 3.561E-12 2.888E-08-2.238E-08-6.055E-09-2.340E-07 3.657E-12 3.338E-12
```

III. PROGRAM CHECKOUT AND VALIDATION

STEP checkout consisted of four separate phases -- checking the nonlinear differential equations of motion; checking the linear equations of motion; checking the measurement equations; and checking the overall system operation. Before preceding with the checkout, a check trajectory was synthesized and PQR and FIT tapes generated using a separate trajectory program. In the following discussion, the check trajectory is described and the results of the various checkout phases discussed.

A. Check Trajectory

The check trajectory corresponded to the flight of a maneuverable lifting reentry vehicle, having a hypersonic L/D of approximately 0.6. The vehicle had the following physical characteristics:

Mass versus time (linear)

m, slugs	25.5763	24.8013	24.0262
t, sec	0.0	200.0	400.0

Reference area, $S = 5.0\ ft^2$.

The aerodynamic characteristics correspond to an axisymmetric vehicle where

$$C_{N_\eta} = N_1(M)\ \sin\ [\eta + N_2(M)] + N_3(M)$$

$$C_{Y_\eta} = 0$$

$$C_A = A_1(M)\ \cos\ [\eta + A_2(M)] + A_3(M)$$

The coefficients $N_1(M)$, $N_2(M)$, $N_3(M)$, $A_1(M)$, $A_2(M)$, and $A_3(M)$ are characterized by the following second-degree polynomials:

$$N_1(M) = \begin{cases} 1.5900 + 0.0300M - 0.4200M^2 & (0 \le M \le .8) \\ 2.5879 - 2.4649M + 1.1393M^2 & (0.8 < M < 1.2) \\ 0.9058 + 0.3386M - 0.0288M^2 & (1.2 \le M \le 5.874) \end{cases}$$

$$N_2(M) = \begin{cases} -71.500 - 2.500M + 27.000M^2 & (0 \le M \le 0.8) \\ -132.98 + 151.21M - 69.068M^2 & (0.8 < M < 1.2) \\ -31.285 - 18.289M + 1.5561M^2 & (1.2 \le M \le 5.877) \end{cases}$$

$$N_3(M) = \begin{cases} 1.5100 + 0.0100M - 0.5800M^2 & (0 \le M \le 0.8) \\ 2.8984 - 3.4611M + 1.5894M^2 & (0.8 < M < 1.2) \\ 0.5577 + 0.4401M - 0.0361M^2 & (1.2 \le M \le 6.101) \end{cases}$$

$$A_1(M) = \begin{cases} 0.3450 + 0.1250M - 0.4700M^2 & (0 \le M \le 0.8) \\ 1.2228 - 2.0695M + 0.9016M^2 & (0.8 < M < 1.2) \\ -0.0885 + 0.1161M - 0.0091M^2 & (1.2 \le M \le 6.368) \end{cases}$$

$$A_2(M) = \begin{cases} -50.000 + 22.000M + 28.000M^2 & (0 \le M \le 0.8) \\ -143.64 + 256.11M - 118.32M^2 & (0.8 < M < 1.2) \\ 30.823 - 34.668M + 2.8385M^2 & (1.2 \le M \le 6.107) \end{cases}$$

$$A_3(M) = \begin{cases} -0.1300 + 0.0100M + 0.3200M^2 & (0 \le M \le 0.8) \\ -0.8065 + 1.7013M - 0.7371M^2 & (0.8 < M < 1.2) \\ 0.2637 - 0.0823M + 0.0061M^2 & (1.2 \le M \le 6.722) \end{cases}$$

Above the upper Mach number limit in the above inequalities the coefficients are constant. C_A and C_{N_η} are then transformed to C_A, C_Y, and C_N or C_D, C_Y, and C_L versus α and β. A plot of C_L, C_D, and C_Y is presented in figure 2 for Mach number 0., 1.0, 2.5, and 7.0.

The check trajectory has the following initial conditions at $t = 0$.

$$V_R = 18\,000.0 \text{ fps}, \quad \gamma_R = -6.0°, \quad \lambda_R = 30.0°$$

$$h_o = 250\,000.0 \text{ ft}, \quad \varphi_D = 15.0°, \quad \theta = 25.0°$$

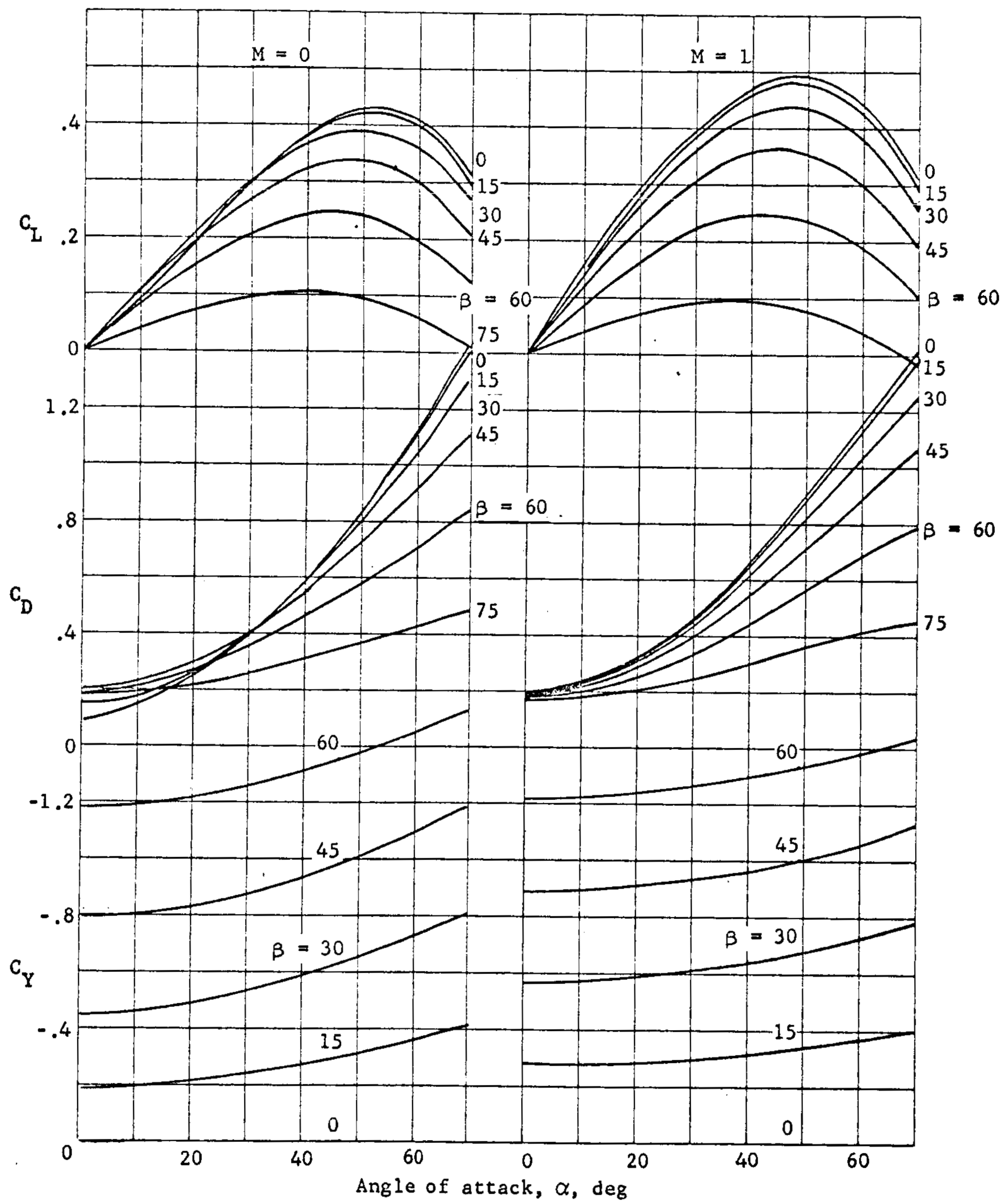


Figure 2.- Aerodynamic Coefficients of Checkout Vehicle

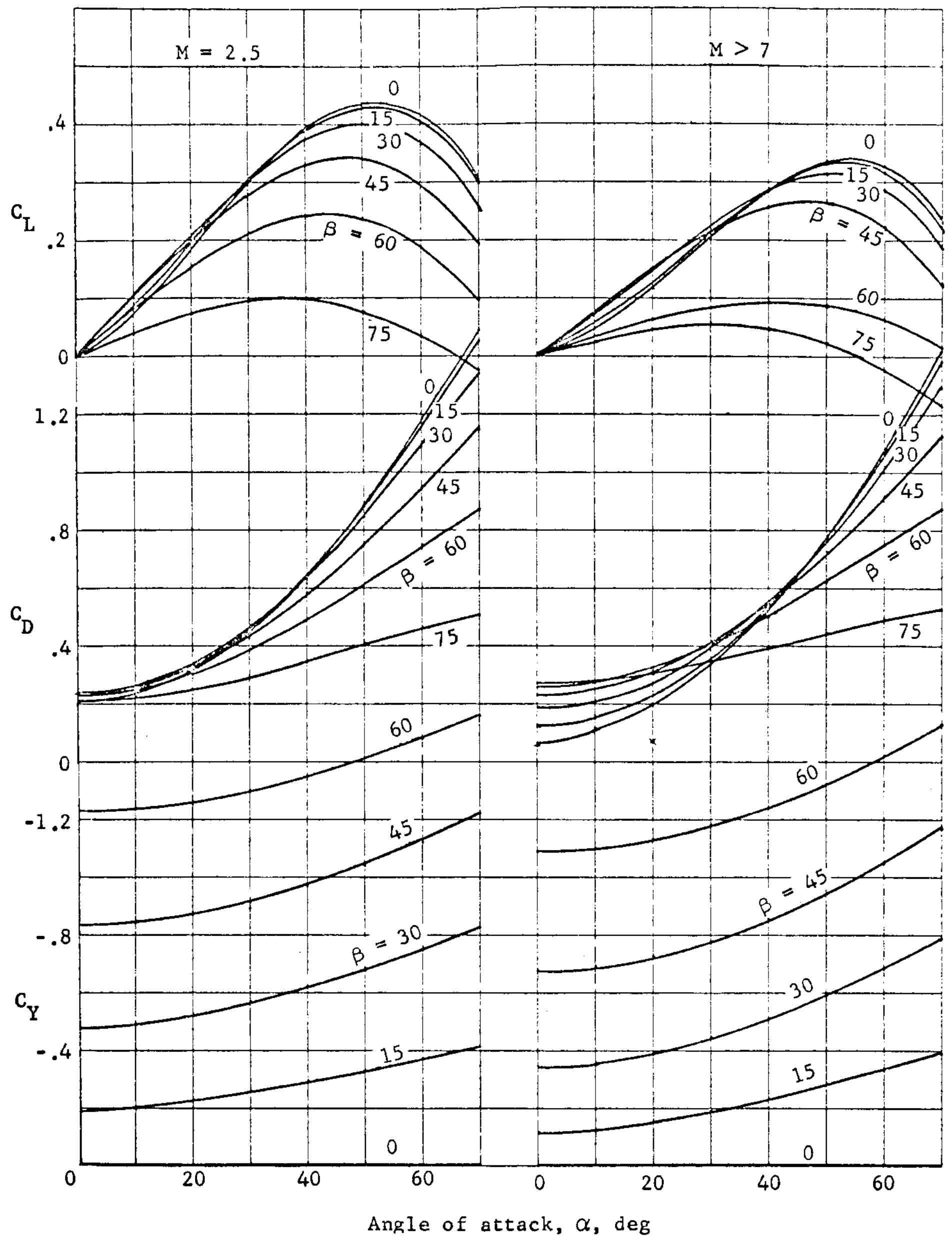
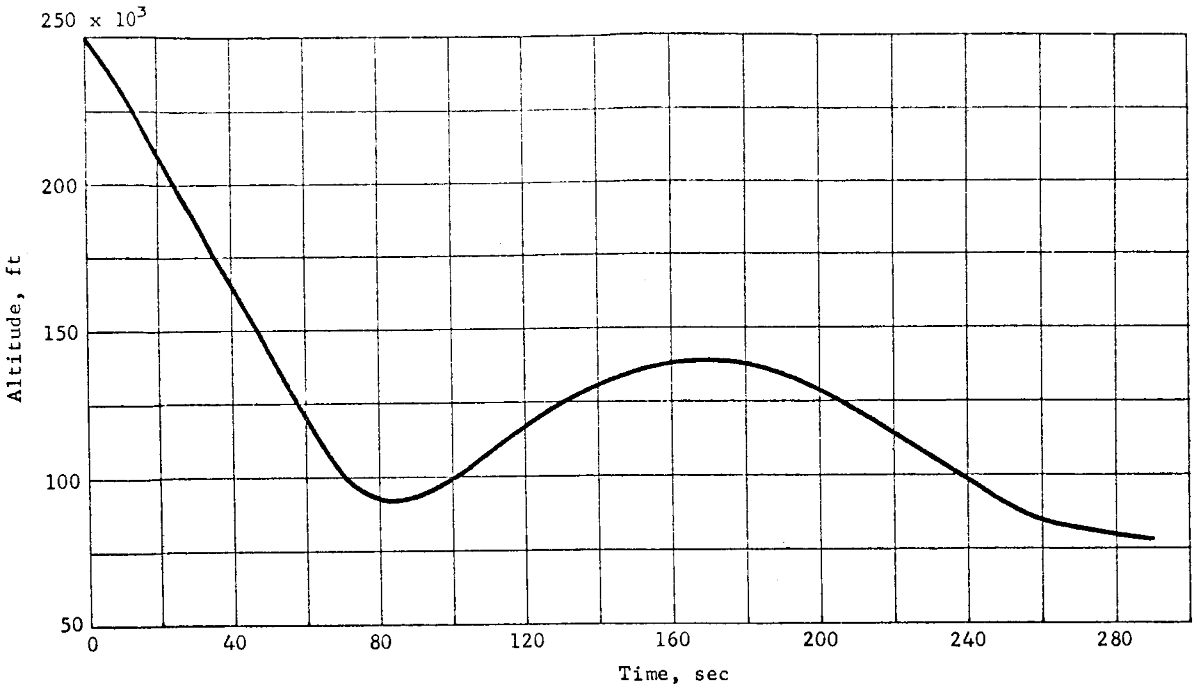


Figure 2.- Concluded

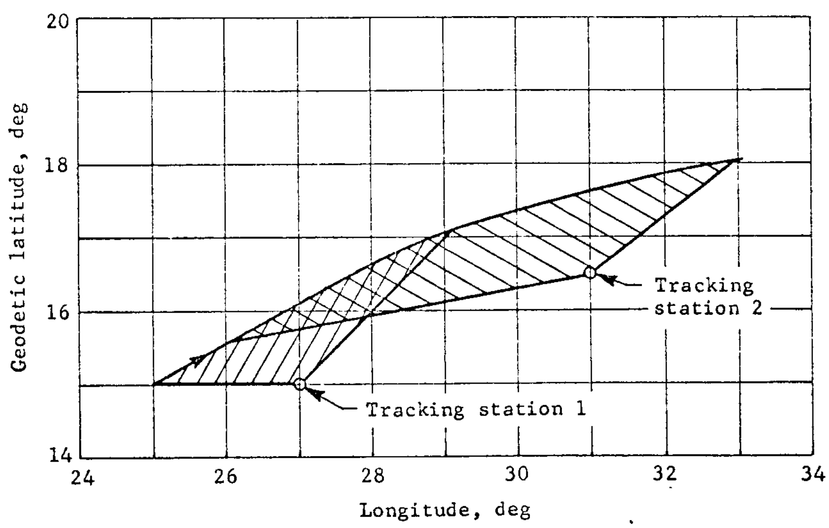
The angle of attack, α, sideslip angle, β, and roll angle, σ, varied linearly with time as follows:

t, sec	0	200.0	400.0
α, deg	20.0	30.0	40.0
β, deg	-1.0	-1.0	-1.0
σ, deg	30.0	20.0	10.0

The altitude versus time and latitude versus longitude are presented in figures 3(a) and 3(b). The PQR tape generated for the check trajectory contained data every 0.25 sec. Tracking stations were located at latitudes and longitudes of 15.0, 27.0 and 16.5, 31.0. The tracking spans, during which time the evaluation angle is larger than 4° is shown in figure 3(b). The tracking data were stored on the FIT tape at a rate of two points per second.



(a) Altitude vs Time



(b) Latitude vs Longitude

Figure 3.- Checkout Trajectory

B. Checkout of Nonlinear Equations of Motion

To check the nonlinear equations of motion, the inertial angular rates (and accelerometers for STEP2) were integrated from t = 0 to t = 250.0 sec with a computing interval of 0.5 sec. The state time history was then compared with the known check trajectory and agreement to at least six significant digits was obtained. This accuracy is, no doubt, limited by the 0.25-sec interval of the PQR tape data. Several 50-sec problems were solved using different computing intervals for the numerical integration. These results are tabulated below.

	V	h	α	β
Check	17 791.5507	141 872.977	22.5000000	-1.00000000
STEP2 (Δt = 0.25)	17 791.5337	141 873.085	22.4999995	-1.00000006
STEP2 (Δt = 0.50)	17 791.5515	141 872.975	22.5000000	-1.00000001
STEP2 (Δt = 5.0)	17 791.4553	141 873.334	22.5000023	- .99999864

A rather interesting result is exhibited in the above comparison. As the computing interval gets large, errors are introduced as a result of truncation. As the computing interval is reduced, errors are reduced until the computing interval becomes smaller than twice the PQR tape data interval (0.25 sec for this check problem). At this point, the PQR tape requires interpolation. The linear interpolation introduces errors because the data are not linear. Normally, a computing interval twice the PQR tape data interval is most satisfactory because each PQR tape data point is then used in the fourth-order Runge Kutta integration and interpolation is not required. For very nonlinear problems where the threshold for truncation error forces the computing interval to be smaller than the PQR tape data interval, one must either live with the error or increase the frequency at which the PQR tape data are recorded.

C. Checkout of Linear Equations of Motion

The checkout of the linear equations of motion is a very cumbersome task. The solution to the linear equations of motion is the state transition matrix for the expanded state vector. Because any of the model parameters can potentially be components of the expanded state vector, it is necessary to check a vector solution for each state and model parameter or equivalently, a 9x160 matrix.

The elements of the state transition matrix are partial derivatives of the expanded state vector, Z, at time t_1 with respect to Z components at t_o. Therefore, the elements can be checked by approximating the partial derivatives by finite differences. This is done by generating a reference nonlinear solution between t_o and t_1. A perturbed nonlinear solution is next calculated wherein Z_i, the i^{th} component of Z, is perturbed from the nominal at t_o. The response of the state variables at t_1 divided by the perturbation in Z_i, which caused them provides a one-sided finite difference approximation to the partial derivative

$$\frac{\Delta Z(t_1)}{\Delta Z_i(t_o)} \cong \frac{\partial Z(t_i)}{\partial Z_i(t_o)}$$

The programs were checked by calculating two-sided finite difference approximations to the state transition matrix. The variable Z_i was perturbed first by $+\Delta Z_i$ then by $-\Delta Z_i$. The difference between the + and - responses in Z at t_1, divided by $2\Delta Z_i$ yield the desired finite difference approximation.

All state variables and model parameters were perturbed in this manner. A checkout subroutine was mechanized into the STEP decks to do the bookkeeping. It controlled the variable perturbations, recorded the responses, performed the differencing, and printed the finite difference approximation along with its analytically calculated counterpart. The finite difference approximation required only that the nonlinear equations of motion be correct. Agreement with the state transition matrix elements calculated by integrating the linear equations of motion assured that the linear equations of motion were formulated correctly and were being solved correctly. In all cases, the finite difference approximation agreed with its analytical counterpart to at least six significant digits. When disagreement occurred during checkout, it was difficult to determine which coefficient in the linear differential equations was the cause. Therefore, finite difference approximations of the F matrix, equation (6a), were simultaneously calculated by the check subroutines. Comparison of the finite difference approximation and its analytical counterpart immediately disclosed the erroneous coefficient.

D. Checkout of Measurement Equations

Checking the nonlinear measurement equations was accomplished by comparison with solutions known to be correct. The coefficients in the linear measurement equations, G in equation (4), were checked by comparing the analytically obtained partial derivatives with a finite difference approximation. This comparison study was performed simultaneously with the state transition matrix checkout and was automated by a checkout subroutine discussed previously.

E. Checkout of Input/Output Transformations

The transformations presented in Section VI of Volume I for transforming the input and output were checked by the finite difference approximation comparison technique. Agreement to better than six significant digits were obtained.

F. Checkout Complete STEP

Having assured that the nonlinear and linear state and measurement equations were being solved correctly, the matrix propagation and minimum variance estimation logic was next checked.

1. Matrix propagation.- Two 20-sec problems were solved using the error analysis operating mode. In the first problem, five model parameters were included in the expanded state vector, thus yielding a 15x15 covariance matrix P. In the second problem, three of the model parameters were removed from the expanded state vector Z and placed in the uncertain model parameter vector U. Thus, the second problem has a 12x12 covariance matrix P and a 12x3 correlation matrix CVZ. As indicated in equation (50) of Volume I, CUZ from the second problem should be a submatrix in the 15x15 covariance matrix, P, of the first problem, as indeed it was. Thus, this check confirmed that the covariance and correlation matrices P and CUZ were being propagated correctly.

The next check runs used the filtering mode of operation to test the minimum variance recursive updating equations, equations (53). In the first problem several tracking points from station 1 were processed with infinite variances. As expected, no change in the state variables occurred. In the second problem, tracking data from station 1 were processed with zero variances. The initial covariance matrix of state P was specified to be very large. While attempting to process the ninth tracking point, a variance element in P became negative causing the run to terminate. Because the dynamic model in STEP is a ninth-order system (nine independent first-order differential equations), after processing nine data points of absolute certainty (zero variance), the covariance matrix P should be zero if it is initially specified to be infinity. In the check problem, a finite initial covariance matrix P was specified. Therefore, processing the ninth tracking point produces the expected result of causing a variance of P to become slightly negative in attempting to seek zero.

In the next check runs, the smoothing option was tested. Operating in the error analysis mode with no tracking data, the covariance matrix was input using input/output option 1 and 2. The covariance matrix P was transformed to internal units, propagated forward in time for 20 sec, propagated back to intial time, and transformed back to the output variable of option 1 and 2. The state and covariance matrix at the completion of the run matched those inputted to better than eight significant digits.

The final series of tests checked the filtering mode of operation. Twenty-second problems were run, during which tracking data from stations 1 and 2 were discretely selected from the FIT tape and processed. The problems ran for as many as three iterations using both the updated and nonupdated references, and both vector and scalar data processing options. Typical of the convergence of these runs is the two iteration problems presented in table 2 for STEP2. The expanded state vector was randomly noised at initial time by an amount consistent with the initial covariance. The random noise was normally distributed with zero mean. Table 2 shows the correct state at initial time, the noised state at initial time and the converged state at initial time after two iterations. Also shown are the estimated standard deviations at initial time after the second iteration. Comparing these residuals with their calculated variances 83% lie within one-sigma and 100% within two-sigma. Theoretically, 68.3% should be within one-sigma, 95.5% within two sigma, and 99.7% within three sigma.

TABLE 2.- THREE ITERATION PROBLEMS, STEP2

Iter	V	γ	λ	h	φ	θ	σ	β	α
Nom	15 946.18	-5.13088	63.3697	101 549.4	16.736757	28.040927	26.50	-1.0000	23.5000
0	16 001.20	-6.75088	63.0352	100 584.0	16.758397	28.041838	29.8270	-4.4320	19.3930
1	15 943.70	-5.12694	63.3490	101 550.2	16.736733	28.040986	30.6531	-3.2016	23.4552
2	15 945.99	-5.12227	63.3601	101 547.5	16.736749	28.040935	30.2796	-3.0292	23.5244
Iter	**δV**	**$\delta\gamma$**	**$\delta\lambda$**	**δh**	**$\delta\varphi$**	**$\delta\theta$**	**$\delta\sigma$**	**$\delta\beta$**	**$\delta\alpha$**
0	55.02	-1.6200	-.3345	-965.4	.02164	.00091	3.327	-3.432	-4.107
1	-2.48	.0039	-.0207	- .8	.00002	.00005	4.153	-2.202	- .045
2	- .19	.0086	-.0096	- 1.9	.00001	.00000	3.779	-2.029	.024
Est. σ	(2.52)	(.0135)	(.0128)	(20.59)	(.00006)	(.00005)	(2.254)	(1.210)	(.112)

IV. PROGRAM STRUCTURE AND LOGIC

STEP is completely coded in FORTRAN 2.0 using an executive structure wherein the MAIN program controls logic to and from the various subordinate subroutines. Each of the subroutines performs a specific logical task. Because of the versatility of the programs, the logic is somewhat complex. This is necessary to accommodate the numerous options described in the input section. STEP contains the following contractor-developed subroutines

STEP1	STEP2
MAIN (called ST1)	MAIN (called ST2)
AERO	DATAB
AEROIN	FXXU
ATMDAT	INDAT
CRV	INTAG
DATAB	MINVAR
DERIVE	MOTION
FUNCT	OBSERV
FXXU	OUTPUT
INDAT	PRESET
INTAG	PROP
MINVAR	RKUTTA
MOTION	SETUP
OBSERV	SMOOTH
OUTPUT	STAT
PRESET	TAB
PROP	
RKUTTA	
SETUP	
SMOOTH	
STAT	
TAB	

The subroutines common to STEP1 and STEP2 are nearly identical.

A. Overall Program Logic

The overall program logic for filtering, error analysis, and deterministic problems is described below with the aid of figure 4.

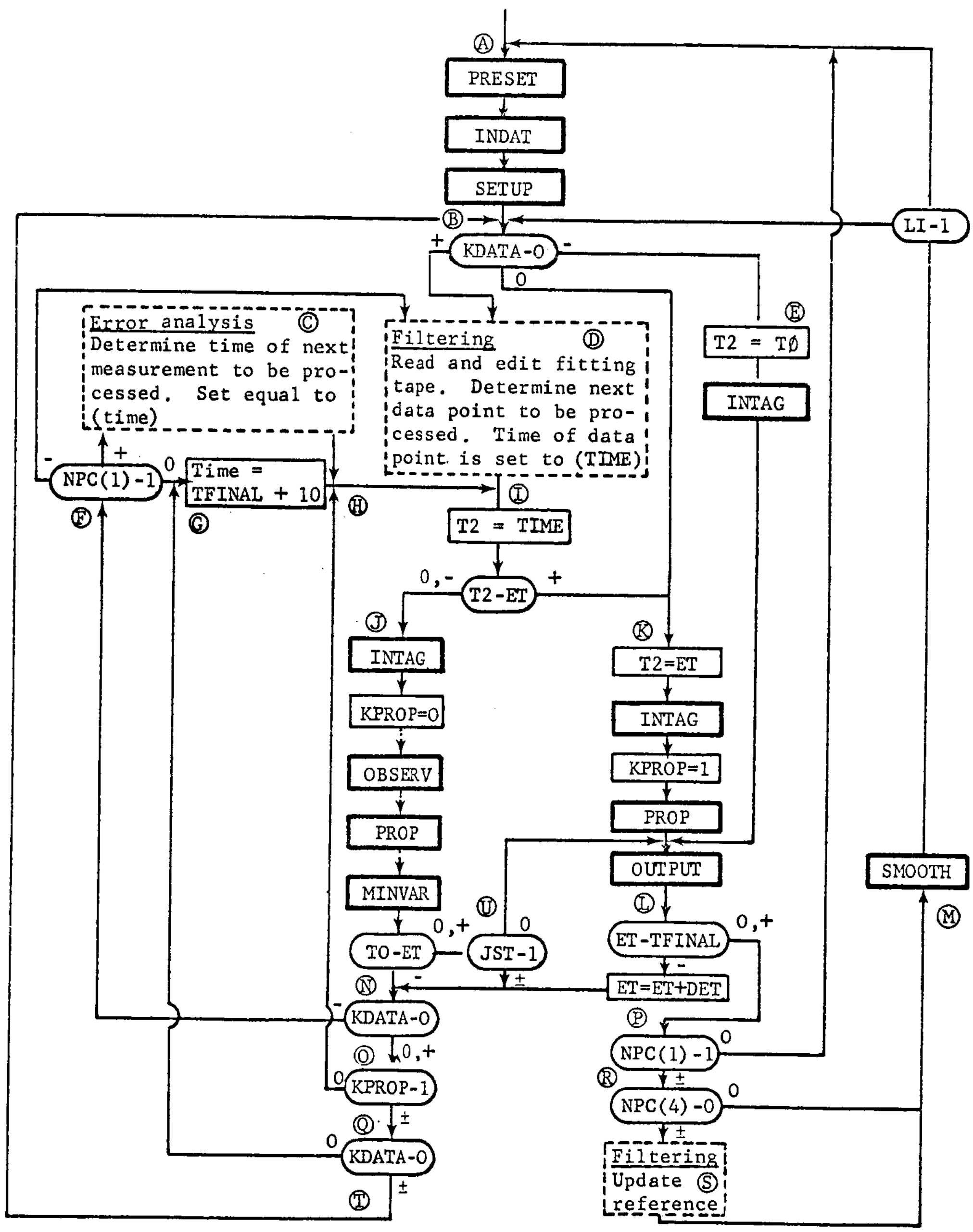


Figure 4.- Overall Program Logic

1. Filtering problems [NPC(1) = 0].- Referring to figure 4, the program commences at (A) by calling PRESET, INDAT, and SETUP. PRESET specifies values for variables and triggers. INDAT reads all input data from cards. SETUP performs all necessary calculations on the input data and controls in preparation for the processing that follows. In SETUP, the fixed point triggers KDATA and KPROP are set equal to -1, and TØ is set equal to the initial problem time. From (B) the logic transfers to (E) where INTAG is called. INTAG always integrates the nonlinear and linear differential equations of motion from TØ to T2 to yield the nonlinear state at T2 and the state transition matrix that relates variables between TØ and T2. Obviously, when INTAG is called at (E), TØ = T2 so that the computation involves only an evaluation of the differential equations and auxiliary variables. The logic then proceeds to OUTPUT where the state variables and auxiliary parameters are block printed at the initial time. The time at which the block print occurs is called ET. Thus, after OUTPUT, ET is compared with final problem time TFINAL, at (L), and, because it is smaller, it is updated by the print interval DET. Logic then proceeds to (N) where KDATA is tested. Because it is negative, the logic transfers to (F) where NPC(1) is tested. For filtering problems, NPC(1) equals zero so the logic branches to (D) where the FIT tape is read and edited to determine the next data point to be processed. The time corresponding to the data point is called TIME. If more than one data point is to be processed at TIME, the trigger JST is set to 2, otherwise JST equals 1. At (I) T2 is set equal to TIME and is then compared with the next block print time. If TIME is less than or equal to ET, the logic proceeds to (J) where, in INTAG, the nonlinear state and state transition matrix are calculated between TØ and T2. After INTAG, TØ is equated to T2 (although not shown in fig. 4). OBSERV then calculates the nonlinear measurement variables and their partial derivatives with respect to the expanded state vector components. PROP next propagates the covariance and correlation matrices, P, CUZ, and CVZ to T2, and MINVAR performs the minimum variance update of the expanded state, its covariance, and correlation matrices. TØ is then compared with ET, and, if smaller, the logic goes to the KDATA test at (N). Because KDATA corresponds to the number of points per record on the FIT tape, it is now a positive integer. Therefore, the logic drops to (O) where KPROP is tested. However, KPROP was set to 1 in the processing path (J) - (N), so the logic drops to (Q), then to (T) and back to (D) to determine the next point to be processed. As long as data are available on the FIT tape, the program loops through the circuit (D) - (I) - (J) - (N) - (T) - (B) - (D). If a block print time intervenes the logic forks after

(I) to (K). From (K) to (L) the state, covariance and correlation matrices are propagated to time ET, the print occurs in output, and the logic transfers back to (N). Because KPROP is set zero in the propagation path (K) - (L), the test below (O) now transfers back to (H) then to (I). Because during propagation (K) - (L) no data are processed, it is not necessary to determine another data point at (D). If a data time, TIME, equals the print time, ET, the processing path (J) - (N) is traversed until the last point at time TIME is processed (JST = 1). Then the logic branches through (U) to output to perform the block printing. When final time, TFINAL, is ultimately equaled or exceeded, the test below (L) transfers the logic to (P) and then to (R) because NPC(1) = 0. Below (R) the reference is updated and logic returns to (M) where SMOOTH is called. SMOOTH performs the backward smoothing from final time to initial time as well as calculation of residuals and loss function. After SMOOTH, the logic transfers to (B) where the next iteration is commenced. On the last iteration, logic is transferred back to (A) to begin the next problem. If the FIT tape runs out of data at any time during the processing, the program will read a zero record (having KDATA = 0) last from the tape. Subsequently the transfer from (Q) goes to (G) where TIME is set to exceed the final problem time. This forces the program to traverse the circuit (H) - (I) - (K) - (L) - (N) - (O) - (H) thereafter.

2. Error analysis problems [NPC(1) = 2].- The error analysis logic proceeds the same as the filtering logic with the following exceptions. The data processing time TIME is determined at (C) from card inputs rather than at (D) from the FIT tape. In MINVAR, the expanded state vector is not updated. At (S) the reference need not be updated.

3. Deterministic problems [NPC(1) = 1].- On deterministic problems the logic flows through (A) - (B) - (E) - (L) - (N) - (F) - (G). At (G), TIME is set larger than final time and the logic continues through (I) - (K) - (L) - (N) - (F) - (I) until final time is equaled or exceeded at (L). From (L) the logic branches to (P) and then to (A) to commence the next problem.

B. Subroutine Logic

The logic and equations solved in each of the subroutines will next be described with the aid of the flow diagrams and source listings in Sections VI and VII. The subroutines are grouped into five classifications -- initialization and problem setup; integration of differential equations; measurement calculations and minimum variance updating; smoothing; and output.

1. Initialization and problem setup.- The initialization and problem setup is performed by subroutines PRESET, INDAT, and SETUP. These subroutines are only called once per problem. Therefore, they have been arranged to form an overlay. After SETUP has completed its execution, the remainder of the program is read into core overlaying the cells previously occupied by PRESET, INDAT, and SETUP.

a. PRESET: The first operations performed by STEP during an execution are to preset variables. These operations are performed in subroutine PRESET, and consist of prespecifying values for variables, counters, and triggers. Should values for these variables not be input, the prespecified values will be used in the problem execution.

b. INDAT: All card inputs are read from subroutine INDAT. The unique inputting logic divides the card input data into 20 categories described in Subsection II.A.1. As each category is input, the category number on the lead card is read as a fixed point number. The remaining data on the lead card is read in the A-mode to preclude the computer from transforming the data to binary numbers immediately. By means of the category number, logic is transferred, through a computed GO TO to the category specified. Here, the A-mode numbers are decoded into binary numbers and equated to specific variables in this category. Note that the decoding of the A-mode numbers is the one feature in STEP that limits the use to CDC computers. If more data are required for the particular category, they are input by standard READ instructions. In STEP1, subroutine AEROIN is called to input the trivariate tables containing the aerodynamic coefficients. Logic is then transferred back to the beginning of INDAT where the next category input is read.

Insofar as INDAT is concerned, the order in which the categories are input makes no difference. However, some data are required before others, and it is a good rule to input the categories in numerical order, especially categories 2, 3, 5, 10, and 14. When all data have been input, the category 20 card is read to end the input loop and return to PRESET.

Note how the expanded state vector components, uncertain model and measurement vector components, and other model parameters are stored. The nonlinear state variables are input into an array called ZO(i). These are later transformed into internal variables and stored into the array X(i), i = 1, 10 in SETUP. All model parameters, whether to be estimated or not, are stored in the array C(i), the argument (i) being their identification number. The model parameters to be estimated are counted as they are input, the counter being NSTC, and their identification numbers loaded into the array NC(i). If at any later time these model parameters are required, their identification numbers are stored in NC(i), i = 1, NSTC, and their values are stored in the C(i) array. The uncertain model parameters not to be estimated, U, are counted by NPU, their identification numbers stored in MC(i), and their values stored in the C(i) array. Uncertain measurement parameters not to be estimated, V, are counted by NPV, their identification numbers stored in MCC(i), and their values in C(i). All constants known with absolute certainty are stored in the C(i) array. The number of fitting data sources or types is counted by NSTA.

c. SETUP: After all data have been input, subroutine SETUP is called to set up the problem to be executed. SETUP commences by interrogating the program controls and printing comments to inform the user of the options he has selected. In the process of doing this, the gravitational, geophysical, and atmospheric (STEP1 only) data are printed in units either specified by the user via NPC(2) or in units in which the data were input.

In STEP1, all atmospheric data are calculated in subroutine ATMDAT during problem execution. This subroutine requires base pressure and temperature/altitude slope data. These are calculated in SETUP using the built-in altitude and temperature base points and sea-level pressure corresponding to the 1962 U.S. Standard or 1959 ARDC atmospheres or data provided by the user in INDAT.

The fitting data sources are next listed. For tracking stations, the geodetic latitude is transformed to geocentric and subroutine STAT called to calculate tracking station related information to be used during the execution. Tests are also performed to determine if rate terms on tracking station k are to be estimated [MØ(k) = 1] or not estimated [MØ(k) = 0]. Also, the number of tracking stations whose station locations are to be estimated (KSS) are determined and their station numbers (MTYP = 1, 2,... 5) are stored in the array NSS(i), i = 1, KSS.

The inputted state variables are next transformed into internal variables u, v, w, h, φ, θ, e_0, e_1, e_2, and e_3 as are their covariance and correlation matrices. The transformations are presented in Section VI of Volume 1. The transformation matrix $N_I(i,j)$ is stored in the array DUB(i,j).

All angles at this point are in radians. SETUP next prints tables of the internal state variables and model parameters and their standard deviations. Finally, near the end of SETUP, the number of differential equations to be integrated is determined (NALL), and counters, triggers, arrays, and variables are set to their required values to commence the forward integration.

2. Integration of differential equations.- All linear and nonlinear differential equations are integrated in subroutine INTAG. Supporting subroutines required by INTAG to perform the integration are RKUTTA, DATAB, MOTION, FXXU, and in STEP1 AERO and DERIVE are also required.

a. INTAG: When subroutine INTAG is called, it will integrate the nonlinear differential equation in filtering and error analysis problems, from time TØ to time T2. Initial conditions must have been specified for the linear and nonlinear equation dependent variables before calling INTAG. The equations are integrated by a fourth-order Runge Kutta integration scheme as one large system. The dependent variables in this system are stored in the array X(i). Consider a problem where only the basic state variables are to be estimated. INTAG would then integrate the 10 nonlinear differential equations plus 10 independent vector solutions for the 10 linear differential equation, or 110 differential equations in all. In the array X(i); X(1-10) contains the solution to the nonlinear equations; X(11-20) contains the first solution to the linear equations (the first column of the state transition matrix Ø); X(21-30) contains the second solution to the linear equations; ... X(101-110) contains the tenth solution to the linear equations. Note that there is but one system of linear differential equations, having time varying coefficients. Ten separate solutions to this system must be obtained. At the beginning of a problem, the initial conditions for the state variables, X(1-10), are provided from SETUP. The initial conditions for the linear equations are specified in SETUP to be the identity matrix, i.e., X(11-110) = 0 except X(11) = X(22) = X(33) = ... = X(99) = X(110) = 1. Inspection of the flow diagrams and listing of INTAG reveals that the array PH(i,j) is frequently used instead of X(10-110) discussed previously. This is possible because, in common INTGRL, X(i) is dimensioned 10 and is followed by PH(i,j) dimensioned 10 x 30. Thus, X(11) is equivalent to PH(1,1), X(12) equivalent to PH(2,1) and so on. PH(i,j) is therefore the state transition matrix Ø.

The numerical integration logic is contained in subroutine RKUTTA. RKUTTA relies on the fixed point indicator LRK. The integration is commenced by setting LRK = 4 and calling RKUTTA. Upon returning from KRUTTA, LRK will equal 1, 2 or 3. When LRK = 1, the differential equations must be evaluated and logic returned to RKUTTA. LRK = 2 each time RKUTTA has completed an integration cycle. Thus, upon returning from RKUTTA with LRK = 2, the state can be printed if a printout is desired at a frequency dictated by the fixed computing interval DCOMP. The STATE tape is written during the backward integration whenever LRK = 2. RKUTTA returns with LRK = 3 when it has integrated to exactly the time T2. Because INTAG only integrates to T2, logic is returned to the main program whenever LRK = 3 on the return from RKUTTA.

b. RKUTTA: RKUTTA is a general numerical integration subroutine developed specifically for use in STEP. Its internal variables differ from those in the rest of STEP, thus through common block INTGRL, the RKUTTA variables are made equivalent to the STEP variables. As discussed above, LRK (or L in RKUTTA) is the indicator that RKUTTA uses to communicate with INTAG.

c. DATAB: Subroutine DATAB contains the logic for reading the PQR tape, writing a scratch tape containing the PQR data and interpolating tables to obtain P_M Q_M R_M and a_{X_M} a_{Y_M} a_{Z_M} (STEP2 only). At the beginning of a problem, DATAB reads the PQR tape until the last time on a PQR tape record exceeds the initial problem time. It then stores between 20 and 40 of the PQR data points in core. As the integration proceeds the core stored tables are interpolated until time exceeds that of the second to the last point in core. All points in core are then transferred to the SCRACH tape, the last two points in core are transferred to the beginning of the table, and more PQR tape data is read into core until between 20 and 40 points are in core. The integration again proceeds until time exceeds that of the second to last point in core. Again, the core-stored points are transferred to the SCRACH tape, the last two core-stored points moved to the beginning of the table and more PQR tape data read. This continues until final time is exceeded and the backward integration (smoothing) commenced. Before commencing the backward integration, a false record is written on the SCRACH tape so that later, in DATAB, two backspaces will position the tape to read the last valid record. At final time, sufficient data are already in core to commence the backward integration. When time becomes smaller than that of the second point in core, the

SCRACH tape is backspaced twice and a record is read into core. This record already contains tie-in points, i.e., the first two points previously in core. Whenever time becomes smaller than the second point in core, the SCRACH tape is backspaced twice and read. In this manner the integration proceeds to initial time. During all remaining iterations, the PQR tape is used on the forward integration and the SCRACH tape used on the backward integration.

On filter problems, capability is included to calculate residuals of the measurements and a loss function on the backward integration. To do this requires that the data from the FIT tape, which were processed on the forward integration, be stored. This storage is accomplished without requiring an additional tape unit, by having logic included in DATAB to store the processed FIT data on the SCRACH tape also. Therefore, when the SCRACH tape is written on the forward integration, all but the last two good (not edited, eligible to be processed) FIT data points in core are written on the SCRACH tape simultaneously with the PQR data. The two FIT data points, which are held back, will be written on the next SCRACH record. Therefore, during the backward smoothing, the core always contains a FIT data point at a time smaller than the PQR data in core. Subsequently it will be seen that this is necessary in subroutine SMOOTH.

d. MOTION: In subroutine MOTION, the nonlinear differential equations of motion are calculated. The state variables X(1-10) are used to calculate the state variable time derivatives DX(1-10) presented in equations (127) thru (129) of Volume I. The major portion of MOTION is used in calculating the auxiliary equations, equations (130) thru (160). In STEP1, MOTION calls subroutine ATMDAT to determine the atmosphere data, and subroutine AERO to determine the aerodynamic coefficients.

An entry point AUXIL is provided to calculate quantities needed by subroutine OUTPUT. These quantities are altered whenever the nominal is changed (i.e., when a data point is processed during an updated nominal filter problem); thus, OUTPUT must evaluate these equations immediately before printing. Entry point AUXIL is used only by subroutine OUTPUT.

e. FXXU: Subroutine FXXU calculates the coefficients in the linear differential equations of motion, equation (165). These coefficients are presented in Section V of Volume I.

Subroutine FXXU always evaluates the equations in table 4 of vol. 1. The STEP1 program calculates the equation in table 5 also. Logic is included, which determines if the state vector has been augmented to include model parameters or if any model parameters have been included in the U array (uncertain, but not to be estimated). If so, the partial derivatives associated with these model parameters are evaluated.

For STEP1, entry point PAXPC is included in subroutine FXXU. This allows equations that are common to the linearized equations of motion and the linearized model for the accelerometers to be coded only once. Entry point PAXPC is used only by subroutine OBSERV.

f. ATMOS (STEP1 only): In subroutine ATMOS, all atmospheric parameters are calculated. These quantities are presented in equations (121) thru (126) in Volume I.

g. AERO (STEP1 only): Subroutine AERO determines all aerodynamic coefficients for the vehicle. The trivariate tables are linearly interpolated by subroutine FCTN. The interpolated coefficients are either C_A, C_Y, C_N; C_D, C_Y, C_L; or C_A, C_{Y_η}, C_{N_η}. In AERO, these coefficients are then transformed to $-C_A$, C_Y, $-C_N$ by equations (134) and the error model, equation (147), solved before returning to MOTION.

Subroutine AERO has a seven-variable argument list. The first two arguments are determined by IPC_8 in category II (α, β if $IPC_8 = 0$ or -1; η, ξ if $IPC_8 = 1$). Arguments 3 and 4 are Mach and Reynolds numbers, respectively. Arguments 5, 6, and 7 are C_X, C_Y, and C_Z, which are returned to the calling program by AERO. Note that $C_X = -C_A$ and $C_Z = -C_N$.

Subroutine AERO has been written to facilitate its removal and replacement by a subroutine specialized to the particular vehicle being analyzed. The tables interpolated by subroutine AERO are the only quantities in common block AERTAB; after reading the category 11 card, subroutine INDAT transfers control to subroutine AEROIN, which reads in the tables, and then returns control to INDAT, which then continues to read in the remaining input data.

If subroutine AERO is replaced with a subroutine written for a specific vehicle, several facts should be considered. The input control IPC_8 in category 11 should be specified so the correct angles are passed to AERO; these angles are in radians, not degrees. The coefficients calculated must be transformed to C_X, C_Y, and C_Z. The error model, equation (147), must be included in subroutine AERO. Subroutine AEROIN may be modified to simplify any inputs to the new subroutine AERO.

h. TAB and FCTN: Subroutine FCTN is used solely for interpolating the trivariate aerodynamic coefficients tables. All other interpolations are performed by TAB. TAB was specifically developed for STEP and is limited to linear interpolation of monovariate tables. Each table contains an array of three fixed point numbers, N(i), i = 1, 3, which TAB requires. N(1) is the number of points in the table, N(2) is the argument of the tabular array which bounds the last interpolated value on the lower side. TAB commences searching the table from this point the next time interpolation is required. N(3) is zero for an increasing independent variable, and one for decreasing independent variable.

<u>3. Measurement calculations and minimum variance updating</u>.- On filtering and error analysis problems, the calculation of the nonlinear measurements, $Y(t_i)$, and their partial derivatives with respect to state variables and model parameters, G and H, are performed in OBSERV. In subroutine PROP, the covariance and correlation matrices, P, C_{uz}, and C_{vz}, are propagated to the measurement time by equation (54). In subroutine MINVAR, the minimum variance updating by means of equation (53) is accomplished.

The logic controlling the execution of OBSERV, PROP, and MINVAR is located in MAIN and is complicated by the optional ways that the data can be processed, i.e., scalar or vector processing with an updated or nonupdated reference of a one-, two-, or three-component measurement. The options are tested in MAIN and the flag KK is set to 1 if only one circuit through the OBSERV, PROP, and MINVAR loop is required. The only time more than one circuit is required is when processing a multicomponent vector, scalar point at a time in the updated reference mode for which case KK equals the number of vector components to be processed.

KN is set to 0 if only one component of the measurement vector and its partial derivatives need be specified in OBSERV. If more than one component must be calculated KN = 1. A loop is then commenced through OBSERV, PROP, and MINVAR with index KAR going from 1 to KK.

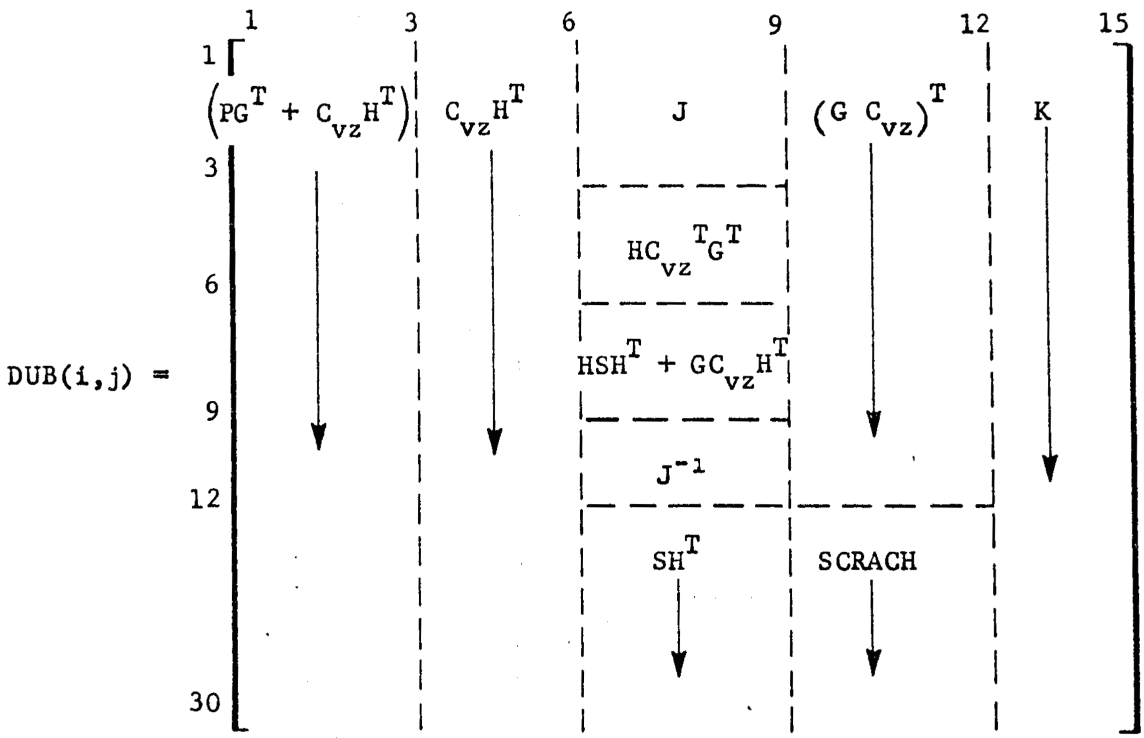
a. OBSERV: Subroutine OBSERV commences by testing the measurement type indicator, MTYP, and then branches the logic to calculate the nonlinear measurement specified. These measurement equations are presented in Volume I, Section VII. Measurements from all five tracking stations as solved from the same equations with the model parameters symbolically specified as a function of MTYP.

The three components of the measurement vector are stored in the array DFM(1-3). If only one component is to be processed (KN = 0), it is stored in DATC(KAR). If more than one component is to be processed, they are stored sequentially in DATC(i). Thus, if the second component of the measurement vector has been edited out by input controls, DATC(1) equals component one and DATC(2) equals component three. The integer KOB is set to KAR if only one component is to be processed or to JNBR if multiple components are to be processed. Thus, looping on i from KAR to KOB sequentially specifies the components of DATC(i) to be processed.

OBSERV next calculate the G and H matrices in equations (51a) and (51b). The matrices contain the partial derivative of the measurement components to be processed with respect to state variables and model parameters. The equations for these partial derivatives are presented in Section VII of Volume I. The partial derivatives are located in an array G(i,j) where i denotes the measurement vector components or rows. The rows of G are arranged similar to the elements of DATC(i), not necessarily sequentially with component number.

b. PROP: Subroutine PROP propagates the covariance and correlation matrices to the current measurement time by means of equations (54). Because the covariance matrix P is symmetric, only its upper right triangle is propagated. Before returning from PROP, the state transition matrix ϕ is reinitialized to the identity matrix.

c. MINVAR: Subroutine MINVAR next solves the minimum variance equations, equations (53). Rather than introduce additional FORTRAN variables, the dummy array DUB(i,j) is used throughout MINVAR. The order in which equations are solved and the location in DUB where the matrices are solved are as follows. Equation (53f) is first synthesized. The matrix product PG^T and GPG^T are calculated (using only the upper right triangle of P) and stored in DUB(1-3, 1-3) and DUB(1-3, 7-9), respectively. For convenience, a mapping of DUB is shown below. If uncertain measurement parameters are included (NPV > 0) then $GC_{vz}H^T$ and SH^T are calculated and stored in DUB (1-3, 7-9) and DUB (13-15, 7-9). $C_{vz}H^T$ is also added to PG^T in DUB (1-3, 1-3). Inspection of equations (53e) and (53f) reveals the J is being accumulated in DUB (1-3, 7-9) and the matrix $PG^T + C_{vz}H^T$ in DUB (1-3, 1-3). MINVAR continues to calculate matrix products and accumulating matrix sums using DUB as shown below.



The matrix J is inverted by means of the Gauss-Jordan reduction method coded in MINVAR. The 3x3 matrices shown in DUB are only calculated when a three-component vector is being processed. When the measurement vector contains fewer components, the matrices are smaller (2x2 or 1x1).

Because of the dependency of e_3 on e_0, e_1, and e_2 the tenth column of G, and the tenth row and column of P and the tenth rows of C_{uz} and C_{vz} are set to zero before their calculations in MINVAR. This produces the same results as not having the rows and columns to begin with. Before returning from MINVAR, these rows and columns of P, C_{vz}, and C_{uz} are calculated by equations (163) and (164). The correction for e_3, is calculated from equation (162).

4. Smoothing.- When the forward integration is completed, subroutine SMOOTH is called to integrate the best estimate of the state, and possibly covariance, back to initial time. SMOOTH commences by testing program controls to determine if the residuals and loss function are to be calculated, if the linear equations of motion are to be integrated to propagate the covariance and correlation matrices, or if smoothing is even to be performed. The flag KSM is set to zero if the residuals and loss function are not to be calculated, and to one if they are to be calculated. NPC(1) is temporarily reset to 1 if the linear equation are not to be integrated. SMOOTH then proceeds to integrate backwards in INTAG to either a block print time, ET, or a measurement data time TIME. The measurement data are obtained from the SCRACH tape. OBSERV is called to calculate the best estimate of the measurement. During the backward integration, OBSERV returns after calculating only the nonlinear measurements. After SMOOTH has integrated back to the initial time, it resets counters, triggers, and tapes for the next iteration or problem before returning.

5. Output.- Subroutine OUTPUT performs all the block printing on both the forward and backward integration. The variables printed are described in Subsection II.C.2. If a state update has occurred before OUTPUT, MOTION is called, via entry point AUXIL, to recalculate auxiliary variables needed for outputting. After the state and auxiliary variables have been printed, the covariance and correlation matrices are transformed to output variables specified by NPC(3). This transformation is described in Section VI of Volume I. The transformation matrix N_0 is

stored in the DUB(i,j) array as follows:

$$N_0^T(1\text{-}3,\ 1\text{-}10) = DUB(1\text{-}10,\ 1\text{-}3)$$

$$N_0^T(7\text{-}9,\ 1\text{-}10) = DUB(1\text{-}10,\ 4\text{-}6)$$

The middle three rows of N_0 contain only the identity matrix for h, φ, and θ so they are not stored.

V. DEFINITION OF FORTRAN SYMBOLS

A(i,j)	Transformation matrix G between the G-frame and B-frame, see equation (133).
AB(i), AC(i)	Accelerations a_{XB}, a_{YB}, a_{ZB} acting through the center of gravity in body axes directions, see equations (134) and (146).
AG(i)	Accelerations a_{XG}, a_{YG}, a_{ZG} in the G-frame axes directions, see equation (132).
ALPH	Angle of attack, α, see equation (135).
AM(i)	Accelerations a_{XM}, a_{YM}, a_{ZM} modeled or measured at the inertial measuring unit in body axes directions with systematic error included, see equations (159) and (258).
AMDOT(i)	Time derivative of a_{XM}, a_{YM}, a_{ZM}, see equation (160).
AP(i)	AP(1-3) are accelerations a_{XP}, a_{YP}, a_{ZP} acting at the inertial measuring unit in body axes directions without systematic error; AP(4) is $\overline{a}_p$, see equations (158) and (159).
APDOT(i)	Time derivatives of a_{XP}, a_{YP}, a_{ZP}, and $\overline{a}_p$, see equations (160) and (157).
AS	Speed of sound c_S, see equation (125).
ASU	Sutherland constant, S, see equation (126).
AX(i), AY(i), AZ(i)	Tabular array of the accelerations a_{XM}, a_{YM}, a_{ZM} from the PQR tape.
B(i)	Array used in INDAT to input A-format information.
BETA	Sideslip angle, β, see equation (136).

C(i)	Model parameters C_1, C_2, ..., C_{152}.
CALP	Cosine of the angle of attack, α, in equation (135).
CBET	Cosine of the sideslip angle, β, in equation (136).
CCAPHI(i)	Cos Φ corresponding to tracking station i, see equation (236).
CDP	Cosine pitch angle δ_P in equation (269).
CDY	Cosine yaw angle δ_Y in equation (269).
CETA	Cosine of total angle of attack, η in equations (137) and (187).
CF(i)	Array containing the aerodynamic coefficiets $-C_A$, C_Y, $-C_N$.
CGR	Cosine γ_R in equation (271b).
CGM(i)	Modeled distances, x_{PM}, y_{PM}, and z_{PM}, from the center of gravity to the IMU along the body axes directions.
CI(i,j)	Matrix inverse, C^{-1} calculated in OBSERV, see equation (266).
CJ(i,j)	Product of matrix inverse in equation (259) times the transformation matrix involving P, Q, and R in equation (260); calculated in OBSERV.
CLR	Cos λ_R in equation (271a).
CONRD	Conversion from degrees to radians; equal to 0.017453292.
CPH	Cosine of geocentric latitude φ.
CPHIDT(i)	Cos φ_{DT} corresponding to tracking station i, see equation (237).

CT(i)	Dummy array of cosines used in SETUP and OUTPUT for $\cos \bar{\gamma}_A$, $\cos \lambda_A$, $\cos \frac{\bar{\psi}}{2}$, or $\cos \frac{\sigma}{2}$, $\cos \frac{\bar{\theta}}{2}$, $\cos \frac{\beta}{2}$, and $\cos \frac{\bar{\phi}}{2}$ or $\cos \frac{\alpha}{2}$.
CTM(i)	Used in OBSERV for calculating R_c, A_c, E_c, $\dot{R}_c$, $\dot{A}_c$, and $\dot{E}_c$, respectively from equations (238) and (240).
CUZ(i,j), CVZ(i,j)	Correlation matrices C_{uz} and C_{vz}, see equations (53) and (54).
CXZI	Cosine of the steering angle, ξ, in equations (138) and (187).
D(i)	Variances of uncertain model parameters not being estimated, see equations (41) and (54).
DADX(i,j)	Partial derivatives of α and β with respect to state variables, see equation (186).
DAT(i,j)	Array of data from FIT tape that is stored in core; subscript i indicates component.
DATA(i)	Components of the current measurement vector being processessed; see MAIN and MINVAR.
DATAS(i)	Components of the next measurement vector to be processed; see MAIN.
DATC(i)	Measurement vector calculated in OBSERV; note that DATC(1), DATC(2), and DATC(3) do not correspond to the components of the data vector but to only those components to be processed, e.g., DATC(1) = E_M, DATC(2) = A_M if R_M is edited out.
DCDY(i,j)	Partial of $-C_A$, C_Y, $-C_N$ with respect to V_A, h_o, α, and β, see equation (186).

DCOMP	Fixed computing interval for the numerical integration.
DERIV(i,j)	Partial of a_{XB}, a_{YM}, a_{ZB} with respect to state variables, see equations (182) and (184).
DET	Print interval for block output.
DFIT(i)	Minimum time span between data points processed from the FIT tape of measurement type i.
DFM(i)	Components 1, 2, and 3 of the calculated measurement vector in OBSERV.
DPH(i,j)	Time derivative of the dependent variable in the linear differential equations of motion; equivalently the time derivative of the state transition matrix.
DRDH	Partial derivative of ρ with respect to h_o in equation (185).
DRDP	Partial derivative of r with respect to φ.
DTI(i)	Time to commence processing data of type i.
DTF(i)	Time to stop processing data of type i.
DTRAN(i,j)	Transformation matrix in equation (189).
DUB(i,j)	Dummy array used throughout program; especially in SETUP and OUTPUT for input/output transformations N_I and N_o, and in MINVAR.
DUD(i)	Dummy array.
DUE(i)	Dummy array.
DUF(i)	Dummy array.
DWDH(i)	Array containing du_{WM}/dh_o and dv_{WM}/dh_o calculated in DERIV.

DX(i)	Time derivatives of the internal state variables and state transition matrix.
DZ(i)	Expanded state variable perturbations $\hat{z}(t_i \mid t_i)$ calculated in MINVAR, see equation (54a).
ET	Time to print block output.
F1(i,j)	Coefficients of first three linear differential equations of motion calculated in MOTION and FXXU; also used in OBSERV for calculating adjoint matrix in equation (264).
F2(i,j)	Coefficients of middle three linear differential equations of motion.
F3(i,j)	Coefficients of last four linear differential equations of motion.
FIT	Logical tape number of the FIT tape, Fit = 1.
FLOS	Loss function calculated in SMOOTH.
G(i,j)	Partial derivative of the measurement vector with respect to expanded state vector components, G, see equation (6b).
GG(i)	Temporary storage for each column of G(i,j) as they are calculated in OBSERV.
H(i,j)	Partial derivative of the measurement vector with respect to uncertain measurement vector components, see equations (51a) and (51b).
HI(i)	Dummy array.
HO	Altitude above an oblate planet surface.
ICOUNT	Iteration counter.
IDN	Dummy integer variable used in INDAT.
II	Dummy integer variable.
IN	Logical input tape number, IN = 5.

IP	Dummy integer variable.
IPC	Temporary storage for input controls.
JJ	Dummy integer variable.
JN	Dummy integer variable.
JNBR	Number of components to be processed in the current measurement vector.
JNBRS	Number of components to be processed in the next measurement vector.
JST	Specified in MAIN to be 1 if only one measurement is to be processed at time TIME; JST $>$ 1 if more than one vector measurement is to be processed at TIME.
KA	Dummy variable.
KAR	Index on loop through OBSERV, PROP, and MINVAR, see MAIN program.
KC(i)	Point numbering array used when printing the measurement schedule in OBSERV.
KD(i)	Point numbering array used in SMOOTH for printing the residuals.
KDAP	Number of points on PQR tape record, see DATAB.
KDATA	Number of points on previous FIT tape record, see MAIN.
KDATAS	Number of points on present FIT tape record, see MAIN.
KDUM	Dummy variable in INDAT.
KG	Counter for acceptable data points read from the FIT tape, which are stored in core and either have been or are to be processed.

KG2	Number of measurement points from FIT tape that are written on the SCRACH tape in DATAB.
KK	Number of times loop is traversed between OBSERV and MINVAR in MAIN.
KN	Flag set in MAIN to 0 if the reference is updated between processing of the measurement components; KN = 1 if state is not updated between component processing.
KØB	Largest measurement vector component to be processessed in MINVAR; loop goes from KAR to KOB; KOB is set in OBSERV.
KPRØP	Indicator that is initialized -1 in SETUP, is set to 0 after each minimum variance update, and set to 1 after each propagation; see figure 4.
KS	Number of FIT tape measurement vectors stored in core; updated in MAIN.
KSS	Number of tracking stations whose location is being estimated.
KSM	Indicator in SMOOTH; if KSM = 0, do not calculate residuals; if KSM = 1, calculate residuals and loss function.
K1	Dummy variable.
LC(i)	Array containing the component numbers of the current measurement vector DATA(j) to be processed, e.g., LC(1) = 2, LC(2) = 3 if component 1 is edited out.
LCS(i)	Same as LC(i), but corresponding to the next data vector DATAS(j).
LS	Number of PQR tape points that are stored in the arrays TT(i), TP(i) . . . AZ(i); counter is updated in DATAB.
LT	Dummy variable.

L1	Flag used to determine if fitting schedule title should be printed from OBSERV; print titles if L1 = 2, do not print if L1 ≠ 2; also used to print residual schedule titles in SMOOTH if L1 ≠ 0, and used as a transfer flag at return from SMOOTH; If L1 = 0, start next problem, if L1 = 1, start next iteration.
MC(i)	Array containing the identification numbers for variables in U, the uncertain model parameters not to be estimated.
MCC(i)	Array containing the identification numbers for variables in V, the uncertain measurement parameters not to be estimated.
MØ(i)	Flag set to 1 if rate terms are included in the radar tracking error model, equation (239), from station i; otherwise zero.
MR(i,j)	Flag equal to 1 if component i of measurement type j of the FIT tape data is to be processed; flag equals 0 if not to be processed.
MTP(i)	Array containing the data type identification number corresponding the FIT tape data that has been read into core.
MTP	Measurement data type identification number for current measurement to be processed MTP = 1, 2, 3, 4, 5 for radar tracking from stations 1, 2, 3, 4, 5; = 6 for airborne radar; = 9 for accelerations; = 8 for position vector; and = 7 for velocity vector.
MTYPS	Same as MTP, but corresponds to next data vector to be processed.
NC(i)	Array containing the identification numbers for model parameters to be estimated.
NCOUNT	Processing measurement point counter.
NNN	Dummy variable.

NPC(i)	Program controls specified in INDAT, see Section II.A.2.
NPTS	Dummy variable used in INDAT to denote the number of points on a table; one point consists of two scalars numbers, the independent variable and dependent variable.
NPU	Number of components in U, the uncertain model parameter vector not to be estimated.
NPV	Number of component in V, the uncertain measurement parameters not to be estimated.
NS(i)	Array containing the measurement type indicators (MTYP) for measurement to be processed from FIT tape; NS(i) is set in INDAT.
NSS(i)	Array containing the station numbers of radar stations whose location is to be estimated.
NST	Counter for state variable components input in INDAT.
NSTA	Measurement data type counter input in INDAT.
NSTC	Counter for model parameters which are to be estimated.
NSTX	Number of components in expanded state vector to be estimated; NSTX = 10 + NSTC.
NT	Dummy variable.
NTR(i)	Flag used in editing measurement type i in MAIN; NTR(i) is initialized to zero at beginning of problem; thereafter, NTR(i) = 2 indicates the last fitting point was accepted, = 3 indicates the last fitting point was rejected.
N4(i)	Array used by TAB for interpolating $m_M(t)$ table.

N5(i)	Array used by TAB for interpolating $u_{WM}(h)$ table.
N6(i)	Array used by TAB for interpolating $v_{WM}(h)$ table.
NS(i)	Array used by TAB for interpolating $x_{PM}(t)$ table.
N9(i)	Array used by TAB for interpolating $y_{PM}(t)$ table.
N10(i)	Array used by TAB for interpolating $z_{PM}(t)$ table.
N11(i)	Array used by TAB for interpolating $P_M(t)$, $Q_M(t)$, $R_M(t)$ tables.
OMEGA	Planet rotation rate Ω_P.
ØUT	Logical output tape number, ·OUT = 6.
P(i,j)	Covariance matrix P.
PA(i)	Inertial angular rates, P, Q, R calculated in MOTION.
PAR(i)	Parameters calculated in MOTION and subsequently used in other subroutines.
PAXP(i,j)	Partial of a_{XP}, a_{YP}, a_{ZP} with respect to state variables, see equation (189).
PDOT(i)	Time derivative of P, Q, and R calculated in MOTION.
PE	Atmospheric pressure calculated in ATMDAT from equation (123).
PH(i,j)	State transition matrix ϕ.
PHIDT(i)	Geodetic latitude of tracking station i.
PM(i)	Measured inertial angular rates P_M, Q_M, and R_M.

PMDOT(i)	Time derivatives of P_M, Q_M, and R_M.
PQR	Logical tape number of the PQR tape, PQR = 2.
R	Radial distance from planet center to vehicle, r.
RCC(i)	Partial derivatives of R_c, A_c, E_c with respect to C_i calculated in OBSERV from equation (250).
RCDC(i)	Partial derivatives of $\dot{R}_c$, $\dot{A}_c$, $\dot{E}_c$ with respect to C_i, calculated in OBSERV from equation (251).
RCDX(i,j)	Partial derivatives of $\dot{R}_c$, $\dot{A}_c$, $\dot{E}_c$ with respect to u, v, w, h, φ, θ calculated in OBSERV from equation (245).
RCX(i,j)	Partial derivative of R_c, A_c, E_c with respect to h, φ, θ calculated in OBSERV from equation (244).
REØ	Equitorial radius of planet, R_E.
RERP2	Parameter $(R_E/R_P)^2$.
RES(i)	Residuals and weighted residuals in SMOOTH.
RØ	Radius of oblate planet, R_o.
RØE	Atmospheric density corrected for systematic error in MOTION from equation (151).
RØEC	The exponential portion of the atmospheric density correction; calculated In MOTION from equation (151).
RØEM	Modeled atmospheric density ρ_M, transferred to MOTION from ATMDAT where it is calculated via equation (124).

RPØ	Polar radius of planet.
RT(i)	Radial distance from planet center to tracking station i.
ROT(i)	Radius of oblate planet at latitude of tracking station i.
S(i)	Variances of uncertain measurement parameters not being estimated, see equations (44) and (53).
SALP	Sine of angle of attack α, see equation (135).
SBET	Sine of sideslip angle β, see equation (136).
SCAPHI(i)	Cos Φ corresponding to tracking station i.
SCRACH	Logical tape number of the SCRACH tape, SCRACH = 3.
SDP	Sine of pitch angle δ_p in equation (269).
SDY	Sine of yaw angle δ_Y in equation (269).
SETA	Sine of total angle of attack η, see equation (137).
SGR	Sin γ_R in equation (271b).
SI(i)	Standard deviations of the fitting measurements processed in MINVAR.
SIG(i,j)	Array of standard deviations from FIT tape, which are stored in core.
SIGM(i)	Standard deviations of measurement vector currently being processed.
SIGMS(i)	Standard deviations of next measurement vector to be processed.

SLR	Sin λ_R in equation (271a).
SP(i)	Scratch pad or dummy variable used throughout program.
SPD(i)	Scratch pad or dummy variable used throughout program.
SPH	Sine of the geocentric latitude φ.
SPHIDT(i)	Sin φ_{DT} corresponding to tracking station i, see equation (237).
SREF	Reference area for aerodynamic coefficients.
ST(i)	Dummy array of sines used in SETUP and OUTPUT for sin γ_A, sin λ_A, sin $\frac{\bar{\psi}}{2}$, or sin $\frac{\sigma}{2}$. sin $\frac{\bar{\theta}}{2}$, or sin $\frac{\beta}{2}$, and sin $\frac{\bar{\phi}}{2}$, or sin $\frac{\alpha}{2}$.
STATE	Logical tape number for STATE tape, STATE = 4.
SUM	Dummy variable use primarily for accumulating matrix products.
SUM2	Dummy variable used similarly to SUM.
SXZI	Sine of steering angle ξ, see equation (138).
SYG(i,j)	Incremental standard deviations on component i of measurement type j input in category 7 input.
T	Running variable for time used in RKUTTA.
TABLES(i,j)	Array containing all three aerodynamic coefficient tables in AERO.
TAU	Time increment $t-t_1$ calculated in MOTION from equation (150).
TCX(i)	First aero table C_A or C_D in AERO; note TCX(i) is equivalent to TABLE(i,1).

TCY(i)	Second aero table C_Y in AERO; note TCY(i) is equivalent to TABLE (i,2).
TCZ(i)	Third aero table C_N or C_L in AERO; note TCZ(i) is equivalent to TABLE(i,3).
TFIT(i)	Minimum time for processing the next type i measurement in MAIN; all type i data before TFIT(i) will be rejected.
TFINAL	Final problem time; stop the forward integration after the first block printout that equals or exceeds TFINAL.
TIME	Time corresponding to the current measurement vector being processed.
TIMES	Time corresponding to the next measurement vector to be processed.
TMAS(i)	Independent variable on the vehicle mass table $m_M(t)$.
TØ	Time to initiate integration in INTAG
TONE	Time t_1 at which mass correction commences, see equations (149) and (150).
TP(i)	Table containing the inertial angular rate data, P_M, read from the PQR tape.
TPH	Tangent of geocentric latitude φ.
TQ(i)	Table containing the inertial angular rate data, Q_M, read from the PQR tape.
TR(i)	Table containing the inertial angular rate data, R_M, read from the PQR tape.
TRAN(i,j)	Transformation matrix containing P, Q, and R in equation (146).
TT(i)	Table containing the independent variable (time) for TP(i), TQ(i), TR(i) and AX(i), AY(i), AZ(i) (STEP2 only).

T2WØ	Time t_2 at which mass correction stops, see equation (149) and (150).
TUW(i)	Independent variable on atmospheric wind table $u_{WM}(h_o)$.
TVW(i)	Independent variable on atmospheric wind table $v_{WM}(h_o)$.
TXCG(i)	Table containing the independent variable (time) for $x_{PM}(t)$.
TYCG(i)	Table containing the independent variable (time) for $y_{PM}(t)$.
TYM(i)	Table containing the measurement data times from the FIT tape.
TZCG(i)	Table containing the independent variable (time) for $z_{PM}(t)$.
TZERØ	Initial problem time.
T2	Time to stop the integration in INTAC and return to the point where INTAG was called.
UW	Atmospheric wind component u_W calculated in MOTION from equation (152).
VA(i)	Array containing u_A, v_A, w_A, V_A, and V_A^2, calculated in MOTION from equations (140) and (141).
VB(i)	Components of V_A in body axes directions, u_B, v_B, w_B; calculated in MOTION from equation (139).
VØ(i)	Dummy array used for temporary storage.
VW	Atmospheric wind component v_W calculated in MOTION from equation (152).

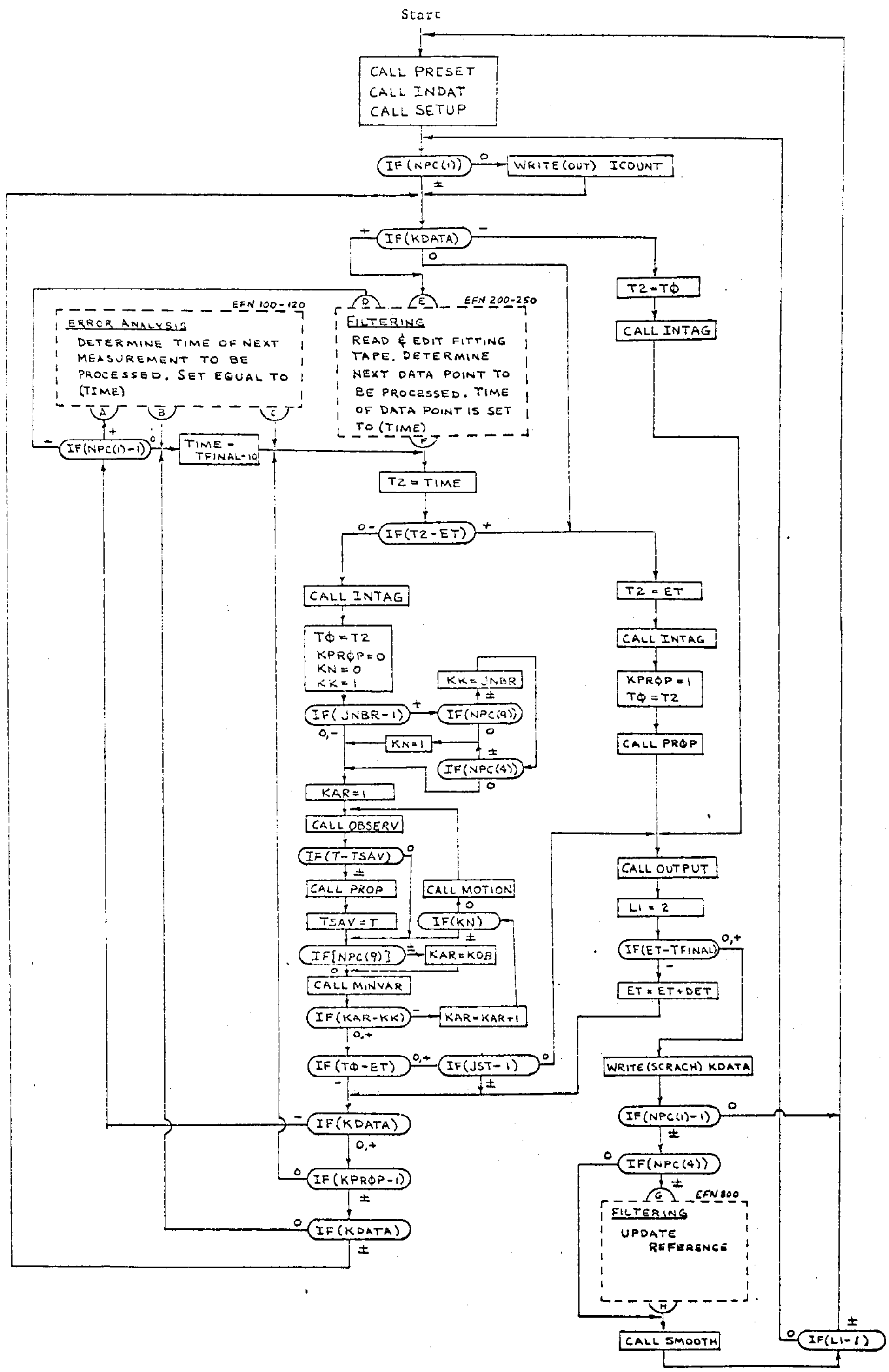
X(i)	Dependent variables in the nonlinear differential equations of motion.
XA(i)	Array containing a_1, a_2, ..., a_5 in equation (277).
XAX(i,j)	Partial derivatives of a_1, a_2, ..., a_5 with respect to h, φ, θ, e_o, e_1, e_2, e_3; calculated in OBSERV from equation (283).
XC(i)	Array containing C_1, C_2, C_3; calculated in OBSERV from equation (276).
XCX(i,j)	Partial derivatives of C_1, C_2, C_3 with respect to h, φ, θ, e_o, e_1, e_2, e_3; calculated in OBSERV from equation (282).
XIND1, XIND2	Independent variable α, β or η, ξ used in AERO to calculate aerodynamic coefficients.
XJ2	Coefficient of the second gravitational harmonic, J_2, see equation (106).
XLB(i)	Direction cosines l_B, m_B, n_B; calculated in OBSERV from equation (269).
XLG(i)	Direction cosines l_G, m_G, n_G; calculated in OBSERV from equation (270).
XLGE(i,j)	Partial derivatives of l_G, m_G, n_G with respect to e_o, e_1, e_2, e_3; calculated in OBSERV from equation (286).
XLGX(i,j)	Partial derivatives of λ_R and γ_R with respect to e_o, e_1, e_2, e_3; calculated in OBSERV from equations (284) and (285).
XLREF	Vehicle reference length, l, used in calculating Reynolds number, see equation (148).

XM	Mach number, M; calculated in MOTION from equation (143).
XMAS(i)	Dependent variable on the $m_M(t)$ table.
XMU	Coefficient of the first gravitational harmonic μ; see equation (106).
XMUJ	The product $2 \mu J$; see equation (106).
XNU	Atmospheric viscosity μ_A; transferred to MOTION from ATMDAT where it is calculated via equation (126).
XP(i)	The center-of-gravity/IMU distances x_p, y_p, z_p in equation (153).
XQ	Dynamic pressure, q; calculated in MOTION from equation (142).
XQSM	Parameter equal to XQ*SREF/YMAS calculated in MOTION.
XRE	Reynolds number, R_E; calculated in MOTION from equation (148).
XS(i)	Cartesean components of the radar tracking vector, x_s, y_s, z_s; calculated in OBSERV from equation (235).
XSC(i)	Partial derivatives of x_s, y_s, z_s with respect to C_{88}, C_{89} or C_{90}; calculated in OBSERV from equation (252).
XSD(i)	Time derivatives $\dot{x}_s$, $\dot{y}_s$, $\dot{z}_s$ calculated in OBSERV from equation (241).
XSDC(i)	Partial derivatives of $\dot{x}_s$, $\dot{y}_s$, $\dot{z}_s$ with respect to C_{88}, C_{89}, or C_{90}; calculated in OBSERV from equation (253).

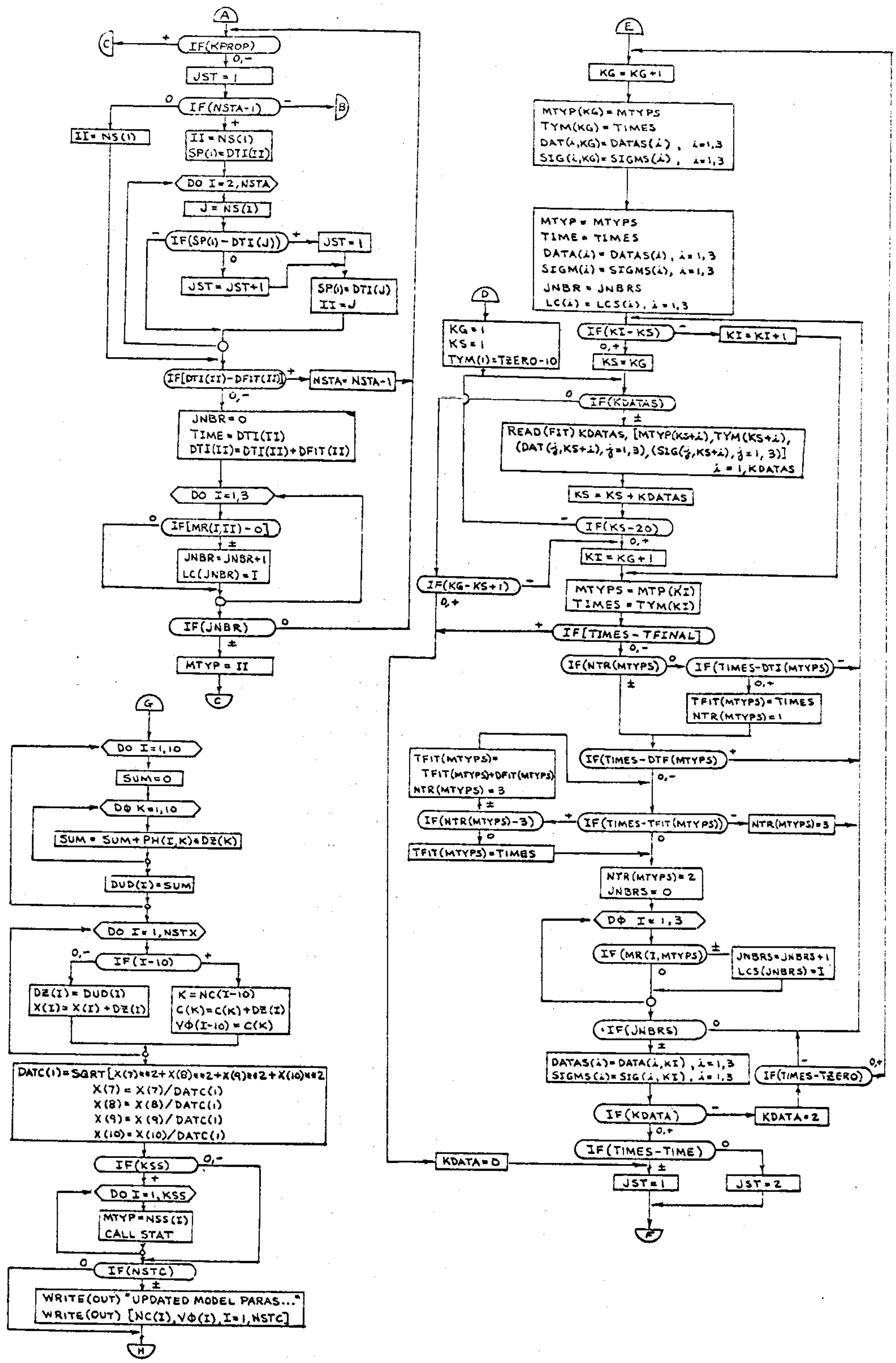
XDSX(i,j)	Partial derivatives of $\dot{x}_s$, $\dot{y}_s$, $\dot{z}_s$ with respect to u, v, w, h, φ, θ; calculated in OBSERV from equation (247).
XSX(i,j)	Partial derivatives of x_s, y_s, z_s with respect to h, φ, θ calculated in OBSERV from equation (246).
XTEMP	Atmospheric temperature transferred to MOTION from ATMDAT.
XUW(i)	Dependent variable on atmospheric wind table $u_{WM}(h_o)$.
XVW(i)	Dependent variable on atmospheric wind table $v_{WM}(h_o)$.
XXCG(i)	Dependent variable on $x_{PM}(t)$ table, see equation (153).
XYCG(i)	Dependent variable on $y_{PM}(t)$ table, see equation (153).
XZCG(i)	Dependent variable on $z_{PM}(t)$ table, see equation (153).
XZI	Steering angle ξ, see equations (138) and (187).
YMAS	Vehicle mass, m, corrected for systemmatic error; calculated in MOTION from equation (149).
YMASM	Vehicle mass as interpolated from the model mass table.
ZØ(i)	Dummy array used for temporary storage.

VI. FLOW DIAGRAMS

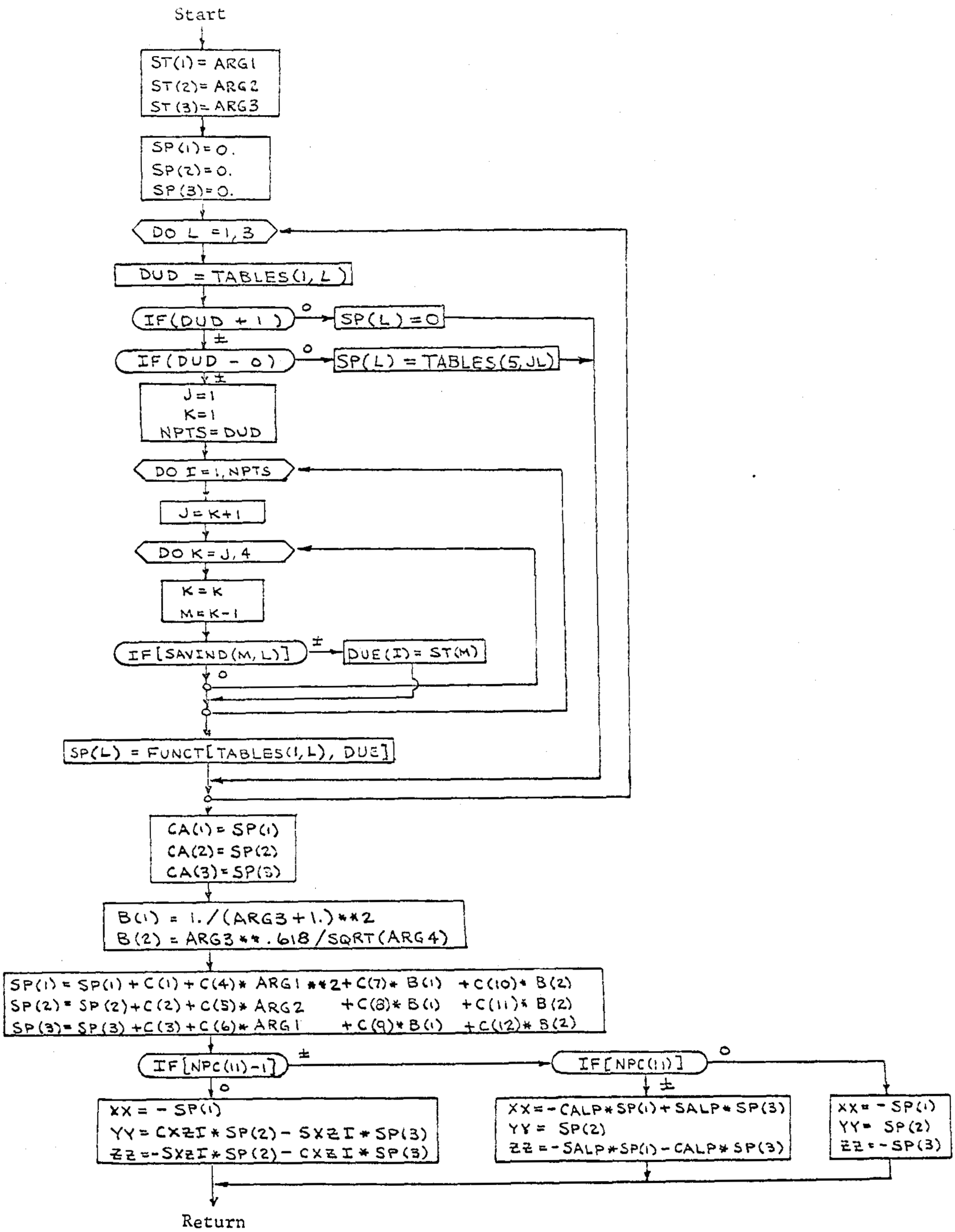
Subroutine flow diagrams are presented in this section. The following listing indicates the page on which the various flow diagrams appear.



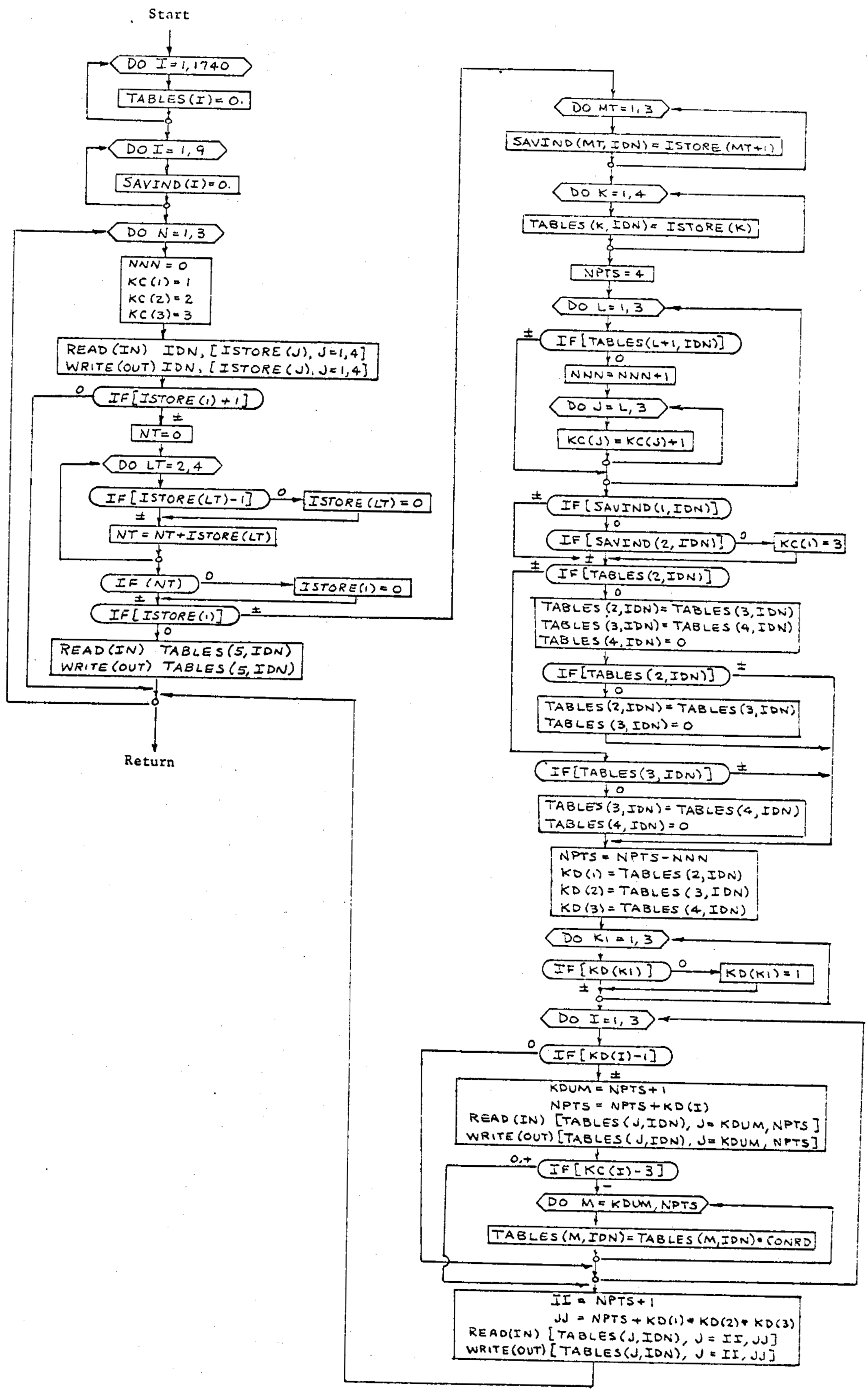
MAIN Program



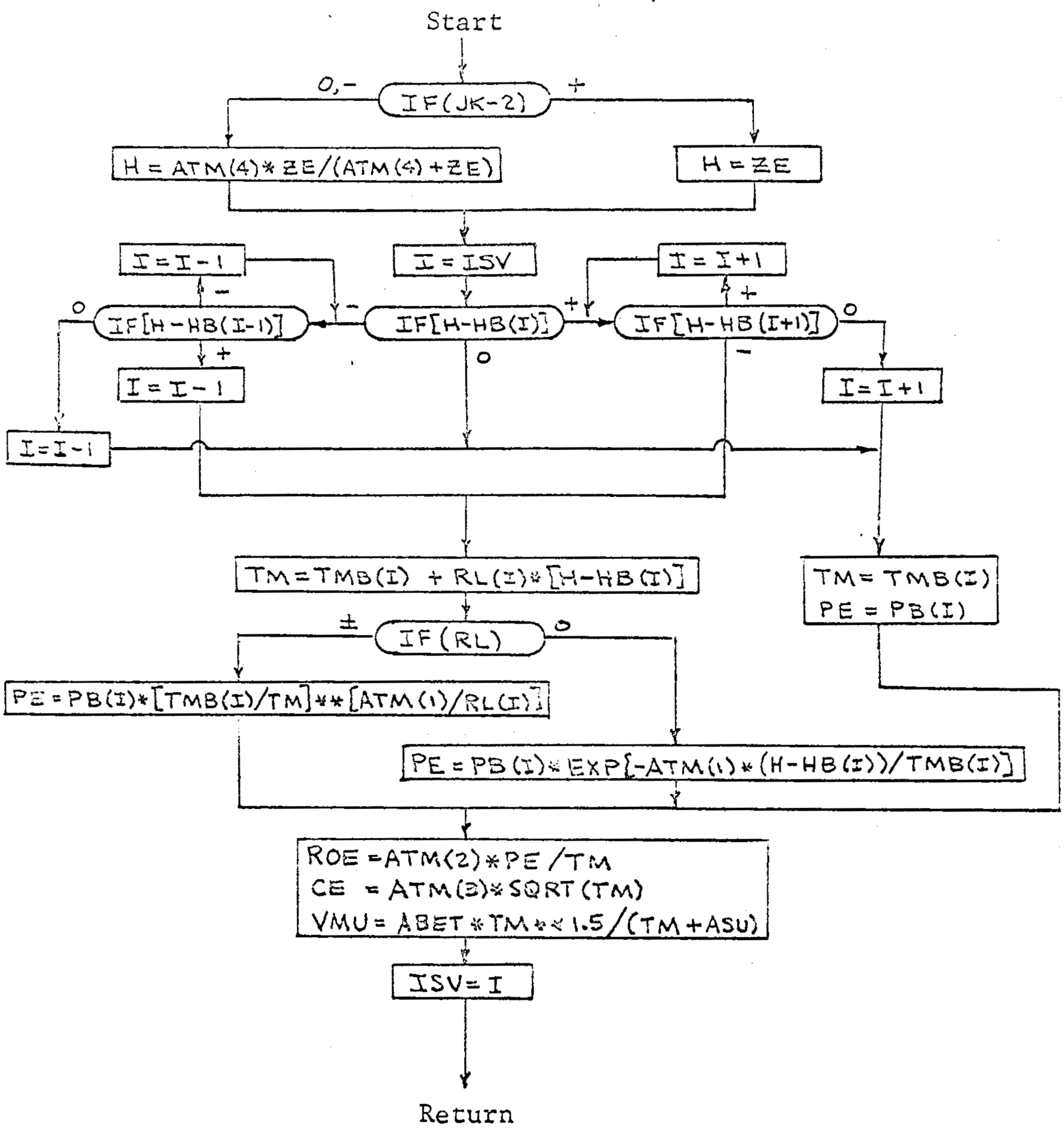
MAIN Program - Concluded



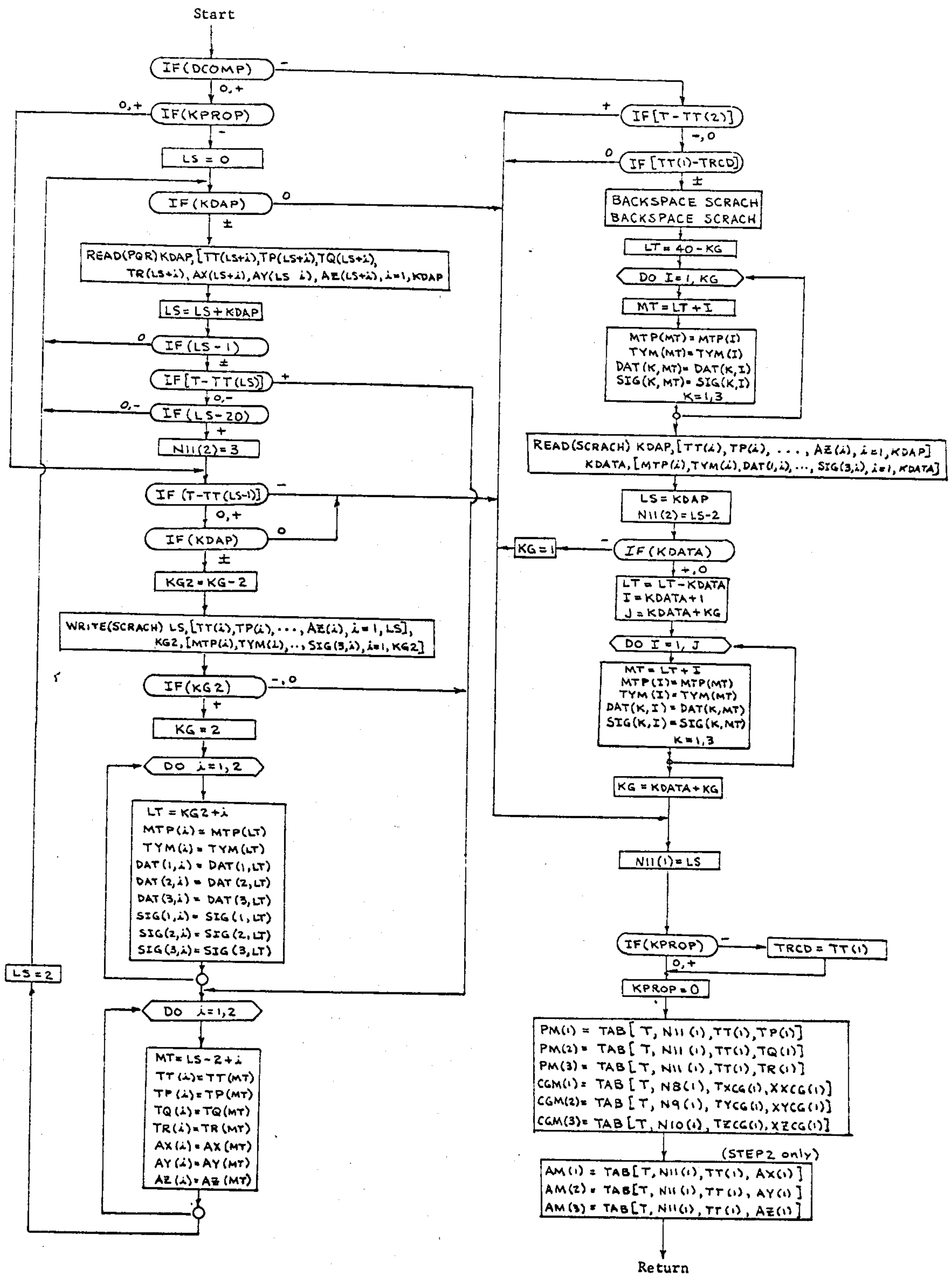
Subroutine AERO (ARG1, ARG2, ARG3, ARG4, XX, YY, ZZ)



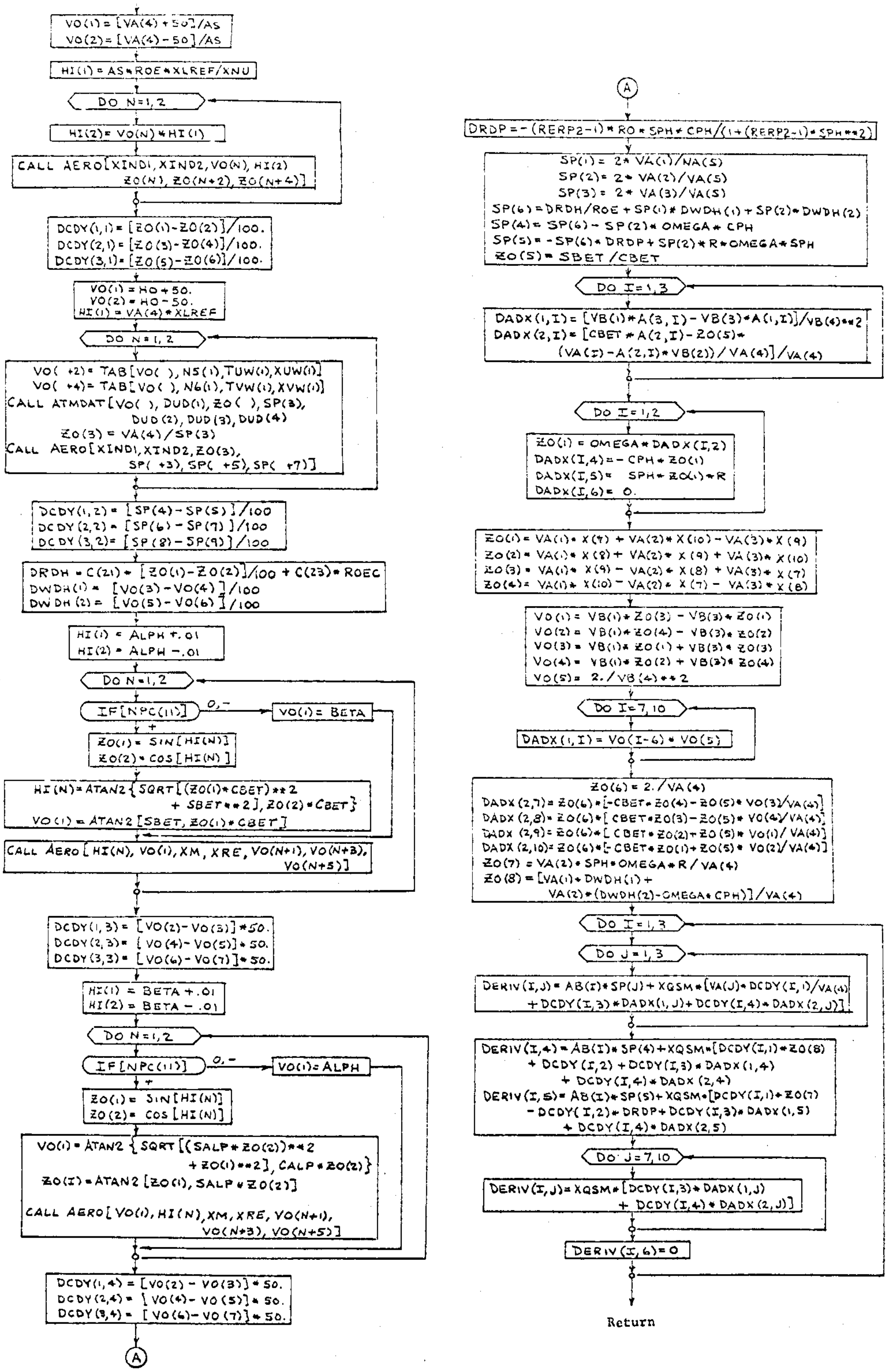
Subroutine AEROIN



Subroutine ATMDAT (ZE, PE, ROE, CE, TM, VMU, ISV)



Subroutine DATAB



Subroutine DERIVE

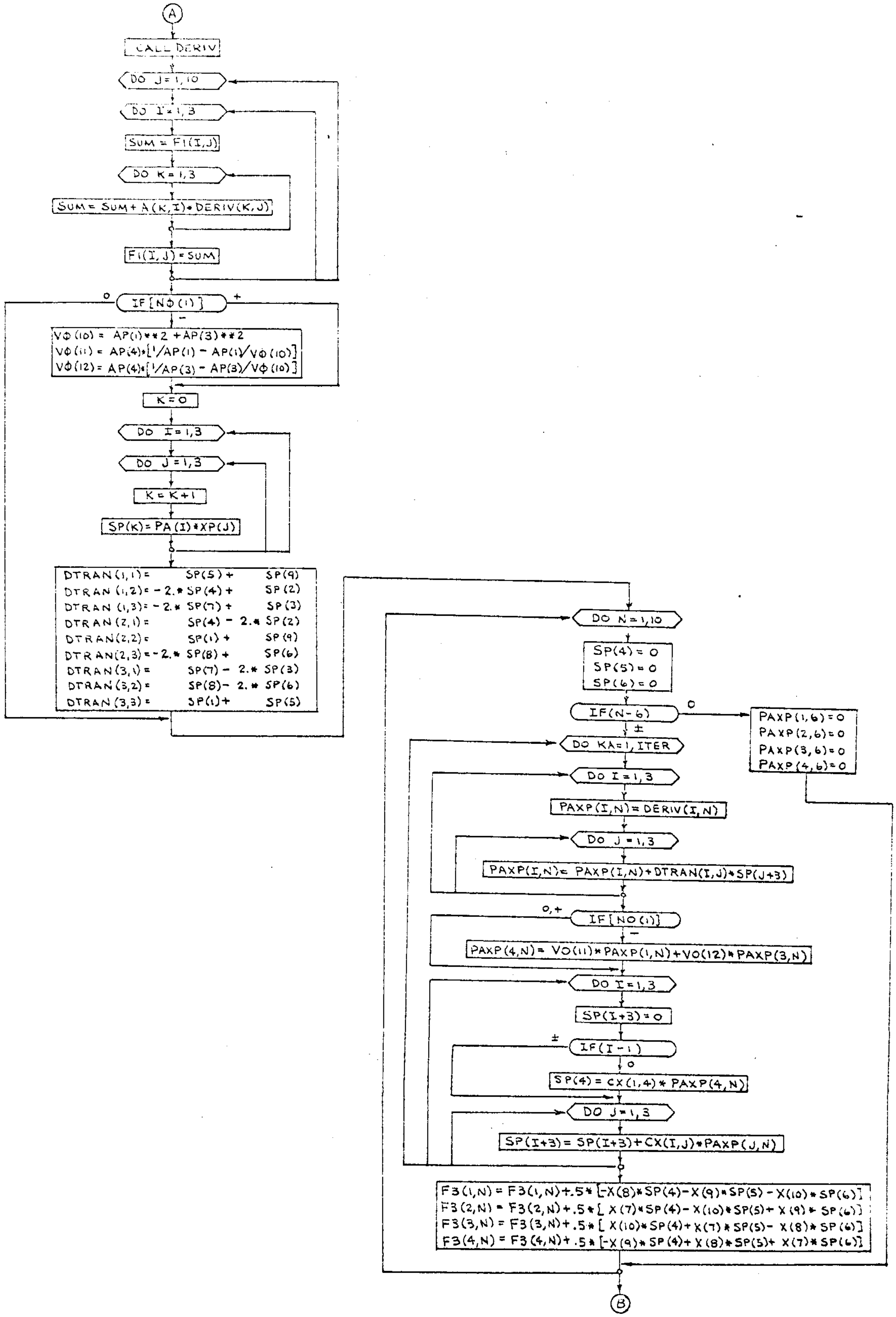
Start

```
SP(1) = PAR(4)*X(2)/CPH**2
F2(2,1) = 1./R
F1(1,3) = F2(2,1)*X(1)
F1(3,1) = -F1(1,3) - F1(1,3)
F1(3,2) = -F1(2,3) - F1(2,3)
F1(1,1) = F2(2,1)*X(3)
F1(1,2) = -F1(2,1) - F1(2,1)
F1(1,4) = [-PAR(5)*F2(2,1)
     +4.*PAR(3)*SPH*CPH]*F2(2,1)
F1(1,5) = SP(1)*F1(3,2)*R
     - PAR(3)*(CPH+SPH)*(CPH-SPH)
F1(2,4) = -F1(2,2)*F1(2,3)
F1(2,5) = -SP(1)*F1(3,1)*R
F1(3,4) = [PAR(1)*F2(2,1)-PAR(2)
    -PAR(2)+4.*PAR(3)*PAR(6)]*F2(2,1)
F1(3,5) = -3.*PAR(3)*SPH*CPH
F2(2,4) = -F1(1,3)*F2(2,1)
F2(3,2) = F2(2,1)/CPH
F2(3,4) = -F1(2,3)*F2(3,2)
F2(3,5) = F1(2,1)/CPH
F3(1,1) = -X(9)*PAR(4)
F3(2,1) = X(10)*PAR(4)
F3(3,1) = X(7)*PAR(4)
F3(4,1) = -X(8)*PAR(4)
F3(1,2) = [X(8)-X(10)*TPH]*PAR(4)
F3(2,2) = -[X(7)+X(9)*TPH]*PAR(4)
F3(3,2) = [X(10) + X(8)*TPH]*PAR(4)
F3(4,2) = -[X(9)- X(7)*TPH]*PAR(4)
F3(1,4) = -F1(1,3)*F3(1,1) - F1(2,3)*F3(1,2)
F3(2,4) = -F1(1,3)*F3(2,1) - F1(2,3)*F3(2,2)
F3(3,4) = -F1(1,3)*F3(2,2) - F1(2,3)*F3(3,2)
F3(4,4) = -F1(1,3)*F3(4,1) - F1(2,3)*F3(4,2)
F3(1,5) = -X(10)*SP(1)
F3(2,5) = -X(9)*SP(1)
F3(3,5) = X(8)*SP(1)
F3(4,5) = X(7)*SP(1)
F1(1,7) = 2.*[X(7)*AB(1)-X(10)*AB(2) + X(9)*AB(3)]
F1(1,8) = 2.*[X(8)*AB(1) + X(9)*AB(2) + X(10)*AB(3)]
F1(1,9) = 2.*[-X(9)*AB(1) + X(8)*AB(2) + X(7)*AB(3)]
F1(1,10) = 2.*[X(10)*AB(1) + X(7)*AB(2) - X(8)*AB(3)]
F1(2,7) = -F1(1,10)
F1(2,8) = -F1(1,9)
F1(2,9) = F1(1,8)
F1(2,10) = F1(1,7)
F1(3,7) = F1(1,9)
F1(3,8) = -F1(1,10)
F1(3,9) = -F1(1,7)
F1(3,10) = F1(1,8)
F3(2,7) = -F3(1,8)
F3(3,7) = -F3(1,9)
F3(3,8) = -F3(2,9)
F3(4,7) = -F3(1,10)
F3(4,8) = -F3(2,10)
F3(4,9) = -F3(3,10)
F2(1,3) = -1.
```

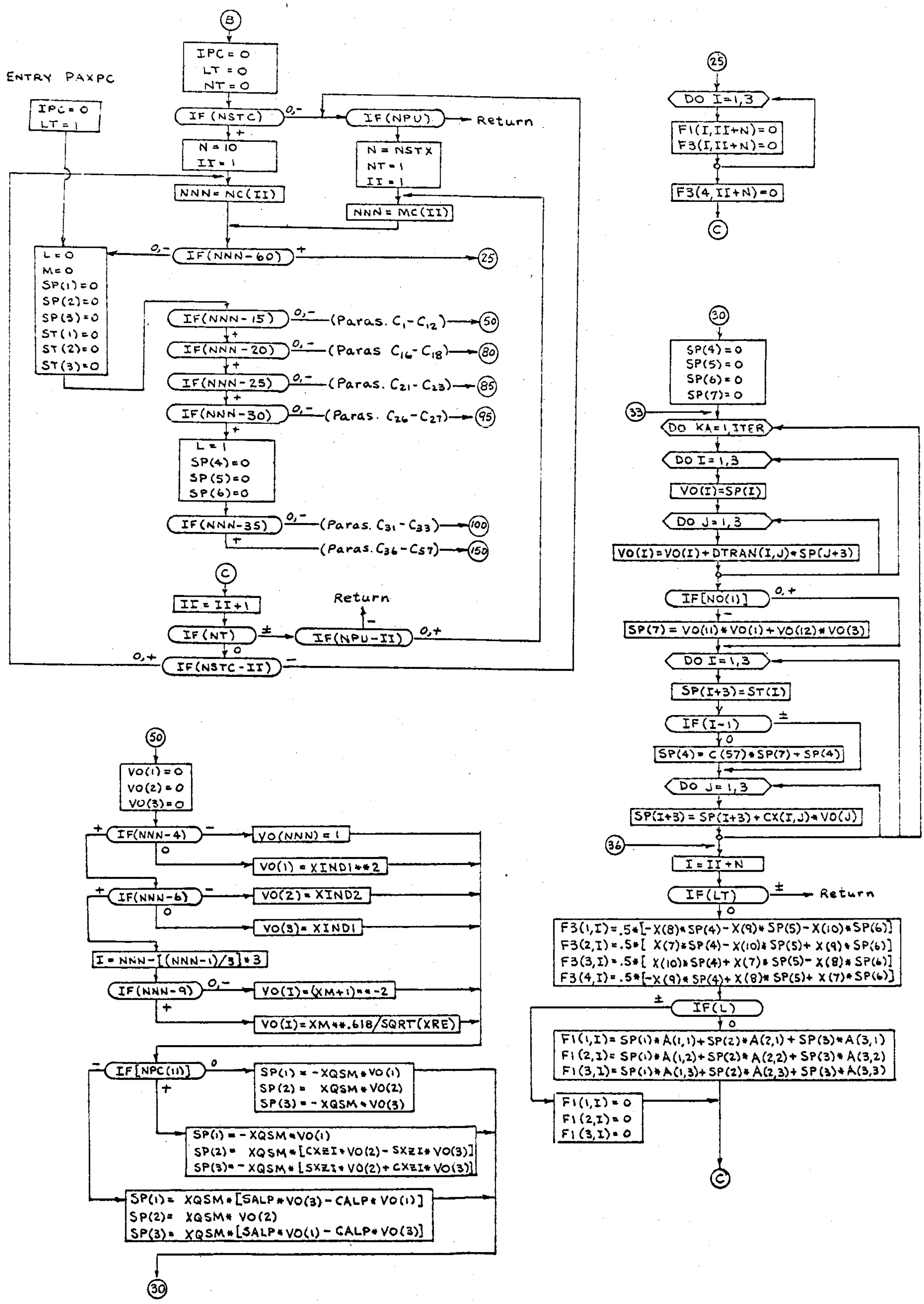
```
F1(1,6) = 0.
F1(2,6) = 0
F1(3,6) = 0
F1(3,3) = 0
F2(1,1) = 0
F2(1,2) = 0
F2(1,4) = 0
F2(1,5) = 0
F2(2,2) = 0
F2(2,3) = 0.
F2(2,5) = 0
F2(3,1) = 0
F2(3,3) = 0
F3(1,3) = 0
F3(2,3) = 0
F3(3,3) = 0
F3(4,3) = 0
F3(1,6) = 0
F3(2,6) = 0
F3(3,6) = 0
F3(4,6) = 0
F3(1,7) = 0
F3(2,8) = 0
F3(3,9) = 0
F3(4,10) = 0
```

(A)

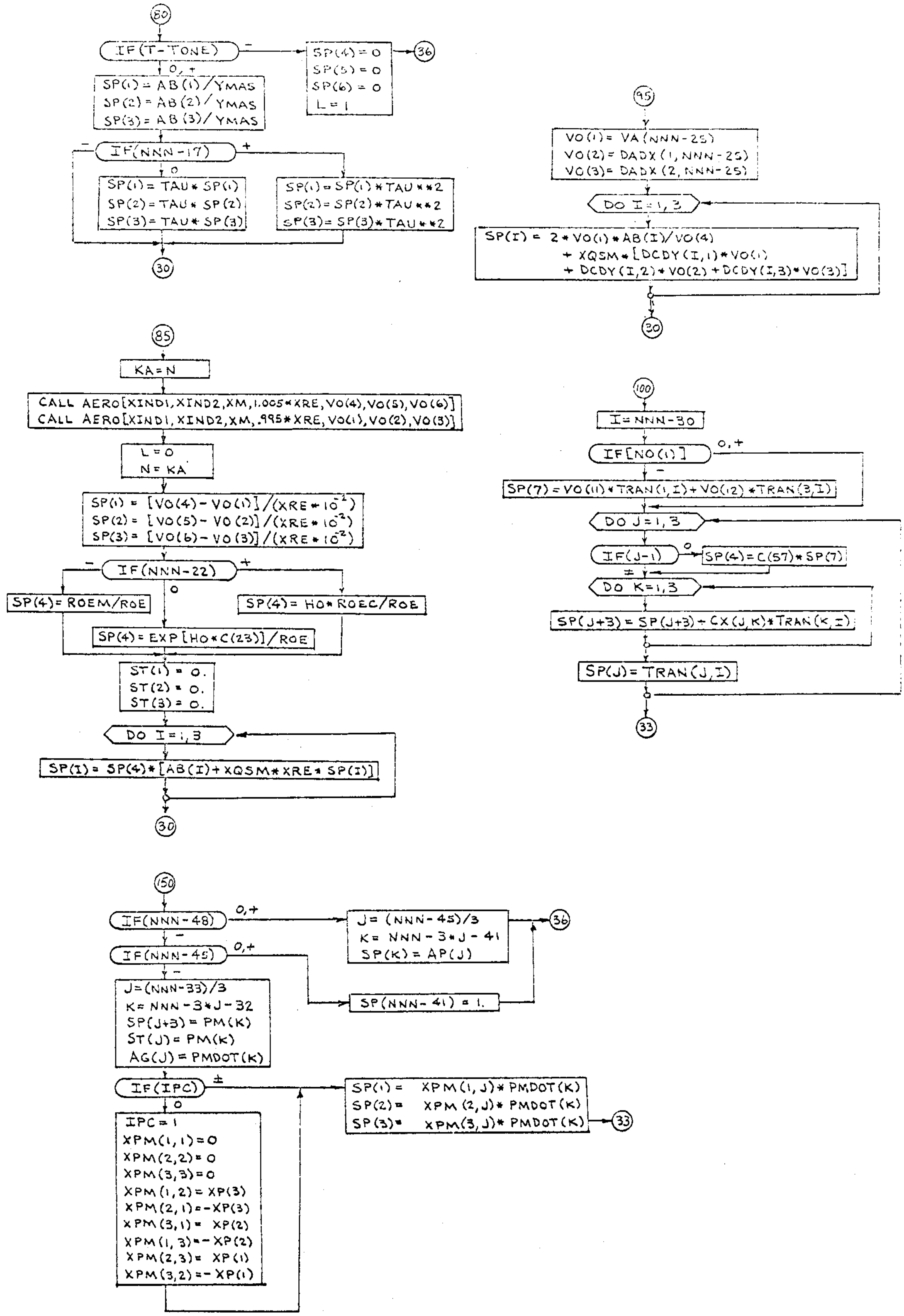
Subroutine FXXU (STEP1, STEP2)



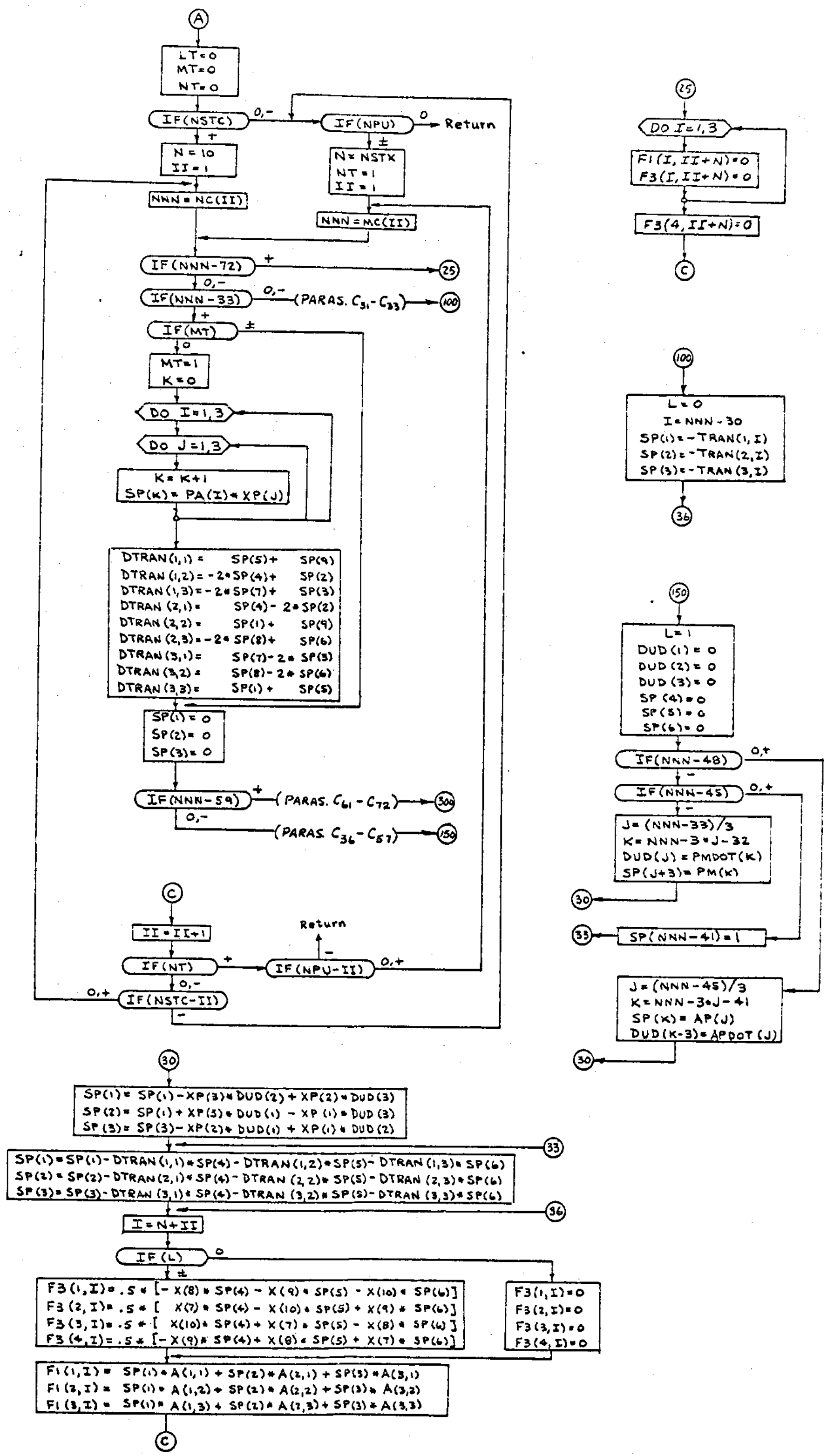
Subroutine FXXU (STEP1) - Continued



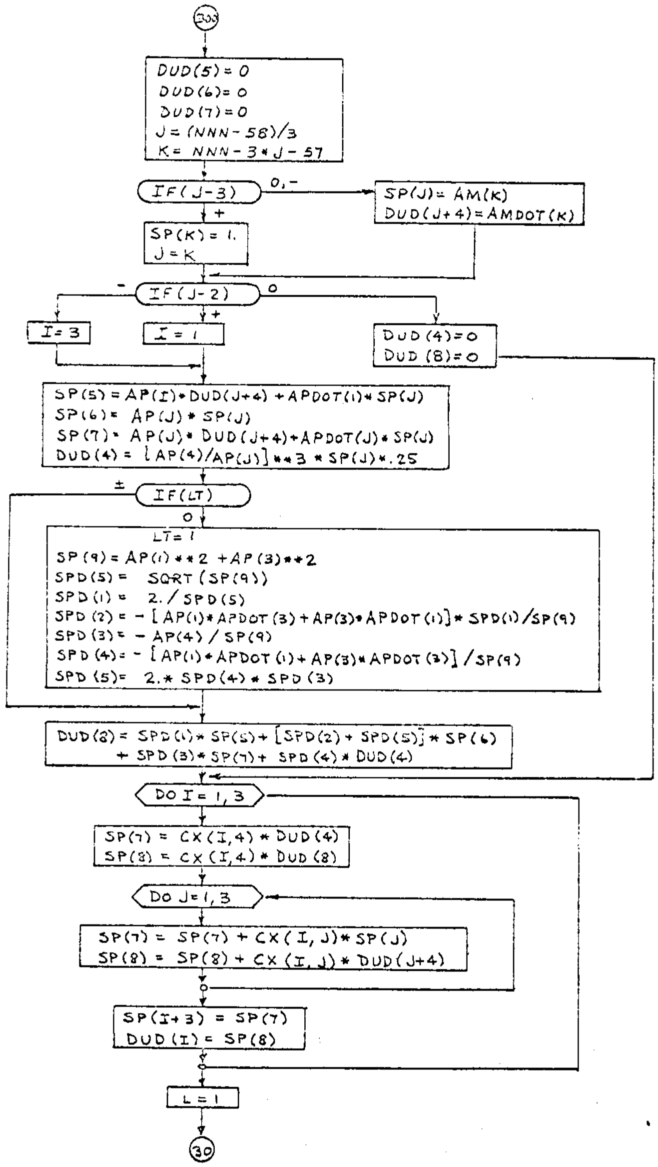
Subroutine FXXU (STEP1) - Continued



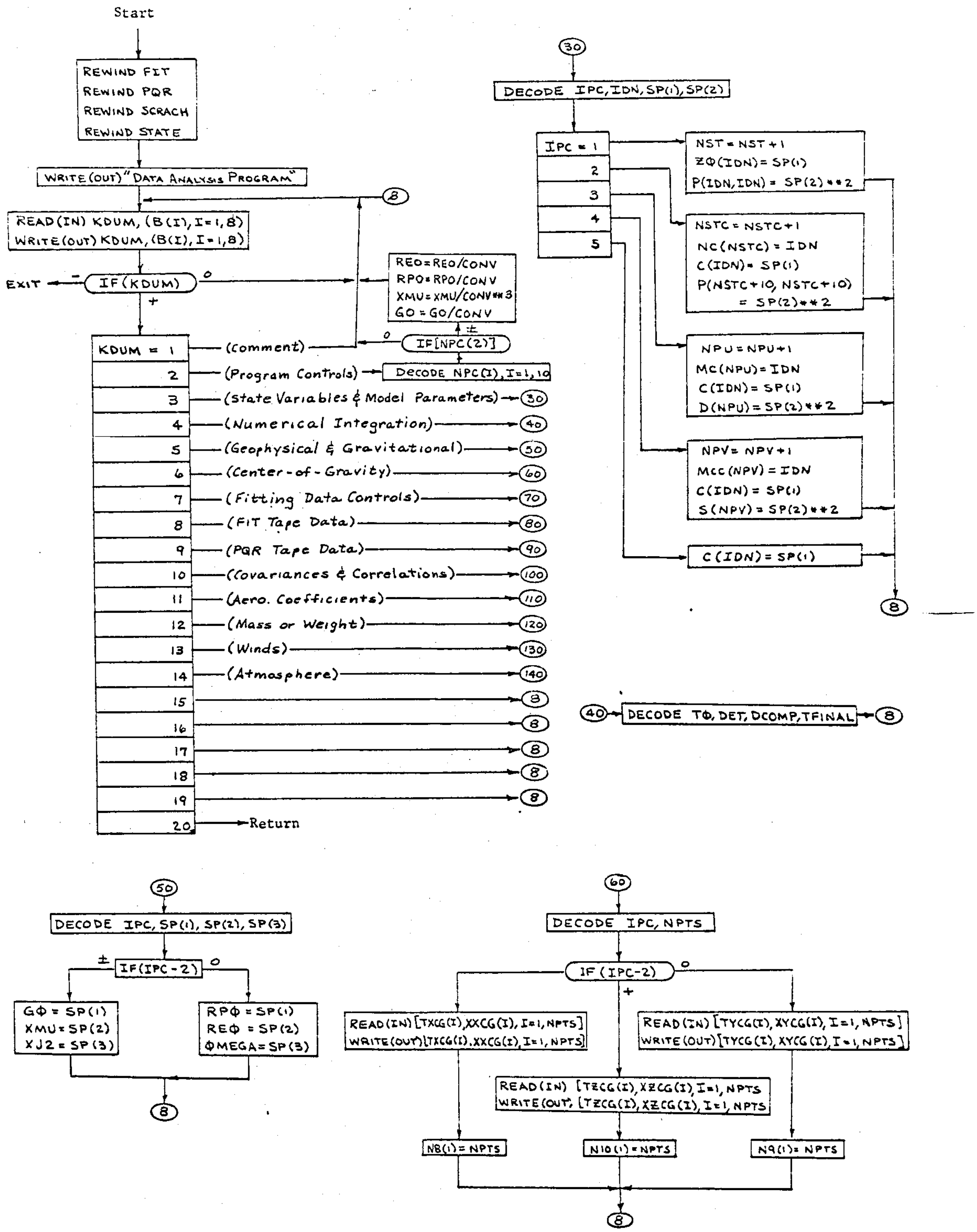
Subroutine FXXU (STEP1) - Continued



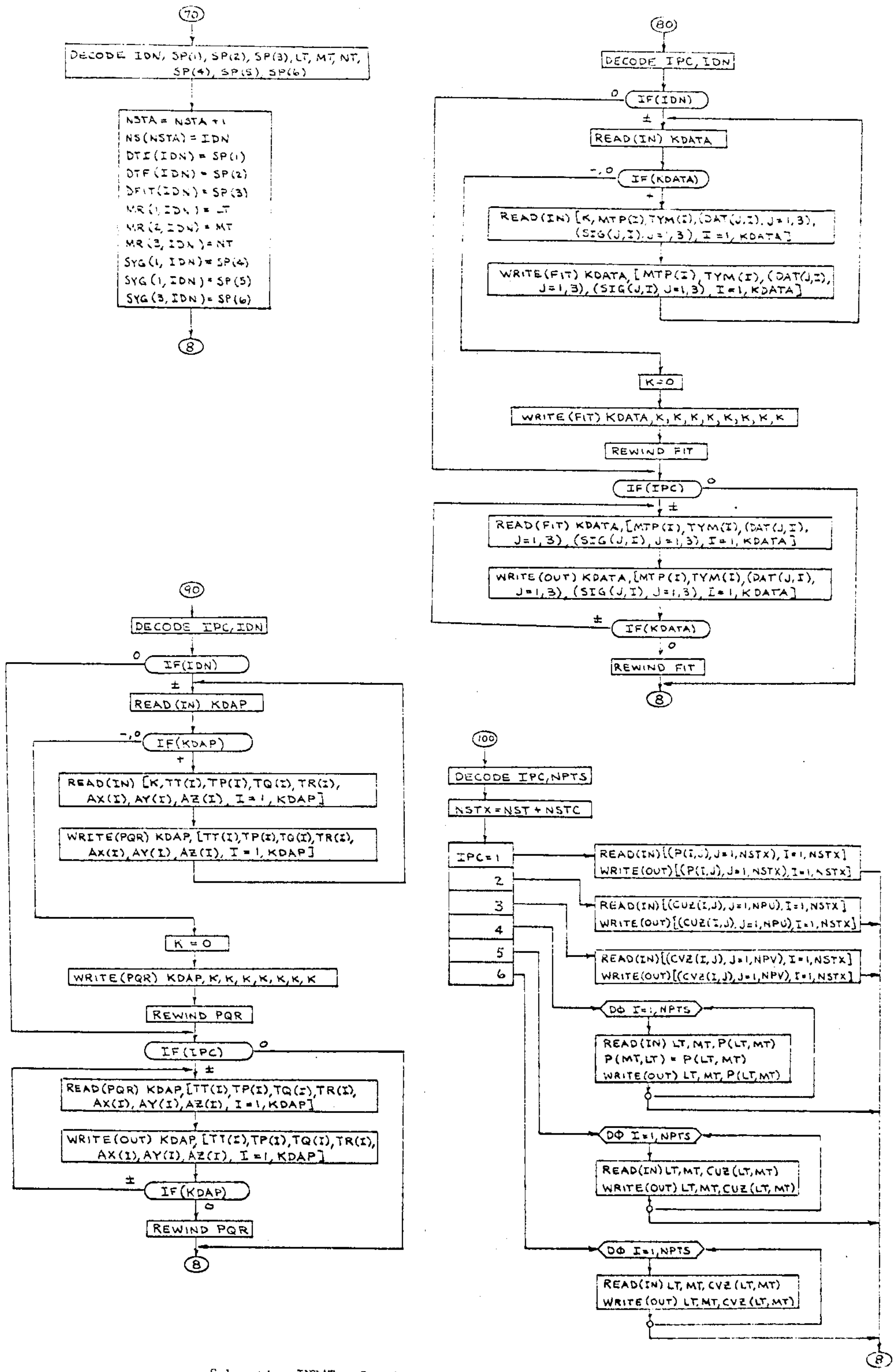
Subroutine FXXU (STEP2) - Continued



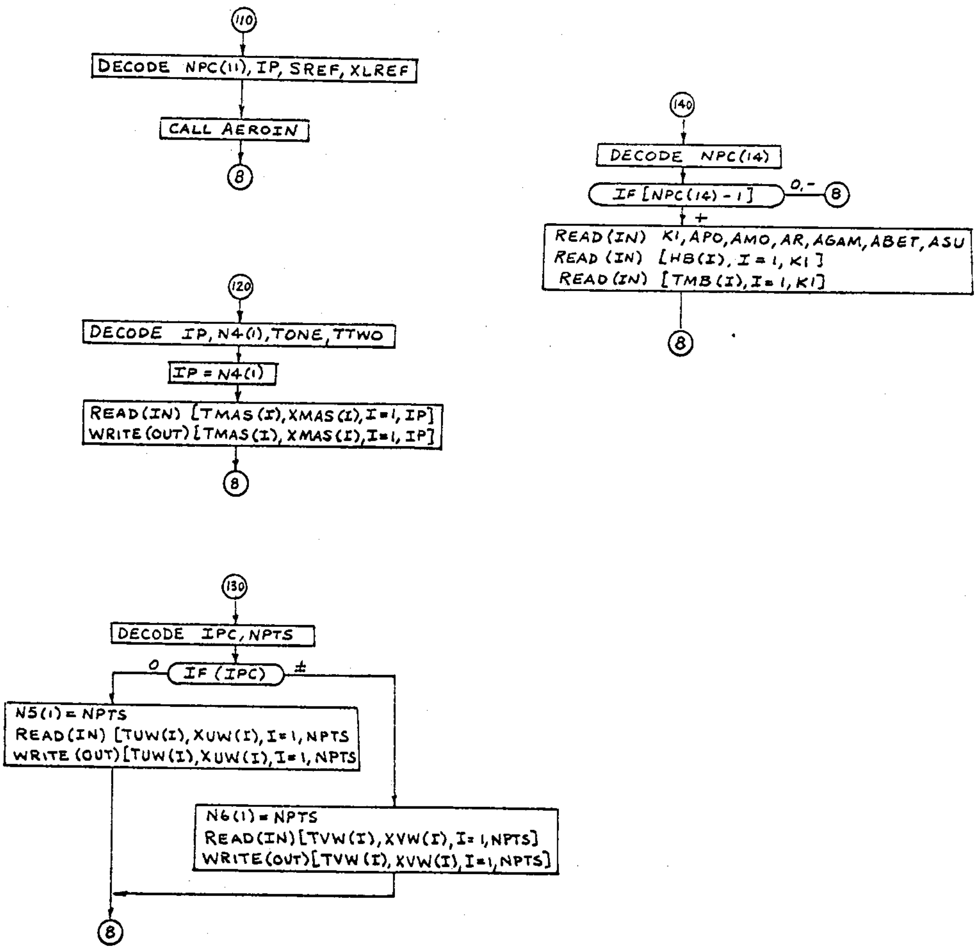
Subroutine FXXU (STEP2) - Concluded



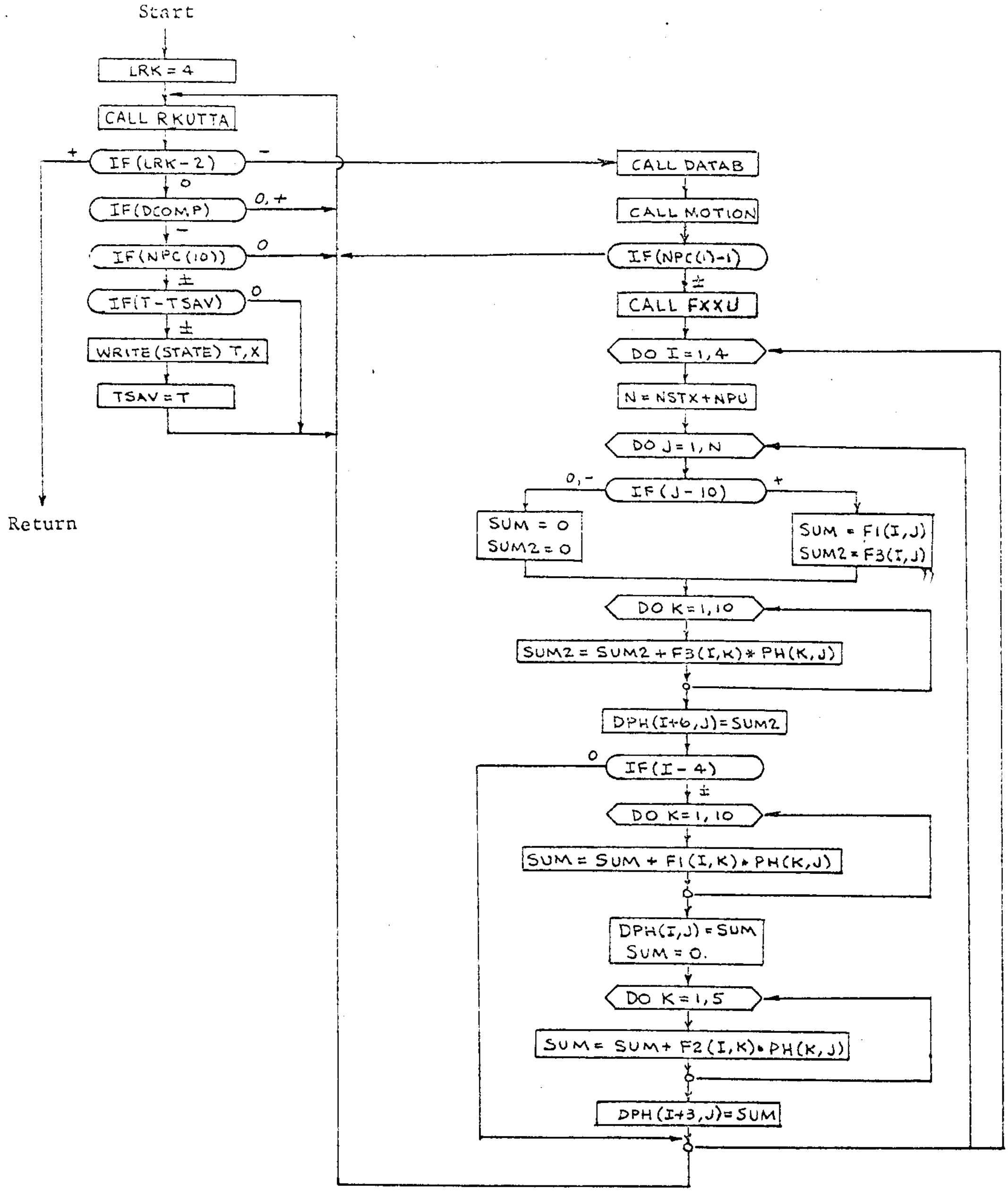
Subroutine INDAT



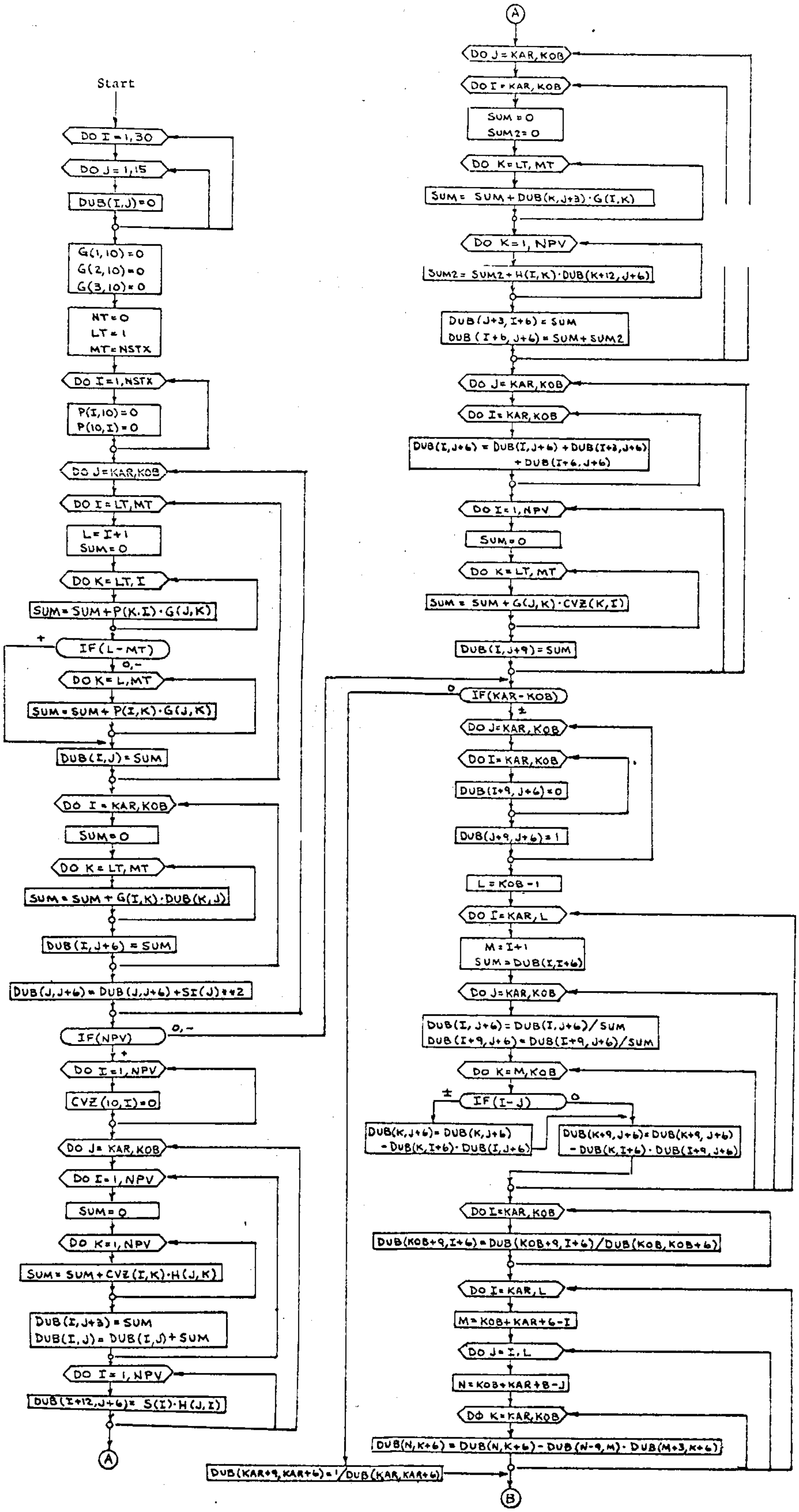
Subroutine INDAT - Continued



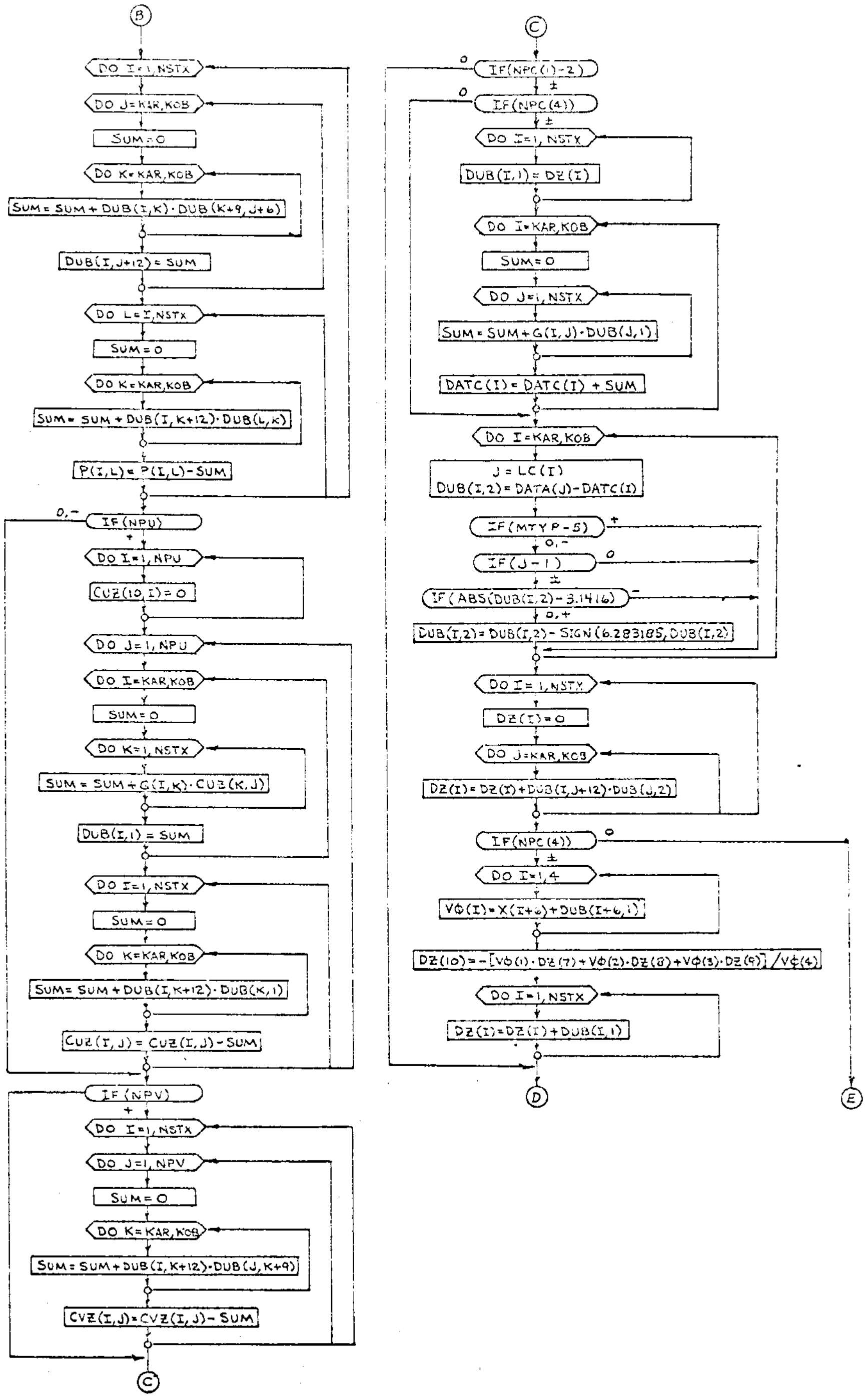
Subroutine INDAT - Concluded



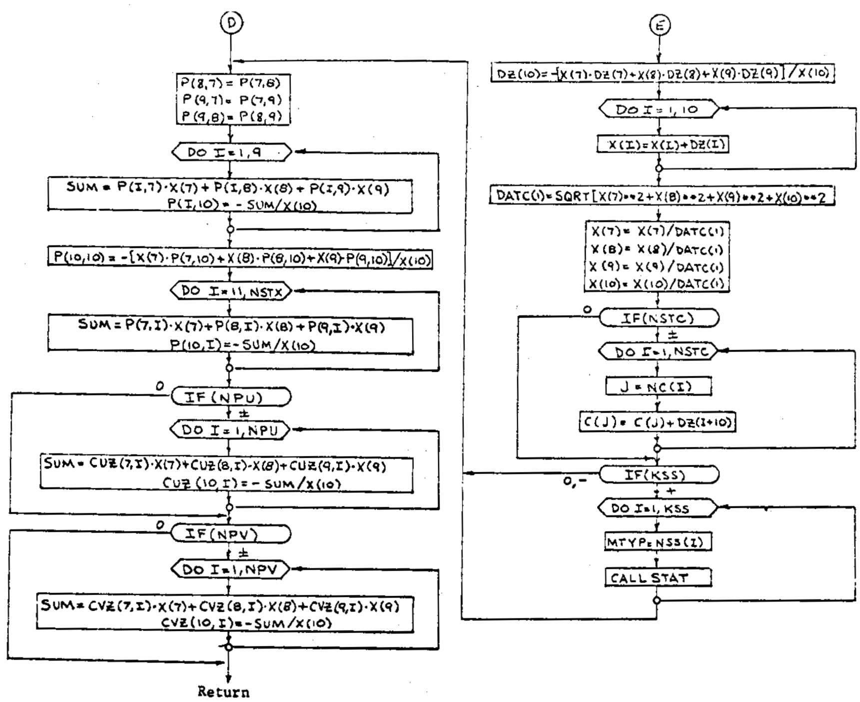
Subroutine INTAG



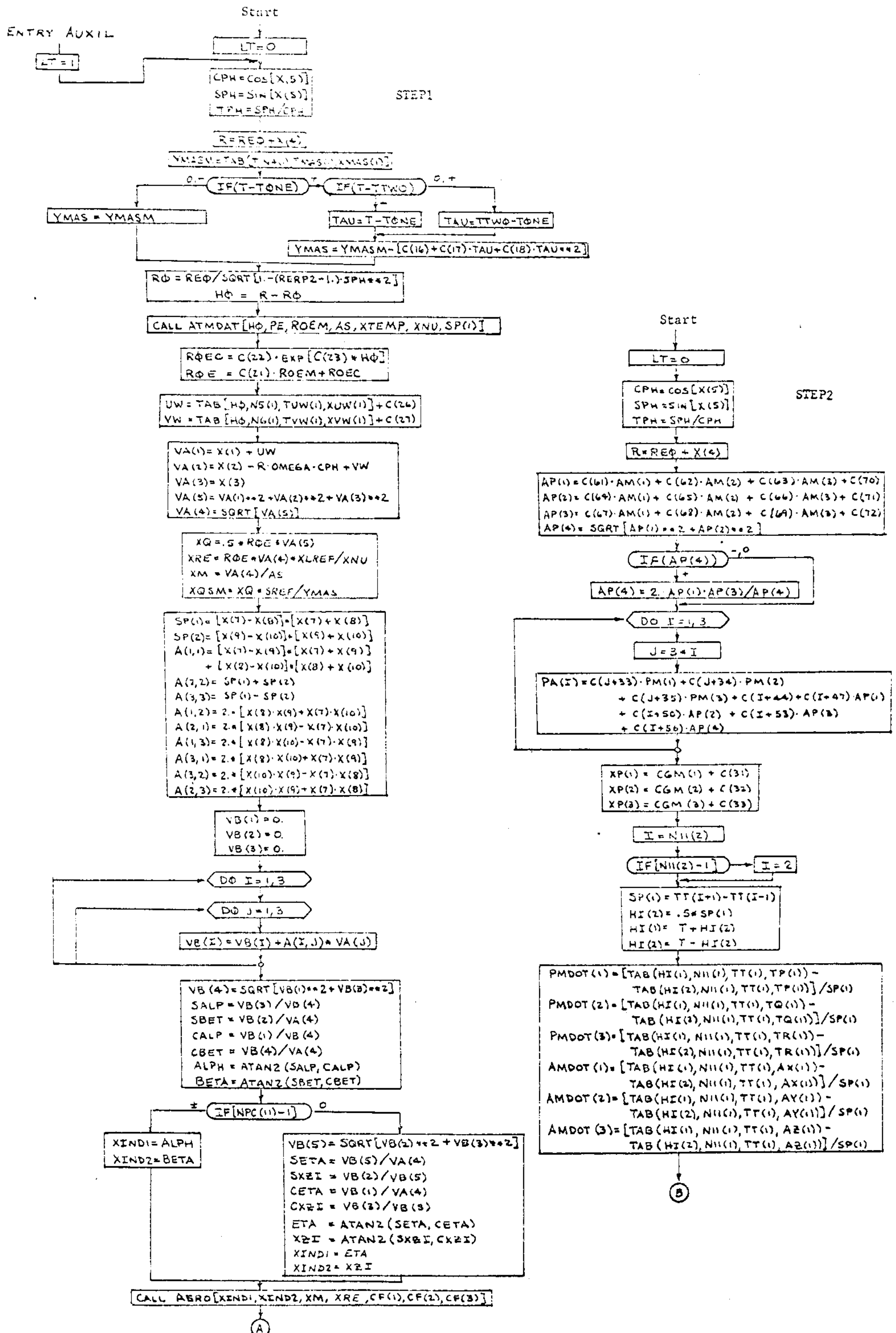
Subroutine MINVAR



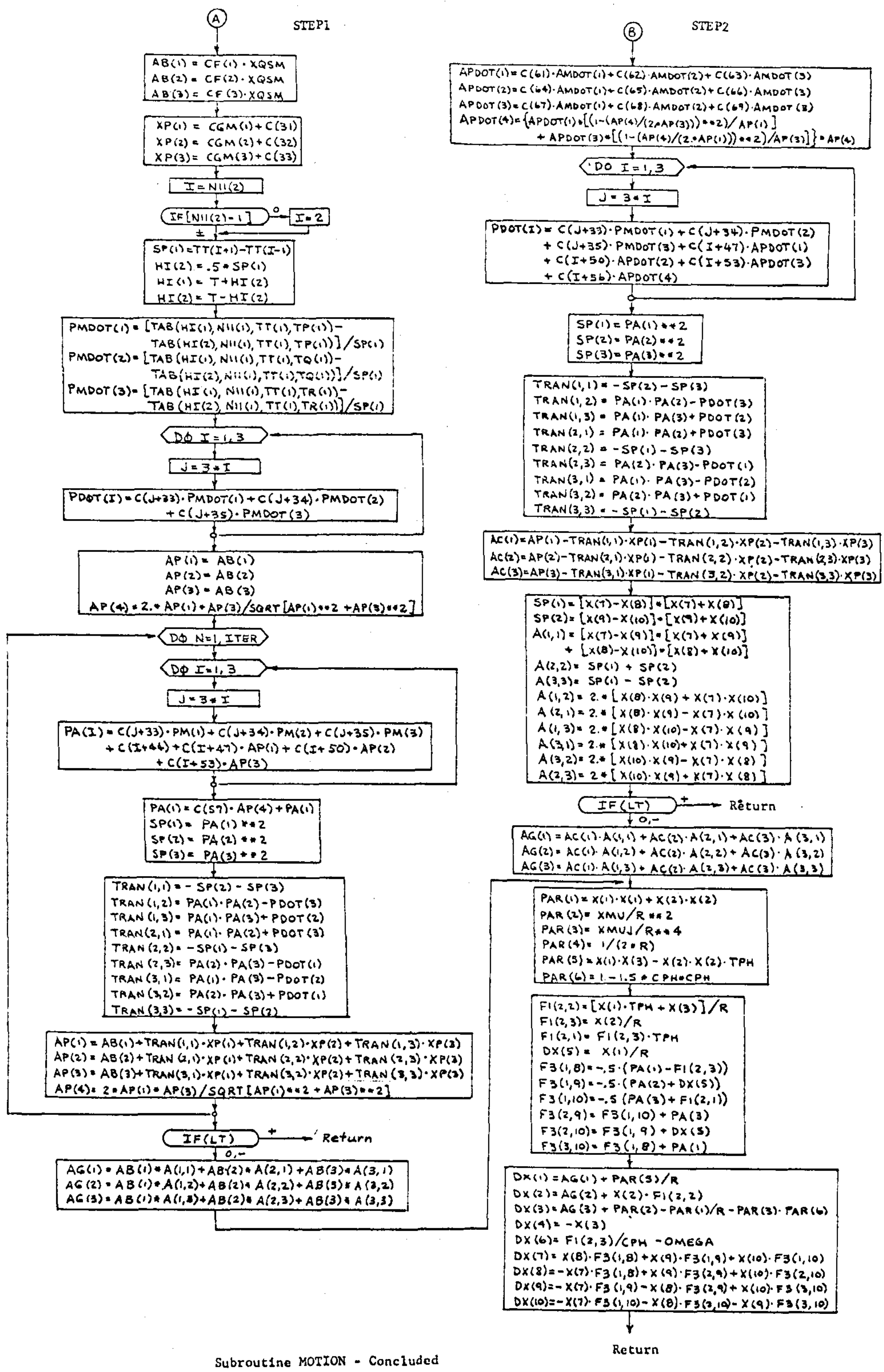
Subroutine MINVAR - Continued



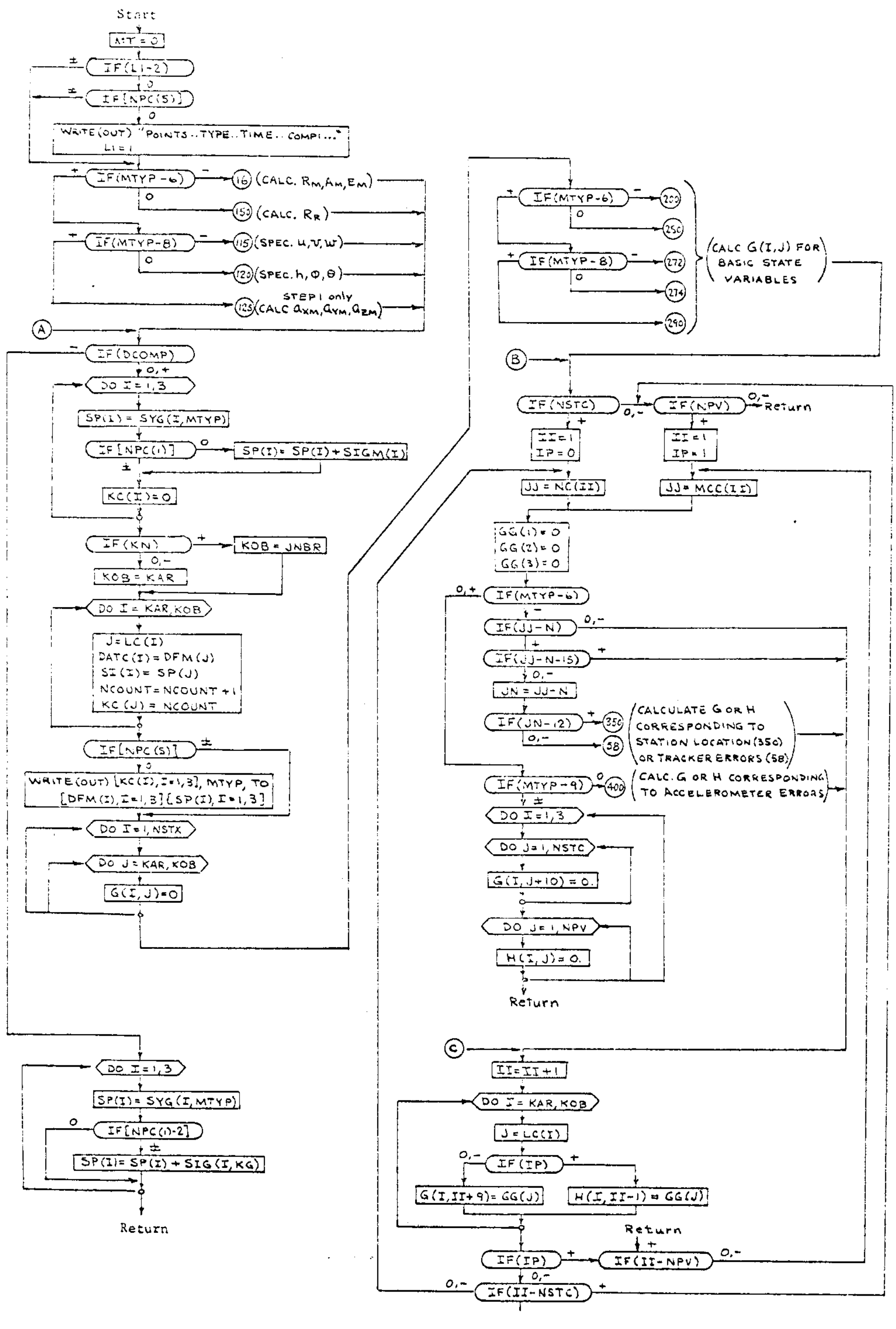
Subroutine MINVAR - Concluded



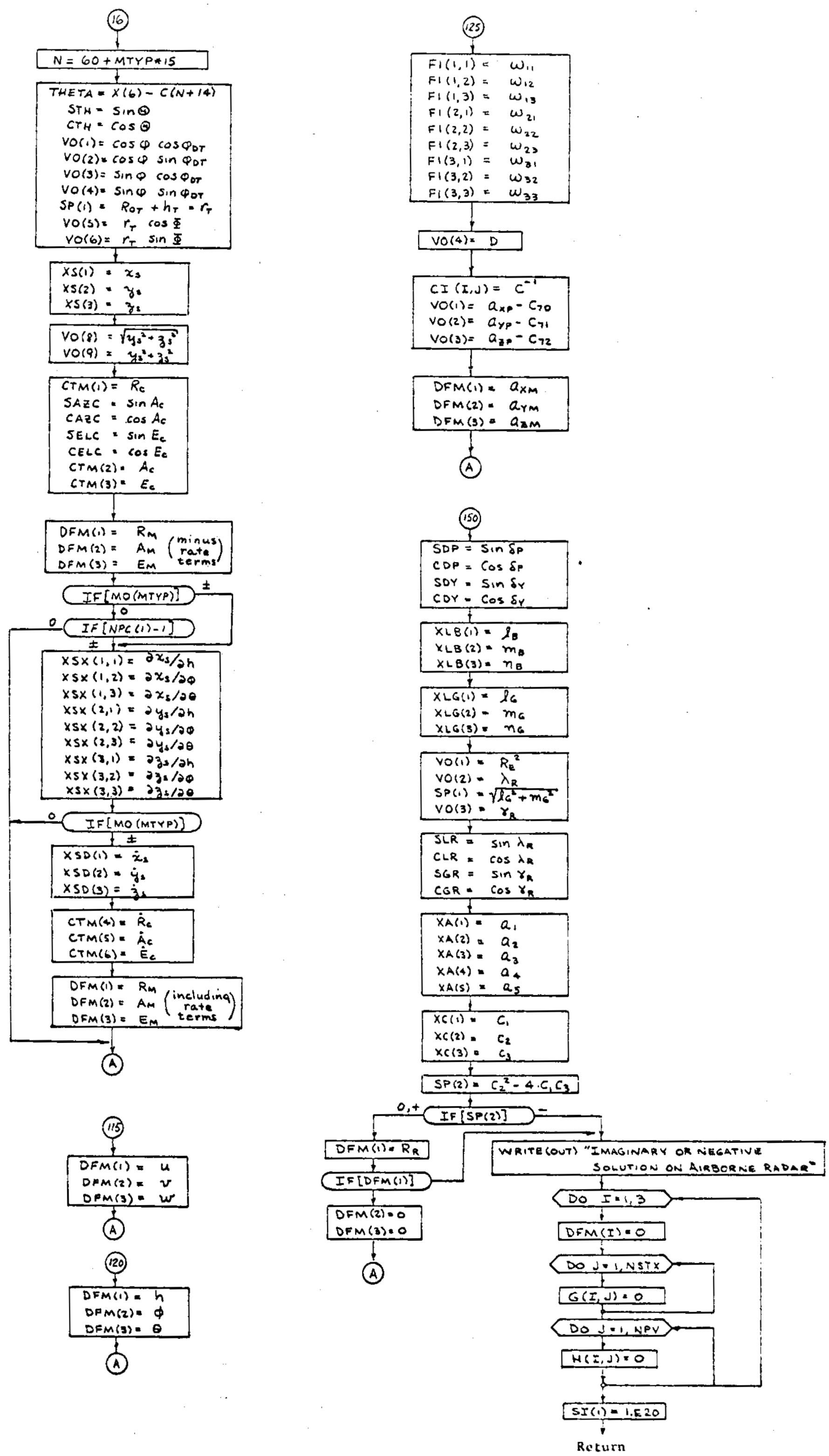
Subroutine MOTION



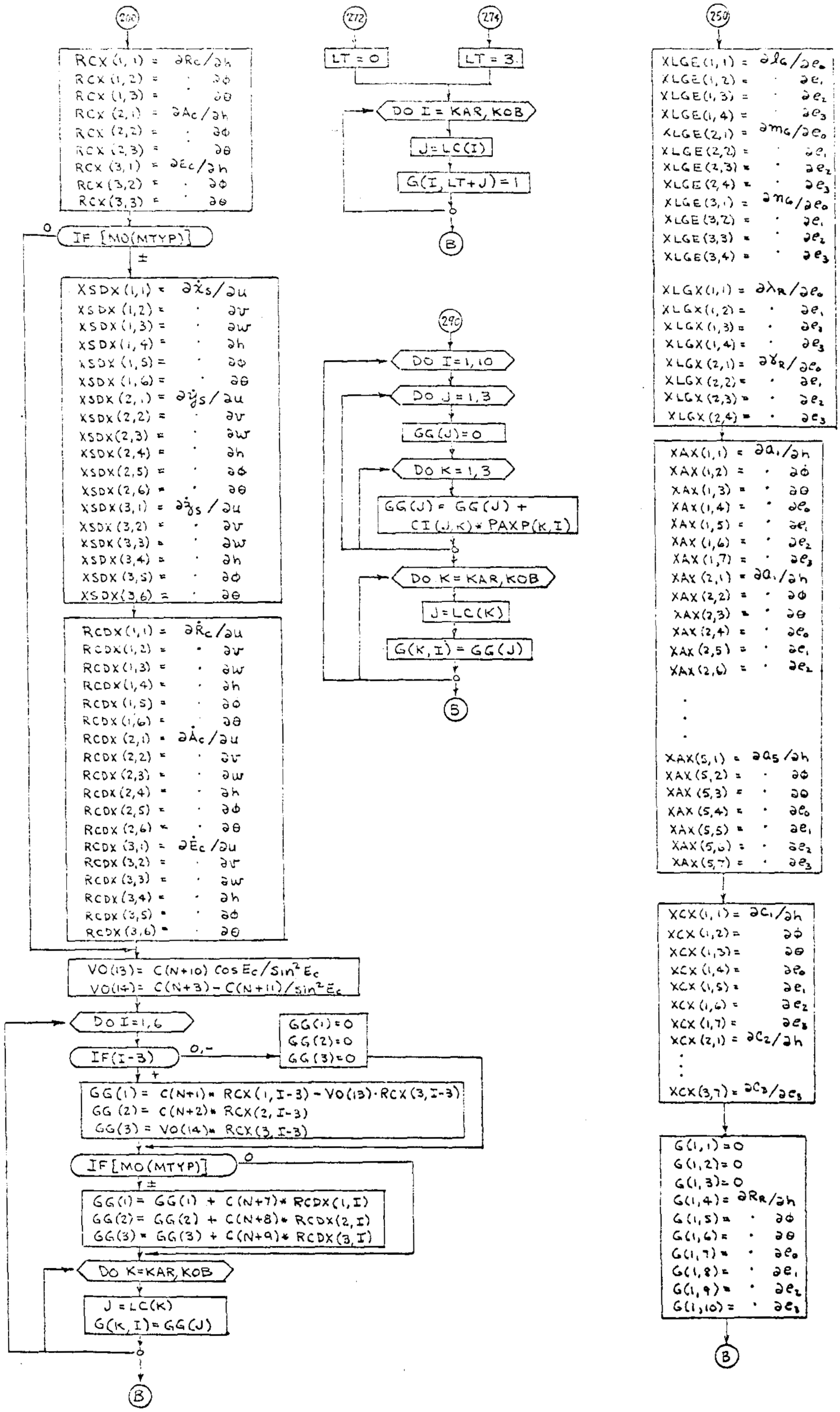
Subroutine MOTION - Concluded



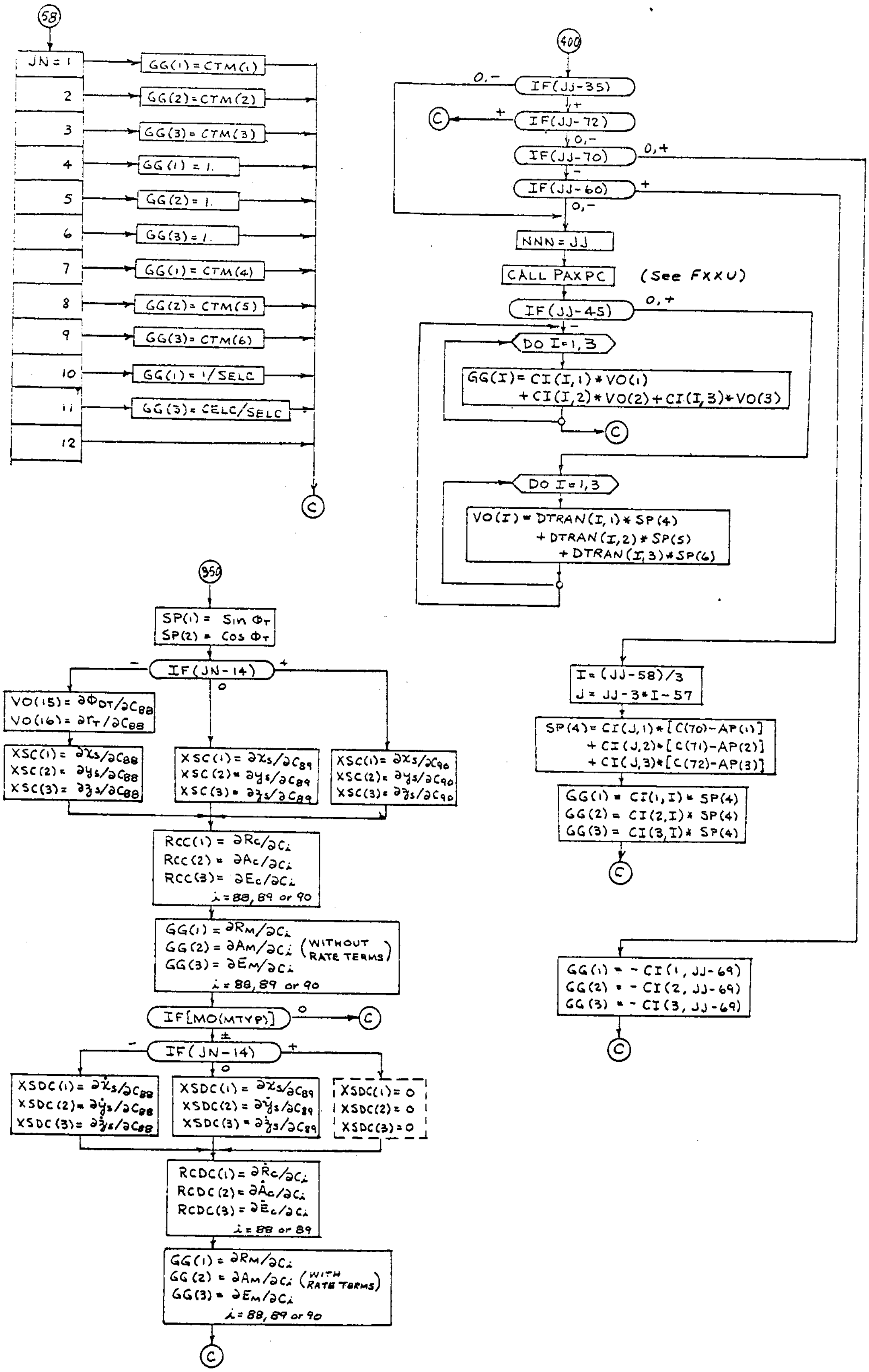
Subroutine OBSERV



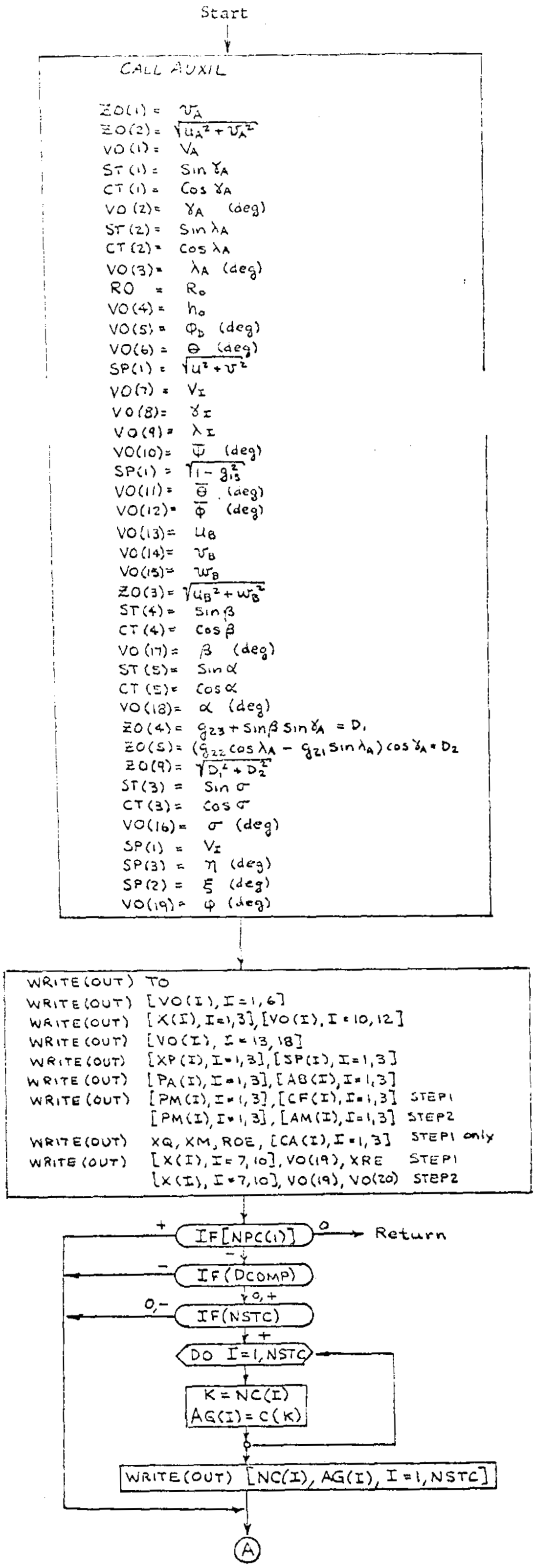
Subroutine OBSERV - Continued



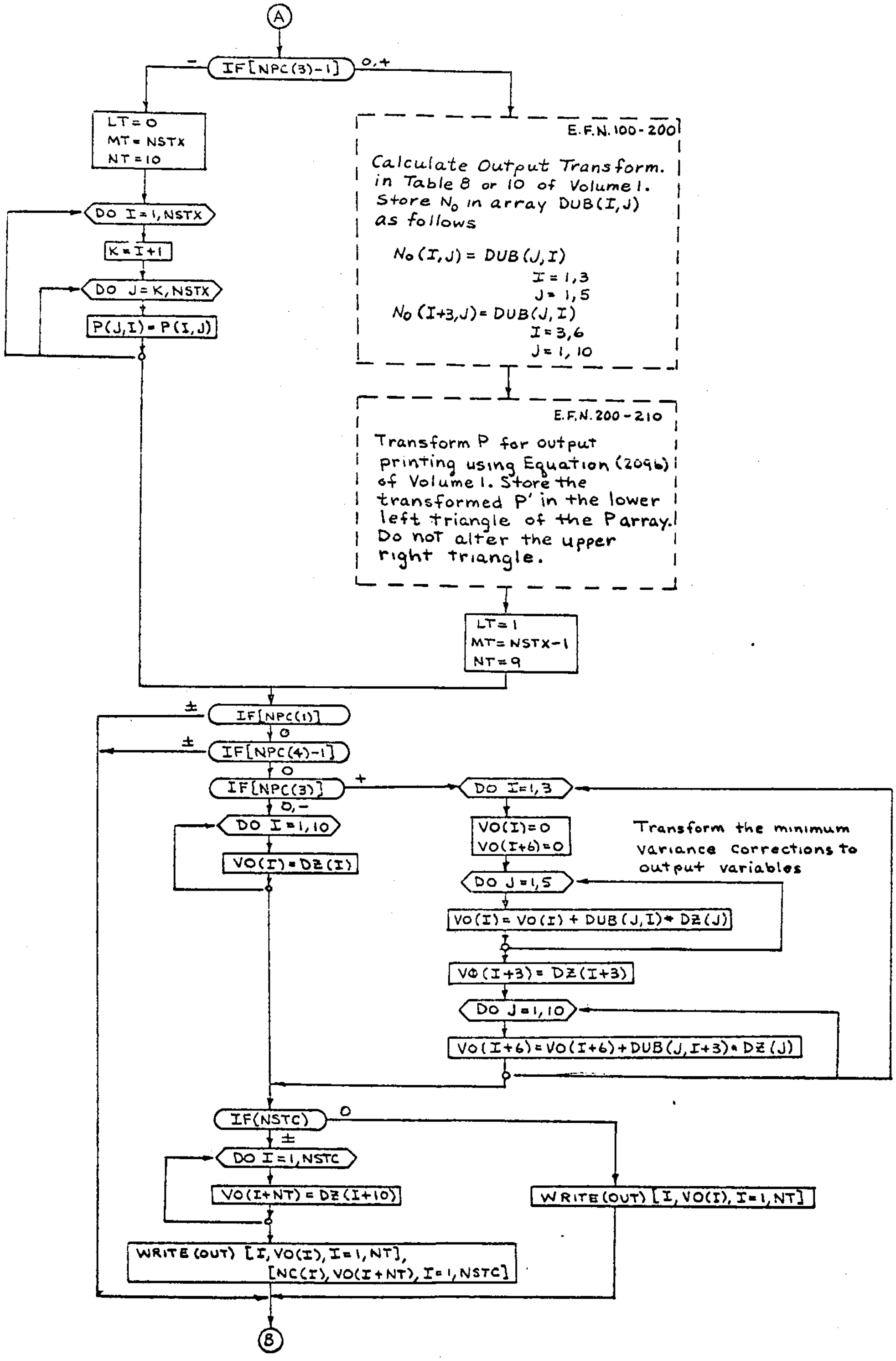
Subroutine OBSERV - Continued



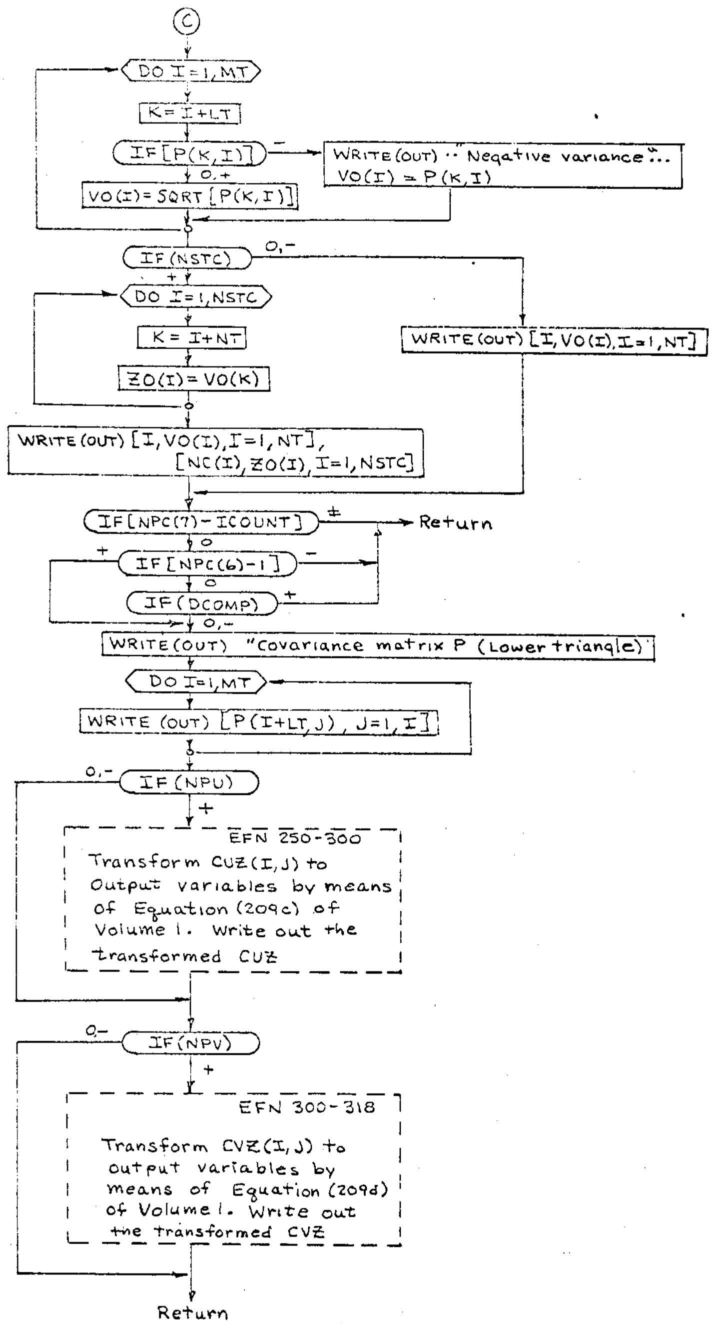
Subroutine OBSERV - Concluded



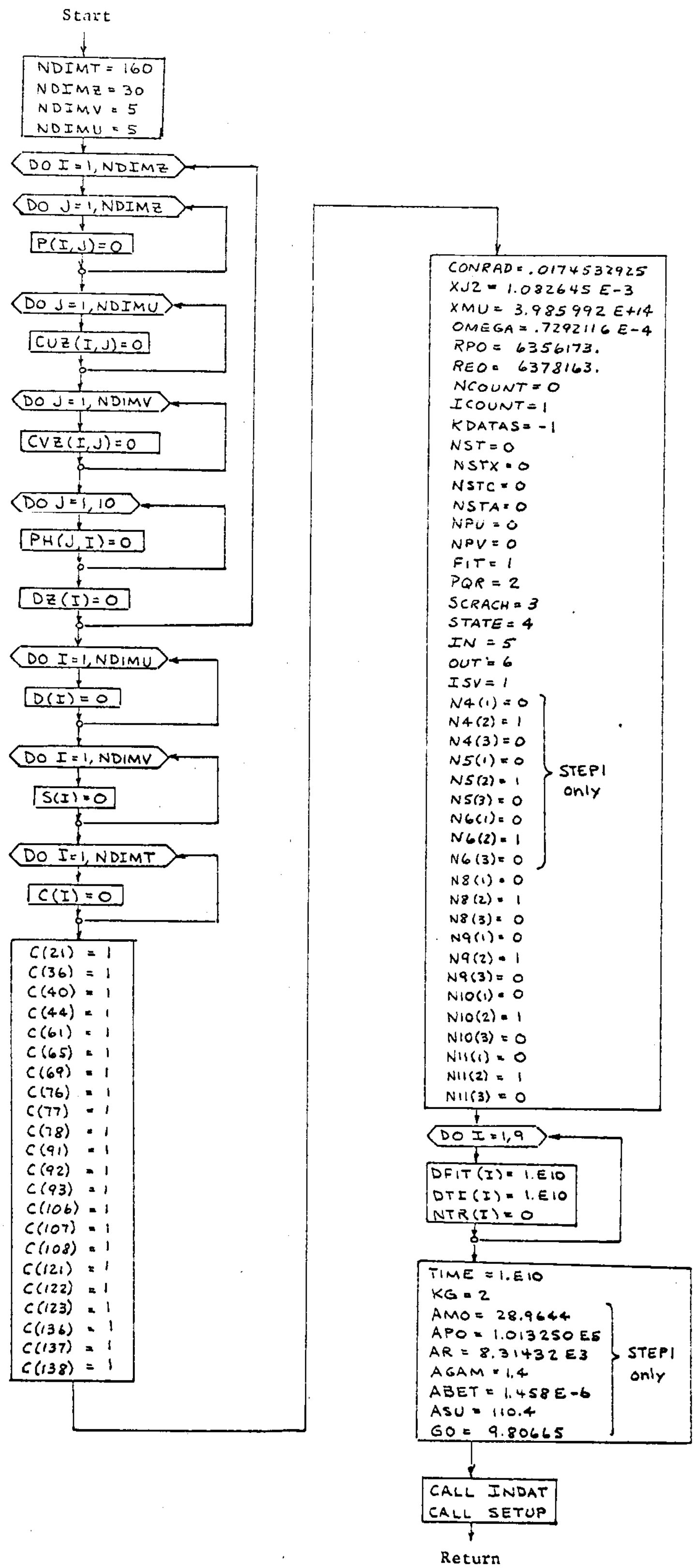
Subroutine OUTPUT



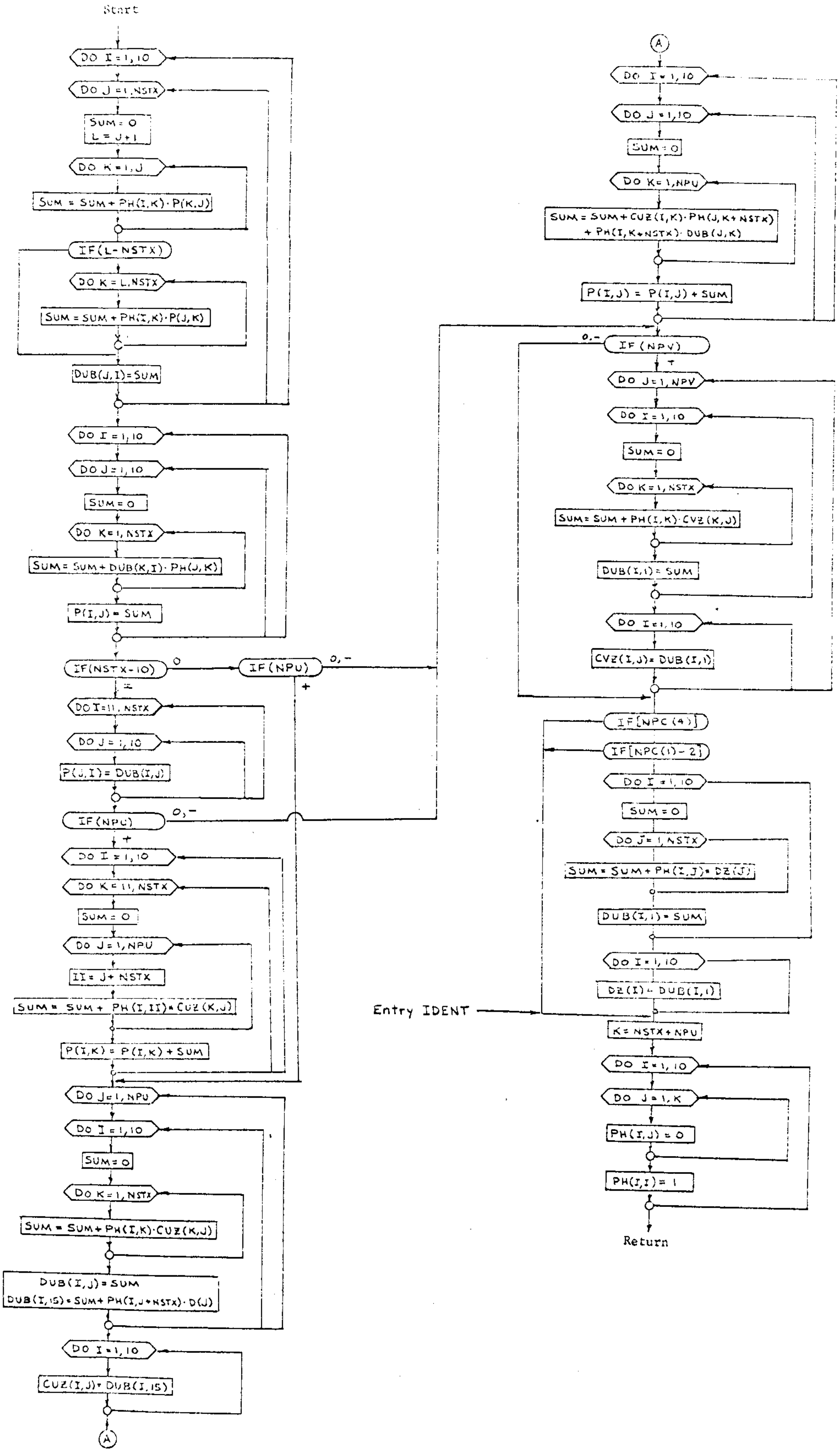
Subroutine OUTPUT - Continued



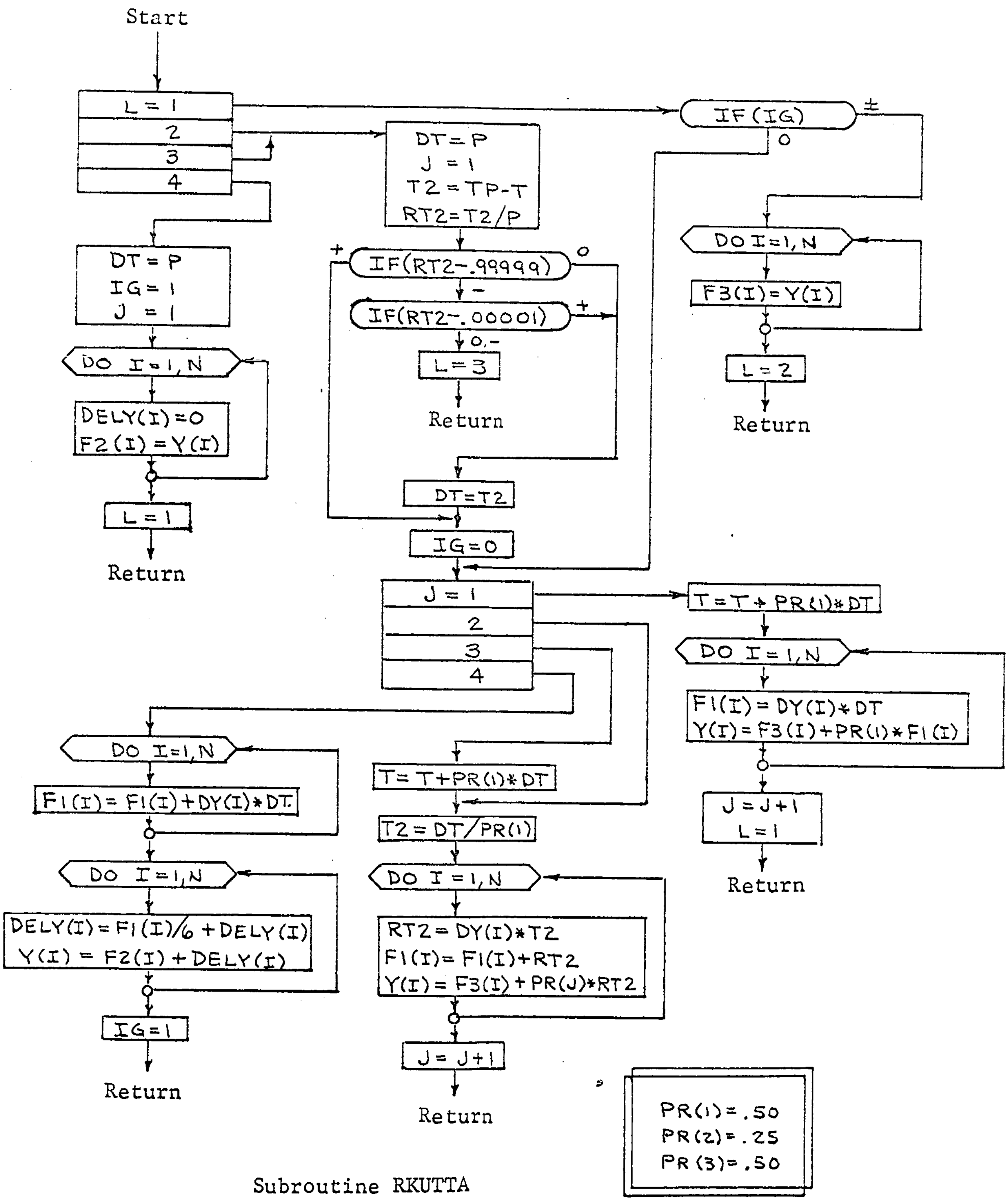
Subroutine OUTPUT - Concluded



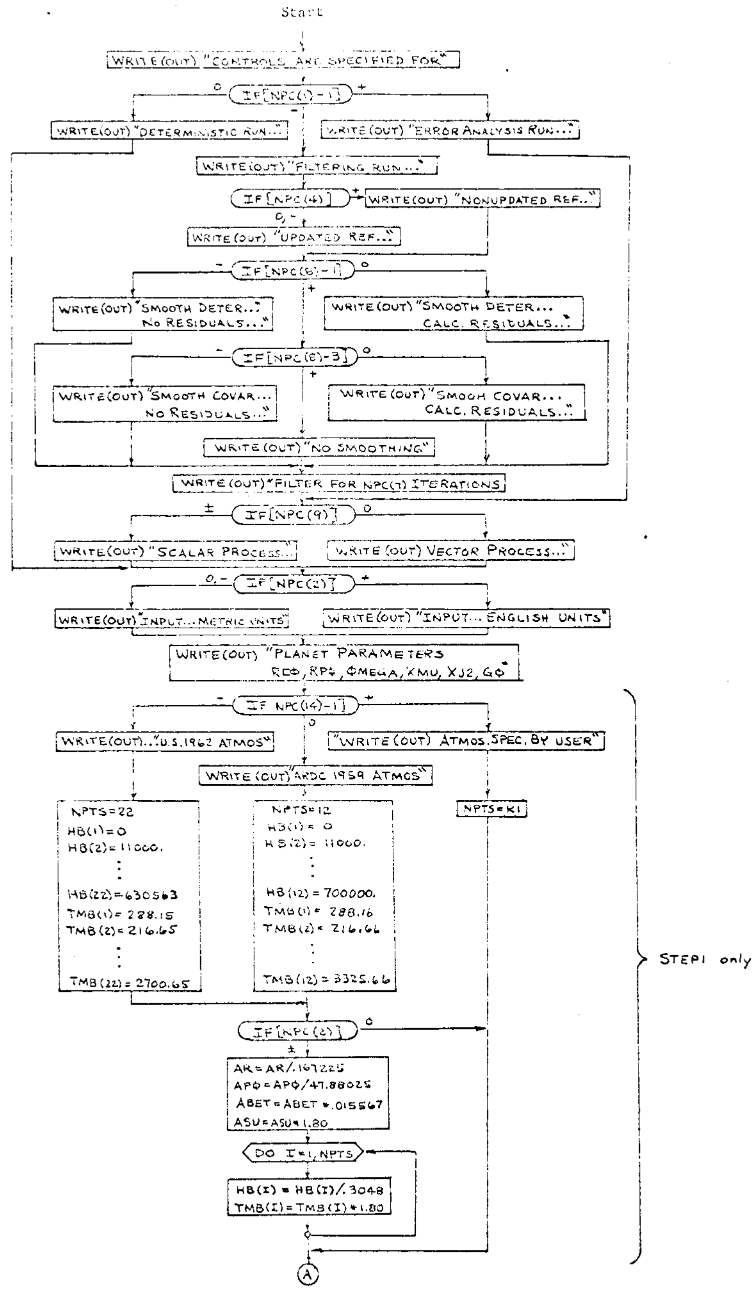
Subroutine PRESET



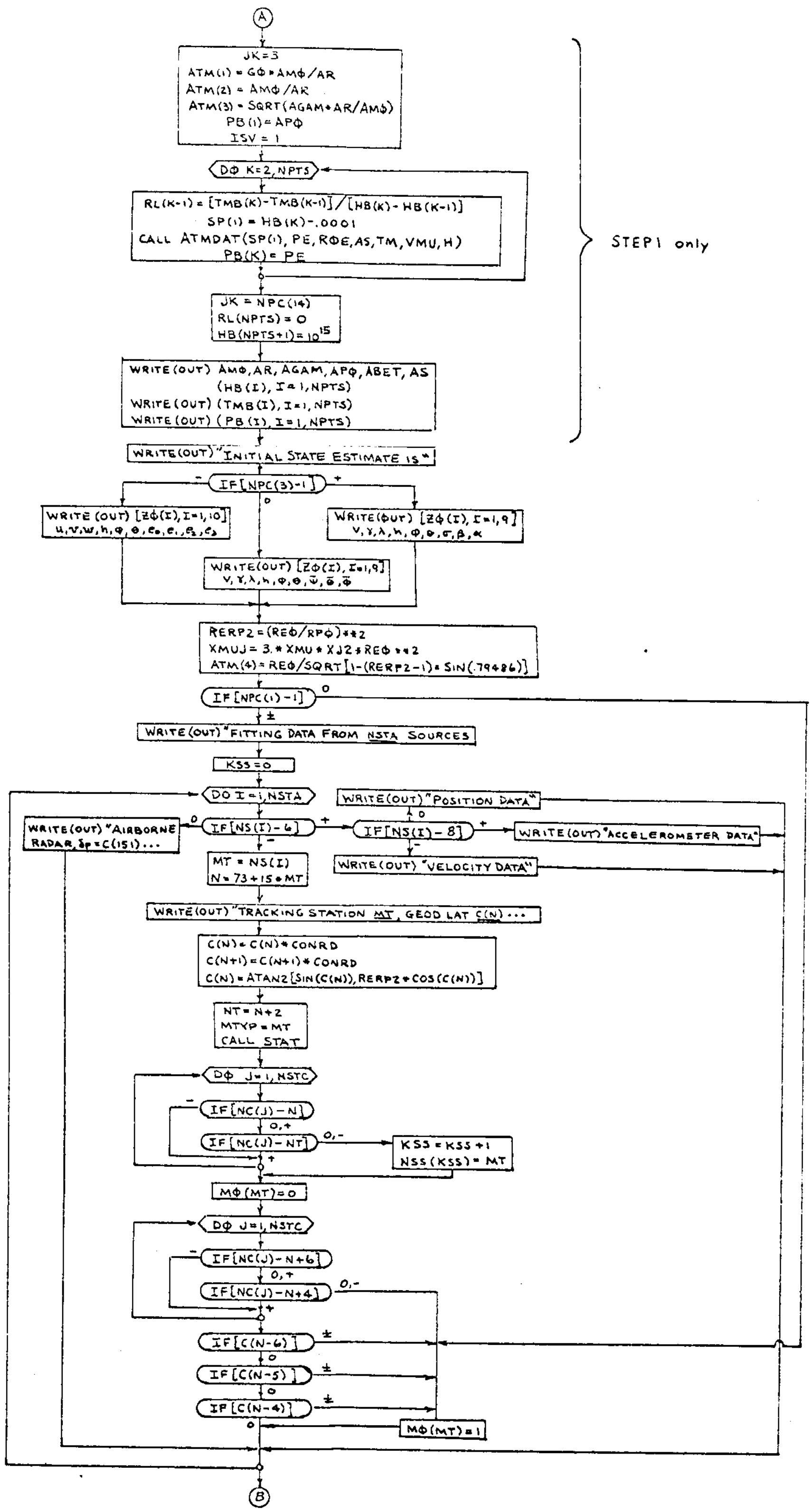
Subroutine PROP



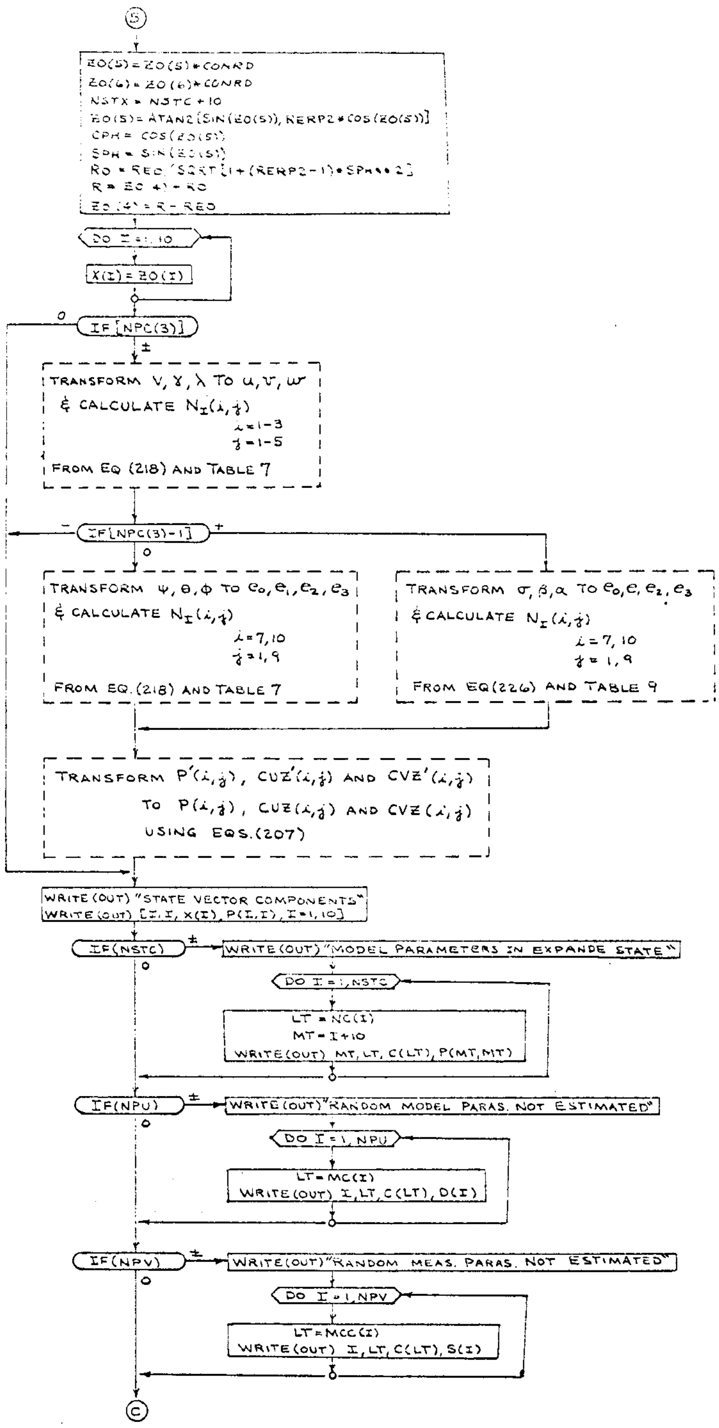
Subroutine RKUTTA



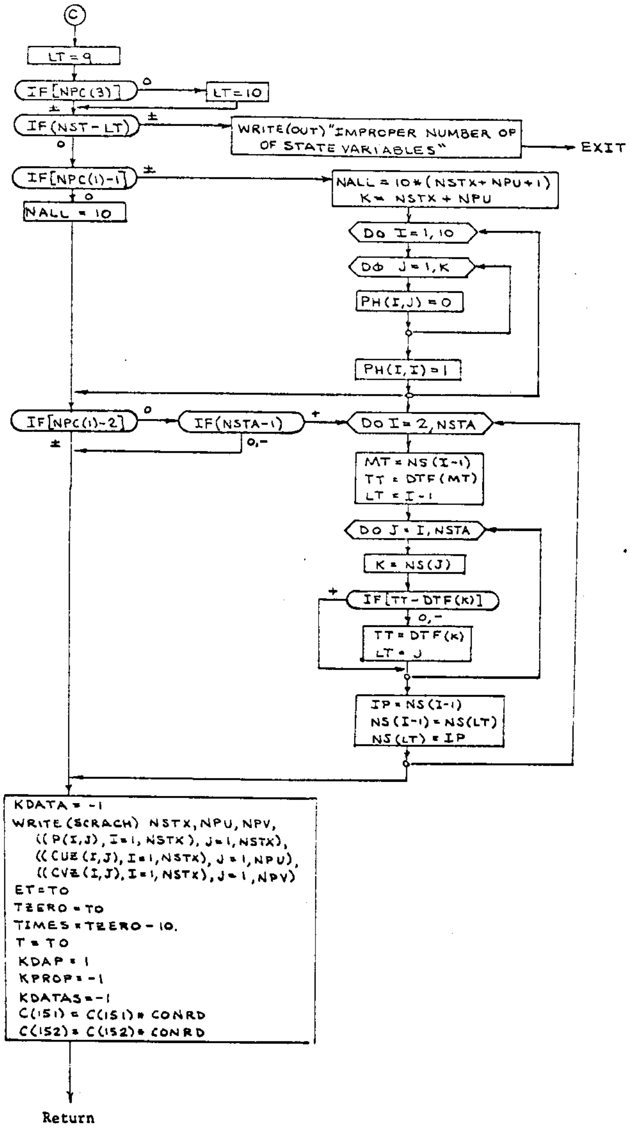
Subroutine SETUP



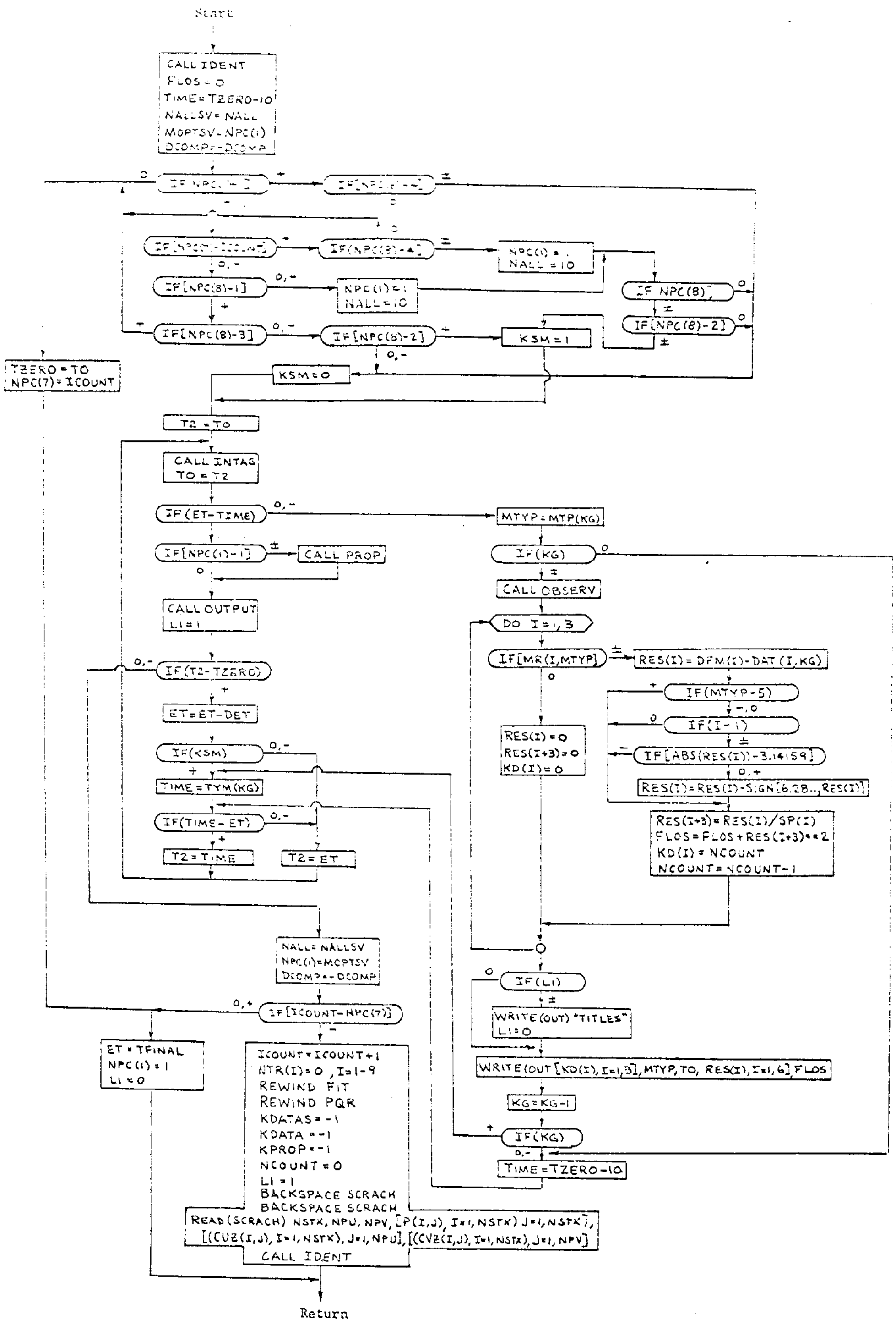
Subroutine SETUP - Continued



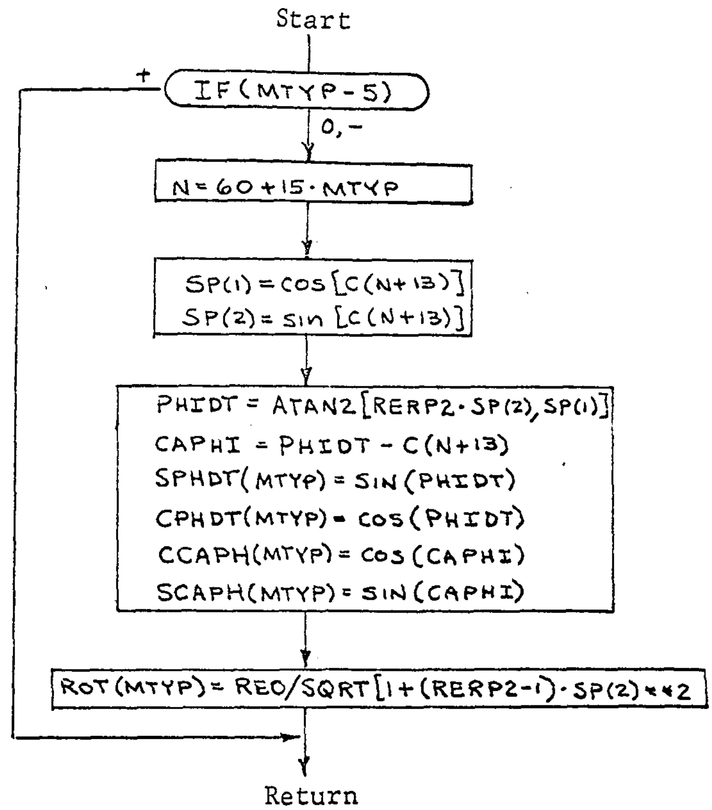
Subroutine SETUP - Continued



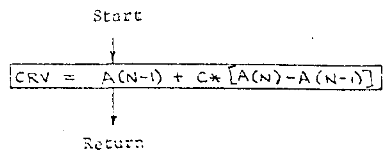
Subroutine SETUP - Concluded



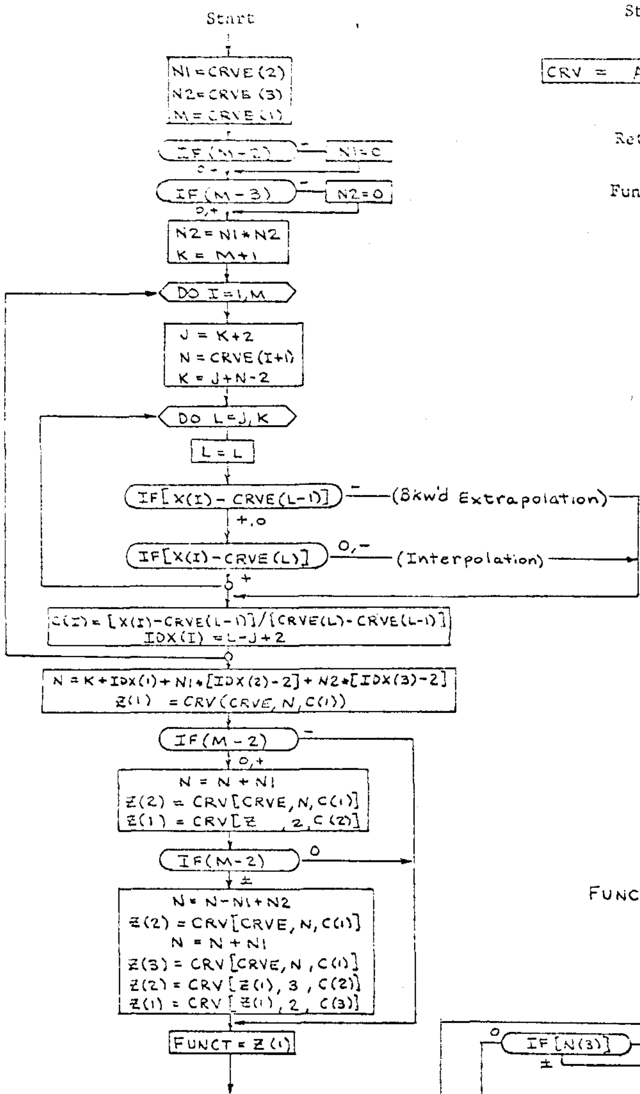
Subroutine SMOOTH



Subroutine STAT

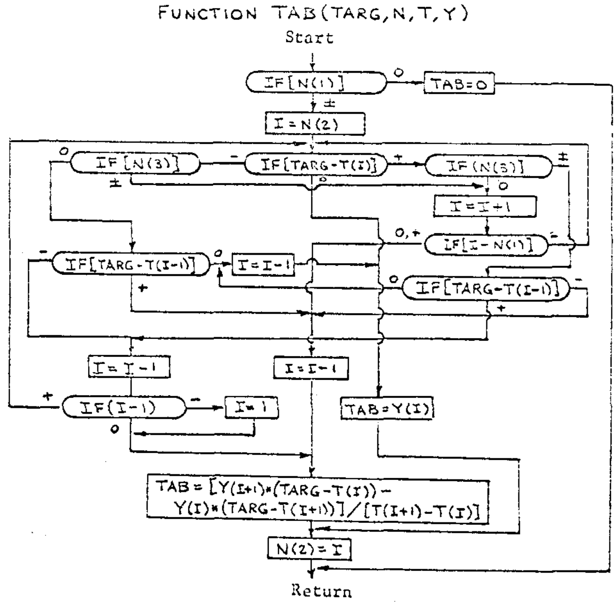


Function CRV (A, N, C)



Function FUNCT (CRVE, X)

FUNCTION TAB(TARG,N,T,Y)



Function TAB (TARG, N, T, Y)

VII. SOURCE LISTING

A. STEP1 Listing

The STEP1 source listing is presented in the following subsection. The following listing indicates the pages on which the STEP1 subroutine listings appear:

The STEP2 listings commence on page 216.

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     2 TAPE7,        TAPE1,TAPE2,TAPE3,TAPE4)
      COMMON /INTGRL/ JCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *     NALL,LRR
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP       ,CBET      ,CETA      ,
     1     CF(3),JCS ,CX2I        ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  ,
     2     DRDP,DADH(2),ETA,HO,N4(3)      ,N5(3)      ,N6(3)      ,PAXP(3,30),
     3     ME        ,ROE        ,ROEC       ,ROEM       ,SALP       ,SBET      ,
     4     SETA      ,SREF       ,SXZI       ,TAU        ,TMAS(20)   ,TONE      ,
     5     TTWO      ,TUW(20)    ,TVW(20)    ,UW         ,VA(5)      ,VB(5)     ,
     6     VW                    ,XIND1      ,XIND2      ,XLREF      ,XM        ,
     7     XMAS(20)  ,XNU        ,XQ         ,XOSM       ,XRE        ,XTEMP     ,
     8     XUW(20)   ,XVW(20)    ,XZI        ,YMAS       ,YMASM
      COMMON /TAICE/ A(3,3)     ,CPH        ,CT(5)      ,SP(10)     ,ST(5)
     1     ,TPH      ,VO(20)     ,ZO(20)     ,AG(3)      ,AMDOT(3)   ,APDOT(4)
     2     ,B(8)     ,CGM(3)     ,DFM(3)     ,DTRAN(3,3),DUD(10,1),DUE(4)
     3     ,DUF(4)   ,F1(3,40)   ,F2(3,5)    ,F3(4,40)   ,GG(3)      ,HI(2)
     4     ,IDN ,I1  ,IP   ,IPC  ,JJ   ,JN   ,KA ,KD(3),KDUM ,KG2
     5     ,K1  ,NNN ,NPTS       ,PAR(6)     ,PDOT(3)    ,PMDOT(3)   ,RES(6)
     6     ,RC       ,SPD(5)     ,SPH        ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)      ,AM(3)      ,AP(4)      ,AX(40)     ,AY(40)
     1     ,AZ(40)   ,C(160)     ,CCAPH(5)   ,CONRD      ,CPHDT(5)   ,CUZ(30,5)
     2     ,CVZ(30,5),D(5)       ,DAT(3,40),DATA(3)      ,DATAS(3)   ,DATC(3)
     3     ,DET      ,DFIT(9)    ,DTF(9)     ,DTI(9)     ,DZ(30)     ,ET
     4     ,FLOS     ,G(3,30)    ,H(3,5)     ,I  ,ICOUNT,J      ,JNBR,JNBRS,JST
     5     ,K   ,KAR ,KC(3)      ,KDAP       ,KDATA      ,KDATAS     ,KG,KI,KK
     6     ,KN  ,KOD ,KPROP,KS,KSM,KSS,L ,LC(3)     ,LCS(3)     ,LS    ,LT
     7     ,L1  ,M   ,MC(5)      ,MCC(5)     ,MO(5)      ,MR(3,9)    ,MT
     8     ,MTP(40)  ,MTYP       ,MTYPS      ,N  ,NC(30),NCOUNT      ,NPC(15)
     9     ,NPU ,NPV ,NS(9)      ,NSS(5)     ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)
     A     ,N8(3)    ,N9(3)      ,N10(3)     ,N11(3)     ,OMEGA      ,P(30,30)
     B     ,PA(3)                ,PM(3)      ,R          ,REO        ,RERP2
     C     ,RPO      ,ROT(5)     ,S(5)       ,SCAPH(5)   ,SI(3)      ,SIG(3,40)
     D     ,SIGM(3)  ,SIGMS(3)   ,SPHDT(5)   ,SUM ,SUM2,SYG(3,9)    ,TFINAL
     E     ,TFI1(9)  ,TIME       ,TIMES      ,TO         ,TP(40)     ,TQ(40)
     F     ,TR(40)   ,TI(40)     ,TXCG(20)   ,TYCG(20)   ,TYM(40)    ,TZCG(20)
     G     ,TZERO    ,XO2        ,XMU        ,XMUJ       ,XP(3)      ,XXCG(20)
     H     ,XYCG(20) ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK
     A    ,PB(23),RL(23),TMB(23)
      COMMON /EXTRA / GO
C---------STATISTICAL TRAJECTORY ESTIMATION PROGRAM, STEP1
C                    DEVELOPED BY W.E.WAGNER AND A.C.SEROLD
C                                     MARTIN MARIETTA CORPORATION
C                                     DENVER, COLORADO
    2 CONTINUE
      CALL PRESET
    4 WRITE(OUT,1001)
      IF(NPC(1).EQ.0) WRITE(OUT,1000) ICOUNT
 1000 FORMAT(19H  ITERATION NUMBER I2//)
 1001 FORMAT(1H1)
   10 IF(KDATA) 12,42,202
   12 T2=TO
      CALL INTAG
      GO TO 44
```

```
   24 T2=TIME
      IF(T2-ET)260,260,42                                              STP10160
   30 IF(TO-ET) 34,32,32                                               STP10170
   32 IF(JST.EQ.1) GO TO 44                                            STP10180
   34 IF(KDATA) 38,36,36                                               STP10190
   36 IF(KPROP.EQ.1) GO TO 24                                          STP10200
      IF(KDATA) 10,40,10                                               STP10210
   38 IF(NPC(1)-1) 200,40,100                                          STP10220
   40 TIME=TFINAL+100.                                                 STP10230
      GO TO 24                                                         STP10240
   42 T2=ET                                                            STP10250
C-------------PROPAGATION LOOP, TO EFN 44                              STP10260
      CALL INTAG                                                       STP10270
      KPROP=1                                                          STP10280
      TO=T2                                                            STP10290
      CALL PROP                                                        STP10300
   44 CALL OUTPUT                                                      STP10310
      L1=2                                                             STP10320
      IF(ET-TFINAL) 46,50,50                                           STP10330
   46 ET=ET+DET                                                        STP10340
      GO TO 34                                                         STP10350
   50 WRITE(SCRACH) KDATA                                              STP10360
      IF(NPC(1).EQ.1) GO TO 2                                          STP10370
   56 IF(NPC(4)) 300,58,300                                            STP10380
   58 CALL SMOOTH                                                      STP10390
      IF(L1-1) 2,4,2                                                   STP10400
C-------------DETERMINE MEASUREMENT TIME FOR ERROR                      STP10410
C                                 ANALYSIS PROBLEMS, THRU EFN 118--    STP10420
  100 IF(KPROP.GT.0) GO TO 24                                          STP10430
      JST=1                                                            STP10432
      IF(NSTA-1) 40,102,104                                            STP10434
  102 II = NS(1)                                                       STP10440
      GO TO 112                                                        STP10450
  104 II = NS(1)                                                       STP10460
      SP(1) = DTI(II)                                                  STP10470
      DO 110 I=2,NSTA                                                  STP10480
      J = NS(I)                                                        STP10500
      IF(SP(1)-DTI(J)) 110,106,107                                     STP10510
  106 JST=JST+1                                                        STP10520
      GO TO 108                                                        STP10530
  107 JST=1                                                            STP10540
  108 SP(1) = DTI(J)                                                   STP10550
      II = J                                                           STP10560
  110 CONTINUE                                                         STP10570
  112 IF(DTI(II)-DTF(II))  116,116,114                                 STP10580
  114 NSTA=NSTA-1                                                      STP10590
      GO TO 100                                                        STP10600
  116 JNBR=0                                                           STP10610
      TIME=DTI(II)                                                     STP10620
      DTI(II)=DTI(II)+DFIT(II)                                         STP10630
      DO 118 I=1,3                                                     STP10640
      IF(MR(I,II).EQ.0) GO TO 118                                      STP10650
      JNBR=JNBR+1                                                      STP10660
      LC(JNBR)=I                                                       STP10670
  118 CONTINUE                                                         STP10680
      IF(JNBR.EQ.0) GO TO 100                                          STP10690
      MTYP = II                                                        STP10700
      GO TO 24                                                         STP10710
C-------------DATA EDITING LOGIC FOR FILTER PROBLEMS, THRU EFN 250------STP10720
  200 KG = 1                                                           STP10730
                                                                       STP10740
```

```
      KS = 1
      TYM(1) = TZERO - 10.
      GO TO 211
  202 KG=KG+1
      MTP(KG)=MTYPS
      TYM(KG)=TIMES
      DAT(1,KG)=DATAS(1)
      DAT(2,KG)=DATAS(2)
      DAT(3,KG)=DATAS(3)
      SIG(1,KG)=SIGMS(1)
      SIG(2,KG)=SIGMS(2)
      SIG(3,KG)=SIGMS(3)
      KDATA=KDATAS
      MTYP=MTYPS
      TIME=TIMES
      DATA(1)=DATAS(1)
      DATA(2)=DATAS(2)
      DATA(3)=DATAS(3)
      SIGM(1)=SIGMS(1)
      SIGM(2)=SIGMS(2)
      SIGM(3)=SIGMS(3)
      LC(1)=LCS(1)
      LC(2)=LCS(2)
      LC(3)=LCS(3)
      JNBR=JNBRS
  204 IF(KI-KS) 206,208,208
  206 KI=KI+1
      GO TO 216
  208 KS=KG
  210 CONTINUE
  211 IF(KDATAS.EQ.0) GO TO 212
      READ(FIT) KDATAS,(MTP(KS+I),TYM(KS+I),(DAT(J,KS+I),J=1,3),(SIG(J
     X,KS+I),J=1,3),I=1,KDATAS)
      KS=KS+KDATAS
      IF(KS-20) 210,215,215
  212 IF(KG+1-KS) 215,213,213
  213 KDATA=0
      GO TO 248
  215 KI=KG+1
  216 MTYPS=MTP(KI)
      TIMES=TYM(KI)
      IF(TIMES.GT.TFINAL) GO TO 213
      IF(NTR(MTYPS)) 220,218,220
  218 IF(TIMES.LT.DTI(MTYPS)) GO TO 204
      TFIT(MTYPS)=TIMES
      NTR(MTYPS)=1
  220 CONTINUE
  222 IF(TIMES-DTF(MTYPS)) 224,224,204
  224 IF(TIMES-TFIT(MTYPS))226,234,228
  226 NTR(MTYPS)=3
      GO TO 204
  228 IF(NTR(MTYPS)-3) 230,232,230
  230 TFIT(MTYPS)=TFIT(MTYPS)+DFIT(MTYPS)
      NTR(MTYPS)=3
      GO TO 224
  232 TFIT(MTYPS)=TIMES
  234 NTR(MTYPS)=2
      JNBRS=0
      DO 238 I=1,3
      IF(MK(I,MTYPS)) 236,238,236
```

```
  236 JNBRS=JNBRS+1                                                       STP11350
      LCS(JNBRS)=I                                                        STP11360
  238 CONTINUE                                                            STP11370
      IF(JNBRS) 240,204,240                                               STP11380
  240 DATAS(1)=DAT(1,KI)                                                  STP11390
      DATAS(2)=DAT(2,KI)                                                  STP11400
      DATAS(3)=DAT(3,KI)                                                  STP11410
      SIGMS(1)=SIG(1,KI)                                                  STP11420
      SIGMS(2)=SIG(2,KI)                                                  STP11430
      SIGMS(3)=SIG(3,KI)                                                  STP11440
      IF(KDATA) 242,244,244                                               STP11450
  242 KDATA=2                                                             STP11455
      IF(TIMES-TZERO) 204,202,202                                         STP11460
  244 IF(TIMES-TIME) 248,246,248                                          STP11470
  246 JST=2                                                               STP11480
      GO TO 250                                                           STP11490
  248 JST=1                                                               STP11500
  250 GO TO 24                                                            STP11510
C------------PROCESSING LOOP, THRU EFN 278-----------------------------STP11520
  260 CALL INTAG                                                          STP11530
      TO=T2                                                               STP11540
      KPROP=0                                                             STP11550
      KN=0                                                                STP11560
      KK = 1                                                              STP11570
      IF(JNBR-1) 270,270,262                                              STP11580
  262 IF(NPC(9)) 268,264,268                                              STP11590
  264 KN=1                                                                STP11600
      GO TO 270                                                           STP11610
  268 KK=JNBR                                                             STP11620
      IF(NPC(4)) 264,270,264                                              STP11630
  270 KAR=1                                                               STP11640
  272 CALL OBSERV                                                         STP11650
  273 IF(T-TSAV) 274,276,274                                              STP11660
  274 CALL PROP                                                           STP11670
      TSAV=T                                                              STP11680
  276 IF(NPC(9).NE.0) KOB = KAR                                           STP11690
      CALL MINVAR                                                         STP11700
      IF(KAR-KK) 278,30,30                                                STP11710
  278 KAR=KAR+1                                                           STP11720
      IF(KN.NE.0) GO TO 276                                               STP11730
      CALL MOTION                                                         STP11740
      GO TO 272                                                           STP11750
C------------UPDATE STATE, EQ.(55A), THRU EFN 312--------------------STP11760
  300 DO 306 I=1,10                                                       STP11770
      SUM=0.                                                              STP11780
      DO 304 K=1,10                                                       STP11790
  304 SUM=SUM+PH(I,K)*DZ(K)                                               STP11800
  306 DUD(I)=SUM                                                          STP11810
      DO 312 I=1,NSTX                                                     STP11820
      IF(I-10) 308,308,310                                                STP11830
  308 DZ(I)=DUD(I)                                                        STP11840
      X(I)=X(I)+DZ(I)                                                     STP11850
      GO TO 312                                                           STP11860
  310 K=NC(I-10)                                                          STP11870
      C(K)=C(K)+DZ(I)                                                     STP11880
      VO(I-10)=C(K)                                                       STP11890
  312 CONTINUE                                                            STP11900
C------------NORMALIZE QUATERNION--------------------------------------STP11910
      DATC(1)= X(7)**2 + X(8)**2 + X(9)**2 + X(10)**2                     STP11920
      X(7) = X(7)/DATC(1)                                                 STP11930
```

```
      X(8) = X(8)/DATC(1)                                               STP11940
      X(9) = X(9)/DATC(1)                                               STP11950
      X(10)=X(10)/DATC(1)                                               STP11960
C-------------UPDATE TRACKING STATION LOCATIONS-------------------------STP11970
      IF(KSS.LE.0) GO TO 316                                            STP11980
      DO 314 I=1,KSS                                                    STP11990
      MTYP=NSS(I)                                                       STP12000
  314 CALL STAT                                                         STP12010
  316 WRITE(OUT,1003)                                                   STP12020
 1003 FORMAT(39H UPDATED MODEL PARAMETERS AT FINAL TIME)                STP12030
      WRITE(OUT,1002) (NC(I),VC(I),I=1,NSTC)                            STP12040
 1002 FORMAT(6(2H C,I3,E15.8))                                          STP12050
      GO TO 58                                                          STP12060
      END                                                               STP12070
```

```
      SUBROUTINE AERO (ARG1,ARG2,ARG3,ARG4,XX,YY,ZZ)
      COMMON /INTGRL/ LCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),      COMM0010
     *    NALL,LRK                                                        COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET      ,CETA      , COMM0030
     1    CF(3),JCS ,CAZ1      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  , COMM0040
     2    DRDP,DWDH(2),ETA,HO,N4(3)      ,N5(3)     ,N6(3)     ,PAXP(3,30), COMM0050
     3    PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET      , COMM0060
     4    SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE      , COMM0070
     5    TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     , COMM0080
     6    VW                   ,XIND1     ,XIND2     ,XLREF     ,XM        , COMM0090
     7    XMAS(20)  ,XNU       ,XQ        ,XQSM      ,XRE       ,XTEMP     , COMM0100
     8    XUW(20)   ,XVW(20)   ,XZI       ,YMAS      ,YMASM                  COMM0110
      COMMON /TWICE/ A(3,3)    ,CPH       ,CT(5)     ,SP(10)    ,ST(5)      COMM0120
     1    ,TPH      ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)   COMM0130
     2    ,B(8)     ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)      COMM0140
     3    ,DUF(4)   ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)      COMM0150
     4    ,IDN ,II  ,IP   ,IPC ,JJ   ,JN  ,KA  ,KD(3),KDUM ,KG2             COMM0160
     5    ,K1   ,NNN ,NPTS     ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)     COMM0170
     6    ,RO       ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)                 COMM0180
      DIMENSION DUB(30,15)                                                 COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                         COMM0200
      COMMON /METH2/ AB(3)     ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)     COMM0210
     1    ,AZ(40)   ,C(160)    ,CCAPH(5) ,CONRD      ,CPHDT(5)  ,CUZ(30,5)  COMM0220
     2    ,CVZ(30,5),D(5)      ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)    COMM0230
     3    ,LET      ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET         COMM0240
     4    ,FLOS     ,G(3,30)   ,H(3,5)    ,I ,ICOUNT,J     ,JNBR,JNBRS,JST  COMM0250
     5    ,K    ,KAR ,KC(3)    ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK   COMM0260
     6    ,KN   ,KOB ,KPROP,KS,KSM,KSS,L  ,LC(3)     ,LCS(3)    ,LS   ,LT   COMM0270
     7    ,L1   ,M   ,MC(5)    ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT         COMM0280
     8    ,MTP(40)  ,MTYP      ,MTYPS     ,N ,NC(30),NCOUNT     ,NPC(15)    COMM0290
     9    ,NPU ,NPV ,NS(9)     ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)    COMM0300
     A    ,N8(3)    ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)   COMM0310
     B    ,PA(3)               ,PM(3)     ,R         ,REO       ,RERP2      COMM0320
     C    ,RPO      ,ROT(5)    ,S(5)      ,SCAPH(5) ,SI(3)      ,SIG(3,40)  COMM0330
     D    ,SIGM(3)  ,SIGMS(3)  ,SPHDT(5)  ,SUM  ,SUM2,SYG(3,9)  ,TFINAL     COMM0340
     E    ,TFIT(9)  ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)     COMM0350
     F    ,TR(40)   ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)   COMM0360
     G    ,TZERO    ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)   COMM0370
     H    ,XYCG(20) ,XZCG(20)                                               COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                     COMM0390
      COMMON /TAPENC/ IN,OUT,FIT,STATE,SCRACH,PQR                          COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK        COMM0410
     A  ,PB(23),RL(23),TMB(23)  .                                          COMM0420
      COMMON /EXTRA / GO                                                   COMM0430
      COMMON /AERTAB/ TABLES(580,3),SAVIND(3,3)                            AERO0010
      DIMENSION TCX(580),TCY(580),TCZ(580),ISTORE(1)                       AERO0020
      EQUIVALENCE (TCX,TABLES(1,1)),(TCY,TABLES(1,2)),(TCZ,TABLES(1,3)),AERO0030
     1            (ISTORE(1),KD(1))                                        AERO0040
C  ***   STORE THE INDEPENDENT ARGUMENTS                                   AERO0050
      ST(1)=ARG1                                                           AERO0060
      ST(2)=ARG2                                                           AERO0070
      ST(3)=ARG3                                                           AERO0080
C  ***   INITIALIZE TABLE-LOOKUP-VALUES ARRAY                              AERO0090
   60 SP(1)=0.                                                             AERO0100
      SP(2)=0.                                                             AERO0110
      SP(3)=0.                                                             AERO0120
C  ***   FIND ALL AERODYNAMIC COEFFICIENTS                                 AERO0130
      DO 90 L=1,3                                                          AERO0140
      DUD=TABLES(1,L)                                                      AERO0150
```

```
      IF(DUD.NE.-1.) GO TO 10
      SP(L)=0.
      GO TO 90
   10 IF(DUD.NE.0) GO TO 15
      SP(L)=TABLES(5,L)
      GO TO 90
   15 J=1
      K=1
      NPTS=DUD
C   ***   DELETE INDEPENDENT ARGUMENTS HAVING VALUE ZERO - - -
C   ***   FOR PURPOSES OF TABLE LOOK-UP
      DO 40 I=1,NPTS
      J=K+1
C     IF(J.GT.4) * * * * * * * * * ERROR MESSAGE
      DO 20 K=J,4
      K=K
      M=K-1
      IF(SAVIND(M,L).EQ.0) GO TO 20
      DUE(I)=ST(M)
      GO TO 40
   20 CONTINUE
   40 CONTINUE
      SP(L)=FUNCT(TABLES(1,L),DUE)
   90 CONTINUE
      CA(1) = SP(1)
      CA(2) = SP(2)
      CA(3) = SP(3)
C   ***   ADJUST THE COEFFICIENTS JUST FOUND
      B(1)=1./(ARG3+1.)**2
      B(2)=ARG3**.618/SQRT(ARG4)
      SP(1)=SP(1)+C(1)+C(4)*ARG1**2+C(7)*B(1)+C(10)*B(2)
      SP(2)=SP(2)+C(2)+C(5)*ARG2   +C(8)*B(1)+C(11)*B(2)
      SP(3)=SP(3)+C(3)+C(6)*ARG1   +C(9)*B(1)+C(12)*B(2)
C   ***   ADJUST CF VECTOR ACCORDING TO NPC(11) LOGIC CONTROL
      IF(NPC(11).NE.1) GO TO 100
      XX=-SP(1)
      YY= CXZI*SP(2)-SXZI*SP(3)
      ZZ=-SXZI*SP(2)-CXZI*SP(3)
      GO TO 120
  100 IF(NPC(11).EQ.0) GO TO 110
      XX=-CALP*SP(1)+SALP*SP(3)
      YY= SP(2)
      ZZ=-SALP*SP(1)-CALP*SP(3)
      GO TO 120
  110 XX=-SP(1)
      YY= SP(2)
      ZZ=-SP(3)
  120 RETURN
      END
```

```
      SUBROUTINE AEROIN
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *     NALL,LRK
      COMMON /MOD1/ ALPH,AS  ,BETA,CA(6),CALP      ,CBET      ,CETA      ,
     1     CF(3),JCS ,CXZI      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH   ,
     2     DRDP,DWDH(2),ETA,HO,N4(3)     ,N5(3)     ,N6(3)     ,PAXP(3,30),
     3     PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET      ,
     4     SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE      ,
     5     TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     ,
     6     VW                   ,XIND1     ,XIND2     ,XLREF     ,XM        ,
     7     XMAS(20)  ,XNU       ,XG        ,XGSM      ,XRE       ,XTEMP     ,
     8     XUW(20)   ,XVW(20)   ,XZI       ,YMAS      ,YMASM
      COMMON /TWICE/ A(3,3)    ,CPH       ,CT(5)     ,SP(10)    ,ST(5)
     1     ,TPH      ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)
     2     ,B(8)     ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)
     3     ,DUF(4)   ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)
     4     ,IDN ,I1  ,IP  ,IPC  ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2
     5     ,K1  ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)
     6     ,RO       ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)     ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)
     1     ,AZ(40)   ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5)
     2     ,CVZ(30,5),D(5)      ,DAT(3,40),DATA(3)   ,DATAS(3)  ,DATC(3)
     3     ,DET      ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET
     4     ,FLOS     ,G(3,30)   ,H(3,5)    ,I ,ICOUNT,J   ,JNBR,JNBRS,JST
     5     ,K   ,KAR ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK
     6     ,KN  ,KOB ,KPROP,KS,KSM,KSS,L  ,LC(3)     ,LCS(3)    ,LS   ,LT
     7     ,L1  ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT
     8     ,MTP(40)  ,MTYP      ,MTYPS     ,N ,NC(30),NCOUNT    ,NPC(15)
     9     ,NPU ,NPV ,NS(9)     ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)
     A     ,N8(3)    ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)
     B     ,PA(3)               ,PM(3)     ,R         ,REO       ,RERP2
     C     ,RPC      ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40)
     D     ,SIGM(3)  ,SIGMS(3)  ,SPHDT(5)  ,SUM ,SUM2,SYG(3,9)  ,TFINAL
     E     ,TFIT(9)  ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)
     F     ,TR(40)   ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)
     G     ,TZERO    ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)
     H     ,XYCG(20) ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK
     A  ,PB(23),RL(23),TMB(23)
      COMMON /EXTRA / GO
      COMMON /AERTAB/ TABLES(580,3),SAVIND(3,3)
      DIMENSION TCX(580),TCY(580),TCZ(580),ISTORE(1)
      EQUIVALENCE (TCX,TABLES(1,1)),(TCY,TABLES(1,2)),(TCZ,TABLES(1,3)),
     1            (ISTORE(1),KD(1))
C   ***   ZERO TABLES ARRAY
      DO 10 I=1,1740
   10 TABLES(I)=0.
      DO 20 I=1,9
   20 SAVIND(I)=0.
C   ***   READ IN ALL THREE AERODYNAMIC-COEFFICIENT TABLES
      DO 200 N=1,3
      NNN=0
      KC(1)=1
      KC(2)=2
      KC(3)=3
```

```
C     ***    READ IN TABLE IDENTIFIER AND INDEPENDENT
C     ***    VARIABLE-AND-VALUE COUNTERS
         READ ( IN,30) IDN,(ISTORE(J),J=1,4)
         WRITE(OUT,30) IDN,(ISTORE(J),J=1,4)
      30 FORMAT(5I4)
         IF(ISTORE(1).EQ.-1) GO TO 200
         NT=0
C     ***    (ONE) OR (NO) INDEPENDENT VALUES FOR A GIVEN
C     ***    INDEPENDENT VARIABLE ARE SPECIAL CASES
         DO 40 LT=2,4
         IF(ISTORE(LT).EQ.1) ISTORE(LT)=0
      40 NT=NT+ISTORE(LT)
         IF(NT.EQ.0) ISTORE(1)=0
         IF(ISTORE(1).NE.0) GO TO 70
         READ ( IN,50) TABLES(5,IDN)
         WRITE(OUT,60) TABLES(5,IDN)
      50 FORMAT(12E6.4)
      60 FORMAT(12F10.6)
         GO TO 200
C     ***    SAVE INDEPENDENT VALUE INFORMATION FOR USE IN SUBROUTINE AERO
      70 DO 80 MT=1,3
      80 SAVIND(MT,IDN)=ISTORE(MT+1)
         DO 90 K=1,4
      90 TABLES(K,IDN)=ISTORE(K)
         NPTS=4
C     ***    COUNT INDEPENDENT VARIABLES TO BE ENTERED AND
C     ***    SET UP CONVERSION INDICATORS
         DO 110 L=1,3
         IF(TABLES(L+1,IDN).NE.0.) GO TO 110
         NNN=NNN+1
         DO 100 J=L,3
     100 KC(J)=KC(J)+1
     110 CONTINUE
C     ***    COMPRESS INPUT WHEN ENTERING MONOVARIATE OR BIVARIATE TABLES
         IF(SAVIND(1,IDN).EQ.0.AND.SAVIND(2,IDN).EQ.0) KC(1)=3
         IF(TABLES(2,IDN).NE.0) GO TO 120
         TABLES(2,IDN)=TABLES(3,IDN)
         TABLES(3,IDN)=TABLES(4,IDN)
         TABLES(4,IDN)=0.
         IF(TABLES(2,IDN).NE.0) GO TO 130
         TABLES(2,IDN)=TABLES(3,IDN)
         TABLES(3,IDN)=0.
         GO TO 130
     120 IF(TABLES(3,IDN).NE.0) GO TO 130
         TABLES(3,IDN)=TABLES(4,IDN)
         TABLES(4,IDN)=0.
     130 NPTS=NPTS-NNN
         KD(1)=TABLES(2,IDN)
         KD(2)=TABLES(3,IDN)
         KD(3)=TABLES(4,IDN)
C     ***    SET FACTORS TO BE USED IN COMPUTING LOOP LIMITS BELOW
         DO 140 K1=1,3
         IF(KD(K1).EQ.0) KD(K1)=1
     140 CONTINUE
C     ***    READ IN ALL INDEPENDENT VALUES AND CONVERT WHERE NECESSARY
         DO 170 I=1,3
         IF(KD(I).NE.1) GO TO 150
         GO TO 170
     150 KDUM=NPTS+1
         NPTS=NPTS+KD(I)
```

```
      READ ( IN,50) (TABLES(J,IDN),J=KDUM,NPTS)                          ARIN0760
      WRITE(OUT,60) (TABLES(J,IDN),J=KDUM,NPTS)                          ARIN0770
      IF(KC(I).GE.3) GO TO 180                                           ARIN0780
      DO 160 M=KDUM,NPTS                                                 ARIN0790
  160 TABLES(M,IDN)=TABLES(M,IDN)*CONRD                                  ARIN0800
  170 CONTINUE                                                           ARIN0810
C    ***   READ IN ALL DEPENDENT VALUES FOR ONE TABLE                    ARIN0820
  180 II=NPTS+1                                                          ARIN0830
      JJ=NPTS+KD(1)*KD(2)*KD(3)                                          ARIN0840
      READ ( IN,50) (TABLES(J,IDN),J=II,JJ)                              ARIN0850
      WRITE(OUT,60) (TABLES(J,IDN),J=II,JJ)                              ARIN0860
  200 CONTINUE                                                           ARIN0870
      RETURN                                                             ARIN0880
      END                                                                ARIN0890
```

```
      SUBROUTINE ATMDAT(ZE,PE,ROE,CE,TM,VMU,H,ISVX)
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK      ATMD0010
     A    ,PB(23),RL(23),TMB(23)                                          ATMD0020
      COMMON /EXTRA / GO                                                  ATMD0030
      IF(JK-2) 20,20,22                                                   ATMD0040
   20 H= ATM(4)*ZE/(ATM(4)+ZE)                                            ATMD0050
      GO TO 24                                                            ATMD0060
   22 H= ZE                                                               ATMD0070
   24 CONTINUE                                                            ATMD0080
    3 I = ISV                                                             ATMD0090
      IF(H-HB(I))4,5,6                                                    ATMD0100
    4 IF(H-HB(I-1))7,8,9                                                  ATMD0110
    5 TM = TMB(I)                                                         ATMD0120
      PE = PB(I)                                                          ATMD0130
      GO TO 15                                                            ATMD0140
    6 IF(H -HB(I+1))10,11,12                                              ATMD0150
    7 I = I-1                                                             ATMD0160
      GO TO 4                                                             ATMD0170
    8 I = I - 1                                                           ATMD0180
      GO TO 5                                                             ATMD0190
    9 I = I - 1                                                           ATMD0200
      GO TO 10                                                            ATMD0210
   11 I = I + 1                                                           ATMD0220
      GO TO 5                                                             ATMD0230
   12 I = I + 1                                                           ATMD0240
      GO TO 6                                                             ATMD0250
   10 TM = TMB(I) + RL(I)*(H-HB(I))                                       ATMD0260
      IF(RL(I))13,14,13                                                   ATMD0270
   13 PE = PB(I)*(TMB(I)/TM)**(ATM(1)/RL(I))                              ATMD0280
      GO TO 15                                                            ATMD0290
   14 PE = PB(I)*EXP(-ATM(1)*(H-HB(I))/TMB(I))                            ATMD0300
   15 ROE= ATM(2)*PE/TM                                                   ATMD0310
      CE = ATM(3)*SQRT(TM)                                                ATMD0320
      VMU= ABET*TM**1.5/(TM+ASU)                                          ATMD0330
      ISV = I                                                             ATMD0340
   99 RETURN                                                              ATMD0350
      END                                                                 ATMD0360
```

```
      SUBROUTINE DATAB
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),     COMM0010
     *    NALL,LRK                                                      COMM0020
      COMMON /MOD1/ ALPH,AS  ,BETA,CA(6),CALP     ,CBET     ,CETA     , COMM0030
     1    CF(3),JCS ,CXZI     ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  , COMM0040
     2    DRDP,DWDH(2),ETA,HO,N4(3)     ,N5(3)    ,N6(3)    ,PAXP(3,30), COMM0050
     3    PE       ,ROE      ,ROEC     ,ROEM     ,SALP     ,SBET     , COMM0060
     4    SETA     ,SREF     ,SXZI     ,TAU      ,TMAS(20) ,TONE     , COMM0070
     5    TTWO     ,TUW(20)  ,TVW(20)  ,UW       ,VA(5)    ,VB(5)    , COMM0080
     6    VW                 ,XIND1    ,XIND2    ,XLREF    ,XM       , COMM0090
     7    XMAS(20) ,XNU      ,XQ       ,XQSM     ,XRE      ,XTEMP    , COMM0100
     8    XUW(20)  ,XVW(20)  ,XZI      ,YMAS     ,YMASM                 COMM0110
      COMMON /TWICE/ A(3,3)  ,CPH      ,CT(5)    ,SP(10)   ,ST(5)       COMM0120
     1    ,TPH     ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)    COMM0130
     2    ,B(8)    ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)     COMM0140
     3    ,DUF(4)  ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)       COMM0150
     4    ,IDN ,II ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2             COMM0160
     5    ,K1  ,NNN ,NPTS    ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)      COMM0170
     6    ,RO      ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)                COMM0180
      DIMENSION DUB(30,15)                                              COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                      COMM0200
      COMMON /METH2/ AB(3)   ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)      COMM0210
     1    ,AZ(40)  ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)   COMM0220
     2    ,CVZ(30,5),D(5)    ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)     COMM0230
     3    ,DET     ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET          COMM0240
     4    ,FLOS    ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J   ,JNBR,JNBRS,JST   COMM0250
     5    ,K   ,KAR ,KC(3)   ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK    COMM0260
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)   ,LCS(3)   ,LS   ,LT    COMM0270
     7    ,L1  ,M   ,MC(5)   ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT          COMM0280
     8    ,MTP(40) ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)     COMM0290
     9    ,NPU ,NPV ,NS(9)   ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)   COMM0300
     A    ,N8(3)   ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)    COMM0310
     B    ,PA(3)             ,PM(3)    ,R        ,REO      ,RERP2       COMM0320
     C    ,RPO     ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)   COMM0330
     D    ,SIGM(3) ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL      COMM0340
     E    ,TFIT(9) ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)      COMM0350
     F    ,TR(40)  ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)    COMM0360
     G    ,TZERO   ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)    COMM0370
     H    ,XYCG(20) ,XZCG(20)                                           COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                  COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                       COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK      COMM0410
     A  ,PB(23),RL(23),TMB(23)                                          COMM0420
      COMMON /EXTRA / GO                                                COMM0430
      IF(DCOMP.LT.0.) GO TO 20                                          DATB0010
      IF(KPROP.GE.0 ) GO TO 10                                          DATB0020
    2 CONTINUE                                                          DATB0030
    4 LS = 0                                                            DATB0040
    6 IF(KDAP.EQ.0)    GO TO 14                                         DATB0050
      READ(PQR) KDAP,(TT(LS+I),TP(LS+I),TQ(LS+I),TR(LS+I),AX(LS+I),     DATB0060
     *    AY(LS+I),AZ(LS+I),I=1,KDAP)                                   DATB0070
      LS = LS+KDAP                                                      DATB0080
      IF(LS.EQ.1) GO TO 6                                               DATB0090
      IF(T.GT.TT(LS)) GO TO 11                                          DATB0100
      IF(LS.LE.20)     GO TO 6                                          DATB0110
      N11(2)=3                                                          DATB0120
   10 IF(T.LT.TT(LS-1)) GO TO 14                                        DATB0130
      IF(KDAP.EQ.0) GO TO 14                                            DATB0140
      KG2 = KG-2                                                        DATB0150
```

```
      WRITE(SCRACH) LS,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),
     *     I=1,LS),KG2,(MTP(I),TYM(I),DAT(1,I),DAT(2,I),DAT(3,I),
     *     SIG(1,I),SIG(2,I),SIG(3,I),I=1,KG2)
      IF(KG2.LE.0) GO TO 11
      KG = 2
      DO 13 I=1,2
      LT = KG2 + I
      MTP(I)= MTP(LT)
      TYM(I)= TYM(LT)
      DAT(1,I)= DAT(1,LT)
      DAT(2,I) = DAT(2,LT)
      DAT(3,I) = DAT(3,LT)
      SIG(1,I) = SIG(1,LT)
      SIG(2,I) = SIG(2,LT)
      SIG(3,I) = SIG(3,LT)
   13 CONTINUE
   11 CONTINUE
      DO 12 I=1,2
      MT = LS  + I - 2
      TT(I) = TT(MT)
      TP(I) = TP(MT)
      TQ(I) = TQ(MT)
      TR(I) = TR(MT)
      AX(I) = AX(MT)
      AY(I) = AY(MT)
      AZ(I) = AZ(MT)
   12 CONTINUE
      LS = 2
      GO TO 6
   14 N11(1)  = LS
      IF(KPROP.LT.0) TRCD=TT(1)
      KPROP = 1
      PM(1) = TAB(T,N11(1),TT(1),TP(1))
      PM(2)=TAB(T,N11(1),TT(1),TQ(1))
      PM(3)=TAB(T,N11(1),TT(1),TR(1))
      CGM(1) = TAB(T,N8(1) ,TXCG(1),XXCG(1))
      CGM(2) = TAB(T,N9(1) ,TYCG(1),XYCG(1))
      CGM(3) = TAB(T,N10(1),TZCG(1),XZCG(1))
   99 RETURN
   20 IF(T.GT.TT(2)) GO TO 14
      IF(TT(1).EQ.TRCD) GO TO 14
      BACKSPACE SCRACH
      BACKSPACE SCRACH
      LT = 40-KG
      DO 22 I=1,KG
      MT = LT+I
      MTP(MT) = MTP(I)
      TYM(MT) = TYM(I)
      DAT(1,MT) = DAT(1,I)
      DAT(2,MT) = DAT(2,I)
      DAT(3,MT) = DAT(3,I)
      SIG(1,MT) = SIG(1,I)
      SIG(2,MT) = SIG(2,I)
   22 SIG(3,MT) = SIG(3,I)
      READ(SCRACH) KDAP,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),
     *     I=1,KDAP),KDATA,(MTP(I),TYM(I),DAT(1,I),DAT(2,I),DAT(3,I),
     *     SIG(1,I),SIG(2,I),SIG(3,I),I=1,KDATA)
      LS=KDAP
      N11(2)=LS-2
      IF(KDATA.LT.0) GO TO 26
```

```
      LT = LT - KDATA
      I = KDATA+1
      J = KDATA+KG
      DO 24 I=I,J
      MT = LT + I
      MTP(I) = MTP(MT)
      TYM(I) = TYM(MT)
      DAT(1,I) = DAT(1,MT)
      DAT(2,I) = DAT(2,MT)
      DAT(3,I) = DAT(3,MT)
      SIG(1,I) = SIG(1,MT)
      SIG(2,I) = SIG(2,MT)
   24 SIG(3,I) = SIG(3,MT)
      KG = KG + KDATA
      GO TO 14
   26 KG = 1
      GO TO 14
      END
```

```
      SUBROUTINE DERIVE
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),      COMM0010
     *     NALL,LRK                                                      COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET      ,CETA      , COMM0030
     1     CF(3),JCS ,CXZI      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  , COMM0040
     2     DRDP,DWDH(2),ETA,HO,N4(3)      ,N5(3)     ,N6(3)     ,PAXP(3,30), COMM0050
     3     PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET      , COMM0060
     4     SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE      , COMM0070
     5     TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     , COMM0080
     6     VW                   ,XIND1     ,XIND2     ,XLREF     ,XM        , COMM0090
     7     XMAS(20)  ,XNU       ,XQ        ,XQSM      ,XRE       ,XTEMP     , COMM0100
     8     XUW(20)   ,XVW(20)   ,XZI       ,YMAS      ,YMASM                 COMM0110
      COMMON /TWICE/ A(3,3)     ,CPH       ,CT(5)     ,SP(10)     ,ST(5)     COMM0120
     1     ,TPH       ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3) ,APDOT(4)   COMM0130
     2     ,B(8)      ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)     COMM0140
     3     ,DUF(4)    ,F1(3,40) ,F2(3,5)   ,F3(4,40) ,GG(3)      ,HI(2)      COMM0150
     4     ,IDN ,II   ,IP   ,IPC ,JJ   ,JN   ,KA ,KD(3),KDUM ,KG2            COMM0160
     5     ,K1  ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3) ,RES(6)      COMM0170
     6     ,RO        ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)                COMM0180
      DIMENSION DUB(30,15)                                               COMM0190
      EGUIVALENCE (AG(1),DUB(1,1))                                       COMM0200
      COMMON /METH2/ AB(3)      ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)    COMM0210
     1     ,AZ(40)    ,C(160)    ,CCAPH(5) ,CONRD    .,CPHDT(5).,CUZ(30,5)   COMM0220
     2     ,CVZ(30,5),D(5)       ,DAT(3,40),DATA(3)   ,DATAS(3) ,DATC(3)    COMM0230
     3     ,DET       ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET         COMM0240
     4     ,FLOS      ,G(3,30)   ,H(3,5)    ,I ,ICOUNT,J    ,JNBR,JNBRS,JST  COMM0250
     5     ,K   ,KAR ,KC(3)      ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK   COMM0260
     6     ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)     ,LCS(3)    ,LS   ,LT     COMM0270
     7     ,L1  ,M   ,MC(5)      ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT         COMM0280
     8     ,MTP(40)   ,MTYP      ,MTYPS     ,N ,NC(30),NCOUNT     ,NPC(15)    COMM0290
     9     ,NPU ,NPV ,NS(9)      ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)    COMM0300
     A     ,N8(3)     ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)   COMM0310
     B     ,PA(3)                ,PM(3)     ,R         ,REO       ,RERP2      COMM0320
     C     ,RPO       ,ROT(5)    ,S(5)      ,SCAPH(5) ,SI(3)      ,SIG(3,40)  COMM0330
     D     ,SIGM(3)   ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL       COMM0340
     E     ,TFIT(9)   ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)     COMM0350
     F     ,TR(40)    ,TT(40)    ,TXCG(20) ,TYCG(20) ,TYM(40)   ,TZCG(20)    COMM0360
     G     ,TZERO     ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)   COMM0370
     H     ,XYCG(20) ,XZCG(20)                                           COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PGR                                   COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PGR                        COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK      COMM0410
     A   ,PB(23),RL(23),TMB(23)                                         COMM0420
      COMMON /EXTRA / GO                                                 COMM0430
      VO(1) = (VA(4)+50.)/AS                                             DERI0010
      VO(2) = (VA(4)-50.)/AS                                             DERI0020
      HI(1) = AS*ROE*XLREF/XNU                                           DERI0030
      DO 2 N=1,2                                                         DERI0040
      HI(2) = VO(N)*HI(1)                                                DERI0050
    2 CALL AERO(XIND1,XIND2,VO(N),HI(2),ZO(N),ZO(N+2),ZO(N+4))           DERI0060
      DCDY(1,1) =(ZO(1)-ZO(2))*1.E-2                                     DERI0070
      DCDY(2,1) =(ZO(3)-ZO(4))*1.E-2                                     DERI0080
      DCDY(3,1) =(ZO(5)-ZO(6))*1.E-2                                     DERI0090
      VO(1) = HO+50.                                                     DERI0100
      VO(2) = HO-50.                                                     DERI0110
      HI(1)=VA(4)*XLREF                                                  DERI0120
      DO 6 N=1,2                                                         DERI0130
      VO(N+2) = TAB(VO(N),N5(1),TUW(1),XUW(1))                           DERI0140
      VO(N+4) = TAB(VO(N),N6(1),TVW(1),XVW(1))                           DERI0150
```

```
      CALL ATMDAT(VO(N),DUD(1),ZO(N),SP(3),DUD(2),DUD(3),DUD(4))
      HI(2)=HI(1)*(C(21)*ZO(N)+C(22)*EXP(C(23)*VO(N)))/DUD(3)
      ZO(3) = VA(4)/SP(3)
      CALL AERO(XIND1,XIND2,ZO(3),HI(2),SP(N+3),SP(N+5),SP(N+7))
    6 CONTINUE
      DCDY(1,2) =(SP(4)-SP(5))*1.E-2
      DCDY(2,2) =(SP(6)-SP(7))*1.E-2
      DCDY(3,2) =(SP(8)-SP(9))*1.E-2
      DRDH = C(21)*(ZO(1)-ZO(2))*1.E-2 + C(23)*ROEC
      DWDH(1) = (VO(3)-VO(4))*1.E-2
      DWDH(2) = (VO(5)-VO(6))*1.E-2
      HI(1) = ALPH + .01
      HI(2) = ALPH - .01
      DO 14 N=1,2
      IF(NPC(11)) 10,10,12
   10 VO(1) = BETA
      GO TO 14
   12 ZO(1) = SIN(HI(N))
      ZO(2) = COS(HI(N))
      HI(N) = ATAN2(SQRT((ZO(1)*CBET)**2+SBET**2),ZO(2)*CBET)
      VO(1) = ATAN2(SBET,ZO(1)*CBET)
   14 CALL AERO(HI(N),VO(1),XM,XRE,VO(N+1),VO(N+3),VO(N+5))
      DCDY(1,3) =(VO(2)-VO(3))*.5E+2
      DCDY(2,3) =(VO(4)-VO(5))*.5E+2
      DCDY(3,3) =(VO(6)-VO(7))*.5E+2
      HI(1) = BETA + .01
      HI(2) = BETA - .01
      DO 22 N=1,2
      IF(NPC(11)) 18,18,20
   18 VO(1) = ALPH
      GO TO 22
   20 ZO(1) = SIN(HI(N))
      ZO(2) = COS(HI(N))
      VO(1) = ATAN2(SQRT((SALP*ZO(2))**2+ZO(1)**2),CALP*ZO(2))
      HI(N) = ATAN2(ZO(1),SALP*ZO(2))
   22 CALL AERO(VO(1),HI(N),XM,XRE,VO(N+1),VO(N+3),VO(N+5))
      DCDY(1,4) =(VO(2)-VO(3))*.5E+2
      DCDY(2,4) =(VO(4)-VO(5))*.5E+2
      DCDY(3,4) =(VO(6)-VO(7))*.5E+2
      DRDP = -(RERP2-1.)*RO*SPH*CPH/(1.+(RERP2-1.)*SPH**2)
      SP(1) = 2.*VA(1)/VA(5)
      SP(2) = 2.*VA(2)/VA(5)
      SP(3) = 2.*VA(3)/VA(5)
      SP(6) = DRDH/ROE + SP(1)*DWDH(1) + SP(2)*DWDH(2)
      SP(4) = SP(6) - SP(2)*OMEGA*CPH
      SP(5) =-SP(6)*DRDP + SP(2)*R*OMEGA*SPH
      ZO(5) = SBET/CBET
      DO 28 I=1,3
      DADX(1,I) = (VB(1)*A(3,I)-VB(3)*A(1,I))/VB(4)**2
      DADX(2,I) = (CBET*A(2,I)-ZO(5)*(VA(I)-A(2,I)*VB(2))/VA(4))/VA(4)
   28 CONTINUE
      DO 30 I=1,2
      ZO(1) = OMEGA*DADX(I,2)
      DADX(I,4) =-CPH*ZO(1)
      DADX(I,5) = SPH*ZO(1)*R
   30 DADX(I,6) = 0.
      ZO(1) = VA(1)*X( 7) + VA(2)*X(10) - VA(3)*X( 9)
      ZO(2) = VA(1)*X( 8) + VA(2)*X( 9) + VA(3)*X(10)
      ZO(3) = VA(1)*X( 9) - VA(2)*X( 8) + VA(3)*X( 7)
      ZO(4) = VA(1)*X(10) - VA(2)*X( 7) - VA(3)*X( 8)
```

```
      VO(1) = VB(1)*ZO(3) - VB(3)*ZO(1)
      VO(2) = VB(1)*ZO(4) - VB(3)*ZO(2)
      VO(3) = VB(1)*ZO(1) + VB(3)*ZO(3)
      VO(4) = VB(1)*ZO(2) + VB(3)*ZO(4)
      VO(5) = 2./VB(4)**2
      DO 32 I=7,10
   32 DADX(1,I) = VO(I-6)*VO(5)
      ZO(6) = 2./VA(4)
      DADX(2,7) = ZO(6)*(-CBET*ZO(4)-ZO(5)*VO(3)/VA(4))
      DADX(2,8) = ZO(6)*( CBET*ZO(3)-ZO(5)*VO(4)/VA(4))
      DADX(2,9) = ZO(6)*( CBET*ZO(2)+ZO(5)*VO(1)/VA(4))
      DADX(2,10)= ZO(6)*(-CBET*ZO(1)+ZO(5)*VO(2)/VA(4))
      ZO(7) = VA(2)*SPH*OMEGA*R/VA(4)
      ZO(8) =(VA(1)*DWDH(1) + VA(2)*(DWDH(2)-OMEGA*CPH))/VA(4)
      DO 36 I=1,3
      DO 34 J=1,3
   34 DERIV(I,J) = AB(I)*SP(J) + XQSM*(VA(J)*DCDY(I,1) /VA(4)
     1     + DCDY(I,3)*DADX(1,J) + DCDY(I,4)*DADX(2,J))
      DERIV(I,4) = AB(I)*SP(4) + XQSM*(DCDY(I,1)*ZO(8) + DCDY(I,2)
     1     + DCDY(I,3)*DADX(1,4) + DCDY(I,4)*DADX(2,4))
      DERIV(I,5) = AB(I)*SP(5) + XQSM*(DCDY(I,1)*ZO(7) - DCDY(I,2)*DRDP
     1     + DCDY(I,3)*DADX(1,5) + DCDY(I,4)*DADX(2,5))
      DO 35 J=7,10
   35 DERIV(I,J) = XQSM*(DCDY(I,3)*DADX(1,J) + DCDY(I,4)*DADX(2,J))
   36 DERIV(I,6) = 0.
      RETURN
      END
```

```
      FUNCTION FUNCT(CRVE,X)
      DIMENSION C(3),IDX(3),CRVE(2),X(3),Z(3)
      DATA IERCNT  /0/
      DATA IDX/3*0/
      N1 = CRVE(2)
      N2 = CRVE(3)
      M  = CRVE(1)
      IF(M.LT.2) N1 = 0
      IF(M.LT.3) N2 = 0
      N2 = N1*N2
      K = M + 1
      DO 25 I=1,M
      J = K + 2
      N = CRVE(I+1)
      K = J + N - 2
      DO 10 L = J,K
      L=L
      IF(X(I).GE.CRVE(L-1)) GO TO 8
C   ***   EXTRAPOLATE BACKWARDS
      C(I)=(X(I)-CRVE(L-1))/(CRVE(L)-CRVE(L-1))
      GO TO 25
    8 IF(X(I).LE.CRVE(L)) GO TO 20
   10 CONTINUE
C   ***   EXTRAPOLATE FORWARDS
      C(I)=(X(I)-CRVE(L-1))/(CRVE(L)-CRVE(L-1))
      GO TO 25
   15 CONTINUE
      Z(1) = 0.0
      GOTO 100
   20 C(I) = (X(I)-CRVE(L-1))/(CRVE(L)-CRVE(L-1))
   25 IDX(I) = L - J + 2
      N = K + IDX(1) + N1*(IDX(2)-2) + N2*(IDX(3)-2)
      Z(1)  = CRV(CRVE,N,C(1))
      IF(M.LT.2) GO TO 100
      N = N + N1
      Z(2) = CRV(CRVE,N,C(1))
      Z(1) = CRV(Z,2,C(2))
      IF(M.EQ.2) GO TO 100
      N = N - N1 + N2
      Z(2) = CRV(CRVE,N,C(1))
      N = N + N1
      Z(3) = CRV(CRVE,N,C(1))
      Z(2) = CRV(Z(1),3,C(2))
      Z(1) = CRV(Z(1),2,C(3))
  100 FUNCT = Z(1)
      RETURN
      END
```

```
      FUNCTION CRV(A,N,C)
      DIMENSION A(2)
      CRV = A(N-1) + C*(A(N)-A(N-1))
      RETURN
      END
```

```
      SUBROUTINE FXXU
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),       COMM0010
     *    NALL,LRK                                                        COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET      ,CETA      , COMM0030
     1    CF(3),JCS ,CXZI      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH   , COMM0040
     2    DRDP,DWDH(2),ETA,HO,N4(3)     ,N5(3)     ,N6(3)     ,PAXP(3,30), COMM0050
     3    PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET      , COMM0060
     4    SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE      , COMM0070
     5    TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     , COMM0080
     6    VW                   ,XIND1     ,XIND2     ,XLREF     ,XM        , COMM0090
     7    XMAS(20) ,XNU        ,XQ        ,XQSM      ,XRE       ,XTEMP     , COMM0100
     8    XUW(20)  ,XVW(20)    ,XZI       ,YMAS      ,YMASM                  COMM0110
      COMMON /TWICE/ A(3,3)    ,CPH       ,CT(5)     ,SP(10)    ,ST(5)       COMM0120
     1    ,TPH      ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)    COMM0130
     2    ,B(8)     ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)       COMM0140
     3    ,DUF(4)   ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)       COMM0150
     4    ,IDN ,II  ,IP  ,IPC ,JJ  ,JN    ,KA ,KD(3),KDUM ,KG2               COMM0160
     5    ,K1  ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)      COMM0170
     6    ,RO       ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)                  COMM0180
      DIMENSION DUB(30,15)                                                   COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                           COMM0200
      COMMON /METH2/ AB(3)     ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)      COMM0210
     1    ,AZ(40)   ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5)   COMM0220
     2    ,CVZ(30,5),D(5)      ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)     COMM0230
     3    ,DET      ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET          COMM0240
     4    ,FLOS     ,G(3,30)   ,H(3,5)    ,I  ,ICOUNT,J    ,JNBR,JNBRS,JST   COMM0250
     5    ,K   ,KAR ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK    COMM0260
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)       ,LCS(3)    ,LS    ,LT   COMM0270
     7    ,L1  ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT          COMM0280
     8    ,MTP(40)  ,MTYP      ,MTYPS     ,N  ,NC(30),NCOUNT    ,NPC(15)     COMM0290
     9    ,NPU ,NPV ,NS(9)     ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)     COMM0300
     A    ,N8(3)    ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)    COMM0310
     B    ,PA(3)               ,PM(3)     ,R         ,REO       ,RERP2       COMM0320
     C    ,RPO      ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40)   COMM0330
     D    ,SIGM(3)  ,SIGMS(3)  ,SPHDT(5)  ,SUM ,SUM2,SYG(3,9)   ,TFINAL      COMM0340
     E    ,TFIT(9)  ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)      COMM0350
     F    ,TR(40)   ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)    COMM0360
     G    ,TZERO    ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)    COMM0370
     H    ,XYCG(20) ,XZCG(20)                                                COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                       COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                            COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK          COMM0410
     A  ,PB(23),RL(23),TMB(23)                                               COMM0420
      COMMON /EXTRA / GO                                                     COMM0430
      DIMENSION CX(3,4)                                                      FXXU0010
      DIMENSION XPM(3,3)                                                     FXXU0020
      EQUIVALENCE (XPM(1,1),DPH(1,1))                                        FXXU0030
      EQUIVALENCE (C(48),CX(1,1))                                            FXXU0040
      DIMENSION NO(1)                                                        FXXU0050
      DATA NO(1) /-1/                                                        FXXU0060
      DATA ITER /5/                                                          FXXU0070
      SP(1) = PAR(4)*X(2)/CPH**2                                             FXXU0080
      F2(2,1) = 1./R                                                         FXXU0090
      F1(1,3) = F2(2,1)*X(1)                                                 FXXU0100
      F1(3,1) =-F1(1,3)-F1(1,3)                                              FXXU0110
      F1(3,2) =-F1(2,3)-F1(2,3)                                              FXXU0120
      F1(1,1) = F2(2,1)*X(3)                                                 FXXU0130
      F1(1,2) =-F1(2,1)-F1(2,1)                                              FXXU0140
      F1(1,4) =(-PAR(5)*F2(2,1)+4.*PAR(3)*SPH*CPH)*F2(2,1)                   FXXU0150
```

```
      F1(1,5) = SP(1)*F1(3,2)*R - PAR(3)*(CPH+SPH)*(CPH-SPH)            FXXU0160
      F1(2,4) =-F1(2,2)*F1(2,3)                                         FXXU0170
      F1(2,5) =-SP(1)*F1(3,1)*R                                         FXXU0180
      F1(3,4) = (PAR(1)*F2(2,1)-PAR(2)-PAR(2) + 4.*PAR(3)*PAR(6))*      FXXU0190
     *          F2(2,1)                                                 FXXU0200
      F1(3,5) =-3.*PAR(3)*SPH*CPH                                       FXXU0210
      F2(2,4) =-F1(1,3)*F2(2,1)                                         FXXU0220
      F2(3,2) = F2(2,1)/CPH                                             FXXU0230
      F2(3,4) =-F1(2,3)*F2(3,2)                                         FXXU0240
      F2(3,5) = F1(2,1)/CPH                                             FXXU0250
      F3(1,1) =-X( 9)*PAR(4)                                            FXXU0260
      F3(2,1) = X(10)*PAR(4)                                            FXXU0270
      F3(3,1) = X( 7)*PAR(4)                                            FXXU0280
      F3(4,1) =-X( 8)*PAR(4)                                            FXXU0290
      F3(1,2) = (X( 8)-X(10)*TPH)*PAR(4)                                FXXU0300
      F3(2,2) =-(X( 7)+X( 9)*TPH)*PAR(4)                                FXXU0310
      F3(3,2) = (X(10)+X( 8)*TPH)*PAR(4)                                FXXU0320
      F3(4,2) =-(X( 9)-X( 7)*TPH)*PAR(4)                                FXXU0330
      F3(1,4) =-F1(1,3)*F3(1,1) - F1(2,3)*F3(1,2)                       FXXU0340
      F3(2,4) =-F1(1,3)*F3(2,1) - F1(2,3)*F3(2,2)                       FXXU0350
      F3(3,4) =-F1(1,3)*F3(3,1) - F1(2,3)*F3(3,2)                       FXXU0360
      F3(4,4) =-F1(1,3)*F3(4,1) - F1(2,3)*F3(4,2)                       FXXU0370
      F3(1,5) =-X(10)*SP(1)                                             FXXU0380
      F3(2,5) =-X( 9)*SP(1)                                             FXXU0390
      F3(3,5) = X( 8)*SP(1)                                             FXXU0400
      F3(4,5) = X( 7)*SP(1)                                             FXXU0410
      F1(1,7) = 2.*(X( 7)*AB(1)-X(10)*AB(2)+X( 9)*AB(3))                FXXU0420
      F1(1,8) = 2.*(X( 8)*AB(1)+X( 9)*AB(2)+X(10)*AB(3))                FXXU0430
      F1(1,9) = 2.*(-X(9)*AB(1)+X( 8)*AB(2)+X( 7)*AB(3))                FXXU0440
      F1(1,10)=-2.*(X(10)*AB(1)+X( 7)*AB(2)-X( 8)*AB(3))                FXXU0450
      F1(2,7) =-F1(1,10)                                                FXXU0460
      F1(2,8) =-F1(1, 9)                                                FXXU0470
      F1(2,9) = F1(1, 8)                                                FXXU0480
      F1(2,10)= F1(1, 7)                                                FXXU0490
      F1(3,7) = F1(1, 9)                                                FXXU0500
      F1(3,8) =-F1(1,10)                                                FXXU0510
      F1(3,9) =-F1(1, 7)                                                FXXU0520
      F1(3,10)= F1(1, 8)                                                FXXU0530
      F3(2,7) =-F3(1,8)                                                 FXXU0540
      F3(3,7) =-F3(1, 9)                                                FXXU0550
      F3(3,8) =-F3(2, 9)                                                FXXU0560
      F3(4,7) =-F3(1,10)                                                FXXU0570
      F3(4,8) =-F3(2,10)                                                FXXU0580
      F3(4,9) =-F3(3,10)                                                FXXU0590
      F2(1,3) = -1.                                                     FXXU0600
      F1(1,6) = 0.                                                      FXXU0610
      F1(2,6) = 0.                                                      FXXU0620
      F1(3,6) = 0.                                                      FXXU0630
      F1(3,3) = 0.                                                      FXXU0640
      F2(1,1) = 0.                                                      FXXU0650
      F2(1,2) = 0.                                                      FXXU0660
      F2(1,4) = 0.                                                      FXXU0670
      F2(1,5) = 0.                                                      FXXU0680
      F2(2,2) = 0.                                                      FXXU0690
      F2(2,3) = 0.                                                      FXXU0700
      F2(2,5) = 0.                                                      FXXU0710
      F2(3,1) = 0.                                                      FXXU0720
      F2(3,3) = 0.                                                      FXXU0730
      F3(1,3) = 0.                                                      FXXU0740
      F3(2,3) = 0.                                                      FXXU0750
```

```
      F3(3,3) = 0.                                                      FXXU0760
      F3(4,3) = 0.                                                      FXXU0770
      F3(1,6) = 0.                                                      FXXU0780
      F3(2,6) = 0.                                                      FXXU0790
      F3(3,6) = 0.                                                      FXXU0800
      F3(4,6) = 0.                                                      FXXU0810
      F3(1,7) = 0.                                                      FXXU0820
      F3(2,8) = 0.                                                      FXXU0830
      F3(3,9) = 0.                                                      FXXU0840
      F3(4,10)= 0.                                                      FXXU0850
      CALL DERIVE                                                       FXXU0860
      DO 410 J=1,10                                                     FXXU0870
      DO 410 I=1,3                                                      FXXU0880
      SUM=F1(I,J)                                                       FXXU0890
      DO 400 K=1,3                                                      FXXU0900
      SUM=SUM+A(K,I)*DERIV(K,J)                                         FXXU0910
  400 CONTINUE                                                          FXXU0920
      F1(I,J)=SUM                                                       FXXU0930
  410 CONTINUE                                                          FXXU0940
      IF(NO(1)) 415,425,420                                             FXXU0950
  415 VO(10)=AP(1)**2+AP(3)**2                                          FXXU0960
      VO(11)=AP(4)*(1./AP(1)-AP(1)/VO(10))                              FXXU0970
      VO(12)=AP(4)*(1./AP(3)-AP(3)/VO(10))                              FXXU0980
  420 CONTINUE                                                          FXXU0990
      K = 0                                                             FXXU1000
      DO 422 I=1,3                                                      FXXU1010
      DO 422 J=1,3                                                      FXXU1020
      K = K+1                                                           FXXU1030
  422 SP(K) = PA(I)*XP(J)                                               FXXU1040
      DTRAN(1,1) =      SP(5) +      SP(9)                              FXXU1050
      DTRAN(1,2) = -2.*SP(4) +      SP(2)                               FXXU1060
      DTRAN(1,3) = -2.*SP(7) +      SP(3)                               FXXU1070
      DTRAN(2,1) =      SP(4) -  2.*SP(2)                               FXXU1080
      DTRAN(2,2) =      SP(1) +      SP(9)                              FXXU1090
      DTRAN(2,3) = -2.*SP(8) +      SP(6)                               FXXU1100
      DTRAN(3,1) =      SP(7) -  2.*SP(3)                               FXXU1110
      DTRAN(3,2) =      SP(8) -  2.*SP(6)                               FXXU1120
      DTRAN(3,3) =      SP(1) +      SP(5)                              FXXU1130
  425 DO 500 N=1,10                                                     FXXU1140
      SP(4)=0.                                                          FXXU1150
      SP(5)=0.                                                          FXXU1160
      SP(6)=0.                                                          FXXU1170
      IF(N.NE.6) GO TO 440                                              FXXU1180
      PAXP(1,6) = 0.                                                    FXXU1190
      PAXP(2,6) = 0.                                                    FXXU1200
      PAXP(3,6) = 0.                                                    FXXU1210
      PAXP(4,6) = 0.                                                    FXXU1220
      GO TO 500                                                         FXXU1230
  440 DO 470 KA=1,ITER                                                  FXXU1240
      DO 450 I=1,3                                                      FXXU1250
      PAXP(I,N)=DERIV(I,N)                                              FXXU1260
      DO 450 J=1,3                                                      FXXU1270
      PAXP(I,N) = PAXP(I,N) + DTRAN(I,J)*SP(J+3)                        FXXU1280
  450 CONTINUE                                                          FXXU1290
      IF(NO(1)) 455,460,460                                             FXXU1300
  455 PAXP(4,N)=VO(11)*PAXP(1,N)+VO(12)*PAXP(3,N)                       FXXU1310
  460 DO 470 I=1,3                                                      FXXU1320
      SP(I+3)=0.                                                        FXXU1330
      IF(I.EQ.1) SP(4) = CX(1,4)*PAXP(4,N)                              FXXU1340
      DO 470 J=1,3                                                      FXXU1350
```

```
      SP(I+3)=SP(I+3)+CX(I,J)*PAXP(J,N)
  470 CONTINUE
      F3(1,N)=F3(1,N)+.5*(-X( 8)*SP(4)-X( 9)*SP(5)-X(10)*SP(6))
      F3(2,N)=F3(2,N)+.5*( X( 7)*SP(4)-X(10)*SP(5)+X( 9)*SP(6))
      F3(3,N)=F3(3,N)+.5*( X(10)*SP(4)+X( 7)*SP(5)-X( 8)*SP(6))
      F3(4,N)=F3(4,N)+.5*(-X( 9)*SP(4)+X( 8)*SP(5)+X( 7)*SP(6))
  500 CONTINUE
      IPC = 0
      LT = 0
      NT = 0
      IF(NSTC) 1,1,6
    1 IF(NPU)  2,99,2
    2 N = NSTX
      NT = 1
      II = 1
    3 NNN = MC(II)
      GO TO 18
    6 N = 10
      II = 1
    9 NNN = NC(II)
      GO TO 18
   12 II = II+1
      IF(NT) 15,13,15
   13 IF(NSTC-II) 1,9,9
   15 IF(NPU -II) 99,3,3
   18 CONTINUE
      IF(NNN-60) 19,19,25
   19 L=0
      M =0
      SP(1) = 0.
      SP(2) = 0.
      SP(3) = 0.
      ST(1) = 0.
      ST(2) = 0.
      ST(3) = 0.
      IF(NNN-15) 50,50,20
   20 IF(NNN-20) 80,80,21
   21 IF(NNN-25) 85,85,22
   22 IF(NNN-30) 95,95,23
   23 L=1
      SP(4)  = 0.
      SP(5)  = 0.
      SP(6)  = 0.
      IF(NNN-35) 100,100,150
   25 DO 26 I=1,3
      F1(I,II+N) = 0.
      F3(I,II+N) = 0.
   26 CONTINUE
      F3(4,II+N) = 0.
      GO TO 12
   30 SP(4)=0.
      SP(5)=0.
      SP(6)=0.
      SP(7) = 0.
   33 CONTINUE
      DO 32 KA=1,ITER
      DO 31 I=1,3
      VO(I)=SP(I)
      DO 31 J=1,3
      VO(I) = VO(I) +DTRAN(I,J)*SP(J+3)
```

```
 31 CONTINUE                                                         FXXU1960
    IF(NO(1)) 600,610,610                                            FXXU1970
600 SP(7) = VO(11)*VO(1)+VO(12)*VO(3)                                FXXU1980
610 DO 32 I=1,3                                                      FXXU1990
    SP(I+3)=ST(I)                                                    FXXU2000
    IF(I-1) 630,620,630                                              FXXU2010
620 SP(4)=C(57)*SP(7) + SP(4)                                        FXXU2020
630 DO 32 J=1,3                                                      FXXU2030
    SP(I+3)=SP(I+3)+CX(I,J)*VO(J)                                    FXXU2040
 32 CONTINUE                                                         FXXU2050
 36 I=II+N                                                           FXXU2060
    IF(LT.NE.0) RETURN                                               FXXU2070
 39 F3(1,I) = .5*(-X( 8)*SP(4)-X( 9)*SP(5)-X(10)*SP(6))              FXXU2080
    F3(2,I) = .5*( X( 7)*SP(4)-X(10)*SP(5)+X( 9)*SP(6))              FXXU2090
    F3(3,I) = .5*( X(10)*SP(4)+X( 7)*SP(5)-X( 8)*SP(6))              FXXU2100
    F3(4,I) = .5*(-X( 9)*SP(4)+X( 8)*SP(5)+X( 7)*SP(6))              FXXU2110
    IF(L) 46,45,46                                                   FXXU2120
 45 F1(1,I) = SP(1)*A(1,1) + SP(2)*A(2,1) + SP(3)*A(3,1)             FXXU2130
    F1(2,I) = SP(1)*A(1,2) + SP(2)*A(2,2) + SP(3)*A(3,2)             FXXU2140
    F1(3,I) = SP(1)*A(1,3) + SP(2)*A(2,3) + SP(3)*A(3,3)             FXXU2150
    GO TO 12                                                         FXXU2160
 46 F1(1,I)=0.                                                       FXXU2170
    F1(2,I)=0.                                                       FXXU2180
    F1(3,I)=0.                                                       FXXU2190
    GO TO 12                                                         FXXU2200
 50 VO(1)=0.                                                         FXXU2210
    VO(2)=0.                                                         FXXU2220
    VO(3)=0.                                                         FXXU2230
    IF(NNN-4) 51,52,53                                               FXXU2240
 51 VO(NNN)=1.                                                       FXXU2250
    GO TO 60                                                         FXXU2260
 52 VO(1)=XIND1**2                                                   FXXU2270
    GO TO 60                                                         FXXU2280
 53 IF(NNN-6) 54,55,56                                               FXXU2290
 54 VO(2)=XIND2                                                      FXXU2300
    GO TO 60                                                         FXXU2310
 55 VO(3)=XIND1                                                      FXXU2320
    GO TO 60                                                         FXXU2330
 56 I=NNN-((NNN-1)/3)*3                                              FXXU2340
    IF(NNN-9) 57,57,58                                               FXXU2350
 57 VO(I)=(XM+1)**-2                                                 FXXU2360
    GO TO 60                                                         FXXU2370
 58 VO(I)=XM**.618/SQRT(XRE)                                         FXXU2380
 60 CONTINUE                                                         FXXU2390
    IF(NPC(11)) 61,62,63                                             FXXU2400
 61 SP(1) =-XQSM*(CALP*VO(1)-SALP*VO(3))                             FXXU2410
    SP(2)= XQSM*       VO(2)                                         FXXU2420
    SP(3) =-XQSM*(SALP*VO(1)+CALP*VO(3))                             FXXU2430
    GO TO 30                                                         FXXU2440
 62 SP(1)=-XQSM*VO(1)                                                FXXU2450
    SP(2)= XQSM*VO(2)                                                FXXU2460
    SP(3)=-XQSM*VO(3)                                                FXXU2470
    GO TO 30                                                         FXXU2480
 63 SP(1)=-XQSM*VO(1)                                                FXXU2490
    SP(2) = XQSM*(CXZI*VO(2)-SXZI*VO(3))                             FXXU2500
    SP(3) =-XQSM*(SXZI*VO(2)+CXZI*VO(3))                             FXXU2510
    GO TO 30                                                         FXXU2520
 80 IF(T.LT.TONE) GO TO 84                                           FXXU2530
    SP(1)=AB(1)/YMAS                                                 FXXU2532
    SP(2)=AB(2)/YMAS                                                 FXXU2534
```

```
      SP(3)=AB(3)/YMAS
      IF(NNN-17) 30,81,82
   81 SP(1)=TAU*SP(1)
      SP(2)=TAU*SP(2)
      SP(3)=TAU*SP(3)
      GO TO 30
   82 SP(1)=SP(1)*TAU**2
      SP(2)=SP(2)*TAU**2
      SP(3)=SP(3)*TAU**2
      GO TO 30
   84 SP(4)=0.
      SP(5)=0.
      SP(6)=0.
      L=1
      GO TO 36
   85 CONTINUE
      KA = N
      CALL AERO(XIND1,XIND2,XM,1.005*XRE,VO(4),VO(5),VO(6))
      CALL AERO(XIND1,XIND2,XM,0.995*XRE,VO(1),VO(2),VO(3))
      L = 0
      N = KA
      SP(1)= (VO(4)-VO(1))/(XRE*E-2)
      SP(2)= (VO(5)-VO(2))/(XRE*E-2)
      SP(3)= (VO(6)-VO(3))/(XRE*E-2)
      IF(NNN-22) 86,87,90
   86 SP(4) = ROEM/ROE
      GO TO 92
   87 SP(4) = EXP(HO*C(23))/ROE
      GO TO 92
   90 SP(4) = HO*ROEC/ROE
   92 CONTINUE
      ST(1)=0.
      ST(2)=0.
      ST(3)=0.
   93 CONTINUE
      DO 94 I=1,3
   94 SP(I) = SP(4)*(AB(I)+XQSM*XRE*SP(I))
      GO TO 30
   95 VO(1)=VA(NNN-25)
      VO(2)=DADX(1,NNN-25)
      VO(3)=DADX(2,NNN-25)
      DO 98 I=1,3
   98 SP(I) =2.*VO(1)*AB(I)/VA(4)+XQSM*(DCDY(I,1)*VO(1)+DCDY(I,2)*VO(2)+
     2                                    DCDY(I,3)*VO(3))
      GO TO 30
  100 I=NNN-30
      IF(NO(1)) 102,103,103
  102 SP(7) = VO(11)*TRAN(1,I)+VO(12)*TRAN(3,I)
  103 DO 112 J=1,3
      IF(J.EQ.1) SP(4) = C(57)*SP(7)
  106 DO 110 K=1,3
      SP(J+3) =SP(J+3)+CX(J,K)*TRAN(K,I)
  110 CONTINUE
  112 SP(J) = TRAN(J,I)
      GO TO 33
  150 CONTINUE
      IF(NNN-48) 160,250,250
  160 IF(NNN-45) 170,200,200
  170 J = (NNN-33)/3
      K = NNN - 3*J - 32
```

```
      SP(J+3) = PM(K)                                                   FXXU3070
      ST(J) = PM(K)                                                     FXXU3080
      AG(J) = PMDOT(K)                                                  FXXU3090
      IF(IPC.EQ.0) GO TO 175                                            FXXU3100
  172 SP(1) = XPM(1,J)*PMDOT(K)                                         FXXU3110
      SP(2) = XPM(2,J)*PMDOT(K)                                         FXXU3120
      SP(3) = XPM(3,J)*PMDOT(K)                                         FXXU3130
      GO TO 33                                                          FXXU3140
  175 IPC = 1                                                           FXXU3150
      XPM(1,1) = 0.                                                     FXXU3160
      XPM(2,2) = 0.                                                     FXXU3170
      XPM(3,3) = 0.                                                     FXXU3180
      XPM(1,2) = XP(3)                                                  FXXU3190
      XPM(2,1) =-XP(3)                                                  FXXU3200
      XPM(3,1) = XP(2)                                                  FXXU3210
      XPM(1,3) =-XP(2)                                                  FXXU3220
      XPM(2,3) = XP(1)                                                  FXXU3230
      XPM(3,2) =-XP(1)                                                  FXXU3240
      GO TO 172                                                         FXXU3250
  200 SP(NNN-41) = 1.                                                   FXXU3260
      GO TO 36                                                          FXXU3270
  250 J = (NNN-45)/3                                                    FXXU3280
      K = NNN - 3*J - 41                                                FXXU3290
      SP(K) = AP(J)                                                     FXXU3300
      GO TO 36                                                          FXXU3310
   99 RETURN                                                            FXXU3320
      ENTRY PAXPC                                                       FXXU3330
      IPC = 0                                                           FXXU3340
      LT = 1                                                            FXXU3350
      GO TO 19                                                          FXXU3360
      END                                                               FXXU3370
```

```
      SUBROUTINE INDAT
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *    NALL,LRK
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP     ,CBET     ,CETA     ,
     1    CF(3),JCS ,CXZI     ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH ,
     2    DRDP,DWDH(2),ETA,HO,N4(3)     ,N5(3)     ,N6(3)     ,PAXP(3,30),
     3    PE        ,ROE      ,ROEC     ,ROEM      ,SALP      ,SBET     ,
     4    SETA      ,SREF     ,SXZI     ,TAU       ,TMAS(20)  ,TONE     ,
     5    TTWO      ,TUW(20)  ,TVW(20)  ,UW        ,VA(5)     ,VB(5)    ,
     6    VW                  ,XIND1    ,XIND2     ,XLREF     ,XM       ,
     7    XMAS(20)  ,XNU      ,XQ       ,XQSM      ,XRE       ,XTEMP    ,
     8    XUW(20)   ,XVW(20)  ,XZI      ,YMAS      ,YMASM
      COMMON /TWICE/ A(3,3)   ,CPH      ,CT(5)     ,SP(10)    ,ST(5)
     1    ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)     ,AMDOT(3)  ,APDOT(4)
     2    ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)
     3    ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40)  ,GG(3)     ,HI(2)
     4    ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2
     5    ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)   ,PMDOT(3)  ,RES(6)
     6    ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)    ,AM(3)    ,AP(4)     ,AX(40)    ,AY(40)
     1    ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD     ,CPHDT(5)  ,CUZ(30,5)
     2    ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)   ,DATAS(3)  ,DATC(3)
     3    ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)    ,DZ(30)    ,ET
     4    ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J   ,JNBR,JNBRS,JST
     5    ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA     ,KDATAS    ,KG,KI,KK
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)     ,LCS(3)    ,LS   ,LT
     7    ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)     ,MR(3,9)   ,MT
     8    ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT     ,NPC(15)
     9    ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)
     A    ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)    ,OMEGA     ,P(30,30)
     B    ,PA(3)              ,PM(3)    ,R         ,REO       ,RERP2
     C    ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5)  ,SI(3)     ,SIG(3,40)
     D    ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9)   ,TFINAL
     E    ,TFIT(9)  ,TIME     ,TIMES    ,TO        ,TP(40)    ,TQ(40)
     F    ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20)  ,TYM(40)   ,TZCG(20)
     G    ,TZERO    ,XJ2      ,XMU      ,XMUJ      ,XP(3)     ,XXCG(20)
     H    ,XYCG(20) ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK
     A ,Pb(23),RL(23),TMB(23)
      COMMON /EXTRA / GO
      DATA CONV /.3048/
      REWIND FIT
      REWIND PQR
      REWIND SCRACH
      REWIND STATE
      WRITE(OUT,1030)
 1030 FORMAT(1H1,22H DATA ANALYSIS PROGRAM//11H INPUT DATA//)
    8 READ(IN,1000) KDUM,(B(I),I=1,8)
 1000 FORMAT(I2,7A10,A8)
      WRITE(OUT,1001) KDUM,(B(I),I=1,8)
 1001 FORMAT(5X,I2,5X,7A10,5X,A8)
      IF(KDUM) 7,8,9
    7 CALL EXIT
    9 GO TO ( 8,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,
     A180,190,200),KDUM
```

```
   20 DECODE(40,1002,B) (NPC(I),I=1,10)
 1002 FORMAT(10I4)
      IF(NPC(2).EQ.0) GO TO 8
      XMU = XMU/CONV**3
      REO = REO/CONV
      RPO = RPO/CONV
      GO = GO/CONV
      GO TO 8
   30 DECODE(30,1003,B) IPC,IDN,SP(1),SP(2)
 1003 FORMAT(I2,I4,E12.4,E12.4)
      GO TO (32,34,36,38,39),IPC
   32 NST=NST+1
      ZO(IDN)=SP(1)
      P(IDN,IDN)=SP(2)**2
      GO TO 8
   34 NSTC = NSTC + 1
      NC(NSTC) = IDN
      C(IDN)=SP(1)
      P(NSTC+10,NSTC+10) = SP(2)**2
      GO TO 8
   36 NPU=NPU+1
      MC(NPU)=IDN
      C(IDN)=SP(1)
      D(NPU)=SP(2)**2
      GO TO 8
   38 NPV=NPV+1
      MCC(NPV)=IDN
      C(IDN)=SP(1)
      S(NPV)=SP(2)**2
      GO TO 8
   39 C(IDN) = SP(1)
      GO TO 8
   40 DECODE(54,1004,B) TO,DET,DCOMP,TFINAL
 1004 FORMAT(6X,4E12.4)
      GO TO 8
   50 DECODE(42,1005,B) IPC,SP(1),SP(2),SP(3)
 1005 FORMAT(I2,4X,3E12.4)
      IF(IPC.EQ.2) GO TO 54
   52 GO = SP(1)
      XMU=SP(2)
      XJ2 = SP(3)
      GO TO 8
   54 RPO = SP(1)
      REO = SP(2)
      OMEGA=SP(3)
      GO TO 8
   60 DECODE(6,1006,B) IPC,NPTS
 1006 FORMAT(I2,I4)
      GO TO(62,64,66),IPC
   62 READ(IN,1031)  (TXCG(I),XXCG(I),I=1,NPTS)
      WRITE(OUT,1031)(TXCG(I),XXCG(I),I=1,NPTS)
 1031 FORMAT(6E12.4)
      N8(1)=NPTS
      GO TO 8
   64 READ(IN,1031)  (TYCG(I),XYCG(I),I=1,NPTS)
      WRITE(OUT,1031)(TYCG(I),XYCG(I),I=1,NPTS)
      N9(1)=NPTS
      GO TO 8
   66 READ(IN,1031)  (TZCG(I),XZCG(I),I=1,NPTS)
      WRITE(OUT,1031)(TZCG(I),XZCG(I),I=1,NPTS)
```

```
      N10(1)=NPTS
      GO TO 8
   70 DECODE(68,1007,B) IDN,SP(1),SP(2),SP(3),LT,MT,NT,SP(4),SP(5),SP(6)
 1007 FORMAT(I2,3E10.4,3I2,3E10.4)
      NSTA=NSTA+1
      NS(NSTA) = IDN
      DTI(IDN)=SP(1)
      DTF(IDN)=SP(2)
      DFIT(IDN)=SP(3)
      MR(1,IDN)=LT
      MR(2,IDN)=MT
      MR(3,IDN)=NT
      SYG(1,IDN)=SP(4)
      SYG(2,IDN)=SP(5)
      SYG(3,IDN)=SP(6)
      GO TO 8
   80 DECODE(6,1006,B) IPC,IDN
      IF(IDN) 81,83,81
   81 READ(IN,1032) KDATA
      IF(KDATA.LE.0) GO TO 82
      READ(IN,1032) (K,MTP(I),TYM(I), (DAT(J,I),J=1,3),(SIG(J,I),J=1,3),
     *     I=1,KDATA)
      WRITE(FIT)    KDATA,(MTP(I),TYM(I),(DAT(J,I),J=1,3),(SIG(J,I),J=1,
     X3),I=1,KDATA)
      GO TO 81
 1032 FORMAT(2I2,7E10.2)
   82 CONTINUE
      K = 0
      WRITE(FIT)KDATA,K,K,K,K,K,K,K,K
      REWIND FIT
   83 IF(IPC) 84,8,84
   84 READ(FIT)      KDATA,(MTP(I),TYM(I),(DAT(J,I),J=1,3),(SIG(J,I),J=1,
     X3),I=1,KDATA)
      WRITE(OUT,1033)KDATA,(MTP(I),TYM(I),(DAT(J,I),J=1,3),(SIG(J,I),J=1
     X,3),I=1,KDATA)
 1033 FORMAT(2I4,7E16.7/(4X,I4,7E16.7))
      IF(KDATA) 84,85,84
   85 REWIND FIT
      GO TO 8
   90 DECODE(6,1006,B) IPC,IDN
      IF(IDN) 91,93,91
   91 READ(IN,1034) KDAP
      IF(KDAP.LE.0) GO TO 92
      READ(IN,1034) (K,TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),I=1,
     *     KDAP)
      WRITE(PQR)KDAP,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),I=1,
     *     KDAP)
      GO TO 91
 1034 FORMAT(I2,7E10.2)
   92 CONTINUE
      K = 0
      WRITE(PQR) KDAP,K,K,K,K,K,K,K
      REWIND PQR
   93 IF(IPC) 94,8,94
   94 READ(PQR) KDAP,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),I=1,KDAP
     X)
      WRITE(OUT,1035) KDAP,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),I=
     X1,KDAP)
 1035 FORMAT(I4,7E16.7/(4X,7E16.7))
      IF(KDAP) 94,95,94
```

```
   95 REWIND PQR
      GO TO 8
  100 DECODE(6,1006,B) IPC,NPTS
      NSTX = NST+NSTC
      GO TO (101,102,103,104,106,108),IPC
  101 READ(IN,1031)    ((P(I,J),J=1,NSTX),I=1,NSTX)
      WRITE(OUT,1031) ((P(I,J),J=1,NSTX),I=1,NSTX)
      GO TO 8
  102 READ(IN,1031)    ((CUZ(I,J),J=1,NPU),I=1,NSTX)
      WRITE(OUT,1031) ((CUZ(I,J),J=1,NPU),I=1,NSTX)
      GO TO 8
  103 READ(IN,1031)    ((CVZ(I,J),J=1,NPV),I=1,NSTX)
      WRITE(OUT,1031) ((CVZ(I,J),J=1,NPV),I=1,NSTX)
      GO TO 8
  104 DO 105 I=1,NPTS
      READ(IN,1036)  LT,MT,P(LT,MT)
      P(MT,LT)=P(LT,MT)
 1036 FORMAT(4X,2I4,E12.4)
  105 WRITE(OUT,1037) LT,MT,P(LT,MT)
 1037 FORMAT(10X,2HP(,I3,1H,I3,3H) =E15.6)
      GO TO 8
  106 DO 107 I=1,NPTS
      READ(IN,1036) LT,MT,CUZ(LT,MT)
  107 WRITE(OUT,1038) LT,MT,CUZ(LT,MT)
 1038 FORMAT(8X,4HCUZ(,I3,1H,I3,3H) =E15.8)
      GO TO 8
  108 DO 109 I=1,NPTS
      READ(IN,1036) LT,MT,CVZ(LT,MT)
  109 WRITE(OUT,1039) LT,MT,CVZ(LT,MT)
 1039 FORMAT(8X,4HCVZ(,I3,1H,I3,3H) =E15.8)
      GO TO 8
 1008 FORMAT(I2,I4,2E12.4)
  110 DECODE(32,1008,B) NPC(11),IP,SREF,XLREF
      CALL AEROIN
      GO TO 8
  120 DECODE(32,1008,B)IP,N4(1),TONE,TTWO
      IP = N4(1)
      READ(IN,1031) (TMAS(I),XMAS(I),I=1,IP)
      WRITE(OUT,1041) (TMAS(I),XMAS(I),I=1,IP)
 1041 FORMAT(5(F10.2,E14.5))
      GO TO 8
  130 DECODE(6,1006,B) IPC,NPTS
      IF(IPC) 134,132,134
  132 N5(1) = NPTS
      READ(IN,1031) (TUW(I),XUW(I),I=1,NPTS)
      WRITE(OUT,1041)(TUW(I),XUW(I),I=1,NPTS)
      GO TO 8
  134 N6(1) = NPTS
      READ(IN,1031) (TVW(I),XVW(I),I=1,NPTS)
      WRITE(OUT,1041)(TVW(I),XVW(I),I=1,NPTS)
      GO TO 8
  140 DECODE(2,1006,B) NPC(14)
      IF(NPC(14)-1) 8,8,142
  142 READ(IN,1048)    K1,APO,AMO,AR,AGAM,ABET,ASU
 1048 FORMAT(I4,6E11.4)
      READ(IN,1031) (HB(I),I=1,K1)
      READ(IN,1031) (TMB(I),I=1,K1)
      GO TO 8
  150 GO TO 8
  160 GO TO 8
```

```
170 GO TO 8                                                         INDT1960
180 GO TO 8                                                         INDT1970
190 GO TO 8                                                         INDT1980
200 RETURN                                                          INDT1990
    END                                                             INDT2000
```

```
      SUBROUTINE INTAG
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),     COMM0010
     *     NALL,LRK                                                     COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP     ,CBET     ,CETA     ,  COMM0030
     1     CF(3),JCS ,CXZI     ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  ,  COMM0040
     2     DRDP,DWDH(2),ETA,HO,N4(3)     ,N5(3)     ,N6(3)     ,PAXP(3,30), COMM0050
     3     PE        ,ROE      ,ROEC     ,ROEM      ,SALP     ,SBET     ,  COMM0060
     4     SETA      ,SREF     ,SXZI     ,TAU       ,TMAS(20) ,TONE     ,  COMM0070
     5     TTWO      ,TUW(20)  ,TVW(20)  ,UW        ,VA(5)    ,VB(5)    ,  COMM0080
     6     VW                  ,XIND1    ,XIND2     ,XLREF    ,XM       ,  COMM0090
     7     XMAS(20)  ,XNU      ,XQ       ,XQSM      ,XRE      ,XTEMP    ,  COMM0100
     8     XUW(20)   ,XVW(20)  ,XZI      ,YMAS      ,YMASM               COMM0110
      COMMON /TWICE/ A(3,3)    ,CPH      ,CT(5)     ,SP(10)    ,ST(5)     COMM0120
     1     ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)     ,AMDOT(3) ,APDOT(4)    COMM0130
     2     ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)      COMM0140
     3     ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)     ,HI(2)       COMM0150
     4     ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2               COMM0160
     5     ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)       COMM0170
     6     ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)                 COMM0180
      DIMENSION DUB(30,15)                                                COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                        COMM0200
      COMMON /METH2/ AB(3)     ,AM(3)    ,AP(4)     ,AX(40)    ,AY(40)    COMM0210
     1     ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD     ,CPHDT(5) ,CUZ(30,5)   COMM0220
     2     ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)   ,DATAS(3) ,DATC(3)     COMM0230
     3     ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)    ,DZ(30)   ,ET          COMM0240
     4     ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J    ,JNBR,JNBRS,JST   COMM0250
     5     ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA     ,KDATAS   ,KG,KI,KK    COMM0260
     6     ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT     COMM0270
     7     ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)     ,MR(3,9)  ,MT          COMM0280
     8     ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT    ,NPC(15)     COMM0290
     9     ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)    COMM0300
     A     ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)    ,OMEGA    ,P(30,30)    COMM0310
     B     ,PA(3)              ,PM(3)    ,R         ,REO      ,RERP2       COMM0320
     C     ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5)  ,SI(3)    ,SIG(3,40)   COMM0330
     D     ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL       COMM0340
     E     ,TFIT(9)  ,TIME     ,TIMES    ,TO        ,TP(40)   ,TQ(40)      COMM0350
     F     ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20)  ,TYM(40)  ,TZCG(20)    COMM0360
     G     ,TZERO    ,XJ2      ,XMU      ,XMUJ      ,XP(3)    ,XXCG(20)    COMM0370
     H     ,XYCG(20) ,XZCG(20)                                          COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                  COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                       COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK     COMM0410
     A   ,PB(23),RL(23),TMB(23)                                         COMM0420
      COMMON /EXTRA / GO                                                COMM0430
    2 LRK = 4                                                           INTG0010
    6 CALL RKUTTA                                                       INTG0020
      IF(LRK-2) 8,10,99                                                 INTG0030
    8 CALL DATAB                                                        INTG0040
      CALL MOTION                                                       INTG0050
      IF(NPC(1)-1) 12,6,12                                              INTG0060
   10 IF(DCOMP.GE.0..OR.NPC(10).EQ.0.OR.T.EQ.TSAV) GO TO 6              INTG0070
      WRITE(STATE) T,X                                                  INTG0090
      TSAV=T                                                            INTG0095
      GO TO 6                                                           INTG0100
   12 CALL FXXU                                                         INTG0110
      DO 22 I=1,4                                                       INTG0120
      N = NSTX+NPU                                                      INTG0130
      DO 22 J=1,N                                                       INTG0140
      IF(J-10) 13,13,14                                                 INTG0150
```

```
13 CONTINUE
   SUM = 0.
   SUM2= 0.
   GO TO 15
14 SUM = F1(I,J)
   SUM2= F3(I,J)
15 DO 16 K=1,10
16 SUM2 = SUM2 + F3(I,K)*PH(K,J)
   DPH(I+6,J) = SUM2
   IF(I.EQ.4) GO TO 22
   DO 18 K=1,10
18 SUM = SUM + F1(I,K)*PH(K,J)
   DPH(I,J) = SUM
   SUM = 0.
   DO 20 K=1,5
20 SUM = SUM + F2(I,K)*PH(K,J)
   DPH(I+3,J) = SUM
22 CONTINUE
   GO TO 6
99 RETURN
   END
```

```
      SUBROUTINE MINVAR
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),      COMM0010
     *    NALL,LRK                                                        COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET      ,CETA      , COMM0030
     1    CF(3),JCS ,CXZI      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH   , COMM0040
     2    DRDP,DWDH(2),ETA,HO,N4(3)       ,N5(3)     ,N6(3)     ,PAXP(3,30), COMM0050
     3    PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET      , COMM0060
     4    SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE      , COMM0070
     5    TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     , COMM0080
     6    VW                   ,XIND1     ,XIND2     ,XLREF     ,XM        , COMM0090
     7    XMAS(20)  ,XNU       ,XQ        ,XQSM      ,XRE       ,XTEMP     , COMM0100
     8    XUW(20)   ,XVW(20)   ,XZI       ,YMAS      ,YMASM                  COMM0110
      COMMON /TWICE/ A(3,3)    ,CPH       ,CT(5)     ,SP(10)    ,ST(5)     COMM0120
     1    ,TPH      ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)  COMM0130
     2    ,B(8)     ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1) ,DUE(4)    COMM0140
     3    ,DUF(4)   ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)     COMM0150
     4    ,IDN  ,II ,IP   ,IPC ,JJ   ,JN  ,KA  ,KD(3),KDUM  ,KG2             COMM0160
     5    ,K1   ,NNN ,NPTS     ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)    COMM0170
     6    ,RO       ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)                 COMM0180
      DIMENSION DUB(30,15)                                                COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                        COMM0200
      COMMON /METH2/ AB(3)     ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)    COMM0210
     1    ,AZ(40)   ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5) COMM0220
     2    ,CVZ(30,5),D(5)      ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)   COMM0230
     3    ,DET      ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET        COMM0240
     4    ,FLOS     ,G(3,30)   ,H(3,5)    ,I  ,ICOUNT,J    ,JNBR,JNBRS,JST  COMM0250
     5    ,K    ,KAR ,KC(3)    ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK    COMM0260
     6    ,KN   ,KOB ,KPROP,KS,KSM,KSS,L  ,LC(3)     ,LCS(3)    ,LS    ,LT  COMM0270
     7    ,L1   ,M   ,MC(5)    ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT        COMM0280
     8    ,MTP(40)  ,MTYP      ,MTYPS     ,N  ,NC(30),NCOUNT    ,NPC(15)   COMM0290
     9    ,NPU  ,NPV ,NS(9)    ,NSS(5)    ,NST  ,NSTA,NSTC,NSTX,NT,NTR(9)  COMM0300
     A    ,N8(3)    ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)  COMM0310
     B    ,PA(3)               ,PM(3)     ,R         ,REO       ,RERP2     COMM0320
     C    ,RPO      ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40) COMM0330
     D    ,SIGM(3)  ,SIGMS(3)  ,SPHDT(5)  ,SUM  ,SUM2,SYG(3,9)  ,TFINAL    COMM0340
     E    ,TFIT(9)  ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)    COMM0350
     F    ,TR(40)   ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)  COMM0360
     G    ,TZERO    ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)  COMM0370
     H    ,XYCG(20) ,XZCG(20)                                              COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                    COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                         COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK       COMM0410
     A   ,PB(23),RL(23),TMB(23)                                           COMM0420
      COMMON /EXTRA / GO                                                  COMM0430
      G(1,10) = 0.                                                        MINV0010
      G(2,10) = 0.                                                        MINV0020
      G(3,10) = 0.                                                        MINV0030
      NT = 0                                                              MINV0040
      LT = 1                                                              MINV0050
      MT = NSTX                                                           MINV0060
      DO 1 I=1,NSTX                                                       MINV0070
      P(I,10) = 0.                                                        MINV0080
    1 P(10,I) = 0.                                                        MINV0090
    2 DO 12 J=KAR,KOB                                                     MINV0100
      DO 6 I=LT,MT                                                        MINV0110
      L = I+1                                                             MINV0120
      SUM = 0.                                                            MINV0130
      DO 3 K=LT,I                                                         MINV0140
    3 SUM = SUM + P(K,I)*G(J,K)                                           MINV0150
```

```
      IF(L.GT.MT) GO TO 6
      DO 4 K=L,MT
    4 SUM = SUM+P(I,K)*G(J,K)
    6 DUB(I,J) = SUM
C-------------CALCULATE EQ.(54F) --------------------------
      DO 10 I=KAR,KOB
      SUM = 0.
      DO  8 K=LT,MT
    8 SUM = SUM+G(I,K)*DUB(K,J)
   10 DUB(I,J+6) = SUM
   12 DUB(J,J+6) = DUB(J,J+6)+SI(J)**2
      IF(NPV) 36,36,14
   14 CONTINUE
      DO 15 I=1,NPV
   15 CVZ(10,I) = 0.
      DO 20 J=KAR,KOB
      DO 17 I=1,NSTX
      SUM= 0.
      DO 16 K=1,NPV
   16 SUM= SUM+CVZ(I,K)*H(J,K)
      DUB(I,J+3) = SUM
   17 DUB(I,J)     = DUB(I,J)+SUM
      DO 18 I=1,NPV
   18 DUB(I+12,J+6) = S(I)*H(J,I)
   20 CONTINUE
      DO 26 J=KAR,KOB
      DO 24 I=KAR,KOB
      SUM = 0.
      SUM2= 0.
      DO 22 K=LT,MT
   22 SUM = SUM + DUB(K,J+3)*G(I,K)
      DO 23 K=1,NPV
   23 SUM2= SUM2+ H(I,K)*DUB(K+12,J+6)
      DUB(J+3,I+6) = SUM
   24 DUB(I+6,J+6) = SUM + SUM2
   26 CONTINUE
      DO 34 J=KAR,KOB
      DO 28 I=KAR,KOB
   28 DUB(I,J+6) = DUB(I,J+6) + DUB(I+3,J+6) + DUB(I+6,J+6)
      DO 32 I=1,NPV
      SUM = 0.
      DO 30 K=LT,MT
   30 SUM = SUM + G(J,K)*CVZ(K,I)
   32 DUB(I,J+9) = SUM
   34 CONTINUE
   36 CONTINUE
      IF(KAR-KOB) 37,46,37
   37 CONTINUE
      DO 39 J=KAR,KOB
      DO 38 I=KAR,KOB
   38 DUB(I+9,J+6) = 0.
   39 DUB(J+9,J+6) = 1.
      L = KOB-1
      DO 42 I=KAR,L
      M  = I+1
      SUM  = DUB(I,I+6)
      DO 42 J=KAR,KOB
      DUB(I   ,J+6) = DUB(I   ,J+6)/SUM
      DUB(I+9,J+6) = DUB(I+9,J+6)/SUM
      DO 42 K=M,KOB
```

```
      IF(I-J) 40,41,40
   40 DUB(K   ,J+6) = DUB(K   ,J+6)-DUB(K,I+6)*DUB(I   ,J+6)
   41 DUB(K+9,J+6) = DUB(K+9,J+6)-DUB(K,I+6)*DUB(I+9,J+6)
   42 CONTINUE
      DO 44 I=KAR,KOB
   44 DUB(KOB+9,I+6) = DUB(KOB+9,I+6)/DUB(KOB,KOB+6)
      DO 45 I=KAR,L
      M = KOB+KAR+6-I
      DO 45 J=I,L
      N = KOB+KAR+8-J
      DO 45 K=KAR,KOB
   45 DUB(N,K+6) = DUB(N,K+6)-DUB(N-9   ,M)*DUB(M+3,K+6)
      GO TO 48
   46 DUB(KAR+9,KAR+6) = 1./DUB(KAR,KAR+6)
   48 CONTINUE
C-------------CALCULATE EQ.(54E) --------------------------
      DO 54 I=1,NSTX
      DO 51 J=KAR,KOB
      SUM = 0.
      DO 50 K=KAR,KOB
   50 SUM = SUM + DUB(I,K)*DUB(K+9,J+6)
   51 DUB(I,J+12) = SUM
      DO 54 L=I,NSTX
      SUM = 0.
C-------------CALCULATE EQ.(54B) --------------------------
      DO 52 K=KAR,KOB
   52 SUM = SUM + DUB(I,K+12)*DUB(L,K)
      P(I,L) = P(I,L) - SUM
   54 CONTINUE
      IF(NPU) 66,66,56
C-------------CALCULATE EQ.(54C) --------------------------
   56 CONTINUE
      DO 57 I=1,NPU
   57 CUZ(10,I) = 0.
      DO 64 J=1,NPU
      DO 60 I=KAR,KOB
      SUM = 0.
      DO 58 K=1,NSTX
   58 SUM = SUM + G(I,K)*CUZ(K,J)
   60 DUB(I,1) = SUM
      DO 64 I=1,NSTX
      SUM = 0.
      DO 62 K=KAR,KOB
   62 SUM = SUM + DUB(I,K+12)*DUB(K,1)
   64 CUZ(I,J) = CUZ(I,J)-SUM
   66 IF(NPV) 72,72,67
C-------------CALCULATE EQ.(54D) --------------------------
   67 DO 70 I=1,NSTX
      DO 70 J=1,NPV
      SUM = 0.
      DO 68 K=KAR,KOB
   68 SUM = SUM + DUB(I,K+12)*DUB(J,K+9)
   70 CVZ(I,J) = CVZ(I,J) - SUM
   72 IF(NPC(1)-2) 74,108,74
   74 IF(NPC(4)) 75,88,75
   75 DO 82 I=1,NSTX
   80 DUB(I,1) = DZ(I)
   82 CONTINUE
C-------------CALCULATE EQ.(54A) --------------------------
      DO 86 I=KAR,KOB
```

```
      SUM = 0.
      DO 84 J=1,NSTX
   84 SUM = SUM + G(I,J)*DUB(J,1)
   86 DATC(I) = DATC(I)+SUM
   88 DO 90 I=KAR,KOB
      J = LC(I)
      DUB(I,2) = DATA(J)-DATC(I)
      IF(MTYP.GT.5) GO TO 90
      IF(J.EQ.1) GO TO 90
      IF(ABS(DUB(I,2)).LT.3.1416) GO TO 90
      DUB(I,2) = DUB(I,2)-SIGN(6.283185307179586,DUB(I,2))
   90 CONTINUE
      DO 92 I=1,NSTX
      DZ(I) = 0.
      DO 92 J=KAR,KOB
   92 DZ(I) = DZ(I)+DUB(I,J+12)*DUB(J,2)
      IF(NPC(4)) 294,298,294
  294 DO 296 I=1,4
  296 VO(I) = X(I+6) + DUB(I+6,1)
      DZ(10) = -(VO(1)*DZ(7)+VO(2)*DZ(8)+VO(3)*DZ(9))/VO(4)
   94 DO 96 I=1,NSTX
   96 DZ(I) = DZ(I)+DUB(I,1)
      GO TO 108
  298 DZ(10) = -(X(7)*DZ(7)+X(8)*DZ(8)+X(9)*DZ(9))/X(10)
   98 DO 100 I=1,10
  100 X(I) = X(I)+DZ(I)
      DATC(1)= SQRT(X(7)**2 + X(8)**2 + X(9)**2 + X(10)**2)
      X(7) = X(7)/DATC(1)
      X(8) = X(8)/DATC(1)
      X(9) = X(9)/DATC(1)
      X(10)=X(10)/DATC(1)
      IF(NSTC.EQ.0) GO TO 103
      DO 102 I=1,NSTC
      J = NC(I)
  102 C(J) = C(J)+DZ(I+10)
  103 CONTINUE
      IF(KSS) 108,108,104
  104 DO 106 I=1,KSS
      MTYP = NSS(I)
  106 CALL STAT
  108 P(8,7) = P(7,8)
      P(9,7) = P(7,9)
      P(9,8) = P(8,9)
      DO 110 I=1,9
      SUM = P(I,7)*X(7) + P(I,8)*X(8) + P(I,9)*X(9)
  110 P(I,10) =-SUM/X(10)
      P(10,10) = -(X(7)*P(7,10)+X(8)*P(8,10)+X(9)*P(9,10))/X(10)
      DO 114 I=11,NSTX
      SUM = P(7,I)*X(7) + P(8,I)*X(8) + P(9,I)*X(9)
  114 P(10,I) =-SUM/X(10)
      IF(NPU.EQ.0) GO TO 117
  115 DO 116 I=1,NPU
      SUM = CUZ(7,I)*X(7) + CUZ(8,I)*X(8) + CUZ(9,I)*X(9)
  116 CUZ(10,I) =-SUM/X(10)
  117 IF(NPV.EQ.0)GO TO 99
      DO 118 I=1,NPV
      SUM = CVZ(7,I)*X(7) + CVZ(8,I)*X(8) + CVZ(9,I)*X(9)
  118 CVZ(10,I) =-SUM/X(10)
   99 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MOTION
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),       COMM0010
     *    NALL,LRK                                                       COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET     ,CETA     ,  COMM0030
     1    CF(3),JCS ,CXZI      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  , COMM0040
     2    DRDP,DWDH(2),ETA,HO,N4(3)      ,N5(3)     ,N6(3)     ,PAXP(3,30),COMM0050
     3    PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET     , COMM0060
     4    SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE     , COMM0070
     5    TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)    , COMM0080
     6    VW                   ,XIND1     ,XIND2     ,XLREF     ,XM       , COMM0090
     7    XMAS(20)  ,XNU       ,XQ        ,XQSM      ,XRE       ,XTEMP    , COMM0100
     8    XUW(20)   ,XVW(20)   ,XZI       ,YMAS      ,YMASM                 COMM0110
      COMMON /TWICE/ A(3,3)    ,CPH       ,CT(5)     ,SP(10)    ,ST(5)      COMM0120
     1    ,TPH      ,V0(20)    ,Z0(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)   COMM0130
     2    ,B(8)     ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)      COMM0140
     3    ,DUF(4)   ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)      COMM0150
     4    ,IDN ,II  ,IP   ,IPC ,JJ   ,JN  ,KA  ,KD(3),KDUM ,KG2             COMM0160
     5    ,K1   ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)    COMM0170
     6    ,RO        ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)                COMM0180
      DIMENSION DUB(30,15)                                                COMM0190
      EQUIVALENCE (AB(1),DUB(1,1))                                        COMM0200
      COMMON /METHD2/ AB(3)    ,AN(3)     ,AP(4)     ,AX(40)    ,AY(40)   COMM0210
     1    ,AZ(40)   ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5)  COMM0220
     2    ,CVZ(30,5),D(5)      ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)    COMM0230
     3    ,DET      ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET         COMM0240
     4    ,FLOS     ,G(3,30)   ,H(3,5)    ,I  ,ICOUNT,J    ,JNBR,JNBRS,JST  COMM0250
     5    ,K    ,KAR ,KC(3)    ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK   COMM0260
     6    ,KN   ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)     ,LCS(3)    ,LS    ,LT   COMM0270
     7    ,L1   ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT        COMM0280
     8    ,MTP(40)  ,MTYP      ,MTYPS     ,N  ,NC(30),NCOUNT    ,NPC(15)    COMM0290
     9    ,NPU ,NPV ,NS(9)     ,NSS(5)    ,NST  ,NSTA,NSTC,NSTX,NT,NTR(9)   COMM0300
     A    ,N8(3)    ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)   COMM0310
     B    ,PA(3)               ,PM(3)     ,R         ,REO       ,RERP2      COMM0320
     C    ,RPO      ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40)  COMM0330
     D    ,SIGM(3)  ,SIGMS(3)  ,SPHDT(5)  ,SUM  ,SUM2,SYG(3,9)  ,TFINAL     COMM0340
     E    ,TFIT(9)  ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)     COMM0350
     F    ,TR(40)   ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)   COMM0360
     G    ,TZERO    ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)   COMM0370
     H    ,XYCG(20) ,XZCG(20)                                             COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                    COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                         COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK       COMM0410
     A  ,PB(23),RL(23),TMB(23)                                            COMM0420
      COMMON /EXTRA / GO                                                  COMM0430
      DATA ITER /5/                                                       MOTN0010
      LT = 0                                                              MOTN0020
    1 CONTINUE                                                            MOTN0030
      CPH = COS(X(5))                                                     MOTN0040
      SPH = SIN(X(5))                                                     MOTN0050
      TPH = SPH/CPH                                                       MOTN0060
      R = REO + X(4)                                                      MOTN0070
C-------------CALCULATE EQ.(149) ---------------------------              MOTN0080
      YMASM=TAB(T,N4(1),TMAS(1),XMAS(1))                                  MOTN0090
      IF(T.GT.TONE) GO TO 3                                               MOTN0100
      YMAS =YMASM                                                         MOTN0110
      GO TO 6                                                             MOTN0120
    3 IF(T.LT.TTWO) GO TO 4                                               MOTN0130
      TAU=TTWO-TONE                                                       MOTN0140
      GO TO 5                                                             MOTN0150
```

```
    4 TAU=T    -TONE
    5 CONTINUE
      YMAS=YMASM-(C(16)+C(17)*TAU+C(18)*TAU**2)
C-------------CALCULATE EQ.(145) --------------------------
    6 RO=REO/SQRT(1.+(RERP2-1.)*SPH**2)
C-------------CALCULATE EQ.(144) --------------------------
      HO = R-RO
      CALL ATMDAT(HO,PE,ROEM,AS,XTEMP,XNU,SP(1))
C-------------CALCULATE EQ.(151) --------------------------
      ROEC=C(22)*EXP(C(23)*HO)
      ROE =C(21)*ROEM+ROEC
C-------------CALCULATE EQ.(152) --------------------------
      UW  = TAB(HO,N5(1),TUW(1),XUW(1)) + C(26)
      VW  = TAB(HO,N6(1),TVW(1),XVW(1)) + C(27)
C-------------CALCULATE EQ.(140) --------------------------
      VA(1)=X(1)+UW
      VA(2)=X(2)-R*OMEGA*CPH+VW
      VA(3)=X(3)
C-------------CALCULATE EQ.(141) --------------------------
      VA(5)=VA(1)**2+VA(2)**2+VA(3)**2
      VA(4)=SQRT(VA(5))
C-------------CALCULATE EQ.(142) --------------------------
      XQ=.5*ROE*VA(5)
C-------------CALCULATE EQ.(148) --------------------------
      XRE=ROE*VA(4)*XLREF/XNU
C-------------CALCULATE EQ.(143) --------------------------
      XM=VA(4)/AS
      XQSM=XQ*SREF/YMAS
C-------------CALCULATE EQ.(133) --------------------------
      SP(1) = (X( 7)-X( 8))*(X( 7)+X( 8))
      SP(2) = (X( 9)-X(10))*(X( 9)+X(10))
      A(1,1) = (X( 7)-X( 9))*(X( 7)+X( 9)) + (X( 8)-X(10))*(X( 8)+X(10))
      A(2,2) = SP(1) + SP(2)
      A(3,3) = SP(1) - SP(2)
      A(1,2) = 2.*(X( 8)*X( 9)+X( 7)*X(10))
      A(2,1) = 2.*(X( 8)*X( 9)-X( 7)*X(10))
      A(1,3) = 2.*(X( 8)*X(10)-X( 7)*X( 9))
      A(3,1) = 2.*(X( 8)*X(10)+X( 7)*X( 9))
      A(3,2) = 2.*(X(10)*X( 9)-X( 7)*X( 8))
      A(2,3) = 2.*(X(10)*X( 9)+X( 7)*X( 8))
C-------------CALCULATE EQ.(139) --------------------------
      VB(1)=0.
      VB(2)=0.
      VB(3)=0.
      DO 7 I=1,3
      DO 7 J=1,3
    7 VB(I)=VB(I)+A(I,J)*VA(J)
      VB(4)=SQRT(VB(1)**2+VB(3)**2)
C-------------CALCULATE EQ.(135) --------------------------
      SALP=VB(3)/VB(4)
      CALP=VB(1)/VB(4)
C-------------CALCULATE EQ.(136) --------------------------
      SBET=VB(2)/VA(4)
      CBET=VB(4)/VA(4)
      ALPH =ATAN2(SALP,CALP)
      BETA =ATAN2(SBET,CBET)
      IF(NPC(11).EQ.1) GO TO 8
      XIND1=ALPH
      XIND2=BETA
      GO TO 9
```

```
    8 VB(5)=SQRT(VB(2)**2+VB(3)**2)
C------------CALCULATE EQ.(137) -------------------------
      SETA=VB(5)/VA(4)
      CETA=VB(1)/VA(4)
C------------CALCULATE EQ.(138) -------------------------
      SXZI=VB(2)/VB(5)
      CXZI=VB(3)/VB(5)
      ETA  =ATAN2(SETA,CETA)
      XZI  =ATAN2(SXZI,CXZI)
      XIND1=ETA
      XIND2=XZI
    9 CALL AERO(XIND1,XIND2,XM,XRE,CF(1),CF(2),CF(3))
      AB(1)=CF(1)*XQSM
      AB(2)=CF(2)*XQSM
      AB(3)=CF(3)*XQSM
      XP(1) = CGM(1) + C(31)
      XP(2) = CGM(2) + C(32)
      XP(3) = CGM(3) + C(33)
C------------CALCULATE EQ.(153) -------------------------
      I=N11(2)
      IF(N11(2)-1)12,10,12
   10 I = 2
   12 CONTINUE
      SP(1)= TT(I+1)-TT(I-1)
      HI(2)= .5*SP(1)
      HI(1)= T    + HI(2)
      HI(2)= T    - HI(2)
      PMDOT(1)= (TAB(HI(1),N11(1),TT(1),TP(1))-TAB(HI(2),N11(1),TT(1),TP
     X(1)))/SP(1)
      PMDOT(2)= (TAB(HI(1),N11(1),TT(1),TQ(1))-TAB(HI(2),N11(1),TT(1),TQ
     X(1)))/SP(1)
      PMDOT(3)= (TAB(HI(1),N11(1),TT(1),TR(1))-TAB(HI(2),N11(1),TT(1),TR
     X(1)))/SP(1)
C------------CALCULATE EQ.(156) -------------------------
   13 DO 14 I=1,3
      J = 3*I
      PDOT(I) = C(J+33)*PMDOT(1) + C(J+34)*PMDOT(2) + C(J+35)*PMDOT(3)
   14 CONTINUE
C     SOLVE EQ.(III.114)
      AP(1)=AB(1)
      AP(2)=AB(2)
      AP(3)=AB(3)
      AP(4) = 2.*AP(1)*AP(3)/SQRT(AP(1)**2+AP(3)**2)
C------------CALCULATE EQ.(158) -------------------------
      DO 16 N = 1,ITER
      DO 11 I=1,3
      J = 3*I
      PA(I)=C(J+33)*PM(1)+C(J+34)*PM(2)+C(J+35)*PM(3)+C(I+44)+C(I+47)*
     X    AP(1) + C(I+50)*AP(2) + C(I+53)*AP(3)
   11 CONTINUE
      PA(1) = C(57)*AP(4)+PA(1)
      SP(1) = PA(1)**2
      SP(2) = PA(2)**2
      SP(3) = PA(3)**2
      TRAN(1,1)=-SP(2)-SP(3)
      TRAN(1,2)=  PA(1)*PA(2)-PDOT(3)
      TRAN(1,3)=  PA(1)*PA(3)+PDOT(2)
      TRAN(2,1)=  PA(1)*PA(2)+PDOT(3)
      TRAN(2,2)=-SP(1)-SP(3)
      TRAN(2,3)=  PA(2)*PA(3)-PDOT(1)
```

```
      TRAN(3,1)=  PA(1)*PA(3)-PDOT(2)
      TRAN(3,2)=  PA(2)*PA(3)+PDOT(1)
      TRAN(3,3)=-SP(1)-SP(2)
      AP(1)=AB(1)+TRAN(1,1)*XP(1)+TRAN(1,2)*XP(2)+TRAN(1,3)*XP(3)
      AP(2)=AB(2)+TRAN(2,1)*XP(1)+TRAN(2,2)*XP(2)+TRAN(2,3)*XP(3)
      AP(3)=AB(3)+TRAN(3,1)*XP(1)+TRAN(3,2)*XP(2)+TRAN(3,3)*XP(3)
      AP(4) = 2.*AP(1)*AP(3)/SQRT(AP(1)**2+AP(3)**2)
   16 CONTINUE
      IF(LT) 15,15,99
   15 CONTINUE
C-------------CALCULATE EQ.(132) ---------------------------
      AG(1)  = AB(1)*A(1,1)+AB(2)*A(2,1)+AB(3)*A(3,1)
      AG(2)  = AB(1)*A(1,2)+AB(2)*A(2,2)+AB(3)*A(3,2)
      AG(3)  = AB(1)*A(1,3)+AB(2)*A(2,3)+AB(3)*A(3,3)
C     PARAMETERS USED FREQUENTLY IN REMAINING SUBROUTINES
      PAR(1) = X(1)*X(1) + X(2)*X(2)
      PAR(2)=XMU/R**2
      PAR(3)=XMUJ/R**4
      PAR(4) = 1./(2.*R)
      PAR(5) = X(1)*X(3)-X(2)*X(2)*TPH
      PAR(6) = 1. - 1.5*CPH*CPH
      F1(2,2)= (X(1)*TPH+X(3))/R
      F1(2,3)= X(2)/R
      F1(2,1)= F1(2,3)*TPH
      DX(5) = X(1)/R
      F3(1,8) =-.5*(PA(1)-F1(2,3))
      F3(1,9) =-.5*(PA(2)+DX(5))
      F3(1,10)=-.5*(PA(3)+F1(2,1))
      F3(2,9) = F3(1,10) + PA(3)
      F3(2,10)= F3(1,9)  + DX(5)
      F3(3,10)= F3(1,8)  + PA(1)
C-------------CALCULATE EQ.(127) ---------------------------
      DX(1) = AG(1) + PAR(5)/R          - PAR(3)*SPH*CPH
      DX(2) = AG(2) + X(2)*F1(2,2)
      DX(3) = AG(3) + PAR(2) - PAR(1)/R - PAR(3)*PAR(6)
C-------------CALCULATE EQ.(128) ---------------------------
      DX(4) = -X(3)
      DX(6) = F1(2,3)/CPH - OMEGA
C-------------CALCULATE EQ.(129) ---------------------------
      DX( 7) = X( 8)*F3(1, 8) + X( 9)*F3(1, 9) + X(10)*F3(1,10)
      DX( 8) =-X( 7)*F3(1, 8) + X( 9)*F3(2, 9) + X(10)*F3(2,10)
      DX( 9) =-X( 7)*F3(1, 9) - X( 8)*F3(2, 9) + X(10)*F3(3,10)
      DX(10) =-X( 7)*F3(1,10) - X( 8)*F3(2,10) - X( 9)*F3(3,10)
   99 RETURN
      ENTRY AUXIL
      LT = 1
      GO TO 1
      END
```

```
      SUBROUTINE OBSERV
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),      COMM0010
     *     NALL,LRK                                                       COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET      ,CETA      , COMM0030
     1     CF(3),JCS ,CXZI     ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH   , COMM0040
     2     DRDP,DWDH(2),ETA,HO,N4(3)      ,N5(3)     ,N6(3)     ,PAXP(3,30), COMM0050
     3     PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET      , COMM0060
     4     SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE      , COMM0070
     5     TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     , COMM0080
     6     VW                   ,XIND1     ,XIND2     ,XLREF     ,XM        , COMM0090
     7     XMAS(20)  ,XNU       ,XQ        ,XQSM      ,XRE       ,XTEMP     , COMM0100
     8     XUW(20)   ,XVW(20)   ,XZI       ,YMAS      ,YMASM                  COMM0110
      COMMON /TWICE/ A(3,3)     ,CPH       ,CT(5)     ,SP(10)    ,ST(5)     COMM0120
     1     ,TPH       ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)  COMM0130
     2     ,B(8)      ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)    COMM0140
     3     ,DUF(4)    ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)     COMM0150
     4     ,IDN ,II   ,IP   ,IPC ,JJ   ,JN  ,KA  ,KD(3),KDUM ,KG2            COMM0160
     5     ,K1   ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)    COMM0170
     6     ,RO        ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)                 COMM0180
      DIMENSION DUB(30,15)                                                COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                        COMM0200
      COMMON /METH2/ AB(3)      ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)    COMM0210
     1     ,AZ(40)    ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5) COMM0220
     2     ,CVZ(30,5),D(5)       ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)   COMM0230
     3     ,DET       ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET        COMM0240
     4     ,FLOS      ,G(3,30)   ,H(3,5)    ,I ,ICOUNT,J    ,JNBR,JNBRS,JST  COMM0250
     5     ,K    ,KAR ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK  COMM0260
     6     ,KN   ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)     ,LCS(3)    ,LS    ,LT   COMM0270
     7     ,L1   ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT        COMM0280
     8     ,MTP(40)   ,MTYP      ,MTYPS     ,N  ,NC(30),NCOUNT    ,NPC(15)   COMM0290
     9     ,NPU  ,NPV ,NS(9)     ,NSS(5)    ,NST  ,NSTA,NSTC,NSTX,NT,NTR(9)  COMM0300
     A     ,N8(3)     ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)  COMM0310
     B     ,PA(3)                ,PM(3)     ,R         ,REO       ,RERP2     COMM0320
     C     ,RPO       ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40) COMM0330
     D     ,SIGM(3)   ,SIGMS(3)  ,SPHDT(5)  ,SUM  ,SUM2,SYG(3,9)  ,TFINAL    COMM0340
     E     ,TFIT(9)   ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)    COMM0350
     F     ,TR(40)    ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)  COMM0360
     G     ,TZERO     ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)  COMM0370
     H     ,XYCG(20)  ,XZCG(20)                                           COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                    COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                         COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK       COMM0410
     A  ,PB(23),RL(23),TMB(23)                                            COMM0420
      COMMON /EXTRA / GO                                                  COMM0430
      DIMENSION      CTM(6)    ,RCC(3)    ,RCDC(3)   ,RCDX(3,6),RCX(3,6) , OBSV0010
     A     XA(5)     ,XAX(5,7) ,XC(3)     ,XCX(3,7)  ,XLB(3)    ,XLG(3)    , OBSV0020
     B     XLGE(3,4),XLGX(2,4),XS(3)      ,XSC(3)    ,XSD(3)    ,XSDC(3)   , OBSV0030
     C     XSDX(3,6),XSX(3,3) ,CI(3,3)    ,CJ(3,3)                          OBSV0040
      EQUIVALENCE    (DPH(1,1) ,CTM(1)     ),    (DPH(7,1) ,RCC(1)      ),  OBSV0050
     N               (DPH(10,1),RCDC(1)    ),    (DPH(3,2) ,RCDX(1,1)),     OBSV0060
     O               (DPH(1,4) ,RCX(1,1) ),      (DPH(9,5) ,XA(1)       ),  OBSV0070
     P               (DPH(4,6) ,XAX(1,1) ),      (DPH(9,9) ,XC(1)       ),  OBSV0080
     Q               (DPH(2,10),XCX(1,1) ),      (DPH(3,12),XLB(1)      ),  OBSV0090
     R               (DPH(6,12),XLG(1)     ),    (DPH(9,12),XLGE(1,1)),     OBSV0100
     S               (DPH(1,14),XLGX(1,1)),      (DPH(9,14),XS(1)       ),  OBSV0110
     T               (DPH(2,15),XSC(1)     ),    (DPH(5,15),XSD(1)      ),  OBSV0120
     U               (DPH(8,15),XSDC(1)    ),    (DPH(1,16),XSDX(1,1)),     OBSV0130
     V               (DPH(9,17),XSX(1,1) ),      (DPH(8,18),CI(1,1)     ),  OBSV0140
     W               (DPH(7,19),CJ(1,1)    ),    (DPH(6,20),CAZC        ),  OBSV0150
```

```
     X               (DPH(7,20),CDP       ),    (DPH(8,20),CDY        ),
     Y               (DPH(9,20),CELC      ),    (DPH(10,20),CGR       ),
     Z               (DPH(1,21),CLR       ),    (DPH(2,21),CTH        ),
     1               (DPH(3,21),SAZC      ),    (DPH(4,21),SDP        ),
     2               (DPH(5,21),SDY       ),    (DPH(6,21),SELC       ),
     3               (DPH(7,21),SGR       ),    (DPH(8,21),SLR        ),
     4               (DPH(9,21),STH       ),    (DPH(10,21),THETA     )
      MT = 0
      IF(L1-2) 14,10,14
   10 IF(NPC(5)) 14,12,14
   12 WRITE(OUT,1000)
 1000 FORMAT(5X,6HPOINTS,5X,4HTYPE,4X,4HTIME,8X,5HCOMP1,10X,5HCOMP2,10X,
     15HCOMP3,10X,4HSIG1,11X,4HSIG2,11X,4HSIG3)
      L1=1
   14 IF(MTYP-6)  16,150,15
   15 IF(MTYP-8) 115,120,125
   16 N = 60+MTYP*15
C-------------CALCULATE EQ.(236) -------------------------
  100 THETA=X(6)-C(N+14)
      STH=SIN(THETA)
      CTH=COS(THETA)
      VO(1)= CPH*CPHDT(MTYP)
      VO(2)= CPH*SPHDT(MTYP)
      VO(3)= SPH*CPHDT(MTYP)
      VO(4)= SPH*SPHDT(MTYP)
      SP(1) = ROT(MTYP) + C(N+15)
      VO(5) = SP(1)*CCAPH(MTYP)
      VO(6) = SP(1)*SCAPH(MTYP)
C-------------CALCULATE EQ.(235) -------------------------
  102 XS(1)= R*(CTH*VO(1)+VO(4))-VO(5)
      XS(2)= R*STH*CPH
      XS(3)=-R*(CTH*VO(2)-VO(3))+VO(6)
      VO(8)= SQRT(XS(2)**2+XS(3)**2)
      VO(9)= VO(8)**2
C-------------CALCULATE EQ.(238A)-------------------------
      CTM(1)= SQRT(XS(1)**2+VO(9))
C-------------CALCULATE EQ.(238B)-------------------------
      SAZC= XS(2)/VO(8)
      CAZC= XS(3)/VO(8)
C-------------CALCULATE EQ.(238C)-------------------------
      SELC= XS(1)/CTM(1)
      CELC= VO(8)/CTM(1)
      CTM(2)= ATAN2(XS(2),XS(3))
      CTM(3)= ATAN2(XS(1),VO(8))
C-------------CALCULATE EQ.(239) -------------------------
      DFM(1)= C(N+1)*CTM(1)+C(N+4)+C(N+10)/SELC
      DFM(2)= C(N+2)*CTM(2)+C(N+5)
      DFM(3)= C(N+3)*CTM(3)+C(N+6)+C(N+11)*CELC/SELC
      IF(MO(MTYP)) 106,104,106
  104 IF(NPC(1)-1) 106,18,106
C-------------CALCULATE EQ.(246) -------------------------
  106 XSX(1,1)= (XS(1)+VO(5))/R
      XSX(1,2)= R*(VO(2)-CTH*VO(3))
      XSX(1,3)= -XS(2)*CPHDT(MTYP)
      XSX(2,1)= XS(2)/R
      XSX(2,2)= -R*STH*SPH
      XSX(2,3)=  R*CTH*CPH
      XSX(3,1)= (XS(3)-VO(6))/R
      XSX(3,2)= R*(CTH*VO(4)+VO(1))
      XSX(3,3)= XS(2)*SPHDT(MTYP)
```

```
      IF(MO(MTYP)) 108,18,108
C------------CALCULATE EQ.(241) -------------------------
  108 DO 112 I=1,3
      SUM=0.
      DO 110 J=1,3
  110 SUM=SUM+XSX(I,J)*DX(J+3)
  112 XSD(I)=SUM
C------------CALCULATE EQ.(240) -------------------------
      CTM(4)= (XS(1)*XSD(1)+XS(2)*XSD(2)+XS(3)*XSD(3))/CTM(1)
      CTM(5)= (XS(3)*XSD(2)-XS(2)*XSD(3))/VO(9)
      CTM(6)= (CTM(1)*XSD(1)-XS(1)*CTM(4))/(CTM(1)*VO(8))
      DFM(1)= DFM(1)+C(N+7)*CTM(4)
      DFM(2)= DFM(2)+C(N+8)*CTM(5)
      DFM(3)= DFM(3)+C(N+9)*CTM(6)
      GO TO 18
  115 DFM(1) =X(1)
      DFM(2)= X(2)
      DFM(3)= X(3)
      GO TO 18
  120 DFM(1) =X(4)
      DFM(2)= X(5)
      DFM(3)= X(6)
      GO TO 18
C------------CALCULATE EQ.(264) -------------------------
  125 F1(1,1)= C(65)*C(69)-C(66)*C(68)
      F1(1,2)=-C(62)*C(69)+C(63)*C(68)
      F1(1,3)= C(62)*C(66)-C(63)*C(65)
      F1(2,1)=-C(64)*C(69)+C(66)*C(67)
      F1(2,2)= C(61)*C(69)-C(63)*C(67)
      F1(2,3)=-C(61)*C(66)+C(63)*C(64)
      F1(3,1)= C(64)*C(68)-C(65)*C(67)
      F1(3,2)=-C(61)*C(68)+C(62)*C(67)
      F1(3,3)= C(61)*C(65)-C(62)*C(64)
C------------CALCULATE EQ.(265) -------------------------
      VO(4)= C(61)*F1(1,1)+C(62)*F1(2,1)+C(63)*F1(3,1)
      DO 128 I=1,3
      DO 127 J=1,3
C------------CALCULATE EQ.(266) -------------------------
  127 CI(I,J)= F1(I,J)/VO(4)
  128 VO(I)= AP(I)- C(69+I)
C------------CALCULATE EQ.(258) -------------------------
      DO 130 I=1,3
      DFM(I) = 0.
      DO 130 J=1,3
  130 DFM(I) = DFM(I) + CI(I,J)*VO(J)
      GO TO 18
C 150 CALCULATE AMR
  150 SDP= SIN(C(151))
      CDP= COS(C(151))
      SDY= SIN(C(152))
      CDY= COS(C(152))
C------------CALCULATE EQ.(269) -------------------------
      XLB(1)= CDY*CDP
      XLB(2)= SDY
      XLB(3)= CDY*SDP
C------------CALCULATE EQ.(270) -------------------------
      DO 152 I=1,3
      XLG(I)=0.
      DO 152 J=1,3
  152 XLG(I)=XLG(I)+A(J,I)*XLB(J)
```

```
      VO(1)= REO**2
C------------CALCULATE EQ.(271A)--------------------------
      VO(2)= ATAN2(XLG(2),XLG(1))
C------------CALCULATE EQ.(271B)--------------------------
      SP(1)= SQRT(XLG(1)**2+XLG(2)**2)
      VO(3)=-ATAN2(XLG(3),SP(1))
      SLR= SIN(VO(2))
      CLR= COS(VO(2))
      SGR= SIN(VO(3))
      CGR= COS(VO(3))
C------------CALCULATE EQ.(277) -------------------------
      XA(1)= SLR*CGR
      XA(2)= R*CPH
      XA(3)= CPH*SGR-SPH*CLR*CGR
      XA(4)= R*SPH
      XA(5)= SPH*SGR+CPH*CLR*CGR
C------------CALCULATE EQ.(276) -------------------------
      XC(1)= XA(1)**2 + XA(3)**2 + RERP2*XA(5)**2
      XC(2)= 2.*(XA(2)*XA(3)+ RERP2*XA(4)*XA(5))
      XC(3)= XA(2)**2 + RERP2*XA(4)**2 - VO(1)
      SP(2)= XC(2)**2 - 4.*XC(1)*XC(3)
      IF(SP(2)) 154,156,156
  154 WRITE(OUT,5022)
 5022 FORMAT(49H IMAGINARY OR NEGATIVE SOLUTION ON AIRBORNE RADAR)
      DO 160 I=1,3
      DFM(I) = 0.
      DO 158 J=1,NSTX
  158 G(I,J) = 0.
      DO 160 J=1,NPV
      H(I,J) = 0.
  160 CONTINUE
      SI(1) = 1.E20
      RETURN
C------------CALCULATE EQ.(273) -------------------------
  156 DFM(1)= -.5*(XC(2)+SQRT(SP(2)))/XC(1)
      IF(DFM(1).LT.0.) GO TO 154
      DFM(2)= 0.
      DFM(3)= 0.
   18 IF(DCOMP.LT.0.) GO TO 80
      DO 20 I=1,3
      SP(I) = SYG(I,MTYP)
      IF(NPC(1).EQ.0) SP(I) = SP(I)+SIGM(I)
   20 KC(I)=0
      IF(KN) 22,22,24
   22 KOB=KAR
      GO TO 26
   24 KOB=JNBR
   26 DO 28 I=KAR,KOB
      J=LC(I)
      DATC(I)=DFM(J)
      SI(I)=SP(J)
      NCOUNT=NCOUNT+1
   28 KC(J) = NCOUNT
   30 IF(NPC(5)) 34,32,34
   32 CONTINUE
      WRITE(OUT,1002) (KC(I),I=1,3),MTYP,TO,(DFM(I),I=1,3),(SP(I),I=1,3)
 1002 FORMAT(1X,I4,1H,I4,1H,I4,2X,I2,1X,F10.4,F15.2,5F15.6)
   34 DO 36 I=1,NSTX
      DO 36 J=KAR,KOB
   36 G(J,I)=0.
```

```
      IF(MTYP-6) 200,250,270
C 200 CALCULATE G(1-10) FOR R,A,E
C-------------CALCULATE EQ.(244) -------------------------
  200 DO 202 I=1,3
      RCX(1,I)= (XSX(1,I)*XS(1)+XSX(2,I)*XS(2)+XSX(3,I)*XS(3))/CTM(1)
      RCX(2,I)= (XSX(2,I)*XS(3)-XSX(3,I)*XS(2))/VO(9)
  202 RCX(3,I)= (XSX(1,I)-RCX(1,I)*SELC)/VO(8)
      IF(MO(MTYP)) 204,218,204
  204 SP(1)= R*CPH
      SP(2)= DX(5)/R
      SP(3)= X(2)/(R*SP(1))
      SP(4)= X(2)*TPH/SP(1)
C-------------CALCULATE EQ.(247)--------------------------
      DO 206 I=1,3
      XSDX(I,1)= XSX(I,2)/R
      XSDX(I,2)= XSX(I,3)/SP(1)
      XSDX(I,3)=-XSX(I,1)
      XSDX(I,4)=-XSX(I,2)*SP(2)-XSX(I,3)*SP(3)
  206 XSDX(I,5)= XSX(I,3)*SP(4)
      VO(10)=-X(3)/R
      SP(4)= DX(6)/R
      SP(5)= DX(5)/R
      SP(6)=-TPH*DX(6)
      SP(7)=-R*DX(5)
      SP(1)= CTH*DX(6)/STH - TPH*DX(5)
      SP(3)= SP(1)
      SP(2)=-STH*DX(6)/CTH - TPH*DX(5)
      DO 208 I=1,3
      XSDX(I,4)= XSDX(I,4)+XSX(I,3)*SP(4)+XSX(I,2)*SP(5)
      XSDX(I,5)= XSDX(I,5)+XSX(I,2)*VO(10)+XSX(I,3)*SP(6)+XSX(I,1)*SP(7)
  208 XSDX(I,6)= XSX(I,3)*VO(10)+XSX(I,3)*SP(I)
      VO(11)=-(XSD(3)+2.*CTM(5)*XS(2))
      VO(12)= XSD(2)-2.*CTM(5)*XS(3)
      SP(1)= -CTM(6)*XS(2)/CELC
      SP(2)= -CTM(6)*XS(3)/CELC
      SP(3)= XSD(1)-CTM(6)*VO(8)
C-------------CALCULATE EQ.(245) -------------------------
      DO 216 I=1,6
      IF(I-3) 210,210,212
  210 RCDX(1,I)=0.
      RCDX(2,I)=0.
      RCDX(3,I)=0.
      GO TO 214
  212 RCDX(1,I)=XSX(1,I-3)*XSD(1)+XSX(2,I-3)*XSD(2)+XSX(3,I-3)*XSD(3)
     X             -RCX(1,I-3)*CTM(4)
      RCDX(2,I)=XSX(3,I-3)*VO(12)+XSX(2,I-3)*VO(11)
      RCDX(3,I)=-XSX(1,I-3)*CTM(4)+XSX(2,I-3)*SP(1)+XSX(3,I-3)*SP(2)
     X             +RCX(1,I-3)*SP(3)
  214 RCDX(1,I)=(RCDX(1,I)+XSDX(1,I)*XS(1)+XSDX(2,I)*XS(2)
     X             +XSDX(3,I)*XS(3))/CTM(1)
      RCDX(2,I)=(RCDX(2,I)+XSDX(2,I)*XS(3)-XSDX(3,I)*XS(2))/VO(9)
  216 RCDX(3,I)=((RCDX(3,I)-RCDX(1,I)*XS(1))/CTM(1)+XSDX(1,I))/VO(8)
  218 VO(13)= C(N+10)*CELC/SELC**2
      VO(14)= C(N+3)-C(N+11)/SELC**2
      DO 230 I=1,6
      IF(I-3) 220,220,222
  220 GG(1)=0.
      GG(2)=0.
      GG(3)=0.
      GO TO 224
```

```
C------------CALCULATE EQ.(243) -------------------------
  222 GG(1)= C(N+1)*RCX(1,I-3)-VO(13)*RCX(3,I-3)
      GG(2)= C(N+2)*RCX(2,I-3)
      GG(3)= VO(14)*RCX(3,I-3)
  224 IF(MO(MTYP)) 226,228,226
  226 GG(1)= GG(1)+ C(N+7)*RCDX(1,I)
      GG(2)= GG(2)+ C(N+8)*RCDX(2,I)
      GG(3)= GG(3)+ C(N+9)*RCDX(3,I)
  228 DO 230 K=KAR,KOB
      J=LC(K)
  230 G(K,I)=GG(J)
  232 CONTINUE
      GO TO 38
C------------CALCULATE EQ.(286)--------------------------
  250 XLGE(1,1)= 2.*( X( 7)*XLB(1)-X(10)*XLB(2)+X( 9)*XLB(3))
      XLGE(1,2)= 2.*( X( 8)*XLB(1)+X( 9)*XLB(2)+X(10)*XLB(3))
      XLGE(1,3)= 2.*(-X( 9)*XLB(1)+X( 8)*XLB(2)+X( 7)*XLB(3))
      XLGE(1,4)= 2.*(-X(10)*XLB(1)-X( 7)*XLB(2)+X( 8)*XLB(3))
      XLGE(2,1)= 2.*( X(10)*XLB(1)+X( 7)*XLB(2)-X( 8)*XLB(3))
      XLGE(2,2)= 2.*( X( 9)*XLB(1)-X( 8)*XLB(2)-X( 7)*XLB(3))
      XLGE(2,3)= 2.*( X( 8)*XLB(1)+X( 9)*XLB(2)+X(10)*XLB(3))
      XLGE(2,4)= 2.*( X( 7)*XLB(1)-X(10)*XLB(2)+X( 9)*XLB(3))
      XLGE(3,1)= 2.*(-X( 9)*XLB(1)+X( 8)*XLB(2)+X( 7)*XLB(3))
      XLGE(3,2)= 2.*( X(10)*XLB(1)+X( 7)*XLB(2)-X( 8)*XLB(3))
      XLGE(3,3)= 2.*(-X( 7)*XLB(1)+X(10)*XLB(2)-X( 9)*XLB(3))
      XLGE(3,4)= 2.*( X( 8)*XLB(1)+X( 9)*XLB(2)+X(10)*XLB(3))
      SP(4)= SGR/CGR
      SP(3)= XLG(1)**2 + XLG(2)**2
      SP(1)= SQRT(SP(3))
      DO 258 I=1,4
C------------CALCULATE EQ.(284) -------------------------
      XLGX(1,I) = (XLG(1)*XLGE(2,I)-XLG(2)*XLGE(1,I))/SP(3)
C------------CALCULATE EQ.(285) -------------------------
  258 XLGX(2,I) =-SP(1)*XLGE(3,I)-SP(4)*(XLG(1)*XLGE(1,I)+XLG(2)*XLGE(2,
     1I))
      VO(4)= -SLR*SGR
      VO(5)=  CLR*CGR
      VO(6)=  CPH*CGR+SPH*CLR*SGR
      VO(7)=  SPH*XA(1)
      VO(9)=  SPH*CGR-CPH*CLR*SGR
      VO(10)= CPH*XA(1)
C------------CALCULATE EQ.(283)--------------------------
      DO 260 I=1,5
      DO 260 J=1,7
  260 XAX(I,J)= 0.
      DO 262 I=1,4
      XAX(1,I+3) = VO(4)*XLGX(2,I) + VO(5)*XLGX(1,I)
      XAX(3,I+3) = VO(6)*XLGX(2,I) + VO(7)*XLGX(1,I)
  262 XAX(5,I+3) = VO(9)*XLGX(2,I) - VO(10)*XLGX(1,I)
      XAX(2,1) = CPH
      XAX(2,2) =-XA(4)
      XAX(3,2) =-XA(5)
      XAX(4,1) = SPH
      XAX(4,2) = XA(2)
      XAX(5,2) = XA(3)
C------------CALCULATE EQ.(282) -------------------------
      DO 264 I=1,7
      XCX(1,I)= 2.*(XA(1)*XAX(1,I)+XA(3)*XAX(3,I)+RERP2*XA(5)*XAX(5,I))
      XCX(2,I)= 2.*(XA(2)*XAX(3,I)+XA(3)*XAX(2,I)+RERP2*(XA(4)*XAX(5,I)
     1         + XA(5)*XAX(4,I)))
```

```
  264 XCX(3,I)= 2.*(XA(2)*XAX(2,I)+RERP2*XA(4)*XAX(4,I))                 OBSV3160
      G(1,1)= 0.                                                         OBSV3170
      G(1,2)= 0.                                                         OBSV3180
      G(1,3)= 0.                                                         OBSV3190
      VO(11)= 2.*XC(1)*DFM(1)+XC(2)                                      OBSV3200
      VO(12)= (DFM(1)+ XC(3)/VO(11))/XC(1)                               OBSV3210
      VO(13)= (.5-.5*XC(2)/VO(11))/XC(1)                                 OBSV3220
      VO(14)= 1./VO(11)                                                  OBSV3230
C-------------CALCULATE EQ.(281) ---------------------------             OBSV3240
      DO 266 I=1,7                                                       OBSV3250
  266 G(1,I+3)= -VO(12)*XCX(1,I)-VO(13)*XCX(2,I)-VO(14)*XCX(3,I)         OBSV3260
      GO TO 38                                                           OBSV3270
  270 IF(MTYP-8) 272,274,290                                             OBSV3280
  272 LT=0                                                               OBSV3290
      GO TO 276                                                          OBSV3300
  274 LT=3                                                               OBSV3310
  276 DO 284 I=KAR,KOB                                                   OBSV3320
      J=LC(I)                                                            OBSV3330
  284 G(I,LT+J) = 1.                                                     OBSV3340
      GO TO 38                                                           OBSV3350
  290 DO 296 I=1,10                                                      OBSV3360
      DO 292 J=1,3                                                       OBSV3370
      GG(J)=0.                                                           OBSV3380
      DO 292 K=1,3                                                       OBSV3390
  292 GG(J)= GG(J)+CI(J,K)*PAXP(K,I)                                     OBSV3400
      DO 294 K=KAR,KOB                                                   OBSV3410
      J=LC(K)                                                            OBSV3420
  294 G(K,I)=GG(J)                                                       OBSV3430
  296 CONTINUE                                                           OBSV3440
   38 CONTINUE                                                           OBSV3450
      IF(NSTC) 40,40,46                                                  OBSV3460
   40 IF(NPV) 99,99,42                                                   OBSV3470
   42 II=1                                                               OBSV3480
      IP=1                                                               OBSV3490
   44 JJ=MCC(II)                                                         OBSV3500
      GO TO 50                                                           OBSV3510
   46 II=1                                                               OBSV3520
      IP=0                                                               OBSV3530
   48 JJ=NC(II)                                                          OBSV3540
   50 GG(1)=0.                                                           OBSV3550
      GG(2)=0.                                                           OBSV3560
      GG(3)=0.                                                           OBSV3570
      IF(MTYP.GE.6) GO TO 375                                            OBSV3580
      IF(JJ-N) 60,60,54                                                  OBSV3590
   54 IF(JJ-N-15) 56,56,60                                               OBSV3600
   56 JN=JJ-N                                                            OBSV3610
      IF(JN-12) 58,58,350                                                OBSV3620
   58 GO TO(300,301,302,303,304,305,306,307,308,309,310,311),JN          OBSV3630
   60 II=II+1                                                            OBSV3640
      DO 66 I=KAR,KOB                                                    OBSV3650
      J=LC(I)                                                            OBSV3660
      IF(IP) 62,62,64                                                    OBSV3670
   62 G(I,II+ 9) = GG(J)                                                 OBSV3680
      GO TO 66                                                           OBSV3690
   64 H(I,II-1) = GG(J)                                                  OBSV3700
   66 CONTINUE                                                           OBSV3710
      IF(IP) 68,68,70                                                    OBSV3720
   68 IF(II-NSTC) 48,48,40                                               OBSV3730
   70 IF(II-NPV) 44,44,99                                                OBSV3740
C-------------CALCULATE EQ.(248) ---------------------------             OBSV3750
```

```
  300 GG(1) = CTM(1)
      GO TO 60
  301 GG(2) = CTM(2)
      GO TO 60
  302 GG(3) = CTM(3)
      GO TO 60
  303 GG(1) = 1.
      GO TO 60
  304 GG(2) = 1.
      GO TO 60
  305 GG(3) = 1.
      GO TO 60
  306 GG(1) = CTM(4)
      GO TO 60
  307 GG(2) = CTM(5)
      GO TO 60
  308 GG(3) = CTM(6)
      GO TO 60
  309 GG(1) = 1./SELC
      GO TO 60
  310 GG(3) = CELC/SELC
  311 GO TO 60
  350 SP(1)= SIN(C(N+13))
      SP(2)= COS(C(N+13))
      IF(JN-14) 352,354,356
  352 VO(15)=RERP2*(CPHDT(MTYP)/SP(2))**2
      VO(16)=-(ROT(MTYP)/REO)**2*(RERP2-1.)*SP(1)*SP(2)*ROT(MTYP)
C-------------CALCULATE EQ.(252) --------------------------
      XSC(1)= XS(3)*VO(15)-VO(6)-VO(16)*CCAPH(MTYP)
      XSC(2)= 0.
      XSC(3)=-XS(1)*VO(15)-VO(5)+ VO(16)*SCAPH(MTYP)
      GO TO 358
  354 XSC(1)=-XSX(1,3)
      XSC(2)=-XSX(2,3)
      XSC(3)=-XSX(3,3)
      GO TO 358
  356 XSC(1)=-CCAPH(MTYP)
      XSC(2)= 0.
      XSC(3)= SCAPH(MTYP)
C-------------CALCULATE EQ.(250) --------------------------
  358 RCC(1)=(XSC(1)*XS(1)+XSC(2)*XS(2)+XSC(3)*XS(3))/CTM(1)
      RCC(2)=(XSC(2)*XS(3)-XSC(3)*XS(2))/VO(9)
      RCC(3)=(XSC(1)-RCC(1)*SELC)/VO(8)
C-------------CALCULATE EQ.(249) --------------------------
      GG(1)= C(N+1)*RCC(1)-VO(13)*RCC(3)
      GG(2)= C(N+2)*RCC(2)
      GG(3)= VO(14)*RCC(3)
      IF(MO(MTYP)) 360,60,360
  360 IF(JN-14) 362,364,366
  362 SP(3)= VO(15)-1.
      SP(5)= XS(2)*DX(6)
C-------------CALCULATE EQ.(253A)--------------------------
      XSDC(1)= VO(10)*(XSC(1)+VO(16)*CCAPH(MTYP)-VO(6)*SP(3))
     X   +VO(15)*(XSX(3,2)*DX(5)+SP(5)*SPHDT(MTYP))
      XSDC(2)= 0.
      XSDC(3)= VO(10)*(XSC(3)-VO(16)*SCAPH(MTYP)-VO(5)*SP(3))
     X   +VO(15)*(-XSX(1,2)*DX(5)+SP(5)*CPHDT(MTYP))
      GO TO 368
  364 SP(1)= XSX(2,3)*DX(6)+XSX(2,2)*DX(5)
C-------------CALCULATE EQ.(253B)--------------------------
```

```
      XSDC(1)= VO(10)*XSC(1)+   SP(1)*CPHDT(MTYP)                          OBSV4360
      XSDC(2)= VO(10)*XSC(2)+    XS(2)*DX(6)+R*CTH*SPH*DX(5)               OBSV4370
      XSDC(3)= VO(10)*XSC(3)-   SP(1)*SPHDT(MTYP)                          OBSV4380
      GO TO 368                                                            OBSV4390
  366 RCDC(1)= 0.                                                          OBSV4400
      RCDC(2)= 0.                                                          OBSV4410
      RCDC(3)= 0.                                                          OBSV4420
      GO TO 370                                                            OBSV4430
C-------------CALCULATE EQ.(251)--------------------------                 OBSV4440
  368 RCDC(1)= XSDC(1)*XS(1)+XSDC(2)*XS(2)+XSDC(3)*XS(3)                   OBSV4450
      RCDC(2)= XSDC(2)*XS(3)-XSDC(3)*XS(2)                                 OBSV4460
      RCDC(3)= XSDC(1)*CTM(1)                                              OBSV4470
  370 SP(1)= CTM(6)*XS(2)/CELC                                             OBSV4480
      SP(2)= CTM(6)*XS(3)/CELC                                             OBSV4490
      SP(3)= XSD(1)-CTM(6)*VO(8)                                           OBSV4500
      SP(4)= CTM(1)*VO(8)                                                  OBSV4510
      RCDC(1)=(RCDC(1)+XSC(1)*XSD(1)+XSC(2)*XSD(2)+XSC(3)*XSD(3)           OBSV4520
     X          -RCC(1)*CTM(4))/CTM(1)                                     OBSV4530
      RCDC(2)=(RCDC(2)+XSC(2)*VO(11)+XSC(3)*VO(12))/VO(9)                  OBSV4540
      RCDC(3)=(RCDC(3)-RCDC(1)*XS(1)-XSC(1)*CTM(4)-XSC(2)*SP(1)            OBSV4550
     X          -XSC(3)*SP(2)+RCC(1)*SP(3))/SP(4)                          OBSV4560
      GG(1)= GG(1) + C(N+7)*RCDC(1)                                        OBSV4570
      GG(2)= GG(2) + C(N+8)*RCDC(2)                                        OBSV4580
      GG(3)= GG(3) + C(N+9)*RCDC(3)                                        OBSV4590
      GO TO 60                                                             OBSV4600
  375 IF(MTYP.EQ.9) GO TO 400                                              OBSV4610
      DO 390 I=1,3                                                         OBSV4620
      DO 385 J=1,NSTC                                                      OBSV4630
  385 G(I,J+10) = 0.                                                       OBSV4640
      DO 390 J=1,NPV                                                       OBSV4650
  390 H(I,J) = 0.                                                          OBSV4660
      RETURN                                                               OBSV4670
  400 IF(JJ.LE.35) GO TO 405                                               OBSV4680
      IF(JJ.GT.72) GO TO 60                                                OBSV4690
      IF(JJ.GE.70) GO TO 435                                               OBSV4700
      IF(JJ.GT.60) GO TO 440                                               OBSV4710
  405 NNN = JJ                                                             OBSV4720
      CALL PAXPC                                                           OBSV4730
      IF(JJ.GE.45) GO TO 415                                               OBSV4740
  407 DO 410 I=1,3                                                         OBSV4750
  410 GG(I) = CI(I,1)*VO(1) + CI(I,2)*VO(2) + CI(I,3)*VO(3)                OBSV4760
C     OBSV4770 THRU OBSV4880 HAVE BEEN DELETED                             OBSV4770
      GO TO 60                                                             OBSV4890
  415 DO 420 I=1,3                                                         OBSV4900
  420 VO(I) = DTRAN(I,1)*SP(4)+DTRAN(I,2)*SP(5)+DTRAN(I,3)*SP(6)           OBSV4910
      GO TO 407                                                            OBSV4920
  435 GG(1) =-CI(1,JJ-69)                                                  OBSV4930
      GG(2) =-CI(2,JJ-69)                                                  OBSV4940
      GG(3) =-CI(3,JJ-69)                                                  OBSV4950
      GO TO 60                                                             OBSV4960
  440 I = (JJ-58)/3                                                        OBSV4970
      J = JJ - 3*I - 57                                                    OBSV4980
      SP(4) = CI(J,1)*(C(70)-AP(1)) + CI(J,2)*(C(71)-AP(2))                OBSV4990
     1    + CI(J,3)*(C(72)-AP(3))                                          OBSV5000
      GG(1) = CI(1,I)*SP(4)                                                OBSV5010
      GG(2) = CI(2,I)*SP(4)                                                OBSV5020
      GG(3) = CI(3,I)*SP(4)                                                OBSV5030
      GO TO 60                                                             OBSV5040
   80 DO 82 I=1,3                                                          OBSV5050
      SP(I)=SYG(I,MTYP)                                                    OBSV5060
```

```
      IF(NPC(1).NE.2)  SP(I)=SP(I)+SIG(I,KG)                          OBSV5070
   82 CONTINUE                                                        OBSV5080
   99 CONTINUE                                                        OBSV5090
      RETURN                                                          OBSV5100
      END                                                             OBSV5110
```

```
      SUBROUTINE OUTPUT
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),        COMM0010
     *     NALL,LRK                                                        COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP       ,CBET       ,CETA     , COMM0030
     1     CF(3),JCS ,CXZI       ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  , COMM0040
     2     DRDP,DWDH(2),ETA,HO,N4(3)      ,N5(3)      ,N6(3)      ,PAXP(3,30), COMM0050
     3     PE          ,ROE        ,ROEC       ,ROEM       ,SALP       ,SBET     , COMM0060
     4     SETA        ,SREF       ,SXZI       ,TAU        ,TMAS(20)   ,TONE     , COMM0070
     5     TWO         ,TUW(20)    ,TVW(20)    ,UW         ,VA(5)      ,VB(5)    , COMM0080
     6     VW                      ,XIND1      ,XIND2      ,XLREF      ,XM       , COMM0090
     7     XMAS(20)    ,XNU        ,XQ         ,XQSM       ,XRE        ,XTEMP    , COMM0100
     8     XUW(20)     ,XVW(20)    ,XZI        ,YMAS       ,YMASM                  COMM0110
      COMMON /TWICE/ A(3,3)      ,CPH        ,CT(5)      ,SP(10)     ,ST(5)      COMM0120
     1     ,TPH        ,VO(20)     ,ZO(20)     ,AG(3)      ,AMDOT(3)  ,APDOT(4)    COMM0130
     2     ,B(8)       ,CGM(3)     ,DFM(3)     ,DTRAN(3,3),DUD(10,1),DUE(4)      COMM0140
     3     ,DUF(4)     ,F1(3,40)  ,F2(3,5)    ,F3(4,40)   ,GG(3)      ,HI(2)       COMM0150
     4     ,IDN ,II    ,IP   ,IPC ,JJ   ,JN   ,KA ,KD(3),KDUM ,KG2                 COMM0160
     5     ,K1   ,NNN ,NPTS        ,PAR(6)     ,PDOT(3)    ,PMDOT(3)  ,RES(6)      COMM0170
     6     ,RO         ,SPD(5)     ,SPH        ,TRAN(3,3),XX(50)                   COMM0180
      DIMENSION DUB(30,15)                                                       COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                               COMM0200
      COMMON /METH2/ AB(3)       ,AM(3)      ,AP(4)      ,AX(40)     ,AY(40)     COMM0210
     1     ,AZ(40)     ,C(160)     ,CCAPH(5)  ,CONRD      ,CPHDT(5)  ,CUZ(30,5)   COMM0220
     2     ,CVZ(30,5),D(5)         ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)      COMM0230
     3     ,DET        ,DFIT(9)    ,DTF(9)     ,DTI(9)     ,DZ(30)     ,ET         COMM0240
     4     ,FLOS       ,G(3,30)   ,H(3,5)     ,I ,ICOUNT,J     ,JNBR,JNBRS,JST   COMM0250
     5     ,K    ,KAR ,KC(3)       ,KDAP       ,KDATA      ,KDATAS     ,KG,KI,KK   COMM0260
     6     ,KN   ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)     ,LS     ,LT       COMM0270
     7     ,L1   ,M    ,MC(5)      ,MCC(5)     ,MO(5)      ,MR(3,9)   ,MT         COMM0280
     8     ,MTP(40)    ,MTYP       ,MTYPS      ,N ,NC(30),NCOUNT     ,NPC(15)     COMM0290
     9     ,NPU ,NPV   ,NS(9)      ,NSS(5)     ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)    COMM0300
     A     ,N8(3)      ,N9(3)      ,N10(3)     ,N11(3)     ,OMEGA      ,P(30,30)   COMM0310
     B     ,PA(3)                  ,PM(3)      ,R          ,REO        ,RERP2      COMM0320
     C     ,RPO        ,ROT(5)     ,S(5)       ,SCAPH(5)  ,SI(3)      ,SIG(3,40)  COMM0330
     D     ,SIGM(3)    ,SIGMS(3)  ,SPHDT(5)  ,SUM ,SUM2,SYG(3,9) ,TFINAL         COMM0340
     E     ,TFIT(9)    ,TIME       ,TIMES      ,TO         ,TP(40)     ,TQ(40)     COMM0350
     F     ,TR(40)     ,TT(40)     ,TXCG(20)  ,TYCG(20)  ,TYM(40)    ,TZCG(20)   COMM0360
     G     ,TZERO      ,XJ2        ,XMU        ,XMUJ       ,XP(3)      ,XXCG(20)   COMM0370
     H     ,XYCG(20)  ,XZCG(20)                                                  COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                           COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                                COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK              COMM0410
     A   ,PB(23),RL(23),TMB(23)                                                  COMM0420
      COMMON /EXTRA / GO                                                         COMM0430
    2 CALL AUXIL                                                                 OUTP0010
    4 ZO(1) = VA(2)                                                              OUTP0020
      ZO(2) = SQRT(VA(1)**2+VA(2)**2)                                            OUTP0030
      VO(1) = VA(4)                                                              OUTP0040
      IF(VO(1).EQ.0.) VO(1)=.00001                                               OUTP0050
      ST(1)=-X(3)/VO(1)                                                          OUTP0060
      CT(1)= ZO(2)/VO(1)                                                         OUTP0070
      VO(2)=ATAN2(ST(1),CT(1))/CONRD                                             OUTP0080
      IF(ZO(2).EQ.0.) ZO(2)=.00001                                               OUTP0090
      ST(2)= ZO(1)/ZO(2)                                                         OUTP0100
      CT(2) = VA(1)/ZO(2)                                                        OUTP0110
      VO(3)=ATAN2(ST(2),CT(2))/CONRD                                             OUTP0120
      RO=REO/SQRT(1.-(1.-RERP2)*SPH**2)                                          OUTP0130
      VO(4)= X(4)+REO-RO                                                         OUTP0140
      VO(5)=ATAN2(RERP2*SPH,CPH)/CONRD                                           OUTP0150
```

```
      VO(6)= X(6)/CONRD
      SP(1)=SQRT(X(1)**2+X(2)**2)
      VO(7)= SQRT(SP(1)**2+X(3)**2)
      IF(SP(1).EQ.0.) SP(1)=.00001
      VO(8)=ATAN2(-X(3),SP(1))/CONRD
      VO(9)=ATAN2(X(2),X(1))/CONRD
      VO(10)=ATAN2(A(1,2),A(1,1))/CONRD
      SP(1) =SQRT(1.-A(1,3)**2)
      VO(11)=-ATAN2(A(1,3),SP(1))/CONRD
      VO(12)=ATAN2(A(2,3),A(3,3))/CONRD
      VO(13) = VB(1)
      VO(14) = VB(2)
      VO(15) = VB(3)
      ZO(3) = VB(4)
      ST(4) = VB(2)/VA(4)
      CT(4) = VB(4)/VA(4)
      VO(17)=ATAN2(ST(4),CT(4))/CONRD
      ST(5) = VB(3)/VB(4)
      CT(5) = VB(1)/VB(4)
      VO(18)=ATAN2(ST(5),CT(5))/CONRD
      ZO(4)=A(2,3)+ST(4)*ST(1)
      ZO(5)=(A(2,2)*CT(2)-A(2,1)*ST(2))*CT(1)
      ZO(9)=SQRT(ZO(4)**2+ZO(5)**2)
      IF(ZO(9).EQ.0.) ZO(9)=.000001
      ST(3)=ZO(4)/ZO(9)
      CT(3)=ZO(5)/ZO(9)
      VO(16)=ATAN2(ST(3),CT(3))/CONRD
      SP(1)=SQRT(VO(14)**2+VO(15)**2)
      SP(3)=ATAN2(SP(1),VO(1))/CONRD
      SP(2)=ATAN2(VO(14),VO(15))/CONRD
      VO(19)=X(5)/CONRD
      WRITE(OUT,1001)
 1001 FORMAT(//)
      WRITE(OUT,1000) TO
      WRITE(OUT,1002) (VO(I),I=1,6)
      WRITE(OUT,1004) X(1),X(2),X(3),VO(10),VO(11),VO(12)
      WRITE(OUT,1006) (VO(I),I=13,18)
      WRITE(OUT,1008) (XP(I),I=1,3),(SP(I),I=1,3)
      WRITE(OUT,1010) (PA(I),I=1,3),(AB(I),I=1,3)
      WRITE(OUT,1012) (PM(I),I=1,3),(CF(I),I=1,3)
      WRITE(OUT,1011) XQ,XM,ROE,(CA(I),I=1,3)
      WRITE(OUT,1014) (X(I),I=7,10),VO(19),XRE
      WRITE(OUT,1015) UW,VW
 1000 FORMAT(5H TIME,E15.8)
 1002 FORMAT(5H V(R),E15.8,5H G(R)E15.8,5H L(R)E15.8,5H ALT E15.8,5H LAT
     1 E15.8,5H LON E15.8)
 1004 FORMAT(5H U(1),E15.8,5H V(I)E15.8,5H W(I)E15.8,5H PSI E15.8,5H THE
     2 E15.8,5H PHI E15.8)
 1006 FORMAT(5H U(B),E15.8,5H V(B)E15.8,5H W(B)E15.8,5H SIG E15.8,5H BET
     3 E15.8,5H ALF E15.8)
 1008 FORMAT(5H XP  ,E15.8,5H YP  E15.8,5H ZP  E15.8,5H V(T)E15.8,5H XZI
     4 E15.8,5H ETA E15.8)
 1010 FORMAT(5H P   ,E15.8,5H Q   E15.8,5H R   E15.8,5H AX  E15.8,5H AY
     5 E15.8,5H AZ  E15.8)
 1011 FORMAT(5H DP  ,E15.8,5H  M  ,E15.8,5H RHO ,E15.8,5H C1  ,E15.8,5H
     1C2  ,E15.8,5H C3  ,E15.8)
 1012 FORMAT(5H PM  ,E15.8,5H QM  E15.8,5H RM  E15.8,5H CX  E15.8,5H CY
     6 E15.8,5H CZ  E15.8)
 1014 FORMAT(5H E0  ,E15.8,5H E1  E15.8,5H E2  E15.8,5H E3  E15.8,5H LAT
     7CE15.8,5H RE  E15.8)
```

```
1015 FORMAT(5H UW  ,E15.8,5H VW  E15.8)
     IF(NPC(1)-1) 20,99,28
  20 IF(DCOMP) 28,22,22
  22 IF(NSTC) 28,28,24
  24 DO 26 I=1,NSTC
     K=NC(I)
  26 AG(I) = C(K)
     WRITE(OUT,1016) (NC(I),AG(I),I=1,NSTC)
1016 FORMAT(6(2H C,I3,E15.8))
  28 IF(NPC(3)-1) 30,100,100
 100 DO 102 I=1,6
     DO 102 K=1,10
 102 DUB(K,I)=0.
     ZO(6) = VA(5)
     ZO(7)=ZO(6)*ZO(2)
     ZO(8) =-DWDH(2)+OMEGA*CPH
     DUB(1,1) = VA(1)/VA(4)
     DUB(2,1) = VA(2)/VA(4)
     DUB(3,1)= X(3)/VO(1)
     DUB(4,1) = DUB(1,1)*DWDH(1) - DUB(2,1)*ZO(8)
     DUB(5,1) =-(DUB(1,1)*DWDH(1)+DUB(2,1)*DWDH(2))*DRDP+DUB(2,1)*R*
    1   OMEGA*SPH
     SP(1)= X(3)/(VO(1)*ZO(2))
     DUB(1,2)= SP(1)*DUB(1,1)
     DUB(2,2)= SP(1)*DUB(2,1)
     DUB(3,2)=-ZO(2)/ZO(6)
     DUB(4,2)= SP(1)*DUB(4,1)
     DUB(5,2)= SP(1)*DUB(5,1)
     SP(1)=ZO(2)**2
     SP(2) = VA(2)*DWDH(1) - VA(1)*DWDH(2)
     DUB(1,3) = -VA(2)/SP(1)
     DUB(2,3) = VA(1)/SP(1)
     DUB(4,3) = -(SP(2)+VA(1)*OMEGA*CPH)/SP(1)
     DUB(5,3) = (SP(2)*DRDH+VA(1)*R*OMEGA*SPH)/SP(1)
     IF(NPC(3)-1) 120,110,120
 110 SP(1)= 2./(A(1,1)**2+A(1,2)**2)
     DUB(7 ,4)=SP(1)*(X(10)*A(1,1)-X( 7)*A(1,2))
     DUB(8 ,4)=SP(1)*(X( 9)*A(1,1)-X( 8)*A(1,2))
     DUB(9 ,4)=SP(1)*(X( 8)*A(1,1)+X( 9)*A(1,2))
     DUB(10,4)=SP(1)*(X( 7)*A(1,1)+X(10)*A(1,2))
     SP(1)= 2./SQRT(1.-A(1,3)**2)
     DUB(7 ,5)= X( 9)*SP(1)
     DUB(8 ,5)=-X(10)*SP(1)
     DUB(9 ,5)= X( 7)*SP(1)
     DUB(10,5)=-X( 8)*SP(1)
     SP(1)= 2./(A(3,3)**2+A(2,3)**2)
     DUB(7 ,6)=SP(1)*(X( 8)*A(3,3)-X( 7)*A(2,3))
     DUB(8 ,6)=SP(1)*(X( 7)*A(3,3)+X( 8)*A(2,3))
     DUB(9 ,6)=SP(1)*(X(10)*A(3,3)+X( 9)*A(2,3))
     DUB(10,6)=SP(1)*(X( 9)*A(3,3)-X(10)*A(2,3))
     GO TO 200
 120 SP(1)= X(1)*X( 7) + ZO(1)*X(10) - X(3)*X( 9)
     SP(2)= X(1)*X( 8) + ZO(1)*X( 9) + X(3)*X(10)
     SP(3)= X(1)*X( 9) - ZO(1)*X( 8) + X(3)*X( 7)
     SP(4)= X(1)*X(10) - ZO(1)*X( 7) - X(3)*X( 8)
     SP(5)= ST(4)/(CT(4)*VO(1))
     SP(6)= 2./VO(1)
     DUB(1 ,5)= (CT(4)*A(2,1)-SP(5)*(X(1) -A(2,1)*VO(14)))/VO(1)
     DUB(2 ,5)= (CT(4)*A(2,2)-SP(5)*(ZO(1)-A(2,2)*VO(14)))/VO(1)
     DUB(3 ,5)= (CT(4)*A(2,3)-SP(5)*(X(3) -A(2,3)*VO(14)))/VO(1)
```

```
      DUB(4 ,5)= -ZO(8)*DUB(2,5)
      DUB(5 ,5)= -DUB(4,5)*R*TPH
      DUB(7 ,5)=(-CT(4)*SP(4)-SP(5)*(VO(13)*SP(1)+VO(15)*SP(3)))*SP(6)
      DUB(8 ,5)= (CT(4)*SP(3)-SP(5)*(VO(13)*SP(2)+VO(15)*SP(4)))*SP(6)
      DUB(9 ,5)= (CT(4)*SP(2)+SP(5)*(VO(13)*SP(3)-VO(15)*SP(1)))*SP(6)
      DUB(10,5)=(-CT(4)*SP(1)+SP(5)*(VO(13)*SP(4)-VO(15)*SP(2)))*SP(6)
      SP(6)= ZO(3)**2
      DUB(1 ,6) = (VO(13)*A(3,1)-VO(15)*A(1,1))/SP(6)
      DUB(2 ,6) = (VO(13)*A(3,2)-VO(15)*A(1,2))/SP(6)
      DUB(3 ,6) = (VO(13)*A(3,3)-VO(15)*A(1,3))/SP(6)
      DUB(4 ,6) = -ZO(8)*DUB(2,6)
      DUB(5 ,6) = -DUB(4,6)*R*TPH
      SP(6)= 2./SP(6)
      DUB(7 ,6) = (VO(13)*SP(3)-VO(15)*SP(1))*SP(6)
      DUB(8 ,6) = (VO(13)*SP(4)-VO(15)*SP(2))*SP(6)
      DUB(9 ,6) = (VO(13)*SP(1)+VO(15)*SP(3))*SP(6)
      DUB(10,6) = (VO(13)*SP(2)+VO(15)*SP(4))*SP(6)
      SP(1)= CT(3)*ST(4)*CT(1)+ST(3)*ST(1)*ZO(5)/CT(1)
      SP(2)= ST(3)*CT(1)*(A(2,2)*ST(2)+A(2,1)*CT(2))
      SP(3)= CT(3)*CT(4)*ST(1)
      DUB(1,4) = (SP(1)*DUB(1,2)+SP(2)*DUB(1,3)+SP(3)*DUB(1,5))/ZO(9)
      DUB(2,4) = (SP(1)*DUB(2,2)+SP(2)*DUB(2,3)+SP(3)*DUB(2,5))/ZO(9)
      DUB(3,4) = (SP(1)*DUB(3,2)+SP(2)*DUB(3,3)+SP(3)*DUB(3,5))/ZO(9)
      DUB(4,4) = (SP(1)*DUB(4,2)+SP(2)*DUB(4,3)+SP(3)*DUB(4,5))/ZO(9)
      DUB(5,4) = (SP(1)*DUB(5,2)+SP(2)*DUB(5,3)+SP(3)*DUB(5,5))/ZO(9)
      SP(6)= ST(3)*CT(1)
      SP(5)= SP(6)*CT(2)
      SP(6)= SP(6)*ST(2)
      DUB(7,4) =(SP(3)*DUB(7 ,5)+2.*(CT(3)*X( 8)-SP(5)*X( 7)-SP(6)*X(10)
     X))/ZO(9)
      DUB(8,4) =(SP(3)*DUB(8 ,5)+2.*(CT(3)*X( 7)+SP(5)*X( 8)+SP(6)*X( 9)
     X))/ZO(9)
      DUB(9,4) =(SP(3)*DUB(9 ,5)+2.*(CT(3)*X(10)-SP(5)*X( 9)+SP(6)*X( 8)
     X))/ZO(9)
      DUB(10,4)=(SP(3)*DUB(10,5)+2.*(CT(3)*X( 9)+SP(5)*X(10)-SP(6)*X( 7)
     X))/ZO(9)
  200 DO 205 I=1,3
      DO 205 J=1,NSTX
      DUB(J,I+6)=0.
      DUB(J,I+9)=P(I+3,J)
      DUB(J,I+12)=0.
      DO 202 K=1,5
      DUM=P(K,J)
      IF(K-J) 202,202,201
  201 DUM=P(J,K)
  202 DUB(J,I+6)=DUB(J,I+6)+DUB(K,I)*DUM
      DO 204 K=1,10
      DUM=P(K,J)
      IF(K-J) 204,204,203
  203 DUM=P(J,K)
  204 DUB(J,I+12)=DUB(J,I+12)+DUB(K,I+3)*DUM
  205 CONTINUE
      DO 218 I=1,3
      DO 208 J=I,3
      P(J+1,I)=0.
      P(J+7,I)=0.
      DO 206 K=1,5
  206 P(J+1,I)=P(J+1,I)+DUB(K,I+6)*DUB(K,J)
      DO 207 K=1,10
  207 P(J+7,I)=P(J+7,I)+DUB(K,I+6)*DUB(K,J+3)
```

```
  208 CONTINUE
      DO 212 J=I,3
      P(J+7,I+6)=0.
      P(J+4,I+3)=P(I+3,J+3)
      DO 212 K=1,10
  212 P(J+7,I+6)=P(J+7,I+6)+DUB(K,I+12)*DUB(K,J+3)
      DO 214 J=1,3
      P(J+4,I)= DUB(J+3,I+6)
  214 P(J+7,I+3)= DUB(I+3,J+12)
      DO 216 J=1,NSTC
      P(J+10,I)= DUB(J+10,I+7)
      P(J+10,I+6)= DUB(J+10,I+12)
  216 P(J+10,I+3)= P(I+3,J+10)
  218 CONTINUE
      DO 210 J=I,NSTC
  210 P(J+10,I+9)= P(I+10,J+10)
      LT=1
      MT=NSTX-1
      NT=9
      GO TO 34
   30 LT=0
      MT=NSTX
      NT=10
      DO 32 I=1,NSTX
      K=I+1
      DO 32 J=K,NSTX
   32 P(J,I)=P(I,J)
   34 IF(NPC(1)) 35,130,35
  130 IF(NPC(4)-1) 35,132,35
  132 IF(NPC(3)) 134,134,138
  134 DO 136 I=1,10
  136 VO(I)= DZ(I)
      GO TO 146
  138 DO 144 I=1,3
      VO(I)=0.
      VO(I+6)=0.
      DO 140 J=1,5
  140 VO(I)=VO(I)+ DUB(J,I)*DZ(J)
      VO(I+3)= DZ(I+3)
      DO 142 J=1,10
  142 VO(I+6)=VO(I+6)+ DUB(J,I+3)*DZ(J)
  144 CONTINUE
  146 IF(NSTC) 148,147,148
  147 WRITE(OUT,1028) (I,VO(I),I=1,NT)
      GO TO 35
  148 DO 150 I=1,NSTC
  150 VO(I+NT)= DZ(I+10)
      WRITE(OUT,1028) (I,VO(I),I=1,NT),(NC(I),VO(I+NT),I=1,NSTC)
 1028 FORMAT(6(3H Z(I3,1H)E13.6))
   35 DO 40 I=1,MT
      K=I+LT
      IF(P(K,I)) 36,38,38
   36 WRITE(OUT,1030) I,I,I,P(   K,I)
 1030 FORMAT(26H ****NEGATIVE VARIANCE ON I3,31HRD TRANSFORMED VARIABLE,
     A SIGMA(I3,1H,I3,4H) = E15.8)
      GO TO 40
   38 VO(I)=SQRT(P(   K,I))
   40 CONTINUE
      IF(NSTC.GT.0) GO TO 50
      WRITE(OUT,1032) (I,VO(I),I=1,NT)
```

```
      GO TO 52
   50 DO 39 I=1,NSTC
      K=I+NT
   39 ZO(I)=VO(K)
      WRITE(OUT,1032) (I,VO(I),I=1,NT),(NC(I),ZO(I)    ,I=1,NSTC)
 1032 FORMAT(6(3H S(I3,1H)E13.6))
   52 IF(NPC(7).NE.ICOUNT) RETURN
      IF(NPC(6)-1) 99,41,42
   41 IF(DCOMP.GT.0)        RETURN
   42 WRITE(OUT,1033)
 1033 FORMAT(41H COVARIANCE MATRIX P (UPPER TRIANGLE ONLY)
      DO 43 I=1,MT
   43 WRITE(OUT,1034) (P(I+LT,J),J=1,I)
 1034 FORMAT(X,12E10.3)
      IF(NPU) 44,44,250
   44 IF(NPV) 46,46,300
   46 CONTINUE
   99 RETURN
  250 IF(LT) 256,252,256
  252 DO 254 I=1,NSTX
      DO 254 J=1,NPU
  254 DUB(I,J+6)=CUZ(I,J)
      GO TO 266
  256 DO 262 I=1,3
      DO 262 J=1,NPU
      DUB(I,J+6)=0.
      DUB(I+6,J+6)=0.
      DO 258 K=1,5
  258 DUB(I,J+6)=DUB(I,J+6)+DUB(K,I)*CUZ(K,J)
      DO 260 K=1,10
  260 DUB(I+6,J+6)=DUB(I+6,J+6)+DUB(K,I+3)*CUZ(K,J)
  262 DUB(I+3,J+6)=CUZ(I+3,J)
      DO 264 I=1,NSTC
      DO 264 J=1,NPU
  264 DUB(I+9,J+6)=CUZ(I+10,J)
  266 WRITE(OUT,1035)
 1035 FORMAT(34H CORRELATION MATRIX (CUZ)TRANSPOSE)
      MT=NSTX-LT
      DO 268 J=1,NPU
  268 WRITE(OUT,1034) (DUB(I,J+6),I=1,MT)
      GO TO 44
  300 IF(LT) 306,302,306
  302 DO 304 I=1,NSTX
      DO 304 J=1,NPV
  304 DUB(I,J+6)=CVZ(I,J)
      GO TO 316
  306 DO 312 I=1,3
      DO 312 J=1,NPV
      DUB(I,J+6)=0.
      DUB(I+6,J+6)=0.
      DO 308 K=1,5
  308 DUB(I,J+6)=DUB(I,J+6)+DUB(K,I)*CVZ(K,J)
      DO 310 K=1,10
  310 DUB(I+6,J+6)=DUB(I+6,J+6)+DUB(K,I+3)*CVZ(K,J)
  312 DUB(I+3,J+6)=CVZ(I+3,J)
      DO 314 I=1,NSTC
      DO 314 J=1,NPV
  314 DUB(I+9,J+6)=CVZ(I+10,J)
  316 WRITE(OUT,1036)
 1036 FORMAT(34H CORRELATION MATRIX (CVZ)TRANSPOSE)
```

```
      MT=NSTX-LT                                                        OUTP3160
      DO 318 J=1,NPV                                                    OUTP3170
  318 WRITE(OUT,1034) (DUB(I,J+6),I=1,MT)                               OUTP3180
      GO TO 46                                                          OUTP3190
      END                                                               OUTP3200
```

```
      SUBROUTINE PRESET
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),     COMM0010
     *    NALL,LRK                                                      COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP     ,CBET     ,CETA     ,  COMM0030
     1    CF(3),JCS ,CXZI     ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH ,  COMM0040
     2    DRDP,DWDH(2),ETA,HO,N4(3)     ,N5(3)     ,N6(3)    ,PAXP(3,30), COMM0050
     3    PE        ,ROE      ,ROEC     ,ROEM      ,SALP     ,SBET     ,  COMM0060
     4    SETA      ,SREF     ,SXZI     ,TAU       ,TMAS(20) ,TONE     ,  COMM0070
     5    TTWO      ,TUW(20)  ,TVW(20)  ,UW        ,VA(5)    ,VB(5)    ,  COMM0080
     6    VW                  ,XIND1    ,XIND2     ,XLREF    ,XM       ,  COMM0090
     7    XMAS(20)  ,XNU      ,XQ       ,XQSM      ,XRE      ,XTEMP    ,  COMM0100
     8    XUW(20)   ,XVW(20)  ,XZI      ,YMAS      ,YMASM                 COMM0110
      COMMON /TWICE/ A(3,3)   ,CPH      ,CT(5)     ,SP(10)   ,ST(5)       COMM0120
     1    ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)     ,AMDOT(3) ,APDOT(4)    COMM0130
     2    ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)      COMM0140
     3    ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40)  ,GG(3)    ,HI(2)       COMM0150
     4    ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2              COMM0160
     5    ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)   ,PMDOT(3) ,RES(6)      COMM0170
     6    ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)                 COMM0180
      DIMENSION DUB(30,15)                                              COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                      COMM0200
      COMMON /METH2/ AB(3)    ,AM(3)    ,AP(4)     ,AX(40)   ,AY(40)      COMM0210
     1    ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD     ,CPHDT(5) ,CUZ(30,5)   COMM0220
     2    ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)   ,DATAS(3) ,DATC(3)     COMM0230
     3    ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)    ,DZ(30)   ,ET          COMM0240
     4    ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J    ,JNBR,JNBRS,JST   COMM0250
     5    ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA     ,KDATAS   ,KG,KI,KK    COMM0260
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)     ,LCS(3)   ,LS   ,LT    COMM0270
     7    ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)     ,MR(3,9)  ,MT          COMM0280
     8    ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT    ,NPC(15)     COMM0290
     9    ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)    COMM0300
     A    ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)    ,OMEGA    ,P(30,30)    COMM0310
     B    ,PA(3)              ,PM(3)    ,R         ,REO      ,RERP2       COMM0320
     C    ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5)  ,SI(3)    ,SIG(3,40)   COMM0330
     D    ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL       COMM0340
     E    ,TFIT(9)  ,TIME     ,TIMES    ,TO        ,TP(40)   ,TQ(40)      COMM0350
     F    ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20)  ,TYM(40)  ,TZCG(20)    COMM0360
     G    ,TZERO    ,XJ2      ,XMU      ,XMUJ      ,XP(3)    ,XXCG(20)    COMM0370
     H    ,XYCG(20) ,XZCG(20)                                           COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                  COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                       COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK     COMM0410
     A ,PB(23),RL(23),TMB(23)                                           COMM0420
      COMMON /EXTRA / GO                                                COMM0430
      NDIMT = 160                                                       PRES0010
      NDIMZ = 30                                                        PRES0020
      NDIMV = 5                                                         PRES0030
      NDIMU = 5                                                         PRES0040
      DO 12 I=1,NDIMZ                                                   PRES0050
      DO 4  J=1,NDIMZ                                                   PRES0060
    4 P(I,J)=0.                                                         PRES0070
      DO 6  J=1,NDIMU                                                   PRES0080
    6 CUZ(I,J) = 0.                                                     PRES0090
      DO 8  J=1,NDIMV                                                   PRES0100
    8 CVZ(I,J) = 0.                                                     PRES0110
      DO 10 J=1,10                                                      PRES0120
   10 PH(J,I)=0.                                                        PRES0130
   12 DZ(I) = 0.                                                        PRES0140
      DO 16 I=1,NDIMU                                                   PRES0150
```

```
16 D(I)=0.
   DO 18 I=1,NDIMV
18 S(I)=0.
   DO 26 I=1,NDIMT
26 C(I) = 0.
   C(21) = 1.
   C(36) = 1.
   C(40) = 1.
   C(44) = 1.
   C(61) = 1.
   C(65) = 1.
   C(69) = 1.
   C(76) = 1.
   C(77) = 1.
   C(78) = 1.
   C(91) = 1.
   C(92) = 1.
   C(93) = 1.
   C(106) = 1.
   C(107) = 1.
   C(108) = 1.
   C(121) = 1.
   C(122) = 1.
   C(123) = 1.
   C(136) = 1.
   C(137) = 1.
   C(138) = 1.
   CONRD =  .0174532925
   XJ2 = 1.082645E-3
   XMU = 3.985992E+14
   OMEGA = .7292116E-4
   RPO = 6356173.
   REO = 6378163.
   NCOUNT = 0
   ICOUNT = 1
   KDATAS = -1
   NST = 0
   NSTX = 0
   NSTC = 0
   NSTA = 0
   NPU = 0
   NPV = 0
   FIT = 1
   PQR = 2
   SCRACH = 3
   STATE = 4
   IN = 5
   OUT = 6
   ISV   = 1
   N4(1) = 0
   N4(2) = 1
   N4(3) = 0
   N5(1) = 0
   N5(2) = 1
   N5(3) = 0
   N6(1) = 0
   N6(2) = 1
   N6(3) = 0
   N8(1) = 0
   N8(2) = 1
```

```
      N8(3) = 0
      N9(1) = 0
      N9(2) = 1
      N9(3) = 0
      N10(1) = 0
      N10(2) = 1
      N10(3) = 0
      N11(1) = 0
      N11(2) = 1
      N11(3) = 0
      DO 200 I=1,9
      DFIT(I) = 1.E10
      DTI(I) = 1.E10
  200 NTR(I) = 0
      TIME = 1.E+10
      KG = 2
      AMO = 28.9644
      APO = 1.013250E5
      AR = 8.31432E3
      AGAM = 1.4
      ABET = 1.458E-6
      ASU= 110.4
      GO=9.80665
      CALL INDAT
      CALL SETUP
   99 RETURN
      END
```

```
      SUBROUTINE PROP
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),     COMM0010
     *    NALL,LRK                                                      COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET      ,CETA     , COMM0030
     1    CF(3),JCS ,CXZI      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  , COMM0040
     2    DRDP,DWDH(2),ETA,HO,N4(3)      ,N5(3)      ,N6(3)     ,PAXP(3,30), COMM0050
     3    PE        ,ROE      ,ROEC      ,ROEM      ,SALP      ,SBET      , COMM0060
     4    SETA      ,SREF     ,SXZI      ,TAU       ,TMAS(20)  ,TONE      , COMM0070
     5    TTWO      ,TUW(20)  ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     , COMM0080
     6    VW                  ,XIND1     ,XIND2     ,XLREF     ,XM        , COMM0090
     7    XMAS(20)  ,XNU      ,XQ        ,XQSM      ,XRE       ,XTEMP     , COMM0100
     8    XUW(20)   ,XVW(20)  ,XZI       ,YMAS      ,YMASM                  COMM0110
      COMMON /TWICE/ A(3,3)    ,CPH       ,CT(5)     ,SP(10)    ,ST(5)     COMM0120
     1    ,TPH      ,VO(20)   ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)    COMM0130
     2    ,B(8)     ,CGM(3)   ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)      COMM0140
     3    ,DUF(4)   ,F1(3,40) ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)       COMM0150
     4    ,IDN ,II  ,IP  ,IPC ,JJ  ,JN   ,KA  ,KD(3),KDUM ,KG2              COMM0160
     5    ,K1  ,NNN ,NPTS     ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)      COMM0170
     6    ,RO       ,SPD(5)   ,SPH       ,TRAN(3,3),XX(50)                  COMM0180
      DIMENSION DUB(30,15)                                               COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                       COMM0200
      COMMON /METH2/ AB(3)     ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)    COMM0210
     1    ,AZ(40)   ,C(160)   ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5)  COMM0220
     2    ,CVZ(30,5),D(5)     ,DAT(3,40) ,DATA(3)   ,DATAS(3)  ,DATC(3)    COMM0230
     3    ,DET      ,DFIT(9)  ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET         COMM0240
     4    ,FLOS     ,G(3,30)  ,H(3,5)    ,I  ,ICOUNT,J    ,JNBR,JNBRS,JST  COMM0250
     5    ,K   ,KAR ,KC(3)    ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK   COMM0260
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L  ,LC(3)     ,LCS(3)    ,LS   ,LT   COMM0270
     7    ,L1  ,M   ,MC(5)    ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT         COMM0280
     8    ,MTP(40)  ,MTYP     ,MTYPS     ,N  ,NC(30),NCOUNT    ,NPC(15)    COMM0290
     9    ,NPU ,NPV ,NS(9)    ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)   COMM0300
     A    ,N8(3)    ,N9(3)    ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)   COMM0310
     B    ,PA(3)              ,PM(3)     ,R         ,REO       ,RERP2      COMM0320
     C    ,RPO      ,ROT(5)   ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40)  COMM0330
     D    ,SIGM(3)  ,SIGMS(3) ,SPHDT(5)  ,SUM  ,SUM2,SYG(3,9)  ,TFINAL     COMM0340
     E    ,TFIT(9)  ,TIME     ,TIMES     ,TO        ,TP(40)    ,TQ(40)     COMM0350
     F    ,TR(40)   ,TT(40)   ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)   COMM0360
     G    ,TZERO    ,XJ2      ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)   COMM0370
     H    ,XYCG(20) ,XZCG(20)                                            COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                   COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                        COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK      COMM0410
     A    ,PB(23),RL(23),TMB(23)                                         COMM0420
      COMMON /EXTRA / GO                                                 COMM0430
C-------------CALCULATE EQ.(55B) -------------------------              PROP0010
      DO 4 I=1,10                                                        PROP0020
      DO  4 J=1,NSTX                                                     PROP0030
      SUM = 0.                                                           PROP0040
      L = J+1                                                            PROP0050
      DO 2 K=1,J                                                         PROP0060
    2 SUM = SUM + PH(I,K)*P(K,J)                                         PROP0070
      IF(L.GT.NSTX) GO TO 4                                              PROP0080
      DO 3 K=L,NSTX                                                      PROP0090
    3 SUM = SUM + PH(I,K)*P(J,K)                                         PROP0100
    4 DUB(J,I) = SUM                                                     PROP0110
      DO 10 I=1,10                                                       PROP0120
      DO  8 J=I,10                                                       PROP0130
      SUM = 0.                                                           PROP0140
      DO  6 K=1,NSTX                                                     PROP0150
```

```
    6 SUM = SUM + DUB(K,I)*PH(J,K)
    8 P(I,J) = SUM
   10 CONTINUE
      IF(NSTX-10) 12,16,12
   12 DO 14 I=11,NSTX
      DO 14 J=1,10
   14 P(J,I) = DUB(I,J)
      IF(NPU) 38,38,60
   60 DO 64 I=1,10
      DO 64 K=11,NSTX
      SUM = 0.
      DO 62 J=1,NPU
      II=J+NSTX
   62 SUM = SUM + PH(I,II)*CUZ(K,J)
   64 P(I,K) = P(I,K) + SUM
      GO TO 20
   16 IF(NPU.LE.0) GO TO 38
C------------CALCULATE EQ.(55C) ------------------------
   20 DO 25 J=1,NPU
      DO 24 I=1,10
      SUM = 0.
      DO 22 K=1,NSTX
      SUM = SUM + PH(I,K)*CUZ(K,J)
   22 CONTINUE
      DUB(I,J) = SUM
   24 DUB(I,15) = SUM + PH(I,J+NSTX)*D(J)
      DO 25 I=1,10
   25 CUZ(I,J) = DUB(I,15)
   26 CONTINUE
   30 DO 34 I=1,10
      DO 34 J=I,10
      SUM = 0.
      DO 32 K=1,NPU
   32 SUM = SUM + CUZ(I,K)*PH(J,K+NSTX) + PH(I,K+NSTX)*DUB(J,K)
   34 P(I,J) = P(I,J) + SUM
   36 CONTINUE
   38 IF(NPV) 99,99,40
C------------CALCULATE EQ.(55D) ------------------------
   40 DO 48 J=1,NPV
      DO 44 I=1,10
      SUM = 0.
      DO 42 K=1,NSTX
   42 SUM = SUM + PH(I,K)*CVZ(K,J)
   44 DUB(I,1) = SUM
      DO 46 I=1,10
   46 CVZ(I,J) = DUB(I,1)
   48 CONTINUE
   99 IF(NPC(4).EQ.0) GO TO 75
      IF(NPC(1).EQ.2) GO TO 75
C------------CALCULATE EQ.(55A) ------------------------
      DO 72 I=1,10
      SUM = 0.
      DO 70 J=1,NSTX
   70 SUM = SUM + PH(I,J)*DZ(J)
   72 DUB(I,1) = SUM
      DO 74 I=1,10
   74 DZ(I) = DUB(I,1)
   75 CONTINUE
      ENTRY IDENT
      K = NSTX+NPU
```

```
      DO 52 I=1,10                                                      PROP0760
      DO 50 J=1,K                                                       PROP0770
   50 PH(I,J)=0.                                                        PROP0780
   52 PH(I,I)=1.                                                        PROP0790
      RETURN                                                            PROP0800
      END                                                               PROP0810
```

```
      SUBROUTINE RKUTTA
      DIMENSION PR(3)                                                  RKUT0010
      DIMENSION F1(410),F2(410),F3(410),DELY(410)                      RKUT0020
      COMMON /INTGRL/ P,T,TP,Y(410),DY(410),N,L                        RKUT0030
      DATA PR /.5,.25,.5/                                              RKUT0040
      GO TO (1,2,2,4),L                                                RKUT0050
    1 IF(IG.EQ.0) GO TO 30                                             RKUT0060
      DO 10 I = 1,N                                                    RKUT0070
10    F3(I)=Y(I)                                                       RKUT0080
      L = 2                                                            RKUT0090
      RETURN                                                           RKUT0100
    2 DT = P                                                           RKUT0110
      J = 1                                                            RKUT0120
      T2 = TP - T                                                      RKUT0130
      RT2=T2/P                                                         RKUT0140
      IF(RT2-.99999)22,26,29                                           RKUT0150
22    IF(RT2.GT..00001) GO TO 26                                       RKUT0160
20    L=3                                                              RKUT0170
      RETURN                                                           RKUT0180
26    DT=T2                                                            RKUT0190
29    IG=0                                                             RKUT0200
30    GO TO(31,32,33,34),J                                             RKUT0210
31    T = T+PR(1)*DT                                                   RKUT0220
      DO 310 I=1,N                                                     RKUT0230
      F1(I)=DY(I)*DT                                                   RKUT0240
310   Y(I)=F3(I)+PR(1)*F1(I)                                           RKUT0250
      J = J+1                                                          RKUT0260
      GO TO 38                                                         RKUT0270
33    T = T+PR(1)*DT                                                   RKUT0280
32    T2=DT/PR(1)                                                      RKUT0290
      DO 320 I=1,N                                                     RKUT0300
      RT2=DY(I)*T2                                                     RKUT0310
      F1(I)=F1(I)+RT2                                                  RKUT0320
320   Y(I)=F3(I)+PR(J)*RT2                                             RKUT0330
35    J = J + 1                                                        RKUT0340
      RETURN                                                           RKUT0350
34    DO 340 I=1,N                                                     RKUT0360
340   F1(I)=F1(I)+DY(I)*DT                                             RKUT0370
   37 DO 39 I = 1,N                                                    RKUT0380
      DELY(I)=F1(I)/6.+DELY(I)                                         RKUT0390
39    Y(I)=F2(I)+DELY(I)                                               RKUT0400
      IG □ 1                                                           RKUT0410
      RETURN                                                           RKUT0420
    4 DT=P                                                             RKUT0430
      IG = 1                                                           RKUT0440
      J = 1                                                            RKUT0450
      DO 40  I = 1,N                                                   RKUT0460
      DELY(I) = 0.D0                                                   RKUT0470
40    F2(I)=Y(I)                                                       RKUT0480
   38 L = 1                                                            RKUT0490
      RETURN                                                           RKUT0500
      END                                                              RKUT0510
```

```
      SUBROUTINE SETUP
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),      COMM0010
     *        NALL,LRK                                                    COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP     ,CBET     ,CETA     ,   COMM0030
     1        CF(3),JCS ,CXZI     ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  ,  COMM0040
     2        DRDP,DWDH(2),ETA,HO,N4(3)   ,N5(3)    ,N6(3)    ,PAXP(3,30),  COMM0050
     3        PE       ,ROE      ,ROEC     ,ROEM     ,SALP     ,SBET     ,   COMM0060
     4        SETA     ,SREF     ,SXZI     ,TAU      ,TMAS(20) ,TONE     ,   COMM0070
     5        TTWO     ,TUW(20)  ,TVW(20)  ,UW       ,VA(5)    ,VB(5)    ,   COMM0080
     6        VW                 ,XIND1    ,XIND2    ,XLREF    ,XM       ,   COMM0090
     7        XMAS(20) ,XNU      ,XQ       ,XQSM     ,XRE      ,XTEMP    ,   COMM0100
     8        XUW(20)  ,XVW(20)  ,XZI      ,YMAS     ,YMASM                  COMM0110
      COMMON /TWICE/ A(3,3)    ,CPH      ,CT(5)    ,SP(10)   ,ST(5)         COMM0120
     1        ,TPH     ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)      COMM0130
     2        ,B(8)    ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)       COMM0140
     3        ,DUF(4)  ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)         COMM0150
     4        ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2              COMM0160
     5        ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)      COMM0170
     6        ,RO      ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)                  COMM0180
      DIMENSION DUB(30,15)                                                    COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                            COMM0200
      COMMON /METH2/ AB(3)     ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)        COMM0210
     1        ,AZ(40)  ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)     COMM0220
     2        ,CVZ(30,5),D(5)    ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)       COMM0230
     3        ,DET     ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET            COMM0240
     4        ,FLOS    ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J  ,JNBR,JNBRS,JST      COMM0250
     5        ,K   ,KAR ,KC(3)   ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK      COMM0260
     6        ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)   ,LCS(3)   ,LS   ,LT      COMM0270
     7        ,L1  ,M   ,MC(5)   ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT            COMM0280
     8        ,MTP(40) ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)       COMM0290
     9        ,NPU ,NPV ,NS(9)   ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)     COMM0300
     A        ,N8(3)   ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)      COMM0310
     B        ,PA(3)             ,PM(3)    ,R        ,REO      ,RERP2         COMM0320
     C        ,RPO     ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)     COMM0330
     D        ,SIGM(3) ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL        COMM0340
     E        ,TFIT(9) ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)        COMM0350
     F        ,TR(40)  ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)      COMM0360
     G        ,TZERO   ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)      COMM0370
     H        ,XYCG(20) ,XZCG(20)                                             COMM0380
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                        COMM0390
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                             COMM0400
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK           COMM0410
     A  ,PB(23),RL(23),TMB(23)                                                COMM0420
      COMMON /EXTRA / GO                                                      COMM0430
      DIMENSION DUA(10,10)                                                    SETP0010
      WRITE(OUT,1000)                                                         SETP0020
 1000 FORMAT(1H1,27H CONTROLS ARE SPECIFIED FOR)                              SETP0030
      IF(NPC(1)-1) 10,11,12                                                   SETP0040
   10 WRITE(OUT,1001) TO,TFINAL                                               SETP0050
 1001 FORMAT(5X,28H*FILTERING RUN FROM TIME TO=F9.4,11H TO TFINAL=F9.4)       SETP0060
      GO TO 13                                                                SETP0070
   11 WRITE(OUT,1002) TO,TFINAL                                               SETP0080
 1002 FORMAT(5X,32H*DETERMINISTIC RUN FROM TIME TO=F9.4,11H TO TFINAL=F9SETP0090
     1.4)                                                                     SETP0100
      GO TO 24                                                                SETP0110
   12 WRITE(OUT,1003) TO,TFINAL                                               SETP0120
 1003 FORMAT(5X,33H*ERROR ANALYSIS RUN FROM TIME TO=F9.4,11H TO TFINAL=FSETP0130
     19.4)                                                                    SETP0140
      GO TO 21                                                                SETP0150
```

```
   13 IF(NPC(4))14,14,15
   14 WRITE(OUT,1004)
 1004 FORMAT(5X,18H*UPDATED REFERENCE)
      GO TO 16
   15 WRITE(OUT,1005)
 1005 FORMAT(5X,21H*NONUPDATED REFERENCE)
   16 IF(NPC(8)-1) 17,18,19
   17 WRITE(OUT,1006)
 1006 FORMAT(5X,57H*SMOOTH DETERMINISTICALLY, NO RESIDUALS NOR LOSS FUNC
     XTION)
      GO TO 20
   18 WRITE(OUT,1007)
 1007 FORMAT(5X,64H*SMOOTH DETERMINISTICALLY, CALCULATE RESIDUALS AND LO
     XSS FUNCTION)
      GO TO 20
   19 IF(NPC(8)-3) 170,172,174
  170 WRITE(OUT,1027)
 1027 FORMAT(5X,49H*SMOOTH COVARIANCE,NO RESIDUALS NOR LOSS FUNCTION)
      GO TO 20
  172 WRITE(OUT,1028)
 1028 FORMAT(5X,56H*SMOOTH COVARIANCE,CALCULATE RESIDUALS AND LOSS FUNCT
     XION)
      GO TO 20
  174 WRITE(OUT,1029)
 1029 FORMAT(5X,13H*NO SMOOTHING)
   20 WRITE(OUT,1009) NPC(7)
 1009 FORMAT(5X,12H*FILTER FOR I2,11H ITERATIONS)
   21 IF(NPC(9).EQ.0) GO TO 23
   22 WRITE(OUT,1010)
 1010 FORMAT(5X,28H*SCALAR PROCESS FITTING DATA)
      GO TO 24
   23 WRITE(OUT,1011)
 1011 FORMAT(5X,36H*VECTOR PROCESS FITTING DATA TRIPLES)
   24 IF(NPC(2)) 25,25,26
   25 WRITE(OUT,1012)
 1012 FORMAT(5X,52H*INPUT AND OUTPUT IN METRIC UNITS(M,M/SEC,M/SEC2,KG))
      GO TO 27
   26 WRITE(OUT,1013)
 1013 FORMAT(5X,62H*INPUT AND OUTPUT IN ENGLISH UNITS (FT,FT/SEC,FT/SEC2
     X,LB,SLUG))
   27 WRITE(OUT,1014) REO,RPO,OMEGA,XMU,XJ2,GO
 1014 FORMAT(5X,18H*PLANET PARAMETERS/20X,11HEQUIT.RAD.=E15.8,13H  POLAR
     1 RAD.=E15.8,13H  ROTATION  =E15.8/20X,11HMU(GRAV.) =E15.8,13H J2(G
     2RAV. ) =E15.8,7X,6HG(0) =E15.8)
      IF(NPC(14)-1) 280,282,284
  280 WRITE(OUT,1040)
 1040 FORMAT(5X,32H*ATMOSPHERE IS 1962 U.S.STANDARD)
      NPTS=22
      HB( 1) = 0.
      HB( 2) = 11000.
      HB( 3) = 20000.
      HB( 4) = 32000.
      HB( 5) = 47000.
      HB( 6) = 52000.
      HB( 7) = 61000.
      HB( 8) = 79000.
      HB( 9) = 88743.
      HB(10) = 98451.235
      HB(11) = 108128.89
      HB(12) = 117776.67
```

```
      HB(13) = 146542.06
      HB(14) = 156071.67
      HB(15) = 165572.08
      HB(16) = 184465.82
      HB(17) = 221968.74
      HB(18) = 286479.92
      HB(19) = 376320.01
      HB(20) = 463539.66
      HB(21) = 548251.81
      HB(22) = 630563.08
      TMB( 1) = 288.15
      TMB( 2) = 216.65
      TMB( 3) = 216.65
      TMB( 4) = 228.65
      TMB( 5) = 270.65
      TMB( 6) = 270.65
      TMB( 7) = 252.65
      TMB( 8) = 180.65
      TMB( 9) = 180.65
      TMB(10) = 210.65
      TMB(11) = 260.65
      TMB(12) = 360.65
      TMB(13) = 960.65
      TMB(14) = 1110.65
      TMB(15) = 1210.65
      TMB(16) = 1350.65
      TMB(17) = 1550.65
      TMB(18) = 1830.65
      TMB(19) = 2160.65
      TMB(20) = 2420.65
      TMB(21) = 2590.65
      TMB(22) = 2700.65
      GO TO 286
  282 WRITE(OUT,1041)
 1041 FORMAT(5X,24H*ATMOSPHERE IS 1959 ARDC)
      NPTS= 12
      HB( 1)= 0.
      HB( 2)= 11000.
      HB( 3)= 25000.
      HB( 4)= 47000.
      HB( 5)= 53000.
      HB( 6)= 79000.
      HB( 7)= 90000.
      HB( 8)= 105000.
      HB( 9)= 160000.
      HB(10)= 170000.
      HB(11)= 200000.
      HB(12)= 700000.
      TMB( 1)= 288.16
      TMB( 2)= 216.66
      TMB( 3)= 216.66
      TMB( 4)= 282.66
      TMB( 5)= 282.66
      TMB( 6)= 165.66
      TMB( 7)= 165.66
      TMB( 8)= 225.66
      TMB( 9)= 1325.66
      TMB(10)= 1425.66
      TMB(11)= 1575.66
      TMB(12)= 3325.66
```

```
      GO TO 286
  284 WRITE(OUT,1042)
 1042 FORMAT(5X,29H*ATMOSPHERE SPECIFIED BY USER)
      NPTS=K1
      GO TO 292
  286 IF(NPC(2)) 288,292,288
  288 AR= AR/.167225472
      APO = APO/47.880258
      ABET= ABET*.01556708573
       ASU = ASU*1.8
      DO 290 I=1,NPTS
      HB(I)=HB(I)/.3048
  290 TMB(I)=TMB(I)*1.8
  292 JK=3
      ATM(1)= GO*AMO/AR
      ATM(2)= AMO/AR
      ATM(3)= SQRT(AGAM*AR/AMO)
      PB(1)=APO
      ISV=1
      DO 294 K=2,NPTS
      RL(K-1)=(TMB(K)-TMB(K-1))/(HB(K)-HB(K-1))
      SP(1)= HB(K)-.0001
      CALL ATMDAT(SP(1),PE,ROE,AS,TM,VMU,H)
  294 PB(K)=PE
      JK=NPC(14)
      RL(NPTS)=0
      HB(NPTS+1)=1.E15
      WRITE(OUT,1043) AMO,AR,AGAM,APO,ABET,ASU,(HB(I),I=1,NPTS)
 1043 FORMAT(26X,5H MO =E15.8,13H     R(STAR) =E15.8,13H      CP/CV =E15.
     18/25X,6HP(0) =E15.8,7X,6HBETA =E15.8,10X,3HS =E15.8/8X,4HHB =6E18.
     28/(12X,6E18.8))
      WRITE(OUT,1044) (TMB(I),I=1,NPTS)
 1044 FORMAT(7X,5HTMB =6E18.8/(12X,6E18.8))
      WRITE(OUT,1046) (PB(I),I=1,NPTS)
 1046 FORMAT(8X,4HPB =6E18.8/(12X,6E18.8))
      WRITE(OUT,1047)
 1047 FORMAT(5X,26H*INITIAL STATE ESTIMATE IS)
      IF(NPC(3)-1) 28,29,30
   28 WRITE(OUT,1015) (ZO(I),I=1,10)
 1015 FORMAT(20X,1HU,9X,1H=E15.8,3H  V,9X,1H=E15.8,3H  W,9X,1H=E15.8/20X
     1,11HALTITUDE   =E15.8,13H  GEOD.LAT. =E15.8,13H  LONGITUDE =E15.8/2
     20X,2HEO,8X,1H=E15.8,4H  E1,8X,1H=E15.8,4H  E2,8X,1H=E15.8/20X,2HE3
     3,8X,1H=E15.8)
      GO TO 31
   29 WRITE(OUT,1016) (ZO(I),I=1,9)
 1016 FORMAT(20X,11HVELOCITY   =E15.8,13H  GAMMA      =E15.8,13H  LAMBDA
     1  =E15.8/20X,11HALTITUDE   =E15.8,13H  GEOD.LAT. =E15.8,13H  LONGIT
     2UDE =E15.8/20X,11HPSI(BAR)   =E15.8,13H  THETA(BAR)=E15.8,13H  PHI(
     3BAR)  =E15.8)
      GO TO 31
   30 WRITE(OUT,1017) (ZO(I),I=1,9)
 1017 FORMAT(20X,11HVELOCITY   =E15.8,13H  GAMMA      =E15.8,13H  LAMBDA
     1  =E15.8/20X,11HALTITUDE   =E15.8,13H  GEOD.LAT. =E15.8,13H  LONGIT
     2UDE =E15.8/20X,11HSIGMA      =E15.8,13H  BETA       =E15.8,13H  ALPH
     3A     =E15.8)
   31 RERP2=(REO/RPO)**2
      XMUJ = 3.*XMU*XJ2*REO**2
      ATM(4)= REO/SQRT(1.-(RERP2-1.)*SIN(.79486646)**2)
      IF(NPC(1)-1)32,40,32
   32 WRITE(OUT,1018) NSTA
```

```
1018 FORMAT(5X,22H*FITTING DATA IS FROM I2,8H SOURCES)
     KSS=0
     DO 42 I=1,NSTA
     IF(NS(I)-6) 33,44,45
  33 MT=NS(I)
     N = 73 +15*MT
     WRITE(OUT,1019)  I,MT,C(N),C(N+1),C(N+2)
1019 FORMAT(10X,I1,19H. TRACKING STATION I1,12H, GEOD.LAT.=F11.6,12H  L
    XONGITUDE=F11.6,12H  ALTITUDE =F11.4,19H ABOVE REF. SURFACE)
     C(N)=C(N)*CONRD
     C(N+1)=C(N+1)*CONRD
     C(N) = ATAN2(SIN(C(N)),RERP2*COS(C(N)))
     NT=N+2
     MTYP = MT
     CALL STAT
     DO 36 J=1,NSTC
     IF(NC(J)-N) 36,35,35
  35 IF(NC(J)-NT)37,37,36
  36 CONTINUE
     GO TO 38
  37 KSS=KSS+1
     NSS(KSS) = MT
  38 MO(MT) = 0
     DO 39 J=1,NSTC
     IF(NC(J).LT.N-6) GO TO 39
     IF(NC(J).LE.N-4) GO TO 40
  39 CONTINUE
     IF(C(N-6).NE.0.) GO TO 40
     IF(C(N-5).NE.0.) GO TO 40
     IF(C(N-4).NE.0.) GO TO 40
     GO TO 42
  44 WRITE(OUT,1020)  I,C(151),C(152)
1020 FORMAT(10X,I1,27H. AIRBORNE RADAR, DELTA(P)=F11.6,11H  DELTA(Y)=F1
    X1.6)
     GO TO 42
  45 IF(NS(I)-8) 43,41,47
  47 WRITE(OUT,1032) I
1032 FORMAT(10X,I1,20H. ACCELEROMETER DATA )
     GO TO 42
  41 WRITE(OUT,1030) I
1030 FORMAT(I11,15H. POSITION DATA)
     GO TO 42
  43 WRITE(OUT,1031) I
1031 FORMAT(I11,15H. VELOCITY DATA)
     GO TO 42
  40 MO(MT) = 1
  42 CONTINUE
  46 ZO(5) = ZO(5)*CONRD
     ZO(6)=ZO(6)*CONRD
     NSTX = NSTC + 10
     ZO(5) = ATAN2(SIN(ZO(5)),RERP2*COS(ZO(5)))
     CPH=COS(ZO(5))
     SPH=SIN(ZO(5))
     RO =REO/SQRT(1.+(RERP2-1.)*SPH**2)
     R= ZO(4)+RO
     ZO(4)=R-REO
     DO 48 I=1,10
  48 X(I)=ZO(I)
     IF(NPC(3).EQ.0) GO TO 100
  50 DO 51 I=7,9
```

```
51 ZO(I)=ZO(I)*CONRD                                                  SETP2560
   DO 52 I=1,2                                                        SETP2570
   ZO(I+1) = ZO(I+1)*CONRD                                            SETP2580
   ST(I)=SIN(ZO(I+1))                                                 SETP2590
52 CT(I)=COS(ZO(I+1))                                                 SETP2600
   DO 53 I=3,5                                                        SETP2610
   ST(I)=SIN(ZO(I+4)/2.)                                              SETP2620
53 CT(I)=COS(ZO(I+4)/2.)                                              SETP2630
   X(3)=ZO(1)*CT(1)                                                   SETP2640
   X(1)=X(3)*CT(2)                                                    SETP2650
   X(2)=X(3)*ST(2)+R*OMEGA*CPH                                        SETP2660
   X(3)=-ZO(1)*ST(1)                                                  SETP2670
   UW = TAB(ZO(4),N5(1),TUW(1),XUW(1))                                SETP2680
   VW = TAB(ZO(4),N6(1),TVW(1),XVW(1))                                SETP2690
   X(1) = X(1)-UW                                                     SETP2700
   X(2) = X(2)-VW                                                     SETP2710
   DO 54 I=1,10                                                       SETP2720
   DO 54 J=1,9                                                        SETP2730
54 DUB(I,J)=0.                                                        SETP2740
   DUB(1,1)= CT(1)*CT(2)                                              SETP2750
   DUB(2,1)= CT(1)*ST(2)                                              SETP2760
   DUB(3,1)=-ST(1)                                                    SETP2770
   SP(1)= ZO(1)*ST(1)                                                 SETP2780
   DUB(1,2)=-SP(1)*CT(2)                                              SETP2790
   DUB(2,2)=-SP(1)*ST(2)                                              SETP2800
   SP(1)= ZO(1)*CT(1)                                                 SETP2810
   DUB(3,2)=-SP(1)                                                    SETP2820
   DUB(1,3)=-SP(1)*ST(2)                                              SETP2830
   DUB(2,3)= SP(1)*CT(2)                                              SETP2840
   DUB(2,4)= OMEGA*CPH                                                SETP2850
   DUB(2,5)=-R*OMEGA*SPH                                              SETP2860
   DUB(4,4)=1.                                                        SETP2870
   DUB(5,5)=1.                                                        SETP2880
   DUB(6,6)=1.                                                        SETP2890
   IF(NPC(3)-1) 100, 56, 60                                           SETP2900
56 DUA(1,1)= CT(3)*CT(4)*CT(5)                                        SETP2910
   DUA(2,1)= CT(3)*CT(4)*ST(5)                                        SETP2920
   DUA(3,1)= CT(3)*ST(4)*CT(5)                                        SETP2930
   DUA(4,1)= ST(3)*CT(4)*CT(5)                                        SETP2940
   DUA(1,2)= ST(3)*ST(4)*ST(5)                                        SETP2950
   DUA(2,2)= ST(3)*ST(4)*CT(5)                                        SETP2960
   DUA(3,2)= ST(3)*CT(4)*ST(5)                                        SETP2970
   DUA(4,2)= CT(3)*ST(4)*ST(5)                                        SETP2980
   X(7) = DUA(1,1)+DUA(1,2)                                           SETP2990
   X(8) = DUA(2,1)-DUA(2,2)                                           SETP3000
   X(9) = DUA(3,1)+DUA(3,2)                                           SETP3010
   X(10)= DUA(4,1)-DUA(4,2)                                           SETP3020
   IF(NPC(1).EQ.1) GO TO 100                                          SETP3030
   DUB(7,7) = -X(10)/2.                                               SETP3040
   DUB(8,7) =-X(9)/2.                                                 SETP3050
   DUB(9,7) = X(8)/2.                                                 SETP3060
   DUB(10,7) = X(7)/2.                                                SETP3070
   DUB(7,8) =-(DUA(3,1) - DUA(3,2))/2.                                SETP3080
   DUB(8,8) =-(DUA(4,1) + DUA(4,2))/2.                                SETP3090
   DUB(9,8) = (DUA(1,1) - DUA(1,2))/2.                                SETP3100
   DUB(10,8)=-(DUA(2,1) + DUA(2,2))/2.                                SETP3110
   DUB(7,9) =-X(8)/2.                                                 SETP3120
   DUB(8,9) = X(7)/2.                                                 SETP3130
   DUB(9,9) = X(10)/2.                                                SETP3140
   DUB(10,9)=-X(9)/2.                                                 SETP3150
```

```
      GO TO 70
   60 ST(1) = SIN(.5*ZO(2))
      CT(1) = COS(.5*ZO(2))
      ST(2) = SIN(.5*ZO(3))
      CT(2) = COS(.5*ZO(3))
      DUE(1)= CT(1)*CT(2)
      DUE(2)=-ST(1)*ST(2)
      DUE(3)= ST(1)*CT(2)
      DUE(4)= CT(1)*ST(2)
      DUA(1,1)= CT(3)*CT(4)*CT(5)
      DUA(2,1)= ST(3)*CT(4)*CT(5)
      DUA(3,1)= CT(3)*CT(4)*ST(5)
      DUA(4,1)= ST(3)*CT(4)*ST(5)
      DUA(1,2)= ST(3)*ST(4)*ST(5)
      DUA(2,2)= CT(3)*ST(4)*ST(5)
      DUA(3,2)= ST(3)*ST(4)*CT(5)
      DUA(4,2)= CT(3)*ST(4)*CT(5)
      DUF(1)= DUA(1,1)-DUA(1,2)
      DUF(2)= DUA(2,1)+DUA(2,2)
      DUF(3)= DUA(3,1)+DUA(3,2)
      DUF(4)= DUA(4,1)-DUA(4,2)
      X(7) = DUE(1)*DUF(1)-DUE(2)*DUF(2)-DUE(3)*DUF(3)-DUE(4)*DUF(4)
      X(8) = DUE(1)*DUF(2)+DUE(2)*DUF(1)+DUE(3)*DUF(4)-DUE(4)*DUF(3)
      X(9) = DUE(1)*DUF(3)-DUE(2)*DUF(4)+DUE(3)*DUF(1)+DUE(4)*DUF(2)
      X(10)= DUE(1)*DUF(4)+DUE(2)*DUF(3)-DUE(3)*DUF(2)+DUE(4)*DUF(1)
      IF(NPC(1).EQ.1) GO TO 100
      DUB(7,2) = -X(9)/2. - DUE(2)*DUF(4) + DUE(4)*DUF(2)
      DUB(8,2) =  X(10)/2.- DUE(2)*DUF(3) - DUE(4)*DUF(1)
      DUB(9,2) =  X(7)/2. + DUE(2)*DUF(2) + DUE(4)*DUF(4)
      DUB(10,2)= -X(8)/2. + DUE(2)*DUF(1) - DUE(4)*DUF(3)
      DUB(7,3) = -X(10)/2.
      DUB(8,3) = -X(9)/2.
      DUB(9,3) =  X(8)/2.
      DUB(10,3)=  X(7)/2.
      DUB(7,7) = -X(8)/2. + DUE(3)*DUF(4) - DUE(4)*DUF(3)
      DUB(8,7) =  X(7)/2. + DUE(3)*DUF(3) + DUE(4)*DUF(4)
      DUB(9,7) = -X(10)/2.- DUE(3)*DUF(2) + DUE(4)*DUF(1)
      DUB(10,7)=  X(9)/2. - DUE(3)*DUF(1) - DUE(4)*DUF(2)
      DUF(1) = DUA(1,1)+DUA(1,2)
      DUF(2) = DUA(2,1)-DUA(2,2)
      DUF(3) = DUA(3,1)-DUA(3,2)
      DUF(4) = DUA(4,1)+DUA(4,2)
      DUB(7,8) = (-DUE(1)*DUF(4)-DUE(2)*DUF(3)-DUE(3)*DUF(2)+DUE(4)*DUF(
     X1))/2.
      DUB(8,8) =  (DUE(1)*DUF(3)-DUE(2)*DUF(4)-DUE(3)*DUF(1)-DUE(4)*DUF(
     X2))/2.
      DUB(9,8) =  (DUE(1)*DUF(2)+DUE(2)*DUF(1)-DUE(3)*DUF(4)+DUE(4)*DUF(
     X3))/2.
      DUB(10,8)= (-DUE(1)*DUF(1)+DUE(2)*DUF(2)-DUE(3)*DUF(3)-DUE(4)*DUF(
     X4))/2.
      DUB(7,9) = -X(9)/2.
      DUB(8,9) = -X(10)/2.
      DUB(9,9) =  X(7)/2.
      DUB(10,9)=  X(8)/2.
   70 DO 74 I=1,10
      DO 74 J=1,NSTX
      SUM = 0.
      DO 72 K=1,9
   72 SUM = SUM + DUB(I,K)*P(K,J)
   74 PH(I,J) = SUM
```

```
      DO 80 I=1,10
      DO 78 J=I,10
      SUM = 0.
      DO 76 K=1,10
   76 SUM = SUM + PH(I,K)*DUB(J,K)
   78 P(I,J) = SUM
      DO 80 J=11,NSTX
   80 P(I,J) = PH(I,J)
      IF(NPU.EQ.0) GO TO 88
      DO 84 I=1,10
      DO 84 J=1,NPU
      SUM = 0.
      DO 82 K=1,9
   82 SUM = SUM + DUB(I,K)*CUZ(K,J)
   84 PH(I,J) = SUM
      DO 86 I=1,10
      DO 86 J=1,NPU
   86 CUZ(I,J) = PH(I,J)
   88 IF(NPV.EQ.0) GO TO 100
      DO 92 I=1,10
      DO 92 J=1,NPV
      SUM = 0.
      DO 90 K=1,9
   90 SUM = SUM + DUB(I,K)*CVZ(K,J)
   92 PH(I,J) = SUM
      DO 94 I=1,10
      DO 94 J=1,NPV
   94 CVZ(I,J) = PH(I,J)
  100 WRITE(OUT,3021)
      WRITE(OUT,1021) (I,I,X(I),P(I,I),I=1,10)
 3021 FORMAT(//,24H STATE VECTOR COMPONENTS/43H COMP.   ID.NO.      EST.
     *VALUE      VARIANCE)
 1021 FORMAT(1X,I4,4X,I4,5X,2E15.8)
      IF(NSTC) 102,104,102
  102 WRITE(OUT,1022)
 1022 FORMAT(42H MODEL PARAMETERS IN EXPANDED STATE VECTOR)
      DO 103 I=1,NSTC
      LT=NC(I)
      MT=I+10
  103 WRITE(OUT,1023) MT,LT,C(LT),P(MT,MT)
 1023 FORMAT(1X,I4,4X,I4,5X,2E15.8)
  104 IF(NPU) 106,108,106
  106 WRITE(OUT,1024)
 1024 FORMAT(55H RANDOM MODEL PARAMETERS NOT BEING ESTIMATED (U-VECTOR))
      DO 107 I=1,NPU
      LT=MC(I)
  107 WRITE(OUT,1023) I,LT,C(LT),D(I)
  108 IF(NPV) 110,112,110
  110 WRITE(OUT,1025)
 1025 FORMAT(61H RANDOM MEASUREMENT PARAMETERS NOT BEING ESTIMATED (V-VE
     XCTOR))
      DO 111 I=1,NPV
      LT=MCC(I)
  111 WRITE(OUT,1023) I,LT,C(LT),S(I)
  112 LT=9
      IF(NPC(3).EQ.0) LT = 10
      IF(NST-LT) 114,120,114
  114 WRITE(OUT,1026)
 1026 FORMAT(48H IMPROPER NUMBER OF STATE VARIABLES ARE INPUTTED)
      CALL EXIT
```

```
120 IF(NPC(1)-1) 124,122,124                                            SETP4360
122 NALL=10                                                             SETP4370
    GO TO 128                                                           SETP4380
124 NALL = 10*(NSTX+NPU+1)                                              SETP4390
    K = NSTX + NPU                                                      SETP4400
    DO 126 I =1,10                                                      SETP4410
    DO 125 J=1,K                                                        SETP4420
125 PH(I,J) = 0.                                                        SETP4430
126 PH(I,I) = 1.                                                        SETP4440
128 IF(NPC(1)-2) 144,130,144                                            SETP4450
130 IF(NSTA-1) 144,144,132                                              SETP4460
132 DO 142 I=2,NSTA                                                     SETP4470
    MT=NS(I-1)                                                          SETP4480
    TT=DTF(MT)                                                          SETP4490
    DO 136 J=I,NSTA                                                     SETP4510
    K=NS(J)                                                             SETP4520
    IF(TT-DTF(K)) 135,135,136                                           SETP4530
135 TT=DTF(K)                                                           SETP4540
    LT=J                                                                SETP4550
136 CONTINUE                                                            SETP4560
137 IP=NS(I-1)                                                          SETP4570
138 NS(I-1)=NS(LT)                                                      SETP4580
139 NS(LT)=IP                                                           SETP4590
142 CONTINUE                                                            SETP4600
144 KDATA=-1                                                            SETP4760
    WRITE(SCRACH) NSTX,NPU,NPV,((P(I,J),I=1,NSTX),J=1,NSTX),            SETP4770
   1     ((CUZ(I,J),I=1,NSTX),J=1,NPU),((CVZ(I,J),I=1,NSTX),J=1,NPV)    SETP4780
    ET = TO                                                             SETP4790
    TZERO = TO                                                          SETP4800
    TIMES=TZERO-10.                                                     SETP4810
    T = TO                                                              SETP4820
    KDAP = 1                                                            SETP4830
    KPROP=-1                                                            SETP4840
    KDATAS =-1                                                          SETP4850
    C(151) = C(151)*CONRD                                               SETP4860
    C(152) = C(152)*CONRD                                               SETP4870
    RETURN                                                              SETP4880
    END                                                                 SETP4890
```

```
      SUBROUTINE SMOOTH
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),        COMM0010
     *     NALL,LRK                                                        COMM0020
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET      ,CETA      ,  COMM0030
     1     CF(3),JCS ,CXZI      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  ,  COMM0040
     2     DRDP,DWDH(2),ETA,HO,N4(3)     ,N5(3)     ,N6(3)     ,PAXP(3,30),  COMM0050
     3     PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET      ,  COMM0060
     4     SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE      ,  COMM0070
     5     TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     ,  COMM0080
     6     VW                   ,XIND1     ,XIND2     ,XLREF     ,XM        ,  COMM0090
     7     XMAS(20)  ,XNU       ,XQ        ,XQSM      ,XRE       ,XTEMP     ,  COMM0100
     8     XUW(20)   ,XVW(20)   ,XZI       ,YMAS      ,YMASM                   COMM0110
      COMMON /TWICE/ A(3,3)     ,CPH       ,CT(5)     ,SP(10)    ,ST(5)       COMM0120
     1     ,TPH      ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)    COMM0130
     2     ,B(8)     ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)       COMM0140
     3     ,DUF(4)   ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)       COMM0150
     4     ,IDN ,II  ,IP  ,IPC ,JJ  ,JN    ,KA  ,KD(3),KDUM ,KG2              COMM0160
     5     ,K1  ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)      COMM0170
     6     ,RO       ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)                  COMM0180
      DIMENSION DUB(30,15)                                                     COMM0190
      EQUIVALENCE (AG(1),DUB(1,1))                                             COMM0200
      COMMON /METH2/ AB(3)      ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)      COMM0210
     1     ,AZ(40)   ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5)   COMM0220
     2     ,CVZ(30,5),D(5)      ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)     COMM0230
     3     ,DET      ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET          COMM0240
     4     ,FLOS     ,G(3,30)   ,H(3,5)    ,I  ,ICOUNT,J    ,JNBR,JNBRS,JST   COMM0250
     5     ,K   ,KAR ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK    COMM0260
     6     ,KN  ,KO6 ,KPROP,KS,KSM,KSS,L   ,LC(3)     ,LCS(3)    ,LS   ,LT    COMM0270
     7     ,L1  ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT          COMM0280
     8     ,MTP(40)  ,MTYP      ,MTYPS     ,N  ,NC(30),NCOUNT    ,NPC(15)     COMM0290
     9     ,NPU ,NPV ,NS(9)     ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)     COMM0300
     A     ,N8(3)    ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)    COMM0310
     B     ,PA(3)               ,PM(3)     ,R         ,REO       ,RERP2       COMM0320
     C     ,RPO      ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40)   COMM0330
     D     ,SIGM(3)  ,SIGMS(3)  ,SPHDT(5)  ,SUM  ,SUM2,SYG(3,9)  ,TFINAL      COMM0340
     E     ,TFIT(9)  ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)      COMM0350
     F     ,TR(40)   ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)    COMM0360
     G     ,TZERO    ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)    COMM0370
     H     ,XYCG(20) ,XZCG(20)                                                COMM0380
       INTEGER OUT,FIT,STATE,SCRACH,PQR                                        COMM0390
       COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                             COMM0400
       COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK           COMM0410
     A  ,PB(23),RL(23),TMB(23)                                                 COMM0420
       COMMON /EXTRA / GO                                                      COMM0430
       CALL IDENT                                                              SMTH0010
       FLOS = 0.                                                               SMTH0020
       TIME = TZERO - 10.                                                      SMTH0030
       NALLSV=NALL                                                             SMTH0040
       MOPTSV=NPC(1)                                                           SMTH0050
       DCOMP=-DCOMP                                                            SMTH0060
       IF(NPC(1)-1) 4,6,2                                                      SMTH0070
    2  IF(NPC(8)-4) 12,6,12                                                    SMTH0080
    4  IF(NPC(7).GT.ICOUNT)     IF(NPC(8)-4) 16,6,16                           SMTH0090
       IF(NPC(8).LE.1) GO TO 8                                                 SMTH0100
       IF(NPC(8).LE.3) GO TO 14                                                SMTH0110
    6  TZERO = TO                                                              SMTH0120
       NPC(7) = ICOUNT                                                         SMTH0130
       GO TO 62                                                                SMTH0140
    8  NPC(1) = 1                                                              SMTH0150
```

```
      NALL=10                                                           SMTH0160
    9 IF(NPC(8).EQ.0) GO TO 12                                          SMTH0170
      IF(NPC(8).EQ.2) GO TO 12                                          SMTH0180
   10 KSM = 1                                                           SMTH0190
      GO TO 18                                                          SMTH0200
   12 KSM = 0                                                           SMTH0210
      GO TO 18                                                          SMTH0220
   14 IF(NPC(8)-2) 12,12,10                                             SMTH0230
   16 NPC(1) = 1                                                        SMTH0240
      NALL=10                                                           SMTH0250
      GO TO 9                                                           SMTH0260
   18 T2 = TO                                                           SMTH0270
      WRITE(OUT,1004)                                                   SMTH0280
   20 CALL INTAG                                                        SMTH0290
      TO=T2                                                             SMTH0300
      IF(ET.LE.TIME) GO TO 150                                          SMTH0310
      IF(NPC(1).NE.1) CALL PROP                                         SMTH0320
   24 CALL OUTPUT                                                       SMTH0330
      L1=1                                                              SMTH0340
   28 IF(T2-TZERO) 60,60,30                                             SMTH0350
   30 ET=ET-DET                                                         SMTH0360
      IF(KSM.LE.0) GO TO 38                                             SMTH0370
   36 TIME = TYM(KG)                                                    SMTH0380
   37 IF(TIME.LE.ET) GO TO 38                                           SMTH0390
      T2 = TIME                                                         SMTH0400
      GO TO 20                                                          SMTH0410
   38 T2 = ET                                                           SMTH0420
      GO TO 20                                                          SMTH0430
   60 NALL=NALLSV                                                       SMTH0440
      NPC(1)=MOPTSV                                                     SMTH0450
      DCOMP=-DCOMP                                                      SMTH0460
      IF(ICOUNT-NPC(7)) 64,62,62                                        SMTH0470
   62 ET=TFINAL                                                         SMTH0480
      NPC(1)=1                                                          SMTH0490
      L1=0                                                              SMTH0500
      GO TO 99                                                          SMTH0510
   64 ICOUNT=ICOUNT+1                                                   SMTH0520
      DO 66 I=1,9                                                       SMTH0530
   66 NTR(I) = 0                                                        SMTH0540
      REWIND FIT                                                        SMTH0550
      REWIND PQR                                                        SMTH0560
      KDATAS=-1                                                         SMTH0570
      KDATA=-1                                                          SMTH0580
      KPROP=-1                                                          SMTH0590
      NCOUNT=0                                                          SMTH0600
      L1=1                                                              SMTH0610
      BACKSPACE SCRACH                                                  SMTH0620
      BACKSPACE SCRACH                                                  SMTH0630
      READ (SCRACH) NSTX,NPU,NPV,((P(I,J),I=1,NSTX),J=1,NSTX),          SMTH0640
     1    ((CUZ(I,J),I=1,NSTX),J=1,NPU),((CVZ(I,J),I=1,NSTX),J=1,NPV)   SMTH0650
      CALL IDENT                                                        SMTH0660
   99 RETURN                                                            SMTH0670
  150 MTYP=MTP(KG)                                                      SMTH0680
      IF(KG.EQ.0) GO TO 158                                             SMTH0690
      CALL OBSERV                                                       SMTH0700
      DO 156 I=1,3                                                      SMTH0710
      IF(MR(I,MTYP)) 152,154,152                                        SMTH0720
  152 RES(I)=DFM(I)-DAT(I,KG)                                           SMTH0730
      IF(MTYP.GT.5) GO TO 153                                           SMTH0740
      IF(I.EQ.1) GO TO 153                                              SMTH0750
```

```
      IF(ABS(RES(I)).LT.3.1416) GO TO 153                               SMTH0760
      RES(I)=RES(I)-SIGN(6.283185307179586,RES(I))                       SMTH0770
  153 CONTINUE                                                          SMTH0780
      RES(I+3)= RES(I)/SP(I)                                            SMTH0790
      FLOS= FLOS + RES(I+3)**2                                          SMTH0800
      KD(I)=NCOUNT                                                      SMTH0810
      NCOUNT=NCOUNT-1                                                   SMTH0820
      GO TO 156                                                         SMTH0830
  154 RES(I)=0.                                                         SMTH0840
      RES(I+3)=0.                                                       SMTH0850
      KD(I) = 0                                                         SMTH0860
  156 CONTINUE                                                          SMTH0870
      IF(L1.NE.0) WRITE(OUT,1000)                                       SMTH0880
      L1=0                                                              SMTH0890
      WRITE(OUT,1002) (KD(I),I=1,3),MTYP,TO,(RES(I),I=1,6),FLOS         SMTH0900
      KG=KG-1                                                           SMTH0910
      IF(KG.GT.0) GO TO 36                                              SMTH0920
  158 TIME=TZERO-10.                                                    SMTH0930
      GO TO 37                                                          SMTH0940
 1000 FORMAT(//5X6HPOINTS4X,4HTYPE,3X,4HTIME,6X,4HRES1,9X,4HRES2,9X,4HRESMTH0950
     XS3,9X,8HWGT.RES1,5X,8HWGT.RES2,5X,8HWGT.RES3,5X,9HLOSS FCTN)      SMTH0960
 1002 FORMAT(1X,I4,1H,I4,1H,I4,1X,I2,1X,F10.3,7(1X,E12.5))              SMTH0970
 1004 FORMAT(///20(1H*),25HBEGIN BACKWARDS SMOOTHING,75(1H*))           SMTH0980
      END                                                               SMTH0990
```

```
      SUBROUTINE STAT
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *    NALL,LRK
      COMMON /MOD1/ ALPH,AS ,BETA,CA(6),CALP      ,CBET      ,CETA      ,
     1    CF(3),JCS ,CXZI      ,DADX(2,10),DCDY(3,4),DERIV(3,10),DRDH  ,
     2    DRDP,DWDH(2),ETA,HO,N4(3)      ,N5(3)     ,N6(3)     ,PAXP(3,30),
     3    PE        ,ROE       ,ROEC      ,ROEM      ,SALP      ,SBET      ,
     4    SETA      ,SREF      ,SXZI      ,TAU       ,TMAS(20)  ,TONE      ,
     5    TTWO      ,TUW(20)   ,TVW(20)   ,UW        ,VA(5)     ,VB(5)     ,
     6    VW                   ,XIND1     ,XIND2     ,XLREF     ,XM        ,
     7    XMAS(20)  ,XNU       ,XQ        ,XQSM      ,XRE       ,XTEMP     ,
     8    XUW(20)   ,XVW(20)   ,XZI       ,YMAS      ,YMASM
      COMMON /TWICE/ A(3,3)    ,CPH       ,CT(5)     ,SP(10)    ,ST(5)
     1    ,TPH      ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)
     2    ,B(8)     ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)
     3    ,DUF(4)   ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)
     4    ,IDN ,II  ,IP  ,IPC  ,JJ  ,JN   ,KA  ,KD(3),KDUM ,KG2
     5    ,K1  ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)
     6    ,RO       ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)     ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)
     1    ,AZ(40)   ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5)
     2    ,CVZ(30,5),D(5)      ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)
     3    ,DET      ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET
     4    ,FLOS     ,G(3,30)   ,H(3,5)    ,I ,ICOUNT,J   ,JNBR,JNBRS,JST
     5    ,K   ,KAR ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L  ,LC(3)     ,LCS(3)    ,LS   ,LT
     7    ,L1  ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT
     8    ,MTP(40)  ,MTYP      ,MTYPS     ,N ,NC(30),NCOUNT    ,NPC(15)
     9    ,NPU ,NPV ,NS(9)     ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)
     A    ,N8(3)    ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)
     B    ,PA(3)               ,PM(3)     ,R         ,REO       ,RERP2
     C    ,RPO      ,ROT(5)    ,S(5)      ,SCAPH(5) ,SI(3)      ,SIG(3,40)
     D    ,SIGM(3)  ,SIGMS(3)  ,SPHDT(5)  ,SUM ,SUM2,SYG(3,9)  ,TFINAL
     E    ,TFIT(9)  ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)
     F    ,TR(40)   ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)
     G    ,TZERO    ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)
     H    ,XYCG(20) ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
      COMMON /ATMCON /ABET,AGAM,AMO,APO,AR,ASU,ATM(4),HB(23),ISV,JK
     A  ,PB(23),RL(23),TMB(23)
      COMMON /EXTRA / GO
      EQUIVALENCE (SP(3),PHIDT),(SP(4),CAPHI)
      IF(MTYP.GT.5) RETURN
      N = 60 + 15*MTYP
      SP(1) = COS(C(N+13))
      SP(2) = SIN(C(N+13))
      PHIDT = ATAN2(RERP2*SP(2),SP(1))
      CAPHI= PHIDT - C(N+13)
      SPHDT(MTYP)=SIN(PHIDT)
      CPHDT(MTYP)=COS(PHIDT)
      CCAPH(MTYP)= COS(CAPHI)
      SCAPH(MTYP)= SIN(CAPHI)
      ROT(MTYP) = REO/SQRT(1.+(RERP2-1.)*SP(2)**2)
   99 RETURN
      END
```

```
      FUNCTION   TAB(TARG,N,T,Y)
      DIMENSION N(1),T(1),Y(1)
      IF(N(1))111,14,111
  111 CONTINUE
      I = N(2)
    6 IF(TARG - T(I))3,2,1
    1 IF(N(3))9,5,9
    5 I = I+1
      IF(I-N(1)) 6,4,4
    4 I = I - 1
    8 TAB = (Y(I+1)*(TARG - T(I)) - Y(I)*(TARG - T(I+1)))/(T(I+1)-T(I))
    7 N(2) = I
   99 RETURN
   11 I = I - 1
    2 TAB = Y(I)
      GO TO 7
    9 IF(TARG - T(I-1)) 4,11,12
    3 IF(N(3))5,10,5
   10 IF(TARG - T(I-1))12,11,4
   12 I = I - 1
      IF(I-1)18,8,6
   18 I=1
      GO TO 8
   14 TAB = 0.
      RETURN
      END
```

B. STEP2 Listing

The STEP2 source listing is presented in the following subsection. The following listing indicates the pages on which the STEP2 subroutine listings appear:

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     2 TAPE7,      TAPE1,TAPE2,TAPE3,TAPE4)
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *     NALL,LRK
      COMMON /TWICE/ A(3,3)     ,CPH        ,CT(5)      ,SP(10)     ,ST(5)
     1     ,TPH        ,VO(20)     ,ZO(20)     ,AG(3)      ,AMDOT(3) ,APDOT(4)
     2     ,B(8)       ,CGM(3)     ,DFM(3)     ,DTRAN(3,3),DUD(10,1),DUE(4)
     3     ,DUF(4)     ,F1(3,40)   ,F2(3,5)    ,F3(4,40)  ,GG(3)     ,HI(2)
     4     ,IDN ,II    ,IP   ,IPC  ,JJ   ,JN   ,KA  ,KD(3),KDUM ,KG2
     5     ,K1   ,NNN  ,NPTS       ,PAR(6)     ,PDOT(3)    ,PMDOT(3)  ,RES(6)
     6     ,RO         ,SPD(5)     ,SPH        ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)      ,AM(3)      ,AP(4)      ,AX(40)     ,AY(40)
     1     ,AZ(40)     ,C(160)     ,CCAPH(5) ,CONRD      ,CPHDT(5)  ,CUZ(30,5)
     2     ,CVZ(30,5),D(5)         ,DAT(3,40),DATA(3)     ,DATAS(3)  ,DATC(3)
     3     ,DET        ,DFIT(9)    ,DTF(9)     ,DTI(9)     ,DZ(30)     ,ET
     4     ,FLOS       ,G(3,30)    ,H(3,5)     ,I  ,ICOUNT,J     ,JNBR,JNBRS,JST
     5     ,K    ,KAR  ,KC(3)      ,KDAP       ,KDATA      ,KDATAS     ,KG,KI,KK
     6     ,KN   ,KOB  ,KPROP,KS,KSM,KSS,L ,LC(3)       ,LCS(3)     ,LS    ,LT
     7     ,L1   ,M    ,MC(5)      ,MCC(5)     ,MO(5)      ,MR(3,9)   ,MT
     8     ,MTP(40)    ,MTYP       ,MTYPS      ,N  ,NC(30),NCOUNT     ,NPC(15)
     9     ,NPU  ,NPV  ,NS(9)      ,NSS(5)     ,NST  ,NSTA,NSTC,NSTX,NT,NTR(9)
     A     ,N8(3)      ,N9(3)      ,N10(3)     ,N11(3)     ,OMEGA      ,P(30,30)
     B     ,PA(3)                  ,PM(3)      ,R          ,REO        ,RERP2
     C     ,RPO        ,ROT(5)     ,S(5)       ,SCAPH(5)  ,SI(3)      ,SIG(3,40)
     D     ,SIGM(3)    ,SIGMS(3)  ,SPHDT(5) ,SUM  ,SUM2,SYG(3,9)  ,TFINAL
     E     ,TFIT(9)    ,TIME       ,TIMES      ,TO         ,TP(40)     ,TQ(40)
     F     ,TR(40)     ,TT(40)     ,TXCG(20) ,TYCG(20)  ,TYM(40)    ,TZCG(20)
     G     ,TZERO      ,XJ2        ,XMU        ,XMUJ       ,XP(3)      ,XXCG(20)
     H     ,XYCG(20)  ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
C---------STATISTICAL TRAJECTORY ESTIMATION PROGRAM, STEP2
C                        DEVELOPED BY W.E.WAGNER AND A.C.SEROLD
C                                        MARTIN MARIETTA CORPORATION
C                                        DENVER, COLORADO
    2 CONTINUE
      CALL PRESET
    4 WRITE(OUT,1001)
      IF(NPC(1).EQ.0) WRITE(OUT,1000) ICOUNT
 1000 FORMAT(19H  ITERATION NUMBER I2//)
 1001 FORMAT(1H1)
   10 IF(KDATA) 12,42,202
   12 T2=TO
      CALL INTAG
      GO TO 44
   24 T2=TIME
      IF(T2-ET)260,260,42
   30 IF(TO-ET) 34,32,32
   32 IF(JST.EQ.1) GO TO 44
   34 IF(KDATA) 38,36,36
   36 IF(KPROP.EQ.1) GO TO 24
      IF(KDATA) 10,40,10
   38 IF(NPC(1)-1) 200,40,100
   40 TIME=TFINAL+100.
      GO TO 24
   42 T2=ET
C-------------PROPAGATION LOOP, TO EFN 44
```

```
      CALL INTAG                                                        STP20280
      KPROP=1                                                           STP20290
      TO=T2                                                             STP20300
      CALL PROP                                                         STP20310
   44 CALL OUTPUT                                                       STP20320
      L1=2                                                              STP20340
      IF(ET-TFINAL) 46,50,50                                            STP20350
   46 ET=ET+DET                                                         STP20360
      GO TO 34                                                          STP20370
   50 WRITE(SCRACH) KDATA                                               STP20380
      IF(NPC(1).EQ.1) GO TO 2                                           STP20390
   56 IF(NPC(4)) 300,58,300                                             STP20400
   58 CALL SMOOTH                                                       STP20410
      IF(L1-1) 2,4,2                                                    STP20420
C-------------DETERMINE MEASUREMENT TIME FOR ERROR                       STP20430
C                                     ANALYSIS PROBLEMS, THRU EFN 118--STP20440
  100 IF(KPROP.GT.0) GO TO 24                                           STP10432
      JST=1                                                             STP10434
      IF(NSTA-1) 40,102,104                                             STP10440
  102 II=NS(1)                                                          STP20460
      GO TO 112                                                         STP20470
  104 II = NS(1)                                                        STP20480
      SP(1) = DTI(II)                                                   STP20490
      DO 110 I=2,NSTA                                                   STP20510
      J=NS(I)                                                           STP20520
      IF(SP(1)-DTI(J)) 110,106,107                                      STP20530
  106 JST=JST+1                                                         STP20540
      GO TO 108                                                         STP20550
  107 JST=1                                                             STP20560
  108 SP(1) = DTI(J)                                                    STP20570
      II=J                                                              STP20580
  110 CONTINUE                                                          STP20590
  112 IF(DTI(II)-DTF(II))  116,116,114                                  STP20600
  114 NSTA=NSTA-1                                                       STP20610
      GO TO 100                                                         STP20620
  116 JNBR=0                                                            STP20630
      TIME=DTI(II)                                                      STP20640
      DTI(II)=DTI(II)+DFIT(II)                                          STP20650
      DO 118 I=1,3                                                      STP20660
      IF(MR(I,II).EQ.0) GO TO 118                                       STP20670
      JNBR=JNBR+1                                                       STP20680
      LC(JNBR)=I                                                        STP20690
  118 CONTINUE                                                          STP20700
      IF(JNBR.EQ.0) GO TO 100                                           STP20710
      MTYP=II                                                           STP20720
      GO TO 24                                                          STP20730
C-------------DATA EDITING LOGIC FOR FILTER PROBLEMS, THRU EFN 250------STP20740
  200 KG = 1                                                            STP20750
      KS = 1                                                            STP20760
      TYM(1) = TZERO - 10.                                              STP20770
      GO TO 211                                                         STP20780
  202 KG=KG+1                                                           STP20790
      MTP(KG)=MTYPS                                                     STP20800
      TYM(KG)=TIMES                                                     STP20810
      DAT(1,KG)=DATAS(1)                                                STP20820
      DAT(2,KG)=DATAS(2)                                                STP20830
      DAT(3,KG)=DATAS(3)                                                STP20840
      SIG(1,KG)=SIGMS(1)                                                STP20850
      SIG(2,KG)=SIGMS(2)                                                STP20860
      SIG(3,KG)=SIGMS(3)                                                STP20870
```

```
      MTYP=MTYPS
      TIME=TIMES
      DATA(1)=DATAS(1)
      DATA(2)=DATAS(2)
      DATA(3)=DATAS(3)
      SIGM(1)=SIGMS(1)
      SIGM(2)=SIGMS(2)
      SIGM(3)=SIGMS(3)
      LC(1)=LCS(1)
      LC(2)=LCS(2)
      LC(3)=LCS(3)
      JNBR=JNBRS
  204 IF(KI-KS) 206,208,208
  206 KI=KI+1
      GO TO 216
  208 KS=KG
  210 CONTINUE
  211 IF(KDATAS.EQ.0) GO TO 212
      READ(FIT) KDATAS,(MTP(KS+I),TYM(KS+I),(DAT(J,KS+I),J=1,3),(SIG(J
     X,KS+I),J=1,3),I=1,KDATAS)
      KS=KS+KDATAS
      IF(KS-20) 210,215,215
  212 IF(KG+1-KS) 215,213,213
  213 KDATA=0
      GO TO 248
  215 KI=KG+1
  216 MTYPS=MTP(KI)
      TIMES=TYM(KI)
      IF(TIMES.GT.TFINAL) GO TO 213
      IF(NTR(MTYPS)) 220,218,220
  218 IF(TIMES.LT.DTI(MTYPS)) GO TO 204
      TFIT(MTYPS)=TIMES
      NTR(MTYPS)=1
  220 CONTINUE
  222 IF(TIMES-DTF(MTYPS)) 224,224,204
  224 IF(TIMES-TFIT(MTYPS))226,234,228
  226 NTR(MTYPS)=3
      GO TO 204
  228 IF(NTR(MTYPS)-3) 230,232,230
  230 TFIT(MTYPS)=TFIT(MTYPS)+DFIT(MTYPS)
      NTR(MTYPS)=3
      GO TO 224
  232 TFIT(MTYPS)=TIMES
  234 NTR(MTYPS)=2
      JNBRS=0
      DO 238 I=1,3
      IF(MR(I,MTYPS)) 236,238,236
  236 JNBRS=JNBRS+1
      LCS(JNBRS)=I
  238 CONTINUE
      IF(JNBRS) 240,204,240
  240 DATAS(1)=DAT(1,KI)
      DATAS(2)=DAT(2,KI)
      DATAS(3)=DAT(3,KI)
      SIGMS(1)=SIG(1,KI)
      SIGMS(2)=SIG(2,KI)
      SIGMS(3)=SIG(3,KI)
      IF(KDATA) 242,244,244
  242 KDATA=2
      IF(TIMES-TZERO) 204,202,202
```

```
  244 IF(TIMES-TIME) 248,246,248
  246 JST=2
      GO TO 250
  248 JST=1
  250 GO TO 24
C------------PROCESSING LOOP, THRU EFN 278-----------------------------
  260 CALL INTAG
      T0=T2
      KPROP=0
      KN=0
      KK = 1
      IF(JNBR-1) 270,270,262
  262 IF(NPC(9)) 268,264,268
  264 KN=1
      GO TO 270
  268 KK=JNBR
      IF(NPC(4)) 264,270,264
  270 KAR=1
  272 CALL OBSERV
  273 IF(T-TSAV) 274,276,274
  274 CALL PROP
      TSAV=T
  276 IF(NPC(9).NE.0) KOB = KAR
      CALL MINVAR
      IF(KAR-KK) 278,30,30
  278 KAR=KAR+1
      IF(KN.NE.0) GO TO 276
      CALL MOTION
      GO TO 272
C------------UPDATE STATE, EQ.(55A), THRU EFN 312---------------------
  300 DO 306 I=1,10
      SUM=0.
      DO 304 K=1,10
  304 SUM=SUM+PH(I,K)*DZ(K)
  306 DUD(I)=SUM
      DO 312 I=1,NSTX
      IF(I-10) 308,308,310
  308 DZ(I)=DUD(I)
      X(I)=X(I)+DZ(I)
      GO TO 312
  310 K=NC(I-10)
      C(K)=C(K)+DZ(I)
      VO(I-10)=C(K)
  312 CONTINUE
C------------NORMALIZE QUATERNION-------------------------------------
      DATC(1)= SQRT(X(7)**2 + X(8)**2 + X(9)**2 + X(10)**2)
      X(7) = X(7)/DATC(1)
      X(8) = X(8)/DATC(1)
      X(9) = X(9)/DATC(1)
      X(10)=X(10)/DATC(1)
C------------UPDATE TRACKING STATION LOCATIONS------------------------
      IF(KSS.LE.0) GO TO 316
      DO 314 I=1,KSS
      MTYP=NSS(I)
  314 CALL STAT
  316 WRITE(OUT,1003)
 1003 FORMAT(39H UPDATED MODEL PARAMETERS AT FINAL TIME)
      WRITE(OUT,1002) (NC(I),VO(I),I=1,NSTC)
 1002 FORMAT(6(2H C,I3,E15.8))
      GO TO 58
      END
```

```
      SUBROUTINE DATAB
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),       COMM0010
     *    NALL,LRK                                                          COMM0020
      COMMON /TWICE/ A(3,3)     ,CPH      ,CT(5)    ,SP(10)   ,ST(5)      COMM0030
     1     ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)    COMM0040
     2     ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)     COMM0050
     3     ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)       COMM0060
     4     ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2              COMM0070
     5     ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)      COMM0080
     6     ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)                COMM0090
      DIMENSION DUB(30,15)                                                 COMM0100
      EQUIVALENCE (AG(1),DUB(1,1))                                         COMM0110
      COMMON /METH2/ AB(3)      ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)     COMM0120
     1     ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)   COMM0130
     2     ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)     COMM0140
     3     ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET          COMM0150
     4     ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J    ,JNBR,JNBRS,JST  COMM0160
     5     ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK    COMM0170
     6     ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT    COMM0180
     7     ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT          COMM0190
     8     ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)     COMM0200
     9     ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)   COMM0210
     A     ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)    COMM0220
     B     ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2       COMM0230
     C     ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)   COMM0240
     D     ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL      COMM0250
     E     ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)      COMM0260
     F     ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)    COMM0270
     G     ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)    COMM0280
     H     ,XYCG(20) ,XZCG(20)                                            COMM0290
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                     COMM0300
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                          COMM0310
      IF(DCOMP.LT.0.) GO TO 20                                             DATB0010
      IF(KPROP.GE.0 ) GO TO 10                                             DATB0020
    2 CONTINUE                                                             DATB0030
    4 LS = 0                                                               DATB0040
    6 IF(KDAP.EQ.0)   GO TO 14                                             DATB0050
      READ(PQR) KDAP,(TT(LS+I),TP(LS+I),TQ(LS+I),TR(LS+I),AX(LS+I),        DATB0060
     *    AY(LS+I),AZ(LS+I),I=1,KDAP)                                      DATB0070
      LS = LS+KDAP                                                         DATB0080
      IF(LS.EQ.1) GO TO 6                                                  DATB0090
      IF(T.GT.TT(LS)) GO TO 11                                             DATB0100
      IF(LS.LE.20)    GO TO 6                                              DATB0110
      N11(2)=3                                                             DATB0120
   10 IF(T.LT.TT(LS-1)) GO TO 14                                           DATB0130
      IF(KDAP.EQ.0) GO TO 14                                               DATB0140
      KG2 = KG-2                                                           DATB0150
      WRITE(SCRACH) LS,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),         DATB0160
     *    I=1,LS),KG2,(MTP(I),TYM(I),DAT(1,I),DAT(2,I),DAT(3,I),           DATB0170
     *    SIG(1,I),SIG(2,I),SIG(3,I),I=1,KG2)                              DATB0180
      IF(KG2.LE.0) GO TO 11                                                DATB0190
      KG = 2                                                               DATB0200
      DO 13 I=1,2                                                          DATB0210
      LT = KG2 + I                                                         DATB0220
      MTP(I)= MTP(LT)                                                      DATB0230
      TYM(I)= TYM(LT)                                                      DATB0240
      DAT(1,I)= DAT(1,LT)                                                  DATB0250
      DAT(2,I) = DAT(2,LT)                                                 DATB0260
```

```
      DAT(3,I) = DAT(3,LT)                                              DATB0270
      SIG(1,I) = SIG(1,LT)                                              DATB0280
      SIG(2,I) = SIG(2,LT)                                              DATB0290
      SIG(3,I) = SIG(3,LT)                                              DATB0300
   13 CONTINUE                                                          DATB0310
   11 CONTINUE                                                          DATB0320
      DO 12 I=1,2                                                       DATB0330
      MT = LS  + I - 2                                                  DATB0340
      TT(I) = TT(MT)                                                    DATB0350
      TP(I) = TP(MT)                                                    DATB0360
      TQ(I) = TQ(MT)                                                    DATB0370
      TR(I) = TR(MT)                                                    DATB0380
      AX(I) = AX(MT)                                                    DATB0390
      AY(I) = AY(MT)                                                    DATB0400
      AZ(I) = AZ(MT)                                                    DATB0410
   12 CONTINUE                                                          DATB0420
      LS = 2                                                            DATB0430
      GO TO 6                                                           DATB0440
   14 N11(1)  = LS                                                      DATB0450
      IF(KPROP.LT.0) TRCD=TT(1)                                         DATB0460
      KPROP = 1                                                         DATB0470
      PM(1) = TAB(T,N11(1),TT(1),TP(1))                                 DATB0480
      PM(2)=TAB(T,N11(1),TT(1),TQ(1))                                   DATB0490
      PM(3)=TAB(T,N11(1),TT(1),TR(1))                                   DATB0500
      AM(1)= TAB(T,N11(1),TT(1),AX(1))                                  DATB0510
      AM(2)= TAB(T,N11(1),TT(1),AY(1))                                  DATB0520
      AM(3)= TAB(T,N11(1),TT(1),AZ(1))                                  DATB0530
      CGM(1) = TAB(T,N8(1)  ,TXCG(1),XXCG(1))                           DATB0540
      CGM(2) = TAB(T,N9(1)  ,TYCG(1),XYCG(1))                           DATB0550
      CGM(3) = TAB(T,N10(1),TZCG(1),XZCG(1))                            DATB0560
   99 RETURN                                                            DATB0570
   20 IF(T.GT.TT(2)) GO TO 14                                           DATB0580
      IF(TT(1).EQ.TRCD) GO TO 14                                        DATB0590
      BACKSPACE SCRACH                                                  DATB0600
      BACKSPACE SCRACH                                                  DATB0610
      LT = 40-KG                                                        DATB0620
      DO 22 I=1,KG                                                      DATB0630
      MT = LT+I                                                         DATB0640
      MTP(MT) = MTP(I)                                                  DATB0650
      TYM(MT) = TYM(I)                                                  DATB0660
      DAT(1,MT) = DAT(1,I)                                              DATB0670
      DAT(2,MT) = DAT(2,I)                                              DATB0680
      DAT(3,MT) = DAT(3,I)                                              DATB0690
      SIG(1,MT) = SIG(1,I)                                              DATB0700
      SIG(2,MT) = SIG(2,I)                                              DATB0710
   22 SIG(3,MT) = SIG(3,I)                                              DATB0720
      READ(SCRACH) KDAP,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),     DATB0730
     *    I=1,KDAP),KDATA,(MTP(I),TYM(I),DAT(1,I),DAT(2,I),DAT(3,I),    DATB0740
     *    SIG(1,I),SIG(2,I),SIG(3,I),I=1,KDATA)                         DATB0750
      LS=KDAP                                                           DATB0760
      N11(2)=LS-2                                                       DATB0770
      IF(KDATA.LT.0) GO TO 26                                           DAT20780
      LT = LT - KDATA                                                   DATB0790
      I = KDATA+1                                                       DATB0800
      J = KDATA+KG                                                      DATB0810
      DO 24 I=I,J                                                       DATB0820
      MT = LT + I                                                       DATB0830
      MTP(I) = MTP(MT)                                                  DATB0840
      TYM(I) = TYM(MT)                                                  DATB0850
      DAT(1,I) = DAT(1,MT)                                              DATB0860
```

```
      DAT(2,I) = DAT(2,MT)                                              DATB0870
      DAT(3,I) = DAT(3,MT)                                              DATB0880
      SIG(1,I) = SIG(1,MT)                                              DATB0890
      SIG(2,I) = SIG(2,MT)                                              DATB0900
   24 SIG(3,I) = SIG(3,MT)                                              DATB0910
      KG = KG + KDATA                                                   DATB0920
      GO TO 14                                                          DATB0930
   26 KG = 1                                                            DATB0940
      GO TO 14                                                          DATB0950
      END                                                               DATB0960
```

```
      SUBROUTINE FXXU
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *    NALL,LRK
      COMMON /TWICE/ A(3,3)     ,CPH       ,CT(5)     ,SP(10)    ,ST(5)
     1     ,TPH       ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)
     2     ,B(8)      ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)
     3     ,DUF(4)    ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)
     4     ,IDN ,II   ,IP   ,IPC ,JJ   ,JN  ,KA  ,KD(3),KDUM ,KG2
     5     ,K1   ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)
     6     ,RO        ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)      ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)
     1     ,AZ(40)    ,C(160)    ,CCAPH(5) ,CONRD     ,CPHDT(5)  ,CUZ(30,5)
     2     ,CVZ(30,5),D(5)       ,DAT(3,40),DATA(3)   ,DATAS(3)  ,DATC(3)
     3     ,DET       ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET
     4     ,FLOS      ,G(3,30)   ,H(3,5)    ,I  ,ICOUNT,J     ,JNBR,JNBRS,JST
     5     ,K    ,KAR ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK
     6     ,KN   ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)     ,LCS(3)    ,LS    ,LT
     7     ,L1   ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT
     8     ,MTP(40)   ,MTYP      ,MTYPS     ,N  ,NC(30),NCOUNT    ,NPC(15)
     9     ,NPU  ,NPV ,NS(9)     ,NSS(5)    ,NST  ,NSTA,NSTC,NSTX,NT,NTR(9)
     A     ,N8(3)     ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)
     B     ,PA(3)                ,PM(3)     ,R         ,REO       ,RERP2
     C     ,RPO       ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40)
     D     ,SIGM(3)   ,SIGMS(3)  ,SPHDT(5)  ,SUM  ,SUM2,SYG(3,9)  ,TFINAL
     E     ,TFIT(9)   ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)
     F     ,TR(40)    ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)
     G     ,TZERO     ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)
     H     ,XYCG(20)  ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
      DIMENSION CX(3,4)
      EQUIVALENCE (C(48),CX(1,1))
C-------------CALCULATE EQS. IN TABLE 4 -------------------
      SP(1) = PAR(4)*X(2)/CPH**2
      F2(2,1) = 1./R
      F1(1,3) = F2(2,1)*X(1)
      F1(3,1) =-F1(1,3)-F1(1,3)
      F1(3,2) =-F1(2,3)-F1(2,3)
      F1(1,1) = F2(2,1)*X(3)
      F1(1,2) =-F1(2,1)-F1(2,1)
      F1(1,4) =(-PAR(5)*F2(2,1)+4.*PAR(3)*SPH*CPH)*F2(2,1)
      F1(1,5) = SP(1)*F1(3,2)*R - PAR(3)*(CPH+SPH)*(CPH-SPH)
      F1(2,4) =-F1(2,2)*F1(2,3)
      F1(2,5) =-SP(1)*F1(3,1)*R
      F1(3,4) = (PAR(1)*F2(2,1)-PAR(2)-PAR(2) + 4.*PAR(3)*PAR(6))*
     *          F2(2,1)
      F1(3,5) =-3.*PAR(3)*SPH*CPH
      F2(2,4) =-F1(1,3)*F2(2,1)
      F2(3,2) = F2(2,1)/CPH
      F2(3,4) =-F1(2,3)*F2(3,2)
      F2(3,5) = F1(2,1)/CPH
      F3(1,1) =-X( 9)*PAR(4)
      F3(2,1) = X(10)*PAR(4)
      F3(3,1) = X( 7)*PAR(4)
      F3(4,1) =-X( 8)*PAR(4)
      F3(1,2) = (X( 8)-X(10)*TPH)*PAR(4)
      F3(2,2) =-(X( 7)+X( 9)*TPH)*PAR(4)
```

```
      F3(3,2) = (X(10)+X( 8)*TPH)*PAR(4)                                FXXU0280
      F3(4,2) =-(X( 9)-X( 7)*TPH)*PAR(4)                                FXXU0290
      F3(1,4) =-F1(1,3)*F3(1,1) - F1(2,3)*F3(1,2)                       FXXU0300
      F3(2,4) =-F1(1,3)*F3(2,1) - F1(2,3)*F3(2,2)                       FXXU0310
      F3(3,4) =-F1(1,3)*F3(3,1) - F1(2,3)*F3(3,2)                       FXXU0320
      F3(4,4) =-F1(1,3)*F3(4,1) - F1(2,3)*F3(4,2)                       FXXU0330
      F3(1,5) =-X(10)*SP(1)                                             FXXU0340
      F3(2,5) =-X( 9)*SP(1)                                             FXXU0350
      F3(3,5) = X( 8)*SP(1)                                             FXXU0360
      F3(4,5) = X( 7)*SP(1)                                             FXXU0370
      F1(1,7) = 2.*(X( 7)*AB(1)-X(10)*AB(2)+X( 9)*AB(3))                FXXU0380
      F1(1,8) = 2.*(X( 8)*AB(1)+X( 9)*AB(2)+X(10)*AB(3))                FXXU0390
      F1(1,9) = 2.*(-X(9)*AB(1)+X( 8)*AB(2)+X( 7)*AB(3))                FXXU0400
      F1(1,10)=-2.*(X(10)*AB(1)+X( 7)*AB(2)-X( 8)*AB(3))                FXXU0410
      F1(2,7) =-F1(1,10)                                                FXXU0420
      F1(2,8) =-F1(1, 9)                                                FXXU0430
      F1(2,9) = F1(1, 8)                                                FXXU0440
      F1(2,10)= F1(1, 7)                                                FXXU0450
      F1(3,7) = F1(1, 9)                                                FXXU0460
      F1(3,8) =-F1(1,10)                                                FXXU0470
      F1(3,9) =-F1(1, 7)                                                FXXU0480
      F1(3,10)= F1(1, 8)                                                FXXU0490
      F3(2,7) =-F3(1,8)                                                 FXXU0500
      F3(3,7) =-F3(1, 9)                                                FXXU0510
      F3(3,8) =-F3(2, 9)                                                FXXU0520
      F3(4,7) =-F3(1,10)                                                FXXU0530
      F3(4,8) =-F3(2,10)                                                FXXU0540
      F3(4,9) =-F3(3,10)                                                FXXU0550
      F2(1,3) = -1.                                                     FXXU0560
      F1(1,6) = 0.                                                      FXXU0570
      F1(2,6) = 0.                                                      FXXU0580
      F1(3,6) = 0.                                                      FXXU0590
      F1(3,3) = 0.                                                      FXXU0600
      F2(1,1) = 0.                                                      FXXU0610
      F2(1,2) = 0.                                                      FXXU0620
      F2(1,4) = 0.                                                      FXXU0630
      F2(1,5) = 0.                                                      FXXU0640
      F2(2,2) = 0.                                                      FXXU0650
      F2(2,3) = 0.                                                      FXXU0660
      F2(2,5) = 0.                                                      FXXU0670
      F2(3,1) = 0.                                                      FXXU0680
      F2(3,3) = 0.                                                      FXXU0690
      F3(1,3) = 0.                                                      FXXU0700
      F3(2,3) = 0.                                                      FXXU0710
      F3(3,3) = 0.                                                      FXXU0720
      F3(4,3) = 0.                                                      FXXU0730
      F3(1,6) = 0.                                                      FXXU0740
      F3(2,6) = 0.                                                      FXXU0750
      F3(3,6) = 0.                                                      FXXU0760
      F3(4,6) = 0.                                                      FXXU0770
      F3(1,7)=0.                                                        FXXU0780
      F3(2,8)=0.                                                        FXXU0790
      F3(3,9)=0.                                                        FXXU0800
      F3(4,10)=0.                                                       FXXU0810
      LT= 0                                                             FXXU0820
      MT=0                                                              FXXU0830
      NT = 0                                                            FXXU0840
      IF(NSTC) 1,1,6                                                    FXXU0850
    1 IF(NPU)  2,99,2                                                   FXXU0860
    2 N = NSTX                                                          FXXU0870
```

```
      NT = 1
      II = 1
    3 NNN = MC(II)
      GO TO 18
    6 N = 10
      II = 1
    9 NNN = NC(II)
      GO TO 18
   12 II = II+1
      IF(NT) 13,13,15
   13 IF(II-NSTC) 9,9,1
   15 IF(II-NPU ) 3,3,99
   18 CONTINUE
      IF(NNN.GT.72) GO TO 25
   19 IF(NNN-33) 100,100,20
   20 IF(MT) 21,24,21
   21 CONTINUE
      SP(1) = 0.
      SP(2) = 0.
      SP(3) = 0.
      IF(NNN-59) 150,150,300
   25 DO 26 I=1,3
      F1(I,II+N) = 0.
      F3(I,II+N) = 0.
   26 CONTINUE
      F3(4,II+N) = 0.
      GO TO 12
   24 CONTINUE
      MT=1
      K = 0
      DO 27 I=1,3
      DO 27 J=1,3
      K = K+1
   27 SP(K) = PA(I)*XP(J)
      DTRAN(1,1) =      SP(5) +      SP(9)
      DTRAN(1,2) = -2.*SP(4) +      SP(2)
      DTRAN(1,3) = -2.*SP(7) +      SP(3)
      DTRAN(2,1) =      SP(4)   -2.*SP(2)
      DTRAN(2,2) =      SP(1) +      SP(9)
      DTRAN(2,3) = -2.*SP(8) +      SP(6)
      DTRAN(3,1) =      SP(7)   -2.*SP(3)
      DTRAN(3,2) =      SP(8) -  2.*SP(6)
      DTRAN(3,3) =      SP(1) +      SP(5)
      GO TO 21
   30 CONTINUE
      SP(1) = SP(1)-XP(3)*DUD(2)+XP(2)*DUD(3)
      SP(2) = SP(2)+XP(3)*DUD(1)-XP(1)*DUD(3)
      SP(3) = SP(3)-XP(2)*DUD(1)+XP(1)*DUD(2)
   33 CONTINUE
C-------------CALCULATE EQ.(172) -------------------------
      SP(1) = SP(1)-DTRAN(1,1)*SP(4)-DTRAN(1,2)*SP(5)-DTRAN(1,3)*SP(6)
      SP(2) = SP(2)-DTRAN(2,1)*SP(4)-DTRAN(2,2)*SP(5)-DTRAN(2,3)*SP(6)
      SP(3) = SP(3)-DTRAN(3,1)*SP(4)-DTRAN(3,2)*SP(5)-DTRAN(3,3)*SP(6)
   36 I = N + II
      IF(L) 39,42,39
C-------------CALCULATE EQ.(169) -------------------------
   39 F3(1,I) = .5*(-X( 8)*SP(4)-X( 9)*SP(5)-X(10)*SP(6))
      F3(2,I) = .5*( X( 7)*SP(4)-X(10)*SP(5)+X( 9)*SP(6))
      F3(3,I) = .5*( X(10)*SP(4)+X( 7)*SP(5)-X( 8)*SP(6))
      F3(4,I) = .5*(-X( 9)*SP(4)+X( 8)*SP(5)+X( 7)*SP(6))
```

```
      GO TO 45                                                          FXXU1480
   42 F3(1,I) = 0.                                                      FXXU1490
      F3(2,I) = 0.                                                      FXXU1500
      F3(3,I) = 0.                                                      FXXU1510
      F3(4,I) = 0.                                                      FXXU1520
C------------CALCULATE EQ.(167) -------------------------               FXXU1530
   45 F1(1,I) = SP(1)*A(1,1) + SP(2)*A(2,1) + SP(3)*A(3,1)              FXXU1540
      F1(2,I) = SP(1)*A(1,2) + SP(2)*A(2,2) + SP(3)*A(3,2)              FXXU1550
      F1(3,I) = SP(1)*A(1,3) + SP(2)*A(2,3) + SP(3)*A(3,3)              FXXU1560
      GO TO 12                                                          FXXU1570
  100 CONTINUE                                                          FXXU1580
C------------CALCULATE EQ.(170) -------------------------               FXXU1590
      L = 0                                                             FXXU1600
      I = NNN - 30                                                      FXXU1610
      SP(1) = -TRAN(1,I)                                                FXXU1620
      SP(2) = -TRAN(2,I)                                                FXXU1630
      SP(3) = -TRAN(3,I)                                                FXXU1640
      GO TO 36                                                          FXXU1650
C------------CALCULATE EQ.(173) -------------------------               FXXU1660
C------------CALCULATE EQ.(174) -------------------------               FXXU1670
  150 L = 1                                                             FXXU1680
      DUD(1) = 0.                                                       FXXU1690
      DUD(2) = 0.                                                       FXXU1700
      DUD(3) = 0.                                                       FXXU1710
      SP(4)  = 0.                                                       FXXU1720
      SP(5)  = 0.                                                       FXXU1730
      SP(6)  = 0.                                                       FXXU1740
      IF(NNN.GE.48) GO TO 250                                           FXXU1750
      IF(NNN.GE.45) GO TO 200                                           FXXU1760
      J = (NNN-33)/3                                                    FXXU1770
      K = NNN - 3*J - 32                                                FXXU1780
      DUD(J) = PMDOT(K)                                                 FXXU1790
      SP(J+3) = PM(K)                                                   FXXU1800
      GO TO 30                                                          FXXU1810
  200 CONTINUE                                                          FXXU1820
      SP(NNN-41) = 1.                                                   FXXU1830
      GO TO 33                                                          FXXU1840
  250 J = (NNN-45)/3                                                    FXXU1850
      K = NNN - 3*J - 41                                                FXXU1860
      SP(K) = AP(J)                                                     FXXU1870
      DUD(K-3) = APDOT(J)                                               FXXU1880
      GO TO 30                                                          FXXU1890
C------------CALCULATE EQ.(176) -------------------------               FXXU1900
  300 CONTINUE                                                          FXXU1910
C------------CALCULATE EQ.(175) -------------------------               FXXU1920
      DUD(5) = 0.                                                       FXXU1930
      DUD(6) = 0.                                                       FXXU1940
      DUD(7) = 0.                                                       FXXU1950
      J = (NNN-58)/3                                                    FXXU1960
      K = NNN - 3*J - 57                                                FXXU1970
      IF(J-3) 301,301,302                                               FXXU1980
  301 SP(J) = AM(K)                                                     FXXU1990
      DUD(J+4) = AMDOT(K)                                               FXXU2000
      GO TO 304                                                         FXXU2010
  302 SP(K) = 1.                                                        FXXU2020
      J = K                                                             FXXU2030
  304 CONTINUE                                                          FXXU2040
      IF(J-2) 306,308,310                                               FXXU2050
  306 CONTINUE                                                          FXXU2060
      I = 3                                                             FXXU2070
```

```
      GO TO 312
  308 DUD(4) = 0.
      DUD(8) = 0.
      GO TO 316
  310 CONTINUE
      I = 1
  312 CONTINUE
      SP(5) = AP(I)*DUD(J+4) + APDOT(I)*SP(J)
      SP(6) = AP(J)*SP(J)
      SP(7) = AP(J)*DUD(J+4) + APDOT(J)*SP(J)
      DUD(4) = (AP(4)/AP(J))**3*SP(J)*.25
      IF(LT) 314,313,314
  313 CONTINUE
      LT=1
      SP(9)  = AP(1)**2 + AP(3)**2
      SPD(5) = SQRT(SP(9))
      SPD(1)= 2./SPD(5)
      SPD(2) =-(AP(1)*APDOT(3)+AP(3)*APDOT(1))*SPD(1)/SP(9)
      SPD(3)=-AP(4)/SP(9)
      SPD(4) =-(AP(1)*APDOT(1)+AP(3)*APDOT(3))/SP(9)
      SPD(5) = 2.*SPD(4)*SPD(3)
C-------------CALCULATE EQ.(181) ---------------------------
  314 DUD(8) = SPD(1)*SP(5) + (SPD(2)+SPD(5))*SP(6) + SPD(3)*SP(7) +
     *    SPD(4)*DUD(4)
  316 CONTINUE
      DO 320 I=1,3
      SP(7) = CX(I,4)*DUD(4)
      SP(8) = CX(I,4)*DUD(8)
      DO 318 J=1,3
      SP(7) = SP(7)+CX(I,J)*SP(J)
  318 SP(8) = SP(8)+CX(I,J)*DUD(J+4)
      SP(I+3) = SP(7)
  320 DUD(I)  = SP(8)
      L = 1
      GO TO 30
   99 RETURN
      END
```

```
      SUBROUTINE INDAT
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),      COMM0010
     *    NALL,LRK                                                       COMM0020
      COMMON /TWICE/ A(3,3)    ,CPH      ,CT(5)    ,SP(10)   ,ST(5)      COMM0030
     1    ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)    COMM0040
     2    ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)     COMM0050
     3    ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)       COMM0060
     4    ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2             COMM0070
     5    ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)      COMM0080
     6    ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)                COMM0090
      DIMENSION DUB(30,15)                                               COMM0100
      EQUIVALENCE (AG(1),DUB(1,1))                                       COMM0110
      COMMON /METH2/ AB(3)     ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)     COMM0120
     1    ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)   COMM0130
     2    ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)     COMM0140
     3    ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET          COMM0150
     4    ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J    ,JNBR,JNBRS,JST  COMM0160
     5    ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK    COMM0170
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT    COMM0180
     7    ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT          COMM0190
     8    ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)     COMM0200
     9    ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)   COMM0210
     A    ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)    COMM0220
     B    ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2       COMM0230
     C    ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)   COMM0240
     D    ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL      COMM0250
     E    ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)      COMM0260
     F    ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)    COMM0270
     G    ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)    COMM0280
     H    ,XYCG(20) ,XZCG(20)                                            COMM0290
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                   COMM0300
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                        COMM0310
      DATA CONV /.3048/                                                  INDT0010
      REWIND FIT                                                         INDT0020
      REWIND PQR                                                         INDT0030
      REWIND SCRACH                                                      INDT0040
      REWIND STATE                                                       INDT0050
      WRITE(OUT,1030)                                                    INDT0060
 1030 FORMAT(1H1,22H DATA ANALYSIS PROGRAM//11H INPUT DATA//)            INDT0070
    8 READ(IN,1000) KDUM,(B(I),I=1,8)                                    INDT0080
 1000 FORMAT(I2,7A10,A8)                                                 INDT0090
      WRITE(OUT,1001) KDUM,(B(I),I=1,8)                                  INDT0100
 1001 FORMAT(5X,I2,5X,7A10,5X,A8)                                        INDT0110
      IF(KDUM) 7,8,9                                                     INDT0120
    7 CALL EXIT                                                          INDT0130
    9 GO TO ( 8,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,INDT0140
     X180,190,200),KDUM                                                  INDT0150
   20 DECODE(40,1002,B) (NPC(I),I=1,10)                                  INDT0160
 1002 FORMAT(10I4)                                                       INDT0170
      IF(NPC(2).EQ.0) GO TO 8                                            INDT0180
      XMU = XMU/CONV**3                                                  INDT0190
      REO = REO/CONV                                                     INDT0200
      RPO = RPO/CONV                                                     INDT0210
      GO TO 8                                                            INDT0220
   30 DECODE(30,1003,B) IPC,IDN,SP(1),SP(2)                              INDT0230
 1003 FORMAT(I2,I4,E12.4,E12.4)                                          INDT0240
      GO TO (32,34,36,38,39),IPC                                         INDT0250
   32 NST=NST+1                                                          INDT0260
      ZO(IDN)=SP(1)                                                      INDT0270
```

```
      P(IDN,IDN)=SP(2)**2
      GO TO 8
   34 NSTC = NSTC + 1
      NC(NSTC) = IDN
      C(IDN)=SP(1)
      P(NSTC+10,NSTC+10) = SP(2)**2
      GO TO 8
   36 NPU=NPU+1
      MC(NPU)=IDN
      C(IDN)=SP(1)
      D(NPU)=SP(2)**2
      GO TO 8
   38 NPV=NPV+1
      MCC(NPV)=IDN
      C(IDN)=SP(1)
      S(NPV)=SP(2)**2
      GO TO 8
   39 C(IDN) = SP(1)
      GO TO 8
   40 DECODE(54,1004,B) TO,DET,DCOMP,TFINAL
 1004 FORMAT(6X,4E12.4)
      GO TO 8
   50 DECODE(42,1005,B) IPC,SP(1),SP(2),SP(3)
 1005 FORMAT(I2,4X,3E12.4)
      IF(IPC.EQ.2) GO TO 54
   52 XMU=SP(2)
      XJ2 = SP(3)
      GO TO 8
   54 RPO = SP(1)
      REO = SP(2)
      OMEGA=SP(3)
      GO TO 8
   60 DECODE(6,1006,B) IPC,NPTS
 1006 FORMAT(I2,I4)
      GO TO(62,64,66),IPC
   62 READ(IN,1031)  (TXCG(I),XXCG(I),I=1,NPTS)
      WRITE(OUT,1031)(TXCG(I),XXCG(I),I=1,NPTS)
 1031 FORMAT(6E12.4)
      N8(1)=NPTS
      GO TO 8
   64 READ(IN,1031)  (TYCG(I),XYCG(I),I=1,NPTS)
      WRITE(OUT,1031)(TYCG(I),XYCG(I),I=1,NPTS)
      N9(1)=NPTS
      GO TO 8
   66 READ(IN,1031)  (TZCG(I),XZCG(I),I=1,NPTS)
      WRITE(OUT,1031)(TZCG(I),XZCG(I),I=1,NPTS)
      N10(1)=NPTS
      GO TO 8
   70 DECODE(68,1007,B) IDN,SP(1),SP(2),SP(3),LT,MT,NT,SP(4),SP(5),SP(6)
 1007 FORMAT(I2,3E10.4,3I2,3E10.4)
      NSTA=NSTA+1
      NS(NSTA) = IDN
      DTI(NSTA)=SP(1)
      DTF(NSTA)=SP(2)
      DFIT(NSTA)=SP(3)
      MR(1,NSTA)=LT
      MR(2,NSTA)=MT
      MR(3,NSTA)=NT
      SYG(1,NSTA)=SP(4)
      SYG(2,NSTA)=SP(5)
```

```
      SYG(3,NSTA)=SP(6)
      GO TO 8
   80 DECODE(6,1006,B) IPC,IDN
      IF(IDN) 81,83,81
   81 READ(IN,1032) KDATA
      IF(KDATA.LE.0) GO TO 82
      READ(IN,1032) (K,MTP(I),TYM(I), (DAT(J,I),J=1,3),(SIG(J,I),J=1,3),
     *    I=1,KDATA)
      WRITE(FIT)    KDATA,(MTP(I),TYM(I),(DAT(J,I),J=1,3),(SIG(J,I),J=1,
     X3),I=1,KDATA)
      GO TO 81
 1032 FORMAT(2I2,7E10.2)
   82 CONTINUE
      K = 0
      WRITE(FIT)KDATA,K,K,K,K,K,K,K,K
      REWIND FIT
   83 IF(IPC) 84,8,84
   84 READ(FIT)     KDATA,(MTP(I),TYM(I),(DAT(J,I),J=1,3),(SIG(J,I),J=1,
     X3),I=1,KDATA)
      WRITE(OUT,1033)KDATA,(MTP(I),TYM(I),(DAT(J,I),J=1,3),(SIG(J,I),J=1
     X,3),I=1,KDATA)
 1033 FORMAT(2I4,7E16.7/(4X,I4,7E16.7))
      IF(KDATA) 84,85,84
   85 REWIND FIT
      GO TO 8
   90 DECODE(6,1006,B) IPC,IDN
      IF(IDN) 91,93,91
   91 READ(IN,1034) KDAP
      IF(KDAP.LE.0) GO TO 92
      READ(IN,1034) (K,TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),I=1,
     *    KDAP)
      WRITE(PQR)KDAP,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),I=1,
     *    KDAP)
      GO TO 91
 1034 FORMAT(I2,7E10.2)
   92 CONTINUE
      K = 0
      WRITE(PQR) KDAP,K,K,K,K,K,K,K
      REWIND PQR
   93 IF(IPC) 94,8,94
   94 READ(PQR) KDAP,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),I=1,KDAP
     X)
      WRITE(OUT,1035) KDAP,(TT(I),TP(I),TQ(I),TR(I),AX(I),AY(I),AZ(I),I=
     X1,KDAP)
 1035 FORMAT(I4,7E16.7/(4X,7E16.7))
      IF(KDAP) 94,95,94
   95 REWIND PQR
      GO TO 8
  100 DECODE(6,1006,B) IPC,NPTS
      NSTX = NST+NSTC
      GO TO (101,102,103,104,106,108),IPC
  101 READ(IN,1031)    ((P(I,J),J=1,NSTX),I=1,NSTX)
      WRITE(OUT,1031) ((P(I,J),J=1,NSTX),I=1,NSTX)
      GO TO 8
  102 READ(IN,1031)    ((CUZ(I,J),J=1,NPU),I=1,NSTX)
      WRITE(OUT,1031) ((CUZ(I,J),J=1,NPU),I=1,NSTX)
      GO TO 8
  103 READ(IN,1031)    ((CVZ(I,J),J=1,NPV),I=1,NSTX)
      WRITE(OUT,1031) ((CVZ(I,J),J=1,NPV),I=1,NSTX)
      GO TO 8
```

```
  104 DO 105 I=1,NPTS
      READ(IN,1036)  LT,MT,P(LT,MT)
      P(MT,LT)=P(LT,MT)
 1036 FORMAT(4X,2I4,E12.4)
  105 WRITE(OUT,1037) LT,MT,P(LT,MT)
 1037 FORMAT(10X,2HP(,I3,1H,I3,3H) =E15.8)
      GO TO 8
  106 DO 107 I=1,NPTS
      READ(IN,1036) LT,MT,CUZ(LT,MT)
  107 WRITE(OUT,1038) LT,MT,CUZ(LT,MT)
 1038 FORMAT(8X,4HCUZ(,I3,1H,I3,3H) =E15.8)
      GO TO 8
  108 DO 109 I=1,NPTS
      READ(IN,1036) LT,MT,CVZ(LT,MT)
  109 WRITE(OUT,1039) LT,MT,CVZ(LT,MT)
 1039 FORMAT(8X,4HCVZ(,I3,1H,I3,3H) =E15.8)
      GO TO 8
  110 GO TO 8
  120 GO TO 8
  130 GO TO 8
  140 GO TO 8
  150 GO TO 8
  160 GO TO 8
  170 GO TO 8
  180 GO TO 8
  190 GO TO 8
  200 RETURN
      END
```

```
      SUBROUTINE INTAG
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),      COMM0010
     *     NALL,LRK                                                      COMM0020
      COMMON /TWICE/ A(3,3)    ,CPH      ,CT(5)    ,SP(10)   ,ST(5)      COMM0030
     1     ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)   COMM0040
     2     ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)    COMM0050
     3     ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)      COMM0060
     4     ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2            COMM0070
     5     ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)     COMM0080
     6     ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)               COMM0090
      DIMENSION DUB(30,15)                                               COMM0100
      EQUIVALENCE (AG(1),DUB(1,1))                                       COMM0110
      COMMON /METH2/ AB(3)     ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)     COMM0120
     1     ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)  COMM0130
     2     ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)    COMM0140
     3     ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET         COMM0150
     4     ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J   ,JNBR,JNBRS,JST  COMM0160
     5     ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK   COMM0170
     6     ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT   COMM0180
     7     ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT         COMM0190
     8     ,MTP(40)  ,MTYP     ,MTYPS    ,N  ,NC(30),NCOUNT  ,NPC(15)    COMM0200
     9     ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)  COMM0210
     A     ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)   COMM0220
     B     ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2      COMM0230
     C     ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)  COMM0240
     D     ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL     COMM0250
     E     ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)     COMM0260
     F     ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)   COMM0270
     G     ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)   COMM0280
     H     ,XYCG(20) ,XZCG(20)                                           COMM0290
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                   COMM0300
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                        COMM0310
    2 LRK = 4                                                            INTG0010
    6 CALL RKUTTA                                                        INTG0020
      IF(LRK-2) 8,10,99                                                  INTG0030
    8 CALL DATAB                                                         INTG0040
      CALL MOTION                                                        INTG0050
      IF(NPC(1)-1) 12,6,12                                               INTG0060
   10 IF(DCOMP.GE.0..OR.NPC(10).EQ.0.OR.T.EQ.TSAV) GO TO 6               INTG0070
      WRITE(STATE) T,X                                                   INTG0090
      TSAV=T                                                             INTG0095
      GO TO 6                                                            INTG0100
   12 CALL FXXU                                                          INTG0110
      DO 22 I=1,4                                                        INTG0120
      N = NSTX+NPU                                                       INTG0130
      DO 22 J=1,N                                                        INTG0140
      IF(J-10) 13,13,14                                                  INTG0150
   13 CONTINUE                                                           INTG0160
      SUM = 0.                                                           INTG0170
      SUM2= 0.                                                           INTG0180
      GO TO 15                                                           INTG0190
   14 SUM = F1(I,J)                                                      INTG0200
      SUM2= F3(I,J)                                                      INTG0210
   15 DO 16 K=1,10                                                       INTG0220
   16 SUM2 = SUM2 + F3(I,K)*PH(K,J)                                      INTG0230
      DPH(I+6,J) = SUM2                                                  INTG0240
      IF(I.EQ.4) GO TO 22                                                INTG0250
      DO 18 K=1,10                                                       INTG0260
   18 SUM = SUM + F1(I,K)*PH(K,J)                                        INTG0270
```

```
      DPH(I,J) = SUM                                                    INTG0280
      SUM = 0.                                                          INTG0290
      DO 20 K=1,5                                                       INTG0300
   20 SUM = SUM + F2(I,K)*PH(K,J)                                       INTG0310
      DPH(I+3,J) = SUM                                                  INTG0320
   22 CONTINUE                                                          INTG0330
      GO TO 6                                                           INTG0340
   99 RETURN                                                            INTG0350
      END                                                               INTG0360
```

```
      SUBROUTINE MINVAR
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *    NALL,LRK
      COMMON /TWICE/ A(3,3)    ,CPH      ,CT(5)    ,SP(10)   ,ST(5)
     1     ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)
     2     ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)
     3     ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)
     4     ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2
     5     ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)
     6     ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)     ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)
     1     ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)
     2     ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)
     3     ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET
     4     ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J    ,JNBR,JNBRS,JST
     5     ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK
     6     ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT
     7     ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT
     8     ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)
     9     ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)
     A     ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)
     B     ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2
     C     ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)
     D     ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL
     E     ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)
     F     ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)
     G     ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)
     H     ,XYCG(20) ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
      G(1,10) = 0.
      G(2,10) = 0.
      G(3,10) = 0.
      NT = 0
      LT = 1
      MT = NSTX
      DO 1 I=1,NSTX
      P(I,10) = 0.
    1 P(10,I) = 0.
    2 DO 12 J=KAR,KOB
      DO 6 I=LT,MT
      L = I+1
      SUM = 0.
      DO 3 K=LT,I
    3 SUM = SUM + P(K,I)*G(J,K)
      IF(L.GT.MT) GO TO 6
      DO 4 K=L,MT
    4 SUM = SUM+P(I,K)*G(J,K)
    6 DUB(I,J) = SUM
C-------------CALCULATE EQ.(54F) ------------------------
      DO 10 I=KAR,KOB
      SUM = 0.
      DO  8 K=LT,MT
    8 SUM = SUM+G(I,K)*DUB(K,J)
   10 DUB(I,J+6) = SUM
   12 DUB(J,J+6) = DUB(J,J+6)+SI(J)**2
      IF(NPV) 36,36,14
```

```
   14 CONTINUE                                                          MINV0280
      DO 15 I=1,NPV                                                     MINV0290
   15 CVZ(10,I) = 0.                                                    MINV0300
      DO 20 J=KAR,KOB                                                   MINV0310
      DO 17 I=1,NSTX                                                    MINV0320
      SUM= 0.                                                           MINV0330
      DO 16 K=1,NPV                                                     MINV0340
   16 SUM= SUM+CVZ(I,K)*H(J,K)                                          MINV0350
      DUB(I,J+3) = SUM                                                  MINV0360
   17 DUB(I,J)     = DUB(I,J)+SUM                                       MINV0370
      DO 18 I=1,NPV                                                     MINV0380
   18 DUB(I+12,J+6) = S(I)*H(J,I)                                       MINV0390
   20 CONTINUE                                                          MINV0400
      DO 26 J=KAR,KOB                                                   MINV0410
      DO 24 I=KAR,KOB                                                   MINV0420
      SUM = 0.                                                          MINV0430
      SUM2= 0.                                                          MINV0440
      DO 22 K=LT,MT                                                     MINV0450
   22 SUM = SUM + DUB(K,J+3)*G(I,K)                                     MINV0460
      DO 23 K=1,NPV                                                     MINV0470
   23 SUM2= SUM2+ H(I,K)*DUB(K+12,J+6)                                  MINV0480
      DUB(J+3,I+6) = SUM                                                MINV0490
   24 DUB(I+6,J+6) = SUM + SUM2                                         MINV0500
   26 CONTINUE                                                          MINV0510
      DO 34 J=KAR,KOB                                                   MINV0520
      DO 28 I=KAR,KOB                                                   MINV0530
   28 DUB(I,J+6) = DUB(I,J+6) + DUB(I+3,J+6) + DUB(I+6,J+6)            MINV0540
      DO 32 I=1,NPV                                                     MINV0550
      SUM = 0.                                                          MINV0560
      DO 30 K=LT,MT                                                     MINV0570
   30 SUM = SUM + G(J,K)*CVZ(K,I)                                       MINV0580
   32 DUB(I,J+9) = SUM                                                  MINV0590
   34 CONTINUE                                                          MINV0600
   36 CONTINUE                                                          MINV0610
      IF(KAR-KOB) 37,46,37                                              MINV0620
   37 CONTINUE                                                          MINV0630
      DO 39 J=KAR,KOB                                                   MINV0640
      DO 38 I=KAR,KOB                                                   MINV0650
   38 DUB(I+9,J+6) = 0.                                                 MINV0660
   39 DUB(J+9,J+6) = 1.                                                 MINV0670
      L = KOB-1                                                         MINV0680
      DO 42 I=KAR,L                                                     MINV0690
      M  = I+1                                                          MINV0700
      SUM  = DUB(I,I+6)                                                 MINV0710
      DO 42 J=KAR,KOB                                                   MINV0720
      DUB(I   ,J+6) = DUB(I   ,J+6)/SUM                                 MINV0730
      DUB(I+9,J+6) = DUB(I+9,J+6)/SUM                                   MINV0740
      DO 42 K=M,KOB                                                     MINV0750
      IF(I-J) 40,41,40                                                  MINV0760
   40 DUB(K   ,J+6) = DUB(K   ,J+6)-DUB(K,I+6)*DUB(I   ,J+6)            MINV0770
   41 DUB(K+9,J+6) = DUB(K+9,J+6)-DUB(K,I+6)*DUB(I+9,J+6)               MINV0780
   42 CONTINUE                                                          MINV0790
      DO 44 I=KAR,KOB                                                   MINV0800
   44 DUB(KOB+9,I+6) = DUB(KOB+9,I+6)/DUB(KOB,KOB+6)                    MINV0810
      DO 45 I=KAR,L                                                     MINV0820
      M = KOB+KAR+6-I                                                   MINV0830
      DO 45 J=I,L                                                       MINV0840
      N = KOB+KAR+8-J                                                   MINV0850
      DO 45 K=KAR,KOB                                                   MINV0860
   45 DUB(N,K+6) = DUB(N,K+6)-DUB(N-9   ,M)*DUB(M+3,K+6)                MINV0870
```

```
      GO TO 48
   46 DUB(KAR+9,KAR+6) = 1./DUB(KAR,KAR+6)
   48 CONTINUE
C-------------CALCULATE EQ.(54E) -------------------------
      DO 54 I=1,NSTX
      DO 51 J=KAR,KOB
      SUM = 0.
      DO 50 K=KAR,KOB
   50 SUM = SUM + DUB(I,K)*DUB(K+9,J+6)
   51 DUB(I,J+12) = SUM
      DO 54 L=I,NSTX
      SUM = 0.
C-------------CALCULATE EQ.(54B) -------------------------
      DO 52 K=KAR,KOB
   52 SUM = SUM + DUB(I,K+12)*DUB(L,K)
      P(I,L) = P(I,L) - SUM
   54 CONTINUE
      IF(NPU) 66,66,56
C-------------CALCULATE EQ.(54C) -------------------------
   56 CONTINUE
      DO 57 I=1,NPU
   57 CUZ(10,I) = 0.
      DO 64 J=1,NPU
      DO 60 I=KAR,KOB
      SUM = 0.
      DO 58 K=1,NSTX
   58 SUM = SUM + G(I,K)*CUZ(K,J)
   60 DUB(I,1) = SUM
      DO 64 I=1,NSTX
      SUM = 0.
      DO 62 K=KAR,KOB
   62 SUM = SUM + DUB(I,K+12)*DUB(K,1)
   64 CUZ(I,J) = CUZ(I,J)-SUM
   66 IF(NPV) 72,72,67
C-------------CALCULATE EQ.(54D) -------------------------
   67 DO 70 I=1,NSTX
      DO 70 J=1,NPV
      SUM = 0.
      DO 68 K=KAR,KOB
   68 SUM = SUM + DUB(I,K+12)*(DUB(J,K+9)+DUB(J+12,K+6))
   70 CVZ(I,J) = CVZ(I,J) - SUM
   72 IF(NPC(1)-2) 74,108,74
   74 IF(NPC(4)) 75,88,75
   75 DO 82 I=1,NSTX
   80 DUB(I,1) = DZ(I)
   82 CONTINUE
C-------------CALCULATE EQ.(54A) -------------------------
      DO 86 I=KAR,KOB
      SUM = 0.
      DO 84 J=1,NSTX
   84 SUM = SUM + G(I,J)*DUB(J,1)
   86 DATC(I) = DATC(I)+SUM
   88 DO 90 I=KAR,KOB
      J = LC(I)
      DUB(I,2) = DATA(J)-DATC(I)
      IF(MTYP.GT.5) GO TO 90
      IF(J.EQ.1) GO TO 90
      IF(ABS(DUB(I,2)).LT.3.1416) GO TO 90
      DUB(I,2) = DUB(I,2)-SIGN(6.283185307179586,DUB(I,2))
   90 CONTINUE
```

```
      DO 92 I=1,NSTX                                                    MINV1480
      DZ(I) = 0.                                                        MINV1490
      DO 92 J=KAR,KOB                                                   MINV1500
   92 DZ(I) = DZ(I)+DUB(I,J+12)*DUB(J,2)                                MINV1510
      IF(NPC(4)) 294,298,294                                            MINV1520
  294 DO 296 I=1,4                                                      MINV1530
  296 VO(I) = X(I+6) + DUB(I+6,1)                                       MINV1540
      DZ(10) = -(VO(1)*DZ(7)+VO(2)*DZ(8)+VO(3)*DZ(9))/VO(4)             MINV1550
   94 DO 96 I=1,NSTX                                                    MINV1560
   96 DZ(I) = DZ(I)+DUB(I,1)                                            MINV1570
      GO TO 108                                                         MINV1580
  298 DZ(10) = -(X(7)*DZ(7)+X(8)*DZ(8)+X(9)*DZ(9))/X(10)                MINV1590
   98 DO 100 I=1,10                                                     MINV1600
  100 X(I) = X(I)+DZ(I)                                                 MINV1610
      DATC(1)= SQRT(X(7)**2 + X(8)**2 + X(9)**2 + X(10)**2)             MINV1620
      X(7) = X(7)/DATC(1)                                               MINV1630
      X(8) = X(8)/DATC(1)                                               MINV1640
      X(9) = X(9)/DATC(1)                                               MINV1650
      X(10)=X(10)/DATC(1)                                               MINV1660
      IF(NSTC.EQ.0) GO TO 103                                           MINV1670
      DO 102 I=1,NSTC                                                   MINV1680
      J = NC(I)                                                         MINV1690
  102 C(J) = C(J)+DZ(I+10)                                              MINV1700
  103 CONTINUE                                                          MINV1710
      IF(KSS) 108,108,104                                               MINV1720
  104 DO 106 I=1,KSS                                                    MINV1730
      MTYP = NSS(I)                                                     MINV1740
  106 CALL STAT                                                         MINV1750
  108 P(8,7) = P(7,8)                                                   MINV1760
      P(9,7) = P(7,9)                                                   MINV1770
      P(9,8) = P(8,9)                                                   MINV1780
      DO 110 I=1,9                                                      MINV1790
      SUM = P(I,7)*X(7) + P(I,8)*X(8) + P(I,9)*X(9)                     MINV1800
  110 P(I,10) =-SUM/X(10)                                               MINV1810
      P(10,10) = -(X(7)*P(7,10)+X(8)*P(8,10)+X(9)*P(9,10))/X(10)        MINV1820
      DO 114 I=11,NSTX                                                  MINV1830
      SUM = P(7,I)*X(7) + P(8,I)*X(8) + P(9,I)*X(9)                     MINV1840
  114 P(10,I) =-SUM/X(10)                                               MINV1850
      IF(NPU.EQ.0) GO TO 117                                            MINV1860
  115 DO 116 I=1,NPU                                                    MINV1870
      SUM = CUZ(7,I)*X(7) + CUZ(8,I)*X(8) + CUZ(9,I)*X(9)               MINV1880
  116 CUZ(10,I) =-SUM/X(10)                                             MINV1890
  117 IF(NPV.EQ.0)GO TO 99                                              MINV1900
      DO 118 I=1,NPV                                                    MINV1910
      SUM = CVZ(7,I)*X(7) + CVZ(8,I)*X(8) + CVZ(9,I)*X(9)               MINV1920
  118 CVZ(10,I) =-SUM/X(10)                                             MINV1930
   99 CONTINUE                                                          MINV1940
      RETURN                                                            MINV1950
      END                                                               MINV1960
```

```
      SUBROUTINE MOTION
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *       NALL,LRK
      COMMON /TWICE/ A(3,3)     ,CPH       ,CT(5)     ,SP(10)    ,ST(5)
     1      ,TPH        ,V0(20)    ,Z0(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)
     2      ,B(8)       ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)
     3      ,DUF(4)     ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)
     4      ,IDN  ,II   ,IP   ,IPC ,JJ   ,JN   ,KA  ,KD(3),KDUM  ,KG2
     5      ,K1   ,NNN  ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)
     6      ,RO         ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)      ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)
     1      ,AZ(40)     ,C(160)    ,CCAPH(5) ,CONRD     ,CPHDT(5)  ,CUZ(30,5)
     2      ,CVZ(30,5),D(5)        ,DAT(3,40),DATA(3)   ,DATAS(3)  ,DATC(3)
     3      ,DET        ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET
     4      ,FLOS       ,G(3,30)   ,H(3,5)    ,I  ,ICOUNT,J    ,JNBR,JNBRS,JST
     5      ,K    ,KAR  ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK
     6      ,KN   ,KOB  ,KPROP,KS,KSM,KSS,L ,LC(3)      ,LCS(3)    ,LS    ,LT
     7      ,L1   ,M    ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT
     8      ,MTP(40)    ,MTYP      ,MTYPS     ,N  ,NC(30),NCOUNT    ,NPC(15)
     9      ,NPU  ,NPV  ,NS(9)     ,NSS(5)    ,NST  ,NSTA,NSTC,NSTX,NT,NTR(9)
     A      ,N8(3)      ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)
     B      ,PA(3)                 ,PM(3)     ,R         ,REO       ,RERP2
     C      ,RPO        ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40)
     D      ,SIGM(3)    ,SIGMS(3)  ,SPHDT(5)  ,SUM  ,SUM2,SYG(3,9)  ,TFINAL
     E      ,TFIT(9)    ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)
     F      ,TR(40)     ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)
     G      ,TZERO      ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)
     H      ,XYCG(20)   ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
      LT = 0
    1 CONTINUE
      CPH = COS(X(5))
      SPH = SIN(X(5))
      TPH = SPH/CPH
      R = REO + X(4)
C-------------CALCULATE EQ.(159) ---------------------------
      AP(1) = C(61)*AM(1)+C(62)*AM(2)+C(63)*AM(3)+C(70)
      AP(2) = C(64)*AM(1)+C(65)*AM(2)+C(66)*AM(3)+C(71)
      AP(3) = C(67)*AM(1)+C(68)*AM(2)+C(69)*AM(3)+C(72)
C-------------CALCULATE EQ.(155) ---------------------------
      AP(4) = SQRT(AP(1)**2+ AP(3)**2)
      IF(AP(4)) 9,9,7
    7 CONTINUE
      AP(4) = 2.*AP(1)*AP(3)/AP(4)
    9 CONTINUE
C-------------CALCULATE EQ.(154) ---------------------------
      DO 11 I=1,3
      J = 3*I
      PA(I)=C(J+33)*PM(1)+C(J+34)*PM(2)+C(J+35)*PM(3)+C(I+44)+C(I+47)*
     X      AP(1) + C(I+50)*AP(2) + C(I+53)*AP(3) + C(I+56)*AP(4)
   11 CONTINUE
C-------------CALCULATE EQ.(153) ---------------------------
      XP(1) = CGM(1) + C(31)
      XP(2) = CGM(2) + C(32)
      XP(3) = CGM(3) + C(33)
      I=N11(2)
```

```
      IF(N11(2)-1)10,10,12
   10 I = 2
   12 CONTINUE
      SP(1)= TT(I+1)-TT(I-1)
      HI(2)= .5*SP(1)
      HI(1)= T     + HI(2)
      HI(2)= T     - HI(2)
      PMDOT(1)= (TAB(HI(1),N11(1),TT(1),TP(1))-TAB(HI(2),N11(1),TT(1),TP
     X(1)))/SP(1)
      PMDOT(2)= (TAB(HI(1),N11(1),TT(1),TQ(1))-TAB(HI(2),N11(1),TT(1),TQ
     X(1)))/SP(1)
      PMDOT(3)= (TAB(HI(1),N11(1),TT(1),TR(1))-TAB(HI(2),N11(1),TT(1),TR
     X(1)))/SP(1)
      AMDOT(1)= (TAB(HI(1),N11(1),TT(1),AX(1))-TAB(HI(2),N11(1),TT(1),AX
     X(1)))/SP(1)
      AMDOT(2)= (TAB(HI(1),N11(1),TT(1),AY(1))-TAB(HI(2),N11(1),TT(1),AY
     X(1)))/SP(1)
      AMDOT(3)= (TAB(HI(1),N11(1),TT(1),AZ(1))-TAB(HI(2),N11(1),TT(1),AZ
     X(1)))/SP(1)
C-------------CALCULATE EQ.(158) --------------------------
      APDOT(1) = C(61)*AMDOT(1) + C(62)*AMDOT(2) + C(63)*AMDOT(3)
      APDOT(2) = C(64)*AMDOT(1) + C(65)*AMDOT(2) + C(66)*AMDOT(3)
      APDOT(3) = C(67)*AMDOT(1) + C(68)*AMDOT(2) + C(69)*AMDOT(3)
C-------------CALCULATE EQ.(157) --------------------------
      APDOT(4) = (APDOT(1)*((1.-(AP(4)/(2.*AP(3))))**2)/AP(1))
     X          + APDOT(3)*((1.-(AP(4)/(2.*AP(1))))**2)/AP(3)))*AP(4)
C-------------CALCULATE EQ.(156) --------------------------
   13 DO 14 I=1,3
      J = 3*I
      PDOT(I) = C(J+33)*PMDOT(1) + C(J+34)*PMDOT(2) + C(J+35)*PMDOT(3)
     X         + C(I+47)*APDOT(1) + C(I+50)*APDOT(2) + C(I+53)*APDOT(3)
     Y         + C(I+56)*APDOT(4)
   14 CONTINUE
C-------------CALCULATE EQ.(146) --------------------------
      SP(1) = PA(1)**2
      SP(2) = PA(2)**2
      SP(3) = PA(3)**2
      TRAN(1,1)=-SP(2)-SP(3)
      TRAN(1,2)=  PA(1)*PA(2)-PDOT(3)
      TRAN(1,3)=  PA(1)*PA(3)+PDOT(2)
      TRAN(2,1)=  PA(1)*PA(2)+PDOT(3)
      TRAN(2,2)=-SP(1)-SP(3)
      TRAN(2,3)=  PA(2)*PA(3)-PDOT(1)
      TRAN(3,1)=  PA(1)*PA(3)-PDOT(2)
      TRAN(3,2)=  PA(2)*PA(3)+PDOT(1)
      TRAN(3,3)=-SP(1)-SP(2)
      AB(1)  = AP(1)-TRAN(1,1)*XP(1)-TRAN(1,2)*XP(2)-TRAN(1,3)*XP(3)
      AB(2)  = AP(2)-TRAN(2,1)*XP(1)-TRAN(2,2)*XP(2)-TRAN(2,3)*XP(3)
      AB(3)  = AP(3)-TRAN(3,1)*XP(1)-TRAN(3,2)*XP(2)-TRAN(3,3)*XP(3)
C-------------CALCULATE EQ.(133) --------------------------
      SP(1) = (X( 7)-X( 8))*(X( 7)+X( 8))
      SP(2) = (X( 9)-X(10))*(X( 9)+X(10))
      A(1,1) = (X( 7)-X( 9))*(X( 7)+X( 9)) + (X( 8)-X(10))*(X( 8)+X(10))
      A(2,2) = SP(1) + SP(2)
      A(3,3) = SP(1) - SP(2)
      A(1,2) = 2.*(X( 8)*X( 9)+X( 7)*X(10))
      A(2,1) = 2.*(X( 8)*X( 9)-X( 7)*X(10))
      A(1,3) = 2.*(X( 8)*X(10)-X( 7)*X( 9))
      A(3,1) = 2.*(X( 8)*X(10)+X( 7)*X( 9))
      A(3,2) = 2.*(X(10)*X( 9)-X( 7)*X( 8))
```

```
      A(2,3) = 2.*(X(10)*X( 9)+X( 7)*X( 8))
      IF(LT) 15,15,99
   15 CONTINUE
C------------CALCULATE EQ.(132) -------------------------
      AG(1)  = AB(1)*A(1,1)+AB(2)*A(2,1)+AB(3)*A(3,1)
      AG(2)  = AB(1)*A(1,2)+AB(2)*A(2,2)+AB(3)*A(3,2)
      AG(3)  = AB(1)*A(1,3)+AB(2)*A(2,3)+AB(3)*A(3,3)
C     PARAMETERS USED FREGUENTLY IN REMAINING SUBROUTINES
      PAR(1) = X(1)*X(1) + X(2)*X(2)
      PAR(2)=XMU/R**2
      PAR(3)=XMUJ/R**4
      PAR(4) = 1./(2.*R)
      PAR(5) = X(1)*X(3)-X(2)*X(2)*TPH
      PAR(6) = 1. - 1.5*CPH*CPH
      F1(2,2)= (X(1)*TPH+X(3))/R
      F1(2,3)= X(2)/R
      F1(2,1)= F1(2,3)*TPH
      DX(5) = X(1)/R
      F3(1,8) =-.5*(PA(1)-F1(2,3))
      F3(1,9) =-.5*(PA(2)+DX(5))
      F3(1,10)=-.5*(PA(3)+F1(2,1))
      F3(2,9) = F3(1,10) + PA(3)
      F3(2,10)= F3(1,9)  + DX(5)
      F3(3,10)= F3(1,8)  + PA(1)
C------------CALCULATE EQ.(127) -------------------------
      DX(1) = AG(1) + PAR(5)/R          - PAR(3)*SPH*CPH
      DX(2) = AG(2) + X(2)*F1(2,2)
      DX(3) = AG(3) + PAR(2) - PAR(1)/R - PAR(3)*PAR(6)
C------------CALCULATE EQ.(128) -------------------------
      DX(4) = -X(3)
      DX(6) = F1(2,3)/CPH - OMEGA
C------------CALCULATE EQ.(129) -------------------------
      DX( 7) = X( 8)*F3(1, 8) + X( 9)*F3(1, 9) + X(10)*F3(1,10)
      DX( 8) =-X( 7)*F3(1, 8) + X( 9)*F3(2, 9) + X(10)*F3(2,10)
      DX( 9) =-X( 7)*F3(1, 9) - X( 8)*F3(2, 9) + X(10)*F3(3,10)
      DX(10) =-X( 7)*F3(1,10) - X( 8)*F3(2,10) - X( 9)*F3(3,10)
   99 RETURN
      ENTRY AUXIL
      LT = 1
      GO TO 1
      END
```

```
      SUBROUTINE OBSERV
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),     COMM0010
     *     NALL,LRK                                                     COMM0020
      COMMON /TWICE/ A(3,3)     ,CPH      ,CT(5)    ,SP(10)   ,ST(5)    COMM0030
     1     ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)  COMM0040
     2     ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)   COMM0050
     3     ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)     COMM0060
     4     ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2           COMM0070
     5     ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)    COMM0080
     6     ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)              COMM0090
      DIMENSION DUB(30,15)                                              COMM0100
      EQUIVALENCE (AG(1),DUB(1,1))                                      COMM0110
      COMMON /METH2/ AB(3)      ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)   COMM0120
     1     ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5) COMM0130
     2     ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)   COMM0140
     3     ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET        COMM0150
     4     ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J    ,JNBR,JNBRS,JST COMM0160
     5     ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK  COMM0170
     6     ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS  ,LT   COMM0180
     7     ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT        COMM0190
     8     ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)   COMM0200
     9     ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9) COMM0210
     A     ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)  COMM0220
     B     ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2     COMM0230
     C     ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40) COMM0240
     D     ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL    COMM0250
     E     ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)    COMM0260
     F     ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)  COMM0270
     G     ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)  COMM0280
     H     ,XYCG(20) ,XZCG(20)                                          COMM0290
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                  COMM0300
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                       COMM0310
      DIMENSION      CTM(6)    ,RCC(3)    ,RCDC(3)   ,RCDX(3,6),RCX(3,6) , OBSV0010
     A     XA(5)     ,XAX(5,7) ,XC(3)     ,XCX(3,7)  ,XLB(3)   ,XLG(3)   , OBSV0020
     B     XLGE(3,4),XLGX(2,4),XS(3)      ,XSC(3)    ,XSD(3)   ,XSDC(3)  , OBSV0030
     C     XSDX(3,6),XSX(3,3) ,CI(3,3)    ,CJ(3,3)                       OBSV0040
      EQUIVALENCE    (DPH(1,1) ,CTM(1)     ),    (DPH(7,1)  ,RCC(1)    ), OBSV0050
     N               (DPH(10,1),RCDC(1)    ),    (DPH(3,2)  ,RCDX(1,1)), OBSV0060
     O               (DPH(1,4) ,RCX(1,1)   ),    (DPH(9,5)  ,XA(1)     ), OBSV0070
     P               (DPH(4,6) ,XAX(1,1)   ),    (DPH(9,9)  ,XC(1)     ), OBSV0080
     Q               (DPH(2,10),XCX(1,1)   ),    (DPH(3,12),XLB(1)     ), OBSV0090
     R               (DPH(6,12),XLG(1)     ),    (DPH(9,12),XLGE(1,1)), OBSV0100
     S               (DPH(1,14),XLGX(1,1)),      (DPH(9,14),XS(1)      ), OBSV0110
     T               (DPH(2,15),XSC(1)     ),    (DPH(5,15),XSD(1)     ), OBSV0120
     U               (DPH(8,15),XSDC(1)    ),    (DPH(1,16),XSDX(1,1)), OBSV0130
     V               (DPH(9,17),XSX(1,1)   ),    (DPH(8,18),CI(1,1)    ), OBSV0140
     W               (DPH(7,19),CJ(1,1)    ),    (DPH(6,20),CAZC       ), OBSV0150
     X               (DPH(7,20),CDP        ),    (DPH(8,20),CDY        ), OBSV0160
     Y               (DPH(9,20),CELC       ),    (DPH(10,20),CGR       ), OBSV0170
     Z               (DPH(1,21),CLR        ),    (DPH(2,21),CTH        ), OBSV0180
     1               (DPH(3,21),SAZC       ),    (DPH(4,21),SDP        ), OBSV0190
     2               (DPH(5,21),SDY        ),    (DPH(6,21),SELC       ), OBSV0200
     3               (DPH(7,21),SGR        ),    (DPH(8,21),SLR        ), OBSV0210
     4               (DPH(9,21),STH        ),    (DPH(10,21),THETA     )  OBSV0220
      IF(L1-2) 14,10,14                                                 OBSV0230
   10 IF(NPC(5)) 14,12,14                                               OBSV0240
   12 WRITE(OUT,1000)                                                   OBSV0250
 1000 FORMAT(5X,6HPOINTS,5X,4HTYPE,4X,4HTIME,8X,5HCOMP1,10X,5HCOMP2,10X,OBSV0260
     15HCOMP3,10X,4HSIG1,11X,4HSIG2,11X,4HSIG3)                         OBSV0270
```

```
      L1=1
   14 IF(MTYP-6) 16,150,15
   15 IF(MTYP-8) 115,120,125
   16 N = 60+MTYP*15
C-------------CALCULATE EQ.(236) --------------------------
  100 THETA=X(6)-C(N+14)
      STH=SIN(THETA)
      CTH=COS(THETA)
      VO(1)= CPH*CPHDT(MTYP)
      VO(2)= CPH*SPHDT(MTYP)
      VO(3)= SPH*CPHDT(MTYP)
      VO(4)= SPH*SPHDT(MTYP)
      SP(1) = ROT(MTYP) + C(N+15)
      VO(5) = SP(1)*CCAPH(MTYP)
      VO(6) = SP(1)*SCAPH(MTYP)
C-------------CALCULATE EQ.(235) --------------------------
  102 XS(1)= R*(CTH*VO(1)+VO(4))-VO(5)
      XS(2)= R*STH*CPH
      XS(3)=-R*(CTH*VO(2)-VO(3))+VO(6)
      VO(8)= SQRT(XS(2)**2+XS(3)**2)
      VO(9)= VO(8)**2
C-------------CALCULATE EQ.(238A)--------------------------
      CTM(1)= SQRT(XS(1)**2+VO(9))
C-------------CALCULATE EQ.(238B)--------------------------
      SAZC= XS(2)/VO(8)
      CAZC= XS(3)/VO(8)
C-------------CALCULATE EQ.(238C)--------------------------
      SELC= XS(1)/CTM(1)
      CELC= VO(8)/CTM(1)
      CTM(2)= ATAN2(XS(2),XS(3))
      CTM(3)= ATAN2(XS(1),VO(8))
C-------------CALCULATE EQ.(239) --------------------------
      DFM(1)= C(N+1)*CTM(1)+C(N+4)+C(N+10)/SELC
      DFM(2)= C(N+2)*CTM(2)+C(N+5)
      DFM(3)= C(N+3)*CTM(3)+C(N+6)+C(N+11)*CELC/SELC
      IF(MO(MTYP)) 106,104,106
  104 IF(NPC(1)-1) 106,18,106
C-------------CALCULATE EQ.(246) --------------------------
  106 XSX(1,1)= (XS(1)+VO(5))/R
      XSX(1,2)= R*(VO(2)-CTH*VO(3))
      XSX(1,3)= -XS(2)*CPHDT(MTYP)
      XSX(2,1)= XS(2)/R
      XSX(2,2)= -R*STH*SPH
      XSX(2,3)=  R*CTH*CPH
      XSX(3,1)= (XS(3)-VO(6))/R
      XSX(3,2)= R*(CTH*VO(4)+VO(1))
      XSX(3,3)= XS(2)*SPHDT(MTYP)
      IF(MO(MTYP)) 108,18,108
C-------------CALCULATE EQ.(241) --------------------------
  108 DO 112 I=1,3
      SUM=0.
      DO 110 J=1,3
  110 SUM=SUM+XSX(I,J)*DX(J+3)
  112 XSD(I)=SUM
C-------------CALCULATE EQ.(240) --------------------------
      CTM(4)= (XS(1)*XSD(1)+XS(2)*XSD(2)+XS(3)*XSD(3))/CTM(1)
      CTM(5)= (XS(3)*XSD(2)-XS(2)*XSD(3))/VO(9)
      CTM(6)= (CTM(1)*XSD(1)-XS(1)*CTM(4))/(CTM(1)*VO(8))
      DFM(1)= DFM(1)+C(N+7)*CTM(4)
      DFM(2)= DFM(2)+C(N+8)*CTM(5)
```

```
      DFM(3)= DFM(3)+C(N+9)*CTM(6)                                      OBSV0880
      GO TO 18                                                          OBSV0890
  115 DFM(1)= X(1)                                                      OBSV0900
      DFM(2)= X(2)                                                      OBSV0910
      DFM(3)= X(3)                                                      OBSV0920
      GO TO 18                                                          OBSV0930
  120 DFM(1)= X(4)                                                      OBSV0940
      DFM(2)= X(5)                                                      OBSV0950
      DFM(3)= X(6)                                                      OBSV0960
      GO TO 18                                                          OBSV0970
  125 WRITE(OUT,5000) MTYP                                              OBSV0980
 5000 FORMAT(35H ILLIGITIMATE MTYP IN OBSERV, MTYP=I3)                  OBSV0990
      CALL EXIT                                                         OBSV1000
C 150 CALCULATE AMR                                                     OBSV1010
  150 SDP= SIN(C(151))                                                  OBSV1020
      CDP= COS(C(151))                                                  OBSV1030
      SDY= SIN(C(152))                                                  OBSV1040
      CDY= COS(C(152))                                                  OBSV1050
C-------------CALCULATE EQ.(269) --------------------------              OBSV1060
      XLB(1)= CDY*CDP                                                   OBSV1070
      XLB(2)= SDY                                                       OBSV1080
      XLB(3)= CDY*SDP                                                   OBSV1090
C-------------CALCULATE EQ.(270) --------------------------              OBSV1100
      DO 152 I=1,3                                                      OBSV1110
      XLG(I)=0.                                                         OBSV1120
      DO 152 J=1,3                                                      OBSV1130
  152 XLG(I)=XLG(I)+A(J,I)*XLB(J)                                       OBSV1140
      VO(1)= REO**2                                                     OBSV1150
C-------------CALCULATE EQ.(271A)--------------------------              OBSV1160
      VO(2)= ATAN2(XLG(2),XLG(1))                                       OBSV1170
C-------------CALCULATE EQ.(271B)--------------------------              OBSV1180
      SP(1)= SQRT(XLG(1)**2+XLG(2)**2)                                  OBSV1190
      VO(3)=-ATAN2(XLG(3),SP(1))                                        OBSV1200
      SLR= SIN(VO(2))                                                   OBSV1210
      CLR= COS(VO(2))                                                   OBSV1220
      SGR= SIN(VO(3))                                                   OBSV1230
      CGR= COS(VO(3))                                                   OBSV1240
C-------------CALCULATE EQ.(277) --------------------------              OBSV1250
      XA(1)= SLR*CGR                                                    OBSV1260
      XA(2)= R*CPH                                                      OBSV1270
      XA(3)= CPH*SGR-SPH*CLR*CGR                                        OBSV1280
      XA(4)= R*SPH                                                      OBSV1290
      XA(5)= SPH*SGR+CPH*CLR*CGR                                        OBSV1300
C-------------CALCULATE EQ.(276) --------------------------              OBSV1310
      XC(1)= XA(1)**2 + XA(3)**2 + RERP2*XA(5)**2                       OBSV1320
      XC(2)= 2.*(XA(2)*XA(3)+ RERP2*XA(4)*XA(5))                        OBSV1330
      XC(3)= XA(2)**2 + RERP2*XA(4)**2 - VO(1)                          OBSV1340
      SP(2)= XC(2)**2 - 4.*XC(1)*XC(3)                                  OBSV1350
      IF(SP(2)) 154,156,156                                             OBSV1360
  154 WRITE(OUT,5022)                                                   OBSV1370
 5022 FORMAT(49H IMAGINARY OR NEGATIVE SOLUTION ON AIRBORNE RADAR)      OBSV1380
      DO 160 I=1,3                                                      OBSV1390
      DFM(I)= 0.                                                        OBSV1400
      DO 158 J=1,NSTX                                                   OBSV1410
  158 G(I,J)= 0.                                                        OBSV1420
      DO 160 J=1,NPV                                                    OBSV1430
      H(I,J) = 0.                                                       OBSV1440
  160 CONTINUE                                                          OBSV1450
      SI(1)=1.E20                                                       OBSV1460
      GO TO 99                                                          OBSV1470
```

```
C-------------CALCULATE EQ.(273) --------------------------             OBSV1480
  156 DFM(1)= -.5*(XC(2)+SQRT(SP(2)))/XC(1)                               OBSV1490
      IF(DFM(1).LT.0.) GO TO 154                                          OBSV1500
      DFM(2)= 0.                                                          OBSV1510
      DFM(3)= 0.                                                          OBSV1520
   18 IF(DCOMP.LT.0.) GO TO 80                                            OBSV1530
      DO 20 I=1,3                                                         OBSV1540
      SP(I) = SYG(I,MTYP)                                                 OBSV1550
      IF(NPC(1).EQ.0) SP(I) = SP(I)+SIGM(I)                               OBSV1560
   20 KC(I)=0                                                             OBSV1570
      IF(KN) 22,22,24                                                     OBSV1580
   22 KOB=KAR                                                             OBSV1590
      GO TO 26                                                            OBSV1600
   24 KOB=JNBR                                                            OBSV1610
   26 DO 28 I=KAR,KOB                                                     OBSV1620
      J=LC(I)                                                             OBSV1630
      DATC(I)=DFM(J)                                                      OBSV1640
      SI(I)=SP(J)                                                         OBSV1650
      NCOUNT=NCOUNT+1                                                     OBSV1660
   28 KC(J) = NCOUNT                                                      OBSV1670
   30 IF(NPC(5)) 34,32,34                                                 OBSV1680
   32 CONTINUE                                                            OBSV1690
      WRITE(OUT,1002) (KC(I),I=1,3),MTYP,TO,(DFM(I),I=1,3),(SP(I),I=1,3)OBSV1700
 1002 FORMAT(1X,I4,1H,I4,1H,I4,2X,I2,1X,F10.4,F15.2,5F15.6)               OBSV1710
   34 DO 36 I=1,NSTX                                                      OBSV1720
      DO 36 J=KAR,KOB                                                     OBSV1730
   36 G(J,I)=0.                                                           OBSV1740
      IF(MTYP-6) 200,250,270                                              OBSV1750
C 200 CALCULATE G(1-10) FOR R,A,E                                         OBSV1760
C-------------CALCULATE EQ.(244) --------------------------              OBSV1770
  200 DO 202 I=1,3                                                        OBSV1780
      RCX(1,I)= (XSX(1,I)*XS(1)+XSX(2,I)*XS(2)+XSX(3,I)*XS(3))/CTM(1)     OBSV1790
      RCX(2,I)= (XSX(2,I)*XS(3)-XSX(3,I)*XS(2))/VO(9)                     OBSV1800
  202 RCX(3,I)= (XSX(1,I)-RCX(1,I)*SELC)/VO(8)                            OBSV1810
      IF(MO(MTYP)) 204,218,204                                            OBSV1820
  204 SP(1)= R*CPH                                                        OBSV1830
      SP(2)= DX(5)/R                                                      OBSV1840
      SP(3)= X(2)/(R*SP(1))                                               OBSV1850
      SP(4)= X(2)*TPH/SP(1)                                               OBSV1860
C-------------CALCULATE EQ.(247)---------------------------              OBSV1870
      DO 206 I=1,3                                                        OBSV1880
      XSDX(I,1)= XSX(I,2)/R                                               OBSV1890
      XSDX(I,2)= XSX(I,3)/SP(1)                                           OBSV1900
      XSDX(I,3)=-XSX(I,1)                                                 OBSV1910
      XSDX(I,4)=-XSX(I,2)*SP(2)-XSX(I,3)*SP(3)                            OBSV1920
  206 XSDX(I,5)= XSX(I,3)*SP(4)                                           OBSV1930
      VO(10)=-X(3)/R                                                      OBSV1940
      SP(4)= DX(6)/R                                                      OBSV1950
      SP(5)= DX(5)/R                                                      OBSV1960
      SP(6)=-TPH*DX(6)                                                    OBSV1970
      SP(7)=-R*DX(5)                                                      OBSV1980
      SP(1)= CTH*DX(6)/STH - TPH*DX(5)                                    OBSV1990
      SP(3)= SP(1)                                                        OBSV2000
      SP(2)=-STH*DX(6)/CTH - TPH*DX(5)                                    OBSV2010
      DO 208 I=1,3                                                        OBSV2020
      XSDX(I,4)= XSDX(I,4)+XSX(I,3)*SP(4)+XSX(I,2)*SP(5)                  OBSV2030
      XSDX(I,5)= XSDX(I,5)+XSX(I,2)*VO(10)+XSX(I,3)*SP(6)+XSX(I,1)*SP(7)OBSV2040
  208 XSDX(I,6)= XSX(I,3)*VO(10)+XSX(I,3)*SP(I)                           OBSV2050
      VO(11)=-(XSD(3)+2.*CTM(5)*XS(2))                                    OBSV2060
      VO(12)= XSD(2)-2.*CTM(5)*XS(3)                                      OBSV2070
```

```
      SP(1)= -CTM(6)*XS(2)/CELC                                          OBSV2080
      SP(2)= -CTM(6)*XS(3)/CELC                                          OBSV2090
      SP(3)= XSD(1)-CTM(6)*VO(8)                                         OBSV2100
C-------------CALCULATE EQ.(245) --------------------------              OBSV2110
      DO 216 I=1,6                                                       OBSV2120
      IF(I-3) 210,210,212                                                OBSV2130
  210 RCDX(1,I)=0.                                                       OBSV2140
      RCDX(2,I)=0.                                                       OBSV2150
      RCDX(3,I)=0.                                                       OBSV2160
      GO TO 214                                                          OBSV2170
  212 RCDX(1,I)=XSX(1,I-3)*XSD(1)+XSX(2,I-3)*XSD(2)+XSX(3,I-3)*XSD(3)    OBSV2180
     X              -RCX(1,I-3)*CTM(4)                                   OBSV2190
      RCDX(2,I)=XSX(3,I-3)*VO(12)+XSX(2,I-3)*VO(11)                      OBSV2200
      RCDX(3,I)=-XSX(1,I-3)*CTM(4)+XSX(2,I-3)*SP(1)+XSX(3,I-3)*SP(2)     OBSV2210
     X              +RCX(1,I-3)*SP(3)                                    OBSV2220
  214 RCDX(1,I)=(RCDX(1,I)+XSDX(1,I)*XS(1)+XSDX(2,I)*XS(2)               OBSV2230
     X              +XSDX(3,I)*XS(3))/CTM(1)                             OBSV2240
      RCDX(2,I)=(RCDX(2,I)+XSDX(2,I)*XS(3)-XSDX(3,I)*XS(2))/VO(9)        OBSV2250
  216 RCDX(3,I)=((RCDX(3,I)-RCDX(1,I)*XS(1))/CTM(1)+XSDX(1,I))/VO(8)     OBSV2260
  218 VO(13)= C(N+10)*CELC/SELC**2                                       OBSV2270
      VO(14)= C(N+3)-C(N+11)/SELC**2                                     OBSV2280
      DO 230 I=1,6                                                       OBSV2290
      IF(I-3) 220,220,222                                                OBSV2300
  220 GG(1)=0.                                                           OBSV2310
      GG(2)=0.                                                           OBSV2320
      GG(3)=0.                                                           OBSV2330
      GO TO 224                                                          OBSV2340
C-------------CALCULATE EQ.(243) --------------------------              OBSV2350
  222 GG(1)= C(N+1)*RCX(1,I-3)-VO(13)*RCX(3,I-3)                         OBSV2360
      GG(2)= C(N+2)*RCX(2,I-3)                                           OBSV2370
      GG(3)= VO(14)*RCX(3,I-3)                                           OBSV2380
  224 IF(MO(MTYP)) 226,228,226                                           OBSV2390
  226 GG(1)= GG(1)+ C(N+7)*RCDX(1,I)                                     OBSV2400
      GG(2)= GG(2)+ C(N+8)*RCDX(2,I)                                     OBSV2410
      GG(3)= GG(3)+ C(N+9)*RCDX(3,I)                                     OBSV2420
  228 DO 230 K=KAR,KOB                                                   OBSV2430
      J=LC(K)                                                            OBSV2440
  230 G(K,I)=GG(J)                                                       OBSV2450
  232 CONTINUE                                                           OBSV2460
      GO TO 38                                                           OBSV2470
C-------------CALCULATE EQ.(286)---------------------------              OBSV2480
  250 XLGE(1,1)= 2.*( X( 7)*XLB(1)-X(10)*XLB(2)+X( 9)*XLB(3))            OBSV2490
      XLGE(1,2)= 2.*( X( 8)*XLB(1)+X( 9)*XLB(2)+X(10)*XLB(3))            OBSV2500
      XLGE(1,3)= 2.*(-X( 9)*XLB(1)+X( 8)*XLB(2)+X( 7)*XLB(3))            OBSV2510
      XLGE(1,4)= 2.*(-X(10)*XLB(1)-X( 7)*XLB(2)+X( 8)*XLB(3))            OBSV2520
      XLGE(2,1)= 2.*( X(10)*XLB(1)+X( 7)*XLB(2)-X( 8)*XLB(3))            OBSV2530
      XLGE(2,2)= 2.*( X( 9)*XLB(1)-X( 8)*XLB(2)-X( 7)*XLB(3))            OBSV2540
      XLGE(2,3)= 2.*( X( 8)*XLB(1)+X( 9)*XLB(2)+X(10)*XLB(3))            OBSV2550
      XLGE(2,4)= 2.*( X( 7)*XLB(1)-X(10)*XLB(2)+X( 9)*XLB(3))            OBSV2560
      XLGE(3,1)= 2.*(-X( 9)*XLB(1)+X( 8)*XLB(2)+X( 7)*XLB(3))            OBSV2570
      XLGE(3,2)= 2.*( X(10)*XLB(1)+X( 7)*XLB(2)-X( 8)*XLB(3))            OBSV2580
      XLGE(3,3)= 2.*(-X( 7)*XLB(1)+X(10)*XLB(2)-X( 9)*XLB(3))            OBSV2590
      XLGE(3,4)= 2.*( X( 8)*XLB(1)+X( 9)*XLB(2)+X(10)*XLB(3))            OBSV2600
      SP(4)= SGR/CGR                                                     OBSV2610
      SP(3)= XLG(1)**2 + XLG(2)**2                                       OBSV2620
      SP(1)= SQRT(SP(3))                                                 OBSV2630
      DO 258 I=1,4                                                       OBSV2640
C-------------CALCULATE EQ.(284) --------------------------              OBSV2650
      XLGX(1,I) = (XLG(1)*XLGE(2,I)-XLG(2)*XLGE(1,I))/SP(3)              OBSV2660
C-------------CALCULATE EQ.(285) --------------------------              OBSV2670
```

```
  258 XLGX(2,I) =-SP(1)*XLGE(3,I)-SP(4)*(XLG(1)*XLGE(1,I)+XLG(2)*XLGE(2,
     1I))
      VO(4)= -SLR*SGR
      VO(5)=  CLR*CGR
      VO(6)=  CPH*CGR+SPH*CLR*SGR
      VO(7)=  SPH*XA(1)
      VO(9)=  SPH*CGR-CPH*CLR*SGR
      VO(10)= CPH*XA(1)
C-------------CALCULATE EQ.(283)---------------------------
      DO 260 I=1,5
      DO 260 J=1,7
  260 XAX(I,J)= 0.
      DO 262 I=1,4
      XAX(1,I+3) = VO(4)*XLGX(2,I) + VO(5)*XLGX(1,I)
      XAX(3,I+3) = VO(6)*XLGX(2,I) + VO(7)*XLGX(1,I)
  262 XAX(5,I+3) = VO(9)*XLGX(2,I) - VO(10)*XLGX(1,I)
      XAX(2,1) = CPH
      XAX(2,2) =-XA(4)
      XAX(3,2) =-XA(5)
      XAX(4,1) = SPH
      XAX(4,2) = XA(2)
      XAX(5,2) = XA(3)
C-------------CALCULATE EQ.(282) --------------------------
      DO 264 I=1,7
      XCX(1,I)= 2.*(XA(1)*XAX(1,I)+XA(3)*XAX(3,I)+RERP2*XA(5)*XAX(5,I))
      XCX(2,I)= 2.*(XA(2)*XAX(3,I)+XA(3)*XAX(2,I)+RERP2*(XA(4)*XAX(5,I)
     1         + XA(5)*XAX(4,I)))
  264 XCX(3,I)= 2.*(XA(2)*XAX(2,I)+RERP2*XA(4)*XAX(4,I))
      G(1,1)= 0.
      G(1,2)= 0.
      G(1,3)= 0.
      VO(11)= 2.*XC(1)*DFM(1)+XC(2)
      VO(12)= (DFM(1)+ XC(3)/VO(11))/XC(1)
      VO(13)= (.5-.5*XC(2)/VO(11))/XC(1)
      VO(14)= 1./VO(11)
C-------------CALCULATE EQ.(281) --------------------------
      DO 266 I=1,7
  266 G(1,I+3)= -VO(12)*XCX(1,I)-VO(13)*XCX(2,I)-VO(14)*XCX(3,I)
      GO TO 38
  270 IF(MTYP-8) 272,274,375
  272 LT=0
      GO TO 276
  274 LT=3
  276 DO 284 I=KAR,KOB
      J=LC(I)
  284 G(I,LT+J)=1.
   38 CONTINUE
      IF(NSTC) 40,40,46
   40 IF(NPV) 99,99,42
   42 II=1
      IP=1
   44 JJ=MCC(II)
      GO TO 50
   46 II=1
      IP=0
   48 JJ=NC(II)
   50 GG(1)=0.
      GG(2)=0.
      GG(3)=0.
      IF(MTYP.GE.6) GO TO 375
```

```
      IF(JJ-N) 60,60,54
   54 IF(JJ-N-15) 56,56,60
   56 JN=JJ-N
      IF(JN-12) 58,58,350
   58 GO TO(300,301,302,303,304,305,306,307,308,309,310,311),JN
   60 II=II+1
      DO 66 I=KAR,KOB
      J=LC(I)
      IF(IP) 62,62,64
   62 G(I,II+ 9) = GG(J)
      GO TO 66
   64 H(I,II-1) = GG(J)
   66 CONTINUE
      IF(IP) 68,68,70
   68 IF(II-NSTC) 48,48,40
   70 IF(II-NPV) 44,44,99
C-------------CALCULATE EQ.(248) --------------------------
  300 GG(1) = CTM(1)
      GO TO 60
  301 GG(2) = CTM(2)
      GO TO 60
  302 GG(3) = CTM(3)
      GO TO 60
  303 GG(1) = 1.
      GO TO 60
  304 GG(2) = 1.
      GO TO 60
  305 GG(3) = 1.
      GO TO 60
  306 GG(1) = CTM(4)
      GO TO 60
  307 GG(2) = CTM(5)
      GO TO 60
  308 GG(3) = CTM(6)
      GO TO 60
  309 GG(1) = 1./SELC
      GO TO 60
  310 GG(3) = CELC/SELC
  311 GO TO 60
  350 SP(1)= SIN(C(N+13))
      SP(2)= COS(C(N+13))
      IF(JN-14) 352,354,356
  352 VO(15)=RERP2*(CPHDT(MTYP)/SP(2))**2
      VO(16)=-(ROT(MTYP)/REO)**2*(RERP2-1.)*SP(1)*SP(2)*ROT(MTYP)
C-------------CALCULATE EQ.(252) --------------------------
      XSC(1)= XS(3)*VO(15)-VO(6)-VO(16)*CCAPH(MTYP)
      XSC(2)= 0.
      XSC(3)=-XS(1)*VO(15)-VO(5)+ VO(16)*SCAPH(MTYP)
      GO TO 358
  354 XSC(1)=-XSX(1,3)
      XSC(2)=-XSX(2,3)
      XSC(3)=-XSX(3,3)
      GO TO 358
  356 XSC(1)=-CCAPH(MTYP)
      XSC(2)= 0.
      XSC(3)= SCAPH(MTYP)
C-------------CALCULATE EQ.(250) --------------------------
  358 RCC(1)=(XSC(1)*XS(1)+XSC(2)*XS(2)+XSC(3)*XS(3))/CTM(1)
      RCC(2)=(XSC(2)*XS(3)-XSC(3)*XS(2))/VO(9)
      RCC(3)=(XSC(1)-RCC(1)*SELC)/VO(8)
```

```
C-------------CALCULATE EQ.(249) --------------------------
      GG(1)= C(N+1)*RCC(1)-VO(13)*RCC(3)
      GG(2)= C(N+2)*RCC(2)
      GG(3)= VO(14)*RCC(3)
      IF(MO(MTYP)) 360,60,360
  360 IF(JN-14) 362,364,366
  362 SP(3)= VO(15)-1.
      SP(5)= XS(2)*DX(6)
C-------------CALCULATE EQ.(253A)--------------------------
      XSDC(1)= VO(10)*(XSC(1)+VO(16)*CCAPH(MTYP)-VO(6)*SP(3))
     X    +VO(15)*(XSX(3,2)*DX(5)+SP(5)*SPHDT(MTYP))
      XSDC(2)= 0.
      XSDC(3)= VO(10)*(XSC(3)-VO(16)*SCAPH(MTYP)-VO(5)*SP(3))
     X    +VO(15)*(-XSX(1,2)*DX(5)+SP(5)*CPHDT(MTYP))
      GO TO 368
  364 SP(1)= XSX(2,3)*DX(6)+XSX(2,2)*DX(5)
C-------------CALCULATE EQ.(253B)--------------------------
      XSDC(1)= VO(10)*XSC(1)+  SP(1)*CPHDT(MTYP)
      XSDC(2)= VO(10)*XSC(2)+   XS(2)*DX(6)+R*CTH*SPH*DX(5)
      XSDC(3)= VO(10)*XSC(3)-  SP(1)*SPHDT(MTYP)
      GO TO 368
  366 RCDC(1)= 0.
      RCDC(2)= 0.
      RCDC(3)= 0.
      GO TO 370
C-------------CALCULATE EQ.(251)---------------------------
  368 RCDC(1)= XSDC(1)*XS(1)+XSDC(2)*XS(2)+XSDC(3)*XS(3)
      RCDC(2)= XSDC(2)*XS(3)-XSDC(3)*XS(2)
      RCDC(3)= XSDC(1)*CTM(1)
  370 SP(1)= CTM(6)*XS(2)/CELC
      SP(2)= CTM(6)*XS(3)/CELC
      SP(3)= XSD(1)-CTM(6)*VO(8)
      SP(4)= CTM(1)*VO(8)
      RCDC(1)=(RCDC(1)+XSC(1)*XSD(1)+XSC(2)*XSD(2)+XSC(3)*XSD(3)
     X          -RCC(1)*CTM(4))/CTM(1)
      RCDC(2)=(RCDC(2)+XSC(2)*VO(11)+XSC(3)*VO(12))/VO(9)
      RCDC(3)=(RCDC(3)-RCDC(1)*XS(1)-XSC(1)*CTM(4)-XSC(2)*SP(1)
     X          -XSC(3)*SP(2)+RCC(1)*SP(3))/SP(4)
      GG(1)= GG(1)  + C(N+7)*RCDC(1)
      GG(2)= GG(2)  + C(N+8)*RCDC(2)
      GG(3)= GG(3)  + C(N+9)*RCDC(3)
      GO TO 60
  375 DO 390 I=1,3
      DO 385 J=1,NSTC
  385 G(I,J+10) = 0.
      DO 390 J=1,NPV
  390 H(I,J)= 0.
      RETURN
   80 DO 82 I=1,3
      SP(I)=SYG(I,MTYP)
      IF(NPC(1).NE.2)  SP(I)=SP(I)+SIG(I,KG)
   82 CONTINUE
   99 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OUTPUT
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *     NALL,LRK
      COMMON /TWICE/ A(3,3)     ,CPH       ,CT(5)     ,SP(10)    ,ST(5)
     1     ,TPH       ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)
     2     ,B(8)      ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1),DUE(4)
     3     ,DUF(4)    ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)
     4     ,IDN ,II   ,IP  ,IPC  ,JJ  ,JN   ,KA ,KD(3),KDUM ,KG2
     5     ,K1   ,NNN ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)
     6     ,RO        ,SPD(5)    ,SPH       ,TRAN(3,3),XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)      ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)
     1     ,AZ(40)    ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5)
     2     ,CVZ(30,5),D(5)       ,DAT(3,40),DATA(3)    ,DATAS(3)  ,DATC(3)
     3     ,DET       ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET
     4     ,FLOS      ,G(3,30)   ,H(3,5)    ,I ,ICOUNT,J      ,JNBR,JNBRS,JST
     5     ,K    ,KAR ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK
     6     ,KN   ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)     ,LCS(3)    ,LS     ,LT
     7     ,L1   ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT
     8     ,MTP(40)   ,MTYP      ,MTYPS     ,N ,NC(30),NCOUNT     ,NPC(15)
     9     ,NPU ,NPV  ,NS(9)     ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)
     A     ,N8(3)     ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)
     B     ,PA(3)                ,PM(3)     ,R         ,REO       ,RERP2
     C     ,RPO       ,ROT(5)    ,S(5)      ,SCAPH(5) ,SI(3)      ,SIG(3,40)
     D     ,SIGM(3)   ,SIGMS(3)  ,SPHDT(5)  ,SUM ,SUM2,SYG(3,9)   ,TFINAL
     E     ,TFIT(9)   ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)
     F     ,TR(40)    ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)
     G     ,TZERO     ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)
     H     ,XYCG(20)  ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
    2 CALL AUXIL
    4 ZO(1)=X(2)-R*OMEGA*CPH
      ZO(2)=SQRT(X(1)**2+ZO(1)**2)
      VO(1)=SQRT(ZO(2)**2+X(3)**2)
      IF(VO(1).EQ.0.) VO(1)=.00001
      ST(1)=-X(3)/VO(1)
      CT(1)= ZO(2)/VO(1)
      VO(2)=ATAN2(ST(1),CT(1))/CONRD
      IF(ZO(2).EQ.0.) ZO(2)=.00001
      ST(2)= ZO(1)/ZO(2)
      CT(2)= X(1)/ZO(2)
      VO(3)=ATAN2(ST(2),CT(2))/CONRD
      RO=REO/SQRT(1.-(1.-RERP2)*SPH**2)
      VO(4)= X(4)+REO-RO
      VO(5)=ATAN2(RERP2*SPH,CPH)/CONRD
      VO(6)= X(6)/CONRD
      SP(1)=SQRT(X(1)**2+X(2)**2)
      VO(10)=ATAN2(A(1,2),A(1,1))/CONRD
      SP(1) =SQRT(1.-A(1,3)**2)
      VO(11)=-ATAN2(A(1,3),SP(1))/CONRD
      VO(12)=ATAN2(A(2,3),A(3,3))/CONRD
      SP(1)=X(1)
      SP(2)=ZO(1)
      SP(3)=X(3)
      DO 10 I=1,3
      VO(I+12)=0.
      DO 10 J=1,3
```

```
   10 VO(I+12)=VO(I+12)+A(I,J)*SP(J)
      ZO(3)=SQRT(VO(13)**2+VO(15)**2)
      SP(1)=SQRT(ZO(3)**2+VO(14)**2)
      IF(SP(1).EQ.0.) SP(1)=.00001
      ST(4)=VO(14)/SP(1)
      CT(4)=ZO(3)/SP(1)
      VO(17)=ATAN2(ST(4),CT(4))/CONRD
      IF(ZO(3).EQ.0.) ZO(3)=.00001
      ST(5)= VO(15)/ZO(3)
      CT(5)= VO(13)/ZO(3)
      VO(18)=ATAN2(ST(5),CT(5))/CONRD
      ZO(4)=A(2,3)+ST(4)*ST(1)
      ZO(5)=(A(2,2)*CT(2)-A(2,1)*ST(2))*CT(1)
      ZO(9)=SQRT(ZO(4)**2+ZO(5)**2)
      IF(ZO(9).EQ.0.) ZO(9)=.000001
      ST(3)=ZO(4)/ZO(9)
      CT(3)=ZO(5)/ZO(9)
      VO(16)=ATAN2(ST(3),CT(3))/CONRD
      SP(1)=SQRT(VO(14)**2+VO(15)**2)
      SP(3)=ATAN2(SP(1),VO(1))/CONRD
      SP(2)=ATAN2(VO(14),VO(15))/CONRD
      SP(1) = SQRT(X(1)**2+X(2)**2+X(3)**2)
      VO(7) = SP(1)
      VO(19)=X(5)/CONRD
      VO(20)=SQRT(AG(1)**2+AG(2)**2+AG(3)**2)
      WRITE(OUT,1001)
 1001 FORMAT(//)
      WRITE(OUT,1000) TO
      WRITE(OUT,1002) (VO(I),I=1,6)
      WRITE(OUT,1004) X(1),X(2),X(3),VO(10),VO(11),VO(12)
      WRITE(OUT,1006) (VO(I),I=13,18)
      WRITE(OUT,1008) (XP(I),I=1,3),(SP(I),I=1,3)
      WRITE(OUT,1010) (PA(I),I=1,3),(AB(I),I=1,3)
      WRITE(OUT,1012) (PM(I),I=1,3),(AM(I),I=1,3)
      WRITE(OUT,1014) (X(I),I=7,10),VO(19),VO(20)
 1000 FORMAT(5H TIME,E15.8)
 1002 FORMAT(5H V(A),E15.8,5H G(A)E15.8,5H L(A)E15.8,5H ALT E15.8,5H LAT
     1DE15.8,5H LON E15.8)
 1004 FORMAT(5H U(I),E15.8,5H V(I)E15.8,5H W(I)E15.8,5H PSI E15.8,5H THE
     2 E15.8,5H PHI E15.8)
 1006 FORMAT(5H U(B),E15.8,5H V(B)E15.8,5H W(B)E15.8,5H SIG E15.8,5H BET
     3 E15.8,5H ALF E15.8)
 1008 FORMAT(5H XP  ,E15.8,5H YP  E15.8,5H ZP  E15.8,5H V(T)E15.8,5H XZI
     4 E15.8,5H ETA E15.8)
 1010 FORMAT(5H P   ,E15.8,5H Q   E15.8,5H R   E15.8,5H AX  E15.8,5H AY
     5 E15.8,5H AZ  E15.8)
 1012 FORMAT(5H PM  ,E15.8,5H QM  E15.8,5H RM  E15.8,5H AXM E15.8,5H AYM
     6 E15.8,5H AZM E15.8)
 1014 FORMAT(5H E0  ,E15.8,5H E1  E15.8,5H E2  E15.8,5H E3  E15.8,5H LAT
     7CE15.8,5H A(T)E15.8)
      IF(NPC(1)-1) 20,99,28
   20 IF(DCOMP) 28,22,22
   22 IF(NSTC) 28,28,24
   24 DO 26 I=1,NSTC
      K=NC(I)
   26 AG(I) = C(K)
      WRITE(OUT,1016) (NC(I),AG(I),I=1,NSTC)
 1016 FORMAT(6(2H C,I3,E15.8))
   28 IF(NPC(3)-1)  30,100,100
  100 DO 102 I=1,6
```

```
      DO 102 K=1,10
  102 DUB(K,I)=0.
      ZO(6)=VO(1)**2
      ZO(7)=ZO(6)*ZO(2)
      ZO(8)=OMEGA*CPH
      DUB(1,1)= X(1)/VO(1)
      DUB(2,1)= ZO(1)/VO(1)
      DUB(3,1)= X(3)/VO(1)
      DUB(4,1)= - DUB(2,1)*ZO(8)
      DUB(5,1)= -DUB(4,1)*R*TPH
      SP(1)= X(3)/(VO(1)*ZO(2))
      DUB(1,2)= SP(1)*DUB(1,1)
      DUB(2,2)= SP(1)*DUB(2,1)
      DUB(3,2)=-ZO(2)/ZO(6)
      DUB(4,2)= SP(1)*DUB(4,1)
      DUB(5,2)= SP(1)*DUB(5,1)
      SP(1)=ZO(2)**2
      DUB(1,3)=-ZO(1)/SP(1)
      DUB(2,3)= X(1)/SP(1)
      DUB(4,3)=-X(1)*ZO(8)/SP(1)
      DUB(5,3)=-DUB(4,3)*R*TPH
      IF(NPC(3)-1) 120,110,120
  110 SP(1)= 2./(A(1,1)**2+A(1,2)**2)
      DUB(7 ,4)=SP(1)*(X(10)*A(1,1)-X( 7)*A(1,2))
      DUB(8 ,4)=SP(1)*(X( 9)*A(1,1)-X( 8)*A(1,2))
      DUB(9 ,4)=SP(1)*(X( 8)*A(1,1)+X( 9)*A(1,2))
      DUB(10,4)=SP(1)*(X( 7)*A(1,1)+X(10)*A(1,2))
      SP(1)= 2./SQRT(1.-A(1,3)**2)
      DUB(7 ,5)= X( 9)*SP(1)
      DUB(8 ,5)=-X(10)*SP(1)
      DUB(9 ,5)= X( 7)*SP(1)
      DUB(10,5)=-X( 8)*SP(1)
      SP(1)= 2./(A(3,3)**2+A(2,3)**2)
      DUB(7 ,6)=SP(1)*(X( 8)*A(3,3)-X( 7)*A(2,3))
      DUB(8 ,6)=SP(1)*(X( 7)*A(3,3)+X( 8)*A(2,3))
      DUB(9 ,6)=SP(1)*(X(10)*A(3,3)+X( 9)*A(2,3))
      DUB(10,6)=SP(1)*(X( 9)*A(3,3)-X(10)*A(2,3))
      GO TO 200
  120 SP(1)= X(1)*X( 7) + ZO(1)*X(10) - X(3)*X( 9)
      SP(2)= X(1)*X( 8) + ZO(1)*X( 9) + X(3)*X(10)
      SP(3)= X(1)*X( 9) - ZO(1)*X( 8) + X(3)*X( 7)
      SP(4)= X(1)*X(10) - ZO(1)*X( 7) - X(3)*X( 8)
      SP(5)= ST(4)/(CT(4)*VO(1))
      SP(6)= 2./VO(1)
      DUB(1 ,5)= (CT(4)*A(2,1)-SP(5)*(X(1) -A(2,1)*VO(14)))/VO(1)
      DUB(2 ,5)= (CT(4)*A(2,2)-SP(5)*(ZO(1)-A(2,2)*VO(14)))/VO(1)
      DUB(3 ,5)= (CT(4)*A(2,3)-SP(5)*(X(3) -A(2,3)*VO(14)))/VO(1)
      DUB(4 ,5)= -ZO(8)*DUB(2,5)
      DUB(5 ,5)= -DUB(4,5)*R*TPH
      DUB(7 ,5)=(-CT(4)*SP(4)-SP(5)*(VO(13)*SP(1)+VO(15)*SP(3)))*SP(6)
      DUB(8 ,5)= (CT(4)*SP(3)-SP(5)*(VO(13)*SP(2)+VO(15)*SP(4)))*SP(6)
      DUB(9 ,5)= (CT(4)*SP(2)+SP(5)*(VO(13)*SP(3)-VO(15)*SP(1)))*SP(6)
      DUB(10,5)=(-CT(4)*SP(1)+SP(5)*(VO(13)*SP(4)-VO(15)*SP(2)))*SP(6)
      SP(6)= ZO(3)**2
      DUB(1 ,6) = (VO(13)*A(3,1)-VO(15)*A(1,1))/SP(6)
      DUB(2 ,6) = (VO(13)*A(3,2)-VO(15)*A(1,2))/SP(6)
      DUB(3 ,6) = (VO(13)*A(3,3)-VO(15)*A(1,3))/SP(6)
      DUB(4 ,6) = -ZO(8)*DUB(2,6)
      DUB(5 ,6) = -DUB(4,6)*R*TPH
      SP(6)= 2./SP(6)
```

```
      DUB(7 ,6) = (VO(13)*SP(3)-VO(15)*SP(1))*SP(6)
      DUB(8 ,6) = (VO(13)*SP(4)-VO(15)*SP(2))*SP(6)
      DUB(9 ,6) = (VO(13)*SP(1)+VO(15)*SP(3))*SP(6)
      DUB(10,6) = (VO(13)*SP(2)+VO(15)*SP(4))*SP(6)
      SP(1)= CT(3)*ST(4)*CT(1)+ST(3)*ST(1)*ZO(5)/CT(1)
      SP(2)= ST(3)*CT(1)*(A(2,2)*ST(2)+A(2,1)*CT(2))
      SP(3)= CT(3)*CT(4)*ST(1)
      DUB(1,4) = (SP(1)*DUB(1,2)+SP(2)*DUB(1,3)+SP(3)*DUB(1,5))/ZO(9)
      DUB(2,4) = (SP(1)*DUB(2,2)+SP(2)*DUB(2,3)+SP(3)*DUB(2,5))/ZO(9)
      DUB(3,4) = (SP(1)*DUB(3,2)+SP(2)*DUB(3,3)+SP(3)*DUB(3,5))/ZO(9)
      DUB(4,4) = (SP(1)*DUB(4,2)+SP(2)*DUB(4,3)+SP(3)*DUB(4,5))/ZO(9)
      DUB(5,4) = (SP(1)*DUB(5,2)+SP(2)*DUB(5,3)+SP(3)*DUB(5,5))/ZO(9)
      SP(6)= ST(3)*CT(1)
      SP(5)= SP(6)*CT(2)
      SP(6)= SP(6)*ST(2)
      DUB(7,4) =(SP(3)*DUB(7 ,5)+2.*(CT(3)*X( 8)-SP(5)*X( 7)-SP(6)*X(10)
     X))/ZO(9)
      DUB(8,4) =(SP(3)*DUB(8 ,5)+2.*(CT(3)*X( 7)+SP(5)*X( 8)+SP(6)*X( 9)
     X))/ZO(9)
      DUB(9,4) =(SP(3)*DUB(9 ,5)+2.*(CT(3)*X(10)-SP(5)*X( 9)+SP(6)*X( 8)
     X))/ZO(9)
      DUB(10,4)=(SP(3)*DUB(10,5)+2.*(CT(3)*X( 9)+SP(5)*X(10)-SP(6)*X( 7)
     X))/ZO(9)
  200 DO 205 I=1,3
      DO 205 J=1,NSTX
      DUB(J,I+6)=0.
      DUB(J,I+9)=P(I+3,J)
      DUB(J,I+12)=0.
      DO 202 K=1,5
      SUM=P(K,J)
      IF(K-J) 202,202,201
  201 SUM=P(J,K)
  202 DUB(J,I+6)=DUB(J,I+6)+DUB(K,I)*SUM
      DO 204 K=1,10
      SUM=P(K,J)
      IF(K-J) 204,204,203
  203 SUM=P(J,K)
  204 DUB(J,I+12)=DUB(J,I+12)+DUB(K,I+3)*SUM
  205 CONTINUE
      DO 218 I=1,3
      DO 208 J=I,3
      P(J+1,I)=0.
      P(J+7,I)=0.
      DO 206 K=1,5
  206 P(J+1,I)=P(J+1,I)+DUB(K,I+6)*DUB(K,J)
      DO 207 K=1,10
  207 P(J+7,I)=P(J+7,I)+DUB(K,I+6)*DUB(K,J+3)
  208 CONTINUE
      DO 212 J=I,3
      P(J+7,I+6)=0.
      P(J+4,I+3)=P(I+3,J+3)
      DO 212 K=1,10
  212 P(J+7,I+6)=P(J+7,I+6)+DUB(K,I+12)*DUB(K,J+3)
      DO 214 J=1,3
      P(J+4,I)= DUB(J+3,I+6)
  214 P(J+7,I+3)= DUB(I+3,J+12)
      DO 216 J=1,NSTC
      P(J+10,I)= DUB(J+10,I+7)
      P(J+10,I+6)= DUB(J+10,I+12)
  216 P(J+10,I+3)= P(I+3,J+10)
```

```
  218 CONTINUE
      DO 210 I=1,NSTC
      DO 210 J=I,NSTC
  210 P(J+10,I+9)= P(I+10,J+10)
      LT=1
      MT=NSTX-1
      NT=9
      GO TO 34
   30 LT=0
      MT=NSTX
      NT=10
      DO 32 I=1,NSTX
      K=I+1
      DO 32 J=K,NSTX
   32 P(J,I)=P(I,J)
   34 IF(NPC(1)) 35,130,35
  130 IF(NPC(4)-1) 35,132,35
  132 IF(NPC(3)) 134,134,138
  134 DO 136 I=1,10
  136 VO(I)= DZ(I)
      GO TO 146
  138 DO 144 I=1,3
      VO(I)=0.
      VO(I+6)=0.
      DO 140 J=1,5
  140 VO(I)=VO(I)+ DUB(J,I)*DZ(J)
      VO(I+3)= DZ(I+3)
      DO 142 J=1,10
  142 VO(I+6)=VO(I+6)+ DUB(J,I+3)*DZ(J)
  144 CONTINUE
  146 IF(NSTC) 148,147,148
  147 WRITE(OUT,1028) (I,VO(I),I=1,NT)
      GO TO 35
  148 DO 150 I=1,NSTC
  150 VO(I+NT)= DZ(I+10)
      WRITE(OUT,1028) (I,VO(I),I=1,NT),(NC(I),VO(I+NT),I=1,NSTC)
 1028 FORMAT(6(3H Z(I3,1H)E13.6))
   35 DO 40 I=1,MT
      K=I+LT
      IF(P(K,I)) 36,38,38
   36 WRITE(OUT,1030) I,I,I,P(   K,I)
      VO(I) = P(K,I)
 1030 FORMAT(26H ****NEGATIVE VARIANCE ON I3,31HRD TRANSFORMED VARIABLE,
     A SIGMA(I3,1H,I3,4H) = E15.8)
      GO TO 40
   38 VO(I)=SQRT(P(   K,I))
   40 CONTINUE
      IF(NSTC.GT.0) GO TO 50
      WRITE(OUT,1032) (I,VO(I),I=1,NT)
      GO TO 52
   50 DO 39 I=1,NSTC
      K=I+NT
   39 ZO(I)=VO(K)
      WRITE(OUT,1032) (I,VO(I),I=1,NT),(NC(I),ZO(I)    ,I=1,NSTC)
 1032 FORMAT(6(3H S(I3,1H)E13.6))
   52 IF(NPC(7).NE.ICOUNT) RETURN
      IF(NPC(6)-1) 99,41,42
   41 IF(DCOMP.GT.0)        RETURN
   42 WRITE(OUT,1033)
 1033 FORMAT(42H COVARIANCE MATRIX P (LOWER TRIANGLE ONLY) )
```

```
      DO 43 I=1,MT
   43 WRITE(OUT,1034) (P(I+LT,J),J=1,I)
 1034 FORMAT(X,12E10.3)
      IF(NPU) 44,44,250
   44 IF(NPV) 46,46,300
   46 CONTINUE
   99 RETURN
  250 IF(LT) 256,252,256
  252 DO 254 I=1,NSTX
      DO 254 J=1,NPU
  254 DUB(I,J+6)=CUZ(I,J)
      GO TO 266
  256 DO 262 I=1,3
      DO 262 J=1,NPU
      DUB(I,J+6)=0.
      DUB(I+6,J+6)=0.
      DO 258 K=1,5
  258 DUB(I,J+6)=DUB(I,J+6)+DUB(K,I)*CUZ(K,J)
      DO 260 K=1,10
  260 DUB(I+6,J+6)=DUB(I+6,J+6)+DUB(K,I+3)*CUZ(K,J)
  262 DUB(I+3,J+6)=CUZ(I+3,J)
      DO 264 I=1,NSTC
      DO 264 J=1,NPU
  264 DUB(I+9,J+6)=CUZ(I+10,J)
  266 WRITE(OUT,1035)
 1035 FORMAT(34H CORRELATION MATRIX (CUZ)TRANSPOSE)
      MT=NSTX-LT
      DO 268 J=1,NPU
  268 WRITE(OUT,1034) (DUB(I,J+6),I=1,MT)
      GO TO 44
  300 IF(LT) 306,302,306
  302 DO 304 I=1,NSTX
      DO 304 J=1,NPV
  304 DUB(I,J+6)=CVZ(I,J)
      GO TO 316
  306 DO 312 I=1,3
      DO 312 J=1,NPV
      DUB(I,J+6)=0.
      DUB(I+6,J+6)=0.
      DO 308 K=1,5
  308 DUB(I,J+6)=DUB(I,J+6)+DUB(K,I)*CVZ(K,J)
      DO 310 K=1,10
  310 DUB(I+6,J+6)=DUB(I+6,J+6)+DUB(K,I+3)*CVZ(K,J)
  312 DUB(I+3,J+6)=CVZ(I+3,J)
      DO 314 I=1,NSTC
      DO 314 J=1,NPV
  314 DUB(I+9,J+6)=CVZ(I+10,J)
  316 WRITE(OUT,1036)
 1036 FORMAT(34H CORRELATION MATRIX (CVZ)TRANSPOSE)
      MT=NSTX-LT
      DO 318 J=1,NPV
  318 WRITE(OUT,1034) (DUB(I,J+6),I=1,MT)
      GO TO 46
      END
```

```
      SUBROUTINE PRESET
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),      COMM0010
     *    NALL,LRK                                                        COMM0020
      COMMON /TWICE/ A(3,3)    ,CPH      ,CT(5)    ,SP(10)   ,ST(5)      COMM0030
     1    ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)    COMM0040
     2    ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)     COMM0050
     3    ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)       COMM0060
     4    ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2              COMM0070
     5    ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)      COMM0080
     6    ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)                COMM0090
      DIMENSION DUB(30,15)                                                COMM0100
      EQUIVALENCE (AG(1),DUB(1,1))                                        COMM0110
      COMMON /METH2/ AB(3)     ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)     COMM0120
     1    ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)   COMM0130
     2    ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)     COMM0140
     3    ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET          COMM0150
     4    ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J    ,JNBR,JNBRS,JST  COMM0160
     5    ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK    COMM0170
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT    COMM0180
     7    ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT          COMM0190
     8    ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)     COMM0200
     9    ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)   COMM0210
     A    ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)    COMM0220
     B    ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2       COMM0230
     C    ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)   COMM0240
     D    ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL      COMM0250
     E    ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)      COMM0260
     F    ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)    COMM0270
     G    ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)    COMM0280
     H    ,XYCG(20) ,XZCG(20)                                            COMM0290
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                    COMM0300
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                         COMM0310
      NDIMT = 160                                                         PRES0010
      NDIMZ = 30                                                          PRES0020
      NDIMV = 5                                                           PRES0030
      NDIMU = 5                                                           PRES0040
      DO 12 I=1,NDIMZ                                                     PRES0050
      DO 4  J=1,NDIMZ                                                     PRES0060
    4 P(I,J)=0.                                                           PRES0070
      DO 6  J=1,NDIMU                                                     PRES0080
    6 CUZ(I,J) = 0.                                                       PRES0090
      DO 8  J=1,NDIMV                                                     PRES0100
    8 CVZ(I,J) = 0.                                                       PRES0110
      DO 10 J=1,10                                                        PRES0120
   10 PH(J,I)=0.                                                          PRES0130
   12 DZ(I) = 0.                                                          PRES0140
      DO 16 I=1,NDIMU                                                     PRES0150
   16 D(I)=0.                                                             PRES0160
      DO 18 I=1,NDIMV                                                     PRES0170
   18 S(I)=0.                                                             PRES0180
      DO 26 I=1,NDIMT                                                     PRES0190
   26 C(I) = 0.                                                           PRES0200
      C(36) = 1.                                                          PRES0210
      C(40) = 1.                                                          PRES0220
      C(44) = 1.                                                          PRES0230
      C(61) = 1.                                                          PRES0240
      C(65) = 1.                                                          PRES0250
      C(69) = 1.                                                          PRES0260
      C(76) = 1.                                                          PRES0270
```

```
      C(77) = 1.
      C(78) = 1.
      C(91) = 1.
      C(92) = 1.
      C(93) = 1.
      C(106) = 1.
      C(107) = 1.
      C(108) = 1.
      C(121) = 1.
      C(122) = 1.
      C(123) = 1.
      C(136) = 1.
      C(137) = 1.
      C(138) = 1.
      CONRD =  .0174532925
      XJ2 = 1.082645E-3
      XMU = 3.985992E+14
      OMEGA = .7292116E-4
      RPO = 6356173.
      REO = 6378163.
      NCOUNT = 0
      ICOUNT = 1
      KDATAS = -1
      NST = 0
      NSTX = 0
      NSTC = 0
      NSTA = 0
      NPU = 0
      NPV = 0
      FIT = 1
      PQR = 2
      SCRACH = 3
      STATE = 4
      IN = 5
      OUT = 6
      N8(1) = 0
      N8(2) = 1
      N8(3) = 0
      N9(1) = 0
      N9(2) = 1
      N9(3) = 0
      N10(1) = 0
      N10(2) = 1
      N10(3) = 0
      N11(1) = 0
      N11(2) = 1
      N11(3) = 0
      DO 200 I=1,9
      DFIT(I) = 1.E10
      DTI(I) = 1.E10
  200 NTR(I) = 0
      TIME = 1.E+10
      KG = 2
      CALL INDAT
      CALL SETUP
   99 RETURN
      END
```

```
      SUBROUTINE PROP
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),     COMM0010
     *    NALL,LRK                                                      COMM0020
      COMMON /TWICE/ A(3,3)    ,CPH      ,CT(5)    ,SP(10)   ,ST(5)     COMM0030
     1    ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)   COMM0040
     2    ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)    COMM0050
     3    ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)      COMM0060
     4    ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2            COMM0070
     5    ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)     COMM0080
     6    ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)               COMM0090
      DIMENSION DUB(30,15)                                              COMM0100
      EQUIVALENCE (AG(1),DUB(1,1))                                      COMM0110
      COMMON /METH2/ AB(3)     ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)    COMM0120
     1    ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)  COMM0130
     2    ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)    COMM0140
     3    ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET         COMM0150
     4    ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J    ,JNBR,JNBRS,JST COMM0160
     5    ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK   COMM0170
     6    ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT   COMM0180
     7    ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT         COMM0190
     8    ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)    COMM0200
     9    ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)  COMM0210
     A    ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)   COMM0220
     B    ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2      COMM0230
     C    ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)  COMM0240
     D    ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL     COMM0250
     E    ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)     COMM0260
     F    ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)   COMM0270
     G    ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)   COMM0280
     H    ,XYCG(20) ,XZCG(20)                                           COMM0290
      INTEGER OUT,FIT,STATE,SCRACH,PGR                                  COMM0300
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PGR                       COMM0310
C------------CALCULATE EQ.(55B) -------------------------               PROP0010
      DO 4 I=1,10                                                       PROP0020
      DO  4 J=1,NSTX                                                    PROP0030
      SUM = 0.                                                          PROP0040
      L = J+1                                                           PROP0050
      DO 2 K=1,J                                                        PROP0060
    2 SUM = SUM + PH(I,K)*P(K,J)                                        PROP0070
      IF(L.GT.NSTX) GO TO 4                                             PROP0080
      DO 3 K=L,NSTX                                                     PROP0090
    3 SUM = SUM + PH(I,K)*P(J,K)                                        PROP0100
    4 DUB(J,I) = SUM                                                    PROP0110
      DO 10 I=1,10                                                      PROP0120
      DO   8 J=I,10                                                     PROP0130
      SUM = 0.                                                          PROP0140
      DO   6 K=1,NSTX                                                   PROP0150
    6 SUM = SUM + DUB(K,I)*PH(J,K)                                      PROP0160
    8 P(I,J) = SUM                                                      PROP0170
   10 CONTINUE                                                          PROP0180
      IF(NSTX-10) 12,16,12                                              PROP0190
   12 DO 14 I=11,NSTX                                                   PROP0200
      DO 14 J=1,10                                                      PROP0210
   14 P(J,I) = DUB(I,J)                                                 PROP0220
      IF(NPU) 38,38,60                                                  PROP0230
   60 DO 64 I=1,10                                                      PROP0240
      DO 64 K=11,NSTX                                                   PROP0250
      SUM = 0.                                                          PROP0260
      DO 62 J=1,NPU                                                     PROP0270
```

```
      II=J+NSTX
   62 SUM = SUM + PH(I,II)*CUZ(K,J)
   64 P(I,K) = P(I,K) + SUM
      GO TO 20
   16 IF(NPU) 38,38,20
C-------------CALCULATE EQ.(55C) -------------------------
   20 DO 25 J=1,NPU
      DO 24 I=1,10
      SUM = 0.
      DO 22 K=1,NSTX
      SUM = SUM + PH(I,K)*CUZ(K,J)
   22 CONTINUE
      DUB(I,J) = SUM
   24 DUB(I,15) = SUM + PH(I,J+NSTX)*D(J)
      DO 25 I=1,10
   25 CUZ(I,J) = DUB(I,15)
   26 CONTINUE
   30 DO 34 I=1,10
      DO 34 J=I,10
      SUM = 0.
      DO 32 K=1,NPU
   32 SUM = SUM + CUZ(I,K)*PH(J,K+NSTX) + PH(I,K+NSTX)*DUB(J,K)
   34 P(I,J) = P(I,J) + SUM
   36 CONTINUE
   38 IF(NPV) 99,99,40
C-------------CALCULATE EQ.(55D) -------------------------
   40 DO 48 J=1,NPV
      DO 44 I=1,10
      SUM = 0.
      DO 42 K=1,NSTX
   42 SUM = SUM + PH(I,K)*CVZ(K,J)
   44 DUB(I,1) = SUM
      DO 46 I=1,10
   46 CVZ(I,J) = DUB(I,1)
   48 CONTINUE
   99 IF(NPC(4).EQ.0) GO TO 75
      IF(NPC(1).EQ.2) GO TO 75
C-------------CALCULATE EQ.(55A) -------------------------
      DO 72 I=1,10
      SUM = 0.
      DO 70 J=1,NSTX
   70 SUM = SUM + PH(I,J)*DZ(J)
   72 DUB(I,1) = SUM
      DO 74 I=1,10
   74 DZ(I) = DUB(I,1)
   75 CONTINUE
      ENTRY IDENT
      K = NSTX+NPU
      DO 52 I=1,10
      DO 50 J=1,K
   50 PH(I,J)=0.
   52 PH(I,I)=1.
      RETURN
      END
```

```
      SUBROUTINE RKUTTA
      DIMENSION PR(3)
      DIMENSION F1(410),F2(410),F3(410),DELY(410)
      COMMON /INTGRL/ P,T,TP,Y(410),DY(410),N,L
      DATA PR /.5,.25,.5/
      GO TO (1,2,2,4),L
    1 IF(IG.EQ.0) GO TO 30
      DO 10 I = 1,N
10    F3(I)=Y(I)
      L = 2
      RETURN
    2 DT = P
      J = 1
      T2 = TP - T
      RT2=T2/P
      IF(RT2-.99999)22,26,29
22    IF(RT2.GT..00001) GO TO 26
20    L=3
      RETURN
26    DT=T2
29    IG=0
30    GO TO(31,32,33,34),J
31    T = T+PR(1)*DT
      DO 310 I=1,N
      F1(I)=DY(I)*DT
310   Y(I)=F3(I)+PR(1)*F1(I)
      J = J+1
      GO TO 38
33    T = T+PR(1)*DT
32    T2=DT/PR(1)
      DO 320 I=1,N
      RT2=DY(I)*T2
      F1(I)=F1(I)+RT2
320   Y(I)=F3(I)+PR(J)*RT2
35    J = J + 1
      RETURN
34    DO 340 I=1,N
340   F1(I)=F1(I)+DY(I)*DT
   37 DO 39 I = 1,N
      DELY(I)=F1(I)/6.+DELY(I)
39    Y(I)=F2(I)+DELY(I)
      IG = 1
      RETURN
    4 DT=P
      IG = 1
      J = 1
      DO 40  I = 1,N
      DELY(I) = 0.D0
40    F2(I)=Y(I)
   38 L = 1
      RETURN
      END
```

```
      SUBROUTINE SETUP
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),     COMM0010
     *     NALL,LRK                                                     COMM0020
      COMMON /TWICE/ A(3,3)   ,CPH       ,CT(5)    ,SP(10)   ,ST(5)     COMM0030
     1      ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4) COMM0040
     2      ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)  COMM0050
     3      ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)    COMM0060
     4      ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2          COMM0070
     5      ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)   COMM0080
     6      ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)             COMM0090
      DIMENSION DUB(30,15)                                              COMM0100
      EQUIVALENCE (AG(1),DUB(1,1))                                      COMM0110
      COMMON /METH2/ AB(3)    ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)     COMM0120
     1      ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5)COMM0130
     2      ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)  COMM0140
     3      ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET       COMM0150
     4      ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J   ,JNBR,JNBRS,JSTCOMM0160
     5      ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK COMM0170
     6      ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT COMM0180
     7      ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT       COMM0190
     8      ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)  COMM0200
     9      ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)COMM0210
     A      ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30) COMM0220
     B      ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2    COMM0230
     C      ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40)COMM0240
     D      ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL   COMM0250
     E      ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)   COMM0260
     F      ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20) COMM0270
     G      ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20) COMM0280
     H      ,XYCG(20) ,XZCG(20)                                         COMM0290
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                  COMM0300
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                       COMM0310
      DIMENSION DUA(4,2)                                                SETP0010
      WRITE(OUT,1000)                                                   SETP0020
 1000 FORMAT(1H1,27H CONTROLS ARE SPECIFIED FOR)                        SETP0030
      IF(NPC(1)-1) 10,11,12                                             SETP0040
   10 WRITE(OUT,1001) TO,TFINAL                                         SETP0050
 1001 FORMAT(5X,28H*FILTERING RUN FROM TIME TO=F9.4,11H TO TFINAL=F9.4) SETP0060
      GO TO 13                                                          SETP0070
   11 WRITE(OUT,1002) TO,TFINAL                                         SETP0080
 1002 FORMAT(5X,32H*DETERMINISTIC RUN FROM TIME TO=F9.4,11H TO TFINAL=F9SETP0090
     1.4)                                                               SETP0100
      GO TO 24                                                          SETP0110
   12 WRITE(OUT,1003) TO,TFINAL                                         SETP0120
 1003 FORMAT(5X,33H*ERROR ANALYSIS RUN FROM TIME TO=F9.4,11H TO TFINAL=FSETP0130
     19.4)                                                              SETP0140
      GO TO 21                                                          SETP0150
   13 IF(NPC(4))14,14,15                                                SETP0160
   14 WRITE(OUT,1004)                                                   SETP0170
 1004 FORMAT(5X,18H*UPDATED REFERENCE)                                  SETP0180
      GO TO 16                                                          SETP0190
   15 WRITE(OUT,1005)                                                   SETP0200
 1005 FORMAT(5X,21H*NONUPDATED REFERENCE)                               SETP0210
   16 IF(NPC(8)-1) 17,18,19                                             SETP0220
   17 WRITE(OUT,1006)                                                   SETP0230
 1006 FORMAT(5X,57H*SMOOTH DETERMINISTICALLY, NO RESIDUALS NOR LOSS FUNCSETP0240
     XTION)                                                             SETP0250
      GO TO 20                                                          SETP0260
   18 WRITE(OUT,1007)                                                   SETP0270
```

```
1007 FORMAT(5X,64H*SMOOTH DETERMINISTICALLY, CALCULATE RESIDUALS AND LOSETP0280
    XSS FUNCTION)                                                       SETP0290
     GO TO 20                                                           SETP0300
  19 IF(NPC(8)-3) 170,172,174                                           SETP0310
 170 WRITE(OUT,1027)                                                    SETP0320
1027 FORMAT(5X,49H*SMOOTH COVARIANCE,NO RESIDUALS NOR LOSS FUNCTION)    SETP0330
     GO TO 20                                                           SETP0340
 172 WRITE(OUT,1028)                                                    SETP0350
1028 FORMAT(5X,56H*SMOOTH COVARIANCE,CALCULATE RESIDUALS AND LOSS FUNCTSETP0360
    XION)                                                               SETP0370
     GO TO 20                                                           SETP0380
 174 WRITE(OUT,1029)                                                    SETP0390
1029 FORMAT(5X,13H*NO SMOOTHING)                                        SETP0400
  20 WRITE(OUT,1009) NPC(7)                                             SETP0410
1009 FORMAT(5X,12H*FILTER FOR I2,11H ITERATIONS)                        SETP0420
  21 IF(NPC(9).EQ.0) GO TO 23                                           SETP0430
  22 WRITE(OUT,1010)                                                    SETP0440
1010 FORMAT(5X,28H*SCALAR PROCESS FITTING DATA)                         SETP0450
     GO TO 24                                                           SETP0460
  23 WRITE(OUT,1011)                                                    SETP0470
1011 FORMAT(5X,36H*VECTOR PROCESS FITTING DATA TRIPLES)                 SETP0480
  24 IF(NPC(2)) 25,25,26                                                SETP0490
  25 WRITE(OUT,1012)                                                    SETP0500
1012 FORMAT(5X,52H*INPUT AND OUTPUT IN METRIC UNITS(M,M/SEC,M/SEC2,KG))SETP0510
     GO TO 27                                                           SETP0520
  26 WRITE(OUT,1013)                                                    SETP0530
1013 FORMAT(5X,62H*INPUT AND OUTPUT IN ENGLISH UNITS (FT,FT/SEC,FT/SEC2SETP0540
    X,LB,SLUG))                                                         SETP0550
  27 WRITE(OUT,1014) REO,RPO,OMEGA,XMU,XJ2                              SETP0560
1014 FORMAT(5X,18H*PLANET PARAMETERS/20X,11HEQUIT.RAD.=E15.8,13H  POLARSETP0570
    1 RAD.=E15.8,13H  ROTATION  =E15.8/20X,11HMU(GRAV.) =E15.8,13H  J2(SETP0580
    2GRAV.) =E15.8/5X,26H*INITIAL STATE ESTIMATE IS)                    SETP0590
     IF(NPC(3)-1) 28,29,30                                              SETP0600
  28 WRITE(OUT,1015) (ZO(I),I=1,10)                                     SETP0610
1015 FORMAT(20X,1HU,9X,1H=E15.8,3H  V,9X,1H=E15.8,3H  W,9X,1H=E15.8/20XSETP0620
    1,11HALTITUDE  =E15.8,13H  GEOD.LAT. =E15.8,13H  LONGITUDE =E15.8/2SETP0630
    20X,2HE0,8X,1H=E15.8,4H  E1,8X,1H=E15.8,4H  E2,8X,1H=E15.8/20X,2HE3SETP0640
    3,8X,1H=E15.8)                                                      SETP0650
     GO TO 31                                                           SETP0660
  29 WRITE(OUT,1016) (ZO(I),I=1,9)                                      SETP0670
1016 FORMAT(20X,11HVELOCITY  =E15.8,13H  GAMMA     =E15.8,13H  LAMBDA  SETP0680
    1  =E15.8/20X,11HALTITUDE  =E15.8,13H  GEOD.LAT. =E15.8,13H  LONGITSETP0690
    2UDE =E15.8/20X,11HPSI(BAR)  =E15.8,13H  THETA(BAR)=E15.8,13H  PHI(SETP0700
    3BAR)  =E15.8)                                                      SETP0710
     GO TO 31                                                           SETP0720
  30 WRITE(OUT,1017) (ZO(I),I=1,9)                                      SETP0730
1017 FORMAT(20X,11HVELOCITY  =E15.8,13H  GAMMA     =E15.8,13H  LAMBDA  SETP0740
    1  =E15.8/20X,11HALTITUDE  =E15.8,13H  GEOD.LAT. =E15.8,13H  LONGITSETP0750
    2UDE =E15.8/20X,11HSIGMA     =E15.8,13H  BETA      =E15.8,13H  ALPHSETP0760
    3A     =E15.8)                                                      SETP0770
  31 RERP2=(REO/RPO)**2                                                 SETP0780
     XMUJ = 3.*XMU*XJ2*REO**2                                           SETP0790
     IF(NPC(1)-1)32,40,32                                               SETP0800
  32 WRITE(OUT,1018) NSTA                                               SETP0810
1018 FORMAT(5X,22H*FITTING DATA IS FROM I2,8H SOURCES)                  SETP0820
     KSS=0                                                              SETP0830
     DO 42 I=1,NSTA                                                     SETP0840
     IF(NS(I)-6)  33,44,45                                              SETP0850
  33 MT = NS(I)                                                         SETP0860
     N = 73 +15*MT                                                      SETP0870
```

```
      WRITE(OUT,1019) I,MT,C(N),C(N+1),C(N+2)
 1019 FORMAT(10X,I1,19H. TRACKING STATION I1,12H, GEOD.LAT.=F11.6,12H  L
     XONGITUDE=F11.6,12H  ALTITUDE =F11.4,19H ABOVE REF. SURFACE)
      C(N)=C(N)*CONRD
      C(N+1)=C(N+1)*CONRD
      C(N) = ATAN2(SIN(C(N)),RERP2*COS(C(N)))
      NT=N+2
      MTYP = MT
      CALL STAT
      DO 36 J=1,NSTC
      IF(NC(J)-N) 36,35,35
   35 IF(NC(J)-NT)37,37,36
   36 CONTINUE
      GO TO 38
   37 KSS=KSS+1
      NSS(KSS) = MT
   38 MO(MT) = 0
      DO 39 J=1,NSTC
      IF(NC(J).LT.N-6) GO TO 39
      IF(NC(J).LE.N-4) GO TO 40
   39 CONTINUE
      IF(C(N-6).NE.0.) GO TO 40
      IF(C(N-5).NE.0.) GO TO 40
      IF(C(N-4).NE.0.) GO TO 40
      GO TO 42
   44 WRITE(OUT,1020) I,C(151),C(152)
 1020 FORMAT(10X,I1,27H. AIRBORNE RADAR, DELTA(P)=F11.6,11H  DELTA(Y)=F1
     X1.6)
      GO TO 42
   45 IF(NS(I)-7) 41,41,43
   41 WRITE(OUT,1031) I
      GO TO 42
   43 WRITE(OUT,1030) I
      GO TO 42
 1030 FORMAT(I11,15H. POSITION DATA)
 1031 FORMAT(I11,15H. VELOCITY DATA)
   40 MO(MT) = 1
   42 CONTINUE
   46 ZO(5) = ZO(5)*CONRD
      ZO(6)=ZO(6)*CONRD
      NSTX = NSTC + 10
      ZO(5) = ATAN2(SIN(ZO(5)),RERP2*COS(ZO(5)))
      CPH=COS(ZO(5))
      SPH=SIN(ZO(5))
      RO =REO/SQRT(1.+(RERP2-1.)*SPH**2)
      R= ZO(4)+RO
      ZO(4)=R-REO
      DO 48 I=1,10
   48 X(I)=ZO(I)
      IF(NPC(3).EQ.0) GO TO 100
   50 DO 51 I=7,9
   51 ZO(I)=ZO(I)*CONRD
      DO 52 I=1,2
      ZO(I+1) = ZO(I+1)*CONRD
      ST(I)=SIN(ZO(I+1))
   52 CT(I)=COS(ZO(I+1))
      DO 53 I=3,5
      ST(I)=SIN(ZO(I+4)/2.)
   53 CT(I)=COS(ZO(I+4)/2.)
      X(3)=ZO(1)*CT(1)
```

```
      X(1)=X(3)*CT(2)
      X(2)=X(3)*ST(2)+R*OMEGA*CPH
      X(3)=-ZO(1)*ST(1)
      DO 54 I=1,10
      DO 54 J=1,9
   54 DUB(I,J)=0.
      DUB(1,1)= CT(1)*CT(2)
      DUB(2,1)= CT(1)*ST(2)
      DUB(3,1)=-ST(1)
      SP(1)= ZO(1)*ST(1)
      DUB(1,2)=-SP(1)*CT(2)
      DUB(2,2)=-SP(1)*ST(2)
      SP(1)= ZO(1)*CT(1)
      DUB(3,2)=-SP(1)
      DUB(1,3)=-SP(1)*ST(2)
      DUB(2,3)= SP(1)*CT(2)
      DUB(2,4)= OMEGA*CPH
      DUB(2,5)=-R*OMEGA*SPH
      DUB(4,4)=1.
      DUB(5,5)=1.
      DUB(6,6)=1.
      IF(NPC(3)-1) 100,56,60
   56 DUA(1,1)= CT(3)*CT(4)*CT(5)
      DUA(2,1)= CT(3)*CT(4)*ST(5)
      DUA(3,1)= CT(3)*ST(4)*CT(5)
      DUA(4,1)= ST(3)*CT(4)*CT(5)
      DUA(1,2)= ST(3)*ST(4)*ST(5)
      DUA(2,2)= ST(3)*ST(4)*CT(5)
      DUA(3,2)= ST(3)*CT(4)*ST(5)
      DUA(4,2)= CT(3)*ST(4)*ST(5)
      X(7) = DUA(1,1)+DUA(1,2)
      X(8) = DUA(2,1)-DUA(2,2)
      X(9) = DUA(3,1)+DUA(3,2)
      X(10)= DUA(4,1)-DUA(4,2)
      IF(NPC(1).EQ.1) GO TO 100
      DUB(7,7) = -X(10)/2.
      DUB(8,7) =-X(9)/2.
      DUB(9,7) = X(8)/2.
      DUB(10,7) = X(7)/2.
      DUB(7,8) =-(DUA(3,1) - DUA(3,2))/2.
      DUB(8,8) =-(DUA(4,1) + DUA(4,2))/2.
      DUB(9,8) = (DUA(1,1) - DUA(1,2))/2.
      DUB(10,8)=-(DUA(2,1) + DUA(2,2))/2.
      DUB(7,9) =-X(8)/2.
      DUB(8,9) = X(7)/2.
      DUB(9,9) = X(10)/2.
      DUB(10,9)=-X(9)/2.
      GO TO 70
   60 ST(1) = SIN(.5*ZO(2))
      CT(1) = COS(.5*ZO(2))
      ST(2) = SIN(.5*ZO(3))
      CT(2) = COS(.5*ZO(3))
      DUE(1)= CT(1)*CT(2)
      DUE(2)=-ST(1)*ST(2)
      DUE(3)= ST(1)*CT(2)
      DUE(4)= CT(1)*ST(2)
      DUA(1,1)= CT(3)*CT(4)*CT(5)
      DUA(2,1)= ST(3)*CT(4)*CT(5)
      DUA(3,1)= CT(3)*CT(4)*ST(5)
      DUA(4,1)= ST(3)*CT(4)*ST(5)
```

```
      DUA(1,2)= ST(3)*ST(4)*ST(5)                                       SETP2080
      DUA(2,2)= CT(3)*ST(4)*ST(5)                                       SETP2090
      DUA(3,2)= ST(3)*ST(4)*CT(5)                                       SETP2100
      DUA(4,2)= CT(3)*ST(4)*CT(5)                                       SETP2110
      DUF(1)= DUA(1,1)-DUA(1,2)                                         SETP2120
      DUF(2)= DUA(2,1)+DUA(2,2)                                         SETP2130
      DUF(3)= DUA(3,1)+DUA(3,2)                                         SETP2140
      DUF(4)= DUA(4,1)-DUA(4,2)                                         SETP2150
      X(7) = DUE(1)*DUF(1)-DUE(2)*DUF(2)-DUE(3)*DUF(3)-DUE(4)*DUF(4)    SETP2160
      X(8) = DUE(1)*DUF(2)+DUE(2)*DUF(1)+DUE(3)*DUF(4)-DUE(4)*DUF(3)    SETP2170
      X(9) = DUE(1)*DUF(3)-DUE(2)*DUF(4)+DUE(3)*DUF(1)+DUE(4)*DUF(2)    SETP2180
      X(10)= DUE(1)*DUF(4)+DUE(2)*DUF(3)-DUE(3)*DUF(2)+DUE(4)*DUF(1)    SETP2190
      IF(NPC(1).EQ.1) GO TO 100                                         SETP2200
      DUB(7,2) = -X(9)/2. - DUE(2)*DUF(4) + DUE(4)*DUF(2)               SETP2210
      DUB(8,2) =  X(10)/2.- DUE(2)*DUF(3) - DUE(4)*DUF(1)               SETP2220
      DUB(9,2) =  X(7)/2. + DUE(2)*DUF(2) + DUE(4)*DUF(4)               SETP2230
      DUB(10,2)= -X(8)/2. + DUE(2)*DUF(1) - DUE(4)*DUF(3)               SETP2240
      DUB(7,3) = -X(10)/2.                                              SETP2250
      DUB(8,3) = -X(9)/2.                                               SETP2260
      DUB(9,3) =  X(8)/2.                                               SETP2270
      DUB(10,3)=  X(7)/2.                                               SETP2280
      DUB(7,7) = -X(8)/2. + DUE(3)*DUF(4) - DUE(4)*DUF(3)               SETP2290
      DUB(8,7) =  X(7)/2. + DUE(3)*DUF(3) + DUE(4)*DUF(4)               SETP2300
      DUB(9,7) = -X(10)/2.- DUE(3)*DUF(2) + DUE(4)*DUF(1)               SETP2310
      DUB(10,7)=  X(9)/2. - DUE(3)*DUF(1) - DUE(4)*DUF(2)               SETP2320
      DUF(1) = DUA(1,1)+DUA(1,2)                                        SETP2330
      DUF(2) = DUA(2,1)-DUA(2,2)                                        SETP2340
      DUF(3) = DUA(3,1)-DUA(3,2)                                        SETP2350
      DUF(4) = DUA(4,1)+DUA(4,2)                                        SETP2360
      DUB(7,8) = (-DUE(1)*DUF(4)-DUE(2)*DUF(3)-DUE(3)*DUF(2)+DUE(4)*DUF(SETP2370
     X1))/2.                                                            SETP2380
      DUB(8,8) =  (DUE(1)*DUF(3)-DUE(2)*DUF(4)-DUE(3)*DUF(1)-DUE(4)*DUF(SETP2390
     X2))/2.                                                            SETP2400
      DUB(9,8) =  (DUE(1)*DUF(2)+DUE(2)*DUF(1)-DUE(3)*DUF(4)+DUE(4)*DUF(SETP2410
     X3))/2.                                                            SETP2420
      DUB(10,8)= (-DUE(1)*DUF(1)+DUE(2)*DUF(2)-DUE(3)*DUF(3)-DUE(4)*DUF(SETP2430
     X4))/2.                                                            SETP2440
      DUB(7,9) = -X(9)/2.                                               SETP2450
      DUB(8,9) = -X(10)/2.                                              SETP2460
      DUB(9,9) =  X(7)/2.                                               SETP2470
      DUB(10,9)=  X(8)/2.                                               SETP2480
   70 DO 74 I=1,10                                                      SETP2490
      DO 74 J=1,NSTX                                                    SETP2500
      SUM = 0.                                                          SETP2510
      DO 72 K=1,9                                                       SETP2520
   72 SUM = SUM + DUB(I,K)*P(K,J)                                       SETP2530
   74 PH(I,J) = SUM                                                     SETP2540
      DO 80 I=1,10                                                      SETP2550
      DO 78 J=I,10                                                      SETP2560
      SUM = 0.                                                          SETP2570
      DO 76 K=1,10                                                      SETP2580
   76 SUM = SUM + PH(I,K)*DUB(J,K)                                      SETP2590
   78 P(I,J) = SUM                                                      SETP2600
      DO 80 J=11,NSTX                                                   SETP2610
   80 P(I,J) = PH(I,J)                                                  SETP2620
      IF(NPU.EQ.0) GO TO 88                                             SETP2630
      DO 84 I=1,10                                                      SETP2640
      DO 84 J=1,NPU                                                     SETP2650
      SUM = 0.                                                          SETP2660
      DO 82 K=1,9                                                       SETP2670
```

```
   82 SUM = SUM + DUB(I,K)*CUZ(K,J)
   84 PH(I,J) = SUM
      DO 86 I=1,10
      DO 86 J=1,NPU
   86 CUZ(I,J) = PH(I,J)
   88 IF(NPV.EQ.0) GO TO 100
      DO 92 I=1,10
      DO 92 J=1,NPV
      SUM = 0.
      DO 90 K=1,9
   90 SUM = SUM + DUB(I,K)*CVZ(K,J)
   92 PH(I,J) = SUM
      DO 94 I=1,10
      DO 94 J=1,NPV
   94 CVZ(I,J) = PH(I,J)
  100 WRITE(OUT,3021)
      WRITE(OUT,1021) (I,I,X(I),P(I,I),I=1,10)
 3021 FORMAT(//,24H STATE VECTOR COMPONENTS/43H COMP.   ID.NO.      EST.
     *VALUE      VARIANCE)
 1021 FORMAT(1X,I4,4X,I4,5X,2E15.8)
      IF(NSTC) 102,104,102
  102 WRITE(OUT,1022)
 1022 FORMAT(42H MODEL PARAMETERS IN EXPANDED STATE VECTOR)
      DO 103 I=1,NSTC
      LT=NC(I)
      MT=I+10
  103 WRITE(OUT,1023) MT,LT,C(LT),P(MT,MT)
 1023 FORMAT(1X,I4,4X,I4,5X,2E15.8)
  104 IF(NPU) 106,108,106
  106 WRITE(OUT,1024)
 1024 FORMAT(55H RANDOM MODEL PARAMETERS NOT BEING ESTIMATED (U-VECTOR))
      DO 107 I=1,NPU
      LT=MC(I)
  107 WRITE(OUT,1023) I,LT,C(LT),D(I)
  108 IF(NPV) 110,112,110
  110 WRITE(OUT,1025)
 1025 FORMAT(61H RANDOM MEASUREMENT PARAMETERS NOT BEING ESTIMATED (V-VE
     XCTOR))
      DO 111 I=1,NPV
      LT=MCC(I)
  111 WRITE(OUT,1023) I,LT,C(LT),S(I)
  112 LT=9
      IF(NPC(3).EQ.0) LT=10
      IF(NST-LT) 114,120,114
  114 WRITE(OUT,1026)
 1026 FORMAT(48H IMPROPER NUMBER OF STATE VARIABLES ARE INPUTTED)
      CALL EXIT
  120 IF(NPC(1)-1) 124,122,124
  122 NALL=10
      GO TO 128
  124 NALL = 10*(NSTX+NPU+1)
      K = NSTX + NPU
      DO 126 I =1,10
      DO 125 J=1,K
  125 PH(I,J) = 0.
  126 PH(I,I) = 1.
  128 IF(NPC(1)-2) 144,130,144
  130 IF(NSTA-1) 144,144,132
  132 DO 142 I=2,NSTA
      MT=NS(I-1)
```

```
      TT=DTF(MT)                                                        SETP4490
      LT=I-1                                                            SETP4500
      DO 136 J=I,NSTA                                                   SETP4510
      K=NS(J)                                                           SETP4520
      IF(TT-DTF(K)) 135,135,136                                         SETP4530
  135 TT=DTF(K)                                                         SETP4540
      LT=J                                                              SETP4550
  136 CONTINUE                                                          SETP4560
  137 IP=NS(I-1)                                                        SETP4570
  138 NS(I-1)=NS(LT)                                                    SETP4580
  139 NS(LT)=IP                                                         SETP4590
  142 CONTINUE                                                          SETP4600
  144 KDATA=-1                                                          SETP3550
      WRITE(SCRACH) NSTX,NPU,NPV,((P(I,J),I=1,NSTX),J=1,NSTX),          SETP3560
     1     ((CUZ(I,J),I=1,NSTX),J=1,NPU),((CVZ(I,J),I=1,NSTX),J=1,NPV)  SETP3570
      ET = TO                                                           SETP3580
      TZERO = TO                                                        SETP3590
      T = TO                                                            SETP3600
      TIMES=TZERO-10.                                                   SETP3610
      KDAP = 1                                                          SETP3620
      KPROP=-1                                                          SETP3630
      KDATAS=-1                                                         SETP3640
      C(151)= C(151)*CONRD                                              SETP3650
      C(152)= C(152)*CONRD                                              SETP3660
      RETURN                                                            SETP3670
      END                                                               SETP3680
```

```
      SUBROUTINE SMOOTH
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),
     *    NALL,LRK
      COMMON /TWICE/ A(3,3)     ,CPH       ,CT(5)     ,SP(10)    ,ST(5)
     1    ,TPH       ,VO(20)    ,ZO(20)    ,AG(3)     ,AMDOT(3)  ,APDOT(4)
     2    ,B(8)      ,CGM(3)    ,DFM(3)    ,DTRAN(3,3),DUD(10,1) ,DUE(4)
     3    ,DUF(4)    ,F1(3,40)  ,F2(3,5)   ,F3(4,40)  ,GG(3)     ,HI(2)
     4    ,IDN ,II   ,IP  ,IPC  ,JJ  ,JN  ,KA  ,KD(3) ,KDUM ,KG2
     5    ,K1  ,NNN  ,NPTS      ,PAR(6)    ,PDOT(3)   ,PMDOT(3)  ,RES(6)
     6    ,RO        ,SPD(5)    ,SPH       ,TRAN(3,3) ,XX(50)
      DIMENSION DUB(30,15)
      EQUIVALENCE (AG(1),DUB(1,1))
      COMMON /METH2/ AB(3)      ,AM(3)     ,AP(4)     ,AX(40)    ,AY(40)
     1    ,AZ(40)    ,C(160)    ,CCAPH(5)  ,CONRD     ,CPHDT(5)  ,CUZ(30,5)
     2    ,CVZ(30,5) ,D(5)      ,DAT(3,40) ,DATA(3)   ,DATAS(3)  ,DATC(3)
     3    ,DET       ,DFIT(9)   ,DTF(9)    ,DTI(9)    ,DZ(30)    ,ET
     4    ,FLOS      ,G(3,30)   ,H(3,5)    ,I ,ICOUNT ,J    ,JNBR,JNBRS,JST
     5    ,K    ,KAR ,KC(3)     ,KDAP      ,KDATA     ,KDATAS    ,KG,KI,KK
     6    ,KN   ,KOB ,KPROP,KS,KSM,KSS,L   ,LC(3)     ,LCS(3)    ,LS   ,LT
     7    ,L1   ,M   ,MC(5)     ,MCC(5)    ,MO(5)     ,MR(3,9)   ,MT
     8    ,MTP(40)   ,MTYP      ,MTYPS     ,N ,NC(30) ,NCOUNT    ,NPC(15)
     9    ,NPU ,NPV  ,NS(9)     ,NSS(5)    ,NST ,NSTA,NSTC,NSTX,NT,NTR(9)
     A    ,N8(3)     ,N9(3)     ,N10(3)    ,N11(3)    ,OMEGA     ,P(30,30)
     B    ,PA(3)                ,PM(3)     ,R         ,REO       ,RERP2
     C    ,RPO       ,ROT(5)    ,S(5)      ,SCAPH(5)  ,SI(3)     ,SIG(3,40)
     D    ,SIGM(3)   ,SIGMS(3)  ,SPHDT(5)  ,SUM ,SUM2,SYG(3,9)   ,TFINAL
     E    ,TFIT(9)   ,TIME      ,TIMES     ,TO        ,TP(40)    ,TQ(40)
     F    ,TR(40)    ,TT(40)    ,TXCG(20)  ,TYCG(20)  ,TYM(40)   ,TZCG(20)
     G    ,TZERO     ,XJ2       ,XMU       ,XMUJ      ,XP(3)     ,XXCG(20)
     H    ,XYCG(20)  ,XZCG(20)
      INTEGER OUT,FIT,STATE,SCRACH,PQR
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR
      CALL IDENT
      FLOS = 0.
      TIME = TZERO - 10.
      NALLSV=NALL
      MOPTSV=NPC(1)
      DCOMP=-DCOMP
      IF(NPC(1)-1) 4,6,2
    2 IF(NPC(8)-4) 12,6,12
    4 IF(NPC(7).GT.ICOUNT)    IF(NPC(8)-4) 16,6,16
      IF(NPC(8).LE.1) GO TO 8
      IF(NPC(8).LE.3) GO TO 14
    6 TZERO = TO
      NPC(7) = ICOUNT
      GO TO 62
    8 NPC(1) = 1
      NALL=10
    9 IF(NPC(8).EQ.0) GO TO 12
      IF(NPC(8).EQ.2) GO TO 12
   10 KSM = 1
      GO TO 18
   12 KSM = 0
      GO TO 18
   14 IF(NPC(8)-2) 12,12,10
   16 NPC(1) = 1
      NALL=10
      GO TO 9
   18 T2 = TO
```

```
      WRITE(OUT,1004)
   20 CALL INTAG
      TO=T2
      IF(ET.LE.TIME) GO TO 150
      IF(NPC(1).NE.1) CALL PROP
   24 CALL OUTPUT
      L1=1
   28 IF(T2-TZERO) 60,60,30
   30 ET=ET-DET
      IF(KSM.LE.0) GO TO 38
   36 TIME = TYM(KG)
   37 IF(TIME.LE.ET) GO TO 38
      T2 = TIME
      GO TO 20
   38 T2 = ET
      GO TO 20
   60 NALL=NALLSV
      NPC(1)=MOPTSV
      DCOMP=-DCOMP
      IF(ICOUNT-NPC(7)) 64,62,62
   62 ET=TFINAL
      NPC(1)=1
      L1=0
      GO TO 99
   64 ICOUNT=ICOUNT+1
      DO 66 I=1,6
   66 NTR(I) = 0
      REWIND FIT
      REWIND PQR
      KDATAS=-1
      KDATA=-1
      KPROP=-1
      NCOUNT=0
      L1=1
      BACKSPACE SCRACH
      BACKSPACE SCRACH
      READ (SCRACH) NSTX,NPU,NPV,((P(I,J),I=1,NSTX),J=1,NSTX),
     1    ((CUZ(I,J),I=1,NSTX),J=1,NPU),((CVZ(I,J),I=1,NSTX),J=1,NPV)
      CALL IDENT
   99 RETURN
  150 MTYP=MTP(KG)
      IF(KG.EQ.0) GO TO 158
      CALL OBSERV
      DO 156 I=1,3
      IF(MR(I,MTYP)) 152,154,152
  152 RES(I)=DFM(I)-DAT(I,KG)
      IF(MTYP.GT.5) GO TO 153
      IF(I.EQ.1) GO TO 153
      IF(ABS(RES(I)).LT.3.1416) GO TO 153
      RES(I)=RES(I)-SIGN(6.283185307179586,RES(I))
  153 CONTINUE
      RES(I+3)= RES(I)/SP(I)
      FLOS= FLOS + RES(I+3)**2
      KD(I)=NCOUNT
      NCOUNT=NCOUNT-1
      GO TO 156
  154 RES(I)=0.
      RES(I+3)=0.
      KD(I) = 0
  156 CONTINUE
```

```
      IF(L1.NE.0) WRITE(OUT,1000)                                       SMTH0880
      L1=0                                                              SMTH0890
      WRITE(OUT,1002) (KD(I),I=1,3),MTYP,TO,(RES(I),I=1,6),FLOS         SMTH0900
      KG=KG-1                                                           SMTH0910
      IF(KG.GT.0) GO TO 36                                              SMTH0920
  158 TIME=TZERO-10.                                                    SMTH0930
      GO TO 37                                                          SMTH0940
 1000 FORMAT(//5X6HPOINTS4X,4HTYPE,3X,4HTIME,6X,4HRES1,9X,4HRES2,9X,4HRESSMTH0950
     XS3,9X,8HWGT.RES1,5X,8HWGT.RES2,5X,8HWGT.RES3,5X,9HLOSS FCTN)      SMTH0960
 1002 FORMAT(1X,I4,1H,I4,1H,I4,1X,I2,1X,F10.3,7(1X,E12.5))              SMTH0970
 1004 FORMAT(///20(1H*),25HBEGIN BACKWARDS SMOOTHING,75(1H*))           SMTH0980
      END                                                               SMTH0990
```

```
      SUBROUTINE STAT
      COMMON /INTGRL/ DCOMP,T,T2,X(10),PH(10,40),DX(10),DPH(10,40),     COMM0010
     *     NALL,LRK                                                     COMM0020
      COMMON /TWICE/ A(3,3)     ,CPH      ,CT(5)    ,SP(10)   ,ST(5)    COMM0030
     1     ,TPH      ,VO(20)   ,ZO(20)   ,AG(3)    ,AMDOT(3) ,APDOT(4)  COMM0040
     2     ,B(8)     ,CGM(3)   ,DFM(3)   ,DTRAN(3,3),DUD(10,1),DUE(4)   COMM0050
     3     ,DUF(4)   ,F1(3,40) ,F2(3,5)  ,F3(4,40) ,GG(3)    ,HI(2)     COMM0060
     4     ,IDN ,II  ,IP  ,IPC ,JJ  ,JN  ,KA ,KD(3),KDUM ,KG2           COMM0070
     5     ,K1  ,NNN ,NPTS     ,PAR(6)   ,PDOT(3)  ,PMDOT(3) ,RES(6)    COMM0080
     6     ,RO       ,SPD(5)   ,SPH      ,TRAN(3,3),XX(50)              COMM0090
      DIMENSION DUB(30,15)                                              COMM0100
      EQUIVALENCE (AG(1),DUB(1,1))                                      COMM0110
      COMMON /METH2/ AB(3)      ,AM(3)    ,AP(4)    ,AX(40)   ,AY(40)   COMM0120
     1     ,AZ(40)   ,C(160)   ,CCAPH(5) ,CONRD    ,CPHDT(5) ,CUZ(30,5) COMM0130
     2     ,CVZ(30,5),D(5)     ,DAT(3,40),DATA(3)  ,DATAS(3) ,DATC(3)   COMM0140
     3     ,DET      ,DFIT(9)  ,DTF(9)   ,DTI(9)   ,DZ(30)   ,ET        COMM0150
     4     ,FLOS     ,G(3,30)  ,H(3,5)   ,I ,ICOUNT,J  ,JNBR,JNBRS,JST  COMM0160
     5     ,K   ,KAR ,KC(3)    ,KDAP     ,KDATA    ,KDATAS   ,KG,KI,KK  COMM0170
     6     ,KN  ,KOB ,KPROP,KS,KSM,KSS,L ,LC(3)    ,LCS(3)   ,LS   ,LT  COMM0180
     7     ,L1  ,M   ,MC(5)    ,MCC(5)   ,MO(5)    ,MR(3,9)  ,MT        COMM0190
     8     ,MTP(40)  ,MTYP     ,MTYPS    ,N ,NC(30),NCOUNT   ,NPC(15)   COMM0200
     9     ,NPU ,NPV ,NS(9)    ,NSS(5)   ,NST ,NSTA,NSTC,NSTX,NT,NTR(9) COMM0210
     A     ,N8(3)    ,N9(3)    ,N10(3)   ,N11(3)   ,OMEGA    ,P(30,30)  COMM0220
     B     ,PA(3)              ,PM(3)    ,R        ,REO      ,RERP2     COMM0230
     C     ,RPO      ,ROT(5)   ,S(5)     ,SCAPH(5) ,SI(3)    ,SIG(3,40) COMM0240
     D     ,SIGM(3)  ,SIGMS(3) ,SPHDT(5) ,SUM ,SUM2,SYG(3,9) ,TFINAL    COMM0250
     E     ,TFIT(9)  ,TIME     ,TIMES    ,TO       ,TP(40)   ,TQ(40)    COMM0260
     F     ,TR(40)   ,TT(40)   ,TXCG(20) ,TYCG(20) ,TYM(40)  ,TZCG(20)  COMM0270
     G     ,TZERO    ,XJ2      ,XMU      ,XMUJ     ,XP(3)    ,XXCG(20)  COMM0280
     H     ,XYCG(20) ,XZCG(20)                                          COMM0290
      INTEGER OUT,FIT,STATE,SCRACH,PQR                                  COMM0300
      COMMON /TAPENO/ IN,OUT,FIT,STATE,SCRACH,PQR                       COMM0310
      EQUIVALENCE (SP(3),PHIDT),(SP(4),CAPHI)                           STAT0010
      IF(MTYP.GT.5) RETURN                                              STAT0020
      N = 60 + 15*MTYP                                                  STAT0030
      SP(1) = COS(C(N+13))                                              STAT0040
      SP(2) = SIN(C(N+13))                                              STAT0050
C-------------CALCULATE EQ.(237) --------------------------            STAT0060
      PHIDT = ATAN2(RERP2*SP(2),SP(1))                                  STAT0070
      CAPHI= PHIDT - C(N+13)                                            STAT0080
      SPHDT(MTYP)=SIN(PHIDT)                                            STAT0090
      CPHDT(MTYP)=COS(PHIDT)                                            STAT0100
      CCAPH(MTYP)= COS(CAPHI)                                           STAT0110
      SCAPH(MTYP)= SIN(CAPHI)                                           STAT0120
      ROT(MTYP) = REO/SQRT(1.+(RERP2-1.)*SP(2)**2)                      STAT0130
   99 RETURN                                                            STAT0140
      END                                                               STAT0150
```

```
      FUNCTION    TAB(TARG,N,T,Y)                                         TAB 0000
      DIMENSION N(1),T(1),Y(1)                                            TABT0010
      IF(N(1))111,14,111                                                  TABT0020
  111 CONTINUE                                                            TABT0030
      I = N(2)                                                            TABT0040
    6 IF(TARG - T(I))3,2,1                                                TABT0050
    1 IF(N(3))9,5,9                                                       TABT0060
    5 I = I + 1                                                           TABT0070
      IF(I-N(1)) 6,4,4                                                    TABT0080
    4 I = I - 1                                                           TABT0090
    8 TAB = (Y(I+1)*(TARG - T(I)) - Y(I)*(TARG - T(I+1)))/(T(I+1)-T(I))   TABT0100
    7 N(2) = I                                                            TABT0110
   99 RETURN                                                              TABT0120
   11 I = I - 1                                                           TABT0130
    2 TAB = Y(I)                                                          TABT0140
      GO TO 7                                                             TABT0150
    9 IF(TARG - T(I-1)) 4,11,12                                           TABT0160
    3 IF(N(3))5,10,5                                                      TABT0170
   10 IF(TARG - T(I-1))12,11,4                                            TABT0180
   12 I = I - 1                                                           TABT0190
      IF(I-1)18,8,6                                                       TABT0200
   18 I=1                                                                 TABT0210
      GO TO 8                                                             TABT0220
   14 TAB = 0.                                                            TABT0230
      RETURN                                                              TABT0240
      END                                                                 TABT0250
```

Martin Marietta Corporation
Denver, Colorado, June 6, 1969

APPENDIX

DATA CONDITIONING

During postflight analyses studies the measurement data must be conditioned before being used in STEP. The airborne measurements received from the telemetry signal must be decoded, digitized, calibrated, time corrected, edited, smoothed, and formatted for use on the PQR tape. The data are not smoothed if they are to be used on the FIT tape. The radar tracking data must be edited, and formatted for use on the FIT tape. Methods that were used on the PRIME program postflight analyses are presented for accomplishing these data conditioning tasks. These methods are certainly not unique, but nevertheless do suggest one way of performing the data conditioning.

Consider a single channel of telemetry data that has been decoded, calibrated, and digitized. For each data point, their corresponds a time. This time must be consistent with the range time that is recorded with the tracking data. If the telemetry time is obtained from an airborne clock, then time corrections on the telemetry data may be necessary. These time corrections can be obtained by comparing range time and telemetry time at points where discrete events occur. This comparison will yield corrections that can be made to the telemetry time bringing it into agreement with range time. The measurement data may appear as shown in figure A1.

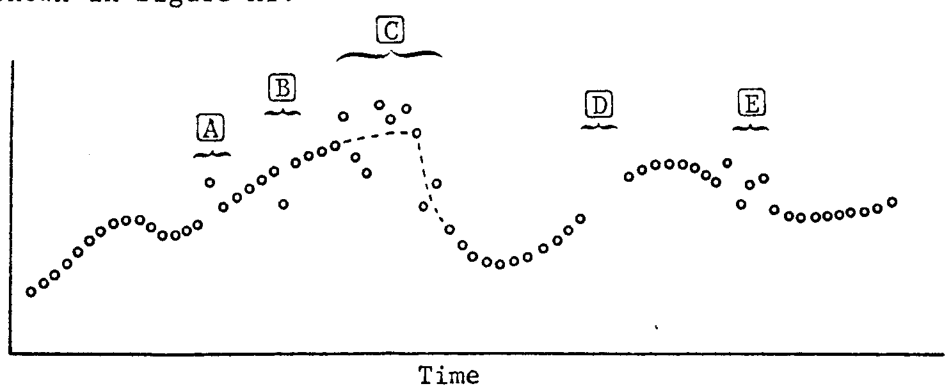


Figure A1.- Measurement Time History

Several types of anomolies are presented in figure A1. At A and B single wild points can be seen; at C is a series of approximately nine erroneous data points; at D is missing data, and at E another series of erroneous data points. The first problem encountered in the data conditioning task is that of determining where such anomolies occur. One must then decide if and how the data should be replaced. Finally, the data are smoothed if they are to be used on the PQR tape.

A. Determination of Anomolies

Eratic data can be manually identified by plotting the data. This, however, is very time consuming. Another way is to form a difference table for the data as tabulated below

t	X	ΔX	$\Delta^2 X$	$\Delta^3 X$
0	0.563			
		0.302		
0.1	0.865		0.008	
		0.310		-0.002
0.2	1.175		0.006	
		0.316		0.001
0.3	1.491		0.007	
		0.323		0.001
0.4	1.814		0.008	
		0.331		-0.001
0.5	2.145		0.007	
		0.338		-0.001
0.6	2.483		0.006	
		0.344		-0.006
0.7	2.827		0	
		0.344		0.005
0.8	3.171		0.005	
		0.349		
0.9	3.520			
.				
.				
.				

where $$\Delta^n X_i = \Delta^{n-1} X_{i+1} - \Delta^{n-1} x_i$$

For well-behaved data the higher differences remain small. Should a wild point occur it can immediately be identified by a rapid change in the higher differences. For example, if at $t = 0.5$ let $X = 4.145$ instead of 2.145. The errors in the differences fan to the right as follows:

t	X	ΔX	$\Delta^2 X$	$\Delta^3 X$
0	0.563			
		0.302		
0.1	0.865		·0.008	
		0.310		-0.002
0.2	1.175		0.006	
		0.316		0.001
0.3	1.491		0.007	
		0.323		2.000
0.4	1.814		2.008	
		2.331		-6.001
0.5	4.145		-3.993	
		-1.662		5.999
0.6	2.483		2.006	
		.344		-2.006
0.7	2.827		0	
		.344		.005
0.8	3.171		.005	
		.349		
0.9	3.520			

Such errors can be detected by testing the second or higher differences. When an eratic change occurs, a wild point or abrupt change in the data is detected. Where abrupt changes are known to occur (e.g., engine ignition or shutdown) the difference test should be neglected.

The difference test is implemented in the data conditioning task as follows. The n data channels are stored in chronological order on magnetic tape where each record contains N data points formatted as follows

N	T(1)	X(1,1)	X(1,2)	X(1,3)	. . .	X(1,m)
	T(2)	X(2,1)	X(2,2)	X(2,3)	. . .	X(2,m)
	.					
	.					
	.					
	T(N)	X(N,1)	X(N,2)	X(N,3)	. . .	X(N,m)

where m is the number of data channels (or separate measurements), and N is the number of time points per record. The data are then processed through a computer program that tests the second difference. When a difference exceeds a given value the erroneous point is flagged on a tab printout as well as on a magnetic tape on which the data are recorded using the following format:

N T(1) I(1,1), I(1,2) . . . I(1,m) X(1,1) X(1,2) . . . X(1,m)
T(2) I(2,1), I(2,2) . . . I(2,m) X(2,1) X(2,2) . . . X(2,m)
.
.
.
T(N) I(N,1) I(N,2) . . . I(N,m) X(N,1) X(N,2) . . . X(N,m)

The m fixed point flags for each line of data I(.,1) I(.,2) . . . I(.,2) are normally zero. If a wild point is detected in channel k its flag I(.,k) is set to one. Inspection of these flags on the tab printout allows immediate identification of the time and data channel for which an eratic point has been detected.

If unequally spaced time intervals occur in the data, divided differences can be used instead of the ordinary differences shown above. Should data gaps occur as in D of figure A1, the data should be replaced if they are to be used in the PQR tape. For data to be used in the FIT tape, the gap need not be replaced. Therefore, as the data are being tested for wild points, the time is simultaneously tested for gaps. When a gap is detected, the line of data is written on the tab printout and output tape. Because the missing data are unknown, zeros are written for the Xs and the flags I(i,j) set to one. The time is obtained by linear interpolation using points adjacent to the gap.

The output tape, called output tape 1, and tab printout, which are ultimately produced by this first stage of processing contains the times, flags and data. Where the data are correct, the flag is zero. Where data have been detected to be erroneous or missing, the flag is 1.

B. Data Replacement

Data to be used on the FIT tape requires no further processing. Final formatting will use only those points having zero flags on output tape 1 above. For data to be used on the PQR tape, replacement of wild points and gaps is performed next. The output tape 1, resulting from the processing in Subsection A above, is read into a program, which replaces all data with flags equal to one (wild points and gaps). The point replacement is accomplished by fitting a cubic polynomial to four "good" point bounding the gap or wild point. The points are then replaced by interpolating the cubic polynomial for the erroneous or missing data. The fitting points for the interpolating polynomial are spread as shown in figure A2 so that noise in the fitting data will not be exaggerated.

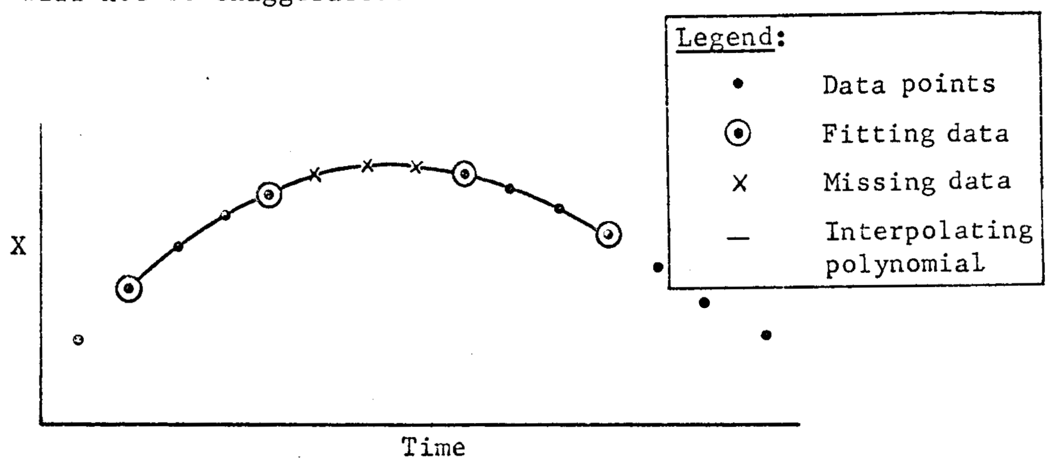


Figure A2.- Replacement by Interpolation

The points to which the polynomial are fit are circled in figure A2. If k points are to be interpolated in the gap, then the first and k + 1st points on each side of the gap are used to fit the cubic polynomial. As the gaps and wild points are replaced by interpolation their flags are set to 2.

Occasionally, cubic polynomials cannot be used to replace wild points or gaps. In set C of figure A1, it can be seen that a cubic polynomial would undoubtedly replace the data improperly. However, an individual knowing what physically occurred during this time might be better able to fair a reasonable curve through the region. Therefore, a second way of replacing data is by card input. The user specifies the time, channel, and value for the data, and the program inserts it on output tape 2 flagged with a 3 to denote that it was provided by the user.

C. Smoothing

The data on output tape 2 resulting from the data replacement task above, is next inputted into a program that smoothes it. If time derivatives of the data are required (as is the case with strapped-down accelerometers and gyros), smoothing and differentiating occurs. Smoothing and differentiating formulas used on the PRIME program are classified as linear finite memory smoothing, and smoothing and differentiating formulas (see ref. 3). The smoothing formula assumes the form

$$C_n = \sum_{k=o}^{n-1} a_k r_{n-k} \tag{A1}$$

where C_n is the value of a smoothed parabola fit to n data points r_1, r_2, . . ., r_n and evaluated at the end point n. The a_k are the coefficients of the linear filter and are calculated as follows;

$$a_k = \mu_1 + \mu_2 k + \mu_3 k^2 \tag{A2}$$

with

$$\mu_1 = \frac{18(2n - 1)\alpha + 30\alpha^2 + 3(3n^2 - 3n + 2)}{n(n + 1)(n + 2)} \tag{A3}$$

$$\mu_2 = - \frac{12(2n - 1)(8n - 11)\alpha}{n(n - 1)(n + 1)(n^2 - 4)} - \frac{180\alpha^2}{n(n + 1)(n^2 - 4)} \tag{A4}$$

$$\mu_3 = \frac{180\alpha}{n(n + 1)(n^2 - 4)} + \frac{180\alpha^2}{n(n^2 - 1)(n^2 - 4)} + \frac{30}{n(n + 1)(n + 2)} \tag{A5}$$

where α is the amount of lag or lead $\left(\text{i.e. } C_n = R_{n+\alpha}\right)$. For an end point filter $\alpha = 0$, for a midpoint filter $\alpha = - \frac{n-1}{2}$ with n odd.

For smoothing and differentiating, C_n in equation (A1) is the first derivative of a smoothed parabola fit to the n data points r_1, r_2, . . ., r_n and evaluated at point n with lag or lead α. The coefficients are calculated from equation (A2) with

$$\mu_1 = \frac{6(6n - 3 + 10\alpha)}{nT(n + 1)(n + 2)} \tag{A6}$$

$$\mu_2 = \frac{12[(8n - 11)(2n - 1) + 30(n - 1)\alpha]}{nT(n^2 - 1)(n^2 - 4)} \tag{A7}$$

$$\mu_3 = \frac{180(n - 1 + 2\alpha)}{nT(n^2 - 1)(n^2 - 4)} \tag{A8}$$

where T is the fixed time interval between adjacent data points.

Equation (A1) is an end point filter since to find C_n the n data points before and including r_n are used. By a simple transformation the end point filter can be converted to any other type filter by merely shifting the points being used to the right of C_n by means of α.

An obvious difficulty arises when attempting to apply the filter near the beginning, or end of a table or across a discrete such as thrust ignition or termination. At the beginning a table the filter is commenced as an end point filter and "grown" to a midpoint (or whatever is desired) filter as the filter is walked through the data. At the end of a table, a midpoint filter is grown into an end point filter. At discontinuities, the filter is changed from midpoint to endpoint at the discontinuity. Then it is restarted as an end point filter on the other side of the discontinuity and grown back to a midpoint filter. On the PRIME program, the filter span (number of points N) was decreased as the data nonlinearity increased. When the vehicle was out of atmosphere and the data very smooth an eleven point filter was used. This was reduced to a five-point filter during in-atmosphere maneuvers.

REFERENCES

1. Wagner, W. E.: Reentry Filtering, Prediction, and Smoothing. J. Spacecraft Rockets, vol. 3, no. 9, Sept. 1966, pp. 1321-1327.

2. Clark, S. P., ed.: Handbook of Physical Constants. Geological Society of America, Inc., 1966.

3. Monroe, A. J.: Digital Processes for Sampled Data Systems. John Wiley and Sons, New York, 1962.