NASA CR-57031

# THREE DIMENSIONAL ~~DIMENSIONAL~~ TRAJECTORY OPTIMIZATION STUDY

## PART II - COMPUTER PROGRAM & USER'S MANUAL

First Revision

BR-3136-2

25 September 1964


Prepared by

Ralph S. Goodell


Submitted to

LANGLEY RESEARCH CENTER

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Hampton, Virginia


Contract NAS1-4056



FACILITY FORM 602

X65 10934 (ACCESSION NUMBER)

140 (PAGES)

CR 57031 (NASA CR OR TMX OR AD NUMBER)

(THRU)

(CODE)

08 (CATEGORY)


RAYTHEON


~~Available to NASA Offices, NASA Centers, and NASA Contractors Only.~~

# THREE DIMENSIONAL TRAJECTORY OPTIMIZATION STUDY

## PART II - COMPUTER PROGRAM AND USER'S MANUAL

First Revision

BR-3136-2 25 September 1964


Prepared by

Ralph S. Goodell


Submitted to

LANGLEY RESEARCH CENTER

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Hampton, Virginia


Contract NAS1-4056



RAYTHEON COMPANY

SYSTEMS PERFORMANCE DEPARTMENT

SPACE AND INFORMATION SYSTEMS DIVISION

Bedford, Massachusetts



~~Available to NASA Offices and NASA Contractors Only~~

# ABSTRACT

10934

The computer program for the Raytheon three-dimensional trajectory optimization deck (TOS) is described together with instructions for its use. This deck determines the angle-of-attack and bank-angle histories that optimizes a specified payoff function subject to as many as nine terminal constraints. A steepest-ascent optimization technique is employed and an automatic-convergence procedure is incorporated to facilitate convergence. The program is written in FORTRAN IV for the IBM 7090/7094 operating under the Basic Monitor (IBSYS).

II

# FOREWORD

This volume describes the current version of the Raytheon three-dimensional trajectory optimization computer program (TOS) which is written principally in FORTRAN IV. It supersedes a volume bearing the same title and issued as Raytheon Company Report BR-1759-2 (also, Aeronautical Systems Division Report No. ASD-TDR-62-295, Part II) which described the original TOS deck.

The conversion of this program to FORTRAN IV and its modification to include an automatic-convergence procedure and other features that will increase its usefulness was supported by the NASA Langley Research Center under Contract NAS 1 - 4056.

A description of the mathematical formulation of this program is given in Raytheon Company Report BR-3136-1, Three Dimensional Trajectory Optimization Study, Part I - Optimum Programming Formulation.

II

TABLE OF CONTENTS

| | Page |
|---|---|
| ABSTRACT | iii |
| FOREWORD | v |
| 1. INTRODUCTION | 1-1 |
| 2. SYSTEM AND COMPUTER REQUIREMENTS | 2-1 |
| 3. PROGRAM DESCRIPTION | 3-1 |
| 3.1 List of Subprograms | 3-1 |
| 3.2 General Description of TOS | 3-2 |
| 3.3 $IBFTC TML | 3-4 |
| 3.4 $IBFTC MVL | 3-5 |
| 3.5 $IBMAP CLR | 3-5 |
| 3.6 $IBMAP EMK | 3-5 |
| 3.7 RM ATM1 | 3-6 |
| 3.8 RM BLN1 | 3-7 |
| 3.9 RM LIN1 | 3-9 |
| 3.10 $IBFTC MIV | 3-10 |
| 3.11 $IBFTC OUT | 3-12 |
| 3.12 MARK | 3-14 |
| 3.13 RM SMK1 | 3-15 |
| 3.14 $IBFTC VFI | 3-16 |
| 4. INPUT | 4-1 |
| 4.1 General Description | 4-1 |
| 4.2 Input Symbols | 4-3 |
| 4.3 List of Payoff, Stopping, and Constraint Functions | 4-10 |
| 5. OUTPUT | 5-1 |
| 6. SYSTEM INFORMATION | 6-1 |
| 7. RESTRICTIONS | 7-1 |
| 8. PROGRAM LISTING | 8-1 |

APPENDIX I

MARK INTEGRATION ROUTINE I-1

APPENDIX II

EQUATIONS II-1

APPENDIX III

DEFINITION OF VALUES IN COMMON STORAGE III-1

# 1. INTRODUCTION

TOS is a computer program for the IBM 7090/7094, which determines trajectories in three dimensions that maximize or minimize a specified payoff function subject to terminal constraints using the steepest-ascent technique. The steepest-ascent technique employs the concept of local linearization around a nominal path. The effect on the terminal conditions of small changes in the control-variable programs is determined by numerical integration of the adjoint differential equations and computation of the associated impulse response functions. These functions make possible the determination of the new control-variable programs that yield a maximum change in the payoff function for a given squared perturbation in the control-variable programs while simultaneously changing terminal quantities by desired amounts. By repeating this procedure, control-variable programs that optimize one terminal quantity and yield specified values of other terminal quantities can be approached as closely as desired.

The TOS computer program is written for the IBM 7090/7094 to be run under the Basic Monitor (IBSYS). TOS is written in FORTRAN IV except for the input routine, the differential equation solving routine, and several auxiliary routines. The main features of the TOS computer program are:

1) It solves the trajectory of a vehicle in three dimensions subject to two control variables, angle of attack and bank angle. The vehicle is treated as a particle and the earth is approximated as an oblate spheroid having an atmosphere rotating with it. Tables for lift and drag coefficients as functions of angle of attack and mach number are provided for three boost stages and a glide stage. The thrust and mass histories for the boost stages are input as tables also.
2) Equations for total heat, pilot acceleration dose, and heating-rate and altitude penalty functions are integrated along with the trajectory. Space is provided in the program for two additional penalty functions.

3) A library of 23 functions is provided from which may be selected the payoff, stopping, and terminal constraint functions. Other functions may be added easily.

4) An automatic convergence procedure controls the size of steepest-ascent step, modifies the computed control-variable programs during the integration of the state equations, and introduces other features that facilitate rapid convergence.

5) Flexible input and output routines are incorporated in the deck.

## 2. SYSTEM AND COMPUTER REQUIREMENTS

The TOS computer program is written for the IBM 7090/7094 to be run under the Basic Monitor (IBSYS).

TOS uses the overlay feature and requires one additional tape above those necessary for the Monitor.

# 3. PROGRAM DESCRIPTION

## 3.1 List of Subprograms

| Name | Function | Coding |
|---|---|---|
| TOS | Main Program | FORTRAN IV |
| STE | Forward Trajectory | FORTRAN IV |
| ADT | Adjoint Trajectory | FORTRAN IV |
| CLD | Lift and Drag Coefficients | FORTRAN IV |
| PRL | Partial Derivatives | FORTRAN IV |
| FCT | Payoff, Stopping, and Constraint Functions | FORTRAN IV |
| SFR | Save Forward Trajectory | FORTRAN IV |
| PIN | Read and Print Input | FORTRAN IV |
| ITM | Slave to OUT | FORTRAN IV |
| TML | Triple Matrix Product (TOS) | FORTRAN IV |
| MVL | Matrix Inversion (TOS) | FORTRAN IV |
| CLR | Clear Memory | MAP |
| EMK | Obtain MARK Error Information | MAP |
| UN8 | File Definition for Scratch Tape | MAP |
| ATM1 | Atmosphere Subroutine | MAP |
| BLN1 | Bivariate Linear Interpolation | MAP |
| LIN1 | Linear Interpolation | MAP |
| MIV | Matrix Inversion | FORTRAN IV |
| OUT | Output Routine | FORTRAN IV |
| MARK | Differential Equation Solving Routine | MAP |
| SMK1 | FORTRAN Driver for MARK | MAP |
| VFI | Variable Field Input Routine | MAP |

## 3.2 General Description of TOS

### 3.2.1 Input and Initialization (PIN)

The first thing TOS does is to initialize various quantities, set nominal values for input, and set up the symbol table for the input routine. TOS uses a variable field input routine, VFI, written by AVCO and modified by Raytheon for use in this program. Section 4 describes the input necessary for this program and also includes a discussion of the features of the variable field input program used by TOS. The complete input for each problem is listed as the first printout of the problem using PIN. This feature is preferred to the PRINT option in VFI since that option merely lists the specific cards read to initiate a particular problem.

### 3.2.2 Forward Trajectory (STE)

TOS next computes the forward trajectory by integrating the six differential equations of motion plus six auxiliary equations, the last two of which are undefined in the current deck and set to zero. The differential equation solving routine used by TOS is MARK. MARK is driven by another routine, SMK. MARK and SMK are described in Sections 3.12 and 3.13. The twelve differential equations are integrated in the variable Adams-Moulton method using second differences.

The input for this section consists of the initial values of the stage equations and the control-variable tables for angle of attack, $\alpha$, and bank angle, $\sigma$. These values are read from the input tape.

The output of this section, disregarding the print output, is a table of the independent variable, time, and the twelve dependent variables at each integration step. These values are saved on tape in subroutine SRF. This routine is entered after every integration step and when the stopping condition is reached. These values along with the end conditions are used as input to the adjoint trajectory.

The print output of the forward trajectory is selected through input. A description of this output can be found in Section 5.

The forward trajectory is terminated when the stopping condition is reached. The stopping condition is specified through the stopping function

number and the stopping value which are part of input. The stopping function number is used by function FCT to get the value of the stopping function at any time.

### 3.2.3 Adjoint Trajectory (ADT)

This section computes the adjoint trajectory associated with the forward trajectory previously calculated. For the payoff function and each terminal-constraint function there are twelve adjoint differential equations. Only six of these differential equations need be integrated since six are constant.

The initial values of the adjoint differential equations are computed via the subroutine PRL, which evaluates the partial derivatives of the payoff and constraint functions with respect to the state variables.

These differential equations are also computed by MARK. A second difference, variable Adams-Moulton method is used.

The adjoint trajectory uses the end conditions of the forward trajectory together with the tape constructed during the forward trajectory to compute the adjoint differential equations.

The adjoint trajectory is divided into 200 equally spaced intervals (201 points). At these 201 points the impulse response functions are saved in a table, GL. There are two impulse response functions per each payoff and constraint function, one with respect to $\alpha$, the other with respect to $\sigma$.

At the termination of the adjoint trajectory the $I_{\Phi\Phi}$, $I_{\psi\Phi}$, and $I_{\psi\psi}$ integrals are computed.

### 3.2.4 New Control-Variable Programs

The impulse response function table, GL, the $I_{\Phi\Phi}$, $I_{\psi\Phi}$, and $I_{\psi\psi}$ integrals, and the current value of the steepest-ascent step size, $(dP)^2$, are used to compute new control-variable programs. The control-variable programs consist of two tables, ALPHX and SIGMX. ALPHX is composed of 201 values of $\alpha$ and SIGMX 201 values of $\sigma$.

### 3.2.5 Analysis

This section falls between the forward trajectory and adjoint trajectory. Here several convergence criteria are tested and the steepest-ascent step size is changed depending on these tests.

A check is also made for the stopping conditions for the case.

### 3.2.6 Miscellaneous Subprograms

FCT

FCT is a function subprogram which computes the payoff, stopping, and constraint functions. The routine is entered with the function number and returns with the function. FCT has 23 functions which may be used as payoff, stopping, or constraint functions. Others may be added. The table FNAME in TOS contains the names of these functions for use with print out.

CLD

CLD is a subroutine which obtains the lift and drag coefficients from input tables by bivariate linear interpolation.

ITM

ITM is a subroutine to OUT which obtains the functions specified by the reference numbers in the IGO statement for use in the printout.

UN8

UN8 provides the file definition for the scratch tape.

## 3.3 $IBFTC TML

Triple Matrix Product Function Subprogram for TOS program

Purpose: Given three matrices to compute their product eliminating specified rows and columns.

Usage: Calling sequence:

X = TMAML(A, B, C, L)

where A is a 1 x 8 matrix
B is an 8 x 8 matrix
C is an 8 x 1 matrix
and L is a 1 x 8 matrix

X will equal the product of A x B x C where the rows and columns of all the matrices are eliminated if the respective component in the L matrix equals 0.

A, B, and C are real matrices. L may be a real or integer matrix.

## 3.4 $IBFTC MVL

Matrix Inversion Subroutine for the TOS Program

Purpose: Given a matrix find its inverse eliminating specified rows and columns.

Usage: Calling sequence:

CALL MINVL (A, AI, ID)

where A is an 8 x 8 matrix to be inverted
AI is the inverted matrix (8 x 8)
and ID is a 1 x 8 matrix which designates which columns and rows are to be eliminated.

If a component in the ID matrix = 0 the respective rows and columns in A are eliminated. The A matrix is then inverted and stored in the AI matrix. The rows and columns that were eliminated in the A matrix are set to 0 in the inverse.

This subroutine uses SUBROUTINE MATINV to obtain the inverse.

## 3.5 $IBMAP CLR

Clear Storage Subroutine

Purpose: To clear specified portions of memory.

Usage: Calling sequence:

CALL CLEAR (A, B, C, D, . . . . . )

Core locations A through B, C through D, etc., will be cleared to zero.

## 3.6 $IBMAP EMK

Obtain Mark Error Information

Purpose: To obtain the step size and truncation error from the MARK integration routine.

Usage: Calling sequence:

CALL EMARK(HC, RGERR)

The subroutine returns with the actual step size in HC and the truncation error in RGERR.

## 3.7 RM-ATM1

U. S. Standard Atmosphere, 1962

J. A. Maltais - Aug. 30, 1963

Raytheon Company, Missile Systems Division

### Purpose

ATM 01 is a closed subroutine which furnishes a model of the U. S. Standard Atmosphere, 1962 for the properties: pressure, density, and speed of sound.

### Description

The U. S. Standard Atmosphere, 1962 is described for the altitude range -5 km (-16404 ft) to 700 km (2, 296, 588 ft). The defining property of this atmosphere is the molecular scale temperature. On the basis of the molecular scale temperature the atmosphere is divided into a system of twenty-one layers. The base of each layer has associated with it a particular molecular scale temperature with a constant temperature gradient across the layer.

The numerical computation of the atmospheric properties results in a dichotomy of the calculations for altitudes less than and greater than 90 km (295, 276 ft). The variation in the calculations directly concerns molecular scale temperature and pressure and indirectly density and speed of sound. For geometric altitudes (z) less than 90 km the approximation for geopotential altitude (H) is $H = z\left(\frac{a}{a+z}\right)$ where (a) is the earth's equatorial radius. For geometric altitudes greater than 90 km pressure is expressed as an integral and is evaluated using a three point Simpson's rule quadrature formula. The approximation used for gravity which appears in the expression for pressures above 90 km is

$$g = \left\{ \frac{GM}{r^2} \left[ 1 + J \left(\frac{a}{r}\right)^2 (3 \sin^2 \psi - 1) \right] - r \Omega^2 \cos^2 \psi \right\}$$

For altitudes greater than 90 km, the value of the speed of sound is set arbitrarily at the value for 90 km. At altitudes greater than 700 km, the values for pressure, density, and speed of sound are set to zero.

Input and output to this subroutine is expressed in English units, but the internal calculations are performed in the metric system of units.

Usage

The following is a description of the way ATM 01 may be used:

CALL ATM (H, A)

where H is the location containing the altitude in feet

and A is the first cell of a block of three locations where the calculated values of pressure, density, and speed of sound are stored by the subroutine. Pressure is stored at A (1), density at A (2), and speed of sound at A (3).

A (1) Pressure, in $LBF/FT^2$

A (2) Density, in $SLUGS/FT^3$

A (3) Speed of sound, in FT/SEC

Restrictions

ATM 01 requires the use of three subroutine subprograms: namely, LOG, EXP, and SQRT.

## 3.8 RM BLN1

Bivariate Linear Interpolation
7040/7044 7090/7094 IBMAP
R. Goodell - July 21, 1964
Raytheon Company, Missile Systems Division

Purpose: Given the arguments x and y compute $z_1$, $z_2$, etc., by double linear interpolation from unequally spaced tables of x's, y's, $z_1$'s, $z_2$'s, etc. If x and/or y is outside the range of their tables, the z's are set equal to the nearest value in the table.

Usage: Calling sequence:

CALL BILIN(N, X, XTAB, Y, YTAB, Z1, Z1TAB, Z2, Z2TAB
as many dependent variables and tables as desired)

where N = return indicator
1 - x and y within range
2 - x low
3 - x high
4 - y low (y off range returns override the x off range returns)
5 - y high

X = argument x
XTAB = location of table of x's
Y = argument y
YTAB = location of table of y's
Z1 = dependent variable $z_1$
Z1TAB = location of table of $z_1$'s
Z2 = dependent variable $z_2$
Z2TAB = location of table of $z_2$'s

Notes

1) The x's and y's tables are terminated by a nonmonotonically increasing value.
2) The x's and y's tables must be monotonically increasing (may be equal).
3) There must be at least two values in both x's and y's tables.
4) The tables of z's are stored contiguously with the x's varying first.

Example

| | | x → | | |
|---|---|---|---|---|
| | | -1 | 0 | 2 |
| y | -10 | 6 | 8 | 18 |
| ↓ | -7 | -3 | 5 | 24 |

Table as it would appear in storage:

```
XTAB    DEC    -1.
        DEC     0.
        DEC     2.
        DEC     0.        nonmonotonic value
YTAB    DEC   -10.
        DEC    -7.
        DEC  -100.        nonmonotonic value
ZTAB    DEC     6.
        DEC     8.
        DEC    18.
        DEC    -3.
        DEC     5.
        DEC    24.
```

## 3.9 RM LIN1

Linear Interpolation Subroutine
7040/7044 7090/7094 IBMAP
R. Goodell - July 21, 1964
Raytheon Company, Missile Systems Division

Purpose: Given the argument x, compute $y_1$, $y_2$, etc., by linear interpolation from unequally spaced tables of x's, $y_1$'s, $y_2$'s, etc. If the argument x is not within limits of the x's table, the subroutine returns with an indicator designating which end of the table is nearest and the y's are set to the values at the nearest end.

Usage: Calling sequence:

CALL MULG(NVAL, N, X, XTAB, Y1, Y1TAB, Y2, Y2TAB,
as many dependent variables and tables as desired)

where NVAL = maximum number of values in tables
N = return indicator
1 = x within table
2 = x off low end
3 = x off high end

| | | |
|---|---|---|
| X | = | argument x |
| XTAB | = | location of table of x's |
| Y1 | = | $y_1$ |
| Y1TAB | = | location of table of $y_1$'s |
| Y2 | = | $y_2$ |
| Y2TAB | = | location of table of $y_2$'s |

Notes

1) The end of the x's table is indicated by either the maximum number of values or a nonmonotonic (strictly) value in the table.
2) x's table must be monotonically increasing (may be equal).
3) There must be at least two values in the x's table.

## 3.10 $IBFTC MIV

AN F402, Matrix Inversion with Accompanying Solution of Linear Equations (FORTRAN II)

Burton S. Garbow - February 23, 1959

Argonne National Laboratory, Lemont, Illinois

Purpose

FORTRAN II Subroutine solves the matrix equation AX = B, where A is a square coefficient matrix and B is a matrix of constant vectors. $A^{-1}$ is also obtained; indeed, inversion may be the sole aim in a particular usage. Finally, the determinant of A is available; other possibly useful information is in COMMON storage.

Method

Jordan's method is used to reduce a matrix A to the identity matrix I through a succession of elementary transformations; $\ell_n \ell_{n-1} \cdots \ell_1 A = I$. If these transformations are simultaneously applied to I and to a matrix B of constant vectors, the result is $A^{-1}$ and X where AX = B.

Usage

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

CALL MATINV (A, N, B, M, DETERM)

where 1) N is the order of A; $N \geq 1$.

2) M is the number of column vectors in B.

3) DETERM is the location in which the determinant is to be placed.

Suitable variable names may replace the dummy variables listed above.

For compatibility purposes with the subroutine which was compiled on the basis of N = 20, dimension statement entries for the 2-dimensional arrays, A and B, in the calling program must have row dimension equal to 20; e. g., A(20, 10), or B(20, 2). If this imposes too severe a storage requirement, or on the other hand if it is desired that N > 20, the subroutine can be recompiled with a new DIMENSION statement replacing all entries of 20 by the desired value.

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

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

Space required: $711_8$ ($457_{10}$) locations in addition to an extent of 60 + N locations at COMMON distributed as follows down from upper memory:

N locations for PIVOT - the array of pivot elements used in the inversion.

20 - N locations not used.

N locations for column 1 of INDEX - a 2-column array which records consecutive row interchanges.

20 - N locations not used.

N locations for column 2 of INDEX.

20 - N locations not used.

N locations for IPIVOT - an array used to prevent duplicate pivotings on any single row.

Coding Information

Timing

Running time is approximately proportional to $N^3$; for N = 75 (using recompiled version) it takes about 4-1/2 minutes.

Card Decks

1) FORTRAN statements: 85 cards numbered ANF40201, F4020002, F4020003, . . . , F4020085.
2) Object deck: 23 cards numbered ANF40201 - 23.

## 3.11 $IBFTC OUT

General FORTRAN Output Routine

M. Frazier, Aug. 4, 1964

Raytheon Company, Space and Information Systems Division

Two things are required for use of the general output routine. First is the actual call to the routine, second is a method of referencing labeled COMMON. Discussing the latter first, the user must provide a FORTRAN function ITEM (I, J), which returns the Ith item in the Jth COMMON block, as follows:

```
      REAL FUNCTION ITEM (I, J)
      COMMON/ABLE/X, Y, Z
      COMMON/BAKER/UP, SIDE, DOWN   } Complete user's common of,
      COMMON/ZEBRA/DONE, AT, LAST   } say, 6 labeled common areas.
      REAL X(1), UP(1), DONE(1)       (first item in each block is called
                                       real, whatever it may be actually.)
      GO TO (10, 20, 30, 40, 50, 60), J
   10 ITEM = X(I)
      RETURN
   20 ITEM = UP(I)
      RETURN
      .    .    .
   60 ITEM = DONE(I)
      RETURN
      END
```

To generate output, the user simply codes

CALL OUT (IGO, HFMT, DFMT)

where IGO, HFMT, and DFMT are arrays of quantities described below.

| Name | Length | |
|---|---|---|
| IGO | 1 | Print if = 1, punch if = 2, otherwise do nothing |
| ITYP | 1 | Output type, see below for explanation |
| INH | 1 | Number of lines produced by heading format |
| INL | 1 | Number of lines produced by data format |
| M | 1 | Number of data items |
| IT | | Data item table |
| | 2, 20 | ITYP = 1 |
| | 2, 40 | ITYP $\neq$ 1 |
| HFMT | | Data heading format |
| | 40 | ITYP = 1 |
| | unlimited | ITYP $\neq$ 1 |
| DFMT | | Data format |
| | 40 | ITYP = 1 |
| | unlimited | ITYP $\neq$ 1 |

Some of these quantities deserve fuller explanation.* There are four entries to the subroutine, selected by ITYP = 1, 2, 3, or 4. ITYP = 1 signifies a page heading. A CALL OUT whose array contains an ITYP = 1 will immediately restore the paper, then write the contents of HFMT (if INH $\neq$ 0), followed (if M $\neq$ 0) by M pieces of data specified by the IT table in the format specified by DFMT. This output will be repeated automatically every time the line counter overflows. A new page heading format will be generated by calling OUT with ITYP = 1 again, otherwise the one generated by the first such call will be used. Calling OUT with ITYP = 2 is the standard method of putting out assorted data. The contents of HFMT will be printed immediately after a page is restored (to provide column headings), but not otherwise. M items will be printed in format given by DFMT, the items printed will be selected by the ITEM subroutine using the IT table as follows:

*See Figure 4-1 for an example of the use of this routine.

```
      DO 10 I = 1, M
   10 TAB(I) = ITEM (IT(1, I), IT (2, I))
      WRITE (6, DFMT) (TAB(I), I = 1, M)
```

(If punching this becomes WRITE (7, . . . ) etc. )
This is also the way the IT table is used in printing out the heading data above. ITYP = 3 is used for printing out vectors and 2 dimensional arrays where the first subscript varies most rapidly. Only one such array is printed per call, and it is preceded by the contents of HFMT. The IT table is handled slightly differently. M is the length of the vector (or array column) to be printed. IT (1, 1) and IT (2, 1) are used as above, IT (1, 2) = the number of columns in the array (= 1 for vectors), and IT (2, 2) = the first dimension of the array in core (M $\leq$ IT (2, 2)), but IT (2, 2) is not needed for printing vectors (IT (1, 2) = 1). When an array is being printed, INL is the number of lines required to print 1 column.

Finally, ITYP = 4 is used for printing out arrays in row order, and is described in detail by interchanging the words "row" and "column" above.

Incidentally, if IGO = 2 (for punched output), only data is punched, and headings and page headings are ignored.

Continuing through the IGO block, INH is used to control the line counter. If it is zero, the contents of HFMT is not printed. INL is also used to control the line counter. For two dimensional arrays, INL is the number of lines needed to print one column (or row if ITYP = 4).

M is the number of data items or, if a two-dimensional array is being printed, the number of items per column (or row if ITYP = 4).

IT has been explained above. HFMT and DFMT are the output formats, including the surrounding parentheses.

The IGO block may be filled with a DATA statement, or by input, as desired, which gives the great flexibility of this output routine.

## 3.12 MARK

The MARK integration routine is described in Appendix I.

## 3.13 RM SMK1

FORTRAN setup routine for JP MARK
7040/44 7090/94 IBMAP
R. Goodell - July 22, 1964
Raytheon Company, Missile Systems Division

### Purpose

This subroutine is buffer between JP-MARK, a differential equation solving routine written in MAP language, and a main program written in FORTRAN. SMK1 allows a program written in FORTRAN IV to use most of the features of JP-MARK.

### Usage

A knowledge of JP-MARK is assumed.

Calling sequences:

```
CALL SMARK (KIND, N, HBANK, NRTN, NTRG,
EUBAR, ELBAR, HMAXT, HMINT, YCLOW,
LV1, TV1
LV2, TV2,
---- up to 10 triggers)
```

where KIND = type of integration
- 0 = fixed AM integration
- 2 = RK integration
- 4 = variable AM integration

N = actual number of differential equations

HBANK = location of a bank of storage used by JP-MARK equivalent to HBANK-3

NRTN = return indicator from SMARK
- 1 = EOS
- 2 = DER1
- 3 = DER2
- 4 = trigger return
- 5 = error return

NTRG = return indicator from SMARK which designates which trigger has been activated
1 for first trigger, 2 for second, etc.

EUBAR, ELBAR, HMAXT, HMINT, YCLOW same as in JP-MARK

$LV_i$ = location of variable being tested

$TV_i$ = location of desired value of the variable being tested

CALL TRA 14 returns control to SMARK and causes a TRA 1, 4 return to JP-MARK.

CALL TRA 24 returns control to SMARK and causes a TRA 2, 4 return to JP-MARK

CALL ON(N) turns trigger N on

CALL OFF(N) turns trigger N off

All triggers are turned on when SMARK is called.

The order of the differences to be carried in AM integration must be stored in HBANK(1), the nominal step size in HBANK(4), the maximum number of equations allowable in HBANK(5), and all independent and dependent variables initialized before calling SMARK. The double precision part of the independent variable HBANK(7) is set to zero by SMARK.

## 3.14 $IBFTC VFI

Variable Field Input Subroutines*

AVCO, Research and Advanced Development Division

Purpose

1) BCDCON: To generate a table of six character BCD names vs associated core addresses.
2) SYMBLS: To read, convert, list, and store fixed, floating, or hollerith input data.
3) HDTAPE: To write a 72 hollerith field read by SYMBLS on a specified tape.

*Several minor modifications have been made by Raytheon Company.

Usage

Calling sequence:

1) CALL BCDCON (nHA, B (10), C(10, 10), D(10, 10, 10), etc., E, F, G, H, etc.)

where n $\geq$ 12 times number of single subscripted variables
+ 18 times number of multisubscripted variables

2) CALL SYMBLS (INCOLI)

3) CALL HDTAPE

Complete description of procedures for proper use of these subroutines:

1) BCDCON: The n hollerith field described above will contain any BCD names followed by the dimension of their associated variables appearing in the rest of the list. Thus using the example above BCDCON will generate at object time a table which will be correct if F were dimensioned 10, G were dimensioned 10 by 10, and H were dimensioned 10 by 10 by 10. This table then makes the association of the hollerith field names and dimensions with the other items in the list.

(e.g. Symbol A corresponds to core location defined by E.
Symbol B corresponds to core location of the beginning of the array defined by F(10).
Symbol C corresponds to core location of the beginning of the array defined by G(10, 10).
Symbol D corresponds to core location of the beginning of the array defined by H(10, 10, 10). )

If it were desired to define other than beginning locations of arrays, this can be done in the normal FORTRAN manner of subscripting the variables in the list; (e.g., if you wanted to define Symbol C with the second column of G, then instead of writing G, write G(10, 2). This, however, does not affect the dimension associated with the symbol C.)

In the processing of the hollerith field, commas and parentheses are the breaking symbols. That is, a symbol will be considered complete when a comma or a left parenthesis is encountered. When a comma is present immediately following a symbol, no dimensionality is expected. When a left parenthesis

is encountered, a dimensionality is expected which will be terminated by a right parenthesis. Each item of the dimension must be separated by a comma. After a right parenthesis, a comma is expected.

Errors

There are three possible errors that can be committed in the call of BCDCON.

a) More than three subscripts given. Usually occurs when a right parenthesis on the dimension is missing.
b) More variables in the list than in the hollerith field. Usually happens when there are commas missing in the hollerith field, when the total number of hollerith characters (n) is too small, or when the programmer has simply put more variables in the list than he meant to.
c) More symbols in the hollerith field than variables defined in the rest of the list. Usually happens when the total number of hollerith characters (n) is too large, when there are too many commas in the variable field, or again when the programmer has miscounted the items.

All these errors will result in a defining error message and cause the execution of the rest of the program to be deleted.

Cautions

BCDCON is used to generate a table for use in SYMBLS. Where this table is located is passed by BCDCON to SYMBLS. Now, while multiple calls of BCDCON are readily permissible, it must be remembered that to call BCDCON in more than one subprogram could be dangerous (not necessarily fatal); since the first call defines the beginning of the table, you will be replacing core locations in one subprogram originally defined in the other subprograms. The inherent dangers are obvious.

Also, BCDCON will allow more than six characters to represent a symbol in the hollerith field; however, it will use only the first six characters as its table entry. Therefore, any symbols greater than six characters having their

first six the same will generate the same symbol in the table and only the first one encountered will be effective; (e.g., CALL BCDCON(24H ABCDEFG, ABCDEFH , A, B) produces as table entries, ABCDEF vs A, and ABCDEF vs B).

2) SYMBLS: Making use of the table generated by BCDCON, SYMBLS will now read in input from the input tape, list it on the output tape, obtain the location where the data is to be stored by finding the symbol punched on the card in the BCDCON generated table, convert the data as prescribed and then store it into the core location obtained above. This process is continued until a numeric punch in column one is encountered. At this time, this number is stored away in INCOL1 of the above calling sequence example, and control is returned to the program. Automatic exit from the program to the system is obtained by calling SYMBLS and having the next card to be read have a numeral two punched in column one.

The breaking character used to separate items on the cards is a blank. Therefore, the way in which data is presented to the subroutine is completely variable as it appears in columns 2 - 72, with the exception that a symbol or a piece of data cannot be continued from one card to another. An example of how an input card might look is shown as Card 1 in Figure 3-1.

These items can be separated into three categories.

a) Core location defining symbols. These in turn may divided into:

1) The symbol - A set of characters only the first six of which are used for looking into the BCDCON table. This symbol must begin with an alphabetic character and will be terminated by a blank or a left parenthesis.

2) The subscript - Appearing to the right of the symbol it is to modify, it is enclosed in parenthesis. Multiple subscripts are separated by commas. In the case of multidimensioned variables the way in which the subscript is written will determine whether the ensuing data will be stored sequentially or not. If either no

```
COLUMNS:
1         10         20         30         40         50

Card 1.   Separate Data Items:
 A 1. B(2)   3.0 C(1,2) 4.0     D(1,1,3)      5.0E3

Card 2.   Elements of an Array - First Method:
 D(1,1,2)   3.0  4.0   (1,2,2)    5.0 6.0     (4,3,2)  7.0 8.0

Card 3.   Elements of an Array - Second Method:
 D( 2, 1-3,  3 )     5.0

Cards 4 and 5.  Decimal and Octal Data:

     B    10.
Ø  4321   57030

Cards 6 and 7.  Alpha-Numeric Data in an Array:

A   B(1)  3
ABCD FGH JKLMN Ø P

Card 8.  Single Storage of Alpha-Numeric Data:
 B(3)   /ABCD      /EFG

Card 9.  Table Generating Data:
 B   1.0(.5)2.5(-2.5E-3)2.49
```

Figure 3-1. Sample Input Cards

subscript is given or a subscript is given which has conformity to the dimensions given (e.g., D or D (1, 2, 3) where it is dimensionalized (10, 10, 10)), then the ensuing data will be stored starting in the core location defined by the symbol modified by the subscript, if given, by varying the right most subscript. If a single subscript is given to a multidimensioned variable (e.g., D(3) where H is dimensionalized (10, 10, 10)), then the ensuing data will be stored sequentially.

In addition, a subscript may appear by itself. In this case it will modify the current core location as determined by the most recent symbol. As an example, refer to Card 2 of Figure 3-1 and the BCDCON calling sequence. The card shown will generate the same data as the following FORTRAN statements:

```
H(1, 1, 2) = 3.0
H(1, 1, 3) = 4.0
H(1, 2, 2) = 5.0
H(1, 2, 3) = 6.0
H(4, 3, 2) = 7.0
H(4, 3, 3) = 8.0
```

There is also a provision for using subscripts to define areas to be set to a value defined by a piece of data which follows. See Card 3 of Figure 3-1 for example. This is equivalent to the FORTRAN statements:

```
    DO  1  I = 1,3
  1 H (2, I, 3) = 5.0
```

In general, the three possible forms this provision can take are: D(I, J, K1-K2); D(I, J1 - J2, K); and D(I1 - I2, J, K) where all the subscripts are unsigned integers and KI<K2, J1<J2, and I1<I2.

The first subscript is separated from the second by a minus sign in that subscript which defines the area to be set. There is no provision for varying more than one subscript in a single definition; (i.e., D(I1 - I2, J1 - J2, K) is not permissible).

Finally, in the subscript field it is not necessary that the characters be packed. That is, blanks are permissible, as shown in Figure 3-1.

b) Data defining fields. These can be divided into five items.

1) Floating point numbers - These will be defined by the appearance of a decimal point or a decimal point and an E in the field. They can be negative or positive in value, as can the integral exponent. When the E notation is used, if no sign appears between the E and the exponent, or if a single blank separates these two items, a plus will be assumed. In general, there are three permissible forms:

±X. XXXXXXXE±YY

±X. XXXXXXX±YY

±X. XXXXXXX

Except for the optional blank after the E to denote a plus sign, there must be no imbedded blanks. The decimal point can appear anywhere in the field. The exponent can only be integral. All the X's and Y's above are numeric. There can be one or more X's defined, but only the first eight will be converted. There can be one or two Y's.

2) Integers - These will be defined by the lack of a decimal point in the field. They can be negative or positive in value. If there is no sign, a plus is assumed. It is assumed the integer will not exceed 32767. It will be stored in the decrement portion of the core location defined by the symbol preceding it. The only permissible form is:

XXXXX

where all the X's are numeric. There can be one to five X's.

3) Octal numbers - These are defined in a different way than the preceding two pieces of data. An Ø in column one signifies to SYMBLS that columns 2-72 of this card contain only octal data and nothing else. The words of octal data can be one to twelve numbers long, and in all cases are right justified when stored. There are no checks for the appearance of eights or nines in the fields. Again the octal numbers to be stored are separated by blanks. Where these numbers will be stored will be determined by the last previously defined symbol. Cards 4 and 5 of Figure 3-1, with reference to the BCDCON calling sequence, will produce the equivalent to the following FORTRAN statements:

```
        F(1) =  10.0
B       F(2) =  4321
B       F(3) = 57030
```

4) Alpha-numeric data - There are three possible ways in which to define this type of data.

a) Hollerith card intended for use with HDTAPE (discussed below). This type of card is signified by an H in column one. Columns 2-72 will be read into a specially set aside block as hollerith characters. The H will be changed to a one. Nothing else can be done with the card.

b) Alphas-numeric data intended for arrays. There are two cards associated with this option. The first card must have an A in column one, a location defining symbol and a word count (1 to 12) anywhere in columns 2-72. The second card, which must follow immediately, will contain the alpha-numeric data to be stored. After the number of words specified in the first card has been processed

and stored, the rest of the card is ignored. Additional groups of 12 or less words each must be entered in the same manner as the first group of 12 words using two cards. Cards 6 and 7 of Figure 3-1, with reference to the BCDCON calling sequence, will produce the equivalent to the following FORTRAN statements:

```
F (2) = 6HABCD F
F (3) = 6HCH  JKL
F (4) = 6HMN Ø P
```

c) Alpha-numeric data intended for single storage. This type of data is processed in a manner more similar to floating point numbers and integers. The symbol slash (/) signifies the beginning of such a field and the first blank encountered terminates the processing of the field. The maximum number of characters allowable is six. If there are less than six, the characters defined are left justified and the remainder of the word is filled with hollerith blanks. The word is stored in the last previously defined core location. Card 8 in Figure 3-1, with reference to the BCDCON calling sequence, will produce the equivalent to the following FORTRAN statements.

```
F (3) = 4HABCD
F (4) = 3HEFG
```

5) Table generating data - This is a feature which will allow the input data to be generated, with a minimum of punching, when numbers are desired ranging from a lower limit to an upper limit in steps of certain given deltas. The data specified can be either floating point numbers or integers,

but cannot be mixed. The expression defining the numbers to be generated must be packed (i.e., no imbedded blanks) since a blank terminates the field. There can be as many deltas as desired, and they can be both positive and negative. The general form of the expression is:

$$X_1(X_2)X_3(X_4)X_4 \ldots X_{m-2}(X_{m-1})X_m$$

This will generate from $X_1$ to $X_3$ in steps of $X_2$.
$X_3$ to $X_5$ in steps of $X_4$, etc.
down to $X_{m-2}$ to $X_m$ in steps of $X_{m-1}$.

Referring to Card 9 in Figure 3-1 and the BCDCON calling sequence, the card shown will produce the equivalent of the following FORTRAN statements.

```
      F(1)     = 1.0
      DO 1 I   = 1, 3
1     F(I+1)   = F(I) + 0.5
      DO 2 I   = 4, 7
2     F(I+1)   = F(I) - 0.0025
```

c) Operation instructions:

1) Transfer card - This card will cause transfer to the calling program. It is signified by a numeral one punched in column one. Return will be made to the caller, only after columns 2-72 have been normally processed, and the numeric value punched in column one has been stored, as an integer, in the only item in the list of the calling sequence.

2) End-of-job card - This card, which consists of a numeral two punched in column one, signifies to SYMBLS that there is no more data to be processed and that the caller wishes to terminate his run. The routine will then transfer to EXIT.

In addition to the above described items, there are certain internal controls that effect the running of SYMBLS:

a) Printing suppression - The appearance of the characters *NOPRINT in columns 1-8 will suppress the listing of succeeding cards on the output tape. The appearance of *PRINT in columns 1-6 will cause SYMBLS to resume the listing of input cards on the output tape. There is no need for a *PRINT card if you want all your input listed, since SYMBLS is normally in the listing mode. Anything may appear on the rest of these two control cards, but it will be entirely ignored.

b) On-line comment - The appearance of an * in column one of a card will cause the entire contents of this card to be printed off-line and on-line, as long as the card does not have NOPRI in columns 2-6 or PRINT in columns 2-6. Data appearing on this card will not be transferred to storage.

c) Off-line comment - The appearance of a $ any place in columns 2-72 will cause the rest of this card to be printed off-line if the *PRINT option is effective. Data appearing after the $ will not be transferred to storage.

d) COMMON symbol - The appearance of COMMON as a symbol on a card will cause ensuing data to be stored in the FORTRAN defined common area. This symbol can have only single subscripts.

## Errors

The only error which is caught by symbols is the nonexistence of a data card symbol in the BCDCON generated table. All such errors encountered up to and including the next transfer card will be indicated by both on-line and off-line print-outs. After the first error is found, listing of the rest of the input will be deleted. When the transfer card is encountered and checked, then the execution of the program will be deleted and return made to the system.

3) HDTAPE

This will write on a specified tape the hollerith card explained above. The tape is specified by an integer, such as NUMTAP in the calling sequence example. Since a one replaced the original H, the off-line printer will restore the page, if under program control. If an H card was not read in by SYMBLS the printer will merely restore a page.

Errors

The appearance of an illegal tape number in the calling sequence will cause the same errors to be indicated as in FORTRAN decimal tape output statements.

# 4. INPUT

## 4.1 General Description

A description of the input routine, VFI, that is incorporated in the TOS deck is given in Section 3. A brief review of the specific features used by TOS follows.

TOS generates a table of symbols versus core locations (via the input routine). By using this table, the input routine reads a data item from the input tape, obtains the location where the item is to be stored by finding the symbol punched on the card in the table of symbols, and then stores the item in the proper location. This process continues until a numeric punch in column one is encountered. At this point, control is returned from the input routine to TOS. If a 1 is punched in column one, computation will start. A 2 or greater will terminate the program. Optionally, each card that is read is listed on the output tape. The input routine is called following the completion of each problem; a new problem may be initiated by merely entering the data items that are different from those currently stored.

The breaking character used to separate items on the cards is a blank. Therefore, the way in which data is presented is completely variable as it appears in column 2-72, with the exception that a symbol or a piece of data cannot be continued from one card to another.

Input items are of two categories:

1) Core location defining symbols - A symbol is a set of up to six characters. The symbol may be modified by a subscript. The subscript appears to the right of the symbol. TOS uses only single subscripts. Data following a symbol is stored sequentially starting in the core location defined by the symbol, modified by the subscript if any.

2) Data fields

a) Floating-point numbers - These numbers are defined by the appearance of a decimal point, or a decimal point and an E, in the field. They can be negative or positive in value, as can the integral exponent. If no sign separates E and the exponent, or if these two items are separated by a single blank, a plus sign is assumed. Three permissible forms are:

± x. xxxxxxxE±YY

● x. xxxxxxx+YY

± x. xxxxxxx

Aside from the optional blank after the E to denote a plus sign, there must be no imbedded blanks. The decimal point can appear anywhere in the field. All TOS input data must be floating-point numbers except in the IGO block or where specifically shown to the contrary.

b) Integers - These are defined by the lack of a decimal point in the field. They can be negative or positive in value.

c) Alpha-numeric data - Two cards are associated with this option. The first card must have an A in column one, and a location defining symbol and a word count (1 to 12) anywhere in columns 2-72. The second card, which must follow immediately, contains the alpha-numeric data in desired format beginning in column 1. After the number of words specified in the first card has been processed and stored, the rest of the second card is ignored. If more than 12 words are to be stored, each additional group of 12 or less words must be entered by two cards in the same manner.

## 4.2 Input Symbols

A number following a definition is a nominal value which need not be inputted unless changed. These nominal values are incorporated as a convenience for the user, but they are not necessarily recommended for use in all problems. The input area of the program is initially cleared to zero, then the nominal values stored. Once a nominal value has been changed it remains changed for succeeding problems in the same sequence unless reset by input.

| Symbol | Definition | |
|---|---|---|
| CASENO(1) | Case number xxxxxxxx. | |
| CASENO(2) | Iteration number xx. Set to zero before input is read for every case. | |
| A ID(1) 3 | 18 characters of identification (e.g., engineer's name and date) starting in column 1 of next card. | |
| VEL | Initial velocity magnitude in ft/sec. | |
| GAMMA | Initial flight-path angle in deg. | |
| BETA | Initial heading angle in deg. | |
| H | Initial height in ft. | |
| THETA | Initial co-latitude in deg. | |
| PHI | Initial longitude in deg. | |
| GS | Reference area during glide phase in $ft^2$. | |
| GMASS | Mass during glide phase in slugs. | |
| PAYOFF | Library number of payoff function. Sign of number determines whether to maximize (+) or minimize (-). | |
| STOP | Library number of stopping function. + if the function is to be increasing, - if the function is to be decreasing when reaching the stopping value. | |
| STOPV | Stopping value. | |
| PRTINT | Print interval for forward trajectory in sec. | |
| NIT | Number of iterations desired; 0 implies forward trajectory only. | |
| DPSQ | Steepest-ascent step size $(dP)^2$. | |
| DPINC | Constant for increase of $(dP)^2$. | 2.0 |
| DPDEC | Constant for decrease of $(dP)^2$. | 0.50 |
| PSIEG | Constant dividing excellent and good values of $\Psi$. | 0.85 |

| Symbol | Definition | |
|---|---|---|
| PSIGP | Constant dividing good and poor values of $\Psi$. | 0.85 |
| PSIRJ | Constant for reject of iteration based on $\Psi$. | 0.1 |
| PHIGE | Constant (> 1) dividing good and excellent $\underline{\Phi}$. | 1.05 |
| PHIEG | Constant (< 1) dividing excellent and good $\underline{\Phi}$. | 0.85 |
| PHIGP | Constant dividing good and poor $\underline{\Phi}$. | 0.40 |
| PHIRJ | Constant for reject of iteration based on $\underline{\Phi}$. | 0.1 |
| NDELF | Number of times $\lvert \Delta \underline{\Phi} \rvert < \Delta \underline{\Phi}_{MIN}$ before stop. | 0.0 |
| CDELM | Minimum actual composite gradient. | 0.0 |
| DPSQM | Minimum $(dP)^2$ before stop. | 0.0 |
| KQC | Constant for convective heating. | |
| RQC | Constant for convective heating. | |
| NQC | Constant for convective heating. | |
| KQR | Constant for radiative heating. | |
| RQR | Constant for radiative heating. | |
| NQR | Constant for radiative heating. | |
| AH1 | Constant for altitude penalty function. | |
| AH2 | Constant for altitude penalty function. | |
| AQD1 | Constant for heat-rate penalty function. | |
| AQD2 | Constant for heat-rate penalty function. | |
| SLPE | Sea level atmospheric pressure. | 2116.2 |
| MU | Gravitation constant in $ft^3/sec^2$. | $1.407698 \times 10^{16}$ |
| J | Gravitation constant. | $1.62341 \times 10^{-3}$ |
| RF | Reference radius of earth for gravitational expansion in ft. | 20925631. |
| OMEGA | Angular velocity of earth in radians/sec. | $7.29211508 \times 10^{-5}$ |
| RE | Radius at equator of oblate spheroid in ft. | 20926428. |
| RPC | Radius at pole of oblate spheroid in ft. | 20855965. |
| TLIM | Time limit bound on forward trajectory in sec. | 9999. |
| HLIM | Height limit bound on forward trajectory in ft. | $1.0 \times 10^6$ |
| ORDA | *Order of Adams-Moulton forward. | 2.0 |
| HMOMA | *Initial step size forward in sec. | 0.01 |
| EUA | *Upper bound on Adams-Moulton error forward. | $1.0 \times 10^{-5}$ |
| ELA | *Lower bound on Adams-Moulton error forward. | $1.0 \times 10^{-7}$ |

*Constants for JP MARK

| Symbol | Definition | |
|---|---|---|
| HMAXA | *Maximum step size forward in sec. | 5.0 |
| HMINA | *Minimum step size forward in sec. | 0.005 |
| YCA | *YCLOW forward. | 0.001 |
| ORDB | *Order of Adams-Moulton adjoint. | 2.0 |
| HNOMB | *Initial step size adjoint in sec. | 0.01 |
| EUB | *Upper bound on Adams-Moulton error adjoint. | $1.0 \times 10^{-3}$ |
| ELB | *Lower bound on Adams-Moulton error adjoint. | $1.0 \times 10^{-5}$ |
| HMAXB | *Maximum step size adjoint in sec. | 5.0 |
| HMINB | *Minimum step size adjoint in sec. | 0.005 |
| YCB | *YCLOW adjoint. | 0.001 |
| LAM | LAMCOS-DO equal interval indicator. A number between 2.0 and 10.0. LAMCOS-DO will be performed this many times at equal intervals. If 0. use LAMCOS-DO table. | 10. |
| LAMT(1) | LAMCOS-DO table of up to 9 values of time in sec. LAMCOS-DO will be performed at these times (if LAM = 0.). The times must be in increasing order and if less than 9 values are used the last must be 0.0. If LAMT(1) = 0.0, use LAM. | 0.0 |
| ALPHA(1) | Nominal control variable program of $\alpha$ vs time. Up to 50 values of time monotonically increasing (may be =) in sec. There must be at least 2 values and if 50 values are not used the end of the table is indicated by a 0.0. [See DELTA(1).] | |
| ALPHA(51) | Up to 50 values of $\alpha$ in deg. | |
| SIGMA(1) | Nominal control-variable program of $\sigma$ vs time. Up to 50 values of time in sec. [See ALPHA(1).] | |
| SIGMA(51) | Up to 50 values of $\sigma$ in deg. | |
| CON(1) | Library numbers of up to 8 constraints (see list of payoff, stopping, and constraint functions). If no constraints are desired put 0. in CON(1). If less than 8 values are used the table is terminated by a 0. | |

*Constants for JP MARK

| Symbol | Definition |
|---|---|
| SIT(1) | Upper limits on constraints. |
| SIB(1) | Lower limits on constraints. |
| EPS(1) | Small tolerances on constraint limits. |
| EOST(1) | End of stage times in sec. Three boost stages are provided. If less than 3 stages, the table is terminated by a 0. 0. If only glide stage, set EOST(1) to 0. and omit the next seven items on this input list. 0. 0 |
| AE(1) | Exit areas for up to 3 boost stages in $ft^2$. |
| S(1) | Reference areas for up to 3 boost stages in $ft^2$. |
| IT(1) | $i_T$ for up to 3 boost stages in deg. |
| DT(1) | $\delta_T$ for up to 3 boost stages in deg. |
| TMT(1) | Thrust and mass tables versus time for boost stages. Up to 50 values of time in seconds. If less than 50, the last must be 0. 0. There must be at least 2 values (if the table is used) and they must be monotonically (may be equal) increasing. This table is for all stages. |
| TMT(51) | Up to 50 values of sea level thrust in lbs. |
| TMT(101) | Up to 50 values of mass in slugs. |
| MCLCD(1) | Lift and drag coefficients, $C_L$ and $C_D$, as a function of mach number and $\alpha$. There are tables for 4 stages, the 3 boost, if any, and the glide stage. 1st stage mach number tables. Up to 10 values (monotonically increasing) followed by 0. 0. |
| ACLCD(1) | 1st stage $\alpha$ tables in deg. Up to 25 values (monotonically increasing) followed by 0. 0. |
| CL(1) | Lift coefficient $C_L$ as functions of mach number and $\alpha$ for 1st stage. The order of the $C_L$ table is with the mach number varying first. $C_L(M_1, \alpha_1)$, $C_L(M_2, \alpha_1)$, $C_L(M_3, \alpha_1)$, . . . $C_L(M_1, \alpha_2)$, . . . |
| CD(1) | Drag coefficients $C_D$ as functions of mach number and $\alpha$ for 1st stage. |
| MCLCD(12) | 2nd stage mach numbers. |
| ACLCD(27) | 2nd stage $\alpha$. |
| CL(151) | 2nd stage $C_L$. |

| Symbol | Definition |
|---|---|
| CD(151) | 2nd stage $C_D$. |
| MCLCD(23) | 3rd stage mach numbers. |
| ACLCD(53) | 3rd stage $\alpha$. |
| CL(301) | 3rd stage $C_L$. |
| CD(301) | 3rd stage $C_D$. |
| MCLCD(34) | 4th stage mach numbers. |
| ACLCD(79) | 4th stage $\alpha$. |
| CL(451) | 4th stage $C_L$. |
| CD(451) | 4th stage $C_D$. |

Notes: (1) The stages are numbered as they occur during the forward trajectory. That is, for no boost stages, the glide stage is the 1st stage, and the other stages are not used. If 3 boost stages, glide stage is the 4th stage.

(2) As an additional restriction, the number of values in the MCLCD table for a particular stage times the number of values in the ACLCD table for that stage must be less than or equal to 150.

(3) Only positive values of $\alpha$ are needed; the relationships $C_L(-\alpha, M) = -C_L(\alpha, M)$ and $C_D(-\alpha, M) = C_D(\alpha, M)$ are used.

(4) $C_L$ and $C_D$ are computed by bivariate linear interpolation using SUBROUTINE CLCD.

WT(1) Weighting matrix $\begin{vmatrix} w_{11}(t) & 0 \\ 0 & w_{12}(t) \end{vmatrix}$. Up to 50 values of time in seconds. If less than 50 the last must be 0.0. There must be at least 2 values and they must be monotonically increasing (may be equal). An identity matrix (i.e., $w_{11}=w_{22}=1.0$) implies no weighting. This identity matrix is a nominal input in the program.

WT(51) $w_{11}$.

WT(101) $w_{22}$.

TAU(1) Acceleration tolerance function. Up to 50 values of acceleration in g's. If thess than 50 the last must be 0.0. There must be at least two values (if the table is used) and they must be monotonically increasing (may be equal).

TAU(51) Time of tolerance.

Output Options:

PRTOPT(1) 0.0 implies print last acceptable control-variable table
1.0 implies print all control-variable tables

PRTOPT(2) 0.0 implies punch no control-variable tables
1.0 implies punch last acceptable control-variable table
2.0 implies punch all control-variable tables

PRTOPT(3) 0.0 implies no adjoint print
1.0 implies print adjoint solution
2.0 implies print adjoint solution and F and G matrices

PRTOPT(4) 0.0 implies do not punch complete input data
1.0 implies punch the complete input data

The nominal values for all output options are 0.

PRTOPT(2) = 1.0 or 2.0 punches the control variable tables in the format:

DELTA(1) $\Delta t$ between values in control-variable tables generated by the program. This value is set to 0.0 just before the input for a case is read. If a value $\neq 0.0$ is inputted it indicates the control variables should be computed from these tables rather than from the nominal control-variable tables, which should be omitted from the input deck.

ALPHAX(1) $\alpha$ in degrees (201 values)

SIGMAX(1) $\sigma$ in degrees (201 values)

PRTOPT(4) = 1.0 punches the complete input for a problem in a format that is compatible with the input routine for use in restarting the problem, if required.

The output of the forward trajectory, as well as the provision for blocks of supplementary printout following the forward and adjoint trajectories, is controlled entirely through input via subroutines OUT and ITEM. A knowledge of subroutine OUT is necessary for an understanding of this input. A nominal input for these blocks, as illustrated in Figure 4-1, gives a reasonable complete printout that will prove useful for many problems.

| | | | |
|---|---|---|---|
| A | HFMT(1) | 12 | Format statement for title of forward trajectory. Up to 40 words. Alpha-numeric data. |
| A | DFMT(1) | 12 | Format statement for data used in title of forward trajectory. Up to 40 words. Alpha-numeric data. |
| | IGO(1) | | Print indicator; set to 1 |
| | IGO(2) | | Page title indicator; set to 1 |
| | IGO(3) | | Number of lines produced by heading format |
| | IGO(4) | | Number of lines produced by data format |
| | IGO(5) | | Number of data items |
| | IGO(6) | | Data item table for use with item function. See write-up of subroutine OUT. All data in the IGO blocks are integers, i.e., without decimal points. Up to 40 words or 20 items. |
| A | HFMT(41) | 12 | Format statement for column heading. Up to 100 words. |
| A | DFMT(41) | 12 | Format statement for data used with above heading. Up to 100 words. |
| | IGO(86) | | Print indicator; set to 1 |
| | IGO(87) | | Column heading indicator; set to 2 |
| | IGO(88) | | Number of lines produced by column headings. |
| | IGO(89) | | Number of lines produced by data format |
| | IGO(90) | | Number of data items |
| | IGO(91) | | Data item table for use with ITEM function. Up to 80 words or 40 items. |
| A | HFMT(141) | 12 | Format statement for title of supplementary data printed following forward trajectory. Up to 40 words. |
| A | DFMT(141) | 12 | Format statement for supplementary data printed following forward trajectory. Up to 40 words. |
| | IGO(171) | | Print indicator; set to 1 if data are to be printed, otherwise to 0. |

| | | |
|---|---|---|
| | IGO(172) | Use similar to IGO(2) . . . IGO(6) except for supplementary data following forward trajectory. Up to 80 words or 40 items. |
| | . . . | |
| | IGO(176) | |
| A | HFMT(181) 12 | Same as preceding items except for data printed following adjoint trajectory. |
| A | DFMT(181) 12 | |
| | IGO(256) | |
| | . . . | |
| | IGO(261) | |

Each item is composed of 2 integers, (I, J). J refers to the Jth labelled common while I refers to Ith item in the Jth labelled common.

TOS has 4 main labelled common areas which are described in Appendix 3:

1. COMMON / INPUT /
2. COMMON / VAR /
3. COMMON / ANAL /
4. COMMON / IVAR /

For example, in Figure 4-1 the intergers 196 2 following IGO(91) indicate that the independent variable t, the 196st item in the 2nd labelled common block will be the first item of data printed.

## 4.3 List of Payoff, Stopping, and Constraint Functions

| Number | |
|---|---|
| 0 | null |
| 1 | t - independent variable, time in seconds |
| 2 | u - component of velocity in ft. /sec. |
| 3 | v - component of velocity in ft. /sec. |
| 4 | w - component of velocity in ft. /sec. |
| 5 | r - radial distance in ft. |
| 6 | $\theta$ - colatitude, in deg. |
| 7 | $\phi$ - longitude in deg. |
| 8 | heat |
| 9 | acceleration penalty function |

```
A HFMT(1)        12
(107H  RAYTHEON THREE-DIMENSIONAL TRAJECTORY OPTIMIZATION PROGRAM  TOS9   (1)*
A HFMT(13)       12
  FORWARD TRAJECTORY        CASENO    ITNO     )
A DFMT(1)        12                                                      (2)
(92X,F8.0,F7.1,2X,3A6)
A HFMT(41)       12
(116H0    TIME       VEL    GAMMA     BETA         H      THETA       PHI    TH
A HFMT(53)       12
ETA*      PHI*       MU        NU      ALPHA     SIGMA    /                (3)
A HFMT(65)       12
114H      SEC     FT/SEC    DEG       DEG          FT     DEG         DEG    DEG
A HFMT(77)       12
         DEG       DEG       DEG       DEG       DEG  //
A HFMT(89)       12
129H                ACCEL    DYN PRESS       ENERGY        HEAT      ACCEL PF
A HFMT(101)      12
     ALT PF     HT RT PF     CL        CD    MACH      HC      RGERR     /   (4)
A HFMT(113)      12
119H                 GS      LB/FT**2        FT-LB
A HFMT(125)      12
                                                   SEC          )
A DFMT(41)       12
( / F9.2,F9.1,2F8.2,F10.0,8F9.3 / F18.2,1P6E12.4,0P2F8.3,F7.2,F7.3,   (5)
A DFMT(53)       12
E11.3 )
 IGO(1)  1  1  1  1  5                                                   (6)
 IGO(6)  1  1  2  1   3  1  4  1  5  1                                   (7)
 IGO(86)  1 2 6 3 25                                                     (8)
 IGO(91)  196 2 24 2  184  2  183 2  20 2  185 2  186 2  182  2  181  2
 179 2 180 2 11 2 12  2  63 2  187 2  188 2  204 2  205 2  206 2  207 2   (9)
 66 2 67 2 28 2  189 2  190 2
 IGO(171)  0                                                             (10)
 IGO(256)  0                                                             (11)
```

*EXPLANATION:

(1) Page title
(2) Format for data on line following title
(3) First two lines of column headings; third line is blank
(4) Second two lines of column headings
(5) Format for two lines of data at each print time
(6) Control for page title
(7) Data table for line following title
(8) Control for columns of data
(9) Data table for columns of data
(10) Control for printout following forward trajectory
(11) Control for printout following adjoint trajectory

Figure 4-1. Typical Input for Control of Printout

| Number | |
|---|---|
| 10 | altitude penalty function |
| 11 | heat rate penalty function |
| 12 | null |
| 13 | null |
| 14 | $\mu$ - coordinate angle in deg. |
| 15 | $\nu$ - coordinate angle in deg. |
| 16 | $\theta^*$ - coordinate angle in deg. |
| 17 | $\phi^*$ - coordinate angle in deg. |
| 18 | h - altitude in ft. |
| 19 | $V_T$ - velocity in ft./sec. |
| 20 | E - energy in ft.-lbs. |
| 21 | $\gamma$ - flight path angle in deg. |
| 22 | $\beta$ - heading angle in deg. |
| 23 | dynamic pressure in lbs./ft.$^2$ |

# 5. OUTPUT

## 5.1 Input

The input cards are listed by the input routine as they are read, if the print option of VFI is operative. The main program provides for the printing of the entire input applicable for the case, at the start of the case. Print option 4, PRTOPT(4), controls the punching of cards giving the entire input. If PRTOPT(4) = 0., no cards are punched; if PRTOPT(4) = 1., the entire input is punched for use in restarting the case.

## 5.2 Forward Trajectory

The output from the forward trjectory is entirely flexible as it is controlled through input. Section 4.2 of this report describes the format for this input. SUBROUTINE OUT is the routine which controls this output.

## 5.3 Adjoint Trajectory

Print option 3, PRTOPT(3), controls the output of the adjoint trajectory. If PRTOPT(3) = 0. the adjoint print output is omitted completely. PRTOPT(3) = 1. prints the adjoint differential equations, and PRTOPT(3) = 2 prints the adjoint differential equations, their derivatives, and the F and G matrices.

## 5.4 Analysis

At the end of each forward trajectory, data concerning the payoff and constraint functions are printed together with measures of performance of the steepest-ascent convergence process and related information.

## 5.5 Control Variable Programs

The control variable output is controlled through print options 1 and 2, PRTOPT(1) and PRTOPT(2). If PRTOPT(1) = 0. only the control variable programs associated with the last acceptable trajectory are printed. If PRTOPT(1) = 1., all acceptable control variable programs are printed. If PRTOPT(2) = 1., the last acceptable control variable program is punched. PRTOPT(2) = 2. punches all acceptable control variable programs. The control variable programs are punched in a form acceptable to the input routine and when the cards are used, they will override any nominal control programs. PRTOPT(2) = 0. eliminates this punching.

## 5.6 Miscellaneous Data

At the conclusion of the forward and adjoint trajectories, the user may elect to have any data then in storage printed out. The format for this print out is controlled through input; see Section 4.2.

# 6. SYSTEM INFORMATION

## 6.1 Language

TOS is primarily written in FORTRAN IV for 32K 7090/7094 IBSYS.

## 6.2 Overlay

TOS uses the overlay feature of IBSYS. The deck arrangement is as follows:

| (LINK1) | (LINK2) | (LINK3) |
|---|---|---|
| | (LINK0) | |
| | TOS | |
| | CLD | |
| | PRL | |
| | FCT | |
| | EMK | |
| | CLR | |
| | TML | |
| | MVL | |
| | LIN1 | |
| | BLN1 | |
| | ATM1 | |
| | MIV | |
| | SMK1 | |
| | MARK | |
| PIN | STE | ADT |
| CFI | ITM | |
| | OUT | |
| | SFR | |

## 6.3 File Specification for Scratch Tape

The file specifications for the scratch tape, logical 8, are

FILE, B(1), BIN, BLK = 256, INOUT, READY

# 7. RESTRICTIONS

There are several "singularities" in the equations of motion. The colatitude, $\theta$, and the velocity $V_T$, must never equal zero during the entire trajectory. The flight path angle, $\gamma$, must never equal 90 degrees.

The control variable programs are limited to 201 equal-time-spaced values for each $\alpha$ and $\sigma$ because of storage consideration; consequently, problems to be solved are restricted to trajectories where sufficient accuracy can be obtained using 201 values for each of these variables.

# 8. PROGRAM LISTING

A complete listing of the TOS program is given on the following pages.


| Routine | | Page |
|---|---|---|
| TOS | . . . | 8-3 |
| PIN | . . . | 8-13 |
| STE | . . . | 8-20 |
| ADJ | . . . | 8-28 |
| CLD | . . . | 8-39 |
| PRL | . . . | 8-40 |
| FCT | . . . | 8-42 |
| SFR | . . . | 8-45 |
| ITM | . . . | 8-47 |
| TML | . . . | 8-49 |
| MVL | . . . | 8-50 |
| MIV | . . . | 8-51 |
| OUT | . . . | 8-53 |
| VFI | . . . | 8-55 |
| MARK | . . . | 8-70 |
| CLR | . . . | 8-100 |
| EMK | . . . | 8-101 |
| UN8 | . . . | 8-102 |
| SMK1 | . . . | 8-103 |
| ATM1 | . . . | 8-107 |
| BLN1 | . . . | 8-112 |
| LIN1 | . . . | 8-115 |

```
$IBFTC TOS
C TITLE          TRAJECTORY OPTIMIZATION STUDY (7094)
C
C PURPOSE        MAIN PROGRAM
C
C NOTES          FORTRAN IV
C
C     C O M M O N   F O R   A L L   P R O G R A M S
C
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      COMMON / VAR / RADIAN,D10,D11,D12,D13,D14,D20,D30,D31,D32,
     1ALPHD,SIGMD,ALPHR,SIGMR,CSIGM,SSIGM,CTHET,STHET,R,H,VHSQ,
     2VTSQ,VH,VT,PEA,RHO,SOFS,MACH,MASS,TH,THR,COSIT,SINIT,COSDT,
     3SINDT,THRX,THRY,THRZ,CALPH,SALPH,C1,C2,C3,C4,C5,C6,THRR,THRT,
     4THRP,S,X1,X2,X3,X4,X5,X6,X7,X8,X9,RSQ,R4TH,SQRHO,ACELG,TAU,
     5TIMEH,CL,CD,TFINAL,B56,B58,PEH,RHOH,SOFSH,PEL,RHOL,SOFSL,
     6DPERH,DRRH,DSRH,CLHM,CDHM,CLLM,CDLM,PCLRM,PCDRM,DRPRT,B10,
     7B20,B21,B30,B31,B40,B41,B50,B51,B52,B53,B54,B55,B57,B60,B61,
     8B62,B70,B71,B81,B82,B90,B91,B92,B93,B100,B101,B102,B103,B104,
     9R3RD,R5TH,VT3,B200,B201,VH3,B202,B204,PTRRU,PTTRU,PTPRU
      COMMON / VAR / PTRRV,PTTRV,PTPRV,PTRRW,PTTRW,PTPRW,PTHRH,
     1PTXRH,PTYRH,PTZRH,B300,B301,B302,PTRRR,PTTRR,PTPRR,B404,
     2B405,B400,SCLCD,DTRA,B401,B402,B403,B406,B407,PTRRA,PTTRA,
     3PTPRA,PTRRS,PTTRS,PTPRS,E1,E2,E3,E4,E5,E6,CLAL,CDAL,CLAH,
     4CDAH,PCLRA,PCDRA,B130,B131,PTRG,STRG,DTRG,GTRG,SRAT,
     5MUD,NUD,PHISD,THETSD,BETAD,GAMMAD,THETD,PHID,DYNA,ENER,AST,RG
      COMMON / VAR / SFRM(1500),GL(9,2,201),ATABS(201),STABS(201),
     1PARTS(14),PARTC(14),CADJ(6,9),F(6,6),CF(6,6),G(6,2),CG(6,2)
      COMMON / VAR / LAMP(9),VARX(25)
      REAL MACH,MASS,LAMP
      COMMON / ANAL / HI,HIS,SIA(8),SIAS(8),SIE(8),SIES(8),CSIA,
     1CSIAS,CDSIP,CDSIA,CRDSI,CHIA,CHIAS,CDHIA,CDHIP,CRDHI,DSIA(8),
     2RDSI(8),DHIA,DHIP,RDHI,DP,EHI,CEHI,INTGL(9,9),IHH,ISH(8),
     3ISS(8,8),WT(2,2),F1,F2,SID(8),SFC,DSI(8),DSIJ,DBETA(8),
     4ISSIL(8,8),DPSQ,DPSQK,IHHK,ISHK(8),ISSK(8,8),
     5ISSILK(8,8),DBETAK(8),STVRS(12,9),LAMBDA(6,8,9),SFCK,F1K,
     6F2K
      REAL INTGL,IHH,ISH,ISS,ISSIL,INTGLK,IHHK,ISHK,ISSK,ISSILK,
     1LAMBDA
      COMMON / IVAR / JTAPE,KTAPE,LTAPE,ITN,ITTN,ISTAGE,NC,
     1NT,NRTN,NTRG,KACC,KA,KS,NIF,NG,L(8),LS(8)
      EQUIVALENCE(SFRM(2000),NORA(2000),HNA(1997),NEQA(1996),
     1TIMEA(1995),U(1993),V(1992),W(1991),RX(1990),THETR(1989),
     2PHIR(1988),PA(1987),PB(1986),PC(1985),PD(1984),PE(1983),
     3PF(1982),U1(1981),V1(1980),W1(1979),RX1(1978),THETR1(1977),
     4PHIR1(1976),PA1(1975),PB1(1974),PC1(1973),PD1(1972),PE1(1971),
```

```
     5PF1(1970))
      DIMENSION ADJ(6,9), DADJ(6,9)
      EQUIVALENCE(SFRM(2000),NORB(1981),HNB(1978),NEQB(1977),
     1TIMEB(1976),ADJ(1974),DADJ(1920))
      DIMENSION STVR(12)
      EQUIVALENCE(STVR(1),U)
C
C     E N D  O F  C O M M O N  F O R  A L L  P R O G R A M S
C
      COMMON / FORW / STRAJ(20,13)
      DIMENSION FNAME(25)
      DATA FNAME / 150HNULL  TIME  U     V     W     R     THETA PHI   H
     1EAT  ACELPFALTPF HTRTPFNULL  NULL  MU    NU    THETA*PHI*  H     V
     2EL   ENER  GAMMA BETA  DYN         /
      DATA MIN,MAX / 3HMIN, 3HMAX /
C     PRTOPT(1) PRINT CONTROL TABLES 0.=LAST, 1.=ALL
C     PRTOPT(2) PUNCH CONTROL TABLES 0.=NONE,1.=LAST,2.=ALL
C     PRTOPT(3) ADJOINT PRINT 0.=NONE,1.=ADJOINT ONLY,2.=ALL
C     VALUES SET FOR ALL CASES
    1 CONTINUE
C     CLEAR STORAGE AREAS   CALL CLEAR(FROM,TO,FROM,TO,ETC)
      CALL CLEAR(CASENO(1),IGO(340),RADIAN,VARX(25),HI,F2K,JTAPE,LS(8))
      RADIAN=57.2957795
C     NIF IS THE NUMBER OF CONTROL VARIABLE POINTS SO AS TO GIVE 200
C     EQUALLY SPACED POINTS ALONG THE TRAJECTORY
      NIF=201
C     JTAPE = PRINT OUTPUT TAPE
C     KTAPE = PUNCH OUTPUT TAPE
C     LTAPE = BUFFER TAPE
      JTAPE=6
      KTAPE=7
      LTAPE=8
C     NOMINAL VALUES FOR CERTAIN VALUES OF INPUT
      A11=2.0
      A12=.50
      A21=.85
      A22=.40
      A23=.1
      A31=1.05
      A32=.85
      A33=.40
      A34=.1
      DELF=0.0
      CEHIM=0.0
      DPSQM=0.0
      LAM = 10.
      SLPE=2116.2
      MU=1.407698E16
      RF=2.0925631E7
      GJ=1.62341E-3
      OMEGA=7.29211508E-5
      RE=20926428.
      RPC=20855965.
      TLIM=9999.
      HLIM=1000000.
      ORDA=2.0
      HNOMA=.01
```

```
      EUA=1.E-5
      ELA=1.E-7
      HMAXA=5.0
      HMINA=.005
      YCA=.001
      ORDB=2.0
      HNOMB=.01
      EUB=1.E-3
      ELB=1.E-5
      HMAXB=5.0
      HMINB=.005
      YCB=.001
      DO 10 I=1,4
   10 PRTOPT(I)=0.0
C     SUBROUTINE PRNTIN READS AND PRINTS INPUT
C     RETURN FOR NEXT CASE
 1000 CALL PRNTIN
C     VALUES SET FOR EACH CASE
      ITN=0
      NIT=AIT
      NDELF=DELF
      KA=1
      IF( XDPSQ .LT. 0.0 ) KA=2
      DPSQ=ABS(XDPSQ)
      AM=SIGN(1.0,PAYOFF)
      DO 1010 I=1,8
      SIA(I)=0.0
      SIAS(I)=0.0
      SIE(I)=0.0
      SIES(I)=0.0
      SID(I)=0.0
      DSI(I)=0.0
      DBETA(I)=0.0
      LS(I)=0
 1010 L(I)=0
C     DETERMINE NUMBER OF CONSTRAINTS
      NC=0
      DO 1020 I=1,8
      IF(CON(I) .EQ. 0.0) GO TO 1030
      NC=NC+1
 1020 L(I)=1
 1030 CONTINUE
      NT=NC+1
C     TERMS CONSTANT FOR ENTIRE CASE
      D10=MU*GJ*RF**2
      D11=4.0*D10
      D12=8.0*D10
      D13=6.0*D10
      D14=2.0*D10
      D20=2.0*MU
      D30=OMEGA**2
      D31=2.0*OMEGA
      D32=2.0*D30
C     SET UP LAMCOS-DO TABLE LAMP
C     LAMP(I)*TFINAL ARE TIMES FOR LAMCOS-DO
      IF(LAM .EQ. 0.0) GO TO 1072
      IF(LAM .LT. 2.0) LAM=2.0
```

```
      IF(LAM .GT. 10.0) LAM=10.0
      DO 1071 I=1,9
      LAMP(I)=AINT((FLOAT(I)/LAM + .001)*100.)/100.
      IF(LAMP(I) .GT. .98) LAMP(I)=0.0
 1071 CONTINUE
      GO TO 1077
 1072 DO 1073 I=1,9
 1073 LAMP(I)=0.0
C     THE VALUES IN LAMTAB TABLE MUST BE IN INCREASING ORDER
      DO 1075 I=1,9
      LAMP(I)=AINT(LAMTAB(I)/TFINAL*100.)/100.
      IF(LAMP(I) .GT. .98 .OR. LAMP(I) .EQ. 0.0) GO TO 1076
 1075 CONTINUE
      GO TO 1077
 1076 LAMP(I)=0.0
 1077 CONTINUE
C
C     SUBROUTINE STATE COMPUTES THE FORWARD TRAJECTORY
 2000 CALL STATE(NF)
C     NF ERROR INDICATOR 1=OK,2=ERROR
C     IF ERROR GET NEXT CASE
      GO TO(4000,1000),NF
C
C     ANALYZE FORWARD TRAJECTORY
C
 4000 CONTINUE
C     HEADINGS FOR ANALYSIS SECTION
      WRITE(JTAPE,9050) CASENO,ID
      WRITE(JTAPE,9051)
      MINMAX=MIN
      IF(AM .GT. 0.0) MINMAX=MAX
C     FUNCTION FUNCT COMPUTES PAYOFF AND CONSTRAINT FUNCTIONS
      HI=FUNCT(PAYOFF)
      IF(NC .EQ. 0) GO TO 4030
      DO 4020 I=1,NC
      SIA(I)=FUNCT(CON(I))
      IF(SIA(I) .LT. (SIT(I)+EPS(I))) GO TO 4010
      SIE(I)=SIA(I)-SIT(I)
      GO TO 4020
 4010 IF(SIA(I) .GT. (SIB(I)-EPS(I))) GO TO 4015
      SIE(I)=SIA(I)-SIB(I)
      GO TO 4020
 4015 SIE(I)=0.0
 4020 CONTINUE
 4030 IF(ITN .NE. 0) GO TO 4060
C     ZEROTH ITERATION
 4040 I=ABS(PAYOFF) + 1.0
      WRITE(JTAPE,9052) MINMAX, FNAME(I),HI
      IF(NC .EQ. 0) GO TO 4046
      WRITE(JTAPE,9053)
      DO 4043 I=1,NC
      J=CON(I) + 1.0
 4043 WRITE(JTAPE,9054) FNAME(J),SIA(I),L(I),SIT(I),SIB(I)
 4046 CONTINUE
C     IF ZERO ITERATIONS GO TO NEXT CASE
 4050 IF(NIT .EQ. 0)GO TO 1000
      KACC=1
```

```
C     ZEROTH ITERATION - GO DIRECTLY TO ADJOINT
      GO TO 6010
 4060 CONTINUE
C     NOT ZEROTH ITERATION
      IF(NC .EQ. 0) GO TO 4080
C     ANY CONSTRAINTS APPLICABLE
      DO 4061 I=1,NC
      IF(L(I) .NE. 0)GO TO 4070
 4061 CONTINUE
      GO TO 4080
C     CONSTRAINTS APPLICABLE
 4070 CSIA=0.0
      CSIAS=0.0
      DO 4071 I=1,NC
      CSIAS=SIES(I)**2/ISS(I,I)+CSIAS
      CSIA=SIE(I)**2/ISS(I,I)+CSIA
 4071 CONTINUE
      CSIAS=SQRT(CSIAS)
      CSIA=SQRT(CSIA)
      CDSIP=-SFC*CSIAS
      CDSIA=CSIA-CSIAS
      CRDSI=CDSIA/CDSIP
      CHIAS=HIS-TMAML(ISH,ISSIL,SIES,L)
      CHIA=HI-TMAML(ISH,ISSIL,SIE,L)
      CDHIA=CHIA-CHIAS
      IF(F1 .EQ. 0.0)GO TO 4072
      CDHIP=AM*SQRT((DPSQ-TMAML(DBETA,ISSIL,DBETA,L))
     1*(IHH - TMAML(ISH,ISSIL,ISH,L)))
      CRDHI=CDHIA/CDHIP
      GO TO 4090
 4072 CDHIP=0.0
      CRDHI=0.0
      GO TO 4090
 4080 CSIA=0.0
      IF(NC .EQ. 0) GO TO 4082
      DO 4081 I=1,NC
 4081 CSIA=SIE(I)**2/ISS(I,I)+CSIA
      CSIA=SQRT(CSIA)
 4082 CSIAS=0.0
      CDSIP=0.0
      CDSIA=CSIA
      CRDSI=0.0
      CHIA=HI
      CHIAS=HIS
      CDHIA=CHIA-CHIAS
      CDHIP=AM*SQRT(DPSQ*IHH)
      CRDHI=CDHIA/CDHIP
 4090 CONTINUE
      IF(NC .EQ. 0)GO TO 4100
      DO 4094 I=1,NC
      DSIA(I)=SIA(I)-SIAS(I)
      RDSI(I)=DSIA(I)/DSI(I)
 4094 CONTINUE
 4100 DHIA=HI-HIS
      DHIP=CDHIP+TMAML(ISH,ISSIL,DBETA,L)
      RDHI=DHIA/DHIP
      DP=SQRT(DPSQ)
```

```
      EHI=DHIA/DP
      CEHI=CDHIA/DP
      I=ABS(PAYOFF) + 1.0
      WRITE(JTAPE,9055) MINMAX,FNAME(I),HIS,HI,DHIP,DHIA,RDHI
      IF( NC .EQ. 0) GO TO 4121
      WRITE(JTAPE,9053)
      DO 4120 I=1,NC
      J=CON(I) + 1.0
      WRITE(JTAPE,9056) FNAME(J),SIAS(I),SIA(I),L(I),DSI(I),DSIA(I),
     1RDSI(I),SIT(I),SIB(I),SIES(I),SIE(I)
 4120 CONTINUE
 4121 CONTINUE
      WRITE(JTAPE,9057) CHIAS,CHIA,CDHIP,CDHIA,CRDHI
      IF(NC .EQ. 0) GO TO 4125
      WRITE(JTAPE,9058) CSIAS,CSIA,CDSIP,CDSIA, CRDSI
 4125 WRITE(JTAPE,9059) IHH
      IF(NC .EQ. 0) GO TO 4130
      WRITE(JTAPE,9060) (ISH(I), I=1,NC)
      WRITE(JTAPE,9061)
      DO 4127 I=1,NC
 4127 WRITE(JTAPE,9062) (ISS(I,J),J=1,NC)
 4130 CONTINUE
      WRITE(JTAPE,9063)DPSQ,SFC,EHI,CEHI
C
C     CHOOSE NEW DPSQ AND CHECK FOR STOPS
C
 5000 KS=1
      KACC=1
C     KA=1 VARIABLE DPSQ, KA=2 FIXED DPSQ
      GO TO(5010,5230),KA
C     VARIABLE DPSQ
 5010 IF(F1 .GT. 0.0) GO TO 5020
C     USE PSI PERFORMANCE INDICATOR
      NDEL=0
      IF(CRDSI .GE. A21) GO TO 5200
      IF(CRDSI .GE. A22) GO TO 5210
      GO TO 5300
C     USE PHI PERFORMANCE INDICATOR
 5020 IF( CRDHI .LT. A33) GO TO 5300
 5030 IF( ABS(CEHI) .GE. CEHIM) NDEL=-1
      NDEL=NDEL + 1
      IF(CRDHI .LT. A32) GO TO 5210
      IF(CRDHI .GE. A31) GO TO 5300
C     INCREASE DPSQ
 5200 DPSQ=A11*DPSQ
C     SAME DPSQ
C     ACCEPTABLE ITERATION - CHECK FOR STOPS
 5210 IF( NDEL .LE. NDELF ) GO TO 5230
      KS=3
      GO TO 5400
 5220 IF(DPSQ .GT. DPSQM) GO TO 5230
      KS=4
      GO TO 5400
 5230 IF(NIT .LE. ITN) KS=2
      GO TO 5400
C     DECREASE DPSQ
 5300 DPSQ=A12*DPSQ
```

```
C     IS TRAJECTORY ACCEPTABLE
      IF(F1 .GT. 0.0) GO TO 5310
      IF(CRDSI .GT. A23) GO TO 5210
      GO TO 5320
 5310 IF(CRDHI .GT. A34) GO TO 5030
C     UNACCEPTABLE TRAJECTORY
 5320 KACC=2
      IF(DPSQ .LT. .01*DPSQM) KS=5
      IF(ITTN .GE. 5) KS=6
 5400 WRITE(JTAPE,9064)DPSQ,KS,KACC,KA
      GO TO(6000,5403,5403,5403,5408,5408),KS
C     KS=1      DO NOT STOP
C     KS=2      REQUESTED NUMBER OF ITERATIONS COMPLETE
C     KS=3      COMPOSITE GRADIENT LESS THAN MINIMUM VALUE FOR NDELF
C               ITERATIONS
C     KS=4      CURRENT ITERATION ACCEPTABLE BUT NEW DPSQ LESS THAN DPSQM
C     KS=5      CURRENT ITERATION UNACCEPTABLE AND DPSQ LESS THAN
C               .01*DPSQM
C     KS=6      TRIAL ITERATION COUNT .GE. 5
C     LAST ITERATION ACCEPTABLE
 5403 WRITE(JTAPE,9065) CASENO,ID
      WRITE(JTAPE,9015) DELTA
      WRITE(JTAPE,9013)
      WRITE(JTAPE,9011) (ATABX(I),I=1,NIF)
      WRITE(JTAPE,9014)
      WRITE(JTAPE,9011) (STABX(I),I=1,NIF)
      IF(PRTOPT(2) .EQ. 0.0) GO TO 1000
      PUNCH 9010,CASENO,DELTA
      PUNCH 9013
      PUNCH 9012,(ATABX(I),I=1,NIF)
      PUNCH 9014
      PUNCH 9012,(STABX(I),I=1,NIF)
      GO TO 1000
C     LAST ITERATION UNACCEPTABLE RECOVER CONTROL TABLES
 5408 DO 5412 I=1,NIF
      ATABX(I)=ATABS(I)*RADIAN
 5412 STABX(I)=STABS(I)*RADIAN
      GO TO 5403
C     KACC=1 LAST ITERATION ACCEPTABLE,KACC=2 NOT ACCEPTABLE
 6000 CONTINUE
      GO TO(6001,7030),KACC
C     ITERATION ACCEPTABLE
 6001 IF(PRTOPT(1) .EQ. 0.0) GO TO 6004
C     PRINT CONTROL VARIABLE TABLES
      WRITE(JTAPE,9065) CASENO,ID
      WRITE(JTAPE,9015) DELTA
      WRITE(JTAPE,9013)
      WRITE(JTAPE,9011) (ATABX(I),I=1,NIF)
      WRITE(JTAPE,9014)
      WRITE(JTAPE,9011) (STABX(I),I=1,NIF)
 6004 CONTINUE
      IF(PRTOPT(2) .LT. 2.0) GO TO 6006
      PUNCH 9010,CASENO,DELTA
      PUNCH 9013
      PUNCH 9012,(ATABX(I),I=1,NIF)
      PUNCH 9014
      PUNCH 9012,(STABX(I),I=1,NIF)
```

```
 6006 CONTINUE
C     SUBROUTINE ADJNT COMPUTES ADJOINT TRAJECTORY
 6010 CALL ADJNT(NF)
C     NF=1 ADJOINT OK  2=  ADJOINT ERROR
      GO TO(7020,1000),NF
C
C
C     THIS SECTION COMPUTES NEW CONTROL VARIABLES AND RETURNS TO COMPUTE
C     A NEW FORWARD TRAJECTORY
C     ITERATION ACCEPTABLE
 7020 ITN=ITN+1
      ITTN=0
      CASENO(2)=AINT(CASENO(2) + 1.0)
      HIS=HI
      IF(NC .NE. 0) GO TO 7024
 7021 F1=DPSQ
      F2=SQRT(DPSQ/IHH)
      SFC=0.0
      GO TO 7329
 7024 DO 7025 I=1,NC
      L(I)=0
      SIAS(I)=SIA(I)
      SIES(I)=SIE(I)
      IF( SIA(I) .GE. (SIT(I) + EPS(I))) GO TO 7023
      IF( SIA(I) .LE. (SIB(I) - EPS(I))) GO TO 7023
      GO TO 7025
 7023 L(I)=1
 7025 LS(I)=L(I)
      GO TO 7090
C     ITERATION UNACCEPTABLE
 7030 ITTN=ITTN+1
      CASENO(2)=CASENO(2) + .1
      IF(NC .EQ. 0) GO TO 7021
      DO 7031 I=1,NC
 7031 L(I)=LS(I)
 7090 DO 7091 I=1,NC
 7091 SID(I)=-SIES(I)
 7092 CALL MINVL(ISS,ISSIL,L)
      TEMP=TMAML(SID,ISSIL,SID,L)
      F1=DPSQ-TEMP
      SFC=1.0
      IF(F1 .GE. 0.0) GO TO 7100
      SFC=SQRT(DPSQ/TEMP)
      F1=0.0
 7100 DO 7101 I=1,NC
 7101 DSI(I)=SFC*SID(I)
      F2=SQRT(F1/(IHH-TMAML(ISH,ISSIL,ISH,L)))
C     ARE THERE ANY INACTIVE CONSTRAINTS
      DO 7145 I=1,NC
      IF (L(I) .EQ. 0)GO TO 7150
 7145 CONTINUE
C     NO-ALL ACTIVE
      GO TO 7320
C     ONE OR MORE CONSTRAINTS INACTIVE
 7150 KCI=1
      DO 7240 I=1,NC
      IF(L(I) .NE. 0)GO TO 7240
```

```
      DSIJ=AM*F2*(ISH(I)-TMAML(ISS(1,I),ISSIL,ISH,L))
     1+TMAML(ISS(1,I),ISSIL,DSI,L)
      IF((SIAS(I)+DSIJ) .LT. (SIT(I)+EPS(I)))GO TO 7200
      SID(I)=SIT(I)-SIAS(I)
      L(I)=1
      KCI=2
      GO TO 7240
 7200 IF(SIAS(I) + DSIJ .GE. (SIB(I)-EPS(I))) GO TO 7230
      SID(I)=SIB(I)-SIAS(I)
      L(I)=1
      KCI=2
      GO TO 7240
 7230 DSI(I)=DSIJ
 7240 CONTINUE
C     KCI=1 NO CONSTRAINTS BECAME APPLICABLE KCI=2 YES
      GO TO (7300,7092), KCI
 7300 CONTINUE
 7320 DO 7321 I=1,NC
 7321 DBETA(I)=DSI(I)
 7329 DO 7330 I=1,NIF
      TIME=FLOAT(I-1)*DELTA
      CALL MULG(50,NERR,TIME,WTAB,WT(1,1),WTAB(1,2),WT(2,2),WTAB(1,3))
      ATABX(I)=AM*F2/WT(1,1)*(GL(1,1,I)-TMAML(GL(2,1,I),ISSIL,ISH,L))
     1+TMAML(GL(2,1,I),ISSIL,DBETA,L)/WT(1,1) + ATABS(I)
      ATABX(I)=ATABX(I)*RADIAN
      STABX(I)=AM*F2/WT(2,2)*(GL(1,2,I)-TMAML(GL(2,2,I),ISSIL,ISH,L))
     1+TMAML(GL(2,2,I),ISSIL,DBETA,L)/WT(2,2) + STABS(I)
      STABX(I)=STABX(I)*RADIAN
 7330 CONTINUE
C     ADDITIONAL OUTPUT AT END OF ADJOINT TRAJECTORY
      CALL OUT(IGO(256),HFMT(181),DFMT(181))
C     RETURN TO COMPUTE NEXT FORWARD TRAJECTORY WITH NEW CONTROL TABLES
      GO TO 2000
C     FORMAT STATEMENTS
 9010 FORMAT(10H CASENO(1)  F10.0,F10.1,5X, 8HDELTA(1)  F10.6  )
 9011 FORMAT( 11F12.5)
 9012 FORMAT( 6F12.5)
 9013 FORMAT( 10H ALPHAX(1) )
 9014 FORMAT( 10H SIGMAX(1) )
 9015 FORMAT( 9HODELTA(1)  F10.6  )
 9050 FORMAT( 107H1 RAYTHEON THREE-DIMENSIONAL TRAJECTORY OPTIMIZATION P
     1ROGRAM  TOS9   ANALYSIS                    CASENO   ITNO  /
     292X,F8.0,F7.1,2X,3A6)
 9051 FORMAT( 1HO  26X,104HOLD            NEW   STATUS     PREDICT      AC
     1TUAL   RATIO   HI LIM     LOW LIM     OLD ERROR   NEW ERROR   )
 9052 FORMAT( 8HO PAYOFF  2X, A3, 1X, A6, E26.6)
 9053 FORMAT( 13HO CONSTRAINTS  )
 9054 FORMAT( 14X,A6, E26.6, I3, E47.4, E12.4)
 9055 FORMAT( 8HO PAYOFF  2X,A3,1X,A6,2E13.6,6X,2E12.5,F8.2)
 9056 FORMAT( 14X, A6, 2E13.6, I3, 3X, 2E12.4, F8.2, 4E12.4 )
 9057 FORMAT( 11HO PHI INDEX   9X, 2E13.6, 6X, 2E12.4, F8.2)
 9058 FORMAT( 11HO PSI INDEX   9X,2E13.6,6X, 2E12.4, F8.2)
 9059 FORMAT( 12HO I PHI PHI   E12.5)
 9060 FORMAT( 12HO I PSI PHI   8E12.5)
 9061 FORMAT( 12HO I PSI PSI   )
 9062 FORMAT( E24.5,7E12.5)
 9063 FORMAT( 14HO CURRENT DPSQ  E16.5 / 5H  SFC  E25.5 / 13H  PAYOFF GR
```

```
    1AD  E17.5 / 16H  PHI INDEX GRAD  E14.5  )
9064 FORMAT(10H  NEW DPSQ  E20.5  // 4H  KS   I6/ 6H  KACC  I4 /
    14H  KA   I6 )
9065 FORMAT( 107H1 RAYTHEON THREE-DIMENSIONAL TRAJECTORY OPTIMIZATION P
    1ROGRAM  TOS9   CONTROL VARIABLES        CASENO   ITNO  /
    292X,F8.0,F7.1,2X,3A6)
     END
```

```
$IBFTC PIN
      SUBROUTINE PRNTIN
C
C     C O M M O N   F O R   A L L   P R O G R A M S
C
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      COMMON / VAR / RADIAN,D10,D11,D12,D13,D14,D20,D30,D31,D32,
     1ALPHD,SIGMD,ALPHR,SIGMR,CSIGM,SSIGM,CTHET,STHET,R,H,VHSQ,
     2VTSQ,VH,VT,PEA,RHO,SOFS,MACH,MASS,TH,THR,COSIT,SINIT,COSDT,
     3SINDT,THRX,THRY,THRZ,CALPH,SALPH,C1,C2,C3,C4,C5,C6,THRR,THRT,
     4THRP,S,X1,X2,X3,X4,X5,X6,X7,X8,X9,RSQ,R4TH,SQRHO,ACELG,TAU,
     5TIMEH,CL,CD,TFINAL,B56,B58,PEH,RHOH,SOFSH,PEL,RHOL,SOFSL,
     6DPERH,DRRH,DSRH,CLHM,CDHM,CLLM,CDLM,PCLRM,PCDRM,DRPRT,B10,
     7B20,B21,B30,B31,B40,B41,B50,B51,B52,B53,B54,B55,B57,B60,B61,
     8B62,B70,B71,B81,B82,B90,B91,B92,B93,B100,B101,B102,B103,B104,
     9R3RD,R5TH,VT3,B200,B201,VH3,B202,B204,PTRRU,PTTRU,PTPRU
      COMMON / VAR / PTRRV,PTTRV,PTPRV,PTRRW,PTTRW,PTPRW,PTHRH,
     1PTXRH,PTYRH,PTZRH,B300,B301,B302,PTRRR,PTTRR,PTPRR,B404,
     2B405,B400,SCLCD,DTRA,B401,B402,B403,B406,B407,PTRRA,PTTRA,
     3PTPRA,PTRRS,PTTRS,PTPRS,E1,E2,E3,E4,E5,E6,CLAL,CDAL,CLAH,
     4CDAH,PCLRA,PCDRA,B130,B131,PTRG,STRG,DTRG,GTRG,SRAT,
     5MUD,NUD,PHISD,THETSD,BETAD,GAMMAD,THETD,PHID,DYNA,ENER,AST,RG
      COMMON / VAR / SFRM(1500),GL(9,2,201),ATABS(201),STABS(201),
     1PARTS(14),PARTC(14),CADJ(6,9),F(6,6),CF(6,6),G(6,2),CG(6,2)
      COMMON / VAR / LAMP(9),VARX(25)
      REAL MACH,MASS,LAMP
      COMMON / ANAL / HI,HIS,SIA(8),SIAS(8),SIE(8),SIES(8),CSIA,
     1CSIAS,CDSIP,CDSIA,CRDSI,CHIA,CHIAS,CDHIA,CDHIP,CRDHI,DSIA(8),
     2RDSI(8),DHIA,DHIP,RDHI,DP,EHI,CEHI,INTGL(9,9),IHH,ISH(8),
     3ISS(8,8),WT(2,2),F1,F2,SID(8),SFC,DSI(8),DSIJ,DBETA(8),
     4ISSIL(8,8),DPSQ,DPSQK,IHHK,ISHK(8),ISSK(8,8),
     5ISSILK(8,8),DBETAK(8),STVRS(12,9),LAMBDA(6,8,9),SFCK,F1K,
     6F2K
      REAL INTGL,IHH,ISH,ISS,ISSIL,INTGLK,IHHK,ISHK,ISSK,ISSILK,
     1LAMBDA
      COMMON / IVAR / JTAPE,KTAPE,LTAPE,ITN,ITTN,ISTAGE,NC,
     1NT,NRTN,NTRG,KACC,KA,KS,NIF,NG,L(8),LS(8)
      EQUIVALENCE(SFRM(2000),NORA(2000),HNA(1997),NEQA(1996),
     1TIMEA(1995),U(1993),V(1992),W(1991),RX(1990),THETR(1989),
     2PHIR(1988),PA(1987),PB(1986),PC(1985),PD(1984),PE(1983),
     3PF(1982),U1(1981),V1(1980),W1(1979),RX1(1978),THETR1(1977),
     4PHIR1(1976),PA1(1975),PB1(1974),PC1(1973),PD1(1972),PE1(1971),
     5PF1(1970))
      DIMENSION ADJ(6,9), DADJ(6,9)
      EQUIVALENCE(SFRM(2000),NORB(1981),HNB(1978),NEQB(1977),
     1TIMEB(1976),ADJ(1974),DADJ(1920))
```

```
      DIMENSION STVR(12)
      EQUIVALENCE(STVR(1),U)
C
C     E N D   O F   C O M M O N   F O R   A L L   P R O G R A M S
C
C     SET UP SYMBOL TABLE FOR INPUT
      CALL BCDCON( 12HCASENO(2)    ,CASENO)
      CALL BCDCON( 12HID(3)        ,ID     )
      CALL BCDCON( 12HVEL          ,XVT    )
      CALL BCDCON( 12HGAMMA        ,XGAMMD)
      CALL BCDCON( 12HBETA         ,XBETAD)
      CALL BCDCON( 12HH            ,XH     )
      CALL BCDCON( 12HTHETA        ,XTHETD)
      CALL BCDCON( 12HPHI          ,XPHID )
      CALL BCDCON( 12HGS           ,GS     )
      CALL BCDCON( 12HGMASS        ,GMASS )
      CALL BCDCON( 12HPAYOFF       ,PAYOFF)
      CALL BCDCON( 12HSTOP         ,STOP   )
      CALL BCDCON( 12HSTOPV        ,STOPV )
      CALL BCDCON( 12HPRTINT       ,HP     )
      CALL BCDCON( 12HNIT          ,AIT    )
      CALL BCDCON( 12HDPSQ         ,XDPSQ )
      CALL BCDCON( 12HDPINC        ,A11    )
      CALL BCDCON( 12HDPDEC        ,A12    )
      CALL BCDCON( 12HPSIEG        ,A21    )
      CALL BCDCON( 12HPSIGP        ,A22    )
      CALL BCDCON( 12HPSIRJ        ,A23    )
      CALL BCDCON( 12HPHIGE        ,A31    )
      CALL BCDCON( 12HPHIEG        ,A32    )
      CALL BCDCON( 12HPHIGP        ,A33    )
      CALL BCDCON( 12HPHIRJ        ,A34    )
      CALL BCDCON( 12HNDELF        ,DELF   )
      CALL BCDCON( 12HCDELM        ,CEHIM )
      CALL BCDCON( 12HDPSQM        ,DPSQM )
      CALL BCDCON( 12HKQC          ,K1     )
      CALL BCDCON( 12HRQC          ,R1     )
      CALL BCDCON( 12HNQC          ,N1     )
      CALL BCDCON( 12HKQR          ,K2     )
      CALL BCDCON( 12HRQR          ,R2     )
      CALL BCDCON( 12HNQR          ,N2     )
      CALL BCDCON( 12HAH1          ,AH1    )
      CALL BCDCON( 12HAH2          ,AH2    )
      CALL BCDCON( 12HAQD1         ,AHR1   )
      CALL BCDCON( 12HAQD2         ,AHR2   )
      CALL BCDCON( 12HLAM          ,LAM    )
      CALL BCDCON( 12HSLPE         ,SLPE   )
      CALL BCDCON( 12HMU           ,MU     )
      CALL BCDCON( 12HRF           ,RF     )
      CALL BCDCON( 12HJ            ,GJ     )
      CALL BCDCON( 12HOMEGA        ,OMEGA )
      CALL BCDCON( 12HRE           ,RE     )
      CALL BCDCON( 12HRPC          ,RPC    )
      CALL BCDCON( 12HTLIM         ,TLIM   )
      CALL BCDCON( 12HHLIM         ,HLIM   )
      CALL BCDCON( 12HORDA         ,ORDA   )
      CALL BCDCON( 12HHNOMA        ,HNOMA )
      CALL BCDCON( 12HEUA          ,EUA    )
```

```
      CALL BCDCON( 12HELA          ,ELA    )
      CALL BCDCON( 12HHMAXA        ,HMAXA  )
      CALL BCDCON( 12HHMINA        ,HMINA  )
      CALL BCDCON( 12HORDB         ,ORDB   )
      CALL BCDCON( 12HHNOMB        ,HNOMB  )
      CALL BCDCON( 12HEUB          ,EUB    )
      CALL BCDCON( 12HELB          ,ELB    )
      CALL BCDCON( 12HHMAXB        ,HMAXB  )
      CALL BCDCON( 12HHMINB        ,HMINB  )
      CALL BCDCON( 12HYCB          ,YCB    )
      CALL BCDCON( 12HYCA          ,YCA    )
      CALL BCDCON( 18HALPHA(100)          ,ANTABX)
      CALL BCDCON( 18HSIGMA(100)          ,SNTABX)
      CALL BCDCON( 18HCON(8)              ,CON    )
      CALL BCDCON( 18HSIT(8)              ,SIT    )
      CALL BCDCON( 18HSIB(8)              ,SIB    )
      CALL BCDCON( 18HEPS(8)              ,EPS    )
      CALL BCDCON( 18HEOST(4)             ,EOST   )
      CALL BCDCON( 18HAE(3)               ,AE     )
      CALL BCDCON( 18HS(3)                ,REFA   )
      CALL BCDCON( 18HIT(3)               ,IT     )
      CALL BCDCON( 18HDT(3)               ,DT     )
      CALL BCDCON( 18HTMT(150)            ,TMTAB  )
      CALL BCDCON( 18HTAU(100)            ,TAUTAB)
      CALL BCDCON( 18HWT(50,3)            ,WTAB   )
      CALL BCDCON( 18HMCLCD(11,4)         ,MTAB   )
      CALL BCDCON( 18HACLCD(26,4)         ,ATAB   )
      CALL BCDCON( 18HCL(150,4)           ,CLTAB  )
      CALL BCDCON( 18HCD(150,4)           ,CDTAB  )
      CALL BCDCON( 18HDELTA(1)            ,DELTA  )
      CALL BCDCON( 18HALPHAX(201)         ,ATABX  )
      CALL BCDCON( 18HSIGMAX(201)         ,STABX  )
      CALL BCDCON( 18HLAMT(9)             ,LAMTAB)
      CALL BCDCON( 18HPRTOPT(4)           ,PRTOPT)
      CALL BCDCON( 18HHFMT(140)           ,HFMT   )
      CALL BCDCON( 18HDFMT(140)           ,DFMT   )
      CALL BCDCON( 18HIGO(170)            ,IGO    )
      CALL BCDCON( 18HINPUT(10)           ,INPUTX)
 1000 DELTA=0.0
      CASENO(2)=0.0
C     READ INPUT
      CALL SYMBLS(INCOL1)
C     PROGRAM EXITS IF A 2 OR GREATER APPEARS IN COLUMN 1 OF THE INPUT
      IF(INCOL1 .GT. 1) CALL EXIT
C     PRINT INPUT
      WRITE(JTAPE,9100) CASENO,ID
      WRITE(JTAPE,9101) CASENO
      WRITE(JTAPE,9102) XVT,XGAMMD,XBETAD,XH,XTHETD,XPHID
      WRITE(JTAPE,9103) GS,GMASS,PAYOFF,STOP,STOPV,HP
      WRITE(JTAPE,9104) AIT,XDPSQ,A11,A12,A21,A22
      WRITE(JTAPE,9105) A23,A31,A32,A33,A34,DELF
      WRITE(JTAPE,9106) CEHIM,DPSQM,K1,R1,N1,K2
      WRITE(JTAPE,9107) R2,N2,AH1,AH2,AHR1,AHR2
      WRITE(JTAPE,9108) SLPE,MU,RF,GJ,OMEGA,RE
      WRITE(JTAPE,9109) RPC,TLIM,HLIM,ORDA,HNOMA,EUA
      WRITE(JTAPE,9110) ELA,HMAXA,HMINA,YCA,ORDB,HNOMB
      WRITE(JTAPE,9111) EUB,ELB,HMAXB,HMINB,YCB,LAM
```

```
      DO 20 I=1,9
      KT=I
      IF(LAMTAB(I) .EQ. 0.0) GO TO 30
   20 CONTINUE
   30 WRITE(JTAPE,9112) (LAMTAB(I),I=1,KT)
      IF( DELTA .NE. 0.0) GO TO 80
C     NOMINAL CONTROL TABLES
      DO 40 I=3,50
      NTAB =I
      IF(ANTABX(I) .LT. ANTABX(I-1)) GO TO 50
   40 CONTINUE
      WRITE(JTAPE,9113) (ANTABX(I),I=1,50)
      WRITE(JTAPE,9114) (ANTABX(I),I=51,100)
      GO TO 51
   50 WRITE(JTAPE,9113) (ANTABX(I),I=1,NTAB)
      NTAB=NTAB+49
      WRITE(JTAPE,9114) (ANTABX(I),I=51,NTAB)
   51 CONTINUE
      DO 60 I=3,50
      NTAB=I
      IF(SNTABX(I) .LT. SNTABX(I-1)) GO TO 70
   60 CONTINUE
      WRITE(JTAPE,9115) (SNTABX(I),I=1,50)
      WRITE(JTAPE,9116) (SNTABX(I),I=51,100)
      GO TO 71
   70 WRITE(JTAPE,9115) (SNTABX(I),I=1,NTAB)
      NTAB=NTAB+49
      WRITE(JTAPE,9116) (SNTABX(I),I=51,NTAB)
   71 CONTINUE
   80 CONTINUE
      DO 90 I=1,8
      NC=I
      IF(CON(I) .EQ. 0.0) GO TO 100
   90 CONTINUE
      WRITE(JTAPE,9117) (CON(I),I=1,8)
      GO TO 109
  100 WRITE(JTAPE,9117) (CON(I),I=1,NC)
      NC=NC-1
      IF(NC .EQ. 0) GO TO 110
  109 WRITE(JTAPE,9118) (SIT(I),I=1,NC)
      WRITE(JTAPE,9119) (SIB(I),I=1,NC)
      WRITE(JTAPE,9120) (EPS(I),I=1,NC)
  110 CONTINUE
      DO 120 I=1,3
      ISTAGE=I
      IF(EOST(I) .EQ. 0.0) GO TO 130
  120 CONTINUE
      WRITE(JTAPE,9121) (EOST(I),I=1,3)
      GO TO 131
  130 WRITE(JTAPE,9121) (EOST(I),I=1,ISTAGE)
      ISTAGE=ISTAGE-1
      IF(ISTAGE .EQ. 0) GO TO 160
  131 WRITE(JTAPE,9122) (AE(I),I=1,ISTAGE)
      WRITE(JTAPE,9123) (REFA(I),I=1,ISTAGE)
      WRITE(JTAPE,9124) (IT(I),I=1,ISTAGE)
      WRITE(JTAPE,9125)  (DT(I),I=1,ISTAGE)
      DO 140 I=3,50
```

```
      NTAB=I
      IF(TMTAB(I) .LT. TMTAB(I-1)) GO TO 150
  140 CONTINUE
      WRITE(JTAPE,9126) (TMTAB(I),I=1,50)
      WRITE(JTAPE,9127) (TMTAB(I),I=51,100)
      WRITE(JTAPE,9130) (TMTAB(I),I=101,150)
      GO TO 160
  150 WRITE(JTAPE,9126) (TMTAB(I),I=1,NTAB)
      NTAB=NTAB+49
      WRITE(JTAPE,9127) (TMTAB(I),I=51,NTAB)
      NTAB=NTAB+50
      WRITE(JTAPE,9128) (TMTAB(I),I=101,NTAB)
  160 CONTINUE
      ISTAGE=ISTAGE+1
      DO 210 N=1,ISTAGE
      J=(N - 1)*11 + 1
      DO 170 I=3,11
      NTAB1=I
      IF(MTAB(I,N) .LT. MTAB(I-1,N)) GO TO 180
  170 CONTINUE
  180 WRITE(JTAPE,9129) J,(MTAB(I,N),I=1,NTAB1)
      J=(N - 1)*26 + 1
      DO 190 I=3,26
      NTAB2=I
      IF(ATAB(I,N) .LT. ATAB(I-1,N)) GO TO 200
  190 CONTINUE
  200 WRITE(JTAPE,9130) J,(ATAB(I,N),I=1,NTAB2)
      NTAB=(NTAB1-1)*(NTAB2-1)
      J=(N-1)*150 + 1
      WRITE(JTAPE,9131) J,(CLTAB(I,N),I=1,NTAB)
  210 WRITE(JTAPE,9132) J,(CDTAB(I,N),I=1,NTAB)
      DO 220 I=3,50
      NTAB=I
      IF(WTAB(I,1) .LT. WTAB(I-1,1)) GO TO 230
  220 CONTINUE
      WRITE(JTAPE,9133) (WTAB(I,1),I=1,50)
      GO TO 231
  230 WRITE(JTAPE,9133) (WTAB(I,1),I=1,NTAB)
      NTAB=NTAB-1
  231 WRITE(JTAPE,9134) (WTAB(I,2),I=1,NTAB)
      WRITE(JTAPE,9135) (WTAB(I,3), I=1,NTAB)
      DO 240 I=3,50
      NTAB=I
      IF( TAUTAB(I) .LT. TAUTAB(I-1) .OR. TAUTAB(I) .EQ. 0.0) GO TO 250
  240 CONTINUE
      WRITE(JTAPE,9136) (TAUTAB(I),I=1,50)
      WRITE(JTAPE,9137) (TAUTAB(I),I=51,100)
      GO TO 251
  250 WRITE(JTAPE,9136) (TAUTAB(I),I=1,NTAB)
      NTAB=NTAB+49
      WRITE(JTAPE,9137) (TAUTAB(I),I=51,NTAB)
  251 CONTINUE
      WRITE(JTAPE,9138) (PRTOPT(I),I=1,4)
      IF(DELTA .EQ. 0.0) GO TO 300
      WRITE(JTAPE,9139) DELTA
      WRITE(JTAPE,9140) (ATABX(I),I=1,NIF)
      WRITE(JTAPE,9141) (STABX(I),I=1,NIF)
```

```
 300 CONTINUE
     RETURN
9100 FORMAT( 107H1 RAYTHEON THREE-DIMENSIONAL TRAJECTORY OPTIMIZATION P
    1ROGRAM  TOS9   INPUT                          CASENO   ITNO  /
    292X,F8.0,F7.1,2X,3A6)
9101 FORMAT( 11H0 CASENO(1)   F9.0,F21.1 )
9102 FORMAT( 8H0 VEL      F12.1, 3X, 6HGAMMA   F12.3, 3X, 6HBETA
    1F12.3, 3X, 6HH       F12.0, 3X, 6HTHETA  F12.3, 3X, 6HPHI
    2F12.3  )
9103 FORMAT(  8H0 GS      F12.2, 3X, 6HGMASS  F12.2, 3X, 6HPAYOFF
    1F12.0,  3X, 6HSTOP    F12.0, 3X, 6HSTOPV   E12.5, 3X, 6HPRTINT
    2F12.2  )
9104 FORMAT( 8H0 NIT      F12.0, 3X, 6HDPSQ   E12.5, 3X, 6HDPINC  F12.2,
    13X, 6HDPDEC  F12.2, 3X, 6HPSIEG   F12.2, 3X, 6HPSIGP F12.2 )
9105 FORMAT( 8H0 PSIRJ    F12.2,3X,6HPHIGE  F12.2,3X,6HPHIEG   F12.2,
    13X, 6HPHIGP  F12.2, 3X, 6HPHIRJ F12.2, 3X, 6HNDELF  F12.2)
9106 FORMAT( 8H0 CDELM    E12.5,3X,6HDPSQM  E12.5,3X,6HKQC    E12.5 ,
    13X, 6HRQC   E12.5, 3X, 6HNQC   E12.5, 3X, 6HKQR   E12.5)
9107 FORMAT( 8H0 RQR      E12.5,3X,6HNQR    E12.5,3X,6HAH1    E12.5,
    13X, 6HAH2   E12.5,  3X,  6HAQD1   E12.5,  3X, 6HAQD2   E12.5)
9108 FORMAT(  8H0 SLPE    F12.1, 3X, 6HMU     E12.5, 3X, 6HRF
    1 E12.5, 3X, 6HJ      E12.5, 3X, 6HOMEGA   E12.5, 3X, 6HRE
    2  E12.5  )
9109 FORMAT(  8H0 RPC     E12.5, 3X, 6HTLIM    F12.0, 3X, 6HHLIM
    1 F12.0, 3X, 6HORDA  F12.0, 3X, 6HHNOMA  F12.4, 3X, 6HEUA
    2  E12.5  )
9110 FORMAT(  8H0 ELA     E12.4, 3X, 6HHMAXA   F12.2, 3X, 6HHMINA
    1F12.4, 3X, 6HYCA   F12.5, 3X, 6HORDB   F12.0, 3X, 6HHNOMB
    2 F12.4  )
9111 FORMAT( 8H0 EUB      E12.5, 3X, 6HELB     E12.5, 3X, 6HHMAXB
    1 F12.2,  3X, 6HHMINB    F12.4,  3X,  6HYCB     F12.5,  3X, 6HLAM
    2     F12.1)
9112 FORMAT( 12H0 LAMT(1)       9F12.2)
9113 FORMAT( 12H0 ALPHA(1)      10F12.2  / ( 11F12.2))
9114 FORMAT( 12H  ALPHA(51)      10F12.2  / (11F12.2))
9115 FORMAT( 12H0 SIGMA(1)      10F12.2  / (11F12.2))
9116 FORMAT( 12H  SIGMA(51)       10F12.2  / (11F12.2))
9117 FORMAT( 12H0 CON(1)        8F12.0  )
9118 FORMAT( 12H  SIT(1)        8E12.5)
9119 FORMAT( 12H  SIB(1)        8E12.5  )
9120 FORMAT( 12H  EPS(1)        8E12.5 )
9121 FORMAT( 12H0 EOST(1)        3F12.2  )
9122 FORMAT( 12H  AE(1)        3F12.3  )
9123 FORMAT( 12H  S(1)          3F12.3)
9124 FORMAT( 12H  IT(1)         3F12.2)
9125 FORMAT( 12H  DT(1)        3F12.2  )
9126 FORMAT( 12H0 TMT(1)         10F12.2  / (11F12.2))
9127 FORMAT( 12H  TMT(51)        10F12.1 /(11F12.1))
9128 FORMAT( 12H  TMT(101)      10F12.2  / (11F12.2))
9129 FORMAT( 8H0 MCLCD( I2, 2H)  10F12.2  / (11F12.2))
9130 FORMAT( 8H  ACLCD( I2 , 2H)   10F12.2  /(11F12.2))
9131 FORMAT( 5H  CL(  I3,  4H)       10F12.4  / (11F12.4))
9132 FORMAT(  5H  CD(  I3,  4H)         10F12.4  / (11F12.4))
9133 FORMAT( 12H0 WT(1)         10F12.2  / (11F12.2))
9134 FORMAT( 12H  WT(51)        10F12.2  / (11F12.2))
9135 FORMAT( 12H  WT(101)        10F12.2  / (11F12.2))
9136 FORMAT( 12H0 TAU(1)      10F12.2  / (11F12.2))
```

```
9137 FORMAT( 12H  TAU(51)        10F12.2 /(11F12.2))
9138 FORMAT( 12H0 PRTOPT(1)     4F12.0)
9139 FORMAT( 12H0 DELTA(1)    F12.6)
9140 FORMAT( 12H  ALPHAX(1)     10F12.5/(11F12.5))
9141 FORMAT( 12H  SIGMAX(1)     10F12.5 /( 11F12.5))
     END
```

```
$IBFTC STE
      SUBROUTINE STATE(NF)
C PURPOSE       FORWARD TRAJECTORY
C     THIS SUBROUTINE COMPUTES THE FORWARD TRAJECTORY USING THE CONTROL
C     VARIABLES TABLES INPUTED OR COMPUTED BY THE PROGRAM
C     THE OUTPUT IS A TAPE OF ALL STATE VARIABLES AT EACH INTEGRATION
C     POINT
C
C     C O M M O N   F O R   A L L   P R O G R A M S
C
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      COMMON / VAR / RADIAN,D10,D11,D12,D13,D14,D20,D30,D31,D32,
     1ALPHD,SIGMD,ALPHR,SIGMR,CSIGM,SSIGM,CTHET,STHET,R,H,VHSQ,
     2VTSQ,VH,VT,PEA,RHO,SOFS,MACH,MASS,TH,THR,COSIT,SINIT,COSDT,
     3SINDT,THRX,THRY,THRZ,CALPH,SALPH,C1,C2,C3,C4,C5,C6,THRR,THRT,
     4THRP,S,X1,X2,X3,X4,X5,X6,X7,X8,X9,RSQ,R4TH,SQRHO,ACELG,TAU,
     5TIMEH,CL,CD,TFINAL,B56,B58,PEH,RHOH,SOFSH,PEL,RHOL,SOFSL,
     6DPERH,DRRH,DSRH,CLHM,CDHM,CLLM,CDLM,PCLRM,PCDRM,DRPRT,B10,
     7B20,B21,B30,B31,B40,B41,B50,B51,B52,B53,B54,B55,B57,B60,B61,
     8B62,B70,B71,B81,B82,B90,B91,B92,B93,B100,B101,B102,B103,B104,
     9R3RD,R5TH,VT3,B200,B201,VH3,B202,B204,PTRRU,PTTRU,PTPRU
      COMMON / VAR / PTRRV,PTTRV,PTPRV,PTRRW,PTTRW,PTPRW,PTHRH,
     1PTXRH,PTYRH,PTZRH,B300,B301,B302,PTRRR,PTTRR,PTPRR,B404,
     2B405,B400,SCLCD,DTRA,B401,B402,B403,B406,B407,PTRRA,PTTRA,
     3PTPRA,PTRRS,PTTRS,PTPRS,E1,E2,E3,E4,E5,E6,CLAL,CDAL,CLAH,
     4CDAH,PCLRA,PCDRA,B130,B131,PTRG,STRG,DTRG,GTRG,SRAT,
     5MUD,WUD,PHISD,THETSD,BETAD,GAMMAD,THETD,PHID,DYNA,ENER,AST,RG
      COMMON / VAR / SFRM(1500),GL(9,2,201),ATABS(201),STABS(201),
     1PARTS(14),PARTC(14),CADJ(6,9),F(6,6),CF(6,6),G(6,2),CG(6,2)
      COMMON / VAR / LAMP(9),VARX(25)
      REAL MACH,MASS,LAMP
      COMMON / ANAL / HI,HIS,SIA(8),SIAS(8),SIE(8),SIES(8),CSIA,
     1CSIAS,CDSIP,CDSIA,CRDSI,CHIA,CHIAS,CDHIA,CDHIP,CRDHI,DSIA(8),
     2RDSI(8),DHIA,DHIP,RDHI,DP,EHI,CEHI,INTGL(9,9),IHH,ISH(8),
     3ISS(8,8),WT(2,2),F1,F2,SID(8),SFC,DSI(8),DSIJ,DBETA(8),
     4ISSIL(8,8),DPSQ,DPSQK,IHHK,ISHK(8),ISSK(8,8),
     5ISSILK(8,8),DBETAK(8),STVRS(12,9),LAMBDA(6,8,9),SFCK,F1K,
     6F2K
      REAL INTGL,IHH,ISH,ISS,ISSIL,INTGLK,IHHK,ISHK,ISSK,ISSILK,
     1LAMBDA
      COMMON / IVAR / JTAPE,KTAPE,LTAPE,ITN,ITTN,ISTAGE,NC,
     1NT,NRTN,NTRG,KACC,KA,KS,NIF,NG,L(8),LS(8)
      EQUIVALENCE(SFRM(2000),NORA(2000),HNA(1997),NEQA(1996),
     1TIMEA(1995),U(1993),V(1992),W(1991),RX(1990),THETR(1989),
     2PHIR(1988),PA(1987),PB(1986),PC(1985),PD(1984),PE(1983),
     3PF(1982),U1(1981),V1(1980),W1(1979),RX1(1978),THETR1(1977),
```

```
     4PHIR1(1976),PA1(1975),PB1(1974),PC1(1973),PD1(1972),PE1(1971),
     5PF1(1970))
      DIMENSION ADJ(6,9), DADJ(6,9)
      EQUIVALENCE(SFRM(2000),NORB(1981),HNB(1978),NEQB(1977),
     1TIMEB(1976),ADJ(1974),DADJ(1920))
      DIMENSION STVR(12)
      EQUIVALENCE(STVR(1),U)
C
C     E N D  O F  C O M M O N  F O R  A L L  P R O G R A M S
C
      COMMON / ISTATE / UN,VN,WN,RXN,THETN,PHIN,VHN,CTHTN,STHTN
      REAL MUR,MUD,NUR,NUD
      REAL LIFT
      EQUIVALENCE(VARX(25),LIFT(25),DRAG(24),DA(23),DS(22))
 2000 REWIND LTAPE
C     NF=1 NORMAL RETURN INDICATOR
      NF=1
C     COMPUTE INITIAL VALUES OF DIFFERENTIAL EQUATIONS OF MOTION
C     FORWARD TRAJECTORY TITLE
      CALL OUT(IGO(1),HFMT(1),DFMT(1))
      TIMEA=0.0
      U=XVT*SIN(XGAMMD/RADIAN)
      V=XVT*COS(XGAMMD/RADIAN)*COS(XBETAD/RADIAN)
      W=XVT*COS(XGAMMD/RADIAN)*SIN(XBETAD/RADIAN)
      THETR=XTHETD/RADIAN
      RX = XH - (RE - RPC)*COS(THETR)**2
      PHIR=XPHID/RADIAN
      PA=0.
      PB=0.
      PC=0.
      PD=0.
      PE=0.
      PF=0.
C     SAVE INITIAL STATE VARIABLES
      UN=U
      VN=V
      WN=W
      RXN=RX
      THETN=THETR
      PHIN=PHIR
      VHN=SQRT(VN**2 + WN**2)
      CTHTN=COS(THETN)
      STHTN=SIN(THETN)
      NH=1
      PTRG=0.0
      STRG=0.0
      DTRG=0.0
      NORA=ORDA
      HNA=HNOMA
      NEQA=12
      ISTAGE=1
      CALL SMARK(4,12,NORA,NRTN,NTRG,EUA,ELA,HMAXA,HMINA,YCA,
     1TIMEA,PTRG,
     2TIMEA,DTRG,
     3TIMEA,STRG,
     4SRAT,0.0,
     5H,AH2,
```

```
     6TIMEA,TLIM,
     7H,HLIM )
C     TRIGGER 1      OUTPUT
C     TRIGGER 2      LAMCOS-DO
C     TRIGGER 3      STAGING
C     TRIGGER 4      STOPPING
C     TRIGGER 5      ALTITUDE PENALTY
C     TRIGGER 6      TIME LIMIT
C     TRIGGER 7      HEIGHT LIMIT
      GO TO(2100,2200,2300,2400,8001),NRTN
C     NTRN=1         END OF STEP ROUTINE
C     NTRN=2         DERIVATIVE 1 ROUTINE
C     NTRN=3         DERIVATIVE 2 ROUTINE
C     NTRN=4         TRIGGER ROUTINES
C     NTRN=5         ERROR MARK
C
C     END OF STEP ROUTINE FORWARD
 2100 CALL SFRWD(1)
      SRAT=FUNCT(STOP) - STOPV
      CALL TRA14
C
C     FORWARD DERIVATIVE ROUTINE 1
 2200 IF(DELTA.NE.0.0)GO TO 2202
      CALL MULG(50,NERR,TIMEA,ANTABX,ALPHD,ANTABX(51))
      CALL MULG(50,NERR,TIMEA,SNTABX,SIGMD,SNTABX(51))
      GO TO 2206
 2202 I=TIMEA/DELTA+1.0
      IF(I.GE.NIF)GO TO 2204
      D=AMOD(TIMEA,DELTA)/DELTA
      ALPHD=(ATABX(I+1)-ATABX(I))*D+ATABX(I)
      SIGMD=(STABX(I+1)-STABX(I))*D+STABX(I)
      GO TO 2206
 2204 ALPHD=ATABX(NIF)
      SIGMD=STABX(NIF)
 2206 ALPHR=ALPHD/RADIAN
      SIGMR=SIGMD/RADIAN
      CSIGM=COS(SIGMR)
      SSIGM=SIN(SIGMR)
C     FORWARD DERIVATIVE ROUTINE 2
 2300 CTHET=COS(THETR)
      STHET=SIN(THETR)
      R=RX + RE
      H=R-RE+(RE-RPC)*CTHET**2
      VHSQ = V**2+W**2
      VTSQ = VHSQ + U**2
      VH = SQRT(VHSQ)
      VT = SQRT(VTSQ)
C     ATMOSPHERIC DATA
      CALL ATM(H,PEA)
      MACH = VT/SOFS
C     LIFT AND DRAG COEFFICIENTS
      CALL CLCD(ISTAGE,ALPHR,MACH,CL,CD)
C     CHECK FOR THRUST
      IF(EOST(ISTAGE) .EQ. 0.0) GO TO 2350
      CALL MULG(50,NERR,TIMEA,TMTAB(1),TH,TMTAB(51),MASS,TMTAB(101))
      THR=TH + (SLPE - PEA)*AE(ISTAGE)
      COSIT = COS(IT(ISTAGE)/RADIAN)
```

```
      SINIT = SIN(IT(ISTAGE)/RADIAN)
      COSDT=COS(DT(ISTAGE)/RADIAN)
      SINDT = SIN(DT(ISTAGE)/RADIAN)
      THRX=THR*COSIT*COSDT
      THRY=THR*COSIT*SINDT
      THRZ=THR*SINIT
      CALPH = COS(ALPHR)
      SALPH = SIN(ALPHR)
      C1=CALPH/VT
      C2=SALPH/VT
      C3=CSIGM/VT
      C4=SSIGM/VT
      C5=W*SSIGM/VH+V*U*C3/VH
      C6=V*SSIGM/VH-W*U*C3/VH
      THRR = (U*C1 + VH*C2*CSIGM)*THRX - VH*C4*THRY +
     1(VH*C1*CSIGM - U*C2)*THRZ
      THRT=(V*C1-SALPH*C5)*THRX+(V*U*C4/VH-W*CSIGM/VH)*THRY
     1-(V*C2+CALPH*C5)*THRZ
      THRP = (W*C1 + SALPH*C6)*THRX + (V*CSIGM/VH + W*U*C4/VH)*THRY
     1+(CALPH*C6-W*C2)*THRZ
      S = REFA(ISTAGE)
      GO TO 2352
C     GLIDE STAGE
 2350 THRR=0.0
      THRT=0.0
      THRP=0.0
      S = GS
      MASS = GMASS
 2352 X1=.5*RHO*S*VT/MASS
      X2=CD*X1
      X3=CL*X1
      X4=CSIGM*X3
      X5=U*X4/VH
      X6=SSIGM*X3*VT/VH
      X7=-U/R
      X8=CTHET/STHET*W/R
      X9=STHET*R*D30+D31*W
      RSQ=R*R
      R4TH=RSQ**2
C     DERIVATIVES FOR EQUATIONS OF MOTION
      U1=VHSQ/R+X9*STHET-U*X2+VH*X4-MU/RSQ
     1-D10*(1.0-3.0*CTHET**2)/R4TH+THRR/MASS
      V1=V*X7+W*X8+CTHET*X9-V*X2-V*X5-W*X6
     1+D14*CTHET*STHET/R4TH+THRT/MASS
      W1=W*X7-V*X8-D31*(U*STHET+V*CTHET)
     1-W*X2-W*X5+V*X6+THRP/MASS
      RX1=U
      THETR1=V/R
      PHIR1=W/(R*STHET)
C     DERIVATIVES FOR PENALTY FUNCTIONS
      SQRHO=SQRT(RHO)
      PA1=K1*R1*SQRHO*(VT/1000.)**N1
     1+K2*R2*RHO*SQRHO*(VT/10000.)**N2
      ACELG=RHO*VTSQ*S/64.4*SQRT(CL**2+CD**2)/MASS
      CALL MULG(50,NERR,ACELG,TAUTAB(1),TAU,TAUTAB(51))
      PB1=1.0/TAU
      PC1=0.0
```

```
C     HAS H EQUALED AH2  NH=1 NO, NH=2 YES
      GO TO (2360,2362),NH
 2362 IF(H.LE.AH2)GO TO 2360
      PC1=AH1*(H/AH2 - 1.0)**2
 2360 CONTINUE
      PD1=0.0
      IF ( PA1 .LE. AHR2 ) GO TO 2370
      PD1=AHR1*(PA1/AHR2 - 1.0)**2
 2370 CONTINUE
      PE1=0.0
      PF1=0.0
C     RETURN TO MARK
      CALL TRA14
C     TRIGGER RETURN
 2400 GO TO(2401,2451,2501,2401,2651,2701,2751),NTRG
C     OUTPUT TRIGGER
 2401 PTRG=PTRG+HP
      MUD=0.0
      NUD=0.0
      PHISD=0.0
      THETSD=0.0
      IF(TIMEA .EQ. 0.0) GO TO 2403
      SPMPN=SIN(PHIR-PHIN)
      CPMPN=COS(PHIR-PHIN)
      CNSM=(STHET*(VN*CTHTN*CPMPN + WN*SPMPN) -
     1VN*STHTN*CTHET)/VHN
      CNCM=CTHTN*CTHET + STHTN*STHET*CPMPN
      MUR=ATAN2(CNSM,CNCM)
      SINMU=SIN(MUR)
      SINNU=(WN*STHTN*CTHET - STHET*(WN*CTHTN*
     1CPMPN - VN*SPMPN))/VHN
      NUR=ATAN2(SINNU*SINMU,CNSM)
      TANNU=SINNU*SINMU/CNSM
      PHIS=ATAN2(TANNU,SINMU)
      THETS=ATAN2(SQRT(SINNU**2 + COS(NUR)**2*SINMU**2),COS(NUR)
     1*COS(MUR))
      MUD=MUR*RADIAN
      NUD NUR*RADIAN
      PHISD=PHIS*RADIAN
      THETSD=THETS*RADIAN
 2403 BETAD=ATAN2(W,V)*RADIAN
      GAMMAD=ATAN2(U,VH)*RADIAN
      THETD=THETR*RADIAN
      PHID=PHIR*RADIAN
      DYNA=.5*RHO*VT**2
      ENER=MASS*(VT**2/2.0 - MU/R - (1.0 - 3.0*CTHET**2)/3.0
     1*(RF/R)**3*MU*GJ/RF)
      LIFT=.5*RHO*VTSQ*S*CL
      DRAG=.5*RHO*VTSQ*S*CD
      DA=0.0
      DS=0.0
      IF(DELTA .EQ. 0.0) GO TO 2404
      I=TIMEA/DELTA + 1.0
      IF(I .GE. NIF) GO TO 2404
      D=AMOD(TIMEA,DELTA)/DELTA
      DA=((ATABS(I+1) - ATABS(I))*D + ATABS(I))*RADIAN - ALPHD
      DS=((STABS(I+1) - STABS(I))*D + STABS(I))*RADIAN - SIGMD
```

```
 2404 CONTINUE
      CALL EMARK(AST,RG)
      CALL OUT(IGO(86),HFMT(41),DFMT(41))
C     NTRG=4 STOPPING CONDITION REACHED
      IF(NTRG .EQ. 4) GO TO 2601
      CALL TRA14
C
C     LAMCOS-DO TRIGGER
C
 2451 IF(LAMP(1) .NE. 0.0 .AND. ITN .NE. 0 .AND. NC .NE. 0) GO TO 2453
 2452 CALL OFF(2)
      CALL TRA14
 2453 IF(TIMEA .NE. 0.0) GO TO 2459
      KT=1
 2454 DTRG=TFINAL*LAMP(KT)
      CALL TRA14
C     COMPUTE INTEGRALS FROM TIMEA TO TFINAL
 2459 DO 2460 I=1,9
      DO 2460 J=1,9
 2460 INTGL(I,J)=0.0
      DPSQK=0.0
      KN=TIMEA/DELTA+1.1
      DO 2462 I=KN,NIF
      TIME=FLOAT(I-1)*DELTA
      CALL MULG(50,NERR,TIME,WTAB,WT(1,1),WTAB(1,2),WT(2,2),WTAB(1,3))
      SC=1.0
      IF(I.EQ.KN.OR.I.EQ.NIF)GO TO 2461
      SC=4.0
      IF(MOD(I,2).NE.0)SC=2.0
 2461 CONTINUE
      DO 2463 J=1,NT
      DO 2463 K=1,J
      DO 2463 M=1,2
 2463 INTGL(J,K)=SC*GL(J,M,I)/WT(M,M)*GL(K,M,I) + INTGL(J,K)
 2462 DPSQK=SC*((ATABX(I)/RADIAN-ATABS(I))**2/WT(1,1)
     1+(STABX(I)/RADIAN - STABS(I))**2/WT(2,2)) + DPSQK
      DO 2464 J=1,NT
      DO 2464 K=1,J
      INTGL(J,K)=INTGL(J,K)*DELTA/3.0
 2464 INTGL(K,J)=INTGL(J,K)
      DPSQK=DPSQK*DELTA/3.0
      IHHK=INTGL(1,1)
      DO 2466 I=1,8
 2466 ISHK(I)=INTGL(I+1,1)
      DO 2468 I=1,8
      DO 2468 J=1,8
 2468 ISSK(I,J)=INTGL(I+1,J+1)
      CALL MINVL(ISSK,ISSILK,L)
      DO 2472 I=1,8
 2472 DBETAK(I)=0.0
      IF(NC.EQ.0)GO TO 2474
      DO 2473 I=1,NC
      DO 2473 J=1,6
 2473 DBETAK(I)=DBETAK(I) - LAMBDA(J,I,KT)*(STVR(J) -STVRS(J,KT))
      DO 2471 I=1,NC
      DO 2471 J=1,6
 2471 DBETAK(I)=DBETAK(I) - CADJ(J,I+1)*(STVR(J+6) - STVRS(J,KT))
```

```
      DO 2475  I=1,NC
 2475 DBETAK(I)=DSI(I) -DBETAK(I)
 2474 CONTINUE
      TEMP=TMAML(DBETAK,ISSILK,DBETAK,L)
      F1K=DPSQK-TEMP
      SFCK=1.0
      IF(F1K.GT.0.0)GO TO 2478
      F1K=0.0
      SFCK=SQRT(DPSQK/TEMP)
 2478 CONTINUE
      DO 2476 I=1,8
 2476 DBETAK(I)=SFCK*DBETAK(I)
      F2K=SQRT(F1K/(IHHK-TMAML(ISHK,ISSILK,ISHK,L)))
      KN=KN+1
      DO 2480 I=KN,NIF
      TIME=FLOAT(I-1)*DELTA
      CALL MULG(50,NERR,TIME,WTAB,WT(1,1),WTAB(1,2),WT(2,2),WTAB(1,3))
      ATABX(I)=AM*F2K/WT(1,1)*(GL(1,1,I)-TMAML(GL(2,1,I),ISSILK,ISHK,L))
     1+TMAML(GL(2,1,I),ISSILK,DBETAK,L)/WT(1,1)+ATABS(I)
      STABX(I)=AM*F2K/WT(2,2)*(GL(1,2,I)-TMAML(GL(2,2,I),ISSILK,ISHK,L))
     1+TMAML(GL(2,2,I),ISSILK,DBETAK,L)/WT(2,2) + STABS(I)
      ATABX(I)=ATABX(I)*RADIAN
 2480 STABX(I)=STABX(I)*RADIAN
      KT=KT+1
      IF(KT .EQ. 10) GO TO 2452
      IF(LAMP(KT) .EQ. 0.0) GO TO 2452
      GO TO 2454
C     BOOST TRIGGER
 2501 IF(TIMEA .NE. 0.0) GO TO 2502
      IF( EOST(ISTAGE) .NE. 0.0) GO TO 2503
      CALL OFF(3)
      CALL TRA14
 2503 STRG=EOST(ISTAGE)
      CALL TRA14
 2502 ISTAGE=ISTAGE+1
      IF ( EOST(ISTAGE) .NE. 0.0) GO TO 2504
      CALL OFF(3)
      CALL TRA24
 2504 STRG=EOST(ISTAGE)
      CALL TRA24
C
C     STOPPING TRIGGER
 2601 CALL SFRWD(2)
C     ADDITIONAL OUTPUT AT END OF FORWARD TRAJECTORY
      CALL OUT(IGO(171),HFMT(141),DFMT(141))
      GO TO(2602,2603),NH
 2602 TIMEH=TIMEA
 2603 RETURN
C     PENALTY FUNCTON C TRIGGER
 2651 NH=2
      TIMEH=TIMEA
      CALL OFF(5)
      CALL TRA14
C     TIME LIMIT TRIGGER
 2701 WRITE(JTAPE,9005)
 9005 FORMAT(16H       TIME LIMIT   )
C     NF=2 ERROR RETURN INDICATOR
```

```
      NF=2
      RETURN
2751  WRITE(JTAPE,9006)
9006  FORMAT (18H       HEIGHT LIMIT   )
      NF=2
      RETURN
8001  WRITE(JTAPE,9010)
9010  FORMAT( 11H ERROR MARK )
      NF=2
      RETURN
      END
```

```
$IBFTC ADT
      SUBROUTINE ADJNT(NF)
C PURPOSE      ADJOINT TRAJECTORY
C     THIS SUBROUTINE COMPUTES THE ADJOINT TRAJECTORY OF THE FORWARD
C     TRAJECTORY PREVIOUSLY RUN.
C     THE INPUT TO THIS SUBROUTINE IS THE TAPE CONSTRUCTED DURING THE
C     FORWARD TRAJECTORY
C     THE OUTPUT FROM THIS SUBROUTINE IS A TABLE OF INFLUENCE FUNCTIONS
C     GL(I,J,K) WHERE I IS THE INDEX FOR THE PAYOFF FUNCTION AND
C     CONSTRAINTS, J IS THE INDEX FOR THE 2 CONTROL VARIABLES, ALPHA AND
C     SIGMA, AND K IS THE INDEX FOR THE 200 EQUALLY SPACED POINTS ALONG
C     THE TRAJECTORY
C     THIS SUBROUTINE ALSO COMPUTES THE INTEGRALS OF THE INFLUENCE
C     FUNCTIONS
C
C     C O M M O N   F O R   A L L   P R O G R A M S
C
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      COMMON / VAR / RADIAN,D10,D11,D12,D13,D14,D20,D30,D31,D32,
     1ALPHD,SIGMD,ALPHR,SIGMR,CSIGM,SSIGM,CTHET,STHET,R,H,VHSQ,
     2VTSQ,VH,VT,PEA,RHO,SOFS,MACH,MASS,TH,THR,COSIT,SINIT,COSDT,
     3SINDT,THRX,THRY,THRZ,CALPH,SALPH,C1,C2,C3,C4,C5,C6,THRR,THRT,
     4THRP,S,X1,X2,X3,X4,X5,X6,X7,X8,X9,RSQ,R4TH,SQRHO,ACELG,TAU,
     5TIMEH,CL,CD,TFINAL,B56,B58,PEH,RHOH,SOFSH,PEL,RHOL,SOFSL,
     6DPERH,DRRH,DSRH,CLHM,CDHM,CLLM,CDLM,PCLRM,PCDRM,DRPRT,B10,
     7B20,B21,B30,B31,B40,B41,B50,B51,B52,B53,B54,B55,B57,B60,B61,
     8B62,B70,B71,B81,B82,B90,B91,B92,B93,B100,B101,B102,B103,B104,
     9R3RD,R5TH,VT3,B200,B201,VH3,B202,B204,PTRRU,PTTRU,PTPRU
      COMMON / VAR / PTRRV,PTTRV,PTPRV,PTRRW,PTTRW,PTPRW,PTHRH,
     1PTXRH,PTYRH,PTZRH,B300,B301,B302,PTRRR,PTTRR,PTPRR,B404,
     2B405,B400,SCLCD,DTRA,B401,B402,B403,B406,B407,PTRRA,PTTRA,
     3PTPRA,PTRRS,PTTRS,PTPRS,E1,E2,E3,E4,E5,E6,CLAL,CDAL,CLAH,
     4CDAH,PCLRA,PCDRA,B130,B131,PTRG,STRG,DTRG,GTRG,SRAT,
     5MUD,NUD,PHISD,THETSD,BETAD,GAMMAD,THETD,PHID,DYNA,ENER,AST,RG
      COMMON / VAR / SFRM(1500),GL(9,2,201),ATABS(201),STABS(201),
     1PARTS(14),PARTC(14),CADJ(6,9),F(6,6),CF(6,6),G(6,2),CG(6,2)
      COMMON / VAR / LAMP(9),VARX(25)
      REAL MACH,MASS,LAMP
      COMMON / ANAL / HI,HIS,SIA(8),SIAS(8),SIE(8),SIES(8),CSIA,
     1CSIAS,CDSIP,CDSIA,CRDSI,CHIA,CHIAS,CDHIA,CDHIP,CRDHI,DSIA(8),
     2RDSI(8),DHIA,DHIP,RDHI,DP,EHI,CEHI,INTGL(9,9),IHH,ISH(8),
     3ISS(8,8),WT(2,2),F1,F2,SID(8),SFC,DSI(8),DSIJ,DBETA(8),
     4ISSIL(8,8),DPSQ,DPSQK,IHHK,ISHK(8),ISSK(8,8),
     5ISSILK(8,8),DBETAK(8),STVRS(12,9),LAMBDA(6,8,9),SFCK,F1K,
     6F2K
      REAL INTGL,IHH,ISH,ISS,ISSIL,INTGLK,IHHK,ISHK,ISSK,ISSILK,
```

```
     1LAMBDA
      COMMON / IVAR / JTAPE,KTAPE,LTAPE,ITN,ITTN,ISTAGE,NC,
     1NT,NRTN,NTRG,KACC,KA,KS,NIF,NG,L(8),LS(8)
      EQUIVALENCE(SFRM(2000),NORA(2000),HNA(1997),NEQA(1996),
     1TIMEA(1995),U(1993),V(1992),W(1991),RX(1990),THETR(1989),
     2PHIR(1988),PA(1987),PB(1986),PC(1985),PD(1984),PE(1983),
     3PF(1982),U1(1981),V1(1980),W1(1979),RX1(1978),THETR1(1977),
     4PHIR1(1976),PA1(1975),PB1(1974),PC1(1973),PD1(1972),PE1(1971),
     5PF1(1970))
      DIMENSION ADJ(6,9), DADJ(6,9)
      EQUIVALENCE(SFRM(2000),NORB(1981),HNB(1978),NEQB(1977),
     1TIMEB(1976),ADJ(1974),DADJ(1920))
      DIMENSION STVR(12)
      EQUIVALENCE(STVR(1),U)
C
C     E N D  O F  C O M M O N  F O R  A L L  P R O G R A M S
C
      COMMON / FORW / STRAJ(20,13)
      EQUIVALENCE (F(36),PFURU(36),PFVRU(35),PFWRU(34),PFRRU(33),
     1PFTRU(32),PFPRU(31),PFURV(30),PFVRV(29),PFWRV(28),PFRRV(27),
     2PFTRV(26),PFPRV(25),PFURW(24),PFVRW(23),PFWRW(22),PFRRW(21),
     3PFTRW(20),PFPRW(19),PFURR(18),PFVRR(17),PFWRR(16),PFRRR(15),
     4PFTRR(14),PFPRR(13),PFURT(12),PFVRT(11),PFWRT(10),PFRRT(9),
     5PFTRT(8),PFPRT(7),PFURP(6),PFVRP(5),PFWRP(4),PFRRP(3),
     6PFTRP(2),PFPRP(1))
      EQUIVALENCE (CF(36),PFARU(36),PFBRU(35),PFCRU(34),PFDRU(33),
     1PFERU(32),PFFRU(31),PFARV(30),PFBRV(29),PFCRV(28),PFDRV(27),
     2PFERV(26),PFFRV(25),PFARW(24),PFBRW(23),PFCRW(22),PFDRW(21),
     3PFERW(20),PFFRW(19),PFARR(18),PFBRR(17),PFCRR(16),PFDRR(15),
     4PFERR(14),PFFRR(13),PFART(12),PFBRT(11),PFCRT(10),PFDRT(9),
     5PFERT(8),PFFRT(7),PFARP(6),PFBRP(5),PFCRP(4),PFDRP(3),PFERP(2),
     6PFFRP(1))
      EQUIVALENCE (G(12),PFURA(12),PFVRA(11),PFWRA(10),PFRRA(9),
     1PFTRA(8),PFPRA(7),PFURS(6),PFVRS(5),PFWRS(4),PFRRS(3),PFTRS(2),
     2PFPRS(1))
      EQUIVALENCE (CG(12),PFARA(12),PFBRA(11),PFCRA(10),PFDRA(9),
     1PFERA(8),PFFRA(7),PFARS(6),PFBRS(5),PFCRS(4),PFDRS(3),PFERS(2),
     2PFFRS(1))
      REAL LOW
C
C
C
 6000 TFINAL=TIMEA
      DELTB=TFINAL/FLOAT(NIF-1)
      NF=1
      TIMEB =0.0
C     INITIAL VALUES OF ADJOINT EQUATIONS
      DO 6061 I=1,6
      DO 6061 J=1,9
      ADJ(I,J)=0.0
 6061 CADJ(I,J)=0.0
C     SUBROUTINE PARTAL COMPUTES THE PARTIAL DERIVATIVES OF FUNCTIONS
C     WITH RESPECT TO THE STATE VARIABLES AND TIME
      CALL PARTAL(STOP,PARTS)
      CALL PARTAL(PAYOFF,PARTC)
      DO 6050 I=1,6
C     THE ADJOINT DIFFERENTIAL EQUATIONS ARE DIVIDED IN TO TWO GROUPS.
```

```
C     ADJ - NON ZERO DERIVATIVES
C     CADJ - ZERO DERIVATIVES
      ADJ(I,1)=PARTC(I)-PARTC(14)/PARTS(14)*PARTS(I)
      CADJ(I,1) = PARTC(I+6) - PARTC(14)/PARTS(14)*PARTS(I+6)
 6050 CONTINUE
      IF(NC.EQ.0)GO TO 6060
      DO 6056 J=1,NC
      CALL PARTAL( CON(J),PARTC)
      DO 6056 I=1,6
      ADJ(I,J+1)=PARTC(I)-PARTC(14)/PARTS(14)*PARTS(I)
 6056 CADJ(I,J+1)=PARTC(I+6)-PARTC(14)/PARTS(14)*PARTS(I+6)
 6060 CONTINUE
C     SET UP FOR SUBROUTINE SMARK
      NORB=ORDB
      HNB=HNOMB
      NEQB=54
      TIMEB=0.0
      NEQBA=NT*6
      GTRG=0.0
      STRG=0.0
      DTRG=0.0
      CALL SMARK(4,NEQBA,NORB,NRTN,NTRG,EUB,ELB,HMAXB,HMINB,YCB,
     1TIMEB,GTRG,
     2TIMEB,STRG,
     3TIMEB,DTRG)
C     TRIGGER 1      INFLUENCE FUNCTION
C     TRIGGER 2      STAGING
C     TRIGGER 3      LAMCOS-DO
      GO TO(6500,6600,6700,6800,8001),NRTN
C     NTRN=1         END OF STEP ROUTINE
C     NTRN=2         DERIVATIVE 1 ROUTINE
C     NTRN=3         DERIVATIVE 2 ROUTINE
C     NTRN=4         TRIGGER ROUTINES
C     NTRN=5         ERROR MARK
C     END OF STEP ROUTINE
 6500 CALL EMARK(AST,RG)
      CALL TRA14
C
C     ADJOINT DERIVATIVE ROUTINE 1
 6600 TIMEA=TFINAL-TIMEB
      IF(TIMEB .NE. 0.0) GO TO 6604
      GO TO 6603
C     FIRST TO LAST
 6601 DO 6602 I=1,13
 6602 STRAJ(20,I)=STRAJ(1,I)
 6603 BACKSPACE LTAPE
C     READ LTAPE TO GET TABLES FROM FORWARD TRAJECTORY
      READ(LTAPE)N,((STRAJ(I,J),I=1,19),J=1,13)
      BACKSPACE LTAPE
      IF(N .LT. 2) GO TO 6601
 6604 IF(TIMEA .LT. 0.0) TIMEA=0.0
C     INTERPOLATE TO GET STATE VARIABLES
      CALL MULG(20,NERR,TIMEA,STRAJ(1,1),U,STRAJ(1,2),V,STRAJ(1,3),
     1W,STRAJ(1,4),RX,STRAJ(1,5),THETR,STRAJ(1,6),PHIR,STRAJ(1,7),
     2PA,STRAJ(1,8),PB,STRAJ(1,9),PC,STRAJ(1,10),PD,STRAJ(1,11),PE,
     3STRAJ(1,12),PF,STRAJ(1,13))
C     NERR=1 TIMEA WITHIN TABLE, NERR=2 OFF LOW END, NERR=3 OFF HIGH END
```

```
      GO TO (6605,6601,6605), NERR
 6605 CONTINUE
C     GET ALPHA AND SIGMA FROM INPUT TABLES
      IF(DELTA.NE.0.0)GO TO 6020
      CALL MULG(50,NERR,TIMEA,ANTABX,ALPHD,ANTABX(51))
      CALL MULG(50,NERR,TIMEA,SNTABX,SIGMD,SNTABX(51))
      GO TO 6040
C     GET ALPHA AND SIGMA FROM COMPUTED TABLES
 6020 I=TIMEA/DELTA+1.0
      IF( I .GE. NIF ) GO TO 6030
      D=AMOD(TIMEA,DELTA)/DELTA
      ALPHD=(ATABX(I+1)-ATABX(I))*D+ATABX(I)
      SIGMD=(STABX(I+1)-STABX(I))*D+STABX(I)
      GO TO 6040
 6030 ALPHD=ATABX(NIF)
      SIGMD=STABX(NIF)
 6040 ALPHR=ALPHD/RADIAN
      SIGMR=SIGMD/RADIAN
      R=RX + RE
      H=R-RE+(RE-RPC)*CTHET**2
      B56 = V**2 + W**2
      B58 = B56 + U**2
      VH = SQRT(B56)
      VT = SQRT(B58)
C     ATM ATMOSPHERE SUBROUTINE
      CALL ATM(H,PEA)
      CALL ATM(H+500.,PEH)
      CALL ATM(H-500.,PEL)
C     PARTIAL OF PRESSURE,DENSITY,AND SPEED OF SOUND RESPECT TO HEIGHT
      DPERH=(PEH-PEL)/1000.
      DRRH=(RHOH-RHOL)/1000.
      DSRH=(SOFSH-SOFSL)/1000.
      MACH=VT/SOFS
C     CALCULATE CL AND CD
      CALL CLCD(ISTAGE,ALPHR,MACH,CL,CD)
C     PARTIALS OF CL AND CD RESPECT TO MACH
      CALL CLCD(ISTAGE,ALPHR,MACH+.05,CLHM,CDHM)
      CALL CLCD(ISTAGE,ALPHR,MACH-.05,CLLM,CDLM)
      PCLRM=(CLHM-CLLM)/.1
      PCDRM=(CDHM-CDLM)/.1
      CTHET = COS(THETR)
      STHET = SIN(THETR)
C     DERIVATIVE OF RP RESPECT TO THETR
      DRPRT=2.0*(RE-RPC)*CTHET*STHET
      CSIGM = COS(SIGMR)
      SSIGM = SIN(SIGMR)
C     CHECK FOR THRUST
      IF( EOST(ISTAGE) .EQ. 0.0 ) GO TO 6650
      S=REFA(ISTAGE)
C     IF THRUST GET THRUST AND MASS
      CALL MULG(50,NERR,TIMEA,TMTAB(1),TH,TMTAB(51),MASS,TMTAB(101))
      GO TO 6651
 6650 S=GS
      MASS=GMASS
 6651 CONTINUE
C     COMMON TERMS
      B10=RHO/2.0*S/MASS
```

```
      B20=CD*VT
      B21=CL*VT
      B30=(CD+MACH*PCDRM)/VT
      B31=(CL+MACH*PCLRM)/VT
      B40=CSIGM/VH
      B41=SSIGM/VH
      B50=U**2
      B51=V**2
      B52=W**2
      B53=U*V
      B54=U*W
      B55=V*W
      B57=B53*W
      B60=U/R
      B61=V/R
      B62=W/R
      B70=2.0*CD+MACH*PCDRM
      B71=2.0*CL+MACH*PCLRM
      B81=D31*CTHET
      B82=D31*STHET
      B90=(CTHET/STHET)/R
      B91=CTHET*STHET
      B92=CTHET**2
      B93=STHET**2
      B100=RHO*MACH/SOFS*DSRH
      B101=CD*DRRH-B100*PCDRM
      B102=CL*DRRH-B100*PCLRM
      B103=S/(2.0*MASS)
      B104=D30*STHET
      RSQ = R*R
      R3RD = RSQ*R
      R4TH = R*R3RD
      R5TH = R*R4TH
C
      IF(EOST(ISTAGE).EQ.0.0)GO TO 6101
      THR=TH + (SLPE - PEA)*AE(ISTAGE)
      COSIT=COS(IT(ISTAGE)/RADIAN)
      SINIT=SIN(IT(ISTAGE)/RADIAN)
      COSDT=COS(DT(ISTAGE)/RADIAN)
      SINDT=SIN(DT(ISTAGE)/RADIAN)
      THRX=THR*COSIT*COSDT
      THRY=THR*COSIT*SINDT
      THRZ=THR*SINDT
      CALPH=COS(ALPHR)
      SALPH=SIN(ALPHR)
      VT3=B58*VT
      B200=(THRX*CALPH-THRZ*SALPH)/VT3
      B201=(THRX*CSIGM*SALPH-THRY*SSIGM+THRZ*CSIGM*CALPH)/VT3
      VH3=B56*VH
      B202=(THRX*SSIGM*SALPH+THRY*CSIGM+THRZ*SSIGM*CALPH)/VH3
      B204=B56+B58
      PTRRU=B56*B200-U*VH*B201
      PTTRU=-B53*B200-V*VH*B201
      PTPRU=-B54*B200-W*VH*B201
      PTRRV=-B53*B200+B53*U/VH*B201
      PTTRV=(B50+B52)*B200+B55*B202-U*(B56*B58-B51*B204)/VH3*B201
      PTPRV=-B55*B200+B52*B202+B57*B204/VH3*B201
```

```
      PTRRW=-B54*B200+B50*W/VH*B201
      PTTRW=-B55*B200-B51*B202+B57*B204/VH3*B201
      PTPRW = (B50+B51)*B200-B55*B202-U*(B56*B58-B52*B204)/VH3*B201
      PTHRH=-AE(ISTAGE)*DPERH
      PTXRH=PTHRH*COSIT*COSDT
      PTYRH=PTHRH*COSIT*SINDT
      PTZRH=PTHRH*SINDT
      B300=(PTXRH*CALPH-PTZRH*SALPH)/VT
      B301=(PTXRH*CSIGM*SALPH-PTYRH*SSIGM+PTZRH*CSIGM*CALPH)/VT
      B302=(PTXRH*SSIGM*SALPH+PTYRH*CSIGM+PTZRH*SSIGM*CALPH)/VH
      PTRRR=U*B300+VH*B301
      PTTRR = V*B300 - W*B302 - B53/VH*B301
      PTPRR = W*B300 + V*B302 - B54/VH*B301
 6101 CONTINUE
C     COMPUTE MATRIX F - SEE EQUIVALENCES
      PFURU=B10*(CSIGM*VH*U*B31-B20-(B50*B30))
      PFVRU=-B61-B10*(B53*B30+B40*V*(B21+B50*B31)+
     1B41*B54*B71)
      PFWRU=-B62-B82-B10*(B54*B30+B40*W*(B21+
     1B50*B31)-B41*B53*B71)
      PFRRU=1.0
      PFTRU = 0.0
      PFPRU = 0.0
      PFURV=2.0*B61+B10*(B40*V*(B21+B56*B31)-B53*B30)
      PFVRV=-B60-B10*(B20+B51*B30+B40*U*(B52/B56*B21+
     1B51*B31)+B41*B55*(B71-B21/B56*VT))
      PFWRV=-W*B90-B81-B10*(B55*B30+B40*B57*(B31-
     1B21/B56)-B41*(B52/B56*B21*VT+B51*B71))
      PFRRV = 0.0
      PFTRV=1.0/R
      PFPRV = 0.0
      PFURW=2.0*B62+B82+B10*(B40*W*(B21+B56*B31)-B54*B30)
      PFVRW=2.0*W*B90+B81-B10*(B55*B30+B40*B57*(B31-B21/B56)
     1+B41*(B51/B56*B21*VT+B52*B71))
      PFWRW=-B60-V*B90-B10*(B20+B52*B30+B40*U*(B51*B21/B56+
     1B52*B31)-B41*B55*(B71-B21/B56*VT))
      PFRRW = 0.0
      PFTRW = 0.0
      PFPRW=1.0/(R*STHET)
      PFURH=B103*(-U*VT*B101+CSIGM*VH*VT*B102)
      PFVRH=-B103*(V*VT*B101+B102*(B40*B53*VT+B41*W*B58))
      PFWRH=-B103*(W*VT*B101+B102*(B40*B54*VT-B41*V*B58))
      PFURR=-B56/RSQ + B104*STHET + PFURH +D20/R3RD
     1+D11*(1.0 - 3.0*B92)/R5TH
      PFVRR=B53/RSQ - W*B62*B90 + B104*CTHET + PFVRH - D12/R5TH*B91
      PFWRR=B60*B62 + B61*W*B90 + PFWRH
      PFRRR = 0.0
      PFTRR=-V/RSQ
      PFPRR=-W/(RSQ*STHET)
      PFURT=D32*R*B91+W*B81-D13/R4TH*B91-PFURH*DRPRT
      PFVRT=-(B52/R)/B93+R*D30*(B92-B93)-B82*W+D14/R4TH*(B92-B93)
     1-PFVRH*DRPRT
      PFWRT=B61*W/B93 - B81*U + B82*V - PFWRH*DRPRT
      PFRRT = 0.0
      PFTRT = 0.0
      PFPRT=-W*B90/STHET
      PFURP = 0.0
```

```
      PFVRP = 0.0
      PFWRP = 0.0
      PFRRP = 0.0
      PFTRP = 0.0
      PFPRP = 0.0
C     COMPUTE CF MATRIX
      SQRHO=SQRT(RHO)
      B404=(VT/1000.)**N1
      B405=(VT/10000.)**N2
      B400=K1*R1*SQRHO*N1*B404/(1000.*B58)
     1+K2*R2*SQRHO*RHO*N2*B405/(10000.*B58)
      PFARR=K1*R1*.5/SQRHO*DRRH*B404
      PFARU=B400*U
      PFARV=B400*V
      PFARW=B400*W
      PFARR=K1*R1*.5/SQRHO*DRRH*B404 + K2*R2*1.5*SQRHO*DRRH*B405
      PFART=-PFARR*DRPRT
      PFARP=0.0
      SCLCD=SQRT(CL**2+CD**2)
      ACELG=RHO*B58/64.4*S/MASS*SCLCD
      CALL MULG(50,NERR,ACELG,TAUTAB(1),TAU,TAUTAB(51))
      CALL MULG(50,NERR,ACELG-.1,TAUTAB(1),LOW,TAUTAB(51))
      CALL MULG(50,NERR,ACELG+.1,TAUTAB(1),HIGH,TAUTAB(51))
C     DTRA-DERIVATIVE OF TAU WITH RESPECT TO ACELG
      DTRA=(HIGH-LOW)/.2
      B401=ACELG*DTRA/TAU**2
      B402=2.0+MACH/(CL**2+CD**2)*(CD*PCDRM+CL*PCLRM)
      B403=-B401/B58*B402
      PFBRU=B403*U
      PFBRV=B403*V
      PFBRW=B403*W
      PFBRR=-B401/RHO*(DRRH - RHO*MACH/SOFS*DSRH/
     1(CD**2 + CL**2)*(CD*PCDRM + CL*PCLRM))
      PFBRT=-PFBRR*DRPRT
      PFBRP=0.0
      PFCRU=0.0
      PFCRV=0.0
      PFCRW=0.0
      PFCRR=0.0
      PFCRT=0.0
      PFCRP=0.0
      IF(TIMEA .LE. TIMEH)GO TO 6280
      IF(H.LT.AH2)GO TO 6280
      PFCRR=2.0*AH1/AH2*(H/AH2 - 1.0)
      PFCRT=-PFCRR*DRPRT
 6280 CONTINUE
      B406=K1*R1*SQRHO*B404+K2*R2*SQRHO*RHO*B405
      IF(B406 .LT. AHR2)GO TO 6291
      B407=2.0*AHR1/AHR2*(B406/AHR2-1.0)
      PFDRU=B407*PFARU
      PFDRV=B407*PFARV
      PFDRW=B407*PFARW
      PFDRR=B407*PFARR
      PFDRT=B407*PFART
      PFDRP=B407*PFARP
      GO TO 6292
 6291 PFDRU=0.0
```

```
      PFDRV=0.0
      PFDRW=0.0
      PFDRR=0.0
      PFDRT=0.0
      PFDRP=0.0
 6292 CONTINUE
C     ADD ON TERMS DUE TO THRUST
      IF(EOST(ISTAGE) .EQ. 0.0) GO TO 6300
      PFURU=PFURU+PTRRU/MASS
      PFVRU=PFVRU+PTTRU/MASS
      PFWRU=PFWRU+PTPRU/MASS
      PFURV=PFURV+PTRRV/MASS
      PFVRV=PFVRV+PTTRV/MASS
      PFWRV=PFWRV+PTPRV/MASS
      PFURW=PFURW+PTRRW/MASS
      PFVRW=PFVRW+PTTRW/MASS
      PFWRW=PFWRW+PTPRW/MASS
      PFURR=PFURR+PTRRR/MASS
      PFVRR=PFVRR+PTTRR/MASS
      PFWRR=PFWRR+PTPRR/MASS
      PFURT=PFURT-DRPRT*PTRRR/MASS
      PFVRT=PFVRT-DRPRT*PTTRR/MASS
      PFWRT=PFWRT-DRPRT*PTPRR/MASS
 6700 CONTINUE
C     ADJOINT DERIVATIVE ROUTINE 2
C     COMPUTE DERIVATIVES
 6300 DO 6350 J=1,NT
      DO 6350 I=1,6
 6350 DADJ(I,J) = 0.0
      DO 6351 J=1,NT
      DO 6351 I=1,6
      DO 6351 K=1,6
 6351 DADJ(I,J)=F(K,I)*ADJ(K,J) + CF(K,I)*CADJ(K,J) + DADJ(I,J)
      CALL TRA14
C
C     ADJOINT TRIGGER RETURN
C     COMPUTE G MATRIX, COMPUTE GL, AND PRINT ADJOINT
 6800 GO TO(6801,6851,6900),NTRG
 6801 IF(EOST(ISTAGE) .NE. 0.0) GO TO 6802
      PTRRA=0.0
      PTTRA=0.0
      PTPRA = 0.0
      PTRRS = 0.0
      PTTRS = 0.0
      PTPRS = 0.0
      GO TO 6803
 6802 E1=CALPH/VT
      E2 = SALPH/VT
      E3 = VH*CSIGM
      E4 = VH*SSIGM
      E5 = CSIGM/VH
      E6 = SSIGM/VH
      PTRRS = -THRX*E4*E2 - THRY*E3/VT - THRZ*E4*E1
      PTTRS = THRX*SALPH*(B53*E6/VT - W*E5) + THRY*(E5*B53/VT +
     1W*E6) + THRZ*CALPH*(B53*E6/VT - W*E5)
      PTPRS = THRX*SALPH*(V*E5 + E6*B54/VT) + THRY*(E5*B54/VT -
     1V*E6) + THRZ*CALPH*(V*E5 + E6*B54/VT)
```

```
      PTRRA = THRX*(E1*E3 - U*E2) - THRZ*(E1*U + E3*E2)
      PTTRA = -THRX*(V*E2 + CALPH*(W*E6 + B53/VT*E5)) +
     1THRZ*(SALPH*(W*E6 + B53/VT*E5) - V*E1)
      PTPRA = THRX*(CALPH*(V*E6 - B54*E5/VT) - E2*W) -
     1THRZ*(SALPH*(V*E6 - B54*E5/VT) + W*E1)
 6803 CONTINUE
C     PARTIALS OF CL AND CD RESPECT TO ALPHA
      CALL CLCD(ISTAGE,ALPHR-.005,MACH,CLAL,CDAL)
      CALL CLCD(ISTAGE,ALPHR+.005,MACH,CLAH,CDAH)
      PCLRA=(CLAH-CLAL)/.01
      PCDRA=(CDAH-CDAL)/.01
      B130=VT*PCDRA
      B131=VT*PCLRA
C     COMPUTE INFLUENCE FUNCTIONS AND EXIT TO 7000 WHEN LAST POINT SAVED
      PFURA=-B10*(U*B130 - CSIGM*VH*B131) + PTRRA/MASS
      PFVRA=-B10*(V*B130+B40*B53*B131+B41*W*VT*B131)+PTTRA/MASS
      PFWRA=-B10*(W*B130+B40*B54*B131-B41*V*VT*B131)+PTPRA/MASS
      PFURS=-B10*SSIGM*VH*B21+PTRRS/MASS
      PFVRS=B10*(B41*B21*B53-B40*B21*W*VT)+PTTRS/MASS
      PFWRS=B10*(B41*B21*B54+B40*B21*V*VT)+PTPRS/MASS
      PFRRA=0.0
      PFTRA=0.0
      PFPRA=0.0
      PFRRS=0.0
      PFTRS=0.0
      PFPRS=0.0
      PFARA=0.0
      PFARS=0.0
      PFBRA=-B401/(CD**2 + CL**2)*(CD*PCDRA + CL*PCLRA)
      PFBRS=0.0
      PFCRA=0.0
      PFCRS=0.0
      PFDRA=0.0
      PFDRS=0.0
      PFERA=0.0
      PFERS=0.0
      PFFRA=0.0
      PFFRS=0.0
      IF(TIMEB .EQ. 0.0) NG=NIF+1
      NG=NG-1
      DO 6811 I=1,9
      DO 6811 J=1,2
 6811 GL(I,J,NG)=0.0
      DO 6812 I=1,NT
      DO 6812 J=1,2
      DO 6812 K=1,6
 6812 GL(I,J,NG)=GL(I,J,NG)+G(K,J)*ADJ(K,I)+CG(K,J)*CADJ(K,I)
C     SAVE CONTROL VARIABLES
      ATABS(NG)=ALPHR
      STABS(NG)=SIGMR
C     SET TRIGGER FOR NEXT RETURN
      GTRG=GTRG + DELTB
      IF(MOD(NG,10) .NE. 1) GO TO 6821
      IF(PRTOPT(3) .EQ. 0.0) CALL TRA14
      IF(TIMEB .NE. 0.0) GO TO 6820
C     PRINT ADJOINT HEADINGS
      WRITE(JTAPE,9070) CASENO,ID
```

```
      WRITE(JTAPE,9071) ((CADJ(I,J),I=1,6),J=1,NT)
 6820 WRITE(JTAPE,9072) TIMEA,((ADJ(I,J),I=1,6),(GL(J,I,NG),I=1,2),
     1J=1,NT)
      IF(PRTOPT(3) .LE. 1.0) GO TO 6821
      WRITE(JTAPE,9073) ((DADJ(I,J),I=1,6),J=1,NT)
      WRITE(JTAPE,9074) ((F(I,J),J=1,6),I=1,6),((CF(I,J),J=1,6),I=1,6)
      WRITE(JTAPE,9075) ((G(I,J),I=1,6),J=1,2),((CG(I,J),I=1,6),J=1,2)
 6821 CONTINUE
      IF( NG .NE. 1 ) CALL TRA14
      GO TO 7000
C
 6851 IF(TIMEB .EQ. 0.0) GO TO 6853
      ISTAGE=ISTAGE-1
 6853 IF(ISTAGE .EQ. 1) GO TO 6852
      STRG = TFINAL - EOST(ISTAGE-1)
      CALL TRA24
 6852 CALL OFF(2)
      CALL TRA24
 6900 IF(LAMP(1) .NE. 0.0 .AND. NC .NE. 0) GO TO 6903
 6901 CALL OFF(3)
      CALL TRA14
 6903 IF(TIMEB .NE. 0.0) GO TO 6908
      DO 6905 I=2,9
      KT=I
      IF(LAMP(I) .EQ. 0.0) GO TO 6906
 6905 CONTINUE
      KT=10
 6906 KT=KT-1
 6907 DTRG=TFINAL - LAMP(KT)*TFINAL
      CALL TRA14
 6908 DO 6909 I=1,12
 6909 STVRS(I,KT)=STVR(I)
      DO 6910 J=1,8
      DO 6910 I=1,6
 6910 LAMBDA(I,J,KT)=ADJ(I,J+1)
      KT=KT - 1
      IF(KT .EQ. 0) GO TO 6901
      GO TO 6907
C
 7000 CONTINUE
      DELTA=DELTB
C     COMPUTE INTEGRALS SIMPSON-S RULE
      DO 7001 I=1,9
      DO 7001 J=1,9
 7001 INTGL(I,J)=0.0
      DO 7005 I=1,NIF
      TIME=FLOAT(I-1)*DELTA
      CALL MULG(50,NERR,TIME,WTAB,WT(1,1),WTAB(1,2),WT(2,2),WTAB(1,3))
      SC=1.0
      IF(I .EQ. 1 .OR. I.EQ. NIF)GO TO 7003
      SC=4.0
      IF(MOD(I,2) .NE. 0) SC=2.0
 7003 CONTINUE
      DO 7005 J=1,NT
      DO 7005 K=1,J
      DO 7005 M=1,2
 7005 INTGL(J,K)=SC*GL(J,M,I)/WT(M,M)*GL(K,M,I)+INTGL(J,K)
```

```
      DO 7007 J=1,NT
      DO 7007 K=1,J
      INTGL(J,K)=INTGL(J,K)*DELTA/3.0
 7007 INTGL(K,J)=INTGL(J,K)
      IHH=INTGL(1,1)
      DO 7009 I=1,8
 7009 ISH(I)=INTGL(I+1,1)
      DO 7011 I=1,8
      DO 7011 J=1,8
 7011 ISS(I,J)=INTGL(I+1,J+1)
      RETURN
 8001 WRITE(JTAPE,9010)
 9010 FORMAT( 11H ERROR MARK )
      NF=2
 9011 FORMAT ( 6E15.5 )
      RETURN
 9070 FORMAT( 107H1 RAYTHEON THREE-DIMENSIONAL TRAJECTORY OPTIMIZATION P
     1ROGRAM  TOS9   ADJOINT TRAJECTORY       CASENO   ITNO  /
     292X,F8.0,F7.1,2X,3A6)
 9071 FORMAT( 28H0 CONSTANT ADJOINT EQUATIONS  / 12X, 48H     LHEAT
     1LACEL PF     LALT PF    LHTRT PF      / (1PE24.4,5E12.4))
 9072 FORMAT( 108H0       TIME           LU           LV           LW
     1   LR           LT           LP          GLA          GLS    /
     2 F12.3,1P8E12.4   /(E24.4,7E12.4))
 9073 FORMAT( 13H0 DERIVATIVES   / (1PE24.4,5E12.4))
 9074 FORMAT( 10H0 F MATRIX   / (1X,1P6E12.4))
 9075 FORMAT( 10H0 G MATRIX      / (1X,1P6E12.4))
      END
```

```
$IBFTC CLD
      SUBROUTINE CLCD(ISTAGE,ALPHR,MACH,CL,CD)
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      ALPHA=ABS(ALPHR*57.2957795)
      CALL BILIN(N,MACH,MTAB(1,ISTAGE),ALPHA,ATAB(1,ISTAGE),
     1CL,CLTAB(1,ISTAGE),CD,CDTAB(1,ISTAGE))
      CL=CL*SIGN(1.0,ALPHR)
      RETURN
      END
```

```
$IBFTC PRL
      SUBROUTINE PARTAL(FNO,STOR)
C
C     C O M M O N   F O R   A L L   P R O G R A M S
C
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      COMMON / VAR / RADIAN,D10,D11,D12,D13,D14,D20,D30,D31,D32,
     1ALPHD,SIGMD,ALPHR,SIGMR,CSIGM,SSIGM,CTHET,STHET,R,H,VHSQ,
     2VTSQ,VH,VT,PEA,RHO,SOFS,MACH,MASS,TH,THR,COSIT,SINIT,COSDT,
     3SINDT,THRX,THRY,THRZ,CALPH,SALPH,C1,C2,C3,C4,C5,C6,THRR,THRT,
     4THRP,S,X1,X2,X3,X4,X5,X6,X7,X8,X9,RSQ,R4TH,SQRHO,ACELG,TAU,
     5TIMEH,CL,CD,TFINAL,B56,B58,PEH,RHOH,SOFSH,PEL,RHOL,SOFSL,
     6DPERH,DRRH,DSRH,CLHM,CDHM,CLLM,CDLM,PCLRM,PCDRM,DRPRT,B10,
     7B20,B21,B30,B31,B40,B41,B50,B51,B52,B53,B54,B55,B57,B60,B61,
     8B62,B70,B71,B81,B82,B90,B91,B92,B93,B100,B101,B102,B103,B104,
     9R3RD,R5TH,VT3,B200,B201,VH3,B202,B204,PTRRU,PTTRU,PTPRU
      COMMON / VAR / PTRRV,PTTRV,PTPRV,PTRRW,PTTRW,PTPRW,PTHRH,
     1PTXRH,PTYRH,PTZRH,B300,B301,B302,PTRRR,PTTRR,PTPRR,B404,
     2B405,B400,SCLCD,DTRA,B401,B402,B403,B406,B407,PTRRA,PTTRA,
     3PTPRA,PTRRS,PTTRS,PTPRS,E1,E2,E3,E4,E5,E6,CLAL,CDAL,CLAH,
     4CDAH,PCLRA,PCDRA,B130,B131,PTRG,STRG,DTRG,GTRG,SRAT,
     5MUD,NUD,PHISD,THETSD,BETAD,GAMMAD,THETD,PHID,DYNA,ENER,AST,RG
      COMMON / VAR / SFRM(1500),GL(9,2,201),ATABS(201),STABS(201),
     1PARTS(14),PARTC(14),CADJ(6,9),F(6,6),CF(6,6),G(6,2),CG(6,2)
      COMMON / VAR / LAMP(9),VARX(25)
      REAL MACH,MASS,LAMP
      COMMON / ANAL / HI,HIS,SIA(8),SIAS(8),SIE(8),SIES(8),CSIA,
     1CSIAS,CDSIP,CDSIA,CRDSI,CHIA,CHIAS,CDHIA,CDHIP,CRDHI,DSIA(8),
     2RDSI(8),DHIA,DHIP,RDHI,DP,EHI,CEHI,INTGL(9,9),IHH,ISH(8),
     3ISS(8,8),WT(2,2),F1,F2,SID(8),SFC,DSI(8),DSIJ,DBETA(8),
     4ISSIL(8,8),DPSQ,DPSQK,IHHK,ISHK(8),ISSK(8,8),
     5ISSILK(8,8),DBETAK(8),STVRS(12,9),LAMBDA(6,8,9),SFCK,F1K,
     6F2K
      REAL INTGL,IHH,ISH,ISS,ISSIL,INTGLK,IHHK,ISHK,ISSK,ISSILK,
     1LAMBDA
      COMMON / IVAR / JTAPE,KTAPE,LTAPE,ITN,ITTN,ISTAGE,NC,
     1NT,NRTN,NTRG,KACC,KA,KS,NIF,NG,L(8),LS(8)
      EQUIVALENCE(SFRM(2000),NORA(2000),HNA(1997),NEQA(1996),
     1TIMEA(1995),U(1993),V(1992),W(1991),RX(1990),THETR(1989),
     2PHIR(1988),PA(1987),PB(1986),PC(1985),PD(1984),PE(1983),
     3PF(1982),U1(1981),V1(1980),W1(1979),RX1(1978),THETR1(1977),
     4PHIR1(1976),PA1(1975),PB1(1974),PC1(1973),PD1(1972),PE1(1971),
     5PF1(1970))
      DIMENSION ADJ(6,9), DADJ(6,9)
      EQUIVALENCE(SFRM(2000),NORB(1981),HNB(1978),NEQB(1977),
     1TIMEB(1976),ADJ(1974),DADJ(1920))
```

```
      DIMENSION STVR(12)
      EQUIVALENCE(STVR(1),U)
C
C     E N D  O F  C O M M O N  F O R  A L L  P R O G R A M S
C
      DIMENSION STOR(14),HS(13)
C     SUBROUTINE TO COMPUTE PARTIAL DERIVATIVES OF FUNCTION(FNO)
C     FNO=NUMBER OF FUNCTION
C     STOR=STORAGE FOR PARTIAL DERIVATIVES
      EQUIVALENCE(U,STATE(1)),(U1,DERV(1))
      DIMENSION STATE(13),DERV(13)
C     STOR(1)THROUGH STOR(13)PARTIAL DERIVATIVES OF FUNCTION(FNO)
C     WITH RESPECT TO U,V,W,H,THETR,PHIR,A,B,C,D,E,F,AND TIME,
C     RESPECTIVELY
C     STOR(14)TOTAL DERIVATIVE OF FUNCTION(FNO)WITH RESPECT TO TIME
      REAL LOW
    1 HS(1)=1.0
      HS(2)=1.0
      HS(3)=1.0
      HS(4)=500.0
      HS(5)=.01
      HS(6)=.01
      HS(7)=1.0
      HS(8)=1.0
      HS(9)=1.0
      HS(10)=1.0
      HS(11)=1.0
      HS(12)=1.0
      HS(13)=1.0
      DO 10 I=1,12
      STATE(I)=STATE(I)-HS(I)
      LOW=FUNCT(FNO)
      STATE(I)=STATE(I)+HS(I)+HS(I)
      HIGH=FUNCT(FNO)
      STOR(I)=(HIGH-LOW)/(2.0*HS(I))
      STATE(I)=STATE(I)-HS(I)
   10 CONTINUE
      TIMEA=TIMEA-HS(13)
      LOW=FUNCT(FNO)
      TIMEA=TIMEA + HS(13) + HS(13)
      HIGH=FUNCT(FNO)
      TIMEA=TIMEA - HS(13)
      STOR(13)=(HIGH - LOW)/(2.0*HS(13))
      STOR(14)=STOR(13)
      DO 20 I=1,12
      STOR(14)=STOR(14)+STOR(I)*DERV(I)
   20 CONTINUE
      RETURN
      END
```

```
$IBFTC FCT
      FUNCTION FUNCT(FNO)
C
C     C O M M O N   F O R   A L L   P R O G R A M S
C
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      COMMON / VAR / RADIAN,D10,D11,D12,D13,D14,D20,D30,D31,D32,
     1ALPHD,SIGMD,ALPHR,SIGMR,CSIGM,SSIGM,CTHET,STHET,R,H,VHSQ,
     2VTSQ,VH,VT,PEA,RHO,SOFS,MACH,MASS,TH,THR,COSIT,SINIT,COSDT,
     3SINDT,THRX,THRY,THRZ,CALPH,SALPH,C1,C2,C3,C4,C5,C6,THRR,THRT,
     4THRP,S,X1,X2,X3,X4,X5,X6,X7,X8,X9,RSQ,R4TH,SQRHO,ACELG,TAU,
     5TIMEH,CL,CD,TFINAL,B56,B58,PEH,RHOH,SOFSH,PEL,RHOL,SOFSL,
     6DPERH,DRRH,DSRH,CLHM,CDHM,CLLM,CDLM,PCLRM,PCDRM,DRPRT,B10,
     7B20,B21,B30,B31,B40,B41,B50,B51,B52,B53,B54,B55,B57,B60,B61,
     8B62,B70,B71,B81,B82,B90,B91,B92,B93,B100,B101,B102,B103,B104,
     9R3RD,R5TH,VT3,B200,B201,VH3,B202,B204,PTRRU,PTTRU,PTPRU
      COMMON / VAR / PTRRV,PTTRV,PTPRV,PTRRW,PTTRW,PTPRW,PTHRH,
     1PTXRH,PTYRH,PTZRH,B300,B301,B302,PTRRR,PTTRR,PTPRR,B404,
     2B405,B400,SCLCD,DTRA,B401,B402,B403,B406,B407,PTRRA,PTTRA,
     3PTPRA,PTRRS,PTTRS,PTPRS,E1,E2,E3,E4,E5,E6,CLAL,CDAL,CLAH,
     4CDAH,PCLRA,PCDRA,B130,B131,PTRG,STRG,DTRG,GTRG,SRAT,
     5MUD,NUD,PHISD,THETSD,BETAD,GAMMAD,THETD,PHID,DYNA,ENER,AST,RG
      COMMON / VAR / SFRM(1500),GL(9,2,201),ATABS(201),STABS(201),
     1PARTS(14),PARTC(14),CADJ(6,9),F(6,6),CF(6,6),G(6,2),CG(6,2)
      COMMON / VAR / LAMP(9),VARX(25)
      REAL MACH,MASS,LAMP
      COMMON / ANAL / HI,HIS,SIA(8),SIAS(8),SIE(8),SIES(8),CSIA,
     1CSIAS,CDSIP,CDSIA,CRDSI,CHIA,CHIAS,CDHIA,CDHIP,CRDHI,DSIA(8),
     2RDSI(8),DHIA,DHIP,RDHI,DP,EHI,CEHI,INTGL(9,9),IHH,ISH(8),
     3ISS(8,8),WT(2,2),F1,F2,SID(8),SFC,DSI(8),DSIJ,DBETA(8),
     4ISSIL(8,8),DPSQ,DPSQK,IHHK,ISHK(8),ISSK(8,8),
     5ISSILK(8,8),DBETAK(8),STVRS(12,9),LAMBDA(6,8,9),SFCK,F1K,
     6F2K
      REAL INTGL,IHH,ISH,ISS,ISSIL,INTGLK,IHHK,ISHK,ISSK,ISSILK,
     1LAMBDA
      COMMON / IVAR / JTAPE,KTAPE,LTAPE,ITN,ITTN,ISTAGE,NC,
     1NT,NRTN,NTRG,KACC,KA,KS,NIF,NG,L(8),LS(8)
      EQUIVALENCE(SFRM(2000),NORA(2000),HNA(1997),NEQA(1996),
     1TIMEA(1995),U(1993),V(1992),W(1991),RX(1990),THETR(1989),
     2PHIR(1988),PA(1987),PB(1986),PC(1985),PD(1984),PE(1983),
     3PF(1982),U1(1981),V1(1980),W1(1979),RX1(1978),THETR1(1977),
     4PHIR1(1976),PA1(1975),PB1(1974),PC1(1973),PD1(1972),PE1(1971),
     5PF1(1970))
      DIMENSION ADJ(6,9), DADJ(6,9)
      EQUIVALENCE(SFRM(2000),NORB(1981),HNB(1978),NEQB(1977),
     1TIMEB(1976),ADJ(1974),DADJ(1920))
```

```
      DIMENSION STVR(12)
      EQUIVALENCE(STVR(1),U)
C
C     E N D  O F  C O M M O N  F O R  A L L  P R O G R A M S
C
      COMMON / ISTATE / UN,VN,WN,RXN,THETN,PHIN,VHN,CTHTN,STHTN
      REAL MUR,MUD,NUR,NUD
      DIMENSION A(3)
C     SUBROUTINE TO COMPUTE PAYOFF,STOPPING, AND CONSTRAINT FUNCTIONS
      N=ABS(FNO)
      IF(N .NE. 0) GO TO 100
   99 FUNCT=0.0
      RETURN
  100 IF( N .GT. 23) GO TO 99
      GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23)
     1,N
    1 FUNCT=TIMEA
      RETURN
    2 FUNCT=U
      RETURN
    3 FUNCT=V
      RETURN
    4 FUNCT=W
      RETURN
    5 FUNCT=RX+RE
      RETURN
    6 FUNCT=THETR*57.2957795
      RETURN
    7 FUNCT=PHIR*57.2957795
      RETURN
    8 FUNCT=PA
      RETURN
    9 FUNCT=PB
      RETURN
   10 FUNCT=PC
      RETURN
   11 FUNCT=PD
      RETURN
   12 FUNCT=PE
      RETURN
   13 FUNCT=PF
      RETURN
   14 CONTINUE
   15 CONTINUE
   16 CONTINUE
   17 N=N-13
      SPMPN=SIN(PHIR-PHIN)
      CPMPN=COS(PHIR-PHIN)
      CNSM=(SIN(THETR)*(VN*CTHTN*CPMPN + WN*SPMPN) -
     1VN*STHTN*COS(THETR))/VHN
      CNCM=CTHTN*COS(THETR) + STHTN*SIN(THETR)*CPMPN
      MUR=ATAN2(CNSM,CNCM)
      SINMU=SIN(MUR)
      SINNU=(WN*STHTN*COS(THETR) - SIN(THETR)*(WN*CTHTN*
     1CPMPN - VN*SPMPN))/VHN
      NUR=ATAN2(SINNU*SINMU,CNSM)
      TANNU=SINNU*SINMU/CNSM
```

```
      PHIS=ATAN2(TANNU,SINMU)
      THETS=ATAN2(SQRT(SINNU**2 + COS(NUR)**2*SINMU**2),COS(NUR)
     1*COS(MUR))
      MUD=MUR*RADIAN
      NUD=NUR*RADIAN
      PHISD=PHIS*RADIAN
      THETSD=THETS*RADIAN
      GO TO (124,125,126,127),N
  124 FUNCT=MUD
      RETURN
  125 FUNCT=NUD
      RETURN
  126 FUNCT=THETSD
      RETURN
  127 FUNCT=PHISD
      RETURN
   18 FUNCT=RX + (RE - RPC)*COS(THETR)**2
      RETURN
   19 FUNCT=SQRT(U**2 + V**2 + W**2)
      RETURN
   20 R=RX + RE
      FUNCT=MASS*((U**2+V**2+W**2)/2.0-MU/R-(1.0-3.0*COS(THETR)
     1**2)/3.0*(RF/R)**3*MU*GJ/RF)
      RETURN
   21 FUNCT=ATAN2(U,SQRT(V**2 + W**2))*57.2957795
      RETURN
   22 FUNCT=ATAN2(W,V)*57.2957795
      RETURN
   23 CALL ATM(RX + (RE - RPC)*COS(THETR)**2,A)
      FUNCT=.5*A(2)*(U**2 + V**2 + W**2)
      RETURN
      END
```

```
$IBFTC SFR
      SUBROUTINE SFRWD(M)
C
C     C O M M O N   F O R   A L L   P R O G R A M S
C
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      COMMON / VAR / RADIAN,D10,D11,D12,D13,D14,D20,D30,D31,D32,
     1ALPHD,SIGMD,ALPHR,SIGMR,CSIGM,SSIGM,CTHET,STHET,R,H,VHSQ,
     2VTSQ,VH,VT,PEA,RHO,SOFS,MACH,MASS,TH,THR,COSIT,SINIT,COSDT,
     3SINDT,THRX,THRY,THRZ,CALPH,SALPH,C1,C2,C3,C4,C5,C6,THRR,THRT,
     4THRP,S,X1,X2,X3,X4,X5,X6,X7,X8,X9,RSQ,R4TH,SQRHO,ACELG,TAU,
     5TIMEH,CL,CD,TFINAL,B56,B58,PEH,RHOH,SOFSH,PEL,RHOL,SOFSL,
     6DPERH,DRRH,DSRH,CLHM,CDHM,CLLM,CDLM,PCLRM,PCDRM,DRPRT,B10,
     7B20,B21,B30,B31,B40,B41,B50,B51,B52,B53,B54,B55,B57,B60,B61,
     8B62,B70,B71,B81,B82,B90,B91,B92,B93,B100,B101,B102,B103,B104,
     9R3RD,R5TH,VT3,B200,B201,VH3,B202,B204,PTRRU,PTTRU,PTPRU
      COMMON / VAR / PTRRV,PTTRV,PTPRV,PTRRW,PTTRW,PTPRW,PTHRH,
     1PTXRH,PTYRH,PTZRH,B300,B301,B302,PTRRR,PTTRR,PTPRR,B404,
     2B405,B400,SCLCD,DTRA,B401,B402,B403,B406,B407,PTRRA,PTTRA,
     3PTPRA,PTRRS,PTTRS,PTPRS,E1,E2,E3,E4,E5,E6,CLAL,CDAL,CLAH,
     4CDAH,PCLRA,PCDRA,B130,B131,PTRG,STRG,DTRG,GTRG,SRAT,
     5MUD,NUD,PHISD,THETSD,BETAD,GAMMAD,THETD,PHID,DYNA,ENER,AST,RG
      COMMON / VAR / SFRM(1500),GL(9,2,201),ATABS(201),STABS(201),
     1PARTS(14),PARTC(14),CADJ(6,9),F(6,6),CF(6,6),G(6,2),CG(6,2)
      COMMON / VAR / LAMP(9),VARX(25)
      REAL MACH,MASS,LAMP
      COMMON / ANAL / HI,HIS,SIA(8),SIAS(8),SIE(8),SIES(8),CSIA,
     1CSIAS,CDSIP,CDSIA,CRDSI,CHIA,CHIAS,CDHIA,CDHIP,CRDHI,DSIA(8),
     2RDSI(8),DHIA,DHIP,RDHI,DP,EHI,CEHI,INTGL(9,9),IHH,ISH(8),
     3ISS(8,8),WT(2,2),F1,F2,SID(8),SFC,DSI(8),DSIJ,DBETA(8),
     4ISSIL(8,8),DPSQ,DPSQK,IHHK,ISHK(8),ISSK(8,8),
     5ISSILK(8,8),DBETAK(8),STVRS(12,9),LAMBDA(6,8,9),SFCK,F1K,
     6F2K
      REAL INTGL,IHH,ISH,ISS,ISSIL,INTGLK,IHHK,ISHK,ISSK,ISSILK,
     1LAMBDA
      COMMON / IVAR / JTAPE,KTAPE,LTAPE,ITN,ITTN,ISTAGE,NC,
     1NT,NRTN,NTRG,KACC,KA,KS,NIF,NG,L(8),LS(8)
      EQUIVALENCE(SFRM(2000),NORA(2000),HNA(1997),NEQA(1996),
     1TIMEA(1995),U(1993),V(1992),W(1991),RX(1990),THETR(1989),
     2PHIR(1988),PA(1987),PB(1986),PC(1985),PD(1984),PE(1983),
     3PF(1982),U1(1981),V1(1980),W1(1979),RX1(1978),THETR1(1977),
     4PHIR1(1976),PA1(1975),PB1(1974),PC1(1973),PD1(1972),PE1(1971),
     5PF1(1970))
      DIMENSION ADJ(6,9), DADJ(6,9)
      EQUIVALENCE(SFRM(2000),NORB(1981),HNB(1978),NEQB(1977),
     1TIMEB(1976),ADJ(1974),DADJ(1920))
```

```
      DIMENSION STVR(12)
      EQUIVALENCE(STVR(1),U)
C
C     E N D  O F  C O M M O N  F O R  A L L  P R O G R A M S
C
      COMMON / FORW / STRAJ(20,13)
      GO TO (1,2),M
    1 CONTINUE
      IF(TIMEA .EQ. 0.0) GO TO 5
      IF(TIMEA .LT. PTIMEA) RETURN
      PTIMEA=TIMEA
      GO TO 7
    5 PTIMEA=0.0
      N=0
    7 N=N+1
      STRAJ(N,1)=TIMEA
      STRAJ(N,2)=U
      STRAJ(N,3)=V
      STRAJ(N,4)=W
      STRAJ(N,5)=RX
      STRAJ(N,6)=THETR
      STRAJ(N,7)=PHIR
      STRAJ(N,8)=PA
      STRAJ(N,9)=PB
      STRAJ(N,10)=PC
      STRAJ(N,11)=PD
      STRAJ(N,12)=PE
      STRAJ(N,13)=PF
      IF(N .NE. 19) RETURN
    2 CONTINUE
      WRITE(LTAPE) N,((STRAJ(I,J),I=1,19),J=1,13)
      N=0
      DO 10 I=1,19
      DO 10 J=1,13
   10 STRAJ(I,J)=0.0
      RETURN
      END
```

```
$IBFTC ITM
      REAL FUNCTION ITEM(I,J)
C
C     C O M M O N   F O R   A L L   P R O G R A M S
C
      COMMON / INPUT / CASENO(2),ID(3),XVT,XGAMMD,XBETAD,XH,XTHETD,
     1XPHID,GS,GMASS,PAYOFF,STOP,STOPV,HP,AIT,XDPSQ,A11,A12,A21,
     2A22,A31,A32,A33,DELF,CEHIM,DPSQM,K1,R1,N1,K2,R2,N2,AH1,AH2,
     3AHR1,AHR2,LAM,SLPE,MU,RF,GJ,OMEGA,RE,RPC,TLIM,HLIM,
     4A23,A34,ORDA,HNOMA,EUA,ELA,HMAXA,HMINA,YCA,
     5ORDB,HNOMB,EUB,ELB,HMAXB,HMINB,YCB
      COMMON / INPUT / ANTABX(100),SNTABX(100),CON(8),SIT(8),SIB(8),
     1EPS(8),EOST(4),AE(3),REFA(3),IT(3),DT(3),TMTAB(150),TAUTAB(100),
     2WTAB(50,3),MTAB(11,4),ATAB(26,4),CLTAB(150,4),CDTAB(150,4),DELTA,
     3ATABX(201),STABX(201),LAMTAB(9),PRTOPT(4),INPUTX(10)
      COMMON / INPUT / HFMT(220),DFMT(220),IGO(340)
      REAL K1,N1,K2,N2,IT,MU,LAM,LAMTAB
      COMMON / VAR / RADIAN,D10,D11,D12,D13,D14,D20,D30,D31,D32,
     1ALPHD,SIGMD,ALPHR,SIGMR,CSIGM,SSIGM,CTHET,STHET,R,H,VHSQ,
     2VTSQ,VH,VT,PEA,RHO,SOFS,MACH,MASS,TH,THR,COSIT,SINIT,COSDT,
     3SINDT,THRX,THRY,THRZ,CALPH,SALPH,C1,C2,C3,C4,C5,C6,THRR,THRT,
     4THRP,S,X1,X2,X3,X4,X5,X6,X7,X8,X9,RSQ,R4TH,SQRHO,ACELG,TAU,
     5TIMEH,CL,CD,TFINAL,B56,B58,PEH,RHOH,SOFSH,PEL,RHOL,SOFSL,
     6DPERH,DRRH,DSRH,CLHM,CDHM,CLLM,CDLM,PCLRM,PCDRM,DRPRT,B10,
     7B20,B21,B30,B31,B40,B41,B50,B51,B52,B53,B54,B55,B57,B60,B61,
     8B62,B70,B71,B81,B82,B90,B91,B92,B93,B100,B101,B102,B103,B104,
     9R3RD,R5TH,VT3,B200,B201,VH3,B202,B204,PTRRU,PTTRU,PTPRU
      COMMON / VAR / PTRRV,PTTRV,PTPRV,PTRRW,PTTRW,PTPRW,PTHRH,
     1PTXRH,PTYRH,PTZRH,B300,B301,B302,PTRRR,PTTRR,PTPRR,B404,
     2B405,B400,SCLCD,DTRA,B401,B402,B403,B406,B407,PTRRA,PTTRA,
     3PTPRA,PTRRS,PTTRS,PTPRS,E1,E2,E3,E4,E5,E6,CLAL,CDAL,CLAH,
     4CDAH,PCLRA,PCDRA,B130,B131,PTRG,STRG,DTRG,GTRG,SRAT,
     5MUD,NUD,PHISD,THETSD,BETAD,GAMMAD,THETD,PHID,DYNA,ENER,AST,RG
      COMMON / VAR / SFRM(1500),GL(9,2,201),ATABS(201),STABS(201),
     1PARTS(14),PARTC(14),CADJ(6,9),F(6,6),CF(6,6),G(6,2),CG(6,2)
      COMMON / VAR / LAMP(9),VARX(25)
      REAL MACH,MASS,LAMP
      COMMON / ANAL / HI,HIS,SIA(8),SIAS(8),SIE(8),SIES(8),CSIA,
     1CSIAS,CDSIP,CDSIA,CRDSI,CHIA,CHIAS,CDHIA,CDHIP,CRDHI,DSIA(8),
     2RDSI(8),DHIA,DHIP,RDHI,DP,EHI,CEHI,INTGL(9,9),IHH,ISH(8),
     3ISS(8,8),WT(2,2),F1,F2,SID(8),SFC,DSI(8),DSIJ,DBETA(8),
     4ISSIL(8,8),DPSQ,DPSQK,IHHK,ISHK(8),ISSK(8,8),
     5ISSILK(8,8),DBETAK(8),STVRS(12,9),LAMBDA(6,8,9),SFCK,F1K,
     6F2K
      REAL INTGL,IHH,ISH,ISS,ISSIL,INTGLK,IHHK,ISHK,ISSK,ISSILK,
     1LAMBDA
      COMMON / IVAR / JTAPE,KTAPE,LTAPE,ITN,ITTN,ISTAGE,NC,
     1NT,NRTN,NTRG,KACC,KA,KS,NIF,NG,L(8),LS(8)
      EQUIVALENCE(SFRM(2000),NORA(2000),HNA(1997),NEQA(1996),
     1TIMEA(1995),U(1993),V(1992),W(1991),RX(1990),THETR(1989),
     2PHIR(1988),PA(1987),PB(1986),PC(1985),PD(1984),PE(1983),
     3PF(1982),U1(1981),V1(1980),W1(1979),RX1(1978),THETR1(1977),
     4PHIR1(1976),PA1(1975),PB1(1974),PC1(1973),PD1(1972),PE1(1971),
     5PF1(1970))
      DIMENSION ADJ(6,9), DADJ(6,9)
      EQUIVALENCE(SFRM(2000),NORB(1981),HNB(1978),NEQB(1977),
     1TIMEB(1976),ADJ(1974),DADJ(1920))
```

```
      DIMENSION STVR(12)
      EQUIVALENCE(STVR(1),U)
C
C     E N D  O F  C O M M O N  F O R  A L L  P R O G R A M S
C
      EQUIVALENCE(COM1(1),CASENO(1)),(COM2(1),RADIAN),(COM3(1),HI),
     1(COM4(1),JTAPE)
      REAL JTAPE
      DIMENSION COM1(1),COM2(1),COM3(1),COM4(1)
      IF( J .GE. 1 .AND. J .LE. 4) GO TO 10
C     CHECK FOR J WITHIN LIMITS
      ITEM=0.0
      RETURN
   10 CONTINUE
      GO TO(1,2,3,4),J
    1 ITEM=COM1(I)
      RETURN
    2 ITEM=COM2(I)
      RETURN
    3 ITEM=COM3(I)
      RETURN
    4 ITEM=COM4(I)
      RETURN
      END
```

```
$IBFTC TML
      FUNCTION TMAML(A,B,C,L)
      DIMENSION A(8),B(8,8),C(8),L(8),D(8),E(8,8)
C     TRIPLE MATRIX PRODUCT
C     ELIMINATING ROWS AND COLUMNS WHERE L(I) =0
      DO 10 I=1,8
   10 D(I)=0.0
      DO 30 I=1,8
      IF(L(I) .EQ.0)GO TO 30
      DO 20 J=1,8
      IF( L(J) .EQ. 0 ) GO TO 20
      D(I)=D(I)+A(J)*B(J,I)
   20 CONTINUE
   30 CONTINUE
      TMAML=0.0
      DO 40 I=1,8
      TMAML=TMAML+D(I)*C(I)
   40 CONTINUE
      RETURN
      END
```

```
$IBFTC MVL
C       TO FIND INVERSE OF A MATRIX ELIMINATING SPECIFIED
C       ROWS AND COLUMNS
        SUBROUTINE MINVL(A,AI,ID)
        DIMENSION A(8,8),ID(8),AI(8,8)
        DIMENSION B(8,8)
        DO 99 I=1,6
        DO 99 J=1,6
   99   AI(I,J)=0.0
        IROW=0
        DO 100 I=1,8
        IF(ID(I) .EQ. 0)GO TO 100
        IROW=IROW+1
        ICOL=0
        DO 102 J=1,8
        IF(ID(J) .EQ. 0)GO TO 102
        ICOL=ICOL+1
        B(IROW,ICOL)=A(I,J)
  102   CONTINUE
  100   CONTINUE
        IF( IROW .EQ. 0 ) GO TO 204
        CALL MATINV(B,IROW,DUMMY,0,DUMMY)
C       PUT INVERSE BACK WITH ZERO'S REPLACING ELEMENTS ELIMINATED
        IROW=0
        DO 200 I=1,8
        IF(ID(I) .EQ. 0)GO TO 200
        IROW=IROW+1
        ICOL=0
        DO 202 J=1,8
        IF(ID(J) .EQ. 0)GO TO 202
        ICOL=ICOL+1
        AI(I,J)=B(IROW,ICOL)
  202   CONTINUE
  200   CONTINUE
  204   RETURN
        END
```

```
$IBFTC MIV
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
      SUBROUTINE MATINV(A,N,B,M,DETERM)
C
      DIMENSION IPIVOT(8),A(8,8),B(8,1),INDEX(8,2),PIVOT(8)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
C
C     INITIALIZATION
C
   10 DETERM=1.0
   15 DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J)-1) 60, 105, 60
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1) 80, 100, 740
   80 IF (ABS(AMAX)-ABS(A(J,K))) 85, 100, 100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW-ICOLUM) 140, 260, 140
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L) = A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  205 IF(M) 260, 260, 210
  210 DO 250 L=1, M
  220 SWAP=B(IROW,L)
  230 B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT(I)=A(ICOLUM,ICOLUM)
  320 DETERM=DETERM*PIVOT(I)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICOLUM,L) = A(ICOLUM,L)/PIVOT(I)
  355 IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)
C
```

```
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L) = A(L1,L) - A(ICOLUM,L)*T
  455 IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
```

```
$IBFTC OUT
      SUBROUTINE OUT(AR,HFMT,DFMT)
      DIMENSION AR(85),ARI(85),AH(85),IT(2,40),ITH(2,20),HFMT(40),
     1HFMTH(40),DFMT(40),DFMTH(40),TAB(40),THAB(20)
      EQUIVALENCE(ARI(85),IGO(85),ITYPE(84),INH(83),INL(82),M(81),
     1IT(80)),                    (AH(85),INHH(83),INLH(82),MH(81),
     2ITH(80),HFMTH(40))
      LOGICAL LGC     ,PRINTG,PUNCHG,HEADG,DATAG,COLVEC,ROWVEC
      DIMENSION LGC(10)
      LOGICAL NEWPG
      REAL ITEM
      EQUIVALENCE (LGC(10),PRINTG(10),PUNCHG(9),HEADG(8),DATAG(7),
     1COLVEC(6),ROWVEC(5))
      MM=46
      DO 10 I=1,85
   10 ARI(I)=AR(I)
      DO 20 I=1,10
   20 LGC(I)=.FALSE.
      IF(IGO.EQ.1)PRINTG=.TRUE.
      IF(IGO.EQ.2)PUNCHG=.TRUE.
      IF(.NOT.(PRINTG.OR.PUNCHG))RETURN
      GO TO (30,40,50,60),ITYPE
   30 IF(PUNCHG)RETURN
      HEADG=.TRUE.
      DO 31 I=1,45
   31 AH(I)=ARI(I)
      DO 310 I=1,40
      HFMTH(I)=HFMT(I)
  310 DFMTH(I)=DFMT(I)
      ASSIGN 70 TO LINK
   32 WRITE(6,1001)
 1001 FORMAT( 1H1 )
      LCT=1
      IF(INHH.EQ.0)GO TO 33
      LCT=LCT+INHH
       WRITE(6,HFMTH)
   33 IF(MH.EQ.0)GO TO 35
      DO 34 K=1,MH
   34 THAB(K)=ITEM(ITH(1,K),ITH(2,K))
      WRITE(6,DFMTH)(THAB(K),K=1,MH)
      LCT=LCT+INLH
   35 NEWPG=.TRUE.
      GO TO LINK,(70,71,72)
   70 RETURN
   40 DATAG=.TRUE.
      IF(PUNCHG)GO TO 41
      IF(LCT+INL.LE.MM)GO TO 71
      ASSIGN 71 TO LINK
      GO TO 32
   71 IF(.NOT.NEWPG)GO TO 41
      NEWPG=.FALSE.
      IF(INH.EQ.0)GO TO 41
      LCT=LCT+INH
      WRITE(6,HFMT)
   41 IF(M.EQ.0)RETURN
      DO 42 I=1,M
   42 TAB(I)=ITEM(IT(1,I),IT(2,I))
```

```
      IF(PRINTG)WRITE(6,DFMT)(TAB(I),I=1,M)
      IF(PUNCHG)WRITE(7,DFMT)(TAB(I),I=1,M)
      IF(PRINTG)LCT=LCT+INL
      RETURN
   50 COLVEC=.TRUE.
      GO TO 61
   60 ROWVEC=.TRUE.
   61 IF(PUNCHG)GO TO 62
      IF(LCT+INH+INL*IT(1,2).LE.MM)GO TO 72
      ASSIGN 72 TO LINK
      GO TO 32
   72 IF(INH.EQ.0)GO TO 62
      LCT=LCT+INH
      WRITE(6,HFMT)
   62 LOOP=IT(1,2)
      IF((LOOP.EQ.0).OR.(M.EQ.0))RETURN
      DO 63 J=1,LOOP
      IF(PUNCHG)GO TO 73
      IF(LCT+INL.LE.MM)GO TO 73
      WRITE(6,1001)
      LCT=1
      IF(INHH.EQ.0)GO TO 93
      LCT=LCT+INHH
      WRITE(6,HFMTH)
   93 IF(MH.EQ.0)GO TO 95
      DO 94 K=1,MH
   94 THAB(K)=ITEM(ITH(1,K),ITH(2,K))
      WRITE(6,DFMTH)(THAB(K),K=1,MH)
      LCT=LCT+INLH
   95 NEWPG=.TRUE.
   73 DO 64 I=1,M
      IF(COLVEC)GO TO 52
      TAB(I)=ITEM(IT(1,1)+J-1+IT(2,2)*(I-1),IT(2,1))
      GO TO 64
   52 TAB(I)=ITEM(IT(1,1)+I-1+IT(2,2)*(J-1),IT(2,1))
   64 CONTINUE
      IF(PRINTG)LCT=LCT+INL
      IF(PRINTG)WRITE(6,DFMT)(TAB(I),I=1,M)
      IF(PUNCHG)WRITE(7,DFMT)(TAB(I),I=1,M)
   63 CONTINUE
      NEWPG=.TRUE.
      RETURN
      END
```

```
$IBMAP VFI      LIST,REF,DECK,M94,()OK,MFTC
*SYMBLS       E240****        SOVA                        30
*                JOB      SYMBLS MOD FOR $EXIT (NO SYMB..IN TBL)+TAPE OUTP
*      FAP
        ENTRY SYMBLS
        ENTRY BCDCON
        ENTRY HDTAPE
ARSYM1 REM  SYMBOLIC MULTIDIMENSION FN II INPUT SUBROUTINE. AVCO (AR)
        REM  CALL SYMBL S    ( IN )
        REM      WHERE  IN       =  LOCATION TO STORE COL.1 EXIT CODE.
   NSC BOOL 76100
   NSO BOOL 76100
        REM
SYMBLS SXD JRTN1,1
        SXD JRTN2,2
        SXD JRTN4,4
        LFTM
        CLA      TRAFND
        STO      FND
        CLA 3,4                  MOVE =IN= TO SET EXIT CODE
        STZ*     3,4
        STA L(IN)
        CLA      NSIX
        TZE      NOPR
        CALL     .FWRD.(.UN06.,RSTORE)
        TSX      .FFIL.,4
 NOPR   STZ JK
        STZ SSCPTC
        STZ LOC
CRDTAP TSX TAPE,1
        LXA (5),1                CHECK COLUMN 1 FOR NUMERIC FOR EXIT.
        LXA (13),2
        LDQ BCD
 (13)   PXD 13,-
        STP OCTSW1                 RESET =OCTAL= SWITCHES
        STP OCTSW2
 (6)    LGL 6
 (5)    PAX 5,4
        TXL      *+2,4,43        TEST FOR ASTERISK, NO
        TXL      CONTRL,4,44     THIS IS ASTERISK, PROCESS CONTROL
        TRA      NOCNTL
CONTRL XCA
        CAS      NOPRNT          NO DATA PRINT
        TXL      *+2,,-
        TXL      SUPRS,0,0       SUPRESS TSH
        CAS      PRINT           DATA PRINT
        TXL      *+2,0,0
        TXL      NSPRS,0,0       TRIGGER TSIT
        CALL     .FPRN.(LINFRM)          PRINT ON LINE
        AXT      14,2
        CLA      BCD+14,2
        TSX      .FCNV.,4
        TIX      *-2,2,1
        TSX      .FFIL.,4
        TRA      CRDTAP
 SUPRS STZ       NSIX
        TRA *-2
```

```
 NSPRS CLA      N6
       STO      NSIX
       TRA      CRDTAP
NOCNTL TXL L(IN),4,9            YES, BYPASS SELECT, ETC.
       TXH TX1,4,23             27)8 TEST FOR =G=
       TXL TX1,4,22             26)8
       CAL TXL
       TXL L(IN),,-             BY-PASS =RCD/NOP=
 TX1   CAL TXL                  OBTAIN SWITCH RETURN.
       TXH *+2,4,21             25)8 TEST FOR =E=
       TXH ENDCRD,4,20          24)8 YES, OFF MACHINE
       TXH *+3,4,17      21)8   TEST FOR =A=
       TXL *+2,4,15      17)8
       STP ASWTCH                 YES OPEN SWITCH
       TXH *+2,4,24             30)8 TEST FOR =H=
       TXH HEADCD,4,23          27)8 YES, SAVE LINE FOR LATER USE
       TXH *+4,4,38      46)8   TEST FOR =O=, OCTAL NUMBERS
       TXL *+3,4,36      44)8
       STP OCTSW1               SET OCTAL SWITCH
       STP OCTSW2
 L(IN) SXA **,4                 STORE EXIT CODE.
       STP XITSWT               SET EXIT SWITCH
RESET  PXD -,-                  RESET ALL CELLS AND SWITCHES
       STP      EBKSW
       STP GSTRTS
       STZ COUNT
       STP FESW
       STZ SIGN
       STZ COUNTR
       STP ACUMSW
       STP FLPIN1
       STP FLPIN2
RSET1A STZ SUM
BLANK  SYN *
RESET1 TSX NXTCHR,4             GET NEXT CHARACTER FROM LINE
       TXL XITSWT,,**           COL. 73 = GET NEXT RECORD
       PAX -,4                  OCTAL. CHARACTER(S).ACTION
       TXL DIGIT,4,9            00-11)8 0 - 9 DIGIT
       TXL MINUS,4,12           14)8 - REDUNDANT MINUS
       TXL PLUSGN,4,16          20)8 + BY-PASS PLUS SIGN
       TXL ALPHA,4,25           21-31)8 A - I START OF SYMBOL
       TXL POINT,4,27           33)8 . DECIMAL POINT
       TXL MINUS,4,32           40)8 - MINUS SIGN
       TXL ALPHA,4,41           41-51)8 J - R START OF SYMBOL
       TXL STAR,4,43       53)8     $        DOLLAR SIGN, GET NEXT CARD
       TXL BLANK,4,48           60)8 BYPASS BLANK
       TXL      SLASH,4,49      61)8 TEST FOR SLASH
       TXL ALPHA,4,57           62-71)8 S - Z START OF SYMBOL
OPEN(  CLA      FYNC
       STO      TRAPT
       LXA (0),4                ASSUME START OF SUBSCRIPT
       STZ J
       STZ K
OPEN(2 SXD SSCPTC,4
OPEN(1 TSX NXTCHR,4
       TXL SUBDEL-1,,-          ASSUME END OF SUBSCRIPT
       PAX -,4
```

```
        TXL SUBDGT,4,9          (0-9)
        TXL SUBDEL,4,32  40)8 ASSUME =-= OR =)=.SUBSCRIPT END OR BREAK
        TXL OPEN(1,4,48         60)8 WALK OFF BLANK
        LXD SSCPTC,4            ASSUME =,=. MOVE UP SUBSCRIPT
        TXH SUBERR,4,3          ERROR, MORE THAN 3 SUBSCRIPTS
        TRA *+3,4
        CLA J                   MOVE SUBSCRIPTS UP ONE.
        STO K
        CLA SUM
        SUB (1)
        STO J
        STZ SUM
        TXI OPEN(2,4,2
        LXD E)E,4               COL 73 ASSUME =)=
SUBDEL  CLA SUM                 FOUND EITHER =-= OR =)=.
        STO I
        CLA SSCPTC
        TZE SUBD1               TEST FOR MULTISUBSCRIPT
        CLA JK
        STO TRANS+3
        CLA SUM                 CONVERT SUBSCRIPTS.
        SUB (1)
        TNZ *+2
        STO TRANS+3
        ALS 18
        STO TRANS+4
        CLA J
        TNZ *+2
        STA TRANS+3
        ADD TRANS+4
        TZE ADDK-1
        STO TRANS
        STQ TRANS+1
        LDQ TRANS
        MPY TRANS+3
        STQ TRANS
        ARS 1
        LDQ TRANS+1
        ADD TRANS
        ADD (1)
ADDK    ADD K
        STA SUM
        CLA DIFPRM              SET =DIFF= FOR MULTI-SUBSCRIPT.
   E)E  TXL SBSCT),4,28         34)8 OUT IF E)E ENTRY.
        STO DIFF
        CLA I
        STO DELTA
        CAL TXL                 SET UP AREA SET GENERATOR.
        STP ARGNSW
        STP GNSW2
        CAL LOC
        STP RETNSW
SUB1    ADD     SUM
        STZ SUM
        STA STOWRD
        TXH OPEN(,4,28
        TXL RESET,,-
```

```
SUBDGT TSX ACUM,4
       TXL OPEN(1,,-
SUBD1  CLA (1)                  SET =DIFF= TO (1)
       TXH E)E+1,4,28
GNSW2  SYN *
SBSCT) PZE SUBSCT,0,NSC         COMPUTE STORING LOCATION
       STO DIFF
       CLA LOC
       TXL SUB1,,-
SUBSCT CLA I                    COMPUTE COUNT FOR AREA SET GENERATOR
       SUB DELTA
       ADD (1)
       STO DELCNT
       TXL RSET1A,,-
       REM                      FOUND EITHER MINUS (-) (11) OR (8-4)
MINUS  STP     EBKSW
       SSM
       STO SIGN
       TXL DIGIT+1,,-           START NUMBER
       REM                      FOUND . (12-8-4).
POINT  CLA (1)                  CONVERT NUMBER AS FLOATING POINT
       STO COUNTR
       CAL     TXL
       TRA     TXL-1
       REM                      GET-NEXT CHARACTER
NXTCHR SYN *
 (2)   PXD 2,-
       TXL TRA14,2,1            COL.73 ATTEMPT HAS BEEN MADE.
       LGL 6
       TIX TRA24,1,1            TEST FOR EMPTY MQ.
       LDQ BCD+14,2             LOAD UP MQ
       TIX *+1,2,1              DECREASE RECORD-WORD COUNT
       LXA (6),1                RESET CHARACTER COUNT
TRA24  TRA 2,4                  EXIT
       PXD -,-
       STP ACUMSW
RESETA STP FESW                 SET F/E SWITCH
       STZ SIGN                 ZERO OUT CELLS
       STZ COUNTR
       STZ SUM
RESETB STP FLPIN1               SET F/I SWITCHES
       STP FLPIN2
       TXL PLUSGN,,-
       REM                      FOUND DIGIT 0-9, START CONVERTING
DIGIT  TSX ACUM,4               ACCUMULATE DIGIT
PLUSGN SYN *
       TSX NXTCHR,4             GET NEXT CHARACTER
       TXL TGENSW,,-            ASSUME BLANK
 (12)  PAX 12,4
       TXL DIGIT,4,9            0-11)8 ACCUMULATE THIS DIGIT
       TXL PLUSGN,4,16          20)8 BYPASS + SIGN.
       TXL EXPONT,4,21          25)8 ASSUME =E= FOUND, NUMBER HAS EXPONT.
       TXL POINT,4,27           33)8 ASSUME =.= FOUND, NUMBER IS FLTG-PT.
       TXL CLOSE),4,28          34)8 CONVERT DELTA,THEN BACK FOR TOP VALUE
       TXL MINUS,4,32    40)8 SET SIGN MINUS AND CONTINUE PROCESSING.
       TXH (OPEN,4,59           73/74)8 SET UP FOR DELTA ENTRY
EBKSW  PZE     EEK,0,NSC
```

```
        REM                 ASSUME BLANK FOUND. PROCESS NUMBER PER SIGNALS
        REM                       ( SET UP DURING ACCUMULATION.
TGENSW  PZE GENTOP,0,NSC          NORMAL OR GENERATOR MODE SWITCH.
        TSX CONVRT,4
 STOTX  STO     TX
 TRAPT  SYN     *
        TSX     **,4
        TXI *+2,0,0
        CALL    EXIT
        LDQ     TRANMQ
ASWTCH  PZE ACARD,0,NSC =A-CARD= SWITCH AFTER PROCESSING 1ST INTEGER
        NZT     NOFND
STOWRD  STO **
        CAL STOWRD                INCREMENT STORING ADDRESS
        ADD DIFF                  ROW-WISE INCREMENT = 1, J, OR J*K
        STA STOWRD
RETNSW  SVN RESET,0,NSO           GET NEXT NUMBER / CONTINUE IN GENERATIONG.
ARGNSW  PZE ARGEN1,0,NSC
FLPIN1  PZE GENFLO,0,NSC
        CLA TSTART                ** INTEGER GENERATION
        ADD DELTA
        STO TSTART
        TXL DELTST,,-             OUT TO TEST
GENFLO  STQ TRANMQ                ** FLOATING POINT GENERATION
        CLA DELCNT                UP COUNT
        FAD 1PO
        STO DELCNT
        LDQ DELTA                 CREATE NEW ENTRY
        FMP DELCNT
        FAD TSTART
        LDQ TRANMQ
DELTST  STO TRANS                 TEST IF ENTRY IS OK
        SUB TEND
DELSW   CHS                 NOP
        TZE CLATX                 OK, ENTER NUMBER
        TMI CLATX
GENEND  CLA TX
GXITSW  PZE STOTST,0,NSC
        PXD -,-                   RESET SWITCHES
        STP 2ND(SW
        STP TGENSW
GXIT2   CAL TXL
        STP RETNSW
        TXL RESET,,-
(OPEN   CAL TXL
2ND(SW  PZE GENTOP,0,NSC
        STP TGENSW                OPEN SWITCHES
        STP 2ND(SW
        PXD -,-
        STP RETNSW                CLOSE SWITCH
        TSX CONVRT,4
STOTST  STO TSTART
        TXL RESETA-2,,-
CLOSE)  TSX CONVRT,4
        STO DELTA
        STZ DELCNT
        LXD NOP,4
```

```
        TPL  *+2
        LXD  CHS,4
        SXD  DELSW,4
        TXL  RESETA-2,,-
GENTOP  STP  GXITSW              ACCUMULATED FINAL VALUE,STORE
        TSX  CONVRT,4            AND START GENERATING TABLE
        STO  TEND
GSTRTS  PZE  FLPIN1,0,NSC
        CAL  TXL                 OPEN SWITCH FOR GENERATOR
        STP  GSTRTS
        CLA  TSTART
        TXL  STOTX,,-
CLATX   CLA  TRANS
        TXL  STOTX,,-
        REM                 FOUND =E=. SET UP FOR EXPONENT + FLTG-PT NUMBER
EXPONT  CLA  SIGN
 CHS    CLM
        ORA  SUM
        STO  TRANS
        CAL  TXL
        STP      EBKSW
        STP      ACUMSW
TXL     TXL  RESETA,4,21         25)8 IF =E= GO ZERO OUT CELLS
        TXL  RESETB,,-           IF =.= ONLY SET FLP/INT SIGNAL.
EEK     STP      EBKSW
        TRA      PLUSGN
        REM                      NUMBER TO BE CONVERTED TO FLOATING POINT
CONVRT  SYN  *
FLPIN2  PZE  FLP,0,NSC           SWITCH TO SET CONVERSION
OCTSW1  PZE  OCTCNV,0,NSC        OCTAL CONVERSION
        CLA  SIGN                INTEGER CONVERSION
        CLM
        ORA  SUM
        STP  ACUMSW              OPEN =ACUMSW= TO COUNT FRAC. DIGITS
        TRA  1,4
OCTCNV  CLA  SUM
        TMI  OCTCV1
        CLA  SIGN
        CLM
        ORA  SUM
OCTCV1  TRA  1,4
FLP     STQ  TRANMQ              FLOATING POINT CONVERSION
        SXD  CNVX4,4
  FESW  PZE  FPXPNT,0,NSC        OUT TO EXPONENT CONVERSION
        CLA  1P0                 FLOATING POINT (NO EXPOENTN) CONVERSION
        STO  MQ                  SET M/Q VALUE = 1.0
        CLA  SIGN
        CLM
        ORA  SUM
FLP1    ORA  MGNUM
        FAD  MGNUM
        LXA  COUNT,4
        STZ  COUNT
        TXL  LRS35,4,0
        TXL  *+4,4,9
        FDP  10X10
        LLS  35
```

```
        TXI *-4,4,-10
        FDP 10X10+10,4
CNVX4   TXL *+2,,**
 LRS35  LRS 35
        FMP MQ
        LDQ TRANMQ              RESET MQ + XR-4, EXIT
        LXD CNVX4,4
        TRA 1,4
        REM                     EXPONENT CONVERSION
FPXPNT  LXA SUM,4               OBTAIN VALUE OF EXPONENT
        CLA 1PO
        TXL FPXP1,4,0           BY-PASS IF EXPONENT = 0
        LRS 35                  COMPUTE M/Q
        TXL *+3,4,9
        FMP 10X10
        TXI *-4,4,-10
        FMP 10X10+10,4
 FPXP1  STO MQ
        CLA SIGN                TEST IF M/Q SHOULD BE RECIPROCATED
        TPL FPXP2               NO
        CLA 1PO                 YES
        FDP MQ
        STQ MQ
FPXP2   CLA TRANS               OBTAIN SIGNED INTEGER
        TXL FLP1,,-
ARGEN1  CLA DELCNT              ** AREA SET GENERATOR
        SUB (1)                 TEST IF DONE
        TZE GENXIT              YES
        STO DELCNT              NO
        CLA TX
        TXL STOWRD,,-
GENXIT  PXD -,-                 DONE ENTERING
        STP ARGNSW              RETURN SWITCHES.
        STP GNSW2
        TXL GXIT2,,-
STAR    SYN *
XITSWT  PZE CRDTAP,0,NSC        YES
        LXD JRTN1,1
        LXD JRTN2,2
        EFTM
        CLA     CASE
        ADD     (1)
        STO     CASE
        ZET     NOFND
        CALL    EXIT
        TRA     GOBCK         DO THIS CASE
 GOBCK  LXD JRTN4,4
        TRA     1,4         SWITCH OFF  DON'T STOP
 SKFRM  BCI     3,(6H CASE I5////)
  CASE  BSS 1
   (1   HTR 0,0,1
        REM                     FOUND SYMBOL
ALPHA   CLA     FYNC
        STO     TRAPT
        PXA     -,4
        LXA     (5),4
 ALPHP  ALS     6
```

```
        SLW SUM
        SXD ALFAX4,4
        TSX NXTCHR,4
ALFAX4  TXL ALFA3,,-
        PAX ,4
        SXD SYM,4
        TXH *+2,4,60            74)8 TEST FOR =(-
        TXH ALFA3+1,4,59        73)8 YES, OBTAIN SYMBOL AS IS.
        ORA SUM                 CREATE SYMBOL TO DATE.
        TXH *+2,4,48            60)8
        TXH ALFA2,4,47          57)8
        LXD ALFAX4,4
        TIX     ALPHP,4,1
        SLW SYMBOL              STORE SYMBOL, 6 CHARACTERS
ALFA6   TSX NXTCHR,4            WALK OFF EXTRA LETTERS TO =(= OR BLANK
        TXL ALFA5,,-            END OF CARD ALSO ENDS CHARACTER NAME
        PAX ,4
        TXH ALFA5,4,59
        TXH ALFA6,4,48          NON-BLANK AND NON-=(= BY-PASS.
        TXL ALFA6,4,47
ALFA5   SXD SYM,4               SET =SYM= TO LAST CHARACTER GOTTEN
        TXL *+2,,-
ALFA1   SLW SYMBOL
        LXA M,4                 SET-UP FOR TABLE-LOOK-UP ON SYMBOL.
 B)TBL  CAL **,4                B)+M OBTAIN ENTRY
        ANA MASK1               TEST FOR BCD NAME
        TZE B)TBL2              NO BCD WORD
B)TBL1  CLA **,4                B)+M
        CAS SYMBOL
        TXL *+2,,-
        TXL FOUND,,-
        TIX B)TBL,4,2
B)TBL2  TIX B)TBL,4,1
        CLA CMMON               CHECK FOR COMMON LOAD
        CAS SYMBOL
        TRA ERRLN
        TRA COMN
 ERRLN  STO     NOFND
        SXD JUNIR4,4            START OF ERROR CARD
        STQ     TRANMQ
        CALL    .FPRN.(FORMAT)
        TSX     .FFIL.,4
        CALL    .FWRD.(.UN06.,FORMAT)
        TSX     .FFIL.,4
        LXD     JUNIR4,4
        LDQ     TRANMQ
 FND    NOP     FOUND   NOP FOR BCDCON, TRA FOR SYMBLS
        CALL    .FPRN.(PRODER)
        TSX     .FFIL.,4
        CALL    EXIT
ALFA3   SXD SYM,4
        CAL SUM
        ORA 1BLANK
ALFA2   LXD ALFAX4,4
        TNX ALFA1,4,1
        ALS 6
        ORA 1BLANK
```

```
SYM     TXL ALFA2+1,,**
FOUND   CAL **,4                B)+M+1
        STA STOWRD
        SUB     (1)
        STA LOC
 FND1   CAL **,4          B)+M+2 OBTAIN THIRD WORD IN CASE OF MULTISUBSCT.
        TZE SINGLE              END OF TABLE, SPECIAL CASE
        ANA MASK1               TEST FOR MULTI-SUBSCRIPT
        TNZ SINGLE              NO
 FND2   CAL **,4                B)+M+2
        STO JK
        ARS 18
        TXL SINGLE+2,,-
SINGLE  CLA (1)
        STZ JK
        STO DIFPRM
        STO DIFF
        LXD SYM,4               TEST IF LAST CHARACTER WAS =(=
        TXL RESET,4,59          73)8 NO.
        STZ SUM                 YES, GO PROCESS SUBSCRIPT.
        TXL OPEN(,,-
TAPE    CALL     .FRDD.(.UN05.,INFRM)
        CALL     .FSLI.(BCD,M.)
        TSX      .FRTN.,4
        CLA      NOFND
        TNZ      NOUT2
        CAL NSIX
        TZE      NOUT2
        CALL     .FWRD.(.UN06.,LINFRM)
        CALL     .FSLO.(BCD,M.)
        TSX      .FFIL.,4
NOUT2   TRA      1,1
ACUMSW  SYN *                   *** ACCUMULATE DIGIT
  ACUM  PZE ACUM1,0,NSC
        TZE 1,4                 BY-PASS ALL LEADING ZEROS.
        STO TRANS+3             PRESERVE DIGIT.
        CAL TXL                 OPEN SWITCH FOR RES OF DIGITS
        STP ACUMSW
        TXL DIGNTR,,-
ACUM1   STO TRANS+3
OCTSW2  PZE OCTSUM,0,NSC
        CLA HIGHTS              TEST IF OK TO ENTER NEXT DIGIT
        SUB SUM
        TMI 1,4
DIGNTR  CLA SUM
        ALS 1
        STO SUM
        ALS 2
        ADD SUM
        ADD TRANS+3
        STO SUM
FRCNT   CLA COUNT
        ADD COUNTR
        STO COUNT
TRA14   TRA 1,4
OCTSUM  CAL SUM
        ALS 3
```

```
        ORA TRANS+3
        SLW SUM
        TRA 1,4
HEADCD  REM                     FOUND =H=. MOVE LINE INTO BUFFER
HEADCD  AXT 14,4
        CLA (1)
        LGL 30
        SLW BCD
        CLA BCD+14,4            MOVE LINE OF TEXT
        STO LINE+14,4
        TIX *-2,4,1
        TXL CRDTAP,,**
ACARD   STP ASWTCH              ACCUM.CONTAINS WORD COUNT IN DECREMENT.
        ALS     18
        STD ACNTR                       RESET =ASWTCH=
        ARS 18                    OBTAIN STORING LOCATION AND SET FOR
        ACL STOWRD                   NEXT AVAILABLE WORD AFTER ALPHA INFO.
        STA STOWRD
        STA ACRD2
        NOP
ACRD1   TSX TAPE,1             GET NEXT INPUT RECORD
        AXT     14,1
        LXD ACNTR,2
        CAL BCD+14,1
ACRD2   SLW **,2
        TNX CRDTAP,2,1          EXIT FOR NEXT RECORD UNDER NORMAL CONTROL
        TIX ACRD2-1,1,1                 ASSUME REST OF CARD SKIPPED.
        SXD ACNTR,2
ACNTR   TXL ACRD1-1,,**
ENDCRD  CALL    EXIT
        REM
        REM TO CONVERT BCD LIST TO TABLE USABLE FOR INPUT PROGRAM
        REM
BCDCON  CLA 1,4                 TEST IF ENTRY HAS BEEN MADE FROM THERE
        STO 0,4                 UP  RETURN IMMEDIATELY
        SXD JRTN1,1
        SXD JRTN2,2
        SXD JRTN4,4
        CLA 3,4
 TRIG   PZE BCD1,0,NSC          BYPASS EXCEPT FIRST TIME. SET UP ENTRY
        STA B)                  STORE FOR =M= COMPUTATION.
        STA NTRNAM              STORE FOR TABLE ENTRY.
        ADD =0100000000001
        STP TRIG                CHANGE SWITCH
        STA NTRLOC
        ADD (1)
        STA NTRJK
        SUB (2A)
 BCD1   ADD (95)
        STA BCD2
        STZ LOC
        LXA (95),1              MOVE TEXT TO TRANSIENT AREA (SET XR1=95)
        LXA (95),2              MOVE TEXT TO TRANSIENT AREA (SET XR2=95)
 BCD2   CLA **,2                B)+95
        STO BCD+95,2
        TIX BCD2,2,1
        TIX *+1,1,1             DROP XR 1 FOR 2ND WORD OF =BCD=
```

```
        LDQ BCD                 SET-UP FOR CHARACTER PICK-UP
        LXA (6),2
        CLA (2)                 SET NAME COUNT = 2 FOR STARTING VALUE
RETURN  STA NAMCNT
        TIX *+1,4,1             SET XR4 FOR NEXT NAME
        SXD BCDXR4,4
        STZ J
        STZ K
        STZ SUM
        STZ NAME
        TSX BCDCHR,4
 (95)   PAX 95,4
        TXH ALLDON,4,62         76)8 DONE WITH LIST. EXIT
        TXH (95)-1,4,58         72)8 WALK OFF =,= + =(=
        TXH *+2,4,48            60)8
        TXH (95)-1,4,47         57)8 BY-PASS BLANKS
        LXA (6),4               ASSUME CHARACTER SO START ACCUMULATING
 NAME1  ORA NAME                ( NAME
        TNX NM4,4,1             TEST FOR 6-CHARACTER ENTRY.
        ALS 6
        SLW NAME
        SXD CHACNT,4
 NM1    TSX BCDCHR,4            GET NEXT CHARACTER
        STA CHAR                PRESERVE CHARACTER FOR LATER TESTING
 CHAR   PAX **,4
        TXH NM2,4,58            72)8 EITHER =,= OR =(= OR =77)8=
        TXH *+2,4,48            60)8 CHARACTER = S - Z =
        TXH NM1,4,47            57)8 WALK OF BLANKS.
        LXD CHACNT,4            ALL OTHER CHARACTERS
CHACNT  TXL NAME1,,**
  NM2   LXD CHACNT,4            OR IN BLANKS TO FILL UP =NAME=
        CAL NAME
        TXL *+2,,-
        ALS 6
        ORA 1BLANK
        TIX *-2,4,1
 NM4    SLW NAME
        LXA CHAR,4              RE-LOAD XR WITH ENTRY CHARACTER.
  NM3   TXH SETNAM,4,62         76)8 END OF RUN= CHARACTER =77)8=
        TXH SSCPT,4,59          73)8 CHARACTER =(=, START SUBSCRIPT
        TXH SETNAM,4,58         72)8 CHARACTER =,=, STORE ENTRY.
        TSX BCDCHR,4            GET NEXT CHARACTER
  M     PAX **,4
        TXL NM3,,-
SETNAM  LXD BCDXR4,4            STORE =NAME=, =LOC,,I=, =K,,J= BACK IN
        CAL 3,4                 OBTAIN =LOC.NAME= ( TABLE.
        STA LOC
        ARS 15                 TEST FOR 0 PREFIX AND DECREMENT
        TNZ NAMERR              ERROR, RUN OUT OF NAMES
        LXA SUM,4               CREATE =LOC,,I= (FOR NO SUBSCRIPT, I=0 )
        SXD LOC,4
        LXA J,4                 BY-PASS FOR SINGLE DIMENSION
        TXL NTRNAM-1,4,0        1 DIMENSIONAL
        CLA K
        TZE SET2                2 DIMENSIONAL
        STQ TRANS               3 DIMENSIONAL
        LDQ K
```

```
        MPY J
        LLS 53
        LDQ TRANS
        ADD K
        TXL *+3,,-
 SET2   PXD ,4
        ADD (1)
        SLW K
        CAL NAME                MOVE =NAM= BACK INTO NEW TABLE
NTRNAM  SLW **                  B)
        CAL LOC                 MOVE =LOC,,I= BACK INTO TABLE
NTRLOC  SLW **                  B)+1
        CAL *-1
        TXL NTR1,4,0            TEST FOR THIRD ENTRY ( XR4 = J STILL)
        CAL K                   ENTER THIRD WORD
NTRJK   SLW **                  B)+2
        CAL *-1
 NTR1   ADD (1)
        STA NTRNAM
        ADD (1)
        STA NTRLOC
        ADD (1)
        STA NTRJK
        CAL NAMCNT              UP =NAME COUNT= RE-ENTRY EXIT.
        ADD (1)
        LXD BCDXR4,4            OBTAIN XR4 FOR MODIFICATION.
BCDXR4  TXL RETURN,,**
SSCPT   LXA (0),4               FOUND A SUBSCRIPT,
        STZ J                   ZERO OUT CELLS
        STZ K
  SS1   SXD SSCPTC,4
  SS2   TSX BCDCHR,4            GET CHARACTER
  (0)   PAX 0,4
        TXL SSDIG,4,9           (0-9) TEST FOR DIGIT
        TXL SETNAM,4,28         34)8 ASSUME =)= WHICH ENDS SUBSCRIPT.
        TXL SS2,4,48            60)8 ASSUME BLANK, WALK OFF.
        LXD SSCPTC,4            ASSUME =,= MOVE AS SUBSCRIPT + STORE
        TXH SUBERR,4,3
        TRA *+3,4               UP SUBSCRIPTS
        CLA J
        STO K
        CLA SUM
        STO J
        STZ SUM
        TXI SSI,4,2
SUBERR  CLA LSSCER              ERROR, MORE THAN 3 SUBSCRIPTS GIVEN
        ARS 18
        STA FORM
        STA     FORM+6
LSSCER  TXL ERRLN,,SSCERR
NAMERR  CLA NMERRL
NMERRL  TXL SUBERR+1,,FNMERR
SSDIG   TSX ACUM,4
        TXL SS2,,-
ALLDON  LXD BCDXR4,4            ALL DONE PROCESSING. CHECK IF NEXT WORD
        CLA 3,4                 IN SET-UP IS IN =NTR=
        ARS 15                 TEST FOR 0 PREFIX AND DECREMENT
```

```
        TNZ DN1                 OK
        CLA MERRL
MERRL   TXL SUBERR+1,,LISTER
 DN1    CAL NTRNAM              MOVE =B)+M= INTO TABLE LOOK-1+
        STA B)TBL
        STA B)TBL1
        STA *+1
        STZ **                  SET END OPERATION
        ADD (1)
        STA FOUND
        ADD (1)
        STA FND1
        STA FND2
        SUB (2A)
        SUB B)                  COMPUTE =M= FOR TABLE LOOK-UP TEST
        STA M
        LXD JRTN1,1
        LXD JRTN2,2
        LXD JRTN4,4
        CAL NAMCNT
        STO 3,4
        TRA 1,4
NAMCNT  PZE
BCDCHR  PXD -,-
        LGL 6
        TIX TRA14,2,1
        LXA (6),2
        LDQ BCD+95,1
        TXI TRA14,1,-1
  NOP   NOP
FORMAT  BCI 3,(28H  NO SYMBOL  =
SYMBOL  BCI 1,
        BCI 2,=LISTED    )
LISTER  BCI 7,(34H1MORE ENTRIES IN LIST THAN IN TEXT )
FNMERR  BCI 6,(27H1LIST OF ENTRIES TOO SHORT  )
SSCERR  BCI 7,(33H1MORE THAN 3 SUBSCRIPTS ARE GIVEN )
LINFRM  BCI     2,(1H 14A6)
RSTORE  BCI 4,(17H1INPUT CARDS READ)
INFRM   BCI 1,(14A6)
HDFORM  BCI     1,(14A6)
 NSIX   HTR 6
N5      HTR 5
  N6    HTR 6
  M.    HTR 14
1BLANK  OCT 60
MGNUM   OCT 233000000000
 (1)    DEC 1
 (2A)   DEC 2
HIGHTS  DEC 13421771            ENTRY TEST ONE MORE DIGIT CAN STILL GO
MASK1   OCT -300000000000
  MASK  OCT -375777000000
 B)     SLW ** B)
 10X10  DEC 1.E10,1.E9,1.E8,1.E7,1.E6,1.E5,1.E4,1.E3,1.E2,1.E1
   1PO  DEC 1.0
 DONE   SYN *
 COUNT  BSS 1
 LOC    BSS 1
```

```
SUM     BSS 1
SIGN    BSS 1
COUNTR  BSS 1
TRANMQ  BSS 1
 MQ     BSS 1
TSTART  BSS 1
TX      BSS 1
TEND    BSS 1
DELTA   BSS 1
DELCNT  BSS 1
DIFPRM  BSS 1
SSCPTC  BSS 1
 DIFF   BSS 1
 I      BSS 1
 NAME   BSS 1
 J      BSS 1
 K      BSS 1
 JK     BSS 1
 TRANS  BSS 7
 BCD    BSS 100
 JRTN1  BSS 1
JRTN2   BSS 1
 JRTN4  BSS 1
JUNIR4  BSS 1
        REM                     HEAD TAPE (I),ALWAYS TO TAPE, IF SENSE
HDTAPE  SXD JRTN1,1
        SXD JRTN4,4
        NZT NSIX
        TRA     HRTRN
        CALL    .FWRD.(.UN06.,HDFORM)
        AXT 14,1
        CLA     LINE+14,1
        TSX     .FCNV.,4
        TIX *-2,1,1
        TSX     .FFIL.,4
 HRTRN  LXD     JRTN1,1
        LXD     JRTN4,4
        TRA     1,4
LINE    BSS 14
SLASH   STZ     BASE
        AXT     36,4
        TRA     SL2
SL1     CAL     SLCH
        CAS     SLBL
        TRA     *+2
        TRA     SL3
SL2     SXA.    *+3,4
        TSX     NXTCHR,4
        TSX     $EXIT,4
        AXT     **,4
        SLW     SLCH
SL3     ALS     30
        ARS     36,4
        ORS     BASE
        TIX     SL1,4,6
  SLX   CLA FYNC
        STO TRAPT
```

```
        CLA     BASE
        STO     TX
        TRA ASWTCH
HOLCOD  PZE     30,0,6
OCTCOD  PZE     33,0,3
 BASE   BSS     1
SLBL    BCI     1,00000
SLCH    BSS     1
 COMN   CAL CMMON+1              SET UP STORE FOR COMMON
        STA STOWRD
        SUB     (1)
        STA LOC
        TRA SINGLE
 CMMON  BCI 1,COMMON
        OCT 000000077461
 NOFND  DEC     0
 PRINT  BCI     1,PRINTO
NOPRNT  BCI     1,NOPRIO
 FYNC   TRA ASWTCH
PRODER  BCI     9,(49H IF PRODUCTION, RELOAD JOB OFF LINE AND TRY AGAIN)
TRAFND  TRA     FOUND
XMASK1  OCT 777777700000
 FORM   SYN     ERRLN+6
   E.   BSS 1
        END
```

```
$IBMAP MARK     1700,LIST,REF,DECK,M90
*      MARK     RUNGE-KUTTA,ADAMS-MOULTON INTEGRATOR PACKAGE
*      CALLING SEQUENCE
*      CALL     MARK
*      PZE      HBANK,P,EOS
*      PZE      DER1,PHI,DER2
*      ERROR RETURN
*      PZE,MZE BJ,,YJ
*      PZE      ZJ
*      PZE      0
*      MARK,HC,NI
       ENTRY    MARK
       ENTRY    HC
       ENTRY    NI
       ENTRY    TGLO
       ENTRY    Y
       ENTRY    YDOT
       ENTRY    Y(2)
       ENTRY    YO
       ENTRY    YO(2)
       ENTRY    EUBAR
       ENTRY    ELBAR
       ENTRY    HMAXT
       ENTRY    HMINT
       ENTRY    YCLOW
       ENTRY    RGERR
 OMAR  SET      50
  MARK SXA      I4,4
 RSTRT LXA      I4,4
       SXA      AEOS,4
       LXA      I4,1
       CLA      4,1
       TZE      *+6
       PDX      0,2
       TXL      *+3,2,0
       CLA      *
       STT      4,1
       TXI      *-6,1,-2
       CLA      1,4
       STT      P
       PAX      *,1
       TXI      *+1,1,-3
       SXA      L(M),1
       TXI      *+1,1,5
       SXA      L(T1),1
       TXI      *+1,1,1
       SXA      L(T2),1
       AXT      0,1                STORE USER H BANK
       CLA*     L(M)               IN MARK COMMUNC AREA
       STO      M,1
       TXI      *+1,1,-1
       TXH      *-3,1,-4
       CLA*     L(M)
       STA      N
       ARS      18
       STA      (N)
       CLA*     L(T1)
```

```
STO     T1
CLA*    L(T2)
STO     T2
LXA     I4,4
CLA     2,4
STA     DER1
LRS     18
STA     DER2
PXD
XCA                         PHI TEST
ARS     32
STO     PHI
PAX     *,1
TRA     *+5,1
ARS     2                   PHI=4 - AM WITH AEC
TRA     *+3
COM                         PHI=2 - NO AM
TRA     *+6
STO     E                   PHI=0 - AM NO AEC
CLA     LSTRT
STA     TRIGO
CLA*    L(T1)
LDQ*    L(T2)
STO     TRG2
STQ     TRG2+1
CLA     H
STO     HC
STZ     J                   INITIALIZE J COUNT
TSX     ADDR,4
LXA     E,1
TXI     *+1,1,3
PXA     0,1                 FORM BSS LENGTH
ADD     M
XCA
MPY     N
STQ     TEMP
XCA
ADD     TEMP
STO     TEMP
LDQ     N
MPY     E
XCA
ADD     TEMP
NZT     E
TRA     *+3
ADD     N
ADD     N
PAX     0,2
LAC     YDOT,1              CLEAR USER BSS AREA
STZ     0,1
TXI     *+1,1,-1
TIX     *-2,2,1
AXT     -1,1                INITIALIZE NH AND ND
STZ     NH                  IN MARK AND USER BANK
STZ     ND
STZ     HD
CLA*    L(M)
```

```
       ALS     18
       STD*    L(M)
       TXI     *+1,1,-1
       CLA*    L(M)
       ALS     18
       STD*    L(M)
       NZT     E
       TRA     *+5
       CLA     RGFRK
       STO     GT2
       STZ     A              STET FLAG WORD =0 FOR 2.0X (AEC)
       TRA     *+3
       CLA     ADAMS+2
       STO     GT2
       XEC     DER1
       CLA     T1
       SSP
       LDQ     H
       TLQ     *+2
       XCA                    MAX(H,T) IN AC
       SUB     HR09           DIVIDE BY 2**26
       ZET     P
       SUB     HR09           DIVIDE AGAIN IF D.P. TIME
       STO     DELU           DELTA SUB U
       TSX     ABTB,4              INITIALIZE INTERRUPT S.R.
       TRA     *+2
       TRA     RKC
       LXA     I4,4
       TRA     3,4                 ERROR RETURN
    HC PZE
    NI PZE     0
     J PZE
  TMIN PZE
 TMIN2 PZE
  BMIN TSX     **,4
   TGO PZE
  TGO2 PZE
  DELU PZE
    TL PZE
   TL2 PZE
    TR PZE
   TR2 PZE
  TEMP BSS     10
 TRIGO PZE
       PZE     TRG2
  AEOS NOP
  ASET PZE     SET,4
 AFLAG PZE     FLAG,4
  TRG2 BSS     2
     Y PZE     **,1
  YDOT PZE     **,1
  Y(2) PZE     **,1
    YO PZE     **,1
 YO(2) PZE     **,1
   MP1 PZE
    I4 PZE
     M PZE
```

```
   NH PZE
   ND PZE
    H PZE
    N PZE
  (N) PZE
   T1 PZE
   T2 PZE
 L(M) PZE      **,1
L(T1) PZE
L(T2) PZE
    P PZE
  PHI PZE
 DER2 TSX      **,4
 DER1 TSX      **,4
    E PZE
   HD PZE
  HDC PZE
LSTRT PZE      START
 DELO DEC      0
GSIGM DEC      0
  ERC PZE
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
 DELX PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
 DELY PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
      PZE      **,1
 DELZ PZE      **,1
```

```
   YN PZE     **,1
  YN2 PZE     **,1
EUBAR PZE                        UPPER LIMIT OF E(N+1)
ELBAR PZE                        LOINER LIMIT OF E(N+1)
HMAXT PZE                        MAXIMUM DELTA T
HMINT PZE                        MINIMUM DELTA T
YCLOW PZE                        LOWER BOUND OF Y(N+1)C
RGFRK TRA     RGINT
 ADDR LXA     (N),1               SET TABLE ADDRESSES
      SXD     *+13,1
      LXA     N,1
      SXD     RGAD,1
      SXD     *+13,1
      SXD     *+17,1
      LXA     M,1
      TXI     *+1,1,1
      SXA     MP1,1
      TXI     *+1,1,1
      SXA     *+9,1
      LXA     L(M),1
      TXI     *+1,1,7
      AXT     5,2
      TXI     *+1,1,*
      PXA     *,1
      STA     YO(2)+1,2
      TXI     *+1,1,*             N IN DECRE
      TIX     *-3,2,1
      AXT     *,2                 M+2 IN ADD
      PXA     *,1
      STA     DELX+1,2
      TXI     *+1,1,*             N IN DECRE
      TIX     *-3,2,1
      LXA     N,2
      SXD     *+5,2
      LXA     MP1,2
      TXI     *+1,2,1
      PXA     *,1
      STA     DELY+1,2
      TXI     *+1,1,*             N IN DECRE
      TIX     *-3,2,1
      NZT     E
      TRA     RGAD+2
      LXA     N,2
      SXD     *+5,2
      LXA     MP1,2
      TXI     *+1,2,1
      PXA     *,1
      STA     DELZ+1,2
      TXI     *+1,1,**
      TIX     *-3,2,1
      AXT     2,2
      PXA     *,1
      STA     YN2+1,2
 RGAD TXI     *+1,1,*             N IN DECRE
      TIX     *-3,2,1
      TRA     1,4
 ABTB SXA     HA01,1
```

```
      SXA     HA02,2
      SXA     HA03,4
      REM     ASET AND AFLAG SHOULD BE ASSEMBLED AS
      REM     ASET PZE SET,4   AND AFLAG PZE FLAG,4
      CLA     HA12
      STD     ASET            ASET NOW CONTAINS NOP
      STD     AFLAG           AFLAG NOW CONTAINS NOP
      CLA     TRIGO
      STA     BMIN
      CLA     TRIGO+1
      SUB     HA11
      STA     *+1
      CLA     **
      STO     TMIN2
      CLA*    TRIGO+1
      STO     TMIN
      LXA     I4,1                 FIND MARK
HA04  CLA     4,1
      TZE     HA10                 END OF TRIGGERS
      TPL     *+2
      TXI     HA04,1,-2       SKIP NEGATIVE TRIGGERS
HA11  PDX     1,2                  SET DEP. VAR. FLAG
      TXL     *+5,2,0               IF INDEPENDENT VARIABLE, JUMP
      CLA     HA13
      STD     ASET            ASET NOW CONTAINS TSX
      STD     AFLAG           AFLAG NOW CONTAINS TSX
      TXI     HA04,1,-2
      CLA     5,1
      STA     HA05
      SUB     HA11
      STA     HA06
HA05  CLA     **                   T1(I)
      NZT     P
      TRA     *+4
      FAD*    HA06                 T2(I)
      STQ*    HA06
      STO*    HA05                 T1(I)
      CAS     TMIN
      TXI     HA04,1,-2       T1(I)-TMIN=+,CONTINUE SEARCH
      TRA     HA07                 T1(I)-TMIN=0,CHECK TMIN2 IF P=1
      STO     TMIN                 T1(I)-TMIN=-,REPLACE TMIN AND TMIN2
      CLA     4,1
      STA     BMIN
      NZT     P
      TXI     HA04,1,-2            P=0, CONTINUE SEARCH
HA06  CLA     **                   T2 (I)
      STO     TMIN2
      TXI     HA04,1,-2            NEW TMIN TMIN2, CONTINUE SEARCH
HA07  NZT     P
      TXI     HA04,1,-2            P=0,CONTINUE SEARCH
      CLA*    HA06                 T2(I)
      CAS     TMIN2
      NOP
      TXI     HA04,1,-2            T2(I)-TMIN2=+,CONTINUE SEARCH
      STO     TMIN2                T2(I)-TMIN2=-, REPLACE TMIN2
      CLA     4,1
      STA     BMIN
```

```
      TXI     HA04,1,-2          CONTINUE SEARCH
HA10  LXA     I4,1               FIND MARK
      CLA     2,1
      STA     DER1
      STZ     TEMP
      STT     TEMP               PHI IN T(TEMP)
      ARS     18
      STA     DER2
      CLA     1,1
      STD     HA08
      ARS     18
      STO TEMP+1
      TNZ     *+4                GO COMPARE EOS WITH A (AEOS)
      CLA     HA12
      STD     AEOS               SET C(AEOS)=NOP
      TRA     HA09               CONTINUE
      LAC     AEOS,2
HA08  TXI     *+1,2,**
      TXL     HA09,2,0           EOS=A(AEOS)
      CLA     HA13
      STO     AEOS               C(AEOS)=TSX **,4
      CLA TEMP+1
      STA     AEOS               A(AEOS)=NEW EOS
      XEC     AEOS
HA09  CLA     TEMP
      ARS     15
      STO     TEMP               PHI IN A(TEMP)
      SUB     PHI
      TZE     HA16                PHI UNCHANGED
      TRA     RSTRT              PHI AND RESTART
HA16  AXT     -4,1
      CLA*    L(M)           PICK UP N,,(N)
      ARS     18
      CAS     (N)
      TRA     RSTRT              (N) BIGGER, RESTART
      TRA     *+3            (N) UNCHANGED, CONTINUE
      STO     (N)
      TSX     ADDR,4         FIX DELX ADDRESSES FOR NEW (N)
      AXT     -3,1
      CLA*    L(M)
      STO     H
      AXT     -1,1               MODIFY NH AND ND BY
      CLA*    L(M)               AMOUNT OF USER CHANGE
      ADD     NH
      STA     NH
      ALS     18
      STO*    L(M)
      TXI     *+1,1,-1
      CLA*    L(M)
      ADD     ND
      STA     ND
      ALS     18
      STO*    L(M)
HA03  AXT     **,4
HA02  AXT     **,2
HA01  AXT     **,1
      CLA     TMIN
```

```
      FSB    T1                 IF TMIN-T1=- AND P=0
      NZT    P                  THEN CHECK TO SEE IF TMIN
      FAD    DELU               IS WITHIN DELU OF T1
      TMI    1,4
      TRA    2,4
HA12  NOP
HA13  TSX    **,4
HA14  OCT    2
 RKC  CLA    T1
      SSP
      LDQ    H
      TLQ    *+2
      XCA                       MAX (H,T) IN AC
      SUB    HR09               DIVIDE BY 2**26
      ZET    P
      SUB    HR13               DIVIDE AGAIN IF D.P. TIME
      STO    DELU               DELTA
HR04  CLA    TMIN
      FSB    T1
      STO    TEMP
      XCA
      FSB    T2
      FAD    TMIN2
      FAD    TEMP
      SLW    HR03
      FAD    DELU
      TPL    *+3
HR11  LXA    I4,4
      TRA    3,4                TIME ERROR
      CLA    HR03
      LDQ    DELU
      TLQ    HR02               TRANSFER IF T NOT WITHIN
      XEC    BMIN               DELTA OF TMIN
      TRA    *+1
      TSX    ABTB,4             GET NEW TMIN
      TRA    HR11
      TRA    HR04
HR02  CLA    HR03               TMIN-T
      LDQ    H
      STQ    HC
      TLQ    *+2
      STO    HC                 HC=MIN(H,TMIN-T)
      XEC    ASET
      TSX    KUTTA,4            INTEGRATE TO T+DT
      CLA    T1
      LDQ    T2
      STO    TGO
      STQ    TGO2
      XEC    AEOS
      XEC    AFLAG
      TRA    RKC                NORMAL RETURN
HR12  TSX    SRCH,4             FLAG RETURN
      TRA    HR05               CON
      STO    HC                 BAK
HR01  TSX    KUTTA,4
      XEC    AEOS
      TRA    HR12
```

```
 HR05 LXA     I4,1
 HR07 CLA     4,1
      TZE     RKC                 END OF TRIGGERS
      ANA     HR10
      TZE     HR08
      SXA     RAY1,1
      CLA     4,1
      STA     *+1
      TSX     **,4                EXECUTE BMIN
      TRA     *+1
      LXA     RAY1,1
      TSX     ABTB,4              UPDATE TMIN
      TRA     HR11
 HR08 TXI     HR07,1,-2
 HR09 OCT     32000000000
 HR10 OCT     400000
 HR13 OCT     32000000000
 HR03 OCT     0
RAY1  PZE
  AMC NZT     P                   ADAMS-MOULTON CONTROL
      TRA     *+5
      TSX     DSUB,4
      PZE     T1,0,TMIN
      SSP
      TRA     *+4
      CLA     T1
      FSB     TMIN
      SSP
      CAS     DELU
      TRA     GT1
      NOP
      XEC     BMIN
      TRA     *+2
      TRA     RSTRT               RESTART
  GTO TSX     ABTB,4              ENTRY FROM START
      TRA     GT10+1              ERROR RETURN
      TRA     AMC
  GT1 XEC     ASET                DO SET ROUTINE
      NZT     P
      TRA     *+5
      TSX     DSUB,4
      PZE     T1,0,TGO
      SSP
      TRA     *+4
      CLA     T1
      FSB     TGO
      SSP
      CAS     DELU
      TRA     GT3
      NOP
  GT2 NOP                         AEC FORK
      NZT     NH
      TRA     *+8
      TSX     X.50,4              HALVE INTERVAL
      AXT     -1,1
      LXA     NH,2
      TXI     *+1,2,-1
```

```
       SXA     NH,2
       PXD     0,2
       STD*    L(M)
       NZT     ND
       TRA     *+12
       ZET     HD
       TRA     *+10
       TSX     X2.0,4              DOUBLE INTERVAL
       AXT     -2,1
       LXA     ND,2
       TXI     *+1,2,-1
       SXA     ND,2
       PXD     0,2
       STD*    L(M)
       TSX     ABTB,4              FIND NEW TMIN
       TRA     GT10+1              ERROR RETURN
RGINT  TSX     ADAMS,4             INTEGRATE
       CLA     T1
       LDQ     T2
       STO     TGO
       STQ     TGO2
       SSP                         FORM DELU
       LDQ     HC
       TLQ     *+2
       XCA                         HC IN AC
       ZET     P
       SUB     GT10
       SUB     GT10
       STO     DELU
  GT3  CLA     DELU
       CHS
       FAD     TGO2
       FAD     TGO
       STO     TEMP
       STQ     TEMP+1
       CLA     TEMP
       CAS     TMIN
       TRA     GT5
       TRA     *+2
       TRA     GT4
       CLA     TEMP+1
       CAS     TMIN2
       NOP
       TRA     GT5
  GT4  CLA     TGO
       STO     T1
       STO*    L(T1)
       CLA     TGO2
       STO     T2
       STO*    L(T2)
       TSX     PUTB,4
       XEC     AEOS
       TRA     GT6
       TRA     RSTRT               RESTART
  GT5  CLA     TMIN
       STO     T1
       STO*    L(T1)
```

```
       CLA     TMIN2
       STO     T2
       STO*    L(T2)
       TSX     INTRP,4
  GT6  XEC     AFLAG              FLAG SUBROUTINE
       TRA     AMC
       TSX     SRCH,4             FLAG RETURN
       TRA     GT8
       CLA     T2
       FAD     HIC
       FAD     T1
       STO     T1
       STO*    L(T1)
       STQ     T2
       STQ*    L(T2)
       TSX     INTRP,4
       TRA     GT6+2
  GT8  STZ     STAR
       LXA     I4,4
       CLA     4,4
       TZE     GT9
       TMI     GT11
       ARS     17
       LBT
 GT11  TXI     GT8+2,4,-2
       SXA     *+6,4
       CLA     4,4
       STA     *+1
       TSX     *,4                EXECUTE INTERRUPTION
       TRA     *+2
       TRA     GT12
 GT13  AXT     *,4
       CLA     *-2            CLEAR 5 FROM TAG
       STT     4,4
       TRA     GT11
  GT9  TSX     ABTB,4
       TRA     GT10+1             ERROR RETURN
       XEC     ASET               SET SUBROUTINE
       ZET     STAR
       TRA     RSTRT
       TRA     GT3
 GT12  CLA     *
       STO     STAR
       TRA     GT13
 STAR  PZE                        FLAG ROR RESTART
 GT10  OCT     32000000000
       LXA     I4,4
       TRA     3,4                ERROR RETURN
KUTTA  SXA     HK08,1
       SXA     HK09,2
       SXA     HK10,4
       CLA     HC
       FDH     HK12
       STQ     TEMP+1             DT/2
       STZ     TEMP
       LXA     (N),1
       CLA*    Y
```

```
      STO*    YO                  SAVE INITIAL VALUES OF Y(I)
      TIX     *-2,1,1
      AXT     4,4
HK01  LXA     (N),1
      PXA     1,4
      ANA     *-1                 GET BIT 35
      PAX     ,2                  C(XR2)=1 OR 0
      LDQ*    YDOT
      FMP     HC
      TRA*    HK11,4              SWITCH FOR PASSES 1-4
HK03  TIX     *-3,1,1             DO (N)
      CLA     TEMP+1,2            GET 0 OR H/2
      FAD*    L(T1)
      STO*    L(T1)
      XCA
      FAD*    L(T2)
      FAD*    L(T1)
      STQ*    L(T2)
      STO*    L(T1)
      SXA     *+4,1
      SXA     *+4,2
      SXA     *+4,4
      XEC     DER1,2              DO DER1 OR DER2
      AXT     **,1
      AXT     **,2
      AXT     **,4
      TIX     HK01,4,1            OUTER LOOP, DO 4
      CLA*    L(T1)               NEW TIME IN BUFFER
      LDQ*    L(T2)
      STO     T1
      STQ     T2
HK08  AXT     **,1
HK09  AXT     **,2
HK10  AXT     **,4
      TRA     1,4
HK02  STO*    YO(2)               K(I) STORAGE
      FDP     HK12
      XCA
HK06  FAD*    YO
      STO*    Y                   YO + K1/2, K2/2, K3
      TRA     HK03
HK04  STO     TEMP+2
      XCA
      FMP     HK12
      FAD*    YO(2)
      STO*    YO(2)               K1 + 2 K2
      CLA     TEMP+2
      TRA     HK02+1
HK05  STO     TEMP+2
      XCA
      FMP     HK12
      FAD*    YO(2)
      STO*    YO(2)               K1 + 2 K2 + I K3
      CLA     TEMP+2
      TRA     HK06
HK07  FAD*    YO(2)
      FDP     HK13
```

```
       XCA                          (K1 + IK2 + IK3 + K4)/6
       FAD*    Y(2)
       FAD*    YO
       STO*    Y
       STQ*    Y(2)
       TRA     HK03
               HK02
               HK04
               HK05
               HK07
 HK11  OCT     1000000000
 HK12  DEC     2.
 HK13  DEC     6.
ADAMS  TRA     ADAMS+7
       TRA     *+2
       NOP
       AXT     *,1
       AXT     *,2
       AXT     *,4
       TRA     I,4                  EXIT
       SXA     ADAMS+3,1
       SXA     ADAMS+4,2
       SXA     ADAMS+5,4
       NZT     E
       TRA     *+6
       NZT     A
       TRA     *+4
       STZ     A
       TSX     ABTB,4
       TRA     GT10+1
       TSX     GAINT,4
       PZE
       CLA     HC
       FAD     T1
       STO     TEMP
       XCA
       FAD     T2
       FAD     TEMP
       STO     T1
       STQ     T2
       STO*    L(T1)
       STQ*    L(T2)
       XEC     DER1                 TO DERIV BOX 1
       NZT     E
       TRA     RGUP
       LXA     (N),1
RGET1  LXA     MP1,2                ERROR TEST 1
       CLA*    DELX,2               SAVE YN,DELTA N
       STO*    DELZ,2
       TIX     *-2,2,1
       NZT     E
       TRA     *+3
       TXI     *+1,2,-1
       TXL     *-6,2,0
       CLA*    YO
       STO*    YN
       CLA*    YO(2)
```

```
       STO*    YN2
       TIX     RGET1,1,1
       CLA     RFIX1
       STO     NI
       LXA     MP1,2
       CLA     GCOFP,2
       FDP     GCOFC-1,2
       XCA
       SSP
       STO     RGA
       TRA     RGUP
 RGUP  TSX     UPDAT,4
       CLA     NI
       TNZ     GND1                    TO CORRECT
  GND  LXA     (N),1
       CLA*    Y
       STO*    YO
       CLA*    Y(2)
       STO*    YO(2)
       TIX     *-4,1,1
       NZT     E
       TRA     *+4
       NZT     A
       TRA     *+2
       TSX     X2.0,4
       TRA     ADAMS+3                 TO EXIT
 GND1  STO     TEMP+3
       TMI     *+3
       CLA     ADAMS+2
       TRA     *+2
       CLA     ADAMS+1
       STO     GFRK2
       STZ     GZP
       TSX     PUTB,4
       LXA     (N),1
       STZ*    YO
       STZ*    YO(2)
       TIX     *-2,1,1
       TSX     GAINT,4                 CORRECT
       OCT     77777
       PXD     0,0
       LXA     MP1,1
       FAD     GCOFC,1                 SUM M+1 COEFF
       TIX     *-1,1,1
       STO     GSIGM
  GLP  XEC     DER2                    TIME UNCH D.B.
       LXA     MP1,4
       LXA     (N),1
       CLA*    YDOT
       FSB*    DELX,4
       FSB*    YO(2)
       STO*    YO
       FAD*    YO(2)
       STO*    YO(2)
       TIX     *-6,1,1
       NZT     E                       AEC TEST 2
       TRA     GFRK2
```

```
      STZ     RGERR                ERROR TEST 2
      STZ     RGYPC
      LXA     (N),1
  RG1 CLA*    Y
      SSP
      CAS     YCLOW
      TRA     *+3
      TRA     *+2
      CLA     YCLOW
      STO     RGDI
      TSX     RGMAX,4
      SSP
      FDP     RGDI
      XCA
      CAS     RGYPC
      TRA     *+3
      TRA     *+2
      CLA     RGYPC
      STO     RGYPC
      FDP     RGA
      CLA     RGERR
      TLQ     *+2
      STQ     RGERR
      TIX     RG1,1,1
      CLA     RGERR
      CAS     ELBAR
      TRA     RG3
      TRA     *+1
      ZET     HD
      TRA     GFRK2
      CLA     HC
      FAD     HC
      CAS     HMAXT
      TRA     GFRK2
      TRA     GFRK2
      CLA     •
      STO     A
      TRA     GFRK2
  RG3 CLA     RGERR
      CAS     EUBAR
      TRA     RGX.2
      TRA     GFRK2
      TRA     GFRK2
RGX.2 LDQ     HC
      FMP     GCOFP-2              0.5 DELTA T
      CAS     HMINT
      TRA     *+3
      TRA     GFRK2
      TRA     GFRK2
      CLA     RFIX1                HALVE
      STO     NH
      CLS     HC
      FAD     T1
      STO     TEMP
      XCA
      FAD     T2
      FAD     TEMP
```

```
       STO     T1
       STQ     T2
       STO*    L(T1)
       STQ*    L(T2)                RESTORE TN
       LXA     (N),1
  RG4  AXT     2,4
       LXA     MP1,2
       CLA*    DELZ,2
       STO*    DELX,2
       TNX     *+2,4,1
       STO*    YDOT
       TIX     *-4,2,1
       TXI     *+1,2,-1
       TXL     *-6,2,0
       CLA*    YN
       STO*    YO
       STO*    Y
       CLA*    YN2
       STO*    YO(2)
       STO*    Y(2)
       TIX     RG4,1,1
       XEC     AEOS
       STZ     ND
       STZ     HD
       CLA     RBIG
       STO     TRG2
       TSX     ABTB,4               FIND NEW TMIN
       TRA     GT10+1               ERROR RETURN
       TRA     GT2+1
 GFRK2 NOP
       LXA     TEMP+3,2
       TXI     *+1,2,-1
       SXA     TEMP+3,2
       TXH     *+2,2,0
       TRA     GADN-2
       LXA     (N),1                FORM NEXT ITERATE
       LDQ     HC
       FMP     GSIGM
       STO     TEMP+1
       LDQ     TEMP+1
       FMP*    YO
       FAD*    Y(2)
       FAD*    Y
       STO*    Y
       STQ*    Y(2)
       TIX     *-6,1,1
       TRA     GLP
       NOP
       LXA     MP1,2                FIXED ITERATIONS END
  GADN LXA     (N),1
       CLA*    YO(2)
       FAD*    DELX,2
       STO*    DELX,2
       TIX     *-3,1,1
       TIX     *-5,2,1              DIFF TABLE UPDATED
       TRA     GND                  TO EXIT
 GAINT TRA     *+7                  PREDICTOR-CORRECTOR SR
```

```
       PZE     GCOFP                CALL SEQ
       PZE     GCOFC                TSX GAINT,4
       AXT     *,1                  PZE K
       AXT     *,2                  NOR RET
       AXT     *,4                  IF K=0,PREDICT
       TRA     2,4                  IF K NOT 0,CORRECT
       SXA     GAINT+3,1
       SXA     GAINT+4,2
       SXA     GAINT+5,4
       CLA     I,4
       TZE     *+3
       CLA     GAINT+2
       TRA     *+2
       CLA     GAINT+1
       STA     *+5
       LXA     (N),1
       LXA     MP1,2
       AXT     1,4
       STZ     TEMP
       LDQ     *,4
       FMP*    DELX,2
       FAD     TEMP
       STO     TEMP
       TXI     *+1,4,1
       TIX     *-5,2,1
       XCA
       FMP     HC
       FAD*    Y
       STO     TEMP
       XCA
       FAD*    Y(2)
       FAD     TEMP
       STO*    Y
       STQ*    Y(2)
       TIX     *-18,1,1
       TRA     GAINT+3              TO EXIT
       DEC     0.2870754484
       DEC     0.294868003
       DEC     0.304224539
       DEC     0.315591936
       DEC     0.329861111
       DEC     0.348611111
       DEC     0.375
       DEC     0.416666666
       DEC     0.5
       DEC     1.0
GCOFP  DEC     -0.0078925542
       DEC     -0.0093565362
       DEC     -0.0113673950
       DEC     -0.0142691795
       DEC     -0.1875E-1
       DEC     -0.263888888E-1
       DEC     -0.416666666E-1
       DEC     -0.833333333E-1
       DEC     -0.5
       DEC     1.0
GCOFC  PZE
```

```
  RGA PZE                          COEFF. FOR ERROR DETER.
RGERR PZE                          E(N+1)
 RGDI PZE                       /D(I)/
RGYPC BSS     3
    A PZE     0                 FLAG WORD FOR TURNING ON DOUBLING(AEC)
RGMAX SXA     RGMX,4
      SXA     RGMX+1,2
      LXA     MP1,2
      AXT     1,4
      STZ     RGYPC+1
      LDQ     GCOFC,4
      FMP*    DELX,2
      FAD     RGYPC+1
      STO     RGYPC+1
      TXI     *+1,4,1
      TIX     *-5,2,1
      LXA     MP1,2
      AXT     1,4
      STZ     RGYPC+2
      LDQ     GCOFP,4
      FMP*    DELZ,2
      FAD     RGYPC+2
      STO     RGYPC+2
      TXI     *+1,4,1
      TIX     *-5,2,1
      FSB     RGYPC+1
      XCA
      FMP     HC
 RGMX AXT     0,4
      AXT     0,2
      TRA     1,4
 FLAG SXA     OOP,1                SAVE INDEX 1
      SXA     OUCH,2               SAVE INDEX 2
      SXA     OUT,4                SAVE INDEX 4
      AXT     OMAR,2
      LXA     I4,4                 INDEX FROM TSX MARK,4
      STZ     ORGY
 OMNI CLA     4,4
      TZE     OUT                  LAST TRIGGER
      TMI     OMER                 IGNORE NEGATIVE TRIGGERS
      PDC     ,1                   ADDRESS IN INDEX1
      TXL     OMER,1,0             IGNORE TIME STOPS
      CLA     0,1                  YSUBJ
      FSB*    5,4
      STO     TEMP
      TMI     OBEY                 SIGN OF DIFFERENCE TEST
      CLA*    L                    YSUBJ-ZSUBJ AT TL
      TPL     OINK                 IF SAME SIGN IGNORE
OZONE CLA     TEMP
      STO*    R                    RJ=WJ FOR FLAG
      STO*    W
      CLA     FLAG
      STT     ORGY                 TEST FOR FLAG RETURN
 ODEN STT     4,4                  FLAG
      TXI     *+1,2,-1
 OMER TXI     *+1,4,-2
      TXH     OMNI,2,0
```

```
      LXA    I4,4          ERROR RETURN FOR TOO MANY
      TRA    3,4
  OUT AXT    ,4            DEPENDENT VARIABLE STOPS
      ZET    ORGY          IF ZERO
      TXI    OOP,4,-1
  OOP AXT    ,1            OTHERWISE NORMAL
 OUCH AXT    ,2
      TRA    1,4
 OBEY CLA*   L             YSUBJ-ZSUBJ AT TL
      TPL    OZONE         SET FLAG IF SIGN NOT EQUAL
 OINK CLA    •             OTHERWISE CLEAR TAG
      TRA    ODEN          AND CONTINUE
  SET SXA    OBIT,4        SAVE INDEX 4
      SXA    OOZE,2        SAVE INDEX 2
      SXA    ODE,1         SAVE INDEX 1
      LXA    I4,4
      AXT    OMAR,2
      CLA    T1
      STO    TL
      CLA    T2            OR
      STO    TL2           OTHERWISE
 OVER CLA    4,4
      TZE    OBIT          LAST TRIGGER
      TMI    OMEN          IGNORE NEGATIVE TRIGGER
      PDC    ,1
      TXL    OMEN,1,0      IGNORE TIME STOPS
      CLA    0,1           YSUBJ
      FSB*   5,4
      STO*   L             YSUBJ-ZSUBJ
      TXI    *+1,2,-1
 OMEN TXI    OPINE,4,-2
OPINE TXH    OVER,2,0
      LXA    I4,4          ERROR RETURN FOR TOO MANY
      TXI    ODE,4,-2      DEPENDENT VARIABLE STOPS
 OBIT AXT    ,4
  ODE AXT    ,1
 OOZE AXT    ,2
      TRA    1,4
 ORGY PZE                  TAG STORAGE
    L PZE    LTAB+OMAR,2
    R PZE    RTAB+OMAR,2
    W PZE    WTAB+OMAR,2
 OMAR EQU    50
 LTAB BSS    OMAR
 RTAB BSS    OMAR
 WTAB BSS    OMAR
 SRCH SXA    OBOY,4        SAVE INDEX 4
      SXA    OVAL,2        SAVE INDEX 2
      SXA    ORB,1         SAVE INDEX 1
      CLA    OBESE
      STO    OGEE          HM=BIG
      STO    HP
      LXA    I4,4          INDEX FROM TSX MARK,4
      AXT    OMAR,2
ONSET CLA    4,4
      TZE    OBOY          LAST TRIGGER
      TMI    OMIT          OMIT NEGATIVE TRIGGERS
```

```
       PDC     0,1
       TXL     OMIT,1,0          OMIT INDEPENDENT STOPS
       STT     ORGY
       NZT     ORGY              TEST WORD
       TRA     ODOR              EQUAL ZERO
       CLA     0,1               NOT EQUAL ZERO
       FSB*    5,4
       STO*    W                 WSUBJ=YSUBJ-ZSUBJ
       FSB*    L
       STO     TEMP+2
       CAL*    W
       ERA*    L
       PBT
       TRA     REACH             EQUAL SIGN SGN WJ=SGN LJ
       SXA     OVARY,4           OPPOSITE IN SIGN
       SXA     OVARY-1,2
       TSX     DSUB,4
       PZE     TL,,T1
       AXT     **,2
OVARY  AXT     **,4
       FDP     TEMP+2            (TL-T)/(WJ-LJ)
       FMP*    W
       STO     TEMP        R
       CLA     OGEE              HM
       LDQ     TEMP              R
       TLQ     *+2
       TRA     RSTOR
       STQ     OGEE              R LESS THAN HM
RSTOR  CLA     TEMP
       SSP
       FSB     DELU
       TPL     OCCUR
       CLA     OFT
       TRA     ONION
OCCUR  CLA     ORB
ONION  STT     4,4
       TRA     ODOR
REACH  CAL*    W
       ERA*    R
       PBT
       TRA     RFLAG             SGN WJ=SGN RJ
       CLA*    W
       FSB*    R
       STO     TEMP+2            WJ-RJ
       SXA     *+4,2
       SXA     *+4,4
       TSX     DSUB,4
       PZE     TR,,T1
       AXT     **,2
       AXT     **,4
       FDP     TEMP+2            (TR-T)/(WJ-RJ)
       FMP*    W
       STO     TEMP
       LDQ     HP
       TLQ     RSTOR
       STO     HP
       TRA     RSTOR
```

```
RFLAG CLA     *
      STT     4,4
 ODOR TXI     *+1,2,-1
 OMIT TXI     *+1,4,-2
      TXH     ONSET,2,0
      LXA     I4,4
      TRA     3,4                 ERROR RETURN (TOO MANY DEPEND. VAR TRI
 OBOY AXT     **,4
  ORB AXT     **,1
      CLA     OGEE
      FSB     OBESE
      TNZ     RSTRJ
      AXT     OMAR,2
      CLA*    W
      STO*    L                   LJ=WJ
      TIX     *-2,2,1
      CLA     T1
      LDQ     T2
      STO     TL
      STQ     TL2
      CLA     HP
      TRA     RSTOT
RSTRJ AXT     OMAR,2
      CLA*    W
      STO*    R                   RJ=WJ
      TIX     *-2,2,1
      CLA     T1
      LDQ     T2
      STO     TR
      STQ     TR2
      CLA     OGEE
RSTOT STO     HIC
      SSP
      LDQ     DELU
 OVAL AXT     **,2
      TLQ     *+2
      TRA     1,4
      CLA     HIC
      TRA     2,4
 OGEE PZE      HM
   HP PZE
OBESE OCT     377777777777
  OFT PZE     ,5
      REM     INTERPOLATION ROUTINE FOR ADAMS-MOULTON (MARK)
      REM     TSX INTRP,4
      REM     NORMAL RETURN
INTRP SXA     RICH,4              ENTER
      SXA     RICH+1,1
      SXA     RICH+2,2
      TSX     DSUB,4
      PZE     TGO,,T1
      STO     RIMU-1              HC(MU)
      FDP     HC
      STQ     RIM                 MU
      LXA     RLOC1+3,2           IR2=2
      LXA     M,4                 IR4=M
      CLA     RIB
```

```
       STO     RIC
       STO     RFACT
       STO     RIMU-2,4
RLOC1  CLA     RIC
       FAD     RIB
       STO     RIC
       XCA     2
       FMP     RFACT
       STO     RFACT
       CLA     RIM            MU,MU-1,.....
       FSB     RIB
       STO     RIM
RZERO  XCA     1
       FMP     RIMU-2,4
       STO     RIMU-1,4
       FDP     RFACT
       XCA
       TNX     RLOC2,2,1
       CHS
       TRA     RLOC2+1
RLOC2  TXI     RLOC2+1,2,1
       STO     RAJ,4                 ASUBJ J=1,...,M
       TIX     RLOC1,4,1
       LXA     M,4
RLOC3  PXD     0,4
       PDX     0,2
       LXA     M,1
       CLA     RIB,2
       STO     RSUMC
       LDQ     RAJ,1
       FMP     RIB+1,2
       FAD     RSUMC
       STO     RSUMC
       TXI     *+1,1,-1
       TIX     RLOC3+5,2,1
       STO     RICJ,4
       TIX     RLOC3,4,1
       LXA     (N),1                 NO. OF. EQUAT.
       LXA     M,2
       LXA     RZERO,4
       STZ     RSUMD
RLOC4  LDQ*    DELX,2                DIFFERENCES
       FMP     RICJ,4
       FAD     RSUMD
       STO     RSUMD
       TXI     *+1,4,1
       TIX     RLOC4,2,1
       FAD*    DELX,4
       XCA
       FMP     RIMU-1
       STO     RSAVE
       CLA*    YO(2)
       FSB     RSAVE
       FAD*    YO
       STO*    Y
       STQ*    Y(2)
       TIX     RLOC4-3,1,1
```

```
       SXA     GZP,1
       XEC     DER1
       XEC     AEOS
 RICH  AXT     **,4
       AXT     **,1
       AXT     **,2
       TRA     1,4                     EXIT
 RIMU  BES     17
  RIM  PZE                             MU(MU-1)(MU-2)...(MU-I)
 RMP1  PZE                             M+1
RSAVE  PZE
       DEC     -0.382689955E-2              B15
       DEC     -0.421495223E-2              B14
       DEC     -0.467749840E-2              B13
       DEC     -0.523669325E-2              B12
       DEC     -0.592405641E-2              B11
       DEC     -0.678584998E-2              B10
       DEC     -0.7892554012E-2             B9
       DEC     -0.935653659E-2              B8
       DEC     -.1136739417E-1         B7
       DEC     -0.1426917989E-1        B6
       DEC     -0.01875                B5
       DEC     -0.0263888889           B4
       DEC     -0.0416666667           B3
       DEC     -0.0833333333           B2
       DEC     -0.5                    B1
  RIB  DEC     1.0                     B0 BSUBJ VALUES (0-10)
  RIC  PZE
RFACT  PZE                             FACTORIALS
  RAJ  BES     15                      A SUB J
 RICJ  BES     15                      C SUB J
RSUMC  PZE
RSUMD  PZE
UPDAT  SXA     HU03,1
       SXA     HU04,2
       LXA     (N),1                   C(XR1)=(N)
       LXA     MP1,2                   C(XR2)=M+1
       ZET     E
       TXI     *+1,2,1                 BUMP XR2 IF E=1
       SXA     HU01,2
       CLA     HU05
       ADD     E
       STA     HU06                    ADDRESS IS DELX + E
       STA     HU07
 HU01  AXT     **,2
       CLA*    YDOT
       STO     TEMP
 HU02  CLA     TEMP
       STO     TEMP+1
 HU06  FSB*    **,2
       STO     TEMP
       CLA     TEMP+1
 HU07  STO*    **,2
       TIX     HU02,2,1                INNER LOOP, DO M+ 1+E
       TIX     HU01,1,1                OUTER LOOP, DO (N)
 HU03  AXT     **,1
 HU04  AXT     **,2
```

```
       TRA     1,4
 HU05  PZE     DELX
 DADD  TRA     DSUB+6              TSX DADD-DSUB,4
 DSUB  TRA     DSUB+3              PZE L(A1),0,L(B1)
       PZE
       PZE
       CLA     DSUB
       STO     DSUB+2
       TRA     *+2
       STZ     DSUB+2
       CLA     1,4
       STO     DSUB+1
       LDC     DSUB+1,1
       LAC     DSUB+1,2
       CLA     1,1
       ZET     DSUB+2
       CHS
       FAD     1,2
       STO     TEMP
       CLA     0,1
       ZET     DSUB+2
       CHS
       FAD     0,2
       STO     TEMP+1
       XCA
       FAD     TEMP
       FAD     TEMP+1
       TRA     2,4
 START SXA     GRT1,4              TEST TRANSITION FROM RK TO AM
       CLA     H
       FAD     T1
       STO     TEMP
       XCA
       FAD     T2
       FAD     TEMP
       STO     TRG2
       STQ     TRG2+1
       TSX     UPDAT,4
       CLA     J
       ADD     START+30
       STO     J
       ZET     E
       SUB     START+30
       SUB     MP1
       TNZ     GRT1
       CLA     START+31
       STO     TRG2
       CLA     H
       STO     HC
       LXA     (N),1
       CLA*    Y
       LDQ*    Y(2)
       STO*    YO
       STQ*    YO(2)
       TIX     *-4,1,1
       TRA     GTO                 TO AMC
 GRT1  AXT     **,4
```

```
       TRA     1,4
       DEC     1
       OCT     377777777777
 PUTB  SXA     PTB1,4
       LXA     (N),1                RESET DEP VAR BANK
       CLA*    YO                   FROM SECONDARY BANK
       STO*    Y
       CLA*    YO(2)
       STO*    Y(2)
       TIX     *-4,1,1
       ZET     GZP
       XEC     DER1
       STZ     GZP
 PTB1  AXT     **,4
       TRA     1,4
  GZP  PZE
• SUBR FOR DOUBLING (MARK)
*      TSX     X2.0
•      NORMAL RETURN
 X2.0  SXA     RX2.0,4
       SXA     RX2.0+1,1
       SXA     RX2.0+2,2
       CLA     M
       ZET     E
       CLA     MP1
       ACL     RMAG
       FAD     RMAG
       XCA
       FMP     HC
       STO     TEMP
       XCA
       FAD     T2
       STO     TEMP+1
       CLA     T1
       FAD     TEMP
       STO     TEMP
       XCA
       FAD     TEMP+1
       FAD     TEMP
       STO     TRG2
       STQ     TRG2+1
       CLA     RALF2
       STA     TRIGO
       CLA     RMAG
       STO     HD
       TSX     RSUM,4
       LXA     (N),1
 RAGS  LXA     MP1,2
 ROGS  CLA*    DELX,2
       STO*    DELY,2
       TIX     ROGS,2,1
       NZT     E
       TRA RIGS
       TXI     *+1,2,-1
       TXL     ROGS,2,0
 RIGS  TIX     RAGS,1,1
       TSX     RUPDA,4
```

```
RX2.0 AXT     **,4
      AXT     **,1
      AXT     **,2
      TRA     1,4
 RISK SXA     RX2.0,4
      SXA     RX2.0+1,1
      SXA     RX2.0+2,2
      STZ     HD
      LDQ     HC
      FMP     RDUBL
      STO     HC                    HC=2.0HC
      CLA     RBIG
      STO     TRG2
      TSX     RSUM,4
      LXA     (N),1
 RIN1 CLA     RAD1
      STA     RIN2
      LXA     MP1,2
      SXD     RIN3,2
      SXD     RIN4,2
      AXT     1,2
      AXT     1,4
      NZT     E
      TRA     RIN2
      AXT     0,2
      AXT     0,4
 RIN2 CLA*    DELY,2
      STO*    DELY,4
      TXI     *+1,2,2
      TXI     *+1,4,1
 RIN3 TXL     RIN2,2,0
 RIN4 TXL     ROCK,4,0
      TRA     ROLL
 ROCK CLA     RAD2
      STA     RIN2
      AXT     2,2
      CLA     M
      LBT
      TRA     RIN5
      ZET     E                     M ODD
      AXT     2,2                   ERR CONTROL(SKIP FOR NO ERR CONTROL
      TRA     RIN2
 RIN5 AXT     3,2                   M EVEN
      NZT     E
      TRA     RIN2                  NO ERROR CONTROL
      AXT     1,2
      TRA     RIN2                  ERROR CONTROL
 ROLL TIX     RIN1,1,1
      TSX     RUPDA,4
      TRA     RX2.0
 RMAG OCT     233000000000
RALF2 TSX     RISK,4
RDUBL DEC     2.0
 RBIG OCT     377777777777
 RAD1 PZE     DELY
 RAD2 PZE     DELX
* SUBR FOR HALVING(MARK)
```

```
*       TSX     X.50,4
*       NORMAL  RETURN
  X.50  SXA     RX.5,4
        SXA     RX.5+1,1
        SXA     RX.5+2,2
        STZ     ND
        STZ     HD
        CLA     RBIG
        STO     TRG2
        LDQ     HC
        FMP     RO.5
        STO     HC
        CLA     M
        ZET     E
        CLA     MP1
        ACL     RMAGN
        FAD     RMAGN
        STO     RFLOM
        STO     RFACM-1
        FSB     RFLO1
        STO     RFACM-2
        LXA     M,2
        ZET     E
        LXA     MP1,2
        TXI     RNTR5,2,-2
 RNTR5  LDQ     RFACM-1
        FMP     RFACM-2
        STO     RFACM-1
        CLA     RFACM-2
        FSB     RFLO1
        STO     RFACM-2
        TIX     RNTR5,2,1
        LXA     (N),1
        TSX     RSUM,4
 RNTR1  STZ     RN
        STZ     RNPRI
        LXA     MP1,2
        LXA     MP1,4
 RNTRO  CLA     RNPRI
        LBT
        TRA     REVEN           N EVEN
  RODD  STZ     RKO             N ODD
        SXA     RSAV2,2
        SXA     RSAV3,4
        LXA     M,4
        NZT     E
        TXI     *+1,4,-1
        LDQ     RN
        FMP     RO.5
        CHS
        STO     RIDEL-1
        FAD     RFLO1
        STO     RIDEL-2
        FAD     RFLO1
        STO     RIDEL-3
        STZ     RSUM5
 RNTR3  LDQ     RIDEL-2
```

```
        FMP     RIDEL-3
        STO     RIDEL-2
        CLA     RIDEL-3
        FAD     RFLO1
        STO     RIDEL-3
        TIX     RNTR3,4,1
        CLA     RIDEL-2
        FDP     RFACM-1
        STQ     RAO
        LXA     MP1,2
RNTR2   LDQ     RAO
        FMP*    DELX,2
        FAD     RSUM5
        STO     RSUM5
        CLA     RFLOM
        FSB     RKO
        STO     RFACM-2
        CLA     RIDEL-1
        FAD     RKO
        STO     RIDEL-3
        FAD     RFLO1
        STO     RIDEL-2
        CLA     RKO
        FAD     RFLO1
        STO     RKO
        LDQ     RKO
        FMP     RIDEL-2
        STO     RTEMP
        LDQ     RFACM-2
        FMP     RIDEL-3
        FDP     RTEMP
        FMP     RAO
        CHS
        STO     RAO
        TIX     RNTR2,2,1
        NZT     E
        TRA     RSAV2
        TXI     *+1,2,-1
        TXL     RNTR2,2,0
RSAV2   AXT     **,2
RSAV3   AXT     **,4
        CLA     RSUM5
        TRA     REVEN+1
REVEN   CLA*    DELX,2
        STO*    DELY,4
        CLA     RN
        FAD     RFLO1
        STO     RN
        CLA     RNPRI
        ADD     RFIX1
        STO     RNPRI
        LBT
        TRA     *+2
        TXI     *+1,2,-1
        TIX     RNTRO,4,1
        NZT     E
        TRA     RNTR4
```

```
       TXI     *+1,4,-1
       TXL     RNTR0,4,0
RNTR4  TIX     RNTR1,1,1
       TSX     RUPDA,4
 RX.5  AXT     **,4
       AXT     **,1
       AXT     **,2
       TRA     1,4
RMAGN  OCT     233000000000
 R0.5  DEC     0.5
RFLO1  DEC     1.0
RFLOM  PZE                          FLOAT M
   RN  PZE                          N
  RKO  PZE                          K
RIDEL  BES     3
RSUM5  PZE
RFACM  BES     3
  RAO  PZE
RTEMP  PZE
RNPRI  PZE
RFIX1  PZE     1
RUPDA  SXA     RUP3,4
       SXA     RUP3+1,1
       SXA     RUP3+2,2
       LXA     MP1,2
       SXD     RUP2,2
       AXT     0,2
 RUP0  LXA     (N),1
 RUP1  CLA*    DELY,2
       STO*    YDOT
       TIX     RUP1,1,1
       TSX     UPDAT,4
       TXI     RUP2,2,1
 RUP2  TXL     RUP0,2,0
 RUP3  AXT     **,4
       AXT     **,1
       AXT     **,2
       TRA     1,4
* SUBR TO OBTAIN DERIVATIVES FROM DIFFERENCES
*      TSX     RSUM,4
*      NORMAL  RETURN
 RSUM  SXA     RIRS,4
       SXA     RIRS+1,1
       SXA     RIRS+2,2
       NZT     E
       TRA     REO
       AXT     1,2
       CLA     MP1
       STA     RSUM1
       STA     RSUM2-1
 RECH  CLA     RADDS,2
       STA     RSUM3
       CLA     RADDS-1,2
       STA     RSUM4
       STA RSUM7
       LXA     (N),1               IR1=(N)
RSUM1  AXT     0,2
```

```
        SXD     RSUM6,2
        AXT     0,4
RSUM2   AXT     1,2
RSUM3   CLA*    DELX-1,2
RSUM4   FSB*    DELX,2
RSUM7   STO*    DELX,2
        TXI     RSUM6,2,1
RSUM6   TXL     RSUM3,2,**
        LXD     RSUM6,2
        TXI     *+1,2,-1
        SXD     RSUM6,2
        TIX     RSUM2,4,1
        TIX     RSUM1,1,1
 RIRS   AXT     **,4
        AXT     **,1
        AXT     **,2
        TRA     1,4
  REO   CLA     M
        STA     RSUM1
        STA     RSUM2-1
        AXT     0,2
        TRA     RECH
        PZE     DELX+1
        PZE     DELX
 RADDS  PZE     DELX-1
TGLO    SYN     TGO
        END
```

```
$IBMAP CLR
*              SUBROUTINE TO CLEAR MEMORY LOCATIONS
       ENTRY   CLEAR
 CLEAR SAVE    1,2,4
*              TEST FOR END OF ARGUMENTS
 BEGIN CAL     3,4
       ANA     =0777777000000
       TNZ     RETRN
       CLA     4,4
       STA     LOC
       SUB     3,4
       TMI     *+5
       PAX     **,1
       STZ*    LOC
       TIX     *-1,1,1
       AXT     0,1
       STZ*    LOC
       TXI     BEGIN,4,-2
 RETRN RETURN  CLEAR
 LOC   PZE     **,1
       END
```

```
$IBMAP EMK
*      CALL EMARK(HC,RGERR)
        ENTRY   EMARK
 EMARK  SAVE    1,2,4
        EXTERN  HC
        EXTERN  RGERR
        CLA     HC
        STO*    3,4
        CLA     RGERR
        STO*    4,4
        RETURN  EMARK
        END
```

```
$IBMAP UN8
       ENTRY   .UN08.
.UN08. PZE     UNIT08
UNIT08 FILE    ,B(1),BIN,BLK=256,INOUT,READY
       END
```

```
$IBMAP SMK1
*7044 FORTRAN IV SET UP ROUTINE FOR JP-MARK
*                PROGRAMMER-R.GOODELL
*                ASSEMBLED-MARCH 17,1964
*                MAP-7044
*       CALL     SMARK(KIND,N,HBANK-3,NRTN,NTRG,EU,EL,HMX,HMN,YL,
*       ETC      LV1,TV1,
*       ETC      LV2,TV2,
*       ETC      UP TP TEN TRIGGERS    )
        ENTRY    SMARK
 SMARK  SAVE     1,2,4
        TXI      *+1,4,-2
        CAL*     1,4
        ALS      15
        ORA      ADER            DER1,0,DER2
        SLW      CMARK+2
        CLA      3,4
        ADD      =3
        STA      CMARK+1
        ADD      =1
        STA      STON
        ADD      =1
        STA      AINV
        ADD      =1
        STA      *+1
        STZ      **
        CLA*     2,4
        ALS      18
 STON   STD      **
        CLA      4,4
        STA      NRTN
        CLA      5,4
        STA      NTRG
        CLA*     6,4
        STO      EUBAR
        EXTERN   EUBAR
        CLA*     7,4
        STO      ELBAR
        EXTERN   ELBAR
        CLA*     8,4
        STO      HMAXT
        EXTERN   HMAXT
        CLA*     9,4
        STO      HMINT
        EXTERN   HMINT
        CLA*     10,4
        STO      YCLOW
        EXTERN   YCLOW
        STZ      RGERR
        EXTERN   RGERR
        AXT      0,1
 STRG   CLA      11,4
        ANA      =0777777000000
        TNZ      ENTRG
        CLA      11,4
        CAS      AINV            INDEPENDENT VARIABLE
        TRA      *+2
```

```
       CLA     =0
       ALS     18
       STO     TRG,1
       CLA     TRG,1            TURN ON TRIGGER
       SSP
       STO     TRG,1
       CLA     12,4
       STA     TRG+1,1
       TXI     *+1,1,-2
       TXL     CMARK,1,-20
       TXI     STRG,4,-2
ENTRG  CLA     TRG,1            TURN OFF REMAINING TRIGGERS
       SSP
       CHS
       STO     TRG,1
       TXI     *+1,1,-2
       TXH     ENTRG,1,-20
CMARK  TSX     MARK,4
       EXTERN  MARK
       PZE     **,0,EOS
       PZE     DER1,,DER2
       TRA     EMARK
TRG    PZE     TRG1
       PZE
       PZE     TRG2
       PZE
       PZE     TRG3
       PZE
       PZE     TRG4
       PZE
       PZE     TRG5
       PZE
       PZE     TRG6
       PZE
       PZE     TRG7
       PZE
       PZE     TRG8
       PZE
       PZE     TRG9
       PZE
       PZE     TRG10
       PZE
       PZE
EOS    CLA     =1
       TRA     EXIT
DER1   CLA     =2
       TRA     EXIT
DER2   CLA     =3
       TRA     EXIT
GTRG   CLA     =4
EXIT   SXA     IX4,4
       SXA     IX1,1
       SXA     IX2,2
       STO*    NRTN
       RETURN  SMARK
TRA14  AXT     0,0
       LXA     IX4,4
```

```
        LXA     IX1,1
        LXA     IX2,2
        TRA     1,4
        ENTRY   TRA14
TRA24   AXT     0,0
        LXA     IX4,4
        LXA     IX1,1
        LXA     IX2,2
        TRA     2,4
        ENTRY   TRA24
*       CALL    ON(NOTRG)
        ENTRY   ON
ON      SAVE    1,2,4
        CLA*    3,4
        ALS     1
        PAC     **,1
        CLA     TRG-2,1
        SSP
        STO     TRG-2,1
        RETURN  ON
*       CALL    OFF(NOTRG)
        ENTRY   OFF
OFF     SAVE    1,2,4
        CLA*    3,4
        ALS     1
        PAC     **,1
        CLA     TRG-2,1
        SSP
        CHS
        STO     TRG-2,1
        RETURN  OFF
TRG1    CLA     =1
        TRA     ETRG
TRG2    CLA     =2
        TRA     ETRG
TRG3    CLA     =3
        TRA     ETRG
TRG4    CLA     =4
        TRA     ETRG
TRG5    CLA     =5
        TRA     ETRG
TRG6    CLA     =6
        TRA     ETRG
TRG7    CLA     =7
        TRA     ETRG
TRG8    CLA     =8
        TRA     ETRG
TRG9    CLA     =9
        TRA     ETRG
TRG10   CLA     =10
ETRG    STO*    NTRG
        CLA     =4
        TRA     EXIT
EMARK   CLA     =5
        TRA     EXIT
IX1     PZE
IX2     PZE
```

```
IX4    PZE
ADER   PZE      DER1,0,DER2
AINV   PZE
NRTN   PZE
NTRG   PZE
       END
```

```
$IBMAP ATM1
       REM U. S. STANDARD ATMOSPHERE , 1962
*              J.MALTAIS  RAYTHEON
       REM
       REM
       REM
       REM
       EXTERN  ALOG
       EXTERN  EXP
       EXTERN  SQRT
       ENTRY   ATM
 ATM   SAVE    1,2,4
       TXI     *+1,4,-2
       LDQ*    1,4
       FMP FTKM                  CONVERT ALTITUDE TO KILOMETERS
       STO Z
       CLA 2,4
       PAC     **,2
       LXA ALTM,1
       CLA Z
       CAS ALTM+22
       TRA *+3                   GEOMETRIC ALT GR(700KM)
       TRA *+2                   GEOMETRIC ALT=(700KM)
       TRA *+5
       STZ     0,2
       STZ     1,2
       STZ     2,2
       TRA FIN
       CAS ALTM+9
       TRA OVER90                GEOMETRIC ALT GR(90KM)
       TRA *+1                   GEOMETRIC ALT=(90KM)
       REM
       REM COMPUTE GEOPOTENTIAL ALTITUDE
       REM
       FAD RE
       STO TEMP                  (RE+Z)
       CLA Z
       FDP TEMP
       FMP RE
       STO H
       CAS ALTM+1,1
       TXI *-1,1,-1
       TXI *-2,1,-1
       FSB ALTM,1
       STO TEMP+1                (H-HB)
       LDQ MOVR
       FMP GZERO
       STO TEMP+2                (GO*MO/R)
       REM
       LDQ TEMP+1
       FMP LM,1
       FAD TMK,1
       STO TM
       REM
       REM COMPUTE SPEED OF SOUND
       REM
       LDQ TM
```

```
        FMP GMOVR
        STO     ARG
        CALL    SQRT(ARG)
        STO     2,2
        REM
        REM COMPUTE PRESSURE FOR ALTITUDES LESS THAN 90KM
        REM
        CLA LM,1
        TZE     A1
        LDQ LM,1
        FMP TEMP+1
        FAD TMK,1
        FDP TMK,1
        STQ TEMP
        CALL    ALOG(TEMP)
        FDP LM,1
        TRA *+3
A1      CLA     TEMP+1
        FDP TMK,1
        FMP TEMP+2
        CHS
        FAD     LOG.P.,1
SAME    STO     ARG
        CALL    EXP(ARG)
        STO TEMP
        LDQ TEMP
        FMP PCONF
        STO     0,2
        REM
        REM COMPUTE DENSITY
        REM
        CLA TEMP
        FDP TM
        FMP MOVR
        STO TEMP
        LDQ TEMP
        FMP DCONF
        STO     1,2
        CLA RE
        FAD Z
        STO ZB
        TSX GETG,4
        STO TEMP
        LDQ TEMP
        FMP VCONF
        STO TEMP
        CLA     1,2
        FDP TEMP
        STQ     1,2
        REM
FIN     RETURN  ATM
        REM
        REM
OVER90  CAS ALTM+10,1
        TXI *-1,1,-1
        TXI *-2,1,-1
        FSB ALTM+9,1
```

```
        STO TEMP+1              (Z-ZB)
        REM
        LDQ TEMP+1
        FMP LM+8,1
        FAD TMK+9,1
        STO TM
        REM
        CLA ALTM+9,1
        FAD RE
        STO ZB
        CLA TMK+9,1
        FDP LM+8,1
        STQ TEMP
        CLA     =883.99
        STO     2,2
        REM
        TSX GETG,4
        FDP TEMP
        STQ TEMP+2
        CLA TEMP+1
        FDP FL2
        STQ TEMP+4
        CLA TEMP+4
        FAD ZB
        STO ZB
        CLA TEMP
        FAD TEMP+4
        STO TEMP+3
        TSX GETG,4
        FDP TEMP+3
        STQ TEMP+3
        CLA TEMP+3
        ADD C2
        FAD TEMP+2
        STO TEMP+2
        CLA ZB
        FAD TEMP+4
        STO ZB
        CLA TEMP
        FAD TEMP+1
        STO TEMP+3
        TSX GETG,4
        FDP TEMP+3
        STQ TEMP+3
        CLA TEMP+3
        FAD TEMP+2
        FDP FL3
        FMP TEMP+4
        FDP LM+8,1
        FMP MOVR
        CHS
        FAD     LOG.P.+9,1
        TRA SAME
        REM
        REM
GETG    LDQ GCON2
        FMP ZB
```

```
        STO TEMP+5
        CLA RE
        FDP ZB
        STQ TEMP+6
        FMP TEMP+6
        STO TEMP+6
        LDQ J
        FMP GCON1
        STO TEMP+7
        LDQ TEMP+6
        FMP TEMP+7
        FAD FL1
        STO TEMP+6
        CLA GM
        FDP ZB
        FMP TEMP+6
        FDP ZB
        STQ TEMP+6
        CLA TEMP+6
        FSB TEMP+5
        TRA 1,4
        REM
        REM ALTITUDE IN KILOMETERS
        REM
ALTM    DEC 0.,11.,20.,32.,47.,52.,61.,79.,88.75
        DEC 90.,100.,110.,120.,150.,160.,170.,190.,230.
        DEC 300.,400.,500.,600.,700.
        REM
        REM MOLECULAR SCALE TEMPERATURE(DEGREES KELVIN)
        REM
TMK     DEC 288.15,216.65,216.65,228.65,270.65,270.65,252.65
        DEC 180.65,180.65
        DEC 180.65,210.65,260.65,360.65,960.65,1110.65,1210.65
        DEC 1350.65,1550.65,1830.65,2160.65,2420.65,2590.65
        DEC 2700.65
        REM
        REM GRADIENT ... (DEGREES KELVIN)/KILOMETER
        REM
LM      DEC -6.5,0.,1.,2.8,0.,-2.,-4.,0.
        DEC 3.,5.,10.,20.,15.,10.,7.,5.,4.,3.3,2.6,1.7,1.1
        REM
        REM LOG(PRESS) IN NEWTON/METER**2
        REM
LOG.P.  DEC 11.526088,10.027120,8.6079238,6.7662078,4.7086739
        DEC 4.0775459,2.9019652,.37006687E-1,-.18055744E1
        DEC -.18055744E1,-3.5040611,-4.9124566,-5.9828220
        DEC -7.5886380,-7.9035492,-8.1833674,-8.6884560
        DEC -9.5726885,-10.879634,-12.421645,-13.724117
        DEC -14.879663,-15.942631
        REM
        REM
DCONF   DEC 6.24277551E-5
PCONF   DEC 2.08854676E-2
VCONF   DEC 3.2808399
FTKM    DEC 3.048E-4
GZERO   DEC 9.80665             M/SEC**2
MOVR    DEC 3.48367
```

```
GMOVR   DEC 4325.74681
RE      DEC 6378.178
GM      DEC 3.9862216E8
J       DEC 1.6234950E-3
GCON1   DEC .48994698          (3*(SIN(PSI))**2 - 1)
GCON2   DEC 2.6765661E-7       (OMEGA**2)*(COS(PSI))**2
Z                              GEOMETRIC ALT
A
H                              GEOPOTENTIAL ALT
TM                             MOLECULAR SCALE TEMPERATURE
ZB
F.001   DEC .001
FL1     DEC 1.
FL2     DEC 2.
FL3     DEC 3.
C2      OCT 2000000000
TEMP    BSS 8
 ARG    PZE
        END
```

```
$IBMAP BLN1
*              BIVARIATE LINEAR INTERPOLATION
*      CALL    BILIN(NERR,X,XTAB,Y,YTAB,Z1,Z1TAB,Z2,Z2TAB,ETC.)
       ENTRY   BILIN
 BILIN SAVE    1,2,4
       CLA     5,4
       STA     XTAB
       ADD     =1
       STA     XTAB+1
       CLA     7,4
       STA     YTAB
       ADD     =1
       STA     YTAB+1
       CLA     =1                NORMAL RETURN INDICATOR
       STO*    3,4
       AXT     0,1               XTABLE LOOK UP
       CLA*    XTAB+1
       CAS*    XTAB
       TRA     *+3
       TRA     *+2
       TXI     BLN1,1,1
       CAS*    4,4               X
       TRA     BLN1
       TRA     BLN1
       TXI     *-8,1,-1
 BLN1  CLA*    XTAB+1
       FSB*    XTAB
       STO     TEMP
       CLA*    4,4               X
       FSB*    XTAB
       FDP     TEMP
       STQ     DELX
       SXA     NX,1
       LAC     NX,2
       SXA     NX,2
*              OFF TABLE DETERMINATION
       CLA     DELX
       CAS     =-0.0
       TRA     BLN2
       TRA     BLN2
       STZ     DELX              X LOW
       CLA     =2
       STO*    3,4
       TRA     BLN3
 BLN2  CAS     =1.0
       TRA     *+3
       TRA     BLN3
       TRA     BLN3
       CLA     =1.0              X HIGH
       STO     DELX
       CLA     =3
       STO*    3,4
 BLN3  TXI     *+1,1,-1          NO. OF VALUES IN X TABLE
       CLA*    XTAB+1
       CAS*    XTAB
       TRA     *-3
       TRA     *-4
```

```
        SXA     NXTAB,1
        LAC     NXTAB,2
        TXI     *+1,2,1
        SXA     NXTAB,2
        AXT     0,1             Y TABLE LOOK UP
        CLA*    YTAB+1
        CAS*    YTAB
        TRA     *+3
        TRA     *+2
        TXI     BLN4,1,1
        CAS*    6,4             Y
        TRA     BLN4
        TRA     BLN4
        TXI     *-8,1,-1
BLN4    CLA*    YTAB+1
        FSB*    YTAB
        STO     TEMP
        CLA*    6,4             Y
        FSB*    YTAB
        FDP     TEMP
        STQ     DELY
        SXA     NY,1
        LAC     NY,2
        SXA     NY,2
*               OFF TABLE DETERMINATION
        CLA     DELY
        CAS     =-0.0
        TRA     BLN5
        TRA     BLN5
        STZ     DELY
        CLA     =4              Y LOW
        STO*    3,4
        TRA     BLN6
BLN5    CAS     =1.0
        TRA     *+3
        TRA     BLN6
        TRA     BLN6
        CLA     =1.0            Y HIGH
        STO     DELY
        CLA     =5
        STO*    3,4
BLN6    LDQ     NXTAB
        MPY     NY
        STQ     NXNY
BLN7    CLA     8,4
        ANA     =0777777000000
        TNZ     BLN8
        CLA     9,4
        STA     ZTAB
        ADD     =1
        STA     ZTAB+1
        CLA     NXNY
        ADD     NX
        PAC     **,1
        CLA*    ZTAB
        STO     Z1
        CLA*    ZTAB+1
```

```
        STO     Z2
        CLA     NXNY
        ADD     NX
        ADD     NXTAB
        PAC     **,1
        CLA*    ZTAB
        STO     Z3
        CLA*    ZTAB+1
        STO     Z4
        CLA     Z2
        FSB     Z1
        STO     TEMP
        LDQ     TEMP
        FMP     DELX
        FAD     Z1
        STO     Z12
        CLA     Z4
        FSB     Z3
        STO     TEMP
        LDQ     TEMP
        FMP     DELX
        FAD     Z3
        FSB     Z12
        STO     TEMP
        LDQ     TEMP
        FMP     DELY
        FAD     Z12
        STO*    8,4
        TXI     BLN7,4,-2
BLN8    RETURN  BILIN
XTAB    PZE     **,1
        PZE     **,1
YTAB    PZE     **,1
        PZE     **,1
ZTAB    PZE     **,1
        PZE     **,1
TEMP    PZE
DELX    PZE
DELY    PZE
NX      PZE
NY      PZE
NXTAB   PZE
NXNY    PZE
Z1      PZE
Z2      PZE
Z3      PZE
Z4      PZE
Z12     PZE
        END
```

```
$IBMAP LIN1
*               CALL MULG(NDEP,NERR,X,XTAB,Y1,Y1TAB,ETC.)
        ENTRY   MULG
MULG    SAVE    1
        CLA     3,4             NDEP
        STA     *+1
        LAC     **,1            IX1=-NDEP
        TXI     *+1,1,1
        SXD     M2,1
        CLA     6,4
        STA     XTAB
        ADD     =1B35
        STA     XTAB1
        LXA     =0B35,1
M1      CLA*    XTAB1           XTAB+1,1
        CAS*    XTAB            XTAB,1
        TRA     *+3             XTAB LESS THAN XTAB+1
        TRA     *+2             XTAB EQUALS XTAB+1
        TXI     M3,1,1          XTAB GREATER THAN XTAB+1
        CAS*    5,4             X
        TRA     M3              X LESS THAN XTAB+1
        TRA     M3              X EQUALS XTAB+1
        TXI     *+1,1,-1        X GREATER THAN XTAB+1
M2      TXH     M1,1,**         TXH M1,1,-NDEP+1
        TXI     *+1,1,1
M3      CLA*    XTAB1                   XTAB+1,1
        FSB*    XTAB            XTAB,1
        STO     D
        CLA*    5,4             X
        FSB*    XTAB            XTAB,1
        STO     DELTA
        TZE     *+2
        TMI     LOW
        CAS     D
        TRA     HIGH
        NOP
OK      FDP     D
        STQ     DELTA
        CLA     =1B35
        STO*    4,4
        TRA     M7
LOW     CLA     =2B35
        STO*    4,4
        STZ     DELTA
        TRA     M7
HIGH    CLA     =3B35
        STO*    4,4
        CLA     =1.0
        STO     DELTA
M7      TXI     *+1,4,-6
        CLA     1,4             PZE Y
        ANA     =0777777000000
        TZE     M9
        RETURN  MULG
M9      CLA     2,4
        STA     YTAB
        ADD     =1B35
```

```
        STA     YTAB1
M8      CLA*    YTAB1           YTAB+1
        FSB*    YTAB            YTAB
        STO     D
        LDQ     D
        FMP     DELTA
        FAD*    YTAB            YTAB
        STO*    1,4             Y
        TXI     M7+1,4,-2
D       PZE
DELTA   PZE
XTAB    PZE     XTAB,1
XTAB1   PZE     XTAB+1,1
YTAB    PZE     YTAB,1
YTAB1   PZE     YTAB+1,1
        END
```

# Appendix I

## MARK INTEGRATION ROUTINE

### IDENTIFICATION

MARK: Adams-Moulton, Runge-Kutta Integrator
Donald E. Richardson, George Gianopulos, 2/19/62
Jet Propulsion Laboratory
IBM 7090 - FAP

### ABSTRACT

MARK is a closed subroutine designed to solve the first n of a set, N, of first order differential equations simultaneously utilizing Adams-Moulton open or open and closed formula types. A Runge-Kutta 4th order integrator is used as a starting routine to generate backward differences initially. Provision is made for interrupting the integration process at specified values of either the independent or the dependent variables. The order of differences (m) used in the Adams-Moulton mode is less than or equal to nine (9). ($m \leq 9$)

### RESTRICTIONS

1. MARK will not integrate backwards in the independent variable. The nominal step-size, H, must be positive. Changes in H must be accomplished by the use of a "doubling" or "halving" procedure in MARK that will double (set H = 2H) or halve (set H = 0.5H) the integration step size.

2. Underflow and overflow are not checked internally.

3. The user must provide the necessary interruption subroutines, an auxiliary program to evaluate the n first order derivatives, and a bank of storage for internal calculations.

4. This is a FAP program and is <u>not</u> FORTRAN compatible.

### METHOD

1. MARK permits the user to solve the N differential equations by one of three options:

   a. Runge-Kutta 4th order

   b. Adams-Moulton with a fixed step size, H, and the ability to alter H by the doubling and/or halving procedure using Runge-Kutta to initially generate

backward differences. This applies either a predictor or a predictor with q corrections (open or open/closed type formulas).

c. Adams-Moulton as mentioned in b using an automatic variable step size control. Halving and doubling are controlled automatically. The correction formula is applied only once. These methods will be described in further detail in Appendix A.

2. Both the independent and the dependent variables are automatically carried internally in partial double precision to control round-off error locally. The user, however, will recognize the variables only as single precision quantities. However, the user may carry the independent variable in full double precision by option.

USAGE

1. Calling Sequence:

```
CALL MARK or TSX $MARK,4
PZE HBANK,P,EOS
PZE DER1,  Φ,  DER2
ERROR RETURN
Pfx B1,,Y1
PZE Z1
Pfx B2,,Y2
PZE Z2
.
.
.
Pfx BJ,,YJ
PZE ZJ
PZE O
```

where the symbols are defined as follows:

HBANK - The location of a bank of storage to be described below.

P { 0 - The independent variable is carried in partial double precision (single precision to the user).
  { 1 - The independent variable is carried in full double precision

EOS - The location of a user "end of step" routine. This routine must terminate with a TRA 1,4 command. It is used to evaluate variables that are needed only after a full integration step is completed.

DER1 - The location of the entry to the user's derivative routine that carries out all calculations that involve the independent variable. This routine must terminate with a TRA 1,4 command.

DER2 - The location of the entry to that portion of the user's derivative routine that carries out all calculations that do not involve the independent variable but are required to evaluate the derivatives.

A simple example of the use of DER1, DER2 follows:

Suppose we are to solve:

$$\frac{dy}{dx} = ax^2 + by$$

Then:

```
DER1 ax^2

DER2 by
     dy/dx
TRA 1,4
```

Thus the DER1 entry calculates the extra term involving the independent variable x. This provides a saving of real machine time, particularly during the Runge-Kutta phase of integration, but also saves machine time when the closed type formula is used with Adams-Moulton integration.

$\Phi$ {
- 0 - Adams-Moulton integration with fixed step size
- 2 - Runge-Kutta integration only
- 4 - Adams-Moulton using automatic variable step size control

The pairs of locations in the calling sequence specified as:

Pfx BJ,,YJ are defined as "triggers".

PZE ZJ

These triggers are the linkage control to the user's interruption subroutines. The triggers state that control is

transferred to location BJ when the contents of location YJ are equal to the contents of location ZJ. Thus BJ is the location of a user's interruption subroutine, YJ is the location of a variable being checked, and ZJ is the location that contains the desired value for YJ.

Triggers are separated into two (2) classes:

1. Independent variable triggers, called T-stops. These triggers interrupt on values of the independent variable of integration. All T-stops <u>must</u> have YJ = 0. That is, they must have the following format in the calling sequence:

```
Pfx   BJ
PZE   ZJ
```

The logic used to execute T-stops is as follows:

Let $t_{s1}, t_{s2}, t_{s3}, \ldots\ldots t_{sk}$ be a set of values of the independent variable for which interruptions are desired.

MARK sets $t_m = \text{Min} \left\{ t_{s1}, t_{s2}, \ldots\ldots, t_{sk} \right\}$

Integration continues normally until the independent variable reaches the condition:

$$t_\eta < t_m \leqq t_\eta + 1$$

The step size is set $= (t_{\eta+1} - t_m)$ and integration is carried to $t_m$ where all the values of the variables including derivatives and end of step values are calculated and control is then transferred to the user's corresponding interruption subroutine. After control is returned from the user's interuption routine, all values are reset to station $t_{\eta+1}$ and the next $t_m$ is determined. If no other $t_m$ exists within this step, integration continues. Thus, interruption routines for all $t_m$ within a given step are executed before integration continues. There is no limitation on the number of T-stops permitted (except for machine size, of course).

2. Dependent variable triggers, called Y-stops. These triggers are interrogated at the beginning of an integration step and a value

$$l_j = y_\eta - y_j$$

is calculated and saved for each of the j Y-stops. At the end of the integration step the difference

$$r_j = y_{\eta+1} - y_j$$

is calculated and the algebraic sign of $r_j$ is compared to $l_j$:

$$\text{If} \quad \text{sgn } l_j \neq \text{sgn } r_j$$

then the condition $y = y_j$ has occurred within the integration step and a linear interpolation search procedure is executed to determine the value of the independent variable, t, such that $y = y_j$. When the $\Delta$ t calculated by the search procedure is such that

$$|\Delta t| \leq \partial_U \qquad \text{where } \partial_U = \begin{cases} 2^{-26} \text{Max} \; |H, t_{\eta+1}| & \text{for } P = 0 \\ 2^{-42} \text{Max} \; |H, t_{\eta+1}| & \text{for } P = 1 \end{cases}$$

then convergence to $t_j$ is assured. At this point all values of the dependent variable including their respective derivatives and any end of step calculations are determined and control for the corresponding Y-stop is returned to the user's interruption routine. If more than one Y-stop trigger occurs within an integration step, then the triggers are executed in the order of the smallest value of the independent variable determined for the respective Y-stops. Thus, the order of execution is determined by the independent variable. After all Y-stops within an integration step have been determined and executed, the conditions at station $t_{\eta+1}$ are restored for all dependent variables and their derivatives and end of step calculations, if any. Integration then continues normally.

Up to and including fifty (50) dependent variable triggers are permitted. However, this number may be altered by changing the symbolic card "OMAR EQU 50" in the symbolic program deck to the desired number.

It remains to define Pfx of the trigger pair. This is utilized to permit the user to render triggers "active" or "inactive". Active means that a trigger is to be interrogated and executed if necessary. Inactive means that the trigger is to be ignored.

Thus, if: $Pfx = \begin{cases} \text{PZE - trigger is active} \\ \text{MZE - trigger is inactive} \end{cases}$

The interruption routines provided by the user must terminate with either a TRA 1,4 command or a TRA 2,4 command.

TRA 1,4 is used when the interruption does not constitute a discontinuity in any of the calculations.

TRA 2,4 is used when a discontinuity exists. Under this condition a "restart" procedure is instigated by MARK by continuing beyond the discontinuity point using Runge-Kutta until a sufficient number of backward differences are determined to switch to Adams-Moulton integration.

Comments on triggers:

1. There is no limitation on how many times a trigger may be executed.

2. Care must be exercised in updating the ZJ of triggers. If the ZJ are not updated after a trigger returns control to the user, a machine loop will result, since MARK will continue to return control to the user's respective interruption routine on the basis of the current ZJ. Thus, a trigger must either be updated or rendered inactive to prevent looping.

3. In all cases where more than one trigger is to be executed at a single point ($t_j$) the triggers will be executed in order of their ascending appearance in the calling sequence.

4. Control is returned to the error return of the calling sequence whenever $t_m < (t_\eta - \partial_u)$ or when the number of Y-stops exceeds 50.

The entire list of triggers <u>must</u> be terminated with

PZE 0

This is the end of the calling sequence for MARK.

The bank of storage specified by the location HBANK is as follows:

```
        PZE m
        PZE NH
        PZE ND
HBANK   DEC H
```

PZE N,,n
DEC $t_1$
DEC $t_2$
DEC $y_1$
DEC $y_2$
.
.
.
DEC $y_n$
.
.
.
DEC $y_N$
DEC $y_1'$
DEC $y_2'$
.
.
.
DEC $y_n'$
.
.
.
DEC $y_N'$

$$\text{BSS} \quad \begin{cases} 3N + 2N\ (m + 1) & \text{for } \Phi = 0, 2 \\ 5N + 3N\ (m + 2) & \text{for } \Phi = 4 \end{cases}$$

where:

m = order of differences to be carried in the Adams Moulton mode. $m \leq 9$ (fixed point in the address portion of the word) for $\Phi = 0$.

NH = number of times to sequentially halve the step size in the Adams-Moulton mode. (fixed point in the address portion of the word.)

ND = number of times to sequentially double the step size in the Adams-Moulton mode. (fixed point in the address portion of the word.)

NOTE 1:

NH takes precedence over ND and doubling is not executed until the number of times to halve is completed. If these numbers are introduced initially in the HBANK, the procedure is commenced automatically when conversion from Runge-Kutta to Adamx-Moulton is completed. NH and ND are ignored when using the automatic variable step-size mode. NH and ND may be set by dependent variable or independent variable interruption routines in the Adams-Moulton fixed mode. Anytime control is returned to the user through an interruption routine the number of times halving and/or doubling have/has been completed is available in the decrement portion of NH and/or ND. If additional halving and/or doubling requests are entered in the address portions of NH and/or ND before a preceeding request is completed, the sum of the additional request and those remaining uncompleted will be executed.

H = nominal step-size (floating point)

N = total number of 1st order differential equations. (fixed point)

n = total number of the first n 1st order differential equations to be integrated by MARK. $n \leq N$ (fixed point)

NOTE 2:

H and N must not be altered unless a restart procedure is executed after the initial entry to MARK. n may be altered after the initial entry to MARK through an interruption routine. If n is increased, MARK restarts. Care should be exercised in setting the initial conditions corresponding to the additional equations to be integrated. If n is decreased, MARK continues normally integrating the new n set of differential equations.

$t_1$ = single precision value of the independent variable in floating point.

$t_2$ = second precision value of the independent variable in floating point. This must be zero initially if P = 0

(single precision).

$y_1$ to $y_N$ Values of the N differential equations for the dependent variables. The initial or starting values must be predetermined and set by the user. (floating point)

$y_1'$ to $y_N'$ Values of the derivatives of the dependent variables calculated and stored by the user's derivative routine (DER1, DER2). An initial pass is executed through DER1, DER2, and EOS by MARK before the integration porcess is commenced. (floating point)

## ENTRY POINTS

Provision is made through entry points to MARK to transmit certain information to MARK or to render certain information available to the user that is stored internally in MARK:

HC By using the command

```
CLA*   $HC
```

the user has direct access to the current step-size being used in the integration process. This is not necessarily the nominal step-size, H, introduced by the user in the HBANK (floating point).

NI By using the command

```
STO*   $NI
```

the user informs MARK that he desires i corrections to be performed on the predictor formula used in the Adams-Moulton fixed mode of integration. See Appendix A for descriptions of the predictor-corrector formulas being used. In the automatic step-size control mode i is automatically 1, and MARK ignores NI. Thus 1 correction is made for each prediction in this mode (fixed point).

TGLO By using the command

```
CLA*   $TGLO
```

the user has direct access to the most recent $t_{\eta+1}$ calculated, where $t_{\eta+1}$ represents the value of the independent variable at the end of an integration step (floating point).

Y The command

```
CLA*   $Y
```

gives the user access to the location of the dependent variables (single precision) in the HBANK. This appears as L(Y),1 where index register 1 set to n and counted down renders all the variables to the user (floating point).

YDOT The command

```
CLA*   $YDOT
```

performs the same function as Y for the derivatives of the dependent variables (floating point).

Y(2) The command

```
CLA*   $Y(2)
```

renders the location of the second precision part of the dependent variables available to the user (floating point)

YO
YO(2) The commands

```
CLA*   $YO
CLA*   $YO(2)
```

render the locations of the single and double precision values of the dependent variables at $t_\eta$ available to the user. $t_\eta$ represents the value of the independent variable at the beginning of an integration step (floating point).

The following symbols refer to entry points used for the automatic step-size mode. See Appendix A.

EUBAR The command

```
STO*   $EUBAR
```

stores $\overline{E}$ for use in automatic error control (floating point).

ELBAR The command

STO* $ELBAR

stores E in floating point for use with AEC.

HMAXT

STO* $HMAXT

stores maximum allowable H for AEC (automatic error control) (floating point).

HMINT

STO* $HMINT

stores minimum allowable H for AEC in floating point.

YCLOW

STO* $YCLOW

stores Y for AEC in floating point.

RGERR

CLA* $RGERR

permits access to the maximum $E_{\eta+1}$ for the user in floating point.

NOTE: EUBAR through YCLOW are consecutive locations in MARK.

SPACE REQUIRED

MARK required $3453_8 = 1835_{10}$ storage locations. No COMMON is required. The user must supply 5N + 7 + 2N (m + 1) storage locations for $\Phi = 0$, 2 or 7N + 7 + 3N (m + 2) for $\Phi = 4$. N = maximum number of differential equations; m = order of differences to be carried in the Adams-Moulton mode. $\Phi = 0$, 2 is for Runge-Kutta integration or for Adams-Moulton integration in the fixed mode. Also, whatever storage is required for the user's derivative box and trigger control must be supplied.

CODING INFORMATION

Timing: MARK will do approximately forty (40) integration intervals per second. (this time was obtained from solving a set of 14 first order differential equations)

## CHECKOUT

MARK has been checked out rather extensively using a variety of programs at the Jet Propulsion Laboratory. These programs include the JPL tracking program, a low thrust trajectory program, and a program of a general nature that solves a system of differential equations starting with five (5) equations, repeating these five (5) and adding sets of five (5) with repetition until a maximum of thirty (30) equations have been reached and integrated.

## APPENDIX A

1. The classical Runge-Kutta 4th order equations.

Let the system of equations to be solved be in the form

$$y_j' = f_j(t, y_1 y_2 . . ., y_n) \quad j = 1, 2, . . . N$$

Let $y_{j,\eta}$ be the value of $y_j$ at $t = t_\eta$ and $f_{j,\eta}$ be the derivative of $y_j$ at $t = t_\eta$. Let h be the step-size of the independent variable t. Then

$$K_1 = h\, f_j(t_\eta, y_{j,\eta})$$

$$K_2 = h\, f_j(t_\eta + 1/2\, h,\ y_{j,\eta} + \frac{K_1}{2})$$

$$K_3 = h\, f_j(t_\eta + 1/2\, h,\ y_{j,\eta} + \frac{K_2}{2})$$

$$K_4 = h\, f_j(t_\eta + \Delta t,\ y_{j,\eta} + K_3)$$

$$y_{j,\eta+1} = y_{j,\eta} + 1/6\,(K_1 + 2K_2 + 2K_3 + K_4)$$

2. The Adams-Moulton predictor-corrector equations:

Let $y_j$, $y_j'$ be defined as above. Then

$$y^p_{j,\eta+1} = y_{j,\eta} + h(a_o \nabla^o + a_1 \nabla^1 + . . . + a_m \nabla^m)\, y_j' \text{ (open type)}$$

where $\nabla$ is a backward difference operator operating on $y'_{j,\eta}$ where

$$\nabla^o y'_{j,\eta} = y'_{j,\eta}$$

The predictor coefficients $a_m$ are:

$a_o = 1.0$
$a_1 = 0.5$
$a_2 = 0.416666666$
$a_3 = 0.375$
$a_4 = 0.348611111$
$a_5 = 0.329861111$
$a_6 = 0.315591936$
$a_7 = 0.304224539$
$a_8 = 0.294868003$
$a_9 = 0.2870754484$

$$y'^{p}_{j,\eta+1} = f_j(t_\eta, y_j) \qquad j = 1, \ldots, N$$

$$y^{1}_{j,\eta+1} = y_{j,\eta} + h(b_o\nabla^o + b_1\nabla^1 + \ldots + b_m\nabla^m)\, y'^{p}_{j,\eta+1} \quad \text{(closed type)}$$

where $\nabla$ is defined as above, 1 is the first corrector application, and the corrector coefficients $b_m$ are:

| | |
|---|---|
| $b_0 = 1.0$ | $b_5 = -0.01875$ |
| $b_1 = -0.5$ | $b_6 = -0.0142691795$ |
| $b_2 = -0.0833333333$ | $b_7 = -0.0113673950$ |
| $b_3 = -0.0416666666$ | $b_8 = -0.0093565362$ |
| $b_4 = -0.0263888888$ | $b_9 = -0.0078925543$ |

NOTE: $b_{m+1} = a_{m+1} - a_m$

continuing

$$y^{2}_{j,\eta+1} = y_{j,\eta} + h(b_o\nabla^o + b_1\nabla^1 + \ldots + b_m\nabla^m)\, y'^{1}_{j,\eta+1}$$

$$y_{j,\eta+1}^{(i+1)} = y_{j,\eta+1}^{(i)} + h\,\sigma\,\epsilon^{(i)}$$

where $\sigma = \sum_{\ell=0}^{m} b_m$ ; $\epsilon^{(i)} = y'^{(i)}_{j,\eta+1} - y'^{(i-1)}_{j,\eta+1}$

i is the i th correction on the predictor formula.

3. The formula for interpolation to interruption on a dependent variable in the Adams-Moulton mode is:

$$q_j = (-1)^q \left|{\mu \atop j}\right| \text{ where } \mu = \frac{t_{\eta+1} - t_\mu}{h_c} \geq 0$$

$$\text{and } \left|{\mu \atop j}\right| = \frac{(\mu-1)(\mu-2)\ldots(\mu-j)}{(j+1)!}, \quad j = 1, \ldots, m$$

$$c_j = b_j + \sum_{i=0}^{j} q_i\, b_{j-i}, \quad j = 1, \ldots, m; \quad b_j = \text{corrector coefficients described in 2 above.}$$

$$d_j = c_j \nabla^j, \qquad j = 1, \ldots, m$$

$$y_{\ell,\mu} = y_{\ell,\eta+1} - h_\mu \left(y'_{\ell,\eta+1} + \sum_{j=1}^{m} d_j\right), \quad \ell = 1, \ldots, n$$

See Figure I-1.

4. The formula for interpolation to halve the step-size (H), dropping the subscript j, is as follows:

$$y'(\bar{t}) = \sum_{k=0}^{m} q_{-k}^{(m)}(\mu)\, y'(t_{\eta-k})$$

where:

$$q_{-k}^{(m)}(\mu) = \frac{1}{m!} \prod_{i=1}^{m} (i+\mu)$$

$$\bar{t} = t_\eta - n\ell h, \quad n = 1, 2, \ldots; \qquad \ell = 1/2, 1/3, \ldots$$

$$\mu = \frac{t_\eta - n\ell h - t_\eta}{h} = -n\ell. \quad \text{Let } \ell = \frac{1}{2} \text{ then}$$

$\mu = -\frac{1}{2}n$ where n represents the absolute value of the subscript of $\bar{t}$ in Figure I-2.

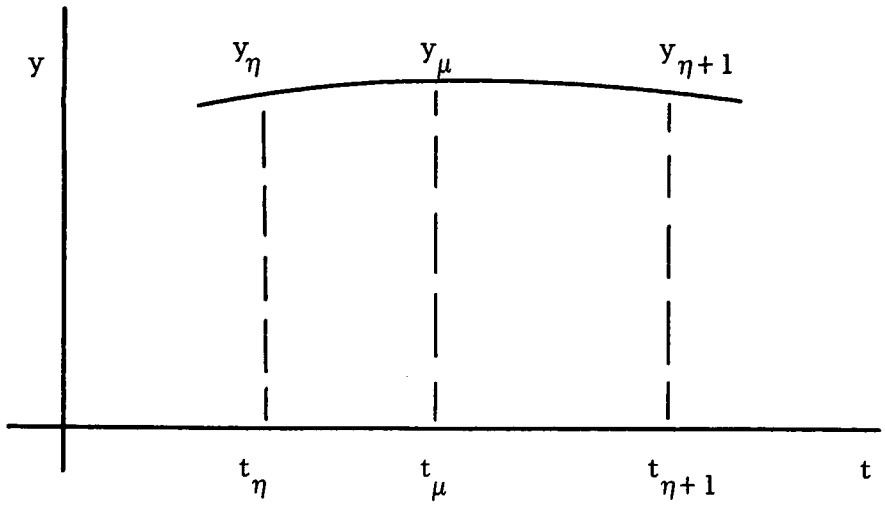


Figure I-1

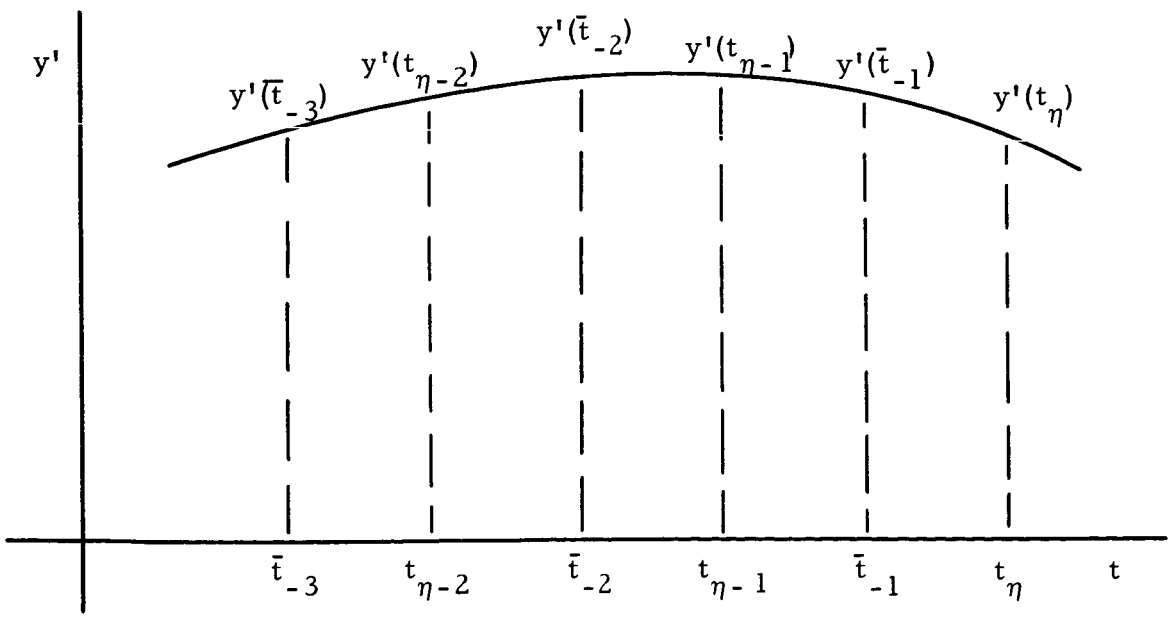


Figure I-2

In the program

$$q_o = \frac{1}{m!} \prod_{i=1}^{m} (i + \mu) \text{ and } q_{k+1} = - q_k \frac{(\mu + k)(m - k)}{(\mu + k + 1)(k + 1)}$$

where k is the absolute value of the subscript of t in Figure I-2.

5. Automatic step-size control.

$$E_{\eta + 1} = \text{MAX} \left| \frac{y^p_{j, \eta + 1} - y^c_{j, \eta + 1}}{A \; D_j} \right|$$

where

$$A = \left| \frac{a_m}{b_m + 1} \right| \quad D_j = \text{MAX} \left| y^c_{j, \eta + 1}, \underline{y} \right|$$

$E_{\eta + 1}$ represents the maximum error in any of the dependent variables in the final iterate $y_{j, \eta + 1}$ due to truncation error in the step from $t_\eta$ to $t_{\eta + 1}$. The user, through the entry points, supplied MARK with a set of values to be described as follows:

1. $\overline{E}$ - upper bound on the truncation error $E_{\eta + 1}$.
2. $\underline{E}$ - lower bound on the truncation error $E_{\eta + 1}$.
3. $h_{max}$ - maximum allowable value of the step-size.
4. $h_{min}$ - minimum allowable value of the step-size.
5. $\underline{y}$ - a constant used to prevent unnecessary reduction in h whenever $\left| y_{j, \eta + 1} \right|$ is small. $\underline{y} > 0$.

The step-size, h, is doubled, left alone, or halved depending on the following inequalities:

(1) If $E_{\eta + 1} \leq \underline{E}$ for m successive steps, the step-size, h, is set to 2 h.

(2) If $\underline{E} < E_{\eta + 1} \leq \overline{E}$, the step-size, h, is left alone.

(3) If $E_{\eta + 1} > \overline{E}$, the step-size, h, is set = 1/2 h.

The program preserves all the conditions $y_j$, $\nabla^{m+1}_j$ at $t_\eta$ and integrates to $t_{\eta + 1}$.

If (1) holds then MARK sets up the doubling procedure and integrates m + 1 more steps checking that (1) holds at each step. If (1) holds, the doubling procedure is completed and h = 2 h.

If (2) holds integration continues normally.

If (3) holds then MARK restores $y_{j,\eta}$, $\nabla^{m+1}_{j,\eta}$, $t_\eta$, $y'_{j,\eta}$. It executes the end of step box to restore those values at $t_\eta$. Finally, the halving procedure is executed and $h = \frac{1}{2}$ h. Thus, it is never necessary on the basis of error control to restart the integration procuedre in the Runge-Kutta mode. $\overline{E}$ is approximately equivalent to specifying the number of significant figures to preserve locally throughout the integration. $\overline{E}$ should normally range from $10^{-8}$ to $10^{-3}$. $\underline{y}$ may be determined by the user but should probably range from $10^{-5}$ to 1.

References:

1. Hildebrand, F. B., Introduction to Numerical Analysis, Chapter 6.
2. Ford, L. R., Differential Equations, Chapter 6.
3. Causey, R. L., Tobey Jean, RWDE2F, "Floating Point Adams-Moulton, Runge-Kutta Integration." The Ramo-Wooldridge Company, Los Angles, California, February 10, 1958.

II

# APPENDIX II

## EQUATIONS

A. Initial Values of Differential Equations of State*

$$t = 0$$
$$u = V_{To} \sin \gamma_o$$
$$v = V_{To} \cos \gamma_o \cos \beta_o$$
$$w = V_{To} \cos \gamma_o \sin \beta_o$$
$$r = h_o + R(\theta_o)$$
$$\theta = \theta_o$$
$$\phi = \phi_o$$
$$Q = 0$$
$$p_a = 0$$
$$p_H = 0$$
$$p_{QD} = 0$$

B. Differential Equations of State

Note: The state differential equations are integrated forward in time from $t_o = 0$ until the terminal time T, defined by the stopping condition $\Omega = 0$, is reached.

$$\frac{du}{dt} = \frac{V_H^2}{r} + r\Omega_e^2 \sin^2\theta + 2\Omega_e w \sin\theta + \frac{\rho S V_T}{2m} (-uC_D + C_L V_H \cos\sigma) - \frac{\mu_e}{r^2} - \frac{\mu_e J R_f^2}{r^4} (1 - 3\cos^2\theta) + \frac{T_r}{m}$$

*See Section I of this Appendix for Glossary of Symbols.

$$\frac{dv}{dt} = -\frac{uv}{r} + \frac{w^2 \cot\theta}{r} + r\Omega_e^2 \cos\theta \sin\theta$$

$$+ 2\Omega_e w \cos\theta - \frac{\rho S V_T}{2m}\left(vC_D + \frac{uv\, C_L \cos\sigma}{V_H} + \frac{w}{V_H} C_L V_T \sin\sigma\right)$$

$$+ \frac{2\mu_e J R_f^2 \cos\theta \sin\theta}{r^4} + \frac{T_\theta}{m}$$

$$\frac{dw}{dt} = -\frac{wu}{r} - \frac{vw \cot\theta}{r} - 2\Omega_e (u \sin\theta + v \cos\theta)$$

$$- \frac{\rho S V_T}{2m}\left(wC_D + \frac{wu\, C_L \cos\sigma}{V_H} - \frac{v}{V_H} C_L V_T \sin\sigma\right) + \frac{T_\phi}{m}$$

$$\frac{dr}{dt} = u$$

$$\frac{d\theta}{dt} = \frac{v}{r}$$

$$\frac{d\phi}{dt} = -\frac{w}{r \sin\theta}$$

$\frac{dQ}{dt}$, $\frac{dp_a}{dt}$, $\frac{dp_H}{dt}$, $\frac{dp_{QR}}{dt}$ : See Section G of this Appendix

where

$$V_H = (v^2 + w^2)^{1/2}$$

$$V_T = (u^2 + v^2 + w^2)^{1/2}$$

$\rho = \rho(h)$ , U.S. Standard Atmosphere, 1962

$$h = r - R(\theta)$$

$$R(\theta) = R_e - (R_e - R_p) \cos^2\theta$$

$C_L = C_L(\alpha, M, \text{Stage No.})$ , input

$C_D = C_D(\alpha, M, \text{Stage No.})$ , input

$$M = V_T/c$$

$c = c(h)$ , U.S. Standard Atmosphere, 1962

$m = m(t, \text{Stage No.})$ , input

$S = S(\text{Stage No.})$ , input

$$T_r = \left(\frac{u \cos\alpha}{V_T} + \frac{V_H \sin\alpha \cos\sigma}{V_T}\right) T_x - \left(\frac{V_H \sin\sigma}{V_T}\right) T_y$$

$$+ \left(\frac{V_H \cos\alpha \cos\sigma}{V_T} - \frac{u \sin\alpha}{V_T}\right) T_z$$

$$T_\theta = \left[\frac{v \cos\alpha}{V_T} - \sin\alpha \left(\frac{w \sin\sigma}{V_H} + \frac{vu \cos\sigma}{V_T V_H}\right)\right] T_x$$

$$+ \left(\frac{vu \sin\sigma}{V_T V_H} - \frac{w \cos\sigma}{V_H}\right) T_y$$

$$- \left[\frac{v \sin\alpha}{V_T} + \cos\alpha \left(\frac{w \sin\sigma}{V_H} + \frac{vu \cos\sigma}{V_T V_H}\right)\right] T_z$$

$$T_\phi = \left[\frac{w \cos\alpha}{V_T} + \sin\alpha \left(\frac{v \sin\sigma}{V_H} - \frac{wu \cos\sigma}{V_T V_H}\right)\right] T_x$$

$$+ \left(\frac{v \cos\sigma}{V_H} + \frac{wu \sin\sigma}{V_T V_H}\right) T_y$$

$$+ \left[\cos\alpha \left(\frac{v \sin\sigma}{V_H} - \frac{wu \cos\sigma}{V_T V_H}\right) - \frac{w \sin\alpha}{V_T}\right] T_z$$

$$T_x = T \cos i_T \cos\delta_T$$

$$T_y = T \cos i_T \sin\delta_T$$

$$T_z = T \sin i_T$$

$$T = T_{SL} + (P_e - P_\infty) A_e$$

$$P_\infty = P_\infty(h) \quad , \quad \text{U.S. Standard Atmosphere, 1962}$$

$A_e = A_e$ (Stage No. ) , input

$i_T = i_T$ (Stage No. ) , input

$\delta_T = \delta_T$ (Stage No. ) , input

$T = T$ (t, Stage No. ) , input

C. Additional Equations for Output

$$E = m\left[\frac{V_T^2}{2} - \frac{\mu_e}{r} - \frac{\mu_e J R_f^2}{r^3}\left(\frac{1}{3} - \cos^2\theta\right)\right]$$

$$q = \rho V_T^2 /2$$

$$\mu = \tan^{-1}\left(\frac{\cos\nu \sin\mu}{\cos\nu \cos\mu}\right)$$

$$\nu = \tan^{-1}\left(\frac{\sin\nu \sin\mu}{\cos\nu \sin\mu}\right)$$

$$\phi^* = \tan^{-1}\left(\frac{\sin\nu}{\cos\nu \sin\mu}\right)$$

$$\theta^* = \tan^{-1}\left(\frac{(\sin^2\nu + \cos^2\nu \sin^2\mu}{\cos\nu \cos\mu}\right), \quad 0 \le \theta^* \le 180^\circ$$

$$\gamma = \tan^{-1}\left(\frac{u}{\sqrt{v^2 + w^2}}\right), \quad -90^\circ \le \gamma \le 90^\circ$$

$$\beta = \tan^{-1}\left(\frac{w}{v}\right)$$

$$L = \rho S V_T^2 C_L /2$$

$$D = \rho S V_T^2 C_D /2$$

where

$$\cos\nu \sin\mu = \sin\theta\left(\frac{v_o}{V_{Ho}} \cos\theta_o \cos(\phi_o-\phi) - \frac{w_o}{V_{Ho}} \sin(\phi_o-\phi)\right) - \frac{v_o}{V_{Ho}} \sin\theta_o \cos\theta$$

$$\cos\nu \cos\mu = \sin\theta_o \sin\theta \cos(\phi_o-\phi) + \cos\theta_o \cos\theta$$

$$\sin\nu = \sin\theta_o \cos\theta \frac{w_o}{V_{Ho}} - \sin\theta\left(\frac{w_o}{V_{Ho}} \cos\theta_o \cos(\phi_o-\phi) + \frac{v_o}{V_{Ho}} \sin(\phi_o-\phi)\right)$$

D. <u>Initial Values of Adjoint Differential Equations</u>

<u>Note:</u> A set of adjoint differential equations, as defined in Section E of this Appendix, is integrated backward in time from $t = T$ to $t = 0$ for the pay-off function $\Phi$ and for each terminal constraint function $\psi_1, \psi_2, \ldots \psi_p$, $0 \leq p \leq 8$. These sets of equations are distinguished by their respective "initial" condition at $t = T$. The initial conditions are defined here for the set of equations associated with $\Phi$, or, to be consistent with the notation in Part I of this report, with $\Phi\Omega$. The initial conditions for each $\psi_j\Omega$ set of adjoints is defined in a similar manner.

$$\lambda_{\Phi\Omega u} = \left(\frac{\partial\Phi}{\partial u} - \frac{\dot{\Phi}}{\dot{\Omega}} \frac{\partial\Omega}{\partial u}\right)_{t=T}$$

$$\lambda_{\Phi\Omega v} = \left(\frac{\partial\Phi}{\partial v} - \frac{\dot{\Phi}}{\dot{\Omega}} \frac{\partial\Omega}{\partial v}\right)_{t=T}$$

$$\lambda_{\Phi\Omega w} = \left(\frac{\partial\Phi}{\partial w} - \frac{\dot{\Phi}}{\dot{\Omega}} \frac{\partial\Omega}{\partial w}\right)_{t=T}$$

$$\lambda_{\Phi\Omega r} = \left(\frac{\partial\Phi}{\partial r} - \frac{\dot{\Phi}}{\dot{\Omega}}\frac{\partial\Omega}{\partial r}\right)_{t=T}$$

$$\lambda_{\Phi\Omega\theta} = \left(\frac{\partial\Phi}{\partial\theta} - \frac{\dot{\Phi}}{\dot{\Omega}}\frac{\partial\Omega}{\partial\theta}\right)_{t=T}$$

$$\lambda_{\Phi\Omega\phi} = \left(\frac{\partial\Phi}{\partial\phi} - \frac{\dot{\Phi}}{\dot{\Omega}}\frac{\partial\Omega}{\partial\phi}\right)_{t=T}$$

$$\lambda_{\Phi\Omega Q} = \left(\frac{\partial\Phi}{\partial Q} - \frac{\dot{\Phi}}{\dot{\Omega}}\frac{\partial\Omega}{\partial Q}\right)_{t=T}$$

$$\lambda_{\Phi\Omega p_a} = \left(\frac{\partial\Phi}{\partial p_a} - \frac{\dot{\Phi}}{\dot{\Omega}}\frac{\partial\Omega}{\partial p_a}\right)_{t=T}$$

$$\lambda_{\Phi\Omega p_H} = \left(\frac{\partial\Phi}{\partial p_H} - \frac{\dot{\Phi}}{\dot{\Omega}}\frac{\partial\Omega}{\partial p_H}\right)_{t=T}$$

$$\lambda_{\Phi\Omega p_{QD}} = \left(\frac{\partial\Phi}{\partial p_{QD}} - \frac{\dot{\Phi}}{\dot{\Omega}}\frac{\partial\Omega}{\partial p_{QD}}\right)_{t=T}$$

where

$\Phi(t,u,v\ w\ r,\theta,\phi,Q,p_a,p_H,p_{QD})$ is payoff function (or for $\lambda_{\psi_j\Omega}$, constraint function $\psi_j$)

$\Omega(t,u\ v\ w,r,\theta,\phi,Q,p_a,p_H,p_{QD})$ is stopping condition

$$\dot{\Phi} = \left(\frac{\partial\Phi}{\partial t} + \frac{\partial\Phi}{\partial u}\frac{du}{dt} + \frac{\partial\Phi}{\partial v}\frac{dv}{dt} + \frac{\partial\Phi}{\partial w}\frac{dw}{dt} + \frac{\partial\Phi}{\partial r}\frac{dr}{dt} + \frac{\partial\Phi}{\partial\theta}\frac{d\theta}{dt} + \frac{\partial\Phi}{\partial\phi}\frac{d\phi}{dt} + \frac{\partial\Phi}{\partial Q}\frac{dQ}{dt} + \frac{\partial\Phi}{\partial p_a}\frac{dp_a}{dt} + \frac{\partial\Phi}{\partial p_H}\frac{dp_H}{dt} + \frac{\partial\Phi}{\partial p_{QD}}\frac{dp_{QD}}{dt}\right)_{t=T}$$

$$\dot{\Omega} = \left(\frac{\partial\Omega}{\partial t} + \frac{\partial\Omega}{\partial u}\frac{du}{dt} + \frac{\partial\Omega}{\partial v}\frac{dv}{dt} + \frac{\partial\Omega}{\partial w}\frac{dw}{dt} + \frac{\partial\Omega}{\partial r}\frac{dr}{dt} + \frac{\partial\Omega}{\partial\theta}\frac{d\theta}{dt} + \frac{\partial\Omega}{\partial\phi}\frac{d\phi}{dt} + \frac{\partial\Omega}{\partial Q}\frac{dQ}{dt} + \frac{\partial\Omega}{\partial p_a}\frac{dp_a}{dt} + \frac{\partial\Omega}{\partial p_H}\frac{dp_H}{dt} + \frac{\partial\Omega}{\partial p_{QD}}\frac{dp_{QD}}{dt}\right)_{t=T}$$

Partial derivatives are computed from the formula $\frac{df}{dh} = \frac{f(x+h) - f(x-h)}{2h}$.

The h's used for $t, u, v, w, r, \theta, \phi, Q, p_a, p_H$, and $p_{QD}$ are .1, 1, 1, 1, 50, .01, .01, 1, 1, 1, and 1 respectively.

E. Adjoint Differential Equations

Note: This set of equations is integrated for each set of initial conditions, as defined in Section D, to give the corresponding adjoint solutions. The subscript $\Phi\Omega$, or $\psi_j\Omega$ as appropriate, has been omitted from each $\lambda$ in the equations as presented here.

$$\dot{\lambda}_u = \frac{\partial \dot{u}}{\partial u}\lambda_u + \frac{\partial \dot{v}}{\partial u}\lambda_v + \frac{\partial \dot{w}}{\partial u}\lambda_w + \frac{\partial \dot{r}}{\partial u}\lambda_r + \frac{\partial \dot{\theta}}{\partial u}\lambda_\theta + \frac{\partial \dot{\phi}}{\partial u}\lambda_\phi + \frac{\partial \dot{Q}}{\partial u}\lambda_Q + \frac{\partial p_a}{\partial u}\lambda_{pa} + \frac{\partial \dot{p}_H}{\partial u}\lambda_{pH} + \frac{\partial \dot{p}_{QD}}{\partial u}\lambda_{p_{QD}}$$

$$\dot{\lambda}_v = \frac{\partial \dot{u}}{\partial v}\lambda_u + \frac{\partial \dot{v}}{\partial v}\lambda_v + \frac{\partial \dot{w}}{\partial v}\lambda_w + \frac{\partial \dot{r}}{\partial v}\lambda_r + \frac{\partial \dot{\theta}}{\partial v}\lambda_\theta + \frac{\partial \dot{\phi}}{\partial v}\lambda_\phi + \frac{\partial \dot{Q}}{\partial v}\lambda_Q + \frac{\partial \dot{p}_a}{\partial v}\lambda_{pa} + \frac{\partial \dot{p}_H}{\partial v}\lambda_{pH} + \frac{\partial \dot{p}_{QD}}{dv}\lambda_{p_{QD}}$$

$$\dot{\lambda}_w = \frac{\partial \dot{u}}{\partial w}\lambda_u + \frac{\partial \dot{v}}{\partial w}\lambda_v + \frac{\partial \dot{w}}{\partial w}\lambda_w + \frac{\partial \dot{r}}{\partial w}\lambda_r + \frac{\partial \dot{\theta}}{\partial w}\lambda_\theta + \frac{\partial \dot{\phi}}{\partial w}\lambda_\phi + \frac{\partial \dot{Q}}{\partial w}\lambda_Q + \frac{\partial \dot{p}_a}{\partial w}\lambda_{pa} + \frac{\partial \dot{p}_H}{\partial w}\lambda_{pH} + \frac{\partial \dot{p}_{QD}}{\partial w}\lambda_{p_{QD}}$$

$$\dot{\lambda}_r = \frac{\partial \dot{u}}{\partial r}\lambda_u + \frac{\partial \dot{v}}{\partial r}\lambda_v + \frac{\partial \dot{w}}{\partial r}\lambda_w + \frac{\partial \dot{r}}{\partial r}\lambda_r + \frac{\partial \dot{\theta}}{\partial r}\lambda_\theta + \frac{\partial \dot{\phi}}{\partial r}\lambda_\phi$$

$$+ \frac{\partial \dot{Q}}{\partial r}\lambda_Q + \frac{\partial \dot{p}_a}{\partial r}\lambda_{pa} + \frac{\partial \dot{p}_H}{\partial r}\lambda_{pH} + \frac{\partial \dot{p}_{QD}}{\partial r}\lambda_{p_{QD}}$$

$$\dot{\lambda}_\theta = \frac{\partial \dot{u}}{\partial \theta}\lambda_u + \frac{\partial \dot{v}}{\partial \theta}\lambda_v + \frac{\partial \dot{w}}{\partial \theta}\lambda_w + \frac{\partial \dot{r}}{\partial \theta}\lambda_r + \frac{\partial \dot{\theta}}{\partial \theta}\lambda_\theta + \frac{\partial \dot{\phi}}{\partial \theta}\lambda_\phi$$

$$+ \frac{\partial \dot{Q}}{\partial \theta}\lambda_Q + \frac{\partial \dot{p}_a}{\partial \theta}\lambda_{pa} + \frac{\partial \dot{p}_H}{\partial \theta}\lambda_{pH} + \frac{\partial \dot{p}_{QD}}{\partial \theta}\lambda_{p_{QD}}$$

$$\dot{\lambda}_\phi = \dot{\lambda}_Q = \dot{\lambda}_{p_a} = \dot{\lambda}_{p_H} = \dot{\lambda}_{p_{QD}} = 0$$

where

$$\frac{\partial \dot{u}}{\partial u} = \frac{\rho S}{2m}\left[\frac{V_H u \cos\sigma}{V_T}\left(C_L + M\frac{\partial C_L}{\partial M}\right) - C_D V_T - \frac{u^2}{V_T}\left(C_D + M\frac{\partial C_D}{\partial M}\right)\right] + \frac{1}{m}\frac{\partial T_r}{\partial u}$$

$$\frac{\partial \dot{v}}{\partial u} = -\frac{v}{r} - \frac{\rho S}{2m}\left\{\frac{uv}{V_T}\left(C_D + M\frac{\partial C_D}{\partial M}\right) + \frac{v\cos\sigma}{V_H}\left[C_L V_T + \frac{u^2}{V_T}\left(C_L + M\frac{\partial C_L}{\partial M}\right)\right] + \frac{u\,w\sin\sigma}{V_H}\left(2C_L + M\frac{\partial C_L}{\partial M}\right)\right\} + \frac{1}{m}\frac{\partial T_\theta}{\partial u}$$

$$\frac{\partial \dot{w}}{\partial u} = -\frac{w}{r} - 2\Omega_e \sin\theta - \frac{\rho S}{2m}\left\{\frac{uw}{V_T}\left(C_D + M\frac{\partial C_D}{\partial M}\right) + \frac{w\cos\sigma}{V_H}\left[C_L V_T + \frac{u^2}{V_T}\left(C_L + M\frac{\partial C_L}{\partial M}\right)\right] - \frac{u\,v\sin\sigma}{V_H}\left(2C_L + M\frac{\partial C_L}{\partial M}\right)\right\} + \frac{1}{m}\frac{\partial T_\phi}{\partial u}$$

$$\frac{\partial \dot{r}}{\partial u} = 1$$

$$\frac{\partial \dot{\theta}}{\partial u} = \frac{\partial \dot{\phi}}{\partial u} = 0$$

$\frac{\partial \dot{Q}}{\partial u}$, $\frac{\partial \dot{p}_a}{\partial u}$, $\frac{\partial \dot{p}_H}{\partial u}$, $\frac{\partial \dot{p}_{QD}}{\partial u}$ : See Section G of this Appendix

$$\frac{\partial \dot{u}}{\partial v} = \frac{2v}{r} + \frac{\rho S}{2m}\left\{\frac{v\cos\sigma}{V_H}\left[C_L V_T + \frac{V_H^2}{V_T}\left(C_L + M\frac{\partial C_L}{\partial M}\right)\right] - \frac{uv}{V_T}\left(C_D + M\frac{\partial C_D}{\partial M}\right)\right\} + \frac{1}{m}\frac{\partial T_r}{\partial v}$$

$$\frac{\partial \dot{v}}{\partial v} = -\frac{u}{r} - \frac{\rho S}{2m}\left[C_D V_T + \frac{v^2}{V_T}\left(C_D + M\frac{\partial C_D}{\partial M}\right)\right] + \frac{u\cos\sigma}{V_H}\left[\frac{w^2}{V_H^2}C_L V_T + \frac{v^2}{V_T}\left(C_L + M\frac{\partial C_L}{\partial M}\right)\right] + \frac{vw\sin\sigma}{V_H}\left\{2C_L + M\frac{\partial C_L}{\partial M} - \frac{C_L V_T^2}{V_H^2}\right\} + \frac{1}{m}\frac{\partial T_\theta}{\partial v}$$

$$\frac{\partial \dot{w}}{\partial v} = -\frac{w \cot \theta}{r} - 2\Omega_e \cos \theta - \frac{\rho S}{2m} \left\{ \frac{vw}{V_T} \left( C_D + M \frac{\partial C_D}{\partial M} \right) \right.$$

$$+ \frac{u\ v\ w \cos \sigma}{V_H} \left[ \frac{1}{V_T} \left( C_L + M \frac{\partial C_L}{\partial M} \right) - \frac{C_L\ V_T}{V_H^2} \right]$$

$$\left. - \frac{\sin \sigma}{V_H} \left[ \frac{w^2}{V_H^2} C_L\ V_T^2 + v^2 \left( 2C_L + M \frac{\partial C_L}{\partial M} \right) \right] \right\} + \frac{1}{m} \frac{\partial T_\phi}{\partial v}$$

$$\frac{\partial \dot{\theta}}{\partial v} = \frac{1}{r}$$

$$\frac{\partial \dot{r}}{\partial v} = \frac{\partial \dot{\phi}}{\partial v} = 0$$

$\frac{\partial \dot{Q}}{\partial v}$, $\frac{\partial \dot{p}_a}{\partial v}$, $\frac{\partial \dot{p}_H}{\partial v}$, $\frac{\partial \dot{p}_{QD}}{\partial v}$ : See Section G of this Appendix.

$$\frac{\partial \dot{u}}{\partial w} = \frac{2w}{r} + 2\Omega_e \sin \theta + \frac{\rho S}{2m} \left\{ \frac{w \cos \sigma}{V_H} \left[ C_L\ V_T + \frac{V_H^2}{V_T} \left( C_L \right.\right.\right.$$

$$\left.\left.\left. + M \frac{\partial C_L}{\partial M} \right) \right] - \frac{uw}{V_T} \left( C_D + M \frac{\partial C_D}{\partial M} \right) \right\} + \frac{1}{m} \frac{\partial T_r}{\partial w}$$

$$\frac{\partial \dot{v}}{\partial w} = \frac{2\ w \cot \theta}{r} + 2\Omega_e \cos \theta - \frac{\rho S}{2m} \left\{ \frac{vw}{V_T} \left( C_D + M \frac{\partial C_D}{\partial M} \right) \right.$$

$$+ \frac{u\ v\ w \cos \sigma}{V_H} \left[ \frac{1}{V_T} \left( C_L + M \frac{\partial C_L}{\partial M} \right) - \frac{C_L\ V_T}{V_H^2} \right]$$

$$\left. + \frac{\sin \sigma}{V_H} \left[ \frac{v^2 V_T^2 C_L}{V_H^2} + w^2 \left( 2\ C_L + M \frac{\partial C_L}{\partial M} \right) \right] \right\} + \frac{1}{m} \frac{\partial T_\theta}{\partial w}$$

$$\frac{\partial \dot{w}}{\partial w} = -\frac{u}{r} - \frac{v \cot\theta}{r} - \frac{\rho S}{2m}\left\{C_D V_T + \frac{w^2}{V_T}\left(C_D + M\frac{\partial C_D}{\partial M}\right)\right.$$

$$+ \frac{u\cos\sigma}{V_H}\left[\frac{v^2 C_L V_T}{V_H^2} + \frac{w^2}{V_T}\left(C_L + M\frac{\partial C_L}{\partial M}\right)\right]$$

$$- \frac{vw\sin\sigma}{V_H}\left(2C_L + M\frac{\partial C_L}{\partial M} - \frac{C_L V_T^2}{V_H^2}\right) + \frac{1}{m}\frac{\partial T_\phi}{\partial w}$$

$$\frac{\partial \dot{r}}{\partial w} = \frac{\partial \dot{\theta}}{\partial w} = 0$$

$$\frac{\partial \dot{\phi}}{\partial w} = \frac{1}{r\sin\theta}$$

$\frac{\partial \dot{Q}}{\partial w}$, $\frac{\partial \dot{p}_a}{\partial w}$, $\frac{d\dot{p}_H}{\partial w}$, $\frac{d\dot{p}_{QD}}{\partial D}$ : See Section G of this Appendix

$$\frac{\partial \dot{u}}{\partial r} = -\frac{V_H^2}{r^2} + \Omega_e^2 \sin^2\theta + \frac{S}{2m}\left[-uV_T\left(C_D\frac{d\rho}{dh} - \frac{\rho M}{c}\frac{dc}{dh}\frac{\partial C_D}{\partial M}\right)\right.$$

$$\left. + V_H V_T \cos\sigma\left(C_L\frac{d\rho}{dh} - \frac{\rho M}{c}\frac{dc}{dh}\frac{\partial C_L}{\partial M}\right)\right] + \frac{2\mu_e}{r^3}$$

$$+ \frac{4\mu_e J R_f^2}{r^5}(1 - 3\cos^2\theta) + \frac{1}{m}\frac{\partial T_r}{\partial r}$$

$$\frac{\partial \dot{v}}{\partial r} = \frac{uv}{r^2} - \frac{w^2\cot\theta}{r^2} + \Omega_e^2 \sin\theta\cos\theta$$

$$- \frac{S}{2m}\left[vV_T\left(C_D\frac{d\rho}{dh} - \frac{\rho M}{c}\frac{dc}{dh}\frac{\partial C_D}{\partial M}\right)\right.$$

$$\left. + \left(\frac{uvV_T\cos\sigma}{V_H} + \frac{wV_T^2\sin\sigma}{V_H}\right)\left(C_L\frac{d\rho}{dh} - \frac{\rho M}{c}\frac{dc}{dh}\frac{\partial C_L}{\partial M}\right)\right]$$

$$- \frac{8\mu_e J R_f^2 \cos\theta\sin\theta}{r^5} + \frac{1}{m}\frac{\partial T_\theta}{\partial r}$$

$$\frac{\partial \dot{w}}{\partial r} = \frac{uw}{r^2} + \frac{vw \cot\theta}{r^2} - \frac{S}{2m} \left[ wV_T \left( C_D \frac{d\rho}{dh} - \frac{\rho M}{c} \frac{dc}{dh} \frac{\partial C_D}{\partial M} \right) + \left( \frac{uw\, V_T \cos\sigma}{V_H} - \frac{vV_T^2 \sin\sigma}{V_H} \right) \left( C_L \frac{d\rho}{dh} - \frac{\rho M}{c} \frac{dc}{dh} \frac{\partial C_L}{\partial M} \right) \right] + \frac{1}{m} \frac{\partial T_\phi}{\partial r}$$

$$\frac{\partial \dot{r}}{\partial r} = 0$$

$$\frac{\partial \dot{\theta}}{\partial r} = -\frac{v}{r^2}$$

$$\frac{\partial \dot{\phi}}{\partial r} = -\frac{w}{r^2 \sin\theta}$$

$\frac{\partial \dot{Q}}{\partial r}$, $\frac{\partial \dot{p}_a}{\partial r}$, $\frac{\partial \dot{p}_H}{\partial r}$, $\frac{\partial \dot{p}_{QD}}{\partial r}$ : See Section G of this Appendix.

$$\frac{\partial \dot{u}}{\partial \theta} = 2\Omega_e^2 r \cos\theta \sin\theta + 2\Omega_e w \cos\theta - \frac{6\mu_e J R_f^2}{r^4} \sin\theta \cos\theta - \frac{S}{2m} \left[ -uV_T \left( C_D \frac{d\rho}{dh} - \frac{\rho M}{c} \frac{dc}{dh} \frac{\partial C_D}{\partial M} \right) + V_H V_T \cos\sigma \left( C_L \frac{d\rho}{dh} - \frac{\rho M}{c} \frac{dc}{dh} \frac{\partial C_L}{\partial M} \right) \right] \frac{dR}{d\theta} - \frac{1}{m} \frac{\partial T_r}{\partial r} \frac{dR}{d\theta}$$

$$\frac{\partial \dot{v}}{\partial \theta} = -\frac{w^2}{r \sin^2\theta} + \Omega_e^2 r \left(\cos^2\theta - \sin^2\theta\right) - 2\Omega_e w \sin\theta$$

$$+ \frac{2\mu_e J R_f^2}{r^4} \left(\cos^2\theta - \sin^2\theta\right) + \frac{S}{2m} \left[ vV_T \left[ C_D \frac{d\rho}{dh} \right.\right.$$

$$\left. - \frac{\rho M}{c} \frac{dc}{dh} \frac{\partial C_D}{\partial M} \right] + \frac{uvV_T \cos\sigma}{V_H} + \frac{wV_T^2 \sin\sigma}{V_H} \left[ C_L \frac{d\rho}{dh} \right.$$

$$\left.\left. - \frac{\rho M}{c} \frac{dc}{dh} \frac{\partial C_L}{M} \right] \right] \frac{dR}{d\theta} - \frac{1}{m} \frac{\partial T_\theta}{\partial r} \frac{dR}{d\theta}$$

$$\frac{\partial \dot{w}}{\partial \theta} = \frac{vw}{r \sin^2\theta} - 2\Omega_e u \cos\theta + 2\Omega_e v \sin\theta + \frac{S}{2m} \left[ wV_T \left[ C_D \frac{d\rho}{dh} \right.\right.$$

$$\left. - \frac{\rho M}{c} \frac{dc}{dh} \frac{\partial C_D}{\partial M} \right] + \left( \frac{uwV_T \cos\sigma}{V_H} - \frac{vV_T^2 \sin\sigma}{V_H} \right) \left[ C_L \frac{d\rho}{dh} \right.$$

$$\left.\left. - \frac{\rho M}{c} \frac{\partial C_L}{\partial M} \frac{dc}{dh} \right] \right] \frac{dR}{d\theta} - \frac{1}{m} \frac{\partial T_\phi}{\partial r} \frac{dR}{d\theta}$$

$$\frac{\partial \dot{r}}{\partial \theta} = \frac{\partial \dot{\theta}}{\partial \theta} = 0$$

$$\frac{\partial \dot{\phi}}{\partial \theta} = -\frac{w \cot\theta}{r \sin\theta}$$

$\frac{\partial \dot{Q}}{\partial \theta}$, $\frac{\partial \dot{p}_a}{\partial \theta}$, $\frac{\partial \dot{p}_H}{\partial \theta}$, $\frac{\partial \dot{p}_{QD}}{\partial \theta}$ : See Section G of this Appendix

$$\frac{\partial \dot{u}}{\partial \phi} = \frac{\partial \dot{v}}{\partial \phi} = \frac{\partial \dot{w}}{\partial \phi} = \frac{\partial \dot{r}}{\partial \phi} = \frac{\partial \dot{\theta}}{\partial \phi} = \frac{\partial \dot{\phi}}{\partial \phi} = 0$$

$\frac{\partial \dot{Q}}{\partial \phi}$, $\frac{\partial \dot{p}_H}{\partial \phi}$, $\frac{\partial \dot{p}_H}{\partial \phi}$, $\frac{\partial \dot{p}_{QD}}{\partial \phi}$ : See Section G of this Appendix

and where:

$$\frac{\partial T_r}{\partial u} = \frac{V_H^2}{V_T^3} (T_x \cos \alpha - T_z \sin \alpha) - \frac{uV_H}{V_T^3} (T_x \cos \sigma \sin \alpha - T_y \sin \sigma$$

$$+ T_z \cos \sigma \cos \alpha)$$

$$\frac{\partial T_\theta}{\partial u} = - \frac{uv}{V_T^3} (T_x \cos \alpha - T_z \sin \alpha) - \frac{vV_H}{V_T^3} (T_x \cos \sigma \sin \alpha - T_y \sin \sigma$$

$$+ T_z \cos \sigma \cos \alpha)$$

$$\frac{\partial T_\phi}{\partial u} = - \frac{uw}{V_T^3} (T_x \cos \alpha - T_z \sin \alpha) - \frac{wV_H}{V_T^3} (T_x \cos \sigma \sin \alpha - T_y \sin \sigma$$

$$+ T_z \cos \sigma \cos \alpha)$$

$$\frac{\partial T_r}{\partial v} = - \frac{uv}{V_T^3} (T_x \cos \alpha - T_z \sin \alpha) + \frac{u^2 v}{V_T^3 V_H} (T_x \cos \sigma \sin \alpha - T_y \sin \sigma$$

$$+ T_z \cos \sigma \cos \alpha)$$

$$\frac{\partial T_\theta}{\partial v} = \frac{u^2+w^2}{V_T^3} (T_x \cos \alpha - T_z \sin \alpha) + \frac{vw}{V_H^3} (T_x \sin \sigma \sin \alpha + T_y \cos \sigma$$

$$+ T_z \sin \sigma \cos \alpha) - u \left( \frac{(V_H V_T)^2 - v^2 (V_H^2 + V_T^2)}{(V_H V_T)^3} \right) (T_x \cos \sigma \sin \alpha$$

$$- T_y \sin \sigma + T_z \cos \sigma \cos \alpha)$$

$$\frac{\partial T_\phi}{\partial v} = -\frac{vw}{V_T^3}(T_x \cos\alpha - T_z \sin\alpha) + \frac{w^2}{V_H^3}(T_x \sin\sigma \sin\alpha + T_y \cos\sigma$$

$$+ T_z \sin\sigma \cos\alpha) + \frac{uvw}{(V_T V_H)^3}\left(V_T^2 + V_H^2\right)(T_x \cos\sigma \sin\alpha$$

$$-T_y \sin\sigma + T_z \cos\sigma \cos\alpha)$$

$$\frac{\partial T_r}{\partial w} = -\frac{uw}{V_T^3}(T_x \cos\alpha - T_z \sin\alpha) + \frac{u^2 w}{V_T^3 V_H}(T_x \cos\sigma \sin\alpha$$

$$-T_y \sin\sigma + T_z \cos\sigma \cos\alpha)$$

$$\frac{\partial T_\theta}{\partial w} = -\frac{vw}{V_T^3}(T_x \cos\alpha - T_z \sin\alpha) - \frac{v^2}{V_H^3}(T_x \sin\sigma \sin\alpha + T_y \cos\sigma$$

$$+ T_z \sin\sigma \cos\alpha) + \frac{uvw}{(V_T V_H)^3}(V_T^2 + V_H^2)(T_x \cos\sigma \sin\alpha$$

$$- T_y \sin\sigma + T_z \cos\sigma \cos\alpha)$$

$$\frac{\partial T_\phi}{\partial w} = \frac{u^2 + v^2}{V_T^3}(T_x \cos\alpha - T_z \sin\alpha) - \frac{vw}{V_H^3}(T_x \sin\sigma \sin\alpha + T_y \cos\sigma$$

$$+ T_z \sin\sigma \cos\alpha) - \frac{u}{(V_T V_H)^3}\Big[(V_T V_H)^2$$

$$- w^2(V_H^2 + V_T^2)\Big](T_x \cos\sigma \sin\alpha - T_y \sin\sigma + T_z \cos\sigma \cos\alpha)$$

$$\frac{\partial T_r}{\partial r} = \frac{u}{V_T}\left(\frac{\partial T_x}{\partial h}\cos\alpha - \frac{\partial T_z}{\partial h}\sin\alpha\right) + \frac{V_H}{V_T}\left(\frac{\partial T_x}{\partial h}\cos\sigma\sin\alpha - \frac{\partial T_y}{\partial h}\sin\sigma + \frac{\partial T_z}{\partial h}\cos\sigma\cos\alpha\right)$$

$$\frac{\partial T_\theta}{\partial r} = \frac{v}{V_T}\left(\frac{\partial T_x}{\partial h}\cos\alpha - \frac{\partial T_z}{\partial h}\sin\alpha\right) - \frac{w}{V_H}\left(\frac{\partial T_x}{\partial h}\sin\sigma\sin\alpha + \frac{\partial T_y}{\partial h}\cos\sigma + \frac{\partial T_z}{\partial h}\sin\sigma\cos\alpha\right) - \frac{uv}{V_H V_T}\left(\frac{\partial T_x}{\partial h}\cos\sigma\sin\alpha - \frac{\partial T_y}{\partial h}\sin\sigma + \frac{\partial T_z}{\partial h}\cos\sigma\cos\alpha\right)$$

$$\frac{\partial T_\phi}{\partial r} = \frac{w}{V_T}\left(\frac{\partial T_x}{\partial h}\cos\alpha - \frac{\partial T_z}{\partial h}\sin\alpha\right) + \frac{v}{V_H}\left(\frac{\partial T_x}{\partial h}\sin\sigma\sin\alpha + \frac{\partial T_y}{\partial h}\cos\sigma + \frac{\partial T_z}{\partial h}\sin\sigma\cos\alpha\right) - \frac{uw}{V_H V_T}\left(\frac{\partial T_x}{\partial h}\cos\sigma\sin\alpha - \frac{\partial T_y}{\partial h}\sin\sigma + \frac{\partial T_z}{\partial h}\cos\sigma\cos\alpha\right)$$

$$\frac{\partial T_x}{\partial h} = -A_e\cos i_T\cos\delta_T\frac{dP_\infty}{dh}$$

$$\frac{\partial T_y}{\partial h} = -A_e\cos i_T\sin\delta_T\frac{dP_\infty}{dh}$$

$$\frac{\partial T_z}{\partial h} = -A_e\sin i_T\frac{dP_\infty}{dh}$$

F. Impulse Response Functions

Note: A set of impulse-response functions is computed for the payoff function, $\Phi$, and for each of the terminal constraint functions $\psi_1, \psi_2, \ldots \psi_p$, $0 \leq p \leq 8$. The equations are written here for $\Phi$, where the $\lambda_{\Phi\Omega}$ are solutions to the adjoint equations, Section E, when the initial conditions, Section D, are computed on the basis of derivatives of the payoff function, $\Phi$. The impulse-response functions for the $\psi_j$ are obtained in similar manner utilizing the corresponding sets of adjoint solutions. Terms for which the partial derivatives with respect to the control variable, $\alpha$ or $\sigma$, are zero have been omitted from these equations*.

$$G'\lambda_{\Phi\Omega\alpha} = \frac{\partial\dot{u}}{\partial\alpha}\lambda_{\Phi\Omega u} + \frac{\partial\dot{v}}{\partial\alpha}\lambda_{\Phi\Omega v} + \frac{\partial\dot{w}}{\partial\alpha}\lambda_{\Phi\Omega w} + \frac{\partial\dot{p}_a}{\partial\alpha} I_{\Phi\Omega p_a}$$

$$G'\lambda_{\Phi\Omega\sigma} = \frac{\partial\dot{u}}{\partial\sigma}\lambda_{\Phi\Omega u} + \frac{\partial\dot{v}}{\partial\sigma}\lambda_{\Phi\Omega v} + \frac{\partial\dot{w}}{\partial\sigma}\lambda_{\Phi\Omega w}$$

where

$$\frac{\partial\dot{u}}{\partial\alpha} = -\frac{\rho S}{2m}\left(uV_T\frac{\partial C_D}{\partial\alpha} - V_H V_T \cos\sigma \frac{\partial C_L}{\partial\alpha}\right) + \frac{1}{m}\frac{\partial T_r}{\partial\alpha}$$

$$\frac{\partial\dot{v}}{\partial\alpha} = -\frac{\rho S}{2m}\left(vV_T\frac{\partial C_D}{\partial\alpha} + uv\,V_T\frac{\cos\sigma}{V_H}\frac{\partial C_L}{\partial\alpha} + wV_T^2\frac{\sin\sigma}{V_H}\frac{\partial C_L}{\partial\alpha}\right) + \frac{1}{m}\frac{\partial T_\theta}{\partial\alpha}$$

*In matrix notation,

$$G'\lambda_{\Phi\Omega} = \begin{vmatrix} G'\lambda_{\Phi\Omega\alpha} \\ G'\lambda_{\Phi\Omega\sigma} \end{vmatrix}$$

$$\frac{\partial \dot{w}}{\partial \alpha} = -\frac{\rho S}{2m}\left(wV_T \frac{\partial C_D}{\partial \alpha} + uw\, V_T \frac{\cos\sigma}{V_H}\frac{\partial C_L}{\partial \alpha} - vV_T^2 \frac{\sin\sigma}{V_H}\frac{\partial C_L}{\partial \alpha}\right) + \frac{1}{m}\frac{\partial T_\phi}{\partial \alpha}$$

$$\frac{\partial \dot{u}}{\partial \sigma} = -\frac{\rho S}{2m}\left(V_H\, C_L\, V_T\right)\sin\sigma + \frac{1}{m}\frac{\partial T_r}{\partial \sigma}$$

$$\frac{\partial \dot{v}}{\partial \sigma} = \frac{\rho S}{2m}\left(C_L\, V_T\, uv \frac{\sin\sigma}{V_H} - C_L\, V_T^2\, w \frac{\cos\sigma}{V_H}\right) + \frac{1}{m}\frac{\partial T_\theta}{\partial \sigma}$$

$$\frac{\partial \dot{w}}{\partial \sigma} = \frac{\rho S}{2m}\left(C_L\, V_T\, uw \frac{\sin\sigma}{V_H} + C_L\, V_T^2\, v \frac{\cos\sigma}{V_H}\right) + \frac{1}{m}\frac{\partial T_\theta}{\partial \sigma}$$

$\frac{\partial \dot{p}_a}{\partial \alpha}$ : See Section G of this Appendix

$$\frac{\partial T_r}{\partial \alpha} = \frac{T_x}{V_T}\left(V_H \cos\alpha \cos\sigma - u \sin\alpha\right) - \frac{T_z}{V_T}\left(u \cos\alpha + V_H \cos\sigma \sin\alpha\right)$$

$$\frac{\partial T_\theta}{\partial \alpha} = -\frac{T_x}{V_T\, V_H}\left[v\, V_H \sin\alpha + \cos\alpha\left(w\, V_T \sin\sigma + uv \cos\sigma\right)\right] + \frac{T_z}{V_T\, V_H}\left[\sin\alpha\left(w\, V_T \sin\sigma + uv \cos\sigma\right) - v\, V_H \cos\alpha\right]$$

$$\frac{\partial T_\phi}{\partial \alpha} = \frac{T_x}{V_T\, V_H}\left[\cos\alpha\left(v\, V_T \sin\sigma - u\, w \cos\sigma\right) - w\, V_H \sin\alpha\right] - \frac{T_z}{V_T\, V_H}\left[\sin\alpha\left(v\, V_T \sin\sigma - u\, w \cos\sigma\right) + w\, V_H \cos\alpha\right]$$

$$\frac{\partial T_r}{\partial \sigma} = -\frac{T_x V_H}{V_T} \sin\sigma \sin\alpha - \frac{T_y V_H}{V_T} \cos\sigma - \frac{T_z V_H}{V_T} \sin\sigma \cos\alpha$$

$$\frac{\partial T_\theta}{\partial \sigma} = \frac{T_x \sin\alpha}{V_T V_H} \left(uv \sin\sigma - w V_T \cos\sigma\right) + \frac{T_y}{V_T V_H} \left(uv \cos\sigma + w V_T \sin\sigma\right) + \frac{T_z \cos\alpha}{V_T V_H} \left(uv \sin\sigma - w V_T \cos\sigma\right)$$

$$\frac{\partial T_\phi}{\partial \sigma} = \frac{T_x \sin\alpha}{V_T V_H} \left(v V_T \cos\sigma + uw \sin\sigma\right) + \frac{T_y}{V_T V_H} \left(uw \cos\sigma - v V_T \sin\sigma\right) + \frac{T_z \cos\alpha}{V_T V_H} \left(v V_T \cos\sigma + uw \sin\sigma\right)$$

G. <u>Additional State Equations and Penalty Functions</u>

1. Total Heat Absorbed per Unit Area at Stagnation Point

$$\frac{dQ}{dt} = K_{QC} R_{QC} \rho^{1/2} \left(\frac{V_T}{1000}\right)^{N_{QC}} + K_{QR} R_{QR} \rho^{3/2} \left(\frac{V_T}{10000}\right)^{N_{QR}}$$

$K_{QC}$, $R_{QC}$, $N_{QC}$, $K_{QR}$, $R_{QR}$, $N_{QR}$ are constants (input)

$$\frac{\partial \dot{Q}}{\partial u} = \left[K_{QC} R_{QC} N_{QC} \rho^{1/2} \left(\frac{V_T}{1000}\right)^{N_{QC}} + K_{QR} R_{QR} N_{QR} \rho^{3/2} \left(\frac{V_T}{10000}\right)^{N_{QR}}\right] \frac{u}{V_T^2}$$

$$\frac{\partial \dot{Q}}{\partial v} = \left[K_{QC} R_{QC} N_{QC} \rho^{1/2} \left(\frac{V_T}{1000}\right)^{N_{QC}} + K_{QR} R_{QR} N_{QR} \rho^{3/2} \left(\frac{V_T}{10000}\right)^{N_{QR}}\right] \frac{v}{V_T^2}$$

$$\frac{\partial \dot{Q}}{\partial w} = \left[K_{QC} R_{QC} N_{QC} \rho^{1/2} \left(\frac{V_T}{1000}\right)^{N_{QC}} + K_{QR} R_{QR} N_{QR} \rho^{3/2} \left(\frac{V_T}{10000}\right)^{N_{QR}}\right] \frac{w}{V_T^2}$$

$$\frac{\partial \dot{Q}}{\partial r} = \left[ \frac{K_{QC}\ R_{QC}}{2\ \rho^{1/2}} \left(\frac{V_T}{1000}\right)^{N_{QC}} + \frac{3\ K_{QR} R_{QR}\ \rho^{1/2}}{2} \left(\frac{V_T}{10000}\right)^{N_{QC}} \right] \frac{d\rho}{dh}$$

$$\frac{\partial \dot{Q}}{\partial \theta} = - \frac{\partial \dot{Q}}{\partial r} \frac{dR}{d\theta}$$

$$\frac{\partial \dot{Q}}{\partial \phi} = \frac{\partial \dot{Q}}{\partial Q} = \frac{\partial \dot{Q}}{\partial p_a} = \frac{\partial \dot{Q}}{\partial p_H} = \frac{\partial \dot{Q}}{\partial p_{QD}} = \frac{\partial \dot{Q}}{\partial \alpha} = \frac{\partial \dot{Q}}{\partial \sigma} = 0$$

2. Pilot Acceleration Dose or Acceleration Penalty Function

$$\frac{dp_a}{dt} = \frac{1}{\tau(a)}$$

$\tau(a)$ input, table

$$a = \frac{\rho\ V_T^2 S}{64.4} \sqrt{C_D^2 + C_L^2}$$

Acceleration in g's.

$$\frac{\partial \dot{p}_a}{\partial u} = - \frac{1}{\tau(a)^2} \frac{d\ \tau\ (a)}{da} \left(\frac{\rho\ S}{64.4\ m}\right) \left(u \sqrt{C_L^2 + C_D^2}\right) \left[2 + \frac{M}{C_D^2 + C_L^2} \left(C_D \frac{\partial C_D}{\partial M} + C_L \frac{\partial C_L}{\partial M}\right)\right]$$

$$\frac{\partial \dot{p}_a}{\partial v} = - \frac{1}{\tau(a)^2} \frac{d\tau(a)}{da} \left(\frac{\rho\ S}{64.4\ m}\right) \left(v \sqrt{C_L^2 + C_D^2}\right) \left[2 + \frac{M}{C_D^2 + C_L^2} \left(C_D \frac{\partial C_D}{\partial M} + C_L \frac{\partial C_L}{\partial M}\right)\right]$$

$$\frac{\partial \dot{p}_a}{\partial w} = -\frac{1}{\tau(a)^2}\frac{d\tau(a)}{da}\left(\frac{\rho S}{64.4m}\right) w \sqrt{C_L^2 + C_D^2}\left[2 + \frac{M}{C_D^2 + C_L^2}\left(C_D \frac{\partial C_D}{\partial M} + C_L \frac{\partial C_L}{\partial M}\right)\right]$$

$$\frac{\partial \dot{p}_a}{\partial r} = -\frac{1}{\tau(a)^2}\frac{d\tau(a)}{da}\left(\frac{V_T^2 S}{64.4m}\right)\sqrt{C_L^2 + C_D^2}\left[\left(\frac{d\rho}{dh} - \frac{\rho M}{c\left(C_D^2 + C_L^2\right)}\frac{dc}{dh}\right)\left(C_D \frac{\partial C_D}{\partial M} + C_L \frac{\partial C_L}{\partial M}\right)\right]$$

$$\frac{\partial \dot{p}_a}{\partial \Theta} = -\frac{\partial \dot{p}_a}{\partial r}\frac{\partial R}{\partial \Theta}$$

$$\frac{\partial \dot{p}_a}{\partial \alpha} = -\frac{1}{\tau(a)^2}\frac{d\tau(a)}{da}\left(\frac{\rho V_T^2 S}{64.4m}\right)\left(C_D \frac{\partial C_D}{\partial \alpha} + C_L \frac{\partial C_L}{\partial \alpha}\right)\left(C_L^2 + C_D^2\right)^{-1/2}$$

$$\frac{\partial \dot{p}_a}{\partial \phi} = \frac{\partial \dot{p}_a}{\partial Q} = \frac{\partial \dot{p}_a}{\partial p_a} = \frac{\partial \dot{p}_a}{\partial p_H} = \frac{\partial \dot{p}_a}{\partial p_{QD}} = \frac{\partial \dot{p}_a}{\partial \sigma} = 0$$

3. Altitude Penalty Function

$$\frac{dp_H}{dt} = \begin{cases} A_{H1}\left(\frac{h-A_{H2}}{AH_2}\right)^2 & h > A_{H2}\text{, provided h has been less than } A_{H2} \\ 0 & h \le A_{H2} \end{cases}$$

$A_{H1}$, $A_{H2}$ input

$$\frac{\partial \dot{p}_H}{\partial r} = \begin{cases} \frac{2A_{H1}(h-A_{H2})}{A_{H2}^2} & \dot{p}_H \neq 0 \\ 0 & \dot{p}_H = 0 \end{cases}$$

$$\frac{\partial \dot{p}_H}{\partial \theta} = - \frac{\partial \dot{p}_H}{\partial r} \frac{dR}{d\theta}$$

$$\frac{\partial \dot{p}_H}{\partial u} = \frac{\partial \dot{p}_H}{\partial v} = \frac{\partial \dot{p}_H}{\partial w} = \frac{\partial \dot{p}_H}{\partial \phi} = \frac{\partial \dot{p}_H}{\partial Q} = \frac{\partial \dot{p}_H}{\partial p_a} = \frac{\partial \dot{p}_H}{\partial p_H} = \frac{\partial \dot{p}_H}{\partial p_{QD}} = \frac{\partial \dot{p}_H}{\partial \alpha} = \frac{\partial \dot{p}_H}{\partial \sigma} = 0$$

4. Heating Rate Penalty Function

$$\frac{dP_{QD}}{dt} = \begin{cases} A_{QD1} \left( \frac{\dot{Q} - A_{QD2}}{A_{QD2}} \right)^2 & \dot{Q} > A_{QD2} \\ 0 & \dot{Q} \leq A_{QD2} \end{cases}$$

$A_{QD1}$ , $A_{QD2}$ input

For $\dot{Q} > A_{QD2}$ :

$$\frac{\partial \dot{p}_{QD}}{\partial u} = \frac{2A_{QD1}(\dot{Q} - A_{QD2})}{A_{QD2}^2} \frac{\partial \dot{Q}}{\partial u}$$

$$\frac{\partial \dot{p}_{QD}}{\partial v} - \frac{2A_{QD1}(\dot{Q} - A_{QD2})}{A_{QD2}^2} \frac{\partial \dot{Q}}{\partial v}$$

$$\frac{\partial \dot{p}_{QD}}{\partial w} = \frac{2A_{QD1}(\dot{Q} - A_{QD2}}{A_{QD2}^2} \frac{\partial \dot{Q}}{\partial w}$$

$$\frac{\partial \dot{p}_{QD}}{\partial r} = \frac{2A_{QD1}(\dot{Q}-A_{QD2})}{A_{QD2}^2} \frac{\partial \dot{Q}}{\partial r}$$

$$\frac{\partial \dot{p}_{QD}}{\partial \theta} = \frac{2A_{QD1}(\dot{Q}-A_{QD2})}{A_{QD2}^2} \frac{\partial \dot{Q}}{\partial \theta}$$

For $\dot{Q} \leq A_{QD2}$:

$$\frac{\partial \dot{p}_{QD}}{\partial u} = \frac{\partial \dot{p}_{QD}}{\partial v} = \frac{\partial \dot{p}_{QD}}{\partial w} = \frac{\partial \dot{p}_{QD}}{\partial r} = \frac{\partial \dot{p}_{QD}}{\partial \theta} = 0$$

For all $\dot{Q}$:

$$\frac{\partial \dot{p}_{QD}}{\partial \phi} = \frac{\partial \dot{p}_{QD}}{\partial Q} = \frac{\partial \dot{p}_{QD}}{\partial p_a} = \frac{\partial \dot{p}_{QD}}{\partial p_H} = \frac{\partial \dot{p}_{QD}}{\partial p_{QD}} = \frac{\partial \dot{p}_{QD}}{\partial \alpha} = \frac{\partial \dot{p}_{QD}}{\partial \sigma} = 0$$

## H. Optimization Equations

1. Definitions (Also see List of Symbols in Section I)

$N_c$ = Total number of inequality and equality terminal constraints specified for given problem $0 \leq N_c \leq 8$

L = 8-vector indicating status of terminal constraints.

$$L_i = \begin{cases} 1, & 0 < i \leq N_c \text{ if constraint } \psi_i \text{ is "active"} \\ 0, & 0 < i \leq N_c \text{ if constraint } \psi_i \text{ is "inactive"} \\ 0, & N_c < i \leq 8, \text{ constraint not specified} \end{cases}$$

Inequality terminal constraints having current values that lie outside of specified limits and all equality terminal constraints are active; inequality terminal constraints having current values that lie within specified limits are inactive, see Section H.9. When L is used as a subscript on a matrix, it implies that the matrix consists of only those rows and/or columns for which the corresponding $L_i = 1$.

$F_1 = (dP)^2 - \psi'_{dL} I^{-1}_{\psi\psi L} \psi_{dL}$, a factor computed during optimization process, section H.10. Since $\sqrt{F_1}$ is required, $F_1$ must be $\geq 0$ ; if $F_1$ is computed as $< 0$ , then $\psi_d$ are scaled by $S_{FC}$

2. Errors in Terminal Constraints

$$\psi_{\varepsilon i} = \begin{cases} \psi_{Ai} - \psi_{Ti} & \text{if } \psi_{Ai} \geq \psi_{Ti} + \varepsilon_i \\ \psi_{Ai} - \psi_{Bi} & \text{if } \psi_{Ai} \leq \psi_{Bi} - \varepsilon_i \\ 0 & \text{if } \psi_{Bi} - \varepsilon_i < \psi_{Ai} < \psi_{Ti} + \varepsilon_i \end{cases}$$

$$i = 1, 2 \ldots N_c$$

3. Performance Indices

If $N_c = 0$:

$$\underline{\Phi}^*_A = \Phi^*$$

$$\underline{\Phi}_A = \Phi$$

$$d\Phi_A = d\underline{\Phi}_A = \underline{\Phi}_A - \underline{\Phi}^*_A$$

$$d\Phi_P = d\underline{\Phi}_P = a_m \left[(dP)^2 \; I_{\Phi\Phi}\right]^{1/2}$$

$$R_{d\Phi} = R_{d\underline{\Phi}} = d\underline{\Phi}_A / d\underline{\Phi}_P$$

$$\Delta\Phi = \Delta\underline{\Phi} = d\underline{\Phi}_A / dP$$

If $N_c \neq 0$:

$$\Psi_A^* = \left( \sum_{i=1}^{N_c} \frac{\psi_{\varepsilon i}^{*2}}{I_{\psi_i \psi_i}} \right)^{1/2}$$

$$\Psi_A = \left( \sum_{i=1}^{N_c} \frac{\psi_{\varepsilon i}^{2}}{I_{\psi_i \psi_i}} \right)^{1/2}$$

$$d\Psi_A = \Psi_A - \Psi_A^*$$

$$d\Psi_P = - S_{FC} \; \Psi_A^*$$

$$R_{d\Psi} = d\Psi_A / d\Psi_P$$

$$\underline{\Phi}_A^* = \Phi^* - I'_{\psi\Phi L} \; I^{-1}_{\Psi\Psi L} \; \Psi^*_{\varepsilon L}$$

$$\underline{\Phi}_A = \Phi - I'_{\psi\Phi L} \; I^{-1}_{\psi\psi L} \; \psi_{\varepsilon L}$$

$$d\underline{\Phi}_A = \underline{\Phi}_A - \underline{\Phi}_A^*$$

$$d\underline{\Phi}_P = \begin{cases} a_m \left\{ \left[(dP)^2 - d\beta'_L \; I^{-1}_{\psi\psi L} \; d\beta_L\right] \left(I_{\Phi\Phi} - I'_{\psi\Phi L} \; I^{-1}_{\psi\psi L} \; I_{\psi\Phi L}\right) \right\}^{1/2}, & \text{if } F_1 > 0 \\ 0 \quad , \quad \text{if } F_1 \leq 0 & \end{cases}$$

$$R_{d\underline{\Phi}} = \begin{cases} d\underline{\Phi}_A / d\underline{\Phi}_P & , \text{ if } F_1 > 0 \\ 0 & , \text{ if } F_1 \leq 0 \end{cases}$$

$$d\psi_A = \psi_A - \psi_A^*$$

$$R_{d\psi} = d\psi_A - d\psi$$

$$d\Phi_A = \Phi - \Phi^*$$

$$d\Phi_P = d\underline{\Phi}_P + I\psi'_{\Phi L} \; I^{-1}_{\psi\psi L} \; d\beta_L$$

$$R_{d\Phi} = d\Phi_A / d\Phi_P$$

$$\Delta \underline{\Phi} = d\underline{\Phi}_A / dP$$

$$\Delta \Phi = d\Phi_A / dP$$

4. New $(dP)^2$

The option of using a constant $(dP)^2$ is provided by setting the input constant $K_A = 2.0$; to use the variable step-size feature described by the following logic, set $K_A = 1.0$.

If $F_1 \leq 0$ and if

$R_{d\Psi} \geq A_{21}$ multiply $(dP)^2$ by $A_{11}$

$A_{21} > R_{d\Psi} \geq A_{22}$ continue with same $(dP)^2$

$A_{22} > R_{d\Psi}$ multiply $(dP)^2$ by $A_{12}$

Or, if $F_1 > 0$ and if

$R_{d\underline{\Phi}} \geq A_{31}$ multiply $(dP)^2$ by $A_{12}$

$A_{31} > R_{d\underline{\Phi}} \geq A_{32}$ multiply $(dP)^2$ by $A_{11}$

$A_{32} > R_{d\underline{\Phi}} \geq A_{33}$ continue with same $(dP)^2$

$A_{33} > R_{d\underline{\Phi}}$ multiply $(dP)^2$ by $A_{12}$

$A_{11} > 1 > A_{12} > 0$; $A_{11}$ , $A_{12}$ , input as DPINC, DPDEC

$A_{21} > A_{22}$ (usually, $1 > A_{21} > A_{22} > 0$); $A_{21}$ , $A_{22}$ input as PSIEG , PSIGP

$A_{31} > A_{32} > A_{33}$ (usually $A_{31} > 1 > A_{32} > A_{33} > 0$); $A_{31}$ , $A_{32}$ , $A_{33}$ input as PHIGE , PHIEG , PHIGP

6. Rejection of Unacceptable Iteration

When operating in variable $(dP)^2$ mode, $K_A = 1.0$, the current choice for $\delta\alpha$ and the associated integration of the state equations is rejected and the program proceeds immediately to Section H.10 of these equations if

$F_1 \leq 0$ and $R_{d\Psi} < A_{23}$

Or, $F_1 > 0$ and $R_{d\underline{\Phi}} < A_{34}$

$A_{23} < A_{22}$ ; $A_{23}$ input as PSIRJ

$A_{34} < A_{33}$ ; $A_{34}$ input as PHIRJ

7. Stops for Termination of Iterations

Only the first two of these stops are applicable if $K_A = 2.0$; all are applicable if $K_A = 1.0$. When a stop is encountered, other than $K_S = 1$, the program terminates the current case and calls the input for the next case, if any.

| | |
|---|---|
| $K_S = 1$ | do not stop |
| $K_S = 2$ | requested number of iterations, NIT, completed; NIT = 0 results in integration of only the state equation; NIT input. |
| $K_S = 3$ | $\lvert\Delta\underline{\Phi}\rvert \leq \Delta\underline{\Phi}_{MIN}\rvert$ for $N_{\Delta F}$ consecutive iterations; $\Delta\underline{\Phi}_{MIN}$, $N_{\Delta F}$ input as CDELM, NDELF |
| $K_S = 4$ | current iteration acceptable but $(dP)^2 \leq (dP)^2_{MIN}$ selected for next iterations; $(dP)^2_{MIN}$ input as DPSQM |
| $K_S = 5$ | current choice of control programs and all previous choices since $(dP)^2$ became $\leq (dP)^2_{MIN}$ are unacceptable and new $(dP)^2 \leq 0.01\ (dP)^2_{MIN}$ |
| $K_S = 6$ | last five choices of $(dP)^2$ have yielded unacceptable control programs |

8. Integrals of Products of Impulse Response Functions

$$I_{\psi\psi} = \int_{to}^{T} (G'\lambda_{\psi\Omega})' \ W^{-1} G'\lambda_{\psi\Omega} \ dt$$

$$I_{\psi\Phi} = \int_{to}^{T} (G'\lambda_{\psi\Omega})' \ W^{-1} G'\lambda_{\Phi\Omega} \ dt$$

$$I_{\Phi\Phi} = \int_{to}^{T} (G'\lambda_{\Phi\Omega})' \; W^{-1} \; G'\lambda_{\Phi\Omega} \; dt$$

The G matrix is computed for the last acceptable integration of the state equations.

9. Status of Constraints

Determine status of constraints after each acceptable integration of state equations; do not change L if trajectory is unacceptable.

$$L_i = \begin{cases} 1 & \text{if } \psi_{Ai} \geq \psi_{Ti} + \varepsilon_i \\ & \text{Or } \psi_{Ai} \leq \psi_{Bi} + \varepsilon_i \\ 0 & \text{if } \psi_{Bi} - \varepsilon_i < \psi_{Ai} < \psi_{Ti} + \varepsilon_i \\ & \text{Or } i^{th} \text{ constraint not specified} \end{cases}$$

$$i = 1, 2, \ldots 8$$

10. Selection of Changes to be made in Constraints

$$\psi_d = -\psi_\varepsilon^*$$

$$F_1 = (dP)^2 - \psi_{dL}' I_{\psi\psi L}^{-1} \psi_{dL}$$

If $F_1 \geq 0$:

$$S_{FC} = 1$$

Or, if $F_1 < 0$:

$$S_{FC} = \left[(dP)^2 / \psi'_{dL} I^{-1}_{\psi\psi L} \psi_{dL}\right]^{1/2}$$

Set $F_1 = 0$

$$d\psi = S_{FC} \psi_d$$

11. Control of Inactive Constraints

For <u>each</u> i, i = 1, 2, . . . $N_c$, for which $L_i = 0$ do the following computations:

$$d\psi_i = a_m \left[(I_{\psi_i\Phi} - I_{\psi_i\psi L} I^{-1}_{\psi\psi L} I_{\psi\Phi L}) \sqrt{\frac{(dP)^2 - d\psi'_L I^{-1}_{\psi\psi L} d\psi_L}{I_{\Phi\Phi} - I'_{\psi\Phi L} I^{-1}_{\psi\psi L} I_{\psi\Phi L}}} + I_{\psi_i\psi L} I^{-1}_{\psi\psi L} d\psi_L\right]$$

If $\psi^*_{Ai} + d\psi_i \geq \psi_{Ti} + \varepsilon_i$:

$$\psi_{di} = \psi_{Ti} - \psi^*_{Ai}$$

$$L_i = 1$$

$$K_{CI} = 2$$

Or, if $\psi_{Ti} + \varepsilon_i > \psi^*_{Ai} + d\psi_i > \psi_{Bi} - \varepsilon_i$:

$$d\psi_i = d\psi_i$$

$$L_i = 0$$

$$K_{CI} = 1$$

Or, if $\psi_{Bi} - \varepsilon_i \geq \psi^*_{Ai} + d\psi_i$:

$$\psi_{di} = \psi_{Bi} - \psi^*_{Ai}$$

$$L_i = 1$$

$$K_{CI} = 2$$

Upon completion of preceding computation for all inactive constraint,

If $K_{CI} = 2$ , return to the <u>second</u> equation of Section H.10 and repeat subsequent steps in Sections H.10 and H.11.

Or, if $K_{CI} = 1$, proceed to Section 12.

12. New Control Programs

$$d\beta = d\psi \qquad \text{(Assumption, } \delta x(t_o)=0\text{)}$$

$$\delta\alpha(t) = a_m W^{-1}(G'\lambda_{\Phi\Omega} - G'\lambda_{\psi\Omega L} \; I^{-1}_{\psi\psi L} \; I_{\psi\Phi L})\sqrt{\frac{(dP)^2 - d\beta'_L I^{-1}_{\psi\psi L} d\beta_L}{I_{\Phi\Phi} - I'_{\psi\Phi L} I^{-1}_{\psi\psi L} I_{\psi\Phi L}}}$$

$$+ W^{-1} G' \lambda_{\psi\Omega L} \; I^{-1}_{\psi\psi L} \; d\beta_L$$

$$\alpha(t) = \alpha^*(t) + \delta\alpha(t)$$

where

$$\alpha(t) = \begin{bmatrix} \alpha(t) \\ \sigma(t) \end{bmatrix}$$

13. LAMCOS-DO

During first integration of state equations or when $N_c = 0$, LAMCOS-DO is not used. Determine time for application of LAMCOS-DO from

$$t_k = \frac{kT}{n} \qquad k = 1, 2, \ldots n-1 \leq 9$$

n, input as LAM

Or, $t_k = t_i$ , $i = 1, 2, \ldots m \leq 9$ , $0 < t_1 < t_2 \ldots < t_m$

$t_i$ , input as LAMT(I)

do not use any $t_i > .98T$

For each $t_k$ make following computations:

$$I_{\psi\psi k} = \int_{t_k}^{T} (G'\lambda_{\psi\Omega})' \, W^{-1} \, (G'\lambda_{\psi\Omega}) \, dt$$

$$I_{\psi\Phi} = \int_{t_k}^{T} (G'\lambda_{\psi\Omega})' \, W^{-1} \, (G'\lambda_{\Phi\Omega}) \, dt$$

$$I_{\Phi\Phi} = \int_{t_k}^{T} (G'\lambda_{\Phi\Omega})' \, W^{-1} \, (G'\lambda_{\Phi\Omega}) \, dt$$

$$(dP_k)^2 = (dP)^2 - \int_{o}^{t_k} \delta\alpha \, W\delta\alpha \, dt$$

$$d\beta_k = d\psi - \lambda'_{\psi\Omega k} \left[ x(t_k) - x^*(t_k) \right]$$

$$F_{2k} = (dP_k)^2 - d\beta'_{kL}\, I^{-1}_{\psi\psi kL}\, d\beta_{kL}$$

If $F_{2k} \geq 0$ proceed with computation

Or, if $F_{2k} < 0$

$$\text{set } d\beta_k = \left[\frac{(dP_k)^2}{d\beta'_{kL}\, I^{-1}_{\psi\psi kL}\, d\beta_{kL}}\right]^{1/2} d\beta_k$$

$$A_k = \left[\frac{(dP_h)^2 - d\beta'_{kL} I^{-1}_{\psi\psi kL}\, d\beta_{kL}}{I_{\Phi\Phi k} - I'_{\psi\Phi kL}\, I^{-1}_{\psi\psi kL}\, I_{\psi\Phi kL}}\right]^{1/2}$$

For $t_k < t \leq t_{k+1}$:

$$\delta\alpha(t) = a_m\, W^{-1}(G'\lambda_{\Phi\Omega} - G'\lambda_{\psi\Omega L}\, I^{-1}_{\psi\psi kL}\, I_{\psi\Phi kL})\, A_k$$

$$+ W^{-1}\, G'\lambda_{\psi\Omega L}\, I^{-1}_{\psi\psi kL}\, d\beta_{kL}$$

where

$$\delta\alpha(t) = \begin{bmatrix} \partial\alpha(t) \\ \partial\sigma(t) \end{bmatrix}$$

## I. List of Symbols

| | |
|---|---|
| $A_{11}$ | factor, > 1, by which $(dP)^2$ is increased, input as DPINC |
| $A_{12}$ | factor, < 1, > 0, by which $(dP)^2$ is decreased, input as DPDEC |
| $A_{21}$ | factor, < 1, > $A_{22}$, separating excellent and good values of $\Psi$, input as PSIEG |
| $A_{22}$ | factor, < $A_{21}$, > $A_{23}$, separating good and poor values of $\Psi$, input as PSIGP |
| $A_{23}$ | value of $\Psi$, < $A_{22}$, ≥ 0, below which iteration is rejected if $\Psi$ index is being used, input as PSIRJ |
| $A_{31}$ | factor, > 1, separating good and excellent values of $\underline{\Phi}$, input as PHIGE |
| $A_{32}$ | factor, < 1, > $A_{33}$, separating excellent and good values of $\underline{\Phi}$, input as PHIEG |
| $A_{33}$ | factor, < $A_{32}$, > $A_{34}$, separating good and poor values of $\underline{\Phi}$, input as PHIGP |
| $A_{34}$ | value of $\underline{\Phi}$, < $A_{33}$, ≥ 0, below which iteration is rejected if $\underline{\Phi}$ index is being used, input as PHIRJ |
| $A_{H1}$ | constant in altitude penalty function, input |
| $A_{H2}$ | upper bound of altitude during flight, input |
| $A_{QD1}$ | constant in heating-rate penalty function, input |
| $A_{QD2}$ | upper bound on the heating rate during flight, input |
| $A_e$ | exit area of rocket motor, $ft^2$, input for each thrust stage |
| a | aerodynamic acceleration, g's |
| $a_m$ | constant, +1 or -1, indicating that $\Phi$ is being maximized or minimized, respectively, input by sign of payoff function library number |
| $C_D$ | drag coefficient, dimensionless, input as $C_D$ ($\alpha$, M, Stage No.) |
| $C_L$ | lift coefficient, dimensionless, input as $C_L$ ($\alpha$, M, Stage No.) |

| | |
|---|---|
| $c$ | local speed of sound, ft $sec^{-1}$ |
| $D$ | aerodynamic drag force, lbs |
| $(dP)^2$ | step size or the integral of the weighted square of the variation in control variables for an iteration, $radians^2$ sec |
| $d\beta$ | composite of predicted changes in terminal constraint functions and actual changes in initial state |
| $E$ | total energy of vehicle, ft lb |
| $f_i$ | known functions of state variables, control variables, and time; $= \dot{x}_i$ |
| $G$ | matrix of partial derivatives of $f_i$ with respect to $\alpha_j$ along nominal path |
| $g$ | gravitational acceleration, ft $sec^{-2}$ |
| $h$ | local altitude, ft |
| $I_{\Phi\Phi}$ | integral of weighted square of impulse response functions for changes in payoff function |
| $I_{\psi\Phi}$ | integral of weighted product of impulse response functions for changes in terminal constraint functions and changes in payoff function |
| $I_{\psi\psi}$ | integral of weighted square of impulse response functions for changes in terminal constraint functions |
| $i_T$ | thrust misalignment angle in the x-z plane of vehicle, degrees, input for each stage |
| $J$ | constant in the gravitational expression, dimensionless |
| $K_{QC}$ | weighting or scale factor for convective heating, input |
| $K_{QR}$ | weighting or scale factor for radiative heating, input |
| $L$ | aerodynamic lift force, lbs |
| $\underline{L}$ | vector indicating status of terminal constraint functions |
| $M$ | mach number, dimensionless |
| $m$ | mass of vehicle, slugs, input for each stage |
| $N_{\Delta F}$ | number of consecutive times $\lvert\Delta\underline{\Phi}\rvert \leq \Delta\underline{\Phi}_{MIN}$ before stop |

| | |
|---|---|
| $N_{QC}$ | exponent of $V_T$ for convective heating, input |
| $N_{QR}$ | exponent of $V_T$ for radiative heating, input |
| $P_e$ | ambient atmospheric pressure at sea level, lb $ft^{-2}$, input |
| $P_\infty$ | local atmospheric pressure, lb $ft^{-2}$ |
| $p_H$ | normalized altitude penalty function |
| $p_{QD}$ | normalized heating rate ($\dot{Q}$) penalty function |
| $p_a$ | pilot acceleration dose, acceleration penalty function |
| Q | heat absorbed at stagnation point, units depend on $K_{QC}$, $K_{QR}$ |
| q | dynamic pressure, lb $ft^{-2}$ |
| R | radius of earth computed from spheroid approximation, ft |
| $R_{NQC}$ | effective nose radius for convective heating, ft |
| $R_{QC}$ | function of $N_{QC}$ and $R_{QC}$, input |
| $R_{QR}$ | function of $N_{QR}$ and $R_{QR}$, input |
| $R_{d\underline{\Phi}}$ | ratio of actual to predicted change in $\underline{\Phi}$; similarly, for $R_{d\Psi}$, $R_{d\Phi}$, $R_{d\psi}$ |
| $R_e$ | equatorial radius of earth in oblate spheroid approximation, ft, input |
| $R_f$ | reference radius of earth used in gravitational expansion, ft, input |
| $R_p$ | polar radius of earth in oblate spheroid approximation, ft, input as RPC |
| r | radial distance from center of earth in spherical coordinate system, ft |
| S | reference area of vehicle, $ft^2$, input for each stage |
| $S_{FC}$ | scale factor, $\leq 1, > 0$, by which desired changes in terminal constraints are multiplied |
| $S_{Fk}$ | Supplementary scale factor, $\leq 1$, $> 0$, introduced by LAMCOS-DO at $t_k$ |
| T | terminal time of trajectory, sec |
| T | thrust, lbs |

| | |
|---|---|
| $T_{SL}$ | basic sea-level rocket thrust, lbs, input as $T_{SL}(t)$ |
| $t$ | time, sec |
| $u$ | component of velocity in $\hat{r}$ direction, ft $sec^{-1}$ |
| $V_H$ | component of velocity in local $\hat{\theta}$ - $\hat{\phi}$ plane, ft $sec^{-1}$ |
| $V_T$ | magnitude of velocity relative to the rotating earth, ft $sec^{-1}$ |
| $v$ | component of velocity in $\hat{\theta}$ direction, ft $sec^{-1}$ |
| $W$ | diagonal weighting matrix used to facilitate convergence, input |
| $w$ | component of velocity in $\hat{\phi}$ direction, ft $sec^{-1}$ |
| $x$ | axis in the longitudinal plane of vehicle |
| $x_i$ | state variables |
| $y$ | axis in the lateral plane of vehicle |
| $z$ | axis in the vertical plane of vehicle |
| $\alpha$ | angle-of-attack, radians (input-output, degrees) |
| $\alpha_i$ | control variables |
| $\beta$ | heading angle from the local $\hat{\theta}$ direction, radians input-output, degrees) |
| $\gamma$ | flight path angle to the local $\hat{\theta}$ - $\hat{\phi}$ plane, radians (input-output, degrees) |
| $\Delta\underline{\Phi}$ | actual gradient of $\underline{\Phi}$ with respect to dP |
| $\Delta\underline{\Phi}_{MIN}$ | $\lvert\Delta\underline{\Phi}\rvert$ below which convergence is assumed |
| $\Delta\Phi$ | actual gradient of $\Phi$ with respect to dP |
| $\delta_T$ | thrust misalignment angle in the x-y plane of vehicle, degrees, input for each stage |
| $\varepsilon$ | tolerances in meeting specified values of inequality terminal constraints, input as EPS |
| $\theta^*$ | co-latitude angle in "initial condition" oriented spherical coordinate system, degrees |
| $\lambda$ | adjoint function, units vary |

| | |
|---|---|
| $\mu$ | longitude in "initial condition" oriented spherical coordinate system, degrees |
| $\mu_e$ | a constant in the gravitational expansion, $ft^3\ sec^{-2}$ |
| $\nu$ | latitude in "initial condition" oriented spherical coordinate system, degrees |
| $\rho$ | density, slugs $ft^{-3}$ |
| $\sigma$ | bank angle, radians (input-output, degrees) |
| $\tau(a)$ | acceleration-endurance function, sec, input |
| $\underline{\Phi}$ | PHI index for performance of optimization procedure when variations in control programs are being used, at least in part, to optimize $\Phi$ |
| $\Phi$ | payoff function used in optimization procedure, input |
| $\phi$ | azimuthal angle in spherical coordinate system, radians (input-output, degrees) |
| $\phi^*$ | azimuthal angle in "initial condition" oriented spherical system, degrees |
| $\Psi$ | PSI index for performance of optimization procedure when variations in control programs are being used solely to achieve terminal constraints |
| $\psi$ | terminal constraints in optimization procedure, input as CON |
| $\psi_B$ | lower or bottom limits on terminal constraints, input as SIB |
| $\psi_T$ | upper or top limits on terminal constraints, input as SIT |
| $\psi_\varepsilon$ | errors in meeting specified values of terminal constraints |
| $\Omega$ | stopping condition in optimization procedure, input as STOP and STOPV |
| $\Omega_e$ | angular velocity of earth, radians $sec^{-1}$, input |

Subscripts

| | |
|---|---|
| A | actual |
| i, j | indices |
| k | index designating time at which LAMCOS-DO is employed |

| | |
|---|---|
| L | matrix consists of only those rows and/or columns for which the elements of L are nonzero |
| o | initial condition |
| P | predicted |
| r, $\theta$, $\phi$ / x, y, z | component along the related coordinate axis |

Superscripts

| | |
|---|---|
| . | differentiation by time |
| ' | transpose of the matrix |
| * | associated with last acceptable iteration (except where used with coordinate angles $\theta$* and $\phi$*) |

# APPENDIX III

## DEFINITION OF VALUES IN COMMON STORAGE

| COMMON / INPUT / | | | COMMON NO. 1 |
|---|---|---|---|
| 1 | CASENO(2) | | Case Number and iteration number. Input symbol CASENO |
| 3 | ID(3) | | Identification, alpha-numeric. Input sumbol ID |
| 6 | XVT | $V_0$ | Initial velocity in ft./sec. Input sumbol VEL |
| 7 | XGAMMD | $\gamma_0$ | Initial flight path angle in deg. Input symbol GAMMA |
| 8 | XBETAD | $\beta_0$ | Initial heading angle in deg. Input symbol BETA |
| 9 | XH | $h_0$ | Initial height in ft. Input symbol H |
| 10 | XTHETD | $\theta_0$ | Initial co-latitude in deg. Input symbol THETA |
| 11 | XPHID | $\phi_0$ | Initial longitude in deg. Input symbol PHI |
| 12 | GS | S | Glide reference area in ft.$^2$ Input symbol GS |
| 13 | GMASS | m | Glide mass in slugs. Input symbol GMASS |
| 14 | PAYOFF | | Library number of payoff function. Input symbol PAYOFF |
| 15 | STOP | | Library number of stopping function. Input symbol STOP |
| 16 | STOPV | | Stopping value in units of stopping function. Input symbol STOPV |
| 17 | HP | | Print interval for forward trajectory in sec. Input symbol PRTINT |
| 18 | AIT | | Maximum number of iterations. Input symbol NIT |
| 19 | XDPSQ | $(dP)^2$ | Initial steepest-descent step size. Input symbol DPSQ |
| 20 | A11 | $A_{11}$ | Constant for increase of $(dP)^2$. Input symbol DPINC |
| 21 | A12 | $A_{12}$ | Constant for decrease of $(dP)^2$. Input symbol DPDEC |

| | | | |
|---|---|---|---|
| 22 | A21 | $A_{21}$ | Constant dividing excellent and good $\Psi$. Input symbol PSIEG |
| 23 | A22 | $A_{22}$ | Constant dividing good and poor $\Psi$. Input symbol PSIGP |
| 24 | A31 | $A_{31}$ | Constant (>1) dividing good and excellent $\underline{\Phi}$. Input symbol PHIGE |
| 25 | A32 | $A_{32}$ | Constant (<1) dividing excellent and good $\underline{\Phi}$. Input symbol PHIEG |
| 26 | A33 | $A_{33}$ | Constant dividing good and poor $\underline{\Phi}$. Input symbol PHIGP |
| 27 | DELF | $N_{\Delta F}$ | Number of times $\lvert\Delta\underline{\Phi}\rvert \leq \Delta\underline{\Phi}_{MIN}$ before stop. Input symbol NDELF |
| 28 | CEHIM | $\Delta\underline{\Phi}_{MIN}$ | Minimum actual gradient. Input symbol CDELM |
| 29 | DPSQM | $(dP)^2_{MIN}$ | Minimum $(dP)^2$. Input symbol DPSQM |
| 30 | K1 | $K_{QC}$ | Constant for convective heating. Input symbol KQC |
| 31 | R1 | $R_{QC}$ | Constant for convective heating. Input symbol RQC |
| 32 | N1 | $N_{QC}$ | Constant for convective heating. Input symbol NQC |
| 33 | K2 | $K_{QR}$ | Constant for radiative heating. Input symbol KQR |
| 34 | R2 | $R_{QR}$ | Constant for radiative heating. Input symbol RQR |
| 35 | N2 | $N_{QR}$ | Constant for radiative heating. Input symbol NQR |
| 36 | AH1 | $A_{H1}$ | Constant for altitude penalty function. Input symbol AH1 |
| 37 | AH2 | $A_{H2}$ | Constant for altitude penalty function. Input symbol AH2 |
| 38 | AHR1 | $A_{QD1}$ | Constant for heat rate penalty function. Input symbol AQD1 |
| 39 | AHR2 | $A_{QD2}$ | Constant for heat rate penalty function. Input symbol AQD2 |

| | | | |
|---|---|---|---|
| 40 | LAM | | LAMCOS-DO equal interval indicator. Input symbol LAM |
| 41 | SLPE | $P_e$ | Sea level pressure lbs./ft.$^2$. Input symbol SLPE |
| 42 | MU | $\mu_e$ | Gravitation constant in ft.$^3$/sec.$^2$ Input symbol MU |
| 43 | GJ | J | Gravitation constant. Input symbol J |
| 44 | RF | $R_f$ | Reference radius for gravitation of earth in ft. Input symbol RF |
| 45 | OMEGA | $\Omega_e$ | Angular velocity of planet in rad./sec. Input symbol OMEGA |
| 46 | RE | $R_e$ | Radius of oblate spheroid at equator in ft. Input symbol RE |
| 47 | RPC | $R_p$ | Radius of oblate spheroid at pole in ft. Input symbol RPC |
| 48 | TLIM | | Time limit in sec. Input symbol TLIM |
| 49 | HLIM | | Height limit in ft. Input symbol HLIM |
| 50 | A23 | $A_{23}$ | Constant for reject based on $\Psi$. Input symbol PSIRJ |
| 51 | A34 | $A_{34}$ | Constant for reject based on $\underline{\Phi}$. Input symbol PHIRJ |
| 52 | ORDA | | Order of differences for MARK during forward integration. Input symbol ORDA |
| 53 | HNOMA | | Nominal step-size for MARK during forward integration. Input symbol HNOMA |
| 54 | EUA | | $\bar{E}$ for MARK during forward integration. Input symbol EUA |
| 55 | ELA | | $\underline{E}$ for MARK during forward integration. Input symbol ELA |
| 56 | HMAXA | | Maximum step-size for MARK during forward integration. Input symbol HMAXA |
| 57 | HMINA | | Minimum step-size for MARK during forward integration. Input symbol HMINA |
| 58 | YCA | | YCLOW for MARK during forward integration. Input symbol YCA |

| | | | |
|---|---|---|---|
| 59 | ORDB | | Order of differences for MARK during adjoint integration. Input symbol ORDB |
| 60 | HNOMB | | Nominal step-size for MARK during adjoint integration. Input symbol HNOMB |
| 61 | EUB | | $\bar{E}$ for MARK during adjoint integration. Input symbol EUB |
| 62 | ELB | | $\underline{E}$ for MARK during adjoint integration. Input symbol ELB |
| 63 | HMAXB | | Maximum step-size for MARK during adjoint integration. Input symbol HMAXB |
| 64 | HMINA | | Minimum step-size for MARK during adjoint integration. Input symbol HMINB |
| 65 | YCB | | YCLOW for MARK during adjoint integration. Input symbol YCB |
| 66 | ANTABX(100) | $\alpha(t)$ | Nominal control variable program of $\alpha(t)$. Input symbol ALPHA |
| 166 | SNTABX(100) | $\sigma(t)$ | Nominal control variable program of $\sigma(t)$. Input symbol SIGMA |
| 266 | CON(8) | $\psi$ | Numbers of constraint function. Input symbol CON |
| 274 | SIT(8) | $\psi_T$ | Upper limits on constraints. Input symbol SIT |
| 282 | SIB(8) | $\psi_B$ | Lower limits on constraints. Input symbol SIB |
| 290 | EPS(8) | $\varepsilon$ | Small tolerance on constraints limits. Input symbol EPS |
| 298 | EOST(4) | | End of stage times in sec. Input symbol EOST |
| 302 | AE(3) | $A_e$ | Exit areas in ft.$^2$ for boost stages. Input symbol AE |
| 305 | REFA(3) | S | Reference area in ft.$^2$ for boost stages. Input symbol REFA |
| 308 | IT(3) | $i_T$ | $i_T$ in degrees for boost stages. Input symbol IT |
| 311 | DT(3) | $\delta_T$ | $\delta_T$ in degrees for boost stages. Input symbol DT |
| 314 | TMTAB(150) | $T_{SL}(t)$ $m(t)$ | Thrust and mass tables versus time for boost stages. Input symbol TMT |
| 464 | TAUTAB(100) | $\tau(a)$ | Acceleration penalty function. Input symbol TAU |
| 564 | WTAB(50,3) | $W(t)$ | Weighting function. Input symbol WT |

| | | |
|---|---|---|
| 714 | MTAB(11,4) | Mach tables for lift and drag coefficients. Input symbol MCLCD |
| 758 | ATAB(26,4) | $\alpha$ tables for lift and drag coefficients. Input symbol ACLCD |
| 862 | CLTAB(150,4) $C_L(\alpha, M)$ | Lift coefficients. Input symbol CL |
| 1462 | CDTAB(150,4) $C_D(\alpha, M)$ | Drag coefficients. Input symbols CD |
| 2062 | DELTA | Interval in sec. between computed points in control variable tables. Input symbol DELTA |
| 2063 | ATABX(201) $\alpha(t)$ | Computed control variable table for $\alpha$ in deg. Input symbol ALPHX |
| 2264 | STABX(201) $\sigma(t)$ | Computed control variable table for $\sigma$ in deg. Input symbol SIGMX |
| 2465 | LAMTAB(9) | Times in sec. at which LAMCOS-DO will be performed. Input symbol LAMT |
| 2474 | PRTOPT(4) | Print and punch options. Input symbol PRTOPT |
| 2478 | INPUTX(10) | Extra input area. Input symbol INPUT |
| 2488 | HFMT(140) | Title format area. Input symbol HFMT |
| 2528 | DFMT(140) | Column heading format area. Input symbol DFMT |
| 2668 | IGO(170) | IGO blocks for subroutine OUT. Input symbol IGO |

| COMMON / VAR / | | COMMON NO. 2 |
|---|---|---|
| 1 | RADIAN | Constant 57.2957795; conversion factor degrees per radian |
| 2 | D10 | Common term; $= \mu JR_f^2$ |
| 3 | D11 | " " ; $= 4\mu JR_f^2$ |
| 4 | D12 | " " ; $= 8\mu JR_f^2$ |
| 5 | D13 | " " ; $= 6\mu JR_f^2$ |
| 6 | D14 | " " ; $= 2\mu JR_f^2$ |
| 7 | D20 | " " ; $= 2\mu$ |

| | | | |
|---|---|---|---|
| 8 | D30 | | Common term; $= \Omega_e^2$ |
| 9 | D31 | | " " ; $= 2\Omega_e$ |
| 10 | D32 | | " " ; $= 2\Omega_e^2$ |
| 11 | ALPHD | $\alpha$ | Control variable; degrees |
| 12 | SIGMD | $\sigma$ | Control variable; degrees |
| 13 | ALPHR | $\alpha$ | Control variable; radians |
| 14 | SIGMR | $\sigma$ | Control variable; radians |
| 15 | CSIGM | $\cos\sigma$ | Cosine control variable $\sigma$ |
| 16 | SSIGM | $\sin\sigma$ | Sine control variable $\sigma$ |
| 17 | CTHET | $\cos\theta$ | Cosine state variable $\theta$ |
| 18 | STHET | $\sin\theta$ | Sine state variable $\theta$ |
| 19 | R | $r$ | Radial distance in spherical coordinates; feet |
| 20 | H | $h$ | Altitude; feet |
| 21 | VHSQ | $V_H^2$ | |
| 22 | VTSQ | $V_T^2$ | |
| 23 | VH | $V_H$ | $\sqrt{v^2+w^2}$ |
| 24 | $V_T$ | $V_T$ | Velocity; $\sqrt{u^2+v^2+w^2}$ ; feet/second |
| 25 | PEA | $P_\infty$ | Atmospheric pressure; pounds/foot$^2$ |
| 26 | RHO | $\rho$ | Atmospheric density; slugs/foot$^3$ |
| 27 | SOFS | $c$ | Atmospheric speed of sound; in feet/second |
| | | | The atmospheric variables are computed by subroutine ATM |
| 28 | MACH | $M$ | Mach number; real |
| 29 | MASS | $m$ | Mass; slugs; input |
| 30 | TH | $T_{SL}$ | Sea level thrust; pounds; input |
| 31 | THR | $T$ | Thrust adjusted for altitude; pounds |
| 32 | COSIT | $\cos i_T$ | |
| 33 | SINIT | $\sin i_T$ | |
| 34 | COSDT | $\cos \delta_T$ | |
| 35 | SINDT | $\sin \delta_T$ | |

| | | | |
|---|---|---|---|
| 36 | THRX | $T_x$ | Component of thrust |
| 37 | THRY | $T_Y$ | " " |
| 38 | THRZ | $T_z$ | " " |
| 39 | CALPH | $\cos\alpha$ | Cosine control variable $\alpha$ |
| 40 | SALPH | $\sin\alpha$ | Sine control variable $\alpha$ |
| 41 | C1 | | Common term; $\cos\alpha/V_T$ |
| 42 | C2 | | " " ; $\sin\alpha/V_T$ |
| 43 | C3 | | " " ; $\cos\sigma/V_T$ |
| 44 | C4 | | " " ; $\sin\sigma/V_T$ |
| 45 | C5 | | " " ; $w\sin\sigma/V_H+(uv\cos\sigma)/V_HV_T$ |
| 46 | C6 | | " " ; $v\sin\sigma/V_H-(uw\cos\sigma)/V_HV_T$ |
| 47 | THRR | $T_r$ | Component of thrust in $\hat{r}$ direction |
| 48 | THRT | $T_\theta$ | " " in $\hat{\theta}$ direction |
| 49 | THRP | $T_\phi$ | " " in $\hat{\phi}$ direction |
| 50 | S | S | Reference area |
| 51 | X1 | | Common term; $\rho SV_T/2m$ |
| 52 | X2 | | " " ; $\rho SV_TC_D/2m$ |
| 53 | X3 | | " " ; $\rho SV_TC_L/2m$ |
| 54 | X4 | | " " ; $(\rho SV_TC_L\cos\sigma)2m$ |
| 55 | X5 | | " " ; $(u SV_TC_L\cos\sigma)/2V_H m$ |
| 56 | X6 | | " " ; $(\rho SV_T^2C_L\sin\sigma)/2V_H m$ |
| 57 | X7 | | " " ; $-u/r$ |
| 58 | X8 | | " " ; $(w\cot\theta)/r$ |
| 59 | X9 | | " " ; $r\Omega_e^2\sin\theta + 2\Omega_e w$ |
| 60 | RSQ | | " " ; $r^2$ |
| 61 | R4TH | | " " ; $r^4$ |
| 62 | SQRHO | | $\sqrt{\rho}$ |

| | | | |
|---|---|---|---|
| 63 | ACELG | $a$ | Acceleration in g's |
| 64 | TAU | $\tau(a)$ | A function of acceleration; input |
| 65 | TIMEH | | Time when altitude equals AH2 (use by penalty function C) |
| 66 | CL | $C_L(M,\alpha)$ | Lift coefficient; a function of Mach and $\alpha$; input |
| 67 | CD | $C_D(M,\alpha)$ | Drag coefficient; a function of Mach and $\alpha$; input |
| 68 | TFINAL | | Time at end of forward trajectory |
| 69 | B56 | | Common term; $v^2+w^2$ |
| 70 | B58 | | Common term; $u^2+v^2+w^2$ |
| 71 | PEH | | Atmospheric pressure at h+500 ft.; pounds/foot$^2$ |
| 72 | PHOH | | Atmospheric density at h+500 ft.; slugs/foot$^3$ |
| 73 | SOFSH | | Atmospheric speed of sound at h+500 ft.;foot/sec. |
| 74 | PEL | | Atmospheric pressure at h-500 ft.;pounds/foot$^2$ |
| 75 | PHOL | | Atmospheric density at h-500 ft.; slugs/foot$^3$ |
| 76 | SOFSL | | Atmospheric speed of sound at h-500 ft.;feet/sec |
| 77 | DPERH | $dP_\infty/dh$ | Derivative atmospheric pressure respect to altitude |
| 78 | DRRH | $d\rho/dh$ | Derivative atmospheric density respect to altitude |
| 79 | DSRH | $dc/dh$ | Derivative atmospheric speed of sound respect to altitude |
| 80 | CLHM | | $C_L(\alpha,M+.05)$ |
| 81 | CDHM | | $C_D(\alpha,M+.05)$ |
| 82 | CLLM | | $C_L(\alpha,M-.05)$ |
| 83 | CDLM | | $C_D(\alpha,M-.05)$ |
| 84 | PCLRM | $\partial C_L/\partial M$ | Partial of $C_L$ respect to Mach |
| 85 | PCDRM | $\partial C_D/\partial M$ | Partial of $C_D$ respect to Mach |
| 86 | DRPRT | $dR/d\theta$ | $2(R_e-R_{PC})\cos\theta\ \sin\theta$; derivative of radius of earth respect to $\theta$ |

| | | |
|---|---|---|
| 87 | B10 | Common term; $\rho\ S/2m$ |
| 88 | B20 | " " ; $C_D V_T$ |
| 89 | B21 | " " ; $C_L V_T$ |
| 90 | B30 | " " ; $(C_D + M\partial C_D/\partial M)/V_T$ |
| 91 | B31 | " " ; $(C_L + M\partial C_L/\partial M)/V_T$ |
| 92 | B40 | " " ; $(\cos\sigma)/V_H$ |
| 93 | B41 | " " ; $(\sin\sigma)/V_H$ |
| 94 | B50 | " " ; $u^2$ |
| 95 | B51 | " " ; $v^2$ |
| 96 | B52 | " " ; $w^2$ |
| 97 | B53 | " " ; $uv$ |
| 98 | B54 | " " ; $uw$ |
| 99 | B55 | " " ; $vw$ |
| 100 | B57 | " " ; $uvw$ |
| 101 | B60 | " " ; $u/r$ |
| 102 | B61 | " " ; $v/r$ |
| 103 | B62 | " " ; $w/r$ |
| 104 | B70 | " " ; $2C_D + M\partial C_D/\partial M$ |
| 105 | B71 | " " ; $2C_L + M\partial C_L/\partial M$ |
| 106 | B81 | " " ; $2\Omega_e \cos\theta$ |
| 107 | B82 | " " ; $2\Omega_e \sin\theta$ |
| 108 | B90 | " " ; $(\cot\theta)/r$ |
| 109 | B91 | " " ; $\cos\theta\ \sin\theta$ |
| 110 | B92 | " " ; $\cos^2\theta$ |
| 111 | B93 | " " ; $\sin^2\theta$ |
| 112 | B100 | " " ; $(\rho\ M/c)dc/dh$ |

| | | | |
|---|---|---|---|
| 113 | B101 | | Common term; $C_D\ d\rho/dh-(\rho M/c)(dc/dh)(\partial C_D/\partial M)$ |
| 114 | B102 | | " " ; $C_L\ d\rho/dh-(\rho M/c)(dc/dh)(\partial C_L/\partial M)$ |
| 115 | B103 | | " " ; $S/2m$ |
| 116 | B104 | | " " ; $\Omega_e^2 \sin\theta$ |
| 117 | R3RD | $r^3$ | |
| 118 | R5TH | $r^5$ | |
| 119 | VT3 | $V_T^3$ | |
| 120 | B200 | | Common term |
| 121 | B201 | | " " |
| 122 | VH3 | $V_H^3$ | |
| 123 | B202 | | Common term |
| 124 | B204 | | " " |
| 125 | PTRRU | $\partial T_r/\partial u$ | |
| 126 | PTTRU | $\partial T_\theta/\partial u$ | |
| 127 | PTPRU | $\partial T_\phi/\partial u$ | |
| 128 | PTRRV | $\partial T_r/\partial v$ | |
| 129 | PTTRV | $\partial T_\theta/\partial v$ | |
| 130 | PTPRV | $\partial T_\phi/\partial v$ | |
| 131 | PTRRW | $\partial T_r/\partial w$ | |
| 132 | PTTRW | $\partial T_\theta/\partial w$ | |
| 133 | PTPRW | $\partial T_\phi/\partial w$ | |
| 134 | PTHRH | $dP_\infty/dh$ | Derivative of atmospheric pressure respect to altitude |
| 135 | PTXRH | $\partial T_x/\partial h$ | |
| 136 | PTYRH | $\partial T_y/\partial h$ | |
| 137 | PTZRH | $\partial T_z/\partial h$ | |
| 138 | B300 | | Common term |
| 139 | B301 | | " " |
| 140 | B302 | | " " |

| | | |
|---|---|---|
| 141 | PTRRR | $\partial T_r/\partial r$ |
| 142 | PTTRR | $\partial T_\theta/\partial r$ |
| 143 | PTPRR | $\partial T_\phi/\partial r$ |
| 144 | B404 | Common term |
| 145 | B405 | " " |
| 146 | B400 | " " |
| 147 | SCLCD | $(C_L{}^2+C_D{}^2)^{1/2}$ |
| 148 | DTRA | $d\tau(a)/da$ |
| 149 | B401 | Common term; $a\ (d\tau(a)\ /da)/\tau(a)^2$ |
| 150 | B402 | " " |
| 151 | B403 | " " |
| 152 | B406 | " " |
| 153 | B407 | " " |
| 154 | PTRRA | $\partial T_r/\partial\alpha$ |
| 155 | PTTRA | $\partial T_\theta/\partial\alpha$ |
| 156 | PTPRA | $\partial T_\phi/\partial\alpha$ |
| 157 | PTRRS | $\partial T_r/\partial\sigma$ |
| 158 | PTTRS | $\partial T_\theta/\partial\sigma$ |
| 159 | PTPRS | $\partial T_\phi/\partial\sigma$ |
| 160 | E1 | Common term; $= (\cos\alpha)/V_T$ |
| 161 | E2 | " " ; $= (\sin\alpha)/V_T$ |
| 162 | E3 | " " ; $= V_H \cos\sigma$ |
| 163 | E4 | " " ; $= V_H \sin\sigma$ |
| 164 | E5 | " " ; $= (\cos\sigma)/V_H$ |
| 165 | E6 | " " ; $= (\sin\sigma)/V_H$ |
| 166 | CLAL | $C_L(\alpha-.005,M)$ |
| 167 | CDAL | $C_D(\alpha-.005,M)$ |

| | | | |
|---|---|---|---|
| 168 | CLAH | | $C_L(\alpha+.005,M)$ |
| 169 | CDAH | | $C_D(\alpha+.005,M)$ |
| 170 | PCLRA | $\partial C_L/\partial\alpha$ | |
| 171 | PCDRA | $\partial C_D/\partial\alpha$ | |
| 172 | B130 | | Common term |
| 173 | B131 | | " " |
| 174 | PTRG | | Print trigger variable |
| 175 | STRG | | Staging trigger variable |
| 176 | DTRG | | LAMCOS-DO trigger variable |
| 177 | GTRG | | Influence function trigger |
| 178 | SRAT | | Stopping trigger. Forward trajectory is terminated when SRAT = 0 |
| 179 | MUD | $\mu$ | Coordinate angle; degrees; real |
| 180 | NUD | $\nu$ | Coordinate angle; degrees; real |
| 181 | PHISD | $\phi^*$ | Coordinate angle; degrees |
| 182 | THETSD | $\theta^*$ | " " ; " |
| 183 | BETAD | $\beta$ | Heading angle; degrees |
| 184 | GAMMAD | $\gamma$ | Flight path angle; degrees |
| 185 | THETD | $\theta$ | State variable; degrees |
| 186 | PHID | $\phi$ | State variable; degrees |
| 187 | DYNA | | Dynamic pressure; lbs./ft.$^2$ |
| 188 | ENER | | Energy; ft./lbs. |
| 189 | AST | | Actual integration step-size used by MARK; seconds |
| 190 | RG | | MARK truncation error |
| 191 | SFRM(1500) | | Storage for MARK |
| 191 | NORA | | Order of forward integration |
| 194 | HNA | | Nominal forward integration step size; seconds |
| 195 | NEQA | | Number of equations to be integrated |
| 196 | TIMEA | t | Independent variable forward integration; seconds |

| | | | |
|---|---|---|---|
| 198 | U | u | State variable; velocity component; ft./sec |
| 199 | V | v | " " ; " " ; " " |
| 200 | W | w | " " ; " " ; " " |
| 201 | RX | | State variable; radius; ft.; = $r-R_e$ |
| 202 | THETR | $\theta$ | " " ; co-latitude; degrees |
| 203 | PHIR | $\phi$ | " " ; longitude; degrees |
| 204 | PA | Q | Heat |
| 205 | PB | $p_a$ | Acceleration penalty function |
| 206 | PC | $p_H$ | Altitude penalty function |
| 207 | PD | $p_{QD}$ | Heat-rate penalty function |
| 208 | PE | | Null |
| 209 | PF | | Null |
| 210 | U1 | $\dot{u}$ | Derivative |
| 211 | V1 | $\dot{v}$ | " |
| 212 | W1 | $\dot{w}$ | " |
| 213 | RX1 | $\dot{r}$ | " |
| 214 | THETR1 | $\dot{\theta}$ | " |
| 215 | PHIR1 | $\dot{\phi}$ | " |
| 216 | PA1 | | " |
| 217 | PB1 | | " |
| 218 | PC1 | | " |
| 219 | PD1 | | " |
| 220 | PE1 | | " |
| 221 | PF1 | | " |
| 210 | NORB | | Order of adjoint integration |
| 213 | HNB | | Nominal adjoint integration step size; in seconds |

| | | |
|---|---|---|
| 214 | NEQB | Maximum number of equations to be integrated |
| 215 | TIMEB | Independent variable adjoint integration |
| 217 | ADJ(6,9) | Variable adjoint differential equations. 1st subscript for adjoint $\lambda_u, \lambda_v, \lambda_w, \lambda_r, \lambda_\theta$, and $\lambda_\phi$, 2nd subscript for payoff function and 0 to 8 constraints |
| 271 | DADJ(6,9) | Derivatives for the adjoint differential equations |
| 1691 | GL(9,2,201) | Storage for influence functions. 1st subscript for payoff function and 0 to 8 constraints. 2nd subscript for the two control variables, $\alpha$ and $\sigma$. 3rd subscript for the 201 equal-time-spaced points. |
| 5309 | ATABS(201) | Storage for control variable $\alpha$; radians |
| 5510 | STABS(201) | Storage for control variable $\sigma$; radians |
| 5711 | PARTS(14) | Storage for partial derivatives of stopping function. PARTS(14) = total derivative |
| 5725 | PARTC(14) | Storage for partial derivatives of payoff and constraint functions. PARTC(14) = total derivative |
| 5739 | CADJ(6,9) | Storage for constant adjoint differential equations. 1st subscript for the constant adjoint differential equations. 2nd subscript for payoff and 0 to 8 constraints |
| 5793 | F(6,6) | F matrix associated with variable adjoint differential equations |
| 5793 | PFURU | $\partial\dot{u}/\partial u$ |
| 5794 | PFVRU | $\partial\dot{v}/\partial u$ |
| 5795 | PFWRU | $\partial\dot{w}/\partial u$ |
| 5796 | PFRRU | $\partial\dot{r}/\partial u$ |
| 5797 | PFTRU | $\partial\dot{\theta}/\partial u$ |
| 5798 | PFPRU | $\partial\dot{\phi}/\partial u$ |
| 5799 | PFURV | $\partial\dot{u}/\partial v$ |

| | | |
|---|---|---|
| 5800 | PFVRV | $\partial\mathring{v}/\partial v$ |
| 5801 | PFWRV | $\partial\mathring{w}/\partial v$ |
| 5802 | PFRRV | $\partial\mathring{r}/\partial v$ |
| 5803 | PFTRV | $\partial\mathring{\theta}/\partial v$ |
| 5804 | PFPRV | $\partial\mathring{\phi}/\partial v$ |
| 5805 | PFURW | $\partial\mathring{u}/\partial w$ |
| 5806 | PFVRW | $\partial\mathring{v}/\partial w$ |
| 5807 | PFWRW | $\partial\mathring{w}/\partial w$ |
| 5808 | PFRRW | $\partial\mathring{r}/\partial w$ |
| 5809 | PFTRW | $\partial\mathring{\theta}/\partial w$ |
| 5810 | PFPRW | $\partial\mathring{\phi}/\partial w$ |
| 5811 | PFURR | $\partial\mathring{u}/\partial r$ |
| 5812 | PFVRR | $\partial\mathring{v}/\partial r$ |
| 5813 | PFWRR | $\partial\mathring{w}/\partial r$ |
| 5814 | PFRRR | $\partial\mathring{r}/\partial r$ |
| 5815 | PFTRR | $\partial\mathring{\theta}/\partial r$ |
| 5816 | PFPRR | $\partial\mathring{\phi}/\partial r$ |
| 5817 | PFURT | $\partial\mathring{u}/\partial\theta$ |
| 5818 | PFVRT | $\partial\mathring{v}/\partial\theta$ |
| 5819 | PFWRT | $\partial\mathring{w}/\partial\theta$ |
| 5820 | PFRRT | $\partial\mathring{r}/\partial\theta$ |
| 5821 | PFTRT | $\partial\mathring{\theta}/\partial\theta$ |
| 5822 | PFPRT | $\partial\mathring{\phi}/\partial\theta$ |
| 5823 | PFURP | $\partial\mathring{u}/\partial\phi$ |
| 5824 | PFVRP | $\partial\mathring{v}/\partial\phi$ |
| 5825 | PFWRP | $\partial\mathring{w}/\partial\phi$ |
| 5826 | PFRRP | $\partial\mathring{r}/\partial\phi$ |
| 5827 | PFTRP | $\partial\mathring{\theta}/\partial\phi$ |
| 5828 | PFPRP | $\partial\mathring{\phi}/\partial\phi$ |

| | | | |
|---|---|---|---|
| 5829 | CF(6,6) | | F matrix associated with constant adjoint differential equations |
| 5829 | PFARU | $\partial\dot{a}/\partial u$ | |
| 5830 | PFBRU | $\partial\dot{b}/\partial u$ | (Note: In 5829-5864 and 5877-5888, a, b, c, d, e, f, represent the functions PA, PB, . . . PF respectively; See 204-209) |
| 5831 | PFCRU | $\partial\dot{c}/\partial u$ | |
| 5832 | PFDRU | $\partial\dot{d}/\partial u$ | |
| 5833 | PFERU | $\partial\dot{e}/\partial u$ | |
| 5834 | PFFRU | $\partial\dot{f}/\partial u$ | |
| 5835 | PFARV | $\partial\dot{a}/\partial v$ | |
| 5836 | PFBRV | $\partial\dot{b}/\partial v$ | |
| 5837 | PFCRV | $\partial\dot{c}/\partial v$ | |
| 5838 | PFDRV | $\partial\dot{d}/\partial v$ | |
| 5839 | PFERV | $\partial\dot{e}/\partial v$ | |
| 5840 | PFFRV | $\partial\dot{f}/\partial v$ | |
| 5841 | PFARW | $\partial\dot{a}/\partial w$ | |
| 5842 | PFBRW | $\partial\dot{b}/\partial w$ | |
| 5843 | PFCRW | $\partial\dot{c}/\partial w$ | |
| 5844 | PFDRW | $\partial\dot{d}/\partial w$ | |
| 5845 | PFERW | $\partial\dot{e}/\partial w$ | |
| 5846 | PFFRW | $\partial\dot{f}/\partial w$ | |
| 5847 | PFARR | $\partial\dot{a}/\partial r$ | |
| 5848 | PFBRR | $\partial\dot{b}/\partial r$ | |
| 5849 | PFCRR | $\partial\dot{c}/\partial r$ | |
| 5850 | PFDRR | $\partial\dot{d}/\partial r$ | |
| 5851 | PFERR | $\partial\dot{e}/\partial r$ | |
| 5852 | PFFRR | $\partial\dot{f}/\partial r$ | |
| 5853 | PFART | $\partial\dot{a}/\partial\theta$ | |
| 5854 | PFBRT | $\partial\dot{b}/\partial\theta$ | |
| 5855 | PFCRT | $\partial\dot{c}/\partial\theta$ | |
| 5856 | PFDRT | $\partial\dot{d}/\partial\theta$ | |
| 5857 | PFERT | $\partial\dot{e}/\partial\theta$ | |
| 5858 | PFFRT | $\partial\dot{f}/\partial\theta$ | |

| | | | |
|---|---|---|---|
| 5859 | PFARP | $\partial\dot{a}/\partial\phi$ | |
| 5860 | PFBRP | $\partial\dot{b}/\partial\phi$ | |
| 5861 | PFCRP | $\partial\dot{c}/\partial\phi$ | |
| 5862 | PFDRP | $\partial\dot{d}/\partial\phi$ | |
| 5863 | PFERP | $\partial\dot{e}/\partial\phi$ | |
| 5864 | PFFRP | $\partial\dot{f}/\partial\phi$ | |
| 5865 | G(6,2) | | G matrix associated with variable adjoint differential equations |
| 5865 | PFURA | $\partial\dot{u}/\partial\alpha$ | |
| 5866 | PFVRA | $\partial\dot{v}/\partial\alpha$ | |
| 5867 | PFWRA | $\partial\dot{w}/\partial\alpha$ | |
| 5868 | PFRRA | $\partial\dot{r}/\partial\alpha$ | |
| 5869 | PFTRA | $\partial\dot{\theta}/\partial\alpha$ | |
| 5870 | PFPRA | $\partial\dot{\phi}/\partial\alpha$ | |
| 5871 | PFURS | $\partial\dot{u}/\partial\sigma$ | |
| 5872 | PFVRS | $\partial\dot{v}/\partial\sigma$ | |
| 5873 | PFWRS | $\partial\dot{w}/\partial\sigma$ | |
| 5874 | PFRRS | $\partial\dot{r}/\partial\sigma$ | |
| 5875 | PFTRS | $\partial\dot{\theta}/\partial\sigma$ | |
| 5876 | PFPRS | $\partial\dot{\phi}/\partial\sigma$ | |
| 5877 | CG(6,2) | | G matrix associated with constant adjoint differential equations |
| 5877 | PFARA | $\partial\dot{a}/\partial\alpha$ | (See Note at 5829) |
| 5878 | PFBRA | $\partial\dot{b}/\partial\alpha$ | |
| 5879 | PFCRA | $\partial\dot{c}/\partial\alpha$ | |
| 5880 | PFDRA | $\partial\dot{d}/\partial\alpha$ | |
| 5881 | PFERA | $\partial\dot{e}/\partial\alpha$ | |
| 5882 | PFFRA | $\partial\dot{f}/\partial\alpha$ | |
| 5883 | PFARS | $\partial\dot{a}/\partial\sigma$ | |
| 5884 | PFBRS | $\partial\dot{b}/\partial\sigma$ | |
| 5885 | PFCRS | $\partial\dot{c}/\partial\sigma$ | |

| | | | |
|---|---|---|---|
| 5886 | PFDRS | $\partial\dot{d}/\partial\sigma$ | |
| 5887 | PFERS | $\partial\dot{e}/\partial\sigma$ | |
| 5888 | PFFRS | $\partial\dot{f}/\partial\sigma$ | |
| 5889 | LAMP(9) | | LAMP*TFINAL give times, in seconds, for LAMCOS-DO |
| 5898 | VARX(25) | | Extra space for additional variables |
| 5898 | LIFT | | $\rho S V_T^2 C_L/2$ |
| 5899 | DRAG | | $\rho S V_T^2 C_D/2$ |
| 5900 | DA | $\delta\alpha$ | |
| 5901 | DS | $\delta\alpha$ | |

| COMMON / ANAL / | | | COMMON NO. 3 |
|---|---|---|---|
| 1 | HI | $\phi$ | Value of payoff function |
| 2 | HIS | $\phi^*$ | Value of payoff function last acceptable iteration |
| 3 | SIA(8) | $\psi$ | Matrix of values of terminal constraint functions |
| 11 | SIAS(8) | $\psi^*$ | Matrix of values of terminal constraint functions for last acceptable iteration |
| 19 | SIE(8) | $\psi_e$ | Matrix of the errors between the actual values of $\psi$ and the desired values |
| 27 | SIES(8) | $\psi_e^*$ | Matrix of the errors between the actual values of $\psi$ and the desired values for last acceptable iteration |
| 35 | CSIA | $\Psi_A$ | Composite performance indicator for constraints only |
| 36 | CSIAS | $\Psi_A{}^*$ | Composite performance indicator last acceptable iteration |
| 37 | CDSIP | $d\Psi_p$ | Predicted change in $\Psi$ |
| 38 | CDSIA | $d\Psi_A$ | Actual change in $\Psi$ |
| 39 | CRDSI | $R_{d\Psi}$ | Ratio of $d\Psi_A$ to $d\Psi_p$ |
| 40 | CHIA | $\underline{\Phi}_A$ | Composite performance indicator |
| 41 | CHIAS | $\underline{\Phi}^*_A$ | Composite performance indicator for last acceptable iteration |

| | | | |
|---|---|---|---|
| 42 | CDHIA | $d\underline{\Phi}_A$ | Actual change in $\underline{\Phi}$ |
| 43 | CDHIP | $d\underline{\Phi}_P$ | Predicted change in $\underline{\Phi}$ |
| 44 | CRDHI | $Rd_\Phi$ | Ratio of $d\underline{\Phi}_A$ to $d\underline{\Phi}_P$ |
| 45 | DSIA(8) | $d\psi_A$ | Actual change in terminal constraints |
| 53 | RDSI(8) | $R_{d\psi}$ | Ratio of actual change over predicted change in terminated constraints |
| 61 | DHIA | $d\Phi_A$ | Actual change in payoff function |
| 62 | DHIP | $d\Phi_P$ | Predicted change in payoff function |
| 63 | RDHI | $R_{d\Phi}$ | Ratio of actual to predicted change in payoff function |
| 64 | DP | dP | Square root of steepest-ascent step size |
| 65 | EHI | $\Delta\Phi$ | Actual gradient of payoff function |
| 66 | CEHI | $\Delta\underline{\Phi}$ | Actual composite gradient |
| 147 | INTGL(9,9) | | Storage for computation of integrals; real |
| 148 | IHH | $I_{\Phi\Phi}$ | Real |
| 150 | ISH(8) | $I_{\psi\Phi}$ | Real |
| 158 | ISS(8,8) | $I_{\psi\psi}$ | Real |
| 222 | WT(2,2) | | Weighting function |
| 226 | F1 | $F_1$ | $(dP)^2 - \psi'_{dL} I^{-1}_{\psi\psi L} \psi_{dL}$ |
| 227 | F2 | $F_2$ | $F1/(I_{\phi\phi} - I'_{\psi\phi L} I^{-1}_{\psi\psi L} I_{\psi\phi L})$ |
| 228 | SID(8) | $\psi_d$ | Desired change in terminal constraints |
| 237 | SFC | $S_{FC}$ | Scale factor |
| 238 | DSI(8) | $d\psi$ | Predicted change in terminal constraints |
| 246 | DSIJ | | Predicted change in terminal constraint that is inactive |
| 254 | DBETA(8) | | $d\psi - \lambda'_{\psi\Omega o} \delta x_o$ |
| 262 | ISSIL(8,8) | $I^{-1}_{\psi\psi L}$ | |
| 326 | DPSQ | $(dP)^2$ | Steepest-ascent step size |
| 327 | DPSQK | $(dP)^2_k$ | Steepest-ascent step size left; LAMCOS-DO |
| 328 | IHHK | $I_{\Phi\Phi k}$ | Real |

| | | | |
|---|---|---|---|
| 329 | ISHK(8) | $I_{\psi\Phi k}$ | Real |
| 337 | ISSK(8,8) | $I_{\psi\psi k}$ | Real |
| 401 | ISSILK(8,8) | $I^{-1}_{\psi\psi kL}$ | Real |
| 465 | DBETAK(8) | $d\beta_k$ | |
| 473 | STVRS(12,9) | | Storage for the state variables to be used by LAMCOS-DO |
| 581 | LAMBDA(6,8,9) | | Storage for variable adjoint differential equations to be used by LAMCOS-DO. 1st subscript adjoint differential equations. 2nd subscript 0 to 8 constraints. 3rd subscript 1 to 9 times; Real |
| 1013 | SFCK | $S_{Fk}$ | |
| 1014 | F1K | $F_{1k}$ | |
| 1015 | F2K | $F_{2k}$ | |

| COMMON / IVAR / | | COMMON NO. 4 |
|---|---|---|
| 1 | JTAPE | Print output tape 6 |
| 2 | KTAPE | Not used |
| 3 | LTAPE | Scratch tape 8 |
| 4 | ITN | Iteration count |
| 5 | ITTN | Trial iteration count |
| 6 | ISTAGE | Boost stage count |
| 7 | NC | Number of constraints |
| 8 | NT | Number of constraints + 1 (for payoff function) |
| 9 | NRTN | Return indicator from MARK |
| 10 | NRTG | Trigger return indicator from MARK |
| 11 | KACC | Iteration acceptable indicator<br>1 = acceptable, 2 = not acceptable |
| 12 | KA | 1 = variable $(dP)^2$, 2 = fixed $(dP)^2$ |

| | | |
|---|---|---|
| 13 | KS | Case stop indicator<br>1 = Do not stop<br>2 = Requested number of iterations complete<br>3 = Composite gradient less than minimum value $N_{\Delta F}$ times<br>4 = Current iteration acceptable but new $(dP)^2 < (dP)^2_{MIN}$<br>5 = Current iteration unacceptable but new $(dP)^2 < .01(dP)^2_{MIN}$<br>6 = Trial iteration > 5 |
| 14 | NIF | Number of equal-time-spaced points to save influence functions along the trajectory. NIF=201 |
| 15 | NG | Influence function count runs from 201 to 1 |
| 16 | L(8) | Active constraint matrix<br>1 = yes, 0 = no |
| 24 | LS(8) | Active constraint matrix last acceptable iteration |

COMMON / FORW / STRAJ(20,13) Storage for the forward trajectory written and read from tape. Referenced by decks TOS, SFR, and ADT

COMMON / ISTATE /

| | | |
|---|---|---|
| 1 | UN | $u_0$ |
| 2 | VN | $v_0$ |
| 3 | WN | $w_0$ |
| 4 | RXN | $r_0-R_e$ |

| | | |
|---|---|---|
| 5 | THETN | $\theta_0$ |
| 6 | PHIN | $\phi_0$ |
| 7 | VHN | $V_{H_0}$ |
| 8 | CTHTN | $\cos\theta_0$ |
| 9 | STHTN | $\sin\theta_0$ |