AD-A034 920 ARMY MISSILE RESEARCH DEVELOPMENT AND ENGINEERING LAB--ETC F/G 19/7

FREE ROCKET SIX DEGREE OF FREEDOM DIGITAL SIMULATION.(U)

NOV 76 M P STAMPS

UNCLASSIFIED RF-77-1 NL

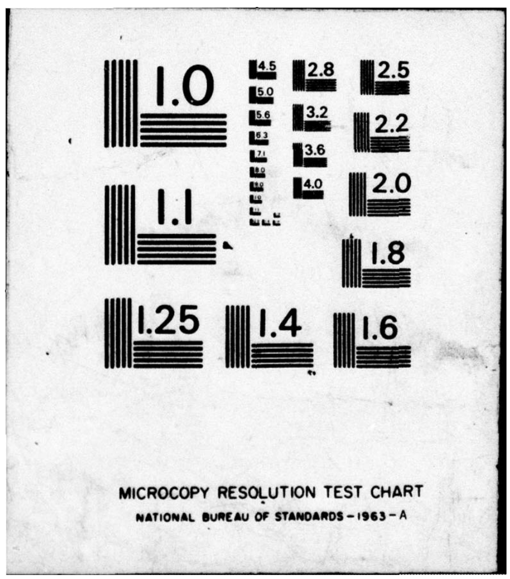
| OF |

AD A034920

END

DATE FILMED

3-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

6

TECHNICAL REPORT RF-77-1

FREE ROCKET SIX DEGREE OF FREEDOM DIGITAL SIMULATION

Mickie P. Stamps
Advanced Systems Concepts Office
US Army Missile Research, Development and Engineering Laboratory
US Army Missile Command
Redstone Arsenal, Alabama 35809

8 November 1976

Approved for public release; distribution unlimited

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35809

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RF-77-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FREE ROCKET SIX DEGREE OF FREEDOM DIGITAL SIMULATION.		5. TYPE OF REPORT & PERIOD COVERED Technical Report.
		6. PERFORMING ORG. REPORT NUMBER RF-77-1
7. AUTHOR(s) Mickie P. Stamps		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Commander US Army Missile Command Attn: DRSMI-RF Redstone Arsenal, Ala. 35809		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA 1W362303A214 AMCMS 632303.214111.01
11. CONTROLLING OFFICE NAME AND ADDRESS Commander US Army Missile Command Attn: DRSMI-RPR Redstone Arsenal, Ala. 35809		12. REPORT DATE 8 November 1976
		13. NUMBER OF PAGES 32 (12) 34p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Free rocket simulation
Six degrees of freedom
Equations of motion
Digital simulation

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

This report describes in detail a free rocket six degree of freedom trajectory program which utilizes a fourth order RUNGE-KUTTA subroutine. Several options such as programmed roll, directional control, and fold out fins are described. The program includes a listing of the equations, flow charts, program description, input and output information, and a complete program listing.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 401 195

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

I. INTRODUCTION

The purpose of this report is to describe a six degree of freedom missile trajectory program in sufficient detail for the user to be able to run the program and/or make modifications to fit his specific needs. This program utilizes a nonrolling missile coordinate system to speed up run time when high (greater than 15 cps) roll rates occur. Several options are available such as directional control force, fold-out fins, and print-out of steady-state angles and velocity angular errors at boost burn out.

The program is in modular form. All variables to be integrated are stored in an array. A subroutine, DESUB, consists of the twelve simultaneous differential equations describing the behavior of these variables. A general RUNGE-KUTTA subroutine is utilized to integrate this set of equations. This report includes a listing of these equations, program description and flow charts, definitions of all symbols used, and a complete program listing.

II. PROGRAM DESCRIPTION

A. Equations of Motion

This program was written to simulate the following equations of motion to describe a free rocket trajectory from ignition to impact.

$$\dot{u} = (T - 1/2\rho S C_D u^2)/m + rv - qw - g\,SIN\,(QE + \theta)$$

$$\dot{v} = -ru + (1/2\rho S u^2 C_{N\alpha}\left[(v - w_v)/(u - w_u)\right] + T\delta_{tm} COS\,(\phi + \phi_o) + k\psi)/m$$

$$\dot{w} = g\,COS\,(QE + \theta) + qu + (-1/2\rho S U^2 C_{N\alpha}\left[(w - w_w)/(u - w_u)\right] + T\delta_{tm} SIN\,(\phi + \phi_o) - K\theta)/m$$

$$\dot{r} = (1/2\rho S d u^2 l_s C_{N\alpha}\left[(v - w_v)/(u - w_u)\right] - 1/4\rho S d^2 ru C_{mq} - T\delta_{tm} l_T d\,COS\,(\phi + \phi_o) - k l_c d\psi + I_x pq)/I_y$$

$$\dot{q} = (1/2\rho S d u^2 l_s C_{n\alpha}\left[(w - w_w)/(u - w_u)\right] - 1/4\rho S d^2 qu C_{mq} + T\delta_{tm} d l_T SIN\,(\phi + \phi_o) - k l_c d\theta - I_x pr)/I_y$$

$$\dot{p} = (1/2\rho S d u^2 C_{l\delta}\delta_c - 1/4\rho S d^2 pu C_{lp} + 1/2\rho S d u^2 C_{l0})/I_x + P_{spin}$$

$$\dot{x} = u\mathrm{COS}\ (QE + \theta) + w\mathrm{SIN}\ (QE + \theta)\ \mathrm{COS}\psi - v\mathrm{SIN}\psi$$

$$\dot{y} = u\mathrm{COS}\ (QE + \theta)\ \mathrm{SIN}\psi + w\mathrm{SIN}\ (QE + \theta)\ \mathrm{SIN}\psi + v\mathrm{COS}\psi$$

$$\dot{z} = w\mathrm{COS}\ (QE + \theta) - u\mathrm{SIN}\ (QE + \theta)$$

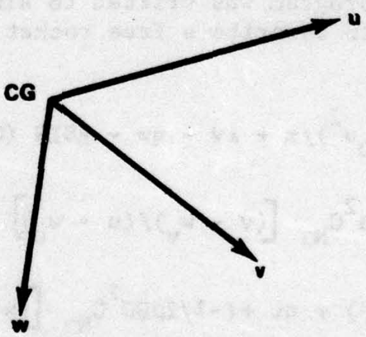
$$\dot{\theta} = q$$

$$\dot{\psi} = r/\mathrm{COS}\ (QE + \theta)$$

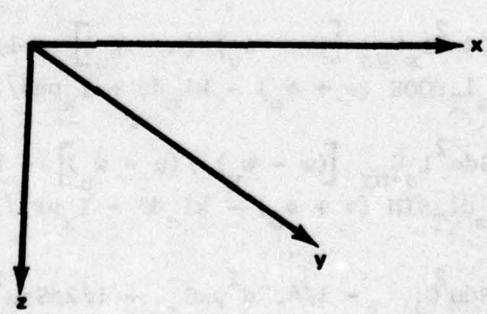
$$\dot{\phi} = p$$

If a constant roll rate or a programmed roll rate is to be implemented, this can be done by setting $p = 0$ and $\dot{\phi} = A_{spin}$. This is noted in the program listing. In post flight analysis, frequently roll history can be acquired from telemetry data. This can be stored in tabular data in A_{spin} and utilized in the preceding manner.

These equations were converted to Fortran code by using the Fortran names listed in Table 1. These equations are calculated in the following coordinate systems:



Missile coordinate system.



Earth coordinate system.

B. Program Outline

Solving the equations of motion requires current values of necessary atmospheric, aerodynamic, and missile physical characteristic data. The program is set up to read these data in by way of NAMELIST. This is one of the easiest methods of inputting data. An example of a data card for a constant is

x = 75.,

This is done with no regard for format except that only columns 2 through 72 may be used. If x is an array of five numbers, it is read as:

x (1) = 6., 4., 1., 17., 0.,

This can be continued onto the next card is necessary. All Fortran names used in the program are defined in Table 2. Program inputs are listed in Table 3.

Next, all variables to be integrated as a funtion of time are stored in an array, VARTB. VARTB (1) is time. The remainder of the array is ordered consistent with the order of the corresponding differential equations in DESUB.

The current values of atmospheric data are now calculated using the following equations:

$$G = G_0 \left[1/(1 - z/20855531.5)\right]^2$$

$$A = -z/(1 - z/20855531.5)$$

If $A \leq 36098.2388$,

$$T = T_0 + (-0.00356616\ A)$$

$$P = P_0\ (T/T_0)^{0.1633654G_0}$$

$$\rho = 28.966P/49719.6257T$$

$$V_s = \sqrt{2403.075\ T}$$

If $A > 36089.2388$,

$$T = 389.9878$$

$$P = 472.6765\ e^{-0.00000149G_0\ (A-36089.2388)}$$

ρ = 28.966 P/19390047.44

V_s = 968.075

where G = gravity

T = temperature

P = atmospheric pressure

ρ = atmospheric density

V_s = speed of sound.

The subscript 0 denotes the value of each of these variables at sea level.

Next, the wind components are transformed from the earth to missile coordinate system. These values are used to calculate missile total velocity and angles of attack.

Current values of missile physical characteristics and aerodynamic data are now calculated. The subroutine XINT is utilized for this purpose. This is simply a linear interpolation routine with one independent variable. Logic is employed in this section to determine if the missile is in boost or coast phase and whether fins have been unfolded.

Now the fourth-order RUNGE-KUTTA subroutine, SOLVDE/SSIMDE* is called. The integration time step, DELT, is an input to this subroutine. A rule to follow in determining the size of the time step is not to allow any angle to change more than 7° to 8° during one step. This does not include the roll angle unless a thrust misalignment case is being run. The variable DELMIN is the minimum the program will allow DELT to be changed to. This is done in SOLVDE/SSIMDE. DELMIN can be set to the same size as DELT. Other inputs to this routine are VARTB, the table of variables to be integrated; WWW, the dummy working array; IC; and N, the number of variables to be integrated. DESUB, the routine containing all the differential equations to be solved, is also required as an input argument. SOLVDE/SSIMDE is a two-purpose routine for integration. It has the capability of solving one nth order differential equation or n first order simultaneous differential equations.

After returning from the integration subroutine, several variables are calculated for output and printed if it is time for printing. If the missile altitude has become positive (below sea level) or if maximum

*Sellers, William R., SOLVDE/SSIMDE, Directorate for Management Information Systems, US Army Missile Command.

simulation time has been reached, the program searches for the next case of data. Otherwise, it loops back to calculate atmospheric data and hence back through the program. Program output is defined in Table 4.

C. Subroutine DESUB

This subroutine contains all differential equations to be solved in the order that the variables are stored in VARTB in the main program. The subroutine inputs are V which corresponds to VARTB in MAIN, and J, the number of differential equations. A computed GO TO as a function of J controls the flow through the routine. For example, if J = 1, the first differential equation is solved and control is returned to the integration routine. The output is F which is the calculated value of the differential equation for the current value of J. Following is an example of a simple DESUB for solving three simultaneous differential equations, $\dot{V}(1) = x$, $\dot{V}(2) = y$, and $\dot{V}(3) = z$.

```
      SUBROUTINE DESUB (V, F, J)
      COMMON X,Y,Z
      DIMENSION V(4)
      GO TO (10, 20, 30), J
10    F = X
      RETURN
20    F = Y
      RETURN
30    F = Z
      RETURN
      END
```

TABLE 1. FORTRAN NAMES

Fortran Name	Symbol
VARTB (1)	time
VARTB (2)	w
VARTB (3)	u
VARTB (4)	v
VARTB (5)	r
VARTB (6)	q
VARTB (7)	x
VARTB (8)	y
VARTB (9)	z
VARTB (10)	θ
VARTB (11)	ψ
VARTB (12)	p
VARTB (13)	ϕ
XT	T
P	ρ
S	S
CD	C_D
XM	m
GRAV	g
QE	QE
CNA	$C_{N\alpha}$
WV	w_v
WU	w_u
WW	w_w
DTM	δ_{tm}
XK	k
DX	d
SL	l_s
CMQ	C_{mq}

Table 1. (Concluded)

Fortran Name	Symbol
TL	l_T
CLA	l_c
ROLLI	I_x
XI	I_y
CLD	$C_{l\delta}$
DCANT	δ_c
CLP	C_{lp}
CLO	C_{l0}
PSPIN	P_{spin}

TABLE 2. SYMBOLS

Fortran Name	Definition
TEMPO	Temperature at sea level
PRSSO	Atmospheric pressure at sea level
GRAVO	Gravity at sea level
THETA	Flight path angle, pitch
IC	Initial condition for RUNGE-KUTTA routine
VARTB	Table of variables to be integrated
TT	Current value of time
WO	Initial velocity component of missile in pitch plane
U	Initial velocity component of missile along the longitudinal axis
VO	Initial velocity component of missile in yaw plane
RO	Initial yaw rate of missile
QO	Initial pitch rate of missile
PROLL	Initial roll rate of missile
XR	Initial range of missile in earth coordinate system
Y	Initial lateral displacement in earth coordinate system
Z	Initial vertical displacement in earth coordinate system
THETO	Initial pitch angle
PSIO	Initial yaw angle
PHIIN	Initial roll angle
WTIN	Initial missile total weight
WT	Current missile weight
XXTIN	Time of first print out
XXT	Time of next print out
KELL	Index indicating whether missile is on launcher
GRAV	Current value of gravity

TABLE 2. (Continued)

Fortran Name	Definition
TEMP	Current value temperature
PRSS	Current value of atmospheric pressure
P	Current value of atmospheric density
OM	Current value of speed of sound
WY	Boost phase lateral wind component
WY1	Coast phase lateral wind component
WX	Boost phase head wind component
WX1	Coast phase head wind component
WU	Component of head wind along longitudinal missile axis
WW	Component of head wind along vertical axis in missile coordinate system
VTOT	Total missile velocity
ALP	Angle of attack in vertical plane
ALPHA	Total angle of attack
UM	Current Mach number
ALTFT	Current altitude (positive up)
XT	Current missile thrust
T	Time table, independent variable for thrust table
A	Thrust table, f(T)
PSPIN	Roll acceleration due to spin motor
TTA	Time table, independent variable for PSPIN table
AR	PSPIN table, f(TTA)
TTA	Time table, independent variable for ASPIN table (Note both ASPIN and PSPIN will not be used in same simulation)
AK	ASPIN table, f(TTA)
ASPIN	Constant or programmed roll rate
CLP	Roll damping coefficient

TABLE 2. (Continued)

Fortran Name	Definition
UG	Mach number table, independent variable for CLP table
AO	CLP table, f(UG)
CLD	Roll moment coefficient due to fin deflection
UH	Mach number table, independent variable for CLD table
AP	CLD table, f(UH)
XK	Control force for system with direction control
TTB	Time table, independent variable for XK table
AL	XK table, f(TTB)
CMQ	Pitch damping coefficient
UE	Mach number table, independent variable for CMQ table
AN	CMQ table, f(UE)
CNA	Normal force coefficient
UB	Mach number table, independent variable for CNA table
AG	CNA table, f(UB), power on
CLO	Roll moment coefficient due to wraparound fin effect
UI	Mach number table, independent variable for CLO table
AS	CLO table, f(UI)
RNLM	Reynolds number per body length times Mach number
ZAL	Altitude table, independent variable for RNLM table
AU	RNLM table, f(ZAL)
BL	Missile body length (ft)
BLOD	Missile length divided by diameter (l/d)

TABLE 2. (Continued)

Fortran Name	Definition
RN	Reynolds number
CDF	Coefficient of drag due to skin friction
TB	Boost burn time
SPIMP	Specific impulse
TIMP	Total impulse
TPROP	Total propellant weight
DELT	Time integration step
XII	Initial transverse moment of inertia
XIB	Final transverse moment of inertia
TTO	Initial time for table values
XI	Current value of transverse moment of inertia
TLI	Initial thrust lever arm
TLF	Final thrust lever arm
TL	Current value of thrust lever arm
CLAI	Initial value of control lever arm
CLA	Current value of control lever arm
ROLLI	Current value of roll moment of inertia
ROLLA	Initial value of roll moment of inertia
ROLLB	Final value of roll moment of inertia
SL	Static margin
UA	Mach number table, independent variable for SL table
AI	SL table, power on, f(UA)
CD	Drag coefficient
UC	Mach number table independent variable for CD table, power on
AH	CD table, power on, f(UC)
UA	Mach number table, independent variable for SL table, power off
AJ	SL table, power off, f(UA)

TABLE 2. (Continued)

Fortran Name	Definition
UD	Mach number table, independent variable for CD table, power off
AM	CD table, power off, f(UD)
UJ	Mach number table, independent variable for CNA table, power off
AT	CNA table, power off, f(UJ)
XM	Missile mass
TFLD	Time of fin folding
DELMIN	Minimum time integration step
AX	Velocity along the x-axis in the earth coordinate system
AY	Velocity along the y-axis in the earth coordinate system
AZ	Velocity along the z-axis in the earth coordinate system
DRAG	Total missile drag
ACC	Total missile acceleration
THETS	Steady-state pitch angle after burn out
PSITS	Steady-state yaw angle after burn out
IBO	Index indicating whether to print out angular errors after burn
ZERR	Pitch angular error at burn out
YERR	Yaw angular error at burn out
XX	Number of time-steps per print out
TMAX	Maximum time - simulation stops after this time
LCS	Index indicating last case of data
QE	Quadrant elevation
S	Missile reference area
DTM	Thrust misalignment angle
PHIO	Initial thrust misalignment direction
DX	Missile diameter

TABLE 2. (Continued)

Fortran Name	Definition
ELL	Effective launcher length
DCANT	Fin cant angle
PCD	Percent effective drag coefficient-utilized in error budgets
J	Variable number to be integrated
V	Table of variables to be integrated-peculiar to subroutines DESUB and SOLVDE
W,WWW	Working arrays for integration routine
N	Number of differential equations
XX	Array of independent variables for interpolation purposes
YY	Array of dependent variables for interpolation purposes
X	Current value of independent variable
Y	Current value of dependent variable
TIME	Current value of time
U	Component of missile velocity along longitudinal axis
V	Component of missile velocity in yaw plane
W	Component of missile velocity in pitch plane
THET	Pitch angle
PSIO	Yaw angle
PHI	Roll angle
Q	Missile pitch rate
R	Missile yaw rate
ACC	Missile acceleration
VZ	Velocity along the z-axis in the earth coordinate system
VY	Velocity along the y-axis in the earth coordinate system
CNAI	Constant value of CNA before fins are unfolded

TABLE 2. (Concluded)

Fortran Name	Definition
SLI	Constant value of SL before fins are unfolded
VARTB (1)	Current value of time
VARTB (2)	Current value of W missile velocity component
VARTB (3)	Current value of U missile velocity component
VARTB (4)	Current value of V missile velocity component
VARTB (5)	Current value of R, missile yaw rate
VARTB (6)	Current value of Q, missile pitch rate
VARTB (7)	Current value of missile range
VARTB (8)	Current value of missile lateral displacement
VARTB (9)	Current value of missile altitude
VARTB (10)	Current value of missile pitch angle
VARTB (11)	Current value of missile yaw angle
VARTB (12)	Current value of missile roll rate
VARTB (13)	Current value of missile roll angle

TABLE 3. PROGRAM CONSTANT INPUTS

Fortran Name	Definition	Units
THETO	Initial pitch angle	rad
PSIO	Initial yaw angle	rad
PHIIN	Initial roll angle	rad
WTIN	Initial missile total weight	lb
TT	Current value of time	sec
WO	Initial velocity component of missile in pitch plane	ft/sec
U	Initial velocity component of missile along the longitudinal axis	ft/sec
VO	Initial velocity component of missile in yaw plane	ft/sec
RO	Initial yaw rate of missile	rad/sec
QO	Initial pitch rate of missile	rad/sec
PROLL	Initial roll rate of missile	rad/sec
XR	Initial range of missile in earth coordinate system	ft
Y	Initial lateral displacement in earth coordinate system	ft
Z	Initial vertical displacement in earth coordinate system	ft
XX	Number of time-steps per print out	–
XXTIN	Time of first print out	sec
TIMP	Total impulse	lb-sec
TPROP	Total propellant weight	lb
DELT	Time integration step	sec
XII	Initial transverse moment of inertia	$slug\text{-}ft^2$
XIB	Final transverse moment of inertia	$slug\text{-}ft^2$
TTO	Initial time for table values	sec
TLI	Initial thrust lever arm	cal
TLF	Final thrust lever arm	cal
WY	Boost phase lateral wind component	ft/sec
WY1	Coast phase lateral wind component	ft/sec

TABLE 3. (Continued)

Fortran Name	Definition	Units
WX	Boost phase head wind component	ft/sec
WX1	Coast phase head wind component	ft/sec
TFLD	Time of fin folding	sec
DELMIN	Minimum time integration step	sec
ROLLA	Initial value of roll moment of inertia	slug-ft^2
ROLLB	Final value of roll moment of inertia	slug-ft^2
CLAI	Initial value of control lever arm	cal
TB	Boost burn time	sec
IBO	Index indicating whether to print out angular errors after burn	0 - no print 1 - extra print
BL	Missile body length	ft
TMAX	Maximum time - simulation stops after this time	sec
LCS	Index indicating last case of data	0-not last case; 1-last case
QE	Quadrant elevation	rad
S	Missile reference area	ft^2
DTM	Thrust misalignment angle	rad
PHIO	Initial thrust misalignment direction	rad
DX	Missile diameter	ft
ELL	Effective launcher length	ft
DCANT	Fin cant angle	rad
PCD	Percent effective drag coeffieient utilized in error budgets	–
CNAI	Constant value of CNA before fins are unfolded	/rad
SLI	Constant value of SL before fins are unfolded	cal

TABLE 3. (Continued)

Fortran Name	Definition	Units
T	Time table, independent variable for thrust table	sec
A	Thrust table, f(T)	lb
TTA	Time table, independent variable for PSPIN table	sec
AR	PSPIN table, f(TTA)	rad/sec^2
TTA	Time table, independant variable for ASPIN table (Both ASPIN and PSPIN will not be used in same simulation.)	sec
AK	ASPIN table, f(TTA)	rad/sec
UA	Mach number table independent variable for SL table, power off	–
AJ	SL table, power off, f(UA)	cal
UD	Mach number table, independent variable for CD table, power off	–
AM	CD table, power off, f(UD)	–
UJ	Mach number table, independent variable for CNA table, power off	–
AT	CNA table, power off, f(UJ)	/rad
UC	Mach number table, independent variable for CD table, power on	–
AH	CD table, power on, f(UC)	–
UI	Mach number table, independent variable for CLO table	–
AS	CLO table, f(UI)	–
UG	Mach number table, independent variable for CLP table	–
AO	CLP table, f(UG)	/rad
UB	Mach number table, independent variable for CNA table	–
AG	CNA table, f(UB), power on	/rad
UE	Mach number table, independent variable for CMQ table	–

TABLE 3. (Concluded)

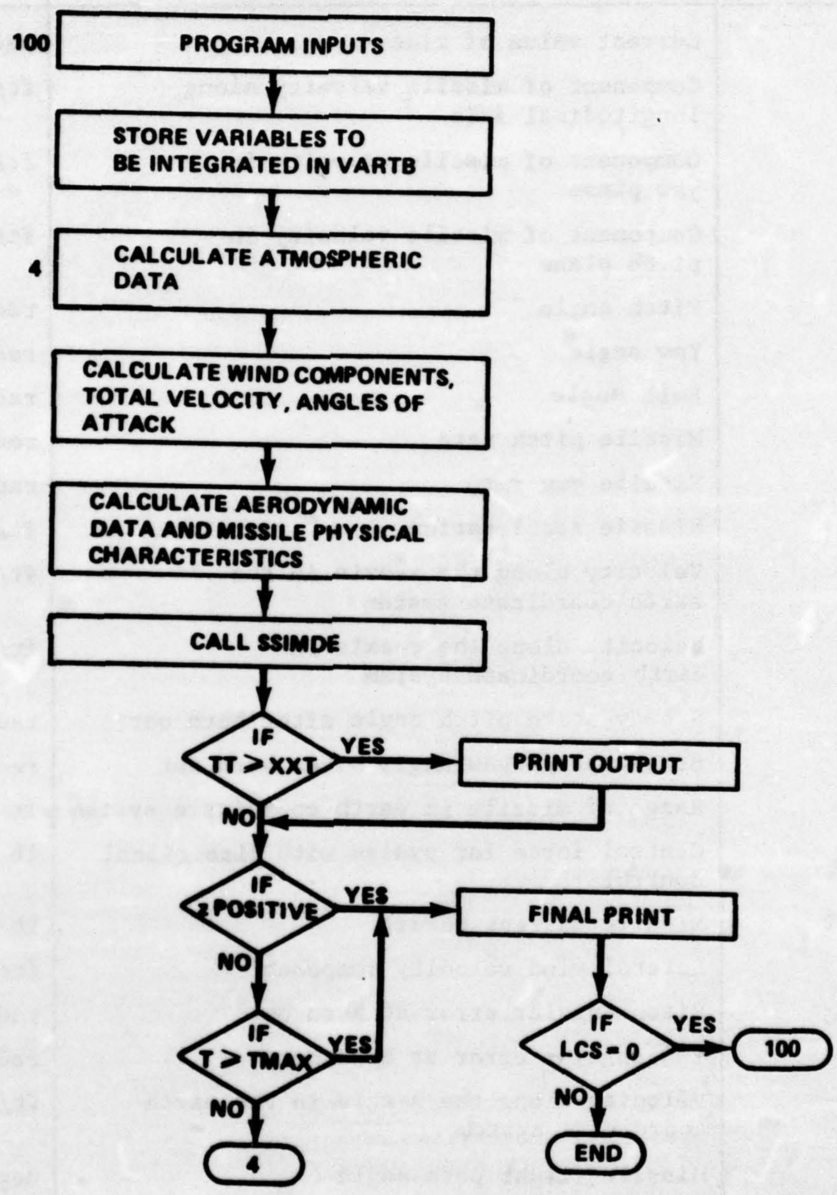
Fortran Name	Definition	Units
AN	CMQ table, f(UE)	/rad
TTB	Time table, independent variable for XK table	sec
AL	XK table, f(TTB)	lb
UH	Mach number table, independent variable for CLD table	–
AP	CLD table, f(UH)	/rad
UA	Mach number table, independent variable for SL table	–
AI	SL table, power on, f(UA)	cal
ZAL	Altitude table independent variable for RNLM table	ft
AU	RNLM table, f(ZAL)	/ft

TABLE 4. PROGRAM OUTPUT

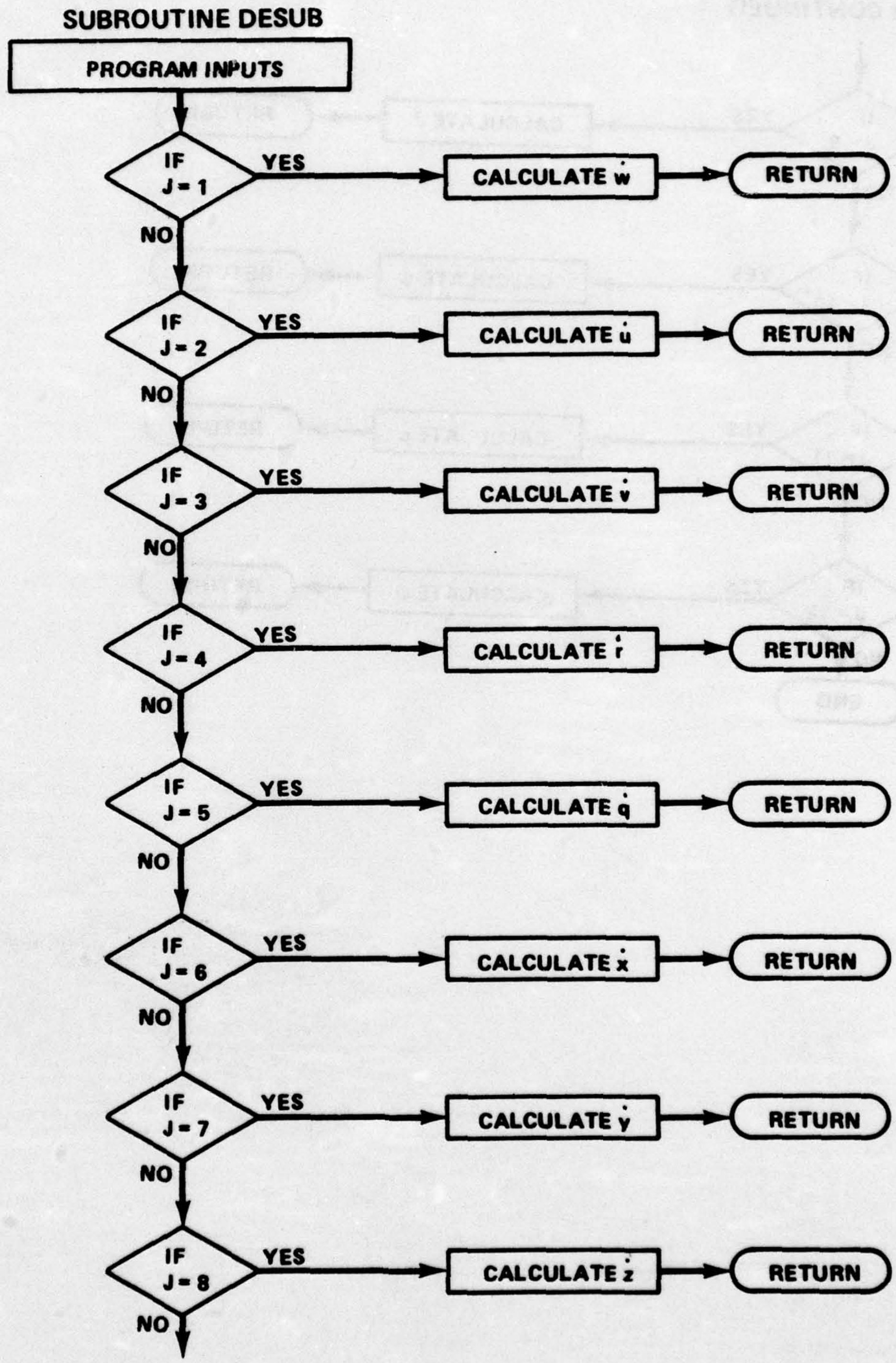
Fortran Nane	Definition	Units
TIME	Current value of time	sec
U	Component of missile velocity along longitudinal axis	ft/sec
V	Component of missile velocity in yaw plane	ft/sec
W	Component of missile velocity in pitch plane	ft/sec
THET	Pitch angle	rad
PSIO	Yaw angle	rad
PHI	Roll angle	rad
Q	Missile pitch rate	rad/sec
R	Missile yaw rate	rad/sec
ACC	Missile acceleration	ft/sec^2
VZ	Velocity along the z-axis in the earth coordinate system	ft/sec
VY	Velocity along the y-axis in earth coordinate system	ft/sec
THETS	Steady-state pitch angle after burn out	rad
PSITS	Steady-state yaw angle after burn out	rad
XR	Range of missile in earth coordinate system	ft
XK	Control force for system with directional control	lb
THRUST	Missile current thrust	lb
WY	Lateral wind velocity component	ft/sec
ZE	Pitch angular error at burn out	rad
YE	Yaw angular error at burn out	rad
XDOT	Velocity along the x-axis in the earth coordinate system	ft/sec
THA	Missile flight path angle	deg
ALP	Missile angle of attack in the vertical plane	rad

D. Flow Chart

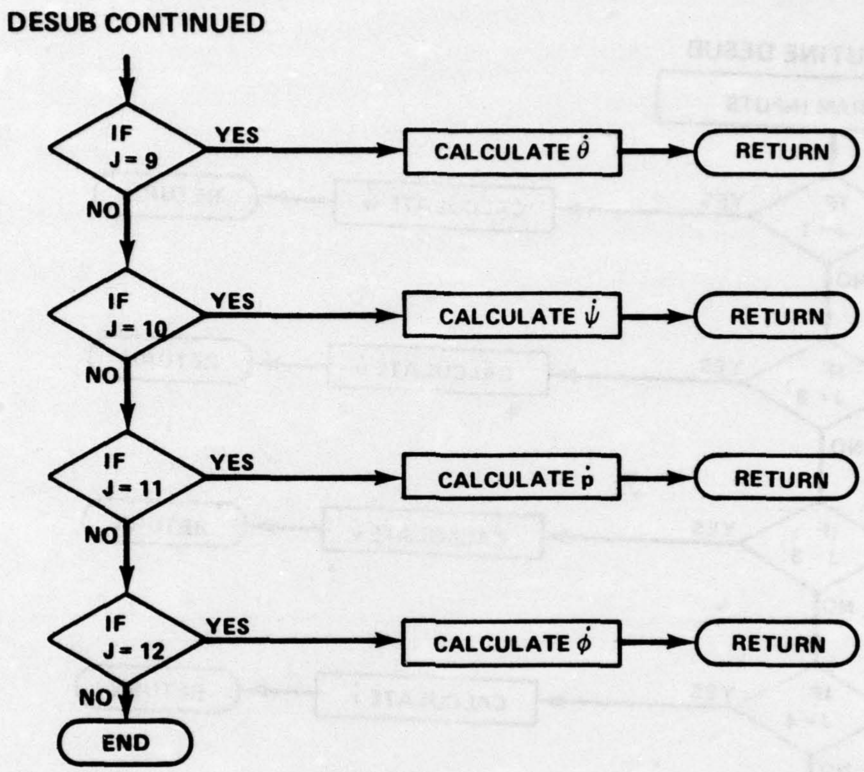
PROGRAM MAIN



E. Flow Chart


SUBROUTINE DESUB
PROGRAM INPUTS
IF J = 1
YES
CALCULATE ẇ
RETURN
NO
IF J = 2
YES
CALCULATE u̇
RETURN
NO
IF J = 3
YES
CALCULATE v̇
RETURN
NO
IF J = 4
YES
CALCULATE ṙ
RETURN
NO
IF J = 5
YES
CALCULATE q̇
RETURN
NO
IF J = 6
YES
CALCULATE ẋ
RETURN
NO
IF J = 7
YES
CALCULATE ẏ
RETURN
NO
IF J = 8
YES
CALCULATE ż
RETURN
NO

F. Flow Chart


DESUB CONTINUED
IF J = 9
YES
NO
CALCULATE $\dot{\theta}$
RETURN
IF J = 10
YES
NO
CALCULATE $\dot{\psi}$
RETURN
IF J = 11
YES
NO
CALCULATE $\dot{p}$
RETURN
IF J = 12
YES
NO
CALCULATE $\dot{\phi}$
RETURN
END

III. PROGRAM LISTING

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION  VARTB(13),WWW(3)
      DIMENSION T(4),A(4),UA(13),AI(13),UB(13),AG(13),UC(8),AH(8),
     *AJ(13),UG(2),AO(2),UH(2),AP(2),TTA(4),AR(4),AK(4),UI(7),AS(7),
     *UJ(13),AT(13),ZAL(4),AU(4),TTB(2),AL(2),UD(8),AM(8),UE(2),AN(2)
      COMMON GRAV,QE,P,S,CNA,AM,XT,DTM,PHI,PHIO,XK,CD,WV,SL,XI,DX,CMQ
      COMMON TL,CLA,ROLLI,APHI,ELL,KELL,WW,WU,PSPIN,CLP,CLD,DCANT,PDOT
      COMMON PCD,ASPIN,CLO
      EXTERNAL DESUB
      NAMELIST/NAMA/THETO,PHIIN,Z,Y,TT,U,XR,WTIN,QO,WO,RO,VO,PSIO,
     1 XXTIN,DTM,WY,ELL,QE,S,PHIO,DX,TLI,TLF,ROLLA,XX,TB,PROLL
     2XII,XIB,TIMP,TPROP,TTO,CLAT,DELT,TMAX,T,A,UA,AI,UB,AG,UC,AH,AJ,
     3TTB,AL,UD,AM,UE,AN,DELMIN,ROLLB,IHO,WX,LCS,DCANT,UG,AO,UH,AP,
     4TTA,AR,WY1,WX1,PCD,AK,UI,AS,UJ,AT,TFLO,ZAL,AU,BL,CNA1,SLI
      BLOD=BL/DX
      LCS=0
C     ATMOSPHERIC DATA AT SEALEVEL
      TEMPO=518.688
      PRSSO=2116.217
      GRAVO=32.17404
      RE=GRAVO*.1637653
  100 READ(5,NAMA)
      WRITE(6,1097)
 1097 FORMAT(1H1)
      WRITE(6,NAMA)
C     INITIAL CONDITIONS
      THETA=QE*57.3
      IC=0
      VARTB(1)=TT
      VARTB(2)=WO
      VARTB(3)=U
      VARTB(4)=VO
      VARTB(5)=RO
      VARTB(6)=QO
      VARTB(7)=XR
      VARTB(8)=Y
      VARTB(9)=Z
      VARTB(10)=THETO
      VARTB(11)=PSIO
      VARTB(12)=PROLL
      VARTB(13)=PHIIN
      WT=WTIN
      XXT=XXTIN
      KELL=1
C     ATMOSPHERIC DATA CALCULATED
    4 CAD=1.-VARTB(9)/20855531.5
      GRAV=GRAVO*(1./CAD)**2
      ALT=-VARTB(9)/CAD
      IF(ALT-36089.2388) 45,45,46
   45 TEMP=TEMPO+(-.00356616*ALT)
      PRSS=PRSSO*(TEMP/TEMPO)**RE
      P=28.9660*PRSS/(49719.6257*TEMP)
      VS=(TEMP*2403.075)**.5
      GO TO 50
   46 TEMP=389.9878
      PRSS=472.6765*EXP(GRAVO*(-.00000149)*(ALT-36089.2388))
      P=28.9660*PRSS/19390047.44
      VS=968.075
   50 WV=WY
      IF(TT.GT.TB)WV=WY1
C     AERODYNAMIC DATA CALCULATED
      IF(TT.GT.TB)WX=WX1
      WU=WX*COS(VARTB(10)+QE)
```

```
      WW=WX*SIN(VARTB(10)+QE)
      VTOT=SQRT((VARTB(3)-WL)**2+(VARTB(4)-WV)**2+(VARTB(2)-WW)**2)
      ALP=(VARTB(2)-WW)/VTOT
      ALPHA=SQRT((VARTB(2)-WW)**2+(VARTB(4)-WV)**2)/VTOT
      UM=VTOT/OM
      ALTFT=-VARTB(9)
      CALL XINT (T,A,TT,XT)
      CALL XINT(TTA,AW,TT,PSPIN)
      CALL XINT(TTA,AK,TT,ASPIN)
      CALL XINT(UG,AO,UM,CLP)
      CALL XINT(UH,AP,UM,CLD)
      CALL XINT(TTB,AL,TT,XK)
      CALL XINT (UE,AN,UM,CMQ)
      CALL XINT(UB,AG,UM,CNA)
      CALL XINT(UI,AS,UM,CLO)
      CALL XINT(ZAL,AU,ALTFT,RNLM)
      RN=RNLM*UM*HL
      CDF=.296*HLOD*(1.0392-.1051*UM+.0035*UM**2)/(RN**.2)
      IF(TT.GT.TB) GO TO 101
C     PHYSICAL CHARACTERISTICS UPDATED
    6 SPIMP=TIMP/TPROP
      WTDOT=XT/SPIMP
      WT=WT-WTDOT*DELT
      BBB=(XIB-XII)/(TB-TTO)
      XI=XII+BBB*TT
      BBT=(TLF-TLI)/(TB-TTO)
      TL=TLI+BBT*TT
      CLA=ABS(CLAI-TL)
      ROLLI=ROLLA+((TT-TTO)/(TB-TTO))*(ROLLB-ROLLA)
      CALL XINT (UA,AI,UM,SL)
      CALL XINT (UC,AH,UM,CD)
      GO TO 5
  101 CALL XINT (UA,AJ,UM,SL)
      CALL XINT (UD,AM,UM,CD)
      CALL XINT(UJ,AT,UM,CNA)
    5 XM=WT/GRAV
      CD=CD+CDF
      IF(TT.GT.TFLD)GO TO 7
      CNA=CNAI
      SL=SLI
    7 CONTINUE
C     INTEGRATION SUBROUTINE
      IF(VARTB(13).GE.6.2832)VARTB(13)=VARTB(13)-6.2832
      IF(VARTB(13).LE.-6.2832)VARTB(13)=VARTB(13)+6.2832
      CALL SSIMDE(VARTB,WWW,DELT,DELMIN,12,IC,DESUB)
      TT=VARTB(1)
      AX=VARTB(3)*COS(VARTB(10)+QE)+VARTB(2)*SIN(VARTB(10)+QE)*
     *COS(VARTB(11))-VARTB(4)*SIN(VARTB(11))
      AY=VARTB(4)+VARTB(3)*SIN(VARTB(11))
      AZ=VARTB(2)*COS(VARTB(10)+QE)-VARTB(3)*SIN(VARTB(10)+QE)
      THETA=ATAN((-AZ)/AX)*57.3
      DRAG=.5*P*S*CD*(VARTB(3)**2)
      ACC=(XT-DRAG)/XM
      IF(TT.LT.TB)GO TO 200
C     STEADY STATE ANGLES
      THETS=VARTB(10)-VARTB(2)/VARTB(3)-VARTB(6)*XI/(DX*XM*VARTB(3)*SL)+
     1(VARTB(2)-WZ)*P*S*DX*CMQ/(4.*XM*VARTB(3)*SL)
      PSITS=VARTB(11)+VARTB(4)/(VARTB(3)*COS(QE)+VARTB(2)*SIN(QE))-
     1VARTB(5)*XI/(DX*XM*VARTB(3)*SL*COS(QE))-(VARTB(4)-WY)*P*S*DX*CMQ/
     2(4.*XM*VARTB(3)*SL*COS(QE))
      IF(TT.GE.XXT)WRITE(6,1099)THETS,PSITS
  200 IF(IHO.EQ.0)GO TO 201
C     ANGULAR ERRORS AT BOOST BURN OUT
      VZ=VARTB(2)*COS(VARTB(10))-VARTB(3)*SIN(VARTB(10))
      VY=VARTB(3)*COS(VARTB(10))*SIN(VARTB(11))+VARTB(4)*COS(VARTB(11))
     1+VARTB(2)*SIN(VARTB(10))*SIN(VARTB(11))
```

```
      ZERR=VZ/VTOT
      YERR=VY/VTOT
      IF(TT.GE.XXT)WRITE(6,1098)VZ,VY,ZERR,YERR
  201 IF(TT-XXT)91,90,90
   90 WRITE(6,1090)VARTR(1),VARTR(3),VARTR(7),XM
      WRITE(6,1091)VARTR(10),VARTR(2),AZ,VARTR(9)
      WRITE(6,1092)VARTR(11),VARTR(4),AY,VARTR(8)
      WRITE(6,1093)VARTR(13),VARTR(6),VARTR(5),DRAG
      WRITE(6,1095)XT,WY,XK,ACC
      WRITE(6,1100)AX,THETA,ALP
      WRITE(6,1096)
 1090 FORMAT(8H TIME=   ,E14.7,2X,4HU=   ,E14.7,2X,4HXR= ,E14.7,2X,
     17HMASS=   ,E14.7)
 1091 FORMAT(8H THET=   ,E14.7,2X,4HW=   ,E14.7,2X,4HAZ= ,E14.7,2X,
     17HZ=      ,E14.7)
 1092 FORMAT(8H PSIO=   ,E14.7,2X,4HV=   ,E14.7,2X,4HAY= ,E14.7,2X,
     17HY=      ,E14.7)
 1093 FORMAT(8H PHI=    ,E14.7,2X,4HQ=   ,E14.7,2X,4HR=  ,E14.7,2X,
     17HDRAG=   ,E14.7)
 1095 FORMAT(8H THRUST=,E14.7,2X,4HWY= ,E14.7,2X,4HXK= ,E14.7,2X,4HACC=,
     1E14.7)
 1096 FORMAT(1H )
 1098 FORMAT(8H VZ=     ,E14.7,2X,4HVY= ,E14.7,2X,4HZE= ,E14.7,2X,4HYE= ,
     1E14.7)
 1099 FORMAT(8H THETS= ,E14.7,2X,4HPTS=,E14.7)
 1100 FORMAT(8H XDOT=   ,E14.7,2X,4HTHA=,E14.7,2X,4HALP=,E14.7)
      XXT=XXT+XX*DELT
   91 CONTINUE
C     PROGRAM STOPS ON NEGATIVE ALTITUDE OR MAX TIME
      IF(VARTR(9).GT.10.)GO TO 300
      IF(TT-TMAX)4,300,300
  300 WRITE(6,1090)VARTR(1),VARTR(3),VARTR(7),XM
      WRITE(6,1091)VARTR(10),VARTR(2),AZ,VARTR(9)
      WRITE(6,1092)VARTR(11),VARTR(4),AY,VARTR(8)
      WRITE(6,1093)VARTR(13),VARTR(6),VARTR(5),DRAG
      WRITE(6,1095)XT,WY,XK,AU
      WRITE(6,1098)VZ,VY,ZERR,YERR
C     LAST SET OF DATA SIGNALLED BY LCS=1
  301 IF(LCS.EQ.0)GO TO 100
      CALL EXIT
      END
      SUBROUTINE DESUB(V,F,J)
      COMMON GRAV,QE,P,S,CNA,XM,XT,DTM,PHI,PHIO,XK,CD,WV,SL,XI,DX,CMQ
      COMMON TL,CLA,ROLLI,APHI,ELL,KELL,WW,WU,PSPIN,CLP,CLD,DCANT,PDOT
      COMMON PCD,ASPIN,CLO
      DIMENSION V(13)
      IF(V(7).GT.ELL)KELL=2
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120),J
   10 F=0
      GO TO (12,11),KELL
   11 F=GRAV*COS(V(10)+QE)+V(5)*V(3)-(V(3)**2)*P*S*CNA*(V(2)-WW)/
     1(2.*XM*(V(3)-WU))+
     2(XT*DTM*SIN(V(13)+PHIO)-XK*V(10))/XM
   12 RETURN
   20 F=(XT-.5*P*S*PCD*CD*V(3)**2)/XM+V(4)*V(5)-V(2)*V(6)
      GO TO (22,21), KELL
   21 F=(XT-.5*P*S*PCD*CD*V(3)**2)/XM-GRAV*SIN(V(10)+QE)+V(4)*V(5)-V(2)*
     1V(6)
   22 RETURN
   30 F=0
      GO TO (32,31),KELL
   31 F=(-1.*V(5)*V(3))-(V(4)-WV)*(V(3)**2)*P*S*CNA/(2.*XM*(V(3)-WU))+
     1(XT*DTM*COS(V(13)+PHIO)+XK*V(11))/XM
   32 RETURN
   40 F=0
      GO TO (42,41),KELL
```

```
41 F=(V(3)**2)*P*S*DX*CNA*(V(4)-WV)*SL/(2.*XI*(V(3)-WU))-
  1 .5*(DX**2)*P*S*V(5)*V(3)*
  2 CMQ/(2.*XI)+(-1.*XT*DTM*COS(V(13)+PHI0)*TL*DX-1.*DX*CLA*XK*
  3V(11))/XI+ROLLI*V(12)*V(6)/XI
42 RETURN
50 F=0
   GO TO(52,51),KELL
51 F=(-1.*(V(3)**2)*P*S*DX*CNA*(V(2)-WW)*SL)/(2.*XI*(V(3)-WU))-
  1(.5*P*S*(DX**2)*V(6)*
  2V(3)*CMQ)/(2.*XT)+(DX*TL*XT*DTM*SIN(V(13)+PHI0)-DX*XK*V(13)*CLA)/
  3XI-ROLLI*V(12)*V(5)/XI
52 RETURN
60 F=V(3)*COS(V(10)+QE)+V(2)*SIN(V(10)+QE)*COS(V(11))-V(4)*SIN(V(11))
   RETURN
70 F=0
   GO TO (72,71),KELL
71 F=V(3)*COS(V(10)+QE)*SIN(V(11))+V(2)*SIN(V(10)+QE)*SIN(V(11))+
  1V(4)*COS(V(11))
72 RETURN
80 F=V(2)*COS(V(10)+QE)-V(3)*SIN(V(10)+QE)
   RETURN
90 F=0
   GO TO (92,91),KELL
91 F=V(6)
92 RETURN
100 F=0
   GO TO (102,101),KELL
101 F=V(5)/COS(V(10)+QE)
102 RETURN
110 F=-P*S*(DX**2)*V(3)*V(12)*CLP/(4.*ROLLI)+P*S*(V(3)**2)*DX*CLD*
  1DCANT/(2.*ROLLI)+PSPIN+.5*P*S*DX*CLO*V(3)**2/ROLLI
   F=0.   <---
   PDOT=F        REMOVE THESE TWO CARDS IF THE ROLL
   RETURN        RATE IS NOT CONSTANT OR PROGRAMMED
120 F=V(12)
   F=ASPIN <---
   RETURN
   END
   SUBROUTINE SOLVDE(V,W,DEL,DELMIN,N,IC,DESUB)
C
C        GIVEN AN N TH ORDER DIFFERENTIAL EQUATION
C        D(N)Y/DX(N) = F(X,Y,YP,Y2P,....,Y(N-1)P) WHERE P DENOTES PRIME
C
C        1 V       VARIABLE TABLE IE.
C                     V(1)      = X
C                     V(2)      = Y
C                     V(3)      = Y PRIME
C                     V(4)      = Y TWO PRIME
C                     .....................
C                     V(N+1)    = Y (N-1) PRIME
C                     V(N+2)    = D(N)Y/DX(N) UPON RETURN
C        2 W       WORKING ARRAY. A DIFFERENT ARRAY MUST BE USED FOR
C                  EACH DIFFERENT EQUATION OR SYSTEM OF EQUATIONS BEING
C                  EVALUATED. THE DIMENSION MUST BE AT LEAST W(3).
C        3 DEL     STEP SIZE FOR INDEPENDENT VARIABLE. AN INCREASING-
C                  DECREASING VARIABLE STEP SIZE (H) IS USED INTERNALLY
C                  IN THE ROUTINE. RETURN TO THE CALLING PROGRAM WILL
C                  BE AT V(1)+DEL.
C        4 DELMIN  MINIMUN VALUE TO BE USED FOR (H) WHEN DECREASING
C                  THE INTEGRATION INCREMENT.
C        5 N       ORDER OF THE EQUATION TO BE SOLVED.
C        6 IC      USED FOR INTERNAL CONTROL. THIS VARIABLE MUST BE
C                  SET EQUAL TO ZERO FOR THE FIRST INTEGRATION STEP.
C                  A DIFFERENT VARIABLE MUST BE USED FOR EACH DIFFERENT
C                  EQUATION OR SYSTEM OF EQUATIONS BEING EVALUATED. DO
C                  NOT USE A LITERAL SINCE IC IS CONTINUALLY CHANGED.
```

```
C         7 DESUB  NAME OF ROUTINE CONTAINING THE EQUATION TO BE SOLVED.
C                  THE NAME USED IN THE CALLING PROGRAM MUST BE DEFINED
C                  IN A EXTERNAL STATEMENT IN THE CALLING PROGRAM.
C
      SUBROUTINE SSIMDE(V,W,DEL,DELMIN,N,IC,DESUB)
C
C         GIVEN THE FIRST ORDER SYSTEM OF DIFFERENTIAL EQUATIONS
C            DY1/DX = F1(X,Y1,...,YN)
C            DY2/DX = F2(X,Y1,...,YN)
C            ........................
C            DYN/DX = FN(X,Y1,...,YN)
C
C         1 V         VARIABLE TABLE IE.
C                        V(1)       = X
C                        V(2)       = Y1
C                        V(3)       = Y2
C                        ...........
C                        V(N+1) = YN
C
C         2 W         SAME AS 2 ABOVE.
C         3 DEL       SAME AS 3 ABOVE.
C         4 DELMIN SAME AS 4 ABOVE.
C         5 N         NUMBER OF SIMULTANEOUS EQUATIONS TO BE SOLVED.
C         6 IC        SAME AS 6 ABOVE.
C         7 DESUB  SAME AS 7 ABOVE.
C
C         THE VARIABLES C AND VS IN THIS PROGRAM MUST BE DIMENSIONED
C         AT LEAST C(4,N) AND VS(N+2).
C                                             10 APRIL 69
C                                             RALPH SELLERS
C
C     DOUBLE PRECISION V,W,DEL,DELMIN,C,VS,TSTOP,H,H2,F,TMP1
C
      EXTERNAL DESUB
      DIMENSION V(1),W(1),C(4,15),VS(17)
C
      ISYS = 0
      NP1 = N+1
      NP2 = NP1+1
      NTOT = NP2
      GO TO 3
      ENTRY SSIMDE
      ISYS = 1
      NP1 = N+1
      NP2 = NP1+1
      NTOT = NP1
C         COMPUTE STOP FOR THIS INTEGRATION STEP
    3 TSTOP = V(1) + DEL
C         CHECK FOR FIRST TIME INTO ROUTINE
   10 IF(IC .GT. 0)   GO TO 20
C         FIRST TIME IN
   13 H = DEL
      NH = 0
      W(2) = DEL
      GO TO 60
C         RESTART IF INPUT DEL HAS CHANGED
   20 IF(DEL .NE. W(2))   GO TO 13
      H = W(1)
      NH= W(3)
C         TEST FOR DOUBLING H. INCREASE TEST AS NUMBER OF HALVINGS
C         INCREASES.
   40 IF(IC .LT. 10*(NH+1))   GO TO 60
      IC = 1
C         DOUBLE H
   50 H2 = 2.0*H
      IF(V(1)+H2 .GT. TSTOP)   GO TO 60
```

```
      IF(H2 .LE. DEL)   GO TO 53
      GO TO 60
   53 H = H2
C          INCREASE COUNTER
   60 IC = IC + 1
C          INTEGRAT USING R-K
C          SAVE V TABLE VALUES IN VS ARRAY
  105 DO 106  I=1,NTOT
  106 VS(I) = V(I)
C          FIRST PASS THRU R-K COMPUTATION
  110 DO 119  J=1,N
      IF(ISYS .NE. 0)   GO TO 115
      IF(J .EQ. N   )   GO TO 115
      F = V(J+2)
      GO TO 119
  115 CALL DESUB(V,F,J)
  119 C(1,J) = F * H
C          SECOND AND THIRD PASS THRU R-K CALCULATION
  120 V(1) = VS(1) + .5*H
      DO 150  I=2,3
      DO 130  J=2,NP1
  130 V(J) = VS(J) + .5*C(I-1,J-1)
      DO 140  J=1,N
      IF(ISYS .NE. 0)   GO TO 135
      IF(J .EQ. N   )   GO TO 135
      F = V(J+2)
      GO TO 140
  135 CALL DESUB(V,F,J)
  140 C(I,J) = F * H
  150 CONTINUE
      IF(.5*H .LT. DELMIN)   GO TO 160
C          TEST FOR HALVING H
      DO 153  J=1,N
      TMP1 = (C(2,J)-C(3,J)) / (C(1,J)-C(2,J))
      IF(ABS(TMP1) .GT. .025)   GO TO 155
  153 CONTINUE
      GO TO 160
C          HALVE H
  155 H = .5*H
      NH = NH+1
      IC = 1
      DO 157  J=1,NTOT
  157 V(J) = VS(J)
      GO TO 110
C          FOURTH PASS THRU R-K CALCULATION
  160 V(1) = VS(1) + H
      DO 161  J=2,NP1
  161 V(J) = VS(J) + C(3,J-1)
      DO 170  J=1,N
      IF(ISYS .NE. 0)  GO TO 165
      IF(J .EQ. N   )  GO TO 165
      F = V(J+2)
      GO TO 170
  165 CALL DESUB(V,F,J)
  170 C(4,J) = F * H
C          UPDATE V TABLE
      DO 180  J=1,N
  180 V(J+1)=VS(J+1)+(C(1,J) + 2.0D0*(C(2,J)+C(3,J)) + C(4,J))/ 6.0D0
C          TEST FOR END OF INTEGRATION STEP
  200 IF(V(1) + H/2.0 .GT. TSTOP)   GO TO 220
      GO TO 40
C          END OF REQUIRED INTEGRATION
  220 V(1) = TSTOP
      IF(ISYS .NE. 0)    GO TO 999
      CALL DESUB(V,V(NP2),1)
  999 CONTINUE
```

```
      W(1) = H
      W(2) = DEL
      W(3) = NH
C
      RETURN
      END
      SUBROUTINE XINT(XX,YY,X,Y)
      DIMENSION XX(15),YY(15)
      I=1
    5 IF(X.LT.XX(I+1))GO TO 10
      I=I+1
      GO TO 5
   10 Y=YY(I)+((X-XX(I))/(XX(I+1)-XX(I)))*(YY(I+1)-YY(I))
      RETURN
      END
000000000000000000000
```

DISTRIBUTION

	No. of Copies
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	6
Commander US Army Materiel Development and Readiness Command	
ATTN: DRCRD	1
DRCDL	1
5001 Eisenhower Avenue Alexandria, Virginia 22333	
DRCPM-RSE, Mr. Sullivan	1
Mr. Oldham	1
DRSMI-R, Dr. McDaniel	1
Dr. Kobler	1
-LP, Mr. Voigt	1
-FR, Mr. Strickland	1
-RF, Mrs. Stamps	5
-RBD	3
-RPR (Record Set)	1
(Reference Copy)	1